Calcul de valeurs propres

VIET HUNG NGUYEN - FABIEN RICO
Hung.Nguyen@lip6.fr

EPU Pierre et Marie Curie - *Science de la Terre*

Introduction

Décomposition QR

Il existe différents moyens de décomposer une matrice quelconque. L'une des plus utilisée est la décomposition QR.

Théorème *Soit A une matrice quelconque de $\mathcal{M}_{n,m}(\mathbb{C})$.*
$\exists Q \in \mathcal{M}_{m,m}$ *une matrice* orthogonale*,*
$\exists R \in \mathcal{M}_{n,m}(\mathbb{C})$ *une matrice* triangulaire supérieure *telles que*

$$A = QR$$

- On peut utiliser cette décomposition pour résoudre le système linéaire
  $Ax = b \quad \Leftrightarrow \quad QRx = b \quad \Leftrightarrow \quad Rx = {}^{t}Qb$
- A n'est pas forcement carrée (système surdéterminé).
- La décomposition n'est pas unique.

Rappel

Définition (Matrice orthogonales) *On appelle* matrice orthogonale *une matrice dont les colonnes sont* orthonormées*. C'est à dire les matrices O telles que :*

$${}^{t}OO = I$$

- Les matrices orthogonales sont les matrices de changement de bases orthogonales.
- Si O est orthogonale, $det(O) = \pm 1$.
- Elles ne changent pas la norme associée au produit scalaire : $\forall u \in \mathbb{R}^m$

$$\|Ou\| = \|u\|$$

- Le produit de deux matrices orthogonales est une matrice orthogonale :

$${}^{t}(OO')OO' = {}^{t}O'{}^{t}OOO' = I$$

Exemple

Par exemple :
- Matrices de rotation,

$$A = \begin{pmatrix} \cos\theta & -\sin(\theta) \\ \sin\theta & \cos(\theta) \end{pmatrix}$$

- Matrices de permutation,

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

- Matrices de symétrie.

Intéret

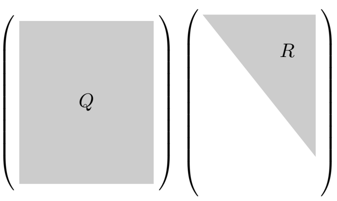
Comme la décomposition LU, cette décomposition transforme une matrice quelconque en produit de matrices plus simples.

$$A = QR$$

- Une matrice orthogonale est facile à inverser (${}^{t}Q = Q^{-1}$).
  ⇒*cela peut être utilisé pour la résolution d'équations linéaires : au lieu de résoudre $Ax = b$, on résout $Rx = {}^{t}Qb$*
- Cela fonctionne pour les matrices rectangulaires
  ⇒*résolution d'équations linéaires surdéterminés (avec plus d'équations que d'inconnues)*
- Les matrices orthogonales ont de bonnes propriétés :
  - le déterminant de A est égale à celui de R ;
  - multiplier par une matrice orthogonale ne change pas la norme (Problème des moindres carrés) ;
  - le conditionnement de A est égale à celui de R.
- Cette décomposition est aussi utilisée pour le calcul des valeurs propres d'une matrice (voir cours).
- ...

Le problème

On dispose d'une matrice A que l'on veut décomposer en Q orthogonale et R triangulaire supérieure :

$A =$ 

- Le problème est le même que de trouver H orthogonale telle que :

$$HA = R$$

  car $Q = {}^{t}H$
- On procède par étapes comme pour le pivot de GAUSS

Décomposition $\mathbf{QR}$

Étape de la décomposition

- On procède de la même façon que le pivot de GAUSS

$$H_{n-1}\dots H_2 H_1 A \quad = \quad \begin{pmatrix} & r_{ij} \\ & \end{pmatrix}.$$

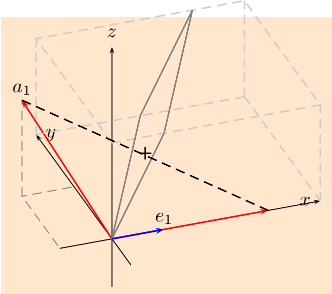
- Mais les matrices H_i sont des matrices orthogonales, donc $H_i^{-1} = {}^{t}H_i$
- Alors :

$$A \quad = \quad {}^{t}(H_{n-1}\dots H_1)R$$
$$Q \quad = \quad {}^{t}H_1\,{}^{t}H_2\dots{}^{t}H_{n-1}$$

Introduction
Décomposition $\mathbf{QR}$
Étape de la décomposition
Symétries
Transformation de HOUSEHOLDER
Comment trouver u
Première itération
Itérations suivantes
Algorithme
Introduction
Algorithme de la puissance itérée
Méthode de déflation
Méthode de la puissance inverse
Conditions de convergence
Matrices semblables
Conclusion

Symétries

- Soit $a_1, a_2, \dots, a_n$ les vecteurs formés par les colonnes de A.
- La matrice H_1 doit envoyer le vecteur a_1 sur le premier vecteur de la base canonique : e_1
- Pour cela on utilise la symétrie par le plan qui passe entre ces deux vecteurs.



Introduction
Décomposition $\mathbf{QR}$
Étape de la décomposition
Symétries
Transformation de HOUSEHOLDER
Comment trouver u
Première itération
Itérations suivantes
Algorithme
Introduction
Algorithme de la puissance itérée
Méthode de déflation
Méthode de la puissance inverse
Conditions de convergence
Matrices semblables
Conclusion

Transformation de HOUSEHOLDER

A quoi ressemble une matrice de symétrie ?

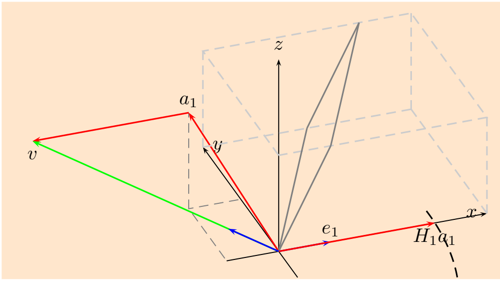
Définition (Matrice de HOUSEHOLDER) *On appelle* matrice de HOUSEHOLDER*, une matrice de la forme :*

$$H \quad = \quad I - 2u\,{}^{t}u, \text{ où } u \in \mathbb{R}^n \text{ avec } \|u\| = 1$$

Toute matrice de HOUSEHOLDER est symétrique et orthogonale. La matrice de HOUSEHOLDER est la matrice de la symétrie par rapport à l'hyperplan perpendiculaire à u.

Introduction
Décomposition $\mathbf{QR}$
Étape de la décomposition
Symétries
Transformation de HOUSEHOLDER
Comment trouver u
Première itération
Itérations suivantes
Algorithme
Introduction
Algorithme de la puissance itérée
Méthode de déflation
Méthode de la puissance inverse
Conditions de convergence
Matrices semblables
Conclusion

Comment trouver u



- La symétrie conserve la norme, donc $H_1(a_1) = \|a_1\|\,e_1$
- Soit $\alpha = \|a_1\|$, et $v = a_1 - \alpha e_1$, v est perpendiculaire au plan de symétrie.
- On peut choisir $u = \frac{1}{\|v\|}v$

Introduction
Décomposition $\mathbf{QR}$
Étape de la décomposition
Symétries
Transformation de HOUSEHOLDER
Comment trouver u
Première itération
Itérations suivantes
Algorithme
Introduction
Algorithme de la puissance itérée
Méthode de déflation
Méthode de la puissance inverse
Conditions de convergence
Matrices semblables
Conclusion

Première itération

Pour une étape,

- On pose $\alpha = \|a_1\|$ et $v = a - \alpha e_1$
- $$\begin{aligned} \|v\|^2 &= {}^{t}(a - \alpha e_1)(a - \alpha e_1) \\ &= ({}^{t}a - \alpha\,{}^{t}e_1)(a - \alpha e_1) \\ &= {}^{t}aa - \alpha\,{}^{t}e_1 a - \alpha\,{}^{t}a e_1 + \alpha^2\,{}^{t}e_1 e_1 \\ &= 2\alpha^2 - 2\alpha\,{}^{t}a e_1 \end{aligned}$$
- En posant $\beta = \alpha^2 - \alpha\,{}^{t}a e_1 = \frac{\|v\|^2}{2}$

$$\begin{aligned} H_1 &= I - \frac{2}{\|v\|^2} v\,{}^{t}v \\ &= I - \frac{1}{\beta} v\,{}^{t}v \end{aligned}$$

Alors, on pose $A^2 = H_1 A$

Introduction
Décomposition $\mathbf{QR}$
Étape de la décomposition
Symétries
Transformation de HOUSEHOLDER
Comment trouver u
Première itération
Itérations suivantes
Algorithme
Introduction
Algorithme de la puissance itérée
Méthode de déflation
Méthode de la puissance inverse
Conditions de convergence
Matrices semblables
Conclusion

Itérations suivantes

Pour continuer le processus,

- On considère la matrice réduite $\tilde{A}^2$ qui est obtenue en supprimant la première ligne et la première colonne de A^2.
- Alors, on peut appliquer le même processus pour trouver H_2 qui envoie la première colonne de $\tilde{A}^2$ sur e_2
- On recommence avec $\tilde{A}^2, A^3 \dots$ jusqu'à obtenir $R = A^n$.

$$\begin{pmatrix} 1 & & & \\ & \ddots & & \\ & & 1 & \\ & & & h_{n-1} \end{pmatrix} \dots \begin{pmatrix} 1 & 0 \cdots 0 \\ 0 & \\ \vdots & h_2 \\ 0 & \end{pmatrix} \begin{pmatrix} I - \frac{1}{\beta_1} v_1\,{}^{t}v_1 \end{pmatrix} A \quad = \quad \begin{pmatrix} & r_{ij} \\ & \end{pmatrix}.$$

Algorithme

Données : A
début
  $H \leftarrow I$
  pour k=1,…, n-1 **faire**
    $\alpha \leftarrow \sqrt{\sum_{i=k}^{m} A_{ik}^2}$
    $\beta \leftarrow \alpha^2 - \alpha A_{kk}$
    // construction du vecteur v
    $v_k \leftarrow A_{kk} - \alpha$
    pour i=k+1,…, m **faire**
      $v_i \leftarrow A_{ik}$
    pour j=k,…, n **faire**
      // Construction de A^{k+1}
      $c \leftarrow \frac{1}{\beta}\sum_{i=k}^{m} v_i A_{ij}$
      pour i=k,…, m **faire**
        $A_{ij} \leftarrow A_{ij} - c v_i$
    pour j=1,…, m **faire**
      // Construction de $H = H_k \cdots H_2 H_1$
      $c \leftarrow \frac{1}{\beta}\sum_{i=k}^{m} v_i H_{ij}$
      pour i=k,…, m **faire**
        $H_{ij} \leftarrow H_{ij} - c v_i$
  $Q \leftarrow {}^{t}H$
fin
retourner Q, $R = A$

Introduction

Introduction
Décomposition $\mathbf{QR}$
Introduction
Le problème
Algorithme de la puissance itérée
Méthode de déflation
Méthode de la puissance inverse
Conditions de convergence
Matrices semblables
Conclusion

Le problème

Soit une matrice A, répondant à certaines conditions par exemple :

- A est symétrique, hermitienne
- A est diagonalisable
- Les valeurs propres de A sont toutes différentes

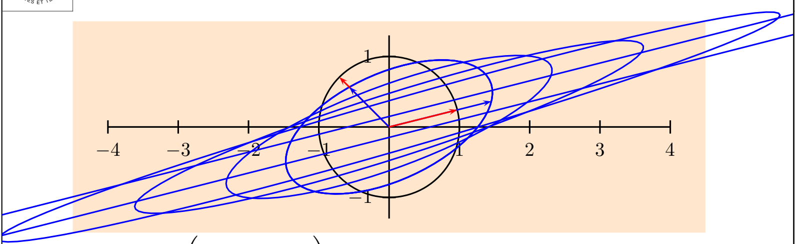
On cherche à obtenir les valeurs propres et les vecteurs propres de A.

- Méthodes basées sur le polynôme caractéristique
- Méthodes basées sur les matrices semblables
- Méthodes itératives

Les méthodes itératives sont basées sur une suite de vecteurs $(x_k)_{k\in\mathbb{N}}$ qui tendent vers un vecteur propre.

Comment construire cette suite ?

Algorithme de la puissance itérée



matrice $A = \begin{pmatrix} 1,36 & 0,56 \\ 0.14 & 0.94 \end{pmatrix}$ dont les vecteurs propres sont :

- $\begin{pmatrix} 4 \\ 1 \end{pmatrix}$ de valeur propre $1,5$
- $\begin{pmatrix} -1 \\ 1 \end{pmatrix}$ de valeur propre $0,8$

Principe

Soit la matrice $A \in \mathcal{M}_{n,n}(\mathbb{R})$ dont on cherche les valeurs propres,

- supposons que A possède n valeurs propres de modules différents

$$\lambda_1, \lambda_2, \ldots, \lambda_n \quad \text{telles que} \quad |\lambda_1| < |\lambda_2| < \cdots < |\lambda_n|$$

et soient $\overrightarrow{u_1}, \overrightarrow{u_2}, \ldots, \overrightarrow{u_n}$ les vecteurs propres associés

- Alors, si $\overrightarrow{x_0}$ est un vecteur quelconque,

$$\overrightarrow{x_0} = a_1\overrightarrow{u_1} + a_2\overrightarrow{u_2} + \cdots + a_n\overrightarrow{u_n}$$

- Supposons que $a_n \neq 0$, on calcule la suite $\overrightarrow{x_{k+1}} = A\overrightarrow{x_k}$

c'est à dire

$$\begin{aligned} \overrightarrow{x_k} &= A^k\overrightarrow{x_0} \\ \overrightarrow{x_k} &= \lambda_1^k a_1\overrightarrow{u_1} + \lambda_2^k a_2\overrightarrow{u_2} + \cdots + \lambda_n^k a_n\overrightarrow{u_n} \\ \overrightarrow{x_k} &= \lambda_n^k\left(\left(\frac{\lambda_1}{\lambda_n}\right)^k a_1\overrightarrow{u_1} + \cdots + \left(\frac{\lambda_{n-1}}{\lambda_n}\right)^k a_{n-1}\overrightarrow{u_{n-1}} + a_n\overrightarrow{u_n}\right) \end{aligned}$$

Or si $i \neq n$, $\left(\frac{\lambda_i}{\lambda_n}\right)^k \to 0$ donc le terme prépondérant devient $\lambda_n^k a_n\overrightarrow{u_n}$

Principe (suite)

Au bout d'un « certain nombre » d'itérations :

- La plus grande valeur propre est $|\lambda_n| \simeq \dfrac{\|\overrightarrow{x_{k+1}}\|}{\|\overrightarrow{x_k}\|}$
- Le vecteur propre associé est $\overrightarrow{u_n} \simeq \overrightarrow{x_k}$

En pratique :

- Si λ_n est grand, il peut y avoir dépassement de capacité
  $\Rightarrow$*On normalise la suite*

$$\overrightarrow{b_k} = \frac{\overrightarrow{x_k}}{\|\overrightarrow{x_k}\|} \quad \textit{et} \quad \overrightarrow{x_{k+1}} = A\overrightarrow{b_k}$$

- On cherche à obtenir λ_n, pour cela on utilise le produit scalaire :

$$\begin{aligned} {}^t\overrightarrow{b_k}\,\overrightarrow{x_{k+1}} &= {}^t\overrightarrow{b_k}A\overrightarrow{b_k} \\ &\simeq \lambda_n{}^t\overrightarrow{b_k}\,\overrightarrow{b_k} \\ &\simeq \lambda_n \qquad \text{si les } \overrightarrow{b_k} \text{ sont normalisés} \end{aligned}$$

Algorithme

Données : A, $\overrightarrow{x_0}$ et ε
début
$\quad \overrightarrow{x} \leftarrow \overrightarrow{x_0}$
$\quad \lambda_{anc} \leftarrow 1$; $\lambda \leftarrow 0$
$\quad$ **tant que** $|\lambda - \lambda_{anc}| > \varepsilon$ **faire**
$\quad\quad \lambda_{anc} \leftarrow \lambda$
$\quad\quad \overrightarrow{b} \leftarrow \dfrac{\overrightarrow{x}}{\|\overrightarrow{x}\|}$
$\quad\quad \overrightarrow{x} \leftarrow A\,\overrightarrow{b}$
$\quad\quad \lambda \leftarrow {}^t\overrightarrow{b}\,\overrightarrow{x}$
$\quad$ **retourner** λ, b
fin

- On peut ajouter un test de convergence sur le vecteur pour obtenir un vecteur propre.
- Ici, on utilise la norme euclidienne $\|\overrightarrow{x}\| = \sqrt{{}^t\overrightarrow{x}\,\overrightarrow{x}}$ pour utiliser une autre norme, il faut changer le calcul de λ : $\lambda \leftarrow \dfrac{{}^t\overrightarrow{b}\,\overrightarrow{x}}{{}^t\overrightarrow{b}\,\overrightarrow{b}}$

Vitesse de Convergence

- supposons que A possède n valeurs propres de modules différents

$$\lambda_1, \lambda_2, \ldots, \lambda_n \quad \text{telles que} \quad |\lambda_1| < |\lambda_2| < \cdots < |\lambda_n|$$

et soient $\overrightarrow{u_1}, \overrightarrow{u_2}, \ldots, \overrightarrow{u_n}$ les vecteurs propres associés

$$\overrightarrow{x_0} = a_1\overrightarrow{u_1} + a_2\overrightarrow{u_2} + \cdots + a_n\overrightarrow{u_n}$$

En normalisant,

$$\begin{aligned} \overrightarrow{b_k} &= \frac{\lambda_n^k}{\|A^k\overrightarrow{x_0}\|}\left(\frac{\lambda_1^k}{\lambda_n^k}a_1\overrightarrow{u_1} + \cdots + \frac{\lambda_{n-1}^k}{\lambda_n^k}a_{n-1}\overrightarrow{u_{n-1}} + a_n\overrightarrow{u_n}\right) \\ &= \frac{\lambda_n^k}{\|A^k\overrightarrow{x_0}\|}a_n\overrightarrow{u_n} + \frac{\lambda_n^k}{\|A^k\overrightarrow{x_0}\|}\left(\frac{\lambda_1^k}{\lambda_n^k}a_1\overrightarrow{u_1} + \cdots + \frac{\lambda_{n-1}^k}{\lambda_n^k}a_{n-1}\overrightarrow{u_{n-1}}\right) \end{aligned}$$

Le terme d'erreur le plus important est $\left(\frac{\lambda_{n-1}}{\lambda_n}\right)^k a_n\overrightarrow{u_{n-1}}$

Vitesse de convergence

Par définition, $\frac{|\lambda_n|^k}{\|A^k\overrightarrow{x_0}\|}$ est borné car $\overrightarrow{b_k}$ est de norme 1. Donc il existe C tel que :

$$\begin{aligned} \left\|\overrightarrow{b_k} - \frac{\lambda_n^k}{\|A^k\overrightarrow{x_0}\|}a_n\overrightarrow{u_n}\right\| &\leq C\left\|\frac{\lambda_1^k}{\lambda_n^k}a_1\overrightarrow{u_1} + \cdots + \frac{\lambda_{n-1}^k}{\lambda_n^k}a_{n-1}\overrightarrow{u_{n-1}}\right\| \\ &\leq C\frac{|\lambda_{n-1}|^k}{|\lambda_n|^k}\left(\|a_1\overrightarrow{u_1}\| + \|a_2\overrightarrow{u_2}\| + \|a_{n-1}\overrightarrow{u_{n-1}}\|\right) \end{aligned}$$

La convergence dépend donc du rapport entre la plus grande et la deuxième plus grande valeur propre :

$$\frac{|\lambda_{n-1}|}{|\lambda_n|}$$

$\Rightarrow$*Si les valeurs propres sont très proches, la convergence sera lente*

Méthode de déflation

Introduction
Décomposition $\mathbf{QR}$
Introduction
Algorithme de la puissance itérée
Méthode de déflation
Méthode de déflation
Méthode de déflation (suite)
Utilisation de la méthode
Méthode de la puissance inverse
Conditions de convergence
Matrices semblables
Conclusion

Méthode de déflation

Comment faire pour trouver la deuxième plus grande valeur propre ?

Idée : trouver une matrice qui a pour valeurs propres $\lambda_1, \dots, \lambda_{n-1}$ et 0

Soient $\overrightarrow{u_1}, \dots, \overrightarrow{u_n}$ les vecteurs propres de A, ils forment une base de $\mathbb{R}^n$. Il existe alors une *base duale* $\overrightarrow{v_1}, \cdots, \overrightarrow{v_n}$ telle que

$$\forall i,j \qquad {}^t\overrightarrow{v_i}\overrightarrow{u_j} = \delta_{ij}$$

Rappel : $\delta_{ij} = 0$ si $i \neq j$, $\delta_{ii} = 1$

Alors soit

$$B = A - \lambda_n \overrightarrow{u_n}\,{}^t\overrightarrow{v_n}$$

$\forall i \in \{1, 2, \dots, n\}$

$$\begin{aligned} B(\overrightarrow{u_i}) &= A\overrightarrow{u_i} - \lambda_n \overrightarrow{u_n}({}^t\overrightarrow{v_n}\overrightarrow{u_i}) \\ &= \lambda_i \overrightarrow{u_i} - \lambda_n \delta_{in}\overrightarrow{u_n} \\ &= \begin{cases} 0 \text{ si } i = n \\ \lambda_i \overrightarrow{u_i} \text{ si } i \neq n \end{cases} \end{aligned}$$

Donc B a pour valeurs propres $\lambda_1, \dots, \lambda_{n-1}$ et 0 et les mêmes vecteurs propres que A.

Méthode de déflation (suite)

En appliquant la méthode de la puissance itérée à B, on obtient λ_{n-1} et $\overrightarrow{u_{n-1}}$

Connaissant λ_n et $\overrightarrow{u_n}$ et A, comment trouver le vecteur de la base duale $\overrightarrow{v_n}$?

- Si A est symétrique, les vecteur propres sont orthogonaux donc $\overrightarrow{v_n} = \frac{\overrightarrow{u_n}}{{}^t\overrightarrow{u_n}\overrightarrow{u_n}}$.
- Sinon, considérons ${}^tA$,
  - Elle a les mêmes valeurs propres que A.
  - Soit $\overrightarrow{v}$ le vecteur propre associé à λ_n,

$$\begin{aligned} {}^t\overrightarrow{v}A\overrightarrow{u_i} &= ({}^t\overrightarrow{v}A)\,\overrightarrow{u_i} &= \lambda_n {}^t\overrightarrow{v}\,\overrightarrow{u_i} \\ &= {}^t\overrightarrow{v}\,(A\overrightarrow{u_i}) &= \lambda_i {}^t\overrightarrow{v}\,\overrightarrow{u_i} \end{aligned}$$

Donc si $i \neq n$, ${}^t\overrightarrow{v}\,\overrightarrow{u_i} = 0$, la base duale est celle des vecteurs propres de ${}^tA$.

Pour construire la matrice de déflation, il suffit donc de trouver un vecteur propre associé à la plus grande valeur propre de ${}^tA$.

Introduction
Décomposition $\mathbf{QR}$
Introduction
Algorithme de la puissance itérée
Méthode de déflation
Méthode de déflation
Méthode de déflation (suite)
Utilisation de la méthode
Méthode de la puissance inverse
Conditions de convergence
Matrices semblables
Conclusion

Utilisation de la méthode

- La méthode de déflation permet de trouver toutes les valeurs propres de la matrice
  - Sans les connaître
  - En obtenant des vecteurs propres
- Il y a accumulation de l'erreur
  - La méthode de la puissance donne une approximation de la plus grande valeur propre et du vecteur propre.
  - La matrice B utilisée n'est pas exactement la bonne.
  - Il y a dégradation du résultat.

Introduction
Décomposition $\mathbf{QR}$
Introduction
Algorithme de la puissance itérée
Méthode de déflation
Méthode de déflation
Méthode de déflation (suite)
Utilisation de la méthode
Méthode de la puissance inverse
Conditions de convergence
Matrices semblables
Conclusion

Méthode de la puissance inverse

Introduction
Décomposition $\mathbf{QR}$
Introduction
Algorithme de la puissance itérée
Méthode de déflation
Méthode de la puissance inverse
Algorithme de la puissance inverse
Algorithme
Méthode de décalage
Exemple d'utilisation
Méthode de décalage (fin)
Conditions de convergence
Matrices semblables
Conclusion

Algorithme de la puissance inverse

Que faire si on recherche la valeur propre de plus petit module ?

Rq : Les valeurs propres de A^{-1} sont les inverses des valeurs propres de A

Si A est inversible, on applique l'algorithme de la puissance itérée à A^{-1} :

- On effectue la décomposition PLU de A : $PA = LU$
- soit $\overrightarrow{b}_0$ le vecteur de départ,
- On construit les suites $\overrightarrow{b}_k$ et λ_k :
  - On résout le système $LU\overrightarrow{x}_{k+1} = P\overrightarrow{b}_k$
  - On pose :

$$\overrightarrow{b}_k = \frac{\overrightarrow{x}_k}{\|\overrightarrow{x}_k\|} \qquad \text{et} \qquad \lambda_k = \frac{1}{{}^t\overrightarrow{x}_{k+1}\overrightarrow{b}_k}$$

- On s'arrête quand $|\lambda_{k+1} - \lambda_k| < \varepsilon$

La suite λ_k converge vers l'inverse de la plus grande valeur propre de A^{-1} donc vers la valeur propre de A de plus petit module.

Algorithme

Données : A, P, L, U tels que $PA = LU$, $\overrightarrow{x}_0$ et ε

début

- $\overrightarrow{x} \leftarrow \overrightarrow{x}_0$
- $\lambda_{anc} \leftarrow 1$
- $\lambda \leftarrow 0$
- **tant que** $|\lambda - \lambda_{anc}| > \varepsilon$ **faire**
  - $\lambda_{anc} \leftarrow \lambda$
  - $\overrightarrow{b} \leftarrow \frac{\overrightarrow{x}}{\|\overrightarrow{x}\|}$
  - résoudre $LU\overrightarrow{x} = P\overrightarrow{b}$
  - $\lambda \leftarrow \frac{1}{{}^t\overrightarrow{x}\,\overrightarrow{b}}$
- **retourner** λ *et* $\overrightarrow{b}$

fin

Introduction
Décomposition $\mathbf{QR}$
Introduction
Algorithme de la puissance itérée
Méthode de déflation
Méthode de la puissance inverse
Algorithme de la puissance inverse
Algorithme
Méthode de décalage
Exemple d'utilisation
Méthode de décalage (fin)
Conditions de convergence
Matrices semblables
Conclusion

Méthode de décalage

Quelles sont les valeurs propres de la matrice $A - \alpha I$? Si λ_i est valeur propre de A de vecteur propre $\overrightarrow{u}_i$,

$$\begin{aligned} (A - \alpha I)\,\overrightarrow{u}_i &= A\overrightarrow{u}_i - \alpha\overrightarrow{u}_i \\ &= \lambda_i\,\overrightarrow{u}_i - \alpha\overrightarrow{u}_i \\ &= (\lambda_i - \alpha)\,\overrightarrow{u}_i \end{aligned}$$

Que se passe-t-il si on applique à la matrice $A - \alpha I$

- la méthode de la puissance itérée ?
  - On obtient la valeur propre $\lambda_k - \alpha$ de module maximum.
  - Cela permet d'accélérer le calcul d'une valeur propre.
- la méthode de la puissance inverse ?
  - Cela permet d'obtenir la valeur propre $\lambda_k - \alpha$ de module minimum
  - C'est la valeur propre la plus proche de α

  $\Rightarrow$ *C'est un moyen d'améliorer le calcul d'une valeur propre et d'obtenir un vecteur propre correspondant à cette valeur*

Introduction
Décomposition $\mathbf{QR}$
Introduction
Algorithme de la puissance itérée
Méthode de déflation
Méthode de la puissance inverse
Algorithme de la puissance inverse
Algorithme
Méthode de décalage
Exemple d'utilisation
Méthode de décalage (fin)
Conditions de convergence
Matrices semblables
Conclusion

Exemple d'utilisation

Supposons qu'un premier calcul est fourni une approximation d'une valeur propre : α.

Données : $A, \alpha, \overrightarrow{x}_0$ et ε
début
- $\overrightarrow{x} \leftarrow \overrightarrow{x}_0$
- $\lambda_{anc} \leftarrow 0$
- $\lambda \leftarrow \alpha$
- **tant que** $|\lambda - \lambda_{anc}| > \varepsilon$ **faire**
  - $\lambda_{anc} \leftarrow \lambda$
  - $\overrightarrow{b} \leftarrow \frac{\overrightarrow{x}}{\|\overrightarrow{x}\|}$
  - résoudre $(A - \alpha I)\overrightarrow{x} = \overrightarrow{b}$
  - $\lambda \leftarrow \frac{1}{{}^t\overrightarrow{x}\,\overrightarrow{b}}$
- **retourner** λ *et* $\overrightarrow{b}$

fin

Cet algorithme améliore l'approximation et permet de calculer un vecteur propre associé à la valeur propre.

Méthode de décalage (fin)

Cette méthode permet
- de calculer un vecteur propre correspondant à une valeur propre connue approximativement
- d'améliorer la précision de la valeur propre.
- Avantages :
  - Convergence plus rapide
  - Permet de s'approcher indéfiniment d'une valeur et d'un vecteur propre
- Inconvénients :
  - Utilisation d'un système linéaire presque singulier
  - Peut converger vers une autre valeur propre.

Conditions de convergence

Conditions

Plusieurs conditions sont nécessaires pour que la méthode fonctionne :
- A est une matrice de dimension n possédant n valeurs propres de modules différents $|\lambda_1| < |\lambda_2| < \cdots < |\lambda_n|$

  $\Longrightarrow$ *A est diagonalisable*
- Soient $\overrightarrow{u}_1, \overrightarrow{u}_2, \ldots, \overrightarrow{u}_n$ les vecteurs propres associés aux valeurs propres de A. On doit partir d'un vecteur $\overrightarrow{x}_0$

$$\overrightarrow{x}_0 = a_1\overrightarrow{u}_1 + \cdots + a_n\overrightarrow{u}_n$$

  où $\underline{\mathbf{a_n \neq 0}}$

  Que se passe-t-il si $a_n = 0$?

si $a_n = 0$

Si on part du point de départ

$$\overrightarrow{x}_0 = a_1\overrightarrow{u}_1 + \cdots + a_{n-1}\overrightarrow{u}_{n-1} + 0\overrightarrow{u}_n$$

- Lorsqu'on calcule le premier terme

$$\begin{aligned}\overrightarrow{x}_1 &= A\overrightarrow{x}_0\\ &= \lambda_1 a_1\overrightarrow{u}_1 + \cdots + \lambda_{n-1}a_{n-1}\overrightarrow{u}_{n-1} + 0\overrightarrow{u}_n\end{aligned}$$

  à cause des erreurs de calcul, on obtient en fait

$$\widetilde{\overrightarrow{x}_1} = \widetilde{\lambda_1 a_1}\overrightarrow{u}_1 + \cdots + \widetilde{\lambda_{n-1}a_{n-1}}\overrightarrow{u}_{n-1} + \varepsilon\overrightarrow{u}_n$$

- Si λ_n est beaucoup plus grande que les autres valeurs propres, les calculs suivants font ressortir le coefficient associé à $\overrightarrow{u}_n$ alors qu'ils annulent les autres.
- Donc l'algorithme nous donne toujours $\overrightarrow{u}_n$ et λ_n grâce aux erreurs de calcul.

si $a_n = 0$ (suite)

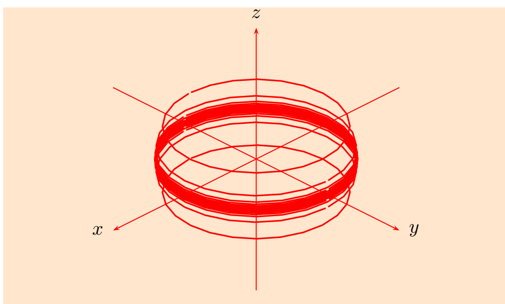
- Si les deux plus grandes valeurs propres λ_n et λ_{n-1} sont proches l'une de l'autre,
- l'algorithme peut s'arrêter en obtenant λ_{n-1} et $\overrightarrow{u}_{n-1}$ au lieu de λ_n et $\overrightarrow{u}_n$.

Si on généralise le phénomène en utilisant la méthode de déflation, cela signifie que les valeurs propres qui sont proches les unes des autres ne sont pas forcément obtenues dans le bon ordre.

Autres conditions de convergences

Jusqu'à présent on a considéré que les valeurs propres de la matrice A ont des modules différents. Mais que se passe-t-il si :
- A possède une valeur propre dégénérée
  - Un espace propre de dimension > 1
- A possède des valeurs propres différentes mais de module identique

Sous espace propre de dimension > 1



Matrice $A = \begin{pmatrix} 3 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 1 \end{pmatrix}$ on trace $\frac{Ax}{\|Ax\|}$

Sous espace propre de dimension > 1

La suite $\vec{x}_k$ tend vers *un* vecteur du sous espace propre

- Soit λ la valeur propre de plus grand module et d la dimension de son sous espace propre.
- Soit $\lambda_1, \lambda_2, \dots, \lambda_{n-d}$ les autres valeurs propres et $\vec{u}_1, \vec{u}_2, \dots, \vec{u}_{n-d}$ les vecteurs propres associés
- En reprenant le raisonnement précédant

$$\vec{x}_0 = a_1\vec{u}_1 + \dots + a_{n-d}\vec{u}_{n-d} + \vec{u}$$

où $\vec{u}$ est un vecteur propre de valeur propre λ

- On calcule la suite $\vec{x}_{k+1} = A\vec{x}_k$

  c'est à dire

$$\begin{aligned}\vec{x}_k &= A^k\vec{u}_0\\ \vec{x}_k &= \lambda_1^k a_1\vec{u}_1 + \dots + \lambda_{n-d}^k a_{n-d}\vec{u}_{n-d} + \lambda^k\vec{u}\\ \vec{x}_k &= \lambda_n^k\left(\frac{\lambda_1^k}{\lambda_n^k}a_1\vec{u}_1 + \dots + \frac{\lambda_{n-d}^k}{\lambda_n^k}a_{n-d}\vec{u}_{n-d} + \vec{u}\right)\end{aligned}$$

Le terme prépondérant devient $\lambda_n^k\vec{u}$

Donc la suite normalisée $\frac{\vec{x}_k}{\|\vec{x}_n\|}$ tend vers *un* vecteur du sous espace propre

- Selon la valeur de $\vec{x}_0$ la suite $\frac{\vec{x}_k}{\|\vec{x}_k\|}$ tend vers différents vecteurs du sous espace vectoriel.
- Ce cas ne peut être détecté que si on calcule plusieurs fois la suite à partir de points de départ différents

Introduction
Décomposition QR
Introduction
Algorithme de la puissance itérée
Méthode de déflation
Méthode de la puissance inverse
Conditions de convergence
Conditions
si $a_n = 0$
si $a_n = 0$ (suite)
Autres conditions de convergences
Sous espace propre de dimension > 1
Sous espace propre de dimension > 1
Valeurs de même module
Résumé
Matrices semblables
Conclusion

Valeurs de même module

La situation est-elle la même que pour les valeurs propres réelles d'ordre > 1 ?

Si A possède des valeurs propres différentes mais de module identique

Soit $\lambda_n, \dots \lambda_{n-k} \in \mathbb{C}^{k+1}$ ces valeurs propres. Et soit $E_{\lambda_n}, \dots, E_{\lambda_{n-1}}$ les sous-espaces propres associés à ces valeurs.

- La suite $\frac{\vec{x}_k}{\|\vec{x}_k\|}$ tend vers le sous-espace engendré par les sous-espaces propres

$$E = E_{\lambda_n} \oplus E_{\lambda_{n-1}} \oplus \dots \oplus E_{\lambda_{n-k}}$$

- Mais contrairement au cas précédent, E n'est pas un espace propre.
- Il n'y a pas de convergence

Dans le cas de deux valeurs propres seulement, il est possible de modifier l'algorithme pour assurer la convergence en faisant 2 itérations successives.

Introduction
Décomposition QR
Introduction
Algorithme de la puissance itérée
Méthode de déflation
Méthode de la puissance inverse
Conditions de convergence
Conditions
si $a_n = 0$
si $a_n = 0$ (suite)
Autres conditions de convergences
Sous espace propre de dimension > 1
Sous espace propre de dimension > 1
Valeurs de même module
Résumé
Matrices semblables
Conclusion

Résumé

- L'algorithme de la puissance itérée fonctionne si la matrice A
  - a des valeurs propres toutes différentes
  - est diagonalisable
  - est symétrique
    - ses valeurs propres sont alors réelles
    - le seul cas gênant est celui des valeurs propres de signes différents
- Il ne fonctionne pas s'il existe plusieurs valeurs propres complexes de même module Par exemple

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

- L'algorithme converge d'autant mieux si les valeurs propres sont de modules très différents.
- Si ce n'est pas le cas, on peut les obtenir dans le désordre.

Introduction
Décomposition QR
Introduction
Algorithme de la puissance itérée
Méthode de déflation
Méthode de la puissance inverse
Conditions de convergence
Conditions
si $a_n = 0$
si $a_n = 0$ (suite)
Autres conditions de convergences
Sous espace propre de dimension > 1
Sous espace propre de dimension > 1
Valeurs de même module
Résumé
Matrices semblables
Conclusion

Matrices semblables

Valeurs propres de matrices semblables

Définition *On dit que deux matrices A et $B \in \mathcal{M}_{nn}(R)$ sont semblables si $\exists P \in \mathcal{M}_{nn}(R)$ une matrice inversible telle que :*

$$B = P^{-1}AP$$

Théorème *Deux matrices semblables ont les mêmes valeurs propres*

En effet,

$$\begin{aligned}\text{Si}\quad A\vec{v} &= \lambda\vec{v}\\ BP^{-1}\vec{v} &= P^{-1}APP^{-1}\vec{v}\\ &= P^{-1}A\vec{v}\\ &= \lambda P^{-1}\vec{v}\end{aligned}$$

$\Rightarrow$*Si on construit une suite de matrices semblables qui tend vers une matrice triangulaire on obtient toutes les valeurs propres*

Introduction
Décomposition QR
Introduction
Algorithme de la puissance itérée
Méthode de déflation
Méthode de la puissance inverse
Conditions de convergence
Matrices semblables
Valeurs propres de matrices semblables
Algorithme QR
Algorithme QR
Convergence
Résumé
Conclusion

Algorithme QR

Pour construire cette suite on utilise la décomposition QR.

- Si $A = QR$ avec Q matrice orthogonale et R matrice triangulaire, alors

$$\begin{aligned}A' &= RQ \text{ est semblable à } A \text{ car}\\ {}^tQAQ &= {}^tQQRQ\\ &= RQ\\ &= A'\end{aligned}$$

- On peut donc construire la suite :

$$\begin{cases} A_0 = A\\ Q_k, R_k \text{ tel que } A_k = Q_kR_k\\ A_{k+1} = R_kQ_k\end{cases}$$

- Cette suite tend vers une matrice triangulaire ayant les mêmes valeurs propres que A

Introduction
Décomposition QR
Introduction
Algorithme de la puissance itérée
Méthode de déflation
Méthode de la puissance inverse
Conditions de convergence
Matrices semblables
Valeurs propres de matrices semblables
Algorithme QR
Algorithme QR
Convergence
Résumé
Conclusion

Algorithme QR

```
Données : A, ε
début
    A_n ← A
    max ← max_{i<j}(A_n(i,j))
    // tant que la matrice A_n n'est pas triangulaire
    tant que max > ε faire
        //faire la décomposition Q,R de A_n
        [Q,R] ← qr(A_n)
        A_n ← R× Q
        max ← max_{i<j}(A_n(i,j))
fin
Résultat : (A_n(i,i))_{i=1...n}
```

Introduction
Décomposition QR
Introduction
Algorithme de la puissance itérée
Méthode de déflation
Méthode de la puissance inverse
Conditions de convergence
Matrices semblables
Valeurs propres de matrices semblables
Algorithme QR
Algorithme QR
Convergence
Résumé
Conclusion

Convergence

Théorème *Si A est une matrice inversible de valeurs propres réelles différentes, la suite de matrice A_k converge vers une matrice triangulaire supérieure dont la diagonale est constituée des valeurs propres de A*

Idée de la démonstration

- Si A_∞ est triangulaire, ses valeurs propres sont sur sa diagonale (preuve par le polynôme caractéristique).
- La suite A_k est une suite de matrices semblables à A donc A_k a les mêmes valeurs propres que A et par continuité, A_∞ aussi.
- On peut montrer par récurrence que

$$\begin{aligned} A_{k+1} &= ({}^tQ_k \dots {}^tQ_1) A (Q_1 \dots Q_k) \\ \text{et} \quad A^k &= (Q_1 \dots Q_k)(R_k \dots R_1) \end{aligned}$$

Cette méthode revient à appliquer l'algorithme de la puissance simultanément sur l'espace $\mathbb{R}^n$, $E_{\lambda_n}^\perp$, $\left(E_{\lambda_n} \oplus E_{\lambda_{n-1}}\right)^\perp \dots$

Introduction
Décomposition $\mathbf{QR}$
Introduction
Algorithme de la puissance itérée
Méthode de déflation
Méthode de la puissance inverse
Conditions de convergence
Matrices semblables
Valeurs propres de matrices semblables
Algorithme QR
Algorithme QR
Convergence
Résumé
Conclusion

Résumé

- La convergence de la méthode QR est basée sur celle de la méthode de la puissance
  - Cela fonctionne si la matrice de départ A a des valeurs propres de modules différents $|\lambda_1| < |\lambda_2| < \dots < |\lambda_n|$.
  - La vitesse de convergence dépend du rapport entre deux valeurs propres successives $\frac{|\lambda_k|}{|\lambda_{k+1}|}$
  - Si deux valeurs propres ont des modules proches, il est possible de les obtenir dans le désordre.
- Elle ne calcule pas directement les vecteurs propres
- On peut la modifier pour calculer seulement les m premières valeurs propres $\Rightarrow$ *méthode d'itération du sous-espace*.
- La convergence est lente mais peut être accélérée, en réduisant A en une matrice plus simple
  - pour les matrices quelconques, forme de HESSENBERG (quasi triangulaire)
  - pour les matrices symétriques, forme tri-diagonale.

Introduction
Décomposition $\mathbf{QR}$
Introduction
Algorithme de la puissance itérée
Méthode de déflation
Méthode de la puissance inverse
Conditions de convergence
Matrices semblables
Valeurs propres de matrices semblables
Algorithme QR
Algorithme QR
Convergence
Résumé
Conclusion

Conclusion

- **Puissance itérée**
  - Avantages :
    - Il n'utilise que des produits de matrices par des vecteurs
    - La convergence peut être accélérée
  - Inconvénients :
    - La convergence peut être lente
    - On obtient une approximation ce qui donne de mauvais résultats avec la méthode de déflation
- **Méthode QR**
  - Avantage :
    - calcul de toutes les valeurs propres.
  - Inconvénients :
    - convergence lente,
    - chaque itération demande une décomposition QR.

Introduction
Décomposition $\mathbf{QR}$
Introduction
Algorithme de la puissance itérée
Méthode de déflation
Méthode de la puissance inverse
Conditions de convergence
Matrices semblables
Conclusion
Conclusion