

# Approximation des valeurs propres et des vecteurs propres

**Réalisé par :** Ismail ElFayk - Ikram Chebbac  
**Encadré par :** M. Elmassoudi M'hamed

Master Mathématiques Appliquées et Systèmes Intelligents  
Faculté des Sciences Dhar El Mehratz  
Université Sidi Mohammed Ben Abdellah

12 mai 2025



**FACULTÉ DES SCIENCES DHAR EL MAHRAZ**  
UNIVERSITÉ SIDI MOHAMMED BEN ABDELLAH

# Table des matières

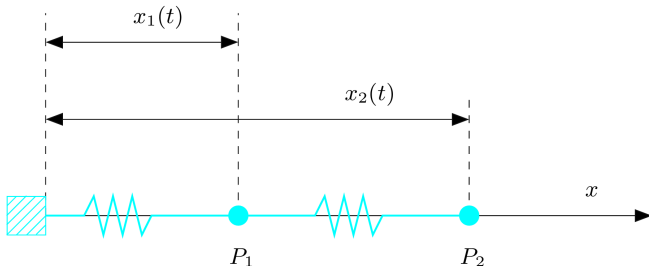
- 1 Introduction
- 2 Méthodes de type puissance itérée
- 3 Méthode de Jacobi
- 4 La méthode de Givens-Householder (bissection)
- 5 Méthode QR
- 6 Méthode SVD (Décompositon en Valeurs Singulières)
- 7 Méthode de Lanczos
- 8 Conlusion

# Sommaire



## Motivation : Problème (Ressorts élastiques)

Considérons le système de la figure suivante constitué de deux corps ponctuels  $P_1$  et  $P_2$  de masse  $m$ , reliés par deux ressorts et libres de se déplacer le long d'une ligne joignant  $P_1$  et  $P_2$ . Soit  $x_i(t)$  la position de  $P_i$  au temps  $t$ , pour  $i = 1, 2$ .



Le principe fondamentale de la dynamique donne

$$m\ddot{x}_1 = K(x_2 - x_1) - Kx_1, \quad m\ddot{x}_2 = K(x_1 - x_2)$$

où  $K$  est le coefficient de raideur des deux ressorts.



# Sommaire

## 1 Introduction

## 2 Méthodes de type puissance itérée

- Puissance itérée
- Puissance inverse
- Méthode de la puissance inverse avec translation
- Méthode de la déflation

### 3 Méthode de Jacobi

#### 4 La méthode de Givens-Householder (bissection)

## 5 Méthode QR

## 7 Método de Lagrange

# Sommaire

## 1 Introduction

## 2 Méthodes de type puissance itérée

- Puissance itérée
- Puissance inverse
- Méthode de la puissance inverse avec translation
- Méthode de la déflation

### 3 Méthode de Jacobi

#### 4 La méthode de Givens-Householder (bissection)

## 5 Méthode QR

### Método de Lagrange





# Puissance itérée

## Théorème

Soit  $A$  une matrice carrée d'ordre  $n$ , diagonalisable et de valeurs propres  $\lambda_i, 1 \leq i \leq n$ , comptées avec leur ordre de multiplicité, et vérifiant :

$$0 \leq |\lambda_1| \leq \dots \leq |\lambda_{n-1}| < |\lambda_n|$$

Soit  $(u_i)_{1 \leq i \leq n}$  une base normée de vecteurs propres de  $A$  associés aux valeurs propres  $\lambda_i$ . Si  $x^{(0)} = \sum_{i=1}^n \alpha_i u_i$  est tel que  $\alpha_n \neq 0$ , alors la méthode de la puissance converge au sens suivant :

$$\lim_{k \rightarrow +\infty} \lambda^{(k)} = \lambda_n \quad (3)$$

$$\lim_{k \rightarrow +\infty} (x^{(k)} - \beta^k \gamma u_n) = 0 \quad (4)$$

où  $\beta = \frac{\lambda_n}{|\lambda_n|}$  et  $\gamma$  est un nombre complexe de module 1.



# Puissance itérée

Or, pour tout  $i \in \{1, 2, \dots, n-1\}$ , on a  $|\lambda_i| < |\lambda_n|$ . Et par suite, on obtient

$$\lim_{k \rightarrow +\infty} \frac{z^{(k)}}{\lambda_n^k} = \alpha_n u_n \quad (8)$$

et donc

$$\lim_{k \rightarrow +\infty} \frac{\|z^{(k)}\|_2}{|\lambda_n^k|} = |\alpha_n| \|u_n\|_2 = |\alpha_n| \quad (9)$$

puisque  $u_n$  est de norme égale à 1 . Utilisant l'hypothèse  $\alpha_n \neq 0$ , on obtient, d'après (8) et (9) :

$$\lim_{k \rightarrow +\infty} \frac{z^{(k)}}{\|z^{(k)}\|_2} \frac{|\lambda_n^k|}{\lambda_n^k} = \frac{\alpha_n}{|\alpha_n|} u_n \quad (10)$$







# Puissance itérée

**Exemple.**

Soit la matrice  $A = \begin{bmatrix} 2 & -12 \\ 1 & -5 \end{bmatrix}$ , avec le vecteur propre  $\mathbf{v} = \begin{bmatrix} 3 \\ 1 \end{bmatrix}$  et la valeur propre  $\lambda = -2$ .









## Puissance inverse

C'est la méthode de la puissance appliquée à la matrice  $A^{-1}$ , qui permet d'obtenir la valeur propre de plus petit module de  $A$  et un vecteur propre qui lui est associé.

## Puissance inverse

### Recherche de la valeur propre de plus petit module

La valeur propre de plus grand module de  $A^{-1}$  est égale à l'inverse de la valeur propre de  $A$  de plus petit module, puisque

$$\max_i \frac{1}{|\lambda_i|} = \frac{1}{\min_i |\lambda_i|}$$

On applique alors la méthode de la puissance à  $A^{-1}$ , puis on inverse la valeur propre trouvée, pour obtenir la valeur propre de  $A$  de plus petit module ainsi qu'un vecteur propre associé. Cela conduit, en partant de l'algorithme de puissance itérée, à l'algorithme suivant :

$$\left\{ \begin{array}{l} x^{(0)} \quad \text{donné dans } \mathbf{C}^n \text{ de norme } 1 \\ \text{Pour } k = 0, 1, \dots \text{ calculer} \\ y^{(k+1)} = A^{-1}x^{(k)} \\ x^{(k+1)} = \frac{y^{(k+1)}}{\|y^{(k+1)}\|_2} \\ \gamma^{(k+1)} = \frac{1}{(x^{(k)})^* A^{-1} x^{(k)}} = \frac{1}{(x^{(k)})^* y^{(k+1)}} \end{array} \right.$$

# Puissance inverse

Dans la pratique, on ne calcule pas  $A^{-1}$ , mais on résout successivement les systèmes linéaires  $Ay^{(k+1)} = x^{(k)}$ , par exemple en ayant factorisé  $A$  une fois pour toute sous la forme du produit de deux matrices triangulaires :  $A = LU$  (ou  $PA = LU$  lorsque cela est nécessaire).

## Algorithme de la méthode de la puissance inverse

$$\left\{ \begin{array}{l} x^{(0)} \quad \text{donné dans } \mathbf{C}^n \text{ de norme } 1 \\ \text{Pour } k = 0, 1, \dots \text{ calculer} \\ \quad y^{(k+1)} \text{ en résolvant le système } Ay^{(k+1)} = x^{(k)} \\ \quad x^{(k+1)} = \frac{y^{(k+1)}}{\|y^{(k+1)}\|_2} \\ \quad \gamma^{(k+1)} = \frac{1}{(x^{(k)})^* y^{(k+1)}} \end{array} \right. \quad (17)$$

On a le résultat de convergence suivant.



# Sommaire

## 1 Introduction

## 2 Méthodes de type puissance itérée

- Puissance itérée
- Puissance inverse
- **Méthode de la puissance inverse avec translation**
- Méthode de la déflation

### 3 Méthode de Jacobi

#### 4 La méthode de Givens-Householder (bissection)

## 5 Méthode QR

### Método de Lagrange



## Puissance inverse avec translation

Elle permet d'obtenir la valeur propre la plus proche d'un nombre donné. Soit  $\mu$  un nombre donné. Considérons la matrice  $(A - \mu I)$  obtenue à partir de  $A$  par la "translation  $-\mu I$ " (shift en anglais). On applique l'algorithme de la méthode de la puissance inverse à la matrice  $(A - \mu I)$ . Le plus grand module des valeurs propres de  $(A - \mu I)^{-1}$  est  $\frac{1}{\min_i |\lambda_i - \mu|}$ , où les  $\lambda_i$  sont les valeurs propres de  $A$ .

$$\left\{ \begin{array}{l} x^{(0)} \quad \text{donné dans } \mathbf{C}^n \text{ de norme 1} \\ \text{Pour } k = 0, 1, \dots \text{ calculer} \\ y^{(k+1)} \text{ en résolvant le système } (A - \mu I)y^{(k+1)} = x^{(k)} \\ x^{(k+1)} = \frac{y^{(k+1)}}{\|y^{(k+1)}\|_2} \\ \tilde{\gamma}^{(k+1)} = \mu + \frac{1}{(x^{(k)})^* (A - \mu I)^{-1} x^{(k)}} = \mu + \frac{1}{(x^{(k)})^* y^{(k+1)}} \end{array} \right.$$

# Puissance inverse avec translation

## Corollaire

Soit  $A$  une matrice diagonalisable de valeurs propres  $\lambda_i, 1 \leq i \leq n$ . Soit  $(u_i)_{1 \leq i \leq n}$  une base normée de vecteurs propres de  $A$  associés aux valeurs propres  $\lambda_i$ . Si  $\lambda_{i_0}$  est la valeur propre de  $A$  la plus proche de  $\mu$  avec de plus

$$|\lambda_{i_0} - \mu| < |\lambda_i - \mu|, \quad \text{pour } \lambda_i \in Sp(A) \setminus \{\lambda_{i_0}\}$$

et si  $x^{(0)} = \alpha_{i_0} u_{i_0} + \sum_{\substack{i=1 \\ i \neq i_0}}^n \alpha_i u_i$  est tel que  $\alpha_{i_0} \neq 0$ , alors la méthode de la puissance inverse avec translation converge :

$$\lim_{k \rightarrow +\infty} \tilde{\gamma}^{(k)} = \lambda_{i_0}$$

**Démonstration.** Il suffit d'appliquer le théorème du puissance itérée à la matrice  $(A - \mu I)^{-1}$ .

# Sommaire

## 1 Introduction

## 2 Méthodes de type puissance itérée

- Puissance itérée
- Puissance inverse
- Méthode de la puissance inverse avec translation
- Méthode de la déflation

### 3 Méthode de Jacobi

#### 4 La méthode de Givens-Householder (bissection)

## 5 Méthode QR

### ➤ Méthode de Lagrange



# La méthode de déflation

On déduit donc qu'une façon d'obtenir la *deuxième plus grande* valeur propre en module consiste simplement à appliquer la puissance itérée en partant d'un vecteur

$$x_0 = x - \langle x, u_n \rangle u_n$$

i.e. un vecteur orthogonale à l'espace propre  $E(u_n)$ .

**Remarque :** C'est équivalent à appliquer la puissance itérée à la matrice

$$A_2 = A - \lambda_n u_n u_n^t$$

dont il est facile de vérifier que le spectre est

$$sp(A_2) = \{0, \lambda_1, \dots, \lambda_{n-1}\}$$









# Sommaire

- ## 8 Conclusion

# Méthode de Jacobi [3] [7]

Cette méthode s'applique aux matrices *symétriques réelles*. Elle permet d'obtenir une approximation de toutes les valeurs propres et tous les vecteurs propres.

La matrice  $A$  étant symétrique, alors elle est diagonalisable. Et par suite, il existe une matrice orthogonale  $O$  telle que  $O^T A O$  soit diagonale :

$$O^T A O = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n),$$

où les nombres  $\lambda_i, 1 \leq i \leq n$ , sont les valeurs propres de la matrice  $A$ , comptées avec leur ordre de multiplicité. Rappelons que les vecteurs colonnes de la matrice  $O$  forment un ensemble orthonormal de vecteurs propres, le  $i$ -ème vecteur colonne étant un vecteur propre associé à la valeur propre  $\lambda_i$ .





# Matrice $\Omega$

Les matrices  $Q_k$  qu'on utilise sont des matrices orthogonales (que nous appelons les rotations de Jacobi) de la forme :

$$\Omega = \begin{pmatrix} 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & & \vdots & & \vdots \\ 0 & \cdots & \cos \theta & \cdots & \sin \theta & \cdots & 0 \\ \vdots & & \vdots & \ddots & \vdots & & \vdots \\ 0 & \cdots & -\sin \theta & \cdots & \cos \theta & \cdots & 0 \\ \vdots & & \vdots & & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & 1 \end{pmatrix} \quad (18)$$

$\Omega_{pp} = \Omega_{qq} = \cos \theta$ ,  $\Omega_{pq} = -\Omega_{qp} = \sin \theta$ ,  $\Omega_{ii} = 1$ ,  
*si  $i \neq p$  et  $i \neq q$ ,  $\Omega_{ij} = 0$ , si  $i \neq j$ ,  $(i, j) \neq (p, q)$  et  $(i, j) \neq (q, p)$ .*

## Méthode de Jacobi

## Théorème

- 1) Si la matrice  $A = (a_{ij})_{1 \leq i, j \leq n}$  est symétrique et si  $B = \Omega^T A \Omega$ , alors  $B = (b_{ij})_{1 \leq i, j \leq n}$  est symétrique; et on a

$$\sum_{i,j} b_{ij}^2 = \sum_{i,j} a_{ij}^2$$

- 2) Si  $a_{pq} \neq 0$ , alors il existe un unique  $\theta \in ]-\frac{\pi}{4}, 0[ \cup ]0, \frac{\pi}{4}[$  tel que  $b_{pq} = 0$ , et  $\theta$  est déterminé par :

$$\cot 2\theta = \frac{a_{qq} - a_{pp}}{2a_{pq}}, -\frac{\pi}{4} \leq \theta \leq \frac{\pi}{4}$$

De plus, on a dans ce cas :

$$\sum_i b_{ii}^2 = \sum_i a_{ii}^2 + 2a_{pq}^2$$

## Méthode de Jacobi

**Remarque.**

- i) La première égalité signifie que la norme de Frobenius définie par :

$$\|A\|_F^2 = \sum_{i,j} a_{ij}^2$$

est conservée.

- ii) Si  $a_{pq} \neq 0$ , la diagonale de  $B$  est globalement plus dominante que celle de  $A$ . Notons que par différence

$$\sum_{i \neq j} b_{ij}^2 = \sum_{i \neq j} a_{ij}^2 - 2a_{pq}^2.$$

En d'autres termes, la transformation réduit les éléments hors-diagonaux et renforce la diagonale.

La répétition de ce type d'opérations devrait "creuser" de plus en plus la partie hors diagonale des matrices successives.



## Méthode de Jacobi

Examinons  $B = \Omega^T A \Omega$ . On a

$$b_{ij} = \sum_{\alpha=1}^N \omega_{\alpha i} \sum_{k=1}^N a_{\alpha k} \omega_{kj} = \sum_{k=1}^N \sum_{\alpha=1}^N \omega_{\alpha i} \omega_{kj} a_{\alpha k}$$

Ainsi

$$\left\{ \begin{array}{l} \text{si } i \neq p, q \text{ et } j \neq p, q \quad b_{ij} = a_{ij} \\ \text{si } i = p \text{ et } j \neq p, q \quad b_{pj} = \cos \theta a_{pj} - \sin \theta a_{qj} \\ \text{si } i = q \text{ et } j \neq p \quad b_{qj} = \sin \theta a_{pj} + \cos \theta a_{qj} \\ \text{pour } i = j = p \quad b_{pp} = \cos^2 \theta a_{pp} + \sin^2 \theta a_{qq} - \sin 2\theta a_{pq} \\ \text{pour } i = j = q \quad b_{qq} = \sin^2 \theta a_{pp} + \cos^2 \theta a_{qq} + \sin 2\theta a_{pq} \\ \text{pour } i = p, j = q \quad b_{pq} = \cos 2\theta a_{pq} + \frac{\sin 2\theta}{2} (a_{pp} - a_{qq}) \\ \text{le reste est obtenu par symétrie.} \end{array} \right.$$

# Méthode de Jacobi

Si  $a_{pq} \neq 0$ , on choisit  $\theta$  pour que  $b_{pq} = 0$  : en particulier tel que

$$\cot 2\theta = \frac{a_{qq} - a_{pp}}{2a_{pq}}, \quad -\frac{\pi}{4} \leq \theta \leq \frac{\pi}{4}.$$

Pour ce choix de  $\theta$ , pour calculer les autres coefficients  $b_{ij}$  de  $B$ , il n'est en fait pas nécessaire de calculer  $\theta$  lui-même ; on a seulement besoin de calculer  $\cos \theta$  et  $\sin \theta$ .

En effet, on remarque que

$$\begin{bmatrix} b_{pp} & b_{pq} \\ b_{qp} & b_{qq} \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} a_{pp} & a_{pq} \\ a_{qp} & a_{qq} \end{bmatrix} \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \quad (19)$$

Posons pour simplifier  $c = \cos \theta$  et  $s = \sin \theta$ .



# Algorithme de la méthode de Jacobi

Partant de  $A_0 = A$ , une itération de la méthode de Jacobi consiste à :

- Déterminer  $(p_k, q_k)$  vérifiant  $|(A_k)_{p_k q_k}| = \max_{1 \leq i < j \leq n} |(A_k)_{ij}|$  ;
- Déterminer la matrice orthogonale  $Q_{k+1}$  définie par

$$Q_{k+1} = \Omega,$$

où  $\Omega$  est donnée par (18) avec  $\theta = \theta_k$ ,  $p = p_k$  et  $q = q_k$ , et où  $\theta_k$  est donnée par

$$\cot 2\theta_k = \frac{(A_k)_{q_k q_k} - (A_k)_{p_k p_k}}{2(A_k)_{p_k q_k}};$$

- Calculer  $A_{k+1} = Q_{k+1}^T A_k Q_{k+1}$  (Le coefficient  $(A_{k+1})_{p_k q_k} = 0$ .)

# Résultats de convergence de la méthode de Jacobi

Posons

$$A_k = D_k + B_k$$

où  $D_k$  est la matrice diagonale formée par les coefficients diagonaux de  $A_k$ , et  $(B_k)_{ij} = (A_k)_{ij}$  si  $i \neq j$ , et  $(B_k)_{ii} = 0$ .

## Théorème

La matrice  $B_k$  donnée par (5.45), vérifie

$$\lim_{k \rightarrow \infty} B_k = 0, \quad \text{dans } \mathcal{M}_n(\mathbb{R})$$

## Théorème

Soit  $\mathcal{S}_n$  l'ensemble des permutations de  $1, 2, \dots, n$ . Alors, il existe  $\sigma \in \mathcal{S}_n$  telle que

$$\lim_{k \rightarrow \infty} D_k = \text{diag}(\lambda_{\sigma(1)}, \lambda_{\sigma(2)}, \dots, \lambda_{\sigma(n)}).$$



# Sommaire

- 1 Introduction
- 2 Méthodes de type puissance itérée
- 3 Méthode de Jacobi
- 4 La méthode de Givens-Householder (bissection)
  - Méthode de Householder
  - Méthode de Givens
- 5 Méthode QR
- 6 Méthode SVD (Décompositon en Valeurs Singulières)
- 7 Méthode de Lanczos

# Introduction [1]

C'est une méthode particulièrement bien adaptée à la recherche de valeurs propres sélectionnées d'une matrice **symétrique**, par exemple toutes les valeurs propres situées dans un intervalle déterminé à l'avance, ou bien les valeurs propres d'un rang donné (en les supposant ordonnées par ordre croissant), etc. Par contre, cette méthode ne permet pas le calcul des vecteurs propres.

La méthode de Givens-Householder (ou bissection) comprend deux étapes :

## - 1ère étape : tridiagonalisation

Étant donnée une matrice symétrique  $A$ , on détermine une matrice orthogonale  $P$  (d'un type particulier) telle que la matrice (symétrique)  $P^T A P$  soit tridiagonale. Cette étape, qui ne nécessite qu'un nombre fini d'opérations élémentaires, constitue la méthode de Householder de réduction d'une matrice symétrique à la forme tridiagonale.

## – 2ème étape : bissection

On est ainsi ramené au calcul des valeurs propres d'une matrice symétrique tridiagonale, qui s'effectue par la méthode de Givens, appelée encore méthode de la bissection.



# Sommaire

- 1 Introduction
- 2 Méthodes de type puissance itérée
- 3 Méthode de Jacobi
- 4 La méthode de Givens-Householder (bissection)
  - Méthode de Householder
  - Méthode de Givens
- 5 Méthode QR
- 6 Méthode SVD (Décompositon en Valeurs Singulières)
- 7 Méthode de Lanczos





# Factorisation de Householder

Or,  $v^*v = \|v\|^2$ , donc :

$$= I_n - 4 \frac{vv^*}{\|v\|^2} + 4 \frac{vv^*}{\|v\|^2} = I_n.$$

On remarque que  $H(v)v^\perp = v^\perp$ . Ainsi, 1 est valeur propre de  $H(v)$  de multiplicité  $n - 1$ . De plus,  $H(v)v = -v$ . Ainsi,  $-1$  est valeur propre de  $H(v)$  de multiplicité 1. Le déterminant d'une matrice étant égal au produit de ses valeurs propres, on obtient bien  $\det(H(v)) = -1$ .

1000

$$H(w_{\pm})v = \mp \|v\| \exp(i\alpha) e_1.$$

[illegible]









# Méthode de Householder

On a alors :

$$\begin{aligned}
 H_1^T A H_1 &= \left( \begin{array}{c|c} 1 & O \\ \hline O & \tilde{H}_1^T = \tilde{H}_1 \end{array} \right) \left( \begin{array}{c|c} a_{11} & a_1^t \\ \hline a_1 & \tilde{A}_1 \end{array} \right) \left( \begin{array}{c|c} 1 & O \\ \hline O & \tilde{H}_1 \end{array} \right) \\
 &= \left( \begin{array}{c|c} a_{11} & a_1^t \\ \hline \tilde{H}_1 a_1 & \tilde{H}_1 \tilde{A}_1 \end{array} \right) \left( \begin{array}{c|c} 1 & O \\ \hline O & \tilde{H}_1 \end{array} \right) = \left( \begin{array}{c|c} a_{11} & a_1^t \tilde{H}_1 = (\alpha e^{(1)})^t \\ \hline \alpha & \\ 0 & \\ \vdots & \tilde{H}_1 \tilde{A}_1 \tilde{H}_1 \\ 0 & \end{array} \right)
 \end{aligned}$$

\_\_\_\_\_

[illegible]





## Méthode de Householder

Si le vecteur  $a_k$  est nul, on choisit  $\tilde{H}_k = I$ , matrice identité de  $\mathcal{M}_{n-k}(\mathbb{R})$ . Utilisant les propriétés de la multiplication par blocs des matrices, on obtient :

$$A_{k+1} = H_k^T A_k H_k = \left( \begin{array}{ccc|c|c} \times & \times & & \circ & \\ \times & \cdot & \cdot & & \circ \\ & \cdot & \cdot & \times & \\ \circ & & \times & \times & \boxed{\left(\tilde{H}_k a_k\right)^t} \\ \hline & & & & \tilde{H}_k \tilde{A}_k \tilde{H}_k \\ \circ & & \tilde{H}_k a_k & & \end{array} \right)$$

## Méthode de Householder

Plus précisément, on a

$$A_{k+1} = \left( \begin{array}{ccc|ccc} \times & \times & \bigcirc & & & \\ \times & \cdot & \cdot & & \bigcirc & \\ & \cdot & \cdot & \times & & \\ \bigcirc & & \times & \times & \alpha & 0 & \cdot & 0 \\ \hline & & & \alpha & & & & \\ & \bigcirc & & 0 & \tilde{H}_k \tilde{A}_k \tilde{H}_k & & & \\ & & & \cdot & & & & \\ & & & 0 & & & & \end{array} \right)$$

avec  $a_{ij}^{k+1} = a_{ij}^k$  pour  $1 \leq i, j \leq k$ .

## Méthode de Householder

À la fin de la  $(n - 2)$ -ème étape, on obtient une matrice  $A_{n-1}$  tridiagonale et semblable à la matrice  $A$  :

$$A_{n-1} = (H_1 H_2 \cdots H_{n-2})^T A (H_1 H_2 \cdots H_{n-2})$$

Récapitulons ce que l'on vient de faire :

## Théorème

Étant donnée une matrice symétrique  $A$  d'ordre  $n$ , il existe une matrice orthogonale  $P$ , produit de  $(n - 2)$  matrices de Householder, telle que la matrice  $P^T A P$  soit tridiagonale et semblable à  $A$ .





## Méthode de Givens

Passons ensuite à la description de la *méthode de Givens* de recherche des valeurs propres d'une matrice symétrique tridiagonale :

$$B = \begin{pmatrix} b_1 & c_1 & & & \\ c_1 & b_2 & c_2 & & \\ & c_2 & \ddots & \ddots & \\ & & \ddots & b_{n-1} & c_{n-1} \\ & & & c_{n-1} & b_n \end{pmatrix}.$$

On observe tout d'abord que si l'un des éléments  $c_i$  est nul, la matrice  $B$  est déjà décomposée en deux sous-matrices diagonales du même type. On peut donc supposer, sans restreindre la généralité, que

$$c_i \neq 0, \quad 1 \leq i \leq n-1$$

## Méthode de Givens

Nous allons commencer par établir que les racines des polynômes caractéristiques des sous-matrices

$$B_i = \begin{pmatrix} b_1 & c_1 & & & \\ c_1 & b_2 & c_2 & & \\ & c_2 & \ddots & \ddots & \\ & & \ddots & b_{i-1} & c_{i-1} \\ & & & c_{i-1} & b_i \end{pmatrix}, \quad 1 \leq i \leq n$$

ont des propriétés "d'emboîtement" tout à fait remarquables, illustrées à la figure 6.2-1 ci-après.

# Méthode de Givens

## Théorème

Les polynômes  $p_i(\lambda)$ ,  $\lambda \in R$ , définis pour  $i = 0, 1, \dots, n$ , par les formules de récurrence :

$$p_0(\lambda) = 1$$

$$p_1(\lambda) = b_1 - \lambda$$

$$p_i(\lambda) = (b_i - \lambda) p_{i-1}(\lambda) - c_{i-1}^2 p_{i-2}(\lambda), \quad 2 \leq i \leq n$$

ont les propriétés suivantes :

(1) Le polynôme  $p_i$  est le polynôme caractéristique de la matrice  $B_i, 1 \leq i \leq n$ ;

$$(2) \lim_{\lambda \rightarrow -\infty} p_i(\lambda) = +\infty, \quad 1 \leq i \leq n;$$
$$(3) \ p_j(\lambda_0) = 0 \Rightarrow p_{j-1}(\lambda_0) p_{j+1}(\lambda_0) < 0, \quad 1 \leq i \leq n-1;$$

(4) Le polynôme  $p_i$  possède  $i$  racines réelles distinctes, qui séparent les  $(i + 1)$  racines du polynôme  $p_{i+1}$ ,  $1 \leq i \leq n - 1$ , comme l'indique la figure ci-dessous.









# Méthode de Givens

## Théorème

Soit  $i$  un entier vérifiant  $1 \leq i \leq n$ . Étant donné un nombre  $\mu \in \mathcal{M}$ , on pose

$$\operatorname{sgn} p_i(\mu) = \begin{cases} \text{signe de } p_i(\mu) & \text{si } p_i(\mu) \neq 0 \\ \text{signe de } p_{i-1}(\mu) & \text{si } p_i(\mu) = 0 \end{cases}$$

Alors le nombre  $N(i; \mu)$  de changements de signes entre éléments consécutifs de l'ensemble ordonné

$$E(i; \mu) = \{+, \operatorname{sgn} p_1(\mu), \operatorname{sgn} p_2(\mu), \dots, \operatorname{sgn} p_l(\mu)\}$$

est égal au nombre de racines du polynôme  $p_i$  qui sont  $< \mu$ .

**Démonstration.** voir *Introduction à l'analyse numérique matricielle et à l'optimisation* de Philippe G. Ciarlet, page 122.



## Méthode de Givens

**Remarque.**

Appelons  $M(i; \mu)$  le nombre de paires consécutives de même signe ( $\{+, +\}$  ou  $\{-, -\}$ ) trouvées dans l'ensemble ordonné  $E(i; \mu)$ ; par exemple, si

$$E(10; \mu) = \{+, +, -, +, -, -, +, -, -, -, +\}$$

$M(10; \mu) = 4$ , tandis que  $N(10; \mu) = 6$ . On vérifie d'ailleurs facilement que

$$M(i; \mu) + N(i; \mu) = i, \quad \text{pour } i = 1, \dots, n$$

de sorte que le nombre  $M(i; \mu)$  est égal au nombre de racines du polynôme  $p_i$  qui sont  $\geq \mu$ .

Le résultat du théorème précédent permet d'approcher d'aussi près qu'on veut les valeurs propres de la matrice  $B = B_n$ , et même de calculer directement une valeur propre de rang donné. Supposons en effet qu'on souhaite approcher la  $i$ -ème valeur propre  $\lambda_i = \lambda_i^n$  de la matrice  $B$ , l'entier  $i$  étant fixé (on suppose comme précédemment les valeurs propres  $\lambda_1, \dots, \lambda_n$  rangées par ordre croissant ; on rappelle qu'elles sont toutes distinctes d'après le théorème de Givens).

# Méthode de Givens

### Application :

On souhaite calculer la valeur propre de  $B$  de rang  $j_0$  :

$$\lambda_1 < \lambda_2 < \cdots < \lambda_{i_0} < \cdots < \lambda_n.$$

On part d'un intervalle  $[\alpha_0, \beta_0]$  qui contient la valeur propre que l'on cherche, par exemple :

$$[\alpha_0, \beta_0] = [-\|B\|, \|B\|]$$

où  $\| \cdot \|$  est une norme matricielle. Soit alors :

$$\mu_0 = \frac{\alpha_0 + \beta_0}{2}.$$

# Méthode de Givens

Utilisant le théorème précédent, on a :

Si  $N(n, \mu_0) \geq j_0$ , alors  $\lambda_{j_0} \in [\alpha_0, \mu_0]$ ,  $\beta_1 = \mu_0$ .

Si  $N(n, \mu_0) < j_0$ , alors  $\lambda_{j_0} \in [\mu_0, \alpha_0]$ ,  $\alpha_1 = \mu_0$ .

De telle sorte que  $\lambda_{j_0} \in [\alpha_k, \beta_k]$ .

Ainsi, on construit une suite d'intervalles  $[\alpha_k, \beta_k]$  tels que  $\lambda_{j_0} \in [\alpha_k, \beta_k]$ .

On arrête alors les itérations lorsque  $|\alpha_k - \beta_k| \leq \varepsilon$ , où  $\varepsilon$  est un paramètre assez petit,

et on prend comme approximation de  $\lambda_{j_0}$  la valeur  $(\alpha_k + \beta_k)/2$ .

# Sommaire

- 1 Introduction
- 2 Méthodes de type puissance itérée
- 3 Méthode de Jacobi
- 4 La méthode de Givens-Householder (bissection)
- 5 Méthode QR
- 6 Méthode SVD (Décompositon en Valeurs Singulières)
- 7 Méthode de Lanczos
- 8 Conlusion

introduction [1]

La méthode QR, due à J. C. F. Francis et à V. N. Kublanovskaya, est la méthode la plus couramment utilisée pour le calcul de l'ensemble des valeurs propres d'une *matrice quelconque*, notamment non symétrique. Naturellement, elle s'applique *a fortiori* aux matrices symétriques, pour lesquelles elle est au moins autant efficace que la méthode de Jacobi. Nous indiquerons ensuite quelques compléments, concernant notamment la mise en œuvre pratique de la méthode.







## Méthode QR

On obtient ainsi une suite de matrices  $A_k$  qui sont toutes semblables à la matrice  $A$ , puisque

$$A_2 = R_1 Q_1 = Q_1^* A Q_1,$$

...

$$A_{k+1} = R_k Q_k = Q_k^* A_k Q_k = \cdots = (Q_1 Q_2 \cdots Q_k)^* A (Q_1 Q_2 \cdots Q_k).$$

Sous des hypothèses assez restrictives, nous établissons dans le théorème ci-dessous que les matrices  $A_k$  "deviennent" triangulaires supérieures, en ce sens que

$$\lim_{k \rightarrow \infty} (A_k)_{ij} = 0 \quad \text{pour } j < i,$$

tandis que les éléments diagonaux des matrices  $A_k$  convergent vers les valeurs propres de la matrice  $A$ .



# Méthode QR

## Théorème

*On suppose que la matrice  $A$  est inversible et que ses valeurs propres sont toutes de modules différents. Il existe donc (au moins) une matrice inversible  $P$  telle que*

$$A = P\Lambda P^{-1}, \quad \text{avec } \Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n),$$

*et*

$$|\lambda_1| > |\lambda_2| > \dots > |\lambda_n| > 0.$$

*On suppose que la matrice  $P^{-1}$  admet une factorisation  $LU$ . Alors la suite de matrices  $(A_k)_{k \geq 1}$  est telle que*

$$\lim_{k \rightarrow \infty} (A_k)_{ii} = \lambda_i, \quad 1 \leq i \leq n,$$

$$\lim_{k \rightarrow \infty} (A_k)_{ij} = 0, \quad 1 \leq j < i \leq n.$$

# Méthode QR

**Démonstration.** voir *Introduction à l'analyse numérique matricielle et à l'optimisation de Philippe G. Ciarlet, page 125.*





1. *Journal of Management Studies*, 1997, 34, 1, 1-14.

(1) D. 1.

[illegible]

Figure 1. The effect of the concentration of the *Agrobacterium* suspension on the transformation efficiency of *Agrobacterium* strains.

(2) On a :

$$\forall k \in 1, n, \quad u_k = \sum_{j=1}^k \langle u_k, e_j \rangle e_j.$$

De plus, la matrice de passage de  $\mathcal{B}$  à  $\mathcal{U}$  est triangulaire supérieure avec des coefficients diagonaux  $\langle u_k, e_k \rangle$  strictement positifs pour  $k = 1, \dots, n$ .



# Code - Partie 1

```

1 % Definition de la matrice A
2 A = [1 2 3; 4 5 6; 7 8 0];
3
4 % Recuperation des dimensions de la matrice A
5 [n, m] = size(A);
6
7 % Initialisation de Q avec A
8 Q = A;
9
10 % Normalisation de la premiere colonne de Q
11 Q(:,1) = Q(:,1) / norm(Q(:,1));
12
13 % Processus de Gram-Schmidt pour orthonormaliser les
    colonnes de Q
14 for i = 2:m % Parcours des colonnes de Q
15     for j = 1:i-1 % Projection sur les vecteurs deja
        orthonormes
16         Q(:,i) = Q(:,i) - (A(:,i)' * Q(:,j)) * Q(:,j);
17     end
18     % Normalisation de la colonne i
19     Q(:,i) = Q(:,i) / norm(Q(:,i));
20 end

```

[illegible]

\_\_\_\_\_

1. [Introduction](#)

mais cette méthode, pourtant “naturelle”, est à écarter en règle générale, car elle conduit à des propagations parfois désastreuses d’erreurs d’arrondi : on lui préférera donc sans hésiter la méthode basée sur l’utilisation des matrices de Householder.

Pour  $1 \leq k \leq n$  on calcule  $A_k = H_k A_{k-1}$  où la matrice  $H_k$  est construite de façon à annuler les  $n - k$  dernières coordonnées de la  $k$ -ième colonne de  $A_{k-1}$ . On vérifie que  $H_k A_{k-1}$  laisse invariant les  $k - 1$  premières lignes et colonnes de  $A_{k-1}$ .

\_\_\_\_\_

- \_\_\_\_\_

—

—

### 3. Choisissez

$$H_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & H'_3 \end{bmatrix}$$

de manière à ce que

$$A_3 \equiv H_3 A_2 = \begin{bmatrix} x & x & x & x \\ 0 & x & x & x \\ 0 & 0 & x & x \\ 0 & 0 & 0 & x \end{bmatrix}.$$

On obtient donc  $H_n \cdots H_1 A = A_n$  où  $A_n = R$  est triangulaire supérieure avec des éléments strictement positifs sur la diagonale.

Donc,  $A = (H_1 \cdots H_n)A_n = QR$  où  $Q = H_1 \cdots H_n$  vérifie  $Q^t Q = I_n$ .

# Code - Partie 1

```

1 % Definition de la matrice A
2 A = [1 2 3; 4 5 6; 7 8 0];
3 % Initialisation de R avec la matrice A
4 R = A;
5 % Determination des dimensions de la matrice A
6 [n,m] = size(A);
7 % Initialisation de la matrice identite de taille m (
   matrice orthogonale Q)
8 Q = eye(m);
9 % Initialisation de l indice de ligne
10 j = 1;
11 % Boucle pour effectuer la decomposition de Householder
12 for i = 1:m
13 % Extraction de la i-eme colonne de R
14 a = R(:,i);
15 % Selection du sous-vecteur a partir de l element j jusqu
   a la fin
16 v = a(j:end);
17 % Construction du vecteur Householder
18 if v(1) >= 0
19 w = v;
20 w(1) = w(1) + norm(w); % Modification du premier element

```

## Code - Partie 2

```

1 else
2     w = v;
3     w(1) = w(1) - norm(w); % Modification du premier element
4 end
5 % Creation d'une matrice identite de la taille de w
6 I = eye(length(w));
7 % Calcul de la matrice de transformation de Householder
8 M = I - (2 * w * w') / (w' * w);
9 % Construction de la matrice de transformation globale H
10 H = eye(m);
11 H(j:m, j:m) = M;
12 % Application de la transformation sur R
13 R = H * R;
14 % Mise a jour de la matrice orthogonale Q
15 Q = Q * H;
16 % Incrementation de j pour traiter la sous-matrice
    suivante
17 j = j + 1;
18 end
19 % Affichage des resultats
20 R, Q, Q'*Q, Q*R

```



# QR Householder

Les résultats sont les suivants :

$$R = \begin{pmatrix} -8.1240 & -9.6011 & -3.3235 \\ 0.0000 & 0.9045 & 4.5227 \\ 0.0000 & 0.0000 & 3.6742 \end{pmatrix}$$

$$Q = \begin{pmatrix} -0.1231 & 0.9045 & -0.4082 \\ -0.4924 & 0.3015 & 0.8165 \\ -0.8616 & -0.3015 & -0.4082 \end{pmatrix}$$

$$Q' * Q = \begin{pmatrix} 1.0000 & 0.0000 & 0.0000 \\ 0.0000 & 1.0000 & 0.0000 \\ 0.0000 & 0.0000 & 1.0000 \end{pmatrix}$$

$$Q * R = \begin{pmatrix} 1.0000 & 2.0000 & 3.0000 \\ 4.0000 & 5.0000 & 6.0000 \\ 7.0000 & 8.0000 & 0.0000 \end{pmatrix}$$

Après avoir déterminé les matrices  $Q$  et  $R$  de la décomposition, nous revenons à l'algorithme QR initial afin de déterminer une matrice tridiagonale semblable à la matrice  $A$ .

# Méthode QR

```
1  % Initialisation de la matrice A
2  A = [1 2 3; 4 5 6; 7 8 0];
3
4  % Boucle pour appliquer l algorithme QR 100 fois
5  for i = 1:100
6  % Decomposition QR de la matrice A
7  [Q,R] = qr(A);
8
9  % Mise a jour de A avec le produit R*Q
10 A = R*Q;
11 end
12
13 % Affichage de la matrice resultante A
14 A
15
16 % Calcul et affichage des valeurs propres de la matrice
    originale
17 V = eig(A)
```

$$A = \begin{pmatrix} 12.1229 & -3.7399 & 3.0183 \\ 0.0000 & -5.7345 & 0.9503 \\ 0.0000 & 0.0000 & -0.3884 \end{pmatrix}$$

$$V = \begin{pmatrix} 12.1229 \\ -5.7345 \\ -0.3884 \end{pmatrix}$$





## Méthode QR

En particulier :

$$A\Omega_k e_1 = \Omega_k T e_1 = \lambda_1 \Omega_k e_1$$

et  $\Omega_k e_1$ , c'est-à-dire le premier vecteur colonne de  $\Omega_k$ , est vecteur propre de  $A$  pour la valeur propre  $\lambda_1$ . Ensuite, on vérifie que pour  $i \geq 2$ , le vecteur  $q^i = (q_j^i)_{j=1, \dots, N}$  est vecteur propre pour  $\lambda_i$  s'il vérifie :

$$\begin{cases} q_j^i = 0 & \forall j = i+1, \dots, N \\ q_i^i = 1 \\ q_j^i = -(t_{j,j+1}q_{j+1}^i + \dots + t_{ji}q_i^i)/(\lambda_j - \lambda_i) & \forall j = i-1, \dots, 1. \end{cases}$$

Si on ne s'intéresse qu'à quelques vecteurs propres, on peut aussi utiliser la méthode de la puissance itérée avec translation une fois déterminée la valeur propre.

**7. Performances :** QR reste moins performant que des méthodes plus sophistiquées comme celles de Jacobi lorsqu'il s'agit d'obtenir simultanément toutes les valeurs propres et vecteurs propres.



# Sommaire

- 1 Introduction
- 2 Méthodes de type puissance itérée
- 3 Méthode de Jacobi
- 4 La méthode de Givens-Householder (bissection)
- 5 Méthode QR
- 6 Méthode SVD (Décompositon en Valeurs Singulières)**
- 7 Méthode de Lanczos
- 8 Conlusion

## La décomposition en valeurs singulières [4] [5]

## Théorème (Décomposition en valeurs singulières)

Soit  $A \in \mathbb{C}^{m \times n}$  (réelle ou complexe). Il existe deux matrices orthogonales  $U \in \mathbb{C}^{m \times m}$  et  $V \in \mathbb{C}^{n \times n}$  telles que :

$$A = U\Sigma V^\top \quad \text{avec} \quad \Sigma = \text{diag}(\sigma_1, \dots, \sigma_p, 0, \dots, 0) \in \mathbb{R}^{m \times n}.$$

avec  $p = \min(m, n)$  et  $\sigma_1 \geq \dots \geq \sigma_p \geq 0$ . Les  $\sigma_i$  sont les valeurs singulières de  $A$ .

Les matrices sont obtenues comme suit :

- (i) les valeurs singulières  $\sigma_i$  sont les racines carrées des valeurs propres  $\lambda_i$  à la fois de  $A^T A$  et  $AA^T$  :

$$\sigma_j = \sqrt{\lambda_j}$$

- (ii)  $U$  est la matrice des vecteurs propres de  $A^T A$ .
- (iii)  $V$  est la matrice des vecteurs propres de  $AA^T$ .



## SVD : pseudo-inverse

**Définition : pseudo-inverse :** Soit  $A \in \mathbb{R}^{m \times n}$  une matrice de rang  $r$ , ainsi que sa décomposition en valeurs singulières  $A = U\Sigma V^T$ . La matrice  $A^\dagger = V\Sigma^\dagger U^T$  est appelée matrice pseudoinverse ou inverse généralisée de  $A$ , avec :

$$\Sigma^\dagger = \text{diag}(1/\sigma_1, \dots, 1/\sigma_r, 0, \dots, 0)$$

**Remarques :**

- $A^\dagger A = I_r$  (matrice identité de rang  $r$ )
- Si  $\text{rg}(A) = n < m$ , alors  $A^\dagger = (A^T A)^{-1} A^T$
- Si  $\text{rg}(A) = n = m$ ,  $A^\dagger = A^{-1}$

# SVD

Un système linéaire  $Ax = b$  avec  $A \in \mathbb{R}^{m \times n}$  est dit *sous-déterminé* si  $m < n$  (infinité de solutions), et *sur-déterminé* si  $m > n$  (pas de solution en général).

Dans de nombreuses applications : imagerie médicale, sismique, météorologie..., le nombre d'observations  $b_j$  est rarement égal au nombre d'inconnues  $x_i$ .

On peut toutefois résoudre le système d'équations normales

$$A^T A x = A^T b: \quad (21)$$

Le système (21) est inversible si et seulement si  $A$  est de rang maximal  $n$ , et dans ce cas, la solution  $x^*$  du problème  $Ax = b$ . existe et est unique.



Les équations normales deviennent :

$$V\Sigma^T\Sigma V^T x = V\Sigma^T U^T b$$

Puisque  $V$  est orthogonale ( $V^T V = I$ ), on multiplie des deux côtés par  $V^T(\Sigma^T\Sigma)^{-1}V$  :

$$\Sigma^T\Sigma x = \Sigma^T U^T b$$

La matrice  $\Sigma^T\Sigma$  est diagonale avec les termes  $\sigma_i^2$  (les carrés des valeurs singulières). On peut donc résoudre pour  $x$  :

$$x = V(\Sigma^T\Sigma)^{-1}\Sigma^T U^T b$$

Ainsi, la solution des équations normales devient :

$$x = A^\dagger b,$$

où  $A^\dagger$  est la pseudo-inverse de  $A$ .





## Example :

L'image (Lena) suivante est largement utilisée pour tester les algorithmes de compression. Dans MATLAB, elle est chargée et stockée dans une matrice  $A$   $512 \times 512$ .



L'image originale est à gauche. Celle au centre, obtenue avec les 20 premières valeurs singulières, bien que simplifiée reste reconnaissable. Avec  $k = 60$ , l'image à droite est très proche de l'originale et nécessite le stockage de 61 500 coefficients (deux matrices de taille  $512 \times 60$  et 60 valeurs singulières), au lieu de 262 144 pour l'image originale.

Bref, la SVD utilise les valeurs singulières et les matrices  $U, V$  dans diverses applications, notamment :

- ◇ Résoudre des systèmes linéaires via la pseudo-inverse.
- ◇ Réduire la dimension en conservant les grandes  $\sigma_i$
- ◇ Analyser les propriétés (rang, conditionnement, normes) de la matrice.
- ◇ Traiter le bruit et résoudre des problèmes complexes.

# Sommaire

- 1 Introduction
- 2 Méthodes de type puissance itérée
- 3 Méthode de Jacobi
- 4 La méthode de Givens-Householder (bissection)
- 5 Méthode QR
- 6 Méthode SVD (Décompositon en Valeurs Singulières)
- 7 Méthode de Lanczos**
- 8 Conlusion





## Méthode de Lanczos

Pour générer une base orthonormale d'un sous-espace de Krylov, nous exploitons la connexion entre la tridiagonalisation de  $A$  et la factorisation QR de  $K(A, q_1, n)$ .

Rappelons que si  $Q^T A Q = T$  est tridiagonale et  $Q Q^T = I_n$ , alors

$$K(A, q_1, n) = QQ^TK(A, q_1, n) = Q[e_1 | Te_1 | T^2e_1 | \dots | T^{n-1}e_1]$$

est la factorisation QR de  $K(A, q_1, n)$ , où  $e_1$  et  $q_1$  sont respectivement les premières colonnes de  $I_n$  et  $Q$ . Ainsi, les colonnes de  $Q$  peuvent être générées en tridiagonalisant  $A$  avec une matrice orthogonale dont la première colonne est  $q_1$ .







# Méthode de Lanczos

pour  $k = 1 : n - 1$ .

L'orthonormalité des vecteurs  $q$  implique

$$\alpha_k = q_k^T A q_k.$$

(Une autre manière de voir cela est que  $T_{ij} = q_i^T A q_j$ ).

De plus, si nous définissons le vecteur  $r_k$  par

$$r_k = (A - \alpha_k I) q_k - \beta_{k-1} q_{k-1}$$

et s'il est non nul, alors

$$q_{k+1} = \frac{r_k}{\beta_k}$$

où

$$\beta_k = \pm \|r_k\|_2.$$









## Théorème

*L'itération de Lanczos (Algorithme) s'exécute jusqu'à  $k = m$ , où*

$$m = \text{rang}(K(A, q_1, n)).$$

De plus, pour  $k = 1 : m$ , nous avons

$$AQ_k = Q_k T_k + r_k e_k^T$$

où  $Q_k = [q_1 \mid \cdots \mid q_k]$  a des colonnes orthonormales qui engendrent  $\mathcal{K}(A, q_1, k)$ ,  $e_k = I_n(:, k)$ ,

*et*

$$T_k = \begin{pmatrix} \alpha_1 & \beta_1 & & & 0 \\ \beta_1 & \alpha_2 & \beta_2 & & \\ & \beta_2 & \ddots & \ddots & \\ & & \ddots & \alpha_{k-1} & \beta_{k-1} \\ 0 & & & \beta_{k-1} & \alpha_k \end{pmatrix}.$$

**Démonstration.** Matrix Computations 4 edition p.550 .

# Sommaire

- 1 Introduction
- 2 Méthodes de type puissance itérée
- 3 Méthode de Jacobi
- 4 La méthode de Givens-Householder (bissection)
- 5 Méthode QR
- 6 Méthode SVD (Décompositon en Valeurs Singulières)
- 7 Méthode de Lanczos
- 8 **Conlusion**







## Références I

- [1] P.G. Ciarlet. *Introduction à l'analyse numérique matricielle et à l'optimisation: cours et exercices corrigés*. Mathématiques appliquées pour la maîtrise. Masson, 1988.
- [2] James W. Demmel. *Applied Numerical Linear Algebra*. Society for Industrial and Applied Mathematics (SIAM), 1997. ISBN : 978-0898713619.
- [3] Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, Baltimore, Maryland, fourth edition, 2013. Library of Congress Control Number : 2012943449. Also available as eBook ISBN 978-1-4214-0859-0.
- [4] Valérie Perrier and Roger Mohr. *La Décomposition en Valeurs Singulières : Analyse numérique et Application à la Vision*, 2011. 50 ans de l'Ensimag, Mai 2011.
- [5] Alfio Quarteroni and Paola Gervasio. *Calcul Scientifique : Exercices et problèmes résolus avec MATLAB et Octave*. Springer-Verlag Italia, 2010.

