

Chapitre 2

Calcul numérique de valeurs propres

Ce chapitre est consacré à l'approximation numérique des valeurs propres $\lambda \in \mathbb{K}$ et vecteurs propres $v \in \mathbb{K}^n$ d'une matrice $A \in \mathcal{M}_{n,n}(\mathbb{K})$ solution du système

$$A v = \lambda v,$$

où \mathbb{K} est un corps (\mathbb{R} ou \mathbb{C}). C'est un problème important rencontré dans le traitement des équations d'évolution, d'équations aux dérivées ordinaires ou partielles et dans l'étude des propriétés algébriques d'une matrice comme le calcul du conditionnement. Les méthodes de calcul sont des algorithmes itératifs car le calcul exact des valeurs propres est en général impossible pour $n \geq 5$. Nous étudierons deux types de méthodes : les algorithmes partiels fournissant seulement certaines valeurs propres et les méthodes globales qui rendent compte de l'ensemble du spectre.

2.1 Quelques exemples de problèmes aux valeurs propres

2.1.1 Algorithme de Google

Dans cette partie nous proposons d'expliquer l'idée de base proposée par Sergey Brin et Larry Page pour mettre au point le moteur de recherche Google en 1999. C'est un algorithme mathématique très efficace, appelé PageRank, qui détermine le degré d'importance d'une page Web. En effet, la plupart des pages Web incluent des liens hypertextes vers d'autres pages, c'est pourquoi lors d'une requête le moteur de recherche sélectionne les pages qui sont en adéquation avec la requête en analysant le contenu de la page Web et, d'autre part, propose un classement des pages en fonction de leur indice de popularité. Ce dernier ne dépend pas de la consultation de la page par les internautes mais seulement du nombre de liens qui pointent sur cette page à partir d'autres pages Web. C'est cet indice qui distingua Google de ses concurrents dès le départ. Nous nous intéressons ici au calcul de cet indice et nous allons voir que celui-ci fait intervenir le calcul des vecteurs propres et des valeurs propres d'une grande matrice. Pour cela, nous considérons qu'il y a n pages Web au total, dans la pratique n est de l'ordre de 8 à 9 milliards en 2005. La structure du Web peut être représentée simplement par une matrice $C \in \mathcal{M}_{n,n}(\mathbb{R}^+)$ telle que

$$c_{i,j} = \begin{cases} 1 & \text{si la page } j \text{ pointe sur la page } i, \\ 0 & \text{sinon.} \end{cases}$$

Les liens d'une page sur elle-même ne sont pas pris en compte, c'est pourquoi nous posons $c_{i,i} = 0$.

La Figure 2.1 illustre par exemple les liens entre quatre pages Web, il est possible de construire une matrice $C \in \mathcal{M}_{4,4}(\mathbb{R})$ ne contenant que des 1 ou des 0 correspondant à ce graphe :

$$C = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

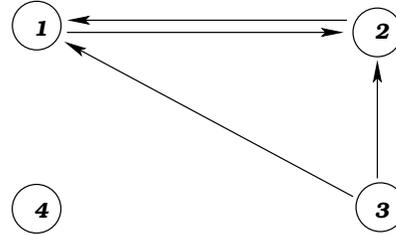


FIG. 2.1 – Exemple de liens entre quatre pages Web.

Nous souhaitons attribuer à chaque page Web i un indice de pertinence $x_i \in \mathbb{R}^+$ de façon à pouvoir classer l'ensemble des pages par score décroissant et présenter à l'utilisateur une liste classée des pages relatives à sa requête. Il s'agit donc de calculer l'indice x_i correspondant à la page i , en partant du principe qu'un lien de la page j pointant sur la page i contribue au score de cette dernière, avec une pondération par x_j (une page ayant un indice élevé a plus de poids qu'une page n'ayant qu'un indice médiocre) et par le nombre total de liens présents sur ladite page $n_j = \sum_{k=1}^n c_{k,j}$ (une page ayant beaucoup de liens a moins d'influence). La composante x_i vérifie ainsi l'équation suivante

$$x_i = \sum_{j=1}^n c_{i,j} \frac{x_j}{n_j}.$$

Le problème du classement des pages du Web se trouve ainsi ramené à la recherche d'un vecteur propre associé à la valeur propre 1 de la matrice \tilde{C} dont les composantes sont $(\tilde{C}_{i,j} = \frac{c_{i,j}}{n_j})_{1 \leq i,j \leq n}$. Cependant, nous ne sommes pas assurés que 1 soit valeur propre, puisque cette matrice contient des lignes entières pleines de zéros. C'est pourquoi nous rajoutons des liens artificiels pondérés par $1/n$ qui tend vers zéro et considérons plutôt la matrice $P = (p_{i,j})_{1 \leq i,j \leq n}$ telle que

$$p_{i,j} = \frac{1}{n_j} c_{i,j} + \frac{1}{n} d_j,$$

où $d \in \mathbb{R}^n$ est tel que

$$d_j = \begin{cases} 1 & \text{si } n_j = 0, \\ 0 & \text{sinon.} \end{cases}$$

La matrice P est la transposée d'une matrice stochastique, c'est-à-dire que ses coefficients sont tous positifs et la somme des coefficients de chaque colonne vaut un, pour le vecteur $e^T = (1, \dots, 1) \in \mathbb{R}^n$, nous avons alors $e^T P = e^T$, ce qui signifie que P admet bien la valeur propre 1.

Dans l'exemple précédent, nous fabriquons des liens artificiels de manière à obtenir une matrice $P \in \mathcal{M}_{4,4}(\mathbb{R})$ dont la transposée est stochastique (voir Figure 2.2).

$$P = \begin{pmatrix} 0 & 1 & 1/2 & 1/4 \\ 1 & 0 & 1/2 & 1/4 \\ 0 & 0 & 0 & 1/4 \\ 0 & 0 & 0 & 1/4 \end{pmatrix}.$$

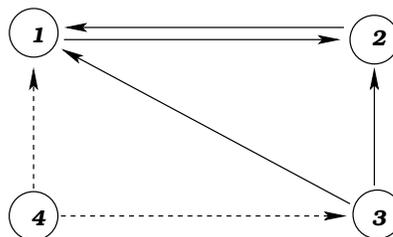


FIG. 2.2 – Exemple de liens entre quatre pages Web et d'ajout de liens artificiels.

Finalement, pour assurer que cette valeur propre est simple, nous posons $A = \alpha P + (1 - \alpha)\frac{1}{n}ee^T$, où nous choisissons $0 < \alpha < 1$. Notons que pour Google, $\alpha = 0,85$ est optimal ! Nous sommes alors conduits à rechercher le vecteur $x \in \mathbb{R}^n$ tel que $Ax = x$, c'est-à-dire le vecteur propre associé à la valeur propre 1 de la matrice A . En appliquant le théorème de Perron-Frobenius, nous montrons que A admet 1 comme valeur propre simple et $\rho(A) = 1$.

Ce système n'est en général pas facile à résoudre pour $n = 4$, et encore moins lorsqu'il s'agit de tenir compte de l'ensemble des pages Web et de l'ensemble des requêtes. Il faut donc résoudre un problème aux valeurs propres avec un grand nombre d'inconnues et pratiquement de manière instantanée.

Dans la suite du chapitre, nous mettons en œuvre une méthode numérique permettant d'approcher la plus grande valeur propre d'une matrice (ici la valeur propre 1), c'est ce que nous appelons une méthode partielle.

2.1.2 Mouvement de ressorts

Considérons un système de deux billes de masse unité reliées par trois ressorts de raideur unité. Notons $x_1(t)$ et $x_2(t)$ les positions des deux billes au temps t , par rapport à leur position d'équilibre. Soit $F_1(t)$, $F_2(t)$ et $F_3(t)$ les forces appliquées sur les billes : ce sont les forces de rappel des trois ressorts. Nous allons voir que l'étude des positions des billes au cours du temps, $x_1(t)$ et $x_2(t)$, se ramène à un calcul de valeurs propres.

Pour cela, nous écrivons d'abord les équations de Newton pour les deux billes :

masse \times *accélération*
= *somme des forces extérieures*, ce qui se traduit par

$$\begin{cases} x_1''(t) = +F_1(t) + F_2(t), \\ x_2''(t) = -F_2(t) + F_3(t). \end{cases}$$

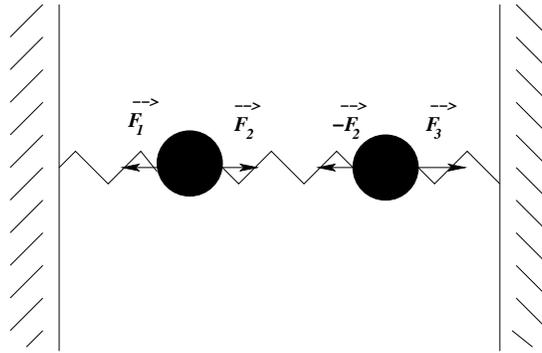


FIG. 2.3 – Étude du mouvement de deux billes maintenues par trois ressorts.

En considérant le cas simplifié où les forces sont proportionnelles à l'allongement du ressort, nous avons

$$F_1(t) = -x_1(t), \quad F_2(t) = x_2(t) - x_1(t), \quad F_3(t) = -x_2(t),$$

et donc

$$\begin{cases} x_1''(t) = -2x_1(t) + x_2(t), \\ x_2''(t) = x_1(t) - 2x_2(t). \end{cases}$$

Cette relation peut s'écrire sous forme matricielle en posant

$$A = \begin{pmatrix} +2 & -1 \\ -1 & +2 \end{pmatrix}, \quad x(t) = \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix},$$

les équations du mouvement deviennent $x''(t) + Ax(t) = 0$. Le vecteur $x(t)$ est alors solution d'un système d'équations différentielles linéaire qui peut être résolu de manière exacte. Les positions des billes $x_1(t)$ et $x_2(t)$ s'écrivent sous la forme $x_1(t) = \alpha_1 \cos(\omega t)$ et $x_2(t) = \alpha_2 \cos(\omega t)$, où les grandeurs α_1 , α_2 et ω restent à déterminer. En substituant ces relations dans les équations du mouvement, il vient

$$\begin{cases} x_1''(t) = -\omega^2 \alpha_1 \cos(\omega t), \\ x_2''(t) = -\omega^2 \alpha_2 \cos(\omega t). \end{cases}$$

Puis après simplification par $\cos(\omega t)$, nous avons

$$A \begin{pmatrix} \alpha_1 \\ \alpha_2 \end{pmatrix} = \omega^2 \begin{pmatrix} \alpha_1 \\ \alpha_2 \end{pmatrix}. \quad (2.1)$$

Le problème consiste donc à chercher α_1 , et α_2 tels que les deux équations ci-dessus soient satisfaites.

Ainsi, la connaissance des valeurs propres et vecteurs propres de la matrice A , c'est-à-dire λ_1 , $\lambda_2 \in \mathbb{R}$ et $v_1, v_2 \in \mathbb{R}^2$ tels que $A v_1 = \lambda_1 v_1$ et $A v_2 = \lambda_2 v_2$ permet de résoudre complètement le système différentiel. En effet, puisque A est symétrique définie positive, les valeurs propres λ_1 et λ_2 sont réelles strictement positives, les grandeurs α_1, α_2 et ω solutions de (2.1) sont alors données par $\omega = \sqrt{\lambda_1}$ et $(\alpha_1, \alpha_2)^T = v_1^T$ ou $\omega = \sqrt{\lambda_2}$ et $(\alpha_1, \alpha_2) = v_2^T$. Dans le cas présent, nous obtenons $\lambda_1 = 3$ et $v_1 = (1, -1)^T$ ou $\lambda_2 = 1$ et $v_2 = (1, 1)^T$.

Nous avons considéré un système de deux billes et trois ressorts, ce qui conduit à la résolution d'un problème de valeurs propres pour une matrice 2×2 . Cependant, nous pouvons généraliser au cas du système de n billes et $n + 1$ ressorts (n grand), ce qui équivaut à la recherche de valeurs propres d'une matrice de taille $n \times n$.

Par la suite, nous mettons au point des algorithmes permettant d'approcher l'ensemble des valeurs propres et des vecteurs propres d'une matrice, c'est ce que nous appelons une méthode globale.

Notons que dans la pratique, il n'est pas toujours nécessaire de chercher toutes les valeurs propres et vecteurs propres. Par exemple, pour l'étude de la réponse dynamique d'un pont, il est intéressant de déterminer simplement les valeurs propres correspondant aux fréquences induites par des piétons marchant sur ce pont.

Définition 2.1.1 Soit $A \in \mathcal{M}_{n,n}(\mathbb{K})$, nous appelons élément propre le couple formé d'une valeur propre et d'un vecteur propre associé : $(\lambda, v) \in \mathbb{K} \times \mathbb{K}^n$ tel que $A v = \lambda v$.

Nous présentons d'abord quelques résultats intermédiaires sur la localisation des valeurs propres. Puis poursuivons avec la méthode de la puissance qui permet d'évaluer la plus grande des valeurs propres. Cet algorithme est d'ailleurs utilisé par le moteur de recherche Google pour le calcul de l'indice de pertinence. Enfin, nous proposons de décrire brièvement les méthodes de Jacobi et QR pour le calcul numérique des valeurs propres et des vecteurs propres d'une matrice quelconque.

2.2 Localisation des valeurs propres

Dans la première partie, nous donnons quelques résultats généraux sur la localisation des valeurs propres puis illustrons rapidement les problèmes de stabilité que nous pouvons rencontrer dans l'approximation des éléments propres.

2.2.1 Approximation des valeurs propres

Commençons par énoncer un résultat, facile à démontrer, qui pourra se révéler utile par la suite pour le calcul d'éléments propres.

Théorème 2.2.1 (Théorème de Gershgorine) Soient $A \in \mathcal{M}_{n,n}(\mathbb{K})$ et $\lambda \in \text{Sp}(A)$ une valeur propre de A . Il existe un indice $i \in \{1, \dots, n\}$ tel que

$$|\lambda - a_{i,i}| \leq \sum_{j \neq i} |a_{i,j}|,$$

c'est-à-dire que toutes les valeurs propres $\lambda \in \text{Sp}(A)$ se trouvent dans $D = \cup_{i=1}^n \mathcal{D}_i$ avec

$$\mathcal{D}_i := \left\{ \lambda \in \mathbb{C}, \quad |\lambda - a_{i,i}| \leq \sum_{j \neq i} |a_{i,j}| \right\}.$$

La démonstration est présentée dans l'Exercice 2.5.2.

Ce théorème fournit une première estimation de la localisation des valeurs propres de la matrice A . Cependant, il ne fournit pas un algorithme pour le calcul des valeurs propres. Avant de construire un tel algorithme, étudions la sensibilité des valeurs propres par rapport aux coefficients de la matrice. Autrement dit, une petite variation sur les coefficients de la matrice A implique-t-elle une forte variation des valeurs propres de A ?

Il arrive en effet que suite à des erreurs d'arrondi, les coefficients d'une matrice $A = (a_{i,j})_{1 \leq i,j \leq n}$ ne soient pas connus de manière exacte. Supposons que les coefficients vérifient

$$a_{i,j}(\varepsilon) = a_{i,j} + \varepsilon c_{i,j}, \quad 1 \leq i, j \leq n,$$

avec $|c_{i,j}| \leq 1$ et le paramètre ε est de l'ordre de la précision de l'ordinateur et est supposé très petit. Lorsque nous voulons calculer une approximation des valeurs propres de la matrice A , il est très important d'étudier l'influence qu'auront ces perturbations à chaque étape du calcul. Montrons pour cela le résultat suivant.

Théorème 2.2.2 (Stabilité des valeurs propres) *Soit $A \in \mathcal{M}_{n,n}(\mathbb{K})$ une matrice diagonalisable : il existe $U \in \mathcal{M}_{n,n}(\mathbb{K})$ inversible et telle que $U^{-1} A U = \text{diag}(\lambda_i)$, où λ_i est valeur propre de A . Notons $A(\varepsilon) = A + \varepsilon C$, la matrice donnée par $a_{i,j}(\varepsilon) = a_{i,j} + \varepsilon c_{i,j}$ où $C \in \mathcal{M}_{n,n}(\mathbb{K})$ quelconque. Alors pour chaque valeur propre $\lambda(\varepsilon)$ de $A(\varepsilon)$, il existe un $\lambda_i \in \text{Sp}(A)$ vérifiant*

$$|\lambda(\varepsilon) - \lambda_i| \leq \varepsilon \text{cond}_\infty(U) \|C\|_\infty.$$

Démonstration : Soit $\lambda(\varepsilon)$ une valeur propre de la matrice approchée $A(\varepsilon)$. D'abord, transformons la matrice $A(\varepsilon)$ de manière à avoir une représentation de la matrice A sous sa forme diagonale :

$$U^{-1} A(\varepsilon) U = U^{-1} A U + \varepsilon U^{-1} C U.$$

Notons par $e_{i,j}$ les éléments de $U^{-1} C U$ et appliquons le Théorème 2.2.1 à la matrice $U^{-1} A(\varepsilon) U$, ce qui implique l'existence d'un indice i tel que

$$|\lambda(\varepsilon) - (U^{-1} A(\varepsilon) U)_{i,i}| \leq \sum_{\substack{j=1 \\ j \neq i}}^n |(U^{-1} A(\varepsilon) U)_{i,j}|.$$

Or, $(U^{-1} A(\varepsilon) U)_{i,i} = \lambda_i + \varepsilon e_{ii}$, où λ_i est une valeur propre de A . L'inégalité triangulaire donne alors

$$|\lambda(\varepsilon) - \lambda_i| \leq |\lambda(\varepsilon) - (\lambda_i + \varepsilon e_{ii})| + \varepsilon |e_{ii}|.$$

En rassemblant les deux dernières inégalités, il vient

$$\begin{aligned} |\lambda(\varepsilon) - \lambda_i| &\leq \varepsilon \sum_{\substack{j=1 \\ j \neq i}}^n |e_{i,j}| + \varepsilon |e_{ii}| \\ &= \varepsilon \sum_{j=1}^n |e_{i,j}| \\ &\leq \varepsilon \max_{1 \leq i \leq n} \sum_{j=1}^n |e_{i,j}| \\ &= \varepsilon \|U^{-1} C U\|_\infty \leq \varepsilon \text{cond}_\infty(U) \|C\|_\infty, \end{aligned}$$

ce qui démontre l'énoncé du théorème. □

2.2.2 Ce qu’il ne faut pas faire !

La première méthode que nous proposons était déjà utilisée par Joseph Louis Lagrange¹ pour calculer les valeurs propres d’une matrice. Elle consiste à d’abord calculer les coefficients du polynôme caractéristique $P(\lambda) = \det(A - \lambda I_n)$, puis à déterminer les racines de ce polynôme $P(\lambda) = (-1)^n \lambda^n + a_{n-1} \lambda^{n-1} + \dots + a_1 \lambda + a_0$.

Si la dimension de la matrice A est très petite (disons $n \leq 3$) ou si nous faisons le calcul en arithmétique exacte, cet algorithme peut être très utile. En revanche, si nous faisons le calcul avec des erreurs d’arrondi, cet algorithme peut donner de mauvaises surprises.

Considérons, par exemple, le problème du calcul des valeurs propres de la matrice diagonale $A = \text{diag}(1, \dots, n)$, dont le polynôme caractéristique est

$$\begin{aligned} P(\lambda) &= (1 - \lambda) \dots (n - \lambda), \\ &= (-1)^n \lambda^n + a_{n-1} \lambda^{n-1} + \dots + a_1 \lambda + a_0. \end{aligned}$$

Supposons de plus que les coefficients calculés ne soient pas connus de manière exacte mais $\hat{a}_i = a_i(1 + \varepsilon_i)$ avec $|\varepsilon_i| \leq \varepsilon \simeq 10^{-8}$. Nous calculons alors les racines d’un polynôme $\hat{P}(\lambda)$ formé par les coefficients $(\hat{a}_i)_{0 \leq i \leq n-1}$. La perturbation des coefficients provoque une grande erreur dans le calcul des racines du polynôme.

Les résultats numériques pour $n = 13$ sont représentés sur la Figure 2.4 pour les polynômes $P(\lambda)$ et $\hat{P}(\lambda)$ et quelques racines sont reportées ci-dessous.

Racines de $P(\lambda)$	Racines de $\hat{P}(\lambda)$
8	$8,086 + 1,001i$
9	$8,086 - 1,001i$
10	$10,439 + 1,683i$
11	$10,439 - 1,683i$
12	$13,105 + 0,963i$
13	$13,105 - 0,963i$

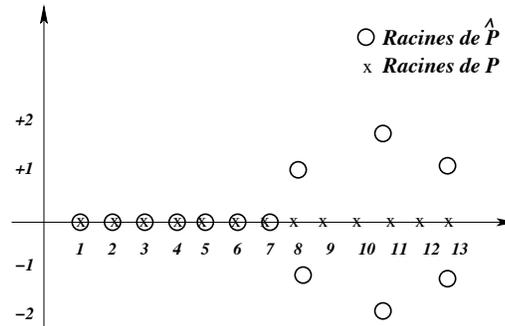


FIG. 2.4 – Racines de $P(\lambda)$ et d’une petite perturbation $\hat{P}(\lambda)$ pour $n = 13$.

Les sept premières racines de P et \hat{P} sont relativement proches les unes des autres avec une erreur de l’ordre de 10^{-4} . En revanche, pour les racines suivantes, l’erreur est de l’ordre de 1, ce qui est largement supérieur à la perturbation des coefficients qui est de l’ordre de 10^{-8} .

¹Mathématicien français et italien (1736-1813). Il est l’un des fondateurs de l’École polytechnique où il enseigna l’analyse.

En conclusion, il faut éviter le calcul des coefficients du polynôme caractéristique car un tel algorithme est numériquement instable.

2.3 Méthode de la puissance

2.3.1 L'algorithme

La méthode de la puissance est une méthode numérique qui permet de déterminer la valeur propre λ_1 de module maximal d'une matrice réelle $A \in \mathcal{M}_{n,n}(\mathbb{R})$. Nous supposons que λ_1 est de multiplicité p , c'est-à-dire,

$$|\lambda_1| = \dots = |\lambda_p| > |\lambda_{p+1}| \geq \dots \geq |\lambda_n|.$$

Soit $x_0 \in \mathbb{K}^n$ un vecteur colonne quelconque, en espérant que x_0 ne soit pas orthogonal à l'espace vectoriel $\text{Ker}(A - \lambda_1 I_n)$ (nous trouverons une explication de cela dans la preuve du Théorème 2.3.1) et pour $k \geq 0$ calculons la suite récurrente $(x^{(k)})_{k \in \mathbb{N}}$ telle que

$$\begin{cases} x^{(0)} = x_0 \in \mathbb{K}^n, \\ x^{(k+1)} = Ax^{(k)}, \quad k \in \mathbb{N}. \end{cases} \quad (2.2)$$

Notons bien que d'un point de vue pratique la méthode de la puissance ne peut pas être implémentée en l'état car l'algorithme pourrait conduire à un `overflow`, c'est-à-dire que les valeurs $\|x^{(k+1)}\|$ sont susceptibles de devenir de plus en plus grandes. Pour remédier à cela et pour fixer la norme du vecteur limite de la suite $(x^{(k)})_{k \in \mathbb{N}}$, nous ajoutons une étape de renormalisation de sorte qu'il existe $j_{k+1} \in \{1, \dots, n\}$ tel que

$$x_{j_{k+1}}^{(k+1)} = \|x^{(k+1)}\|_\infty = 1, \quad (2.3)$$

c'est-à-dire que nous fixons la norme $\|\cdot\|_\infty$ de chaque itéré égale à 1.

Plus précisément, l'algorithme s'écrit sous la forme suivante. Pour $\varepsilon > 0$ un petit paramètre fixé et $x_0 \in \mathbb{R}^n$,

Algorithme 1. Méthode de la puissance

Poser $x^{(0)} = x_0$, $k = 0$ et $\varepsilon^{(0)} = 2\varepsilon$.

Tant que $\varepsilon^{(k)} \geq \varepsilon$

- calculer $z^{(k+1)}$ tel que $z^{(k+1)} = Ax^{(k)}$,

- normaliser le résultat $x^{(k+1)} = z^{(k+1)}/c_{k+1}$ où c_{k+1} est

tel qu'il existe $j_{k+1} \in \{1, \dots, n\}$ vérifiant

$$x_{j_{k+1}}^{(k+1)} = \|x^{(k+1)}\|_{\infty} = 1,$$

- calculer une approximation de la valeur propre

$$\beta^{(k+1)} = (Ax^{(k+1)})_{j_{k+1}},$$

- calculer un résidu : $\varepsilon^{(k+1)} = |\beta^{(k+1)} - \beta^{(k)}|$,

- itérer $k \leftarrow k + 1$.

Fin de tant que

Cet algorithme permet ainsi d'approcher la valeur propre de module maximal et un vecteur propre associé de norme égale à 1. Cependant, la convergence d'un tel algorithme n'est pas toujours assurée.

2.3.2 Un résultat de convergence

Montrons par exemple un résultat de convergence de la méthode de la puissance pour les matrices diagonalisables.

Théorème 2.3.1 (Convergence de la méthode de la puissance) Soient $A \in \mathcal{M}_{n,n}(\mathbb{K})$ une matrice diagonalisable dans \mathbb{K} et λ_1 la plus grande valeur propre en module de multiplicité $p \geq 1$. Supposons qu'il n'y ait pas de valeur propre $\lambda \in \text{Sp}(A)$ telle que $\lambda \neq \lambda_1$ et $|\lambda| = |\lambda_1|$.

Si $x_0 \in \mathbb{K}^n$ est tel que $x_0 \notin \text{Ker}(A - \lambda_1 I_n)^T$. Alors la suite $(x^{(k)})_{k \in \mathbb{N}}$, fournie par l'algorithme (2.2)-(2.3), converge vers un vecteur propre $x \in \mathbb{K}^n$ associé à la valeur propre $\lambda_1 : Ax = \lambda_1 x$.

En outre la suite $\beta^{(k)}$, construite dans l'Algorithme 1, converge vers λ_1 .

Démonstration : Comme la matrice A est diagonalisable dans \mathbb{K} , il existe une base normée de vecteurs propres $(v^i)_{1 \leq i \leq n}$. Pour $1 \leq i \leq n$, notons par λ_i la valeur propre associée au vecteur propre v^i , c'est-à-dire $Av^i = \lambda_i v^i$.

De plus, ordonnons les valeurs propres de sorte que

$$|\lambda_1| = \dots = |\lambda_p| > \dots \geq |\lambda_n| \geq 0,$$

où p désigne la multiplicité de λ_1 .

Le vecteur $x_0 \in \mathbb{K}^n$ s'écrit alors dans la base $\{v^1, \dots, v^n\}$, c'est-à-dire il existe $(\alpha_i)_{1 \leq i \leq n} \subset \mathbb{K}$ tels que $x_0 = \sum_{i=1}^n \alpha_i v^i$ et donc

$$A x_0 = \sum_{i=1}^n \alpha_i \lambda_i v^i.$$

En appliquant une première fois l'algorithme de la puissance (2.2), nous obtenons d'abord $z^{(1)} = A x_0$, puis construisons $x^{(1)}$ par une étape de normalisation. En notant $j_1 \in \{1, \dots, n\}$ tel que $|z_{j_1}^{(1)}| = \|z^{(1)}\|_\infty$, nous posons

$$x^{(1)} = \frac{z^{(1)}}{c_1}, \quad \text{avec } c_1 = z_{j_1}^{(1)} \neq 0,$$

ce qui donne $x_{j_1}^{(1)} = \|x^{(1)}\|_\infty = 1$. En utilisant l'écriture de x_0 dans la base formée par les vecteurs propres $(v^i)_{1 \leq i \leq n}$, nous obtenons

$$x^{(1)} = \frac{1}{c_1} A x_0 = \frac{1}{c_1} \sum_{i=1}^n \alpha_i \lambda_i v^i.$$

Ainsi en appliquant l'algorithme (2.2), nous obtenons à l'issue de la k -ème étape

$$z^{(k)} = A x^{(k-1)} = \frac{1}{c_1 \dots c_{k-1}} A^k x_0 = \frac{1}{c_1 \dots c_{k-1}} \sum_{i=1}^n \alpha_i \lambda_i^k v^i.$$

Puis, en notant $j_k \in \{1, \dots, n\}$ le plus petit indice vérifiant $|z_{j_k}^{(k)}| = \|z^{(k)}\|_\infty$, nous construisons le vecteur $x^{(k)}$,

$$x^{(k)} = \frac{z^{(k)}}{c_k} = \frac{1}{c_1 \dots c_k} \sum_{i=1}^n \alpha_i \lambda_i^k v^i, \quad \text{avec } c_k = z_{j_k}^{(k)} \neq 0$$

de sorte que $x_{j_k}^{(k)} = 1 = \|x^{(k)}\|_\infty$. En factorisant par λ_1^k , il s'écrit encore

$$x^{(k)} = \frac{\lambda_1^k}{c_1 \dots c_k} \sum_{i=1}^n \alpha_i \left(\frac{\lambda_i}{\lambda_1} \right)^k v^i. \quad (2.4)$$

Nous voulons démontrer que ce vecteur tend vers un vecteur propre de A associé à λ_1 . Pour cela, notons d'abord qu'il n'existe pas de valeur propre à la fois différente de λ_1 et de même module. Posons alors $p \in \{1, \dots, n\}$ l'entier vérifiant $\lambda_1 = \dots = \lambda_p$ et $|\lambda_{p+1}| < |\lambda_1|$, puis $z := \sum_{i=1}^p \alpha_i v^i$. Nous obtenons une nouvelle écriture de $x^{(k)}$ telle que

$$x^{(k)} = \frac{\lambda_1^k}{c_1 \dots c_k} z + \frac{\lambda_1^k}{c_1 \dots c_k} \sum_{i=p+1}^n \alpha_i \left(\frac{\lambda_i}{\lambda_1} \right)^k v^i. \quad (2.5)$$

L'inégalité triangulaire permet d'établir

$$1 = \|x^{(k)}\|_\infty \geq \left| \frac{\lambda_1^k}{c_1 \dots c_k} \left\| \|z\|_\infty - \left\| \sum_{i=p+1}^n \alpha_i \left(\frac{\lambda_i}{\lambda_1} \right)^k v^i \right\|_\infty \right\| \right|$$

et puisque $\lim_{k \rightarrow \infty} (\lambda_i/\lambda_1)^k = 0$, pour tout $i \geq p+1$, ceci assure que $\left| \frac{\lambda_1}{c_1 \dots c_k} \right| \|z\|_\infty \leq C$, où $C > 0$ est une constante ne dépendant pas de $k \in \mathbb{N}$. Par conséquent, pour tout $k \geq 0$, la suite complexe $\left(\frac{\lambda_1}{c_1 \dots c_k} z \right)_{k \in \mathbb{N}}$ est bornée.

D'autre part en prenant la norme $\|\cdot\|_\infty$ et en passant à la limite dans (2.5), nous obtenons, compte tenu de ce que nous venons de démontrer,

$$\begin{aligned} 1 = \lim_{k \rightarrow +\infty} \|x^{(k)}\|_\infty &= \lim_{k \rightarrow +\infty} \left\| \frac{\lambda_1^k}{c_1 \dots c_k} \left\| z + \sum_{i=p+1}^n \alpha_i \left(\frac{\lambda_i}{\lambda_1} \right)^k v^i \right\|_\infty \right\|, \\ &= \lim_{k \rightarrow +\infty} \left\| \frac{\lambda_1^k}{c_1 \dots c_k} \|z\|_\infty \right\|. \end{aligned}$$

Notons alors $\bar{j} \in \{1, \dots, n\}$ le plus petit indice tel que $|z_{\bar{j}}| = \|z\|_\infty$ et posons $x = z/z_{\bar{j}}$ le vecteur colinéaire à z tel que $\|x\|_\infty = 1 = x_{\bar{j}}$. Puisque la suite vectorielle $(x^{(k)})_{k \in \mathbb{N}}$ est bornée dans l^∞ , il existe une sous-suite convergente. En écrivant simplement (2.5) pour la composante $\bar{j} \in \{1, \dots, n\}$, il vient alors pour une sous-suite extraite convergente, toujours notée $(x^{(k)})_{k \in \mathbb{N}}$,

$$1 = x_{\bar{j}} = \lim_{k \rightarrow +\infty} x_{\bar{j}}^{(k)} = \lim_{k \rightarrow +\infty} \frac{\lambda_1^k}{c_1 \dots c_k} z_{\bar{j}},$$

d'où

$$\lim_{k \rightarrow \infty} \frac{\lambda_1^k}{c_1 \dots c_k} = \frac{1}{z_{\bar{j}}}.$$

Ceci signifie que toutes les sous-suites extraites convergentes ont la même limite, c'est donc toute la suite $(x^{(k)})_{k \in \mathbb{N}}$ qui converge vers le vecteur x .

Étant donné que les vecteurs $(v^i)_{1 \leq i \leq p}$ sont des vecteurs propres associés à la valeur propre λ_1 , nous vérifions facilement que $Ax = \lambda_1 x$. Assurons-nous également que $x \neq 0_{\mathbb{K}^n}$. En effet, puisque $x_0 \notin \text{Ker}(A - \lambda_1 I_n)^\perp$, il existe donc $1 \leq j_0 \leq p$ tel que $\alpha_{j_0} \neq 0$.

Enfin, montrons que la suite $(\beta^{(k)})$ converge bien vers la valeur propre λ_1 . Par construction, nous avons $\beta^{(k)} = (Ax^{(k)})_{j_k}$, où j_k est l'indice tel que $x_{j_k}^{(k)} = 1$, ceci montre bien que la suite $(\beta^{(k)})_{k \in \mathbb{N}}$ tend vers λ_1 puisque la suite $(x^{(k)})_{k \in \mathbb{N}}$ converge vers le vecteur propre x associé à la valeur propre λ_1 , elle vérifie, par continuité de l'application norme, $(Ax^{(k)})_{j_k} \rightarrow (Ax)_{\bar{j}} = \lambda_1$, lorsque $k \rightarrow \infty$. \square

Des problèmes de convergence peuvent survenir en utilisant la méthode de la puissance lorsque la valeur propre de module maximal n'est pas unique.

Exemple 2.3.1 Lorsque $|\lambda_1| = |\lambda_2|$ avec $\lambda_2 = -\lambda_1$ ou $\lambda_2 = \bar{\lambda}_1$, la méthode ne converge pas forcément. Nous pouvons constater certaines de ces difficultés en calculant la suite $(x^{(k)})_{k \in \mathbb{N}}$ pour

les exemples suivants

$$\begin{pmatrix} 1 & 1 \\ 0 & -1 \end{pmatrix}, \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} \text{ et } \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & -1 \end{pmatrix}.$$

Pour la première matrice, l'ensemble des valeurs propres est donné par $\{-1, 1\}$ mais la méthode de la puissance ne converge pas lorsque nous choisissons $x^{(0)} = (1, -1)^T$.

Dans le second cas, les valeurs propres sont $\{-1, -1, 1, 1\}$ et la méthode de la puissance donne les résultats présentés ci-contre. Ici $(x^{(k)})_{k \in \mathbb{N}}$ ne converge pas vers un vecteur propre de A lorsque k tend vers l'infini.

k	$x^{(k)}$	$\ x^{(k)}\ _\infty$
0	$(1, 0, 1, 1)^T$	1
1	$(1, 0, -1, -1)^T$	1
2	$(1, 0, 1, 1)^T$	1
3	$(1, 0, -1, -1)^T$	1

2.4 Méthode de Jacobi

Nous cherchons à déterminer numériquement les valeurs propres et vecteurs propres d'une matrice A symétrique réelle ($A^T = A$). Nous savons qu'une telle matrice est diagonalisable, c'est-à-dire qu'il existe une matrice réelle et orthogonale U telle que $D = U^T A U$ soit diagonale, la diagonale étant composée des valeurs propres de A . La diagonalisation consiste donc à trouver la matrice U , c'est-à-dire à trouver une base dans laquelle la représentation de A est diagonale.

Commençons par un exemple simple en dimension deux et généralisons ensuite le procédé aux dimensions supérieures.

2.4.1 Cas de la dimension deux

Soient $A \in \mathcal{M}_{2,2}(\mathbb{K})$ une matrice symétrique et P une matrice de rotation

$$A = \begin{pmatrix} \alpha & \beta \\ \beta & \gamma \end{pmatrix}, \quad P = \begin{pmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{pmatrix}.$$

Nous décidons de choisir θ pour que la matrice $PA P^T$ soit diagonale. Pour cela, notons $A^{(0)} = A$ et calculons

$$A^{(1)} = PA^{(0)}P^T = \begin{pmatrix} \alpha' & \beta' \\ \beta' & \gamma' \end{pmatrix},$$

il suffit donc de chercher pour quel angle $\theta \in [0, 2\pi]$ le coefficient β' devient nul. Vérifions alors que θ donné par

$$\cotan(2\theta) = \frac{\cos^2 \theta - \sin^2 \theta}{2 \cos \theta \sin \theta} = \frac{\alpha - \gamma}{2\beta}$$

convient et implique $\beta' = 0$, ce qui donne une matrice $A^{(1)}$ diagonale. En particulier, puisque P est une rotation, c'est donc une matrice orthogonale et elle conserve la norme de Frobenius (définie au Chapitre 1 par (1.3)), d'où

$$\alpha'^2 + 2\beta'^2 + \gamma'^2 = \alpha^2 + 2\beta^2 + \gamma^2.$$

Ainsi, le fait d'annuler le coefficient β' implique que $\alpha'^2 + \gamma'^2 \geq \alpha^2 + \gamma^2$, c'est-à-dire que la méthode de Jacobi consiste à accroître la somme des carrés des termes diagonaux et à diminuer la somme des carrés des termes extradiagonaux. En dimension deux, une seule étape suffit à diagonaliser une matrice A symétrique, la matrice $A^{(1)}$ contient les valeurs propres de A et les vecteurs colonnes de P fournissent les vecteurs propres de la matrice A .

Voyons maintenant comment étendre cette méthode aux dimensions supérieures.

2.4.2 Cas général

Chaque étape k de la méthode de Jacobi consiste à construire une matrice $A^{(k)}$ à partir de la matrice précédente $A^{(k-1)}$ et d'une matrice de rotation choisie de façon à annuler des éléments non diagonaux de $A^{(k-1)}$. Une rotation, d'un angle θ dans le plan défini par les vecteurs d'indices p et q avec $q > p$, est donnée par la matrice orthogonale (*rotation de Givens*)

$$(P_{pq})_{i,j} = \begin{cases} 1 & \text{si } i = j, \text{ avec } j \neq p, j \neq q, \\ \cos \theta & \text{si } i = j = p \text{ ou } i = j = q, \\ \sin \theta & \text{si } i = p \text{ et } j = q, \\ -\sin \theta & \text{si } i = q \text{ et } j = p, \\ 0 & \text{sinon.} \end{cases}$$

Nous avons aussi

$$P_{pq} = \begin{pmatrix} I_{p-1} & 0 & 0 & 0 & 0 \\ 0 & \cos \theta & \sin \theta & 0 & 0 \\ 0 & 0 & I_{q-p-1} & 0 & 0 \\ 0 & -\sin \theta & 0 & \cos \theta & 0 \\ 0 & 0 & 0 & 0 & I_{n-q} \end{pmatrix}.$$

La matrice $P_{pq} \in \mathcal{M}_{n,n}(\mathbb{R})$ ainsi construite satisfait les propriétés suivantes.

Lemme 2.4.1 *La matrice P_{pq} vérifie les propriétés suivantes.*

– Pour la norme de Froebenius donnée par $\|A\|_F = \sqrt{\text{tr}(A^*A)}$,

$$\|P_{pq}\|_F^2 = \sum_{i,j=1}^n |(P_{pq})_{i,j}|^2 = n.$$

– La matrice P_{pq} est orthogonale, c'est-à-dire $P_{pq} P_{pq}^T = I_n$.

Démonstration : Le premier résultat est évident $\|P_{pq}\|_F^2 = (n-2) \times 1^2 + 2 \times (\cos^2 \theta + \sin^2 \theta) = n$. D'autre part en écrivant le produit $P_{pq} P_{pq}^T$ et en effectuant un produit matriciel par bloc, nous obtenons le résultat $P_{pq} P_{pq}^T = I_n$. \square

Posons alors $A^{(0)} = A$ puis pour $k \geq 0$, calculons par récurrence $A^{(k+1)}$ comme la rotation de $A^{(k)}$ par la matrice de rotation P_{pq} , $A^{(k+1)} = P_{pq} A^{(k)} P_{pq}^T$. La représentation de la matrice $A^{(k+1)} = (a_{i,j}^{(k+1)})_{1 \leq i,j \leq n}$ (qui reste symétrique) dans la nouvelle base obtenue après rotation s'écrit en posant $c = \cos \theta$ et $s = \sin \theta$

$$a_{i,j}^{(k+1)} = \begin{cases} a_{i,j}^{(k)} & \text{si } j \neq p, q \text{ et } i \neq p, q, \\ c a_{i,p}^{(k)} - s a_{i,q}^{(k)} & \text{si } j = p \text{ avec } i \neq p, q, \\ c a_{i,q}^{(k)} + s a_{i,p}^{(k)} & \text{si } j = q \text{ avec } i \neq p, q, \\ c^2 a_{p,p}^{(k)} + s^2 a_{q,q}^{(k)} - 2s c a_{p,q}^{(k)} & \text{si } i = p \text{ et } j = p, \\ s^2 a_{p,p}^{(k)} + c^2 a_{q,q}^{(k)} + 2s c a_{p,q}^{(k)} & \text{si } i = q \text{ et } j = q, \\ (c^2 - s^2) a_{p,q}^{(k)} + c s (a_{p,p}^{(k)} - a_{q,q}^{(k)}) & \text{si } (i, j) = (p, q) \text{ ou } (i, j) = (q, p). \end{cases}$$

Observons que seuls les éléments sur les lignes et colonnes p et q sont modifiés. L'idée consiste alors à choisir l'angle de la rotation de façon à annuler le terme $a_{p,q}^{(k+1)}$, ce qui conduit à

$$\cotan(2\theta) = \frac{\cos^2 \theta - s^2}{2cs} = \frac{a_{q,q}^{(k)} - a_{p,p}^{(k)}}{2a_{p,q}^{(k)}},$$

mais ce choix modifie les autres éléments non diagonaux.

Nous résumons les modifications induites par la rotation dans la proposition suivante :

Proposition 2.4.1 *Supposons que la matrice $A^{(0)} = A$ soit symétrique. Alors pour tout $k \geq 0$ nous construisons la matrice $A^{(k+1)} = P_{pq}^T A^{(k)} P_{pq}$, laquelle est symétrique et vérifie $\|A^{(k+1)}\|_F = \|A^{(k)}\|_F$.*

D'autre part, en notant $S^{(k+1)}$ la somme des carrés des termes diagonaux de $A^{(k+1)}$, nous avons

$$S^{(k+1)} := \sum_{i=1}^n [a_{i,i}^{(k+1)}]^2 = S^{(k)} + 2[a_{p,q}^{(k)}]^2 \geq S^{(k)}.$$

Démonstration : D'une part, nous rappelons que $\|A\|_F^2 = \text{tr}(A A^*)$, ce qui signifie en particulier pour toute matrice A , si U est orthogonale $\|AU\|_F = \|A\|_F$. D'où le résultat pour $A^{(k+1)}$ et $A^{(k)}$, $\|A^{(k)}\|_F = \|A^{(k+1)}\|_F$. Ensuite, en calculant la somme des éléments extra-diagonaux au carré, il vient

$$\begin{aligned} \sum_{\substack{i,j=1 \\ i \neq j}}^n |a_{i,j}^{(k+1)}|^2 &= \sum_{\substack{i=1 \\ i \neq p,q}}^n \left(\sum_{\substack{j=1 \\ j \neq i,p,q}}^n |a_{i,j}^{(k)}|^2 + (c a_{i,p}^{(k)} - s a_{i,q}^{(k)})^2 + (c a_{i,q}^{(k)} + s a_{i,p}^{(k)})^2 \right), \\ &= \sum_{\substack{i,j=1 \\ i \neq j}}^n |a_{i,j}^{(k)}|^2 - 2[a_{p,q}^{(k)}]^2. \end{aligned}$$

En combinant ce dernier résultat avec la conservation de la norme de Froebenius, nous avons donc $S^{(k+1)} = S^{(k)} + 2[a_{p,q}^{(k)}]^2$. \square

Nous vérifions que la somme $S^{(k)}$ croît lorsque l'itération k augmente et est bornée par la somme des carrés de tous les termes, c'est-à-dire la norme de Froebenius de $\|A\|_F^2$. Ainsi, pour une matrice $A \in \mathcal{M}_{n,n}(\mathbb{R})$ donnée et un petit paramètre $\varepsilon > 0$ fixé, l'algorithme de la méthode de Jacobi s'écrit sous forme compacte :

Algorithme 2. Méthode de Jacobi pour le calcul de valeurs propres

Poser $A^{(0)} = A, k = 0$.

Tant que $\max_{p \neq q} |a_{p,q}^{(k)}| \geq \varepsilon$

- chercher $p_0, q_0 \in \{1, \dots, n\}$ tel que $p_0 \neq q_0$,

$$|a_{p_0, q_0}^{(k)}| = \max\{|a_{p,q}^{(k)}|, p, q \in \{1, \dots, n\}, p \neq q\},$$

- construire la matrice orthogonale $P^{(k)} := P_{p_0 q_0}$

de rotation d'angle θ tel que

$$\cotan(2\theta) = \frac{\cos^2 \theta - \sin^2 \theta}{2 \cos \theta \sin \theta} = \frac{a_{q_0, q_0}^{(k)} - a_{p_0, p_0}^{(k)}}{2 a_{p_0, q_0}^{(k)}},$$

- construire la matrice $A^{(k+1)} = (P^{(k)})^T A^{(k)} P^{(k)}$,

- itérer $k \leftarrow k + 1$.

Fin de tant que.

En appliquant ce procédé à une matrice de départ $A \in \mathcal{M}_{n,n}(\mathbb{K})$, les itérés successifs vont converger vers une matrice diagonale D ne contenant que les valeurs propres de A sur sa diagonale.

La méthode de Jacobi fournit également les vecteurs propres de A . En effet, pour (λ, x) un élément propre de A , il vérifie $Ax = \lambda x$. L'application de la méthode de Jacobi donne

$$A^{(k+1)} = (P^{(k)})^T A^{(k)} P^{(k)},$$

où $P^{(k)}$ est une matrice de rotation et donc par récurrence en notant $Q^{(k)}$ la matrice orthogonale définie par $Q^{(k)} := \prod_{i=0}^k P^{(i)}$, il vient $A^{(k+1)} = (Q^{(k)})^T A Q^{(k)}$. En notant $y^{(k+1)} = (Q^{(k)})^T x$, nous avons $A^{(k+1)} y^{(k+1)} = \lambda y^{(k+1)}$, ce qui signifie que $y^{(k+1)}$ est vecteur propre de $A^{(k+1)}$. Puisque $A^{(k+1)}$ converge vers une matrice diagonale D dont les vecteurs propres sont les vecteurs de la base canonique, le vecteur propre x de A est donné par $x = \lim_{k \rightarrow \infty} Q^{(k)} e_i$ où e_i est le vecteur de la base canonique associé à la valeur propre λ , c'est-à-dire $D_{i,i} = \lambda$.

En définitive la méthode de Jacobi pour l'approximation d'éléments propres est un algorithme itératif, c'est-à-dire que la convergence est asymptotique lorsque k tend vers l'infini. Nous montrons alors le résultat général suivant [7].

Théorème 2.4.1 Soient $A \in \mathcal{M}_{n,n}(\mathbb{R})$ une matrice symétrique et $(A^{(k)})_{k \in \mathbb{N}} \subset \mathcal{M}_{n,n}(\mathbb{R})$ la suite définie par l'algorithme de Jacobi. Alors la suite $A^{(k)}$ converge vers une matrice diagonale D ne

contenant que les valeurs propres de A . De plus, les vecteurs colonnes du produit des matrices de passage P_{pq} construites à chaque itération convergent vers les vecteurs propres associés.

2.5 La méthode QR pour le calcul des valeurs propres

Nous avons déjà vu au Chapitre 1[section 1.3.4] qu'une matrice A inversible admet une factorisation QR avec Q une matrice orthogonale $Q^T Q = I_n$ et R une matrice triangulaire supérieure. La méthode QR pour la recherche de valeurs propres est donnée par l'algorithme suivant. Pour $A \in \mathcal{M}_{n,n}(\mathbb{R})$ et $\varepsilon > 0$ un petit paramètre fixé,

Algorithme 3. Calcul de l'ensemble des valeurs propres par la méthode QR

Poser $A^{(0)} = A$ et $k = 0$.

Tant que $\max_{i \neq j} |a_{i,j}^{(k)}| \geq \varepsilon$

- calculer $R^{(k)}$ et $Q^{(k)}$ telles que

$$A^{(k)} = Q^{(k)} R^{(k)},$$

- calculer $A^{(k+1)}$ par

$$A^{(k+1)} = R^{(k)} Q^{(k)},$$

- itérer $k \leftarrow k + 1$.

Fin de tant que.

La matrice $A^{(k)}$ est alors censée converger vers une matrice diagonale ne comprenant que les valeurs propres de A . Plus précisément, nous avons le résultat suivant [25]

Théorème 2.5.1 Soit $A \in \mathcal{M}_{n,n}(\mathbb{R})$ une matrice dont les valeurs propres $(\lambda_i)_{1 \leq i \leq n}$ sont telles que $|\lambda_1| > \dots > |\lambda_n| > 0$. Alors la suite des matrices $A^{(k)}$ construite par la méthode QR donnée ci-

dessus converge vers une matrice triangulaire T contenant les valeurs propres de A sur la diagonale

$$\lim_{k \rightarrow \infty} A^{(k)} = \begin{pmatrix} \lambda_1 & t_{1,2} & \dots & t_{1,n} \\ 0 & \lambda_2 & t_{2,3} & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_n \end{pmatrix}.$$

Si de plus A est symétrique, alors T est diagonale.

Nous présentons dans la Figure 2.5 une illustration de ce que donne cet algorithme de factorisation QR pour une matrice symétrique aléatoire. En quelques itérations l'algorithme converge vers une matrice diagonale contenant les valeurs propres approchées et les coefficients extradiagonaux sont en dessous d'un seuil $\varepsilon > 0$ fixé.

En partant d'une matrice pleine aléatoire $A \in M_{40,40}(\mathbb{R})$, les images de la Figure 2.5 représentent différentes itérations. Après seulement 6 itérations, la matrice $A^{(6)}$ contient un fort pourcentage de coefficients extradiagonaux en dessous du seuil fixé $\varepsilon = 10^{-4}$ mais pour obtenir que l'ensemble des coefficients extradiagonaux soit en dessous de ε , cela nécessite environ 200 itérations.

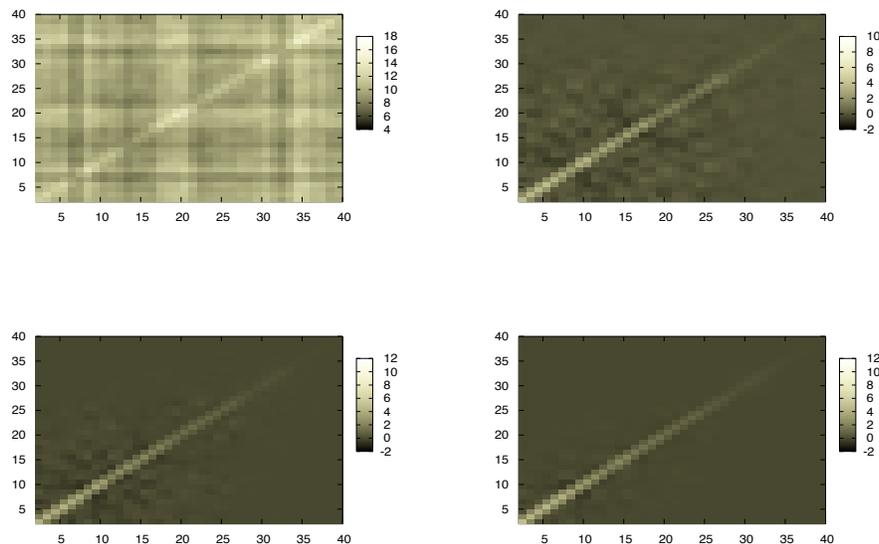


FIG. 2.5 – Illustration des itérations successives obtenues par la méthode QR pour une matrice A symétrique aléatoire.