

1.6 Valeurs propres et vecteurs propres

Les techniques de recherche des éléments propres, c.à.d. des valeurs et vecteurs propres (voir Définition 1.2 page 7) d'une matrice sont essentielles dans de nombreux domaines d'application, par exemple en dynamique des structures : la recherche des modes propres d'une structure peut s'avérer importante pour le dimensionnement de structures sous contraintes dynamiques ; elle est essentielle dans la compréhension des phénomènes acoustiques.

On peut se demander pourquoi on parle dans ce chapitre, intitulé "systèmes linéaires" du problème de recherche des valeurs propres : il s'agit en effet d'un problème non linéaire, les valeurs propres étant les solutions du polynôme caractéristique, qui est un polynôme de degré n , où n est la dimension de la matrice. Il n'est malheureusement pas possible de calculer numériquement les valeurs propres comme les racines du polynôme caractéristique, car cet algorithme est instable : une petite perturbation sur les coefficients du polynôme peut entraîner une erreur très grande sur les racines (voir par exemple le chapitre 5 du polycopié d'E. Hairer, cité dans l'introduction de ce cours, en ligne sur le web). De nombreux algorithmes ont été développés pour le calcul des valeurs propres et vecteurs propres. Ces méthodes sont en fait assez semblables aux méthodes de résolution de systèmes linéaires. Dans le cadre de ce cours, nous nous restreignons à deux méthodes très connues : la méthode de la puissance (et son adaptation de la puissance inverse), et la méthode dite *QR*.

1.6.1 Méthode de la puissance et de la puissance inverse

Pour expliquer l'algorithme de la puissance, commençons par un exemple simple. Prenons par exemple la matrice

$$A = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}$$

dont les valeurs propres sont 1 et 3, et les vecteurs propres associés $\mathbf{f}^{(1)} = \frac{\sqrt{2}}{2} \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ et $\mathbf{f}^{(2)} = \frac{\sqrt{2}}{2} \begin{bmatrix} 1 \\ -1 \end{bmatrix}$. Partons de $\mathbf{x} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ et faisons tourner scilab en itérant les instructions suivantes :

```
-->x = A * x ; x = x/norm(x)
```

ce qui correspond à la construction de la suite

$$\mathbf{x}^{(0)} = \frac{\mathbf{x}}{\|\mathbf{x}\|}, \mathbf{x}^{(1)} = \frac{A\mathbf{x}^{(0)}}{\|A\mathbf{x}^{(0)}\|}, \dots, \mathbf{x}^{(k+1)} = \frac{A\mathbf{x}^{(k)}}{\|A\mathbf{x}^{(k)}\|} \quad (1.141)$$

où $\|\mathbf{x}\|$ désigne la norme euclidienne.

On obtient les résultats suivants :

0.8944272	0.7808688	0.7327935	0.7157819	0.7100107	0.7080761	0.7074300
-0.4472136	-0.6246950	-0.6804511	-0.6983239	-0.7061361	-0.7067834	-0.7069999

On voit clairement sur cet exemple que la suite $\mathbf{x}^{(k)}$ converge vers $\mathbf{f}_2 = \frac{\sqrt{2}}{2} \begin{bmatrix} -1 \\ 1 \end{bmatrix}$ lorsque $k \rightarrow +\infty$. Si maintenant on fait tourner Scilab en lui demandant de calculer ensuite le produit scalaire de $A\mathbf{x}$ avec \mathbf{x} :

```
-->x= A*x; x=x/norm(x); mu=(A*x)' * x
```

ce qui correspond au calcul de la suite $\mu_k = A\mathbf{x}^{(k)} \cdot \mathbf{x}^{(k)}$, $k \geq 0$, on obtient la suite :

2.8, 2.9756098, 2.9972603, 2.9996952, 2.9999661, ...

qui a tout l'air de converger vers 3 ! En fait on a le théorème suivant, qui montre que dans un certain nombre de cas, on a effectivement convergence de l'algorithme vers la valeur propre dite dominante (celle qui correspond au rayon spectral).

Théorème 1.61 (Convergence de la méthode de la puissance). Soit A une matrice de $\mathcal{M}_n(\mathbb{C})$. On note $\lambda_1, \dots, \lambda_n$ les valeurs propres de A , $(\mathbf{f}_1, \dots, \mathbf{f}_n)$ une base orthonormée de trigonalisation de A telle que $A\mathbf{f}_n = \lambda_n\mathbf{f}_n$. On suppose que la valeur propre λ_n est dominante, c.à.d. que

$$|\lambda_n| > |\lambda_{n-1}| \geq \dots \geq |\lambda_1|,$$

et on suppose de plus que $\lambda_n \in \mathbb{R}$. Soit $\mathbf{x}^{(0)} \notin \text{Vect}(\mathbf{f}_1, \dots, \mathbf{f}_{n-1})$. Alors, la suite de vecteurs \mathbf{x}_{2k} définie par (1.141) converge vers un vecteur unitaire qui est vecteur propre de A pour la valeur propre dominante λ_n . De plus, si la norme choisie dans l'algorithme (1.141) est la norme 2, alors la suite $(A\mathbf{x}_k \cdot \mathbf{x}_k)_{k \in \mathbb{N}}$ converge vers λ_n lorsque $k \rightarrow +\infty$.

Démonstration. La démonstration de ce résultat fait l'objet de l'exercice 71 dans le cas plus simple où A est une matrice symétrique, et donc diagonalisable dans \mathbb{R} . \square

La méthode de la puissance souffre de plusieurs inconvénients :

1. Elle ne permet de calculer que la plus grande valeur propre. Or très souvent, on veut pouvoir calculer la plus petite valeur propre.
2. De plus, elle ne peut converger que si cette valeur propre est simple.
3. Enfin, même dans le cas où elle est simple, si le rapport des deux plus grandes valeurs propres est proche de 1, la méthode va converger trop lentement.

De manière assez miraculeuse, il existe un remède à chacun de ces maux :

1. Pour calculer plusieurs valeurs propres simultanément, on procède par blocs : on part de p vecteurs orthogonaux $\mathbf{x}_1^{(0)}, \dots, \mathbf{x}_p^{(0)}$ (au lieu d'un seul). Une itération de la méthode consiste alors à multiplier les p vecteurs par A et à les orthogonaliser par Gram-Schmidt. En répétant cette itération, on approche, si tout se passe bien, p valeurs propres et vecteurs propres de A , et la vitesse de convergence de la méthode est maintenant $\frac{\lambda_{n-p}}{\lambda_n}$.
2. Si l'on veut calculer la plus petite valeur propre, on applique la méthode de la puissance à A^{-1} . On a alors convergence (toujours si tout se passe bien) de $A^{-1}\mathbf{x}_k \cdot \mathbf{x}_k$ vers $1/|\lambda_1|$. Bien sûr, la mise en oeuvre effective ne s'effectue pas avec l'inverse de A , mais en effectuant une décomposition LU de A qui permet ensuite la résolution du système linéaire $A\tilde{\mathbf{x}}_{k+1} = \mathbf{x}^{(k)}$ (et $\mathbf{x}_{k+1} = \tilde{\mathbf{x}}_{k+1}/\|\tilde{\mathbf{x}}_{k+1}\|$).
3. Enfin, pour accélérer la convergence de la méthode, on utilise une translation sur A , qui permet de se rapprocher de la valeur propre que l'on veut effectivement calculer. Voir à ce propos l'exercice 72.

1.6.2 Méthode QR

Toute matrice A peut se décomposer sous la forme $A = QR$, où Q est une matrice orthogonale et R une matrice triangulaire supérieure. Dans le cas où A est inversible, cette décomposition est unique. On a donc le théorème suivant :

Théorème 1.62 (Décomposition QR d'une matrice). Soit $A \in \mathcal{M}_n(\mathbb{R})$. Alors il existe Q matrice orthogonale et R matrice triangulaire supérieure à coefficients diagonaux positifs ou nuls tels que $A = QR$. Si la matrice A est inversible, alors cette décomposition est unique.

La démonstration est effectuée dans le cas inversible dans la question 1 de l'exercice 75. La décomposition QR d'une matrice A inversible s'obtient de manière très simple par la méthode de Gram-Schmidt, qui permet de

construire une base orthonormée q_1, \dots, q_n (les colonnes de la matrice Q), à partir de n vecteurs vecteurs indépendants a_1, \dots, a_n (les colonnes de la matrice A). On se reportera à l'exercice 73 pour un éventuel rafraîchissement de mémoire sur Gram-Schmidt. Dans le cas où A n'est pas inversible (et même non carrée), la décomposition existe mais n'est pas unique. La démonstration dans le cadre général se trouve dans le livre de Ph. Ciarlet conseillé en début de ce cours.

L'algorithme QR pour la recherche des valeurs propres d'une matrice est extrêmement simple : Si A est une matrice inversible, on pose $A_0 = A$, on effectue la décomposition QR de A : $A = A_0 = Q_0 R_0$ et on calcule $A_1 = R_0 Q_0$. Comme le produit de matrices n'est pas commutatif, les matrices A_0 et A_1 ne sont pas égales, mais en revanche elles sont semblables ; en effet, grâce à l'associativité du produit matriciel, on a :

$$A_1 = R_0 Q_0 = (Q_0^{-1} Q_0) R_0 Q_0 = Q_0^{-1} (Q_0 R_0) Q_0 = Q_0^{-1} A Q_0.$$

Les matrices A_0 et A_1 ont donc même valeurs propres.

On recommence alors l'opération : à l'itération k , on effectue la décomposition QR de A_k : $A_k = Q_k R_k$ et on calcule $A_{k+1} = R_k Q_k$.

Par miracle, pour la plupart des matrices, les coefficients diagonaux de la matrice A_k tendent vers les valeurs propres de la matrice A , et les colonnes de la matrice Q_k vers les vecteurs propres associés. On sait démontrer cette convergence pour certaines matrices ; on pourra trouver par exemple dans les livres de Serre ou Hubbard-Hubert la démonstration sous une hypothèse assez technique et difficile à vérifier en pratique ; l'exercice 75 donne la démonstration (avec la même hypothèse technique) pour le cas plus simple d'une matrice symétrique définie positive.

Pour améliorer la convergence de l'algorithme QR , on utilise souvent la technique dite de "shift" (translation en français). A l'itération n , au lieu d'effectuer la décomposition QR de la matrice A_n , on travaille sur la matrice $A_n - bI$, où b est choisi proche de la plus grande valeur propre. En général on choisit le coefficient $b = a_{nn}^{(k)}$. L'exercice 74 donne un exemple de l'application de la méthode QR avec shift.