

# TP 2: Équations linéaires: décomposition QR

Philippe Després

Date de remise: 2 mars 2018

---

PHY-3500 – Physique numérique (H18)

Professeur : Philippe Després

---

## Algorithme QR

Dans ce devoir, vous allez développer un algorithme permettant de calculer les vecteurs et valeurs propres d'une matrice réelle symétrique à partir de la décomposition QR.

Soit une matrice carrée et réelle  $\mathbf{A}$  que l'on souhaite exprimer comme

$$\mathbf{A} = \mathbf{Q}\mathbf{R} \tag{1}$$

où  $\mathbf{Q}$  est une matrice orthogonale et  $\mathbf{R}$  une matrice triangulaire supérieure. Pour y arriver, imaginons la matrice  $\mathbf{A}$  de dimensions  $N \times N$  comme un ensemble de  $N$  vecteurs colonnes  $\mathbf{a}_0 \dots \mathbf{a}_{N-1}$

$$\begin{pmatrix} | & | & | & \dots \\ \mathbf{a}_0 & \mathbf{a}_1 & \mathbf{a}_2 & \dots \\ | & | & | & \dots \end{pmatrix} \tag{2}$$

où la numérotation des vecteurs débute à 0 pour reproduire l'indexation en Python, ce qui sera utile pour écrire le code.

Définissons maintenant deux nouveaux ensembles de vecteurs  $\mathbf{u}_0 \dots \mathbf{u}_{N-1}$  et  $\mathbf{q}_0 \dots \mathbf{q}_{N-1}$  tels que

$$\begin{aligned}\mathbf{u}_0 &= \mathbf{a}_0, & \mathbf{q}_0 &= \frac{\mathbf{u}_0}{|\mathbf{u}_0|}, \\ \mathbf{u}_1 &= \mathbf{a}_1 - (\mathbf{q}_0 \cdot \mathbf{a}_1)\mathbf{q}_0, & \mathbf{q}_1 &= \frac{\mathbf{u}_1}{|\mathbf{u}_1|}, \\ \mathbf{u}_2 &= \mathbf{a}_2 - (\mathbf{q}_0 \cdot \mathbf{a}_2)\mathbf{q}_0 - (\mathbf{q}_1 \cdot \mathbf{a}_2)\mathbf{q}_1, & \mathbf{q}_2 &= \frac{\mathbf{u}_2}{|\mathbf{u}_2|},\end{aligned}$$

et ainsi de suite. Les formules générales pour calculer  $\mathbf{u}_i$  et  $\mathbf{q}_i$  sont

$$\mathbf{u}_i = \mathbf{a}_i - \sum_{j=0}^{i-1} (\mathbf{q}_j \cdot \mathbf{a}_i) \mathbf{q}_j, \quad \mathbf{q}_i = \frac{\mathbf{u}_i}{|\mathbf{u}_i|} \quad (3)$$

a. Démontrez, par induction ou autrement, que les vecteurs  $\mathbf{q}_i$  sont orthonormaux.

En réarrangeant la définition des vecteurs, on a

$$\begin{aligned}\mathbf{a}_0 &= |\mathbf{u}_0| \mathbf{q}_0 \\ \mathbf{a}_1 &= |\mathbf{u}_1| \mathbf{q}_1 + (\mathbf{q}_0 \cdot \mathbf{a}_1) \mathbf{q}_0 \\ \mathbf{a}_2 &= |\mathbf{u}_2| \mathbf{q}_2 + (\mathbf{q}_0 \cdot \mathbf{a}_2) \mathbf{q}_0 + (\mathbf{q}_1 \cdot \mathbf{a}_2) \mathbf{q}_1\end{aligned} \quad (4)$$

et ainsi de suite. Les vecteurs  $\mathbf{q}_i$  peuvent être groupés pour former les colonnes d'une matrice qui permet d'exprimer les équations précédentes comme une seule opération matricielle :

$$\mathbf{A} = \begin{pmatrix} | & | & | & \dots \\ \mathbf{a}_0 & \mathbf{a}_1 & \mathbf{a}_2 & \dots \\ | & | & | & \dots \end{pmatrix} = \begin{pmatrix} | & | & | & \dots \\ \mathbf{q}_0 & \mathbf{q}_1 & \mathbf{q}_2 & \dots \\ | & | & | & \dots \end{pmatrix} \begin{pmatrix} |\mathbf{u}_0| & \mathbf{q}_0 \cdot \mathbf{a}_1 & \mathbf{q}_0 \cdot \mathbf{a}_2 & \dots \\ 0 & |\mathbf{u}_1| & \mathbf{q}_1 \cdot \mathbf{a}_2 & \dots \\ 0 & 0 & |\mathbf{u}_2| & \dots \end{pmatrix}. \quad (5)$$

Vous pouvez vous en convaincre en effectuant la multiplication matricielle à droite pour retrouver les vecteurs  $\mathbf{a}_i$ .

Sous cette forme, la matrice contenant les vecteurs  $\mathbf{q}_i$  est orthogonale car il a été démontré à la question a que les  $\mathbf{q}_i$  sont orthonormaux. De toute évidence, la matrice multipliant celles des  $\mathbf{q}_i$  plus haut est triangulaire supérieure. Nous avons donc trouvé la décomposition QR de  $\mathbf{A}$ , soit

$$\mathbf{Q} = \begin{pmatrix} | & | & | & \dots \\ \mathbf{q}_0 & \mathbf{q}_1 & \mathbf{q}_2 & \dots \\ | & | & | & \dots \end{pmatrix}, \quad \mathbf{R} = \begin{pmatrix} |\mathbf{u}_0| & \mathbf{q}_0 \cdot \mathbf{a}_1 & \mathbf{q}_0 \cdot \mathbf{a}_2 & \dots \\ 0 & |\mathbf{u}_1| & \mathbf{q}_1 \cdot \mathbf{a}_2 & \dots \\ 0 & 0 & |\mathbf{u}_2| & \dots \end{pmatrix}. \quad (6)$$

- b. Écrivez un programme Python prenant en argument une matrice carrée réelle  $\mathbf{A}$  et qui retourne deux matrices  $\mathbf{Q}$  et  $\mathbf{R}$  telles que définies plus haut. À titre de test, obtenez la décomposition QR de

$$\begin{pmatrix} 1 & 4 & 8 & 4 \\ 4 & 2 & 3 & 7 \\ 8 & 3 & 6 & 9 \\ 4 & 7 & 9 & 2 \end{pmatrix} \quad (7)$$

et assurez-vous du résultat en vérifiant que le produit de  $\mathbf{Q}$  et de  $\mathbf{R}$  donne bien  $\mathbf{A}$ .

- c. À l'aide de cette fonction, écrivez un programme permettant de calculer les vecteurs et valeurs propres d'une matrice carrée réelle (voir notes de cours ou le Newman pour l'algorithme). La nouvelle fonction prendra un deuxième argument, soit la valeur  $\epsilon$  que devra atteindre tous les éléments non-diagonaux (i.e. tous plus petits que  $\epsilon$  en valeur absolue). Testez votre programme sur la matrice précédente et une valeur  $\epsilon = 10^{-6}$ . Que se passe-t-il avec  $\epsilon = 10^{-12}$  ou  $\epsilon = 10^{-18}$  ?

## Puits quantique asymétrique

Certains problèmes en mécanique quantique peuvent être formulés de façon matricielle et résolus numériquement. Supposons par exemple une particule de masse  $M$  dans un puits de potentiel unidimensionnel de largeur  $L$ . Au lieu de considérer une forme carrée, supposons plutôt que le potentiel dans le puits varie selon  $V(x)$ . Ce type de problème ne peut pas, en général, être résolu de façon analytique.

Pour un état d'énergie  $E$ , la composante spatiale de la fonction d'onde obéit à l'équation de Schrödinger indépendante du temps  $\hat{H}\psi(x) = E\psi(x)$  où l'opérateur Hamiltonien  $\hat{H}$  est donné par

$$\hat{H} = -\frac{\hbar^2}{2M} \frac{d^2}{dx^2} + V(x). \quad (8)$$

Considérons pour faire plus simple que les murs du puits sont infiniment hauts, ce qui fait que la fonction d'onde est nulle à l'extérieur du puits et qu'elle doit être nulle aussi à  $x = 0$  et  $x = L$ . Dans ce cas, la fonction d'onde peut être exprimée par une série de Fourier telle que

$$\psi(x) = \sum_{n=1}^{\infty} \psi_n \sin \frac{\pi n x}{L}, \quad (9)$$

où  $\psi_1, \psi_2, \dots$  sont les coefficients de Fourier.

- d. Considérant que pour des entiers  $n, m$  positifs

$$\int_0^L \sin \frac{\pi m x}{L} \sin \frac{\pi n x}{L} dx = \begin{cases} L/2 & \text{si } m = n \\ 0 & \text{autrement,} \end{cases} \quad (10)$$

démontrez que l'équation de Schrödinger  $\hat{H}\psi(x) = E\psi(x)$  implique que

$$\sum_{n=1}^{\infty} \psi_n \int_0^L \sin \frac{\pi m x}{L} \hat{H} \sin \frac{\pi n x}{L} dx = \frac{1}{2} L E \psi_m. \quad (11)$$

Donc, en définissant une matrice  $\mathbf{H}$  ayant les éléments

$$H_{mn} = \frac{2}{L} \int_0^L \sin \frac{\pi m x}{L} \hat{H} \sin \frac{\pi n x}{L} dx \quad (12)$$

$$= \frac{2}{L} \int_0^L \sin \frac{\pi m x}{L} \left[ -\frac{\hbar^2}{2M} \frac{d^2}{dx^2} + V(x) \right] \sin \frac{\pi n x}{L} dx, \quad (13)$$

démontrez que l'équation de Schrödinger peut être écrite sous forme matricielle comme  $\mathbf{H}\boldsymbol{\psi} = E\boldsymbol{\psi}$  où  $\boldsymbol{\psi}$  est le vecteur  $(\psi_1, \psi_2, \dots)$ .

Autrement dit,  $\boldsymbol{\psi}$  est un vecteur propre de la matrice Hamiltonienne  $\mathbf{H}$  avec la valeur propre associée  $E$ . En calculant les vecteurs propres de cette matrice, nous retrouvons les états d'énergie permis dans le puits.

- e. Pour le cas  $V(x) = ax/L$ , évaluez analytiquement l'intégrale dans  $H_{mn}$  pour trouver une expression générale des éléments de la matrice. Démontrez que cette matrice est réelle et symétrique. Il pourrait être intéressant de savoir que

$$\int_0^L x \sin \frac{\pi m x}{L} \sin \frac{\pi n x}{L} dx = \begin{cases} 0 & \text{si } m \neq n \text{ et } m, n \text{ sont pairs ou impairs} \\ -\left(\frac{2L}{\pi}\right)^2 \frac{mn}{(m^2 - n^2)^2} & \text{si } m \neq n \text{ si un est pair et l'autre impair} \\ L^2/4 & \text{si } m = n. \end{cases} \quad (14)$$

Écrivez un programme Python pour évaluer les  $H_{mn}$  pour des valeurs arbitraires de  $m$  et de  $n$ , pour un électron dans un puits de largeur  $5 \text{ \AA}$  caractérisé par  $a = 10 \text{ eV}$ . La masse et la charge de l'électron sont respectivement  $9.1094 \times 10^{-31} \text{ kg}$  et  $1.6022 \times 10^{-19} \text{ C}$ .

- f. La matrice  $\mathbf{H}$  est en théorie infiniment grande et il n'est pas possible de calculer toutes ses valeurs propres. Il est cependant possible d'obtenir une solution relativement juste pour les premiers niveaux en considérant seulement ses premiers éléments. Modifiez le programme écrit pour la question e pour créer la matrice  $\mathbf{H}$  jusqu'à  $m = n = 10$  et calculez ses valeurs propres, i.e. les dix premiers niveaux d'énergie que vous exprimerez en eV. Comparez votre fonction servant à calculer les valeurs propres, basée sur la décomposition QR (question c), avec la fonction appropriée de `numpy.linalg` et notez vos observations. Vous devriez trouver que l'état fondamental correspond approximativement à une énergie de  $5.84 \text{ eV}$ .

- g. Modifiez maintenant votre programme pour utiliser une matrice de  $100 \times 100$  et comparez les dix premiers niveaux d'énergie à ceux obtenus à la question précédente avec une matrice  $100 \times 100$ . Que pouvez-vous conclure par rapport à l'exactitude des résultats ?
- h. Améliorez votre programme pour qu'il puisse calculer la fonction d'onde  $\psi(x)$  de l'état fondamental ainsi que celles des deux premiers états excités. Faites un graphique des densités de probabilités  $|\psi(x)|^2$  associées à ces trois états, en prenant bien soin de normaliser la fonction d'onde, i.e. respecter cette condition :  $\int_0^L |\psi(x)|^2 dx = 1$ . Est-ce bien le cas avec vos fonctions d'onde ?

## Instructions pour la remise

Le travail devra être complété individuellement ou en binôme sous format de cahier de bord IPython/jupyter (.ipynb) et remis dans la boîte de dépôt ENA créée à cette fin. Ce document contiendra **toutes informations pertinentes** permettant au lecteur d'apprécier vos résultats et conclusions, incluant le code Python utilisé et d'éventuelles références bibliographiques. La qualité de la présentation est très importante (utilisation de sections, de graphiques appropriés, de mise en contexte, etc.).

Prenez soin de bien indiquer votre (ou vos) nom(s) dans le cahier de bord. Pour faciliter la tâche de classification, utilisez la nomenclature suivante pour le fichier transmis (un seul) :

TPn\_nom1.ipnb ou TPn\_nom1\_nom2.ipnb