2TSI. DEVOIR LIBRE N°08

Inspiré fortement du sujet CCS Math 2 TSI 2020 et concaténé avec du Python

À rendre le 01 Avril 2021

Procédé de Gram-Schmidt et décomposition QR d'une matrice inversible

La transposée d'une matrice $M \in \mathcal{M}_n(\mathbf{R})$ se note M^T. Une matrice $M \in \mathcal{M}_n(\mathbf{R})$ est triangulaire supérieure lorsque $a_{i,j} = 0$ dès que $1 \leqslant j < i \leqslant n$ et on rappelle que le produit de deux matrices triangulaires supérieures et l'inverse d'une matrice triangulaire supérieure inversible est encore une matrice triangulaire supérieure.

Dans tout le problème, on note $A = \begin{pmatrix} 1 & -3 & 4 \\ 2 & 0 & -1 \\ -2 & 3 & 1 \end{pmatrix}$.

Partie I

Algorithme de Gram-Schmidt

On note c_1, c_2, c_3 les colonnes de A considérées comme des vecteurs de $\mathbf{R}^3$.

1. Justifier que A est inversible. En déduire que $\mathcal{B}' = (c_1, c_2, c_3)$ est une base de $\mathbf{R}^3$.
2. On rappelle le procédé d'orthonormalisation de Gram-Schmidt :
   Soit F un sous-espace de dimension finie de E et $(\vec{u}_i)_{1\leqslant i\leqslant n}$ une base de F. On pose $\vec{v}_1 = \vec{u}_1$ et pour k variant de 1 à $n-1$:

   $$\boxed{\vec{v}_{k+1} = \vec{u}_{k+1} - \sum_{i=1}^{k} \frac{\langle \vec{u}_{k+1}, \vec{v}_i\rangle}{\|\vec{v}_i\|^2}\vec{v}_i}.$$

   Alors la famille $(\vec{v}_i)_{1\leqslant i\leqslant n}$ est une base orthogonale *de F adaptée aux espaces $U_p = \mathsf{Vect}\left((\vec{u}_i)_{1\leqslant i\leqslant p}\right)$, c'est-à-dire que $\mathsf{Vect}\left((\vec{v}_i)_{1\leqslant i\leqslant p}\right) = \mathsf{Vect}\left((\vec{u}_i)_{1\leqslant i\leqslant p}\right)$.*
   Pour obtenir une base orthonormale *de F, il suffit de poser ensuite : $\vec{w}_i = \dfrac{1}{\|\vec{v}_i\|}\vec{v}_i$, pour $i \in [\![1, n]\!]$.*
   Ici, l'on pose $n = 3$, $F = \mathbf{R}^3$ et pour tout $i \in [\![1, 3]\!]$, $\vec{u}_i = c_i$.
   Déterminer une base orthonormale $\mathcal{B}'' = (\vec{w}_1, \vec{w}_2, \vec{w}_3)$ par le procédé d'orthonormalisation de Gram-Schmidt rappelé plus haut appliqué à $\mathcal{B}'$.
3. On importe *numpy* avec l'alias *np* puis on considère la fonction PYTHON suivante, appliquée à une matrice $P \in \mathrm{GL}_n(\mathbf{R})$.

```
>>> def UNKNOW(P) :
        m = np.size(P,0) ; n = np.size(P,1) ; W = np.zeros((m,n))
        W[:,0] = P[:,0]
        for k in range(1,n) :
            W[:,k] = P[:,k] - sum(np.vdot(P[:,k],W[:,i]) * W[:,i]
            /np.vdot(W[:,i],W[:,i]) for i in range(0,k))
        for k in range(0,n) :
            W[:,k] = W[:,k]/np.sqrt(np.vdot(W[:,k],W[:,k]))
        return(W)
```

   Que fait $UNKNOW$? Appliquer alors $UNKNOW$ avec la matrice A du sujet. Que retrouve t-on ?

T.S.V.P →

Partie II

Décomposition QR

On reprend les notations de la partie I. On note $T_n^+(\mathbf{R})$ l'ensemble des matrices triangulaires supérieures à coefficients diagonaux strictement positifs et $\mathrm{O}_n(\mathbf{R})$ l'ensemble des matrices orthogonales carrées d'ordre n.

1. **L'exemple de la matrice A.**
   (a) Soit Q la matrice de passage de la base canonique $\mathcal{B}$ de $\mathbf{R}^3$ à la base $\mathcal{B}''$ (Q a donc pour colonnes les vecteurs $\vec{w}_i$ pour $i \in [\![1,3]\!]$). Justifier que Q est une matrice orthogonale et que $Q^{-1} = Q^T$.
   (b) Reconnaître l'endomorphisme associé à Q et déterminer ses éléments caractéristiques.
   (c) Écrire la matrice de passage R de la base $\mathcal{B}''$ à la base $\mathcal{B}'$.
   On constatera que $R \in T_3^+(\mathbf{R})$.
   (d) Montrer que $A = QR$.
2. **Cas général**. Soit $P \in \mathcal{M}_n(\mathbf{R})$ une matrice inversible (donc $P \in \mathrm{GL}_n(\mathbf{R})$), on veut montrer dans cette question : $\exists\,!\,(Q,R) \in \mathrm{O}_n(\mathbf{R}) \times T_n^+(\mathbf{R}), P = QR$.
   On s'inspire du cas particulier $P = A$.
   (a) On se place dans le cadre de $\mathbf{R}^n$ euclidien, rapporté à $\mathcal{B}$ sa base canonique. Si $\mathcal{B}' = \{\vec{u}_1, ..., \vec{u}_n\}$ est la famille des vecteurs colonnes de P, et si $\mathcal{B}'' = \{\vec{w}_1, ...\vec{w}_n\}$, la famille orthonormée obtenue à partir de $\mathcal{B}'$ par le procédé de Gram-Schmidt alors si l'on pose Q la matrice de passage (orthogonale) de $\mathcal{B}$ à $\mathcal{B}''$ et R la matrice de passage de $\mathcal{B}''$ à $\mathcal{B}'$, montrer que (Q,R) répond à la question, c'est-à-dire que $P = QR$.
   (b) Soit M une matrice de $\mathcal{M}_n(\mathbf{R})$ et on suppose que $M \in \mathcal{O}_n(\mathbf{R}) \cap T_n^+(\mathbf{R})$.
   Montrer que $M = I_n$.
   (c) On considère les matrices Q_1, Q_2, R_1, R_2 de $\mathcal{M}_n(\mathbf{R})$ telles que pour tout $i \in [\![1,2]\!]$, $Q_i \in \mathcal{O}_n(\mathbf{R})$ et $R_i \in T_n^+(\mathbf{R})$ avec $Q_1R_1 = Q_2R_2$.
   Montrer que $Q_1 = Q_2$ et $R_1 = R_2$.
3. **Mise en Python de l'algorithme de décomposition QR.**
   Créer une fonction Python nommée *Decomp_QR* d'argument une matrice P et qui retourne la liste $[Q,R]$ composée de l'unique matrice $Q \in \mathrm{O}_n(\mathbf{R})$ et de l'unique matrice $R \in T_n^+(\mathbf{R})$ telles que $P = QR$.
   Indication : on remarquera que $P = QR \Rightarrow R = Q^TP$ et que la commande np.dot(np.transpose(Q), P) retourne Q^TP. De plus, on utilisera dans notre procédure la fonction UNKNOW de la partie I.
   Appliquer ensuite avec $P = \begin{pmatrix} 12 & -51 & 4 \\ 6 & 167 & -68 \\ -4 & 24 & -41 \end{pmatrix}$ et à la matrice A de l'énoncé.
4. **Application de la décomposition QR à la résolution d'un système linéaire**.
   (a) On considère le système $(\mathcal{S})$: $Ax = \begin{pmatrix} 1 \\ -1 \\ 2 \end{pmatrix}$. Montrer que si R est la matrice de la décomposition $A = QR$ alors le système $(\mathcal{S})$ se ramène à $Rx = y$, où y est à expliciter. Résoudre alors ce système.
   (b) Soit b un vecteur de $\mathbf{R}^n$ et P une matrice inversible de $\mathcal{M}_n(\mathbf{R})$. Expliquer l'intérêt de la décomposition $P = QR$, avec $Q \in \mathcal{O}_n(\mathbf{R})$ et $R \in T_n^+(\mathbf{R})$, pour résoudre le système linéaire $Px = b$, d'inconnue $x \in \mathbf{R}^n$.