Correction. Devoir libre 08

Partie I

Algorithme de Gram-Schmidt

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

Un développement de $\mathsf{Det}\,(A)$ suivant sa $2^{\text{ème}}$ colonne permet de voir que

$$\mathsf{Det}\,(A) = -(-3)\begin{vmatrix} 2 & -1 \\ -2 & 1 \end{vmatrix} - 3\begin{vmatrix} 1 & 4 \\ 2 & -1 \end{vmatrix} = 3\cdot 0 - 3\cdot(-9) = 27 \neq 0,$$

ce qui prouve bien que la matrice A est inversible.
On en déduit que $\mathcal{B}' = (c_1, c_2, c_3)$ est une base de $\mathbf{R}^3$ car la suite (c_1, c_2, c_3) des colonnes de A est libre dans un espace vectoriel de dimension 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 $U_p = \mathsf{Vect}\,((\vec{u}_i)_{1\leqslant i\leqslant p})$, c'est-à-dire que $\mathsf{Vect}\,((\vec{v}_i)_{1\leqslant i\leqslant p}) = \mathsf{Vect}\,((\vec{u}_i)_{1\leqslant i\leqslant p})$.*
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 $i \in [\![1,3]\!]$, $\vec{u}_i = c_i$. Déterminons une base orthonormale $\mathcal{B}'' = (\vec{w}_1, \vec{w}_2, \vec{w}_3)$ par le procédé d'orthonormalisation de Gram-Schmidt appliqué à $\mathcal{B}'$.

beginitemize

$$w_1 = \frac{1}{\|u_1\|}\cdot c_1 = \frac{1}{3}\cdot u_1 = \begin{pmatrix} 1/3 \\ 2/3 \\ -2/3 \end{pmatrix}$$

$$v_2 = u_2 - \langle u_2; w_1\rangle \cdot w_1 = \begin{pmatrix} -3 \\ 0 \\ 3 \end{pmatrix} - (-3)\cdot\begin{pmatrix} 1/3 \\ 2/3 \\ -2/3 \end{pmatrix} = \begin{pmatrix} -2 \\ 2 \\ 1 \end{pmatrix},$$

$$w_2 = \frac{1}{\|v_2\|}\cdot v_2 = \begin{pmatrix} -2/3 \\ 2/3 \\ 1/3 \end{pmatrix}$$

$$v_3 = u_3 - \langle u_3; w_1\rangle\cdot w_1 - \langle u_3; w_2\rangle \cdot w_2 = u_3 - 0\cdot w_1 - (-3)\cdot w_2 = \begin{pmatrix} 4 \\ -1 \\ 1 \end{pmatrix} + \begin{pmatrix} -2 \\ 2 \\ 1 \end{pmatrix} = \begin{pmatrix} 2 \\ 1 \\ 2 \end{pmatrix},$$

$$w_3 = \frac{1}{\|v_3\|}\cdot v_3 = \begin{pmatrix} 2/3 \\ 1/3 \\ 2/3 \end{pmatrix}$$

Conclusion : la famille $(w_1; w_2; w_3)$ donnée par

$$\boxed{w_1 = \begin{pmatrix} 1/3 \\ 2/3 \\ -2/3 \end{pmatrix},\ w_2 = \begin{pmatrix} -2/3 \\ 2/3 \\ 1/3 \end{pmatrix},\ w_3 = \begin{pmatrix} 2/3 \\ 1/3 \\ 2/3 \end{pmatrix}}$$

forme une base orthonormée $\mathcal{B}''$ dans $\mathbb{R}^3$.

On importe *numpy* avec l'alias *np* puis on considère la fonction PYTHON suivante, appliquée à une matrice A.

```
>>> import numpy as np
>>> def UNKNOW(A) :
        m = np.size(A,0) ; n = np.size(A,1) ; W = np.zeros((m,n))
        W[:,0] = A[:,0]
        for k in range(1,n) :
            W[:,k] = A[:,k] - sum(np.vdot(A[:,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)
```

$UNKNOW$ renvoie la liste des coordonnées des vecteurs de la base orthonormale par Gram-Schmidt issue des vecteurs colonnes de A. Appliquons alors $UNKNOW$ avec la matrice A du sujet.

```
>>> A = np.array([[1,-3,4],[2,0,-1],[-2,3,1]])
>>> UNKNOW(A)
```

```
array([[0.333333333, -0,6666666667, 0.666666667],
[0.666666667, 0.666666667, 0.333333333]
[−0.666666667, 0.333333333, 0.666666667]])
```

On récupère les colonnes de cette matrice.

$$\vec{w}_1 = \frac{1}{3}(1,2,-2),\ \vec{w}_2 = \frac{1}{3}(-2,2,1) \text{ et } \vec{w}_3 = \frac{1}{3}(2,1,2).$$

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.

1-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 $i \in [\![1,3]\!]$). Justifions que Q est une matrice orthogonale et que $Q^{-1} = Q^T$.

Tant la base canonique $\mathcal{B}$ que la nouvelle base $\mathcal{B}''$ constituent des bases orthonormées pour le produit scalaire standard dans $\mathbb{R}^3$; donc les matrices de passage $Q = \mathrm{coord}_{\mathcal{B}}(\mathcal{B}'')$ et $Q^{-1} = \mathrm{coord}_{\mathcal{B}''}(\mathcal{B})$ sont *orthogonales*, ce qui revient à dire que $\boxed{Q^{-1} = Q^T}$.

1-b Reconnaîssons l'endomorphisme associé à Q et déterminons ses éléments caractéristiques.

On voit que $w_1 \wedge w_2 = w_3$ et donc Q représente une rotation vectorielle r. De plus, les vecteurs invariants sont donnés par :

$$QX = X \Leftrightarrow \frac{1}{3}\begin{pmatrix} 1 & -2 & 2 \\ 2 & 2 & 1 \\ -2 & 1 & 2 \end{pmatrix}\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} x \\ y \\ z \end{pmatrix} \Leftrightarrow \begin{cases} x - 2y + 2z &= 3x \\ 2x + 2y + z &= 3y \\ -2x + y + 2z &= 3z \end{cases} \Leftrightarrow \begin{cases} x = 0 \\ y = z \end{cases}$$

Et donc $E_1(Q) = \mathsf{Vect}\,((0, 1/\sqrt{2}, 1/\sqrt{2})$. Puis $2\cos\theta + 1 = 5/3 \Rightarrow \theta = \pm \arccos 1/3$.
Le signe de θ se trouve en prenant par exemple $\vec{a}(1,0,0)$ et

$$\frac{1}{\sqrt{2}}.\frac{1}{3}\,\mathsf{Det}\left(\begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 2 \\ -2 \end{pmatrix}\right) = \frac{-4}{3\sqrt{2}} = \sin\theta < 0.$$

Donc r est la rotation d'axe dirigée et orientée par $\dfrac{1}{\sqrt{2}}\begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}$ et d'angle $\theta = -\arccos(1/3)$.

1-c Écrivons la matrice de passage R de la base $\mathcal{B}''$ à la base $\mathcal{B}'$.
On constatera que $R \in T_3^+(\mathbf{R})$.

On a : $R = \text{coord}_{\mathcal{B}''(\mathcal{B}')}$, et d'après les calculs effectués plus haut :

- $w_1 = \dfrac{1}{3} \cdot c_1$, donc $c_1 = 3 \cdot w_1$,
- $w_2 = \dfrac{1}{3} \cdot (c_2 + c_1)$, donc $c_2 = 3 \cdot w_2 - 3 \cdot w_1$,
- $w_3 = \dfrac{1}{3} \cdot (c_3 + 3 \cdot w_2)$, donc $c_3 = 3 \cdot w_3 - 3 \cdot w_2$,

ainsi

$$\boxed{R = \text{coord}_{\mathcal{B}''}(\mathcal{B}') = \text{coord}_{(w_1,w_2,w_3)}(c_1,c_2,c_3) = \begin{pmatrix} 3 & -3 & 0 \\ 0 & 3 & -3 \\ 0 & 0 & 3 \end{pmatrix}}$$

1-d Montrons que $A = QR$.

On fait : $QR = \dfrac{1}{3}\begin{pmatrix} 1 & -2 & 2 \\ 2 & 2 & 1 \\ -2 & 1 & 2 \end{pmatrix} \times \begin{pmatrix} 3 & -3 & 0 \\ 0 & 3 & -3 \\ 0 & 0 & 3 \end{pmatrix} = \begin{pmatrix} 1 & -3 & 4 \\ 2 & 0 & -1 \\ -2 & 3 & 1 \end{pmatrix}$.

2. Cas général. Soit $P \in \mathcal{M}_n(\mathbf{R})$ une matrice inversible (donc $P \in \text{GL}_n(\mathbf{R})$), on veut montrer dans cette question : $\exists\,!\,(Q,R) \in \text{O}_n(\mathbf{R}) \times T_n^+(\mathbf{R}), P = QR$.
On s'inspire du cas particulier $P = A$.

2-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 $\mathcal{B}'$, montrons que (Q,R) répond à la question, c'est-à-dire que $P = QR$.

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 A, on considère A comme la matrice de $Id_{\mathbf{R}^n}$, la base de départ est $\mathcal{B}'$ et la base d'arrivée est $\mathcal{B}$. Alors $A = M_{\mathcal{B}',\mathcal{B}}(Id)$. Introduisons $\mathcal{B}'' = \{\vec{w}_1, ...\vec{w}_n\}$, la famille orthonormée obtenue à partir de $\mathcal{B}'$ par le procédé de Gram-Schmidt. Alors comme, d'après Gram-Schmidt, $\vec{w}_i \in \mathsf{Vect}\,(\vec{u}_1, ..., \vec{u}_i)$ pour tout i, la matrice de passage de $\mathcal{B}'$ à $\mathcal{B}''$, notée $P_{\mathcal{B}'}^{\mathcal{B}''}$ est triangulaire supérieure et ses coefficients diagonaux sont strictement positifs (et il en est de même de son inverse $P_{\mathcal{B}''}^{\mathcal{B}'}$). En effet, on remarque avec la **proposition ??** que la matrice de passage de la base $\mathcal{B}' = \{\vec{u}_1, ..., \vec{u}_n\}$ à la base orthogonale $\{\vec{v}_1, ..., \vec{v}_n\}$ est triangulaire supérieure de coefficients diagonaux tous égaux à 1. Si l'on normalise la base orthogonale, la matrice a pour coefficients diagonaux $\{\frac{1}{\|\vec{v}_1\|}, ..., \frac{1}{\|\vec{v}_n\|}\}$.
C'est $P_{\mathcal{B}'}^{\mathcal{B}''}$, de coefficients diagonaux > 0. Comme son inverse $P_{\mathcal{B}''}^{\mathcal{B}'}$.
Il reste à utiliser la formule qui permet de transformer une matrice d'une application linéaire (ici $Id_{\mathbf{R}^n}$), la base de départ étant elle aussi transformée. On rappelle la formule revue dans le chapitre sur les matrices et applications linéaires :

$$M_{\mathcal{B}',\mathcal{B}}(\phi) = \left(P_{\mathcal{B}}^{\mathcal{B}}\right)^{-1} M_{\mathcal{B}'',\mathcal{B}}(\phi) P_{\mathcal{B}''}^{\mathcal{B}'}.$$

Comme $P_{\mathcal{B}}^{\mathcal{B}} = I_n$, en posant $R = P_{\mathcal{B}''}^{\mathcal{B}'}$ et $Q = M_{\mathcal{B}'',\mathcal{B}}(\phi)$, on a bien le résultat voulu car Q est la matrice de passage d'une base orthonormée sur une autre, c'est donc une matrice orthogonale.

2-b Soit M une matrice de $\mathcal{M}_n(\mathbf{R})$ et on suppose que $M \in \mathcal{O}_n(\mathbf{R}) \cap T_n^+(\mathbf{R})$. Montrons que $M = I_n$.

On observe que $M^{-1} = M^T$ est encore triangulaire supérieure, et comme M^T est triangulaire inférieure, elle est diagonale, en sorte que M est à la fois orthogonale et diagonale ; mais alors, chacun des coeff. diagonaux de M se doit d'être de la forme $m_{i,i} = \pm 1$, et comme par ailleurs M est à coeff. diagonaux > 0, on obtient bien $M = I_n$.

2-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 $Q_1R_1 = Q_2R_2$.
Montrons que $Q_1 = Q_2$ et $R_1 = R_2$.

Les matrices triangulaires R_1 et R_2 ayant chacune un déterminant strictement positif, elles sont toutes deux inversibles ; l'identité $Q_1R_1 = Q_2R_2$ donne alors successivement $Q_1 = Q_2R_2R_1^{-1}$ puis $Q_2^{-1}Q_1 = R_2R_1^{-1}$. Mais $Q_2^{-1}Q_1$ est une matrice orthogonale (comme produit de matrices orthogonales), tandis que $R_2R_1^{-1}$ est triangulaire supérieure et à coefficients diagonaux strictement positifs, comme produit de 2 telles matrices. D'après plus haut, il s'ensuit que $Q_2^{-1}Q_1 = R_2R_1^{-1} = I_n$, ce qui donne bien $Q_2 = Q_1$ et $R_2 = R_1$.

3. Mise en Python de l'algorithme de décomposition QR.
Créons une fonction Python nommée *Decomp_QR* d'argument une matrice P et qui retourne la liste $[Q,R]$ composée de l'unique matrice $Q \in \text{O}_n(\mathbf{R})$ et de l'unique matrice $R \in T_n^+(\mathbf{R})$ telles que $P = QR$.

Indication : on remarquera que $P = QR \Rightarrow R = Q^T P$ et que la commande $np.dot(np.transpose(Q), P)$ retourne $Q^T P$. De plus, on utilisera dans notre procédure la fonction $UNKNOW$ de la partie I.

On remarque que $Q^{-1} = Q^T$. Ainsi, pour déterminer la décomposition QR de A, on détermine Q par Gram-Schmidt (car Q est la matrice W de la procédure $UNKNOW$ de la partie I) et quant à R, on utilise :
$A = QR \Rightarrow Q^T A = Q^T QR = R$. Il suffit donc de faire le produit de la transposée de Q par A.

On tape :

```
>>> def Decomp_QR(A) :
        W = UNKNOW(A) ; R = np.dot(np.transpose(W), A)
        L = [W, R] ; return(L)
```

Appliquons ensuite avec $P = \begin{pmatrix} 12 & -51 & 4 \\ 6 & 167 & -68 \\ -4 & 24 & -41 \end{pmatrix}$ et à la matrice A de l'énoncé.

```
>>> Decomp_QR(np.array([[12, −51, 4], [6, 167, −68], [−4, 24, −41]]))
```

```
[array([[0.85714286 , −0.39428571 , −0.33142857] , [0.42857143 , 0.90285714 , 0.03428571] ,
    [−0.28571429 , 0.17142857 , −0.94285714]]) , array([[1.4000e + 01 , 2.1000e + 01 , −1.4000e + 01 ],
    [−6.66133815e − 16 , 1.75000e + 02 , −7.000e + 01] , [0.0000e + 00 , 1.42108547e − 14 , 3.5000e + 01]])]
```

De même :

```
>>> Decomp_QR(np.array([[1, −3, 4], [2, 0, −1], [−2, 3, 1]]))
```

```
    [array([[0.33333333 , −0.666666667 , 0.66666667] , [0.66666667 , 0.66666667 , 0.33333333] ,
    [−0.66666667 , 0.33333333 , 0.66666667]])
array([[3.0000000000e + 00 , −3.00000000e + 00 , 0.000000000e + 00 ],
    [0.000000e + 00 , 3.0000000 + e00 , −3.000000000 + e00],
[0.000000e + 00 , −1.11022302e − 16 , 3.000000000 + e00]])]
```

On retrouve bien Q et R car $3.0000000 + e00 = 3$, $-1.11022302e - 16 = 0$, $0.66666667 = 2/3$ et $0.33333333 = 1/3$.

Remarque : Pour P précédent, le calcul symbolique permet de retrouver les fractions exactes.

Plus exactement, il existe une fonction *qr* du sous-module *numpy.linalg* qui permet de remplacer $Decomp\_QR$ et la fonction $QRdecomposition(\,)$ du module *sympy*. Illustrons avec P.

```
>>> import numpy.linalg as alg ; from sympy import *
>>> alg.qr(np.array([[12, −51, 4], [6, 167, −68], [−4, 24, −41]]))
```

```
(array([[−0.85714286 , 0.39428571 , 0.33142857] , [−0.42857143 , −0.90285714 , −0.03428571] ,
    [0.28571429 , −0.17142857 , 0.94285714]]) , array([[−14., −21. , 14.], [0., −175., 70.], [0., 0., −35.]]))
```

```
>>> (Q, R) = Matrix([[12, −51, 4], [6, 167, −68], [−4, 24, −41]]).QRdecomposition( )
>>> Q
[6/7, −69/175, −58/175], [3/7, 158/175, 6/175], [−2/7, 6/35, −33/35]
```

On peut remarquer que *alg.qr* donne $-Q$ et $-R$, comme $P = QR = (-Q)(-R)$, le produit reste juste mais attention, alors $-R \in T_n^-(\mathbf{R})$. Enfin, Q et R sont exacts avec le module *sympy*.

4. Application de la décomposition QR à la résolution d'un système linéaire.

4-a On considère le système $(\mathcal{S})$: $Ax = \begin{pmatrix} 1 \\ -1 \\ 2 \end{pmatrix}$.

Montrons 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. Résolvons alors ce système.

$$QRx = \begin{pmatrix} 1 \\ -1 \\ 2 \end{pmatrix} \Leftrightarrow Rx = Q^T \begin{pmatrix} 1 \\ -1 \\ 2 \end{pmatrix} = \frac{1}{3} \begin{pmatrix} 1 & -2 & 2 \\ 2 & 2 & 1 \\ -2 & 1 & 2 \end{pmatrix} \begin{pmatrix} 1 \\ -1 \\ 2 \end{pmatrix} = \begin{pmatrix} 7/3 \\ 2/3 \\ 1/3 \end{pmatrix} = y.$$

Si l'on pose $x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$, il reste :

$$Rx = \begin{pmatrix} 3 & -3 & 0 \\ 0 & 3 & -3 \\ 0 & 0 & 3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 7/3 \\ 2/3 \\ 1/3 \end{pmatrix} \Leftrightarrow \begin{cases} 3x_1 - 3x_2 &= 7/3 \\ 3x_2 - 3x_3 &= 2/3 \\ 3x_3 &= 1/3 \end{cases}$$

La dernière équation donne $x_3 = 1/9$ puis on remonte, $x_2 = 1/3$ et $x_1 = 10/9$.

4-b Soit b un vecteur de $\mathbf{R}^n$ et P une matrice inversible de $\mathcal{M}_n(\mathbf{R})$. Expliquons 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 $x \in \mathbf{R}^n$.

L'équation donnée est alors équivalente à

$$Rx = Q^{-1}b = Q^T b = y$$

en posant $y = Q^T b = (y_1\, y_2\, \ldots\, y_n)^T$.

Comme R est une matrice triangulaire supérieure et à coefficients diagonaux strictement positifs, on pourra résoudre cette dernière équation "en remontant". En effet, on commencera par observer que $x_n = \dfrac{1}{r_{n,n}} y_n$, ce qui permettra d'effectuer une substitution pour obtenir x_{n-1} à partir de l'identité $x_{n-1} = \dfrac{1}{r_{n-1,n-1}} (y_n - r_{n-1,n} x_n)$, ebenda.