Chapitre 7 : Exemples d'exercices corrigés

Enoncé, Exercice 7.1

On note $E = M_2(\mathbb{R})$. On pose pour $X = \begin{pmatrix} x \\ y \end{pmatrix}$ et $Y = \begin{pmatrix} x' \\ y' \end{pmatrix}$: $< X, Y >= 2xx' + xy' + x'y + yy'$

Montrer que $(E, <, >)$ est un espace euclidien.

Correction

On remarque que l'on peut écrire ce produit scalaire matriciellement en posant $A = \begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix}$.
On a alors $< X, Y >= X^T AY$

Soit $(X, Y, Z) \in E^3$ et $\lambda \in \mathbb{R}$. On écrit $X = \begin{pmatrix} x \\ y \end{pmatrix}$ avec $(x, y) \in \mathbb{R}^2$

Alors : i) $< X, Y + \lambda Z >= X^T A(Y + \lambda Z) = X^T AY + \lambda X^T AZ =< X, Y > +\lambda < X, Z >$
ii) $< X, Y >= X^T AY = (X^T AY)^T$ car on un réel (vu comme une matrice 1×1).
Et donc $< X, Y >= Y^T A^T X = Y^T AX =< Y, X >$ car $A^T = A$
iii) $< X, X >= 2x^2 + 2xy + y^2 = x^2 + x^2 + 2xy + y^2 = x^2 + (x + y)^2 \geq 0$
iv) $< X, X >= 0 \Rightarrow x^2 + (x + y)^2 = 0 \Rightarrow \begin{cases} x = 0 \\ x + y = 0 \end{cases} \Rightarrow x = y = 0 \Rightarrow X = 0_E$

On a donc : $\forall (X, Y, Z) \in E^3$, $\forall \lambda \in \mathbb{R}$ $\begin{cases} < X, Y + \lambda Z >=< X, Y > +\lambda < X, Z > \\ < X, Y >=< Y, X > \\ < X, X >\geq 0 \\ X, X >= 0 \Rightarrow X = 0_E \end{cases}$

Donc $<, >$ est un produit scalaire sur E.

Comme E est de dimension finie : $\boxed{(E, <, >) \text{ est un espace euclidien.}}$

Enoncé, Exercice 7.2

Soit $E = \mathbb{R}^3$ et sa structure euclidienne canonique. Soit $B = (i, j, k)$ la base orthonormée canonique de $\mathbb{R}^3$ et P le plan d'équation : $x + y - z = 0$ dans B.

Déterminer les matrices relativement à B de p la projection orthogonale et de s la réflexion de plan P.

Correction

$i + j - k$ est normal à P donc $p(i + j - k) = 0_E$ et par linéarité $p(i) + p(j) - p(k) = 0_E \Leftrightarrow (1)$
$i - j \in P$ donc $p(i - j) = i - j$ et par linéarité $p(i) - p(j) = i - j \Leftrightarrow (2)$
$i + k \in P$ donc $p(i + k) = i + k$ et par linéarité $p(i) + p(k) = i + k \Leftrightarrow (3)$

$(1) + (2) + (3) \Rightarrow p(i) = \frac{2i-j+k}{3}$

Alors $(2) \Rightarrow p(j) = \frac{-i+2j+k}{3}$ et $(3) \Rightarrow p(k) = \frac{i+j+2k}{3}$

On a donc $M_B(p) = \frac{1}{3}\begin{pmatrix} 2 & -1 & 1 \\ -1 & 2 & 1 \\ 1 & 1 & 2 \end{pmatrix}$

Comme $s = 2p - Id_E$ alors $M_B(s) = 2M_B(p) - I_3 = \frac{1}{3}\begin{pmatrix} 1 & -2 & 2 \\ -2 & 1 & 2 \\ 2 & 2 & 1 \end{pmatrix}$

Bilan : $\boxed{M_B(p) = \frac{1}{3}\begin{pmatrix} 2 & -1 & 1 \\ -1 & 2 & 1 \\ 1 & 1 & 2 \end{pmatrix} \text{ et } M_B(s) = \frac{1}{3}\begin{pmatrix} 1 & -2 & 2 \\ -2 & 1 & 2 \\ 2 & 2 & 1 \end{pmatrix}}$

Enoncé, Exercice 7.3

Appliqué le procédé d'orthonormalisation de Schmidt à la famille (u, v, w) de $\mathbb{R}^3$ muni du produit scalaire canonique avec $u = \begin{pmatrix} 1 \\ 1 \\ 2 \end{pmatrix}$, $v = \begin{pmatrix} 2 \\ 2 \\ 1 \end{pmatrix}$, et $w = \begin{pmatrix} 1 \\ -1 \\ 1 \end{pmatrix}$

Correction

On suit l'algorithme du cours.

Etape 1 : On pose $e_1 = \frac{u}{||u||}$

$< u, u >=< \begin{pmatrix} 1 \\ 1 \\ 2 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \\ 2 \end{pmatrix} >= 1^2 + 1^2 + 2^2 = 6$ et donc $e_1 = \frac{1}{\sqrt{6}}\begin{pmatrix} 1 \\ 1 \\ 2 \end{pmatrix}$

Etape 2 : On pose $e_2 = \frac{v-<v,e_1>e_1}{||v-<v,e_1>e_1||}$

Application numérique : $<v, e_1> = < \begin{pmatrix} 2 \\ 2 \\ 1 \end{pmatrix}, \frac{1}{\sqrt{6}} \begin{pmatrix} 1 \\ 1 \\ 2 \end{pmatrix} > = \frac{1}{\sqrt{6}}(2 \times 1 + 2 \times 1 + 1 \times 2) = \frac{6}{\sqrt{6}} = \sqrt{6}$

Donc $v- < v, e_1 > e_1 = \begin{pmatrix} 2 \\ 2 \\ 1 \end{pmatrix} - \sqrt{6}\frac{1}{\sqrt{6}} \begin{pmatrix} 1 \\ 1 \\ 2 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ -1 \end{pmatrix}$ qui est de norme $\sqrt{3}$.

Donc $e_2 = \frac{1}{\sqrt{3}} \begin{pmatrix} 1 \\ 1 \\ -1 \end{pmatrix}$

Etape 3 : On pose $e_3 = \frac{w-<w,e_1>e_1-<w,e_2>e_2}{||w-<w,e_1>e_1-<w,e_2>e_2||}$

$$w- < w, e_1 > e_1 - < w, e_2 > e_2$$
$$= \begin{pmatrix} 1 \\ -1 \\ 1 \end{pmatrix} - \frac{1}{6} < \begin{pmatrix} 1 \\ -1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \\ 2 \end{pmatrix} > \begin{pmatrix} 1 \\ 1 \\ 2 \end{pmatrix} - \frac{1}{3} < \begin{pmatrix} 1 \\ -1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \\ -1 \end{pmatrix} > \begin{pmatrix} 1 \\ 1 \\ -1 \end{pmatrix}$$
$$= \begin{pmatrix} 1 \\ -1 \\ 1 \end{pmatrix} - \frac{1}{3} \begin{pmatrix} 1 \\ 1 \\ 2 \end{pmatrix} + \frac{1}{3} \begin{pmatrix} 1 \\ 1 \\ -1 \end{pmatrix}$$
$$= \begin{pmatrix} 1 \\ -1 \\ 0 \end{pmatrix}$$

On a donc $e_3 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \\ 0 \end{pmatrix}$

Conclusion

La base orthogonale obtenue en appliquant le procédé d'orthonormalisation de Schmidt est donc :

$$\boxed{(e_1, e_2, e_3) \text{ avec } e_1 = \frac{1}{\sqrt{6}} \begin{pmatrix} 1 \\ 1 \\ 2 \end{pmatrix}, e_2 = \frac{1}{\sqrt{3}} \begin{pmatrix} 1 \\ 1 \\ -1 \end{pmatrix} \text{ et } e_3 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \\ 0 \end{pmatrix}}$$

Enoncé, Exercice 7.4

On note $E = \mathbb{R}[X]$ et $F = \mathbb{R}_2[X]$. On pose $\forall (P,Q) \in E \quad < P, Q >= \int\limits_0^{+\infty} e^{-t} P(t)Q(t)dt$

a) Montrer que $<,>$ définit un produit scalaire sur E.
b) Déterminer d la distance de X^3 à F.

Correction

a) • On commence par montrer que l'intégrale définissant $<,>$ est convergente.
Soit $R \in E$. Alors $t \mapsto e^{-t}R(t)$ est continue sur $[0, +\infty[$.
De plus $\frac{e^{-t}R(t)}{e^{-t/2}} = e^{-t/2}R(t) \underset{t\to+\infty}{\longrightarrow} 0$ donc $e^{-t}R(t) = o(e^{-t/2})$ au voisinage de $+\infty$.
Comme $t \mapsto e^{-t/2}$ est intégrable sur $[0, +\infty[$, alors, par négligeabilité $t \mapsto e^{-t}R(t)$ est intégrable sur $[0, +\infty[$
On en déduit que $\int\limits_0^{+\infty} e^{-t}R(t)dt$ est convergente.
En particulier si $(P,Q) \in E^2$, alors en posant $R = PQ$ on a : $\int\limits_0^{+\infty} e^{-t}P(t)Q(t)dt$ convergente et donc $<,>$ est bien définie sur E^2.

• Soit $(P,Q,R) \in E^3$ et $\lambda \in \mathbb{R}$. Alors :
i) $< P, Q + \lambda R >= \int_0^{+\infty} e^{-t}P(t)(Q(t) + \lambda R(t))dt = \int_0^{+\infty} e^{-t}P(t)Q(t)dt + \lambda \int_0^{+\infty} e^{-t}P(t)R(t)dt$ car les intégrales sont convergentes et donc $< P, Q + \lambda R >=< P, Q > +\lambda < P, R >$
ii) On a : $< P, Q >=< Q, P >$, évident par commutativité de la multiplication dans $\mathbb{R}$.
iii) $< P, P >= \int_0^{+\infty} e^{-t}P(t)^2dt \geq 0$ car $\forall t \geq 0$, $e^{-t}P(t)^2 \geq 0$ et positivité de l'intégrale.
iv) $< P, P >= 0 \Rightarrow \int_0^{+\infty} e^{-t}P(t)^2dt$
Mais $t \mapsto e^{-t}P(t)^2$ est continue et positive sur $[0, +\infty[$.
Donc par le théorème de l'intégrale nulle, on a $\forall t \geq 0$, $e^{-t}P(t)^2 = 0$ et donc $P(t) = 0$
P est alors un polynôme ayant une infinité de racines et donc $P = 0_E$

On a alors : $\forall (P,Q,R) \in E^3$, $\forall \lambda \in \mathbb{R}$, $\begin{cases} i) < P, Q + \lambda R >=< P, Q > +\lambda < P, R > \\ ii) < P, Q >=< Q, P > \\ iii) < P, P > \geq 0 \\ iv) < P, P >= 0 \Longrightarrow P = 0_E \end{cases}$ et on en déduit que : $\boxed{<,> \text{ est un produit scalaire sur } E.}$

b) • Calculs préliminaires. On pose $\forall k \in \mathbb{N}$, $I_k = \int\limits_0^{+\infty} t^k e^{-t}dt$
Alors, pour $k \geq 1$ et par intégration par parties, comme $\lim\limits_{t\to+\infty} t^k e^{-t} = 0$ et que les intégrales sont convergentes :

$$I_k = [-e^{-t}t^k]_0^{+\infty} + \int\limits_0^{+\infty} e^{-t}kt^{k-1}dt \text{ et donc } I_k = kI_{k-1}$$

$$I_0 = \int\limits_0^{+\infty} e^{-t}dt = [-e^{-t}]_0^{+\infty} = 1$$

On reconnaît la formule de récurrence définissant la factorielle (on peut aussi faire une récurrence) et on a : $\forall k \in \mathbb{N}$, $I_k = k!$

On remarquera pour les calculs que $< P, Q >=< PQ, 1 >$ et que $< X^k, 1 >= I_k$...

• On va commencer par trouver une base orthogonale de F en appliquant le procédé de Schmidt à la base canonique de F.
On pose $P_0 = 1$, $P_1 = X$ et $P_2 = X^2$, ainsi la base canonique est la base (P_0, P_1, P_2) et on construit (Q_0, Q_1, Q_2) une base orthonormée de F.

Etape 1 : On pose $Q_0 = \frac{P_1}{||P_1||}$

$||P_1||^2 =< P_1, P_1 >= \int\limits_0^{+\infty} e^{-t}dt = I_0 = 0! = 1$

On a donc $Q_0 = 1$

Etape 2 : On pose $Q_1 = \frac{P_1 - <P_1, Q_0> Q_0}{||P_1 - <P_1, Q_0> Q_0||}$
$< P_1, Q_0 >=< X, 1 >= I_1 = 1$ donc $\tilde{Q_1} = P_1 - < P_1, Q_0 > Q_0 = X - 1$
$< \tilde{Q_1}, \tilde{Q_1} >=< X - 1, X - 1 >=< X^2 - 2X + 1, 1 >= I_2 - 2I_1 + I_0 = 2! - 2.1 + 1 = 1$ et donc $Q_1 = X - 1$

Etape 3 : On pose $Q_2 = \frac{P_2 - <P_2, Q_0> Q_0 - <P_2, Q_1> Q_1}{||P_2 - <P_2, Q_0> Q_0 - <P_2, Q_1> Q_1||}$
On pose $\tilde{Q_2} = P_2 - < P_2, Q_0 > Q_0 - < P_2, Q_1 > Q_1$
$< P_2, Q_0 >=< X^2, 1 >= I_2 = 2$, $< P_2, Q_1 >=< X^2, X - 1 >= I_3 - I_2 = 6 - 2 = 4$
Donc $\tilde{Q_2} = X^2 - 2 - 4(X - 1) = X^2 - 4X + 2$
$< \tilde{Q_2}, \tilde{Q_2} >=< X^2 - 4X + 2, X^2 - 4X + 2 >=< (X^2 - 4X + 2)^2, 1 >$
donc $< \tilde{Q_2}, \tilde{Q_2} >=< X^4 + 16X^2 + 4 + 4X^2 - 16X - 8X^3, 1 >$
donc $< \tilde{Q_2}, \tilde{Q_2} >= I_4 - 8I_3 + 20I_2 - 16I_1 + 4I_0 = 4.6 - 8.6 + 40 - 16 + 4 = -24 + 24 + 4 = 4$
Donc $Q_2 = \frac{X^2 - 4X + 2}{2}$

Bilan : $(Q_0, Q_1, Q_2) = (1, X - 1, \frac{X^2 - 4X + 2}{2})$ est une base orthonormée de F.

• Par définition $d = d(X^3, F) = \left|\left|X^3 - R\right|\right|$ ou R est la projection orthogonale de X^3 sur F.
Par le théorème de projection orthogonale, comme (Q_0, Q_1, Q_2) est une base orthonormée de F :
$R =< X^3, Q_0 > Q_0 + < X^3, Q_1 > Q_1 + < X^3, Q_2 > Q_2$
$< X^3, Q_0 >=< X^3, 1 >= I_3 = 6$,
$< X^3, Q_1 >=< X^3, X - 1 >= I_4 - I_3 = 24 - 6 = 18$,
$< X^3, Q_2 >=< X^3, \frac{X^2 - 4X + 2}{2} >= \frac{I_5 - 4I_4 + 2I_3}{2} = \frac{120 - 96 + 12}{2} = 18$
Donc $R = 6 + 18(X - 1) + 18\frac{X^2 - 4X + 2}{2} = 9X^2 - 18X + 6$

On a alors : $d = \left|\left|X^3 - R\right|\right|$ qui, après application numérique (longue) donne : $\boxed{d = 6}$

Remarque : on a fait la fin des calculs avec l'ordinateur, d'où la solution Python suivante :

```
from numpy.polynomial import Polynomial
              # produit scalaire
def fact(k):
    if k==0:
        return 1
    return k*fact(k-1)
def ps(P,Q):
    R=P*Q
    n=R.degree()
    r=0
    for i in range(n+1):
        r=r+R.coef[i]*fact(i)
    return r
               # création base canonique
N=3+1 # de R_3[X]
BC=[Polynomial([0]*k+[1]) for k in range(N)]
               # algo de scmidt à base canonique
NewB=[Polynomial([1])] # NewB=nouvelle base
for i in range(1,N):
    NP=BC[i]
    for j in range(0,i):
        NP=NP-(ps(BC[i],NewB[j])/ps(NewB[j],NewB[j]))*NewB[j]
    NewB.append(NP)
                     # affichage de la base orthogonale
for i in range(len(NewB)):
    print(NewB[i].coef)
                        # projection de X^3 sur R_2[X]
R,PR=BC[3],Polynomial([0])
for k in range(3):
    PR=PR+(ps(R,NewB[k])/ps(NewB[k],NewB[k]))*NewB[k]

print('projection de X^3',PR.coef)
                                      # distance
d=ps(R-PR,R-PR)**(1/2)
print('d=',d)
```