

HEC Paris – ESSEC

Mathématiques I

Mathématiques approfondies

Sujet adapté – 2025

Table des matières

1	Partie I : autour de l'adjoint	2
	<i>Thèmes abordés : espaces vectoriels ; applications linéaires ; algèbre bilinéaire ; réduction ; Python.</i>	2
2	Partie II : minimisation d'une fonction quadratique	3
	<i>Thèmes abordés : fonctions de plusieurs variables ; algèbre bilinéaire ; variables à densité ; estimations ; Python.</i>	3
3	Partie III : minimisation d'une fonction non différentiable	5
	<i>Thèmes abordés : fonctions de plusieurs variables ; algèbre bilinéaire ; applications linéaires ; Python.</i>	5
	Annexe : formulaire de commandes Python	6

Notations. Pour tout $n \in \mathbb{N}^*$, I_n désigne la matrice identité de $M_n(\mathbb{R})$. Pour $(n, p) \in (\mathbb{N}^*)^2$ et $A \in M_{n,p}(\mathbb{R})$,

$$\ker(A) = \{X \in M_{p,1}(\mathbb{R}) \mid AX = 0\}, \quad \text{Im}(A) = \{AX \mid X \in M_{p,1}(\mathbb{R})\}.$$

Une matrice de $M_1(\mathbb{R})$ est identifiée au réel qui la constitue. Le coefficient situé à la ligne i et à la colonne j d'une matrice A est noté $A_{i,j}$. Si $X \in M_{p,1}(\mathbb{R})$, sa i -ième composante est notée x_i . Le produit scalaire canonique de $M_{n,1}(\mathbb{R})$ est $\langle X, Y \rangle = {}^tXY = \sum_{i=1}^n x_i y_i$, et $\|X\| = \sqrt{\langle X, X \rangle} = \sqrt{{}^tXX}$.

Toutes les variables aléatoires sont définies sur un même espace probabilisé; leur espérance et leur variance sont notées respectivement $\mathbb{E}$ et $\mathbb{V}$ lorsqu'elles existent. Les éléments de $\mathbb{R}^k$ sont considérés comme des vecteurs colonnes.

Dans tout le sujet, n, m, p sont des entiers naturels non nuls tels que $n \geq 4$ et $n \geq p$. Les matrices réelles fixes $A \in M_{n,p}(\mathbb{R})$ et $B \in M_{m,p}(\mathbb{R})$ sont données. On pose $M = {}^tAA$ et, pour $X \in M_{p,1}(\mathbb{R})$,

$$D(X) = MX - {}^tAY.$$

On s'intéresse au problème d'optimisation

$$\text{Minimiser} \quad \frac{1}{2} \|AX - Y\|^2 + \mu \|BX\|, \quad X \in M_{p,1}(\mathbb{R}),$$

où $Y \in M_{n,1}(\mathbb{R})$ est fixé, sauf indication contraire, et $\mu \in \mathbb{R}$. Dire que X réalise un minimum global de G signifie que $G(X)$ est un minimum global de G .

Le sujet comporte trois parties. La partie I établit des résultats d'algèbre linéaire utiles ensuite. La partie II étudie le problème lorsque $\mu = 0$; la partie III le traite lorsque $\mu = 1$. Pour les programmes Python, on peut utiliser les bibliothèques suivantes :

```
import numpy as np
import matplotlib.pyplot as plt
import numpy.random as rd
import numpy.linalg as al
```

1 Partie I : autour de l'adjoint

Thèmes abordés : espaces vectoriels ; applications linéaires ; algèbre bilinéaire ; réduction ; Python.

Dans cette partie, E et F sont deux espaces euclidiens de dimensions respectives p et n . On note $\langle \cdot, \cdot \rangle_E$ le produit scalaire de E et $\mathcal{B}_E = (e_1, \dots, e_p)$ une base orthonormale de E ; de même, $\mathcal{B}_F = (f_1, \dots, f_n)$ est une base orthonormale de F . Pour un sous-espace G de E ou de F , $G^\perp$ désigne son supplémentaire orthogonal. L'application linéaire $u : E \rightarrow F$ a pour matrice A dans les bases $\mathcal{B}_E$ et $\mathcal{B}_F$.

1. Soit $\ell : E \rightarrow \mathbb{R}$ linéaire. Montrer qu'il existe un unique vecteur $a_0 \in E$ tel que

$$\forall x \in E, \quad \ell(x) = \langle a_0, x \rangle_E.$$

2. En déduire que, pour tout $y \in F$, il existe un unique $z_y \in E$ tel que

$$\forall x \in E, \quad \langle u(x), y \rangle_F = \langle z_y, x \rangle_E. \quad (1)$$

3. Montrer que l'application $u^* : F \rightarrow E$, définie par $y \mapsto z_y$, est linéaire. Elle est appelée l'adjointe de u ; on a ainsi

$$\forall x \in E, \forall y \in F, \quad \langle u(x), y \rangle_F = \langle x, u^*(y) \rangle_E.$$

4. Montrer que $\text{Mat}_{\mathcal{B}_E, \mathcal{B}_F}(u^*) = {}^tA$, puis en déduire que $\text{rg}(u^*) = \text{rg}(u)$ et $(u^*)^* = u$.
5. Montrer que $\text{Im}(u^*) = \ker(u)^\perp$.
6. Montrer que $\ker(u^* \circ u) = \ker(u)$, puis en déduire que $\text{Im}(u^* \circ u) = \text{Im}(u^*)$.
7. Sur $\text{Im}(u^*)$, on définit w par $w(x) = u^* \circ u(x)$. Montrer que w est un isomorphisme et donner une interprétation matricielle de ce résultat.
8. Soit π le projecteur orthogonal de F sur $\text{Im}(u)$ et Q sa matrice dans $\mathcal{B}_F$.
- a) Montrer que ${}^tAQ = {}^tA$.
- b) Montrer que $\text{Tr}(Q) = \text{rg}(u)$.
9. Montrer que M est inversible si et seulement si A est de rang p .
10. On suppose A de rang p .
- a) Montrer que $Q = A(M^{-1})^tA$.
- b) Écrire une fonction Python `Calcule_Q(A)` qui reçoit une matrice `array A` de taille $n \times p$, teste si elle est de rang p et renvoie Q dans ce cas, et un message d'erreur sinon.
11. Montrer que, pour tout $X \in M_{p,1}(\mathbb{R})$, ${}^tXMX \geq 0$.

2 Partie II : minimisation d'une fonction quadratique

Thèmes abordés : fonctions de plusieurs variables ; algèbre bilinéaire ; variables à densité ; estimations ; Python.

On considère $J_0 : M_{p,1}(\mathbb{R}) \rightarrow \mathbb{R}$ définie par

$$J_0(X) = \frac{1}{2} \|AX - Y\|^2.$$

12. Montrer que, pour tous $X, H \in M_{p,1}(\mathbb{R})$,

$$J_0(X + H) - J_0(X) = \langle D(X), H \rangle + \frac{1}{2} {}^tHMH. \quad (2)$$

13. Montrer que J_0 possède un minimum global en X si et seulement si $D(X) = 0$. On note alors

$$S_0 = \{X \in M_{p,1}(\mathbb{R}) \mid D(X) = 0\}.$$

14. Montrer qu'il existe $X_0 \in \ker(A)^\perp$ tel que $S_0 \cap \ker(A)^\perp = \{X_0\}$.

15. Soit $X \in S_0$.

a) Justifier l'identité $AX = QY$. (3)

b) Montrer que $X - X_0 \in \ker(A)$, puis en déduire que si $X \neq X_0$, alors $\|X\| > \|X_0\|$.

c) Si A est de rang p , montrer que $S_0 = \{X\}$ avec $X = M^{-1}{}^tAY$. (4)

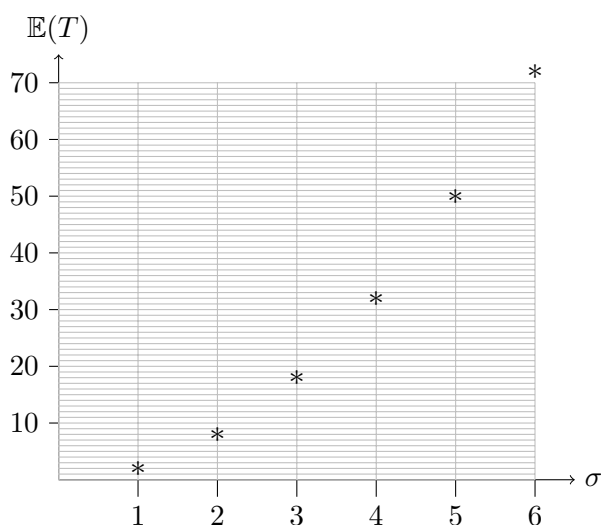
16. On suppose A de rang p et $Y = AU_0 + Z$, où U_0 est un vecteur constant et $Z = {}^t(z_1, \dots, z_n)$, les z_i étant indépendantes et de même loi normale $\mathcal{N}(0, \sigma^2)$. On définit X par (4), puis $T = \|A(X - U_0)\|^2$.

- a) Montrer que $T = \|QZ\|^2 = {}^tZQZ$.
- b) Proposer une fonction `simuleT(A,sigma)` recevant A et le flottant `sigma`, et simulant T .
- c) Proposer une fonction `esperanceT(A,sigma)` renvoyant une valeur approchée de l'espérance de T .
- d) Pour $\sigma = 1$, les essais menés avec les matrices

$$A_1 = \begin{pmatrix} 1 & 0 \\ 2 & 1 \\ 3 & 2 \end{pmatrix}, \quad A_2 = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 3 & 2 & 1 \\ 4 & 3 & 2 \end{pmatrix}, \quad A_3 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 & 0 \\ 3 & 2 & 1 & 0 & 0 \\ 4 & 3 & 2 & 1 & 0 \\ 5 & 4 & 3 & 2 & 1 \\ 6 & 5 & 4 & 3 & 2 \end{pmatrix}$$

fournissent respectivement 1,99, 3,01 et 4,97. Que peut-on conjecturer ? Le programme suivant, avec A_1 , conduit au graphique ci-dessous. Que peut-on conjecturer ?

```
sig=np.arange(1,7)
rep=np.zeros(6)
for k in range(6):
    rep[k]=esperanceT(A1,sig[k])
plt.plot(sig,rep,"*k")
plt.grid()
plt.show()
```



Prouvons ces conjectures dans les questions suivantes.

- e) Exprimer $\mathbb{E}(z_1^2)$, $\mathbb{E}(z_1^3)$ et $\mathbb{E}(z_1^4)$ en fonction de σ .
- f) Montrer que $T = T_1 + 2T_2$, avec $T_1 = \sum_{i=1}^n Q_{i,i} z_i^2$ et $T_2 = \sum_{i=1}^n \sum_{j=i+1}^n Q_{i,j} z_i z_j$. En déduire $\mathbb{E}(T)$ et vérifier les conjectures.
- g) Montrer que $\mathbb{E}(T_1 T_2) = 0$.
- h) En posant $q = \sum_{k=1}^n Q_{k,k}^2$, montrer que $\mathbb{E}(T_1^2) = \sigma^4(2q + p^2)$ et $\mathbb{E}(T_2^2) = \frac{\sigma^4}{2}(p - q)$.
- i) En déduire $\mathbb{V}(T) = 2p\sigma^4$, puis montrer que pour tout $\alpha > 0$,

$$\mathbb{P}(T > (1 + \alpha)p\sigma^2) \leq \frac{2}{p\alpha^2}.$$

3 Partie III : minimisation d'une fonction non différentiable

Thèmes abordés : fonctions de plusieurs variables ; algèbre bilinéaire ; applications linéaires ; Python.

On suppose à nouveau que Y est fixé. On étudie la fonction

$$J(X) = \frac{1}{2} \|AX - Y\|^2 + \|BX\|, \quad X \in M_{p,1}(\mathbb{R}). \quad (5)$$

Pour $X \in M_{p,1}(\mathbb{R})$, on note

$$N(X) = \{ {}^tBV \mid V \in M_{m,1}(\mathbb{R}), \|V\| \leq 1 \text{ et } \|BX\|V - BX = 0 \}.$$

17. Soient $u, v \in \mathbb{R}^n$ avec $u \neq 0$. Montrer que

$$\lim_{t \rightarrow 0} \frac{\|u + tv\| - \|u\|}{t} = \frac{\langle u, v \rangle}{\|u\|}. \quad (6)$$

18. Soit X_0 réalisant le minimum global de J . Montrer que si $BX_0 \neq 0$, alors nécessairement

$$-D(X_0) = {}^tB \frac{BX_0}{\|BX_0\|},$$

puis en déduire que $-D(X_0) \in N(X_0)$.

19. Soit X_0 réalisant le minimum global de J et vérifiant $BX_0 = 0$.

- Montrer que, pour tout $H \in M_{p,1}(\mathbb{R})$, $\langle D(X_0), H \rangle \leq \|BH\|$.
- En déduire que $D(X_0) \in \ker(B)^\perp$.
- En déduire qu'il existe $W \in M_{m,1}(\mathbb{R})$ tel que $D(X_0) = {}^tBW$.
- En déduire qu'il existe $V \in M_{p,1}(\mathbb{R})$ tel que $D(X_0) = {}^tBBV$.
- Montrer que $\|BV\| \leq 1$ et en déduire que $-D(X_0) \in N(X_0)$.

20. Soient $u, v \in \mathbb{R}^n$, avec $u \neq 0$.

a) Montrer que

$$\|u + v\|^2 - \left(\|u\| + \frac{\langle u, v \rangle}{\|u\|} \right)^2 = \frac{\|u\|^2 \|v\|^2 - \langle u, v \rangle^2}{\|u\|^2}.$$

b) En déduire que $\|u + v\| - \|u\| \geq \frac{\langle u, v \rangle}{\|u\|}$.

21. Soient $X_0, X \in M_{p,1}(\mathbb{R})$ et $W \in N(X_0)$.

- Montrer que $\|BX\| - \|BX_0\| \geq \langle W, X - X_0 \rangle$.
- En déduire que $J(X) - J(X_0) \geq \langle D(X_0) + W, X - X_0 \rangle$.

22. En déduire que X_0 réalise un minimum global de J si et seulement si $-D(X_0) \in N(X_0)$.

23. Dans toute la suite, $n = p = m$, $A = I_n$, et B est diagonale, définie par

$$B_{i,i} = \begin{cases} \alpha_i & \text{si } 1 \leq i \leq k, \\ 0 & \text{si } k+1 \leq i \leq n, \end{cases}$$

où $1 \leq k < n$ et les α_i sont non nuls. Posons $\rho(Y) = \sum_{i=1}^k \frac{y_i^2}{\alpha_i^2}$, et définissons, pour $\lambda \geq 0$,

$$F(\lambda) = -1 + \sum_{i=1}^k \frac{\alpha_i^2 y_i^2}{(\lambda + \alpha_i^2)^2}.$$

- a) Montrer que, pour tout $\lambda > 0$, $F(\lambda) \leq -1 + \frac{1}{4\lambda} \sum_{i=1}^k y_i^2$.
- b) En supposant uniquement ici que $\rho(Y) > 1$, montrer qu'il existe un unique $\lambda_0 > 0$ tel que $F(\lambda_0) = 0$, et justifier $\lambda_0 \leq \frac{1}{4} \sum_{i=1}^k y_i^2$.
- c) Posons $\beta = 0$ si $\rho(Y) \leq 1$ et $\beta = \lambda_0$ sinon. On définit $X_0 = Y - BV$, avec

$$v_i = \begin{cases} \frac{\alpha_i y_i}{\beta + \alpha_i^2} & \text{si } 1 \leq i \leq k, \\ 0 & \text{si } k+1 \leq i \leq n. \end{cases}$$

Montrer que X_0 réalise un minimum global de J .

- d) **Python.** Les données sont un tableau unidimensionnel **alpha** contenant $\alpha_1, \dots, \alpha_k$, un tableau **Y** contenant le vecteur Y , un flottant positif **lda** représentant λ et un flottant positif **epsilon** représentant ε .
- Écrire **FoncSom(alpha,Y,lda)** qui renvoie $F(\lambda)$.
 - Écrire **CalcBeta(alpha,Y,epsilon)** qui renvoie une valeur approchée β_a de β à ε près, obtenue par dichotomie dans $\left]0, \frac{1}{4} \sum_{i=1}^k y_i^2\right]$ lorsque $\rho(Y) > 1$.
 - Écrire **CalcSolution(alpha,Y,epsilon)** qui renvoie un tableau contenant la solution X_0 ci-dessus.

Annexe : formulaire de commandes Python

Mathématiques générales

- **np.linspace(a,b,n)** crée un tableau de n valeurs uniformément réparties entre **a** et **b**.
- **np.zeros([n,m])** et **np.zeros(n)** créent respectivement une matrice nulle de taille $n \times m$ et un vecteur nul de taille n .
- **np.arange(a,b,eps)** renvoie les flottants de **a** à **b** exclu, de pas **eps**.
- **np.shape(M)** donne la taille de **M**; **np.transpose(M)** renvoie sa transposée.
- **np.dot(M,P)**, **M.dot(P)** et **M@P** calculent le produit matriciel MP .

Algèbre linéaire

al.inv(M) renvoie l'inverse de **M**; **al.matrix_rank(M)** son rang; **al.matrix_power(M,n)** sa puissance n -ième.

Simulations probabilistes

- **rd.exponential(a,[q,r])** et **rd.exponential(a,n)** simulent des coefficients indépendants suivant la loi exponentielle de paramètre $1/a$.
- **rd.normal(m,d,[q,r])** et **rd.normal(m,d,n)** simulent des coefficients indépendants suivant $\mathcal{N}(m, d^2)$.
- **rd.gamma(a,1,[q,r])** et **rd.gamma(a,1,n)** simulent des coefficients indépendants suivant $\gamma(a)$.

Graphiques

plt.plot(X,Y,options) trace les points définis par les listes **X** et **Y**; **plt.grid()** affiche un quadrillage et **plt.show()** affiche le graphique.