Problème de Mathématiques

Référence pp1605 — Version du 6 décembre 2025

À chaque matrice
$$M = (m_{i,j})_{1\leqslant i\leqslant p, 1\leqslant j\leqslant q} \in \mathfrak{M}_{p,q}(\mathbb{R}),$$
on associe la **norme de** M, définie par
$$\|M\| = \max_{\substack{1\leqslant i\leqslant p\\ 1\leqslant j\leqslant q}} |m_{i,j}| \in \mathbb{R}.$$
On remarque que ce réel est toujours positif et ne s'annule que si M est la matrice nulle.

1. Écrire une fonction `Norme(M)` qui, une matrice M de taille quelconque étant donnée, calcule et retourne le nombre réel $\|M\|$.

2. Écrire une fonction `Normalise(v)` qui, une matrice colonne v *non nulle* étant donnée, retourne la matrice colonne
$$\widetilde{v} = \frac{v}{\|v\|}.$$

3. On se donne une matrice carrée $A \in \mathfrak{M}_p(\mathbb{R})$ et une matrice colonne $v_0 \in \mathfrak{M}_{p,1}(\mathbb{R})$. En supposant qu'aucun des termes n'appartient au noyau de A, on peut former la suite $(v_n)_{n\in\mathbb{N}}$ définie par la relation de récurrence :
$$\forall\, n \in \mathbb{N}, \quad v_{n+1} = \frac{Av_n}{\|Av_n\|}.$$

3. a. On suppose que la matrice A n'est pas nilpotente. Démontrer qu'il existe un hyperplan H de $\mathfrak{M}_{p,1}(\mathbb{R})$ tel que la suite $(v_n)_{n\in\mathbb{N}}$ soit bien définie quel que soit le choix de v_0 dans H^c.

En déduire pourquoi, en choisissant v_0 aléatoirement, la suite $(v_n)_{n\in\mathbb{N}}$ est presque sûrement bien définie.

3. b. Écrire une fonction `PuissanceIteree(A, n)` qui, une matrice carrée A et un entier $n \in \mathbb{N}$ étant donnés, détermine la taille p de A, choisit aléatoirement une matrice colonne $v_0 \in \mathfrak{M}_{p,1}(\mathbb{R})$ et retourne la matrice colonne v_n.

4. On suppose que la matrice A est diagonalisable : elle est semblable à
$$\Delta = \mathrm{Diag}(\lambda_1, \lambda_2, \dots, \lambda_p)$$
où les valeurs propres sont indexées de telle sorte que
$$\lambda_1 > |\lambda_2| \geqslant \cdots \geqslant |\lambda_p|.$$
On considère enfin le sous-espace
$$V = \bigoplus_{\substack{\lambda\in \mathrm{Sp}(A)\\ \lambda\neq\lambda_1}} \mathrm{Ker}(A - \lambda I_p),$$

4. a. Quelle est la dimension de $\mathrm{Ker}(A - \lambda_1 I_p)$? Expliquer pourquoi, en choisissant v_0 de façon aléatoire, on a presque sûrement $v_0 \notin V$.

4. b. On suppose que $v_0 \notin V$. Expliquer pourquoi la suite $(v_n)_{n\in\mathbb{N}}$ est bien définie et converge vers un vecteur propre v de A associé à la valeur propre λ_1.

4. c. On a maintenant pour objectif d'écrire une fonction `VecteurPropre(A, e)` qui, étant donnés une matrice A qui satisfait les hypothèses décrites ci-dessus et un nombre $e > 0$, calcule les termes de la suite $(v_n)_{n\in\mathbb{N}}$ jusqu'à ce que $\|v_n - v_{n+1}\| < e$ et retourne alors v_{n+1}.

Indiquer succinctement ce qui ne va pas dans les programmes incorrects.

Programme A

```
def VecteurPropre(A, e):
    d = A.shape
    v = np.matrix(rand(d[0],1))
    v = Normalise(v)
    w = Normalise(A*v)
    ecart = Norme(v-w)
```

```
    while ecart>=e:
        v = w
        w = Normalise(A*v)
    return w
```

Programme B

```
def VecteurPropre(A, e):
    d = A.shape
    v = np.matrix(rand(d[0],1))
    v = Normalise(v)
    while Norme(v-Normalise(A*v))>=e:
        v = Normalise(A*v)
    return Normalise(A*v)
```

Solution ❀ Calcul approché d'un vecteur propre

1. On calcule le maximum de $|m_{i,j}|$ lorsque les indices i et j parcourent l'ensemble des valeurs possibles.

```
def Norme(M):
    # Norme infinie de la matrice M
    p,q = M.shape
    N = np.max(np.abs(M[i,j]) \
               for i in range(p) \
               for j in range(q))
    return N
```

2. On divise la matrice v par sa norme, quelle que soit la taille de la matrice (matrice colonne ou autre). On ne tient pas compte du fait que la matrice v peut être nulle : c'est dangereux, il faudra en assumer les conséquences.

```
def Normalise(v):
    return v/Norme(v)
```

3. a. On sait que la suite des noyaux $\operatorname{Ker} A^n$ est une suite croissante (pour $\subset$) de sous-espaces vectoriels et (dimension finie!) que cette suite est stationnaire. Comme la matrice A n'est pas nilpotente, alors

$$\max_{n\in\mathbb{N}} \operatorname{Ker} A^n \subsetneq \mathfrak{M}_{p,1}(\mathbb{R}).$$

D'après le théorème de la base incomplète, tout sous-espace strict est contenu dans un hyperplan. Il existe donc un hyperplan H de $\mathfrak{M}_{p,1}(\mathbb{R})$ tel que

$$\forall\, n \in \mathbb{N}, \quad \operatorname{Ker} A^n \subset H.$$

❧ Comme la norme est **positivement homogène**, le vecteur v_{n+1} est défini si, et seulement si, $A^{n+1}v_0 \neq 0$ et

$$v_{n+1} = \frac{A^{n+1}v_0}{\|A^{n+1}v_0\|}$$

(démonstration par récurrence).

En choisissant v_0 dans H^c, on sait que $v_0 \notin \operatorname{Ker} A^{n+1}$, donc $A^{n+1}v_0 \neq 0$ pour tout $n \in \mathbb{N}$, ce qui prouve que la suite $(v_n)_{n\in\mathbb{N}}$ est bien définie.

❧ En choisissant aléatoirement le vecteur v_0, on est presque sûr qu'il n'appartient pas à l'hyperplan H et par conséquent la suite $(v_n)_{n\in\mathbb{N}}$ est bien définie.

REMARQUE.— Questions pour les sceptiques :

— Si on choisit un point au hasard dans le plan, quelle est la probabilité pour que ce point se situe sur l'axe des abscisses ?

— Si on choisit un point au hasard dans $\mathbb{R}^3$, quelle est la probabilité pour que ce point se situe sur le plan $[z = 0]$?

3. b. On initialise la matrice colonne v en choisit une colonne au hasard et on itère n fois la relation de récurrence.

```
def PuissanceIteree(A, n):
    p = A.shape[0]
    v = np.matrix(rand(p,1))
    for i in range(n):
        v = Normalise(A*v)
    return v
```

On peut choisir une matrice quelconque

```
A = np.matrix(rand(3,3))
```

et itérer 10^5 fois `PuissanceIteree(A,10)` sans jamais rencontrer une division par zéro!

4. a. Comme A et Δ sont semblables, leurs sous-espaces propres sont isomorphes :

$$\dim \operatorname{Ker}(A - \lambda_1 I_p) = \dim \operatorname{Ker}(\Delta - \lambda_1 I_p) = 1$$

puisque $(\Delta - \lambda_1 I_p)$ est diagonale et qu'*un seul* de ses coefficients diagonaux est nul :

$$\forall\, 2 \leqslant k \leqslant p, \quad |\lambda_k - \lambda_1| \geqslant |\lambda_1| - |\lambda_k| > 0.$$

❧ Comme A est diagonalisable, le sous-espace V est un supplémentaire de $\operatorname{Ker}(A - \lambda_1 I_p)$, donc c'est un hyperplan. Comme au **3.a.**, la probabilité qu'un vecteur v_0 pris au hasard appartienne à un hyperplan est nulle!

4. b. Comme A est diagonalisable, on peut décomposer le vecteur v_0 dans une base de vecteurs propres :

$$v_0 = \sum_{k=1}^{p} \alpha_k \cdot u_k$$

où u_k est associé à la valeur propre λ_k. Comme $v_0 \notin V$, alors $\alpha_1 \neq 0$ et, pour tout $n \in \mathbb{N}$,

$$\begin{aligned} A^n v_0 &= \sum_{k=1}^{p} \alpha_k \lambda_k^n \cdot u_k \\ &= \lambda_1^n \Big[\underbrace{\alpha_1 \cdot u_1}_{\neq 0} + \sum_{k=2}^{p} \alpha_k \underbrace{\Big(\frac{\lambda_k}{\lambda_1}\Big)^n}_{\to 0} \cdot u_k \Big] \end{aligned}$$

donc $A^n v_0 \neq 0$ (puisque la composante sur u_1 n'est jamais nulle) et lorsque n tend vers $+\infty$,

$$A^n v_0 \sim \lambda_1^n \alpha_1 \cdot u_1$$

donc

$$\begin{aligned} v_n = \frac{A^n v_0}{\|A^n v_0\|} &\sim \frac{\lambda_1^n \alpha_1 \cdot u_1}{\|\lambda_1^n \alpha_1 \cdot u_1\|} \\ &= \pm \frac{u_1}{\|u_1\|} \in \operatorname{Ker}(A - \lambda_1 I_p) \end{aligned}$$

où le signe en facteur est celui du scalaire α_1.

4. c. Dans le programme A, la valeur de la variable `ecart` n'est jamais modifiée, donc la boucle *while* ne se terminera que par `Ctrl+C`.

Dans le programme B, on effectue `Normalise(A*v)` *deux fois* par boucle : c'est une fois de trop! Mieux vaut utiliser deux variables et n'effectuer qu'une fois par boucle le calcul du terme suivant.

Voici un exemple de code convenable.

```
def VecteurPropre(A, e):
    d = A.shape
    v = np.matrix(rand(d[0],1))
    v = Normalise(v)
    w = Normalise(A*v)
    while Norme(v-w)>=e:
        v = w
        w = Normalise(A*v)
    return w
```