

Révisions/Formulaire 06

Révisions Algèbre: vecteurs, matrices et systèmes.

Vecteurs dans \mathbb{K}^n , opérations, numpy

Cette section doit se lire deux fois. Une première fois en remplaçant \mathbb{K} par \mathbb{R} , une deuxième fois en remplaçant \mathbb{K} par \mathbb{C} . Les éléments de \mathbb{K} se nomment les *scalaires*. Ce sont des nombres.

Fixons $n \in \mathbb{N}^*$. Un élément x de \mathbb{K}^n est un n -uplet d'éléments de \mathbb{K} que l'on note

$$x = \underbrace{(x_1, \dots, x_n)}_{n \text{ termes}} \text{ ou } x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$$

L'égalité de deux n -uplets est équivalente à l'égalité de chacune des *composantes* (ou de chacune des *entrées*) des n -uplets.

$$(x_1, \dots, x_n) = (y_1, \dots, y_n) \Leftrightarrow \forall k \in \{1, \dots, n\}, x_k = y_k$$

La première notation est la notation mathématique standard pour un n -uplet. La deuxième, officiellement au programme, est utile lorsque l'on utilise un logiciel de calcul standard (MATLAB, module Python numpy...) et est assez pratique d'un point de vue typographique.

Cette deuxième notation identifie implicitement un n -uplet de \mathbb{K}^n avec une matrice « colonne » $n \times 1$ à entrées dans \mathbb{K} . On **n'**identifie **pas** automatiquement¹ un n -uplet (x_1, \dots, x_n) avec la matrice « ligne » $(x_1 \ \dots \ x_n)$.

On note \mathbb{K}^n l'ensemble des n -uplets d'éléments de \mathbb{K} .

$$\mathbb{K}^n = \{x = (x_1, \dots, x_n), x_1 \in \mathbb{K}, \dots, x_n \in \mathbb{K}\}$$

Les deux notations (en ligne avec des ,, en colonne) traduisent l'identification quasi automatique entre \mathbb{K}^n et $\mathcal{M}_{n,1}(\mathbb{K})$ l'ensemble des matrices possédant une colonne et n lignes.

Sur \mathbb{K}^n , on définit deux opérations :

— L'une, *interne*, l'*addition*, notée $+$, définie par

$$\forall x, y, z \in \mathbb{K}^n, (z = x + y \Leftrightarrow \forall k \in \{1, \dots, n\}, z_k = x_k + y_k)$$

On notera que les deux arguments et le résultat d'une addition dans \mathbb{K}^n sont des éléments de \mathbb{K}^n

— L'autre, *externe*, la *multiplication par un scalaire*, notée \cdot , définie par

$$\forall \lambda \in \mathbb{K}, \forall x, y \in \mathbb{K}^n, (y = \lambda \cdot x \Leftrightarrow \forall k \in \{1, \dots, n\}, y_k = \lambda \cdot x_k)$$

On notera que dans une telle multiplication, le premier argument est un élément de \mathbb{K} , le second et le résultat sont des éléments de \mathbb{K}^n

1. Noter la différence subtile de notation avec présence ou pas de virgule

- On pose, pour $k \in \{1, \dots, n\}$, le k -ième vecteur de la base canonique de \mathbb{K}^n , $e_k = (0, \dots, 0, 1, 0, \dots, 0)$ où le 1 apparait en position k . Ceci permet de réécrire², à l'aide des opérations tout juste définies : $(x_1, \dots, x_n) = \sum_{k=1}^n x_k \cdot e_k$. De manière équivalente, en utilisant la notation en colonne

$$\begin{pmatrix} x_1 \\ \vdots \\ \cdot \\ \cdot \\ \cdot \\ \vdots \\ x_n \end{pmatrix} = \sum_{k=1}^n x_k \cdot e_k, \quad (e_k = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \rightarrow \text{ligne } k, k \in \{1, \dots, n\})$$

- Pour $x, y \in \mathbb{K}^n$, $\lambda, \mu \in \mathbb{K}$, on a :

$$\begin{aligned} \lambda \cdot x + \mu \cdot y &= \lambda \cdot \left(\sum_k x_k \cdot e_k \right) + \mu \cdot \left(\sum_k y_k \cdot e_k \right) \\ \text{[déf. } \lambda \cdot x, \mu \cdot y] &= \sum_k (\lambda \cdot x_k) \cdot e_k + \sum_k (\mu \cdot y_k) \cdot e_k \\ \text{[déf. } \lambda \cdot x + \mu \cdot y] &= \sum_k (\lambda \cdot x_k + \mu \cdot y_k) \cdot e_k \end{aligned}$$

Dans ce petit jeu d'écriture, il faut identifier la nature des opérations : $+$ c'est une addition entre vecteurs ou entre scalaires ? \cdot c'est une multiplication scalaire.vecteur ou scalaire.scalar ? A vous de mettre des couleurs !

- Le vecteur nul dans \mathbb{K}^n se note 0 (ou 0_n pour le distinguer du 0 scalaire habituel). Il vérifie :

$$\forall x \in \mathbb{K}^n, 0 + x = 0 \text{ et } \forall x \in \mathbb{K}^n, 0 \cdot x = 0$$

On peut utiliser l'ensemble d'indices $\{0, \dots, n-1\}$ à la place de $\{1, \dots, n\}$. Cela ne change rien aux grands principes mais peut mener à de petits décalages d'indices dans les formules.

Python/numpy

En Python, le module numpy importé sous l'alias habituel par l'instruction `import numpy as np`, les vecteurs numériques sont représentés par les « tableaux numpy », de type `ndarray`, indicés de 0 à $n-1$.

Un vecteur peut se définir en transformant une liste de nombres en tableau numpy par exemple :

```
e1 = np.array([1,0,0]); e2 = np.array([0,1,0]); e3 = np.array([0,0,1]);
```

Le vecteur nul de \mathbb{K}^n se représente/définit avec :

```
zero_n = np.zeros( (n, ) )
```

Le vecteur de \mathbb{K}^n ne comportant que des 1 se représente/définit avec :

```
un_n = np.ones( (n, ) )
```

On peut modifier un vecteur déjà défini en accédant à ses composantes grâce à la notation indiciaire Python `identificateur[indice]` :

```
e = [np.zeros( (n, ) ) for _ in range(n)] #Base canonique de K^n
for k in range(n) : e[k][k] = 1.0
```

Enfin les opérations $+$ et \cdot , se notent avec les identificateurs usuels $+$ et $*$:

```
x = 2*e[3] - 5*e[4]
```

2. Tout vecteur x de \mathbb{K}^n s'écrit (de manière unique) comme combinaison linéaire des vecteurs de la famille $\mathcal{E}_n = (e_1, \dots, e_n)$, la base canonique de \mathbb{K}^n .

Matrices à coefficients dans \mathbb{K}^n , numpy

Soit n et p deux entiers naturels non nuls. Une matrice M à coefficients dans \mathbb{K} , comportant n lignes et p colonnes est une famille de nombres dans \mathbb{K} (doublement) indexée par $\{1, \dots, n\} \times \{1, \dots, p\}$.

La donnée de l'ensemble d'indices $\{1, \dots, n\} \times \{1, \dots, p\}$ (ou plus simplement du couple (n, p)) est la donnée de la « taille³ » de la matrice.

L'ensemble des matrices M de taille $n \times p$ à coefficients dans \mathbb{K} est noté $\mathcal{M}_{n,p}(\mathbb{K})$.

Si $p = n$, on dit que la matrice M est carrée de taille n et l'ensemble des matrices carrées M de taille n à coefficients dans \mathbb{K} est noté $\mathcal{M}_n(\mathbb{K})$.

On note $M = (m_{i,j})_{1 \leq i \leq n, 1 \leq j \leq p}$ pour indiquer que $m_{i,j}$ est l'élément de M d'indice (i, j) où $(i, j) \in \{1, \dots, n\} \times \{1, \dots, p\}$. On dit aussi que m_{ij} (la virgule entre i et j est souvent oubliée au niveau typographique) est l'entrée de M en position (i, j) (ou d'indice (i, j)). On notera parfois directement $[M]_{i,j}$ pour désigner l'entrée de M en position (i, j) .

On peut, lorsque la place le permet, écrire exhaustivement la matrice M sous forme de tableau (d'où la dénomination en « ligne »/« colonne »)

Par exemple, la matrice $M = \left(\frac{1}{i+j^2}\right)_{1 \leq i \leq 2, 1 \leq j \leq 3}$ comporte 2 lignes, 3 colonnes, est de taille 2×3 et s'écrit exhaustivement :

$$M = \begin{pmatrix} \frac{1}{1+1^2} & \frac{1}{1+2^2} & \frac{1}{1+3^2} \\ \frac{1}{2+1^2} & \frac{1}{2+2^2} & \frac{1}{2+3^2} \end{pmatrix} = \begin{pmatrix} \frac{1}{2} & \frac{1}{5} & \frac{1}{10} \\ \frac{1}{3} & \frac{1}{6} & \frac{1}{11} \end{pmatrix}$$

Voici quelques exemples de matrices usuelles, avec du code Python/Numpy associé :

— La matrice nulle de taille $n \times p$, parfois notée $0_{n,p}$, ne comportant que des 0 :

```
zero_np = np.zeros( (n,p) )
```

— La matrice identité de $\mathcal{M}_n(\mathbb{K})$, notée I_n , vérifiant : $\forall (i, j) \in (\{1, \dots, n\})^2, [I_n]_{ij} = \begin{cases} 1 & \text{si } i = j \\ 0 & \text{si } i \neq j \end{cases}$:

```
I_n = np.eye(n)
```

— Une matrice diagonale D (carrée !!), i.e. telle que : $\forall (i, j) \in (\{1, \dots, n\})^2, i \neq j \Rightarrow [D]_{ij} = 0$:

```
D = np.diag([d1,d2,d3]) #Ici matrice 3x3. d1,d2,d3 sur la diagonale
```

— On peut modifier une matrice déjà définie en accédant à ses entrées grâce à la notation indiciaire double Python/Numpy $M[i, j]$:

```
D[1,2] = 5.0
```

ou, pour définir une matrice déjà rencontrée :

```
M = np.zeros((2,3))
for ip in range(2) : #ATTN décalage d'indice Python<>Math.
    for jp in range(3) :
        M[ip,jp] = 1/((ip+1)+(jp+1)*2) #i = ip+1; j = jp +1
```

— La taille d'une matrice est accessible via la méthode `shape` :

```
(n,p) = M.shape; n = M.shape[0]; p = M.shape[1]
```

3. ne pas utiliser le terme « dimension » qui est déjà surchargé de sens en mathématiques ; on pourrait parler de la « forme » de la matrice, ce qui n'est pas reconnu dans les livres de mathématiques mais est cohérent avec la dénomination anglo-saxonne *shape*, cf. opérateur en Python/Numpy

Matrices à coefficients dans \mathbb{K}^n , opérations et numpy

Concernant l'ensemble $\mathcal{M}_{n,p}(\mathbb{K})$, on peut y définir, comme sur \mathbb{K}^n , deux opérations :

— L'une, *interne*, l'*addition*, notée $+$, définie par

$$\forall A, B, C \in \mathcal{M}_{n,p}(\mathbb{K}), (C = A + B \Leftrightarrow \forall (i, j) \in \{1, \dots, n\} \times \{1, \dots, p\}, [C]_{ij} = [A]_{ij} + [B]_{ij})$$

Les deux arguments et le résultat d'une addition dans $\mathcal{M}_{n,p}(\mathbb{K})$ sont des éléments de $\mathcal{M}_{n,p}(\mathbb{K})$

— L'autre, *externe*, la *multiplication par un scalaire*, notée \cdot , définie par

$$\forall \lambda \in \mathbb{K}, \forall A, B \in \mathcal{M}_{n,p}(\mathbb{K}), (B = \lambda \cdot A \Leftrightarrow \forall (i, j) \in \{1, \dots, n\} \times \{1, \dots, p\}, [B]_{ij} = \lambda \cdot [A]_{ij})$$

Dans une telle multiplication, le premier argument est un élément de \mathbb{K} , le second et le résultat sont des éléments de $\mathcal{M}_{n,p}(\mathbb{K})$.

En Python/Numpy, ces opérations se notent avec les opérateurs binaires $+$ et $*$. Le langage Python analyse les opérandes de chaque opération, détermine leur type et utilise les versions appropriées à la situation de l'addition et de la multiplication.

L'opération la plus délicate est la multiplication matricielle. La situation se résume en :

- Si A comporte n lignes et p colonnes, B comporte p lignes et m colonnes alors le produit $A \cdot B$ est possible ;
- dans ce cas, le produit $A \cdot B$ est une matrice comportant n lignes et m colonnes avec :

$$\forall (i, j) \in \{1, \dots, n\} \times \{1, \dots, m\}, [A \cdot B]_{ij} = \sum_k [A]_{ik} \cdot [B]_{kj}$$

La somme a lieu sur l'ensemble $\{1, \dots, p\}$ des indices k décrivant les numéros de colonnes de A ainsi que les numéros de lignes de B .

- Si x est un vecteur de \mathbb{K}^p , on l'identifie naturellement avec la matrice colonne x de taille $p \times 1$ telle que $\forall k, 1 \leq k \leq p, [x]_{k1} = x_k$. Si maintenant A est une matrice de taille $n \times p$, le produit $A \cdot x$ est possible et a pour résultat la matrice y de taille $n \times 1$, identifié au vecteur $y \in \mathbb{K}^n$ défini par :

$$\forall i \in \{1, \dots, n\}, y_i = \sum_{k=1}^p [A]_{ik} \cdot x_k$$

- Si K est une matrice colonne $p \times 1$ et L une matrice ligne $1 \times p$ alors :

1. le produit $L \cdot K$ est bien défini, c'est une matrice 1×1 que l'on peut identifier à un nombre :

$$[L \cdot K]_{11} = \sum_{k=1}^p [L]_{1k} \cdot [K]_{k1}.$$

2. le produit $K \cdot L$ est bien défini, c'est une matrice $p \times p$ vérifiant

$$\forall (k, \ell) \in (\{1, \dots, p\})^2, [K \cdot L]_{k\ell} = [K]_{k1} \cdot [L]_{1\ell}.$$

3. Il est clair que $L \cdot K$ et $K \cdot L$ sont deux objets différents dès que $p > 1$. La question de leur égalité ne se pose même pas.

- On peut définir une matrice par ses lignes $A = \begin{pmatrix} L_1 \\ \vdots \\ L_n \end{pmatrix}$, ou par ses colonnes $B = (K_1 | \dots | K_m)$ et si le produit $A \cdot B$ est possible alors on a

$$\forall (i, j) \in \{1, \dots, n\} \times \{1, \dots, m\}, [A \cdot B]_{ij} = L_i \cdot K_j = \sum_{k=1}^p [L_i]_{1k} \cdot [K_j]_{k1}$$

En Python/Numpy, on peut effectuer :

— un produit matrice \times vecteur par l'instruction `np.dot(M, x)` ou la notation opératoire `M@x`.

Ici, pour que le produit soit possible, `M, x` sont des `ndarray` avec `M.shape=(m, n)`, `x.shape=(n,)` et on a `(np.dot(M, x)).shape = (m,)`

— un produit matrice \times matrice par l'instruction `np.dot(A, B)` ou la notation opératoire `A@B`.

Ici, pour que le produit soit possible, `A, B` sont des `ndarray` avec `A.shape[1] = B.shape[0]`, *i.e.* `A.shape=(m, p)`, `B.shape=(p, n)` et on a `(np.dot(A, B)).shape = (m, n)`

```
z = A @ (x + y); z = np.dot(A, x + y)
D = A @ B @ C ; D = np.dot(A, np.dot(B, C))
```

Matrices à coefficients dans \mathbb{K}^n , transposition, symétrie, antisymétrie

Si $A \in \mathcal{M}_{n,p}(\mathbb{K})$ (une matrice de taille $n \times p$) alors sa *transposée* $A^\top \in \mathcal{M}_{p,n}(\mathbb{K})$ (matrice de taille $p \times n$) est définie par :

$$\forall (j, i) \in \{1, \dots, p\} \times \{1, \dots, n\}, [A^\top]_{ji} = [A]_{ij}$$

Pour une telle matrice, les matrices produit $A.A^\top$ et $A^\top.A$ sont toutes les deux bien définies (quelles sont les tailles de ces matrices), et lorsque $n = p$, n'ont aucune raison d'être égales.

```
At = A.T #Transposition
MtM = M.T @ M ; MMt = M @ M.T #Quelles tailles ?
```

Une matrice A est dite *symétrique*, resp. *anti-symétrique* si $A = A^\top$, resp. $A = -A^\top$.

Concernant les opérations $+$ et $.$, l'application de transposition $(.)^\top : A \mapsto A^\top$ est linéaire :

$$\forall A, B, \forall \lambda, \mu \in \mathbb{K}, (\lambda.A + \mu.B)^\top = \lambda.A^\top + \mu.B^\top$$

Concernant les produits, faire attention au changement de sens :

$$\forall A, B, (A.B)^\top = B^\top.A^\top$$

Matrices à coefficients dans \mathbb{K}^n , Propriétés fondamentales de la multiplication matricielle

On peut commencer par noter qu'en général, même si l'opération est bien définie, il n'y a pas de raison que $A.B = B.A$. Cela arrive **parfois**, et alors A et B sont forcément carrées et de même taille n . Dans ce cas, on dit que A et B *commutent*.

Dans les énoncés abrégés suivants, les tailles des matrices A, B, C, \dots sont de sorte que toutes les opérations sont bien définies, l'ordre dans lequel sont effectuées les opérations (c'est à dire le système de parenthésage) est primordial, ainsi que l'ordre des opérandes.

— (Distributivité à gauche et à droite sur l'addition) :

$$\forall A, B, C, A.(B + C) = (A.B) + (A.C) \text{ et } \forall A, B, C, (A + B).C = (A.C) + (B.C)$$

— (Associativité /multiplication extérieure) :

$$\forall A, B, \forall \lambda \in \mathbb{K}, A.(\lambda.B) = (\lambda.A).B = \lambda.(A.B)$$

— Neutre à gauche, neutre à droite,

$$\forall A, I_n.A = A, A.I_p = A$$

— Nul à gauche, nul à droite,

$$\forall A, O_{mn}.A = O_{mp}, A.O_{pm} = O_{nm}$$

— Associativité⁴ :

$$\forall A, B, C, (A.B).C = A.(B.C)$$

4. C'est assez technique à démontrer ! Réfléchir aux tailles des matrices en jeu

Matrices carrées de taille n : la multiplication matricielle est une opération interne sur $\mathcal{M}_n(\mathbb{K})$.

Lorsque A et B sont $\mathcal{M}_n(\mathbb{K})$, les deux produit $C = A.B$ et $D = B.A$ sont eux-aussi dans $\mathcal{M}_n(\mathbb{K})$. La matrice identité I_n est élément neutre de cette multiplication :

$$\forall A \in \mathcal{M}_n(\mathbb{K}), I_n.A = A.I_n = A$$

et 0_n la matrice carrée nulle de taille n est « absorbante » :

$$\forall A \in \mathcal{M}_n(\mathbb{K}), 0_n.A = A.0_n = 0_n$$

On peut considérer, pour un entier positif p , la puissance p -ième de A définie de manière informelle par

$$A^p = \underbrace{A \dots A}_{p \text{ fois}}$$

et de manière plus sérieuse, par la récurrence de type géométrique :

$$A^0 = I_n \text{ et } \forall p \in \mathbb{N}, A^{p+1} = A^p.A$$

On démontre aussi (par récurrence sur q à p fixé) que :

$$\forall p, q \in \mathbb{N}, A^p.A^q = A^{p+q}$$

et donc que pour tout $p, q \in \mathbb{N}$, A^p et A^q commutent.

Polynômes de matrices⁵ : Si P est un polynôme à coefficients dans \mathbb{K} , $P = \sum_{p=0}^d \alpha_p X^p$, on peut considérer la matrice $P(A) = \sum_{p=0}^d \alpha_p.A^p$. On a les résultats de commutation suivants :

- Si P et Q sont deux polynômes, $P(A).Q(A) = Q(A).P(A) = (P.Q)(A)$.
- Si P et Q sont deux polynômes, $A.B = B.A$, $P(A).Q(B) = Q(B).P(A)$.

Les identités polynomiales classiques (*a.k.a* les identités remarquables) sont valables lorsque l'on a affaire à des matrices qui commutent :

$$\forall p \in \mathbb{N}, \forall A, B \in \mathcal{M}_n(\mathbb{K}), A.B = B.A \Rightarrow \sum_{k=0}^p \binom{p}{k} A^k . B^{p-k} \quad (\text{Bin. NEWTON})$$

et

$$\forall p \in \mathbb{N}, \forall A, B \in \mathcal{M}_n(\mathbb{K}), A.B = B.A \Rightarrow A^p - B^p = (A - B) \left(\sum_{k=1}^p A^{p-k} . B^k \right) \quad (\text{Séries géométriques})$$

Matrices inversibles : Une matrice $A \in \mathcal{M}_n(\mathbb{K})$ est dite inversible, si il existe une $B \in \mathcal{M}_n(\mathbb{K})$ telle que :

$$A.B = B.A = I_n$$

Si A est inversible, la matrice B vérifiant les identités ci-dessus est unique et on la note A^{-1} . L'ensemble des matrice inversibles de taille n est noté $\text{GL}_n(\mathbb{K})$.

5. Cette notion est censée être hors-programme depuis toujours en BCPST, vu sa récurrence dans les exercices de concours, on a va dire qu'elle y est

Quelques faits bien connus à propos de l'inversibilité :

- Une matrice $A \in \mathcal{M}_n(\mathbb{K})$ est inversible si et seulement si son rang (calculé par exemple par l'algorithme du pivot de GAUSS) ou par tout autre moyen vaut n ; On dispose d'un algorithme effectif de calcul de l'inverse via l'algorithme du pivot de GAUSS.
- A est inversible si et seulement si pour tout $y \in \mathbb{K}^n$, l'équation $A.x = y$ d'inconnue $x \in \mathbb{K}^n$ admet une unique solution. Cette solution est en fait $x = A^{-1}.y$, ce qui permet d'identifier A^{-1} .
- On démontre (c'est une conséquence du théorème du rang) que pour une matrice $A \in \mathcal{M}_n(\mathbb{K})$ quelconque, les trois énoncés suivants sont équivalents :

(i) A est inversible.

(ii) $\exists B \in \mathcal{M}_n(\mathbb{K}), A.B = I_n$.

(iii) $\exists B \in \mathcal{M}_n(\mathbb{K}), B.A = I_n$.

— Si $A \in \text{GL}_n(\mathbb{K})$, A et A^{-1} commutent.

— Si $A, B \in \text{GL}_n(\mathbb{K})$, $A.B \in \text{GL}_n(\mathbb{K})$ et $(A.B)^{-1} = B^{-1}.A^{-1}$.

En Python/Numpy,

— La commande `A**p` ne calcule pas A^p , elle calcule la matrice dont les entrées sont les puissances p des entrées de A ;

— Pour calculer A^p , il s'agit d'écrire une boucle calculant les termes d'une suite récurrente :

```
def PuissanceMatricielle(A, p):
    """ retourne A^p au sens matriciel """
    n = A.shape[0] #On suppose A carrée nxn
    Ap = np.eye(n)
    for _ in range(p): Ap = Ap @ A
    return Ap
```

— La fonction `inv` du module `numpy.linalg` calcule l'inverse d'une matrice carrée A .

```
A1 = inv(A)
A = P @ D @ inv(P) #Changement de base
```

Matrices à coefficients dans \mathbb{K}^n , le noyau et l'image d'une matrice, le théorème du rang

On retourne au cas général des matrices de taille $n \times p$. Soit $A \in \mathcal{M}_{n,p}(\mathbb{K})$.

Deux sous-espaces vectoriels (respectivement de \mathbb{K}^p et de \mathbb{K}^n) importants sont associés à la matrice A , son *noyau* et son *image* :

— le noyau de A , noté $\text{Ker}(A)$ est défini (équation cartésienne) par

$$\text{Ker}(A) = \{x \in \mathbb{K}^p \mid A.x = 0\}.$$

— l'image de A , notée $\text{Im}(A)$ est définie (équation paramétrique) par

$$\text{Im}(A) = \{A.x \mid x \in \mathbb{K}^p\} = \{y \in \mathbb{K}^n \mid \exists x \in \mathbb{K}^p, y = A.x\}.$$

D'un point de vue pratique :

— L'algorithme de GAUSS permet de déterminer une *base* de $\text{Ker}(A)$, en résolvant le système linéaire homogène de matrice A , à savoir $A.x = 0$ d'inconnue $x \in \mathbb{K}^p$. En juxtaposant les vecteurs de cette base pour former les colonnes d'une matrice B , on obtient une description paramétrique $\text{Ker}(A) = \text{Im}(B)$.

- Le même algorithme de GAUSS permet de déterminer un système (linéaire homogène) d'équations cartésiennes de $\text{Im}(A)$, ce sont les équations de compatibilité obtenues en résolvant le système linéaire $A.x = y$ d'inconnue $x \in \mathbb{K}^p$, de paramètre $y \in \mathbb{K}^n$.

En résolvant ce système de matrice B , on peut obtenir une base de $\text{Im}(A) = \text{Ker}(B)$.

- L'algorithme de GAUSS, bien compris, permet la définition de la dimension $\dim E$ d'un sous-espace E de \mathbb{K}^m comme étant le cardinal d'une famille libre et génératrice de ce sous-espace E (en d'autres termes, d'une base de E).
- Le rang d'une matrice A est la dimension de $\text{Im}(A)$.
- Le *théorème* du rang dit que pour toute matrice A comportant p colonnes,

$$\dim \text{Ker}(A) + \dim \text{Im}(A) = p$$

Du point de vue systèmes linéaires échelonnés, il s'agit simplement de constater que l'ensemble des inconnues (scalaires) d'un système linéaire homogène échelonné $A.x = 0$ d'inconnue vectorielle $x \in \mathbb{K}^p$ se partitionne en

1. l'ensemble des inconnues secondaires (au nombre de $\dim \text{Ker}(A)$)
2. l'ensemble des inconnues primaires (au nombre de $\dim \text{Im}(A)$)

La mise en place informatique de l'algorithme de GAUSS permettant le calcul d'une base de chacun des espaces $\text{Ker}(A)$ et $\text{Im}(A)$ est l'objet du TP 8 d'informatique.

Matrices à coefficients dans \mathbb{K}^n , application linéaire associée

- Etant donnée une matrice $A \in \mathcal{M}_{n,p}(\mathbb{K})$, on peut définir une application $a : \mathbb{K}^p \rightarrow \mathbb{K}^n$ qui est dite « canoniquement associée » à A . L'application a est définie par

$$\forall x \in \mathbb{K}^p, a(x) = A.x$$

Cette application est linéaire et $\text{Ker}(A) = \text{Ker}(a)$, $\text{Im}(A) = \text{Im}(a)$.

- Etant données deux matrices $A \in \mathcal{M}_{n,p}(\mathbb{K})$, $B \in \mathcal{M}_{p,m}(\mathbb{K})$, les applications $a : \mathbb{K}^p \rightarrow \mathbb{K}^n$, $b : \mathbb{K}^m \rightarrow \mathbb{K}^p$ « canoniquement associées » à A et B , l'application $c = a \circ b$ canoniquement associée à la matrice $C = A.B$.
- L'application nulle $\mathbb{K}^p \rightarrow \mathbb{K}^n$ est canoniquement associée à la matrice nulle 0_{np} .
- L'application id , l'« identité » de \mathbb{K}^n définie par : $\forall x \in \mathbb{K}^n, \text{id}(x) = x$ est canoniquement associée à la matrice I_n .
- Etant données une matrice $A \in \text{GL}_n$ et l'application $a : \mathbb{K}^n \rightarrow \mathbb{K}^n$ canoniquement associée à A , a est bijective et l'application a^{-1} est canoniquement associée à la matrice A^{-1} .

Réciproquement, étant donnée une application linéaire $a : \mathbb{K}^p \rightarrow \mathbb{K}^n$, il existe une matrice $A \in \mathcal{M}_{n,p}(\mathbb{K})$ telle que a est canoniquement associée à A . Il suffit de prendre pour A la matrice dont la j -ième colonne est $A.e_j$ où e_j est le j -ième vecteur de la base canonique de \mathbb{K}^p (cf. premier §).

Exercice 1.—Pour $\theta \in \mathbb{R}$, on définit $M(\theta) = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix}$

1. Calculer $M(\theta).M(\theta')$ et en déduire une expression simple de $M(\theta)^n$ pour $n \in \mathbb{N}$.

2. Montrer que $\forall \theta \in \mathbb{R}, M(\theta)$ est inversible et calculer son inverse.

3. Montrer que la relation trouvée au 1 est encore valable pour $n \in \mathbb{Z}$.

Exercice 2.— Caractérisation de l'inversibilité et inverse (lorsqu'elle existe) d'une matrice carrée d'ordre 2.

Exercice 3.—Soit $f(x) = \lambda.e^x \cos x + \mu.e^x \sin x$. Soit $x_0 \in \mathbb{R}$. Déterminer $\lambda, \mu \in \mathbb{R}$ pour que

$$f(x_0) = 1 \text{ et } f'(x_0) = 0$$

(On pourra établir le système linéaire 2×2 à résoudre et utiliser le déterminant.)

Exercice 4.— Montrer que $A = \begin{pmatrix} 2 & 1 & 4 \\ 3 & -1 & 0 \\ 1 & 1 & 4 \end{pmatrix}$ est inversible et calculer son inverse avec la méthode de GAUSS.

Exercice 5.—Prouver l'inversibilité et inverser, par la méthode du pivot de GAUSS, la matrice

$$A = \begin{pmatrix} 1 & -1 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & -1 \end{pmatrix}$$

Exercice 6.—Montrer que $A = \begin{pmatrix} 0 & 0 & 0 & 3 \\ 0 & -1 & 0 & 0 \\ 2 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$ est inversible (en regardant son rang) et calculer son

inverse en résolvant le système $A.X = Y$

Exercice 7.—Soit

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{pmatrix}$$

En écrivant $A = I_3 + N$ déterminer, pour $n \in \mathbb{N}$, une formule pour A^n .

Exercice 8.—

1. Soit N une matrice carrée $n \times n$, calculer et simplifier, pour $p \in \mathbb{N}^*$, $(I_n - N).(I_n + N + N^2 + \dots + N^p)$

2. Soit

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{pmatrix}$$

En écrivant $A = I_3 - N$, déterminer A^{-1} en utilisant la formule précédente.

Exercice 9.— On considère la matrice :

$$M = \begin{pmatrix} -3 & 4 & 2 \\ -2 & 3 & 1 \\ 2 & -2 & 0 \end{pmatrix}.$$

Calculer M^2 et montrer qu'on peut écrire M^2 comme combinaison linéaire de M et de I_3 . En déduire que M est inversible et donner son inverse.

Exercice 10.— Pour la matrice suivante, calculer A^3 et en déduire que A n'est pas inversible.

$$A = \begin{pmatrix} 2 & 1 & -4 \\ 0 & 1 & -2 \\ 1 & 1 & -3 \end{pmatrix}.$$

Exercice 11.— Calculer A^n avec :

$$A = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}.$$

Exercice 12.— Soit $A \in \mathcal{M}_n(\mathbb{K})$ vérifiant l'égalité matricielle

$$5A^3 - 4A - I_n = (0). \quad (\star)$$

Montrer que A est inversible et déterminer son inverse.

Exercice 13.— Soit $\lambda \in \mathbb{R}$. Appliquer la méthode du pivot de GAUSS pour résoudre le système d'inconnue $(x, y, z) \in \mathbb{R}^3$,

$$\begin{cases} (1 - \lambda)x & -y & +z & = 0 \\ -x & +(1 - \lambda)y & -z & = 0 \\ x & -y & +(1 - \lambda)z & = 0 \end{cases}$$

Exercice 14.— Déterminer les espaces $\text{Ker}(A)$ et $\text{Im}(A)$ (on donnera la dimension et une base de l'espace) dans les cas :

1.

$$A = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}$$

2.

$$A = \begin{pmatrix} -3 & 1 & 1 & 1 \\ 1 & -3 & 1 & 1 \\ 1 & 1 & -3 & 1 \\ 1 & 1 & 1 & -3 \end{pmatrix}$$

Exercice 15.— Pour $A \in \mathcal{M}_n(\mathbb{K})$, on appelle trace de A la somme des éléments diagonaux de A , c'est-à-dire :

$$\text{Si } A = (a_{i,j})_{1 \leq i, j \leq n} \text{ alors } \text{Tr}(A) = \sum_{i=1}^n a_{i,i}.$$

1. Montrer que pour toutes les matrices A et B carrées de taille n et tous les scalaires λ , $\text{Tr}(\lambda A + B) = \lambda \text{Tr}(A) + \text{Tr}(B)$.

2. Montrer que pour toutes les matrices A et B carrées de taille n , $\text{Tr}(AB) = \text{Tr}(BA)$. Indication: Appeler $C = A.B$, $D = B.A$ et écrire les formules abstraites de produit matriciel donnant les coefficients C_{ij} et D_{ij}

3. En déduire qu'il n'existe pas de matrices A et B carrées de taille n telles que $AB - BA = I_n$.

Matrices carrées à coefficients dans \mathbb{K} , éléments propres (BCPST2)

Définition 1. Si $A \in \mathcal{M}_n(\mathbb{K})$, $\lambda \in \mathbb{K}$. Le scalaire (nombre) λ est dit valeur propre de la matrice A s'il existe un vecteur $X \in \mathbb{K}^n$ non nul tel que

$$A.X = \lambda.X$$

Un tel vecteur est appelé vecteur propre de A associé à la valeur propre λ .

- L'ensemble des valeurs propres d'une matrice A s'appelle le spectre de A , il est noté $\text{Spec}(A)$.
- Pour $\lambda \in \mathbb{K}$, on note $E_\lambda(A)$, et on appelle ⁶ sous-espace propre de A associé à λ le noyau de $A - \lambda.I_n$:

$$\lambda \in \text{Spec}(A) \Leftrightarrow \dim \text{Ker}(A - \lambda.I_n) \geq 1$$

- Si (X_1, \dots, X_n) est une base de \mathbb{K}^n composée de vecteurs propres de A , en posant $P = (X_1 | \dots | X_n)$, la matrice P est inversible et $P^{-1}.A.P$ est une matrice diagonale.
- Une telle base « diagonalise » A et A est dite diagonalisable.

La fonction `eig` du module `numpy.linalg` calcule, si possible, les valeurs propres de A et une matrice inversible diagonalisant une matrice carrée A .

```
D, P = np.linalg.eig(A)
```

Exercice 16.— Pour $\theta \in \mathbb{R}$, on définit $M(\theta) = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix}$. Donner les valeurs propres complexes (c'est à dire en considérant que cette matrice est à coefficients complexes) de $M(\theta)$. Déterminer une base de \mathbb{C}^2 diagonalisant $M(\theta)$.

Quel est le spectre réel de $M(\theta)$?

Exercice 17.— Soit $A = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 2 & 0 \\ 3 & 3 & 3 \end{pmatrix}$

1. Donner les valeurs propres de A et déterminer une matrice P diagonalisant A .
2. Pour $n \in \mathbb{N}$, donner une formule pour A^n .

Exercice 18.— Soit $A = \begin{pmatrix} 4 & -1 \\ 2 & 1 \end{pmatrix}$.

1. Chercher les valeurs propres et les vecteurs propres de A .
2. A est-elle diagonalisable ?
3. Si oui, construire P la matrice de passage puis déterminer P^{-1} et donner $P^{-1}.A.P$.
4. Pour $n \in \mathbb{N}$ calculer A^n .

Exercice 19.— Diagonaliser $A_4 = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}$

Exercice 20.—

1. Soit $j = e^{i\frac{2\pi}{3}}$. Soit $B = \begin{pmatrix} 1 & 1 & 1 \\ 1 & j & j^2 \\ 1 & j^2 & j \end{pmatrix}$. Montrer que B est inversible et donner B^{-1} . (Calculer $B.\bar{B}$ comme déjà fait précédemment)

6. Certains ouvrages limitent cette terminologie au cas où $\dim E_\lambda \geq 1$.

2. Soit $(a_0, a_1, a_2) \in \mathbb{C}^3$ et

$$A = \begin{pmatrix} a_0 & a_1 & a_2 \\ a_2 & a_0 & a_1 \\ a_1 & a_2 & a_0 \end{pmatrix}$$

Calculer $A.B$ et en déduire que A est diagonalisable.

Exercice 21.—On suppose que A et B sont deux matrices semblables (rappeler la définition).

1. Montrer que si A est diagonalisable alors B l'est aussi.
2. Montrer que même en l'absence de diagonalisabilité, les spectres de A et B sont identiques et que les dimensions des espaces propres associés à une v.p λ de chacune sont égales.
3. A quelle condition deux matrices diagonalisables sont-elles semblables ?

Diagonalisabilité des matrices carrées (BCPST2)

Toutes les matrices (carrées) ne sont pas diagonalisables.

Théorème-Définition 2. Soit $A \in \mathcal{M}_n(\mathbb{K})$. On a l'équivalence entre :

- (i) Il existe une base de \mathbb{K}^n composée de vecteurs propres de A ;
- (ii) Il existe une matrice P dans $\text{GL}_n(\mathbb{K})$ (invertible donc) telle que $P^{-1}.A.P$ est diagonale.
- (iii) Il existe $\lambda_1, \dots, \lambda_K \in \mathbb{K}$, distincts tels que

$$\sum_{k=1}^K \dim \text{Ker}(A - \lambda_k \cdot I_n) \geq n$$

Dans l'éventualité de l'une des propriétés équivalentes ci-dessus, on dit que A est diagonalisable sur \mathbb{K} .

Remarques :

1. les propriétés (i) et (ii) sont deux formulations du même phénomène. On passe de la base $\mathcal{X} = (X_1, \dots, X_n)$ de la propriété (i) à la matrice invertible P de la propriété (ii) en identifiant :

$$P = (X_1 | \dots | X_n)$$

C'est à dire que P est la matrice de passage de la base canonique de \mathbb{K}^n à la base \mathcal{X} de \mathbb{K}^n .

2. La propriété (iii) est souvent désignée dans les livres ou les cours comme « **Condition nécessaire et suffisante de diagonalisabilité** ». Son équivalence avec (i) ou (ii) est le morceau de choix des démonstrations du cours sur la diagonalisabilité. Il est basé sur la proposition fondamentale :

Proposition 3. Si $A \in \mathcal{M}_n(\mathbb{K})$, $\lambda_1, \dots, \lambda_K \in \text{Spec}_{\mathbb{K}}(A)$, distinctes, associées respectivement aux vecteurs propres X_1, \dots, X_K alors la famille X_1, \dots, X_K est libre dans \mathbb{K}^n .

3. Un critère suffisant (mais très loin d'être nécessaire) pour la diagonalisabilité d'une matrice A est :

Proposition 4. Si $A \in \mathcal{M}_n(\mathbb{K})$ et A admet n valeurs propres distinctes $\lambda_1, \dots, \lambda_n \in \mathbb{K}$ alors A est diagonalisable sur \mathbb{K} .

4. Dans une autre direction, un critère suffisant qu'il est nécessaire de connaître est le « Théorème spectral » (cf. Chapitre **Produit scalaire**) :

Théorème 5 (Théorème spectral, cas réel). Si $A \in \mathcal{M}_n(\mathbb{R})$ et A est symétrique ($A^\top = A$) alors A est diagonalisable sur \mathbb{R} (via une base orthonormée).

Valeurs propres et polynômes de matrices (Exercices BCPST2)

Cet énoncé n'est pas au programme officiel de BCPST2 mais est redémontré, puis utilisé dans nombre d'exercices.

Proposition 6. Soit $A \in \mathcal{M}_n(\mathbb{K})$, $P \in \mathbb{K}[X]$.

1. Si $\lambda \in \text{Spec}_{\mathbb{K}}(A)$ alors pour tout $k \in \mathbb{N}$, $\lambda^k \in \text{Spec}_{\mathbb{K}}(A^k)$;
2. Si $\lambda \in \text{Spec}_{\mathbb{K}}(A)$ alors $P(\lambda) \in \text{Spec}_{\mathbb{K}}(P(A))$;
3. Si $P(A) = 0$, alors $P(\lambda) = 0$.

Exercice 22.—Démontrer ces trois points.

Exercice 23.—On pose $J \in \mathcal{M}_n(\mathbb{C})$ la matrice comportant des zéros partout sauf :

- dans le coin en haut à droite : $[J]_{1n} = 1$,
- sur la première sous-diagonale : $\forall k \in \{1, \dots, n-1\}, [J]_{k+1,k} = 1$.

1. Montrer que $J^n = I_n$. (On pourra calculer l'image de chaque vecteur de la base canonique de \mathbb{C}^n par J et, une fois la liste complète dressée, en déduire l'image de chacun par J^n .)
2. En déduire un ensemble de n valeurs propres complexes possibles de J
3. Montrer que chacune de ces valeurs est effectivement valeur propre de J en exhibant un vecteur propre associé.