Chapitre 17 : Applications linéaires

PTSI B Lycée Eiffel

30 mars 2020

J'ai simplement pensé à l'idée d'une projection, d'une quatrième dimension invisible, autrement dit que tout objet de trois dimensions, que nous voyons froidement, est une projection d'une chose à quatre dimensions, que nous ne connaissons pas.

Marcel Duchamp

Un mathématicien et un ingénieur assistent à une conférence sur les processus physiques intervenant dans les espaces de dimension 9. *Le mathématicien est assis et apprécie beaucoup la conférence, pendant que l'ingénieur fronce les sourcils et semble complètement embrouillé. À la fin, l'ingénieur demande au matheux : « Comment fais-tu pour comprendre tout cela ? » « C'est simple ! D'abord tu visualises le processus en dimension* n*, et ensuite il suffit de prendre* $n = 9$*. »*

Introduction

Ce deuxième chapitre d'algèbre linéaire sera un simple complément du premier, permettant de présenter une notion complètement fondamentale, celle d'application linéaire, qui va éclairer d'un jour nouveau tous les termes vus depuis le début de l'année et faisant intervenir ce fameux mot « linéaire ». En gros, les applications linéaires sont des applications (des fonctions, quoi) « naturelles » dans les espaces vectoriels, qui apparaissent dans tous les domaines des mathématiques, et pour lesquels une étude tout à fait générale et théorique est possible, ce qui permet d'appréhender un peu mieux la puissance de l'algèbre linéaire pour résoudre des problèmes de maths très divers. Ce petit chapitre sera essentiellement constitué de vocabulaire, les quelques calculs à savoir faire se résumant à nouveau la plupart du temps à des résolutions de petits systèmes (linéaires, cela va de soit !).

Objectifs du chapitre :

- maitriser tout le vocabulaire introduit dans ce chapitre.
- comprendre l'intérêt du théorème du rang.
- savoir manipuler les projections et symétries vectorielles.

1 Vocabulaire

1.1 Morphismes

Définition 1. Soient E et F deux espaces vectoriels, une **application linéaire** de E dans F est une application $f : E \to F$ vérifiant les conditions suivantes :

- $\forall (u,v) \in E^2$, $f(u+v) = f(u) + f(v)$
- $\forall \lambda \in \mathbb{R}$, $\forall u \in E$, $f(\lambda u) = \lambda f(u)$

Remarque 1. Autrement dit, une application linéaire est une application compatible avec les deux opérations définissant la structure d'espace vectoriel. Pour les plus curieux, voir le développement de ce principe dans le paragraphe suivant (celui qui est complètement hors-programme). Une application linéaire « transforme les sommes en sommes » et les produits pas des réels en produits par des réels, même si les espaces vectoriels E et F sont de nature complètement différente, et même si les opérations de somme dans E et dans F ne sont techniquement pas « les mêmes ».

Remarque 2. Si $f : E \to F$ est une application linéaire, on a nécessairement $f(0_E) = 0_F$ (exceptionnellement, j'ai précisé ici dans quel espace vectoriel se situait le vecteur nul car c'est très important). En effet, $f(0+0) = f(0) + f(0)$, donc $f(0) = 2f(0)$, ce qui implique manifestement $f(0) = 0$.

Proposition 1. Une application $f : E \to F$ est linéaire si et seulement si $\forall (\lambda, \mu) \in \mathbb{R}^2$, $\forall (u,v) \in E^2$, $f(\lambda u + \mu v) = \lambda f(u) + \mu f(v)$.

Démonstration. Si f vérifie les conditions de la définition, alors $f(\lambda u + \mu v) = f(\lambda u) + f(\mu v) = \lambda f(u) + \mu f(v)$ en utilisant successivement les deux propriétés. Réciproquement, en prenant $\lambda = \mu = 1$, on retrouve la première condition ; et en prenant $v = 0$, on retrouve la deuxième (en exploitant la remarque faite juste avant l'énoncé de la propriété). □

Remarque 3. Autrement dit, une application est linéaire si et seulement si elle est compatible avec les combinaisons linéaires. On a d'ailleurs plus généralement, pour une application linéaire, $f\left(\sum_{i=1}^{k} \lambda_i e_i\right) = \sum_{i=1}^{k} \lambda_i f(e_i)$. C'est d'ailleurs cette propriété (avec seulement deux vecteurs, pas besoin de se compliquer la vie) qu'on vérifiera lorsqu'il faudra prouver dans les exercices qu'une application est linéaire : on vérifie bien que l'application est définie sur E et à valeurs dans F, et on calcule $f(\lambda u + \mu v)$ en essayant de prouver que c'est égal à $\lambda f(u) + \mu f(v)$. Attntion bien sûr à faire attention au fait qu'ici u et v sont des vecteurs de l'espace E, pas de simples coordonnées.

Exemple : Dans l'espace vectoriel $E = \mathcal{M}_2(\mathbb{R})$, on pose $A = \begin{pmatrix} 2 & 1 \\ -1 & 1 \end{pmatrix}$, et on définit l'application $f : E \mapsto E$ par $f(M) = AM$. Pour prouver que cette application est linéaire, même pas besoin de calculer les coordonnées explicites de $f(M)$, on se contente d'écrire que, si M et N sont deux matrices quelconques de E, et $(\lambda, \mu) \in \mathbb{R}^2$, alors $f(\lambda M + \mu N) = A(\lambda M + \mu N) = \lambda AM + \mu AN = \lambda f(M) + \mu f(N)$. On a simplement utilisé les règles de base du calcul matriciel (si on veut faire savant, on a en fait simplement utilisé la linéarité du produit matriciel à gauche, ce qui signifie en fait exactement que l'application f est linéaire).

Exemples : On pourrait penser que les conditions définissant une application linéaire sont restrictives, et qu'il va donc y avoir « peu » d'applications linéaires. C'est en partie vrai : si on se restreint à des espaces très simples (et de très petite dimension), il y a effectivement très peu d'applications qui seront linéaires. Mais dans des espaces plus gros, il y en aura largement assez pour qu'on s'y intéresse de très près, d'autant plus que l'ensemble de ces applications linéaires va être doté d'une structure forte, justement à cause des conditions très restrictives qui les définissent.

- Si on considère toutes les applications $f : \mathbb{R} \to \mathbb{R}$ (autrement dit toutes les fonctions réelles définies sur $\mathbb{R}$ tout entier), les seules à être linéaires sont celles de la forme $x \mapsto ax$, avec $a \in \mathbb{R}$. Ce qui tombe d'ailleurs très bien puisque ce sont justement les fonctions qu'on appelle traditionnellement fonctions linéaires !
- L'application $f : \mathbb{R}^2 \to \mathbb{R}^3$ définie par $f(x,y) = (2x - 3y, 4x + y, -x + 2y)$ est une application linéaire. On le vérifie comme ci-dessus en développant $f(\lambda(x,y,z) + \mu(x',y',z'))$ et en constatant que c'est égal à $\lambda f(x,y,z) + \mu f(x',y',z')$ (ici, il faut vraiment faire le calcul avec les coordonnées, c'est un peu pénible mais essentiellement trivial).

- L'application $f : \mathbb{R}^2 \to \mathbb{R}^3$ définie par $f(x,y) = (2x-3, 4+y, -x+2y)$ n'est pas une application linéaire (on peut constater par exemple qu'en général $f(2x,2y) \neq 2f(x,y)$). En fait, en réfléchissant un peu, on constate assez facilement que les seules applications linéaires de $\mathbb{R}^2$ dans $\mathbb{R}^3$ sont celles pour lesquelles les coordonnées de $f(x,y)$ sont obtenues en faisant des combinaisons linéaires de x et de y. Ce constat s'étend en fait aux applications linéaires entre n'importe quels espaces de dimension finie. Il faut et il suffit que chaque coordonnée de $f(u)$ soit une combinaison linéaire des coordonnées de u pour que f soit une application linéaire.
- L'application $f : \mathcal{M}_3(\mathbb{R}) \to \mathcal{M}_3(\mathbb{R})$ définie par $f(M) = AM$ est une application linéaire, quelle que soit la matrice $A \in \mathcal{M}_3(\mathbb{R})$ (même principe que dans l'exemple ci-dessus).
- L'application $f : \mathcal{M}_3(\mathbb{R}) \to \mathcal{M}_3(\mathbb{R})$ définie par $f(M) = M^2$ n'est pas une application linéaire (en général, $(M+N)^2 \neq M^2 + N^2$, ce qu'on peut résumer en disant justement que l'élévation au carré « n'est pas linéaire »).
- Soit E l'ensemble des suites réelles. L'application $f : E \to \mathbb{R}^3$ définie par $f(u_n) = (u_0, u_8, u_{35})$ est une application linéaire.
- Soit E l'ensemble des fonctions C^∞ de $\mathbb{R}$ dans $\mathbb{R}$. L'application $f : E \to E$ définie par $f(g) = g'$ est une application linéaire (dont la variable est une fonction !). J'ai d'ailleurs déjà du dire au moins dix fois depuis le début de l'année la phrase « La dérivation est linéaire », ce qui signifie exactement cela.
- Soit E l'ensemble des fonctions continues sur l'intervalle $[0;1]$. L'application $f : E \to \mathbb{R}$ définie par $f(g) = \displaystyle\int_0^1 g(t)\,dt$ est une application linéaire. Là encore, on a parlé de linéarité de l'intégrale largement avant d'avoir une définition rigoureuse de ce qu'est une application linéaire.
- L'application $f : \mathbb{R}_n[X] \to \mathbb{R}_n[x]$ définie par $f(P) = 2X^2P'' - XP'$ est une application linéaire (ici, il faut bien faire attention quand on vérifie la linéarité à vérifier que $f(\lambda P + \mu Q) = \lambda f(P) + \mu f(Q)$, il n'y absolument aucune raison de mettre des coefficients λ et μ sur l'indeterminée X à l'intérieur du polynôme, ce qui fait que le X^2 qui est simplement en facteur de P'' ne perturbe pas du tout la linéarité de l'application).

Définition 2. Une application linéaire $f : E \to F$ est aussi appelée **morphisme** de E dans F. On note $\mathcal{L}(E,F)$ l'ensemble de toutes les applications linéaires de E dans F.
Une application linéaire $f : E \to E$ est appelée **endomorphisme** de l'espace vectoriel E. On note plus simplement $\mathcal{L}(E)$ l'ensemble des endomorphismes de E.
Une application linéaire bijective est appelée **isomorphisme**.
Un endomorphisme bijectif est appelée **automorphisme**. L'ensemble des automorphismes d'un espace vectoriel E est noté $GL(E)$.

Remarque 4. Tout ce vocabulaire est assez logique si on a quelques rudiments de grec (mais oui). Le terme morphisme découle du grec $\mu o \rho \varphi \eta$ qui signifie forme. Un morphisme est donc une application qui « respecte la forme » de l'ensemble, c'est-à-dire ici qui respecte la structure d'espace vectoriel des deux ensembles (au départ et à l'arrivée). Il existe en fait des morphismes pour tous types de structures algébriques sur les ensembles, qui seront définis en lien avec la structure. On devrait donc parler non pas simplement de morphismes dans ce chapitre, mais bien de « morphismes d'espaces vectoriels » puisqu'il existe aussi des morphismes de groupes, des morphismes d'algèbres et autres joyeusetés que vous ne devez absolument pas maitriser (cf plus bas pour les curieux).

Les préfixes ajoutés devant le terme morphisme sont tout autant dérivés du grec. Le préfixe endo signifie intérieur, un endomorphisme est donc un morphisme où on reste dans le même espace vectoriel (tout comme un endogame est quelqu'un qui se marie avec une personne de la même catégorie (typiquement socio-professionnelle) que lui). Vous devez déjà avoir croisé le préfixe iso qui signifie « même », surtout si vous avez déjà fait de la thermodynamique avec M.Raimi (isobares, isochores et autres trucs sympathiques ; sinon le terme isotherme est d'un emploi nettement plus fréquent). Ici, un isomorphisme est une application qui permet de comprendre que les espaces vectoriels E et F

ont la « même » structure. Bien sûr les ensembles ne sont pas les mêmes, et les opérations non plus, mais le fait d'avoir un isomorphisme entre E et F permet de les voir comme des espaces qui « se ressemblent » énormément. Ainsi, ils auront forcément la même dimension (s'ils sont de dimension finie), l'image de n'importe quelle base de E par l'isomorphisme f sera une base de F, etc.

Proposition 2. Si E et F sont deux espaces vectoriels réels, c'est aussi le cas de $\mathcal{L}(E, F)$.

Démonstration. La somme de deux applications linéaires est linéaire, l'élement neutre étant l'application nulle, et l'opposé d'une application linéaire étant toujours défini. De plus, les produits par des constantes d'applications linéaires sont linéaires, et les relations de distributivité sont immédiates. Bref, tout est trivial. □

Remarque 5. C'est cette structure d'espace vectoriel sur l'ensemble des applications linéaires de E dans F (attention, il s'agit d'un espace complètement différent de E et de F) qui nous permettra dans le dernier chapitre d'algèbre linéaire que nous ferons cette année de représenter les applications linéaires entre espaces vectoriels de dimension finie par des matrices, et (encore mieux !) d'utiliser toute la puissance du calcul matriciel pour étudier plus efficacement les applications linéaires.

1.2 Compléments hors-programme

Ce paragraphe n'est à lire que si vous voulez comprendre un peu mieux la notion de morphisme et surtout d'isomorphisme qui est omniprésente en algèbre (et pas seulement en algèbre linéaire puisque, comme expliqué plus haut, il n'existe pas que des morphismes entre espaces vectoriels). Comme j'ai essayé de vous le faire comprendre plus haut, l'idée derrière la notion de morphisme est de créer des applications qui préservent la structure (quelle qu'elle soit), et la notion d'isomorphisme permet en fait d'identifier deux ensembles ayant une structure identique. On peut même aller plus loin en essayant de comprendre quelles sont les différentes façons de structurer un ensemble « à partir de rien ». Pour cela il est plus facile de travailler avec des ensembles « petits » (ici on prendra des ensembles finis à peu d'éléments) et une structure plus basique que celle d'espace vectoriel : celle de **groupe**.

C'est en fait assez simple : définir une structure de groupe sur un ensemble E consiste simplement à créer une opération dans l'ensemble E (qu'on notera ici $\star$ car il ne s'agit pas forcément d'une addition) qui vérifie les deux propriétés suivantes :

- il existe un élément de E qui joue le rôle d'élément neutre pour l'opération $\star$, c'est-à-dire un élément $e \in E$ tel que, $\forall x \in E$, $e \star x = x$.
- tout élément de E admet un **inverse** pour l'opération $\star$, c'est-à-dire que, si $x \in E$, il existe $y \in E$ tel que $x \star y = e$, où e est l'élément neutre de ce qu'on appellera désormais le groupe E.

On ajoutera en plus le fait que l'opération $\star$ doit être associative et commutative (et encore, la commutativité n'est même pas obligatoire), et c'est tout. La question est maintenant, si on prend un ensemble fini contenant un certain nombre d'éléments, de savoir le nombre d'opérations différentes qu'on peut créer (et donc le nombre de structures de groupe différentes qu'on peut mettre sur notre ensemble) en respectant ces conditions. Le tout **à isomorphisme près**, ce qui revient à dire que, si on définit une nouvelle opération qui est en fait la même qu'une opération précédemment définie à permutation des éléments près, ça ne compte pas. Donnons des exemples concrets pour éclairer tout ça. Pour représenter une opération sur un ensemble fini, le plus simple est de présenter sa **table**, similaire à une bête table de multiplication, où on met en lignes et en colonnes les éléments de l'ensemble et on indique dans les cases du tableau le résultat de l'opération associant deux éléments de l'ensemble. Par exemple :

	e	a	b	c
e	e	a	b	c
a	a	e	c	b
b	b	c	e	a
c	c	b	a	e

On dispose ici d'un ensemble contenant quatre éléments notés e (qui jouera le rôle d'élément neutre), a, b et c. Et on a décidé que l'opération $\star$ vérifierait que $a \star a = e$, $a \star b = c$ etc (on lit les résultats dans la table ci-dessus). Je vous laisse constater si vous avez du temps à perdre que l'opération ainsi définie est bien associative et commutative, que e en est un élément neutre et que chaque élément admet un inverse (en l'occurrence, ici, chaque élément est son propre inverse). La question qu'on se pose maintenant est : aurait-on pu décider autrement et créer d'autres opérations qui fonctionnent ? Si on décide par exemple de permuter les rôles des quatre éléments (on décide que a devient l'élément neutre, donc $a \star b = b$, etc, et par contre $e \star c = b$ et ainsi de suite), la structure sera la même, ça ne compte pas (on peut prouver qu'il existe un isomorphisme de groupe entre notre ensemble muni de la première opération et le même ensemble muni de la deuxième opération). En fait il existe bel et bien une deuxième solution qui **ne peut pas** être obtenue à partir de la première par simple permutation des variables :

	e	a	b	c
e	e	a	b	c
a	a	b	c	e
b	b	c	e	a
c	c	e	a	b

Il ne s'agit pas de la même structure car ici, contrairement au tableau précédent, il existe des éléments qui ne sont pas leur propre inverse (a et c sont inverses l'un de l'autre, e et b sont leur propre inverse). On peut prouver (mais ce n'est pas évident !) qu'il n'y a en fait que deux structures de groupes différentes sur un ensemble contenant quatre éléments (les deux que nous venons de citer). Si notre ensemble contient un nombre d'éléments qui est un nombre **premier**, c'est encore pire, il n'y a qu'une seule façon de structurer l'ensemble pour en faire un groupe ! D'ailleurs, vous la connaissez déjà, cette structure, sans même le savoir : c'est la structure de l'ensemble $\mathbb{U}_n$ des racines n-èmes de l'unité, muni de l'opération de multiplication (je vous laisse y réfléchir). Pour un nombre d'éléments non premier, il existe en général plusieurs structures possibles, et bien sûr il peut y en avoir beaucoup si le nombre d'éléments est élevé, et encore plus si on accepte les opérations non commutatives. On connait de toute façon bien les structures de groupes sur tous les ensembles finis mais les démonstrations de ces résultats sont assez monstrueuses (elles prennent plusieurs **milliers** de pages, les plus curieux iront par exemple consulter la page Wikipédia « Liste des groupes finis simples », mais n'y comprendront probablement pas grand chose). Plus accessible, la « Liste des petits groupes » toujours disponible sur Wikipédia vous donne le nombre de structures possibles pour des nombres d'éléments inférieurs ou égaux à 20. On a par exemple pas moins de 14 opérations différentes (dont neuf ne sont pas commutatives) si notre ensemble contient 16 éléments.

Un dernier exemple quand même un peu plus tordu que les précédents, puisqu'il s'agit d'une structure de groupe non commutatif, qui a pourtant une origine simple et géométrique. Prenez un beau triangle équilatéral (dessinez-le sur votre feuille si vous voulez) ABC. On s'intéresse aux isométries du plan laissant stable le triangle équilatéral (autrement dit, on va déplacer ou symétriser notre triangle, mais à la fin on doit toujours avoir un triangle équilatéral à la même place). Avec un peu de motivation, on peut prouver qu'il n'existe que six transformations géométriques laissant effectivement stable notre triangle équilatéral :

- l'application identité, notée i, qui ne fait rien bouger.

- la rotation par rapport au centre O du triangle d'angle $\frac{2\pi}{3}$ (qui permute les trois sommets, et donc les trois côtés, du triangle), qu'on notera r_1.
- la rotation par rapport au centre O du triangle d'angle $\frac{4\pi}{3}$, qu'on notera r_2.
- la réflexion par rapport à la droite (AI), où I est le milieu du côté $[BC]$ opposé à A. On notera cete réflexion s_1.
- de même, la réflexion par rapport à la droite (BJ), où J est le milieu du côté $[AC]$. On notera cette réflexion s_2.
- la réflexion par rapport à la droite (CK), où K est le milieu du côté $[AB]$. On notera cette réflexion s_3.

On dispose donc d'un ensemble à six éléments, sur lequel une opération très naturelle va créer une structure de groupe : l'opération de composition (celle qu'on note $\circ$). Et c'est logique : la composée de deux applications laissant stable notre triangle continuera à le laisser stable, on a un élément neutre évident qui est l'identité i, et chaque application a une réciproque (c'est elle qui joue le rôle d'inverse pour l'opération de composition) qui est dans la liste puisqu'elle va elle-même laisser le triangle stable. Si on écrit la table complète de l'opération $\circ$ sur cet ensemble, on obtient :

	i	r_1	r_2	s_1	s_2	s_3
i	i	r_1	r_2	s_1	s_2	s_3
r_1	r_1	r_2	i	s_2	s_3	s_1
r_2	r_2	i	r_1	s_3	s_1	s_2
s_1	s_1	s_3	s_2	i	r_2	r_1
s_2	s_2	s_1	s_3	r_1	i	r_2
s_3	s_3	s_2	s_1	r_2	i	r_1

Attention, la lecture de ce tableau est plus compliquée que pour les précédents puisque l'opération n'est pas commutative : le résultat donné est celui obtenu en composant l'élément de la ligne (à gauche) par l'élément de la colonne (à droite). Ainsi, on a par exemple $r_2 \circ s_1 = s_3$ mais $s_1 \circ r_2 = s_2$. Cette structure est en fait la première structure de groupe non commutatif sur un ensemble fini (on ne peut pas en créer sur des ensembles à moins de six éléments). Les mathématiciens qui font de l'algébre tous les jours et qui aiment le vocabulaire compliqué l'appellent **groupe diédral** d'ordre 6, les autres l'appellent plus simplement groupe des symétries du triangle. On pourrait bien sûr faire de même avec un carré, un hexagone ou même n'importe quel polygone régulier à n cotés. On obtient toujours un groupe non commutatif à $2n$ éléments (il n'y que des rotations et des réflexions, comme pour le triangle).

1.3 Noyau et image

Définition 3. Le **noyau** d'une application linéaire $f : E \to F$ est l'ensemble $\ker(f) = \{u \in E \mid f(u) = 0\}$.
L'**image** d'une application linéaire $f : E \to F$ est l'ensemble $\mathrm{Im}(f) = \{v \in F \mid \exists u \in E, f(u) = v\}$.

Remarque 6. Les lettres Ker sont les premières du mot allemand Kernel qui signifie, comme vous auriez pu le deviner, noyau. L'image correspond en fait tout simplement à la notion d'image d'un ensemble par une application (pas forcément linéaire) que nous avons déjà utilisée plus tôt dans l'année. Le noyau, lui, correspond en fait aux antécédents d'un élément très particulier de l'ensemble d'arrivée, son vecteur nul. On va voir ci-dessous que, lorsqu'une application est linéaire, le fait de connaitre les antécédents de 0 suffit en fait à en déduire des choses sur les antécédents de **tous** les éléments de l'espace d'arrivée. Première remarque : on a dit en début de ce chapitre qu'une application linéaire vérifiait toujours $f(0) = 0$, le noyau de f contient donc toujours au moins un élément, il ne peut jamais être vide. C'est en fait pire que ça :

Proposition 3. Si $f : E \to F$ est une application linéaire, alors l'image d'un sous-espace vectoriel de E est toujours un sous-espace vectoriel de F ; et l'image réciproque de tout sous-espace vectoriel de F est un sous-espace vectoriel de E.

Corollaire 1. Si $f : E \to F$ est une application linéaire, alors $\ker(f)$ est un sous-espace vectoriel de E, et $\mathrm{Im}(f)$ est un sous-espace vectoriel de F.

Démonstration. Soit G un sous-espace vectoriel de E, et $(z, w) \in f(G)^2$, alors $z = f(u)$ et $w = f(v)$, avec $(u, v) \in G^2$. Comme l'application est linéaire, $\lambda z + \mu w = \lambda f(u) + \mu f(v) = f(\lambda u + \mu v) \in f(G)$ puisque $\lambda u + \mu v \in G$ en tant que combinaison linéaire d'éléments du sous-espace vectoriel G. L'image de G est donc stable par combinaison linéaire, c'est un sous-espace vectoriel de G. C'est le même principe pour l'image réciproque : soit H un sous-espace vectoriel de F, et $(u, v) \in f^{-1}(H)$, alors $f(u) = z \in H$ et $f(v) = w \in H$, donc $f(\lambda u + \mu v) = \lambda z + \mu w \in H$, et $\lambda u + \mu v \in f^{-1}(H)$. □

Exemple : En pratique, un noyau se calcule très simplement en résolvant un système. Déterminons par exemple le noyau de l'endomorphisme f de $\mathbb{R}^3$ défini par $f : (x, y, z) \mapsto (x - y + z, 3x - 2y + 5z, -x - 3z)$ (je vous dispense de la vérification pénible du fait que f est bien une application linéaire). Les éléments du noyau sont les triplets de réels (x, y, z) solutions du système $\begin{cases} x - y + z = 0 \\ 3x - 2y + 5z = 0 \\ -x - 3z = 0 \end{cases}$. Le système n'est pas de Cramer ($2L_1 - L_2 = L_3$), les solutions sont les triplets de la forme $(-3z, -2z, z)$, avec $z \in \mathbb{R}$. Autrement dit, $\ker(f) = \mathrm{Vect}((-3, -2, 1))$. On essaiera de systématiquement présenter les noyaux d'applications linéaires « sous forme de Vect ».

Exemple 2 : L'application $f : \mathbb{R}_n[X] \to \mathbb{R}_n[X]$ définie par $f(P) = P'$ a pour noyau l'ensemble des polynômes constants, ce qu'on peut écrire $\ker(f) = \mathrm{Vect}(1)$.

Proposition 4. Une application linéaire $f : E \to F$ est injective si et seulement si $\ker(f) = \{0\}$. Une application linéaire $f : E \to F$ est surjective si et seulement si $\mathrm{Im}(f) = F$.

Démonstration. La deuxième propriété est évidente, c'est la définition de la surjectivité. Démontrons donc la première, qui est beaucoup plus intéressante puisqu'elle revient à dire que, pour démontrer qu'une application linéaire est injective, il suffit de démontrer qu'un seul élément bien particulier (le vecteur nul) n'a pas plus d'un antécédent par f. Supposons d'abord le noyau réduit au vecteur nul et montrons que f est injective : soient $(u, v) \in E^2$ tels que $f(u) = f(v)$, alors $f(u - v) = f(u) - f(v) = 0$, donc $u - v \in \ker(f)$, donc $u - v = 0$, c'est-à-dire $u = v$, ce qui prouve bien l'injectivité. Réciproquement, supposons f injective, alors 0 a un seul antécédent par f. Or, le vecteur nul est toujours un antécédent de 0 par une application linéaire. Ceci prouve bien qu'il est le seul élément de E à appartenir à $\ker(f)$. □

Proposition 5. Soit $f \in \mathcal{L}(E, F)$ et $(e_1, \ldots, e_n)$ une base de E, alors $\mathrm{Im}(f) = \mathrm{Vect}(f(e_1), \ldots, f(e_n))$.

Démonstration. Comme les vecteurs $f(e_1), \ldots, f(e_n)$ appartiennent évidemment à $\mathrm{Im}(f)$, on a nécessairement $\mathrm{Vect}(f(e_1), \ldots, f(e_n)) \subset \mathrm{Im}(f)$. De plus, si $v \in \mathrm{Im}(f)$, alors $v = f(u)$ avec $u \in E$, et comme $(e_1, \ldots, e_n)$ est une base de E, on peut écrire $u = \sum_{i=1}^{i=n} \lambda_i e_i$. Alors $v = f(u) = f\left(\sum_{i=1}^{i=n} \lambda_i e_i\right) = \sum_{i=1}^{i=n} \lambda_i f(e_i)$, donc $v \in \mathrm{Vect}(f(e_1), \ldots, f(e_n))$, et les deux ensembles sont bien égaux. □

Remarque 7. Attention, en général, $(f(e_1), \ldots, f(e_n))$ n'est pas une **base** de $\mathrm{Im}(f)$, mais seulement une famille génératrice (elle n'a aucune raison d'être libre, ce ne sera d'ailleurs en pratique le cas que si f est un isomorphisme). Remarquons également qu'une application linéaire est parfaitement déterminée par la simple donnée des images des vecteurs d'une base, puisqu'on peut reconstituer toutes les autres images par combinaisons linéaires.

Exemple 1 : La méthode élémentaire pour calculer une image est d'utiliser la définition, mais elle n'est pas pratique du tout. Prenons par exemple l'application linéaire de $\mathbb{R}^2$ dans $\mathbb{R}^3$ définie par $f(x,y) = (2x-y, x+2y, -2x+y)$. Un triplet (a,b,c) appartient à $\mathrm{Im}(f)$ si et seulement si le système

$$\begin{cases} 2x & - & y & = & a \\ x & + & 2y & = & b \\ -2x & + & y & = & c \end{cases}$$

admet une solution. Les membres de gauche des deux équations extrêmes étant opposés, il faut nécessairement avoir $a = -c$, et on vérifie facilement que cette condition est suffisante. On a donc $\mathrm{Im}(f) = \{(a,b,-a) \mid a,b \in \mathbb{R}^2\} = \mathrm{Vect}((1,0,-1);(0,1,0))$.

Exemple 2 : En pratique, on utilisera plutôt notre dernière proposition, car c'est beaucoup plus rapide ! Reprenons le même exemple. La base canonique de $\mathbb{R}^2$ est constituée des deux vecteurs $(1,0)$ et $(0,1)$, donc l'image est engendrée par $f(1,0) = (2,1,-2)$ et $f(0,1) = (-1,2,1)$. On a donc $\mathrm{Im}(f) = \mathrm{Vect}((2,1,-2);(-1,2,1))$ (ce ne sont pas les mêmes vecteurs que tout à l'heure mais on peut vérifier qu'ils engendrent le même espace vectoriel).

Proposition 6. Si f est un isomorphisme de E dans F, alors sa réciproque f^{-1} est un isomorphisme de F dans E.

Démonstration. La seule chose à vérifier est que la réciproque est une application linéaire. Soient donc $(z,w) \in F^2$ et notons $u = f^{-1}(z)$ et $v = f^{-1}(w)$. Par linéarité de f, on peut dire que $f(\lambda u + \mu v) = \lambda f(u) + \mu f(v) = \lambda z + \mu w$, donc $f^{-1}(\lambda z + \mu w) = \lambda u + \mu v = \lambda f^{-1}(z) + \mu f^{-1}(w)$, ce qui prouve la linéarité de f^{-1}. □

Remarque 8. L'ensemble $\mathcal{L}(E)$ muni des deux opérations $+$ et $\circ$ est ce qu'on appelle un anneau non commutatif. L'addition joue son rôle usuel et la composition joue à peu de choses près le rôle de la multiplication dans les ensembles de nombres usuels ($\mathbb{R}$ ou $\mathbb{C}$ par exemple). En effet, la composition admet un élément neutre qui est l'application identité, mais toutes les applications linéaires ne sont pas inversibles (seuls les automorphismes le sont), et surtout la composition est distributive par rapport à l'addition, tout comme le produit dans les ensembles de nombres (c'est-à-dire qu'on a toujours $f \circ (g+h) = f \circ g + f \circ h$ et $(g+h) \circ f = g \circ f + h \circ f$, ce qu'on ne se privera pas d'exploiter dans certains calculs). En fait, nous verrons plus tard que la composition d'applications linéaires s'identifie effectivement à un produit, celui des matrices. Pour l'instant, nous utiliserons déjà cette analogie pour justifier l'énorme abus de notation suivant : pour une application linéaire, on notera $f \circ f = f^2$ (un carré au sens « produit » n'aurait en général aucun sens), et plus général f^n la composée de f n fois par elle-même. Faites très attention à ne surtout pas me faire des calculs du type « élévation au carré » quand vous croisez cette notation f^2 dans les exercices, ça ne rimerait absolument à rien !