

3. Théorie de la persistance (1/2) : structure

La théorie de la persistance s'applique à deux niveaux en parallèle : au niveau topologique, sur des filtrations d'espaces topologiques ; au niveau algébrique, sur les modules induits en homologie, appelés *modules de persistance*. Le cœur de la théorie se situe au niveau algébrique et les modules de persistance sont son principal objet d'étude. Dans ce chapitre on s'attachera essentiellement à leurs propriétés de décomposition et au calcul des *codes-barres* induits, tandis que dans le chapitre suivant on regardera leurs propriétés de stabilité.

Dans tout le chapitre on fixe un corps \mathbb{K} et un ensemble totalement ordonné (T, \leq) .

3.1. Filtrations d'espaces topologiques. — Rappelons d'abord le concept classique de filtration d'espaces topologiques et donnons quelques exemples clés pour l'analyse topologique de données.

Définition 3.1. — Une filtration d'espaces topologiques indexée par T est une famille $(X_t)_{t \in T}$ d'espaces topologiques imbriqués au sens que $X_t \subseteq X_{t'}$ pour tous $t \leq t' \in T$.

Exemple 3.1. — Étant donné un espace topologique X et une fonction $f: X \rightarrow \mathbb{R}$, tous deux quelconques, la famille des sous-niveaux $F_t = f^{-1}(\cdot - \infty, t]$ forme une filtration indexée par $T = \mathbb{R}$ muni de l'ordre usuel sur les réels. De manière duale, comme vu au chapitre 1, la famille des sur-niveaux $F^t = f^{-1}([t, +\infty[)$ forme une filtration indexée par $T = \mathbb{R}$ muni de l'ordre opposé.

Définition 3.2. — Une filtration $(X_t)_{t \in T}$ est dite simpliciale si chaque espace X_t est un complexe simplicial et un sous-complexe de $X_{t'}$ pour tout $t' \geq t$ dans l'ensemble T .

Exemple 3.2. — Étant donné un complexe simplicial X et une fonction $f: X \rightarrow \mathbb{R}$ qui associe un réel à chaque simplexe de X , la filtration des sous-niveaux de f est simpliciale. Voir la figure 16 pour une illustration.

3.2. Modules de persistance. — Comme dit en introduction du chapitre, les modules de persistance sont le principal objet d'étude de la théorie de la persistance.

Définition 3.3. — Un \mathbb{K} -module de persistance indexé par T , ou simplement module de persistance lorsque le corps \mathbb{K} et l'ensemble d'indexation T sont clairs dans le contexte, est une famille

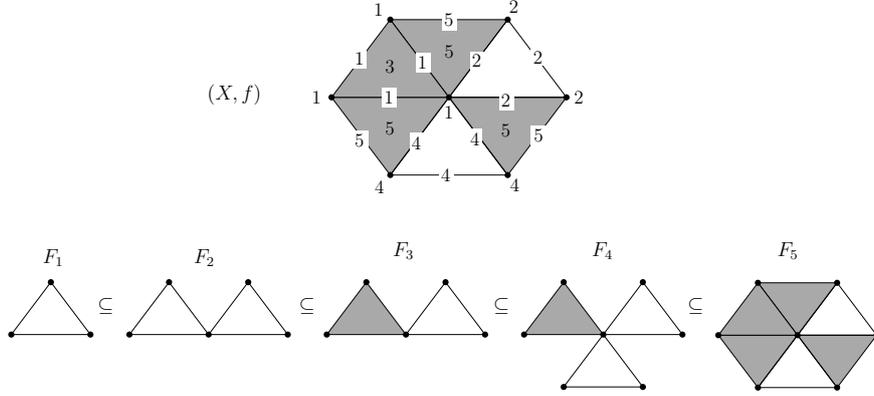


FIGURE 16. En haut : une fonction réelle f sur un complexe simplicial X . En bas : la filtration (simpliciale) des sous-niveaux de f , restreinte aux indices $\{1, 2, 3, 4, 5\}$.

$(M_t)_{t \in T}$ de \mathbb{K} -espaces vectoriels reliés par des applications \mathbb{K} -linéaires $m_{t,t'} : M_t \rightarrow M_{t'}$ pour tous $t \leq t' \in T$ telles que :

$$\begin{cases} m_{t,t} = \text{id}_{M_t} & \forall t \in T \\ m_{t,t''} = m_{t',t''} \circ m_{t,t'} & \forall t \leq t' \leq t'' \in T \end{cases}$$

Les espaces M_t et les applications $m_{t,t'}$ sont appelés respectivement espaces internes et applications internes au module M .

Le cas d'usage typique des modules de persistance en analyse topologique de données est le suivant :

Lemme 3.1. — Pour toute filtration $(X_t)_{t \in T}$ et tout degré k en homologie, la famille des groupes d'homologie $(H_k^{\text{sing}}(X_t))_{t \in T}$, reliés par les applications linéaires $(\iota_{t,t'})_*$ induites par les inclusions $\iota_{t,t'} : X_t \hookrightarrow X_{t'}$, forme un module de persistance indexé par T . Celui-ci est appelé le module de persistance induit par $(X_t)_{t \in T}$ en homologie de degré k .

Démonstration. — Découle directement de la functorialité de l'homologie. \square

Exemple 3.3. — Le module de persistance induit en homologie de degré 1 par la filtration simpliciale de la figure 16 restreinte aux indices $\{1, 2, 3, 4, 5\}$ est le suivant :

$$\mathbb{K} \xrightarrow{\begin{pmatrix} 1 \\ 0 \end{pmatrix}} \mathbb{K}^2 \xrightarrow{\begin{pmatrix} 0 & 1 \end{pmatrix}} \mathbb{K} \xrightarrow{\begin{pmatrix} 0 \\ 1 \end{pmatrix}} \mathbb{K}^2 \xrightarrow{\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}} \mathbb{K}^2$$

Notons que seules les applications internes reliant deux espaces internes successifs sont explicitées ici, les autres se déduisent par composition. La représentation matricielle des applications internes dépend du choix des bases pour les groupes d'homologie : ici on a pris les cycles minimaux entourant chaque "trou" dans le complexe simplicial pour former les bases.

Définition 3.4. — Un module de persistance N est un sous-module d'un autre module de persistance M , noté $N \subseteq M$, si N_t est un sous-espace vectoriel de M_t pour tout $t \in T$, et si $n_{t,t'}$ est la restriction de $m_{t,t'}$ à N_t pour tous $t \leq t' \in T$.

Définition 3.5. — Un morphisme de \mathbb{K} -modules de persistance $\phi: M \rightarrow N$ est une famille d'applications \mathbb{K} -linéaires $(\phi_t: M_t \rightarrow N_t)_{t \in T}$ telles que le diagramme suivant commute pour tous $t \leq t' \in T$:

$$\begin{array}{ccc} M_t & \xrightarrow{m_{t,t'}} & M_{t'} \\ \phi_t \downarrow & & \downarrow \phi_{t'} \\ N_t & \xrightarrow{n_{t,t'}} & N_{t'} \end{array}$$

On a en particulier un *morphisme identité* pour chaque module de persistance M , noté id_M et défini point-à-point par $(\text{id}_M)_t = \text{id}_{M_t}$. On a également un *morphisme nul* $M \xrightarrow{0} N$ entre toute paire de modules de persistance M, N , défini point-à-point par l'application linéaire nulle. La composition des morphismes de modules de persistance, quant à elle, est définie point-à-point par la composition des applications linéaires :

Définition 3.6. — Pour tous morphismes de modules de persistance $L \xrightarrow{\phi} M \xrightarrow{\psi} N$ indexés par T , la composée $\psi \circ \phi$ est définie par :

$$(\psi \circ \phi)_t = \psi_t \circ \phi_t \quad \forall t \in T$$

La composition est donc associative, du fait de l'associativité de la composition des applications linéaires. De plus, pour tout morphisme $\phi: M \rightarrow N$ on a $\phi \circ \text{id}_M = \text{id}_N \circ \phi = \phi$, ainsi les morphismes identités associés aux modules de persistance forment l'identité pour la composition.

3.3. Le point de vue de la théorie des catégories. — Les définitions ci-dessus peuvent être reformulées de manière très succincte en utilisant le langage de la théorie des catégories. On peut en effet voir l'ensemble totalement ordonné T comme une catégorie (notée \mathbb{T}), avec un objet par élément $t \in T$ et un unique morphisme par paire d'éléments comparables $t \leq t' \in T$. La définition 3.3 revient alors à dire que les modules de persistance sont des foncteurs de \mathbb{T} dans la catégorie des \mathbb{K} -espaces vectoriels, noté $\mathbb{T} \rightarrow \mathbf{Vect}_{\mathbb{K}}$, tandis que la définition 3.5 revient à dire que les morphismes entre modules de persistance sont les transformations naturelles entre foncteurs $\mathbb{T} \rightarrow \mathbf{Vect}_{\mathbb{K}}$, dont la composition se fait point-à-point comme rappelé dans la définition 3.6. Ensemble, ces éléments nous disent que les modules de persistance forment une catégorie de foncteurs.

Au-delà de ce langage, la théorie des catégories fournit un ensemble d'outils et de résultats fondamentaux sur lesquels nous pouvons nous appuyer pour justifier les diverses propriétés des modules de persistance. En premier lieu, il est bien connu que les foncteurs à valeurs dans une catégorie abélienne comme $\mathbf{Vect}_{\mathbb{K}}$ forment eux-mêmes une catégorie abélienne, avec les implications suivantes dans notre cas :

- pour tous modules de persistance M, N , l'ensemble des morphismes $M \rightarrow N$ forme un groupe abélien ; de plus, la composition des morphismes de modules de persistance est bilinéaire ;
- il existe un (unique) module de persistance, appelé *zéro-module de persistance* et noté 0 , qui a la propriété que tout morphisme $M \rightarrow 0$ ou $0 \rightarrow N$ est nul, et qui se définit comme ayant tous ses espaces internes et toutes ses applications internes nuls ;
- le produit et la somme directe de modules de persistance existent et sont définis point-à-point comme dans la définition 3.7 ci-dessous ;
- tout morphisme de modules de persistance admet un noyau et un conoyau, définis point-à-point comme dans la définition 3.8 ci-dessous ; de plus, un morphisme ϕ est simplifiable à gauche pour la composition ($\phi \circ \psi = \phi \circ \psi' \Rightarrow \psi = \psi'$) si et seulement si son noyau est nul ; similairement, ϕ est simplifiable à droite si et seulement si son conoyau est nul.

Ainsi, on peut manipuler les modules de persistance et les morphismes qui les relient un peu à la manière des espaces vectoriels et des applications linéaires. D'ailleurs, la théorie de la persistance contient celle des espaces vectoriels puisque la catégorie des modules de persistance indexés par un ensemble singleton (par exemple $T = \{1\}$) est équivalente à celle des espaces vectoriels, chaque module dans ce cas étant

formé d'un unique espace interne et chaque morphisme entre modules étant formé d'une unique application linéaire.

Définition 3.7. — *Étant donnés deux modules de persistance M, N , leur somme directe $M \oplus N$ est définie point-à-point, c'est-à-dire que $(M \oplus N)_t = M_t \oplus N_t$ pour tout $t \in T$, et $m_{t,t'} = m_{t,t'} \oplus n_{t,t'}$ pour tous $t \leq t' \in T$, où $(m_{t,t'})_{t \in T}$ désigne la famille d'applications internes au module $M \oplus N$.*

Définition 3.8. — *Pour tout morphisme de modules de persistance $\phi: M \rightarrow N$, le noyau $\ker \phi$, l'image $\text{im } \phi$ et le conoyau $\text{coker } \phi$ sont tous trois définis point-à-point de la manière suivante (décrite pour le noyau et analogue pour l'image et le conoyau) :*

- pour tout $t \in T$, l'espace $(\ker \phi)_t$ est défini comme étant $\ker \phi_t$;
- pour tous $t \leq t' \in T$, l'application interne $(\ker \phi)_t \rightarrow (\ker \phi)_{t'}$ est définie comme étant la restriction de $m_{t,t'}$ à $\ker \phi_t$.

Un morphisme $\phi: M \rightarrow N$ est appelé *monomorphisme* lorsqu'il est simplifiable à gauche pour la composition, ce qui, comme on l'a vu plus haut, revient à dire que $\ker \phi = 0$ (i.e. $\ker \phi_t = 0$ pour tout $t \in T$). Similairement, ϕ est appelé *épimorphisme* lorsqu'il est simplifiable à droite, ce qui revient à dire que $\text{coker } \phi = 0$ (i.e. $\text{coker } \phi_t = 0$ pour tout $t \in T$). Enfin, ϕ est appelé *isomorphisme* lorsque $\ker \phi = 0 = \text{coker } \phi$, et dans ce cas les modules M et N sont dits *isomorphes*, noté $M \cong N$.

3.4. Le point de vue de l'algèbre commutative. — Nous allons maintenant justifier l'usage du terme *module* dans la dénomination des modules de persistance. Il s'avère en effet que, sous certaines conditions, les modules de persistance peuvent être vus comme des modules sur des algèbres. Cela provient de résultats plus généraux de plongement des catégories abéliennes dans des catégories de modules. Pour simplifier l'exposé nous allons nous concentrer sur le cas particulier où l'ensemble totalement ordonné T est fini : dans ce cas, les modules de persistance peuvent être vus comme des modules sur l'algèbre des polynômes à une variable à coefficients dans \mathbb{K} .

Supposons donc que T est fini, et même, sans perte de généralité, que $T = \{0, 1, \dots, l\} \subset \mathbb{N}$. Soit M un module de persistance indexé par T . On l'étend d'abord en un module de persistance M' indexé par \mathbb{N} en posant

$$\begin{cases} M'_t = M_{\min\{t,l\}} & \forall t \in \mathbb{N} \\ m'_{t,t'} = m_{\min\{t,l\}, \min\{t',l\}} & \forall t \leq t' \in \mathbb{N} \end{cases}$$

On vérifie aisément que M' est bien un module de persistance. On considère ensuite le \mathbb{K} -espace vectoriel

$$\bar{M} = \bigoplus_{t=0}^{\infty} M'_t$$

que l'on munit d'une structure de module à gauche sur $\mathbb{K}[x]$ en posant

$$x \cdot (\alpha_0, \alpha_1, \alpha_2, \dots) = (0, m'_{0,1}(\alpha_0), m'_{1,2}(\alpha_1), \dots)$$

pour tout élément $(\alpha_0, \alpha_1, \alpha_2, \dots)$ de \bar{M} . La structure de module est complétée par composition et linéarité, et l'on observe que c'est en fait une structure de module \mathbb{N} -gradué sur l'algèbre $\mathbb{K}[x]$ (elle-même \mathbb{N} -gradué) dont la composante de degré t est M'_t .

Ainsi on envoie les modules de persistance pdf indexés par T dans les modules \mathbb{N} -gradués sur $\mathbb{K}[x]$. On peut montrer que cette opération induit un plongement de catégorie abélienne, ce qui en pratique signifie que l'on peut sans problème traiter les modules de persistance comme des modules sur l'algèbre des polynômes et utiliser des outils de l'algèbre commutative pour les étudier. On en verra un exemple dans la prochaine section.

3.5. Décomposition des modules de persistance. — L'une des questions fondamentales qui se posent à nous est celle de la classification des modules de persistance à isomorphisme près. Pour y répondre nous allons énoncer un théorème de structure (théorème 3.1) qui décompose les modules de persistance en somme directe de modules structurellement très simples, les *modules intervalles* (définition 3.10). C'est de ces décompositions que seront issus nos codes-barres (définition 3.12), qui, du fait de ce résultat structurel, forment un invariant complet des modules de persistance.

Définition 3.9. — *Si un module de persistance M se décompose (à isomorphisme près) en une somme directe $M \cong L \oplus N$ dont les deux termes sont non-nuls, alors on dit que M est décomposable et les termes L, N sont appelés des sommandes de M . Si au contraire il n'existe pas de décomposition de M de ce type, alors on dit que M est indécomposable.*

Comme on l'a vu à la section 3.3, les modules de persistance sont équivalents aux espaces vectoriels lorsque l'ensemble d'indexation T est un singleton, donc leur décomposition dans ce cas très particulier est facile. Dans le cas général en revanche, la question est beaucoup

plus complexe, et pour s'en convaincre il suffit de savoir que les modules de persistance ne sont alors généralement plus semi-simples, c'est-à-dire que tous leurs sous-modules ne sont pas des sommandes (autrement dit, il n'y a pas d'analogue du théorème de la base incomplète). Voici un exemple.

Exemple 3.4. — Prenons le module de persistance $\mathbb{K} \xrightarrow{\text{id}_{\mathbb{K}}} \mathbb{K}$ indexé sur l'ensemble $T = \{1, 2\}$. Ce module admet $0 \rightarrow \mathbb{K}$ comme sous-module mais pas comme sommande. En effet, si c'était une sommande alors, par additivité de la dimension, tout supplémentaire devrait avoir dimension 1 à l'indice 1 et dimension 0 à l'indice 2 et serait donc isomorphe à $\mathbb{K} \rightarrow 0$. Or, ce dernier ne peut être un sous-module de M car $\text{id}_{\mathbb{K}}$ envoie \mathbb{K} sur \mathbb{K} et non pas sur 0.

Le bon côté des choses est que, sous certaines hypothèses, les modules de persistance indécomposables sont structurellement très simples. On les appelle des *modules intervalles*.

Définition 3.10. — Un intervalle est un sous-ensemble I de T qui est convexe pour l'ordre, c'est-à-dire que pour tous $t \leq t' \leq t'' \in T$ tels que $t, t'' \in I$ on a $t' \in I$. Le module intervalle \mathbb{K}_I associé à I est défini point-à-point par

$$(\mathbb{K}_I)_t = \begin{cases} \mathbb{K} & \text{si } t \in I \\ 0 & \text{sinon} \end{cases}$$

avec pour application interne $(\mathbb{K}_I)_t \rightarrow (\mathbb{K}_I)_{t'}$ l'identité de \mathbb{K} si $t, t' \in I$ et l'application nulle sinon. I est appelé le support de \mathbb{K}_I .

Par exemple, les modules de persistance impliqués dans l'exemple 3.4 sont tous des modules intervalles.

Exercice 3.1. — Montrer que les modules intervalles sont indécomposables.

Il existe tout un éventail de résultats de décomposition en somme directe des modules de persistance. Le résultat ci-dessous est le plus couramment employé. Il fait l'hypothèse que le module de persistance considéré est pdf :

Définition 3.11. — Un module de persistance M est point-à-point de dimension finie (pdf) si $\dim M_t < +\infty$ pour tout indice $t \in T$.

Théorème 3.1 (décomposition). — *Tout module de persistance pdf M indexé par un ensemble totalement ordonné T se décompose en une somme directe (potentiellement avec un nombre infini de termes au total, mais point-à-point finie) dont chaque terme est un module intervalle. Cette décomposition est unique à isomorphisme et réordonnement près des termes.*

Exemple 3.5. — *Le module de persistance M de l'exemple 3.3, indexé par $T = \{1, 2, 3, 4, 5\}$, se décompose comme suit :*

$$M \cong \mathbb{K}_{\{1,2\}} \oplus \mathbb{K}_{\{2,3,4,5\}} \oplus \mathbb{K}_{\{4,5\}}$$

Dans ce cas particulier la décomposition se lit directement dans les matrices des applications internes de M , qui sont diagonales. Dans le cas général toutefois, les matrices n'ont pas de raison d'être diagonales mais le théorème de décomposition dit en substance qu'il existe une famille de bases pour les espaces internes de M qui sont compatibles entre elles au sens qu'elles rendent les matrices diagonales.

Remarque 3.1. — *Lorsque $T = \mathbb{R}$, les intervalles supports des sommandes dans la décomposition peuvent être ouverts, semi-ouverts, ou fermés. Ceci contraste avec les intervalles du chapitre 1, que l'on avait forcés à être semi-ouverts à gauche par convention.*

L'unicité de la décomposition dans le théorème 3.1 implique en particulier celle des intervalles supports des sommandes, avec lesquels on peut ainsi former le *code-barres* du module.

Définition 3.12. — *Pour un module de persistance pdf M , le code-barres de M , noté $B(M)$, est le multiensemble des intervalles supports des sommandes dans la décomposition de M . Lorsque $T = \mathbb{R}$ muni de l'ordre usuel, le diagramme de persistance, noté $D(M)$, est le multiensemble de points (a, b) dans le plan étendu $\mathbb{R} \times]-\infty, +\infty]$ correspondant aux intervalles $[a, b]$, $]a, b]$, $]a, b[$ ou $]a, b[$ de $B(M)$.*

Le théorème de décomposition implique que le code-barres est un invariant complet pour les modules de persistance pdf indexés par $T \subseteq \mathbb{R}$. La nature purement géométrique du code-barres en fait un outil idéal pour, d'une part, l'interprétation visuelle de la structure du module, et d'autre part, la comparaison de cette structure avec celles d'autres modules. On verra en effet au chapitre 4 que l'on peut définir une distance naturelle entre modules de persistance, que les codes-barres permettent de calculer exactement.

Remarque 3.2. — Ici on ne définit le diagramme de persistance que dans le cas où $T = \mathbb{R}$ car dans le cas général les extrémités (ouvertes ou fermées) des intervalles ne sont pas toujours définies de manière unique. Notons par ailleurs suite à la remarque 3.1 que, dans le cas où $T = \mathbb{R}$, le diagramme de persistance n'est plus équivalent au code-barres car il perd l'information sur le caractère ouvert, semi-ouvert ou fermé des intervalles. En particulier, le diagramme de persistance n'est pas un invariant complet des modules de persistance pdf.

Remarque 3.3. — Dans le contexte de l'exemple 3.1, le lemme 3.1 nous fournit un module de persistance par degré en homologie pour la filtration des sur-niveaux d'une fonction $f: X \rightarrow \mathbb{R}$ quelconque. Le théorème 3.1 assure alors que ce module de persistance a un code-barres bien défini, à la seule condition que les groupes d'homologie des sur-niveaux de f soient de dimension finie. Dans le cas particulier de l'homologie de degré 0, cette hypothèse revient à dire que les sur-niveaux de f ont chacun un nombre fini de composantes connexes par arc. Cette condition est beaucoup moins restrictive que celles que nous avons dû énoncer au chapitre 1. En particulier, à présent le code-barres encode les intervalles de persistance de toutes les composantes connexes par arc dans les sur-niveaux de f , et non plus seulement de celles engendrées par les pics de la fonction.

Origine du théorème de décomposition dans le cas où T est fini. — Supposons que T est fini, égal à $\{0, 1, \dots, l\}$ sans perte de généralité. Comme nous l'avons vu dans la section 3.4, on peut envoyer tout module de persistance M sur un module \mathbb{N} -gradué \bar{M} sur l'algèbre graduée des polynômes $\mathbb{K}[x]$, et cette opération définit un plongement de catégorie abélienne. Cela implique que l'on peut décomposer \bar{M} en tant que module \mathbb{N} -gradué sur $\mathbb{K}[x]$ puis en déduire une décomposition de M en tant que module de persistance. C'est ce que nous allons faire maintenant.

Soit donc un module de persistance M pdf indexé par T . Dans la construction du module de persistance M' indexé par \mathbb{N} donnée à la section 3.4, remarquons que les applications internes deviennent des isomorphismes au-delà de l'indice l . En conséquence, les morphismes de transition entre les degrés au-delà du degré l dans \bar{M} sont des isomorphismes, ce qui implique que \bar{M} est de type fini en tant que module gradué sur l'anneau $\mathbb{K}[x]$. Ce dernier étant euclidien, on peut

décomposer \bar{M} en utilisant le théorème des facteurs invariants, ou plutôt sa version graduée⁽⁴⁾, qui nous donne la décomposition suivante :

$$\bar{M} \cong \bigoplus_{j \in \mathcal{J}_{\text{libre}}} x^{a_j} \cdot \mathbb{K}[x] \oplus \bigoplus_{j \in \mathcal{J}_{\text{torsion}}} x^{a_j} \cdot \left(\mathbb{K}[x] / x^{d_j} \cdot \mathbb{K}[x] \right)$$

La première somme directe dans la décomposition est la partie libre de \bar{M} , qui engendre les intervalles s'étendant à droite indéfiniment dans la décomposition du module de persistance M , de la forme $[a_j, +\infty[\cap T = \{a_j, \dots, l\}$ avec $0 \leq a_j \leq l$. La deuxième somme directe dans la décomposition ci-dessus est la torsion du module \bar{M} , qui engendre le reste des intervalles dans la décomposition de M , de la forme $[a_j, a_j + d_j[= \{a_j, \dots, a_j + d_j - 1\}$ avec $0 \leq a_j < a_j + d_j \leq l$ – cette dernière inégalité provient du fait que les morphismes de transition au-delà du degré l dans \bar{M} sont des isomorphismes.

Ainsi, le théorème 3.1 dans le cas particulier où T est fini découle directement du théorème des facteurs invariants. Dans le cas général, le résultat est beaucoup plus subtile et se prouve avec des outils de théorie des représentations d'algèbres développés dans les années 1970 – voir par exemple la preuve de Crawley-Boevey (2015).

3.6. Calcul des codes-barres. — Pour faire des calculs on se place dans le cadre simplicial, avec en entrée une filtration simpliciale de longueur finie dont on suppose que chaque complexe simplicial a un nombre fini de simplexes :

$$\emptyset = X_0 \subseteq X_1 \subseteq \dots \subseteq X_l = X$$

Sans perte de généralité, on suppose de plus que chaque inclusion $X_{i-1} \subseteq X_i$ dans la séquence correspond à l'ajout d'un unique simplexe σ_i dans le complexe. Notons que, du fait que les X_i sont d'honnêtes complexes simpliciaux, σ_i n'apparaît dans la filtration qu'une fois que toutes ses faces sont apparues. Ceci ne veut toutefois pas dire que tous les sommets apparaissent en premier, puis toutes les arêtes, et ainsi de suite.

Sur cette entrée nous allons calculer les code-barres des modules de persistance induits en homologie simpliciale par la filtration pour tous les degrés d'homologie à la fois. La méthode est en fait la même que

4. Une preuve constructive classique du théorème des facteurs invariants dans le cas d'un anneau euclidien repose sur la réduction en forme de Smith de la matrice d'une présentation finie du module. Dans le cas gradué, la preuve est identique une fois que l'on a observé que les idéaux impliqués sont tous homogènes.

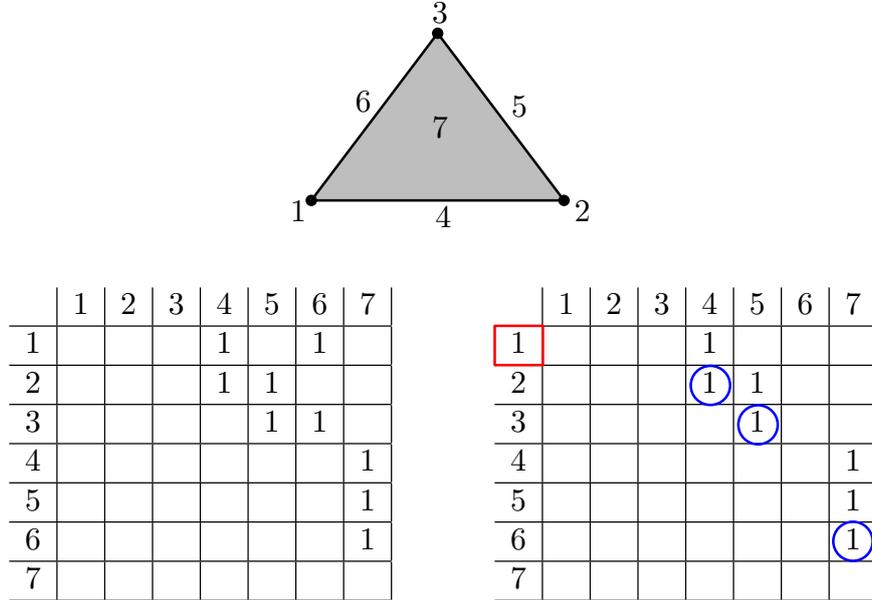


FIGURE 17. En haut : une fonction réelle f sur un complexe simplicial X . En bas : la matrice des opérateurs de bord de tous les degrés de X (gauche), ordonnée selon les valeurs de f , et sa version réduite par le pivot de Gauss (droite).

celle du calcul de l'homologie simpliciale usuelle (voir la remarque 2.1 du chapitre 2), au détail important près que l'on ne peut modifier l'ordre sur les simplexes imposé par la filtration lors de l'exécution du pivot de Gauss pour réduire les matrices des opérateurs de bord. Voici les étapes, illustrées sur un exemple dans la figure 17 :

1. On construit tout d'abord la matrice D contenant les matrices des opérateurs de bord de tous les degrés du complexe simplicial total X . Chaque simplexe σ_i donne lieu à exactement une ligne et une colonne dans D , et ces dernières sont ordonnées par indices croissants dans la filtration. En notant d_i la dimension du simplexe σ_i , l'entrée $D_{i,j}$ de la matrice est nulle si $d_i \neq d_j - 1$, sinon elle est égale à la coordonnée du vecteur $\partial_{d_j}(\sigma_j)$ le long de l'axe σ_i dans l'espace des chaînes $C_{d_i}(X)$. Quand $\mathbb{K} = \mathbb{Z}/2\mathbb{Z}$, cela revient à dire que $D_{i,j}$ vaut 1 si σ_i est une face de σ_j et 0 sinon.

2. On réduit la matrice D à une forme échelonnée en colonnes, sans changer l'ordre des lignes et des colonnes. Le *pivot* d'une colonne non-nulle est son entrée non-nulle la plus basse (c'est-à-dire située sur

Algorithme 3.1 : Algorithme de réduction par pivot de Gauss

Entrées : matrice D de taille $l \times l$ à coefficients dans \mathbb{K}
pour j allant de 1 à l faire
 tant que $\exists i < j$ tel que $\text{pivligne}(i) = \text{pivligne}(j) \neq 0$ faire
 | $D_{\cdot,j} \leftarrow D_{\cdot,j} - \frac{\text{piv}(j)}{\text{piv}(i)} D_{\cdot,i}$
 fin
fin
Output : la matrice D réduite

la ligne d'indice le plus élevé), et la forme échelonnée a la propriété que les pivots se situent tous sur des lignes distinctes. Pour atteindre cet état, on applique une version restreinte du pivot de Gauss décrite dans l'algorithme 3.1, où $D_{\cdot,i}$ désigne la colonne de D d'indice i , la fonction $\text{pivligne}(i)$ renvoie l'indice de la ligne sur laquelle se trouve le pivot de $D_{\cdot,i}$ lorsqu'il existe et zéro sinon, enfin la fonction $\text{piv}(i)$ renvoie la valeur du pivot de $D_{\cdot,i}$ (qui existe lorsque cette fonction est appelée).

3. Une fois la matrice D réduite en forme échelonnée, ses pivots définissent un appariement partiel des simplexes de X , à partir duquel le code-barres de la filtration peut être déduit. Plus précisément, à tout pivot situé à l'intersection d'une ligne i et d'une colonne j correspond la paire de simplexes (σ_i, σ_j) , qui engendrent l'intervalle $[i, j]$ en degré d_i dans le code-barres. Les simplexes impliqués dans aucune paire, quant à eux, engendrent les intervalles infinis du code-barres; plus précisément, le simplexe σ_k engendre l'intervalle $[k, +\infty)$ en degré d_k .

Exemple 3.6. — Prenons la filtration des sous-niveaux de la fonction de la figure 17. Une fois la matrice des opérateurs de bord réduite, il reste 3 pivots qui forment les paires de simplexes $(2, 4)$, $(3, 5)$ et $(6, 7)$ – ici on identifie chaque simplexe avec la valeur de f correspondante car f est injective. Ces paires engendrent les intervalles suivants : $[2, 4)$ en degré 0, $[3, 5)$ en degré 0, et $[6, 7)$ en degré 1. Il reste un simplexe non-apparié (le 1), qui engendre l'intervalle $[1, +\infty)$ en degré 0 dans le code-barres.

Terminaison et complexité. — Il est clair que les étapes 1 et 3 de la méthode terminent, seule l'étape 2 requiert un effort d'analyse. Il n'est en effet pas évident à première vue que l'algorithme 3.1 termine, du fait de la boucle interne **tant que**. L'observation clé est que, lors de

chaque mise à jour de la colonne $D_{.j}$, son pivot est annulé et toutes les entrées en-dessous dans la matrice restent à zéro car $D_{.i}$ a son pivot à la même hauteur que $D_{.j}$ avant la mise à jour. Ainsi, le pivot de $D_{.j}$ remonte strictement dans sa colonne à chaque mise à jour de celle-ci, ce qui implique que le nombre d'itérations de la boucle interne à chaque itération de la boucle externe est au plus le nombre de lignes, soit l . Il y a donc au total $O(l^2)$ mises à jour de colonnes, chacune en temps $O(l)$, ce qui implique la terminaison de l'algorithme 3.1 avec une complexité en temps dans le pire cas en $O(l^3)$. Cette quantité domine les complexités des étapes 1 et 3 de la méthode.

Correction. — L'étape 1 de la méthode ne présente aucun enjeu particulier. Ensuite, que le pivot de Gauss exécuté à l'étape 2 réduise bien la matrice D en forme échelonnée est un fait bien connu, dont la preuve est une variante de celle de la factorisation LU – voir par exemple (Cormen et al., 2009, chapitre 28). Il reste donc à se convaincre que l'interprétation de la structure de la matrice D réduite en termes de code-barres à l'étape 3 est correcte. Nous ne donnerons pas la preuve de ce fait, qui est technique et sans intérêt particulier – voir par exemple (Dey and Wang, 2022, section 3.3). Nous nous contenterons de donner l'intuition derrière, que voici.

L'algorithme 3.1 de réduction de matrice peut être vu comme construisant itérativement un sous-complexe Y du complexe simplicial total X , en commençant par $Y = X_0 = \emptyset$ et en terminant par $Y = X_l = X$, et en maintenant les groupes d'homologie de Y tout au long du processus. À chaque itération, Y transitionne de X_{j-1} à X_j par l'insertion du simplexe σ_j . Au moment de l'insertion, la réduction de la colonne d'indice j dans D détermine si cette colonne est linéairement dépendante des colonnes de D précédemment réduites, ce qui en fait détermine si le bord de σ_j est déjà homologué à zéro dans X_{j-1} ou pas : si c'est le cas (colonne $D_{.j}$ réduite à zéro), alors l'insertion de σ_j crée un nouveau cycle en homologie de degré d_j dans Y ; sinon (colonne non réduite à zéro), l'insertion de σ_j détruit un cycle en homologie de degré d_{j-1} dans Y , plus précisément le cycle créé précédemment par l'insertion du simplexe σ_i dont la ligne correspondante dans la matrice contient le pivot de la colonne $D_{.j}$ après sa réduction. La paire (σ_i, σ_j) engendre donc l'intervalle de persistance $[i, j]$ de ce cycle dans le code-barres en degré $d_i = d_{j-1}$. À la fin de l'algorithme, tout simplexe σ_k non-apparié a créé, lors de son insertion, un cycle en degré d_k qui n'a jamais été détruit ensuite ;

ce simplexe engendre donc l'intervalle de persistance $[k, +\infty)$ de ce cycle dans le code-barres en degré d_k .

Références

- Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2009). *Introduction to Algorithms*. MIT Press, Cambridge, MA, 3rd edition.
- Crawley-Boevey, W. (2015). Decomposition of pointwise finite-dimensional persistence modules. *Journal of Algebra and Its Applications*, 14(05) :1550066.
- Dey, T. K. and Wang, Y. (2022). *Computational topology for data analysis*. Cambridge University Press.