Une construction du groupe de Mathieu M_{12} à partir d'un jeu de taquin sur $\mathbb{P}^2(\mathbb{F}_3)$

Mathilde Daudet

Stage encadré par Jérôme Germoni

Stage effectué entre mai et juillet 2024 à l'Institut Camille Jordan

Table des matières

Introduction

Le groupe de Mathieu M_{12} est un groupe fini simple de cardinal 95 040. Il est généralement défini comme groupe d'automorphismes d'un système de Steiner $S(5,6,12)$. Dans ce rapport, nous étudions une construction de ce groupe développée par Conway, Elkies et Martin dans [CEM05]. On y construit M_{12} à partir d'un jeu de taquin sur le plan projectif $\mathbb{P}^2(\mathbb{F}_3)$, de façon plus naturelle (car sans véritable choix) que les constructions habituelles où des générateurs du groupe M_{12} sont donnés sans que l'on sache d'où ils viennent. Dans ce rapport, nous allons explorer le « paysage » présenté en figure 1, afin de comprendre les liens entre M_{12}, le système de Steiner $S(5,6,12)$, le code de Golay ternaire étendu, les matrices monomiales, et les matrices d'Hadamard 12×12.

Nous nous intéresserons dans un premier temps aux systèmes de Steiner, afin de pouvoir définir le groupe de Mathieu M_{12}. Nous décrirons aussi un outil permettant de mieux comprendre la combinatoire d'un système de Steiner en comptant certains blocs : le triangle d'intersection, décrit par Cameron dans [Cam15].

Nous construirons ensuite des codes de Golay afin d'obtenir une réalisation du système de Steiner $S(5,6,12)$ (dont on admet l'unicité afin de pouvoir définir M_{12}), et nous présenterons une première réalisation concrète du groupe de Mathieu M_{12} obtenue informatiquement.

Enfin, nous décrirons trois versions d'un jeu de taquin sur le plan projectif $\mathbb{P}^2(\mathbb{F}_3)$, qui nous permettrons de retrouver le groupe M_{12} (de plusieurs manières) et le groupe d'automorphismes du code de Golay ternaire étendu, et de montrer que celui-ci est une extension double de M_{12}. Nous décrirons aussi comment le jeu de taquin permet d'obtenir un automorphisme extérieur de M_{12}.

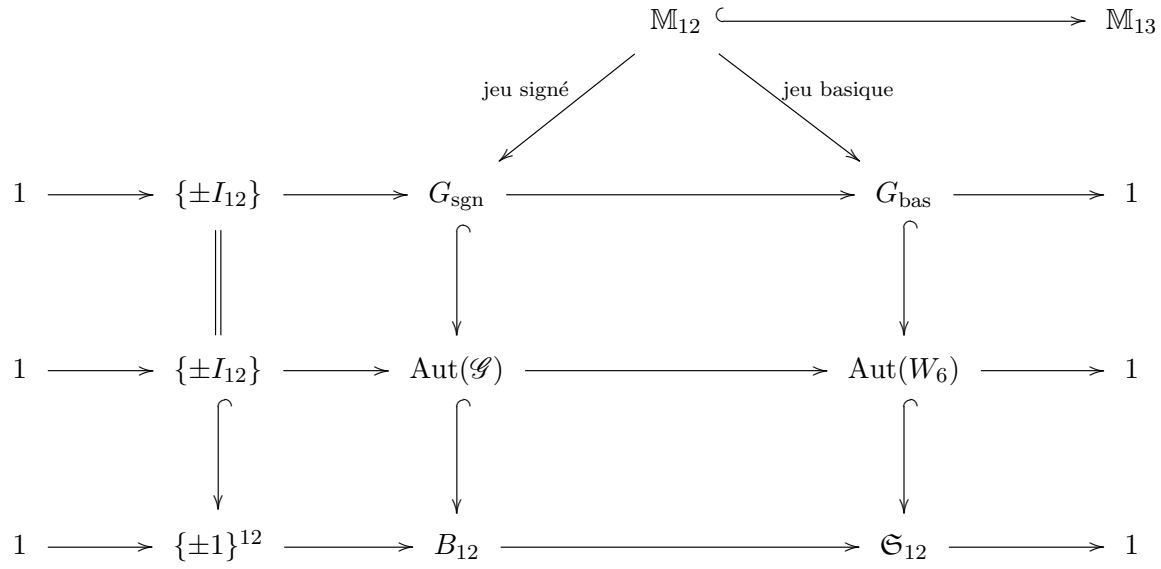

FIGURE 1 – Le « paysage » autour de M_{12}

La figure 1 fait intervenir un certain nombre d'objets, qui seront décrits en détails dans la suite de ce rapport, et dont voici une première liste :

$\mathbb{M}_{12}$	le groupe des chemins fermés du jeu de taquin ;
$\mathbb{M}_{13}$	le groupoïde des chemins du jeu de taquin ;
G_{sgn}	le groupe des matrices associées au jeu signé ;
G_{bas}	le groupe des permutations associées au jeu basique ;
$\mathscr{G}$	le code de Golay ternaire étendu ;
W_6	le système de Steiner $S(5,6,12)$;
B_{12}	le groupe des matrices monomiales 12×12 sur $\mathbb{F}_3$.

1 Systèmes de Steiner

1.1 Définition

Définition. Soient t, k, n trois entiers tels que $0 < t < k < n$. On appelle système de Steiner $S(t, k, n)$ sur un ensemble I à n éléments la donnée d'un ensemble de parties de I de cardinal k, appelées blocs, satisfaisant à la propriété que toute partie à t éléments de I est contenue dans un unique bloc.

Exemple. Considérons les ensembles $I = \{1, 2, 3, 4, 5, 6, 7\}$ et

$$S = \{\{2, 4, 6\}, \{1, 4, 5\}, \{3, 4, 7\}, \{1, 2, 3\}, \{2, 5, 7\}, \{1, 6, 7\}, \{3, 5, 6\}\}$$

Alors S est un système de Steiner $S(2, 3, 7)$ sur I. Il suffit de vérifier que toute paire de points de I appartient à un unique élément de S. La table 1 résume cette vérification.

Paire de points de I	Bloc auquel elle appartient
$\{2, 4\}, \{2, 6\}, \{4, 6\}$	$\{2, 4, 6\}$
$\{1, 4\}, \{1, 5\}, \{4, 5\}$	$\{1, 4, 5\}$
$\{3, 4\}, \{3, 7\}, \{4, 7\}$	$\{3, 4, 7\}$
$\{1, 2\}, \{1, 3\}, \{2, 3\}$	$\{1, 2, 3\}$
$\{2, 5\}, \{2, 7\}, \{5, 7\}$	$\{2, 5, 7\}$
$\{1, 6\}, \{1, 7\}, \{6, 7\}$	$\{1, 6, 7\}$
$\{3, 5\}, \{3, 6\}, \{5, 6\}$	$\{3, 5, 6\}$

TABLE 1 – Correspondance entre paires de points et blocs

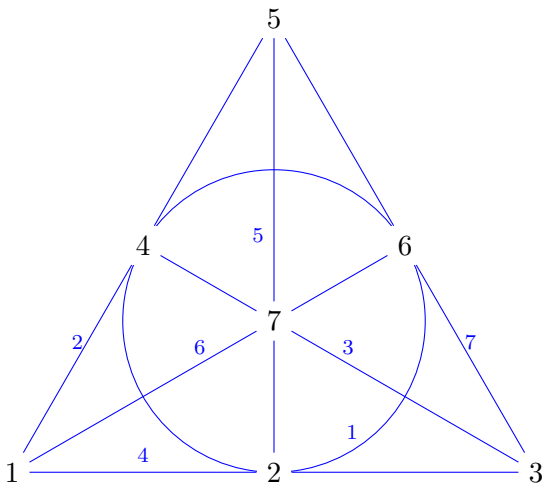

FIGURE 2 – Une représentation d'un système de Steiner $S(2, 3, 7)$

1.2 Plans projectifs finis

Définition. Soit $\mathscr{P}$ un ensemble de points et $\mathscr{L}$ un ensemble de parties de $\mathscr{P}$ appelées droites. Pour tout $p \in \mathscr{P}$, on notera $\mathscr{L}(p)$ l'ensemble des éléments de $\mathscr{L}$ qui contiennent p. On dit que $(\mathscr{P}, \mathscr{L})$ est un plan projectif d'ordre n si les conditions suivantes sont satisfaites :

1. $|\mathscr{P}| = |\mathscr{L}| = n^2 + n + 1$;
2. pour tout $p \in \mathscr{P}$ et tout $l \in \mathscr{L}$, $|\mathscr{L}(p)| = |l| = n + 1$;
3. deux points distincts $p, q \in \mathscr{P}$ déterminent une unique droite, *i.e.* $|\mathscr{L}(p) \cap \mathscr{L}(q)| = 1$;
4. deux droites distinctes $l, m \in \mathscr{L}$ s'intersectent en un unique point, *i.e.* $|l \cap m| = 1$.

Exemple. Le système de Steiner $S(2, 3, 7)$ défini plus haut est en fait un plan projectif fini : il suffit de poser $\mathscr{P} = \{1, 2, 3, 4, 5, 6, 7\}$ et $\mathscr{L} = S$ comme dans l'exemple précédent.

On va voir qu'un plan projectif fini donne lieu à un système de Steiner (et même à deux systèmes de Steiner). Si $(\mathscr{P}, \mathscr{L})$ est un plan projectif d'ordre n tel que défini plus haut, alors $\mathscr{L}$ est un système de Steiner $S(2, n+1, n^2+n+1)$ sur $\mathscr{P}$. En effet, $\mathscr{L}$ est bien un ensemble de parties de cardinal $n+1$ de $\mathscr{P}$, et deux points distincts appartiennent bien à une unique droite. Les axiomes du plan projectif fini font aussi de $\{\mathscr{L}(p) \mid p \in \mathscr{P}\}$ un système de Steiner sur $\mathscr{L}$.

Proposition 1. *Soit $q \in \mathbb{N}^*$. Un système de Steiner $S(2, q+1, q^2+q+q)$ donne deux plans projectifs finis.*

Démonstration. Soit S un système de Steiner $S(2, q+1, q^2+q+q)$ sur un ensemble I. Posons

$$\mathscr{P} = I \text{ et } \mathscr{L} = \{b \mid b \in S\}.$$

On a bien $|\mathscr{P}| = q^2+q+1$. Le fait que pour tout $l \in \mathscr{L}$, $|l| = q+1$ provient de la définition des blocs du système de Steiner $S(2, q+1, q^2+q+1)$. La propriété du système de Steiner permet aussi d'affirmer que deux points distincts déterminent une unique droite, puisque les droites sont les blocs du système de Steiner. Avec les notations et résultats de la partie 1.4, on a $|\mathscr{L}| = t_{0,0} = q^2+q+1$ et, pour tout p, on a $|\mathscr{L}(p)| = t_{1,1} = q+1$. Il reste à voir que deux droites distinctes s'intersectent en un unique point. Si deux droites s'intersectent en deux points distincts, alors chacune de ces droites est un bloc qui contient ces deux points. Or il existe un seul tel bloc. On a bien montré que $(\mathscr{P}, \mathscr{L})$ est un plan projectif fini d'ordre q.

On peut aussi poser

$$\mathscr{P}' = \mathscr{L} \text{ et } \mathscr{L}' = \{\mathscr{L}(p) \mid p \in \mathscr{P}\}$$

et montrer de même que $(\mathscr{P}', \mathscr{L}')$ est assi un plan projectif fini d'ordre q. □

1.3 Plan projectif sur $\mathbb{F}_q$

Proposition 2. *Soit q une puissance d'un nombre premier et $\mathbb{F}_q$ le corps fini de cardinal q. On peut définir le plan projectif $\mathbb{P}^2(\mathbb{F}_q)$, i.e.des droites vectorielles de ^3_q. Alors $(\mathbb{P}^2(\mathbb{F}_q), \mathscr{L})$ est un plan projectif fini, au sens de la définition donnée en partie 1.2, où $\mathscr{L}$ est l'ensemble des droites de $\mathbb{P}^2(\mathbb{F}_q)$, i.e. des plans vectoriels de $\mathbb{F}_q^3$.*

Cette proposition est un corollaire immédiat de la proposition 3 ci-dessous et de la proposition 1. Il existe aussi des plans projectifs finis qui ne sont pas des $\mathbb{P}^2(\mathbb{F}_q)$, mais on ne s'y intéressera pas ici.

Proposition 3. *Soit q une puissance d'un nombre premier et soit $\mathbb{F}_q$ le corps fini de cardinal q. Alors le plan projectif $\mathbb{P}^2(\mathbb{F}_q)$ donne lieu à un système de Steiner $S(2, q+1, q^2+q+1)$ dont les blocs sont les droites de $\mathbb{P}^2(\mathbb{F}_q)$, i.e. les plans vectoriels de $\mathbb{F}_q^3$.*

Démonstration. On vérifie que $\mathbb{P}^2(\mathbb{F}_q)$ compte bien q^2+q+1 points : il s'agit de compter les droites vectorielles de $\mathbb{F}_q^3$. On note (x, y, z) les coordonnées d'un point de $\mathbb{F}_q^3$ dans sa base canonique. On partitionne l'ensemble des droites vectorielles de $\mathbb{F}_q^3$ en deux : les droites incluses dans le plan $z = 0$ et les autres droites. Une droite qui n'est pas incluse dans le plan $z = 0$ contient un unique point de la forme $(x, y, 1)$, et chaque triplet $(x, y, 1)$ donne lieu à une telle droite. Il y a donc q^2 telles droites. Les droites incluses dans le plan $z = 0$ peuvent être comptées comme suit : soit une telle droite est la droite $z = y = 0$, soit elle contient un unique point $(x, 1, 0)$, et chaque $(x, 1, 0)$ donne lieu à une telle droite. Il y a donc $q+1$ droites incluses dans la plan $z = 0$. Il y a donc bien q^2+q+1 droites vectorielles de $\mathbb{F}_q^3$, donc q^2+q+1 points dans $\mathbb{P}^2(\mathbb{F}_q)$.

Soit S l'ensemble des droites de $\mathbb{P}^2(\mathbb{F}_q)$, *i.e.* l'ensemble des plans vectoriels de $\mathbb{F}_q^3$. Chaque élément de S contient $q+1$ points de $\mathbb{P}^2(\mathbb{F}_q)$: c'est le même raisonnement que lorsqu'on a compté les droites vectorielles incluses dans le plan $z = 0$ ci-dessus.

Il reste à vérifier que deux points distincts de $\mathbb{P}^2(\mathbb{F}_q)$ appartiennent à une unique droite. Cela vient simplement du fait que deux droites vectorielles distinctes de $\mathbb{F}_q^3$ sont contenues dans un seul plan vectoriel, celui qu'elles engendrent. □

Exemple. Le système de Steiner $S(2,3,7)$ donné comme exemple plus haut est en fait celui qui résulte du plan projectif fini $\mathbb{P}^2(\mathbb{F}_2)$. Les points de I sont les vecteurs de $\mathbb{F}_2^3 \setminus \{0\}$. On note abc le vecteur de coordonnées $(a,b,c) \in \mathbb{F}_2^3 \setminus \{0\}$. Chaque droite est déterminée par son équation. Le troisième point d'une droite (pq), où p et q sont deux points de $\mathbb{P}^2(\mathbb{F}_2)$, est le point $p+q$. On représente ce plan projectif fini dans la figure 3.

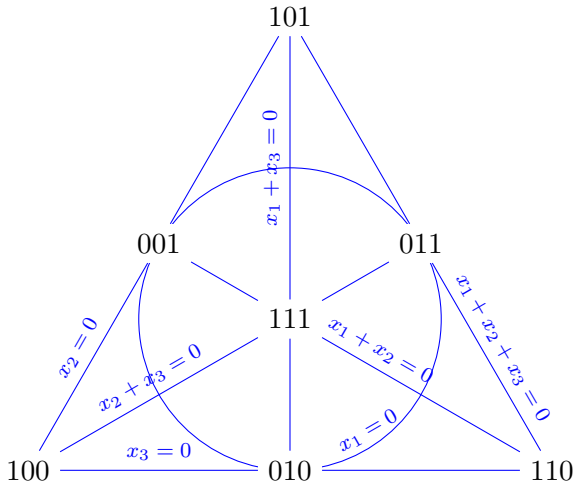

FIGURE 3 – Plan de Fano : vecteurs et équations

1.4 Triangle d'intersection

Pour t, k, n donnés, on peut considérer le triangle d'intersection d'un système de Steiner $S(t,k,n)$, qui est la famille d'entiers $(t_{i,j})_{0 \le j \le i \le k}$ définie de la façon suivante. On fixe un bloc $\{x_1, \dots, x_k\}$. Pour $0 \le i \le k$ et $0 \le j \le i$, on note $t_{i,j}$ le nombre de blocs du système de Steiner qui contiennent les points $x_1, \dots, x_j$ mais pas $x_{j+1}, \dots, x_i$.

Proposition 4. *1. Les nombres $t_{i,j}$ ne dépendent que du système de Steiner et pas du bloc $x_1, \dots, x_k$ considéré ni de son énumération ;*

2. $t_{i,i} = \begin{cases} \frac{\binom{n-i}{t-i}}{\binom{k-i}{t-i}} & \text{si } i \le t \\ 1 & \text{si } t \le i \le k \, ; \end{cases}$

3. le triangle d'intersection satisfait à une formule de Pascal : pour $0 \le i < k$, et $0 \le j \le i$,

$$t_{i,j} = t_{i+1,j} + t_{i+1,j+1}.$$

Démonstration. Le nombre $t_{i,i}$ est au nombre de blocs qui contiennent les points $x_1, \dots, x_i$. Or un bloc est déterminé par t points, et i points sont ici déjà fixés, donc il y a $\binom{n-i}{t-i}$ choix possibles. On a ici compté chaque bloc $\binom{k-i}{t-i}$ fois. D'où la formule 2. Si $t \le i \le k$, la propriété du système de Steiner et le fait que $\{x_1, \dots, x_k\}$ est un bloc assurent que $t_{i,i} = 1$.

On raisonne par récurrence sur $i - j$ pour montrer que $t_{i,j}$ ne dépend pas de l'ordre des points du bloc $\{x_1, \dots, x_k\}$. Pour $i - j = 0$, on vient de le voir.

Soit $0 \le r \le k-1$. On suppose que les $t_{i,j}$ pour $i - j = r$ ne dépendent pas de l'ordre des points du bloc $\{x_1, x_2, x_3, x_4, x_5, x_6\}$. Soient i, j tels que $i - j = r + 1$. Comptons les blocs qui contiennent $x_1, \dots, x_j$ mais pas $x_{j+1}, \dots, x_i$. Ce sont les blocs qui contiennent $x_1, \dots, x_j$ mais pas $x_{j+1}, \dots, x_{i-1}$ (il y en a $t_{i-1,j}$) parmi lesquels on retire les blocs contenant $x_1, \dots, x_j, x_i$ mais pas $x_{j+1}, \dots, x_{i-1}$ (il y en a $t_{i,j+1}$ vu que $i - (j+1) = r$). Ce nombre est donc $t_{i,j} = t_{i-1,j} - t_{i,j+1}$, qui ne dépend pas de l'ordre des points du bloc $\{x_1, \dots, x_k\}$.

On a aussi montré le point 3.

Enfin, le triangle d'intersection ne dépend pas du bloc $\{x_1, \dots, x_k\}$ choisi, puisque le calcul de ses coefficients s'effectue à partir de coefficients binomiaux, qui ne dépendent que de i, j, et des paramètres du système de Steiner. □

Le triangle d'intersection d'un système de Steiner $S(2,3,7)$ est représenté en table 2. On peut vérifier à la main ces coefficients à l'aide de la figure 3 ou de la table 1.

			7			
		4		3		
	2		2		1	
0		2		0		1

TABLE 2 – Triangle d'intersection d'un $S(2,3,7)$

Ce triangle est en fait un cas particulier de celui d'un système de Steiner $S(2,q+1,q^2+q+1)$, qui est donné en table 3.

						q^2+q+1								
					q^2				$q+1$					
				$q(q-1)$				q				1		
			$q(q-2)$				q				0		1	
		$\iddots$			$\iddots$			$\iddots$			$\ddots$		$\ddots$	
0			q			0			$\dots$			0		1

TABLE 3 – Triangle d'intersection d'un $S(2,q+1,q^2+q+1)$

On trace en table 4 le triangle d'intersection d'un système de Steiner $S(5,6,12)$ (dont on admet provisoirement l'existence), qui sera utile par la suite.

										132												
								66				66										
						30				36				30								
				12				18				18				12						
			4			8				10				8				4				
		1			3			5				5				3			1			
	1			0			3			2			3			0			1			

TABLE 4 – Triangle d'intersection d'un $S(5,6,12)$

On remarque sur ce triangle que le complémentaire d'un bloc est un bloc : c'est exactement ce dit le coefficient $t_{6,0} = 1$.

1.5 Automorphismes d'un système de Steiner

Définition. Soient S et S' deux systèmes de Steiner $S(t,k,n)$ définis respectivement sur des ensembles de points I et I'. Un isomorphisme de systèmes de Steiner entre S et S' est une bijection de I vers I' qui induit une bijection de S vers S'. Un automorphisme du système de Steiner S est un automorphisme de I qui induit une bijection de S dans S.

Définition. On dit qu'un système de Steiner $S(t,k,n)$ est unique à isomorphisme près, ou unique, si tous les systèmes de Steiner $S(t,k,n)$ S et S' sont isomorphes.

Proposition 5. *Le système de Steiner $S(2,3,7)$ est unique et son groupe d'automorphismes est $\mathrm{GL}_3(\mathbb{F}_2)$, et il est d'ordre* 168.

Démonstration. Montrons que tout système de Steiner $S(2,3,7)$ est isomorphe à celui donné par le plan projectif $\mathbb{P}^2(\mathbb{F}_2)$ décrit en table 1 et en figure 2. Un isomorphisme entre ces deux systèmes de Steiner est déterminé par l'image de trois points non alignés (*i.e.* l'image d'un ovale, qui est elle-même un ovale). Il existe un ovale car pour tous points distinct a et b, il existe un unique autre point sur la droite passant par a et b, donc il y a $7-3=4$ choix de point c tels que a, b et c ne sont pas colinéaires. Si on a l'image de trois points non alignés, on a aussi les images des troisièmes points des

droites reliant deux des points de notre ovale. Il ne reste plus qu'un septième point, dont on a donc aussi l'image. On vérifie que la bijection ainsi définie définit bien une bijection sur les droites. Ainsi, le système de Steiner $S(2,3,7)$ est unique à isomorphisme près.

Montrons maintenant que son groupe d'automorphismes est $\mathrm{GL}_3(\mathbb{F}_2)$. Ce qui précède montre que le nombre d'automorphismes est le nombre d'ovales, donc $7 \times 6 \times 4$.

La donnée de trois points non alignés est exactement celle d'une base de $\mathbb{F}_2^3$. En effet, trois vecteurs sont linéairement indépendants si et seulement s'ils ne sont pas alignés. Puisqu'un ovale correspond exactement à un base de $\mathbb{F}_2^3$, la donnée d'un ovale comme image d'un ovale donne bien un endomorphisme de $\mathbb{F}_2^3$ (défini par l'image d'une base), et c'est bien un automorphisme (puisque l'image de cette base est une base).

Ainsi, le groupe des automorphismes de ce système de Steiner est $\mathrm{GL}_3(\mathbb{F}_2)$, et son ordre est 168. □

2 Le groupe de Mathieu M_{12}

On admet provisoirement l'existence, ainsi que définitivement l'unicité à isomorphisme près, d'un système de Steiner $S(5,6,12)$.

Définition. On appelle groupe de Mathieu 12 et on note M_{12} le groupe des automorphismes du système de Steiner $S(5,6,12)$.

On peut construire le groupe d'automorphismes du système de Steiner $S(5,6,12)$. On note pour cela S le système de Steiner et $\mathscr{P} \setminus \{12\} = \{0,1,\dots,11\}$ l'ensemble sur lequel il est défini [1].

Pour construire un tel automorphisme, il suffit de choisir les images respectives (y_0,y_1,y_2,y_3,y_4) de 5 points (x_0,x_1,x_2,x_3,x_4). Le théorème suivant, qui justifie ce fait, sera démontré à la fin de la partie 2.

Theorème 6. *Le groupe* $\mathrm{Aut}(S(5,6,12))$ *des automorphismes du système de Steiner* $S(5,6,12)$ *est simplement 5 fois transitif, i.e. pour tous* $x_0,\dots,x_4 \in \mathscr{P}$ *distincts et tous* $y_0,\dots,y_4 \in \mathscr{P}$ *distincts, il existe un unique automorphisme* $\sigma \in \mathrm{Aut}(S(5,6,12))$ *tel que*

$$\forall i \in \{0,\dots,4\},\ \sigma(x_i) = y_i\ ;$$

et de plus, $|\mathrm{Aut}(S(5,6,12))| = 95040$.

On remarque déjà que le point x_5 qui complète $\{x_0,x_1,x_2,x_3,x_4\}$ en un bloc b_0 est nécessairement envoyé sur le point y_5 qui complète $\{y_0,y_1,y_2,y_3,y_4\}$ en un bloc. Il reste à déterminer les images des points de $b_1 = \mathscr{P} \setminus b_0$.

Définition. Soit b un bloc de S. Un synthème est une partition de b en 3 paires.

Lemme 7. *Il y a* 15 *synthèmes dans un bloc* b *fixé.*

Démonstration. On a $\binom{6}{2}\binom{4}{2}\binom{2}{2}/3! = 15$. □

Notation. Soit b un bloc de S. On notera $\mathscr{Q}_b$ l'ensemble des quadruplets de points de b et $\mathscr{S}_b$ l'ensemble des synthèmes de b.

Lemme 8. *Soit* $b_0 \in S$ *un bloc. On pose* $b_1 = \mathscr{P} \setminus b_0$. *Alors*

1. *l'application* TP_{b_0}, *qui à un quadruplet* q *inclus dans* b_0 *associe l'ensemble des paires* π *incluses dans* b_1 *telles que* $q \cup \pi$ *est un bloc* [2], *est une application de* $\mathscr{Q}_{b_0}$ *vers l'ensemble* $\mathscr{S}_{b_1}$ *des synthèmes de* b_1 ;
2. *pour tout automorphisme* f *du système de Steiner* S, *l' application* $\mathrm{TP}_{b_0} : \mathscr{Q}_{b_0} \to \mathscr{S}_{b_1}$ *satisfait*

$$f(\mathrm{TP}_{b_0}(q)) = \mathrm{TP}_{f(b_0)}(f(q)),\ \textit{pour tout}\ q \in \mathscr{Q}_{b_0}.$$

1. Ici, $\mathscr{P} = \{0,1,\dots,12\}$ comme dans la partie 3.1.
2. Cette application est notée TP_{b_0} pour « triplet de paires ».

Démonstration. Le premier point découle de $t_{6,4} = 3$ dans le triangle d'intersection du système de Steiner $S(5, 6, 12)$ en table 3. Le second point provient du fait qu'un automorphisme f de S envoie un bloc sur un bloc. □

Pour déterminer l'image des points de b_1, on procède de la façon suivante : Pour une paire π d'éléments de b_1, on choisit q et q' deux quadruplets de b_0 tels que π est l'unique paire commune des images $\mathrm{TP}_{b_0}(q)$ et $\mathrm{TP}_{b_0}(q')$. Puisque l'image des points de b_0 est déjà connue, on peut appliquer la fonction $\mathrm{TP}_{f(b_0)}$ à l'image, et $\mathrm{TP}_{f(b_0)}(f(q))$ et $\mathrm{TP}_{f(b_0)}(f(q'))$ auront une seule paire commune. Cette paire commune est nécessairement l'image de π par f.

Il reste à déterminer l'image de chaque point de b_1. Pour un point x de b_1, on choisit deux autres points x' et x'' distincts dans b_1, et on sait déterminer les paires images de $\{x, x'\}$ et $\{x, x''\}$. Le point commun aux deux paires images est nécessairement l'image de x.

Cette méthode fonctionne car, une paire $\pi \subset b_1$ étant donnée, il est possible de choisir q et q' deux quadruplets de b_0 tels que π est l'unique paire commune des images $\mathrm{TP}_{b_0}(q)$ et $\mathrm{TP}_{b_0}(q')$. C'est le programme informatique qui, puisqu'il aboutit, justifie la méthode (voir la partie 7 du code en annexe).

Exemple. Construisons un automorphisme f du système de Steiner S qui envoie les points 0, 1, 2, 3, et 4 sur les images décrites dans la table 5. On utilisera la réalisation du système de Steiner S calculée par ordinateur pour savoir quels sont les blocs.

Point de I	0	1	2	3	4
Image par f	4	7	2	10	9

Table 5 – Exemple de correspondance entre les points de I et leurs images

Le sixième point de l'unique bloc qui contient $\{0, 1, 2, 3, 4\}$ est 5. Il doit être envoyé par f sur le sixième point de l'unique bloc qui contient $\{4, 7, 2, 10, 9\}$, donc sur 8.

On pose $b_0 = \{0, 1, 2, 3, 4, 5\}$ et $B_0 = \{4, 7, 2, 10, 9, 8\}$. Soient $b_1 = \mathscr{P} \setminus b_0$ et $B_1 = \mathscr{P} \setminus B_0$. On a donc $b_1 = \{6, 7, 8, 9, 10, 11\}$ et $B_1 = \{0, 1, 3, 5, 6, 11\}$.

On considère le quadruplet $q = \{0, 1, 3, 5\}$ de b_0. On a $\mathrm{TP}_{b_0}(q) = \{\{6, 11\}, \{7, 8\}, \{9, 10\}\}$. On considère aussi le quadruplet $q' = \{0, 1, 2, 4\}$ de b_0. On a $\mathrm{TP}_{b_0}(q') = \{\{6, 11\}, \{7, 9\}, \{8, 10\}\}$. La seule paire commune à $\mathrm{TP}_{b_0}(q)$ et $\mathrm{TP}_{b_0}(q')$ est $\{6, 11\}$. On a aussi $f(q) = \{4, 7, 10, 8\}$, et les paires associées correspondantes $\mathrm{TP}_{B_0}(f(q)) = \{\{0, 6\}, \{1, 3\}, \{5, 11\}\}$. On a aussi $f(q') = \{4, 7, 2, 9\}$, et les paires associées correspondantes $\mathrm{TP}_{B_0}(f(q')) = \{\{0, 6\}, \{1, 11\}, \{3, 5\}\}$. La seule paire commune est donc $\{0, 6\}$. Donc f envoie $\{6, 11\}$ sur $\{0, 6\}$.

On effectue le même travail pour la paire $\{6, 9\}$ de b_1. On trouve que cette paire est envoyée sur la paire $\{5, 6\}$. Ainsi, 6 est envoyé sur 6 par f.

On peut raisonner de la même façon pour déterminer les images des autres points de b_1. On obtient le résultat décrit par la table 6.

Point de I	0	1	2	3	4	5	6	7	8	9	10	11
Image par f	4	7	2	10	9	8	6	1	3	5	11	0

Table 6 – Un exemple d'automorphisme

On vérifie informatiquement que chaque bijection ainsi construite est un automorphisme, *i.e.*envoie bien les blocs sur les blocs (voir la partie 7 du code) : ce sont bien des automorphismes du système de Steiner. Il y a donc $\mathsf{A}_{12}^5 = 12 \times \cdots \times 8 = 95\,040$ automorphismes du système de Steiner $S(5, 6, 12)$.

On peut maintenant montrer le théorème 6.

Théorème 6. *Le groupe* $\mathrm{Aut}(S(5, 6, 12))$ *des automorphismes du système de Steiner* $S(5, 6, 12)$ *est simplement 5 fois transitif, i.e. pour tous* $x_0, \ldots, x_4 \in \mathscr{P}$ *distincts et tous* $y_0, \ldots, y_4 \in \mathscr{P}$ *distincts, il existe un unique automorphisme* $\sigma \in \mathrm{Aut}(S(5, 6, 12))$ *tel que*

$$\forall i \in \{0, \ldots, 4\},\ \sigma(x_i) = y_i\ ;$$

et de plus, $|\operatorname{Aut}(S(5,6,12))| = 95040$.

Démonstration. On se réfère aux fonctions définies dans la partie 7 du code. L'algorithme qui calcule l'automorphisme qui envoie les x_i sur les y_i aboutit pour $(x_0, \ldots, x_4) = (0, \ldots, 4)$ et pour tout choix des y_i. D'où l'existence d'un unique tel automorphisme pour chaque tel choix. Ainsi, si on fixe des points $x_0, \ldots, x_4 \in \mathscr{P}$, un automorphisme de $S(5,6,12)$ est uniquement déterminé par les images respectives $y_0, \ldots, _4$ de $x_0, \ldots, x_4$. On a donc $|\operatorname{Aut}(S(5,6,12))| = 12 \times 11 \times 10 \times 9 \times 8 = 95040$. □

3 Plan projectif $\mathbb{P}^2(\mathbb{F}_3)$, codes de Golay et construction de $S(5,6,12)$

3.1 Plan projectif $\mathbb{P}^2(\mathbb{F}_3)$

On construit le plan projectif fini $\mathbb{P}^2(\mathbb{F}_3)$.

Pour cela, on note $\mathbb{F}_3 = \{0,1,2\}$ le corps de cardinal 3 et on considère $\mathbb{F}_3^3$. Un point de $\mathbb{P}^2(\mathbb{F}_3)$ est une droite vectorielle de $\mathbb{F}_3^3$. On choisit de représenter les éléments de $\mathbb{P}^2(\mathbb{F}_3)$ par les points de coordonnées homogènes $[a : b : 1]$, $[a : 1 : 0]$, et enfin $[1 : 0 : 0]$, où a, b sont dans $\mathbb{F}_3$. On a ainsi les 13 points de $\mathbb{P}^2(\mathbb{F}_3)$.

On définit un ordre sur $\{0,1,2\}^3$ grâce à l'écriture en base trois : on dit que $(a,b,c) \leq (d,e,f)$ si $a + 3b + 9c \leq d + 3e + 9f$. Comme on a choisi des triplets pour décrire les points de $\mathbb{P}^2(\mathbb{F}_3)$, cela induit un ordre sur $\mathbb{P}^2(\mathbb{F}_3)$ et une bijection de $\mathscr{P} = \{0, 1, \ldots, 12\}$ sur $\mathbb{P}^2(\mathbb{F}_3)$, qui est résumée dans la table 7.

Nom du point dans $\mathscr{P}$	0	1	2	3	4	5	6
Coordonnées dans $\mathbb{P}^2(\mathbb{F}_3)$	[1 : 0 : 0]	[0 : 1 : 0]	[1 : 1 : 0]	[2 : 1 : 0]	[0 : 0 : 1]	[1 : 0 : 1]	[2 : 0 : 1]
Nom du point dans $\mathscr{P}$	7	8	9	10	11	12	
Coordonnées dans $\mathbb{P}^2(\mathbb{F}_3)$	[0 : 1 : 1]	[1 : 1 : 1]	[2 : 1 : 1]	[0 : 2 : 1]	[1 : 2 : 1]	[2 : 2 : 1]	

TABLE 7 – Numérotation de $\mathbb{P}^2(\mathbb{F}_3)$

On décrit ensuite l'ensemble $\mathscr{L}$ des droites du plan projectif combinatoire $\mathbb{P}^2(\mathbb{F}_3)$ de la façon suivante. On introduit la forme bilinéaire standard sur $\mathbb{F}_3^3$: $\langle v, w \rangle = \sum_{i=0}^{2} v_i w_i$ pour $v, w \in \mathbb{F}_3^3$. Pour $p \in \mathscr{P}$, on note $\overline{p} = l_p$ l'ensemble des $q \in \mathscr{P}$ tels que les points d'indices p et q correspondent à des droites vectorielles orthogonales. La réunion des droites vectorielles d'indice q pour q dans l_p forme une droite projective et toute droite projective est de cette forme. On note $\mathscr{L}$ l'ensemble des l_p pour p dans $\mathscr{P}$.

On obtient la numérotation suivante :

$$\mathscr{L} = \{\{1,4,7,10\}, \{0,4,5,6\}, \{3,4,9,11\}, \{2,4,8,12\}, \{0,1,2,3\}, \{1,6,9,12\}, \{1,5,8,11\},$$
$$\{0,10,11,12\}, \{3,6,8,10\}, \{2,5,9,10\}, \{0,7,8,9\}, \{2,6,7,11\}, \{3,5,7,12\}\}.$$

Cette construction a été reproduite dans la partie 1 du code.

Remarque. La numérotation est auto-duale : en notant $\mathscr{L} = \{\overline{0}, \overline{1}, \ldots, \overline{12}\}$ dans l'ordre de la numérotation ci-dessus, on a, pour tous $x, y \in \mathscr{P}$, $x \in \overline{y}$ si et seulement si $y \in \overline{x}$.

3.2 Codes de Golay et construction de $S(5,6,12)$

Dans cette partie, on utilisera le vocabulaire des codes défini en annexe 1.

Soit $X = \mathbb{F}_3^{\mathscr{P}}$, dont on notera $\{x_p \mid p \in \mathscr{P}\}$ la base canonique : c'est un espace vectoriel de dimension 13 sur le corps $\mathbb{F}_3$. Les vecteurs de X seront écrits sous la forme $v = \sum_{p \in \mathscr{P}} v_p x_p$, où les v_p sont dans $\mathbb{F}_3$. On munit X du produit scalaire défini par

$$v \cdot w = \sum_{p \in \mathscr{P}} v_p w_p. \tag{1}$$

Pour $l \in \mathscr{L}$, on définit

$$h_l = \sum_{p \in l} x_p.$$

Soit $\mathscr{C} \subset X$ le sous-espace vectoriel engendré par les $\{h_l\}_{l \in \mathscr{L}}$.

Soit $\mathscr{C}' = \{c \in \mathscr{C} \mid \sum_{p \in \mathscr{P}} c_p = 0\}$.

On construit aussi ces codes par un programme informatique (voir section 1 du code).

Proposition 9. *Soit $c \in \mathscr{C}$. Alors :*

1. $\sum_{p \in \mathscr{P}} c_p^2 = (\sum_{p \in \mathscr{P}} c_p)^2$.
2. $\mathrm{wt}(c) \equiv 0$ *ou* $1 \pmod 3$.
3. $c \in \mathscr{C}'$ *ssi* $\mathrm{wt}(c) \equiv 0 \pmod 3$.
4. *Pour tout $l \in \mathscr{L}$,*
$$\sum_{p \in \mathscr{P}} c_p = \sum_{p \in l} c_p.$$
5. $\mathscr{C}' = \mathscr{C}^\perp$, *pour le produit scalaire défini par l'équation 1.*
6. $\dim \mathscr{C} = 7$ *et* $\dim \mathscr{C}' = 6$.
7. $\mathrm{wt}_{\min}(\mathscr{C}) = 4$ *et* $\mathrm{wt}_{\min}(\mathscr{C}') = 6$. *De plus, les mots de poids* 4 *dans $\mathscr{C}$ sont exactement les* $\{\pm h_l \mid l \in \mathscr{L}\}$.

Démonstration. La démonstration dans [CEM05] est facile à suivre à la main. On choisit ici de vérifier la proposition par ordinateur, ce qui permet de valider l'implémentation. Cette vérification se trouve dans la partie 4 du code. □

Cette proposition est utilisée dans [CEM05] pour reconnaître que les $\mathscr{G}_p$ sont isomorphes au code de Golay ternaire étendu ; cependant, la démonstration repose sur une caractérisation connue mais compliquée dudit code de Golay par Vera Pless dans [Ple89] donc il a semblé plus naturel de le faire par ordinateur, ce qui permet de valider l'implémentation des différents objets.

Pour tout $p \in \mathscr{P}$, on définit un sous-code $\mathscr{C}_p$ de la façon suivante :

$$\mathscr{C}_p = \{c \in \mathscr{C} \mid c_p = -\sum_{q \in \mathscr{P}} c_q\}.$$

Pour $p \in \mathscr{P}$, on identifie $\mathbb{F}_3^{\mathscr{P} \setminus \{p\}}$ avec $\mathbb{F}_3^{12}$ via la bijection croissante $\mathscr{P} \setminus \{p\} \to \{0, \ldots, 11\}$. On définit $\mathscr{G}_p$ comme l'image de $\mathscr{C}_p$ par la projection

$$\pi_p : \mathbb{F}_3^{13} \to \mathbb{F}_3^{12}, (v_q)_{q \in \mathscr{P}} \mapsto (v_q)_{q \neq p}.$$

Proposition 10. *Pour tout $p \in \mathscr{P}$, $\mathscr{G}_p$ est isomorphe au code de Golay ternaire étendu.*

Démonstration. On le montre pour $p = 12$, et on verra plus tard que $\mathscr{G}_p$ et $\mathscr{G}_0$ sont isomorphes pour tout p. On part de la matrice génératrice du code de Golay ternaire étendu et on retrouve la matrice génératrice du code $\mathscr{G}_{12}$ par permutations de lignes ou colonnes et changelent de signe de lignes ou colonnes. Ces calculs sont faits dans la partie 3 du code. □

On souhaite maintenant utiliser les $\mathscr{G}_p$ pour construire un système de Steiner $S(5, 6, 12)$, ce qui permettra d'en montrer l'existence admise jusqu'à présent.

On indique dans la table 8 le nombre de vecteurs de chaque poids dans $\mathscr{G}_{12}$ (qui est en fait valable pour tout $\mathscr{G}_p$). Le tableau a été obtenu en partie 4 du code.

Le seul vecteur de poids 0 est le vecteur nul. Il y a $264 = 2 \times 132$ vecteurs de poids 6 (soit 132 droites) : ils vont permettre la construction d'un système de Steiner $S(5, 6, 12)$. Les vecteurs de poids 12 vont permettre de construire une matrice d'Hadamard.

Poids	0	1	2	3	4	5	6	7	8	9	10	11	12
Nombre de vecteurs	1	0	0	0	0	0	264	0	0	440	0	0	24

TABLE 8 – Poids des vecteurs de $\mathscr{G}_{12}$

Proposition 11. *Soit $p \in \mathscr{P}$. On définit S_p comme l'ensemble des supports des vecteurs de poids 6 dans $\mathscr{G}_p$. Alors S_p est un système de Steiner $S(5,6,12)$ sur $\{0,\dots,11\}$.*

Démonstration. Il suffit de le montrer pour S_{12}, car un isomorphisme de code f préserve le poids des vecteurs, et si τ est la permutation associée à la matrice monomiale f, alors $\mathrm{Supp}(f(v)) = \tau(\mathrm{Supp}(v))$. Par la suite, on notera d'ailleurs $W_6 = S_{12}$ ce système de Steiner. On vérifie informatiquement que W_6 est bien un système de Steiner $S(5,6,12)$. Cette vérification se trouve dans la partie 5. □

Définition. Une matrice d'Hadamard d'ordre n est une matrice carrée à coefficients dans $\{\pm 1\} \subset \mathbb{Z}$ de taille $n \times n$ dont les lignes sont toutes orthogonales entre elles.

Pour $p \in \mathscr{P}$, on note H_p la matrice de taille 12×12 dont les lignes correspondent aux vecteurs de poids 12 du code $\mathscr{G}_p$ dont la coordonnée 0 est 1, où on les lit les $1 \in \mathbb{F}_3$ comme des $1 \in \mathbb{Z}$ et les $2 \in \mathbb{F}_3$ comme des $-1 \in \mathbb{Z}$.

Exemple. La matrice H_{12} est

$$H_{12} = \begin{pmatrix}
1 & 1 & -1 & 1 & 1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 \\
1 & 1 & 1 & -1 & 1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 \\
1 & -1 & -1 & -1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 \\
1 & -1 & 1 & 1 & -1 & 1 & 1 & -1 & 1 & 1 & -1 & 1 \\
1 & -1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 \\
1 & 1 & 1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & 1 & -1 \\
1 & -1 & 1 & 1 & 1 & -1 & 1 & 1 & -1 & 1 & 1 & -1 \\
1 & 1 & -1 & 1 & 1 & -1 & 1 & 1 & 1 & -1 & -1 & 1 \\
1 & -1 & 1 & -1 & 1 & -1 & -1 & -1 & 1 & -1 & 1 & 1 \\
1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\
1 & 1 & -1 & -1 & -1 & -1 & 1 & -1 & -1 & 1 & 1 & 1 \\
1 & -1 & -1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & -1 & -1
\end{pmatrix}.$$

4 Construction de M_{12} à partir d'un jeu de taquin sur $\mathbb{P}^2(\mathbb{F}_3)$

4.1 Jeu basique

On a donné dans la partie 3.1 une numérotation des points et des droites du plan projectif $\mathbb{P}^2(\mathbb{F}_3)$. On rappelle la numérotation des droites dans la table 9. Par construction, cette numérotation est auto-duale : les droites qui contiennent le point x ont les mêmes numéros que les points de la droite $\overline{x}$. Autrement dit, $y \in \overline{x}$ si et seulement si $x \in \overline{y}$.

$$\begin{array}{cccc}
\overline{0} = \{1,4,7,10\} & \overline{1} = \{0,4,5,6\} & \overline{2} = \{3,4,9,11\} & \overline{3} = \{2,4,8,12\} \\
\overline{4} = \{0,1,2,3\} & \overline{5} = \{1,6,9,12\} & \overline{6} = \{1,5,8,11\} & \overline{7} = \{0,10,11,12\} \\
\overline{8} = \{3,6,8,10\} & \overline{9} = \{2,5,9,10\} & \overline{10} = \{0,7,8,9\} & \overline{11} = \{2,6,7,11\} \\
\overline{12} = \{3,5,7,12\} & & &
\end{array}$$

TABLE 9 – Numérotation de $\mathscr{L}$

On décrit un jeu de taquin sur $\mathbb{P}^2(\mathbb{F}_3)$. On commence par placer des pièces numérotées de 1 à 12 sur les points de $\mathbb{P}^2(\mathbb{F}_3)$, en laissant le point 0 vide : c'est le « trou ». On codera par la suite le trou par une pièce numérotée 0.

Un mouvement élémentaire du jeu est alors défini comme suit. Supposons que le trou soit au point p. Soit q un autre point. Soit $l = \{p,q,r,s\}$ l'unique droite contenant p et q. Le mouvement $[p,q]$

consiste alors à déplacer la pièce placée en q vers p, puis à échanger les pièces placées en r et s. Le trou est alors en q, donc le mouvement suivant doit être de la forme $[q, t]$ pour un certain point t.

Une suite de mouvements du jeu (une composition de mouvements élémentaires) peut être entièrement décrite par le chemin emprunté par le trou :

$$[p_0, p_1, \dots, p_n] = [p_{n-1}, p_n] \circ \cdots \circ [p_1, p_0].$$

Par convention, le mouvement $[p, p]$ est trivial. Il y a donc 12 mouvements autorisés à partir de chaque position du jeu.

Le mouvement $[p, q]$ induit la permutation $(p\ q)(r\ s) \in \mathfrak{S}_{\mathscr{P}}$, et un chemin $[p_0, p_1, \dots, p_n]$ induit la permutation $(p_{n-1}\ p_n)(q_n\ r_n) \cdots (p_0\ p_1)(q_1\ r_1)$, où q_i, r_i sont les deux autres points de l'unique droite contenant p_i et p_{i-1}.

Exemple. Considérons le chemin $[0, 6, 12, 1, 8, 0]$. Le premier mouvement $[0, 6]$ induit la permutation $(0\ 6)(4\ 5)$, puisque la droite qui contient 0 et 6 est $\overline{1} = \{0, 4, 5, 6\}$. La permutation associée au chemin entier est

$$(8\ 0)(7\ 9) \circ (1\ 8)(5\ 11) \circ (12\ 1)(6\ 9) \circ (6\ 12)(1\ 9) \circ (0\ 6)(4\ 5) = (1\ 6\ 8)(4\ 11\ 5)(7\ 9\ 12)$$

On dira qu'un chemin $[p_0, p_1, \dots, p_n]$ est fermé si $p_0 = p_n$. Deux chemins sont dits équivalents s'ils induisent la même permutation. On remarque que si p, q, r sont colinéaires, alors $[p, q, r]$ et $[p, r]$ sont équivalents. Tout chemin est donc équivalent à un chemin où trois points consécutifs ne sont jamais colinéaires. Un tel chemin est dit non dégénéré.

L'ensemble des chemins à équivalence près, muni de la concaténation de chemins, forme un groupoïde [3] que l'on note $\mathbb{M}_{13}$. L'ensemble des chemins fermés $[p_0, \dots, p_n]$ avec $p_0 = p_n = 12$ forme un groupe que l'on note $\mathbb{M}_{12}$ [4] On a alors un morphisme de groupoïdes

$$\mathbb{M}_{13} \to \mathfrak{S}_{\mathscr{P}},$$

qui à un chemin associe la permutation des pièces numérotées (où le trou est numéroté par 0) donnée par la suite de mouvements du jeu basique. On a aussi un morphisme de groupes

$$\mathbb{M}_{12} \to \mathfrak{S}_{\mathscr{P}},$$

qui à un chemin fermé associe la permutation des pièces numérotées (où le trou est numéroté par 0) donnée par la suite de mouvements du jeu basique.

Définition. Le groupe du jeu basique est le groupe des permutations induites par des chemins fermés avec $p_n = p_0 = 12$. On le note G_{bas}. C'est un sous-groupe de $\mathfrak{S}_{\mathscr{P}\setminus\{12\}} \cong \mathfrak{S}_{12}$.

Proposition 12. *L'ordre du groupe G_{bas} est au moins* 95 040.

Démonstration. Si on choisit 3 chemins de la forme $[12, a, b, c, d, 12]$ au hasard, on peut obtenir un groupe engendré de cardinal 95040. Ainsi, les chemins

$$[12, 12, 9, 11, 0, 12],\ [12, 6, 11, 6, 8, 12] \text{ et } [12, 6, 3, 1, 7, 12]$$

engendrent un groupe de cardinal 95040. Donc G_{bas} a un sous-groupe de cardinal 95040. Cette vérification est effectuée dans la partie 8 du code. □

3. Dans un groupoïde, on a une loi partiellement définie : on peut calculer la concaténation $[p_0, \dots, p_n] \circ [q_0, \dots, q_n]$ si $q_n = p_0$. L'inversibilité se voit car on travaille à équivalence de chemins près.

4. Attention à bien faire la différence entre $\mathbb{M}_{12}$ et M_{12}. On verra que $\mathbb{M}_{12}$ est une extension double de M_{12}.

4.2 Jeu signé

On suppose maintenant que les pièces numérotées ont chacune deux faces non identiques. On suppose que le trou est en p. Soit q un autre point et $l = \{p, q, r, s\}$ la droite qui contient p et q. Le mouvement $[p, q]$ consiste alors à déplacer la pièce posée en q vers p et à échanger les pièces posées en r et s en les retournant.

Soit $[p, q]$ un mouvement du jeu signé. On suppose que la droite qui contient p et q est $l = \{p, q, r, s\}$. Pour $w \in X$, on définit $w' = [p, q] \cdot w$ par

$$\begin{array}{lll} w'_p = w_q, & w'_r = -w_s, & w'_t = w_t \text{ pour } t \notin l, \\ w'_q = -w_p - w_q, & w'_s = -w_r. & \end{array} \tag{2}$$

On vérifie par un calcul facile que cette action est compatible avec la composition des mouvements du jeu signé.

On définit ainsi un morphisme de groupoïdes $\mathbb{M}_{13} \to \mathrm{GL}_{13}(\mathbb{F}_3)$ qui à un chemin associe la matrice de l'action de ce chemin comme ci-dessus.

Lemme 13. *Pour tous $p, q \in \mathscr{P}$, on a $[p, q] \cdot \mathscr{C}_p = \mathscr{C}_q$.*

Démonstration. On suppose que la droite qui contient p et q est $l = \{p, q, r, s\}$. Il suffit de montrer que $[p, q] \cdot \mathscr{C}_p \subset \mathscr{C}_q$, puisque l'action décrite par l'équation 2 est inversible. Soit $c \in C_p$ et soit $d = [p, q] \cdot c$. D'après la proposition 9,

$$c_p = -\sum_{t \in \mathscr{P}} c_t = -\sum_{t \in l} c_t = -c_p - c_q - c_r - c_s$$

donc $c_p = c_q + c_r + c_s$ puisqu'on travaille sur $\mathbb{F}_3$. Alors

$$\begin{aligned} c - d &= \sum_{t \in l} (c_t - d_t) x_t \\ &= (c_p - c_q) x_p + (c_q - (-c_p - c_q)) x_q + (c_r + c_s) x_r + (c_s + c_r) x_s \\ &= (c_p - c_q)(x_p + x_q) + (c_r + c_s)(x_r + x_s) \\ &= (c_p - c_q) h_l. \end{aligned}$$

Donc $c - d \in C$ et $d \in C$. De plus,

$$\begin{aligned} \sum_{t \in \mathscr{P}} d_t = \sum_{t \in l} d_t &= d_p + d_q + d_r + d_s \\ &= c_p + c_q = -d_q. \end{aligned}$$

Donc $d \in C_q$. □

Proposition 14. *La restriction de $\mathbb{M}_{13} \to \mathrm{GL}_{13}(\mathbb{F}_3)$ à $\mathbb{M}_{12}$ donne en quotientant par $\mathbb{F}_3 x_{12}$ un morphisme $\mathbb{M}_{12} \to B_{12}$ (où x_{12} est le vecteur de la base canonique de $\mathbb{F}_3^{\mathscr{P}}$ et B_{12} est le groupe des matrices monomiales de $\mathrm{GL}_{12}(\mathbb{F}_3)$).*

Cela revient en fait à associer une matrice monomiale M à un mouvement $[p, q]$ de la façon suivante si $l = \{p, q, r, s\}$ est la droite qui contient p et q :

$$\text{pour } i, j \in \{0, \ldots, 12\}, \text{ on a } M_{i,j} = \begin{cases} 1 & \text{si}(i, j) \in \{(p, q), (q, p)\} \\ -1 & \text{si}(i, j) \in \{(r, s), (s, r)\} \\ 0 & \text{sinon,} \end{cases}$$

puis à quotienter par par $\mathbb{F}_3 x_{12}$ les matrices des chemins de $\mathbb{M}_{12}$ pour obtenir des matrices 12×12.

Exemple. Le chemin $[12, 8, 5, 12]$ induit la permutation $(2, 12)(3, 5)(4, 8)(6, 9)$ (jeu basique) et la matrice monomiale suivante dans B_{12} :

$$\begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 \\
0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{pmatrix}$$

Définition. Le groupe des matrices monomiales induites par des chemins fermés avec $p_0 = p_n = 12$ est appelé groupe du jeu signé et noté G_{sgn}.

Ainsi, un chemin $[p_0, \ldots, p_n]$ du jeu signé induit un isomorphisme entre C_{p_0} et C_{p_n}, et un chemin fermé avec $p_0 = p_n$ induit un automorphisme de C_{p_0}. Ainsi, le groupe G_{sgn} est un sous-groupe de $\text{Aut}(\mathscr{G}_{12})$.

Proposition 15. *Pour $p, q \in \mathscr{P}$, les codes $\mathscr{G}_p$ et $\mathscr{G}_q$ sont isomorphes.*

Démonstration. Cela vient du fait que $\mathscr{C}_p$ et $\mathscr{C}_q$ sont isomorphes via le chemin $[p, q]$. On vérifie l'isomorphisme entre $\mathscr{G}_p$ et $\mathscr{G}_q$ dans la partie 3 du code. □

Proposition 16. *Le groupe G_{sgn} est isomorphe à $2 \cdot \text{M}_{12}$ et le groupe G_{bas} est isomorphe à un sous-groupe de M_{12}.*

Démonstration. On a une application $\text{Aut}(\mathscr{G}_{12}) \to \text{Aut}(W_6)$ qui provient du fait qu'un automorphisme du code $\mathscr{G}_0$ préserve naturellement les supports des vecteurs de poids 6, donc préserve W_6. De plus, on vérifie par un calcul informatique (voir partie 8 du code) que le noyau de cette application est $\{\pm I_{12}\}$. Ainsi, en quotientant par $\{\pm I_{12}\}$, $\text{Aut}(\mathscr{G}_{12})/\{\pm I_{12}\}$ est isomorphe à un sous-groupe de $\text{Aut}(W_6)$.

Par ailleurs, G_{sgn} est un sous-groupe de $\text{Aut}(\mathscr{G}_{12})$. Le chemin $[12, 7, 10, 12, 1, 2, 8, 1, 4, 6, 9, 4, 12]$ retourne toutes les pièces sans les déplacer : il correspond à la matrice $-I_{12}$. On a donc $-I_{12} \in G_{\text{sgn}}$. Alors $G_{\text{sgn}}/\{\pm I_{12}\}$ est isomorphe à un sous-groupe de $\text{Aut}(\mathscr{G}_{12})/\{\pm I_{12}\}$. Or $G_{\text{sgn}}/\{\pm I_{12}\} = G_{\text{bas}}$ car quotienter par $\{\pm I_{12}\}$ revient à oublier les retournements de pièces. Donc G_{bas} est un sous-groupe de $\text{Aut}(W_6) = \text{M}_{12}$, et $G_{\text{sgn}} \cong 2 \cdot \text{M}_{12}$. □

Theorème 17. *On a les isomorphismes de groupes suivants :*

1. $G_{\text{bas}} \cong \text{M}_{12}$,
2. $G_{\text{sgn}} \cong 2 \cdot \text{M}_{12}$.

Démonstration. On a vu que G_{bas} est un sous-groupe de M_{12}, et que son cardinal $|G_{\text{bas}}|$ est au moins $95\,040 = |\text{M}_{12}|$. Le second point découle de la proposition précédente. □

4.3 Jeu dual

On étend le jeu basique en plaçant un second jeu de pièces numérotées, cette fois sur les droites de $\mathbb{P}^2(\mathbb{F}_3)$, en laissant à nouveau un « trou », codé par un pièce numérotée 0. Les mouvements sont définis de la même façon que ceux du jeu basique, avec la condition supplémentaire que le point-trou doit toujours appartenir à la droite-trou.

Supposons que le point-trou soit en p et que la droite-trou soit en l, avec $\mathscr{L}(p) = \{l, m, n, k\}$ et $l = \{p, q, r, s\}$. Le mouvement (sur les points) $[p, q]$ est défini comme dans le jeu basique, et le

mouvement (sur les droites) $[l, m]$ consiste à déplacer la pièce située en m vers l et à échanger les pièces situées en n et k.

Un chemin est de la forme

$$([p_0, \dots, p_n], [l_0, \dots, l_n]) = ([l_{n-1}, l_n], [p_{n-1}, p_n]) \circ \cdots \circ ([l_0, l_1], [p_0, p_1])$$

avec les conditions

$$\begin{array}{ll} p_i, p_{i+1} \in l_i & \text{pour tout } i, \\ l_i, l_{i+1} \in \mathscr{L}(p_{i+1}) & \text{pour tout } i. \end{array}$$

Chaque chemin induit une paire de permutations $\sigma = (\sigma_{\mathscr{P}}, \sigma_{\mathscr{L}})$ où $\sigma_{\mathscr{P}}$ agit sur les points et $\sigma_{\mathscr{L}}$ agit sur les droites.

Un chemin est dit fermé si $p_0 = p_n$ et $l_0 = l_n$.

Lemme 18. *Soit $([p_0, \dots, p_n], [l_0, \dots, l_n])$ un chemin du jeu dual. Alors le chemin $[p_0, \dots, p_n]$ est non dégénéré si et seulement si le chemin $[l_0, \dots, l_n]$ l'est.*

Démonstration. Supposons $[p_0, \dots, p_n]$ dégénéré. Il existe alors $i \leq n-2$ tel que p_i, p_{i+1} et p_{i+2} sont colinéaires. Donc p_{i+2} appartient à l'unique droite contenant p_i et p_{i+1}. Or $p_i, p_{i+1} \in l_i$ et $p_{i+1}, p_{i+2} \in l_{i+1}$ donc $l_i = l_{i+1}$. Le chemin $[l_0, \dots, l_n]$ est donc dégénéré. La réciproque s'obtient par dualité. □

Lemme 19. *Tout chemin $([p_0, \dots, p_n], [l_0, \dots, l_n])$ du jeu dual est équivalent à un chemin où les deux chemins $[p_0, \dots, p_n]$ et $[l_0, \dots, l_n]$ sont non dégénérés.*

Démonstration. Soit $([p_0, \dots, p_n], [l_0, \dots, l_n])$ un chemin du jeu dual. On sait que le chemin $[p_0, \dots, p_n]$ est équivalent à un chemin non dégénéré $[p'_0, \dots, p'_m]$. De plus, on a $p_0, p_1 \in l_0$ donc l_0 est uniquement déterminée. Par récurrence, l_i est uniquement déterminée pour tout $i < n$. Enfin, $l_n = l_0$ puisque le chemin est fermé. Donc $[l_0, \dots, l_n]$ est entièrement déterminé par $[p_0, \dots, p_n]$. De même, $[p'_0, \dots, p'_m]$ détermine un unique chemin $[l'_0, \dots, l'_m]$ sur les droites. Celui-ci est non dégénéré d'après le lemme précédent. □

Définition. Le groupe engendré par les paires de permutations induites par les chemins fermés avec $p_0 = p_n = 12$ et $l_0 = l_n = \overline{12}$ est appelé G_{dual}.

Theorème 20. *Si $\sigma = (\sigma_{\mathscr{P}}, \sigma_{\mathscr{L}})$ est une paire de permutations induite par une paire de chemins fermés, alors $\sigma_{\mathscr{L}}$ est uniquement déterminée par $\sigma_{\mathscr{P}}$. De plus, $G_{\text{dual}} \cong \mathrm{M}_{12}$.*

Démonstration. Soit $\sigma = (\sigma_{\mathscr{P}}, \sigma_{\mathscr{L}})$ induite par une suite de mouvements du jeu dual. Soit $[p_0, \dots, p_n]$ le chemin de points qui induit $\sigma_{\mathscr{P}}$, avec $p_0 = p_n$. Alors $p_0, p_1 \in l_0$ donc l_0 est uniquement déterminée. Par récurrence, l_i est uniquement déterminée pour tout $i < n$. Enfin, $l_n = l_0$ puisque le chemin est fermé. Donc $\sigma_{\mathscr{L}}$ est entièrement déterminée. De plus, on a un morphisme $G_{\text{dual}} \to G_{\text{bas}}$ donnée par $(\sigma_{\mathscr{P}}, \sigma_{\mathscr{L}}) \mapsto \sigma_{\mathscr{P}}$, et ce qui précède donne le morphisme réciproque. □

Définition. On définit l'application

$$\begin{array}{rccc} \theta & : & G_{\text{dual}} & \to & G_{\text{dual}} \\ & & (\sigma_{\mathscr{P}}, \sigma_{\mathscr{L}}) & \mapsto & (\sigma_{\mathscr{L}}, \sigma_{\mathscr{P}}) \end{array}$$

L'application θ est bien définie car si $(\sigma_{\mathscr{P}}, \sigma_{\mathscr{L}}) \in G_{\text{dual}}$ provient d'un chemin $([p_0, \dots, p_n], [l_0, \dots, l_n])$, alors la suite de mouvements $([l_0, \dots, l_n], [p_0, \dots, p_n])$ satisfait aux conditions énoncées plus haut et est donc bien un chemin du jeu dual, donc $(\sigma_{\mathscr{L}}, \sigma_{\mathscr{P}}) \in G_{\text{dual}}$.

Proposition 21. *L'application θ est un automorphisme extérieur de $G_{\text{dual}} \cong \mathrm{M}_{12}$.*

Démonstration. L'application θ est bien un morphisme de groupes. Elle est clairement surjective, donc c'est un automorphisme de G_{dual}. Il reste à montrer que θ n'est pas la conjugaison par un élément de G_{dual}.

On considère les chemins (pour les points) suivants

$$\pi_1 = [12, 3, 8, 12] \quad \pi_2 = [12, 5, 9, 12] \quad \pi_3 = [12, 7, 11, 12]$$

qui induisent les permutations respectives suivantes

$$\alpha_1 = (3, 5)(4, 9)(6, 8)(7, 11) \quad \alpha_2 = (2, 7)(3, 11)(4, 8)(6, 10) \quad \alpha_3 = (1, 11)(3, 7)(4, 6)(8, 12)$$

D'après la numérotation de $\mathscr{L}$ que l'on a choisie, le chemin de droites correspondant à α_i est α_i^{-1} pour $i \in \{1, 2, 3\}$. Donc $\theta(\alpha_i) = \alpha_i^{-1} = \alpha_i$. Ainsi, si θ est la conjugaison par un élément σ de G_{dual} alors σ commute avec chaque α_i. Un calcul par ordinateur nous donne que l'intersection des commutants des α_i contient un seul élément non trivial, qui est

$$\sigma = (1, 12)(2, 10)(3, 4)(5, 9)(6, 7)(8, 11).$$

Considérons le chemin $[12, 3, 0, 12]$. Son chemin dual associé est $[12, 4, 7, 12]$. Les permutations associées sont respectivement $\pi_{\mathscr{P}} = (1, 4)(2, 3)(6, 8)(11, 12)$ et $\pi_{\mathscr{L}} = (2, 11)(3, 9)(4, 6)(5, 8)$. Ainsi, θ échange ces deux permutations, or elles ne sont pas conjuguées par σ. Donc θ n'est pas la conjugaison par un élément de G_{dual}, donc c'est un automorphisme extérieur de G_{dual}. □

On peut aussi utiliser le jeu dual pour retrouver le groupe M_{12} à partir du groupe des automorphismes d'une matrice d'Hadamard H bien choisie. Ainsi, on peut montrer que $G_{\text{dual}} \cong \mathrm{Aut}(H)/\{\pm 1\}$. Cette réalisation de M_{12} est décrite dans [CEM05].

Références

[Ple89] Vera Pless. "Extremal codes are homogeneous". In : *IEEE Trans. Inf. Theory* 35 (1989), p. 1329-1330.

[CEM05] John H. Conway, Noam D. Elkies et Jeremy L. Martin. *The Mathieu group M_{12} and its pseudogroup extension M_{13}*. 2005. arXiv : `math/0508630 [math.GR]`. URL : `https://arxiv.org/abs/math/0508630`.

[Cam15] Peter J. Cameron. *From M_{12} to M_{24}*. 2015.

Annexe 1 : vocabulaire des codes

Définition. On appelle code linéaire ternaire de longueur n et de dimension k un sous-espace vectoriel de dimension k de $\mathbb{F}_3^n$.

Dans ce qui suit, on indexe par $I = \{1, \ldots, n\}$ la base canonique de $\mathbb{F}_3^n$, et on note v_i la coordonnée d'indice i d'un vecteur $v \in \mathbb{F}_3^n$.

Définition. Pour $v \in \mathbb{F}_3^n$, on appelle poids de v, et on note $\mathrm{wt}(v)$, le cardinal du support du vecteur v, où le support de v est $\mathrm{Supp}(v) = \{i \in I \mid v_i \neq 0\}$.

Définition. Le poids minimal d'un code $C \subset \mathbb{F}_3^n$, noté $\mathrm{wt}_{\min}(C)$, est

$$\mathrm{wt}_{\min}(C) = \min\{\mathrm{wt}(c) \mid c \in C \setminus \{0\}\}$$

Définition. On appelle matrice monomiale une matrice carrée dont chaque ligne et chaque colonne contient exactement un coefficient non nul.

Définition. On appelle isomorphisme de codes entre deux codes linéaires $C_1 \subset \mathbb{F}_3^n$ et $C_2 \subset \mathbb{F}_3^n$ une matrice monomiale A de taille $n \times n$ telle que $A \cdot C_1 = C_2$.

Définition. On appelle matrice génératrice d'un code linéaire $C \subset \mathbb{F}_3^n$ la matrice de l'application linéaire $\phi : \mathbb{F}_3^k \to \mathbb{F}_3^n$ telle que $\phi(\mathbb{F}_3^k) = C$ dans les bases canoniques.

Définition. Le code de Golay ternaire étendu est le code linéaire de longueur 12 et de dimension 6 sur $\mathbb{F}_3$ dont une matrice génératrice est

$$\left[\begin{array}{cccccc|cccccc} 1 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 1 & 2 & 1 & 2 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 2 & 2 & 2 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 2 & 2 & 2 \\ 0 & 0 & 0 & 0 & 1 & 0 & 2 & 1 & 2 & 2 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 2 & 1 & 2 & 2 & 1 \end{array}\right]$$

Annexe 2 : code

1 Définitions d'ensembles

[1]:
```
F3=GF(3)
S13=SymmetricGroup(13)
S12=SymmetricGroup(12)
M12=MathieuGroup(12)
Golay=codes.GolayCode(GF(3))

#On construit le plan projectif P^2(F_3)
P = [vector((1,0,0))] + [vector((a,1,0)) for a in F3] + [vector((a,b,1)) for b␣
 ↪in F3 for a in F3]
L = [[k for k in range(13) if P[k].dot_product(P[l])==0] for l in range(13)]

#On construit les codes C et Cpr
h=[vector([1 if i in l else 0 for i in range(13)]) for l in L]
C=span(h,F3)
Cpr=span([x for x in C if sum(x[i] for i in range(13))==0],F3)

#On définit des fonctions utiles à la construction des codes Gp
def supp(v):
    return [p for p in range(len(v)) if v[p]!=0]

def wt(v):
    return len(supp(v))

#On construit les codes Cp et Gp
Cp=[[c for c in C if c[p]==-sum([c[q] for q in list(range(13))])] for p in␣
 ↪list(range(13))]
Gp=[span([vector([c[q] for q in list(range(13)) if q !=p]) for c in Cp[p]],F3)␣
 ↪for p in list(range(13))]
```

[2]:
```
#histogramme des poids des vecteurs de C
print([len([v for v in C if wt(v)==i]) for i in range(14)])

#histogramme des poids des vecteurs de Cpr
print([len([v for v in Cpr if wt(v)==i]) for i in range(14)])
```

```
#histogramme des poids des vecteurs de Gp
print([len([v for v in Gp[0] if wt(v)==i]) for i in range(14)])
```

```
[1, 0, 0, 0, 26, 0, 156, 624, 0, 494, 780, 0, 78, 28]
[1, 0, 0, 0, 0, 0, 156, 0, 0, 494, 0, 0, 78, 0]
[1, 0, 0, 0, 0, 0, 264, 0, 0, 440, 0, 0, 24, 0]
```

2 Fonctions utiles

[3]:
```
def incidence(p,l):
    return any(p==L[l][i] for i in range(4))

def line(p1,p2):
    l=0
    while not (incidence(p1,l) and incidence(p2,l)):
        l+=1
    return l

def intersection(l1,l2):
    p=0
    while not (incidence(p,l1) and incidence(p,l2)):
        p+=1
    return p

def action(p,q):
    if p!=q:
        matrice=[[1 if i==j else 0 for j in range(13)] for i in range(13)]
        l=L[line(p,q)]
        i,j=0,0
        while not((l[i]!=q) and (l[i]!=p)):
            i+=1
        while not((l[j]!=q) and (l[j]!=p) and (j!=i)):
            j+=1
        r,s=l[i],l[j]
        matrice[r][s]=-1
        matrice[s][r]=-1
        matrice[r][r],matrice[s][s],matrice[p][p]=0,0,0
        matrice[p][q]=1
        matrice[q][p],matrice[q][q]=-1,-1
        return matrix(matrice)
    else:
        return identity_matrix(F3,13)
```

3 Isomorphismes entre les codes de Golay

[4]:
```
#On vérifie d'abord que les Gp sont isomorphes entre eux :

def iso(p,q):
    return matrix([[action(p,q)[i,j] for j in range(13) if j!=p] for i in␣
 ↪range(13) if i!=q])

print("On vérifie que les Gp sont tous isomorphes à G12 :")
print(all(all(iso(12,p)*v in Gp[p] for v in Gp[12].basis()) for p in range(12)))
print(all(all(iso(p,12)*v in Gp[12] for v in Gp[p].basis()) for p in range(12)))

print('\n')
#On veut ensuite exhiber un isomorphsime entre G12 et le code de Golay ternaire␣
 ↪étendu :
def diag(L,n=12):
    return diagonal_matrix([-1 if i in L else 1 for i in range(n)])

def perm(i,j,n=12):
    A = identity_matrix(n).change_ring(GF(3))
    A[i,i]=0; A[i,j]=1; A[j,j]=0 ; A[j,i]=1
    return A

base=Golay.basis()
base2=Gp[12].basis()
MAT = matrix(Gp[12].basis())
GOL = matrix(Golay.basis())

M=matrix(F3,[x[6:] for x in base])
N=matrix(F3,[x[6:] for x in base2])
A=perm(0,2,6)*diag([2],6)*diag([0,1,3],6)
B=perm(0,3,6)*block_matrix(F3,2,2,[matrix(F3, 1, 1, [ 1␣
 ↪]),0,0,perm(2,4,5)])*block_matrix(F3,2,2,[matrix(F3, 1, 1, [ 1␣
 ↪]),0,0,perm(0,3,5)])
B=B*block_matrix(F3,2,2,[matrix(F3, 1, 1, [ 1␣
 ↪]),0,0,perm(1,4,5)])*block_matrix(F3,2,2,[matrix(F3, 2, 2,␣
 ↪[1,0,0,1]),0,0,perm(2,3,4)])
B=block_matrix(F3,2,2,[perm(0,2,6)*diag([0,1,2,3],6),0,0,B])

print("On vérifie que MAT==A*GOL*B : {}".format(MAT==A*GOL*B))
print("On vérifie visuellement que les matrices A et B sont monomiales :")
print(A)
print('\n')
print(B)
```

```
On vérifie que les Gp sont tous isomorphes à G12 :
True
```

```
True


On vérifie que MAT==A*GOL*B : True
On vérifie visuellement que les matrices A et B sont monomiales :
[0 0 2 0 0 0]
[0 2 0 0 0 0]
[2 0 0 0 0 0]
[0 0 0 2 0 0]
[0 0 0 0 1 0]
[0 0 0 0 0 1]


[0 0 2 0 0 0|0 0 0 0 0 0]
[0 2 0 0 0 0|0 0 0 0 0 0]
[2 0 0 0 0 0|0 0 0 0 0 0]
[0 0 0 2 0 0|0 0 0 0 0 0]
[0 0 0 0 1 0|0 0 0 0 0 0]
[0 0 0 0 0 1|0 0 0 0 0 0]
[-----------+-----------]
[0 0 0 0 0 0|0 0 1 0 0 0]
[0 0 0 0 0 0|0 0 0 0 0 1]
[0 0 0 0 0 0|0 0 0 0 1 0]
[0 0 0 0 0 0|1 0 0 0 0 0]
[0 0 0 0 0 0|0 1 0 0 0 0]
[0 0 0 0 0 0|0 0 0 1 0 0]
```

4 Vérification de la proposition 9

[5]:
```
#On vérifie les poids minimaux de C et Cpr, et on vérifie que wt(v) est congru à␣
↪0 ou 1 modulo 3pour v dans C,
#et que v est dans Cpr ssi wt(v) est congru à 0 modulo 3
#On voit aussi que les vecteurs de poids 4 de C sont exactement les {h_1,-h_1}␣
↪puisqu'il n'y en a que 26
#(points 2, 3 et 7 de la proposition)

#histogramme des poids des vecteurs de C
hist = [len([v for v in C if wt(v)==i]) for i in range(14)]
print("Histogramme des poids des vecteurs de C : {}".format(hist))

#histogramme des poids des vecteurs de Cpr
hist = [len([v for v in Cpr if wt(v)==i]) for i in range(14)]
print("Histogramme des poids des vecteurs de Cpr : {}".format(hist))

#histogramme des poids des vecteurs de G_12
hist = [len([v for v in Gp[12] if wt(v)==i]) for i in range(14)]
```

```
print("Histogramme des poids des vecteurs de G_12 : {}".format(hist))

#On vérifie que la somme des c_p^2 pour p dans P est égale au carré de la somme␣
 ↪des c_p pour p dans P
#(point 1 de la proposition)
verif = all(add(c[p]^2 for p in list(range(13))) == add(c[p] for p in␣
 ↪list(range(13)))^2 for c in C)
print("Vérification du point 1 de la proposition : {}".format(verif))

#On vérifie que la somme des c_p pour p dans P est égale à la somme des c_p pour␣
 ↪p dans l
#(point 4 de la proposition)
verif = all(add(c[p] for p in list(range(13))) == add(c[p] for p in l) for l in␣
 ↪L for c in C)
print("Vérification du point 4 de la proposition : {}".format(verif))

#On verifie que tout vecteur d'une base de C est orthogonal a tout vecteur d'une␣
 ↪base de de Cpr
#(point 5 de la proposition)
verif = all(v.dot_product(w)==0 for v in Cpr.basis() for w in C.basis())
print("Vérification du point 5 de la proposition : {}".format(verif))

#On calcule les dimensions de C et Cpr
#(point 6 de la proposition)

print("Dimension de C : {}".format(C.dimension()))
print("Dimension de Cpr : {}".format(Cpr.dimension()))
```

```
Histogramme des poids des vecteurs de C : [1, 0, 0, 0, 26, 0, 156, 624, 0, 494,
780, 0, 78, 28]
Histogramme des poids des vecteurs de Cpr : [1, 0, 0, 0, 0, 0, 156, 0, 0, 494,
0, 0, 78, 0]
Histogramme des poids des vecteurs de G_12 : [1, 0, 0, 0, 0, 0, 264, 0, 0, 440,
0, 0, 24, 0]
Vérification du point 1 de la proposition : True
Vérification du point 4 de la proposition : True
Vérification du point 5 de la proposition : True
Dimension de C : 7
Dimension de Cpr : 6
```

5 Construction d'un système de steiner S(5,6,12)

[6]:
```
#On construit un système de Steiner S(5,6,12) à partir des codes de Golay Gp
S=[]
for v in Gp[0]:
    if wt(v)==6 and supp(v) not in S: S.append(supp(v))
```

```
#On vérifie que c'est un système de Steiner
def included_in(E1,E2):
    return all(x in E2 for x in E1)

def is_steiner(E,k):
    #vérifie si S est un système de Steiner S(k,m,n), où m est la taille des␣
 ↪blocs
    n=max(max(l) for l in E)+1
    return all(len([l for l in E if included_in(p,l)])==1 for p in␣
 ↪Combinations(list(range(n)),5).list())

print("Vérification du fait que S est un S(5,6,12) : {}".format(is_steiner(S,5)))
```

```
Vérification du fait que S est un S(5,6,12) : True
```

6 Matrices d'Hadamard

[23]:
```
#On construit les matrices d'Hadamard H_p:
H=[matrix(QQ,12,12,[[1 if x==1 else -1 for x in v] for v in Gp[p] if wt(v)==12␣
 ↪and v[0]==1]) for p in range(13)]
print("La matrice H_12 :")
print(H[12])

#On vérifie que ce sont des matrices d'Hadamard :
verif = all(H[p]*H[p].transpose()==12*identity_matrix(QQ,12) for p in range(13))
verif2 = all(H[p][i,j]==1 or H[p][i,j]==-1 for i in range(12) for j in range(12)␣
 ↪for p in range(13))
print("On vérifie que ce sont des matrices d'Hadamard : {}".format(verif and␣
 ↪verif2))
```

```
La matrice H_12 :
[ 1  1 -1  1  1  1 -1 -1  1  1  1 -1]
[ 1  1  1 -1  1  1 -1  1 -1  1 -1  1]
[ 1 -1 -1 -1  1  1  1 -1 -1 -1 -1 -1]
[ 1 -1  1  1 -1  1  1 -1  1  1 -1  1]
[ 1 -1 -1  1 -1  1 -1  1 -1 -1  1  1]
[ 1  1  1 -1 -1  1  1  1  1 -1  1 -1]
[ 1 -1  1  1  1 -1  1  1 -1  1  1 -1]
[ 1  1 -1  1  1 -1  1  1  1 -1 -1  1]
[ 1 -1  1 -1  1 -1 -1 -1  1 -1  1  1]
[ 1  1  1  1 -1 -1 -1 -1 -1 -1 -1 -1]
[ 1  1 -1 -1 -1 -1  1 -1 -1  1  1  1]
[ 1 -1 -1 -1 -1 -1 -1  1  1  1 -1 -1]
On vérifie que ce sont des matrices d'Hadamard : True
```

7 Construction des automorphismes du S(5,6,12)

[8]:
```
#fonctions utiles pour coder les automorphismes

def bloc(p):
    #renvoie l'unique bloc qui contient une pentade p
    return [B for B in S if all(x in B for x in p)][0]

def sixieme(p):
    #renvoie le sixième élément de l'unique bloc qui contient une pentade p
    B=bloc(p)
    return [x for x in B if not x in p][0]

def paires(b0,q):
    #renvoie TP(b0,q)
    paires=[]
    b1=[x for x in range(12) if x not in b0]
    for p in b1:
        p2=sixieme(q+[p])
        if not sorted([p,p2]) in paires:
            paires.append(sorted([p,p2]))
    return paires

def quadruplets(b):
    #renvoie les quadruplets inclus dans le bloc b
    return [[x for x in partie] for partie in Combinations(b,4).list()]

def memespaires(q,qpr,b0):
    #vérifie si les quadruplets q et qpr ont la même image par la fonction TP
    return all(paire in paires(b0,qpr) for paire in paires(b0,q))

def paire_commune(paires1,paires2):
    #renvoie la paire commune à deux ensembles de paires, si cette paire commune
 ↪est unique
    comm=[paire for paire in paires1 if paire in paires2]
    if len(comm)==1:
        return comm[0]

def image_paire(b,imageb,paire):
    #calcule l'image de paire par l'automorphsime qui envoie le bloc b sur imageb
    #b et imageb ordonnés de sorte à ce que sigma[b[i]]==imageb[i]
    quad=quadruplets(b)
    sigma=[-1 for i in range(12)]
    for i in range(len(b)):
        sigma[b[i]]=imageb[i]
    #sigma est la liste incomplète des images par sigma. -1 désigne une valeur
 ↪encore inconnue (et inutile ici)
```

```
    j=0
    while not paire in paires(b,quad[j]):#on cherche un quadruplet q dans b0 tel␣
 ↪que paire soit l'une des paires associées
        j+=1
    q=quad[j]
    j=0
    while q==quad[j] or not paire in paires(b,quad[j]):#on cherche un deuxième␣
 ↪tel quadruplet qpr
        j+=1
    qpr=quad[j]
    #ainsi, paire est l'unique paire commune à paires(b,q) et paires(b,qpr)
    Q=[sigma[x] for x in q]
    Qpr=[sigma[x] for x in qpr]
    return paire_commune(paires(imageb,Q),paires(imageb,Qpr))
```

[9]:
```
#Calcul d'un automorphisme du S(5,6,12)

def automorphisme(p0,imagep0):
    b0=bloc(p0)
    x5=sixieme(p0)
    b1=[x for x in range(12) if not x in b0]
    sigma=[-1 for i in range(12)]
    B0=bloc(imagep0)
    B1=[x for x in range(12) if not x in B0]
    y5=sixieme(imagep0)
    for i in range(5):
        sigma[p0[i]]=imagep0[i]
    sigma[x5]=y5
    for x in b1:
        if sigma[x]==-1:
            j=0
            while b1[j]==x:
                j+=1
            y=b1[j]
            j=0
            while b1[j]==x or b1[j]==y:
                j+=1
            z=b1[j]
            paire1=image_paire(b0,[sigma[t] for t in b0],sorted([x,y]))
            paire2=image_paire(b0,[sigma[t] for t in b0],sorted([x,z]))
            sigma[x]=[t for t in paire1 if t in paire2][0]
            sigma[y]=[t for t in paire1 if t!=sigma[x]][0]
            sigma[z]=[t for t in paire2 if t!=sigma[x]][0]
    sigma=Permutation(map(lambda u : u+1, sigma))
    return sigma

#Exemple d'automorphisme : on veut envoyer [0,1,2,3,4] sur [4,7,2,10,9]
```

```
print(automorphisme([0,1,2,3,4],[4,7,2,10,9]))
```

```
[5, 8, 3, 11, 10, 9, 7, 2, 4, 6, 12, 1]
```

[13]:
```
#Calcul du groupe d'automorphismes du système de Steiner S :
#Attention, le temps d'exécution de cette cellule est assez long
t = cputime()
p0=[0,1,2,3,4]
M12exp=[]
for y in Arrangements(list(range(12)),5):
    M12exp.append(automorphisme(p0,y))

print("Temps d'exécution : {}".format(cputime(t)))
```

```
Temps d'exécution : 2640.2093100000006
```

[12]:
```
#On vérifie le nombre d'automorphismes obtenus :
len(M12exp)
```

[12]:
```
95040
```

8 Jeu de taquin sur $(P)^2((F)_3)$

[25]:
```
def taquin(p,q,sgn=1):
    l=L[line(p,q)]
    i,j=0,0
    while not((l[i]!=q) and (l[i]!=p)):
        i+=1
    while not((l[j]!=q) and (l[j]!=p) and (j!=i)):
        j+=1
    x,y=l[i],l[j]
    if sgn==1:
        if p==q:
            return Permutation(list(range(1,14)))
        else:
            sigma=list(range(13))
            sigma[p]=q
            sigma[q]=p
            sigma[x]=y
            sigma[y]=x
            return Permutation([k+1 for k in sigma])
    else :
        if p==q:
            return matrix([[1 if i==j else 0 for i in range(13)] for j in␣
 ↪range(13)])
        else:
```

```
            m=matrix(F3,[[1 if i==j else 0 for i in range(13)] for j in␣
 ↪range(13)])
            m[p,p],m[p,q]=0,1
            m[q,q],m[q,p]=0,1
            m[x,x],m[x,y]=0,-1
            m[y,y],m[y,x]=0,-1
            return m

def jeudual(l,m):
    p=intersection(l,m)
    lines=[n for n in range(13) if incidence(p,n)]
    i,j=0,0
    while not((lines[i]!=l) and (lines[i]!=m)):
        i+=1
    while not((lines[j]!=l) and (lines[j]!=m) and (j!=i)):
        j+=1
    n,k=lines[i],lines[j]
    if l==m:
        return Permutation(list(range(1,14)))
    else:
        sigma=list(range(13))
        sigma[l]=m
        sigma[m]=l
        sigma[n]=k
        sigma[k]=n
        return Permutation([k+1 for k in sigma])

def chemin(chemin,sgn=1):
    if sgn==1:
        r=Permutation([k+1 for k in range(13)])
    else :
        r=matrix(F3,[[1 if i==j else 0 for i in range(13)] for j in range(13)])
    for k in range(len(chemin)-1):
        sigma=taquin(chemin[k],chemin[k+1],sgn)
        r=r*sigma
    return r

def chemindual(chemin, chemindual):
    r1=Permutation([k+1 for k in range(13)])
    r2=Permutation([k+1 for k in range(13)])
    for k in range(len(chemin)-1):
        sigma=taquin(chemin[k],chemin[k+1])
        r1=sigma*r1
    for k in range(len(chemindual)-1):
        sigma=jeudual(chemindual[k],chemindual[k+1])
        r2=sigma*r2
    return r1,r2
```

[24]:
```
#Vérification que les mouvements [12,p,q,12] du jeu signé induisent bien des␣
↪matrices monomiales :
def monomiale(mat):
    verif = all(len([mat[i,j] for j in range(mat.ncols()) if mat[i,j]!=0])==1␣
↪for i in range(mat.nrows()))
    verif2 = all(len([mat[i,j] for i in range(mat.nrows()) if mat[i,j]!=0])==1␣
↪for j in range(mat.ncols()))
    return verif and verif2

print(all(monomiale(matrix([[action(q,12)*action(p,q)*action(12,p)[i,j] for j in␣
↪range(12)] for i in range(12)])) for q in range(12) for p in range(12)))
```

```
True
```

[26]:
```
#Exemple de mouvement du jeu signé et comparaison avec le jeu basique :
print("Le chemin [12,8,5,12] induit la permutation {}".
↪format(S13(chemin([12,8,5,12]))))
print("Le chemin [12,8,5,12] induit la matrice monomiale \n{}".
↪format(chemin([12,8,5,12],-1)))
```

```
Le chemin [12,8,5,12] induit la permutation (2,12)(3,5)(4,8)(6,9)
Le chemin [12,8,5,12] induit la matrice monomiale
[1 0 0 0 0 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0 0 0 0 2]
[0 0 0 0 2 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 2 0 0 0 0]
[0 0 2 0 0 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0 1 0 0 0]
[0 0 0 0 0 0 1 0 0 0 0 0]
[0 0 0 2 0 0 0 0 0 0 0 0]
[0 0 0 0 0 1 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0 0 1 0 0]
[0 0 0 0 0 0 0 0 0 0 1 0]
[0 2 0 0 0 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0 0 0 0 1]
```

[17]:
```
#On montre que l'ordre de G_bas est au moins 95040 :

test=[[12]+[randint(0,12) for j in range(4)]+[12] for i in range(3)]
print("On choisit 3 chemins au hasard : {}".format(test))
print("On associe une permutation à chacun de ces chemins :")
print([S13(chemin(c)) for c in test])
Group=S13.subgroup([chemin(c) for c in test])
print("L'ordre du groupe engendré est {}.".format(Group.order()))
```

```
On choisit 3 chemins au hasard : [[12, 0, 5, 8, 3, 12], [12, 1, 7, 8, 12, 12],
[12, 1, 7, 3, 4, 12]]
On associe une permutation à chacun de ces chemins :
```

```
[(1,4,9,8,6)(2,12,7,5,11), (1,10,7)(2,9,8)(3,5,11),
(2,6,5,11,4,8)(3,9)(7,12,10)]
L'ordre du groupe engendré est 95040.
```

[18]:
```
def mat(g):
    return matrix([[1 if g(i)==j else 0 for j in range(1,13)] for i in␣
 ↪range(1,13)])
base = Golay.basis()

#Automporhismes du code de Golay :
GolayAut=[]
for g in M12.gens():
    M=mat(g)
    indices=[(i,j) for i in range(12) for j in range(12) if M[i][j]==1]
    for P in Combinations(indices):
        N=matrix([[-1 if (i,j) in P else M[i][j] for j in range(12)] for i in␣
 ↪range(12)])
        if all(N*w in Golay for w in base):
            GolayAut.append(N)
groupe=MatrixGroup(GolayAut)
print("Il y a {} automorphismes du code de Golay.".format(groupe.order()))


#Vérifions qu'il n'y a pas plus d'automorphismes du code de Golay, i.e. que␣
 ↪l'oubli de signe est 2:1
#On regarde le noyau de l'application de GolayAut vers M_12 qui oublie les signes
ker=[]
for g in groupe :
    M=matrix(g)
    N=matrix([[1 if M[i][j]!=0 else 0 for j in range(12)] for i in range(12)])
    I=identity_matrix(F3,12)
    if N==I :
        ker.append(M)

print("On calcule le noyau de l'application de Aut(Golay) vers M_12 :")
for m in ker:
    print(m)
```

```
Il y a 190080 automorphismes du code de Golay.
On calcule le noyau de l'application de Aut(Golay) vers M_12 :
[1 0 0 0 0 0 0 0 0 0 0 0]
[0 1 0 0 0 0 0 0 0 0 0 0]
[0 0 1 0 0 0 0 0 0 0 0 0]
[0 0 0 1 0 0 0 0 0 0 0 0]
[0 0 0 0 1 0 0 0 0 0 0 0]
[0 0 0 0 0 1 0 0 0 0 0 0]
[0 0 0 0 0 0 1 0 0 0 0 0]
[0 0 0 0 0 0 0 1 0 0 0 0]
```

```
[0 0 0 0 0 0 0 0 1 0 0 0]
[0 0 0 0 0 0 0 0 0 1 0 0]
[0 0 0 0 0 0 0 0 0 0 1 0]
[0 0 0 0 0 0 0 0 0 0 0 1]
[-1  0  0  0  0  0  0  0  0  0  0  0]
[ 0 -1  0  0  0  0  0  0  0  0  0  0]
[ 0  0 -1  0  0  0  0  0  0  0  0  0]
[ 0  0  0 -1  0  0  0  0  0  0  0  0]
[ 0  0  0  0 -1  0  0  0  0  0  0  0]
[ 0  0  0  0  0 -1  0  0  0  0  0  0]
[ 0  0  0  0  0  0 -1  0  0  0  0  0]
[ 0  0  0  0  0  0  0 -1  0  0  0  0]
[ 0  0  0  0  0  0  0  0 -1  0  0  0]
[ 0  0  0  0  0  0  0  0  0 -1  0  0]
[ 0  0  0  0  0  0  0  0  0  0 -1  0]
[ 0  0  0  0  0  0  0  0  0  0  0 -1]
```

[19]:
```
#proposition 21 : automorphisme extérieur de M_12
def dualpath(chemin):
    return [line(chemin[i], chemin[i+1]) for i in␣
 ↪range(len(chemin)-1)]+[line(chemin[0],chemin[1])]

def permutation(chemin,p):
    chemin=[x for x in chemin if x!=p+1]
    return S12([x if x<p+1 else x-1 for x in chemin])

p=[k for k in range(13) if incidence(k,k)][-1]
pi=[[p,q,[r for r in range(13) if line(q,r)==r and r !=p and r !=q][0],p] for q␣
 ↪in L[p] if q!=p]
print("On considère les trois chemins suivants : {}".format(pi))
alpha1=chemin(pi[0])
alpha2=chemin(pi[1])
alpha3=chemin(pi[2])
print("On associe une permutation à chacun de ces chemins :")
print(permutation(alpha1,p),permutation(alpha2,p),permutation(alpha3,p))
print("\n")
intersectionCommutants=S12.subgroup([s for s in S12.
 ↪centralizer(permutation(alpha1,p)) if s in S12.
 ↪centralizer(permutation(alpha2,p)) if s in S12.
 ↪centralizer(permutation(alpha3,p))])
print("On calcule l'intersection des commutants de ces permutations : ")
print(intersectionCommutants)
q=[q for q in L[p] if q!=p][0]
r=[r for r in range(13) if r!=p and r!=q and not incidence(r,p)][0]
path=[p,q,r,p]
pi=permutation(chemindual(path,dualpath(path))[0],p)
pidual=permutation(chemindual(path,dualpath(path))[1],p)
```

```
sigma=intersectionCommutants.gens()[1]
print("Le seul élément non trivial du sous-groupe ci-dessus est {}".
 ↪format(sigma))
print("\n")
print("On considère le chemin {}. Le chemin sur les droites associé est {}.".
 ↪format(path,dualpath(path)))
print("Les permutations associées à ces chemins sont respectivement {} et {}.".
 ↪format(pi,pidual))
print("On vérifie si ces permutations sont conjuguées sous l'action de sigma :␣
 ↪{}".format(sigma*pi*sigma==pidual))
```

```
On considère les trois chemins suivants : [[12, 3, 8, 12], [12, 5, 9, 12], [12,
7, 11, 12]]
On associe une permutation à chacun de ces chemins :
(3,5)(4,9)(6,8)(7,11) (2,7)(3,11)(4,8)(6,10) (1,11)(3,7)(4,6)(8,12)


On calcule l'intersection des commutants de ces permutations :
Subgroup generated by [(), (1,12)(2,10)(3,4)(5,9)(6,7)(8,11)] of (Symmetric
group of order 12! as a permutation group)
Le seul élément non trivial du sous-groupe ci-dessus est
(1,12)(2,10)(3,4)(5,9)(6,7)(8,11)


On considère le chemin [12, 3, 0, 12]. Le chemin sur les droites associé est
[12, 4, 7, 12].
Les permutations associées à ces chemins sont respectivement
(1,4)(2,3)(6,8)(11,12) et (2,11)(3,9)(4,6)(5,8).
On vérifie si ces permutations sont conjuguées sous l'action de sigma : False
```

[20]:
```
#On cherche un chemin qui induit la matrice -I_12 :
chemin([12,7,10,12,1,2,8,1,4,6,9,4,12],-1)
```

[20]:
```
[-1  0  0  0  0  0  0  0  0  0  0  0  0]
[ 0 -1  0  0  0  0  0  0  0  0  0  0  0]
[ 0  0 -1  0  0  0  0  0  0  0  0  0  0]
[ 0  0  0 -1  0  0  0  0  0  0  0  0  0]
[ 0  0  0  0 -1  0  0  0  0  0  0  0  0]
[ 0  0  0  0  0 -1  0  0  0  0  0  0  0]
[ 0  0  0  0  0  0 -1  0  0  0  0  0  0]
[ 0  0  0  0  0  0  0 -1  0  0  0  0  0]
[ 0  0  0  0  0  0  0  0 -1  0  0  0  0]
[ 0  0  0  0  0  0  0  0  0 -1  0  0  0]
[ 0  0  0  0  0  0  0  0  0  0 -1  0  0]
[ 0  0  0  0  0  0  0  0  0  0  0 -1  0]
[ 0  0  0  0  0  0  0  0  0  0  0  0  1]
```