Programme Caml

La fonction du programme est le calcul du groupe des isométries linéaires (voir partie III) d'un tore algébrique associé à une norme polygonale.

Pour toute partie convexe bornée C de $\mathbb{R}^2$, si C est un voisinage de l'origine et si C est invariant par symétrie centrale, alors il existe une unique norme N_C sur $\mathbb{R}^2$ dont $int(C)$ et $adh(C)$ soient respectivement la boule ouverte / la boule fermée. (c'est l'application $u \mapsto inf(\{t \in \mathbb{R}_+ \mid u \in t.C\})$).

(le résultat se généralise à tout $\mathbb{R}$-espace vectoriel normé, et à tout $\mathbb{C}$-vectoriel normé si l'on remplace la condition d'invariance par symétrie centrale par l'invariance sous l'action du groupe unité)

Dans le cas où C est un polygone, c'est-à-dire s'il existe de plus un entier naturel n et des points $A_1, ..., A_n$ donc C soit l'enveloppe convexe, on dit que N_C est une norme polygonale et c'est donc les tores associés qui sont les objets du calcul.

Sachant que les isométries linéaires se lisent dans le plan comme des automorphismes d'espace vectoriel de $\mathbb{R}^2$ codés par des matrices à coefficients entiers, et qu'il en existe un nombre fini à norme fixée (**lemme IV.1**), une énumération via l'informatique est possible.

On peut montrer en utilisant les propriétés de conservations des barycentres d'une application affine et le caractère bijectif d'un automorphisme f que l'image d'un polygone convexe de points extrémaux donnés dans le sens direct $A_1, ..., A_n$ voisinage de l'origine et invariant par symétrie centrale en est un autre et ses points extrémaux donnés dans le sens direct sont $f(A_1), ..., f(A_n)$ ou $f(A_n), ..., f(A_1)$ selon que $\det(f)$ soit positif ou non. (et ici pour les isométries on aura $\det(f) \in \{-1; 1\}$, voir partie III)

Du reste, une application linéaire est une isométrie si et seulement si elle stabilise la boule unité. Ainsi une application bijective $f : \mathbb{R}^2 \to \mathbb{R}^2$ est une isométrie linéaire de $(\mathbb{R}^2, N_C)$ si et seulement si $f(\{A_1, ..., A_n\}) \subset \{A_1, ..., A_n\}$, test qui se code facilement avec des listes.

Le programme teste les applications linéaires susceptibles de satisfaire cette propriété.

La convexité de C et le fait que c'est un voisinage de l'origine permet de montrer que deux points extrémaux de C distincts forment une base de $\mathbb{R}^2$, et leurs images par f suffisent à déterminer f.

Si on choisit les deux points adjacents (les notions d'extrémalité et d'adjacence peuvent ici se traiter en termes de barycentres à l'aide de quantificateurs, et sont donc conservées par transformations affines bijectives), alors leurs images seront deux points adjacents.

Ainsi, $(f(A_1), f(A_2))$ sera parmi $\{(A_1, A_2); (A_2, A_1); ...; (A_{n-1}, A_n)(A_n, A_{n-1}); (A_n, A_1); (A_1, A_n)\}$, ce qui ne fait que $2n$ isométries linéaires potentielles.

On génère en résolvant des systèmes 2 x 2 les $2n$ applications satisfaisant la condition d'appartenance ci-dessus et on effectue à chaque fois le test $f(\{A_1, ..., A_n\}) \subset \{A_1, ..., A_n\}$. Les applications pour lesquelles le test est positif sont exactement les isométries linéaires de $(\mathbb{R}^2, N_C)$, c'est-à-dire les avatars des isométries linéaires de $\mathbb{T}_{N_C}$.

Les applications linéaires sont codées par des quadruplets d'entiers, par des fonctions int * int → int * int, et dans le résultat par des listes de quatre entiers.

Les normes polygonales sont codées par des listes de couples d'entiers représentant les points extrémaux de la boule unité, et donnés dans l'ordre. Elles sont créées à la main en premier lieu, et générées en construisant leur image par des automorphismes linéaires.

Les groupes d'isométrie sont des listes d'applications linéaires ((int list) list).

- La fonction morphisme : int → int → int → int → (int * int) → (int * int) génère une application linéaire à partir de la donnée de quatre entiers.
- La fonction stable : int → int → int → int → (int * int) list → bool teste si une application linéaire est une isométrie pour la norme polygonale considérée.
- La fonction syst : int → int → int → int → int → int → int → int → int list détermine l'application linéaire envoyant les deux points extrémaux adjacents témoins sur deux autres points extrémaux adjacents donnés en argument.
- La fonction isometries : (int * int) list → (int list) list calcule le groupe des isométries linéaires de la norme polygonale considérée.
- La fonction imagepoly : (int * int) → (int * int) → (int * int) list → (int * int) list calcule l'image d'un ensemble de points extrémaux par une application linéaire.
- La fonction recherche : int → (int * int) list → (int * int * int * int) list détermine les applications linéaires aux coefficients majorés par une constance donnée en argument dont les images de l'ensemble des points extrémaux induisent une norme polygonale dont le groupe des isométries linéaires est intéressant, au sens où il n'est pas inclus dans le groupe des isométries linéaires du plan euclidien classique.

– La fonction interessant : int → (int * int) list → (int list) list renvoie le plus gros groupe d'isométries intéressant obtenu pour les images d'une liste donnée et toujours pour les coefficients des matrices utilisées limités par un argument de la fonction.

Le programme donne, pour des petites valeurs du nombre de points extrémaux et de la constance de majoration des coefficients des matrices générant d'autres listes de points extrémaux, des résultats qui vont dans le sens d'une certaine variété des groupes des isométries linéaires des tores algébriques. Je pense que la caractérisation donnée dans la partie III est donc déjà assez proche de l'optimalité, et je doute qu'on puisse majorer le cardinal de ces groupes d'isométries. Il faudrait s'y intéresser de plus près.