README

Thomas Fernique and Carole Porrier

August 5, 2022

The present repo accompanies the conference paper "A General Approach to Ammann Bars for Aperiodic Tilings" (LATIN 2022). We provide here more detailed explanations for readers who find that too many calculations are swept under the rug. Some of these calculations are difficult and rely on functions that we wrote using SageMath (Python with classical mathematical functions that are very useful here). The purpose of this Readme is to explain how these functions work, while the code is in the sage file.

1 Computing cut and project tilings

We use the duality between *multigrids* [2] and cut and project tilings. A *grid* is a set of regularly spaced parallel hyperplanes (lines in the case of an $n \to 2$ tiling). A multigrid is then a $d \times n$ matrix G whose rows define the direction and spacing of each grid (normal vector to each hyperplane, with the norm giving the spacing), and a *shift vector* S which specifies how each grid is translated away from the origin. The shift must be chosen such that no more than d hyperplanes intersect in a point (this is generic).

The function `generators_to_grid(E)` converts a slope E, given by d vectors of $\mathbb{R}^n$ which generate it, into a multigrid G.

Then, to each intersection of d hyperplanes corresponds a tile: it is generated by the directions of the hyperplanes, and the i-th coordinate of its position is the number of hyperplanes in the i-th direction between the origin and the considered intersection.

The function `dual(G,S,k)` computes the *dual* of the multigrid G with shift vector S and $2k+1$ grids in each direction (since we can only compute a subset of the infinite tiling). This is a set of tiles represented each by a pair (t, pos) where t is the d-tuple of indices of the vectors of the standard basis which define the prototile and pos is the integer translation applied on it (in $\mathbb{Z}^n$).

2 Finding the integer entries of a subperiod

Consider a slope E generated by the vectors $u_1, ..., u_d \in \mathbb{R}^n$. We assume that the entries of the u_i's are in $\mathbb{Q}(a)$ for some algebraic number a, because it has been proven in [1] to be a necessary condition to have local rules. We want to find a subperiod p of E, that is a vector of E with $d+1$ integer coordinates – the *type* of p gives us the indices of its integer coordinates $a_{i_1}, ..., a_{i_{d+1}}$ $(1 \leq i_1 < i_2 < ... < i_{d+1} \leq n)$. For this, we consider the $n \times (d+1)$ matrix N whose columns are the u_j's and the subperiod p to be determined:

$$N = \left(\begin{array}{c|c|c|c|c} & & & & \cdot \\ \cdot & \cdot & \cdots & \cdot & a_{i_1} \\ \cdot & \cdot & \cdots & \cdot & a_{i_2} \\ & & & & \vdots \\ u_1 & u_2 & \cdots & u_d & p \\ \cdot & \cdot & \cdots & \cdot & a_{i_k} \\ & & & & \vdots \\ \cdot & \cdot & \cdots & \cdot & a_{i_{d+1}} \\ & & & & \cdot \end{array}\right)$$

Since $p \in E$, this matrix has rank d and all the $(d+1)$-minors are thus zero. In particular, the nullity of the minor obtained by selecting the lines $i_1, \ldots, i_d$ yields a linear equation on the a_{i_k}. Indeed, developing the minor along the last column yields

$$(a_{i_1}, ..., a_{i_{d+1}}) \times \begin{pmatrix} (-1)^1 G_{i_2 \ldots i_{d+1}} \\ \vdots \\ (-1)^k G_{i_1 \ldots \widehat{i_k} \ldots i_{d+1}} \\ \vdots \\ (-1)^{d+1} G_{i_1 \ldots i_d} \end{pmatrix} = 0,$$

where $G_{i_1 \ldots \widehat{i_k} \ldots i_{d+1}}$ is the determinant of the $d \to d$ matrix obtained by taking the coordinates $i_1, \ldots i_{d+1}$ except i_k of the u_i's (it is also known as *Grassmann coordinates* of the plane). Since the a_i's must be integer, this can equivalently be rewritten as follows. Replace each coefficient $(-1)^k G_{i_1 \ldots \widehat{i_k} \ldots i_{d+1}}$ by a line of length $\deg(a)$, the algebraic degree of a, whose i-th entry is the coefficient of a^i in $(-1)^k G_{i_1 \ldots \widehat{i_k} \ldots i_{d+1}}$. This yields a $n \times \deg(a)$ integer matrix. Then, the a_i's are obtained by computing the left kernel of this matrix.

The above algorithm is implemented in the function `subperiods(E)`, which outputs a list of subperiods represented each by its type and its $d+1$ integer entries.

Let us illustrate this with the Cyrenaic tiling. The slope E is generated by the lines of the matrix

$$\begin{pmatrix} a & 0 & 1 & 1 \\ 1 & a-1 & -1 & 1 \end{pmatrix},$$

where $a = \sqrt{3}$. Let us search a subperiod whose first three entries are integer.

The matrix N is

$$N = \begin{pmatrix} a & 1 & a_1 \\ 0 & a-1 & a_2 \\ 1 & -1 & a_3 \\ 1 & 1 & * \end{pmatrix},$$

where $*$ denotes the non-integer entry we are here not interested in. Consider the 3-minor obtained with the three first lines and develop it along the last column. We get:

$$(a_1, a_2, a_3) \times \begin{pmatrix} -a+1 \\ a+1 \\ -a+3 \end{pmatrix} = 0.$$

Since $a = \sqrt{3}$, $\deg(a) = 2$ and this is equivalent to

$$(a_1, a_2, a_3) \times \begin{pmatrix} -1 & 1 \\ 1 & 1 \\ -1 & 3 \end{pmatrix} = 0.$$

This 3×2 integer matrix turns out to have rank 2. Its left kernel has dimension 1 and the a_i's are given by a prime integer vector in this kernel:

$$(a_1, a_2, a_3) = (2, 1, -1).$$

This yields the subperiod $p_3 = (2, 1, -1, *)$, as claimed in Subsection 3.2 of the article.

3 Determining whether a slope is characterized by its subperiods

To determine whether a slope E is characterized by its subperiods, we first compute the subperiods of E as explained in the previous section. Then, we form the matrix whose columns are these subperiods, with variables x_i's for the non-integer coordinates (since only the integer coordinates are known). Since these subperiods must belong to the d-dimensional plane E, any $(d+1)$-minor of this matrix must be zero. This yields an equation. By considering all the minors, we get a system of polynomial equation. The slope is characterized by its subperiods if and only if this system has dimension zero.

This is implemented in the function `is_determined_by_subperiods(E)`.

4 Finding the non-integer entries of a subperiod

To compute the subperiods of a slope E seen as vectors in E, we first compute the integer entries of the subperiods as explained above. We then consider the unknown non-integer entries has variables x_i's. The matrix formed by the subperiods has rank d since all the subperiods must be in the d-dim. plane E.

Its $(d+1)$-minors must thus all be zero. This yields a polynomial system of equations in the x_i's. We take the solution which is in E: this yields all the coordinates of the subperiods.

This is implemented in the function `lifted_subperiods(E)`.

5 Finding a good projection

The method to find a good projection is described in Subsection 3.2. It is implemented by the function `good_projection(E)`. We also implemented the function `valid_projection(A,E)`, which checks whether the projection A is valid for the slope E.

6 Computing the r-atlas

Computing the r-atlas of a cut and project tiling mainly relies on the notions of *window* and *region*. We here briefly recall these notions; the interested reader can find more details in [1].

Consider a strongly planar $n \to d$ tiling with a given slope E. Consider the simplest pattern: a single edge directed by $\pi(e_i)$. To decide whether this pattern appears somewhere in the tiling, we have to decide whether there exists a vertex x of the tiling such that $x + \pi(e_i)$ in also a vertex of the tiling. By definition, a vertex x belongs to the tiling if and only if its lift $\widehat{x}$ belongs to the tube $E + [0,1]^n$. The idea is to look in the space orthogonal to E, denoted by E' (sometimes called *internal space*, while E is called *real space*). Denote by π' the orthogonal projection onto E'. Now, a vertex x belongs to the tiling iff $\pi'(\widehat{x})$ belongs to the polytope $W := \pi'([0,1]^n)$. This polytope is called the *window* of the tiling. Similarly, $x + \pi(e_i)$ belongs to the tiling iff $\pi'(\widehat{x} + e_i)$ belongs to W, that is, iff $\pi'(\widehat{x})$ belongs to $W - \pi'(e_i)$. Hence, there exists two vertices x and $x + \pi(e_i)$ of the tiling if and only if the following polytope is not empty:

$$R(e_i) := W \cap (W - \pi' e_i).$$

The polytope $R(e_i)$ is called the *region* of the pattern formed by a single edge directed by $\pi(e_i)$. This can be extended to any pattern P: such a pattern appears somewhere in the tiling iff its region, defined as follows, is not empty:

$$R(P) := \bigcap_{x \in P} W - \pi'(\widehat{x}),$$

where the intersection is taken over the vertices of P. This is easily implemented in the function `region(W,ip,P)`, which takes as parameter the window, the internal projection and the pattern.

Let us now explain how to use regions to compute the r-atlas. We shall maintain a list of the already computed r-maps, together with their regions in the window W. We start with an empty list and fill it progressively as follows.

While the already computed regions do not cover the whole W, we first pick at random a point z in W which is not in one of the already computed region. We then associate with z the set of points $u \in \mathbb{Z}^n$ with norm at most r such that $z + \pi' u \in W$. This set is an r-map, which is new because its region does not overlap the already computed regions. We compute the region of this r-map and we add both the map and the region in our list.

There is an additional tricky minor detail. Before computing the region of an r-map, we must "close" this r-map, that is, add the tiles that are forced by the r-map: whenever two consecutive segments on the boundary of the pattern form a notch where a single tile can be added, then we must add this tile (because we know that there is no other segments dividing this notch - the information that there is nothing is indeed an information on its own).

All this is implemented in the function `atlas(E,r)`, which use two auxiliary functions.

7 Computing the decorated tiles

To compute the decorated tiles, we first compute a sufficiently large atlas (in the case of the Cyrenaic tilings the 6-atlas is sufficient). Then, for each pattern in this atlas, we compute the lines directed by the subperiods which go through each vertex of the pattern and intersect them with the tile at the origin of the pattern: this yields one decorated tile. This is implemented in the function `decorated_tile`. We proceed similarly for each pattern of the atlas (function `decorated_tiles`) to get the whole decorated tileset.

References

[1] Nicolas Bédaride and Thomas Fernique. Canonical projection tilings defined by patterns. *Geometriae Dedicata*, 208(1):157–175, feb 2020.

[2] N. G. de Bruijn. Algebraic theory of Penrose's non-periodic tilings of the plane. *Mathematics Proceedings*, A84:39–66, 1981.