An envelope operator for full convexity to define polyhedral models in digital spaces

Fabien Feschet[1] and Jacques-Olivier Lachaud[2*†]

[1]Université Clermont Auvergne, CNRS, ENSMSE, LIMOS, Clermont-Ferrand, F-63000, France. 0000-0001-5178-0842.

[2*]Université Savoie Mont Blanc, CNRS, LAMA, Chambéry, F-73000, France. 0000-0003-4236-2133.

*Corresponding author(s). E-mail(s): jacques-olivier.lachaud@univ-smb.fr;
Contributing authors: fabien.feschet@u-auvergne.fr;
†These authors contributed equally to this work.

Abstract

In a recent work, *full convexity* has been proposed as an alternative definition of digital convexity. It solves many problems related to its usual definitions, for instance: fully convex sets are digitally convex in the usual sense, but are also connected and simply connected. However, full convexity is not a monotone property, hence intersections of fully convex sets may be neither fully convex nor connected. This defect might forbid digital polyhedral models with fully convex faces and edges. This can be detrimental since classical standard and naive planes are fully convex. In this paper we study several methods that builds a fully convex set from a digital set. One is particularly appealing and is based on an iterative process: this *envelope operator* solves in arbitrary dimension the problem of extending a digital set into a fully convex set, while leaving fully convex sets invariant. This extension naturally leads to digital polyhedra whose cells are fully convex. Then a relative envelope operator is proposed, which can be used to force digital planarity of fully convex sets. We provide experiments showing that our method produces coherent polyhedral models for any polyhedron in arbitrary dimension. Finally we study how we can speed up full convexity checks and envelope operations, with a worst-case complexity lowered by a factor $\boldsymbol{2^d}$ in $\mathbb{Z}^{\boldsymbol{d}}$.

Keywords: Digital geometry, Digital convexity, Full convexity, Polyhedral model, Envelope operator

1 Introduction

Convexity is a classical property in various domains of mathematics and computer science. It allows for instance guarantees for optimization, containment property via its separability with hyperplanes, and many convergence results in real or discrete analysis need convexity assumptions. While it has been primarily developed in $\mathbb{R}^d$, several extensions have been proposed in the past. Two main paths are possible for extending convexity: either going more abstract to adapt convexity to generic spaces or building more specialized versions for dedicated spaces like the digital space $\mathbb{Z}^d$ for instance. Most general extensions of convexity rely on hull systems [Lau06], K-convexity

This work has been partly funded by CoMeDiC ANR-15-CE40-0006 research grant.

and simplicial convexity [Lli02] or closure (hull) operators [And06]. Those general extensions do not necessarily embed a geometric vision of convexity, so convex sets do not have a geometric structure in the same veins as in $\mathbb{R}^d$. More resembling extensions rely on anti-matroids notably with the anti-exchange property [RS03] or cellular extensions based on discrete hyperplanes [Web01, RS03]. They induce spaces of convex sets with more geometric interpretations, but also fail to be connected in some situations. Several extensions have also been proposed in the optimization community using convexity and digital convexity as certificates of optimality [MS01]. For digital spaces $\mathbb{Z}^d$, digital convexity was first defined as the intersection of real convex sets of $\mathbb{R}^d$ with $\mathbb{Z}^d$ (e.g. see survey [Ron89]). Many works have then tried to enforce the connectedness of such sets, for instance by relying on digital lines [KR82b, Eck01] or extensions of digital functions [Kis04, Kis22]. Most works are limited to 2D, and 3D extensions do not solve all geometric issues [KR82a].

This paper considers the recently introduced notion of *full convexity* [Lac21, Lac22]. It extends digital convex sets while enforcing connectedness of fully convex sets. This notion is also computational in the sense that verifying full convexity is an easy task. Furthermore classical standard and naive planes are fully convex, so this convexity is appealing for building polyhedral models in any dimensions. However, since intersections of fully convex sets are not always fully convex, full convexity cannot be used directly for building faces and edges of polyhedra. Indeed the full convexity does not verify the monotonicity property of classical hull operators and thus fully convex hull is not a properly defined hull operator. This is a problem if we wish to build digital polyhedra in arbitrary dimension. In 3D, graceful lines and planes have been proposed in [BB02] to define edges consistent with triangular faces. It permits to fix varying arithmetical thickness between interior and boundary of digital triangles by construction but it is limited to 3D.

Our objective is to define polyhedral models in digital space $\mathbb{Z}^d$ which are based on full convexity. Our proposal lets us freely choose the thickness of digital faces, is canonic in arbitrary dimension, and benefits from the nice properties of fully convex sets. Indeed, naive, standard or even thicker pieces of arithmetical planes can be reconstructed in the proposed unified framework. In the course of this construction, we define a fully convex envelope operator, which guarantees the properties of our new polyhedral models. This paper is an extension of [FL22], and adds several construction of fully convex sets, new properties and characterizations, and several algorithms and data structures to check full convexity or compute fully convex enveloppe, with significant speed-up.

We start by defining the *fully convex envelope* in Section 2. We propose a pre-hull operator without the monotonicity property, which builds a fully convex set containing any input digital sets. Our process is iterative, fully parallel at each iteration and ends after a finite number of iterations. It solely uses classical operators in the cubical complex $\mathscr{C}^d$ associated to $\mathbb{Z}^d$. Our construction lets fully convex sets to be invariant sets. We also characterize the operator for general real sets. We then adapt this operation, in Section 3, to define a fully convex envelope *relative to another fully convex set*. Since thick enough digital planes are known to be fully convex, we can define fully convex subsets of digital planes in arbitrary dimension. We then study several models for building fully convex envelope and compare their behaviors in Section 4. The simultaneous use of a convex envelope and its relative extension are then used to build edges and faces for meshes with planar faces or meshes with non planar faces in Section 5. Experiments show that the induced *polyhedral models* are visually appealing and preserve the connectivity graphs between faces and edges of original models. Then we focus on implementation details in Section 6 to present an efficient implementation of the operators presented in the paper, all of them publicly available in an open-source library. We provide a discussion about efficient data structure and efficient computations to reduce worst-case complexity and to obtain important speed-up comparatively to the previous implementation described in [Lac22]. We then conclude the paper in Section 7, and present open questions for extending the present proposal.

2 Full convexity and fully convex envelope

2.1 Definitions

Cubical cell complex.

We consider the (cubical) cell complex $\mathscr{C}^d$ induced by the lattice $\mathbb{Z}^d$, such that its 0-cells are the points of $\mathbb{Z}^d$, its 1-cells are the open unit segments joining two 0-cells at distance 1, its 2-cells are the open unit squares formed by these segments, ..., and its d-cells are the d-dimensional unit hypercubes with vertices in $\mathbb{Z}^d$. We denote by $\mathscr{C}^d_k$ the set of its k-cells. We call *complex/subcomplex* any subset of cells of $\mathscr{C}^d$, e.g. any single cell is a subcomplex. A *digital set* is a subset of $\mathbb{Z}^d$.

The *(topological) boundary* ∂Y of a subset Y of $\mathbb{R}^d$ is the set of points in its closure but not in its interior. The *star* of a cell σ in $\mathscr{C}^d$, denoted by $\mathrm{Star}\,(\sigma)$, is the set of cells of $\mathscr{C}^d$ whose boundary contains σ and it contains the cell σ itself. The *closure* $\mathrm{Cl}\,(\sigma)$ of σ contains σ and all the cells in its boundary. In this paper, the boundary operator maps a k-cell to all its proper faces, that is k'-cells, $k' < k$ and not only $(k-1)$-cells.

Lemma 1. *$c' \in \partial c \iff \mathrm{Star}\,(c') \supsetneq \mathrm{Star}\,(c)$.*

We extend these definitions to any subcomplex K of $\mathscr{C}^d$ by taking unions:

$$\mathrm{Star}\,(K) := \bigcup_{\sigma \in K} \mathrm{Star}\,(\sigma)\,,$$
$$\mathrm{Cl}\,(K) := \bigcup_{\sigma \in K} \mathrm{Cl}\,(\sigma)\,.$$

In combinatorial topology, a subcomplex K with $\mathrm{Star}\,(K) = K$ is *open*, while being *closed* when $\mathrm{Cl}\,(K) = K$. The *body* of a subcomplex K, i.e. the union of its cells in $\mathbb{R}^d$, is written $\|K\|$. We denote by $\mathrm{Extr}\,(K) := \mathrm{Cl}\,(K) \cap \mathbb{Z}^d$ the extreme $0-$cells identified to points in $\mathbb{Z}^d$.

Intersection complex.

If Y is any subset of the Euclidean space $\mathbb{R}^d$, we denote by $\bar{\mathscr{C}}^d_k[Y]$ the set of k-cells whose topological closure intersects Y, i.e.

$$\bar{\mathscr{C}}^d_k[Y] := \{c \in \mathscr{C}^d_k, \bar{c} \cap Y \neq \emptyset\}. \tag{1}$$

Note that $\bar{c} = \|\mathrm{Cl}\,(c)\|$ for any cell c. The complex that is the union of every $\bar{\mathscr{C}}^d_k[Y]$, $0 \leqslant k \leqslant d$, is called the *intersection (cubical) complex of* Y and is denoted by $\bar{\mathscr{C}}^d[Y]$.

It is worth to note that, for any complex K, $\mathrm{Star}\,(K) = \bar{\mathscr{C}}^d[\|K\|]$. Hence, for any subset $Y \subset \mathbb{R}^d$, it is natural to define $\mathrm{Star}\,(Y) := \bar{\mathscr{C}}^d[Y]$, which coincides with the standard definition of star on subsets of $\mathscr{C}^d$ or $\mathbb{Z}^d$. It should be noted that for a digital set $X \subset \mathbb{Z}^d$, we identify its points to the 0-dimensional cells of its star. So an equality of type $K = X$ where K is a complex and X is a digital set means that the complex K only contains the 0-dimensional cells of $\mathrm{Star}\,(X)$.

$\mathrm{Star}\,()$ ***inverse.***

There are at least two natural ways of defining an inverse operation to the $\mathrm{Star}\,()$ operator. We here present those two possibilities. We always call *Skeleton* or $\mathrm{Skel}\,()$ an inverse operator to $\mathrm{Star}\,()$.

We can define a kind of inverse operation to the star using the dimensions of cells. For any complex $K \subset \mathscr{C}^d$, let:

- $\mathrm{Skel}_0\,(K) := \{c \in \mathscr{C}^d_0, \mathrm{Star}\,(c) \subset K\}$
- $\forall k = 1, \ldots, d, \quad \mathrm{Skel}_k\,(K) := \mathrm{Skel}_{k-1}\,(K) \cup \{c \in \mathscr{C}^d_k \setminus \mathrm{Star}\,(\mathrm{Skel}_{k-1}\,(K))\,, \mathrm{Star}\,(c) \subset K\}$.

Then the *(recursive) skeleton* $\mathrm{Skel}\,(K)_R$ of the complex K is equal to $\mathrm{Skel}_d\,(K)$. The next Lemma is immediate from the definition:

Lemma 2.

$$\mathrm{Skel}\,(K)_R = \bigcup_{0 \leqslant k \leqslant d} \{\, c \in \mathscr{C}^d_k, \mathrm{Star}\,(c) \subset K \text{ and } \forall c' \in K \setminus \{c\}, \mathrm{Star}\,(c) \not\subset \mathrm{Star}\,(c')\}\,.$$

This notion of skeleton is a right inverse to $\mathrm{Star}\,(\cdot)$ for open complexes:

Lemma 3. *For any complex K, K open $\iff$ $\mathrm{Star}\,(\mathrm{Skel}\,(K)_R) = K$.*

Proof Since K is open, we have $K = \mathrm{Star}\,(K)$. So, $K = \bigcup_{c \in K} \mathrm{Star}\,(c)$. To remove redundant cells in $\mathrm{Star}\,()$, we denote by $R(K)$ the set $\{c \in K \;/\; \forall c' \in K \setminus \{c\},\ \mathrm{Star}\,(c) \not\subset \mathrm{Star}\,(c')\}$. We have $K = \bigcup_{c \in R(K)} \mathrm{Star}\,(c)$ which is equal to $\mathrm{Star}\,(\mathrm{Skel}\,(K)_R)$, by Lemma 2. □

We can also define a kind of inverse operation to the star sandwiching K between another set K' and $\text{Star}(K')$. For any complex $K \subset \mathscr{C}^d$, let:

$$\text{Skel}(K)_E := \bigcap_{K' \subset K \subset \text{Star}(K')} K'.$$

This (extended) version of the skeleton also provides characterizations for open sets mainly.

Lemma 4. *For any complex K, $K \subseteq \text{Star}(\text{Skel}(K)_E)$ with equality when K is open and $\subsetneq$ otherwise. We also have $\text{Skel}(K)_E = \text{Skel}(\text{Star}(K))_E$.*

Lemma 5. *For any open complex K, $\text{Star}(\text{Skel}(K)_E) = K$.*

Proof (⊃) $K \subset \text{Star}\left(\text{Skel}(K)_E\right)$ by construction.
(⊂) $\text{Skel}(K)_E \subset K$ because $K \subset \text{Star}(K)$. Star () being increasing, $\text{Star}\left(\text{Skel}(K)_E\right) \subset \text{Star}(K) = K$ since K is open. □

We now provide a proof that this second notion of skeleton can be rephrased in the general context of *extreme cells* [And06] in closure space. Let us recall that a set function $\tau(.)$ is a closure operator if it satisfies: (i) $A \in \tau(A)$, (ii) if $A \subset B \Rightarrow \tau(A) \subset \tau(B)$ and (iii) $\tau(\tau(A)) = \tau(A)$. We indeed notice that $\text{Star}(.)$ is a closure operator on the sets of cells.

Lemma 6. *For any complex K, $\text{Skel}(K)_E$ is the set of extreme cells of K with respect to $\text{Star}(.)$.*

Proof Following [And06], the set of extreme cells of any complex $K \subset \mathscr{C}^d$ is

$$\mathscr{E}(K) := \{p \in K, p \notin \text{Star}(K \setminus \{p\})\}.$$

We can remark that $p \in \mathscr{E}(K) \iff \forall c \in K \setminus \{p\}, p \notin \text{Star}(c)$. Indeed, $p \in \mathscr{E}(K) \iff p \in K$ and $p \notin \text{Star}(K \setminus \{p\})$. But since $\text{Star}(K \setminus \{p\}) = \cup_{c \in K \setminus \{p\}} \text{Star}(c)$, and since p does not belong to the union of all Star (), p is not in any Star ().

We can now prove that $\text{Skel}(K)_E$ is $\mathscr{E}(K)$.

(⊃) Let us consider a complex $K' \subset K \subset \text{Star}(K')$. Then $\text{Star}(K') = \text{Star}(K)$, such that $\text{Star}(p) \subset \text{Star}(K')$. Hence there exists a collection of cells $c_q \in K'$ such that $\text{Star}(p) \subset \bigcup_q \text{Star}(c_q)$ and for each q, $\text{Star}(c_q) \cap \text{Star}(p) \neq \emptyset$. But the star of c_q and p intersect if and only if either $c_q = p$ or $p \in \partial c_q$ or $c_q \in \partial p$. In the first case, we have $p \in K'$. If the third case exists then we have found some $c \in K \setminus \{p\}$ with $p \in \text{Star}(c)$. This is a contradiction. If only the second case exists then by using Lemma 1, we obtain a contradiction since in this case the set $\bigcup_q \text{Star}(c_q)$ is strictly included in $\text{Star}(p)$. So, we necessarily have $p \in K'$ for any such K'. This implies $p \in \text{Skel}(K)_E$.

(⊂) If there exists $c \in K \setminus \{p\}$, $p \in \text{Star}(c)$ then $p \notin \text{Skel}(K)_E$ hence $p \in \mathscr{E}(K)$. □

We can remark that, by definition of extreme cells, any point of a digital set X is an extreme cell of $\text{Star}(X)$, because 0-dimensional cells cannot be built using Star (.) on other cells than themselves. We can further see that $\mathscr{E}(\text{Star}(X))$ is exactly X viewed as a set of 0-dimensional cells. Hence we get the following Lemma,

Lemma 7. *For any digital set X we have $\text{Skel}(\text{Star}(X))_E = X$.*

Relations between skeletons.

We know study the relations between the two previously defined notions of skeleton.

Lemma 8. *For any complex K, let us denote by $\mathring{K}$ the largest open complex included in K. Then,*

$$\text{Skel}(K)_R = \text{Skel}\left(\mathring{K}\right)_E.$$

Lemma 9. *The two* Skel () *definitions coincide on open complexes.*

In other words, on an open complex, the set of its extreme points and its skeleton coincide for the two definitions. They differ for non open complexes. Hence, in the sequel, we define the *skeleton of K* as

$$\text{Skel}(K) := \bigcap_{K' \subset K \subset \text{Star}(K')} K'. \tag{2}$$

We can thus summarize why, for open complexes, Skel () can be considered as a right inverse of Star ().

Lemma 10. *For any complex K, $K \subset \text{Star}(\text{Skel}(K))$.*

Lemma 11. *For any open complex K, $\text{Star}(\text{Skel}(K)) = K$.*

Skeleton of Star for real sets.

We provide the following characterization for the Skeleton of the Star of real sets.

Lemma 12. *Let us consider $Y \subset \mathbb{R}^d$.*
For any $c \in \mathscr{C}^d$, $c \in \mathrm{Skel}\,(\mathrm{Star}\,(Y)) \Leftrightarrow \|c\| \cap Y \neq \emptyset$ and $\|\partial c\| \cap Y = \emptyset$.

Proof
($\Rightarrow$) Let $c \in \mathscr{C}^d$ with $c \in \mathrm{Skel}\,(\mathrm{Star}\,(Y))$. Since $\mathrm{Skel}\,(\mathrm{Star}\,(Y)) \subset \mathrm{Star}\,(Y)$, we have $\bar{c} \cap Y \neq \emptyset$. Suppose that $\|c\| \cap Y = \emptyset$ then there exists $d \in \partial c$ such that $\|d\| \cap Y \neq \emptyset$. Hence $d \in \mathrm{Star}\,(Y)$. Moreover, the dimension of d is strictly lower than the dimension of c such that $\mathrm{Star}\,(c) \subset \mathrm{Star}\,(d)$.

Now, for any $e \in \mathscr{C}^d$, $e \in \mathrm{Star}\,(d) \Rightarrow d \in \mathrm{Cl}\,(e)$. Since $\|d\| \cap Y \neq \emptyset$, we get that $\bar{e} \cap Y \neq \emptyset$ such that $e \in \mathrm{Star}\,(Y)$. So $\mathrm{Star}\,(c) \subset \mathrm{Star}\,(d) \subset \mathrm{Star}\,(Y)$. Hence $c \notin \mathrm{Skel}\,(\mathrm{Star}\,(Y))$, which is a contradiction. We can then conclude that $\|c\| \cap Y \neq \emptyset$ and that $\|\partial c\| \cap Y = \emptyset$.
($\Leftarrow$) Let $c \in \mathscr{C}^d$ with $\|c\| \cap Y \neq \emptyset$ and $\|\partial c\| \cap Y = \emptyset$. Then, c is in $\mathrm{Star}\,(Y)$. Now $\mathrm{Star}\,(c) \subset \mathrm{Star}\,(e)$ with $e \in \mathscr{C}^d$ implies that $e \in \partial c$. Hence, $c \notin \mathrm{Skel}\,(\mathrm{Star}\,(Y))$ implies that there exists $e \in \partial c$ with $e \in \mathrm{Star}\,(Y)$. So $\bar{e} \cap Y \neq \emptyset$. Considering such a cell e with minimal dimension, this implies that $\|e\| \cap Y \neq \emptyset$ which is a contradiction since $\|\partial c\| \cap Y = \emptyset$. ☐

In other words, a cell belongs to the Skeleton of the Star of a real set if and only if the cell intersects the set but the cell boundary does not intersect the set.

2.2 Full convexity

For a set $A \subset \mathbb{R}^d$, its *convex hull* $\mathrm{CvxH}\,(A)$ is the intersection of all convex sets that contains A.

Definition 1 (Full convexity [Lac21, Lac22])**.** *A digital set $X \subset \mathbb{Z}^d$ is* digitally k-convex *for $0 \leqslant k \leqslant d$ whenever $\bar{\mathscr{C}}^d_k[X] = \bar{\mathscr{C}}^d_k[\mathrm{CvxH}\,(X)]$. The set X is* fully (digitally) convex *if it is digitally k-convex for all $k, 0 \leqslant k \leqslant d$.*

The following characterization will be useful:

Lemma 13. *A digital set X is fully convex iff* $\mathrm{Star}\,(X) = \mathrm{Star}\,(\mathrm{CvxH}\,(X))$.

Proof Indeed, the set X is fully convex iff

$$\forall k, 0 \leqslant k \leqslant d, \bar{\mathscr{C}}^d_k[X] = \bar{\mathscr{C}}^d_k[\mathrm{CvxH}\,(X)]$$

$$\Leftrightarrow \bar{\mathscr{C}}^d[X] = \bar{\mathscr{C}}^d[\mathrm{CvxH}\,(X)]$$
$$\Leftrightarrow \mathrm{Star}\,(X) = \mathrm{Star}\,(\mathrm{CvxH}\,(X))\,,$$

by definition of $\mathrm{Star}\,(\cdot)$. ☐

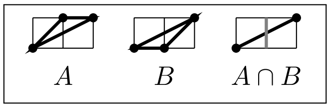
2.3 Fully convex envelope

Convex hull is one of the most fundamental tool in continuous geometry. We wish to design a digital analogue to the convex hull. The question is then how to build a fully convex set from an arbitrary digital subset of $\mathbb{Z}^d$. For instance can we build this fully convex envelope with intersections of fully convex set? We do have this rather straightforward property:

Lemma 14. *If A and B are digitally 0-convex, then $A \cap B$ is digitally 0-convex.*

Proof
Since $\mathrm{CvxH}\,(\cdot)$ is increasing, we have $\mathrm{CvxH}\,(A \cap B) \cap \mathbb{Z}^d \subset \mathrm{CvxH}\,(A) \cap \mathrm{CvxH}\,(B) \cap \mathbb{Z}^d$. Then using that A and B are digitally 0-convex, we get $\mathrm{CvxH}\,(A) \cap \mathrm{CvxH}\,(B) \cap \mathbb{Z}^d = A \cap B$. ☐

However, intersections of fully convex sets are generally not fully convex. As a very simple example, just pick $A = \{(0,0),(1,1),(2,1)\}$ and $B = \{(0,0),(1,0),(2,1)\}$, which are both fully convex. Then the set $A \cap B = \{(0,0),(2,1)\}$ is not fully convex, not even connected as seen below.



Therefore, we propose another way to build a fully convex set from an arbitrary digital set, which uses the cells intersected by the convex hull of this set, and which is defined through an iterative process.

Each iteration composes these operations, for $X \subset \mathbb{R}^d$:

$$\mathrm{FC}(X) := \mathrm{Extr}\,(\mathrm{Skel}\,(\mathrm{Star}\,(\mathrm{CvxH}\,(X))))$$

First the Euclidean convex hull of the set is computed, letting $Y = \mathrm{CvxH}\,(X)$, then its covering $\mathrm{Star}\,(Y)$ by cells of the cellular grid is determined. The skeleton of these cells is their smallest subset such that $\mathrm{Star}\,(\mathrm{Skel}\,(\mathrm{Star}\,(Y))) \supset Y$. Then for a complex K, $\mathrm{Extr}\,(K) := \|\mathrm{Cl}\,(K)\| \cap \mathscr{C}^d_0$, i.e. all the vertices of the cells of K. So finally $\mathrm{FC}(X)$ is composed of the grid vertices of the skeleton cells. The last operation implies that $\mathrm{FC}(X) \subset \mathbb{Z}^d$. Refer to Figure 1 for an illustration of FC operation and fully convex envelope computation.

Definition 2 (Fully convex envelope). *For any integer $n \geqslant 0$, the* n-th convex envelope *of $X \subset \mathbb{R}^d$ is the n times composition of operation* FC.

$$\mathrm{FC}^n(X) := \underbrace{\mathrm{FC} \circ \cdots \circ \mathrm{FC}}_{n \text{ times}}(X).$$

The fully convex envelope *of X is the limit of $\mathrm{FC}^n(X)$ when $n \to \infty$:*

$$\mathrm{FC}^*(X) := \lim_{n\to\infty} \mathrm{FC}^n(X).$$

We have to show that this process has a limit for every subset X.

Theorem 1. *For any finite digital set $X \subset \mathbb{Z}^d$, there exists a finite n such that $\mathrm{FC}^n(X) = \mathrm{FC}^{n+1}(X)$, which implies that $\mathrm{FC}^*(X)$ exists and is equal to $\mathrm{FC}^n(X)$.*

It is an immediate consequence of Lemma 15 and Lemma 16 below: the first one tells that FC is increasing, the second that X and $\mathrm{FC}(X)$ have the same bounding box.

Lemma 15. *For any $X \subset \mathbb{Z}^d$, $X \subset \mathrm{FC}(X)$.*

Proof Let $x \in X \subset \mathbb{Z}^d = \mathscr{C}_0^d$. Obviously $x \in \mathrm{CvxH}(X)$. It follows that $x \in \mathrm{Star}(\mathrm{CvxH}(X))$ and, since $\mathrm{Star}(\cdot)$ is idempotent, $\mathrm{Star}(x) \subset \mathrm{Star}(\mathrm{CvxH}(X))$. The whole star of x belonging to the subcomplex $K := \mathrm{Star}(\mathrm{CvxH}(X))$, the 0-cell x belongs to the skeleton of K. Since all 0-cells of a subcomplex are extremal points, it is an extremal point of $\mathrm{Skel}(K)$, which concludes. □

Lemma 16. *For any finite $X \subset \mathbb{Z}^d$, X and $\mathrm{FC}(X)$ have the same bounding box.*

Proof Let $p \subset \mathbb{Z}^d$ be the lowest point of the axis-aligned bounding box of X, i.e. $\forall i, 1 \leqslant i \leqslant d, p^i = \min_{z\in X} z^i$. Obviously, it is also the lowest point of the bounding box of $\mathrm{CvxH}(X)$. Let $K := \mathrm{Star}(\mathrm{CvxH}(X))$. Since $\forall x \in \mathrm{CvxH}(X), p^i \leqslant x^i$, any cell c of K that lie below point q along some coordinate axis j has a twin cell $e \in K$ in its boundary, such that e is closed along coordinate j and $e^j = p^j$. Continuing the argument along every coordinate axis k where e is below point p, we know that there is a digital point $z \in K$ in the boundary of c, such that z is not below p. Point z being a 0-cell it follows that $z \in \mathrm{Skel}(K)$ while all m-cells incident to z, $m > 0$, are not in $\mathrm{Skel}(K)$. We have just shown that no cells of $\mathrm{Skel}(K)$ can be lower than p. The reasoning is the same for the uppermost point. □

A first observation is that operation FC does not modify fully convex sets, so the fully convex envelope of a fully convex set X is X itself.

Lemma 17. *If $X \subset \mathbb{Z}^d$ is fully convex, then $\mathrm{FC}(X) = X$. So $\mathrm{FC}^*(X) = X$.*

Proof Indeed we have

$$\begin{aligned}
\mathrm{FC}(X) &= \mathrm{Extr}(\mathrm{Skel}(\mathrm{Star}(\mathrm{CvxH}(X)))) \\
&= \mathrm{Extr}(\mathrm{Skel}(\mathrm{Star}(X))) && \text{(Lemma 13)} \\
&= \mathrm{Extr}(X) && \text{(Lemma 7)} \\
&= X && (X \subset \mathbb{Z}^d)
\end{aligned}$$

□

Reciprocally, non fully convex sets are modified through operation FC.

Lemma 18. *If $X \subset \mathbb{Z}^d$ is not fully convex, then $X \subsetneq \mathrm{FC}(X)$*

Proof By Lemma 15 we already know that $X \subset \mathrm{FC}(X)$. Let us show that there is a digital point $z \in \mathrm{FC}(X)$ that is not in X. Since X is not fully convex, there exists some cell $c \in \mathrm{Star}(\mathrm{CvxH}(X))$ such that $c \notin \mathrm{Star}(X)$. It is possible that there are other cells c' in $\bar{c}$ such that $c' \in \mathrm{Star}(\mathrm{CvxH}(X))$ and $c' \notin \mathrm{Star}(X)$. To avoid ambiguities, we pick one, say b, with lowest dimension.

Let $z \in \bar{b} \cap \mathbb{Z}^d$ be a grid vertex of this cell (which may be b itself). Then $z \notin X$. Otherwise, $\mathrm{Star}(z) \subset \mathrm{Star}(X)$, hence the cell b, which belongs to $\mathrm{Star}(z)$ (through the equivalence $z \subset \bar{b} \Leftrightarrow b \in \mathrm{Star}(z)$), would thus belong to $\mathrm{Star}(X)$, a contradiction with the hypothesis.

Let us show now that $z \in \mathrm{FC}(X)$. Recall that

$$\mathrm{FC}(X) = \mathrm{Extr}(\mathrm{Skel}(\mathrm{Star}(\mathrm{CvxH}(X)))).$$

We have $b \in \mathrm{Star}(\mathrm{CvxH}(X))$. Furthermore b belongs to the skeleton of $\mathrm{Star}(\mathrm{CvxH}(X))$, since it is a cell of $\mathrm{Star}(\mathrm{CvxH}(X))$ with lowest dimension in the closure

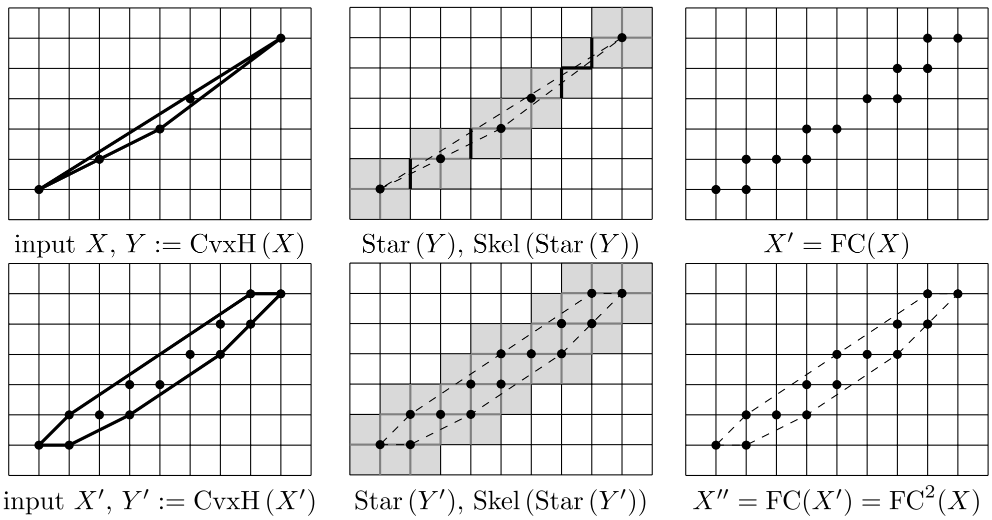


Fig. 1 Illustration of FC operation and fully convex envelope construction. Left: input digital set X and its convex hull, middle: Star (CvxH (X)) (gray and thick black) and its skeleton (thick black), right: extremal points of the skeleton, i.e. $\mathrm{FC}(X)$. Here X is digitally 0-convex but not fully convex. $\mathrm{FC}(X)$ is not even digitally 0-convex, while $\mathrm{FC}(\mathrm{FC}(X))$ is fully convex and is therefore the fully convex envelope to X.

of c. Finally grid vertex z is an extremal point of b, so belongs to $\mathrm{FC}(X)$. We conclude since $z \notin X$ holds. □

Note that the Lemma also indicates *where* operation FC add digital points. Indeed, they are the vertices of the cells touched by the convex hull but not by the digital set itself. Lemmas 17 and 18 lead immediately to a characterization of fully convex sets:

Theorem 2. *$X \subset \mathbb{Z}^d$ is fully convex iff $X = FC(X)$.*

It also induces the most important property of the fully convex envelope operation: it always outputs fully convex sets.

Theorem 3. *For any finite $X \subset \mathbb{Z}^d$, $\mathrm{FC}^*(X)$ is fully convex.*

Proof By Theorem 1, $\mathrm{FC}^*(X)$ exists and there exists some n such that $\mathrm{FC}^*(X) = \mathrm{FC}^n(X)$. Hence, $\mathrm{FC}(\mathrm{FC}^n(X)) = \mathrm{FC}^n(X)$. By Theorem 2, $\mathrm{FC}^n(X)$ is fully convex, and so is $\mathrm{FC}^*(X)$. □

The operator $\mathrm{FC}^*(.)$ is thus increasing and idempotent. It however fails to be monotone because Skel (.) is not a monotone operator with respect to inclusion. So, it is not a hull operator [And06]. Nevertheless, it induces a preorder $\mathscr{R}_{FC^*}$, i.e. a reflexive and transitive binary relation, on digital sets using

$$X \mathscr{R}_{FC^*} Y \iff \mathrm{FC}^*(X) = \mathrm{FC}^*(Y).$$

It induces equivalent classes among the set of digital sets. It has its own topology through its associated Alexandrov topology.

2.4 Algorithmic aspects

We now look at the algorithmic aspects of computing FC^*. Since the computation of FC^* is done in a loop, we compute the complexity for each iteration. At the beginning of iteration k the points set is $\mathrm{FC}^{k-1}(X)$. Using Quickhull, the convex hull can be computed in $O(nf_r/r)$ [BDH96] with n the number of input points, r the number of processed points and f_r the maximum number of facets of r vertices ($f_r = O(r^{\lfloor d/2 \rfloor}/\lfloor d/2 \rfloor!)$). Obviously $r \leqslant n$, such that the complexity is bounded by $O(f_n)$ with $f_n = O(n^{\lfloor d/2 \rfloor}/\lfloor d/2 \rfloor!)$. Here, n is the number of points in $\mathrm{FC}^{k-1}(X)$. As described in [Lac21], Star (CvxH (.)) can be computed using 2^d Quickhull calls with the morphological characterizations of full convexity. It is the most intensive part of the computation. Then, Skel and Extr are extracted by simple traversal over the volume of Star (CvxH (.)). It is thus linear in the volume of Star (CvxH (.)) which is bounded above by the volume of the bounding box of $\mathrm{FC}^{k-1}(X)$. Hence the complexity of one iteration is bounded by $O(n^{\lfloor d/2 \rfloor})$. A precise bound on the number of iterations is still under study. In practice 1-4 iterations are generally observed in 3D, but we have come along examples with depth about ten.

3 Relative fully convex envelope

We now specialize operator FC in order to stay into a given fully convex set. This creates fully convex sets relative to a given fully convex set. Given $Y \subset \mathbb{Z}^d$ a fully convex set and $X \subset Y$, the FC operator relative to Y is defined as

$$\mathrm{FC}_{|Y}(X) := \mathrm{FC}(X) \cap Y.$$

As previously, $\mathrm{FC}^n_{|Y}(X) := \mathrm{FC}_{|Y} \circ \cdots \circ \mathrm{FC}_{|Y}(X)$, composed n times. The *fully convex envelope of X relative to Y* is obtained at the limit:

$$\mathrm{FC}^*_{|Y}(X) := \lim_{n \to \infty} \mathrm{FC}^n_{|Y}(X).$$

We thus have $\mathrm{FC}^*(X) = \mathrm{FC}^*_{|\mathbb{Z}^d}(X)$. In practice, for X not included in Y, we compute $\mathrm{FC}^*_{|Y}(X \cap Y)$ to get the fully convex envelope of $X \cap Y$.

As seen on Figure 2, the relative fully convex envelope extends sets only using points of the fully convex set Y. So when considering two naive lines X and Y having disconnected intersection, both subsets $\mathrm{FC}^*_{|Y}(X \cap Y)$ and $\mathrm{FC}^*_{|X}(X \cap Y)$ are fully convex, hence are connected intersections.

Theorem 4. *For any finite $X \subset \mathbb{Z}^d$ and any fully convex set $Y \subset \mathbb{Z}^d$, the digital set $\mathrm{FC}^*_{|Y}(X \cap Y)$ is fully convex and is included in Y.*

Proof Let $X' = X \cap Y$. To see that $\mathrm{FC}^*_{|Y}(X')$ is well defined, we rely on previous properties of $\mathrm{FC}^*()$. By construction, since FC() is increasing, so is $\mathrm{FC}_{|Y}()$. Moreover Lemma 16 readily extends to say that X' and $\mathrm{FC}_{|Y}(X')$ have the same bounding box. It is also true that if X' is fully convex then $\mathrm{FC}_{|Y}(X') = X' \cap Y$ and so $\mathrm{FC}^*_{|Y}(X') = X'$. Let us now see why Lemma 18 also extends to this situation. We hence suppose that X' is not fully convex. Let us then consider any cell b such that $b \in \mathrm{Star}(\mathrm{CvxH}(X'))$ but $b \notin \mathrm{Star}(X')$. Since $\mathrm{CvxH}(X') \subset \mathrm{CvxH}(Y)$, we deduce that $b \in \mathrm{Star}(\mathrm{CvxH}(Y)) = \mathrm{Star}(Y)$ since Y is fully convex. Moreover as in Lemma 18, we have $\bar{b} \cap \mathbb{Z}^d \cap X' = \emptyset$. But since $Y \subset \mathbb{Z}^d$ and $b \in \mathrm{Star}(Y)$, we deduce that $\bar{b} \cap \mathbb{Z}^d \cap Y \neq \emptyset$. Hence at least one point in Y is added by $\mathrm{FC}_{|Y}()$. This implies that $X' \subsetneq \mathrm{FC}_{|Y}(X')$. We can thus mimic Theorem 1 and Theorem 2 to get that $\mathrm{FC}^*_{|Y}(X')$ exists and is fully convex. It is included in Y by construction. □

Arithmetical planes with thickness at least as thick as naive planes are fully convex [Lac21, Theorem 7]. Hence the set Y can be chosen to be either a naive or a standard plane. Then the fully convex hull of X relative to Y is a fully convex subset of Y containing $X \cap Y$. Hence, $\mathrm{FC}^*_{|Y}(X)$ is a simply connected piece of the arithmetical plane Y. To compute $\mathrm{FC}^*_{|Y}(.)$, we only have to incorporate the intersection with Y at each iteration. This is directly linked to the complexity of deciding if a point p is in Y. If Y is a digital plane then this complexity is constant but in general it can be up to the order of $O(\log(\sharp Y))$.

4 Specific fully convex sets

In this section, we study a variety of digital sets and examine their full convexity properties.

4.1 Fully convex sets using Minkowski's sum

The closed positive unit hypercube H^+ in $\mathbb{R}^d$ is the set $[0,1]^d$. The closed negative unit hypercube H^- in $\mathbb{R}^d$ is the set $[-1,0]^d$. Considering Minkowski's sum between a convex X and a closed unit hypercube H^+, for instance, we have that $\mathrm{CvxH}(X \oplus H^+) = \mathrm{CvxH}(X) \oplus \mathrm{CvxH}(H^+)$, or equivalently that $Y = X \oplus H^+$ is a convex set in $\mathbb{R}^d$. We can prove that $Y \cap \mathbb{Z}_d$ is a fully convex set.

Lemma 19. *Let X be a real closed convex set, then $(X \oplus H^+) \cap \mathbb{Z}^d$ is a fully convex set.*

Proof Let us denote by Y the set $X \oplus H^+$ and by Z the set $Y \cap \mathbb{Z}^d$. Let us consider the set $Y_{\mathrm{in}} = \mathrm{CvxH}(Z)$. It is clear that $Y_{\mathrm{in}} \subset Y$. We also have $Y_{\mathrm{in}} \cap \mathbb{Z}^d = Z$ due to the convexity of Y.

A cell c belongs to $\mathrm{Star}(Y_{\mathrm{in}})$ either if it is included in Y_{in} or if its closure $\bar{c}$ intersects Y_{in}. In any cases, the cell e which implies that c is in $\mathrm{Star}(Y_{\mathrm{in}})$ must intersect Y_{in}. It could be c or a cell on its boundary. So for this cell e there are two cases, (i) either it is

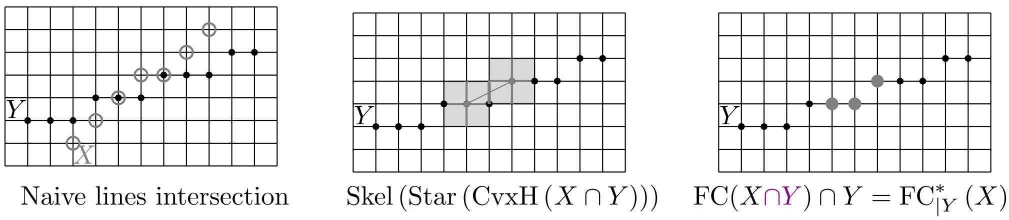


Fig. 2 Relative fully convex envelope for naive lines having disconnected intersection.

included in Y_{in}, or (ii) it is not and its intersection with Y_{in} is not empty.

In case (i) since Y_{in} is closed, the 0-cell on the boundary of e also belong to Y_{in}. Hence e is in $\text{Star}\,(Z)$. In case (ii), let us call $E = \|e\| \cap Y_{\text{in}} \neq \emptyset$. Since $E \subset Y$, there exists a set $E_x \subset X$ such that $E \subset E_x \oplus H^+$. But, $E_x \oplus H^+ \cap \mathbb{Z}^d \neq \emptyset$ by definition of H^+. Hence considering $z \in E_x \oplus H^+ \cap \mathbb{Z}^d$, we have $e \in \text{Star}\,(z)$.

So we have $\text{Star}\,(Y_{\text{in}}) \subset \text{Star}\,(Z)$ and since the converse is always true, we get the equality of those sets meaning that Z is fully convex. □

Let us remark that the main argument is that if $x \in \mathbb{R}_d$, then $(\{x\} \oplus H^+) \cap \mathbb{Z}^d \neq \emptyset$. This implies that the result is also true for H^- and for super set of the unit hypercubes such that the 2-hypercube $[-1,1]^d$.

4.2 Fully convex sets from Star

We now show that for a convex set $Y \subset \mathbb{R}^d$, $\text{Extr}\,(\text{Star}\,(Y))$ is fully convex. This implies that $\text{Extr}\,(\text{Star}\,(\text{CvxH}\,(X)))$ is fully convex for any real set X. In general, this fully convex set can be thicker than $\text{FC}^*(X)$.

Lemma 20. *Let $c \in \mathscr{C}^d$,*

$$\text{Extr}\,(\text{Star}\,(c)) = \left(\|c\| \oplus [-1,1]^d\right) \cap \mathbb{Z}^d.$$

Lemma 21. *Let $Y \subset \mathbb{R}^d$ a closed convex set, then*

$$\text{Extr}\,(\text{Star}\,(Y)) = \left(Y \oplus [-1,1]^d\right) \cap \mathbb{Z}^d.$$

Proof Let $K_\cap = \{c \in \mathscr{C}^d, \|c\| \cap Y \neq \emptyset\}$. We have $K_\cap \subset \text{Star}\,(Y)$. The converse is false when a cell boundary intersects Y but the cell does not. Let us also introduced the dilated version of $K_\cap$ as $Y_\oplus = \|K_\cap\| \oplus [-1,1]^d$. It must be noted that $Y_\oplus \subset \left(Y \oplus [-1,1]^d\right)$ and that $Y_\oplus = \bigcup_{c \in K_\cap} \|c\| \oplus [-1,1]^d$.

We note that $\|c\| \cap Y \neq \emptyset$ implies that either (i) $\|c\|$ is fully included in Y or (ii) $\|c\|$ intersects both Y and its complimentary.

In case (i), since Y is closed, $\|\bar{c}\| \subset Y$. So in particular all 0-dimensional cells of ∂c belong to Y. Hence all 0-dimensional cells in the boundary of c belong to $Y_\oplus$. This implies, using Lemma 20, that $\text{Extr}\,(\text{Star}\,(Y))$ and $\left(Y \oplus [-1,1]^d\right) \cap \mathbb{Z}^d$ coincide on those cells included in Y.

In case (ii), since Y is closed, $\|c\| \cap Y$ is closed. Hence, $(\|c\| \cap Y) \oplus [-1,1]^d$ is closed. But, using for instance ∞-norm arguments, all 0-dimensional cells in ∂c belong to $(\|c\| \cap Y) \oplus [-1,1]^d$. Still using Lemma 20, then $\text{Extr}\,(\text{Star}\,(Y))$ and $\left(Y \oplus [-1,1]^d\right) \cap \mathbb{Z}^d$ coincide on those cells not included in Y but intersected by Y.

However, we know that some cells might be in $\text{Star}\,(Y)$ but not in $K_\cap$. We hence consider such *outside* cells O. Any such cell $o \in O$ verifies $\|o\| \cap Y = \emptyset$ with $\|\bar{o}\| \cap Y \neq \emptyset$. In other word, there always exists a cell $e_o \in \partial o$ such that $\|e_o\| \cap Y \neq \emptyset$. Hence $e_o \in K_\cap$. So the 0-dimensional cells on the boundary of e_o belong to $Y_\oplus$. This implies that the 0-dimensional cells on the boundary of o not captured by case (i) and (ii) are indeed captured by the dilation of 0-dimensional cells on the boundary of e_o. So, even on those cells, $\text{Extr}\,(\text{Star}\,(Y))$ and $\left(Y \oplus [-1,1]^d\right) \cap \mathbb{Z}^d$ coincide.

□

The characterization given in Lemma 21 implies that $\text{Extr}\,(\text{Star}\,(Y))$ is obtained by a simple Minkowski's sum with the 2-hypercube. This implies the following.

Lemma 22. *Let $Y \subset \mathbb{R}^d$ a closed convex set. Then $\text{Extr}\,(\text{Star}\,(Y))$ is fully convex.*

Proof With the characterization given by Lemma 21 and Lemma 19 applied to the 2-hypercube $[-1,1]^d$, we derive the full convexity of $\text{Extr}\,(\text{Star}\,(Y))$. □

4.3 Projection of fully convex sets along an axis

We here study the stability of fully convex sets with respect to orthogonal projections along axis in $\mathbb{R}^d$. We denote by $\mathscr{P}_j$ the orthogonal projector associated to the j-th axis, which consists in omitting the j-th coordinates for all points of $\mathbb{Z}^d$. By direct extension, those projectors are defined for cells in $\mathscr{C}^d$. $\mathscr{P}_j$ are called *axis projectors*. Those projectors share the property that the image of a cell $c \in \mathscr{C}^d$ is a cell in $\mathscr{C}^{d-1}$ and those projections are the only projections for which this property is true. Moreover, the image of a point in $\mathbb{Z}^d$ by any axis projector is a point in $\mathbb{Z}^{d-1}$.

Lemma 23. *Let X be a full convex set in $\mathbb{Z}^d$ and any axis projector $\mathscr{P}_j$, then $\mathscr{P}_j(X)$ is a full convex set in $\mathbb{Z}^{d-1}$.*

Proof For the real set $Y = \mathrm{CvxH}(X)$, its star is composed of cells intersected by Y and of cells for which only their topological boundaries intersect Y (but the cell itself do not). We therefore study the behavior of those two disjoint subsets of cells respectively in case (i) and in case (ii).

Case (i). Since X is a digital set in $\mathbb{Z}^d$ then $\hat{X}_j = \mathscr{P}_j(X)$ is a digital set in $\mathbb{Z}^{d-1}$. We consider a cell $c \in \mathscr{C}^{d-1}$ intersected by $\mathrm{CvxH}\left(\hat{X}_j\right)$, the convex hull of $\hat{X}_j$. The inverse image $\mathscr{P}_j^{-1}(c)$ is an infinite open convex set. To the real set $\mathscr{P}_j^{-1}(c)$ we associate the set of cells $C_c = \{e \in \mathscr{C}^d, \|e\| \cap \mathscr{P}_j^{-1}(c) \neq \emptyset\}$. Let us denote by K the intersection $\mathscr{P}_j^{-1}(c) \cap \mathrm{CvxH}(X)$, which is a non empty open convex set K. We consider the set of cells $K_c = \mathrm{Star}(K) \cap C_c$. Since X is a fully convex set, every cell in $\mathrm{Star}(\mathrm{CvxH}(X))$ is in $\mathrm{Star}(X)$. This is in particular true for all cells in K_c such that every cell in K_c is in $\mathrm{Star}(X)$. So for each $e \in K_c$, we associate a point z_e in X such that e is in $\mathrm{Star}(z_e)$. The projection of e is, by definition, c and the projection of z_e is a point $\hat{z}_e$ in $\hat{X}_j$. We can remark that c is in $\mathrm{Star}(\hat{z}_e) \subset \mathrm{Star}\left(\hat{X}_j\right)$.

Case (ii). Now if we consider a cell c such that $\|c\| \cap \mathrm{CvxH}\left(\hat{X}_j\right) = \emptyset$. Then, c belongs to $\mathrm{Star}\left(\mathrm{CvxH}\left(\hat{X}_j\right)\right)$ if and only if there exists $c' \in \partial c$ such that $\|c'\| \cap \mathrm{CvxH}\left(\hat{X}_j\right) \neq \emptyset$. Applying the same construction that the one of the first case, we can have a point $z_{e'}$ in X such that its projection $\hat{z}_{e'}$ contains c' in its star. But, since $c' \in \partial c$, Lemma 1 implies that $\mathrm{Star}(c) \subset \mathrm{Star}(c')$. So, since $\mathrm{Star}(c') \subset \mathrm{Star}(\hat{z}_{e'})$ we have $\mathrm{Star}(c) \subset \mathrm{Star}(\hat{z}_{e'}) \subset \mathrm{Star}\left(\hat{X}_j\right)$.

We hence have proved that $\mathrm{Star}\left(\mathrm{CvxH}\left(\hat{X}_j\right)\right) \subset \mathrm{Star}\left(\hat{X}_j\right)$. Since $\mathrm{Star}(.)$ is increasing and $\hat{X}_j \subset \mathrm{CvxH}\left(\hat{X}_j\right)$, the converse is true. Hence, $\mathrm{Star}\left(\hat{X}_j\right) = \mathrm{Star}\left(\mathrm{CvxH}\left(\hat{X}_j\right)\right)$ which means that $\hat{X}_j$ is fully convex. □

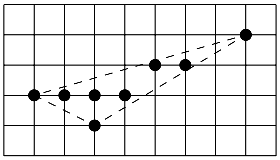


Fig. 3 A non fully convex set whose slices are all fully convex.

One could wonder if asking every axis-aligned slices of a digital object to be fully convex is a stronger property than just asking the projections to be fully convex. As for the projections, we also have the implication: it has been proved in [Lac22] that for fully convex, their slices along any canonical axis are all fully convex. However, as shown in Figure 3, the reciprocal property is generally false. The problem lies in the empty slices: an empty slice is indeed fully convex by definition of convexity of empty sets. The reciprocal property for the projections is still an open question.

Open Problem 1. *Let X be a set in $\mathbb{Z}^d$ such that for all j, $\mathscr{P}_j(X)$ is full convex. Is X a fully convex set ?*

4.4 Discussion on fully convex sets

We have presented several ways to build a fully convex set from an arbitrary digital set X. Indeed we can use the envelope operator $\mathrm{FC}^*(X)$, any asymmetric Minkowski sum such as $(\mathrm{CvxH}(X) \oplus H^+) \cap \mathbb{Z}^d$, or directly compute $\mathrm{Extr}(\mathrm{Star}(\mathrm{CvxH}(X)))$ (which is equivalent to a symmetric Minkowski sum). This is illustrated on Figure 4. Clearly these operators are not equivalent and we list on Table 1 their respective pros and cons.

We denote by $Z := f(X)$ for an operation f. Operation $(\mathrm{CvxH}(X) \oplus H^+) \cap \mathbb{Z}^d$ is the fastest to compute, but the resulting digital set Z is an asymmetric expansion of X in an arbitrary direction, with a quite large number of points in $Z \setminus X$. Operation $\mathrm{Extr}(\mathrm{Star}(\mathrm{CvxH}(X)))$ is a little bit slower to compute, yet remains symmetric. However the set $Z \setminus X$ is really large (and the worse it is in higher dimension). Both operations are not

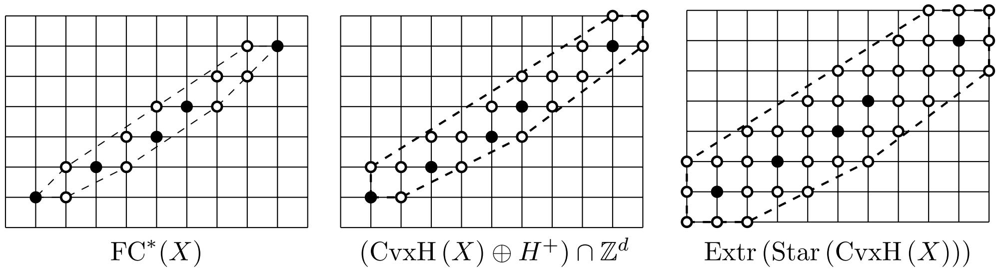


Fig. 4 Visual comparison of the different operators that can build a fully convex set from an arbitrary digital set.

operator	Id. on fully cvx.	idempotence	symmetry	$\#(Out)/\#(In)$	efficiency
$\mathrm{FC}^*(X)$	yes	yes	yes	low	unclear
$(\mathrm{CvxH}(X) \oplus H^+) \cap \mathbb{Z}^d$	no	no	no	medium	yes
$\mathrm{Extr}(\mathrm{Star}(\mathrm{CvxH}(X)))$	no	no	yes	high	yes

Table 1 Illustration of the respective qualities of the different operators that can build a fully convex set from a digital set X: (i) operator f, (ii) $f(X) = X$ holds for X fully convex, (iii) $f(f(X)) = X$ holds, (iv) whether f is invariant to symmetries and rotations, (v) size of $\#(f(X))$ wrt $\#(X)$, (vi) the efficiency to compute f ('yes' means direct bound on the computational complexity).

really envelope operators since they modify input sets even when they are fully convex.

On the contrary, the envelope operation $\mathrm{FC}^*(X)$ is generally slower to compute, since it requires sometimes several iterations of convex hull computations. However, the set $Z \setminus X$ is the smallest among all three operations. The set X is expanded symmetrically. It leaves fully convex sets invariant. Note also that the number of iterations of operator $\mathrm{FC}(\cdot)$ is upper bounded by $\#(Z \setminus X)$, and this upper bounded is greatly overestimated in general. Overall the envelope operator $\mathrm{FC}^*(\cdot)$ is much more interesting, since it has nice theoretical properties while being still computable in a reasonable amount of time.

5 Digital polyhedron

We now present digital models for Euclidean polyhedra based on envelopes. A *polyhedron* $\mathscr{P}$ is a collection of finite convex sets called *cells*, such that each cell σ is characterized by a finite number of points $V(\sigma)$ called vertices. Cell σ is a *face* of cell σ' if $V(\sigma) \subset V(\sigma')$. The vertices V of the polyhedron are the union of the vertices of all cells. Generally an abstract dimension is attached to cells, 0 for vertices, 1 for edges, 2 for faces, etc., and must be consistent with the face relation. We take an interest here in polyhedra with maximal dimension $d-1$, i.e. surfaces, whose $(d-1)$-cells are called *facets*. Figure 5, left, shows two polyhedra in 3D space: a quadrangulated surface $\mathscr{Q}$ with non planar facets and a triangulated surface $\mathscr{T}$ with planar facets.

Assuming each vertex of $\mathscr{P}$ to be a point of $\mathbb{Z}^d$, the *(generic) digital polyhedron* $\mathscr{P}^*$ associated to $\mathscr{P}$ is the collection of digital cells that are subsets of $\mathbb{Z}^d$ such that: if σ is a cell of $\mathscr{P}$, then σ^* is a cell of $\mathscr{P}^*$ with $\sigma^* := \mathrm{FC}^*(V(\sigma))$. Such a digital polyhedron is illustrated on Figure 5, top row.

When vertices of facets are coplanar, we can build a digital polyhedron whose facets are pieces of arithmetic planes. Pure simplicial complexes of dimension $d-1$ are important examples of such polyhedron. For $T \subset \mathbb{Z}^d$ made of coplanar points, let us denote by $P_1(T)$ the median standard plane (resp. $P_\infty(T)$ the median naive plane) defined by T.

The *standard (resp. naive) digital polyhedron* $\mathscr{P}_1^*$ (resp. $\mathscr{P}_\infty^*$) is the collection of digital cells subsets of $\mathbb{Z}^d$, defined as follows. For $p \in \{1, \infty\}$, if σ is a facet of $\mathscr{P}$, then σ_p^* is a cell of $\mathscr{P}_p^*$ with $\sigma_p^* := \mathrm{FC}^*_{|P_p(V(\sigma))}(V(\sigma))$. For any cell τ that is not a facet, then it has as many geometric realizations as incident facets σ and each pair (τ, σ) is digitized as $(\tau, \sigma)_p^* := \mathrm{FC}^*_{|\sigma_p^*}(V(\tau))$. Cell pairs have the same role as *half-edges* in winged-edge data structures and more generally *darts* in combinatorial maps. Note that other thicknesses could be chosen for a digital polyhedron but naive and standard are the most common ones. A standard

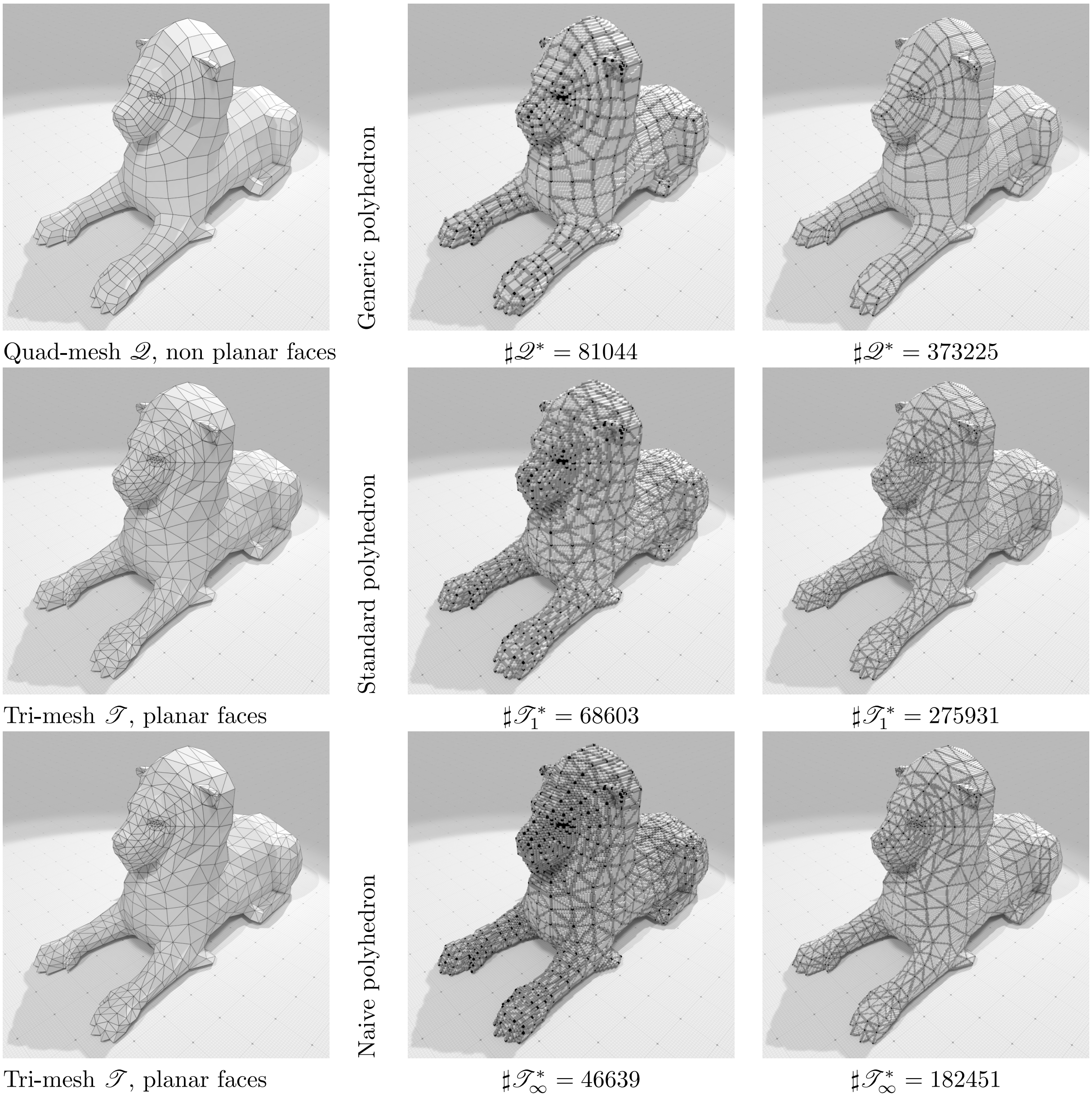
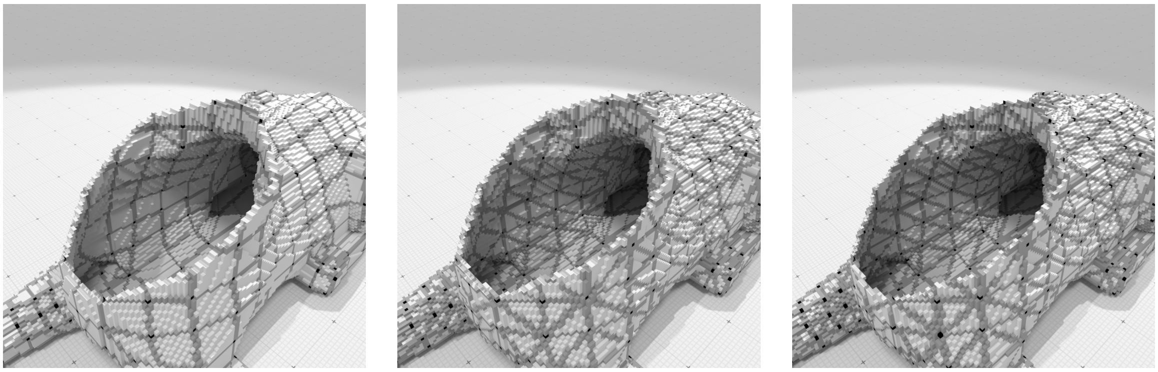


Fig. 5 Discretization of Euclidean polyhedral models without or with planar facets (left), at gridstep $h = 1$ (middle) and $h = 0.5$ (right).

(resp. naive) digital polyhedron associated to a triangulated mesh is illustrated on Figure 5, middle row (resp. bottom row). They require less digital points than the generic digital points, while keeping their separation properties. We also display on Figure 6 clipped and zoomed versions of these three polyhedral models. One can see that these models are surfacic (they are hollow) and that their thickness depends on the chosen model.

To better understand the three defined polyhedra, let us consider a single triangle and its edges and vertices: its three digital models are displayed in Figure 7. All induced cells are fully convex, but we notice that standard cells are thinner while naive cells are even thinner. What might be surprising is that relative fully convex envelope may create larger subset than expected, especially for the naive triangle example. One should keep in mind that expanding a set inside a naive plane to



Generic polyhedron $\mathscr{Q}^*$ $\sharp\mathscr{Q}^* = 81044$ | Standard polyhedron $\mathscr{T}_1^*$ $\sharp\mathscr{T}_1^* = 68603$ | Naive polyhedron $\mathscr{T}_\infty^*$ $\sharp\mathscr{T}_\infty^* = 46639$

Fig. 6 Close view on clipped polyhedral models (digitized at gridstep $h = 1$. It can be seen that the naive polyhedral model is thinner than the standard polyhedral model, while the generic polyhedral model is the thickest.

become fully convex is a very restrictive transform: edges have to expand more within naive plane P_∞ than within standard plane P_1. Of course, this is quite an extreme example and edges are narrower in most cases.

The following property is quite straightforward, but shows that every digital polyhedron covers well the cells of its associated Euclidean polyhedron, and that the inclusion/face property between cells is satisfied in the digital domain. Digitizing a polyhedron at different gridstep h is just a matter of embedding every real vertex point q as a digital vertex $q^* = \text{round}(q/h)$ (see Figure 5).

Proposition 1. *Let σ^* be a digital cell of a generic, standard or naive digital polyhedron. Then it is fully convex, hence digitally connected and simply connected. We have* $\text{Star}\,(\text{CvxH}\,(V(\sigma))) \subset \text{Star}\,(\sigma^*)$. *For any pair of cells (τ, σ) such that σ is a face of τ,* $\text{Star}\,(\tau^*)$ *covers* $\text{Star}\,(\text{CvxH}\,(V(\sigma)))$.

6 Implementation details

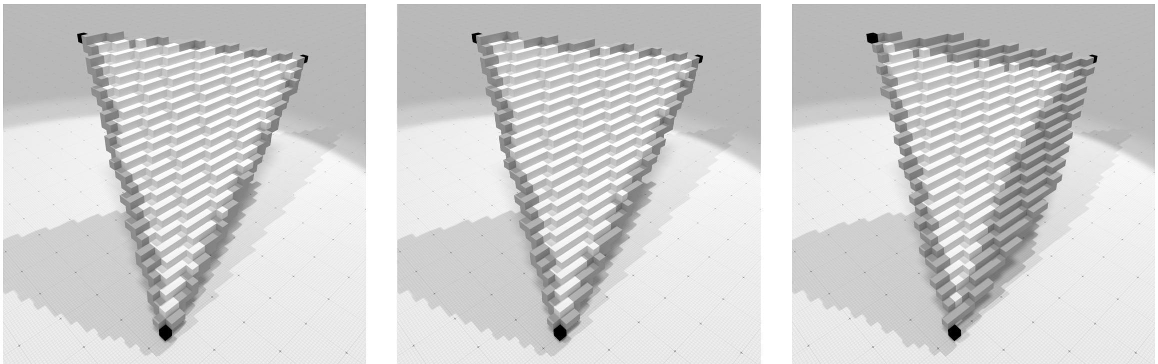
From the different definitions of full convexity and fully convex envelope, it is clear that several operations like $\text{Star}\,(\cdot)$, $\text{CvxH}\,(\cdot)$, $\text{Skel}\,(\cdot)$ must be implemented in a careful way to be efficient and dimension independent. We present in this section several improvements we have made that makes these computations much more efficient than a naive implementation. They involve an implementation of a convex hull algorithm, a new characterization of fully convex sets, a careful choice of a data structure to represent lattice points and cells and a trick to count lattice points within a polytope.

Convex hull computation.

The most common algorithm for computing the convex hull of a given set of points is the classical *quickhull* algorithm [BDH96]. The most famous dD implementation is `Qhull`.[1] However this program represents point coordinates with floating-point numbers and may return erroneous and approximate convex hulls when points are not in general position. This is a considerable issue when dealing with lattice points, where co-sphericities are extremely common. Another common dD implementation is `CGAL`,[2] which has the advantage of offering kernels with guaranteed results, at the price of slower computations. Another drawback is that it always outputs a triangulation, so co-sphericities are arbitrarily triangulated. Note also that there exists GPU implementations of quickhull [MZXZ18]. They are up to ten times faster than `Qhull`, but for ten millions of points. For sets with less than 500K points, the GPU program is much slower. Note also that these GPU implementations are limited to 2D and 3D.

We propose thus a new implementation of *quickhull* algorithm dedicated to lattice points (or rational points), as a `DGtal` package.[3] Lattice

[1] www.qhull.org
[2] www.cgal.org
[3] www.dgtal.org



generic faces $\sharp\mathscr{T}^* = 1193$ standard faces $\sharp\mathscr{T}_1^* = 985$ naive faces $\sharp\mathscr{T}_\infty^* = 567$

Fig. 7 A generic digital triangle $\mathscr{T}^*$ with its darker edges and black vertices (p, q, r) (left); corresponding standard digital triangle $\mathscr{T}_1^*$ which lies in the median standard plane $P_1(p, q, r)$ (middle); corresponding naive digital triangle $\mathscr{T}_\infty^*$ which lies in the median naive plane $P_\infty(p, q, r)$ (right).

points may have fixed-sized integer coordinates (`int32`, `int64`) or even arbitrary size integer coordinates (using `GMP` library[4]). In order to have correct geometric computations (inside/outside tests and determinant computations to get the normal vector), the user can choose also the type of integer numbers for the geometric computations. The table below shows the maximum width of the bounding box containing all lattice points in order to get exact convex-hull-computations in all cases when choosing `int64` both for coordinates and geometry computations. When the width exceeds this bound (which depends on the dimension), GMP arbitrary size integers must be used, with an approximate 25 times slow-down factor.

dimension	2	3	4	5	6
max. width	2e9	1.5e6	4e4	5e3	1.2e3

Experiments show that our implementation is as fast as `qhull` but with correct results, and is generally two times faster than `CGAL` version. See the package documentation[5] for more details.

Full convexity test and operator $\mathrm{FC}(\cdot)$ ***with one convex hull computation.***

The morphological characterization of full convexity induces an algorithm for checking full convexity that requires up to $2^d - 1$ convex hull computations. Let us recall how it works. We introduce a discrete analog of Minkowski sums of unit axis-aligned edges, squares, cubes, etc. Let $I^d = \{1, 2, \ldots, d\}$ be the set of directions of the space. Let $U_\emptyset(Z) := Z$, and, for $\alpha \subset I^d$ and $i \in \alpha$, we define recursively $U_\alpha(Z) := U_{\alpha\setminus i}(Z) \cup \mathbf{e}_i(U_{\alpha\setminus i}(Z))$. Note that $\mathbf{e}_i(Z)$ is simply the set Z translated by the i-th axis unit vector. The previous definition is consistent since it does not depend on the order of the sequence $i \in \alpha$.

Then [Lac22, Theorem 6] states: a subset $X \subset \mathbb{Z}^d$ is digitally k-convex for $0 \leqslant k \leqslant d$ iff, for any $\alpha \subset I^d$, $\#(\alpha) = k$, it holds that $U_\alpha(X) = \mathrm{CvxH}(U_\alpha(X)) \cap \mathbb{Z}^d$. It is thus *fully convex* if the previous relation holds for all $k, 0 \leqslant k \leqslant d$.

There are 2^d different subsets of I^d, and since the last d-convexity check is not necessary (see Lemma 4 of [Lac22]), this characterization induces an algorithm that must compute $2^d - 1$ convex hull and enumerate the lattice points within.

In fact, one computation is enough. We will achieve this through Theorem 5. Now let us recall the definition of mapping $\mathscr{Z}\colon \mathscr{C}^d \to \mathbb{Z}^d$ which associates to any cell σ, the digital vertex of $\bar{\sigma}$ with highest coordinates. Its restriction to $\mathscr{C}^d_\alpha$ is denoted by $\mathscr{Z}_\alpha$. It was shown in [Lac22, Lemma 8] that $\mathscr{Z}_\alpha$ is a bijection for any $\alpha \subset I^d$. Since we use $\alpha = I^d$ in the remaining of the proof, we omit it in the notations. Hence $\mathscr{Z}$ is a bijection from d-cells to lattice points.

We say that a cell $c \in \mathscr{C}^d$ is *surrounded* by a set of d-cells $D \subset \mathscr{C}^d_d$ whenever the d-dimensional cells of $\mathrm{Star}(c)$ form a subset of D, and we write $c \prec D$.

Theorem 5. *It holds that* $\mathrm{Star}(\mathrm{CvxH}(X)) = \{c \in \mathscr{C}^d, c \prec \mathscr{Z}^{-1}(\mathrm{CvxH}(U_{I^d}(X)) \cap \mathbb{Z}^d)\}$.

[4] www.gmplib.org

[5] https://dgtal-team.github.io/doc-nightly/moduleQuickHull.html

Proof Let us remind that H^+ is the unit hypercube with lowest point **0**. We recall [Lac22, Lemma 9] (specialized for $\alpha = I^d$): for any $Y \subset \mathbb{R}^d$, $\mathscr{X}(\bar{\mathscr{C}}_d^d[Y]) = (Y \oplus H^+) \cap \mathbb{Z}^d$. We recall now [Lac22, Lemma 11] which implies $\mathrm{CvxH}\,(X) \oplus H^+ = \mathrm{CvxH}\,(U_{I^d}(X))$. Applying [Lac22, Lemma 9] and this property on $Y = \mathrm{CvxH}\,(X)$ gives:

$$\begin{aligned}\mathscr{X}(\bar{\mathscr{C}}_d^d[\mathrm{CvxH}\,(X)]) &= (\mathrm{CvxH}\,(X) \oplus H^+) \cap \mathbb{Z}^d \\ &= \mathrm{CvxH}\,(U_{I^d}(X)) \cap \mathbb{Z}^d, \end{aligned} \quad (3)$$

which tells that $\mathrm{CvxH}\,(U_{I^d}(X)) \cap \mathbb{Z}^d$ corresponds one-to-one with the d-cells touched by $\mathrm{CvxH}\,(X)$.

It remains thus to be shown that $\mathrm{Star}\,(\mathrm{CvxH}\,(X)) = \{c \in C, c \prec \bar{\mathscr{C}}_d^d[\mathrm{CvxH}\,(X)]\}$, or in other terms, $\bar{\mathscr{C}}^d[\mathrm{CvxH}\,(X)] = \{c \in C, c \prec \bar{\mathscr{C}}_d^d[\mathrm{CvxH}\,(X)]\}$. Clearly the d-cells of these two sets are the same. Pick any k-cell c in $\mathscr{C}^d$, $k < d$.

If $c \in \bar{\mathscr{C}}^d[\mathrm{CvxH}\,(X)]$, so $\bar{c} \cap \mathrm{CvxH}\,(X) \neq \emptyset$. But for any $e \in \mathrm{Star}\,(c)$, $\bar{e} \supset \bar{c}$ so we also have $\bar{e} \cap \mathrm{CvxH}\,(X) \neq \emptyset$. Letting E be the d-cells of $\mathrm{Star}\,(c)$. Then every cell of E touches $\mathrm{CvxH}\,(X)$, so $E \subset \bar{\mathscr{C}}_d^d[\mathrm{CvxH}\,(X)]$, so $c \prec E \subset \bar{\mathscr{C}}_d^d[\mathrm{CvxH}\,(X)]$.

Conversely, let c be any k-cell such that $c \prec \bar{\mathscr{C}}_d^d[\mathrm{CvxH}\,(X)]$. Let E be the d-cells of $\bar{\mathscr{C}}_d^d[\mathrm{CvxH}\,(X)]$ surrounding c. Let us denote $e_1, \ldots, e_n$ the cells of E (a finite set with $n := 2^{d-k}$ elements). Since each cell e_i touches $\mathrm{CvxH}\,(X)$, then there exists for each one a point $x_i \in \bar{e}_i$ with $x_i \in \mathrm{CvxH}\,(X)$. By convexity, $\mathrm{CvxH}\,(\{x_i\}_{i=1,\ldots,n}) \subset \mathrm{CvxH}\,(X)$. But c is at the center of its surrounding d-cells (e_i), so according to Lemma 24 (in Appendix), we have $\mathrm{CvxH}\,(\{x_i\}_{i=1,\ldots,n}) \cap \bar{c} \neq \emptyset$. We have just found a common real point between $\mathrm{CvxH}\,(X)$ and $\bar{c}$, so $c \in \bar{\mathscr{C}}^d[\mathrm{CvxH}\,(X)]$. □

Since Lemma 13 tells that a digital set X is fully convex iff $\mathrm{Star}\,(X) = \mathrm{Star}\,(\mathrm{CvxH}\,(X))$, then Theorem 5 above states that $\mathrm{Star}\,(\mathrm{CvxH}\,(X))$ can be computed directly from $\mathrm{CvxH}\,(U_{I^d}(X)) \cap \mathbb{Z}^d$, which involves only one convex hull computation and lattice point enumeration within the polytope. This theorem also shows that $\mathrm{Star}\,(\mathrm{CvxH}\,(X))$ in operator $\mathrm{FC}(\cdot)$ also requires only one convex hull computation.

A row-oriented data structure for sets of lattice points.

The idea here is to notice that the lattice points within a polytope form an interval of consecutive points along any axis. And more generally lattice sets representing digital shapes are not composed of randomly distributed points and can be compactly and efficiently represented with intervals. Therefore we represent a set of integers N as an *interval sequence* $J(N)$ of k integral pairs (a_i, b_i) such that:

- $u \in N \Leftrightarrow \exists i, 1 \leqslant i \leqslant k, a_i \leqslant u \leqslant b_i$ (set),
- $\forall i, 1 \leqslant i \leqslant k, a_i \leqslant b_i$ (sorted sequence),
- $\forall i, 1 \leqslant i \leqslant k-1, b_i < a_{i+1} - 1$ (minimality).

Given an axis direction j, we define the *row representation* of a set of lattice points X as an associative map $M^j(X)$ from lattice points to interval sequences (i.e. dictionary), as follows:

If $x \in \mathbb{Z}^d$, we denote by $\hat{x}^j$ the projected point of $\mathbb{Z}^{d-1}$ with omitted coordinate x^j. For $p \in \mathbb{Z}^{d-1}$, let $X_p^j = \{x \in X, \hat{x}^j = p\}$. It is clear that every point x of X_p^j can be identified by exactly one integer, its j-th coordinate x^j. We define thus the set of integers $N(X_p^j)$ as $\{x^j, x \in X_p^j\}$. Then the value of $M^j(X)(p)$ is defined as the interval sequence $J(N(X_p^j))$.

Testing if a point q belongs to X is equivalent to check if $q^j \in M_j(X)(\hat{q}^j)$, i.e. $q^j \in J(N(X_{\hat{q}^j}^j))$. This data structure is implemented as the class `LatticeSetByIntervals`[6] in `DGtal`. A few further remarks on the row representation of a lattice set:

- its worst-case space complexity is the same the usual set data structure;
- per row, a sequence of intervals is more compact to store than a set of points except when there is only one or two points on the row (and the higher the dimension the more compact is the row representation);
- if the lattice set is digitally convex, each row stores at most one interval;
- no data is associated to an empty row in the associative container;
- checking if a point belongs to a row representation is at least as fast as checking if a point belongs to the usual set data structure (faster on average);
- counting the number of points is much faster (amounts to count $b_i - a_i + 1$ for each interval);
- usual set operations (inclusion, intersection, union, etc) are much faster in practice in row representation, and worst-cases are the same;
- a set of lattice cells can easily be represented as a set of lattice points through the Khalimsky

[6] https://dgtal-team.github.io/doc-nightly/classDGtal_1_1LatticeSetByIntervals.html

coordinates of the cells, and their induced row representation is also efficient;

- any set of lattice cells defined by the intersection with a polytope also induces a row representation with one interval per row;
- Operations $\mathrm{Star}(\cdot)$, $\mathrm{Skel}(\cdot)$ and $\mathrm{Extr}(\mathrm{Skel}(\cdot))$ are also efficiently implemented in row representation;
- the best choice for direction j is generally the direction of the set where it is the most elongated.

Enumerating lattice points within a polytope.

Although Barvinok's theory [Bar94] leads to theoretically faster algorithms for enumerating lattice points within a polytope, it is not straightforward to implement. We are aware of only one complete implementation, `LattE`,[7] and it cannot directly be interfaced with C/C++ libraries. Their command line interface is unfortunately slow and it is yet not clear for what size of data that kind of approach can be faster than naive enumeration within the bounding box of the polytope.

We choose to optimize the naive enumeration within the bounding box of the polytope. The idea is to examine the intersection of axis-aligned rays with the polytope, to extract the interval of lattice points that is common to the ray and the polytope.

More precisely, the convex hull computation builds a finite polytope P with m facets, defined as m inequalities $\mathbf{n}_i \cdot \mathbf{x} \leq \mu_i$, for $\mathbf{x}$ a lattice point of $\mathbb{Z}^d$. Let B be the tight bounding box of P and let j be the axis direction where B is the most elongated. Let $B_j := \mathscr{P}_j(B)$ be the projection of B along direction j, hence a rectangular domain of $\mathbb{Z}^{d-1}$. Let also r and s be the minimal and maximal coordinates of B along j. For every point $p \in B_j$, with $\mathbf{q} := (p^1, \dots, p^{j-1}, r, p^j, \dots, p^{d-1})$, we compute the intersection of the row along j containing p with the polytope P as follows:

$$x_M := \max_{\mathbf{n}_i^j > 0} \left\{ x \in \mathbb{Z} \;/\; x \leqslant 1 + \frac{\mu_i - \mathbf{q} \cdot \mathbf{n}_i}{\mathbf{n}_i^j} \right\},$$

$$x_m := \min_{\mathbf{n}_i^j < 0} \left\{ x \in \mathbb{Z} \;/\; x \geqslant \frac{\mu_i - \mathbf{q} \cdot \mathbf{n}_i}{\mathbf{n}_i^j} \right\}.$$

Then the interval of integer coordinates satisfying the i-th inequality is $[r + x_m, r + x_M[$ which may be empty. We then intersect all the intervals given by every inequality to obtain an interval $[a, b[$. If it is not empty, the lattice points of polytope P along the row align with j and going through p are exactly:

$$\{\mathbf{q} \in \mathbb{Z}^d, \text{ s.t. } \mathbf{q} = p + x\mathbf{e}_j, a \leqslant x < b, x \in \mathbb{Z}\},$$

where p is canonically injected in $\mathbf{Z}^d$ with null j-th coordinate. Computing x_m and x_M can be done in constant time using Euclidean division, assuming integers fit in a standard 32-bits or 64-bits register. We provide in Algorithm 1 and Algorithm 2 the detailed code that computes interval $[a, b[$.

Algorithm 1 UPDATE: given an interval of integers $[a, b[$, compute its subset $[a', b'[$ such that $\forall x, a' \leqslant x < b', (p + x\mathbf{e}_j) =: \mathbf{x}$ satisfies $\mathbf{n} \cdot \mathbf{x} \leq \mu$.

Input (a, b): an interval of integers $[a, b[$
Input $(\mathbf{n}, \mu)$: the inequalities $\mathbf{n} \cdot \mathbf{x} \leq \mu$
Input j: an integer such that $1 \leqslant j \leqslant d$
Input p: a point of $\mathbb{Z}^d$ with $p^j = 0$
Output (a', b'): an interval of integers $[a', b'[$

$p^j \leftarrow a$
$c \leftarrow \mathbf{n} \cdot p$
if $(n^j = 0) \wedge (c > \mu)$ **then**
  $b \leftarrow a$ ▷ Empty interval
else if $n^j > 0$ **then**
  $d \leftarrow \mu - c$
  if $d < 0$ **then**
    $b \leftarrow a$ ▷ Empty interval
  else
    $b \leftarrow \min(b, a + d/n + 1)$
  end if
else if $n^j < 0$ **then**
  $d \leftarrow c - \mu$
  if $d \geqslant 0$ **then**
    $a \leftarrow \max(a, a + (d - n^j - 1)/(-n^j))$
  end if
end if
return (a, b)
▷ Note that / denotes the Euclidean division

It is clear that the complexity of computing the lattice points within polytope P along the row passing by p and aligned with axis j is $O(m)$ as worst case. The global complexity is thus

[7] https://www.math.ucdavis.edu/~latte/

Algorithm 2 ComputeIntersection: compute the intersection of the row along j containing point p with the polytope P as an interval of integer coordinates $[a, b[$.

Input (r, s): the coordinates of the bounding box of the polytope along direction j
Input $(\mathbf{n}_i, \mu_i)_{i=1,\dots,m}$: the m inequalities $\mathbf{n}_i \cdot \mathbf{x} \leq \mu_i$ of the polytope
Input j: an integer such that $1 \leqslant j \leqslant d$
Input p: a point of $\mathbb{Z}^d$ with $p^j = 0$
Output (a, b): an interval of integers

```
i ← 1
(a, b) ← (r, s + 1)
while (i ⩽ m) ∧ (a < b) do
    (a, b) ← Update((a, b), (n_i, μ_i), j, p)
    i ← i + 1
end while
return (a, b)
```

$O(m\#(B_j))$, which is much better than the naive $O(m\#(B))$.

This approach to lattice point enumeration has been coded in class `BoundedLatticePolytopeCounter`[8] and experiments in 2D, 3D and 4D indicates a practical speed-up between 5× to 200× depending on the size of convex shapes.

Note finally that this "row intersection" approach to lattice point enumeration within the polytope allows a straightforward build of a row representation for this set of lattice points.

7 Conclusion and perspectives

We have studied in this paper several functions for building fully convex sets containing an arbitrary digital set. Among them, the operator $\mathrm{FC}^*(.)$ is particularly interesting since it has all the property of an envelope operator. Indeed, for any digital set X, $\mathrm{FC}^*(X)$ is proved to be fully convex and $X \subset \mathrm{FC}^*(X)$. Furthermore this operator leaves fully convex sets unchanged. Moreover, the operator is well defined in arbitrary dimension as well as computable. This operator can be restricted to stay within a fully convex set Y, leading to the relative envelope operator $\mathrm{FC}^*_{|Y}(X)$. It builds fully convex sets within Y. Since classical naive and standard planes are fully convex, this leads to a straightforward computation of digital analogues to polyhedral models of $\mathbb{R}^d$. The obtained results are quite appealing: we can control the incidence relationship between cells, while their full convexity guarantees their topological and geometrical properties. These digital polyhedral models embrace both meshes with planar or non planar faces. We also describe how to provide an efficient implementation of the envelope operator when the dimension increases. We provide a publicly available implementation in the `DGtal`[9] library.

In future works, we would like to study more precisely the iterative process of $\mathrm{FC}^*(.)$, in order to localize where full-convexity defects reside. This could further accelerate the operator by providing more practical bounds on the number of iterations. Incremental quickhull should also be considered. A more general goal is to extend the envelope process to a true convex-hull operator. The difficulty is to ensure the monotone property. In case of success, full convexity would then be a digital analogue to convexity for digital spaces.

A Useful results

We relate convexity and intersections with the cubical grid complex. The following property is useful to demonstrate that we can get a fast implementation of $\mathrm{Star}(\mathrm{CvxH}(\cdot))$.

Lemma 24. *Let c be a k-cell of $\mathscr{C}^d$ and let $D = (\sigma_1, \dots, \sigma_n)$ be the d-dimensional cells surrounding c (i.e., $\mathrm{Star}(c) \cap \mathscr{C}^d_d = D$), with $n = 2^{d-k}$. Picking one point $\mathbf{x}_i$ in each $\bar{\sigma}_i$, then it holds that there exists a point of $\bar{c}$ that belongs to $\mathrm{CvxH}(\{\mathbf{x}_i\}_{i=1,\dots,n})$.*

Proof Without loss of generality, we can choose the cell c to have its infimum point at position $\mathbf{0}$. So the points of $\bar{c}$ have their i-th coordinate equal to 0 if c is closed along direction i while they have their i-th coordinate in $[0, 1]$ if c is open along direction i. If $\mathbf{x}$ is a point, we denote by x^j its j-th coordinate.

The proof is by induction on $d-k$. For $k = d-1$, let j be the axis where the $(d-1)$-cell c is closed. D is composed of two d-cells. Choosing σ_1 to be the one further

[8] See package Digital Convexity and class BoundedLatticePolytopeCounter.

[9] www.dgtal.org

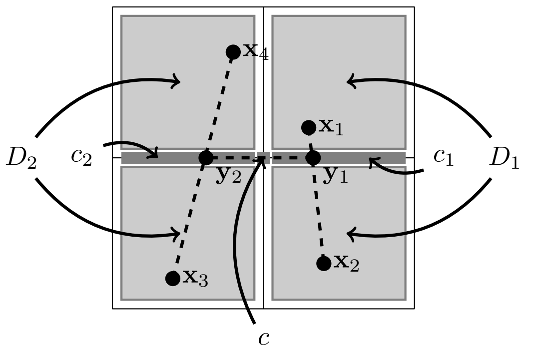


Fig. 8 Illustration of the proof of Lemma 24. The induction divide the problem into two subproblems carried by sets of d-cells D_1 and D_2. On each subproblem D_i, it is possible to build a point $\mathbf{y}_i$ on cell $\bar{c}_i$ that is a convex combination of points $\mathbf{x}_{2i+1}$ and $\mathbf{x}_{2i+2}$. Then $\mathbf{y}$ is a convex combination of points $\mathbf{y}_1$ and $\mathbf{y}_2$, and belongs to $\bar{c}$. Since c is a 0-cell here, we have $\mathbf{y} = c$.

along axis j, we have necessarily $\mathbf{x}_1^j \geqslant 0$ and $\mathbf{x}_2^j \leqslant 0$. Assume both coordinates are zero, then either $\mathbf{x}_1$ or $\mathbf{x}_2$ belongs to $\bar{c}$ and obviously to $\mathrm{CvxH}(\{\mathbf{x}_1, \mathbf{x}_2\})$, and we conclude this case. If at least one of x_1^j, x_2^j is non null, then $x_1^j - x_2^j$ is strictly positive. Let $\lambda = \frac{-x_2^j}{x_1^j - x_2^j}$. Clearly $0 \leqslant \lambda \leqslant 1$. Let now $\mathbf{y} := \lambda \mathbf{x}_1 + (1-\lambda)\mathbf{x}_2$. This point $\mathbf{y}$ is in $\mathrm{CvxH}(\{\mathbf{x}_1, \mathbf{x}_2\})$, being a convex combination of two points of this set. Now a short computation gives $y^j = 0$. Along every direction $m \neq j$, cells c and σ_1, σ_2 are open, so $0 \leqslant x_1^m \leqslant 1$, $0 \leqslant x_2^m \leqslant 1$. It follows that $0 \leqslant y^m \leqslant 1$ since $\mathbf{y}$ is a convex combination of these points. We have just shown that $\mathbf{y} \in \bar{c}$, which concludes for this case.

Let us assume that the property holds for a given $d-k$, $1 \leqslant d-k < d$, and let us show that it is true $d-k+1$. The proof is illustrated on Figure 8. Pick j a direction where c is closed. Let D_1 (resp. D_2) be the d-cells surrounding c having positive (resp. negative) j-th coordinate. Let also c_1 (resp. c_2) be the $d-k$-cell such that $\mathrm{Star}(c_1) = D_1$ (resp. $\mathrm{Star}(c_2) = D_2$). Then by induction we have a point $\mathbf{y}_1 \in \bar{c}_1$ and a point $\mathbf{y}_2 \in \bar{c}_2$ that both belong to $\mathrm{CvxH}\left(\{\mathbf{x}_i\}_{i=1,\dots,n}\right)$. The same reasoning as above is made to compute a coefficient $\lambda = \frac{-y_2^j}{y_1^j - y_2^j}$. One can check that, denoting $\mathbf{y} := \lambda \mathbf{y}_1 + (1-\lambda)\mathbf{y}_2$, we have that: $y^j = 0$, $y^m = 0$ for every direction m where c_1 or c_2 (or c) is closed (by induction), and $0 \leqslant y^{m'} \leqslant 1$ for every direction m' where c_1 or c_2 (or c) is open. We have just shown $\mathbf{y} \in \bar{c}$ and also $\mathbf{y} \in \mathrm{CvxH}\left(\{\mathbf{x}_i\}_{i=1,\dots,n}\right)$ since $\mathrm{CvxH}(\{\mathbf{y}_1, \mathbf{y}_2\}) \subset \mathrm{CvxH}\left(\{\mathbf{x}_i\}_{i=1,\dots,n}\right)$ by convexity. □

Declarations

Ethical Approval

Not applicable.

Competing interests

The authors have no competing interests as defined by Springer, or other interests that might be perceived to influence the results and/or discussion reported in this paper.

Authors' contributions

These authors contributed equally to this work.

Funding

This work has been partly funded by Agence Nationale de la Recherche, CoMeDiC ANR-15-CE40-0006 research grant.

Availability of data and materials

Not applicable.

References

[And06] Kazutoshi Ando. Extreme points axioms for closure spaces. *Discrete Mathematics*, 306:3181–3188, 2006.

[Bar94] Alexander I. Barvinok. Computing the Ehrhart polynomial of a convex lattice polytope. *Discrete & Computational Geometry*, 12(1):35–48, 1994.

[BB02] Valentin E. Brimkov and Reneta P. Barneva. Graceful planes and lines. *Theoretical Computer Science*, 283(1):151–170, 2002.

[BDH96] C. Bradford Barber, David P. Dobkin, and Hannu Huhdanpaa. The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software*, 22 (4):469–483, 1996.

[Eck01] Ulrich Eckhardt. Digital lines and digital convexity. In *Digital and Image Geometry, Advanced Lectures [Based on a Winter School Held at Dagstuhl Castle, Germany in December 2000]*, page 209–228. Springer-Verlag, 2001.

[FL22] Fabien Feschet and Jacques-Olivier Lachaud. Full convexity for polyhedral models in digital spaces. In Étienne Baudrier, Benoît Naegel, Adrien Krähenbühl, and Mohamed Tajine, editors, *Discrete Geometry and Mathematical Morphology*, volume 13493 of *Lecture Notes in Computer Science*, pages 98–109, Cham, 2022. Springer International Publishing.

[Kis04] Christer Oscar Kiselman. Convex functions on discrete sets. In R. Klette and J. Žunić, editors, *Combinatorial Image Analysis. 10th International Workshop, IWCIA*, LNCS 3322, pages 443–457. Springer-Verlag, 2004.

[Kis22] Christer Oscar Kiselman. *Elements of Digital Geometry, Mathematical Morphology, and Discrete Optimization.* World Scientific, 2022.

[KR82a] Chul E. Kim and Azriel Rosenfeld. Convex digital solids. *IEEE Trans. Pattern Anal. Machine Intel.*, 6:612–618, 1982.

[KR82b] Chul E. Kim and Azriel Rosenfeld. Digital straight lines and convexity of digital regions. *IEEE Trans. Pattern Anal. Machine Intel.*, 2:149–153, 1982.

[Lac21] Jacques-Olivier Lachaud. An alternative definition for digital convexity. In J. Lindblad, F. Malmberg, and N. Sladoje, editors, *Discrete Geometry and Mathematical Morphology - First International Joint Conference, DGMM 2021, Uppsala, Sweden, May 24-27, 2021, Proceedings*, volume 12708 of *LNCS*, pages 269–282. Springer, 2021.

[Lac22] Jacques-Olivier Lachaud. An alternative definition for digital convexity. *Journal of Mathematical Imaging and Vision*, 64:718–735, 2022.

[Lau06] Dietlinde Lau. *Function algebras on finite sets.* Springer-Verlag, 2006.

[Lli02] Juan-Vicente Llinares. Abstract convexity, some relations and applications. *Optimization*, 51(6):797–818, 2002.

[MS01] Kazuo Murota and Akiyoshi Shioura. Relationship of M-/L-convex functions with discrete convex functions by Miller and Favati–Tardella. *Discrete Applied Mathematics*, 115:151–176, 2001.

[MZXZ18] Gang Mei, Jiayin Zhang, Nengxiong Xu, and Kunyang Zhao. A sample implementation for parallelizing divide-and-conquer algorithms on the gpu. *Heliyon*, 4(1):e00512, 2018.

[Ron89] Christian Ronse. A bibliography on digital and computational convexity (1961–1988). *IEEE Trans. Pattern Anal. Machine Intel.*, 11(2):181–190, 1989.

[RS03] Anthony J. Roy and John G. Stell. Convexity in discrete space. In W. Kuhn, M. Worboys, and S. Timpf, editors, *COSIT*, pages 253–269. LNCS 2825, Springer-Verlag, 2003.

[Web01] Julian Webster. Cell complexes and digital convexity. In Gilles Bertrand, Atsushi Imiya, and Reinhard Klette, editors, *Digital and Image Geometry: Advanced Lectures*, volume 2243 of *Lecture Notes in Computer Science*, pages 272–282. Springer Berlin Heidelberg, 2001.