Tracing isomanifolds in R^d in time polymonial in d using Coxeter-Freudenthal-Kuhn triangulations

Jean-Daniel Boissonnat, Siargey Kachanovich, Mathijs Wintraecken

Tracing Isomanifolds in $\mathbb{R}^d$ in Time Polynomial in d using Coxeter-Freudenthal-Kuhn Triangulations

Jean-Daniel Boissonnat
Université Côte d'Azur, Inria
[Sophia-Antipolis, France]

Siargey Kachanovich
Université Côte d'Azur, Inria
[Sophia-Antipolis, France]

Mathijs Wintraecken
IST Austria
[Klosterneuburg, Austria]

Abstract

Isomanifolds are the generalization of isosurfaces to arbitrary dimension and codimension, i.e. submanifolds of $\mathbb{R}^d$ defined as the zero set of some multivariate multivalued smooth function $f : \mathbb{R}^d \to \mathbb{R}^{d-n}$, where n is the intrinsic dimension of the manifold. A natural way to approximate a smooth isomanifold $\mathcal{M}$ is to consider its Piecewise-Linear (PL) approximation $\hat{\mathcal{M}}$ based on a triangulation $\mathcal{T}$ of the ambient space $\mathbb{R}^d$. In this paper, we describe a simple algorithm to trace isomanifolds from a given starting point. The algorithm works for arbitrary dimensions n and d, and any precision D. Our main result is that, when f (or $\mathcal{M}$) has bounded complexity, the complexity of the algorithm is polynomial in d and $\delta = 1/D$ (and unavoidably exponential in n). Since it is known that for $\delta = \Omega(d^{2.5})$, $\hat{\mathcal{M}}$ is $O(D^2)$-close and isotopic to $\mathcal{M}$, our algorithm produces a faithful PL-approximation of isomanifolds of bounded complexity in time polynomial in d. Combining this algorithm with dimensionality reduction techniques, the dependency on d in the size of $\hat{M}$ can be completely removed with high probability. We also show that the algorithm can handle isomanifolds with boundary and, more generally, isostratifolds. The algorithm for isomanifolds with boundary has been implemented and experimental results are reported, showing that it is practical and can handle cases that are far ahead of the state-of-the-art.

2012 ACM Subject Classification Theory of computation → Computational geometry

Keywords and phrases Coxeter triangulation, Kuhn triangulation, permutahedron, PL-approximations, isomanifolds/solution manifolds/isosurfacing

Related Version A full version of this paper is available at `https://hal.archives-ouvertes.fr/hal-`

Funding The research leading to these results has received funding from the European Research Council (ERC) under the European Union's Seventh Framework Programme (FP/2007-2013) / ERC Grant Agreement No. 339025 GUDHI (Algorithmic Foundations of Geometry Understanding in Higher Dimensions).
Jean-Daniel Boissonnat: Supported by the French government, through the 3IA Côte d'Azur Investments in the Future project managed by the National Research Agency (ANR) with the reference number ANR-19-P3IA-0002.
Mathijs Wintraecken: Supported by the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 754411.

Acknowledgements We thank Dominique Attali, Guilherme de Fonseca, Arijit Ghosh, Vincent Pilaud and Aurélien Alvarez for their comments and suggestions. We also acknowledge the reviewers.

1 Introduction

Given a surface represented in $\mathbb{R}^3$ as the zero set of a function $f : \mathbb{R}^3 \to \mathbb{R}$, the goal of isosurfacing is to find a piecewise linear (PL) approximation of the surface. This question naturally extends to isomanifolds of higher dimensions and codimensions defined as the zero set of multivariate multivalued smooth functions $f : \mathbb{R}^d \to \mathbb{R}^{d-n}$. Isosurfaces play a crucial role in medical imaging, computer graphics and geometry processing [36]. Higher dimensional isomanifolds are also of fundamental importance in many fields like statistics [16], dynamical systems [41], econometrics, or mechanics [36].

State-of-the-art. The most widely used algorithm to trace isomanifolds is the Marching Cube (MC) algorithm and its numerous variants [29, 44]. The MC algorithm uses a cubical grid to tesselate the ambient space. In many applications in 3-dimensions, the ambient space is decomposed into unstructured tetrahedral meshes, which led to the development of a variant of the MC algorithm named the Marching Tetrahedra algorithm. In higher dimensions, any tessellation of the ambient space has a complexity that depends exponentially on the ambient dimension. Hence a key to extending marching algorithms to higher dimensions is to circumvent the curse of dimensionality by using an *implicit* representation of the ambient tessellation. This is impossible for general triangulations but easy to do if one uses a grid. However, using a grid has other drawbacks and is not sufficient to break the exponential barrier. The reason for this is that the number of configurations inside a cubical cell grows exponentially with the dimension [44]. Hence the most promising approach seems to be to subdivide the ambient space $\mathbb{R}^d$ using a highly regular triangulation such as the Freudenthal-Kuhn triangulation. Some early work along this direction has been published in Applied Mathematics [2, 25, 41], and a slightly more recent paper by Dobkin et al. [23] attracted the interest of the Computer Graphics community to the related Coxeter triangulations. Dobkin et al. however only considered the case of curves ($n = 1$). The most advanced work we are aware of is due to Min [35]. Min's method uses the Freudenthal-Kuhn triangulation over a dyadic grid of $\mathbb{R}^d$ and applies to isomanifolds of any dimension and codimension. The time complexity of Min's method is, with our notations, $O(\delta^n \log \delta)$, where $\delta = 1/D$ and D is the maximal diameter of the simplices. The ambient dimension d is a constant hidden in the big O. The fact that the exponent of δ is the intrinsic dimension n, and not the ambient dimension d is a clear improvement over earlier methods. However, although not explicitly analysed by Min, the complexity in d remains exponential, and the method seems to be limited to small ambient dimensions. Experimental results are only reported in 3, and 4D.

Contributions. This paper discusses an efficient algorithm to compute a PL-approximation of isomanifolds. We extend the work of Dobkin et al. [23] and describe a simple algorithm to trace an n-dimensional isomanifold $\mathcal{M}$ of $\mathbb{R}^d$ for arbitrary n and d. Our algorithm uses any triangulation of a family of regular triangulations of $\mathbb{R}^d$ that includes the Coxeter and the Freudenthal-Kuhn triangulations. Contrary to Min [35], our results are obtained with a uniform triangulation leading to a very simple algorithm. Key to our results, is a data structure that can implicitly store the full facial structure of such triangulations. The data structure is very compact and allows to retrieve the faces or the cofaces of a simplex of any dimension in an output sensitive way. Using this data structure, one can trace a connected submanifold of $\mathbb{R}^d$, starting from a given initial point on the manifold (Section 3). Our algorithm produces a PL-approximation of size polynomial in d and $\delta = 1/D$, and exponential in n. The complexity of the algorithm is also polynomial in d, and δ, and exponential in n.

Moreover, by taking δ large enough, the PL-approximation output by the algorithm is a faithful approximation of the isomanifold. Specifically, as shown in the full version of [13] and recalled in Section 2.2, if we take $\delta = \Omega(d^{2.5})$, the PL-approximation $\hat{\mathcal{M}}$ is $O(D^2)$-close and isotopic to the isomanifold. Here the constants in the O depend on f and its derivatives. Hence, our algorithm constructs geometrically close and topologically correct PL-approximation of isomanifolds of bounded complexity in polynomial time.

Our algorithm can be extended in several directions. First, the dependency on d in the size of $\hat{\mathcal{M}}$ can be completely removed by combining our algorithm with dimensionality reduction (Section 3.4). We can also extend the algorithm to the case of isomanifolds with boundary and, more generally, to stratifolds (Section 3.5).

The algorithm has been implemented. In Section 4, we report on experimental results which show that the algorithm is practical and can handle cases that are far 16 ahead of the state-of-the-art. We also present an application in Algebraic Geometry that was used to verify a conjecture on projective varieties defined by polynomial equations in the complex projective plane. Following numerous experiments on various projective varieties, the conjecture was ultimately proved by Alvarez and Deroin [4].

The approximation of a manifold that is the zero set of a function is an example of the more general question of how to triangulate a manifold which has a long history in Mathematics. In particular, Whitney [45] introduced a construction that has some similarity with the present algorithm (see [10]). A major difference though is that topological guarantees can only be obtained if some intricate perturbations of the ambient triangulation are performed (Section 5). These techniques are at the moment incompatible with polynomial complexity.

2 Background

2.1 Permutahedral representation of CFK-triangulations

In this section, we give the most important definitions and basic properties of Coxeter and Freudenthal-Kuhn triangulations. An extensive discussion can be found in Appendix A.1.

Both Coxeter and Freudenthal-Kuhn triangulations can be described as an arrangement of hyperplanes. They are related by an affine transformation. Let E be a finite set of vectors of $\mathbb{R}^d$ and consider the set of hyperplanes $H_E = \{x \in \mathbb{R}^d \mid \langle x, u\rangle = k, u \in E, k \in \mathbb{Z}\}$. Let, in addition, H be the hyperplane of $\mathbb{R}^{d+1}$ of equation $\langle x, \mathbf{1}\rangle = 0$ where $\mathbf{1}$ is the vector of $\mathbb{R}^{d+1}$ whose coordinates are all 1.

▶ **Definition 1.** *The Freudenthal-Kuhn triangulation is the hyperplane arrangement $\mathcal{H}_{E_{FK}}$ associated to the set of vectors $E_{FK} = \{e_1, \ldots, e_d\} \cup \{u_{i,j} = e_j - e_i \mid 1 \leq i < j \leq d\}$. The Coxeter triangulation of type $\tilde{A}_d$ is the hyperplane arrangement $\mathcal{H}_{E_C}$ in $\mathbb{R}^{d+1}$ associated to the set of vectors $E_C = \{r_{i,j} = e_i - e_{j+1} \mid 1 \leq i \leq j \leq d\}$, restricted to $H \simeq \mathbb{R}^d$.*

Two important facts follows. On one hand, because Coxeter and Freudenthal-Kuhn triangulations are related by an affine transformation, they have the same combinatorial structure. We call any triangulation that is the image of a Freudenthal-Kuhn triangulation under an affine transformation a CFK-triangulation. In this paper, we restrict our attention to Coxeter and Freudenthal triangulations since they are the simplest, but any CFK-triangulation could be used. The second fact is that each simplex in such a triangulation can be represented as a cell in an arrangement of $d(d-1)/2$ families of parallel hyperplanes which are known and do not need to be stored.

The next crucial observation relates CFK-triangulations and permutahedra, which allows to represent CFK-triangulations in a compact way (The definition and some combinatorial

properties of permutahedra are given in Appendix A.1.1). We first recall that two complexes are dual if there is a bijection between their faces that inverses the inclusion relationships.

▶ **Proposition 2.** *The star of a vertex in a CFK-triangulation is combinatorially dual to a permutahedron.*

Since the facial structure of a permutahedron is fully described by ordered partitions, any simplex σ in a CFK-triangulation is characterized by a star that contains σ and by the ordered partition that specifies which simplex in the star is precisely σ. Since a simplex appears in several stars, we take the one that is centered at the lowest vertex of σ in the lexicographic order. This representation is called the permutahedral representation of a CFK-triangulation, see Figure 1. We further have:

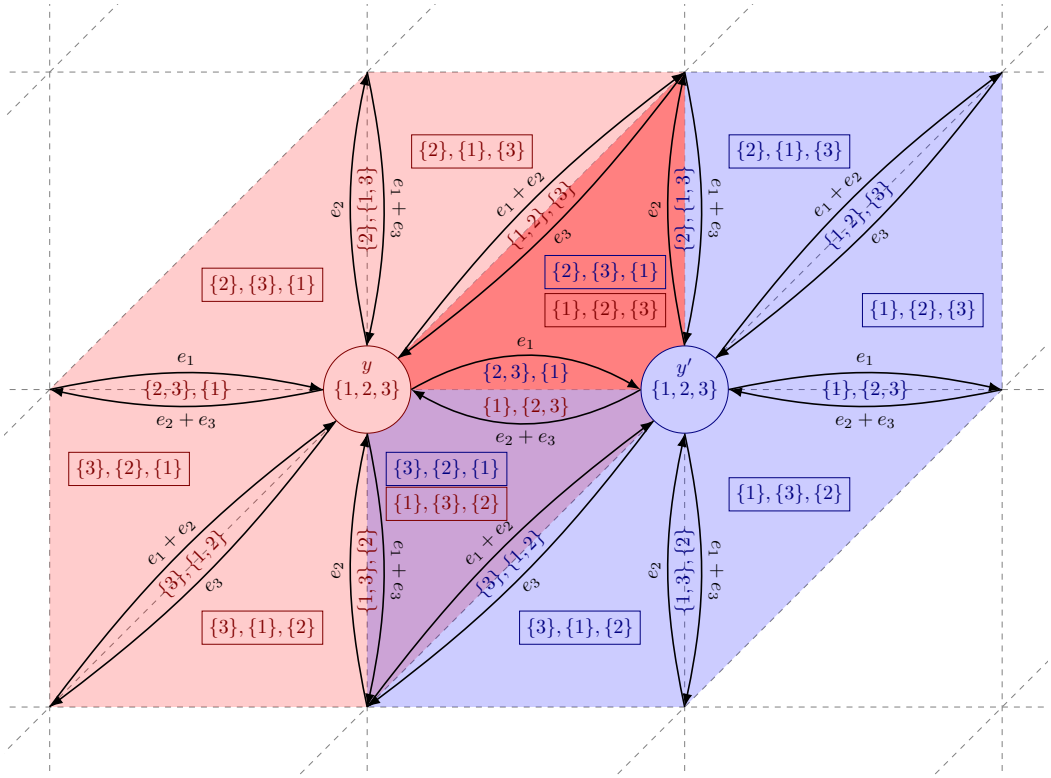

Figure 1 The permutahedral representation of the simplices in the stars of vertices y and y'.

▶ **Lemma 3** (Face computation). *Let σ be an l-simplex in the FK-triangulation of $\mathbb{R}^d$. Computing all its k-faces can be done in time $O(ds)$, where $s = \binom{l+1}{k+1}$ is the number of k-faces of an l simplex. The space complexity of the algorithm is $O(l)$.*

▶ **Lemma 4** (Coface computation). *Let σ be a k-simplex in the FK-triangulation of $\mathbb{R}^d$ given by its permutahedral representation. Computing the permutahedral representations of all its l-cofaces can be done in time $O(ds)$, where $s \leq \frac{1}{2^{\min(l,d-l)}} \binom{d-k}{d-l} (d-k+1)!$ is the number of l-cofaces of a k-simplex in the FK-triangulation. The space complexity of the algorithm is $O(d)$.*

2.2 PL-approximation of isomanifolds

We first recall sufficient conditions under which the PL-approximation $\hat{\mathcal{M}}$ output by the algorithm faithfully reproduces the original isomanifold. These conditions are fully described in the full version of [13] and we simply state here the main results specialized to the case of CFK-triangulations.

We will say that f has *bounded complexity* if the three following quantities $\gamma_{\max}$, $\lambda_{\min}$ and $\alpha_{\max}$ are positive and bounded.

$$\gamma_{\max} = \max_{x\in\mathcal{T}_0} (\max_i |\mathrm{grad} f^i(x)|) \quad \lambda_{\min} = \min_{x\in\mathcal{T}_0} \lambda_{\min}(x), \quad \alpha_{\max} = \max_{x\in\mathcal{T}_0} \max_i \|\mathrm{Hes}(f^i)(x)\|_2$$

where

- $\mathcal{T}_0$ denotes the set of all $\sigma \in \mathcal{T}$, such that $(f^i)^{-1}(0) \cap \sigma \neq \emptyset$ for all i.
- $\mathrm{grad} f^i = (\partial_j f_i)_j$ denotes the gradient of component f^i, for $i \in [1, d-n]$,
- $\mathrm{Gram}(\nabla f)$ denotes the Gram matrix whose elements are $\nabla f^i \cdot \nabla f^j$ where $\cdot$ stands for the dot product.
- $\lambda_{\min}(x)$ denotes the smallest absolute value of the eigenvalues of $\mathrm{Gram}(\nabla f(x))$,[1]
- $\mathrm{Hes}(f) = (\partial_k \partial_l f_i)_{k,l}$ denotes the Hessian matrix of second order derivatives,
- $|\cdot|$ denotes the Euclidean norm of a vector and $\|\cdot\|_2$ the operator 2-norm of a matrix.[2]

We can now restate the topological result of [13] :

▶ **Theorem 5.** *Assume that the function f has bounded complexity. If the precision of the CFK-triangulation satisfies $D = O(d^{-5/2})$, where the constant in the big O depends on $\gamma_{\max}$, $\lambda_{\min}$ and $\alpha_{\max}$, then $\hat{\mathcal{M}}$ is a manifold isotopic to the zero set $\mathcal{M}$ of f.*

Moreover, we can bound the Fréchet distance between $\mathcal{M}$ and $\hat{\mathcal{M}}$. The Fréchet distance is a quite strong notion of distance and, in particular, it bounds the Hausdorff distance.

▶ **Definition 6** (Fréchet distance for embedded manifolds)**.** *Let $\mathcal{M}_a$ and $\mathcal{M}_b$ be two homeomorphic, compact submanifolds of $\mathbb{R}^d$. Write $\mathcal{H}$ for the set of all homeomorphisms from $\mathcal{M}_a$ to $\mathcal{M}_b$. The Fréchet distance between $\mathcal{M}_a$ and $\mathcal{M}_b$ is $d_F(\mathcal{M}_a, \mathcal{M}_b) = \inf_{h\in\mathcal{H}} \sup_{x\in\mathcal{M}_a} d(x, h(x))$.*

▶ **Theorem 7.** *Assume that the function f has bounded complexity. Then, $d_F(\mathcal{M}, \hat{\mathcal{M}}) = O(D^2)$ where the constant in the big O depends on $\gamma_{\max}$, $\lambda_{\min}$ and $\alpha_{\max}$.*

3 Tracing isomanifolds

In this section, we describe an algorithm that computes a PL-approximation $\hat{\mathcal{M}}$ of an isomanifold $\mathcal{M}$. The algorithm has some similarity with the Marching Cube algorithm [33] but departs from it in two fundamental ways. First, because of the curse of dimensionality, we cannot afford to look at all the cells in the grid and need to limit the search to cells that are close to $\mathcal{M}$. The problem of computing $\hat{\mathcal{M}}$ can be naturally decomposed into two subproblems: locating the various components of $\mathcal{M}$ (i.e., finding at least one point in each connected component), and then tracing around each component, using the fact that the components are connected. This decomposition is used by various authors, see for example [44, 23]. In this paper, we focus on the tracing problem, although we discuss very briefly

[1] Because a Gram matrix is a symmetric square matrix, its eigenvalues are well defined and real.

[2] The operator norm is defined as $\|A\|_p = \max_{x\in\mathbb{R}^n} \frac{|Ax|_p}{|x|_p}$, with $|\cdot|_p$ the p-norm on $\mathbb{R}^n$.

(Section 3.2) the problem of locating the components. As pointed out by Dobkin et al. many applications supply their own starting points.

The second major difference with the original marching cube algorithm is to replace the usual cubical grid by a CFK-triangulation of the ambient space. Taking a CFK-triangulation instead of a grid is a major advantage in high dimensions that has been recognized in the pioneering works of Allgower, and Schmidt [3] and of Dobkin et al. [23], see also [35]. The novelty here is to use the data structure of Section 2.1 to represent a CFK-triangulation. As a consequence, we will keep two main advantages of using grids: very limited storage and fast basic operations.

3.1 Isomanifolds

Let $f : \mathbb{R}^d \to \mathbb{R}^{d-n}$ be a smooth (C^2 suffices) function, and suppose that 0 is a regular value of f, meaning that at every point x such that $f(x) = 0$, the Jacobian of f is non-degenerate. Then the zero set of f is an n-dimensional manifold as a direct consequence of the implicit function theorem, see for example [24, Section 3.5]. We further assume that $f^{-1}(0)$ is compact. As in [1] we consider a triangulation $\mathcal{T}$ of $\mathbb{R}^d$. The function $\hat{f}$ is the linear interpolation of the values of f at the vertices if restricted to a single simplex $\sigma \in \mathcal{T}$, i.e.

$$\forall x \in \sigma : \quad \hat{f}(x) = \sum_{v \in \sigma} \lambda_v(x) f(v), \tag{1}$$

where the λ_v are the barycentric coordinates of x with respect to the vertices v of σ. For any function $g : \mathbb{R}^d \to \mathbb{R}^{d-n}$ we write g^i, with $i = 1, \ldots, d-n$, for the components of g.

The PL-approximation is now defined as $\hat{f}^{-1}(0) = \hat{\mathcal{M}}$. Locally, $\hat{f}|_\sigma^{-1}(0)$ is generically the intersection of an n-flat H_σ with σ. More precisely we note that $\hat{f}|_\sigma^{-1}(0)$ is an n-flat if the gradients of $\hat{f}^i|_\sigma$ are linearly independent, which can be easily achieved by perturbing f infinitesimally (or at least its values at the vertices). Let τ_j^{d-n} and τ_j^{d-n-1} be faces of σ of dimension $d-n$ and $d-n-1$. An infinitesimal perturbation of f, can prevent either $\hat{f}|_\sigma^{-1}(0)$ from intersecting the faces τ_j^{d-n-1}, or the gradients of $\hat{f}^i|_\sigma$ and the normal spaces of τ_j^{d-n} (for each fixed j) from failing to span $\mathbb{R}^d$. More precise statements on the geometric and topological stability of the triangulation under perturbations of f can be found in the full version of [13, Section 5]. Because $\hat{f}|_\sigma^{-1}(0)$ is (generically) the intersection of an n-flat (H_σ) and σ, it is an n-dimensional polytope denoted by C_σ. The PL-approximation or mesh $\hat{\mathcal{M}}$ of $\mathcal{M}$ is the polytopal cell complex obtained by gluing the polytopes C_σ associated to all the simplices σ in $\mathcal{T}$.

3.2 Manifold tracing algorithm

Let $\mathcal{M}$ be the zero set of some function $f : \mathbb{R}^d \to \mathbb{R}^{d-n}$, and let $\hat{\mathcal{M}}$ be the associated PL-approximation defined over a triangulation $\mathcal{T}$ of the ambient space $\mathbb{R}^d$. Both n, and d are known but arbitrary, and will be considered as parameters in the complexity analysis. We write $k = d - n$ for the codimension of $\mathcal{M}$. The algorithm will use for $\mathcal{T}$ a CFK-triangulation stored using the data structure from Section A.1. We assume that the manifold $\hat{\mathcal{M}}$, and the triangulation $\mathcal{T}$ satisfy the following genericity hypothesis:

▶ **Hypothesis 8** (Genericity). *Let σ be a d-simplex of $\mathcal{T}$ that intersects H_σ. No subface of σ of dimension less than k intersects H_σ, and any subface of σ of dimension k intersects H_σ in at most one point and transversally.*

We note that this condition can be satisfied by an infinitesimal perturbation for isomanifolds. This requires some explanation. We recall that the CFK-triangulation is a hyperplane arrangement, and up to translation there are a finite number of k-flats that contain all k-simplices in the CFK triangulation. Hypothesis 8 is not satisfied, if either the flat H_σ is not linearly independent of these k-flats, or if H_σ does intersect some $(k-1)$-flat in the CFK-triangulation. In the previous section, we have already seen that an infinitesimal perturbation ensures that H_σ is n-dimensional. Because two affine spaces whose dimensions do not add up to the ambient dimension don't intersect with generically and two affine spaces whose dimensions add up to exactly the ambient dimension intersect in a single point, we see that genericity can be achieved by perturbing f infinitesimally. We further remark that, generically, any vertex of the PL-approximation $\hat{\mathcal{M}}$ is the intersection point between a k-simplex σ of $\mathcal{T}$ with the n-flat H_σ that interpolates f inside σ.

Algorithm 1 Manifold tracing algorithm

input : the permutahedral representation of a triangulation $\mathcal{T}$ of $\mathbb{R}^d$,
the codimension of the isomanifold $k = d - n$,
a seed k-simplex τ_0 that intersects $\hat{\mathcal{M}}$
oracle : Given a k-simplex σ of $\mathcal{T}$, decide whether σ intersects H_σ and, in the affirmative, report the corresponding vertex $\sigma \cap H_\sigma = \sigma \cap \hat{\mathcal{M}}$.
output : Set $\mathcal{S}$ of the simplices in $\mathcal{T}$ of dimension k that intersect $\hat{\mathcal{M}}$, represented by their permutahedral representation, and the corresponding set $\hat{\mathcal{M}}_0$ of intersection points
Initialize the queue $\mathcal{Q}$ and the set $\mathcal{S}$ with τ_0
while *the queue $\mathcal{Q}$ is not empty* **do**
  Pop a k-dimensional simplex τ from $\mathcal{Q}$
  foreach *cofacet ϕ of τ* **do**
    foreach *facet σ of ϕ* **do**
      if *σ does not lie in $\mathcal{S}$ and intersects $\hat{\mathcal{M}}$ (which can be decided using the oracle)* **then**
        Insert σ into the queue $\mathcal{Q}$
        Insert σ into $\mathcal{S}$ together with the intersection point provided by the oracle

The algorithm essentially computes the set $\mathcal{S}$ of k-simplices of $\mathcal{T}$ that intersect $\hat{\mathcal{M}}$. The elements of $\mathcal{S}$ are in 1-1 correspondence with the vertices of $\hat{\mathcal{M}}$ thanks to the Genericity hypothesis. The so-called intersection oracle is a basic ingredient of the algorithm:
Intersection oracle: *Given a k-simplex σ of $\mathcal{T}$, decide whether σ intersects H_σ and, in the affirmative, report the corresponding vertex $\sigma \cap H_\sigma$.*
It is easy to see that the intersection oracle reduces to solving a linear system. Indeed, generically, a vertex is the intersection of a k-simplex σ of $\mathcal{T}$ with the m-flat H_σ that interpolates f inside σ. One can compute the barycentric coordinates of $\sigma \cap H_\sigma$ by solving a linear system of k equations, and k unknowns. It then remains to check whether the barycentric coordinates are all non-negative (to ensure that the intersection point lies inside σ). It follows that the intersection oracle reduces to evaluating f at the $k+1$ vertices of σ plus solving a $k \times k$ linear system.
In addition, we need to provide a set of k-simplices of $\mathcal{T}$ to initialize the tracing. These simplices must intersect all the connected components of the isomanifold and are called

seed simplices. If $\mathcal{M}$ consists of multiple connected components, then a seed simplex must be provided per each connected component and we proceed in the same manner for each component. So we will assume for now that $\mathcal{M}$ is connected.

The seed simplices are given as part of the input and we don't discuss in this paper the problem of their construction. We simply observe that they can be obtained by computing a critical point (e.g., a point with smallest x_1-coordinate) on each connected component of the isomanifold, which reduces to finding a solution to a system of equations, on which a large body of literature exists. See for example [38, 37, 23] and also the discussion in Wenger's book [44, Section 8.4]. Once such a seed point has been computed, we simply translate and rotate the triangulation $\mathcal{T}$ so that the seed point coincides with the barycenter of a k-simplex of $\mathcal{T}$ and the intersection with the manifold is transversal as demanded by the genericity hypothesis (for numerical stability it is convenient if the angle between the tangent space of the manifold and the starting k-simplex is large, which is easy to ensure). If the distance between $\mathcal{M}$ and $\hat{\mathcal{M}}$ is small enough, then $\hat{\mathcal{M}}$ also intersects the same k simplex, see Section 2.2.

The algorithm is described as Algorithm 1. It takes as input the permutahedral representation of an ambient FKC-triangulation $\mathcal{T}$ and a seed k-simplex τ_0 of $\mathcal{T}$. We assume that $\mathcal{T}$ satisfies the Genericity Hypothesis 8, which can be enforced by infinitesimal perturbations of f as discussed in Section 3.1.

The algorithm maintains the subset $\mathcal{S}$ of the simplices in $\mathcal{T}$ of dimension k that intersect $\hat{\mathcal{M}}$. $\mathcal{S}$ is initialized with the seed simplex τ_0 and stored as a hash table so that we can decide in constant time if a given k-simplex belongs to $\mathcal{S}$. Then, starting from τ_0, we look at all its cofacets and consider all the facets of those cofacets that are not in $\mathcal{S}$ (i.e. they have not been considered yet). This can be done using a queue $\mathcal{Q}$ of candidate k-simplices. Each of these simplices is queried with the intersection oracle and, if it is found to intersect $\hat{\mathcal{M}}$, it is added to $\mathcal{S}$ if not already present. Upon termination, $\mathcal{S}$ contains all the k-dimensional simplices of $\mathcal{T}$ that intersect $\hat{\mathcal{M}}$. Each such intersection, which consists of a single point (by the Genericity hypothesis), is a vertex of $\hat{\mathcal{M}}$. Hence $\hat{\mathcal{M}}_0$ is the vertex set of $\hat{\mathcal{M}}$.

Note that our algorithm essentially traverses the adjacency graph of the k and $(k+1)$-simplices of $\mathcal{T}$ that intersect $\hat{\mathcal{M}}$. It therefore identifies not only the set $\hat{\mathcal{M}}_0$ of vertices of $\hat{\mathcal{M}}$, but also the edges joining two such vertices (associated to the cofacets of the k-simplices in $\mathcal{S}$). By simply reporting those cofacets on the fly, the algorithm can output the 1-skeleton $\hat{\mathcal{M}}_1$ of the n-dimensional polytopal cell complex $\hat{\mathcal{M}}$. The higher dimensional faces of $\hat{\mathcal{M}}$ are the polytopes $C_\tau = \tau \cap H_\tau$ for all the cofaces τ of the k-simplices of $\mathcal{S}$. If needed, the full Hasse diagram of $\hat{\mathcal{M}}$ can be computed from $\hat{\mathcal{M}}_0$. This can be done in an output sensitive manner by using the permutahedral representation of $\mathcal{T}$ and the algorithm of Section A.1 to compute cofaces by increasing dimensions.

3.3 Complexity analysis

We can easily bound the complexity of the manifold tracing algorithm as a function of the size of the output.

▶ **Proposition 9.** *The time complexity of the algorithm is $O(k2^n I|\mathcal{S}|)$ where I is the time complexity of one call of the intersection oracle, and $|\mathcal{S}|$ is the number of simplices of dimension k output by the algorithm.*

Since, the intersection oracle reduces to evaluating f at the $k+1$ vertices of σ plus solving a $k \times k$ linear system, $I = O(k^\omega)$ where $\omega \approx 2.375$.

We will now express the size of the output in terms of quantities that depend on the manifold, the ambient dimension d, and the resolution of the triangulation (the diameter D of a simplex) which bounds the density of the output sample, and the precision of the approximation. Our result holds for K-sparse manifolds, i.e. submanifolds whose intersection with any k-flat consists of at most K points. In practical situations, K is usually small and, in particular, K is a constant for algebraic isomanifolds of bounded degree.

▶ **Proposition 10** (Size of the output). *Assume that $\mathcal{M}$ is contained in the unit cube $C_d = [0,1]^d$, and that any k-flat intersects $\mathcal{M}$ at most K times. Writing $|\mathcal{S}| = N_C$ when $\mathcal{T}$ is a Coxeter triangulation and $|\mathcal{S}| = N_{FK}$ for a Freudenthal triangulation, we have $N_C \leq \frac{K}{n!} \times \left(\frac{d^2\sqrt{d(d+2)}}{2\sqrt{2}D}\right)^n$ and $N_{FK} \leq \frac{K}{n!} \times \left(\frac{d^3}{\sqrt{2}D}\right)^n$ where D is the diameter of a simplex of $\mathcal{T}$.*

We see that Coxeter triangulations lead to smaller samples than FK-triangulations by a factor of roughly 2^n. This will be confirmed experimentally (see Figure 3).
As noticed in Section 3.2, a simple variant of the algorithm can compute the full Hasse diagram of $\hat{\mathcal{M}}$ in an output sensitive manner. The following lemma shows that the combinatorial complexity of $\hat{\mathcal{M}}$ is of the same order as the combinatorial complexity as $\hat{\mathcal{M}}_0$.

▶ **Proposition 11.** *The combinatorial complexity of $\hat{\mathcal{M}}$ is $|\mathcal{S}| \times (\frac{3}{2})^n (n+1)!$, where $|\mathcal{S}|$ is bounded in Proposition 10. If $n = O(1)$, the combinatorial complexity of $\hat{\mathcal{M}}$ is polynomial in d, and $\delta = 1/D$.*

We combine Propositions 9, 10, and 11 to obtain our main result.

▶ **Theorem 12.** *Assume that $\mathcal{M}$ is contained in the unit cube $[0,1]^d$ and that any affine k-flat intersects $\mathcal{M}$ at most K times (K is usually small, and is in particular a constant for algebraic isomanifolds of bounded degree). Let, in addition, D be the precision required on the approximation (the diameter of a simplex in the ambient triangulation $\mathcal{T}$). The size of the output, and the time complexity of the algorithm are polynomial in the ambient dimension d, and in $\delta = 1/D$, and exponential in the intrinsic dimension n. The same result holds for the full PL-approximation $\hat{\mathcal{M}}$ of $\mathcal{M}$.*

3.4 Dimensionality reduction

As seen from Proposition 10, the size $|\mathcal{S}|$ of the output of the algorithm, considered as a function of the resolution D of the triangulation, depends exponentially on n (which is to be expected), and only polynomially on d (which is fortunate). Nevertheless, the computing time of our algorithm and the size of the output depend on d. Removing the dependency on d in the time complexity is impossible since we need to evaluate a vector-valued function f at a number of points of $\mathbb{R}^d$, which takes $\Omega(d)$ time per evaluation. However, we will see that we can reduce the size of the mesh produced by our algorithm.
Examples of samples of $\mathcal{M}$ whose sizes depend on n but not on d, and lead to good approximations are known. Especially important are D-nets [17, 9]. A D-net consists of a finite number of sample points of $\mathcal{M}$ such that no point of $\mathcal{M}$ is at distance more than D from a sample point (density condition), and no two sample points are closer than cD for some positive constant c (separation condition). A simple volume argument shows that the size of a D-net of a n-dimensional smooth submanifolds is $O(1/D^n)$ [8, Lemma 5.3]. The sample produced by our algorithm is D-dense on the piecewise linear approximation. This implies that we have a sample that has a Hausdorff distance of $D + d_F(\mathcal{M}, \hat{\mathcal{M}})$ to the manifold, where $d_F(\mathcal{M}, \hat{\mathcal{M}})$ is bounded in Theorem 7.

Since its cardinality depends on d, it is not well separated and, in particular, not a D-net of $\mathcal{M}$. If we are mostly interested in the output sample, we can easily sparsify it to obtain a D-net. However, by doing so, we will lose the combinatorial structure of the mesh.
We now show how to compute a D-dense sample of $\mathcal{M}$ of size independent of d, together with a mesh. Specifically, we will reduce dimensionality using a variant of the celebrated Johnson-Lindenstrauss lemma for manifolds. Doing so, we depart from our previous worst-case analysis by allowing some approximation factor ε and tolerate a guarantee that holds only with high probability.

▶ **Theorem 13** (Johnson-Lindenstrauss lemma for manifolds [20, 42]). *Pick any $\varepsilon, \eta > 0$, and let $d' = \Omega\left(\frac{n}{\varepsilon^2}\log\frac{1}{\varepsilon} + \frac{1}{\varepsilon^2}\log\frac{\Gamma}{\eta}\right)$, where Γ is a quantity that depends only on intrinsic properties of $\mathcal{M}$. Let Φ be the projection on a random affine subspace of dimension d'. Then, with probability $> 1-\eta$, for all $x, y \in \mathcal{M}$, we have $(1-\varepsilon)\sqrt{\frac{d'}{d}} \leq \frac{\|\Phi x - \Phi y\|}{\|x-y\|} \leq (1+\varepsilon)\sqrt{\frac{d'}{d}}$.*

Let $\Psi = \sqrt{\frac{d}{d'}}\,\Phi$. By the theorem, the image $\Psi(\mathcal{M})$ of $\mathcal{M}$ is a submanifold of dimension n embedded in $\mathbb{R}^{d'}$. One can now run the manifold tracing algorithm in $\mathbb{R}^{d'}$ to sample, and mesh $\Psi(\mathcal{M})$. The algorithm works as described before except that we need another oracle that, given a $(d'-n)$-simplex σ of the CFK-triangulation of $\mathbb{R}^{d'}$, decides whether its inverse image $\Psi^{-1}(\sigma)$ intersects $\mathcal{M}$ or not. Note that $\Psi^{-1}(\sigma)$ is a $(d-d')$-dimensional flat strip (that is the product of a face and an affine subspace) in $\mathbb{R}^d$, and that the complexity of this new oracle is the same as the complexity of the basic intersection oracle, i.e. polynomial in d. Due to the scaling factor $\sqrt{d/d'}$, the resolution of the triangulation in the low dimensional space $\mathbb{R}^{d'}$ has to be scaled by the same factor if one wants to satisfy a given sampling density on $\mathcal{M}$. Since the geometry of the manifold is also scaled in the same way [26], the analysis of the algorithm will be unchanged. Proposition 10 then shows that the size of the output sample does not depend on d but only on n and D for fixed ε, and η. Moreover, since the complexities of the projection and of the new oracle are polynomial in d, Proposition 9 implies that the overall complexity is still polynomial in d.

3.5 Isomanifolds with boundary, and stratifolds

The case of isomanifolds with boundary and, more generally, of isostratifolds can be handled in very much the same way. By an isomanifold of dimension n with boundary, we mean that, on top of a function $f : \mathbb{R}^d \to \mathbb{R}^{d-n}$, we are given another function $f_\partial : \mathbb{R}^d \to \mathbb{R}$, and the set we consider is $\mathcal{M} = f^{-1}(0) \cap f_\partial^{-1}([0,\infty))$. We note that $\partial\mathcal{M} = f^{-1}(0) \cap f_\partial^{-1}(0)$.
Similarly to (1), we also define $\hat{f}_\partial|_\tau(x) = \sum_{v\in\sigma} \lambda_v(x)\, f_\partial(v)$. We write $\hat{f}$ for the (global) piecewise linear function that coincides with $\hat{f}|_\tau$ on each τ of $\mathcal{T}$, and $\hat{f}_\partial$ for the (global) piecewise linear function that coincides with $\hat{f}_\partial|_\tau$ on each τ of $\mathcal{T}$. We note that the piecewise linear approximation of the boundary $\partial\hat{\mathcal{M}} = \hat{f}_\partial^{-1}(0) \cap \hat{f}^{-1}(0)$ is a subset of $\hat{f}^{-1}(0)$, i.e. the piecewise linear approximation of the manifold ignoring the boundary. The piecewise linear approximation $\hat{\mathcal{M}}$ of the manifold with boundary consists of the following cells:

- For each τ of $\mathcal{T}$, such that $\hat{f}_\partial|_\tau$ is positive on τ, and $(\hat{f}|_\tau)^{-1}(0) \cap \tau \neq \emptyset$, we add $(\hat{f}|_\tau)^{-1}(0) \cap \tau$.
- For each τ of $\mathcal{T}$, such that $(\hat{f}|_\tau)^{-1}(0) \cap \tau \neq \emptyset$, and $(\hat{f}_\partial|_\tau)^{-1}(0) \cap \tau \neq \emptyset$, we add $(\hat{f}|_\tau)^{-1}(0) \cap (\hat{f}_\partial|_\tau)^{-1}([0,\infty)) \cap \tau$.

We will assume that the Genericity Hypothesis 8 holds for both $\hat{\mathcal{M}}$, and $\partial\hat{\mathcal{M}}$.
We can now adapt the algorithm of Section 3.2 as follows. In addition to reporting the set S_k of k-faces of the triangulation $\mathcal{T}$ that intersect $\hat{\mathcal{M}}$, the algorithm will also report the set

S_{k+1} of $(k+1)$-faces of the triangulation $\mathcal{T}$ that intersect $\partial\hat{\mathcal{M}}$. The computation of S_{k+1} is done by the following simple modification of Algorithm 1: if the k-dimensional facet σ of τ intersects $\hat{f}^{-1}(0)$ at a point x such that $\hat{f}_\partial|_\tau(x) < 0$ (i.e. x is not in $\hat{\mathcal{M}}$), we then compute the intersection point of τ with $\hat{f}_\partial^{-1}(0)$, and put τ in S_{k+1}.

As for the case of manifolds without boundary (see the discussion at the end of Section 3.2), the algorithm traverses (and therefore computes) the 1-skeleton of $\hat{\mathcal{M}}$. Under the Genericity Hypothesis 8, the vertices of $\hat{\mathcal{M}}_1$ are in bijection with the simplices of $S_k \cup S_{k+1}$. The edges are obtained by applying the following rules below (we identify a simplex in S_k (resp. S_{k+1}) and the intersection point $S_k \cap \hat{\mathcal{M}}$ (resp. $S_{k+1} \cap \partial\hat{\mathcal{M}}$):

1. Two simplices σ_1, and σ_2 of S_k are joined by an edge in $\hat{\mathcal{M}}_1$ if and only if there exists a simplex in $\mathcal{T}_{k+1}$ with faces σ_1 and σ_2.
2. Two simplices τ_1, and τ_2 of S_{k+1} are joined by an edge in $\widehat{\partial\mathcal{M}}_1$ if and only if there exists a simplex in $\mathcal{T}_{k+2}$ with faces τ_1 and τ_2.
3. A simplex σ of S_k, and a simplex τ of S_{k+1} are joined by an edge in $\widehat{\partial\mathcal{M}}_1$ if and only if σ is a facet of τ.

The three rules above together with the permutahedral representation of $\mathcal{T}$ provide a way to construct the 1-skeleton of $\hat{\mathcal{M}}$ on the fly. The total cost is output sensitive. If needed, the entire combinatorial structure of $\hat{\mathcal{M}}$ can be computed by traversing the full triangulation $\mathcal{T}$. The above construction generalizes easily to arbitrary isostratifolds. Isostratifolds are stratifolds that are defined by equations and inequalities. An example of such a stratifold is an octant of the sphere in $\mathbb{R}^3$ that can be defined by as $x^2+y^2+z^2-1=0$, $x \geq 0$, $y \geq 0$, and $z \geq 0$. We compute the 1-skeleton of $\hat{\mathcal{M}}$ and construct a graph whose nodes are the simplices of dimensions $k, k+1, ..., d$ that intersect the strata of dimension $n, n-1, ..., 0$.

4 Experimental results

The data structure of Section A.1 and the algorithm of Section 3 have been implemented in C++. The code is robust and fast and will be released in the GUDHI library [30]. Full detail on the implementation, including the implementation of the oracle, can be found in [32].

In this section, we explore the dependency of our C++ implementation of the data structure for the ambient CFK-triangulation, and of the manifold tracing algorithm on the properties of the triangulation, and of the input manifold.

4.1 Performance of the algorithm

We show the performance of our implementation of the manifold tracing algorithm for various ambient and intrinsic dimensions in Figure 2. In Figure 3, we can see that using Coxeter triangulation is beneficial in practice as it produces a smaller output in less time (see Proposition 10).

In Figure 4, we present a PL approximation of a two-dimensional Clifford torus without boundary embedded in $\mathbb{R}^{10}$ built by the manifold tracing algorithm. The torus has been rotated and translated in $\mathbb{R}^{10}$ so that the coordinate axes do not play any special role. Note that there is no C^2 isometric embedding of the Clifford torus in $\mathbb{R}^3$.

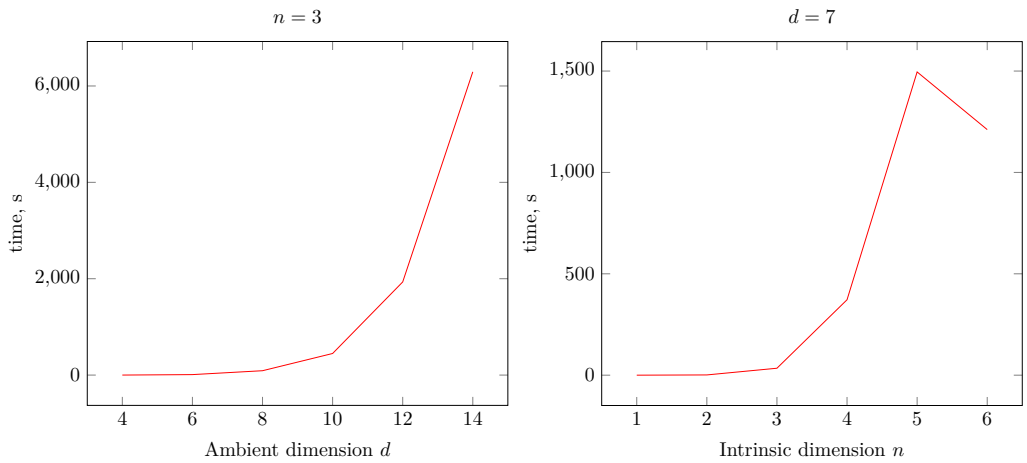

Figure 2 The effect of the ambient dimension d and of the intrinsic dimension n on the computation time of the manifold tracing algorithm. The reconstructed manifold in the tests is the n-dimensional sphere embedded in $\mathbb{R}^d$. The ambient triangulation used is a Coxeter triangulation of type $\tilde{A}_d$. The diameter of the full simplices is fixed for all d.

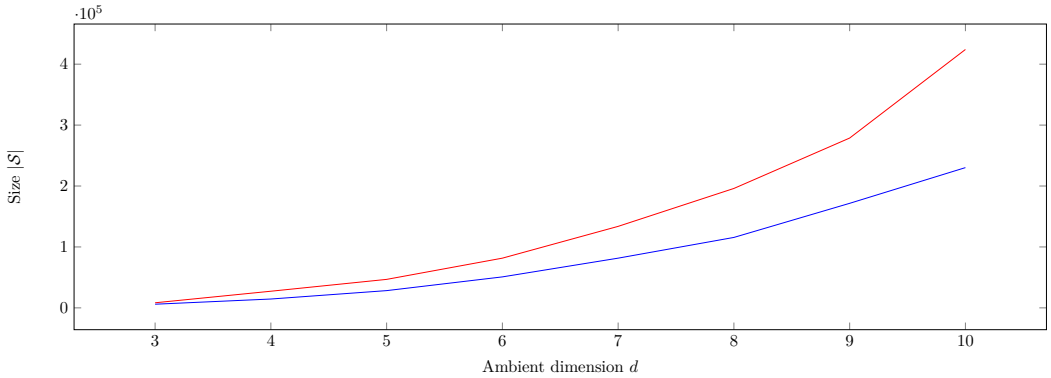

Figure 3 Comparison of the size of the output of the manifold tracing algorithm using two types of ambient triangulations: a Coxeter triangulation of type $\tilde{A}_d$ (in blue), and the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ (in red) with the same diameter $0.07\sqrt{d}$ of d-dimensional simplices. The reconstructed manifold is the 2-dimensional implicit surface "Chair" embedded in $\mathbb{R}^d$ given by the equations: $(x_1^2 + x_2^2 + x_3^2 - 0.8)^2 - 0.4\left((x_3 - 1)^2 - 2x_1^2\right)\left((x_3 + 1)^2 - 2x_2^2\right) = 0$, and $x_i = 0$ for $i > 3$.

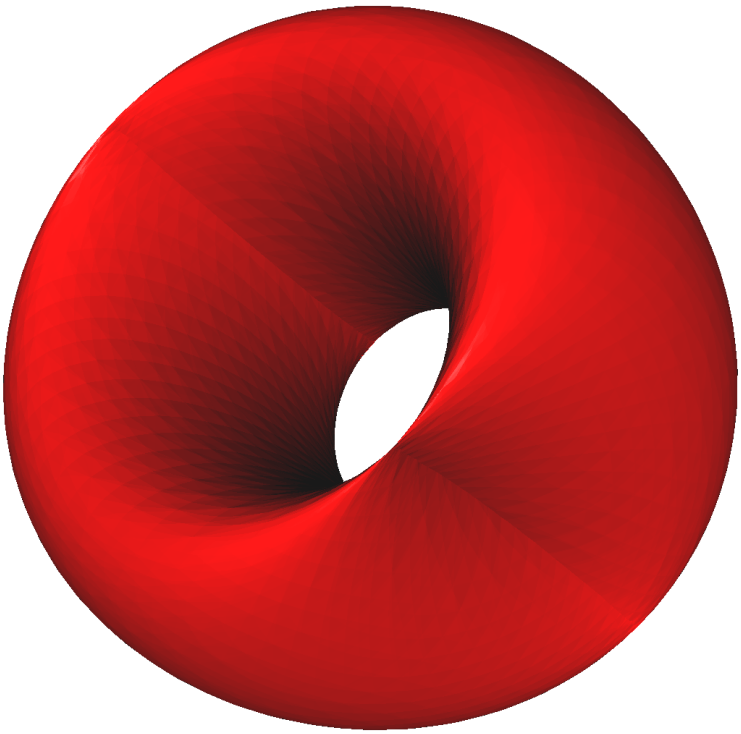

Figure 4 The piecewise-linear approximation of a flat torus embedded in $\mathbb{R}^{10}$ defined by the equations $x_1^2 + x_2^2 = 1$, and $x_3^2 + x_4^2 = 1$, and $x_i = 0$ for $i > 4$, projected to $\mathbb{R}^3$. The ambient triangulation used is a Coxeter triangulation of type $\tilde{A}_{10}$ with the diameter of the full-dimensional simplices 0.23. The output size $|\mathcal{S}|$ is 509 952. The execution time of the algorithm is 231s. The torus has been rotated and translated in $\mathbb{R}^{10}$ so that the coordinate axes do not play any special role.

4.2 Manifolds with boundary

The algorithm has been adapted to handle submanifolds with boundary and surfaces with a piecewise smooth boundary, see Section 3.5. In Figure 5, we present the mesh obtained by our algorithm on a portion of a flat torus embedded in $\mathbb{R}^4$, and cut by a hypersphere. The torus has been rotated and translated in $\mathbb{R}^4$ so that the coordinate axes do not play any special role.

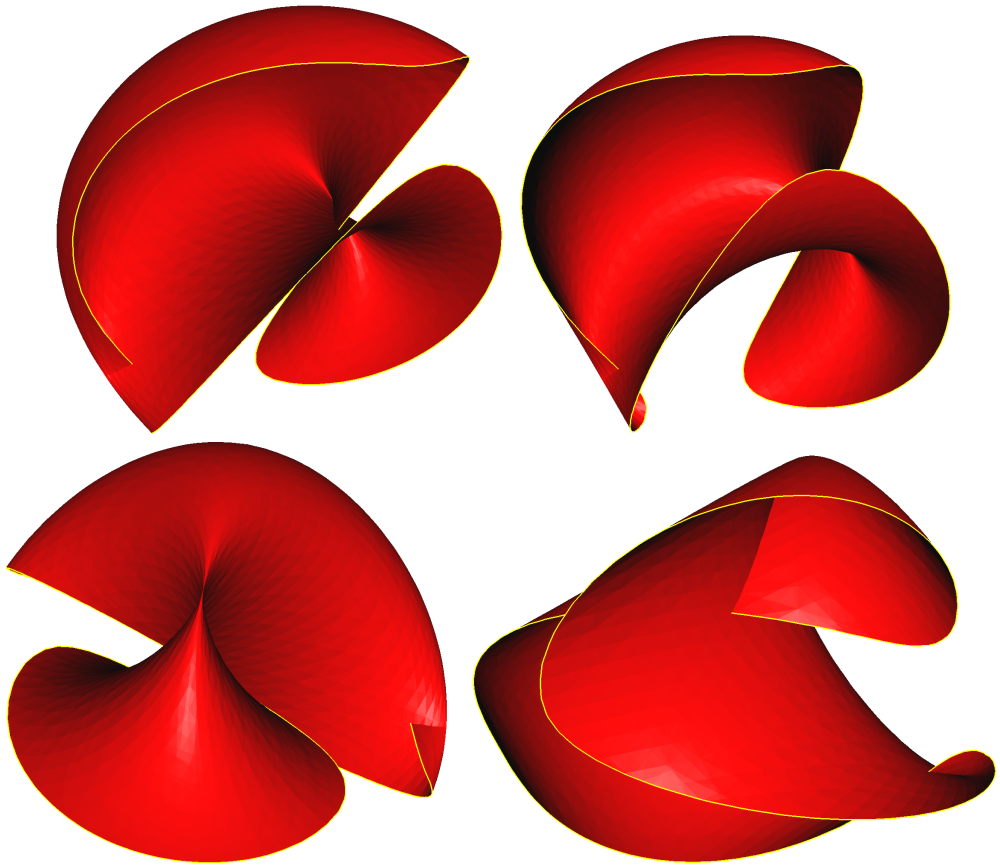

Figure 5 Four views of the flat torus in $\mathbb{R}^4$ given by two equations $x_1^2+x_2^2=1$, and $x_3^2+x_4^2=1$ cut by the hypersphere $(x_1-1)^2+x_2^2+(x_3-1)^2+x_4^2=4$, projected to $\mathbb{R}^3$. The ambient triangulation used is a Coxeter triangulation of type $\tilde{A}_4$ with the diameter 0.15 of the full-dimensional simplices. The reconstructed boundary is highlighted in yellow. The size $|\mathcal{S}|$ of the piecewise-linear approximation is 14 779. The execution time of the algorithm is 1.84s. The torus has been rotated and translated in $\mathbb{R}^4$ so that the coordinate axes do not play any special role.

4.3 An application in algebraic geometry

We also applied our algorithm to a more complicated example of interest in algebraic geometry [4] where an active field of research is to understand the geometry and topology of various projective varieties. Projective varieties are isomanifolds defined by polynomial equations in the complex projective space $\mathbb{CP}^d = (\mathbb{C}^{d+1} \setminus 0)/\mathbb{C}^*$ of complex dimension d. One such example is the complex one-dimensional curve (that is a real dimensional surface) given by the equation $z_1^2\bar{z}_2 + z_2^2\bar{z}_3 + z_3^2\bar{z}_1 = 0$ in $\mathbb{CP}^2$, where $\bar{z}$ denotes the conjugate of the complex number z.

To be able to apply our algorithm, we first need to pass from homogenous coordinates $[z_1 : \ldots : z_{d+1}]$ on $\mathbb{CP}^d$ to affine coordinates $[z'_1 : \cdots : z'_{i-1} : 1 : z'_{i+1} : \cdots : z'_{d+1}]$ by picking the ith coordinate to be equal to 1, that is $z'_j = z_j/z_i$. Given some homogenous coordinates $[z_1 : \ldots : z_{d+1}]$, we can choose the ith coordinate to be set to 1 to be the coordinate whose absolute value is the largest, so that $\mathbb{CP}^d$ can be written as the union of the $d+1$ sets $\left\{[z'_1 : \cdots : z'_{i-1} : 1 : z'_{i+1} : \cdots : z'_{d+1}] \mid |z'_j| \leq 1\right\}$, with the boundaries of these sets identified. Writing $z'_j = x_j + iy_j$ these sets are (seen as real sets) identical to the domain of $\mathbb{R}^{2d}$

$$D_i = \{(x_1, y_1, \ldots, x_{i-1}, y_{i-1}, x_{i+1}, y_{i+1}, \ldots, x_{d+1}, y_{d+1}) \mid x_j^2 + y_j^2 \leq 1\}.$$

Let f be a homogenous polynomial in $d+1$ complex variables and their complex conjugates. For each i, we can fix the ith coordinate to be 1. Writing each variable in terms of its real and imaginary part yields a real inhomogeneous polynomial in $2d$ (real) variables on the domain D_i. Taking the real and imaginary parts of the function yields two real functions $f_{\Re,i}$ and $f_{\Im,i}$ on D_i. As real sets, the projective variety $f = 0$ on $\mathbb{CP}^d$ and the intersection of the sets $f_{\Re,i} = 0$ and $f_{\Im,i} = 0$ on D_i for each i (with the boundaries identified) are the same. We can therefore apply the tracing algorithm to each isomanifold $(f_{\Re,i} = 0, f_{\Im,i} = 0)$ of D_i independently. Since their boundaries coincide, we can then glue these isomanifolds along their boundary to obtain a PL-approximation of the projective variety $f = 0$. This, for example, allows to recover the Euler characteristic of $f = 0$ on $\mathbb{CP}^d$.
This principle generalizes to varieties of higher codimension, that is to varieties defined by a number of homogenous polynomials $f_1, \ldots, f_{d-m}$.

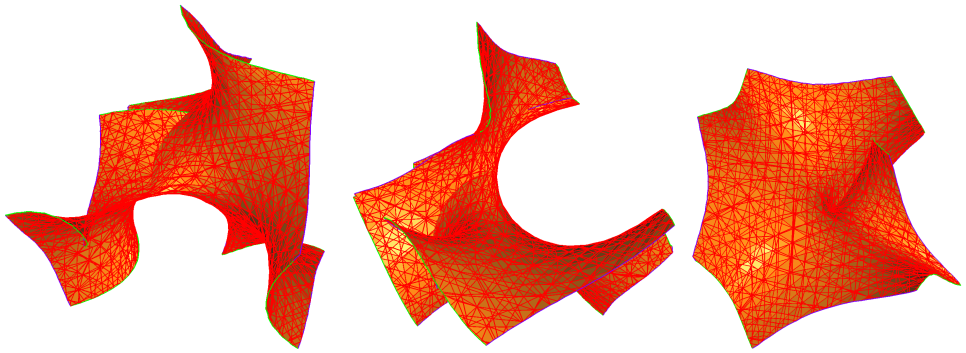

Figure 6 The three triangulated surfaces as discussed in the example of $z_1^2\bar{z}_2 + z_2^2\bar{z}_3 + z_3^2\bar{z}_1 = 0$ in $\mathbb{CP}^2$ after projection from $\mathbb{R}^4$ to $\mathbb{R}^3$.

We illustrate the above construction on the above equation $z_1^2\bar{z}_2 + z_2^2\bar{z}_3 + z_3^2\bar{z}_1 = 0$ in $\mathbb{CP}^2$. By passing to affine coordinates, we recover $z_1^2\bar{z}_2 + z_2^2 + \bar{z}_1 = 0$, $z_1^2 + \bar{z}_3 + z_3^2\bar{z}_1 = 0$, and $\bar{z}_2 + z_2^2\bar{z}_3 + z_3^2 = 0$. By expanding $z_1 = x_1 + iy_1$, $z_2 = x_2 + iy_2$, and $z_3 = x_3 + iy_3$, we find two real equations for each of the complex equations. We give those corresponding to $z_1^2\bar{z}_2 + z_2^2 + \bar{z}_1 = 0$, the other equations being symmetric. For this complex equation, we get the real equations $x_1 + x_1^2x_2 + x_2^2 - x_2y_1^2 + 2x_1y_1y_2 - y_2^2 = 0$ and $-y_1 + 2x_1x_2y_1 - x_1^2y_2 + 2x_2y_2 + y_1^2y_2 = 0$ in $\mathbb{R}^4$. The domain D_3 is in this case determined by the equations $x_1^2 + y_1^2 \leq 1$ and $x_2^2 + y_2^2 \leq 1$. Hence we find a surface in $\mathbb{R}^4$ with a piecewise smooth boundary. The result provided by our algorithm is shown in Figure 6. For visualization purposes, we show the three surfaces separately and projected from $\mathbb{R}^4$ to $\mathbb{R}^3$.

5 Conclusion and open questions

We have presented an efficient, practical and provably correct algorithm to compute the PL-approximation of an isomanifold of any dimension and codimension. Since isomanifolds are a special type of manifolds, it is tempting to see if our algorithm extends to general smooth submanifolds of $\mathbb{R}^d$.

The manifold tracing algorithm itself is quite general and works for any submanifold as soon as we provide a seed point and an oracle that can determine whether a k-simplex of the ambient triangulation intersects $\mathcal{M}$ or not. In this general setting, the simple algorithm described above is sufficient to compute a PL-approximation of the manifold and satisfies the bounds given in Section 3.

However, this is not enough to obtain guarantees on the geometric and topological quality of the output mesh. Such guarantees can be obtained by slightly perturbing the ambient Coxeter triangulation of type $\tilde{A}_d$ so that the following conditions are satisfied:

1. All k-dimensional faces τ in $\mathcal{T}$, with $k \leq d - n - 1$, are far enough from $\mathcal{M}$.
2. The longest edge length of $\mathcal{T}$ is upper bounded and its smallest thickness is lower bounded.

Under these conditions, Algorithm 1 will output a PL-approximation that is topologically equivalent and close in Hausdorff distance to the input manifold [10]. However, the perturbation scheme of [10] perturbs (in the worst case) all the simplices of $\mathcal{T}$ of dimension less than the codimension $d - n$ that are incident on a vertex (in a neighbourhood of $\mathcal{M}$). Since there are exponentially many such simplices, such methods have a complexity that depends exponentially on the ambient dimension d, and have not proved useful in practice except in some simple cases. It remains open whether general smooth manifolds embedded in $\mathbb{R}^d$ can be triangulated in time polynomial in d as we were able to do here in the special case of isomanifolds.

References

1 Eugene Allgower and Kurt Georg. Estimates for piecewise linear approximations of implicitly defined manifolds. *Applied Mathematics Letters*, 2(2):111–115, 1989.

2 Eugene Allgower and Kurt Georg. *Numerical continuation methods: an introduction*, volume 13. Springer Science & Business Media, 1990.

3 Eugene Allgower and Phillip H. Schmidt. An algorithm for piecewise-linear approximation of an implicitly defined manifold. *SIAM Journal on Numerical Analysis*, 22(2):322–346, 1985.

4 Aurélien Alvarez and Bertrand Deroin. Dynamique et topologie du feuilletage de Jouanolou. Preprint, 2019.

5 M. A. Armstrong. *Groups and Symmetry*. Springer, 1988.

6 Dominique Attali, André Lieutier, and David Salinas. Efficient data structure for representing and simplifying simplicial complexes in high dimensions. *International Journal of Computational Geometry & Applications*, 22(04):279–303, 2012.

7 Tom F Banchoff and Wolfgang Kühnel. Equilibrium triangulations of the complex projective plane. *Geometriae Dedicata*, 44(3):313–333, 1992.

8 Jean-Daniel Boissonnat, Frédéric Chazal, and Mariette Yvinec. *Geometric and Topological Inference*. Cambridge Texts in Applied Mathematics. Cambridge University Press, 2018. doi:10.1017/9781108297806.

9 Jean-Daniel Boissonnat and Arijit Ghosh. Manifold reconstruction using tangential Delaunay complexes. *Discrete & Computational Geometry*, 51(1):221–267, 2014.

10 Jean-Daniel Boissonnat, Siargey Kachanovich, and Mathijs Wintraecken. Triangulating submanifolds: An elementary and quantified version of whitney's method. *Discrete & Computational Geometry*, pages 1–49, 2020.

11 Jean-Daniel Boissonnat, CS Karthik, and Sébastien Tavenas. Building efficient and compact data structures for simplicial complexes. *Algorithmica*, 79(2):530–567, 2017.

12 Jean-Daniel Boissonnat and Clément Maria. The simplex tree: An efficient data structure for general simplicial complexes. *Algorithmica*, 70(3):406–427, 2014. doi:10.1007/s00453-014-9887-3.

13 Jean-Daniel Boissonnat and Mathijs Wintraecken. The topological correctness of PL-approximations of isomanifolds. In *34th International Symposium on Computational Geometry, SoCG 2020, June 23-26, 2020, Zurich, Switzerland.*, 2020. Full version. URL: https://hal.inria.fr/hal-02386193.

14 Jean-Daniel Boissonnat and Mariette Yvinec. *Algorithmic Geometry*. Cambridge Texts in Applied Mathematics. Cambridge University Press, 1998.

15 Nicolas Bourbaki. Lie groups and Lie algebras. Chapters 4–6. Translated from the 1968 French original by Andrew Pressley. Elements of Mathematics, 2002.

16 Yen-Chi Chen. Solution manifold and its statistical applications, 2020. arXiv:2002.05297. arXiv:2002.05297.

17 Siu-Wing Cheng, Tamal K Dey, and Edgar A Ramos. Manifold reconstruction from point samples. In *SODA*, pages 1018–1027, 2005.

18 Aruni Choudhary, Siargey Kachanovich, and Mathijs Wintraecken. Coxeter triangulations have good quality. *Mathematics in Computer Science*, 14:141–176, 2020. URL: https://doi.org/10.1007/s11786-020-00461-5.

19 Aruni Choudhary, Michael Kerber, and Sharath Raghvendra. Polynomial-sized topological approximations using the permutahedron. In Sándor P. Fekete and Anna Lubiw, editors, *32nd International Symposium on Computational Geometry, SoCG 2016, June 14-18, 2016, Boston, MA, USA*, volume 51 of *LIPIcs*, pages 31:1–31:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016. doi:10.4230/LIPIcs.SoCG.2016.31.

20 Kenneth L. Clarkson. Tighter bounds for random projections of manifolds. In *Proceedings of the 24th ACM Symposium on Computational Geometry, College Park, MD, USA, June 9-11, 2008*, pages 39–48, 2008. doi:10.1145/1377676.1377685.

21 J. H. Conway and N. J. A. Sloane. *Sphere-packings, Lattices, and Groups*. Springer-Verlag New York, Inc., New York, NY, USA, 1987.

22 Harold S. M. Coxeter. Discrete groups generated by reflections. *Annals of Mathematics*, pages 588–621, 1934.

23 David P. Dobkin, Allan R. Wilks, Silvio V. F. Levy, and William P. Thurston. Contour tracing by piecewise linear approximations. *ACM Transactions on Graphics (TOG)*, 9(4):389–423, 1990.

24 J. J. Duistermaat and J. A. C. Kolk. *Multidimensional Real Analysis I: Differentiation*. Number 86 in Cambridge Studies in Advanced Mathematics. Cambridge University Press, 2004.

25 B. Curtis Eaves. *A course in triangulations for solving equations with deformations*, volume 234. Lecture Notes in Economics and Mathematical Systems, 1984.

26 Armin Eftekhari and Michael B. Wakin. What happens to a manifold under a bi-Lipschitz map? *Discrete & Computational Geometry*, 57(3):641–673, 2017. doi:10.1007/s00454-016-9847-6.

27 Gideon Ehrlich. Loopless algorithms for generating permutations, combinations, and other combinatorial configurations. *Journal of the ACM (JACM)*, 20(3):500–513, 1973.

28 Hans Freudenthal. Simplizialzerlegungen von beschrankter flachheit. *Annals of Mathematics*, pages 580–582, 1942.

29 A. Gomes, I. Voiculescu, J. Jorge, B. Wyvill, and C. Galbraith. *Implicit Curves and Surfaces: Mathematics, Data Structures and Algorithms*. Springer, 2009.

30 GUDHI Project. URL: http://gudhi.gforge.inria.fr/doc/latest/.

31 James E. Humphreys. *Reflection groups and Coxeter groups*, volume 29. Cambridge university press, 1992.

32 Siargey Kachanovich. *Meshing submanifolds using Coxeter triangulations*. Theses, COMUE Université Côte d'Azur (2015 - 2019), October 2019. URL: https://hal.inria.fr/tel-02419148.

33 William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. *ACM siggraph computer graphics*, 21(4):163–169, 1987.

34 M. Maes and B. Kappen. On the permutahedron and the quadratic placement problem. *Philips Journal of Research*, 46(6):267–292, 1992.

35 Chohong Min. Simplicial isosurfacing in arbitrary dimension and codimension. *Journal of Computational Physics*, 190(1):295–310, 2003.

36 Timothy S. Newman and Hong Yi. A survey of the marching cubes algorithm. *Computers & Graphics*, 30(5):854 – 879, 2006. URL: http://www.sciencedirect.com/science/article/pii/S0097849306001336, doi:https://doi.org/10.1016/j.cag.2006.07.021.

37 James M Ortega and Werner C Rheinboldt. *Iterative solution of nonlinear equations in several variables*. SIAM, 2000.

38 Alexander M Ostrowski. *Solution of Equations and Systems of Equations: Pure and Applied Mathematics: A Series of Monographs and Textbooks, Vol. 9*, volume 9. Elsevier, 2016.

39 B.C. Rennie and A.J. Dobson. On Stirling numbers of the second kind. *Journal of Combinatorial Theory*, 7(2):116 – 121, 1969. doi:https://doi.org/10.1016/S0021-9800(69)80045-1.

40 Frank Ruskey and Carla D. Savage. Gray codes for set partitions and restricted growth tails. *Australasian J. Combinatorics*, 10:85–96, 1994.

41 Michael J. Todd. *The computation of fixed points and applications*, volume 124. Lecture Notes in Economics and Mathematical Systems, 1976.

42 Nakul Verma. A note on random projections for preserving paths on a manifold. Technical Report Tech. Report CS2011-0971, UC San Diego, 2011.

43 Timothy R. Walsh. Loop-free sequencing of bounded integer compositions. *Journal of Combinatorial Mathematics and Combinatorial Computing*, 33:323–345, 2000.

44 Rephael Wenger. *Isosurfaces: geometry, topology, and algorithms*. AK Peters/CRC Press, 2013.

45 H. Whitney. *Geometric Integration Theory*. Princeton University Press, 1957.

46 G. M. Ziegler. *Lectures on Polytopes*. Graduate Texts in Mathematics. Springer New York, 2012. URL: https://books.google.fr/books?id=xd25TXSSUcgC.

A A compact data structure for high-dimensional Coxeter-Freudenthal-Kuhn triangulations

Subdivisions of Euclidean space are a major tool to efficiently answer geometric queries, compute approximation of shapes or solve optimization problems. Among the most widely used subdivision schemes are grids and triangulations. Both are subject to the curse of dimensionality and their combinatorial complexity depends exponentially on the dimension of the space. Triangulations are most flexible since their vertex set can be any set of points. Differently, uniform grids depend only on the space but not on a given data set. The rigidity of the grid structure has a major advantage: the grid, although of exponential size, need not be represented explicitly and basic operations like locating a point or computing faces or cofaces of a given cell in the grid can be done without storing an explicit representation of the grid. In fact, the representation can be entirely implicit. This is clearly impossible with general triangulations with arbitrary vertex sets.

The question of designing efficient data structure for triangulations and more general simplicial complexes led to interesting developments recently. On one hand, one can take advantage of the fact that special types of simplicial complexes allow compact representations. Most notably, flag complexes (including the celebrated Vietoris-Rips complex) can be represented by their 1-skeleton (or graph) and higher dimensional faces can be retrieved by computing the cliques of the graph. One can also represent a simplicial complex by its blockers, i.e. the simplices that do not belong to the complex but whose facets do [6].

On a different front, data structures have been proposed to efficiently store general simplicial complexes such as the simplex tree [12] that uses a trie to store the faces of all dimensions, or the Simplex Array List [11] that represents only the maximal faces, which allows an exponential saving in storage since a simplex has exponential complexity. Nevertheless, due to their generality and the fact that the represented complexes don't have any prespecified symmetry, the data structures cannot compete with grids in terms of size and efficiency.

The Coxeter-Freudenthal-Kuhn triangulations in this paper form a middle ground, they form a special class of triangulations of $\mathbb{R}^d$ that have a high regularity. The data structure for such triangulations that can be as compact as for grids.

The Coxeter-Freudenthal-Kuhn triangulations we consider combine two classes of triangulations with different origins and names. The two foundational works are due to Coxeter [22] and Freudenthal [28]. Coxeter triangulations derive from geometric group theory, in particular affine Weyl groups, while Freudenthal triangulations (also called Kuhn triangulations) are combinatorial in nature. Nevertheless, both triangulations are the same up to a linear transformation, as remarked in [23] and fully proved in this paper. This allows us to combine the nice geometric properties of Coxeter triangulations of type $\tilde{A}_d$ with the simple combinatorial definitions of the Freudenthal-Kuhn triangulation, and its connection to permutahedra. Coxeter triangulations of type $\tilde{A}_d$ are geometrically attractive because each simplex is very well shaped (large volume compared to longest edge length), and all d-simplices are identical up to reflections.

Although these triangulations do not depend on a given data set, they proved to be very useful in to interpolate multivariate multivalued functions or to mesh geometric shapes embedded in high dimensional spaces. Freudenthal-Kuhn triangulations have been known in Applied Mathematics [2, 25, 41], and Coxeter triangulations have been used by Dobkin et al. [23] to trace curves in high dimensions and are good candidates to trace manifolds of any codimension [35]. They have also been used in the context of Topological Data Analysis [19].

In Appendix A.1, we study these triangulations. This section recalls and extends to arbitrary

dimensions several results that were disseminated in many different places which are sometimes difficult to access and in different languages (see among others [28, 41, 25, 34, 46, 39, 23]). Based on these results, we introduce in Appendix A.2 a very compact data structure that implicitly stores the full facial structure of such triangulations. The data structure allows to locate a point in the triangulation and to retrieve the faces or the cofaces of a simplex of any dimension in an output sensitive way.
The Data Structure has been implemented and fully tested. Section 4 reports on experimental results and demonstrates that the data structure is highly practical and can handle triangulations of high dimensional spaces. Using our data structure will allow to extend the applicability of the methods based on such triangulations and to significantly improve their performance. It appears to be especially useful to trace low dimensional manifolds embedded in high dimensional spaces as encountered in statistics, dynamical systems, econometrics, or mechanics [16, 41, 36].

A.1 Coxeter-Freudhental-Kuhn triangulations

Freudenthal-Kuhn triangulations are combinatorial structures that come from a specific triangulation of the d-cube. Their connections to permutahedra is at the heart of our data structure. Coxeter triangulations, we introduce in Section A.1.3, have a different flavour and come with very nice geometric properties. Since both types of triangulations are the same up to an affine transformation, as first noted by Dobkin et al [23], they have the same combinatorial structure and our data structure will be able to handle both of them.
Although most ideas in this section were known previously, we give full proofs of the results that were not explicitly mentioned or not proved in full generality in the literature.

A.1.1 Permutahedra

We write $[i] = \{1, \dots, i\}$, and $[i, j] = \{i, \dots, j\}$.

▶ **Definition 14** (Permutahedron). *A d-*permutahedron *is a d-dimensional polytope, which is the convex hull $\mathcal{P}$ of all points in $\mathbb{R}^{d+1}$, the coordinates of which are permutations of $[d+1]$. Formally, this convex hull can be written as:*

$$\mathcal{P} = \text{conv}\{(\sigma(1), \dots, \sigma(d+1)) \in \mathbb{R}^{d+1} \mid \sigma \in \mathfrak{S}_{d+1}\},$$

where $\mathfrak{S}_{d+1}$ denotes the set of permutations of $[d+1]$. If there is a need to distinguish permutahedra of different dimension we write $\mathcal{P}(n)$ the permutahedron of dimension $n-1$.

$\mathcal{P}$ is at most d-dimensional since all its vertices lie on the hyperplane of equation

$$\sum_{i=1}^{d+1} x^i = \frac{d(d+1)}{2}.$$

Moreover, it can be shown that there are $d+1$ affinely independent vertices in $\mathcal{P}$, proving that $\mathcal{P}$ is exactly d-dimensional (see for example [34, Lemma 3.4]). The facial structure of $\mathcal{P}$ is best described in terms of ordered partitions [46]. Refer to Figure 7.

▶ **Definition 15** (Ordered partition). *Let T be a finite non-empty set, $|T|$ its cardinality, and $l \leq |T|$ a positive integer. An* ordered partition *of T in l parts is a collection of l indexed subsets $\omega = (\omega_1, \dots, \omega_l)$, such that $\omega_i \subseteq T$, and $\{\omega_1, \dots, \omega_l\}$ is a partition of T. The ω_i are called the* parts *and are ordered by their index. We write $OP_l[d]$ for the set of ordered partitions of $[d]$ with l parts, and just $OP[d]$ for the set of all ordered partitions of $[d]$.*

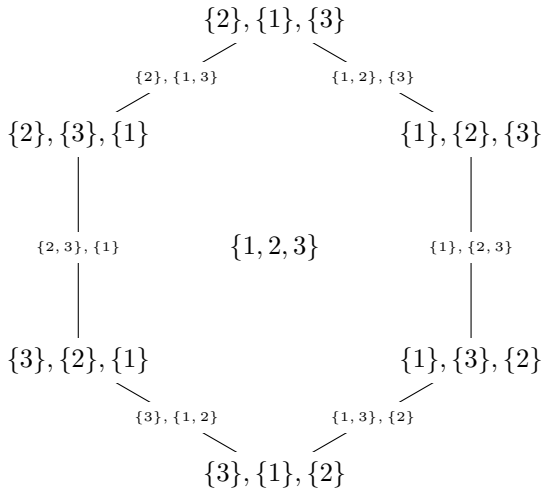

Figure 7 The 2-permutahedron and the ordered partitions associated to its faces.

▶ **Definition 16** (Refinement). *Let ω, and ϖ be two ordered partitions of $[d+1]$ in l, and p parts respectively, with $1 \leq l \leq p \leq d+1$. We say that ϖ is a* refinement *of ω if there exist positive integers $a_1, \ldots, a_l$, such that*

- *$(\varpi_1, \ldots, \varpi_{a_1})$ is an ordered partition of ω_1 in a_1 parts,*
- *$(\varpi_{a_1+1}, \ldots, \varpi_{a_1+a_2})$ is an ordered partition of ω_2 in a_2 parts,*
- *$\ldots$,*
- *$(\varpi_{a_1+\ldots+a_{l-1}+1}, \ldots, \varpi_{a_1+\ldots+a_l})$ is an ordered partition of ω_l in a_l parts.*

We recall Theorem 3.6 of [34]:

▶ **Lemma 17** (Facial structure of the permutahedron). *The faces of a d-permutahedron are in bijection with the ordered partitions of $[d+1]$. More precisely, the i-faces of $\mathcal{P}$ correspond to ordered partitions of $[d+1]$ into $l = d+1-i$ parts $(\omega_1, \ldots, \omega_l)$. If σ, and τ are two faces of a d-permutahedron, σ is a subface of τ (denoted $\sigma \subseteq \tau$) if and only if the ordered partition associated to σ is a refinement of the ordered partition associated to τ.*

We also need the following result from [34, Corollary 3.15], and [39, Theorem 3].

▶ **Corollary 18.** *The number of $(d-i)$-dimensional faces in a d-permutahedron is $(i+1)!\,S(d+1, i+1)$, where $S(\cdot,\cdot)$ is the Stirling number of the second kind. It is bounded by $2^{2(d+1)\log(i+1)}$.*

The following three corollaries seem to be new.

▶ **Corollary 19.** *The number $p_{0,i}$ of vertices of a i-face of a d-permutahedron is at most $(i+1)!$, and at least $2^{\min(i,d-i)}$.*

The proof of Corollary 19 is based on:

▶ **Lemma 20** (Lemma 3.11 of [34]). *The face of a permutahedron corresponding to an ordered partition $\omega = (\omega_1, \ldots, \omega_{l+1})$ is combinatorially*

$$\mathcal{P}(|\omega_1|) \times \cdots \times \mathcal{P}(|\omega_{l+1}|),$$

where $|\omega_p|$ denotes the length of the p-th part of the ordered partition, and $\mathcal{P}(n)$ the permutahedron of dimension $n-1$.

Proof of Corollary 19. Write $l = d - i$. Since the number of vertices of the product of two polytopes is the product of the vertices, and a $(n-1)$-dimensional permutadedron has $n!$ vertices, we see that the total number of vertices of a i-face of a d-dimensional permutahedron corresponding to an ordered partition $\omega = (\omega_1, \ldots, \omega_{l+1})$ is

$$\prod_{p=1}^{l+1}(|\omega_p|!).$$

Let $1 \leq j < k \leq d$ be integers such that $j + k = d + 1$. By definition $j!k! < (j-1)!(k+1)!$, and thus $j!k! \leq 1!d!$. Generalizing this, we see that the product of the $|\omega_p|!$ is maximal when all parts are singletons except the biggest part which has $d + 1 - l$ elements. Therefore

$$\prod_{p=1}^{l+1}(|\omega_p|!) \leq (d-l+1)!.$$

Using the inverse argument, the lower bound is obtained when each part in the ordered partition are as small as possible that is all parts have almost equal size. In this case, $|\omega_p| \geq \lfloor \frac{d+1}{l+1} \rfloor$, so that

$$\prod_{p=1}^{l+1}(|\omega_p|!) \geq \left(\left\lfloor \frac{d+1}{l+1} \right\rfloor !\right)^{l+1}$$

More accurately, let r' be the remainder of $d+1$ after division by $l+1$, that is $r' = d+1 \mod l+1$ then:

$$\prod_{p=1}^{l+1}(|\omega_p|!) \geq \left(\left\lfloor \frac{d+1}{l+1} \right\rfloor !\right)^{l-r'+1} \left(\left(\left\lfloor \frac{d+1}{l+1} \right\rfloor + 1\right)!\right)^{r'}$$

We now distinguish two cases:

- If $\lfloor \frac{d+1}{l+1} \rfloor \geq 2$, and thus $\frac{d+1}{2} \geq l+1$, then we see that

$$\prod_{p=1}^{l+1}(|\omega_p|!) \geq 2^{l+1}.$$

- If $\lfloor \frac{d+1}{l+1} \rfloor = 1$, we have that $\frac{d+1}{2} < l+1 \leq d+1$, and thus $r' = d+1 \mod l+1 = d-l$. Hence,

$$\prod_{p=1}^{l+1}(|\omega_p|!) \geq 2^{d-l}.$$

Because $l + 1 = d - l$, or $2l + 1 = d$ is precisely the point where you go from the first to the second case we see that

$$\prod_{p=1}^{l+1}(|\omega_p|!) \geq 2^{\min\{l+1, d-l\}}$$

◀

▶ **Corollary 21.** *The number of facets of an i-face σ of a d-permutahedron is at most 2^{i+1}.*

Proof. Write $l = d - i$. We first recall a set of $d > 2$ objects can be subdivided in two non-empty ordered subsets A and B in $2^d - 2$ ways. This is not hard to see. Because we pick for each element if it will be put in A or B there are 2^d possibilities. Excluding that A or B is empty gives $2^d - 2$. Let $\omega = (\omega_1, \ldots, \omega_l)$ again be an ordered partition. To find a refinement of ω in $l + 1$ parts, we need to first pick a $1 \leq p \leq l$, such that $|\omega_p| > 1$, and then we need to break ω_p up into two (ordered) parts, for which there are $2^{|\omega_p|} - 2$ possibilities as we have seen above. This means that if $I = \{p \mid 1 \leq p \leq l, |\omega_p| > 1\}$, the number of refinements is

$$\sum_{p \in I} 2^{|\omega_p|} - 2.$$

Let now $1 \leq s < t \leq d$ be integers such that $s + t = d + 1$. Then $2^s + 2^t < 2^{s-1} + 2^{t+1}$. Generalizing this, we see that the sum of the $2^{|\omega_p|} - 2$ is maximal when all $|\omega_p| = 1$ except the biggest part which has $d - l + 1 = i + 1$ elements. ◀

▶ **Corollary 22.** *Let $p_{i,j}$ denote the number of i-faces of a j-face of the d-permutahedron. We have*

$$p_{i,j} \leq \frac{1}{2^{\min(i,d-i)}} \binom{j}{i} (j+1)!$$

Corollary 22 generalizes the previous two corollaries. For $i = 0$, the bound in Corollary 22 is the same as the upper bound in Corollary 19. For $i = j - 1$, the bound is comparable but weaker than the bound in Corollary 21.

Proof of Corollary 22. Let σ be a j-face of the d-permutahedron. Write $F_{i,\sigma}$ for the set of i-faces of σ, and c_v for the number of i-cofaces of a vertex v of σ, i.e. the number of simplices of $F_{i,\sigma}$ that contain v. For $\tau \in F_{i,\sigma}$, we write p_τ for the number of vertices of τ. By double counting the incidences between vertices, and i-faces inside σ, we have

$$\sum_{\tau \in F_{i,\sigma}} p_\tau = \sum_{v \in F_{0,\sigma}} c_v.$$

Now observe that the d-permutahedron is a simple polytope (this follows from the fact that its dual is simplicial since it is a star in the FK-triangulation). The faces of simple polytopes are also simple polytopes which implies that the vertices of a j-face are incident to $\binom{j}{i}$ faces of dimension i [14, Lemma 7.1.14]. Moreover, $|F_{0,\sigma}| \leq (j+1)!$ by Corollary 19. Hence

$$\sum_{v \in F_{0,\sigma}} c_v = \binom{j}{i} |F_{0,\sigma}| \leq \binom{j}{i} (j+1)!.$$

In addition, by Corollary 19, we have

$$\sum_{\tau \in F_{i,\sigma}} p_\tau \geq 2^{\min(i,d-i)} |F_{i,\sigma}|$$

The inequality follows since σ is any j-face of the d-permutahedron. ◀

A.1.2 Freudenthal-Kuhn triangulation

The Freudenthal-Kuhn (FK for short) triangulation is obtained from the d-grid, i.e. the unit cubical tessellation of $\mathbb{R}^d$ that consists of copies of the unit d-cube along the integer lattice $\mathbb{Z}^d$. By triangulating each d-cube in the grid in an appropriate way to be described now, we obtain the FK-triangulation of $\mathbb{R}^d$. The results and definitions below were known to Freudenthal [28], Todd [41], or Eaves [25], mainly for top dimensional simplices and in different guises. We combine these results and extend to simplices of arbitrary (co)dimension, where necessary.

▶ **Definition 23.** *Let $x \in \mathbb{R}^d$, and write $z^i = x^i - \lfloor x^i \rfloor$. We denote by $e_1, \ldots, e_d$ the basis vectors and introduce, for reasons that will be clear later, the extra vector*

$$e_{d+1} = -\sum_{i=1}^{d} e_i.$$

We introduce the convention that $z^{d+1} = 0$. We associate to x the ordered partition $\omega = (\omega_1, \ldots, \omega_{l+1})$ of $[d+1]$ where the ω_i are obtained by sorting the z^i in decreasing order. Specifically, with $\omega_i = \{\omega_i(1), \ldots, \omega_i(m_i)\}$, we have (see Figure 8):

$$\begin{aligned} 1 > z^{\omega_1(1)} = \cdots = z^{\omega_1(m_1)} > \cdots > z^{\omega_l(1)} = \cdots = z^{\omega_l(m_l)} \\ > z^{\omega_{l+1}(1)} = \cdots = z^{\omega_{l+1}(m_{l+1})} = 0. \end{aligned} \tag{2}$$

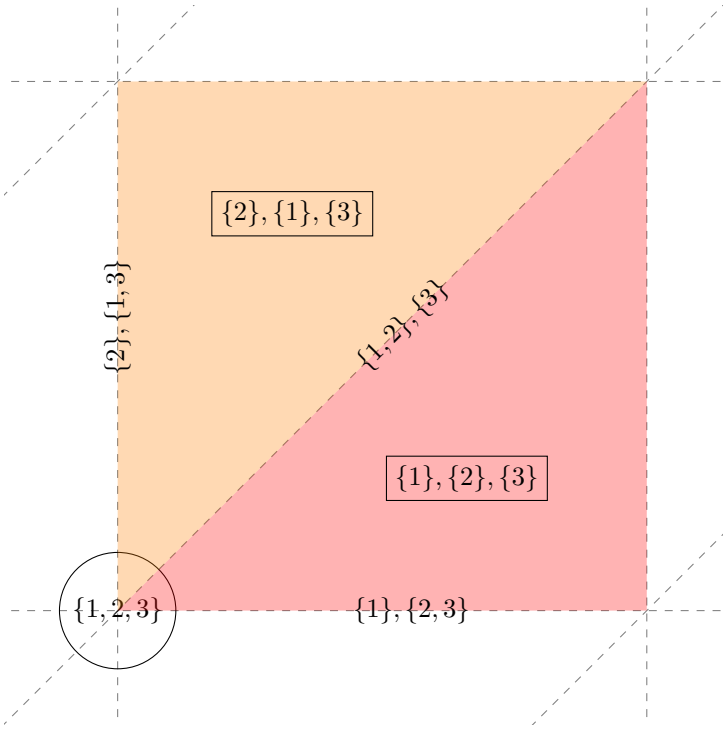

Figure 8 The ordered partitions associated to the faces of the FK-triangulation of $\mathbb{R}^2$ that have the same minimal vertex v_0 (circled).

▶ **Lemma 24.** *Suppose that $\omega = (\omega_1, \ldots, \omega_{l+1})$ is an ordered partition of $[d+1]$ such that $d+1 \in \omega_{l+1}$, and let $\sigma = \{v_0, \ldots, v_l\}$ be the l-simplex whose vertices are the points*

$$v_0 = (\lfloor x^1 \rfloor, \ldots, \lfloor x^d \rfloor), \qquad v_i = v_{i-1} + \sum_{j \in \omega_i} e_j \qquad i = 1, \ldots, l. \tag{3}$$

Then x is a point in the relative interior of σ if and only if $z^i = x^i - \lfloor x^i \rfloor$, $i = 1, \ldots, d+1$ (with, as above, $z^{d+1} = 0$), satisfy (2).

Proof. Because the whole problem is translation invariant, we assume that $v_0 = 0$ without loss of generality, so that the expressions are shorter. Using barycentric coordinates, $z \in \sigma$ can be written as

$$\begin{aligned} z &= \sum_{i=0}^{l} \lambda_i v_i = \sum_{i=0}^{l} \lambda_i \sum_{k=1}^{i} \sum_{j \in \omega_i} e_j \\ &= \lambda_l \left(\sum_{k \in \omega_l} e_k \right) + (\lambda_l + \lambda_{l-1}) \left(\sum_{k \in \omega_{l-1}} e_k \right) + \cdots + (\lambda_l + \cdots + \lambda_1) \left(\sum_{k \in \omega_1} e_k \right). \end{aligned} \tag{4}$$

Here the $\lambda_i > 0$, $i \in [0, l]$, $\sum_{i=0}^{l} \lambda_i = 1$, are the barycentric coordinates of z in σ. We have

$$\begin{aligned} \alpha_{\omega_l(1)} &= \cdots = \alpha_{\omega_l(m_l)} = \lambda_l \\ &\vdots \\ \alpha_{\omega_1(1)} &= \cdots = \alpha_{\omega_1(m_1)} = \lambda_l + \cdots + \lambda_1 \end{aligned} \tag{5}$$

From (4), we see that $\alpha_{\omega_i(j)}$ is the $\omega_i(j)$th coordinate of z, denoted by $z^{\omega_i(j)}$, while all coordinates $z^{\omega_{l+1}(1)}, \ldots, z^{\omega_{l+1}(m_{l+1})}$ are zero, because $e_{\omega_{l+1}(i)}$ does not occur in (4), for all i. Moreover, because $\lambda_l + \cdots + \lambda_i > \lambda_l + \cdots + \lambda_{i-1}$, we see that (2) is satisfied.
Conversely, given a point z such that its coordinates satisfy (2), we can read of its barycentric coordinates with respect to the v_i, as defined by (3), from (5). ◀

▶ **Theorem 25.** *The equivalence classes of the points of $\mathbb{R}^d$ with a same ordered partition are the simplices of a triangulation of $\mathbb{R}^d$ called the FK-triangulation (see Figure 8).*

Proof. Lemma 24 implies that:

- Any face of a simplex in the FK-triangulation also lies in the FK-triangulation.
- The intersection of two simplices in the FK-triangulation also lie in the FK-triangulation.
- For any point $x \in \mathbb{R}^d$, there is a unique simplex σ such that x lies in the relative interior of σ. Indeed, x has uniquely defined barycentric coordinates with respect to the vertices of σ, and thus is mapped to a unique point in σ.

Hence the partition we have defined is a well-defined triangulation of $\mathbb{R}^d$. ◀

▶ **Remark 26.** *We note that, by construction, v_0 in Lemma 24 is the smallest vertex of σ in the lexicographical order. Lemma 24 also implies an observation of Freudenthal [28]: all d-simplices in the FK-triangulation can be described by monotone paths along the edges of the cube from vertex $(0, \ldots, 0) + v_0$ to vertex $(1, \ldots, 1) + v_0$. Conversely, any monotone path along the edges of the cubes from $(0, \ldots, 0) + v_0$ to $(1, \ldots, 1) + v_0$ gives a simplex in the FK-triangulation.*

A.1.3 CFK-triangulations

Freudenthal-Kuhn triangulations are closely related to Coxeter triangulations of type $\tilde{A}_d$ [18] and both are arrangements of hyperplanes as demonstrated now.
Let E be a finite set of vectors of $\mathbb{R}^d$, and consider the set of hyperplanes $H_E = \{x \in \mathbb{R}^d \mid \langle x, u\rangle = k, u \in E, k \in \mathbb{Z}\}$. In generalization of the theory of Coxeter triangulations, we call the set E the set of roots for historic reasons, as mentioned below.
These hyperplanes partition $\mathbb{R}^d$ in a cell complex called the *arrangement* of the hyperplanes. We denote it by $\mathcal{H}_E$.

▶ **Lemma 27.** *The Freudenthal-Kuhn triangulation is the hyperplane arrangement $\mathcal{H}_{E_{FK}}$ associated to the set of vectors $E_{FK} = \{e_1, \dots, e_d\} \cup \{u_{i,j} = e_j - e_i \mid 1 \le i < j \le d\}$.*

Proof. Thanks to Lemma 24, we know that if $x \in \sigma$, where σ is a simplex of dimension less than d, there is at least one equality in (2) on top of $z^{d+1} = 0$. That is $x^i - \tilde{v}_0^i = x^j - \tilde{v}_0^j$ or $x^i - \tilde{v}_0^i = 0$ for some $i, j \neq d+1$. Note that $\tilde{v}_0^i, \tilde{v}_0^j \in \mathbb{Z}$. The converse direction of Lemma 24 gives that if $x^i - \tilde{v}_0^i = x^j - \tilde{v}_0^j$ or $x^i - \tilde{v}_0^i = 0$ for some $i, j \neq d+1$ for $x \in \mathbb{R}^d$, then there is a simplex σ of dimension strictly less than d in the FK-triangulation such that $x \in \sigma$. ◀

Observe that the norms of the vectors in E_{FK} are either 1 or $\sqrt{2}$. By definition, this implies that the distance between the two hyperplanes $\langle x, u\rangle = k$, and $\langle x, u\rangle = k+1$, where $u \in E_{FK}$, is either 1 or $1/\sqrt{2}$.
Let H be the hyperplane of $\mathbb{R}^{d+1}$ of equation $\langle x, \mathbf{1}\rangle = 0$ where $\mathbf{1}$ is the vector of $\mathbb{R}^{d+1}$ whose coordinates are all 1. We now define a linear map μ from $\mathbb{R}^d$ to H by showing how it acts on the basis of $\mathbb{R}^d$: $\mu(e_i) = r_{1,i} = \sum_{j=1}^{i} s_j$, where $s_i = e_i - e_{i+1},\ i = 1, ..., d$. The vectors s_j are called simple roots and play an important role in algebra. Reviewer complained here but have no idea what he meant We refer to [18] for more information.

▶ **Lemma 28.** *μ maps E_{FK} bijectively onto the set E_C defined as*

$$E_C = \left\{ r_{i,j} = \sum_{l=i}^{j} s_l = e_i - e_{j+1} \,\middle|\, 1 \le i \le j \le d \right\}$$

Proof. The vector $\mu(e_i) = r_{1,i}$ lies in E_C, by definition. For $u_{i,j} \in E_{FK}$, with $i < j$, we see that

$$\mu(u_{i,j}) = \mu(e_j - e_i) = \mu(e_j) - \mu(e_i) = r_{1,j} - r_{1,i} = \sum_{l=1}^{j} s_l - \sum_{l=1}^{i} s_l = \sum_{l=i+1}^{j} s_l = r_{i+1,j}.$$

Hence $\mu(u_{i,j})$ lies in E_C. By reading the previous calculation backwards, we see that μ^{-1} maps each $r \in E_C$ to a vector in E_{FK}. ◀

Observe that all vectors in E_C have length $\sqrt{2}$. By definition, this implies that the distance between the two hyperplanes $\langle x, u\rangle = k$, and $\langle x, u\rangle = k+1$, where $u \in E_C$, is $1/\sqrt{2}$.
The image by μ of the Freudenthal-Kuhn triangulation is a triangulation of $\mathbb{R}^d$ which is the arrangement $\mathcal{H}_{E_C}$ associated to the set of vectors E_C. This triangulation is called the *Coxeter triangulation* of type $\tilde{A}_d$ of $\mathbb{R}^d$. By definition, it has the same combinatorial structure as the FK-triangulation. In addition, it has remarkable geometric properties [23, 18]. First, it is a non-degenerate Delaunay triangulation, and its dual complex is a Voronoi diagram. Moreover, its simplices have an exceptionally large thickness (the ratio of the smallest altitude of a simplex over its diameter or longest edge length).

We will call any triangulation of $\mathbb{R}^d$ that is the image of a Freudenthal-Kuhn triangulation under a non-degenerate affine map a Coxeter-Freudenthal-Kuhn triangulation, or CFK-triangulation for short. This includes the Coxeter triangulation of type $\tilde{A}_d$ (as embedded in $\mathbb{R}^d$).

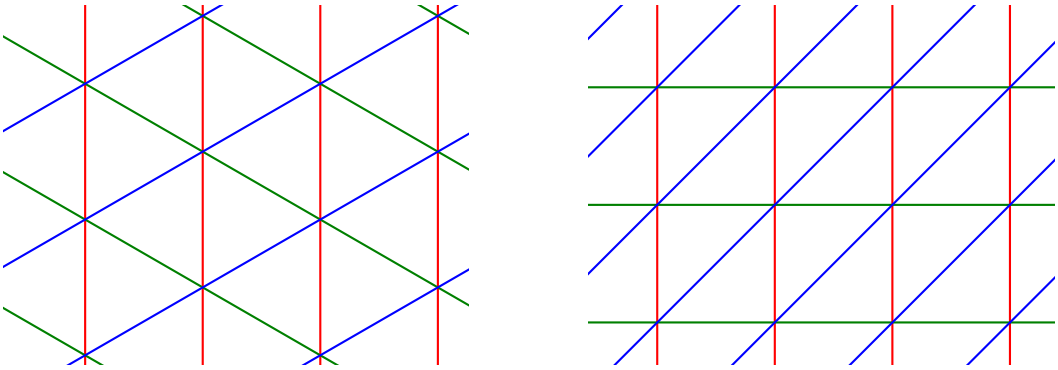

Figure 9 The Coxeter and Freudenthal-Kuhn triangulations in the plane.

A.2 Data Structure

We introduce our data structure in this section. We first consider the case of FK-triangulations in Sections A.2.1 and A.2.2. The extension to CFK-triangulations in Section A.3 is straightforward, since all these triangulations have the same combinatorial structure.

A.2.1 Permutahedral representation of FK-triangulations

Cycles and the permutahedron. In Remark 26 we have seen that simplices can be described by monotone paths (increasing coordinates) along the edges of the cube. As observed by Eaves [25], these monotone paths can be made into a cycle using the extra vector $e_{d+1} = -\sum_{i=1}^{d} e_i$ because by construction

$$v_0 = v_l + \sum_{i \in \omega_{l+1}} e_i,$$

with ω as in Definition 23. Because it is a cycle, we can take any vertex of the cycle as a starting point, which means that v_0 no longer has a special role as a starting point of a monotone edge walk. A cycle can now be represented by an ordered partition of $[d+1]$, for which it is not longer necessary that $d+1$ lies in ω_{l+1}, and an (arbitrary) starting point. We now formalize these general cyclical paths:

▶ **Definition 29** (Permutahedral representation). *Let $(v_0, \omega) \in \mathbb{Z}^d \times OP_{l+1}[d]$. To this pair we associate a simplex $\sigma^\omega = \{v_0 = v_0^\omega, v_1^\omega, \ldots, v_l^\omega\}$ with*

$$v_i^\omega = v_{i-1}^\omega + \sum_{i \in \omega_i} e_i \qquad\qquad i = 1, \ldots, l. \tag{6}$$

We say that (v_0, ω) is the permutahedral representation of the simplex σ^ω. If $d+1 \in \omega_{l+1}$, we say that (v_0, ω) is the canonical permutahedral representation *of σ^ω. In this case, σ^ω is a simplex in the FK-triangulation in the cube of which v_0 is the minimal vertex with respect to the lexicographical order, as we have seen above.*

In Lemma 32, and Proposition 33, we will see that, more generally, $\{(v_0, \omega) \mid \omega \in OP[d+1]\}$ is the star of v_0 in the FK-triangulation, where we identify simplices with their permutahedral representations.

▶ **Definition 30** (Cyclic shifts). *Let (v_0, ω) be a permutahedral representation. We define the cyclic shift of (v_0, ω) of length k to the left as (v_0', ω'), where*

$$v_0' = v_0 + \sum_{j=1}^{k} \sum_{i \in \omega_j} e_i \qquad \omega_j' = \omega_{(j+k-1) \bmod (l+1)+1}. \tag{7}$$

Here we use the convention that the sum from 1 to 0 is empty. We write $(v_0', \omega') = (v_0, \omega) \oplus k$.

▶ **Lemma 31.** *The cyclic shift $(v_0', \omega') = (v_0, \omega) \oplus k$ defines the same simplex as (v_0, ω).*

Proof. Follows by inserting (7) in (6). ◀

We now prove that the all permutahedral representations for a fixed v_0, form the star of v_0. This is a crucial property that will be used to efficiently compute faces, and cofaces, and traverse the triangulation.

▶ **Lemma 32.** *The set $\{(v_0, \omega) \mid \omega \in OP[d+1]\}$, where $OP[d+1]$ is the set of all ordered partitions of $[d+1]$, gives all the simplices in the star of v_0 in FK-triangulation.*

Proof. Let (v_0, ω), with $\omega \in OP_{l+1}[d+1]$, be such that $d+1 \in \omega_k$. Let $(v_0', \omega') = (v_0, \omega) \oplus (l-k+1)$. By Definition 30, and Lemma 31, (v_0, ω), and (v_0', ω') represent the same simplex. Moreover $d+1 \in \omega_{l+1}'$, that is (v_0', ω') is a canonical permutahedral representation. This implies that (v_0', ω') lies in the FK-triangulation by Lemma 24, and Theorem 25.
Conversely, suppose that (v_0', ω') is the canonical permutahedral representation of a simplex in the star of v_0, that is there is some k such that $v_k' = v_0$, with v_k' as in (3). Then $(v_0, \omega) = (v_0', \omega') \oplus k$ is also a permutahedral representation of the same simplex. ◀

Faces. From (6) it is clear that merging two consecutive parts in the ordered partition $\omega = (\omega_1, \ldots, \omega_{l+1})$ corresponds to removing a vertex from the simplex, that is taking a facet. Here we stress that we allow to merge ω_1, and ω_{l+1}, but in that case we have to change the base point of the cycle to $v_0 + \sum_{l \in \omega_1} e_l$ to obtain the canonical representation. For example, when looking at the two dimensional example in Figure 1, we see that the edges that contain y in the red triangle with permutahedral representation $(y, (\{1\}, \{2\}, \{3\}))$ are $(y, (\{1,2\}, \{3\}))$, and $(y, (\{1\}, \{2,3\}))$. The third edge of the red triangle is $(y', (\{2\}, \{1,3\}))$. Generally, given an ordered partition ω in $l+1$ parts all $(l-j)$-faces can be found by merging j consecutive parts in ω (for example merging ω_1 with ω_2, and ω_3 with ω_4), where we allow ω_{l+1} to merge with ω_1, but in this case we again need to change the base point to obtain the canonical representation.
Because the combinatorial structure of the faces is compatible with the permutahedron, Lemma 32 immediately gives the following result. We recall that two complexes are dual if there is a bijection between the faces that inverses the inclusion relationships (see for example [14, Section 11.3]).

▶ **Proposition 33.** *The star of a vertex in a CFK-triangulation is combinatorially dual to a permutahedron.*

This proposition explains the nomenclature permutahedral representation.
This is equivalent to the following more geometric result:

▶ **Proposition 34.** *The Voronoi cell of a Coxeter triangulation of type $\tilde{A}_d$ is a permutahedron.*

This can be found in [21, Chapter 21, Section 3.F], see also [19]. We note that Coxeter triangulations of type $\tilde{A}_d$ are combinatorially equivalent to FK-triangulations as discussed below in Section A.1.3. In the appendix we give a new proof that is more direct than the one in [21].

Duality. We can associate to a FK-triangulation $\mathcal{T}$ its dual complex $\mathcal{T}^*$. Since $\mathcal{T}$ is a simplicial complex, $\mathcal{T}^*$ is a simple complex, that is a cell complex whose faces are all simple polytopes [14]. By Proposition 33, each d-dimensional cell of $\mathcal{T}^*$ is a d-permutahedron.

A.2.2 Basic operations on FK-triangulations

Point location. Given a point $x \in \mathbb{R}^d$ Lemma 24 tells us how to find the canonical permutahedral representation of the simplex in which x is contained. The complexity of point location is dominated by the sorting of the $z^i = x^i - \lfloor x^i \rfloor$, which takes $O(d \log d)$ time, and requires $O(d)$ space.

Face computation. Let σ be an l-simplex whose canonical permutahedral representation is (v_0, ω), where ω is an ordered partition of $[d+1]$ into $l+1$ parts. The computation of all k-faces of σ goes as follows. We use Ehrlich's subset generation algorithm [27] to compute all the subsets of $k+1$ elements from $\{v_0, \dots, v_l\}$. Let $\tau = \{v_{m_0}, \dots, v_{m_k}\}$ be such a subset. τ is a k-face of σ. We then compute the canonical permutahedral representation of all those k-faces τ.

We first sort the m_i so that $m_0 < \cdots < m_k$ using counting sort. Then, the canonical permutahedral representation $(\tilde{v}'_0, \omega')$ of τ is found by merging consecutive parts of ω so as to obtain $k+1$ parts as follows :

$$v'_0 = v_{m_0} = v_0 + \sum_{j \in \omega_1} e_j + \ldots + \sum_{j \in \omega_{m_0 - 1}} e_j$$

$$\omega'_i = \omega_{m_{i-1}} \cup \ldots \cup \omega_{m_i - 1} \quad \text{for } i \in \{1, \dots, k\}$$

$$\omega'_{k+1} = (\omega_1 \cup \ldots \cup \omega_{m_0 - 1}) \cup (\omega_{m_k} \cup \ldots \cup \omega_{l+1}).$$

The complexity of computing all subsets of $k+1$ vertices of σ using Ehrlich's algorithm takes time $O(k+s)$ where $s = \binom{l+1}{k+1}$ is the number of subsets. Computing, for each such k-simplex its permutahedral representation takes $O(d)$ time.

▶ **Lemma 35** (Face computation (full version))**.** *Let σ be an l-simplex in the FK-triangulation of $\mathbb{R}^d$ given by its canonical permutahedral representation. Computing the canonical permutahedral representations of all its k-faces can be done in time $O(ds)$, where $s = \binom{l+1}{k+1}$ is the number of k-faces of an l simplex. The space complexity of the algorithm is $O(l)$.*

Coface computation. Computing the faces of a simplex σ consists in coarsifying its ordered partition. The computation of cofaces is the reverse. Here we refine the ordered partition. Specifically, if σ is a k-simplex represented by its canonical permutahedral representation (v_0, ω), and we want to compute its l-cofaces, we need to compute all refinements of ω into $l+1$ parts.

More precisely, we need to subdivide each ω_i in $a_i \leq |\omega_i|$ subparts so that $\sum_{i=1}^{k+1} a_i = l+1$. This can be done in time proportional to the number $k+1$ of the generated subparts. We then need to consider all the permutations of these subparts since we are interested in ordered

partitions. Using known algorithms by Walsh [43], and Ruskey and Savage [40], we can compute all the ordered partitions associated to the l-cofaces of σ in time proportional to the number of such cofaces. We thus obtain all the permutahedral representations (v_0, ω') of all the l-cofaces of σ.

It is important to notice that all cofaces of σ have v_0 as a vertex. However v_0 is not necessarily the minimal vertex of some of the computed cofaces. We thus have to identify the minimal vertex of each computed coface, and use cyclic shifts (as in Lemma 32) to obtain the canonical permutahedral representation of the coface. The next Lemma follows. The bound on the number s of cofaces follows, by duality, from Corollary 22.

▶ **Lemma 36** (Coface computation (full version)). *Let σ be a k-simplex in the FK-triangulation of $\mathbb{R}^d$ given by its permutahedral representation. Computing the permutahedral representations of all its l-cofaces can be done in time $O(ds)$, where*

$$s = p_{d-l,d-k} \leq \frac{1}{2^{\min(l,d-l)}} \binom{d-k}{d-l} (d-k+1)!$$

is the number of l-cofaces of a k-simplex in the FK-triangulation. The space complexity of the algorithm is $O(d)$.

A.3 Data structure for CFK-triangulations

We store a CFK-triangulation as follows. The combinatorial structure of the triangulation is given through the canonical permutahedral representation of its simplices, and the algorithms from Section A.1.2. The geometry of the triangulation is specified by the affine transformation that maps the FK-triangulation of $\mathbb{R}^d$ to the CFK-triangulation. The affine transformation is given by a $d \times d$ matrix Λ, and a d-vector b. For the FK-triangulation, Λ is the identity matrix, and $b = 0$; therefore no storage is required. For the Coxeter triangulation of type $\tilde{A}_d$, Λ is sparse, as can be seen by inspection of the proof of Lemma 28.

A.4 Experimental results for the datastructure

The data structure and the basic operations have been implemented in C++ and are currently under review to be integrated in the GUDHI library. We report on the execution time of the face and coface generation algorithms for the FK-triangulations.

In Tables 1-4, we consider an ambient space of moderate dimension $d = 30$ and compute the higher dimensional faces of various high dimensional simplices, of dimensions ranging from 22 to 30.

Each entry in Table 1 corresponds to the total time in milliseconds of computing all the k-dimensional faces of a set of l-dimensional simplices in $\mathbb{R}^{30}$. The l-dimensional simplices are picked at random in the triangulation and the results are averaged over 1 000 simplices. Note that the time $11\,904.7ms$ is the time of computing all 5 852 925 faces of dimension 22 of a simplex of dimension 30.

Table 2 shows the same running times per computed face. As we can see, except for the case $l = k$, the running time per computed face is around $2\mu s$.

Face dimension k		22	23	24	25	26	27	28	29
Simplex dimension l	22	0.006							
	23	0.042	0.006						
	24	0.503	0.05	0.008					
	25	4.88	0.645	0.058	0.008				
	26	33.76	5.697	0.697	0.062	0.008			
	27	162.114	35.108	6.824	0.758	0.064	0.008		
	28	885.293	190.441	40.856	6.906	0.739	0.058	0.006	
	29	3420.99	973.455	246.88	49.896	6.657	0.735	0.058	0.006
	30	11904.7	4175.92	1247.97	275.776	50.932	7.348	0.778	0.058

Table 1 Total running time of the face generation algorithm (in milliseconds).

Face dimension k		22	23	24	25	26	27	28	29
Simplex dimension l	22	0.006							
	23	0.0018	0.006						
	24	0.0017	0.002	0.008					
	25	0.0019	0.002	0.0022	0.008				
	26	0.0019	0.0019	0.002	0.0023	0.008			
	27	0.0016	0.0017	0.0021	0.002	0.0023	0.006		
	28	0.0019	0.0016	0.0017	0.0019	0.0018	0.002	0.006	
	29	0.0017	0.0016	0.0017	0.0018	0.0016	0.0017	0.0019	0.006
	30	0.0015	0.0016	0.0017	0.0016	0.0016	0.0016	0.0017	0.0019

Table 2 Running time of the *face* generation algorithm *per computed face* (in milliseconds).

In Tables 3 and 4, we present analogous tables for the coface computation algorithm. Similarly, the running time per computed coface in Table 4 is around $2\mu s$ with the exception of when k is close to l.

	Coface dimension l	23	24	25	26	27	28	29	30
Simplex dimension k	22	0.11	1.274	9.577	43.848	86.699	96.407	59.935	15.487
	23	0.043	0.114	0.729	3.499	9.337	13.523	10.058	3.049
	24		0.047	0.1	0.381	1.183	2.132	1.871	0.653
	25			0.046	0.097	0.23	0.423	0.426	0.193
	26				0.047	0.076	0.128	0.15	0.093
	27					0.049	0.069	0.081	0.063
	28						0.047	0.061	0.054
	29							0.05	0.053
	30								0.05

Table 3 Total running time of the *coface* generation algorithm (in milliseconds).

	Coface dimension l	23	24	25	26	27	28	29	30
Simplex dimension k	22	0.002	0.0013	0.0013	0.0015	0.0016	0.0016	0.0016	0.0017
	23	0.042	0.003	0.0017	0.0016	0.0016	0.0016	0.0016	0.0017
	24		0.045	0.004	0.0019	0.0017	0.0017	0.0017	0.0018
	25			0.045	0.0053	0.0025	0.002	0.0019	0.0022
	26				0.047	0.0073	0.0035	0.0028	0.0036
	27					0.048	0.0103	0.0058	0.0068
	28						0.048	0.0145	0.0133
	29							0.05	0.026
	30								0.05

Table 4 Running time of the *coface* generation algorithm *per computed face* (in milliseconds).

The next results are motivated by the problem of tracing a manifold of low dimension m embedded in $\mathbb{R}^d$ for high d. The crucial operations in this context consist in computing the facets and cofacets of simplices of codimension m in a triangulation of $\mathbb{R}^d$, as is clear from Algorithm 1 .

In Table 5, we present the execution time of the *facet* generation algorithm applied to simplices of low codimension m, ranging from 1 to 7, in FK-triangulations of high dimensions d (up to $d = 400$). In Table 6, we present the execution time of the *cofacet* generation algorithm under the same circumstances.

Ambient dimension d		50	100	150	200	250	300	350	400
Face codimension m	1	0.166	0.612	1.438	2.862	5.376	8.69	12.184	15.924
	2	0.166	0.643	1.417	2.858	5.607	8.375	11.806	16.261
	3	0.168	0.607	1.395	2.888	5.866	8.232	12.008	16.527
	4	0.162	0.589	1.373	2.864	5.491	8.447	11.936	16.08
	5	0.154	0.587	1.349	2.76	5.77	8.371	11.814	15.88
	6	0.148	0.579	1.321	2.737	5.735	8.351	12.038	15.798
	7	0.136	0.575	1.313	2.553	5.701	8.313	12.11	15.754

Table 5 Average running times in milliseconds of the *facet* generation algorithm.

Ambient dimension d		50	100	150	200	250	300	350	400
Simplex codimension m	1	0.068	0.134	0.228	0.281	0.423	0.605	0.611	0.848
	2	0.082	0.17	0.267	0.341	0.483	0.723	0.731	0.966
	3	0.098	0.194	0.303	0.401	0.525	0.733	0.866	1.124
	4	0.112	0.226	0.351	0.467	0.665	0.806	0.974	1.295
	5	0.132	0.265	0.423	0.545	0.966	0.928	1.128	1.477
	6	0.162	0.329	0.515	0.713	0.948	1.124	1.361	1.76
	7	0.2	0.415	0.651	0.878	1.166	1.421	1.784	2.283

Table 6 Average running times in milliseconds of the *cofacet* generation algorithm.

A graphical display of the results of Tables 5 and 6 is shown in Figure 10.

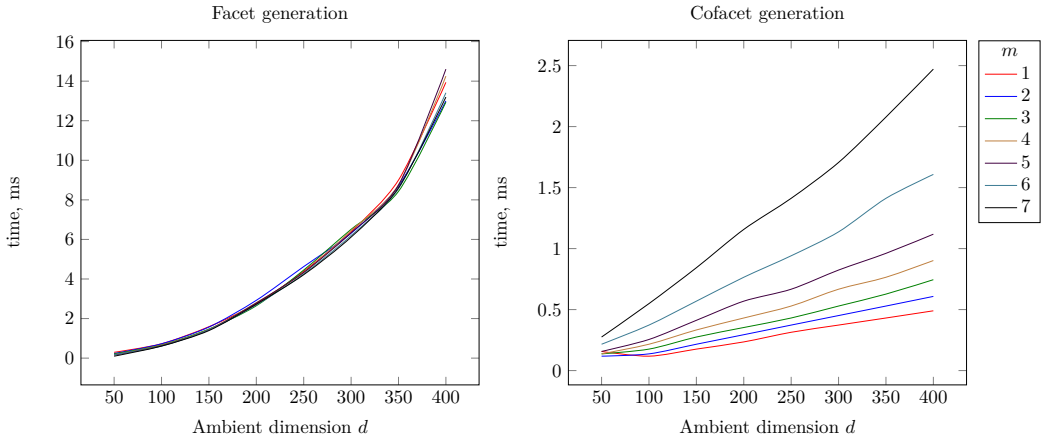

Figure 10 Graphical display of the results of Tables 5 and 6.

B Proofs

Proof of Proposition 9. The complexity of the initialization is $O(d)$. The complexity of each iteration of the while loop consists of computing the cofacets of the popped k-dimensional simplex in the queue, computing facets of these cofacets, and applying the intersection oracle on each of these facets. An upper bound on the number of cofacets of a k-simplex in a CFK-triangulation follows, by duality, from Corollary 21, specifically $O(2^n)$. Each of these cofacets has $k+2$ facets. Therefore, for each iteration of the while loop, the algorithm applies the intersection oracle on $O(k2^n)$ simplices. By using this observation, and the complexities in Lemmas 35, and 36, the total time complexity of each iteration of the while loop follows:

$$O(d2^n) + O(dk2^n) + O(k2^n I) = O(k2^n(d+I)) = O(k2^n I).$$

Since there are $|\mathcal{S}|$ iterations of the while loop, the result follows. ◀

Proof of Proposition 10. By the definition of CFK-triangulations in Section A.1, $\mathcal{T}$ is an arrangement of $d(d-1)/2$ families H_u of hyperplanes, $u \in E_{\mathcal{T}}$. Each family H_u consists of the hyperplanes $H_{u,k}, k \in \mathbb{Z}$, all orthogonal to u. Let $L_{\mathcal{T}}$ be the length of the longest edge of a simplex in $\mathcal{T}$ and R_τ be the maximal norm of the vectors u. Note that the distance between two consecutive hyperplanes in family H_u is $1/\|u\| \geq 1/R_\tau$.
We will rescale the arrangement of hyperplanes so that the maximal diameter of the simplices is D, the required precision. Hence the distance between two consecutive hyperplanes in H_u is $D/(L_{\mathcal{T}}\|u\|)$. It follows that at most $\sqrt{d}L_{\mathcal{T}}\|u\|/D$ hyperplanes of family H_u intersect the unit cube C_d that contains $\mathcal{M}$ (which has diameter $\sqrt{d}$). Consider any subset of n families among the $d(d-1)/2$ families, and write I for the associated subset of indices, $I \subset [1, d(d-1)/2], |I| = n$. Now take n hyperplanes, one in each family $H_{u_i}, i \in I$. Their common intersection is an affine space of dimension $k = d-n$. This affine space intersects $\mathcal{M}$ in at most K points under the general position assumption and the fact that $\mathcal{M}$ is K-sparse. The total number of intersection points $N_{\mathcal{T}} = \mathcal{T}_k \cap \mathcal{M}$ is thus bounded as follows

$$N_{\mathcal{T}} \leq K \binom{d(d-1)/2}{n} \times \prod_{i \in I} \frac{\sqrt{d}L_{\mathcal{T}}\|u_i\|}{D} \leq \frac{K}{n!} \times \left(\frac{d^2\sqrt{d}L_{\mathcal{T}}R_{\mathcal{T}}}{2D}\right)^n. \tag{8}$$

Here the binomial coefficient arises as the number of choices of n families of hyperplanes.
Consider now more specifically Coxeter triangulations of type $\tilde{A}_d$ and FK-triangulations. It follows from Section A.1 that $R_C = R_{FK} = \sqrt{2}$. The longest edge L_{FK} in a Freudenthal-Kuhn triangulation has length at most (in fact exactly) $\sqrt{d}$ since each simplex is contained in a cubical cell of the d-dimensional unit grid. Furthermore, it is proved in [18, point 6 of $\tilde{A}_d$ in Section 6] that the longest edge length in the Coxeter triangulation of type $\tilde{A}_d$ is

$$L_C = \begin{cases} \frac{\sqrt{d+1}}{2} & \text{if } d \text{ is odd,} \\ \frac{1}{2}\sqrt{\frac{d(d+2)}{(d+1)}} & \text{if } d \text{ is even,} \end{cases} \tag{9}$$

and hence $L_C < \frac{\sqrt{d+2}}{2}$. We then deduce from (8)

$$N_C \leq \frac{K}{n!} \times \left(\frac{d^2\sqrt{d(d+2)}}{2\sqrt{2}D}\right)^n$$

$$N_{FK} \leq \frac{K}{n!} \times \left(\frac{d^3}{\sqrt{2}D}\right)^n.$$

◀

Proof of Proposition 11. Let σ be a k-simplex of a CFK-triangulation that intersects $\hat{\mathcal{M}}$, and let σ^* be its dual cell. By definition, σ^* is a m-dimensional face of $\mathcal{T}^*$, the polytopal cell complex dual to $\mathcal{T}$. The collection of all σ^* associated to the k-simplices σ of $\mathcal{T}$ that intersect $\hat{\mathcal{M}}$ form a cell complex $\hat{\mathcal{M}}^*$ dual to $\hat{\mathcal{M}}$. To bound the number of faces of all dimensions of the PL-approximation $\hat{\mathcal{M}}$, it is therefore sufficient to bound the number of faces of $\hat{\mathcal{M}}^*$. Each d-dimensional cell in $\mathcal{T}^*$ is a permutahedron (Proposition 33). Hence, σ^* is a n-face of a d-permutahedron. The number of faces of σ^* of dimensions 0 to $n-1$ (or equivalently the number of cofaces of σ of dimensions $n+1$ to d) is:

$$\sum_{i=0}^{n-1} p_{i,n} \leq \sum_{i=0}^{n-1} \frac{1}{2^i} \binom{n}{i} (n+1)! = \frac{3^n-1}{2^n}(n+1)!$$

where $p_{i,j}$ denotes the number of i-faces of a j-face of the d-permutahedron and is bounded in Corollary 22. The last equality can be easily verified using Mathematica. The overall combinatorial complexity of $\hat{\mathcal{M}}$ is therefore

$$|\mathcal{S}| \times \frac{3^n-1}{2^n}(n+1)!,$$

where $\mathcal{S}$ is bounded in Proposition 10. ◀

C Alternative Proof of Proposition 34

Proof of Proposition 34. We start by recalling a number of results. Let $P = \{(x^i) \in \mathbb{R}^{d+1} \mid \sum_i x^i = 0\}$ and consider the d-simplex with vertices u_k in P.

$$u_0 = \left(0^{\{d+1\}}\right) \qquad u_k = \left(\left(-\frac{d+1-k}{d+1}\right)^{\{k\}}, \left(\frac{k}{d+1}\right)^{\{d+1-k\}}\right), \qquad k \in [d],$$

where $x^{\{k\}}$ denotes k consecutive coordinates x. This simplex is a simplex in the Coxeter triangulation, as defined in Section A.1.3. In [18] we have seen that the circumcentre of this simplex is

$$c = \left(-\frac{d-2i}{2(d+1)}\right),$$

with $i \in \{0, \ldots, d\}$. The circumcentre of a Delaunay simplex is a Voronoi vertex. We recall that

- All simplices in the star of 0 in the Coxeter triangulation are found by consecutive reflection of a single simplex (in this star) in the hyperplanes of $\mathcal{H}_{E_C}$ that go through 0, that is the hyperplanes with normals $r_{j,k} = e_j - e_k$, with $j \neq k$. See for example [15, 31, 18]. We also call these reflections the action of the Weyl group.
- The reflection $R_{j,k}$ in a plane that goes through the origin with normal $r_{j,k}$ is given by

$$R_{j,k}(v) = v - 2\frac{v \cdot r_{j,k}}{r_{j,k} \cdot r_{j,k}} r_{j,k} = v - (v \cdot r_{j,k}) r_{j,k}.$$

We find that

$$R_{j,k}\,(c)^i = (c - (c \cdot r_{j,k}) r_{j,k})^i = -\frac{d-2i}{2(d+1)} - \frac{2j-2k}{2(d+1)}(\delta_{ij} - \delta_{ik}),$$

which permutes the jth and kth coordinate of c. Here we used the upper index i to denote the ith coordinate. Using the cycle notation for the permutation group, see for example [5, Chapter 6], this coincides the 2-cycle $(j\,k)$. Let now

$$c_\pi = \left(-\frac{d-2\pi_i}{2(d+1)}\right),$$

with $\{\pi_i\}$ some permutation of $\{0,\dots,d\}$. We find that

$$R_{j,k}\left(c_\pi\right)^i = \left(c_\pi - (c_\pi \cdot r_{j,k})r_{j,k}\right)^i = -\frac{d-2\pi_i}{2(d+1)} - \frac{2\pi_j - 2\pi_k}{2(d+1)}(\delta_{ij}-\delta_{ik}),$$

which again permutes the jth and kth coordinate. Now recall that all permutations are generated by 2-cycles, see for example [5, Theorem 6.1]. This implies that, for any permutation π, we can find c_π from c by the action of the Weyl group. This also means that we have explicitly described the Voronoi cell of 0 in the Coxeter triangulation of type $\tilde{A}_d$ as a permutahedron. Because of symmetry, this now holds for any Voronoi cell. ◀