Tracing Isomanifolds of Fixed Dimension in Polynomial Time

Jean-Daniel Boissonnat, Siargey Kachanovich, Mathijs Wintraecken

Tracing Isomanifolds of Fixed Dimension in Polynomial Time

Jean-Daniel Boissonnat*, Siargey Kachanovich†, and Mathijs Wintraecken‡

July 3, 2020

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 $f : \mathbb{R}^d \to \mathbb{R}^{d-m}$ where m 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 or mesh $\hat{\mathcal{M}}$ based on a triangulation $\mathcal{T}$ of the ambient space $\mathbb{R}^d$. In this paper, we present a simple algorithm to construct such an approximation for arbitrary m and d up to a given precision D. The complexity of our algorithm is polynomial in d and δ, and exponential in m. Since, by previous results, $\hat{\mathcal{M}}$ is $O(D^2)$-close and isotopic to $\mathcal{M}$ when $\delta = \Omega(d^{2.5})$, our algorithm constructs a geometrically close and topologically correct PL-approximation of isomanifolds of low dimensions in polynomial time. The algorithm is practical and can handle cases that are far ahead of the state-of-the-art. Combining this algorithm with dimensionality reduction techniques, the dependency on d in the sizes of the output sample and mesh can be completely removed with high probability.

The crux of our algorithm is to use for the ambient triangulation $\mathcal{T}$ a regular triangulation from a particular family. This family consists of Freudenthal-Kuhn triangulations and their images through affine mappings. It also includes Coxeter triangulations of type $\tilde{A}_d$. We introduce an elegant and very compact data structure to implicitly store the full facial structure of such triangulations. This data structure allows to retrieve the faces or the cofaces of a simplex of any dimension in an output sensitive way, which is essential for our application and is of independent interest.

Keywords: Coxeter triangulation, Kuhn triangulation, permutahedron, PL-approximations, isomanifolds

*Université Côte d'Azur, Inria, France, email: jean-daniel.boissonnat@inria.fr
†Université Côte d'Azur, Inria, France, email: siargey.kachanovich@inria.fr
‡IST Austria, email: m.h.m.j.wintraecken@gmail.com

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 higher dimensions and codimensions, in which case the generalized surface is called an isomanifold. Isosurfaces play a crucial role in medical imaging, computer graphics and geometry processing [NY06]. Higher dimensional isomanifolds are also of fundamental importance in many fields like statistics [Che20], dynamical systems [Tod76], econometrics or mechanics [NY06].

State-of-the-art. 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. Although early work considered the existence of triangulations rather than their actual construction [Cai34, Whi40] , Whitney's approach is more constructive and has some similarity with the Marching cube algorithm [Whi57, BKW18]. Algorithmic results have been obtained more recently for surfaces of $\mathbb{R}^3$ in the Computational Geometry and Computer Graphics communities. Among the widely used methods, are the marching algorithms and Delaunay refinement [GVJ$^+$09, Wen13, CDS12, BCSM$^+$06]. The marching algorithms use a tesselation of the ambient space (typically a cubical grid or a triangulation). They start from a seed point, which lies on the manifold, and then march in the tesselation around the manifold, while computing an approximation of the manifold on the fly. Delaunay refinement algorithms typically construct a point set that samples the manifold using a furthest point strategy, and maintain the Delaunay triangulation of the sample points in the ambient space. The approximation of the manifold is finally extracted from this Delaunay triangulation. Both methods have strong limitations in higher dimensions due to the fact that any tesselation of the ambient space has a complexity that depends exponentially on the ambient dimension that can be large in applications.

A key to circumvent the curse of dimensionality is to use an *implicit* representation of the ambient tesselation. This is easy if one uses a grid, but using a grid 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 [Wen13]. The most promising approach seems then to subdivide the ambient space $\mathbb{R}^d$ using a triangulation of the cubical grid such as the Freudenthal-Kuhn triangulation. By varying the diameter D of the cubical cells, we can control the precision of the approximation. Some early work along this direction has been published in Applied Mathematics [AG90, Eav84, Tod76] and a slightly more recent paper by Dobkin et al. [DWLT90] attracted the interest of the Computer Graphics community to the related Coxeter triangulations. Dobkin et al. however only considered the case of curves ($m = 1$). The most advanced work we are aware of is due to Min [Min03]. Min's method uses the Freudenthal-Kuhn triangulation over a dyadic grid of $\mathbb{R}^d$ and applies to iso-manifolds of any dimension and codimension. The complexity of the method is, with our notations, $O(\delta^m \log \delta)$ where $\delta = 1/D$ and the ambient dimension d is a constant hidden in the big O. The fact that the exponent of δ is the intrinsic dimension m and not the ambient dimension d is a clear improvement over earlier methods. However, although not explicitly analysed, the complexity in d remains exponential and the method seems limited to small ambient dimensions. Experimental results are only reported in dimensions 3 and 4.

Contributions. In this paper, we extend the work of Dobkin et al. [DWLT90] and describe a rather simple algorithm to sample and mesh an m-dimensional isomanifold $\mathcal{M}$ of $\mathbb{R}^d$ for arbitrary m 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. D is the largest diameter of the simplices of the triangulation. We study these triangulations and introduce an elegant and very compact data structure to implicitly store the full facial structure of such triangulations (Section 2). This data structure allows to retrieve the faces or the cofaces of a simplex of any dimension in an output sensitive way. Contrary to Min [Min03], our results are obtained with a uniform triangulation leading to a much simpler algorithm.

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 m. The complexity of the algorithm is also polynomial in d and δ, and exponential in m.

Remarkably, as proved in our companion paper [BW20], if we take δ to be polynomially large as a function of d, specifically $\delta = \Omega(d^{2.5})$, then the PL-approximation $\dot{\mathcal{M}}$ is $O(D^2)$-close and isotopic to the isomanifold. Hence, our algorithm constructs a geometrically close and topologically correct PL-approximation of isomanifolds of low dimensions in polynomial time.

As shown by numerous experiments, the algorithm is practical and can handle cases that are far ahead of the state-of-the-art (Section 4).

A number of extensions are considered in Section 5. First, we show that the dependency on d in the size of $\hat{\mathcal{M}}$ can be completely removed by combining our algorithm with dimensionality reduction. We also discuss the case of isomanifolds with boundary and stratifolds. Finally, we show that the algorithm can be easily adapted to approximate general smooth submanifolds. However, in this case, topological guarantees can only be obtained using perturbation techniques à la Whitney which, at least currently, are incompatible with polynomial complexity.

2 Coxeter-Freudhental-Kuhn triangulations

Coxeter and Freudhental-Kuhn triangulations have different origins. Coxeter triangulations derive from geometric group theory, in particular affine Weyl groups while Freudenthal-Kuhn triangulations are combinatorial by nature. Nevertheless, both triangulations are the same up to a linear transformation, as remarked in [DWLT90] and fully proved in the appendix. 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.

The first part of this section is expository. It recalls several useful results and ideas that were disseminated in many different areas and difficult to access. These results are combined and extended to arbitrary dimensions so as to provide an efficient data structure to represent a Coxeter or Freudenthal-Kuhn triangulation.

2.1 Permutahedra

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

Definition 1 (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: $\mathcal{P} = \text{conv}\{(\sigma(1), \ldots, \sigma(d+1)) \in \mathbb{R}^{d+1} \mid \sigma \in \mathfrak{S}_{d+1}\}$, where $\mathfrak{S}_{d+1}$ denotes the set of permutations of $[d+1]$.*

$\mathcal{P}$ 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 [MK92, Lemma 3.4]). The facial structure of $\mathcal{P}$ is best described in terms of ordered partitions [Zie12].

Definition 2 (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 an ordered collection of l subsets $\omega = (\omega_1, \ldots, \omega_l)$, such that $\omega_i \subseteq T$ and $\{\omega_1, \ldots, \omega_l\}$ is a partition of T. The ω_i are called the* parts*. 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 $[d]$.*

Definition 3 (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 [MK92]:

Lemma 4 (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 [MK92, Corollary 3.15] and [RD69, Theorem 3].

Corollary 5. *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 $2^{2(d+1)\log(i+1)}$.*

Corollary 6. *The number $p_{0,i}$ of vertices of a i-face of a d-permutahedron is at most $(i+1)!$ and at least $2^{\min(i, d-i)}$.*

Corollary 7. *The number of facets of an i-face σ of a d-permutahedron is at most 2^{i+1}.*

Corollary 8. *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 8 generalizes the previous two corollaries. For $i = 0$, the bound in Corollary 8 is the same as the upper bound in Corollary 6. For $i = j - 1$, the bound is comparable but weaker than the bound in Corollary 7.

2.2 Freudhental-Kuhn triangulation

The Freudhental-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 $\mathbb{R}^d$.

Definition 9. *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*

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

Lemma 10. *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*

$$v_0 = (\lfloor x^1 \rfloor, \ldots, \lfloor x^d \rfloor), \qquad v_i = v_{i-1} + \sum_{j \in \omega_i} e_j \qquad i = 1, \ldots, l. \tag{2}$$

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 (1).

Theorem 11. *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 6).*

Remark 12. *We note that, by construction, v_0 in Lemma 10 is the smallest vertex of σ in the lexicographical order. Lemma 10 also implies an observation of Freudenthal [Fre42]: 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.*

Cycles and the permutahedron. As observed by Eaves [Eav84], this monotone path 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 9. 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 13 (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*

$$v_i^\omega = v_{i-1}^\omega + \sum_{i \in \omega_i} e_i \qquad i = 1, \ldots, l. \tag{3}$$

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 16 and Proposition 17, 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 14 (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*

$$v_0' = v_0 + \sum_{j=1}^{k} \sum_{i \in \omega_j} e_i \qquad \omega_j' = \omega_{(j+k-1) \bmod (l+1)+1}. \tag{4}$$

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

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 16. *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.*

Faces. From (3) 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 7, 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 16 immediately gives:

Proposition 17. *The star of v_0 is dual to a permutahedron (combinatorially).*

Here duality is used in the sense of [BY98, Section 11.3]. This proposition also explains the nomenclature permutahedral representation.

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 [BY98]. By Proposition 17, each d-dimensional cell of $\mathcal{T}^*$ is a d-permutahedron.

2.3 Basic operations

Point location. Given a point $x \in \mathbb{R}^d$ Lemma 10 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 [Ehr73] to compute all the subsets of $k+1$ elements from $\{v_0, \ldots, v_l\}$. Let $\tau = \{v_{m_0}, \ldots, v_{m_k}\}$ 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 :

$$\begin{aligned} v_0' = v_{m_0} &= v_0 + \sum_{j \in \omega_1} e_j + \ldots + \sum_{j \in \omega_{m_0 - 1}} e_j \\ \omega_i' &= \omega_{m_{i-1}} \cup \ldots \cup \omega_{m_i - 1} \quad \text{for } i \in \{1, \ldots, k\} \\ \omega_{k+1}' &= (\omega_1 \cup \ldots \cup \omega_{m_0 - 1}) \cup (\omega_{m_k} \cup \ldots \cup \omega_{l+1}). \end{aligned}$$

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 18. *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 $O(l)$ from the counting sort.*

Coface computation. Computing the faces of a simplex σ consisted 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 [Wal00] and Ruskey and Savage [RS94], 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 16 to obtain the canonical permutahedral representation of the coface.

Lemma 19. *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 is the number of l-cofaces of a k-simplex in the FK-triangulation. The space complexity of the algorithm is $O(d)$.*

2.4 CFK-triangulations

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.

These hyperplanes partition $\mathbb{R}^d$ in a cell complex called the *arrangement* of the hyperplanes. We denote it by $\mathcal{H}_E$.

Lemma 20. *The Freudenthal-Kuhn triangulation is the hyperplane arrangement $\mathcal{H}_{E_{FK}}$ associated to the set of vectors $E_{FK} = \{e_1, \ldots, e_d\} \cup \{u_{i,j} = e_j - e_i \mid 1 \leq i < j \leq d\}$.*

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 $\mathbb{R}^d$:

$$\mu(e_i) = r_{1,i} = \sum_{j=1}^{i} s_j \quad \text{where} \quad s_i = e_i - e_{i+1},\ i = 1, ..., d.$$

Lemma 21. *μ maps E_{FK} bijectively onto the set E_C defined as*

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

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 [DWLT90, CKW20]. 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). This fact will be used in our complexity analysis

(Proposition 24) and the superiority of the Coxeter triangulation will also be demonstrated experimentally (see Section 4).
We will call any triangulation of $\mathbb{R}^d$ that is the image of a Freudenthal-Kuhn or Coxeter triangulation under a non-degenerate affine map a Coxeter-Freudenthal-Kuhn triangulation, or CFK-triangulation for short. We store a CFK-triangulation as follows. The combinatorics of the triangulation is given through the canonical permutahedral representation of its simplices and the algorithms from Section 2.2. 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 indicated above.

3 Tracing isomanifolds

In this section, we describe an algorithm that computes a PL-approximation of an isomanifold. The algorithm can be considered as an alternative to the Marching Cube algorithm [LC87] where the usual cubical grid is replaced by a CFK triangulation of the ambient space. Taking a triangulation instead of a grid is a major advantage in high dimensions that has been recognized in the pioneering works of Allgower and Schmidt [AS85] and of Dobkin et al. [DWLT90]. See also [Min03]. By taking the data structure of Section 2 to represent a CFK-triangulation, we 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 [DK04, Section 3.5]. We further assume that $f^{-1}(0)$ is compact. As in [AG89] 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.

$$\forall x \in \sigma : \quad \hat{f}(x) = \sum_{v \in \sigma} \lambda_v(x) f(v), \tag{5}$$

where the λ_v are the barycentric coordinates of x with respect to the vertices 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 a $(j-d+m)$-flat H_σ with σ. This intersection is (again generically) a $(j-d+m)$-dimensional polytope or cell denoted by C_σ. The PL-approximation or mesh $\hat{\mathcal{M}}$ of $\mathcal{M}$ is the polytopal cell complex obtained by gluing these cells associated to all the simplices in $\mathcal{T}$.

3.2 Manifold tracing algorithm

Let $\mathcal{M}$ be the zero set of some function $f : \mathbb{R}^d \to \mathbb{R}^{d-m}$ and let $\hat{\mathcal{M}}$ be the associated PL-approximation defined over a triangulation $\mathcal{T}$ of the ambient space $\mathbb{R}^d$. Both m and d are known but arbitrary and will be considered as parameters in the complexity analysis. We write $k = d - m$ for the codimension of $\mathcal{M}$. The algorithm will use for $\mathcal{T}$ a CFK-triangulation stored using the data structure from Section 2. We assume that the manifold $\hat{\mathcal{M}}$ and the triangulation $\mathcal{T}$ satisfy the following genericity hypothesis:

Hypothesis 22 (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.*

We note that this condition is satisfied with probability 1 for isomanifolds, because two affine spaces whose dimension does not add up to the ambient dimension don't intersect with probability 1. 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 m-flat H_σ that interpolates f inside σ.
We assume that we know a point on the manifold $x_0 \in \mathcal{M}$, from which the algorithm starts. If $\mathcal{M}$ consists of multiple connected components, then a seed point per each connected component must be provided and we proceed in the same manner for each component. So we will assume for now that $\mathcal{M}$ is connected.

Algorithm 1: Manifold tracing algorithm

input : Triangulation $\mathcal{T}$ of $\mathbb{R}^d$, manifold $\hat{\mathcal{M}}$ of dimension m, seed point $x_0 \in \hat{\mathcal{M}}$
output: Set $\mathcal{S}$ of the simplices in $\mathcal{T}$ of dimension $k = d - m$ that intersect $\hat{\mathcal{M}}$, represented by their permutahedral representation

```
Translate T so that x_0 coincides with the barycentre of a k-dimensional face τ_0 in T
Initialize the queue Q and the set S with τ_0
while the queue Q is not empty do
    Pop a k-dimensional simplex τ from Q
    foreach cofacet φ of τ do
        foreach facet σ of φ do
            if σ does not lie in S and intersects M̂ then
                Insert σ to the queue Q
                Insert σ together with the intersection point to the output set S
```

In this paper, we consider that a seed point is given and don't discuss the problem of computing a seed point that reduces to finding a solution to a system of equations, on which a large body of literature exists. See for example [Ost16, OR00, DWLT90].

The algorithm essentially computes the set $\mathcal{S}$ of k-simplices of $\mathcal{T}$ that intersect $\hat{\mathcal{M}}$. The elements of $\mathcal{S}$ which 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.

The algorithm is described as Algorithm 1. We first translate the coordinate frame so that x_0 is the barycenter of a k-simplex of $\mathcal{T}$ (any such simplex is fine). This simplex is put in the set $\mathcal{S}$ of the simplices in $\mathcal{T}$ of dimension $k = d - m$ that intersect $\hat{\mathcal{M}}$. Then, given such a simplex, we look at all its cofacets that have not been considered yet and consider all the facets of those cofacets that have not been considered yet. This can be done using a queue $\mathcal{Q}$ of simplices to consider. Each of these simplices is queried with the intersection oracle and, if it is found to intersect $\hat{\mathcal{M}}$, it is added to $\mathcal{S}$. Upon termination, $\mathcal{S}$ contains all the k-dimensional simplices of $\mathcal{T}$ that intersect $\hat{\mathcal{M}}$. Each such intersection consists of a single point (by the Genericity hypothesis) and is a vertex of $\hat{\mathcal{M}}$.

Notice that, by construction, our algorithm produces not only the vertices of $\hat{\mathcal{M}}$ but also edges between them (associated to the $(k+1)$-cofaces of the k-simplices in $\mathcal{S}$). In fact, the algorithm outputs the 1-skeleton $\hat{\mathcal{M}}_1$ of the m-dimensional polytopal cell complex $\hat{\mathcal{M}}$. If needed, the full Hasse diagram of $\hat{\mathcal{M}}$ can easily be computed from $\hat{\mathcal{M}}_1$ using the permutahedral representation of $\mathcal{T}$.

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 23. *The time complexity of the algorithm is $O(k2^m I|\mathcal{S}|)$ 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 24 (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 $N_C \leq \frac{K}{m!} \times \left(\frac{d^2\sqrt{d(d+2)}}{2\sqrt{2}D}\right)^m$ and $N_{FK} \leq \frac{K}{m!} \times \left(\frac{d^3}{\sqrt{2}D}\right)^m$ 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^m.
As noticed at the end of Section 3.2, the algorithm computes the 1-skeleton $\hat{\mathcal{M}}_1$ of the polytopal cell complex $\hat{\mathcal{M}}$. It is possible to compute the full Hasse diagram of $\hat{\mathcal{M}}$ by computing the cofaces (of dimension $k+1, ..., d$) of the k-simplices of $\mathcal{S}$ output by the algorithm. This can be done in an output sensitive manner using our data structure. The following lemma shows that the combinatorial complexity of $\hat{\mathcal{M}}$ is of the same order as the combinatorial complexity of its 1-skeleton $\hat{\mathcal{M}}_1$.

Proposition 25. *The combinatorial complexity of $\hat{\mathcal{M}}$ is $|\mathcal{S}| \times (\frac{3}{2})^m (m+1)!$, where $\mathcal{S}$ is bounded in Proposition 24. If $m = O(1)$, the combinatorial complexity of $\hat{\mathcal{M}}$ is polynomial in d and $\delta = 1/D$.*

We combine Propositions 23, 24 and 25 to obtain our main result.

Theorem 26. *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 m. The same result holds for the full PL-approximation $\hat{\mathcal{M}}$ of $\mathcal{M}$.*

3.4 Dimensionality reduction

As seen from Proposition 24, the size $\mathcal{S}$ of the output of the algorithm, considered as a function of the resolution D of the triangulation, depends exponentially on m (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 vectored-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 m but not on d and lead to good approximations are known. Especially important are D-nets [CDR05, BG14]. 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 m-dimensional smooth submanifolds is $O(1/D^m)$ [BCY18, Lemma 5.3]. The sample produced by our algorithm is D-dense but, 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 loose 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 27 (Johnson-Lindenstrauss lemma for manifolds [Cla08, Ver11])**.** *Pick any $\varepsilon, \eta > 0$, and let $d' = \Omega\left(\frac{m}{\varepsilon^2}\log\frac{1}{\varepsilon} + \frac{1}{\varepsilon^2}\log\frac{\Gamma}{\eta}\right)$, 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 $x, y \in \mathcal{M}$*

$$(1-\varepsilon)\sqrt{\frac{d'}{d}} \leq \frac{\|\Phi x - \Phi y\|}{\|x-y\|} \leq (1+\varepsilon)\sqrt{\frac{d'}{d}}.$$

Let $\Psi = \sqrt{\frac{d}{d'}}\,\Phi$. By the theorem, the image $\Psi(\mathcal{M})$ of $\mathcal{M}$ is a submanifold of dimension m 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 $(d'-m)$-simplex σ of the CFK-triangulation of $\mathbb{R}^{d'}$, decides

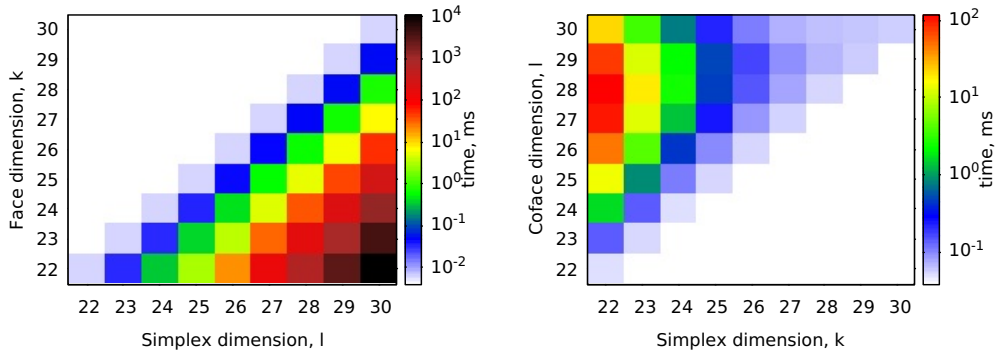

Figure 1: On the left: comparison of the execution time of the face and the coface generation algorithm for simplices of various dimensions in a CFK-triangulation of $\mathbb{R}^{30}$. Because the average computation time of a face or coface is constant, the presented time is proportional to the number of faces or cofaces of respective simplices.

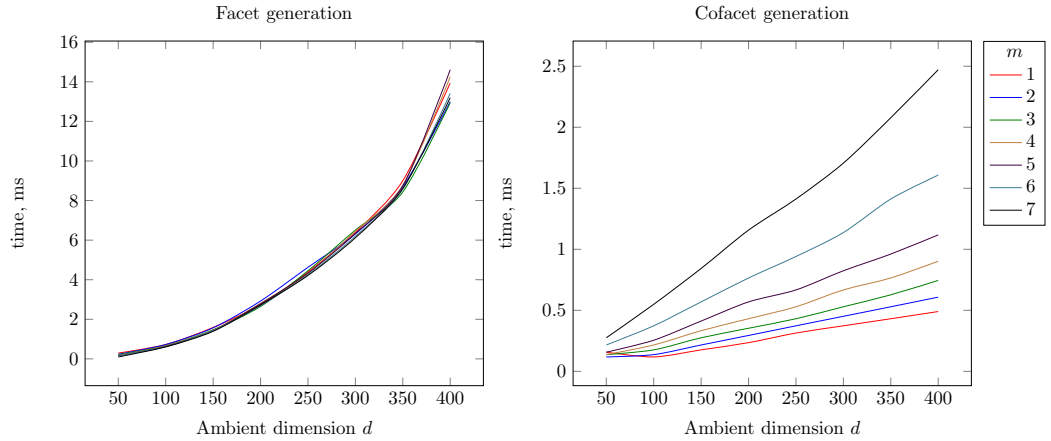

Figure 2: Execution time of the facet and cofacet computation depending on the dimension d of the triangulation and the codimension m of the input simplex.

whether its inverse image $\Psi^{-1}(\sigma)$ intersects $\mathcal{M}$ or not. Note that $\Psi^{-1}(\sigma)$ is a $(d-d')$-dimensional flat strip 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 plane 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 [EW17], the analysis of the algorithm will be unchanged. Proposition 24 then shows that the size of the output sample does not depend on d but only on m and D for fixed ε and η. Moreover since the complexities of the projection and of the new oracle are polynomial in d, Proposition 23 implies that the overall complexity is still polynomial in d.

4 Experimental results

The data structure of Section 2 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 [GUD]. Full detail on the implementation, including the implementation of the oracle, can be found in [Kac19].

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.

Data structure. In Figure 1, we present the time of generating all faces (on the left) and all cofaces (on the right) of various dimensions of simplices in a CFK-triangulation of $\mathbb{R}^{30}$ using algorithms from Section 2.2. The

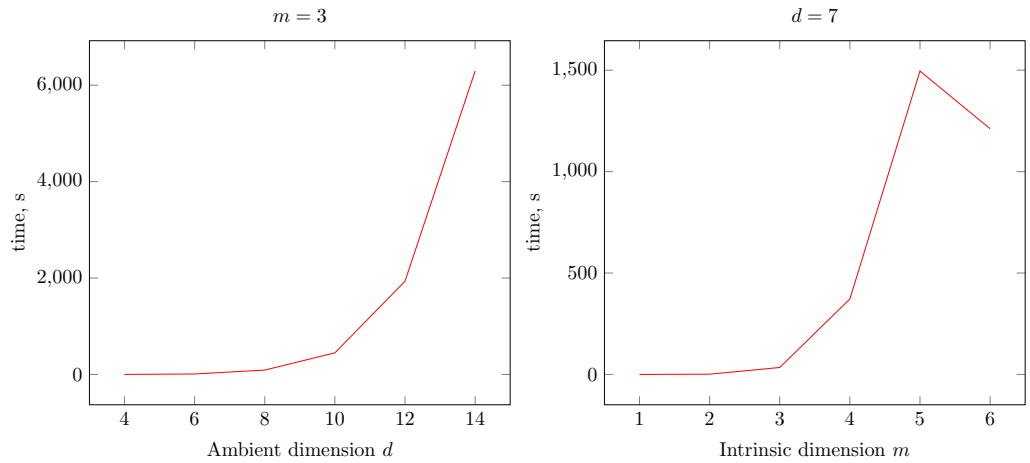

Figure 3: The effect of the ambient dimension d and of the intrinsic dimension m on the computation time of the of the manifold tracing algorithm. The reconstructed manifold in the tests is the m-dimensional sphere embedded in $\mathbb{R}^d$. The ambient triangulation used is a Coxeter triangulation of type $\tilde{A}_d$. The diameter of the full simplices is fixed for all d.

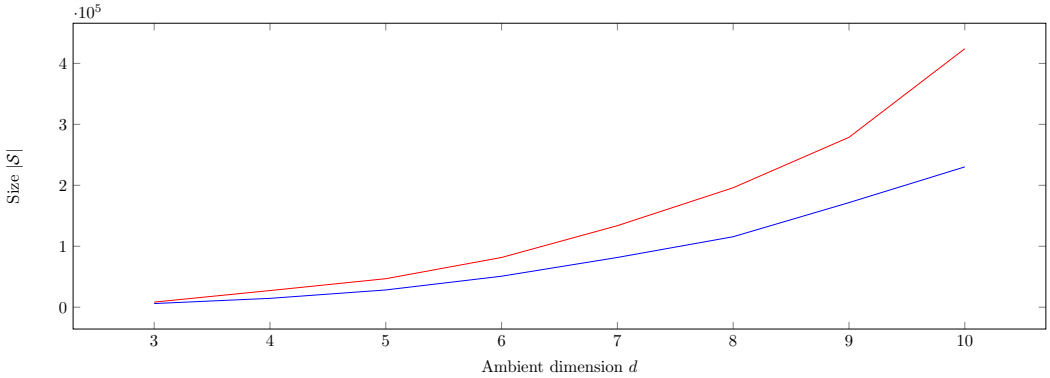

Figure 4: Comparison of the size of the output of the manifold tracing algorithm using two types of ambient triangulations: a Coxeter triangulation of type $\tilde{A}_d$ (in blue) and the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ (in red) with the same diameter $0.07\sqrt{d}$ of d-dimensional 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 $i > 3$.

presented execution time is averaged over 500 tests. Note that both for face and coface generation algorithms, the execution time is proportional to the number of computed elements. On average, these algorithms take time 0.001-0.002 ms per computed face or coface, regardless of the dimensions of the input simplex and of the computed element. In Figure 2, we further illustrate the particular case of facet and cofacet computation, which is essential in the manifold tracing algorithm. We show the dependency of the execution time on two parameters: the ambient dimension d and the codimension m of the input simplex, which corresponds to the intrinsic dimension of the input manifold in the manifold tracing algorithm.

Manifold tracing algorithm. We show the performance of our implementation of the manifold tracing algorithm for various ambient and intrinsic dimensions in Figure 3. In Figure 4, we can see that using Coxeter triangulation is beneficial in practice as it produces a smaller output in less time (see Proposition 24). In Figure 8, we present a PL approximation of a two-dimensional flat torus without boundary embedded in $\mathbb{R}^{10}$ built by the manifold tracing algorithm. The algorithm can be easily adapted to handle submanifolds with boundary. In Figure 9, 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. Both surfaces in Figure 8 and 9 are rotated and translated in their respective ambient spaces for visualization purposes. Note that there is no C^2 embedding of the flat torus in $\mathbb{R}^3$.

5 Further results

In this section, we complement the results of Section 3 in three directions. We first show how to further improve the complexity of the output mesh. Then, we discuss the case of manifolds with boundary and of stratified manifolds. We lastly consider the case of general smooth manifolds.

5.1 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 m with boundary, we mean that, on top of a function $f : \mathbb{R}^d \to \mathbb{R}^{d-m}$, 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 (5), we also define

$$\hat{f}_\partial|_\tau(x) = \sum_{i=1}^{j+1} \lambda_i(x)\, f_\partial(p_i).$$

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 22 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 Lines 6-9: 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 22, 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 $m, m-1, ..., 0$.

5.2 General smooth manifolds

The manifold tracing algorithm 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, to obtain guarantees on the geometric and topological quality of the output mesh, we need to perturb the ambient Coxeter triangulation of type $\tilde{A}_d$ so that the triangulation is $(d-m-1)$-skeleton safe [Whi57, BKW18], whose definition we'll now recall.

Definition 28 (Definition 19 of [BKW18])**.** *We say that an ambient triangulation $\mathcal{T}$ is $(d-m-1)$-*skeleton safe *with respect to the m-dimensional manifold $\mathcal{M}$ if all k-dimensional faces τ in $\mathcal{T}$, with $k \leq d-m-1$, remain far enough from $\mathcal{M}$ while the longest edge length of $\mathcal{T}$ is upper bounded and its smallest thickness is lower bounded. We refer to [BKW18] for details.*

If the triangulation $\mathcal{T}$ is $(d-m-1)$-skeleton safe, we can apply Algogithm 1 and the output mesh is topologically equivalent and close in Hausdorff distance to the input manifold [BKW18]. Enforcing the $(d-m-1)$-skeleton safety can be done using perturbations. However, in the worst case, we need to perturb all the simplices of $\mathcal{T}$ of dimension less than the codimension $d-m$ that are incident on a vertex. Since there are exponentially many such simplices, see Sections 2.1 and 4 of [BKW18], the complexity of such methods is exponential in the ambient dimension d and have not proved useful in practice except in some simple cases. It remains open whether general smooth submanifolds of fixed dimension can be triangulated in polynomial time as we were able to do here in the special case of iso-manifolds.

Acknowledgements

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) and the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 754411. This work has also been 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. We thank Dominique Attali, Guilherme de Fonseca, Arijit Ghosh, Vincent Pilaud and Aurélien Alvarez for their comments and suggestions.

A Proofs

The proof of Corollary 6 is based on:

Lemma 29 (Lemma 3.11 of [MK92])**.** *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 6. 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

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

□

Proof of Corollary 7. 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. □

Proof of Corollary 8. 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 [BY98, Lemma 7.1.14]. Moreover, $|F_{0,\sigma}| \leq (j+1)!$ by Corollary 6. 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 6, 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. □

Proof of Lemma 10. 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

$$z = \sum_{i=0}^{l} \lambda_i v_i = \sum_{i=0}^{l} \lambda_i \sum_{k=1}^{i} \sum_{j \in \omega_i} e_j = \lambda_l \left(\sum_{k \in \omega_l} e_k \right) + (\lambda_l + \lambda_{l-1}) \left(\sum_{k \in \omega_{l-1}} e_k \right) + \cdots + (\lambda_l + \cdots + \lambda_1) \left(\sum_{k \in \omega_1} e_k \right). \quad (6)$$

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

$$\begin{aligned} \alpha_{\omega_l(1)} &= \cdots = \alpha_{\omega_l(m_l)} = \lambda_l \\ &\vdots \\ \alpha_{\omega_1(1)} &= \cdots = \alpha_{\omega_1(m_1)} = \lambda_l + \cdots + \lambda_1 \end{aligned} \quad (7)$$

From (6), 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 (6), for all i. Moreover, because $\lambda_l + \cdots + \lambda_i > \lambda_l + \cdots + \lambda_{i-1}$, we see that (1) is satisfied.
Conversely, given a point z such that its coordinates satisfy (1), we can read of its barycentric coordinates with respect to the v_i, as defined by (2), from (7). □

Proof of Theorem 11. Lemma 10 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 $\mathbb{R}^d$. □

Proof of Lemma 15. Follows by inserting (4) in (3). □

Proof of Lemma 16. 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 14 and Lemma 15, (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 10 and Theorem 11.
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 (2). Then $(v_0, \omega) = (v_0', \omega') \oplus k$ is also a permutahedral representation of the same simplex. □

Proof of Lemma 20. Thanks to Lemma 10, we know that if $x \in \sigma$, where σ is a simplex of dimension less than d, there is at least one equality in (1) 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 10 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 $x \in \sigma$. □

Proof of Lemma 21. 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

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

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}. □

Proof of Proposition 23. 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. From Corollary 7, the number of cofacets is $O(2^m)$. Each of these cofacets has $k+2$ facets. Therefore, for each iteration of the while loop, the algorithm applies the intersection oracle on $O(k2^m)$ simplices. By using this observation and the complexities in Lemmas 18 and 19, the total time complexity of each iteration of the while loop follows:

$$\begin{aligned}O(d2^m) + O(dk2^m) + O(k2^m I) &= O(k2^m(d+I)) \\ &= O(k2^m I).\end{aligned}$$

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

Proof of Proposition 24. By Lemmas 20 and 21, $\mathcal{T}$ 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 m families among the $d(d-1)/2$ families and write I for the associated subset of indices, $I \subset [1, d(d-1)/2], |I| = m$. Now take m hyperplanes, one in each family $H_{u_i}, i \in I$. Their common intersection is an affine space of dimension $k = d - m$. This affine space intersects $\mathcal{M}$ in at most one point under the general position assumption. The total number of intersection points $N_{\mathcal{T}} = \mathcal{T}_k \cap \mathcal{M}$ is thus bounded as follows

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

Here the binomial coefficient arises as the number of choices of m families of hyperplanes.

Consider now more specifically Coxeter triangulation of type $\tilde{A}_d$ and FK-triangulations. We have seen that, for both triangulations, $R_C = R_{FK} = \sqrt{2}$. Moreover, in [CKW20, point 6 of $\tilde{A}_d$ in Section 6], we have seen 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 $L_C < \frac{\sqrt{d+2}}{2}$. It follows from Remark 12, that the longest edge length L_{FK} in the FK-triangulation is $\sqrt{d}$. We then deduce from (8)

$$\begin{aligned}N_C &\leq \frac{1}{m!} \times \left(\frac{d^2\sqrt{d(d+2)}}{2\sqrt{2}D}\right)^m \\ N_{FK} &\leq \frac{1}{m!} \times \left(\frac{d^3}{\sqrt{2}D}\right)^m.\end{aligned}$$

We note that the bound for Coxeter triangulations is about 2^m times better.

Consider now the case of a non-flat manifold $\mathcal{M}$ of dimension m and assume that any affine flat in the hyperplane arrangement $\mathcal{H}_{FK}$ intersects $\mathcal{M}$ in at most K points. Note that generically $K = O(1)$ if $\mathcal{M}$ is an algebraic isomanifold of bounded degree. The previous analysis then remains unchanged except that the bound has to be multiplied by K. □

Proof of Proposition 25. Let σ be a k-simplex of a CFK-triangulation that intersects $\hat{\mathcal{M}}$ and let σ^* be its dual cell (see Section 2.2). 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 17). Hence, σ^* is a m-face of a d-permutahedron. The number of faces of σ^* of dimensions 0 to $m-1$ (or equivalently the number of cofaces of σ of dimensions $m+1$ to d) is (using Mathematica):

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

The overall combinatorial complexity of $\hat{\mathcal{M}}$ is therefore

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

where $\mathcal{S}$ is bounded in Proposition 24. □

B Figures

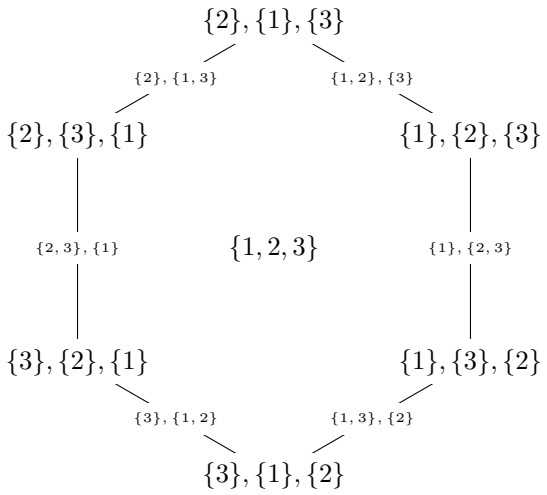

Figure 5: The 2-permutahedron and the ordered partitions associated to its faces.

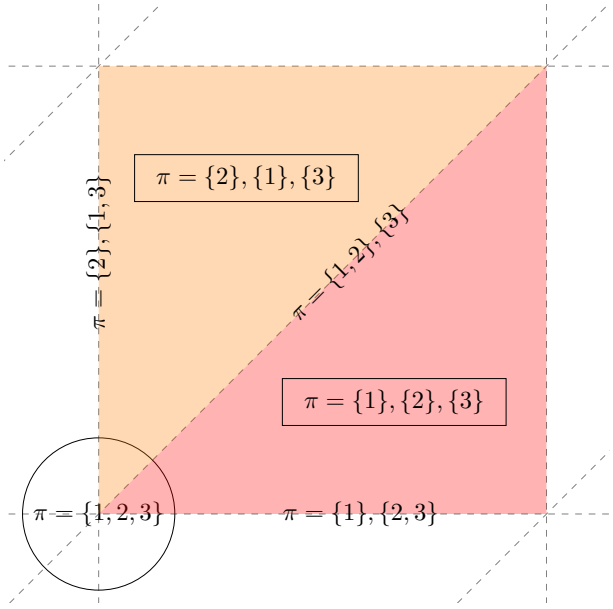

Figure 6: The ordered partitions associated to the faces of the FK-triangulation of $\mathbb{R}^2$ that have the same minimal vertex v_0 (circled).

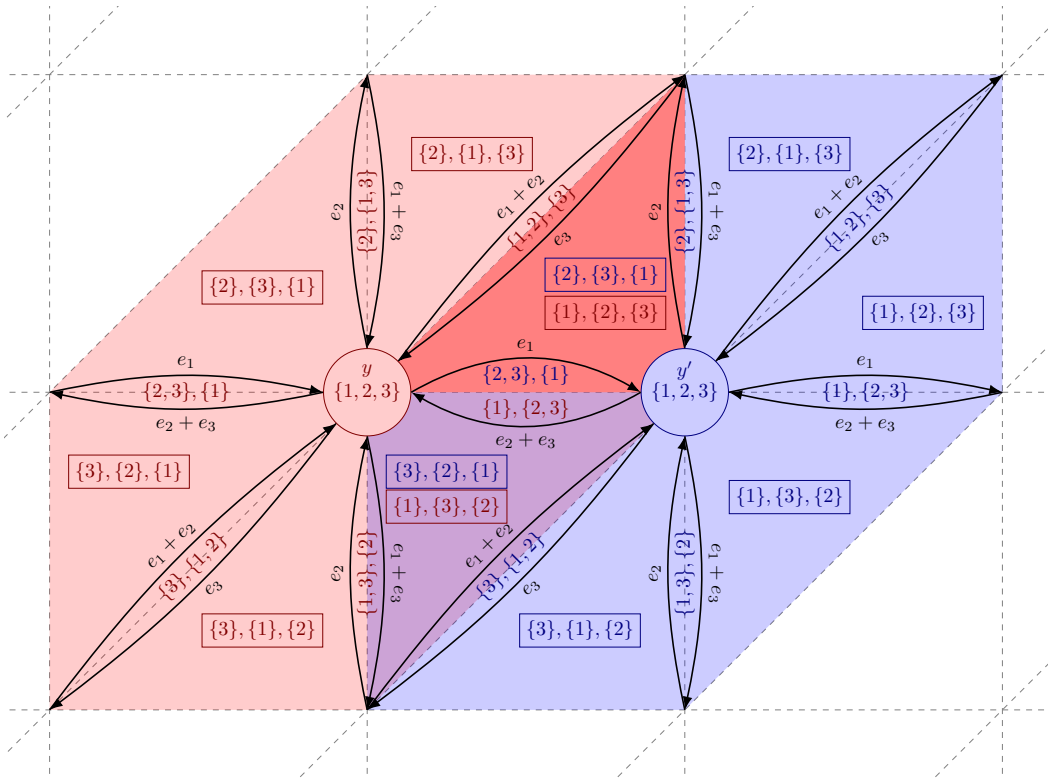

Figure 7: The permutahedral representation of the simplices in the stars of vertices y and y'.

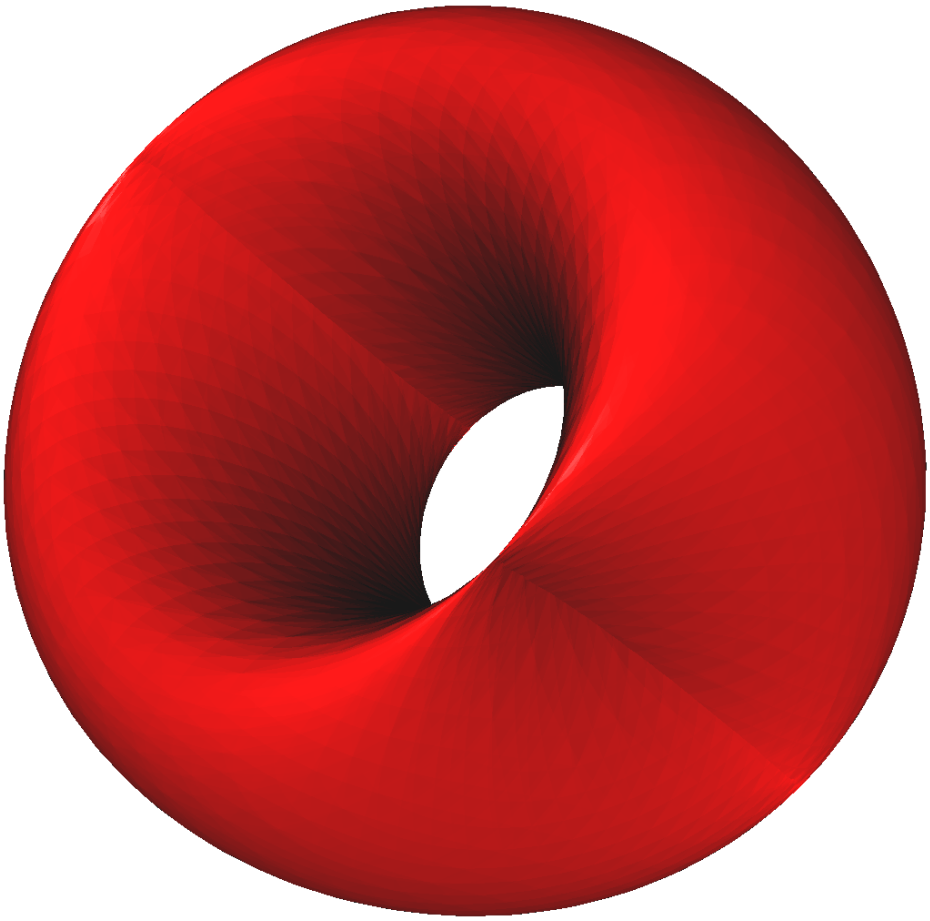

Figure 8: The piecewise-linear approximation of a flat torus embedded in $\mathbb{R}^{10}$ defined by the equations $x_1^2 + x_2^2 = 1$ and $x_3^2 + x_4^2 = 1$ and $x_i = 0$ for $i > 4$, projected to $\mathbb{R}^3$. The ambient triangulation used is a Coxeter triangulation of type $\tilde{A}_{10}$ with the diameter of the full-dimensional simplices 0.23. The output size $|\mathcal{S}|$ is 509952. The execution time of the algorithm is 231s.

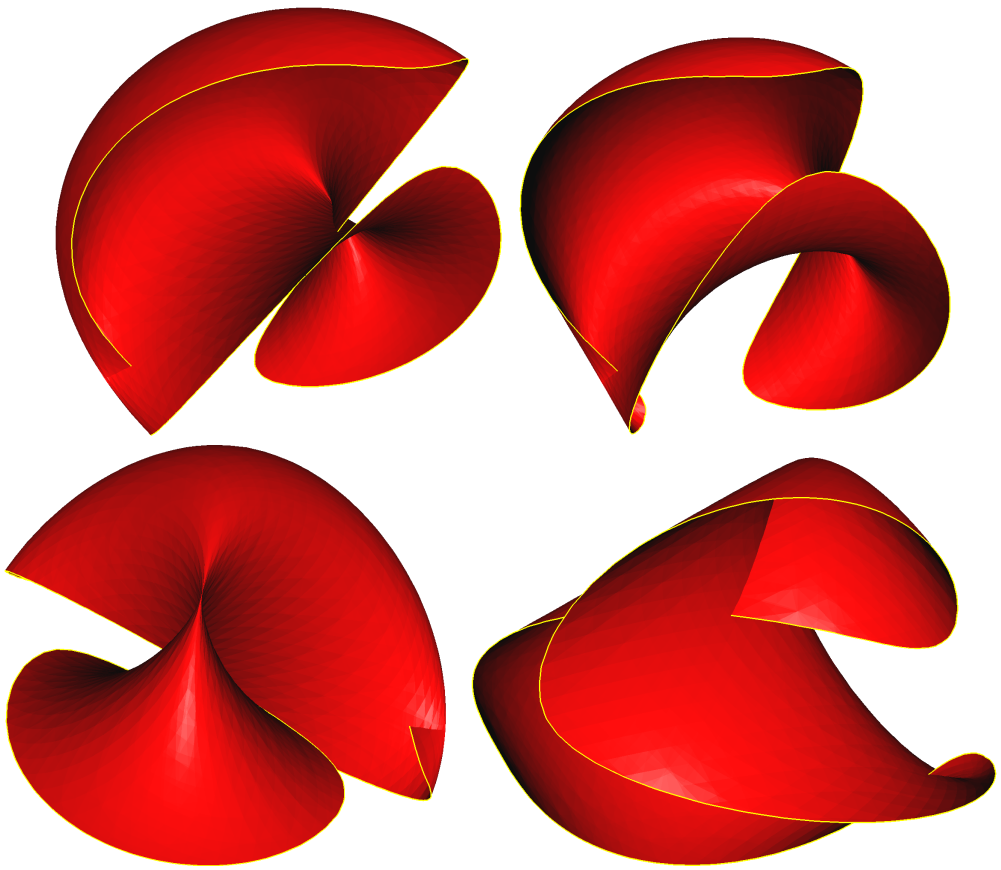

Figure 9: Four views of the flat torus in $\mathbb{R}^4$ given by two equations $x_1^2 + x_2^2 = 1$ and $x_3^2 + x_4^2 = 1$ cut by the hypersphere $(x_1 - 1)^2 + x_2^2 + (x_3 - 1)^2 + x_4^2 = 4$, projected to $\mathbb{R}^3$. The ambient triangulation used is a Coxeter triangulation of type $\tilde{A}_4$ 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 14779. The execution time of the algorithm is 1.84s.

References

[AG89] Eugene Allgower and Kurt Georg. Estimates for piecewise linear approximations of implicitly defined manifolds. *Applied Mathematics Letters*, 2(2):111–115, 1989.

[AG90] Eugene Allgower and Kurt Georg. *Numerical continuation methods: an introduction*, volume 13. Springer Science & Business Media, 1990.

[AS85] 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.

[BCSM+06] Jean-Daniel Boissonnat, David Cohen-Steiner, Bernard Mourrain, Günter Rote, and Gert Vegter. Meshing of surfaces. In Jean-Daniel Boissonnat and Monique Teillaud, editors, *Effective Computational Geometry for Curves and Surfaces*, pages 181–229. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

[BCY18] Jean-Daniel Boissonnat, Frédéric Chazal, and Mariette Yvinec. *Geometric and Topological Inference*. Cambridge Texts in Applied Mathematics. Cambridge University Press, 2018.

[BG14] Jean-Daniel Boissonnat and Arijit Ghosh. Manifold reconstruction using tangential Delaunay complexes. *Discrete & Computational Geometry*, 51(1):221–267, 2014.

[BKW18] Jean-Daniel Boissonnat, Siargey Kachanovich, and Mathijs Wintraecken. Triangulating submanifolds: An elementary and quantified version of Whitney's method. hal-01950149, December 2018.

[BW20] Jean-Daniel Boissonnat and M Wintraecken. The Topological Correctness of PL-Approximations of Isomanifolds. working paper or preprint, June 2020.

[BY98] Jean-Daniel Boissonnat and Mariette Yvinec. *Algorithmic Geometry*. Cambridge Texts in Applied Mathematics. Cambridge University Press, 1998.

[Cai34] S. S. Cairns. On the triangulation of regular loci. *Annals of Mathematics. Second Series*, 35(3):579–587, 1934.

[CDR05] Siu-Wing Cheng, Tamal K Dey, and Edgar A Ramos. Manifold reconstruction from point samples. In *SODA*, volume 5, pages 1018–1027, 2005.

[CDS12] Siu-Wing Cheng, Tamal K Dey, and Jonathan Shewchuk. *Delaunay mesh generation*. CRC Press, 2012.

[Che20] Yen-Chi Chen. Solution manifold and its statistical applications, 2020. arXiv:2002.05297.

[CKW20] Aruni Choudhary, Siargey Kachanovich, and Mathijs Wintraecken. Coxeter triangulations have good quality. *Mathematics in Computer Science*, 14:141176, 2020.

[Cla08] 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.

[DK04] 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.

[DWLT90] 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.

[Eav84] B. Curtis Eaves. *A course in triangulations for solving equations with deformations*, volume 234. Lecture Notes in Economics and Mathematical Systems, 1984.

[Ehr73] Gideon Ehrlich. Loopless algorithms for generating permutations, combinations, and other combinatorial configurations. *Journal of the ACM (JACM)*, 20(3):500–513, 1973.

[EW17] Armin Eftekhari and Michael B. Wakin. What happens to a manifold under a bi-lipschitz map? *Discrete & Computational Geometry*, 57(3):641–673, 2017.

[Fre42] Hans Freudenthal. Simplizialzerlegungen von beschrankter flachheit. *Annals of Mathematics*, pages 580–582, 1942.

[GUD] GUDHI Project "http://gudhi.gforge.inria.fr/doc/latest/" .

[GVJ+09] A. Gomes, I. Voiculescu, J. Jorge, B. Wyvill, and C. Galbraith. *Implicit Curves and Surfaces: Mathematics, Data Structures and Algorithms*. Springer, 2009.

[Kac19] Siargey Kachanovich. *Manifold meshing using Coxeter triangulations*. PhD thesis, Université Côte d'Azur, 2019.

[LC87] 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.

[Min03] Chohong Min. Simplicial isosurfacing in arbitrary dimension and codimension. *Journal of Computational Physics*, 190(1):295–310, 2003.

[MK92] M. Maes and B. Kappen. On the permutahedron and the quadratic placement problem. *Philips Journal of Research*, 46(6):267–292, 1992.

[NY06] Timothy S. Newman and Hong Yi. A survey of the marching cubes algorithm. *Computers & Graphics*, 30(5):854 – 879, 2006.

[OR00] James M Ortega and Werner C Rheinboldt. *Iterative solution of nonlinear equations in several variables*. SIAM, 2000.

[Ost16] 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.

[RD69] B.C. Rennie and A.J. Dobson. On Stirling numbers of the second kind. *Journal of Combinatorial Theory*, 7(2):116 – 121, 1969.

[RS94] Frank Ruskey and Carla D. Savage. Gray codes for set partitions and restricted growth tails. *Australasian J. Combinatorics*, 10:85–96, 1994.

[Tod76] Michael J. Todd. *The computation of fixed points and applications*, volume 124. Lecture Notes in Economics and Mathematical Systems, 1976.

[Ver11] Nakul Verma. A note on random projections for preserving paths on a manifold. Technical Report Tech. Report CS2011-0971, UC San Diego, 2011.

[Wal00] Timothy R. Walsh. Loop-free sequencing of bounded integer compositions. *Journal of Combinatorial Mathematics and Combinatorial Computing*, 33:323–345, 2000.

[Wen13] Rephael Wenger. *Isosurfaces: geometry, topology, and algorithms*. AK Peters/CRC Press, 2013.

[Whi40] J. H. C. Whitehead. On C^1-complexes. *Annals of Mathematics*, 41(4):809–824, 1940.

[Whi57] H. Whitney. *Geometric Integration Theory*. Princeton University Press, 1957.

[Zie12] G. M. Ziegler. *Lectures on Polytopes*. Graduate Texts in Mathematics. Springer New York, 2012.