

TRACING ISOMANIFOLDS IN \mathbb{R}^d IN TIME POLYNOMIAL IN d USING COXETER–FREUDENTHAL–KUHN TRIANGULATIONS*

JEAN-DANIEL BOISSONNAT[†], SIARGEY KACHANOVICH[‡], AND
MATHIJS WINTRAECKEN^{†§}

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 $f: \mathbb{R}^d \rightarrow \mathbb{R}^{d-n}$, where n is the intrinsic dimension of the manifold. A natural way to approximate a smooth isomanifold $\mathcal{M} = f^{-1}(0)$ 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 $\hat{\mathcal{M}}$ 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.

Key words. Coxeter triangulation, Freudenthal–Kuhn triangulation, permutahedron, PL-approximations, isomanifolds/solution manifolds/isosurfacing

MSC codes. 65D18, 68U05

DOI. 10.1137/21M1412918

1. Introduction. Given a surface represented in \mathbb{R}^3 as the zero set of a function $f: \mathbb{R}^3 \rightarrow \mathbb{R}$, with 0 regular value,¹ 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 $f: \mathbb{R}^d \rightarrow \mathbb{R}^{d-n}$. Isosurfaces play a crucial role in medical imaging, computer graphics, and geometry processing [35]. Higher dimensional isomanifolds are also of fundamental importance in many fields like statistics [15], dynamical systems [40], econometrics, or mechanics [35].

* Received by the editors April 16, 2021; accepted for publication (in revised form) November 7, 2022; published electronically April 12, 2023. The results of this paper have been presented at the Symposium on Computational Geometry, 2021.

<https://doi.org/10.1137/21M1412918>

Funding: The authors have received funding from the European Research Council under the European Union’s ERC grant agreement 339025 GUDHI (Algorithmic Foundations of Geometric Understanding in Higher Dimensions). The first author was 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 ANR-19-P3IA-0002. The third author was supported by the European Union’s Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement 754411 and the FWF (Austrian Science Fund) grant M 3073.

[†]INRIA Sophia Antipolis - Méditerranée, 06902 Sophia Antipolis, France, and Université Côte d’Azur, France (jean-daniel.boissonnat@inria.fr, <https://www-sop.inria.fr/members/Jean-Daniel.Boissonnat/>).

[‡]INRIA Sophia Antipolis - Méditerranée, 06902 Sophia Antipolis, France (siargey.kachanovich@gmail.com).

[§]IST Austria, Klosterneuburg 3400, Austria (mathijs.wintraecken@inria.fr).

¹We refer readers to section 3.1 for a precise definition.

State-of-the-art. The most widely used algorithm to trace isomanifolds is the Marching Cube (MC) algorithm and its numerous variants [28, 43]. The MC algorithm uses a cubical grid to tessellate the ambient space. Extensions of the MC algorithm to higher dimensions have been proposed [43]. A nice property of uniform grids is that, due to their simple structure, they don't need to be represented explicitly. This is a major advantage since the combinatorial complexity of the grid (restricted to the cube) is $\Omega(\delta^d)$, where $\delta = 1/D$ and D is the diameter of a d -cell of the 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 [43]. A natural way to circumvent this difficulty is to use a triangulation of the ambient space instead of a grid, which led to the development of a variant of the MC algorithm named the Marching Tetrahedra algorithm. This is, however, not enough to circumvent the curse of dimensionality and we need triangulations of the ambient space that can be represented implicitly as it can be done with grids. This is impossible for general triangulations but doable using highly regular triangulations to subdivide the ambient space \mathbb{R}^d . Some early work along this direction has been published in applied mathematics [2, 24, 40], and a slightly more recent paper by Dobkin et al. [22] 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 [34]. 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 notation, $O(\delta^n \log \delta)$, 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 analyzed 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 4 dimensions.

Contributions. This paper discusses an efficient algorithm to compute a PL-approximation of isomanifolds. We extend the work of Dobkin et al. [22] 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. The properties of this family of triangulations is discussed in Part I of this paper. Freudenthal–Kuhn triangulations can (and have been) applied in many other contexts, see e.g., [24, 33, 40], and therefore Part I is of independent interest. Contrary to Min [34], 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 (section 2.3). The data structure is very compact and allows us to retrieve the faces or the cofaces of a simplex of any dimension in an output sensitive way.

Subsequently, in Part II of the paper (section 3), we show how to use this data structure to trace a connected submanifold of \mathbb{R}^d , starting from a given initial point on the manifold. Our algorithm produces a PL-approximation of size polynomial in d and $\delta = 1/D$, and exponential in n . Here D is the diameter, that is, the length of the longest edge in the ambient triangulation. 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 [12] and recalled in section 3.3, 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-approximations 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 3.5). We can also extend the algorithm to the case of isomanifolds with boundary and, more generally, to stratifolds (section 3.6).

The algorithm has been implemented. In Part III of the paper (section 4), we report on experimental results which show that the algorithm is practical and can handle cases that are far ahead of the state-of-the-art. We also present an application of the tracing algorithm 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 [44] introduced a construction that has some similarity to 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. Part I: Coxeter–Freudenthal–Kuhn triangulations.

2.1. Introduction and state-of-the-art. 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 [7] 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, i.e., they form a special class of triangulations of \mathbb{R}^d that have a high regularity. The

data structure to represent such triangulations is almost as compact as for grids and allows for efficient face and coface computation.

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 [21] and Freudenthal [27]. 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 [22] 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 interpolating multivariate multivalued functions or in meshing geometric shapes embedded in high dimensional spaces. Freudenthal–Kuhn triangulations have been known in applied mathematics [2, 24, 40], and Coxeter triangulations have been used by Dobkin et al. [22] to trace curves in high dimensions and are good candidates to trace manifolds of any codimension [34]. They have also been used in the context of topological data analysis [18].

In section 2.2, 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 [22, 24, 27, 33, 38, 40, 45]).

Based on these results, we introduce in section 2.3 a very compact data structure that implicitly stores the full facial structure of such triangulations. The data structure allows us 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 4.2 reports on experimental results and demonstrates that the data structure is remarkably efficient and practical. It is especially useful in tracing low dimensional manifolds embedded in high dimensional spaces as encountered in statistics, dynamical systems, econometrics, or mechanics [15, 35, 40].

2.2. Definitions. 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, to be introduced in section 2.2.3, have a different flavor 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. [22], 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.

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

DEFINITION 2.1 (permutahedron). *A d -permutahedron is a d -dimensional polytope, which is the convex hull \mathcal{P} of all $((d+1)!) points in \mathbb{R}^{d+1} , the coordinates of which are permutations of $[d+1]$. Formally, this convex hull can be written as$*

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

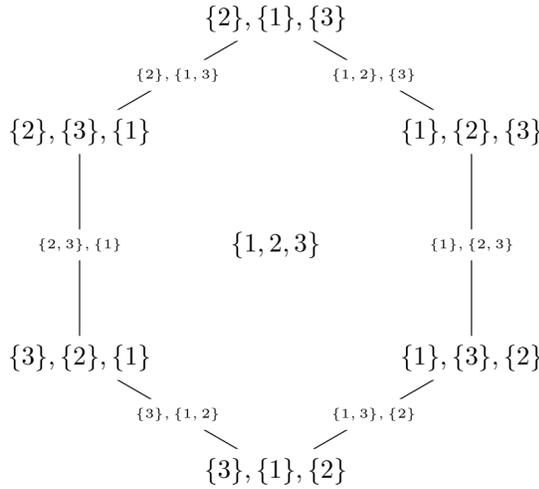


FIG. 1. The 2-permutahedron and the ordered partitions associated to its faces. The edges correspond to ordered partitions of length 2 and the vertices to those of length 3, that is, permutations.

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

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

where x^i denotes the i th Euclidean coordinate. 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, [33, Lemma 3.4]).

The following observation follows from duality (see the paragraph “Duality” in section 2.3.1) and a result in section 2.2.2.

Remark 2.2. The d -permutahedron is a simple polytope.

The facial structure of \mathcal{P} is best described in terms of ordered partitions [45]. Refer to Figure 1.

DEFINITION 2.3 (ordered partition). *Let T be a finite nonempty 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 $OP_l[d + 1]$ for the set of ordered partitions of $[d + 1]$ with l parts and just $OP[d + 1]$ for the set of all ordered partitions of $[d + 1]$.*

DEFINITION 2.4 (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, \dots, a_l , such that*

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

We recall Theorem 3.6 of [33].

LEMMA 2.5 (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 $(\omega_1, \dots, \omega_l)$. 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 [33, Corollary 3.15] and [38, Theorem 3].

COROLLARY 2.6. *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 $2^{2(d+1)\log(i+1)}$.*

Corollaries 2.7, 2.9, and 2.10 in the following seem to be new.

COROLLARY 2.7. *The number $p_{0,i}$ of vertices of an i -face of a d -permutahedron is at most $(i+1)!$ and at least $2^{\min(i, d-i+1)}$.*

The proof of Corollary 2.7 is based on the following lemma.

LEMMA 2.8 (Lemma 3.11 of [33]). *The face of a permutahedron corresponding to an ordered partition $\omega = (\omega_1, \dots, \omega_{l+1})$ is combinatorially*

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

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

Proof of Corollary 2.7. Write $l = d-i$. Since the number of vertices of the product of two polytopes is the product of the numbers of their vertices, and an $(n-1)$ -dimensional permutahedron has $n!$ vertices, we see that the total number of vertices of an i -face of a d -dimensional permutahedron corresponding to an ordered partition $\omega = (\omega_1, \dots, \omega_{l+1})$ 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 is as small as possible, that is, when 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, $r' = d+1 \bmod l+1$; 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 $\lfloor \frac{d+1}{l+1} \rfloor \geq 2$,

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

- If $\lfloor \frac{d+1}{l+1} \rfloor = 1$, we have $r' = d - l$. Hence

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

In the first case, $\frac{d+1}{2} \geq l+1$, or equivalently $l+1 \leq d-l$. In the second case, $r' = d-l$ and thus $d-l < l+1$. Hence, in both cases, we have

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

COROLLARY 2.9. *The number of facets of an i -face σ of a d -permutahedron is at most $2^{i+1} - 2$.*

Proof. Write $l = d - i$. We first recall a set of $m > 2$ objects can be subdivided in two nonempty ordered subsets A and B in $2^m - 2$ ways. This is not hard to see. We choose for each element whether it will be put in A or B , there are 2^m possibilities. Excluding that A or B is empty gives $2^m - 2$. Let $\omega = (\omega_1, \dots, \omega_l)$ 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. \square

COROLLARY 2.10. *Let $p_{i,j}$ denote the number of i -faces of a j -face of the d -permutahedron. We have*

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

Corollary 2.10 generalizes the previous two corollaries. For $i = 0$, the bound in Corollary 2.10 is the same as the upper bound in Corollary 2.7. For $i = j - 1$, the bound is comparable but weaker than the bound in Corollary 2.9.

Proof of Corollary 2.10. 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

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

As we have seen in Remark 2.2 the d -permutahedron is a simple polytope. 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 [13, Lemma 7.1.14]. Moreover, $|F_{0,\sigma}| \leq (j+1)!$ by Corollary 2.7. 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 2.7, we have

$$\sum_{\tau \in F_{i,\sigma}} p_\tau \geq 2^{\min\{i,d-i+1\}} |F_{i,\sigma}|.$$

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

2.2.2. Freudenthal–Kuhn triangulation. The Freudenthal–Kuhn (FK) 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 [27], Todd [40], or Eaves [24], mainly for top dimensional simplices and in different guises. We combine these results and extend to simplices of arbitrary (co)dimension, where necessary.

DEFINITION 2.11. Let $x \in \mathbb{R}^d$, and write $z^i = x^i - \lfloor x^i \rfloor$. We denote by e_1, \dots, 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 $\omega = (\omega_1, \dots, \omega_{l+1})$ of $[d+1]$, where the ω_i are obtained by sorting the z^i in decreasing order. Specifically, with $\omega_i = \{\omega_i(1), \dots, \omega_i(m_i)\}$, we have (see Figure 2)

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

We stress that by definition $d+1 \in \omega_{l+1}$.

LEMMA 2.12. Suppose that $\omega = (\omega_1, \dots, \omega_{l+1})$ is an ordered partition of $[d+1]$ such that $d+1 \in \omega_{l+1}$, and let $\sigma = \{v_0, \dots, v_l\}$ be the l -simplex whose vertices are the points

$$(2) \quad v_0 = (\lfloor x^1 \rfloor, \dots, \lfloor x^d \rfloor), \quad v_i = v_{i-1} + E_{\omega_i}, \quad i = 1, \dots, l,$$

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

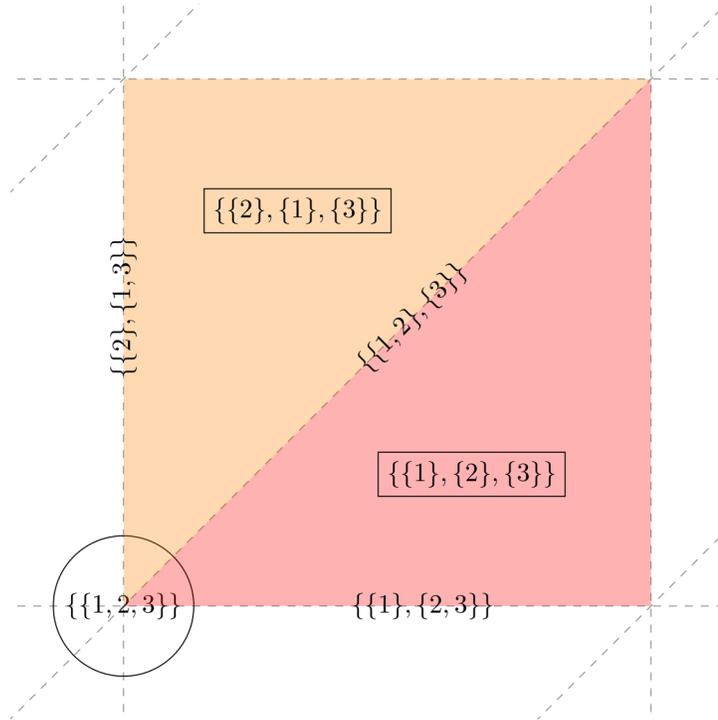


FIG. 2. The ordered partitions associated to the faces of the FK-triangulation of \mathbb{R}^2 that have the same minimal vertex v_0 (circled). We go over some examples: First we consider a vertex, of which there is only one, namely the circled one. Because the circled vertex has coordinates $z^1 = z^2 = 0$ and by convention $z^3 = 0$, it is associated to the ordered partition $\{\{1, 2, 3\}\}$. Second, let us consider the (interior of the) diagonal edge. Because for any point (z^1, z^2) in this edge we have $1 > z^1 = z^2 > 0$, the first part of the associated ordered partition is $\{1, 2\}$. By convention $z^3 = 0$ and therefore the second part of the partition is $\{3\}$, so that the complete ordered partition is $\{\{1, 2\}, \{3\}\}$. Finally let's consider the lower triangle, where the points in the relative interior are given by $1 > z^1 > z^2 > 0$. Because $z^1 > z^2$, we have that $\{1\}$ and $\{2\}$ are in different parts of the partition and $\{1\}$ precedes $\{2\}$. Because $z^2 > 0$ and again $z^3 = 0$ by convention, we have that $\{2\}$ and $\{3\}$ are in different parts of the partition and $\{2\}$ precedes $\{3\}$. Therefore the complete ordered partition is $\{\{1\}, \{2\}, \{3\}\}$.

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

$$\begin{aligned}
 z &= \sum_{i=0}^l \lambda_i v_i \\
 &= \sum_{i=0}^l \lambda_i \sum_{k=1}^i E_{\omega_k} \\
 (3) \quad &= \lambda_l E_{\omega_l} + (\lambda_l + \lambda_{l-1}) E_{\omega_{l-1}} + \cdots + (\lambda_l + \cdots + \lambda_1) E_{\omega_1},
 \end{aligned}$$

where the $\lambda_i > 0$ (with $i \in [0, l]$ and $\sum_{i=0}^l \lambda_i = 1$) are the barycentric coordinates of z in σ . We define

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

With this definition (3) can be rewritten as

$$\begin{aligned}
 z &= \alpha_{\omega_l(1)}e_{\omega_l(1)} + \cdots + \alpha_{\omega_l(m_l)}e_{\omega_l(m_l)} \\
 &\quad + \alpha_{\omega_{l-1}(1)}e_{\omega_{l-1}(1)} + \cdots + \alpha_{\omega_{l-1}(m_{l-1})}e_{\omega_{l-1}(m_{l-1})} \\
 &\quad + \dots \\
 (5) \quad &\quad + \alpha_{\omega_1(1)}e_{\omega_1(1)} + \cdots + \alpha_{\omega_1(m_1)}e_{\omega_1(m_1)}.
 \end{aligned}$$

We therefore see that $\alpha_{\omega_i(j)}$ is the $\omega_i(j)$ th coordinate of z , denoted by $z^{\omega_i(j)}$, while all coordinates $z^{\omega_{l+1}(1)}, \dots, z^{\omega_{l+1}(m_{l+1})}$ are zero, because $e_{\omega_{l+1}(i)}$ does not occur in (5), 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 off its barycentric coordinates with respect to the v_i , as defined by (2), from (4). \square

THEOREM 2.13. *To each point $x \in \mathbb{R}^d$ we can associate an element of \mathbb{Z}^d , namely $(\lfloor x^i \rfloor)$, and an ordered partition as defined in Definition 2.11. The equivalence relation that identifies points in \mathbb{R}^d if they are associated to the same element in \mathbb{Z}^d and the same ordered partition yields equivalence classes. Each equivalence class is a simplex in a triangulation of \mathbb{R}^d . This triangulation is called the FK-triangulation.*

In the next section we'll see that this triangulation can be viewed as a hyperplane arrangement.

Proof of Theorem 2.13. Lemma 2.12 implies the following:

- 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 lies 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 . \square

Remark 2.14. We note that, by construction, v_0 in Lemma 2.12 is the smallest vertex of σ in the lexicographical order. The vertex v_0 is therefore called the minimal vertex of σ . Lemma 2.12 also implies an observation of Freudenthal [27]: all d -simplices in the FK-triangulation can be described by monotone paths along the edges of the cube from vertex $(0, \dots, 0) + v_0$ to vertex $(1, \dots, 1) + v_0$. By monotone we mean that all the coordinates are nondecreasing. Conversely, any monotone path along the edges of the cubes from $(0, \dots, 0) + v_0$ to $(1, \dots, 1) + v_0$ gives a simplex in the FK-triangulation.

2.2.3. CFK-triangulations. Freudenthal–Kuhn triangulations are closely related to Coxeter triangulations of type \tilde{A}_d [17], and both are arrangements of hyperplanes as demonstrated below.

Let E be a finite set of vectors of \mathbb{R}^d , and consider the set of hyperplanes $H_E = \{x \in \mathbb{R}^d \mid \langle x, u \rangle = k, u \in E, k \in \mathbb{Z}\}$. We call the set E the set of roots. This is to conform to the terminology used in the theory of Coxeter triangulations.

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

LEMMA 2.15. *The Freudenthal–Kuhn triangulation is the hyperplane arrangement $\mathcal{H}_{E_{FK}}$ associated to the set of vectors $E_{FK} = \{e_1, \dots, e_d\} \cup \{u_{i,j} = e_j - e_i \mid 1 \leq i < j \leq d\}$.*

Proof. By Definition 2.11, Lemma 2.12 and Theorem 2.13, $x \in \mathbb{R}^d$ lies in some simplex σ given by v_0 and ω if and only if

$$(6) \quad \begin{aligned} 1 &> x^{\omega_1(1)} - v_0^{\omega_1(1)} = \dots = x^{\omega_1(m_1)} - v_0^{\omega_1(m_1)} > \dots \\ &> x^{\omega_l(1)} - v_0^{\omega_l(1)} = \dots = x^{\omega_l(m_l)} - v_0^{\omega_l(m_l)} \\ &> x^{\omega_{l+1}(1)} - v_0^{\omega_{l+1}(1)} = \dots = x^{\omega_{l+1}(m_{l+1})} - v_0^{\omega_{l+1}(m_{l+1})} = 0, \end{aligned}$$

where the $v_0^j \in \mathbb{Z}$. We see that $x^i - v_0^i = x^j - v_0^j$ rewrites into $x^i - x^j = \langle x, u_{i,j} \rangle = v_0^i - v_0^j \in \mathbb{Z}$ and $x^i - v_0^i = 0$ rewrites into $x^i = \langle x, e_i \rangle = v_0^i \in \mathbb{Z}$. Similar statements hold when the equalities are replaced by inequalities. Moreover, $1 > x^i - v_0^i = x^j - v_0^j > 0$ further translates into $1 > x^i - v_0^i - (x^j - v_0^j)$ and thus $v_0^i - v_0^j + 1 > x^i - x^j = \langle x, u_{i,j} \rangle$. Note that (6) contains as many (in)equalities as E_{FK} hyperplanes if one also counts nonconsecutive (in)equalities. This means that every simplex in an FK-triangulation can be written as a cell in the hyperplane arrangement.

Conversely, let x be a point in the interior of a d -dimensional cell in the hyperplane arrangement $\mathcal{H}_{E_{FK}}$; then $\langle x, e_i \rangle \notin \mathbb{Z}$ and $\langle x, u_{i,j} \rangle \notin \mathbb{Z}$. This implies that x lies in the interior of a simplex in the FK-triangulation. If we walk from x in a direction $s \in S^{d-1}$, we leave the cell as soon as $\langle x + \lambda s, e_i \rangle \in \mathbb{Z}$ or $\langle x + \lambda s, u_{i,j} \rangle \in \mathbb{Z}$. This point must lie on a lower (at least $d-1$) dimensional face of the FK-triangulation. This implies that a d -dimensional cell in the hyperplane arrangement coincides with a d -simplex in the FK-triangulation. Because the lower dimensional cells and simplices are the intersections of higher dimensional cells and simplices, the result follows. \square

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 parallel 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 the 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 $s_i = e_i - e_{i+1}$, $i = 1, \dots, d$. The vectors s_j are called simple roots and play an important role in algebra. We refer the reader to [17] for more information.

LEMMA 2.16. μ maps E_{FK} bijectively onto the set E_C defined as

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

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} . \square

Observe that all vectors in E_C have length $\sqrt{2}$. By definition, this implies that the distance between the two parallel 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 FK-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

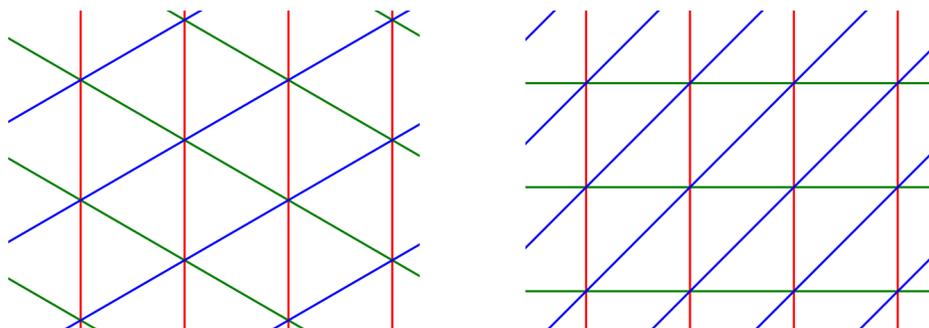


FIG. 3. The Coxeter and Freudenthal–Kuhn triangulations in the plane.

the *Coxeter triangulation* of type \tilde{A}_d of \mathbb{R}^d . See Figure 3 for a depiction of the Coxeter triangulation and the FK-triangulation of the plane. By definition, it has the same combinatorial structure as the FK-triangulation. In addition, it has remarkable geometric properties [22, 17]. First, it is a nondegenerate 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 an FK-triangulation under a nondegenerate 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).

2.3. Data structure. We introduce our data structure in this section. We first consider the case of FK-triangulations in sections 2.3.1 and 2.3.2. The extension to CFK-triangulations in section 2.3.3 is straightforward, since all these triangulations have the same combinatorial structure.

2.3.1. Permutahedral representation of FK-triangulations.

Cycles and the permutahedron. In Remark 2.14 we have seen that simplices can be described by monotone paths (increasing coordinates) along the edges of the cube. As observed by Eaves [24], 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

$$v_0 = v_l + E_{\omega_{l+1}},$$

with $E_{\omega_{l+1}} = \sum_{i \in \omega_{l+1}} e_i$ as before, and ω as in Definition 2.11. Until now, we took $d + 1 \in \omega_{l+1}$, but this will be changed in the following as discussed now. Because the path is now thought of as 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 no longer necessary that $d + 1$ lies in ω_{l+1} , and by an (arbitrary) starting point. We now formalize these general cyclical paths.

DEFINITION 2.17 (permutahedral representation). *Let $(v_0, \omega) \in \mathbb{Z}^d \times OP_{l+1}[d+1]$. To this pair we associate a simplex $\sigma^\omega = \{v_0 = v_0^\omega, v_1^\omega, \dots, v_l^\omega\}$ with*

$$(7) \quad v_i^\omega = v_{i-1}^\omega + E_{\omega_i}, \quad i = 1, \dots, l.$$

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 2.20, and Proposition 2.21, 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 2.18 (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*

$$(8) \quad v_0' = v_0 + \sum_{j=1}^k \sum_{i \in \omega_j} e_i, \quad \omega'_j = \omega_{(j+k) \bmod (l+1)}.$$

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

Proof. The proof follows by inserting (8) in (7). \square

We now prove that 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 2.20. *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 the 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 2.18, and Lemma 2.19, (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 2.12 and Theorem 2.13.

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. \square

Faces. From (7) it is clear that merging two consecutive parts in the ordered partition $\omega = (\omega_1, \dots, \omega_{l+1})$ corresponds to removing a vertex from the simplex. Hence the resulting simplex is a facet of the original simplex. Here we stress that we are allowed to merge ω_1 and ω_{l+1} , but in that case we have to change the base point of the cycle to $v_0 + \sum_{i \in \omega_1} e_i$ to obtain the canonical representation. Looking at the two-dimensional example in Figure 4, we see that the red triangle, with permutahedral representation $(y, (\{1\}, \{2\}, \{3\}))$, has two edges that contain y , namely $(y, (\{1, 2\}, \{3\}))$ and $(y, (\{1\}, \{2, 3\}))$. The (canonical) permutahedral representation of 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 j merges of two-consecutive-set parts in ω (for example, merging ω_1 with ω_2 , and ω_3 with ω_4). 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.

Duality. We recall that two complexes are dual if there is a bijection between the faces that inverts the inclusion relationships (see, for example, [13, section 11.3]). We

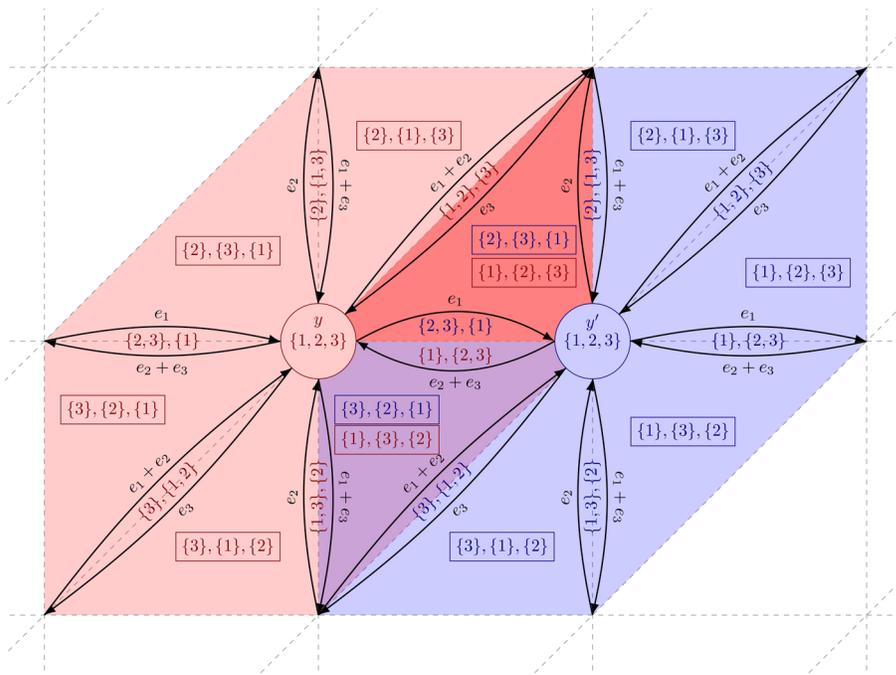


FIG. 4. The permutahedral representation of the simplices in the stars of vertices y and y' . The arrows indicate the steps in the edge walk; see Remark 2.14 and the paragraph “Cycles and the permutahedron.” (Color available online.)

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 [13].

We will now see that each d -dimensional cell of \mathcal{T}^* is a d -permutahedron. Indeed, Lemma 2.20 fully characterizes the facial structure of the star of a vertex in the FK-triangulation in terms of ordered partitions. This result is reminiscent of Lemma 2.5 that characterizes the facial structure of a permutahedron. In fact, it is easy to see that both structures are dual as claimed in the next proposition.

PROPOSITION 2.21. *The star of a vertex in a CFK-triangulation is combinatorially dual to a permutahedron.*

This proposition explains the nomenclature permutahedral representation.

The above proposition follows also from the following stronger and more geometric result.

PROPOSITION 2.22. *The Voronoi cell of a Coxeter triangulation of type \tilde{A}_d is a permutahedron.*

This can be found in [20, Chapter 21, section 3.F]; see also [18] and Appendix A for a new and more direct proof. We stress that the Coxeter triangulation of type \tilde{A}_d is Delaunay,² but this does not hold for all types of Coxeter triangulations [17]. As we have seen in section 2.2.3, Coxeter triangulations of type \tilde{A}_d are combinatorially

²We recall that a triangulation of a finite set of points of Euclidean space is called Delaunay if its simplices can be circumscribed by empty balls, i.e., balls whose interiors contain no point of the set.

equivalent to FK-triangulations. Hence, Proposition 2.21 follows from Proposition 2.22.

2.3.2. Basic operations on FK-triangulations. We now describe how to implement some basic operations on an FK-triangulation whose simplices are represented by their permutahedral representations. Notice that the permutahedral representation of any simplex requires $O(d)$ space. From now on, we assume that the floor function can be computed in constant time.

Point location. Given a point $x \in \mathbb{R}^d$, Lemma 2.12 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 [26] 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 in $O(d)$ time so that $m_0 < \dots < m_k$ using counting sort. Then, the canonical permutahedral representation (\tilde{v}'_0, ω') of τ is found by merging consecutive parts of ω so as to obtain $k+1$ parts as follows :

$$\begin{aligned} v'_0 &= v_{m_0} = v_0 + \sum_{j \in \omega_1} e_j + \dots + \sum_{j \in \omega_{m_0-1}} e_j, \\ \omega'_i &= \omega_{m_{i-1}} \cup \dots \cup \omega_{m_i-1} \quad \text{for } i \in \{1, \dots, k\}, \\ \omega'_{k+1} &= (\omega_1 \cup \dots \cup \omega_{m_0-1}) \cup (\omega_{m_k} \cup \dots \cup \omega_{l+1}). \end{aligned}$$

Ehrlich's algorithm takes a constant amount of time between successive subsets. The overall complexity of computing all subsets of $k+1$ vertices of σ using Ehrlich's algorithm is $O(k+s)$, 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 2.23 (face computation). *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.*

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 into $a_i \leq |\omega_i|$ 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 [42] and Ruskey and Savage [39], 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 2.20) to obtain the canonical permutahedral representation of the coface. The next lemma follows. The bound on the number s of cofaces follows, by duality (see Proposition 2.21), from Corollary 2.10.

LEMMA 2.24 (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 $s = p_{d-l, d-k}$ is the number of l -cofaces of a k -simplex in the FK-triangulation.*

As established in Corollary 2.10,

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

A more straightforward and slightly better bound for the number of cofacets of a k -simplex (i.e., $l = k + 1$) is given by Corollary 2.9. If we write $n = d - k$, that is, n is the codimension of the simplex, then we have the following.

Remark 2.25. The number of cofacets of a k -simplex, $k = d - n$, is upper bounded by $s = 2^{n+1}$.

We stress that this bound does not depend on d or k , but only on the codimension n . This will be crucial for the triangulation and tracing algorithm in Part II.

2.3.3. 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 2.2.2. 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 can be seen by inspection of the proof of Lemma 2.16.

3. Part II: 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 [32] but departs from its basic version 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} . To circumvent this difficulty, we decompose the problem of computing $\hat{\mathcal{M}}$ 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, [22, 43]. In this paper, we focus on the tracing problem, although we discuss very briefly (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 compared to the original Marching Cube algorithm is that we replace the usual cubical grid by a CFK-triangulation of the ambient space. Taking a CFK-triangulation instead of a grid allows us to easily construct a PL-approximation of \mathcal{M} , which 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. [22]; see also [34]. The novelty here is using the data structure of section 2.3 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 $f : \mathbb{R}^d \rightarrow \mathbb{R}^{d-n}$ 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 nondegenerate. Then the zero set of f is an n -dimensional manifold as a direct consequence of the implicit function theorem; see, for example, [23, section 3.5]. We further assume that $f^{-1}(0)$ is compact. As in [1] we consider a triangulation \mathcal{T} of the ambient space \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.,

$$(9) \quad \forall x \in \sigma : \hat{f}(x) = \sum_{v \in \sigma} \lambda_v(x) f(v),$$

where the λ_v are the barycentric coordinates of x with respect to the vertices v of σ . For any function $g : \mathbb{R}^d \rightarrow \mathbb{R}^{d-n}$ we write g^i , with $i = 1, \dots, d-n$, 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 [12, 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. We recall that the length of the longest edge in the ambient triangulation \mathcal{T} , called the diameter of \mathcal{T} , is denoted by 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 2.2. We assume that the manifold $\hat{\mathcal{M}}$ and the triangulation \mathcal{T} satisfy the following hypothesis.

Genericity Hypothesis 3.1. Suppose that σ is a d -simplex of \mathcal{T} that intersects H_{σ} . Then the following two statements hold:

- No subface of σ of dimension less than k intersects H_{σ} .
- 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. Genericity Hypothesis 3.1 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 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 (alternatively one could perturb

the planes in the hyperplane arrangement by perturbing the matrix Λ). 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

```

1 Initialize the queue  $\mathcal{Q}$  and the set  $\mathcal{S}$  with  $\tau_0$ 
2 while the queue  $\mathcal{Q}$  is not empty do
3   Pop a  $k$ -dimensional simplex  $\tau$  from  $\mathcal{Q}$ 
4   foreach cofacet  $\phi$  of  $\tau$  do
5     foreach facet  $\sigma$  of  $\phi$  do
6       if  $\sigma$  does not lie in  $\mathcal{S}$  and intersects  $\hat{\mathcal{M}}$  (which can be decided using
7         the oracle) then
8           Insert  $\sigma$  into the queue  $\mathcal{Q}$ 
           Insert  $\sigma$  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 3.1. 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 $\sigma = [v_0, \dots, v_k]$ of \mathcal{T} with the n -flat H_σ that interpolates f inside σ . One can compute the $k+1$ barycentric coordinates λ_i , $i = 0, \dots, k$, of $x = \sigma \cap H_\sigma$ by solving a linear system of $k+1$ equations, namely $\sum \lambda_i = 1$ and $\sum \lambda_i f(v_i) = 0$. It then remains to check whether the λ_i are all nonnegative (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 $(k+1) \times (k+1)$ 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, [22, 36, 37] and also the discussion in Wenger's book [43, 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 3.1 (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 3.3.

The algorithm is described as Algorithm 1. It takes as input the permutahedral representation of an ambient CFK-triangulation \mathcal{T} and a seed k -simplex τ_0 of \mathcal{T} . We assume that \mathcal{T} satisfies the Genericity Hypothesis 3.1, 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 whether 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 the k -dimensional simplices of \mathcal{T} that have been found to intersect $\hat{\mathcal{M}}$. Each such intersection, which consists of a single point (by the Genericity Hypothesis 3.1), is a vertex 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 $\hat{\mathcal{M}}$. Since this graph is connected, the algorithm identifies the set $\hat{\mathcal{M}}_0$ of all the vertices of $\hat{\mathcal{M}}$ and 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 2.2 to compute cofaces by increasing dimensions.

3.3. Topological guarantees for the 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 [12], and we simply state here the main results specialized to the case of CFK-triangulations.

We say that f has *bounded complexity* if the following three quantities γ_{\max} , λ_{\min} , and α_{\max} are positive and bounded:

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

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 ;
- $\text{grad} f^i = (\partial_j f_i)_j$ denotes the gradient of component f^i for $i \in [1, d - n]$;
- $\text{Gram}(\nabla f)$ denotes the Gram matrix whose elements are $\nabla f^i \cdot \nabla f^j$, where \cdot stands for the dot product. $\lambda_{\min}(x)$ is the smallest absolute value of the eigenvalues of $\text{Gram}(\nabla f(x))$;³
- $\text{Hes}(f) = (\partial_k \partial_l f_i)_{k,l}$ 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.⁴

We can now restate the topological result of [12].

THEOREM 3.2. *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 γ_{\max} , λ_{\min} , and α_{\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 3.3 (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*

$$d_F(\mathcal{M}_a, \mathcal{M}_b) = \inf_{h \in \mathcal{H}} \sup_{x \in \mathcal{M}_a} d(x, h(x)),$$

where the distance $d(\cdot, \cdot)$ is the Euclidean one.

THEOREM 3.4. *Assume that the function f has bounded complexity. Then, we have $d_F(\mathcal{M}, \hat{\mathcal{M}}) = O(D^2)$ where the constant in the big O depends on γ_{\max} , λ_{\min} , and α_{\max} .*

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

PROPOSITION 3.5. *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. Recall that n is the dimension of \mathcal{M} and $k = d - n$ is its codimension.*

Proof. 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 is $O(2^n)$. This is the bound from Remark 2.25. We recall that by duality (see Proposition 2.21) a k -simplex in an FK-triangulation corresponds to an n -face of the permutahedron where $n = d - k$. Moreover, the cofacets s in the FK-triangulation correspond to facets in the permutahedron. The bound $O(2^n)$ is given in Remark 2.25, which in turn is a consequence of Corollary 2.9. Each of these cofacets has $k + 2$ facets. Therefore, for each iteration of the while loop, the algorithm

³Because a Gram matrix is a symmetric square matrix, its eigenvalues are well defined and real.

⁴The 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 .

applies the intersection oracle on $O(k2^n)$ simplices. By using this observation and the complexities in Lemmas 2.23 and 2.24, 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. \square

Since the intersection oracle reduces to evaluating f at the $k + 1$ vertices of σ plus solving a $(k + 1) \times (k + 1)$ linear system, we have $I = O(k^\omega)$, 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 longest edge length D (or diameter) of a simplex in the triangulation which bounds the density of the output sample, and the Fréchet distance 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 3.6 (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 an FK-triangulation, we have*

$$N_C \leq \frac{K}{n!} \times \left(\frac{d(d+1)\sqrt{d(d+2)}}{2\sqrt{2}D} \right)^n$$

and

$$N_{FK} \leq \frac{K}{n!} \times \left(\frac{d^2(d+1)}{\sqrt{2}D} \right)^n,$$

where D is the diameter of a simplex of \mathcal{T} .

Proof. By the definition of CFK-triangulations in section 2.2, \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 $R_{\mathcal{T}}$ be the maximal norm of the vectors u . Note that the distance between two consecutive hyperplanes in family H_u is $1/\|u\| \geq 1/R_{\mathcal{T}}$.

We will rescale the arrangement of hyperplanes so that the maximal diameter of the simplices is D . 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:

$$(10) \quad N_{\mathcal{T}} \leq K \binom{d(d+1)/2}{n} \times \prod_{i \in I} \frac{\sqrt{d}L_{\mathcal{T}}\|u_i\|}{D} \leq \frac{K}{n!} \times \left(\frac{d(d+1)\sqrt{d}L_{\mathcal{T}}R_{\mathcal{T}}}{2D} \right)^n.$$

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 2.2 that $R_C = R_{FK} = \sqrt{2}$. The longest edge L_{FK} in an FK -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 [17, section 6, point 6 of \tilde{A}_d] that the longest edge length in the Coxeter triangulation of type \tilde{A}_d is

$$(11) \quad L_C = \begin{cases} \frac{\sqrt{d+1}}{2} & \text{if } d \text{ is odd,} \\ \frac{1}{2} \sqrt{\frac{d(d+2)}{(d+1)}} & \text{if } d \text{ is even,} \end{cases}$$

and hence $L_C < \frac{\sqrt{d+2}}{2}$. We then deduce from (10) that

$$N_C \leq \frac{K}{n!} \times \left(\frac{d(d+1)\sqrt{d(d+2)}}{2\sqrt{2}D} \right)^n,$$

$$N_{FK} \leq \frac{K}{n!} \times \left(\frac{d^2(d+1)}{\sqrt{2}D} \right)^n. \quad \square$$

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 7).

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 3.7. *The combinatorial complexity of $\hat{\mathcal{M}}$ is $|\mathcal{S}| \times (\frac{3}{2})^n (n+1)!$, where $|\mathcal{S}|$ is bounded in Proposition 3.6. If $n = O(1)$, the combinatorial complexity of $\hat{\mathcal{M}}$ is polynomial in d , and $\delta = 1/D$.*

Proof. 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 an n -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 2.21). Hence, σ^* is an 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 2.10. 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 3.6. □

We combine Propositions 3.5, 3.6, and 3.7 and Theorem 3.2 to obtain our main result that states that the tracing algorithm constructs geometrically close and

topologically correct PL-approximations of isomanifolds of bounded complexity in polynomial time.

THEOREM 3.8. *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). In addition, let D be the length of the longest edge in the ambient triangulation \mathcal{T} . The size of the output and the time complexity of the tracing 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} .*

Moreover, 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.

3.5. Dimensionality reduction. As seen from Proposition 3.6, 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 [9, 16]. 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 an n -dimensional smooth submanifold 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 3.4.

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 $2D$ -net.⁵ However, by doing so, we will lose the combinatorial structure of the mesh.

We now show how to compute a sample of \mathcal{M} whose one-sided Hausdorff distance to the manifold is D 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 3.9 (Johnson–Lindenstrauss lemma for manifolds [19, 41]). *Pick any $\varepsilon, \eta > 0$, and let $d' = \Omega(\frac{n}{\varepsilon^2} \log \frac{1}{\varepsilon} + \frac{1}{\varepsilon^2} \log \frac{\Gamma}{\eta})$, 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 $(1 - \varepsilon) \sqrt{\frac{d'}{d}} \leq \frac{\|\Phi x - \Phi y\|}{\|x - y\|} \leq (1 + \varepsilon) \sqrt{\frac{d'}{d}}$.*

⁵Consider each point of the original set in turn for inclusion in the sparsified set. A point is inserted only if it is at distance larger than D from all the already inserted points.

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 [25], the analysis of the algorithm will be unchanged. Proposition 3.6 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 3.5 implies that the overall complexity is still polynomial in d (and d'). Note that we assume that $d' < d$ and therefore d' can be safely absorbed.

3.6. Isomanifolds with boundary, and isostratifolds. 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 $f : \mathbb{R}^d \rightarrow \mathbb{R}^{d-n}$, we are given another function $f_\partial : \mathbb{R}^d \rightarrow \mathbb{R}$, 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 (9), 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}_∂ 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 $\hat{f}_\partial^{-1}(0) \cap \hat{f}^{-1}(0)$ 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 $(\hat{f}|_\tau)^{-1}(0) \cap (\hat{f}_\partial|_\tau)^{-1}([0, \infty)) \cap \tau$.

Because $\partial\mathcal{M}$ is itself a manifold without boundary, $\widehat{\partial\mathcal{M}}$ is as defined before. We will assume that the Genericity Hypothesis 3.1 holds for both $\hat{\mathcal{M}}$ and $\widehat{\partial\mathcal{M}}$.

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 $\widehat{\partial\mathcal{M}}$. 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 3.1, 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 (we identify a simplex in S_k (resp., S_{k+1}) and the intersection point $S_k \cap \hat{\mathcal{M}}$ (resp., $S_{k+1} \cap \widehat{\partial\mathcal{M}}$):

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 stratified spaces 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 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 $k, k+1, \dots, d$ that intersect the strata of dimension $n, n-1, \dots, 0$.

4. Part III: Experiments and applications. In sections 4.1 and 4.2 we discuss the experimental results on the data structure and tracing algorithm, respectively. Section 4.3 concerns an application in algebraic geometry.

4.1. Experimental results for the data structure. The data structure and the basic operations have been implemented in C++ and have been integrated in the GUDHI library [29]. 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 $2\mu s$.

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	0.042	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 2

Running time of the face generation algorithm per computed face (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 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	0.11	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 4

Running time of the coface generation algorithm per computed face (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

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 .

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 the 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.

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

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 6
Average running times in milliseconds of the cofacet 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

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

4.2. Experimental results for the tracing algorithm. The data structure of section 2.2 and the algorithm of section 3 have been implemented in C++. The code is robust and fast and has been released in the GUDHI library [29]. Full detail on the implementation, including the implementation of the oracle, can be found in [31].

In this section, we explore the dependency of our C++ implementation of our data structure and of the manifold tracing algorithm. We will compare the running time of our code for the Coxeter (A_d) triangulation with the running time for the Freudenthal–Kuhn triangulation. We discuss the results of our algorithm on several examples, such as the “Chair” and (possibly clipped) Clifford torus.

4.2.1. Performance of the tracing algorithm. We show the performance of our implementation of the manifold tracing algorithm for various ambient and intrinsic dimensions in Figure 6. In Figure 7, we can see that using Coxeter triangulation is beneficial in practice as it produces a smaller output in less time (see Proposition 3.6).

In Figure 8, 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 .

4.2.2. Manifolds with boundary. The algorithm has been adapted to handle submanifolds with boundary and surfaces with a piecewise smooth boundary; see section 3.6. 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. The torus has been rotated and translated in \mathbb{R}^4 so that the coordinate axes do not play any special role.

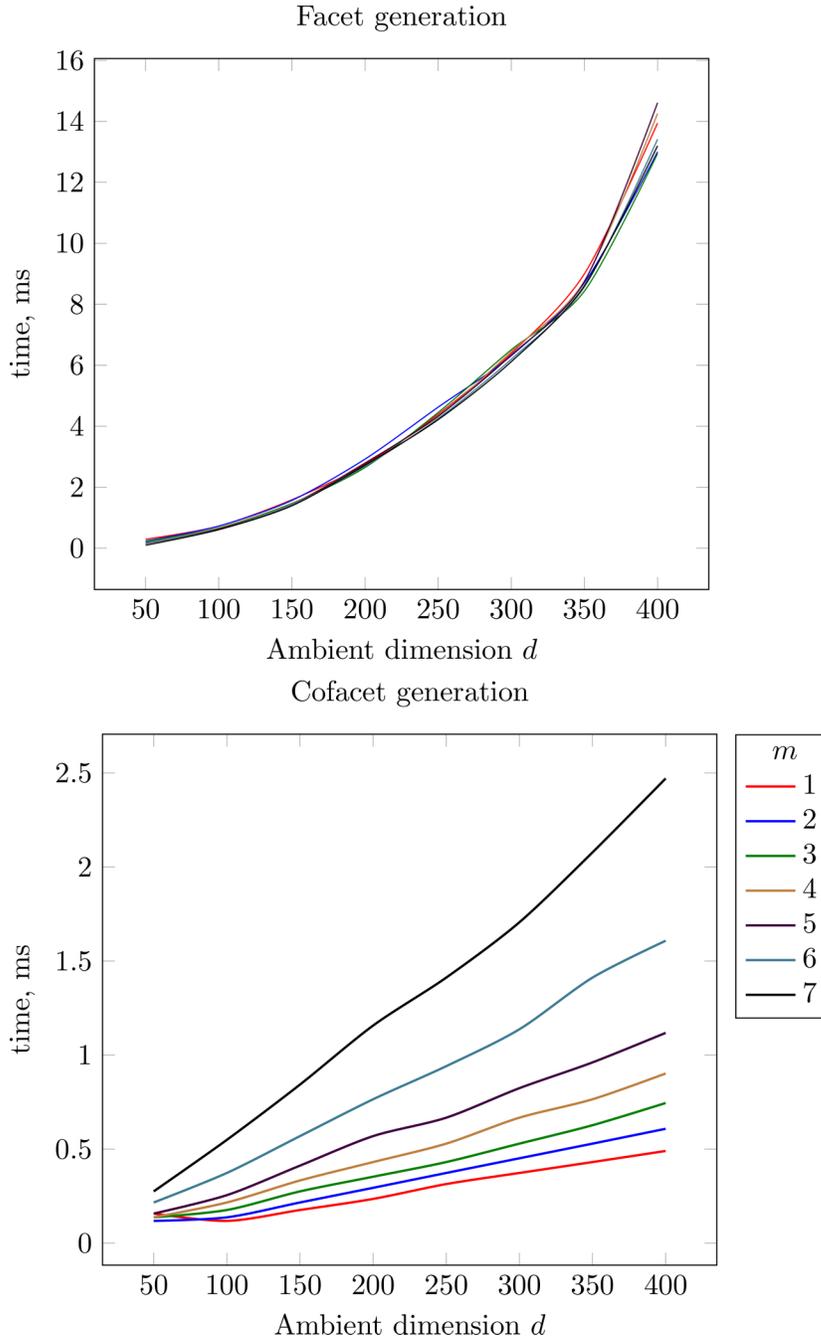


FIG. 5. Graphical display of the results of Tables 5 and 6.

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

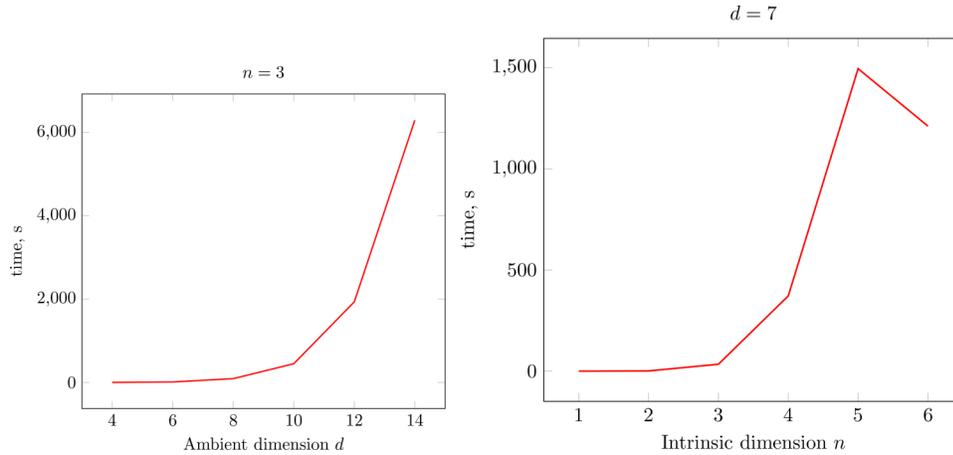


FIG. 6. The effect of the ambient dimension d and of the intrinsic dimension n on the computation time of the manifold tracing algorithm. The reconstructed manifold in the tests is the n -dimensional sphere embedded in \mathbb{R}^d . The ambient triangulation used is a Coxeter triangulation of type \tilde{A}_d . The diameter of the full simplices is fixed for all d .

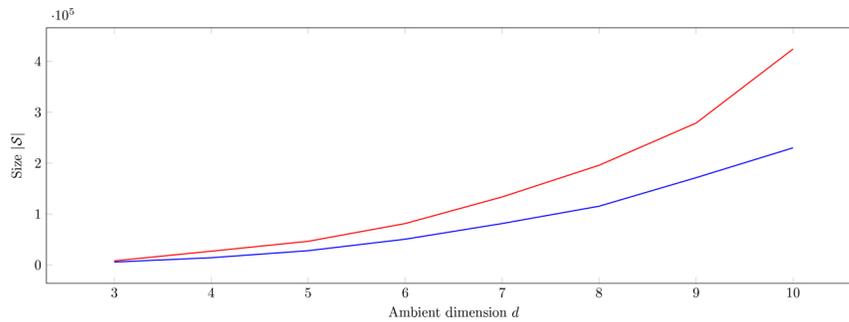


FIG. 7. Comparison of the size of the output of the manifold tracing algorithm using two types of ambient triangulations: a Coxeter triangulation of type \tilde{A}_d (in blue) and the FK-triangulation of \mathbb{R}^d (in red) with the same diameter $0.07\sqrt{d}$ of d -dimensional simplices. The reconstructed manifold is the two-dimensional implicit surface “Chair” embedded in \mathbb{R}^d given by the equations $(x_1^2 + x_2^2 + x_3^2 - 0.8)^2 - 0.4((x_3 - 1)^2 - 2x_1^2)((x_3 + 1)^2 - 2x_2^2) = 0$ and $x_i = 0$ for $i > 3$. (Color available online.)

complex projective space $\mathbb{C}\mathbb{P}^d = (\mathbb{C}^{d+1} \setminus 0)/\mathbb{C}^*$ 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 $\mathbb{C}\mathbb{P}^2$, 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 $[z_1 : \dots : z_{d+1}]$ on $\mathbb{C}\mathbb{P}^d$ to affine coordinates $[z_1' : \dots : z_{i-1}' : 1 : z_{i+1}' : \dots : z_{d+1}']$ by picking the i th coordinate to be equal to 1, that is, $z_j' = z_j/z_i$. Given some homogenous coordinates $[z_1 : \dots : z_{d+1}]$, we can choose the i th coordinate to be set to 1 to be the coordinate whose absolute value is the largest, so that $\mathbb{C}\mathbb{P}^d$ can be written as the union of the $d + 1$ sets $\{[z_1' : \dots : z_{i-1}' : 1 : z_{i+1}' : \dots : z_{d+1}'] \mid |z_j'| \leq 1\}$, 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} ,

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

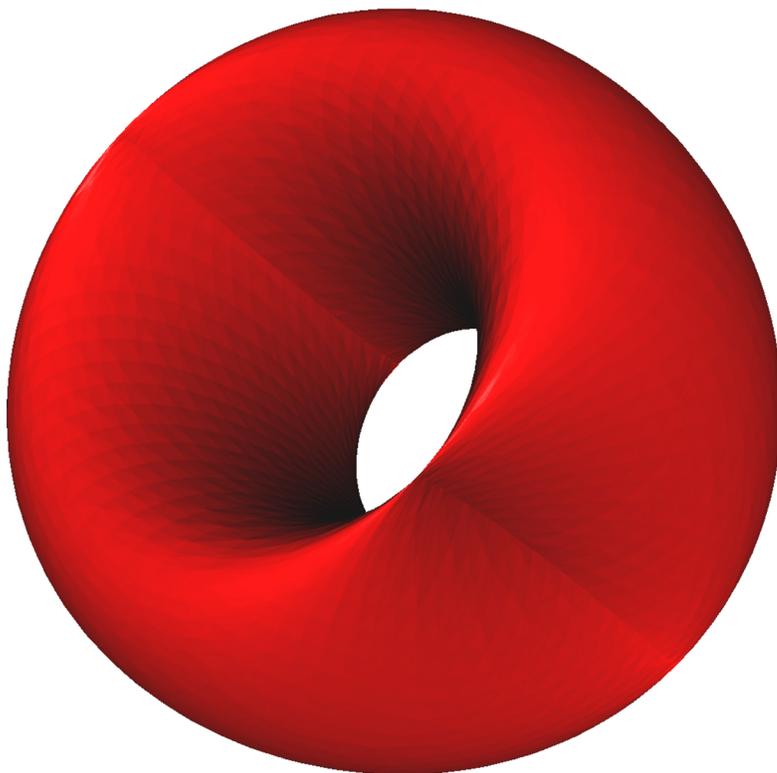


FIG. 8. The piecewise-linear approximation of a flat torus embedded in \mathbb{R}^{10} defined by the equations $x_1^2 + x_2^2 = 1$, and $x_3^2 + x_4^2 = 1$, and $x_i = 0$ for $i > 4$, projected to \mathbb{R}^3 . The ambient triangulation used is a Coxeter triangulation of type \tilde{A}_{10} with the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

Let f be a homogenous polynomial in $d + 1$ complex variables and their complex conjugates. For each i , we can fix the i th 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 $\mathbb{C}\mathbb{P}^d$ 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 us to recover the Euler characteristic of $f = 0$ on $\mathbb{C}\mathbb{P}^d$.

This principle generalizes to varieties of higher codimension, that is, to varieties defined by a number of homogenous polynomials f_1, \dots, f_{d-m} .

We illustrate the above construction on the equation $z_1^2 \bar{z}_2 + z_2^2 \bar{z}_3 + z_3^2 \bar{z}_1 = 0$ in $\mathbb{C}\mathbb{P}^2$. 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 $x_1 + x_1^2 x_2 + x_2^2 - x_2 y_1^2 + 2x_1 y_1 y_2 - y_2^2 = 0$

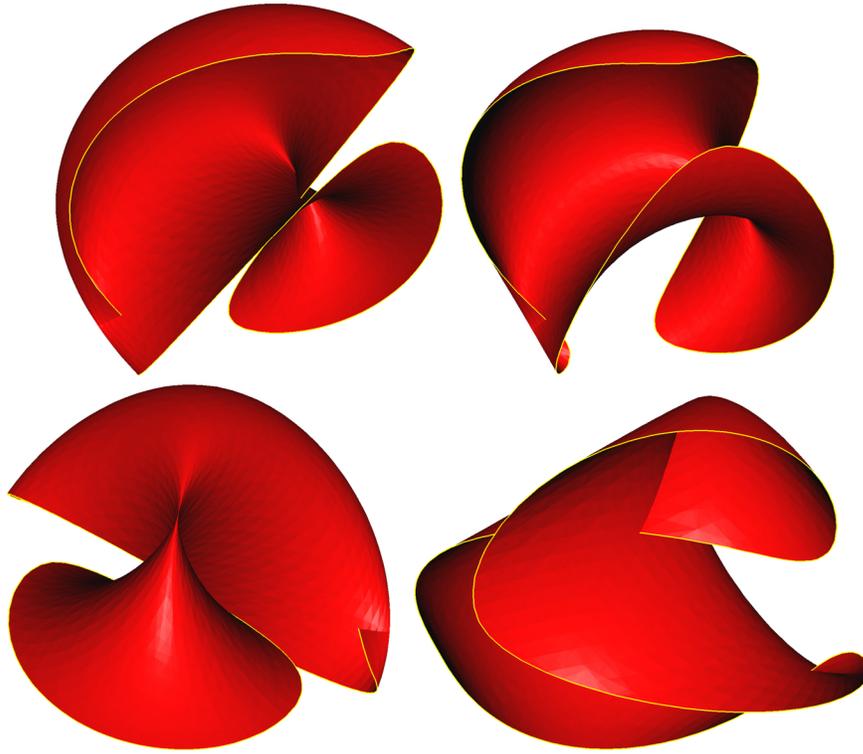


FIG. 9. Four views of the flat torus in \mathbb{R}^4 given by two equations $x_1^2 + x_2^2 = 1$, and $x_3^2 + x_4^2 = 1$ cut by the hypersphere $(x_1 - 1)^2 + x_2^2 + (x_3 - 1)^2 + x_4^2 = 4$, projected to \mathbb{R}^3 . The ambient triangulation used is a Coxeter triangulation of type \tilde{A}_4 with the diameter 0.15 of the full-dimensional simplices. The reconstructed boundary is highlighted in yellow. The size $|S|$ 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.

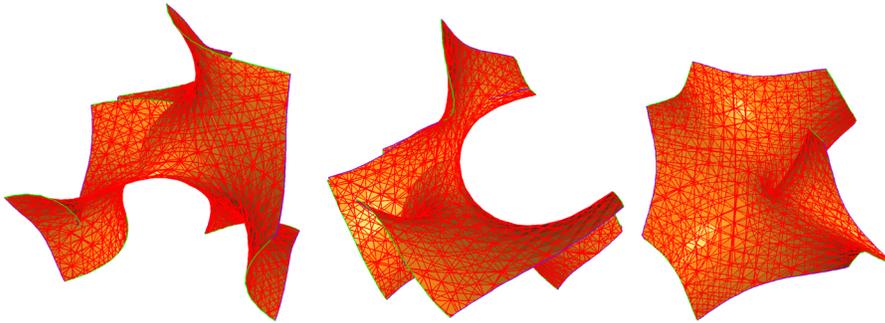


FIG. 10. The three triangulated surfaces as discussed in the example of $z_1^2 \bar{z}_2 + z_2^2 \bar{z}_3 + z_3^2 \bar{z}_1 = 0$ in $\mathbb{C}\mathbb{P}^2$ after projection from \mathbb{R}^4 to \mathbb{R}^3 .

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 10. 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. The complexity is polynomial in the ambient dimension so that our algorithm should be beneficial in the numerous areas where one considers isomanifolds of low intrinsic dimensions embedded in high dimensional spaces.

Since isomanifolds are a special type of manifolds, it is tempting to see whether 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 (per connected component) and an oracle that can determine whether a k -simplex of the ambient triangulation intersects \mathcal{M} or not. In this general (non-isomanifold) 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, 44]. 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 neighborhood of \mathcal{M}). Since there are exponentially many such simplices, the perturbation scheme has a complexity that depends exponentially on the ambient dimension d . 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.

Appendix A. Alternative proof of Proposition 2.22.

Proof of Proposition 2.22. 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 .

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

where $x^{\{k\}}$ denotes k consecutive coordinates x . This simplex is a simplex in the Coxeter triangulation, as defined in section 2.2.3. In [17] we have seen that the circumcenter of this simplex is

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

with $i \in \{0, \dots, d\}$. The circumcenter of a Delaunay simplex is a Voronoi vertex. We recall the following:

- 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}_{EC} that go through 0, that is, the hyperplanes with normals $r_{j,k} = e_j - e_k$, with $j \neq k$. See, for example, [14, 17, 30]. 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 j th and k th coordinates of c . Here we used the upper index i to denote the i th coordinate. Using the cycle notation for the permutation group (see, for example, [5, Chapter 6]), this coincides with the 2-cycle (jk) . 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

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

which again permutes the j th and k th coordinates. 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. \square

Acknowledgments. We thank Dominique Attali, Guilherme da Fonseca, Arijit Ghosh, Vincent Pilaud, and Aurélien Alvarez for their comments and suggestions. We also acknowledge the reviewers, whose comments improved the exposition significantly.

REFERENCES

- [1] E. ALLGOWER AND K. GEORG, *Estimates for piecewise linear approximations of implicitly defined manifolds*, Appl. Math. Lett., 2 (1989), pp. 111–115, [https://doi.org/10.1016/0893-9659\(89\)90001-3](https://doi.org/10.1016/0893-9659(89)90001-3).
- [2] E. ALLGOWER AND K. GEORG, *Numerical Continuation Methods: An Introduction*, Springer Ser. Comput. Math. 13, Springer Science & Business Media, 1990, <https://doi.org/10.1007/978-3-642-61257-2>.
- [3] E. L. ALLGOWER AND P. H. SCHMIDT, *An algorithm for piecewise-linear approximation of an implicitly defined manifold*, SIAM J. Numer. Anal., 22 (1985), pp. 322–346, <https://doi.org/10.1137/0722020>.
- [4] A. ALVAREZ AND B. DEROIN, *Dynamique et topologie du feuilletage de Jouanolou*, preprint available on request from the first author, 2019, <https://hal.archives-ouvertes.fr/FOURIER/medihal-02274656>.
- [5] M. A. ARMSTRONG, *Groups and Symmetry*, Springer, 1988.
- [6] D. ATTALI, A. LIEUTIER, AND D. SALINAS, *Efficient data structure for representing and simplifying simplicial complexes in high dimensions*, Internat. J. Comput. Geom. Appl., 22 (2012), pp. 279–303.
- [7] J. BOISSONNAT AND C. MARIA, *The simplex tree: An efficient data structure for general simplicial complexes*, Algorithmica, 70 (2014), pp. 406–427, <https://doi.org/10.1007/s00453-014-9887-3>.
- [8] J.-D. BOISSONNAT, F. CHAZAL, AND M. YVINEC, *Geometric and Topological Inference*, Cambridge Texts Appl. Math., Cambridge University Press, 2018, <https://doi.org/10.1017/9781108297806>.

- [9] J.-D. BOISSONNAT AND A. GHOSH, *Manifold reconstruction using tangential Delaunay complexes*, Discrete Comput. Geom., 51 (2014), pp. 221–267, <https://doi.org/10.1007/s00454-013-9557-2>.
- [10] J.-D. BOISSONNAT, S. KACHANOVICH, AND M. WINTRAECKEN, *Triangulating submanifolds: An elementary and quantified version of Whitney’s method*, Discrete Comput. Geom., 66 (2021), pp. 386–434, <https://doi.org/10.1007/s00454-020-00250-8>.
- [11] J.-D. BOISSONNAT, C. KARTHIK, AND S. TAVENAS, *Building efficient and compact data structures for simplicial complexes*, Algorithmica, 79 (2017), pp. 530–567.
- [12] J.-D. BOISSONNAT AND M. WINTRAECKEN, *The topological correctness of PL-approximations of isomanifolds*, Found. Comput. Math., 22 (2022), pp. 967–1012, <https://doi.org/10.1007/s10208-021-09520-0>.
- [13] J.-D. BOISSONNAT AND M. YVINEC, *Algorithmic Geometry*, Cambridge Texts Appl. Math., Cambridge University Press, 1998.
- [14] N. BOURBAKI, *Lie Groups and Lie Algebras. Chapters 4-6*, Elements of Mathematics, Springer-Verlag, 2002; translated from the 1968 French original by A. Pressley.
- [15] Y.-C. CHEN, *Solution Manifold and Its Statistical Applications*, preprint, <https://arxiv.org/abs/2002.05297>, 2021.
- [16] S.-W. CHENG, T. K. DEY, AND E. A. RAMOS, *Manifold reconstruction from point samples*, in Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), 2005, pp. 1018–1027.
- [17] A. CHOUDHARY, S. KACHANOVICH, AND M. WINTRAECKEN, *Coxeter triangulations have good quality*, Math. Comput. Sci., 14 (2020), pp. 141–176, <https://doi.org/10.1007/s11786-020-00461-5>.
- [18] A. CHOUDHARY, M. KERBER, AND S. RAGHVENDRA, *Polynomial-sized topological approximations using the permutahedron*, in 32nd International Symposium on Computational Geometry (SoCG 2016), S. P. Fekete, and A. Lubiw, eds., LIPIcs. Leibniz Int. Proc. Inform. 51, Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2016, 31, <https://doi.org/10.4230/LIPIcs.SocG.2016.31>.
- [19] K. L. CLARKSON, *Tighter bounds for random projections of manifolds*, in Proceedings of the 24th ACM Symposium on Computational Geometry, College Park, MD, 2008, pp. 39–48, <https://doi.org/10.1145/1377676.1377685>.
- [20] J. H. CONWAY AND N. J. A. SLOANE, *Sphere-Packings, Lattices, and Groups*, Springer-Verlag, New York, 1987.
- [21] H. S. M. COXETER, *Discrete groups generated by reflections*, Ann. of Math. (2), 35 (1934), pp. 588–621.
- [22] D. P. DOBKIN, A. R. WILKS, S. V. F. LEVY, AND W. P. THURSTON, *Contour tracing by piecewise linear approximations*, ACM Trans. Graph., 9 (1990), pp. 389–423, <https://doi.org/10.1145/88560.88575>.
- [23] J. J. DUISTERMAAT AND J. A. C. KOLK, *Multidimensional Real Analysis I: Differentiation*, Cambridge Stud. Adv. Math. 86, Cambridge University Press, 2004, <https://doi.org/10.1017/CBO9780511616716>.
- [24] B. C. EAVES, *A Course in Triangulations for Solving Equations with Deformations*, Lecture Notes in Econom. and Math. Systems 234, Springer-Verlag, 1984, <https://doi.org/10.1007/978-3-642-46516-1>.
- [25] A. EFTEKHARI AND M. B. WAKIN, *What happens to a manifold under a bi-Lipschitz map?*, Discrete Comput. Geom., 57 (2017), pp. 641–673, <https://doi.org/10.1007/s00454-016-9847-6>.
- [26] G. EHRLICH, *Loopless algorithms for generating permutations, combinations, and other combinatorial configurations*, J. ACM, 20 (1973), pp. 500–513.
- [27] H. FREUDENTHAL, *Simplizialzerlegungen von beschränkter Flachheit*, Ann. of Math. (2), 43 (1942), pp. 580–582.
- [28] A. GOMES, I. VOICULESCU, J. JORGE, B. WYVILL, AND C. GALBRAITH, *Implicit Curves and Surfaces: Mathematics, Data Structures and Algorithms*, Springer, 2009, <https://doi.org/10.1007/978-1-84882-406-5>.
- [29] GUDHI Project, <https://gudhi.inria.fr/>.
- [30] J. E. HUMPHREYS, *Reflection Groups and Coxeter Groups*, Cambridge Stud. Adv. Math. 29, Cambridge University Press, 1992.
- [31] S. KACHANOVICH, *Meshing Submanifolds Using Coxeter Triangulations*, thesis, COMUE Université Côte d’Azur (2015 - 2019), 2019, <https://hal.inria.fr/tel-02419148>.
- [32] W. E. LORENSEN AND H. E. CLINE, *Marching cubes: A high resolution 3D surface construction algorithm*, ACM SIGGRAPH Comput. Graph., 21 (1987), pp. 163–169, <https://doi.org/10.1145/37401.37422>.

- [33] M. MAES AND B. KAPPEN, *On the permutahedron and the quadratic placement problem*, Philips J. Res., 46 (1992), pp. 267–292.
- [34] C. MIN, *Simplicial isosurfacing in arbitrary dimension and codimension*, J. Comput. Phys., 190 (2003), pp. 295–310, [https://doi.org/10.1016/S0021-9991\(03\)00275-4](https://doi.org/10.1016/S0021-9991(03)00275-4).
- [35] T. S. NEWMAN AND H. YI, *A survey of the marching cubes algorithm*, Comput. Graph., 30 (2006), pp. 854–879, <https://doi.org/10.1016/j.cag.2006.07.021>.
- [36] J. M. ORTEGA AND W. C. RHEINOLDT, *Iterative Solution of Nonlinear Equations in Several Variables*, Classics Appl. Math. 30, SIAM, 2000, <https://doi.org/10.1137/1.9780898719468>.
- [37] A. M. OSTROWSKI, *Solution of Equations and Systems of Equations*, Pure and Applied Mathematics: A Series of Monographs and Textbooks 9, Elsevier, 2016.
- [38] B. RENNIE AND A. DOBSON, *On Stirling numbers of the second kind*, J. Combinatorial Theory, 7 (1969), pp. 116–121, [https://doi.org/10.1016/S0021-9800\(69\)80045-1](https://doi.org/10.1016/S0021-9800(69)80045-1).
- [39] F. RUSKEY AND C. D. SAVAGE, *Gray codes for set partitions and restricted growth tails*, Australas. J. Combin., 10 (1994), pp. 85–96.
- [40] M. J. TODD, *The Computation of Fixed Points and Applications*, Lecture Notes in Econom. and Math. Systems 124, Springer-Verlag, 1976, <https://doi.org/10.1007/978-3-642-50327-6>.
- [41] N. VERMA, *A Note on Random Projections for Preserving Paths on a Manifold*, Technical Report CS2011-0971, UC San Diego, 2011.
- [42] T. R. WALSH, *Loop-free sequencing of bounded integer compositions*, J. Combin. Math. Combin. Comput., 33 (2000), pp. 323–345.
- [43] R. WENGER, *Isosurfaces: Geometry, Topology, and Algorithms*, AK Peters/CRC Press, 2013.
- [44] H. WHITNEY, *Geometric Integration Theory*, Princeton University Press, 1957, <https://doi.org/10.1515/9781400877577>.
- [45] G. M. ZIEGLER, *Lectures on Polytopes*, Grad. Texts in Math. 152, Springer, New York, 2012, <https://doi.org/10.1007/978-1-4613-8431-1>.