

Tracing isomanifolds in R^d in time polymonial in d

Jean-Daniel Boissonnat, Siargey Kachanovich, Mathijs Wintraecken

▶ To cite this version:

Jean-Daniel Boissonnat, Siargey Kachanovich, Mathijs Wintraecken. Tracing isomanifolds in \mathbb{R}^d in time polymonial in d. 2020. hal-03006663v1

HAL Id: hal-03006663 https://inria.hal.science/hal-03006663v1

Preprint submitted on 16 Nov 2020 (v1), last revised 17 Mar 2021 (v3)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers. L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Tracing isomanifolds in \mathbb{R}^d in time polymonial in d

Jean-Daniel Boissonnat

Université Côte d'Azur, Inria [Sophia-Antipolis, France] jean-daniel.boissonnat@inria.fr

Siargey Kachanovich

Université Côte d'Azur, Inria [Sophia-Antipolis, France] siargey.kachanovich@inria.fr

Mathijs Wintraecken

IST Austria [Klosterneuburg, Austria] m.h.m.j.wintraecken@gmail.com

Abstract

Isomanifolds are the generalization of isosurfaces to arbitrary dimension and codimension, i.e. 2 submanifolds of \mathbb{R}^d defined as the zero set of some multivariate multivalued smooth function 3 $f: \mathbb{R}^d \to \mathbb{R}^{d-n}$, where n is the intrinsic dimension of the manifold. A natural way to approximate Λ smooth isomanifold \mathcal{M} is to consider its Piecewise-Linear (PL) approximation $\hat{\mathcal{M}}$ based on a 5 triangulation \mathcal{T} of the ambient space \mathbb{R}^d . In this paper, we describe a simple algorithm to trace 6 isomanifolds from a given starting point. The algorithm works for arbitrary dimensions n and d, and 7 any precision D. Our main result is that, when f (or \mathcal{M}) has bounded complexity, the complexity 8 of the algorithm is polynomial in d and $\delta = 1/D$ (and unavoidably exponential in n). Since it is 9 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 10 PL-approximation of isomanifolds of bounded complexity in time polynomial in d. Combining 11 this algorithm with dimensionality reduction techniques, the dependency on d in the size of \hat{M} 12 can be completely removed with high probability. We also show that the algorithm can handle 13 isomanifolds with boundary and, more generally, isostratifolds. The algorithm has been implemented 14 and experimental results are reported, showing that it is practical and can handle cases that are far 15 ahead of the state-of-the-art.

16

2012 ACM Subject Classification Theory of computation \rightarrow Computational geometry

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

Funding The research leading to these results has received funding from the European Research Council (ERC) under the European Union's Seventh Framework Programme (FP/2007-2013) / ERC Grant Agreement No. 339025 GUDHI (Algorithmic Foundations of Geometry Understanding in Higher Dimensions).

Jean-Daniel Boissonnat: Supported by the French government, through the 3IA Côte d'Azur Investments in the Future project managed by the National Research Agency (ANR) with the reference number ANR-19-P3IA-0002.

Mathijs Wintraecken: Supported by the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 754411.

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





LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

17 **1** Introduction

Given a surface represented in \mathbb{R}^3 as the zero set of a function $f : \mathbb{R}^3 \to \mathbb{R}$, the goal of isosurfacing is to find a piecewise linear (PL) approximation of the surface. This question naturally extends to isomanifolds of higher dimensions and codimensions defined as the zero set of multivariate multivalued smooth functions $f : \mathbb{R}^d \to \mathbb{R}^{d-n}$. Isosurfaces play a crucial role in medical imaging, computer graphics and geometry processing [24]. Higher dimensional isomanifolds are also of fundamental importance in many fields like statistics [11], dynamical systems [27], econometrics, or mechanics [24].

State-of-the-art. The most widely used algorithm to trace isomanifolds is the Marching 25 Cube (MC) algorithm and its numerous variants [19, 29]. The MC algorithm uses a cubical 26 grid to tesselate the ambient space. In many applications in 3-dimensions, the ambient space 27 is decomposed into unstructured tetrahedral meshes, which led to the development of a 28 variant of the MC algorithm named the Marching Tetrahedra algorithm. In higher dimensions, 29 any tessellation of the ambient space has a complexity that depends exponentially on the 30 ambient dimension. Hence a key to extending marching algorithms to higher dimensions is 31 to circumvent the curse of dimensionality by using an *implicit* representation of the ambient 32 tessellation. This is impossible for general triangulations but easy to do if one uses a grid. 33 However, using a grid has other drawbacks and is not sufficient to break the exponential 34 barrier. The reason for this is that the number of configurations inside a cubical cell 35 grows exponentially with the dimension [29]. Hence the most promising approach seems to 36 subdivide the ambient space \mathbb{R}^d using a highly regular triangulation such as the Freudenthal-37 Kuhn triangulation. Some early work along this direction has been published in Applied 38 Mathematics [2, 17, 27], and a slightly more recent paper by Dobkin et al. [15] attracted the 39 interest of the Computer Graphics community to the related Coxeter triangulations. Dobkin 40 et al. however only considered the case of curves (n = 1). The most advanced work we are 41 aware of is due to Min [23]. Min's method uses the Freudenthal-Kuhn triangulation over 42 a dyadic grid of \mathbb{R}^d and applies to isomanifolds of any dimension and codimension. The 43 complexity of Min's method is, with our notations, $O(\delta^m \log \delta)$, where $\delta = 1/D$ and D is the 44 maximal diameter of the simplices. The ambient dimension d is a constant hidden in the 45 big O. The fact that the exponent of δ is the intrinsic dimension m, and not the ambient 46 dimension d is a clear improvement over earlier methods. However, although not explicitly 47 analysed by Min, the complexity in d remains exponential, and the method seems limited to 48 small ambient dimensions. Experimental results are only reported in dimensions 3, and 4. 49

Contributions. This paper discusses an efficient algorithm to compute a PL-approxi-50 mation of isomanifolds. We extend the work of Dobkin et al. [15] and describe a simple 51 algorithm to trace an *n*-dimensional isomanifold \mathcal{M} of \mathbb{R}^d for arbitrary *n* and *d*. Our 52 algorithm uses any triangulation of a family of regular triangulations of \mathbb{R}^d that includes 53 the Coxeter and the Freudenthal-Kuhn triangulations. Contrary to Min [23], our results are 54 obtained with a uniform triangulation leading to a very simple algorithm. Key to our results, 55 is a recently developed data structure that can implicitly store the full facial structure of 56 such triangulations. As shown in [9] and recalled in Section 2.1, the data structure is very 57 compact and allows to retrieve the faces or the cofaces of a simplex of any dimension in an 58 output sensitive way. Using this data structure, one can trace a connected submanifold of 59 \mathbb{R}^d , starting from a given initial point on the manifold (Section 3). Our algorithm produces a 60 PL-approximation of size polynomial in d and $\delta = 1/D$, and exponential in n. The complexity 61 of the algorithm is also polynomial in d, and δ , and exponential in n. 62

Moreover, by taking δ large enough, the PL-approximation output by the algorithm

J.-D. Boissonnat, S. Kachanovich, and M. Wintraecken

⁶⁴ is a faithful approximation of the isomanifold. Specifically, as shown in the full version ⁶⁵ of [10] and recalled in Section 2.2, if we take $\delta = \Omega(d^{2.5})$, the PL-approximation $\hat{\mathcal{M}}$ is ⁶⁶ $O(D^2)$ -close and isotopic to the isomanifold. Here the constants in the O depend on f and ⁶⁷ its derivatives. Hence, our algorithm constructs geometrically close and topologically correct ⁶⁸ PL-approximation of isomanifolds of bounded complexity in polynomial time.

⁶⁹ Our algorithm can be extended in several directions. First, the dependency on d in ⁷⁰ the size of $\hat{\mathcal{M}}$ can be completely removed by combining our algorithm with dimensionality ⁷¹ reduction (Section 3.4). We can also extend the algorithm to the case of isomanifolds with ⁷² boundary and, more generally, to stratifolds (Section 3.5).

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

The approximation of a manifold that is the zero set of a function is an example of the more general question of how to triangulate a manifold which has a long history in Mathematics. In particular, Whitney [30] introduced a construction that has some similarity with the present algorithm (see [7]). 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.

85 2 Background

⁸⁶ 2.1 Permutahedral representation of CFK-triangulations

In this section, we recall the definitions and basic properties of Freudenthal-Kuhn and Coxeter triangulations. We also briefly describe a compact data structure to store such triangulations which is fully described in [8].

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

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

The Coxeter triangulation of type \tilde{A}_d is the hyperplane arrangement \mathcal{H}_{E_C} in \mathbb{R}^{d+1} associated to the set of vectors $E_C = \{r_{i,j} = e_i - e_{j+1} | 1 \le i \le j \le d\}$, restricted to $\mathcal{H} \simeq \mathbb{R}^d$.

Two important facts follows. On one hand, Coxeter and Freudenthal-Kuhn triangulations are related by an affine transformation and have thus the same combinatorial structure. We call any triangulation that is the image of a Freudenthal-Kuhn triangulation under an affine transformation a CFK-triangulation. In this paper, we restrict our attention to Coxeter and Freudenthal triangulations since they are the most important but any CFK-triangulation could be used.

The second fact is that each simplex in such a triangulation can be represented as a cell in an arrangement of d(d-1)/2 hyperplanes which are known and do not need to be stored. The next crucial observation relates CFK-triangulations and permutahedra, which allows to represent CFK-triangulations in a compact way (The definition and some combinatorial properties of permutahedra are given in Appendix A). ▶ Proposition 2. The star of a vertex in a CFK-triangulation is dual¹ to a permutahedron
 (combinatorially).

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

¹¹⁷ CFK-triangulation. See Figure 1 from [8] for an illustration.



¹¹⁸ **Figure 1** The permutahedral representation of the simplices in the stars of vertices y and y'.

¹¹⁹ The following results are given in [8]

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

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

¹⁰⁹ ¹ Two complexes are dual if there is a bijection between their faces that inverses the inclusion relationships.

J.-D. Boissonnat, S. Kachanovich, and M. Wintraecken

Although all CFK-triangulations have the same combinatorial structure, they geometric properties are different (this point will be further discussed in Section 3.3).

¹³⁰ 2.2 PL-approximation of isomanifolds

¹³¹ We first recall sufficient conditions under which the PL-approximation $\hat{\mathcal{M}}$ output by the ¹³² algorithm faithfully reproduces the original isomanifold. These conditions are fully described ¹³³ in the full version of [10] and we simply state here the main results specialized to the case of ¹³⁴ CFK-triangulations.

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

137
$$\gamma_{\max} = \max_{x \in \mathcal{T}_0} \left(\max_i |\operatorname{grad} f^i(x)| \right)$$

138 $\lambda_{\min} = \min_{x \in \mathcal{T}_0} \lambda_{\min}(x),$

139 $\alpha_{\max} = \max_{x \in \mathcal{T}_0} \max_i \|\operatorname{Hes}(f^i)(x)\|_2$

140 where

 $\mathcal{T}_{141} = \mathcal{T}_0$ denotes the set of all $\sigma \in \mathcal{T}$, such that $(f^i)^{-1}(0) \cap \sigma \neq \emptyset$ for all i.

- $\operatorname{grad} f^{i} = (\partial_{j} f_{i})_{j}$ denotes the gradient of component f^{i} , for $i \in [1, d-n]$,
- Gram (∇f) denotes the Gram matrix whose elements are $\nabla f^i \cdot \nabla f^j$ where \cdot stands for the dot product.
- $\lambda_{\min}(x)$ denotes the smallest absolute value of the eigenvalues of $\operatorname{Gram}(\nabla f(x))^2$

Hes $(f) = (\partial_k \partial_l f_i)_{k,l}$ denotes the Hessian matrix of second order derivatives,

 $|\cdot|$ = $|\cdot|$ denotes the Euclidean norm of a vector and $||\cdot||_2$ the operator 2-norm of a matrix.³

We can now restate the topological result of [10]:

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

¹⁵⁴ Moreover, we can bound the Fréchet distance between \mathcal{M} and $\hat{\mathcal{M}}$. The Fréchet distance ¹⁵⁵ is a quite strong notion of distance and, in particular, it bounds the Hausdorff distance.

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

¹⁵⁹
$$d_F(\mathcal{M}_a, \mathcal{M}_b) = \inf_{h \in \mathcal{H}} \sup_{x \in \mathcal{M}_a} d(x, h(x)).$$

We can now state the other main result of [10]:

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

¹⁴⁵ ² Because a Gram matrix is a symmetric square matrix, its eigenvalues are well defined and real.

¹⁴⁸ ³ The operator norm is defined as $||A||_p = \max_{x \in \mathbb{R}^n} \frac{|Ax|_p}{|x|_p}$, with $|\cdot|_p$ the *p*-norm on \mathbb{R}^n .

3 Tracing isomanifolds

In this section, we describe an algorithm that computes a PL-approximation $\hat{\mathcal{M}}$ of an 164 isomanifold \mathcal{M} . The algorithm has some similarity with the Marching Cube algorithm [22] 165 but departs from it in two fundamental ways. First, because of the curse of dimensionality, 166 we cannot afford to look at all the cells in the grid and need to limit the search to cells 167 that are close to \mathcal{M} . The problem of computing $\hat{\mathcal{M}}$ can be naturally decomposed into two 168 subproblems: locating the various components of \mathcal{M} (i.e., finding at least one point in each 169 component), and then tracing around each component, using the fact that the components 170 are connected. This decomposition is used by various authors, see for example [29, 15]. In 171 this paper, we focus on the tracing problem, although we discuss very briefly (Section 3.2) 172 the problem of locating the components. As pointed out by Dobkin et al. many applications 173 supply their own starting points. 174

The second major difference with the original marching cube algorithm is to replace the usual cubical grid by a CFK-triangulation of the ambient space. Taking a CFK-triangulation instead of a grid is a major advantage in high dimensions that has been recognized in the pioneering works of Allgower, and Schmidt [3] and of Dobkin et al. [15], see also [23]. The novelty here is to use the data structure of Section 2.1 to represent a CFK-triangulation. As a consequence, we will keep two main advantages of using grids: very limited storage and fast basic operations.

182 3.1 Isomanifolds

Let $f : \mathbb{R}^d \to \mathbb{R}^{d-n}$ be a smooth (C^2 suffices) function, and suppose that 0 is a regular value of f, meaning that at every point x such that f(x) = 0, the Jacobian of f is non-degenerate. Then the zero set of f is an n-dimensional manifold as a direct consequence of the implicit function theorem, see for example [16, Section 3.5]. We further assume that $f^{-1}(0)$ is compact. As in [1] we consider a triangulation \mathcal{T} of \mathbb{R}^d . The function \hat{f} is the linear interpolation of the values of f at the vertices if restricted to a single simplex $\sigma \in \mathcal{T}$, i.e.

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

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

The PL-approximation is now defined as $\hat{f}^{-1}(0) = \hat{\mathcal{M}}$. Locally, $\hat{f}|_{\sigma}^{-1}(0)$ is generically 192 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 193 the gradients of $\hat{f}^i|_{\sigma}$ are linearly independent, which can be easily achieved by perturbing f 194 infinite simally (or at least its values at the vertices). Let τ_i^{d-n} and τ_i^{d-n-1} be faces of σ of 195 dimension d-n and d-n-1. An infinitesimal perturbation of f, can prevent either $\hat{f}|_{\sigma}^{-1}(0)$ 196 from intersecting the faces τ_i^{d-n-1} , or the gradients of $\hat{f}^i|_{\sigma}$ and the normal spaces of τ_i^{d-n} 197 (for each fixed i) from failing to span \mathbb{R}^d . More precise statements on the geometric and 198 topological stability of the triangulation under perturbations of f can be found in the full 199 version of [10, Section 5]. Because $\hat{f}|_{\sigma}^{-1}(0)$ is (generically) the intersection of an *n*-flat (H_{σ}) 200 and σ , it is an *n*-dimensional polytope denoted by C_{σ} . The PL-approximation or mesh $\hat{\mathcal{M}}$ 201 of \mathcal{M} is the polytopal cell complex obtained by gluing the polytopes C_{σ} associated to all the 202 simplices σ in \mathcal{T} . 203

²⁰⁴ 3.2 Manifold tracing algorithm

Let \mathcal{M} be the zero set of some function $f : \mathbb{R}^d \to \mathbb{R}^{d-n}$, and let $\hat{\mathcal{M}}$ be the associated PL-approximation defined over a triangulation \mathcal{T} of the ambient space \mathbb{R}^d . Both n, and d are known but arbitrary, and will be considered as parameters in the complexity analysis. We write k = d - n for the codimension of \mathcal{M} . The algorithm will use for \mathcal{T} a CFK-triangulation stored using the data structure from Section 2.1. We assume that the manifold $\hat{\mathcal{M}}$, and the triangulation \mathcal{T} satisfy the following genericity hypothesis:

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

We note that this condition can be satisfied by an infinitesimal perturbation for isomani-214 folds. This requires some explanation. We recall that the CFK-triangulation is a hyperplane 215 arrangement, and up to translation there are a finite number of k-flats that contain all 216 k-simplices in the CFK triangulation. Hypothesis 8 is not satisfied, if either the flat H_{σ} 217 is not linearly independent of these k-flats, or if H_{σ} does intersect some (k-1)-flat in 218 the CFK-triangulation. In the previous section, we have already seen that an infinitesimal 219 perturbation ensures that H_{σ} is n-dimensional. Because two affine spaces whose dimensions 220 do not add up to the ambient dimension don't intersect with generically and two affine spaces 221 whose dimensions add up to exactly the ambient dimension intersect in a single point, we see 222 that genericity can be achieved by perturbing f infinitesimally. refer to We further remark 223 that, generically, any vertex of the PL-approximation $\hat{\mathcal{M}}$ is the intersection point between a 224 k-simplex σ of \mathcal{T} with the *n*-flat H_{σ} that interpolates f inside σ . 225

26	Algorithm 1 Manifold tracing algorithm
226	input : the permutahedral representation of a triangulation \mathcal{T} of \mathbb{R}^d ,
	²²⁷ the codimension of the isomanifold $k = d - n$,
	228 a seed k-simplex $ au_0$ that intersects $\hat{\mathcal{M}}$
229	oracle : Given a k-simplex σ of \mathcal{T} , decide whether σ intersects H_{σ} and, in the
	affirmative, report the corresponding vertex $\sigma \cap H_{\sigma} = \sigma \cap \hat{\mathcal{M}}$.
230	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
231	Initialize the queue \mathcal{Q} and the set \mathcal{S} with τ_0
232	while the queue Q is not empty do
233	Pop a k-dimensional simplex τ from Q
234	for each cofacet ϕ of τ do
235	for each facet σ of ϕ do
236	if σ does not lie in S and intersects $\hat{\mathcal{M}}$ (which can be decided using the
	oracle) then
237	Insert σ in the queue Q
238	Insert σ together with the intersection point in S provided by the oracle

The algorithm essentially computes the set S of k-simplices of T that intersect $\hat{\mathcal{M}}$. The elements of S which are in 1-1 correspondence with the vertices of $\hat{\mathcal{M}}$ thanks to the Genericity hypothesis. The so-called intersection oracle is a basic ingredient of the algorithm: Intersection oracle: Given a k-simplex σ of \mathcal{T} , decide whether σ intersects H_{σ} and, in the affirmative, report the corresponding vertex $\sigma \cap H_{\sigma}$.

It is easy to see that the intersection oracle reduces to solving a linear system. Indeed, generically, a vertex is the intersection of a k-simplex σ of \mathcal{T} with the m-flat H_{σ} that interpolates f inside σ . One can compute the barycentric coordinates of $\sigma \cap H_{\sigma}$ by solving a linear system of k equations, and k unknowns. It then remains to check whether the barycentric coordinates are all non-negative (to ensure that the intersection point lies inside σ). It follows that the intersection oracle reduces to evaluating f at the k + 1 vertices of σ plus solving a $k \times k$ linear system.

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 244 problem of their construction. We simply observe that they can be obtained by computing a 245 critical point (e.g., a point with smallest x_1 -coordinate) on each connected component of the 246 isomanifold, which reduces to finding a solution to a system of equations, on which a large 247 body of literature exists. See for example [26, 25, 15] and also the discussion in Wenger's 248 book [29, Section 8.4]. Once such a seed point has been computed, we simply translate 249 and rotate the triangulation \mathcal{T} so that the seed point coincides with the barycenter of a 250 k-simplex of \mathcal{T} and the intersection with the manifold is transversal as demanded by the 251 genericity hypothesis (for numerical stability it is convenient if the angle between the tangent 252 space of the manifold and the starting k-simplex is large, which is easy to ensure). If the 253 distance between \mathcal{M} and $\hat{\mathcal{M}}$ is small enough, then $\hat{\mathcal{M}}$ also intersects the same k simplex, see 254 Section 2.2. 255

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

The algorithm maintains the subset \mathcal{S} of the simplices in \mathcal{T} of dimension k that intersect 260 \mathcal{M} . S is initialized with the seed simplex τ_0 and stored as a hash table so that we can decide 261 in constant time if a given k-simplex belongs to S. Then, starting from τ_0 , we look at all its 262 cofacets and consider all the facets of those cofacets that are not in \mathcal{S} (i.e. they have not 263 been considered yet). This can be done using a queue \mathcal{Q} of candidate k-simplices. Each of 264 these simplices is queried with the intersection oracle and, if it is found to intersect $\hat{\mathcal{M}}$, it 265 is added to \mathcal{S} if not already present. Upon termination, \mathcal{S} contains all the k-dimensional 266 simplices of \mathcal{T} that intersect \mathcal{M} . Each such intersection, which consists of a single point (by 267 the Genericity hypothesis), is a vertex of $\hat{\mathcal{M}}$. Hence $\hat{\mathcal{M}}_0$ is the vertex set of $\hat{\mathcal{M}}$. 268

Note that our algorithm essentially traverses the adjacency graph of the k and (k+1)-269 simplices of \mathcal{T} that intersect $\hat{\mathcal{M}}$. It therefore identifies not only the set $\hat{\mathcal{M}}_0$ of vertices of $\hat{\mathcal{M}}$, 270 but also the edges joining two such vertices (associated to the cofacets of the k-simplices in 271 \mathcal{S}). By simply reporting those cofacets on the fly, the algorithm can output the 1-skeleton 272 $\hat{\mathcal{M}}_1$ of the *n*-dimensional polytopal cell complex $\hat{\mathcal{M}}$. The higher dimensional faces of $\hat{\mathcal{M}}$ are 273 the polytopes $C_{\tau} = \tau \cap H_{\tau}$ for all the cofaces τ of the k-simplices of S. If needed, the full 274 Hasse diagram of $\hat{\mathcal{M}}$ can be computed from $\hat{\mathcal{M}}_0$. This can be done in an output sensitive 275 manner by using the permutahedral representation of \mathcal{T} and the algorithm of Section 2.1 to 276 compute cofaces by increasing dimensions. 277

278 3.3 Complexity analysis

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

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

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

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

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

We see that Coxeter triangulations lead to smaller samples than FK-triangulations by a factor of roughly 2^n . This will be confirmed experimentally (see Figure 3).

As noticed in Section 3.2, a simple variant of the algorithm can compute the full Hasse diagram of $\hat{\mathcal{M}}$ in an output sensitive manner. The following lemma shows that the combinatorial complexity of $\hat{\mathcal{M}}$ is of the same order as the combinatorial complexity as $\hat{\mathcal{M}}_0$.

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

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

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

312 3.4 Dimensionality reduction

As seen from Proposition 10, the size $|\mathcal{S}|$ of the output of the algorithm, considered as a function of the resolution D of the triangulation, depends exponentially on n (which is to be expected), and only polynomially on d (which is fortunate). Nevertheless, the computing time of our algorithm and the size of the output depend on d. Removing the dependency on d in the time complexity is impossible since we need to evaluate a vectored-valued function f at a number of points of \mathbb{R}^d , which takes $\Omega(d)$ time per evaluation. However, we will see that we can reduce the size of the mesh produced by our algorithm.

Examples of samples of \mathcal{M} whose sizes depend on n but not on d, and lead to good 320 approximations are known. Especially important are D-nets [12, 6]. A D-net consists of 321 a finite number of sample points of \mathcal{M} such that no point of \mathcal{M} is at distance more than 322 D from a sample point (density condition), and no two sample points are closer than cD323 for some positive constant c (separation condition). A simple volume argument shows that 324 the size of a D-net of a n-dimensional smooth submanifolds is $O(1/D^n)$ [5, Lemma 5.3]. 325 The sample produced by our algorithm is *D*-dense on the piecewise linear approximation. 326 This implies that we have a sample that has a Hausdorff distance of $D + d_F(\mathcal{M}, \hat{\mathcal{M}})$ to the 327 manifold, where $d_F(\mathcal{M}, \hat{\mathcal{M}})$ is bounded in Theorem 7. 328

Since its cardinality depends on d, it is not well separated and, in particular, not a D-net of \mathcal{M} . If we are mostly interested in the output sample, we can easily sparsify it to obtain a D-net. However, by doing so, we will loose the combinatorial structure of the mesh.

We now show how to compute a *D*-dense sample of \mathcal{M} of size independent of *d*, together with a mesh. Specifically, we will reduce dimensionality using a variant of the celebrated Johnson-Lindenstrauss lemma for manifolds. Doing so, we depart from our previous worstcase analysis by allowing some approximation factor ε and tolerate a guarantee that holds only with high probability.

Theorem 13 (Johnson-Lindenstrauss lemma for manifolds [14, 28])). Pick any ε, η > 0, and let d' = Ω $\left(\frac{n}{\varepsilon^2} \log \frac{1}{\varepsilon} + \frac{1}{\varepsilon^2} \log \frac{\Gamma}{\eta}\right)$, where Γ is a quantity that depends only on intrinsic properties of \mathcal{M} . Let Φ be the projection on a random affine subspace of dimension d'. Then, with probability > 1 − η, for all x, y ∈ \mathcal{M} , we have $(1 - \varepsilon) \sqrt{\frac{d'}{d}} \le \frac{\|\Phi x - \Phi y\|}{\|x - y\|} \le (1 + \varepsilon) \sqrt{\frac{d'}{d}}$.

Let $\Psi = \sqrt{\frac{d}{d'}} \Phi$. By the theorem, the image $\Psi(\mathcal{M})$ of \mathcal{M} is a submanifold of dimension n341 embedded in $\mathbb{R}^{d'}$. One can now run the manifold tracing algorithm in $\mathbb{R}^{d'}$ to sample, and 342 mesh $\Psi(\mathcal{M})$. The algorithm works as described before except that we need another oracle 343 that, given a (d'-n)-simplex σ of the CFK-triangulation of $\mathbb{R}^{d'}$, decides whether its inverse 344 image $\Psi^{-1}(\sigma)$ intersects \mathcal{M} or not. Note that $\Psi^{-1}(\sigma)$ is a (d-d')-dimensional flat strip 345 (that is the product of a face and an affine subspace) in \mathbb{R}^d , and that the complexity of this 346 new oracle is the same as the complexity of the basic intersection oracle, i.e. polynomial in d. 347 Due to the scaling factor $\sqrt{d/d'}$, the resolution of the triangulation in the low dimensional 348 space $\mathbb{R}^{d'}$ has to be scaled by the same factor if one wants to satisfy a given sampling density 349 on \mathcal{M} . Since the geometry of the manifold is also scaled in the same way [18], the analysis 350 of the algorithm will be unchanged. Proposition 10 then shows that the size of the output 351 sample does not depend on d but only on n and D for fixed ε , and η . Moreover, since the 352 complexities of the projection and of the new oracle are polynomial in d, Proposition 9 353 implies that the overall complexity is still polynomial in d. 354

355 3.5 Isomanifolds with boundary, and stratifolds

The case of isomanifolds with boundary and, more generally, of isostratifolds can be handled in very much the same way. By an isomanifold of dimension n with boundary, we mean that, on top of a function $f : \mathbb{R}^d \to \mathbb{R}^{d-n}$, we are given another function $f_{\partial} : \mathbb{R}^d \to \mathbb{R}$, and the set we consider is $\mathcal{M} = f^{-1}(0) \cap f_{\partial}^{-1}([0,\infty))$. We note that $\partial \mathcal{M} = f^{-1}(0) \cap f_{\partial}^{-1}(0)$. Similarly to (1), we also define $\hat{f}_{\partial}|_{\tau}(x) = \sum_{i=1}^{j+1} \lambda_i(x) f_{\partial}(p_i)$. We write \hat{f} for the (global)

Similarly to (1), we also define $f_{\partial|_{\tau}}(x) = \sum_{i=1}^{j+1} \lambda_i(x) f_{\partial}(p_i)$. We write 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 $\partial \hat{\mathcal{M}} = \hat{f}_{\partial}^{-1}(0) \cap \hat{f}^{-1}(0)$ is a subset of $\hat{f}^{-1}(0)$, i.e. the

J.-D. Boissonnat, S. Kachanovich, and M. Wintraecken

- ³⁶⁴ piecewise linear approximation of the manifold ignoring the boundary. The piecewise linear
- approximation $\hat{\mathcal{M}}$ of the manifold with boundary consists of the following cells:

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

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

We will assume that the Genericity Hypothesis 8 holds for both $\hat{\mathcal{M}}$, and $\partial \hat{\mathcal{M}}$.

We can now adapt the algorithm of Section 3.2 as follows. In addition to reporting the set S_k of k-faces of the triangulation \mathcal{T} that intersect $\hat{\mathcal{M}}$, the algorithm will also report the set S_{k+1} of (k+1)-faces of the triangulation \mathcal{T} that intersect $\partial \hat{\mathcal{M}}$. The computation of S_{k+1} is done by the following simple modification of Lines 6-9: if the k-dimensional facet σ of τ intersects $\hat{f}^{-1}(0)$ at a point x such that $\hat{f}_{\partial}|_{\tau}(x) < 0$ (i.e. x is not in $\hat{\mathcal{M}}$), we then compute the intersection point of τ with $\hat{f}_{\partial}^{-1}(0)$, and put τ in S_{k+1} .

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

1. Two simplices σ_1 , and σ_2 of S_k are joined by an edge in $\hat{\mathcal{M}}_1$ if and only if there exists a simplex in \mathcal{T}_{k+1} with faces σ_1 , and σ_2 .

2. Two simplices τ_1 , and τ_2 of S_{k+1} are joined by an edge in $\partial \mathcal{M}_1$ if and only if there exists a simplex in \mathcal{T}_{k+2} with faces τ_1 , and τ_2 .

386 **3.** A simplex σ of S_k , and a simplex τ of S_{k+1} are joined by an edge in $\widehat{\partial M}_1$ if and only if 387 σ is a facet of τ .

The three rules above together with the permutahedral representation of \mathcal{T} provide a way 388 to construct the 1-skeleton of $\hat{\mathcal{M}}$ on the fly. The total cost is output sensitive. If needed, the 389 entire combinatorial structure of $\hat{\mathcal{M}}$ can be computed by traversing the full triangulation \mathcal{T} . 390 The above construction generalizes easily to arbitrary isostratifolds. Isostratifolds are 391 stratifolds that are defined by equations, and inequalities. An example of such a stratifold is 392 an octant of the sphere in \mathbb{R}^3 that can be defined by as $x^2 + y^2 + z^2 - 1 = 0, x \ge 0, y \ge 0$, 393 and $z \ge 0$. We compute the 1-skeleton of $\hat{\mathcal{M}}$, and construct a graph whose nodes are the 394 simplices of dimensions k, k+1, ..., d that intersect the strata of dimension n, n-1, ..., 0. 395

396 4 Experimental results

The data structure of Section 2.1 and the algorithm of Section 3 have been implemented in C++. The code is robust and fast and will be released in the GUDHI library [20]. Full detail on the implementation, including the implementation of the oracle, can be found in [21].

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

403 4.1 Performance of the algorithm

We show the performance of our implementation of the manifold tracing algorithm for various ambient and intrinsic dimensions in Figure 2. In Figure 3, we can see that using



Figure 2 The effect of the ambient dimension d and of the intrinsic dimension n on the computation time of the of the manifold tracing algorithm. The reconstructed manifold in the tests is the ndimensional 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.

406 Coxeter triangulation is beneficial in practice as it produces a smaller output in less time
407 (see Proposition 10).

In Figure 4, we present a PL approximation of a two-dimensional flat torus without boundary embedded in \mathbb{R}^{10} built by the manifold tracing algorithm. The 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 flat torus in \mathbb{R}^3 .



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



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

427 4.2 Manifolds with boundary

 $_{\mathtt{428}}$ The algorithm has been adapted to handle submanifolds with boundary and surfaces with a

⁴²⁹ piecewise smooth boundary, see Appendix 3.5. In Figure 5, we present the mesh obtained by

⁴³⁰ our algorithm on a portion of a flat torus embedded in \mathbb{R}^4 , and cut by a hypersphere. The ⁴³¹ torus has been rotated and translated in \mathbb{R}^4 so that the coordinate axes do not play any

432 special role.



Figure 5 Four views of the flat torus in \mathbb{R}^4 given by two equations $x_1^2 + x_2^2 = 1$, and $x_3^2 + x_4^2 = 1$ cut by the hypersphere $(x_1-1)^2 + x_2^2 + (x_3-1)^2 + x_4^2 = 4$, projected to \mathbb{R}^3 . The ambient triangulation used is a Coxeter triangulation of type \tilde{A}_4 with the diameter 0.15 of the full-dimensional simplices. The reconstructed boundary is highlighted in yellow. The size $|\mathcal{S}|$ of the piecewise-linear approximation is 14779. 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.

439 4.3 An application in algebraic geometry

We also applied our algorithm to a more complicated example of interest in algebraic geometry [4] where an active field of research is to understand the geometry and topology of various projective varieties. Projective varieties are isomanifolds defined by polynomial equations in the complex projective space $\mathbb{CP}^d = (\mathbb{C}^{d+1} \setminus 0)/\mathbb{C}^*$ of complex dimension d. One such example is the complex one-dimensional curve (that is a real dimensional surface) given by the equation $z_1^2 \bar{z}_2 + z_2^2 \bar{z}_3 + z_3^2 \bar{z}_1 = 0$ in \mathbb{CP}^2 , where \bar{z} denotes the conjugate of the complex number z. To be able to apply our algorithm, we first need to pass from homogenous coordinates $[z_1 : \ldots : z_{d+1}]$ on \mathbb{CP}^d to affine coordinates $[z'_1 : \cdots : z'_{i-1} : 1 : z'_{i+1} : \cdots : z'_{d+1}]$ by picking the *i*th coordinate to be equal to 1, that is $z'_j = z_j/z_i$. Given some homogenous coordinates $[z_1 : \ldots : z_{d+1}]$, we can choose the *i*th coordinate to be set to 1 to be the coordinate whose absolute value is the largest, so that \mathbb{CP}^d can be written as the union of the d + 1 sets $\{[z'_1 : \cdots : z'_{i-1} : 1 : z'_{i+1} : \cdots : z'_{d+1}] \mid |z'_j| \leq 1\}$, with the boundaries of these sets identified. Writing $z'_i = x_j + iy_j$ these sets are (seen as real sets) identical to the domain of \mathbb{R}^{2d}

⁴⁵⁴
$$D_i = \{(x_1, y_1, \dots, x_{i-1}, y_{i-1}, x_{i+1}, y_{i+1}, \dots, x_{d+1}, y_{d+1}) \mid x_i^2 + y_i^2 \le 1\}.$$

Let f be a homogenous polynomial in d+1 complex variables and their complex conjugates. 455 For each *i*, we can fix the *i*th coordinate to be 1. Writing each variable in terms of its real 456 and imaginary part yields a real inhomogeneous polynomial in 2d (real) variables on the 457 domain D_i . Taking the real and imaginary parts of the function yields two real functions 458 $f_{\mathfrak{R},i}$ and $f_{\mathfrak{I},i}$ on D_i . As real sets, the projective variety f = 0 on \mathbb{PC}^d and the intersection 459 of the sets $f_{\mathfrak{R},i} = 0$ and $f_{\mathfrak{I},i} = 0$ on D_i for each *i* (with the boundaries identified) are the 460 same. We can therefore apply the tracing algorithm to each isomanifold $(f_{\mathfrak{R},i}=0,f_{\mathfrak{I},i}=0)$ 461 of D_i independently. Since their boundaries coincide, we can then glue these isomanifolds 462 along their boundary to obtain a PL-approximation of the projective variety f = 0. This, for 463 example, allows to recover the Euler characteristic of f = 0 on \mathbb{PC}^d . 464

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



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

We illustrate the above construction on the above equation $z_1^2 \bar{z}_2 + z_2^2 \bar{z}_3 + z_3^2 \bar{z}_1 = 0$ in 469 \mathbb{PC}^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 470 $\bar{z}_2 + z_2^2 \bar{z}_3 + z_3^2 = 0$. 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 471 equations for each of the complex equations. We give those corresponding to $z_1^2 \bar{z}_2 + z_2^2 + \bar{z}_1 = 0$, 472 the other equations being symmetric. For this complex equation, we get the real equations 473 $x_1 + x_1^2 x_2 + x_2^2 - x_2 y_1^2 + 2x_1 y_1 y_2 - y_2^2 = 0$ and $-y_1 + 2x_1 x_2 y_1 - x_1^2 y_2 + 2x_2 y_2 + y_1^2 y_2 = 0$ in 474 \mathbb{R}^4 . The domain D_3 is in this case determined by the equations $x_1^2 + y_1^2 \leq 1$ and $x_2^2 + y_2^2 \leq 1$. 475 Hence we find a surface in \mathbb{R}^4 with a piecewise smooth boundary. The result provided by 476 our algorithm is shown in Figure 6. For visualization purposes, we show the three surfaces 477 separately and projected from \mathbb{R}^4 to \mathbb{R}^3 . 478

479 **5** Conclusion and open questions

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

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

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

⁴⁹² 1. All k-dimensional faces τ in \mathcal{T} , with $k \leq d - n - 1$, are far enough from \mathcal{M} .

 $_{493}$ 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 494 equivalent and close in Hausdorff distance to the input manifold [7]. However, in the worst 495 case, we need to perturb all the simplices of \mathcal{T} of dimension less than the codimension 496 d-n that are incident on a vertex. Since there are exponentially many such simplices, such 497 methods have a complexity that depends exponentially on the ambient dimension d, and 498 have not proved useful in practice except in some simple cases. It remains open whether 499 general smooth manifolds embedded in \mathbb{R}^d can be triangulated in time polynomial in d as we 500 were able to do here in the special case of isomanifolds. 501

502		References
503	1	Eugene Allgower and Kurt Georg. Estimates for piecewise linear approximations of implicitly
504		defined manifolds. Applied Mathematics Letters, 2(2):111–115, 1989.
505	2	Eugene Allgower and Kurt Georg. Numerical continuation methods: an introduction, volume 13.
506		Springer Science & Business Media, 1990.
507	3	Eugene Allgower and Phillip H. Schmidt. An algorithm for piecewise-linear approximation of
508		an implicitly defined manifold. SIAM journal on numerical analysis, 22(2):322–346, 1985.
509	4	Aurélien Alvarez and Bertrand Deroin. Dynamique et topologie du feuilletage de Jouanolou.
510		Preprint, 2019.
511	5	Jean-Daniel Boissonnat, Frédéric Chazal, and Mariette Yvinec. Geometric and Topological
512		Inference. Cambridge Texts in Applied Mathematics. Cambridge University Press, 2018.
513		doi:10.1017/9781108297806.
514	6	Jean-Daniel Boissonnat and Arijit Ghosh. Manifold reconstruction using tangential Delaunay
515		complexes. Discrete & Computational Geometry, 51(1):221–267, 2014.
516	7	Jean-Daniel Boissonnat, Siargey Kachanovich, and Mathijs Wintraecken. Triangulating
517		submanifolds: An elementary and quantified version of Whitney's method. hal-01950149,
518		December 2018.
519	8	Jean-Daniel Boissonnat, Siargey Kachanovich, and Mathijs Wintraecken. A compact data
520		structure for high dimensional Coxeter-Freudenthal-Kuhn triangulations. Submitted to the
521		International Symposium on Computational Geometry (SoCG2021), 2020.
522	9	Jean-Daniel Boissonnat, Siargey Kachanovich, and Mathijs Wintraecken. Tracing isomanifolds
523		in \mathbb{R}^d in time polynomial in d. Submitted to the International Symposium on Computational
524		Geometry (SoCG2021), 2020.
525	10	Jean-Daniel Boissonnat and Mathijs Wintraecken. The topological correctness of PL-
526		approximations of isomanifolds. In 34th International Symposium on Computational Ge-
527		ometry, SoCG 2020, June 23-26, 2020, Zurich, Switzerland., 2020. Full version. URL:
528		https://hal.inria.fr/hal-02386193.
529	11	Yen-Chi Chen. Solution manifold and its statistical applications, 2020. arXiv:2002.05297.
530	10	arXiv:2002.05297.
531	12	Siu-Wing Cheng, Tamal K Dey, and Edgar A Ramos. Manifold reconstruction from point
532	10	samples. In $SODA$, volume 5, pages 1018–1027, 2005.
533	13	Aruni Choudhary, Siargey Kachanovich, and Mathijs Wintraecken. Coxeter triangulations
534		have good quality. Mainematics in Computer Science, 14:141–170, 2020. URL: https://doi.org/10.1007/g11786-020-00461-5
535	1/	Kannath I. Clarkson Tichter hounds for rendem projections of manifolds. In Proceedings
536	14	of the 2/th ACM Sumposium on Computational Geometry, College Park, MD, USA, June
537		9.11 2008 pages 39-48 2008 UBL: https://doi.org/10.1145/1377676.1377685.doi:
530		10 1145/1377676 1377685
540	15	David P Dobkin Allan B Wilks Silvio V F Levy and William P Thurston Contour tracing
541	10	by piecewise linear approximations. ACM Transactions on Graphics (TOG), 9(4):389–423.
542		1990.
543	16	J. J. Duistermaat and J. A. C. Kolk. Multidimensional Real Analysis I: Differentiation.
544	-	Number 86 in Cambridge Studies in Advanced Mathematics. Cambridge University Press,
545		2004.
546	17	B. Curtis Eaves. A course in triangulations for solving equations with deformations, volume
547		234. Lecture Notes in Economics and Mathematical Systems, 1984.
548	18	Armin Eftekhari and Michael B. Wakin. What happens to a manifold under a bi-lipschitz
549		map? Discrete & Computational Geometry, 57(3):641-673, 2017. URL: https://doi.org/10.
550		1007/s00454-016-9847-6, doi:10.1007/s00454-016-9847-6.
551	19	A. Gomes, I. Voiculescu, J. Jorge, B. Wyvill, and C. Galbraith. Implicit Curves and Surfaces:
552		Mathematics, Data Structures and Algorithms. Springer, 2009.

- 553 20 GUDHI Project "http://gudhi.gforge.inria.fr/doc/latest/" . URL: http://gudhi.gforge. 554 inria.fr/doc/latest/.
- 555 21 Siargey Kachanovich. Meshing submanifolds using Coxeter triangulations. Theses, COMUE Uni-
- versité Côte d'Azur (2015 2019), October 2019. URL: https://hal.inria.fr/tel-02419148.
- ⁵⁵⁷ 22 William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3d surface ⁵⁵⁸ construction algorithm. *ACM siggraph computer graphics*, 21(4):163–169, 1987.
- Chohong Min. Simplicial isosurfacing in arbitrary dimension and codimension. Journal of Computational Physics, 190(1):295–310, 2003.
- Timothy S. Newman and Hong Yi. A survey of the marching cubes algorithm. Computers & Graphics, 30(5):854 879, 2006. URL: http://www.sciencedirect.com/science/article/
 pii/S0097849306001336, doi:https://doi.org/10.1016/j.cag.2006.07.021.
- James M Ortega and Werner C Rheinboldt. Iterative solution of nonlinear equations in several
 variables. SIAM, 2000.
- Alexander M Ostrowski. Solution of Equations and Systems of Equations: Pure and Applied
 Mathematics: A Series of Monographs and Textbooks, Vol. 9, volume 9. Elsevier, 2016.
- Michael J. Todd. *The computation of fixed points and applications*, volume 124. Lecture Notes in Economics and Mathematical Systems, 1976.
- Nakul Verma. A note on random projections for preserving paths on a manifold. Technical
 Report Tech. Report CS2011-0971, UC San Diego, 2011.
- Rephael Wenger. Isosurfaces: geometry, topology, and algorithms. AK Peters/CRC Press, 2013.
- 574 30 H. Whitney. Geometric Integration Theory. Princeton University Press, 1957.

575 **A** Permutahedra

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

577 • Definition 14 (Permutahedron). A d-permutahedron is a d-dimensional polytope, which is

the convex hull \mathcal{P} of all points in \mathbb{R}^{d+1} , the coordinates of which are permutations of [d+1]. Formally, this convex hull can be written as:

580
$$\mathcal{P} = \operatorname{conv}\{(\sigma(1), \dots, \sigma(d+1)) \in \mathbb{R}^{d+1} \mid \sigma \in \mathfrak{S}_{d+1}\},\$$

where \mathfrak{S}_{d+1} denotes the set of permutations of [d+1].



- ⁵⁸² **Figure 7** The 2-permutahedron and the ordered partitions associated to its faces.
- ⁵⁸³ The following two corollaries are proved in [9].

Solution Corollary 15. The number of facets of an *i*-face σ of a *d*-permutahedron is at most 2^{i+1} .

Corollary 16. Let $p_{i,j}$ denote the number of *i*-faces of a *j*-face of the *d*-permutahedron. We have $p_{i,j} \leq \frac{1}{2\min(i,d-i)} {j \choose i} (j+1)!$

587 **B Proofs**

Proof of Proposition 9. The complexity of the initialization is O(d). The complexity of each 588 iteration of the while loop consists of computing the cofacets of the popped k-dimensional 589 simplex in the queue, computing facets of these cofacets, and applying the intersection oracle 590 on each of these facets. An upper bound on the number of cofacets of a k-simplex in a 591 CFK-triangulation follows, by duality, from Corollary 15, specifically $O(2^n)$. Each of these 592 cofacets has k+2 facets. Therefore, for each iteration of the while loop, the algorithm applies 593 the intersection oracle on $O(k2^n)$ simplices. By using this observation, and the complexities 594 in Lemmas 3, and 4, the total time complexity of each iteration of the while loop follows: 595

596
$$O(d2^n) + O(dk2^n) + O(k2^nI) = O(k2^n(d+I)) = O(k2^nI).$$

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

Proof of Proposition 10. By the definition of CFK-triangulations in Section 2.1, \mathcal{T} is an arrangement of d(d-1)/2 families H_u of hyperplanes, $u \in E_{\mathcal{T}}$. Each family H_u consists of the hyperplanes $H_{u,k}, k \in \mathbb{Z}$, all orthogonal to u. Let $L_{\mathcal{T}}$ be the length of the longest edge of a simplex in \mathcal{T} and R_{τ} be the maximal norm of the vectors u. Note that the distance between two consecutive hyperplanes in family H_u is $1/||u|| \ge 1/R_{\tau}$.

We will rescale the arrangement of hyperplanes so that the maximal diameter of the 603 simplices is D, the required precision. Hence the distance between two consecutive hyperplanes 604 in H_u is $D/(L_T ||u||)$. It follows that at most $\sqrt{dL_T ||u||}/D$ hyperplanes of family H_u intersect 605 the unit cube C_d that contains \mathcal{M} (which has diameter \sqrt{d}). Consider any subset of n 606 families among the d(d-1)/2 families, and write I for the associated subset of indices, 607 $I \subset [1, d(d-1)/2], |I| = n$. Now take n hyperplanes, one in each family $H_{u_i}, i \in I$. Their 608 common intersection is an affine space of dimension k = d - n. This affine space intersects \mathcal{M} 609 in at most K points under the general position assumption and the fact that \mathcal{M} is K-sparse. 610 The total number of intersection points $N_{\mathcal{T}} = \mathcal{T}_k \cap \mathcal{M}$ is thus bounded as follows 611

$${}^{_{612}} \qquad N_{\mathcal{T}} \le K \left(\frac{d(d-1)/2}{n} \right) \times \prod_{i \in I} \frac{\sqrt{dL_{\mathcal{T}}} \|u_i\|}{D} \le \frac{K}{n!} \times \left(\frac{d^2 \sqrt{dL_{\mathcal{T}}} R_{\mathcal{T}}}{2D} \right)^n.$$

$$\tag{2}$$

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 2.1 that $R_C = R_{FK} = \sqrt{2}$. The longest edge L_{FK} in a Freudenthal-Kuhn triangulation has length at most (in fact exactly) \sqrt{d} since each simplex is contained in a cubical cell of the *d*-dimensional unit grid. Furthermore, it is proved in [13, point 6 of \tilde{A}_d in Section 6] that the longest edge length in the Coxeter triangulation of type \tilde{A}_d is

$$L_C = \begin{cases} \frac{\sqrt{d+1}}{2} & \text{if } d \text{ is odd,} \\ \frac{1}{2}\sqrt{\frac{d(d+2)}{(d+1)}} & \text{if } d \text{ is even,} \end{cases}$$
(3)

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

621

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

623

622

Proof of Proposition 11. Let σ be a k-simplex of a CFK-triangulation that intersects $\hat{\mathcal{M}}$, and let σ^* be its dual cell. By definition, σ^* is a m-dimensional face of \mathcal{T}^* , the polytopal cell complex dual to \mathcal{T} . The collection of all σ^* associated to the k-simplices σ of \mathcal{T} that intersect $\hat{\mathcal{M}}$ form a cell complex $\hat{\mathcal{M}}^*$ dual to $\hat{\mathcal{M}}$. To bound the number of faces of all dimensions of the PL-approximation $\hat{\mathcal{M}}$, it is therefore sufficient to bound the number of faces of $\hat{\mathcal{M}}^*$.

Each *d*-dimensional cell in \mathcal{T}^* is a permutahedron (Proposition 2). Hence, σ^* is a *n*-face of a *d*-permutahedron. The number of faces of σ^* of dimensions 0 to n-1 (or equivalently the number of cofaces of σ of dimensions n+1 to *d*) is:

632
$$\sum_{i=0}^{n-1} p_{i,n} \le \sum_{i=0}^{n-1} \frac{1}{2^i} \binom{n}{i} (n+1)! = \frac{3^n - 1}{2^n} (n+1)!$$

where $p_{i,j}$ denotes the number of *i*-faces of a *j*-face of the *d*-permutahedron and is bounded in Corollary 16. The last equality can be easily verified using Mathematica. The overall

 $_{635}$ $\,$ combinatorial complexity of $\hat{\mathcal{M}}$ is therefore

636
$$|\mathcal{S}| \times \frac{3^n - 1}{2^n} (n+1)!,$$

 $_{\rm 637}$ $\,$ where ${\cal S}$ is bounded in Proposition 10.

◀