Meshing submanifolds using Coxeter triangulations

Siargey Kachanovich

UNIVERSITÉ CÔTE D'AZUR

ÉCOLE DOCTORALE

SCIENCES ET TECHNOLOGIES DE L'INFORMATION ET DE LA COMMUNICATION

THÈSE DE DOCTORAT

Maillage de variétés avec les triangulations de Coxeter

Siargey KACHANOVICH

Inria Sophia Antipolis – Méditerranée, Équipe-projet Datashape

Présentée en vue de l'obtention du grade de docteur en informatique
d'Université Côte d'Azur
Dirigée par : Jean-*Daniel BOISSONNAT*
Co-encadrée par : M*athijs WINTRAECKEN*
Soutenue le : 23/10/2019

Devant le jury, composé de :
Pierre Alliez, Inria Sophia Antipolis
Aurélien Alvarez, ENS Lyon
Dominique Attali, CNRS & Gipsa-lab
Jean-Daniel Boissonnat, Inria Sophia Antipolis
André Lieutier, Dassault Systèmes
Vincent Pilaud, CNRS & École polytechnique
Mathijs Wintraecken, IST Austria

गुडी GUDHI Geometry Understanding in Higher Dimensions

Inria informatiques mathématiques

Maillage de variétés avec les triangulations de Coxeter

Jury :

Président du jury :	Pierre ALLIEZ	INRIA Sophia Antipolis – Méditerranée
Rapporteurs :	Aurélien ALVAREZ	ENS Lyon
	Dominique ATTALI	Gipsa-lab
	Vincent PILAUD	École Polytechnique
Examinateurs :	Jean-Daniel BOISSONNAT	INRIA Sophia Antipolis – Méditerranée
	Mathijs WINTRAECKEN	INRIA Sophia Antipolis – Méditerranée
	André LIEUTIER	Dassault Systèmes

Mis en page avec la classe thesul.

Résumé

Mots-clés: Génération de maillages, triangulations de Coxeter, triangulations de Freudenthal-Kuhn, qualité des simplexes, triangulations de l'espace Euclidien.

Cette thèse s'adresse au problème du maillage d'une variété donné dans une dimension arbitraire. Intuitivement, on peux supposer que l'on est donné une variété — par exemple l'interieur d'un tore plongé dans $\mathbb{R}^9$, et notre objectif est de construire une maillage de cette variété (par exemple une triangulation).

Nous proposons trois contributions principales. La première est l'*algorithme du tracé des variétés* qui reconstruit un complexe cellulaire approchant une variété compacte et lisse de dimension m dans l'espace Euclidien $\mathbb{R}^d$, pour m et d arbitraires. L'algorithme proposé utilise une triangulation $\mathcal{T}$ qui est supposé d'être une transformation linéaire de la triangulation de Freudenthal-Kuhn de $\mathbb{R}^d$. La complexité dépend linéairement de la taille de la sortie dont chaque élément est calculé en temps seulement polynomial en la dimension ambiante d. Cet algorithme nécessite que la variété soit accedée par un oracle d'intersection qui répond si un simplexe $(d-m)$-dimensionel donné intersecte la variété. À ce titre, ce cadre est général et couvre plusieures représentations des variétés populaires, telles que le niveau d'une fonction multivariée ou les variétés données par un nuage de points.

Notre deuxième contribution est une structure de données qui représente la triangulation de Freudenthal-Kuhn de R^d. À chaque moment de l'execution, l'espace utilisé par la structure de données est au plus $O(d^2)$. La structure de données supporte plusieurs opérations d'une manière efficace telles que la localisation d'un point dans la triangulation et accès aux faces et cofaces d'un simplexe donné. Les simplexes dans une triangulation de Freudenthal-Kuhn de $\mathbb{R}^d$ sont encodés par une nouvelle représentation qui généralise celle de Freudenthal pour les simplexes d-dimensionels [Fre42].

Enfin, nous étudions la géométrie et la combinatoire des deux types de triangulations étroitement liés : des triangulations de Freudenthal-Kuhn et des triangulations de Coxeter. Pour les triangulations de Coxeter, on démontre que la qualité des simplexes d-dimensionels est $O(1/\sqrt{d})$ comparé au simplexe régulier. Par ailleurs, nous établissons lesquelles des triangulations sont de Delaunay. Nous considérons aussi l'extension de la propriété d'être Delaunay qui s'appelle la protection et qui mesure la généricité de la triangulation de Delaunay. En particulier, nous montrons qu'une famille de triangulations de Coxeter atteint la protection $O(1/d^2)$. Nous proposons une conjecture que les deux bornes sont optimaux entre les triangulations de l'espace Euclidien.

Abstract

Keywords: Mesh generation, Coxeter triangulations, Freudenthal-Kuhn triangulations, simplex quality, triangulations of the Euclidean space.

This thesis addresses the manifold meshing problem in arbitrary dimension. Intuitively, suppose we are given a manifold — such as the interior of a torus — embedded in a space like $\mathbb{R}^9$, our goal is to build a mesh of this manifold (for example, a triangulation).

We propose three principal contributions. The central one is the *manifold tracing algorithm*, which constructs a piecewise-linear approximation of a given compact smooth manifold of dimension m in the Euclidean space $\mathbb{R}^d$, for any m and d. The proposed algorithm operates in an ambient triangulation $\mathcal{T}$ that is assumed to be an affine transformation of the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$. It is output-sensitive and its time complexity per computed element in the output depends only polynomially on the ambient dimension d. It only requires the manifold to be accessed *via* an intersection oracle that answers if a given $(d-m)$-dimensional simplex in $\mathbb{R}^d$ intersects the manifold or not. As such, this framework is general, as it covers many popular manifold representations such as the level set of a multivariate function or manifolds given by a point cloud.

Our second contribution is a data structure that represents the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$. At any moment during the execution, this data structure requires at most $O(d^2)$ storage. With this data structure, we can access in a time-efficient way the simplex that contains a given point, the faces and the cofaces of a given simplex. The simplices in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ are encoded using a new representation that generalizes the representation of the d-dimensional simplices introduced by Freudenthal [Fre42].

Lastly, we provide a geometrical and combinatorial study of the Freudenthal-Kuhn triangulations and the closely-related Coxeter triangulations. For Coxeter triangulations, we establish that the quality of the simplices in all d-dimensional Coxeter triangulations is $O(1/\sqrt{d})$ of the quality of the d-dimensional regular simplex. We further investigate the Delaunay property for Coxeter triangulations. Finally, we consider an extension of the Delaunay property, namely protection, which is a measure of non-degeneracy of a Delaunay triangulation. In particular, one family of Coxeter triangulations achieves the protection $O(1/d^2)$. We conjecture that both bounds are optimal for triangulations in Euclidean space.

Acknowledgments

The research leading to the results in this thesis 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).

The writing of this thesis could not be done without the massive support of the people around me. First and foremost, I would like to thank my PhD advisor Jean-Daniel Boissonnat for accepting to mentor me, his support and guidance. His comments stimulated me to make choices that shaped this thesis. I am very indebted to my co-advisor Mathijs Wintracken whose help and advice made this thesis possible. Together, Jean-Daniel and Mathijs did a tremendous work in reviewing this thesis and helping me to make it more pedagogical and I am very thankful to them for that.

I thank Aurélien Alvarez, Dominique Attali and Vincent Pilaud for accepting to review my thesis, I am very grateful for their invaluable feedback. I also thank André Lieutier and Pierre Alliez for accepting to be in the jury during the defense.

I would like to thank my team at Inria Sophia Antipolis who changed its name from Geometrica to Datashape, but never changed its friendly spirit. Many thanks go to my officemates Alba Chiara de Vitis, Owen Rouillé, Harry Lang, Hannah Schreiber, Hannah Santa Cruz Baur and Shreya Arya for making the life in the office so enjoyable. You are all wonderful people and I wish you only the best. Thank you, Arijit Ghosh, Ramsay Dyer, Alfredo Hubard, Rémy Thomasse, Mael Rouxel-Labbé, François Godi, Siddharth Pritam, Kunal Dutta, Clément Maria, David Cohen-Steiner and Guilherme Dias da Fonseca, for all the insightful discussions that we shared. All in different ways, you were models of a scientist to me, and I learned a lot from you.

My life in Sophia Antipolis would not be the same without many friends I made there. I am grateful to people from Titane team, Liuyun Duan, Jean-Philippe Bauchet, Manish Mandad, Mohammad Rouhani, Emmanuel Maggiori, Sven Oesau, Jean-Dominique Favereau, Muxingzi Li, Hao Fang, Onur Tasar, Cédric Portaneri, Nicolas Girard and Flora Quilichini for their kindness, a good balance of seriousness and of sense of humour, and for sharing the floors of the Byron building during the weekends. You cannot even imagine how much it was enjoyable to share a coffee with you over a nice chat. One big special thanks goes to Mathieu Desbrun for being a wholesome dude. Your stays at Inria were always bringing me joy and your genuine care for PhD students is simply contageous. I wish to be able to make such cool presentations as yours someday. Many thanks to my fellow PhD students Simon Marillet, Nathalie Gayraud, Augustin Chevallier, Romain Tetley, Claude Stolze, Méliné Simsir, Denys Bulavka, Timothée O'Donnell, Karyna Gogunska, Vitalii Poliakov, Dimitra Politaki, Milica Tomašević and also Osama Arouk, for the interesting discussions we shared. I would also like to thank the permanent researchers in Titane and ABS teams, Frédéric Cazals, Dorian Mazauric, Pierre Alliez, Florent Lafarge and Yuliya Tarabalka, for their enthusiasm and knowledge they shared with me despite their lack of time. I am really grateful to the wonderful assistants in Geometrica/Datashape teams, Nelly Bessega, Florence Barbara and Sophie Honnorat for their enormous help and attention.

Thank you, Clément Jamin, Vincent Rouvereau and Marc Glisse, for accepting me in the Gudhi project. Thanks to you, I gained a vast knowledge on writing a proper C++ code, CMake and other tools, which helped me a lot during this thesis. I would also like to express my gratitude to my colleagues from Paris, Frédéric Chazal, Miroslav Kramár, Steve Oudot, Raphaël Tinarrage, Mathieu Carrière and Claire Brécheteau for making the visits to Paris something to look forward to.

A special thanks goes to Paweł Dłotko. Your invitation to the workshop in Będlewo was an impactful event in the destiny of this thesis. There I met Yasuaki Hiraoka who invited me to visit his team to Tohoku university in Sendai. Thanks to Hiraoka-sensei and his team I found strength and motivation to keep pushing when I was on the bringe of giving up, and I am really grateful to them. Thank you, Hiraoka-sensei, Hayakawa-san, Obayashi-san, Emerson-san, Wada-san, Mickael-san, Takeuchi-san, Kusano-san, Kanazawa-san, Miyanaga-san, Mikuni-san, Sakurai-san. Working in your team made that summer truly special to me.

Along the way, I met many wonderful people and made new friends. I would like to thank people from Gamble team in Nancy, Olivier Devilliers, Monique Teillaud, Charles Dumenil and especially Iordan Iordanov, with whom we had many enriching discussions on computational geometry during their visits to Sophia and JGA. During my trips to conferences and workshops, I met many interesting people I had pleasure talking to, such as Éric Colin de Verdière, Dejan Govc, José Luis Licón Saláiz, Sara Kališnik Verovšek and many others.

I would like to thank my dear friends Tristan Vaccon and Salomé Oudet for their unconditional moral support and understanding while I was writing this thesis. I am indebted to Nicolas Blanchard and Leila Gabasova for their hospitality during the last months of writing the manuscript. Thanks to you and the wholesome environment you provided I managed to keep my sanity until the manuscript was finally done. Many thanks also go to Édouard Thomas who helped me to improve the introduction of this thesis.

With this thesis, a whole chapter of my life in France is coming to a close. I cannot forget Philippe and Boris Kalitine who invited me to pursue my studies in France and to open this chapter. I am grateful to my teachers and classmates in Lycée Jean Moulin in Forbach, in Lycée du Parc in Lyon, in ENS Rennes (formerly ENS Cachan Antenne de Bretagne) and Parisian Masters in Research in Computer Science (MPRI). I met many great teachers such as Skander Zannad, Cédric Grange, Alain Chilles, Paulette Legroz, François Laurent, Luc Bougé, François Schwarzentrüber and others who not only taught me the bases of their subject but also helped me to integrate into French society.

Finally, nothing would be possible without the love and support from my family. First and foremost, I am grateful to my parents Svetlana and Mikalai Kachanovich who accepted all the choices I made during this eventful ride. My sister Palina and my cousins Uladzik, Zhenya, Iharok and Vitalka whom I all adore and I am very proud of. My grandparents, aunts and uncles whose wisdom and sense of humour is something to take example from. Thank you all for everything.

Contents

Introduction **1**

Chapter 1
Background **7**

1.1 Basic notions . . . 7
1.2 Root systems and Coxeter triangulations . . . 17
1.3 The Voronoi diagram of a Coxeter triangulation of type $\tilde{A}_d$ . . . 30

Chapter 2
Quality of Coxeter triangulations **41**

2.1 Quality definitions . . . 42
2.2 Main result . . . 47
2.3 Delaunay criterion for monohedral triangulations of $\mathbb{R}^d$ . . . 48
2.4 Geometrical analysis of each family of Coxeter triangulations . . . 51
2.5 Protection value of a triangulation of type $\tilde{A}_d$ . . . 70
2.6 Numerical values of quality measures of simplices in Coxeter triangulations . . . 73

Chapter 3
Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ **75**

3.1 Definition of Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ . . . 76
3.2 Eaves notation for d-dimensional simplices . . . 87
3.3 Permutahedral representation of simplices of arbitrary dimension . . . 92
3.4 Point location in the Freudenthal-Kuhn triangulation . . . 108
3.5 Generation of faces and cofaces in a Freudenthal-Kuhn triangulation . . . 110

Chapter 4
Manifold tracing algorithm **127**

4.1 The data structure to represent an ambient triangulation . . . 130
4.2 Manifold tracing algorithm . . . 134

4.3 Special case of the implicit manifold intersection oracle 144
4.4 Experimental results 152
4.5 Implementation details 154
4.6 Discussion 169

Bibliography **171**

Introduction

Problem and motivation. In this thesis, we investigate the problem of *manifold meshing*. It is formulated as follows. Assume that we are given a submanifold $\mathcal{M}$ (with or without a boundary) of arbitrary dimension m embedded in a high-dimensional Euclidean space $\mathbb{R}^d$ via a so-called *intersection oracle*. The intersection oracle allows us to determine if a given $(d-m)$-dimensional simplex intersects the manifold $\mathcal{M}$ or not. Our goal is to find a piecewise-linear approximation of the manifold $\mathcal{M}$.

The manifold meshing has many important applications, such as the computer graphics or finding numerical solutions to partial differential equations. Notably, in the medical imaging, computerized tomography (CT) scans and magnetic resonance imaging (MRI) scans provide the surface of any organ in the human body as a manifold embedded in $\mathbb{R}^3$. However, the application range of the implicit manifold reconstruction is not limited only to three dimensions. Even problems that seem to be two- or three-dimensional may require higher-dimensional approaches to find a solution. Take the example of the *motion planning* in robotics. If a robot has m degrees of freedom, then its configuration space is an m-dimensional manifold that is embedded in some high-dimensional Euclidean space $\mathbb{R}^d$. It is often important to understand the topology of this manifold in order not to lose potential solutions in motion planning.

The problem of manifold meshing has been extensively studied in the case of surfaces embedded in $\mathbb{R}^3$, especially in the computer graphics and the geometry processing literature. However, the applications such as motion planning require techniques that are generalized to higher dimensions. The existing methods for manifold meshing in $\mathbb{R}^2$ and $\mathbb{R}^3$, when generalized to the ambient space of arbitrary dimension d, gain exponential complexity in d (a phenomenon informally known as the *curse of dimensionality*). This makes these methods impractical for the applications in high ambient dimensions [Bel57].

For some methods, this exponential complexity comes from the explicit storage by these algorithms of a subdivision of the ambient space, the size of which grows exponentially with the ambient dimension. This is the problem of many meshing algorithms that are based on Delaunay triangulations and Voronoi diagrams (see [CDS12]). The famed marching cube algorithm [LC87] is another example of exponential dependency in ambient dimension. It needs to explore 2^{2^d} sign configurations on the vertices of a d-dimensional cube and to store them in a lookup table [BWC00]. Our primary goal in this thesis is to find an algorithm for the manifold meshing that has a polynomial dependence on the ambient dimension per computed mesh element and is efficient in practice.

Related work

Implicit manifolds. An important subclass of the manifold meshing problem is the *implicit manifold meshing*, on which there is a vast body of literature. The most popular and the most used is the so-called *marching cube algorithm* for the implicit surface reconstruction in

$\mathbb{R}^3$ introduced in [LC87]. See [NY06, Wen13] for extensive surveys on marching cubes and its variants used for implicit surface reconstruction in $\mathbb{R}^3$.

We will now outline a quick description of the marching cube algorithm, following the original paper [LC87]. The marching cube algorithm exploits a decomposition of a box-shaped domain of interest into a cubical grid. The surface is given as the zero-set $F^{-1}(0)$ of a function $F : \mathbb{R}^3 \to \mathbb{R}$. For each cube in the cubical grid, the marching cube algorithm evaluates the value of the function F at the vertices of the cube. Depending on the values of F at the vertices of the cube, the algorithm then associates a polygon that locally approximates $F^{-1}(0)$ in the cube. Once all such polygons are constructed for all cubes in the cubical grid, the marching cube algorithm terminates.

There is one polygon per so-called sign configuration of a cube. The sign configuration consists of the signs of the values of the function F at the eight vertices of the cube. All sign configurations and the corresponding polygons are stored in a lookup table that is precomputed before the algorithm starts.

The marching cube, while being simple, does not provide, in its simplest formulation, a topologically-consistent surface. This is due to a problem, known as the *ambiguous configurations*, first pointed out by Dürst [Dür88] (illustrated in Figure 1). There has been numerous techniques to handle the ambiguous configuration problem, as for example the ones proposed in [NH91, MSS94, ZSK94]. See [Wen13] for a more complete overview on the topic.

Even in $\mathbb{R}^3$, the marching cube method and its variants do not provide guarantees of being homeomorphic to the reconstructed surface. The only exception, to our knowledge, of a provably correct implicit surface reconstruction method in $\mathbb{R}^3$ is described in the paper by Plantinga and Vegter [PV04].

The marching cube has been extended to the implicit manifold reconstruction in a high-dimensional ambient space in the work by Bhaniramka *et al.* [BWC00]. Their method is able to reconstruct a codimension-one hypersurface using a lookup table, which takes into account the ambiguous configuration problem mentioned before. A big disadvantage of the use of the marching cube algorithm in high dimensions is the large size of the possible configurations stored in the lookup table. Just generating the table is difficult, as all possible sign configurations of the values of F on vertices need to be checked [BWC00]. There are 2^d vertices, therefore the number of sign configurations to be checked is 2^{2^d}.

One way to overcome the combinatorial explosion in high dimensions is to use simplices

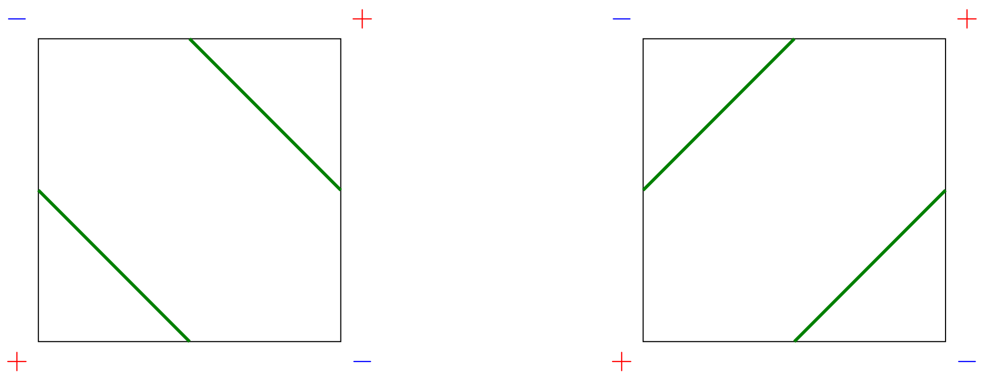


Figure 1: An ambiguous configuration in a square and two possible sets of polygonal pieces that fit it. Here, the signs at the vertices of the square indicate the signs of the values of the implicit function F at these vertices.

as ambient grid elements [Min03]. In numerous methods, these simplices come from simplicial decompositions of the ambient cubes, such as the barycentric decomposition, used for example in [WB96], or the Freudenthal-Kuhn's decomposition, used for example in [Min03]. While the number of possible sign configurations in a simplex is polynomial, the number of simplices in one such decomposition of a cube is at least factorial in the ambient dimension [Min03].

It is therefore advantageous to look for methods that do not visit all the simplices in the triangulation of the domain, but rather only the *relevant* ones. This idea lies at the core of the manifold tracing algorithm. Bloomenthal also used this idea for the implicit surface reconstruction in $\mathbb{R}^3$ [Blo88]. In this work, it is assumed that a seed point on the implicit surface is given, and then the algorithm propagates the reconstruction from the simplex that contains the seed point to all adjacent simplices in a front-like manner. This method was later known as the *seed propagation* [Wen13], or the *simplicial continuation method* [Hen07]. The manifold tracing algorithm that we present in this thesis belongs to the class of simplicial continuation methods.

To our knowledge, the only simplicial continuation algorithm that can be applied to arbitrary ambient dimension and arbitrary dimension of the manifold is the so-called *pattern algorithm* of Allgower and Schmidt [AS85, AG90]. However, by the choice of the representation, their algorithm is inefficient and complex (more on this below).

There is some body of work on implicit submanifold reconstruction using Coxeter triangulations. The tetrahedron that generates a Coxeter triangulation of type $\tilde{A}_3$ was used as a marching element in a tetrahedral generalization of marching cube in [CP98, TPG99, LS07]; however, all these methods are limited to the three-dimensional ambient space.

While most of the related work is limited to the implicit manifold meshing problem, there exist algorithms in the literature that work in a more generic framework. In particular, the meshing algorithm by Boissonnat and Oudot in $\mathbb{R}^3$ [BO05] only needs to know the surface through an oracle that can compute the intersection of any given segment with the surface (we call this oracle an *intersection oracle*). Our manifold tracing algorithm works in a generalization of this framework to any dimension of the Euclidean space and of the submanifold.

Coxeter triangulations and Freudenthal-Kuhn triangulations. One of the key tools that we use to efficiently solve the implicit manifold meshing problem is a family of triangulations of the Euclidean space $\mathbb{R}^d$, which are called *Coxeter triangulations*. Coxeter triangulations are hyperplane arrangements of $\mathbb{R}^d$ whose cells are d-dimensional simplices with a special property that two adjacent d-dimensional simplices in a Coxeter triangulation are orthogonal reflections of one another. Coxeter triangulations exist for any ambient dimension. We illustrate the three possible Coxeter triangulations of $\mathbb{R}^2$ in Figure 2. There are two attractive properties that the Coxeter triangulations share:

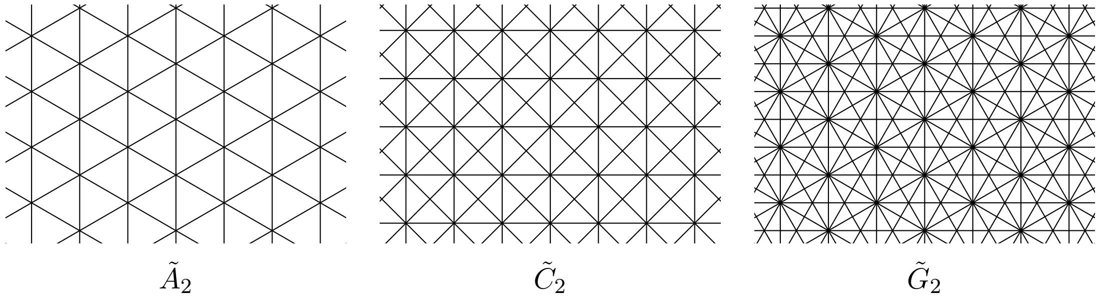


Figure 2: The two-dimensional Coxeter triangulations.

1. Coxeter triangulations have *good simplex quality* (defined further below), which is often desirable in manifold reconstruction [CDR05, CDS12, BDG14], but also in such fields as the finite element methods [BA76, Jam76, Kří92, Syn57];

2. Coxeter triangulations have *rich structure.* A compact and practical description of the d-dimensional simplices in a subfamily of Coxeter triangulations was provided by Freudenthal [Fre42], though formulated for a closely related but different Freudenthal-Kuhn triangulation of $\mathbb{R}^d$. It was shown much later in [DWLT90] that these triangulations are identical up to a linear transformation, hence they have the same combinatorial structure. Therefore, the compact representation of the d-dimensional simplices in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ can be used for this subfamily of Coxeter triangulations too.

The Freudenthal-Kuhn triangulation of the Euclidean space was invented independently by Freudenthal [Fre42] (in German) and Kuhn [Kuh60]. This triangulation is known under many names: *K1 triangulations* [Tod76], *Freudenthal's triangulations* [Eav84, Dan95, EK12], *Kuhn's triangulations* [Moo92].

Both Freudenthal and Kuhn were not aware of the closely related work by Coxeter, who introduced more general Coxeter triangulations in his study of the reflection groups [Cox34]. It was only much later in the work by Dobkin *et al.* [DWLT90] that the link between Coxeter triangulations of type $\tilde{A}_d$ and the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ was finally established.

The original motivation in Freudenthal's pioneer paper [Fre42] was the construction of an infinite series of triangulations of the unit cube, such that the next triangulation in the series is a subdivision of the previous one and the simplices do not become arbitrarily flat as the series progresses. In each triangulation of the unit cube, Freudenthal [Fre42] defined a representation (which we call the *Freudenthal representation* in the thesis) for every d-dimensional simplex that consists of two components:

- a vector $y \in \mathbb{Z}^d$ that encodes the position of the minimal vertex $v_0 = y$ of the simplex in the lexicographical order,

- a permutation $\pi : \{1, \ldots, d\} \to \{1, \ldots, d\}$ that defines all other vertices $v_1, \ldots, v_d$ of the simplex in the following way:

$$v_i = v_{i-1} + e_{\pi(i)} \quad \text{for } i \in \{1, \ldots, d\}.$$

This paper by Freudenthal [Fre42] seems to have been forgotten for almost thirty years after its publication. Meanwhile, Kuhn reinvented the same Freudenthal-Kuhn triangulation of the unit cube in his alternative proof of Brouwer fixed point theorem using Sperner's lemma [Kuh60].

The first practical application of the Freudenthal-Kuhn triangulation appeared in a note by Kuhn [Kuh68]. There, the author adapted the existing algorithm of Scarf [Sca67] to find a fixed point given by Brouwer theorem by using the Freudenthal-Kuhn triangulation. In the same work, Kuhn introduced two algorithms that use the Freudenthal representation of d-dimensional simplices in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$: point location and adjacent d-dimensional simplex computation.

The Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ appeared in the book by Todd [Tod76, Chapter III]. It became a staple tool in the subsequent work on approximating the fixed point given by Brouwer theorem (even further popularized by the survey [AG80]). It was in the same book that Todd pointed out for the first time that Freudenthal's triangulation and Kuhn's triangulation are in fact the same.

The first application of the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ to the piecewise-linear approximation of implicit submanifolds of $\mathbb{R}^d$ of arbitrary codimension appeared in Allgower and Schmidt [AS85]. The authors used the Freudenthal representation for the d-dimensional simplices and a collection of Cartesian coordinates of vertices to represent all other simplices. This representation of simplices is inefficient and unnatural when applied to the algorithm. The reader can find the PASCAL and FORTRAN codes in [Gnu88] and [AG90, Program 5] respectively.

A related algorithm is the contour tracing algorithm by Dobkin *et al.* [DWLT90]. There, the authors used Coxeter triangulations as the ambient triangulations to mesh one-dimensional curves. Compared to the existing work by Allgower and Schmidt [AS85], the algorithm by Dobkin *et al.* is simpler and more accurate when reconstructing curves embedded in the Euclidean space. While the authors briefly discuss a possible extension to manifolds of general dimension, they do not provide any details in how to do so.

Contributions

After the introductory Chapter 1, the three following chapters correspond to the three main contributions of this thesis.

- In Chapter 2, we provide a comprehensive study of the geometric properties of the simplices for all families of Coxeter triangulations in Euclidean space. We show that not all Coxeter triangulations are Delaunay triangulations. We are also interested in the quality of simplices in the Coxeter triangulations. Here, by simplex quality we mean the following ratio-based qualities of a simplex that were previously used in the computational geometry community:

  - the ratio of the minimal height to the diameter, called *thickness* [Mun66, Vav96],
  - the ratio of the inscribed ball radius to the circumscribed ball radius, called *radius ratio* [Whi57],
  - the ratio of the volume to the dth power of the diameter, called *fatness* [CFF85].

  For any simplex quality above, by taking the infimum over all d-dimensional simplices in a triangulation, we can define the corresponding quality of the whole triangulation. For Delaunay triangulations, we can add to the list of triangulation qualities the so-called *protection*, introduced in [BDG13]. We show that one family of Coxeter triangulations (the $\tilde{A}$ family) consists of Delaunay triangulations with the currently best known protection value over all triangulations in each respective dimension.

  For each family of Coxeter triangulations, we provide explicit measurements of simplices (all of which are similar in a given triangulation). This allows the reader to compute other quality measures than the ones listed. We show an example of such a custom quality measure, which is the ratio of the minimal height and the radius of the circumscribed ball that we call *aspect ratio*.

- In Chapter 3, we present a new representation of simplices in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ called the *permutahedral representation*. The novelty of the permutahedral representation with respect to the existing Freudenthal representation is that the permutahedral representation can represent the simplices of arbitrary dimension in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$. The storage complexity of the permutahedral representation of a simplex in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ is $O(d)$ — the same as for the

Freudenthal representation. By encoding simplices in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ using the permutahedral representation, we can access in a time- and space-efficient way:

- the simplex that contains a given point,
- the faces of a given simplex,
- the cofaces of a given simplex.

- In Chapter 4, we introduce a data structure that stores an arbitrary linear transformation of the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ using only $O(d^2)$ storage. This data structure is based on the permutahedral representation introduced in Chapter 3. We apply this data structure for the *manifold tracing algorithm* that computes a piecewise-linear approximation of a given compact smooth manifold. We only assume that the manifold can be accessed *via* an intersection oracle that answers whether a given simplex in the ambient Euclidean space intersects the manifold or not. This makes the manifold tracing algorithm applicable to various representations of the input manifold. In particular, we discuss two such representations:

  - the implicit manifolds given as the zero-set of a function,
  - the manifolds given by point clouds.

  We show the experimental results of the manifold tracing algorithm on the implicit manifolds and the manifolds given by point clouds. This implementation is at the moment under review to be included in Gudhi library [GUD].

Chapter 1

Background

In this chapter, we will give an overview of the basic definitions, which will be useful in the thesis. The chapter is split into three sections:

- In Section 1.1, we recall some basic definitions and results related to linear algebra, group theory, graph theory, order theory, geometry, topology and algorithmics that will be useful later in the thesis.
- In Section 1.2, we give the formal definition of the Coxeter triangulations of $\mathbb{R}^d$. The definition of Coxeter triangulations relies on the concepts of root systems and root lattices that we present in Section 1.2 as well. The presentation of this section is largely based on the lecture notes by Top [Top] and the book by Humphreys [Hum92].
- In Section 1.3, we study the face structure of a polytope, known as *permutahedron*, which appears as the full-dimensional cells in the Voronoi diagram of a Coxeter triangulation of type $\tilde{A}$. The results mentioned in this section are used later in Chapter 3, where we introduce the permutahedral coordinates of simplices in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$.

Most of the results mentioned in Chapter 1 are known. For instance, the definitions in Section 1.1 and 1.2 are standard and any reader familiar with them should feel free to skip these sections and consult them only when needed. In Section 1.3.2, we do a standard computation of the number of the k-dimensional subfaces of a given l-dimensional face of a d-dimensional permutahedron, for some given k, l and d. This result is part of folklore and is included for completeness.

For a more complete introduction we refer the reader to the pioneering paper on reflection groups by Coxeter [Cox34], a book on Coxeter systems by Humphreys [Hum92] and the classical book on Lie groups and algebras by Bourbaki [Bou02].

1.1 Basic notions

In this section, we recall standard definitions and fix the notations that are used in the thesis.

1.1.1 Notations in linear algebra

The Euclidean space (which is endowed with the standard scalar product) is denoted by $\mathbb{R}^d$, where d is a non-negative integer that stands for dimension. The vectors in the canonical basis of $\mathbb{R}^d$ is denoted by $e_1, \ldots, e_d$.

Vector spaces or *linear spaces* in this thesis always refer to the linear subspaces of $\mathbb{R}^d$. Therefore, all vector spaces in the thesis are finite-dimensional, and any set of vectors is of finite rank.

The scalar product of two vectors $x, y \in \mathbb{R}^d$ will be denoted by $\langle x, y \rangle$ and the norm of a vector $x \in \mathbb{R}^d$ is denoted by $\|x\|$. In the following, we confound vectors and points in Euclidean space. Sometimes, though, to make a separation between points and vectors, especially in geometrical proofs, an arrow is used over the vectors, like $\vec{x}$. This includes vectors between two points: for example, $\overrightarrow{xy}$ denotes the vector $y - x$ for two points x and y. The Euclidean distance between two points x and y will be denoted by $d(x, y)$ or alternatively by $\|x - y\|$.

Finally, for any subset $S \subseteq \mathbb{R}^d$, we will denote by $\mathrm{aff}(S)$ the affine hull of S.

1.1.2 Group theory

We will now recall some standard definitions from group theory that will be used in the thesis. For most statements, we follow the formulation of Rotman [Rot12]. The same holds for the notations, with the exception of dihedral groups.

Definition 1.1.1 (Group)**.** *A group $(G, \star)$ is a non-empty set G equipped with an associative operation $\star : G \times G \to G$, such that:*

1. *There exists an element $e \in G$, which satisfies for all $g \in G$:*

$$g \star e = e \star g = g.$$

The element e is called the identity*.*

2. *For all $g \in G$, there exists an element $h \in G$, such that:*

$$g \star h = h \star g = e.$$

The element h is called the inverse *of g and is often denoted by g^{-1}.*

The $\star$ notation for the operation will be dropped for the rest of the section.

We will see two types of products of groups in the following. We will now briefly recall the definitions of these products.

Definition 1.1.2 (Direct product)**.** *If G and H are groups, then their* direct product*, denoted by $G \times H$, is the group with elements all ordered pairs (g, h), where $g \in G$ and $h \in H$, and with operation $(g, h)(g', h') = (gg', hh')$.*

Definition 1.1.3 (Normal subgroup)**.** *A nonempty subset H of a group G is a* subgroup *of G if $s \in H$ implies $s^{-1} \in H$, and $s, t \in H$ implies $st \in H$.*

A subgroup H of G is a normal subgroup*, denoted by $H \lhd G$, if $gHg^{-1} \subseteq H$ for every $g \in G$.*

Definition 1.1.4 (Semidirect product)**.** *Let H and Q be two subgroups of G. Denote by e the identity of the group G. The subgroup Q is said to be a* complement *of H in G if $H \cap Q = \{e\}$ and every $g \in G$ can be represented as kq for some $k \in H$ and $q \in Q$.*

A group G is a semidirect product *of H by Q, denoted by $G = H \rtimes Q$ if $H \lhd G$ and H has a complement Q_1 isomorphic to Q.*

We will now introduce actions of groups on sets.

Definition 1.1.5 (Action of a group)**.** *One says that a group G* acts *on a set X, if there exists a function $\alpha : G \times X \to X$, called* (left) action[1]*, denoted by $\alpha : (g, x) \mapsto gx$, such that:*

(i). $ex = x$ for all $x \in X$, and

(ii). $g(hx) = (gh)x$ for all $g, h \in G$ and $x \in X$.

Definition 1.1.6 (Orbit)**.** *Suppose that G acts on X. Let $x \in X$. Then the G-orbit of x is the set:*

$$\mathcal{O}(x) = \{gx : g \in G\} \subseteq X.$$

Definition 1.1.7 (Transitive action)**.** *The action of a group G on a set X is called* transitive*, if it has only one orbit; that is, for every $x, y \in X$, there exists $g \in G$ with $y = gx$. The action is called* simply transitive *if for any x and y, such an element g exists and is unique.*

Now, we will recall a few important families of groups.

Definition 1.1.8 (Symmetric group)**.** *We call a* permutation *on a set $\{1, \dots, n\}$ a bijective function $\{1, \dots, n\} \to \{1, \dots, n\}$. The group of all permutations on a set of $\{1, \dots, n\}$ and whose group operation is the composition, is called the* symmetric group *and is denoted as $\mathfrak{S}_n$.*

Definition 1.1.9 (Dihedral group)**.** *The* dihedral group $\mathfrak{D}_n$ *for $n \geq 2$, is a group of order $2n$ (meaning it contains $2n$ elements), which is generated by two elements s and t, such that:*

$$s^2 = e, \quad t^n = e \quad \textit{and} \quad sts = t^{-1},$$

where e denotes the identity.

1.1.3 Graph theory

We will now recall basic definitions in graph theory. For more information on graph theory, we refer the reader to standard textbooks, such as Harary [Har69].

Definition 1.1.10 (Graph)**.** *A* directed graph *(respectively* undirected graph*) is a pair of sets $G = (N, E)$, where E consists of the ordered (respectively* unordered*) pairs of elements in N.*

The elements in N are called nodes*, and the elements in E are called* edges*. The two nodes $n_1 \in N$ and $n_2 \in N$ in an edge $e = (n_1, n_2) \in E$ are called the* endpoints *of e.*

Remark 1.1.11. *Note that E in Definition 1.1.10 is a* set*. Therefore, no repetition of edges can occur in a graph as defined above. Hence, we only consider here* simple *graphs (as opposed to multigraphs, for which E is a multiset).*

Definition 1.1.12 (Degree in a directed graph)**.** *Let $G = (N, E)$ be a directed graph. For a node $n \in N$, the cardinality of the set $\{n' \in N \mid (n, n') \in E\}$ (respectively cardinality of the set $\{n' \in N \mid (n', n) \in E\}$) is called the* out-degree *of the node n (respectively the* in-degree *of the node n).*

Definition 1.1.13 (Degree in an undirected graph)**.** *Let $G = (N, E)$ be an undirected graph. For a node $n \in N$, the cardinality of the set $\{n' \in N \mid \{n, n'\} \in E\}$ is called the* degree *of the node n.*

[1] One can also define the *right* action in the same way.

Definition 1.1.14 (Path)**.** *A* path *in the graph $G = (N, E)$ is a sequence of m pairwise distinct edges $\Pi = e_0, e_1, ..., e_m$ for some integer $m \geq 0$, where for all $(i, i+1) \in \{0, \dots, m-1\}^2$, the two edges e_i and e_{i+1} share an endpoint.*

Remark 1.1.15. *Note that all graphs that we deal with in the thesis are* finite*. Since a finite graph has a finite number of edges, this implies that any path is finite as well.*

Definition 1.1.16 (Nodes on a path)**.** *Let $\Pi = e_0, e_1, ..., e_m$ be a path for some integer $m \geq 0$. Let $(n_i)_{i \in \{0,\dots,m\}}$ be the sequence of nodes such that for all $i \in \{0, \dots, m-1\}$, the edge e_i has endpoints n_i and n_{i+1}. For any $i \in \{0, \dots, m\}$, we will say that n_i* lies *on the path Π, or that n_i is a* node on the path Π*. We will call the elements n_0 and n_m the* endpoints *of the path Π. We will also say in this case that Π is a* path from n_0 to n_m.

Definition 1.1.17 (Connected graphs)**.** *A graph $G = (N, E)$, such that for any two nodes n and n' in N, there exists a path from n to n' or there exists a path from n' to n, is called* connected*. If from any node in G there exists a path to any other node, then G is called* strongly connected*.*

Remark 1.1.18. *Note that any undirected graph that is connected is also strongly connected.*

Definition 1.1.19 (Cycle)**.** *A path is called a* cycle *if its endpoints are the same node. A* loop *is a particular case of a cycle that consists of one edge.*

The special type of graphs without cycles will be of special importance for the data structure discussed in Chapter 4.1.

Definition 1.1.20 (Directed acyclic graph)**.** *A directed graph $G = (N, E)$ that does not contain any cycles is called a* directed acyclic graph *(abbreviated as* DAG*).*

Definition 1.1.21 (Tree)**.** *Let $G = (N, E)$ be a directed acyclic graph. If there exists an element $r \in N$, such that from any node $n \in N$, there exists a unique path from n to r, the graph G is called a* (directed) rooted tree*. The node r in such a graph is called the* root *of the tree G.*

For a given edge $(n, n') \in E$ in a rooted tree $G = (N, E)$, the node n is called a child node *of n'. Conversely, the node n' is called a* parent node *of n. By extension, we will say that n' is an* ancestor *of n and that n is a* descendent *of n' if there exists a path in G from n to n'.*

A node in a rooted tree that has no child node (or equivalently has in-degree 0) is called a leaf*. The maximal length of a path from a leaf to the root in a rooted tree is called the* depth *of the tree.*

Remark 1.1.22. *Note that the edges in Definition 1.1.21 point from children to parents. A directed tree can equivalently be defined with edges pointing from parents to children.*

Trees have an intrinsic recursive structure, which can be expressed in terms of subtrees.

Definition 1.1.23 (Subtree)**.** *To any node $n \in N$ in a directed rooted tree $G = (N, E)$ we can put in correspondence a graph, such that its set of nodes N' consists of n itself and all descendants of n in G, and the set of edges E' consists of all edges in G with endpoints in N'. Such a graph will be called a* subtree of G rooted at n*, or alternatively a* branch of G rooted at n.

An analogous definition of trees exists for the undirected graphs.

Definition 1.1.24 (Undirected tree)**.** *An undirected graph $G = (N, E)$ that does not contain any cycles is called an* undirected tree *(or simply a* tree*). A node in an undirected tree that has degree 1 is called a* leaf.

When it is clear that a graph is directed, the term *tree* in the following will denote a directed rooted tree. If the graph is clearly undirected, the term *tree* will denote an undirected tree as in the definition.

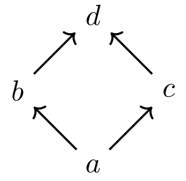


Figure 1.1: Hasse diagram of a partially-ordered set $(\{a, b, c, d\}, \leqslant)$ with $\leqslant$ defined for the following pairs: $a \leqslant a$, $a \leqslant b$, $a \leqslant c$, $a \leqslant d$, $b \leqslant b$, $b \leqslant d$, $c \leqslant c$, $c \leqslant d$, $d \leqslant d$. All loops around nodes are omitted from the figure.

1.1.4 Ordered sets

The goal of this section is to introduce the diverse notions of orders that are used in the thesis. For more information, we refer the reader to textbooks on the topic, such as Simovici and Djeraba [SD08]. We will start with the definition of a partially-ordered set.

Definition 1.1.25. *A* partial order $\leqslant$ *over a set S is a binary relation, which satisfies all of the following:*

(Reflexivity). *For all $a \in S$, we have $a \leqslant a$.*

(Antisymmetry). *For all $a, b \in S$, if $a \leqslant b$ and $b \leqslant a$, then $a = b$.*

(Transitivity). *For all $a, b, c \in S$, if $a \leqslant b$ and $b \leqslant c$, then $a \leqslant c$.*

A set $(S, \leqslant)$ endowed with a partial order $\leqslant$ is called a partially-ordered set *(sometimes contracted to* poset*).*

If for two elements $a \in S$ and $b \in S$, we have $a \leqslant b$, then we will say that b is *greater* than a in the order $\leqslant$. In this case, we also say that a is *smaller* than b in the order $\leqslant$. We will omit "in the order $\leqslant$" whenever it is clear which partial order is implied.

A finite partially-ordered set can be represented with the help of a *Hasse diagram.*

Definition 1.1.26. *The* Hasse diagram *of a partially-ordered set $(S, \leqslant)$ is a directed acyclic graph, where the nodes are elements of S, and there is an edge*[2] *going from a node $a \in S$ to a different node $b \in S$ if and only if both of the two statements below hold:*

- *$a \leqslant b$,*
- *for all $c \in S$ such that $a \leqslant c \leqslant b$, we have either $c = a$ or $c = b$.*

An example of Hasse diagram of a partially-ordered set is illustrated in Figure 1.1.

A defining property of Hasse diagrams is that $a \leqslant b$ is equivalent to the existence of a path from a to b in the Hasse diagram.

It will be useful in the following to have the notions of upper and lower bounds in the context of the partially-ordered sets.

Definition 1.1.27. *Let $(S, \leqslant)$ be a partially-ordered set and let $R \subseteq S$ be a subset of S.*

An element $a \in S$ is an upper bound *of R (respectively* lower bound *of R) if for all $r \in R$, we have $r \leqslant a$ (respectively $a \leqslant r$).*

[2] An alternative definition with an edge going from b to a is also possible.

If it exists, the smallest upper bound of R (respectively the greatest lower bound of R) is called the supremum *of R (respectively the* infimum *of R) and is denoted as* $\sup R$ *(respectively* $\inf R$*).*

If the supremum of R (respectively the infimum of R) belongs to R, then it is called the maximum *of R (respectively the* minimum *of R), and is denoted as* $\max R$ *(respectively* $\min R$*).*

In Section 1.3, we talk about the *isomorphisms* between two partially-ordered sets.

Definition 1.1.28 (Isomorphism between two posets)**.** *Let $(S, \leqslant)$ and $(T, \leqslant')$ be two partially-ordered sets. We say that a map $\varphi : S \to T$ is an* order embedding*, if for all $a, b \in S$, we have:*

$$a \leqslant b \text{ if and only if } \varphi(a) \leqslant' \varphi(b).$$

An order embedding $\varphi : S \to T$ that is surjective is called isomorphism *between the two partially-ordered sets $(S, \leqslant)$ and $(T, \leqslant')$. If there exists an isomorphism between $(S, \leqslant)$ and $(T, \leqslant')$, then we say that the two partially-ordered sets are* isomorphic*.*

Lastly, we will define a stronger notion of *linear order.*

Definition 1.1.29 (Linear order)**.** *Let S be a set. A relation $\leqslant$ is a* linear order *if it is a partial order over S and for any two elements a and b in S, at least one of $a \leqslant b$ and $b \leqslant a$ is true.*

When the set S is finite, it is possible to *extend* a given partial order $\leqslant$ to a linear order $\leqslant^*$.

Lemma 1.1.30 (Linear extension of a partial order)**.** *Let S be a finite set and $\leqslant$ be a partial order over S. There exists a linear order $\leqslant^*$ over S such that for all $a, b \in S$, if $a \leqslant b$, then $a \leqslant^* b$.*

The proof of Lemma 1.1.30 is classical and is done constructively by a so-called topological sort (see for example the depth-first search procedure PREORDER in [Tar76] applied to the Hasse diagram).

Finally, for any linear order (and any partial order in general) we can associate a strict order.

Definition 1.1.31 (Strict order)**.** *For each partial order $\leq$ over a set S there exists an associated relation $<$, such that for any $a, b \in S$, we have $a < b$ if and only if $a \leq b$ and $a \neq b$. Such relation is called a* strict order*.*

1.1.5 General topology and geometry

In this section, we will present a few notions in general topology and geometry that will be essential for the rest of the thesis. For a more complete introduction on topology, we refer the reader to any standard book on topology, such as [Mun66].

There are two uses of the term *closed* in topological literature. The most standard one is *closed set*, which refers to the complementary of an open set (as used for example in Munkres [Mun66]). The other one is *closed manifold*, which refers to *compact manifold without boundary* (see for example Spivak [Spi99, p.19]). In this thesis, we restrict the term *closed* only to closed sets, and the closed manifolds are called compact manifolds without boundary to avoid any confusion.

We will now present the standard definition of the reach of a manifold (illustrated in Figure 1.2), introduced by Federer [Fed59].

Definition 1.1.32 (Reach [BCY18, Section 7.1.2])**.** *Let $\mathcal{M}$ be a submanifold of $\mathbb{R}^d$. The* medial axis *of $\mathcal{M}$ is defined as the closure of the set of points $x \in \mathbb{R}^d$ that have more than one closest point on $\mathcal{M}$.*

The reach *of $\mathcal{M}$, denoted by* $\mathrm{rch}(\mathcal{M})$*, is an infimum of the distance from a point on $\mathcal{M}$ to the medial axis.*

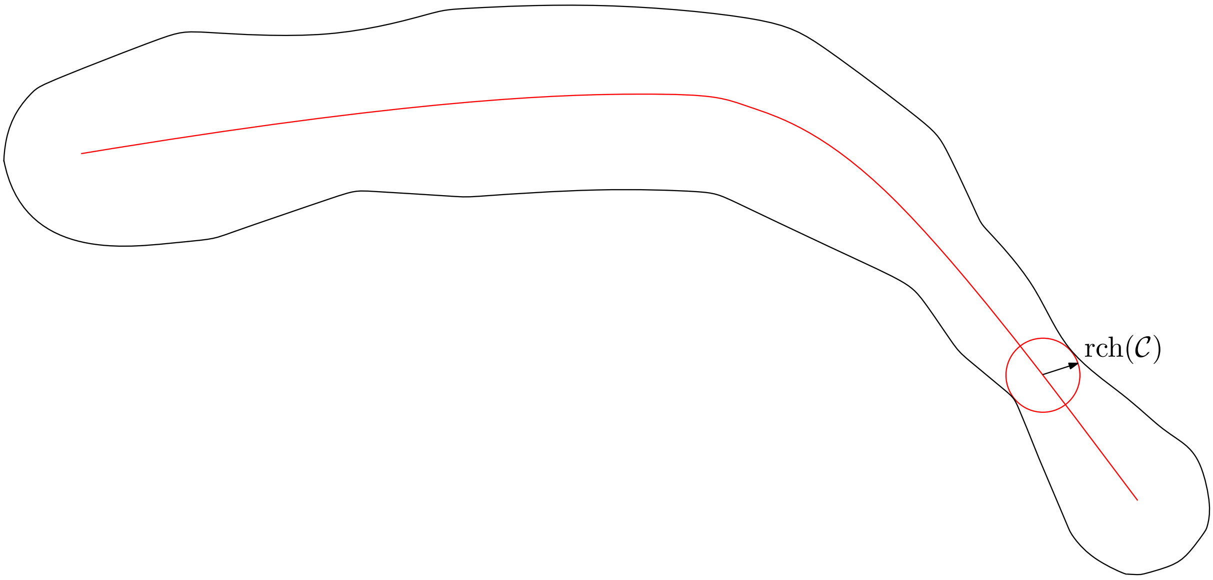


Figure 1.2: Illustration of the reach of a curve in $\mathbb{R}^2$. The medial axis of the curve is shown in red.

There are multiple notions of a "topological equivalence" that are mentioned in the thesis. First of them is the notion of homeomorphism.

Definition 1.1.33 (Homeomorphism)**.** *Two topological spaces X and Y are called* homeomorphic *if there exists a bijective continuous map $X \to Y$, such that its inverse is continuous as well. Such a map is called a* homeomorphism *between X and Y.*

A weaker notion of topological equivalence is that of homotopy equivalence.

Definition 1.1.34 (Homotopy equivalence)**.** *Let X and Y be two topological spaces. Two maps $f, g : X \to Y$ are called* homotopic *if there exists a continuous map $F : X \times [0,1] \to Y$, such that for all $x \in X$, we have $F(x,0) = f(x)$ and $F(x,1) = g(x)$. Such a map is called a* homotopy*.*

Two topological spaces X and Y are called homotopy equivalent *if there exist continuous maps $f : X \to Y$ and $g : Y \to X$, such that $f \circ g$ is homotopic to the identity map id_Y on Y and $g \circ f$ is homotopic to the identity map id_X on X. Both maps f and g are called* homotopy equivalences*.*

We will use the following notation for the volume of the unit sphere.

Definition 1.1.35 (Volume of the unit sphere)**.** *We denote by V_d the volume of the d-dimensional unit sphere.*

The volume of the unit sphere can be expressed using the Gamma function $\Gamma(z) = \int_0^\infty t^{z-1}e^{-t}dt$.

Proposition 1.1.36 (Volume of the unit sphere [DLMF, Equation 5.19.4])**.** *The volume V_d of the d-dimensional unit sphere is equal to:*

$$V_d = \frac{\pi^{(d-1)/2}}{\Gamma\left(\frac{d-1}{2}+1\right)}.$$

Using Stirling formula, we can also derive the following asymptotic approximation:

$$V_d = O\left(\left(\frac{2\pi e}{d}\right)^{d/2}\right).$$

There are two definitions of simplices in computational geometry: geometrical and abstract. The definition that is used in this thesis is the geometrical one.

An *m-dimensional simplex* in $\mathbb{R}^d$ is defined as a convex hull of some $m+1$ points in general position, with $m \leqslant d$ called the *dimension* of the simplex. These points are called *vertices* of the simplex, and are regarded as 0-dimensional simplices themselves. We will reserve the word *vertices* only for simplices (as well as polytopes and polyhedra in general, see Definition 1.1.39 below) and use the word *node* for all other uses (for example, in graphs). If the dimension of a simplex in $\mathbb{R}^d$ is d, we will call such simplex *full-dimensional.* The 1-dimensional simplices are called *edges* (or sometimes *segments*), the 2-dimensional simplices are called *triangles* and the 3-dimensional simplices are called *tetrahedra.*

Definition 1.1.37 (Halfspace)**.** *A* halfspace *H is a subset of $\mathbb{R}^d$ of the form $\{x \in \mathbb{R}^d \mid \langle n, x\rangle \leq b\}$ for some $n \in \mathbb{R}^d$ and $b \in \mathbb{R}$.*

Remark 1.1.38. *Any hyperplane $\{x \in \mathbb{R}^d \mid \langle n, x\rangle = b\}$ for some $n \in \mathbb{R}^d$ and $b \in \mathbb{R}$ defines two halfspaces: $\{x \in \mathbb{R}^d \mid \langle n, x\rangle \leq b\}$ and $\{x \in \mathbb{R}^d \mid \langle n, x\rangle \geq b\}$.*

Definition 1.1.39 (Convex polyhedron)**.** *A* polyhedron *in $\mathbb{R}^d$ is the non-empty intersection of a finite set of halfspaces. The* dimension *of a polyhedron is the dimension of its affine hull.*

Any convex hull of a set of points in $\mathbb{R}^d$ is in fact a bounded polyhedron, which we will call a *convex polytope*, or simply a *polytope.* In particular, any simplex is a polytope.

Definition 1.1.40 (Face)**.** *Let σ be a convex polyhedron in $\mathbb{R}^d$. Let H be a hyperplane in $\mathbb{R}^d$, such that the intersection of σ and H is non-empty and σ lies entirely in one of the halfspaces defined by H (see Remark 1.1.38). The intersection of σ and H is itself a polyhedron and is called a* face *of σ.*

In this thesis, we will deal with cell complexes and simplicial complexes.

Definition 1.1.41 (Cell complex)**.** *A* (geometrical) cell complex *is a collection $\mathcal{K}$ of polyhedra such that:*

- *any face of a polyhedron in $\mathcal{K}$ belongs to $\mathcal{K}$, and*
- *the intersection of any two polyhedra H_1 and H_2 in $\mathcal{K}$ is a face of both H_1 and H_2.*

Polyhedra that belong to a cell complex are called cells *of the cell complex.*

A (geometrical) simplicial complex *is a cell complex where all polyhedra are simplices.*

Remark 1.1.42. *Sometimes the word* polyhedron *refers to the cell complex itself. In this thesis, we will reserve the word polyhedron only to the geometrical object defined in Definition 1.1.39.*

In Section 1.3, we will use the following result.

Lemma 1.1.43 ([Zie12, Section 2.2])**.** *The inclusion relation on the cells in cell complexes is a partial order.*

Lemma 1.1.43 implies that any cell complex equipped with the inclusion relation can be regarded as a partially-order set.

Definition 1.1.44 (Face poset)**.** *The partially-ordered set of cells in a cell complex endowed with the inclusion relation is called* face poset*.*

As a consequence of Lemma 1.1.43, Hasse diagrams (discussed in Section 1.1.4) are one of the common ways to represent a cell complex [BCY18, p.18].

We will adopt the following vocabulary. An $(m-1)$-dimensional face of an m-dimensional polyhedron will be called a *facet.* A cell τ in a cell complex is a face (respectively a facet) of σ, we will say that σ is a *coface* of τ (respectively a *cofacet* of τ).

Definition 1.1.45 (Star)**.** *Let $\mathcal{K}$ be a cell complex. The* star *of a simplex σ in the cell complex $\mathcal{K}$ is the set of all its cofaces in $\mathcal{K}$. The* closed star *is the minimal polytope that contains the star.*

The *barycentre* of a simplex is the barycentre of its vertices.

For a full-dimensional simplex we can define a unique Euclidean ball, the boundary of which contains the $d+1$ vertices of the simplex. This ball is called the *circumscribed ball* of the simplex (or *circumball*); its boundary is called *circumscribed sphere* (or *circumsphere*) of the simplex. The centre of the circumscribed ball is called the *circumcentre* and its radius is called the *circumradius.* We can also define a unique Euclidean ball inside the full-dimensional simplex that is tangent to all $d+1$ facets of the simplex. This ball is called the *inscribed ball* of the simplex. Similarly to circumscribed balls, we also define *inscribed sphere*, *incentre* and *inradius.*

The *height* of a simplex σ that falls on a facet τ of σ is the distance $d(v, \text{aff}(\tau))$ from the vertex v of σ not in τ to the affine hull of τ. For two facets τ and τ' of a polyhedron σ, such that τ and τ' have a common facet υ, we can define the *dihedral angle* between τ and τ', which is the angle between the affine hulls of τ and τ'.

For an edge, its *length* is the distance between its vertices. A simplex such that all its edges (1-dimensional faces) have the same length is called *regular.*

The maximum distance between two points in a compact subset S of $\mathbb{R}^d$ is called the *diameter* of S. If the compact set is a simplex, its diameter coincides with maximum edge length.

Definition 1.1.46 (Triangulation)**.** *A* triangulation *of a topological space X is a simplicial complex $\mathcal{K}$, homeomorphic to X, together with the homeomorphism $h : \mathcal{K} \to X$.*

Here, we only consider triangulations of adef Euclidean space $\mathbb{R}^d$. The cell complex $\mathcal{K}$ from Definition 1.1.46 will be embedded in $\mathbb{R}^d$, as we will see in Section 1.2. This makes that the homeomorphism h is the identity map of $\mathbb{R}^d$.

Definition 1.1.47 (Monohedral triangulation)**.** *Two polyhedra are called* similar*, if one can be obtained from the other by a transformation $\varphi : \mathbb{R}^d \to \mathbb{R}^d$ of the form:*

$$\varphi(x) = rOx + t,$$

where $r \in \mathbb{R}^d$ is a scale factor, O is an orthogonal $d \times d$ matrix and t is a translation vector.

If all d-dimensional simplices in a triangulation are similar, then such a triangulation is called monohedral.

Coxeter triangulations, which are the main topic of this thesis, are an example[3] of monohedral triangulations of $\mathbb{R}^d$. Some of these triangulations are also *Delaunay triangulations.*

Definition 1.1.48 (Delaunay triangulation)**.** *Let $\mathcal{T}$ be a triangulation of $\mathbb{R}^d$ (in the sense above). We will say that the triangulation $\mathcal{T}$ is* Delaunay*, if for all d-dimensional simplices $\sigma \in \mathcal{T}$, the interior of the circumscribed ball of σ does not contain any vertices in $\mathcal{T}$.*

[3] Coxeter triangulations are not the only example of monohedral triangulations. See Figure 1.12 for an example of a monohedral triangulation in $\mathbb{R}^2$, which is not a Coxeter triangulation.

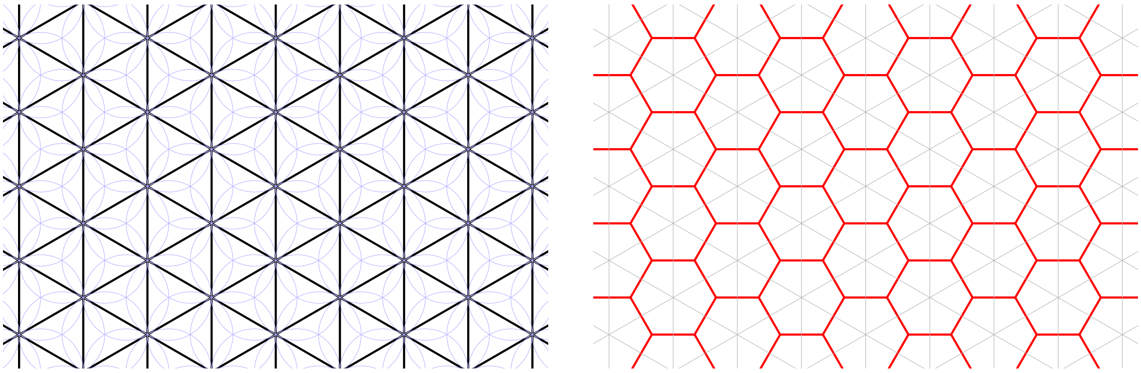


Figure 1.3: On the left: an example of a Delaunay triangulation of $\mathbb{R}^2$ with the circumscribed circles in faint blue. On the right: the corresponding Voronoi diagram in red.

When a triangulation $\mathcal{T}$ is a Delaunay triangulation, one can associate a so-called *Voronoi diagram* to it:

Definition 1.1.49 (Voronoi diagram)**.** *Let $\mathcal{T}$ be a Delaunay triangulation of $\mathbb{R}^d$. For any simplex σ in $\mathcal{T}$, the* dual *of σ is the convex hull of the circumcentres of all full-dimensional cofaces of σ. The set of duals of all simplices σ in $\mathcal{T}$ forms a cell complex, called* Voronoi diagram*.*

This definition of Voronoi diagrams is compatible to a somewhat more traditional definition of the Voronoi diagram of a point set [BCY18, Theorem 4.3]:

Definition 1.1.50. *A Voronoi diagram of a discrete point set P in $\mathbb{R}^d$ is a cell complex induced by a covering of $\mathbb{R}^d$ by the full-dimensional polyhedra* $\mathrm{Vor}(p)$ *for all $p \in P$, defined as follows:*

$$\mathrm{Vor}(p) = \{x \in \mathbb{R}^d \mid d(x,p) \leq d(x,q),\ \forall q \in P\}.$$

A Delaunay triangulation of the plane and the corresponding Voronoi diagram are illustrated in Figure 1.3.

Another important class of cell complexes in this thesis are defined by a set of hyperplanes.

Definition 1.1.51 (Arrangement)**.** *Let $\mathcal{H}$ be a set of hyperplanes in $\mathbb{R}^d$. The* arrangement *of $\mathcal{H}$ is the cell complex that consists of the faces of the polyhedra defined as (the closures of) the connected components of $\mathbb{R}^d \setminus (\cup_{H\in\mathcal{H}} H)$.*

Lastly, we will define the Cartesian products of two subsets of the Euclidean space.

Definition 1.1.52 (Cartesian product)**.** *Let $P \subseteq \mathbb{R}^m$ and $Q \subseteq \mathbb{R}^n$ be two subsets of Euclidean spaces of dimensions m and n respectively. We define the* Cartesian product *of P and Q, denoted by $P \times Q$, as the following subset of $\mathbb{R}^{m+n}$:*

$$P \times Q = \{(p,q) \in \mathbb{R}^{m+n} \mid p \in \mathbb{R}^m,\ q \in \mathbb{R}^n\}.$$

In the thesis, we will be mostly concerned with the Cartesian products of polytopes. Whenever we refer to the Cartesian products of polytopes, we will use the following classical result.

Lemma 1.1.53 ([Zie12, Chapter 0])**.** *The Cartesian product of two polytopes is a polytope.*

1.1.6 Real RAM computational model for the description of algorithms

The computation model that we adopt when describing the algorithms in the thesis is that of *real random-access machine* (or *real RAM*). In this model, each memory unit can hold a real value with unbounded precision. We will assume that all the following operations can be executed in $O(1)$ time:

1. reading a real number stored in a memory location,
2. comparing between any two real numbers,
3. the four arithmetic operations: addition, subtraction, multiplication and division,
4. the integer part (floor) computation.

As such, when discussing algorithms in this chapter, we distance ourselves from the problems that are related to numerical accuracy, although they might be relevant in practice.

Representation of Cartesian coordinates. We store the Cartesian coordinates of a given point $x \in \mathbb{R}^d$ in a data structure that supports random access. This means that the access to any Cartesian coordinate $x_i \in \mathbb{R}$ in a tuple $(x_1, \dots, x_d)$ is done in time $O(1)$. The space complexity to store one tuple of Cartesian coordinates is $O(d)$.

1.2 Root systems and Coxeter triangulations

The Coxeter triangulations we use originate in group theoretical studies of reflections. This section provides a group-theoretical background for and gives a brief introduction to Coxeter triangulations.

This section follows the lecture notes by Top [Top].

We first explain root systems as sets of vectors generated by undirected graphs. To an undirected graph without loops we associate a symmetric matrix. If the matrix is positive definite, it defines a (non-Euclidean) inner product and (non-Euclidean) orthogonal reflections. These orthogonal reflections generate a group and a set of vectors, called a root system. A classification of undirected graphs that generate finite root systems is presented.

We then proceed to study the geometrical properties of root systems in Euclidean space. Ultimately, we state the classification of root systems, which expands on the classification of the undirected graphs presented earlier.

After this, we proceed to define root lattices. Finally, we will give essential definitions and results from Chapter 4 of Humphreys [Hum92] on affine reflection groups and conclude by defining Coxeter triangulations and classifying them.

1.2.1 Graphs and Cartan matrices

We consider undirected connected graphs without loops with d nodes $\{n_1, \dots, n_d\}$. One can associate an incidence matrix $A = (a_{ij})$ to any such graph. This is the symmetric $d \times d$-matrix with $a_{ij} = 1$ if there is an edge between n_i and n_j and $a_{ij} = 0$ otherwise.

Definition 1.2.1 (Cartan matrix)**.** *The Cartan matrix of a graph Σ with incidence matrix A is defined by $C = 2I - A$, where I is the identity matrix.*

Note that Cartan matrices are symmetric. To each $d \times d$ Cartan matrix C we can associate a symmetric bilinear form $\langle \cdot, \cdot \rangle_C$ on $\mathbb{R}^d$ defined by $\langle u, v \rangle_C = u^t C v$, with $u, v \in \mathbb{R}^d$ and where u^t denotes the transposition of u.

We are now interested in identifying the graphs for which the symmetric bilinear form is positive definite and therefore defines an inner product. Each such $d \times d$ Cartan matrix gives us d linear maps σ_i, one for each vector e_i of the canonical basis, defined by:

$$\begin{array}{rll} \sigma_i : & \mathbb{R}^d & \to \mathbb{R}^d \\ & x & \mapsto x - \langle x, e_i \rangle_C \, e_i = x - (x^t C e_i) e_i. \end{array}$$

If the bilinear form $\langle \cdot, \cdot \rangle_C$ is positive definite, the linear map σ_i is the orthogonal reflection[4] through the hyperplane, which is orthogonal to e_i with respect to the inner product determined by C.

Definition 1.2.2 (Weyl group). *The Weyl group W_Σ of a graph Σ is the group of invertible linear maps generated by all σ_i.*

The roots *of Σ are the vectors in the set*

$$R_\Sigma = \{\sigma(e_i) \mid 1 \leq i \leq d, \sigma \in W_\Sigma\}.$$

We now have the following:

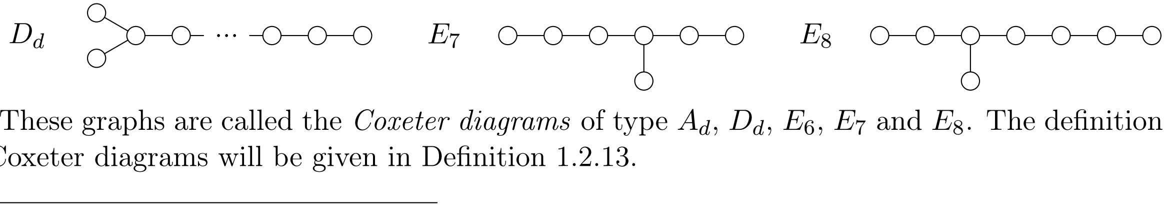
Theorem 1.2.3 ([Hum92, Section 1.3]). *For a graph Σ with Cartan matrix C, Weyl group W_Σ and root set R_Σ, the following three statements are equivalent:*

- *$\langle \cdot, \cdot \rangle_C$ is positive definite, that is it defines an inner product.*
- *The root set R_Σ is finite.*
- *The Weyl group W_Σ is finite.*

Section 1.8 of [Hum92] also gives us the following proposition:

Proposition 1.2.4. *For any graph Σ with a positive definite inner product, the action of elements of Weyl group W_Σ is simply transitive on the root set R_Σ.*

The connected graphs Σ that give a positive definite form $\langle \cdot, \cdot \rangle_C$ have been classified to be the following (see Section 2.4 of [Hum92]):



These graphs are called the *Coxeter diagrams* of type A_d, D_d, E_6, E_7 and E_8. The definition of Coxeter diagrams will be given in Definition 1.2.13.

[4] Usually, there is a 2 factor in the definition of the orthogonal reflection. This factor is hidden in the definition of Cartan matrix C.

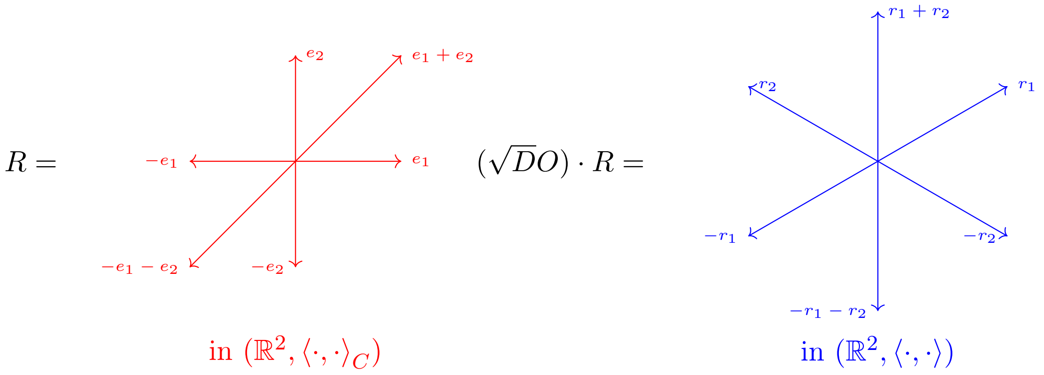


Figure 1.4: An example of the root sets of the A_2 diagram before and after multiplying by $\sqrt{D}O$.

1.2.2 Root systems in Euclidean space

We now want to make the Weyl groups as concrete as possible. To do this, we need to find vectors $r_1, \dots r_d \in \mathbb{R}^m$, for $m \geq d$, and $\mathbb{R}^m$ endowed with the standard inner product $\langle \cdot, \cdot \rangle$, such that $\langle r_i, r_j \rangle = c_{ij}$. These vectors are linearly independent because the Cartan matrix C is invertible. In general, such a matrix with scalar products as coefficients is called a *Gram* matrix.

Proposition 1.2.5. *Let $C = (c_{ij})$ be a positive definite Cartan matrix. There exist vectors $r_1, \dots, r_d \in \mathbb{R}^d$, such that $\langle r_i, r_j \rangle = c_{ij}$.*

Proof. The matrix C is positive definite, so it can be diagonalized by an orthogonal matrix O as $O^t D O = C$. Let us write $D = (d_{ij})$, with $d_{ij} = 0$, if $i \neq j$. If $\sqrt{D}$ denotes the matrix with $\sqrt{d_{ii}}$ on the diagonal, the vector r_i can be found as the ith column of the matrix $\sqrt{D}O$ (see Figure 1.4). □

Remark 1.2.6. *Note that this choice is not unique. However, in the context of root systems, nicer roots can be chosen, if we allow the roots to lie in $\mathbb{R}^m$ for $m > d$. According to Humphreys [Hum92, Section 2.10], the nice choices are not so obvious, and historically arose from close scrutiny of simple Lie algebras.*

As we have seen in the previous section, a root set R of each of the diagrams A_d, D_d, E_6, E_7 and E_8 is stable under the reflections of its roots. These reflections therefore generate a finite Weyl group. We will now construct more of finite Weyl groups based on root systems in Euclidean space in a bit larger sense than what was discussed before.

Definition and properties. We work in $\mathbb{R}^d$ endowed with the standard inner product $\langle \cdot, \cdot \rangle$. For $r \in \mathbb{R}^d$, with $r \neq 0$, the reflection σ_r in the hyperplane $\{v \in \mathbb{R}^d | \langle v, r \rangle = 0\}$ is given by

$$\sigma_r(x) = x - 2\frac{\langle x, r \rangle}{\langle r, r \rangle} r.$$

We can now redefine root systems from a completely geometric point of view.

Definition 1.2.7 (Root system)**.** *A root system in $\mathbb{R}^d$ is a finite set $R \subset \mathbb{R}^d$ that satisfies:*

- *$0 \notin R$ and R contains a basis of $\mathbb{R}^d$,*

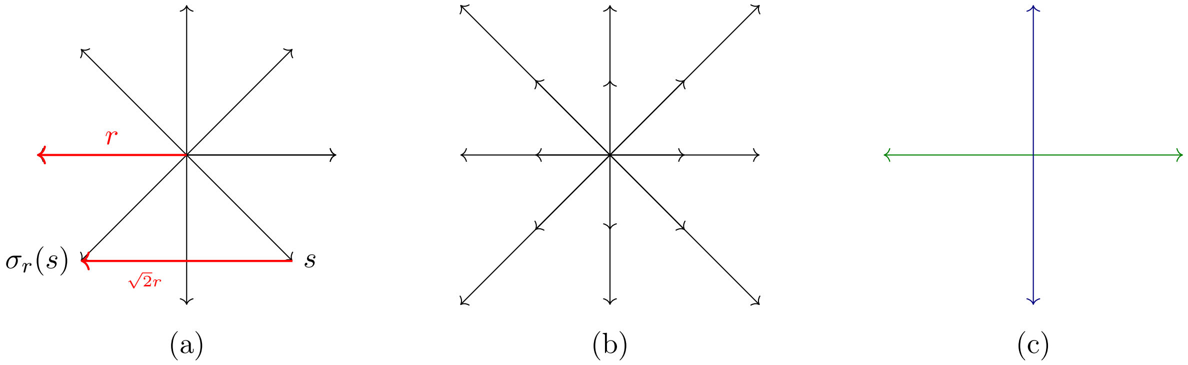


Figure 1.5: Examples of root systems that are (a) not crystallographic, (b) not reduced, (c) not irreducible.

- *for all $r \in R$, $\sigma_r(R) \subset R$.*

A root system is called crystallographic *if for all $r, s \in R$, $\sigma_r(s) - s$ is an integer multiple of r.*

A root system is called reduced *if $r \in R$ and $\lambda r \in R$ imply $\lambda = \pm 1$.*

A root system is called irreducible *if there is no decomposition $R = R_1 \cup R_2$ with $R_1 \neq \emptyset \neq R_2$ and $\langle r_1, r_2 \rangle = 0$ for all $r_1 \in R_1$ and $r_2 \in R_2$ (that is R_1 and R_2 are orthogonal to one another).*

In Figure 1.5, we present examples of root systems that lack exactly one of the properties from Definition 1.2.7. In the example (a) in Figure 1.5, the roots in the presented root system lie on a regular octagon. It is easy to check that the root system is reduced and irreducible. As we see, the image of the root s by a reflection σ_r differs from s by a vector $\sqrt{2}r$, which is not an integer multiple of r. Therefore the presented root system is not crystallographic.

In the example (b) in Figure 1.5 there are more than two collinear roots, so the root system is not reduced. It can be easily checked that the root system is, however, crystallographic and irreducible.

In the example (c) in Figure 1.5 the set of vertical roots and the set of horizontal roots are orthogonal one to another. Therefore, the root system is not irreducible. It can be easily checked that the root system is however crystallographic and reduced.

As before, the *Weyl group* of a root system R is the (finite) group W generated by the reflections σ_r for all $r \in R$. The root systems that are not reduced or irreducible are not interesting for the classification of Weyl groups for the following reasons.

If a root system is not reduced, we can associate a corresponding reduced root system that shares the same Weyl group. Now, assume that a root system R is not irreducible. Because R is finite, by a simple induction, we can represent R as a union:

$$R = \bigcup_{i=1}^{m} R_i$$

with:

1. all R_i being irreducible root systems in the sense of Definition 1.2.7, and

2. all R_i being orthogonal one to another.

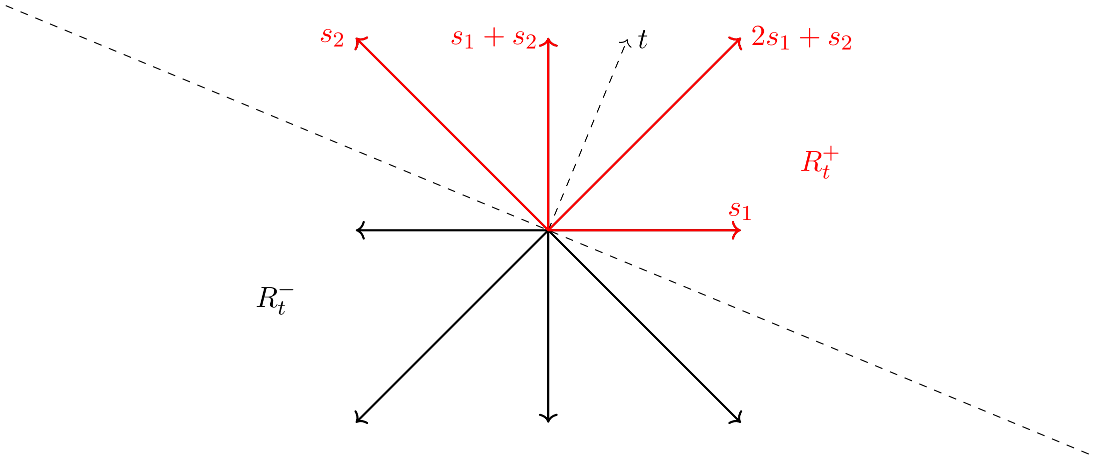


Figure 1.6: Illustration of positive and negative roots in a two-dimensional root system for a given vector $t \in \mathbb{R}^2$. The simple root set S_t consists of s_1 and s_2.

The Weyl group that corresponds to R is, in fact, the direct product of the Weyl groups that correspond to the R_i.

The crystallographic property of the root systems will be important for the definition of root lattices and affine Weyl groups, so we will focus on root systems that are crystallographic. From this point onward we will assume that every root system under consideration is crystallographic, reduced and irreducible, unless stated otherwise.

Simple roots. We will now define *simple roots* that form a basis of a root system.

Definition 1.2.8. *Assume that for a root system R we are given an arbitrary $t \in \mathbb{R}^d$ such that $\langle t, r\rangle \neq 0$ for all $r \in R$. The root system now decomposes as $R = R_t^+ \cup R_t^-$ into* positive roots *$R_t^+ = \{r \in R \mid \langle r, t\rangle > 0\}$ and* negative roots *$R_t^- = \{r \in R \mid \langle r, t\rangle < 0\}$ (see Figure 1.6). A root $r \in R_t^+$ is called* decomposable *if $r = r_1 + r_2$, with $r_1, r_2 \in R_t^+$, and a root $r \in R_t^+$ is called* simple *if it is not decomposable. The set of simple roots with respect to t is denoted by S_t.*

An important result is that all sets of simple roots S_t are the same up to a reflection and a rotation.

Lemma 1.2.9. *The action of the Weyl group that corresponds to a root system R on the collection of all sets of simple roots of R is simply transitive.*

The proof of the lemma can be found in Section 1.8 of Humphreys [Hum92]. Because all S_t are similar one to another (in the sense of Definition 1.1.47), we will omit the index t.

Lemma 1.2.10. *Let R be a root system, and S be a set of simple roots in R.*

- *For any two distinct simple roots $r, s \in S$, we have $\langle r, s\rangle \leq 0$.*
- *The set S forms a basis of $\mathbb{R}^d$.*
- *The Weyl group is generated by the reflections associated to the simple roots in S.*

We refer to Sections 1.3 and 1.5 of Humphreys [Hum92] for the proof of the lemma.

Definition 1.2.11 (Partial order on positive roots)**.** *Because simple roots S form a basis of the root system R, any root can be decomposed as a sum of simple roots with integer coefficients. Using the decomposition, the positive roots (see Definition 1.2.8) can be characterized as the roots that have non-negative coefficients when expressed in terms of simple roots. For any two positive roots r and r', we can write $r = \sum_{i=1}^{d} c_i s_i$ and $r' = \sum_{i=1}^{d} c'_i s_i$, where $S = \{s_1, \ldots, s_d\}$ and the coefficients c_i and c'_i are integers. This gives a partial order $\preccurlyeq$ on the set of roots R, which compares the coefficients of two roots meaning that $r \preccurlyeq r'$ if and only if $c_i \leq c'_i$ for all $1 \leq i \leq d$.*

Angle between roots. The angle φ, with $0 \leq \varphi \leq \pi$, between two roots r and s is given by $\cos\varphi = \frac{\langle r,s\rangle}{\|r\|\|s\|}$. Note that, due to the crystallographic condition, we have:

$$\sigma_r(s) - s = 2\frac{\langle r, s\rangle}{\langle r, r\rangle} r \in \mathbb{Z} \cdot r.$$

Definition 1.2.12. *For any two roots $r, s \in R$, we define:*

$$n(s, r) := 2\frac{\langle r, s\rangle}{\langle r, r\rangle} \in \mathbb{Z}.$$

This integer is called Cartan integer.

It follows that $4\cos^2\varphi = n(s,r)n(r,s) \in \mathbb{Z}$. In the case $\langle r, s\rangle \neq 0$, observe that the ratio of the squared norms of these two roots is given by $\frac{\langle s,s\rangle}{\langle r,r\rangle} = \frac{2\langle r,s\rangle}{\langle r,r\rangle}\frac{\langle s,s\rangle}{2\langle r,s\rangle} = \frac{n(s,r)}{n(r,s)}$. This gives us, up to symmetry, the following table (see also Figure 1.7):

$4\cos^2\varphi$	$n(s,r)$	$n(r,s)$	φ	length relation
4	2	2	0	$\|s\| = \|r\|$
4	-2	-2	π	$\|s\| = \|r\|$
3	3	1	$\pi/6$	$\|s\| = \sqrt{3}\|r\|$
3	-3	-1	$5\pi/6$	$\|s\| = \sqrt{3}\|r\|$
2	2	1	$\pi/4$	$\|s\| = \sqrt{2}\|r\|$
2	-2	-1	$3\pi/4$	$\|s\| = \sqrt{2}\|r\|$
1	1	1	$\pi/3$	$\|s\| = \|r\|$
1	-1	-1	$2\pi/3$	$\|s\| = \|r\|$
0	0	0	$\pi/2$	(undetermined)

By inspection of the table, we observe that the length ratio of any two roots[5] can only be 1, $\sqrt{2}$ or $\sqrt{3}$. It implies that there are at most two different norms of roots. In this case, we speak about *short* and *long* roots.

Because $n(r,s) \in \{-3,-2,-1,0\}$ and $n(s,r)n(r,s) = 4\cos^2\varphi$, the angle φ between r and s equals one of $\frac{\pi}{2}$, $\frac{2\pi}{3}$, $\frac{3\pi}{4}$ or $\frac{5\pi}{6}$ (see Figure 1.7).

From Lemma 1.2.9, we know that all simple root sets are similar one to another, therefore have the same angles. The information about the angles of simple roots can be represented in a graph.

Definition 1.2.13 (Coxeter diagram)**.** *Let R be a root system and S be a set of simple roots. The* Coxeter diagram *of R consists of a graph with the following data: for each $r \in S$, we insert*

[5] If all three ratios 1, $\sqrt{2}$ and $\sqrt{3}$ between norms were present, we would also have ratios $\sqrt{6}$ or $\sqrt{\frac{3}{2}}$, which are not possible.

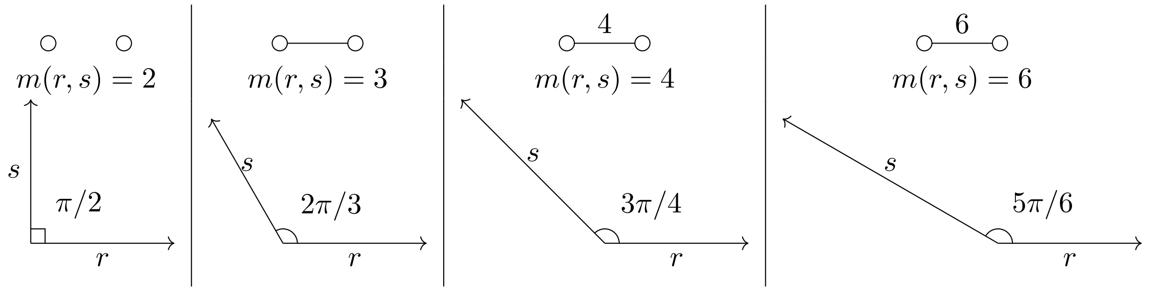


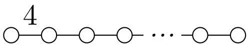
Figure 1.7: All possible angles between two simple roots r and s and the corresponding values of $m(r, s)$. On the top, the corresponding edge in a Coxeter diagram is shown.

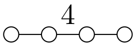
one vertex. For every pair $r \neq s$ in S with $\langle r, s\rangle \neq 0$ we define a number $m(r, s) \in \{2, 3, 4, 6\}$, such that $\frac{\pi}{m(r,s)} = \arccos|\cos\varphi|$, where φ is the angle between r and s (see Figure 1.7). We then insert an edge between r and s and write the number $m(r, s)$ next to it.

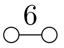
We further follow the convention not to draw an edge labelled 2 and not to denote label 3 next to an edge.

Classification of root systems. We will now state the complete classification of Coxeter diagrams of crystallographic, reduced and irreducible root systems.

Theorem 1.2.14. *The complete list of Coxeter diagrams of crystallographic, reduced and irreducible root systems consists of A_d, D_d, E_6, E_7 and E_8 and the following diagrams:*

$B_d (= C_d)$ 

F_4 

G_2 

In this list, the root system C_d is defined as dual to B_d (see Definition 1.2.19). As dual root systems, they share the same Coxeter diagram and Weyl group (see Section 1.2.3).

See for example Sections 2.4 and 2.7 of Humphreys [Hum92] for a proof and more information on the classification of Coxeter diagrams.

The following theorem gives explicit sets of simple roots in Euclidean space:

Theorem 1.2.15 ([Bou02])**.** *Let $\{e_1, \ldots, e_d\}$ be the canonical basis in $\mathbb{R}^d$. The complete list of simple root sets (up to scale, rotation and permutation) is the following:*

- $\mathbf{A_d}$ *(in $\mathbb{R}^{d+1}$): $s_1 = e_1 - e_2$, $s_2 = e_2 - e_3$, $\ldots$, $s_d = e_d - e_{d+1}$.*
- $\mathbf{B_d}$*: $s_1 = e_1 - e_2$, $s_2 = e_2 - e_3$, $\ldots$, $s_{d-1} = e_{d-1} - e_d$, $s_d = e_d$.*
- $\mathbf{C_d}$*: $s_1 = e_1 - e_2$, $s_2 = e_2 - e_3$, $\ldots$, $s_{d-1} = e_{d-1} - e_d$, $s_d = 2e_d$.*
- $\mathbf{D_d}$*: $s_1 = e_1 - e_2$, $s_2 = e_2 - e_3$, $\ldots$, $s_{d-1} = e_{d-1} - e_d$, $s_d = e_{d-1} + e_d$.*
- $\mathbf{E_6}$ *(in $\mathbb{R}^8$): $s_1 = \frac{1}{2}(e_1 + e_8) - \frac{1}{2}(e_2 + e_3 + e_4 + e_5 + e_6 + e_7)$, $s_2 = e_1 + e_2$, $s_3 = e_2 - e_1$, $s_4 = e_3 - e_2$, $s_5 = e_4 - e_3$, $s_6 = e_5 - e_4$.*

- $\mathbf{E_7}$ *(in* $\mathbb{R}^8$*):* $s_1 = \frac{1}{2}(e_1 + e_8) - \frac{1}{2}(e_2 + e_3 + e_4 + e_5 + e_6 + e_7)$, $s_2 = e_1 + e_2$, $s_3 = e_2 - e_1$, $s_4 = e_3 - e_2$, $s_5 = e_4 - e_3$, $s_6 = e_5 - e_4$, $s_7 = e_6 - e_5$.

- $\mathbf{E_8}$*:* $s_1 = \frac{1}{2}(e_1 + e_8) - \frac{1}{2}(e_2 + e_3 + e_4 + e_5 + e_6 + e_7)$, $s_2 = e_1 + e_2$, $s_3 = e_2 - e_1$, $s_4 = e_3 - e_2$, $s_5 = e_4 - e_3$, $s_6 = e_5 - e_4$, $s_7 = e_6 - e_5$, $s_8 = e_7 - e_6$.

- $\mathbf{F_4}$*:* $s_1 = e_2 - e_3$, $s_2 = e_3 - e_4$, $s_3 = e_4$, $s_4 = \frac{1}{2}(e_1 - e_2 - e_3 - e_4)$.

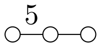
- $\mathbf{G_2}$ *(in* $\mathbb{R}^3$*):* $s_1 = e_1 - e_2$, $s_2 = -2e_1 + e_2 + e_3$.

This list is important for the calculations in Section 2.4.

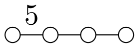
Remark 1.2.16. *Later in Section 1.2.4, we define the hyperplane arrangements from crystallographic reduced and irreducible root systems. It is important to note that the choice of simple root sets as in Theorem 1.2.15 of types* $\tilde{A}_d$, $\tilde{E}_6$, $\tilde{E}_7$ *and* $\tilde{G}_2$ *do not define hyperplane arrangements that are essential*[6] *in their respective ambient Euclidean spaces.*

Remark 1.2.17. *If we drop the hypothesis that the root system is crystallographic, then the list of Coxeter diagrams also includes the following diagrams:*

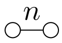
H_3



H_4



$I_2(n)$



The corresponding reflection groups are the symmetry groups[7] *of the icosahedron* (H_3)*, the isometry group of a regular 120-sided solid with dodecahedral faces in* $\mathbb{R}^4$ (H_4) *and the dihedral group* $\mathfrak{D}_n$ ($I_2(n)$)*. We refer to Humphreys [Hum92, Sections 2.8 and 2.13] for further reading.*

1.2.3 Root lattices

We can now define lattices based on the roots we discussed above. These lattices will be essential later in Section 1.2.4.

Definition 1.2.18 (Root lattice)**.** *The root lattice* Λ_R *of a crystallographic reduced and irreducible root system* R *is defined as:*

$$\Lambda_R = \left\{ \sum_{r \in R} n_r r \mid n_r \in \mathbb{Z} \right\}.$$

It is indeed a lattice in the sense that it is a group under addition of vectors, contains a basis of $\mathbb{R}^d$ and any bounded region contains only a finite number of elements.

Definition 1.2.19 (Dual root system)**.** *For each root* $r \in R$*, define its* coroot *(or* dual *root) to be* $r^\vee = \frac{2r}{\langle r,r \rangle}$*. The set of coroots forms a root system [Hum92, Section 2.9] called* dual root system *and is denoted by* $R^\vee$*.*

[6] The hyperplane arrangement is *essential* if the set of all normal vectors of the hyperplanes spans the ambient space.

[7] Group of transformations under which a polytope is invariant.

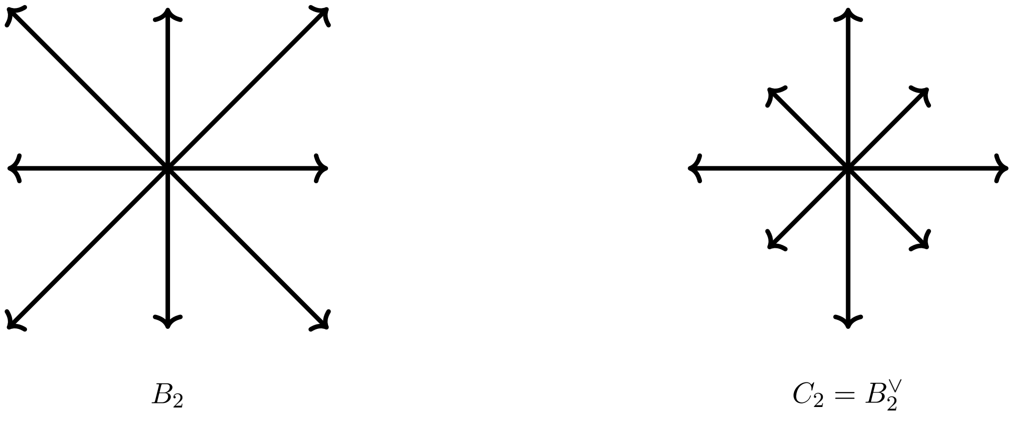


Figure 1.8: An example of two dual root systems B_2 and C_2 in $\mathbb{R}^2$.

Observe that coroots are rescaled versions of roots such that the inner product between the root and the coroot is 2, see Figure 1.8. It implies that coroots share the same reflections σ_r as roots. Therefore the dual root system generates the same Weyl group W. In most cases, the root system $R^\vee$ is identical to R up to scale and rotation; however, the dual root systems B_d and C_d are not isomorphic (see [Hum92, Section 2.9], see also Figure 1.8). Short roots in a system R of type B_d give rise to long roots in a system $R^\vee$ of type C_d and vice versa.

The duals of the three root systems in $\mathbb{R}^2$ are illustrated in Figure 1.10.

Remark 1.2.20. *Using the definition of the coroot, the Cartan integer $n(r, s)$ from Definition 1.2.12 can be interpreted as:*

$$n(r,s) = 2\frac{\langle r,s\rangle}{\|r\|^2} = \left\langle r^\vee, s\right\rangle .$$

Definition 1.2.21 (Coroot lattice)**.** *Similarly to root lattices, we define the* coroot lattice*:*

$$\Lambda_{R^\vee} = \left\{ \sum_{r\in R} n_r r^\vee \mid n_r \in \mathbb{Z} \right\}.$$

Another important family of lattices is so-called *weight lattices*:

Definition 1.2.22 (Weight and coweight lattices)**.** *The set of points which has an integer inner product with all coroots is called the* weight lattice*:*

$$\Lambda_R^w = \left\{ x \in \mathbb{R}^d \mid \left\langle x, r^\vee \right\rangle \in \mathbb{Z},\ \forall r \in R \right\}.$$

Similarly, the coweight lattice *is defined as:*

$$\Lambda_{R^\vee}^w = \left\{ x \in \mathbb{R}^d \mid \langle x, r\rangle \in \mathbb{Z},\ \forall r \in R \right\}.$$

For a given root system, the coweight lattice in Definition 1.2.22 has a strong connection with the set of vertices of the Coxeter triangulation that will be defined in the Section 1.2.4 (see also Lemma 2.4.3).

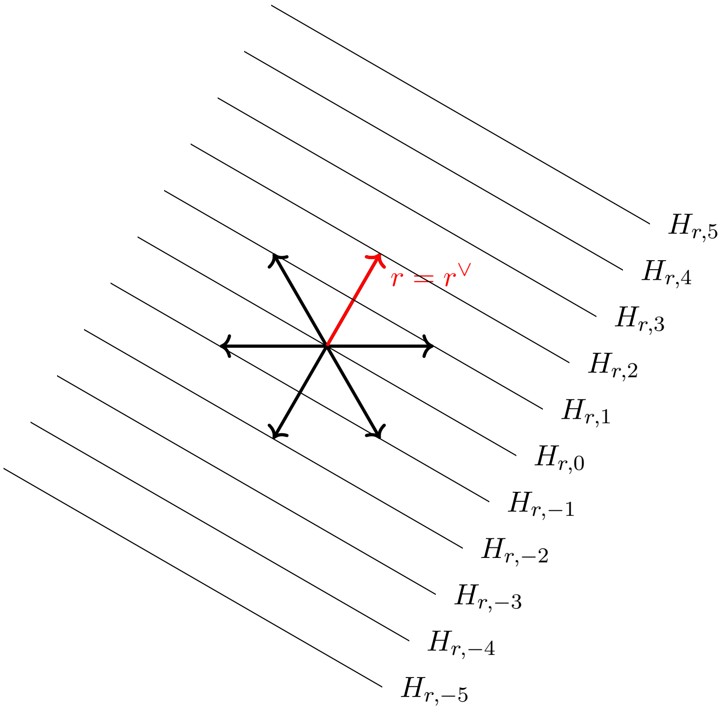


Figure 1.9: Root system A_2 and the hyperplanes $H_{r,k}$ corresponding to the root r. Here we have $\|r\|^2 = 2$, therefore the primal and the coroots coincide and the hyperplane $H_{r,1}$ goes halfway through $r = r^\vee$, so the image of reflecting 0 by $\sigma_{r,1}$ is $r^\vee$.

1.2.4 Affine reflection groups

The goal for us now is to define a triangulation of the Euclidean space associated to every root system.

First, we need the following definitions:

Definition 1.2.23. *We call a* family of parallel hyperplanes relative to a normal vector $u \in \mathbb{R}^d$ and indexed by $\mathbb{Z}$ *a set of hyperplanes* $\mathcal{H}_u = \{H_{u,k},\ k \in \mathbb{Z}\}$ *with* $H_{u,k} = \{x \in \mathbb{R}^d \mid \langle x, u\rangle = k\}$.

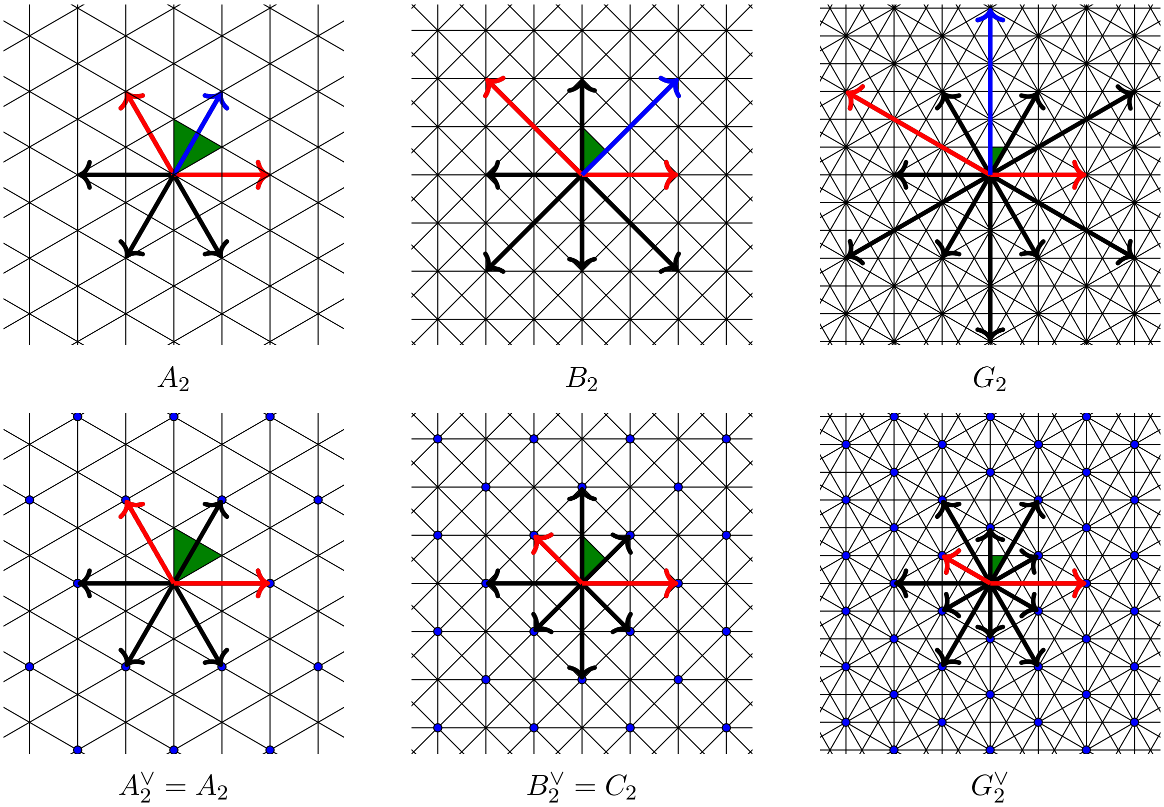
An example of a family of parallel hyperplanes relative to a root in a root system is illustrated in Figure 1.9.

Definition 1.2.24 (Affine Weyl group)**.** *Let $R \subset \mathbb{R}^d$ be a finite root system. The set of affine hyperplanes $H_{r,k}$ for all $r \in R$ and $k \in \mathbb{Z}$ will be denoted as $\mathcal{H}$. To each $H_{r,k}$ we can associate an affine reflection*[8] *$\sigma_{r,k} : x \mapsto x - (\langle x, r\rangle - k)r^\vee$. These reflections generate a subgroup of the group of affine transformations of $\mathbb{R}^d$, which is called the* affine Weyl group *and is denoted by W_a.*

Remark 1.2.25. *Positive and negative roots define the same hyperplanes and affine reflections. So the definition does not change if one restricts to only positive roots.*

Roughly speaking, the affine Weyl group is a combination of the Weyl group and translations along a lattice. This can be made more precise using the coroot lattice $\Lambda_{R^\vee}$ defined in Definition 1.2.21:

[8] Note that the usual factor 2 is hidden in the definition of $r^\vee$.



A_2 B_2 G_2

$A_2^\vee = A_2$ $B_2^\vee = C_2$ $G_2^\vee$

Figure 1.10: Above: Three root systems and corresponding affine hyperplanes in $\mathbb{R}^2$. Simple roots are marked in red, the highest root in blue and the fundamental domain in green. Each triangle in the background is an alcove. Below: The dual root systems put on the same grid of affine hyperplanes.

Proposition 1.2.26. *Let R be a root system, W be a corresponding Weyl group and T be the translation group corresponding to the coroot lattice $\Lambda_{R^\vee}$. The group T is a normal subgroup of the affine Weyl group W_a and W_a is a semidirect product $T \rtimes W$.*

We refer to Section 4.2 of [Hum92] for more information.

The positions of hyperplanes $H_{r,k}$ with respect to primal and dual root systems in $\mathbb{R}^2$ are illustrated in Figure 1.10. We notice that the open regions in between the hyperplanes $H_{r,k}$ are similar triangles. These regions are called alcoves and are the subject of our study in the following. We now formalize:

Definition 1.2.27 (Alcove)**.** *Define $\mathcal{A}$ to be the set of connected components of $\mathbb{R}^d \setminus \bigcup_{H \in \mathcal{H}} H$. Each element in $\mathcal{A}$ is called an* alcove*.*

Let R_+ be a set of positive roots and S the corresponding simple system. An alcove is characterized by a set of inequalities of the form: $\forall r \in R_+, k_r < \langle x, r \rangle < k_r + 1$, with k_r integers. We will denote by A^o the particular alcove for which all k_r are equal to 0:

$$A^o = \{x \in \mathbb{R}^d \mid \forall r \in R_+, 0 < \langle x, r \rangle < 1\}$$

Most of the inequalities that define A^o are redundant. If we want to eliminate the redundant inequalities, we first need to define the so called *highest root* (illustrated in Figure 1.10).

Proposition 1.2.28 (Existence and uniqueness of the highest root). *For a root system R and a set $S \subset R$ of simple roots, there is a maximum $\tilde{s} \in R^+$ for the partial order $\preccurlyeq$ (see Definition 1.2.11), which is called the highest root.*

For the proof of Proposition 1.2.28 and more details on the highest root, we refer to Section 2.9 of Humphreys [Hum92].

The following proposition states that there are exactly $d + 1$ hyperplanes that define the facets of A^o: d hyperplanes corresponding to simple roots and one corresponding to the highest root (see also Figure 1.10):

Proposition 1.2.29 ([Hum92, Section 4.3]). *Let R be a root system and $S \subset R$ a set of simple roots. The alcove A^o is an open simplex delimited by $d + 1$ hyperplanes. Of them, d hyperplanes are of the form $H_{s,0} = \{x \in \mathbb{R}^d \mid \langle x, s\rangle = 0\}$, one for each simple root $s \in S$ and the final hyperplane is $H_{\tilde{s},1} = \{x \in \mathbb{R}^d \mid \langle x, \tilde{s}\rangle = 1\}$ where $\tilde{s}$ is the highest root.*

Now, we are interested in the closure of the alcove A^o, which is a full-dimensional simplex. This simplex will be the starting point of the triangulations we will now construct.

Definition 1.2.30. *Let R be a root system and $S \subset R$ a set of simple roots. Let A^o be the alcove as above. The closure F of A^o is called the* fundamental domain *(or the* fundamental simplex*) of R with respect to S.*

The reason behind the name *fundamental domain* is the following proposition.

Proposition 1.2.31 ([Hum92, Section 4.3]). *The affine Weyl group W_a acts simply transitively on $\mathcal{A}$.*

By the simple transitivity of the action of W_a, all alcoves are similar to the fundamental alcove. This means that the closures of elements of $\mathcal{A}$ are all full-dimensional simplices in a monohedral triangulation of $\mathbb{R}^d$.

Corollary 1.2.32. *The arrangement of $\mathcal{H}$ is a triangulation of $\mathbb{R}^d$.*

We call such triangulations *Coxeter triangulations.*

Definition 1.2.33. *The* Coxeter diagrams *for affine Weyl groups are defined in the same way as in Definition 1.2.13, except that we use not only the simple roots, but also the opposite of the highest root. This means that the nodes correspond to simple roots and the opposite of the highest root, and the edges correspond to angles between them.*

The classification of all affine Weyl groups is possible thanks to the notion of subgraph and the following lemma:

Definition 1.2.34. *A subgraph of a Coxeter diagram G is a Coxeter diagram G' obtained by omitting some nodes (and adjacent edges) of G' or by decreasing the labels on one or more edges.*

Lemma 1.2.35 ([Hum92, Corollary 2.6]). *Every subgraph of an affine Coxeter diagram has a positive definite Cartan matrix.*

Based on these results we can now derive:

Theorem 1.2.36. *The complete list of affine Weyl groups and the corresponding Coxeter diagrams is as follows:*

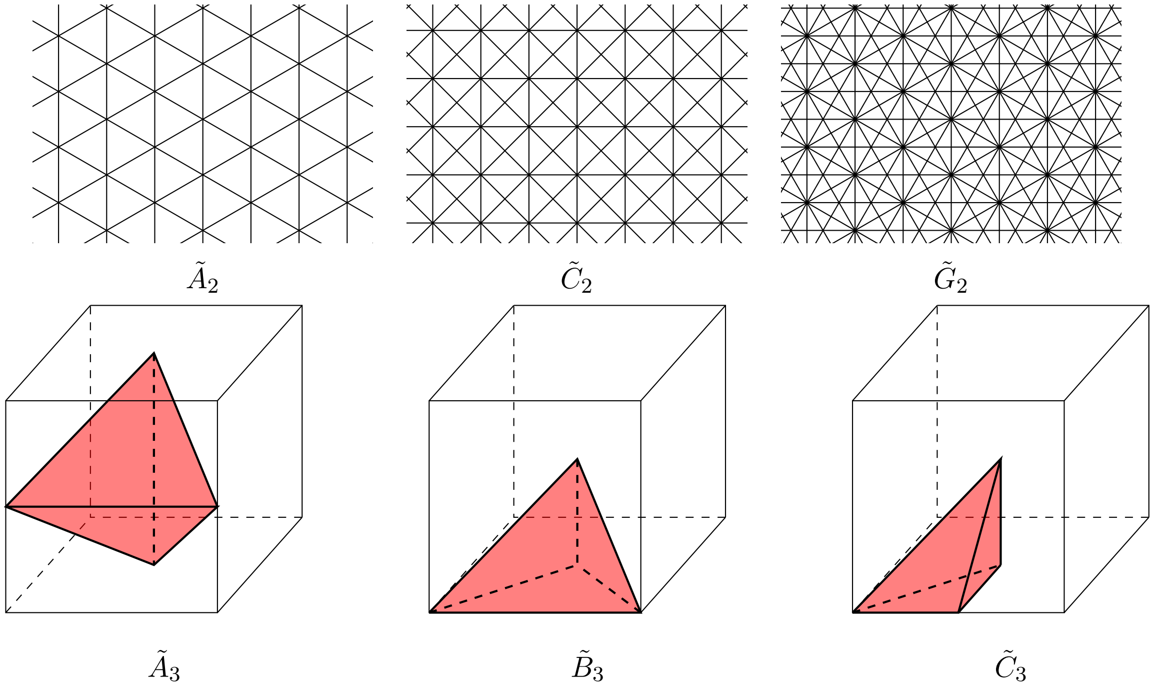
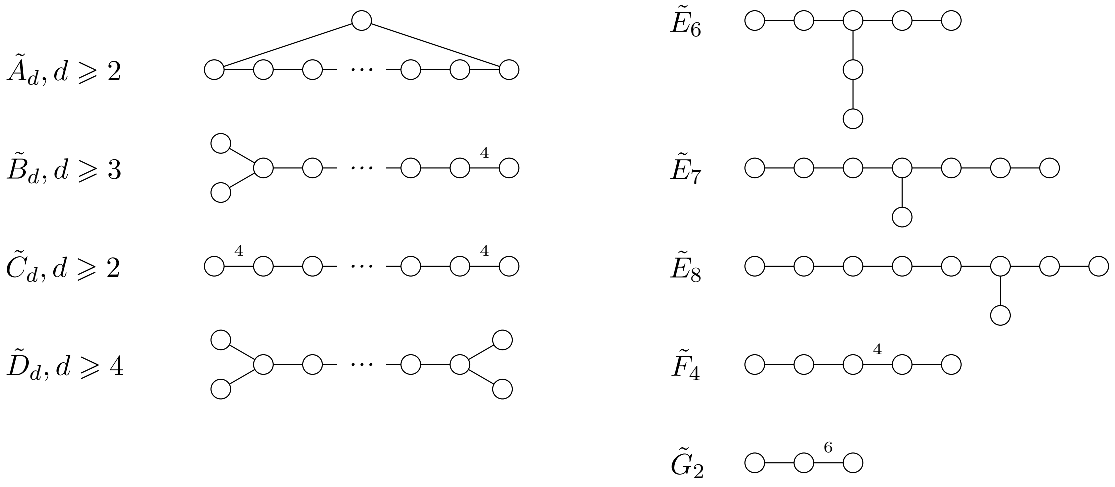


Figure 1.11: On the top: Coxeter triangulations in $\mathbb{R}^2$. On the bottom: simplices of Coxeter triangulations in $\mathbb{R}^3$ represented as a portion of a cube.


$\tilde{A}_d, d \geqslant 2$
$\tilde{B}_d, d \geqslant 3$
4
$\tilde{C}_d, d \geqslant 2$
4
4
$\tilde{D}_d, d \geqslant 4$
$\tilde{E}_6$
$\tilde{E}_7$
$\tilde{E}_8$
$\tilde{F}_4$
4
$\tilde{G}_2$
6

For a proof, we refer to Sections 2.5 and 2.7 of [Hum92].

All three two-dimensional Coxeter triangulations are presented on the top of Figure 1.11. On the bottom of Figure 1.11, we illustrate the simplices of the three-dimensional Coxeter triangulations with vertices placed in vertices, centres of edges, centres of faces and at the centre of a cube.

We now have a classification of Coxeter triangulations, whose properties and applications are the topic of this thesis.

Remark 1.2.37. *It is important to note that Coxeter triangulations of $\mathbb{R}^d$ are monohedral triangulations, generated by orthogonal reflections and that are arrangements of hyperplanes. The condition of being an arrangement of hyperplanes is important, as Figure 1.12 shows. There, we*

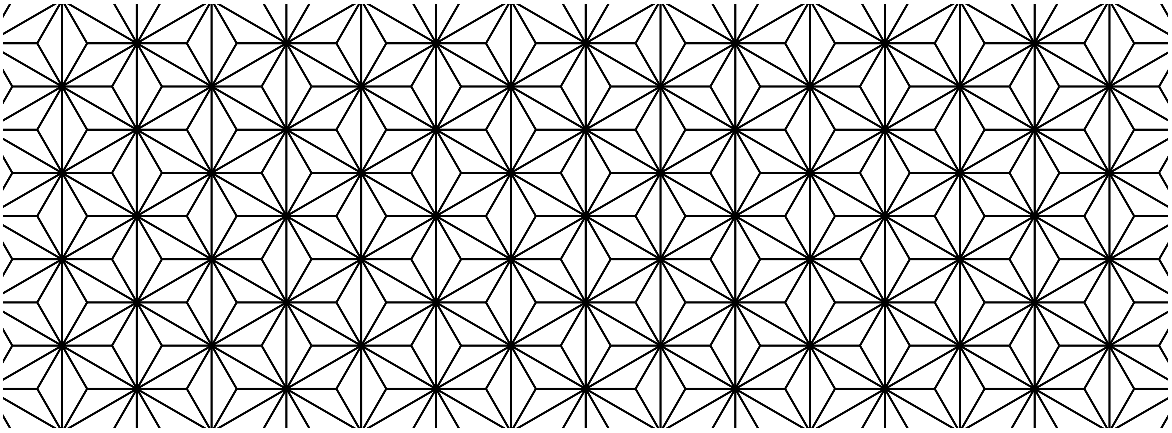


Figure 1.12: Example of a monohedral triangulation of $\mathbb{R}^2$ generated by orthogonal reflections that is not a Coxeter triangulation.

illustrate a triangulation of $\mathbb{R}^2$ *that is monohedral and generated by orthogonal reflections, but is not an arrangement of hyperplanes. It is easy to check that the triangulation in Figure 1.12 does not match any of two-dimensional Coxeter triangulations (shown on the top of Figure 1.11).*

Remark 1.2.38. *If we drop the irreducibility condition from Definition 1.2.7, we can still define affine Weyl groups. Let* R *be a crystallographic reduced root system. As in Section 1.2.2, we can represent* R *as a union of orthogonal irreducible root systems:* $R = \bigcup_{i=1}^{m} R_i$*. We have seen that the Weyl group that corresponds to* R *is the direct product of the Weyl groups that correspond to* R_i*. The same is true for the affine Weyl group of* R*: it is, again, the direct product of the affine Weyl groups that correspond to the root systems* R_i*.*

Let $d_1, \ldots, d_m$ *be the ranks of* $R_1, \ldots, R_m$ *respectively. An alcove is not a simplex any more, but it is, in fact, the* Cartesian product of m simplices *of dimensions* $d_1, \ldots, d_m$*. Instead of triangulations we now talk about cell complexes, each full-dimensional cell of which is a certain Cartesian product of simplices. Here are some examples of such decompositions:*

- $\tilde{A}_1 \times \ldots \times \tilde{A}_1$ *corresponds to a decomposition of* $\mathbb{R}^d$ *into cubes, which are Cartesian products of* d *segments.*

- $\tilde{A}_2 \times \tilde{A}_1$, $\tilde{B}_2 \times \tilde{A}_1$ *and* $\tilde{G}_2 \times \tilde{A}_1$ *correspond to decompositions of* $\mathbb{R}^3$ *into triangular prisms with bases being the fundamental triangles of* $\tilde{A}_2$, $\tilde{B}_2$ *and* $\tilde{G}_2$ *respectively. Each prism is the Cartesian product of the base and a segment.*

1.3 The Voronoi diagram of a Coxeter triangulation of type $\tilde{A}_d$

In this section we will give an overview of results on the cells of Voronoi diagrams of Coxeter triangulation of type $\tilde{A}_d$. This is well-defined, as we will establish in Section 2.4.1 that the Coxeter triangulations of type $\tilde{A}_d$ are Delaunay triangulations. The combinatorial structure of the Voronoi diagram (see Definition 1.1.50) of a Coxeter triangulation of type $\tilde{A}_d$ will be important for the coface generation algorithm in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ that we define in Chapter 3. The cells in the Voronoi diagrams of the Coxeter triangulations of type $\tilde{A}_d$ will also appear as the constructing blocks in the manifold tracing algorithm in Chapter 4.

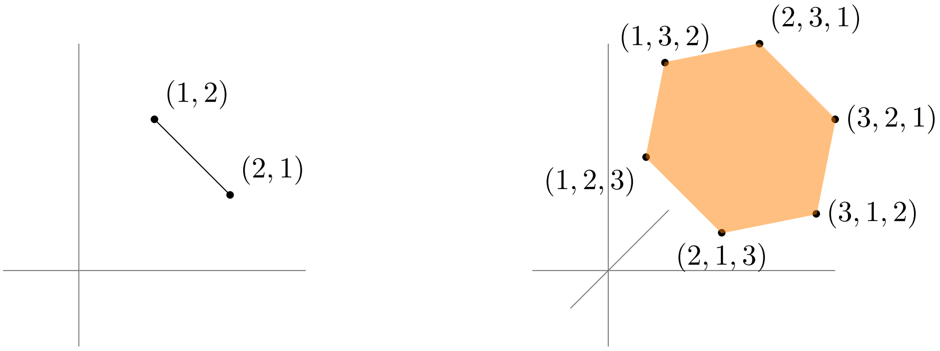


Figure 1.13: The one- and two-dimensional permutahedra defined as convex hulls of permutations in $\mathbb{R}^2$ and $\mathbb{R}^3$ respectively.

The Voronoi diagram of a Coxeter triangulation of type $\tilde{A}_d$ has been studied in [CS87, Chapter 21, Section 3.F]. Its cells are polytopes that are commonly known as *permutahedra*[9].

Definition 1.3.1 (Permutahedron)**.** *A* permutahedron *is a polytope, which is similar (in the sense of Definition 1.1.47) to the d-dimensional convex hull P of all points in $\mathbb{R}^{d+1}$, the coordinates of which are permutations of* $(1, 2, \dots, d+1) \in \mathbb{R}^{d+1}$*. More formally, this convex hull can be written as:*

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

To see why the polytope P is at most d-dimensional, observe that all vertices in P lie on the hyperplane:

$$\left\{ (x_1, \dots, x_{d+1}) \in \mathbb{R}^{d+1} \;\middle|\; \sum_{i=1}^{d+1} x_i = \frac{d(d+1)}{2} \right\}.$$

By further observing that there are $d+1$ affinely independent vertices in P, we can conclude that P is exactly d-dimensional (see for example [MK92, Lemma 3.4] for the whole proof).

The one-dimensional permutahedron is a segment, the two dimensional permutahedron is a regular hexagon (see Figure 1.13). The three-dimensional permutahedron is known as the truncated octahedron, illustrated in Figure 1.14.

1.3.1 Faces of a permutahedron

The faces of a permutahedron are not necessarily permutahedra. For example, the square faces on a three-dimensional permutahedron (see Figure 1.15) are not two-dimensional permutahedra (recall that two-dimensional permutahedra are hexagons). Instead, these square faces can be seen as the Cartesian products of two segments, which are one-dimensional permutahedra. The following lemma is the generalization of this observation.

Lemma 1.3.2 ([MK92])**.** *Any proper face of a permutahedron is the Cartesian product of some lower-dimensional permutahedra.*

The partially-ordered set of faces of a permutahedron (with the partial order from Lemma 1.1.43) has a special structure [Zie12, Chapter 0]. To any face of a permutahedron, we can associate a so-called ordered partition of the set $\{1, \dots, d+1\}$.

[9] In the literature, other spellings are sometimes used. Some works call this polytope a *permutohedron* (for example in [CS87]), or even a *permutation polytope* (for example in [Tho93])

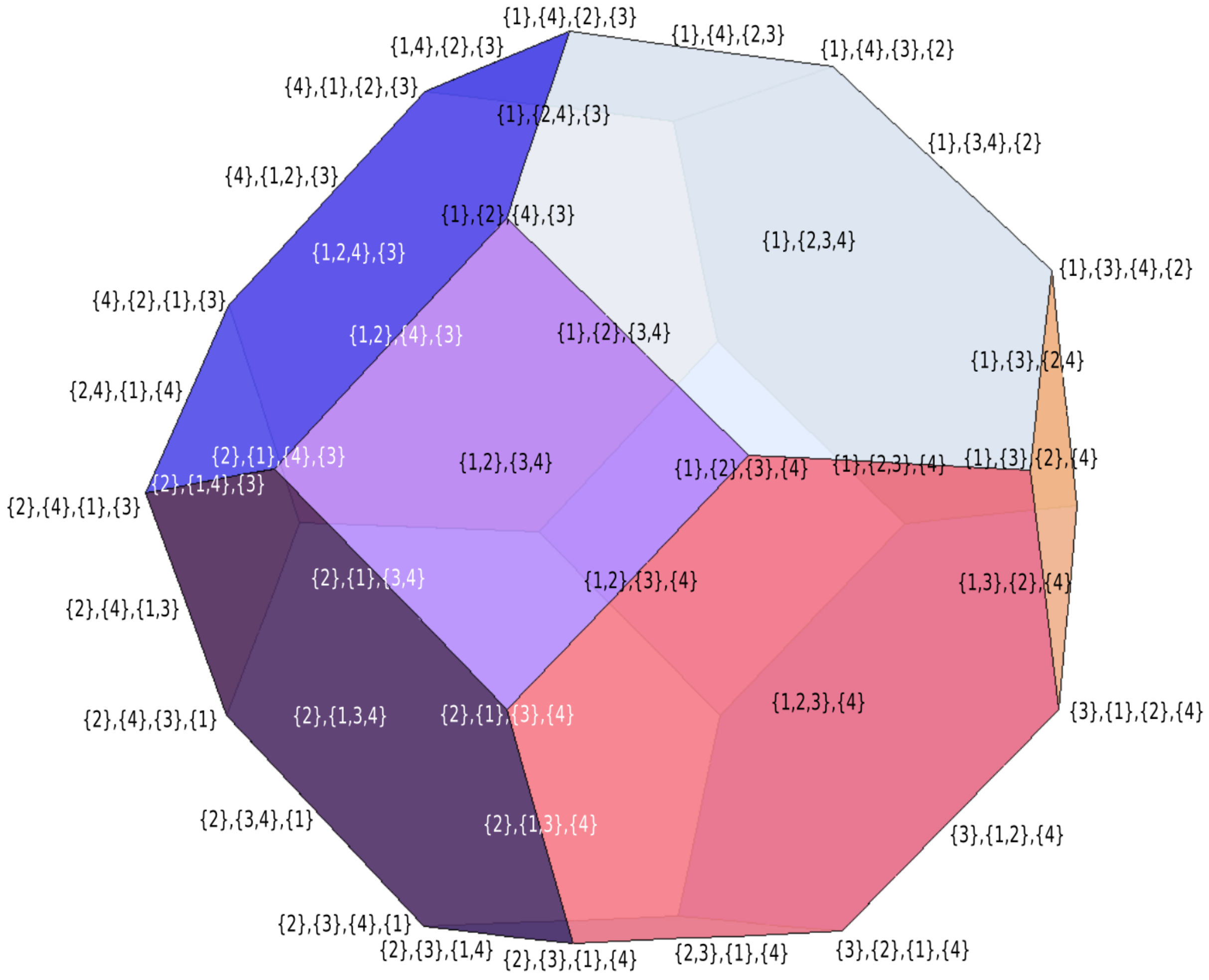


Figure 1.14: The three-dimensional permutahedron. The faces of the permutahedron are labeled by the corresponding ordered partitions of the set $\{1, 2, 3, 4\}$.

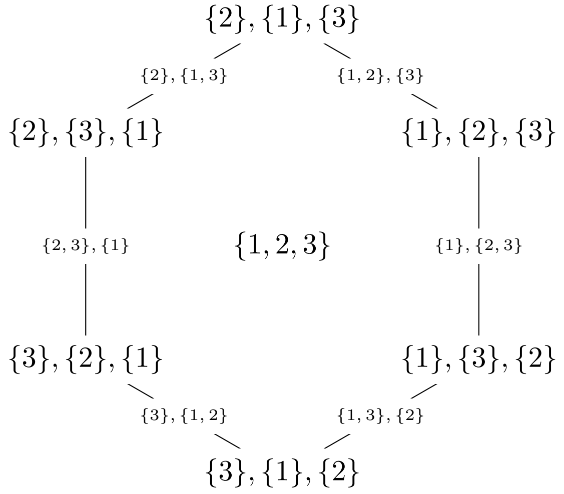


Figure 1.15: The ordered partitions of $\{1, 2, 3\}$ that correspond to the faces of a two-dimensional permutahedron. Note that each vertex has three parts, each edge has two parts and the hexagon itself has one part in their corresponding partitions.

Definition 1.3.3 (Partition). *Let T be a finite non-empty set and k be a positive integer that is at most the cardinality $|T|$. A* (non-ordered) partition *$\mathcal{P}$ of T in k parts is a collection of k subsets of T (called* parts*) that satisfy:*

- *all parts in the partition $\mathcal{P}$ are non-empty,*
- *all parts in the partition $\mathcal{P}$ are pairwise disjoint,*
- *the union of all parts in $\mathcal{P}$ is T.*

Definition 1.3.4 (Ordered partition). *Let T be a finite non-empty set and k be a positive integer that is at most the cardinality $|T|$. Denote by $\mathcal{P}(T)$ the collection of all subsets of T. An* ordered partition *of T in k parts is a function $\pi : \{1, \ldots, k\} \to \mathcal{P}(T)$ that maps indices $i \in \{1, \ldots, k\}$ to subsets $\pi(i) \subseteq T$ (called* parts*) such that $\{\pi(1), \ldots, \pi(k)\}$ is a partition of T in k parts.*

Let π be an ordered partition of a set T in k parts, for some $k \in \{1, \ldots, |T|\}$. We will follow a convention to write a sequence of subsets $\pi(1), \ldots, \pi(k)$ to represent the ordered partition π.

The way the ordered partitions of $\{1, \ldots, d + 1\}$ are associated to the faces of d-dimensional permutahedra is illustrated in Figure 1.14 and Figure 1.15. Note that the order in which the subsets are given matters, as it distinguishes different faces.

Now, we want to make a connection between the ordered partitions of $\{1, \ldots, d + 1\}$ and the partially-ordered set of faces of a d-dimensional permutahedron.

Definition 1.3.5 (Concatenation of ordered partitions). *Let T be a finite set and k be a positive integer that is less than the cardinality $|T|$. Let $T_1, \ldots, T_k$ be k mutually disjoint sets and T be the union $T_1 \cup \ldots \cup T_k$. Let $\rho_1, \ldots, \rho_k$ be ordered partitions of $T_1, \ldots, T_k$ respectively. Let $a_1, \ldots, a_k$ be the numbers of parts in $\rho_1, \ldots, \rho_k$ respectively. Define the integers:*

$$c_0 = 0, \quad c_1 = a_1, \quad c_2 = a_1 + a_2 \quad \ldots \quad c_k = \sum_{j=1}^{k} a_j.$$

The concatenation λ *of* $\rho_1, \ldots, \rho_k$ *is an ordered partition of the set* T*, such that for all* $i \in \{1, \ldots, |T|\}$*:*

$$\lambda(i) = \rho_j(i - c_j), \quad \textit{with } j \textit{ is such that } i \in \{c_j + 1, \ldots, c_{j+1}\}.$$

Remark 1.3.6. *Note that in Definition 1.3.5, we have* $c_k = |T|$*. Therefore, for any* $i \in \{1, \ldots, |T|\}$*, there exists an index* $j \in \{1, \ldots, k\}$ *such that* $i \in \{c_j + 1, \ldots, c_{j+1}\}$*.*

Definition 1.3.7 (Refinement)**.** *Let* π *and* ρ *be two ordered partitions of* $\{1, \ldots, d+1\}$ *in* k *parts and* l *parts respectively, with* $k, l \in \{1, \ldots, d+1\}$ *such that* $k \leq l$*. We say that* ρ *is a* refinement *of* π *(in* l *parts), if there exist positive integers* $a_1, \ldots, a_k$ *and ordered partitions* $\rho_1, \ldots, \rho_k$ *such that* ρ *is a concatenation of* $\rho_1, \ldots, \rho_k$ *and:*

$$\begin{gathered}
\rho_1 = \rho(1), \ldots, \rho(a_1) \textit{ is an ordered partition of } \pi(1) \textit{ in } a_1 \textit{ parts,} \\
\rho_2 = \rho(a_1 + 1), \ldots, \rho(a_1 + a_2) \textit{ is an ordered partition of } \pi(2) \textit{ in } a_2 \textit{ parts,} \\
\vdots \\
\rho_k = \rho(a_1 + \ldots + a_{k-1} + 1), \ldots, \rho(a_1 + \ldots + a_k) \textit{ is an ordered partition of } \pi(k) \textit{ in } a_k \textit{ parts.}
\end{gathered}$$

We denote the refinement relation by $\rho \sqsubseteq \pi$*.*

Definition 1.3.8 (Set of refinements)**.** *Let* π *be an ordered partition of* $\{1, \ldots, d+1\}$ *in* k *parts, for some* $k \in \{1, \ldots, d+1\}$*. Let* $l \in \{k, \ldots, d+1\}$ *be such that* $k \leq l$*. We denote by* $\mathcal{R}(\pi, l)$ *the set of refinements of* π *in* l *parts.*

Remark 1.3.9. *In terms of Definition 1.3.7, from* ρ *being the concatenation of* $\rho_1, \ldots, \rho_k$*, the number of parts in* ρ *is the sum of the number of parts in* $\rho_1, \ldots, \rho_k$*. It follows that the sum of* a_i *for all* $i \in \{1, \ldots, k\}$ *is* l*.*

For example, let $T = \{1, 2, 3, 4, 5\}$. If $\{1, 5\}, \{2, 3, 4\}$ is an ordered partition of T, then $\{1, 5\}, \{2\}, \{4\}, \{3\}$ is a refinement of $\{1, 5\}, \{2, 3, 4\}$. On the other hand, $\{2\}, \{1, 5\}, \{4\}, \{3\}$ is not a refinement of $\{1, 5\}, \{2, 3, 4\}$, despite being a (non-ordered) subpartition.

Definition 1.3.10. *We denote by* $OP(T)$ *the set of all ordered partitions of a finite set* T*.*

The refinement partial order $\sqsubseteq$ defines a partial order on the set of ordered partitions $OP(T)$ [MK92, Section 3.1.2]. Consequentially, $(OP(T), \sqsubseteq)$ is a partially-ordered set, called *refinement poset*.

Now, we are ready to state the main result of this section.

Theorem 1.3.11 ([MK92, Theorem 3.6])**.** *The partially-ordered set of faces of a* d*-dimensional permutahedron is isomorphic to the refinement poset* $(OP(T), \sqsubseteq)$*.*

Theorem 1.3.11 is illustrated in Figures 1.14 and 1.15. Note how the ordered partitions that correspond to subfaces are refinements.

We will now state the connection between the dimension of a cell and the number of parts in the corresponding partition.

Proposition 1.3.12 ([MK92, Corollary 3.12])**.** *Let* τ *be a* k*-dimensional face of a* d*-dimensional permutahedron, for some* $k \in \{0, \ldots, d\}$*. The ordered partition of* $\{1, \ldots, d+1\}$ *that is associated with the face* τ *via an isomorphism from Theorem 1.3.11 has* $d + 1 - k$ *parts.*

Remark 1.3.13. *If* σ *is an* m*-dimensional simplex in a Coxeter triangulation of type* $\tilde{A}_d$ *and* τ *is its* k*-dimensional dual cell in the Voronoi diagram, then we have* $m = d - k$*. It follows from Proposition 1.3.12 that the ordered partition that corresponds to the dual face* τ *has* $m + 1$ *parts. We will use this observation when we define a general representation of simplices of arbitrary dimensions in the Freudenthal-Kuhn triangulation of* $\mathbb{R}^d$ *in Section 3.3.*

1.3.2 Numbers of vertices and facets of a face of a permutahedron.

We now want to find an upper bound on the number of subfaces of a given dimension l of a k-dimensional face in a d-dimensional permutahedron. The number of l-dimensional faces of a d-dimensional permutahedron is well-known [Zie12, Chapter 0]:

$$(d-l)!\ S(d, d-l),$$

where $S(d, d-k)$ is the Stirling number of the second kind (introduced by Stirling in [Sti30]). However, we did not find in the literature the number of l-dimensional subfaces of a k-dimensional face of a d-dimensional permutahedron. Perhaps, the reason for this is that this number is not uniquely defined[10] for given l, k and d. Here, we will partially fill this knowledge gap by expressing *upper bounds* on the number of two particular kinds of faces: vertices ($l = 0$) and facets ($l = k-1$). These upper bounds are used later in Chapter 4 to compute the complexity of the manifold tracing algorithm. We also provide the *lower bound* on the number of vertices of a k-dimensional face of a d-dimensional permutahedron that we will use in the proof of Proposition 4.2.8.

Number of vertices of a face in a permutahedron. First, let us start with counting the vertices of a face of a d-dimensional permutahedron.

Proposition 1.3.14 (Number of vertices)**.** *Let τ be a k-dimensional face of a d-dimensional permutahedron, for some $k \in \{0, \ldots, d\}$. Let π be the ordered partition of $\{1, \ldots, d+1\}$ in $d+1-k$ parts that is associated to τ via an isomorphism from Theorem 1.3.11. Denote by $W(\tau)$ the number of vertices of τ. Then:*

$$W(\tau) = \prod_{i=1}^{d+1-k} |\pi(i)|!\ .$$

Proof. By Theorem 1.3.11, each vertex of τ is in one-to-one correspondence with a refinement of π in $d+1$ parts. Therefore, the number of vertices of τ is the cardinality of the set $\mathcal{R}(\pi, d+1)$. To find the cardinality $|\mathcal{R}(\pi, d+1)|$, we will define a bijection $\chi : \mathfrak{S}_{|\pi(1)|} \times \ldots \times \mathfrak{S}_{|\pi(d+1-k)|} \to \mathcal{R}(\pi, d+1)$. The cardinality of $\mathfrak{S}_{|\pi(1)|} \times \ldots \times \mathfrak{S}_{|\pi(d+1-k)|}$ is $\prod_{i=1}^{d+1-k} |\pi(i)|!$. Hence, if such a bijection χ indeed exists, then the result of the proposition follows.

Let $\sigma = (\sigma_1, \ldots, \sigma_{d+1-k}) \in \mathfrak{S}_{|\pi(1)|} \times \ldots \times \mathfrak{S}_{|\pi(d+1-k)|}$. For all $i \in \{1, \ldots, d+1-k\}$, we define ρ_i to be an ordered partition of $\pi(i)$ in $|\pi(i)|$ singletons in the following way. Write the elements of $\pi(i)$ as $\pi(i) = \{p_1, \ldots, p_{|\pi(i)|}\}$ such that $p_1 < \ldots < p_{|\pi(i)|}$. For all $j \in \{1, \ldots, |\pi(i)|\}$, we pose:

$$\rho_i(j) = \{p_{\sigma_i(j)}\}.$$

The concatenation of $\rho_1, \ldots, \rho_{d+1-k}$ is an ordered partition of π in $d+1$ singletons. We define $\chi(\sigma)$ to be this concatenation.

Now we will define the inverse function χ^{-1}. Let ρ be an element of $\mathcal{R}(\pi, d+1)$. Because ρ is a refinement of π in $d+1$ singletons, by Definition 1.3.7 there exist ordered partitions $\rho_1, \ldots, \rho_k$ such that ρ is a concatenation of $\rho_1, \ldots, \rho_k$ and for all $i \in \{1, \ldots, d+1-k\}$:

$$\rho_i = \rho\left(\sum_{j=1}^{i-1} |\pi(j)| + 1\right), \ldots, \rho\left(\sum_{j=1}^{i} |\pi(j)|\right)$$

[10] For example, the two-dimensional faces in the two-dimensional permutahedron in Figure 1.14 have either 4 or 6 vertices.

is an ordered partition of $\pi(i)$ in $|\pi(i)|$ singletons. We will define for all $i \in \{1, \ldots, d+1-k\}$, a permutation $\sigma_i \in \mathfrak{S}_{|\pi(i)|}$ in the following way. Fix $i \in \{1, \ldots, d+1-k\}$. Write the elements of $\pi(i)$ as $\pi(i) = \{p_1, \ldots, p_{|\pi(i)|}\}$ such that $p_1 < \ldots < p_{|\pi(i)|}$. For all $j \in \{1, \ldots, |\pi(i)|\}$, we pose $\sigma_i(j)$ to be such that the sole element in the singleton $\rho_i(j)$ is $p_{\sigma_i(j)}$. We pose $\chi^{-1}(\rho)$ to be the tuple $(\sigma_1, \ldots, \sigma_{d+1-k}) \in \mathfrak{S}_{|\pi(1)|} \times \ldots \times \mathfrak{S}_{|\pi(d+1-k)|}$.

From the definitions of the functions χ and χ^{-1} it follows that:

1. for any $\rho \in \mathcal{R}(\pi, d+1)$, we have $\chi(\chi^{-1}(\rho)) = \rho$,

2. for any $\sigma \in \mathfrak{S}_{|\pi(1)|} \times \ldots \times \mathfrak{S}_{|\pi(d+1-k)|}$, we have $\chi^{-1}(\chi(\sigma)) = \sigma$.

Thus, the two functions χ and χ^{-1} are inverses of each another. It follows that χ is a bijection from $\mathfrak{S}_{|\pi(1)|} \times \ldots \times \mathfrak{S}_{|\pi(d+1-k)|}$ to $\mathcal{R}(\pi, d+1)$, as desired. Thus, the cardinalities of these two sets are the same, equal to:

$$\prod_{i=1}^{d+1-k} |\pi(i)|! \,.$$

Since $W(\tau) = |\mathcal{R}(\pi, d+1)|$, the result follows. □

The following corollary expresses the number of vertices in the big O notation.

Corollary 1.3.15 (Upper bound on the number of vertices). *Let $k \in \{0, \ldots, d\}$. The number of vertices of a k-dimensional face of a d-dimensional permutahedron is:*

$$O((k+1)!).$$

Note that the bound $(k+1)!$ is tight for the k-dimensional faces of a d-dimensional permutahedra which are themselves permutahedra.

Proof. Let τ be a k-dimensional face of a d-dimensional permutahedron.[11] Let π be the ordered partition of $\{1, \ldots, d+1\}$ in $d+1-k$ parts that is associated to τ *via* an isomorphism from Theorem 1.3.11. According to Proposition 1.3.14, the number of vertices of τ is:

$$W(\tau) = \prod_{i=1}^{d+1-k} |\pi(i)|! \,.$$

Denote for all $i \in \{1, \ldots, d+1-k\}$, the cardinality $|\pi(i)|$ by b_i. Recall that for all $i \in \{1, \ldots, d+1-k\}$, we have $b_i \geq 1$ and the sum of all b_i is $d+1$. Therefore, there is only finitely many possibilities for the values of the b_i. It follows that there exists at least one configuration of the b_i that maximizes the product of factorials.

Without loss of generality, assume that $b_1 \geq b_2 \geq \ldots \geq b_{d+1-k}$. We want to show that this maximum of the product of factorials of the b_i is achieved when $b_1 = k+1$ and all other b_i are equal to 1.

Assume that there exists $j \in \{2, \ldots, d+1-k\}$ such that $b_j > 1$. Define the sequence of numbers b'_i:

$$\begin{aligned} b'_1 &= b_1 + 1 \\ b'_j &= b_j - 1 \\ b'_i &= b_i, \quad \text{for all } i \neq j. \end{aligned}$$

[11] The proof is taken from this answer on math.stackexchange.com.

By definition, for all $i \in \{1, \ldots, d+1-k\}$, we have $b'_i \geq 1$ and the sum of all b'_i is $d+1$. On the other hand, when we compute the ratio between the products of factorials of the b'_i and of the factorials of b_i, we find:

$$\frac{\prod_{i=1}^{d+1-k} b'_i!}{\prod_{i=1}^{d+1-k} b_i!} = \frac{((b_1+1)!)(b_2!)\ldots(b_{j-1}!)((b_j-1)!)(b_{j+1}!)\ldots(b_{d+1-k}!)}{(b_1!)\ldots(b_{d+1-k}!)} = \frac{b_1+1}{b_j} > 1.$$

Thus, whenever there exists $j \in \{2, \ldots, d+1-k\}$ such that $b_j > 1$, there also exists a configuration of the b_i which yields a greater product of factorials. We conclude that the maximum of the product of factorials of the b_i is achieved when $b_1 = k+1$ and all other b_i are equal to 1. Thus, we have:

$$W(\tau) = \prod_{i=1}^{d+1-k} |\pi(i)|! \leq (k+1)!\ .$$

We conclude that $W(\tau) = O((k+1)!)$, as desired.

□

We can use the same proof for a configuration of the b_i where each b_i is either $\lfloor \frac{d}{d+1-k} \rfloor$ or $\lceil \frac{d}{d+1-k} \rceil$. This way, we also get the lower bound on the number of vertices of a k-dimensional cell of a d-dimensional permutahedron.

Corollary 1.3.16 (Lower bound on the number of vertices)**.** *Let $k \in \{0, \ldots, d\}$. The number of vertices of a k-dimensional face of a d-dimensional permutahedron is:*

$$\Omega\left(\left(\frac{d}{d+1-k}\right)^{d+1-k}\right).$$

We will use this lower bound later in the proof of Proposition 4.2.8.

Number of facets of a face of a permutahedron. Now we will state a result for facets, which is analogous to Proposition 1.3.14.

Proposition 1.3.17 (Number of facets)**.** *Let τ be a k-dimensional face of a d-dimensional permutahedron, for some $k \in \{1, \ldots, d\}$. Let π be the ordered partition of $\{1, \ldots, d+1\}$ in $d+1-k$ parts that is associated to τ via an isomorphism from Theorem 1.3.11. Denote by $Y(\tau)$ the number of facets of τ. Then:*

$$Y(\tau) = \sum_{i=1}^{d+1-k} \left(2^{|\pi(i)|} - 2\right).$$

Proof. By Theorem 1.3.11, each facet of τ is in one-to-one correspondence with a refinement of π in $d-k+2$ parts. Therefore, the number of facets of τ is the cardinality of the set $\mathcal{R}(\pi, d-k+2)$. To find the cardinality $|\mathcal{R}(\pi, d-k+2)|$, we will define a bijection χ from the set $\mathcal{R}(\pi, d-k+1)$ to a set $\mathcal{X}$ defined in the following way. Each element of $\mathcal{X}$ is a pair that consists of an index $j \in \{1, \ldots, d+1-k\}$ and a *non-constant* map $\pi(j) \to \{0,1\}$. For a fixed j, there are $2^{|\pi(i)|} - 2$ such maps. It follows that the cardinality $|\mathcal{X}|$ is $\sum_{i=1}^{d+1-k} \left(2^{|\pi(i)|} - 2\right)$. Hence, if a bijection $\chi : \mathcal{R}(\pi, d-k+2) \to \mathcal{X}$ exists, then the result of the proposition follows.

Let ρ be an element of $\mathcal{R}(\pi, d-k+2)$. Since π consists of $d-k+1$ parts and ρ is a refinement of π in $d-k+2$ parts, there exists $j \in \{1, \ldots, d-k+1\}$, such that:

$$\begin{aligned} \pi(i) &= \rho(i), \quad \text{for all } i < j, \\ \pi(j) &= \rho(j) \cup \rho(j+1), \\ \pi(i) &= \rho(i+1), \quad \text{for all } i > j. \end{aligned}$$

Define a map $\mu : \pi(j) \to \{0, 1\}$ such that for all $l \in \pi(j)$:

$$\mu(l) = \begin{cases} 0 & \text{if } l \in \rho(j), \\ 1 & \text{if } l \in \rho(j+1). \end{cases}$$

Neither of the sets $\rho(j)$ and $\rho(j+1)$ is empty, hence the map μ is not constant. The pair (j, μ) therefore belongs to the set $\mathcal{X}$. We define χ to be this pair.

Now we will define the inverse function χ^{-1}. Let $\sigma = (j, \mu) \in \mathcal{X}$ be a pair of an index j and a map $\mu : \pi(j) \to \{0, 1\}$. Define an ordered partition ρ of $\{1, \ldots, d+1\}$ in $d-k+2$ parts defined as follows:

$$\begin{aligned} \rho(i) &= \pi(i), \quad \text{for all } i < j, \\ \rho(j) &= \{l \in \pi(j) \mid \mu(l) = 0\}, \\ \rho(j+1) &= \{l \in \pi(j) \mid \mu(l) = 1\}, \\ \rho(i) &= \pi(i-1), \quad \text{for all } i > j+1. \end{aligned}$$

Because μ is not constant, no part in ρ is empty. The ordered partition ρ is hence a refinement of π in $d-k+2$ parts. We pose $\chi^{-1}(\sigma) = \rho$.

From the definitions of the functions χ and χ^{-1} it follows that:

1. for any $\rho \in \mathcal{R}(\pi, d-k+2)$, we have $\chi^{-1}(\chi(\rho)) = \rho$.

2. for any $\sigma \in \mathcal{X}$, we have $\chi(\chi^{-1}(\sigma)) = \sigma$,

Thus, the two functions χ and χ^{-1} are inverses of each another. It follows that χ is a bijection from $\mathcal{R}(\pi, d-k+2)$ to $\mathcal{X}$, as desired. Thus, the cardinalities of these two sets are the same, equal to:

$$\sum_{i=1}^{d+1-k} \left(2^{|\pi(i)|} - 2\right).$$

Since $Y(\tau) = |\mathcal{R}(\pi, d-k+2)|$, the result follows. □

The following corollary expresses the number of facets in the big O notation.

Corollary 1.3.18 (Upper bound on the number of facets). *Let $k \in \{1, \ldots, d\}$. The number of facets of a k-dimensional face of a d-dimensional permutahedron is:*

$$O\left(2^k\right).$$

Proof. The proof follows similar arguments as in the proof of Corollary 1.3.15. Let τ be a k-dimensional face of a d-dimensional permutahedron for some $k \in \{1, \dots, d\}$. Let π be the ordered partition of $\{1, \dots, d+1\}$ in $d+1-k$ parts that is associated to τ *via* an isomorphism from Theorem 1.3.11. According to Proposition 1.3.17, the number of vertices of τ is:

$$Y(\tau) = \sum_{i=1}^{d+1-k} \left(2^{|\pi(i)|} - 2\right) \leq \sum_{i=1}^{d+1-k} 2^{|\pi(i)|}.$$

Denote for all $i \in \{1, \dots, d+1-k\}$, the cardinality $|\pi(i)|$ by b_i. Recall that for all $i \in \{1, \dots, d+1-k\}$, we have $b_i \geq 1$ and the sum of all b_i is $d+1$. Therefore, there is only finitely many possibilities for the values of the b_i. It follows that there exists at least one configuration of the b_i that maximizes the sum of the 2^{b_i}.

Without loss of generality, assume that $b_1 \geq b_2 \geq \ldots \geq b_{d+1-k}$. We want to show that this maximum of the sum of the 2^{b_i} is achieved when $b_1 = k+1$ and all other b_i are equal to 1.

For this, assume that there exists $j \in \{2, \dots, d+1-k\}$ such that $b_j > 1$. Define the sequence of numbers b'_i:

$$\begin{aligned} b'_1 &= b_1 + 1 \\ b'_j &= b_j - 1 \\ b'_i &= b_i, \quad \text{for all } i \neq j. \end{aligned}$$

By definition, for all $i \in \{1, \dots, d+1-k\}$, we have $b'_i \geq 1$ and the sum of all b'_i is $d+1$. On the other hand, when we compute the difference between the sum of the $2^{b'_i}$ and the sum of the 2^{b_i}, we find:

$$\begin{aligned} \sum_{i=1}^{d+1-k} 2^{b'_i} - \sum_{i=1}^{d+1-k} 2^{b_i} &= \left(2^{b_1+1} + 2^{b_2} + \ldots + 2^{b_{j-1}} + 2^{b_j-1} + 2^{b_{j+1}} + \ldots + 2^{b_{d+1-k}}\right) \\ &\quad - \left(2^{b_1} + \ldots + 2^{b_{d+1-k}}\right) \\ &= \left(2^{b_1+1} - 2^{b_1}\right) + \left(2^{b_j-1} - 2^{b_j}\right) = 2^{b_1} - 2^{b_j-1} > 0. \end{aligned}$$

Thus, whenever there exists $j \in \{2, \dots, d+1-k\}$ such that $b_j > 1$, there also exists a configuration of the b_i which yields a greater sum of the 2^{b_i}. We conclude that the maximum of the sum of the 2^{b_i} is achieved when $b_1 = k+1$ and all other b_i are equal to 1. Thus, we have:

$$Y(\tau) \leq \sum_{i=1}^{d+1-k} 2^{|\pi(i)|} \leq 2^{k+1}.$$

We conclude that $Y(\tau) = O(2^k)$, as desired.

□

Remark 1.3.19. *Note that the bound in Corollary 1.3.18 is tight for the following reason. For any $k, d \in \mathbb{Z}$ such that $1 \leq k \leq d$, a d-dimensional permutahedron has k-dimensional faces that are themselves k-dimensional permutahedra. Any of these permutahedral k-dimensional faces have $2^k - 2$ facets.*

Number of full-dimensional cofaces and cofacets of simplices in a Coxeter triangulation. By duality, we can restate Corollaries 1.3.15, 1.3.18 and 1.3.16 in terms of the number of full-dimension cofaces and cofacets of full-dimension cofaces of a given simplex in a Coxeter triangulation of type $\tilde{A}_d$.

Corollary 1.3.20. *Let τ be a simplex of dimension $k \in \{0, \dots, d-1\}$ in a Coxeter triangulation of type $\tilde{A}_d$. Then:*

1. *The number of full-dimensional cofaces of τ is $O((d-k+1)!)$ and $\Omega\left(\left(\frac{d}{d-k+1}\right)^{d-k+1}\right)$.*

2. *The number of cofacets of τ is $O(2^{d-k})$.*

Chapter 2

Quality of Coxeter triangulations

Well-shaped simplices are of importance for various fields of application such as finite element methods and manifold meshing [Syn57, BA76, Jam76, Kří92, CDR05, CDS12]. Poorly-shaped simplices may induce various problems in finite element methods, such as large discretization errors or ill-conditioned stiffness matrices. Determining whether a simplex is well-shaped or not can be done with the help of various *quality measures*. Some examples of quality measures are: the ratio between minimal height and maximal edge length ratio called *thickness* [Mun66, Vav96], the ratio between volume and a power of the maximal edge length called *fatness* [Whi57], and the *inradius-circumradius ratio* [CFF85]. Bounds on *dihedral angles* can also be included in the list of quality measures. We stress that there are many other quality measures in use and authors often find useful to introduce measures that are specific to whatever problem they study (see for example [She02] for an overview on quality measures). Finding triangulations, even in Euclidean space, of which all simplices have good quality is a non-trivial exercise in arbitrary dimension.

In this chapter, we shall discuss the quality of Coxeter triangulations. To our knowledge, Coxeter triangulations are the triangulations with the best quality in arbitrary dimension. In particular, all dihedral angles of simplices in Coxeter triangulations are 45°, 60° or 90°, with the exception of the $\tilde{G}_2$ triangulation of the plane where we also can find an angle of 30°. This is a clear sign of the exceptional quality of the simplices involved. Our goal in this chapter is to exhibit the extraordinary properties of Coxeter triangulations.

The self-similarity and symmetry are another of the attractive points of Coxeter triangulations. Upon the first introduction, an unprepared reader might expect that the vertex sets of such triangulations form lattices, however it is not necessarily true, as we show in Section 2.2.

We are also interested in the stronger requirement of *protection* [BDG13], which is specific to Delaunay triangulations. For a point set P to have a unique well-defined Delaunay triangulation in the Euclidean space $\mathbb{R}^d$, one demands that there is no ball with at least $d + 2$ points in P on the boundary and no point in P in the interior. Protection requires not only that there is no other vertex on the (empty) circumsphere of $d + 1$ vertices, but that any other vertex is at distance δ away from the circumsphere. It has been proven that protection guarantees good quality [BDG13]. Some algorithms were introduced for the construction of a protected set, such as the perturbation-based algorithms in [BDG14] and [BDG15]. Both of these algorithms take a general ε-net in $\mathbb{R}^d$ as input and output a δ-protected net with δ of the order just $\Omega(2^{-d^2}\varepsilon)$. On the other hand, take the d-dimensional Coxeter triangulation $\tilde{A}_d$. As we will see in the following, this highly-structured triangulation is Delaunay with protection $O(\frac{1}{d^2}\varepsilon)$. This protection value is the greatest in a general d-dimensional Delaunay triangulation we know.

We start the chapter by defining the quality measures in Section 2.1. We also state the theorem of optimality of the regular d-simplex for each of the chosen quality measures. This theorem justifies the definition of the normalized versions of these quality measures.

In Section 2.2, we state Theorem 2.2.1, which is the main result of the section. In the theorem, we give explicit expressions of a number of quality measures of Coxeter triangulations for all dimensions. This result is an extension of the work by Dobkin *et al.* [DWLT90] who presented a table of the values of the inradius-circumradius ratio for the Coxeter triangulations up to dimension 8.

We provide the proof of Theorem 2.2.1 in Sections 2.4 and 2.5. The proof is organized as a case study for each family of Coxeter triangulations with the same plan for each case. We provide explicit measures of the corresponding simplices in Section 2.4, allowing the reader to compute quality measures other than the ones listed. Determining whether a given triangulation is Delaunay or not is done using the criterion that we establish in Section 2.3.

We finish the chapter by providing tables with numerical values of simplex quality in Section 2.6, similar to [DWLT90].

2.1 Quality definitions

The quality measures we are interested in are *aspect ratio*, *fatness*, *thickness* and *radius ratio*. Their formal definitions are as follows:

Definition 2.1.1. *Let $h(\sigma)$ denote the minimal height, $r(\sigma)$ the inradius, $R(\sigma)$ the circumradius, $vol(\sigma)$ the volume and $L(\sigma)$ the maximal edge length of a given d-simplex σ.*

- *The* aspect ratio *of σ is the ratio of its minimal height to the diameter of its circumscribed ball: $\alpha(\sigma) = \frac{h(\sigma)}{2R(\sigma)}$.*
- *The* fatness *of σ is the ratio of its volume to its maximal edge length taken to the power d: $\Theta(\sigma) = \frac{vol(\sigma)}{L(\sigma)^d}$.*
- *The* thickness *of σ is the ratio of its minimal height to its maximal edge length: $\theta(\sigma) = \frac{h(\sigma)}{L(\sigma)}$.*
- *The* radius ratio *of σ is the ratio of its inradius to its circumradius: $\rho(\sigma) = \frac{r(\sigma)}{R(\sigma)}$.*

To be able to compare the presented quality measures between themselves, we will normalize them by their respective maximum value. As we show in Theorem 2.1.2, all of these quality measures are maximized by regular simplices.

Theorem 2.1.2. *Out of all d-dimensional simplices, the regular d-simplex has the highest aspect ratio, fatness, thickness and radius ratio.*

We prove Theorem 2.1.2 at the end of the current section. For a quality measure κ we will define the normalized quality measure $\hat{\kappa}$, such that for each d-simplex σ, $\hat{\kappa}(\sigma) = \frac{\kappa(\sigma)}{\kappa(\Delta)}$, where Δ is the regular d-simplex. The continuity of the quality measures $\hat{\rho}$, $\hat{\alpha}$, $\hat{\theta}$ and $\hat{\Theta}$ and Theorem 2.1.2 ensure that these quality measures take their values in $[0, 1]$ surjectively.

Because some of the triangulations that interest us here are Delaunay, we will also look at their *protection* values.

Definition 2.1.3. *Let* $\mathrm{vert}(\sigma)$ *denote the set of vertices of a simplex σ. The* protection *of a d-simplex σ in a Delaunay triangulation on a point set P is the minimal distance of points in $P \setminus \mathrm{vert}(\sigma)$ to the circumscribed ball of σ:*

$$\delta(\sigma) = \inf_{p \in P \setminus \mathrm{vert}(\sigma)} d(p, B(\sigma)), \textit{ where } B(\sigma) \textit{ is the circumscribed ball of } \sigma.$$

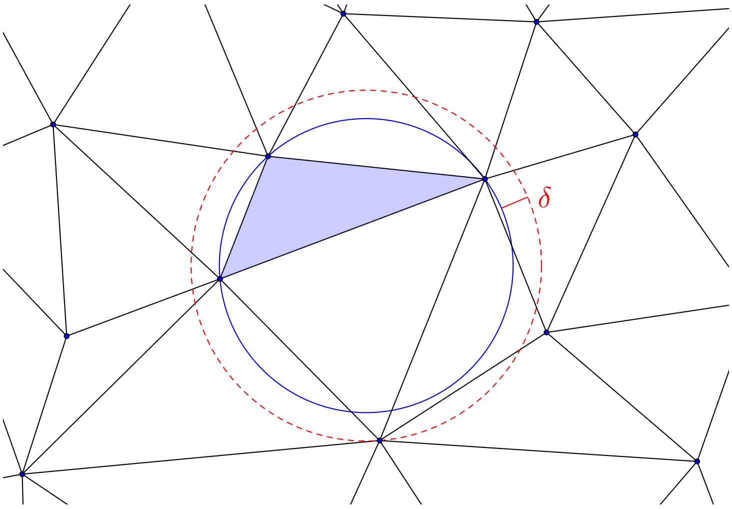


Figure 2.1: Illustration of the protection δ of a d-simplex in a Delaunay triangulation.

The protection δ *of a Delaunay triangulation* $\mathcal{T}$ *is the infimum over the d-simplices of the triangulation:* $\delta = \inf_{\sigma \in \mathcal{T}} \delta(\sigma)$*. A triangulation with a positive protection is called* protected*.*

We define the relative protection $\hat{\delta}(\sigma)$ *of a given d-simplex* σ *to be the ratio of the protection to its circumscribed radius:* $\hat{\delta}(\sigma) = \frac{\delta(\sigma)}{R(\sigma)}$*.*

The relative protection $\hat{\delta}$ *of a Delaunay triangulation* $\mathcal{T}$ *is the infimum over the d-simplices of the triangulation:* $\hat{\delta} = \inf_{\sigma \in \mathcal{T}} \hat{\delta}(\sigma)$*.*

The protection of a simplex is illustrated in Figure 2.1.

Proof of Theorem 2.1.2 In this section we will prove Theorem 2.1.2. This section is added for completeness and can be skipped on the first reading. The proof will be subdivided into lemmas that are dedicated to each individual quality measure.

The first result for radius ratio is due to Klamkin [Kla85]:

Lemma 2.1.4 (Maximum radius ratio [Kla85])**.** *Out of all d-simplices, the regular d-simplex* Δ *has the maximum radius ratio equal to* $\rho(\Delta) = \frac{1}{d}$*.*

We can adapt the proof by Klamkin to show the result for aspect ratio:

Lemma 2.1.5 (Maximum aspect ratio)**.** *Out of all d-simplices, the regular d-simplex* Δ *has the maximum aspect ratio equal to* $\alpha(\Delta) = \frac{d+1}{2d}$*.*

Proof. Let σ be a d-simplex. We will now prove that $\alpha(\sigma) = \frac{h(\sigma)}{2R(\sigma)} \leq \frac{d+1}{2d}$.

Let F_i be the $(d-1)$-dimensional volume of the ith facet (see Figure 2.2). Denote v_i the opposite vertex.

If p is an arbitrary point, we have by Cauchy-Schwarz inequality:

$$\sum_{i=0}^{d} F_i \cdot \sum_{i=0}^{d} F_i \overrightarrow{pv_i}^2 \geq \left(\sum_{i=0}^{d} F_i \overrightarrow{pv_i} \right)^2 . \tag{2.1}$$

Note that the equality takes place if and only if p is the circumcentre (making all $\overrightarrow{pv_i}^2$ equal).

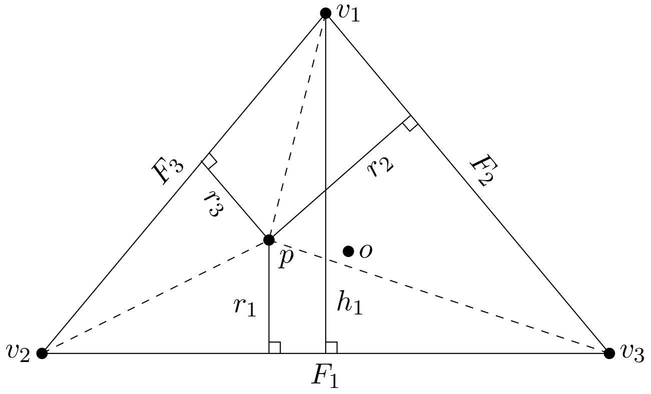


Figure 2.2: Construction for the proof of Lemma 2.1.5.

Let $F = \sum_{i=0}^{d} F_i$. Let $o \in \mathbb{R}^d$ be the circumcentre of σ and $q \in \mathbb{R}^d$ be such that $\overrightarrow{oq} = \sum_{i=0}^{d} F_i \overrightarrow{ov_i}/F$. Then:

$$\sum_{i=0}^{d} F_i \overrightarrow{pv_i}^2 \quad = \quad \sum_{i=0}^{d} F_i \left(\overrightarrow{op} - \overrightarrow{ov_i}\right)\left(\overrightarrow{op} - \overrightarrow{ov_i}\right) = F\left(R(\sigma)^2 + \overrightarrow{op}^2 - 2\overrightarrow{op}\cdot\overrightarrow{oq}\right).$$

Since $2\overrightarrow{op}\cdot\overrightarrow{oq} = \overrightarrow{op}^2 + \overrightarrow{oq}^2 - \left(\overrightarrow{oq} - \overrightarrow{op}\right)^2 = \overrightarrow{op}^2 + \overrightarrow{oq}^2 - \overrightarrow{pq}^2$, we have:

$$\sum_{i=0}^{d} F_i \cdot \sum_{i=0}^{d} F_i \overrightarrow{pv_i} = F^2\left(R(\sigma)^2 + \overrightarrow{pq}^2 - \overrightarrow{oq}^2\right). \tag{2.2}$$

Denote by h_i and r_i the distances from v_i and p respectively to the affine hull of the ith facet. Then $d(p, v_i)$ is greater than $h_i - r_i$ as the hypotenuse is greater than a leg in a right triangle (see Figure 2.2). So:

$$\sum_{i=0}^{d} F_i\ d(p, v_i) \geq \sum_{i=0}^{d} F_i h_i - \sum_{i=0}^{d} F_i r_i.$$

Note that if V denotes the volume of the simplex, then for a fixed i we have $F_i h_i = dV$.

On the other hand, for all i, the d-simplex formed by p and F_i has a height r_i with respect to F_i (see Figure 2.2). It yields that $\sum_{i=0}^{d} F_i r_i = dV$. Therefore:

$$\sum_{i=0}^{d} F_i\ d(p, v_i) \geq (d+1)dV - dV = d^2 V. \tag{2.3}$$

As before, let $h(\sigma) = \min_i h_i$. Now observe that:

$$(d+1)dV = \sum_{i=0}^{d} F_i h_i \geq \sum_{i=0}^{d} F_i h(\sigma) = F h(\sigma). \tag{2.4}$$

Hence, if we combine the equations (2.3) and (2.4):

$$\sum_{i=0}^{d} F_i\ d(p, v_i) \geq \frac{dFh(\sigma)}{d+1}. \tag{2.5}$$

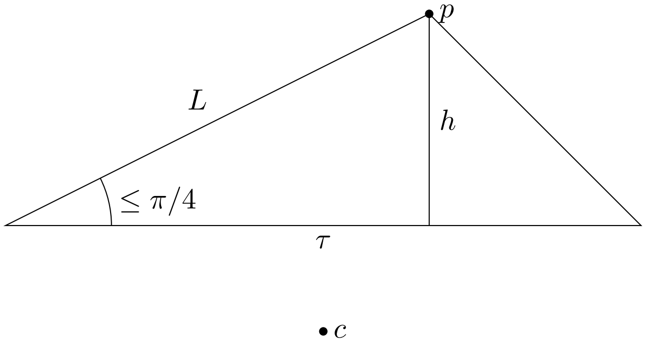


Figure 2.3: Illustration for the proof of Lemma 2.1.8. The plane of the figure contains the edge of length L and the height h. The intersection of this plane and the simplex is an obtuse triangle, and the angle adjacent to the longest edge is less than $\pi/4$.

Thus combining (2.1), (2.2) and (2.5), we get:

$$F^2\left(R(\sigma)^2 + \overrightarrow{pq}^2 - \overrightarrow{oq}^2\right) = \sum_{i=0}^{d} F_i \sum_{i=0}^{d} F_i \overrightarrow{pv_i} \geq \left(\sum_{i=0}^{d} F_i \overrightarrow{pv_i}\right)^2 \geq \left(\frac{dFh(\sigma)}{d+1}\right)^2.$$

Setting p equal to q removes the $\overrightarrow{pq}$ term:

$$R(\sigma)^2 \geq \left(\frac{d}{d+1}h(\sigma)\right)^2 + \overrightarrow{oq}^2 \geq \left(\frac{d}{d+1}h(\sigma)\right)^2,$$

which gives the inequality that we sought to prove.

By a simple computation, we can further establish that the aspect ratio of the regular simplex is $\frac{d}{d+1}$, and thus we conclude the proof. □

To prove the result for thickness, we will use Jung's theorem[Jun01].

Definition 2.1.6. *The* minimal enclosing ball *of a bounded subset S of $\mathbb{R}^d$ is the Euclidean ball of the smallest radius that contains S.*

Theorem 2.1.7 (Jung's theorem)**.** *Let $S \subseteq \mathbb{R}^d$ be a compact set. Let $L(S)$ be its diameter and $R_{enc}(S)$ be the radius of the minimum enclosing ball. Then:*

$$R_{enc}(S) \leq L(S)\sqrt{\frac{d}{2(d+1)}}.$$

We can combine Lemma 2.1.5 with Jung's theorem[Jun01] to prove the result for thickness:

Lemma 2.1.8 (Maximum thickness)**.** *Out of all d-simplices, the regular d-simplex Δ has the maximum thickness equal to $\theta(\Delta) = \sqrt{\frac{d+1}{2d}}$.*

Proof. Assume that σ is a d-simplex. We will distinguish two cases: namely σ is self-centred or not (as in Definition 2.3.1).

Let σ be not self-centred. Let τ be a facet of σ that separates σ from its circumcentre c. Let p be the vertex of σ opposite to τ. Denote by h the height from p to τ and L be the length of the longest edge e adjacent to p (see Figure 2.3). By construction, $h \leq L/\sqrt{2}$, because the angle of the edge e with τ is less or equal to $\pi/4$. Therefore:

$$\theta(\sigma) = \frac{h(\sigma)}{L(\sigma)} \leq \frac{h}{L} \leq \frac{1}{\sqrt{2}} \leq \sqrt{\frac{d+1}{2d}}.$$

From now on we assume that σ is self-centred. This implies in particular that the circumradius of σ is equal to the radius of the minimal enclosing ball of σ. So, by combining Lemma 2.1.5 and Jung's theorem we get:

$$\theta(\sigma) = \frac{h(\sigma)}{L(\sigma)} = \frac{h(\sigma)}{2R(\sigma)} \frac{2R(\sigma)}{L(\sigma)} = \frac{h(\sigma)}{2R(\sigma)} \frac{2R_{enc}(\sigma)}{L(\sigma)} \leq \frac{d+1}{2d} \cdot 2\sqrt{\frac{d}{2(d+1)}} = \sqrt{\frac{d+1}{2d}}.$$

Jung's theorem becomes equality in the case of the regular d-simplex [DGK63], so its thickness is $\theta(\Delta) = \sqrt{\frac{d+1}{2d}}$. □

Finally, we prove the result for fatness.

Lemma 2.1.9 (Maximum fatness)**.** *Out of all d-simplices, the regular d-simplex Δ has the maximum fatness equal to $\Theta(\Delta) = \frac{1}{d!}\sqrt{\frac{d+1}{2^d}}$.*

Proof. Let σ_d be a d-simplex. We define a sequence of faces $\sigma_1 \subset ... \subset \sigma_d$ of σ_d, such that for each $i \in \{1, ..., d-1\}$, σ_i has dimension i and the height falling on σ_i is the minimum height in σ_{i+1}.

We can now express the volume of σ in the following manner:

$$V(\sigma_d) = \frac{1}{d!} \prod_{i=1}^{d} h(\sigma_i).$$

Observe that for each i, the set of edges of σ_d is a subset of the set of edges of σ, therefore we have $L(\sigma_i) \leq L(\sigma_d)$. By combining this fact with the volume expression we get:

$$\Theta(\sigma_d) = \frac{V(\sigma_d)}{L(\sigma_d)^d} = \frac{1}{d!} \prod_{i=1}^{d} \frac{h(\sigma_i)}{L(\sigma_d)} \leq \frac{1}{d!} \prod_{i=1}^{d} \frac{h(\sigma_i)}{L(\sigma_i)} = \frac{1}{d!} \prod_{i=1}^{d} \theta(\sigma_i).$$

From Lemma 2.1.8, we have for each i, $\theta(\sigma_i) \leq \sqrt{\frac{i+1}{2i}}$. Therefore:

$$\Theta(\sigma_d) \leq \frac{1}{d!} \prod_{i=1}^{d} \sqrt{\frac{i+1}{2i}} = \frac{1}{d!}\sqrt{\frac{d+1}{2^d}}.$$

By a simple computation we can further establish that the fatness of the regular simplex is $\frac{1}{d!}\sqrt{\frac{d+1}{2^d}}$, and thus we conclude the proof. □

The proof of Theorem 2.1.2 follows from Lemmas 2.1.4, 2.1.5, 2.1.8 and 2.1.9.

	Fatness $\hat{\Theta}^{1/d}$	Aspect Ratio $\hat{\alpha}$	Thickness $\hat{\theta}$	Radius Ratio $\hat{\rho}$	Delaunay?
$\tilde{A}_d$, d odd	$\frac{2^{3/2}}{\left(\sqrt{d+1}\right)^{1+2/d}}$	$\sqrt{\frac{6d}{(d+1)(d+2)}}$	$\frac{2\sqrt{d}}{d+1}$	$\sqrt{\frac{6d}{(d+1)(d+2)}}$	✓
$\tilde{A}_d$, d even	$\frac{2^{3/2}(\sqrt{d+1})^{1-2/d}}{\sqrt{d(d+2)}}$		$\frac{2}{\sqrt{d+2}}$		
$\tilde{B}_d$	$\frac{2^{1/2+1/d}}{\sqrt{d}\left(\sqrt{d+1}\right)^{1/d}}$	$\frac{d\sqrt{2}}{(d+1)\sqrt{d+2}}$	$\frac{1}{\sqrt{d+1}}$	$\frac{2d}{\sqrt{d+2}(1+(d-1)\sqrt{2})}$	✗
$\tilde{C}_d$	$\frac{\sqrt{2}}{\sqrt{d}\left(\sqrt{d+1}\right)^{1/d}}$	$\frac{\sqrt{2d}}{d+1}$	$\frac{1}{\sqrt{d+1}}$	$\frac{2\sqrt{d}}{2+(d-1)\sqrt{2}}$	✓
$\tilde{D}_d$	$\frac{2^{1/2+2/d}}{\sqrt{d}\left(\sqrt{d+1}\right)^{1/d}}$	$\frac{d\sqrt{2}}{(d+1)\sqrt{d+4}}$	$\frac{1}{\sqrt{d+1}}$	$\frac{d\sqrt{2}}{(d-1)\sqrt{d+4}}$	✗
$\tilde{E}_6$	$\sqrt[12]{\frac{64}{137781}}$	$\frac{2}{7}$	$\frac{1}{\sqrt{14}}$	$\frac{1}{2}$	✗
$\tilde{E}_7$	$\sqrt[14]{\frac{1}{177147}}$	$\frac{7\sqrt{13}}{104}$	$\frac{\sqrt{21}}{24}$	$\frac{14\sqrt{13}}{117}$	✗
$\tilde{E}_8$	$\sqrt[8]{\frac{1}{3240}}$	$\frac{8\sqrt{19}}{171}$	$\frac{2\sqrt{19}}{57}$	$\frac{8\sqrt{19}}{95}$	✗
$\tilde{F}_4$	$\sqrt[8]{\frac{1}{405}}$	$\frac{4\sqrt{2}}{15}$	$\frac{2\sqrt{5}}{15}$	$\frac{4\sqrt{2}}{3(2+\sqrt{2})}$	✓
$\tilde{G}_2$	$\frac{\sqrt{2}}{2}$	$\frac{1}{\sqrt{3}}$	$\frac{1}{2}$	$\frac{2}{1+\sqrt{3}}$	✓

Table 2.1: Summary of quality measures of Coxeter triangulations.

2.2 Main result

In this section we present a table that summarizes explicit expressions of quality measures of Coxeter triangulations. Many of the provably good mesh generation algorithms are based on Delaunay triangulations [CDS12]. This motivated us to investigate if Coxeter triangulations have the Delaunay property. We thus identify which Coxeter triangulations are Delaunay and give their protection values. Finally, we identify which Coxeter triangulations have vertex sets on a lattice.

Theorem 2.2.1. *The normalized fatness, aspect ratio, thickness and radius ratio of simplices in Coxeter triangulations, as well as Delaunay property are presented in Table 2.1. Out of Delaunay triangulations, only $\tilde{A}$ family triangulations have a non-zero relative protection value equal to:*

$$\hat{\delta} = \frac{\sqrt{d^2+2d+24} - \sqrt{d^2+2d}}{\sqrt{d^2+2d}} \sim \frac{12}{d^2}$$

Only $\tilde{A}$ family, $\tilde{C}$ family and $\tilde{D}_4$ triangulations have vertex sets on a lattice.

The proof of this theorem and the table without normalization can be found in Section 2.4.

The corresponding quality measures for the regular d-simplex Δ (which does not correspond to a monohedral triangulation of $\mathbb{R}^d$ in general) are:

	Fatness Θ	Aspect Ratio α	Thickness θ	Radius Ratio ρ
Δ	$\frac{1}{d!}\sqrt{\frac{d+1}{2^d}}$	$\frac{d+1}{2d}$	$\sqrt{\frac{d+1}{2d}}$	$\frac{1}{d}$

All simplex quality measures in the table above are normalized with respect to the regular simplex. The non-normalized quality values are presented in Table 2.2.

Note that the fatness values in the table are given powered $1/d$. This is due to the fact that fatness is a volume-based simplex quality.

	Fatness Θ	Aspect Ratio α	Thickness θ	Radius Ratio ρ
$\tilde{A}_d$, d odd	$\frac{2^d}{\left(\sqrt{d+1}\right)^{d+1} d!}$	$\sqrt{\frac{3(d+1)}{2d(d+2)}}$	$\sqrt{\frac{2}{d+1}}$	$\sqrt{\frac{6}{d(d+1)(d+2)}}$
$\tilde{A}_d$, d even	$\frac{2^d(\sqrt{d+1})^{d-1}}{\left(\sqrt{d(d+2)}\right)^d d!}$		$\sqrt{\frac{2(d+1)}{d(d+2)}}$	
$\tilde{B}_d$	$\frac{2}{d^{d/2}d!}$	$\frac{1}{\sqrt{2(d+2)}}$	$\frac{1}{\sqrt{2d}}$	$\frac{2}{\sqrt{d+2}(1+(d-1)\sqrt{2})}$
$\tilde{C}_d$	$\frac{1}{d^{d/2}d!}$	$\frac{1}{\sqrt{2d}}$	$\frac{1}{\sqrt{2d}}$	$\frac{2}{\sqrt{d}(2+(d-1)\sqrt{2})}$
$\tilde{D}_d$	$\frac{4}{d^{d/2}d!}$	$\frac{1}{\sqrt{2(d+4)}}$	$\frac{1}{\sqrt{2d}}$	$\frac{\sqrt{2}}{\sqrt{d+4}(d-1)}$
$\tilde{E}_6$	$\frac{\sqrt{3}}{174960}$	$\frac{1}{6}$	$\frac{1}{2\sqrt{6}}$	$\frac{1}{12}$
$\tilde{E}_7$	$\frac{\sqrt{3}}{14696640}$	$\frac{1}{2\sqrt{13}}$	$\frac{1}{4\sqrt{3}}$	$\frac{2}{9\sqrt{13}}$
$\tilde{E}_8$	$\frac{1}{696729600}$	$\frac{1}{2\sqrt{19}}$	$\frac{1}{6\sqrt{2}}$	$\frac{1}{5\sqrt{19}}$
$\tilde{F}_4$	$\frac{1}{576}$	$\frac{\sqrt{2}}{6}$	$\frac{\sqrt{2}}{6}$	$\frac{\sqrt{2}}{3(2+\sqrt{2})}$
$\tilde{G}_2$	$\frac{\sqrt{3}}{8}$	$\frac{\sqrt{3}}{4}$	$\frac{\sqrt{3}}{4}$	$\frac{1}{1+\sqrt{3}}$

Table 2.2: The non-normalized values for fatness, aspect ratio, thickness and radius ratio of Coxeter triangulations.

Also note that all normalized simplex qualities for the families $\tilde{A}_d$, $\tilde{B}_d$, $\tilde{C}_d$ and $\tilde{D}_d$ behave as $O(\frac{1}{\sqrt{d}})$.

The numerical values of these quality measures are given in Tables 2.4, 2.5, 2.6 and 2.7 in Section 2.6, and plotted in Figure 2.4 below. A quick glance suffices to see that Coxeter triangulations of type $\tilde{A}_d$ achieve the greatest aspect ratio, fatness, thickness and radius ratio among the Coxeter triangulations in each dimension d.

2.3 Delaunay criterion for monohedral triangulations of $\mathbb{R}^d$

The goal of this section is to establish a criterion that tells if a given monohedral triangulation of $\mathbb{R}^d$ is Delaunay. We would like to stress that the triangulations in this section are infinite and do not have boundary. Most of the results in this section are not applicable to triangulations with a finite number of vertices.

This section deals with general monohedral triangulations of $\mathbb{R}^d$; the criterion for the particular case of Coxeter triangulations is presented in Theorem 2.3.8 at the end of the section.

Definition 2.3.1. *A simplex is called* self-centred *if it contains its circumcentre inside or on the boundary.*

From Rajan [Raj94] we know that:

Lemma 2.3.2 ([Raj94, Theorem 5])**.** *If a triangulation of $\mathbb{R}^d$ consists of only self-centred simplices, then it is a Delaunay triangulation.*

For any monohedral triangulation of $\mathbb{R}^d$, it is sufficient to check if one simplex in the triangulation is self-centred to conclude that the triangulation is Delaunay.

Corollary 2.3.3. *If a simplex in a monohedral triangulation of $\mathbb{R}^d$ is self-centred, then this triangulation is Delaunay.*

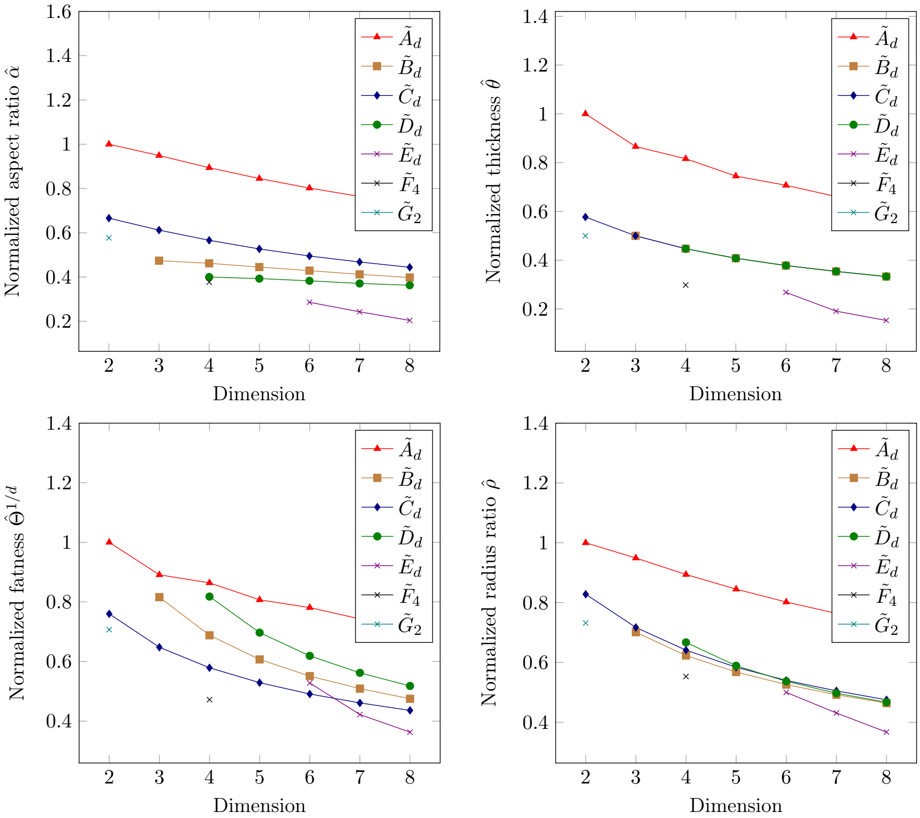


Figure 2.4: The visual representation of the normalized aspect ratio value, thickness, fatness and the radius ratio for simplices of $\tilde{A}_d$, $\tilde{B}_d$, $\tilde{C}_d$ and $\tilde{D}_d$ triangulations.

We now want to consider the converse:

Lemma 2.3.4. *Let R be the maximum circumradius in a Delaunay triangulation $\mathcal{D}$ of $\mathbb{R}^d$. Then any simplex in $\mathcal{D}$ with the circumradius R is self-centred.*

We would like to emphasize that Lemma 2.3.4 does not generalize to the standard definition of Delaunay triangulation on a finite set of points. This can be seen on a simple example of three points on a plane that lie on an obtuse triangle.

Proof. Let σ be a simplex in $\mathcal{D}$ with circumsphere S, whose circumcentre is c and circumradius is R. We assume that the circumcentre c lies outside of σ and derive a contradiction. Because the circumcentre lies outside σ, there exists a facet τ of σ whose supporting hyperplane H_τ separates c from σ. We denote by σ' the simplex in $\mathcal{D}$ that shares the facet τ with σ. We write S' for the circumsphere of σ' and denote its circumcentre by c' and circumradius by R'. The $(d-2)$-dimensional sphere T that forms the intersection of S and S', divides S into a part that lies inside S' and a cap that lies outside S'.

We shall call ℓ the line on which c and c' lie. If we restrict to any two-dimensional plane P that contains ℓ, we see that $S \cap P$ and $S' \cap P$ are circles. We shall think of ℓ as vertical in P (see Figure 2.5). The circumcentres cannot coincide, because otherwise they would lie on H_τ. So the circles $S \cap P$ and $S' \cap P$ intersect transversely in two points. This means that $S \cap P$ is subdivided into two parts: a part inside $S' \cap P$ and a cap outside $S' \cap P$.

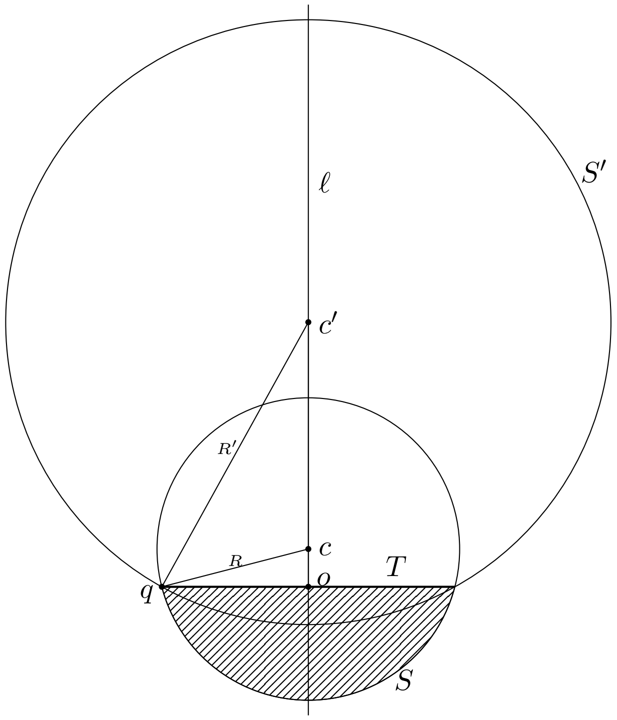


Figure 2.5: Construction for the proof of Lemma 2.3.4. The construction is confined to the plane containing the segments $[oc]$ and $[oq]$.

We shall assume that the cap lies below the part inside $S' \cap P$, see Figure 2.5. This implies that c lies below c' on ℓ. This can be seen as follows: assume that both S and H_τ are fixed. This means that points of intersection of $S \cap P$ and $S' \cap P$ are fixed, but the circumcentre c' can be varied. We now let c' descend along ℓ from $+\infty$ to $-\infty$. As long as c and c' do not coincide, the cap of $S \cap P$ outside $S' \cap P$ remains stable. At the point of coincidence of c and c' the inside and outside switch. It follows by considering the limit cases that the cap $S \cap P$ outside $S' \cap P$ lies below the cap inside as long as c lies below c' on ℓ, as in Figure 2.5.

We now make two important observations. Firstly, because the triangulation $\mathcal{D}$ is Delaunay, the sphere S' does not contain the vertex p of σ not on τ. Hence p lies on the cap of S which is outside S'. Secondly, as we assume that the circumcentre c lies outside of σ, c lies above $H_\tau \cap P$.

Now let o be the centre of $T = S \cap S'$ and let q be a vertex of τ. Because $d(c, o) < d(c', o)$, the distance $d(c, q)$ is strictly smaller than $d(c', q)$ (see Figure 2.5). By remarking that $d(c, q) = R$ and $d(c', q) = R'$, we get $R < R'$, which is a contradiction with the hypothesis that R is the largest circumradius in $\mathcal{D}$. □

In monohedral triangulations, all simplices have the same circumradius. So trivially, all simplices have the maximum circumradius in the triangulation. This observation along with Lemma 2.3.4 leads us to conclude that if a simplex in a monohedral triangulation of $\mathbb{R}^d$ is not self-centred, then the triangulation is not Delaunay. Combining this with Corollary 2.3.3 yields:

Theorem 2.3.5. *A monohedral triangulation of $\mathbb{R}^d$ is Delaunay if and only if its simplices are self-centred.*

In the spirit of Theorem 2.3.5, we can easily spot if a Delaunay triangulation of $\mathbb{R}^d$ is not protected with the help of the following lemma:

Lemma 2.3.6. *A Delaunay triangulation of $\mathbb{R}^d$ where a simplex with the maximal circumradius contains its circumcentre on the boundary is not protected.*

Proof. We use the same construction and the same notations as in the proof of Lemma 2.3.4 (see Figure 2.5). Observe that by hypothesis o coincides with c. If c was different from c', then the radius R' would be greater than R, which would lead to a contradiction. Therefore c coincides with c', so σ and σ' share the same circumball. We conclude the proof by observing that the vertex of σ' not in τ lies on the circumball of σ, therefore the protection of σ is zero. □

We can state the converse of Lemma 2.3.6 for monohedral triangulations of $\mathbb{R}^d$. For this, we slightly modify Lemma 8 and Theorem 5 in [Raj94] by replacing all non-strict inequalities to strict inequalities to get a criterion for a non-zero protection:

Lemma 2.3.7. *If a simplex of a monohedral triangulation of $\mathbb{R}^d$ contains the circumcentre strictly inside, then the triangulation is Delaunay with a non-zero protection.*

By using the fact that Coxeter triangulations are monohedral, we derive from Theorem 2.3.5, Lemma 2.3.6 and Lemma 2.3.7 the following theorem:

Theorem 2.3.8. *Let $\mathcal{T}$ be a Coxeter triangulation.*

- *If a simplex in $\mathcal{T}$ is not self-centred, then $\mathcal{T}$ is not Delaunay.*
- *If a simplex in $\mathcal{T}$ contains its circumcentre on the boundary, then $\mathcal{T}$ is Delaunay with zero protection.*
- *If a simplex in $\mathcal{T}$ contains its circumcentre strictly inside, then $\mathcal{T}$ is Delaunay with non-zero protection.*

2.4 Geometrical analysis of each family of Coxeter triangulations

In this section, we present the proof of Theorem 2.2.1. The proof consists of a case study for each individual family of Coxeter triangulations. All cases are independent one from another. We provide explicit measures, so that any reader interested in a quality measure that is not covered by the current study can compute its value for Coxeter triangulations. This can be useful for a comparison of Coxeter triangulations with any other triangulation based on any custom quality measure.

The study for each case starts by presenting two Coxeter diagrams. Each vertex in the Coxeter diagram on the left has a label. These labels are numbers proportional to the inverse heights of the fundamental simplex in the Coxeter triangulation (recall from Proposition 1.2.29 that the fundamental simplex is defined by hyperplanes that are orthogonal to simple roots and the highest root). This information is taken from [DWLT90]. The Coxeter diagram on the right indicates the notations of the corresponding facets in the proofs.

Each case follows the same plan:

1. Let σ be the fundamental simplex in the Coxeter triangulation generated by the root system with simple roots from Theorem 1.2.15. We give equations of the hyperplanes that contain the facets τ_i of the simplex σ. The equations of the hyperplanes can be found in [Bou02] or [CS87], or alternatively can be deduced from Theorem 1.2.15.

$\tilde{A}_d$, given in $\mathbb{R}^{d+1}$, $d \geq 2$	$u_0 = \left(0^{\{d+1\}}\right)$ $u_k = \left((-\frac{d+1-k}{d+1})^{\{k\}}, (\frac{k}{d+1})^{\{d+1-k\}}\right), \quad \forall k \in \{1, \dots, d\}$
$\tilde{B}_d$, given in $\mathbb{R}^d$, $d \geq 3$	$u_0 = \left(0^{\{d\}}\right)$ $u_1 = \left(1, 0^{\{d-1\}}\right)$ $u_k = \left(\frac{1}{2}^{\{k\}}, 0^{\{d-k\}}\right), \forall k \in \{2, \dots, d\}$
$\tilde{C}_d$, given in $\mathbb{R}^d$, $d \geq 2$	$u_k = \left(\frac{1}{2}^{\{k\}}, 0^{\{d-k\}}\right), \forall k \in \{0, \dots, d\}$
$\tilde{D}_d$, given in $\mathbb{R}^d$, $d \geq 4$	$u_0 = \left(0^{\{d\}}\right)$ $u_1 = \left(1, 0^{\{d-1\}}\right)$ $u_k = \left(\frac{1}{2}^{\{k\}}, 0^{\{d-k\}}\right), \forall k \in \{2, \dots, d-2\}$ $u_{d-1} = \left(\frac{1}{2}^{\{d-1\}}, -\frac{1}{2}\right)$ $u_d = \left(\frac{1}{2}^{\{d\}}\right)$
$\tilde{E}_6$, given in $\mathbb{R}^8$	$u_0 = \left(0^{\{8\}}\right)$ $u_1 = \left(0^{\{5\}}, -\frac{2}{3}^{\{2\}}, \frac{2}{3}\right)$ $u_2 = \left(\frac{1}{4}^{\{5\}}, -\frac{1}{4}^{\{2\}}, \frac{1}{4}\right)$ $u_3 = \left(-\frac{1}{4}, \frac{1}{4}^{\{4\}}, -\frac{5}{12}^{\{2\}}, \frac{5}{12}\right)$ $u_4 = \left(0^{\{2\}}, \frac{1}{3}^{\{3\}}, -\frac{1}{3}^{\{2\}}, \frac{1}{3}\right)$ $u_5 = \left(0^{\{3\}}, \frac{1}{2}^{\{2\}}, -\frac{1}{3}^{\{2\}}, \frac{1}{3}\right)$ $u_6 = \left(0^{\{4\}}, 1, -\frac{1}{3}^{\{2\}}, \frac{1}{3}\right)$
$\tilde{E}_7$, given in $\mathbb{R}^8$	$u_0 = \left(0^{\{8\}}\right)$ $u_1 = \left(0^{\{6\}}, \frac{1}{2}, -\frac{1}{2}\right)$ $u_2 = \left(-\frac{1}{4}^{\{6\}}, \frac{1}{2}, -\frac{1}{2}\right)$ $u_3 = \left(\frac{1}{6}, -\frac{1}{6}^{\{5\}}, \frac{1}{2}, -\frac{1}{2}\right)$ $u_4 = \left(0^{\{2\}}, -\frac{1}{4}^{\{4\}}, \frac{1}{2}, -\frac{1}{2}\right)$ $u_5 = \left(0^{\{3\}}, -\frac{1}{3}^{\{3\}}, \frac{1}{2}, -\frac{1}{2}\right)$ $u_6 = \left(0^{\{4\}}, -\frac{1}{2}^{\{2\}}, \frac{1}{2}, -\frac{1}{2}\right)$ $u_7 = \left(0^{\{5\}}, -1, \frac{1}{2}, -\frac{1}{2}\right)$
$\tilde{E}_8$, given in $\mathbb{R}^8$	$u_0 = \left(0^{\{8\}}\right)$ $u_1 = \left(0^{\{7\}}, 1\right)$ $u_2 = \left(\frac{1}{6}^{\{7\}}, \frac{5}{6}\right)$ $u_3 = \left(-\frac{1}{8}, \frac{1}{8}^{\{6\}}, \frac{7}{8}\right)$ $u_4 = \left(0^{\{2\}}, \frac{1}{6}^{\{5\}}, \frac{5}{6}\right)$ $u_5 = \left(0^{\{3\}}, \frac{1}{5}^{\{4\}}, \frac{4}{5}\right)$ $u_6 = \left(0^{\{4\}}, \frac{1}{4}^{\{3\}}, \frac{3}{4}\right)$ $u_7 = \left(0^{\{5\}}, \frac{1}{3}^{\{2\}}, \frac{2}{3}\right)$ $u_8 = \left(0^{\{6\}}, \frac{1}{2}^{\{2\}}\right)$
$\tilde{F}_4$, given in $\mathbb{R}^4$	$u_0 = (0, 0, 0, 0)$ $u_1 = (\frac{1}{2}, \frac{1}{2}, 0, 0)$ $u_2 = (\frac{2}{3}, \frac{1}{3}, \frac{1}{3}, 0)$ $u_3 = (\frac{3}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4})$ $u_4 = (1, 0, 0, 0)$
$\tilde{G}_2$, given in $\mathbb{R}^2$	$u_0 = (0, 0)$ $u_1 = (1, 0)$ $u_2 = (1, \sqrt{3})$

Table 2.3: Vertices of a simplex per Coxeter diagram type. Although a partial list has been provided by Conway and Sloane [CS87] we are not aware of any complete explicit overview. We believe that such a list will be essential for many practitioners. The powers in point coordinates correspond to duplications of the same coordinate, for example $(\frac{1}{2}^{\{3\}}, -\frac{1}{2})$ is the same as $(\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, -\frac{1}{2})$.

2. We give the coordinates of the vertices u_i of the simplex σ. The vertices for each of the families of Coxeter triangulations are summarized in Table 2.3.

3. By using the augmented Coxeter diagram we find which height is the shortest (denoted by $h(\sigma)$). We then compute it as the distance from the corresponding vertex to the corresponding hyperplane.

4. We find the circumradius $R(\sigma)$ and the inradius $r(\sigma)$ of the simplex.

5. We find the length of the longest edge $L(\sigma)$, using the coordinates of the vertices found previously.

6. We compute the (non-normalized) thickness $\theta(\sigma) = \frac{h(\sigma)}{L(\sigma)}$, the aspect ratio $\alpha(\sigma) = \frac{h(\sigma)}{2R(\sigma)}$ and the radius ratio $\rho(\sigma) = \frac{r(\sigma)}{R(\sigma)}$.

7. We compute the volume $vol(\sigma)$ and the (non-normalized) fatness $\Theta(\sigma) = \frac{vol(\sigma)}{L(\sigma)^d}$.

8. We determine if the triangulation is Delaunay with the help of Theorem 2.3.8. If it is, we also determine if its protection is non-zero. The computation of the protection value of Coxeter triangulations of $\tilde{A}$ family is done separately in Section 2.5.

9. We determine if the vertex set of the triangulation form a lattice.

We will adopt the following writing convention: the powers in point coordinates correspond to duplications of the same coordinate. For example: $(0, 1^{\{3\}}, 0)$ is the same as $(0, 1, 1, 1, 0)$.

2.4.1 $\tilde{\mathbf{A}}_{\mathbf{d}}, \mathbf{d} \geqslant 2$

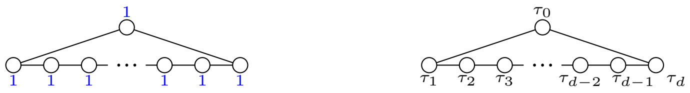


Figure 2.6: On the left: the Coxeter diagram for the $\tilde{A}_d$ triangulation. The labels on vertices are the inverse ratios of heights. On the right: the notations of the facets of the fundamental simplex that are used in the proofs.

1. The hyperplanes that contain the facets of the fundamental simplex σ in $\mathbb{R}^{d+1}$ can be defined as the intersection of the hyperplane $\sum_{i=0}^{d} x_i = 0$ and the following hyperplanes:

   for $\tau_0 : \ -x_0 + x_d = 1$
   for $\tau_k : \ x_k - x_{k-1} = 0 \qquad \forall k \in \{1, \dots, d\}$.

2. From the equations of hyperplanes it is easy to check that the following points in $\mathbb{R}^{d+1}$:
   $u_0 = \left(0^{\{d+1\}}\right)$
   $u_k = \left((-\frac{d+1-k}{d+1})^{\{k\}}, (\frac{k}{d+1})^{\{d+1-k\}}\right), \quad \forall k \in \{1, \dots, d\},$ are the vertices of σ.

3. According to Figure 2.6 left, all heights are equal. By computing, for example, the distance from u_0 to the hyperplane $-x_0 + x_d = 1$ we get:

$$h(\sigma) = \frac{1}{\sqrt{2}}.$$

4. It is easy to verify that the barycentre:

$$c = \left(-\frac{d}{2(d+1)}, -\frac{d-2}{2(d+1)}, -\frac{d-4}{2(d+1)}, \dots, \frac{d}{2(d+1)}\right)$$

is equidistant from all vertices, so it is the circumcentre. The distance from c to vertices is the circumradius: $R(\sigma) = \sqrt{\frac{d(d+2)}{12(d+1)}}$.

Again, it is easy to verify that c is also equidistant from all hyperplanes that contain τ_i, therefore c is also the incentre. The distance from c to the hyperplanes is the inradius: $r(\sigma) = \frac{1}{\sqrt{2}(d+1)}$.

5. The edges of σ are described by differences $u_k - u_j$, for certain $j, k \in \{0, \dots, d\}$ with $j < k$.

The squared norm of such a difference is equal to $\|\vec{u}_k - \vec{u}_j\|^2 = \frac{(k-j)(d+1-k+j)}{d+1}$.

An easy analysis (substitute $k-j$ as a new variable) yields that this function on k and j is maximal when $k - j = (d+1)/2$.

So the maximal edge in σ has length $L(\sigma) = \begin{cases} \frac{\sqrt{d+1}}{2} & \text{if } d \text{ is odd,} \\ \frac{1}{2}\sqrt{\frac{d(d+2)}{(d+1)}} & \text{if } d \text{ is even.} \end{cases}$

6. The aspect ratio, the thickness and the radius ratio are:

$$\begin{aligned} \alpha(\sigma) = \tfrac{h(\sigma)}{R(\sigma)} &= \sqrt{\tfrac{3(d+1)}{2d(d+2)}} \\ \theta(\sigma) = \tfrac{h(\sigma)}{L(\sigma)} &= \begin{cases} \sqrt{\frac{2}{d}} & \text{if } d \text{ is odd,} \\ \sqrt{\frac{2(d+1)}{d(d+2)}} & \text{if } d \text{ is even.} \end{cases} \\ \rho(\sigma) = \tfrac{r(\sigma)}{R(\sigma)} &= \sqrt{\tfrac{6}{d(d+1)(d+2)}} \end{aligned}$$

7.

Lemma 2.4.1. *Simplices in the $\tilde{A}_d$ triangulation have fatness:*

$$\Theta(\sigma) = \begin{cases} \frac{2^d}{\left(\sqrt{d+1}\right)^{d+1} d!} & \textit{if } d \textit{ is odd,} \\ \frac{2^d(\sqrt{d+1})^{d-1}}{\left(\sqrt{d(d+2)}\right)^{d} d!} & \textit{if } d \textit{ is even.} \end{cases}$$

The lemma above comes mostly from the fact that the fundamental simplex σ has volume:

$$vol(\sigma) = \frac{1}{\sqrt{d+1}\; d!}.$$

Proof. Fix the dimension $d \geq 2$. Let a_k denote $-\frac{d+1-k}{d+1}$ and b_k denote $\frac{k}{d+1}$.

Denote by σ the d-simplex of a $\tilde{A}_d$ triangulation described above. We build a $(d+1)$-simplex σ', which consists of the d-face σ and a vertex $u_{d+1} = (1^{\{d+1\}})$. Note that the edge $u_0 u_{d+1}$ is orthogonal to the face σ.

The $(d+1)$-dimensional volume of σ' can be expressed as a determinant:

$$vol(\sigma') = \left| \frac{1}{(d+1)!} \det \begin{pmatrix} a_1 & a_2 & \dots & a_d & 1 \\ b_1 & a_2 & \dots & a_d & 1 \\ b_1 & b_2 & \dots & a_d & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ b_1 & b_2 & \dots & b_d & 1 \end{pmatrix} \right|.$$

First, we subtract the last row from all other rows. It gives us:

$$\begin{aligned} vol(\sigma') &= \left| \frac{1}{(d+1)!} \det \begin{pmatrix} a_1 - b_1 & a_2 - b_2 & \dots & a_d - b_d & 0 \\ 0 & a_2 - b_2 & \dots & a_d - b_d & 0 \\ 0 & 0 & \dots & a_d - b_d & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ b_1 & b_2 & \dots & b_d & 1 \end{pmatrix} \right| \\ &= \left| \frac{1}{(d+1)!} \det \begin{pmatrix} a_1 - b_1 & a_2 - b_2 & \dots & a_d - b_d \\ 0 & a_2 - b_2 & \dots & a_d - b_d \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & a_d - b_d \end{pmatrix} \right| \\ &= \left| \frac{1}{(d+1)!} \prod_{k=1}^{d} (a_k - b_k) \right| = \left| \frac{1}{(d+1)!} \prod_{k=1}^{d} \left(-\frac{d+1-k}{d+1} - \frac{k}{d+1} \right) \right| = \frac{1}{(d+1)!}. \end{aligned}$$

The volume of σ can be found as:

$$vol(\sigma) = \frac{(d+1) vol(\sigma')}{\|(1^{\{d+1\}})\|} = \frac{(d+1)}{\sqrt{d+1}(d+1)!} = \frac{1}{\sqrt{d+1}\ d!}.$$

□

8. Because the circumcentre coincides with the incentre, it lies strictly inside the simplex. So by Theorem 2.3.8 the triangulation is Delaunay with non-zero protection. The exact value of protection is shown in Section 2.5.

9.

Proposition 2.4.2. *The vertex set of a $\tilde{A}_d$ triangulation is a lattice for all $d \geq 2$.*

As we will show, this lattice is in fact the coweight lattice introduced in Section 1.2.3. The fact that the vertex set is a lattice is not entirely obvious. This lattice happens to be equal to the permutahedral lattice studied in for example [CS87].

Before proving the proposition, we will need the following lemmas:

Lemma 2.4.3. *Let R be a crystallographic root system. Let V be the vertex set of the Coxeter triangulation corresponding to R. The coweight lattice $\Lambda^w_{R^\vee}$ is a subset of V.*

Proof. Let $S = \{s_1, \dots, s_d\}$ be a set of simple roots in R. For any point p in $\Lambda^w_{R^\vee}$ there exist integers $n_1, \dots, n_d$ such that for every i, $\langle p, s_i \rangle = n_i$. The point p lies in the intersection of the hyperplanes H_{s_i,k_i}. Therefore, being in the intersection of d linearly independent hyperplanes in $\mathcal{H}$, it is a vertex in the Coxeter triangulation. □

Lemma 2.4.4. *Let R be a crystallographic root system. The coweight lattice $\Lambda^w_{R^\vee}$ is stable by the action of elements of the affine group W_a that corresponds to R.*

Proof. Take $x \in \Lambda^w_{R^\vee}$. Let $r \in R$ and $k \in \mathbb{Z}$. The image of x by the affine reflection $\sigma_{r,k}$ is:

$$\sigma_{r,k}(x) = x - (\langle x, r\rangle - k)r^\vee.$$

Let r' be another root in R. The scalar product of $\sigma_{r,k}(x)$ with r' is:

$$\left\langle \sigma_{r,k}(x), r' \right\rangle = \left\langle x, r' \right\rangle - (\langle x, r\rangle - k) \left\langle r^\vee, r' \right\rangle . \tag{2.6}$$

The first term in equation 2.6 is an integer by assumption that x lies in the coweight lattice. By Remark 1.2.20, the second term in equation 2.6 can be expressed as $(\langle x, r\rangle - k)n(r, r')$ (see the definition of $n(r, r')$ in Section 1.2.2). The scalar product $\langle x, r\rangle$ is an integer by assumption that x lies in the coweight lattice. From Section 1.2.2, we know that $n(r, r')$ is an integer, and k is an integer by definition.

We conclude that $\langle \sigma_{r,k}(x), r'\rangle$ is an integer for all $r' \in R$. This implies that $\sigma_{r,k}(x)$ lies in the coweight lattice for all $r \in R$ and $k \in \mathbb{Z}$.

By definition, the affine Weyl group W_a is generated by the affine reflections $\sigma_{r,k}$ for all $r \in R$ and $k \in \mathbb{Z}$. So we conclude that the coweight lattice $\Lambda^w_{R^\vee}$ is stable by the action of elements of the affine group W_a. □

Now, we are ready to prove Proposition 2.4.2.

Proof (of Proposition 2.4.2). Fix a Coxeter triangulation of type $\tilde{A}_d$ that corresponds to a root system R and let V be its vertex set. We will prove the proposition by double inclusion.

The inclusion $\Lambda^w_{R^\vee} \subseteq V$ follows from Lemma 2.4.3. We will now prove that $V \subseteq \Lambda^w_{R^\vee}$.

Let v be a vertex in V. Let σ be the fundamental simplex from the beginning of the section, and σ' be an simplex in the triangulation, such that v is its vertex. Let ω be an affine group transformation that maps σ to σ'. In particular, ω maps a vertex in σ to v.

Observe that all vertices in σ have integer scalar products with all roots, so they lie in the coweight lattice. With this observation, Lemma 2.4.4 yields that v lies in the coweight lattice too.

Thus, we conclude that $V \subseteq \Lambda^w_{R^\vee}$, so the double inclusion is complete. □

2.4.2 $\tilde{\mathbf{B}}_\mathbf{d}, \mathbf{d} \geqslant 3$

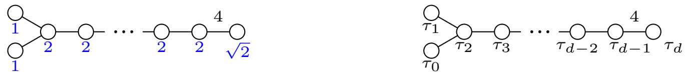


Figure 2.7: On the left: the Coxeter diagram for a triangulation of type $\tilde{B}_d$. The labels on vertices are the inverse ratios of heights. On the right: the notations of the facets of the fundamental simplex that are used in the proofs.

1. The hyperplanes that contain facets of the fundamental simplex σ can be defined as follows:
   for $\tau_0 :\ x_1 + x_2 = 1$
   for $\tau_k :\ x_k - x_{k+1} = 0, \forall k \in \{1, \dots, d-1\}$
   for $\tau_d :\ x_d = 0.$

2. From the equations of hyperplanes it is easy to check that the following points:
   $u_0 = \left(0^{\{d\}}\right)$
   $u_1 = \left(1, 0^{\{d-1\}}\right)$ are the vertices of σ.
   $u_k = \left(\frac{1}{2}^{\{k\}}, 0^{\{d-k\}}\right), \forall k \in \{2, \dots, d\}.$

3. According to Figure 2.7, the minimal height falls on any of τ_k for k in $\{2, \dots, d-1\}$. By computing, for example, the distance from u_1 to the hyperplane $x_1 - x_2 = 0$ we get:
$$h(\sigma) = \frac{1}{2\sqrt{2}}.$$

4. The circumcentre of the simplex is $(\frac{1}{2}, 0, \frac{1}{4}^{\{d-2\}})$ and the circumradius is $R(\sigma) = \frac{\sqrt{d+2}}{4}$, which is easily verifiable. The incentre is $\left(\frac{1+(d-1)\sqrt{2}}{2(1+(d-1)\sqrt{2})}, \frac{1+(d-2)\sqrt{2}}{2(1+(d-1)\sqrt{2})}, \dots, \frac{1}{2(1+(d-1)\sqrt{2})}\right)$ and the inradius is $r(\sigma) = \frac{1}{2(1+(d-1)\sqrt{2})}$, which is easily verifiable.

5. The longest edge is given by $\vec{u}_d - \vec{u}_0$ and is equal to $L(\sigma) = \frac{\sqrt{d}}{2}$.

6. The aspect ratio is:
$$\alpha(\sigma) = \frac{h(\sigma)}{R(\sigma)} = \frac{1}{\sqrt{2(d+2)}}.$$
   The thickness is:
$$\theta(\sigma) = \frac{h(\sigma)}{L(\sigma)} = \frac{1}{\sqrt{2d}}.$$
   The radius ratio is:
$$\rho(\sigma) = \frac{r(\sigma)}{R(\sigma)} = \frac{2}{\sqrt{d+2}(1+(d-1)\sqrt{2})}.$$

7. The volume of the simplex is given by the formula:
$$vol(\sigma) = \frac{1}{d!} \det \begin{pmatrix} 1 & 1/2 & \dots & 1/2 \\ 0 & 1/2 & & 1/2 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & 1/2 \end{pmatrix} = \frac{1}{2^{d-1} d!}.$$
   The fatness of the simplex is:
$$\Theta(\sigma) = \frac{vol(\sigma)}{L(\sigma)^d} = \frac{2}{d^{d/2} d!}.$$

8. Observe that the inner products of the normal vector s_2 to the hyperplane that contains τ_2 with the circumcentre and u_2 have different signs:
$$\langle s_2, c \rangle = -\frac{1}{4} \quad \text{and} \quad \langle s_2, u_2 \rangle = \frac{1}{2}.$$
   Therefore the hyperplane that contains τ_2 separates the circumcentre c from u_2, so by Theorem 2.3.8 the triangulation is not Delaunay.

9.

Proposition 2.4.5. *No vertex set of a $\tilde{B}_d$ triangulation for $d \geq 3$ is a lattice.*

Proof. A point $v \in \mathbb{R}^d$ is a vertex of the triangulation if and only if it lies in the intersection of at least d hyperplanes in $\mathcal{H}$ (as defined in Section 1.2.4) that are linearly independent. Equivalently, the inner products of at least d linearly independent positive roots should be integers.

Let $E = \{e_1, ..., e_d\}$ be the canonical basis of $\mathbb{R}^d$. Bourbaki [Bou02, Planche II] gives the explicit expressions of positive roots:

$$\begin{aligned} R^+ \quad = \quad & \{e_i \mid i \in \{1, \dots, d\}\} \cup \\ & \{e_i - e_j \mid i, j \in \{1, \dots, d\} \text{ and } i < j\} \cup \\ & \{e_i + e_j \mid i, j \in \{1, \dots, d\} \text{ and } i < j\}. \end{aligned}$$

We will now prove that $v = u_d - u_{d-1} = \frac{1}{2}e_d$ is not in the vertex set, which implies that V does not form a lattice.

The positive roots that have integer inner product with v are exactly:

$$\begin{aligned} & \{e_i \mid i \in \{1, \dots, d-1\}\} \cup \\ & \{e_i - e_j \mid i, j \in \{1, \dots, d-1\} \text{ and } i < j\} \cup \\ & \{e_i + e_j \mid i, j \in \{1, \dots, d-1\} \text{ and } i < j\}, \end{aligned}$$

which is easily verifiable.

These roots span the vector space $\mathrm{Vect}(E \setminus \{e_d\})$, which is $(d-1)$-dimensional. Thus, there do not exist $(r_1, ..., r_d) \subseteq R^+$ that are linearly independent and have integer inner products with v. Therefore v does not belong to V.

□

2.4.3 $\tilde{\mathbf{C}}_\mathbf{d}, \mathbf{d} \geqslant \mathbf{2}$

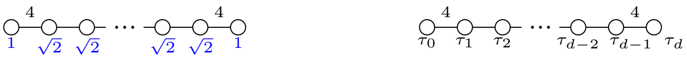


Figure 2.8: On the left: the Coxeter diagram for a triangulation of type $\tilde{C}_d$. The labels on vertices are the inverse ratios of heights. On the right: the notations of the facets of the fundamental simplex that are used in the proofs.

1. The hyperplanes that contain facets of the fundamental simplex σ can be defined as follows:

   for $\tau_0 :\ 2x_1 = 1$
   for $\tau_k :\ x_k - x_{k+1} = 0, \forall k \in \{1, \dots, d-1\}$
   for $\tau_d :\ x_d = 0.$

2. From the equations of hyperplanes it is easy to check that the following points:

   $u_k = \left(\frac{1}{2}^{\{k\}}, 0^{\{d-k\}}\right), \forall k \in \{0, \dots, d\}.$ are the vertices of σ.

3. According to Figure 2.8, the minimal height falls on any of τ_k for k in $\{1, \dots, d-1\}$. By computing, for example, the distance from u_1 to the hyperplane $x_1 - x_2 = 0$ we get:
$$h(\sigma) = \frac{1}{2\sqrt{2}}.$$

4. The circumcentre of the simplex is $(\frac{1}{4}^{\{d\}})$ and the circumradius is $R(\sigma) = \frac{\sqrt{d}}{4}$, which is easily verifiable. The incentre is $\left(\frac{1+(d-1)\sqrt{2}}{2(2+(d-1)\sqrt{2})}, \frac{1+(d-2)\sqrt{2}}{2(2+(d-1)\sqrt{2})}, \dots, \frac{1}{2(2+(d-1)\sqrt{2})}\right)$ and the inradius is $r(\sigma) = \frac{1}{2(2+(d-1)\sqrt{2})}$, which is easily verifiable.

5. The longest edge is given by $\vec{u}_d - \vec{u}_0$ and is equal to $L(\sigma) = \frac{\sqrt{d}}{2}$.

6. The aspect ratio is:
$$\alpha(\sigma) = \frac{h(\sigma)}{R(\sigma)} = \frac{1}{\sqrt{2d}}.$$
The thickness is:
$$\theta(\sigma) = \frac{h(\sigma)}{L(\sigma)} = \frac{1}{\sqrt{2d}}.$$
The radius ratio is:
$$\rho(\sigma) = \frac{r(\sigma)}{R(\sigma)} = \frac{2}{\sqrt{d}(2+(d-1)\sqrt{2})}.$$

7. The volume of the simplex is given by the formula:
$$vol(\sigma) = \frac{1}{d!} \det \begin{pmatrix} 1/2 & 1/2 & \dots & 1/2 \\ 0 & 1/2 & & 1/2 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & 1/2 \end{pmatrix} = \frac{1}{2^d d!}.$$
The fatness of the simplex is:
$$\Theta(\sigma) = \frac{vol(\sigma)}{L(\sigma)^d} = \frac{1}{d^{d/2} d!}.$$

8. Observe that all inner products of the normals s_i to hyperplanes that contain τ_i and the corresponding opposite vertices u_i are positive. Observe as well that the inner products with the circumcentre $\langle s_i, c\rangle$ are either positive or zero. It implies that the circumcentre lies on the boundary of the simplex, therefore by Theorem 2.3.8 the triangulation is non-protected Delaunay.

9.

Proposition 2.4.6. *The vertex set of a $\tilde{C}_d$ triangulation is a lattice for $d \geq 2$.*

Proof. A point $v \in \mathbb{R}^d$ is a vertex of the triangulation if and only if it lies in the intersection of at least d hyperplanes in $\mathcal{H}$ (as defined in Section 1.2.4) that are linearly independent. Equivalently, the inner products of at least d linearly independent positive roots should be integers.

Let $E = \{e_1, ..., e_d\}$ be the canonical basis of $\mathbb{R}^d$. Bourbaki [Bou02, Planche III] gives the explicit expressions of positive roots:

$$\begin{aligned} R^+ \;\; = \;\; & \{2e_i \mid i \in \{1, \dots, d\}\} \cup \\ & \{e_i - e_j \mid i, j \in \{1, \dots, d\} \text{ and } i < j\} \cup \\ & \{e_i + e_j \mid i, j \in \{1, \dots, d\} \text{ and } i < j\}. \end{aligned}$$

We will denote by V the vertex set of the $\tilde{C}_d$ triangulation that contains the simplex σ. The goal is to prove that V is equal to $(\frac{1}{2}\mathbb{Z})^d$, which is a lattice.

We will first prove that $V \subseteq (\frac{1}{2}\mathbb{Z})^d$. This follows from the fact that $(\frac{1}{2}\mathbb{Z})^d$ is stable by the action of elements of the affine Weyl group W_a. Indeed, for any $x \in (\frac{1}{2}\mathbb{Z})^d$, its image by an affine reflection $\sigma_{r,k}$ for $r \in R$ and $k \in \mathbb{Z}$, as defined in Section 1.2.4, is $\sigma_{r,k}(x) = x - (\langle x, r\rangle - k)r^\vee$. For any $r \in R$ and $x \in (\frac{1}{2}\mathbb{Z})^d$, the inner product $\langle x, r\rangle$ is in $\frac{1}{2}\mathbb{Z}$, hence $(\langle x, r\rangle - k)$ is in $\frac{1}{2}\mathbb{Z}$ as well. The dual root $r^\vee$ has integer coefficients, so $\sigma_{r,k}(x) = x - (\langle x, r\rangle - k)r^\vee$ is in $(\frac{1}{2}\mathbb{Z})^d$. Now recall that one can obtain any simplex from another one by an element of W_a, which is generated by affine reflections $\sigma_{r,k}$. All vertices of σ are in $(\frac{1}{2}\mathbb{Z})^d$, so with the above observation we conclude that any vertex in V is in $(\frac{1}{2}\mathbb{Z})^d$ as well.

To prove the other inclusion, observe that $\{2e_i \mid i \in \{1, \dots, d\}\}$ is a set of d linearly independent positive roots that have integer inner products with any element in $(\frac{1}{2}\mathbb{Z})^d$.

With both inclusions proven, we conclude that $V = (\frac{1}{2}\mathbb{Z})^d$, which is a lattice.

□

2.4.4 $\tilde{\mathbf{D}}_{\mathbf{d}}, \mathbf{d} \geqslant 4$

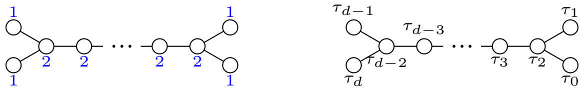


Figure 2.9: On the left: the Coxeter diagram for a triangulation of type $\tilde{D}_d$. The labels on vertices are the inverse ratios of heights. On the right: the notations of the facets of the fundamental simplex that are used in the proofs.

1. The hyperplanes that contain facets of the fundamental simplex σ can be defined as follows:

   for $\tau_0 :\; x_1 + x_2 = 1$
   for $\tau_k :\; x_k - x_{k+1} = 0, \forall k \in \{1, \dots, d-1\}$
   for $\tau_d :\; x_{d-1} + x_d = 0.$

2. From the equations of hyperplanes it is easy to check that the following points:

   $$\begin{aligned} u_0 \;\; &= \left(0^{\{d\}}\right) \\ u_1 \;\; &= \left(1, 0^{\{d-1\}}\right) \\ u_k \;\; &= \left(\tfrac{1}{2}^{\{k\}}, 0^{\{d-k\}}\right), \forall k \in \{2, \dots, d-2\} \\ u_{d-1} &= \left(\tfrac{1}{2}^{\{d-1\}}, -\tfrac{1}{2}\right) \\ u_d \;\; &= \left(\tfrac{1}{2}^{\{d\}}\right). \end{aligned}$$

   are the vertices of σ.

3. According to Figure 2.9, the minimal height falls on any of τ_k for k in $\{2, \ldots, d-2\}$. By computing, for example, the distance from u_2 to the hyperplane $x_2 - x_3 = 0$ we get:

$$h(\sigma) = \frac{1}{2\sqrt{2}}.$$

4. The circumcentre of the simplex is $\left(\frac{1}{2}, 0, \frac{1}{4}^{\{d-4\}}, \frac{1}{2}, 0\right)$ and the circumradius is $R(\sigma) = \frac{\sqrt{d+4}}{4}$, which is easily verifiable. The incentre is $\left(\frac{1}{2}, \frac{d-2}{2(d-1)}, \frac{d-3}{2(d-1)}, \ldots, \frac{1}{2(d-1)}, 0\right)$ and the inradius is $r(\sigma) = \frac{1}{2\sqrt{2}(d-1)}$, which is easily verifiable.

5. The longest edge is given by $\vec{u}_d - \vec{u}_0$ and is equal to $L(\sigma) = \frac{\sqrt{d}}{2}$.

6. The aspect ratio is:

$$\alpha(\sigma) = \frac{h(\sigma)}{R(\sigma)} = \frac{1}{\sqrt{2(d+4)}}.$$

The thickness is:

$$\theta(\sigma) = \frac{h(\sigma)}{L(\sigma)} = \frac{1}{\sqrt{2d}}.$$

The radius ratio is:

$$\rho(\sigma) = \frac{r(\sigma)}{R(\sigma)} = \frac{\sqrt{2}}{\sqrt{d+4}(d-1)}.$$

7. The volume of the simplex is:

$$vol(\sigma) = \frac{1}{d!} \det \begin{pmatrix} 1 & 1/2 & \ldots & 1/2 & 1/2 & 1/2 \\ 0 & 1/2 & & 1/2 & 1/2 & 1/2 \\ 0 & 0 & \ddots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \ddots & 1/2 & 1/2 & 1/2 \\ 0 & 0 & \ldots & 0 & 1/2 & 1/2 \\ 0 & 0 & \ldots & 0 & -1/2 & 1/2 \end{pmatrix} =$$

$$\frac{1}{2d!} \det \begin{pmatrix} 1 & 1/2 & \ldots & 1/2 & 1/2 \\ 0 & 1/2 & \ldots & 1/2 & 1/2 \\ \vdots & \ddots & \ddots & \vdots & \vdots \\ 0 & \ldots & 0 & 1/2 & 1/2 \\ 0 & \ldots & 0 & 0 & 1/2 \end{pmatrix} + \frac{1}{2d!} \det \begin{pmatrix} 1 & 1/2 & \ldots & 1/2 & 1/2 \\ 0 & 1/2 & \ldots & 1/2 & 1/2 \\ \vdots & \ddots & \ddots & \vdots & \vdots \\ 0 & \ldots & 0 & 1/2 & 1/2 \\ 0 & \ldots & 0 & 0 & 1/2 \end{pmatrix} = \frac{1}{2^{d-2}d!}.$$

The fatness of the simplex is:

$$\Theta(\sigma) = \frac{vol(\sigma)}{L(\sigma)^d} = \frac{4}{d^{d/2}d!}.$$

8. Observe that the inner products of the normal vector s_{d-2} to the hyperplane that contains τ_{d-2} with the circumcentre and u_{d-2} have different signs:

$$\langle s_{d-2}, c \rangle = -\frac{1}{4} \quad \text{and} \quad \langle s_{d-2}, u_{d-2} \rangle = \frac{1}{2}.$$

Therefore the hyperplane that contains τ_{d-2} separates the circumcentre from u_{d-2}, so by Theorem 2.3.8 the triangulation is not Delaunay.

9.

Proposition 2.4.7.

(a) The vertex set of a $\tilde{D}_4$ triangulation is a lattice.

(b) No other vertex set of a $\tilde{D}_d$ triangulation for $d \geq 5$ is a lattice.

Proof. (a) A point $v \in \mathbb{R}^d$ is a vertex of the triangulation if and only if it lies in the intersection of at least d hyperplanes in $\mathcal{H}$ (as defined in Section 1.2.4) that are linearly independent. Equivalently, the inner products of at least d linearly independent positive roots should be integers.

Let $E = \{e_1, ..., e_d\}$ be the canonical basis of $\mathbb{R}^d$. Bourbaki [Bou02, Planche IV] gives the explicit expressions of positive roots:

$$\begin{aligned} R^+ \quad = \quad & \{e_i - e_j \mid i, j \in \{1, \dots, d\} \text{ and } i < j\} \cup \\ & \{e_i + e_j \mid i, j \in \{1, \dots, d\} \text{ and } i < j\}. \end{aligned}$$

Denote by V the vertex set of the triangulation. Let $\Lambda = \{x \in (\frac{1}{2}\mathbb{Z})^4 \mid x_1 + x_2 + x_3 + x_4 \in \mathbb{Z}\}$. We will now prove that $V = \Lambda$.

We will first prove that $V \subseteq \Lambda$. This follows from the fact that Λ is stable by the action of elements of the affine Weyl group W_a. Indeed, for any $x \in \Lambda$, its image by an affine reflection $\sigma_{r,k}$ for $r \in R$ and $k \in \mathbb{Z}$, as defined in Section 1.2.4, is $\sigma_{r,k}(x) = x - (\langle x, r\rangle - k)r^\vee$. For any $r \in R$ and $x \in (\frac{1}{2}\mathbb{Z})^4$, the inner product $\langle x, r\rangle$ is in $\frac{1}{2}\mathbb{Z}$, hence $(\langle x, r\rangle - k)$ is in $\frac{1}{2}\mathbb{Z}$ as well. The coefficients of the dual root $r^\vee$ are integers, so $(\langle x, r\rangle - k)r^\vee \in (\frac{1}{2}\mathbb{Z})^4$, hence $\sigma_{r,k}(x) \in (\frac{1}{2}\mathbb{Z})^4$. By assumption, the sum of coefficients of x is an integer. For any $r \in R^+$ defined above, the sum of coefficients of $\frac{1}{2}r$ is an integer as well. Therefore $\sigma_{r,k}(x) \in \Lambda$. Now recall that one can obtain any simplex from another simplex by an element of W_a, which is generated by affine reflections $\sigma_{r,k}$. All vertices of σ are in Λ, so with the above observation we conclude that any vertex in V is in Λ as well.

To prove the other inclusion, observe that $\{e_1 + e_2, e_1 - e_2, e_3 + e_4, e_3 - e_4\}$ is a set of four linearly independent positive roots that have integer inner products with any element in Λ.

With both inclusions proven, we conclude that $V = \Lambda$, which is a lattice.

(b) Let V be the the vertex set of the triangulation. We will now prove that $v = u_3 - u_2 = \frac{1}{2}e_3$ is not in the vertex set, which implies that V does not form a lattice.

One can easily check that the positive roots that have integer inner product with v are exactly:

$$\begin{aligned} & \{e_i - e_j \mid i, j \in \{1, \dots, d\} \setminus \{3\} \text{ and } i < j\} \cup \\ & \{e_i + e_j \mid i, j \in \{1, \dots, d\} \setminus \{3\} \text{ and } i < j\} \end{aligned}$$

and these roots span the vector space $\mathrm{Vect}(E \setminus \{e_3\})$, which is $(d-1)$-dimensional. Thus, there does not exist $(r_1, ..., r_d) \subseteq R^+$ that are linearly independent and have integer inner products with v. We therefore conclude that v does not belong to V.

□

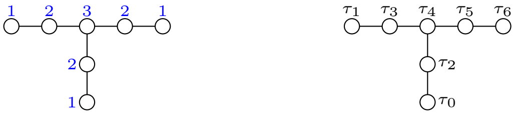


Figure 2.10: On the left: the Coxeter diagram for a triangulation of type $\tilde{E}_6$. The labels on vertices are the inverse ratios of heights. On the right: the notations of the facets of the fundamental simplex that are used in the proofs.

2.4.5 $\tilde{\mathbf{E}}_6$

1. The hyperplanes that contain facets of the fundamental simplex σ can be defined as follows:

   for $\tau_0 : \frac{1}{2}((x_1 + x_2 + x_3 + x_4 + x_5 + x_8) - (x_6 + x_7)) = 1$
   for $\tau_1 : (x_1 + x_8) - (x_2 + x_3 + x_4 + x_5 + x_6 + x_7) = 0$
   for $\tau_2 : x_1 + x_2 = 0$
   for $\tau_3 : x_1 - x_2 = 0$
   for $\tau_4 : x_2 - x_3 = 0$
   for $\tau_5 : x_3 - x_4 = 0$
   for $\tau_6 : x_4 - x_5 = 0$

2. The vertices of σ are (in $\mathbb{R}^8$) [CS87, Chapter 21]:

   $u_0 = \left(0^{\{8\}}\right)$
   $u_1 = \left(0^{\{5\}}, -\frac{2}{3}^{\{2\}}, \frac{2}{3}\right)$
   $u_2 = \left(\frac{1}{4}^{\{5\}}, -\frac{1}{4}^{\{2\}}, \frac{1}{4}\right)$
   $u_3 = \left(-\frac{1}{4}, \frac{1}{4}^{\{4\}}, -\frac{5}{12}^{\{2\}}, \frac{5}{12}\right)$
   $u_4 = \left(0^{\{2\}}, \frac{1}{3}^{\{3\}}, -\frac{1}{3}^{\{2\}}, \frac{1}{3}\right)$
   $u_5 = \left(0^{\{3\}}, \frac{1}{2}^{\{2\}}, -\frac{1}{3}^{\{2\}}, \frac{1}{3}\right)$
   $u_6 = \left(0^{\{4\}}, 1, -\frac{1}{3}^{\{2\}}, \frac{1}{3}\right)$

3. According to Figure 2.10 left, the smallest height falls on τ_4. By computing, the distance from u_4 to the hyperplane $x_2 - x_3 = 0$ we get:

   $$h(\sigma) = \frac{\sqrt{2}}{6}.$$

4. The circumcentre is $(0^{\{2\}}, -\frac{1}{6}^{\{2\}}, \frac{1}{3}, -\frac{1}{3}^{\{2\}}, \frac{1}{3})$ and the circumradius is $R(\sigma) = \frac{1}{\sqrt{2}}$, which is easily verifiable. The incentre is $(0, \frac{1}{12}, \frac{1}{6}, \frac{1}{4}, \frac{1}{3}, -\frac{1}{3}, -\frac{1}{3}, \frac{1}{3})$ and the inradius is $r(\sigma) = \frac{1}{12\sqrt{2}}$, which is easily verifiable.

5. The longest edge is $L(\sigma) = \frac{2}{\sqrt{3}}$.

6. The aspect ratio, the thickness and the radius ratio are:

   $$\alpha(\sigma) = \frac{h(\sigma)}{R(\sigma)} = \frac{1}{6} \quad \text{and} \quad \theta(\sigma) = \frac{h(\sigma)}{L(\sigma)} = \frac{1}{2\sqrt{6}} \quad \text{and} \quad \rho(\sigma) = \frac{r(\sigma)}{R(\sigma)} = \frac{1}{12}$$

7. The volume of the simplex σ is: $vol(\sigma) = \frac{\sqrt{3}}{51840}$.

   Therefore, the fatness is:

$$\Theta(\sigma) = \frac{vol(\sigma)}{L(\sigma)^6} = \frac{\sqrt{3}}{174960} \sim 9.900 \cdot 10^{-6}$$

8. Observe that the inner products of the normal vector $n = (0^{\{3\}}, 1, 0^{\{4\}})$ to the hyperplane $x_4 = 0$ with the circumcentre and the vertices have different signs: the inner product $\langle n, c \rangle = -\frac{1}{6}$ is strictly negative, whereas the inner products $\langle n, u_i \rangle$ are all non-negative. Therefore the hyperplane $x_4 = 0$ separates the circumcentre from the rest of the simplex, so by Theorem 2.3.8 the triangulation is not Delaunay.

9.

Proposition 2.4.8. *The vertex set of a $\tilde{E}_6$ triangulation does not form a lattice.*

Proof. A point $v \in \mathbb{R}^8$ is a vertex of the triangulation if and only if it lies in the intersection of at least 6 hyperplanes in $\mathcal{H}$ (as defined in Section 1.2.4) that are linearly independent. Equivalently, the inner products of at least 6 linearly independent positive roots should be integers.

Let $E = \{e_1, ..., e_8\}$ be the canonical basis of $\mathbb{R}^8$. Bourbaki [Bou02, Planche V] gives the explicit expressions of positive roots:

$$\begin{aligned} R^+ \;\; = \;\; & \{e_j - e_i \mid i,j \in \{1,\ldots,5\} \text{ and } i < j\} \cup \\ & \{e_i + e_j \mid i,j \in \{1,\ldots,5\} \text{ and } i < j\} \cup \\ & \left\{ \frac{1}{2}\left(e_8 - e_7 - e_6 + \sum_{i=1}^{5} (-1)^{\nu(i)} e_i \right) \;\middle|\; \sum_{i=1}^{5} \nu(i) \text{ is even} \right\}. \end{aligned}$$

Let V be the vertex set of the corresponding triangulation. We will now prove that $v = -2u_1 + 3u_2 + 3u_3 - 2u_4 = \frac{3}{2}e_2$ is not in the vertex set, which implies that V does not form a lattice.

One can easily check that the positive roots that have integer inner product with v are exactly:

$$\begin{aligned} & \{e_j - e_i \mid i,j \in \{1,3,4,5\} \text{ and } i < j\} \cup \\ & \{e_i + e_j \mid i,j \in \{1,3,4,5\} \text{ and } i < j\} \end{aligned}$$

and these roots span the vector space $\mathrm{Vect}(\{e_1, e_3, e_4, e_5\})$, which is 4-dimensional.

Thus, there do not exist $(r_1, ..., r_6)$ that are linearly independent and have integer inner products with v. We therefore conclude that v does not belong to V.

□

2.4.6 $\tilde{\mathbf{E}}_7$

1. The hyperplanes that contain facets of the fundamental simplex σ can be defined as follows:

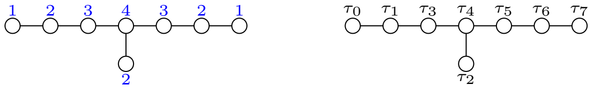


Figure 2.11: On the left: the Coxeter diagram for a triangulation of type $\tilde{E}_7$. The labels on vertices are the inverse ratios of heights. On the right: the notations of the facets of the fundamental simplex that are used in the proofs.

for $\tau_0 : x_7 - x_8 = 1$
for $\tau_1 : (x_1 + x_8) - (x_2 + x_3 + x_4 + x_5 + x_6 + x_7) = 0$
for $\tau_2 : x_1 + x_2 = 0$
for $\tau_3 : x_1 - x_2 = 0$
for $\tau_4 : x_2 - x_3 = 0$
for $\tau_5 : x_3 - x_4 = 0$
for $\tau_6 : x_4 - x_5 = 0$
for $\tau_7 : x_5 - x_6 = 0$

2. The vertices of σ are (in $\mathbb{R}^8$) [CS87, Chapter 21]:
   $u_0 = \left(0^{\{8\}}\right)$
   $u_1 = \left(0^{\{6\}}, \frac{1}{2}, -\frac{1}{2}\right)$
   $u_2 = \left(-\frac{1}{4}^{\{6\}}, \frac{1}{2}, -\frac{1}{2}\right)$
   $u_3 = \left(\frac{1}{6}, -\frac{1}{6}^{\{5\}}, \frac{1}{2}, -\frac{1}{2}\right)$
   $u_4 = \left(0^{\{2\}}, -\frac{1}{4}^{\{4\}}, \frac{1}{2}, -\frac{1}{2}\right)$
   $u_5 = \left(0^{\{3\}}, -\frac{1}{3}^{\{3\}}, \frac{1}{2}, -\frac{1}{2}\right)$
   $u_6 = \left(0^{\{4\}}, -\frac{1}{2}^{\{2\}}, \frac{1}{2}, -\frac{1}{2}\right)$
   $u_7 = \left(0^{\{5\}}, -1, \frac{1}{2}, -\frac{1}{2}\right)$

3. According to Figure 2.6 left, the smallest height falls on τ_4. By computing, the distance from u_4 to the hyperplane $x_2 - x_3 = 0$ we get:
$$h(\sigma) = \frac{\sqrt{2}}{8}.$$

4. The circumcentre is $(-\frac{1}{8}^{\{2\}}, 0^{\{3\}}, -\frac{1}{2}, \frac{1}{4}, -\frac{1}{4})$ and the circumradius is $R(\sigma) = \frac{1}{4}\sqrt{\frac{13}{2}}$, which is easily verifiable. The incentre is $(0, -\frac{1}{18}, -\frac{1}{9}, -\frac{1}{6}, -\frac{2}{9}, -\frac{5}{18}, \frac{17}{36}, -\frac{17}{36})$ and the inradius is $r(\sigma) = \frac{1}{18\sqrt{2}}$, which is easily verifiable.

5. The longest edge is $L(\sigma) = \sqrt{\frac{3}{2}}$.

6. The aspect ratio, the thickness and the radius ratio are:
$$\alpha(\sigma) = \frac{h(\sigma)}{R(\sigma)} = \frac{1}{2\sqrt{13}} \quad \text{and} \quad \theta(\sigma) = \frac{h(\sigma)}{L(\sigma)} = \frac{1}{4\sqrt{3}} \quad \text{and} \quad \rho(\sigma) = \frac{r(\sigma)}{R(\sigma)} = \frac{2}{9\sqrt{13}}$$

7. The volume of σ is: $vol(\sigma) = \frac{\sqrt{2}}{2903040}$.

Therefore, the fatness is:

$$\Theta(\sigma) = \frac{vol(\sigma)}{L(\sigma)^7} = \frac{\sqrt{3}}{14696640} \sim 1.179 \cdot 10^{-7}$$

8. Observe that the inner products of the normal vector s_4 to the hyperplane that contains τ_4 with the circumcentre and the vertices have different signs: the inner product $\langle s_4, c\rangle = -\frac{1}{8}$ is strictly negative, whereas the inner products $\langle n, u_i\rangle$ are all non-negative. Therefore the hyperplane that contains τ_4 separates the circumcentre from the rest of the simplex, so by Theorem 2.3.8 the triangulation is not Delaunay.

9.

Proposition 2.4.9. *The vertex set of a $\tilde{E}_7$ triangulation does not form a lattice.*

Proof. A point $v \in \mathbb{R}^8$ is a vertex of the triangulation if and only if it lies in the intersection of at least 7 hyperplanes in $\mathcal{H}$ (as defined in Section 1.2.4) that are linearly independent. Equivalently, the inner products of at least 7 linearly independent positive roots should be integers.

Let $E = \{e_1, ..., e_8\}$ be the canonical basis of $\mathbb{R}^8$. Bourbaki [Bou02, Planche VI] gives the explicit expressions of positive roots:

$$\begin{aligned} R^+ \;=\; & \{e_j - e_i \mid i,j \in \{1,\dots,6\} \text{ and } i<j\} \cup \\ & \{e_i + e_j \mid i,j \in \{1,\dots,6\} \text{ and } i<j\} \cup \\ & \left\{ \frac{1}{2}\left(e_8 - e_7 + \sum_{i=1}^{6} (-1)^{\nu(i)} e_i \right) \;\middle|\; \sum_{i=1}^{6} \nu(i) \text{ is odd} \right\}. \end{aligned}$$

Let V be the vertex set of the corresponding triangulation. We will now prove that $v = -6u_1 + 2u_2 + 3u_3 - 4u_4 = e_2$ is not in the vertex set, which implies that V does not form a lattice.

One can easily check that the positive roots that have integer inner product with v are exactly:

$$\begin{aligned} & \{e_j - e_i \mid i,j \in \{1,\dots,6\} \text{ and } i<j\} \cup \\ & \{e_i + e_j \mid i,j \in \{1,\dots,6\} \text{ and } i<j\} \end{aligned}$$

and these roots span the vector space $\mathrm{Vect}(\{e_1, ..., e_6\})$, which is 6-dimensional. Thus, there does not exist $(r_1, ..., r_7) \subseteq R^+$ that are linearly independent and have integer inner products with v. We therefore conclude that v does not belong to V. □

2.4.7 $\tilde{\mathbf{E}}_8$

1. The hyperplanes that contain facets of the fundamental simplex σ can be defined as follows:

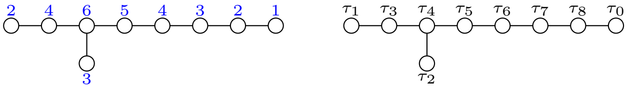


Figure 2.12: On the left: the Coxeter diagram for a triangulation of type $\tilde{E}_8$. The labels on vertices are the inverse ratios of heights. On the right: the notations of the facets of the fundamental simplex that are used in the proofs.

for $\tau_0 : x_7 + x_8 = 1$
for $\tau_1 : (x_1 + x_8) - (x_2 + x_3 + x_4 + x_5 + x_6 + x_7) = 0$
for $\tau_2 : x_1 + x_2 = 0$
for $\tau_3 : x_1 - x_2 = 0$
for $\tau_4 : x_2 - x_3 = 0$
for $\tau_5 : x_3 - x_4 = 0$
for $\tau_6 : x_4 - x_5 = 0$
for $\tau_7 : x_5 - x_6 = 0$
for $\tau_8 : x_6 - x_7 = 0$

2. The vertices of σ are [CS87, Chapter 21]:

$$\begin{aligned}
u_0 &= \left(0^{\{8\}}\right)\\
u_1 &= \left(0^{\{7\}}, 1\right)\\
u_2 &= \left(\tfrac{1}{6}^{\{7\}}, \tfrac{5}{6}\right)\\
u_3 &= \left(-\tfrac{1}{8}, \tfrac{1}{8}^{\{6\}}, \tfrac{7}{8}\right)\\
u_4 &= \left(0^{\{2\}}, \tfrac{1}{6}^{\{5\}}, \tfrac{5}{6}\right)\\
u_5 &= \left(0^{\{3\}}, \tfrac{1}{5}^{\{4\}}, \tfrac{4}{5}\right)\\
u_6 &= \left(0^{\{4\}}, \tfrac{1}{4}^{\{3\}}, \tfrac{3}{4}\right)\\
u_7 &= \left(0^{\{5\}}, \tfrac{1}{3}^{\{2\}}, \tfrac{2}{3}\right)\\
u_8 &= \left(0^{\{6\}}, \tfrac{1}{2}^{\{2\}}\right)
\end{aligned}$$

3. The circumcentre is $(-\frac{1}{12}^{2}, 0^{5}, \frac{1}{2})$ and the circumradius is $R(\sigma) = \frac{\sqrt{38}}{12}$, which is easily verifiable. The incentre is $(0, \frac{1}{30}, \frac{1}{15}, \frac{1}{10}, \frac{2}{15}, \frac{1}{6}, \frac{1}{5}, \frac{23}{30})$ and the inradius is $r(\sigma) = \frac{1}{30\sqrt{2}}$, which is easily verifiable.

4. According to Figure 2.6 left, the smallest height falls on τ_4. By computing, the distance from u_4 to the hyperplane $x_2 - x_3 = 0$ we get:

$$h(\sigma) = \frac{\sqrt{2}}{12}.$$

5. The longest edge is $L(\sigma) = 1$.

6. The aspect ratio, the thickness and the radius ratio are:

$$\alpha(\sigma) = \frac{h(\sigma)}{R(\sigma)} = \frac{1}{2\sqrt{19}} \quad \text{and} \quad \theta(\sigma) = \frac{h(\sigma)}{L(\sigma)} = \frac{1}{6\sqrt{2}} \quad \text{and} \quad \rho(\sigma) = \frac{r(\sigma)}{R(\sigma)} = \frac{1}{5\sqrt{19}}$$

7. The volume of σ is: $vol(\sigma) = \frac{1}{696729600} \sim 1.435 \cdot 10^{-9}$.

   Therefore the fatness is:

$$\Theta(\sigma) = \frac{vol(\sigma)}{L(\sigma)^8} = \frac{1}{696729600} \sim 1.435 \cdot 10^{-9}$$

8. Observe that the inner products of the normal vector $n = (0, 1, 0^{\{6\}})$ to the hyperplane $x_2 = 0$ with the circumcentre and the vertices have different signs: the inner product $\langle n, c \rangle = -\frac{1}{12}$ is strictly negative, whereas the inner products $\langle n, u_i \rangle$ are all non-negative. Therefore the hyperplane $x_2 = 0$ separates the circumcentre from the rest of the simplex, so by Theorem 2.3.8 the triangulation is not Delaunay.

9.

Proposition 2.4.10. *The vertex set of a $\tilde{E}_8$ triangulation does not form a lattice.*

Proof. A point $v \in \mathbb{R}^8$ is a vertex of the triangulation if and only if it lies in the intersection of at least 8 hyperplanes in $\mathcal{H}$ (as defined in Section 1.2.4) that are linearly independent. Equivalently, the inner products of at least 8 linearly independent positive roots should be integers.

Let $E = \{e_1, ..., e_8\}$ be the canonical basis of $\mathbb{R}^8$. Bourbaki [Bou02, Planche VII] gives the explicit expressions of positive roots:

$$\begin{aligned} R^+ \quad = \quad & \{e_j - e_i \mid i, j \in \{1, \ldots, 7\} \text{ and } i < j\} \cup \\ & \{e_i + e_j \mid i, j \in \{1, \ldots, 7\} \text{ and } i < j\} \cup \\ & \left\{ \frac{1}{2} \left(e_8 + \sum_{i=1}^{7} (-1)^{\nu(i)} e_i \right) \,\middle|\, \sum_{i=1}^{7} \nu(i) \text{ is even} \right\}. \end{aligned}$$

Let V be the vertex set of the corresponding triangulation. We will now prove that $v = -u_1 + 3u_2 + 4u_3 - 6u_4 = e_2$ is not in the vertex set, which implies that V does not form a lattice.

One can easily check that the positive roots that have integer inner product with v are exactly:

$$\begin{aligned} & \{e_j - e_i \mid i, j \in \{1, \ldots, 7\} \text{ and } i < j\} \cup \\ & \{e_i + e_j \mid i, j \in \{1, \ldots, 7\} \text{ and } i < j\} \end{aligned}$$

and these roots span the vector space Vect$(\{e_1, ..., e_7\})$, which is 7-dimensional. Thus, there does not exist $(r_1, ..., r_8) \subseteq R^+$ that are linearly independent and have integer inner products with v. We therefore conclude that v does not belong to V.

□

2.4.8 $\tilde{\mathbf{F}}_4$

1. The hyperplanes that contain facets of the fundamental simplex σ can be defined as follows:

   for $\tau_0 : x_1 + x_2 = 1$
   for $\tau_1 : x_2 - x_3 = 0,$
   for $\tau_2 : x_3 - x_4 = 0,$
   for $\tau_3 : x_4 = 0$
   for $\tau_4 : x_1 - x_2 - x_3 - x_4 = 0$

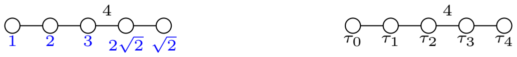


Figure 2.13: On the left: the Coxeter diagram for a triangulation of type $\tilde{F}_4$. The labels on vertices are the inverse ratios of heights. On the right: the notations of the facets of the fundamental simplex that are used in the proofs.

2. From the equations of hyperplanes it is easy to check that the following points:
$$\begin{array}{l} u_0 = (0,0,0,0) \\ u_1 = \left(\frac{1}{2},\frac{1}{2},0,0\right) \\ u_2 = \left(\frac{2}{3},\frac{1}{3},\frac{1}{3},0\right) \\ u_3 = \left(\frac{3}{4},\frac{1}{4},\frac{1}{4},\frac{1}{4}\right) \\ u_4 = (1,0,0,0) \end{array} \quad \text{are the vertices of } \sigma.$$

3. The inverse height proportions in Figure 2.13 left suggest that the smallest height corresponds to τ_2. By computing, the distance from u_2 to the hyperplane $x_3 - x_4 = 0$ we get:
$$h(\sigma) = \frac{\sqrt{2}}{6}.$$

4. The circumcentre of the simplex is $(\frac{1}{2},0,0,0)$ and the circumradius is $R(\sigma) = \frac{1}{2}$, which is easily verifiable. The incentre is $\left(\frac{6+5\sqrt{2}}{6(2+\sqrt{2})}, \frac{4+\sqrt{2}}{6(2+\sqrt{2})}, \frac{1}{6}, \frac{\sqrt{2}}{6(2+\sqrt{2})}\right)$ and the inradius is $r(\sigma) = \frac{\sqrt{2}}{6(2+\sqrt{2})}$, which is easily verifiable.

5. The longest edge is given by $\vec{u}_4 - \vec{u}_0$ and is equal to $L(\sigma) = 1$.

6. The aspect ratio, the thickness and the radius ratio are:
$$\alpha(\sigma) = \frac{h(\sigma)}{R(\sigma)} = \frac{\sqrt{2}}{6} \quad \text{and} \quad \theta(\sigma) = \frac{h(\sigma)}{L(\sigma)} = \frac{\sqrt{2}}{6} \quad \text{and} \quad \rho(\sigma) = \frac{r(\sigma)}{R(\sigma)} = \frac{\sqrt{2}}{3(2+\sqrt{2})}$$

7. The volume of σ is: $vol(\sigma) = \frac{1}{576} = 0.00174$.

   Therefore the fatness is:
$$\Theta(\sigma) = \frac{vol(\sigma)}{L(\sigma)^4} = \frac{1}{576} \sim 0.00174$$

8. Observe that all inner products of the normals s_i to hyperplanes that contain τ_i and the corresponding opposite vertices u_i are positive. Observe as well that the inner products with the circumcentre $\langle s_i, c\rangle$ are either positive or zero. It implies that the circumcentre lies on the boundary of the simplex, therefore by Theorem 2.3.8 the triangulation is non-protected Delaunay.

9.

Proposition 2.4.11. *The vertex set of a $\tilde{F}_4$ triangulation does not form a lattice.*

Proof. A point $v \in \mathbb{R}^4$ is a vertex of the triangulation if and only if it lies in the intersection of at least 4 hyperplanes in $\mathcal{H}$ (as defined in Section 1.2.4) that are linearly independent.

Equivalently, the inner products of at least 4 linearly independent positive roots should be integers.

Let $E = \{e_1, ..., e_4\}$ be the canonical basis of $\mathbb{R}^4$. Bourbaki [Bou02, Planche VIII] gives the explicit expressions of positive roots:

$$R^+ = E \cup \{e_i \pm e_j \mid i,j \in \{1,\dots,4\} \text{ and } i<j\} \cup \left\{\frac{1}{2}\left(e_1 \pm e_2 \pm e_3 \pm e_4\right)\right\}.$$

Let V be the vertex set of the corresponding triangulation. We will now prove that $v = u_1 + u_2 = \left(\frac{7}{6}, \frac{5}{6}, \frac{1}{3}, 0\right)$ is not in the vertex set, which implies that V does not form a lattice.

One can easily check that the positive roots that have integer inner product with v are exactly

$$\left\{e_4, e_1 + e_2, \frac{1}{2}(e_1 - e_2 - e_3 \pm e_4)\right\}.$$

Note that $\frac{1}{2}(e_1 - e_2 - e_3 + e_4) = \frac{1}{2}(e_1 - e_2 - e_3 - e_4) + e_4$, so these roots are not linearly independent. Thus, there does not exist $(r_1, ..., r_4) \subseteq R^+$ that are linearly independent and have integer inner products with v. We therefore conclude that v does not belong to V.

□

2.4.9 $\tilde{\mathbf{G}}_2$ $\circ\overset{6}{-}\circ-\circ$

Simplex of $\tilde{G}_2$ triangulation (also called *Kisrhombille tiling*) is the right triangle with $\pi/6$ angle. By an easy computation, the aspect ratio, thickness and radius ratio are as follows:

$$\alpha(\sigma) = \theta(\sigma) = \frac{\sqrt{3}}{4} \quad \text{and} \quad \rho(\sigma) = \frac{1}{1+\sqrt{3}}$$

The fatness is $\Theta(\sigma) = \frac{\sqrt{3}}{8}$.

The circumcentre of any triangle lies on its hypotenuse, therefore by Theorem 2.3.8 the triangulation is non-protected Delaunay.

It is obvious from Figure 1.10 that the vertex set of the triangulation does not form a lattice.

2.5 Protection value of a triangulation of type $\tilde{A}_d$

In this section we will prove the bound on the protection stated in Theorem 2.2.1.

Lemma 2.5.1. *A Coxeter triangulation of type $\tilde{A}_d$ has a relative protection equal to*

$$\hat{\delta} = \frac{\sqrt{d^2 + 2d + 24} - \sqrt{d^2 + 2d}}{\sqrt{d^2 + 2d}} \sim \frac{12}{d^2}.$$

Proof. Let V be the vertex set of the triangulation containing the simplex in the $\tilde{A}_d$ row of Table 2.3. By Proposition 2.4.2 in Section 2.4, V coincides with the coweight lattice, which corresponds to a root system R of type A_d.

Take the following basis vectors of the coweight lattice V:

$$\vec{u}_k = \frac{1}{d+1}\left((k-(d+1))^{\{k\}}, k^{\{d+1-k\}}\right), \text{ for } 0 \le k \le d.$$

This means that any point of V can be expressed as $\sum_{k=1}^{d} \alpha_k \vec{u}_k$, with $\alpha_k \in \mathbb{Z}$ for all $k \in \{1,\dots,d\}$. Each coordinate of $(d+1)\vec{u}_k$ is an integer congruent to k modulo $(d+1)$. Following the definition of [ABD10], we call a point a *remainder-k point* if its coordinates are congruent to k modulo $(d+1)$. Since any lattice point z in V can be written as $z = \sum \alpha_k \vec{u}_k$, it follows that z is a $\tilde{k}$-remainder point, where $\tilde{k}$ is congruent to $\left(\sum \alpha_k \cdot k\right)$ modulo $(d+1)$.

We will now look at the circumcentre $c = \frac{1}{d+1}\left(-\frac{d}{2}, -\frac{d-2}{2}, -\frac{d-4}{2}, \dots, \frac{d}{2}\right)$ of σ and find its nearest neighbours in the lattice. Take

$$z = \frac{1}{d+1}\left(n_1(d+1)+\tilde{k}, \cdots, n_{d+1}(d+1)+\tilde{k}\right),$$

a remainder-$\tilde{k}$ lattice point, for some $\tilde{k} \in \{0,\dots,d\}$. Note that because the lattice V resides in the hyperplane $\sum_{i=1}^{d+1} x_i = 0$, we have $\left\langle z, (1^{\{d+1\}})\right\rangle = 0$, and therefore $\sum n_i = -\tilde{k}$. We need to find the value of $n = (n_1, \cdots, n_{d+1}) \in N = \left\{x \in \mathbb{Z}^{d+1} \middle| \sum x_i = -\tilde{k}\right\}$ which minimizes $d(z,c)$ for a given $\tilde{k}$. We now state the following claim, which will be proved below.

Claim 1. *For each $k \in \mathbb{Z}/(d+1)\mathbb{Z}$, the closest remainder-$k$ point in V to c is u_k. A remainder-k point that is the second closest corresponds to the vector n equal to*

$$\begin{cases} (-1^{\{k-1\}}, 0, -1, 0^{\{d-k\}}), & \textit{for } k \in \{1,...,d\} \\ (-1, 0^{\{d-1\}}, 1), & \textit{if } k = 0 \end{cases}$$

and has the coordinates

$$\begin{cases} ((k-(d+1))^{\{k-1\}}, k, k-(d+1), k^{\{d-k\}}), & \textit{for } k \in \{1,...,d\} \\ (-(d+1), 0^{\{d-1\}}, d+1), & \textit{if } k = 0 \end{cases}.$$

Each of these points are at a distance

$$R' = \sqrt{\frac{d(d+2)}{12(d+1)} + \frac{2}{d+1}}$$

from c.

This claim implies that the protection radius of σ is

$$R' - R = \sqrt{\frac{d(d+2)}{12(d+1)} + \frac{2}{d+1}} - \sqrt{\frac{d(d+2)}{12(d+1)}}.$$

Thus, the relative protection is

$$\hat{\delta} = \frac{\sqrt{d^2+2d+24} - \sqrt{d^2+2d}}{\sqrt{d^2+2d}} = \sqrt{1 + \frac{24}{d^2+2d}} - 1 \sim \left(1 + \frac{12}{d^2+2d}\right) - 1 \sim \frac{12}{d^2}.$$

This completes the proof of Lemma 2.5.1 apart from the claim, which we will focus on now. □

The main ingredient for the proof of the Claim is the following lemma.

Lemma 2.5.2. *The difference between the two squared distances $d(z,c)^2$ and $d(z',c)^2$ for two points z and z' of the lattice V is of the form $\frac{2q}{d+1}$ where q is an integer.*

Proof. It is enough to compare the distances for z and z' of the same congruence class. Indeed, each congruence class k already contains the point u_k, which is of squared distance R^2 from c. The difference between any two squared distances $d(z,c)^2$ and $d(z',c)^2$, for z and z' a remainder-k and a remainder-k' point respectively, can be expressed as

$$d(z,c)^2 - d(z',c)^2 = (d(z,c)^2 - d(z,u_k)^2) - (d(z',c)^2 - d(z',u_{k'})'^2).$$

If both differences are in $\frac{2}{d+1}\mathbb{Z}$, the overall difference is in $\frac{2}{d+1}\mathbb{Z}$ as well.

So let:

$$\begin{aligned} z &= \frac{1}{d+1}\big(n_0(d+1)+k, \cdots, n_d(d+1)+k\big) \text{ and} \\ z' &= \frac{1}{d+1}\big(n'_0(d+1)+k, \cdots, n'_d(d+1)+k\big) \end{aligned}$$

be two remainder-k points of V, with n_i and n'_i integers for all $i \in \{1, \dots, d+1\}$. The difference between $d(z,c)^2$ and $d(z',c)^2$ can be expressed as:

$$\begin{aligned} d(z,c)^2 - d(z',c)^2 &= (\|z\|^2 - 2\langle z,c\rangle + \|c\|^2) - (\|z'\|^2 - 2\langle z',c\rangle + \|c\|^2) \\ &= \|z\|^2 - \|z'\|^2 - 2\langle z - z', c\rangle \\ &= \sum_{i=0}^{d}\left(n_i + \frac{k}{d+1}\right)^2 - \sum_{i=0}^{d}\left(n'_i + \frac{k}{d+1}\right)^2 \\ &\quad -2\sum_{i=0}^{d}\left(n_i + \frac{k}{d+1} - n'_i - \frac{k}{d+1}\right)\left(-\frac{d}{2(d+1)} + \frac{i}{d+1}\right) \\ &= \sum_{i=0}^{d}(n_i - n'_i)\left(n_i + n'_i + \frac{2k}{d+1}\right) + 2\sum_{i=0}^{d}(n_i - n'_i)\left(\frac{d/2 - i}{d+1}\right) \\ &= \sum_{i=0}^{d}(n_i - n'_i)\left(n_i + n'_i + \frac{2k + d - 2i}{d+1}\right) \end{aligned}$$

Because $\sum_{i=0}^{d} n_i = -k$, we can express $n_0 = -k - \sum_{i=1}^{d} n_i$. Therefore:

$$\begin{aligned} d(z,c)^2 - d(z',c)^2 &= \sum_{i=1}^{d}(n_i - n'_i)\left(n_i + n'_i + \frac{2k + d - 2i}{d+1}\right) \\ &\quad +(n_0 - n'_0)\left(n_0 + n'_0 + \frac{2k+d}{d+1}\right) \\ &= \sum_{i=1}^{d}(n_i - n'_i)\left(n_i + n'_i + \frac{2k + d - 2i}{d+1}\right) \\ &\quad -\left(\sum_{i=1}^{d}(n_i - n'_i)\right)\left(-2k - \sum_{i=1}^{d}(n_i + n'_i) + \frac{2k+d}{d+1}\right) \end{aligned}$$

The last expression is further simplified to

$$\sum_{i=1}^{d}(n_i - n'_i)\left(n_i + n'_i + 2k + \sum_{j=1}^{d}(n_j + n'_j) - \frac{2i}{d+1}\right).$$

We will split this expression into two parts:

$$\sum_{i=1}^{d}(n_i - n_i') \left(n_i + n_i' + \sum_{j=1}^{d}(n_j + n_j') \right) + \frac{2}{d+1}\sum_{i=1}^{d}(n_i - n_i')((d+1)k - i). \tag{2.7}$$

The second part of the expression (2.7) is already of the form $\frac{2q'}{d+1}$ with q' an integer, so we will be interested in the first part.

Let $a_i \in \mathbb{Z}/2\mathbb{Z}$ be the congruence class of $(n_i - n_i')$ modulo 2. At the same time, a_i is the congruence class of $(n_i + n_i')$ modulo 2. So the first part of the expression has the congruence class modulo 2 as follows:

$$\sum_{i=1}^{d} a_i \left(a_i + \sum_{j=1}^{d} a_j \right) = \sum_{i=1}^{d} a_i^2 + \left(\sum_{i=1}^{d} a_i \right)^2 = 2\sum_{i=1}^{d}\sum_{\substack{j=0 \\ j\neq i}}^{d} a_i a_j = 0.$$

It implies that the first part of (2.7) is an even integer and it can be represented as $\frac{2q''}{d+1}$ for some integer q''. Thus, the difference $d(z,c)^2 - d(z',c)^2$ can be expressed as $\frac{2q}{d+1}$ for some integer q. □

Proof (of Claim). A consequence of Lemma 2.5.2 is that the minimal positive difference between two squared distances is $\frac{2}{d+1}$. By Theorem 2.3.8 we know that the triangulation has a non-zero protection, so out of all points in V only the vertices of σ are distant from c by the circumradius R, and they are unique to be so in their respective congruence class. The squared distance R'^2 between the points in the statement of the Claim and the circumcentre c is exactly different from R^2 by $\frac{2}{d+1}$, so these points are indeed second closest points to c after the vertices of σ, which concludes the proofs of the Claim and Lemma 2.5.1.

□

Remark 2.5.3. *One can verify that the points in the statement of the Claim are the only second closest points to c in V, but it goes beyond the scope of this thesis.*

2.6 Numerical values of quality measures of simplices in Coxeter triangulations

In Tables 2.4, 2.5, 2.6 and 2.7 we present the numerical values of the normalized aspect ratio, radius ratio, fatness and thickness respectively.

$d =$	2	3	4	5	6	7	8
$\tilde{A}_d$	1.000	0.949	0.894	0.845	0.802	0.764	0.730
$\tilde{B}_d$	-	0.474	0.462	0.445	0.429	0.412	0.398
$\tilde{C}_d$	0.666	0.612	0.566	0.527	0.495	0.468	0.444
$\tilde{D}_d$	-	-	0.400	0.393	0.383	0.371	0.363
$\tilde{E}_d$	-	-	-	-	0.286	0.243	0.204
$\tilde{F}_d$	-	-	0.377	-	-	-	-
$\tilde{G}_d$	0.577	-	-	-	-	-	-

Table 2.4: A comparative table showing the normalized aspect ratio $\hat{\alpha}$ value for Coxeter triangulations.

$d =$	2	3	4	5	6	7	8
$\tilde{A}_d$	1.000	0.949	0.894	0.845	0.802	0.764	0.730
$\tilde{B}_d$	-	0.701	0.623	0.568	0.526	0.492	0.464
$\tilde{C}_d$	0.828	0.717	0.641	0.584	0.540	0.505	0.475
$\tilde{D}_d$	-	-	0.667	0.589	0.537	0.497	0.467
$\tilde{E}_d$	-	-	-	-	0.500	0.431	0.367
$\tilde{F}_d$	-	-	0.553	-	-	-	-
$\tilde{G}_d$	0.732	-	-	-	-	-	-

Table 2.5: A comparative table showing the normalized radius ratio $\hat{\rho}$ value for Coxeter triangulations.

$d =$	2	3	4	5	6	7	8
$\tilde{A}_d$	1.000	0.891	0.864	0.807	0.781	0.743	0.721
$\tilde{B}_d$	-	0.816	0.688	0.607	0.551	0.509	0.475
$\tilde{C}_d$	0.760	0.648	0.579	0.529	0.491	0.461	0.436
$\tilde{D}_d$	-	-	0.818	0.697	0.619	0.562	0.518
$\tilde{E}_d$	-	-	-	-	0.528	0.422	0.363
$\tilde{F}_d$	-	-	0.472	-	-	-	-
$\tilde{G}_d$	0.707	-	-	-	-	-	-

Table 2.6: A comparative table showing the normalized fatness $\hat{\Theta}^{1/d}$ value for Coxeter triangulations.

$d =$	2	3	4	5	6	7	8
$\tilde{A}_d$	1.000	0.866	0.816	0.745	0.707	0.661	0.632
$\tilde{B}_d$	-	0.500	0.447	0.408	0.378	0.354	0.333
$\tilde{C}_d$	0.577	0.500	0.447	0.408	0.378	0.354	0.333
$\tilde{D}_d$	-	-	0.447	0.408	0.378	0.354	0.333
$\tilde{E}_d$	-	-	-	-	0.268	0.191	0.153
$\tilde{F}_d$	-	-	0.298	-	-	-	-
$\tilde{G}_d$	0.500	-	-	-	-	-	-

Table 2.7: A comparative table showing the normalized thickness $\hat{\theta}$ value for Coxeter triangulations.

Chapter 3

Freudenthal-Kuhn triangulation of $\mathbb{R}^d$

In this chapter, we will define the *Freudenthal-Kuhn triangulation* of the Euclidean space $\mathbb{R}^d$. Freudenthal-Kuhn triangulations were invented independently by Freudenthal [Fre42] and Kuhn [Kuh60]. These triangulations are also known as *K1 triangulations* [Tod76], *Freudenthal's triangulations* [Eav84, Dan95, EK12], *Kuhn's triangulations* [Moo92]. These are not independent from Coxeter triangulations of type $\tilde{A}$ that we discussed in the previous chapters. In fact, the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ is identical to Coxeter triangulations of type $\tilde{A}_d$ up to a linear transformation (see Theorem 3.1.7 and Figure 3.1 below). For this reason, Freudenthal-Kuhn triangulations are also called *Coxeter-Freudenthal-Kuhn's triangulations* [Hen07].

Our main motivation for the study of Freudenthal-Kuhn triangulations is that the following queries can be answered in a time- and space-efficient manner:

Point location: Compute the simplex in the triangulation that contains a given point.

Face generation: Generate the faces of a specific dimension of a given simplex in the triangulation.

Coface generation: Generate the cofaces of a specific dimension of a given simplex in the triangulation.

We introduce the algorithms that answer these three queries in Sections 3.4 and 3.5. These

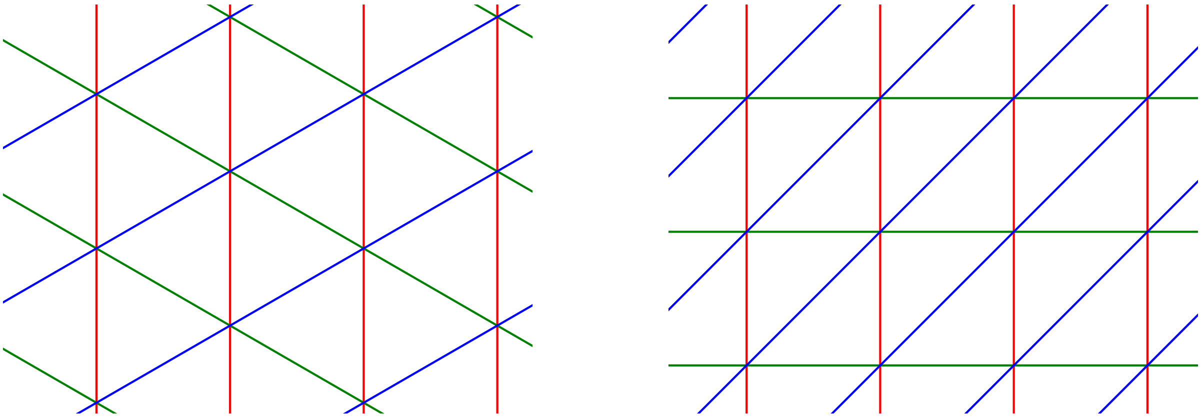


Figure 3.1: Coxeter triangulation of type $\tilde{A}_2$ (left) and the Freudenthal-Kuhn triangulation of $\mathbb{R}^2$ (right).

algorithms are an essential part of the data structure in Chapter 4 that we use for the manifold tracing algorithm.

Freudenthal-Kuhn triangulations were studied extensively in the past. In addition to describing the known results on Freudenthal-Kuhn triangulations, we introduce in this chapter new definitions and algorithms. Here is our two main contributions in the chapter:

- Freudenthal [Fre42] defined a representation for every d-dimensional simplex that lies in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$. Inspired by the work of Eaves [Eav84] and by the description of the facial structure of a permutahedron in Section 1.3, we generalize the representation of d-dimensional simplices in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ to simplices of arbitrary dimension. We call the new representation of simplices of arbitrary dimension the *permutahedral representation*.

- Kuhn [Kuh68] provided two algorithms that use the Freudenthal representations of d-dimensional simplices in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$: point location and adjacent d-dimensional simplex computation. We introduce the face generation algorithm and the coface generation algorithm that use the permutahedral representation of simplices of arbitrary dimension. We use these two algorithms in Chapter 4 for our manifold tracing algorithm.

The chapter is organized as follows. In Section 3.1, we define the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ and the Freudenthal representation of its d-dimensional simplices. In Section 3.1.1, we describe the two algorithms by Kuhn [Kuh68]: point location and adjacent d-dimensional simplex computation. In Section 3.2, we define Eaves notation. Eaves notation is an important intermediate step for generalizing the Freudenthal representation for d-dimensional simplices to the permutahedral representation for simplices of arbitrary dimension. We define the permutahedral representation in Section 3.3. In Section 3.4, we generalize the point location operation in Section 3.1.1 to take into account lower-dimensional simplices using the introduced permutahedral representation. Lastly, we introduce the face and coface generation algorithms for simplices in Freudenthal-Kuhn triangulations in Section 3.5.

3.1 Definition of Freudenthal-Kuhn triangulation of $\mathbb{R}^d$

Although Coxeter triangulations of type $\tilde{A}_d$ and Freudenthal-Kuhn triangulations are identical up to a linear transformation, they are defined in very different ways. Recall that Coxeter triangulations are defined as hyperplane arrangements (see Corollary 1.2.32). In contrast, Freudenthal-Kuhn triangulations are defined using two things:

- The *unit cubical partition* of the Euclidean space $\mathbb{R}^d$ (defined in Definition 3.1.1 below).

- A specific triangulation, in which each cube in the unit cubical partition is subdivided.

The Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ is the union of the triangulations of all cubes in the unit cubical partition. We will now define the unit cubical partition and the triangulation of the cubes.

Unit cubical partitions We start with the definition of the unit cube and the unit cubical partition of the Euclidean space $\mathbb{R}^d$.

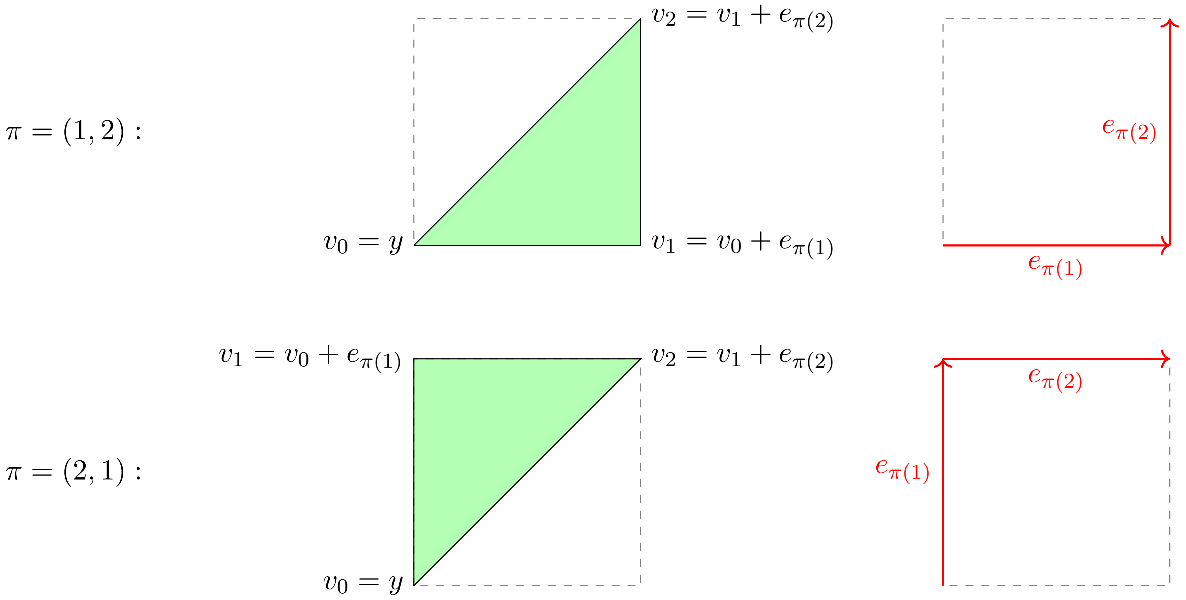


Figure 3.2: Two 2-dimensional simplices $\hat{\sigma}(y, \pi)$ (in green) for a fixed y and two possible values of π. The unit cube to which $\hat{\sigma}(y, \pi)$ belongs is shown in dashed lines. On the right: the corresponding paths (in red) along the edges of the unit cube (see Remark 3.1.6).

Definition 3.1.1 (Unit cubical partition)**.** *The* d-dimensional unit cube *is the polytope defined as the Cartesian product of d unit intervals:* $[0, 1]^d$.

The unit cubical partition *of the Euclidean space $\mathbb{R}^d$ is a cell complex that consists of copies of the unit cube and its faces translated along the integer lattice $\mathbb{Z}^d$.*

Remark 3.1.2. *Note that all vertices in a unit cubical partition have integer Cartesian coordinates.*

We will refer to a d-dimensional cube in the unit cubical partition of the Euclidean space $\mathbb{R}^d$ by its vertex that has the smallest coordinates with respect to each canonical basis vector. For example, the three-dimensional cube defined as the Cartesian product of three intervals $[4, 5] \times [2, 3] \times [-1, 0]$ will be referred by three coordinates $(4, 2, -1)$.

Freudenthal representation. We will now define the triangulations of the cubes in the unit cubical partition. For this, we will define each of the d-dimensional simplices in the triangulation by parametrizing them with a permutation [Fre42].

Definition 3.1.3 (Freudenthal representation)**.** *Let $y \in \mathbb{Z}^d$. Let π be a permutation of* $\{1, \dots, d\}$. *The d-dimensional simplex $\hat{\sigma}(y, \pi)$ is the convex hull of $d+1$ vertices $v_0, \dots, v_d$ defined as follows:*

$$\begin{aligned} v_0 &= y \\ v_i &= v_{i-1} + e_{\pi(i)}, \quad \text{for } i \in \{1, \dots, d\}. \end{aligned}$$

Here $e_{\pi(i)}$ stands for the $\pi(i)$-th vector in the canonical basis of $\mathbb{R}^d$. We will refer to the notation $\hat{\sigma}(y, \pi)$ as the Freudenthal representation.

Remark 3.1.4. *Note that the size complexity of storing the Freudenthal representation of a simplex is $2d$. This complexity is smaller than the size complexity $O(d^2)$ of storing the Cartesian coordinates of the $d+1$ vertices.*

Remark 3.1.5. *In the following, we adopt the following convention. We put a hat on the notations of simplices that are exclusive for d-dimensional simplices (the Freudenthal representation $\hat{\sigma}(y,\pi)$ above and Eaves notations $\hat{\varphi}(y,\pi)$ in Section 3.2). The corresponding notations without a hat (generalized Eaves notation $\varphi(y,\omega)$ and the permutahedral representation $\sigma(y,\omega)$ in Section 3.3.6) indicate the respective generalizations to simplices of arbitrary dimension.*

Two two-dimensional simplices given by their Freudenthal representations are illustrated in Figure 3.2. The vertex y in a Freudenthal representation $\sigma(y,\pi)$ indicates the cube in the unit cubical partition of $\mathbb{R}^d$, in which the d-dimensional simplex lies. The permutation π identifies the specific d-dimensional simplex in the triangulation of the cube. In the following, we will use the tuple $(\pi(1),\ldots,\pi(d))$ to represent a permutation π.

Remark 3.1.6 (Freudenthal representation and paths on a cube)**.** *The permutation π can be visualized as a path in the unit cube along the vectors $e_{\pi(1)},\ldots,e_{\pi(d)}$ (see Figure 3.2 right).*

Freudenthal-Kuhn triangulation of $\mathbb{R}^d$. A classical result is that the simplices $\hat{\sigma}(y,\pi)$ define a triangulation of $\mathbb{R}^d$.

Theorem 3.1.7. *Let $\mathcal{F}$ be the set of d-dimensional simplices $\hat{\sigma}(y,\pi)$ for all points $y \in \mathbb{Z}^d$ and permutations $\pi : \{1,\ldots,d\} \to \{1,\ldots,d\}$. The simplices in $\mathcal{F}$ and their faces define a triangulation of $\mathbb{R}^d$.*

A proof of Theorem 3.1.7 can be found, for example, in Todd [Tod76, Chapter III, Lemma 3.2].

Definition 3.1.8. *The triangulation in Theorem 3.1.7 is called the* Freudenthal-Kuhn triangulation of $\mathbb{R}^d$.

Remark 3.1.9. *Note that different d-dimensional simplices in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ have different Freudenthal representations. Combined with Theorem 3.1.7, it follows that the Freudenthal representations $\hat{\sigma}(y,\pi)$ for all points $y \in \mathbb{Z}^d$ and permutations $\pi : \{1,\ldots,d\} \to \{1,\ldots,d\}$ are in one-to-one correspondence with the d-dimensional simplices in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$.*

3.1.1 Operations using the Freudenthal representation

In this section, we will give an overview of the operations in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ that were introduced by Kuhn [Kuh68]. These operations are *point location* and *adjacent simplex computation.*

Point location. The first operation introduced by Kuhn [Kuh68] is point location. This operation takes as input an array containing the Cartesian coordinates of a point

$$x = (x_1,\ldots,x_d) \in \mathbb{R}^d.$$

It then outputs the Freudenthal representation $\hat{\sigma}(y,\pi)$ of a d-dimensional simplex lying in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ that contains the point x. The point location consists of two steps.

1. The integer part $\lfloor x_i \rfloor$ and the fractional part $\{x_i\}$ are computed for each of the coordinates $x_1, \dots, x_d$. The integer-valued vector y in the output Freudenthal representation is assigned to be:
$$y = (\lfloor x_1 \rfloor, \dots, \lfloor x_d \rfloor).$$

2. The fractional parts $\{x_1\}, \dots, \{x_d\}$ are sorted in decreasing order. The permutation π in the output Freudenthal representation is assigned to be such that the following inequalities hold:
$$1 \geq \{x_{\pi(1)}\} \geq \ldots \geq \{x_{\pi(d)}\} \geq 0.$$

Proposition 3.1.10. *The point location outputs the Freudenthal representation $\sigma(y, \pi)$ of a d-dimensional simplex that contains the input point x. If all inequalities between the fractional parts $\{x_1\}, \dots, \{x_d\}$ are strict, then such a simplex is unique.*

The reader can find the proof of Proposition 3.1.10 in Todd [Tod76, Chapter III, Lemma 3.2].

The time complexity of the point location operation is dominated by the complexity of sorting the fractional parts $\{x_1\}, \dots, \{x_d\}$. Therefore:

Proposition 3.1.11. *The time complexity of one call of the point location is $O(d \log d)$.*

In Section 3.4, we will generalize the point location by Kuhn to an algorithm that outputs a simplex of arbitrary dimension (using the permutahedral representation from Section 3.3). The output simplex in the generalized point location is the smallest simplex that contains the input point x. Hence, in contrast to Kuhn's point location described above, the output simplex is uniquely defined.

Computing the adjacent simplices. The second operation introduced by Kuhn [Kuh68] is the adjacent d-dimensional simplex computation (see Figure 3.3). The problem is the following. We are given:

- a d-dimensional simplex in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ given by its Freudenthal representation $\sigma(y, \pi)$,
- an integer $k \in \{0, \dots, d\}$.

Let v_k be the vertex of the d-dimensional simplex $\sigma(y, \pi)$ as in Definition 3.1.3. The goal is to find the Freudenthal representation $\sigma(y', \pi')$ of the d-dimensional simplex that is adjacent by the facet of $\sigma(y, \pi)$ that is opposite to the vertex v_k.

For this, we treat three cases depending on the value of k. In each case, we explicitly give y' and π'.

- If $k = 1$, then y' and π' are as follows:
$$\begin{aligned} y' &= v_1 = y + e_{\pi(1)} \\ \pi' &= (\pi(2), \dots, \pi(d), \pi(1)) \end{aligned}$$
Note that $y', y' + e_{\pi(2)}, \dots, y' + e_{\pi(2)} + \ldots + e_{\pi(d)}$ are the vertices $v_1, \dots, v_d$ of the simplex $\hat{\sigma}(y, \pi)$. These vertices define the common facet of the simplices $\hat{\sigma}(y, \pi)$ and $\hat{\sigma}(y', \pi')$. In a similar way, we can identify the vertices of the common facet in all other cases.

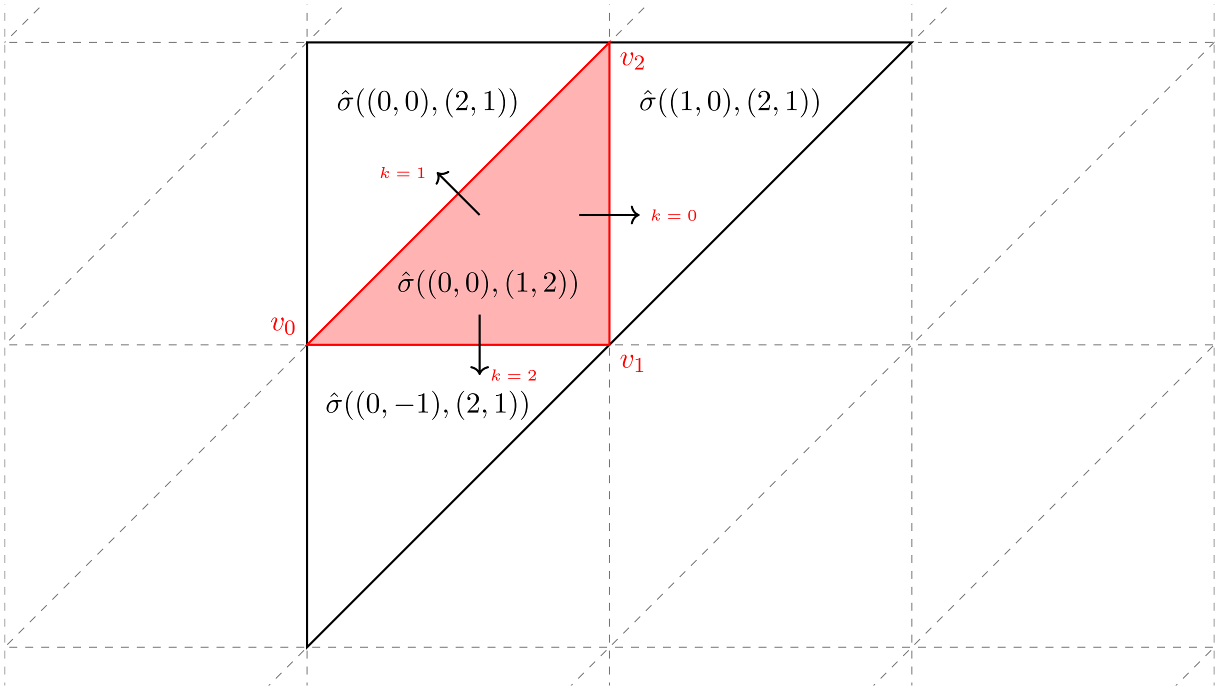

$\hat{\sigma}((0,0),(2,1))$
v_2
$\hat{\sigma}((1,0),(2,1))$
$k=1$
$k=0$
$\hat{\sigma}((0,0),(1,2))$
v_0
v_1
$k=2$
$\hat{\sigma}((0,-1),(2,1))$

Figure 3.3: Freudenthal representations of the simplices in the Freudenthal-Kuhn triangulation of $\mathbb{R}^2$ (in grey dashed) that are adjacent to the simplex $\hat{\sigma}(y=(0,0),\pi=(1,2))$ (highlighted in red) by a facet. The vertices v_0, v_1, v_2 of the simplex $\hat{\sigma}(y=(0,0),\pi=(1,2))$ are as in Definition 3.1.3. The values of k are shown on the arrows between two adjacent simplices.

- If $1 < k < d$, then y' and π' are as follows:

$$\begin{aligned} y' &= y \\ \pi' &= (\pi(1),\dots,\pi(k-2),\pi(k),\pi(k-1),\pi(k+1),\dots,\pi(d)) \end{aligned}$$

- If $k = d$, then y' and π' are as follows:

$$\begin{aligned} y' &= y - e_{\pi(d)} \\ \pi' &= (\pi(d),\pi(1),\dots,\pi(d-1)) \end{aligned}$$

All cases are illustrated in Figure 3.3. Note that if $1 < k < d$, then y' is the same as y. In this case, it means that the output simplex lies in the same cube of the unit cubical partition of $\mathbb{R}^d$ as the input simplex.

3.1.2 Link with Coxeter triangulations of type $\tilde{A}_d$

The Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ from Definition 3.1.8 is closely related to Coxeter triangulations of type $\tilde{A}_d$, see Section 1.2.4. These two triangulations can be obtained from one another *via* a linear transformation.

Theorem 3.1.12 ([DWLT90, p.405])**.** *The Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ is identical to a Coxeter triangulation of type $\tilde{A}_d$ up to a linear transformation.*

A proof of Theorem 3.1.12 can be found in [DWLT90, p.405]. Here we give an alternative and completely explicit proof, which is based on Proposition 3.1.17 in Section 3.1.3 and Theorem 3.1.13 below. The latter explicitly gives the normal vectors of the hyperplanes that induce the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ as a hyperplane arrangement. Proposition 3.1.17 shows that the set of these normal vectors is identical to a set of positive roots in a root system of type A_d up to an explicitly given linear transformation.

Theorem 3.1.13. *Denote by $e_1, \ldots, e_d \in \mathbb{R}^d$ the vectors in the canonical basis of $\mathbb{R}^d$. The Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ is an arrangement of a family of hyperplanes:*

$$\mathcal{H} = \{H_{u,k} \mid u \in E,\ k \in \mathbb{Z}\},$$

where the hyperplanes $H_{u,k}$ are defined as in Definition 1.2.23:

$$H_{u,k} = \{x \in \mathbb{R} \mid \langle x, u \rangle = k\},$$

and E is the following set of vectors:

$$E = \{e_1, \ldots, e_d\} \cup \{u_{i,j} = e_j - e_i \mid 1 \leq i < j \leq d\}.$$

The proof of Theorem 3.1.13 is based on Lemmas 3.1.14 and 3.1.15, which will be treated first. We show Theorem 3.1.13 after the proof of Lemma 3.1.15.

Lemma 3.1.14 below gives explicit supporting hyperplanes for d-dimensional simplices in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$.

Lemma 3.1.14. *For any d-dimensional simplex in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$, its supporting hyperplanes belong to the family $\mathcal{H}$.*

More precisely, let $\hat{\sigma}(y, \pi)$ be the Freudenthal representation of a d-dimensional simplex for some $y = (y_1, \ldots, y_d) \in \mathbb{Z}^d$ and a permutation $\pi : \{1, \ldots, d\} \to \{1, \ldots, d\}$. Define the vertices $v_0 \ldots, v_d$ of τ as in Definition 3.1.3. For each $i \in \{0, \ldots, d\}$, we can express each vertex v_i as:

$$v_i = y + \sum_{j=1}^{i} e_{\pi(j)}.$$

1. *The facet of τ that contains the vertices $v_1, \ldots, v_d$ lies on the hyperplane:*

$$H_{e_{\pi(1)}, y_{\pi(1)}+1} = \{x \in \mathbb{R} \mid \langle x, e_{\pi(1)} \rangle = y_{\pi(1)} + 1\}.$$

2. *The facet of τ that contains the vertices $v_0, \ldots, v_{d-1}$ lies on the hyperplane:*

$$H_{e_{\pi(d)}, y_{\pi(d)}} = \{x \in \mathbb{R} \mid \langle x, e_{\pi(d)} \rangle = y_{\pi(d)}\}.$$

3. *For any $i \in \{1, \ldots, d-1\}$, the facet of τ that contains the vertices $v_0, \ldots, v_{i-1}, v_{i+1}, \ldots, v_d$ lies on the hyperplane:*

$$H_{e_{\pi(i+1)} - e_{\pi(i)}, y_{\pi(i+1)} - y_{\pi(i)}} = \{x \in \mathbb{R} \mid \langle x, e_{\pi(i+1)} - e_{\pi(i)} \rangle = y_{\pi(i+1)} - y_{\pi(i)}\}.$$

Proof. Follows from a simple calculation. □

Now, we will show that for any hyperplane $H_{u,k} \in \mathcal{H}$, there exists a $(d-1)$-dimensional facet in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ that lies on this hyperplane.

Lemma 3.1.15. *Let $u \in E$ and $k \in \mathbb{Z}$. For any point x on the hyperplane $H_{u,k}$, there exists a $(d-1)$-dimensional simplex in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ that contains x.*

Proof. Let $u \in E$ and $k \in \mathbb{Z}$. Let $x = (x_1, \ldots, x_d)$ be a point on the hyperplane $H_{u,k}$. Define $y = (y_1, \ldots, y_d)$, where $y_i = \lfloor x_i \rfloor$, for all $i \in \{1, \ldots, d\}$. We order the fractional parts of the coordinates of the point x in the descending order:

$$1 \geq \{x_{i_1}\} \geq \ldots \geq \{x_{i_d}\} \geq 0.$$

We will separate the proof into the study of two types of vectors $u \in E$.

- Assume that u is of the form e_j. Note that because x lies on $H_{e_j,k}$, we have $x_j = k \in \mathbb{Z}$. Therefore, we have $\{x_j\} = 0$. Without loss of generality, we assume $i_d = j$. The point x lies on two d-dimensional simplices in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ (see Proposition 3.1.10). One of them is the simplex $\hat{\sigma}(y, \pi)$ with $\pi = (i_1, \ldots, i_d)$. The other simplex is $\hat{\sigma}(y', \pi')$ with $y' = y - e_j = (y_1, \ldots, y_{j-1}, y_j - 1, y_{j+1}, \ldots, y_d)$ and $\pi = (i_d, i_1, \ldots, i_{d-1})$. The point x lies in the intersection of the two simplices $\hat{\sigma}(y, \pi)$ and $\hat{\sigma}(y', \pi')$, which is a $(d-1)$-dimensional simplex in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$, as desired.

- Now assume that u is of the form $e_j - e_i$. Note that because x lies on $H_{e_j - e_i,k}$, we have $x_j - x_i = k \in \mathbb{Z}$. Therefore, we have $\{x_j\} = \{x_i\}$. Without loss of generality, we assume $i_l = j$ and $i_{l+1} = j$ for some $l \in \{1, \ldots, d-1\}$. The point x lies on two d-dimensional simplices in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ (see Proposition 3.1.10). One of them is the simplex $\hat{\sigma}(y, \pi)$ with $\pi = (i_1, \ldots, i_d)$. The other simplex is $\hat{\sigma}(y, \pi')$ with $\pi = (i_1, \ldots, i_{l-1}, i_{l+1}, i_l, i_{l+1}, \ldots, i_d)$. The point x lies in the intersection of the two simplices $\hat{\sigma}(y, \pi)$ and $\hat{\sigma}(y, \pi')$, which is a $(d-1)$-dimensional simplex in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$, as desired.

□

We are now ready to prove Theorem 3.1.13.

Proof (of Theorem 3.1.13). The result follows from three facts:

1. The Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ is a triangulation of $\mathbb{R}^d$ (Theorem 3.1.7).

2. The $(d-1)$-dimensional simplices in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ lie on hyperplanes of type $H_{u,k}$ where $u \in E$ and $k \in \mathbb{Z}$ (Lemma 3.1.14).

3. For any $u \in E$, $k \in \mathbb{Z}$ and any $x \in H_{u,k}$, there exists a $(d-1)$-dimensional simplex in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ that contains x (Lemma 3.1.15).

From these three statements it follows that the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ is an arrangement of hyperplanes in the set $\mathcal{H}$. □

3.1.3 Linear transformation from the Freudenthal-Kuhn triangulation to a Coxeter triangulation of type $\tilde{A}_d$

We will now construct the linear transformation μ that maps the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ to a Coxeter triangulation of type $\tilde{A}_d$, therefore proving Theorem 3.1.12. The construction of the linear transformation μ in this section will be important in Chapter 4, as it allows us to encode a Coxeter triangulation of type $\tilde{A}_d$ using the data structure from Section 4.1.

Both the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ and Coxeter triangulations of type $\tilde{A}_d$ are hyperplane arrangments. We will first focus on the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$. From Theorem 3.1.13, the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ is a hyperplane arrangement of the $d(d+1)/2$ families of hyperplanes that are orthogonal to the following vectors:

$$\begin{aligned} &e_i, && \text{for all } i \in \{1,\dots,d\},\\ &u_{i,j} = e_j - e_i, && \text{for all } i,j \in \{1,\dots,d\} \text{ such that } i < j, \end{aligned}$$

where $e_1,\dots,e_d$ are the vectors of the canonical basis of $\mathbb{R}^d$.

Definition 3.1.16. *As in Theorem 3.1.13, we will denote by E the set of all vectors e_i and $u_{i,j}$:*

$$E = \{e_i \mid 1 \leq i \leq d\} \cup \{u_{i,j} \mid 1 \leq i \leq j \leq d\}.$$

Let $S = \{s_1,\dots,s_d\}$ be a set of simple roots (Definition 1.2.8) in a root system R of type A_d. A matrix of such simple roots (that we will also call S by abuse of notation) can be obtained by diagonalizing the $d \times d$ Cartan matrix of a root system of type $\tilde{A}_d$ (see the proof of Proposition 1.2.5):

$$C = \begin{pmatrix} 2 & -1 & 0 & \dots & 0 \\ -1 & 2 & -1 & \ddots & \vdots \\ 0 & -1 & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & -1 \\ 0 & \dots & 0 & -1 & 2 \end{pmatrix}.$$

Write $C = O^t D O$, where O is an orthogonal matrix and D is a diagonal matrix with diagonal entries d_{ii}. Since C is positive definite, for all $i \in \{1,\dots,d\}$, we have $d_{ii} > 0$. Write:

$$S = O^t \sqrt{D},$$

where $\sqrt{D}$ is a diagonal matrix with diagonal entries $\sqrt{d_{ii}}$. For all $i \in \{1,\dots,d\}$, the row i in the matrix S is the simple root s_i.

Recall that the corresponding Coxeter triangulation of type $\tilde{A}_d$ is the arrangement of families of hyperplanes that are orthogonal to positive roots induced by the simple roots in S (see Corollary 1.2.32). According to Bourbaki [Bou02, Planche II], the set of $d(d+1)/2$ positive roots R_+ is expressed in the following manner:

$$R_+ = \left\{ \sum_{l=i}^{j} s_l \;\middle|\; 1 \leq i \leq j \leq d \right\}.$$

We introduce the notation for positive roots, for all $1 \leq i \leq j \leq d$:

$$r_{i,j} = \sum_{l=i}^{j} s_l \in R_+.$$

Let $\mu : \mathbb{R}^d \to \mathbb{R}^d$ be the linear transformation such that for all $i \in \{1,\dots,d\}$:

$$\mu(e_i) = r_{1,i} = \sum_{i=1}^{j} s_i.$$

Since the vectors $e_1, \dots, e_d$ form a basis of $\mathbb{R}^d$, the linear transformation μ is well-defined. Moreover, because the vectors $r_{1,1}, \dots, r_{1,d}$ also form a basis of $\mathbb{R}^d$, the linear transformation μ is bijective.

We will now show that μ transforms the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ to the Coxeter triangulation of type $\tilde{A}_d$ that corresponds to S. For this, we need to show that μ transforms bijectively the hyperplanes that define the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ into the hyperplanes that define the Coxeter triangulation of type $\tilde{A}_d$. This is equivalent to showing that μ maps the set E of normal vectors in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ to the set R_+ of normal vectors in the triangulation $\mathcal{T}$.

Proposition 3.1.17. *The linear transformation μ maps any vector E to a vector in R_+.*

Proof. We treat the vectors e_i and $u_{i,j}$ separately.

- Let $e_i \in E$ for some $i \in \{1, \dots, d\}$. By definition of μ, the vector $\mu(e_i) = r_{1,i}$ indeed lies in R_+.

- Now let $u_{i,j} \in E$ for some $i, j \in \{1, \dots, d\}$ such that $i < j$. We have:

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

  Hence $\mu(u_{i,j})$ lies in R_+.

□

Proof of Theorem 3.1.12. We will now show that the linear transformation μ maps the Freudenthal-Kuhn triangulation $\mathcal{T}_0$ of $\mathbb{R}^d$ to the Coxeter triangulation of type $\tilde{A}_d$ defined by the simple roots in S.

Proof (of Theorem 3.1.12). According to Theorem 3.1.13, the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ is an arrangement of a family of hyperplanes:

$$\mathcal{H} = \{H_{u,k} \mid u \in E,\ k \in \mathbb{Z}\},$$

where the hyperplanes $H_{u,k}$ are defined as in Definition 1.2.23:

$$H_{u,k} = \{x \in \mathbb{R} \mid \langle x, u \rangle = k\}.$$

By applying the linear transformation μ on the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$, we obtain the arrangement of another family of hyperplanes:

$$\mathcal{H}' = \{H_{\mu(u),k} \mid u \in E,\ k \in \mathbb{Z}\}.$$

From Proposition 3.1.17, we know that this set is equal to $\{H_{r,k} \mid r \in R_+,\ k \in \mathbb{Z}\}$. Thus, the arrangement of hyperplanes in $\mathcal{H}'$ is the Coxeter triangulation of type $\tilde{A}_d$ defined by the simple roots in S. The result follows. □

Matrix Λ_C. Now, we are interested in finding the matrix Λ_C such that $\mu(x) = \Lambda_C x$. This construction will be used in Section 4.1. Instead of describing the matrix Λ_C itself, it is easier to describe the inverse matrix Λ_C^{-1} that maps the vectors of the canonical basis e_i to the vectors $r_{1,i} = \sum_{l=1}^{i} s_l$. Each row i of the matrix Λ_C^{-1} represents the vector $r_{1,i}$. The matrix Λ_C^{-1} is thus the product of a lower-triangular $d \times d$ matrix and the matrix S (found by diagonalizing the matrix C above):

$$\Lambda_C^{-1} = \begin{pmatrix} — & s_1 & — \\ — & s_1 + s_2 & — \\ & \vdots & \\ — & s_1 + \ldots + s_d & — \end{pmatrix} = \begin{pmatrix} 1 & 0 & \ldots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & 0 \\ 1 & \ldots & \ldots & 1 \end{pmatrix} \begin{pmatrix} — & s_1 & — \\ — & s_2 & — \\ & \vdots & \\ — & s_d & — \end{pmatrix} = \begin{pmatrix} 1 & 0 & \ldots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & 0 \\ 1 & \ldots & \ldots & 1 \end{pmatrix} S.$$

By inverting this matrix, we get the matrix Λ_C as desired.

3.1.4 Simplex quality of Freudenthal-Kuhn triangulations

In this section, we will give an overview of some geometric properties of the d-dimensional simplices in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$. In general, these simplices are not similar (in the sense of Definition 1.1.47), as was pointed out by Moore [Moo92, Section 2.3.1]. However, there are measures that are common to all d-dimensional simplices τ in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$:

- The diameter $L(\tau) = \sqrt{d}$ is the length of the diagonal of the unit cube that contains the simplex τ.
- The circumradius $R(\tau) = \sqrt{d}/2$ is half of the diameter, since the circumcentre of τ lies in the center of the unit cube that contains τ and cuts the diagonal in two.
- The inradius $r(\tau) = \frac{1}{\sqrt{2}(d-1+\sqrt{2})}$, as reported by Moore [Moo92, Theorem 6].
- The volume $vol(\tau) = \frac{1}{d!}$, as reported by Dahmen and Miccheli [DM82, Lemma 2.1].
- The minimal height $h(\tau) = 1/\sqrt{2}$, unless when $d = 1$, in which case $h(\tau) = 1$ (Proposition 3.1.18 below).

Because the measures above are common to all d-dimensional simplices τ in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$, so are the quality measures from Definition 2.1.1 as well:

Proposition 3.1.18. *Let τ be any d-dimensional simplex in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$. Then:*

- *The fatness of τ is:*

$$\Theta(\tau) = \frac{vol(\tau)}{L(\tau)^d} = \frac{1}{d^{d/2}\, d!}.$$

- *The radius ratio of τ is:*

$$\rho(\tau) = \frac{r(\tau)}{R(\tau)} = \frac{\sqrt{2}}{\sqrt{d}(d-1+\sqrt{2})}.$$

- *The thickness of τ is:*
$$\theta(\tau) = \frac{h(\tau)}{L(\tau)} = \frac{1}{\sqrt{2d}}.$$

- *The aspect ratio of τ is:*
$$\alpha(\tau) = \frac{h(\tau)}{2R(\tau)} = \frac{1}{\sqrt{2d}}.$$

The only thing left to show is the value of the minimal height.

Lemma 3.1.19. *The minimal height of any d-dimensional simplex in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ is:*

- $1/\sqrt{2}$*, if* $d \geq 2$*,*
- 1*, if* $d = 1$*.*

Proof. If $d = 1$, the one-dimensional simplex has length 1, which is also its (minimal) height. We will assume $d \geq 2$ for the rest of the proof.

Let τ be a d-dimensional simplex in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$. Write τ using its Freudenthal representation $\hat{\sigma}(y, \pi)$ for some $y = (y_1, \ldots, y_d) \in \mathbb{Z}^d$ and permutation $\pi : \{1, \ldots, d\} \to \{1, \ldots, d\}$. Define the vertices $v_0, \ldots, v_d$ of τ as in Definition 3.1.3. For each v_i, the corresponding height of τ is the distance from v_i to the hyperplane specified in Lemma 3.1.14. For each $i \in \{0, \ldots, d\}$, we can express each vertex v_i as:

$$v_i = y + \sum_{j=1}^{i} e_{\pi(j)}.$$

We will now compute this distance for each case from Lemma 3.1.14.

1. The distance from $v_0 = y$ to the hyperplane $H_{e_{\pi(1)}, y_{\pi(1)}+1}$ is:
$$\frac{\| \left\langle v_0, e_{\pi(1)} \right\rangle - y_{\pi(1)} - 1 \|}{\| e_{\pi(1)} \|} = 1.$$

2. The distance from $v_d = y + \sum_{j=1}^{d} e_{\pi(j)}$ to the hyperplane $H_{e_{\pi(d)}, y_{\pi(d)}}$ is:
$$\frac{\| \left\langle v_d, e_{\pi(d)} \right\rangle - y_{\pi(d)} \|}{\| e_{\pi(d)} \|} = 1.$$

3. For each $i \in \{1, \ldots, d-1\}$, the distance from $v_i = y + \sum_{j=1}^{i} e_{\pi(j)}$ to the hyperplane $H_{e_{\pi(i+1)} - e_{\pi(i)}, y_{\pi(i+1)} - y_{\pi(i)}}$ is:
$$\frac{\| \left\langle v_i, e_{\pi(i+1)} - e_{\pi(i)} \right\rangle - (y_{\pi(i+1)} - y_{\pi(i)}) \|}{\| e_{\pi(i+1)} - e_{\pi(i)} \|} = \frac{1}{\sqrt{2}}.$$

We conclude that if $d \geq 2$, then the minimal height of τ is $1/\sqrt{2}$. □

3.1.5 Goal for the following two sections

The Freudenthal representation is limited to d-dimensional simplices. In the following Sections 3.2 and 3.3, we want to generalize the Freudenthal representation to simplices of arbitrary dimension. The key to this generalization lies in the following two observations:

- A Coxeter triangulation of type $\tilde{A}_d$ and the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ share the same combinatorial structure (Theorem 3.1.12).
- The dual of the star of a vertex in a Coxeter triangulation of type $\tilde{A}_d$ is a permutahedron (see Section 1.3).

From these two observations, it follows that the dual of the star of a vertex in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ is isomorphic to a d-dimensional permutahedron. We can therefore label the simplices in the star of a vertex in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ using ordered partitions, similarly to how we labelled faces of a permutahedron in Theorem 1.3.11. For this reason, we call the new representation of simplices of arbitrary dimension in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ the *permutahedral representation*, which we ultimately define in Definition 3.3.15.

Before we can define the permutahedral representation of simplices of arbitrary dimension, we first need a few intermediate notations for simplices in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$. These notations, called *Eaves notation* for d-dimensional simplices and *generalized Eaves notation* for simplices of arbitrary dimension, are the subject of Sections 3.2 and 3.3.6.

3.2 Eaves notation for d-dimensional simplices

In this section, we describe a notation that encodes the d-dimensional simplices in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ suggested by Eaves [Eav84]. We call this notation *Eaves notation.* This notation is the first step in generalizing the Freudenthal representation to simplices of arbitrary dimension.

When describing a d-dimensional simplex using Eaves notation $\hat{\varphi}(y, \Pi)$, we indicate:

- a vertex y that represents a star in the triangulation, rather than a cube as in the case of the Freudenthal representation,
- a permutation $\Pi : \{1, \ldots, d+1\} \to \{1, \ldots, d+1\}$ that encodes the position of the d-dimensional simplex in the star of y.

Recall that the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ and a Coxeter triangulation of type $\tilde{A}_d$ share the same combinatorial structure. The dual of the star of a vertex in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ is hence isomorphic to the Voronoi cell in a Coxeter triangulation of type $\tilde{A}_d$, which is a d-dimensional permutahedron (see Section 1.3).

The dual structure of the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ gives us an important interpretation of the role of the permutation π in Eaves notation $\hat{\varphi}(y, \Pi)$ (something that Eaves was unaware of [Eav84]). In fact, the permutations Π used in Eaves notations $\hat{\varphi}(y, \Pi)$ are the same permutations that encode vertices in a d-dimensional permutahedron dual to the star of the vertex y. Later in Section 3.3, we will replace these permutations by ordered partitions (as we did for a permutahedron in Section 1.3) to generalize Eaves notation to simplices of arbitrary dimension.

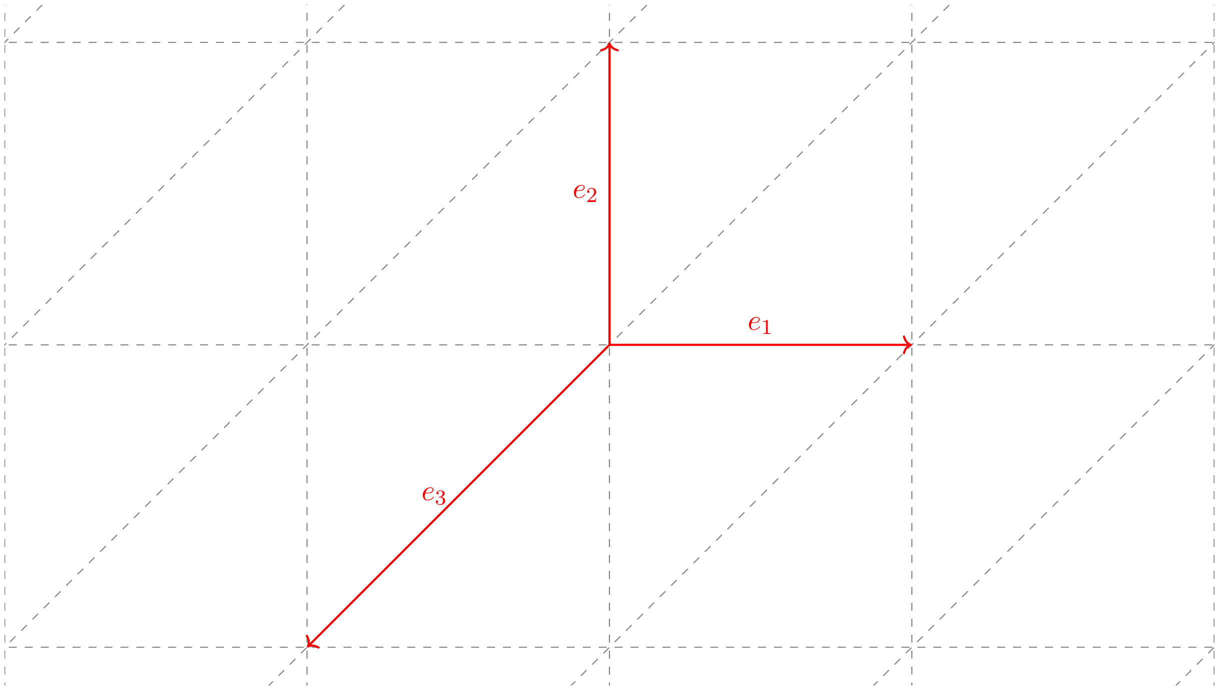


Figure 3.4: Three vectors e_1, e_2 and e_3 defined for the Freudenthal-Kuhn triangulation of $\mathbb{R}^2$.

This section is organized as follows. First, we formally define Eaves notation in Section 3.2.1. In Section 3.2.2, we prove that the simplices given by Eaves notation are the d-dimensional simplices in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$. In Section 3.2.3, we show that any d-dimensional simplex in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ can be written using Eaves notation with respect to each of its $d+1$ vertices. Lastly, we show in Section 3.2.4 that the permutations involved in Eaves notation are in bijection with the d-dimensional simplices in the star of a vertex.

3.2.1 Definition of Eaves notation for d-dimensional simplices

We will now formally define Eaves notation that labels d-dimensional simplices in the Freudenthal-Kuhn triangulations of $\mathbb{R}^d$. The exposition in this section follows Eaves [Eav84] who introduced a similar notation.

Define in addition to the canonical basis a new vector:

$$e_{d+1} = -\sum_{j=1}^{d} e_j. \tag{3.1}$$

In terms of cubes in the unit cubical partition, the vector e_{d+1} corresponds to the major diagonal of a cube in the opposite direction. The three two-dimensional vectors e_1, e_2 and e_3 are illustrated in Figure 3.4.

Definition 3.2.1 (Eaves notation for d-dimensional simplices)**.** *Let $y \in \mathbb{Z}^d$. Let Π be a permutation $\{1,\dots,d+1\} \to \{1,\dots,d+1\}$. The d-dimensional simplex $\hat{\varphi}(y,\Pi) \subseteq \mathbb{R}^d$ is the convex* *hull of $d+1$ vertices $v_0, \dots, v_d$ defined as follows:*

$$\begin{aligned} v_0 &= y \\ v_i &= v_{i-1} + e_{\Pi(i)}, \quad \textit{for } i \in \{1, \dots, d\}. \end{aligned}$$

We will refer to the notation $\hat{\varphi}(y, \Pi)$ as Eaves notation*.*

Remark 3.2.2. *Note that the vector $e_{\Pi(d+1)}$ does not participate in the definitions of the vertices $v_0, \dots, v_d$ in Definition 3.2.1. It follows that if $\Pi(d+1) = d+1$, then the restriction π of Π to $\{1, \dots, d\}$ is a permutation $\{1, \dots, d\} \to \{1, \dots, d\}$; the Freudenthal representation $\hat{\sigma}(y, \pi)$ and the Eaves notation $\hat{\varphi}(y, \Pi)$ then coincide.*

Remark 3.2.3. *In the following, we follow a convention to denote the permutation $\{1, \dots, d+1\} \to \{1, \dots, d+1\}$ in Eaves notation by a capital letter Π. This is done to prevent confusion with the permutation $\pi : \{1, \dots, d\} \to \{1, \dots, d\}$ used in the Freudenthal representation.*

We will show in Section 3.2.2 that the simplices $\hat{\varphi}(y, \Pi)$ are simplices in Freudenthal-Kuhn triangulation of $\mathbb{R}^d$.

Remark 3.2.4 (Eaves notation and cycles on a cube)**.** *In Remark 3.1.6, we described a link between the permutation $\pi : \{1, \dots, d\} \to \{1, \dots, d\}$ in a Freudenthal representation $\hat{\sigma}(y, \pi)$ and a path on the cube, which follows the vectors $e_{\pi(1)}, \dots, e_{\pi(d)}$. In Eaves notation $\hat{\varphi}(y, \Pi)$, an additional $(d+1)$-st vector is added to this list to complete the path into a cycle along $e_{\Pi(1)}, \dots, e_{\Pi(d+1)}$ (see Figure 3.5).*

Following the vectors $e_{\Pi(1)}, \dots, e_{\Pi(d+1)}$ always ends up to be a cycle thanks to the choice of the vector e_{d+1}. Indeed, because $(\Pi(1), \dots, \Pi(d+1))$ is a permutation of $\{1, \dots, d+1\}$, the sum of all vectors $e_{\Pi(1)}, \dots, e_{\Pi(d+1)}$ is the same as the sum of the vectors $e_1, \dots, e_{d+1}$. This sum is equal to:

$$\sum_{j=1}^{d+1} e_{\Pi(j)} = \sum_{j=1}^{d+1} e_j = \sum_{j=1}^{d} e_j + e_{d+1} = \sum_{j=1}^{d} e_j - \sum_{j=1}^{d} e_j = 0.$$

3.2.2 Link between Eaves notation and the Freudenthal-Kuhn triangulation

The simplices defined in Definition 3.2.1 are d-dimensional simplices in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$.

Theorem 3.2.5 ([Eav84, Lemma 6.8])**.** *Let $y \in \mathbb{Z}^d$. Let $\Pi : \{1, \dots, d+1\} \to \{1, \dots, d+1\}$ be a permutation. The simplex given by the Eaves notation $\hat{\varphi}(y, \Pi)$ belongs to the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$.*

Conversely, any d-dimensional simplex in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ admits an Eaves notation.

Theorem 3.2.6 ([Eav84, Lemma 6.5])**.** *For any simplex τ in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$, there exists a vertex $y \in \mathbb{Z}^d$ and a permutation $\Pi : \{1, \dots, d+1\} \to \{1, \dots, d+1\}$ such that $\tau = \hat{\varphi}(y, \Pi)$.*

Both Theorems 3.2.5 and 3.2.6 were shown in Eaves [Eav84]. We will show analogous Theorems 3.3.4 and 3.3.5 in Section 3.3 for the generalization of Eaves notation for simplices of arbitrary dimension.

Remark 3.2.7. *We can prove Theorem 3.2.6 using Remark 3.2.4.*

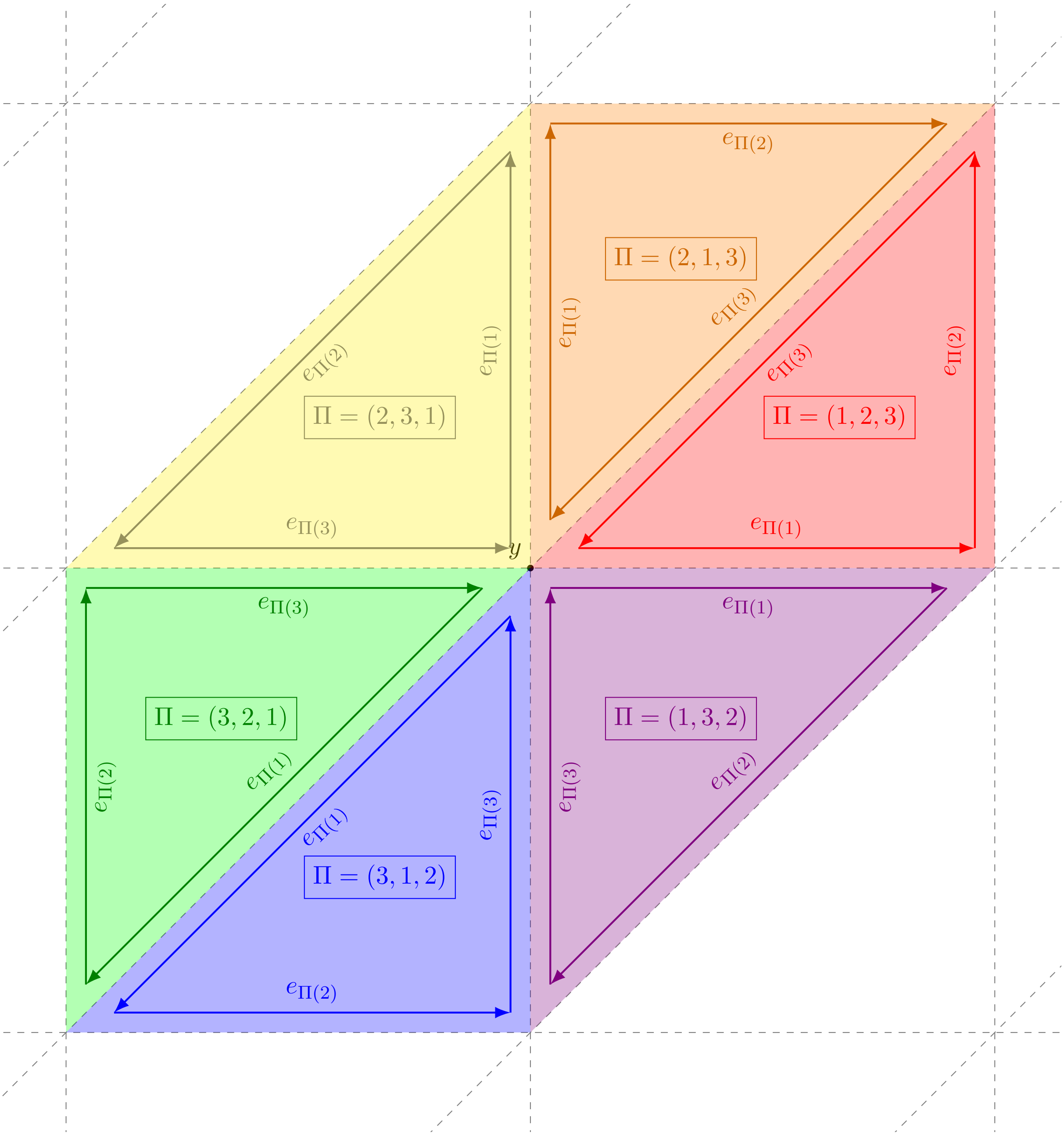


Figure 3.5: Triangles in the star of a vertex y in the two-dimensional Freudenthal-Kuhn triangulation. For each triangle, the permutations Π such that $\hat{\varphi}(y, \Pi)$ is the corresponding Eaves notation for the fixed y is indicated. The permutations Π are visualized as cycles along the vectors $e_{\Pi(1)}, e_{\Pi(2)}, e_{\Pi(3)}$.

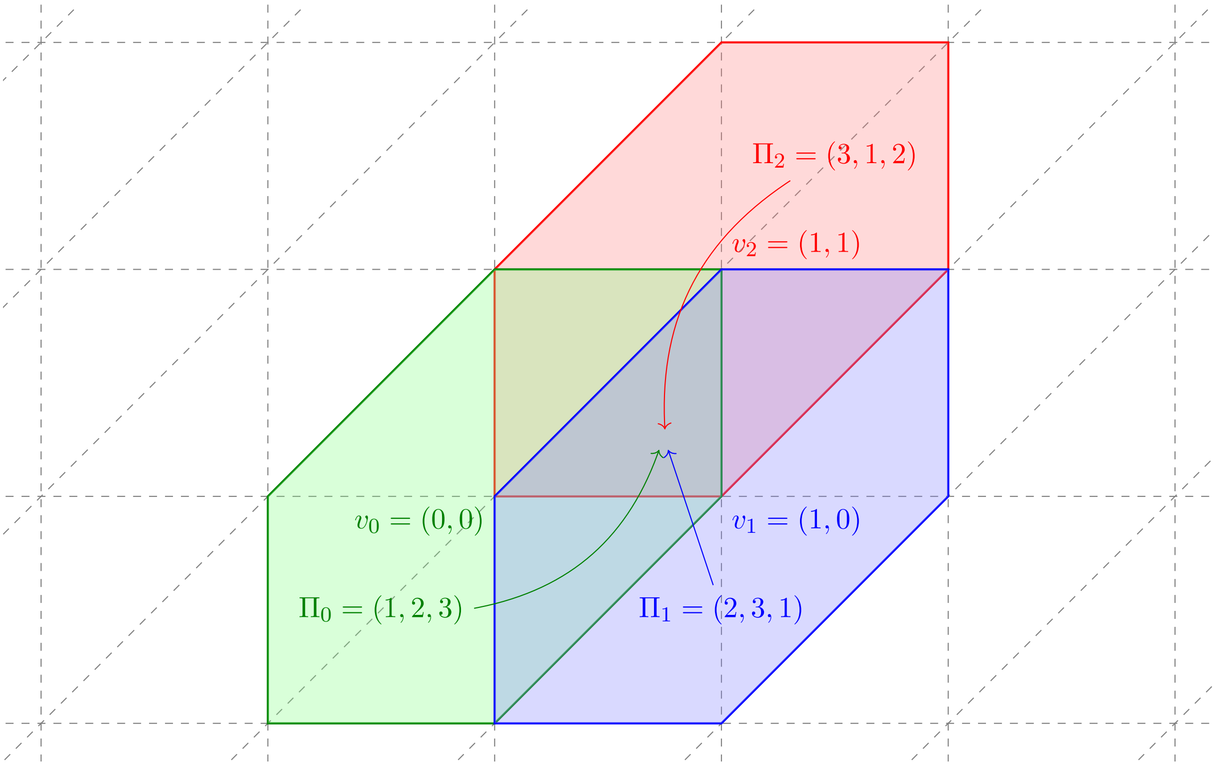


Figure 3.6: A triangle in the two-dimensional Freudenthal-Kuhn triangulation and the three stars of its vertices v_0, v_1, v_2 to which it belongs. For each vertex v_k, the permutation Π_k is indicated such that $\hat{\varphi}(v_k, \Pi_k)$ is an Eaves notation of the triangle (see Proposition 3.2.9).

3.2.3 Eaves notation with respect to each vertex

As was noted in Eaves [Eav84], the same d-dimensional simplex in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ can be encoded using $d+1$ different Eaves notations: one per vertex of the simplex. This is coherent with the observation that the vertex y in Eaves notation $\hat{\varphi}(y, \pi)$ represents the star of y in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$. Since a d-dimensional simplex lies in $d+1$ different stars of vertices, there are $d+1$ different Eaves notations for the simplex (see Figure 3.6).

We will introduce the following (non-standard) notation.

Definition 3.2.8. *Let Π be a permutation $\{1, \ldots, d+1\} \to \{1, \ldots, d+1\}$. Let $k \in \mathbb{Z}$ be an integer. We denote by $\Pi \oplus k$ the permutation defined as k elementary cyclic shifts in the positive direction applied on Π. Formally, if we denote $k' = k \bmod (d+1)$, then $\Pi' = \Pi \oplus k$ is defined as:*

- $\Pi'(i) = \Pi(i + k')$, *if* $i \in \{1, \ldots, d+1-k'\}$,
- $\Pi'(i) = \Pi(i + k' - d - 1)$, *if* $i \in \{d+2-k', \ldots, d+1\}$.

Proposition 3.2.9 (Equivalent Eaves notations [Eav84, Lemma 10.2])**.** *Let $y \in \mathbb{Z}^d$. Let Π be a permutation $\{1, \ldots, d+1\} \to \{1, \ldots, d+1\}$. Let $v_0, \ldots, v_d$ be the vertices of $\hat{\varphi}(y, \Pi)$ as in*

Definition 3.1.3:

$$\begin{aligned} &v_0 = y \\ &v_i = v_{i-1} + e_{\Pi(i)}, \quad \textit{for } i \in \{1, \dots, d\}. \end{aligned}$$

For all $k \in \{0, \dots, d\}$, define $\Pi_k = \Pi \oplus k$. Then, the Eaves notations $\hat{\varphi}(v_k, \Pi_k)$ for all $k \in \{0, \dots, d\}$ encode the same simplex.

The reader can find the proof of Proposition 3.2.9 in Eaves [Eav84, Lemma 10.2]. Alternatively, the proof of Proposition 3.2.9 follows from Proposition 3.3.8, which is a more general result for generalized Eaves notation.

3.2.4 Bijection between permutations Π and d-dimensional simplices in a star

Thanks to Proposition 3.2.9, we get that the d-dimensional simplices in the star of a vertex in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ are in bijection with permutations $\{1, \dots, d+1\} \to \{1, \dots, d+1\}$. This can be seen, for example, in Figure 3.5.

Theorem 3.2.10. *Let $y \in \mathbb{Z}^d$ be a vertex in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$. Denote by star(y) the star of y in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$. The map $\Pi \mapsto \hat{\varphi}(y, \Pi)$ is a bijection from the set of permutations $\{1, \dots, d+1\} \to \{1, \dots, d+1\}$ to the set of d-dimensional simplices in star(y).*

Proof. Fix y. According to Proposition 3.2.9, for each d-dimensional simplex τ in the star of y, there exists a permutation $\Pi : \{1, \dots, d+1\} \to \{1, \dots, d+1\}$, such that $\tau = \hat{\varphi}(y, \Pi)$. The surjectivity of the map $\Pi \mapsto \hat{\varphi}(y, \Pi)$ follows.

By Definition 3.2.1, if $\Pi \neq \Pi'$, then $\hat{\varphi}(y, \Pi) \neq \hat{\varphi}(y, \Pi')$. The injectivity of the map $\Pi \mapsto \hat{\varphi}(y, \Pi)$ follows. Since this map is both surjective and injective, we conclude that it is bijective. □

3.3 Permutahedral representation of simplices of arbitrary dimension

In this section, we present a bijective representation of simplices of arbitrary dimension in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$, which is our first contribution in the chapter. The new representation is called the *permutahedral representation*.

The permutahedral representation of simplices is used in algorithms in Section 3.4 and 3.5. These algorithms are the reason why the time complexity of the manifold tracing algorithm in Chapter 4 depends polynomially on ambient dimension.

As an intermediate step towards defining the permutahedral representation, we generalize Eaves notation to simplices of arbitrary dimension. Namely, we replace the permutation $\Pi : \{1, \dots, d+1\} \to \{1, \dots, d+1\}$ in the definition of Eaves notation for d-dimensional simplices by an ordered partition ω of $\{1, \dots, d+1\}$. We refer to the new construction as *generalized Eaves notation*. The permutahedral representation, which is the goal of the section, is one particular way to write a simplex in generalized Eaves notation. By choice, the permutahedral representations are in bijection with the simplices of all dimension in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$.

This section is organized as follows. First, we define generalized Eaves notation in Section 3.3.1. In Section 3.3.2, we prove that all simplices given by generalized Eaves notation belong to the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$. In Section 3.3.3, we prove the converse: any simplex in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ can be written using generalized Eaves notation. In Section 3.3.4, we provide the generalized Eaves notations with respect to each vertex in the simplex (generalization of Proposition 3.2.9 to simplices of arbitrary dimension). Lastly, we show in Section 3.3.6 that the ordered partitions involved in generalized Eaves notation form a poset that is isomorphic to the face poset of the star of a vertex (in the same way as in Theorem 1.3.11).

3.3.1 Generalization of Eaves notation to simplices of arbitrary dimension

In this section, we generalize Eaves notation (Definition 3.1.3) to simplices of arbitrary dimension in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$. This generalization is inspired by the way the faces of a d-dimensional permutahedron are encoded using the ordered partitions of $\{1, \dots, d+1\}$ in Theorem 1.3.11. With this in mind, we replace the permutation Π in the definition of Eaves notation for d-dimensional simplices by an ordered partition ω.

Definition 3.3.1 (Eaves notation for simplices of arbitrary dimension)**.** *Let $y \in \mathbb{Z}^d$ be a vertex in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$. Let ω be an ordered partition of $\{1, \dots, d+1\}$ in $l+1$ parts, for some $l \in \{0, \dots, d\}$. For $i \in \{1, \dots, d\}$, let e_i be the i-th vector in the canonical basis of $\mathbb{R}^d$. By convention, we define e_{d+1} as in (3.1). For all $i \in \{1, \dots, l+1\}$, denote by w_i the sum of the vectors e_j with the indices $j \in \omega(i)$:*

$$w_i = \sum_{j \in \omega(i)} e_j.$$

The l-dimensional simplex $\varphi(y, \omega) \subseteq \mathbb{R}^d$ is the convex hull of $l+1$ vertices $v_0, \dots, v_l$ defined as follows:

$$\begin{aligned} &v_0 = y \\ &v_i = v_{i-1} + w_i, \quad \text{for all } i \in \{1, \dots, l\}. \end{aligned}$$

Abusively, we will refer to the notation $\varphi(y, \omega)$ as Eaves notation*, the same as to the notation $\hat{\varphi}(y, \Pi)$ for the d-dimensional simplices. To differentiate the two notations, we will sometimes call $\varphi(y, \omega)$* generalized Eaves notation*.*

As in Section 1.3, we will write the sequence of subsets $\omega(1), \dots, \omega(l+1)$ to represent an ordered partition ω.

Remark 3.3.2. *Recall that the vectors $e_{\omega(i)}$ form a cycle in a simplex defined by Eaves notation $\hat{\varphi}(y, \omega)$. In the same way, the vectors $w_i = \sum_{j \in \omega(i)} e_j$ in Definition 3.3.1 form a cycle in a simplex defined by generalized Eaves notation $\varphi(y, \omega)$ (see Figure 3.7).*

As the following lemma shows, the definition of $\varphi(y, \omega)$ in Definition 3.3.1 indeed generalizes Eaves notation for d-dimensional simplices that we defined in Section 3.2.

Lemma 3.3.3. *Let $y \in \mathbb{Z}^d$. Let Π be a permutation $\{1, \dots, d+1\} \to \{1, \dots, d+1\}$. Define an ordered partition ω of $\{1, \dots, d+1\}$ into $d+1$ singletons as follows:*

$$\omega = \{\Pi(1)\}, \dots, \{\Pi(d+1)\}.$$

Then, we have the equality of the two d-dimensional simplices:

$$\hat{\varphi}(y, \Pi) = \varphi(y, \omega).$$

Proof. The $d+1$ vertices $v_0, \dots, v_d$ of the two simplices $\hat{\varphi}(y, \Pi)$ and $\varphi(y, \omega)$ defined in Definitions 3.2.1 and 3.3.1 respectively coincide. The result follows. □

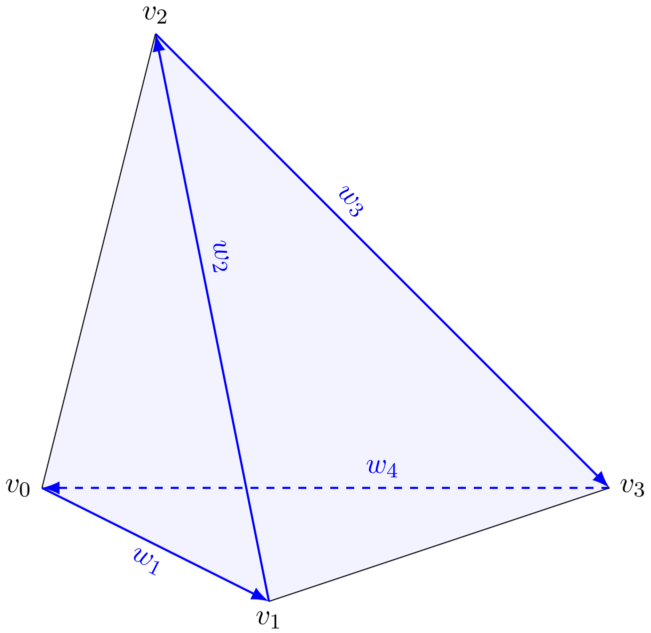


Figure 3.7: Illustration of the vertices v_0, v_1, v_2, v_3 and the vectors w_1, w_2, w_3, w_4 defined for a simplex given by a generalized Eaves notation in Definition 3.3.1. The vectors w_i form a cycle (see Remark 3.3.2).

3.3.2 Proof that Eaves notation encodes simplices in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$

The goal of this section is to prove that a simplex $\varphi(y, \omega)$ from Definition 3.3.1 belongs to the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ for any point $y \in \mathbb{Z}^d$ and any ordered partition ω of $\{1, \dots, d+1\}$. This result (Theorem 3.3.4 below) is a generalization of the result by Eaves ([Eav84, Lemma 6.8] cited as Theorem 3.2.5) to simplices of arbitrary dimension. Later in Section 3.3.3, we will also show the converse of Theorem 3.3.4: any simplex in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ can be expressed using Eaves notation (Theorem 3.3.5).

Theorem 3.3.4. *Let $y \in \mathbb{Z}^d$. Let ω be an ordered partition of $\{1, \dots, d+1\}$. The simplex $\varphi(y, \omega)$ belongs to the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$.*

To prove Theorem 3.3.4, we show that for any y and ω, the simplex $\varphi(y, \omega)$ is a face of a d-dimensional simplex in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$. More precisely, we show that $\varphi(y, \omega)$ is a face of a d-dimensional simplex $\hat{\varphi}(y, \Pi)$ (as in Definition 3.2.1), where Π is a permutation $\{1, \dots, d+1\} \to \{1, \dots, d+1\}$ expressed in terms of the elements in the parts $\omega(i)$. In the proof, we rely on the result by Eaves (Theorem 3.2.5, see also [Eav84, Lemma 6.5]) which states that the d-dimensional simplex $\hat{\varphi}(y, \Pi)$ (as in Definition 3.2.1) belongs to the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$.

Proof (of Theorem 3.3.4). We will show that the simplex $\varphi(y, \omega)$ is a face of a d-dimensional simplex in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$. Define:

$$m_0 = 0, \quad m_1 = |\omega(1)|, \quad m_2 = |\omega(1)| + |\omega(2)| \quad \dots \quad m_{l+1} = \sum_{j=1}^{l+1} |\omega(j)|.$$

Note that because ω is an ordered partition of $\{1, \dots, d+1\}$, we have $m_{l+1} = d+1$. Write the elements in the sets $\omega(1), \dots, \omega(l+1)$ as:

$$\omega(1) = \{i_1, \dots, i_{m_1}\}, \quad \omega(2) = \{i_{m_1+1}, \dots, i_{m_2}\} \quad \dots \quad \omega(l+1) = \left\{i_{m_l+1}, \dots, i_{m_{l+1}}\right\}.$$

Define vertices $v_0, \dots, v_d$ as:

$$\begin{aligned} v_0 &= y, \\ v_j &= v_{j-1} + e_{i_j}, \quad \forall j \in \{1, \dots, d\}. \end{aligned}$$

Define a permutation $\Pi : \{1, \dots, d+1\} \to \{1, \dots, d+1\}$ as follows:

$$\Pi = (i_1, \dots, i_{d+1}).$$

The d-dimensional simplex defined by the vertices $\{v_0, \dots, v_d\}$ can be written as $\hat{\varphi}(y, \Pi)$ by Definition 3.2.1. According to Theorem 3.2.5, the simplex $\hat{\varphi}(y, \Pi)$ belongs to the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$.

Observe that the vertices $\{v_{m_0}, \dots, v_{m_l}\}$ are the vertices of the simplex $\varphi(y, \omega)$. Since these vertices form a subset of $\{v_0, \dots, v_d\}$, it follows that $\varphi(y, \omega)$ is a face of the d-dimensional simplex $\hat{\varphi}(y, \Pi)$. Because $\hat{\varphi}(y, \Pi)$ belongs to the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$, it follows that $\varphi(y, \omega)$ also belongs to the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$. □

3.3.3 Proof that any simplex in the Freudenthal-Kuhn's triangulation can be written using Eaves notation

In Theorem 3.3.4, we showed that a simplex given by Eaves notation belongs to the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$. We will now show the converse: each simplex in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ can be written using Eaves notation. This result (Theorem 3.3.5 below) is a generalization of the result by Eaves (Theorem 3.2.6, see also [Eav84, Lemma 6.5]) for d-dimensional simplices.

Theorem 3.3.5. *Let τ be a simplex in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$. There exists $y \in \mathbb{Z}^d$ and an ordered partition ω of $\{1, \dots, d+1\}$ such that $\tau = \varphi(y, \omega)$.*

The proof of Theorem 3.3.5 consists of two steps.

1. We first prove the theorem for the d-dimensional simplices τ in Lemma 3.3.6. For a d-dimensional simplex, Theorem 3.2.6 yields Eaves notation $\hat{\varphi}(y, \Pi)$ for d-dimensional simplices (Definition 3.2.1). With the help of Lemma 3.3.3, we find generalized Eaves notation $\varphi(y, \omega)$ of the same d-dimensional simplex.

2. Next, we express the simplex τ as a face of a d-dimensional simplex σ in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$. From Lemma 3.3.6, we can express σ using generalized Eaves notation. We denote the vertices $\{v_0, \dots, v_d\}$ of σ as in Definition 3.2.1. Let k be the dimension of τ. We express the vertices of τ (as a face of σ) as $\{v_{m_0}, \dots, v_{m_k}\}$, for some $m_0, \dots, m_k \in \{0, \dots, d\}$ such that $m_0 < \dots < m_k$. We then construct the vertex y and the ordered partition ω from Eaves notation of the d-dimensional simplex σ and the indices $m_0, \dots, m_k$ in such a way that $\tau = \varphi(y, \omega)$.

We show a more general version of the last step separately in Lemma 3.3.7 below. Other than being used for the proof of Theorem 3.3.5, this lemma also guarantees the correctness of the face generation algorithm in Section 3.5.1. We give the proof of Theorem 3.3.5 after Lemma 3.3.7.

Step 1. The proof for the case of d-dimensional simplices. To prove Theorem 3.3.5, we first need to show the case of d-dimensional simplices. The existence of generalized Eaves notation for d-dimensional simplices will allow us to use Lemma 3.3.7 later in the proof of Theorem 3.3.5.

Lemma 3.3.6. *Let σ be a d-dimensional simplex in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$. There exists $y \in \mathbb{Z}^d$ and an ordered partition ω of $\{1, \ldots, d+1\}$ such that $\sigma = \varphi(y, \omega)$.*

Proof. According to Theorem 3.2.6, there exists a vertex $y \in \mathbb{Z}^d$ and a permutation Π such that $\sigma = \hat{\varphi}(y, \Pi)$. Lemma 3.3.3 then yields an ordered partition ω such that $\sigma = \hat{\varphi}(y, \Pi) = \varphi(y, \omega)$. □

Step 2. Eaves notation of a face given by vertices. Let τ be a simplex of arbitrary dimension in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$. To prove Theorem 3.3.5 for simplices of arbitrary dimension, we will first write τ as a face of a d-dimensional simplex σ in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$. According to Lemma 3.3.6, there exist a vertex $y \in \mathbb{Z}^d$ and an ordered partition ω of $\{1, \ldots, d+1\}$, such that $\sigma = \varphi(y, \omega)$. As the construction below shows, from y and ω, we can construct a vertex $y' \in \mathbb{Z}^d$ and an ordered partition ω' such that $\tau = \varphi(y', \omega')$. This construction proves Theorem 3.3.5.

The construction of y' and ω' that we provide below follows a more general setting, in which σ is not necessarily d-dimensional but is of some dimension $l \in \{0, \ldots, d\}$. This generalization is vital for the proof of correctness of the face generation algorithm in Section 3.5.1.

Construction of y' and ω'. Let us now describe the construction of the new vertex y' and the new ordered partition ω'. Assume that τ and σ are two simplices of the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ of dimensions k and l respectively for some $0 \leq k \leq l \leq d$, such that τ is a face of σ. Define the vectors $w_1, \ldots, w_{l+1}$ and the vertices $v_0, \ldots, v_l$ of σ as in Definition 3.3.1:

$$w_i = \sum_{j \in \omega(i)} e_j, \text{ for all } i \in \{1, \ldots, l+1\},$$

and:

$$\begin{aligned} &v_0 = y \\ &v_i = v_{i-1} + w_i, \quad \text{for } i \in \{1, \ldots, l\}. \end{aligned}$$

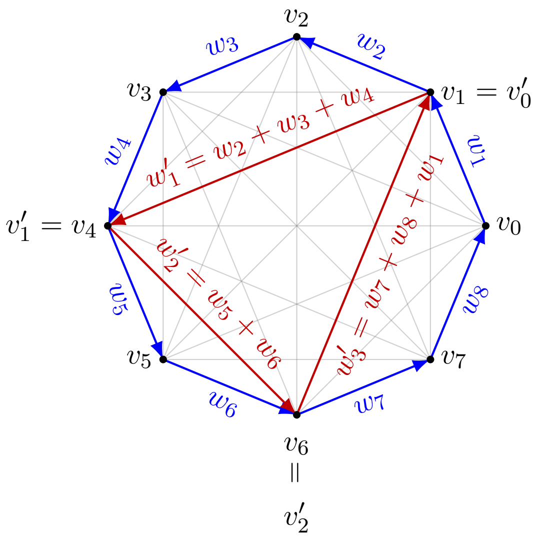
Since τ is a face of σ, its vertices form a subset of $\{v_0, \ldots, v_l\}$ (see Figure 3.8). Denote the vertices of τ by $v_{m_0}, \ldots, v_{m_k}$ with $0 \leq m_0 < \ldots < m_k \leq d$. The vertex y' in the constructed Eaves notation $\varphi(y', \omega')$ of τ can be any element in the set $\{v_{m_0}, \ldots, v_{m_k}\}$. Here, we fix $y' = v_{m_0}$ for reasons that will be clear in Section 3.5.1. Denote the vertices $v_{m_0}, \ldots, v_{m_k}$ of τ as $v'_0, \ldots, v'_k$ respectively. Thanks to the choice $y' = v_{m_0}$, the labelling of vertices $v'_0, \ldots, v'_k$ is coherent with Definition 3.3.1 (for any other choice of y', the indices in the v'_i are cyclically shifted).

For $i \in \{1, \ldots, l\}$, the vector w_i defined for σ can be expressed as the differences:

$$w_i = v_i - v_{i-1},$$

and w_{l+1} can be expressed as $w_{l+1} = v_0 - v_l$. We express the vectors w'_i for the face τ as differences of consecutive v'_j in the same way. Recall that by definition we have $v'_i = v_{m_i}$. We write the vectors w'_i as telescoping sums of consecutive vectors w_j in the following way (see also Figure 3.8). For all $i \in \{1, \ldots, k\}$, we have:

$$w'_i = v'_{i+1} - v'_i = v_{m_{i+1}} - v_{m_i} = (v_{m_{i+1}} - v_{m_{i+1}-1}) + \ldots + (v_{m_i+1} - v_{m_i}) = w_{m_{i+1}} + \ldots + w_{m_i+1}.$$



$\{m_0, m_1, m_2\} = \{1, 4, 6\}$

$\omega'(1) = \omega(2) \cup \omega(3) \cup \omega(4)$
$\omega'(2) = \omega(5) \cup \omega(6)$
$\omega'(3) = \omega(7) \cup \omega(8) \cup \omega(1)$

Figure 3.8: A seven-dimensional simplex $\sigma = \varphi(v_0, \omega)$ and a two-dimensional face $\tau = \varphi(v_0', \omega')$ of σ. The vertices $v_0, \ldots, v_7$ of σ and the vectors $w_1, \ldots, w_8$ are indexed according to Definition 3.3.1. The three vertices of τ are $v_0' = v_{m_0}$, $v_1' = v_{m_1}$ and $v_2' = v_{m_2}$, with $m_0 = 1$, $m_1 = 4$ and $m_2 = 6$. The composition of parts $\omega'(i)$ as union of the $\omega(j)$ illustrated on the right matches the compositions of vectors w_i' as sums of the vectors w_j on the left.

We express the vector w_{k+1}' in a similar way:

$$w_{k+1}' = v_{m_0} - v_{m_k} = (v_{m_0} - v_0) + (v_0 - v_l) + (v_l - v_{m_k}) = (w_{m_0} + \ldots + w_0) + w_{l+1} + (w_l + \ldots + w_{m_k+1}).$$

In order to match the parts $\omega'(i)$ in the new ordered partition ω' to the vectors w_i' as in Definition 3.3.1, we set:

$$\begin{aligned} \omega'(i) &= \omega(m_{i-1} + 1) \cup \omega(m_{i-1} + 2) \cup \ldots \cup \omega(m_i), && \text{for } i \in \{1, \ldots, k\}, \\ \omega'(k+1) &= (\omega(1) \cup \ldots \cup \omega(m_0)) \cup (\omega(m_k + 1) \cup \ldots \cup \omega(l+1)). \end{aligned}$$

The composition of the newly constructed parts in $\omega'(i)$ in terms of $\omega(j)$ is illustrated in Figure 3.9 below.

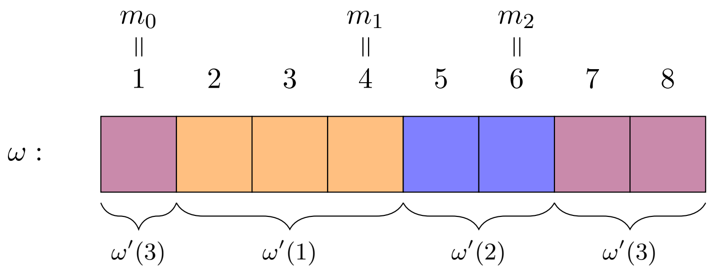


Figure 3.9: An example of sets in $\omega'(i)$ from Figure 3.8 represented as unions of sets $\omega(i)$ shown as boxes.

We summarize the construction of y' and ω' in Lemma 3.3.7.

Lemma 3.3.7 (Eaves notation of a face given by vertices)**.** *Let $l \in \{0, \dots, d\}$. Let σ be an l-dimensional simplex in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ such that $\sigma = \varphi(y, \omega)$ for some vertex $y \in \mathbb{Z}^d$ and some ordered partition ω of $\{1, \dots, d+1\}$ in $l+1$ parts. Define the vectors $w_1, \dots, w_{l+1}$ and the vertices $v_0, \dots, v_l$ of σ as in Definition 3.3.1:*

$$w_i = \sum_{j \in \omega(i)} e_j,$$

and:

$$\begin{aligned} v_0 &= y \\ v_i &= v_{i-1} + w_i, \quad \text{for } i \in \{1, \dots, l\}. \end{aligned}$$

Let τ be a k-dimensional face of σ for some $k \in \{0, \dots, l\}$. Assume that the vertices of τ form a subset $\{v_{m_0}, \dots, v_{m_k}\}$ of the vertices of σ indexed by some $0 \le m_0 < \dots < m_k \le l$. Define the ordered partition ω' as:

$$\begin{aligned} \omega'(i) &= \omega(m_{i-1}+1) \cup \omega(m_{i-1}+2) \cup \dots \cup \omega(m_i), && \text{for } i \in \{1, \dots, k\}, \\ \omega'(k+1) &= (\omega(1) \cup \dots \cup \omega(m_0)) \cup (\omega(m_k+1) \cup \dots \cup \omega(l+1)). \end{aligned}$$

Then, we have:

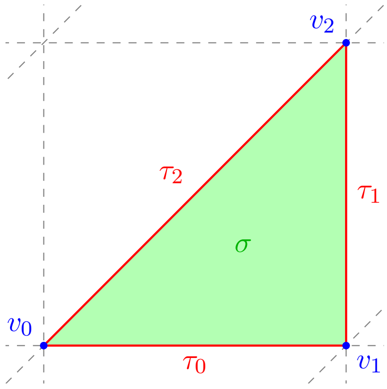
$$\tau = \varphi(v_{m_0}, \omega').$$

Proof. Follows by construction. □

We illustrate the constructed Eaves notations for all faces of a triangle in Figure 3.10.

Proof of Theorem 3.3.5. We will now prove Theorem 3.3.5. As we will see, this proof follows from Lemmas 3.3.6 and 3.3.7.

Proof (of Theorem 3.3.5). Let σ be a d-dimensional coface of τ in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$. According to Lemma 3.3.6, there exists a vertex $y \in \mathbb{Z}^d$ and an ordered partition ω of $\{1, \dots, d+1\}$ such that $\sigma = \varphi(y, \omega)$. Since τ is a face of σ and σ admits a generalized Eaves notation, we can apply Lemma 3.3.7. Hence, there exists a vertex $y' \in \mathbb{Z}^d$ and an ordered partition ω' of $\{1, \dots, d+1\}$, such that $\tau = \varphi(y', \omega')$. The proof of the theorem follows. □



	vertices	constructed Eaves' notation $\varphi(y', \omega')$
σ	$\{v_0, v_1, v_2\}$	$y' = v_0 \quad \omega' = \{1\}, \{2\}, \{3\}$
τ_0	$\{v_0, v_1\}$	$y' = v_0 \quad \omega' = \{1\}, \{2, 3\}$
τ_1	$\{v_1, v_2\}$	$y' = v_1 \quad \omega' = \{2\}, \{1, 3\}$
τ_2	$\{v_0, v_2\}$	$y' = v_0 \quad \omega' = \{1, 2\}, \{3\}$
v_0	$\{v_0\}$	$y' = v_0 \quad \omega' = \{1, 2, 3\}$
v_1	$\{v_1\}$	$y' = v_1 \quad \omega' = \{1, 2, 3\}$
v_2	$\{v_2\}$	$y' = v_2 \quad \omega' = \{1, 2, 3\}$

Figure 3.10: The constructed Eaves notations $\varphi(y', \omega')$ of all faces of a triangle in the two-dimensional Freudenthal-Kuhn triangulation: the triangle σ itself, the three edges τ_0, τ_1, τ_2 and the three vertices v_0, v_1, v_2. Each line in the table on the right has pair of values of y' and ω' such that the Eaves notation $\varphi(y', \omega')$ encodes the corresponding simplex.

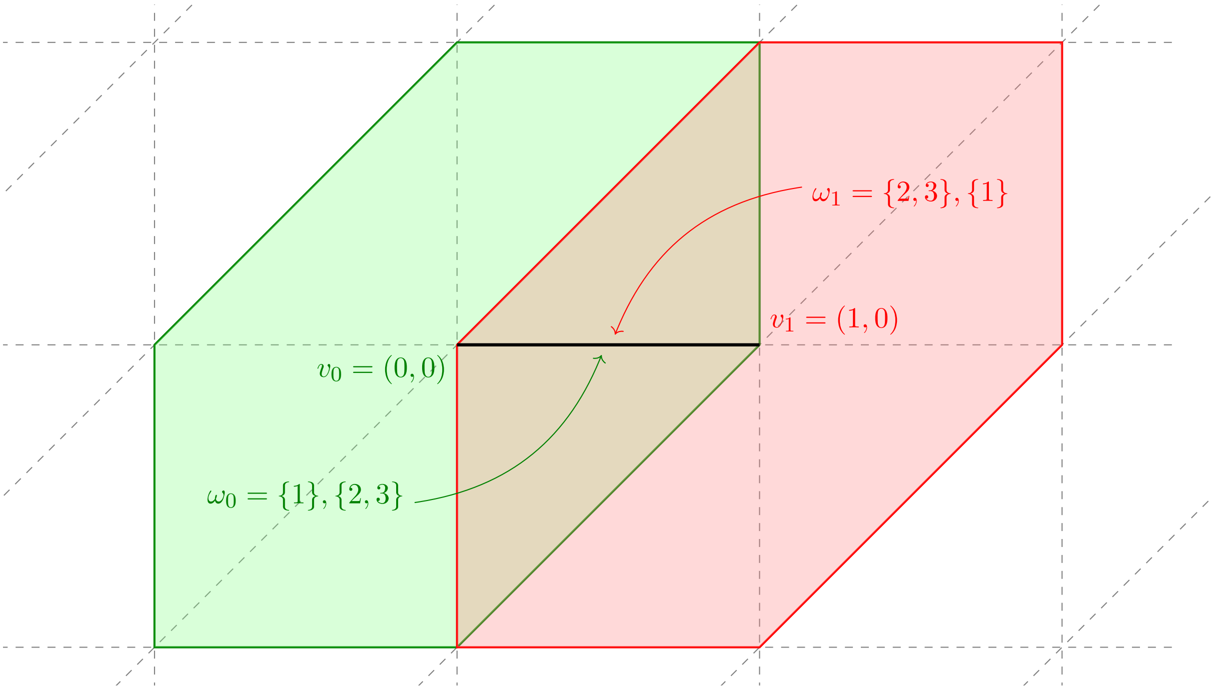


Figure 3.11: An edge in the Freudenthal-Kuhn triangulation of $\mathbb{R}^2$ and the two stars of its vertices v_0 and v_1 to which it belongs. For each vertex v_k, the ordered partition ω_k is indicated such that $\varphi(v_k, \omega_k)$ is a generalized Eaves notation of the edge (see Proposition 3.3.8).

Note that thanks to Lemma 3.3.6, the d-dimensional simplices in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ admit generalized Eaves notation. This is the key reason why we can apply Lemma 3.3.7 in the proof of Theorem 3.3.5.

3.3.4 Generalized Eaves notation with respect to each vertex

In this section, we generalize Proposition 3.2.9 to simplices of arbitrary dimension. Proposition 3.3.8 below shows that the same l-dimensional simplex in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ can be encoded using $l+1$ different Eaves notations: one per vertex of the simplex. This is coherent with the observation that the vertex y in Eaves notation $\varphi(y, \pi)$ represents the star of the vertex y in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$. Since an l-dimensional simplex lies in $l+1$ different stars of vertices, there are $l+1$ different Eaves notations for the simplex (see Figure 3.11).

As for the permutations in Definition 3.2.8, we denote by $\omega \oplus k$ the result of k elementary cyclic shifts in the positive direction applied on the ordered partition ω.

Proposition 3.3.8 (Generalized Eaves notations with respect to each vertex)**.** *Let σ be a simplex of some dimension $l \in \{0, \dots, d\}$ in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$. Let $y \in \mathbb{Z}^d$ and ω be an ordered partition of $\{1, \dots, d+1\}$ in $l+1$ parts such that $\sigma = \varphi(y, \omega)$. Let $v_0, \dots, v_l$ be the vertices of σ as in Definition 3.1.3. For all $k \in \{0, \dots, l\}$, define $\omega_k = \omega \oplus k$. Then for all $k \in \{0, \dots, l\}$, we have:*

$$\varphi(v_k, \omega_k) = \sigma.$$

Remark 3.3.9. *In the proof of Proposition 3.3.8, we follow the following convention. We see the indices i of the vertices v_i, the vectors w_i and the parts $\omega(i)$ as elements of the cyclic group*

$\mathbb{Z}/(l+1)\mathbb{Z}$, where l is the dimension of the simplex σ. Whenever there is a + symbol involved with such an index, we see it as the addition in the cyclic group $\mathbb{Z}/(l+1)\mathbb{Z}$.

To denote the elements and the subsets of $\mathbb{Z}/(l+1)\mathbb{Z}$, we will use integers in $\{0,\dots,l\}$ or in $\{1,\dots,l+1\}$ whenever appropriate.

Proof (of Proposition 3.3.8). Fix $k \in \mathbb{Z}/(l+1)\mathbb{Z}$. Write the vectors $w'_1,\dots,w'_{l+1}$ and the vertices $v'_0,\dots,v'_l$ of $\varphi(v_k,\omega_k)$ as in Definition 3.3.1:

$$w'_i = \sum_{j\in\omega(i)} e_j, \quad \text{for all } i \in \{1,\dots,l+1\},$$

and

$$\begin{aligned} v'_0 &= v_k, \\ v'_i &= v'_{i-1} + w'_i, \quad \text{for } i \in \{1,\dots,l\}. \end{aligned}$$

We will now show that the two sets of vertices $\{v_0,\dots,v_l\}$ and $\{v'_0,\dots,v'_l\}$ are equal. This is sufficient to conclude that $\varphi(v_k,\omega_k) = \sigma$. To prove the equality $\{v_0,\dots,v_l\} = \{v'_0,\dots,v'_l\}$, we will show by induction on i for all $i \in \mathbb{Z}/(l+1)\mathbb{Z}$ that we have $v'_i = v_{i+k}$.

Base case. The base case $i = 0$ of the induction follows from the definition of v'_0, which is equal to v_k.

Induction step. For the proof of the induction step, let $i \in \{1,\dots,l\}$ be such that $v'_{i-1} = v_{i-1+k}$. We will now show that we also have $v'_i = v_{i+k}$.

By definition of ω_k, we have $\omega_k(i) = \omega(i+k)$ for all $i \in \{1,\dots,l+1\}$. This implies $w'_i = w_{i+k}$. By combining this and the induction hypothesis $v'_{i-1} = v_{i-1+k}$, we get:

$$\begin{aligned} v'_i &= v'_{i-1} + w'_i \\ &= v_{i-1+k} + w_{i+k}. \end{aligned}$$

By definition of the vectors $w_1,\dots,w_{l+1}$ and the vertices $v_0,\dots,v_l$, the last expression is equal to v_{i+k}. We conclude that $v'_i = v_{i+k}$. The proof of the induction step is complete.

With the base case and the induction step shown, we conclude that for all $i \in \mathbb{Z}/(l+1)\mathbb{Z}$, we have $v'_i = v_{i+k}$. We conclude that the vertices of the two simplices σ and $\varphi(v_k,\omega_k)$ are identical. Thus, these two simplices are equal.

□

3.3.5 Isomorphism between the refinement poset of ordered partitions and the face poset of simplices in a star

In this section, we will show the following result. If we fix a vertex y in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$, then the refinement poset of the ordered partitions $\omega \in OP(\{1,\dots,d+1\})$ is isomorphic to the face poset of $\mathrm{star}(y)$. This can be seen in Figure 3.12. Note that the ordered partitions associated to the simplices in the star in Figure 3.12 are identical to the ordered partitions associated to the faces of the hexagon (two-dimensional permutahedron) in Figure 1.15.

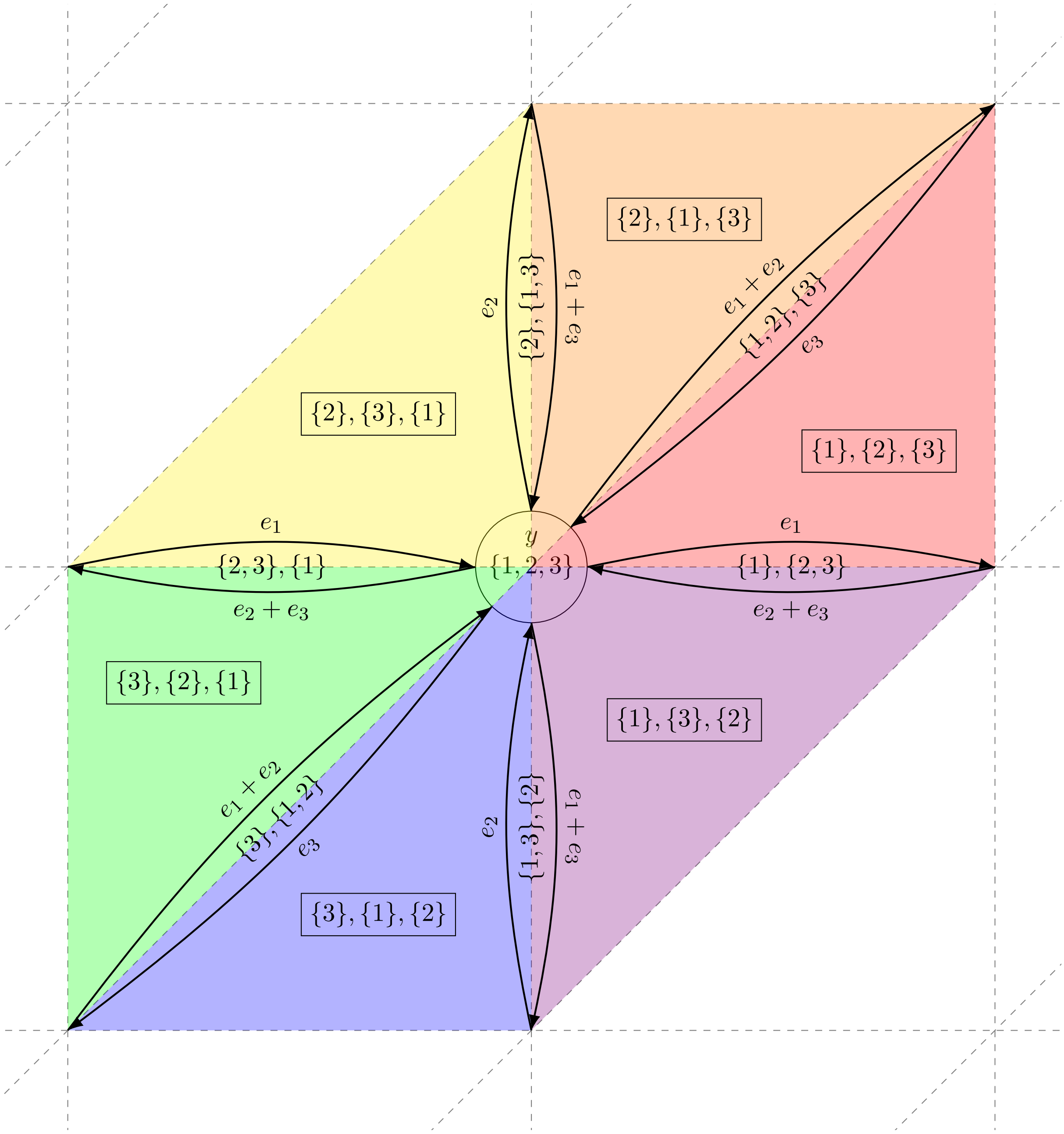


Figure 3.12: Simplices in the star of a vertex y (in the centre) in the Freudenthal-Kuhn triangulation of $\mathbb{R}^2$. For each simplex in the star, the indicated ordered partition ω is such that $\varphi(y, \omega)$ is an Eaves notation of the simplex. For edges, the cycles along the sums of vectors $w_1 = \sum_{j\in\omega(1)} e_j$ and $w_2 = \sum_{j\in\omega(2)} e_j$ are indicated.

Theorem 3.3.10. *Let $y \in \mathbb{Z}^d$ be a vertex in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$. Denote by star(y) the star of y in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$. The map $\omega \mapsto \varphi(y, \omega)$ is an isomorphism between the refinement poset of ordered partitions of $(OP(\{1, \dots, d+1\}), \sqsubseteq)$ and the face poset of star(y).*

To prove that the map $\omega \mapsto \varphi(y, \omega)$ is an isomorphism of posets, we need to show that it is an order embedding and it is surjective (see Definition 1.1.28). The surjectivity of this map follows from Proposition 3.3.8. The following corollary of Lemma 3.3.7 also shows that it is an order embedding.

Corollary 3.3.11 (Order embedding)**.** *Let $y \in \mathbb{Z}^d$ be a vertex in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$. Let $k, l \in \{0, \dots, d\}$ be such that $k \leq l$. Let σ and τ be two simplices in star(y) of dimensions l and k respectively. Let ω_σ and ω_τ be the two ordered partition of $\{1, \dots, d+1\}$ in $l+1$ and $k+1$ parts respectively such that $\sigma = \varphi(y, \omega_\sigma)$ and $\tau = \varphi(y, \omega_\tau)$. Then, we have the following equivalence: σ is a coface of τ if and only if ω_σ is a refinement of ω_τ.*

Proof (of Corollary 3.3.11). Let us first show the ($\Rightarrow$) direction. Assume that the simplex σ is a coface of the simplex τ. This implies that τ is a face of σ. We can apply Lemma 3.3.7. By the way the ordered partition ω_τ is defined in terms of the parts $\omega_\sigma(i)$ in Lemma 3.3.7, the ordered partition ω_τ is a refinement of ω_σ. The result follows.

Let us now show the ($\Leftarrow$) direction. From Definition 3.3.1 it follows that the vertices in the simplex $\varphi(y, \omega)$ form a subset of the vertices in the simplex $\varphi(y, \omega')$. Therefore $\varphi(y, \omega')$ is a coface of $\varphi(y, \omega)$, as desired. □

Remark 3.3.12. *Other than being used in the proof of Theorem 3.3.10, Corollary 3.3.11 also guarantees the correctness of the coface generation algorithm in Section 3.5.2.*

Now, we are ready to prove Theorem 3.3.10.

Proof (of Theorem 3.3.10). Let τ be a simplex in star(y). It follows from Proposition 3.3.8 that there exists an ordered partition of $\{1, \dots, d+1\}$ such that $\tau = \varphi(y, \omega)$. Thus, the map $\omega \mapsto \varphi(y, \omega)$ is surjective. Furthermore, according Corollary 3.3.11, this map is an order embedding. We conclude that the map $\omega \mapsto \varphi(y, \omega)$ is an isomorphism of posets. □

3.3.6 Definition of the permutahedral representation

In this section we will define the permutahedral representation. As we will see, the permutahedral representations are in bijection with the simplices of arbitrary dimension in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$.

Remark 3.3.13. *Recall that the Freudenthal representations from Section 3.1 are in bijection with the d-dimensional simplices in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$. We can therefore see the permutahedral representation as a generalization of the Freudenthal representation to simplices of arbitrary dimension in Freudenthal-Kuhn triangulation of $\mathbb{R}^d$.*

Definition of the permutahedral representation. To define the permutahedral representation, we will need Corollary 3.3.14 below, which is based on Proposition 3.3.8. It states that each simplex of arbitrary dimension in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ admits a unique Eaves notation with the property that $d+1$ lies in the last part of the ordered partition.

Corollary 3.3.14. *Let σ be a simplex of some dimension $l \in \{0, \dots, d\}$ in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$. There exists a unique vertex $y \in \mathbb{Z}^d$ and a unique ordered partition ω of $\{1, \dots, d+1\}$ with $d+1 \in \omega(l+1)$ such that $\sigma = \varphi(y, \omega)$.*

Proof. According to Theorem 3.3.5, there exists a vertex $v_0 \in \mathbb{Z}^d$ and an ordered partition ω_0 of $\{1, \dots, d+1\}$ in $l+1$ parts such that $\sigma = \varphi(v_0, \omega_0)$. Write the vertices $v_0, \dots, v_l$ of σ as in Definition 3.3.1. Let t be the index in $\{1, \dots, l+1\}$ such that $d+1 \in \omega_0(t)$. Define the ordered partition $\omega_t = \omega_0 \oplus t$. By definition of ω_t, we have:

$$\omega_t(l+1) = \omega_0(l+1+t-l-1) = \omega_0(t).$$

Therefore, we have $d+1 \in \omega_t(l+1)$. Moreover, from Proposition 3.3.8, we get $\sigma = \varphi(v_t, \omega_t)$. This means that the vertex v_t and the ordered partition ω_t satisfy the statement of the corollary. Since the index t is uniquely defined, we conclude that such v_t and ω_t are unique as well. □

We define the permutahedral representation from the unique Eaves notation in Corollary 3.3.14.

Definition 3.3.15 (Permutahedral representation)**.** *Let $y \in \mathbb{Z}^d$ be a vertex in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$. Let $l \in \{0, \dots, d\}$ and let ω be an ordered partition of $\{1, \dots, d+1\}$ in $l+1$ parts such that:*

$$d+1 \in \omega(l+1),$$

the l-dimensional simplex $\sigma(y, \omega)$ is defined to be the simplex $\varphi(y, \omega)$. We will refer to the notation $\sigma(y, \omega)$ as the permutahedral representation.

An effect of imposing the condition $d+1 \in \omega(l+1)$ on the ordered partition ω in the definition of the simplex $\sigma(y, \omega)$ is that the vertex y is the minimal vertex of $\sigma(y, \omega)$ in the lexicographical order.

Lemma 3.3.16 (Minimal vertex in lexicographical order)**.** *Let $y \in \mathbb{Z}^d$. Let $l \in \{0, \dots, d\}$. Let ω be an ordered partition of $\{1, \dots, d+1\}$ in $l+1$ parts such that $d+1 \in \omega(l+1)$. The vertex y is the minimal vertex of the simplex $\sigma(y, \omega)$ in the lexicographical order on the coordinates.*

Proof. Define the vectors $w_1, \dots, w_{l+1}$ and the vertices $v_0, \dots, v_l$ of the simplex $\sigma(y, \omega)$ as in Definition 3.3.1:

$$w_i = \sum_{j \in \omega(i)} e_j, \text{ for all } i \in \{1, \dots, l+1\},$$

and:

$$\begin{aligned} v_0 &= y \\ v_i &= v_{i-1} + w_i, \quad \text{for } i \in \{1, \dots, l\}. \end{aligned}$$

In particular, for all $i \in \{1, \dots, l\}$, we can express the vertex v_i as:

$$v_i = v_0 + \sum_{j=1}^{i} w_j.$$

Because $d+1 \in \omega(l+1)$, the only vector e_{d+1} out of $e_1, \dots, e_{d+1}$ that is not a vector in the canonical basis of $\mathbb{R}^d$ is a term in w_{l+1}. It follows that the vectors $w_1, \dots, w_l$ are sums of vectors in the canonical basis of $\mathbb{R}^d$. Thus, for all $i \in \{1, \dots, d\}$, the vertex v_0 is smaller than the vertex v_i in the lexicographic order. We conclude that v_0 is the smallest vertex of the simplex $\sigma(y, \omega)$ in the lexicographical order. □

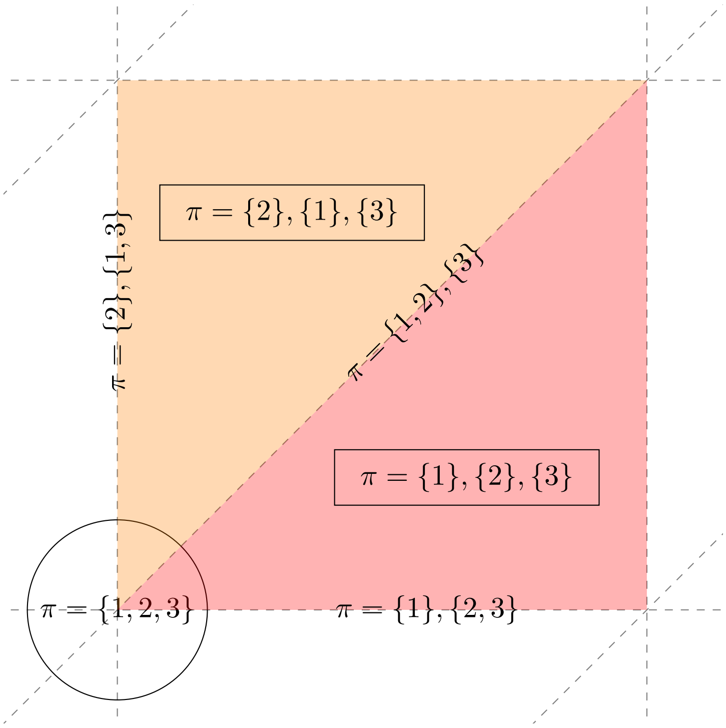


Figure 3.13: Simplices in the Freudenthal-Kuhn triangulation of $\mathbb{R}^2$ that admit the permutahedral representation $\sigma(y,\omega)$ with a fixed vertex y (circled) and some ordered partition ω of $\{1,2,3\}$. For each such simplex, the ordered partition ω is indicated.

Using Lemma 3.3.16, we can determine the set of simplices that admit the permutahedral representations $\sigma(y,\omega)$ with a fixed vertex $y \in \mathbb{Z}^d$. Since these simplices admit generalized Eaves notations $\varphi(y,\omega)$, they lie in the star of the vertex y in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$. On the other hand, according to Lemma 3.3.16, the vertex y is the minimal vertex in the lexicographical order in these simplices. It follows that the simplices $\sigma(y,\omega)$ lie in the cube associated to the vertex y in the unit cubical partition of $\mathbb{R}^d$. Thus, the simplices $\sigma(y,\omega)$ lie in the intersection of the star of y and the cube associated to the vertex y in the unit cubical partition of $\mathbb{R}^d$ (illustrated in Figure 3.13).

Bijection between the permutahedral representations and the simplices. As the following theorem shows, the permutahedral representations are in bijection with the simplices of the arbitrary dimension in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$.

Theorem 3.3.17. *The set of simplices of arbitrary dimension in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ is in bijection with the set of the pairs (y,ω) consisting of a vertex $y \in \mathbb{Z}^d$ and of an ordered partition ω of $\{1,\dots,d+1\}$ in $l+1$ parts with $l \in \{0,\dots,d\}$ such that $d+1 \in \omega(l+1)$.*

Proof. Define two maps that are inverses of one another.

- The map μ that associates to a simplex of arbitrary dimension in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ the pair (y,ω) of a vertex $y \in \mathbb{Z}^d$ and an ordered partition ω of $\{1,\dots,d+1\}$ in $l+1$ parts with $l \in \{0,\dots,d\}$ such that $d+1 \in \omega(l+1)$. This map is given by Corollary 3.3.14.

- The map μ^{-1} that associates to a vertex $y \in \mathbb{Z}^d$ and an ordered partition ω of $\{1, \ldots, d+1\}$ in $l+1$ parts with $l \in \{0, \ldots, d\}$ such that $d+1 \in \omega(l+1)$ the corresponding simplex $\sigma(y, \omega)$ in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$. This map is given by Definition 3.3.15.

Because μ and μ^{-1} are inverses of one another, we conclude that these two maps are bijections. The result follows. □

3.3.7 Characterization of the points in a simplex using a chain of inequalities

In this section, we will characterize the points in a simplex in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ using its permutahedral representation. This characterization (Proposition 3.3.19) will play a key role in the correctness proof of the point location algorithm in Section 3.4.

The points in a simplex are characterized by a chain of inequalities that we will now define.

Remark 3.3.18. *In this section and in Section 3.4, all simplices are open in their affine hull.*

Fix an l-dimensional simplex τ in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$, for some $l \in \{0, \ldots, d\}$. Write $\tau = \sigma(y, \omega)$, where $y \in \mathbb{Z}^d$ is a vertex and ω is an ordered partition of $\{1, \ldots, d+1\}$ in $l+1$ parts such that $d+1 \in \omega(l+1)$. Denote by $m_1, \ldots, m_{l+1} \in \{1, \ldots, d+1\}$ the following sums of the cardinalities $|\omega(j)|$:

$$m_1 = |\omega(1)|, \quad m_2 = |\omega(1)| + |\omega(2)| \quad \ldots \quad m_{l+1} = \sum_{j=1}^{l+1} |\omega(j)|.$$

Note that because ω is an ordered partition of $\{1, \ldots, d+1\}$, we have $m_{l+1} = d+1$. Write the elements in the sets $\omega(j)$ as:

$$\omega(1) = \{q_1, \ldots, q_{m_1}\}, \quad \omega(2) = \{q_{m_1+1}, \ldots, q_{m_2}\} \quad \ldots \quad \omega(l+1) = \{q_{m_l+1}, \ldots, q_{d+1}\}.$$

Note that $q_1, \ldots, q_{d+1}$ form a permutation of $1, \ldots, d+1$.

We will now show that the points x in the simplex τ are characterized by the following chain of inequalities:

$$1 > z_{q_1} = \ldots = z_{q_{m_1}} > z_{q_{m_1+1}} = \ldots = z_{q_{m_2}} > \ldots\ldots > z_{q_{m_l}+1} = \ldots = z_{q_{d+1}} = 0, \qquad (3.2)$$

where the z_k are defined as:

- for all $k \in \{1, \ldots, d\}$, z_k is the kth coordinate of the vector $z = (z_1, \ldots, z_d) = x - y$,
- by convention $z_{d+1} = 0$.

Proposition 3.3.19 (Characterization of the points in a simplex)**.** *The simplex $\tau = \sigma(y, \omega)$ and the subset of $\mathbb{R}^d$ that satisfies the chain of inequalities (3.2) are equal.*

We prove Proposition 3.3.19 by double inclusion. The proof is inspired by the proof of Proposition 3.3.19 in the particular case of d-dimensional simplices in Todd [Tod76, Section III.3.4].

In the following, we define the vectors $w_1, \ldots, w_{l+1}$ and the vertices $v_0, \ldots, v_l$ of τ as in Definition 3.3.1:

$$w_i = \sum_{j \in \omega(i)} e_j, \text{ for all } i \in \{1, \ldots, l+1\},$$

and:

$$\begin{aligned} v_0 &= y \\ v_i &= v_{i-1} + w_i, \quad \text{for } i \in \{1, \dots, l\}. \end{aligned}$$

We will use the following expression of the vertex v_i, for all $i \in \{0, \dots, l\}$:

$$v_i = v_0 + \sum_{j=1}^{i} w_j. \tag{3.3}$$

The proof that the points in the simplex τ satisfy the chain of inequalities (3.2) relies on Lemma 3.3.20 below. It expresses the coordinates z_k of the vector $z = x - y$ in terms of the barycentric coordinates of the point x in the simplex τ. We will prove Proposition 3.3.19 after Lemma 3.3.20.

Lemma 3.3.20 (Expression of the z_k)**.** *Let x be a point in the simplex τ and let $\lambda_0, \dots, \lambda_l \in (0,1]$ be its barycentric coordinates, i.e. $\sum_{i=0}^{l} \lambda_i = 1$ and:*

$$x = \sum_{i=0}^{l} \lambda_i v_i.$$

Let $k \in \{1, \dots, d\}$ and $t \in \{1, \dots, l+1\}$ be such that $k \in \omega(t)$. Then, we have:

$$z_k = \sum_{i=t}^{l} \lambda_i.$$

Proof. Substitute the v_i in the expression of x using (3.3):

$$x = \sum_{i=0}^{l} \lambda_i v_i = \lambda_0 v_0 + \sum_{i=1}^{l} \lambda_i \left(v_0 + \sum_{j=1}^{i} \lambda_i w_j \right) = \sum_{i=0}^{l} \lambda_i v_0 + \sum_{i=1}^{l} \sum_{j=1}^{i} \lambda_i w_j. \tag{3.4}$$

Observe that $\sum_{i=0}^{l} \lambda_i v_0 = \left(\sum_{i=0}^{l} \lambda_i \right) v_0 = v_0$ and that $v_0 = y$. Using these observations, we get from (3.4) an expression of z in terms of the w_j and the λ_i:

$$z = x - y = \sum_{i=1}^{l} \sum_{j=1}^{i} \lambda_i w_j = \sum_{j=1}^{l} \left(\sum_{i=j}^{l} \lambda_i \right) w_j. \tag{3.5}$$

Fix $k \in \{1, \dots, d\}$. We get from (3.5):

$$z_k = \langle z, e_k \rangle = \left\langle \sum_{i=1}^{l} \left(\sum_{j=1}^{i} \lambda_i \right) w_j, e_k \right\rangle = \sum_{j=1}^{l} \left(\sum_{i=j}^{l} \lambda_i \right) \langle w_j, e_k \rangle . \tag{3.6}$$

Recall that because $d+1$ lies in $\omega(l+1)$, the vectors $w_1, \dots, w_l$ are sums of vectors in the canonical basis of $\mathbb{R}^d$. Because k is assumed to lie in the part $\omega(t)$, it follows that:

$$\langle w_j, e_k \rangle = \begin{cases} 1 & \text{if } j = t, \\ 0 & \text{otherwise.} \end{cases}$$

Thus, all terms $\left(\sum_{i=j}^{l} \lambda_i\right) \langle w_j, e_k \rangle$ in the sum in (3.6) are 0 except for $j = t$. We finally get:

$$z_k = \sum_{i=t}^{l} \lambda_i \langle w_t, e_k \rangle = \sum_{i=t}^{l} \lambda_i.$$

□

We are now ready to prove Proposition 3.3.19. As we mentioned before, we prove Proposition 3.3.19 by double inclusion. To prove the first inclusion, we show that the points in the simplex τ satisfy the chain of inequalities (3.2). For this, we use Lemma 3.3.20. To prove the other inclusion we show that the points x that satisfy the chain of inequalities (3.2) lie in the simplex τ. For this, we will construct $\lambda_0, \dots, \lambda_l$ and show that these are the barycentric coordinates of x in τ.

Proof (of Proposition 3.3.19). We will show by double inclusion that the points in the simplex τ and the subset of $\mathbb{R}^d$ that satisfies the chain of inequalities (3.2) are equal.

Proof of ($\subseteq$). Take a point $x \in \sigma(y, \omega)$. We will now show that $z = x - y = (z_1, \dots, z_d)$ satisfies the chain of inequalities (3.2). For this, we need to show the following three claims:

1. If $k, k' \in \{1, \dots, d\}$ both lie in the same $\omega(t)$ for some $t \in \{1, \dots, l+1\}$, then $z_k = z_{k'}$.
2. If $k, k' \in \{1, \dots, d\}$ lie in some $\omega(t)$ and $\omega(t')$ for some $t, t' \in \{1, \dots, l+1\}$ such that $t < t'$, then $z_k \geq z_{k'}$.
3. If $k \in \{1, \dots, d\}$ lies in $\omega(l+1)$, then $z_k = 0$.

Since x lies in the simplex τ, we can express x using the barycentric coordinates $\lambda_0, \dots, \lambda_l \in [0, 1]$ such that $\sum_{i=0}^{l} \lambda_i = 1$ and:

$$x = \sum_{i=0}^{l} \lambda_i v_i.$$

We will now use Lemma 3.3.20 to prove the three claims in the beginning of the proof.

1. Let $k, k' \in \{1, \dots, d\}$ be such that both k and k' lie in the same $\omega(t)$ for some $t \in \{1, \dots, l+1\}$. From Lemma 3.3.20 we get:

   $$z_k = \sum_{i=t}^{l} \lambda_i = z_{k'},$$

   as desired.

2. Let $k, k' \in \{1, \dots, d\}$ be such that k lies in $\omega(t)$ and k' lies in $\omega(t')$ for some $t, t' \in \{1, \dots, l+1\}$ such that $t < t'$. From Lemma 3.3.20 and the positivity of $\lambda_0, \dots, \lambda_l$ we get:

   $$z_k = \sum_{i=t}^{l} \lambda_i > \sum_{i=t'}^{l} \lambda_i = z_{k'},$$

   as desired.

3. Let $k \in \{1, \dots, d\}$ be such that k lies in $\omega(l+1)$. The sum in the expression of z_k in Lemma 3.3.20 is empty. Hence this sum is 0. Therefore we get $z_k = 0$, as desired.

With these three claims shown, we conclude that the points in the simplex τ satisfy the chain of inequalities (3.2).

Proof of ($\supseteq$). Let x be such that $z = x - y$ satisfies the chain of inequalities (3.2). Define $\lambda_0, \dots, \lambda_l \in \mathbb{R}$ as follows:

$$\begin{aligned} &\lambda_0 = 1 - z_{q_{m_1}} \\ &\lambda_i = z_{q_{m_i}} - z_{q_{m_{i+1}}} \quad \text{for all } i \in \{1, \dots, l\}. \end{aligned}$$

Since z satisfies the chain of inequalities (3.2), it follows that $\lambda_0, \dots, \lambda_l$ all lie in $(0, 1]$. Moreover, we have:

$$\sum_{i=0}^{l} \lambda_i = (1 - z_{q_{m_1}}) + (z_{q_{m_1}} - z_{q_{m_2}}) + \ldots + (z_{q_{m_l}} - z_{q_{m_{l+1}}}) = 1 - z_{q_{m_{l+1}}} = 1,$$

since $z_{q_{m_{l+1}}} = z_{q_{d+1}} = 0$.

We will now show that $\sum_{i=0}^{l} \lambda_i v_i = x$. Let $k \in \{1, \dots, d+1\}$ and $t \in \{1, \dots, l+1\}$ be such that $k \in \omega(t)$. By definition of ω and the chain of inequalities (3.2), from $k \in \omega(t)$ it follows $z_k = z_{q_{m_t}}$. From the definition of $\lambda_0, \dots, \lambda_l$, it follows that:

$$z_k = z_{q_{m_t}} = z_{q_{m_t}} - z_{q_{m_{l+1}}} = (z_{q_{m_t}} - z_{q_{m_{t+1}}}) + \ldots + (z_{q_{m_l}} - z_{q_{m_{l+1}}}) = \sum_{i=t}^{l} \lambda_i. \tag{3.7}$$

From the chain of inequalities (3.2) and the definition of ω, we also get that for all $k \in \omega(l+1)$ we have $z_k = 0$. With the help of (3.7), we can therefore write z as:

$$z = \sum_{t=1}^{l} \sum_{k \in \omega(t)} z_k e_k = \sum_{t=1}^{l} \sum_{k \in \omega(t)} \left(\sum_{i=t}^{l} \lambda_i \right) e_k = \sum_{t=1}^{l} \left(\sum_{i=t}^{l} \lambda_i \right) \sum_{k \in \omega(t)} e_k = \sum_{t=1}^{l} \left(\sum_{i=t}^{l} \lambda_i \right) w_t.$$

We change the order of summation:

$$z = \sum_{t=1}^{l} \sum_{i=t}^{l} \lambda_i w_t = \sum_{i=1}^{l} \lambda_i \left(\sum_{t=1}^{i} w_t \right).$$

We now add v_0 to both sides of the equation. Observe that $x = v_0 + z$ and $\sum_{i=0}^{l} \lambda_i = 1$. We get:

$$x = \sum_{i=0}^{l} \lambda_i v_0 + \sum_{i=1}^{l} \lambda_i \left(\sum_{t=1}^{i} w_t \right) = \lambda_0 v_0 + \sum_{i=1}^{l} \lambda_i \left(v_0 + \sum_{t=1}^{i} w_t \right) = \lambda_0 v_0 + \sum_{i=1}^{l} \lambda_i v_i = \sum_{i=0}^{l} \lambda_i v_i.$$

Since all $\lambda_0, \dots, \lambda_l$ belong to $(0, 1]$ and their sum is 1, we conclude that x lies in the simplex τ.

With both inclusions proven, we conclude that the simplex τ and the subset of $\mathbb{R}^d$ defined by the chain of inequalities (3.2) are equal. □

3.4 Point location in the Freudenthal-Kuhn triangulation

In this section, we will generalize the point location algorithm from Section 3.1.1 with the help of the permutahedral representation from Definition 3.3.15.

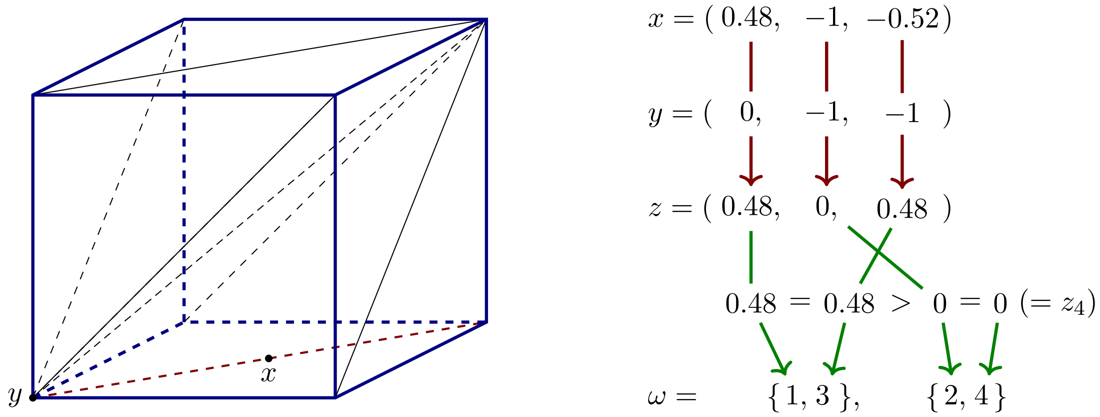


Figure 3.14: Scheme of the execution of the point location algorithm called on a point $x \in \mathbb{R}^3$. The output of the algorithm consists of the edge $\sigma(y, \omega)$ (coloured red in the cube on the left) with y and ω as shown on the right. The edge $\sigma(y, \omega)$ is the smallest simplex in the Freudenthal-Kuhn triangulation of $\mathbb{R}^3$ that contains the point x.

Description of the algorithm.

Remark 3.4.1. *Similarly to Section 3.3.7, the simplices in this section are open in their affine hull.*

The point location algorithm takes as input a point $x \in \mathbb{R}^d$ given as a vector of Cartesian coordinates $(x_1, \ldots, x_d)$. The output consists of the permutahedral representation $\sigma(y, \omega)$ of the unique simplex τ of some arbitrary dimension in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ that contains the input point x (illustrated in Figure 3.14).

Remark 3.4.2. *Note that the simplices that are open in their affine hull form a partition of $\mathbb{R}^d$. So, unlike the algorithm in Section 3.1.1, the point location algorithm that we present here has a uniquely defined output.*

The point location algorithm first computes the following two vectors:

$$y = (y_1, \ldots, y_d) = (\lfloor x_1 \rfloor, \ldots, \lfloor x_d \rfloor) \in \mathbb{Z}^d$$
$$z = (z_1, \ldots, z_d) = x - y = (\{x_1\}, \ldots, \{x_d\}) \in [0, 1)^d,$$

where each $\{x_i\}$ denotes the fractional part of x_i. The coordinates $z_1, \ldots, z_d$ of the vector z are then sorted as follows:

$$1 > z_{q_1} \geq \ldots \geq z_{q_d} \geq 0, \tag{3.8}$$

where $q_1, \ldots, q_d$ are such that $(q_1, \ldots, q_d)$ is a permutation of $\{1, \ldots, d\}$. By convention we pose:

$$q_{d+1} = d + 1 \quad \text{and} \quad z_{q_{d+1}} = 0.$$

Let $m_1, \ldots, m_{l+1} \in \{1, \ldots, d + 1\}$ be such that the chain of inequalities (3.8) is rewritten as:

$$1 > z_{q_1} = \ldots = z_{q_{m_1}} > z_{q_{m_1+1}} = \ldots = z_{q_{m_2}} > \ldots\ldots > z_{q_{m_l}+1} = \ldots = z_{q_{d+1}} = 0, \tag{3.9}$$

and $m_{l+1} = d + 1$.

The point location algorithm outputs the permutahedral representation $\sigma(y, \omega)$, where $y \in \mathbb{Z}^d$ is the vector computed earlier and ω is the following ordered partition of $\{1, \ldots, d + 1\}$ in $l + 1$ parts:

$$\omega = \{q_1, \ldots, q_{m_1}\}, \{q_{m_1+1}, \ldots, q_{m_2}\}, \ldots, \{q_{m_{l-1}+1}, \ldots, q_{m_l}\}, \{q_{m_l+1}, \ldots, q_{d+1}\}.$$

Correctness of the algorithm. We will now show that the point location algorithm outputs the permutahedral representation of the simplex in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ that contains the input point $x \in \mathbb{R}^d$.

Theorem 3.4.3 (Correctness of the point location algorithm). *The simplex $\sigma(y, \omega)$ given in the output by the point location algorithm called on a point $x \in \mathbb{R}^d$ is the smallest simplex in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ that contains the input point x.*

The proof of Theorem 3.4.3 relies on Proposition 3.3.19 that we showed in Section 3.3.7.

Proof (of Theorem 3.4.3). From Proposition 3.3.19, it follows that the points in the simplex τ satisfy the following chain of inequalities:

$$1 > z_{q_1} = \ldots = z_{q_{m_1}} > z_{q_{m_1+1}} = \ldots = z_{q_{m_2}} > \ldots\ldots > z_{q_{m_l}+1} = \ldots = z_{q_{d+1}} = 0, \tag{3.10}$$

where $q_1, \ldots, q_{d+1}$ are such that ω is written as:

$$\omega = \{q_1, \ldots, q_{m_1}\}, \{q_{m_1+1}, \ldots, q_{m_2}\}, \ldots, \{q_{m_{l-1}+1}, \ldots, q_{m_l}\}, \{q_{m_l+1}, \ldots, q_{d+1}\}.$$

Since x satisfies the chain of inequalities (3.10), x lies in the simplex τ. Because the simplices that we consider here form a partition of $\mathbb{R}^d$, the simplex that contains x is unique.

□

Time and space complexity of the algorithm. The time complexity of the point location operation is dominated by the complexity of sorting the coordinates $z_1, \ldots, z_d$. Therefore:

Proposition 3.4.4. *The time complexity of one call of the point location is $O(d \log d)$.*

The only storage that the point location algorithm uses is $O(d)$ to store the vector z and the output. We conclude that:

Proposition 3.4.5. *The space complexity of one call of the point location is $O(d)$.*

3.5 Generation of faces and cofaces in a Freudenthal-Kuhn triangulation

In this section, we will define two algorithms that generate faces and cofaces of a given simplex in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$. Let us first formally define what "generating faces and cofaces" means.

Definition 3.5.1 (Generating simplices). *Let $F = \{\tau_1, \ldots, \tau_n\}$ be a finite ordered set of simplices in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$.* Generating simplices of F *means computing every simplex (in some arbitrary representation) in the set F one after another, each simplex being computed once.*

Remark 3.5.2. *Whenever we say* first simplex *or* last simplex*, we tacitly assume that there exists a deterministic order on the generated simplices.*

We are interested in two examples of sets of simplices to generate: sets $\mathrm{fac}(\tau, k)$ and $\mathrm{cof}(\tau, l)$ from Definitions 3.5.3 and 3.5.4 below.

Definition 3.5.3 (Set of faces)**.** *Let τ be an l-dimensional simplex in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ for some $l \in \{0, \dots, d\}$. Let $k \in \{0, \dots, d\}$ be such that $k \leq l$. We denote by* $\operatorname{fac}(\tau, k)$ *the set of k-dimensional faces of the simplex τ.*

Definition 3.5.4 (Set of cofaces)**.** *Let τ be a k-dimensional simplex in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ for some $k \in \{0, \dots, d\}$. Let $l \in \{0, \dots, d\}$ be such that $k \leq l$. We denote by* $\operatorname{cof}(\tau, l)$ *the set of l-dimensional cofaces of the simplex τ in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$.*

Remark 3.5.5. *In the following, we will follow a convention regarding the notations of the dimensions of simplices. Whenever we need to distinguish two dimensions, we denote by k the smallest and by l the greatest of them. Hence, we always assume the following chain of inequalities: $0 \leq k \leq l \leq d$.*

In Sections 3.5.1 and 3.5.2, we will introduce two algorithms that generate the sets $\operatorname{fac}(\tau, k)$ and $\operatorname{cof}(\tau, l)$ for a given simplex τ and a given integer k or l. These two algorithms are called the *face generation algorithm* and the *coface generation algorithm* respectively. Both face and coface generation algorithms are used in the manifold tracing algorithm in Chapter 4 as subroutines. The time and space complexities of face and coface computation algorithms are given by Theorems 3.5.6 and 3.5.7 below.

Theorem 3.5.6 (Complexity of the face generation algorithm)**.** *Let σ be an l-dimensional simplex in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ for some l. Let $k \leq l$. The time complexity to generate the k-dimensional faces of σ is as follows:*

1. *The complexity of computing the permutahedral representation of the first k-dimensional face of σ is $O(k)$.*
2. *The decision of whether a computed k-dimensional face of σ is the final*[12] *one is obtained in $O(1)$ operations.*
3. *Computing the permutahedral representation of a k-dimensional face of σ from the previous one takes $O(d)$ operations.*

The internal storage needed for the face generation is $O(l)$.

Note that the storage needed for the face generation is $O(k)$, which is less than the $O(d)$ storage of the permutahedral representation. This is because the faces of σ are not stored using permutahedral representation, but using a more compact internal representation (see Section 3.5.1).

Theorem 3.5.7 (Complexity of the coface generation algorithm)**.** *Let τ be a k-dimensional simplex in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ for some k. Let $l \geq k$. The time complexity to generate the l-dimensional cofaces of τ is as follows:*

1. *The complexity of computing the permutahedral representation of the first l-dimensional coface of τ is $O(d)$.*
2. *The decision of whether a computed l-dimensional coface of τ is the final one is obtained in $O(1)$ operations.*

[12] Here, *final* means that all k-dimensional faces of σ have been generated. We will use the word *final* in the same sense when talking about the generation of pemutations, ordered partitions, etc.

3. *Computing the permutahedral representation of an l-dimensional coface from the previous one takes $O(d)$ operations.*

The internal storage needed for the coface generation is $O(d)$.

Similarly to the face generation algorithm, the cofaces of τ are not stored using permutahedral representation, but using another internal representation (see Section 3.5.2).

Remark 3.5.8. *Note that the complexity to compute the permutahedral representation is at least the storage $O(d)$ of the permutahedral representation. In this sense, the face and coface generation algorithms are in fact optimal for the purpose of the manifold tracing algorithm.*

3.5.1 Face generation algorithm

In this section we will describe the face generation algorithm.

Description of the algorithm. The *face generation algorithm* takes as input:

- a record consisting of a vertex $y \in \mathbb{Z}^d$ and of an ordered partition ω of $\{1, \ldots, d+1\}$, such that $\sigma(y, \omega)$ is the permutahedral representation of an l-dimensional simplex τ in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$, for some $l \in \{0, \ldots, d\}$;
- an integer $k \in \{0, \ldots, d\}$ such that $k \leq l$.

Define the $l+1$ vertices $v_0, \ldots, v_l$ of the input simplex τ as in Definition 3.3.1. A k-dimensional face τ' of τ can be uniquely identified by $k+1$ indices $\{m_0, \ldots, m_k\} \subseteq \{0, \ldots, l\}$ of the $k+1$ vertices $v_{m_0}, \ldots, v_{m_k}$ of the face τ'. We use the subset of $k+1$ indices $\{m_0, \ldots, m_k\} \subseteq \{0, \ldots, l\}$ as the internal data structure that stores the currently generated k-dimensional face in the face generation algorithm (see Figure 3.15).

Definition 3.5.9. *We will refer to this subset of $\{0, \ldots, l\}$ of size $k+1$ as an* internal representation *of the k-dimensional face of τ.*

To fully describe the face generation algorithm, we need to discuss two of its aspects:

- The *transition* from a k-dimensional face of τ to the next one (red arrows in Figure 3.15). By transition, we mean computing the internal representation of a k-dimensional face of τ from the internal representation of the previously computed one.
- The *transformation* of the internal representation of the k-dimensional face of τ to its permutahedral representation (green arrows in Figure 3.15).

We will now describe these two aspects of the face generation algorithm in detail.

Transition. Let us start with the transition from a k-dimensional face of the input simplex τ to the next one. As we stated above, the k-dimensional faces of τ are stored internally as subsets of $k+1$ indices in $\{0, \ldots, l\}$. Generating subsets of a set with a fixed number of elements is a classical problem in combinatorial generation and there exists an algorithm for subset generation of Ehrlich [Ehr73] that has the following complexities:

1. The complexity of computing the first subset of size $k+1$ of $\{0, \ldots, l\}$ is $O(k)$.

2. The decision of whether a computed subset of size $k+1$ of $\{0, \dots, l\}$ is the final one is obtained in $O(1)$ operations.

3. Computing a subset of size $k+1$ of $\{0, \dots, l\}$ from the previous one takes $O(1)$ operations.

Remark 3.5.10. *A generation algorithm that has a constant worst-case transition time complexity is called* loopless *[Ehr73].*

Every time we need to compute a new k-dimensional face, we use the algorithm of Ehrlich [Ehr73] as a black box that generates the new subset of $k+1$ indices (illustrated by red arrows in Figure 3.15).

Lemma 3.5.11 below shows that with this transition procedure, the face generation algorithm satisfies the complexities of 1 and 2 in Theorem 3.5.6. It will also help to show later that the complexity of 3 in Theorem 3.5.6 is also satisfied.

Lemma 3.5.11. *The time complexity of computing the first k-dimensional face is $O(k)$. The time complexities of:*

1. *the decision of whether a computed k-dimensional face is the final one,*

2. *computing the internal representation of a k-dimensional face of the input simplex τ from the internal representation of the previous one*

are both $O(1)$.

Proof. The proof follows from the time complexities of the algorithm for the subset generation by Ehrlich [Ehr73]. □

Transformation of the internal representation into the permutahedral representation. Now we will describe how the subset of $k+1$ indices $\{m_0, \dots, m_k\} \subseteq \{0, \dots, l\}$ can be transformed into the permutahedral representation of the corresponding face τ' of the input simplex τ.

There are two steps in the transformation procedure.

1. Ehrlich's subset generation algorithm does not necessarily store the indices $m_0, \dots, m_k$ in ascending order. Therefore we first need to sort the indices $m_0, \dots, m_k$. For this we perform the *counting sort algorithm*, introduced by Seward [Sew54]. Its time and space complexities are both $O(l)$. We will assume in the following that the order of the indices is the following:
$$m_0 < \dots < m_k.$$

2. After sorting the indices $m_0, \dots, m_k$, we use the construction from Section 3.3.3. We assign:
$$y' = v_0 + \sum_{j \in \omega(1)} e_j + \dots + \sum_{j \in \omega(m_0)} e_j$$
and:
$$\begin{aligned} \omega'(i) &= \omega(m_{i-1}+1) \cup \dots \cup \omega(m_i), && \text{if } i \in \{1, \dots, k\}, \\ \omega'(k+1) &= (\omega(1) \cup \dots \cup \omega(m_0)) \cup (\omega(m_k+1) \cup \dots \cup \omega(l+1)). \end{aligned}$$
The output consists of the permutahedral representation $\sigma(y', \omega')$.

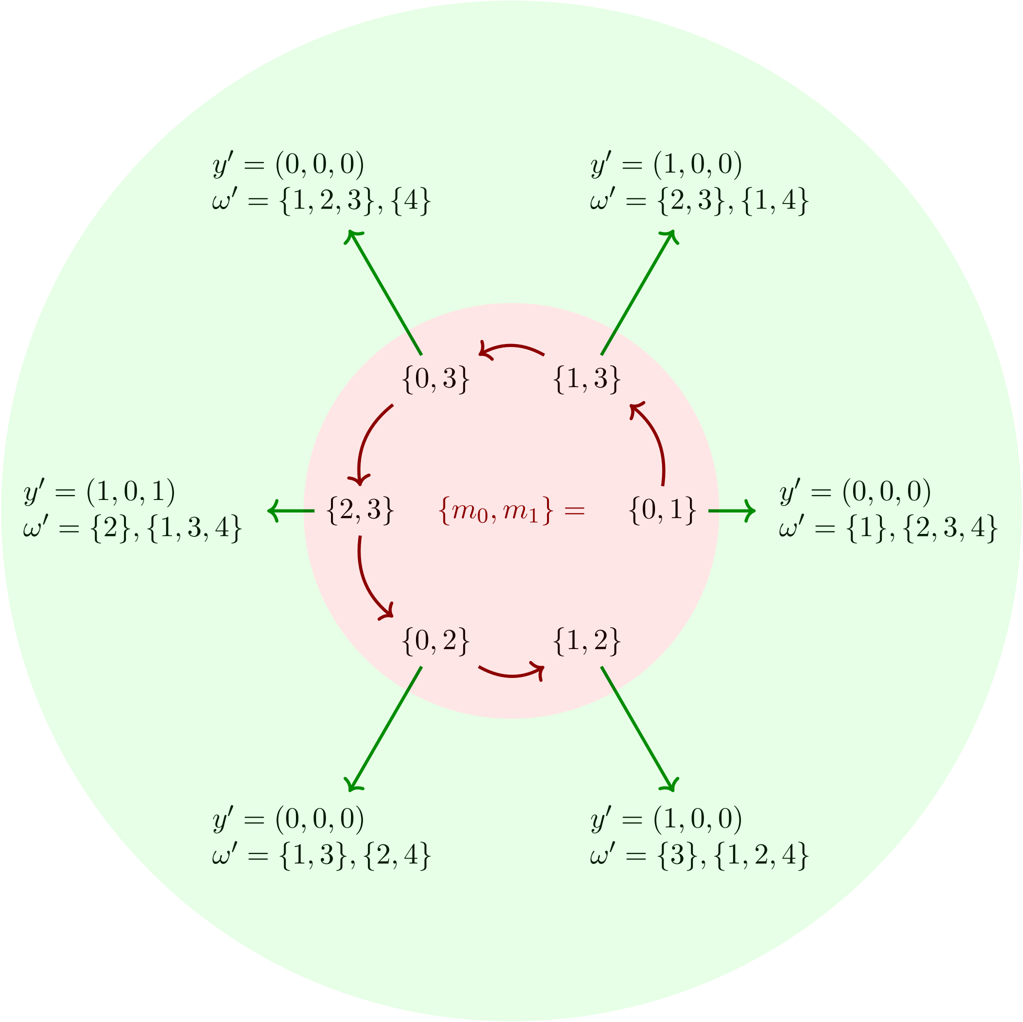

$y' = (0,0,0)$
$\omega' = \{1,2,3\},\{4\}$
$y' = (1,0,0)$
$\omega' = \{2,3\},\{1,4\}$
$\{0,3\}$
$\{1,3\}$
$y' = (1,0,1)$
$\omega' = \{2\},\{1,3,4\}$
$\{2,3\}$
$\{m_0, m_1\} =$
$\{0,1\}$
$y' = (0,0,0)$
$\omega' = \{1\},\{2,3,4\}$
$\{0,2\}$
$\{1,2\}$
$y' = (0,0,0)$
$\omega' = \{1,3\},\{2,4\}$
$y' = (1,0,0)$
$\omega' = \{3\},\{1,2,4\}$

Figure 3.15: Scheme of the execution of the face generation algorithm called on a 3-dimensional simplex $\tau = \sigma(y, \omega)$ in the Freudenthal-Kuhn triangulation of $\mathbb{R}^3$ with $y = (0,0,0)$ and $\omega = \{1\},\{3\},\{2\},\{4\}$, and $k = 1$. The inner red circle contains the subsets of 2 elements of $\{0,1,2,3\}$. The sequence of the subsets is given by the algorithm of Ehrlich [Ehr73]. The outer green circle contains y' and ω' in the permutahedral representation of the corresponding edge $\sigma(y', \omega')$ of τ. The green arrows indicate the transformation of the internal representations of a k-dimensional faces of τ into their permutahedral transformations.

Note that the permutahedral representation $\sigma(y', \omega')$ is well-defined. Indeed, because $\sigma(y, \omega)$ is the permutahedral representation of τ', we have $d + 1 \in \omega(l + 1)$. By construction of ω', we have $\omega(l + 1) \subseteq \omega'(k + 1)$. It follows that $d + 1 \in \omega'(k + 1)$. Hence $\sigma(y', \omega')$ is well-defined permutahedral representation.

From Lemma 3.3.7 and the remark above, it follows that the algorithm above is correct. By this, we mean that the simplex $\sigma(y', \omega')$ is the k-dimensional face of the input simplex τ defined by the vertices $v_{m_0}, \dots, v_{m_k}$.

Lemma 3.5.11 below shows that the time complexity of the transformation procedure is $O(d)$. This lemma shows that the face generation algorithm satisfies the complexity 3 of Theorem 3.5.6.

Lemma 3.5.12 (Time complexity of the transformation procedure)**.** *The time complexity of transforming the subset of $k + 1$ indices $\{m_0, \dots, m_k\} \subseteq \{0, \dots, l\}$ into the permutahedral representation of the corresponding face τ' of the input simplex τ is $O(d)$.*

Proof. We will decompose the overall time complexity into three parts:

1. the complexity of applying the counting sort algorithm on the indices $m_0, \dots, m_k$,
2. the complexity of assigning y' to its value and
3. the complexity of assigning ω' to its value.

The complexity of the counting sort algorithm is the sum of (see [Sew54]):

- the difference between the largest m_k and the smallest element m_0 in the set $\{m_0, \dots, m_k\}$,
- the number $k + 1$ of elements to sort.

Since all indices are bounded between 0 and l, the difference $m_k - m_0$ is at most l. By assumption, we have $k \leq l$. We conclude that the time complexity of applying the counting sort algorithm on the indices $m_0, \dots, m_k$ is $O(l)$.

Now, we will analyse the complexity of assigning y' to its value. The new value of y' is obtained by adding at most $O(d)$ canonical basis vectors and eventually the vector e_{d+1} to v_0. Adding a canonical basis vector amounts to adding 1 to the corresponding coordinate. This counts as an elementary operation. Adding the vector e_{d+1} amounts to subtracting 1 from all d coordinates, which has complexity $O(d)$. We conclude that computing the new value of y' has overall time complexity $O(d)$.

Finally, we will analyse the complexity of assigning ω' to its value. Computing the new value of ω' amounts to reading the indices in $\omega(1), \dots, \omega(k + 1)$ and inserting these indices in the appropriate sets $\omega'(i)$. There are $d + 1$ indices overall, therefore, the complexity is $O(d)$.

When we sum the three complexities, we get that the overall time complexity of the algorithm is:

$$O(l) + O(d) + O(d) = O(d).$$

□

Time and space complexity. Thanks to Lemmas 3.5.11 and 3.5.12, we can conclude that the face generation algorithm satisfies the time complexities in Theorem 3.5.6.

The additional storage in the face generation algorithm come from:

- the space complexity of the subset generation algorithm by Ehrlich, which is $O(k)$,

- the space complexity of the counting sort, which is $O(l)$.

The total space complexity is $O(l)$ [Ehr73]. Therefore, the face generation algorithm satisfies the space complexity in Theorem 3.5.6.

To compute the permutahedral representations of all k-dimensional faces of the input simplex, the total complexity is the following.

Theorem 3.5.13. *Let τ be an l-dimensional simplex in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$, for some $l \in \{0, \dots, d\}$. Let $k \in \{0, \dots, d\}$ be such that $k \leq l$. Let:*

$$F(\tau, k) = \binom{l+1}{k+1}$$

be the number of k-dimensional faces of the simplex τ. The time complexity of computing the permutahedral representations of all k-dimensional faces of τ is $O(dF(\tau, k))$.

Proof. The time complexity to compute one face by the face generation algorithm is $O(1)$ (see Lemma 3.5.11). The time complexity of computing the permutahedral representation of a generated face is $O(d)$ (see Lemma 3.5.12). Therefore, the total time complexity spent per computed face is $O(d)$. By multiplying this time complexity by the number $F(\tau, k)$ of k-dimensional faces of τ, we get the result. □

Remark 3.5.14. *Since the space complexity to store a permutahedral representation is $\Theta(d)$, the bound on the time complexity in Theorem 3.5.13 is optimal.*

3.5.2 Coface generation algorithm

In this section, we will describe the coface generation algorithm.

Description of the coface generation algorithm. The coface generation algorithm takes as input:

- a record consisting of a vertex $y \in \mathbb{Z}^d$ and an ordered partition ω of $\{1, \dots, d+1\}$ in $k+1$ parts, such that $\sigma(y, \omega)$ is the permutahedral representation of a k-dimensional simplex τ in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$, for some $k \in \{0, \dots, d\}$;
- an integer $l \in \{0, \dots, d\}$ such that $l \geq k$.

The coface generation algorithm then generates the set $\mathrm{cof}(\tau, l)$ of the l-dimensional cofaces of τ in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ (see Definition 3.5.4).

Remark 3.5.15. *In the following, when we mention cofaces of τ, we will tacitly assume that these cofaces lie in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$.*

Remark 3.5.16. *Note that the dimension of the simplex τ in the description of the coface generation algorithm is denoted by k, while the dimension of its cofaces is l. This changes with respect to the description of the face generation algorithm (Section 3.5.1), where the dimension of the input simplex was denoted by l, while the dimension of its faces was k. This choice of notations is consistent with the convention in Remark 3.5.5 that k denotes the smallest dimension, while l denotes the greatest dimension.*

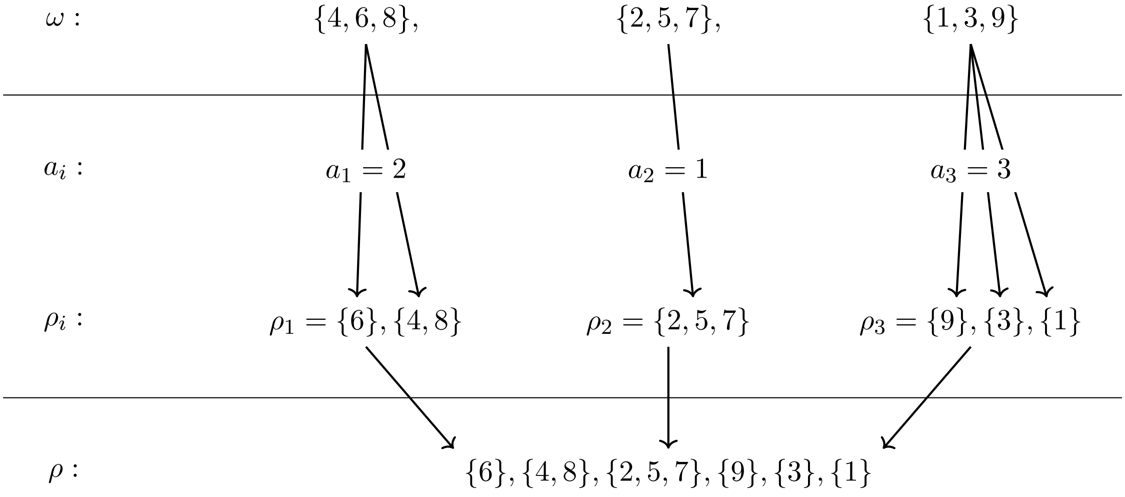


Figure 3.16: A refinement ρ of an ordered partition ω and the tuple $(a, \rho_1, \rho_2, \rho_3)$ that is in bijection. Here $k+1 = 3$ is the number of parts in ω, and $l+1 = 6$ is the number of parts in ρ.

The generated l-dimensional cofaces are represented in the following way. Recall that by definition of the permutahedral representation, y is a vertex of the input simplex τ. All cofaces of τ (including the l-dimensional cofaces that interest us) share the vertex y. Therefore all simplices in $\mathrm{cof}(\tau, l)$ that the coface generation algorithm generates lie in $\mathrm{star}(y)$. From Theorem 3.3.10 it follows that each simplex τ' in $\mathrm{cof}(\tau, l)$ is in one-to-one correspondence with a refinement ρ of the input ordered partition ω into $l+1$ parts. We encode each such refinement ρ using a record $(a, \rho_1, \dots, \rho_{k+1})$ that consists of (see Figure 3.16):

- An integer composition $a = a_1, \dots, a_{k+1}$ of $l+1$ in $k+1$ terms (see Definition 3.5.17 below). Each a_i is at most the cardinality $|\omega(i)|$ and represents the number of subparts in which we need to subdivide $\omega(i)$ to obtain the corresponding parts in the refinement ρ.

- Refinements $\rho_1, \dots, \rho_{k+1}$ of $\omega(1), \dots, \omega(k+1)$ into $a_1, \dots, a_{k+1}$ parts respectively. Each refinement ρ_i represents the specific way to subdivide $\omega(i)$ to obtain the corresponding parts in the refinement ρ.

We will now formally define integer compositions.

Definition 3.5.17 (Integer composition)**.** *Let n and k be two positive integers.*

A composition *of a given positive integer n in k terms is a sequence of positive integers $a_1, \dots, a_k$, such that $n = a_1 + \dots + a_k$.*

Given positive integers $b_1, \dots, b_k$, a bounded composition *of n in k terms is a composition of n in k terms $a_1, \dots a_k$ that are upper bounded by $b_1, \dots, b_k$ respectively:*

$$\forall i \in \{1, \dots, k\}, \; a_i \leq b_k.$$

To generate the entries of the record $(a, \rho_1, \dots, \rho_d)$, the coface generation algorithm uses the following classical algorithms as black boxes.

- The integer compositions a of $l+1$ in $k+1$ terms bounded by $|\omega(1)|, \dots, |\omega(k+1)|$ are generated by the algorithm of Walsh [Wal00]. This algorithm is loopless (in the sense of Remark 3.5.10), which is important for the coface generation algorithm to satisfy the complexities of Theorem 3.5.7.

- We also want to generate the ordered partitions $\rho_1, \ldots, \rho_{k+1}$. It seems that there are no *ordered* set partition generation algorithms in the literature. However, it is not hard to make an ordered set partition generation algorithm from an *unordered* set partition generation algorithm (such as the algorithm of Ruskey and Savage [RS94]). This is done in the following way.

  In practice, when an algorithm generates an unordered set partition, it stores the parts in memory in some specific order: $\hat{\rho}_i(1), \ldots, \hat{\rho}_i(a_i)$. Such $\hat{\rho}_i$ can be seen as ordered set partitions with some specific order on the parts. We want all possible orders on these parts. For this, we generate separately from the unordered set partitions the permutations $\sigma_i : \{1, \ldots, a_i\} \to \{1, \ldots, a_i\}$ using the algorithm of Ehrlich [Ehr73]. The ordered partition ρ_i can be seen as a combination of $\hat{\rho}_i$ given by the unordered set partition generation algorithm on the parts and a permutation σ_i in the following way:

  $$\rho_i = \hat{\rho}_i(\sigma_i(1)), \ldots, \hat{\rho}_i(\sigma_i(a_i)).$$

  Both generation algorithms of Ruskey and Savage [RS94] and of Ehrlich [Ehr73] are loopless, which will be important in the complexity analysis of the coface generation algorithm later on.

We will now describe the following aspects of the coface computation algorithm:

- initialization,
- transition from one computed l-dimensional coface of τ to the next one,
- decision of whether a computed coface is the final one or not,
- transformation of the record $(a, \rho_1, \ldots, \rho_{k+1})$ to the permutahedral representation $\sigma(y', \rho')$ of the corresponding l-dimensional coface of τ.

Initialization. Recall that the elements of the stored record are:

$$a,\ \rho_1,\ \rho_2,\ \ldots,\ \rho_{k+1}.$$

It is important that the entry a is initialized first, since the sizes of $\rho_1,\ \rho_2,\ \ldots,\ \rho_{k+1}$ depend on a. The order of initialization of the entries other than a can be arbitrary.

Lemma 3.5.18. *The time complexity of the initialization of the record $(a,\ \rho_1,\ \rho_2,\ \ldots,\ \rho_{k+1})$ is $O(d)$.*

Proof. By the choice of the algorithms to generate $a,\ \rho_1,\ \rho_2,\ \ldots,\ \rho_{k+1}$, their initialization time complexities are equal to the respective storage sizes. These sizes are the following. The storage of a consists of $k+1$ integers $a_1, \ldots, a_k + 1$. Hence the storage size of a is $O(k)$.

For all $i \in \{1, \ldots, k+1\}$, the storage size of ρ_i is the sum of the storage sizes of $\hat{\rho}_i$ and σ_i. The storage size of $\hat{\rho}_i$ is $O(|\omega(i)|)$ and the storage size of σ_i is $O(a_i)$. Because the sum $|\omega(1)| + \ldots + |\omega(k+1)|$ is $d+1$, the total storage size of all ordered partitions $\hat{\rho}_i$ is therefore $O(d)$. Because the sum $a_1 + \ldots + a_{k+1}$ is $l+1$, the total storage size of all permutations σ_i is therefore $O(l)$. Thus, the total time complexity of the initialization phase is:

$$O(k + d + l) = O(d).$$

□

Transition. Now, we will describe how the coface generation algorithm computes the next l-dimensional coface of τ from the previous one. As we stated above, an l-dimensional coface of τ is stored as a record $(a,\ \rho_1,\ \rho_2,\ \dots,\ \rho_{k+1})$. To make a transition from one l-dimensional coface to another, the algorithm updates the entries of the stored record in the reverse order from the initialization:

$$\rho_{k+1},\ \dots,\ \rho_2,\ \rho_1,\ a.$$

The updates are done in the following way. If all entries in the record have the final values in the respective generating algorithms ([Ehr73, RS94] for ρ_i and [Wal00] for a), then the corresponding l-dimensional coface is the final one. This can be decided in $O(k)$ operations (see Lemma 3.5.20 below). In this case, the coface generation algorithm terminates.

Otherwise, take the leftmost entry o in the list above such that it does not have its final value in the corresponding generating algorithm.

- If the entry o is not a, then update o to its next value. This means that $o = \rho_i$ for some $i \in \{1, \dots, k+1\}$. Because the algorithms of Ruskey and Savage [RS94] and of Ehrlich [Ehr73] are loopless, this update takes $O(1)$ operations in the worst case. Then, reinitialize all entries that are to the left from ρ_i in the list above (which are $\rho_{k+1}, \dots, \rho_{i+1}$) to the first values in their respective generation algorithms. The total time complexity of the reinitialization is bounded by the time complexity $O(d)$ from Lemma 3.5.18.

- If the entry o is a, then the algorithm updates a to the next value given by the algorithm by Walsh [Wal00]. Since this algorithm is loopless, $O(1)$ changes in a are made in the worst case. The entries $\rho_1,\ \rho_2,\ \dots,\ \rho_{k+1}$ are then reinitialized to the first values in their respective generation algorithms. Again, the total time complexity of the reinitialization is bounded by the time complexity $O(d)$ from Lemma 3.5.18.

From the description above we conclude the time complexity of computing an l-dimensional coface of τ from the previous one.

Lemma 3.5.19. *Let τ be a k-dimensional simplex in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$. The time complexity of computing an l-dimensional coface of τ from the previous one by the coface generation algorithm is $O(d)$.*

Deciding if the current coface is the last. We will now describe how the coface generation algorithm decides whether the currently computed coface is last or not. During a transition from one coface to another, the algorithm checks whether the entries:

$$\rho_{k+1},\ \dots,\ \rho_2,\ \rho_1,\ a$$

in the stored record have the final value in their respective generating algorithms. If all $k+2$ entries have the final values, then the algorithm signals that the currently stored record represents the last coface, and terminates. The complexity of deciding if the current coface is the final one is hence the following:

Lemma 3.5.20. *The time complexity of deciding whether the last computed coface by the coface generation algorithm is the final one is $O(k)$.*

Transformation into the permutahedral representation. Now, we will describe an algorithm that computes the permutahedral representation of the l-dimensional coface τ' of τ from the stored record:

$$a,\ \rho_1,\ \rho_2,\ \ldots,\ \rho_{k+1}.$$

This Eaves notation of τ' will be transformed into the permutahedral representation of τ' in the next step. This algorithm consists of two major steps.

1. First, we transform the record above to the specific generalized Eaves notation $\varphi(y, \rho)$ of τ' such that:

   - the vertex y is the same as in the input,
   - ρ is some refinement of ω in $l + 1$ parts.

   The vertex y is available in the input, so there is nothing to be done to find it. The ordered partition ρ is the concatenation (Definition 1.3.5) of the ordered partitions $\rho_1, \ldots, \rho_{k+1}$. Therefore, computing ρ has the time complexity of appending $d+1$ elements to an ordered partition, which is $O(d)$.

2. Then, from y and ρ we find a vertex y' and another ordered partition ρ' such that $\sigma(y', \rho')$ is the permutahedral representation of τ'. Finding the permutahedral representation from an Eaves notation is done in the same way as in the proof of Corollary 3.3.14. See also Figure 3.17 for an illustration of Eaves notations of cofaces and the permutahedral representations side by side.

   Let $v_0, \ldots, v_l$ be the vertices of the l-dimensional coface τ' of τ that are defined as in Definition 3.3.1 from the Eaves notation $\sigma(y, \rho)$:

$$\begin{aligned} v_0 &= y \\ v_i &= v_{i-1} + \sum_{j \in \rho(i)} e_j, \quad \text{for } i \in \{1, \ldots, l\}. \end{aligned}$$

   Let $t \in \{1, \ldots, l+1\}$ be an index such that $d+1 \in \rho(t)$. The index t can be found by a scan of the elements in the parts of the ordered partition ρ. The time complexity of one such scan is $O(d)$. By Definition 3.3.15, the permutahedral representation of τ is $\sigma(v', \rho')$ with $v' = v_t$ and $\rho' = \rho \oplus t$. Now, we will describe the describe the time complexities of computing v' and ρ'.

   - To find the vertex v', the algorithm adds to y the $O(d)$ vectors e_i for all $i \in \rho(1) \cup \ldots \cup \rho(t)$. Each such addition takes one addition of integers, unless $i = d+1$. In the latter case, the addition has complexity $O(d)$. Overall, to compute the vertex v', the algorithm takes $O(d)$ operations.
   - To find the ordered partition ρ', the algorithm copies the contents of the parts in the ordered partition ρ in the order $\rho(t+1), \ldots, \rho(l+1), \rho(1), \ldots, \rho(t)$. This is done in $O(d)$ operations.

From the description above, we find the time complexity of computing the permutahedral representation from the corresponding record stored by the coface generation algorithm.

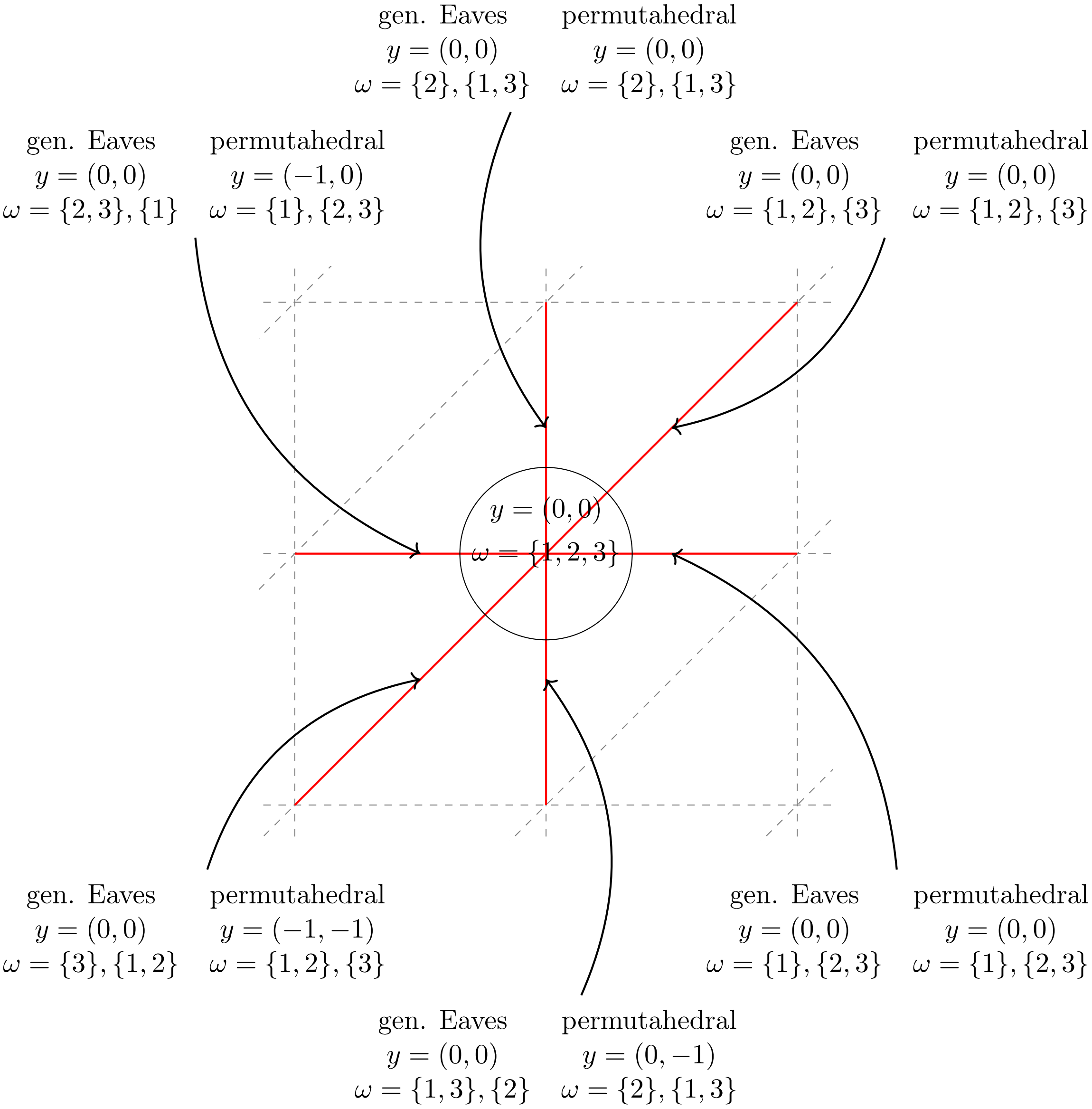


Figure 3.17: Generalized Eaves notations and the permutahedral representations of the one-dimensional cofaces of the vertex at the origin in the Freudenthal-Kuhn triangulation of $\mathbb{R}^2$. The ordered partitions in Eaves notations are the refinements of $\{1,2,3\}$ into 2 parts.

Lemma 3.5.21. *Let τ be a simplex in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$. Transforming the record with entries:*

$$a,\ \rho_1,\ \rho_2,\ \ldots,\ \rho_{k+1}$$

stored by the coface generation algorithm into the permutahedral representation of the corresponding l-dimensional coface of τ takes time $O(d)$.

Correctness of the algorithm. Now, we will show that the coface generation algorithm is correct. From Proposition 3.5.22 below, it follows that each l-dimensional coface of τ is generated by the coface generation algorithm exactly once.

From Theorem 3.3.10 it follows that each simplex τ' in $\operatorname{cof}(\tau, l)$ is in one-to-one correspondence with a refinement ρ of the input ordered partition ω into $l+1$ parts. By construction these refinements are in one-to-one correspondence with the records $(a, \rho_1, \ldots, \rho_{k+1})$. From this, it follows that:

Proposition 3.5.22. *Let τ be a k-dimensional simplex in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ for some $k \in \{0, \ldots, d\}$. Let $y \in \mathbb{Z}^d$ be a vertex and ω be an ordered partition of $\{1, \ldots, d+1\}$ into $k+1$ parts such that $\sigma(y, \omega)$ is the permutahedral representation of τ. Let $l \in \{0, \ldots, d\}$ be such that $k \leq l$. There exists a bijection from the set $\operatorname{cof}(\tau, l)$ of l-dimensional cofaces of τ to the set of the vectors $(a, \rho_1, \ldots, \rho_d)$, each of which consists of:*

- *an integer composition a of $l+1$ in $k+1$ parts $a_1, \ldots, a_{k+1}$ bounded by $|\omega(1)|, \ldots, |\omega(k+1)|$ respectively, and*
- *ordered partitions $\rho_1, \ldots, \rho_{k+1}$ of $\omega(1), \ldots, \omega(k+1)$ in $a_1, \ldots, a_{k+1}$ parts respectively.*

Time and space complexity. We are now ready to show that the time and space complexity of the coface generation algorithm satisfy Theorem 3.5.7.

Proof (of Theorem 3.5.7). The space complexity of the coface generation algorithm is the storage size of the record $(a, \rho_1, \ldots, \rho_{k+1})$ that represents the current coface. Its size was discussed in the proof of Lemma 3.5.18 and is equal to $O(d)$.

The time complexity of 1 is given by Lemma 3.5.18. The time complexity of 2 is given by Lemma 3.5.20. The time complexity of 3 is given by Lemmas 3.5.19 and 3.5.21. With all complexities found, we conclude the proof. □

From Theorem 3.5.7 we can find the total complexity of computing the permutahedral representation of all cofaces of a given dimension.

Theorem 3.5.23 (Total complexity of computing the permutahedral representations of cofaces)**.** *Let τ be a k-dimensional simplex in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$, for some $k \in \{0, \ldots, d\}$. Let $l \in \{0, \ldots, d\}$ be such that $k \leq l$. Let $N(\tau, l)$ be the number of l-dimensional cofaces of τ in the Freudenthal-Kuhn's triangulation of $\mathbb{R}^d$. The time complexity of computing the permutahedral representations of all l-dimensional cofaces of τ is $O(dN(\tau, l))$.*

Remark 3.5.24. *The space complexity to store the permutahedral representation of a simplex in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ is $\Theta(d)$. Therefore the bound on the time complexities in Theorem 3.5.23 and Propositions 3.5.25 and 3.5.26 below are optimal.*

The number $N(\tau, l)$ is given by Corollary 1.3.20. In particular:

- If $l = d$, then we have $N(\tau, d) = O((d-k+1)!)$.

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 3.1: The table of average running times in milliseconds of the face generation algorithm to compute all k-dimensional faces of the simplices of various dimensions l in the Freudenthal-Kuhn triangulation of $\mathbb{R}^{30}$. The tests were performed on the permutahedral coordinates of all faces of 500 random full-dimensional simplices that are obtained by applying the point location algorithm on random points on a sphere centered in the origin.

- If $l = k + 1$, then we have $N(\tau, k + 1) = O(2^{d-k})$.

The following (optimal) complexities follow from the observation above and Theorem 3.5.23.

Proposition 3.5.25 (Total complexity of computing the permutahedral representations of maximal cofaces)**.** *Let τ be a k-dimensional simplex in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$, for some $k \in \{0, \dots, d\}$. The time complexity of computing the permutahedral representations of all d-dimensional cofaces of τ is $O(d(d - k + 1)!)$.*

Proposition 3.5.26 (Total complexity of computing the permutahedral representations of cofacets)**.** *Let τ be a k-dimensional simplex in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$, for some $k \in \{0, \dots, d - 1\}$. The time complexity of computing the permutahedral representations of all cofacets of τ is $O(d2^{d-k})$.*

3.5.3 Experimental analysis

In this section, we present the execution time of the face and coface generation algorithms implemented in C++.

In Table 3.1, we show how the execution time of the face generation algorithm depends on the dimension of the input simplex and the dimension of the computed faces in a fixed ambient dimension 30. Each entry in Table 3.1 corresponds to the average total time in milliseconds of computing all k-dimensional faces of 1000 simplices of some dimension l in $\mathbb{R}^{30}$. Note that the time 11904.7ms is the time of computing all 5852925 22-dimensional faces of a simplex of dimension 30.

We present in Table 3.2 the running time in milliseconds of the face computation algorithm per computed face. As we can see, except for the case $l = k$, the running time per computed face is around 2µs. This is coherent with Theorem 3.5.13, in which we established that this running time is $O(d)$, which is constant in our case.

In Tables 3.3 and 3.4, we present analogous tables for the coface computation algorithm. Similarly, the running time per computed coface in Table 3.4 is around 2µs with the exception of when k is close to l.

In Tables 3.5 and 3.6, we present the execution time of the coface generation algorithm applied on simplices of low codimension in various ambient Euclidean spaces of high dimensions

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 3.2: The table of average running times in milliseconds per computed face of the face generation algorithm to compute the k-dimensional faces of the simplices of various dimensions l in the Freudenthal-Kuhn triangulation of $\mathbb{R}^{30}$.

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

Table 3.3: The table of average running times in milliseconds of the coface generation algorithm to compute all cofaces of the simplices of various dimensions in the Freudenthal-Kuhn triangulation of $\mathbb{R}^{30}$. The tests were performed on the permutahedral coordinates of all faces of 1000 random full-dimensional simplices that are obtained by applying the point location algorithm on random points on a sphere centered in the origin.

to generate the permutahedral coordinates of the cofacets and of the full-dimensional cofaces respectively. Note that the execution times in Table 3.5 are lower than in Table 3.6. This supports our choice of the adjacency graph for the manifold computation algorithm in Chapter 4 (see Remark 4.2.3).

In the same way, we present in Tables 3.7 and 3.8 the execution time of the face generation algorithm in various ambient Euclidean spaces of high dimensions. In Table 3.7, we compute the facets of $(d-m+1)$-dimensional simplices (that we encounter in the manifold tracing algorithm later in Chapter 4), and in Table 3.8, we compute the $(d-m)$-dimensional faces of the full-dimensional simplices. Once again, the execution times in Table 3.7 are lower than in Table 3.8, in an even more striking way than for the coface generation algorithm. This gives us another reason for the choice of the adjacency graph for the manifold computation algorithm later in Chapter 4 (see Remark 4.2.3).

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

Table 3.4: The table of average running times in milliseconds per computed coface of the coface generation algorithm to compute the cofaces of the simplices of various dimensions in the Freudenthal-Kuhn triangulation of $\mathbb{R}^{30}$.

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

Table 3.5: The table of average running times in milliseconds of the coface generation algorithm to compute all cofacets of the $(d-m)$-dimensional simplices in the Freudenthal-Kuhn triangulation of various ambient dimensions. Here m is the codimension of a simplex, which in terms of the manifold tracing algorithm (Chapter 4) corresponds to the dimension of the reconstructed manifold.

Ambient dimension d		50	100	150	200	250	300	350	400
Simplex codimension m	1	0.068	0.134	0.234	0.375	0.471	0.617	0.683	0.82
	2	0.074	0.15	0.236	0.411	0.517	0.673	0.754	0.998
	3	0.096	0.186	0.309	0.499	0.635	0.842	0.936	1.122
	4	0.156	0.307	0.505	0.786	1.006	1.206	1.381	1.681
	5	0.379	0.776	1.267	1.946	2.485	3.062	3.491	4.164
	6	1.433	2.966	4.76	7.236	9.507	11.411	13.449	15.525
	7	6.709	14.331	25.076	34.297	44.405	52.796	63.926	73.752

Table 3.6: The table of average running times in milliseconds of the coface generation algorithm to compute all d-dimensional cofaces of the $(d-m)$-dimensional simplices in the Freudenthal-Kuhn triangulation of various ambient dimensions. Here m is the codimension of a simplex, which in terms of the manifold tracing algorithm (Chapter 4) corresponds to the dimension of the reconstructed manifold.

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 3.7: The table of average running times in milliseconds of the face generation algorithm to compute all $(d-m)$-dimensional facets of the $(d-m+1)$-dimensional simplices in the Freudenthal-Kuhn triangulation of various ambient dimensions. Here m is the codimension of a simplex, which in terms of the manifold tracing algorithm (Chapter 4) corresponds to the dimension of the reconstructed manifold.

Ambient dimension d		50	100	150	200	250	300	350	400
m	1	0.3	0.7	1.6	2.6	4.1	6.2	8.8	12.9
	2	6.2	35.4	112.9	275.5	518.7	957.2	1700.2	2667.9

Table 3.8: The table of average running times in milliseconds of the face generation algorithm to compute all $(d-m)$-dimensional facets of the $(d-m+1)$-dimensional simplices in the Freudenthal-Kuhn triangulation of various ambient dimensions. Here m is the codimension of a simplex, which in terms of the manifold tracing algorithm (Chapter 4) corresponds to the dimension of the reconstructed manifold.

Chapter 4

Manifold tracing algorithm

In this chapter, we present an algorithm, called the *manifold tracing algorithm*, that constructs a piecewise-linear approximation of a given submanifold $\mathcal{M}$ in Euclidean space $\mathbb{R}^d$, of arbitrary dimension and codimension. We assume that the input manifold is compact and smooth. For simplicity, we also assume that the manifold is boundaryless throughout the first part of this chapter. The case of a manifold with boundary will be briefly discussed in Section 4.3.5. The manifold tracing algorithm can be applied in such contexts as piecewise-linear approximation of the invariant manifold for dynamical systems (see for example [SP93, HP16]), the study of energy landscapes in physical chemistry [MTCW10], etc.

The manifold tracing algorithm operates based on an ambient triangulation $\mathcal{T}$. The triangulation $\mathcal{T}$ is assumed to be the result of a bijective affine transformation applied on the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$. In this chapter, we fix m to be the dimension of the manifold $\mathcal{M}$ and:

$$k = d - m.$$

The construction of the piecewise-linear approximation of the manifold $\mathcal{M}$ is done in two stages:

1. In the first stage, the algorithm constructs a set $\mathcal{S}$ of k-dimensional simplices of $\mathcal{T}$ that intersect the manifold $\mathcal{M}$.
2. In the second stage, the algorithm constructs a cell complex $\mathcal{H}$ from the set $\mathcal{S}$ that approximates the manifold $\mathcal{M}$.

Under a specific condition (transversality hypothesis below), there is always a way to construct the cell complex $\mathcal{H}$ from the set of k-dimensional simplices $\mathcal{S}$ using the face structure of the ambient triangulation $\mathcal{T}$. We can therefore see the set $\mathcal{S}$ as a compact way of storing the cell complex $\mathcal{H}$. Thus, the set $\mathcal{S}$ will serve us as the output of the manifold tracing algorithm in the following.

For the output set $\mathcal{S}$ to be well-defined, we assume that the manifold $\mathcal{M}$ and the triangulation $\mathcal{T}$ satisfy the following genericity hypothesis:

Hypothesis 4.0.1 (Genericity)**.** *The manifold has an empty intersection with all simplices of dimensions strictly lower than k in the triangulation $\mathcal{T}$.*

For the cell complex $\mathcal{H}$ to be well-defined, in addition to Hypothesis 4.0.1, we also need the following hypothesis:

Hypothesis 4.0.2 (Transversality)**.** *The intersection of the manifold $\mathcal{M}$ and any k-dimensional simplex in the triangulation $\mathcal{T}$ is a single point.*

Contribution

This chapter has two objectives. Let $\mathcal{T}$ be a triangulation obtained from the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ by a bijective affine transformation.

Data structure. Our first contribution in this chapter is a data structure that allows us to answer the following queries on the triangulation $\mathcal{T}$ in a time- and space-efficient manner:

Point location: Compute the simplex in the triangulation $\mathcal{T}$ that contains a given point.

Face generation: Generate the faces of a specific dimension of a given simplex in the triangulation $\mathcal{T}$.

Coface generation: Generate the cofaces of a specific dimension of a given simplex in the triangulation $\mathcal{T}$.

Cartesian coordinates: Compute the Cartesian coordinates of a vertex from its internal representation (see below).

All simplices in $\mathcal{T}$ are represented using the permutahedral representation (introduced in Chapter 3) of the corresponding simplices in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ before applying the affine transformation. The algorithm for the point location and the face and coface queries are implemented using the time- and space-efficient algorithms from Sections 3.4, 3.5.1 and 3.5.2. Notably, to answer the face and coface queries, the data structure generates the permutahedral coordinates of faces and cofaces one by one. The complexity to generate one face or coface is the storage complexity $O(d)$ of a permutahedral representation. Moreover, at any point in time the storage needed for the generation of faces and cofaces does not exceed $O(d)$.

Important examples of the triangulations $\mathcal{T}$ supported by this data structure are Coxeter triangulations of type $\tilde{A}_d$ (see Section 1.2.4). Coxeter triangulations of type $\tilde{A}_d$ have *good simplex quality* (see Section 2.2). This is a crucial condition for some methods, such as the homeomorphic manifold reconstruction technique via perturbations by Boissonnat *et al.* [BKW18].

Manifold tracing algorithm. Our second contribution in this chapter is the *manifold tracing algorithm* that can serve two purposes:

1. It can be used to compute a piecewise-linear approximation of the input manifold $\mathcal{M}$ using an ambient triangulation $\mathcal{T}$.
2. It can also be used to compute an ε-sampling of the input manifold $\mathcal{M}$ for an arbitrary positive ε.

The ambient triangulation $\mathcal{T}$ is stored using the data structure above. The manifold tracing algorithm that we present here is similar in vein to the pattern algorithm by Allgower and Schmidt [AS85] and to the contour tracing by Dobkin *et al.* [DWLT90].

The input of the manifold tracing algorithm consists of:

- an affinely transformed Freudenthal-Kuhn triangulation $\mathcal{T}$ of $\mathbb{R}^d$ stored in the data structure above,
- a submanifold $\mathcal{M}$ of $\mathbb{R}^d$ of some known dimension m, such that it satisfies the genericity hypothesis (Hypothesis 4.0.1) with the triangulation $\mathcal{T}$,

- a seed point x_0 on the manifold $\mathcal{M}$.

The output of the manifold tracing algorithm is the set $\mathcal{S}$ of k-dimensional simplices in the triangulation $\mathcal{T}$ that intersect the manifold $\mathcal{M}$. As we said before, under the transversality hypothesis (Hypothesis 4.0.2), this set $\mathcal{S}$ can be naturally transformed into a cell complex $\mathcal{H}$ that serves as a piecewise-linear approximation of $\mathcal{M}$.

The manifold $\mathcal{M}$ is given via a so-called intersection oracle. The intersection oracle takes as input a simplex $\tau \in \mathcal{T}$ of dimension k and decides whether τ intersects the manifold $\mathcal{M}$ or not. If the transversality hypothesis (Hypothesis 4.0.2) holds, we further require that the intersection oracle provides an approximate intersection point of the simplex τ and the manifold $\mathcal{M}$. In this case, these intersection points serve as the vertices in the embedding of the cell complex $\mathcal{H}$.

By design, the user is free to choose the intersection oracle, and hence the representation of the manifold. This makes our algorithm *general*, as it can be applied independently of the way the input manifold is given.

We are particularly interested in two special cases of manifolds: the implicit manifolds and the manifolds given by point clouds. By implicit manifold, we mean the m-dimensional zero-set of a smooth function $F : \mathbb{R}^d \to \mathbb{R}^k$, under a condition that the zero is a regular value of F. Using a result by Cheng and Chiu [CC14], we can build a function from a point cloud, such that its zero-set is homeomorphic to the manifold under some specific conditions that we will mention in Section 4.3.4. Using this result, we can reduce the case of the manifolds given by point clouds to the implicit manifold case. For this reason, we give more attention to the implicit manifold case in this chapter.

It is worth mentioning that there exist other methods in the literature that allow to fit a manifold to a given point cloud, notably the one proposed by Fefferman *et al.* [FIK$^+$18]. However, instead of defining the manifold as a zero-set of an implicit function the authors construct a projection function on the manifold. This makes this method less directly applicable in our case than the one suggested by Cheng and Chiu [CC14].

In Section 4.3, we define an intersection oracle for the implicit manifold case based on the linear interpolation of the values of the input function F on the vertices. Using this oracle, the constructed cell complex $\mathcal{H}$ has the following guarantees:

- The cell complex $\mathcal{H}$ is a piecewise-linear manifold.
- The points on the cell complex $\mathcal{H}$ are close to the manifold $\mathcal{M}$ (by one-sided Hausdorff distance).

We provide the *space and time complexity analysis* of the manifold tracing algorithm, something that has been neglected in the related algorithms in [AS85] and [DWLT90]. In particular, we show in Proposition 4.2.6 that the manifold tracing algorithm is *output sensitive*, meaning that its time complexity is linear in the output size $|\mathcal{S}|$. In the particular case of the implicit manifold, this complexity is:

$$O(m2^m(kE + k^\omega))|\mathcal{S}|,$$

where $O(E)$ is the complexity of evaluating the function F and $O(k^\omega)$ is the time complexity of computing the product of two $k \times k$ matrices. The time complexity per element in the output is *intrinsic dimension sensitive*. This means that the time complexity depends exponentially on the intrinsic dimension m but only mildly on the ambient dimension d.

While the manifold tracing algorithm is output sensitive, the size $|\mathcal{S}|$ of the output can be large depending on the scale of the ambient triangulation $\mathcal{T}$. We show in Section 4.2.2 that if δ is the diameter of the simplices in the ambient triangulation $\mathcal{T}$, Θ is their fatness (see

Definition 2.1.1), and V_k is the volume of the k-dimensional unit ball, then the size of the output is $|\mathcal{S}| = O\left(V_k \frac{1}{\Theta}\text{vol}_m(\mathcal{M})\, m^{-1/2}(2/\delta)^m\right)$, where $V_k = O\left(\left(\frac{2\pi e}{k}\right)^{k/2}\right)$ (Proposition 1.1.36).

Additional assumptions on the input

Note that we assume that the dimension m of the manifold is given in the input. We also assume that a point x_0 on the manifold $\mathcal{M}$ is known. Finding the point x_0 on a manifold given as a zero-set of a function might be challenging. There exist some methods to estimate a seed point set of the given function, such as the contour tree extraction proposed by Carr *et al.* [CSA03, CS09]. These methods are limited to the codimension one case (also known as hypersurfaces), but can be extended easily to general codimension. More precisely, we can apply the same method k times to find the seed point of the manifold of codimension k seen as an intersection of k hypersurfaces.

The outline of the chapter

This chapter is organized as follows. In Section 4.1, we introduce the data structure that is used to store the ambient triangulation $\mathcal{T}$. In Section 4.2, we present the manifold tracing algorithm and show its complexity. There we also give a bound on the size $|\mathcal{S}|$ of the output. In Section 4.3, we describe an example of the intersection oracle for the particular case of implicit manifolds. There we discuss about the complexity of the intersection query and also the theoretical guarantees on the resulting cell complex $\mathcal{H}$. We also discuss how the implicit manifold case can be applied to reconstruct manifolds from point clouds using a result by Cheng and Chiu [CC14]. Lastly, in Section 4.4, we report on the experimental results.

4.1 The data structure to represent an ambient triangulation

Let $\mathcal{T}$ be a triangulation of $\mathbb{R}^d$ obtained from the Freudenthal-Kuhn triangulation $\mathcal{T}_0$ of $\mathbb{R}^d$ by a bijective affine transformation $x \mapsto \Lambda x + b$, where $\Lambda \in \mathbb{R}^{d\times d}$ is an invertible matrix and $b \in \mathbb{R}^d$ is the translation vector.

In this section, we introduce a data structure that represents such a triangulation $\mathcal{T}$ and that allows the user to answer the following queries on it:

- point location of a given point $p \in \mathbb{R}^d$ in the triangulation $\mathcal{T}$,
- the k-dimensional[13] faces of a given l-dimensional simplex $\tau \in \mathcal{T}$ for any $k \leq l$,
- the l-dimensional cofaces of a given k-dimensional simplex $\tau \in \mathcal{T}$ for any $k \leq l$,
- computing the Cartesian coordinates of a given vertex $v \in \mathcal{T}$.

This data structure is used later in Section 4.2 for the manifold tracing algorithm.

In the following, points in $\mathbb{R}^d$ are stored as arrays of Cartesian coordinates, as described in Section 1.1.6. Any simplex in the triangulation $\mathcal{T}$ is addressed using the permutahedral representation (see Definition 3.3.15) of the corresponding simplex in $\mathcal{T}_0$.

[13] As in Section 3.5, we follow the convention of denoting by k and l the smallest and the greatest dimension respectively. The k in this section is unrelated to the dimension $k = d - m$ in the description of the manifold tracing algorithm.

Members of the data structure. The data structure stores three members:

- the matrix Λ,
- the inverse matrix Λ^{-1},
- the translation vector b.

The matrix Λ and the vector b are used to compute the Cartesian coordinates of a vertex in the triangulation $\mathcal{T}$. The inverse matrix Λ^{-1} and the vector b are instrumental in answering the point location query.

The storage size of the data structure is the sum of the storage sizes of the two matrices Λ and Λ^{-1} and of the vector b.

Proposition 4.1.1. *The total storage size of the data structure is $O(d^2)$.*

If the triangulation $\mathcal{T}$ is the Freudenthal-Kuhn triangulation $\mathcal{T}_0$ of $\mathbb{R}^d$, then the matrix Λ is the identity matrix I_d and the vector b is 0. In this case we can avoid storing the matrices Λ and Λ^{-1} and the vector b, leaving us with the $O(1)$ storage complexity.

Matrix Λ for the case of Coxeter triangulation of type $\tilde{A}_d$. Coxeter triangulations of type $\tilde{A}_d$ have many properties that are interesting in practice.

- We showed in Chapter 2 that Coxeter triangulations of type $\tilde{A}_d$ have a good simplicial quality and are Delaunay triangulations with positive protection. These two properties are crucial for some manifold reconstruction methods, such as the manifold reconstruction algorithm by perturbations by Boissonnat *et al.* [BKW18].
- We will show later in Section 4.4 that in practice, the size of the output of the manifold tracing algorithm called on a Coxeter triangulation of type $\tilde{A}_d$ is smaller compared to the Freudenthal-Kuhn triangulation $\mathcal{T}_0$ of $\mathbb{R}^d$.

Using the construction from Section 3.1.3, we get the matrix $\Lambda = \Lambda_C$ that corresponds to a Coxeter triangulation of type $\tilde{A}_d$. We can thus represent a Coxeter triangulation of type $\tilde{A}_d$ with the help of the current data structure.

4.1.1 Queries on the data structure

We will now give an overview of the queries that are supported by the data structure. For each type of the queries, we provide the time and space complexities to answer them.

The time complexities in this section are parameterized by:

- the ambient dimension d,
- the intrinsic dimension m of the manifold $\mathcal{M}$,
- the time complexity $M(\Lambda)$ of the multiplication of a vector in $\mathbb{R}^d$ by the matrix Λ,
- the time complexity $M(\Lambda^{-1})$ of the multiplication of a vector in $\mathbb{R}^d$ by the matrix Λ^{-1}.

The time complexities $M(\Lambda)$ and $M(\Lambda^{-1})$ can largely vary depending on the matrices Λ and Λ^{-1}. In general, the time complexity of the multiplication of a $d \times d$ matrix by a vector in $\mathbb{R}^d$ is $O(d^2)$. For some special cases of the matrices Λ and Λ^{-1}, the time complexities $M(\Lambda)$ and $M(\Lambda^{-1})$ can be much lower. For example, if both Λ and Λ^{-1} are identity matrices (which is the case of $\mathcal{T} = \mathcal{T}_0$), the complexities $M(\Lambda)$ and $M(\Lambda^{-1})$ are the storage complexity $O(d)$ of the output.

Point location. The point location query takes as input a point $p \in \mathbb{R}^d$. The output consists of the permutahedral representation of the simplex $\tau \in \mathcal{T}$ of smallest dimension that contains the point p.

The algorithm to answer the point location query is based on the point location algorithm in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ that we described in Section 3.4. This algorithm has two steps:

- First, we compute the reverse image x of p under the bijective affine transformation:

$$x = \Lambda^{-1}(p - b).$$

- After this, we apply the point location algorithm on the point x in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ from Section 3.4. The permutahedral representation in the output of the point location algorithm is the output of the query.

The time and space complexities to answer the point location query are expressed as follows.

Proposition 4.1.2. *The time complexity to answer the point location query on a given point $p \in \mathbb{R}^d$ is $O(M(\Lambda^{-1}) + d\log(d))$. The additional storage used by the point location query is $O(d)$.*

Proof. The time complexity of computing $x = \Lambda^{-1}(p - b)$ is the sum of:

- the time complexity $O(d)$ of an addition of two d-dimensional vectors, and
- the time complexity $O(M(\Lambda^{-1}))$ of one multiplication of a d-dimensional vector by the matrix Λ^{-1}.

The time complexity of applying on the point x the point location algorithm in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ is $O(d\log(d))$ (Proposition 3.4.4). We conclude that the total time complexity of one call of the point location query is:

$$O(M(\Lambda^{-1}) + d\log(d)).$$

The additional storage used to answer the point location query comes from:

- the storage of the vector x, which is $O(d)$,
- the storage used for the point location algorithm in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$, which is $O(d)$ (Proposition 3.4.5).

We conclude that the additional storage used to answer the point location query is $O(d)$. □

Cartesian coordinates of a vertex. Let $v \in \mathbb{R}^d$ be a vertex in the triangulation $\mathcal{T}$. Let $y \in \mathbb{Z}^d$ be the corresponding vertex in $\mathcal{T}_0$ that appears in the permutahedral representation of v. By definition of $\mathcal{T}$, we have:

$$v = \Lambda y + b.$$

The time complexity of computing the Cartesian coordinates of the vertex v is hence the sum of the time complexity of multiplication of Λ by y and of the time complexity of the addition of two d-dimensional vectors Λy and b. Thus:

Proposition 4.1.3. *The time complexity of computing the Cartesian coordinates of a vertex in the triangulation $\mathcal{T}$ given by its permutahedral representation is $O(M(\Lambda) + d)$.*

Note that the space complexity of answering this query only consists of the storage of the output.

Barycentre of a simplex. Using the algorithm that computes the Cartesian coordinates of a vertex, we can compute the barycentre of a simplex in the triangulation $\mathcal{T}$. Let τ be a simplex of some dimension l. Computing the barycentre consists of generating the vertices of τ, computing their Cartesian coordinates and computing their mean value. The complexities of these operations are as follows:

1. The complexity of generating the vertices of τ is $O(dl)$ (see Theorem 3.5.13).
2. According to Proposition 4.1.4, computing the Cartesian coordinates of one vertex takes $O(M(\Lambda) + d)$ operations. Therefore, computing the Cartesian coordinates of the $l+1$ vertices of τ takes $O(l(M(\Lambda) + d))$ operations.
3. Computing the mean value of the Cartesian coordinates of the $l+1$ vertices takes $O(dl)$ operations.

The total complexity of computing the barycentre of τ is thus the following.

Proposition 4.1.4. *Let $l \in \{0, \dots, d\}$. The time complexity of computing the barycentre of an l-dimensional simplex in the triangulation $\mathcal{T}$ given by its permutahedral representation is $O(l(M(\Lambda) + d))$.*

Note that the vertex generation only needs $O(d)$ space complexity at any point of its execution (follows from Theorem 3.5.6). The only other storage needed to compute the barycentre of a simplex is used to store the output and the Cartesian coordinates of the currently generated vertex. Both take $O(d)$ space. Thus, the total space complexity is $O(d)$.

Remark 4.1.5. *The barycentres of some l-dimensional simplices can be computed in $O(ld)$ operations. Here is an example of such a simplex. Let τ_0 be the l-dimensional simplex given by the permutahedral coordinates $\sigma(y, \omega)$ (see Definition 3.3.1) with:*

$$y = (0, \dots, 0) \text{ and } \omega = \{1\}, \dots, \{l\}, \{l+1, \dots, d\}.$$

We will now show that its barycentre c can be computed efficiently using the columns $col_1, \dots, col_d$ of the matrix Λ.

Define $v_0, \dots, v_l$ be the vertices of τ_0 as in Definition 3.3.1, however notice that we now use the column vectors $col_1, \dots, col_d$ of Λ instead of the canonical basis vectors $e_1, \dots, e_d$:

$$\begin{aligned} v_0 &= y \\ v_i &= v_{i-1} + col_i, \quad \text{for } i \in \{1, \dots, l\}. \end{aligned}$$

The barycentre can be written in the following way:

$$\begin{aligned} c &= \frac{1}{l+1}(v_0 + v_1 + v_2 + \ldots + v_l) \\ &= \frac{1}{l+1}\left(0 + col_1 + (col_1 + col_2) + \ldots + (col_1 + \ldots + col_l)\right) \\ &= \frac{1}{l+1}\left(l \cdot col_1 + (l-1) \cdot col_2 + \ldots + 1 \cdot col_l\right). \end{aligned}$$

As we can see, we can compute c by $O(l)$ additions of d-dimensional vectors. This takes $O(ld)$ operations.

Face query. Let $k, l \in \{0, \dots, d\}$ be such that $k \leq l$. The k-dimensional face query takes as input a permutahedral representation of an l-dimensional simplex τ in $\mathcal{T}$. It then outputs the permutahedral representations of all k-dimensional faces of τ.

Because the triangulation $\mathcal{T}$ is an affine transformation of the Freudenthal-Kuhn triangulation $\mathcal{T}_0$ of $\mathbb{R}^d$, the combinatorial structure of the two triangulations $\mathcal{T}$ and $\mathcal{T}_0$ coincide. For this reason, we can use the face generation algorithm for the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ from Section 3.5.1 for the face query. The time and space complexities to answer the face query hence follow from Theorem 3.5.13.

Proposition 4.1.6. *Let $k, l \in \{0, \dots, d\}$ be such that $k \leq l$. Let τ be an l-dimensional simplex in $\mathcal{T}$. Let:*

$$F(\tau, k) = \binom{l+1}{k+1}$$

be the number of k-dimensional faces of the simplex τ. The complexity to answer the k-dimensional face query called on τ is $O\left(dF(\tau, k)\right)$. Its space complexity is $O(d)$.

Coface query. Let $k, l \in \{0, \dots, d\}$ be such that $k \leq l$. The coface query is defined in a similar way to the face query. The l-dimensional coface query takes as input a permutahedral representation of a k-dimensional simplex τ in $\mathcal{T}$. It then outputs the permutahedral representations of all l-dimensional cofaces of τ in $\mathcal{T}$. Similarly to the face query, to answer the coface query we use the coface generation algorithm from Section 3.5.2. Its time and space complexities follow from Theorem 3.5.23.

Proposition 4.1.7. *Let $k, l \in \{0, \dots, d\}$ be such that $k \leq l$. Let τ be a k-dimensional simplex in $\mathcal{T}$. Let $N(\tau, l)$ be the number of l-dimensional cofaces of τ in the triangulation $\mathcal{T}$. The time complexity to answer the l-dimensional coface query called on τ is $O\left(dN(\tau, l)\right)$. Its space complexity is $O(d)$.*

For the manifold tracing algorithm in Section 4.2, we will be interested in the particular case of cofacets ($l = k + 1$). We will call the coface query in this particular case the *cofacet query.* The time complexity of the cofacet query is given by Proposition 3.5.25.

Proposition 4.1.8. *Let $k \in \{0, \dots, d-1\}$ and τ be an k-dimensional simplex in $\mathcal{T}$. The time complexity to answer the cofacet query called on τ is $O\left(d2^{d-k}\right)$. Its space complexity is $O(d)$.*

4.2 Manifold tracing algorithm

The manifold tracing algorithm (see Algorithm 1) that we propose here is inspired by the simplicial continuation methods such as [AS85] and [DWLT90].

Input. The manifold tracing algorithm is given as input:

- A triangulation $\mathcal{T}$ of $\mathbb{R}^d$, which is identical to the Freudenthal-Kuhn triangulation $\mathcal{T}_0$ of $\mathbb{R}^d$ up to an affine transformation. The triangulation $\mathcal{T}$ is stored using the data structure from Section 4.1.
- An m-dimensional compact submanifold $\mathcal{M}$ of the Euclidean space $\mathbb{R}^d$. We assume that the manifold $\mathcal{M}$ and the triangulation $\mathcal{T}$ satisfy the genericity hypothesis (Hypothesis 4.0.1). The manifold $\mathcal{M}$ is accessed through an oracle that allows to answer whether a k-simplex

Algorithm 1: Manifold tracing algorithm

```
input  : Triangulation $\mathcal{T}$ of $\mathbb{R}^d$, manifold $\mathcal{M}$ of dimension $m$, seed point $x_0 \in \mathcal{M}$
output: Set $\mathcal{S}$ of the simplices in $\mathcal{T}$ of dimension $k = d - m$ that intersect $\mathcal{M}$
Translate $\mathcal{T}$ so that $x_0$ coincides with the barycentre of a $k$-dimensional face $\tau_0$ in $\mathcal{T}$
Initialize the queue $\mathcal{Q}$ with $\tau_0$
Initialize $\mathcal{S}$ with $\tau_0$
while the queue $\mathcal{Q}$ is not empty do
    Pop a $k$-dimensional simplex $\tau$ from $\mathcal{Q}$
    foreach cofacet $\varphi$ of $\tau$ do
        foreach facet $\rho$ of $\varphi$ do
            if $\rho$ does not lie in $\mathcal{S}$ and intersects $\mathcal{M}$ then
                Insert $\rho$ to the queue $\mathcal{Q}$
                Insert $\rho$ together with the intersection point to the output set $\mathcal{S}$
```

in the triangulation $\mathcal{T}$ intersects the manifold $\mathcal{M}$. Here, $k = d - m$ is the codimension of $\mathcal{M}$. In the following, we will refer to this oracle as the *intersection oracle*. In Section 4.3, we will describe an intersection oracle for the case of a manifold given implicitly as the zero-set of a smooth function.

- A seed point on the manifold $x_0 \in \mathcal{M}$, from which the continuation starts.

Remark 4.2.1. *Note that if $\mathcal{M}$ consists of multiple connected components, then a seed point per each connected component must be provided. We will tacitly assume in the following that there is only one connected component in $\mathcal{M}$, in other words, $\mathcal{M}$ is connected.*

Output. The output of the manifold tracing algorithm consists of the set $\mathcal{S}$ of all k-dimensional simplices in the triangulation $\mathcal{T}$ that intersect the manifold $\mathcal{M}$.

Definition 4.2.2 (Piercing simplex)**.** *A simplex in the set $\mathcal{S}$ is called* piercing.

The simplices in the set $\mathcal{S}$ are stored using the permutahedral representation (see Definition 3.3.15). We will also assume that the set $\mathcal{S}$ is stored as a hash table to guarantee the time complexity in Proposition 4.2.6.

If the manifold $\mathcal{M}$ and the triangulation $\mathcal{T}$ satisfy the transversality hypothesis (Hypothesis 4.0.2), then we can define from the set $\mathcal{S}$ a cell complex $\mathcal{H}$ (or equivalently its dual $\mathcal{H}^*$) which are piecewise-linear approximations of the input manifold $\mathcal{M}$. The cell complexes $\mathcal{H}$ and $\mathcal{H}^*$ are described in detail in Section 4.2.3. If the transversality hypothesis holds, we also store the unique intersection points alongside the simplices in $\mathcal{S}$. These intersection points serve as vertices when we embed the cell complex $\mathcal{H}$ in $\mathbb{R}^d$.

Adjacency graph. The manifold tracing algorithm relies on the traversal of a certain graph G (see Figure 4.1), which under the transversality hypothesis (Hypothesis 4.0.2) is isomorphic to the 1-skeleton of the output cell complex $\mathcal{H}$. This graph is defined in the following way:

- The nodes in G correspond to the piercing k-dimensional simplices.
- Two nodes are connected by an edge if and only if the two corresponding k-dimensional simplices share a common cofacet (which is a $(k+1)$-dimensional face).

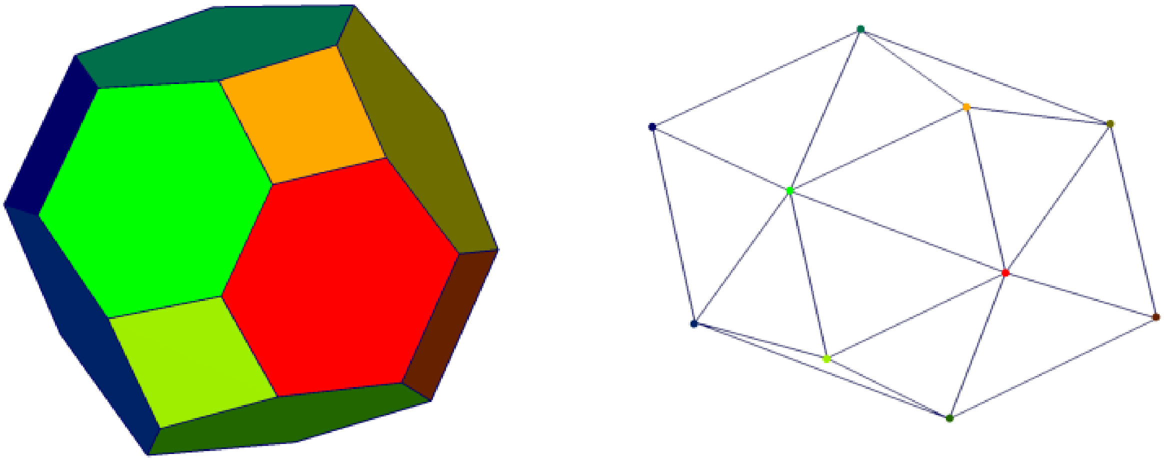


Figure 4.1: A (two-dimensional) cell complex $\mathcal{H}^*$ (defined in Section 4.2.3) and the corresponding adjacency graph. The cells in the illustration are projections of dual cells in the three-dimensional space on the two-dimensional plane.

We refer to the graph G as the *adjacency graph*. Abusively, we refer to the nodes in G by the corresponding k-dimensional simplices.

Remark 4.2.3. *Two nodes in the graph G share an edge if the two corresponding k-simplices have a common $(k+1)$-dimensional coface. One may wonder about a different definition for G. For example, it would be possible to define the graph G with the same nodes but with edges that correspond to common d-dimensional cofaces. This results in a worse time complexity to traverse the graph. There are two reasons for this.*

1. *First, recall from Proposition 3.5.25 that generating the d-dimensional cofaces has the time complexity $O(d(m+1)!)$. This time complexity is worse than the time complexity $O(d2^m)$ from Proposition 3.5.26 for the cofacet computation (compare Tables 3.5 and 3.6).*

2. *The second reason is that the number of k-dimensional faces of a d-dimensional simplex is $\binom{d+1}{k+1}$. This is significantly larger than the number $k+2$ of the facets of a $(k+1)$-simplex (compare Tables 3.7 and 3.8).*

Note that the adjacency graph G is never stored explicitly in the manifold tracing algorithm. This in particular means that the adjacent nodes of a given node n in G are not a priori accessible, but instead have to be computed. We will now specify how this is done.

Computing the adjacent nodes in G. Let τ be the piercing k-dimensional simplex in the triangulation $\mathcal{T}$ that corresponds to a node n in the adjacency graph G. Assume that n' is a node in G that is adjacent to n. The node n' corresponds to a piercing k-dimensional simplex τ' in the triangulation $\mathcal{T}$ that shares a cofacet with τ.

As such, to find all adjacent nodes of a given node n, the manifold tracing algorithm does the following:

- It first computes the set $\mathrm{cof}(\tau, k+1)$ (see Definition 3.5.4) of cofacets of the k-dimensional simplex τ using the cofacet query from Section 4.1.1. Each cofacet in the set $\mathrm{cof}(\tau, k+1)$ corresponds to an edge in G that is incident to the node n.
- Then it computes the set $\mathrm{fac}(\rho, k)$ (see Definition 3.5.3) of facets for each $(k+1)$-dimensional simplex ρ in the set $\mathrm{cof}(\tau, k+1)$ using the face query from Section 4.1.1. The nodes n' in G that are adjacent to the node n are in one-to-one correspondence with the k-dimensional simplices in:
$$\bigcup_{\rho \in \mathrm{cof}(\tau, k+1)} \mathrm{fac}(\rho, k)$$
that intersect the manifold $\mathcal{M}$.

Initialization of the traversal. As we will now see, the manifold tracing algorithm can initialize the traversal with a node that corresponds to any k-dimensional simplex in $\mathcal{T}$. For this, we will translate the ambient triangulation $\mathcal{T}$ in such a way that the seed point x_0 from the input coincides with the barycentre of the chosen simplex.

Fix an arbitrary k-dimensional simplex τ_0 in the triangulation $\mathcal{T}$ and let c be the barycentre of τ_0. The translation of the triangulation $\mathcal{T}$ consists of assigning the offset vector b in the data structure that represents $\mathcal{T}$ in the following way:
$$b := b + (x_0 - c).$$
Note that the set $\mathcal{S}$ and the adjacency graph G change after the translation of the triangulation $\mathcal{T}$. The simplex τ_0 in the translated triangulation $\mathcal{T}$ intersects the manifold $\mathcal{M}$ at the seed point x_0, which makes τ_0 a member of $\mathcal{S}$. We can therefore use τ_0 as the first k-dimensional simplex from which the propagation begins.

Note that because the choice of τ_0 is arbitrary, we can choose the simplex from Remark 4.1.5 for which there is an efficient way to compute the barycentre. The time complexity of the initialization with this choice of τ_0 is the following.

Lemma 4.2.4. *The time complexity of the initialization of the manifold tracing algorithm is $O(kd)$.*

Proof. The time complexity of finding the barycentre c of τ_0 is $O(kd)$ (see Remark 4.1.5). Translating the triangulation $\mathcal{T}$ by the vector $x_0 - c$ takes $O(d)$ operations. Therefore, the total time complexity of the initialization of the manifold tracing algorithm is $O(kd)$. □

Choice of the order of the traversal. The nodes in the adjacency graph G that are to be visited in the future by the manifold tracing algorithm are stored in a variable $\mathcal{Q}$. The traversal order of G is determined by the order of the nodes stored in the variable $\mathcal{Q}$. During the iteration of the while loop in the manifold tracing algorithm, the variable $\mathcal{Q}$ is updated in the following way.

1. The top node n in $\mathcal{Q}$ is removed from $\mathcal{Q}$.
2. All nodes that are adjacent to n in G and that have not been visited before are inserted in $\mathcal{Q}$.

Remark 4.2.5. *Note that each node in the adjacency graph (which corresponds to a k-dimensional simplex) is inserted in $\mathcal{Q}$ at most once.*

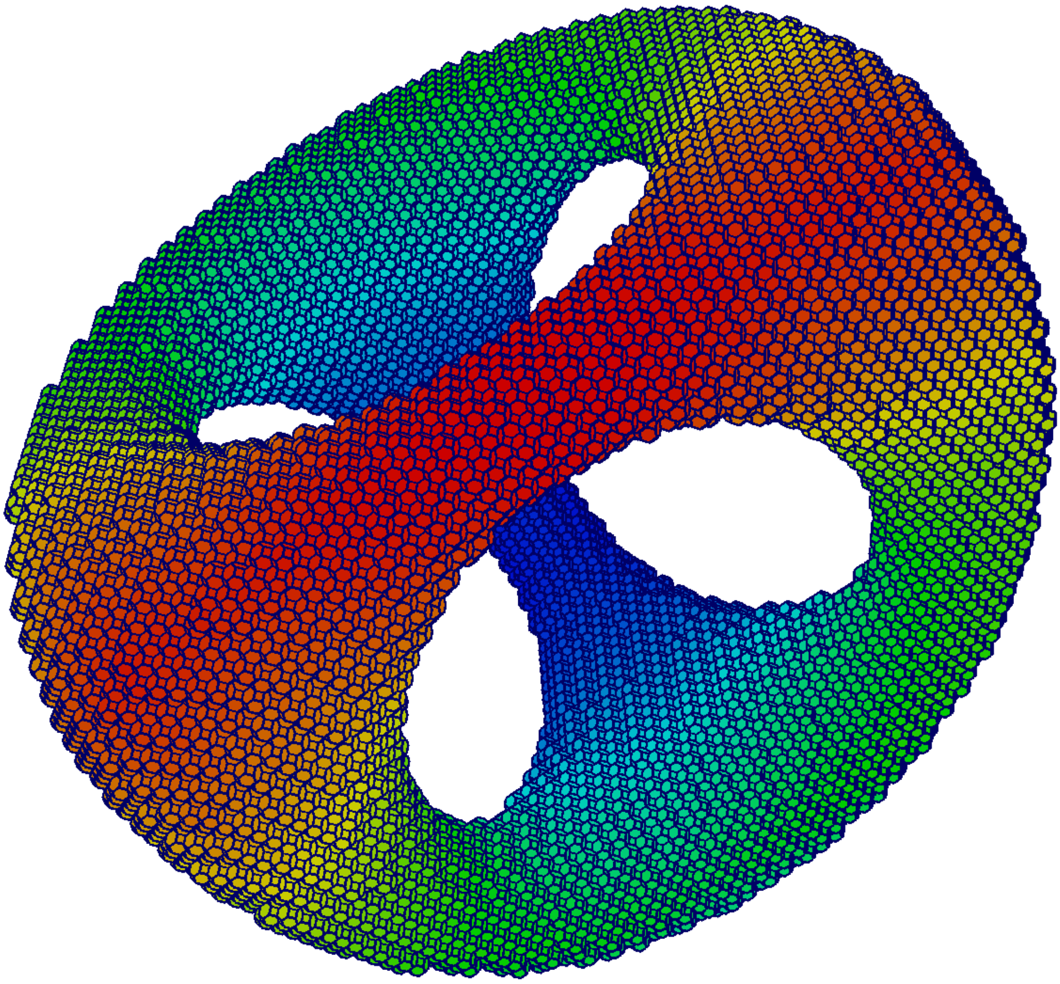


Figure 4.2: A heat map that illustrates the order of construction of the cells in the cell complex $\mathcal{H}^*$ (see Section 4.2.3) on an example of a two-dimensional surface in $\mathbb{R}^3$. The cells with a colour that is closer to blue are constructed before the triangles with a colour that is closer to red.

In practice, the best option for the choice of the data structure of the variable $\mathcal{Q}$ is the *queue* (also known as “first-in-first-out” or FIFO). Choosing $\mathcal{Q}$ to be a queue corresponds to the *breadth-first search* traversal of the graph G. With the breadth-first search traversal of G, the boundary grows most of the time in a roughly uniform way in all directions (see Figure 4.2). This way, the size of the variable $\mathcal{Q}$ remains small during most of the execution of the manifold tracing algorithm.

4.2.1 Complexity of the manifold tracing algorithm

We will now express the time complexity of the manifold tracing algorithm. This complexity depends on the time complexity $O(I)$ of the computation of one call of the intersection oracle, as well as the size $|\mathcal{S}|$ of the output.

Proposition 4.2.6. *The time complexity of the manifold tracing algorithm is:*

$$O\left(kd + m2^m I|\mathcal{S}|\right),$$

where $O(I)$ is the time complexity of one call of the intersection oracle.

Note that the manifold tracing algorithm is *output-sensitive*, as it depends linearly on the size $|\mathcal{S}|$ of the output. After the proof of Proposition 4.2.6, we will further express the size $|\mathcal{S}|$ of the output in terms of the quantities that depend on the manifold and the resolution of the triangulation.

Proof (of Proposition 4.2.6). The time complexity of the initialization of the manifold tracing algorithm is given by Lemma 4.2.4. This complexity is $O(kd)$.

Each iteration of the while loop in Algorithm 1 computes one element in the set $\mathcal{S}$. During the execution of one iteration of the while loop, the manifold tracing algorithm calls the cofacet query on a k-dimensional simplex. Recall from Corollary 1.3.18 that the number of cofacets of an m-dimensional cell is $O(2^m)$. The time complexity of computing all cofacets of a k-dimensional simplex is (see Proposition 4.1.8):

$$O\left(d2^m\right).$$

For each of the computed $O(2^m)$ cofacets, the manifold tracing algorithm computes all its $m+2$ facets in time $O(dm)$ (see Proposition 4.1.6). The total time complexity of the facet computation is hence:

$$O\left(dm2^m\right).$$

The number of m-dimensional simplices that are computed per one iteration of the while loop is $O(m2^m)$. For each computed m-dimensional simplex, the manifold tracing algorithm checks whether this simplex exists in $\mathcal{S}$ or not. Thanks to the choice of storing $\mathcal{S}$ as a hash table, the average time complexity of this check is the size of the permutahedral representation, which is $O(d)$. If this check is successful, the intersection query is called. The time complexity of one call is $O(I)$. Therefore the total time complexity of all existence checks in $\mathcal{S}$ and all calls of the intersection queries is:

$$O(m2^m(d+I)).$$

Because the complexity I takes at least $\Omega(d)$ time needed to store the Cartesian coordinates of the intersection point, we can further simplify this complexity:

$$O(m2^m I).$$

In conclusion, the total complexity per iteration of the while loop is:

$$O\left(d2^m\right) + O\left(dm2^m\right) + O\left(m2^m I\right) = O\left(m2^m I\right).$$

The number of iterations of the while loop is $|\mathcal{S}|$. The result follows. □

In the particular case of an implicit manifold and an ambient Coxeter triangulation of type $\tilde{A}_d$, the manifold tracing algorithm has the following complexity.

Corollary 4.2.7. *Let $\mathcal{M}$ be an m-dimensional manifold that is given as a zero-set of a function. Let the ambient triangulation $\mathcal{T}$ be a Coxeter triangulation of type $\tilde{A}_d$. Denote by δ the diameter of the d-dimensional simplices in $\mathcal{T}$. The complexity of the manifold tracing algorithm is the following:*

$$O\left((k + Ek^\omega) \left(\frac{2\pi e}{k}\right)^{k/2} \frac{d^{(d+1)/2}\, d!}{2^d} \operatorname{vol}_m(\mathcal{M}) \sqrt{m} (4/\delta)^m \right),$$

where $O(k^\omega)$ is the complexity of multiplication of two $k \times k$ matrices and $O(E)$ is the complexity of evaluating on a point in $\mathbb{R}^d$ the function that defines the implicit manifold.

Proof. From Proposition 4.2.6, the time complexity of the manifold tracing algorithm is:

$$O\left(kd + m2^m I |\mathcal{S}|\right).$$

We will see in Proposition 4.2.10 that for a Coxeter triangulation of type $\tilde{A}_d$, the value $|\mathcal{S}|$ is:

$$|\mathcal{S}| = O\left(\left(\frac{2\pi e}{k}\right)^{k/2} \frac{d^{(d+1)/2}\, d!}{2^d} \operatorname{vol}_m(\mathcal{M})\, m^{-1/2} (2/\delta)^m \right).$$

We will also see that the complexity of one call of the intersection oracle I in the case of an implicit manifold is: $I = O(kE + k^\omega)$. The corollary follows. □

4.2.2 Size of the output

We will now express the size of the output in terms of the quantities that depend on the manifold and the resolution of the triangulation.

Proposition 4.2.8 (Size of the output)**.** *The size of the output $\mathcal{S}$ of the manifold tracing algorithm called on the m-dimensional manifold $\mathcal{M}$ and the d-dimensional triangulation $\mathcal{T}$ is expressed as follows:*

$$|\mathcal{S}| \leq \hat{C} \left(\frac{2\pi e}{k}\right)^{k/2} \frac{1}{\Theta} \operatorname{vol}_m(\mathcal{M})\, m^{-1/2} (2/\delta)^m,$$

where:

- $\operatorname{vol}_m(\mathcal{M})$ *is the m-dimensional volume of the manifold $\mathcal{M}$,*
- δ *is the diameter of the d-dimensional simplices in the triangulation $\mathcal{T}$,*
- Θ *is the fatness (see Definition 2.1.1) of the d-dimensional simplices in the triangulation $\mathcal{T}$,*
- $\hat{C}$ *is a constant that does not depend on d, m or δ.*

Proof. Let $\mathcal{N}$ be the set of the d-dimensional cofaces of the simplices in $\mathcal{S}$, and let N be the cardinality of $\mathcal{N}$. The proof of the proposition is subdivided in two steps:

- First, we find an upper bound on N in terms of d, m and δ.
- After that, we bound the cardinality $|\mathcal{S}|$ in terms of N, d and m.

Upper bound on N. Let us first find an upper bound on N. We are interested in the tubular neighbourhood $\mathcal{M}^\delta$ of $\mathcal{M}$ of radius δ. Observe that any d-dimensional simplex in $\mathcal{N}$ lies inside $\mathcal{M}^\delta$. Denote by V^σ the volume of a d-dimensional simplex in the triangulation $\mathcal{T}$. The d-dimensional simplices in $\mathcal{N}$ have pairwise disjoint interiors, so their total volume is upper bounded by the d-dimensional volume $\mathrm{vol}_d(\mathcal{M}^\delta)$ of the tubular neighbourhood $\mathcal{M}^\delta$:

$$N\ V^\sigma \leq \mathrm{vol}_d(\mathcal{M}^\delta). \tag{4.1}$$

The so-called tube formula of Weyl (shown in [Wey39], see also [Gra82]) gives the upper bound on the volume $\mathrm{vol}_d(\mathcal{M}^\delta)$:

$$\mathrm{vol}_d(\mathcal{M}^\delta) \leq C\, V_k\, \mathrm{vol}_m(\mathcal{M})\, \delta^k,$$

where C is a constant that does not depend on d, m or δ. We develop V_k using the asymptotic expression in Proposition 1.1.36:

$$\mathrm{vol}_d(\mathcal{M}^\delta) \leq C' \left(\frac{2\pi e}{k}\right)^{k/2} \mathrm{vol}_m(\mathcal{M})\, \delta^k, \tag{4.2}$$

where C' is a constant that does not depend on d, m or δ. By combining the two inequalities (4.1) and (4.2), we get:

$$N\ V^\sigma \leq \mathrm{vol}_d(\mathcal{M}^\delta) \leq C' \left(\frac{2\pi e}{k}\right)^{k/2} \mathrm{vol}_m(\mathcal{M})\, \delta^k.$$

It follows that:

$$N \leq C' \left(\frac{2\pi e}{k}\right)^{k/2} \frac{\delta^k}{V^\sigma} \mathrm{vol}_m(\mathcal{M}).$$

By definition of fatness (Definition 2.1.1), we get:

$$N \leq C' \left(\frac{2\pi e}{k}\right)^{k/2} \frac{1}{\Theta} \mathrm{vol}_m(\mathcal{M}) (1/\delta)^m. \tag{4.3}$$

where Θ is the fatness of the simplices in the triangulation $\mathcal{T}$. Note that the dependency of N on $1/\delta$ is exponential in m and not in d.

Upper bound on $|\mathcal{S}|$. Now, we want to express $|\mathcal{S}|$ in terms of N, d and m. For this, we count the number INC of incidences of the k-dimensional simplices in $\mathcal{S}$ and the d-dimensional simplices in $\mathcal{N}$. By incidence in this context, we mean an ordered pair (face, coface), hence each pair of a k-dimensional simplex in $\mathcal{S}$ and an incident d-dimensional simplex in $\mathcal{N}$ is counted once. We count the number INC in two ways:

$$\sum_{\tau \in \mathcal{S}} |\mathrm{cof}(\tau, d)| = \mathrm{INC} = \sum_{\sigma \in \mathcal{N}} |\mathrm{fac}(\sigma, k) \cap \mathcal{S}|. \tag{4.4}$$

According to Corollary 1.3.20, there exists a constant C'' that does not depend on d, m or δ such that for any k-dimensional simplex $\tau \in \mathcal{S}$, we have:

$$|\mathrm{cof}(\tau, d)| \geq C'' \left(\frac{d}{m+1}\right)^{m+1}.$$

By applying this lower bound on each simplex $\tau \in \mathcal{S}$, we get:

$$\sum_{\tau \in \mathcal{S}} |\mathrm{cof}(\tau, d)| \geq C'' \left(\frac{d}{m+1}\right)^{m+1} |\mathcal{S}|. \tag{4.5}$$

On the other hand, for each d-dimensional simplex $\sigma \in \mathcal{N}$, we have the following upper bound:

$$|\mathrm{fac}(\sigma, k) \cap \mathcal{S}| \leq |\mathrm{fac}(\sigma, k)| = \binom{d+1}{m+1}.$$

By applying this upper bound on each simplex $\sigma \in \mathcal{N}$, we get:

$$\sum_{\sigma \in \mathcal{N}} |\mathrm{fac}(\sigma, d) \cap \mathcal{S}| \leq \binom{d+1}{m+1} N. \tag{4.6}$$

By combining (4.4) with the inequalities (4.5) and (4.6), we get:

$$C'' \left(\frac{d}{m+1}\right)^{m+1} |\mathcal{S}| \leq \sum_{\tau \in \mathcal{S}} |\mathrm{cof}(\tau, d)| = \sum_{\sigma \in \mathcal{N}} |\mathrm{fac}(\sigma, d) \cap \mathcal{S}| \leq \binom{d+1}{m+1} N.$$

It follows that:

$$|\mathcal{S}| \leq C''^{-1} \frac{\binom{d+1}{m+1}}{\left(\frac{d}{m+1}\right)^{m+1}} N. \tag{4.7}$$

The ratio $\binom{d+1}{m+1} / \left(\frac{d}{m+1}\right)^{m+1}$ has an asymptotic bound $O(2^m m^{-1/2})$ (see Remark 4.2.9 after the proof). With this, we can rewrite (4.7):

$$|\mathcal{S}| \leq C''' 2^m m^{-1/2}\ N, \tag{4.8}$$

where C''' is some constant that does not depend on d, m or δ. By combining (4.3) and (4.8), we finally get:

$$|\mathcal{S}| \leq C''' 2^m m^{-1/2} \cdot C' \left(\frac{2\pi e}{k}\right)^{k/2} \frac{1}{\Theta} \mathrm{vol}_m(\mathcal{M})(1/\delta)^m = \hat{C} \left(\frac{2\pi e}{k}\right)^{k/2} \frac{1}{\Theta} \mathrm{vol}_m(\mathcal{M})\, m^{-1/2} (2/\delta)^m,$$

where $\hat{C} = C' \cdot C'''$ is a constant that is independent of m, d and δ.

□

Remark 4.2.9. *The ratio* $\binom{d+1}{m+1} / \left(\frac{d}{m+1}\right)^{m+1}$ *from (4.7) has a simple asymptotic upper bound. Write:*

$$\binom{d+1}{m+1} = \frac{(d+1) \cdot d \cdot \ldots \cdot (d-m+1)}{(m+1)!} \leq \frac{(d+1)^{m+1}}{(m+1)!}.$$

This yields:

$$\binom{d+1}{m+1} / \left(\frac{d}{m+1}\right)^{m+1} \leq \frac{(d+1)^{m+1}}{(m+1)!} / \left(\frac{d}{m+1}\right)^{m+1} = O\left(\frac{(m+1)^{m+1}}{(m+1)!}\right). \tag{4.9}$$

By applying Stirling formula $(m+1)! \sim \left(\frac{m+1}{e}\right)^{m+1} \sqrt{2\pi(m+1)}$ *on the right hand side of (4.9), we finally get:*

$$\binom{d+1}{m+1} \Big/ \left(\frac{d}{m+1}\right)^{m+1} = O\left(\frac{(m+1)^{m+1}}{\left(\frac{m+1}{e}\right)^{m+1} \sqrt{2\pi(m+1)}}\right) = O(2^m m^{-1/2}).$$

Comparison of Coxeter triangulations and the Freudenthal-Kuhn triangulation. The fatness term Θ in the expression in Proposition 4.2.8 depends on the choice of the triangulation $\mathcal{T}$. Here, we will discuss two possible choices of $\mathcal{T}$ namely: a Coxeter triangulation of type $\tilde{A}_d$ and the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$.

The fatness Θ_{CT} of d-dimensional simplices in a Coxeter triangulation of type $\tilde{A}_d$ was computed in Section 2.4.1:

$$\Theta_{CT} = \begin{cases} \frac{2^d}{\left(\sqrt{d+1}\right)^{d+1} d!} & \text{if } d \text{ is odd,} \\ \frac{2^d(\sqrt{d+1})^{d-1}}{\left(\sqrt{d(d+2)}\right)^{d} d!} & \text{if } d \text{ is even.} \end{cases}$$

From this, we get:

$$\frac{1}{\Theta_{CT}} = O\left(\frac{d^{(d+1)/2}\, d!}{2^d}\right). \tag{4.10}$$

We also know the fatness Θ_{FKT} of d-dimensional simplices in the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ from Proposition 3.1.18:

$$\Theta_{FKT} = \frac{1}{d^{d/2}\, d!}.$$

From this, we get:

$$\frac{1}{\Theta_{FKT}} = O\left(d^{d/2}\, d!\right). \tag{4.11}$$

Substituting the expressions of $\frac{1}{\Theta_{CT}}$ and $\frac{1}{\Theta_{FKT}}$ in Proposition 4.2.8 yields:

Proposition 4.2.10. *We use the same notations as in Proposition 4.2.8. In the case of $\mathcal{T}$ being a Coxeter triangulation of type $\tilde{A}_d$, we have:*

$$|\mathcal{S}| = O\left(\left(\frac{2\pi e}{k}\right)^{k/2} \frac{d^{(d+1)/2}\, d!}{2^d} \operatorname{vol}_m(\mathcal{M})\, m^{-1/2}(2/\delta)^m\right).$$

In the case of $\mathcal{T}$ being the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$, we have:

$$|\mathcal{S}| = O\left(\left(\frac{2\pi e}{k}\right)^{k/2} d^{d/2}\, d! \operatorname{vol}_m(\mathcal{M})\, m^{-1/2}(2/\delta)^m\right).$$

By comparing these two expressions, it follows that by using a Coxeter triangulation of type $\tilde{A}_d$ as the ambient triangulation we expect a smaller output set $\mathcal{S}$ than by using the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ by a factor $2^d/\sqrt{d}$. This can also be seen in the experimental results in Section 4.4 (see Table 4.1).

4.2.3 Two cell complexes $\mathcal{H}$ and $\mathcal{H}^*$ associated to the set $\mathcal{S}$

Assume that the transversality hypothesis (Hypothesis 4.0.2) holds. In this case, we can define a cell complex $\mathcal{H}^*$ from the set $\mathcal{S}$ in the output of the manifold tracing algorithm. This cell complex is illustrated in Figure 4.3.

We define the cell complex $\mathcal{H}^*$ in $\mathbb{R}^d$ using the following observation. The dual $\mathcal{T}^*$ of the triangulation $\mathcal{T}$ can be embedded in $\mathbb{R}^d$ as an affine transformation of the Voronoi diagram of a Coxeter triangulation of type $\tilde{A}_d$ (see Section 1.3). We define the cell complex $\mathcal{H}^*$ to be the collection of the m-dimensional dual cells in $\mathcal{T}^*$ that correspond to the k-dimensional simplices in $\mathcal{S}$ and of all faces of these dual cells. Note that the cell complex $\mathcal{H}^*$ is embedded as a pure m-dimensional subcomplex of $\mathcal{T}^*$.

Remark 4.2.11. *The affine transformations preserve barycentres. In the case of Coxeter triangulation of type $\tilde{A}_d$, the circumcentres of simplices coincide with the barycentres. It follows that with the embedding as above, the vertices of the cell complex $\mathcal{H}^*$ lie at the barycentres of the corresponding d-dimensional simplices in $\mathcal{T}$.*

If $\mathcal{H}^*$ is a piecewise-linear m-dimensional manifold, then we define another cell complex $\mathcal{H}$ (illustrated in Figure 4.3) as the dual of $\mathcal{H}^*$ in the sense of Poincaré duality. This means that a cell in $\mathcal{H}$ of some dimension l is in one-to-one correspondence to an $(m-l)$-dimensional cell in $\mathcal{H}^*$.

Note that with this definition, the k-dimensional simplices in the set $\mathcal{S}$ correspond to vertices in the cell complex $\mathcal{H}$. In general, for any $l \in \{k, \dots, d\}$, an l-dimensional coface of any k-dimensional simplex in the set $\mathcal{S}$ corresponds to a cell in $\mathcal{H}$ of dimension $l-k$. In particular, the d-dimensional cofaces correspond to m-dimensional (i.e. maximal) cells in $\mathcal{H}$.

If the cell complex $\mathcal{H}$ is well-defined, it is always possible to embed $\mathcal{H}$ using the barycentres of the k-dimensional simplices in the set $\mathcal{S}$ as the vertices of $\mathcal{H}$. With this definition, all vertices in each cell of $\mathcal{H}$ of any dimension $j \in \{0, \dots, m\}$ lie on a j-dimensional affine plane. Therefore, the j-dimensional cells of $\mathcal{H}$ can be realized as the j-dimensional convex hulls of their vertices.

Remark 4.2.12. *Note that the stored intersection points of the simplices in $\mathcal{S}$ with the manifold are* not *used in the constructions of the cell complexes $\mathcal{H}$ and $\mathcal{H}^*$ above. In the case of implicit manifolds, we can define $\mathcal{H}$ directly from the set $\mathcal{S}$ (see Section 4.3.2), and this time the stored intersection points serve as vertices in $\mathcal{H}$.*

4.3 Special case of the implicit manifold intersection oracle

The manifold tracing algorithm in Section 4.2 depends on the intersection oracle that was left unspecified. In this section, we will describe how the intersection oracle can be efficiently implemented for the particular case of the implicitly given manifold $\mathcal{M}$ of some dimension m. By implicit, we mean that $\mathcal{M}$ is given as the zero-set $F^{-1}(0)$ of a smooth function $F : \mathbb{R}^d \to \mathbb{R}^k$. As before, k is the *codimension* of $\mathcal{M}$, meaning that $d = m + k$.

We further assume that 0 is a regular value of the function F. With this assumption, the implicit function theorem (see any standard textbook, e.g. [MW97], for a reference) implies that the zero-set $F^{-1}(0)$ is indeed a manifold. Thanks to the celebrated Sard's theorem [Sar42], almost all points in the image of F are regular, therefore the assumption that zero is a regular value is not restrictive in practice.

Motivation. It has been reported by Liang and Zhao [LZ13] that reconstructions of local neighbourhoods on the manifold as the zero sets of locally-defined functions are preferred when solving partial differential equations. It makes solving partial differential equations one of the major applications of the manifold tracing algorithm for implicit manifold reconstruction.

As we will see in Section 4.3.4, solving the implicit manifold case also allows us to reconstruct manifolds from the point clouds.

4.3.1 Approximated intersection query

Detecting the exact intersection point p is not easy in general. Instead, as it is customary in other simplicial continuation methods, we approximate the function F by a piecewise-linear continuous function $\tilde{F}$ (see [Hen07] or [VGdF07]).

The function $\tilde{F}$ takes the values of F on vertices and its restriction to any simplex σ is defined to be linear. The value of the restriction of $\tilde{F}$ to σ is defined using the barycentric coordinates in σ. Let us now define $\tilde{F}$ formally. Fix a k-dimensional simplex σ in the ambient triangulation $\mathcal{T}$. Let $v_0, \ldots, v_k$ be the vertices of σ. Express a point $p = \sum_{i=0}^{k} \lambda_i v_i$ in the simplex σ using barycentric coordinates $\lambda_0, \ldots, \lambda_k$. Then, the value of $\tilde{F}$ at p is defined as:

$$\tilde{F}(p) = \tilde{F}\left(\sum_{i=0}^{k} \lambda_i v_i\right) = \sum_{i=0}^{k} \lambda_i F(v_i). \tag{4.12}$$

Remark 4.3.1. *The function $\tilde{F}$ is linear inside each d-dimensional simplex in $\mathcal{T}$. Therefore, if the zero-set $\tilde{F}^{-1}(0)$ intersects a k-dimensional face, then it intersects it in one point. We get that the zero-set $\tilde{F}^{-1}(0)$ and the triangulation $\mathcal{T}$ satisfy the transversality hypothesis (Hypothesis 4.0.2).*

Using the piecewise-linear approximation function $\tilde{F}$ defined as in (4.12), we introduce the *approximated intersection oracle*, which decides whether its zero-set $\tilde{F}^{-1}(0)$ intersects a k-dimensional simplex τ given by its vertices $v_0, \ldots, v_k$. For this, the oracle expresses the intersection point using the affine coordinates $\lambda_0, \ldots, \lambda_k$, treated as unknown variables such that $\sum_{i=0}^{k} \lambda_i = 1$. Note that if $p = \sum_{i=0}^{k} \lambda_i v_i$ lies in the zero-set $\tilde{F}^{-1}(0)$, then it satisfies, by definition of $\tilde{F}$:

$$\tilde{F}(p) = \sum_{i=0}^{k} \lambda_i F(v_i) = 0 \in \mathbb{R}^k. \tag{4.13}$$

To find the values of the variables $\lambda_0, \ldots, \lambda_k$, the oracle solves the linear equation:

$$\begin{pmatrix} 1 & 1 & \ldots & 1 \\ F(v_0) & F(v_1) & \ldots & F(v_k) \end{pmatrix} \begin{pmatrix} \lambda_0 \\ \lambda_1 \\ \vdots \\ \lambda_k \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}. \tag{4.14}$$

All entries $F(v_i)$ in the matrix on the left in the linear equation (4.14) are seen as column vectors with k entries. Therefore, the matrix on the left has dimensions $(k+1) \times (k+1)$. The column vector on the right has k zero entries, and hence has $k+1$ entries in total. The reader may recognize the k zero entries in this vector as the k-dimensional zero vector from (4.13).

If the values $\lambda_0, \ldots, \lambda_k$ are non-negative (meaning that they are in fact *barycentric coordinates* and p lies in the simplex τ), then the k-dimensional simplex defined by the vertices $v_0, \ldots, v_k$ intersects the zero-set $\tilde{F}^{-1}(0)$ at the point p. In this case, the approximate intersection oracle returns the value *true*. Otherwise, it returns the value $false$.

From the description of the approximate intersection oracle, the complexity follows:

Lemma 4.3.2. *The complexity of one call of the approximate intersection oracle is $O(kE + k^\omega)$, where $O(E)$ is the complexity of evaluating the function F on a point in $\mathbb{R}^d$, and $O(k^\omega)$ is the complexity of the computation of the product of two $k \times k$ matrices.*

Proof. The complexity consists of:

1. evaluating the function F on each of the $k+1$ vertices of the simplex τ,

2. solving the $(k+1) \times (k+1)$ linear system.

The first complexity is $O(kE)$. The second complexity is known to be the same as the complexity $O(k^\omega)$ for the matrix multiplication [Str69]. By summing these two complexities, we get the result. □

4.3.2 Direct construction of the cell complex $\mathcal{H}$ from the set $\mathcal{S}$

Lemma 4.3.3 allows us to construct the cell complex $\mathcal{H}$ directly from the piecewise-linear approximation $\tilde{F}$.

Lemma 4.3.3. *Let $\tilde{F}$ be the piecewise-linear approximation of the input function F defined as in (4.12). We assume that the zero-set $\tilde{F}^{-1}(0)$ and the triangulation $\mathcal{T}$ satisfy the genericity hypothesis (Hypothesis 4.0.1). Then, the collection of all non-empty intersections $\tilde{F}^{-1}(0) \cap \tau$ with $\tau \in \mathcal{T}$ is a cell complex. Moreover, this cell complex is isomorphic to the cell complex $\mathcal{H}$ (defined in Section 4.2.3).*

Proof. Let τ be a simplex of arbitrary dimension at least k in the triangulation $\mathcal{T}$, such that τ has a non-empty intersection with the zero-set $\tilde{F}^{-1}(0)$. From Remark 4.3.1, it follows that the restriction of $\tilde{F}$ on τ is linear. It follows that the intersection $\tilde{F}^{-1}(0) \cap \tau$ is a polytope.

Note that for each face φ of τ, such that φ has a non-empty intersection with $\tilde{F}^{-1}(0)$, we have that $\tilde{F}^{-1}(0) \cap \varphi$ is a face of $\tilde{F}^{-1}(0) \cap \tau$. Thus, the collection of the intersections $\tilde{F}^{-1}(0) \cap \tau$ for all simplices $\tau \in \mathcal{T}$ such that this intersection is not empty is a cell complex. Denote this cell complex as $\mathcal{H}_?$. We will now show that the cell complex $\mathcal{H}_?$ is isomorphic to the cell complex $\mathcal{H}$ defined in Section 4.2.3.

First note that the cells in $\mathcal{H}_?$ and $\mathcal{H}$ are in one-to-one correspondence. Indeed, the cells in both cell complexes $\mathcal{H}_?$ and $\mathcal{H}$ correspond to the simplices in $\mathcal{S}$ and their cofaces in the triangulation $\mathcal{T}$. Moreover, the face structures of both $\mathcal{H}_?$ and $\mathcal{H}$ are inherited from the face structure in the triangulation $\mathcal{T}$ in the same way. Thus, the two cell complexes $\mathcal{H}_?$ and $\mathcal{H}$ are indeed isomorphic.

□

4.3.3 Guarantees on the cell complexes $\mathcal{H}$ and $\mathcal{H}^*$

We will now show that the cell complexes $\mathcal{H}$ and $\mathcal{H}^*$ associated to the output set $\mathcal{S}$ (see Section 4.2.3) approximate the manifold $\mathcal{M}$. This approximation comes with the following guarantees: the small distance from the manifold and the topological consistency. We will now show these two guarantees one by one.

Small distance from the manifold. The first guarantee is that for any positive ε, the triangulation $\mathcal{T}$ can be scaled in such a way that the points on the cell complexes $\mathcal{H}$ and $\mathcal{H}^*$ are ε-close to the manifold $\mathcal{M}$. This result is shown by Allgower and Georg [AG89, Proposition 2.8] based on the following assumptions:

- Denote by $F' : \mathbb{R}^d \to \mathbb{R}^{k \times d}$ the Jacobian of the function F. There exists a constant $\kappa \in \mathbb{R}$ such that:
$$\|F'(x)^+\|_\infty \leq \kappa, \quad \text{for all } x \in \mathbb{R}^d, \tag{4.15}$$
where A^+ denotes the Moore-Penrose inverse: $A^+ = A^t(AA^t)^{-1}$.

- Write $F = (F_1, \dots, F_k)$, where for each $i \in \{1, \dots, k\}$, F_i is a real-valued function. Denote by $F_i'' : \mathbb{R}^d \to \mathbb{R}^{d \times d}$ the Hessian of the function F_i. There exists a constant $\alpha \in \mathbb{R}$, such that:
$$\|F_i''(x)\|_\infty \leq \alpha, \quad \text{for all } i \in \{1, \dots, k\} \text{ and } x \in \mathbb{R}^d. \tag{4.16}$$

Remark 4.3.4. *In Plantinga and Vegter [PV04], the authors used similar assumptions in their marching-cube-like reconstruction of a surface in $\mathbb{R}^3$. With these assumptions, they showed that their construction is homeomorphic to the surface.*

Proposition 4.3.5 shows that the points on the constructed cell complexes $\mathcal{H}$ and $\mathcal{H}^*$ are at bounded distance to the manifold $\mathcal{M}$.

Proposition 4.3.5 ([AG89, Proposition 2.8])**.** *Let $F : \mathbb{R}^d \to \mathbb{R}^k$ be a smooth function that satisfies both conditions (4.15) and (4.16). Let κ and α be as defined in (4.15) and (4.16) respectively. Assume that 0 is a regular value of F. Let $\mathcal{M}$ be the m-dimensional manifold defined as the zero-set $F^{-1}(0)$, where $m = d - k$. Let $\mathcal{T}$ be the triangulation of $\mathbb{R}^d$ obtained from the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ by a bijective affine transformation. Denote by δ the maximum diameter of simplices in the triangulation $\mathcal{T}$. Let $\mathcal{H}$ and $\mathcal{H}^*$ be the cell complexes defined from $\mathcal{M}$ and $\mathcal{T}$ as in Section 4.2.3.*

1. *For each point y on the cell complex $\mathcal{H}$, there exists a point x on the manifold $\mathcal{M}$, such that:*
$$d(x, y) \leq \kappa\alpha\delta^2.$$
2. *For each point y on the cell complex $\mathcal{H}^*$, there exists a point x on the manifold $\mathcal{M}$, such that:*
$$d(x, y) \leq \kappa\alpha\delta^2 + \delta.$$

Proof. The result for the cell complex $\mathcal{H}$ is shown in [AG89, Proposition 2.8]. The result for the cell complex $\mathcal{H}^*$ follows from the fact that the Hausdorff distance between the cell complexes $\mathcal{H}$ and $\mathcal{H}^*$ is at most the maximal diameter δ of a simplex in $\mathcal{T}$. □

Remark 4.3.6. *Note that this is not equivalent to a bounded Hausdorff distance from $\mathcal{H}$ and $\mathcal{H}^*$ to $\mathcal{M}$.*

The following lemma allows us to express the upper bound δ on the diameter of a simplex in the triangulation $\mathcal{T}$.

Lemma 4.3.7. *Let $\mathcal{T}$ be the triangulation obtained from the Freudenthal-Kuhn triangulation $\mathcal{T}_0$ of $\mathbb{R}^d$ by a bijective affine transformation $x \mapsto \Lambda x + b$. Let Λ_C be defined as a matrix such that the corresponding linear transformation maps $\mathcal{T}_0$ to a Coxeter triangulation $\mathcal{T}_C$ of type $\tilde{A}_d$ (defined in Section 3.1.3). Let L_C denote the following upper bound on the edge length of a simplex in the Coxeter triangulation $\mathcal{T}_C$ of type $\tilde{A}_d$ associated to the matrix Λ_C (see Section 2.4.1):*
$$L_C = \sqrt{d+1}/2.$$
Write $a_{max} = \|\Lambda\Lambda_C^{-1}\|_2$. The diameter δ of any simplex in $\mathcal{T}$ is upper bounded by:
$$a_{max} L_C.$$

Proof. The triangulation $\mathcal{T}$ is obtained from the Coxeter triangulation $\mathcal{T}_C$ of type $\tilde{A}_d$ by a bijective affine transformation:
$$\lambda : x \mapsto \Lambda\Lambda_C^{-1} x + b.$$
Let $\tau \in \mathcal{T}$ be a d-dimensional simplex and let $x, y \in \tau$ be two arbitrary points in τ. The simplex $\tau_C = \lambda^{-1}(\tau)$ is a simplex in $\mathcal{T}_C$, and $x_C = \lambda^{-1}(x)$ and $y_C = \lambda^{-1}(y)$ are two points in τ_C. We have:
$$\|x - y\| \leq \|\Lambda\Lambda_C^{-1}(x_C - y_C)\| \leq a_{max}\|x_C - y_C\| \leq a_{max} L_c.$$
The result follows. □

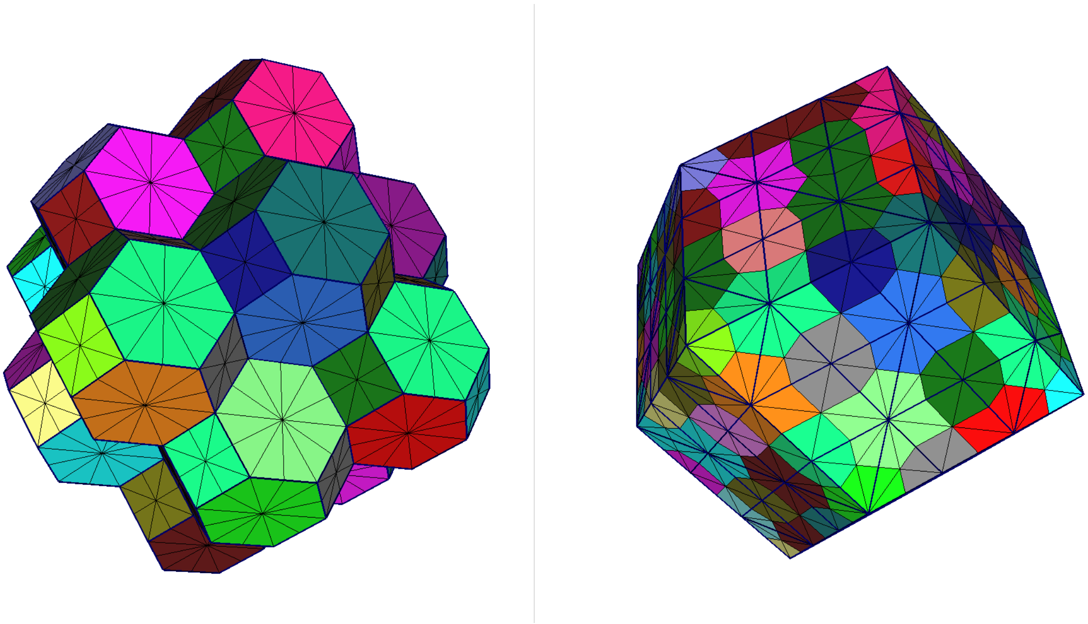


Figure 4.3: Two dual complexes: $\mathcal{H}^*$ on the left and $\mathcal{H}$ on the right. The barycentric subdivisions of cells in $\mathcal{H}^*$ correspond to stars of vertices in the barycentric subdivision of $\mathcal{H}$.

Topological consistency. The second guarantee is that both cell complexes $\mathcal{H}$ and $\mathcal{H}^*$ are piecewise-linear m-dimensional manifolds. Note however that we do not prove here that the cell complexes $\mathcal{H}$ and $\mathcal{H}^*$ have the same topology as the manifold $\mathcal{M}$.

Proposition 4.3.8 (Topological consistency)**.** *Let $\tilde{F}$ be the piecewise-linear approximation of the input function F defined by the approximate intersection oracle. We assume that the zero-set $\tilde{F}^{-1}(0)$ and the triangulation $\mathcal{T}$ satisfy the genericity hypothesis (Hypothesis 4.0.1). Then, both cell complexes $\mathcal{H}$ and $\mathcal{H}^*$ are piecewise-linear m-dimensional manifolds.*

The result for the cell complex $\mathcal{H}$ is shown by Allgower and Georg [AG90, Theorem 15.4.1]. It can also be derived from the following observation. The star of a vertex v in the cell complex $\mathcal{H}$ corresponds to a (closed) m-dimensional cell in the dual cell complex $\mathcal{H}^*$. This cell is an m-dimensional polytope, therefore homeomorphic to an m-dimensional ball. By duality, this implies that the star of a vertex v in $\mathcal{H}$ is homeomorphic to an m-dimensional ball. Therefore, $\mathcal{H}$ is indeed a piecewise-linear manifold.

To show that the cell complex $\mathcal{H}^*$ is also a piecewise-linear m-dimensional manifold, we will prove that the cell complexes $\mathcal{H}^*$ and $\mathcal{H}$ are homeomorphic.

Proposition 4.3.9. *The two cell complexes $\mathcal{H}$ and $\mathcal{H}^*$ are combinatorially equivalent one to another (defined as in [BG05, Definition 44]).*

Proof. We apply the barycentric subdivision on the two cell complexes $\mathcal{H}$ and $\mathcal{H}^*$. As illustrated in Figure 4.3, the simplices in the two barycentric subdivisions are in fact same. It follows that $\mathcal{H}$ and $\mathcal{H}^*$ are combinatorially equivalent. □

4.3.4 Manifold reconstruction from a point cloud

In this section, we will briefly mention a result by Cheng and Chiu [CC14] that allows us to apply the manifold tracing algorithm to reconstruct a manifold from a point cloud.

Result by Cheng and Chiu. Let $\mathcal{M}$ be a compact, smooth and boundaryless submanifold of $\mathbb{R}^d$ of some dimension m. For simplicity, assume that $\mathcal{M}$ has a unit reach (introduced by Federer [Fed59]). Let P be a uniform (ε, κ)-sample of $\mathcal{M}$, which means the following:

- Every point on the manifold $\mathcal{M}$ is at distance at most ε from a point in P.
- The number of sample points inside any d-dimensional ball of radius ε is upper bounded by κ.

The paper by Cheng and Chiu [CC14] provides a construction of a function φ from the point sample P that offers several guarantees. There exists a constant $\varepsilon_0 \in (0,1)$ that decreases as d increases, such that the intersection $\varphi^{-1}(0) \cap \mathcal{M}^{\varepsilon_0}$ of the zero-set $\varphi^{-1}(0)$ of the function φ and of the tubular neighbourhood $\mathcal{M}^{\varepsilon_0}$ near $\mathcal{M}$ of radius ε_0 has the following properties:

1. The set $\varphi^{-1}(0) \cap \mathcal{M}^{\varepsilon_0}$ is homeomorphic to the manifold $\mathcal{M}$.
2. The Hausdorff distance from the set $\varphi^{-1}(0) \cap \mathcal{M}^{\varepsilon_0}$ to the manifold $\mathcal{M}$ is $O(m^{5/2}\varepsilon)$.

The value $\varphi(x)$ at a point $x \in \mathbb{R}^d$ is constructed in the following way. Let $\mathcal{N}_x \subseteq P$ be the set of points in P that lie at distance at most γ from x, where $\gamma = 4\varepsilon$ is a constant called *neighbourhood radius* known in the input (see below). The authors define a weight function $w : \mathbb{R}^d \to \mathbb{R}$ as follows:

$$w_x(p) = \frac{h(\|p-x\|)}{\sum_{q \in P} h(\|p-x\|)},$$

where $h : \mathbb{R} \to \mathbb{R}$ is a function such that:

- h is differentiable in $(0, +\infty)$,
- $h(s)$ and $h'(s)$ are zero for $s \geq m\gamma$.

Note that with this definition, the value of w_x on any point $p \in P$ farther than $m\gamma$ is zero.

It is assumed that approximate tangent spaces of $\mathcal{M}$ at the sample points in P are specified in the input (see below). Therefore, we can assume that a $d \times m$ matrix T_p is given for each sample point $p \in P$ such that the columns of T_p form an orthonormal basis of the approximate tangent space of $\mathcal{M}$ at p. Define the following matrix:

$$C_x = \sum_{p \in P} w_x(p) \, \left(T_p T_p^t\right).$$

The normalized eigenvectors that correspond to the k least dominant eigenvalues of $T_p T_p^t$ form an orthonormal basis of the approximated normal space of $\mathcal{M}$ at p. The purpose of the matrix C_x is that its normalized eigenvectors that correspond to the k least dominant eigenvalues form an orthonormal basis of the "weighted average" of the approximate normal spaces of $\mathcal{M}$ at the sample points near x. Let B_x be the $d \times k$ matrix, the columns of which are these k normalized eigenvectors of C_x. The value of φ at the point x is defined using this matrix B_x:

$$\varphi(x) = B_x^t \left(x - \sum_{p \in P} w_x(p) \, p \right).$$

We will now summarize the knowledge required to be provided at the input for the construction of the function φ.

(a). As for the manifold tracing algorithm, it is assumed that the dimension m of the manifold $\mathcal{M}$ is known. In the case of manifolds given by point clouds, there are many algorithms that estimate the manifold dimension m, such as [TDSL00, HA05, LB05, CWW08, CC09].

(b). The *neighbourhood radius* $\gamma = 4\varepsilon$ is required. As Cheng and Chiu mentioned in the introduction of [CC14], the neighbourhood radius γ can be set by measuring the maximum distance from a sample point to its nth nearest neighbour for some appropriate n.

(c). It is assumed that we are given approximate tangent spaces at points in P such that the true tangent space in $\mathcal{M}$ makes an angle at most $m\gamma$ with the given approximate tangent space at that point. As the authors reported, there exist many methods for estimating tangent spaces (e.g. [SSM98, GW03, CDR05, BSW09, LMR17, BLW18]), which give an $O(\varepsilon)$ angular error.

Manifold tracing algorithm for manifold reconstruction from a point cloud. We can use the construction of the function φ by Cheng and Chiu [CC14] as a preprocessing step to later apply the manifold tracing algorithm endowed with the implicit manifold intersection oracle as in Section 4.3.1 to reconstruct the set $\varphi^{-1}(0) \cap \mathcal{M}^{\varepsilon_0}$.

For this, we need to ensure that the scale of the triangulation $\mathcal{T}$ is sufficiently small. To be precise, we impose the diameter of the simplices in the triangulation $\mathcal{T}$ to be smaller than ε_0. This way, any simplex (of any dimension) in the triangulation $\mathcal{T}$ that intersects the set $\varphi^{-1}(0) \cap \mathcal{M}^{\varepsilon_0}$ lies entirely in the tubular neighbourhood $\mathcal{M}^{\varepsilon_0}$. Therefore, we can apply the manifold tracing algorithm on the restriction of the function φ to the $\mathcal{M}^{\varepsilon_0}$. The zero-set of this function gives us the desired result.

4.3.5 The case of manifolds with boundary

In this section, we will discuss the manifold tracing algorithm in the case of smooth implicit manifolds with boundary. Let m be the dimension of the input manifold. Here, we assume that its boundary is a boundaryless $(m-1)$-dimensional manifold.

An implicit manifold with boundary can be represented as an intersection $\mathcal{M} \cap D_-$ of:

- a (possibly non-compact) manifold $\mathcal{M}$ without boundary given as the zero-set of a function $F : \mathbb{R}^d \to \mathbb{R}^k$, and
- a d-dimensional shape $D_- = D^{-1}(-\infty, 0]$, called *domain*, which is the preimage of the interval $(-\infty, 0]$ of non-positive reals by a continuous function $D : \mathbb{R}^d \to \mathbb{R}$.

Here, we impose that the domain function D is piecewise-linear with the linear pieces defined inside each d-dimensional simplex of the ambient triangulation $\mathcal{T}$. Otherwise, we can always make any function piecewise-linear of this form by applying the definition in equation (4.12).

The main differences of the implicit manifold with boundary in contrast with the boundaryless case are the following (red in Algorithm 2):

1. We impose that the given seed point is in the relative interior $\mathcal{M} \cap D_-$, in contrast with any point $x \in \mathcal{M}$ in the boundaryless case.
2. There is one additional set $\mathcal{B}$ in the output consisting of the $(k+1)$-dimensional simplices in the ambient triangulation $\mathcal{T}$ that intersect the boundary $\mathcal{M} \cap D^{-1}(0)$ of the manifold.

Algorithm 2: Manifold tracing algorithm for implicit manifolds with boundary.

```
input : Triangulation $\mathcal{T}$ of $\mathbb{R}^d$, manifold $\mathcal{M} \cap D_-$ of dimension $m$ with boundary, seed
        point $x_0 \in \mathcal{M} \cap D_-$
output: Set $\mathcal{S}$ of the simplices in $\mathcal{T}$ of dimension $k = d - m$ that intersect $\mathcal{M} \cap D_-$,
        set $\mathcal{B}$ of $(k+1)$-dimensional simplices in $\mathcal{T}$ that intersect $\mathcal{M} \cap D^{-1}(0)$
Translate $\mathcal{T}$ so that $x_0$ coincides with the barycentre of a $k$-dimensional face $\tau_0$ in $\mathcal{T}$
Initialize the queue $\mathcal{Q}$ with $\tau_0$
Initialize $\mathcal{S}$ with $\tau_0$
while the queue $\mathcal{Q}$ is not empty do
    Pop a $k$-dimensional simplex $\tau$ from $\mathcal{Q}$
    foreach cofacet $\varphi$ of $\tau$ do
        foreach facet $\rho$ of $\varphi$ do
            if $\rho$ does not lie in $\mathcal{S}$ and intersects $\mathcal{M}$ then
                $x :=$ intersection point of $\rho$ and $\mathcal{M}$
                if $x$ does not lie in $D_-$ then
                    $y :=$ intersection point of $\varphi$ and $\mathcal{M} \cap D^{-1}(0)$
                    Insert $\varphi$ together with the intersection point $y$ to the output set $\mathcal{B}$
                else
                    Insert $\rho$ to the queue $\mathcal{Q}$
                    Insert $\rho$ together with the intersection point $x$ to the output set $\mathcal{S}$
```

3. There is one additional check in the body of the while loop. If the intersection point lies outside of the (piecewise-linear) domain D, then the algorithm inserts the coface φ in the set $\mathcal{B}$.

Remark 4.3.10. *Note that the inserted $(k+1)$-dimensional coface φ does indeed intersect the boundary $\mathcal{M} \cap D^{-1}(0)$, as the specification of the set $\mathcal{B}$ requires. To see this, recall that the intersection point y of the k-dimensional simplex τ (which we popped from the queue $\mathcal{Q}$) with $\mathcal{M}$ by definition lies inside the domain D_-. On the other hand, the intersection point x of the k-dimensional simplex ρ with $\mathcal{M}$ lies outside D_-. By the intermediate value theorem, the segment $[xy]$, which belongs to the $(k+1)$-dimensional coface φ, intersects the zero-set $D^{-1}(0)$.*

Construction of the cell complex. Similarly to Section 4.3.3, from the sets $\mathcal{S}$ and $\mathcal{B}$, we can define a cell complex $\mathcal{H}$ that approximates the manifold $\mathcal{M} \cap D_-$ with boundary. We will now describe the construction of the Hasse diagram of the cell complex $\mathcal{H}$.

The vertices of $\mathcal{H}$ consist of the points in $\mathcal{S}$ and $\mathcal{B}$. As was the case in Section 4.3.3, to each cell in $\mathcal{H}$ there is a simplex in the ambient triangulation $\mathcal{T}$ in correspondence. We will denote the cells in $\mathcal{H}$ by the corresponding simplices.

Remark 4.3.11. *As we will see, in contrast to the boundaryless construction in Section 4.3.3, the cell-simplex correspondence is not one-to-one this time. A cell on the boundary of $\mathcal{H}$ can have the same corresponding simplex as a cell in the interior of $\mathcal{H}$ (illustrated in Figure 4.4).*

The construction procedure consists of two phases (blue and red in Figure 4.4).

1. In the first phase, the algorithm constructs the part of the Hasse diagram of $\mathcal{H}$ that correspond to simplices in $\mathcal{S}$ and their cofaces (blue part in Figure 4.4).

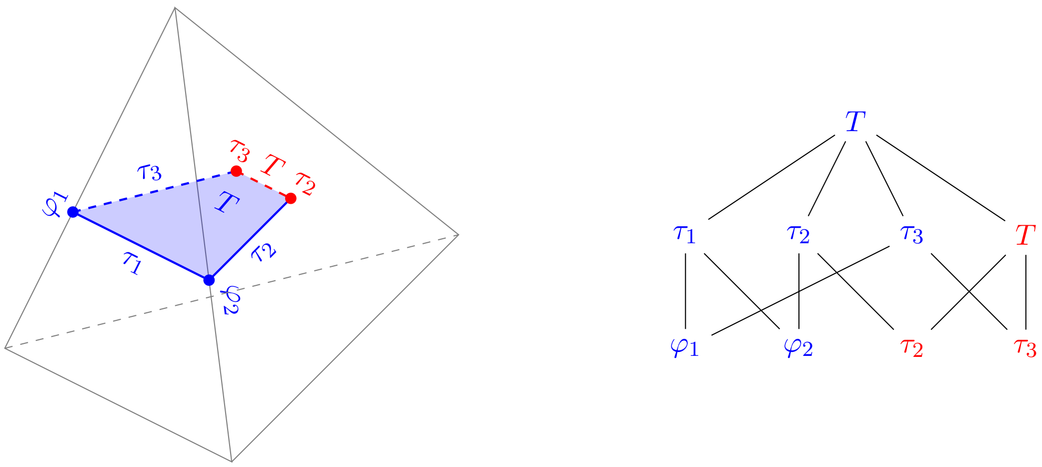


Figure 4.4: The boundary reconstruction inside an ambient tetrahedron T. The boundary cells are in red and the interior cells are in blue. The cells in the reconstruction are labelled by the face of the tetrahedron T they lie on. The vertices of the reconstruction lie on two edges φ_1 and φ_2 and on two triangular facets τ_2 and τ_3 of the tetrahedron T. The edges of the reconstruction lie on three triangular facets τ_1, τ_2 and τ_3 and inside the tetrahedron T. The two-dimensional cell lies inside the tetrahedron T. The Hasse diagram of the reconstruction is shown on the right.

2. In the second phase, the algorithm completes the Hasse diagram with cells that correspond to simplices in $\mathcal{B}$ and their cofaces (red part in Figure 4.4). If there are two cells in $\mathcal{H}$ — one in the interior and one on the boundary — that correspond to the same simplex $\tau \in \mathcal{T}$, the boundary cell is assigned to be a face of the cell in the interior.

4.4 Experimental results

The data structure from Section 4.1 and the manifold tracing algorithm have been implemented in C++ and are under review to be included as part of Gudhi library [GUD]. In this section, we will report on the experimental results using this implementation. We will focus here on the piecewise-linear approximation of compact smooth boundaryless implicit submanifold in Euclidean space $\mathbb{R}^d$. We will investigate the effects on the performance of the following parameters:

- the choice of the ambient triangulation,
- the scale of the ambient triangulation,
- the ambient dimension d,
- the intrinsic dimension m of the manifold.

In Figures 4.6, 4.8, 4.9 and 4.10, we display examples of reconstructed surfaces.

Choice of the triangulation. In Table 4.1, we compare the performance of the manifold tracing algorithm using two ambient triangulations: the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ and a Coxeter triangulation of type $\tilde{A}_d$. The input manifold used in the tests is the “Chair” surface given as the zero-set of the function (visualized in Figure 4.6):

$$f(x, y, z) = (x^2 + y^2 + z^2 - 0.8)^2 - 0.4\left((z-1)^2 - 2x^2\right)\left((z+1)^2 - 2y^2\right).$$

Ambient dimension		3	4	5	6	7	8	9	10
CT	time, s	0.138	0.374	1.127	3.122	6.973	11.545	22.554	39.514
	size $\lvert\mathcal{S}\rvert$	3990	8422	17086	29014	45162	64460	93798	124908
	av. time, ms	0.035	0.044	0.066	0.11	0.15	0.18	0.24	0.32
FKT	time, s	0.247	0.891	3.304	11.064	16.317	38.072	78.827	122.794
	size $\lvert\mathcal{S}\rvert$	8354	20042	48024	98816	108634	211136	273910	435810
	av. time, ms	0.03	0.044	0.069	0.11	0.15	0.18	0.29	0.28

Table 4.1: The comparison of the performance of the manifold tracing algorithm using two types of the ambient triangulation: a Coxeter triangulation of type $\tilde{A}_d$ (CT) and the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ (FKT) 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 equation: $(x_1^2 + x_2^2 + x_3^2 - 0.8)^2 - 0.4\left((x_3-1)^2 - 2x_1^2\right)\left((x_3+1)^2 - 2x_2^2\right) = 0$.

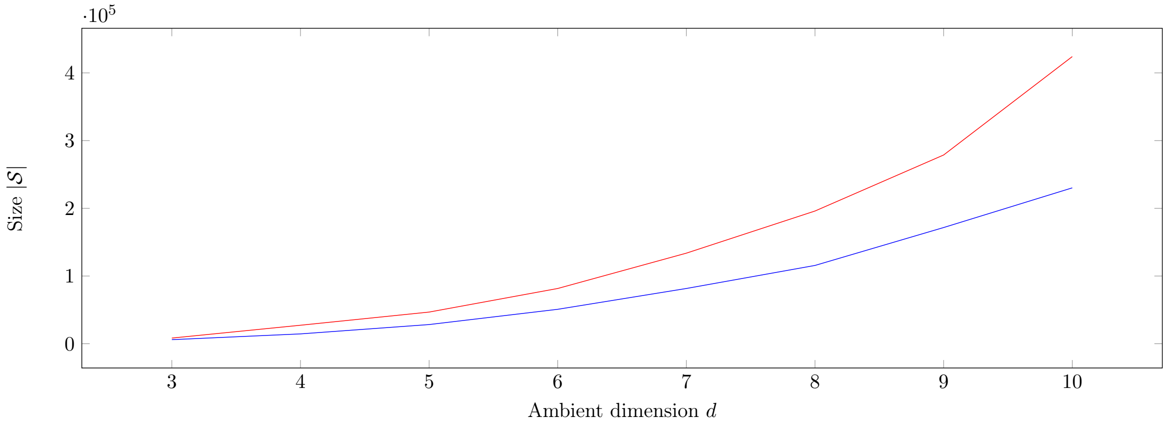


Figure 4.5: The comparison of the size of the output of the manifold tracing algorithm using two types of the ambient triangulation: a Coxeter triangulation of type $\tilde{A}_d$ (in blue) and the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ (in red) with the same diameter $0.07\sqrt{d}$ of d-dimensional simplices. The reconstructed manifold is the two-dimensional implicit surface "Chair" embedded in $\mathbb{R}^d$ given by the equation: $(x_1^2 + x_2^2 + x_3^2 - 0.8)^2 - 0.4\left((x_3-1)^2 - 2x_1^2\right)\left((x_3+1)^2 - 2x_2^2\right) = 0$.

In the tests, we embedded the implicit manifold in a higher dimensional space $\mathbb{R}^d$ for d ranging from 3 to 10.

As we can see in Table 4.1 and in Figure 4.5, the computation time and the size of the output when using a Coxeter triangulation of type $\tilde{A}_d$ are smaller than the computation time when using the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$. However, the average computation time per element in the output (shown in Table 4.1) is roughly the same for the two triangulations. From this, we conclude that the choice of the ambient triangulation does not affect the average time per element in the output, however it does affect significantly the size of the output.

Note that the smaller size of the output in the case of a Coxeter triangulation of type $\tilde{A}_d$ compared to the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ agrees with the estimate in Proposition 4.2.8 and the fact that Coxeter triangulations of type $\tilde{A}_d$ have greater fatness (Proposition 4.2.10).

Choice of the scale. We will now investigate the effect of the scale of the triangulation on the performance of the manifold tracing algorithm. In Table 4.2, we give the execution time and the sizes of the sets $\mathcal{S}$ and $\mathcal{S}_{max}$ that consist of the k-dimensional simplices and the d-dimensional simplices respectively that intersect the surface. The ambient triangulation used in the experiment is a 9-dimensional Coxeter triangulation of type $\tilde{A}_9$ and the reconstructed manifold is the "Chair" surface in Figure 4.6.

In Figure 4.7, we plot in log-log scale the dependency of the output size $|\mathcal{S}|$ in this experiment on the inverse diameter of the simplices in the ambient triangulation. The graph when put in log-log scale is linear, which is coherent with Proposition 4.2.8, which predicts the dependency $\log(|\mathcal{S}|) \sim 2\log(1/\delta)$.

The dependency on the ambient and intrinsic dimension. In Tables 4.3 and 4.4, we show the dependency of the execution time of the manifold tracing algorithm and the size of the output set $\mathcal{S}$ on the ambient dimension d and on the intrinsic dimension m of the manifold. The reconstructed manifolds are m-dimensional spheres embedded in $\mathbb{R}^d$.

Manifolds with boundary. In Figures 4.9, 4.10 and 4.11, we showcase the result of applying the manifold tracing algorithm on implicit manifolds with boundary.

4.5 Implementation details

In this section, we will describe the structure of the module `Coxeter triangulation` in Gudhi C++ library [GUD]. The overall architecture of the module can be summarized in five parts:

1. Class `Permutahedral_representation` represents a simplex in an ambient triangulation $\mathcal{T}$ as described in Section 4.1 using the permutahedral representation (Definition 3.3.15).

2. Triangulation classes `Freudenthal_triangulation` and `Coxeter_triangulation` represent the ambient triangulation $\mathcal{T}$ using data structure in Section 4.1.

3. There are two intersection oracle classes: `Implicit_manifold_intersection_oracle` and `Point_cloud_intersection_oracle` that represent the intersection oracles described in Sections 4.3.1 and 4.3.4 respectively. Additionally some function classes are provided as examples of classes that `Implicit_manifold_intersection_oracle` can take as a template parameter.

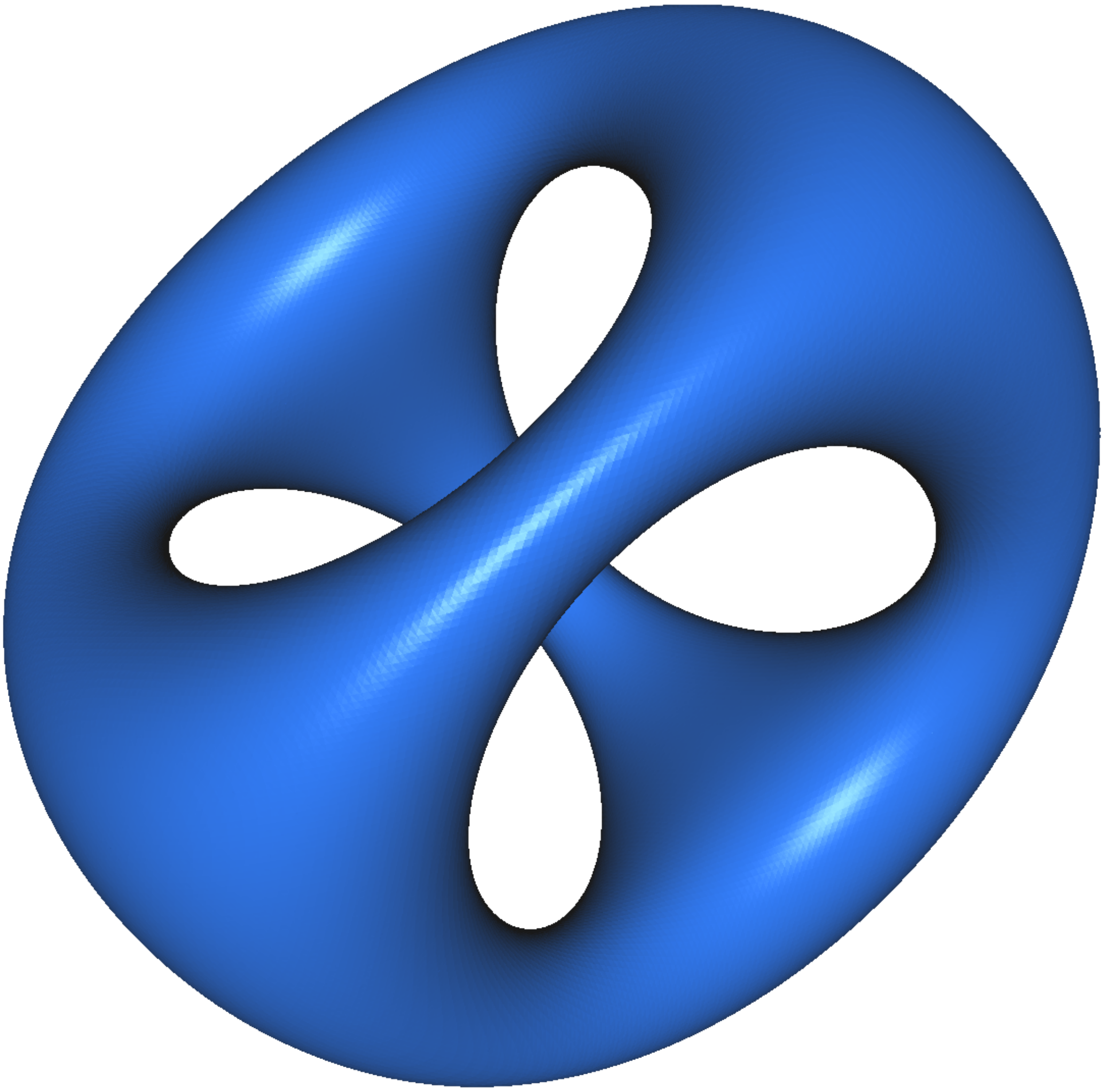


Figure 4.6: The "Chair" surface given by equation: $(x^2 + y^2 + z^2 - 0.8)^2 - 0.4\left((z-1)^2 - 2x^2\right)\left((z+1)^2 - 2y^2\right) = 0$ embedded in $\mathbb{R}^9$ and projected to $\mathbb{R}^3$ reconstructed by the manifold tracing algorithm using Coxeter triangulation of type $\tilde{A}_9$ as the ambient triangulation with the diameter 0.022 of the d-dimensional simplices.

$1/\delta =$	6.0	7.0	8.0	9.0	10.0	11.0	12.0	13.0	14.0
time, s	7.02	9.54	13.2	16.2	20.7	24.9	29.5	35.0	42.1
size $\lvert\mathcal{S}\rvert$	27268	38411	51273	65947	82480	99567	117507	140158	162004
size $\lvert\mathcal{S}_{max}\rvert$	31154	43869	58582	75310	94357	113783	134113	159956	184821

Table 4.2: The effect of the scale parameter on the computation time and the output size. Here δ corresponds to the diameter of the 9-dimensional simplices in the ambient triangulation. The reconstructed manifold in the tests is the two-dimensional implicit surface "Chair" embedded in $\mathbb{R}^9$ (see Figure 4.6). The sets $\mathcal{S}$ and $\mathcal{S}_{max}$ consist of the 7-dimensional simplices and 9-dimensional simplices respectively that intersect the surface. The ambient triangulation is a Coxeter triangulation of type $\tilde{A}_9$.

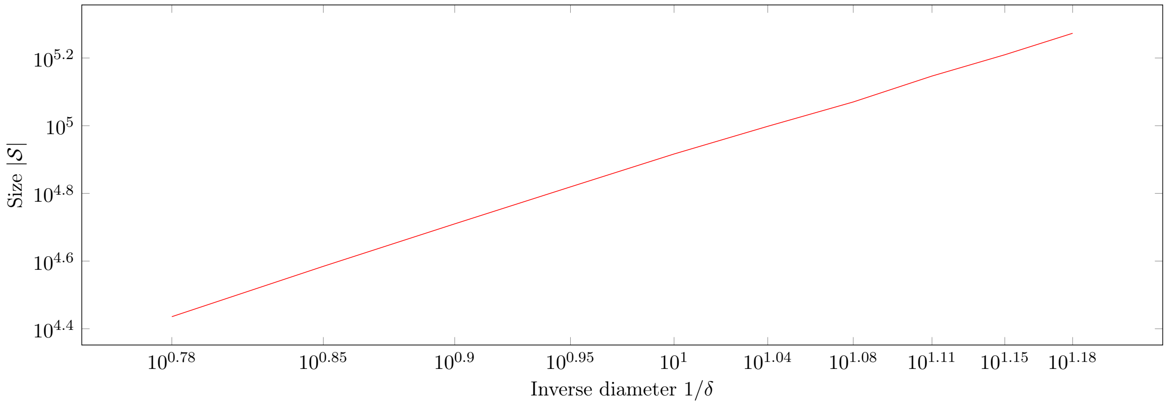


Figure 4.7: The dependency of the output size of the piecewise-linear approximation of the "Chair" surface embedded in $\mathbb{R}^9$ by the manifold tracing algorithm on the inverse of the diameter δ of the 9-dimensional simplices in the ambient triangulation. We take here a log-log scale.

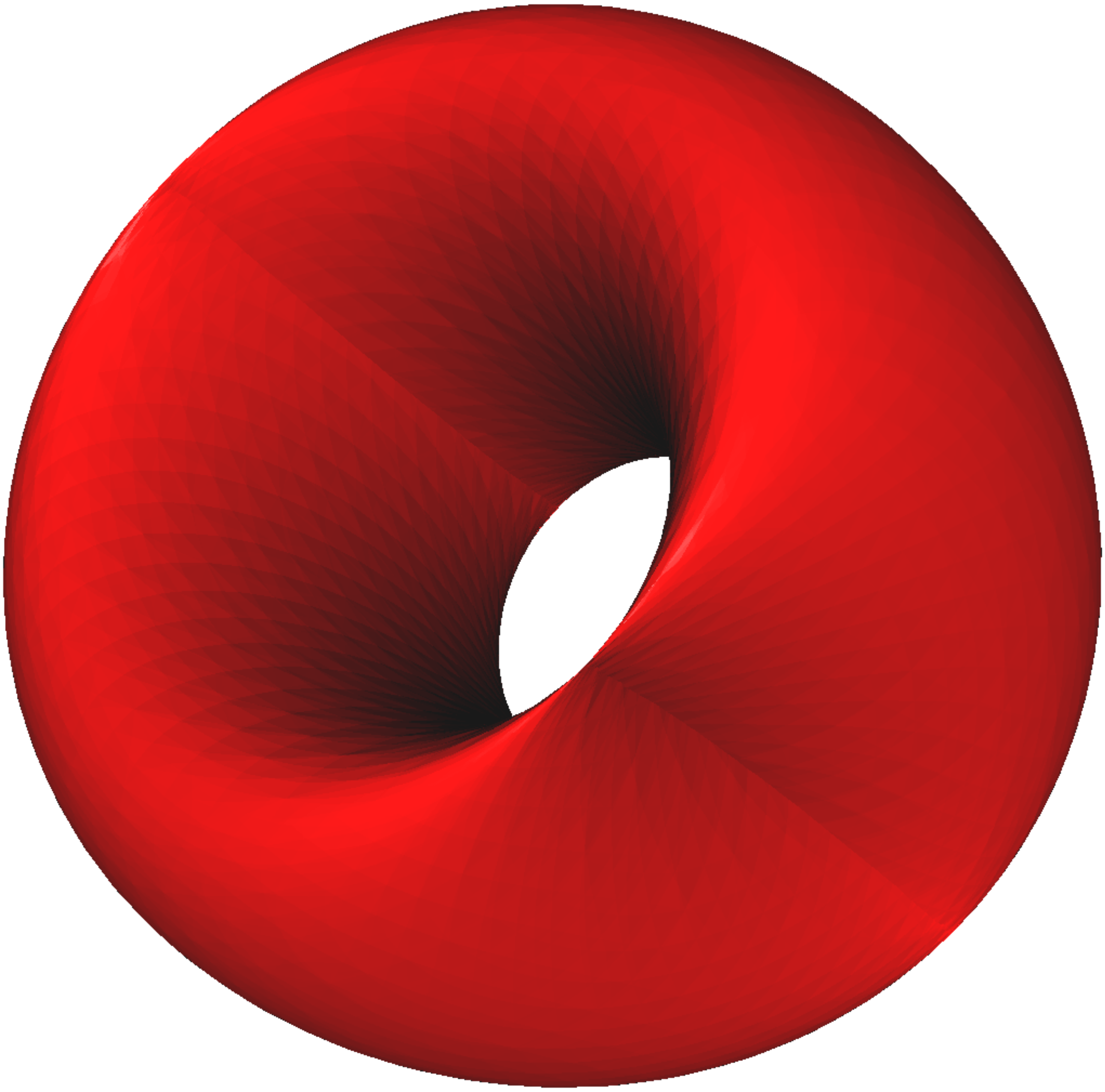


Figure 4.8: The piecewise-linear approximation of a flat torus embedded in $\mathbb{R}^{10}$ defined by equations $x_1^2 + x_2^2 = 1$ and $x_3^2 + x_4^2 = 1$. It is randomly rotated and translated in $\mathbb{R}^{10}$ and then projected to $\mathbb{R}^3$. The ambient triangulation used is a Coxeter triangulation of type $\tilde{A}_{10}$ with the diameter 0.23 of the full-dimensional simplices. The size $|\mathcal{S}|$ of the piecewise-linear approximation is 509952. The execution time of the algorithm is 231s.

d m		2	4	6	8	10	12	14
1	time, s	0.001	0.008	0.012	0.088	0.447	0.135	0.2
	$\lvert\mathcal{S}\rvert$	52	116	184	280	376	494	600
2	time, s		0.095	0.63	2.492	8.842	21.045	52.642
	$\lvert\mathcal{S}\rvert$		1708	5621	14118	27628	46438	76366
3	time, s		0.433	9.863	90.54	448.044	1932.88	6293.73
	$\lvert\mathcal{S}\rvert$		7518	73794	329751	993210	$2.47 \cdot 10^6$	$5.3 \cdot 10^6$

Table 4.3: The effect of the ambient dimension d on the execution time and the output size of the manifold tracing algorithm for various dimensions m of the submanifold. The reconstructed manifold in the tests is the m-dimensional sphere embedded in $\mathbb{R}^d$. The ambient triangulation used is a Coxeter triangulation of type $\tilde{A}_d$. The time shown in the table is the computation time of the set $\mathcal{S}$.

m d		1	2	3	4	5	6
5	time, s	0.01	0.245	2.417	6.556		
	$\lvert\mathcal{S}\rvert$	142	3324	25808	59332		
6	time, s	0.012	0.63	9.863	71.123	101.701	
	$\lvert\mathcal{S}\rvert$	184	5621	73794	320310	432922	
7	time, s	0.032	1.238	34.264	371.976	1495.65	1211.3
	$\lvert\mathcal{S}\rvert$	224	8876	157170	$1.23 \cdot 10^6$	$3.61 \cdot 10^6$	$2.84 \cdot 10^6$

Table 4.4: The effect of the intrinsic dimension m on the execution time and the output size of the manifold tracing algorithm for various ambient dimensions d. The reconstructed manifold in the tests is the m-dimensional sphere embedded in $\mathbb{R}^d$. The ambient triangulation used is a Coxeter triangulation of type $\tilde{A}_d$. The time shown in the table is the computation time of the set $\mathcal{S}$.

4. Class `Manifold_tracing` is a function object class for the manifold tracing algorithm from Section 4.2.

5. Class `Cell_complex` computes and stores the Hasse diagram of the cell complex $\mathcal{H}$ from the output by an object of the class `Manifold_tracing`.

The plan of the current section follows these five parts, with an additional Section 4.5.4 for the function classes.

4.5.1 Class `Permutahedral_representation`

Every simplex manipulated in the module `Coxeter triangulation` has to be an object of the class `Permutahedral_representation`. As the name suggests, an object of this class stores the permutahedral representation $\sigma(y, \omega)$ (defined in Definition 3.3.15) with a vertex $y \in \mathbb{Z}^d$ and an ordered partition ω of the interval $\{1, \ldots, d+1\}$ for some specific positive integer d.

Remark 4.5.1. *It is important to note that the indexation of coordinates in a vertex and elements in a partition in the class `Permutahedral_representation` are different from the indexation before. Namely, as it is customary in programming, the indexation starts with* 0. *For instance*

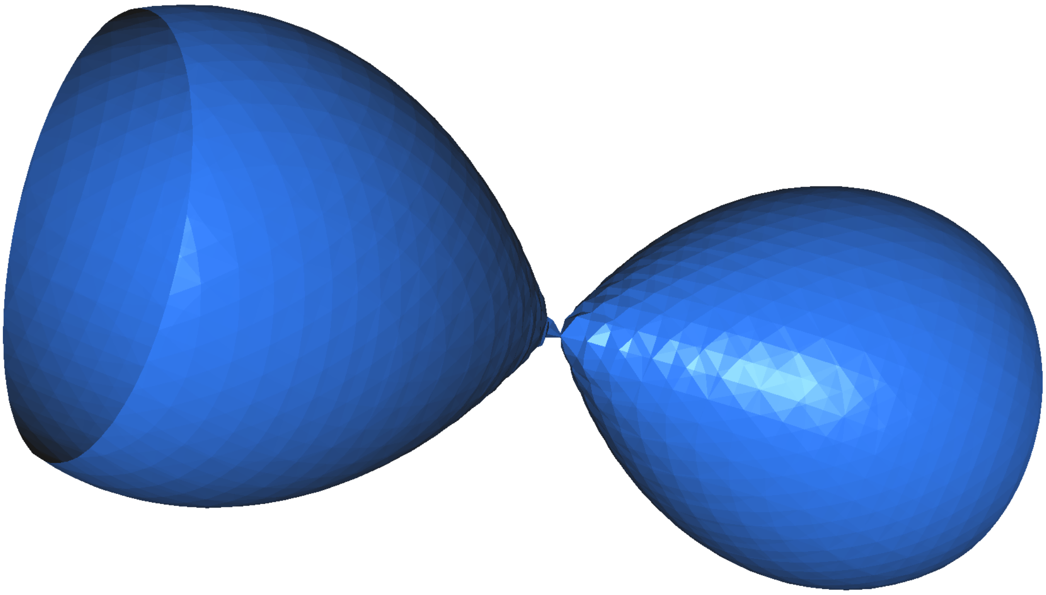


Figure 4.9: The piecewise-linear approximation of a revolution surface obtained from the lemniscate of Bernoulli embedded in $\mathbb{R}^3$ defined by equation $(x^2+y^2+z^2)^2 - 2(x^2-y^2-z^2) = 0$. The surface is cut with a sphere defined by equation $x^2+y^2 = 1.2^2$. The ambient triangulation used is a Coxeter triangulation of type $\tilde{A}_3$ with the diameter 0.07 of the full-dimensional simplices. The size $|\mathcal{S}|$ of the reconstruction is 7499. The execution time of the algorithm is 0.52s.

this means that first coordinate in the vertex v has the index 0*. Also, the domain of the partition* ω *is* $\{0,\ldots,d\}$ *instead of* $\{1,\ldots,d+1\}$*.*

Template parameters. The class `Permutahedral_representation` takes two template parameters `Vertex_` and `Ordered_set_partition_`. These two template parameters specify the data structures for the two constituent parts of a permutahedral representation $\sigma(y,\omega)$: the vertex y and the ordered partition ω respectively. Multiple constraints need to be met by the two template parameters:

- `Vertex_` needs to be a random-access range[14] of integer-type objects.
- `Ordered_set_partition_` needs to be a random-access range of random-access ranges, the latter consisting of integer-typed objects.

Valid permutahedral representations and constructors. There are two constructors for the class `Permutahedral_representation`.

- `Permutahedral_representation()` creates an empty permutahedral representation.
- `Permutahedral_representation(v, omega)` takes two argument: the vertex `v` of type `Vertex_` and the ordered set partition `omega` of type `Ordered_set_partition_`.

[14] Here and in the following, a *range* refers to the standard C++ concept `Range` (documentation). A *random-access range* refers to a range whose iterator type satisfies the concept `RandomAccessIterator` (documentation).

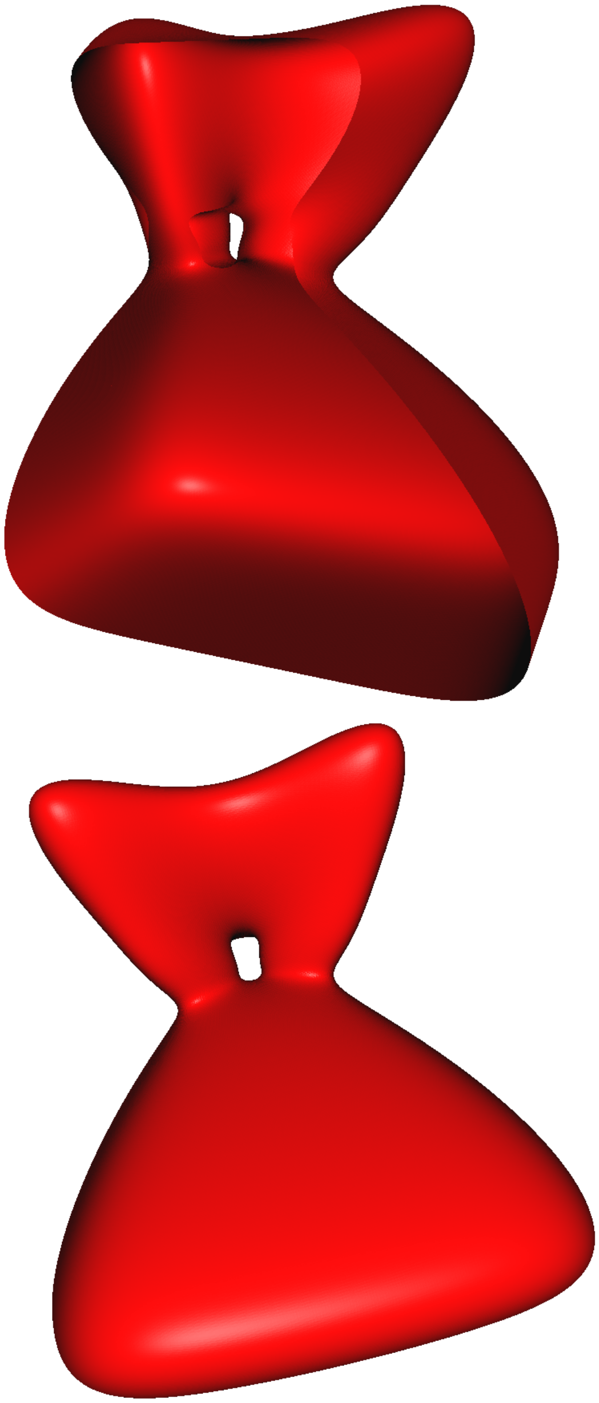


Figure 4.10: Two views of the piecewise-linear approximation of a surface in $\mathbb{R}^3$ given by an equation: $\frac{-x^6-y^6-z^6}{300} + \frac{xy^2z}{2.1} + y^2 + (z-2)^2 = 1$ cut by the hyperplane $x = 0$ in half. The ambient triangulation used is a Coxeter triangulation of type $\tilde{A}_3$ with the diameter 0.23 of the full-dimensional simplices. The size $|\mathcal{S}|$ of the piecewise-linear approximation is 622822. The execution time of the algorithm is 152s.

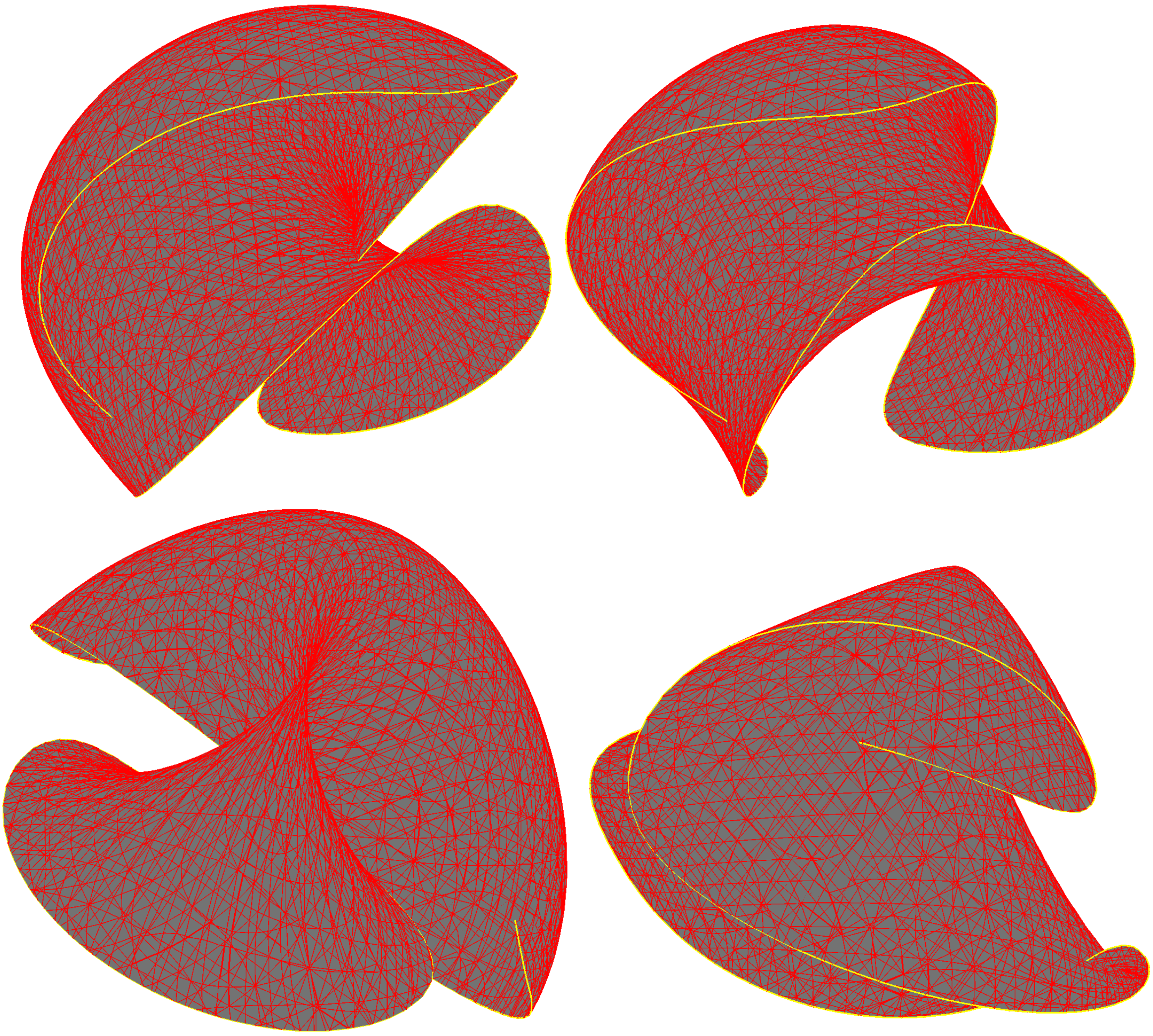


Figure 4.11: Four views of the flat torus in $\mathbb{R}^4$ given by two equations $x_1^2+x_2^2=1$ and $x_3^2+x_4^2=1$ cut by the hypersphere $(x_1-1)^2+x_2^2+(x_3-1)^2+x_4^2=4$, randomly rotated and translated in $\mathbb{R}^4$ and then projected to $\mathbb{R}^3$. The ambient triangulation used is 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.

It is left to the user to control whether the specified vertex `v` and ordered set partition `omega` form a valid permutahedral representation, meaning that v and ω satisfy the following condition:

Condition 4.5.2 (The validity condition)**.** *The elements in the parts in the partition ω form the set $\{0, \dots, d\}$ without repetitions, where d is the size of the vertex v.*

In addition, if the simplex lies in a specific ambient triangulation, d needs to match the dimension of the triangulation.

Element access. Both the vertex v and the ordered partition ω are accessible using the member functions `vertex()` and `partition()` respectively.

Member functions. Multiple other queries are implemented. We tacitly assume here that the stored vertex v and the ordered partition ω satisfy the validity condition (Condition 4.5.2).

- `dimension()` returns the dimension of the simplex. The returned value corresponds to the number of parts in the partition ω minus one.
- `vertex_range()`, `face_range(k)` and `coface_range(l)` return the ranges of vertices, of `k`-dimensional faces and of `l`-dimensional cofaces respectively of the simplex corresponding to the permutahedral representation, for some specified arguments `k` and `l` of a positive integer type. All output ranges are derived from the `boost::iterator_range` class from Boost C++ library[15]. The underlying iterators are derived from the `boost::iterator_facade` class from Boost C++ library[16]. The iterators are implemented using the time- and space-efficient algorithms from Section 3.5. In particular, during the entire iteration over a range, only the current element is explicitly stored. Dereferencing an iterator returns an object of type `Vertex_` for `vertex_range`, and of type `Permutahedral_representation` for `face_range` and `coface_range`.
- `facet_range()` and `cofacet_range()` return facet and cofacet ranges respectively. These two methods are wrappers for `face_range(m-1)` and for `coface_range(m+1)`, where m is the dimension of the simplex, and are added for convenience.
- `is_face_of(other)` takes a parameter `other` of type `Permutahedral_representation` and returns `true` if and only if the simplex represented by the permutahedral representation is a face of the simplex represented by `other`. This function compares the sets of vertices of the two simplices and deciding whether one is a subset of another. This is done by iterating over the vertices for both simplices.

4.5.2 Classes `Freudenthal_triangulation` and `Coxeter_triangulation`

The ambient triangulation $\mathcal{T}$ is represented by the class `Freudenthal_triangulation` or a derived class `Coxeter_triangulation`. Both classes can store any affine transformation of the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ for any positive integer d. The difference is that the constructors `Freudenthal_triangulation(d)` and `Coxeter_triangulation(d)` build the Freudenthal-Kuhn triangulation of $\mathbb{R}^d$ and a Coxeter triangulation of type $\tilde{A}_d$ for the specified d.

[15] The documentation for `boost::iterator_range`.

[16] The documentation for `boost::iterator_facade`.

The underlying data structure is the data structure described in Section 4.1. Internally, the classes `Freudenthal_triangulation` and `Coxeter_triangulation` store a matrix Λ and a vector b (as in Section 4.1.1). Additionally, the dimension d of the triangulation is stored.

Template parameters. Both triangulation classes `Freudenthal_triangulation` and `Coxeter_triangulation` take one template argument `Permutahedral_representation_`, which is initialized by default with type:

```
Permutahedral_representation<std::vector<int>,
    std::vector<std::vector<std::size_t>>>>.
```

The type `Permutahedral_representation_` is required to have the same interface as the class `Permutahedral_representation` as described in Section 4.5.1.

Constructors. There are four available constructors for both triangulation classes:

- The constructors `Freudenthal_triangulation(d)` and `Coxeter_triangulation(d)` take one positive integer argument `d`, which is the dimension of the triangulation. These constructors initialize the offset vector b to 0 and the linear transformation matrix Λ to the identity matrix I_d and to the matrix Λ_C (as in Section 3.1.3) respectively.
- The constructor `Freudenthal_triangulation(d, matrix)` takes two arguments `d` and `matrix`, which are the dimension of the triangulation and the linear transformation matrix respectively. The argument `d` is of a positive integer type. The argument `matrix` is of type `Eigen::MatrixXd` provided by the Eigen C++ library [GJ+10]. This constructor initializes the offset vector b to 0 and the linear transformation matrix Λ to the matrix `matrix`. The derived constructor `Coxeter_triangulation(d, matrix)` is the same.
- The constructor `Freudenthal_triangulation(d, matrix, offset)` takes three arguments `d`, `matrix` and `offset`, which are the dimension of the triangulation, the linear transformation matrix and the offset vector respectively. The argument `d` is of a positive integer type. The arguments `matrix` and `offset` are of type `Eigen::MatrixXd` and `Eigen::VectorXd` respectively provided by the Eigen C++ library [GJ+10]. This constructor initializes the offset vector b to `offset` and the linear transformation matrix Λ to the matrix `matrix`. The derived constructor `Coxeter_triangulation(d, matrix, offset)` is the same.

Access. The dimension d, the matrix Λ and the ordered partition b are accessible using the member functions `dimension()`, `matrix()` and `offset()` respectively.

Member functions. Multiple other queries are implemented.

- `change_matrix(matrix)` changes the matrix Λ to the specified matrix `matrix` of type `Eigen::MatrixXd` provided by the Eigen C++ library [GJ+10].
- `change_offset(offset)` changes the vector b to the specified vector `offset`. The type of `offset` is `Eigen::VectorXd` provided by the Eigen C++ library [GJ+10].
- `locate_point(p)` returns the permutahedral representation of the simplex in the ambient triangulation that contains the specified point `p`. The type of the point `p` is required to be `Eigen::VectorXd` provided by the Eigen C++ library [GJ+10]. The output is of type `Permutahedral_representation_` and always corresponds to the smallest simplex by inclusion that contains `p`.

- `cartesian_coordinates(v)` returns the Cartesian coordinates of the specified vertex `v`. The vertex `v` is required to be of type of type `Permutahedral_representation_::Vertex`. The return value is of type `Eigen::VectorXd` provided by the Eigen C++ library [GJ+10].
- `barycenter(simplex)` returns the barycentre of the specified simplex `simplex` of type `Permutahedral_representation_`. The return value is of type `Eigen::VectorXd` provided by the Eigen C++ library [GJ+10].

4.5.3 Intersection oracle classes

Intersection oracles are represented by two classes:

- `Implicit_manifold_intersection_oracle` for an implicit manifold which is the zero-set of a given function F. Optionally, the manifold can have a boundary, defined by a domain function D (as in Section 4.3.5).
- `Point_cloud_intersection_oracle` for a manifold given by a point cloud P.

The two classes follow the same interface with the exception of the template parameters and the constructors.

Template parameters for `Implicit_manifold_intersection_oracle`. The class takes two parameter arguments `Function_` and `Domain_function_`, which represent the functions F and D. The two types `Function_` and `Domain_function_` are required to have the same interface as the classes in Section 4.5.4. The template argument `Domain_function_` has a default value `Constant_function` (see Section 4.5.4) reserved for the boundaryless implicit manifolds.

Template parameters for `Point_cloud_intersection_oracle`. The class takes one template argument `Point_range_`, which represents the point cloud P. The type `Point_range_` is required to be a random-access range of objects of type `Eigen::VectorXd` provided by the Eigen C++ library [GJ+10].

Constructors for `Implicit_manifold_intersection_oracle`. Two constructors are available:

- `Implicit_manifold_intersection_oracle(F, D)` takes two arguments `F` and `D` of types `Function_` and `Domain_function_` respectively. This constructor is intended for creating an implicit manifold intersection oracle in the case of a manifold with boundary $\mathtt{F}^{-1}(0)\cap\mathtt{D}_-$ (see Section 4.3.5).
- `Implicit_manifold_intersection_oracle(F)` takes an argument `F` of type `Function_`. This constructor is intended for creating a boundaryless implicit manifold intersection oracle $\mathtt{F}^{-1}(0)$. Using this constructor requires the template argument `Domain_function_` to be set to `Constant_function`.

Defining the type with all template arguments can be cumbersome in some cases. For convenience, two static constructor wrappers are available:

- `make_implicit_manifold_intersection_oracle(F, D)` is a wrapper for the constructor `Implicit_manifold_intersection_oracle(F, D)`.
- `make_implicit_manifold_intersection_oracle(F)` is a wrapper for the constructor `Implicit_manifold_intersection_oracle(F)`.

Constructor for `Point_cloud_intersection_oracle`. The constructor is:

- `Point_cloud_intersection_oracle(P, m, gamma)`, where:
  - `P` is a point range of type `Point_range_`,
  - `m` is the intrinsic dimension of the manifold of a positive integer type,
  - `gamma` is the parameter γ from Section 4.3.4 of a floating type (`double` by default).

Member functions. Both intersection oracles have the same member functions.

- `amb_d` returns the ambient dimension of the manifold.
- `cod_d` returns the codimension of the manifold.
- `intersects(simplex, triangulation)` returns `true` if and only if the simplex represented by the object `simplex` that lies in the triangulation `triangulation` intersects the manifold. The type of the argument `simplex` is required to have the same interface as the class `Permutahedral_representation` in Section 4.5.1. The dimension of `simplex` has to be the codimension of the manifold. The type of the argument `triangulation` is required to have the same interface as the class `Freudenthal_triangulation` in Section 4.5.2. The dimension of `triangulation` has to be the same as the codimension of the manifold.
- `intersects_boundary(simplex, triangulation)` returns `true` if and only if the simplex `simplex` that lies in the triangulation `triangulation` intersects the boundary of the manifold. If the manifold does not have boundary, the return value is always `false`. The type of the argument `simplex` is required to have the same interface as the class `Permutahedral_representation` in Section 4.5.1. The dimension of `simplex` has to be one more than the codimension of the manifold. The type of the argument `triangulation` is required to have the same interface as the class `Freudenthal_triangulation` in Section 4.5.2. The dimension of `triangulation` has to be the same as the codimension of the manifold.
- `lies_in_domain(p, triangulation)` returns `true` if and only if the point `p` lies inside the domain D_- (defined in Section 4.3.5). If the manifold does not have boundary, the return value is always `false`. The argument `triangulation` is needed to make a piecewise-linear approximation of the function D and its type is required to have the same interface as the class `Freudenthal_triangulation` (see Section 4.5.2).

4.5.4 Function classes

Any function passed as an argument in `Implicit_manifold_intersection_oracle` is implemented as a class. Here are the implemented functions along with the constructors:

- `Constant_function(d, k, v)` defines a constant function F such that for all $x \in \mathbb{R}^{\mathtt{d}}$, we have $F(x) = \mathtt{v} \in \mathbb{R}^{\mathtt{k}}$. The arguments `d` and `k` are of a positive integer type. The value `v` should be a `k`-dimensional vector of type `Eigen::VectorXd` provided by the Eigen C++ library [GJ+10]. The class `Constant_function` does not define an implicit manifold, but is useful as the domain function when defining boundaryless implicit manifolds in the class `Implicit_manifold_intersection_oracle` (see Section 4.5.3).

- `Function_affine_plane_in_Rd(N, b)` defines an m-dimensional implicit affine plane in the d-dimensional Euclidean space given by a normal matrix `N` and an offset vector `b`. The dimensions m and d are deduced from the dimensions of the matrix `N`, which is $d \times (d-m)$. The dimension of the vector `b` should be the ambient dimension of the manifold. The types of the arguments `matrix` and `b` are `Eigen::MatrixXd` and `Eigen::VectorXd` respectively provided by the Eigen C++ library [GJ+10]. The default value of the argument `b` is the zero-vector.

- `Function_Sm_in_Rd(r, m, d, center)` defines an `m`-dimensional implicit sphere embedded in the `d`-dimensional Euclidean space of radius `r` centered at the point `center`. The dimension of the point `center` should be the ambient dimension of the manifold. The argument `r` is of a floating type (`double` by default). The arguments `m` and `d` are of a positive integer type. The type of the argument `center` is `Eigen::VectorXd` provided by the Eigen C++ library [GJ+10].

- `Function_moment_curve(r, d)` defines the moment curve in the `d`-dimensional Euclidean space of radius `r` given as the parameterized curve (but implemented as an implicit surface):

  $$(r, rt, \dots, rt^{d-1}) \in \mathbb{R}^d, \text{ for } t \in \mathbb{R}.$$

  The argument `r` is of a floating type (`double` by default). The argument `d` are of a positive integer type.

- `Function_torus_in_R3(R, r)` defines a torus in $\mathbb{R}^3$ with the outer radius `R` and the inner radius, given by the equation:

  $$z^2 + (\sqrt{x^2+y^2} - r)^2 - R^2 = 0.$$

  The arguments `R` and `r` are of a floating type (`double` by default).

- `Function_chair_in_R3(a, b, k)` defines the "Chair" surface in $\mathbb{R}^3$ (illustrated in Figure 4.6) defined by the equation:

  $$(x^2+y^2+z^2-ak^2)^2 - b((z-k)^2-2x^2)((z+k)^2-2y^2) = 0.$$

  The arguments `a`, `b` and `k` are of a floating type (`double` by default).

- `Function_iron_in_R3()` defines the "Iron" surface in $\mathbb{R}^3$ (illustrated in Figure 4.10) defined by the equation:

  $$\frac{-x^6-y^6-z^6}{300} + \frac{xy^2z}{2.1} + y^2 + (z-2)^2 = 1.$$

- `Function_lemniscate_revolution_in_R3(a)` defines a revolution surface in $\mathbb{R}^3$ obtained from the lemniscate of Bernoulli (illustrated in Figure 4.9) defined by the equation:

  $$(x^2+y^2+z^2)^2 - 2a^2(x^2-y^2-z^2) = 0.$$

  The argument `a` is of a floating type (`double` by default).

- `Function_whitney_umbrella_in_R3()` defines the Whitney umbrella surface in $\mathbb{R}^3$ defined by the equation:

  $$x^2 - y^2z = 0.$$

The base function classes above can be composed or modified into new functions using the following classes and methods:

- The class `Cartesian_product` expresses the Cartesian product $F_1^{-1}(0) \times \ldots \times F_k^{-1}(0)$ of multiple implicit manifolds as an implicit manifold. For convenience, a static function `make_product_function(functions...)` is provided that takes a pack of function-typed objects as the argument.

- The class `Embed_in_Rd` expresses an implicit manifolds embedded in a higher-dimensional Euclidean space. For convenience, a static function `make_embedding(F, d)` is provided that takes two arguments: a function-type object `F` and a positive integer `d` which represents the dimension of the embedding space.

- The class `Linear_transformation` applies a linear transformation on an implicit manifold. For convenience, a static function `linear_transformation(F, M)` is provided that takes two arguments: a function-type object `F` and a linear-transformation matrix `M` of type `Eigen::MatrixXd` provided by the Eigen C++ library [GJ+10].

- The class `Translate` translates an implicit manifold by a vector. For convenience, a static function `translate(F, v)` is provided that takes two arguments: a function-type object `F` and a translation vector `v` of type `Eigen::VectorXd` provided by the Eigen C++ library [GJ+10].

- The class `Negation` defines the negative of a given function. This class is useful to define the complementary of a given domain, when defining a manifold with boundary. For convenience, a static function `negation(F)` is provided that a function-type object `F` as argument.

- The class `PL_approximation` defines a piecewise-linear approximation of a given function induced by an ambient triangulation. The purpose of this class is to define a piecewise-linear function that is compatible with the description of the function D in Section 4.3.5. For convenience, a static function `make_pl_approximation(F, T)` is provided that takes two arguments: a function-type object `F` and a triangulation `T`. The type of `T` is required to have the same interface as the class `Freudenthal_triangulation` (see Section 4.5.2).

Member functions. All function classes above have the following member functions:

- `amb_d()` outputs the domain dimension of the function, which corresponds to the ambient dimension of the implicit manifold.

- `cod_d()` outputs the codomain dimension of the function, which corresponds to the codimension of the implicit manifold.

- `evaluate(p, result)` that takes as arguments two references `p` and `result` to variables of type `Eigen::VectorXd` provided by the Eigen C++ library [GJ+10]. The method evaluates the function on the point `p` and stores the result in the point `result`.

- `seed(result)` takes as argument a reference to a variable of type `Eigen::VectorXd` provided by the Eigen C++ library [GJ+10]. The method writes in `result` a point in the zero-set of the function. This point can be used as a seed point in the manifold tracing algorithm.

4.5.5 Class Manifold_tracing

The class Manifold_tracing is a wrapper for the manifold tracing algorithm.

Template parameters. The class takes one template parameter Triangulation_ which defines the type of the ambient triangulation. The type Triangulation_ is required to have the same interface as the class Freudenthal_triangulation (see Section 4.5.2).

Constructor. The only available constructor is the default constructor Manifold_tracing().

Member functions. The class Manifold_tracing has two member functions with similar functionality.

- manifold_tracing_algorithm(seeds, T, O, S) performs the manifold tracing algorithm without taking into account the domain, hence reconstructing a boundaryless manifold. It takes three input arguments and one output argument. The input arguments are:

  - A range seeds of seed points of type Eigen::VectorXd provided by the Eigen C++ library [GJ$^+$10].
  - An ambient triangulation T. The type of T is required to have the same interface as the class Freudenthal_triangulation (see Section 4.5.2).
  - An intersection oracle O of type that has the same interface as the classes in Section 4.5.3.

  The output argument is a map S that associates to a simplex the corresponding intersection point. The stored simplices in the map are of type Permutahedral_representation (see Section 4.5.1). The mapped points are of type Eigen::VectorXd provided by the Eigen C++ library [GJ$^+$10]. The set of simplices in the map is the output set $\mathcal{S}$ from Section 4.2.

- manifold_tracing_algorithm(seeds, T, O, S, B) performs the manifold tracing algorithm on a manifold with boundary. This function takes one additional output argument B of the same type as S. The set of simplices in the set B is the output set $\mathcal{B}$ from Section 4.3.5.

4.5.6 Class Cell_complex

The class Cell_complex serves to build a cell complex $\mathcal{H}$ from the maps S and B in the output of the function manifold_tracing_algorithm called on an object of class Manifold_tracing (see Section 4.5.5).

Constructor. The only available constructor is Cell_complex(m) where the argument m stands for the intrinsic dimension of the manifold and the cell complex. The argument m is of a positive integer type.

During the construction of the cell complex, the intrinsic dimension m of the manifold is supposed to be known. We will denote by k the codimension of the manifold.

Cell complex. The cell complex is stored internally as a Hasse diagram in an object of the class Cell_complex. The type of the cells is Hasse_cell<int, double, bool> provided by the class Hasse_diagram in the upcoming module Hasse diagram of Gudhi C++library [GUD]. The cells are stored in a vector hasse_cells_.

Accessing the cells. The cells in the cell complex are accessed through several maps.

- Two vectors of maps `interior_simplex_cell_maps_` and `boundary_simplex_cell_maps_` serve to access the cells that are in one-to-one correspondence with simplices in the sets $\mathcal{S}$ and $\mathcal{B}$ respectively (see Sections 4.2 and 4.3.5). The i-th entry in these two vectors is a map from the simplices of dimension $k+i$ in the set $\mathcal{S}$ or $\mathcal{B}$ (see Section 4.5.1) to the pointers to the associated i-dimensional cells in the vector `hasse_cells_`. The type of the simplices is `Permutahedral_representation`. The map `interior_simplex_cell_maps_[i]` is accessed through the member function `interior_simplex_cell_map(i)`. Similarly, the map `boundary_simplex_cell_maps_[i]` is accessed through the member function `boundary_simplex_cell_map(i)`.

- The map `cell_simplex_map_` serves as the reverse map from pointers to the cells in the vector `hasse_cells_` to simplices in the sets $\mathcal{S}$ and $\mathcal{B}$. This map is accessed through the member function `cell_simplex_map()`.

- The map `cell_point_map_` associates to the 0-dimensional cells in the vector `hasse_cells_` the corresponding Cartesian coordinates of type `Eigen::VectorXd` provided by the Eigen C++ library [GJ+10]. This map is accessed through the member function `cell_point_map()`.

Member functions. The class `Cell_complex` supports four different varieties of the same method `construct_complex` that constructs the cell complex $\mathcal{H}$ from the sets $\mathcal{S}$ and $\mathcal{B}$.

- `construct_complex(S, B, l)` takes as input two maps `S` and `B` in the output of the function `manifold_tracing_algorithm` called on an object of class `Manifold_tracing` (see Section 4.5.5) and a positive integer argument `l`. It then constructs the `l`-skeleton of the cell complex $\mathcal{H}$ using the algorithm from Section 4.3.5.

- `construct_complex(S, B)` is identical to the function `construct_complex(S, B, l)` but it constructs the whole cell complex $\mathcal{H}$ instead of the `l`-skeleton.

- `construct_complex(S, l)` is identical to the function `construct_complex(S, B, l)` but it constructs the `l`-skeleton of the cell complex $\mathcal{H}$ without boundary.

- `construct_complex(S)` is identical to the function `construct_complex(S, l)` but it constructs the whole cell complex $\mathcal{H}$ instead of the `l`-skeleton.

4.6 Discussion

A possible direction for further research on the manifold tracing algorithm is a more detailed analysis of the guarantees on the constructed cell complexes $\mathcal{H}$ and $\mathcal{H}^*$. In particular, two properties are desirable to prove:

- The Hausdorff distance between the cell complexes $\mathcal{H}$ and $\mathcal{H}^*$ and the manifold $\mathcal{M}$ is bounded.

- Both cell complexes $\mathcal{H}$ and $\mathcal{H}^*$ are homeomorphic to the manifold $\mathcal{M}$.

Bibliography

[ABD10] A. Adams, J. Baek, and M. A. Davis. Fast High-Dimensional Filtering Using the Permutohedral Lattice. In *Computer Graphics Forum*, volume 29, pages 753–762. Wiley Online Library, 2010.

[AG80] Eugene Allgower and Kurt Georg. Simplicial and continuation methods for approximating fixed points and solutions to systems of equations. *Siam review*, 22(1):28–85, 1980.

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

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

[AS85] Eugene L Allgower and Phillip H Schmidt. An algorithm for piecewise-linear approximation of an implicitly defined manifold. *SIAM journal on numerical analysis*, 22(2):322–346, 1985.

[BA76] I. Babuška and A. K. Aziz. On the angle condition in the finite element method. *SIAM Journal on Numerical Analysis*, 13(2):214–226, 1976.

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

[BDG13] J.-D. Boissonnat, R. Dyer, and A. Ghosh. The Stability of Delaunay Triangulations. *International Journal of Computional Geometry & Applications*, 23(4-5):303–334, 2013.

[BDG14] Jean-Daniel Boissonnat, Ramsay Dyer, and Arijit Ghosh. Delaunay stability via perturbations. *International Journal of Computational Geometry & Applications*, 24(02):125–152, 2014.

[BDG15] Jean-Daniel Boissonnat, Ramsay Dyer, and Arijit Ghosh. A Probabilistic Approach to Reducing Algebraic Complexity of Delaunay Triangulations. In *Algorithms - ESA 2015 - 23rd Annual European Symposium, Patras, Greece, September 14-16, 2015, Proceedings*, pages 595–606, 2015.

[Bel57] Richard Bellman. *Dynamic programming*. Princeton University Press, 1957.

[BG05] Christian Blanchet and Etienne Gallais. Combinatorial topology and discrete morse theory. In *Differential geometry and topology, discrete and computational geometry*, pages 31–72. IOS Press, 2005.

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

[Blo88] Jules Bloomenthal. Polygonization of implicit surfaces. *Computer Aided Geometric Design*, 5(4):341–355, 1988.

[BLW18] Jean-Daniel Boissonnat, André Lieutier, and Mathijs Wintraecken. The reach, metric distortion, geodesic convexity and the variation of tangent spaces. In *34th International Symposium on Computational Geometry, SoCG 2018, June 11-14, 2018, Budapest, Hungary*, pages 10:1–10:14, 2018.

[BO05] Jean-Daniel Boissonnat and Steve Oudot. Provably good sampling and meshing of surfaces. *Graphical Models*, 67(5):405–451, 2005.

[Bou02] Nicolas Bourbaki. Lie groups and Lie algebras. Chapters 4–6. Translated from the 1968 French original by Andrew Pressley. Elements of Mathematics, 2002.

[BSW09] Mikhail Belkin, Jian Sun, and Yusu Wang. Constructing laplace operator from point clouds in r d. In *Proceedings of the twentieth annual ACM-SIAM symposium on Discrete algorithms*, pages 1031–1040. Society for Industrial and Applied Mathematics, 2009.

[BWC00] Praveen Bhaniramka, Rephael Wenger, and Roger Crawfis. Isosurfacing in higher dimensions. In *Proceedings of the conference on Visualization'00*, pages 267–273. IEEE Computer Society Press, 2000.

[CC09] Siu-Wing Cheng and Man-Kwun Chiu. Dimension detection via slivers. In *Proceedings of the twentieth annual ACM-SIAM symposium on Discrete algorithms*, pages 1001–1010. Society for Industrial and Applied Mathematics, 2009.

[CC14] Siu-Wing Cheng and Man-Kwun Chiu. Implicit manifold reconstruction. In *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms*, pages 161–173. SIAM, 2014.

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

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

[CFF85] James C. Cavendish, David A. Field, and William H. Frey. An apporach to automatic three-dimensional finite element mesh generation. *International Journal for Numerical Methods in Engineering*, 21(2):329–347, 1985.

[Cox34] Harold SM Coxeter. Discrete groups generated by reflections. *Annals of Mathematics*, pages 588–621, 1934.

[CP98] Shek Ling Chan and Enrico O Purisima. A new tetrahedral tesselation scheme for isosurface generation. *Computers & Graphics*, 22(1):83–90, 1998.

[CS87] J. H. Conway and N. J. A. Sloane. *Sphere-packings, Lattices, and Groups.* Springer-Verlag New York, Inc., New York, NY, USA, 1987.

[CS09] Hamish Carr and Jack Snoeyink. Representing interpolant topology for contour tree computation. In *Topology-Based Methods in Visualization II*, pages 59–73. Springer, 2009.

[CSA03] Hamish Carr, Jack Snoeyink, and Ulrike Axen. Computing contour trees in all dimensions. *Computational Geometry*, 24(2):75–94, 2003.

[CWW08] Siu-Wing Cheng, Yajun Wang, and Zhuangzhi Wu. Provable dimension detection using principal component analysis. *International Journal of Computational Geometry & Applications*, 18(05):415–440, 2008.

[Dan95] Chuangyin Dang. *Triangulations and simplicial methods*, volume 421. Lecture notes in Economics and Mathematical Systems, 1995.

[DGK63] Ludwig Danzer, Branko Grünbaum, and Victor Klee. Helly's theorem and its relatives. 1963.

[DLMF] *NIST Digital Library of Mathematical Functions.* http://dlmf.nist.gov/, Release 1.0.22 of 2019-03-15. F. W. J. Olver, A. B. Olde Daalhuis, D. W. Lozier, B. I. Schneider, R. F. Boisvert, C. W. Clark, B. R. Miller and B. V. Saunders, eds.

[DM82] Wolfgang A Dahmen and Charles A Micchelli. On the linear independence of multivariate b-splines, i. triangulations of simploids. *SIAM Journal on Numerical Analysis*, 19(5):993–1012, 1982.

[Dür88] Martin J Dürst. Re: additional reference to marching cubes. *ACM SIGGRAPH Computer Graphics*, 22(2):72–73, 1988.

[DWLT90] David P Dobkin, Allan R Wilks, Silvio VF Levy, and William P Thurston. Contour tracing by piecewise linear approximations. *ACM Transactions on Graphics (TOG)*, 9(4):389–423, 1990.

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

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

[EK12] Herbert Edelsbrunner and Michael Kerber. Dual complexes of cubical subdivisions of $\mathbb{R}^n$. *Discrete & Computational Geometry*, 47(2):393–414, 2012.

[Fed59] Herbert Federer. Curvature measures. *Transactions of the American Mathematical Society*, 93(3):418–491, 1959.

[FIK$^+$18] Charles Fefferman, Sergei Ivanov, Yaroslav Kurylev, Matti Lassas, and Hariharan Narayanan. Fitting a putative manifold to noisy data. In Sébastien Bubeck, Vianney Perchet, and Philippe Rigollet, editors, *Proceedings of the 31st Conference On Learning Theory*, volume 75 of *Proceedings of Machine Learning Research*, pages 688–720. PMLR, 06–09 Jul 2018.

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

[GJ+10] Gaël Guennebaud, Benoît Jacob, et al. Eigen v3. http://eigen.tuxfamily.org, 2010.

[Gnu88] Stefan Gnutzmann. *Stückweise lineare Approximation implizit definierter Mannigfaltigkeiten.* na, 1988.

[Gra82] Alfred Gray. Comparison theorems for the volumes of tubes as generalizations of the weyl tube formula. *Topology*, 21(2):201–228, 1982.

[GUD] GUDHI Project. GUDHI Editorial Board.

[GW03] Joachim Giesen and Uli Wagner. Shape dimension and intrinsic metric from samples of manifolds with high co-dimension. In *Proceedings of the nineteenth annual symposium on Computational geometry*, pages 329–337. ACM, 2003.

[HA05] Matthias Hein and Jean-Yves Audibert. Intrinsic dimensionality estimation of submanifolds in r d. In *Proceedings of the 22nd international conference on Machine learning*, pages 289–296. ACM, 2005.

[Har69] Frank Harary. Graph theory. Technical report, MICHIGAN UNIV ANN ARBOR DEPT OF MATHEMATICS, 1969.

[Hen07] Michael E Henderson. Higher-dimensional continuation. In *Numerical continuation methods for dynamical systems*, pages 77–115. Springer, 2007.

[HP16] George Haller and Sten Ponsioen. Nonlinear normal modes and spectral submanifolds: existence, uniqueness and use in model reduction. *Nonlinear dynamics*, 86(3):1493–1534, 2016.

[Hum92] James E Humphreys. *Reflection groups and Coxeter groups*, volume 29. Cambridge university press, 1992.

[Jam76] Pierre Jamet. Estimations d'erreur pour des éléments finis droits presque dégénérés. *Revue française d'automatique, informatique, recherche opérationnelle. Analyse numérique*, 10(1):43–60, 1976.

[Jun01] H. Jung. Über die kleinste Kugel, die eine räumliche Figur einschliesst. *Journal reine angewandte Mathematik*, 123:241–257, 1901.

[Kla85] MS Klamkin. Inequality for a simplex. *SIAM Review*, 27(4):576, 1985.

[Kří92] Michal Křížek. On the maximum angle condition for linear tetrahedral elements. *SIAM Journal on Numerical Analysis*, 29(2):513–520, 1992.

[Kuh60] Harold W Kuhn. Some combinatorial lemmas in topology. *IBM Journal of research and development*, 4(5):518–524, 1960.

[Kuh68] Harold W Kuhn. Simplicial approximation of fixed points. *Proceedings of the National Academy of Sciences*, 61(4):1238–1242, 1968.

[LB05] Elizaveta Levina and Peter J Bickel. Maximum likelihood estimation of intrinsic dimension. In *Advances in neural information processing systems*, pages 777–784, 2005.

[LC87] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *ACM siggraph computer graphics*, volume 21, pages 163–169. ACM, 1987.

[LMR17] Anna V Little, Mauro Maggioni, and Lorenzo Rosasco. Multiscale geometric methods for data sets i: Multiscale svd, noise and curvature. *Applied and Computational Harmonic Analysis*, 43(3):504–567, 2017.

[LS07] François Labelle and Jonathan Richard Shewchuk. Isosurface stuffing: fast tetrahedral meshes with good dihedral angles. In *ACM Transactions on Graphics (TOG)*, volume 26, page 57. ACM, 2007.

[LZ13] Jian Liang and Hongkai Zhao. Solving partial differential equations on point clouds. *SIAM Journal on Scientific Computing*, 35(3):A1461–A1486, 2013.

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

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

[Moo92] Douglas W Moore. Simplical mesh generation with applications. Technical report, Cornell University, 1992.

[MSS94] Claudio Montani, Riccardo Scateni, and Roberto Scopigno. A modified look-up table for implicit disambiguation of marching cubes. *The visual computer*, 10(6):353–355, 1994.

[MTCW10] Shawn Martin, Aidan Thompson, Evangelos A Coutsias, and Jean-Paul Watson. Topology of cyclo-octane energy landscape. *The journal of chemical physics*, 132(23):234115, 2010.

[Mun66] James R Munkres. *Elementary differential topology*, volume 54. Princeton University Press, 1966.

[MW97] John Willard Milnor and David W Weaver. *Topology from the differentiable viewpoint.* Princeton university press, 1997.

[NH91] Gregory M Nielson and Bernd Hamann. The asymptotic decider: resolving the ambiguity in marching cubes. In *Proceedings of the 2nd conference on Visualization'91*, pages 83–91. IEEE Computer Society Press, 1991.

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

[PV04] Simon Plantinga and Gert Vegter. Isotopic approximation of implicit curves and surfaces. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 245–254. ACM, 2004.

[Raj94] VT Rajan. Optimality of the Delaunay triangulation in $\mathbb{R}^d$. *Discrete & Computational Geometry*, 12(2):189–202, 1994.

[Rot12] Joseph J Rotman. *An introduction to the theory of groups*, volume 148. Springer Science & Business Media, 2012.

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

[Sar42] Arthur Sard. The measure of the critical values of differentiable maps. *Bulletin of the American Mathematical Society*, 48(12):883–890, 1942.

[Sca67] Herbert Scarf. The approximation of fixed points of a continuous mapping. *SIAM Journal on Applied Mathematics*, 15(5):1328–1343, 1967.

[SD08] Dan A Simovici and Chabane Djeraba. Partially ordered sets. In *Mathematical Tools for Data Mining*, pages 129–172. Springer, 2008.

[Sew54] Harold Herbert Seward. *Information sorting in the application of electronic digital computers to business operations*. PhD thesis, Massachusetts Institute of Technology. Department of Electrical Engineering, 1954.

[She02] Jonathan Shewchuk. What is a good linear finite element? interpolation, conditioning, anisotropy, and quality measures (preprint). *University of California at Berkeley*, 73:137, 2002.

[SP93] Steven W Shaw and Christophe Pierre. Normal modes for non-linear vibratory systems. 1993.

[Spi99] Michael Spivak. *A Comprehensive Introduction to Differential Geometry*. Number v. 1 in A Comprehensive Introduction to Differential Geometry. Publish or Perish, Incorporated, 1999.

[SSM98] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural computation*, 10(5):1299–1319, 1998.

[Sta04] Richard P Stanley. An introduction to hyperplane arrangements. *Geometric combinatorics*, 13:389–496, 2004.

[Sti30] James Stirling. *Methodus differentialis, sive Tractatus de summatione et interpolatione serierum infinitarum*. 1730.

[Str69] Volker Strassen. Gaussian elimination is not optimal. *Numerische mathematik*, 13(4):354–356, 1969.

[Syn57] John Lighton Synge. *The hypercircle in mathematical physics*. CUP Archive, 1957.

[Tar76] Robert Endre Tarjan. Edge-disjoint spanning trees and depth-first search. *Acta Informatica*, 6(2):171–185, 1976.

[TDSL00] Joshua B Tenenbaum, Vin De Silva, and John C Langford. A global geometric framework for nonlinear dimensionality reduction. *science*, 290(5500):2319–2323, 2000.

[Tho93] GL Thompson. Generalized permutation polytopes and exploratory graphical methods for ranked data. *The Annals of Statistics*, pages 1401–1430, 1993.

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

[Top] Jaap Top. Dynkin diagrammen en Wortelsystemen.

[TPG99] Graham M Treece, Richard W Prager, and Andrew H Gee. Regularised marching tetrahedra: improved iso-surface extraction. *Computers & Graphics*, 23(4):583–598, 1999.

[Vav96] Stephen A Vavasis. Stable finite elements for problems with wild coefficients. *SIAM journal on numerical analysis*, 33(3):890–916, 1996.

[VGdF07] Luiz Velho, Jonas Gomes, and Luiz H de Figueiredo. *Implicit objects in computer graphics.* Springer Science & Business Media, 2007.

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

[WB96] Chris Weigle and David C Banks. Complex-valued contour meshing. In *Proceedings of the 7th conference on Visualization'96*, pages 173–ff. IEEE Computer Society Press, 1996.

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

[Wey39] Hermann Weyl. On the volume of tubes. *American Journal of Mathematics*, 61(2):461–472, 1939.

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

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

[ZSK94] Chen Zhou, Renben Shu, and Mohan S Kankanhalli. Handling small features in isosurface generation using marching cubes. *Computers & graphics*, 18(6):845–848, 1994.