

Die approbierte Originalversion dieser
Dissertation ist in der Hauptbibliothek der
Technischen Universität Wien aufgestellt und
zugänglich.

<http://www.ub.tuwien.ac.at>



The approved original version of this thesis is
available at the main library of the Vienna
University of Technology.

<http://www.ub.tuwien.ac.at/eng>



TECHNISCHE
UNIVERSITÄT
WIEN

Vienna University of Technology

Dissertation

Geometric Optimization in Minkowski Space

ausgeführt zum Zwecke der Erlangung des akademischen Grades eines
Doktors der technischen Wissenschaften unter der Leitung von

Ao.Univ.Prof. Dr. Martin Peternell
Institut für Diskrete Mathematik und Geometrie (E104)

eingereicht an der Technischen Universität Wien,
Fakultät für Mathematik und Geoinformation,
von

Mag. Bernhard Blaschitz
Matr.Nr.: 0201684
Tamariskengasse 28, 1220 Wien

Wien, am 11.11.2013

Kurzfassung

Für eine 1-parametrische Menge von Kreisen $F(t)$: $(\mathbf{x} - \mathbf{m}(t))^2 = r^2(t)$ ist die Hüllkurve gegeben durch $F(t) \cap F_t(t)$ mit $F_t(t)$: $(\mathbf{x} - \mathbf{m}(t))\dot{\mathbf{m}} + r\dot{r} = 0$. Es ist in diesem Zusammenhang vorteilhaft, ein Punktmodell der Menge der Kreise der euklidischen Ebene zu studieren. Dabei wird jedem Kreis ein Punkt im $\mathbb{R}^{2,1}$ so zugeordnet, dass die ersten beiden Koordinaten der Mittelpunkt und die dritte der Radius ist. Jede Kurve l : $\mathbf{p}(t) = (p_1, p_2, p_3)(t)$, $\dot{\mathbf{p}}(t) \neq 0$ im $\mathbb{R}^{2,1}$ definiert eine 1-parametrische Menge von Kreisen, ihre Hüllkurve ist die Spurkurve der Böschungstorse Γ_l . Diese Spurkurve besteht üblicherweise aus zwei Kurvenzweigen und ist reell für $\langle \dot{\mathbf{p}}, \dot{\mathbf{p}} \rangle_L \geq 0$ und die Zweige sind verschieden für $\langle \dot{\mathbf{p}}, \dot{\mathbf{p}} \rangle_L > 0$, wobei $\langle \mathbf{x}, \mathbf{y} \rangle_L = x_1y_1 + x_2y_2 - x_3y_3$ das pseudo-euklidische Skalarprodukt in $\mathbb{R}^{2,1}$ ist. Das heißt, dass Kurven, deren Ableitungen steiler als $\frac{\pi}{4}$ sind, keine reelle Spurkurven besitzen.

Angenommen, man hat eine diskrete Menge von Kreisen \mathbf{p}_i in der Ebene und sucht eine reelle Hüllkurve. In der Literatur werden dafür Heuristiken verwendet, die z.B. die Tangentenpaare zweier benachbarter Kreise verbinden, wodurch sich eine Einhüllende ergibt, die aus Kreissegmenten und Geraden besteht und bestenfalls C^1 ist.

Ein anderer Zugang besteht darin, die Bilder der \mathbf{p}_i im $\mathbb{R}^{2,1}$ zu betrachten und einen beliebig glatten B-Spline \mathbf{b} mittels einer quadratischen Optimierung zu fitten. Das gewährleistet aber nicht, dass $\langle \dot{\mathbf{b}}, \dot{\mathbf{b}} \rangle_L > 0$, also die Hüllkurve reell ist, weshalb in der Literatur z.B. Hermite-Interpolationen mit flachen Kurvenstücken oder mit Biarcs durchgeführt werden. Der neue Zugang dieser Arbeit ist eine Umformulierung des Problems in ein Optimierungsproblem unter Nebenbedingungen: Die quadratische Zielfunktion minimiert die Distanz zwischen den zyklographischen Bildern der Kreise \mathbf{p}_i und dem kubischen B-Spline \mathbf{b} unter Berücksichtigung des Fußpunktproblems, d.h. in einem iterativen Prozess wird die Kurve auch unparametrisiert, was eine bessere Näherung bewirkt.

Die Nebenbedingung der Realität der Hüllkurve liefert eine quadratische, nichtkonvexe Gleichung, sodass das gesamte Problem nur mit Hilfe einer Linearisierung der Nebenbedingung gelöst werden kann. Das Problem wird mittels der Interior Point Methode gelöst, wobei auch Regularisierungen und andere Optimierungsdetails besprochen werden, insbesondere in Bezug auf die Nichtkonvexität des Raumes $\mathbb{R}^{2,1}$. Die hier beschriebene Fragestellung wird auch auf Kugeln, die durch Punkte im $\mathbb{R}^{3,1}$ repräsentiert werden, angewandt; im 1-parametrischen Fall ergibt sich somit eine Methode zur Interpolation mit Kanalflächen, im 2-parametrischen allgemeine Kugelhüllflächen, die in dieser Form bisher nie untersucht wurden. Hierfür wurde ein Kalkül entwickelt, der sich mit der Frage der Steilheit von 2-Ebenen im $\mathbb{R}^{3,1}$ auseinandersetzt.

Abstract

The envelope of a 1-parameter family of circles $F(t): (\mathbf{x} - \mathbf{m}(t))^2 = r^2(t)$ is given as $F(t) \cap F_t(t)$ with $F_t(t): (\mathbf{x} - \mathbf{m}(t))\dot{\mathbf{m}} + rr' = 0$. We will study the set of circles in the plane in a *point set model*: Every circle is assigned to a point in $\mathbb{R}^{2,1}$ such that the first two coordinates are its center and the third is its radius. Every curve $l: \mathbf{p}(t) = (p_1, p_2, p_3)(t)$, $\dot{\mathbf{p}}(t) \neq 0$ in $\mathbb{R}^{2,1}$ defines a 1-parameter set of circles, their envelope is the intersection of a torsal surface of constant slope Γ_l with the xy -plane. This intersection is a curve and usually consists of two branches, which are real for $\langle \dot{\mathbf{p}}, \dot{\mathbf{p}} \rangle_L \geq 0$ and do not coincide for $\langle \dot{\mathbf{p}}, \dot{\mathbf{p}} \rangle_L > 0$, where $\langle \mathbf{x}, \mathbf{y} \rangle_L = x_1y_1 + x_2y_2 - x_3y_3$ is the pseudo Euclidean inner product in $\mathbb{R}^{2,1}$. Therefore, the images of curves, whose derivative is steeper than $\frac{\pi}{4}$, are not real.

Assuming a discrete set of circles \mathbf{p}_i in the plane, a real envelope is looked for. There exist different heuristics in the literature, e.g. the two common tangents of two neighboring circles, which results in envelopes consisting of circular arcs and line segments; their connection is C^1 at best.

Another approach is to fit an arbitrarily smooth B-Spline \mathbf{b} to the images of the \mathbf{p}_i in $\mathbb{R}^{2,1}$ via a quadratic optimization. That does not ensure that $\langle \dot{\mathbf{b}}, \dot{\mathbf{b}} \rangle_L > 0$, e.g. that the envelope is real, so some authors use a Hermite interpolation with flat curve segments or biars.

The new approach of this work is reformulating the original problem as a constrained optimization: the quadratic objective function minimizes the distance between the cyclographic images of circles \mathbf{p}_i and a cubic B-Spline \mathbf{b} by observing the footpoint problem, i.e. the curve is reparametrized in an iterative procedure, which brings a better fit.

The reality of the envelope results in a quadratic, but non convex constraint, which can be linearized. This linearization is discussed in detail, as its formulation is central to this work.

The optimization problem is solved with an Interior Point Method, and regularizations and other details of optimization are discussed, especially the non-convexity of the space $\mathbb{R}^{2,1}$. The problems discussed for circles are also generalized for spheres; in the 1-parameter case that leads to a new method for interpolation points in $\mathbb{R}^{3,1}$ by curves, which translates to interpolation of spheres by canal surfaces. Approximating 2-parameter sets of points by surfaces in $\mathbb{R}^{3,1}$ gives rise to general envelope surfaces of 2-parameter families of spheres, that have not been studied before in this generality. For this, a calculus was developed, that classifies 2-planes in $\mathbb{R}^{3,1}$ according to their steepness.

Acknowledgments

I want to thank all the people from the Geometric Modeling and Industrial Geometry group at the Technical University of Vienna, my advisor Martin Peternell, and all the colleagues I got to know over my three and a half years of work as a teaching assistant there, especially Simon Flöry and David Gruber.

A good part of this thesis was written during my four months stay at Freie Universität Berlin, where Konrad Polthier was kind enough to invite me, another part during my parental leave in 2012.

None of this work would have been possible without the loving support of my wife, who always believed in this project.

To you all, thank you!

Contents

1	Introduction	1
1.1	Skinning, sweeping and envelopes	2
1.1.1	Skinning	2
1.1.2	Sweeping	2
1.1.3	Envelopes	3
1.2	Applications	3
1.2.1	Error bound paths	3
1.2.2	Gently inclined roads	3
1.2.3	Medial axis transform	4
1.2.4	Offset curve approximation for varying distance	4
1.3	Thesis overview	4
2	Minkowski space	6
2.1	Minkowski geometry	6
2.1.1	Lorentz inner product	6
2.1.2	Types of vectors	8
2.1.3	Lorentz cone Γ	10
2.1.4	Lorentz transformations	11
2.2	Cyclography	11
2.2.1	Laguerre geometry in \mathbb{R}^2	11
2.2.2	Laguerre geometry in \mathbb{R}^3	12
2.2.3	Laguerre transformations	12
2.2.4	Offsets and the connection to Minkowski sums	13
2.2.5	Distances	14
2.3	Curves in Minkowski space	14
2.3.1	Offsets and distance function	14
2.3.2	Cyclographic image of curves in $\mathbb{R}^{2,1}$	16
2.3.3	Cyclographic image of curves in $\mathbb{R}^{3,1}$	17
2.4	Surfaces in Minkowski space	18
3	B-Splines	19
3.1	Interpolation and approximation	19
3.1.1	Curve interpolation	20
3.1.2	Curve approximation	21
3.1.3	B-Spline surfaces	21

3.2	Measuring distances	22
3.2.1	Footpoint problem	22
3.2.2	Distance function	23
3.3	Geometric regularization	24
3.3.1	Tikhonov regularization	25
3.3.2	Curves	25
3.3.3	Surfaces	26
4	Envelopes of Circles	27
4.1	Introduction	27
4.1.1	Overview of this chapter	28
4.1.2	Previous Work	28
4.2	Problem Statement	30
4.2.1	Objective function	32
4.2.2	Quadratic Constraint	34
4.2.3	Optimization procedure	35
4.3	Linearizing the quadratic constraint	36
4.3.1	Local linearization of Γ	36
4.3.2	Projection orthogonal to Γ	37
4.3.3	Matrix formulation of the linearization	38
4.4	Solving the Optimization	38
4.4.1	Algorithm for Interior Point Method	39
4.4.2	Initial position	40
4.4.3	Iteration of the Interior Point Algorithm	42
4.4.4	Computed examples	44
4.5	Testing the results	44
5	Envelopes of Spheres	45
5.1	Canal surfaces	45
5.1.1	Optimization formulation	46
5.1.2	Objective function in $\mathbb{R}^{3,1}$	46
5.1.3	Steepness constraint in $\mathbb{R}^{3,1}$	47
5.1.4	Solving the optimization for canal surfaces	48
5.2	2-planes in $\mathbb{R}^{3,1}$	48
5.2.1	Wedge product in 3-space	48
5.2.2	Wedge product in 4-space	50
5.2.3	Classification of 2-planes in Minkowski space	51
5.3	Hyperbolic paraboloids in $\mathbb{R}^{3,1}$	52
5.3.1	Tangent planes of HP surfaces	54
5.3.2	Computing $\Phi_{3,1}$ for HP surfaces in $\mathbb{R}^{3,1}$	57
5.4	Surface optimization in $\mathbb{R}^{3,1}$	58
5.4.1	Objective function and collocation matrices	60
5.4.2	Notation for B-spline surfaces	60

5.4.3	Constraints for surface optimization	61
5.4.4	Ruled surface optimization	64
5.4.5	Hyperbolic paraboloidal surface optimization	65
5.4.6	Footpoints and Regularization	66
5.5	Optimization framework	66
5.5.1	Variables of the optimization	67
5.5.2	Initial values	68
6	Examples	70
6.1	Envelopes of circles	71
6.1.1	General Strategy	71
6.1.2	Effects of the steepness constraint	76
6.2	1-parameter envelopes of spheres	77
6.3	2-parameter envelopes of spheres	79
6.3.1	Wave example	79
6.3.2	Sine cylinder	80
6.3.3	Peaks example	80
	List of Figures	87
	References	88

1 Introduction

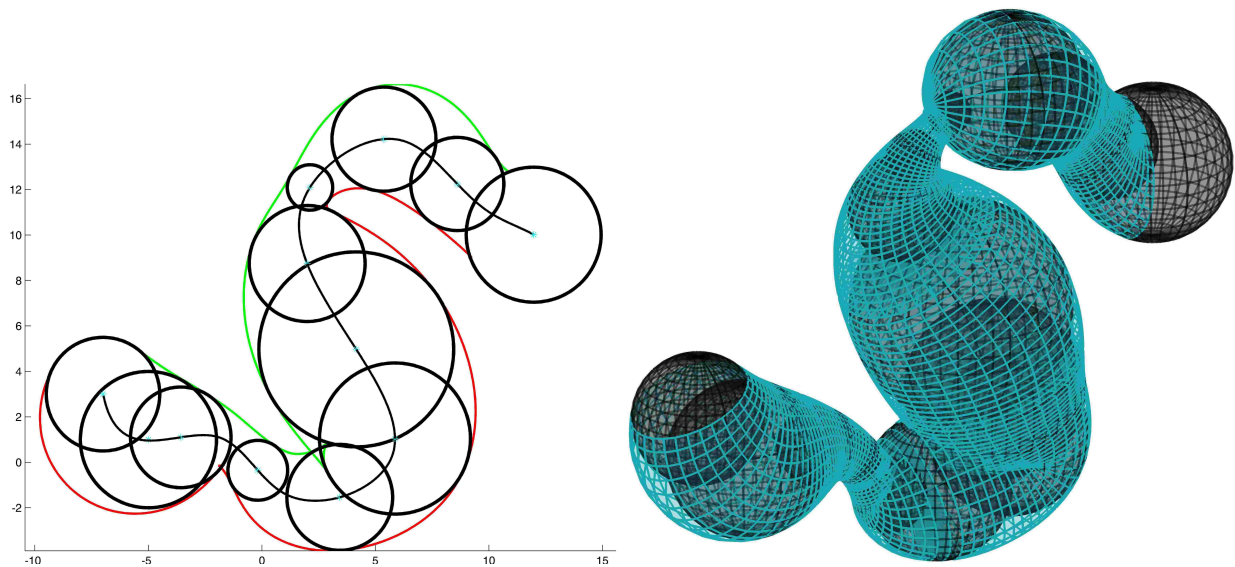


Figure 1.1: *left*: Given a set of circles (*black*), what is the optimal envelope (*green* and *red* curves)? Or equivalently (*right*) given a set of spheres (*black*) what is the optimal canal surface enveloping them?

The main question motivating this work is as follows: Given a set of circles in \mathbb{R}^2 , how can we find an optimal envelope? Equivalently for \mathbb{R}^3 : Given a set of spheres, what is the optimal envelope? To further narrow down the question, the input circles in \mathbb{R}^2 are a one-parameter family and prescribe an ordering. In \mathbb{R}^3 we want to consider one- and two-parameter families of spheres, but we always assume either a linear ordering or a setup on a grid.

Of course there are many curves/surfaces enveloping the input circles/spheres and hence *optimality* can only be stated with respect to certain criteria. If we take one of the two envelope curves in \mathbb{R}^2 , we could ask for the shortest one, but in curve fitting there is always a trade off between the length of a curve and its smoothness, e.g. the shortest curve interpolating a given set of points is the polyline consisting of all the line segments connecting the points (remember we have a predefined ordering, otherwise we would also have to solve the much harder problem of the *traveling salesperson*). As smoothness, we aim for C^2 , i.e. twice differentiable, which we can achieve with cubic B-splines (if there are no cusps or trimmings).

1.1 Skinning, sweeping and envelopes

Related topics in computer graphics and computational geometry are described in the following sections. We point out that different words are used for the same process. We will call the process of finding a smooth geometric object that approximates circles/spheres *enveloping* and that this geometric object shall be called *envelope*.

1.1.1 Skinning

The term *skinning* seems to convey different meanings:

1. In **computer graphics**, see e.g. [21], (character) skinning refers to finding 1.) a skeleton structure that gives a good shape approximation, i.e. a meaningful simplification of a surface and 2.) a correspondence between the skin/mesh/shape and the skeleton. The general assumption is that any movement of the shape only needs to be computed on the skeleton, if correspondences are known.
2. In computer aided geometric design (**CAGD**), see e.g. [34], it refers to connecting curves in \mathbb{R}^3 to a surface, such that the original curves lie on the surface.

The papers [44] (sec. 4.1.2) and [25] (sec. 4.1.2) refer to *ball skinning* as the process of finding an envelope for a discrete set of circles in \mathbb{R}^2 , but actually they use the circles to compute points and tangents and interpolate these, which is closer to the CAGD meaning. [45] is about *ball skinning* in \mathbb{R}^3 and again, spheres are intersected by planes, thus giving circles, which are then interpolated by a surface, hence returning to the original CAGD meaning of skinning (see above).

On the other hand, Cheng and Shi [6], Edelsbrunner [9] and Kruithof and Vegter [24] consider the following problem: given a set of balls (=atoms), find the envelope of their union (=molecule).

1.1.2 Sweeping

Volume sweeping, as for example shown in [49] (envelopes, Minkowski sums), deals with (one-parameter families of) motions of some volumes. Thus the size of the object is always the same, which is just a special case of our problem, but the types of objects it applies to are more general. Overview of methods and many applications in [1].

1.1.3 Envelopes

We mention two survey papers on (smooth) envelopes by Pottmann&Peternell, namely [39] and [40]. The first one finds envelopes for smooth families of spheres and planes, the latter (and the numerous references mentioned therein) describes how to approximate lines, planes and spheres.

More on this subject can be found in section 4.1.2, where we discuss previous work.

1.2 Applications

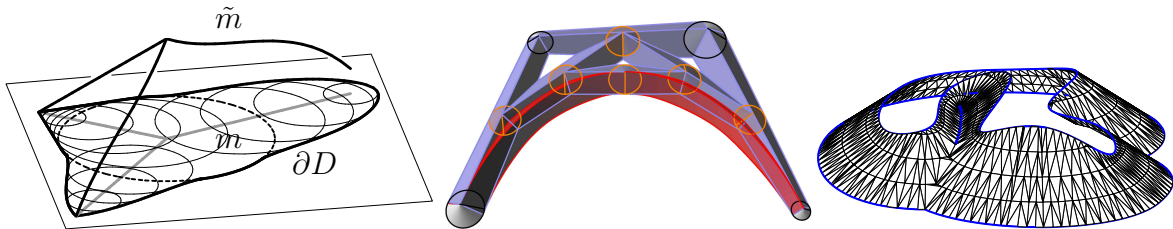


Figure 1.2: Applications of this work in different fields. *left*: The medial axis transform via cyclography (by courtesy of the authors of [22]); *middle*: error bound paths, i.e. the *gray* areas are positions of control points that are not known exactly, intermediate discs of the de Casteljau algorithm in *orange*, the corresponding B-Spline curve is shown in *red*; *right*: a road (*blue* curve) that can not have steeper inclination than a certain angle (by courtesy of the authors of [33]).

The applications of this work could be found in skinning of a family of spheres or sweeping spheres, as described in section 1.1, as well as some less obvious ones, which we want to present here.

1.2.1 Error bound paths

Imagine a set of control points \mathbf{b}_i in \mathbb{R}^2 for a B-spline curve, whose position is known up to an (varying) error ϵ_i . The boundary of all the paths defined by these disks $(\mathbf{b}_i, \epsilon_i) \in \mathbb{R}^{2,1}$ is precisely the cyclographic preimage of the curve in $\mathbb{R}^{2,1}$, see also [27], [48] and [11] and Fig. 1.2, *left*.

1.2.2 Gently inclined roads

As we will see in chapter 4, we solve the problem of finding envelopes of circles by optimizing a curve that is constrained to be inclined only a certain angle with the xy -plane. We assume

this angle to be $\frac{\pi}{4}$, but actually it could be any predefined value. Note also that roads need a sloping foundation, and that this slope is a constant of the underlying material. This application was inspired by [33], which can be seen as the special case with zero inclination - c.f. Fig. 1.2, *middle*.

1.2.3 Medial axis transform

The medial axis m of a planar domain D is the set of the centers of the maximal inscribed discs, see [7], [11] and [38]. If the boundary ∂D of D is well-behaved, we can search for a space curve \tilde{m} , whose cyclographic image is ∂D and then the (trimmed) projection of \tilde{m} is m , see also Fig. 1.2, *right*. For a discrete set of circles, finding the envelope thus also gives the medial axis as a by-product.

1.2.4 Offset curve approximation for varying distance

Computing the offset of a curve can be seen as intersecting certain developable surfaces, see sec. 2.3.1, which are already given if cyclographic images are used, thus the computation of offsets of an envelope of circles is trivial. If the curve has to be approximated, different methods can be used, see [26] and the references cited therein.

A slight change of view brings the following: Assume a given curve \mathbf{c} and one wants to compute an offset where the distance function d is not constant, but varies. The computation of \mathbf{c}_d is then the same as computing the envelopes of circles with centers on \mathbf{c} and radii given by d .

1.3 Thesis overview

We use topics and techniques in this thesis from three different fields of mathematics: sphere-geometry, CAGD and optimization. For all terms used in later chapters, we will give explanations and point to further literature, but we believe that this thesis is self-contained and can be read without consulting other books.

Circles/spheres can be represented by points

Chapter 2 introduces *cyclography* or *Laguerre geometry*, which is a field in non-Euclidean geometry that has a long tradition in Vienna, see Blaschke [4]. Basically, (oriented) circles (spheres in higher dimensions) are represented as points in *Minkowski space* $\mathbb{R}^{2,1}$, curves between them are then envelopes of circles in \mathbb{R}^2 . As we will see, the tangents of these curves in $\mathbb{R}^{2,1}$ have to fulfill certain constraints in order to give real envelopes.

Points are connected by curves/surfaces ...

Computer aided geometric design or *CAGD* is addressed in chapter 3; it subsumes the theory of freeform curves and surfaces (see Farin [10] or Piegl and Tiller [34]) and introduces methods to interpolate and approximate points by continuous objects, which in turn depend on control points.

Envelopes of circles

As it turns out, the task of finding envelopes of circles (spheres) is a constrained, quadratic, non-convex problem in $\mathbb{R}^{2,1}$ ($\mathbb{R}^{3,1}$) and approximation by general curves/surfaces can be solved by primal-dual interior point optimization algorithms. The reader familiar with these fields can go directly to chapter 4, where the findings of this thesis for curves in $\mathbb{R}^{2,1}$ are presented. We discuss previous work in detail, point to the key challenges, that are the non-convexity of Minkowski space and the resulting non-convex quadratic constraints and show how to overcome them. Special attention is paid to the constraints of the optimization, ways to linearize them, initial positions, as well as questions of numerics and stability.

Envelopes of spheres

Chapter 5 extends the optimization framework to dimension $\mathbb{R}^{3,1}$, considering 1-parameter families of spheres, which lead to canal surfaces, and 1-parameter families, which give rise to general envelope surfaces. We will see that the constraint of having Euclidean tangent planes requires special attention as $\mathbb{R}^{3,1} \setminus \Gamma$ (the Minkowski space without the convex Lorentz cone) is not a vector space.

Images and algorithm analysis

Finally, chapter 6 shows and discusses examples for curves and surfaces in $\mathbb{R}^{2,1}$ and $\mathbb{R}^{3,1}$ and compares methods, as well as numerical convergence.

2 Minkowski space

The four-dimensional real vector space with the signed metric $(1, 1, 1, -1)$ originates from (at least) two sources: It appears as the setting of Special Relativity and is there called *Minkowski space* $\mathbb{R}^{3,1}$ and will be described in section 2.1; its transformations are called *Lorentz transformations*.

Independently, this vector space appears as a point model space for spheres in the setting of *Laguerre geometry*, is usually called \mathbb{R}_1^4 and will be explained in section 2.2; we refer to its transformations as *Laguerre transformations*.

Since Minkowski geometry can be generalized (for General Relativity) to *Lorentz geometry* and is much wider known, we will use this notation throughout this work.

Lorentz and Laguerre transformations are identical up to translations, but when acting on oriented circles and lines, resp. oriented planes and spheres, we will speak of Laguerre transformations, see sec. 2.2.3 for details.

2.1 Basics of Minkowski geometry

For references to Special Relativity, consult the book [5], our notation will follow Jüttler's works [22] and [23].

2.1.1 Lorentz inner product

An *inner product* is a map on the product space $\langle \cdot, \cdot \rangle : V \times V \rightarrow F$ of a vector space V over its scalar field F , such that the following properties hold

1. *conjugate symmetry* $\langle \mathbf{x}, \mathbf{y} \rangle = \overline{\langle \mathbf{y}, \mathbf{x} \rangle}$
2. *linearity* in the first argument $\langle a \cdot \mathbf{x}, \mathbf{y} \rangle = a \cdot \langle \mathbf{x}, \mathbf{y} \rangle$
 $\langle \mathbf{x} + \mathbf{y}, \mathbf{z} \rangle = \langle \mathbf{x}, \mathbf{z} \rangle + \langle \mathbf{y}, \mathbf{z} \rangle$
3. *positive definiteness* $\langle \mathbf{x}, \mathbf{x} \rangle \geq 0$ with equality only for $\mathbf{x} = o$

If the third condition is replaced by

3. *nondegeneracy* if $\langle \mathbf{x}, \mathbf{y} \rangle = 0 \forall \mathbf{y} \in V$ then $\mathbf{x} = o$

we call the map $\langle \cdot, \cdot \rangle_L$ an *indefinite inner product*. Note that positive definiteness implies nondegeneracy. A prominent example is the *Lorentz* or *pseudo-Euclidean inner product*, which we define as

$$\langle \mathbf{a}, \mathbf{b} \rangle_L := \mathbf{a}^T \cdot \underbrace{\begin{pmatrix} 1 & & & \\ & \ddots & & \\ & & 1 & \\ & & & -1 \end{pmatrix}}_{E_{pe}} \cdot \mathbf{b} \text{ for } \mathbf{a}, \mathbf{b} \in \mathbb{R}^n \quad (2.1)$$

the subindex *pe* stands for *pseudo-Euclidean* and E_{pe} is sometimes called *Minkowski tensor*. This symmetric bilinear form has a *signature* $(n - 1, 1)$ (=eigenvalues of the Minkowski tensor) and is thus a *Lorentzian metric*. A real four-dimensional vector space with inner product (2.1) shall be called four-dimensional *Minkowski space* and denoted $\mathbb{R}^{3,1}$. Analogously we define a real three-dimensional Minkowski space and write $\mathbb{R}^{2,1}$. Sometimes, the wider class of signed metric spaces are called *pseudo-Euclidean spaces*, but we will use these terms synonymously.

Minkowski-orthogonal and cross product

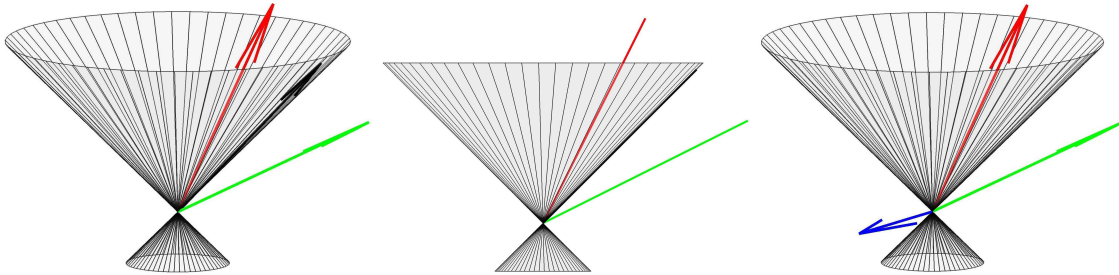


Figure 2.1: *left*: Three types of lines - Euclidean (*green*), isotropic (*black*) and pseudo-Euclidean (*red*). The red and green lines are Minkowski-orthogonal, *middle*: front view; note that Minkowski-orthogonal lines are conjugate w.r.t the Lorentz cone and that isotropic lines are orthogonal to themselves, *right*: the *blue* line is the cross product in $\mathbb{R}^{2,1}$ of the orthogonal lines *red*, *green*, thus the three form a pseudo-Euclidean basis

Two vectors \mathbf{a}, \mathbf{b} are called *Minkowski-orthogonal* if $\langle \mathbf{a}, \mathbf{b} \rangle_L = 0$. Clearly, an isotropic line (Fig. 2.4) is orthogonal to itself, see Fig. 2.1. The *pseudo-Euclidean normal* of a plane $\epsilon : ax + by + cz = 0$ has thus coordinates $\mathbf{n} = (a, b, -c)$ (see [22]). The cross-product in Minkowski is also slightly different from the Euclidean case; for two vectors $\mathbf{u} = (u_1, u_2, u_3)$ and $\mathbf{v} = (v_1, v_2, v_3)$ it is given as

$$\mathbf{w} = \mathbf{u} \times_L \mathbf{v} = (u_2v_3 - u_3v_2, u_3v_1 - u_1v_3, -u_1v_2 + u_2v_1)$$

Pseudo-Euclidean basis

A set of vectors $\{\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3 | \mathbf{b}_i \in \mathbb{R}^{2,1}\}$ satisfying

$$\langle \mathbf{b}_1, \mathbf{b}_1 \rangle_L = \langle \mathbf{b}_2, \mathbf{b}_2 \rangle_L = 1, \quad \langle \mathbf{b}_3, \mathbf{b}_3 \rangle_L = -1 \quad \text{and} \quad \langle \mathbf{b}_i, \mathbf{b}_j \rangle_L = 0, \quad i \neq j \quad (2.2)$$

is called *pseudo-Euclidean basis* of the vector space $\mathbb{R}^{2,1}$, see Fig. 2.1, *right*.

Minkowski norm

The Lorentz inner product (2.1) induces a *Minkowski norm*

$$\|\mathbf{x}\|_L = \sqrt{|\langle \mathbf{x}, \mathbf{x} \rangle_L|}, \quad (2.3)$$

which is not a norm in the regular sense, because it is not subadditive, e.g. the triangle inequality does not (always) hold, but its opposite is true

$$\|\mathbf{x} + \mathbf{y}\|_L \geq \|\mathbf{x}\|_L + \|\mathbf{y}\|_L,$$

if the last component (that causes the degeneracy) is > 0 . According to [5], this is also the reason for the famous *twin paradox*, which is not actually a paradox, but very counter-intuitive.

2.1.2 Types of vectors

Through (2.1) we can define three different kinds of vectors (see Figures 2.1 and 2.4):

1. $\langle \mathbf{a}, \mathbf{a} \rangle_L > 0 \Leftrightarrow \mathbf{a}$ is *space-like* or *Euclidean*
2. $\langle \mathbf{a}, \mathbf{a} \rangle_L = 0 \Leftrightarrow \mathbf{a}$ is *light-like* or *isotropic*
3. $\langle \mathbf{a}, \mathbf{a} \rangle_L < 0 \Leftrightarrow \mathbf{a}$ is *time-like* or *pseudo-Euclidean*

If two vectors \mathbf{a}, \mathbf{b} are Minkowski-orthogonal and \mathbf{a} is space-like, then \mathbf{b} must be time-like (and vice versa). If \mathbf{a} is light-like, so is its orthogonal vector.

Hyperplanes

The extension of the classification of vectors from sec. 2.1.2 to *hyperplanes* ϵ in $\mathbb{R}^{n-1,1}$ is straightforward, if we denote its normal vector by \mathbf{n}_ϵ

1. $\langle \mathbf{n}_\epsilon, \mathbf{n}_\epsilon \rangle_L < 0 \Leftrightarrow \epsilon$ is *space-like* or *Euclidean*
2. $\langle \mathbf{n}_\epsilon, \mathbf{n}_\epsilon \rangle_L = 0 \Leftrightarrow \epsilon$ is *light-like* or *isotropic*
3. $\langle \mathbf{n}_\epsilon, \mathbf{n}_\epsilon \rangle_L > 0 \Leftrightarrow \epsilon$ is *time-like* or *pseudo-Euclidean*

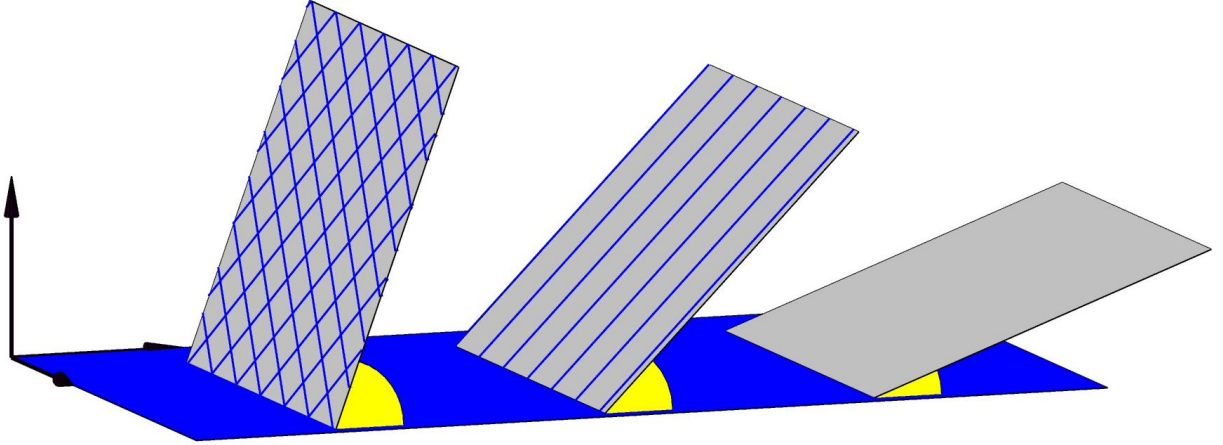


Figure 2.2: Three different types of planes in $\mathbb{R}^{2,1}$: *left*: pseudo-Euclidean, i.e. angle $> \frac{\pi}{4}$ (yellow) and two families of isotropic lines (blue), *middle*: isotropic plane, i.e. angle $= \frac{\pi}{4}$ carrying one family of isotropic lines, *right*: Euclidean, i.e. angle $< \frac{\pi}{4}$. The blue lines are isotropic. The same classification is true for hyperplanes in $\mathbb{R}^{3,1}$.

In general, a vector or hyperplane is

1. space-like \Leftrightarrow its angle with the plane $x_n = 0$ is less than $\frac{\pi}{4}$
2. light-like \Leftrightarrow its angle with the plane $x_n = 0$ is equal to $\frac{\pi}{4}$
3. time-like \Leftrightarrow its angle with the plane $x_n = 0$ is more than $\frac{\pi}{4}$

The connections between hyperplanes and the kind of lines they carry is answered in the following two statements:

Proposition 1. • Assume a hyperplane is Euclidean \Leftrightarrow its normal is pseudo-Euclidean. Then it contains only Euclidean lines.

- Assume a hyperplane is isotropic \Leftrightarrow its normal is isotropic. Then it contains Euclidean and isotropic lines.
- Assume a hyperplane is pseudo-Euclidean \Leftrightarrow its normal is Euclidean. Then it contains Euclidean, isotropic and pseudo-Euclidean lines.

Corollary 1. • A pseudo-Euclidean line can only lie in a pseudo-Euclidean hyperplane.

- A Euclidean hyperplane only contains Euclidean lines.

Following the paper [35], we state that if all objects in $\mathbb{R}^{3,1}$ under consideration lie in a Euclidean hyperplane H , we can map H via a Laguerre transformation (see sec. 2.1.4) to the plane $x_4 = 0$, i.e. envision H as Euclidean \mathbb{R}^3 .

2.1.3 Lorentz cone Γ

When describing different vectors in Minkowski space, see especially Fig. 2.1, we encountered the *Lorentz cone* Γ , which we want to define rigorously in this section.

Definitions

We call a set $C \subseteq \mathbb{R}^d$ *convex* if

$$(1 - \lambda)x + \lambda y \in C \text{ for } x, y \in C, 0 \leq \lambda \leq 1$$

A function $f : C \rightarrow \mathbb{R}$ is called *convex* if C is convex and

$$f((1 - \lambda)x + \lambda y) \leq (1 - \lambda)f(x) + \lambda f(y) \text{ for } x, y \in C, 0 \leq \lambda \leq 1 \quad (2.4)$$

A function is *strictly convex* if we substitute \leq by $<$ in the above equation.

Norms

For a vector space V over K , a mapping $\|\cdot\| : V \rightarrow \mathbb{R}$ is called a *norm*, if

1. $\|\mathbf{v}\| \geq 0 \forall \mathbf{v} \in V$ and $\|\mathbf{v}\| = 0$ if and only if $\mathbf{v} = 0$
2. $\|\alpha\mathbf{v}\| = |\alpha|\|\mathbf{v}\| \forall \alpha \in K, \forall \mathbf{v} \in V$ (positive homogeneity)
3. $\|\mathbf{v} + \mathbf{w}\| \leq \|\mathbf{v}\| + \|\mathbf{w}\|, \forall \mathbf{v}, \mathbf{w} \in V$ (triangle inequality)

Properties 2. and 3. imply that every norm is a convex function.

Let $C \subseteq \mathbb{R}^d$ be a set and $f : C \rightarrow \mathbb{R}$. The *epigraph* of f is the set

$$\text{epi}(f) = \{(x, t) \in \mathbb{R}^d \times \mathbb{R} : x \in C, t \geq f(x)\} \subseteq \mathbb{R}^d \times \mathbb{R} \cong \mathbb{R}^{d+1}$$

It can be shown, that f being a convex function is equivalent to $\text{epi}(f)$ being a convex set (cf. [14]). We conclude that the epigraph of any norm is a convex set, so if

$$\|(x_1, x_2)\|_2 = \sqrt{x_1^2 + x_2^2} \Rightarrow \text{epi}(\|\cdot\|) = \{(x_1, x_2, x_3) \in \mathbb{R}^2 \times \mathbb{R} : x_3^2 \geq x_1^2 + x_2^2\} \quad (2.5)$$

We call $\text{epi}(\|\cdot\|)$ the *Lorentz cone* and denote it by Γ , see also section 2.1. Clearly, the complement of $\text{epi}(\|\cdot\|)$ is not convex and thus any function defined on it cannot be convex by Definition (2.4). Strictly speaking, (2.5) defines Γ in $\mathbb{R}^{2,1}$, but it should be obvious how to carry this definition over to $\mathbb{R}^{n-1,1}$. Henceforth, we will not bother denoting the dimension and always assume the correct Γ to be used.

We conclude our definitions with an important theorem:

Lemma 1. *Let C be open and $f : C \rightarrow \mathbb{R}$ of class \mathcal{C}^2 . Then the following are equivalent:*

1. f is convex.
2. For any $x \in X$ the Hessian of f is positive semi-definite.

Proof: see [14].

2.1.4 Lorentz transformations

We follow the outline of [22]: A linear transform $A : \mathbb{R}^{n-1,1} \rightarrow \mathbb{R}^{n-1,1}$ is called *Lorentz transform* if it preserves the Minkowski inner product 2.1, i.e.

$$\langle \mathbf{A}\mathbf{u}, \mathbf{A}\mathbf{v} \rangle_L = \langle \mathbf{u}, \mathbf{v} \rangle_L \quad \forall \mathbf{u}, \mathbf{v} \in \mathbb{R}^{n-1,1}. \quad (2.6)$$

The group of all Lorentz transforms is the (homogeneous) *Lorentz group* $O(n-1, 1)$, which is the indefinite orthogonal group of signature $(n-1, 1)$.

2.2 Cyclography

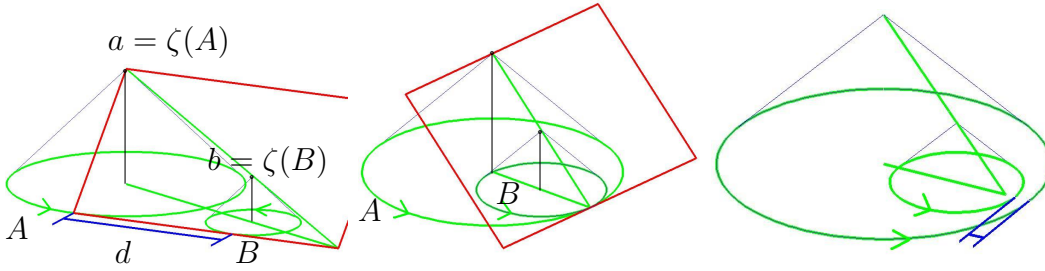


Figure 2.3: Distances of two circles, *from left to right*: positive, zero, negative

Minkowski space as introduced in section 2.1 can also be seen as a point model space for the space of circles (sec. 2.2.1) and spheres (sec. 2.2.2). Among the three classical circle and sphere geometries named after A.F. Möbius, E. Laguerre and S. Lie, which are discussed in detail in the books by Blaschke [4] and Müller & Krames [28], we want to focus on *Laguerre geometry*.

2.2.1 Laguerre geometry in \mathbb{R}^2

The fundamental objects of Laguerre geometry in Euclidean \mathbb{R}^2 are oriented lines and oriented circles, the latter are also called *cycles*. If a circle is traversed counter-clockwise, we assume the radius to be positive. We define the *cyclographic mapping* ζ to be a one-to-one correspondence between the set of cycles in \mathbb{R}^2 and the points $\{\mathbf{x}\}$ of the three-dimensional Minkowski space $\mathbb{R}^{2,1}$ defined in sec. 2.1.1, such that the first two coordinates of \mathbf{x} are a cycles' midpoint and the third coordinate is its radius, so $\zeta : \mathbb{R}^2 \rightarrow \mathbb{R}^{2,1}$.

This *cyclographic model* of Laguerre geometry is a classical *point space model*, where each

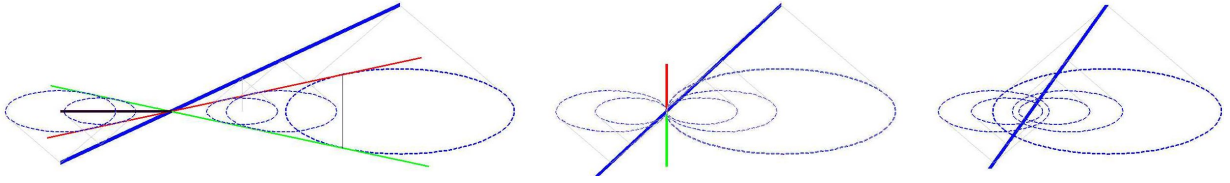


Figure 2.4: Cyclographic images of three different types of lines. *left*: an Euclidean line \mathbf{l}_E (blue) maps to two oriented lines (green and red) which are tangent to all the cones representing \mathbf{l}_E , *middle*: an isotropic line \mathbf{l}_i maps to two identical lines and *right*: a pseudo-Euclidean line \mathbf{l}_{pe} does not have a real image under the cyclographic mapping

geometric object is assigned a point in some space. The fact that $\mathbb{R}^{2,1}$ is also a well-studied object in Special Relativity is a lucky coincidence.

One can also envision the cyclographic mapping by embedding \mathbb{R}^2 into $\mathbb{R}^{2,1}$ (set third coordinate to zero) and erecting a right circular cone through each cycle \mathbf{c} . Since by construction the height of this cone equals \mathbf{c} 's radius, the cone's apex is $\zeta(\mathbf{c})$. Note that the axis of such a cone is parallel to the z -axis and thus all information is contained in the apex.

2.2.2 Laguerre geometry in \mathbb{R}^3

The definition of the *cyclographic mapping* from the set of oriented spheres and oriented planes in \mathbb{R}^3 to points in four-dimensional Minkowski space $\mathbb{R}^{3,1}$ is analogously to sec. 2.2.1. As suggested in [35] and already pointed out in sec. 2.1.2, one can also consider hyperplanar sections of $\mathbb{R}^{3,1}$ and identify them with Euclidean \mathbb{R}^3 , isotropic \mathbb{I}^3 or pseudo-Euclidean $\mathbb{R}^{2,1}$, depending on the hyperplane H being Euclidean, isotropic or pseudo-Euclidean.

2.2.3 Laguerre transformations

A *pseudo-Euclidean* (pe) *similarity* is an affine mapping of the form

$$\mathbf{x}' = \mathbf{a} + \lambda \mathbf{A} \cdot \mathbf{x} \quad (2.7)$$

with a constant $\lambda \neq 0$ and a pseudo-Euclidean orthogonal matrix \mathbf{A} that preserves the Lorentz inner product. Note that the condition (2.6) is equivalent to

$$\mathbf{A}^T \cdot \mathbf{E}_{pe} \cdot \mathbf{A} = \mathbf{E}_{pe} \quad (2.8)$$

where \mathbf{E}_{pe} is the diagonal matrix defined in (2.1). One could also define pe congruences to be the group of affine maps that preserve pe bases (2.2). For $|\lambda| = 1$ we call (2.7) a *pe congruence*.

Obviously, the pe similarities (2.7) are not homogeneous transformations, i.e. the origin is not fixed. Therefore, pe congruences are the same as Lorentz transformations up to the choice of an origin. When mapping a difference vector of two points, they are the same anyways, because then the translational component (a in (2.7)) cancels out. Blaschke [4], p.141 and p.270 defines Lorentz transformations through (2.7) and (2.8).

If we map the points which are acted upon by pe similarities via the cyclographic image to \mathbb{R}^{n-1} , we want to speak of *Laguerre transformations* and keep in mind that the latter also include translations. Thus, Lorentz and Laguerre transformations induced by pe similarities are identical up to translations, but when acting on oriented circles / spheres and oriented lines / planes, we will speak of Laguerre transformations henceforth. Strictly speaking, the set of oriented spheres and the set of oriented planes are disjoint and stay disjoint under Laguerre transformations, so to treat them rigorously, one would have to define them on each of these sets, but we will leave such proofs to the book [4].

Note that a Laguerre transformation is a (bijective) map on the set of spheres and planes respectively in \mathbb{R}^{n-1} and as such preserves oriented contact and noncontact between oriented spheres and hyperplanes, but does not preserve points (they might map to spheres). A oriented hyperplane h can be interpreted as the set of all oriented spheres that are in oriented contact with h , and in general this is true for oriented hypersurfaces.

Properties of Laguerre transformations

- oriented circles/spheres map to oriented circles/spheres (always including points as spheres with radius zero)
- oriented lines/hyperplanes map to oriented lines/hyperplanes
- oriented contact between circles/circles and circles/lines is an invariant property (projective map)
- pseudo-Euclidean lines map to pseudo-Euclidean lines (analogously isotropic to isotropic and Euclidean to Euclidean), so the type of lines (and hyperplanes) stays invariant.

2.2.4 Offsets and the connection to Minkowski sums

For some surface $\mathbf{h} \in \mathbb{R}^3$, a simple example of a pe similarity γ in $\mathbb{R}^{3,1}$ is a *dilatation* of $\zeta(\mathbf{h})$, i.e. adding a constant $a \neq 0$ to the 4th component of $\zeta(\mathbf{h})$, which means increasing the radii of oriented spheres in \mathbb{R}^3 , that envelope some surface \mathbf{h} , see also section 2.4 on surfaces. Thus $\gamma(\mathbf{h})$ is an *offset* of \mathbf{h} . Note that we would have received the same result by taking the *Minkowski sum* (see e.g. Gruber [14]) of \mathbf{h} and the sphere with radius a .

2.2.5 Distances

The circles A, B corresponding to two points $\mathbf{a} = \zeta(A)$, $\mathbf{b} = \zeta(B)$ on an isotropic line are tangent, see Fig. 2.3. If \mathbf{a}, \mathbf{b} lie on a Euclidean line, $d = \sqrt{\langle \mathbf{a} - \mathbf{b}, \mathbf{a} - \mathbf{b} \rangle_L}$ is the *tangential distance* of the two circles.

Notice that the tangential distance of two circles is not transitive. If $\text{pe-dist}(A, B) = 0$ and $\text{pe-dist}(B, C) = 0$ it does NOT follow that $\text{pe-dist}(A, C) = 0$. That is why [40] introduce a new metric.

2.3 Curves in Minkowski space

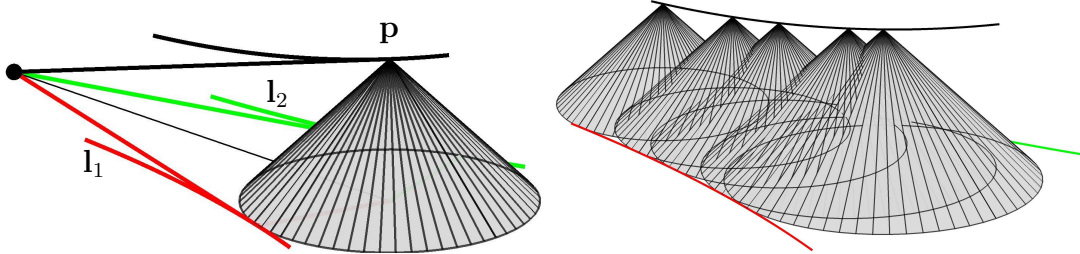


Figure 2.5: Cyclographic image $\mathbf{I}_{1,2}$ (red and green) of a curve \mathbf{p} (black). *left*: The tangents of $\mathbf{I}_{1,2}$ and \mathbf{p} all meet in a point, which is the main idea behind the construction (2.12), *right*: the curve \mathbf{p} and more cones representing the cyclographic mapping.

Now that we know the fundamentals of Minkowski space and its transformations, we will introduce its curves and their images in sections 2.3.2 and 2.3.3. But before looking at cyclographic images of curves in $\mathbb{R}^{2,1}$, we state some well-known facts about planar curves in sec. 2.3.1, see e.g. [8].

2.3.1 Offsets and distance function

For a parametrized curve \mathbf{c} the *offset curve* or *parallel curve* is given by

$$\mathbf{c}_d(t) = \mathbf{c}(t) + d \cdot \frac{\mathbf{n}(t)}{\|\mathbf{n}(t)\|}$$

where d is fixed and $\mathbf{n} = (-\dot{\mathbf{c}}_2, \dot{\mathbf{c}}_1)$ denotes an oriented normal. The parallel curve can have cusps, which occur whenever \mathbf{c}_d intersects the *evolute* of \mathbf{c} , defined by

$$\mathbf{c}^* = \mathbf{c} + \mathbf{n} \frac{\dot{\mathbf{c}} \cdot \dot{\mathbf{c}}}{\det(\dot{\mathbf{c}}, \ddot{\mathbf{c}})}, \quad (2.9)$$

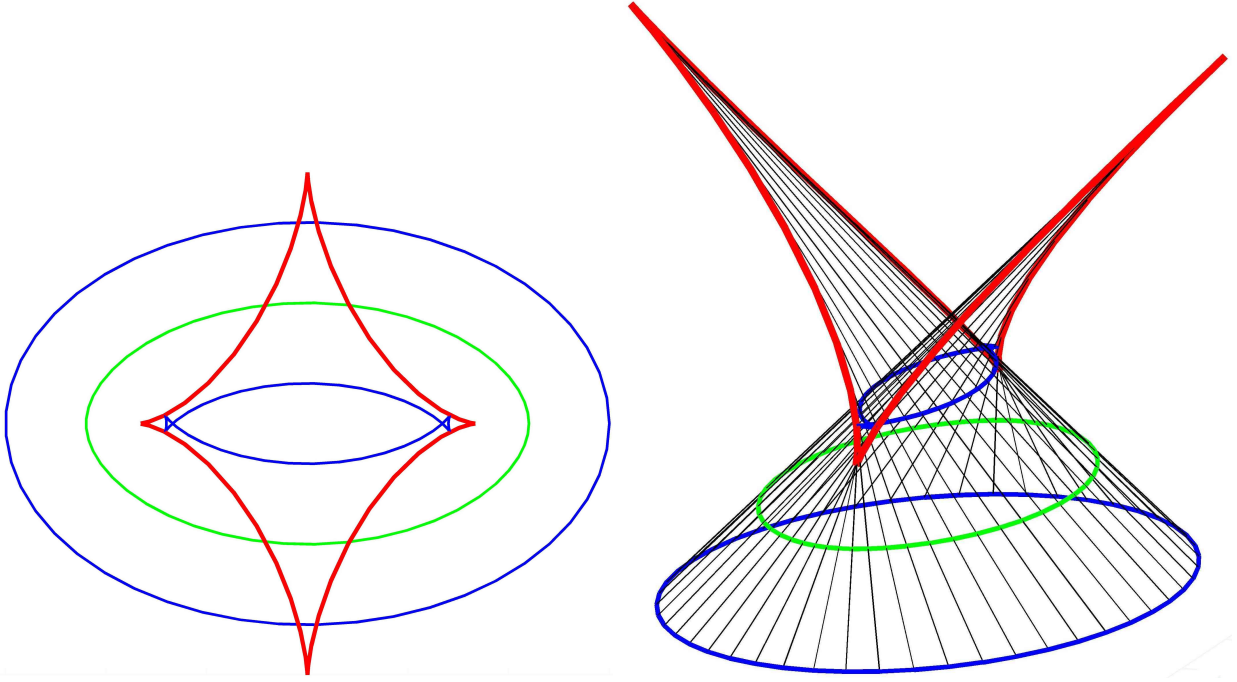


Figure 2.6: *left*: a curve (*green*) and two offset curves (*blue*). The inner offset curve has the cusps according to the intersection with the evolute \mathbf{c}^* (*red*). *Right*: a developable surface of constant slope $D_{\mathbf{c}}$ (*black*) as a model of the signed distance function to the curve (*green*). The evolute is the projection of the curve of regression $\beta(t)$ (*red*) of $D_{\mathbf{c}}$.

see Fig. 2.6. This can be explained as follows (see also [41]): The parallel curves of a planar curve are the level sets of its *distance function*.

If we assume the distance function to have a sign, e.g. the distance is negative on the right side of a curve (for simple closed curve like an ellipse, that's "outside") we can model it as a developable surface $D_{\mathbf{c}}$ of constant slope with the curve's plane π :

$$D_{\mathbf{c}}(t, v) = \mathbf{c}(t) + v \cdot \underbrace{\frac{1}{\sqrt{2}} \left(\frac{-\dot{c}_2}{\|\mathbf{n}\|}, \frac{\dot{c}_1}{\|\mathbf{n}\|}, 1 \right)}_{:=\mathbf{w}} \quad (2.10)$$

$D_{\mathbf{c}}$ has a curve of regression $\beta(t)$ for $v = -\frac{\dot{\mathbf{c}} \cdot \dot{\mathbf{w}}}{\dot{\mathbf{w}} \cdot \dot{\mathbf{w}}}$, which if projected to π , is the evolute of \mathbf{c} (and thus of \mathbf{c}_d , cf.(2.9) and Fig. 2.6). The parallel curves \mathbf{c}_d are planar intersections of $D_{\mathbf{c}}$, β is its singular curve, which explains why and where the singularities of the parallel curves appear.

An idea on how to trim the parallel curve's self-intersections was presented in [33].

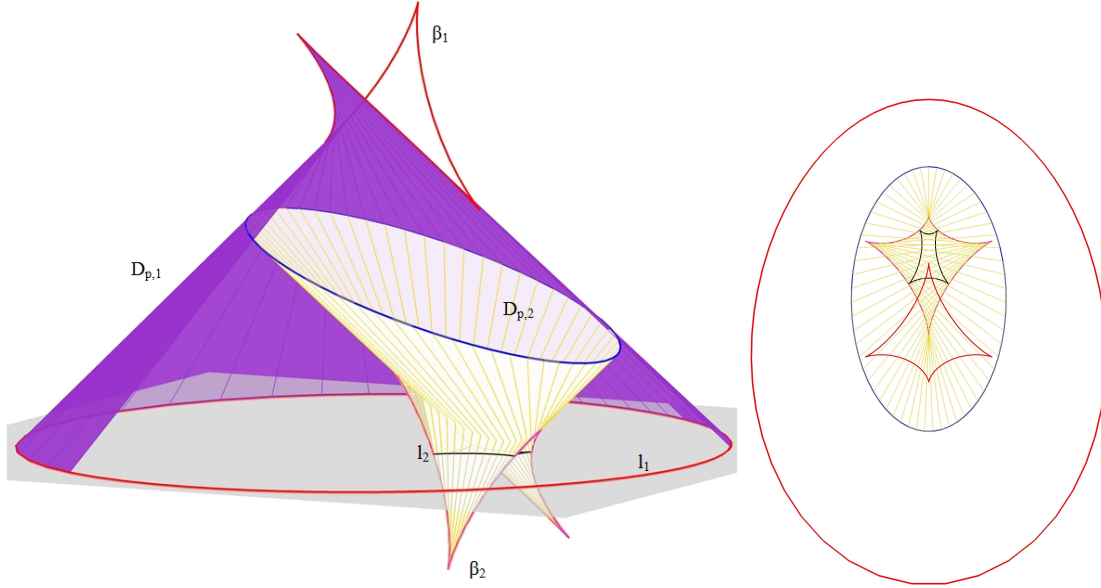


Figure 2.7: The two developable surfaces of constant slope $D_{\mathbf{p}}$ (magenta and yellow) through a curve \mathbf{p} (blue) intersect the xy -plane in two curves $\mathbf{l}_{1,2}$ (red and black)

2.3.2 Cyclographic image of curves in $\mathbb{R}^{2,1}$

Let \mathbf{p} be a C^2 curve in $\mathbb{R}^{2,1}$ with parametrization $\mathbf{p}(t) = (p_1, p_2, p_3)(t)$, $t \in I$ and $\dot{\mathbf{p}}(t) \neq 0$ in I . The inverse cyclographic image $\zeta^{-1}(\mathbf{p})$ of \mathbf{p} is the envelope of the family of circles

$$\zeta^{-1}(\mathbf{p}) : (x - p_1(t))^2 + (y - p_2(t))^2 - (z - p_3(t))^2 = 0 \quad (2.11)$$

which is parametrized as (see Fig. 2.5)

$$\mathbf{l}_{1,2} = \begin{pmatrix} p_1 \\ p_2 \end{pmatrix} - \frac{p_3 \dot{p}_3}{\dot{p}_1^2 + \dot{p}_2^2} \begin{pmatrix} \dot{p}_1 \\ \dot{p}_2 \end{pmatrix} \pm \frac{p_3 \sqrt{\langle \dot{\mathbf{p}}, \dot{\mathbf{p}} \rangle_L}}{\dot{p}_1^2 + \dot{p}_2^2} \begin{pmatrix} -\dot{p}_2 \\ \dot{p}_1 \end{pmatrix} \quad (2.12)$$

These curves are real for $\langle \dot{\mathbf{p}}, \dot{\mathbf{p}} \rangle_L \geq 0$ and non-identical for $\langle \dot{\mathbf{p}}, \dot{\mathbf{p}} \rangle_L > 0$. The centers of the circles (2.11) are of course the projection of the curve \mathbf{p} onto \mathbb{R}^2 , which we denote by $\pi(\mathbf{p}(t)) = (p_1, p_2)$. It is also the *bisector* of the curves $\mathbf{l}_{1,2}$. Note that $\pi(\cdot)$ denotes a projection to the xy -plane, while in other cases π is used to describe the xy -plane itself. Hopefully, this ambiguity of notation will not lead to confusion.

Eq. (2.12) can be derived as follows: For each point on \mathbf{p} take the right circular cone. This family of cones is enveloped by two developable surfaces of constant slope $D_{\mathbf{p}}$. Their intersections with the xy -plane are the inverse cyclographic images $\mathbf{l}_{1,2}$ of \mathbf{p} , see Fig. 2.7.

If \mathbf{p} is entirely contained in a plane parallel to the xy -plane π with distance d , we are precisely in the situation described in section 2.3.1 and the cyclographic images are simply parallel curves of \mathbf{p} at distance d , see Fig. 2.6.

2.3.3 Cyclographic image of curves in $\mathbb{R}^{3,1}$

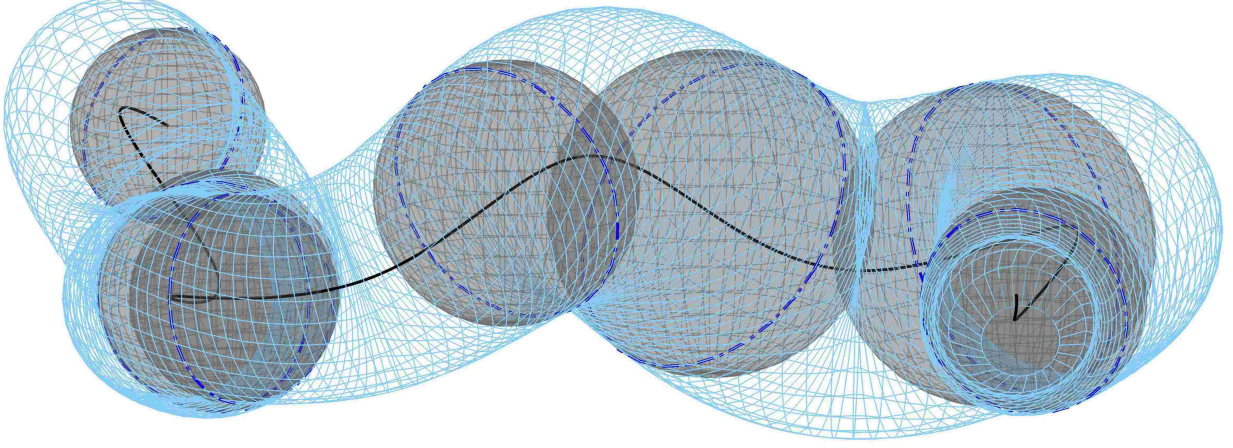


Figure 2.8: Canal surfaces as cyclographic image of a curve $\mathbf{c} \subset \mathbb{R}^{3,1}$: each point of \mathbf{c} corresponds to a sphere (*gray*) in \mathbb{R}^3 that touches the envelopes (*light blue*) along a circle (*blue*) with centers $N(t)$ on spine curve (*black*).

The cyclographic image of curve $\mathbf{c} \subset \mathbb{R}^{3,1}$ in \mathbb{R}^3 is the envelope of a one-parameter family of spheres, i.e. a *canal surface* (for constant radius, or equivalently constant c_4 , it is a *pipe surface*, see Fig. 2.8). The projection $\pi(\mathbf{c})$ is the *spine curve* of the canal surface. Special cases include: cones of revolution (image of a space-like line), cylinders of revolution (line with $c_4 = \text{const}$), Dupin cyclides (cyclographic image of a pe-circle with space-like tangents, see also [38])

In the smooth case, [40] shows how to derive a parametrization of a canal surface: Given a one-parameter family of spheres in \mathbb{R}^3 with centers $M(t) = (m_1, m_2, m_3)(t)$ and radii $r(t)$, or, equivalently, a curve $\mathbf{c} = (m_1, m_2, m_3, r)(t) \subseteq \mathbb{R}^{3,1}$, the envelope consists of a one-parameter family of circles with centers $N(t)$ and radii $r_1(t)$

$$N(t) = M(t) - \frac{r(t)\dot{r}(t)}{\|\dot{M}(t)\|^2} \dot{M}(t), \quad r_1(t) = \frac{r(t)}{\|\dot{M}(t)\|} \sqrt{\|\dot{M}(t)\|^2 - \dot{r}(t)^2}.$$

With the curve's Frenet frame

$$T(t) = \frac{\dot{M}(t)}{\|\dot{M}(t)\|}, \quad E(t) = \frac{\dot{T}}{\|\dot{T}\|} = \frac{\dot{M}(t) \times (\ddot{M}(t) \times \dot{M}(t))}{\|\dot{M}(t) \times (\ddot{M}(t) \times \dot{M}(t))\|} \quad \text{and} \quad F(t) = T(t) \times E(t)$$

the parametrization of the canal surfaces is given by

$$X(t, u) = N(t) + r_1(t) \cos(u)E(t) + r_1(t) \sin(u)F(t).$$

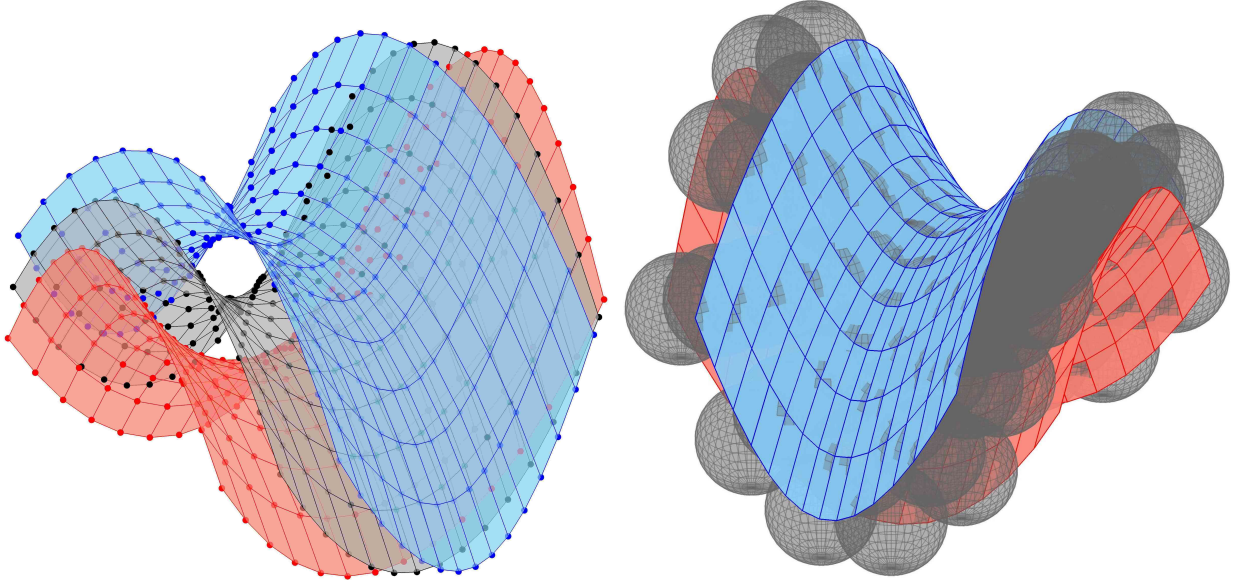


Figure 2.9: Inverse cyclographic image $\Phi \subset \mathbb{R}^3$ of a surface \mathbf{S} in Minkowski space $\mathbb{R}^{3,1}$ consists of two surfaces $\mathbf{L}_{1,2}(u, v)$ (red and blue). The projection $\mathbf{M} = \pi(\mathbf{S})$ is the middle surface (gray) on which the centers of the spheres lie.

2.4 Surfaces in Minkowski space

The inverse cyclographic image $\Phi \subset \mathbb{R}^3$ of a surface $\mathbf{S} = \mathbf{f}(u, v) \subset \mathbb{R}^{3,1}$, $(u, v) \in \mathbb{R}^2$ is the envelope $\mathbf{L} = \mathbf{L}_1 \cup \mathbf{L}_2$ of the spheres $f(u_0, v_0)$ for (u_0, v_0) fixed parameters. The centers of these spheres lie on the *bisector* (surface) \mathbf{M} , which we obtain from \mathbf{S} by projecting it via $x_4 = 0$ onto \mathbb{R}^3 . This is the same top view projection we have used in 2.3.2 and thus we write $\mathbf{M} = \pi(\mathbf{S})$. We see that \mathbf{M} is to \mathbf{L} what the medial axis is to the curves $\mathbf{l}_{1,2}$, and we call the single spheres of $\zeta^{-1}(\mathbf{S})$ the *middle spheres* of \mathbf{L} , see also Fig. 2.9. Thus, \mathbf{S} is the *medial axis transform* of $\mathbf{L}_1 \cup \mathbf{L}_2$.

Parametrization

A parametrization of the inverse cyclographic image of a surface Φ is straight forward: Intersecting the surface \mathbf{S} and its two partial derivatives for each parameter pair (u_0, v_0)

$$\begin{aligned} \mathbf{S} &: (x - \mathbf{M})^2 - r^2 = 0 \\ \mathbf{S}_u &: (x - \mathbf{M})^T \cdot \mathbf{M}_u + r r_u = 0 \\ \mathbf{S}_v &: (x - \mathbf{M})^T \cdot \mathbf{M}_v + r r_v = 0 \end{aligned}$$

yields two points $\mathbf{L}_{1,2}(u_0, v_0)$ which are precisely the contact points of the sphere $(\mathbf{M}_1, \mathbf{M}_2, \mathbf{M}_3, r)(u_0, v_0)$ with the envelope $\mathbf{L}_{1,2}(u_0, v_0)$, see also Fig. 2.9.

3 B-Splines

In this chapter we want to introduce the basics of Computer Aided Geometric Design (*CAGD*), as can be found in standard text books like Piegl & Tiller [34], Farin [10] or Hoschek & Lasser [18]. It is about fitting curves and surfaces to a given set of points, which can either be done by interpolation, see sec. 3.1.1, or approximation (sec. 3.1.2).

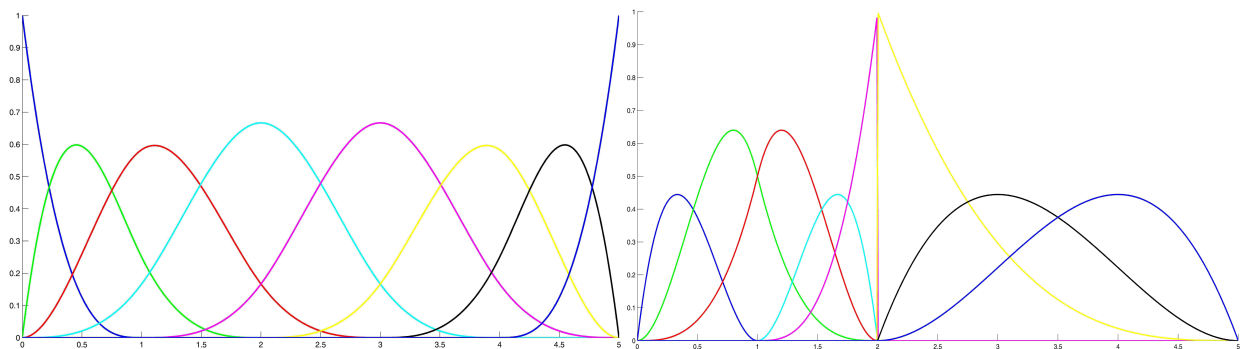


Figure 3.1: Examples for basis functions; *left*: cubic B-spline basis functions over the knot vector $\{0, 0, 0, 0, 1, 2, 3, 4, 5, 5, 5, 5\}$ (*uniform* knot spacing), *right*: basis functions over the knot vector $\{0, 0, 0, 1, 1, 2, 2, 2, 2, 5, 5, 5\}$

3.1 Interpolation and approximation

Let $U = \{u_0, \dots, u_m\}$ be a non-decreasing sequence of real numbers, the u_i are called *knots* and each knot's maximum multiplicity is $p + 1$, U is the *knot vector*. The i th B-spline *basis function* of degree p (order $p + 1$), denoted by $N_{i,p}$ (see Fig. 3.1) is defined recursively as

$$N_{i,0}(u) = \begin{cases} 1 & \text{if } u_i \leq u \leq u_{i+1} \\ 0 & \text{otherwise} \end{cases} \quad (3.1a)$$

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u), \quad (3.1b)$$

thus $N_{i,p}$ is piecewise polynomial. Note that fractions in (3.1b) can become $\frac{0}{0}$, in which case it is set 0. The *B-spline curve* is given as

$$\mathbf{c}(u) = \sum_{i=0}^n N_{i,p}(u)P_i \quad (3.2)$$

and the P_i are called *control points*.

Derivatives of B-Splines

The derivative of a basis function is given by

$$N'_{i,p} = \frac{p}{u_{i+p} - u_i} N_{i,p-1}(u) - \frac{p}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u).$$

3.1.1 Curve interpolation

Suppose a given set of points $\mathbf{Q} = \{Q_k\}, k = 0, \dots, n$ that have to be interpolated by a p th-degree B-Spline curve. We have to find

- parameter values \bar{u}_k for each Q_k
- a knot vector $U = \{u_0, \dots, u_m\}$.

For the \bar{u}_k , we choose *chord length parameters*, i.e. if we set

$$\begin{aligned} d &= \sum_{k=1}^n |Q_k - Q_{k-1}|, \text{ then } \bar{u}_0 = 0 \text{ and } \bar{u}_n = 1 \\ \bar{u}_k &= \bar{u}_{k-1} + \frac{|Q_k - Q_{k-1}|}{d}, \quad k = 1, \dots, n-1 \end{aligned} \quad (3.3)$$

The uniformly distributed knots can be found by an *averaging method*:

$$\begin{aligned} u_0 &= \dots = u_p = 0, \quad u_{m-p} = \dots = u_m = 1 \\ u_{j+p} &= \frac{j}{n-p+1}, \quad j = 1, \dots, n-p \end{aligned}$$

The interpolation for each \bar{u}_k becomes (3.2), i.e.

$$Q_k = \mathbf{c}(\bar{u}_k) = \sum_{i=0}^n N_{i,p}(\bar{u}_k)P_i, \quad k = 0, \dots, n$$

for every k . If we write all these equations at once, i.e.

$$\mathbf{Q} = \underbrace{\begin{bmatrix} N_{0,p}(\bar{u}_0) & \dots & N_{n,p}(\bar{u}_0) \\ \vdots & \ddots & \vdots \\ N_{0,p}(\bar{u}_n) & \dots & N_{n,p}(\bar{u}_n) \end{bmatrix}}_{\mathbf{N}} \cdot \mathbf{P}, \quad (3.4)$$

for each coordinate. Here \mathbf{Q} is the vector containing the points Q_k , \mathbf{P} is the vector of the control points and \mathbf{N} is called *collocation matrix*. Note that it is a sparse and banded matrix by (3.1).

The existence and uniqueness of the solution of (3.4) is given by the following, see [10]:

Lemma 2 (Schoenberg-Whitney Condition). *Let $\bar{u}_0 < \bar{u}_1 < \dots < \bar{u}_n$. For arbitrary Q_k there exists exactly one spline $\mathbf{c}(u)$ of degree p with knot vector $\{u_0, \dots, u_m\}$, such that $Q_k = \mathbf{c}(\bar{u}_k)$ if and only if $u_k \leq \bar{u}_k \leq u_{k+p+1}$ for $k = 0, \dots, n$ with equality only if the knot has multiplicity $p + 1$.*

3.1.2 Curve approximation

Interpolation has, among others, the disadvantage that the number of points m to be interpolated determines the number of control points $n + 1$ of the interpolation curve uniquely. Now we seek a p th-degree B-Spline curve $\mathbf{c}(u)$, $u \in [0, 1]$ with fixed knots, such that $Q_0 = \mathbf{c}(0)$, $Q_m = \mathbf{c}(1)$ and for the remaining Q_k the sum

$$\sum_{k=1}^{m-1} \text{dist}^2(\mathbf{c}(u), Q_k) + \lambda f_s \quad (3.5)$$

is minimized, where f_s is a *regularization term* (see sec. 3.3) and λ is a positive weight. The main challenge is how to define the distance $\text{dist}^2(\mathbf{c}(u), Q_k)$ between a curve and a point. The term $\text{dist}(\cdot, \cdot)$ is, in general, highly nonlinear and not even differentiable everywhere, see sec. 3.2.2. Usually (3.5) is solved iteratively with the help of a *footpoint* (see sec. 3.2), that is a point $\mathbf{c}(u_k)$ on the curve that is in some sense closest to Q_k . Thus

1. for each Q_k find a footpoint $\mathbf{c}_i(u_k)$, i is the iteration counter
2. approximate the squared distance function $\text{dist}^2(\mathbf{c}_i(u_k), Q_k)$
3. minimize the approximation error and generate a new curve $\mathbf{c}_{i+1}(u)$.
If the error is too large, repeat 1.

Since this algorithm utilizes two nested iterations (the outer one for updating the control points and an inner one for finding the footpoints), this is sometimes called a *variable-separation method*, see [2], which also gives examples for *global* algorithms, i.e. ones that do not have separated minimizations.

3.1.3 B-Spline surfaces

We will exclusively consider tensor product B-spline surfaces and leave out e.g. triangular patches. For parameters u and v and two appropriate knot vectors, we define a B-Spline

surface,

$$\mathbf{f}(u, v) = \sum_{k=0}^n \sum_{j=0}^m \mathbf{N}_k^{d_1}(v) \mathbf{N}_j^{d_2}(u) \mathbf{d}_{kj},$$

where \mathbf{d}_{kj} are the control points on a (topologically) regular grid and $\mathbf{N}_k^{d_1}(v), \mathbf{N}_j^{d_2}(u)$ are B-spline basis functions of degrees d_1, d_2 as in (3.1), which - after choosing parameters (u_k, v_k) - lead to the collocation matrices \mathbf{N}_u and \mathbf{N}_v .

We will skip *interpolation* with tensor product B-spline functions and look directly at *approximation*, e.g. a least-squares minimization like (3.5),

$$\sum_{k=1}^{m-1} \text{dist}^2(\mathbf{f}(\bar{u}_k, \bar{v}_k), Q_k) + \lambda f_s$$

which is very similar to the curve case as discussed in sec. 3.1.2. We will iteratively solve this non-linear optimization via a footpoint $\mathbf{f}(\bar{u}_k, \bar{v}_k)$ (see sec. 3.2) and an approximation to the distance function, as described in section 3.2.2. The collocation matrices in the surface case are much bigger and worse conditioned than for curves and thus regularization f_s (sec. 3.3) plays a bigger role. In fact, since we do not assume the data points to lie on a regular grid, as is done e.g. in [34], we will convolute the two collocation matrices into one (see especially sec. 5.4.1), following the example of Farin [10] and thus sometimes speak of *the* collocation matrix.

3.2 Measuring distances

This section is concerned with measuring the distance to a curve or a surface, both in theory and in algorithmic practice. We will only state the basic notation and point to literature for details.

3.2.1 Footpoint problem

If one wants to approximate points $\{Q_i\}$ with a curve $\mathbf{c}(u)$ and minimize the approximation error, each point Q_i is assigned a parameter value u^* , such that $\mathbf{c}(u^*)$, the *footpoint*, is the closest point on the curve, see Hoschek & Lasser [18] for details. In other words, one wants to find a value u^* , such that

$$g(u_i^*) = (\mathbf{c}(u_i^*) - Q_i)^2$$

is minimal.

Newton iteration

This minimization is carried out via Newton iteration $u^{i+1} = u^i + \Delta u^i$ and a second-order Taylor approximation of g in u :

$$\tilde{g}(u_i) = g(u_i) + \nabla g(u_i)^T (u - u_i) + \frac{1}{2} (u - u_i)^T \nabla^2 g(u_i) (u - u_i) + O(u^3).$$

Thus for $g(u_i)$ to be minimal, the gradient $\nabla \tilde{g}(u_i)$ has to be zero. The update can hence be given as

$$\Delta u^i = -\frac{\nabla g(u_i)}{\nabla^2 g(u_i)} = -\frac{(\mathbf{c}(u^i) - Q_i) \cdot \dot{\mathbf{c}}(u^i)}{\dot{\mathbf{c}}^2(u^i) + (\mathbf{c}(u^i) - Q_i) \cdot \ddot{\mathbf{c}}(u^i)}. \quad (3.6)$$

Geometrically, the line connecting Q_i and the footpoint $\mathbf{c}(u_i^*)$ is perpendicular to the curve, i.e. one minimizes the distance between the point Q_i and the curve's tangent (see Fig. 3.2, left).

A good initial value for this iteration can be found by (coarsely) sampling the curve and taking the nearest neighbor.

Footpoints on surfaces

Given a surface $\mathbf{f}(u, v)$ Hoschek & Lasser [18] give a formula for finding an update in the Newton iteration $(u^{i+1}, v^{i+1}) = (u^i, v^i) + (\Delta u^i, \Delta v^i)$, again using a second-order Taylor approximation similar to (3.6):

$$\begin{aligned} \Delta u^i &= -\frac{(Q_i - \mathbf{f}(u^i, v^i)) \cdot \mathbf{f}_u(u^i, v^i)}{(Q_i - \mathbf{f}(u^i, v^i)) \cdot \mathbf{f}_{uu}(u^i, v^i) - \mathbf{f}_u(u^i, v^i)^2} \\ \Delta v^i &= -\frac{(Q_i - \mathbf{f}(u^i, v^i)) \cdot \mathbf{f}_v(u^i, v^i)}{(Q_i - \mathbf{f}(u^i, v^i)) \cdot \mathbf{f}_{vv}(u^i, v^i) - \mathbf{f}_v(u^i, v^i)^2}, \end{aligned}$$

where \mathbf{f}_u is the partial derivative w.r.t. u and analogously the other derivatives. Another approach is shown in [19]. Similar to the curve case, an initial guess is found by sampling an initial surface position and taking nearest neighbors. The iteration is continued until some user-specified threshold is found.

3.2.2 Distance function

Given a curve $\mathbf{c}(u) : I \rightarrow \mathbb{R}^n$, the *distance function*

$$\text{dist}(x, \mathbf{c}(u)) = \min_{p \in \mathbf{C}(u)} \|p - x\| \quad (3.7)$$

assigns a value to each point $x \in \mathbb{R}^n$, i.e. the shortest distance from the point to the curve. Due to the absolute value in the definition of the distance function, it is not differentiable

along the curve $\mathbf{c}(u)$. This problem can be overcome by introducing a *signed distance function* $d(x)$, which is defined as the viscosity solution of the Eikonal equation, see [47]:

$$\nabla d^2 = 1 \text{ or } \|\nabla d\| = 1.$$

For given boundary data, which is, up to a sign, the distance function (3.7), and we will skip the term "signed" from now on. For a planar curve, the graph of the distance function is a torsal surface of constant slope $\frac{\pi}{4}$ against the plane, see [33]. The level sets of the distance function are its offsets and as we have seen in sec. 2.3.1, the the distance function has singular values along the evolute of the curve. Furthermore, it is singular along the medial axis of the curve, see sec. 1.2.3 for a definition.

For a surface, the distance function is not differentiable on its focal surfaces, see sec. 2.4, which includes spatial curves as a special case if seen as canal surfaces with radius zero.

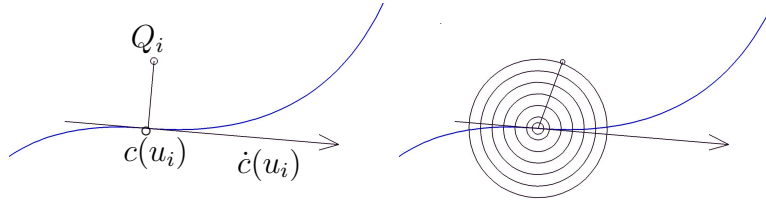


Figure 3.2: *left*: A footpoint is computed via a Newton iteration, such that the connecting line from $c(u_i)$ to the input point Q_i is perpendicular to the curve's tangent $\dot{c}(u_i)$, *right*: level sets of the point distance, i.e. the distance to a point is minimized.

Point distance minimization (PDM)

Point distance minimization first addressed in Hoschek [16], refined by Saux [43], proposes to compute the distance between a point Q_k and a point on the curve $\mathbf{c}(u_k)$, where $u_{k,i}$ is the parameter of the footpoint that is updated at iteration step i , see Fig. 3.2. If the collocation matrix at step i is \mathbf{N}_i , the vector containing all the $\{Q_k\}$ is \mathbf{Q} , then the displacement vector $\Delta\mathbf{P}_i$ is obtained by solving the linear system

$$\mathbf{N}_i^T \mathbf{N}_i \Delta\mathbf{P}_i = \mathbf{N}_i^T (\mathbf{Q} - \mathbf{N}_i \mathbf{P}_i). \quad (3.8)$$

3.3 Geometric regularization

In this section we want to focus on the shape of the solution of a system of equations. Numerically speaking, more equations are added to a given system, which also improves the condition number of a linear system.

3.3.1 Tikhonov regularization

For regularization of discrete ill-posed problems and their solution's implementation we refer to Hansen [15]. Assume we want to solve the linear least-squares problem

$$\min_{\mathbf{x}} \|\mathbf{Ax} - \mathbf{b}\|_2, \quad \mathbf{A} \in \mathbb{R}^{m \times n}, \quad m > n \quad (3.9)$$

where the matrix \mathbf{A} is ill-conditioned, which means the solution \mathbf{x} can not be trusted. If an initial estimate \mathbf{x}^* for the solution is given, we will ask for the constraint

$$\Omega(\mathbf{x}) = \|\mathbf{L}(\mathbf{x} - \mathbf{x}^*)\|_2 \quad (3.10a)$$

$$\text{otherwise } \Omega(\mathbf{x}) = \|\mathbf{Lx}\|_2 \quad (3.10b)$$

to be minimal, where \mathbf{L} is usually either the identity matrix \mathbf{I}_n ("standard form") or a discrete approximation of the derivative operator, see [15], given by

$$\mathbf{L}_1 = \begin{pmatrix} 1 & -1 & & & \\ & \ddots & \ddots & & \\ & & & 1 & -1 \end{pmatrix} \in \mathbb{R}^{(n-1) \times n}, \quad \mathbf{L}_2 = \begin{pmatrix} 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & & 1 & -2 & 1 \end{pmatrix} \in \mathbb{R}^{(n-2) \times n}. \quad (3.11)$$

We will mostly assume (3.10b), especially because we are using an iterative scheme.

As the objective function (3.9) and the constraint (3.10) might interfere, one introduces the Lagrange function

$$\mathbf{x}_\lambda = \operatorname{argmin}\{\|\mathbf{Ax} - \mathbf{b}\|_2^2 + \lambda^2 \|\mathbf{L}(\mathbf{x} - \mathbf{x}^*)\|_2^2\} \quad (3.12)$$

where the choice of the *regularization parameter* λ obviously has a big impact on the solution of the *Tikhonov regularized* solution \mathbf{x}_λ . As the solution of the quadratic problem (3.12) is straight-forward, this means that Tikhonov regularization of a possibly ill-posed linear least-squares problems means adding $\lambda \mathbf{L}^T \mathbf{L}$ to the normal equations

$$(\mathbf{A}^T \mathbf{A} + \lambda^2 \mathbf{L}^T \mathbf{L}) \mathbf{x} = \mathbf{A}^T \mathbf{b}, \quad (3.13)$$

so the solution is given via the *Tikhonov regularized inverse* $\mathbf{A}_\lambda^\dagger = (\mathbf{A}^T \mathbf{A} + \lambda^2 \mathbf{L}^T \mathbf{L})^{-1} \mathbf{A}^T$.

3.3.2 Curves

Constraints on the flexibility of the curve have to be used, i.e. *smoothing* and *regularization*, see [17]. Assume a curve $\mathbf{c}(t)$, we define the energy term

$$F_1 = \int \|\mathbf{c}'(t)\|^2 dt,$$

which minimizes the arc length.

Secondly the second derivate,

$$F_2 = \int \|\mathbf{c}''(t)\|^2 dt,$$

sometimes called *bending energy*, can be minimized, see also [50]. If the curve $\mathbf{c}(t)$ is parametrized by arc length (which is generally not the case for B-Splines), F_2 is a linearization of the curvature. Formulations of the above functionals in matrix notation were already given as eq. 3.11, which are used as practical implementations.

3.3.3 Surfaces

Regularization of surface approximation usually means adding an energy functional, see [18], like that for thin plate splines

$$F_s = \iint (\mathbf{f}_{uu})^2 + 2(\mathbf{f}_{uv})^2 + (\mathbf{f}_{vv})^2 du dv$$

We write this term into a matrix \mathbf{M}_{F_s} and apply it to eq. (3.8), so that we arrive at

$$(\mathbf{N}_i^T \mathbf{N}_i + \lambda \mathbf{M}_{F_s}) \mathbf{P}_{i+1} = \mathbf{N}_i^T \mathbf{Q} - (\mathbf{N}_i^T \mathbf{N}_i + \lambda \mathbf{M}_{F_s}) \mathbf{P}_i$$

for the point distance minimization, compare to the Tikhonov regularized inverse (3.13). The regularization parameter λ is user specified and changes during the iteration, see sec. 3.3.1.

4 Envelopes of Circles

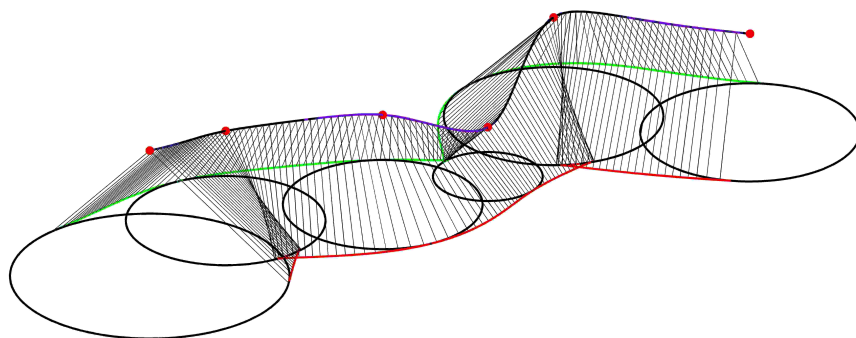


Figure 4.1: Given a set of circles in \mathbb{R}^2 , its envelope (curves in *red* and *green*) can be found as the inverse cyclographic image of an interpolating curve $\mathbf{c}(u)$ (*purple*) in Minkowski space $\mathbb{R}^{2,1}$ through points Q_k (*red* dots) representing the circles. The envelopes are real if and only if $\mathbf{c}(u)$ has no pseudo-Euclidean tangents, which is guaranteed through a constraint of the curve fitting.

We come to the main chapters of this thesis; a reader familiar with cyclography, spline fitting and optimization can start directly here. We try to make them self contained and only point to the theory of the first three chapters for detailed reference.

4.1 Introduction

Circles in \mathbb{R}^2 can be represented by points in Minkowski space $\mathbb{R}^{2,1}$. In order to find envelopes for the circles, we have to approximate the corresponding points by curves in $\mathbb{R}^{2,1}$, see Fig. 4.1. This would be a standard quadratic approximation problem if reality of envelopes would be ignored.

Due to this geometric constraint, the tangents of these curves have to enclose an angle less than or equal to $\frac{\pi}{4}$ with the plane $x_3 = 0$. One can reformulate this constraint by saying that the hodograph (derivative curve) has to stay outside the Lorentz cone Γ ; it follows that this constraint is quadratic and non-convex.

In this chapter we show methods which carry out this optimization in 3-dimensional Minkowski space through several intermediate steps.

4.1.1 Overview of this chapter

We discuss related work of other authors in sec. 4.1.2. In section 4.2 we rigorously define the problem we are about to solve, i.e. finding an envelope to a set of circles and state all assumptions. We have a look at all parts of the optimization separately, that is the objective function as well as the constraints and all their ingredients. Reformulating the optimization in matrix notation proves once more that the quadratic constraint is non-convex. A general outline of an iterative optimization procedure is developed.

Section 4.3 shows that the constraint is in close relationship with the geometry of Minkowski space $\mathbb{R}^{2,1}$ itself and carries out a linearization of this constraint.

The core of this chapter, albeit rather technical, is section 4.4, which presents - in great detail - an interior point algorithm that solves the challenge of finding an envelope to circles in \mathbb{R}^2 .

Finally, section 4.5 states an independent method to verify the results of the optimization.

4.1.2 Previous Work

The problem of finding envelopes of circles has been dealt with before, but not in this generality as we will see in a quick literature overview.

Pottmann and Peternell

Pottmann & Peternell 1998 [38] consider curves \mathbf{c} in $\mathbb{R}^{3,1}$ and interpret them as canal surfaces. If \mathbf{c} is a NURBS curve, its (*convex hull, variation diminishing*) properties translate into certain properties for the rational canal surfaces. Only curves with Euclidean tangents are considered, i.e. the question of reality of envelopes is not touched. The paper gives a full description of Dupin cyclides as images of intersections of three cones in $\mathbb{R}^{3,1}$. They also define *pe circular splines* and *pe biarcs*.

The same authors presented in 2000 [40], where a wider class of approximation problems is addressed, one of them is in the space of spheres and concerns canal surfaces. They introduce a new distance measure for spheres, which turns out to be the Euclidean distance of points in \mathbb{R}^4 , see Theorem 1.

There they also show that any canal surface can be represented as a B-Spline curve, because in their earlier paper [32] it is shown for any envelope of spheres. For canal surfaces they require that the radii of spheres form a monotonous sequence, which is the only constraint (and not really necessary for general canal surfaces) they impose on the approximation; no reality-problems are considered.

Peternell, Odehnal & Sampoli [31] consider continuous one- and two-parameter families of spheres and their envelopes. They build the connection to offset curves and construct quadratic triangular Bézier surfaces in $\mathbb{R}^{3,1}$ and show the connection to rational envelopes

of spheres.

Slabaugh

Slabaugh et al. have published the work [44] in 2008 about envelopes of 1-parameter families of circles and the paper [45] in 2010 as a direct extension to 1-parameter families of spheres.

Their 2D method is to look at the two envelope curves separately and choose a point of contact S_i for every circle, which automatically gives a tangent direction (=the tangent of the circle) T_i . Given the circles, these points of contact can be coded as an angle α_i . For two consecutive circles, a cubic Hermite interpolation with data $S_i(\alpha_i)$, $S_{i+1}(\alpha_{i+1})$, $T_i(\alpha_i)$ and $T_{i+1}(\alpha_{i+1})$ is computed. This interpolation is then set up as an unconstrained quadratic optimization in the unknown α_i with minimization of arc length and curvature as objective function.

In 3D the points of contact are substituted by circles of contact, the rest extends straight forward.

The drawback of this method is that the curves are only C^1 and the knot vector of the piecewise cubic spline is determined by the rules of Hermite interpolation, thus the shape is not very flexible. It can also be doubted that circles that are close together or very different in size can be interpolated very well.

Kunkli and Hoffmann

Kunkli and Hoffmann [25] presented a work in 2010 that does a G^1 interpolation of circles via the circle of Apollonius: for three consecutive circles S_1, S_2, S_3 two other touching circles A_1, A_2 are constructed and the points of contact with A_1 are used for one envelope and the ones with A_2 for the other. Then, for two consecutive circles, tangents at the points of contact are computed and a Hermite interpolation is performed.

Their extension for spheres is to take the plane ϵ through the centers of three consecutive spheres S_i , thus get three circles and once again apply Apollonius' method. Through two touching points a plane orthogonal to ϵ intersects S_i in a circle, which is used for another Hermite interpolation.

This method is claimed to work well in practice, but the result is C^1 at best and as with [44] the shape is not flexible and the knot sequence is predetermined. Also, the set of admissible circles is very restrictive.

A special case without parametrization

We can solve simple special cases through the methods proposed by Flöry [12]: Assume the circles lie in positions similar to Fig. 4.2, i.e. we can distinguish an "interior" and an "exterior" region (=topology of disc with one hole) or the circles cover a region that has

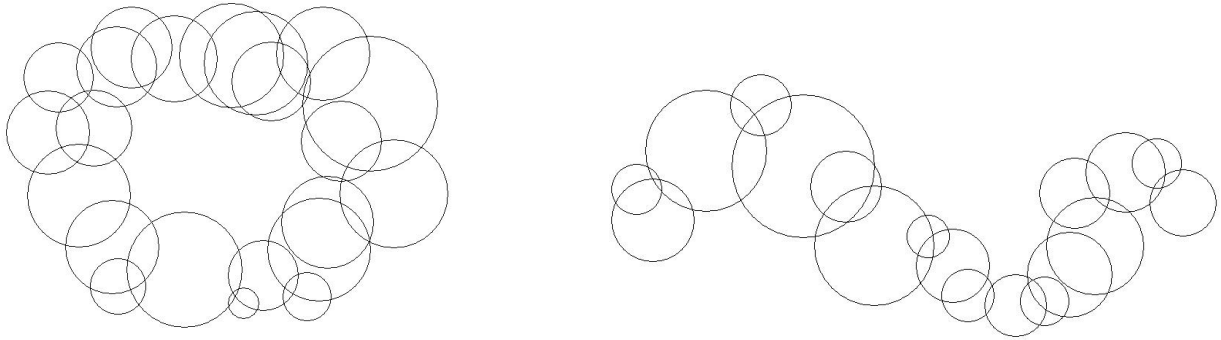


Figure 4.2: Assume these special cases: *left*: the circles allow to determine an interior and an exterior region, i.e. they have the topology of a disc with one hole *right*: the region that the circles cover has the topology of a disc. We sample the points and apply the B-Spline curve fitting as proposed in [12].

the topology of a disc, i.e. the boundary of the region is simply connected. This allows us to circumvent the parametrization problem (cf. [37]) and seems to be standard with *active contours* (cf. [3]). We then sample the circles densely and apply the routines that give us (interior and exterior) contours/envelopes. This method only works if there is a high density of circles, i.e. they overlap significantly.

4.2 Problem Statement

Finding an envelope of an ordered set of circles in \mathbb{R}^2 can be translated by cyclography into the following task: Given a set of points Q_k in $\mathbb{R}^{2,1}$, find a curve $\mathbf{c}(u)$ that interpolates these. The inverse cyclographic mapping maps $\mathbf{c}(u)$ to the two branches of the circles' envelope.

Preliminary remarks

Before we discuss the hidden challenges, let us note a slight misuse of notation. Remember that the cyclographic mapping is actually defined on *cycles*, which are circles with an orientation (clockwise or counter-clockwise), so for n circles, there are actually 2^n cycles and thus (at least) 2^n different envelopes. So strictly speaking, we are looking for envelopes of cycles in \mathbb{R}^2 . So much for the translation between circles and their corresponding points in $\mathbb{R}^{2,1}$.

Next, the set of circles is *ordered*, meaning we have defined which circle comes first and there is a unique successor for every circle except the last - and this ordering does not change. Differently put, the parameter value u_k for every point Q_k satisfies $u_k \leq u_{k+1}$

throughout the optimization we are about to define, even though u_k itself may vary.

There are as many different envelopes to a set of circles as there are different curves through the points Q_k . From now on, we will restrict our attention to cubic B-splines. Approximation with these is a well-studied subject in computer aided geometric design (see chapter 3) and many algorithms are known.

A first look at the objective function

For given points Q_k , we want to

$$\text{minimize } \sum_{k=0}^{n-1} |Q_k - \mathbf{c}(u)|^2, \quad (4.1)$$

which means solving a least squares problem for unknown control points of a cubic B-spline curve $\mathbf{c}(u)$. In fact, this approximation uses a footprint, i.e. a point $\mathbf{c}(u_k)$ which changes in an iterative sub-routine, thus making (4.1) non-linear. Note that the norm in eq. (4.1) is the Euclidean norm, because the Minkowski norm of eq. (2.3) can become zero even for isotropic vectors, which is impractical for this curve fitting.

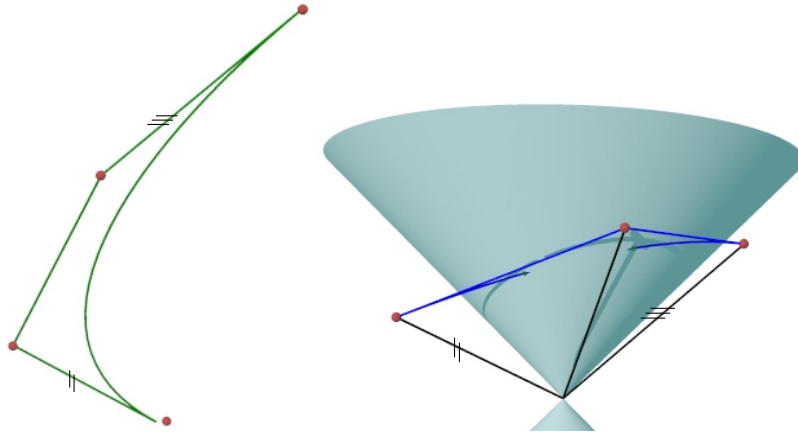


Figure 4.3: *left*: a cubic Bézier curve (*green*) and its control polygon, *right*: the hodograph (*blue*) is the curve of the first derivatives, which is a quadratic Bézier curve in the legs (*black*), i.e. the difference vectors. They originate at the origin of the Lorentz cone Γ (*turquoise*).

Introducing the constraint

The main challenge is that not every curve \mathbf{c} through points Q_k maps to a real envelope. As we have seen in chapter 2, this is only satisfied if

$$c_1'^2 + c_2'^2 \geq c_3'^2, \quad (4.2)$$

where $\frac{\partial \mathbf{c}}{\partial u} = \mathbf{c}' = (c'_1, c'_2, c'_3) \in \mathbb{R}^{2,1}$ is called *hodograph*. Wherever (4.2) is strictly $>$, the envelopes do not coincide. The condition (4.2) is equivalent to saying

$\Leftrightarrow \langle \mathbf{c}', \mathbf{c}' \rangle_L \geq 0$ by using the Lorentz inner product of eq. (2.1), i.e. the derivative vectors are *Euclidean* everywhere.

\Leftrightarrow the angle between \mathbf{c}' and the xy -plane is $\leq \frac{\pi}{4}$,

\Leftrightarrow that the hodograph has to stay outside the Lorentz cone Γ , see Fig. 4.3.

This quadratic constraint is in fact non-convex as we will see shortly, thus standard optimization algorithms can not be applied. We will present a possibility to linearize it in sec. 4.2.2.

4.2.1 Objective function in detail

In this section, we want to give specifics on how to formulate the objective function (4.1) in matrix notation, how to take into account changing parameter values and which regularizations are necessary.

But first we shall give a rough sketch of the overall optimization routine:

Overview of the unconstrained optimization procedure

Fitting a curve to a set of points Q_k translates into minimizing the sum of distances from points to a curve. For that, one usually follows this procedure:

- (1) Assume a given starting position for the curve \mathbf{c}_0 .
- (2) Choose parameters u_k , such that the *footpoint* $\mathbf{c}_i(u_k)$ is close to Q_k for every k .
- (3*) Minimize an approximation to the distance function between $\mathbf{c}_i(u_k)$ and Q_k to get a new curve \mathbf{c}_{i+1} .

Repeat (2) and (3*) until an optimum is reached.

We will specify details on how to find a good starting position (1) for a cubic B-Spline curve in sec. 4.4.2. We will not discuss it here, because the constraint also plays an important role in this matter.

Step (3*) as described here considers no constraint; for the sake of a clearer exposition of the objective function, we will leave it like that for the time being. We will however alter it and present a new procedure in sec. 4.2.3, that takes care of Euclidean tangents.

For the remainder of this section, we want to focus on footpoints (2) and the objective function (3*).

Footpoint makes objective function non-linear

How to find a footpoint $\mathbf{c}(u_k)$ for Q_k was explained in sec. 3.2, so we will only sketch it here: Find an initial parameter value u_k corresponding to Q_k through chord length parametrization. Vary this point until its tangent is perpendicular to the line connecting $\mathbf{c}(u_k)$ and Q_k . This is done through a Newton iteration and stops at some threshold or after a certain number of iterations.

As was pointed out in sec. 3.2.2, the distance function is highly non-linear and not differentiable in some areas, so usually an approximation is used: We will only consider the square of the Euclidean distance from $\mathbf{c}(u_k)$ to Q_k .

Because the footpoints are changing in step (2) of the algorithm, the minimization of the distance function in step (3) is not a linear least squares problem in the control points. Therefore, it needs to be solved iteratively and stops after the change in control points is smaller than some threshold.

Block diagonal matrix notation

Let us write n 3-dimensional curve points in a $n \times 3$ -matrix $\bar{\mathbf{c}}$. Assuming we have already found parameter values in a footpoint routine and a fixed knot vector, the collocation matrix for this cubic B-spline is \mathbf{N} and has dimensions $n \times m$, if the control points $\bar{\mathbf{b}} = [b_1|b_2|b_3]$ are $m \times 3$. We always assume coordinate vectors \mathbf{b}_i and $\mathbf{c}_i, i = 1, 2, 3$ to be column vectors. We can thus write the interpolation as a matrix multiplication

$$[c_1|c_2|c_3] = \mathbf{N} \cdot [b_1|b_2|b_3] \quad (4.3)$$

For reasons explained later, we want to write the $n \times 3$ -matrix $\bar{\mathbf{c}}$ as a $3n \times 1$ -matrix (vector) $\mathbf{c} = [c_1^T|c_2^T|c_3^T]^T$ by simply adjoining the columns; analogously $\bar{\mathbf{b}}$ is reshaped into a $3m \times 1$ -vector \mathbf{b} . We will write the points Q_k as a vector \mathbf{q} , i.e. the coordinate column vectors are adjoining. Eq. (4.3) is therefore rewritten as

$$\mathbf{c} = \left[\begin{array}{c|c|c} \mathbf{N} & & \\ \hline & \mathbf{N} & \\ \hline & & \mathbf{N} \end{array} \right] \cdot \mathbf{b}.$$

Quadratic objective in matrix notation

The objective function eq. (4.1) can be written as $(\mathbf{c}^T - \mathbf{q}^T) \cdot (\mathbf{c} - \mathbf{q}) \rightarrow \min$, thus

$$\begin{aligned} f(\mathbf{b}) &= (\mathbf{b}^T \cdot \text{blkdiag}(\mathbf{N}^T, \mathbf{N}^T, \mathbf{N}^T) - \mathbf{q}^T) \cdot (\text{blkdiag}(\mathbf{N}, \mathbf{N}, \mathbf{N}) \cdot \mathbf{b} - \mathbf{q}) \\ &= \mathbf{b}^T \cdot \underbrace{\left[\begin{array}{c|c|c} \mathbf{N}^T \mathbf{N} & & \\ \hline & \mathbf{N}^T \mathbf{N} & \\ \hline & & \mathbf{N}^T \mathbf{N} \end{array} \right]}_{:=\mathbf{G}} \cdot \mathbf{b} - 2 \cdot \mathbf{b}^T \cdot \underbrace{\left[\begin{array}{c|c|c} \mathbf{N}^T & & \\ \hline & \mathbf{N}^T & \\ \hline & & \mathbf{N}^T \end{array} \right]}_{:=\mathbf{e}} \cdot \mathbf{q} + \underbrace{\mathbf{q}^T \cdot \mathbf{q}}_{:=d \in \mathbb{R}} \quad (4.4) \end{aligned}$$

Obviously, \mathbf{G} is a symmetric block-diagonal matrix and $f(\mathbf{b})$ is a convex function in the control point coordinates \mathbf{b} . The value of $f(\mathbf{b})$ is referred to as *error measure* in chapter 6 on Examples; its graph against the number of iterations will then be called *numerical convergence rate*.

Solution via a linear system

To solve (4.4) iteratively, one writes $\mathbf{b}_{i+1} = \mathbf{b}_i + \Delta\mathbf{b}_i$ and arrives at an optimum for

$$\begin{aligned}\nabla_{\mathbf{b}}f &= \mathbf{G} \cdot (\mathbf{b}_i + \Delta\mathbf{b}_i) - \mathbf{e} = 0 \\ \Leftrightarrow \mathbf{G} \cdot \Delta\mathbf{b}_i &= \mathbf{e} - \mathbf{G} \cdot \mathbf{b}_i,\end{aligned}$$

which is a linear system in $\Delta\mathbf{b}_i$ for fixed parameters u_k . Note that this is exactly the update formula for the Point Distance Minimization (3.8). We will assume to have found an optimum, whenever the absolute value of the change in error measure $\|\nabla f\|$ is smaller than some threshold.

Regularizing the linear system

In an actual optimization routine, the *Hessian* \mathbf{G} will be regularized by adding scalar multiples of different matrices like the identity matrix (*Tikhonov regularization*, cf. sec. 3.3.1) or discrete approximations of the derivative operators. This serves two purposes: numerical stability and smoother shapes.

4.2.2 Non-convex quadratic constraint

The curve $\mathbf{c} \in \mathbb{R}^{2,1}$ shall be constrained to have only Euclidean tangents, or equivalently, its cyclographic preimage shall be a real envelope to circles $\zeta^{-1}(Q_k)$. The derivative curve or *hodograph* is given as

$$[c'_1|c'_2|c'_3] = \mathbf{N}' \cdot [b_1|b_2|b_3],$$

with the same control points $\bar{\mathbf{b}} = [b_1|b_2|b_3]$ as the curve itself, and \mathbf{N}' denoting the collocation matrix of the derivatives of the basis functions. Then the constraint (4.2) can be reformulated as

$$\begin{aligned} & b_1^T \mathbf{N}'^T \mathbf{N}' b_1 + b_2^T \mathbf{N}'^T \mathbf{N}' b_2 - b_3^T \mathbf{N}'^T \mathbf{N}' b_3 \geq 0 \\ \Leftrightarrow & \underbrace{\begin{bmatrix} b_1^T & b_2^T & b_3^T \end{bmatrix}}_{\mathbf{b}^T} \underbrace{\begin{bmatrix} \mathbf{N}'^T \mathbf{N}' & & \\ & \mathbf{N}'^T \mathbf{N}' & \\ & & -\mathbf{N}'^T \mathbf{N}' \end{bmatrix}}_{:=\mathbf{A}} \underbrace{\begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}}_{\mathbf{b}} \geq 0 \end{aligned} \quad (4.5)$$

Note that the collocation matrix of the curve's derivative \mathbf{N}' does not need to be given at the same parameter values u_k as in section 4.2.1, and the u_k do not need to change at

every iteration.

It is advantageous to have the derivatives evaluated at a dense sampling of $\mathbf{c}'(u)$.

Nevertheless, matrix \mathbf{A} is indefinite by construction, so methods like Quadratically constrained quadratic program (QCQP) cannot be applied. Note that since \mathbf{A} stems from evaluating basis functions at finitely many points, this condition is necessary, but not sufficient to satisfy the constraints.

4.2.3 Optimization procedure

Now that we have defined the matrices representing both objective function and the constraint in eq. (4.4) and (4.5) respectively, we can restate the original problem:

$$\text{minimize } \mathbf{b}^T \cdot \mathbf{G} \cdot \mathbf{b} - 2 \cdot \mathbf{b}^T \cdot \mathbf{e} + d \quad (4.6a)$$

$$\text{subject to } \mathbf{b}^T \cdot \mathbf{A} \cdot \mathbf{b} \geq 0, \quad (4.6b)$$

where \mathbf{G} is positive definite and \mathbf{A} is indefinite (both by construction).

Due to the indefiniteness of the constraint, (4.6) is a non-convex problem. If we take reparametrizations into account, which are necessary in order to include footprints, the objective function is nonlinear.

Outline of the optimization procedure

To overcome these challenges, we will follow a procedure, which enhances the one outlined in sec. 4.2.1, where step (2) was explained; (1) will have to wait until sec. 4.4.2:

- (1) Start with a good initial position \mathbf{b}_0 for the curve's control polygon.
- (2) Choose footprints $\mathbf{c}_i(u_k)$.
- (3) Compute a linearization \mathbf{D}_i of the quadratic constraint depending on the current control points \mathbf{b}_i .
- (4) Minimize the distance from \mathbf{c}_i to Q_k while staying feasible w.r.t \mathbf{D}_i and update the control points.

Repeat (2) - (4) until an optimum is reached.

We will make sure that the error introduced in (3) stays small by iteratively adapting the linearization in (2) and (3). Therefore, step (4) turns the non-convex, quadratically constrained problem (4.6) into a Quadratic Program. This important linearization will be carried out in the next section 4.3.

4.3 Linearizing the quadratic constraint

In the last section 4.2.2 we saw that the constraint for real envelopes is quadratic and non-convex. It guarantees that for each sample of the hodograph $\mathbf{c}'(u_k)$, the form $\langle \mathbf{c}'(u_k), \mathbf{c}'(u_k) \rangle_L$ stays positive.

In this section we will present a local linearization and proceed in three steps:

- Show in sec. 4.3.1 that bounding the samples of the hodograph away from the Lorentz cone can locally be accomplished through Γ 's tangent planes.
- Present a routine that computes the corresponding tangent plane for each sample in sec. 4.3.2 with the help of a projection
- Give a matrix formulation of the linearized constraint that only depends on the control points of the last iterate in sec. 4.3.

4.3.1 Local linearization of Γ

As was said in the introduction of sec. 4.2, the quadratic constraint is equivalent to the hodograph \mathbf{c}' staying outside the Lorentz cone Γ . Note that the space $\mathbb{R}^{2,1} \setminus \Gamma$ is not a vector space; a linear combination of two Euclidean vectors is not necessarily Euclidean. So rather than giving a condition on the *control polygon* of the hodograph, we look at *sampling points* $\mathbf{c}'(u_k)$.

The linearization for condition " $\mathbf{c}'(u_k)$ must stay outside Γ " will be " $\mathbf{c}'(u_k)$ must stay on the positive side of a tangent plane of Γ ". The former can be written as $c_1'^2 + c_2'^2 - c_3'^2 \geq 0$, the latter as $\bar{p}_1 c_1' + \bar{p}_2 c_2' - \bar{p}_3 c_3' \geq 0$ for $\bar{\mathbf{p}} = (\bar{p}_1, \bar{p}_2, \bar{p}_3)$ on the cone.

This connection comes naturally when looking at the equation for Γ : We gave a definition of the Lorentz cone via the norm in sec. 2.1.3. Here we only consider the border of the Lorentz cone $\partial\Gamma$, which is given by

$$\partial\Gamma : x^2 + y^2 - z^2 = 0 \quad (4.7)$$

thus the normal vector of its (isotropic) tangential plane ϵ is

$$\nabla(x^2 + y^2 - z^2) = (2x, 2y, -2z) = (x, y, -z).$$

Therefore, given a point $\bar{\mathbf{p}}$ on the cone it is contained in the plane

$$\epsilon : \bar{p}_1 x + \bar{p}_2 y - \bar{p}_3 z = 0 \quad (4.8)$$

For this tangent plane, the normal is pointing away from Γ and thus the linear form $\langle \bar{\mathbf{p}}, \mathbf{x} \rangle_L$ derived from eq. (4.8) is positive for \mathbf{x} close to $\bar{\mathbf{p}}$ and outside Γ . This linear form is also a good local approximation of the quadratic form $\langle \mathbf{x}, \mathbf{x} \rangle_L$ for a suitable $\bar{\mathbf{p}}$.

4.3.2 Projection orthogonal to Γ

We have seen in the last section that a local linearization of Γ depends on the choice of a suitable point $\bar{\mathbf{p}} = (\bar{p}_1, \bar{p}_2, \bar{p}_3)$ on the cone for a point $\mathbf{p} = (p_1, p_2, p_3)$, such that $\langle \bar{\mathbf{p}}, \mathbf{p} \rangle_L$ is positive/negative/zero for a Euclidean/pseudo-Euclidean/isotropic point. Keep in mind that these "points" are actually *derivative vectors* and hence belong to the tangent space of $\mathbb{R}^{2,1}$.

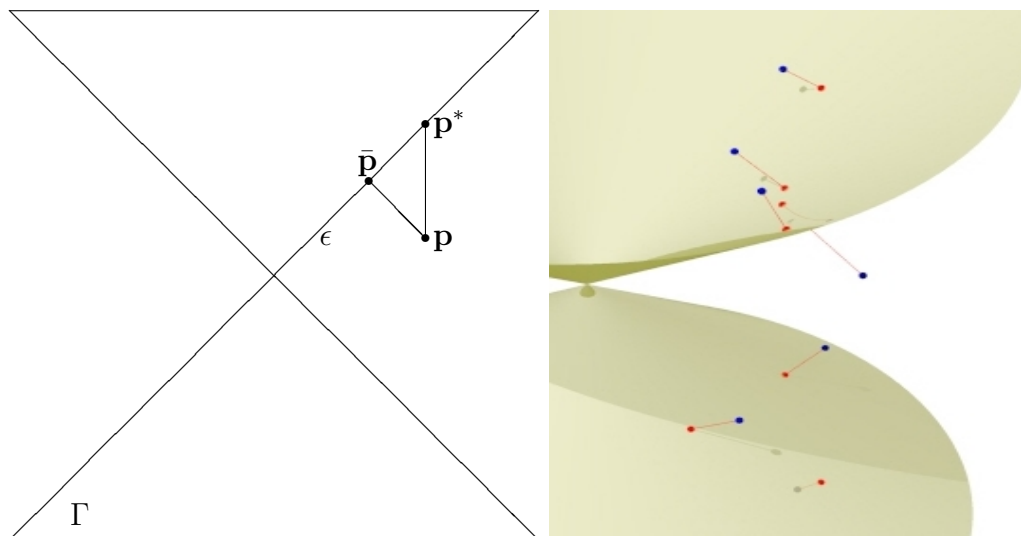


Figure 4.4: Illustration of the projection orthogonal to cone Γ , *left*: the point \mathbf{p} is projected orthogonal to the cone Γ onto point $\bar{\mathbf{p}}$, through the help of an auxiliary point \mathbf{p}^* , which in turn defines a tangent plane ϵ of Γ . (*right*:) Axonometric view of a set of points, original points in *blue*, projected ones in *red*.

If we project \mathbf{p} onto Γ orthogonal to the xy -plane, we get the point $\mathbf{p}^* = (p_1, p_2, \pm\sqrt{p_1^2 + p_2^2})$, the sign of the third coordinate is set to be the sign of p_3 . According to eq. (4.8) the tangent plane ϵ at \mathbf{p}^* is given by

$$\epsilon : p_1 \cdot x + p_2 \cdot y \mp \sqrt{p_1^2 + p_2^2} \cdot z = 0$$

We get the distance λ to the closest point $\bar{\mathbf{p}}$ of \mathbf{p} on Γ by taking the Hesse normal form of ϵ and inserting \mathbf{p} , so

$$\bar{\mathbf{p}} = \mathbf{p} - \lambda \cdot \frac{\mathbf{n}_\epsilon}{\|\mathbf{n}_\epsilon\|}, \quad \lambda = \frac{p_1^2 + p_2^2 \mp p_3 \sqrt{p_1^2 + p_2^2}}{\sqrt{2(p_1^2 + p_2^2)}} = \frac{\mathbf{n}_\epsilon^T \mathbf{p}}{\|\mathbf{n}_\epsilon\|}, \quad (4.9)$$

for the smaller of the two λ 's. See Fig. 4.4 for an illustration.

4.3.3 Matrix formulation of the linearization

We are now ready to write the linearization of the quadratic constraint (4.5) in a matrix formulation that can be used in an optimization algorithm.

We have seen in sec. 4.3.1 that for a derivative vector $\mathbf{d}_k = \mathbf{c}'(u_k) = \mathbf{N}'(u_k) \cdot \bar{\mathbf{b}}$ the quadratic constraint $\langle \mathbf{d}_k, \mathbf{d}_k \rangle_L \geq 0$ can be linearized as

$$\begin{aligned} \langle \bar{\mathbf{d}}_k, \mathbf{d}_k \rangle_L &= [\bar{d}_{k1}, \bar{d}_{k2}, \bar{d}_{k3}] \cdot \begin{pmatrix} 1 & & \\ & 1 & \\ & & -1 \end{pmatrix} \cdot \begin{bmatrix} d_{k1} \\ d_{k2} \\ d_{k3} \end{bmatrix} \\ &= [\bar{d}_{k1} \cdot \mathbf{N}'(u_k), \quad \bar{d}_{k2} \cdot \mathbf{N}'(u_k), \quad -\bar{d}_{k3} \cdot \mathbf{N}'(u_k)] \cdot \mathbf{b} \geq 0, \end{aligned}$$

with $\bar{\mathbf{d}}_k$ the projection of \mathbf{d}_k onto Γ as described in eq. (4.9), $\mathbf{N}'(u_k)$ one row of the collocation matrix of the hodograph \mathbf{c}' and \mathbf{b} the $3m \times 1$ -vector of the m control points of \mathbf{c} .

If we want to carry out this operation for all \mathbf{d}_k with a single matrix multiplication, we have to write the $\bar{\mathbf{d}}_k$ coordinate-wise in diagonal matrices and multiply these with the $n \times m$ -matrix \mathbf{N}' :

$$\mathbf{D} := \left[\begin{array}{c} \left(\begin{array}{ccc} \bar{d}_{11} & & 0 \\ & \ddots & \\ 0 & & \bar{d}_{1n} \end{array} \right) \cdot \mathbf{N}', \quad \left(\begin{array}{ccc} \bar{d}_{21} & & 0 \\ & \ddots & \\ 0 & & \bar{d}_{2n} \end{array} \right) \cdot \mathbf{N}', \quad - \left(\begin{array}{ccc} \bar{d}_{31} & & 0 \\ & \ddots & \\ 0 & & \bar{d}_{3n} \end{array} \right) \cdot \mathbf{N}' \end{array} \right]$$

and get the matrix (=linear form) linearizing the quadratic form (4.6b) of the constraint

$$\mathbf{D} \cdot \mathbf{b} = [\langle \bar{\mathbf{d}}_1, \mathbf{d}_1 \rangle_L, \dots, \langle \bar{\mathbf{d}}_n, \mathbf{d}_n \rangle_L]^T. \quad (4.10)$$

Remember that an entry $\langle \bar{\mathbf{d}}_k, \mathbf{d}_k \rangle_L$ in (4.10) is actually the signed distance for a point $\mathbf{c}'(u_k)$ to a certain (closest) tangent plane of Γ . Also keep in mind that even though \mathbf{N}' might remain unchanged throughout the iteration, \mathbf{D}_i depends on the control points \mathbf{b}_{i-1} of the last iterate, because $\mathbf{c}'_i = \mathbf{N}' \cdot \mathbf{b}_{i-1}$.

4.4 Solving the Optimization

This section will detail the optimization procedure as outlined in sec. 4.2.3. First we restate the KKT conditions for our problem in sec. 4.4.1, give some definitions as well as an outline of an interior point algorithm. We will then initialize the control polygon and define the derivative's collocation matrix \mathbf{N}' in sec. 4.4.2. In the main section 4.4.3 we describe all steps of the iteration that are to be repeated until an optimum is found.

4.4.1 Algorithm for Interior Point Method

We will briefly restate some definitions for the interior point algorithm and highlight the important steps as they appear in our problem. The general theory can be found in Nocedal & Wright [29].

KKT conditions

We have defined the positive semidefinite Hessian matrix \mathbf{G} and the vector \mathbf{e} in eq. (4.4) and the linearized constraint \mathbf{D} in eq. (4.10). The Karush-Kuhn-Tucker conditions for the quadratic problem (4.6) can thus be stated as

$$\mathbf{G}\mathbf{b} - \mathbf{D}^T \lambda_k + \mathbf{e} = 0 \quad (4.11a)$$

$$\mathbf{D}\mathbf{b} - \mathbf{y} = 0 \quad (4.11b)$$

$$\mathbf{y} \geq 0 \quad (4.11c)$$

$$\lambda_k \geq 0 \quad (4.11d)$$

$$y_i \lambda_i = 0, \quad i = 1, 2, \dots, m, \quad (4.11e)$$

with *slack variables* \mathbf{y} and the *Lagrange multipliers* $\boldsymbol{\lambda}$. We call the left hand side of eq. (4.11a) the *dual residual* \mathbf{r}_d and accordingly the left hand side of eq. (4.11b) the *primal residual* \mathbf{r}_p . As is proved in [29] that the KKT conditions (4.11) are not only necessary, but also sufficient.

Put differently, we want to iteratively solve the system of equations

$$F(\mathbf{b}, \mathbf{y}, \boldsymbol{\lambda}) = [\mathbf{r}_d, \mathbf{r}_p, \mathcal{Y}\Lambda e]^T = 0$$

with $\mathcal{Y} = \text{diag}(y_1, y_2, \dots, y_m)$, $\Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_m)$, and $e = (1, 1, \dots, 1)^T$ using Newton's method.

Affine scaling step

The solution of the linear system

$$\begin{bmatrix} \mathbf{G} & 0 & -\mathbf{D}^T \\ \mathbf{D} & -\mathbf{I} & 0 \\ 0 & \Lambda & \mathcal{Y} \end{bmatrix} \begin{bmatrix} \mathbf{b}_{\text{aff}} \\ \mathbf{y}_{\text{aff}} \\ \boldsymbol{\lambda}_{\text{aff}} \end{bmatrix} = -F(\mathbf{b}, \mathbf{y}, \boldsymbol{\lambda}), \quad (4.12)$$

is called *affine scaling direction* and the matrix on the left hand side *Jacobian* $\mathbf{J}(\mathbf{b}, \mathbf{y}, \boldsymbol{\lambda})$ of F . $[\mathbf{b}_{\text{aff}}, \mathbf{y}_{\text{aff}}, \boldsymbol{\lambda}_{\text{aff}}]^T$ can violate (4.11e), so a line search parameter $\alpha > 0$ is needed to guarantee that $(\mathbf{b}_{\text{aff}}, \mathbf{y}_{\text{aff}}, \boldsymbol{\lambda}_{\text{aff}}) + \alpha(\Delta\mathbf{b}_{\text{aff}}, \Delta\mathbf{y}_{\text{aff}}, \Delta\boldsymbol{\lambda}_{\text{aff}})$ is feasible.

Center step

As this step length α might be too short, one introduces the *duality measure* as $\mu = \mathbf{y}^T \boldsymbol{\lambda} / m$ and tries to find a point which gradually reduces μ by a *centering parameter* $\sigma \in [0, 1]$, i.e.

solve

$$\mathbf{J}(\mathbf{b}, \mathbf{y}, \boldsymbol{\lambda}) \cdot [\Delta \mathbf{b}, \Delta \mathbf{y}, \Delta \boldsymbol{\lambda}]^T = [-\mathbf{r}_d, -\mathbf{r}_p, -\mathcal{Y}\Lambda e + \sigma \mu e]^T \quad (4.13)$$

After that, another step length selection is carried out to find the optimal step size $\hat{\alpha}$.

Summary of the Interior Point Method algorithm

After computing a feasible starting position $(\mathbf{x}_0, \mathbf{y}_0, \boldsymbol{\lambda}_0)$ with $(\mathbf{y}_0, \boldsymbol{\lambda}_0)$, the general algorithm follows these steps:

1. Set $(\mathbf{x}, \mathbf{y}, \boldsymbol{\lambda}) = (\mathbf{x}_k, \mathbf{y}_k, \boldsymbol{\lambda}_k)$ and solve (4.12) with $\sigma = 0$ to obtain $(\Delta \mathbf{x}_{\text{aff}}, \Delta \mathbf{y}_{\text{aff}}, \Delta \boldsymbol{\lambda}_{\text{aff}})$
2. calculate *duality measure* $\mu = \mathbf{y}^T \boldsymbol{\lambda} / m$ and *affine step length* $\hat{\alpha}_{\text{aff}} = \max\{\alpha \in (0, 1] \mid (\mathbf{y}, \boldsymbol{\lambda}) + \alpha(\Delta \mathbf{y}_{\text{aff}}, \Delta \boldsymbol{\lambda}_{\text{aff}}) \geq 0\}$ and set $\mu_{\text{aff}} = (\mathbf{y} + \hat{\alpha}_{\text{aff}} \Delta \mathbf{y}_{\text{aff}})^T (\boldsymbol{\lambda} + \hat{\alpha}_{\text{aff}} \Delta \boldsymbol{\lambda}_{\text{aff}}) / m$
3. set centering parameter $\sigma = (\mu_{\text{aff}} / \mu)^3$ and solve (4.13) for full step $(\Delta \mathbf{x}, \Delta \mathbf{y}, \Delta \boldsymbol{\lambda})$
4. choose $\tau_k \in (0, 1) \rightarrow 1$ as k increases and set $\hat{\alpha} = \min(\alpha_{\tau_k}^{\text{pri}}, \alpha_{\tau_k}^{\text{dual}})$, see (4.14)
5. $(\mathbf{x}_{k+1}, \mathbf{y}_{k+1}, \boldsymbol{\lambda}_{k+1}) = (\mathbf{x}_k, \mathbf{y}_k, \boldsymbol{\lambda}_k) + \hat{\alpha}(\Delta \mathbf{x}, \Delta \mathbf{y}, \Delta \boldsymbol{\lambda})$, continue with (1).

This iteration is repeated until no further increases in \mathbf{x} can be obtained.

The step length $\hat{\alpha}$ is the minimum of the step lengths of the primal and dual variable, which are defined dependent on $\tau \in (0, 1)$:

$$\begin{aligned} \alpha_{\tau}^{\text{pri}} &= \max\{\alpha \in (0, 1] \mid \mathbf{y} + \alpha \Delta \mathbf{y} \geq (1 - \tau) \mathbf{y}\} \\ \alpha_{\tau}^{\text{dual}} &= \max\{\alpha \in (0, 1] \mid \boldsymbol{\lambda} + \alpha \Delta \boldsymbol{\lambda} \geq (1 - \tau) \boldsymbol{\lambda}\} \end{aligned} \quad (4.14)$$

4.4.2 Initial position

We would like to have a much bigger number m of control points \mathbf{b} than the number n of points $Q_k \in \mathbb{R}^{2,1}$ representing the circles in order to have more flexibility in optimization. Trying to solve this directly would lead to a rank-deficient linear system whose solution \mathbf{b} can, of course, not be trusted.

To circumvent this problem, we introduce an auxiliary cubic B-spline curve \mathbf{c}_{aux} , interpolating the input Q_k with a minimal knot vector and ignoring the steepness constraint, see Fig. 4.5.

We take $m_1 > m + 4$ equally spaced auxiliary parameter values v_i , a knot vector of length $m + 4$ (multiplicity 4 at the beginning and the end, $m - 4$ equally spaced in between) and compute thus the collocation matrix \mathbf{N}_{aux} of rank m .

The z -coordinates of the control points \mathbf{b}_{aux} of \mathbf{c}_{aux} are then moved half the way to the mean value of the z -coordinates of the Q_k , called $mean(Q_{k,z})$. Should \mathbf{c}_{aux} still have pseudo-

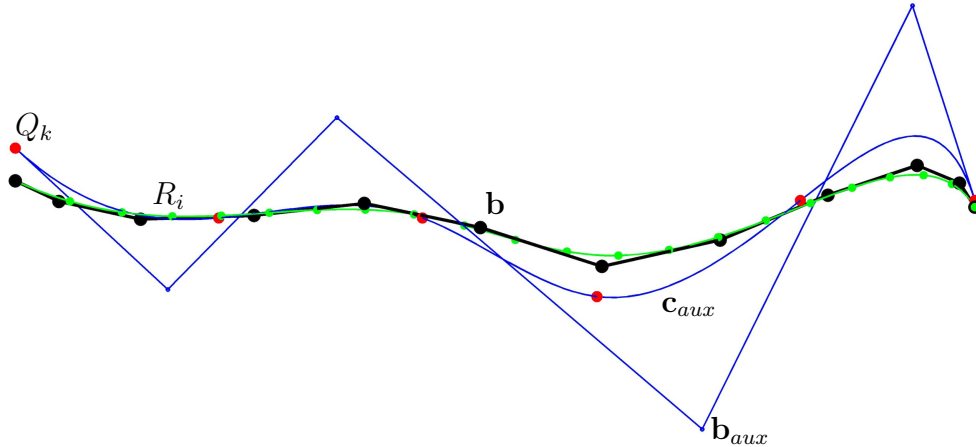


Figure 4.5: Procedure to find the initial position of the control polygon: Compute auxiliary curve \mathbf{c}_{aux} (blue curve) with control polygon \mathbf{b}_{aux} (blue lines) through input points Q_k (red dots). Move this control polygon toward the mean of the z -coordinates of the Q_k until no step tangents appear, then sample points R_i (green dots). A finer control polygon \mathbf{b} (black) for approximating those is found via a linear system.

Euclidean tangents, repeat this scaling; sample the final Euclidean curve $R_i = \mathbf{c}_{aux}(v_i)$. The initial control polygon \mathbf{b} for the optimization procedure is the least squares fit of the over-determined system $\mathbf{N}_{aux} \cdot \mathbf{b} = R_i$. It is Euclidean by construction and the number of control points m is a user-specified number.

Note that due to the repeated scaling of the z -coordinates of \mathbf{c}_{aux} , the B-Spline defined by \mathbf{b} can be quite far from the Q_k (only in z -direction); in the worst case, it is approximating the top view projection of the Q_k in the plane $z = \text{mean}(Q_{k,z})$.

Initializing the derivative's collocation matrix \mathbf{N}'

Set the number h for which the hodograph $c'(u_i)$ shall be guaranteed to be Euclidean. Let these h equally-spaced divisions of the unit interval as parameter values and the knot vector as before define the collocation matrix of the derivatives \mathbf{N}' , i.e. $c'(u_i) = \mathbf{N}' \cdot \mathbf{b}$.

Initializing the Lagrange multipliers λ

Initialize the vector of Lagrange multipliers as $\lambda = (1, \dots, 1)$. This value will be changed drastically again in the first iteration, before being regularly updated with all the other variables during the following iterations.

4.4.3 Iteration of the Interior Point Algorithm

We have presented the general strategy of the Interior Point algorithm in use in sec. 4.4.1 and gave initializations for the control points and other important variables in sec. 4.4.2. Here we will show the updates of all variables that occur during the iteration in detail. The numbering (1), (2), ... is independent of other sections; note that the paragraphs marked (*) only occur in the first iteration after initialization.

Usually this iteration runs until either the change in the control points is less than a threshold or until a maximum number of iterations is reached.

(1) Collocation matrix \mathbf{N}_i

The control points \mathbf{b}_i found in the last iteration define the curve's current position \mathbf{c}_i , so we need to update the parameter values $\mathbf{c}_i(u_k)$ for input points Q_k . Explicit formulas for this footpoint routine were given in sec. 3.2.

The parameter values together with the knot vector give rise to a new collocation matrix \mathbf{N}_i . The matrix $\mathbf{N}_i^T \mathbf{N}_i$ is regularized as $\mathbf{N}_i^T \mathbf{N}_i + \lambda_1 \mathbf{L}_1 + \lambda_2 \mathbf{L}_2 + \lambda_T \mathbf{I}_n$ where \mathbf{L}_1 and \mathbf{L}_2 are discrete approximations of the derivative operator and \mathbf{I}_n is the identity matrix, see sec. 3.3.1 on Tikhonov regularization.

At this point we can also return value of the objective function or *error measure* $f(\mathbf{b}_i) = (\mathbf{N}_i \cdot \mathbf{b}_i - \mathbf{q})^2$, which will be the value of the *numerical convergence rate* in chapter 6.

(2) Matrix \mathbf{D}_i of the linearized constraint

In section 4.3.3 we have defined the matrix \mathbf{D} of the linearized constraint, which apart from \mathbf{N}' contains projections $\bar{\mathbf{d}}$ of derivatives $\mathbf{d} = \mathbf{N}'(u_k) \cdot \bar{\mathbf{b}}_i$ onto Γ . Remember that an entry in the vector $\mathbf{D}_i \cdot \mathbf{b}_i = [\langle \bar{\mathbf{d}}_1, \mathbf{d}_1 \rangle_L, \dots, \langle \bar{\mathbf{d}}_n, \mathbf{d}_n \rangle_L]^T$ is greater than zero if the corresponding derivative vector \mathbf{d} is Euclidean.

This vector is almost (up to the linearization error) the vector of *slack variables* \mathbf{y}_i in the KKT-conditions as introduced in sec. 4.4.1 and should stay positive during the optimization.

(*) Further variable initializations in the first iteration

The first run in the iteration is different from all the following as it is used to further initialize variables: the vector of slack variables is introduced as $\mathbf{y}_1 = \mathbf{D}_1 \cdot \mathbf{b}_1$ and updated again before iteration number 2, as we will see shortly.

(3) Affine scaling step

At this point, all the matrices needed to compute the *affine scaling direction* of equation (4.12) are available, so we can solve

$$\begin{bmatrix} \mathbf{G} & 0 & -\mathbf{D}^T \\ \mathbf{D} & -\mathbf{I} & 0 \\ 0 & \Lambda & \mathcal{Y} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{b}_{\text{aff}} \\ \Delta \mathbf{y}_{\text{aff}} \\ \Delta \boldsymbol{\lambda}_{\text{aff}} \end{bmatrix} = - \begin{bmatrix} \mathbf{G}\mathbf{b} - \mathbf{D}^T \boldsymbol{\lambda} + \mathbf{e} \\ \mathbf{D}\mathbf{b} - \mathbf{y} \\ \Lambda \mathcal{Y}e \end{bmatrix}.$$

Note that the Jacobian \mathbf{J} (matrix on the left-hand side) is sparse and especially for small slack variables \mathbf{y} ill-conditioned, so a Tikhonov regularization is usually carried out, which means adding a scalar multiple of the identity matrix to \mathbf{J} .

(*) Special case for the first iteration

Only in the first iteration, the slack variables \mathbf{y} are slightly corrected by redefining $\mathbf{y} := \max(10^{-5}, \mathbf{y} + 10^{-4} \Delta \mathbf{y}_{\text{aff}})$; the Lagrange multipliers (which were initialized to be one everywhere) are also updated as $\boldsymbol{\lambda} := 10^{-3} \min(\max(0, \boldsymbol{\lambda} + \Delta \boldsymbol{\lambda}_{\text{aff}}), 1)$, which works very well in practice and differs from [29].

The first iteration ends here; the second starts again at (1).

(4) Center step

The duality measure $\mu = \mathbf{y}^T \boldsymbol{\lambda} / m$ is computed. Then find the biggest α_1 and α_2 such that

$$\mathbf{y} + \alpha_1 \Delta \mathbf{y}_{\text{aff}} \geq 0, \quad \text{and} \quad \boldsymbol{\lambda} + \alpha_2 \Delta \boldsymbol{\lambda}_{\text{aff}} \geq 0.$$

and define

$$\hat{\alpha}_{\text{aff}} := \min(\alpha_1, \alpha_2), \quad \mu_{\text{aff}} := (\mathbf{y} + \hat{\alpha}_{\text{aff}} \Delta \mathbf{y}_{\text{aff}})^T (\boldsymbol{\lambda} + \hat{\alpha}_{\text{aff}} \Delta \boldsymbol{\lambda}_{\text{aff}}) / m, \quad \sigma := \mu_{\text{aff}} / \mu.$$

Next compute the *center step*

$$\mathbf{J}(\mathbf{x}, \boldsymbol{\lambda}, \mathbf{s}) \cdot [\Delta \mathbf{b}, \Delta \mathbf{y}, \Delta \boldsymbol{\lambda}]^T = [-\mathbf{r}_d, -\mathbf{r}_p, -\mathcal{Y}\Lambda e + \sigma \mu e]^T.$$

If $\text{mean}(|\Delta \mathbf{b}|) < 10^{-5}$, the optimum ("*KKT point*") is reached and the iteration ended.

Else, the step length $\hat{\alpha}$ is defined to be the minimum of the step lengths of the primal and dual variable:

$$\begin{aligned} \alpha_{\tau}^{\text{pri}} &= \max\{\alpha \in (0, 1] \mid \mathbf{y} + \alpha \Delta \mathbf{y} \geq (1 - \tau) \mathbf{y}\} \\ \alpha_{\tau}^{\text{dual}} &= \max\{\alpha \in (0, 1] \mid \boldsymbol{\lambda} + \alpha \Delta \boldsymbol{\lambda} \geq (1 - \tau) \boldsymbol{\lambda}\}, \end{aligned}$$

which are defined dependent on $\tau \in (0, 1)$; we chose this heuristically to be $(i - 2)/i$ for the number of iteration i .

(5) Updating the variables

We end the iteration by updating the variables:

$$\begin{aligned}\mathbf{b}_{i+1} &= \mathbf{b}_i + \hat{\alpha}\Delta\mathbf{b} \\ \mathbf{y}_{i+1} &= \mathbf{y}_i + \hat{\alpha}\Delta\mathbf{y} \\ \boldsymbol{\lambda}_{i+1} &= \boldsymbol{\lambda}_i + \hat{\alpha}\Delta\boldsymbol{\lambda}.\end{aligned}$$

4.4.4 Computed examples

The iterative algorithm as explained in detail over the course of the last section was applied to all the examples in ch. 6. We will show a computed example in section 6.1.1 that specifies the numeric value of every variable given here as well as showing figures of the initialization, during the iteration and for the result.

4.5 Testing the tangents algebraically

Once we have arrived at a result through optimization as presented in the previous sections, we need to check whether these results actually satisfy the steepness constraint.

Given a B-spline curve $\mathbf{c}(u)$, $u \in [0, 1]$ of degree n , its collocation matrix \mathbf{N} and its control points \mathbf{b}_k , there is an algebraic test to make sure that \mathbf{c}' is always Euclidean: Establish the derivative function $f(u) = c'_x{}^2 + c'_y{}^2 - c'_z{}^2$. Clearly f is a piecewise polynomial of degree $2(n-1)$, its roots correspond to the isotropic tangents of $\mathbf{c}(u)$. If f has no roots in $[0, 1]$, then all tangents are Euclidean.

Since $f(u)$ is a polynomial, one can check with Descartes' rule of signs (see [42]) how many zeros to expect; if all coefficients have the same sign, $f(u)$ has no real zeros.

The zeros of the derivative function $f(u) = a_0 + a_1u + \dots + a_{2n-1}u^{2n-1} + a_{2(n-1)}u^{2(n-1)}$ can also be computed via its *companion matrix*

$$\mathbf{A}(f) = \begin{bmatrix} 0 & \cdots & 0 & -a_0 \\ 1 & \cdots & 0 & -a_1 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 1 & -a_{2(n-1)} \end{bmatrix},$$

see [20]. The roots of $f(u)$ are the eigenvalues of $\mathbf{A}(f)$, standard techniques for the estimation of these suffice.

5 Envelopes of Spheres

This chapter extends the theory of chapter 4 to the 4-dimensional $\mathbb{R}^{3,1}$ in two different ways. In section 5.1 we will look at curves in this Minkowski space, which represent envelopes of 1-parameter families of spheres, i.e. canal surfaces. Basically, the theory stays the same as in chapter 4 and we will not bother repeating the general optimization framework or issues of initial position, only point to the formulas that have to be adapted due to the higher dimension.

The major contribution of the chapter lies with surfaces in $\mathbb{R}^{3,1}$, whose inverse cyclographic image are envelopes of 2-parameter families of spheres. The constraint of having real envelopes means that the tangent planes of these surfaces have to stay Euclidean. In fact, we are talking of 2-planes in 4-space and section 5.2 introduces a calculus that allows to classify such planes through a bilinear form.

In section 5.3 we will take a closer look at hyperbolic paraboloids in $\mathbb{R}^{3,1}$ and state an algorithm that ensures that they stay Euclidean, which is based on projective geometric considerations.

All definitions for the surface optimization, as well as the proper linearizations for the quadratic constraint, are carried out in sec. 5.4. Finally, in section 5.5 the actual optimization of surfaces in $\mathbb{R}^{3,1}$ is described in all detail, including the initialization.

Previous work

A look at previous work has already been carried out in sec. 4.1.2, which includes the case of 1-parameter families of spheres, i.e. canal surfaces and Dupin cyclides as special cases. As we will see in section 5.1, they are direct extensions of the lower dimensional circle approximation. To our knowledge, the case of 2-parameter families of spheres has not yet been considered.

5.1 Canal surfaces

In chapter 4 we mostly spoke of curves in $\mathbb{R}^{2,1}$, their inverse cyclographic image being envelopes of circles, and how to solve the optimization for those. In this section we will provide the necessary changes to the formulas in order to extend our code to $\mathbb{R}^{3,1}$. The inverse cyclographic images of these are envelopes of 1-parameter families of spheres, i.e. canal surfaces.

5.1.1 Optimization formulation

Given a set of spheres, we want to find an envelope for these, i.e. find the interpolating canal surface. Each of these spheres can be represented by a point $Q_k \in \mathbb{R}^{3,1}$; an interpolating curve $\mathbf{c}(u)$ maps to the canal surface via the inverse cyclographic mapping.

Exactly like in the lower dimension, if $\mathbf{c}(u)$ has tangents, which are steeper than $\frac{\pi}{4}$ against the $x_4 = 0$ -hyperplane, the corresponding canal surface is not real locally. Therefore, we introduce a quadratic constraint, which is a direct extension of the lower dimensional case.

5.1.2 Generalization of the objective function to $\mathbb{R}^{3,1}$

The objective function (4.1) stays the same, apart from the fact that given points Q_k are in $\mathbb{R}^{3,1}$, so

$$\text{minimize } \sum_{k=0}^{n-1} |Q_k - \mathbf{c}(u)|^2, \quad (5.1)$$

where we will understand $\mathbf{c}(u)$ to be a cubic B-spline. The distance is the *Euclidean* distance in \mathbb{R}^4 , because for a distance based on the Minkowski norm of eq. (2.3), distances measured in isotropic direction are zero, which causes ambiguities which we want to avoid.

Footpoints

The footpoints for a curve $\mathbf{c}(u)$ in $\mathbb{R}^{2,1}$ were introduced in sec. 3.2, where we gave a definition that works for curves in any dimension. The challenges explained in sec. 4.2.1 are therefore the same in $\mathbb{R}^{3,1}$: because of changing footpoints, the objective function becomes a non-linear least squares problem, which has to be solved iteratively.

Block-diagonal matrix formulation

The formulation of the quadratic objective function with block-diagonal matrices is - in principle - the same in $\mathbb{R}^{3,1}$ as for the lower dimensional case of sec. 4.2.1. One only extends eq. (4.3) by one dimension to get $[c_1|c_2|c_3|c_4] = \mathbf{N} \cdot [b_1|b_2|b_3|b_4]$.

We can thus restate the objective function (4.4) as

$$\mathbf{b}^T \cdot \begin{bmatrix} \mathbf{N}^T \mathbf{N} & & & \\ & \mathbf{N}^T \mathbf{N} & & \\ & & \mathbf{N}^T \mathbf{N} & \\ & & & \mathbf{N}^T \mathbf{N} \end{bmatrix} \cdot \mathbf{b} - 2 \cdot \mathbf{b}^T \cdot \begin{bmatrix} \mathbf{N}^T & & & \\ & \mathbf{N}^T & & \\ & & \mathbf{N}^T & \\ & & & \mathbf{N}^T \end{bmatrix} \cdot \mathbf{q} + \mathbf{q}^T \cdot \mathbf{q},$$

and call the block-diagonal matrix on the left side the *Hessian* \mathbf{G} . All empty blocks are filled with zero matrices, thus these two matrices are very sparse.

Regularization

As before, the *Hessian* \mathbf{G} will be *Tikhonov regularized* by adding scalar multiples of the identity matrix or discrete approximations of the derivative operators, see sec. 3.3.1. The magnitude of the factor for Tikhonov regularization is usually depending on a rough estimation of \mathbf{G} 's condition number and stays in the magnitude of 10^{-5} . As before, this helps with numerical stability and brings smoother shapes.

5.1.3 Generalization of the steepness constraint to $\mathbb{R}^{3,1}$

In sec. 4.2 we introduced the quadratic constraint (4.2) for curves $\mathbf{c}(u) \in \mathbb{R}^{2,1}$, which translates to

$$c_1'^2 + c_2'^2 + c_3'^2 \geq c_4'^2 \quad (5.2)$$

for curves in $\mathbb{R}^{3,1}$. As before, one can give three equivalent formulations for this equation:

- ⇔ The Lorentz inner product $\langle \mathbf{c}', \mathbf{c}' \rangle_L$ of eq. (2.1) has to be ≥ 0 , i.e. the derivative vectors \mathbf{c}' are *Euclidean* everywhere.
- ⇔ The angle between \mathbf{c}' and the $x_4 = 0$ -hyperplane has to be $\leq \frac{\pi}{4}$.
- ⇔ The hodograph has to stay outside the Lorentz cone Γ .

Especially by the last condition we can see that this quadratic constraint is in fact non-convex, thus we have to linearize it in order to solve the optimization by a quadratic program.

Linearization

We can transfer almost all of section 4.3 on the linearization of the constraint in $\mathbb{R}^{2,1}$ to the 4-dimensional Minkowski space verbatim. As the Lorentz cone was defined without specifying the dimension in sec. 2.1.3, we want to redefine (4.7), the border of the Lorentz cone $\partial\Gamma$ by

$$x^2 + y^2 + z^2 - w^2 = 0.$$

Thus a point $\bar{\mathbf{p}}$ on Γ is contained in the plane

$$\epsilon : \bar{p}_1 x + \bar{p}_2 y + \bar{p}_3 z - \bar{p}_4 w = 0,$$

and $\bar{\mathbf{p}}$ is defined through a projection onto the cone as was explained in sec. 4.9.

The matrix \mathbf{D} essential to the linearization of the quadratic steepness constraint thus changes to $\mathbf{D} :=$

$$[diag(\bar{d}_{11} \cdots \bar{d}_{1m_1}) \cdot \mathbf{N}' | diag(\bar{d}_{21} \cdots \bar{d}_{2m_1}) \cdot \mathbf{N}' | diag(\bar{d}_{31} \cdots \bar{d}_{3m_1}) \cdot \mathbf{N}' | -diag(\bar{d}_{41} \cdots \bar{d}_{4m_1}) \cdot \mathbf{N}'].$$

\mathbf{D} will change iteratively during the optimization as $\mathbf{d}_k = \mathbf{c}(u_k)$ is the curve's derivative and $\bar{\mathbf{d}}_k$ its projection; the derivatives' collocation matrix \mathbf{N}' changes as well.

5.1.4 Solving the optimization for canal surfaces

In sec. 5.1.3 we have successfully transferred the non-linear quadratic optimization problem (5.1) with quadratic non-convex constraint (5.2) into a non-linear quadratic program

$$\begin{aligned} & \text{minimize} && \mathbf{b}^T \cdot \mathbf{G} \cdot \mathbf{b} - 2 \cdot \mathbf{b}^T \cdot \mathbf{e} + d \\ & \text{subject to} && \mathbf{D} \cdot \mathbf{b} \geq 0, \end{aligned}$$

by linearizing the constraint. We will solve it with the help of an interior point algorithm, which was described in great detail in sec. 4.4.

Other than the change from three to four coordinates in the control points \mathbf{b} and thus the increased size of matrices \mathbf{G} and \mathbf{D} , everything stays the same and we will not restate the whole procedure here.

Initial position

Note that the initial position for the control points \mathbf{b} can be computed exactly as in sec. 4.4.2 through auxiliary points on a spline that is not subject to the constraint and then lower the last coordinate.

5.2 2-planes in Minkowski 4-space

The aim of this section is to find a classification of planes through the origin (linear homogeneous subspaces of dimension 2) in Minkowski space $\mathbb{R}^{3,1}$ via a bilinear form. We use this in sec. 5.4 for approximating points in $\mathbb{R}^{3,1}$ by flat surfaces, i.e. surfaces that have Euclidean tangent planes.

As a motivation, we show a similar construction in \mathbb{R}^3 and 3-dimensional Minkowski space $\mathbb{R}^{2,1}$ via the wedge product of two vectors, which belong to an exterior algebra.

5.2.1 Wedge product in 3-space

Before defining the wedge product in Minkowski space, we recall some facts from Euclidean \mathbb{R}^3 . This serves both as example and guideline for similar constructions in 4-space. We will then introduce a bilinear form $\Phi_{2,1}$, which is basically the Minkowski norm for the cross product of two vectors and gives a number that is used to classify planes. Again, this helps to generalize such a classification of 2-planes in $\mathbb{R}^{3,1}$ later on.

Wedge product in \mathbb{R}^3

For two vectors $\mathbf{u} = (u_0, u_1, u_2)$, $\mathbf{v} = (v_0, v_1, v_2)$ in \mathbb{R}^3 the cross product is defined as

$$\mathbf{u} \times \mathbf{v} = (u_1v_2 - u_2v_1, -u_0v_2 + u_2v_0, u_0v_1 - u_1v_0),$$

which again is a vector in \mathbb{R}^3 . This cross product is a special case of the wedge product $\mathbf{u} \wedge \mathbf{v}$, see [8]. As we will see in sec. 5.2.2, the fact that the wedge product of two vectors is again a vector is only true in dimension 3.

The inner product on \mathbb{R}^3 is defined as

$$\langle \mathbf{u}, \mathbf{v} \rangle = u_0v_0 + u_1v_1 + u_2v_2.$$

and the following is true for all $\mathbf{u}, \mathbf{v}, \mathbf{x}, \mathbf{y} \in \mathbb{R}^3$, as a simple calculation shows:

$$\langle \mathbf{u} \wedge \mathbf{v}, \mathbf{x} \wedge \mathbf{y} \rangle = \det \begin{pmatrix} \langle \mathbf{u}, \mathbf{x} \rangle & \langle \mathbf{v}, \mathbf{x} \rangle \\ \langle \mathbf{u}, \mathbf{y} \rangle & \langle \mathbf{v}, \mathbf{y} \rangle \end{pmatrix}, \quad (5.3)$$

with the same inner product as before. For $\mathbf{u} = \mathbf{x}$ and $\mathbf{v} = \mathbf{y}$ this brings

$$\langle \mathbf{u} \wedge \mathbf{v}, \mathbf{u} \wedge \mathbf{v} \rangle = \langle \mathbf{u}, \mathbf{u} \rangle \cdot \langle \mathbf{v}, \mathbf{v} \rangle - \langle \mathbf{u}, \mathbf{v} \rangle^2, \quad (5.4)$$

which is 0 if and only if either $\mathbf{u} = \lambda \mathbf{v}$, $\lambda \in \mathbb{R}$ or $\mathbf{u} = \mathbf{o}$ or $\mathbf{v} = \mathbf{o}$.

Wedge product in $\mathbb{R}^{2,1}$

In 3-dimensional Minkowski space $\mathbb{R}^{2,1}$ the inner product of two vectors $\mathbf{u}, \mathbf{v} \in \mathbb{R}^{2,1}$ has been defined in sec. 2.1.1 as

$$\langle \mathbf{u}, \mathbf{v} \rangle_L = u_0v_0 + u_1v_1 - u_2v_2;$$

remember that it is $< 0 / = 0 / > 0$ for *Euclidean/isotropic/pseudo-Euclidean* vectors.

Also in sec. 2.1.1, the pseudo-Euclidean cross product has been defined as

$$\mathbf{u} \times_L \mathbf{v} = (u_1v_2 - u_2v_1, -u_0v_2 + u_2v_0, -u_0v_1 + u_1v_0); \quad (5.5)$$

note that only the third coordinate is different from the Euclidean case. (The formulation with wedge products would be the same, as it does not depend on chosen basis vectors.)

Two linearly independent vectors $\mathbf{u}, \mathbf{v} \in \mathbb{R}^{2,1}$ span a plane ϵ and define the plane's normal as $\mathbf{n}_\epsilon = \mathbf{u} \times_L \mathbf{v}$. We use this notation to reformulate the characterization of hyperplanes in Minkowski space that was given in sec. 2.1.2:

Proposition 2. $\langle \mathbf{u} \times_L \mathbf{v}, \mathbf{u} \times_L \mathbf{v} \rangle_L$ is $\left\{ \begin{array}{l} < 0 \\ = 0 \\ > 0 \end{array} \right\} \Leftrightarrow \epsilon$ is $\left\{ \begin{array}{l} \textit{Euclidean} \\ \textit{isotropic} \\ \textit{pseudo-Euclidean} \end{array} \right\}.$

Note that the proposition leaves it open whether or not \mathbf{u} or \mathbf{v} are Euclidean/isotropic/pseudo-Euclidean. It is possible for two Euclidean lines to span all three types of planes, see Prop. 1 on page 8.

The bilinear form $\Phi_{2,1}$

For four vectors $\mathbf{u}, \mathbf{v}, \mathbf{x}, \mathbf{y} \in \mathbb{R}^{2,1}$ we define a bilinear form $\Phi_{2,1} : \bigwedge^2 \mathbb{R}^{2,1} \times \bigwedge^2 \mathbb{R}^{2,1} \equiv \mathbb{R}^{2,1} \times \mathbb{R}^{2,1} \rightarrow \mathbb{R}$ by

$$\Phi_{2,1}(\mathbf{u} \wedge \mathbf{v}, \mathbf{x} \wedge \mathbf{y}) = -\langle \mathbf{u} \times_L \mathbf{v}, \mathbf{x} \times_L \mathbf{y} \rangle_L.$$

It follows that the pseudo-Euclidean cross product eq. (5.5), which defines the normal vector of the plane ϵ spanned by \mathbf{u} and \mathbf{v} has the coordinates $\mathbf{u} \times_L \mathbf{v} = (l_{12}, -l_{02}, -l_{01}) \in \bigwedge^2 \mathbb{R}^{2,1}$, thus

$$\Phi_{2,1}(\mathbf{u} \wedge \mathbf{v}, \mathbf{u} \wedge \mathbf{v}) = l_{01}^2 - l_{02}^2 - l_{12}^2.$$

With this definition, we reformulate Prop. 2 as:

$$\Phi_{2,1}(\mathbf{u} \wedge \mathbf{v}, \mathbf{u} \wedge \mathbf{v}) \text{ is } \left\{ \begin{array}{l} > 0 \\ = 0 \\ < 0 \end{array} \right\} \Leftrightarrow \epsilon \text{ is } \left\{ \begin{array}{l} \text{Euclidean} \\ \text{isotropic} \\ \text{pseudo-Euclidean} \end{array} \right\}.$$

According to [41], Th.2.2.4, there can not be an equivalent to the Plücker identity eq. (5.6) in this space.

5.2.2 Wedge product in 4-space

We want to define the wedge product $\mathbf{x} \wedge \mathbf{y}$ of two vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^4$ (or $\mathbb{R}^{3,1}$; as we saw in sec. 5.2.1 the differences between those two exterior algebras solely depend on the choice of basis), which is a vector in the $\binom{4}{2} = 6$ -dimensional vector space $\bigwedge^2 \mathbb{R}^4$.

Let $(\mathbf{e}_0, \mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3)$ be a basis of \mathbb{R}^4 , then a basis of $\bigwedge^2 \mathbb{R}^4$ is given as $(\mathbf{e}_0 \wedge \mathbf{e}_1, \mathbf{e}_0 \wedge \mathbf{e}_2, \mathbf{e}_0 \wedge \mathbf{e}_3, \mathbf{e}_1 \wedge \mathbf{e}_2, \mathbf{e}_1 \wedge \mathbf{e}_3, \mathbf{e}_2 \wedge \mathbf{e}_3)$. This linear space is the set of all 2-spaces (=planes through the origin) in 4-space.

Thus for two vectors $\mathbf{x} = (x_0, x_1, x_2, x_3)$, $\mathbf{y} = (y_0, y_1, y_2, y_3)$ the coefficients of the wedge product $\mathbf{x} \wedge \mathbf{y}$ are given by the numbers $l_{ij} = x_i y_j - x_j y_i$ for $i, j = 0, \dots, 3, i \neq j$, and the coordinates of the plane through the origin spanned by $\mathbf{x} \wedge \mathbf{y} \in \bigwedge^2 \mathbb{R}^4$ are $L = (l_{01}, l_{02}, l_{03}, l_{23}, l_{31}, l_{12})$. Note that a vector L satisfies the Plücker identity

$$\Omega_q(L) = l_{01}l_{23} + l_{02}l_{31} + l_{03}l_{12} = 0, \quad (5.6)$$

if and only if it represents a 2-plane in \mathbb{R}^4 ; see [41] and the use of this formalism in *line geometry*. We heavily rely on the fact that (projective) lines in projective three-space \mathcal{P}^3 are isomorphic to 2-planes through the origin in \mathbb{R}^4 .

The bilinear form $\Phi_{3,1}$

We want to calculate the equivalent of eq. (5.3), which relates inner and wedge product for $\bigwedge^2 \mathbb{R}^4$.

Lemma 3. *The Plücker coordinates of a plane $\mathbf{u} \wedge \mathbf{v}$ be l_{ij} for $\mathbf{u}, \mathbf{v} \in \mathbb{R}^{3,1}$ and those of $\mathbf{x} \wedge \mathbf{y}$ be m_{ij} . The following holds*

$$\det \begin{pmatrix} \langle \mathbf{u}, \mathbf{x} \rangle_L & \langle \mathbf{v}, \mathbf{x} \rangle_L \\ \langle \mathbf{u}, \mathbf{y} \rangle_L & \langle \mathbf{v}, \mathbf{y} \rangle_L \end{pmatrix} = l_{01}m_{01} + l_{02}m_{02} + l_{12}m_{12} - l_{03}m_{03} - l_{13}m_{13} - l_{23}m_{23}$$

Proof: Elementary calculation. □

In eq. (5.3) this determinant is equal to an inner product of two wedge products - this is only possible, because $\bigwedge^2 \mathbb{R}^3$ is isomorphic to \mathbb{R}^3 . Thus, we *define* a bilinear form $\Phi_{3,1} : \bigwedge^2 \mathbb{R}^{3,1} \times \bigwedge^2 \mathbb{R}^{3,1} \rightarrow \mathbb{R}$ by setting

$$\Phi_{3,1}(\mathbf{u} \wedge \mathbf{v}, \mathbf{x} \wedge \mathbf{y}) = l_{01}m_{01} + l_{02}m_{02} + l_{12}m_{12} - l_{03}m_{03} - l_{13}m_{13} - l_{23}m_{23},$$

where l_{ij} are the Plücker coordinates of a plane $\mathbf{u} \wedge \mathbf{v}$ and m_{ij} those of $\mathbf{x} \wedge \mathbf{y}$ as in Lemma 3.

Note that for $\mathbf{u} = \lambda \mathbf{v}$, $\lambda \in \mathbb{R}$, the determinant in Lemma 3 evaluates to $\langle \lambda \mathbf{u}, \lambda \mathbf{u} \rangle_L \cdot \langle \mathbf{v}, \mathbf{v} \rangle_L - \langle \lambda \mathbf{u}, \mathbf{v} \rangle_L^2 = 0$ and for $\mathbf{u} = \mathbf{o}$ holds $\mathbf{u} \wedge \mathbf{x} = \mathbf{o} \forall \mathbf{x}$, which is reminiscent of eq. (5.4) and the comment underneath.

Corollary 2. *For dimension $i = 2$ or 3 let $\mathbf{u}, \mathbf{v} \in \mathbb{R}^{i,1}$ be two vectors. Then*

$$\Phi_{i,1}(\mathbf{u} \wedge \mathbf{v}, \mathbf{u} \wedge \mathbf{v}) = \langle \mathbf{u}, \mathbf{u} \rangle_L \cdot \langle \mathbf{v}, \mathbf{v} \rangle_L - \langle \mathbf{u}, \mathbf{v} \rangle_L^2.$$

We saw on page 50 that this number determines the steepness of the planes in $\mathbb{R}^{2,1}$; in the next subsection, we want to extend this to $\mathbb{R}^{3,1}$.

5.2.3 Classification of 2-planes in Minkowski space

We have seen in sec. 5.2.1 that the pseudo-Euclidean cross product together with the inner product can be used to classify planes in $\mathbb{R}^{2,1}$. To carry these ideas to 4-space, we have introduced the wedge product in $\mathbb{R}^{3,1}$ and the bilinear form $\Phi_{3,1}$ on these products in sec. 5.2.2.

In this section we present a classification of 2-planes in $\mathbb{R}^{3,1}$ that fits consistently into this framework, and extends proposition 2 on the classification of planes in lower dimensional Minkowski space.

This important result is formulated as the following theorem:

Theorem 1. *The bilinear form $\Phi_{3,1}$ allows for classification of 2-planes in $\mathbb{R}^{3,1}$:*

1. $\mathbf{u} \wedge \mathbf{v}$ is an Euclidean plane $\Leftrightarrow \Phi_{3,1}(\mathbf{u} \wedge \mathbf{v}, \mathbf{u} \wedge \mathbf{v}) > 0 \Leftrightarrow l_{01}^2 + l_{02}^2 + l_{12}^2 > l_{03}^2 + l_{13}^2 + l_{23}^2$
2. $\mathbf{u} \wedge \mathbf{v}$ is an isotropic plane $\Leftrightarrow \Phi_{3,1}(\mathbf{u} \wedge \mathbf{v}, \mathbf{u} \wedge \mathbf{v}) = 0 \Leftrightarrow l_{01}^2 + l_{02}^2 + l_{12}^2 = l_{03}^2 + l_{13}^2 + l_{23}^2$.

3. $\mathbf{u} \wedge \mathbf{v}$ is a pseudo-Euclidean plane $\Leftrightarrow \Phi_{3,1}(\mathbf{u} \wedge \mathbf{v}, \mathbf{u} \wedge \mathbf{v}) < 0 \Leftrightarrow l_{01}^2 + l_{02}^2 + l_{12}^2 < l_{03}^2 + l_{13}^2 + l_{23}^2$

Proof. For a 2-plane spanned by $\mathbf{u}, \mathbf{v} \in \mathbb{R}^{3,1}$, any vector \mathbf{w} contained in $\mathbf{u} \wedge \mathbf{v}$ can be written as $\mathbf{w} = t \cdot \mathbf{u} + (1 - t) \cdot \mathbf{v}$, $t \in \mathbb{R}$. To check whether \mathbf{w} is Euclidean, pseudo-Euclidean or isotropic, we have to compute $\langle \mathbf{w}, \mathbf{w} \rangle_L$, which is a quadratic equation in t :

$$\langle \mathbf{w}, \mathbf{w} \rangle_L = \langle t \cdot \mathbf{u} + (1 - t) \cdot \mathbf{v}, t \cdot \mathbf{u} + (1 - t) \cdot \mathbf{v} \rangle_L, \quad (5.7)$$

whose discriminant is $\langle \mathbf{u}, \mathbf{v} \rangle_L^2 - \langle \mathbf{u}, \mathbf{u} \rangle_L \cdot \langle \mathbf{v}, \mathbf{v} \rangle_L = -\Phi_{3,1}$ by Cor. 2.

If and only if the discriminant of eq. (5.7) is negative, i.e. $\Phi_{3,1} > 0$ does $\mathbf{u} \wedge \mathbf{v}$ contain only Euclidean vectors \mathbf{w} , which characterizes Euclidean planes by Prop. 1.

If and only if $\Phi_{3,1} = 0$, there exists exactly one real root of eq. (5.7), i.e. exactly one isotropic vector in $\mathbf{u} \wedge \mathbf{v}$, which means $\mathbf{u} \wedge \mathbf{v}$ is isotropic.

Finally, only for $\Phi_{3,1} > 0$ there are two distinct roots of eq. (5.7), which correspond to isotropic vectors in $\mathbf{u} \wedge \mathbf{v}$. Therefore, $\langle \mathbf{w}, \mathbf{w} \rangle_L$ can be positive, negative or zero, thus $\mathbf{u} \wedge \mathbf{v}$ is pseudo-Euclidean by Prop. 1. \square

We will use this classification of 2-planes in sec. 5.4 to ensure that a tangent plane to a surface in $\mathbb{R}^{3,1}$ is Euclidean in an optimization routine.

5.3 Hyperbolic paraboloids in $\mathbb{R}^{3,1}$

The aim of this section is to establish a criterion which helps to classify hyperbolic paraboloids (also *HP surfaces*) in 4-dimensional Minkowski space $\mathbb{R}^{3,1}$ with respect to their *steepness*. In order to achieve this, we will first give a simple algebraic criterion to classify hyperplanes, then investigate the tangent planes of HP surfaces in sec. 5.3.1. After studying the situation at the plane at infinity, an actual algorithm will emerge, whose specifics we will compute in the final subsection 5.3.2.

Types of hyperplanes of $\mathbb{R}^{3,1}$

As was stated many times already, there are three types of hyperplanes in $\mathbb{R}^{3,1}$: Euclidean, isotropic and pseudo-Euclidean. Four points $\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d} \in \mathbb{R}^{3,1}$ in general position span such a hyperplane.

Via the wedge product notation one can define a "vector" (actually this object is an element of $\bigwedge^3 \mathbb{R}^{3,1}$, which we identify with $\mathbb{R}^{3,1}$ - see sections 5.2.1 and 5.2.2 for more detail) that serves as a normal vector to this hyperplane.

Choose one of the four points, for instance \mathbf{a} , and define the difference vectors $\mathbf{c} - \mathbf{a}, \mathbf{b} -$

$\mathbf{a}, \mathbf{d} - \mathbf{a}$, whose wedge product is

$$\begin{aligned}
& (\mathbf{c} - \mathbf{a}) \wedge (\mathbf{b} - \mathbf{a}) \wedge (\mathbf{d} - \mathbf{a}) \\
&= (\mathbf{c} \wedge \mathbf{b} - \mathbf{c} \wedge \mathbf{a} - \mathbf{a} \wedge \mathbf{b} + 0) \wedge (\mathbf{d} - \mathbf{a}) = \dots = \\
&= \mathbf{a} \wedge \mathbf{b} \wedge \mathbf{c} - \mathbf{a} \wedge \mathbf{b} \wedge \mathbf{d} + \mathbf{a} \wedge \mathbf{c} \wedge \mathbf{d} - \mathbf{b} \wedge \mathbf{c} \wedge \mathbf{d}.
\end{aligned} \tag{5.8}$$

This computation shows that the choice of \mathbf{a} has no effect on the outcome, ie. taking difference with any of the four points leads to the same result.

The coefficients of the "normal vector" in eq. (5.8) can be computed via the following determinant:

$$\begin{aligned}
& \begin{bmatrix} c_1 - a_1 & b_1 - a_1 & d_1 - a_1 & e_1 \\ c_2 - a_2 & b_2 - a_2 & d_2 - a_2 & e_2 \\ c_3 - a_3 & b_3 - a_3 & d_3 - a_3 & e_3 \\ c_4 - a_4 & b_4 - a_4 & d_4 - a_4 & e_4 \end{bmatrix} = \\
&= e_1 \cdot \begin{bmatrix} c_2 - a_2 & b_2 - a_2 & d_2 - a_2 \\ c_3 - a_3 & b_3 - a_3 & d_3 - a_3 \\ c_4 - a_4 & b_4 - a_4 & d_4 - a_4 \end{bmatrix} - e_2 \cdot \begin{bmatrix} c_1 - a_1 & b_1 - a_1 & d_1 - a_1 \\ c_3 - a_3 & b_3 - a_3 & d_3 - a_3 \\ c_4 - a_4 & b_4 - a_4 & d_4 - a_4 \end{bmatrix} \\
&+ e_3 \cdot \begin{bmatrix} c_1 - a_1 & b_1 - a_1 & d_1 - a_1 \\ c_2 - a_2 & b_2 - a_2 & d_2 - a_2 \\ c_4 - a_4 & b_4 - a_4 & d_4 - a_4 \end{bmatrix} - e_4 \cdot \begin{bmatrix} c_1 - a_1 & b_1 - a_1 & d_1 - a_1 \\ c_2 - a_2 & b_2 - a_2 & d_2 - a_2 \\ c_3 - a_3 & b_3 - a_3 & d_3 - a_3 \end{bmatrix} = \\
&= e_1 \cdot \lambda_1 + e_2 \cdot \lambda_2 + e_3 \cdot \lambda_3 + e_4 \cdot \lambda_4,
\end{aligned} \tag{5.9}$$

ie. the λ_i are the sub determinants (with sign) defined above and hence the four components of the normal vector of the hyperplane spanned by the four points $\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d} \in \mathbb{R}^{3,1}$. A simple calculation shows that eq. (5.9) is independent of the choice \mathbf{a} and actually leads to the same result as eq. (5.8).

Taking the classification of hyperplanes given in prop. 2 one dimension higher yields:

Proposition 3. *The hyperplane spanned by four points $\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d} \in \mathbb{R}^{3,1}$ in general position is Euclidean/isotropic/pseudo-Euclidean if and only if the number $\lambda_1^2 + \lambda_2^2 + \lambda_3^2 - \lambda_4^2$ is $< 0 / = 0 / > 0$.*

Proof: The construction of the normal vector was given above, the classification via the inner product is by definition. \square

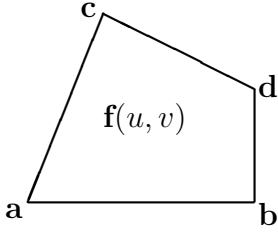
HP surfaces in different hyperplanes

Proposition 3 states which type of hyperplane four points $\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d} \in \mathbb{R}^{3,1}$ span. From Prop. 1 we know that only pseudo-Euclidean hyperplanes can contain pseudo-Euclidean

2-planes. It follows that a 2-surface in $\mathbb{R}^{3,1}$ can only have pseudo-Euclidean tangent planes if it is contained in a pseudo-Euclidean hyperplane.

This general statement is especially true for a hyperbolic paraboloid; it can only have pseudo-Euclidean tangent planes if its four corners $\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d} \in \mathbb{R}^{3,1}$ span a pseudo-Euclidean hyperplane.

5.3.1 Tangent planes of HP surfaces



Let $\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d} \in \mathbb{R}^{3,1}$ be points in general position and $u, v \in [0, 1]$ parameters, then the hyperbolic paraboloid is defined as

$$\mathbf{f}(u, v) = uv \cdot \mathbf{a} + u(1 - v) \cdot \mathbf{b} + (1 - u)v \cdot \mathbf{c} + (1 - u)(1 - v) \cdot \mathbf{d}. \quad (5.10)$$

If we take partial derivatives we get

$$\begin{aligned} \frac{\partial \mathbf{f}}{\partial v} &= u \cdot \mathbf{a} - u \cdot \mathbf{b} + (1 - u) \cdot \mathbf{c} - (1 - u) \cdot \mathbf{d} = u \cdot (\mathbf{a} - \mathbf{b}) + (1 - u) \cdot (\mathbf{c} - \mathbf{d}), \\ \frac{\partial \mathbf{f}}{\partial u} &= v \cdot \mathbf{a} + (1 - v) \cdot \mathbf{b} - v \cdot \mathbf{c} - (1 - v) \cdot \mathbf{d} = v \cdot (\mathbf{a} - \mathbf{c}) + (1 - v) \cdot (\mathbf{b} - \mathbf{d}). \end{aligned} \quad (5.11)$$

These partial derivatives can be seen as the directions of a general tangent plane of $\mathbf{f}(u, v)$.

According to sec. 5.2.2, if we want to define the tangent plane at position (u, v) , we have to take the wedge product of the vectors of eq. (5.11):

$$\begin{aligned} \frac{\partial \mathbf{f}}{\partial v} \wedge \frac{\partial \mathbf{f}}{\partial u} &= u \cdot (\mathbf{a} - \mathbf{b}) \wedge v \cdot (\mathbf{a} - \mathbf{c}) + u \cdot (\mathbf{a} - \mathbf{b}) \wedge (1 - v) \cdot (\mathbf{b} - \mathbf{d}) \\ &\quad + (1 - u) \cdot (\mathbf{c} - \mathbf{d}) \wedge (1 - v) \cdot (\mathbf{b} - \mathbf{d}) + (1 - u) \cdot (\mathbf{c} - \mathbf{d}) \wedge v \cdot (\mathbf{a} - \mathbf{c}) \\ &= uv \cdot (\mathbf{b} - \mathbf{a}) \wedge (\mathbf{c} - \mathbf{a}) - u(1 - v) \cdot (\mathbf{a} - \mathbf{b}) \wedge (\mathbf{d} - \mathbf{b}) \\ &\quad + (1 - u)(1 - v) \cdot (\mathbf{c} - \mathbf{d}) \wedge (\mathbf{b} - \mathbf{d}) - (1 - u)v \cdot (\mathbf{d} - \mathbf{c}) \wedge (\mathbf{a} - \mathbf{c}) \\ &= uv \cdot (\mathbf{b} - \mathbf{a}) \wedge (\mathbf{c} - \mathbf{a}) + u(1 - v) \cdot (\mathbf{d} - \mathbf{b}) \wedge (\mathbf{a} - \mathbf{b}) \\ &\quad + (1 - u)(1 - v) \cdot (\mathbf{c} - \mathbf{d}) \wedge (\mathbf{b} - \mathbf{d}) + (1 - u)v \cdot (\mathbf{a} - \mathbf{c}) \wedge (\mathbf{d} - \mathbf{c}) \\ &= uv \cdot \mathbf{T}(\mathbf{a}) + u(1 - v) \cdot \mathbf{T}(\mathbf{b}) + (1 - u)v \cdot \mathbf{T}(\mathbf{c}) + (1 - u)(1 - v) \cdot \mathbf{T}(\mathbf{d}), \end{aligned} \quad (5.12)$$

where $\mathbf{T}(\mathbf{a}) := (\mathbf{b} - \mathbf{a}) \wedge (\mathbf{c} - \mathbf{a})$ is the tangent plane at corner \mathbf{a} and analogously for the other corners. We also write $\mathbf{T}(\mathbf{f}) := \partial \mathbf{f} / \partial v \wedge \partial \mathbf{f} / \partial u$.

Hence, for a bilinear Bézier surface $\mathbf{f}(u, v)$ as given in (5.10), the general tangent plane $\mathbf{T}(\mathbf{f})(u, v)$ is a bilinear combination of the tangent planes $\mathbf{T}(\mathbf{a}), \mathbf{T}(\mathbf{b}), \mathbf{T}(\mathbf{c})$ and $\mathbf{T}(\mathbf{d})$ as stated in eq. (5.12). Note that these tangent planes are not linearly independent, but span a 3-space:

$$\mathbf{T}(\mathbf{a}) - \mathbf{T}(\mathbf{b}) + \mathbf{T}(\mathbf{c}) - \mathbf{T}(\mathbf{d}) = 0,$$

as can be checked by an elementary calculation.

Situation at infinity

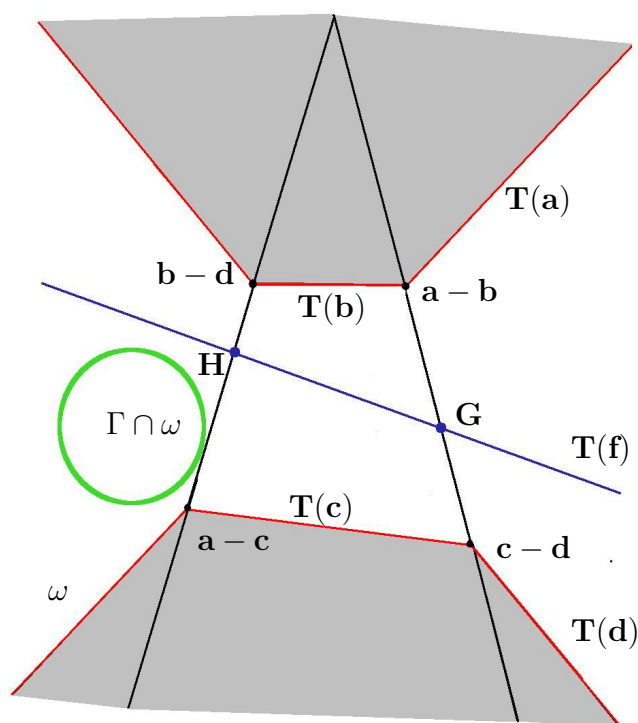


Figure 5.1: At the plane at infinity ω , the directions of the HP surface's edges are points and the planes they span are represented by lines. A general tangent plane $\mathbf{T}(\mathbf{f})$ of the HP surface can be visualized as a line connecting two points $\mathbf{G} \cap \omega$ and $\mathbf{H} \cap \omega$. This plane $\mathbf{T}(\mathbf{f})$ is pseudo-Euclidean, if the line $\mathbf{T}(\mathbf{f}) \cap \omega$ intersects $\Gamma \cap \omega$ for any $v \in [0, 1], u \in [0, 1]$. Note that for brevity, there is a slight misuse of notation in this sketch.

In order to understand the criterion we want to give in Prop. 4 for a general tangent plane $\mathbf{T}(\mathbf{f})$ of a HP surface to be pseudo-Euclidean, we make a sketch of the situation at infinity, see Fig. 5.1. We will denote the ideal plane of the carrier 3-space of \mathbf{f} by ω .

The directions of the HP's edges are $(\mathbf{a} - \mathbf{c}) \cap \omega$, $(\mathbf{c} - \mathbf{d}) \cap \omega$, $(\mathbf{a} - \mathbf{b}) \cap \omega$ and $(\mathbf{b} - \mathbf{d}) \cap \omega$ and thus the intersection of tangent plane $\mathbf{T}(\mathbf{a})$ and the plane at infinity ω is a line at infinity connecting the points $(\mathbf{a} - \mathbf{b}) \cap \omega$ and $(\mathbf{a} - \mathbf{c}) \cap \omega$ (For brevity of notation, this line is simply called $\mathbf{T}(\mathbf{a})$ in Fig. 5.1 and the segment from $(\mathbf{a} - \mathbf{b}) \cap \omega$ and $(\mathbf{a} - \mathbf{c}) \cap \omega$ is not drawn in order to keep the figure simple.).

The intersection of the quadratic cone Γ and the plane at infinity ω is a conic. By construction, we assume that the hyperplane in $\mathbb{R}^{3,1}$ spanned by \mathbf{a} , \mathbf{b} , \mathbf{c} and \mathbf{d} is pseudo-Euclidean. We also assume that the difference vectors $\mathbf{a} - \mathbf{c}$, $\mathbf{c} - \mathbf{d}$, $\mathbf{a} - \mathbf{b}$ and $\mathbf{b} - \mathbf{d}$ are outside $\Gamma \cap \omega$, hence Euclidean. The main question is: How can we ensure that the hyperbolic paraboloid with these four corners points has only Euclidean tangent planes on the *inside* patch?

We have seen in eq. (5.11) that the partial derivatives $\frac{\partial \mathbf{f}}{\partial u}$ and $\frac{\partial \mathbf{f}}{\partial v}$ (their intersections with ω are denoted $\mathbf{G} = u(\mathbf{a} - \mathbf{b}) + (1 - u)(\mathbf{c} - \mathbf{d})$ and $\mathbf{H} = v(\mathbf{a} - \mathbf{c}) + (1 - v)(\mathbf{b} - \mathbf{d})$ in Fig. 5.1 respectively) are convex combinations of two difference vectors, so they are points on a line segment at infinity. Therefore, a general tangent plane $\mathbf{T}(\mathbf{f})$ of the HP surface, which is spanned by $\frac{\partial \mathbf{f}}{\partial u}$ and $\frac{\partial \mathbf{f}}{\partial v}$, intersects ω along the line connecting \mathbf{H} and \mathbf{G} . $\mathbf{T}(\mathbf{f})$ is Euclidean/isotropic/pseudo-Euclidean if and only if $\mathbf{T}(\mathbf{f}) \cap \omega$ have zero/one/two (real) intersections with $\Gamma \cap \omega$.

Note that the sketch in Fig. 5.1 depicts only one of the possible configurations of $\mathbf{a} - \mathbf{c}$, $\mathbf{c} - \mathbf{d}$, $\mathbf{a} - \mathbf{b}$ and $\mathbf{b} - \mathbf{d}$; their relative positions could change, e.g. $\mathbf{a} - \mathbf{b}$ and $\mathbf{b} - \mathbf{d}$ could change position, which would not change much about the figure. One could also remark that statements such as "G is always *between* $\mathbf{a} - \mathbf{b}$ and $\mathbf{c} - \mathbf{d}$ " are meaningless in projective geometry and that one would have to check different possible combinations of \mathbf{G} and \mathbf{H} as well. Just keep in mind that Fig. 5.1 is a motivation for Prop. 4 and observe the sound proof by algebra there.

A necessary condition for $\mathbf{T}(\mathbf{f})$ to be pseudo-Euclidean

Looking at Fig. 5.1, the tangent planes $\mathbf{T}(\mathbf{a})$, $\mathbf{T}(\mathbf{b})$ and $\mathbf{T}(\mathbf{d})$ enclose a convex area (the gray area in the upper half of the figure). Similarly, the tangent planes $\mathbf{T}(\mathbf{a})$, $\mathbf{T}(\mathbf{c})$ and $\mathbf{T}(\mathbf{d})$ enclose another convex area (the gray area in the lower half of the figure). If $\Gamma \cap \omega$ would lie inside either of these convex areas, there could not be a real intersection with $\mathbf{T}(\mathbf{f}) \cap \omega$, because \mathbf{H} is always between $\mathbf{a} - \mathbf{c}$ and $\mathbf{b} - \mathbf{d}$ and \mathbf{G} is always between $\mathbf{a} - \mathbf{b}$ and $\mathbf{c} - \mathbf{d}$, as we are only interested in the surface patch $\mathbf{f}(u, v)$ for which $u, v \in [0, 1] \times [0, 1]$.

Proposition 4. *A general tangent $\mathbf{T}(\mathbf{f})$ of a HP surface in $\mathbb{R}^{3,1}$ is pseudo-Euclidean if at least one of the following is true:*

1. *At least one plane in the pencil of planes spanned by $\mathbf{c} - \mathbf{d}$ and \mathbf{H} is pseudo-Euclidean.*
2. *At least one plane in the pencil of planes spanned by $\mathbf{a} - \mathbf{b}$ and \mathbf{H} is pseudo-Euclidean.*
3. *At least one plane in the pencil of planes spanned by $\mathbf{b} - \mathbf{d}$ and \mathbf{G} , $u \in [0, 1]$ is pseudo-Euclidean.*

4. At least one plane in the pencil of planes spanned by $\mathbf{a}-\mathbf{c}$ and \mathbf{G} is pseudo-Euclidean, where $\mathbf{H} = v(\mathbf{a}-\mathbf{c}) + (1-v)(\mathbf{b}-\mathbf{d})$, $v \in [0, 1]$ and $\mathbf{G} = u(\mathbf{a}-\mathbf{b}) + (1-u)(\mathbf{c}-\mathbf{d})$, $u \in [0, 1]$.

Proof: 1st proof by projective geometry:

The four cases listed above are precisely the four pencils of lines from the corners of the (white) non-convex area of the plane at infinity ω to the opposite line segment, see Fig. 5.1. For each point in this non-convex area there is at least one line of these pencils intersecting it.

2nd proof by algebra:

As was stated in sec.5.3.1, the general tangent plane $\mathbf{T}(\mathbf{f})$ is a bilinear combination of the four tangent planes in the corners. The four cases above mean setting $u = 0, u = 1, v = 0, v = 1$ in eq. 5.12. The first case can hence be rephrased to "At least one of the tangent planes to the HP surface along the edge from \mathbf{c} to \mathbf{d} , ie. at least one of the convex combinations of $\mathbf{T}(\mathbf{c})$ and $\mathbf{T}(\mathbf{d})$ is pseudo-Euclidean." and the others accordingly.

We have thus used special properties of the tangent planes of a hyperbolic paraboloid to look at four linear combinations instead of one bilinear one. \square

5.3.2 Computing $\Phi_{3,1}$ for HP surfaces in $\mathbb{R}^{3,1}$

To see whether a general tangent plane is pseudo-Euclidean, we have to compute

$$\begin{aligned} \Phi_{3,1}\left(\frac{\partial \mathbf{f}}{\partial v} \wedge \frac{\partial \mathbf{f}}{\partial u}, \frac{\partial \mathbf{f}}{\partial v} \wedge \frac{\partial \mathbf{f}}{\partial u}\right) &= \\ &= \Phi_{3,1}(uv \cdot (\mathbf{b}-\mathbf{a}) \wedge (\mathbf{c}-\mathbf{a}) + u(1-v) \cdot (\mathbf{d}-\mathbf{b}) \wedge (\mathbf{a}-\mathbf{b}) \\ &+ (1-u)(1-v) \cdot (\mathbf{c}-\mathbf{d}) \wedge (\mathbf{b}-\mathbf{d}) + (1-u)v \cdot (\mathbf{a}-\mathbf{c}) \wedge (\mathbf{d}-\mathbf{c}), \dots), \end{aligned}$$

but we know from Prop. 4 that it suffices to set u and v to 0 and 1. Hence the resulting quadratic polynomial in v resp. u become

$$\begin{aligned} u = 0 : \quad & \Phi_{3,1}((1-v) \cdot (\mathbf{c}-\mathbf{d}) \wedge (\mathbf{b}-\mathbf{d}) + v \cdot (\mathbf{a}-\mathbf{c}) \wedge (\mathbf{d}-\mathbf{c}), \dots) = \\ & = \Phi_{3,1}((\mathbf{c}-\mathbf{d}) \wedge ((1-v) \cdot (\mathbf{b}-\mathbf{d}) + v \cdot (\mathbf{a}-\mathbf{c})), \dots) \\ u = 1 : \quad & \Phi_{3,1}(v \cdot (\mathbf{b}-\mathbf{a}) \wedge (\mathbf{c}-\mathbf{a}) + (1-v) \cdot (\mathbf{d}-\mathbf{b}) \wedge (\mathbf{a}-\mathbf{b}) = \\ & = \Phi_{3,1}((\mathbf{b}-\mathbf{a}) \wedge (v \cdot (\mathbf{c}-\mathbf{a}) + (1-v) \cdot (\mathbf{d}-\mathbf{b})), \dots) \\ v = 0 : \quad & \Phi_{3,1}(u \cdot (\mathbf{d}-\mathbf{b}) \wedge (\mathbf{a}-\mathbf{b}) + (1-u) \cdot (\mathbf{c}-\mathbf{d}) \wedge (\mathbf{b}-\mathbf{d}) = \\ & = \Phi_{3,1}((\mathbf{d}-\mathbf{b}) \wedge (u \cdot (\mathbf{a}-\mathbf{b}) + (1-u) \cdot (\mathbf{c}-\mathbf{d})), \dots) \\ v = 1 : \quad & \Phi_{3,1}(u \cdot (\mathbf{b}-\mathbf{a}) \wedge (\mathbf{c}-\mathbf{a}) + (1-u) \cdot (\mathbf{a}-\mathbf{c}) \wedge (\mathbf{d}-\mathbf{c}), \dots) = \\ & = \Phi_{3,1}((\mathbf{a}-\mathbf{c}) \wedge (u \cdot (\mathbf{b}-\mathbf{a}) + (1-u) \cdot (\mathbf{d}-\mathbf{c})), \dots). \end{aligned}$$

If we substitute according to the following scheme:

	x	y	z	
u = 0	c - d	b - d	a - c	
u = 1	a - b	b - d	a - c	
v = 0	b - d	a - b	c - d	
v = 1	a - c	a - b	c - d	

(5.13)

these four quadratic equations are all of the general typology

$$\begin{aligned}
& \Phi_{3,1}(\mathbf{x} \wedge ((1-t) \cdot \mathbf{y} + t \cdot \mathbf{z}), \mathbf{x} \wedge ((1-t) \cdot \mathbf{y} + t \cdot \mathbf{z})) = \dots = \\
& = (1-t)^2 \Phi_{3,1}(\mathbf{x} \wedge \mathbf{y}, \mathbf{x} \wedge \mathbf{y}) + t^2 \Phi_{3,1}(\mathbf{x} \wedge \mathbf{z}, \mathbf{x} \wedge \mathbf{z}) \\
& + 2t(1-t) \underbrace{(\langle \mathbf{x}, \mathbf{x} \rangle_L \langle \mathbf{y}, \mathbf{z} \rangle_L - \langle \mathbf{x}, \mathbf{y} \rangle_L \langle \mathbf{x}, \mathbf{z} \rangle_L)}_{=: \mathbf{e}}.
\end{aligned}$$

Therefore their solution for t is always

$$t_{1,2} = \frac{\Phi_{3,1}(\mathbf{x} \wedge \mathbf{y}, \mathbf{x} \wedge \mathbf{y}) - \mathbf{e} \pm \sqrt{\Delta}}{\Phi_{3,1}(\mathbf{x} \wedge \mathbf{y}, \mathbf{x} \wedge \mathbf{y}) + \Phi_{3,1}(\mathbf{x} \wedge \mathbf{z}, \mathbf{x} \wedge \mathbf{z}) - 2\mathbf{e}},$$

and the discriminant is

$$\Delta = (\langle \mathbf{x}, \mathbf{x} \rangle_L \langle \mathbf{y}, \mathbf{z} \rangle_L - \langle \mathbf{x}, \mathbf{y} \rangle_L \langle \mathbf{x}, \mathbf{z} \rangle_L)^2 - \Phi_{3,1}(\mathbf{x} \wedge \mathbf{y}, \mathbf{x} \wedge \mathbf{y}) \cdot \Phi_{3,1}(\mathbf{x} \wedge \mathbf{z}, \mathbf{x} \wedge \mathbf{z}). \quad (5.14)$$

Remember that $\Phi_{3,1}(\mathbf{x} \wedge \mathbf{y}, \mathbf{x} \wedge \mathbf{y})$ and $\Phi_{3,1}(\mathbf{x} \wedge \mathbf{z}, \mathbf{x} \wedge \mathbf{z})$ are the expressions for the steepness of the tangent planes at the two corners of the edge on which we look for a steep tangent plane, so these are known values and by assumption > 0 .

In conclusion, we have reduced the problem of finding a steep tangent plane for a general point on an HP surface to calculating the four numbers that we get from combining eqs. (5.13) and (5.14). We just have to check whether any of these four numbers Δ is bigger than zero in order to know that there is in fact such a steep plane $\mathbf{T}(\mathbf{f})$.

5.4 Formulating the surface optimization in $\mathbb{R}^{3,1}$

In chapter 4 on the optimization of curves in $\mathbb{R}^{2,1}$ and in section 5.1 on curves in $\mathbb{R}^{3,1}$ we were concerned with the optimization of envelopes for one-parameter families of circles or spheres.

For the rest of this chapter, we want to turn our attention to surfaces in $\mathbb{R}^{3,1}$, e.g. the optimization of envelopes for 2-parameter families of spheres.

Problem Statement for Surfaces

The optimization problem thus becomes

$$\text{minimize } \sum_{i=1}^k |Q_i - \mathbf{f}(u, v)|^2 \quad (5.15a)$$

$$\text{subject to } \zeta^{-1}(\mathbf{f}(u, v)) \text{ is real,} \quad (5.15b)$$

and the distance in the objective function (5.15a) is the *Euclidean* distance, rather than the distance based on the *Minkowski norm* of eq. (2.3). In the latter case we would run into all sorts of problems with isotropic directions as described in sec. 5.1.2, which we want to avoid altogether. We give a matrix formulation for the objective function sec. 5.4.1.

The constraint (5.15b) means that the inverse cyclographic image of the surface parametrization in \mathbb{R}^3 stays real, or, equivalently, that the tangent planes of $\mathbf{f}(u, v) \subset \mathbb{R}^{3,1}$ stay Euclidean. The formulation of this constraint is more involved than in the curve case and we will explain it in several steps in sec. 5.4.3.

Computation and parametrization

We have introduced a formalism in sec. 5.2 that allows us to check whether a 2-plane in $\mathbb{R}^{3,1}$ is pseudo-Euclidean, isotropic or Euclidean; here we want to apply it to tangent planes of a surface. We must therefore choose a parametrization $\mathbf{f}(u, v)$ and for practical reasons we restrict our attention to tensor product B-spline surfaces. More on the computational details for such surfaces can be found in sec. 5.4.2.

Handling the constraint - 3 levels of rigidity

The tangent planes of $\mathbf{f}(u, v)$ are spanned by the partial derivatives \mathbf{f}_u and \mathbf{f}_v . We give the specifics in sec. 5.4.3, where we present matrices in eq. (5.18) that linearizes this constraint.

For three different classes of surfaces we will present three levels of rigidity for the steepness constraint:

1. The most general class of surfaces we will look at are *bicubic tensor product B-spline surfaces* $\mathbf{f}(u, v)$, ie. the parameter lines in u - and v -direction are all cubic B-spline curves. On these, we use a sampling of the surface and can only guarantee that the tangent planes are Euclidean at each sampling point, which means that the envelope of a 2-parameter family of spheres $\zeta^{-1}(\mathbf{f}(u, v))$ is real at a discrete number of points. This general approach, which is the foundation of all three, will be explained in sec. 5.4.3.
2. If we consider tensor product B-splines surfaces of bidegree $(3, 1)$, ie. strips of *ruled surfaces*, one direction of parameter lines, say u -lines, will be cubic B-splines and the other direction linear B-splines, ie. polylines. We will sample the cubic u -parameter

lines and given that they are Euclidean (on a sampling), the tangent planes along the rulings (= the v -parameter lines) can be guaranteed to be Euclidean - without the use of a sampling - by an algorithm presented in sec. 5.4.4. The theoretical background for this algorithm was presented in sec. 5.2.

3. If both directions are linear, the single patches of the tensor product surface are *hyperbolic paraboloids* in $\mathbb{R}^{3,1}$, which were investigated in sec. 5.3. For this class of surfaces, we can even guarantee Euclidean surfaces without the need of a sampling, as we will see in sec. 5.4.5.

5.4.1 Objective function and collocation matrices

We are ready to state a matrix formulation of the objective function. Remember that we approximate a set of points $Q_k \in \mathbb{R}^{3,1}$ by a tensor product B-spline surface.

We have already extended the objective function (4.1) to $\mathbb{R}^{3,1}$ for curves in sec. 5.1.2, now we want to expand the surface analogue (5.15a) into matrix notation as

$$\begin{aligned}
& (\mathbf{f}^T - \mathbf{q}^T) \cdot (\mathbf{f} - \mathbf{q}) \rightarrow \min \\
& = (\mathbf{b}^T \cdot (\text{blkdiag}(\mathbf{M}^T, \dots, \mathbf{M}^T) - \mathbf{q}^T) \cdot \text{blkdiag}(\mathbf{M}, \dots, \mathbf{M}) \cdot \mathbf{b} - \mathbf{q}) \\
& = \mathbf{b}^T \cdot \underbrace{\begin{bmatrix} \mathbf{M}^T \mathbf{M} & & & \\ & \mathbf{M}^T \mathbf{M} & & \\ & & \mathbf{M}^T \mathbf{M} & \\ & & & \mathbf{M}^T \mathbf{M} \end{bmatrix}}_{=: \mathbf{G}} \cdot \mathbf{b} - \\
& \quad - 2 \cdot \mathbf{b}^T \cdot \underbrace{\text{blkdiag}(\mathbf{M}^T, \mathbf{M}^T, \mathbf{M}^T, \mathbf{M}^T)}_{=: \mathbf{e}} \cdot \mathbf{q} + \mathbf{q}^T \cdot \mathbf{q},
\end{aligned} \tag{5.16}$$

where *blkdiag* means a block diagonal matrix like in the *Hessian* \mathbf{G} that is very sparse, especially because the blocks themselves consist of sparse matrices. It follows that this matrix has to be *Tikhonov* regularized, precisely as we have seen in sec. 4.4.

5.4.2 Notation for B-spline surfaces

Tensor product B-spline approximation was briefly introduced in sec. 3.1.3, our matrix formulation for surfaces (the 2-parameter analogue of (4.3)) in $\mathbb{R}^{3,1}$ needs some explanation.

Assume control points to be given on a regular m by n grid in u and v -direction respectively. Thus the matrix \mathbf{B}_x (Farin [10] calls it the *geometry matrix*) containing all the x -coordinates

is $n \times m$ and the x -coordinates of the evaluated surface are given as

$$\mathbf{F}_x(u, v) = \mathbf{N}_v \cdot \mathbf{B}_x \cdot \mathbf{N}_u^T, \quad (5.17)$$

and equivalently for the other coordinates $\mathbf{F}_y, \mathbf{F}_z, \mathbf{F}_r$. \mathbf{N}_u and \mathbf{N}_v are the collocation matrices in u - and v -direction, whose number of rows is k . To clarify, the first row of control points in the regular grid are $[\mathbf{B}_x(1, 1), \mathbf{B}_y(1, 1), \mathbf{B}_z(1, 1), \mathbf{B}_r(1, 1)] \in \mathbb{R}^{3,1}$ to $[\mathbf{B}_x(1, n), \mathbf{B}_y(1, n), \mathbf{B}_z(1, n), \mathbf{B}_r(1, n)] \in \mathbb{R}^{3,1}$ and the k th row of \mathbf{N}_u and \mathbf{N}_v are the B-spline basis functions (3.1) in u - and v -direction of the k th point.

The x -coordinates of the partial derivatives in u and v -direction, respectively, are then given by

$$\begin{aligned} \frac{\partial}{\partial u} \mathbf{F}_x(u, v) &= \mathbf{N}_v \cdot \mathbf{B}_x \cdot \frac{\partial}{\partial u} \mathbf{N}_u^T, \\ \frac{\partial}{\partial v} \mathbf{F}_x(u, v) &= \frac{\partial}{\partial v} \mathbf{N}_v \cdot \mathbf{B}_x \cdot \mathbf{N}_u^T, \end{aligned}$$

with $\frac{\partial}{\partial u} \mathbf{N}_u$ and $\frac{\partial}{\partial v} \mathbf{N}_v$ the collocation matrices of the first partial derivatives, i.e. the matrices collecting the derivatives of the basis functions in u - and v -direction and analogously for the other coordinates.

2-parameter collocation matrices

If we reshape the matrix \mathbf{B}_x^T into a column vector \mathbf{b}_x and \mathbf{F}_x^T into a column vector \mathbf{f}_x , we can rewrite (5.17) as $\mathbf{f}_x = \mathbf{M} \cdot \mathbf{b}_x$, with

$$\begin{aligned} \mathbf{M} &= \begin{pmatrix} \mathbf{N}_v(1, 1)\mathbf{N}_u(1, 1) & \dots & \mathbf{N}_v(1, 1)\mathbf{N}_u(1, m) & \mathbf{N}_v(1, 2)\mathbf{N}_u(1, m) & \dots & \mathbf{N}_v(1, n)\mathbf{N}_u(1, m) \\ \vdots & & & & \vdots & \\ \mathbf{N}_v(k, 1)\mathbf{N}_u(k, 1) & \dots & \mathbf{N}_v(k, 1)\mathbf{N}_u(k, m) & \mathbf{N}_v(k, 2)\mathbf{N}_u(k, m) & \dots & \mathbf{N}_v(k, n)\mathbf{N}_u(k, m) \end{pmatrix} \\ &= [\text{diag}(\mathbf{N}_v(1 \dots k, 1)) \cdot \mathbf{N}_u \mid \dots \mid \text{diag}(\mathbf{N}_v(1 \dots k, n)) \cdot \mathbf{N}_u], \end{aligned}$$

where $\mathbf{N}_u(i, j)$ means the matrix' entry in row i , column j . Therefore, \mathbf{M} is the 2-parameter analogue of the curve's collocation matrix \mathbf{N} , c.f. Farin [10]. We also define the collocation matrices of the partial derivatives

$$\begin{aligned} \mathbf{M}_{du} &:= [\text{diag}(\mathbf{N}_v(1 \dots k, 1)) \cdot \frac{\partial}{\partial u} \mathbf{N}_u \mid \dots \mid \text{diag}(\mathbf{N}_v(1 \dots k, n)) \cdot \frac{\partial}{\partial u} \mathbf{N}_u], \\ \mathbf{M}_{dv} &:= [\text{diag}(\frac{\partial}{\partial v} \mathbf{N}_v(1 \dots k, 1)) \cdot \mathbf{N}_u \mid \dots \mid \text{diag}(\frac{\partial}{\partial v} \mathbf{N}_v(1 \dots k, n)) \cdot \mathbf{N}_u]. \end{aligned}$$

5.4.3 Constraints for surface optimization

Analogously to sec. 4.3, we have to give a linearized version of the quadratic constraint (5.15b), which in the surface case is actually slightly more involved. The constraint should

be, that for a surface $\mathbf{f}(u, v)$ its tangent planes stay Euclidean everywhere. We will sample the parameter lines to check the steepness of the partial derivatives.

We describe this constraint in the following steps:

1. We ensure Euclidean tangents in the direction of the parameter lines, i.e. the constraint shall be fulfilled for a sampling of the partial derivatives $\mathbf{f}_u(u_i, v_j)$ and $\mathbf{f}_v(u_i, v_j)$.
2. If both $\mathbf{f}_u(u_i, v_j)$ and $\mathbf{f}_v(u_i, v_j)$ are Euclidean, we test if the tangent plane at $\mathbf{f}(u_i, v_j)$ is Euclidean by computing $\Phi_{3,1}(\mathbf{f}_u \wedge \mathbf{f}_v, \mathbf{f}_u \wedge \mathbf{f}_v)(u_i, v_j)$. We give a matrix formulation for constraining the tangent plane of a surface in $\mathbb{R}^{3,1}$ to be Euclidean.

A summary of this algorithm can be found at the end of this section on page 63.

If we restrict our algorithm to only use strips of ruled surfaces, i.e. tensor product B-spline surfaces of bidegree $(3, 1)$, we can even guarantee more. For the purposes of this derivation, we can assume that the parameter lines in u -direction are cubic curves and those in v -direction are straight line segments; obviously, the sampling in u -direction should be denser in that case.

3. We extend the constraint along the ruling between $\mathbf{f}(u_i, v_j)$ and $\mathbf{f}(u_i, v_{j+1})$ and can thus constrain the ruled surface between $\mathbf{f}(u, v_j)$ and $\mathbf{f}(u, v_{j+1})$, $u \in [0, 1]$ to be Euclidean (given that the cubic curves are Euclidean). More on that in section 5.4.4 and especially in the algorithm on page 65.

For hyperbolic paraboloids (HP) in $\mathbb{R}^{3,1}$, we will employ a similar algorithm that is described in sec. 5.4.5. For that, we only need a sampling of $\mathbf{f}(u, v)$ at the intersections of the parameter lines, thus if $\mathbf{f}(u, v)$ would consist of a single HP surface, we would need only the values at the four corners.

The constraint on parameter lines

We assume $\mathbf{f}(u, v)$ is parametrized as a bicubic tensor product B-spline surface, hence the parameter lines in u and v -direction are cubic B-spline curves. We will not go into the problem of finding such a parametrization for a pointcloud in $\mathbb{R}^{3,1}$.

Therefore, constraining them to have only Euclidean tangents is exactly constraint (5.2), i.e. the constraint for a curve in $\mathbb{R}^{3,1}$ to have only Euclidean tangents.

The difference to the constraint (5.2) for curves in $\mathbb{R}^{3,1}$ is that for parameter lines on a surface, we need the definitions of sec. 5.4.2, where we explained among others the collocation matrix of the partial derivatives \mathbf{M}_{du} and \mathbf{M}_{dv} .

Then, for k sample points on the surface $\mathbf{f}(u, v)$, the $k \times 4$ -matrix of partial derivatives w.r.t. u is given as $\mathbf{f}_u = \mathbf{M}_{du} \cdot \mathbf{b}$ and we define $\bar{\mathbf{f}}_u$ to be the projection - again a $k \times 4$ -matrix

- of \mathbf{f}_u onto Γ as defined in sec. 4.3.2 and carried over to $\mathbb{R}^{3,1}$ in sec. 5.1.3. Then

$$\mathbf{D}_u := [\text{diag}(\bar{\mathbf{f}}_u(1 \dots k, 1)) \cdot \mathbf{M}_{du} | \dots | \text{diag}(\bar{\mathbf{f}}_u(1 \dots k, 3)) \cdot \mathbf{M}_{du} | - \text{diag}(\bar{\mathbf{f}}_u(1 \dots k, 4)) \cdot \mathbf{M}_{du}] \quad (5.18a)$$

$$\mathbf{D}_v := [\text{diag}(\bar{\mathbf{f}}_v(1 \dots k, 1)) \cdot \mathbf{M}_{dv} | \dots | \text{diag}(\bar{\mathbf{f}}_v(1 \dots k, 3)) \cdot \mathbf{M}_{dv} | - \text{diag}(\bar{\mathbf{f}}_v(1 \dots k, 4)) \cdot \mathbf{M}_{dv}] \quad (5.18b)$$

and $\mathbf{D}_u \cdot [\mathbf{b}_x^T | \mathbf{b}_y^T | \mathbf{b}_z^T | \mathbf{b}_w^T]^T = \langle \mathbf{f}_u, \bar{\mathbf{f}}_u \rangle_L$ is a vector, whose k th entry is less than zero if and only if $\mathbf{f}_u(u_k, v_k)$ is pseudo-Euclidean.

The matrix \mathbf{D}_u thus linearizes the quadratic constraint that the angle between the partial derivatives w.r.t u , $\mathbf{f}_u(u_k, v_k)$, and the hyperplane $x_4 = 0$ is less than $\frac{\pi}{4}$ for a sampling of all parameter lines in u -direction. Analogous statements hold for \mathbf{D}_v and $\mathbf{f}_v(u_k, v_k)$.

The constraint for a discrete sampling of the surface

Analogously to what we have seen in sec. 4.4.1 on the KKT conditions, $\mathbf{D}_u \cdot [\mathbf{b}_x^T | \mathbf{b}_y^T | \mathbf{b}_z^T | \mathbf{b}_w^T]^T$ is precisely the vector of *slack variables* \mathbf{y}_u and $\mathbf{D}_v \cdot [\mathbf{b}_x^T | \mathbf{b}_y^T | \mathbf{b}_z^T | \mathbf{b}_w^T]^T = \mathbf{y}_v$. We already know that tangent planes at points $\mathbf{f}(u_i, v_j)$ for which either entry $\mathbf{y}_{u,i}$ or $\mathbf{y}_{v,j}$ is ≤ 0 cannot be Euclidean; the optimization algorithm will take care of those, we can concentrate on the others.

Knowing that the partial derivatives $\mathbf{f}_u(u_i, v_j)$ and $\mathbf{f}_v(u_i, v_j)$ are Euclidean does not necessarily mean that the tangent plane at $\mathbf{f}(u_i, v_j)$ is Euclidean. We thus employ the calculus summarized in Theorem 1: Compute $\Phi_{3,1}(\mathbf{f}_u \wedge \mathbf{f}_v, \mathbf{f}_u \wedge \mathbf{f}_v)(u_i, v_j)$; if it is ≤ 0 , the tangent plane at $\mathbf{f}_v(u_i, v_j)$ is isotropic or pseudo-Euclidean, even though $\mathbf{y}_{u,i} > 0$ and $\mathbf{y}_{v,j} > 0$ by assumption. We hence change the sign of $\Delta \mathbf{y}_{v,j}$ and thus activate this constraint. We could have done the same to $\Delta \mathbf{y}_{u,i}$, but we want to reserve this possibility for a similar idea with ruled surfaces, see sec. 5.4.4.

Algorithm for constraining the tangent planes of a surface

Let us summarize the ideas of this section as an algorithm:

1. For a sampling of a B-spline surface $\mathbf{f}(u_i, v_j)$, compute the matrices \mathbf{D}_u and \mathbf{D}_v of the linearized constraint (5.18).
2. For index pairs (\bar{i}, \bar{j}) for which the slack variables $\mathbf{y}_{u,\bar{i}} > 0$ and $\mathbf{y}_{v,\bar{j}} > 0$, compute the bilinear form $\Phi_{3,1}(\mathbf{f}_u \wedge \mathbf{f}_v, \mathbf{f}_u \wedge \mathbf{f}_v)(u_{\bar{i}}, v_{\bar{j}})$.
3. For those (\bar{i}, \bar{j}) for which $\Phi_{3,1} \leq 0$, change the sign of $\Delta \mathbf{y}_{v,\bar{j}}$ to be minus.

With the simple trick in 3., we have activated the constraint with the help of $\Phi_{3,1}$, even though both partial derivatives are Euclidean. This way, isotropic or pseudo-Euclidean tangent planes enter the optimization.

In summary, we have given a linearization of the quadratic constraint (5.15b) on a discrete number of sampling points $\mathbf{f}(u_k, v_k)$.

5.4.4 Constraints for ruled surface optimization

We have presented two matrices \mathbf{D}_u and \mathbf{D}_v in equations (5.18) that represent the linearized constraint for surface fitting in $\mathbb{R}^{3,1}$. Together with the bilinear form $\Phi_{3,1}(\mathbf{f}_u \wedge \mathbf{f}_v, \mathbf{f}_u \wedge \mathbf{f}_v)$ it can guarantee that a discrete sampling of tangent planes to $\mathbf{f}(u, v)$ stays Euclidean.

In this section we want to show a stricter constraint for the special case of tensor product B-spline surfaces of bidegree (3, 1), i.e. strips of ruled surfaces.

We do not go into all the details of ruled surface fitting, which is usually carried out in \mathbb{R}^3 (but the extension to $\mathbb{R}^{3,1}$ is straightforward); they can be found in [30], [36] or [13] and the vast number of references therein. Suffice it to say that initialization, which we will address in sec. 5.5.2, is of great importance.

A ruled surface strip of $\mathbf{f}(u, v)$

Let $\mathbf{f}(u, v)$ be a tensor product B-spline surfaces of bidegree (3, 1) in $\mathbb{R}^{3,1}$, i.e. the u -parameter lines are cubic curves and the parameter lines in v -direction are line segments. Let $\mathbf{g}(u, \bar{v})$ be the ruled surface strip between $\mathbf{f}(u, v_j)$ and $\mathbf{f}(u, v_{j+1})$, $u \in [0, 1]$ and v is normalized such that $\bar{v} \in [0, 1]$, then

$$\mathbf{g}(u, \bar{v}) = (1 - \bar{v}) \cdot \mathbf{f}(u, v_j) + \bar{v} \cdot \mathbf{f}(u, v_{j+1}).$$

It follows that the strip's partial derivative w.r.t. u is

$$\mathbf{g}_u(u, \bar{v}) = (1 - \bar{v}) \cdot \frac{\partial \mathbf{f}(u, v_j)}{\partial u} + \bar{v} \cdot \frac{\partial \mathbf{f}(u, v_{j+1})}{\partial u}$$

and for any u_0 we set $\mathbf{a} := \frac{\partial \mathbf{f}}{\partial u}(u_0, v_j)$ and $\mathbf{b} := \frac{\partial \mathbf{f}}{\partial u}(u_0, v_{j+1})$. Hence, \mathbf{a} and \mathbf{b} are the directions of the parametrized curves defining the ruled surface $\mathbf{g}(u, \bar{v})$ at a common ruling $\mathbf{r} := \mathbf{g}(u_0, \bar{v})$.

The constraint between two u -parameter lines

Since we are building on the theory of sec. 5.4.3, we can assume that the tangent planes at $\mathbf{f}(u_0, v_j)$ and $\mathbf{f}(u_0, v_{j+1})$ are made Euclidean with the help of matrices \mathbf{D}_u and \mathbf{D}_v of eq. (5.18). However, the tangent planes along the ruling \mathbf{r} change their direction and we want to show how to constrain them to be Euclidean as well.

A tangent plane along the ruling is spanned by the constant direction $\frac{\partial \mathbf{f}}{\partial v}(u_0, v_j)$ of the ruling \mathbf{r} , and a linear combination of the directions \mathbf{a} and \mathbf{b} . To see if and where this linear

combination is an isotropic or pseudo-Euclidean direction, we calculate the two roots of the quadratic equation

$$\begin{aligned} \langle \mathbf{g}_u(u_0, \bar{v}), \mathbf{g}_u(u_0, \bar{v}) \rangle_L &= \langle (1 - \bar{v})\mathbf{a} + \bar{v}\mathbf{b}, (1 - \bar{v})\mathbf{a} + \bar{v}\mathbf{b} \rangle_L = 0 \\ \Leftrightarrow \bar{v}_{12} &= \frac{\langle \mathbf{a}, \mathbf{a} \rangle_L - \langle \mathbf{a}, \mathbf{b} \rangle_L \pm \sqrt{\langle \mathbf{a}, \mathbf{b} \rangle_L^2 - \langle \mathbf{a}, \mathbf{a} \rangle_L \cdot \langle \mathbf{b}, \mathbf{b} \rangle_L}}{\langle \mathbf{a} - \mathbf{b}, \mathbf{a} - \mathbf{b} \rangle_L}. \end{aligned} \quad (5.19)$$

The discriminant equals $-\Phi_{3,1}(\mathbf{a} \wedge \mathbf{b}, \mathbf{a} \wedge \mathbf{b})$. According to Theorem 1, it is > 0 whenever the plane $\mathbf{a} \wedge \mathbf{b}$ is pseudo-Euclidean. We can use this knowledge to construct an algorithm that keeps the tangent planes of a ruled surface Euclidean.

Algorithm for ruled surfaces approximation

For ruled surfaces, we can extend the algorithm of section 5.4.3 on page 63, such that not only tangent planes on a sampling of a surface are guaranteed to be Euclidean, but also all tangent planes along the ruling of a ruled surface.

4. For index triples $(i, j, j+1)$ for which the slack variables $\mathbf{y}_{u,i} > 0$, $\mathbf{y}_{v,j} > 0$ and $\mathbf{y}_{v,j+1} > 0$, compute the bilinear form $\Phi_{3,1}(\mathbf{f}_u(u_i, v_j) \wedge \mathbf{f}_u(u_i, v_{j+1}), \mathbf{f}_u(u_i, v_j) \wedge \mathbf{f}_u(u_i, v_{j+1}))$.
5. For those $(i, j, j+1)$ for which $\Phi_{3,1} \leq 0$, compute the roots of eq. (5.19). If there are any in the interval $[v_j, v_{j+1}]$, change the sign of $\Delta \mathbf{y}_{u,i}$ to minus.

In step 5., we have used a similar trick to the one in 3., with the difference that we also have to compute roots of the quadratic equation (5.19). Again, this allows to incorporate a family of tangent planes into the linearized constraint of eq. (5.18).

In conclusion, we have given a special linearization of the quadratic constraint (5.15b) for tensor product B-spline surfaces of bidegree $(3, 1)$ that not only works on a discrete number of sampling points $\mathbf{f}(u_k, v_k)$, but also for all planes along a ruling $\mathbf{f}(u_0, v)$.

5.4.5 Constraints for hyperbolic paraboloidal surface optimization

Here we want to state an algorithm for ensuring that a bilinear tensor product surface $\mathbf{f}(u, v)$ in $\mathbb{R}^{3,1}$ stays Euclidean. We will rely heavily on the findings of section 5.3 on hyperbolic paraboloids in $\mathbb{R}^{3,1}$.

For two pairs of neighboring parameter lines of $\mathbf{f}(u, v)$, let $\mathbf{a} := \mathbf{f}(u_0, v_0)$, $\mathbf{b} := \mathbf{f}(u_1, v_0)$, $\mathbf{c} := \mathbf{f}(u_0, v_1)$ and $\mathbf{d} := \mathbf{f}(u_1, v_1)$. If we reparametrize u and v such that they are in $[0, 1] \times [0, 1]$ on this patch, we can write $\mathbf{f}(u, v)$ exactly as in eq. (5.10) as a bilinear combination of these four corner points.

Algorithm for constraining the tangent planes of a HP surface

1. At the corners $\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}$ of a bilinear B-spline surface $\mathbf{f}(u, v)$, compute the matrices \mathbf{D}_u and \mathbf{D}_v of the linearized constraint (5.18).
2. Index pairs (\bar{i}, \bar{j}) for which the slack variables $\mathbf{y}_{u,\bar{i}} > 0$ and $\mathbf{y}_{v,\bar{j}} > 0$ are precisely the corners for which the directions $\mathbf{a} - \mathbf{b}, \mathbf{c} - \mathbf{d}, \mathbf{a} - \mathbf{c}$ and $\mathbf{b} - \mathbf{d}$ of the partial derivatives are Euclidean.
For those, compute the bilinear forms $\Phi_{3,1}((\mathbf{b} - \mathbf{a}) \wedge (\mathbf{c} - \mathbf{a}), (\mathbf{b} - \mathbf{a}) \wedge (\mathbf{c} - \mathbf{a}))$, $\Phi_{3,1}((\mathbf{d} - \mathbf{b}) \wedge (\mathbf{a} - \mathbf{b}), (\mathbf{d} - \mathbf{b}) \wedge (\mathbf{a} - \mathbf{b}))$, $\Phi_{3,1}((\mathbf{c} - \mathbf{d}) \wedge (\mathbf{b} - \mathbf{d}), (\mathbf{c} - \mathbf{d}) \wedge (\mathbf{b} - \mathbf{d}))$ and $\Phi_{3,1}((\mathbf{a} - \mathbf{c}) \wedge (\mathbf{d} - \mathbf{c}), (\mathbf{a} - \mathbf{c}) \wedge (\mathbf{d} - \mathbf{c}))$, ie. compute the steepness of the tangent planes $\mathbf{T}(\mathbf{a}), \mathbf{T}(\mathbf{b}), \mathbf{T}(\mathbf{c})$ and $\mathbf{T}(\mathbf{d})$.
3. For those (\bar{i}, \bar{j}) for which $\Phi_{3,1} \leq 0$, ie. for which a tangent plane is pseudo-Euclidean, change the sign of $\Delta \mathbf{y}_{u,\bar{i}}$ to be minus.
4. Compute the four discriminants of eq. (5.14), $\Delta = (\langle \mathbf{x}, \mathbf{x} \rangle_L \langle \mathbf{y}, \mathbf{z} \rangle_L - \langle \mathbf{x}, \mathbf{y} \rangle_L \langle \mathbf{x}, \mathbf{z} \rangle_L)^2 - \Phi_{3,1}(\mathbf{x} \wedge \mathbf{y}, \mathbf{x} \wedge \mathbf{y}) \cdot \Phi_{3,1}(\mathbf{x} \wedge \mathbf{z}, \mathbf{x} \wedge \mathbf{z})$, using the substitution list (5.13) and change the sign of $\Delta \mathbf{y}_{v,\bar{j}}$ to minus, whenever a discriminant is negative.

With the simple trick in 4., we have activated the constraint with the help of $\Phi_{3,1}$, even though both partial derivatives are Euclidean. This way, isotropic or pseudo-Euclidean tangent planes enter the optimization.

5.4.6 Footpoints and Regularization

In order to get a good approximation of the parameter values of closest points, the computation of footpoints is central, see sec. 3.2. The 2-parameter case is explained there in detail and the fact that we work in $\mathbb{R}^{3,1}$ instead of \mathbb{R}^3 makes no mentionable difference. Geometric regularizations (see sec. 3.3) is more important in the surface than in the curve case, as matrices tend to be more sparsely filled and therefore optimization is less flexible to outliers.

5.5 Optimization framework for surfaces in $\mathbb{R}^{3,1}$

In this section we give a detailed explanation of how to use the primal-dual interior point algorithm of sec. 4.4.1 to optimize surfaces in $\mathbb{R}^{3,1}$.

Same as in section 4.4.1, we will use Newton's method to solve a system of equations. We

can thus restate the full Newton step as

$$\begin{bmatrix} \mathbf{G} & 0 & -\mathbf{D}^T \\ \mathbf{D} & -\mathbf{I} & 0 \\ 0 & \Lambda & \mathcal{Y} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{b} \\ \Delta \mathbf{y} \\ \Delta \boldsymbol{\lambda} \end{bmatrix} = - \begin{bmatrix} \mathbf{G}\mathbf{b} - \mathbf{D}^T \boldsymbol{\lambda} + \mathbf{e} \\ \mathbf{D}\mathbf{b} - \mathbf{y} \\ \Lambda \mathcal{Y} e + \Delta \Lambda^{\text{aff}} \Delta \mathcal{Y}^{\text{aff}} e - \sigma \mu e \end{bmatrix}. \quad (5.20)$$

This system is almost the same for surfaces in $\mathbb{R}^{3,1}$ as it is for curves in $\mathbb{R}^{2,1}$ except for the sizes of matrices. The dimension of the Hessian \mathbf{G} has changed as we have seen in sec. 5.4.1, as have all variables whose size depend on the size of the control point vector \mathbf{b} .

Remember that \mathcal{Y} and Λ are diagonal matrices containing the slack variables \mathbf{y} and the Lagrange multipliers $\boldsymbol{\lambda}$ respectively; we have used the notation \mathbf{y}_u and \mathbf{y}_v in sec. 5.4.3 for the slack variables of the constraint in u - and v -direction and simply concatenate these vectors to form \mathbf{y} .

\mathbf{I} is the identity matrix, e is a vector of ones. The first block of entries in the column vector on the right-hand side of eq. (5.20) is called *dual residual* and the vector \mathbf{e} it contains was defined in (5.16).

For *duality measure* μ and *centering parameter* σ we mostly followed Nocedal & Wright [29] and could not see grave changes by deviating from the standard values as explained in sec. 4.4.3 for the curve case.

5.5.1 Variables of the optimization

As mentioned above, the interior point formulation for optimizing surfaces in $\mathbb{R}^{3,1}$ solemnly depends on the definitions of some key ingredients, which we want to discuss in this section.

The Hessian matrix \mathbf{G}

The matrix \mathbf{G} or *Hessian* of the objective function was defined in eq. (5.16). Similar to the curve case we regularized it with Tikhonov, first and second derivative regularization (see sec. 3.3) by adding the identity matrix \mathbf{I} , \mathbf{L}_1 and \mathbf{L}_2 of eq. (3.11) with factors in the magnitude 10^{-4} to 10^{-2} , depending on the example. Very often the whole matrix on the left-hand side of eq. (5.20) (the *Jacobian of the Newton-KKT system*) was also Tikhonov regularized depending on its condition number.

The matrices of the linearized constraint \mathbf{D}

In a practical implementation, $\mathbf{f}(u, v)$ is sampled k -times, the matrices \mathbf{D}_u and \mathbf{D}_v of eq. (5.18) representing the linearized steepness constraint are computed and subsumed as the matrix of linear constraints \mathbf{D} .

Section 5.4.3 is devoted to explain the construction of these matrices of the linear constraint. The algorithm on page 63 describes how the information of tangent planes rather than

partial derivatives is incorporated into these matrices by using the bilinear form $\Phi_{3,1}(\mathbf{f}_u \wedge \mathbf{f}_v, \mathbf{f}_u \wedge \mathbf{f}_v)$, which classifies 2-planes in $\mathbb{R}^{3,1}$.

If one wants to approximate a set of points $Q_k \in \mathbb{R}^{3,1}$ by a *ruled* surface in $\mathbb{R}^{3,1}$, sec. 5.4.4 shows that the matrix of the linearized constraint can be adapted to also include tangent planes along the surface's rulings. Furthermore, in sec. 5.4.5 we explain how to use these matrices to optimize for Euclidean *hyperbolic paraboloidal* surfaces in $\mathbb{R}^{3,1}$.

Lagrange multipliers λ and slack variables \mathbf{y}

The vector $\mathbf{D}_u \cdot [\mathbf{b}_x^T | \mathbf{b}_y^T | \mathbf{b}_z^T | \mathbf{b}_w^T]^T = \langle \mathbf{f}_u, \bar{\mathbf{f}}_u \rangle_L$ is actually the vector \mathbf{y} of slack variables in the KKT-conditions. We already know that this vector is less than zero for pseudo-Euclidean partial derivatives, thus causing the tangent plane to be pseudo-Euclidean.

We make use of the slack variables to encode the steepness of tangent planes, as was explained in great detail in sec. 5.4.3, 5.4.4 and 5.4.5: If partial derivatives w.r.t u and v are Euclidean at a point $\mathbf{f}(u_i, v_j)$, but the tangent plane is not, the sign of the corresponding slack variable difference is changed. This has great effect on the choice of the step size parameter α_{pri} .

5.5.2 Initial values

For our interior point optimization, the vector of control points are the *primal variables*, whose initialization is of foremost importance. Initializing the slack variables as well as the Lagrange multipliers (the *dual variables*) is important as well.

Initializing the net of control polygon while ignoring the constraint

For the examples of sec. 6.3, we only used data for which the points Q_k to be interpolated lie on a rectangular grid in x_1 - and x_2 -direction. We then triangulated this area and made independent scattered data interpolations for the x_3 - and x_4 -coordinates. This gave a good initialization for the net of control points \mathbf{b}_0 .

Finding feasible starting positions for the control polygon

The trivial initialization, i.e. projection to the $x_4 = 0$ -hyperplane, or, equivalently, setting all spheres' radii to zero always works, but this starting position is potentially far from the optimum. Even though that is not a restriction in theory, a practical algorithm might fail to converge, especially when facing a non-convex constraint.

Therefore, we first approximated the points $Q_i \in \mathbb{R}^{3,1}$ with a surface in $\mathbb{R}^{3,1}$ while ignoring the steepness constraint. We then locally reduced the spheres' radii until reality of the envelope surface was reached, very similar to what was done in the curve case, see sec. 4.4.2.

Initializing the slack variable \mathbf{y} and the Lagrange multipliers $\boldsymbol{\lambda}$

We initialized the slack variables as $\mathbf{y}_u := \mathbf{D}_u \cdot \mathbf{b}_0$ and $\mathbf{y}_v = \mathbf{D}_v \cdot \mathbf{b}_0$ for a feasible starting position of the control polygon \mathbf{b}_0 , and the Lagrange multipliers $\boldsymbol{\lambda}$ to be identical to one.

Only in the first iteration, the slack variables \mathbf{y} are slightly corrected by first computing the affine scaling step and then redefining $\mathbf{y} := \max(10^{-5}, \mathbf{y} + 10^{-4} \Delta \mathbf{y}_{\text{aff}})$; the Lagrange multipliers are also updated as $\boldsymbol{\lambda} := 10^{-3} \min(\max(0, \boldsymbol{\lambda} + \Delta \boldsymbol{\lambda}_{\text{aff}}), 1)$, which works very well, see a similar construction in the curve case of sec. 4.4.3.

6 Examples

In the final chapter of this work, we want to present examples for our optimization, which is an implementation of the interior point algorithm of Nocedal & Wright [29]. As we have seen in chapters 4 and 5, the main challenge was to formulate a linearization of the quadratic, non-convex constraint that tangents (for curves) and tangent planes (for surfaces) should stay Euclidean.

We implemented all the code for the algorithms in Matlab [46] without the use of external libraries or optimization solvers. After all, all that is needed are solvers for systems of linear equations, for which we relied on Matlab's built-ins.

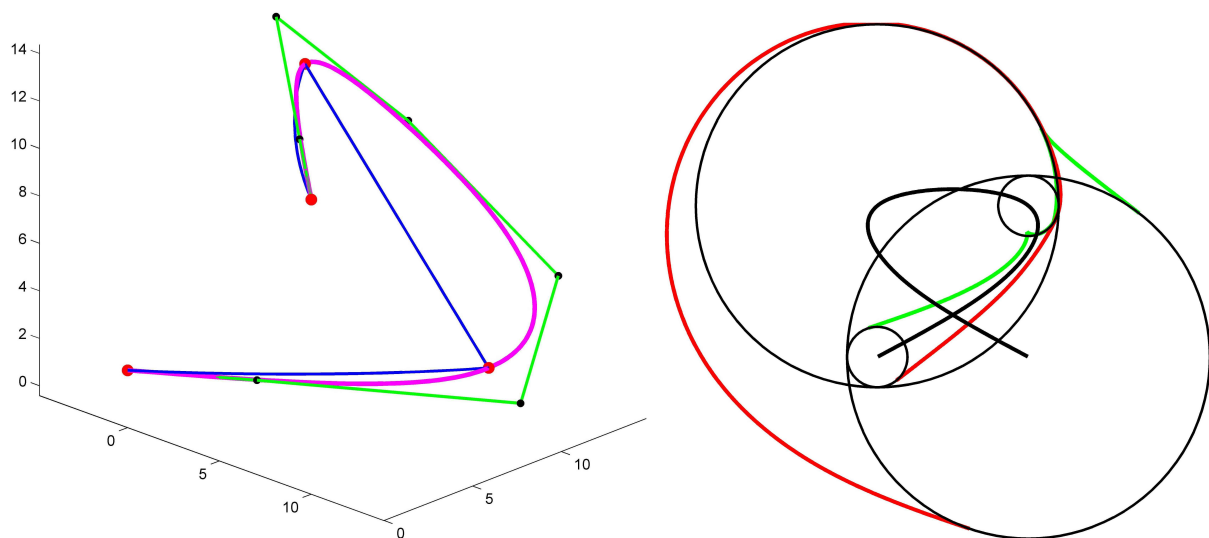


Figure 6.1: The original points in $\mathbb{R}^{2,1}$ in *red* and their connecting lines in *blue* - note that the blue line in the middle has an angle of $\frac{\pi}{4}$ with the xy -plane by construction. The final curve (*purple*) avoids steep tangents, as can be seen by the final control polygon in *green*. *Right* the inverse cyclographic image of the curve in *green* and *red*, as well as the curve's top view in *black*.

Chapter Overview

The order of this chapter follows the order of the previous chapters. In sec. 6.1 we prescribe a set of ordered points Q_k in $\mathbb{R}^{2,1}$, through which we fit a cubic B-spline curve $\mathbf{c}(u)$ in $\mathbb{R}^{2,1}$.

The inverse cyclographic image of this curve $\zeta^{-1}(\mathbf{c}(u))$ are the two branches of the envelopes of the circles $\zeta^{-1}(Q_k)$.

Sec. 6.2 is already one dimension higher, i.e. the 1-parameter set of points Q_k is in $\mathbb{R}^{3,1}$ and so is its interpolating curve. Translating this information via the inverse cyclographic mapping, we find *canal surfaces* that interpolate a set of spheres in \mathbb{R}^3 .

Finally, section 6.3 is dedicated to 2-parameter families of spheres in \mathbb{R}^3 , which is by the cyclographic mapping a 2-parameter set of points $Q_k \in \mathbb{R}^{3,1}$. These Q_k are interpolated by a surface $\mathbf{f}(u, v)$. We show an example for the interpolation with a bicubic tensor product B-spline surfaces, as well as for the case of piecewise ruled surfaces.

6.1 Examples for envelopes of circles

Here we see examples of our algorithms in 2D, e.g. we are given an ordered set of oriented circles or, equivalently, an ordered set of points in $\mathbb{R}^{2,1}$. These points are approximated with a curve that has tangents no steeper than $\frac{\pi}{4}$, see Fig. 6.1.

We will start of with some examples that work very well due to the setting of the circles and gradually introduce challenges.

6.1.1 General Strategy explained by the example of Figure 6.2

We have given a step-by-step description of the interior point algorithm for curves in $\mathbb{R}^{2,1}$ in section 4.4.3; here we want to show an example that follows this iteration.

The points $Q_k, k = 0, \dots, 18$ in $\mathbb{R}^{2,1}$ that are to be approximated are given by $(3, 6.8, 0.13)$, $(2.5, 7.1, 0.25)$, $(1.5, 7.5, 0.8)$, $(0.5, 7.5, 0.5)$, $(0.1, 6.85, 0.4)$, $(0.5, 6, 0.5)$, $(1.2, 5, 1.1)$, $(1.5, 4, 1.25)$, $(1.2, 2.75, 0.85)$, $(0.95, 2, 0.45)$, $(0.95, 1.5, 0.42)$, $(1.2, 0.9, 0.35)$, $(1.6, 0.4, 0.3)$, $(2.2, 0.3, 0.2)$, $(2.6, 0.7, 0.2)$, $(2.7, 1, 0.18)$, $(2.5, 1.4, 0.18)$, $(2.2, 1.6, 0.15)$ and $(1.9, 1.5, 0.13)$. Initial parameter values are chosen by chord length parametrization (3.3).

In order to have enough flexibility for the curve, we want a cubic curve consisting of $n = 100$ segments, which would lead to a rank deficient system, see sec. 3.1.1 for terminology.

We thus first compute a auxiliary spline curve via a least squares approximation, which ignores any constraints of steepness or self intersections. $81 = 100 - 19$ auxiliary points are computed on this auxiliary curve and the number of equally spaced knots is chosen to be $n - \sqrt{n} + 3$, which works fine for all examples.

For this simple example, the auxiliary spline is feasible everywhere, so the steps described in sec. 4.4.2 are unnecessary, i.e. we do not have to lower the z -coordinates of the control points.

Entering the iteration, parameter values for the 19 Q_k are computed via footpoints (see sec. 3.2) and thus the collocation matrix, which is enhanced or stabilized by regularization terms, the Lagrange factors for which depend on some heuristic, start at 10^{-4} and decrease exponentially.

The matrix \mathbf{D} of sec. 4.3 representing the linearized steepness constraint is established.

Then the Newton steps are computed, following the Interior Point algorithm of sec. 4.4.1. Fig. 6.2, *bottom left* shows the convergence rate, which reaches an error of magnitude 10^{-7} after about 30 iterations. This error is always meant to be the value of the objective function (4.1). Fig. 6.2, *bottom right* gives the curve in $\mathbb{R}^{2,1}$ at different stages of the algorithm. Remember that the steepness at sample points (*black dots*) has been fulfilled at all times. The *top row* shows the envelopes at different stages of the algorithm; in the following examples we will only show this situation in \mathbb{R}^2 , as visual inspection is much easier there.

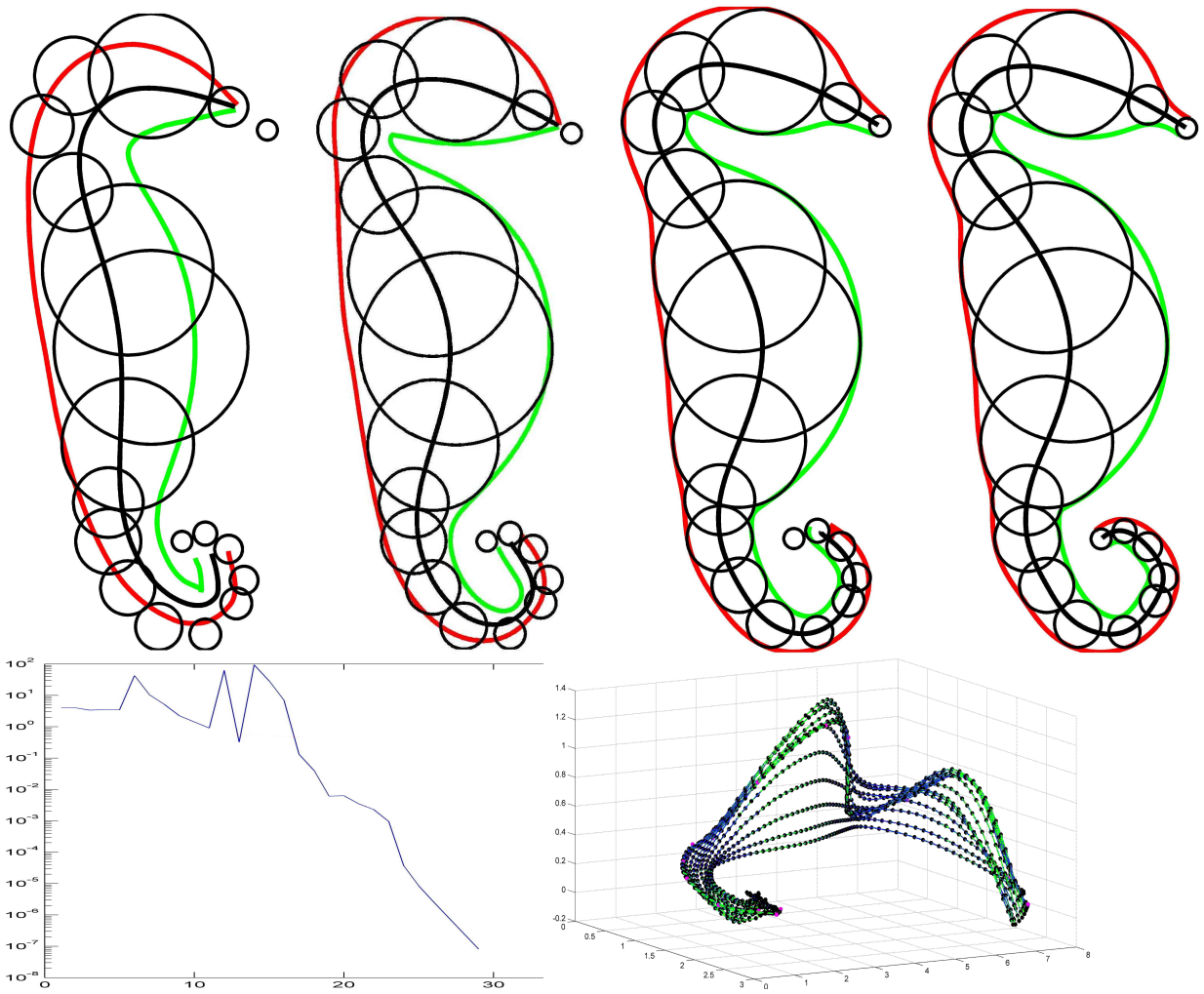


Figure 6.2: An example of envelopes using the Interior Point algorithm at different stages, *top from left to right*: fifth iteration after initialization with a flat curve, after ten iterations, after 15 iterations, final result after 30 iterations. *bottom left* convergence rate for interior point; *bottom right*: curves in $\mathbb{R}^{2,1}$ for the first 15 iterations.

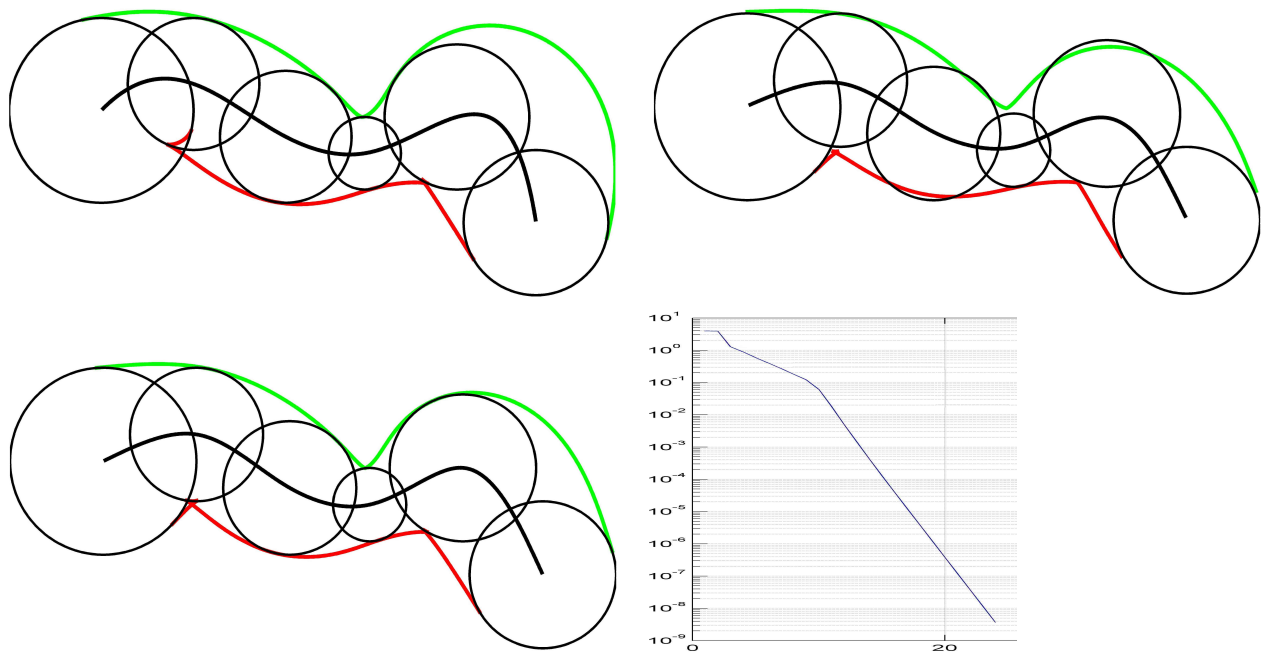


Figure 6.3: Initialization is carried out by first approximating the points $Q_k \in \mathbb{R}^{2,1}$ by a cubic B-spline curve while ignoring the steepness constraint, then lowering the z -coordinate until a feasible starting position has been reached; (*top left*), after 10 iterations of the interior point algorithm (*top right*), final result (*bottom left*) and convergence rate.

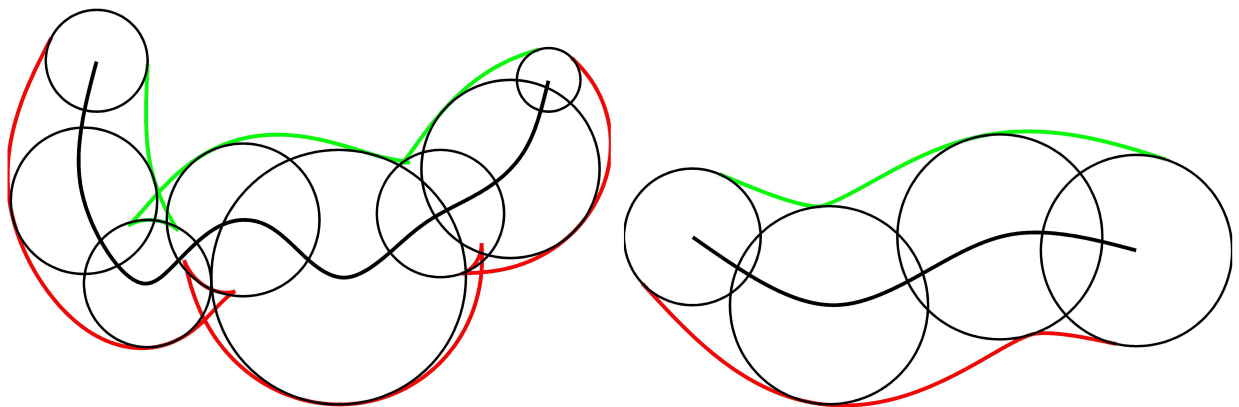


Figure 6.4: Two examples of envelopes using the Interior Point algorithm. Note that the choice of regularization factors λ_1 and λ_2 that govern the weights of the added derivatives operators (see sec. 3.3.1) has to be chosen for every example separately, but is usually between 10^{-6} and 10^{-4} .

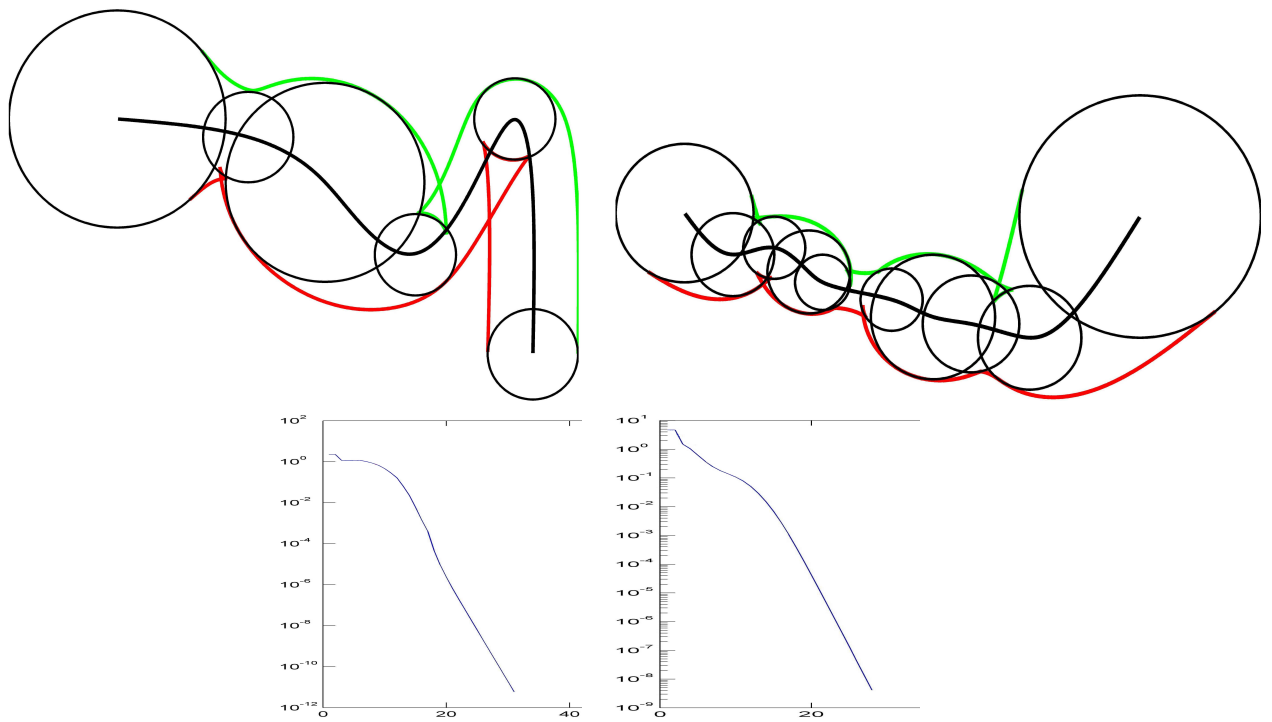


Figure 6.5: Two examples of envelopes (*top row*); *Bottom row*: the graph of convergence rates, where the *y*-axis shows the value of the objective function (4.1) on a logarithmic scale; the *x*-axis shows the number of iterations, *blue* curves represent the interior point algorithm.

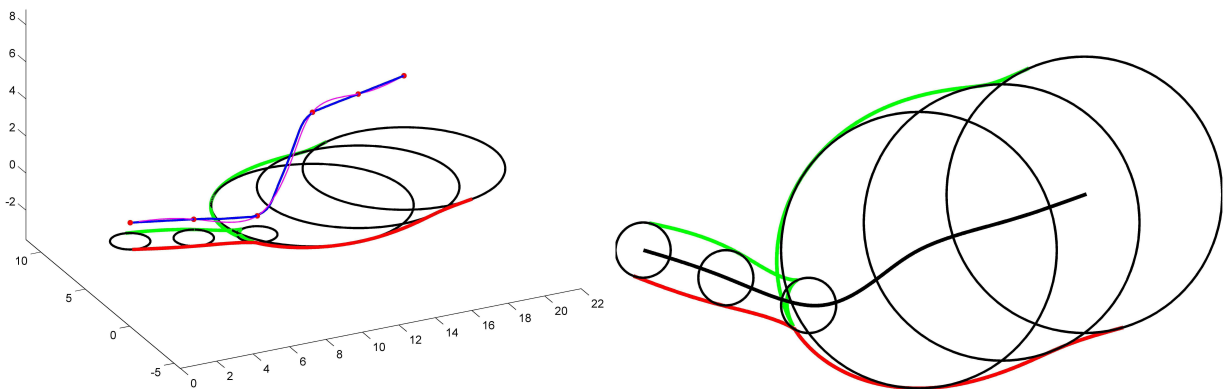


Figure 6.6: An example using the interior point algorithm, *left* the resulting curve in *pink*, the initial curve in *blue*, *right* the top view which shows the envelopes of circles. Note that the *green* envelope can not be realized without self-intersection. See Fig. 6.7, *left* for convergence rates of the interior point algorithm.

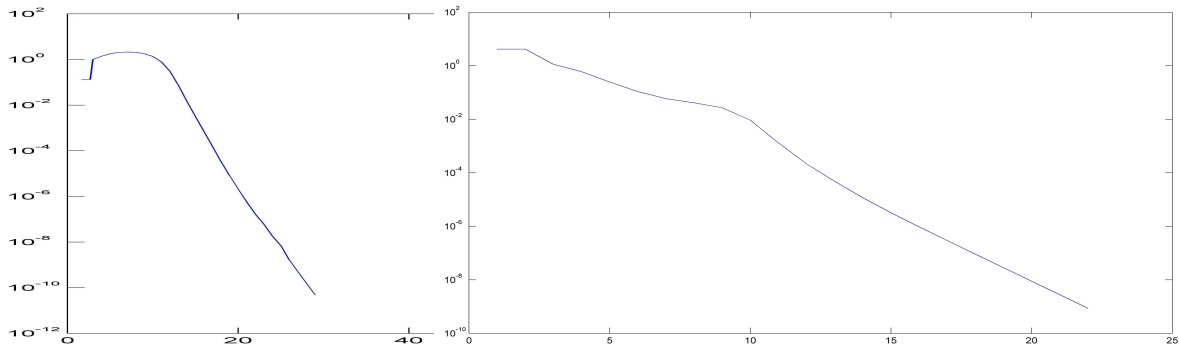


Figure 6.7: Convergence results: the value of the objective function (4.1) and thus the point-wise distance from the input points Q_k to the curve is the error measure in *blue*, given on a logarithmic scale; the x -axis shows the number of iterations. *left* is for the example shown in Fig. 6.6 and *right* the example in Fig. 6.4, *right*.

6.1.2 Effects of the steepness constraint

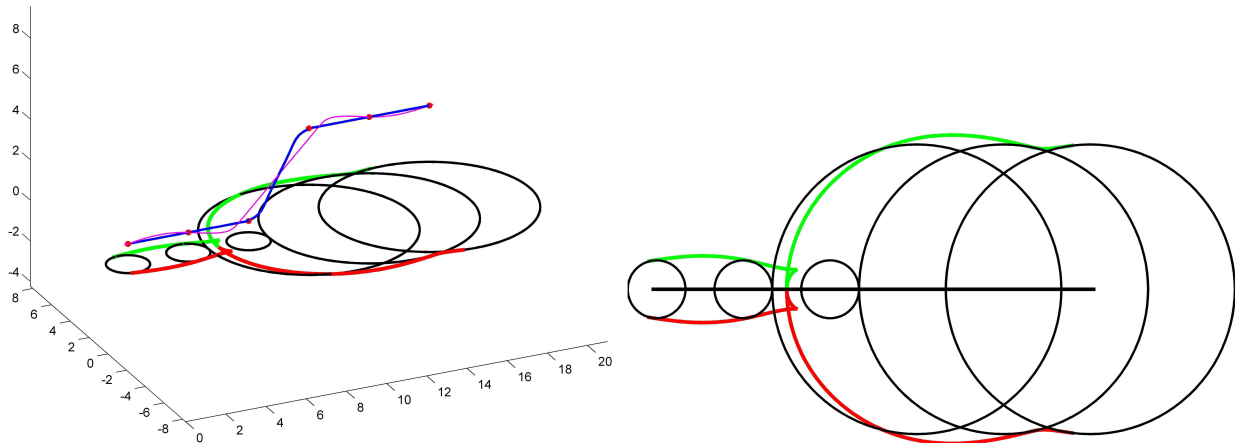


Figure 6.8: An example using the interior point algorithm, *left* the resulting curve in *pink*, the initial in *blue*, *right* the top view which shows the envelopes of circles. Note that one small circle is entirely contained in another, so without optimization the envelope would be imaginary, c.f. Fig.6.9.

As was explained in sec. 2.3.2, the inverse cyclographic image of a curve $\mathbf{c}(u)$ is the envelope of a family of circles, which are two parametrized curves $\mathbf{l}_{1,2}$, defined in eq. (2.12). These curves are real for $\langle \dot{\mathbf{c}}, \dot{\mathbf{c}} \rangle_L \geq 0$, or equivalently, if the angle between $\dot{\mathbf{c}}$ and the xy -plane is $\leq \frac{\pi}{4}$.

The main concern of this thesis is the steepness constraint, which ensures that this criterion is met at a dense sampling of the curve \mathbf{c} . Therefore, if this constraint is left out from the optimization, we expect the $\mathbf{l}_{1,2}$ to be imaginary, thus - as the illustrations in this chapter

only show the real parts of curves - $I_{1,2}$ will have visual discontinuities. This effect can be observed in Fig. 6.9.

Our optimization strategy was outlined in sec. 6.1.1, where the role of a auxiliary spline for a starting position was explained. This auxiliary spline is a simple least squares approximation without any constraint, thus in general violating the steepness constraint at many points - this is the picture shown in Fig. 6.9 for the curves of Fig. 6.8 and Fig. 6.1. As this auxiliary spline is already very close to the final result in many cases, it would be a very good starting position, but unfortunately our implementation of the interior point algorithm needs a feasible starting point, so we lower the z -coordinate until all tangents are Euclidean, see sec. 4.4.2.

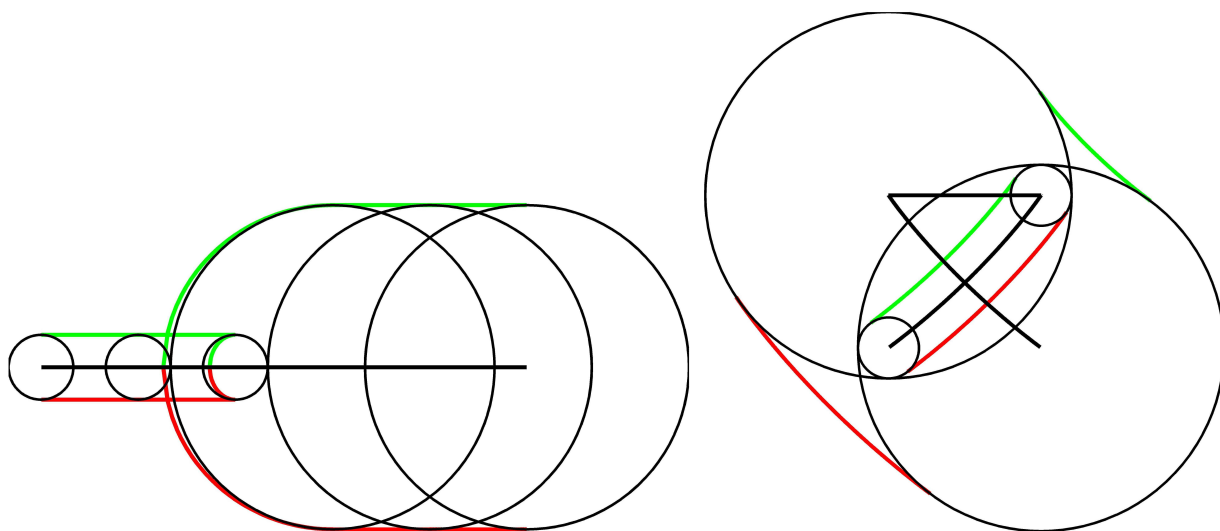


Figure 6.9: When carrying out a curve approximation without the steepness constraint, the inverse cyclographic images of the examples of Fig. 6.8 (*left*) and Fig. 6.1 (*right*) have partly imaginary curve segments, which explains the visual discontinuities of the (real) curves shown here.

6.2 Examples for 1-parameter envelopes of spheres

Given a 1-parameter family of spheres in \mathbb{R}^3 , one wants to find the *canal surface* that envelopes it, see sec. 2.3.3 for notation. By the cyclographic mapping, this problem can be transferred to finding a curve $\mathbf{c}(u) \in \mathbb{R}^{3,1}$ that interpolates points Q_k in 4-dimensional Minkowski space $\mathbb{R}^{3,1}$. If all tangents to $\mathbf{c}(u)$ are *Euclidean*, i.e. all tangents enclose an angle $\leq \frac{\pi}{4}$ with the $x_4 = 0$ -hyperplane, then the corresponding canal surface is real everywhere. We have described an algorithm for carrying out this task in sec. 5.1 and want to show some examples here. The code to generate the examples of this section is based on the

Interior point method for curves in $\mathbb{R}^{2,1}$, whose results we have seen in sec. 6.1. It was extended to curve fitting in $\mathbb{R}^{3,1}$ without difficulty, as there are only minor adjustments.

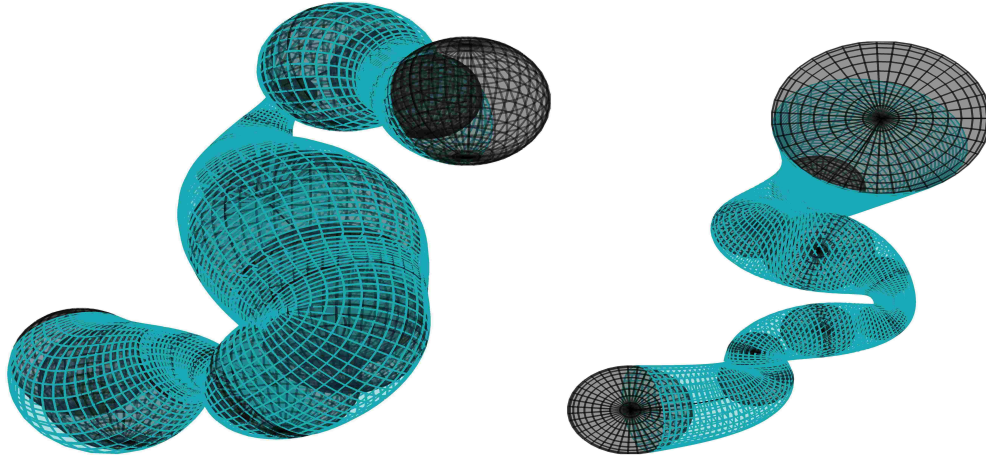


Figure 6.10: Two examples of envelopes of spheres using the Interior Point algorithm generalized to $\mathbb{R}^{3,1}$, e.g. canal surfaces. The original spheres are shown in *black*, the resulting envelope in *turquoise*. The curve approximation and the steepness constraint carry over directly from $\mathbb{R}^{2,1}$ to $\mathbb{R}^{3,1}$.

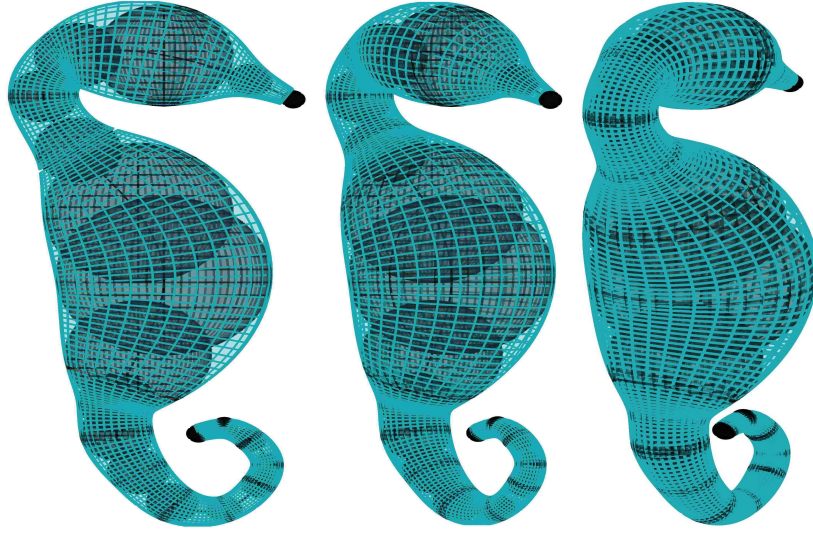


Figure 6.11: Three views of an example of envelopes of spheres using the Interior Point algorithm. The original spheres are shown in *black*, the resulting envelope in *turquoise*. This is the example shown in Fig. 6.2 with an extra dimension added, which does not change computation time.

6.3 Examples for 2-parameter envelopes of spheres

In the final section of this work, we will present examples for envelopes of 2-parameter families of spheres as was explained in sec. 5.4. The first example of sec. 6.3.1 is a bicubic B-spline surface as explained in sec. 5.4.3.

For the examples of sec. 6.3.2 and 6.3.3, we approximate with tensor product B-Spline surfaces of degree $\{3, 1\}$, i.e. one direction is cubic and the other linear, thus giving a piecewise ruled surface, see sec. 5.4.4 for algorithmic details.

6.3.1 Wave example

In this example, the points Q_i lie on a surface $f(u, v) = (u, v, 3 - \frac{r}{2}, 3\frac{\sin(r)}{r})$, with $r = \sqrt{u^2 + v^2} + \epsilon$ and $\epsilon > 0$ very small. This surface is symmetric with respect to interchanging u, v , i.e. rotational symmetry in the x_0x_1 -plane. The measure of steepness thus becomes $g(u) = \langle f_u(u, 0), f_u(u, 0) \rangle_L = \frac{5}{4} - \frac{9}{(u+\epsilon)^2} (\cos(u+\epsilon) - \frac{\sin(u+\epsilon)}{u+\epsilon})^2$, which is symmetric in u , i.e. $g(-u) = g(u)$ and negative for $u \in [1.3472, 2.8423]$ (numerically determined roots). The values for which $\langle f_u, f_u \rangle_L < 0$, $\langle f_v, f_v \rangle_L < 0$, $\langle f_u + f_v, f_u + f_v \rangle_L < 0$ and $\langle f_u - f_v, f_u - f_v \rangle_L < 0$ have been marked *red* in Fig. 6.12, *top row* in order to show the difficulties of this example.

As an initial value, we first approximated the points Q_i without any steepness constraint

and then scaled the control points in the center by a factor $\frac{4}{5}$, which gives a feasible starting point. As can be seen in Fig. 6.13 (*second row right*) the convergence is very fast in the beginning when the iteration takes big steps toward the points; after about 60 iterations no significant improvement can be made due to the strict steepness constraint.

6.3.2 Sine cylinder

For the first of the piecewise ruled surface examples the points Q_i are regular samples of a cylinder with the parametrization $f(u, v) = (x_0, x_1, x_2, x_3) = (u, v, 1, \frac{6}{5} \sin(u))$, $u \in [0, 2\pi]$, $v \in [0, 4]$. The partial derivative $\frac{\partial f}{\partial v} = (0, 1, 0, 0)$ is always Euclidean, but for $f_u := \frac{\partial f}{\partial u} = (1, 0, 0, \frac{6}{5} \cos(u))$ it holds that $\langle f_u, f_u \rangle_L = 1 - \frac{36}{25} \cos^2(u) < 0$ for $u \in [0, \arccos(\frac{5}{6})] \cup (\arccos(-\frac{5}{6}) + \pi, 2\pi]$.

We started with two different initial values for a ruled surface approximation, once we shrunk the x_3 -coordinate by a factor of $\frac{5}{6}$, see Figs 6.14 and 6.15, *left* columns, the other case was stretching the x_0 -coordinate by factor $\frac{6}{5}$, see Figs 6.14 and 6.15, *right* columns. For this simple example, the ruling direction f_v is parallel to x_2 and has little to no effect on the outcome, see Fig. 6.14, *third row*. Therefore, the partial derivative vectors $f_u(u_0, v_j)$ are parallel for fixed u_0 and varying v_j , thus if the vector $f_u(u_0, v_0)$ is Euclidean/isotropic/pseudo-Euclidean, so is the tangent plane at $f(u_0, v_0)$.

6.3.3 Peaks example

This example stems from combining Matlab's [46] **peaks** function $p(u, v)$, which is described to be "[$\cdot \cdot \cdot$]" a function of two variables, obtained by translating and scaling Gaussian distributions, which is useful for demonstrating "[$\cdot \cdot \cdot$]" as x_3 -coordinate, and $x_2 = \sin(u)$, thus $f(u, v) = (u, v, 0.15 \cdot p(u, v), \sin(u))$, see Figs 6.16 and 6.17.

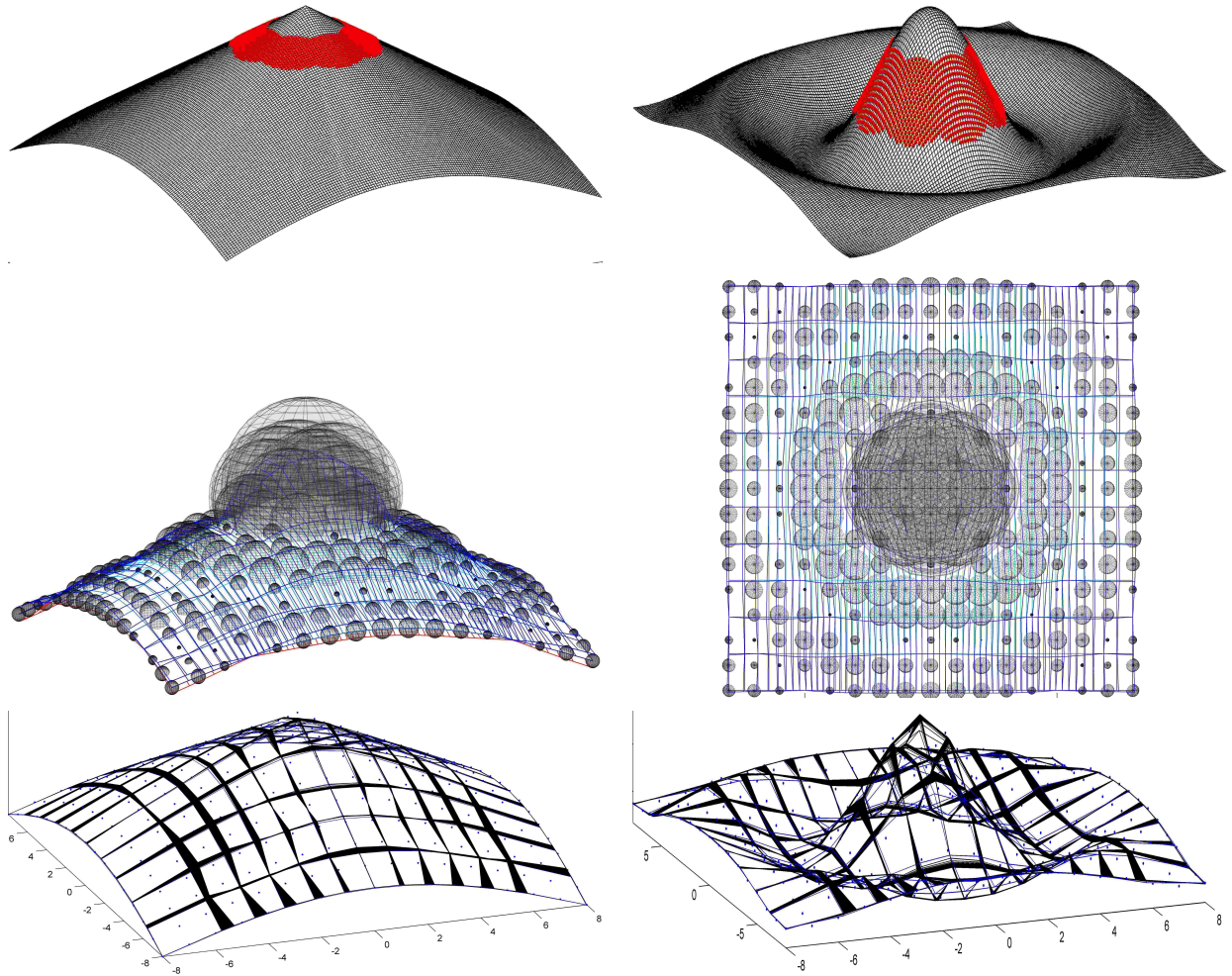


Figure 6.12: The surface $f(u, v) = (u, v, 3 - \frac{r}{2}, 3\frac{\sin(r)}{r})$ of sec. 6.3.1 (*wave example*): *top row left*: an axonometric view of the hyperplanar section $x_3 = 0$, *right*: an axonometric view of the hyperplanar section $x_2 = 0$; surface evaluated on a grid, points with pseudo-Euclidean (mixed) partial derivatives in *red*; *second row*: the cyclographic preimage of an initial value for a bicubic B-spline surface approximation; *last row left*: values in the hyperplanar section $x_3 = 0$, *right* in hyperplanar section $x_2 = 0$ from start (*blue*) and intermediates to finish (*black*).

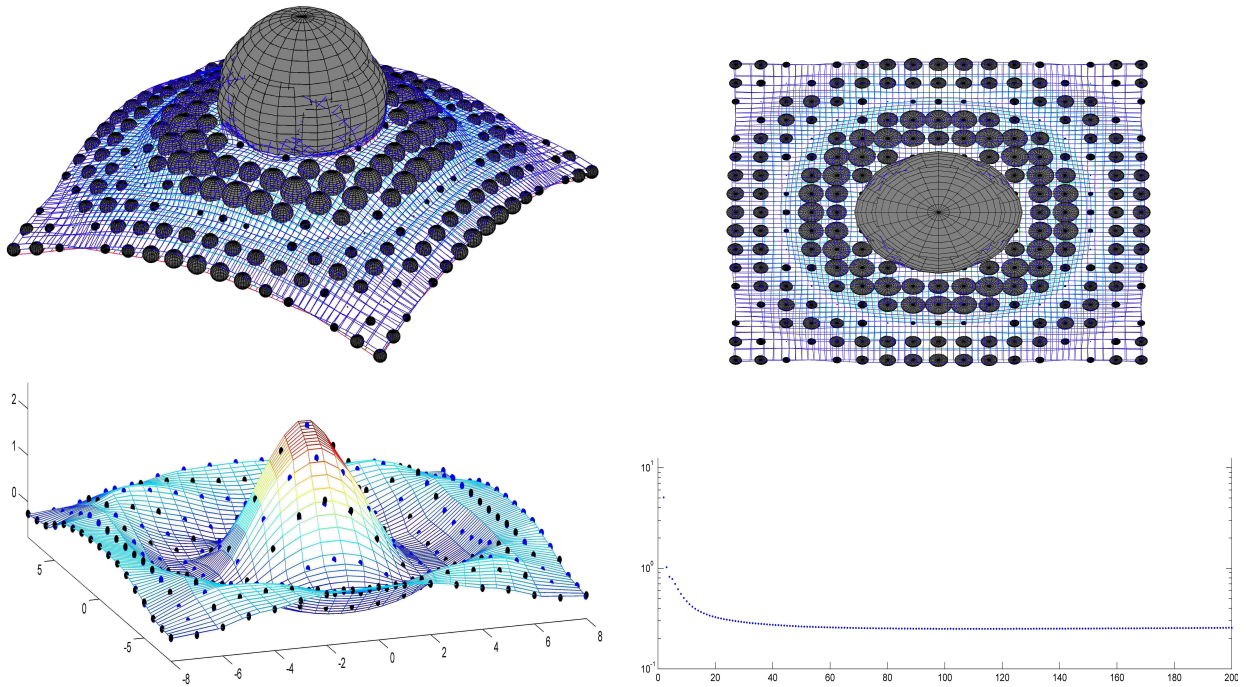


Figure 6.13: *First row*: the cyclographic preimage of the resulting bicubic B-spline surface of the *wave* example of Fig. 6.12. Numerical analysis, *second row left*: the hyperplanar section $x_2 = 0$ of the resulting surface in color, points Q_i in *black*, footpoints in *blue*; *right*: numerical convergence rate on a logarithmic scale measuring the sums of distances from Q_i to its footpoints.

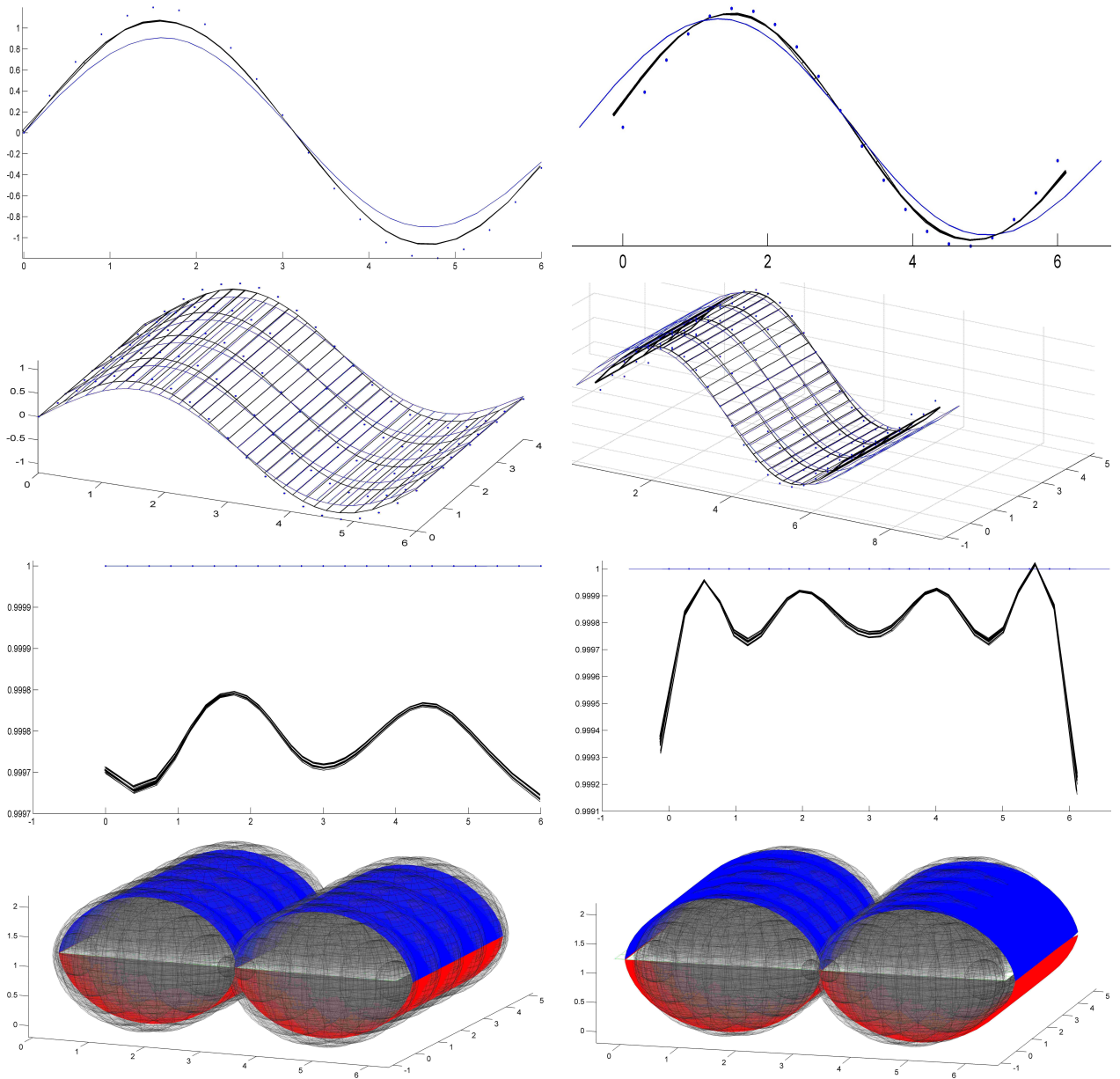


Figure 6.14: The *sine cylinder* example of sec. 6.3.2, *top row*: projections to the x_0x_3 -plane *left*: input points as *blue* dots, for initial position the x_3 -coordinate is shrunk (*blue* curve), the resulting curve in *black*. *right*: for initial position the x_0 -coordinate is stretched (*blue* curve); *second row*: hyperplanar section by $x_2 = 0$; initial values in *blue*, intermediate and final results in *black*; *third row*: projection to the x_0x_2 -plane; initial values in *blue*, intermediate and final results in *black* - note that the scale on the x_2 -axis is 10^5 -times smaller than on the x_0 -axis; *fourth row*: the cyclographic preimages have two branches, input points Q_i rendered as *gray* spheres.

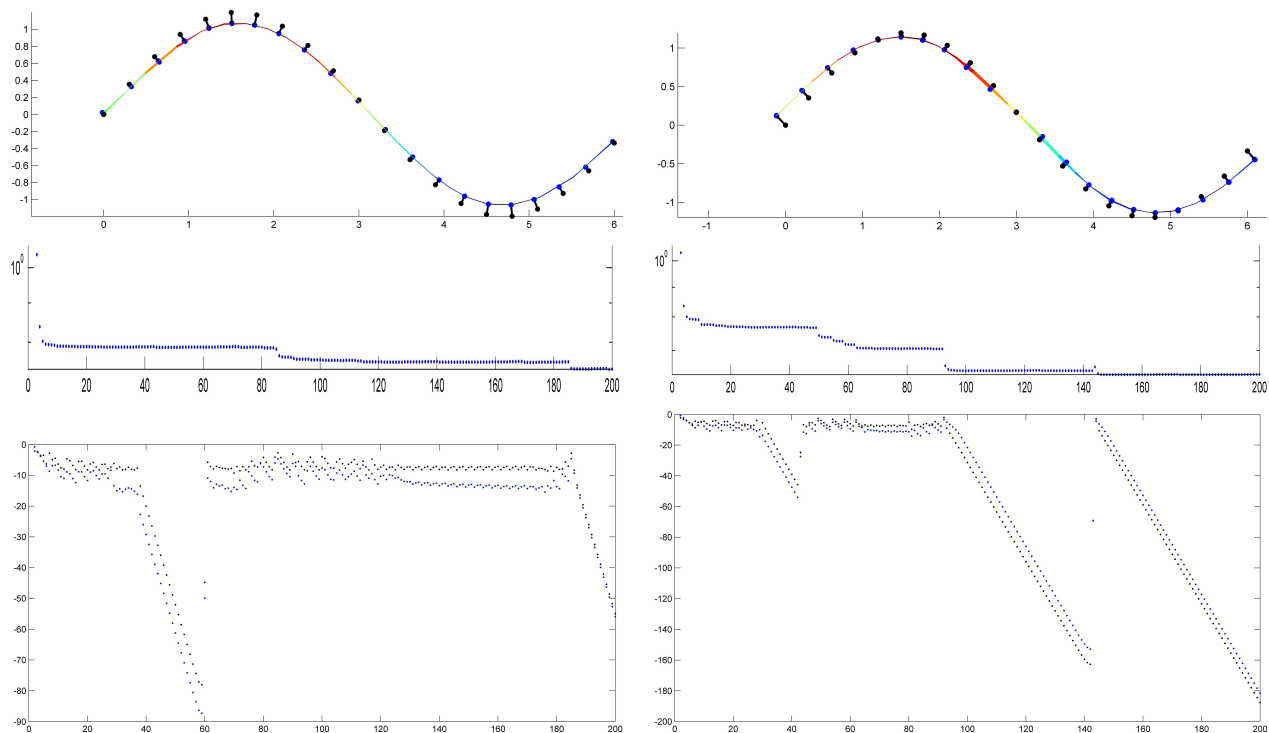


Figure 6.15: The analysis of the *sine cylinder* example of sec. 6.3.2, *top row left*: projection of the resulting surface to the x_0x_3 -plane in colors, input points Q_i in *black*, footpoints in *blue*; the value of the objective function is the sum of distances between input and footpoints, *second row*: the graph of this function versus the iteration is the *numerical convergence rate* (on a logarithmic scale) - *left* initial position x_3 -coordinate shrunk, *right* for initial position x_0 -coordinate stretched; *third row*: the values of the primal stepsize parameter α_{pri} in *blue*, the dual α_{dual} in *black* on a logarithmic scale.

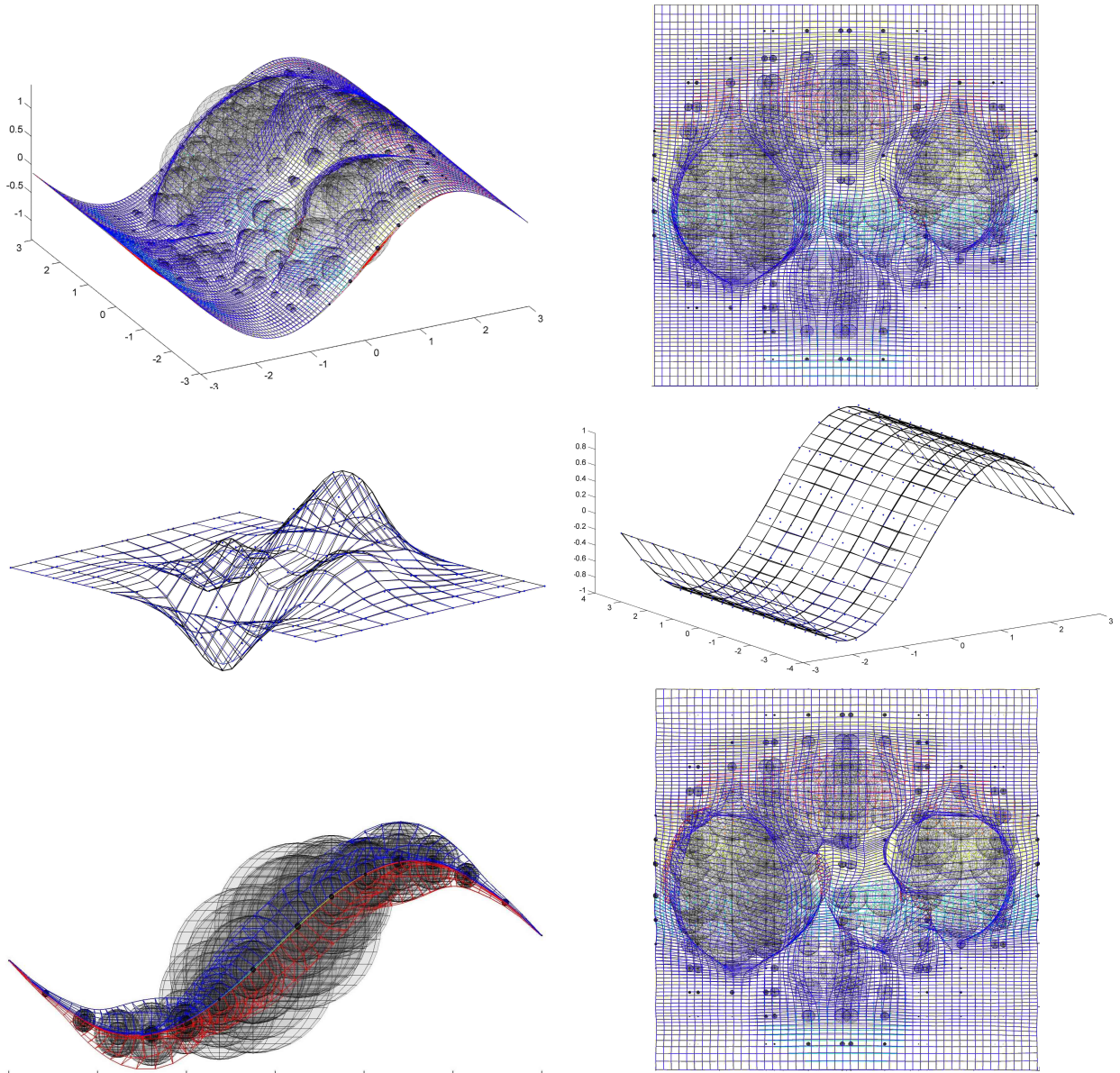


Figure 6.16: The surface $f(u, v) = (u, v, 0.15 \cdot p(u, v), \sin(u))$ of sec. 6.3.3 (*peaks example*):
First row: Starting position in axonometric and top view of the cyclographic preimage; *second row left*: hyperplanar section $x_2 = 0$ and *right* hyperplanar section $x_3 = 0$ from start (*blue*) and intermediates to finish (*black*); *third row*: cyclographic preimage of the resulting piecewise ruling surface.

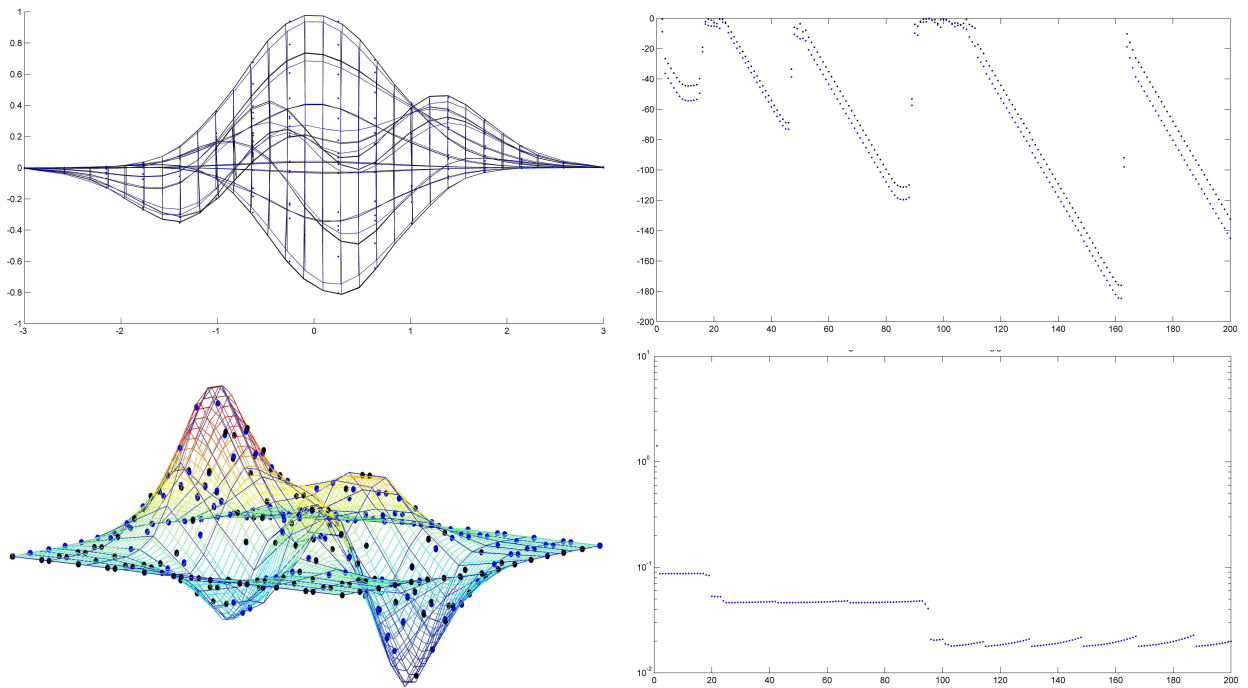


Figure 6.17: *First row left:* hyperplanar section $x_2 = 0$ from start (*blue*) and intermediates to finish (*black*), *right:* stepsize parameters, α_{pri} in *blue* and α_{dual} in *black* on a logarithmic scale for 200 iterations. Numerical analysis: *second row left:* the hyperplanar section $x_2 = 0$ of the resulting surface in color, points Q_i in *black*, footpoints in *blue*. *Right:* numerical convergence rate on a logarithmic scale measuring the sums of distances from Q_i to its footpoints.

List of Figures

1.1	Introductory example	1
1.2	Applications: error paths, gently inclined roads, medial axis transform . . .	3
2.1	Types of lines in Minkowski space	7
2.2	Types of planes in Minkowski space	9
2.3	Distances of circles	11
2.4	Cyclographic images of three different types of lines	12
2.5	Cyclographic image of a curve	14
2.6	Offsets of curves and distance function	15
2.7	The two developable surfaces of constant slope $D_{\mathbf{p}}$	16
2.8	Canal surfaces as cyclographic image of a curve in $\mathbb{R}^{3,1}$	17
2.9	Inverse cyclographic image of a surface in Minkowski space	18
3.1	B-spline basis functions	19
3.2	Footpoint computation	24
4.1	An example explaining the task of the optimization	27
4.2	Unparameterized solution of a special case via [12]	30
4.3	Curve and hodograph	31
4.4	Projection orthogonal to cone Γ	37
4.5	Procedure to find the initial position of the control polygon	41
5.1	A sketch of the situation at infinity	55
6.1	Two examples of optimized curves using the interior point algorithm	70
6.2	An example of envelopes using the Interior Point algorithm in 5 steps . . .	73
6.3	Initialization through a cubic B-spline, final result and convergence rate . .	74
6.4	Two examples of envelopes using the Interior Point algorithm	74
6.5	Two examples of envelopes and their convergence rates	75
6.6	Example of optimized curves using the interior point algorithm	75
6.7	Convergence of two examples	76
6.8	Example of optimized curves using the interior point algorithm	76
6.9	Curve approximation without the steepness constraint	77
6.10	Two examples of envelopes of spheres	78
6.11	Three views of an example of envelopes of spheres	79
6.12	Wave example, position	81

6.13	Wave example, final position and analysis	82
6.14	Sine cylinder example, positions	83
6.15	Sine cylinder example, analysis	84
6.16	Peaks example, starting position	85
6.17	Peaks example, analysis	86

Bibliography

- [1] K. Abdel-Malek, Y. Jingzhou, D. Blackmore, and K. Joy. Swept Volumes: Foundation, perspectives, and applications. *International Journal of Shape Modeling*, 12(1):87–127, 2006.
- [2] S. J. Ahn. *Least squares orthogonal distance fitting of curves and surfaces in space*, volume 3151 of *Lectures Notes in Computer Science*. Springer-Verlag New York Inc, 2004.
- [3] A. Blake and M. Isard. *Active contours*. Springer London, 2000.
- [4] W. Blaschke. *Vorlesungen über Differentialgeometrie*, volume Bd 3: Differentialgeometrie von Kreisen und Kugeln. Springer, 1929.
- [5] J. J. Callahan. *The Geometry of Spacetime*. Springer, 2000.
- [6] H. Cheng and X. Shi. Quality mesh generation for molecular skin surfaces using restricted union of balls. In *Visualization, 2005. VIS 05. IEEE*, pages 399–405. IEEE, 2005.
- [7] H. Choi, S. Choi, and H. Moon. Mathematical theory of medial axis transform. *Pacific Journal of Mathematics*, 181(1):57–88, 1997.
- [8] M. do Carmo. *Differential Geometry of Curves and Surfaces*. Prentice-Hall, New Jersey, 1976.
- [9] H. Edelsbrunner. Deformable smooth surface design. *Discrete and Computational Geometry*, 21(1):87–115, 1999.
- [10] G. Farin. *Curves and Surfaces for CAGD. A practical guide*. Morgan Kaufmann Publishers, fifth edition edition, 2002.
- [11] G. Farin and J. Hoschek. *Handbook of computer aided geometric design*. North Holland, 2002.
- [12] S. Flöry. Fitting curves and surfaces to point clouds in the presence of obstacles. *Computer Aided Geometric Design*, 26:192–202, 2009.
- [13] S. Flöry. *Constrained Matching of Point Clouds and Surfaces*. PhD thesis, Vienna University of Technology, 2010.

- [14] P. M. Gruber. *Convex and discrete geometry*. Springer, 2007.
- [15] P. C. Hansen. *Rank-Deficient and Discrete Ill-Posed Problems*. Siam, 1998.
- [16] J. Hoschek. Intrinsic parametrization for approximation. *Computer Aided Geometric Design*, 5(1):27–31, 1988.
- [17] J. Hoschek and B. Jüttler. Techniques for fair and shape-preserving surface fitting with tensor-product B-splines. In *Shape Preserving Representations in Computer Aided Design (J.M. Peña, ed.)*, pages 163–185. Nova Science publishers, New York, NY, 1999.
- [18] J. Hoschek and D. Lasser. *Grundlagen der geometrischen Datenverarbeitung*. Teubner, 1988.
- [19] S.-M. Hu and J. Wallner. A second order algorithm for orthogonal projection onto curves and surfaces. *Computer Aided Geometric Design*, 22(3):251 – 260, 2005.
- [20] T. W. Hungerford. *Algebra*. Springer, 1974.
- [21] D. James and C. Twigg. Skinning mesh animations. In *ACM SIGGRAPH 2005 Papers*, pages 399–407. ACM, 2005.
- [22] J. Kosinka and B. Jüttler. G1 hermite interpolation by minkowski pythagorean hodograph cubics. *Computer Aided Geometric Design*, 23(5):401 – 418, 2006.
- [23] J. Kosinka and B. Jüttler. C^1 Hermite interpolation by Pythagorean hodograph quintics in Minkowski space. *Adv. Comput. Math.*, 30(2):123–140, 2009.
- [24] N. Kruithof and G. Vegter. Envelope surfaces. In *Proceedings of the twenty-second annual symposium on Computational geometry*, pages 411–420. ACM, 2006.
- [25] R. Kunkli and M. Hoffmann. Skinning of circles and spheres. *Computer Aided Geometric Design*, 2010.
- [26] I. Lee, M. Kim, and G. Elber. Planar curve offset based on circle approximation. *Computer-Aided Design*, 28(8):617–630, 1996.
- [27] Q. Lin and J. Rokne. Disk bézier curves. *Computer Aided Geometric Design*, 15(7):721–737, 1998.
- [28] E. Müller and L. Krames. *Vorlesungen über Darstellende Geometrie II*. Deuticke, Leipzig und Wien, 1929.
- [29] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, second edition edition, 2006.
- [30] M. Peternell. Developable surface fitting to point clouds. *Computer Aided Geometric Design*, 21(8):785–803, 2004.

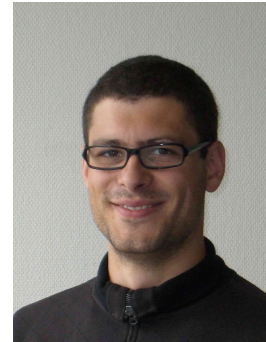
- [31] M. Peternell, B. Odehnal, and M. Sampoli. On quadratic two-parameter families of spheres and their envelopes. *Computer Aided Geometric Design*, 25(4-5):342–355, 2008.
- [32] M. Peternell and H. Pottmann. Computing rational parametrizations of canal surfaces. *Journal of Symbolic Computation*, 23:255–266, 1997.
- [33] M. Peternell and T. Steiner. A geometric idea to solve the eikonal equation. In B. Jüttler, editor, *SCCG '05: Proceedings of the 21st spring conference on Computer graphics*, pages 43–48. ACM Press, 2005.
- [34] L. Piegl and W. Tiller. *The NURBS book*. Springer Verlag, 1997.
- [35] H. Pottmann, P. Grohs, and B. Blaschitz. Edge offset meshes in Laguerre geometry. *Advances in Computational Mathematics*, 33(1):45–73, 2010.
- [36] H. Pottmann and S. Leopoldseder. A concept for parametric surface fitting which avoids the parametrization problem. *Computer Aided Geometric Design*, 20(6):343–362, 2003.
- [37] H. Pottmann, S. Leopoldseder, and M. Hofer. Approximation with active B-spline curves and surfaces. In S. Coquillart, S.-M. Hu, and H.-Y. Shum, editors, *10th Pacific Conference on ComputerGraphics and Applications*, pages 8–25. IEEE Press, 2002.
- [38] H. Pottmann and M. Peternell. Applications of Laguerre geometry in CAGD. *Computer Aided Geometric Design*, 15(2):165–186, 1998.
- [39] H. Pottmann and M. Peternell. Envelopes-Computational theory and applications. *Proceedings, Spring Conference on Computer Graphics and its Applications, Budmerice, Slovakia*, 2000.
- [40] H. Pottmann and M. Peternell. On approximation in spaces of geometric objects. *The Mathematics of Surfaces IX, R. Cipolla and R. Martin, eds., Springer, London*, pages 438–458, 2000.
- [41] H. Pottmann and J. Wallner. *Computational Line Geometry*. Springer, 2001.
- [42] A. Quarteroni, R. Sacco, and F. Saleri. *Numerische Mathematik 1*. Springer, 2002.
- [43] E. Saux and M. Daniel. An improved Hoschek intrinsic parametrization. *Computer Aided Geometric Design*, 20(8-9):513–521, 2003.
- [44] G. Slabaugh, G. Unal, T. Fang, J. Rossignac, and B. Whited. Variational skinning of an ordered set of discrete 2D balls. In *Proceedings of the 5th international conference on Advances in geometric modeling and processing*, pages 450–461. Springer-Verlag, 2008.
- [45] G. Slabaugh, B. Whited, J. Rossignac, T. Fang, and G. Unal. 3D ball skinning using

- PDEs for generation of smooth tubular surfaces. *Computer-Aided Design*, 42(1):18–26, 2010.
- [46] The Mathworks, Inc. Matlab (software package), R2010b. <http://www.mathworks.com/products/matlab/>, 2010.
- [47] Y. Tsai. Rapid and accurate computation of the distance function using grids. *Journal of Computational Physics*, 178(1):175–195, 2002.
- [48] J. Wallner, R. Krasauskas, and H. Pottmann. Error propagation in geometric constructions. *Computer Aided Design*, 32(11):631–642, 2000.
- [49] J. Wallner and Q. Yang. Swept volumes of many poses. In *Proceedings of the third Eurographics symposium on Geometry processing*, page 161. Eurographics Association, 2005.
- [50] W. Wang, H. Pottmann, and Y. Liu. Fitting B-spline curves to point clouds by curvature-based squared distance minimization. *ACM Transactions on Graphics (TOG)*, 25(2):238, 2006.

Curriculum Vitæ

Bernhard Blaschitz

Date of Birth January 12th, 1983 in Vienna
Marital Status Married, one daughter
Nationality Austrian
Address Tamariskengasse 28, 1220 Wien
Email bernhard.blaschitz@waagner-biro.at



Work and Education

since 2012 Advanced Geometry Engineering team at Waagner-Biro Stahlbau AG
2008-2012 University Assistant, Vienna University of Technology
 Institute of Discrete Mathematics and Geometry
2011 Visiting Researcher, Freie Universität Berlin,
 AG Mathematical Geometry Processing, Prof. Konrad Polthier
09/2010 Summer School on Discretization of Geometry and Dynamics,
 by scholarship of and organized by Berlin Mathematical School
2008 Master of Science in Mathematics, University of Vienna
 Completed with distinction
2008 Diploma thesis at Technical University of Vienna supervised by Prof. H.
 Pottmann; title *Discrete Isothermic Surfaces in Isotropic Geometry*
04/2006 Spring School on Abelian Varieties, Leiden University, Netherlands
 by invitation of the Erasmus Mundus program ALGANT
2005/2006 Erasmus stay, University of Padova, Italy
2001 High school diploma *with distinction*, BG/BRG Krems, Piaristengasse 2
1998/1999 High school exchange year, Essex Junction, Vermont, USA