Smoothness Energies in Geometry Processing

Oded Stein

Submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy
under the Executive Committee
of the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY

2020

Abstract

Smoothness Energies in Geometry Processing

Oded Stein

This thesis presents an analysis of several smoothness energies (also called smoothing energies) in geometry processing, and introduces new methods as well as a mathematical proof of correctness and convergence for a well-established method. Geometry processing deals with the acquisition, modification, and output (be it on a screen, in virtual reality, or via fabrication and manufacturing) of complex geometric objects and data. It is closely related to computer graphics, but is also used by many other fields that employ applied mathematics in the context of geometry.

The popular Laplacian energy is a smoothness energy that quantifies smoothness and that is closely related to the biharmonic equation (which gives it desirable properties). Minimizers of the Laplacian energy solve the biharmonic equation. This thesis provides a *proof of correctness and convergence* for a very popular discretization method for the biharmonic equation with zero Dirichlet and Neumann boundary conditions, the piecewise linear Lagrangian mixed finite element method. The same approach also discretizes the Laplacian energy. Such a proof has existed for *flat* surfaces for a long time, but there exists no such proof for the curved surfaces that are

needed to represent the complicated geometries used in geometry processing. This proof will improve the usefulness of this discretization for the Laplacian energy.

In this thesis, the novel *Hessian energy* for curved surfaces is introduced, which also quantifies the smoothness of a functions, and whose minimizers solve the biharmonic equation. This Hessian energy has natural boundary conditions that allow the construction of functions that are not significantly biased by the geometry and presence of boundaries in the domain (unlike the Laplacian energy with zero Neumann boundary conditions), while still conforming to constraints informed by the application. This is useful in any situation where the boundary of the domain is not an integral part of the problem itself, but just an artifact of data representation—be it, because of artifacts created by an imprecise scan of the surface, because information is missing outside of a certain region, or because the application simply demands a result that should not depend on the geometry of the boundary. *Novel discretizations* of this energy are also introduced and analyzed.

This thesis also presents the new *developability energy*, which quantifies a different kind of smoothness than the Laplacian and Hessian energies: how easy is it to unfold a surface so that it lies flat on the plane without any distortion (surfaces for which this is possible are called *developable surfaces*). Developable surfaces are interesting, as they can be easily constructed from cheap material such as paper and plywood, or manufactured with methods such as 5-axis CNC milling. A novel definition of developability for discrete triangle meshes, as well as a variety of *discrete developability energies* are also introduced and applied to problems such as approximation of a surface by a piecewise developable surface, and the design and fabrication of piecewise developable surfaces. This will enable users to more easily take advantages of these cheap and quick fabrication methods.

The novel methods, algorithms and the mathematical proof introduced in this thesis will be useful in many applications and fields, including numerical analysis of elliptic partial differential equations, geometry processing of triangle meshes, character animation, data denoising, data smoothing, scattered data interpolation, fabrication from simple materials, computer-controlled fabrication, and more.

Table of Contents

Table of Contents

List of Figures

Acknowledgements

This thesis would not be possible without the help of many other people who have accompanied me on my journey.

I am forever indebted to my family for their support. From my earliest years they encouraged my love of science, and they steadfastly believed in me throughout my life and my PhD studies.

A special thanks goes to my advisor, Prof. Eitan Grinspun. He taught me the many skills needed to become a successful researcher, as well as a successful person. His supervision and mentorship are responsible for my ability to complete the work in this thesis. I am very grateful to be his student.

Without my collaborators, Alec Jacobson, Keenan Crane, and Max Wardetzky, I could not have completed any of the research that forms the basis of this thesis. Thank you very much for the collaboration, and I hope we will collaborate on many more research projects in the future.

I thank Profs. Qiang Du and Michael Weinstein from APAM, who have been an important source of advice during my time at Columbia.

Acknowledgements

In the summer of 2017 I completed a research project at Pixar Animation Studios under the supervision of Fernando de Goes and Mark Meyer, where I was introduced to the fascinating world of animation studios. I am very grateful to them for this amazing opportunity and the great time I had in Emeryville.

I would like to thank the many friends I have made in New York and Toronto during my stay there. I arrived in both of these places a fish out of water, and thanks to the multitude of people I met there, I felt very welcomed and made myself at home.

I had the pleasure of interacting with many academic communities during my PhD, where I have received support and advice. I would like to specifically thank Henrique Maia, Peter Yichen Chen, Gabriel Cirio, Dingzeyu Li, Raymond Yun Fei, Chang Xiao, and Changxi Zheng at the Columbia Computer graphics group; Alec Jacobson, David Levin, and all their students at the Dynamic Graphics Project at the University of Toronto; and the other PhD students at the department for Applied Physics and Applied Mathematics at Columbia University who helped me study for the difficult qualifying exams, Mel Abler, Zane Martin, Sai Sunku, Rachael Keller, John Brooks, Jared Ginsberg, Melanie Bieli, Joshua Cabrera, Bigeng Wang, and many others.

Montserrat Fernandez-Pinkley, David Barrera, Stella Lau, Jessica Pierson and all the other staff at APAM helped me navigate the intricacies of my PhD, and I am very grateful for that. The same holds for the staff at the CS department, Anne Fleming, Elias Tesfaye, Oana Moldoveanu, Maria Joanta and others. I would also like to thank Columbia University and the entire Columbia community. Columbia University is a great place to be, to learn, and to do research, and I consider myself lucky to be a part of it.

At last, I thank my PhD examination committee for their work in reviewing my thesis.

Introduction

Smoothness energies (sometimes also called smoothing energies) are a fundamental concept in geometry processing that are used in a wide variety of applications, ranging from data processing to physical simulation and digital fabrication. Smoothness energies usually quantify a kind of smoothness—the higher the energy is, the less smooth the measured object is. This thesis discusses three smoothness energies: the Laplacian energy, the Hessian energy, and the developability energy.

In this thesis I will explore two smoothness energies directly related to the biharmonic equation, $\Delta^2 u = f$. The biharmonic equation is one of the fundamental equations in geometry processing. Like its cousin, the Poisson equation ($\Delta u = f$), the biharmonic equation has very smooth solutions. Unlike the Poisson equation, the higher order of the biharmonic equation admits more boundary conditions, which allow for solutions that have more freedom to conform to any particular application. Because of its smooth solution, the biharmonic equation and its associated smoothness energies can be used to smooth data, both to design geometric data as well as to denoise noisy data. The smoothness property can also be used to interpolate and extrapolate data by finding the *smoothest* function that fulfills certain conditions at a set of interpolation points.

Data interpolation is also used to animate characters by associating each point on the surface of a character with a weight belonging to a set of degrees of freedom that can be manipulated to move a character. In physical simulation, the biharmonic equation and its associated smoothness energies are used to model the bending of surfaces and thin shells.

In this thesis I will take an already popular smoothness energy for the biharmonic equation, the Laplacian energy, and provide a proof that a popular discretization method for the energy is convergent on general curved surfaces. This proof was, until now, only available for flat surfaces. I will also introduce a novel smoothness energy based on the biharmonic equation for curved surfaces, the Hessian energy. This new Hessian energy addresses the shortcoming of the Laplacian energy with certain boundary conditions: the solutions are biased by the shape of the boundary.

In this thesis I will also examine energies that allow us to quantify different properties beyond just smoothness, namely the *curvature* and *developability* of surfaces. A smooth surface is developable if it can be flattened without any distortion. Developable surfaces have a wide variety of applications, and are particularly amenable to cheap manufacturing methods such as assembly from paper or plywood, as well as 5-axis CNC milling.

Contributions

This thesis contains the following contributions in the realm of the biharmonic equation and developability energies in geometry processing.

A Proof of Convergence for the Linear Mixed Finite Element Method for the Biharmonic Equation With Dirichlet and Neumann Boundary Conditions. The popular mixed finite

element method for the solution of the biharmonic equation with linear basis functions, which involves optimizing the so-called Laplacian energy while enforcing Dirichlet and Neumann boundary constraints, has been known to converge for flat triangle meshes in $\mathbb{R}^2$ for a very long time. This method, however, is very popular in practice for triangle meshes approximating curved surfaces $\subseteq \mathbb{R}^3$, and convergence is observed in applications, even though a formal proof is missing. In this thesis, I provide a novel formal numerical analysis proof of convergence that confirms the convergence of the method and delineates the exact conditions that are needed to achieve that convergence. This work is also pre-published as 'Stein *et al.* [1]' on arXiv.

The Hessian Energy and its Natural Boundary Conditions. The Dirichlet and Neumann boundary conditions are the most popular boundary conditions for the biharmonic equation in geometry processing, and are usually applied to the Laplacian energy. Neumann boundary conditions, however, are prone to exhibiting bias at the boundary: the isolines of functions that fulfill the zero Neumann boundary conditions always have to be perpendicular to the boundary. As a result, solutions of the biharmonic equation with zero Neumann boundary conditions are strongly biased by the shape of the boundary. While this can be desirable for some applications, many applications actually profit from solutions that are as independent from the boundary shape as possible. This is useful, for example, when the boundary is the result of a noisy scan. I introduce the new *Hessian energy* to curved surfaces. The Hessian energy solves the biharmonic equation, and its natural boundary conditions lead to reduced dependence of the solution on the shape of the boundary. Additionally, I provide two different novel discretizations of the Hessian energy (for flat and for curved surfaces), which converge in experiments. This work is published in two different articles, 'Stein *et al.* [2]' and 'Stein *et al.* [3]'.

Developability Energies. It is of great importance for a variety of applications, mostly in computer-aided fabrication, to produce designs of developable surfaces. In this thesis I define a new smooth developability energy, used to quantify the developability of surfaces; I introduce a novel definition for developable triangle meshes; and I present novel discrete developability energies that can be optimized to turn any triangle mesh into a piecewise developable triangle mesh. Some of this work is published in the article 'Stein *et al.* [4]', and some of this work (the continuous energy in Chapter 4 and its properties) is an as yet unpublished collaboration with Keenan Crane and Eitan Grinspun.

Outline

This thesis is divided into seven chapters. In Chapter 1 I recap a few important textbook definitions and properties from geometry and functional analysis that will be used throughout this thesis. Chapter 2 recaps the widely used Poisson, vector Poisson and biharmonic partial differential equations, which are prior work. In Chapter 3, I introduce the novel Hessian energy for the biharmonic equation, and showcase its applications. Chapter 4 discusses the concept of developable surfaces, and presents the novel developability energy for quantifying the developability of smooth surfaces. Chapter 5 introduces the existing mixed finite element discretization for the biharmonic equation with Dirichlet and Neumann boundary conditions, and presents a novel proof of convergence for the method on curved surfaces discretized by triangle meshes. In Chapter 6, the Hessian energy from Chapter 3 is discretized with a novel approach, both for flat and for curved surfaces. Finally, in Chapter 7, a novel definition for discrete developability for triangle meshes is introduced, and new discrete developability energies are derived that can be optimized to make any triangle mesh piecewise developable.

This thesis is aimed both at an applied mathematics audience as well as a computer graphics audience. Throughout the work I have kept these two audiences in mind, and one consequence of this choice is that some sections might feel a bit too detailed and explaining basic facts. This is hard to avoid with a broad reader base. I have also erred on the side of citing textbooks exhaustively, in order to point readers who are not familiar with certain concepts to the right sources.

Throughout the rest of this thesis, the pronoun "we" is used instead of "I" when referring to myself in the first person. This is somewhat common for PhD theses in this community, and it acknowledges the work of my collaborators on the projects listed above—this work is done by *us*, and not just myself. It also matches more closely the style of scientific papers in mathematics and computer science.

Chapter 1: A Quick Geometry Refresher

This chapter introduces all the definitions that will be used throughout this thesis from the mathematical disciplines of differential geometry and functional analysis. Readers who are already familiar with the topic of surfaces, differential operators, curvature, and Sobolev spaces can feel free to skip this chapter. All notation is standard within the field.

1.1 Smooth Surfaces

What distinguishes geometry from other fields of differential mathematics such as calculus is the fact that everything happens on *manifolds*, which are more intricate than the plain Euclidean space $\mathbb{R}^n$. In this thesis, we will focus on a specific kind of manifolds called *surfaces*, which correspond to our real-world intuition of surfaces.

Definition 1.1 (Surface)**.** *A* surface *is a smooth, compact two-dimensional Riemannian manifold embedded in* $\mathbb{R}^3$*. It has a well-defined unit length normal vector* $\mathbf{n}(p)$ *at every* $p \in \Omega$*.*

It can be with or without boundary. The boundary of a surface Ω *is denoted by* $\partial\Omega$ *and is smooth.*

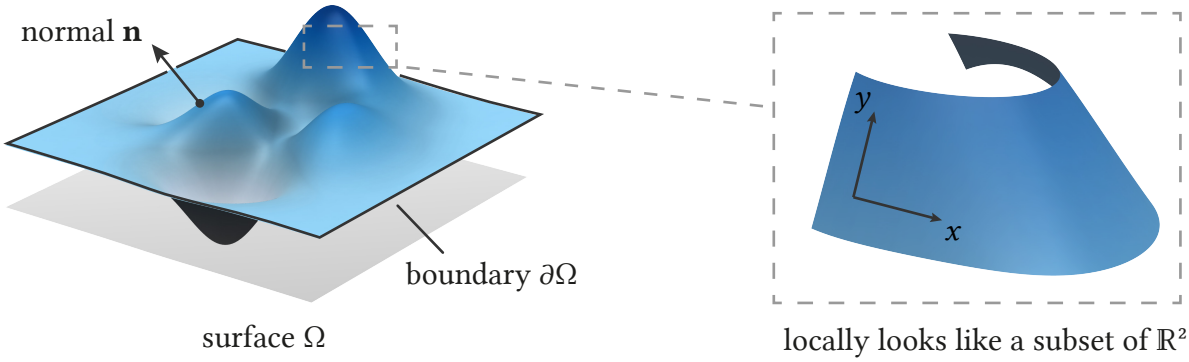


Figure 1.1. A surface is a two-dimensional manifold that is embedded in $\mathbb{R}^3$. It locally looks like a distorted subset of the plane $\mathbb{R}^2$, it has a boundary $\partial\Omega$, and at every point p it has a normal vector $\mathbf{n}(p)$ that is orthogonal to the surface in $\mathbb{R}^3$.

Let Ω be such a surface with a Riemannian metric $g(\cdot, \cdot)$, which corresponds to the Euclidean metric of the surrounding space $\mathbb{R}^3$. The short-hand notation for the Riemannian metric is the dot product, $\mathbf{v} \cdot \mathbf{w} \coloneqq g(\mathbf{v}, \mathbf{w})$. Most of the math in this thesis will take place on Ω.

A surface can be, in broad terms, thought of as an object that locally looks like the plane $\mathbb{R}^2$, possibly with some distortion, and has finite size (see Figure 1.1). A metric is a way to compute the scalar product of two tangent vectors to the surface. For an exact definition and a general introduction to Riemannian manifolds, we refer the reader to the books of do Carmo [5], Lee [6], Petersen [7], and Crane *et al.* [8]. For concepts introduced in this thesis, the definitions are mostly based on the books of Lee [9, 6], although simplified. Each definition features a reference to where the appropriate concepts can be found in these books.

Definition 1.2 (Vectors and covectors)**.** *[9, Chapters 4-6, 11]*

A vector $\mathbf{v} \in \mathbb{R}^3$ *is tangent to* Ω *at the point* $p \in \Omega$ *if it is orthogonal to the surface normal* $\mathbf{n}(p)$ *of* Ω *at* p*. A tangent vector can be equivalently characterized as a way to take a directional derivative of a function along the surface,* $\frac{\partial}{\partial \mathbf{v}}$.

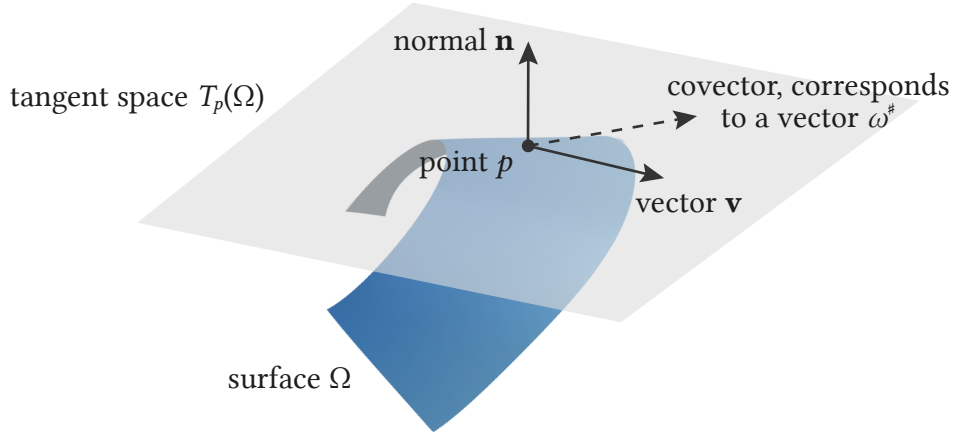


Figure 1.2. A tangent vector **v** to the surface Ω at the point p is a vector that is orthogonal to the normal vector $\mathbf{n}(n)$. All tangent vectors at p form the tangent space $T_p\Omega$. A covector ω maps a tangent vector to a real number (which is like dotting a vector with another vector, $\omega^\sharp$).

$T_p\Omega$ is the linear vector space of all vectors tangent to Ω at p.

$T\Omega$, the disjoint union of all $T_p\Omega$ (where every tangent vector at a point p is identified with the point p), is the tangent space of Ω.

A covector *ω at the point $p \in \Omega$ is a linear function $\omega : T_p\Omega \to \mathbb{R}$.*

$T_p^*\Omega$ is the linear cotangent space of all covectors at p. It is the dual space of $T_p\Omega$ under the metric dual map $\flat : T_p\Omega \to T_p^*\Omega, \quad \mathbf{v}^\flat = g(\mathbf{v}, \cdot)$ (called flat*). The flat operation is invertible, and its inverse $\sharp : T_p^*\Omega \to T_p\Omega$ is called* sharp*.*

Flat and sharp naturally extend the metric $g(\cdot, \cdot)$ to the cotangent space via $g(\omega, \xi) \coloneqq \omega(\xi^\sharp)$.

$T^*\Omega$, the disjoint union of all $T_p^*\Omega$ (where every covector at a point p is identified with the point p) is the cotangent space of Ω.

Depending on the context, the term *vector* will refer to tangent vectors, even when not explicitly specified. Figure 1.2 features an illustration of vectors and covectors.

To represent smoothly varying functions of vectors and covectors, we have the concept of *fields*.

Definition 1.3 (Vector and covector fields)**.** *[9, Chapters 5-6]*

A vector field *is a smooth function* $\Omega \to T\Omega$ *identifying every point in* Ω *with a member of the tangent space at that point,* $T_p\Omega$.

A covector field *is a smooth function* $\Omega \to T^*\Omega$ *identifying every point in* Ω *with a member of the cotangent space at that point,* $T_p^*\Omega$.

Since, at every $p \in \Omega$, $T_p\Omega$ and $T_p^*\Omega$ are linear spaces with an inner product, they admit an oriented orthogonal basis. Oriented means that the order of basis vectors should be determined so that their cross product points in the same direction as the normal vector of the surface. Such a basis can always be extended to a small open set containing p, such that at each other point in this set the extension also forms an orthonormal basis of the respective tangent spaces. These extended smooth bases are called vector frames and covector frames. The usual notation for vector frames will be $\mathbf{x}_1, \mathbf{x}_2$, and the notation for covector frames $\mathrm{d}x^1, \mathrm{d}x^2$ (this is because a covector frame is like a derivative of the coordinate functions x^1, x^2). A vector frame $\mathbf{x}_1, \mathbf{x}_2$ and a covector frame $\mathrm{d}x^1, \mathrm{d}x^2$ are called compatible if

$$dx^i(\mathbf{x}_j) = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}.$$

These frames are not unique, but often their particular choice will not matter.

Vectors and covectors can be used to define tensor product spaces called contravariant and covariant tensors.

Definition 1.4 (Covariant tensors)**.** *[9, Chapter 11] [6, Appendix B]*

A covariant *k-tensor of a vector space V is a multilinear map $\alpha : V^k \to \mathbb{R}$. The space of all covariant k-tensors is written as $T^k(V)$.*

Covariant k-tensors and l-tensors can be multiplied with the tensor product $\otimes$ *defined as the natural extension of the operation $(\alpha \otimes \beta)(\mathbf{v}, \mathbf{w}) = \alpha(\mathbf{v})\beta(\mathbf{w})$.*

The bundle of all tangent covariant k-tensors of Ω is written as T^kM, the disjoint union of $T^k\left(T_pM\right)$ for all points $p \in \Omega$.

The covariant k-tensors can be understood as "covectors with k heads."

Definition 1.5 (Contravariant tensors)**.** *[9, Section 11] [6, Appendix B]*

A contravariant *k-tensor of a vector space V is a linear map $\alpha : (V^*)^k \to \mathbb{R}$, where (V^*) is the linear dual space of V (the space of all linear maps from V to $\mathbb{R}$). The space of all contravariant k-tensors is written as $T_k(V)$.*

Contravariant k-tensors and l-tensors can be multiplied with the tensor product $\otimes$ *defined as the natural extension of the operation $(\mathbf{v} \otimes \mathbf{w})(\alpha, \beta) = \mathbf{v}(\alpha)\mathbf{w}(\beta)$.*

The bundle of all tangent contravariant k-tensors of Ω is written as T_kM, the disjoint union of $T_k\left(T_pM\right)$ for all points $p \in \Omega$.

At first glance, contravariant tensors look like amalgamations of co-covectors. However, contravariant k-tensors can be understood as "vectors with k heads" by the natural identification of the dual space of V^* with V via $v \in V \mapsto (\Phi_v(f) = f(v))$. By this identification, contravariant 1-tensors are actually just vectors.

Definition 1.6 (Generalized tensors)**.** *[9, Chapter 11] [6, Appendix B]*

A (k, l)-tensor of a vector space V is a linear map $\alpha : V^k \times (V^*)^l \to \mathbb{R}$. The space of all (k, l)-tensors is written as $T^k_l(V)$.

The tensor product of co- and contravariant tensors is naturally extended to general tensors

The bundle of all tangent (k, l)-tensors of Ω is written as $T^k_l M$, the disjoint union of $T_k\left(T_pM\right)$ for all points $p \in \Omega$.

The general (k, l)-tensors can be understood as "beasts with k covector heads and l vector heads."

Through the natural pairing between vectors and covectors, tensors can be multiplied with one another in the same way as matrices, by plugging one of the contravariant slots (a vector) into the covariant slot of another tensor. In fact, matrices are simply $(1, 1)$-tensors in the world where column vectors are vectors, and row vectors are covectors.

We sometimes use the free index notation to write tensors. In this notation, ω_i is a covariant 1-tensor with one free index, i. This is an abbreviation of the notation $\sum_i \omega_i \mathrm{d}x^i$, for a covector frame $\mathrm{d}x_i$. T^i_j is a $(1, 1)$-tensor with a covariant slot and a contravariant slot. This is an abbreviation of the notation $\sum_{i,j} T^i_j \ bx_i \otimes \mathrm{d}x^j$, for a vector frame $\mathbf{x}_i$. In both cases, the actual choice of frame does not matter, as this is simply a way to write down an abstract operator, and no index will

ever be evaluated individually. Using free index notation is an easy way to keep track of all the available slots of a (k, l)-tensor for large k and l. This is often coupled with *Einstein notation*, where summation is performed over all repeated indices: $T^i_i = \sum_i T^i_i$. Multiplication of two $(1, 1)$-tensors is written as $(XY)^i_k = X^i_j Y^j_k$ in this notation.

The operations $\sharp$ and $\flat$ can be applied to each of the indices of these tensors individually. Using free index notation, they can be conceptualized as multiplications with the *metric tensor* g_{ij} (which is the tensor corresponding to the surface metric), or its inverse g^{ij}: $\left(\omega^\sharp\right)^i = g^{ij}\omega_j$.

Definition 1.7 (Trace)**.** *[6, Appendix B]*

The trace of a $(1, 1)$*-tensor* T *is defined as*

$$\operatorname{tr} T = T^i_i \ .$$

The trace of a $(2, 0)$*- or* $(0, 2)$*-tensor is defined as applying* $\sharp$ *or* $\flat$ *to one of its indices respectively, and then computing the trace of the resulting* $(1, 1)$*-tensor. For a* $(2, 0)$*-tensor* X*:*

$$\operatorname{tr} X = g^{ij} X_{ij} \ .$$

For a $(0, 2)$*-tensor* Y*:*

$$\operatorname{tr} Y = g_{ij} Y^{ij} \ .$$

Tensor fields are defined similarly to vector and covector fields.

1.2 Differential Operators

We will now repeat the definitions of a few basic differential operators from Riemannian geometry.

Definition 1.8 (Gradient and differential)**.** *[9, Chapter 6]*

Let $f : \Omega \to \mathbb{R}$ be a smooth function. Let $\mathbf{v}$ be a tangent vector to Ω at the point $p \in \Omega$. Then the differential of f at p is the covector

$$\mathrm{d}f_p(\mathbf{v}) = D_{\mathbf{v}} f \ ,$$

where $D_{\mathbf{v}}$ is the directional derivative of a function in the direction of $\mathbf{v}$ on the surface.

The differential of f is the smooth covector field $\mathrm{d}f(p) = \mathrm{d}f_p$.

The gradient of f is the smooth vector field $\nabla f = (\mathrm{d}f)^\sharp$.

The gradient ∇f can also be seen as the restriction of the usual coordinate-wise gradient of $\mathbb{R}^3$ to the tangent space of Ω.

One can generalize the concept of an exterior derivative to higher-order tensors called differential forms.

Definition 1.9 (Differential forms)**.** *[9, Chapters 6, 12]*

A covariant tensor T is alternating, if exchanging two indices gives back the negative of the initial tensor:

$$T(\dots, \mathbf{v}_i, \dots, \mathbf{v}_j, \dots) = -T(\dots, \mathbf{v}_j, \dots, \mathbf{v}_i, \dots) \ .$$

For a vector space V, the set of all alternating covariant k-tensors is written as $\Lambda^k(V)$.

The disjoint union of $\Lambda^k(T_p\Omega)$ for every point $p \in M$ is the set of differential k-forms on Ω. differential 0-forms are simply smooth functions.

For any covectors $\omega^1, ..., \omega^k$, the wedge product *produces the alternating tensor*

$$\left(\omega^1 \wedge \cdots \wedge \omega^k\right)(X_1, ..., X_k) = \det\left(\omega^j(X_i)\right)_{ij} \ ,$$

where the notation $\left(a_{ij}\right)_{ij}$ denotes a matrix with the entry a_{ij} at the i-th row and $j-th$ column. The wedge product is easily extended to alternating covariant k-tensors. For 0-tensors, the wedge product is simply the usual product. For k-tensors with $k > 0$, the wedge-product is anticommutative.

Let $\mathrm{d}x^1, ..., \mathrm{d}x^n$ be a covector frame at a point $p \in \Omega$. Then every k-form ω at p can be written in coordinates as $\omega = \omega_{i_1,...,i_k}\mathrm{d}x^{i_1} \wedge \cdots \wedge \mathrm{d}x^{i_k}$.

Fields of differential forms are defined similarly to vector, covector and tensor fields. We will usually drop the "fields of" and simply refer to fields of differential forms as differential forms. We will also usually drop the "differential", and simply refer to fields of differential forms as forms. On a surface, there can only be 0-, 1- and 2-forms. This will greatly simplify notation going forward.

These differential forms enable multiple flavors of derivatives. Each one corresponds to a different operator we know from Calculus.

Definition 1.10 (Exterior derivative)**.** *[9, Chapter 12]*

The exterior derivative d *turns a field of differential* k*-forms into a field of differential* $k+1$ *forms such that*

1. *For* 0*-forms,* d *is the differential from Definition 1.8.*
2. *If* $\omega = \omega_i \mathrm{d}x^i$ *for a covector frame* $\mathrm{d}x^1, \mathrm{d}x^2$*, it holds* $\mathrm{d}\omega = \mathrm{d}\omega_i \wedge \mathrm{d}x^i$*.*
3. *For a* k*-form* ω *and an* l*-form* ξ*,* $\mathrm{d}(\omega \wedge \xi) = \mathrm{d}\omega \wedge \xi + (-1)^k \omega \wedge \mathrm{d}\xi$*.*
4. $\mathrm{d}\,\mathrm{d} = 0$*.*

The property that $\mathrm{d}^2 = 0$ is reminiscent both of the fact that the curl of a gradient is zero, as well as the fact that the divergence of a curl is zero. Indeed, we can write both curl and divergence using the exterior derivative, if we pair it with the codifferential.

Definition 1.11 (Codifferential)**.** *[6, Chapter 2] [7, Chapter 7]*

Let $\mathrm{d}x^1, \mathrm{d}x^2$ *be an orthonormal covector frame The* Hodge star *is defined as the map* $* : \Lambda^k \Omega \to \Lambda^{2-k}\Omega$ *such that, for a smooth function* $f : \Omega \to \mathbb{R}$*,*

$$
\begin{aligned}
*f &= f\,\mathrm{d}x^1 \wedge \mathrm{d}x^2 \\
*\left(f\,\mathrm{d}x^1\right) &= f\,\mathrm{d}x^2 \\
*\left(f\,\mathrm{d}x^2\right) &= -f\,\mathrm{d}x^1 \\
*\left(f\,\mathrm{d}x^1 \wedge \mathrm{d}x^2\right) &= f\ .
\end{aligned}
$$

For non-orthonormal frames this generalizes using the volume form, which is defined via the determinant of the coordinate matrix of the metric tensor with respect to our covector frame, $(\det g_{ij})\,\mathrm{d}x^1 \wedge \mathrm{d}x^2$*. The hodge star fulfills the relation* $\omega \wedge *\xi = (\omega \cdot \xi)(\det g_{ij})\,\mathrm{d}x^1 \wedge \mathrm{d}x^2$*.*

The codifferential δ turns a field of differential k-forms into a field of differential $k-1$ forms:

$$\delta\omega = (-1)^{(2-k)(k+1)} * \mathrm{d} * \omega \ .$$

The codifferential is adjoint to the exterior derivative via the metric and integration by parts. It holds

$$\int_\Omega \omega \cdot \mathrm{d}f \ \mathrm{d}x = \int_{\partial\Omega} f\omega(\boldsymbol{\nu}) \ \mathrm{d}x + \int_\Omega (\delta\omega) f \ \mathrm{d}x \ ,$$

where $\boldsymbol{\nu}$ is the boundary normal *of Ω (and not the surface normal).*

As far as intuition goes, the codifferential is like a divergence for covectors. Using the codifferential, the divergence of a vector field $\mathbf{v}$ can be written as $\operatorname{div}\mathbf{v} = -\delta\mathbf{v}^\flat$ [6, p. 52]. As expected, the negative divergence is adjoint to the gradient,

$$\int_\Omega \mathbf{v}\cdot\nabla f \ \mathrm{d}x = \int_\Omega \mathbf{v}^\flat \cdot \mathrm{d}f \ \mathrm{d}x = \int_{\partial\Omega} f \ \mathbf{v}\cdot\boldsymbol{\nu} \ \mathrm{d}x + \int_\Omega (\delta\mathbf{v}^\flat) f \ \mathrm{d}x = \int_{\partial\Omega} f \ \mathbf{v}\cdot\boldsymbol{\nu} \ \mathrm{d}x - \int_\Omega (\operatorname{div}\mathbf{v}) f \ \mathrm{d}x \ .$$

The curl of a vector field $\mathbf{v}$ can be written as $\operatorname{curl}\mathbf{v} = * \, \mathrm{d}\mathbf{v}^\flat$ [6, p. 52].

The last differential operator used in this thesis is the covariant derivative. It generalizes the gradient of a vector field in $\mathbb{R}^n$ to surfaces.

Definition 1.12 (Covariant derivative)**.** *[6, Chapters 4-5] [10, pp. 103]*

Given a vector frame $\mathbf{x}_1, \mathbf{x}_2$ that defines the directional derivatives ∂_i, ∂_2, the Christoffel symbols *are defined as*

$$\Gamma_{ij}^k = \frac{1}{2} g^{kl} \left(\partial_i g_{jl} + \partial_j g_{il} - \partial_l g_{ij} \right) .$$

The covariant derivative *(also called the Levi-Civita connection) maps a vector* $\mathbf{v}$ *and a covector field* ω *to the covector field* $\nabla_{\mathbf{v}}\omega$ *such that*

1. $\nabla_{f\mathbf{v}_1+\mathbf{v}_2}\omega = f\nabla_{\mathbf{v}_1}\omega + \nabla_{\mathbf{v}_2}\omega$ *for a smooth function* f*;*
2. $\nabla_{\mathbf{v}}(\alpha\omega_1 + \omega_2) = \alpha\nabla_{\mathbf{v}}\omega_1 + \nabla_b v\omega_2$ *for* $\alpha \in \mathbb{R}$*;*
3. $\nabla_{\mathbf{v}}(f\omega) = f\nabla_{\mathbf{v}}\omega + \omega\,\mathrm{d}f(\mathbf{v})$ *for a smooth function* f*;*
4. $\nabla g_{ij} = 0$ *for the metric tensor* g_{ij}*.*

With respect to the compatible vector and covector frames $\mathbf{x}_1, \mathbf{x}_2$ *and* $\mathrm{d}x^1, \mathrm{d}x^2$*, the covariant derivative is explicitly given by*

$$\nabla_{\mathbf{v}^i\mathbf{x}_i}\left(\omega_j \mathrm{d}x^j\right) = \left(\mathrm{d}\omega_k(\mathbf{v}^i\mathbf{x}_i) - \mathbf{v}^j\omega_i\Gamma^i_{kj}\right)\mathrm{d}x^k\ .$$

The covariant derivative allows the differentiation of a covector field: it measures the rate of change of that covector field along any direction.

When written solely as $\nabla\omega$ (instead of $\nabla_{\mathbf{v}}\omega$), the covariant derivative is understood as the operation that maps a covector field ω to an object that consumes a vector field $\mathbf{v}$ and returns the covector field $\nabla_{\mathbf{v}}\omega$.

Definition 1.13 (Adjoint covariant derivative)**.** *[7, Chapter 7]*

There is an operator ∇^* *that is adjoint to* ∇ *in terms of integration by parts using the metric. It maps a* $(2,0)$*-tensor to a covector. It is defined by the relation*

$$\int_\Omega X \cdot \nabla\omega\ \mathrm{d}x = \int_{\partial\Omega} \omega \cdot X \cdot \boldsymbol{\nu}\ \mathrm{d}x + \int_\Omega \nabla^* X \cdot \omega\ \mathrm{d}x\ .$$

The covariant derivative can also be defined for a vector $\mathbf{v}$ and a vector field $\mathbf{w}$. $\nabla_{\mathbf{v}}\mathbf{w}$ then is the vector field that measures the rate of change of the vector field along any direction. With respect to the compatible vector and covector frames $\mathbf{x}_1, \mathbf{x}_2$ and $\mathrm{d}x^1, \mathrm{d}x^2$, the covariant derivative is explicitly given by $\nabla_{\mathbf{v}^i\mathbf{x}_i}\left(\mathbf{w}^j\mathbf{x}_j\right) = \left(\mathrm{d}\mathbf{w}^k(\mathbf{v}^i\mathbf{x}_i) + \mathbf{v}^j\mathbf{w}^i\Gamma_{ij}^k\right)\mathbf{x}_k$ [10, p. 103]. An adjoint covariant derivative for vector fields similarly exists.

1.3 Curvature

The last important property of surfaces used in this thesis is curvature. The curvature of a surface measures how much a second-order approximation of the surface differs from the flat plane at every point.

Definition 1.14 (Shape operator)**.** *[6, Chapter 8, specifically (8.25)]*

Let $\mathbf{n}$ *be the surface normal of the surface* $\Omega \subseteq \mathbb{R}^3$ *at a point* $p \in \Omega$*. Smoothly extend it to a small open region around* $\Omega \subseteq \mathbb{R}^3$*. The* shape operator *is a* $(1,1)$*-tensor* $S : T\Omega \to T\Omega$ *defined as*

$$S(\mathbf{v}) = -D_{\mathbf{v}}\mathbf{n}\ ,$$

where $D_{\mathbf{v}}$ *is the directional derivative in the ambient space* $\mathbb{R}^3$ *in the direction* $\mathbf{v}$*.*

The shape operator describes the curvature of the surface.

Definition 1.15 (Principal and Gaussian curvature)**.** *[6, Chapter 8] [11, §30]*

The eigenvalues κ_1, κ_2 *of the shape operator* S *at a point* $p \in \Omega$ *are the two* principal curvatures*. By convention,* $|\kappa_1| \leq |\kappa_2|$*.*

The eigenvectors $\mathbf{u}_1, \mathbf{u}_2$ *associated with the respective principal curvatures are the two* principal curvature directions.

$\kappa := \kappa_1\kappa_2$ *is the* Gaussian curvature.

A surface with $\kappa = 0$ *is called a* developable surface. *It can be deformed into a surface* $\subseteq \mathbb{R}^2$ *without any distortion.*

The principal curvatures describe the inverse of the largest and smallest circles that can be fit to the surface at a point $p \in \Omega$. The principal curvature directions are the directions of these circles.

1.4 Function Spaces

In order to talk about functions on surfaces, we need to have function spaces where these functions live. The topic of function spaces and functional analysis is rich and deep, and a more in-depth introduction can be found in the books of Evans [12], Gilbarg and Trudinger [13], and Lax [14]. The simplest of these spaces is the space of smooth functions. Each definition features a reference to where the appropriate concepts can be found in these books.

Definition 1.16 (Continuous and smooth function spaces)**.** *[12, Section 1]*

$C^0(\Omega)$ *is the space of all bounded continuous functions* $f : \Omega \to \mathbb{R}$.

$C^k(\Omega)$ *is the space of all functions* $f : \Omega \to \mathbb{R}$ *such that all* k*-th order partial derivatives of* f *are in* $C^0(\Omega)$.

$C^\infty(\Omega)$ is the space of all functions $f : \Omega \to \mathbb{R}$ such that all partial derivatives of all orders are in $C^0(\Omega)$.

Definition 1.17 (Sobolev space)**.** *[12, Section 5]*

Let

$$\|f\|_p := \begin{cases} \left(\int_\Omega |f(x)|^p \, \mathrm{d}x\right)^{\frac{1}{p}} & 1 \le p < \infty \\ \sup_{x\in\Omega} |f(x)| & p = \infty \end{cases}$$

be the L^p norm, and let $L^p(\Omega) := \left\{ f : \Omega \to \mathbb{R} \;\middle|\; \|f\|_p < \infty \right\}$ be the L^p space for a surface Ω.

For vector-valued functions f, the definition is the same, except that $|\cdot|$ is the Euclidean vector norm induced by the metric, $|f| := \sqrt{g(f(x), f(x))}$.

Let $W^{k,p} := \left\{ f : \Omega \to \mathbb{R} \;\middle|\; \|D^\alpha f\|_p,\ 0 \le \alpha \le p \right\}$, where D^α is the vector of all componentwise weak derivatives of order α, be the Sobolev space. Its norm is the sum of L^p norms of all its derivatives.

Let $H^k := W^{k,2}$.

We usually drop the (Ω) for C^k, L^p and Sobolev spaces, since the surface Ω is always implied.

The L^p-space is the space of all p-integrable functions, and $L^p \subseteq L^q$ for $p \ge q$, as Ω is compact [12, p. 270]. The Sobolev space $W^{k,p}$ is the space of functions with k weak derivatives in L^p. The L^p and Sobolev spaces do not actually contain functions, to be precise, but classes of functions—if the difference of two functions has zero norm, they are considered to be the same element. For a more thorough treatment of Sobolev spaces, the reader is referred to the books of Evans [12], Gilbarg and Trudinger [13], and Lax [14].

Definition 1.18 (Sub-zero spaces)**.** *[12, Section 5]*

Every Sobolev-space $W^{k,p}$ with $k \geq 1$ has a corresponding sub-zero space $W_0^{k,p}$ (or H_0^k). The sub-zero space consists of all functions in $W^{k,p}$ that are zero at the boundary.

Since the members of $W^{k,p}$ are not technically functions, it is imprecise to define the sub-zero space, as we did above, as all functions evaluating to zero at the boundary. The mathematically precise definition defines the sub-zero spaces as the limit points of all smooth functions in $W^{k,p}$ that evaluate to zero on the boundary.

We will use the same notation for scalar-valued and vector-valued functions, relying on context to differentiate between the two. In general, scalar-valued functions have no special emphasis (u), while vector-valued functions are bold ($\mathbf{u}$).

Chapter 2: Partial Differential Equations and Their Boundary Conditions

In this chapter, we will present the well-known Poisson (Section 2.1) and Biharmonic equations (Section 2.3) and their variational formulation, as well as the concept of natural boundary conditions. We will also present a Poisson equation for tangent vectors (Section 2.2). A more in-depth discussion of these classical topics can be found in the books of Evans [12] and Gilbarg and Trudinger [13].

2.1 The Poisson Equation

2.1.1 The Laplacian

In order to formulate the Poisson equation on surfaces, a special Laplacian defined on surfaces is needed.

Definition 2.1 (Laplace-Beltrami operator)**.** *[9, p. 408] [7, p. 204]*

The Laplace-Beltrami operator Δ *operates on a twice differentiable function* f *defined on the surface* Ω*,*

$$\Delta f = \delta \mathrm{d} f = -\operatorname{div} \nabla f \ .$$

The Laplace-Beltrami operator, when restricted to a surface in the plane $\mathbb{R}^2$, is actually *minus* the coordinate-wise planar Laplacian $\left(\frac{\partial}{\partial x^1}\right)^2 + \left(\frac{\partial}{\partial x^2}\right)^2$. This is done in order to have a positive semidefinite operator. The Laplace-Beltrami operator also appears as a *negative* semidefinite operator in literature, depending on the conventions of the author.

Remark 2.1. *The Laplace-Beltrami operator for scalar functions is actually a special case of the more general Hodge-Laplacian for differential forms [7, p. 204].*

The theory of elliptic partial differential equations in general, and the Poisson equation in particular, is very extensive. We will repeat the most relevant textbook facts in this section. Definitions and theory are taken from the books of Evans [12] and Gilbarg and Trudinger [13], which are highly recommended for an in-depth exploration of the topic.

2.1.2 The strong Poisson equation

Definition 2.2 (The Poisson equation)**.** *[12, Section 3]*

Let $f \in C^\infty(\Omega)$*. The Poisson equation is the problem of finding* $u \in C^\infty(\Omega)$ *such that*

$$\Delta u = f \ .$$

The problem from Definition 2.2 is not uniquely solvable. It has to be solved over the correct space of functions in order to have a unique solution. This function space is different, depending on whether Ω has a boundary or not.

Theorem 2.1 (Solvability of the Poisson equation)**.** *If Ω has no boundary, then the Poisson equation $\Delta u = f$ is uniquely solvable over the space of all functions $u \in C^\infty$ that average to zero,*

$$\int_\Omega u \, \mathrm{d}x = 0 \, ,$$

if f averages to zero.

If Ω has a nonempty boundary, then the Poisson equation is uniquely solvable if either of the following two boundary conditions hold:

- *the value of u is 0 at every point of the boundary $\partial\Omega$ (zero* Dirichlet *boundary conditions);*
- *the value of the normal gradient of u, $\nabla u \cdot \boldsymbol{\nu}$, is 0 at every point of the boundary, u averages to zero, and f averages to zero (zero* Neumann *boundary conditions).*

Proof. See, for example, the books of Evans [12] and Gilbarg and Trudinger [13]. □

Many properties that hold for the Dirichlet and Neumann boundary conditions, such as solvability, also hold if the value of u or its normal gradient are prescribed to be a smooth enough nonzero function, so-called nonzero Dirichlet and Neumann boundary conditions [12, Part III].

2.1.3 The weak Poisson equation

In general, in the analysis of partial differential equations, one does often not deal with the *strong form* of the Poisson equation, as given in Definition 2.2. Instead, one considers the weak form of the Poisson equation,

Definition 2.3 (Weak Poisson equation)**.** *[12, Section 6]*

Let $f \in L^2$. The weak Poisson equation with zero Dirichlet boundary conditions is the problem of finding $u \in H_0^1$ such that

$$\int_\Omega \nabla u \cdot \nabla v \, \mathrm{d}x = \int_\Omega f v \, \mathrm{d}x \qquad \forall v \in H_0^1 \, .$$

Let $f \in L^2$ average to zero. The weak Poisson equation with zero Neumann boundary conditions (or in the absence of boundaries) is the problem of finding $u \in H^1$ averaging to zero such that

$$\int_\Omega \nabla u \cdot \nabla v \, \mathrm{d}x = \int_\Omega f v \, \mathrm{d}x \qquad \forall v \in H^1 \, .$$

The weak Poisson equations have the same solutions as the strong Poisson equations, if the right-hand sides f agree.

The weak Poisson equation stems from integration by parts on the strong Poisson equation. We will perform this derivation for the case of Neumann boundary conditions. Let $u \in C^\infty$ solve the strong Poisson equation with Neumann boundary conditions for $f \in C^\infty$ averaging to zero,

$\Delta u = f$. Then, for every $v \in C^\infty$,

$$\int_\Omega (\Delta u) v \, \mathrm{d}x = \int_\Omega f v \, \mathrm{d}x \, .$$

Through integration by parts, using the fact that u has zero normal gradient at the boundary,

$$\int_\Omega f v \, \mathrm{d}x = -\int_{\partial\Omega} v \nabla u \cdot \boldsymbol{\nu} \, \mathrm{d}x + \int_\Omega \nabla v \cdot \nabla u \, \mathrm{d}x = \int_\Omega \nabla v \cdot \nabla u \, \mathrm{d}x \, . \tag{2.1}$$

The function spaces can now be adjusted to match Definition 2.3. This is a standard way of deriving weak formulations of PDEs. The weak way of looking at the Poisson equation inspires the Galerkin method of finite elements.

Even though the weak method involves searching for solutions in H^1, the solutions have higher smoothness.

Theorem 2.2 (Regularity)**.** *Let $f \in H^k$. Let u solve the weak Poisson $\Delta u = f$ with Dirichlet or Neumann boundary conditions (or without any boundary). Then $u \in H^{k+2}$.*

Proof. See, for example, the book of Evans [12, Section 6.3]. □

Theorem 2.2, together with the fact that solutions to the heat equation $\frac{\partial u}{\partial t} = -\Delta u$ are smooth for $t > 0$, are the reason why the Laplacian operator is known for its smoothing capabilities. More complicated versions of this regularity theorem exist [13].

2.1.4 The Dirichlet energy

The last important concept in solving the Poisson equation is the *variational method*. It equates solving the equation with the optimization of an energy.

Definition 2.4 (Dirichlet energy)**.** *[12, Section 8]*

The Dirichlet energy *of a function $u \in H^1$ is defined as*

$$E_\nabla(u) \coloneqq \frac{1}{2}\int_\Omega \|\nabla u\|^2 \,\mathrm{d}x - \int_\Omega f u \,\mathrm{d}x$$

for a given right-hand side $f \in L^2$.

Theorem 2.3 (Minimizers of the Dirichlet energy)**.** *The minimizer $u \in H^1_0$ of E_∇ solves the Poisson equation with right-hand side f with zero Dirichlet boundary conditions.*

The minimizer $u \in H^1$ of E_∇ solves the Poisson equation with right-hand side f (averaging to zero) with zero Neumann boundary conditions.

Proof. A more complete version of this proof can be found in the book of Evans [12]. We are repeating a condensed version here to foreshadow other such upcoming calculations.

A stationary point u of E_∇ can be found by considering all smooth variations $u + hv$ of u with $h > 0$ and $v \in H^1_0, H^1$ (respectively). Then one takes the derivative with respect to h and evaluates

at $h = 0$. This gives

$$\begin{aligned} E_\nabla(u + hv) &= \frac{1}{2}\int_\Omega \|\nabla u + h\nabla v\|^2 \,\mathrm{d}x - \int_\Omega f(u + hv)\,\mathrm{d}x \\ \frac{\partial}{\partial h}E_\nabla(u + hv) &= \int_\Omega (\nabla u + h\nabla v)\cdot\nabla v \,\mathrm{d}x - \int_\Omega fv \,\mathrm{d}x \\ \frac{\partial}{\partial h}E_\nabla(u + hv)\Big|_{h=0} &= \int_\Omega \nabla u\cdot\nabla v \,\mathrm{d}x - \int_\Omega fv \,\mathrm{d}x\,. \end{aligned}$$

This is exactly the weak Poisson equation from Definition 2.3. □

The fact that, when optimized without any restrictions on the solution space, u fulfills zero Neumann boundary conditions, is an important concept.

Definition 2.5 (Natural boundary condition)**.** *[15, Chapter 1, Section 2.4]*

The boundary conditions that arise from minimizing an energy over the space of all functions are called the natural boundary conditions.

Natural boundary conditions often have a special physical meaning, as they model a state with lowest global energy. They impose the *least restrictions* when minimizing an energy. The natural boundary conditions of the Dirichlet energy are the zero Neumann boundary conditions. For many energies one can determine the natural boundary conditions by looking at the boundary terms that are produced when deriving the weak equation, such as in (2.1). Without knowing a priori that the solution fulfills zero Neumann boundary conditions, only knowing that it solves the Poisson equation, we know

$$0 = -\int_{\partial\Omega} v\nabla u\cdot\boldsymbol{\nu}\,\mathrm{d}x + \int_\Omega \nabla v\cdot\nabla u\,\mathrm{d}x - \int_\Omega fv\,\mathrm{d}x = -\int_{\partial\Omega} v\nabla u\cdot\boldsymbol{\nu}\,\mathrm{d}x \tag{2.2}$$

for all $v \in H^1$. Thus, if we plug into v small bump functions around any point $x \in \partial\Omega$, we get

$$\forall x \in \partial\Omega \qquad \nabla u(x) \cdot \boldsymbol{\nu}(x) = 0 \ ,$$

which are the zero Neumann boundary conditions.

The Dirichlet energy E_∇ of Definition 2.4 can be used to *quantify* how smooth a function is. This property is very useful in applications. One such application is data interpolation. Given interpolation data $u_1, \ldots, u_k$ at interpolation points $x_1, \ldots, x_k$, one looks for the smoothest function u (as quantified by the Dirichlet energy) fulfilling the interpolation constraints,

$$\operatorname*{argmin}_{u \in H^1} E_\nabla(u) \qquad \text{s.t. } u(x_i) = u_i \quad \forall i \ . \tag{2.3}$$

This is equivalent to finding the minimal surface fulfilling these constraints, i.e., the surface that locally minimizes its area [16, 17]. An example of such an interpolation can be seen in Figure 2.1. The right side of Figure 2.1 also reveals an undesirable property of interpolation with the Dirichlet energy: when constraining individual points, the resulting function tends to have sharp, spike-like behavior.

The property of the Dirichlet energy to quantify a function's smoothness can also be used to smooth input data. To do so, the Dirichlet smoothness energy E_∇ is combined with a second term, the fidelity term E_f, which measures the fidelity of u compared to an input function $f \in L^2$ that is to be smoothed. One solves optimization problem

$$\operatorname*{argmin}_{u \in H^1} E_\nabla(u) + \alpha E_f(u), \qquad E_f(u) \coloneqq \int_\Omega (u - f)^2 \, \mathrm{d}x \ , \tag{2.4}$$

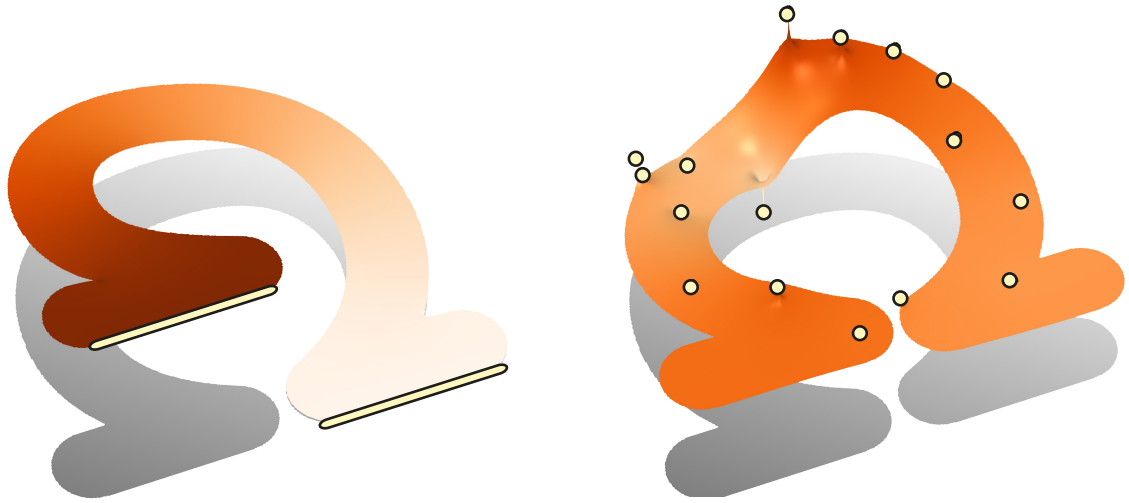


Figure 2.1. Interpolating given data (yellow with black outline) on a flat, Omega-shaped domain. The result is plotted as a height field. The two examples are: Constraining both feet of the Omega (*left*), and only constraining individual points (*right*).

where $\alpha > 0$ is the compromise between perfect smoothness ($u = \text{const}$, $\alpha = 0$) and perfect fidelity ($u = f$, $\alpha = \infty$).

In the formulation of (2.4), the solutions do not completely fulfill zero Neumann boundary conditions anymore—how much the solutions do so depends on the parameter α, as in combination with the additional fidelity energy, the boundary term from (2.2) turns into a penalty on the normal gradient at the boundary. The magnitude of the penalty depends on α, so if the relation of smoothing to fidelity is skewed towards smoothing, the relation will have very small (but nonzero) normal gradient at the boundary.

Minimizers of (2.4) for $\alpha > 0$ do not solve the Poisson equation, but the *Helmholtz equation*, which occurs in the form $\Delta u - k^2 u = f$ in the physics of sound waves (excluding countably many degenerate values of k, and ensuring the right-hand side f integrates to zero) [12, Chapter 6.2].

The property of the Dirichlet energy as a smoothness energy has led to its widespread use in geometry processing and computer graphics; a few examples are listed here. Pinkall and Polthier [16] use the Dirichlet energy to compute discrete minimal surfaces, while Desbrun *et al.* [18] use it to smooth surfaces. Baran and Popovic [19], Joshi *et al.* [20], and Weber *et al.* [21] use the smooth interpolation property of the Dirichlet energy to produce character animations. Furthermore, the Dirichlet energy is popular in image processing [22]. The Dirichlet energy also occurs in mathematics (to characterize minimal surfaces) [17], and physics (where the Dirichlet energy is a part of a model of the vibration of an idealized drum) [23].

2.2 The Vector Poisson Equation

Using the covariant derivative instead of the gradient and the adjoint covariant derivative instead of the divergence, we can define a vector Laplacian.

Definition 2.6 (Bochner Laplacian)**.** *[7, Chapter 7]*

The Bochner Laplacian Δ *operates on a twice differentiable vector field* $\mathbf{u}$ *defined on the surface* Ω*,*

$$\Delta\mathbf{u} \coloneqq \nabla^*\nabla\mathbf{u} \ .$$

Almost exactly like with the scalar Laplacian, there is a corresponding Bochner-Poisson equation $\Delta\mathbf{u} = \mathbf{f}$, a weak formulation $\int_\Omega \nabla\mathbf{u} \cdot \nabla\mathbf{v} \ \mathrm{d}x = \int_\Omega \mathbf{v} \cdot \mathbf{f} \ \mathrm{d}x \quad \forall\mathbf{v} \in H^1$, and a vector Dirichlet energy.

Definition 2.7 (vector Dirichlet energy)**.** *[24]*

Figure 2.2. Smoothing a noisy vector field (*left*) a little bit (*left*, $\alpha = 667$), and a lot (*right*, $\alpha = 167$), using the vector Dirichlet energy.

The vector Dirichlet energy *of a vector-valued function* $\mathbf{u} \in H^1$ *is defined as*

$$E_\nabla(\mathbf{u}) \coloneqq \frac{1}{2}\int_\Omega \|\nabla \mathbf{u}\|^2 \,\mathrm{d}x - \int_\Omega \mathbf{f} \cdot \mathbf{u} \,\mathrm{d}x$$

for a given right-hand side $\mathbf{f} \in L^2$.

The vector Dirichlet energy's natural boundary conditions are the vector zero Neumann boundary conditions, $\nabla \mathbf{u} \cdot \boldsymbol{\nu} = 0$. It can be used for smoothing and scalar data interpolation just like the scalar Dirichlet energy. Figure 2.2 shows the vector Dirichlet energy being used to smooth a vector field with an energy minimization like the one of (2.4), but with vectorial versions of all the operators.

While the vector Dirichlet energy is less prolific than the scalar Dirichlet energy, it still has many uses in computer graphics. Knöppel *et al.* [25] and Liu *et al.* [26] use the vector Dirichlet energy to design smooth vector fields. In Knöppel *et al.* [24], the vector Dirichlet energy is used

to design stripe patterns on surfaces. Custers and Vaxman [27] use the vector Dirichlet energy to compute surface parametrizations. Vector-Dirichlet-like energies are also used to *mesh* surfaces, such as in the work of Jakob *et al.* [28]. The vector Dirichlet energy is used to transport vectors along surfaces in the work of Sharp *et al.* [29]. Further applications can be found in the works of Azencot *et al.* [30, 31] and Corman and Ovsjanikov [32].

2.3 The Biharmonic Equation

2.3.1 Strong and weak biharmonic equation

Definition 2.8 (The biharmonic equation)**.** *[12, p. 6.6.2]*

Let $f \in C^\infty(\Omega)$. The Biharmonic equation is the problem of finding $u \in C^\infty(\Omega)$ such that

$$\Delta^2 u = f \ .$$

Being a fourth-order equation, the biharmonic equation admits more than one set of boundary conditions. As will be later discussed in this thesis, there is also more than one weak formulation of the biharmonic equation.

Theorem 2.4 (Solvability of the biharmonic equation)**.** *The biharmonic equation $\Delta^2 u = f$, $f \in L^2$, is uniquely solvable with zero Dirichlet and zero Neumann boundary conditions (if Ω has a boundary), or if u, f average to 0 (it Ω has no boundary).*

The weak version of this boundary value problem is: find $u \in H_0^2$ such that

$$\int_\Omega \Delta u \, \Delta v \, \mathrm{d}x = \int_\Omega v f \, \mathrm{d}x \qquad \forall v \in H_0^2 \, . \tag{2.5}$$

Proof. We will first show that solving the weak formulation (2.5) is the same as solving the biharmonic equation with zero Dirichlet and Neumann boundary conditions.

Assume that u solves the biharmonic equation, is smooth and vanishes at the boundary. Then, the strong biharmonic equation implies that for all smooth v vanishing at the boundary,

$$\int_\Omega \Delta^2 u \, v \, \mathrm{d}x = \int_\Omega f v \, \mathrm{d}x \, .$$

Applying integration by parts twice gives

$$\begin{aligned} \int_\Omega \Delta^2 u \, v \, \mathrm{d}x &= -\int_{\partial\Omega} v \nabla \Delta u \cdot \boldsymbol{\nu} \, \mathrm{d}x + \int_\Omega \nabla \Delta u \cdot \nabla v \, \mathrm{d}x \\ &= \int_{\partial\Omega} (\Delta u \, \nabla v - v \, \nabla \Delta u) \cdot \boldsymbol{\nu} \, \mathrm{d}x + \int_\Omega \Delta u \, \Delta v \, \mathrm{d}x \, . \end{aligned} \tag{2.6}$$

The boundary term involving v vanishes, since v vanishes at the boundary. For the other boundary term, one can do yet another integration by parts (this time on the domain $\partial\Omega$, which has no boundary), to get $\int_{\partial\Omega} \Delta u \, \nabla v \cdot \boldsymbol{\nu} \, \mathrm{d}x = \int_{\partial\Omega} \Delta v \, \nabla u \cdot \boldsymbol{\nu} \, \mathrm{d}x$. By the zero Neumann boundary condition on u, this term is also zero. Thus, this weak formulation is equivalent to the strong formulation.

To show unique solvability of the weak form, as well as more complete versions of the argument presented in this proof, see the literature: for the case of a surface with boundary, see the book of Evans [12, Problem 6.6.2] or Gazzola *et al.* [33]; for the case of a surface without boundary, the

biharmonic equation decouples into two successive Poisson equations, and Theorem 2.1 applies.

□

2.3.2 The Laplacian energy

The weak formulation from Theorem 2.4 gives rise to one particular energy whose minimizers solve the biharmonic equation.

Definition 2.9. *[23] [18]*

The Laplacian energy *is defined as*

$$E_\Delta(u) \coloneqq \frac{1}{2}\int_\Omega |\Delta u|^2 \, \mathrm{d}x \; .$$

Minimizing E_Δ over H_0^2 is equivalent to solving the weak formulation from (2.5). Zero Neumann boundary conditions can also be enforced independently of the zero Dirichlet boundary conditions.

In this form, the Laplacian energy is very useful for solving interpolation problems, similar to (2.3) [34]. To interpolate the data $u_1, \cdots, u_k$ at points $x_1, \cdots, x_k$, one solves the optimization problem

$$\operatorname*{argmin}_{u \in H^2} E_\Delta(u) \qquad \text{s.t. } u(x_i) = u_i \quad \forall i \; . \tag{2.7}$$

Pointwise interpolation with the biharmonic equation does not exhibit the spikes at the interpolation points that the Dirichlet energy does, as can be seen in Figure 2.3.

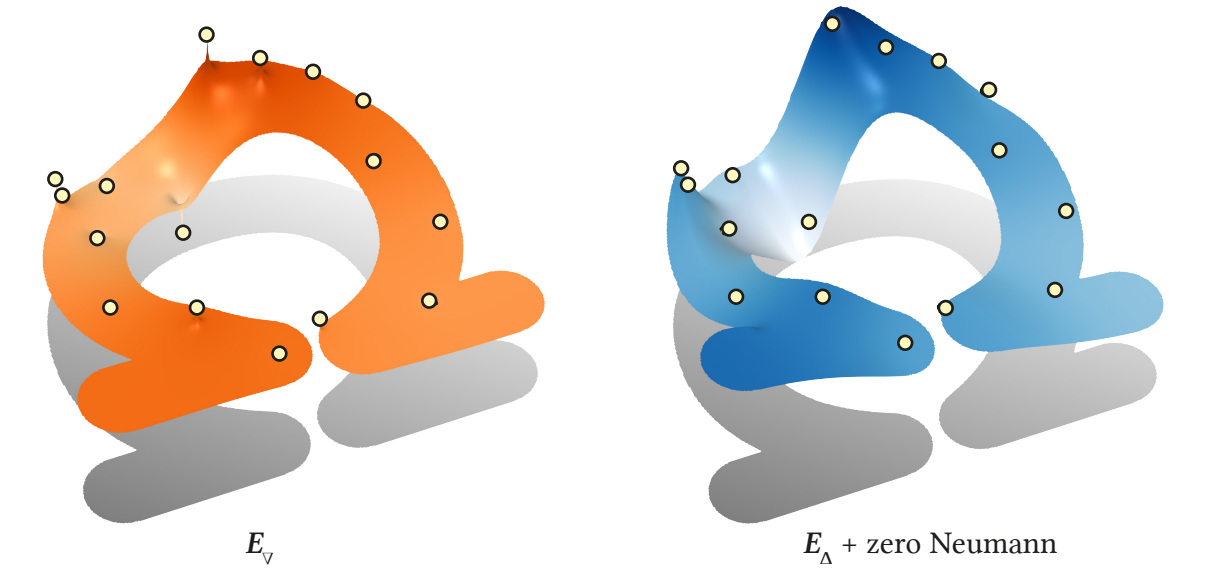


Figure 2.3. Top row: while the Dirichlet energy (*left*) exhibits spikes when interpolating at isolated points, the Laplacian energy with zero Neumann boundary conditions is smooth across interpolation points (*right*). Bottom row: interpolation can be used to generate weight functions to drive an animation technique called linear blend skinning—the spiky behavior of E_∇ results in bad deformations (*middle*), which does not happen with E_Δ and zero Neumann boundary conditions (*right*). (baby image and experiment courtesy of Jacobson *et al.* [35]).

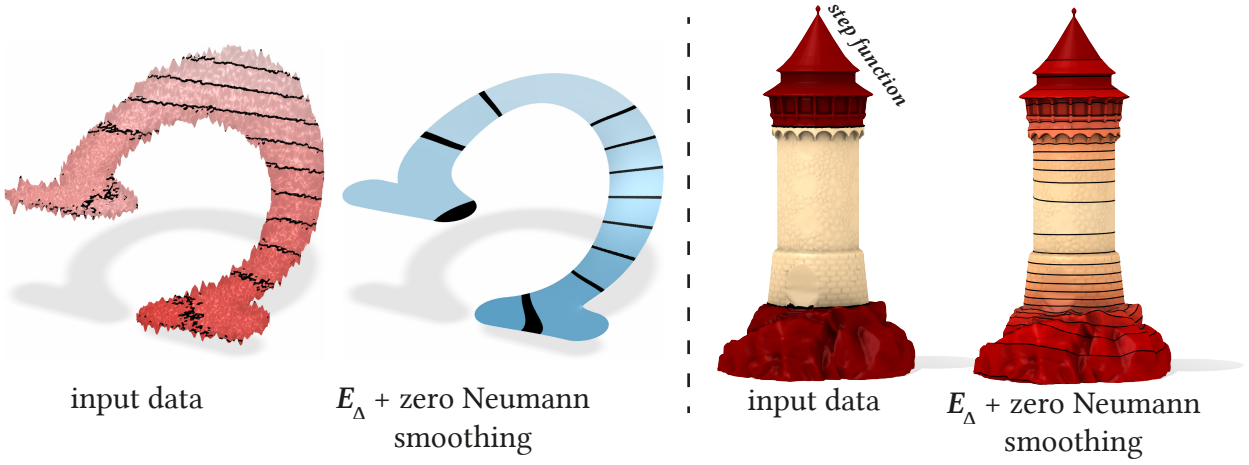


Figure 2.4. Smoothing a noisy input function over a flat, Omega-shaped domain using the Laplacian energy with zero Neumann boundary conditions; the functions are shown as height fields (*left*). Smoothing a noisy input function over a curved, tower-shaped domain; here the domain is curved, and the functions are *not* plotted as a height field (*right*).
Isolines of the functions are rendered as black lines.

Using the approach of (2.4), the Laplacian energy with zero Neumann boundary conditions can be used for smoothing noisy input data [36]. Given an input function f to be smoothed, one solves the optimization problem

$$\operatorname*{argmin}_{u \in H^2} E_\Delta(u) + \alpha E_f(u), \qquad E_f(u) := \int_\Omega (u - f)^2 \, \mathrm{d}x \,, \tag{2.8}$$

where $\alpha > 0$ is the compromise between perfect smoothness ($u = \text{const}$, $\alpha = 0$) and perfect fidelity ($u = f$, $\alpha = \infty$). An example of such Laplacian smoothing can be seen in Figure 2.4.

The Laplacian energy with zero Neumann boundary conditions is also common in geometry processing. Data smoothing with biharmonic functions has been performed, for example, by Weinkauf *et al.* [36]. Desbrun *et al.* [18] use the Laplacian energy to smooth surfaces. In the works of Sorkine *et al.* [37], Zhou *et al.* [38], and Andrews *et al.* [39], the Laplacian energy with zero Neumann boundary conditions is used to smoothly deform surfaces. Further applications

include the computation of geodesic distance [40], surface modeling [41], character animation [35, 42] and more [43, 44, 45, 46, 47, 48, 49, 50, 51]. In this formulation, one does *not* impose Dirichlet boundary conditions when minimizing the energy E_Δ (or modifications of it), but only zero Neumann boundary conditions, $\operatorname{argmin}_{u \in H^2} E_\Delta(u) \quad \text{s.t. } \frac{\partial u}{\partial \mathbf{n}}\Big|_{\partial\Omega} = 0$. This results in the natural boundary condition (2.9), which we will describe in Theorem 2.5, applying.

To end this section, we will talk about the natural boundary conditions of the Laplacian energy. Zero Dirichlet and zero Neumann boundary conditions are *not* the natural boundary conditions of the Laplacian energy.

Theorem 2.5. *The natural boundary conditions of the Laplacian energy E_Δ, which emerge when minimizing over H^2, are*

$$\nabla \Delta u \cdot \boldsymbol{\nu} = 0 \quad \text{on } \partial\Omega \tag{2.9}$$

$$\Delta u = 0 \quad \text{on } \partial\Omega \tag{2.10}$$

Proof. These boundary conditions go back to at least Courant and Hilbert [23]. We will present a short argument here. For simplicity, we will assume that $u, v \in C^\infty$. By the derivation in (2.6), the boundary terms that have to be zero for a stationary point u are

$$\int_{\partial\Omega} (\Delta u \, \nabla v - v \, \nabla \Delta u) \cdot \boldsymbol{\nu} \, \mathrm{d}x = 0 \qquad \forall v \in C^\infty \, .$$

It is possible to make sense of these boundary conditions by testing with different families of functions $v \in C^\infty$. Consider all functions v that have zero normal derivative at the boundary, i.e., $\nabla v(x) \cdot \boldsymbol{\nu}(x) = 0 \quad \forall x \in \partial\Omega$. Such functions definitely exist for a smooth boundary $\partial\Omega$, and, by restriction, these can represent every smooth function w on $\partial\Omega$. It is thus possible to conclude

that

$$\int_{\partial\Omega} w\nabla\Delta u \cdot \boldsymbol{\nu}\, \mathrm{d}x = 0 \quad \forall w \in C^\infty(\partial\Omega) \qquad \Rightarrow \qquad \nabla\Delta u(x) \cdot \boldsymbol{\nu}(x) = 0 \quad \forall x \in \partial\Omega\ .$$

On the other hand, consider all functions v that are zero at the boundary. By restriction, the normal gradient of these functions can represent every smooth function w in $\partial\Omega$. One can thus conclude that

$$\int_{\partial\Omega} \Delta u\, w\, \mathrm{d}x = 0 \quad \forall w \in C^\infty(\partial\Omega) \qquad \Rightarrow \qquad \Delta u(x) = 0 \quad \forall x \in \partial\Omega\ .$$

Picking the correct test families v might seem a bit arbitrary in this case. This picking process is definitely a bit of an art form, and one must make sure that the derived boundary conditions are really all-encompassing, otherwise the derived boundary conditions might be too restrictive and not always true (see the book of Courant and Hilbert [23]). □

Chapter 3: The Hessian Energy

In this section we present the novel Hessian energy for curved surfaces and show its applicability in a wide variety of computer graphics applications. Unlike in the introductory chapters, the ideas presented in this section are new (except where indicated).

3.1 Bias at the Boundary

Section 2.3 shows why smoothing energies based on the biharmonic equation in general, and E_Δ with zero Neumann boundary conditions specifically, are good choices for smoothing energies in a variety of geometry processing applications such as scattered data interpolation and data smoothing. The zero Neumann boundary conditions, however, introduce a significant bias to minimizers of the Laplacian energy. Because of the zero Neumann boundary conditions, isolines of minimizers have to be *perpendicular* to the boundary, which can result in significant distortion of the solution near the boundary. This boundary bias is visible both in smoothing (Figure 3.1) and scattered data interpolation (Figure 3.2).

While zero Neumann boundary conditions might be desirable for some applications, where the boundary has a physical meaning (for example, when modeling the flow of water in a pipe, where the water can't physically leave the pipe), it is undesirable for many others: perhaps the

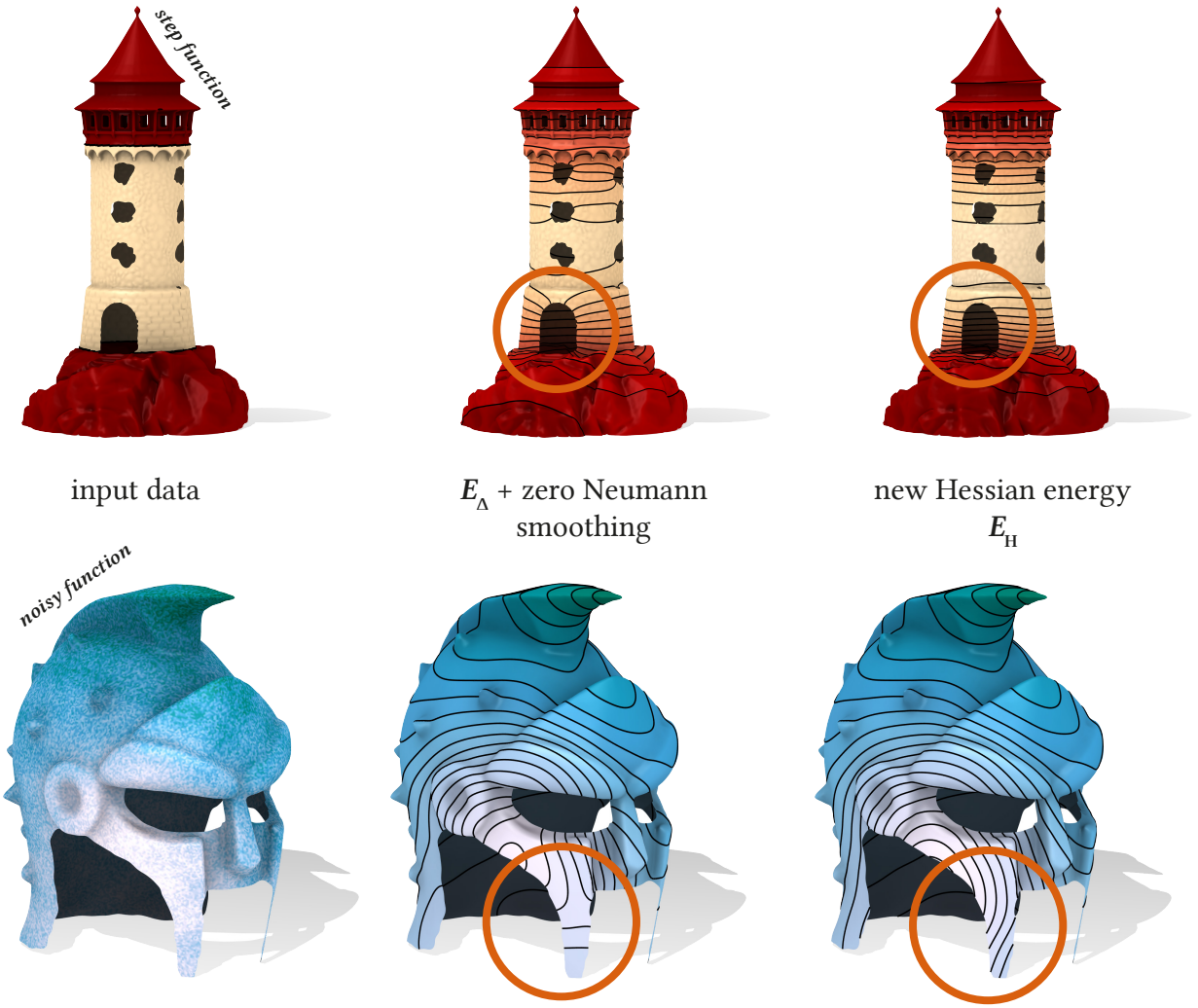


Figure 3.1. A function (*left*) is smoothed using the energy E_Δ with zero Neumann boundary conditions (*middle*), as well as the new Hessian energy $E_{\mathbf{H}}$ (*right*). The zero Neumann boundary conditions lead to significant bias of the isolines near the boundary, as highlighted by the orange circles. The Hessian energy $E_{\mathbf{H}}$ does not suffer from this, its minimizers' isolines are better at ignoring the shape of the boundary.

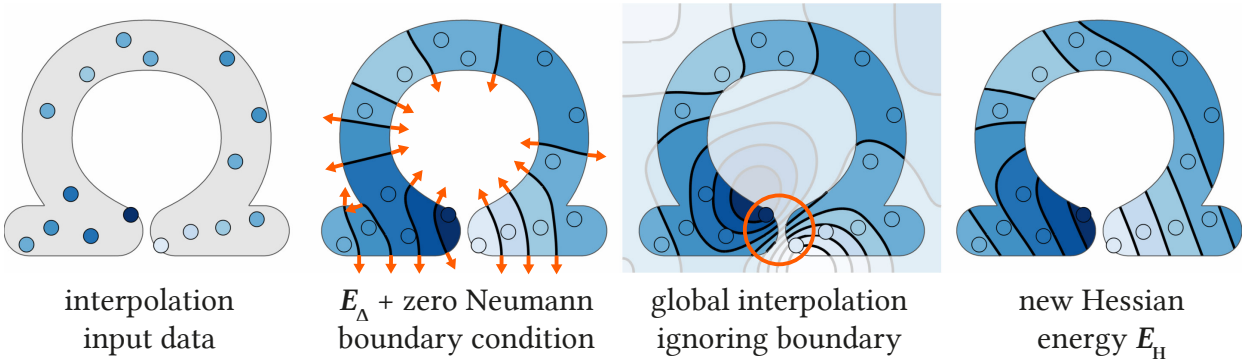


Figure 3.2. Solving an interpolation problem (*far left*) by minimizing an energy. Minimizing E_Δ with zero Neumann boundary conditions (*center left*) leads to a solution that is biased towards the boundary—all isolines are perpendicular to it. Ignoring the boundary and simply interpolating in ambient space (*center right*) leads to the two feet of the Omega, far apart intrinsically, suddenly being close in ambient space, and a bleeding effect across the gap occurs. The minimizer of the new Hessian energy (*far right*) is less biased by the boundary, and does not bleed across the gap in the Omega.

boundary of the domain is an artifact from noisy surface registration (such as a laser scan), or perhaps the boundary of the domain represents areas where no information could be collected, and not an actual physical barrier. In such scenarios, and others, users want to avoid the boundary biasing the result, i.e., small changes in the boundary should not change the result of the energy optimization by much.

The two most obvious solutions to the boundary bias problem fail to produce a satisfying result. For the interpolation problem on flat surfaces, one could conceivably simply ignore the boundary of the domain and interpolate the given data over all of $\mathbb{R}^2$. As is shown in Figure 3.2, this does not work: if the domain Ω is not convex, two parts of the domain can be far away from each other within the domain, but very close to each other in ambient space. A simple ambient interpolation leads to these far away parts of the domain influencing each other, which leads to the bleeding effect observed in Figure 3.2. An alternative idea would be to remove the

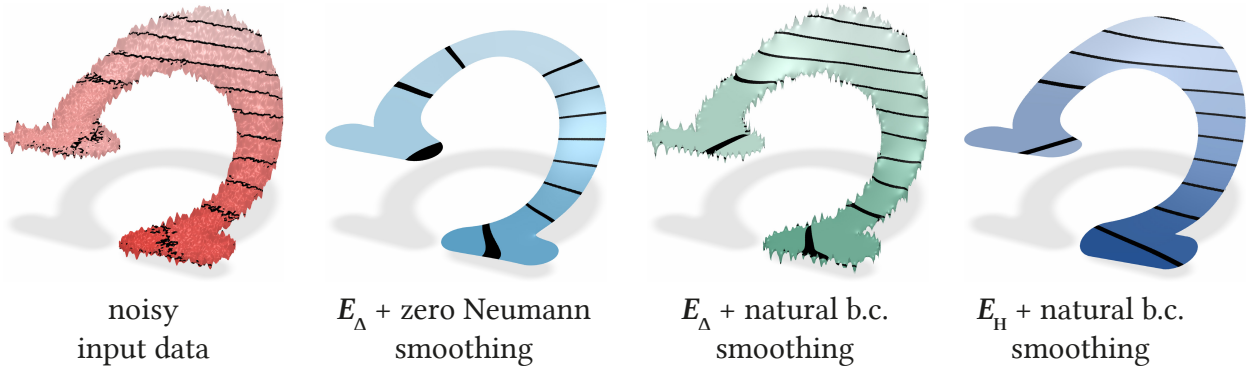


Figure 3.3. Smoothing noisy input data on a flat, Omega-shaped domain (*far left*); the result is plotted as a height field. Minimizing E_Δ with zero Neumann boundary conditions (*center left*) leads to a lot of bias at the boundary, as isolines have to be perpendicular to it. E_Δ with its natural boundary conditions (*center right*) is too permissive: the noise at the boundary is not smoothed at all. The new Hessian energy (*far right*) is less biased by the boundary, and not too permissive.

artificial imposition of zero Neumann boundary conditions and minimize the Laplacian energy with respect to its natural boundary conditions from Theorem 2.5. However, in this case, it turns out that the Laplacian energy's natural boundary conditions are *not restrictive enough*, as can be seen in Figure 3.3. The condition $\Delta u = 0$ on $\partial\Omega$ lets a lot of noise slip through the cracks, as every single harmonic function measures zero energy, and there are a lot of noisy, high-frequency harmonic functions.

What is really needed is a different set of boundary conditions that can be used for smoothing and interpolation that reduce the bias by the boundary, while still providing actually smooth solutions. To achieve this, we introduce the natural boundary conditions of the Hessian energy for the biharmonic equation, also called the *as-linear-as-possible* boundary conditions. These boundary conditions are obtained by minimizing a new kind of energy, the Hessian energy for curved surfaces.

3.2 A Different Energy for the Biharmonic Equation

Minimizers of the Laplacian energy E_Δ, which comes from the weak formulation in (2.5), have natural boundary conditions that are too permissive, as can be seen in Figure 3.3. E_Δ, however, is not the only energy whose minimizers solve the biharmonic equation. By performing different integrations by parts, it is possible to derive a second energy whose minimizers solve the biharmonic equation (and thus the desired biharmonic behavior in the interior of the shape), but that has natural boundary conditions that are better suited for a smoothing energy. A flat $\mathbb{R}^2$ version of this energy appears as far back as the book of Courant and Hilbert [23].

Lemma 3.1 (An alternative weak formulation for the biharmonic equation)**.** *Consider the following weak problem: find $u \in H^2$ such that*

$$\int_\Omega \nabla \,\mathrm{d}\, u \cdot \nabla \,\mathrm{d}\, v + \kappa \;\mathrm{d}\, u \cdot \mathrm{d}\, v \;\mathrm{d}x = \int_\Omega v f \;\mathrm{d}x \qquad \forall v \in H^2 \,. \tag{3.1}$$

Solutions u to the above weak problem solve the biharmonic equation.

Proof. Again, for simplicity, we will assume that u as well as the test function v are smooth, without loss of generality.

Let u be a smooth function solving the biharmonic equation. Then, for all smooth test functions v,

$$\begin{aligned}\int_\Omega f v \;\mathrm{d}x &= \int_\Omega (\Delta\Delta u)\, v \;\mathrm{d}x = \int_\Omega (\delta \,\mathrm{d}\, \delta \,\mathrm{d}\, u)\, v \;\mathrm{d}x \\ &= -\int_{\partial\Omega} (\mathrm{d}\, \delta \,\mathrm{d}\, u)\,(\boldsymbol{\nu})\, v \;\mathrm{d}x + \int_\Omega (\mathrm{d}\, \delta \,\mathrm{d}\, u)\,(\mathrm{d}\, v) \;\mathrm{d}x \,,\end{aligned} \tag{3.2}$$

by integration by parts, where $\boldsymbol{\nu}$ is the oriented boundary normal.

On can now utilize the Weitzenböck identity. It states that $\mathrm{d}\,\delta + \delta\,\mathrm{d} = \Delta + \mathrm{Ric} = \nabla^*\nabla + \mathrm{Ric}$, where Ric is the so-called Ricci curvature tensor [7, Chapter 7].[1] Together with the fact that $\mathrm{d}^2 = 0$, (3.2) gives

$$\begin{aligned}\int_\Omega (\mathrm{d}\,\delta\,\mathrm{d}\,u)\,(\mathrm{d}\,v)\ \mathrm{d}x &= \int_\Omega ((\mathrm{d}\,\delta + \delta\,\mathrm{d})\,\mathrm{d}\,u)\,(\mathrm{d}\,v)\ \mathrm{d}x \\ &= \int_\Omega (\nabla^*\nabla\,\mathrm{d}\,u)\,(\mathrm{d}\,v) + \mathrm{Ric}(\mathrm{d}\,u, \mathrm{d}\,v)\ \mathrm{d}x \\ &= -\int_{\partial\Omega} \boldsymbol{\nu}^i\,(\nabla\,\mathrm{d}\,u)_{ij} \cdot (\mathrm{d}\,v)_j\ \mathrm{d}x \\ &\quad + \int_\Omega (\nabla\,\mathrm{d}\,u) \cdot (\nabla\,\mathrm{d}\,v) + \mathrm{Ric}(\mathrm{d}\,u, \mathrm{d}\,v)\ \mathrm{d}x\,,\end{aligned} \tag{3.3}$$

where free index notation has been added to make it clear which part of the operator is multiplied with what.

The Ricci curvature tensor Ric can be further simplified. For surfaces, which are two-dimensional manifolds, the Ricci curvature tensor can be written as

$$\mathrm{Ric} = \kappa g\,, \tag{3.4}$$

where κ is the Gaussian curvature and g is the metric tensor [7, pp. 38–41].

Combining (3.2), (3.3), and (3.4) gives

$$\int_\Omega f v\ \mathrm{d}x = -\int_{\partial\Omega} (\mathrm{d}\,\delta\,\mathrm{d}\,u)\,(\boldsymbol{\nu})\,v + \boldsymbol{\nu}^i\,(\nabla\,\mathrm{d}\,u)_{ij} \cdot (\mathrm{d}\,v)_j\ \mathrm{d}x + \int_\Omega (\nabla\,\mathrm{d}\,u) \cdot (\nabla\,\mathrm{d}\,v) + \kappa\ \mathrm{d}\,u \cdot \mathrm{d}\,v\ \mathrm{d}x\,.$$

[1] The actual definition of Ric, a $(0, 2)$-tensor, does not matter in this context, as it will disappear in one of the next steps of the calculation.

This is the weak equation from (3.1), except for the boundary terms, which will control behavior of the minimizer at the boundary when set to zero (and about which the lemma says nothing). □

The weak formulation in Lemma 3.1 gives rise to the following energy.

Definition 3.1 (Hessian energy)**.** *The* Hessian energy *is defined as*

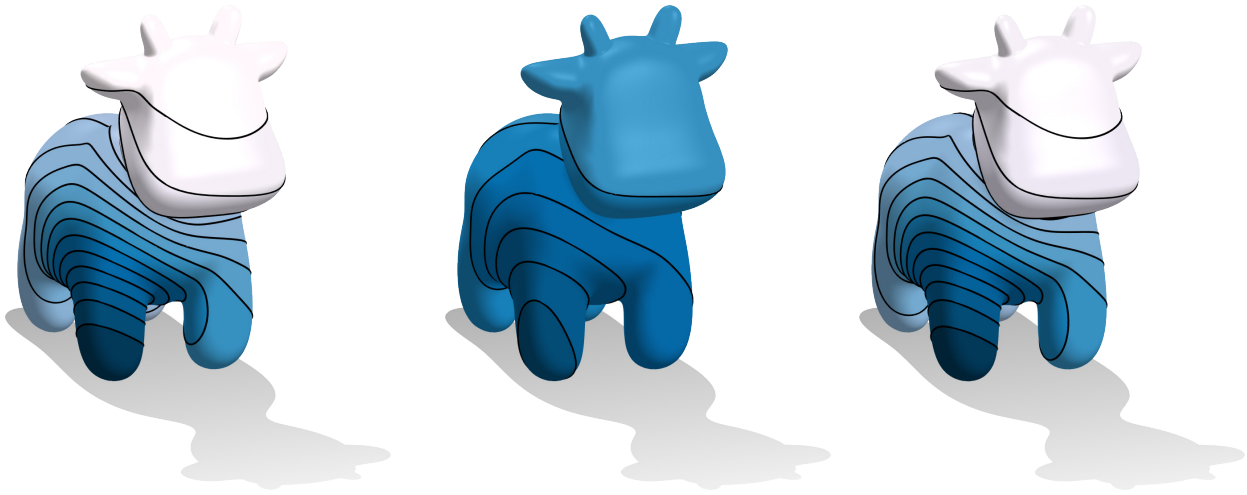
$$E_{\mathbf{H}}(u) \coloneqq \frac{1}{2} \int_{\Omega} |\nabla \,\mathrm{d}\, u|^2 + \kappa \, |\mathrm{d}\, u|^2 \,\mathrm{d}x \,,$$

where the norm $|\cdot|$ *is defined as* $|x|^2 \coloneqq x \cdot x$.

This is called the Hessian energy, because the operator ∇d is the Hessian in differential geometry [6, p. 45]. On planar surfaces, it corresponds to the standard 2×2 Hessian matrix. By the usual variational argument, minimizers of the Hessian energy $E_{\mathbf{H}}$ solve the weak equation (3.1), and thus the biharmonic equation.

On flat surfaces $\subseteq \mathbb{R}^2$, the Gaussian curvature κ is zero everywhere, so only the term $|\nabla du|^2$ remains. For flat surfaces, this is the same as the Frobenius norm of the Hessian matrix for u, $|\nabla du|^2 = \mathrm{Hess}(u)^\top \mathrm{Hess}(u)$. In this flat form, the Hessian energy appears in the work of Courant and Hilbert [23]. It is important not to forget the Gaussian curvature term containing κ on curved surfaces (and consider the Hessian part alone), as this would *not* correctly solve the biharmonic equation (see Figure 3.4).

The drawback of minimizing E_Δ with its natural boundary conditions is that there are too many functions that are in its null space—all harmonic functions. The null space of $E_{\mathbf{H}}$ can contain, at most, all linear functions on a surface, and there are only three.



E_Δ | E_H without curvature term | E_H with curvature term

Figure 3.4. Solving the biharmonic equation with an arbitrary right-hand side by minimizing the energies E_Δ (*left*), $E_{\mathbf{H}}$ without the κ term (*center*), and $E_{\mathbf{H}}$ with the κ term. In the absence of boundaries, they should all produce the same result if they solve the same PDE. This shows that $E_{\mathbf{H}}$ without the κ term is wrong—this term is needed.

3.3 The Hessian Energy Natural Boundary Conditions

The Hessian energy $E_{\mathbf{H}}$ defined in Definition 3.1 has different natural boundary conditions than the Laplacian energy E_Δ.

Theorem 3.1 (Hessian natural boundary conditions)**.** *The natural boundary conditions of the Hessian energy $E_{\mathbf{H}}$, which emerge when minimizing over H^2, are*

$$0 = \boldsymbol{\nu}^i \left(\nabla \, \mathrm{d}\, u\right)_{ij} \boldsymbol{\nu}^j = \left(\nabla \, \mathrm{d}\, u\right)(\boldsymbol{\nu}, \boldsymbol{\nu}) \quad \text{on } \partial\Omega \tag{3.5}$$

$$0 = \left(\mathrm{d}\, \delta \, \mathrm{d}\, u\right)(\boldsymbol{\nu}) + \delta^j_{\partial\Omega} \, \iota_{\partial\Omega} \left(\boldsymbol{\nu}^i \left(\nabla \, \mathrm{d}\, u\right)_{ij}\right) \quad \text{on } \partial\Omega \,, \tag{3.6}$$

where $\delta_{\partial\Omega}$ is the codifferential restricted to the boundary, and $\iota_{\partial\Omega}$ is the restriction to the boundary $\partial\Omega$.

Proof. For simplicity, we will assume that $u, v \in C^\infty$.

As with the derivation of the Laplacian energy natural boundary conditions in Theorem 2.5, the boundary terms that have to be zero for stationary points u of $E_{\mathbf{H}}$ come from the weak formulation,

$$0 = \int_{\partial\Omega} (\mathrm{d}\,\delta\,\mathrm{d}\,u)(\boldsymbol{\nu})\,v + \boldsymbol{\nu}^i\,(\nabla\,\mathrm{d}\,u)_{ij} \cdot (\mathrm{d}\,v)_j\,\mathrm{d}x \qquad \forall v \in C^\infty\ . \tag{3.7}$$

As in the proof of Theorem 2.5, we will probe these boundary terms with specially crafted test functions $v \in C^\infty$. Start by assuming that v is zero on the boundary $\partial\Omega$, and it has arbitrary normal derivative $(\mathrm{d}\,v)(\boldsymbol{\nu}) = g \in C^\infty(\partial\Omega)$. Then (3.7) becomes

$$\begin{aligned} 0 &= \int_{\partial\Omega} \boldsymbol{\nu}^i\,(\nabla\,\mathrm{d}\,u)_{ij}\,(g\boldsymbol{\nu}^j)\,\mathrm{d}x \qquad \forall g \in C^\infty(\partial\Omega) \\ 0 &= \boldsymbol{\nu}^i(x)\,(\nabla\,\mathrm{d}\,u)_{ij}\,(x)\boldsymbol{\nu}^j(x) = (\nabla\,\mathrm{d}\,u)\,(x)(\boldsymbol{\nu}(x), \boldsymbol{\nu}(x)) \qquad \forall x \in \partial\Omega\ , \end{aligned}$$

which is the boundary condition from (3.5).

The second special testing function is a v that has zero normal derivative at the boundary, i.e., on the boundary itself, $v = g \in C^\infty(\partial\Omega)$, and $\mathrm{d}\,v = \mathrm{d}_{\partial\Omega}\,g$. The subscript on $\mathrm{d}_{\partial\Omega}$ denotes that the external derivative happens on the boundary manifold $\partial\Omega$. For this test function, (3.7) becomes

$$0 = \int_{\partial\Omega} g\,(\mathrm{d}\,\delta\,\mathrm{d}\,u)\,(\boldsymbol{\nu}) + \boldsymbol{\nu}^i\,(\nabla\,\mathrm{d}\,u)_{ij} \cdot (\mathrm{d}_{\partial\Omega}\,g)_j\ \mathrm{d}x \qquad \forall g \in C^\infty\ .$$

By integration by parts on the boundary $\partial\Omega$,

$$\int_{\partial\Omega} \boldsymbol{\nu}^i \left(\nabla \,\mathrm{d}\, u\right)_{ij} \cdot \left(\mathrm{d}_{\partial\Omega}\, g\right)_j \;\mathrm{d}x = \int_{\partial\Omega} g\, \delta^j_{\partial\Omega}\, \iota_{\partial\Omega} \left(\boldsymbol{\nu}^i \left(\nabla \,\mathrm{d}\, u\right)_{ij}\right) \;\mathrm{d}x \;,$$

where $\delta_{\partial\Omega}$ is the codifferential restricted to the boundary, and $\iota_{\partial\Omega}$ is the restriction to the boundary $\partial\Omega$. This gives the boundary condition from (3.6). □

These boundary conditions have been known for a long time for flat surfaces, where they were explored by Courant and Hilbert [23, pp. 206–207] in the context of vibrating plates. They did, however, not look at curved surfaces. For vibrating plates, the natural boundary conditions from Theorem 3.1 model a plate that is not restricted at all on its boundary, while the zero Dirichlet and zero Neumann boundary conditions model a plate that is fixed in place and clamped at the boundary. For this reason, in applications, minimizers of the Hessian energy $E_{\mathbf{H}}$ with natural boundary conditions are *less biased by the boundary* than minimizers of the Laplacian energy E_Δ with zero Neumann boundary conditions.

We want to especially highlight the first of the Hessian's natural boundary conditions, (3.5), $(\nabla \,\mathrm{d}\, u)(\boldsymbol{\nu}, \boldsymbol{\nu}) = 0$. It means that the Hessian of u, evaluated in the boundary normal direction, is zero, or that the second derivative of u across the boundary, $\frac{\partial^2 u}{\partial \boldsymbol{\nu}^2}$, is zero. Because of this, we call these boundary conditions *as-linear-as-possible*. Contrast this to the zero Neumann boundary conditions, which specify that the first derivative of u across the boundary, $\frac{\partial u}{\partial \boldsymbol{\nu}}$, is zero. These are the *as-constant-as-possible* boundary conditions. This is illustrated in Figure 3.5.

The fact that the null space of E_Δ contains all harmonic functions, means, in PDE language, that the biharmonic equation with E_Δ natural boundary conditions is only uniquely solvable when disregarding all harmonic functions (in the same way that, in order to solve the harmonic equation

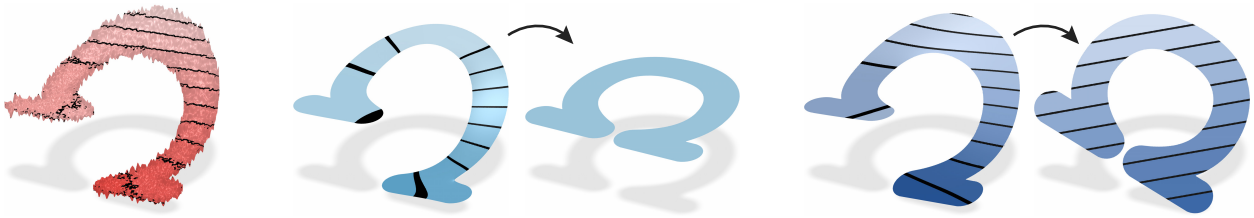


Figure 3.5. Taking a noisy input function (*left*) and smoothing it by minimizing E_Δ with zero Neumann boundary conditions, leads to, in the limit of extreme smoothing and zero fidelity, a constant function (*center*): the boundary conditions are *as-constant-as-possible*. The extreme smoothing limit for $E_\mathbf{H}$ with natural boundary conditions is a linear function (*right*): the boundary conditions are *as-linear-as-possible*. As a result, $E_\mathbf{H}$ is less biased by the boundary.

with zero Neumann boundary conditions, one must disregard all constant functions). The space of harmonic functions is infinite-dimensional. The biharmonic equation with $E_\mathbf{H}$ natural boundary conditions is uniquely solvable if all linear functions are disregarded. The space of all linear functions is at most three-dimensional, so interpolation problems are well-defined after only three points are fixed, and the excess error coming from E_Δ natural boundary conditions in Figure 3.3 is not an issue. Figure 3.6 shows the lowest-eigenvalue eigenfunctions for both E_Δ and $E_\mathbf{H}$ with natural boundary conditions. While the first few eigenfunctions for E_Δ are already high-frequency harmonic functions, $E_\mathbf{H}$ starts with only constant and linear functions, and then moves on to other low-frequency functions.

3.4 Applications

Every application for smoothness energies that was appropriate for the Laplacian energy E_Δ with zero Neumann boundary conditions is a potential application for the Hessian energy $E_\mathbf{H}$ with

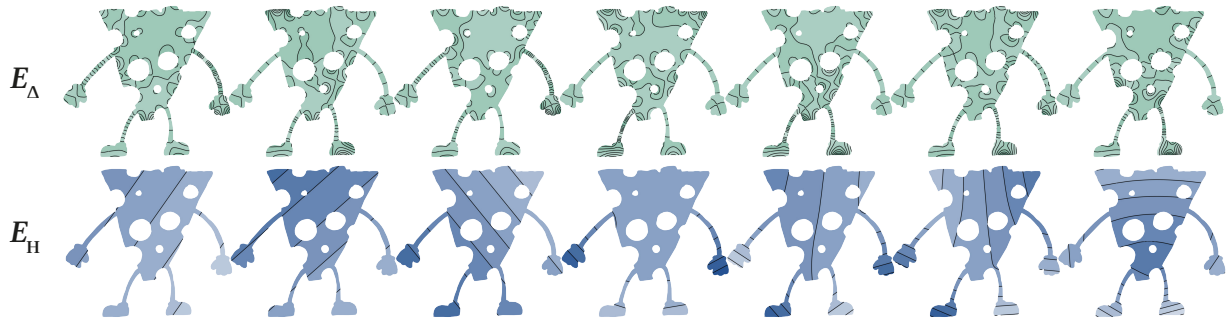


Figure 3.6. The seven lowest-eigenvalue eigenfunctions of E_Δ (*top*) and $E_\mathbf{H}$ (*bottom*) with natural boundary conditions. The first eigenfunctions for E_Δ are very high-frequency, which means that many high-frequency features will go unsmoothed with this energy. The first eigenfunctions for $E_\mathbf{H}$ are affine and other low-frequency functions, which means minimizing $E_\mathbf{H}$ will successfully extract low-frequency features.

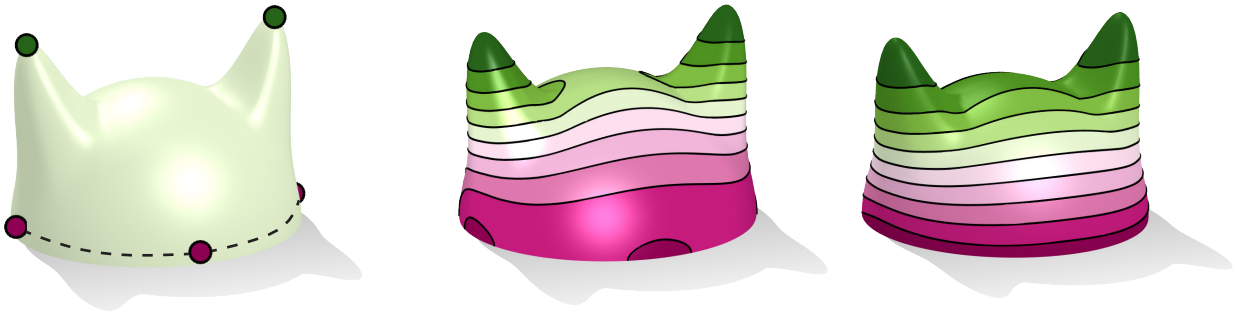
natural boundary conditions. We will explore a few, and contrast the two energies. This list is not final; many more applications can be found in literature (see Section 3.5).

Discretizations of the Hessian energy are discussed in Chapter 6. Figures appearing in this section have been discretized with the methods discussed there.

3.4.1 Data smoothing

The Hessian energy with its natural boundary conditions can be used to smooth data, whether that be noisy inputs that need to be denoised, or any kind of nonsmooth function that should be smoothed. Smoothing is achieved by solving the optimization problem

$$\operatorname*{argmin}_{u \in H^2} E_\mathbf{H}(u) + \alpha E_f(u), \qquad E_f(u) \coloneqq \int_\Omega (u - f)^2 \, \mathrm{d}x \,, \tag{3.8}$$

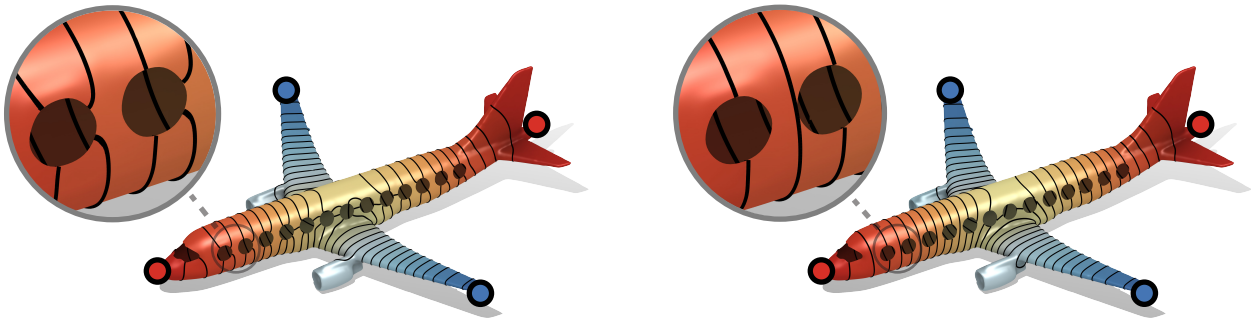


data to interpolate | E_Δ + zero Neumann b.c. | E_H + natural b.c.

Figure 3.7. Solving an interpolation problem on a Viking helmet. The goal here is to preserve the dashed line (which is almost a geodesic) connecting three data points of the same value (*left*). E_Δ with zero Neumann boundary conditions distorts the line near the boundary, as the zero Neumann boundary conditions make the isolines perpendicular to the boundary (*center*). The Hessian energy $E_\mathbf{H}$ correctly accounts for the curvature of the surface and does not suffer from bias at the boundary, interpolating the dashed line as desired (*right*).

where $\alpha > 0$ is the compromise between perfect smoothness ($u = \text{const}$, $\alpha = 0$) and perfect fidelity ($u = f$, $\alpha = \infty$). This approach is employed, for example, by Weinkauf *et al.* [36] using a Laplacian energy.

Examples of the Hessian energy used for smoothing can be seen in Figures 3.1 and 3.3. In these examples it can be seen that the minimizers of E_Δ with zero Neumann boundary conditions are warped by the presence of boundaries. Isolines twist near boundaries so that they can be parallel to them, often distorting the function in the process. This happens much less to the minimizers of $E_\mathbf{H}$ with natural boundary conditions: they try to ignore the presence of boundaries. Figure 3.9 is particularly striking: interpolating on a surface, then adding holes to the surface and interpolating again *massively* changes the minimizer of E_Δ with zero Neumann boundary conditions, while it does not affect the minimizer of $E_\mathbf{H}$ with natural boundary conditions much at all. These examples show that $E_\mathbf{H}$ with natural boundary conditions is a better choice than currently available smoothness energies if the goal is to avoid being biased by the boundary of the surface.



E_Δ + zero Neumann b.c. E_H + natural b.c.

Figure 3.8. Solving an interpolation problem on an airplane. Minimizing E_Δ with zero Neumann boundary conditions distorts the straight isolines of the minimizer near the windows and cockpit of the plane: the isolines bend in order to be perpendicular to the boundary. $E_{\mathbf{H}}$ does not suffer from this problem, its isolines are better at ignoring the holes in the domain.

3.4.2 Scattered data interpolation

Another application of the Hessian energy with its natural boundary conditions is the problem of scattered data interpolation. To interpolate given data $u_1, \dots, u_k$ at points $x_1, \dots, x_k$, one solves

$$\operatorname*{argmin}_{u \in H^2} E_{\mathbf{H}}(u) \qquad \text{s.t. } u(x_i) = u_i \quad \forall i \,. \tag{3.9}$$

Such a method is used, for example, by Jacobson *et al.* [34] with a Laplacian energy.

In contrast to the Laplacian energy E_Δ with zero Neumann boundary conditions, minimizers of the Hessian energy $E_{\mathbf{H}}$ with its natural boundary conditions are less biased by the boundary. As can be seen in Figures 3.7 and 3.8, the minimizers of E_Δ with zero Neumann boundary conditions are very heavily biased by the boundary: near the boundary, their isolines bend so that they can be perpendicular to it. The Hessian energy does not suffer from this: its minimizers are much less biased by the shape of the boundary, and interpolate the function as if the boundary was not there.

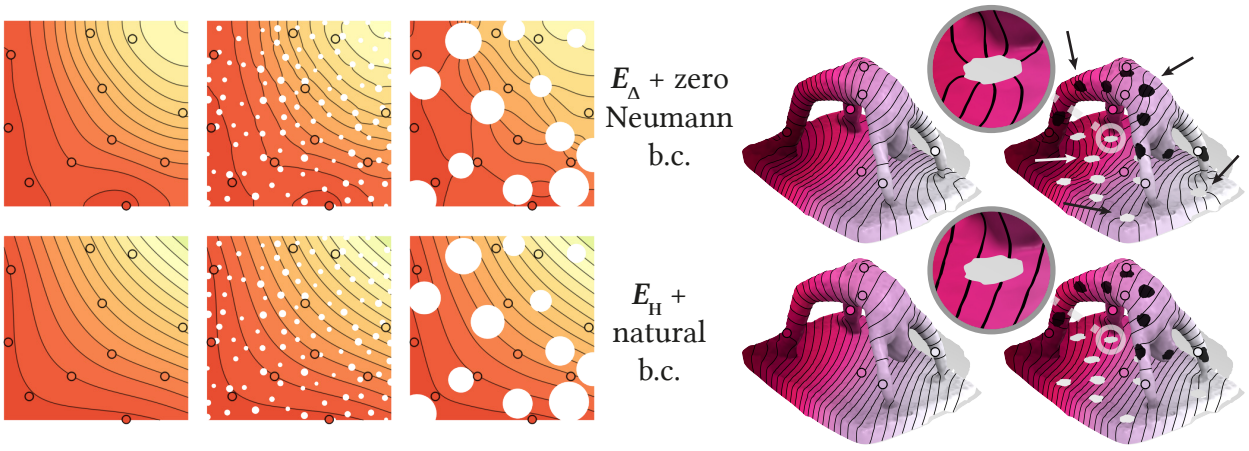


Figure 3.9. Interpolating the same scattered data distribution over surfaces without holes, and the same surfaces with holes added. The minimizers of E_Δ with zero Neumann boundary conditions change significantly when holes are added to the domains. The minimizers of $E_\mathbf{H}$ do not change as much—$E_\mathbf{H}$ is much less biased by boundaries.

3.4.3 Character animation

The problem of animating a character given a set of handles can be related to a series of scattered data interpolation problems using the technique of linear blend skinning. In the particular method of Jacobson *et al.* [35], one selects a series of handles $p_1, \ldots, p_k$ to manipulate, and then solves for deformation weight functions $u_1, \ldots, u_k$:

$$\begin{aligned}
&\operatorname*{argmin}_{u_i} E_\Delta(u_i) \quad \text{with zero Neumann boundary conditions} \\
&u_i(p_j) = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases} \\
&\sum_i u_i \equiv 1 \\
&u_i \geq 0 \quad \forall i \,.
\end{aligned}$$

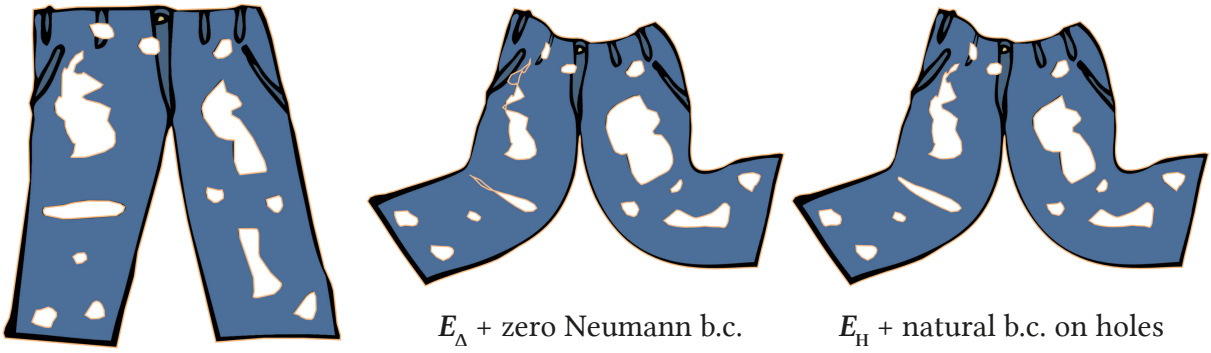


Figure 3.10. Linear blend skinning deformation using bounded biharmonic weights computed with E_Δ and zero Neumann boundary conditions (*center*) and $E_{\mathbf{H}}$ with natural boundary conditions on the interior holes (*right*). $E_{\mathbf{H}}$ causes less distortion in interior holes.

Given a linear deformation P_i to each handle, the deformation applied to each point x is $\sum_i u_i(x)P_i(x)$.

While one might want to apply zero Neumann boundary conditions on the outside of a character being defirmed (since the boundary there has actual semantic meaning), if the character has spurious boundaries, such as holes inside the surface that should be ignored when deforming, zero Neumann boundary conditions can produce bad results. In such situations, one can apply the Hessian energy's natural boundary conditions on boundaries that ought to be ignored, leading to better deformations (Figure 3.10).

3.4.4 L^1 Optimization

There is another special application of a Hessian energy for flat surfaces. The Hessian energy of Definition 3.1 is an L^2-style Hessian energy, since the integral is reminiscent of the L^2 norm, $\left(\int_\Omega |u|^2 \, \mathrm{d}x\right)^{\frac{1}{2}}$. One could also concieve of an L^1-style Hessian energy, reminiscent of the L^1 norm $\int_\Omega |u| \, \mathrm{d}x$. Such L^1 energies are considered, for example, in the work of Bronstein *et al.* [52].

noisy input E_{H} smoothing $E_{\mathrm{H},1}$ smoothing

Figure 3.11. Smoothing a heightfield with the L^2 Hessian energy $E_{\mathbf{H}}$ removes creases, while the L^1 Hessian energy $E_{\mathbf{H,1}}$ concentrates nonsmoothness in creases.

Definition 3.2 (L^1 Hessian energy)**.** *Consider a bounded domain $U \subseteq \mathbb{R}^2$. Let H_u be the 2×2 Hessian matrix of the function $u \in C^\infty(U)$, and let $|H_u|_1$ be the sum of the absolute values of its entries.*

The Hessian energy of u is defined as

$$E_{\mathbf{H,1}}(u) \coloneqq \int_U |H_u|_1 \, \mathrm{d}x \, .$$

Minimizers of the L^1 Hessian energy $E_{\mathbf{H,1}}$ will *not* solve the biharmonic equation. They will, however, inherit a kind of local as-linear-as-possible behavior. Minimizers of $E_{\mathbf{H,1}}$ are affine on parts of the domain, with discontinuous breaks between them. This is reminiscent of the L^1 norm in $\mathbb{R}$, which is linear everywhere except at the origin, where a discontinuity occurs.

The interesting effects of the $E_{\mathbf{H,1}}$ on height fields can be seen in Figure 3.11. Instead of smoothing the entire heightfield, the L^1 optimization flattens parts of it and concentrated nonsmoothness in lines. An L^1-style optimization of the Laplacian energy results in a prickly heightfield, as nonsmoothness is concentrated in isolated points (see Figure 3.12).

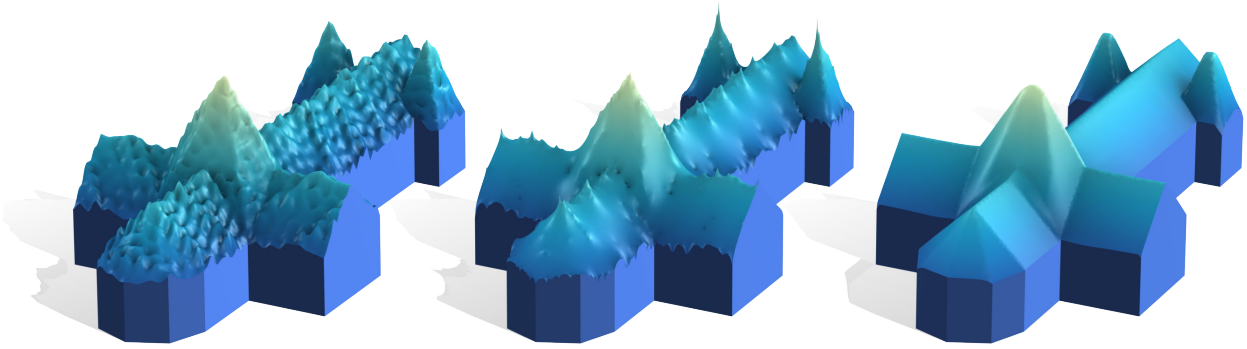


Figure 3.12. Smoothing a heightfield representing a cathedral roof with L^1 optimization. A L^1 Hessian will concentrate nonsmoothness on creases, while a L^1 Laplacian will concentrate it on isolated points, resulting in a prickly appearance.

By naively generalizing the discretization of the 2×2 Hessian matrix for flat surfaces to curved surfaces in $\mathbb{R}^3$, one can even compute an L^1-type flow for surfaces, where an optimization of $E_{\mathbf{H,1}}$ is repeatedly applied to the coordinate functions of the surface. This leads to a piecewise flat surface (see Figure 3.13).

$E_{\mathbf{H,1}}$ is discretized in Appendix A.

3.5 The Hessian Energy in Literature

The Hessian of a function has been the focus of previous geometry processing work [53, 54], but none of these works have introduced the Hessian energy or formulated it as an energy minimization problem to solve the biharmonic equation.

New boundary conditions that are less biased by the boundary than zero Neumann are explored by Wang *et al.* [42], but they are only discrete and not derived from the continuous biharmonic

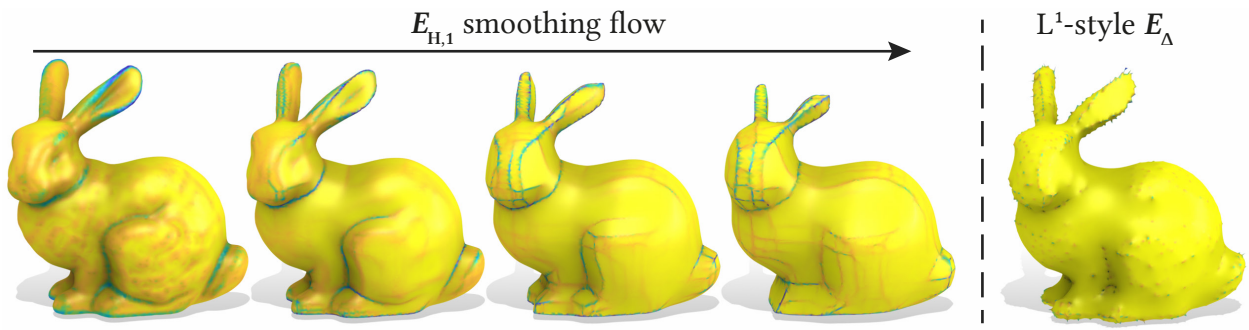


Figure 3.13. Computing an $E_{\mathbf{H},\mathbf{1}}$ flow by repeatedly minimizing the energy of the coordinate functions. The local energy is plotted. The surface flows to a piecewise flat surface, with energy concentrated on creases. Minimizing a L^1-type flow for $E_{\mathbf{H}}$ leads to the energy concentrating at points and a spiky shape.

equation or a continuous energy. Other alternative energies can be found in the works of Fisher *et al.* [55] and Bergou *et al.* [56].

In unbounded flat $\mathbb{R}^2$, a Hessian-like energy is minimized by thin-plate splines. These thin-plate splines are useful basis functions for a variety of unbounded scattered data interpolation problems [57], and are also used for shape deformation [58]. In these applications, however, the boundary conditions are not important, and thus this work does not derive or discuss them.

In image processing, $E_{\mathbf{H}}$ is popular for image denoising, restoration, inpainting, enhancement, and more. Many of these works apply non-natural boundary conditions [59, 60, 61, 62, 63, 64], but a few also use the natural boundary conditions of the Hessian energy by Courant and Hilbert [23] (but not the general variant for curved surfaces) [63, 65, 59, 64]. These works operate on a flat domain in $\mathbb{R}^2$. L^1-like norms of both the Hessian and Laplacian are also popular in image processing [66, 67, 68, 69, 70].

Finally, Hessian-like energies are used to process high-dimensional data [71, 72, 73], and in a variety of physics applications such as elastic beam theory and Kirchhoff plate theory [74, 75].

Chapter 4: The Developability Energy

Developable surfaces are surfaces with zero Gaussian curvature κ (Definition 1.15). A developable surface Ω can be deformed into a flat surface $\subseteq \mathbb{R}^2$ without any portion. This, of course, also means that any developable surface can be assembled entirely by deforming flat surfaces without any distortion. The property of assembly from flat pieces makes developable surfaces interesting for manufacturing: flat sheet materials such as paper, plywood or tin sheets are relatively inexpensive and easy to cut, ship and store; surfaces comprised of developable pieces help reduce cost and improve quality in computer-controlled milling [76]. Nevertheless, there are not many tools for automatic conversion of arbitrary curved surfaces into developable pieces. Most industrial applications still rely on manual interaction and designer expertise [77].

One goal of this thesis is to develop new mathematical and computational tools for the approximation of general surfaces by surfaces that are piecewise developable, both in the continuous and discrete settings, using specific smoothness energies. This chapter focuses on the continuous side, while Chapter 7 focuses on the discrete side.

Some definitions and facts from previous work are repeated in this section, and indicated as such by references. The remaining material is new work.

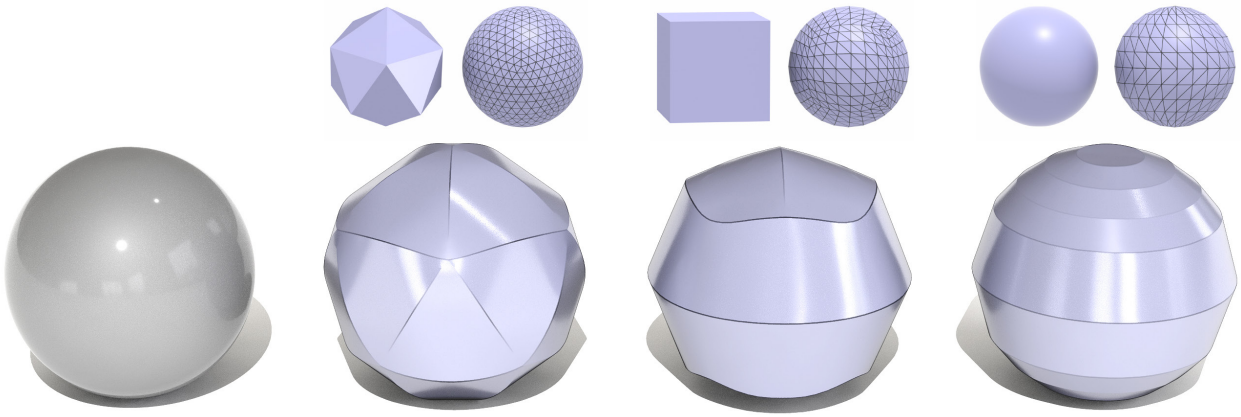


Figure 4.1. Different piecewise developable approximations for the same input sphere. These approximations were computed with different discretizations from Chapter 7. The different initial surfaces can be seen in the top row.

4.1 Developable Surface Approximation

How does one approximate a surface with a developable surface, and, more precisely, with a piecewise developable surface? It is important to note that there is no clear best approximation of a given smooth surface, since one can always find successively closer approximations of smaller and smaller developable pieces, like a crumpled piece of paper or corrugated cardboard. Nevertheless, we will introduce mathematical ways to express what a good approximation is, and the resulting discrete applications will specify how to find one of the many applicable approximations.

By Definition 1.15, a smooth manifold is developable if it has zero Gaussian curvature κ, and a developable surface can be flattened without any distortion. For the sense of correctly characterizing developability, we will introduce a special definition of developability that depends on the smoothness of the surface.

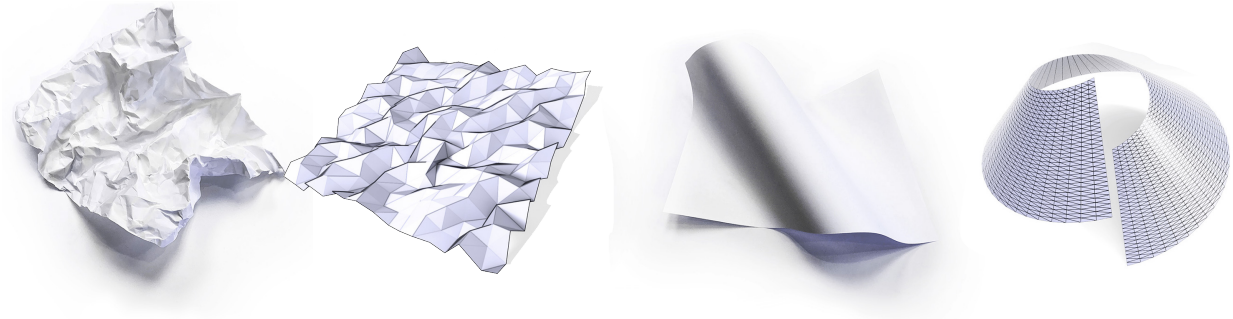


Figure 4.2. Flattenability is not enough to ensure easy fabrication. Both the crumpled piece of paper (*far left*) and the noisy discrete surface (*center left*) are flattenable, but would be very difficult to assemble from flat materials. The smooth flattenable piece of paper (*center right*) and discrete surface (*far right*) are flattenable and smooth, and thus *developable* The also have straight *ruling lines* passing through each point on the surface. These properties make developable surfaces desirable.

Definition 4.1 (Flattenable & developable)**.** *A continuous surface* $\Omega \subseteq \mathbb{R}^3$ *is* flattenable *if it is locally isometric to* $\mathbb{R}^2$ *(i.e., it can be flattened without distortion).*

A flattenable surface $\Omega \subseteq \mathbb{R}^3$ *is* developable *if its inclusion function* $f : \Omega \to \mathbb{R}^3$ *is an immersion and the surface twice differentiable.*[1]

Remark 4.1. *The concepts in Definition 4.1 are not novel, and can be found under different names in the literature [11, §30]. We restate them here, name them unambiguously, and draw a sharp distinction between the two. This will be important later.*

For smooth surfaces, this distinction is irrelevant, as all smooth surfaces are twice differentiable, and developability is the same as flattenability. A surface must also be at least twice differentiable for the Gauss curvature to be well-defined; such a surface is flattenable and developable. This distinction between flattenable and developable is, however, important in practice. While one might think that flattenable surfaces are desirable for fabrication, since they can be assem-

[1] A differentiable function is an immersion if its Jacobian matrix has maximal rank (the dimension of Ω) everywhere.

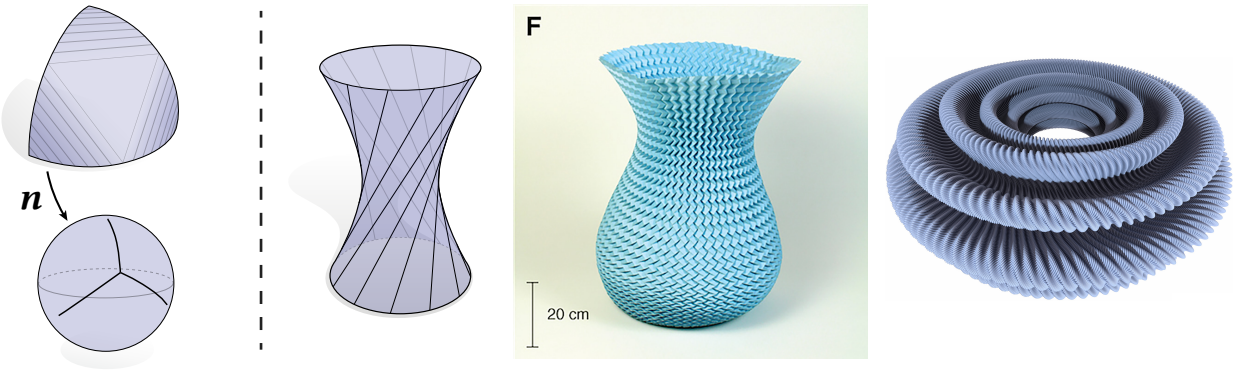


Figure 4.3. Developable surfaces have *ruling lines* along which the surface normal **n** is constant; the set of all normals **n** on the Gauss sphere is a collection of arcs (*left*).
Merely requiring ruling lines is not enough to get a developable surface, as surfaces like the hyperboloid of revolution are curved and ruled; merely requiring flattenability allows for corrugated and wrinkled shapes that are not developable (*right*). (Images [78, 79])

bled from flat pieces, this is not the case. It is really *developability* that is important, as flattenable surfaces can be all kinds of degenerate (see Figure 4.2 and the right side of Figure 4.3).

An additional important property of developable surfaces are ruling lines.

Definition 4.2 (Ruled surface)**.** *[11, §30]*

A surface is ruled if it is parametrized by a smooth collection of straight lines, the ruling lines*.*

Smooth developable surface are always ruled in their non-flat regions: the ruling lines follow the principal curvature direction of the smaller principal curvature, κ_1, as the surface is not curved in this direction and can thus be described as a straight line there. The map which maps the surface normal at every point **n**, to the unit sphere (the *Gauss map*), maps a developable surface to a network of curves meeting in the regions where the surface is flat ($\kappa_1 = \kappa_2$). While all developable surfaces are ruled, not all ruled surfaces are developable, and not all flattenable surfaces are ruled (see Figure 4.3).

4.2 An Energy to Quantify Developability

In order to approximate a surface by a developable approximation, it is useful to quantify the developability of a surface with a developability energy, the same way the Laplacian and Hessian energies quantify smoothness in Section 2.3 and Chapter 3. A natural idea is to measure the L^2 norm of the Gaussian curvature of the surface, since developable surfaces have zero Gaussian curvature,

$$E_\kappa(\Omega) \coloneqq \frac{1}{2}\int_\Omega \kappa^2 \, \mathrm{d}x \, . \tag{4.1}$$

Unlike the quadratic energies E_Δ and $E_{\mathbf{H}}$, it is not possible to find a minimum by solving a simple linear equation—E_κ is not quadratic. One can minimize such an energy by trying to find places where its gradient is zero. Flowing along the gradient of E_κ towards a minimum will push the surface Ω to have smaller Gaussian curvature κ, achieving developability when $E_\kappa(\Omega) = 0$. The trajectory of the gradient is important, though, as the trajectory of the flow determines the geometry of the final surface (there is no unique minimizer of E_κ, as there are many surfaces with $\kappa = 0$). Since κ is a product of the principal curvatures κ_1, κ_2, a near-hemispherical cap will, by symmetry, flow to an almost perfectly flat disk (see Figure 4.4).

Flowing a near-hemispherical cap to a flat disk is a little bit aggressive for the purposes of approximation. E_κ, however, inspires a different way to quantify developability. By definition, $|\kappa_1| \leq |\kappa_2|$, so the Gaussian curvature κ will vanish if and only if $\kappa_1 = 0$. This leads to the definition of the novel *developability energy*.

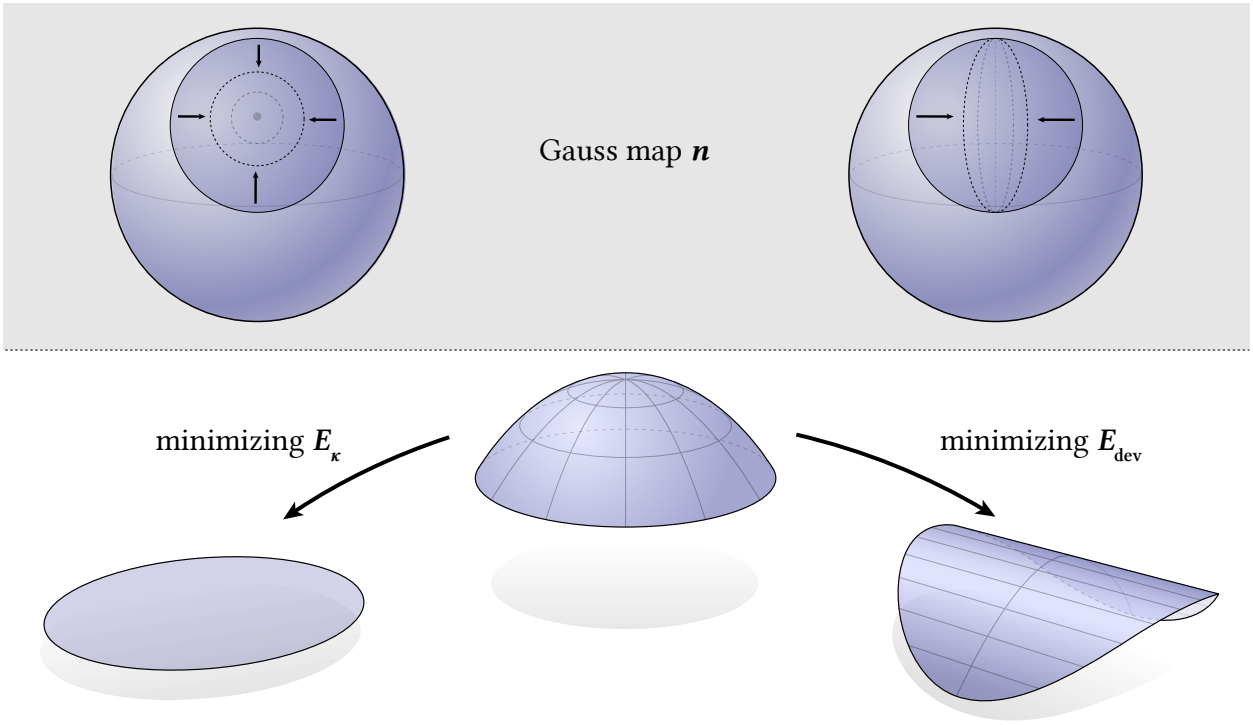


Figure 4.4. Minimizing E_κ (penalizing the Gaussian curvature κ) encourages both principal curvatures to shrink, transforming a near-hemispherical cap to a flat patch (*left*). In contrast, minimizing E_{dev} (penalizing the smaller principal curvature κ_1) gives a bent surface patch, which more closely approximates the original surface.

Definition 4.3 (Developability energy)**.** *The* developability energy *of a surface* $\Omega \subseteq \mathbb{R}^3$ *is defined as*

$$E_{dev}(\Omega) \coloneqq \frac{1}{2}\int_\Omega \kappa_1^2 \, \mathrm{d}x \, .$$

This new developability energy measures the developability of a smooth surface by the failure of the principal curvature directions at each point to look like ruling lines. As $\kappa = 0 \Leftrightarrow \kappa_1 = 0$, if $E_{\text{dev}}(\Omega) = 0$ then Ω must be developable. Gradient flow of E_{dev} is much less aggressive than gradient flow of E_κ: only one of the principal curvatures, κ_1, needs to be reduced (see Figure 4.4). The flow of E_{dev} remains closer to the original surface.

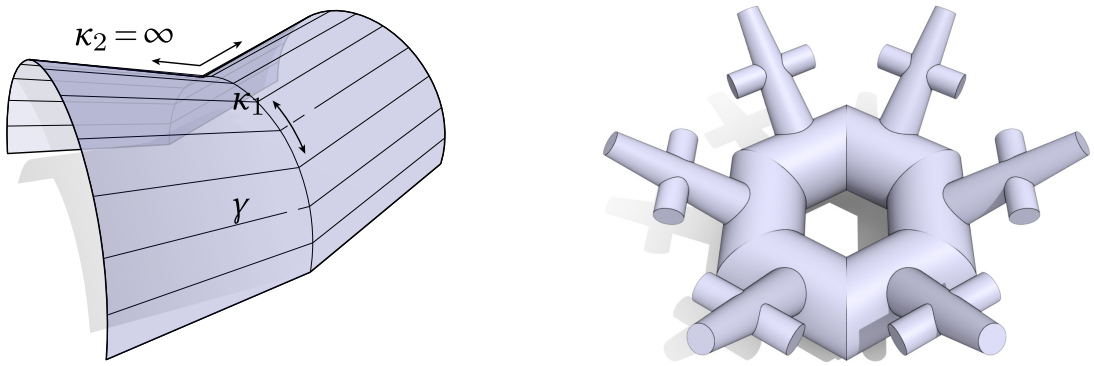


Figure 4.5. Since E_{dev} penalizes only the *smaller* principal curvature, it is exactly zero for two developable patches meeting along a seam of zero area γ (*left*), as well as entire collections of such patches (*right*).

Regularization Since E_{dev} has many degenerate minimizers, as discussed above, one possible mitigation strategy is adding an explicit regularizer. Possible ideas are the Willmore energy $\frac{1}{4}\int_\Omega (\kappa_1 + \kappa_2)^2 \, \mathrm{d}x$ [49], or the Gaussian curvature energy E_κ. Adding such a regularizing term, weighted by $\varepsilon > 0$ chosen to specify the amount of regularity desired by the user, has the potential to yield nicer analytical properties at the cost of not admitting all developable or piecewise developable surfaces as absolute minimizers.

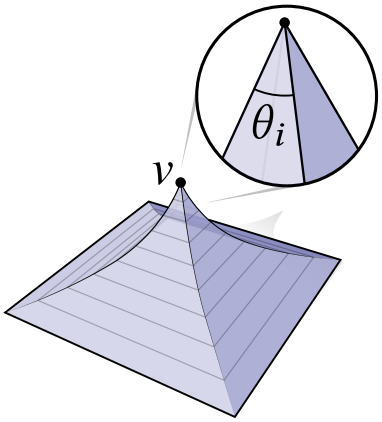
The discretization used in Chapter 7 contains its own implicit regularization caused by numerical smoothing.

4.3 Piecewise Developability

An important feature of E_{dev} is that it is not only minimized by developable surfaces, but also by many *piecewise developable* surfaces. Consider the situation depicted in Figure 4.5: κ_1^2 is zero

inside the two patches left and right of γ, and nonzero but finite on the curve γ. Since γ has zero area, the total energy E_{dev} of the surface is still zero, so the surface still is a general minimizer.

Definition 4.4 (Piecewise developable)**.** *Let Γ be a collection of smooth curves with zero area, finite length, and finite curvature in the continuous surface $\Omega \subseteq \mathbb{R}^3$. Let C be a collection of finitely many points in Ω. Ω is* piecewise developable *if it is twice differentiable and has zero Gaussian curvature in the surface $\Omega \backslash (\Gamma \cap C)$. The curves in Γ are its* seams*, and the points in C are the* cone points*.*



To be exact, the energy E_{dev} is not really well-defined on piecewise developable surfaces such as the one depicted in Figure 4.5, since the surface is not differentiable at seams or at cone points. This can be circumvented by looking at surfaces that are slightly smoothed at the seams, and then taking the limit of infinitesimal smoothness. Near seams, the energy contributes a finite amount over a vanishing area, so their contribution goes to zero. The situation is more complicated near cone points (see inset). At a cone point v, both principal curvatures can approach ∞. However, it is known that the integrated Gaussian curvature κ at such cone points is finite: it is exactly equal to 2π minus the sum of all angles θ_i around the cone point [8]. One can conclude from this that these cone points will contribute a nonzero, but finite amount to the energy E_{dev} of piecewise developable surfaces.

Seam fairing An interesting property of the energy E_{dev} is that it penalizes the curvature of seams on piecewise developable surfaces. Consider a parametrized seam curve $\gamma(s)$ on a piecewise developable, but slightly smoothed surface Ω. The contribution of γ to E_{dev} is $\varepsilon \int_\gamma \kappa_\gamma^2 \, \mathrm{d}x$, where κ_γ

is the scalar curvature of γ, and $\epsilon > 0$ measures the slight smoothing that goes away in the limit. Thus, the energy will try to smooth out seams and make them less curved as it is minimized. Although the effect vanishes as $\epsilon \to 0$, this still has the effect of smoothing out seams during minimization.

4.4 Literature

This section very briefly discusses literature relating to the smooth developability energy E_{dev} and its gradient flow. The many works influencing the discretization, as well as other variants of discrete developability and applications of developability, are discussed in Chapter 7.

The energy flow discussed in this chapter is similar to other classic geometric energies with flows such as Gauss curvature flow [80, 81, 82] or Hamilton's Ricci flow [83]. Unlike these flows, which tend to smooth out the geometry until a global minimum is attained, the developable flow can result in only piecewise smooth surfaces, all of which are global minimizers. This behavior makes the flow quite appealing for applications where one seeks piecewise developable surfaces, rather than a single smooth patch.

Beyond developability, geometric flows and energies are used for a variety of applications such as surface smoothing [18, 49], surface processing [84], and mechanical simulation [56, 85].

Chapter 5: Discretizing E_Δ with zero Neumann Boundary Conditions

There are many ways to discretize the Laplacian energy E_Δ (see Definition 2.9) with zero Neumann and zero Dirichlet boundary conditions whose minimizers u solve $\Delta^2 u = f$. This chapter quickly introduces a variety of finite elements and the *mixed finite element method*, which is a very popular discretization for E_Δ with zero Neumann and zero Dirichlet boundary conditions; a method that has existed for a long time, but which has not been proven to converge for curved surfaces (Section 5.1). Then we prove the convergence of the mixed finite element method with piecewise linear elements for triangle meshes approximating curved surfaces under certain regularity conditions (Section 5.2). This is a novel and hitherto unknown proof. The order of convergence in the L^2 norm is h, which matches the convergence rates observed in practice, as well as the convergence rates known from the flat case.

5.1 Preliminaries: Finite and Mixed Finite Elements

This section offers a quick overview of the finite element method, presents a few important finite elements, and introduces the mixed finite element method that is used to solve the biharmonic

equation with zero Neumann boundary conditions. This is previous work. A comprehensive overview of finite elements can be found, for example, in the book of Braess [86].

5.1.1 The Galerkin method

In order to solve the Poisson equation $\Delta u = f$ in its weak formulation from Definition 2.3, one has to find $u \in H^1$ or H^1_0 such that

$$\int_\Omega \nabla u \cdot \nabla v \, \mathrm{d}x = \int_\Omega f v \, \mathrm{d}x \qquad \forall v \in H^1 \text{ or } H^1_0 \, ,$$

where the choice of H^1 or H^1_0 depends on whether the equation is solved with Dirichlet of Neumann boundary conditions. Without loss of generality, from this point forward, zero Neumann boundary conditions for the Poisson equation will be assumed.

The space H^1 is not discrete, and it is thus challenging to exactly solve over all its functions on a computer. Instead, one solves the equation for a *discrete subspace* $V \subseteq H^1_0$ (i.e., V has a finite basis): find $u \in V$ such that

$$\int_\Omega \nabla u \cdot \nabla v \, \mathrm{d}x = \int_\Omega f v \, \mathrm{d}x \qquad \forall v \in V \, . \tag{5.1}$$

In this discrete space, the equation is hopefully easy to solve on a computer. If the subspace fulfills certain approximation properties, such that there is a series of subspaces $V_1, V_2, \cdots$ with corresponding solutions $u_1, u_2, \cdots$ that converge to the exact solution u in H^1 in some way, the

discrete subspaces can be used to solve the non-discrete problem in H^1 in the limit. This approach is called the *Galerkin method.*[1]

Let the basis of V be $\varphi_1, \dots, \varphi_n$. Then the solution u to (5.1) can be written as $u = \sum_{i=1}^{n} u_i \varphi_i$, and (5.1) can be written as

$$\sum_i u_i \int_\Omega \nabla\varphi_i \cdot \nabla\varphi_j \, \mathrm{d}x = \int_\Omega f\varphi_j \, \mathrm{d}x \qquad \forall j = 1, \dots, n \, .$$

This can be further simplified into the matrix problem

$$\begin{aligned} \mathbf{L}\mathtt{u} &= \mathtt{f} \\ \text{matrix } \mathbf{L}_{ij} = \int_\Omega \nabla\varphi_i \cdot \nabla\varphi_j \, \mathrm{d}x \qquad & \text{vector } \mathtt{f}_i = \int_\Omega f\varphi_j \, \mathrm{d}x \, , \end{aligned} \tag{5.2}$$

where $\mathtt{u} = (u_1, \dots, u_n)$ is the discrete solution vector. The initial problem has now been reduced to the construction of a symmetric matrix and a linear solve.

The discrete Laplacian matrix $\mathbf{L}$ is automatically a discrete Dirichlet energy matrix. Solving (5.2) is equivalent to solving a discrete version of the energy minimization problem from Definition 2.4,

$$\mathtt{u} = \operatorname*{argmin}_{\mathtt{u}} \frac{1}{2} \mathtt{u}^\top \mathbf{L} \mathtt{u} \, , \tag{5.3}$$

and the Dirichlet energy matrix $\mathbf{L}$ can be thus used for applications of the Dirichlet energy.

Whether such a Galerkin method is useful or not depends entirely on the nature of the discrete space V and its basis $\varphi_1, \dots, \varphi_n$. There are many desirable qualities that this space must fulfill in order to be actually useful for the solution of the problem on a computer. The discrete solution has

[1] There is much more to the Galerkin method, such as specific requirements on the equation being solved, requirements on the subspaces, error estimates, and more [86].

to approach the exact solution in H^1 in a certain order of convergence that is proportional to the amount of computational power invested (the more of an improvement additional computational power yields, the better). The matrix $\mathbf{L}$ needs to be easy to assemble, and the resulting linear system must be easy to solve (for example, because $\mathbf{L}$ has a certain sparsity pattern). The discrete Laplacian $\mathbf{L}$ and solution u must fulfill certain desirable properties of the exact solution (for the Laplacian, these are locality, linearity, the maximum principle, and more [87]). Finally, the degrees of freedom u_i of the discrete solution u must be easily translatable into meaningful properties of the exact solution in the space H^1.

5.1.2 The finite element method (FEM)

One way to find a suitable discrete space V for the Galerkin method is the *finite element method*, where the domain Ω is decomposed into a collection of geometric primitives, and degrees of freedom are associated with these primitives. For curved surfaces, the decomposition of the surface is not a direct subset of the surface: the surface is curved arbitrarily, and a discrete approximation will only be piecewise planar, or quadratic, or others. In this thesis, only one decomposition is ever used: the triangle mesh.

Definition 5.1 (Triangle mesh)**.** *[8]*

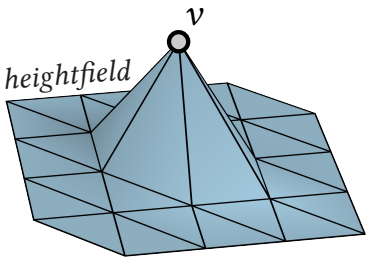
A triangle mesh is a collection of flat nondegenerate triangles. It has vertices (where the triangle vertices meet, or at isolated triangle vertices), edges (where the triangle edges meet, or at isolated triangle edges) and faces (the individual triangles).

A triangle mesh is nice if it is a continuous manifold at every point, and it does not self-intersect. If it is not stated, it is in general assumed that all triangle meshes are nice.

A triangle mesh approximates a surface if, casually speaking, its vertices and faces are somewhat close to the real surface. In more precise mathematical terms, a sequence of triangle meshes approximates a surface Ω if the mesh's vertices and face normals approximate the vertices and normals of the exact surface. The exact degree of approximation needed depends on the problem that is solved, and which element it is solved with.

Lagrangian Finite Elements A very popular finite element for triangle meshes is the linear Lagrangian finite element. For a triangle mesh with vertices $v_i, i = 1, \ldots, n$, there are n linear Lagraigian basis functions $\varphi_i, i = 1, \ldots, n$. The basis function φ_i is associated with the vertex v_i:

- φ_i is 1 on the vertex v_i, and 0 on all other vertices $v_j, j \neq i$;
- φ_i is linear on all triangle faces, and continuous on the entire mesh.



The definition implies that every basis function φ_i is only supported in the faces adjacent to the vertex v_i, and it is zero in all other faces. This is an important property for finite elements: basis functions with a small support guarantee a sparse discrete operator that is amenable to fast solution of linear systems. The resulting basis function φ_i looks like a little hat centered at the vertex v_i (see inset). Because of that, they are sometimes called *hat functions*.

Not only are the basis functions very easy, the resulting operators are also simple and easy to program on a computer. The discrete Laplacian matrix $\mathbf{L}_{ij} = \int_\Omega \nabla\varphi_i \cdot \nabla\varphi_j \, \mathrm{d}x$ is given by [88]

$$\begin{aligned} \mathbf{L}_{ij} &= \begin{cases} -\frac{1}{2}\left(\cot\alpha_{ij} + \cot\beta_{ij}\right) & i,j \text{ neighbors}, i \neq j \\ 0 & i,j \text{ not neighbors}, i \neq j \end{cases} \\ \mathbf{L}_{ii} &= -\sum_{i\neq j} \mathbf{L}_{ij} \ , \end{aligned} \tag{5.4}$$

where α_{ij} and $\beta_i j$ are the two angles opposite the edge that goes from i to j. Because the definition of the Laplacian features cotangents, this discrete Laplacian is sometimes known as the *cotangent Laplacian*. The discrete mass matrix $\mathbf{M}_{ij} = \int_\Omega \varphi_i\varphi_j \, \mathrm{d}x$, which can be used to, for example, discretize a right-hand side f given as a function value per vertex, is given by [88]

$$\sum_{i,j \text{ are neighbors}} \mathbf{M}_{ij} = \frac{1}{3}A_i \qquad A_i \text{ sum of all triangle areas incident to vertex } i \ . \tag{5.5}$$

In practice, the mass matrix $\mathbf{M}$ is often lumped (it is replaced by a diagonal matrix where the diagonal entry corresponds to the sum of the values of each row), as this makes computing and inverting it much easier, without affecting convergence. The lumped matrix is given by $\mathbf{M}_{i,i} = \frac{1}{3}A_i$.

This discretization is very old: it goes back at least as far as the work of MacNeal [89]. It has been popular in geometry processing and computer graphics ever since the works of Pinkall and Polthier [16] and Desbrun *et al.* [18]. It is known to discretize the Dirichlet energy E_∇ in a convergent way, given certain regularity conditions on the mesh Ω_h.

Theorem 5.1 (Convergence of Lagrangian finite elements)**.** *Consider a mesh Ω_h with maximum edge length h such that the ratio of circumcircle to incircle is bounded, independent of the choice of*

h. Piecewise linear Lagrangian finite elements converge for the Poisson equation at the order of h^2 in the L^2 norm and h in the L^2 norm of the gradient of the solution if

(i) the vertices of the mesh Ω_h are inscribed in the surface Ω; or

(ii) the mesh Ω_h converges pointwise to the surface Ω on the order of h, and the normals of Ω_h converge pointwise to the normals of Ω on the order of h.

Proof. For (i), see the work of Dziuk [90]. For (ii), see the work of Wardetzky [91]. □

The approach of (5.4) discretizes the Dirichlet energy with zero Neumann boundary conditions. The zero Neumann boundary conditions are not explicitly enforced: minimizers of L still have some nonzero normal gradient at the boundary. They do, however, converge to a solution with zero Neumann boundary condition. This is caused by the boundary penalty term for the Poisson equation (2.2), which is implicitly applied to everything when just the energy with no additional boundary conditions is minimized. If the values of vertices at the boundary are *explicitly* fixed during the optimization, arbitrary Dirichlet boundary conditions can be enforced, which replace the zero Neumann boundary conditions.

The convergence of linear Lagrangian finite elements for the Poisson equation on curved surfaces was first shown by Dziuk [90]. He generalizes the standard result for solving the Poisson equation with linear finite elements from $\mathbb{R}^2$ to smooth surfaces by working with *inscribed* meshes, i.e., requiring that vertices of the approximating mesh be contained in the limit surface. His work has since been used in advanced numerical methods for curved surfaces [92, 93, 94]. Wardetzky [91] and Hildebrandt *et al.* [95] generalize Dziuk's result to non-inscribed meshes and provide exact estimates of certain geometric quantities.

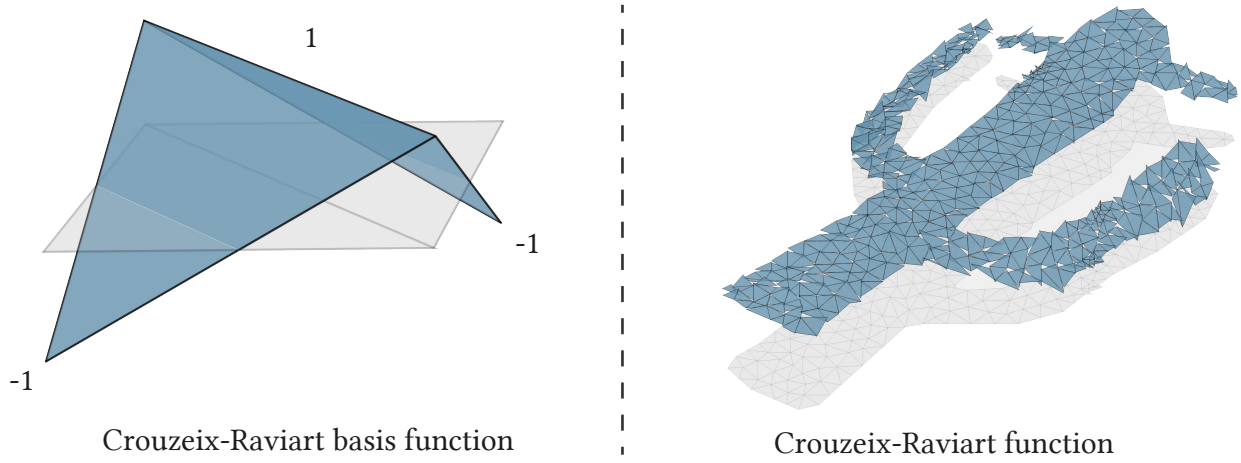


Figure 5.1. A Crouzeix-Raviart basis function is 1 at its associated edge, −1 at the opposite vertices, and linear in between (*left*). Crouzeix-Raviart basis functions are discontinuous, except at edge midpoints (*right*).

Crouzeix-Raviart Finite Elements The Lagrangian finite element basis functions, the hat functions, are all contained in the space H^1 (as they are continuous and piecewise smooth). This makes them *conforming* finite elements. A different approach involves so-called *non-conforming* finite elements which are not contained in the solution space of the Laplace equation, H^1. A simple example of such a non-conforming element is the Crouzeix-Raviart finite element, first introduced by Crouzeix and Raviart [96].

The degrees of freedom of the Crouzeix-Raviart finite element correspond to mesh edges. There is one discontinuous Crouzeix-Raviart basis function φ_e per edge e of the triangle mesh. It is

- 0 outside the two triangles adjacent to e;
- constant 1 at the edge e, and -1 at the vertices opposite e;
- 0 at the edge midpoints not on e;
- linear within the two triangles adjacent to e.

Figure 5.1 (*left*) shows a Crouzeix-Raviart basis function. Linear combinations of these basis functions are, by definition, continuous only at the midpoint of each edge, and discontinuous otherwise. Thus, Crouzeix-Raviart functions, as shown in Figure 5.1 (*right*) are not members of H^1. This makes them non-conforming.

As with the linear Lagrangian finite element, the discrete Laplacian and mass matrix are evaluated by computing the respective integrals for pairwise basis functions. There is one subtlety: one only applies the gradient to each basis function *within* each triangle, and ignores the jumps over triangle edges caused by discontinuity. The resulting discrete Laplacian matrix $\mathbf{L}_{ij} = \int_\Omega \nabla\varphi_i \cdot \nabla\varphi_j \, \mathrm{d}x$ is given by [88]

$$\begin{aligned} \mathbf{L}_{ij} &= \begin{cases} -\cot\theta_{ij} & i,j \text{ share a vertex, } i \neq j \\ 0 & i,j \text{ do not share a vertex, } i \neq j \end{cases} \\ \mathbf{L}_{ii} &= -\sum_{i \neq j} L_{ij} \ , \end{aligned} \tag{5.6}$$

where θ_{ij} denotes the angle between the two edges i, j. The discrete mass matrix $\mathbf{M}_{ij} = \int_\Omega \varphi_i \varphi_j \, \mathrm{d}x$ is given by [88]

$$\mathbf{M}_{ii} = \frac{1}{3} A_i \qquad A_i \text{ area of the two triangles adjacent to } e \ . \tag{5.7}$$

This mass matrix is diagonal by construction. It does not need to be lumped like the scalar Lagrangian finite element mass matrix.

As with the hat functions, the approach of (5.6) discretizes the Dirichlet energy with zero Neumann boundary conditions. If the values of vertices at the boundary are fixed during the optimiza-

tion, arbitrary Dirichlet boundary conditions can be enforced, which replace the zero Neumann boundary conditions.

Since Crouzeix-Raviart functions are discontinuous, one cannot expect their gradient to converge to the gradient of the exact solution. The function value, however, converges, for flat surfaces $\subseteq \mathbb{R}^2$, just like the linear Lagrangian finite element. There is no such proof for curved surfaces.

Theorem 5.2 (Convergence of Crouzeix-Raviart finite elements)**.** *Consider a flat mesh $\Omega_h \subseteq \mathbb{R}^2$ with maximum edge length h such that the ratio of circumcircle to incircle is bounded, independent of the choice of h. Crouzeix-Raviart finite elements converge for the Poisson equation at the order of h^2 in the L^2 norm.*

Proof. See Braess [86, p. III.1.5]. □

The fact that the degrees of freedom of Crouzeix-Raviart finite elements are on edges (instead of the vertices of the linear Lagrangian finite element), as well as their simplicity, make them popular in a variety of computer graphics applications such as in simulation [56, 88, 97] and geometry processing [98, 99].

5.1.3 The mixed finite element method (mixed FEM)

The finite element methods discussed so far are suitable for discretizing the Dirichlet energy (which solves the Poisson equation) and other first-order energies: first-order, because their formulation only involves one derivative. To discretize the Laplacian energy with zero Dirichlet and

zero Neumann boundary conditions, one must be able to take two derivatives, as the energy from Definition 2.9 involves the Laplacian of a function.

A way around this is the *mixed finite element method* of Ciarlet and Raviart [100]. The minimization of the Laplacian energy $E_\Delta(u) = \frac{1}{2}\int_\Omega |\Delta u|^2 \, \mathrm{d}x$ is split into two problems, with two unknown functions as the solution, u and v [41]:

$$\operatorname*{argmin}_{u,v} \frac{1}{2}\int_\Omega |v|^2 \, \mathrm{d}x \qquad \text{s.t. } \Delta u = v \; . \tag{5.8}$$

The new problem now only features a mass matrix, $\frac{1}{2}\int_\Omega |v|^2 \, \mathrm{d}x$, and a Poisson equation, $\Delta u = v$, both of which we already know how to discretize with piecewise linear Lagrangian finite elements.

(5.8) is now minimized using Lagrange multipliers, which turns an optimization problem into a saddle problem,

$$\operatorname*{saddle}_{u,v,\lambda} \frac{1}{2}\int_\Omega |v|^2 \, \mathrm{d}x + \int_\Omega \lambda\left(\Delta u - v\right) \, \mathrm{d}x = \operatorname*{saddle}_{u,v} \frac{1}{2}\int_\Omega |v|^2 \, \mathrm{d}x + \int_\Omega \nabla\lambda \cdot \nabla u - \lambda v \, \mathrm{d}x \; , \tag{5.9}$$

where λ is an auxiliary smooth function introduced to solve the optimization problem. The discrete version of this problem, with $\mathbf{L}$ and $\mathbf{M}$ the Laplacian and mass matrices for hat functions, is: find discrete solutions $\mathtt{u}, \mathtt{v}, \lambda$ that solve the saddle problem

$$\operatorname*{saddle}_{\mathtt{u},\mathtt{v},\lambda} \frac{1}{2}\mathtt{v}^\top \mathbf{M}\mathtt{v} + \lambda^\top\left(\mathbf{L}\mathtt{u} - \mathbf{M}\mathtt{v}\right) \; .$$

At the saddle point, it must be true that $\mathbf{L}\mathtt{u} = \mathbf{M}\mathtt{v}$. Thus, the whole saddle problem reduces to

$$\operatorname*{argmin}_{\mathtt{u}} \frac{1}{2}\mathtt{u}^\top \mathbf{L}\mathbf{M}^{-1}\mathbf{L}\mathtt{u} \,. \tag{5.10}$$

If M is lumped, then inverting it is very simple. Using the mixed finite element method, one can take the simple hat function discretization of the Dirichlet energy and use it to discretize the Laplacian energy with very little additional work.

The energy formulated in (5.10), when optimized over all hat functions, will produce solutions that implicitly fulfill the zero Neumann boundary conditions, just like the minimizers of L do on their own (this is sometimes referred to as the zero Neumann boundary conditions being *baked in*). If, additionally, Dirichlet boundary conditions are enforced, both Dirichlet and Neumann boundary conditions hold at the same time, and the discretization can be used to solve the biharmonic equation from Theorem 2.4.

About the history of mixed finite elements for the biharmonic equation. Ciarlet and Raviart [100] are the first to introduce the mixed finite element method for the biharmonic problem. Their work informs the functional analysis framework that we use for the proof in Section 5.2. They solve the same system of equations that we end up solving (in the flat case), but only show convergence for higher-order Lagrangian finite elements. Their approach is later expanded by Monk [101] to deal with boundary smoothness problems caused by triangulating.

Scholz [102] proves that the mixed finite element method with linear, first-order Lagrange elements can be used to solve the biharmonic problem, and he gives an error bound of $h \log^2 h$ on the L^2 norm of the solution. The result by Scholz is central to understanding the convergence of

the linear finite element method for the biharmonic equation, and forms the basis of the proof in Section 5.2. The result is remarkable, since it shows convergence of the method, even though the standard convergence conditions for mixed finite elements (the *inf-sup* conditions [103]) are not fulfilled. Scholz's error estimate is not optimal, as it relies on an L^∞ estimate of the Ritz approximation error by Nitsche [104]. An application of a later, better bound for the same interpolation error [105] gives convergence of order h.

The concept of mixing different function spaces to solve numerical analysis problems is an even older idea. Using hybrid and mixed finite elements to solve the biharmonic equation with Dirichlet and Neumann boundary conditions in $\mathbb{R}^2$ goes back to the works of Herrmann [106], Brezzi [107], and Miyoshi [108] as well as others, who study a saddle point problem formulation based on the Hessian energy $J(u) \coloneqq \frac{1}{2}\int_\Omega u_{xx}^2 + 2u_{xy}^2 + u_{yy}^2\, dx$. Their method is a so-called *hybrid* method, where, in the derivation of the method, "the variational principle is explicitly dependent upon a given triangulation of the [domain]" [100, pp. 125]. A good summary of hybrid and mixed finite elements and their history can be found in the work of Boffi *et al.* [103].

Oukit and Pierre [109] provide an analysis of the biharmonic equation with Dirichlet and Neumann boundary conditions that combines the hybrid approach (which they call Hermann-Miyoshi) and the mixed approach (which they call Ciarlet-Raviart). Their analysis holds for first and second order elements. The result by Scholz [102] is recovered in the limit $p \longrightarrow \infty$, where p is the degree of the L^p space.

An alternative approach to solving the biharmonic equation using mixed finite elements is the decomposition into four linear equations, such as done by Behrens and Guzmán [110] (which leads to superconvergence of the solution) and Li and Zhang [111].

Meanwhile, other methods for solving the biharmonic equation on surfaces exist in the literature; a few examples follow. Larsson and Larson [112] use the discontinuous Galerkin approach to achieve a method for surfaces without boundary where the L^2 error is of order h. Unlike this approach, we use a mixed formulation with a conforming approximation, and, in out proof in Section 5.2, additionally provide error estimates of the Laplacian, as well as a treatment of surfaces with boundaries. Cockburn and Demlow [113] use a different kind of discontinuous Galerkin method as well as non-conforming mixed finite elements (Raviart-Thomas, Brezzi-Douglas-Marini, Brezzi-Douglas-Fortin-Marini). In contrast to their method, the mixed finite element method presented here uses a *conforming* linear element, and incorporates Dirichlet and Neumann boundary conditions.

Although a convergence proof has been missing so far, discretizations of the biharmonic equation (and its related Helmholtz problem) using a mixed formulation with linear Lagrange elements have been popular in practice in computer graphics and geometry processing. Many of the applications mentioned in Section 2.3.2 (and others) use these mixed finite elements [18, 37, 56, 114, 115, 41, 35, 34]. There is currently no convergence proof that justifies their use.

5.2 A Convergence Proof for the Mixed Finite Element Discretization of E_Δ

This section is a change of pace compared to the other sections. The goal is to prove the convergence of the piecewise linear Lagrangian mixed finite element discretization of the Laplacian energy with zero Dirichlet and zero Neumann boundary conditions on curved surfaces approximated by triangle meshes. Because this section contains a rigorous mathematical proof, we are

especially careful to use very precise mathematical definitions, and we will thoroughly prove every claim, in order to arrive at a watertight theorem at the end. Some definitions from earlier will be repeated, but defined in a more precise way needed for this proof. As a result, this section starts by introducing a variety of new notation for many mathematical concepts, and repeats some of the functional analytic theory of Ciarlet [116]. However, the mixed finite element in this section is more or less the same method as was described, with less mathematical stringency, in Section 5.1.3. In addition, unlike the rest of this thesis, this section assumes that the reader is familiar with various advanced concepts of functional analysis, geometry, and numerical mathematics, beyond what was introduced in Chapter 1, such as is described in the works of Evans [12], Gilbarg and Trudinger [13], and Lax [14].

5.2.1 Problem statement

Consider the *biharmonic equation* on smooth surfaces embedded in three-dimensional Euclidean space: given a function f on a smooth surface, find a function u such that

$$\Delta^2 u = f \ , \tag{5.11}$$

where Δ is the Laplace–Beltrami operator. If boundaries are present, boundary conditions must be taken into account. Consider Dirichlet and Neumann boundary conditions,

$$u = \frac{\partial u}{\partial \mathbf{n}} = 0 \qquad \text{at the boundary.} \tag{5.12}$$

For flat surfaces, this problem is sometimes referred to as the *clamped thin plate problem*. If no boundaries are present, the problem is solved modulo constant functions.

In order to solve the biharmonic problem, we use a *mixed finite element method* with *piecewise linear elements* on polyhedral *triangle meshes*. Every mesh is required to have *uniformly shape regular triangles*. Moreover, we consider sequences of triangle meshes that converge to a given smooth limit surface such that both *positions* and *normals* converge at a certain rate (to be specified later). We show:

- L^2-convergence of order h of discrete solutions u;
- L^2-convergence of order $h^{\frac{3}{4}}$ of discrete solution gradients ∇u;
- L^2-convergence of order $\sqrt{h}$ of the discretization of Δu;

where h is the maximum edge length of the approximating triangles of the mesh. If no boundaries are present, the problem becomes simpler and better convergence rates can be obtained.

The mixed method consists of solving a linear system of two equations; in u_1 (which corresponds to the solution u) and in u_2 (which corresponds to the Laplacian Δu of the solution),

$$\begin{aligned} \int_\Omega \nabla u_2 \cdot \nabla \xi \, \mathrm{d}x &= \int_\Omega f \xi \, \mathrm{d}x \qquad \forall \xi \in H_0^1 \\ \int_\Omega \nabla u_1 \cdot \nabla \eta \, \mathrm{d}x &= \int_\Omega u_2 \eta \, \mathrm{d}x \qquad \forall \eta \in H^1 \, . \end{aligned} \tag{5.13}$$

Here we assume that $f \in L^2$, and we seek solutions $u_1 \in H_0^1$ (or a suitable subspace) and $u_2 \in H^1$ (or a suitable subspace). We employ the geometric convention of working with *positive semidefinite* Laplacians.

A convergence result for the mixed finite element approximation to the biharmonic problem (using piecewise linear Lagrange elements) has long been known for the case of *planar* domains [102]. Our convergence rates can be obtained by combining the results of Scholz [102] with the results of Rannacher and Scott [105]. In a nutshell, the idea is to bound the error between the discrete solution u_2 and the Ritz approximation of the exact solution, and then use the fact that the error of the Ritz approximation can be explicitly controlled. Although this approach might seem easily extensible to the case of curved surfaces, the non-flat geometry brings about additional challenges.

There are multiple conceivable approaches for generalizing Scholz's proof to the curved setting. A first approach might be an attempt to directly bound the error between discrete solutions to (5.13) on polyhedral meshes and the smooth solution on a given limit surface. For this approach, however, a considerable complication arises from the fact that the geometry of the mesh and the geometry of the limit surface are different. A different approach that works for the Poisson problem [91] can also not be used here: It is not possible to first compute a "smooth" solution[2] $u_1 \in H^1_0, u_2 \in H^1$ to (5.13) on the polyhedral mesh and then compare it to the smooth solution on the limit surface, since such smooth solutions on the polyhedral mesh do not necessarily exist. Indeed, even for planar nonconvex Lipschitz domains, problem (5.13) ceases to have a solution [117, Section 4.3].

Perhaps surprisingly, *discrete solutions* to (5.13) using piecewise linear Lagrange elements *do* indeed exist for meshes. Our approach is to relate these discrete solutions on meshes to certain discrete solutions on the smooth limit surface by leveraging an inverse estimate and (inverse)

[2]Geometrically, polyhedral meshes are Euclidean cone manifolds that are intrinsically flat everywhere, except at isolated *cone points*. Even in the presence of cone points, it is possible to define L^2 and H^1 spaces [91]. It is therefore possible to formulate an analogous "smooth" problem (5.13) on Euclidean cone manifolds. This "smooth" problem, however, does not always have a solution.

discrete Laplacians. This allows us to circumvent the above-mentioned problem of nonexistence of a "smooth" solution on polyhedral meshes. Our generalization of Scholz's proof, which relies on a novel bound for the L^∞ error of the linear FEM for the Poisson equation on curved surfaces, can then be used to bound the error between the discrete solution on the smooth limit surface and the exact solution. Together, these two steps result in an error bound for the mixed finite element approximation of the biharmonic equation.

5.2.2 Definitions and notation

5.2.2.1 The biharmonic and weak biharmonic equations

The definitions in this section do not describe novel concepts, but prior work that has been featured in textbooks [12, 13]. We repeat certain definitions in this section in order to prevent ambiguity, and will not reference a textbook for each definition individually.

On the surface Ω with metric g, we define the short-hand $\langle \cdot, \cdot \rangle$ for the inner product on L^2 and $D\langle \cdot, \cdot \rangle$ for the inner product on H^1_0,

$$\begin{aligned} \langle u, v \rangle &= \int_\Omega uv \, \mathrm{d}x & u, v \in L^2 \\ D\langle u, v \rangle &= \int_\Omega g\left(\nabla u, \nabla v\right) \, \mathrm{d}x & u, v \in H^1 \, . \end{aligned}$$

The norm on H^1 is induced by the inner product $\langle \cdot, \cdot \rangle + D\langle \cdot, \cdot \rangle$.

Definition 5.2. *For $f \in L^2$, the* biharmonic equation *is defined as follows: find $u \in H^1_0 \cap H^4$, such that*

$$\Delta^2 u = f \, , \tag{5.14}$$

where Δ is the positive semidefinite Laplace–Beltrami operator.

- *If Ω has a boundary, then zero Dirichlet and Neumann boundary conditions apply, $u = \frac{\partial u}{\partial \mathbf{n}} = 0$;*
- *if Ω is closed, u must have zero mean, i.e., $\int_\Omega u \, \mathrm{d}x = 0$.*

The biharmonic equation has a corresponding weak formulation. For $f \in L^2$, find $u \in H^1_0 \cap H^2$ such that

$$\int_\Omega \Delta u \Delta v \, \mathrm{d}x = \int_\Omega f v \, \mathrm{d}x \qquad \forall v \in H^1_0 \cap H^2 \, . \tag{5.15}$$

We assume that there is a unique solution such that $u \in H^4$, $\|u\|_{H^4} \leq C\|f\|_{L^2}$. For closed surfaces this follows from the fact that the biharmonic equation decouples into two Poisson equations. For planar domains, see Gazzola *et al.* [33, Section 2.5.2]. Additionally, we assume the standard existence and regularity estimates for the Poisson equation: for $g \in L^p$, $1 < p < \infty$ there is a unique $w \in W^{2,p}$ with Dirichlet boundary conditions such that, weakly, $\Delta w = g$ and $\|w\|_{W^{2,p}} \leq C\|g\|_{L^p}$. For planar domains, see, for example, the work of Grisvard [118, Section 2].

With $u_1 \coloneqq u$, and using the intermediate variable $u_2 \coloneqq \Delta u_1$, (5.15) can be rewritten in its mixed form

$$\begin{aligned} D\langle u_2, \xi \rangle &= \langle f, \xi \rangle \qquad \forall \xi \in H^1_0 \\ D\langle u_1, \eta \rangle &= \langle u_2, \eta \rangle \qquad \forall \eta \in H^1 \, . \end{aligned} \tag{5.16}$$

We refer to this system of equations as the smooth mixed formulation of the biharmonic equation with Dirichlet and Neumann boundary conditions. It can be formulated for any $u_1 \in H^1_0$ and $u_2 \in H^1$. By Ciarlet [116, Theorem 7.1.1], (5.16) has a unique solution such that $u_1 \in H^1_0 \cap H^2$, so by our assumptions it has a unique solution such that $u_1 \in H^1_0 \cap H^4$ and $u_2 \in H^2$.

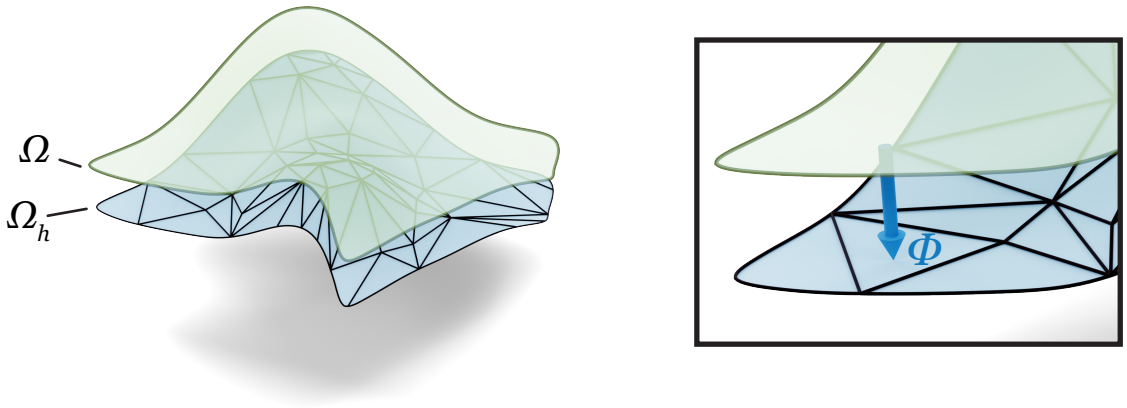


Figure 5.2. A surface Ω, in green, and an approximating mesh Ω_h, in blue (*left*). A close-up of the surface and the mesh with the shortest distance map Φ between them.

5.2.2.2 Meshes approximating a surface

We now mathematically define the domain on which we formulate our discrete method: the triangle mesh, a continuous, piecewise flat, triangular surface. Boundary edges of triangles along the surface boundary are allowed to be curved as long as the curve remains in the plane of the triangle. In the flat case, triangle meshes are only needed to discretize the function space H_0^1 in which the solution lives. In the case of a surface $\Omega \subseteq \mathbb{R}^3$, however, the mesh is also used to discretize the geometry itself. For this reason, we require additional conditions of our mesh that are not present in the work of Scholz [102]. These conditions mirror the ones of Wardetzky [91], whom this section closely follows, and where many definitions (which are not novel) are taken from.

Definition 5.3. *Let X be a topologically closed subset of $\mathbb{R}^3$. The medial axis of X is the set of those points in $\mathbb{R}^3$ that do not have a unique closest neighbor in X. The* reach *of X is the distance of X to its medial axis.*

Let Ω_h be a triangle mesh. Then we can define the following map:

Definition 5.4. *Let Ω_h lie within the reach of Ω. The* shortest distance map *is the map $\Phi : \Omega \to \Omega_h$ that takes every point $p \in \Omega$ to the closest intersection of the line through p parallel to the normal of Ω at p, and Ω_h, see Figure 5.2.*

Throughout, we require the following conditions of our mesh:[3]

(C1) The triangles of Ω_h are uniformly shape regular, i.e., there exist constants $\kappa, K > 0$ such that every triangle contains a circle of radius κh and is contained in a circle of radius Kh.

(C2) The polyhedral surface Ω_h is a *normal graph* over the smooth surface Ω, i.e., Ω_h lies within the reach of Ω and the shortest distance map is a bijective function. In particular, the boundary of Ω is bijectively mapped to the boundary of Ω_h.

(C3) The distance of every point under the shortest distance map is bounded by Ch^γ for some $\gamma \geq \frac{3}{2}$.

(C4) The normals of Ω_h converge to the normals of Ω such that at almost every point $p \in \Omega$ the angle between the surface normal and the normal of Ω_h at $\Phi(p)$ is bounded by Ch^ε for some $\varepsilon \geq 1$.

(C5) $\sigma := \min(\gamma, 2\varepsilon) \geq \frac{3}{2}$.

Condition **(C2)** might seem difficult to satisfy for meshes that have nonempty boundary, since we require that the boundary of Ω maps exactly to the boundary of Ω_h under Φ. However, this condition is not hard to satisfy. Consider a mesh Ω_h within the reach of Ω, where boundary

[3] We adopt the convention that, wherever a constant C occurs, the words "there is a constant $C > 0$, dependent only on the surface Ω and mesh regularity parameters" are implied.

vertices are inscribed into the smooth boundary such that every triangle has at most two vertices on the boundary of the mesh, and these vertices must be connected by a boundary edge of the mesh. Now, every boundary triangle lies in a unique plane. The image of the map Φ onto the triangle's plane is a collection of flat triangles with at most one curved edge. Thus the boundary triangles of Ω_h can be extended such that **(C2)** is satisfied. This makes condition **(C2)** very similar to the condition of Scholz [102] in the flat case, which requires triangles with curved edges that exactly match the boundary of the smooth domain.

Remark 5.1. *Condition **(C2)** is not present in the work of Wardetzky [91]. For this reason, Wardetzky's result on the convergence of the finite element method for the Poisson equation, Theorem 3.3.3, only holds for solutions that are supported sufficiently far away from the boundary. With Condition **(C2)**, and with the finite element spaces that will be defined in Definition 5.7, the results of Wardetzky [91] hold for general solutions $u \in H_0^1 \cap H^2$.*

Remark 5.2. *It might be possible to relax condition **(C2)** to only require triangles with straight edges and vertices that are inscribed into the boundary. However, such a relaxation could lead to lower convergence rates.*

Remark 5.3. *A triangle mesh that is inscribed into the smooth surface Ω and that satisfies **(C1)** and **(C2)**, automatically fulfills **(C3)**, **(C4)**, and **(C5)**.*

Using Conditions **(C1-C5)**, we can relate the metric and the function spaces of the mesh Ω_h to the metric and the function spaces of Ω.

Definition 5.5. *Let g denote the metric tensor on the smooth surface Ω. Consider Ω_h as a Euclidean cone manifold, which is flat almost everywhere except at cone points. Let g_{Ω_h} be the corresponding metric tensor (defined almost everywhere) on Ω_h, and let $g_{\mathbb{R}^3}$ denote the inner product of Euclidean*

space. Then we can define the distorted metric g_A on Ω,

$$g_A(X, Y) \coloneqq g_{\Omega_h}(\mathrm{d}\,\Phi(X), \mathrm{d}\,\Phi(Y)) = g_{\mathbb{R}^3}(\mathrm{d}\,\Phi(X), \mathrm{d}\,\Phi(Y)) \ .$$

We define the unique matrix field A that relates g_A to g almost everywhere,

$$g_A(X, Y) = g(AX, Y) \ . \tag{5.17}$$

We define the distorted inner products that relate the inner products on the mesh to the inner products on the surface,

$$\begin{aligned} \langle u, v\rangle_A &\coloneqq \int_\Omega uv\,|\det A|^{\frac{1}{2}}\ \mathrm{d}x \qquad \textit{for } u, v \in L^2(\Omega) \\ D\langle u, v\rangle_A &\coloneqq \int_\Omega g(A^{-1}\nabla u, \nabla v)\,|\det A|^{\frac{1}{2}}\ \mathrm{d}x \qquad \textit{for } u, v \in H^1(\Omega) \ . \end{aligned} \tag{5.18}$$

We adopt the convention that for every norm, the same norm subscripted with A implies that the norm is taken with respect to the distorted metric. For example, L^2_A is the L^2 norm in the distorted metric.

We can get explicit bounds on the entries of A. From now on, the statement "for small enough h" is implied everywhere.

Lemma 5.1. *It holds that*

$$\begin{aligned} \|A - \mathrm{Id}\|_{L^\infty} &\leq Ch^\sigma \\ \left\| |\det A|^{\frac{1}{2}} - 1 \right\|_{L^\infty} &\leq Ch^\sigma \\ \left\| |\det A|^{\frac{1}{2}} A^{-1} - \mathrm{Id} \right\|_{L^\infty} &\leq Ch^\sigma \ , \end{aligned}$$

*where the L^∞ norm is the essential supremum over the operator norms of the respective matrix fields. The scalar $\sigma > 0$ depends on the convergence of the mesh and is defined in **(C5)**.*

Proof. By Wardetzky [91, Theorem 3.2.1] we know that for any point and orthonormal tangent frame there exists a matrix decomposition $A = PQP$ such that P, Q can be diagonalized (possibly in different bases) as

$$\begin{aligned} P &= \begin{pmatrix} 1 - \phi\kappa_1 & 0 \\ 0 & 1 - \phi\kappa_2 \end{pmatrix} \\ Q &= \begin{pmatrix} \frac{1}{(N \cdot N_h)^2} & 0 \\ 0 & 1 \end{pmatrix} , \end{aligned}$$

where ϕ is the pointwise distance between Ω and Ω_h under the map Φ, N and N_h are the surface normals of Ω and Ω_h respectively, and κ_1, κ_2 are the principal curvatures of the surface.

Let $\angle(N, N_h)$ be the unsigned angle between the two vectors N, N_h in $\mathbb{R}^3$. Using Taylor expansion we obtain, up to higher order terms,

$$|\det A|^{\frac{1}{2}} - 1 \approx -\phi\kappa_1 - \phi\kappa_2 - \frac{1}{2}\angle(N, N_h)^2 \ ,$$

which proves the lemma for $\|A - \mathrm{Id}\|_{L^\infty}$, given that $|\phi| \leq Ch^\gamma$ **(C3)**, $|\angle(N, N_h)| \leq Ch^\varepsilon$ **(C4)**, and $\sigma = \min(\gamma, 2\varepsilon)$ **(C5)**. A similar argument works for the other two expressions. □

Lemma 5.2. *The following equalities hold as equalities of sets:*

$$L^2(\Omega) = L^2_A(\Omega) = L^2(\Omega_h)\,,$$
$$H^1(\Omega) = H^1_A(\Omega) = H^1(\Omega_h)\,,$$
$$H^1_0(\Omega) = H^1_{0,A}(\Omega) = H^1_0(\Omega_h)\,,$$
$$W^{1,\infty}(\Omega) = W^{1,\infty}_A(\Omega) = W^{1,\infty}(\Omega_h)\,,$$

where the bijection Φ is applied where needed in order to map between Ω and Ω_h. The norms of the respective spaces are all equivalent independently of the choice of h (for h small enough).

Proof. Because of the bijectivity of Φ, every function on Ω is also well-defined on Ω_h (and vice-versa).

Under the map Φ, the inner products $\langle\cdot,\cdot\rangle_A$ and $D\langle\cdot,\cdot\rangle_A$ of the spaces ${L^2}_A(\Omega)$, $H^1_A(\Omega)$, $H^1_{0,A}(\Omega)$, and $W^{1,\infty}_A(\Omega)$ are exactly the inner products of the spaces $L^2(\Omega_h)$, $H^1(\Omega_h)$, $H^1_0(\Omega_h)$, and $W^{1,\infty}(\Omega_h)$, respectively. Lemma 5.1 implies that the respective disturbed inner products induce norms that are bounded if and only if the respective undisturbed norms are bounded. Disturbed and undisturbed norms bound each other independently of h, as long as h is small enough. This proves the claim. □

Remark 5.4. *The equivalence of norms independently of h also implies a disturbed Poincaré inequality where the constant is independent of h,*

$$\begin{aligned} \|u\|_{L^2_A} &\leq C\|u\|_{H^1_{0,A}} \qquad \forall u \in H^1_0 \\ \left\|u - u_{\Omega,A}\right\|_{L^2_A} &\leq C\|u\|_{H^1_{0,A}} \qquad \forall u \in H^1 \ , \end{aligned}$$

where $u_{\Omega,A}$ is the average of u with respect to integration in the disturbed metric.

Remark 5.5. *Lemma 5.1 and Lemma 5.2 allow for formulating our discrete method entirely on Ω, even though the actual implementation might compute its operators and results on Ω_h.*

In order to quantify the differences between the disturbed and undisturbed inner products, we introduce the following bilinear forms:

Definition 5.6. *The difference bilinear forms are defined as follows,*

$$\begin{aligned} c(u,v) &\coloneqq \langle u, v\rangle - \langle u, v\rangle_A \\ d(u,v) &\coloneqq D\langle u, v\rangle - D\langle u, v\rangle_A \ . \end{aligned} \tag{5.19}$$

Lemma 5.3. *The difference bilinear forms fulfill the inequalities*

$$\begin{aligned} |c(u,v)| &\leq Ch^\sigma \|u\|_{L^2}\|v\|_{L^2} \qquad \forall u, v \in L^2 \\ |d(u,v)| &\leq Ch^\sigma \|u\|_{H^1_0}\|v\|_{H^1_0} \qquad \forall u, v \in H^1 \ . \end{aligned}$$

Proof. This is a consequence of Lemma 5.1 and the Cauchy-Schwarz inequality. □

5.2.2.3 Discrete function spaces

Having discretized the geometry of Ω, we now turn to the approximation of function spaces. The spaces used in this section are standard in the realm of numerical analysis [86, 116]. The definitions in this section are to prevent ambiguity, and follow the sources mentioned above (but with different names).

Definition 5.7. *Let $S_{h,A}$ be the space of continuous functions on the mesh Ω_h that are linear within each triangle. On boundary triangles (which might have a curved edge), an isoparametric modification is applied [119, 120], which projects a curved edge of a triangle onto a straight edge while keeping the vertices fixed and minimizing distortion.[4] On the resulting triangle, functions are required to be linear.*

The discrete space $S_h \subseteq H^1$ is the pullback of $S_{h,A}$ under the shortest distance map Φ. The space S_h has domain Ω and inherits the natural inner product from $H^1(\Omega)$. When we refer to "piecewise linear functions" on the surface Ω, we refer to the space S_h. Analogously, we define the discrete space $S_{h,0} \coloneqq S_h \cap H^1_0$, which inherits its inner product from $H^1_0(\Omega)$.

Remark 5.6. *The number of degrees of freedom of S_h is the number of mesh vertices. The number of degrees of freedom of $S_{h,0}$ is the number of mesh vertices minus the number of boundary vertices.*

The discrete spaces come with various interpolation operators.

Definition 5.8. *Let $I_h : H^2 \to S_h$ denote the per-vertex interpolation operator,[5] i.e., $(I_h u)(p) = u(p)$.*

[4] This treatment of the boundary, present in the work of Scholz [102], does not appear in the work of Wardetzky [91]. The main results of Wardetzky, however, remain true if an optimal isoparametric element is chosen [121], with minor modifications to the proofs, as long as Conditions **(C1-C5)** hold, see Remark 5.1.

[5] The Sobolev embedding theorem implies that $H^2(\Omega) \subseteq C^0(\Omega)$, which justifies pointwise interpolation.

Moreover, $R_h, R_h^A : H^1 \to S_h$ and $R_{h,0}, R_{h,0}^A : H_0^1 \to S_{h,0}$ are the Ritz interpolation operators,

$$\begin{aligned} D\langle u - R_h u, \eta\rangle &= D\big\langle u - R_h^A u, \eta\big\rangle_A = 0 \qquad \forall \eta \in S_h,\ u \in H^1 \\ \langle u - R_h u, 1\rangle &= \big\langle u - R_h^A u, 1\big\rangle_A = 0 \end{aligned}$$

$$D\Big\langle u - R_{h,0} u, \xi\Big\rangle = D\Big\langle u - R_{h,0}^A u, \xi\Big\rangle_A = 0 \qquad \forall \xi \in S_{h,0},\ u \in H_0^1\ .$$

This is analogous to the flat case of Scholz [102, p. 86].

The interpolation operators satisfy certain interpolation inequalities.

Lemma 5.4. *The Ritz-interpolation is H^1-stable, i.e.,*

$$\begin{aligned} \|R_h u\|_{H^1}, \big\|R_h^A u\big\|_{H^1} &\le C\|u\|_{H^1} \qquad \forall u \in H^1 \\ \big\|R_{h,0} u\big\|_{H_0^1}, \Big\|R_{h,0}^A u\Big\|_{H_0^1} &\le C\|u\|_{H_0^1} \qquad \forall u \in H_0^1\ . \end{aligned}$$

For $u \in H_0^1 \cap H^2$ one has

$$\begin{aligned} \|u - I_h u\|_{H_0^1} &\le Ch\|u\|_{H^2} \\ \big\|u - R_{h,0} u\big\|_{H_0^1} &\le Ch\|u\|_{H^2} \\ \Big\|u - R_{h,0}^A u\Big\|_{H_0^1} &\le Ch\|u\|_{H^2}\ . \end{aligned} \tag{5.20}$$

If, in addition, $u \in W^{2,\infty}$,

$$\|u - I_h u\|_{W^{1,\infty}} \le Ch\|u\|_{W^{2,\infty}}. \tag{5.21}$$

If u is continuous and H^1 or $W^{1,\infty}$, but only H^2 or $W^{2,\infty}$ within each triangle, then

$$\begin{aligned} \|u - I_h\, u\|_{\tilde{H}^1_0} &\leq Ch\|u\|_{\tilde{H}^2} \\ \|u - I_h\, u\|_{\tilde{W}^{1,\infty}} &\leq Ch\|u\|_{\tilde{W}^{2,\infty}}, \end{aligned} \tag{5.22}$$

where the tilde above the norm indicates a per-triangle norm, concatenated over all triangles t in the triangulation: $\|u\|^2_{\tilde{H}^2(\Omega)} = \sum_t \|u\|^2_{H^2(t)}$, $\|u\|_{\tilde{W}^{2,\infty}(\Omega)} = \max_t \|u\|_{W^{2,\infty}(t)}$.

Analogous results hold for functions that are nonzero at the boundary using the appropriate interpolation operators. All estimates with the interpolation inequality also hold per triangle.

Proof. We only treat the case of H^1_0-type interpolations. The case of H^1 follows by using the appropriate norms as well as the condition $\langle u - R_h\, u, 1\rangle = \left\langle u - R^A_h\, u, 1\right\rangle_A = 0$.

It holds that

$$\begin{aligned} \left\|R_{h,0}\, u\right\|^2_{H^1_0} &= D\left\langle u - R_{h,0}\, u, R_{h,0}\, u\right\rangle + D\left\langle R_{h,0}\, u, R_{h,0}\, u\right\rangle = D\left\langle u, R_{h,0}\, u\right\rangle \leq C\|u\|_{H^1_0}\left\|R_{h,0}\, u\right\|_{H^1_0} \\ \left\|R_{h,0}\, u\right\|_{H^1_0} &\leq C\|u\|_{H^1_0}\,. \end{aligned}$$

This proves the stability estimates.

Let T_h be a flat triangle of Ω_h. Both the left and right hand sides of the first line of (5.20) hold by restricting to the subset $\Phi^{-1}(T_h) \subseteq \Omega$, see [91, Thereom 3.3.2]. Although this result was only established for triangles T_h with straight edges, the extension to triangles in Ω_h with curved edges along the surface boundary is straightforward. The first line of (5.20) follows by using that $\Phi : \Omega \to \Omega_h$ is a bijection and by summing over all triangles T_h.

The next line of (5.20) follows by

$$\left\|I_h u - R_{h,0}\, u\right\|_{H_0^1} = \left\|R_{h,0}\ (I_h u - u)\right\|_{H_0^1} \leq C\|I_h u - u\|_{H_0^1}\text{ , and therefore}$$
$$\left\|u - R_{h,0}\, u\right\|_{H_0^1} \leq \|u - I_h u\|_{H_0^1} + \left\|I_h u - R_{h,0}\, u\right\|_{H_0^1} \leq C\|I_h u - u\|_{H_0^1} \leq Ch\|u\|_{H^2}\ .$$

An analogous derivation holds for $R_{h,0}^A$ instead of $R_{h,0}$. This proves the inequalities in (5.20).

The inequality (5.21) is known to hold in the flat case [122, Theorem 4.4.4]. The arguments used by Wardetzky [91, Thereom 3.3.2] to generalize the flat L^2 case to the surface Ω also apply for L^∞, since only pointwise calculations ever take place in the proof.

The inequalities in (5.22) follow by noticing that all calculations above for the interpolation operator only ever happen within triangles and by summation over triangles. □

5.2.3 An L^∞ estimate for the Ritz approximation on curved surfaces

Rannacher and Scott [105] measure the $W^{1,\infty}$ error of the Ritz approximation on convex subsets of $\mathbb{R}^2$. We generalize their result to smooth surfaces with shape-regular triangulations.

Lemma 5.5 (Rannacher and Scott [105])**.** *Let U be a convex, polygonal subset of the plane, triangulated by a shape-regular triangulation. Let $u \in W^{2,\infty}(U) \cap H_0^1(U)$. Then*

$$\left\|u - R_{h,0}\, u\right\|_{W^{1,\infty}(U)} \leq Ch\|u\|_{W^{2,\infty}(U)}\ .$$

We generalize Lemma 5.5 by following the approach of Brenner and Scott [122, Theorem 8.1.11], but applying it to the setting of the Poisson equation solved with linear finite elements on curved surfaces. The most important difference between the planar setting and curved surfaces is the fact that first and second derivatives work somewhat differently: there is no simple notion of a constant vector field with zero derivative, instead we resort to divergence-free, locally-defined vector fields. Moreover, our FEM functions no longer have zero second derivative within curved triangles on the surface, as they do in the planar setting (where they are piecewise linear). Furthermore, we need to apply multiple subtle changes to deal with the definition of various helper functions on curved surfaces.

Theorem 5.3. *Let $u \in W^{2,\infty}(\Omega) \cap H^1_0(\Omega)$, where $\Omega \subseteq \mathbb{R}^3$ is the smooth surface introduced in Section 5.2.2.1. Then it holds that*

$$\left\| R_{h,0}\, u \right\|_{W^{1,\infty}} \leq C \|u\|_{W^{1,\infty}}$$

Proof. We first note that the Ritz approximation is equivalent to solving the Poisson equation with Dirichlet boundary conditions using piecewise linear finite elements. Let $f \coloneqq \Delta u \in L^\infty$. Then $u_h \coloneqq R_{h,0}\, u \in S_{h,0}$ satisfies

$$D\langle u_h, \xi \rangle = \langle f, \xi \rangle \quad \forall \xi \in S_{h,0} \, .$$

This means that the proof boils down to the stability of the linear finite element method for the Poisson equation with Dirichlet boundary conditions.

We will use the method of weighted estimates [122, Section 8.2]. For any $z \in \Omega$ contained in the interior of a triangle T^z (so, for almost any point), consider a smooth function δ^z compactly

supported in an open subset of T^z such that

$$\begin{aligned} \langle \delta^z, \xi \rangle_A &= \xi(z) \quad \forall \xi \in S_{h,0} \\ \left\| D^k \delta^z \right\|_{L^\infty} &\leq C_k h^{-2-k} \quad \forall k \geq 0 \,, \end{aligned} \tag{5.23}$$

where the constant C_k depends only on the triangulation, and D^k is the vector of partial derivatives of order k.

Let $\nabla_A \coloneqq A^{-1}\nabla$ be the gradient pushed from the flat mesh Ω_h to the surface Ω [91, p. 58], and let $\cdot_A$ denote a short-hand for the dot product in the disturbed metric from Definition 5.5. Let $\tilde{v}$ be any tangent vector to Ω at z of length 1 under the disturbed dot product $\cdot_A$. Our goal is to bound $(\tilde{v} \cdot_A \nabla_A u_h)(z)$, which will provide a bound for $\|\nabla_A u_h\|_{L^\infty}$ (since z was arbitrary). By Lemma 5.1, $\|\nabla_A u_h\|_{L^\infty} \leq C \|\nabla u_h\|_{L^\infty}$. As u_h is Lipschitz continuous, $\|u_h\|_{L^\infty} \leq C \|\nabla u_h\|_{L^\infty}$, so a bound for $(\tilde{v} \cdot_A \nabla_A u_h)(z)$ is a bound for $\|u_h\|_{W^{1,\infty}}$.

The vector $\tilde{v}$ can be extended to a vector field on all of T^z via simple translation along the flat triangle $\Phi(T^z)$, resulting in a constant vector field. We have $\operatorname{div}_A \tilde{v} = 0$.

By definition, $\tilde{v} \cdot_A \nabla_A u_h \big|_{T^z}$ is a constant function. This implies

$$\begin{aligned} (\tilde{v} \cdot_A \nabla_A u_h)(z) &= \langle \tilde{v} \cdot_A \nabla_A u_h, \delta^z \rangle_A \qquad \text{(by the definition of } \delta^z\text{)} \\ &= -\langle u_h, \tilde{v} \cdot_A \nabla_A \delta^z \rangle_A \qquad \text{(integration by parts)} \\ &= -\left\langle u_h, |\det A|^{\frac{1}{2}} \tilde{v} \cdot A A^{-1} \nabla \delta^z \right\rangle \\ &= -\langle u_h, \nu \cdot \nabla \delta^z \rangle \,, \end{aligned} \tag{5.24}$$

with $v \coloneqq |\det A|^{\frac{1}{2}} \tilde{v}$. We note that,

$$\begin{aligned} |v|^2 &= v \cdot v \leq C\, v \cdot_A v \quad \text{(by Lemma 5.1 both norms are equivalent)} \\ &\leq C|\det A| \tilde{v} \cdot_A \tilde{v} = C|\det A| \qquad \text{(by the definition of } \tilde{v}\text{)}, \\ |v| &\leq C\left(1 + \left\| 1 - |\det A|^{\frac{1}{2}} \right\|_{L^\infty}\right) \leq C \qquad \text{(by Lemma 5.1, with a different } C\text{)} . \end{aligned} \tag{5.25}$$

Our definition of v makes it divergence-free. This can be seen with the following calculation. Let $\rho \in H_0^1(T^z)$. Denoting by $\mathrm{d}x_A$ the volume element used for integration in the disturbed metric, we obtain

$$\begin{aligned} -\int_{T^z} \rho \ \mathrm{div}\, v \ \mathrm{d}x &= \int_{T^z} \nabla \rho \cdot v \ \mathrm{d}x = \int_{T^z} (A \nabla_A \rho) \cdot v \ \mathrm{d}x = \int_{T^z} \nabla_A \rho \cdot_A v \ \mathrm{d}x \\ &= \int_{T^z} |\det A|^{-\frac{1}{2}} \nabla_A \rho \cdot_A v \ \mathrm{d}x_A = \int_{T^z} \nabla_A \rho \cdot_A \tilde{v} \ \mathrm{d}x_A \\ &= -\int_{T^z} \rho \ \mathrm{div}_A \tilde{v} \ \mathrm{d}x_A = 0 , \end{aligned}$$

and thus

$$\mathrm{div}\, v = 0 . \tag{5.26}$$

We define the helper function $g^z \in H_0^1$ as the solution of a different Poisson problem,

$$D\langle g^z, \xi \rangle = \langle -v \cdot \nabla \delta^z, \xi \rangle \quad \forall \xi \in H_0^1 , \tag{5.27}$$

which is well-defined as δ^z is zero outside of an open set contained in T^z. It has an associated finite element problem with solution $g_h^z \in S_{h,0}$,

$$D\left\langle g_h^z, \xi \right\rangle = \langle -v \cdot \nabla \delta^z, \xi \rangle \quad \forall \xi \in S_{h,0} .$$

We continue the calculation from (5.24),

$$\begin{aligned}
\langle u_h, -\nu \cdot \nabla \delta^z \rangle &= D\big\langle u_h, g_h^z \big\rangle = D\big\langle u, g_h^z \big\rangle = D\langle u, g^z \rangle - D\big\langle u, g^z - g_h^z \big\rangle \\
&= \langle u, -\nu \cdot \nabla \delta^z \rangle - D\big\langle u, g^z - g_h^z \big\rangle \,, \\
\langle u, -\nu \cdot \nabla \delta^z \rangle &= \langle \nu \cdot \nabla u, \delta^z \rangle \qquad \text{by (5.26), therefore} \\
\langle u_h, -\nu \cdot \nabla \delta^z \rangle &= \langle \nu \cdot \nabla u, \delta^z \rangle - D\big\langle u, g^z - g_h^z \big\rangle \,,
\end{aligned} \tag{5.28}$$

using the definitions of g^z, g_h^z, and u_h respectively, as well as integration by parts. The last two terms of (5.28) will now be bounded separately.

The first term from (5.28) can be bounded by a direct calculation using the Hölder inequality,

$$\langle \delta^z, \nu \cdot \nabla u \rangle \leq \|\delta^z\|_{L^1} \|\nu\|_{L^\infty} \|u\|_{W^{1,\infty}} \leq C \|u\|_{W^{1,\infty}} \,.$$

To bound the second term from (5.28) we introduce the weight function $\sigma_z : \Omega \to \mathbb{R}$,

$$\sigma_z(x) = \sqrt{d(x,z)^2 + \kappa^2 h^2} \,. \tag{5.29}$$

where d is the geodesic distance on Ω, and $\kappa > 0$ a constant. Let $U := \{x \in \Omega \mid d(x,z) < \kappa h\}$, $V := \{x \in \Omega \mid d(x,z) \geq \kappa h\}$. The area of U is bounded by $C(\kappa h)^2$. We can estimate, for

small enough $\alpha > 0$,

$$\begin{aligned}
\int_\Omega \sigma_z^{-\alpha-2}\,\mathrm{d}x &= \left(\frac{1}{2}\right)^{\frac{\alpha+2}{2}} \int_\Omega \left(\frac{1}{2}\right)^{\frac{-\alpha-2}{2}} \left(d(x,z)^2 + \kappa^2 h^2\right)^{\frac{-\alpha-2}{2}}\,\mathrm{d}x \\
&\leq 2\left(\frac{1}{2}\right)^{\frac{\alpha+4}{2}} \int_\Omega d(x,z)^{-\alpha-2} + (\kappa h)^{-\alpha-2}\,\mathrm{d}x \qquad \text{(by convexity)} \\
&\leq C\left(\int_U (\kappa h)^{-\alpha-2}\,\mathrm{d}x + \int_V d(x,z)^{-\alpha-2}\,\mathrm{d}x\right) \\
&\leq C\left(\int_0^{\kappa h}\int_0^{2\pi} (\kappa h)^{-\alpha-2}\, r\,\theta(r,\varphi)\,\mathrm{d}\,r\,\mathrm{d}\,\varphi + \int_{\kappa h}^{M(r,\theta)}\int_0^{2\pi} r^{-\alpha-1}\,\theta(r,\varphi)\,\mathrm{d}\,r\,\mathrm{d}\,\varphi\right) \\
&\qquad \text{(switch to polar coordinates with volume form } \theta\text{, bounded from above and} \\
&\qquad \text{below, and surface boundary } M \text{ for integration bounds)} \\
&\leq C\left((\kappa h)^{-\alpha} + \alpha^{-1}\left(1 - (\kappa h)^{-\alpha}\right)\right) \leq C\alpha^{-1}(\kappa h)^{-\alpha}\ .
\end{aligned} \tag{5.30}$$

While this calculation assumes the surface can be parametrized with geodesic polar coordinates around z, a similar argument works if this is not possible by partitioning the surface into parameterizable patches.

Application of Hölder's inequality and (5.30) to the second remaining part of (5.24) gives, for appropriate λ,

$$\begin{aligned}
\left|D\left\langle u, g^z - g_h^z\right\rangle\right| &\leq \|u\|_{W^{1,\infty}} \int_\Omega \left|\nabla\left(g^z - g_h^z\right)\right|\mathrm{d}x \\
&\leq \|u\|_{W^{1,\infty}} \left(\int_\Omega \left|\nabla\left(g^z - g_h^z\right)\right|^2 \sigma_z^{2+\lambda}\,\mathrm{d}x\right)^{\frac{1}{2}} \left(\int_\Omega \sigma_z^{-2-\lambda}\,\mathrm{d}x\right)^{\frac{1}{2}} \\
&\leq C\lambda^{-1}(\kappa h)^{-\frac{\lambda}{2}} \|u\|_{W^{1,\infty}} \left(\int_\Omega \left|\nabla\left(g^z - g_h^z\right)\right|^2 \sigma_z^{2+\lambda}\,\mathrm{d}x\right)^{\frac{1}{2}}\ .
\end{aligned} \tag{5.31}$$

By Lemma B.7, for appropriate $\kappa, \lambda > 0$, we have $\int_\Omega \left|\nabla\left(g^z - g_h^z\right)\right|^2 \sigma_z^{2+\lambda}\,\mathrm{d}x \leq C_{\kappa,\lambda} h^\lambda$. Plugging this into (5.31) gives

$$\left| D\left\langle u, g^z - g_h^z\right\rangle\right| \leq C_{\kappa,\lambda}\lambda^{-1}\kappa^{-\frac{\lambda}{2}}\|u\|_{W^{1,\infty}}\,,$$

Having bounded the two terms from (5.28), we now have

$$\|\nabla u_h\|_{L^\infty} \leq C\|u\|_{W^{1,\infty}}\,,$$

where finally the constant C depends on κ and λ. This proves the theorem. □

Corollary 5.1. *Let $u \in W^{2,\infty}(\Omega)$. Then we have*

$$\left\|u - R_{h,0}\,u\right\|_{W^{1,\infty}} \leq Ch\|u\|_{W^{2,\infty}}$$

Proof. Using the vertexwise interpolation I_h from Wardetzky [91, Section 3.3.2], as well as estimates for the flat case [86] (which transfer seamlessly [91, Lemma 3.3.1]), we know that

$$\begin{aligned}\left\|u - R_{h,0}\,u\right\|_{W^{1,\infty}} &\leq \|u - I_h u\|_{W^{1,\infty}} + \left\|I_h u - R_{h,0}\,u\right\|_{W^{1,\infty}} \leq \|u - I_h u\|_{W^{1,\infty}} + \left\|R_{h,0}\,(I_h u - u)\right\|_{W^{1,\infty}} \\ &\leq \|u - I_h u\|_{W^{1,\infty}} + C\|I_h u - u\|_{W^{1,\infty}} \leq Ch\|u\|_{W^{2,\infty}}\,.\end{aligned}$$

□

5.2.4 The theory behind the mixed finite element method

With the discrete geometry and discrete function spaces in place, we can now turn towards discretizing the problem (5.15) in its mixed form (5.16). We use the mixed finite element approach of Ciarlet [116, Section 7].

Using the disturbed and undisturbed inner products on Ω leads to two discrete mixed problems. The discrete problem with respect to the undisturbed metric takes the form

$$\begin{aligned} D\langle \tilde{u}_2^h, \xi \rangle &= \langle f, \xi \rangle \qquad \forall \xi \in S_{h,0} \\ D\langle \tilde{u}_1^h, \eta \rangle &= \langle \tilde{u}_2^h, \eta \rangle \qquad \forall \eta \in S_h \ , \end{aligned} \tag{5.32}$$

where $\tilde{u}_1^h \in S_{h,0}$, $\tilde{u}_2^h \in S_h$, and $f \in L^2$. This problem is only an auxiliary problem for our proof. Its operators are never computed in practice. In practice, one only solves the discrete problem with respect to the disturbed metric,

$$\begin{aligned} D\langle u_2^h, \xi \rangle_A &= \langle f, \xi \rangle_A \qquad \forall \xi \in S_{h,0} \\ D\langle u_1^h, \eta \rangle_A &= \langle u_2^h, \eta \rangle_A \qquad \forall \eta \in S_h \ , \end{aligned} \tag{5.33}$$

where $u_1^h \in S_{h,0}$, $u_2^h \in S_h$, and $f \in L^2$. If the surface has no boundary, the solutions of (5.32) and (5.33) additionally have to fulfill the zero mean property from Definition 5.2.

Notice that, although the latter problem is formulated in terms of the distorted inner products on Ω, it is actually the discrete mixed problem with operators computed on the mesh Ω_h. The distorted L^2 inner product corresponds to the mass matrix, and the distorted H_0^1 inner product corresponds to the Lagrangian piecewise linear FEM stiffness matrix. In the planar case of Scholz

[102] the two discrete problems coincide.

Existence and uniqueness for (5.32) and (5.33) follow from an argument by Ciarlet [116, Section 7], which we repeat here for convenience.

Definition 5.9. *We define the following three linear spaces:*

$$\begin{aligned} V &:= \{(v_1, v_2) \in H_0^1 \times L^2 \quad | \quad D\langle v_1, \mu\rangle = \langle v_2, \mu\rangle \ \forall \mu \in H^1\} \\ V^h &:= \{(v_1, v_2) \in S_{h,0} \times S_{hL^2} \quad | \quad D\langle v_1, \mu\rangle = \langle v_2, \mu\rangle \ \forall \mu \in S_h\} \\ V_A^h &:= \{(v_1, v_2) \in S_{h,0} \times S_{hL^2} \quad | \quad D\langle v_1, \mu\rangle_A = \langle v_2, \mu\rangle_A \ \forall \mu \in S_h\} \ , \end{aligned} \tag{5.34}$$

where the space S_{hL^2} is the space S_h, but with the L^2 norm instead of its usual H^1 norm.

These spaces are Hilbert spaces, as they are closed linear subspaces of a Hilbert space. As an immediate consequence of Poincaré's inequality we obtain that the resulting product norms on these spaces are equivalent to (simpler) norms that we heavily rely on going forward:

Lemma 5.6. *The product norms on V, V^h, V_A^h are equivalent to the norms induced by the inner products*

$$\begin{aligned} (v_1, v_2), (w_1, w_2) &\mapsto \langle v_2, w_2\rangle \quad \text{on } V, V^h \\ (v_1, v_2), (w_1, w_2) &\mapsto \langle v_2, w_2\rangle_A \quad \text{on } V_A^h \ . \end{aligned} \tag{5.35}$$

Proof. The symmetric bilinear mappings defined by (5.35) are indeed positive definite since $v_2 = 0$ implies $v_1 = 0$ for all elements (v_1, v_2) in V, V^h, V_A^h. Poincaré's inequality implies that

$$\|v_1\|_{H_0^1}^2 = D\langle v_1, v_1\rangle = \langle v_2, v_1\rangle \leq \|v_2\|_{L^2}\|v_1\|_{L^2} \leq C\|v_2\|_{L^2}\|v_1\|_{H_0^1} \ .$$

Therefore, $\|v_1\|_{H_0^1} \leq C\|v_2\|_{L^2}$, which proves the lemma for V. By Lemma 5.2, an identical derivation holds for V^h, V_A^h. □

On these linear spaces we can define the functionals

$$\begin{aligned} J((v_1, v_2)) &\coloneqq \frac{1}{2}\langle v_2, v_2\rangle - F((v_1, v_2)) \\ J_A((v_1, v_2)) &\coloneqq \frac{1}{2}\langle v_2, v_2\rangle_A - F((v_1, v_2)) \end{aligned} \tag{5.36}$$

for some dual function $F \in V', \left(V^h\right)', \left(V_A^h\right)'$.

The functionals from (5.36) have unique minimizers:

Lemma 5.7 (Existence and uniqueness for the mixed biharmonic problem)**.** *Minimizing* J *over* V, V^h *and minimizing* J_A *over* V_A^h *has a unique solution.*

Proof. Define real-valued bilinear forms B on V, V^h, V_A^h by

$$\begin{aligned} B\left((v_1, v_2),\ (w_1, w_2)\right) &= \langle v_2, w_2\rangle \quad \text{for } V, V^h \\ B\left((v_1, v_2),\ (w_1, w_2)\right)_A &= \langle v_2, w_2\rangle_A \quad \text{for } V_A^h\ . \end{aligned}$$

These bilinear forms are symmetric, bounded by Cauchy–Schwarz, and elliptic by Lemma 5.2. F is bounded by definition. It holds $J(v) = \frac{1}{2}B(v, v) - F(v)$ and $J_A(v) = \frac{1}{2}B(v, v)_A - F(v)$ for $v = (v_1, v_2)$. Thus, each of the functionals J, J_A has a unique minimizer. □

We write

$$\begin{aligned}(u_1, u_2) &= \operatorname*{argmin}_{(v_1,v_2)\in V} J((v_1, v_2))\\ \left(\tilde{u}_1^h, \tilde{u}_2^h\right) &= \operatorname*{argmin}_{(v_1,v_2)\in V^h} J((v_1, v_2))\\ \left(u_1^h, u_2^h\right) &= \operatorname*{argmin}_{(v_1,v_2)\in V_A^h} J_A((v_1, v_2))\ .\end{aligned} \tag{5.37}$$

The Euler–Lagrange equations corresponding to these minimization problems take the form

$$\begin{aligned}\langle u_2, v_2\rangle &= F((v_1, v_2)) \qquad \forall (v_1, v_2) \in V\\ \left\langle \tilde{u}_2^h, v_2\right\rangle &= F((v_1, v_2)) \qquad \forall (v_1, v_2) \in V^h\\ \left\langle u_2^h, v_2\right\rangle_A &= F((v_1, v_2)) \qquad \forall (v_1, v_2) \in V_A^h\ .\end{aligned} \tag{5.38}$$

Combined with the linear constraints from the definition of the spaces V, V^h, V_A^h in (5.34), these are the mixed problems from (5.16), (5.32), and (5.33), respectively, by defining $F(v_1, v_2) \coloneqq \langle v_1, f\rangle$ on V, V^h and $F(v_1, v_2) \coloneqq \langle v_1, f\rangle_A$ on V_A^h.

Lemma 5.7 ensures existence and uniqueness of the mixed formulation of the biharmonic equation. Notice, however, that Lemma 5.7 does not show that $u_1 \in H^4$. Indeed, one requires an additional argument to show that $u_1 \in H^2$ [116, Theorem 7.1.1], and then standard regularity estimates [33, Section 2.5.2] to show $u_1 \in H^4$. If these regularity estimates do not hold—for example because the surface or its boundary are not smooth enough—the minimization problem on V still has a unique minimizer, *but this minimizer does not solve the weak biharmonic equation* (5.15), see Stylianou [117, Section 4.3]. This in itself does not lead to problems in our setting, since we

assume regularity of u_1. For the same reason, however, Lemma 5.7 only shows that the H^1_0 norms of $\tilde{u}_1^h, u_1^h$ are bounded independently of h and likewise, that the L^2 norms of $\tilde{u}_2^h, u_2^h$ are bounded independently of h. We *cannot* infer boundedness of the H^1 norms of $\tilde{u}_2^h, u_2^h$ independently of h – these norms could blow up as h decreases.

5.2.5 The convergence proof

It is somewhat surprising that the derivatives of $\tilde{u}_2^h, u_2^h$ appear in the linear systems that we are solving, but the L^2-norms of these derivatives cannot be bounded independently of h. This indeed complicates the task of bounding errors between solutions of (5.16), (5.32), and (5.33). Scholz [102] elegantly solves this issue by utilizing the Ritz interpolation in order to cancel contributions of derivatives of $\tilde{u}_2^h$. However, this approach ceases to work for the case of curved geometries that are approximated by piecewise flat geometries. This is precisely why the curved case is more intricate than the flat one.

In the planar case, we have that $u_1^h = \tilde{u}_1^h$ and $u_2^h = \tilde{u}_2^h$ by construction. Scholz [102] splits up the proof for the planar case into showing that u_2^h converges to u_2 and that u_1^h converges to u_1. In the curved case, a similar argument only works to show that $\tilde{u}_2^h$ converges to u_2, and $\tilde{u}_1^h$ converges to u_1, using the generalized L^∞ estimate of Section 5.2.3. Therefore, we require an additional step in order to show that u_2^h converges to $\tilde{u}_2^h$.

5.2.5.1 *Convergence of the discrete problem on the mesh to the discrete problem on the surface*

In this section and Section 5.2.5.2 we treat the case of surfaces with nonempty boundary; the case of empty boundaries (treated later) is significantly simpler.

We rewrite the linear problem from (5.38) using linear operators.

Lemma 5.8. *Let $L : V \to V'$, $L^h : V^h \to \left(V^h\right)'$, and $L^h_A : V^h_A \to \left(V^h_A\right)'$ denote the respective Riesz maps using the inner products defined in Lemma 5.6. The mixed biharmonic equation can be written as*

$$Lu = F\,, \quad L^h u^h = F^h\,, \quad L^h_A u^h_A = F^h_A, \tag{5.39}$$

for $F(v_1, v_2) \coloneqq \langle v_1, f\rangle$ on V, $F^h(v_1, v_2) \coloneqq \langle v_1, f\rangle$ on V_h, and $F^h_A(v_1, v_2) \coloneqq \langle v_1, f\rangle_A$ on V^h_A.

So far, our treatment for the mixed biharmonic problem has considered the smooth setting alongside the two discrete settings. The next step, however, only works in the two discrete settings. We define a discrete Laplace operator T that maps into the L^2-like space S_{hL^2} from $S_{h,0}$. Such a construction is of course not possible in the smooth setting. In the discrete setting, however, we can identify the dual space of $S_{h,0}$ with a subspace of S_{hL^2}.

Lemma 5.9 (Discrete Laplacians)**.** *There exist bounded linear and injective operators $T^h, T^h_A : S_{h,0} \to S_{hL^2}$ such that*

$$\begin{aligned} \left(v_1, T^h v_1\right) &\in V^h \qquad \forall v_1 \in S_{h,0} \\ \left(v_1, T^h_A v_1\right) &\in V^h_A \qquad \forall v_1 \in S_{h,0}\,. \end{aligned} \tag{5.40}$$

Moreover, every element in V^h can be written as the pair $\left(v_1, T^h v_1\right)$, and every element in V_A^h can be written as the tuple $\left(v_1, T_A^h v_1\right)$.

Proof. We prove the lemma for T^h, the proof for T_A^h is similar. For all $(v_1, v_2) \in V^h$ we have that

$$D\langle v_1, \mu\rangle = \langle v_2, \mu\rangle \qquad \forall \mu \in S_h \ .$$

Written as a discrete linear equation, the right-hand side involves the mass matrix M for piecewise linear Lagrange finite elements. This matrix is invertible. We can thus define $T^h v_1 := v_2 = M^{-1} S v_1$, where S denotes the discrete Laplacian stiffness matrix with columns in $S_{h,0}$ and rows in S_h. The resulting operator is well-defined and linear.

Injectivity follows from the solvability of the Poisson equation. Indeed, $D\langle v_1, \eta\rangle = 0 \ \forall \eta \in S_{h,0}$ has a unique solution $v_1 = 0 \in S_{h,0}$. This implies that the discrete Laplacian stiffness matrix has full column rank (it is not invertible, since we test with all $\mu \in S_h$). Since the mass matrix is invertible, T^h has full column rank and is thus injective.

It remains to show that every element in V^h can be written as a pair $\left(v_1, T^h v_1\right)$. In order to see that, let $(v_1, v_2) \in V^h$. Then, the definition of V^h implies that

$$0 = D\langle 0, \mu\rangle = \left\langle v_2 - T^h v_1, \mu\right\rangle \qquad \forall \mu \in S_h \ .$$

Therefore, $T^h v_1 = v_2$. □

Remark 5.7. *The linear operators T^h, T_A^h are bounded, as they are discrete operators. This bound, however, is* not *independent of h.*

The next result is central for relating solutions from the two discrete spaces.

Lemma 5.10 (Inverse estimate)**.** *We have that*

$$\left\|T^h v_1 - T_A^h v_1\right\|_{L^2} \leq Ch^{\sigma-1}\|v_1\|_{H_0^1} \qquad \forall v_1 \in S_{h,0}\ ,$$

where the constant C is independent of h.

Proof. For $\mu \in S_{hL^2}$ it holds that

$$\begin{aligned}\left\langle T^h v_1 - T_A^h v_1, \mu\right\rangle &= \left\langle T^h v_1, \mu\right\rangle - \left\langle T_A^h v_1, \mu\right\rangle_A - c(T_A^h v_1, \mu) = D\langle v_1, \mu\rangle - D\langle v_1, \mu\rangle_A - c(T_A^h v_1, \mu)\\ &= d(v_1, \mu) - c(T_A^h v_1, \mu) \leq Ch^{\sigma}\|v_1\|_{H_0^1}\|\mu\|_{H_0^1} + Ch^{\sigma}\left\|T_A^h v_1\right\|_{L^2}\|\mu\|_{L^2}\ .\end{aligned}$$

Using the standard inverse estimate, we have that $\left\|T_A^h v_1\right\|_{H_0^1} \leq Ch^{-1}\left\|T_A^h v_1\right\|_{L^2}$. Therefore,

$$\begin{aligned}\left\|T_A^h v_1\right\|_{L^2}^2 &\leq C\left\|T_A^h v_1\right\|_{L_A^2}^2 = C\, D\left\langle v_1, T_A^h v_1\right\rangle_A \leq Ch^{-1}\|v_1\|_{H_0^1}\left\|T_A^h v_1\right\|_{L^2}\\ \left\|T_A^h v_1\right\|_{L^2} &\leq Ch^{-1}\|v_1\|_{H_0^1}\ ,\end{aligned}$$

which proves the lemma. □

Using the maps T^h, T_A^h allows for constructing a map that relates the two spaces V^h and V_A^h.

Lemma 5.11. *Let $W : V^h \to V_A^h$ be the linear map such that*

$$W\left(\left(v_1, T^h v_1\right)\right) = \left(v_1, T_A^h v_1\right)\ . \tag{5.41}$$

W is well-defined, bounded independently of h, invertible, and the inverse is also bounded independently of h.

Proof. Well-definedness is clear by Lemma 5.9. W is invertible, as the inverse mapping is given by $\left(v_1, T_A^h v_1\right) \mapsto \left(v_1, T^h v_1\right)$.

We now show the bound for W; a similar argument works to show boundedness of the inverse. Let $v_1 \in S_{h,0}$. Then we obtain

$$\begin{aligned}\left\|W\left(\left(v_1, T^h v_1\right)\right)\right\| = \left\|T_A^h v_1\right\|_{L_A^2} &\leq C\left\|T_A^h v_1\right\|_{L^2} \leq C\left\|T^h v_1\right\|_{L^2} + C\left\|T_A^h v_1 - T^h v_1\right\|_{L^2} \\ &\leq C\left\|T^h v_1\right\|_{L^2} + Ch^{\sigma-1}\|v_1\|_{H_0^1},\end{aligned}$$

where the last inequality follows from Lemma 5.10. Using that $\sigma > 1$ and the equivalence of norms from Lemma 5.6, the claim follows. Thus, W is bounded independently of h. □

We denote by $W' : \left(V_A^h\right)' \to \left(V^h\right)'$ the dual operator to W. This operator is bounded independently of h, it is invertible, and its inverse is bounded independently of h. The operators W, W' provide the tool for relating the two discrete problems.

Lemma 5.12. *We have*

$$\left\|L^h W^{-1} - W' L_A^h\right\| \leq Ch^{\sigma-1} .$$

Proof. Let $\left(v_1, T_A^h v_1\right) \in V_A^h$, $\left(w_1, T^h w_1\right) \in V^h$. Then

$$\begin{aligned}\left(L^h W^{-1} - W' L_A^h\right)\left(\left(v_1, T_A^h v_1\right)\right)\left(\left(w_1, T^h w_1\right)\right) &= \left\langle T^h v_1, T^h w_1\right\rangle - \left\langle T_A^h v_1, T_A^h w_1\right\rangle_A \\ &= c(T_A^h v_1, T_A^h w_1) + \left\langle T^h v_1 - T_A^h v_1, T^h w_1\right\rangle \\ &\quad + \left\langle T_A^h v_1, T^h w_1 - T_A^h w_1\right\rangle \\ &\leq Ch^{\sigma}\left\|T_A^h v_1\right\|_{L^2}\left\|T^h w_1\right\|_{L^2} + Ch^{\sigma-1}\left\|T_A^h v_1\right\|_{L^2}\left\|T^h w_1\right\|_{L^2} \\ &\leq Ch^{\sigma-1}\left\|T_A^h v_1\right\|_{L^2}\left\|T^h w_1\right\|_{L^2} ,\end{aligned}$$

where we used Lemma 5.6 and Lemma 5.10. □

We can now estimate the error for the solutions of the two discrete linear systems in Lemma 5.8.

Lemma 5.13. *Consider the following two linear problems for $F^h \in \left(V^h\right)'$, $F_A^h \in \left(V_A^h\right)'$,*

$$L^h u^h = F^h \quad \textit{and} \quad L_A^h u_A^h = F_A^h \,. \tag{5.42}$$

Then we have that

$$\left\|W u^h - u_A^h\right\| \leq C h^{\sigma-1} \left\|F_A^h\right\| + C \left\|F^h - W' F_A^h\right\| \,.$$

Proof. By assumption we have that

$$\begin{aligned} W' L_A^h u_A^h = W' F_A^h \quad &\Rightarrow \quad u_A^h = \left(W' L_A^h\right)^{-1} W' F_A^h \\ u^h = \left(L^h\right)^{-1} F^h \quad &\Rightarrow \quad W u^h = \left(L^h W^{-1}\right)^{-1} F^h \,. \end{aligned}$$

Combining these gives

$$\begin{aligned} W u^h &= \left(L^h W^{-1}\right)^{-1} (W' g_A^h) + \left(L^h W^{-1}\right)^{-1} (g^h - W' g_A^h) \\ W u^h - u_A^h &= \left(\left(L^h W^{-1}\right)^{-1} - \left(W' L_A^h\right)^{-1}\right) (W' F_A^h) + \left(L^h W^{-1}\right)^{-1} (F^h - W' F_A^h) \,. \end{aligned}$$

Using Lemma 5.12 and the fact that all relevant operators are invertible and bounded independently of h gives

$$\left\|\left(L^h W^{-1}\right)^{-1} - \left(W' L_A^h\right)^{-1}\right\| \leq \left\|\left(L^h W^{-1}\right)^{-1}\right\| \left\|L^h W^{-1} - W' L_A^h\right\| \left\|\left(W' L_A^h\right)^{-1}\right\| \leq C h^{\sigma-1} \,,$$

and thus

$$\left\|Wu^h - u_A^h\right\| \leq Ch^{\sigma-1}\left\|F_A^h\right\| + C\left\|F^h - W'F_A^h\right\| .$$

□

The main result of this section relates the two discrete problems (5.32) and (5.33) that we use in our mixed finite element method.

Theorem 5.4. *Let $\tilde{u}_1^h, \tilde{u}_2^h$ solve problem (5.32), and let u_1^h, u_2^h solve problem (5.33). Then*

$$\left\|u_2^h - \tilde{u}_2^h\right\|_{L^2} \leq Ch^{\sigma-1}\|f\|_{L^2} .$$

Proof. Let $u^h = \left(\tilde{u}_1^h, \tilde{u}_2^h\right)$, $u_A^h = \left(u_1^h, u_2^h\right)$, $F^h((v_1, v_2)) = \langle f, v_1\rangle$, $F_A^h((v_1, v_2)) = \langle f, v_1\rangle_A$. Then

$$\begin{aligned}
\left\|u_2^h - \tilde{u}_2^h\right\|_{L^2} &= \left\|T_A^h u_1^h - T^h\tilde{u}_1^h\right\|_{L^2} \\
&\leq \left\|T_A^h u_1^h - T_A^h\tilde{u}_1^h\right\|_{L^2} + \left\|T_A^h\tilde{u}_1^h - T^h\tilde{u}_1^h\right\|_{L^2} \\
&\leq C\left\|u_A^h - Wu^h\right\| + \left\|T_A^h\tilde{u}_1^h - T^h\tilde{u}_1^h\right\|_{L^2} \quad \text{(by Lemma 5.11)} \\
&\leq C\left\|u_A^h - Wu^h\right\| + Ch^{\sigma-1}\left\|\tilde{u}_1^h\right\|_{H_0^1} \quad \text{(by Lemma 5.10)} \\
&\leq C\left\|u_A^h - Wu^h\right\| + Ch^{\sigma-1}\left\|F^h\right\| \quad \text{(by Lemma 5.8)} \\
&\leq Ch^{\sigma-1}\left(\left\|F^h\right\| + \left\|F_A^h\right\|\right) + C\left\|F^h - W'F_A^h\right\| \quad \text{(by Lemma 5.13)} .
\end{aligned}$$

It remains to address the right-hand sides. By the equivalence of norms, $\left\|F^h\right\|, \left\|F^h_A\right\| \leq C\|f\|_{L^2}$. For $\left(v_1, T^h v_1\right) \in V^h$ we have that

$$\begin{aligned}
\left(F^h - W' F^h_A\right)\left(\left(v_1, T^h v_1\right)\right) &= \langle f, v_1\rangle - \langle f, v_1\rangle_A = c(f, v_1) \quad \text{and hence} \\
\left|\left(F^h - W' F^h_A\right)\left(\left(v_1, T^h v_1\right)\right)\right| &\leq Ch^\sigma \|f\|_{L^2}\left\|T^h v_1\right\|_{L^2} \quad \text{(by Lemma 5.6)}, \\
\left\|F^h - W' F^h_A\right\| &\leq Ch^\sigma \|f\|_{L^2}\,.
\end{aligned}$$

This proves the result. □

5.2.5.2 *Convergence of the discrete problem on the surface to the exact solution*

Having successfully bounded the error between the discrete problem on the mesh (with solution $\left(u^h_1, u^h_2\right)$) and the discrete problem on the surface (with solution $\left(\tilde{u}^h_1, \tilde{u}^h_2\right)$), we move on to bounding the error between $\left(\tilde{u}^h_1, \tilde{u}^h_2\right)$ and the exact solution, (u_1, u_2). Our proof follows the roadmap laid out by Scholz [102]. However, we require considerable adjustments to extend this approach to curved surfaces.

We start with an extension of Scholz's Lemma to curved surfaces, using the generalization of the theorem of Rannacher and Scott [105] from Section 5.2.3.

Lemma 5.14 (Scholz's Lemma)**.** *Let $u \in H^1_0 \cap W^{2,\infty}$. Let $\eta \in S_h$. Then*

$$\left|D\left\langle u - R_{h,0}\, u, \eta\right\rangle\right| \leq C\sqrt{h}\|u\|_{W^{2,\infty}}\|\eta\|_{L^2}\,.$$

Proof. The proof is virtually identical to the proof of Scholz [102]. Let $\xi \in S_{h,0}$ interpolate η on all interior vertices of the mesh. Let $\varphi := \eta - \xi$.

By the definition of Ritz interpolation, we have that

$$D\big\langle u - R_{h,0}\, u, \eta \big\rangle = D\big\langle u - R_{h,0}\, u, \varphi \big\rangle \ .$$

As φ is only supported on the boundary triangles T, the last equation can be simplified to

$$\big| D\big\langle u - R_{h,0}\, u, \eta \big\rangle \big| = \left| \sum_{t \in T} \int_t \nabla \left(u - R_{h,0}\, u \right) \cdot \nabla \varphi \ \mathrm{d}x \right| \leq C h^2 \big\| u - R_{h,0}\, u \big\|_{W^{1,\infty}} \sum_{t \in T} \| \varphi \|_{W^{1,\infty}(t)} \ ,$$

where we used the fact that the area of a triangle is bounded by Ch^2, where the C depends on the triangle regularity constants.

By the standard inverse estimate we can conclude that $\| \varphi \|_{W^{1,\infty}(t)} \leq C h^{-1} \| \varphi \|_{L^\infty(t)}$. By definition, $\| \varphi \|_{L^\infty(t)} \leq C \| \eta \|_{L^\infty(t)}$. Moreover, using a per-triangle calculation, we can obtain that $\| \eta \|_{L^\infty(t)} \leq C h^{-1} \| \eta \|_{L^2(t)}$. Thus we conclude

$$\big| D\big\langle u - R_{h,0}\, u, \eta \big\rangle \big| \leq C \big\| u - R_{h,0}\, u \big\|_{W^{1,\infty}} \sum_{t \in T} \| \eta \|_{L^2(t)} \leq C h^{-\frac{1}{2}} \big\| u - R_{h,0}\, u \big\|_{W^{1,\infty}} \| \eta \|_{L^2} \ ,$$

where we used the fact that the number of triangles in T is $\sim h^{-1}$.

The estimate by Rannacher and Scott [105], which is generalized to curved surfaces in Corollary 5.1, states that

$$\big\| u - R_{h,0}\, u \big\|_{W^{1,\infty}} \leq C h \| u \|_{W^{2,\infty}} \ ,$$

which proves the lemma.

□

Using this lemma we can now estimate the error in u_2. This mirrors the first part of Theorem 1 by Scholz [102], but we achieve a bound of order $\sqrt{h}$ instead of Scholz's $\sqrt{h}|\log h|^2$ due to the improved Lemma 5.14.

Theorem 5.5. *Let u_1, u_2 solve the smooth mixed biharmonic problem (5.16), and let u_1^h, u_2^h solve the discrete mixed biharmonic problem (5.33) on the mesh. Then one has*

$$\left\|u_2 - u_2^h\right\|_{L^2} \leq C\sqrt{h}\|f\|_{L^2}.$$

Proof. Using Theorem 5.4 (together with the fact that $\sigma \geq \frac{3}{2}$), it suffices to prove the estimate with $\tilde{u}_2^h$ instead of u_2^h. Using Lemma 5.4, it is enough to prove the estimate with $R_h u_2$ instead of u_2.

We note that

$$D\left\langle \tilde{u}_1^h - R_{h,0}\, u_1, \tilde{u}_2^h - R_h\, u_2 \right\rangle = D\left\langle \tilde{u}_1^h - R_{h,0}\, u_1, \tilde{u}_2^h - u_2 \right\rangle = 0\ ,$$

using the definition of Ritz interpolation for the first equality and using the smooth and discrete formulations of the mixed biharmonic problems for the second equality.

Thus we can compute

$$\begin{aligned}\left\|\tilde{u}_2^h - R_h\, u_2\right\|_{L^2}^2 &= \left\langle \tilde{u}_2^h - R_h\, u_2, \tilde{u}_2^h - R_h\, u_2 \right\rangle - D\left\langle \tilde{u}_1^h - R_{h,0}\, u_1, \tilde{u}_2^h - R_h\, u_2 \right\rangle \\ &= \left\langle u_2 - R_h\, u_2, \tilde{u}_2^h - R_h\, u_2 \right\rangle + D\left\langle R_{h,0}\, u_1 - u_1, \tilde{u}_2^h - R_h\, u_2 \right\rangle\ ,\end{aligned}$$

where we again used the (smooth and discrete) formulations of the mixed biharmonic problems. The first of the two remaining terms can be estimated using (5.20). The second remaining

term is covered by Lemma 5.14 and the fact that $\|u_1\|_{W^{2,\infty}} \leq C\|u_1\|_{H^4} \leq C\|f\|_{L^2}$. Division by $\left\|\tilde{u}_2^h - R_h\, u_2\right\|_{L^2}$ then proves the theorem. □

It remains to compute the error in u_1. The next theorem follows the second part of Theorem 1 by Scholz [102], but requires additional work due to the curved geometries. Because of Lemma 5.14, we achieve convergence of order h here.

Theorem 5.6. *We have that*

$$\left\|u_1 - u_1^h\right\|_{L^2} \leq Ch\|f\|_{L^2}.$$

Proof. Since $u_1 - u_1^h \in H_0^1$, by assumption the biharmonic equation $\Delta^2 w = u_1 - u_1^h$ with zero Dirichlet and Neumann boundary conditions has a unique solution $w \in H_0^1 \cap H^4$. As before, we use the geometers' convention that the Laplacian be *positive* semidefinite.

We use the mixed biharmonic PDEs, Ritz interpolation, and integration by parts repeatedly to obtain

$$\begin{aligned}
\left\|u_1 - u_1^h\right\|_{L^2}^2 &= \left\langle u_1 - u_1^h, \Delta^2 w\right\rangle = D\left\langle u_1 - u_1^h, \Delta w\right\rangle \\
&= D\left\langle u_1 - u_1^h, \Delta w - R_h\,\Delta w\right\rangle + \left\langle u_2 - u_2^h, R_h\,\Delta w\right\rangle - d(u_1^h, R_h\,\Delta w) + c(u_2^h, R_h\,\Delta w) \\
&= D\left\langle u_1 - u_1^h, R_h\,\Delta w - \Delta w\right\rangle + \left\langle u_2 - u_2^h, \Delta w - R_h\,\Delta w\right\rangle + D\left\langle u_2 - u_2^h, w\right\rangle \\
&\quad + d(u_1^h, R_h\,\Delta w) - c(u_2^h, R_h\,\Delta w) \\
&= D\left\langle u_1 - u_1^h, \Delta w - R_h\,\Delta w\right\rangle + D\left\langle u_2 - u_2^h, w - R_{h,0}\, w\right\rangle + \left\langle u_2 - u_2^h, R_h\,\Delta w - \Delta w\right\rangle \\
&\quad - d(u_2^h, R_{h,0}\, w) + c(f, R_{h,0}\, w) - d(u_1^h, R_h\,\Delta w) + c(u_2^h, R_h\,\Delta w)\,.
\end{aligned}$$

Using (5.20), the first term of the last expression can be bounded by

$$\left| D\langle u_1 - u_1^h, \Delta w - R_h \Delta w\rangle\right| = \left| D\langle u_1 - R_{h,0}\, u_1, \Delta w - R_h \Delta w\rangle\right| \leq Ch^2 \|f\|_{L^2} \|w\|_{H^4} \ .$$

We can bound the second term as follows,

$$\begin{aligned}
\left| D\langle u_2 - u_2^h, w - R_{h,0}\, w\rangle\right| &\leq \left| D\langle u_2 - R_h\, u_2, w - R_{h,0}\, w\rangle\right| + \left| D\langle u_2^h - R_h\, u_2, w - R_{h,0}\, w\rangle\right| \\
&\leq Ch^2 \|u_2\|_{H^2} \|w\|_{H^2} + \left| D\langle u_2^h - R_h\, u_2, w - R_{h,0}\, w\rangle\right| \quad \text{(by (5.20))} \\
&\leq Ch^2 \|u_2\|_{H^2} \|w\|_{H^2} + Ch \|f\|_{L^2} \|w\|_{W^{2,\infty}} \quad \text{(by Lemma 5.14 and Theorem 5.5)} \\
&\leq Ch \|f\|_{L^2} \|w\|_{H^4} \ .
\end{aligned}$$

Using (5.20) and Theorem 5.5, the bound for the third term is

$$\left| \langle u_2 - u_2^h, \Delta w - R_h \Delta w\rangle\right| \leq \left| \langle R_h\, u_2 - u_2^h, \Delta w - R_h \Delta w\rangle + \langle u_2 - R_h\, u_2, \Delta w - R_h \Delta w\rangle\right| \leq Ch^{\frac{3}{2}} \|f\|_{L^2} \|w\|_{H^4} \ .$$

Finally, three of the remaining terms can be bounded as

$$\left| d(u_1^h, R_h \Delta w)\right| + \left| c(u_2^h, R_h \Delta w)\right| + \left| c(f, R_{h,0}\, w)\right| \leq Ch^\sigma \|f\|_{L^2} \|w\|_{H^4} \ ,$$

where we used Lemma 5.3 and (5.20).

In order to bound the last remaining term, observe that

$$\begin{aligned}
\left|d(u_2^h, R_{h,0}\, w)\right| &\leq Ch^\sigma \left\|u_2^h\right\|_{H_0^1} \left\|R_{h,0}\, w\right\|_{H_0^1} \leq Ch^\sigma \left\|u_2^h\right\|_{H_0^1} \|w\|_{H_0^1} \,, \\
\left\|u_2^h\right\|_{H_0^1} &\leq \|R_h\, u_2\|_{H_0^1} + \left\|u_2^h - R_h\, u_2\right\|_{H_0^1} \leq C\|u_2\|_{H_0^1} + Ch^{-1}\left\|u_2^h - R_h\, u_2\right\|_{L^2} \\
&\leq C\|f\|_{L^2} + Ch^{-\frac{1}{2}}\|f\|_{L^2} \leq Ch^{-\frac{1}{2}}\|f\|_{L^2} \,, \\
\left|d(u_2^h, R_{h,0}\, w)\right| &\leq Ch^{\sigma-\frac{1}{2}}\|f\|_{L^2}\|w\|_{H_0^1} \,,
\end{aligned}$$

where we used Lemma 5.3, Lemma 5.4, and Theorem 5.5.

Using that $\Delta^2 w = u_1 - u_1^h$, we obtain $\|w\|_{H^4} \leq C\left\|u_1 - u_1^h\right\|_{L^2}$. Together with the assumption that $\sigma \geq \frac{3}{2}$, these estimates show that

$$\left\|u_1 - u_1^h\right\|_{L^2}^2 \leq Ch\|f\|_{L^2}\left\|u_1 - u_1^h\right\|_{L^2} \,,$$

which proves the theorem. □

A simple corollary provides a convergence rate of $h^{\frac{3}{4}}$ for the gradient of u_1.

Corollary 5.2. *We have that*

$$\left\|u_1 - u_1^h\right\|_{H_0^1} \leq Ch^{\frac{3}{4}}\|f\|_{L^2}$$

Proof. Using the mixed biharmonic problem, it follows that

$$\begin{aligned}
D\left\langle u_1^h - u_1, u_1^h - u_1\right\rangle &= \left\langle u_2^h - u_2, u_1^h - u_1\right\rangle + d(u_1^h, u_1^h - u_1) - c(u_2^h, u_1^h - u_1) \\
\left\|u_1 - u_1^h\right\|_{H_0^1}^2 &\leq Ch^{\frac{3}{2}}\|f\|_{L^2}^2 + Ch^\sigma\|f\|_{L^2}^2 + Ch^{\sigma+1}\|f\|_{L^2}^2 \,,
\end{aligned}$$

where we applied the estimates from Lemma 5.3, the fact that the solution of the discrete problem is bounded, and going through the Ritz approximation as an intermediate. Since we assumed that $\sigma \geq \frac{3}{2}$, this proves the corollary. □

5.2.5.3 *Convergence in the absence of a boundary*

Here we provide the proof for the case of empty boundary, which is much simpler than the case of a nonempty boundary. If there is no boundary, the mixed formulation decouples, as $H_0^1 = H^1$ and $S_{h,0} = S_h$. In this case, we obtain the better convergence rate:

Theorem 5.7. *It holds that*

$$\left\|u_2^h - u_2\right\|_{L^2} + h\left\|u_2^h - u_2\right\|_{H^1} \leq Ch^2\|f\|_{L^2}$$
$$\left\|u_1^h - u_1\right\|_{L^2} + h\left\|u_1^h - u_1\right\|_{H^1} \leq Ch^2\|f\|_{L^2} .$$

Proof. Since $H_0^1 = H^1$ and $S_{h,0} = S_h$, equations (5.16) and (5.33) decouple into two Poisson equations. By Wardetzky [91, Theorem 3.3.3] it holds that

$$\left\|u_2^h - u_2\right\|_{L^2} + h\left\|u_2^h - u_2\right\|_{H^1} \leq Ch^2\|f\|_{L^2} .$$

We ensured that our solutions are modulo harmonic functions in Definition 5.2.

To bound the error in u_1^h we turn to the solution $v_1 \in S_{h,0}$ of the discrete Poisson problem

$$D\left\langle v_1^h, \eta\right\rangle_A = \langle u_2, \eta\rangle_A \qquad \forall \eta \in S_{h,0} . \tag{5.43}$$

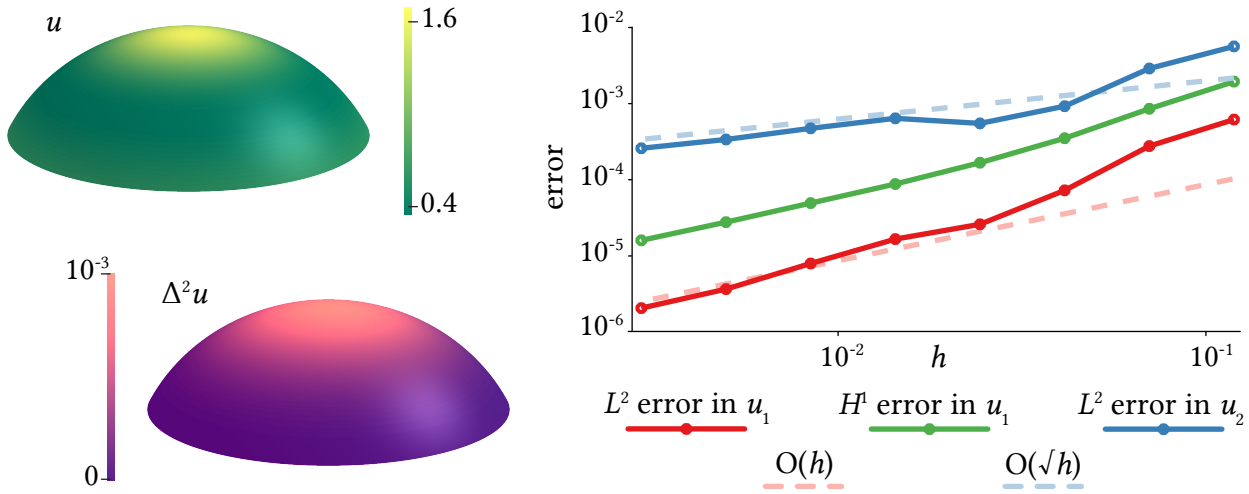


Figure 5.3. Solving the biharmonic equation on a spherical cap with known exact solution (loglog plot). The solution u as well as the right-hand side $\Delta^2 u$ are shown. We observe convergence of the solution with rates corresponding to Section 5.2.5.2.

As v_1^h is the solution to a discrete Poisson problem, we obtain

$$\left\|v_1^h - u_1\right\|_{L^2} + h\left\|v_1^h - u_1\right\|_{H^1} \leq Ch^2\|u_2\|_{L^2} \leq Ch^2\|f\|_{L^2} \,. \tag{5.44}$$

As for the error between u_1^h and v_1^h, we know that

$$\begin{aligned} \left\|u_1^h - v_1^h\right\|_{H_0^1}^2 &\leq C\left|\left\langle u_2^h - u_2, u_1^h - v_1^h\right\rangle_A\right| \leq C\left\|u_2^h - u_2\right\|_{L^2}\left\|u_1^h - v_1^h\right\|_{H_0^1} \,, \\ \left\|u_1^h - v_1^h\right\|_{H_0^1} &\leq C\left\|u_2^h - u_2\right\|_{L^2} \leq Ch^2\|f\|_{L^2} \,. \end{aligned} \tag{5.45}$$

Combining (5.44) and (5.45), and using the classical Aubin-Nitsche trick for the Poisson problem we obtain that

$$\left\|u_1^h - u_1\right\|_{L^2} + h\left\|u_1^h - u_1\right\|_{H^1} \leq Ch^2\|f\|_{L^2} \,,$$

which proves the theorem. □

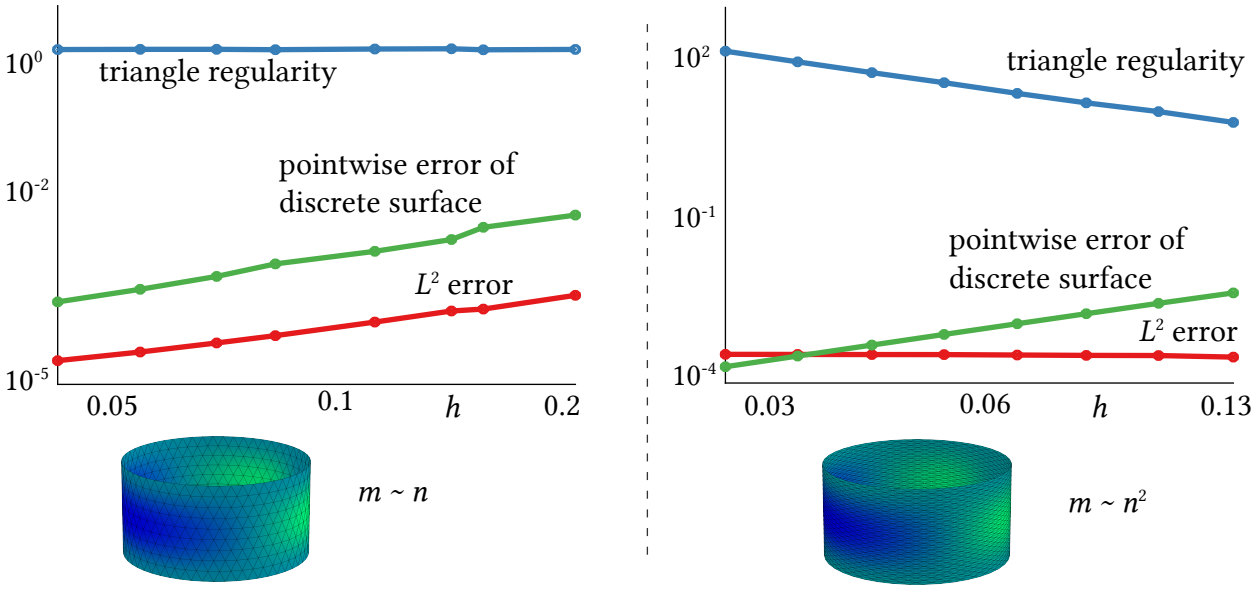


Figure 5.4. Solving the biharmonic equation on a Schwarz lantern with numerically computed high-resolution solution (loglog plot). Convergence depends on the proportion of the number of vertices along the equator m and the number of vertices along the axis of rotational symmetry n. While we observe convergence for $m \sim n$, we no longer observe it for $m \sim n^2$.

5.2.6 Experiments

We performed a variety of experiments using the mixed finite element method to solve the biharmonic equation on a curved surface approximated by a triangle mesh.

Figure 5.3 shows a boundary value problem solved on a spherical cap using the mixed finite element method. The observed convergence rates obey the theoretical convergence rates of Section 5.2.5: convergence of order h in the solution, convergence of order $h^{\frac{3}{4}}$ in the derivative of the solution, and convergence of order $h^{\frac{1}{2}}$ in the Laplacian of the solution.

In Figure 5.4 the importance of the triangle regularity condition can be seen. The standard Schwarz lantern fulfills conditions **(C1-C5)** if it fulfills the triangle regularity condition. This is

only the case if the ratio between the number of vertices along its equator m is proportional to the number of vertices along its axis of rotational symmetry n. If, on refinement, m increases much more quickly than n, such as when $m \sim n^2$, the mixed finite element method does not converge anymore, even though the discrete surface converges to the smooth surface pointwise.

Chapter 6: Discretizing the Hessian and Vector Dirichlet Energies

This section deals with the discretization of the new Hessian energy from Chapter 3. We first introduce a novel discretization of the Hessian energy based on linear mixed finite elements for triangle meshes in $\mathbb{R}^2$. Readers who are only interested in a general-purpose discretization of the Hessian energy can feel free to skip Section 6.1. We then discuss a simple discretization of the vector Dirichlet energy (from Section 2.2) for non-flat triangle meshes based on previous work, and use this discretization to introduce a discretization for the Hessian energy on curved surfaces $\Omega \subseteq \mathbb{R}^3$.

One commonality of all discretizations in this chapter is their simplicity. They can all be very easily built using basic sparse matrix multiplication as well as simple loops over all faces and edges of the triangle meshes. They also appear to converge in experiments, given sufficiently nice triangle meshes. The curved Hessian discretizations is also completely *intrinsic* to the meshes, meaning that it only depends on the edge lengths of the triangles. The discretization is invariant under isometry, just like the continuous energies it discretizes.

6.1 Discretizing the Hessian Energy in $\mathbb{R}^2$

For the entirety of Section 6.1, let $\Omega \subseteq \mathbb{R}^2$ be a flat surface. The fact that we are dealing with flat surfaces makes the discretization of the Hessian energy significantly simpler: the curvature term $\kappa|\mathrm{d}\, u|^2$ of Definition 3.1 is automatically zero, since there is no curvature. Additionally, the curved Hessian $\nabla\, \mathrm{d}\, u$ can be realized as the Hessian coordinate matrix $\begin{pmatrix} \partial_{xx} u & \partial_{xy} u \\ \partial_{yx} u & \partial_{yy} u \end{pmatrix}$, where each coordinate of the matrix is a separate linear function. We use this to derive a finite difference as well as a finite element discretization of the Hessian energy for flat surfaces.

6.1.1 Finite differences

The simplest way to discretize any differential energy is usually with finite differences. If the domain can be represented as a grid, all partial derivatives are easily turned into finite differences. We use standard central finite differences to compute the Hessian at all interior grid points [123].

Let the domain Ω be embedded in a grid with n total nodes and $\hat{n}$ interior nodes. The grid nodes are at equally spaced positions $x_{i,j}$. Let $\mathtt{u}_{i,j}$ be the discrete function at the grid node (i, j), approximating the smooth function $u(x)$ at that point. We define a sparse matrix $\mathbf{H} = \begin{bmatrix} \mathbf{H}_{xx}^\top \mathbf{H}_{xy}^\top \mathbf{H}_{yx}^\top \mathbf{H}_{yy}^\top \end{bmatrix}^\top \in \mathbb{R}^{4\hat{n} \times n}$ such that each row of $\mathbf{H}_{xx}$ approximates the second derivative of the unknown function at the corresponding interior grid node $x_{i,j}$: $\mathbf{H}_{xx}^{i,j}\mathtt{u} \approx \frac{\partial^2}{\partial x^2} u(x_{i,j})$, and analogously for rows in $\mathbf{H}_{xy}$ and $\mathbf{H}_{yy}$.

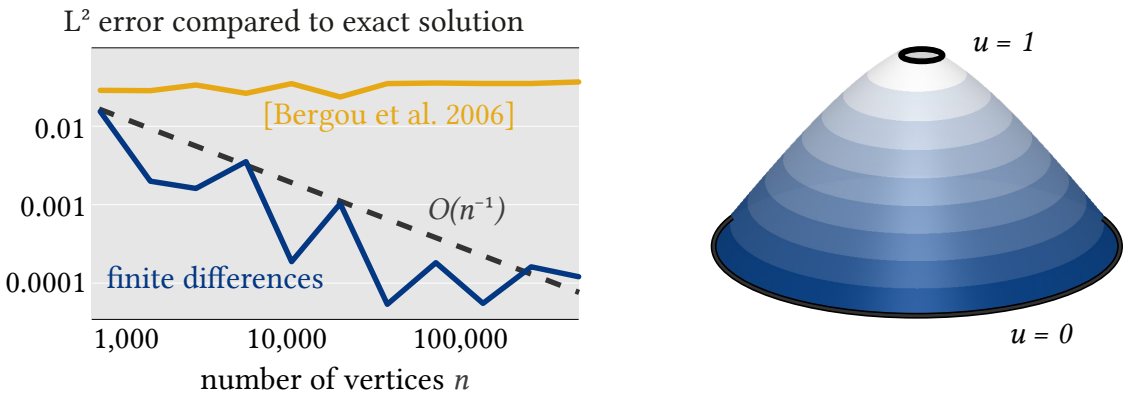


Figure 6.1. Minimizing the Hessian energy $E_{\mathbf{H}}$ on an annulus with the inner boundary set to one and the outer boundary set to zero. While our finite difference discretization converges under refinement, minimizing the discrete energy of Bergou *et al.* [56] does not.

We use standard, second-order stencils for each term, where, for a grid edge length h, the (i, j)-th entry of the discrete Hessian matrix is

$$\mathbf{H}^{i,j}_{xx}\mathtt{u} = \frac{\mathtt{u}_{i,j-1} - 2\mathtt{u}_{i,j} + \mathtt{u}_{i,j+1}}{h^2}$$
$$\mathbf{H}^{i,j}_{yy}\mathtt{u} = \frac{\mathtt{u}_{i-1,j} - 2\mathtt{u}_{i,j} + \mathtt{u}_{i+1,j}}{h^2}$$
$$\mathbf{H}^{i,j}_{xy}\mathtt{u} = \mathbf{H}^{i,j}_{yx}\mathtt{u} = \frac{\mathtt{u}_{i-1,j-1} + \mathtt{u}_{i-1,j+1} - \mathtt{u}_{i+1,j-1} + \mathtt{u}_{i+1,j+1}}{4h^2}.$$

This discretization of the Hessian matrix can now be plugged into the squared Hessian energy from Definition 3.1 for a complete discretization of the continuous Hessian energy.

This discretization appears to converge in experiments on the order of the inverse of the number of vertices, which also corresponds to the edge length (see Figure 6.1). The finite difference discretization can be used for smoothing (Figure 3.3) as well as data interpolation (Figure 3.2), where it achieves the effects discussed in Chapter 3: minimizers are not as biased by the boundary as they are with other discretization methods.

Our finite difference discretization matches discretizations used in image processing on rectangular domains [64], where it is not usually used for non-rectangular domains. For non-convex, non-rectangular irregular domains no special treatment is needed beyond ensuring that all interior nodes have neighboring nodes in all eight directions and are not isolated.

6.1.2 Mixed finite elements

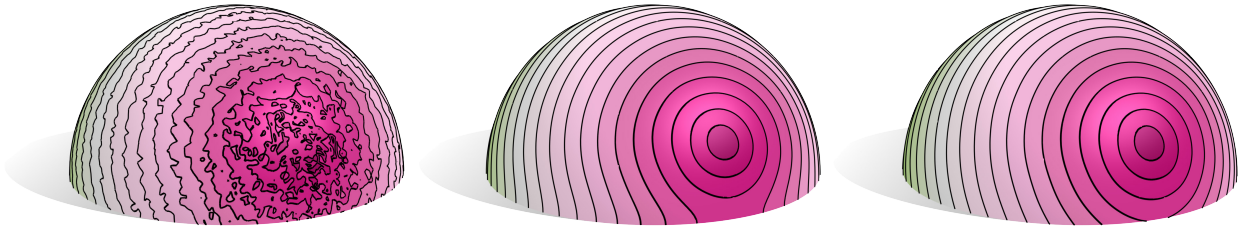
The main problem with finite differences is the fact that the domain has to be discretized as a grid. More flexibility can be achieved with triangle meshes, which are, for that reason, very popular in geometry processing. On triangle meshes, finite difference discretizations are not as straightforward. Because of this, we resort to a novel *mixed finite element* discretization with hat functions, similar to the discretization of the Laplacian energy discussed in Section 5.1.3.

The Hessian energy for flat surfaces (disregarding the right-hand side f), is given by

$$E_{\mathbf{H}}(u) = \int_\Omega \frac{1}{2} H_u \cdot H_u \, \mathrm{d}x \,,$$

where H_u is the Hessian matrix of u and $A \cdot B = \operatorname{tr}(A^\top B)$ is the matrix dot product. This can be turned into a mixed finite element discretization using Lagrange multipliers similar to the Laplacian energy with zero Neumann boundary conditions from (5.9) by replacing H_u with a new, matrix-valued variable V and enforcing $H_u = V$. However, unlike in (5.9), we are careful to use a matrix-valued Lagrange multiplier Λ that evaluates to zero on the boundary, in order to not interfere with the Hessian energy's natural boundary conditions,

$$\operatorname*{saddle}_{u,V,\Lambda} \int_\Omega \frac{1}{2} V \cdot V + \Lambda \cdot (H_u - V) \, \mathrm{d}x \,. \tag{6.1}$$



noisy function on hemisphere

Smoothing with E_Δ and zero Neumann

Smoothing with a 3D extension of E_H flat mixed FEM

Figure 6.2. Smoothing a noisy function (*left*) with the Laplacian energy and zero Neumann boundary conditions introduces a bias at the boundary (*center*), where the function's isolines have to be perpendicular to the boundary. The extension of the flat mixed FEM discretization of the Hessian energy to $\mathbb{R}^3$ (*right*) is less biased.

where V, Λ are matrix-valued functions. Written in components, we can see that $\Lambda \cdot H_u = \sum_{i,j} \Lambda_{ij} \partial_i \partial_j u$. By integration by parts, since Λ evaluates to zero at the boundary, $\int_\Omega \Lambda \cdot H_u \, \mathrm{d}x = -\int_\Omega \sum_{ij} \partial_i \Lambda_{ij} \partial_j u \, \mathrm{d}x = -\int_\Omega (\nabla \cdot \Lambda) \cdot \nabla u \, \mathrm{d}x$, where $\nabla \cdot \Lambda$ is a matrix divergence that acts on the matrix from the left. Using this, we can rewrite the saddle problem from (6.1) as

$$\operatorname*{saddle}_{u,V,\Lambda} \int_\Omega \left(\frac{1}{2} V \cdot V - (\nabla \cdot \Lambda) \cdot \nabla u - \Lambda \cdot V \right) \mathrm{d}x \, .$$

By varying V while keeping Λ, u constant, we see that $V = \Lambda$ (and thus V must be zero at the boundary). The saddle problem becomes

$$\operatorname*{saddle}_{u,V,\Lambda} \int_\Omega \left(-\frac{1}{2} V \cdot V - (\nabla \cdot V) \cdot \nabla u \right) \mathrm{d}x \, . \tag{6.2}$$

We now discretize the expression from (6.2) using the usual operators for hat functions from Section 5.1.2, while treating each vector as two different hat functions (the two coordinate entries) and each matrix as four different hat functions (the four coordinate entries). Let m be the number

noisy function | Denoising with E_Δ and zero Neumann | Denoising with a 3D extension of E_H flat mixed FEM

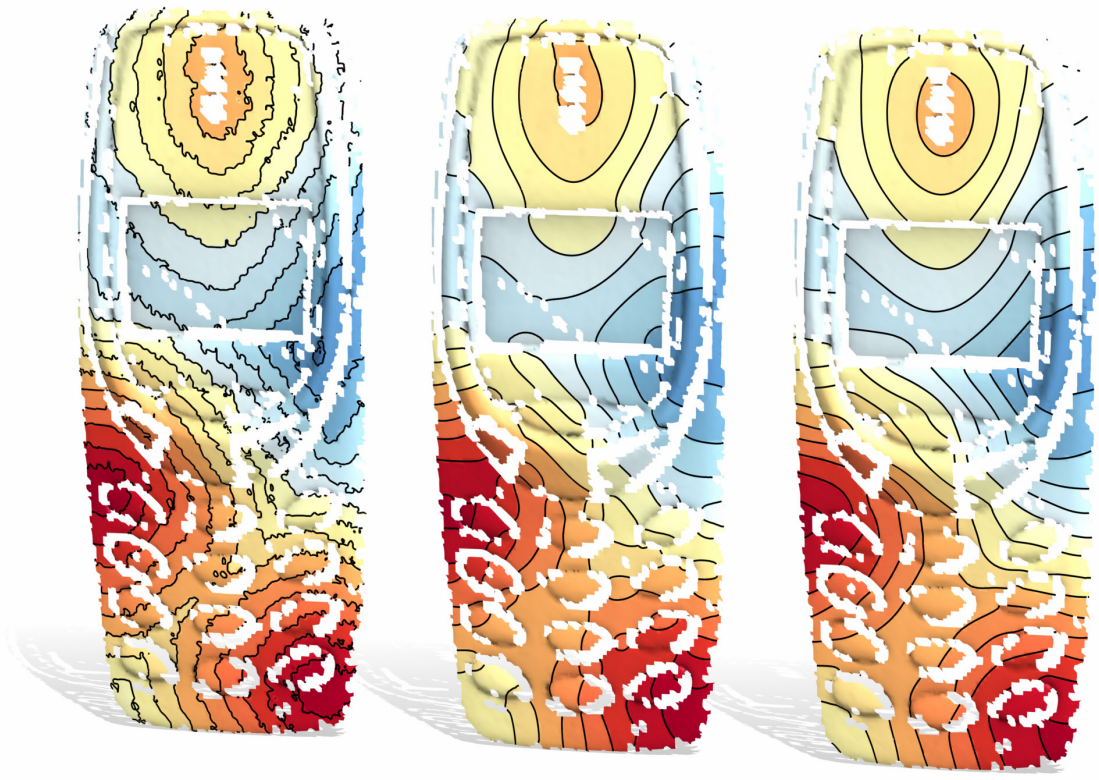
Figure 6.3. Denoising a noisy function (*left*) with the Laplacian energy with zero Neumann boundary conditions leads to bias at the boundary. Our simple extension of the mixed FEM discretization for the Hessian energy on flat surfaces to $\mathbb{R}^3$ combats this bias.

of faces, n the number of vertices, and $\hat{n}$ the number of interior vertices (vertices that are not on the boundary). The discrete operators we need are:

- The discrete gradient $\mathbf{G}$, a $2m \times n$ matrix that maps every per-vertex hat function to two per-face constant gradient functions (one for each coordinate).

- The discrete matrix divergence $\mathbf{D}$, a $2m \times 4\hat{n}$ matrix that maps a discrete matrix field composed of four hat functions to two constant per-face functions, $\mathbf{D} = \begin{pmatrix} \mathrm{G}(x,i) & \mathrm{G}(y,i) & 0 & 0 \\ 0 & 0 & \mathrm{G}(x,i) & \mathrm{G}(y,i) \end{pmatrix}$. $\mathbf{G}(x, i)$ is the discrete gradient matrix, but only the rows corresponding to the x-coordinate of the gradient and the columns corresponding to interior vertices.

- The face area matrix $\mathbf{A}$, a $2m \times 2m$ diagonal matrix that contains all face areas, twice, on the diagonal. This matrix is used to integrate per-face functions.



noisy function

E_Δ & zero Neumann

3D extension of $E_{\mathbf{H}}$ flat mixed FEM

Figure 6.4. Smoothing a function (*left*) on a noisy domain, where many of the boundaries are not true physical boundaries, but simply caused by surface scanning noise. Minimizers of the Laplacian energy E_Δ show bias at the many holes inside the domain, while our extension of the $E_{\mathbf{H}}$ flat mixed FEM discretization is much more robust to the extraneous holes.

Interpolating in $\mathbb{R}^3$ naive extension of E_{H} flat mixed FEM

Figure 6.5. Interpolating scattered data on a surface by simply interpolating in $\mathbb{R}^3$ (*left*) ignores the geometry of the surface, and might sound like a good strategy to avoid boundary bias. This approach, however, leads to massive overshooting as the intrinsic geometry is ignored. Our extension of the flat mixed FEM discretization of $E_{\mathbf{H}}$ to $\mathbb{R}^3$ (*right*) is shape-aware and does not overshoot.

- The modified mass matrix $\hat{\mathbf{M}}$, a $4\hat{n} \times 4\hat{n}$ matrix that repeats the hat function lumped mass matrix for interior vertices, $\mathbf{M}(i,i)$, four times across its diagonal. It is used to integrate matrix-valued functions with one hat function per coordinate.

Discretized with our new operators, (6.2) becomes, for a vector of discrete hat function degrees of freedom $\mathtt{u}$ and a vector of discrete matrix hat function degrees of freedom $\mathtt{V}$,

$$\operatorname*{saddle}_{\mathtt{u},\mathtt{V}} \left(-\frac{1}{2}\mathtt{V}^\top \hat{\mathbf{M}} \mathtt{V} - \mathtt{V}^\top \mathbf{D}^\top \mathbf{A}\mathbf{G}\mathtt{u} \right) .$$

By differentiating with respect to $\mathtt{V}$, we see that $\hat{\mathbf{M}}\mathtt{V} = -\mathbf{D}^\top \mathbf{A}\mathbf{G}\mathtt{u}$. Thus, the discrete Hessian energy for flat surfaces can be written as

$$\boxed{E_{\mathbf{H}}(u) \approx \mathtt{u}^\top \mathbf{G}^\top \mathbf{A}\mathbf{D}\hat{\mathbf{M}}\mathbf{D}^\top \mathbf{A}\mathbf{G}\mathtt{u} \,,} \tag{6.3}$$

and minimizing the discrete Hessian energy is solving $\operatorname{argmin}_{\mathtt{u}} \mathtt{u}^\top \mathbf{G}^\top \mathbf{A}\mathbf{D}\hat{\mathbf{M}}^{-1}\mathbf{D}^\top \mathbf{A}\mathbf{G}\mathtt{u}$.

Figure 6.6. Computing the eigenvector to the first nonzero eigenvalue of E_Δ (*left*) and two discrete Hessian energies. As there is no boundary, they should all agree. The naive extension of flat mixed FEM (*center*) does not produce the correct result, while the more principled discretization of Section 6.3 (*right*) does.

This flat mixed finite element discretization of the Hessian energy can now be used to compute and optimize the Hessian energy on any flat triangle mesh. It can be applied to smoothing, scattered data interpolation (Figure 3.9, *left*), and character animation (Figure 3.10). It exhibits the properties emphasized in Chapter 3: minimizers of this discrete flat Hessian are less biased by the boundary than minimizers of the Laplacian energy with zero Neumann boundary conditions. Unlike the finite difference discretization, this discretization does not force us into using domains that can be represented as grids; triangle meshes have more flexibility.

Extension to $\mathbb{R}^3$. The mixed finite element Hessian for flat triangle meshes from (6.3) can be extended to apply to triangle meshes in $\mathbb{R}^3$ with only very minor modifications. The gradient of a hat functions is now a constant vector with three coordinates, x, y, z. Thus, modifying the discrete gradient matrix $\mathbf{G}$ (and, by extension, also the matrices $\mathbf{D}$, $\mathbf{A}$, $\hat{\mathbf{M}}$), results in the temporary treatment of the discrete Hessian as a 3×3 matrix.

This simple modification allows us to use this discretization to compute an energy whose minimizers are less biased by the boundary than the minimizers of the Laplacian energy with zero Neumann boundary condition on non-flat surfaces. The extension is used for smoothing in Figures 6.2, 6.3, and 6.4, where we can clearly see that this extension reduces bias at the boundary compared to other smoothing methods. It can also be used for scattered data interpolation (Figure 6.5), where it performs better than simply ignoring the surface and interpolating in all of $\mathbb{R}^3$, as the method takes the geometry of the surface into account.

It is of utmost important to note that this extension of the flat method to $\mathbb{R}^3$ *does not* discretize the Hessian energy. While its minimizers are less biased by the boundary than minimizers of the Laplacian energy with zero Neumann boundary conditions, they do not minimize the actual Hessian energy $E_{\mathbf{H}}$ from Definition 3.1 on curved surfaces, as can be seen in Figure 6.6. For that, we need a principled discretization of the Hessian energy for curved surfaces, which is the subject of Section 6.3.

6.1.3 Other discretizations and comparisons

Section 3.5 discusses previous work in in geometry processing that deals with discretizing Hessian-like energies. This section compares some of these discretizations to our flat mixed FEM discretization of $E_{\mathbf{H}}$.

The null space of the continuous Hessian energy $E_{\mathbf{H}}$, as well as our flat mixed FEM discretization (see Figure 3.6) contains only affine functions. Wang *et al.* [42] state that this is exactly the condition needed of a smoothness energy to be used in linear subspace design for shape deformation. With this property in mind, they and other previous works have designed *discrete* energies by modifying the cotangent Laplacian for a triangle mesh from (5.4). These discrete modifications have no known smooth analogs, and as such are more challenging to understand and analyze [124].

Discrete exterior calculus. Fisher *et al.* [55] construct a discrete energy for tangent vector field design using discrete exterior calculus (DEC) on triangle meshes. Their energy measures the sum of all squared discrete divergences of a given vector field v via

$$\mathrm{v}^\top \star_1 \mathrm{d}_0 \star_0^{-1} \underbrace{\mathrm{d}_0^\top \star_1}_{\mathbf{F}} \mathrm{v} \,,$$

where, in this equation, $\star_0, \star_1, \mathrm{d}_0$ are discrete operators for the zero-form and one-form Hodge star and the exterior derivative respectively, followed by specific adjustments to the matrix $\mathbf{F}$ when building the operator for free boundaries. Replacing the vector field v with the differential

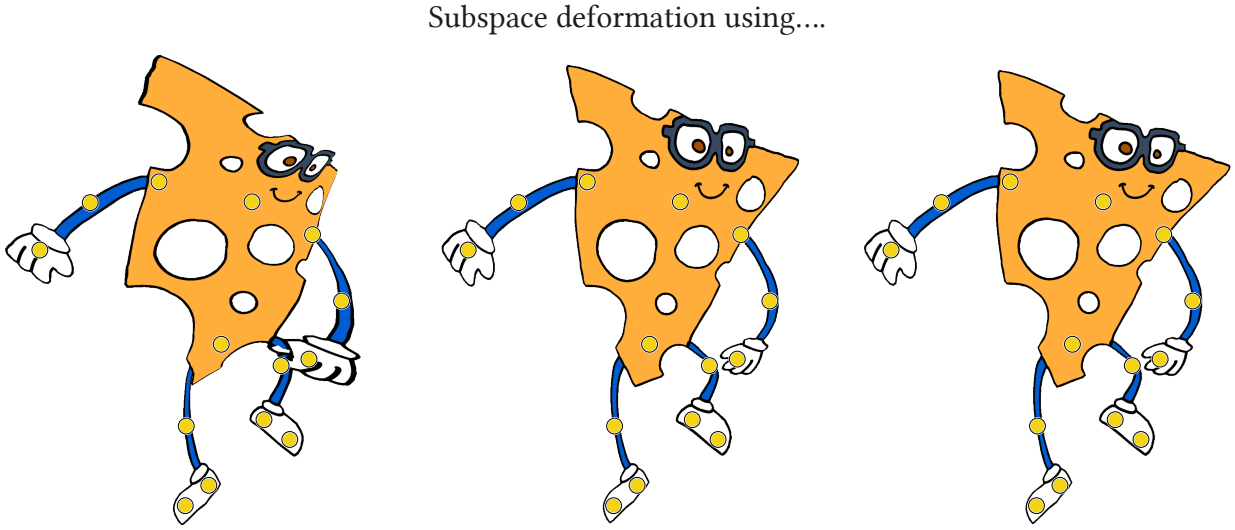


Figure 6.7. The smoothness energy built using (6.4) is not suitable for subspace design (*left*). In contrast, the energy from (6.5) (*center*) and our flat mixed FEM discretization of the Hessian energy (*right*) give good results.

of an unknown scalar field $\mathtt{u}$, we can arrive at a discrete smoothness energy of a scalar field,

$$\mathtt{u}^\top \mathrm{d}_0^\top \mathbf{F}^\top \underbrace{\star_0^{-1}}_{\mathbf{M}^{-1}} \underbrace{\mathbf{F}\,\mathrm{d}_0}_{\mathbf{K}=\mathbf{L}+\mathbf{N}} \mathtt{u}\,, \tag{6.4}$$

where $\mathbf{M}$ is the lumped hat function mass matrix, and $\mathbf{K}$ is a non-symmetric matrix which matches the discrete operator of Wang *et al.* [42] which they construct by adding the discrete normal derivative operator $\mathbf{N}$ to the cotangent Laplacian $\mathbf{L}$.

This matrix $\mathbf{K}$ has been shown to contain at least all affine functions [125]. Unfortunately, for some meshes this matrix (and thus the discrete energy it defines) contains other, non-affine functions in its null space (see inset). From the point of view of linear subspace design, this null space is too big, and it leads to artifacts when applied to surface deformation (see Figure 6.7).


Nonlinear null vector of DEC energy

This does not appear to be a simple matter of mesh resolution or floating point error. Following the DEC construction, taking the divergence of the gradient should correspond to the smooth Laplace operator. If the discrete energy from (6.4) preserved this structure, one would expect it to produce the discrete natural boundary conditions for the Laplacian energy E_Δ and match its null space. However, looking at its eigenspace, the energy from (6.4) has a small (albeit > 0) number of zero eigenvalues, which is too few to correspond to the natural boundary conditions of E_Δ.

Crouzeix-Raviart finite elements. Wang *et al.* [42] use an alternative construction to create the results displayed in their article [124]. Discrete normal derivatives are added to the Crouzeix-Raviart Laplacian $\mathbf{L}$, giving

$$\mathtt{u}^\top \mathbf{E}^\top \mathbf{K}^\top \mathbf{M}^{-1} \underbrace{\mathbf{K}}_{\mathbf{L}+\mathbf{N}} \mathbf{E}\mathtt{u}\,, \tag{6.5}$$

where $\mathbf{E}$ averages vertices onto incident edges, $\mathbf{M}$ is the Crouzeix-Raviart mass matrix, and $\mathbf{N}$ computes discrete normal derivatives at boundary edges.

The energy from (6.5) corresponds to one of two discrete energies proposed by Bergou *et al.* [56] to model plate bending, assuming isometric deformation as the Laplacian energy E_Δ of the displacement coordinate functions. The authors do not discuss how their two discrete energies differ (the other one corresponding to E_Δ with zero Neumann boundary conditions). Perhaps surprisingly, the energy from (6.5) behaves very similarly to $E_\mathbf{H}$, despite its motivation by both Bergou *et al.* [56] and Wang *et al.* [42] as a discrete modification to E_Δ. In contrast to the DEC energy of (6.4), empirically its null space does not contain spurious functions, and the resulting subspace deformations are adequate (see Figure 6.7).

However, while the energy from (6.5) is suitable for some applications, it does not seem to discretize $E_{\mathbf{H}}$ with its natural boundary conditions (see Figure 6.1). It remains unclear whether the discrete energy from (6.5) converges to some other continuous energy.

6.2 Discretizing the Vector Dirichlet Energy

The Hessian energy from Definition 3.1 on curved surfaces contains the vector Dirichlet energy from Definition 2.7 on curved surfaces (disregarding the right-hand side $\mathbf{f}$),

$$E_{\nabla}(\mathbf{u}) = \frac{1}{2} \int_{\Omega} \|\nabla \mathbf{u}\|^2 \, \mathrm{d}x \, ,$$

where $\mathbf{u}$ is a smooth vector field. The vector Dirichlet energy E_{∇}, however, is an interesting subject of study in its own right, as discussed in Section 2.2.

We employ a discretization which is functionally the same as the previous work of Djerbetian [126]. It will be used in the discretization of the Hessian energy on curved surfaces. Our discretization is based on the Crouzeix-Raviart finite elements from Section 5.1.

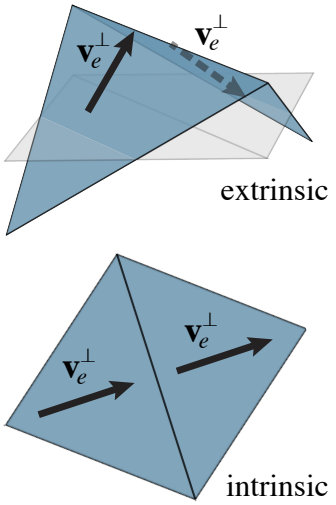
6.2.1 Vector Crouzeix-Raviart finite elements

We construct a *vector* Crouzeix-Raviart element using the scalar Crouzeix-Raviart element from Section 5.1. We start by characterizing the space of all tangent vectors at the midpoint of an edge e. Each tangent vector there can be written as a linear combination of the following two vectors,

- $\mathbf{v}_e^{\parallel}$, the unit vector parallel to e that points in the same direction as the oriented edge e;

- $\mathbf{v}_e^\perp$, the unit vector normal to e that corresponds to $\mathbf{v}_e^\parallel$ rotated by $\pi/2$ in the tangent space.

At first glance, $\mathbf{v}_e^\perp$ seems to be ambiguously defined, since there is no unique normal at the midpoint of e to rotate around. However, *intrinsically*, the tangent space of the mesh viewed as a polyhedron is well-defined away from vertices [91]. A pair of triangles on its own is always intrinsically flat. This means that if, at each triangle adjacent to e, we rotate $\mathbf{v}_e^\parallel$ around the respective triangle normal, we get to different extrinsic representations of $\mathbf{v}_e^\perp$ in each triangle, each corresponding to the same intrinsic tangent vector (see inset). $\mathbf{v}_e^\parallel$ is well-defined at the midpoint of e in both adjacent triangles, as the edge e is contained in both triangles.



Let φ_e be the Crouzeix-Raviart basis function for the edge e. The vectors $\mathbf{v}_e^\parallel$ and $\mathbf{v}_e^\perp$, defined at the midpoint of e, can be easily extended along each of the two faces adjacent to e by transporting them along the flat triangles. This allows us to define two vector basis functions per edge:

$$\mathbf{w}_e^\parallel = \varphi_e \mathbf{v}_e^\parallel$$
$$\mathbf{w}_e^\perp = \varphi_e \mathbf{v}_e^\perp \, .$$

The $\mathbf{w}_e^\parallel, \mathbf{w}_e^\perp$ for every edge e in the triangle mesh form the basis of our discrete vector field space.

We now use the $\mathbf{w}_e^\parallel, \mathbf{w}_e^\perp$ to compute a discretization of the vector Dirichlet operator from Definition 2.7 by plugging each pair of basis functions into the definition of the energy and integrating over triangles. The resulting sparse symmetric matrix $\mathbf{L}$ has two degrees of freedom (DOFs) for each edge e of the mesh, which we will denote by the indices $e^\parallel$ and $e^\perp$. For a discrete vector $\mathtt{u}$ representing all the DOFs of a Crouzeix-Raviart vector field $\mathbf{u}$, $\mathtt{u}^\top \mathbf{L} \mathtt{u}$ discretizes $E_\nabla(\mathbf{u})$. The matrix $\mathbf{L}$ is constructed using the following formula, which is looped over all faces f in the mesh,

and all pairs of consecutive edges $e_i, e_j \in f$,

$$\boxed{\begin{aligned} \mathbf{L}_{e_i^{\parallel},e_i^{\parallel}} \,,\, \mathbf{L}_{e_i^{\perp},e_i^{\perp}} \mathrel{+}= \frac{|e_i|^2}{|f|} \\ \mathbf{L}_{e_i^{\parallel},e_j^{\parallel}} \,,\, \mathbf{L}_{e_i^{\perp},e_j^{\perp}} = 2\, s_{ij} \cot\theta_{ij} \cos\theta_{ij} \\ \mathbf{L}_{e_i^{\parallel},e_j^{\perp}} \,,\, -\mathbf{L}_{e_i^{\perp},e_j^{\parallel}} = 2\, s_{ij} \cos\theta_{ij} \,, \end{aligned}} \tag{6.6}$$

where $|e_i|$ is the length of e_i, $|f|$ is the area of f, and θ_{ij} is the angle between e_i, e_j. s_{ij} is 1 if the local orientations of e_i, e_j within f both agree or disagree with the global orientations of e_i, e_j, and -1 if one of them disagrees (i.e., whether a halfedge in f has the same orientation as an arbitrarily chosen global orientation for every edge). We use the notation += to highlight that diagonal terms are visited twice when looping over all faces and edges, and both entries must be accumulated. The off-diagonal entries must be added in two places each, i.e., $\mathbf{L}_{\alpha,\beta} = \mathbf{L}_{\beta,\alpha}$.

By the same approach, the discretization of $\frac{1}{2}\int_\Omega \|\mathbf{u}\|^2 \, \mathrm{d}x$, the diagonal mass matrix, is given by

$$\boxed{\mathbf{M}_{e^{\parallel},e^{\parallel}} \,,\, \mathbf{M}_{e^{\perp},e^{\perp}} = \frac{A_e}{3} \,,} \tag{6.7}$$

where A_e is the sum of the areas of the two triangles adjacent to e.

6.2.2 Alternative discretizations

There are other conceivable choices for discretizing the vector Dirichlet energy, beyond the edge-based approach of Djerbetian [126] that we follow. These choices includes vertex-based basis

functions [25, 24, 28, 127, 26, 29], halfedge-based basis functions [27], and spectral approaches [30, 31, 32].

6.3 Discretizing the Hessian Energy on Curved Surfaces

In this section we present a novel discretization of the Hessian energy $E_{\mathbf{H}}$ from Definition 3.1 on non-flat triangle meshes approximating curved surfaces. We are using piecewise linear Lagrangian finite elements (from Section 5.1.2) and vector Crouzeix-Raviart finite elements (from Section 6.2.1) in our discretization.

6.3.1 Two-Step Discretization

Discretizing the Hessian energy $E_{\mathbf{H}}$ as written would require us to discretize functions that can be differentiated twice, which is not possible with piecewise linear Lagrangian hat functions or Crouzeix-Raviart functions. To avoid this problem, we use an approach that resembles the mixed finite element method from Section 5.1.3 (but does not do the integration by parts step). Let u be the function whose Hessian energy we want to compute. We introduce an intermediate one-form variable $w = \mathrm{d}\, u$ and formulate the problem of optimizing $E_{\mathbf{H}}$ as

$$\operatorname*{argmin}_{u} \frac{1}{2} \int_{\Omega} (\nabla w) \cdot (\nabla w) + \kappa \, |w|^2 \, \mathrm{d}x, \quad w = \mathrm{d}\, u \, .$$

Using Lagrange multipliers to enforce the constraint $w = \mathrm{d}\, u$, we can write the optimization problem as the saddle problem

$$\underset{u,w,\lambda}{\text{saddle}} \frac{1}{2} \int_\Omega (\nabla w) \cdot (\nabla w) + \kappa \, |w|^2 - \lambda \cdot (w - \mathrm{d}\, u) \, \mathrm{d}x \,, \tag{6.8}$$

where λ is a Lagrange multiplier one-form.

We discretize the space of scalar functions (u) using the standard hat functions from Section 5.1.2, and we write the DOF vector associated with u as $\mathtt{u}$. The space of one-forms w is discretized using Crouzeix-Raviart one-forms: these are almost exactly the same as the vector Couzeix-Raviart elements from Section 6.2.1, except that instead of considering parallel and normal vectors at each edge, we consider the one-forms that are dual to them. This gives us Crouzeix-Raviart one-forms. We write the DOF vector associated with v as $\mathtt{v}$.

Using these discretizations we can construct the one-form Dirichlet matrix

$$\mathbf{L}_{ij} = \int_\Omega (\nabla \eta_i) \cdot \left(\nabla \eta_j\right) \, \mathrm{d}x \,,$$

for Crouzeix-Raviart one-form basis functions η_i, the differential matrix

$$\mathbf{D}_{ij} = \int_\Omega \eta_i \cdot \mathrm{d}\, \varphi_j \, \mathrm{d}x \,,$$

for hat function basis functions φ_i, the mass matrix

$$\mathbf{M}_{ij} = \int_\Omega \eta_i \cdot \eta_j \, \mathrm{d}x \,,$$

and the curvature matrix

$$\mathbf{K}_{ij} = \int_\Omega \kappa\, \eta_i \cdot \eta_j \, \mathrm{d}x \ .$$

Using these matrices, we write the discrete version of (6.8) as the problem of finding a critical point of the expression

$$\frac{1}{2}\mathtt{v}^\top(\mathbf{L}+\mathbf{K})\mathtt{v} - \lambda^\top(\mathbf{M}\mathtt{v} - \mathbf{D}\mathtt{u}) \ ,$$

for hat function DOF vector $\mathtt{u}$ and Crouzeix-Raviart one form DOF vectors $\mathtt{v}, \lambda$. Differentiating with respect to λ we see that $\mathbf{M}\mathtt{v} = \mathbf{D}\mathtt{u}$. As $\mathbf{M}$ is invertible, we arrive at the discretization

$$\boxed{E_{\mathbf{H}}(u) \approx \mathtt{u}^\top \mathbf{D}^\top \mathbf{M}^{-1}(\mathbf{L}+\mathbf{K})\mathbf{M}^{-1}\mathbf{D}\mathtt{u} \ ,} \tag{6.9}$$

and minimizing the discrete Hessian energy is solving $\operatorname{argmin}_{\mathtt{u}} \mathtt{u}^\top \mathbf{D}^\top \mathbf{M}^{-1}(\mathbf{L}+\mathbf{K})\mathbf{M}^{-1}\mathbf{D}\mathtt{u}$.

The Dirichlet matrix $\mathbf{L}$ has the same entries as the vector Dirichlet energy matrix from (6.6). The mass matrix $\mathbf{M}$ has the same entries as the vector Crouzeix-Raviart mass matrix from (6.7). The differential matrix $\mathbf{D}$ can be straightforwardly discretized by plugging in the respective basis functions and integrating. It is constructed by looping over every face f and every edge $e \in f$, and accumulating via the formula

$$\begin{aligned}
-D_{i,e^\parallel}, D_{j,e^\parallel} \ &+= s_e \frac{|f|}{3 l_{ij}} \\
D_{k,e^\parallel} \ &= 0 \\
D_{i,e^\perp} \ &+= -s_e \frac{l_{jk}}{6} \cos\theta_j^f \\
D_{j,e^\perp} \ &+= -s_e \frac{l_{ki}}{6} \cos\theta_i^f \\
D_{k,e^\perp} \ &= \frac{l_{ij}}{6} s_e \ ,
\end{aligned} \tag{6.10}$$

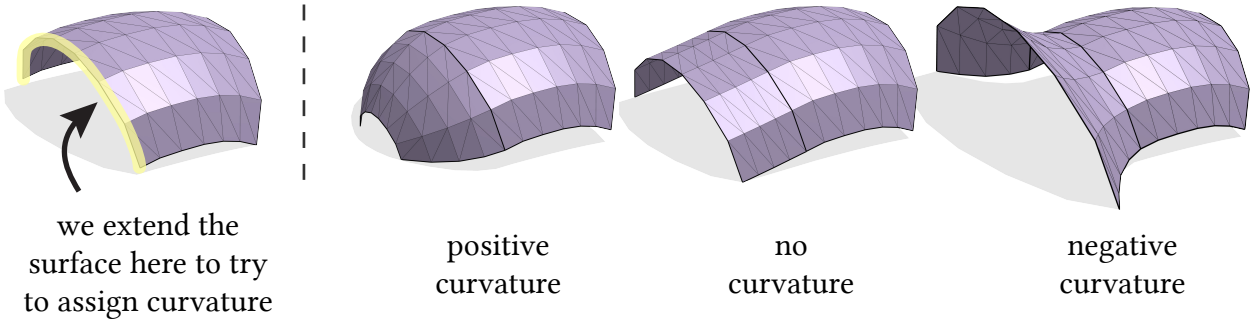


Figure 6.8. For the boundary of a continuous, piecewise linear surface (*far left*) there is no way to uniquely assign curvature at the boundary. The surface can be extended in many different ways that yield different curvatures at the boundary; examples leading to positive (*center left*), no (*center right*), and negative (*far right*) are shown.

where i is the vertex at the tail of the edge e, j is at its tip, and k is the third vertex. θ_i^f is the angle at the vertex i in the face f, $|f|$ is the area of the face f, and l_{ij} is the length of the edge from i to j. If one of the edges has reversed orientation in the face f with respect to its global orientation, s_e is -1, otherwise it is 1. The notation += signifies that certain edge-vertex combinations are visited twice in the loop over all faces, and the contributions of the different faces must be accumulated.

6.3.2 Discretizing the Curvature Term

It remains to discretize the curvature matrix $\mathbf{K}$. Special care needs to be applied when computing its entries via integration of the basis functions over triangles. The Gaussian curvature κ would appear, at first, to be 0 for every point in the triangle (after all, they are all intrinsically flat). But actually, the Gaussian curvature of a polyhedron is entirely concentrated on its vertices (and zero anywhere else). The integrated Gaussian curvature at a vertex is also known as the angle defect

$$\kappa_v \coloneqq 2\pi - \sum_{f \in N(v)} \theta_v^f \,, \tag{6.11}$$

where the sum is over all faces f in the set of faces containing the vertex v, and θ_v^f is the angle at vertex v in face f [128]. The idea of angle defects is very old: it goes back all the way to at least Descartes c. 1630, who showed that the sum of all angle defects of a polyhedron with spherical topology is 4π [129].

We thus interpret the Gaussian curvature of the polyhedron as a collection of Dirac delta functions at every vertex, i.e.,

$$\kappa = \sum_v \kappa_v \delta_v \ ,$$

where κ_v is the angle defect at the vertex v, and δ_v is the Dirac delta function. This means that the integral of κg, where g is a C^0 function, can be written as

$$\int_\Omega \kappa g \ \mathrm{d}x = \sum_v \kappa_v g(v) \ .$$

If the function g itself is only continuous within each triangle (like the Crouzeix-Raviart basis functions), then we need to distribute the contribution of each triangle accordingly. Let $c_{v,f} > 0$ for each vertex v and face f in the neighborhood of v be coefficients that average the contribution of each face at a vertex, i.e., the sum of the $c_{v,f}$ over all faces f adjacent to v is one. Then

$$\int_\Omega \kappa g \ \mathrm{d}x = \sum_v \kappa_v \sum_{f \in \mathcal{N}(v)} c_{v,f} g_f(v) \ ,$$

where g_f is the function g evaluated in the triangle f and $\mathcal{N}(v)$ is the set of faces adjacent to v. We choose to average by tip angle, which corresponds to an integral along a small circle around the vertex. We did not explore other reasonable choices, such as averaging by face area.

One remaining issue with the angle defect as Gaussian curvature is that the angle defect is not defined at boundary vertices. The problem stems from the fact that the notion of curvature at the

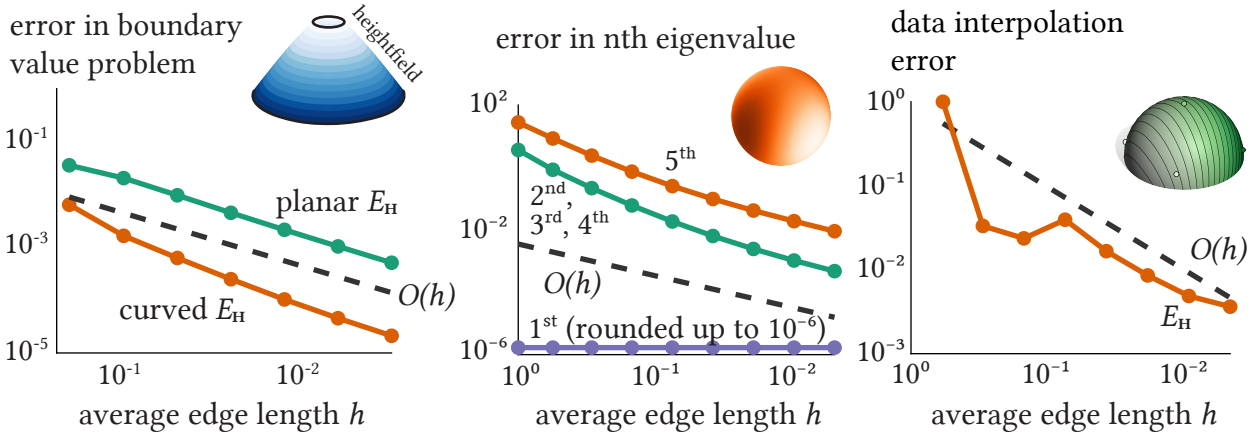


Figure 6.9. Convergence plots for three different problems, all errors are L^2 errors. Boundary value problem with known exact solution on a flat annulus mesh refined by loop subdivision with fixed smooth boundary; both our curved discretization as well as our planar discretization of $E_{\mathbf{H}}$ are shown (*left*). Error in calculating the lowest eigenvalues of the operator associated with $E_{\mathbf{H}}$ on the sphere with icosahedral meshing, with vertices of the mesh inscribed in the smooth limit sphere (*center*). Solving an interpolation problem and computing the error with respect to the highest-resolution solution, refined by loop subdivision with fixed z-coordinate at the boundary (*right*).

boundary of meshes (continuous, piecewise linear surfaces) is not in and of itself meaningful: by choosing to extend the surface in different ways at the boundary we can achieve any arbitrary Gaussian curvature, as can be seen in Figure 6.8. We choose to set the angle defect to 0 for all boundary vertices, thereby choosing the *most developable* (intrinsically linear) extension of all possible extensions. This fits in with our *as-linear-as-possible* boundary conditions, but differs from some conventions of angle defect at the boundary, which define it as the sum of tip angles subtracted from π (which is a discretization of geodesic curvature) [130].

With all of this in mind, the entries of $\mathbf{K}$ are as follows. Looping over all faces f in the mesh, and all pairs of consecutive edges $d, e \in f$ (such that e goes from vertex i to j and d goes from

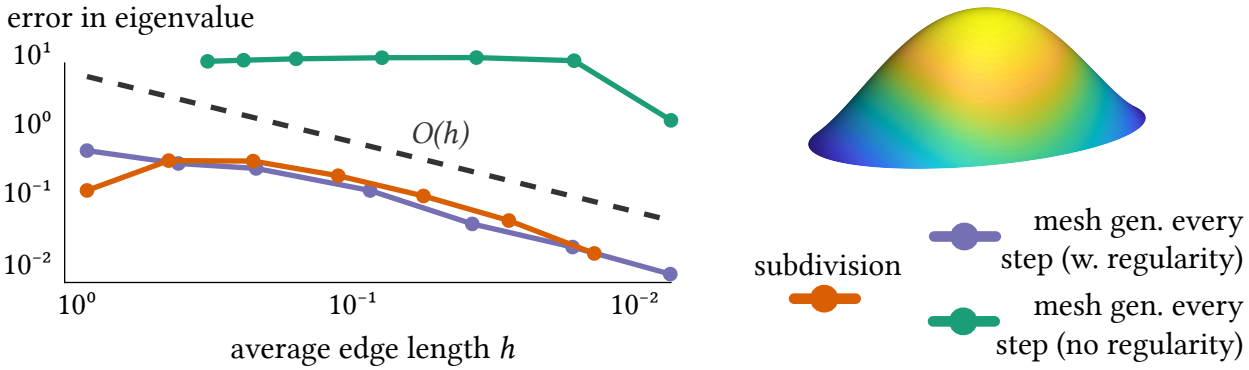


Figure 6.10. Computing the fourth eigenvalue of the Hessian energy $E_{\mathbf{H}}$ on an ellipse that was distorted in the third dimension. Both refinement through Loop subdivision and projection to a given smooth surface, as well as generating a planar mesh of the desired resolution with regular triangles at every step and then projecting to a given smooth surface show convergence to the highest resolution. For simple mesh generation without triangle regularity no convergence is observed.

vertex k to i),

$$
\begin{aligned}
K_{e^{\parallel},e^{\parallel}}, K_{e^{\perp},e^{\perp}} \mathrel{+}= \frac{\theta_i^f}{t_i}\kappa_i + \frac{\theta_j^f}{t_j}\kappa_j + \frac{\theta_k^f}{t_k}\kappa_k \\
K_{e^{\parallel},e^{\perp}} = 0 \\
K_{e^{\parallel},d^{\parallel}} = K_{e^{\perp},d^{\perp}} = s_{d,e}\cos\theta_i^f\left(\frac{\theta_j^f}{t_j}\kappa_j + \frac{\theta_k^f}{t_k}\kappa_k - \frac{\theta_i^f}{t_i}\kappa_i\right) \\
-K_{e^{\parallel},d^{\perp}} = K_{e^{\perp},d^{\parallel}} = s_{d,e}\sin\theta_i^f\left(\frac{\theta_j^f}{t_j}\kappa_j + \frac{\theta_k^f}{t_k}\kappa_k - \frac{\theta_i^f}{t_i}\kappa_i\right),
\end{aligned}
\tag{6.12}
$$

where κ_v is the angle defect at the vertex v and t_v is the angle sum at the vertex v. θ_i^f is the angle at the vertex i in the face f. s_{de} is 1 if the local orientations of d, e within f both agree or disagree with the global orientations of d, e, and -1 if one of them disagrees (i.e., whether a halfedge in f has the same orientation as an arbitrarily chosen global orientation for every edge). We use the notation += to highlight that diagonal terms are visited twice when looping over all faces

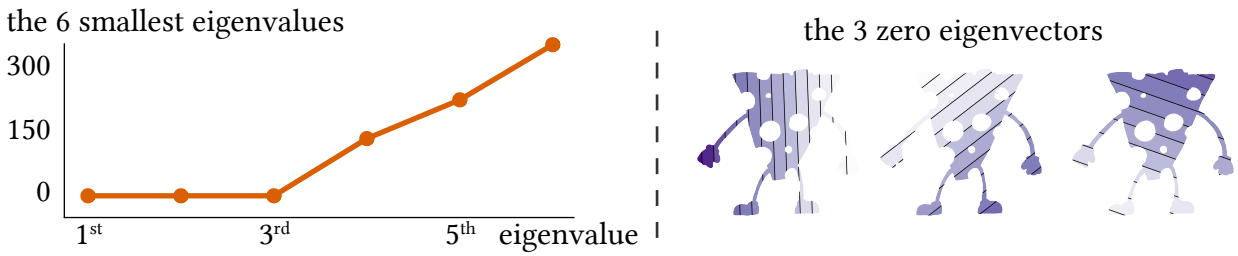


Figure 6.11. The six lowest eigenvalues of the Hessian energy discretized with our non-flat discretization on the cheeseman (*left*). As expected, there are only three zero eigenvalues. The three lowest eigenvectors (*right*) are the linear functions, which corresponds to the smooth Hessian energy.

and edges, and both entries must be accumulated. The off-diagonal entries must be added in two places each, i.e., $\mathbf{K}_{\alpha,\beta} = \mathbf{K}_{\beta,\alpha}$.

6.3.3 Stability and Convergence

Using our discretization of the Hessian energy for non-flat meshes from (6.9) to solve a variety of problems, we observe convergence on the order of the average edge length h (Figure 6.9). As can be seen in Figure 6.10, a successful strategy for obtaining convergence is making sure that the vertices are contained in a smooth surface, and then either refining the mesh through Loop subdivision [131] with a fixed smooth boundary, or generating meshes that fulfill the triangle regularity condition: the ratio of circumcircle to incircle of each triangle (the triangle regularity) is bounded from above and below independent of refinement level. This is reminiscent of the regularity conditions from Section 5.2. However, we do not have a proof of convergence for our method to confirm convergence, regularity condition, or convergence rate.

Figure 6.12 shows that for different triangle meshings of the same surface, very similar results are achieved, and the method is thus robust to remeshing.

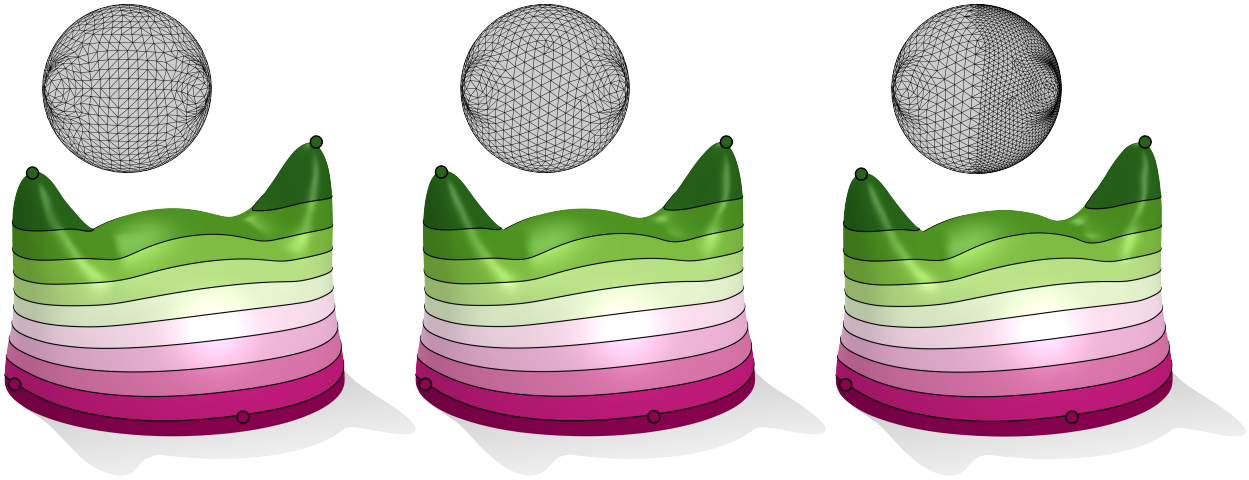


Figure 6.12. The same scattered data interpolation problem solved on different meshes for surfaces similar to the one from Figure 3.7 using the Hessian energy $E_{\mathbf{H}}$. The results are very similar. The wireframe shows each of the meshes before further refinement through loop subdivision with fixed boundary.

Additional convergence experiments can be found in Figure 6.13, which features a series of convergence experiments that shows the convergence of a boundary value problem on a variety of meshes to the highest-resolution solutions, as well as Figure 6.14, where a series of forward problems is solved, in which the Hessian energy of a function is measured on a curved surface, and because both the function and the surface embedding are known, the exact solution is also known and used to measure the error.

We do not have any theoretical guarantees concerning the spectrum of our discrete operator. In practice, however, our method correctly reproduces the first eigenvector of the Laplacian energy on closed surfaces (Figure 6.6, *right*), and our discrete Hessian energy seems to have the correct number of zero eigenvalues and their associated eigenvectors (Figure 6.11).

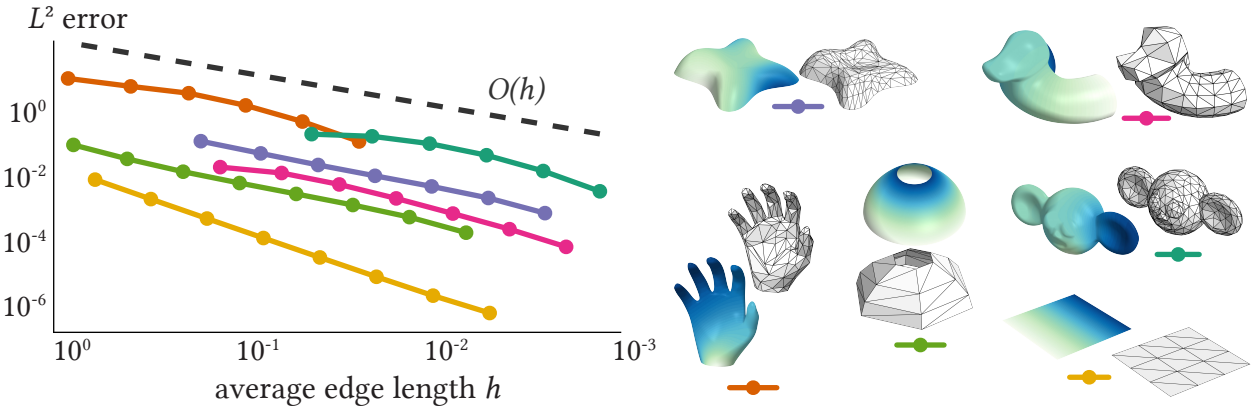


Figure 6.13. Error plot for six different boundary value problems. The minimizer of the Hessian energy $E_{\mathbf{H}}$ discretized with our non-flat discretization is compared to a high-resolution solution with the same discretization. Refinement happens via loop subdivision with various types of fixed boundary. The high-resolution solution as well as the wireframe of the lowest-resolution mesh are displayed for each problem.

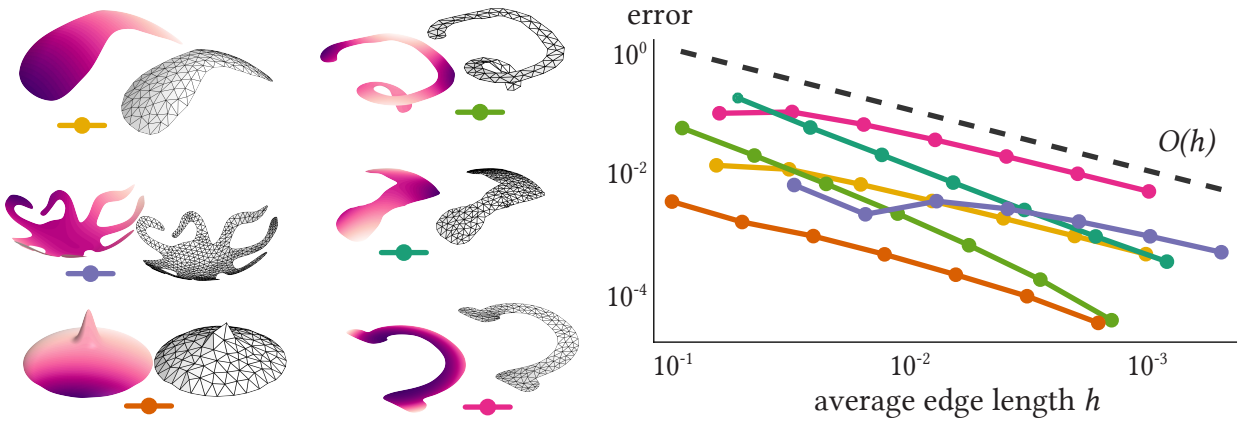


Figure 6.14. The domains are curved surfaces of the form $(x, y, z(x, y)) \in \mathbb{R}^3$, so the integrand of the Hessian energy can be exactly computed pointwise using the properties of Monge patches [132]. Quadrature is then used to compute the exact value of $E_{\mathbf{H}}(f)$. The high-resolution function f as well as the wireframe of the lowest-resolution mesh are displayed for each problem. Refinement happens via loop subdivision, and then projection to the given smooth surface.

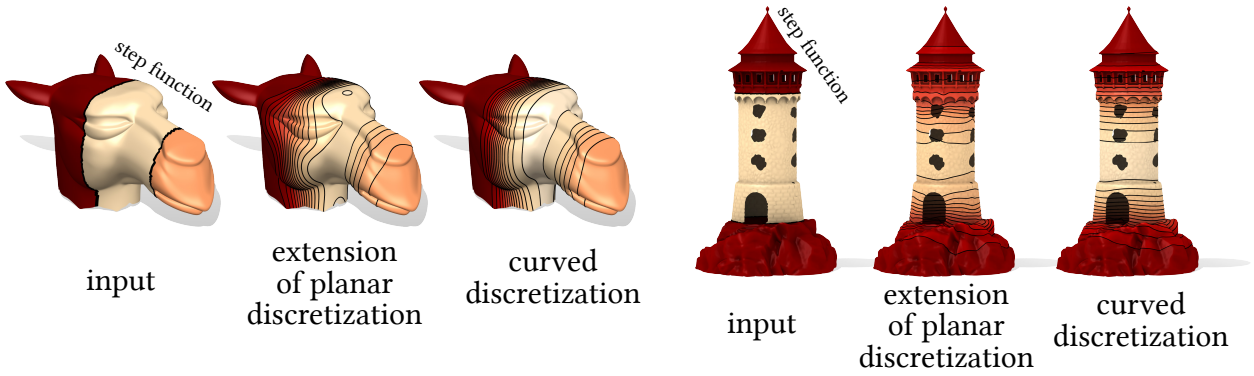


Figure 6.15. Smoothing a step function on a surface using the naive extension of the flat discretization, as well as our curved discretization. The extension of the flat discretization does not correctly account for the curvature of the surface, leading to crooked isolines. The curved discretization correctly accounts for curvature and does not suffer from such problems.

6.3.4 Applications

We implement the optimization of (6.9) by constructing a sparse matrix in `C++` using Eigen [133], and then manipulating and optimizing it in MATLAB [134] with `mex`. For linear equality constraints, we use the optimizer of Jacobson and Panozzo [135, `min_quad_with_fixed`] via the library of Jacobson [136].

6.3.4.1 Data Smoothing

Our curved Hessian discretization can be used to do data smoothing, as described in Section 3.4.1, and some of the examples from that section were solved with our curved Crouzeix-Raviart discretization of $E_{\mathbf{H}}$. Figure 6.15 shows the curved discretization the Hessian energy $E_{\mathbf{H}}$ applied to a standard data smoothing problem. Correctly accounting for the curvature by discretizing the curved Hessian energy is important: the naive extension of the flat discretization from Section 6.1 causes distortion in high-curvature regions when smoothing a step function. In this figure, the

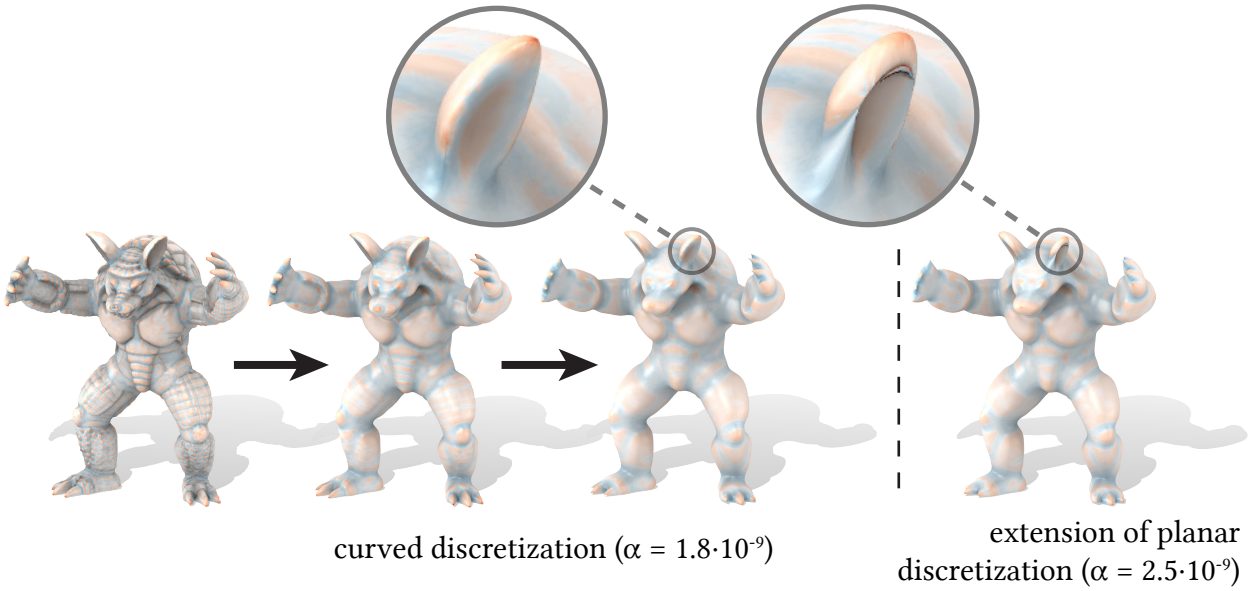


Figure 6.16. Smoothing flow for an armadillo. The surfaces are colored by angle defect. Each step of our curved discretization of $E_{\mathbf{H}}$ (*left*) leads to a smoother result. Smoothing with the naive extension of the planar discretization of $E_{\mathbf{H}}$ (*right*) can lead to artifacts in regions with curvature, such as the highlighted ears. The smoothing parameter α is chosen to produce a similar amount of smoothing in both methods. Three smoothing steps were computed.

smoothing parameters are chosen to give visually similar amounts of smoothing, which means a slightly larger parameter α for the naive extension of the flat discretization.

It is natural to ask why the fact that minimizers of the naive extension of the flat discretization do not solve the biharmonic equation leads to worse results when smoothing the step function of Figure 6.15, but not for the smoothing problems discussed in Section 6.1.2. The smoothing examples from Section 6.1.2 all smooth very noisy functions with a lot of variation everywhere on the surface. The step functions from Figure 6.15 are the exact opposite: the variation is very sparse. This allows the error of not correctly discretizing the curved Hessian to manifest.

Smoothing problems susceptible to bias at the boundary are solved in Figure 3.1 and compared to smoothing with the Laplacian energy E_{Δ} with zero Neumann boundary conditions. As dis-

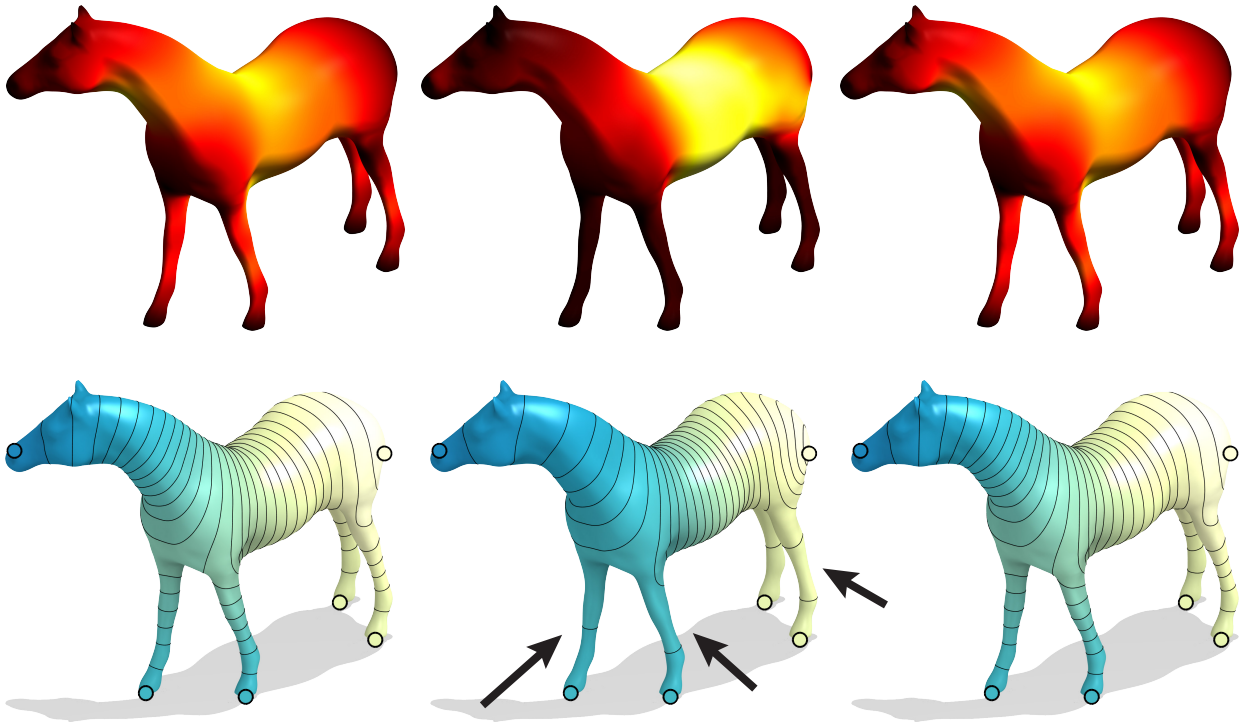
cussed in Section 3.4.1, minimizers of the Hessian energy $E_{\mathbf{H}}$ are not as susceptible to bias at the boundary as minimizers of E_{Δ} with zero Neumann boundary conditions.

The smoothing problem can also be used to smooth the geometry of the surface itself if the input data f from (3.8) is the vertex positions in each coordinate, and the output data u is the new vertex positions. If such a smoothing operation is applied repeatedly, one has a smoothing flow. Figure 6.16 shows our discretization of the Hessian energy $E_{\mathbf{H}}$ applied to such a problem. While the naive extension of the planar discretization can lead to some artifacts due to not accounting for curvature, this does not happen with our correct curved discretization $E_{\mathbf{H}}$.

6.3.4.2 Scattered Data Interpolation

We can use our curved discretization of the Hessian energy $E_{\mathbf{H}}$ to solve the interpolation problems introduced in Section 3.4.2. As can be seen in Figure 6.17, our curved discretization of $E_{\mathbf{H}}$ reproduces the desirable behavior of the Laplacian energy for surfaces without boundary. The naive extension of our planar discretization from Section 6.1.2 fails to do so: the distance between the isolines varies greatly, for example on the legs. The isolines also experience significant bunching at the rump and back of the horse.

As discussed in Section 3.4.2, the Hessian energy with its natural boundary conditions leads to interpolation functions that are less biased by the boundary than E_{Δ} with zero Neumann boundary conditions. Figures 3.7, 3.8 and the right side of Figure 3.9 were solved with the discretization of $E_{\mathbf{H}}$ from this section. Our discretization is robust to the presence of holes in the domain and the geometry of the boundary, and is not as biased by them as the mixed FEM discretization of E_{Δ} with zero Neumann boundary conditions.



Laplacian energy | extension of planar discretization | curved discretization

Figure 6.17. Scattered data interpolation problem solved on a closed surface (*bottom row*) and the gradient of the solution (*top row*). E_Δ (*left*) provides a satisfying result—isolines are relatively evenly spaced, and the gradient is uniform. the naive extension of our flat Hessian discretization (*center*) has large variation in isoline distance (see arrows), and the gradient of the solution is less uniform. Our curved discretization of $E_{\mathbf{H}}$ correctly replicates the behavior of E_Δ.

6.3.5 Related Work

We have discussed literature related to the Hessian energy in Section 3.5, and discretizations related to the Hessian energy in Section 6.1.3.

Chapter 7: Discrete Developability

In Chapter 4, a developability energy E_{dev} is introduced to quantify the developability of surfaces. This energy punishes curvature, and rewards flattenability and the presence of ruling lines. In this section, we present a definition of discrete developability for triangle meshes, as well as a discrete developability energy, inspired by the energy E_{dev} and our continuous definition of developability from Chapter 4. This discrete developability energy can be used to transform an arbitrary triangle mesh into a piecewise developable approximation (Figure 7.1). We show applications of our piecewise developable approximation to the simple fabrication of complex geometric shapes out of inexpensive materials (such as sheets of paper and blocks of plastic).

This chapter describes novel research, except where previous concepts are repeated or previous definitions and previous work is explained (which is indicated).

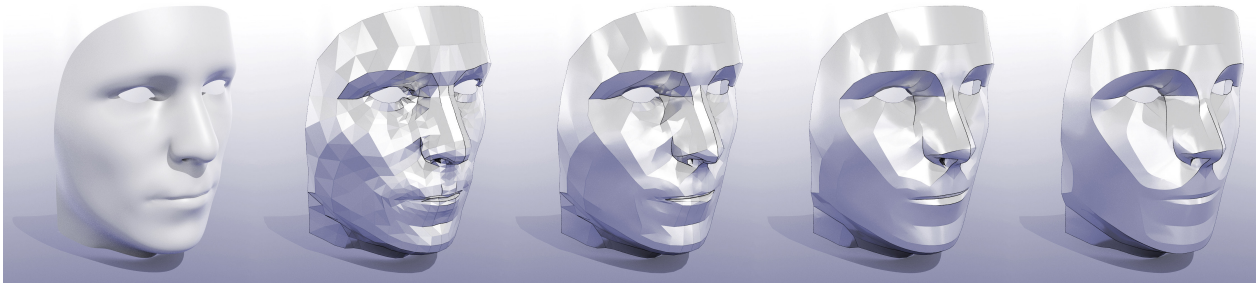


Figure 7.1. By encouraging discrete developability, a surface evolves towards a geometry that can be perfectly flattened away from a sparse collection of regular seam curves.

7.1 Developable Triangle Meshes

Throughout this chapter we will always be working with triangle meshes M characterized by vertices V, edges E, and faces F. This triangle mesh is a subset of $\mathbb{R}^3$ via the inclusion map $f : M \to \mathbb{R}^3$.

7.1.1 Discrete Developability

We quickly recall the most important points of our definition of developability for continuous surfaces from Definition 4.1: a surface is *flattenable* if it can be deformed, without distortion, into a flat surface; a surface is *developable* if it is flat, ruled, and an immersion. These continuous definitions serve as a motivation for our discrete definitions.

Definition 7.1 (Discrete flattenable)**.** *The mesh M is* discrete flattenable *if the angle defect $\kappa(v)$ at every vertex $v \in V$ is zero.*

The angle defect is defined as

$$\kappa(v) \coloneqq 2\pi - \sum_{f \in \mathcal{N}(v)} \theta_v^f \ , \tag{7.1}$$

where the sum is over all faces f in the set of faces containing the vertex v, and θ_v^f is the angle at vertex v in face f [128].

Remark 7.1. *The concept of discrete flattenability from Definition 7.1 is not new [137], and is sometimes also called discrete developable. For the sake of this thesis, we want to sharply differentiate it from our new concept which we will call discrete developability.*

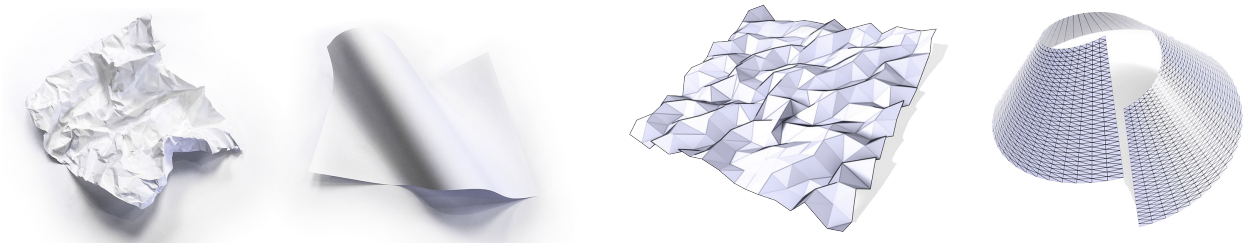


Figure 7.2. Flattenability alone is not enough to ensure that a surface is easy to fabricate. For instance, both the crumpled piece of paper (*far left*) and the noisy triangle mesh (*center left*) can be exactly flattened into the plane, but would be difficult to actually assemble from stiff material. In contrast, the smooth piece of paper (*center right*) and triangle mesh (*far right*) are both exactly flattenable *and* have straight *ruling lines* passing through each point. We seek surfaces of the latter kind.

The angle defect used in Definition 7.1 was featured back in Section 6.3 to discretize the Gaussian curvature of a surface. The flattenability condition ensures that the mesh can be locally flattened in the plane, since the angles around each vertex add up to 2π. Yet, flattenability alone is not sufficient to characterize surfaces that are easily manufactured—consider for instance Figure 7.2 (*bottom left*), which, despite its noisy appearance, has zero angle defect at each vertex. This surface could, in principle, be constructed from an idealized flat sheet, but probably not from real physical materials like sheet metal. Moreover, flattenability alone does not ensure *normal convergence* [138]. A flattenable mesh converging in vertex positions to a smooth surface does not automatically converge to it in normals, and might thus exhibit undesirable behavior when approximating a smooth developable surface.

These observations motivate the need for a stronger condition, namely that (as in the smooth setting) a developable surface should not merely be flattenable, but also exhibit additional regularity that avoids degenerate or pathological behavior. In the smooth setting regularity, was provided by C^2 differentiability (Definition 4.1). In the discrete setting (where we have at most one weak derivative) we replace this continuous condition with a discrete geometric one.

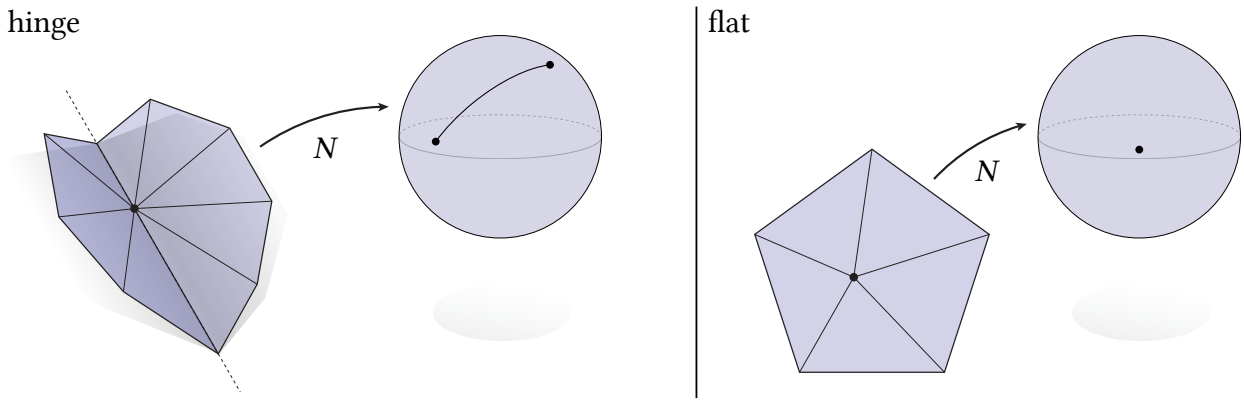


Figure 7.3. A vertex is discrete developable if it looks like a *hinge* or is *flat*—the Gauss map from the vertex neighborhood to the sphere corresponding to the normal vector is a point or a geodesic arc.

Definition 7.2 (Discrete developable)**.** *The vertex star* $\mathrm{St}(v)$ *of a vertex* v *(i.e., the set of all faces containing that vertex) is a* hinge *if the inclusion map* $f : \mathrm{St}(v) \to \mathbb{R}^3$ *is an immersion*[1] *and if the triangles* $f \in \mathrm{St}(v)$ *can be partitioned into two edge-connected regions over which their normals* $\mathbf{N}_f$ *are constant. The vertex* v *is called a* hinge vertex*.*

If the vertex star $\mathrm{St}(v)$ *is completely flat, then* v *is called a* flat vertex *(flat vertices are also hinges).*

Every vertex that is not a hinge vertex or flat is called a seam vertex*. A mesh is* discrete developable *if every vertex is a hinge vertex.*

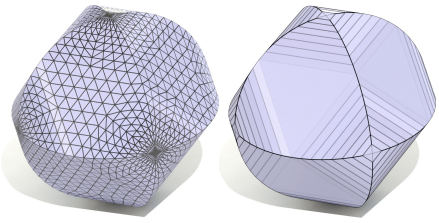
Figure 7.3 illustrates the prototypical configuration for a *hinge*: two planar regions intersecting in a pair of antiparallel edges. A special case is a *flat* configuration (all vertices in a common plane), which admits many partitions into two flat regions. In addition to being locally flattenable, any non-flat discrete developable surface is, as we will show in Lemma 7.2, *discrete ruled*.

[1] A discrete surface (or its inclusion function into $\mathbb{R}^3$) is an immersion if its Jacobian has rank equal to its dimension (i.e., 2) at every point, or, equivalently, if the triangles adjacent to every vertex intersect only at pairwise shared edges.

Definition 7.3 (Discrete ruled)**.** *A mesh M is* discrete ruled *if its inclusion map $f : M \to \mathbb{R}^3$ is an immersion, and if every vertex v is contained in a path of parallel edges $e_1, \cdots, e_m$ with endpoints on the boundary of M (or without endpoints). This path of parallel edges is a* discrete ruling line*.*

Remark 7.2. *Note that the definition of a discrete ruled mesh does not require every edge to be contained in a discrete ruling line, only every vertex.*

Remark 7.3. *The notion of a discrete ruled mesh is not novel, and is used in other geometry processing work that deals with developable surfaces [139, 140].*



Importantly, however, not all discrete ruled surfaces are discrete developable. Discrete developability implies that the triangles between two rulings are all contained in a common plane. Definition 7.2 is therefore compatible with the standard notion of discrete developability for quad meshes [141, 139]: if edges with zero dihedral angle are removed, what remains is a collection of planar quad (PQ) strips. For instance, the inset shows a developable triangulation with edges shaded according to their dihedral angle, revealing a PQ mesh (many other figures in this section are also rendered this way).

Normal Convergence. Discrete developability also seems to imply good approximation of surface normals: roughly speaking, if the normals of a sequence of triangulations approach the normals of an inscribing smooth developable surface (which approach a small arc on the Gauss map), then the triangulations will, by Theorem 7.1, approach discrete developability. Conversely, it seems possible that, if a sequence of discrete developable triangulations is inscribed in a (nowhere flat and sufficiently regular) smooth developable surface, its ruling lines will eventually be con-

tained in the ruling lines of the smooth surface, ensuring normal convergence (although this argument should be taken with care, and more rigorous mathematical examination is needed).

We end this section with three lemmas that show interesting properties of our definition of discrete developability.

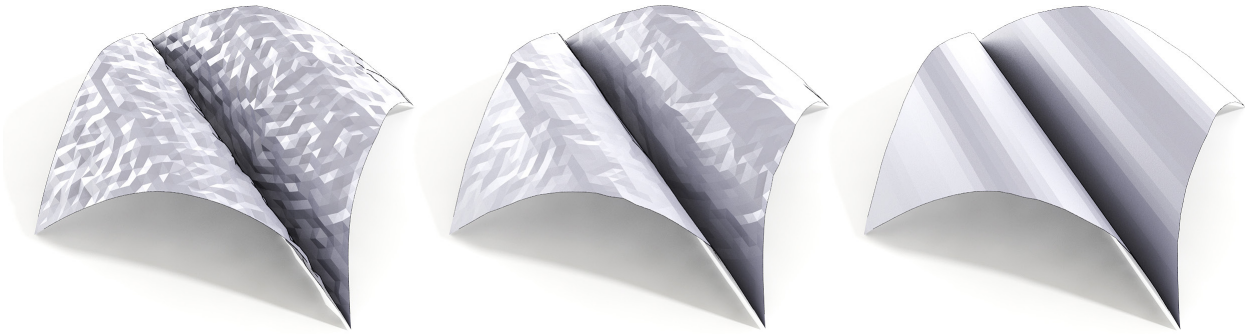
Lemma 7.1. *If a non-boundary vertex v is a non-flat hinge vertex, then it is contained in a pair of antiparallel edges $d, e \in \mathrm{St}(v)$.*

Proof. Let $\mathbf{N}_1, \mathbf{N}_2$ be the normals of the two flat regions of $\mathrm{St}(v)$. Since these regions are edge-connected, there will be exactly two edges d, e that share both normals.

Since v is not a flat vertex, the normals must be distinct ($\mathbf{N}_1 \neq \mathbf{N}_2$). Since it is embedded, they must not be antiparallel ($\mathbf{N}_1 \neq -\mathbf{N}_2$). Hence, the cross products $\mathbf{N}_1 \times \mathbf{N}_2 = -\mathbf{N}_2 \times \mathbf{N}_1$ yield nonzero vectors parallel to the two edges d, e. Both edges start at v and end at different distinct points; by the immersion property d, e can not be parallel. Thus they are antiparallel. □

Lemma 7.2. *Consider a discrete developable mesh M with inclusion function $f : M \to \mathbb{R}^3$ and with no flat vertices. Then M is discrete ruled.*

Proof. By Lemma 7.1, any interior vertex v must be contained in a pair of antiparallel edges d, e that end at two different vertices a, b. Let $\mathbf{N}_1, \mathbf{N}_2$ be the distinct normals determining the edge directions. Since $\mathrm{St}(a)$ and $\mathrm{St}(b)$ each share a pair of triangles with the normals $\mathbf{N}_1, \mathbf{N}_2$, they will each contain a pair of antiparallel edges along the same line (or a single edge in the case of boundary vertices). This pair of antiparallel edges is the ruling line. By jumping from vertex to neighboring vertex repeatedly every such ruling line can be traced from boundary to boundary, or until it loops around. □



noisy surface | minimizing angle defect | minimizing discrete developability energy

Figure 7.4. Denoising a developable sheet of paper (*left*) by simply minimizing angle defect leads to a flattenable but noisy surface (*center*). Minimizing a discrete developability energy yields a smoother ruled surface (*right*).

Lemma 7.3. *Any valence-3 hinge vertex v is necessarily a flat vertex.*

Proof. Suppose v were not flat. Then, by Lemma 7.1, it would have a pair of antiparallel edges d, e that end at two different vertices a, b. But since v has valence 3, d and e must be the edges of one triangle, i.e., v, a, and b are collinear. Thus $\mathrm{St}(i)$ can not be a non-flat a hinge, as its inclusion function would not be an immersion. □

7.1.2 Equivalent Formulations of Discrete Developability

Although it is geometrically straightforward, the notion of discrete developability from Definition 7.2 can be difficult to optimize numerically. We therefore consider an alternative discretization.

Theorem 7.1. *A vertex star $\mathrm{St}(v)$ embedded in $\mathbb{R}^3$ is a hinge if and only if all its triangle normals $\mathbf{N}_f$ are contained in a common plane.*

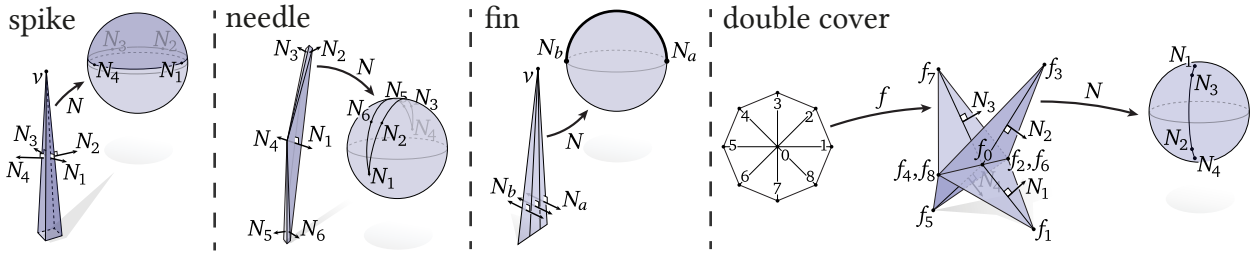


Figure 7.5. A variety of vertex star configurations and their respective Gauss maps (the image of the face normals on the unit sphere). As in the smooth setting, the requirement that the surface be immersed avoids degenerate configurations such as a *needle*, *fin*, or *double cover*. The SPIKE is a degenerate configuration that can occur, if one is not careful, during optimization.

Proof. By definition a hinge has at most two distinct normals, which are always contained in a common plane (as any two vectors are contained in a plane).

Conversely, if all normals are contained in a common plane P, then the cross product of two distinct normals $\mathbf{N}_f \neq \mathbf{N}_g$ from triangles sharing an edge e will be parallel to both the edge vector $\mathbf{v}_e$ and the normal $\mathbf{N}$ of P. Hence, if $\mathrm{St}(v)$ has more than two distinct triangle normals, more than two of its edges will be parallel to $\mathbf{N}$. But since all edges emanate from the same vertex v, $\mathrm{St}(v)$ cannot be embedded. (If $\mathrm{St}(v)$ has two or fewer distinct normals, it is trivially a hinge.) □

The result of Theorem 7.1 is fairly surprising: merely asking that all normals lie in a common plane forces them to bifurcate into two distinct directions, corresponding to the two planes of the hinge. The condition of embeddability plays a role in this bifurcation by preventing the kind of degenerate cases illustrated in Figure 7.5. An easy corollary of Theorem 7.1 is that a vertex is a hinge if and only if the minimum width of its Gauss image (the polygon formed by its normal vectors on the unit sphere) is zero. We will exploit this fact in defining and optimizing our discrete energy (Section 7.2).

Corner Cases. The special configurations illustrated in Figure 7.5 help to further motivate and analyze our definition of discrete developability. The *spike*, *needle*, and *fin* approach configurations where the normals are coplanar and yet the vertex star is not embedded. As in the smooth setting, the condition that the inclusion function $f : M \to \mathbb{R}^3$ must be an immersion provides additional regularity, mitigating undesirable or pathological behavior. The *double cover* further motivates the need for the mesh to be immersed rather than simply nondegenerate: the double cover has a zero-width Gauss map but is not locally injective at the center vertex and hence fails to be a hinge. Degenerate cases present some challenge for numerical optimization, see their discussion in Section 7.2.

7.2 Discrete Developability Energies

Computationally, our goal is to solve the following problem: given a triangle mesh in $\mathbb{R}^3$, find a nearby triangle mesh that is *discretely developable* almost everywhere. We choose to do this by formulating an energy that measures (discrete) developability and evolve the mesh according to its gradient. Although this gradient optimization is not a discretization of the flow of the smooth developability energy E_{dev}, it nonetheless exhibits some of its key features: a given mesh flows towards a piecewise developable surface (as in Figure 7.1), and seam curves are implicitly regularized (Section 7.3.2).

One might also use our notion of discrete developability (Definition 7.2) as the starting point for other numerical algorithms—some ideas are to use discrete developability as a constraint for shape space exploration [142], or to seek a developable triangulation that interpolates points sam-

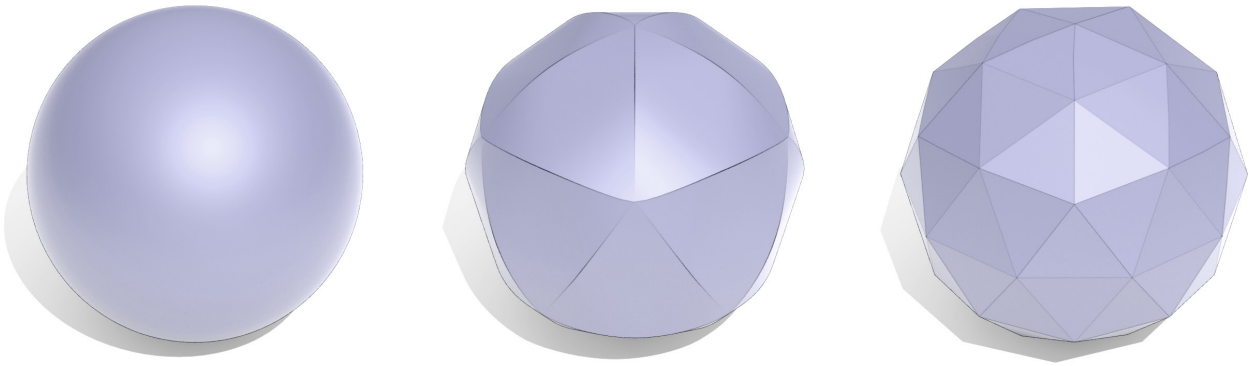


Figure 7.6. Minimizing our discrete energies tends to concentrate the Gaussian curvature of a given smooth surface (*left*) onto a sparse collection of seam curves (*center*). Using an L^1-style optimization to achieve a similar effect results in regions that are flat rather than developable (*right*).

pled from a smooth developable surface [143, 138], although we do not pursue those directions here.

Our smooth energy E_{dev} from Definition 4.3 measures the norm of the first principal curvature κ_1. At each point $p \in M$ this curvature is related to the smallest width of a small ball $B_\epsilon(p) \subset M$ under the Gauss map which maps every point to its normal vector on the unit sphere. The basic idea behind our discrete energy is therefore to penalize the width of the polygon made by the consecutive triangle normals of each vertex star. There is no canonical way to measure this width—our only hard requirement is that the energy of hinge vertices must be zero. We therefore consider two possibilities, namely (i) a simple combinatorial energy based on a direct interpretation of Definition 7.1, and (ii) an energy based on Theorem 7.1 that measures the covariance of triangle normals. Gradient descent on either energy yields an evolution that is reminiscent of the behavior of our smooth energy: empirically, a given triangle mesh tends towards a mesh that is discrete developable away from a sparse collection of seam curves, where all Gaussian curvature is concentrated (see Figure 7.19). Interestingly enough, this behavior does *not* arise from the use of a sparsity-inducing norm, but rather from the simple geometric fact that seam curves are not

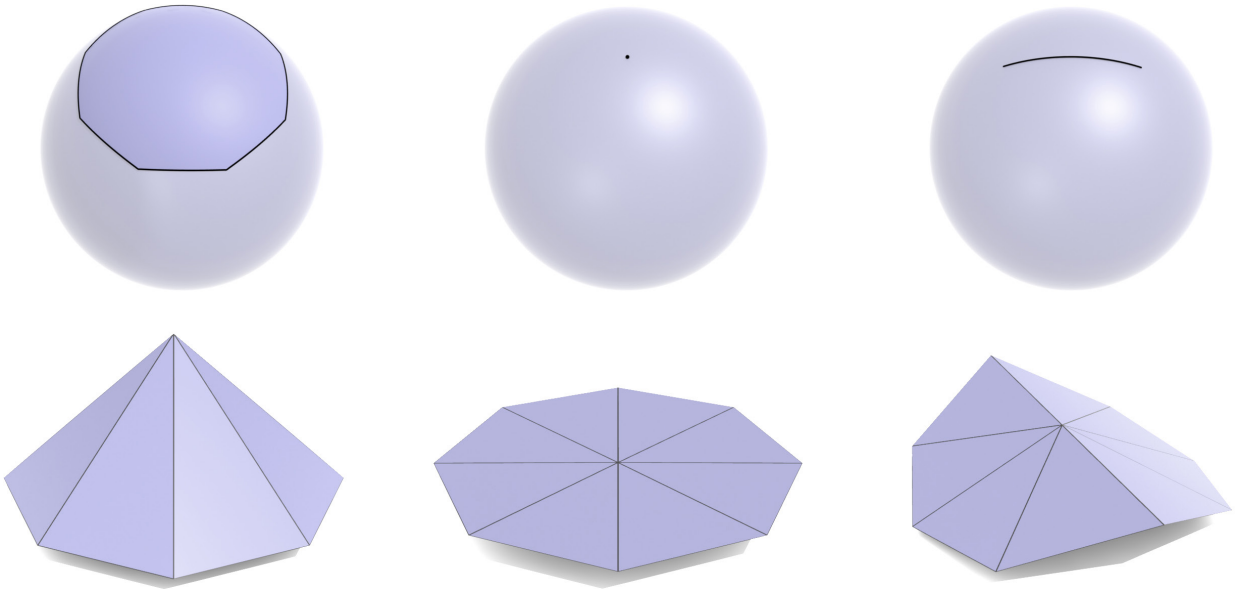


Figure 7.7. Mirroring Figure 4.4, minimizing the squared angle defect of a surface patch (*left*) produces a flat patch (*center*), whereas minimizing our discrete energy (here, $E_{\text{cov}}, E_{\text{comb}}, E_{\overline{\text{cov}}}$), results in a hinge (*right*). The top row features the Gauss maps of the surface patch, the polygon formed by the face normals on the unit sphere.

energetically significant (Section 4.3)—in fact, using an L^1 norm in this context yields surfaces that are piecewise *flat*, rather than piecewise developable, as demonstrated in Figure 7.6. Finally, we do not explicitly encourage pairs of edges to be antiparallel, since flat regions need not be discrete ruled—in fact, flat regions connecting curved pieces may require a triangulation with no antiparallel pairs.

7.2.1 Combinatorial Width

Suppose we partition the faces of a given vertex star $\text{St}(i)$ into two edge-connected regions F_1, F_2 of cardinality n_1, n_2, respectively. Let $\overline{\mathbf{N}}_p \coloneqq \frac{1}{n_p} \sum_{f \in F_p} \mathbf{N}_f$ be the average of the triangle normals $\mathbf{N}_f$ in the region F_p. The degree to which the partition $P \coloneqq \{F_1, F_2\}$ looks like a hinge can be

quantified by the deviation of the normals in each region from their mean:

$$\begin{aligned}\pi(P) &\coloneqq \sum_{p=1,2} \frac{1}{n_p} \sum_{\sigma \in F_p} \left| \mathbf{N}_\sigma - \overline{\mathbf{N}}_p \right|^2 \\ &= \sum_{p=1,2} \frac{1}{n_p^2} \sum_{\sigma_1, \sigma_2 \in F_p} \left| \mathbf{N}_{\sigma_1} - \mathbf{N}_{\sigma_2} \right|^2 ,\end{aligned} \tag{7.2}$$

where $\mathbf{N}_f$ denotes the unit normal of a triangle f. Vertex i is then a hinge if and only if there is a bipartition P for which $\pi(P)$ is zero. Letting $\mathcal{P}_i$ denote the set of all edge-connected bipartitions of $\mathrm{St}(i)$, this motivates the local energy

$$E_{\mathrm{comb}}(i) \coloneqq \min_{P \in \mathcal{P}_i} \pi(P) . \tag{7.3}$$

The *combinatorial energy* E_{comb} is then the sum of $E_{\mathrm{comb}}(i)$ over all vertices i. This energy defines a piecewise differentiable objective that depends continuously on the vertex positions of the mesh, and can hence be optimized via standard gradient-based methods as described in Section 7.3. For the derivation of its gradients, see Appendix C.2. An example of the energy being optimized can be seen in Figure 7.8.

7.2.2 Covariance

Alternatively, consider the characterization given in Theorem 7.1, which says that an embedded vertex star $\mathrm{St}(i)$ forms a hinge if and only if all of its triangle normals are contained in a common plane. To quantify how hinge-like a vertex is, we can therefore measure the average alignment

of the normals with the unit normal $\mathbf{u} \in S^2$ of the best-fit plane:[2]

$$\lambda_i \coloneqq \min_{|\mathbf{u}|=1} \sum_{f \in \mathrm{St}(i)} \theta_i^f \left(\mathbf{u} \cdot \mathbf{N}_f\right)^2 , \tag{7.4}$$

where θ_i^f is the angle at the vertex i in the triangle f, and $\mathbf{N}_f$ is the normal of the face f. Angle weights ensure that the energy does not change if we consider a different tessellation of the same piecewise linear surface. Since (7.4) is just the variational form of an eigenvalue problem, λ_i can also be expressed as the smallest eigenvalue of the 3×3 *normal covariance matrix*

$$A_i \coloneqq \sum_{f \in \mathrm{St}(i)} \theta_i^f \, \mathbf{N}_f \mathbf{N}_f^\top . \tag{7.5}$$

The normal covariance matrix (and matrices similar to it) also arises in the context of surface descriptors [144] and quadric-based mesh simplification [145], where it can be shown that its eigenvalues are related to the squares of the principal curvatures (see, for example, the work of Garland [146, Section 4.4]). This fact provides some intuition for the behavior of this energy: for instance, it makes less aggressive changes to the shape of the vertex star rather than penalizing the discrete Gaussian curvature (the angle defect) squared, since it corresponds to penalizing the smallest width of the polygon generated by the Gauss map, instead of penalizing the area (see Figure 7.7).

The developability of the whole mesh is measured via

$$E_{\mathrm{cov}} \coloneqq \sum_i \lambda_i ,$$

[2] $S^2 = \{\mathbf{u} \in \mathbb{R}^3 \mid |\mathbf{u}| = 1\}$ is the unit sphere in $\mathbb{R}^3$.

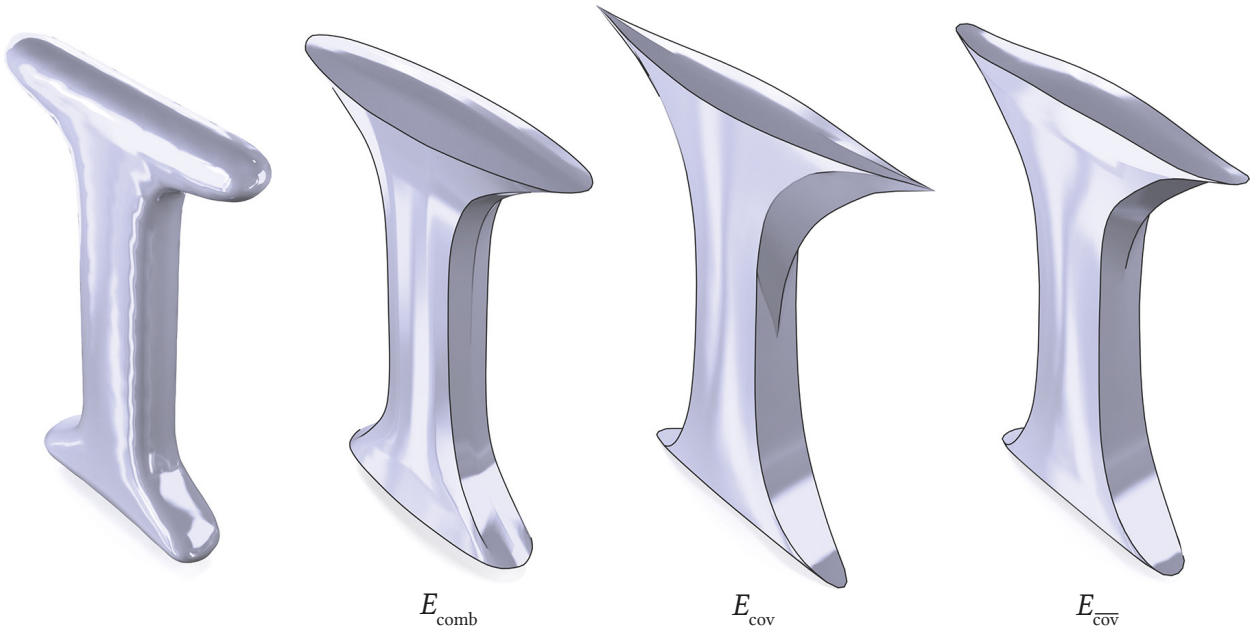


Figure 7.8. A given surface (*left*) flows to a piecewise developable approximation using the combinatorial energy E_{comb} (*center left*) or the covariance energy E_{cov} (*center right*). The enhanced covariance energy $E_{\overline{\text{cov}}}$ reduces the formation of sharp spikes (*right*).

where the sum is over all vertices in the mesh. As with the energy E_{comb}, the only degrees of freedom are the vertex positions. Away from repeated eigenvalues, this energy is smooth and can be minimized using gradient-based techniques, as described in Section 7.3. For the derivation of its gradients, see Appendix C.3. An example of the energy being optimized can be seen in Figure 7.8.

7.2.3 Enhanced Covariance

In certain situations, the minimization of E_{cov} can encourage *spikes* to form; vertices whose normals approach a common plane (Figure 7.5). If a vertex star $\text{St}(v)$ is almost a spike, then all normal vectors are almost contained in the same plane, so λ_v is very small, even though its area under

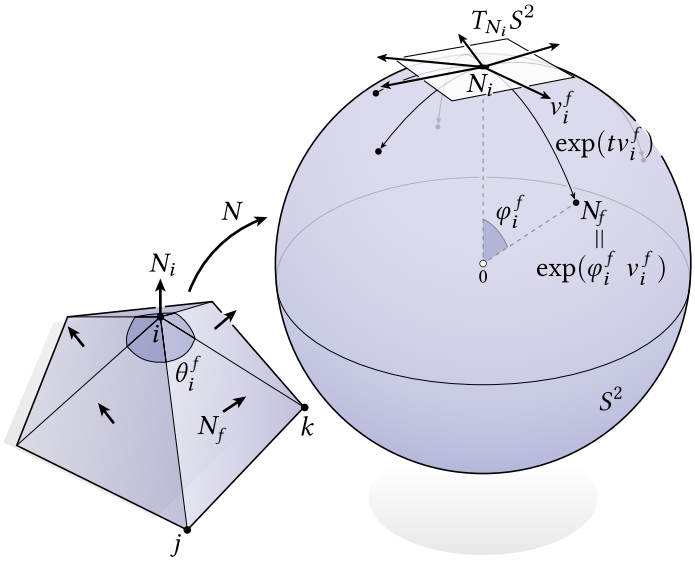


Figure 7.9. For vertex stars $\mathrm{St}(i)$ with a large Gauss image (such as a *spike*-like configuration) one can quantify the width of the spherical polygon as the width of a 2D polygon whose vertices v_i^f are taken to the triangle normals $\widetilde{\mathbf{N}}_f$ by the exponential map exp at the vertex normal $\mathbf{N}_i$.

the Gauss map is very large. When minimizing E_{cov}, such spikes can thus form. This is because E_{cov} does not really measure the minimum width of the Gauss image on the unit sphere itself, but of a three-dimensional proxy for it. This can be remedied by measuring the minimum width of the actual polygon on the Gauss sphere itself, instead of measuring the minimum width of the three-dimensional volume from (7.4), leading to the *enhanced covariance energy*.

A notion of width intrinsic to the Gauss sphere is obtained by expressing the polygon spanned by the face normals in the vertex star in terms of the exponential map at the center of the polygon (Figure 7.9). In particular, if $\mathbf{N}_i$ is the area-weighted vertex normal at a vertex i (i.e., the unit vector in the direction $\sum_{f\in\mathrm{St}(i)} A_f\mathbf{N}_f$) and φ_i^f is the angle from $\mathbf{N}_i$ to some triangle normal $\mathbf{N}_f$ in $\mathrm{St}(i)$, then the triangle normal itself can be expressed as

$$\mathbf{N}_f = \exp_{\mathbf{N}_i}(\varphi_i^f\,\mathbf{v}_i^f)$$

for some unit tangent vector $\mathbf{v}$, where $\exp_p$ denotes the exponential map at a point p on the unit S^2 (see Figure 7.9). More explicitly, this vector can be obtained by simply projecting $\mathbf{N}_f$ onto the plane of $\mathbf{N}_i$ and normalizing:

$$\begin{aligned}\tilde{\mathbf{v}}_i^f &:= \mathbf{N}_f - \left(\mathbf{N}_f, \mathbf{N}_i\right) \mathbf{N}_i,\\ \mathbf{v}_i^f &:= \tilde{\mathbf{v}}_i^f / \left|\tilde{\mathbf{v}}_i^f\right|.\end{aligned}$$

Let now $\tilde{\mathbf{N}}_i^f := \varphi_i^f \mathbf{v}_i^f$. We can quantify how hinge-like a vertex is by measuring the average alignment of the projected normals with the unit direction of minimal width $\mathbf{u} \in S^1$:[3]

$$\tilde{\lambda}_i := \min_{|\mathbf{u}|=1} \sum_{f \in \mathrm{St}(i)} \theta_i^f \left(\mathbf{u} \cdot \tilde{\mathbf{N}}_i^f\right)^2 . \tag{7.6}$$

As with (7.5), the width of the spherical polygon $\tilde{\lambda}_i$ can also be computed by looking for the smallest eigenvalue of the 2×2 matrix

$$\tilde{A}_i := \sum_{f \in \mathrm{St}(i)} \theta_i^f \tilde{\mathbf{N}}_i^f (\tilde{\mathbf{N}}_i^f)^\top .$$

The enhanced covariance energy of the whole mesh is measured via

$$E_{\overline{\mathrm{cov}}} := \sum_i \tilde{\lambda}_i ,$$

where the sum is over all vertices in the mesh.

Again, as with the energies E_{cov} and E_{comb}, the only degrees of freedom are the vertex positions. Away from repeated eigenvalues, this energy is smooth and can be minimized using gradient-

[3] $S^1 = \{\mathbf{u} \in \mathbb{R}^2 \mid |\mathbf{u}| = 1\}$ is the unit sphere in $\mathbb{R}^2$.

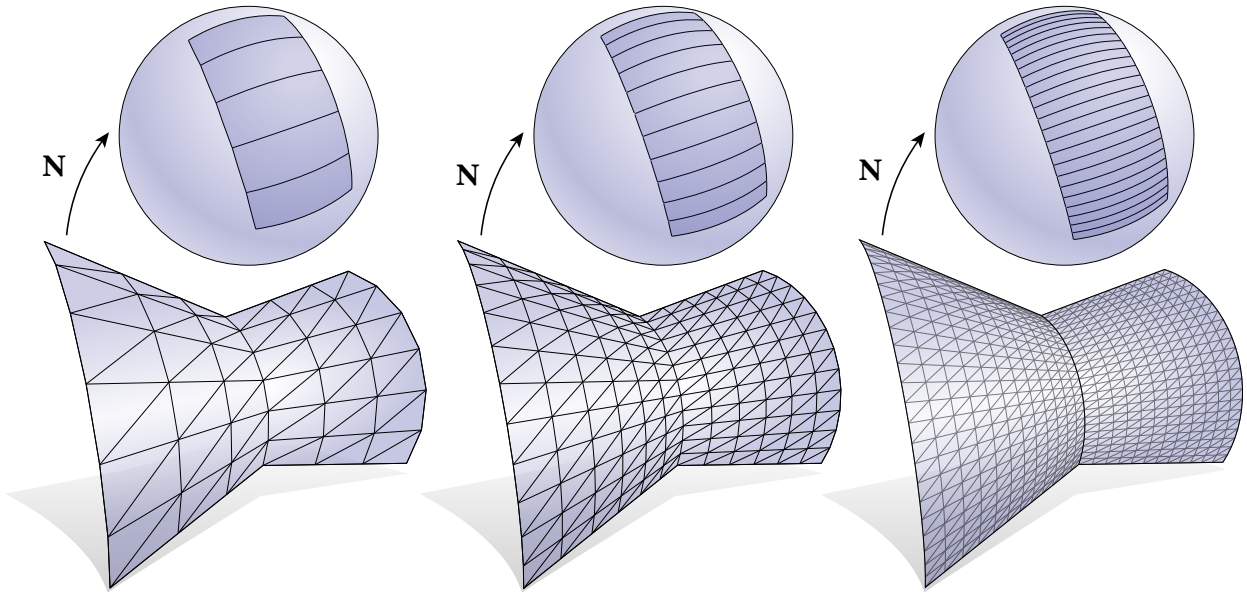


Figure 7.10. For a discrete developable triangulation inscribed in a piecewise developable surface, the developability energy is nonzero only at seam vertices, where the Gauss map (the map from every point to its normal vector on the unit sphere) is nondegenerate. This energy goes to zero under regular refinement.

based techniques, as described in Section 7.3. For the derivation of its gradients, see Appendix C.4. An example of the energy being optimized can be seen in Figure 7.8.

7.2.4 Piecewise Developability

An interesting behavior of the developability energies $E_{\text{comb}}, E_{\text{cov}}$, and $E_{\overline{\text{cov}}}$ is that they tend to encourage not only globally developable surfaces, but also *piecewise developable* surfaces. This is similar to the behavior of the smooth developability energy E_{dev} and its relationship to piecewise developable surfaces discussed in Section 4.3. The behavior of the discrete energies can be understood by considering a refinement sequence such as the one shown in Figure

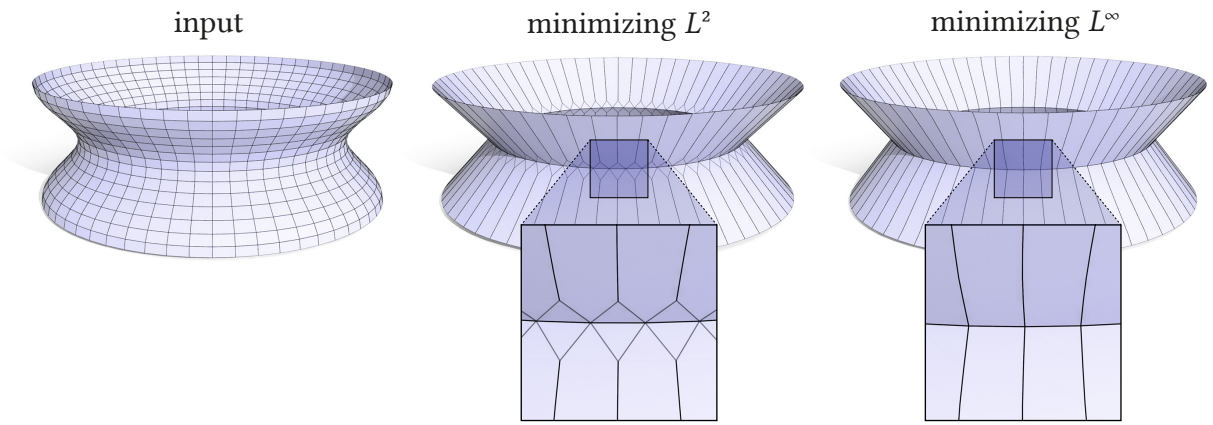
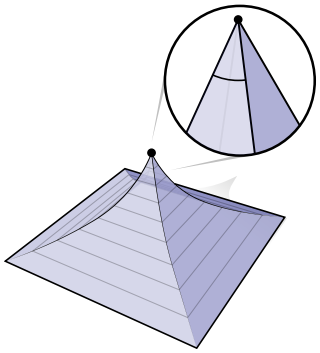


Figure 7.11. Minimizing E_{cov} naturally pushes the input (*left*) toward a piecewise developable surface in an L^2 way, but ruling lines may branch into "V" shapes along seams (*center*). Replacing summation with maximization, in an L^∞ way, yields straight ruling lines (*right*).

7.10, where a mesh is inscribed in a pair of developable pieces meeting along a non-developable seam. Since these meshes are discrete developable away from the seam, the developability energy is nonzero only at seam vertices. However, since E_{comb}, E_{cov}, and $E_{\overline{\text{cov}}}$ measure a quantity related to the smallest width of polygons on the Gauss sphere, the energy contributed by such a seam goes to zero under regular subdivision (consider that the sum of squared widths is a vanishingly small fraction of the sum of widths, which is roughly constant). As a result, surfaces made of multiple developable pieces (see, for example, Figure 4.5 (*right*)) can approach zero energy in the limit of refinement, though the energy will still be nonzero at cone-like points where several seams meet (see inset).



7.2.5 L^∞ energies

At the seams where two discrete developable pieces meet, the discrete energies can produce ruling lines that branch into two (see Figure 7.11, *center*). A simple example can be seen in Figure

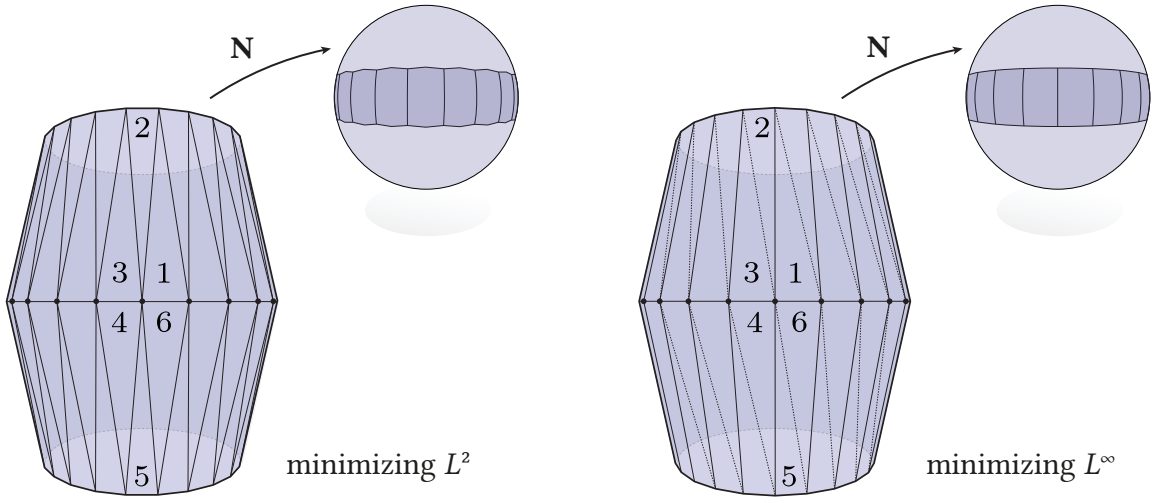


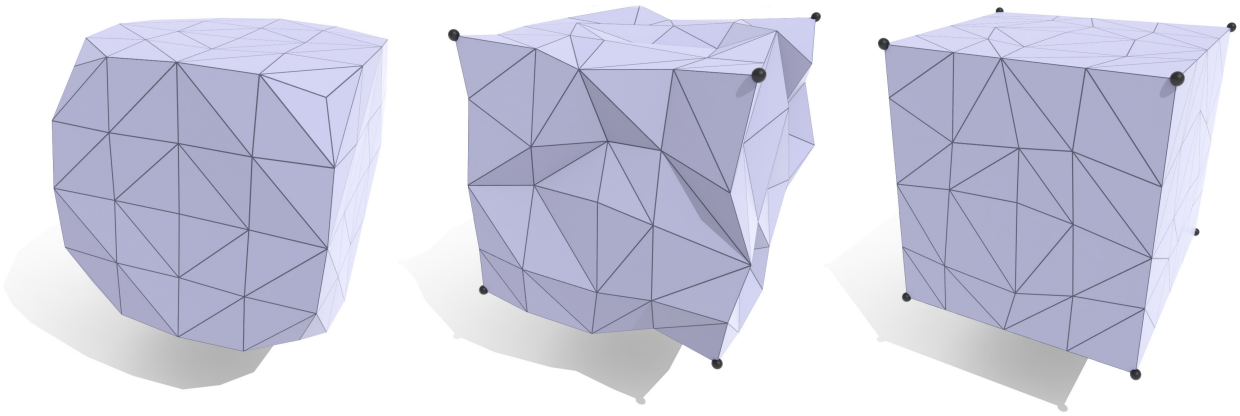
Figure 7.12. Measuring the average deviation from planarity (L^2 deviation) can result in branching behavior at seam vertices where two patches meet (*left*). by minimizing the maximum deviation alignment (L^∞ deviation), we can encourage patches to meet along straight ruling lines (*right*).

7.12, where a branched configuration (*left*) and a perfectly ruled configuration (*right*) yield Gauss images of equal minimum width. However, the branched configuration will have lower energy, since $E_{\text{comb}}, E_{\text{cov}}$ and $E_{\overline{\text{cov}}}$ take a *sum* of terms, thereby providing an *average*, L^2-like notion of polygon width. We can avoid this behavior by simply replacing summation with maximization to get an L^∞-like notion of polygon width. Doing this, $\pi(P)$ from (7.2) becomes

$$\pi^{\max}(P) \coloneqq \max_{F_k \in P} \max_{\mathbf{N}_1, \mathbf{N}_2 \in F_k} (\mathbf{N}_2 - \mathbf{N}_1)^2 \,, \tag{7.7}$$

the objective in (7.4) becomes instead

$$\lambda_i^{\max} \coloneqq \min_{|\mathbf{u}|=1} \max_{f \in \mathrm{St}(i)} \left(\mathbf{u} \cdot \mathbf{N}_f\right)^2 \,, \tag{7.8}$$



flattening valence-3 vertices | input | ignoring valence-3 vertices

Figure 7.13. Valence-3 vertices occur at points where three seams meet. In order to be hinge-like such vertices must be flat, causing rounding of features like corners of the cube (*left*). We therefore omit the contribution of valence-3 vertices from the overall energy, enabling for instance a noisy cube (*center*) to flow to a piecewise developable cuboid (*right*).

(where $\mathbf{u}$ is a three-dimensional unit vector) and the objective in (7.6) becomes

$$\tilde{\lambda}_i^{\max} := \min_{|\mathbf{u}|=1} \max_{f,g \in \mathrm{St}(i)} \left(\mathbf{u} \cdot (\tilde{\mathbf{N}}_i^f - \tilde{\mathbf{N}}_i^g) \right)^2 \tag{7.9}$$

(where $\mathbf{u}$ is a two-dimensional unit vector). We call the new energies $E_{\mathrm{comb}}^{\max}, E_{\mathrm{cov}}^{\max}, E_{\overline{\mathrm{cov}}}^{\max}$ L^∞-like energies, and the energies $E_{\mathrm{comb}}, E_{\mathrm{cov}}, E_{\overline{\mathrm{cov}}}$ L^2-like energies.

In the case of Figure 7.12, the two configurations for the seam vertices now have identical L^∞ maximum energy, and the remaining vertices in the mesh can freely flow toward clean ruling lines. Though more expensive to optimize, this strategy can be used to produce cleaner ruling lines (as demonstrated in Figure 7.11, *right*).

Triple Points. In a piecewise developable surface, seams generically meet at a triple point. This observation fits together well with the fact that valence-3 hinge vertices must be flat (see

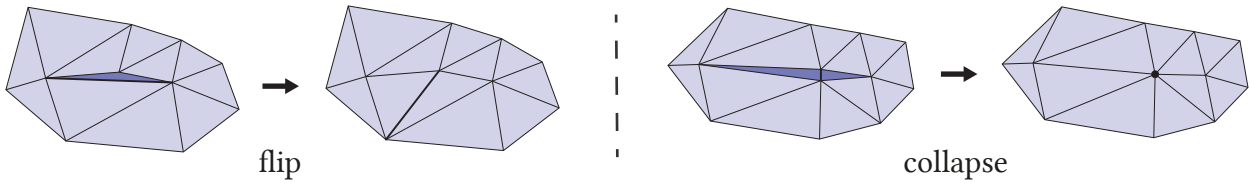


Figure 7.14. To improve numerical stability and avoid degenerate triangles we flip (*left*) or collapse (*right*) triangles with small angles. Other than that, we do not perform any special remeshing to encourage developability.

Lemma 7.3): rather than force such vertices to be developable, we simply omit their contribution to the energy, allowing them to serve as triple points. An example can be seen in Figure 7.13.

7.3 Optimizing the Developability of Surfaces

The next step is to optimize the discrete developability energies of Section 7.2. We choose to apply a straightforward gradient-based optimization, but other optimization strategies are certainly conceivable, and different applications might mandate different approaches.

7.3.1 Numerical Optimization

Energy minimization can be performed via any standard numerical technique for nonsmooth optimization; explicit expressions for gradients are given in Appendix C. We experimented with a variety of methods and settled on a combination of fixed-length and linesearch based gradient descent [147], as well as L-BFGS using the line search of Lewis and Overton [148].

In practice we also found that small interior angles can adversely affect the performance of line search. For triangles with two small angles we therefore perform edge flips; for triangles with a

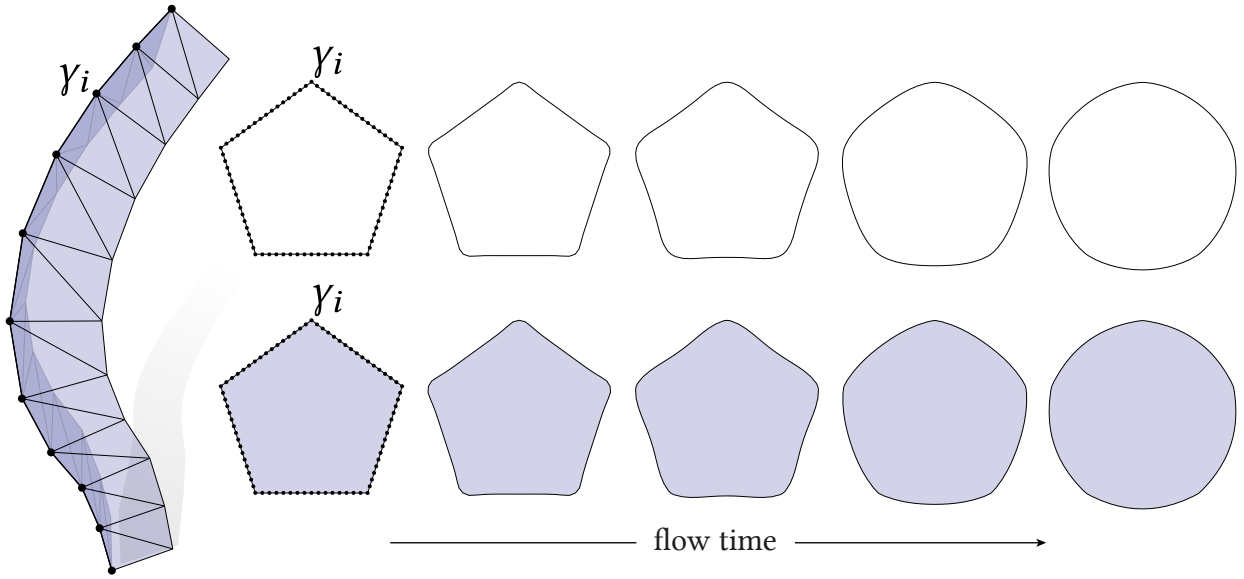


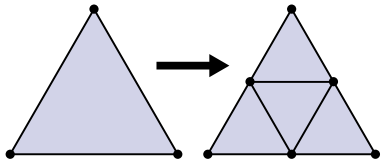
Figure 7.15. Minimizing our discrete energies automatically smooths our seam curves. To compare with planar curvature flow, we construct a surface mesh around a given planar curve (*left*). Gradient-based optimization of our discrete energy results in an evolution (*bottom*) nearly identical to elastic curve flow in the plane (*top*).

single small angle we simply perform an edge collapse (see Figure 7.14). Otherwise, we do not perform any remeshing during the flow. We perform all calculations in `double` precision. For the energies E_{cov} and $E_{\overline{\text{cov}}}$ we found that a sufficiently accurate eigensolver for the matrices $A_i, \tilde{A}_i$ is needed to achieve a stable optimization (we use the solver of Kopp [149]).

7.3.2 Regularization of Seams

In the smooth setting, seams have zero area and finite smaller principal curvature, and hence contribute nothing to the energy (Section 4.3). In the limit of refinement, the discrete seams also do not contribute to the energy (Section 7.2.4). For any finite mesh, however, seam vertices have nonzero contribution to the energy. This regularization provides a natural smoothing of the seam

curves, even though we do nothing to explicitly detect or extract these curves (see Figure 7.15), similar to the seam fairing discussed in Section 4.3. Since the magnitude of energy shrinks as the seam area goes to zero, we can use length scale of the mesh to control the degree of regularization: first, we minimize the energy on an initial coarse mesh to get the basic shape. Once the norm of the energy gradient is below a given threshold (or the overall design is simply satisfactory) we apply regular 4-1 subdivision to all triangles (see inset) and continue minimizing in order to refine seams and improve developability. In practice we tend to start with fairly coarse meshes, and use no more than two or three rounds of subdivision. In Figure 7.1, for instance, we start with a mesh of about 1000 triangles and subdivide twice; notice the natural smoothing of contours in the face.



7.3.3 Cutting and Flattening

During energy minimization, we need not identify which vertices are developable and which are seams. For fabrication or subsequent processing, however, one may wish to cut the model into explicit developable pieces (as in Figure 7.16). We use a simple, automatic strategy: first identify vertices with energy above a user-specified tolerance $\varepsilon > 0$, then compute a cut passing through all such vertices via the method of Erickson and Har-Peled [150] (for surfaces with spherical topology, just a minimal spanning tree). The tolerance can be viewed as a very rough proxy for material stiffness: materials with strong membrane stiffness (like paper) require one to cut more aggressively through non-flat vertices compared to materials that stretch more easily (like leather).

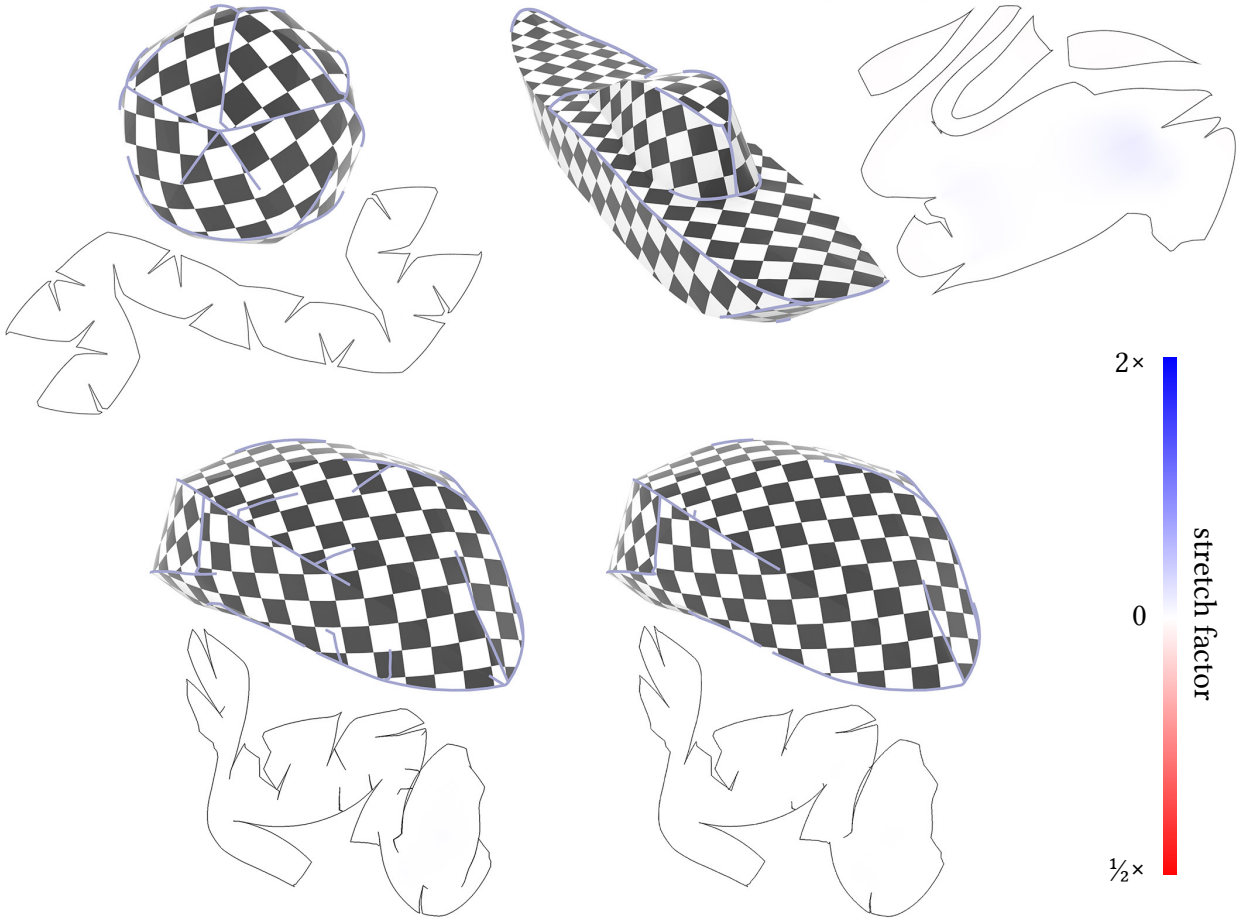


Figure 7.16. Since we do not partition surfaces into patches, they can often be cut and flattened into a single contiguous piece (*top left*), though global overlaps may need to be resolved by cutting into additional pieces (*top right*). Gluing together tiny cracks yields even nicer layouts (*bottom*). (Mostly white coloring indicates that flattening yields virtually zero metric distortion.)

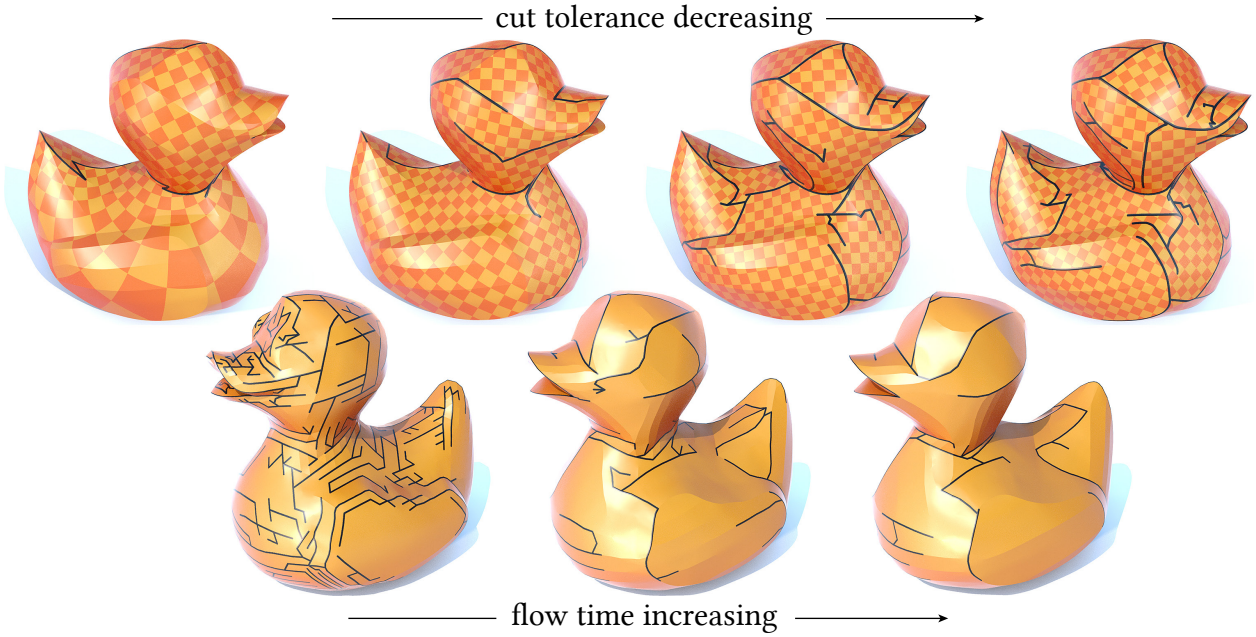
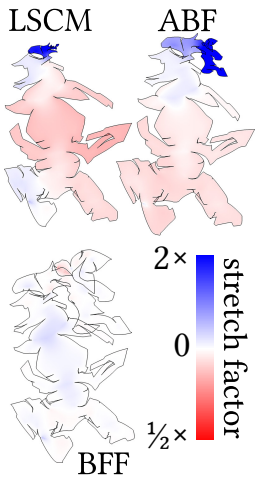


Figure 7.17. Increasing the cut tolerance ε accommodates progressively stiffer materials (*top row*); here setting $\varepsilon = .01, .005, .002, .001$ results in a progressively lower average scale distortion of 2.4, .09, .006, .0004, respectively (also indicated by the checkerboard pattern). For a fixed tolerance ε, our flow tends toward shorter and shorter cuts over time (*bottom row*).



To get aesthetically pleasing cuts, we set edge weights to a linear combination of (i) length and (ii) the dot product between the edge direction and the smaller eigenvector of the matrix A_i at each endpoint (7.5), thereby encouraging cuts to align with principal directions (though such weights are not strictly required for flattening). Since the cut surface has almost no Gaussian curvature, a conformal flattening should, in principle, produce a near-isometric map to the plane. In practice, however, we find that standard methods such as least-squares conformal maps (LSCM) [151] or angle based flattening (ABF) [152] result in large area distortion (see inset), and instead opt for the more recent boundary first flattening (BFF) [153].

Figure 7.17 illustrates the effect of the tolerance ε on the cut, and verifies that our flow tends toward simpler cuts over time. In all figures, we quantify scale distortion via the log conformal factor of the flattening, indicated by the red–blue scale. In practice, we glue together small cracks in parameter space (which have little effect on the final patch layout), and repeat the flattening (see Figure 7.16, *bottom*). Global overlaps are avoided by segmenting the flattening by hand (as in Figure 7.16, *top right*), though automatic tools might be used instead, such as the method of Sorkine *et al.* [154]. An interesting question for future work is to investigate whether the flow itself could be driven to avoid global overlap, perhaps using recent ideas about automatic texture chartification [155].

7.3.4 Algorithm

A top-down description of our algorithm for surface approximation and design is as follows. The input to our algorithm is most typically a coarse mesh of the target surface, obtained either via coarsening (as described below) or by sketching a rough target design, akin to modeling a coarse subdivision cage. Given this initial mesh, we perform energy minimization of one or more of E_{comb}, E_{cov}, $E_{\overline{\text{cov}}}$, $E^{\max}_{\text{comb}}$, $E^{\max}_{\text{cov}}$, and $E^{\max}_{\overline{\text{cov}}}$ until the gradient is sufficiently small, or we are satisfied with the rough shape. We then apply regular 4-1 subdivision (see Section 7.3.2) and repeat this process is until we are satisfied with the quality of the fine mesh. Refinement steps are executed manually based on aesthetic judgments that are part of the design process, however, one might also implement automatic refinement based on the norm of the gradient.

To obtain coarse inputs for our energy optimization we ran the freely-available meshing tool of Jakob *et al.* [28], using a field with triangle symmetry adapted to extrinsic orientation (used in Figures 7.26 (*far right*), 7.8, 7.20, 7.19, 7.21, 7.24, 7.23). Since the tool of Jakob *et al.* [28] aims

for uniform element size, we also ran Willmore flow [56] on Figure 7.20 (*left*), to capture the thin handles on the front of the swingarm.

7.3.5 Discussion

Empirically, different choices of discrete energy, surface tessellation, or even numerical descent strategy have an effect on the resulting piecewise developable approximation, as shown in Figure 7.8. This fact is not at all surprising in light of the ill-posed nature of the problem itself. In general, there is not one clear best way to approximate a smooth surface by developable pieces. As illustrated in Figure 4.1, the effect is particularly pronounced in near-spherical regions (κ_1 close to κ_2), where there is no clear preferred direction for ruling lines. Especially in the initial stage of the flow the choice of mesh and energy has a pronounced effect on the final design. Later in the flow, and at finer levels of tessellation, all energies tend to have very similar behavior.

7.4 Results

In this section we perform numerical experiments and explore preliminary applications of our discrete energies to surface approximation and developable design. We also take a brief forward look towards manufacturing applications.

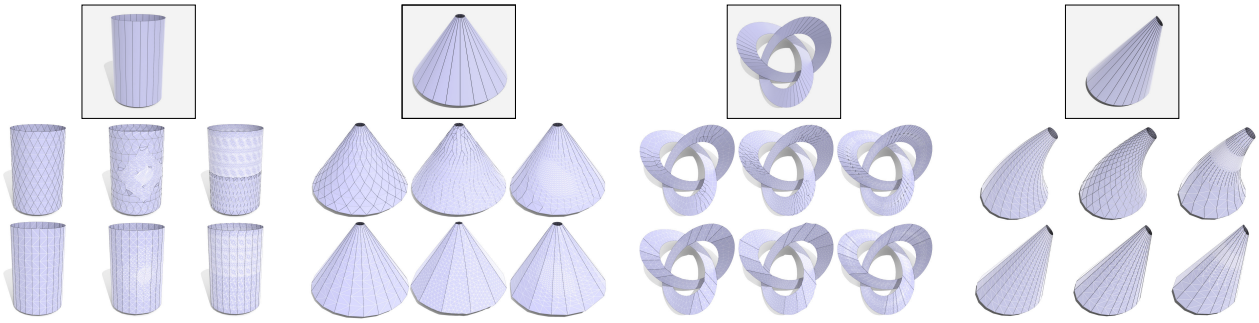


Figure 7.18. Here we examine the effect of the tessellation on surfaces that are developable (*far left, center left, center right*) or nearly developable (*far right*). In each case the middle row shows the input and the bottom row shows the result of optimization. A smooth developable surface is shown at top for reference. Dark edges indicate large dihedral angles, and white edges indicate zero dihedral angles.

7.4.1 Validation

Several experiments indicate that the optimization of our discrete developability energies exhibits the same basic quantitative behavior anticipated by our analysis of the continuous energy E_{dev} from Definition 4.3 (though some features are not captured, as discussed in Section 7.5). Figure 7.19 confirms that our energies concentrate curvature onto seams, becoming highly flattenable everywhere else (as explored in Figure 7.16). Figure 7.4 confirms that merely minimizing the squared angle defect (i.e., the discrete Gaussian curvature squared) yields a surface that is perfectly flattenable yet highly crumpled like a piece of paper. In contrast, we obtain a smoother surface with clear ruling lines using our developability energies. Figure 7.15 confirms that seams are naturally smoothed out by the flow, as discussed in Section 7.3.2. Figure 4.1 emphasizes that for many surfaces, such as the sphere, there is no clear best piecewise developable approximation. Instead, one could guide the flow toward a desired result by picking an initial triangulation that roughly suggests important geometric features.

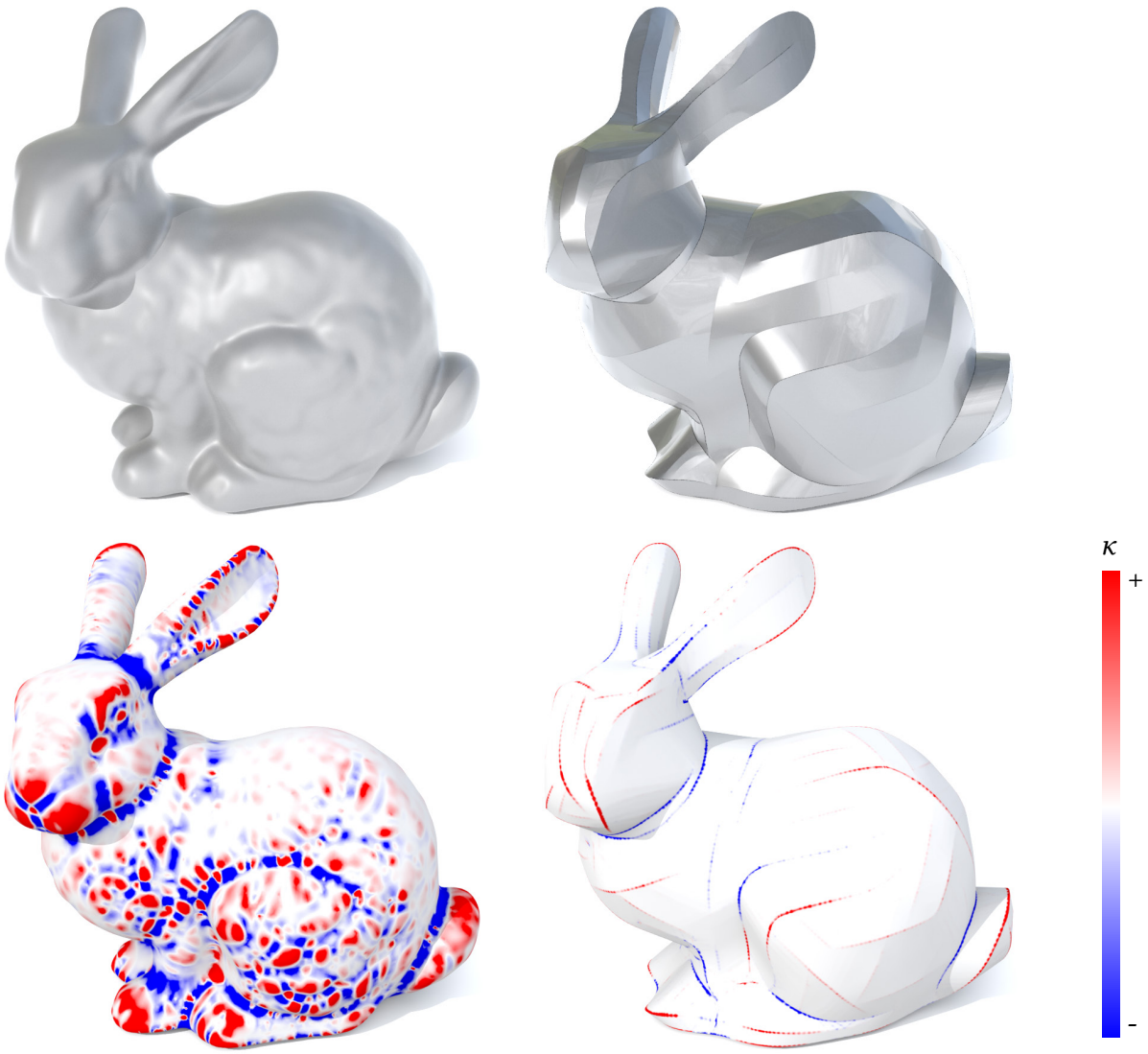


Figure 7.19. Developable flow on the Stanford bunny (*top row*) concentrates the Gaussian curvature κ onto a sparse collection of curves (*bottom row*).

Figure 7.20. Here we approximate a given shape (leftmost in each image) with a surface that is developable away from seam curves. Unlike methods that partition the surface into individual pieces, seams can blend organically into the design. *From left to right:* a swingarm model produced via topological optimization (courtesy of Autodesk); the handle of a drill; and a guitar body, which naturally yields features appearing in real guitar designs (*bottom right*).

Figure 7.18 shows the effect of tessellation on surfaces that are already close to developable. Here we minimize the covariance energy E_{cov}, perform no preprocessing, and apply no constraints or projections to keep the mesh close to the initial surface. For these simple examples, the tessellation has little effect on the overall shape, but still encourages a ruling line parameterization. Note also that the choice of mesh will affect the number of ruling lines. For instance, the cone at center has far fewer vertices on the top boundary component than the bottom—since, by Lemma 7.2, ruling lines must have endpoints on the boundary, there cannot be a discrete developable surface with a ruling line passing through all bottom boundary vertices. For a similar reason, local adaptivity provides little value, since not every vertex in the fine region can belong to its own ruling.

7.4.2 Developable Approximation and Design

A long-standing challenge in digital manufacturing is automatic approximation of general curved surfaces by high-quality ruled developable pieces. Our flow represents the first stage in such a

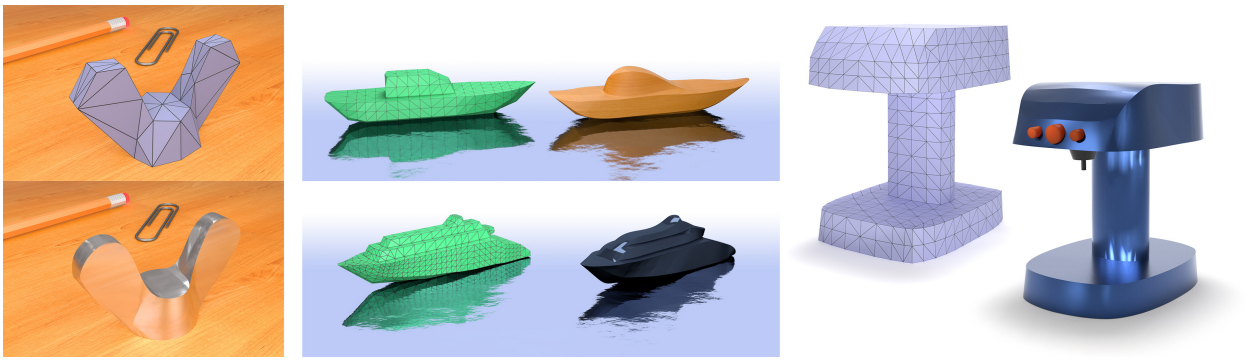


Figure 7.21. Beyond the traditional task of shape approximation, our flow enables developable shape exploration by starting with a coarse mesh and flowing/refining toward a piecewise developable surface with organic flow lines. (Input mesh in each figure is shown in wireframe.)

pipeline, taking any unstructured mesh to one or more piecewise developable designs. A more complete pipeline might entail, e.g., conversion into a clean network of developable splines (as considered by Tang *et al.* [156]). This, in turn, would facilitate tool path planning for cylindrical flank milling [157]. We ran our method on a variety of models, including those pictured in Figures 7.1, 7.19 and 7.20.

Beyond shape approximation, we can use our flow for high-level *form finding* to design piecewise developable surfaces. As described in Section 7.3.4, we sketch a rough, coarse input mesh and refine it through repeated energy minimization and subdivision. Examples are shown in Figure 7.21, where each model captures the basic design intent, but also suggests design possibilities not originally conceived by the user.

Since there are many different piecewise developable approximations of a given surface, the choice of input mesh will influence the final result. This effect is most pronounced for coarse meshes, where different tessellations can have a significant effect on the overall geometry (Figure 7.22). For finer meshes, input tessellations effectively act as different parameterizations, and have a less significant overall geometric effect (Figure 7.18).

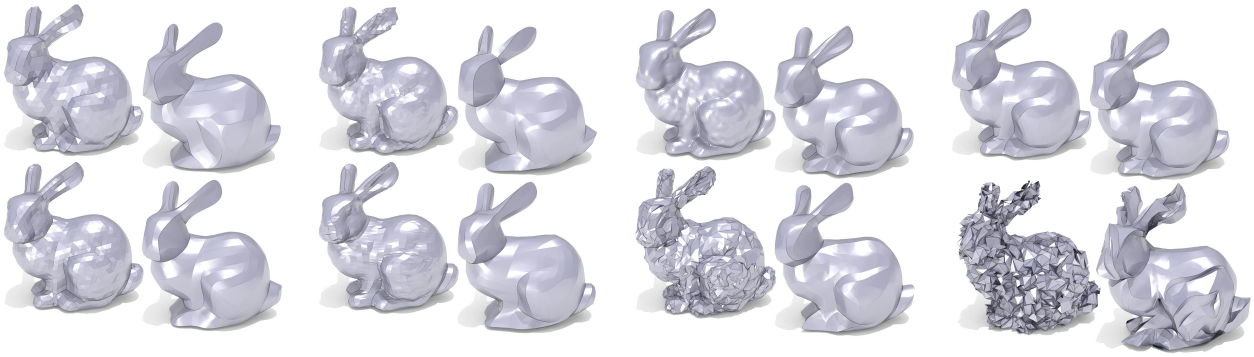


Figure 7.22. Different initial meshes lead to different piecewise developable approximations, though the bulk shape typically remains quite close to the input. Overall, fine meshes tend to yield a closer approximation by a larger number of pieces, and edges roughly aligned with principal curvatures tend to better preserve geometric features. Moderate noise is typically smoothed out. Large amounts of noise can affect the overall shape.

As illustrated in Figure 7.8, different energies will yield different solutions. We used $E_{\text{comb}}, E_{\text{comb}}^{\max}$ for Figures 7.1, 4.1, and 7.24. We used $E_{\text{cov}}, E_{\text{cov}}^{\max}$ for Figures 7.6 (*center*), 7.25 (*bottom*), 7.4. In Figures 7.17, 7.20, and 7.19 we also used $E_{\overline{\text{cov}}}, E_{\overline{\text{cov}}}^{\max}$. In Figures 7.21 and 7.23 we used $E_{\text{comb}}, E_{\text{comb}}^{\max}$ in the coarse phase and $E_{\text{cov}}, E_{\text{cov}}^{\max}$ (or $E_{\overline{\text{cov}}}, E_{\overline{\text{cov}}}^{\max}$) in the fine phase. Though different energies can be used to tweak the design, we find that in general just using one of the energies will produce reasonable default designs for most models.

Runtimes for all examples were on the order of seconds to a few minutes on a 3.4GHz Intel Core i7 laptop with 32GB of RAM. We did not use multithreading, though gradient calculations could easily be parallelized. Especially during the initial coarse phase, the mesh rapidly evolves towards a shape that looks much like the final design. Subsequent refinement takes no more than a few minutes, especially with L-BFGS and an appropriate line search (Section 7.3.1).

Figure 7.23. A simple test of fabricability. An initial mesh of a face (*left*) is optimized to obtain a piecewise developable approximation (*center*). The approximation is then cut from paper, glued together, and spray painted to obtain a physical model (*right*).

7.4.3 Fabrication

As a simple test of physical feasibility, we fabricated a piecewise developable surface using a consumer cutting plotter (Silhouette CAMEO), shown in Figure 7.23. Here additional cuts had to be made to avoid overlap in the flattened domain. An interesting direction for future work is to flow surfaces toward those with *curved folds* [158], so that they can be formed from a single sheet of material. One possibility here is to penalize only the squared angle defect along curved folds (instead of our developability energies).

Our developable energies provide only one small piece in a larger pipeline, but we can still get a glimpse of how our designs might facilitate manufacturing via *flank milling*. As opposed to traditional *point milling*, which carves out material using only the tip of a cutting tool, flank

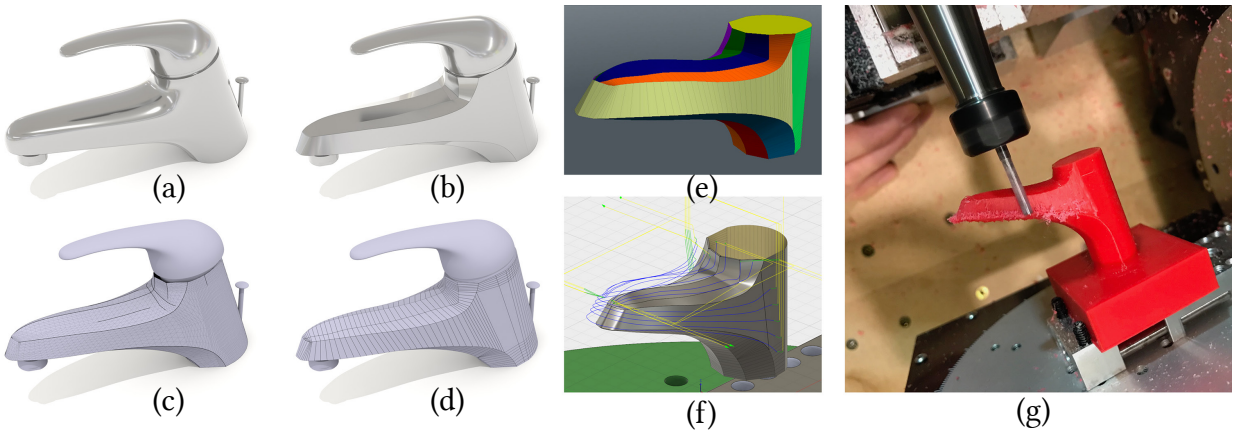


Figure 7.24. Prototype of fabrication via flank milling. An originally smooth faucet design (a) evolves into a piecewise developable surface (b) with fairly clear seam curves and ruling lines (c) which are extracted by hand (d) in a polygon mesh editor and partitioned into PQ strips (e). These strips are then semi-automatically converted into tool paths (f) via NURBS patches. A contour of the final piece is then flank milled using a hobbyist 5-axis CNC mill (g).

milling sweeps a cylindrical bit *along* the surface, allowing for faster cuts and higher-quality surfaces [76]. By starting with a piecewise developable design, one can ensure that material is not erroneously added nor subtracted to the manufactured piece due to out-of-plane twisting of the bit along the toolpath. A common practice is to first cast or 3D print the bulk *near net shape* (thereby avoiding excess waste), then use flank milling to obtain accurate mating surfaces, threads, etc., which must come into precise contact. Figure 7.24 prototypes such a process starting with the minimization of our developability energies, followed by manual extraction of PQ strips in a polygon mesh editor, and subsequent toolpath generation in *Autodesk Fusion 360*. Milling was performed using a *Pocket NC V1*, a USD $5000 hobbyist-level 5-axis mill. Although each individual contour is easily machined, extracting a global spline network, as well as planning a globally feasible toolpath, remains an interesting challenge for future work.

7.5 Limitations and Open Questions

Although we have laid some very basic foundations in understanding the developability energy E_{dev} and our many discrete developability energies, many questions remain. On the theoretical side, there is still the fundamental question of how to formulate the task of finding the best piecewise developable approximation as a well-posed problem. Simply following a gradient flow does not ensure that one arrives at the best solution, and asking for the closest piecewise developable surface (e.g., in the Hausdorff sense) is not meaningful since one can find crumpled piecewise developable surfaces arbitrary close to a given smooth surface (see the discussion in Section 4.1).

On the numerical side, there are interesting open questions about how to faithfully discretize a gradient flow of the developability E_{dev}. Notably, if we seek to approximate the evolution of the parameterization of our mesh, rather than just the mesh itself, gradient flow on our discrete energies clearly does not provide a convergent discretization. Consider, for example, the Schwarz lantern (Figure 7.25, *bottom*), where a mesh inscribed in the cylinder flows to a discrete developable mesh, even though cylinders are stationary points of the smooth flow. Gradient descent on our discrete energy exhibits tangential motion, even though the continuous behavior is in the normal direction only. The question of how to obtain convergent parametric discretizations for such geometric flows has been studied to some degree (see, for instance, the discussion by Perl *et al.* [160]).

At a more practical level, our current numerical optimization strategy does not produce perfectly straight ruling except on fairly simple meshes. A better understanding of this issue could lead to meshes with a cleaner PQ structure (perhaps by incorporating more sophisticated remeshing). More broadly, our flow is only the first step in a fully automated pipeline for taking a given

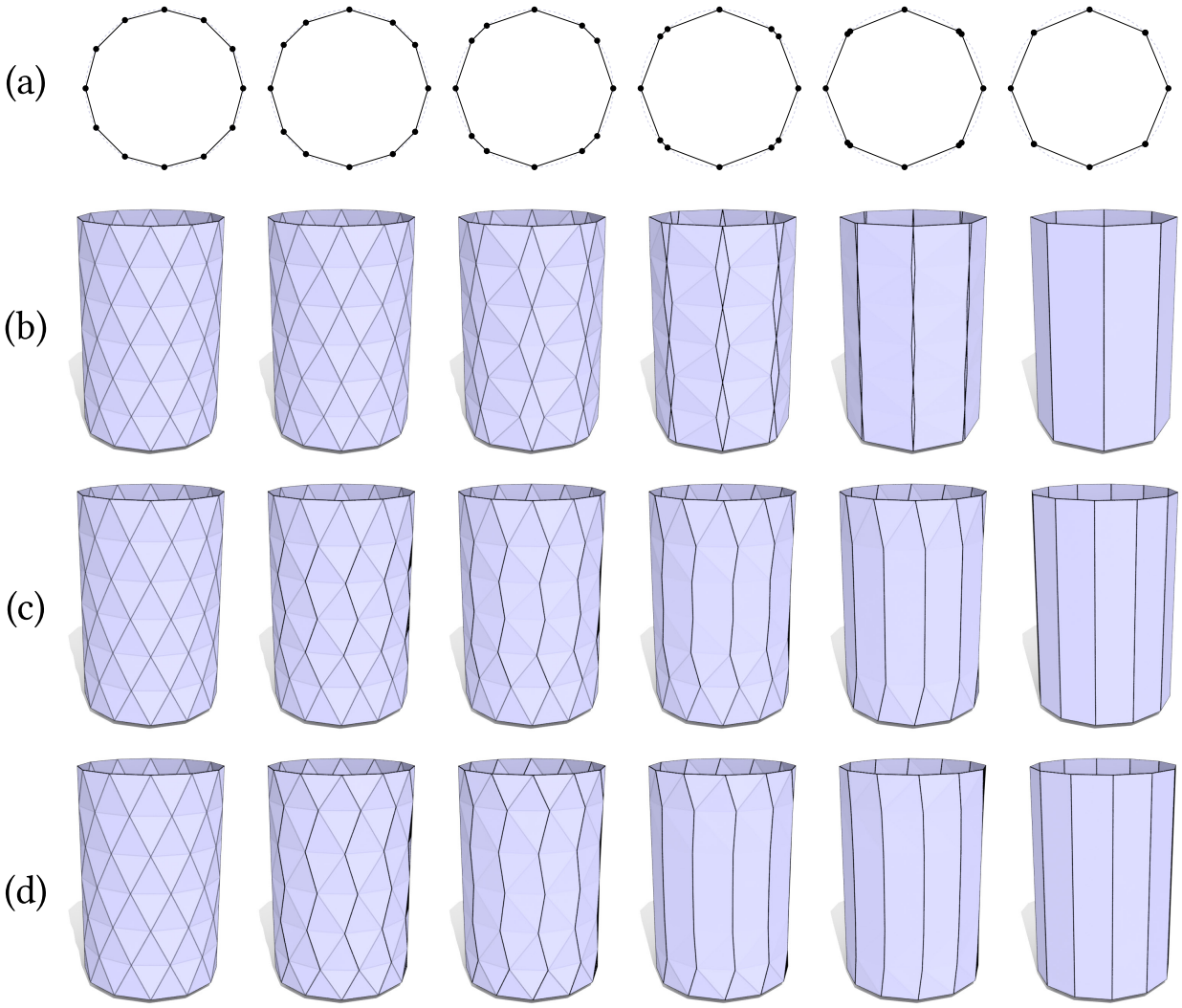


Figure 7.25. In the smooth setting, the total length, total area, Willmore energy, and developability energy are invariant with respect to different parameterizations of the surface. For a triangulated surface (or curve), however, these quantities are, of course, mesh-dependent. As a result, gradient descent on each discrete quantity may yield tangential motion that is not present in the corresponding smooth flows. Here for instance we constrain the geometry to a cylinder (or circle) and run (a) curve shortening flow, (b) mean curvature flow (like Desbrun *et al.* [18]), (c) discrete Willmore flow (like Bobenko and Schröder [159]), and (d) gradient descent of our discrete developability energy E_{cov} starting from the Schwarz lantern, which flows to a mesh of smaller length, area, etc.

input surface all the way through the process of developable approximation, decomposition into clean developable pieces (such as PQ strips or spline developables), and final manufacturing via roll bending or flank milling. For milling, a complete solution would also need to accommodate mechanical constraints such as collision avoidance, possibly via joint optimization of the geometry and the tool path.

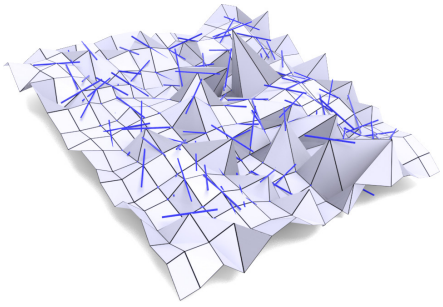
7.6 Literature

In this section we discuss previous work relating to developability on discrete surfaces. A discussion of continuous developability energies and flows in literature can be found in Section 4.4.

7.6.1 Discrete Developability

Recently there has been an interest in *mimetic* notions of developability that aim to preserve key properties of continuous developability even on coarse meshes in the spirit of *discrete differential geometry* [161, 162]. For triangle meshes, a seemingly natural definition is the vanishing of *angle defect*, but this definition permits highly irregular geometry devoid of *ruling lines*, as discussed in Section 7.1.1. Other notions of discrete developability are based primarily on regular quadrilateral nets rather than triangulations, requiring a global quad layout that cannot easily adapt to changes in geometry. For instance, planar quadrilateral (PQ) strips provide a natural analogue of developable surfaces, since they can be isometrically flattened and have well-defined ruling lines [141, 139]. Pottmann *et al.* [163] consider *semi-discrete* developability, based on a similar per-

spective. Solomon *et al.* [164] present a framework where ruling directions are freely variable, but the global patch layout must still be determined a priori. Rabinovich *et al.* [140] propose a definition based on orthogonal geodesics, also assuming a global quad layout. Like the angle defect condition, meshes that exactly satisfy this definition can still be highly crumpled, as ruling lines are defined locally at each vertex, but are not, in general, globally coherent (see inset, which shows a crumpled discrete geodesic net with blue lines indicating local ruling lines). Our notion of discrete developability for triangle meshes (Definition 7.2) is simultaneously compatible with both the zero angle defect notion (thereby ensuring that the mesh can be locally flattened) as well as the PQ definition (thereby providing well-defined ruling lines). Since it is purely local, it applies to any manifold triangle mesh, making it suitable for general-purpose developable approximation and design.



7.6.2 Developable Approximation and Design

Several methods approximate a given mesh by (near-)developable pieces. Wang and Tang [165] directly optimize angle defect, which results in the crumpling behavior discussed in Section 7.1.1. Mitani and Suzuki [137] generate *triangle strips* which can be trivially unfolded onto the plane but lack clear ruling lines. Shatz *et al.* [166] instead fit strictly conical patches which can have interior cone points. Massarwi *et al.* [167] partition surfaces into *ruled* pieces, which become developable only upon triangulation. Rather than augment the geometry, Julius *et al.* [168] find patches that can be flattened with low distortion. All of the methods discussed above apply some sequence of mesh processing operations (clustering, fitting, remeshing, etc.) involving parameters that can

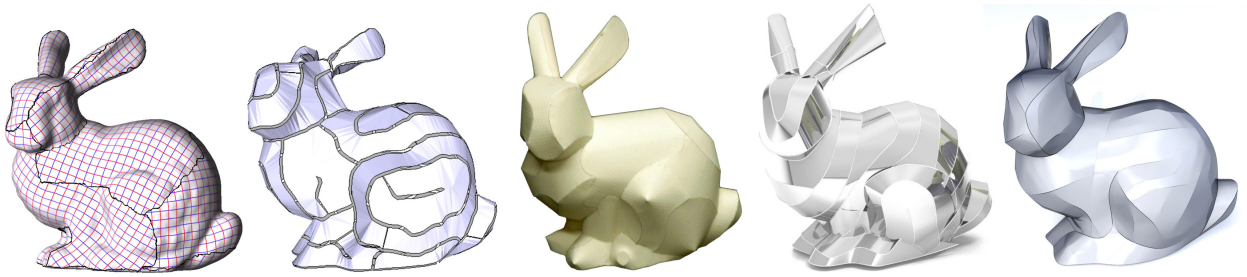


Figure 7.26. A variety of methods have been developed to approximate a given surface by easily flattenable pieces. Left to right: the methods of Julius *et al.* [168], Mitani and Suzuki [137], Shatz *et al.* [166], Tang *et al.* [156] (which requires manual editing), and our method.

be difficult to understand and control. Moreover, they are not suitable for the type of coarse form finding we explore in Section 7.4.2, and most do not provide clear ruling directions (which help facilitate manufacturing).

Other methods focus on user-guided *design* rather than automatic approximation. Kilian *et al.* [158], for instance, explore *curved folding*, Tang *et al.* [156] consider user-driven spline networks, and Rose *et al.* [169] interpolate sketched boundaries. Methods for simulating thin sheets [170, 171] can also be used for design exploration. In terms of output quality, there are no universally accepted criteria for what makes a good developable design. We show the results of several algorithms in Figure 7.26. On the whole, our flow produces models that are at least comparable in quality to previous work, and exhibit some nice features not exhibited by other approaches such as

1. no requirement to partition the surface into disk-like pieces;

2. automatic smoothing of feature lines;

3. natural emergence of ruling directions.

These features of our method arise naturally from our variational approach, i.e., the minimization of an energy supported in the star $\mathrm{St}(i)$ of each vertex i.

7.6.3 Discretizing Geometric Flows

Many of the geometric energies and flows mentioned in Section 4.4 have been discretized in previous work. There are a variety of integrators for flow such as mean curvature [172, 18], Gaussian curvature flow [173], Hamilton's Ricci flow [174], Willmore flow [49], and specific surface processing flows [84, 175]. These previous works, however, do not discretize a flow to specifically encourage discrete developability.

While the gradient optimizations of our discrete developability energies are *not* a discretization of the smooth geometric flow of E_{dev}, they share some of the characteristics of other discrete geometric flows [159]. Rather than a direct finite element approximation of a specific flow, we construct a discrete energy that exactly captures a few key structural properties of the smooth energy. In particular, minimizers must be locally flattenable and exhibit straight ruling lines. Our experiments indicate that gradient optimization of our discrete energies exhibits some of the behaviors anticipated by the smooth analysis in Section 4.3: a given initial surface evolves towards surface that is piecewise developable.

Conclusion

In this thesis, I proved that the piecewise linear Lagrangian mixed finite element method discretization of the Laplacian energy with zero Dirichlet and zero Neumann boundary conditions on non-flat triangle meshes converges with rates reminiscent of the flat case, provided that certain conditions hold on the mesh. I have also introduced and discretized the novel Hessian energy, whose natural boundary conditions produce minimizers that are less biased by the boundary than minimizers of the Laplacian energy with zero Neumann boundary conditions. I presented a new smooth developability energy with an associated gradient flow that is minimized by piecewise developable surfaces, I introduced a new definition for the developability of triangle meshes, and I introduced novel discrete developability energies for triangle meshes that can be minimized to obtain discrete, piecewise developable surfaces.

The research mentioned above inspires a wide variety of future work, some of which is mentioned in the thesis itself. It would be interesting to derive a proof of convergence for the Hessian energy's Crouzeix-Raviart discretization, similarly to the convergence proof for the Laplacian energy that is provided in this thesis. Similarly intriguing would be an approach to use this discretization for a L^1 Hessian energy on curved surfaces. Compelling future work on the topic of

the developability energy could consist of a principled discretization of the developability energy E_{dev}, and a robust optimization of the discrete developability energies.

A common thread through all of the work in this thesis as well as the future work mentioned above is the introduction of principled, precise, and (ideally) theoretically verifiable methods to geometry processing, and the theoretical verification of existing methods. I plan to make this a focal point of my future research in applied mathematics as well: my goal is to mathematically examine, analyze, and explain existing geometry processing methods, and introduce new methods that are founded on a sound mathematical basis. This use of applied mathematics in geometry processing will help make geometry processing more principled, easier to use because of theoretical guarantees, and increase their utility.

Attributions

The following external meshes are used in this work, and I am grateful to the people who provided them.

Venus of Milo by smkmuseum [176]

Tower by jansentee3d [177]

Helmet by Javo [178]

Spot the cow by Crane [179]

Airplane by YahooJAPAN [180]

Back arching statue by Crane [179]

Stanford bunny by Stanford Graphics [181]

Beetle car by Sutherland [182]

Noisy scan of old phone from NIST [183]

Chimpanzee mesh by SpinQuad1976 [184]

Nefertiti by Al-Badri and Nelles [185]

Various shapes (hand, lily, duck, cartoon head) from Jacobson and Panozzo [186]

Camel head from Jacobson and Panozzo [186]

Armadillo by Stanford Graphics [181]

Horse from Jacobson and Panozzo [186]

Fish by Crane [179]

Face meshes from Crane [187]

Rubber duck by willie [188]

Swingarm model by Autodesk [189]

Drill by chachan [190]

Faucet mesh from Crane [187]

The following external images are used in this work. I thank the following sources.

Child with headphones by Jacobson [191]

Cheeseman cartoon by Jacobson [192]

Pants by Jacobson [191]

Corrugated vase by Dudte *et al.* [79]

Embedded flat torus by Borrelli *et al.* [78]

Cartoon cactus by Jacobson [191]

Cartoon octopus and arm by Jacobson [191]

Pictures of electric guitar and Warlock guitar from Crane [193]

Method output images by Julius *et al.* [168], Mitani and Suzuki [137], Shatz *et al.* [166], and Tang *et al.* [156]

References

[1] O. Stein, E. Grinspun, A. Jacobson, and M. Wardetzky, "A mixed finite element method with piecewise linear wlements for the biharmonic equation on surfaces," 2019. arXiv: 1911.08029 [math.NA].

[2] O. Stein, E. Grinspun, M. Wardetzky, and A. Jacobson, "Natural boundary conditions for smoothing in geometry processing," *ACM Trans. Graph.*, vol. 37, no. 2, May 2018.

[3] O. Stein, A. Jacobson, M. Wardetzky, and E. Grinspun, "A smoothness energy without boundary distortion for curved surfaces," *ACM Trans. Graph.*, vol. 39, no. 3, Mar. 2020.

[4] O. Stein, E. Grinspun, and K. Crane, "Developability of triangle meshes," *ACM Trans. Graph.*, vol. 37, no. 4, Jul. 2018.

[5] M. do Carmo, *Riemannian Geometry*. Birkhäuser Basel, 1992.

[6] J. M. Lee, *Introduction to Riemannian Manifolds*, 2, Ed. Springer International Publishing, 2018.

[7] P. Petersen, *Riemannian Geometry*. Springer-Verlag New York, 2006.

[8] K. Crane, F. de Goes, M. Desbrun, and P. Schröder, "Digital geometry processing with discrete exterior calculus," in *ACM SIGGRAPH 2013 courses*, 2013.

[9] J. M. Lee, *Introduction to Smooth Manifolds*. Springer-Verlag New York, 2003.

[10] S. Weinberg, *Gravitation and Cosmology: Principles and Applications of the General Theory of Relativity*. John Wiley & Sons, 1972.

[11] D. Hilbert and S. Cohn-Vossen, *Geometry and the Imagination*, 2nd. American Mathematical Society, 1999.

[12] L. C. Evans, *Partial Differential Equations*. American Mathematical Society, 1998.

[13] D. Gilbarg and N. S. Trudinger, *Elliptic Partial Differential Equations of Second Order*, Third. Springer-Verlag Berlin Heidelberg, 2001.

[14] P. D. Lax, *Functional Analysis*. John Wiley & Sons, 2002.

[15] M. Giaquinta and S. Hildebrandt, *Calculus of Variations I*, 1st. Springer-Verlag Berlin Heidelberg, 2004.

[16] U. Pinkall and K. Polthier, "Computing discrete minimal surfaces and their conjugates," *Experiment. Math.*, vol. 2, no. 1, pp. 15–36, 1993.

[17] R. Courant, *Dirichlet's Principle, Conformal Mapping, and Minimal Surfaces*, 1st. Springer-Verlag New York, 1950.

[18] M. Desbrun, M. Meyer, P. Schröder, and A. H. Barr, "Implicit fairing of irregular meshes using diffusion and curvature flow," in *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '99, New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 1999, pp. 317–324.

[19] I. Baran and J. Popovic, "Automatic rigging and animation of 3d characters," *ACM Trans. Graph.*, vol. 26, no. 3, 72–es, Jul. 2007.

[20] P. Joshi, M. Meyer, T. DeRose, B. Green, and T. Sanocki, "Harmonic coordinates for character articulation," *ACM Trans. Graph.*, vol. 26, no. 3, 71–es, Jul. 2007.

[21] O. Weber, O. Sorkine, Y. Lipman, and C. Gotsman, "Context-aware skeletal shape deformation," *Comput. Graph. Forum*, vol. 26, no. 3, pp. 265–274, 2007.

[22] A. Levin, D. Lischinski, and Y. Weiss, "Colorization using optimization," in *ACM SIGGRAPH 2004 Papers*, Los Angeles, California, 2004, pp. 689–694.

[23] R. Courant and D. Hilbert, *Methoden der Mathematischen Physik – Erster Band*. Julius Springer, Berlin, 1924.

[24] F. Knöppel, K. Crane, U. Pinkall, and P. Schröder, "Stripe patterns on surfaces," *ACM Trans. Graph.*, vol. 34, no. 4, Jul. 2015.

[25] ——, "Globally optimal direction fields," *ACM Trans. Graph.*, vol. 32, no. 4, Jul. 2013.

[26] B. Liu, Y. Tong, F. D. Goes, and M. Desbrun, "Discrete connection and covariant derivative for vector field analysis and design," *ACM Trans. Graph.*, vol. 35, no. 3, Mar. 2016.

[27] B. Custers and A. Vaxman, *Subdivision directional fields*, 2018. arXiv: 1810.06884 [cs.GR].

[28] W. Jakob, M. Tarini, D. Panozzo, and O. Sorkine-Hornung, "Instant field-aligned meshes," *ACM Trans. Graph.*, vol. 34, no. 6, Oct. 2015.

[29] N. Sharp, Y. Soliman, and K. Crane, "The vector heat method," *ACM Trans. Graph.*, vol. 38, no. 3, Jun. 2019.

[30] O. Azencot, M. Ovsjanikov, F. Chazal, and M. Ben-Chen, "Discrete derivatives of vector fields on surfaces – an operator approach," *ACM Trans. Graph.*, vol. 34, no. 3, May 2015.

[31] O. Azencot, E. Corman, M. Ben-Chen, and M. Ovsjanikov, "Consistent functional cross field design for mesh quadrangulation," *ACM Trans. Graph.*, vol. 36, no. 4, Jul. 2017.

[32] E. Corman and M. Ovsjanikov, "Functional characterization of deformation fields," *ACM Trans. Graph.*, vol. 38, no. 1, Jan. 2019.

[33] F. Gazzola, H.-C. Grunau, and G. Sweers, *Polyharmonic Boundary Value Problems*. Springer-Verlag Berlin Heidelberg, 2010.

[34] A. Jacobson, T. Weinkauf, and O. Sorkine, "Smooth shape-aware functions with controlled extrema," *Comput. Graph. Forum*, vol. 31, no. 5, pp. 1577–1586, 2012.

[35] A. Jacobson, I. Baran, J. Popović, and O. Sorkine, "Bounded biharmonic weights for real-time deformation," *ACM Trans. Graph.*, vol. 30, no. 4, 78:1–78:8, Jul. 2011.

[36] T. Weinkauf, Y. Gingold, and O. Sorkine, "Topology-based smoothing of 2d scalar fields with c^1-continuity," *Comput. Graph. Forum*, vol. 29, no. 3, pp. 1221–1230, Jun. 2010.

[37] O. Sorkine, D. Cohen-Or, Y. Lipman, M. Alexa, C. Rössl, and H.-P. Seidel, "Laplacian surface editing," in *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, ACM, 2004, pp. 175–184.

[38] K. Zhou, J. Huang, J. Snyder, X. Liu, H. Bao, B. Guo, and H.-Y. Shum, "Large mesh deformation using the volumetric graph Laplacian," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 496–503, Jul. 2005.

[39] J. Andrews, P. Joshi, and N. Carr, "A linear variational system for modeling from curves," *Comput. Graph. Forum*, vol. 30, pp. 1850–1861, 2011.

[40] Y. Lipman, R. M. Rustamov, and T. A. Funkhouser, "Biharmonic distance," *ACM Trans. Graph.*, vol. 29, no. 3, p. 27, 2010.

[41] A. Jacobson, E. Tosun, O. Sorkine, and D. Zorin, "Mixed finite elements for variational surface modeling," *Comput. Graph. Forum*, vol. 29, no. 5, pp. 1565–1574, 2010.

[42] Y. Wang, A. Jacobson, J. Barbič, and L. Kavan, "Linear subspace design for real-time shape deformation," *ACM Trans. Graph.*, vol. 34, no. 4, 2015.

[43] G. Todor, "Photoshop healing brush: A tool for seamless cloning," *Proceedings of the European Conference on Computer Vision*, 2004.

[44] J. Huang, X. Shi, X. Liu, K. Zhou, L.-Y. Wei, S.-H. Teng, H. Bao, B. Guo, and H.-Y. Shum, "Subspace gradient domain mesh deformation," *ACM Trans. Graph.*, vol. 25, no. 3, pp. 1126–1134, 2006.

[45] K. Zhou, W. Xu, Y. Tong, and M. Desbrun, "Deformation transfer to multi-component objects," *Comput. Graph. Forum*, vol. 29, no. 2, 2010.

[46] M. Finch, J. Snyder, and H. Hoppe, "Freeform vector graphics with controlled thin-plate splines," *ACM Trans. Graph.*, vol. 30, no. 6, 6 2011.

[47] R. M. Rustamov, "Multiscale biharmonic kernels," *Comput. Graph. Forum*, vol. 30, pp. 1521–1531, 2011.

[48] O. Weber, R. Poranne, and C. Gotsman, "Biharmonic coordinates," in *Comput. Graph. Forum*, vol. 31, 2012, pp. 2409–2422.

[49] K. Crane, U. Pinkall, and P. Schröder, "Robust fairing via conformal curvature flow," *ACM Trans. Graph.*, vol. 32, no. 4, 2013.

[50] C. Cao, Y. Weng, S. Zhou, Y. Tong, and K. Zhou, "Facewarehouse: A 3d facial expression database for visual computing," *IEEE Trans. Vis. Comput. Graph.*, vol. 20, no. 3, pp. 413–25, 2014.

[51] C. Cao, D. Bradley, K. Zhou, and T. Beeler, "Real-time high-fidelity facial performance capture," *ACM Trans. Graph.*, vol. 34, no. 4, 2015.

[52] A. M. Bronstein, Y. Choukrun, R. Kimmel, and M. Sela, "Consistent discretization and minimization of the l1 norm on manifolds," *Fourth International Conference on 3D Vision (3DV)*, 2016.

[53] F. de Goes, B. Liu, M. Budninskiy, Y. Tong, and M. Desbrun, "Discrete 2-tensor fields on triangulations," *Comput. Graph. Forum*, vol. 33, pp. 13–24, 2014.

[54] E. Tosun, Y. I. Gingold, J. Reisman, and D. Zorin, "Shape optimization using reflection lines," in *Symposium on Geometry Processing*, 2007, pp. 193–202.

[55] M. Fisher, P. Schröder, M. Desbrun, and H. Hoppe, "Design of tangent vector fields," *ACM Trans. Graph.*, vol. 26, no. 3, 2007.

[56] M. Bergou, M. Wardetzky, D. Harmon, D. Zorin, and E. Grinspun, "A quadratic bending model for inextensible surfaces," in *Proceedings of the Fourth Eurographics Symposium on Geometry Processing*, ser. SGP '06, Cagliari, Sardinia, Italy, 2006, pp. 227–230.

[57] F. L. Bookstein, "Principal warps: Thin-plate splines and the decomposition of deformations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 11, no. 6, 1989.

[58] M. Botsch and L. Kobbelt, "Real-time shape editing using radial basis functions," *Comput. Graph. Forum*, vol. 24, no. 3, pp. 611–621, 2005.

[59] S. Didas, "Higher order variational methods for noise removal in signals and images," Master's thesis, Saarland University, 2004.

[60] S. Lefkimmiatis, A. Bourquard, and M. Unser, "Hessian-based norm regularization for image restoration with biomedical applications," *IEEE Transactions on Image Processing*, vol. 21, no. 3, pp. 983–995, 2012.

[61] M. Lysaker and X.-C. Tai, "Iterative image restoration combining total variation minimization and a second-order functional," *Int. J. Comput. Vision*, 2006.

[62] J. M. Roth, "Higher order anisotropic smoothing of images," PhD thesis, Saarland University, 2009.

[63] D. Terzopoulos, "Multilevel reconstruction of visual surfaces: Variational principles and finite-element representations," in *Multiresolution image processing and analysis*, Springer, 1984, pp. 237–310.

[64] S. Didas, J. Weickert, and B. Burgeth, "Properties of higher order nonlinear diffusion filtering," *J. Math. Imaging Vis.*, vol. 35, no. 3, pp. 208–226, 2009.

[65] D. Terzopoulos, "The computation of visible-surface representations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 10, no. 4, pp. 417–438, 1988.

[66] M. Lysaker, A. Lundervold, and X.-C. Tai, "Noise removal using fourth-order partial differential equation with applications to medical magnetic resonance images in space and time," *IEEE Transactions on Image Processing*, vol. 12, no. 12, pp. 1579–1590, 2003.

[67] G. Steidl, S. Didas, and J. Neumann, "Relations between higher order tv regularization and support vector regression," in *International Conference on Scale-Space Theories in Computer Vision*, 2005, pp. 515–527.

[68] G. Steidl, "A note on the dual treatment of higher-order regularization functionals," *Computing*, vol. 76, pp. 135–148, 2006.

[69] J. Yuan, C. Schnörr, and G. Steidl, "Total-variation based piecewise affine regularization," in *International Conference on Scale Space and Variational Methods in Computer Vision*, 2009, pp. 552–564.

[70] Y.-L. You and M. Kaveh, "Fourth-order partial differential equations for noise removal," *IEEE Trans. Image Proc.*, vol. 9, no. 10, pp. 1723–1730, 2000.

[71] M. Belkin and P. Niyogi, "Semi-supervised learning on riemannian manifolds," *Machine Learning*, vol. 56, pp. 209–239, 2004.

[72] D. L. Donoho and C. Grimes, "Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data," *Proc. of the National Academy of Sciences*, vol. 100, no. 10, pp. 5591–5596, 2003.

[73] F. Steinke and M. Hein, "Non-parametric regression between manifolds," in *Advances in Neural Information Processing Systems 21*, Curran Associates, Inc., 2009, pp. 1561–1568.

[74] W. Li, "Free vibrations of beams with general boundary conditions," *Journal of Sound and Vibration*, vol. 237, no. 4, pp. 709–725, 2000.

[75] C. Felippa, *Nonlinear finite element methods*, `www.colorado.edu/engineering/CAS/courses.d/NFEM.d/`, 2007.

[76] R. F. Harik, H. Gong, and A. Bernard, "Review: 5-axis flank milling: A state-of-the-art review," *Comput. Aided Des.*, vol. 45, no. 3, pp. 796–808, Mar. 2013.

[77] L. Chang, *The Software Behind Frank Gehry's Geometrically Complex Architecture*, `https://priceonomics.com/the-software-behind-frank-gehrys-geometrically/`, 2015.

[78] V. Borrelli, S. Jabrane, F. Lazarus, and B. Thibert, "Flat tori in three-dimensional space and convex integration," *Proc. Natl. Acad. Sci. U. S. A.*, vol. 109, no. 19, pp. 7218–7223, 2012.

[79] L. H. Dudte, E. Vouga, T. Tachi, and L Mahadevan, "Programming curvature using origami tessellations," *Nature materials*, 2016.

[80] W. J. Firey, "On the shapes of worn stones," *Mathematika*, vol. 21, no. 1, pp. 1–11, 1974.

[81] B. Andrews, "Gauss curvature flow: The fate of the rolling stones," *Inventiones Mathematicae*, vol. 138, no. 1, pp. 151–161, Oct. 1999.

[82] K. Crane, C. Weischedel, and M. Wardetzky, "Geodesics in heat: A new approach to computing distance based on heat flow," *ACM Trans. Graph.*, vol. 32, no. 5, Oct. 2013.

[83] R. S. Hamilton, "Three-manifolds with positive ricci curvature," *J. Differential Geom.*, vol. 17, no. 2, pp. 255–306, 1982.

[84] I. Eckstein, J.-P. Pons, Y. Tong, C.-C. J. Kuo, and M. Desbrun, "Generalized surface flows for mesh processing," in *Proceedings of the Fifth Eurographics Symposium on Geometry Processing*, 2007, pp. 183–192.

[85] E. Grinspun, A. N. Hirani, M. Desbrun, and P. Schröder, "Discrete shells," in *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, San Diego, California, Jul. 2003, pp. 62–67.

[86] D. Braess, *Finite Elements*. Cambridge University Press, 2007.

[87] M. Wardetzky, S. Mathur, F. Kälberer, and E. Grinspun, "Discrete laplace operators: No free lunch," in *Proceedings of the Fifth Eurographics Symposium on Geometry Processing*, 2007, pp. 33–37.

[88] M. Wardetzky, M. Bergou, D. Harmon, D. Zorin, and E. Grinspun, "Discrete quadratic curvature energies," *Computer Aided Geometric Design*, vol. 24, no. 8, pp. 499–518, Nov. 2007.

[89] R. MacNeal, "The solution of partial differential equations by means of electrical networks," PhD thesis, The California Institute of Technology, 1949.

[90] G. Dziuk, "Finite elements for the beltrami operator on arbitrary surfaces," in *Partial Differential Equations and Calculus of Variations*, S. Hildebrandt and R. Leis, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1988, pp. 142–155.

[91] M. Wardetzky, "Discrete differential operators on polyhedral surfaces - convergence and approximation," PhD thesis, FU Berlin, Nov. 2006.

[92] M. A. Olshanskii, A. Reusken, and J. Grande, "A finite element method for elliptic equations on surfaces," *SIAM J. Numer. Anal.*, vol. 47, no. 5, pp. 3339–3358, Oct. 2009.

[93] Q. Du, L. Ju, and L. Tian, "Finite element approximation of the cahn–hilliard equation on surfaces," *Computer Methods in Applied Mechanics and Engineering*, vol. 200, no. 29, pp. 2458–2470, 2011.

[94] G. Dziuk and C. Elliott, "Surface finite elements for parabolic equations," *Journal of Computational Mathematics*, vol. 25, no. 4, pp. 385–407, 2007.

[95] K. Hildebrandt, K. Polthier, and M. Wardetzky, "On the convergence of metric and geometric properties of polyhedral surfaces," *Geometriae Dedicata*, vol. 123, no. 1, pp. 89–112, Dec. 2006.

[96] M. Crouzeix and P.-A. Raviart, "Conforming and nonconforming finite element methods for solving the stationary stokes equations i," *ESAIM: Mathematical Modelling and Numerical Analysis - Modélisation Mathématique et Analyse Numérique*, vol. 7, no. R3, pp. 33–75, 1973.

[97] E. English and R. Bridson, "Animating developable surfaces using nonconforming elements," *ACM Trans. Graph.*, vol. 27, no. 3, pp. 1–5, Aug. 2008.

[98] K. Hildebrandt and K. Polthier, "Anisotropic filtering of non-linear surface features," *Comput. Graph. Forum*, vol. 23, no. 3, pp. 391–400, 2004.

[99] J. Huang, Y. Tong, H. Wei, and H. Bao, "Boundary aligned smooth 3d cross-frame field," *ACM Trans. Graph.*, vol. 30, no. 6, pp. 1–8, Dec. 2011.

[100] P. Ciarlet and P. Raviart, "A mixed finite element method for the biharmonic equation," *Proceedings of a Symposium Conducted by the Mathematics Research Center, the University of Wisconsin–Madison*, pp. 125–145, Apr. 1973.

[101] P. Monk, "A mixed finite element method for the biharmonic equation," *SIAM J. Numer. Anal.*, vol. 24, no. 4, pp. 737–749, Aug. 1987.

[102] R. Scholz, "A mixed method for 4th order problems using linear finite elements," *RAIRO. Analyse numérique*, vol. 12, no. 1, pp. 85–90, 1978.

[103] D. Boffi, F. Brezzi, and M. Fortin, *Mixed Finite Elements and Applications*. Springer-Verlag Berlin Heidelberg, 2013.

[104] J. A. Nitsche, "L-infinity convergence of the ritz-method with linear finite elements for second order elliptic boundary value problems," *Kompleksnyi analiz i ego prolizheniia: sbornik statei*, pp. 430–437, 1978.

[105] R. Rannacher and R. Scott, "Some optimal error estimates for piecewise linear finite element approximations," *Mathematics of Computation*, vol. 38, no. 158, pp. 437–445, Apr. 1982.

[106] L. R. Herrmann, "Finite-element bending analysis for plates," *Journal of the Engineering Mechanics Division*, vol. 93, no. 5, pp. 13–26, 1967.

[107] F. Brezzi, "Sur la methode des elements finis hybrides pour le probleme biharmonique," *Numer. Math.*, vol. 24, pp. 103–131, 1975.

[108] T. Miyoshi, "Convergence of finite element solutions represented by a non-conforming basis," *Kumamoto J. Sci. (Math.)*, vol. 9, pp. 11–20, 1972.

[109] A. Oukit and R. Pierre, "Mixed finite element for the linear plate problem: The hermann-miyoshi model revisited," *Numer. Math.*, vol. 74, no. 4, pp. 453–477, 1996.

[110] E. Behrens and J. Guzmán, "A mixed method for the biharmonic problem based on a system of first-order equations," *SIAM J. Numerical Analysis*, vol. 49, pp. 789–817, Jan. 2011.

[111] Z. Li and S. Zhang, "A stable mixed element method for the biharmonic equation with first-order function spaces," *Computational Methods in Applied Mathematics*, vol. 17, no. 4, pp. 601–616, 2017.

[112] K. Larsson and M. G. Larson, "A continuous/discontinuous galerkin method and a priori error estimates for the biharmonic problem on surfaces," *Mathematics of Computation*, vol. 86, no. 308, pp. 2613–2649, 2017.

[113] B. Cockburn and A. Demlow, "Hybridizable discontinuous galerkin and mixed finite element methods for elliptic problems on surfaces," *Mathematics of Computation*, vol. 85, no. 302, pp. 2609–2638, 2016.

[114] A. Garg, E. Grinspun, M. Wardetzky, and D. Zorin, "Cubic shells," in *Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, San Diego, California, 2007, pp. 91–98.

[115] E. Tosun, "Geometric modeling with high order derivatives," PhD thesis, New York University, 2008.

[116] P. G. Ciarlet, *The Finite Element Method for Elliptic Problems*. The Society for Industrial and Applied Mathematics, 2002.

[117] A. Stylianou, "Comparison and sign preserving properties of bilaplace boundary value problems in domains with corners," PhD thesis, University of Cologne, 2010.

[118] P. Grisvard, *Elliptic Problems in Nonsmooth Domains*. The Society for Industrial and Applied Mathematics, 2011.

[119] M. Zlámal, "Curved elements in the finite element method," *SIAM J. Numer. Anal.*, vol. 10, no. 1, pp. 229–240, 1973.

[120] R. Scott, "Finite element techniques for curved boundaries," PhD thesis, Massachusetts Institute of Technology, Jun. 1973.

[121] C. Bernardi, "Optimal finite-element interpolation on curved domains," *SIAM J. Numer. Anal.*, vol. 26, no. 5, pp. 1212–1240, Oct. 1989.

[122] S. C. Brenner and L. R. Scott, *The Mathematical Theory of Finite Element Methods*, Third. SpringerScience+BusinessMedia, 2008.

[123] B. Fornberg, "Generation of finite difference formulas on arbitrarily spaced grids," *Mathematics of computation*, 1988.

[124] Y. Wang, A. Jacobson, J. Barbič, and L. Kavan, "Error in "linear subspace design for real-time shape deformation"," Tech. Report, 2017.

[125] K. Crane, "A cotangent laplacian for surfaces with boundary," Tech. Report, 2009.

[126] A. Djerbetian, "Tangent vector fields on triangulated surfaces – an edge-based approach," Master's thesis, Technion, 2016.

[127] Z. Huang and T. Ju, "Extrinsically smooth direction fields," *Computers & Graphics*, vol. 58, pp. 109–117, 2016, Shape Modeling International 2016.

[128] E. Grinspun, M. Desbrun, K. Polthier, P. Schröder, and A. Stern, "Discrete differential geometry: An applied introduction," in *ACM SIGGRAPH 2006 Courses*, ser. SIGGRAPH '06, New York, NY, USA: ACM, 2006.

[129] P. J. Federico, *Descartes on Polyhedra*. Springer-Verlag New York Inc., 1982.

[130] A. Vaxman, *Alternative ways to define discrete gaussian curvature at the boundary, discussion in a libigl github issue*, `https://github.com/libigl/libigl/issues/516`, May 2017.

[131] C. Loop, "Smooth subdivision surfaces based on triangles," Master's thesis, University of Utah, 1987.

[132] E. W. Weisstein, *Monge patch. from mathworld–a wolfram web resource.* Apr. 2019.

[133] G. Guennebaud, B. Jacob, *et al.*, *Eigen v3*, http://eigen.tuxfamily.org, 2010.

[134] MATLAB, *version 9.6 (R2019a)*. The MathWorks Inc., 2019.

[135] A. Jacobson, D. Panozzo, *et al.*, *libigl: A simple C++ geometry processing library*, `http://libigl.github.io/libigl/`, 2019.

[136] A. Jacobson, *gptoolbox – geometry processing toolbox*, `https://github.com/alecjacobson/gptoolbox`, 2019.

[137] J. Mitani and H. Suzuki, "Making papercraft toys from meshes using strip-based approximate unfolding," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 259–263, Aug. 2004.

[138] B. Thibert, J.-P. Gratier, and J.-M. Morvan, "A direct method for modeling and unfolding developable surfaces and its application to the ventura basin (california)," *Journal of Structural Geology*, vol. 27, no. 2, pp. 303–316, 2005.

[139] Y. Liu, H. Pottmann, J. Wallner, Y.-L. Yang, and W. Wang, "Geometric modeling with conical meshes and developable surfaces," *ACM Trans. Graph.*, vol. 25, no. 3, pp. 681–689, Jul. 2006.

[140] M. Rabinovich, T. Hoffmann, and O. Sorkine-Hornung, "Discrete geodesic nets for modeling developable surfaces," *ACM Trans. Graph.*, vol. 37, no. 2, Feb. 2018.

[141] R. Sauer, *Differenzengeometrie*. Springer, 1970.

[142] Y.-L. Yang, Y.-J. Yang, H. Pottmann, and N. J. Mitra, "Shape space exploration of constrained meshes," *ACM Trans. Graph.*, vol. 30, no. 6, 124:1–124:12, Dec. 2011.

[143] M. Peternell, "Developable surface fitting to point clouds," *Comput. Aided Geom. Des.*, vol. 21, no. 8, pp. 785–803, Oct. 2004.

[144] J. Berkmann and T. Caelli, "Computation of surface geometry and segmentation using covariance techniques," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 11, pp. 1114–1116, Nov. 1994.

[145] M. Garland and P. S. Heckbert, "Surface simplification using quadric error metrics," in *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '97, 1997, pp. 209–216.

[146] M. Garland, "Quadric-based polygonal surface simplification," PhD thesis, Carnegie Mellon University, 1999.

[147] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd. Springer, 2006.

[148] A. S. Lewis and M. L. Overton, "Nonsmooth optimization via quasi-newton methods," *Math. Program.*, vol. A, no. 141, pp. 135–163, 2013.

[149] J. Kopp, "Efficient Numerical Diagonalization of Hermitian 3×3 Matrices," *International Journal of Modern Physics C*, vol. 19, pp. 523–548, 2008.

[150] J. Erickson and S. Har-Peled, "Optimally cutting a surface into a disk," *Discrete & Computational Geometry*, vol. 31, no. 1, pp. 37–59, 2004.

[151] B. Lévy, S. Petitjean, N. Ray, and J. Maillot, "Least squares conformal maps for automatic texture atlas generation," in *Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '02, 2002, pp. 362–371.

[152] A. Sheffer and E. de Sturler, "Parameterization of faceted surfaces for meshing using angle-based flattening," *Engineering with Computers*, vol. 17, no. 3, pp. 326–337, 2001.

[153] R. Sawhney and K. Crane, "Boundary first flattening," *ACM Trans. Graph.*, vol. 37, no. 1, Dec. 2017.

[154] O. Sorkine, D. Cohen-Or, R. Goldenthal, and D. Lischinski, "Bounded-distortion piecewise mesh parameterization," in *Proceedings of the Conference on Visualization '02*, 2002, pp. 355–362.

[155] R. Poranne, M. Tarini, S. Huber, D. Panozzo, and O. Sorkine-Hornung, "Autocuts: Simultaneous distortion and cut optimization for uv mapping," *ACM Trans. Graph.*, vol. 36, no. 6, Nov. 2017.

[156] C. Tang, P. Bo, J. Wallner, and H. Pottmann, "Interactive design of developable surfaces," *ACM Trans. Graph.*, vol. 35, no. 2, Jan. 2016.

[157] C.-H. Chu and J.-T. Chen, "Tool path planning for five-axis flank milling with developable surface approximation," *The International Journal of Advanced Manufacturing Technology*, vol. 29, no. 7, Oct. 2005.

[158] M. Kilian, S. Flöry, Z. Chen, N. J. Mitra, A. Sheffer, and H. Pottmann, "Curved folding," *ACM Trans. Graph.*, vol. 27, no. 3, Aug. 2008.

[159] A. I. Bobenko and P. Schröder, "Discrete Willmore Flow," in *Eurographics Symposium on Geometry Processing 2005*, M. Desbrun and H. Pottmann, Eds., 2005.

[160] R. Perl, P. Pozzi, and M. Rumpf, *A nested variational time discretization for parametric anisotropic willmore flow*, 2015. arXiv: 1503.06971 [math.NA].

[161] K. Crane and M. Wardetzky, "A glimpse into discrete differential geometry," *Notices of the American Mathematical Society*, vol. 64, no. 10, pp. 1153–1159, Nov. 2017.

[162] A. I. Bobenko, P. Schröder, J. M. Sullivan, and G. M. Ziegler, *Discrete Differential Geometry*. Birkhäuser Basel, 2008.

[163] H. Pottmann, A. Schiftner, P. Bo, H. Schmiedhofer, W. Wang, N. Baldassini, and J. Wallner, "Freeform surfaces from single curved panels," *ACM Trans. Graph.*, vol. 27, no. 3, Aug. 2008.

[164] J. Solomon, E. Vouga, M. Wardetzky, and E. Grinspun, "Flexible developable surfaces," *Comput. Graph. Forum*, vol. 31, no. 5, pp. 1567–1576, Aug. 2012.

[165] C. C. Wang and K. Tang, "Achieving developability of a polygonal surface by minimum deformation: A study of global and local optimization approaches," *The Visual Computer*, vol. 20, no. 8, pp. 521–539, 2004.

[166] I. Shatz, A. Tal, and G. Leifman, "Paper craft models from meshes," *The Visual Computer*, vol. 22, no. 9-11, pp. 825–834, 2006.

[167] F. Massarwi, C. Gotsman, and G. Elber, "Papercraft models using generalized cylinders," in *Proceedings of the Pacific Conference on Computer Graphics and Applications, Pacific Graphics 2007*, 2007, pp. 148–157.

[168] D. Julius, V. Kraevoy, and A. Sheffer, "D-charts: Quasi-developable mesh segmentation," *Comput. Graph. Forum*, vol. 24, no. 3, pp. 581–590, 2005.

[169] K. Rose, A. Sheffer, J. Wither, M.-P. Cani, and B. Thibert, "Developable surfaces from arbitrary sketched boundaries," in *Proceedings of the Fifth Eurographics Symposium on Geometry Processing*, 2007, pp. 163–172.

[170] R. Narain, T. Pfaff, and J. F. O'Brien, "Folding and crumpling adaptive sheets," *ACM Trans. Graph.*, vol. 32, no. 4, Jul. 2013.

[171] C. Schreck, D. Rohmer, S. Hahmann, M.-P. Cani, S. Jin, C. C. Wang, and J.-F. Bloch, "Non-smooth developable geometry for interactively animating paper crumpling," *ACM Trans. Graph.*, vol. 35, no. 1, Dec. 2015.

[172] K. A. Brakke, "The surface evolver," *Experiment. Math.*, vol. 1, no. 2, pp. 141–165, 1992.

[173] H. Zhao and G. Xu, "Triangular surface mesh fairing via gaussian curvature flow," *J. Comput. Appl. Math.*, vol. 195, no. 1, pp. 300–311, Oct. 2006.

[174] M. Jin, J. Kim, F. Luo, and X. Gu, "Discrete surface ricci flow," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 5, pp. 1030–1043, Sep. 2008.

[175] M. Kazhdan, J. Solomon, and M. Ben-Chen, "Can mean-curvature flow be modified to be non-singular?" *Comput. Graph. Forum*, vol. 31, no. 5, pp. 1745–1754, Aug. 2012.

[176] smkmuseum, *Venus de milo (aphrodite of milos)*, https://www.thingiverse.com/thing:1892710, Nov. 2016.

[177] jansentee3d, *Dragon tower*, https://www.thingiverse.com/thing:3155868, Oct. 2018.

[178] Javo, *Big gladiator helmet*, https://www.thingiverse.com/thing:1345281, Feb. 2016.

[179] K. Crane, *Keenan's 3d model repository*, https://www.cs.cmu.edu/~kmcrane/Projects/ModelRepository/, 2020.

[180] YahooJAPAN, *Plane*, https://www.thingiverse.com/thing:182252, Nov. 2013.

[181] Stanford Graphics, *The stanford 3d scanning repository*, http://graphics.stanford.edu/data/3Dscanrep/, 2014.

[182] I. Sutherland, *Beetle car mesh*, provided by https://github.com/alecjacobson/common-3d-test-models, 1972.

[183] NIST, *SHREC 2009 dataset*, https://www.nist.gov/itl/iad/shrec-2009-datasets, 2009.

[184] SpinQuad1976, *Realistic chimpanzee*, https://www.turbosquid.com/FullPreview/Index.cfm/ID/447191, 2016.

[185] N. Al-Badri and J. N. Nelles, *Nefertiti mesh*, provided by https://www.cs.cmu.edu/~kmcrane/Projects/ModelRepository/, 2016.

[186] A. Jacobson, D. Panozzo, *et al.*, *Libigl tutorial data*, https://github.com/libigl/libigl-tutorial-data, 2019.

[187] K. Crane, *Meshes provided privately*, 2018.

[188] willie, *Rubber duck*, https://www.thingiverse.com/thing:139894, Aug. 2013.

[189] Autodesk, *Swingarm model*, 2018.

[190] chachan, *Drill 3d model*, https://free3d.com/3d-model/drill--57786.html, Oct. 2014.

[191] A. Jacobson, "Algorithms and interfaces for real-time deformation of 2d and 3d shapes," PhD thesis, ETH Zürich, 2013.

[192] ——, *Cheeseman cartoon*, Privately provided, 2018.

[193] K. Crane, *Images provided privately*, 2018.

[194] E. D. Andersen and K. D. Andersen, "The MOSEK interior point optimizer for linear programming: An implementation of the homogeneous algorithm," in *High Performance Optimization*, Kluwer Academic Publishers, 2000, pp. 197–232.

Appendix A: A discretization of the L^1 Hessian energy

The L^1 Hessian energy from Definition 3.2 can be minimized by introducing an auxiliary matrix-valued variable equal to the element-wise absolute value of the Hessian $Y = |H_u|$ and solving the constrained optimization problem:

$$\begin{aligned} \underset{u,Y}{\text{argmin}} \quad & \int_\Omega \mathbf{1}^\top Y \mathbf{1} \mathrm{d}x \\ \text{subject to} \quad & H_u \leq Y,\ H_u \geq -Y,\ \text{and } Y \geq 0,\ , \end{aligned}$$

where $\mathbf{1}$ is a vector of ones with appropriate length.

Using our discrete matrices from Section 6.1.2, this becomes a linear program

$$\begin{aligned} \underset{\mathtt{u},\mathtt{Y}}{\text{argmin}} \quad & \mathbf{1}^\top \mathbf{M} \mathtt{Y} \\ \text{subject to} \quad & \mathbf{H}\mathtt{u} \geq -\mathtt{Y} \\ & \mathbf{H}\mathtt{u} \leq \mathtt{Y} \\ & \mathtt{Y} \geq 0\ , \end{aligned}$$

where $\mathtt{Y} \in \mathbb{R}^{4\hat{n}}$ is a vectorization of per-vertex 2×2 matrices on a mesh with $\hat{n}$ interior vertices, $\mathtt{u}$ is the finite element degrees of freedom vector, and $\mathbf{H} = \mathbf{D}^\top \mathbf{A} \mathbf{G}$.

When combined with other quadratic energies such as a data-term, this transforms into a quadratic program. The results shown in this thesis have been computed with the quadratic program solver Mosek [194].

Appendix B: An L^∞ Bound for the Linear FEM Discretization of the Laplace Equation

In this section we prove an important lemma that is used in the proof of Theorem 5.3. It corresponds to Lemma 8.2.6 by Brenner and Scott [122], but the proof follows the simpler approach by Rannacher and Scott [105], since we only deal with basic piecewise linear finite elements. The proof is very similar with small adjustments to account for the fact that the calculation happens on curved surfaces. We use the terminology and definitions of Section 5.2.3 in this appendix.

We start by stating a few basic facts about our weight function.

Lemma B.1. *For the weight function $\sigma_z(x) = \sqrt{d(x,z)^2 + \kappa^2 h^2}$ from (5.29), we have*

$$\begin{aligned} \left| D^k \sigma_z^j \right| &\leq C \sigma_z^{j-k} \\ \left| D^k \sigma_z^j \right| &\leq C \left(1 + (\kappa h)^{j-k} \right) , \end{aligned} \tag{B.1}$$

where D^k is the vector of all partial derivatives of order k. The constant C smoothly depends on j and k. For fixed κ and $j - k < 0$, the bound is just $C(\kappa h)^{j-k}$, since there is always a h small enough, and "for small enough h" is implied everywhere.

Additionally, let v be a continuous function in H^1, and let it be H^2 within triangles, i.e., the summed per-triangle H^2 norm is bounded. Let $\alpha \in \mathbb{R}$. We have

$$\int_\Omega \sigma_z^\alpha |\nabla (v - I_h v)|^2 \, \mathrm{d}x \leq Ch^2 \widetilde{\int_\Omega} \sigma_z^\alpha \left|D^2 v\right|^2 \, \mathrm{d}x \,, \tag{B.2}$$

where $\widetilde{\int}$ denotes integration over the interior of each triangle, summed up. The constant C smoothly depends on α (for large enough $\kappa > 0$). If α is uniformly bounded, so is C.

Proof. (B.1) is clear from the definition of σ_z.

For (B.2), we first notice that for every triangle T in our triangulation, since the surface is compact, and $\kappa, h > 0$,

$$\sup_{x \in T} \sigma_z^\alpha(x) / \inf_{x \in T} \sigma_z^\alpha(x) \leq C \,, \tag{B.3}$$

with C independent of the triangle or the triangulation [122, (8.1.4)]. Using (5.22) from Lemma 5.4 per triangle, this proves the lemma. □

A helpful tool that will be used is Young's inequality for products [122, (0.9.5)]. For any $a, b \in \mathbb{R}$, $\delta > 0$

$$ab \leq \frac{\delta}{2} a^2 + \frac{1}{2\delta} b^2 \,.$$

We will use it with $\delta = 2C$ to get $Cab \leq C^2 a^2 + \frac{1}{4} b^2$. This trick allows us to separate a product into a sum, while making one summand so small that it can later be moved to the left side of an inequality.

The main goal of this appendix is bounding $\int_\Omega \left|\nabla\left(g^z - g_h^z\right)\right|^2 \sigma_z^{2+\lambda}\ \mathrm{d}x$. This will be done in a sequence of consecutive lemmas.

Lemma B.2. *For sufficiently large $\kappa > 0$,*

$$\int_\Omega \left|\left|\nabla\left(g^z - g_h^z\right)\right|^2 \sigma_z^{2+\lambda}\right| \mathrm{d}x \leq C \widetilde{\int_\Omega} h^2 \sigma_z^{2+\lambda} \left|D^2\left(g^z - g_h^z\right)\right|^2 + \sigma_z^\lambda \left(g^z - g_h^z\right)^2\ \mathrm{d}x\,.$$

This is very similar to a result by Rannacher and Scott [105], but in their version there is no $D^2 g_h^z$ on the right-hand side of the inequality, since in the flat case that term is zero.

Proof. Let $\psi = \sigma^{2+\lambda}\left(g^z - g_h^z\right)$. Then we have

$$\begin{aligned}
\int_\Omega \left|\left|\nabla\left(g^z - g_h^z\right)\right|^2 \sigma_z^{2+\lambda}\right| \mathrm{d}x &= D\left\langle g^z - g_h^z, \psi\right\rangle - \int_\Omega \left(g^z - g_h^z\right) \nabla\left(g^z - g_h^z\right) \cdot \nabla\left(\sigma_z^{2+\lambda}\right)\ \mathrm{d}x \\
&= D\left\langle g^z - g_h^z, \psi\right\rangle - \frac{1}{2}\int_\Omega \nabla\left(\left(g^z - g_h^z\right)^2\right) \cdot \nabla\left(\sigma_z^{2+\lambda}\right) \qquad \text{(B.4)} \\
&= D\left\langle g^z - g_h^z, \psi\right\rangle - \frac{1}{2}\int_\Omega \left(g^z - g_h^z\right)^2 \Delta\left(\sigma_z^{2+\lambda}\right)\ \mathrm{d}x\,.
\end{aligned}$$

Using the fact that g^z and g_h^z solve the Poisson equation and the discrete Poisson equation, respectively, we have $D\left\langle g^z - g_h^z, I_h\psi\right\rangle = 0$. Using Young's and Hölder's inequalities, this gives

$$\begin{aligned}
\left|D\left\langle g^z - g_h^z, \psi\right\rangle\right| = \left|D\left\langle g^z - g_h^z, \psi - I_h\psi\right\rangle\right| &\leq C\int_\Omega \left|\nabla\left(g^z - g_h^z\right)\right| \left|\nabla\left(\psi - I_h\psi\right)\right| \mathrm{d}x \\
&\leq C\left(\int_\Omega \sigma_z^{2+\lambda}\left|\nabla\left(g^z - g_h^z\right)\right|^2 \mathrm{d}x\right)^{\frac{1}{2}} \left(\int_\Omega \sigma_z^{-2-\lambda}\left|\nabla\left(\psi - I_h\psi\right)\right|^2 \mathrm{d}x\right)^{\frac{1}{2}} \\
&\leq \frac{1}{4}\int_\Omega \sigma_z^{2+\lambda}\left|\nabla\left(g^z - g_h^z\right)\right|^2 \mathrm{d}x + C^2\int_\Omega \sigma_z^{-2-\lambda}\left|\nabla\left(\psi - I_h\psi\right)\right|^2 \mathrm{d}x\,.
\end{aligned}$$

We continue, using (B.2) (and a new constant C),

$$\begin{aligned}
\left| D\left\langle g^z - g_h^z, \psi \right\rangle \right| &\leq \frac{1}{4} \int_\Omega \sigma_z^{2+\lambda} \left| \nabla \left(g^z - g_h^z \right) \right|^2 \, \mathrm{d}x + C \int_\Omega \sigma_z^{-2-\lambda} |\nabla \left(\psi - I_h \psi \right)|^2 \, \mathrm{d}x \\
&\leq \frac{1}{4} \int_\Omega \sigma_z^{2+\lambda} \left| \nabla \left(g^z - g_h^z \right) \right|^2 \, \mathrm{d}x + Ch^2 \widetilde{\int_\Omega} \sigma_z^{-2-\lambda} \left| D^2 \psi \right|^2 \, \mathrm{d}x \, , \\
\widetilde{\int_\Omega} \sigma_z^{-2-\lambda} \left| D^2 \psi \right|^2 \, \mathrm{d}x &\leq \widetilde{\int_\Omega} \sigma_z^{-2-\lambda} \left(\sigma_z^{2+\lambda} \left| D^2 \left(g^z - g_h^z \right) \right| + 2 \left| \nabla \sigma_z^{2+\lambda} \right| \left| \nabla \left(g^z - g_h^z \right) \right| + \left| D^2 \sigma_z^{2+\lambda} \right| \left(g^z - g_h^z \right) \right)^2 \, \mathrm{d}x \\
&\leq C \widetilde{\int_\Omega} \sigma_z^{2+\lambda} \left| D^2 \left(g^z - g_h^z \right) \right|^2 + \sigma_z^\lambda \left| \nabla \left(g^z - g_h^z \right) \right|^2 + \sigma_z^{\lambda-2} \left(g^z - g_h^z \right)^2 \, \mathrm{d}x \, .
\end{aligned}$$

We now use (B.1) to bound σ_z and its derivatives,

$$\begin{aligned}
\widetilde{\int_\Omega} \sigma_z^{-2-\lambda} \left| D^2 \psi \right|^2 \, \mathrm{d}x &\leq C \widetilde{\int_\Omega} \sigma_z^{2+\lambda} \left| D^2 \left(g^z - g_h^z \right) \right|^2 + (\kappa h)^{-2} \sigma_z^{2+\lambda} \left| \nabla \left(g^z - g_h^z \right) \right|^2 + (\kappa h)^{-2} \sigma_z^\lambda \left(g^z - g_h^z \right)^2 \, \mathrm{d}x \, , \\
\left| D\left\langle g^z - g_h^z, \psi \right\rangle \right| &\leq Ch^2 \widetilde{\int_\Omega} \sigma_z^{2+\lambda} \left| D^2 \left(g^z - g_h^z \right) \right|^2 \, \mathrm{d}x + \left(C\kappa^{-2} + \frac{1}{4} \right) \int_\Omega \sigma_z^{2+\lambda} \left| \nabla \left(g^z - g_h^z \right) \right|^2 \, \mathrm{d}x \\
&\quad + C\kappa^{-2} \int_\Omega \sigma_z^\lambda \left(g^z - g_h^z \right)^2 \, \mathrm{d}x \, .
\end{aligned}$$

This bounds the first term on the right side of the last line of (B.4), provided that κ is large enough.

By (B.1),

$$\left| \int_\Omega \left(g^z - g_h^z \right)^2 \Delta \left(\sigma_z^{2+\lambda} \right) \, \mathrm{d}x \right| \leq C \int_\Omega \left(g^z - g_h^z \right)^2 \sigma_z^\lambda \, \mathrm{d}x \, .$$

This bounds the second term on the right side of the last line of (B.4), and proves the lemma. □

The first term on the right side of Lemma B.2 contains the second derivatives of a function, weighted by powers of the function σ_z. The following two lemmas provide a bound for this quantity.

Lemma B.3. *There is $\lambda > 0$ small enough such that,*

$$\int_\Omega \sigma_z^{2+\lambda} \left|D^2 g^z\right|^2 \,\mathrm{d}x \le C \int_\Omega \sigma_z^{2+\lambda} |\nabla \delta^z|^2 + \lambda^{-1} (\kappa h)^{-2} \sigma_z^{2+\lambda} |\delta^z|^2 \,\mathrm{d}x \,,$$

where δ^z was defined in (5.23).

Proof. Using the product rule and (B.1) we have

$$\begin{aligned} \int_\Omega \sigma_z^{2+\lambda} \left|D^2 g^z\right|^2 \,\mathrm{d}x &\le C \int_\Omega \left| D^2 \left(\sigma_z^{\frac{2+\lambda}{2}} g^z \right) \right|^2 \,\mathrm{d}x + C \int_\Omega \sigma_z^\lambda |\nabla g^z|^2 + \sigma_z^{\lambda-2} (g^z)^2 \,\mathrm{d}x \\ &\le C \int_\Omega \left| \Delta \left(\sigma_z^{\frac{2+\lambda}{2}} g^z \right) \right|^2 \,\mathrm{d}x + C \int_\Omega \sigma_z^\lambda |\nabla g^z|^2 + \sigma_z^{\lambda-2} (g^z)^2 \,\mathrm{d}x \,. \end{aligned} \tag{B.5}$$

In the last step, we applied the regularity estimate $\|v\|_{H^2} \le C \|\Delta v\|_{L^2}$ to $\sigma_z^{\frac{2+\lambda}{2}} g^z$. Let $\xi \in H_0^1$. We now use the definition of g^z from (5.27), integration by parts, (5.25), and (B.1) to obtain

$$\begin{aligned} \left\langle \xi, \Delta \left(\sigma_z^{\frac{2+\lambda}{2}} g^z \right) \right\rangle &= D \left\langle \xi, \sigma_z^{\frac{2+\lambda}{2}} g^z \right\rangle = \int_\Omega \nabla \xi \cdot \nabla g^z \sigma_z^{\frac{2+\lambda}{2}} + \nabla \xi \cdot \nabla \left(\sigma_z^{\frac{2+\lambda}{2}} \right) g^z \,\mathrm{d}x \\ &= D \left\langle \xi \sigma_z^{\frac{2+\lambda}{2}}, g^z \right\rangle + \int_\Omega \nabla \xi \cdot \nabla \left(\sigma_z^{\frac{2+\lambda}{2}} \right) g^z - \xi \nabla \left(\sigma_z^{\frac{2+\lambda}{2}} \right) \cdot \nabla g^z \,\mathrm{d}x \\ &= \left\langle \xi \sigma_z^{\frac{2+\lambda}{2}}, -\nu \cdot \nabla \delta^z \right\rangle + \int_\Omega \xi \Delta \left(\sigma_z^{\frac{2+\lambda}{2}} \right) g^z - 2 \xi \nabla \left(\sigma_z^{\frac{2+\lambda}{2}} \right) \cdot \nabla g^z \,\mathrm{d}x \\ &\le C \|\xi\|_{L^2} \left(\left\| \sigma_z^{\frac{2+\lambda}{2}} \nabla \delta^z \right\|_{L^2} + \left\| \sigma_z^{\frac{\lambda}{2}} \nabla g^z \right\|_{L^2} + \left\| \sigma_z^{\frac{\lambda-2}{2}} g^z \right\|_{L^2} \right) \,. \end{aligned}$$

Since H_0^1 is L^2-dense in L^2, this gives us a bound for the L^2 norm of $\Delta \left(\sigma_z^{\frac{2+\lambda}{2}} g^z \right)$.

Plugging this into (B.5) gives

$$\begin{aligned}\int_\Omega \left|\Delta\left(\sigma_z^{\frac{2+\lambda}{2}} g^z\right)\right|^2 \,\mathrm{d}x &\leq C\int_\Omega \sigma_z^{2+\lambda}|\nabla\delta^z|^2 + \sigma_z^\lambda|\nabla g^z|^2 + \sigma_z^{\lambda-2}(g^z)^2 \,\mathrm{d}x\,,\\ \int_\Omega \sigma_z^{2+\lambda}\left|D^2 g^z\right|^2 \,\mathrm{d}x &\leq C\int_\Omega \sigma_z^{2+\lambda}|\nabla\delta^z|^2 \,\mathrm{d}x + C\int_\Omega \sigma_z^\lambda|\nabla g^z|^2 + \sigma_z^{\lambda-2}(g^z)^2 \,\mathrm{d}x\,.\end{aligned} \tag{B.6}$$

Continuing the calculation, we obtain

$$\begin{aligned}\int_\Omega \sigma_z^\lambda|\nabla g^z|^2 \,\mathrm{d}x &= \int_\Omega \nabla g^z\cdot\nabla\left(\sigma_z^\lambda g^z\right) - \nabla g^z\cdot\left(\nabla\sigma_z^\lambda\right)g^z \,\mathrm{d}x \qquad \text{(product rule)},\\ \int_\Omega \nabla g^z\cdot\left(\nabla\sigma_z^\lambda\right)g^z \,\mathrm{d}x &= \int_\Omega (g^z)^2\left(\Delta\sigma_z^\lambda\right) - g^z\nabla g^z\cdot\left(\nabla\sigma_z^\lambda\right)\,\mathrm{d}x \qquad \text{(integration by parts)},\\ \int_\Omega \sigma_z^\lambda|\nabla g^z|^2 \,\mathrm{d}x &= \int_\Omega \nabla g^z\cdot\nabla\left(\sigma_z^\lambda g^z\right) - \frac{1}{2}(g^z)^2\left(\Delta\sigma_z^\lambda\right)\,\mathrm{d}x\\ &\leq C\int_\Omega \nabla g^z\cdot\nabla\left(\sigma_z^\lambda g^z\right) + (g^z)^2\sigma_z^{\lambda-2} \qquad \text{(B.1)}\\ &\leq C\int_\Omega -\nu\cdot\nabla\delta^z\sigma_z^\lambda g^z + (g^z)^2\sigma_z^{\lambda-2}\,\mathrm{d}x \qquad \text{(5.27)}.\end{aligned} \tag{B.7}$$

After using (5.25), we can further bound $\int_\Omega \nu\cdot\nabla\delta^z\sigma_z^\lambda g^z \,\mathrm{d}x$ using Young's and Hölder's inequalities,

$$\begin{aligned}\left|\int_\Omega \nu\cdot\nabla\delta^z\sigma_z^\lambda g^z \,\mathrm{d}x\right| &\leq C\int_\Omega |\nabla\delta^z|\sigma_z^\lambda g^z \,\mathrm{d}x \leq C\left(\int_\Omega |\nabla\delta^z|^2\sigma_z^{\lambda+2}\,\mathrm{d}x\right)^{\frac{1}{2}}\left(\int_\Omega (g^z)^2\sigma_z^{\lambda-2}\,\mathrm{d}x\right)^{\frac{1}{2}}\\ &\leq C\int_\Omega |\nabla\delta^z|^2\sigma_z^{\lambda+2} + (g^z)^2\sigma_z^{\lambda-2}\,\mathrm{d}x\,,\end{aligned}$$

which, plugged back into (B.7) and then (B.6), gives

$$\int_\Omega \sigma_z^{\lambda+2}\left|D^2 g^z\right|^2 \,\mathrm{d}x \leq C\int_\Omega |\nabla\delta^z|^2\sigma_z^{\lambda+2} + (g^z)^2\sigma_z^{\lambda-2}\,\mathrm{d}x\,. \tag{B.8}$$

We now use Hölder's inequality as well as (5.30) to obtain

$$\begin{aligned}\int_\Omega (g^z)^2\,\sigma_z^{\lambda-2}\,\mathrm{d}x &\le \left(\int_\Omega \sigma_z^{-\lambda-2}\,\mathrm{d}x\right)^{\frac{\lambda-2}{-\lambda-2}}\left(\int_\Omega |g^z|^{\frac{\lambda+2}{\lambda}}\,\mathrm{d}x\right)^{\frac{2\lambda}{\lambda+2}}\\ &\le C\left(\lambda^{-1}(\kappa h)^{-\lambda}\right)^{\frac{\lambda-2}{-\lambda-2}}\left(\int_\Omega |g^z|^{\frac{\lambda+2}{\lambda}}\,\mathrm{d}x\right)^{\frac{2\lambda}{\lambda+2}}\,.\end{aligned} \tag{B.9}$$

This application of Hölder's inequality is allowed for appropriate choices of $\lambda > 0$.

Let now $w \in H_0^1 \cap H^2$ such that $\Delta w = \operatorname{sign}(g^z)|g^z|^{\frac{\lambda+2}{\lambda}-1}$. We use the PDEs for g^z (found in (5.27)) and w, integration by parts, as well as (5.26) to get

$$\int_\Omega |g^z|^{\frac{\lambda+2}{\lambda}}\,\mathrm{d}x = \left\langle \operatorname{sign}(g^z)|g^z|^{\frac{\lambda+2}{\lambda}-1}, g^z\right\rangle = D\langle w, g^z\rangle = \langle w, -\nu\cdot\nabla\delta^z\rangle = \langle \nu\cdot\nabla w, \delta^z\rangle\,.$$

The last equality follows because ν was defined carefully to be divergence-free.

To simplify notation, let $\|w\|_{k,p}$ be the $W^{k,p}$ norm, and $\|w\|_p$ the L^p norm. Using the Hölder and Sobolev [13, Theorem 7.10] inequalities, (5.25), (5.30), as well as the standard regularity estimate for the Poisson equation, we have

$$\begin{aligned}\int_\Omega |g^z|^{\frac{\lambda+2}{\lambda}}\,\mathrm{d}x &\le C\|w\|_{1,\frac{4+2\lambda}{2-\lambda}}\|\delta^z\|_{\frac{4+2\lambda}{2+3\lambda}} \le C\|w\|_{2,\frac{2+\lambda}{2}}\|\delta^z\|_{\frac{4+2\lambda}{2+3\lambda}} \le C\|g^z\|_{\frac{2+\lambda}{\lambda}}^{\frac{2}{\lambda}}\|\delta^z\|_{\frac{4+2\lambda}{2+3\lambda}}\,,\\ \|g^z\|_{\frac{2+\lambda}{\lambda}} &\le C\|\delta^z\|_{\frac{4+2\lambda}{2+3\lambda}} \le C\left(\int_\Omega \sigma_z^{\lambda+2}|\delta^z|^2\,\mathrm{d}x\right)^{\frac12}\left(\int_\Omega \sigma_z^{-\frac{(\lambda+2)^2}{2\lambda}}\,\mathrm{d}x\right)^{\frac{\lambda}{2+\lambda}}\\ &\le C\left(\lambda^{-1}\,(\kappa h)^{-\frac{\lambda^2+4}{2\lambda}}\right)^{\frac{\lambda}{2+\lambda}}\left(\int_\Omega \sigma_z^{\lambda+2}|\delta^z|^2\,\mathrm{d}x\right)^{\frac12}\,,\end{aligned}$$

Plugged into (B.9) this gives

$$\int_\Omega (g^z)^2 \, \sigma_z^{\lambda-2} \, \mathrm{d}x \leq C \left(\lambda^{-1}(\kappa h)^{-\lambda}\right)^{\frac{\lambda-2}{-\lambda-2}} \left(\lambda^{-1}(\kappa h)^{-\frac{\lambda^2+4}{2\lambda}}\right)^{\frac{2\lambda}{2+\lambda}} \int_\Omega \sigma_z^{\lambda+2} |\delta^z|^2 \, \mathrm{d}x$$
$$\leq C\lambda^{-1}(\kappa h)^{-2} \int_\Omega \sigma_z^{\lambda+2} |\delta^z|^2 \, \mathrm{d}x \, .$$

Now, plugging this into (B.8) gives

$$\int_\Omega \sigma_z^{\lambda+2} \left|D^2 g^z\right|^2 \, \mathrm{d}x \leq C \int_\Omega |\nabla \delta^z|^2 \sigma_z^{\lambda+2} + \lambda^{-1}(\kappa h)^{-2} \, \sigma_z^{\lambda+2} |\delta^z|^2 \, \mathrm{d}x \, .$$

□

Lemma B.4. *There is $\lambda > 0$ small enough such that,*

$$\int_\Omega \sigma_z^{2+\lambda} |\nabla g^z|^2 \, \mathrm{d}x \leq C \int_\Omega \sigma_z^{2+\lambda} |\nabla \delta^z|^2 + \lambda^{-1}(\kappa h)^{-2} \, \sigma_z^{2+\lambda} |\delta^z|^2 \, \mathrm{d}x \, .$$

Proof. Using the product rule, (B.1), and the standard regularity estimate,

$$\int_\Omega \sigma_z^{2+\lambda} |\nabla g^z|^2 \, \mathrm{d}x \leq C \int_\Omega \left|\nabla \left(\sigma^{\frac{2+\lambda}{\lambda}} g^z\right)\right|^2 + \sigma^\lambda |g^z|^2 \, \mathrm{d}x \leq C \int_\Omega \left|\Delta \left(\sigma^{\frac{2+\lambda}{\lambda}} g^z\right)\right|^2 + \sigma^\lambda |g^z|^2 \, \mathrm{d}x$$

Because the surface Ω is compact, for λ small enough we have $\sigma^\lambda(x) \leq C\sigma(x)^{\lambda-2}$ pointwise. Thus $\int_\Omega \sigma_z^{2+\lambda} |\nabla g^z|^2 \, \mathrm{d}x$ is bounded by the same expression as the right-hand side of (B.5), and Lemma B.3 implies the result of this lemma. □

Lemma B.5. *Let $v, w \in H_0^1$ such that*

$$\Delta v = w \, .$$

Then, for $\lambda > 0$ small enough,

$$\int_\Omega \sigma_z^{-2-\lambda} \left|D^2 v\right|^2 \,\mathrm{d}x \leq C\lambda^{-1} (\kappa h)^{-2} \int_\Omega \sigma_z^{2-\lambda} |\nabla w|^2 \,\mathrm{d}x$$

Proof. Using Hölder's inequality and (5.30), we have, for small enough $\lambda > 0$

$$\int_\Omega \sigma_z^{-2-\lambda} \left|D^2 v\right|^2 \,\mathrm{d}x \leq \left(\int_\Omega \sigma_z^{\frac{-2-\lambda}{\lambda}} \,\mathrm{d}x\right)^{\lambda} \|v\|^2_{2,\frac{2}{1-\lambda}} \leq C\lambda^{-\lambda} (\kappa h)^{-2-\lambda+2\lambda} \|v\|^2_{2,\frac{2}{1-\lambda}} ,$$

where again, to simplify notation, $\|v\|_{k,p}$ is the $W^{k,p}$ norm. Using the standard regularity estimate for the Poisson equation, as well as Sobolev's and Hölders inequality,

$$\begin{aligned}
\int_\Omega \sigma_z^{-2-\lambda} \left|D^2 v\right|^2 \,\mathrm{d}x &\leq C\lambda^{-\lambda} (\kappa h)^{\lambda-2} \|w\|^2_{\frac{2}{1-\lambda}} \leq C\lambda^{-\lambda} (\kappa h)^{\lambda-2} \|\nabla w\|^2_{\frac{2}{2-\lambda}} \\
&\leq C\lambda^{-\lambda} (\kappa h)^{\lambda-2} \left(\int_\Omega \sigma_z^{-\frac{2-\lambda}{1-\lambda}} \,\mathrm{d}x\right)^{1-\lambda} \int_\Omega \sigma_z^{2-\lambda} |\nabla w|^2 \,\mathrm{d}x \\
&\leq C\lambda^{-1} (\kappa h)^{-2} \int_\Omega \sigma_z^{2-\lambda} |\nabla w|^2 \,\mathrm{d}x ,
\end{aligned}$$

where we used (5.30) in the last step. This proves the lemma. □

In their version of Lemma B.5, Rannacher and Scott [105] use the fact that $\lambda > 0$ can be chosen to be arbitrarily small in order to use a regularity estimate of the form

$$\|u\|_{2,2+\mu} \leq C\|\Delta u\|_{2+\mu} .$$

For polygonal boundaries, this only holds for small enough $\mu > 0$, depending on the polygon angles. In this work, we work with smooth boundaries, where this technique is not needed to apply the regularity estimate.

Lemma B.6. *For sufficiently large $\kappa > 0$, dependent on a sufficiently small $\lambda > 0$,*

$$\int_\Omega \sigma_z^\lambda \left(g^z - g_h^z\right)^2 \, \mathrm{d}x \leq C\lambda^{-1}\kappa^{-1} \int_\Omega \sigma^{2+\lambda} \left|\nabla(g^z - g_h^z)\right|^2 \, \mathrm{d}x$$

Proof. We define another auxiliary Poisson problem. Let $v \in H_0^1 \cap H^2$ solve

$$\Delta v = \sigma_z^\lambda (g^z - g_h^z) \, .$$

By Lemma B.5 and (B.1) we have

$$\begin{aligned} \int_\Omega \sigma_z^{-2-\lambda} \left|D^2 v\right|^2 \, \mathrm{d}x &\leq C\lambda^{-1} (\kappa h)^{-2} \int_\Omega \sigma_z^{2-\lambda} \left|\nabla \left(\sigma_z^\lambda (g^z - g_h^z)\right)\right|^2 \, \mathrm{d}x \\ &\leq C\lambda^{-1} (\kappa h)^{-2} \int_\Omega \sigma_z^{\lambda+2} \left|\nabla(g^z - g_h^z)\right|^2 + \sigma_z^\lambda (g^z - g_h^z)^2 \, \mathrm{d}x \end{aligned} \tag{B.10}$$

Using the Poisson equation, the definition of g_h, and the Hölder inequality,

$$\begin{aligned} \int_\Omega \sigma_z^\lambda (g^z - g_h^z)^2 \, \mathrm{d}x = \; & D\left\langle g^z - g_h^z, v \right\rangle = \; D\left\langle g^z - g_h^z, v - I_h v \right\rangle \\ &\leq \left(\int_\Omega \sigma^{2+\lambda} \left|\nabla(g^z - g_h^z)\right|^2 \, \mathrm{d}x \right)^{\frac{1}{2}} \left(\int_\Omega \sigma^{-2-\lambda} |\nabla(v - I_h v)|^2 \, \mathrm{d}x \right)^{\frac{1}{2}} \, . \end{aligned}$$

Using Lemma 5.4 and then (B.10) gives

$$\begin{aligned} \left(\int_\Omega \sigma_z^\lambda (g^z - g_h^z)^2 \, \mathrm{d}x \right)^2 &\leq Ch^2 \left(\int_\Omega \sigma^{2+\lambda} \left|\nabla(g^z - g_h^z)\right|^2 \, \mathrm{d}x \right) \left(\int_\Omega \sigma^{-2-\lambda} \left|D^2 v\right|^2 \, \mathrm{d}x \right) \\ &\leq C\lambda^{-1}\kappa^{-2} \Bigg(\left(\int_\Omega \sigma^{2+\lambda} \left|\nabla(g^z - g_h^z)\right|^2 \, \mathrm{d}x \right)^2 \\ &\qquad + \left(\int_\Omega \sigma^{2+\lambda} \left|\nabla(g^z - g_h^z)\right|^2 \, \mathrm{d}x \right) \left(\int_\Omega \sigma_z^\lambda (g^z - g_h^z)^2 \, \mathrm{d}x \right) \Bigg) \, . \end{aligned} \tag{B.11}$$

Now, using Young's inequality on the last line of (B.11) and choosing κ large enough (possibly dependent on λ) gives

$$\int_\Omega \sigma_z^\lambda (g^z - g_h^z)^2 \, \mathrm{d}x \leq C\lambda^{-1}\kappa^{-1} \int_\Omega \sigma^{2+\lambda} \left|\nabla (g^z - g_h^z)\right|^2 \, \mathrm{d}x \, .$$

□

Putting all of these bounds together, we come to the final result of this appendix.

Lemma B.7. *There is a $\lambda > 0$ such that for sufficiently large $\kappa > 0$*

$$\int_\Omega \left|\nabla \left(g^z - g_h^z\right)\right|^2 \sigma_z^{2+\lambda} \, \mathrm{d}x \leq C_{\kappa,\lambda} h^\lambda \, .$$

The constant $C_{\kappa,\lambda}$ here now depends on λ, κ.

Proof. Combining Lemma B.2 with Lemma B.6 gives

$$\begin{aligned} \int_\Omega \left|\nabla \left(g^z - g_h^z\right)\right|^2 \sigma_z^{2+\lambda} \, \mathrm{d}x &\leq C \widetilde{\int_\Omega} h^2 \sigma_z^{2+\lambda} \left|D^2 \left(g^z - g_h^z\right)\right|^2 + \lambda^{-1}\kappa^{-1}\sigma^{2+\lambda} \left|\nabla (g^z - g_h^z)\right|^2 \, \mathrm{d}x \\ &\leq C \widetilde{\int_\Omega} h^2 \sigma_z^{2+\lambda} \left|D^2 \left(g^z - g_h^z\right)\right|^2 \, \mathrm{d}x \, , \end{aligned} \tag{B.12}$$

for a large enough κ, dependent on λ.

We now deal with g_h^z by doing a pointwise estimate. Let x be a point not on any triangle edge. By Wardetzky [91, Lemma 3.3.1],[1] after exchanging curved and flat triangles and doing all

[1]The scenario mentioned in Wardetzky's Footnote 5 does not apply in our setting, as in our setting the interpolation operator maps into functions which vanish at the boundary (see Remark 5.1). Because of that, Lemma 3.3.1 holds at boundary triangles as well.

computations pointwise, we obtain for a function v which is C^2 within triangles,

$$\left|D^2 v(x)\right| \leq C\left|D_A^2 v(x)\right| + Ch|\nabla v(x)| \,,$$

where D_A^2 is the vector of all second partial derivatives on the flat triangle in the mesh Ω_h. We can use this same relation with D and D_A exchanged. For $g^z - g_h^z$ (which are both C^2 within triangles, as g^z solves a Poisson equation with smooth right-hand side, and g_h^z is a finite element function) this means

$$\left|D^2\left(g^z(x) - g_h^z(x)\right)\right|^2 \leq C\left|D_A^2\left(g^z(x) - g_h^z(x)\right)\right|^2 + Ch^2\left|\nabla\left(g^z(x) - g_h^z(x)\right)\right|^2 .$$

We now use that g_h^z is linear within mesh triangles in the interior, so its second derivatives on these triangles is zero. On boundary triangles, Lemma B.8 applies, and an additional term involving the gradient ∇g_h^z has to be added. Using this, as well as Lemma 5.1, we have

$$\begin{aligned}
\left|D^2\left(g^z(x) - g_h^z(x)\right)\right|^2 &\leq C\left|D_A^2 g^z(x)\right|^2 + Ch^2\left|\nabla g_h^z(x)\right|^2 + Ch^2\left|\nabla\left(g^z(x) - g_h^z(x)\right)\right|^2 \\
&\leq C\left|D_A^2 g^z(x)\right|^2 + Ch^2|\nabla g^z(x)|^2 + Ch^2\left|\nabla\left(g^z(x) - g_h^z(x)\right)\right|^2 \\
&\leq C\left|D^2 g^z(x)\right|^2 + Ch^2|\nabla g^z(x)|^2 + Ch^2\left|\nabla\left(g^z(x) - g_h^z(x)\right)\right|^2 .
\end{aligned}$$

Plugging this back into (B.12), gives, for small enough h,

$$\int_\Omega \left|\nabla\left(g^z - g_h^z\right)\right|^2 \sigma_z^{2+\lambda}\,\mathrm{d}x \leq Ch^2 \int_\Omega \sigma_z^{2+\lambda}\left(\left|D^2 g^z\right|^2 + h^2|\nabla g^z|^2\right)\,\mathrm{d}x \,.$$

Applying Lemma B.3 and Lemma B.4 gives

$$\int_\Omega \left|\nabla\left(g^z - g_h^z\right)\right|^2 \sigma_z^{2+\lambda}\,\mathrm{d}x \leq Ch^2 \int_\Omega \sigma_z^{2+\lambda}|\nabla\delta^z|^2 + \lambda^{-1}(\kappa h)^{-2}\,\sigma_z^{2+\lambda}|\delta^z|^2\,\mathrm{d}x \,.$$

The definition of δ^z gives

$$\int_\Omega \left|\nabla\left(g^z - g_h^z\right)\right|^2 \sigma_z^{2+\lambda}\,\mathrm{d}x \le Ch^2 \int_{T^z} \sigma_z^{2+\lambda}|\nabla\delta^z|^2 + \lambda^{-1}(\kappa h)^{-2}\,\sigma_z^{2+\lambda}|\delta^z|^2\,\mathrm{d}x\,.$$

By definition, the value of $d(x,z)$ within T^z is bounded by the edge length Ch. This means that, within T^z,

$$\sigma_z(x) \le \sqrt{C^2h^2 + \kappa^2h^2} \le \sqrt{C^2+\kappa^2}h\,.$$

With a different C, and for large enough κ,

$$\sigma_z(x) \le C\sqrt{1+\kappa^2}h \le C\kappa h$$

Together with the bounds for δ^z from (5.23) and using the fact that the area of T^z is bounded by Ch^2, this gives

$$\int_\Omega \left|\nabla\left(g^z - g_h^z\right)\right|^2 \sigma_z^{2+\lambda}\,\mathrm{d}x \le Ch^2\kappa^3 \int_{T^z} h^{2+\lambda}h^{-6} + \lambda^{-1}(\kappa h)^{-2}\,h^{-2}h^{2+\lambda}h^{-4}\,\mathrm{d}x \le C\kappa^3h^\lambda\,, \tag{B.13}$$

provided κ is large enough (dependent on λ). This proves the lemma. □

A treatment of curved boundary triangles. In Definition 5.7, we use an isoparametric modification of boundary triangles to achieve a finite element on a discrete surface whose boundary exactly corresponds to the boundary of the continuous surface Ω. This approximation, however, can affect the derivatives of the formerly linear FEM functions on these boundary triangles. In this appendix we summarize a result about this effect.

Lemma B.8. *Let $\Psi_T : \tilde{T} \to T$ be the isoparametric map of Scott [120] that maps a straight-edged triangle $\tilde{T}$ onto the triangle T which has one curved edge. Let $f \in C^2(T)$, $\tilde{f} = f \circ \Psi_T$. Then we have $\forall x \in T,\ x = \Psi_T(y)$*

$$\begin{aligned} \left|\nabla \tilde{f}(y)\right| &\leq C|\nabla f(x)| \\ \left|D^2 \tilde{f}(y)\right| &\leq C\left(\left|D^2 f(x)\right| + h|\nabla f(x)|\right) \ , \end{aligned}$$

where D^2 is the vector of all second derivatives. The constant C is independent of the triangle T, f or h.

The same estimates also hold in reverse, when $\tilde{f}(y)$ is exchanged with $f(x)$.

Proof. Let $\tilde{F}_T : \hat{T} \to \tilde{T}$ be the affine map that maps the standard reference triangle to the flat triangle $\tilde{T}$. Then, by Bernardi [121, Theorem 6.1], there is a map $\Phi_T : \hat{T} \to \mathbb{R}^2$ such that

$$\begin{aligned} \Psi_T &= \mathrm{Id} + \Phi_T \circ \tilde{F}_T^{-1} \\ |D^m \Phi_T(z)| &\leq C h^{m+1} \qquad \forall z \in \hat{T},\ m = 1, 2 \ , \end{aligned}$$

where D^m is the vector of all derivatives of order m.

By definition, $\left|D\left(\tilde{F}_T^{-1}\right)\right| \leq Ch^{-1}$, and $D^2\left(\tilde{F}_T^{-1}\right) = 0$. By the chain rule,

$$\begin{aligned} &|D\Psi_T(y)| \leq 2 + |D\Phi_T|\left|D\left(\tilde{F}_T^{-1}\right)\right| \leq 2 + Ch \leq C \qquad \forall y \in \tilde{T} \ , \\ &|D\Psi_T(y)| \geq 1 \qquad \forall y \in \tilde{T} \ , \\ &\left|D^2\Psi_T(y)\right| \leq \left|D^2\Phi_T\right|\left|D\left(\tilde{F}_T^{-1}\right)\right|^2 + (2 + |D\Phi_T|)\left|D^2\left(\tilde{F}_T^{-1}\right)\right| \leq Ch \qquad \forall y \in \tilde{T} \ . \end{aligned} \tag{B.14}$$

We now calculate the derivatives of $\tilde{f}$. Let $x = \Psi_T(y)$. Using (B.14), we have,

$$\begin{aligned}
\left|\nabla \tilde{f}(y)\right| &= \left|\nabla \left(f \circ \Psi_T\right)(y)\right| \leq |\nabla f(x)||\nabla \Psi_T(y)| \leq C|\nabla f(x)| \\
\left|D^2 \tilde{f}(y)\right| &= \left|D^2 \left(f \circ \Psi_T\right)(y)\right| \leq \left|D^2 f(x)\right||\nabla \Psi_T(y)|^2 + |\nabla f(x)|\left|D^2 \Psi_T(y)\right| \\
&\leq C\left(\left|D^2 f(x)\right| + h|\nabla f(x)|\right) ,
\end{aligned} \tag{B.15}$$

which proves one direction of the lemma.

We now prove the lemma in the other direction, with $\tilde{f}(y)$ exchanged with $f(x)$. Note that, by the inverse function theorem, (B.14) implies for $x = \Psi_T(y)$ that

$$\begin{aligned}
\left|D\Psi_T^{-1}(x)\right| &\leq C|D\Psi_T(y)|^{-1} \leq 3 \leq C \qquad \forall x \in T , \\
\left|D^2\Psi_T^{-1}(x)\right| &\leq C\left|D^2\Psi_T(y)\right||D\Psi_T(y)|^{-3} \leq Ch \qquad \forall x \in T .
\end{aligned}$$

The bounds for Ψ_T^{-1} are the same ones as the bounds for Ψ_T in (B.14), and thus the calculations from (B.15) still hold with $\tilde{f}(y)$ and $f(x)$ exchanged. □

Appendix C: Implementation Details for the Discrete Developability Energies

This section features explicit expressions for evaluating the discrete energies described in Section 7.2, as well as their gradients. Notation from Chapter 7 is used.

C.1 Derivatives of Basic Quantities

Our discrete energies depend only on the triangle areas $\mathcal{A}_f$, triangle normals $\mathbf{N}_f$ and interior angles θ_i^f in the triangle f with vertices i, j, k. The gradients with respect to vertex positions v_i, v_j, v_k (where the vertices are oriented) are

$$\nabla_{v_i} \mathcal{A}_f = \tfrac{1}{2} \mathbf{N}_f \times (v_k - v_j) \,, \tag{C.1}$$

$$\nabla_{v_i} \mathbf{N}_g = \frac{1}{\mathcal{A}_f} ((v_k - v_j) \times \mathbf{N}_f) \mathbf{N}_f^\top \,, \tag{C.2}$$

$$\begin{aligned} \nabla_{v_j} \theta_i^f &= \mathbf{N}_f \times \frac{v_i - v_j}{|v_i - v_j|}, \\ \nabla_{v_k} \theta_i^f &= \mathbf{N}_f \times \frac{v_k - v_i}{|v_k - v_i|}, \\ \nabla_{v_i} \theta_i^f &= -(\nabla_{v_j} \theta_i^f + \nabla_{v_k} \theta_i^f) \,. \end{aligned} \tag{C.3}$$

Since these quantities depend only on the positions of vertices i, j, and k, the gradients with respect to any other vertex are zero.

C.2 Implementation Details for the Combinatorial Energy

To evaluate the gradient of the combinatorial energy $E_{\text{comb}}(i)$ associated with the vertex i, we first identify the partition P minimizing $\pi(P)$ ((7.2)). The gradient of a single term in this sum with respect to the position v_p of any vertex p can then be expressed via

$$\nabla_{v_p}\left|\mathbf{N}_{\sigma_1} - \mathbf{N}_{\sigma_2}\right|^2 = 2(\mathbf{N}_{\sigma_1} - \mathbf{N}_{\sigma_2}) \cdot (\nabla_{v_p}\mathbf{N}_{\sigma_1} - \nabla_{v_p}\mathbf{N}_{\sigma_2}) \,,$$

where the normal gradient is given in (C.2). The energy gradient is then the sum over all such terms. In the case where there are two or more partitions of equal energy, the gradient of any of them will be a subgradient of the piecewise smooth energy E_{comb}, which is still suitable for the first-order descent strategy outlined in Section 7.3.1. To compute the L^∞ energy, the gradient of any maximal term provides a subgradient for $E^{\max}_{\text{comb}}$.

C.3 Implementation Details for the Covariance Energy

At any vertex i, let λ be an eigenvalue of the matrix $A_i := \sum_{f\in\text{St}(i)} \theta_i^f \mathbf{N}_f\mathbf{N}_f^\top$ with associated eigenvector $\mathbf{x}$. Then the gradient of λ with respect to the position $v_p \in \mathbb{R}^3$ of any vertex p is

$$\nabla_{v_p}\lambda = \sum_{f\in\text{St}(i)} (\mathbf{x}^\top\mathbf{N}_f)^2\nabla_{v_p}\theta_i^f + 2\theta_i^f(\mathbf{x}^\top\mathbf{N}_f)(\nabla_{v_p}\mathbf{N}_f)^\top\mathbf{x} \,, \tag{C.4}$$

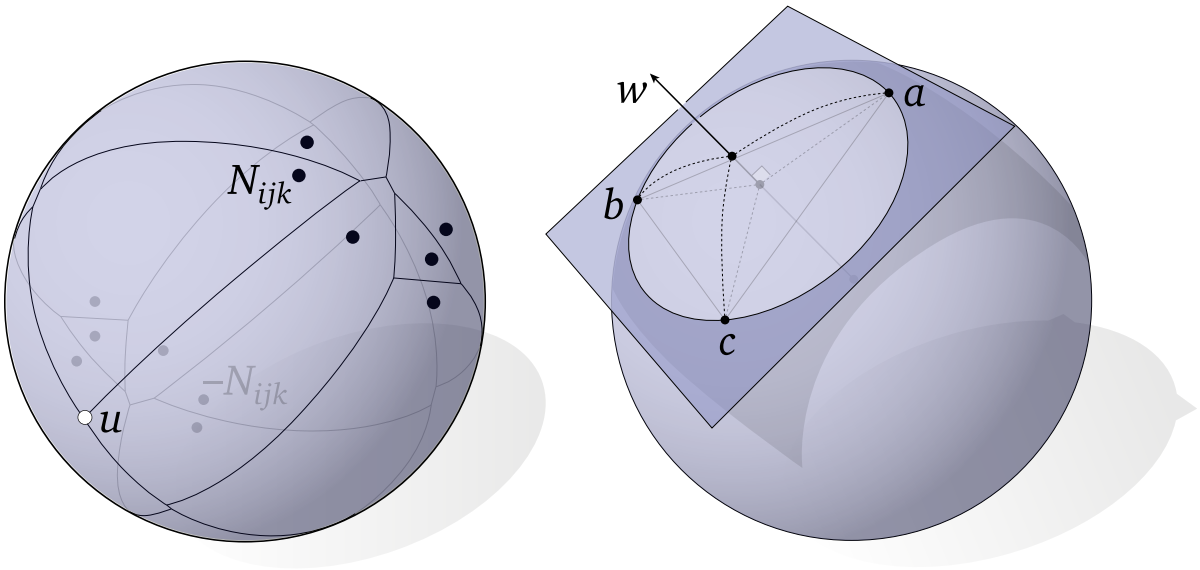


Figure C.1. The maximal covariance energy can be evaluated by sampling the objective at the vertices of a spherical Voronoi diagram (*left*), which are the unit normals of the triangles formed by three points on the unit sphere (*right*).

where we have applied the chain rule and the identity $\nabla_A \lambda = \mathbf{x}\mathbf{x}^\top$. Expressions for $\nabla_{v_p} \mathbf{N}_f$ and $\nabla_{v_p} \theta_i^f$ are given in Appendix C.1.

Computing the L^∞ covariance energy. To evaluate the energy given by (7.8), pick a vertex i and let

$$\varphi(\mathbf{u}) \coloneqq \max_{f \in \mathrm{St}(i)} \left(\mathbf{u} \cdot \mathbf{N}_f\right)^2 .$$

This function is piecewise smooth over spherical Voronoi cells associated with the unit normals $\mathbf{N}_f$ and their antipodes $-\mathbf{N}_f$ (see Figure C.1, *left*). Its minimum is therefore found at a vertex of the spherical Voronoi diagram that will be the spherical centroid of some triple of sites. Since φ achieves a minimum at a Voronoi vertex, minimizing φ over *all* triples necessarily yields the optimal value $\lambda_i^{\max}$. From the perspective of performance and numerical stability, simply evaluating φ for all triples is more attractive than explicitly building the Voronoi diagram, especially since

the number of distinct triples is typically very small. To compute the spherical centroid of three unit vectors $\mathbf{u}, \mathbf{v}, \mathbf{w}$, note that the geodesic circumcenter of a spherical triangle coincides with the unit normal of the plane containing the triangle's vertices (Figure C.1, *right*). The location of the site is therefore just $(\mathbf{v}-\mathbf{u})\times(\mathbf{w}-\mathbf{u})/|(\mathbf{v}-\mathbf{u})\times(\mathbf{w}-\mathbf{u})|$. To avoid a zero denominator, we simply omit redundant sites.

Computing the L^∞ covariance energy subgradient. Since φ is a maximum over a collection of convex differentiable functions, the gradient of any maximizing term provides a subgradient that can be used for optimization (Section 7.3.1). In particular, let $\mathbf{v}$ be the unit vector minimizing φ, let $\mathbf{M}$ be the maximizing normal, and let a, b, c be the triple of triangles whose normals define $\mathbf{v}$. Then the subgradient $\nabla_{v_p}\lambda_i^{\max}$ with respect to the position v_p of a vertex p can be expressed as

$$2\,(\mathbf{v}\cdot\mathbf{M})\left(\frac{\big(\mathbf{v}\cdot(e_p\times\mathbf{M})\big)}{2A_{\mathbf{M}}}\mathbf{M}+\sum_{\sigma\in\{a,b,c\}}\frac{\big(\mathbf{v}\cdot(e_{\sigma|p}\times\mathbf{N}_\sigma)\big)\,((e_\sigma\times\mathbf{v})\cdot\mathbf{M})}{4A_{abc}A_\sigma}\mathbf{N}_\sigma\right),$$

where $\mathbf{N}_\sigma$ is the unit normal of triangle σ, A_M and A_p are the areas of triangles with normals M and triangles p respectively, A_{abc} is the Euclidean area of a triangle with vertices a, b, c, $e_p \in \mathbb{R}^3$ is the edge vector opposite vertex p in the triangle with normal $\mathbf{M}$ (or zero if p is not contained in this triangle), and $e_{\sigma|p}$ is the edge across from vertex p in triangle σ.

C.4 Implementation Details for the Enhanced Covariance Energy

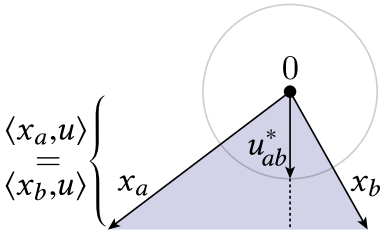
Let $\mathbf{N}_i$ be the area weighted normal at vertex i, let $v_i^f \coloneqq \mathbf{N}_i\hat{\times}\mathbf{N}_f$, let $\mu_i \coloneqq v_i^f\hat{\times}\mathbf{N}_i$, and let $\mu_f \coloneqq v_i^f\hat{\times}\mathbf{N}_f$, where $\mathbf{u}\hat{\times}\mathbf{v} \coloneqq \frac{\mathbf{u}\times\mathbf{v}}{|\mathbf{u}\times\mathbf{v}|}$ denotes the normalized cross product. Then the gradient of $\tilde{\mathbf{N}}_i^f$ with

respect to the position v_p of a vertex p can be expressed as

$$\begin{aligned}\nabla_{v_p}\tilde{\mathbf{N}}_f = &\left(\mu_i\mu_f^\top + \frac{\varphi_i^f}{\sin\varphi_i^f}v_i^f(v_i^f)^\top\right)\nabla_{v_p}\mathbf{N}_f \\ &- \left(\mu_i\mu_i^\top + \varphi_i^f\mathbf{N}_i\mu_i^\top + \frac{\varphi_i^f}{\tan\varphi_i^f}v_i^f(v_i^f)^\top\right)\nabla_{v_p}\mathbf{N}_i\ .\end{aligned} \tag{C.5}$$

The gradients for $\mathbf{N}_i$ and $\mathbf{N}_f$ can be computed using the expressions from Appendix C.1. The gradient of the overall energy can then be expressed by substituting $\tilde{\mathbf{N}}_f$ for $\mathbf{N}_f$ in (C.4).

Computing the L^∞ enhanced covariance energy. One can compute the minimum width of the polygon of $\tilde{\mathbf{N}}_i^f$s from (7.9) via the method of rotating calipers in $O(n\log n)$ time, including construction of the convex hull. However, since n is always quite small (about six on average) a simpler implementation is to just directly minimize the expression (7.9) by enumerating all distinct pairs of vectors $x_a = \pm\tilde{\mathbf{N}}_i^f, x_b := \pm\tilde{\mathbf{N}}_i^g$. The minimizing vector u_{ab}^* for any such pair will be the vector pointing along the altitude of the triangle $(0, x_a, x_b)$ (see inset), and one can easily show that the minimum width of the convex hull is then the minimum value of (7.9) evaluated for all such vectors u_{ab}^*.



The subgradient is found by taking the gradient of the combination of vectors and faces maximizing (7.9). Here the only new expression is the gradient of the unit altitude u_{ab}^*, given by

$$\nabla_{x_a}u_{ab}^* = -\frac{1}{|w|^3}(\mathbf{N}_i\times w)w^\top,$$

where $w := x_b - x_a$ (and similarly for x_b).