Conformal Geometry Processing

Thesis by

Keenan Crane

In Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy

California Institute of Technology

Pasadena, California

2013

(Defended April 15, 2013)

To Frank, for reminding us that there are always so many new things to do...

Acknowledgments

The opportunity to write a dissertation is the consequence of extreme privilege: I am paid to sit around and think while others toil in the field. My thanks go first and foremost to the taxpayer who unknowingly (or unwillingly) donated slivers of precious income so that I might discover something useful. I'm sorry I can't promise much, but please trust that we are doing something good! Thanks also to the community that told me boys can succeed at mathematics and computer science—you might try telling the girls that too. Most of all, this privilege stems from the people I met along the way.

First and foremost: the advisors. Most PhD students are lucky to get one mentor who is switched on, attentive, and supportive. I have had *four!* **Ulrich:** this thesis clearly contains your DNA. Thank you for sharing a beautiful theory, and for being our guide along an authentic "upward spiral!" Thank you for your enthusiasm and your humor: lobsters, snowmen, and giraffe algorithms...who knew it could be so much fun? Most of all, thank you for your humility and your unshakeable patience. I have been sloppy, and I have been stubborn, and I have learned so, so much. **Max:** thanks for being unprofessional—it is much better to work with a friend. Thanks for the (chocolate) cigarette breaks, and for candid advice over way too many beers. (Perhaps someday I will actually take it!) Thanks for being demanding, and not letting me just do it in *Mathematica.* In the words of another good guy, *"Was für ein sexy Schnurrbart, meine junge!"* **Mathieu:** thank you for demonstrating that you can do serious work without turning into a (too-)serious person. Thank you for being amicable and supportive and mature—no matter what. When you did insult me, thank you for keeping it gender-neutral. Finally, thanks for putting up with some *really* terrible jokes. (Do you remember who won the clothing competition?) Last but not least, **Peter:** where do I begin? Thanks for reminding me that we are all perfect the way we are, even as we do the same stupid things over and over again...Thanks for keeping your door open, and listening patiently as I insist *very emphatically* that something is right—even though it's *totally* wrong! Thank you for making me write it in 10 pages; thank you for being a comrade in the war against bad æsthetics (even if they never notice). Thanks for being excited with me; thanks for letting me explore.

Long before Caltech there were two other mentors, without whom things would have ended *badly!* **Eliot:** who in their right mind picks up a high school kid as a research assistant, and wastes hour upon precious research hour explaining totally banal stuff like linear systems and Monte Carlo? You do, apparently—and it has made all the difference. Thanks for showing me the pleasure of finding things out (and for all the fizzy juice). **John:** you saved my ass. Let's just leave it at that! Thanks for being "big on topology; small on topography." Thanks for fielding questions from foreign trolls. Most of all, thanks for believing in me. (By the way, I think this thesis more than qualifies me for induction into the Secret Society of the Quaternion Handshake!)

Then there were my cohorts at Caltech. Thanks to the older ones, for being there when I landed. I asked too many questions. You made me eat too many Oreos! **Patrick:** thanks for bearing with me as I learned the fundamentals. I know it wasn't easy! But it was fundamental. **Lily:** thank you for being my director. What a strange video we made of that poor pig's liver! **Nathan:** why were you still there?! I'm glad you were. Long live the lab in Jorgensen, and lunches on the bridge. Thanks too to the younger ones. **Patrick:** thanks for indulging with me in some truly bizarre questions. Do we know yet how to intersect a city with a ray? **Fernando:** thanks for being part of DDG: The Next Generation. Someday we will know everything there is to know about triangles—and can move on to quads! **Mike:** thanks for being stuck in a room with me for four years! Thanks for letting no one ignorant of geometry enter there. Thanks for putting the Shredder in his place. Thanks for the great debates, and for permitting me to vex you with all things convex. Thanks to **Simon** for talking cars, life, and for reminding me how sexy I am! Do you know the equation for a cow? Thanks be to **Alex**, though he may never know what I am keen on. Thanks to **Katherine**, for an excellent week of eigenvectors. Thanks to **Melissa** for getting us out of the lab and up onto the volcano; thanks to **Tom** for getting us out to the racetrack; thanks to **Christian** for getting us out to the bar. Many thanks to **Maria, Jeri, Sydney, Sheila, Lisa, Pat** and **Dave** for locking me back in when I was locked out, for fixing the things that I broke, and for doing all the things I didn't even realize you were doing!

The scene at Illinois undoubtedly prepared me for what would come next. Thank you **John** for inviting me in, giving me space, for offering to pay. To East Palo Alto Criminals **Jerry and Jared** for so many things: for GPCPU.org, for Metropolis, for Marissa Cooper, for Long Islands...yeah, it really just goes downhill from there, doesn't it? **Andrew, Nathan:** thank you for DEC, geometric mechanics, and for hanging with a young'un. Much was absorbed, much was enjoyed. **Shadi, Mahsa, Hossein, Ali:** ممنون دوست من! May you never end up looking like MethMorph. Thanks to **Nate** for letting me jump on board, thanks to **Scott** for nose holes. Thanks to coauthor **Ryan** for dancing in pants (it wasn't

me!), and for helping me get where I was going next. Thanks to the ACM: **Matt, Vilas, Grier, Yisong, Parisa, Leo, Steve, Keith, Carrino, Dave, Ari, Anthony, Vlad, Peterse*n*,** and yes even you **Clausen**. You taught me the skills I use every day—you made me do things I regret even to this day. (There is so much more to say here... see my email of March 17, 2006.) Finally, thank you to **Michael** for introducing me to digital geometry processing in the first place. Without your most excellent class I would never have applied to Caltech, nor made this stuff my life's work (so far...).

Much time was spent in lands afar. **Mark:** thanks for long meandering discussions of architecture and geometry, and for asking "how do you pick a research topic?" [PKK00] Thanks to **Madeleine Robert** for giving me the keys and looking the other way. **Mario, Sofien, Yuliy, Thibaut, Ian**: thanks for jazz, Gruyère, Mario, meringues, wine on the train, and strange Swiss towns full of erotic alien art...! Then there is the land of Gänseliesel. **Max:** for giving me a kitchen; for making me neighbors with Minkowski. **Clarisse, Ulrich, Alessio, Henrik, Knut, Lars:** thank you for *never* staying on topic, for Cuban cafés, and for the "Gauss" tour. For warning me of the unstylishness of cappuccino after noon. For American movies without English subtitles. For calling me peanut and making me sing Gwen Stefani. For Uli the Bee! In the land of Sachertorte, thanks to **Chris and Olga**, my gracious hosts. I am so lucky to have friends like you two. Thanks to **Elisabeth Hacker** for managing the important stuff. Thanks to **Morten, Brittany, Michael, Filip, and Viktoriia** for schnapps, Heurigers, and death by mafia. Thanks to brother **Justin**, my fellow nomad, for spreading the good word of DDG, and for showing me his black umbrella last Friday night. Finally, there was Berlin. Thanks to **Sasha, Ulrich** and **Annett Gillmeister** for making it possible; thanks to **Boris, Steffen, Stefan, Andre, Felix** for enlightenment in a tiny corner coffee room on the 8th floor. (Big Brother is watching!)

Much was gained in stints at NVIDIA and Autodesk. Out West, thank you **Bengt-Olaf, Steve, John, Nate, Nick**. I drank from the fire hose. Thank you fellow interns **Eric, Greg, Jim, Jered, Ben, Sharif**. Special thanks to **Kapil**, for making PP. Thank you **Jen** and **Jamie**, for making it fun. Thank you Demo Team: **Curtis, Ryan, Eugene, Cam, Alex, Hubert**—what unique talents. Thank you for showing me beautiful things, and for trusting me to do my own thing. Extra thanks to Stepdad for moon rocks, hell hole, and for teaching me about blobs before I even knew who you were. Up North, thank you **Jos**, my taller, blonder brother. The four-simplex is the sign of the Devil, dude! I had a *great* time in Toronto. Thank you **Azam**, for the wig, and for your inimitable singing voice. Thank you research group, **Gord, George, Ramtin, Michael, Hyunyoung, James, Ian, Sean**, for giving me the freedom to explore and invent. Finally, thank you Mesh Mix Master **Ryan** for spasmodic conversations about meshes and matices.

Thanks to friends and family, who give me the things my work cannot. Thank you Chesterians **Nick, Luke, Tucker, Jon, Patrick**, for reminding me to always, *always* put safety second. Thank you **Caitlin, Lauren, Maggie, Eyrún, Mike, Emily, Kim, ``Wolf,'' Kris, Shaun, Artur, David, Dan, Danielle, Matt, Pratyush, Megan** for BBQs, Tubular Tuesdays, Tour de Franzia, hot tub crawls, parties at the Cats, parties on the West Side, parties in Old Town, parties in New Town, parties downtown, pancake parties, birthday parties, holiday parties, house parties, house*warming* parties, dinner parties, melancholy Bollywood dance parties, going away parties, coming back parties, bachelor parties, and at long last, graduation parties. Thank you **Cole, Robin, Anya, Mike** for encouraging me to do my best (and forget the rest). Thank you **Sarah, Orion, Beth Anne, August, Forrest, Metz**, for being there whenever I went back home. Thank you **Harold, Devin, Andrew** for reminding me of art in the midst of all my nerdiness. Thank you **Chris**, for always letting me be candid—our conversations have made a difference. Thank you **Jerry**, for your considerate and selfless character—it is an inspiration. Thank you **Jessica**, for a friendship I will always treasure. It has helped me become a more whole and balanced person. Thank you grandparents **Lee, Trudy, Kathryn, Gina, Frank**, for thinking about my future before I was old enough to say your names or see your faces, and later for your unrelenting effort to understand what I'm doing out here on the lunatic fringe. Thank you parents, **Christine, Dennis, Donna**, for loving me long before I was a Caltech PhD. Thank you Dad for waking up early in the morning to help me with HyperCard. Thank you Mom, for supporting unconditionally all my crazy pursuits. Thank you Donna for helping me become a better adult. Finally, thank you **Rebecca**, my first and best friend, for always being on my side (even when we disagree).

The work in this thesis was funded by a Google PhD Fellowship, the Hausdorff Research Institute for Mathematics, BMBF Research Project GEOMEC, SFB / Transregio 109 "Discretization in Geometry and Dynamics," the TU München Institute for Advanced Study (funded by the German Excellence Initiative), the Center for the Mathematics of Information at Caltech, DFG Research Center Matheon, and the DFG Research Unit Polyhedral Surfaces. Data is provided courtesy of the Stanford Graphics Computer Laboratory, the AIM@SHAPE Shape Repository, Autodesk, 3D Universe, David Bommes, Chris Legasse, John Phillips, Gerald Ganson, Fabian Aiteanu, and Mirela Ben-Chen.

Abstract

This thesis introduces fundamental equations and numerical methods for manipulating surfaces in three dimensions via *conformal transformations*. Conformal transformations are valuable in applications because they naturally preserve the integrity of geometric data. To date, however, there has been no clearly stated and consistent theory of conformal transformations that can be used to develop general-purpose geometry processing algorithms: previous methods for computing conformal maps have been restricted to the flat two-dimensional plane, or other spaces of constant curvature. In contrast, our formulation can be used to produce—for the first time—general surface deformations that are perfectly conformal in the limit of refinement. It is for this reason that we commandeer the title *Conformal Geometry Processing*.

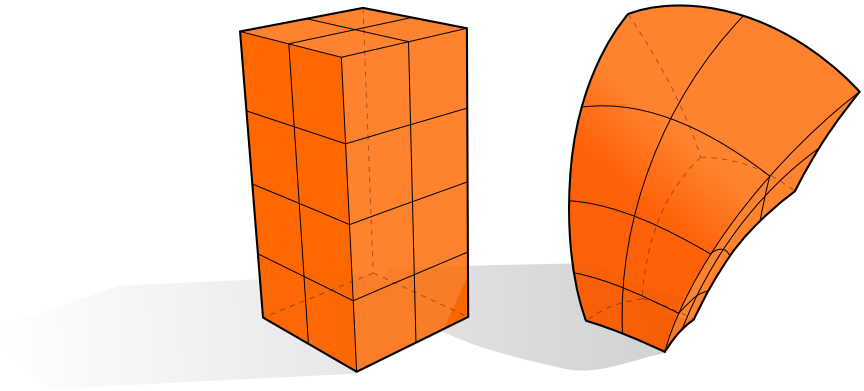
The main contribution of this thesis is analysis and discretization of a certain *time-independent Dirac equation*, which plays a central rôle in our theory. Given an immersed surface, we wish to construct new immersions that (i) induce a conformally equivalent metric and (ii) exhibit a prescribed change in extrinsic curvature. Curvature determines the potential in the Dirac equation; the solution of this equation determines the geometry of the new surface. We derive the precise conditions under which curvature is allowed to evolve, and develop efficient numerical algorithms for solving the Dirac equation on triangulated surfaces.

From a practical perspective, this theory has a variety of benefits: conformal maps are desirable in geometry processing because they do not exhibit *shear*, and therefore preserve textures as well as the quality of the mesh itself. Our discretization yields a sparse linear system that is simple to build and can be used to efficiently edit surfaces by manipulating curvature and boundary data, as demonstrated via several mesh processing applications. We also present a formulation of Willmore flow for triangulated surfaces that permits extraordinarily large time steps and apply this algorithm to surface fairing, geometric modeling, and construction of constant mean curvature (CMC) surfaces.

Contents

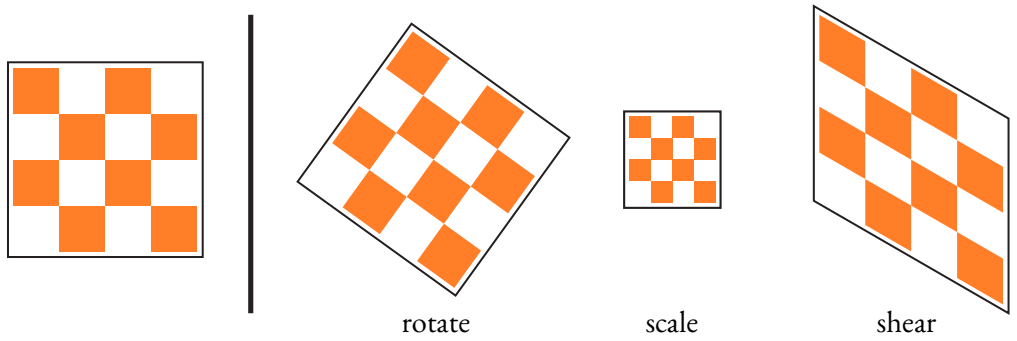
Chapter 1

Introduction



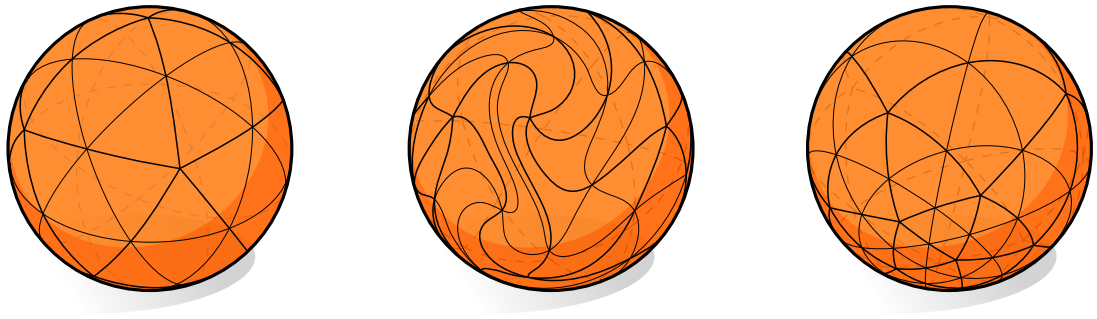
The world around us is full of shapes: airplane wings and cell phones, brain tumors and rising loaves of bread, fossil records and freeform architectural surfaces. To a large extent, our ability to master these domains is limited by our capacity to design, process, and analyze this geometry. The aim of *Digital Geometry Processing* (DGP) is to extend ideas from classical signal processing to new algorithms for working with three-dimensional shape. In doing so, however, one must address a classical signal processing question: how does one process data while preserving its integrity? And what does "integrity" even mean in the context of three-dimensional geometric data? We adopt the perspective that the integrity of a geometric signal can be characterized by preservation of local *metric* properties, i.e., lengths and angles associated with a given surface. As we will see, this approach has some very attractive benefits from a computational perspective; it is also linked to a rich and beautiful body of knowledge from surface theory.

In this thesis we focus specifically on *conformal* transformations, which, roughly speaking, preserve *angle*. Why conformal? To answer this question, it helps to consider the types of transformations that are available in the smooth setting. Locally (i.e., in each tangent plane) we can decompose any smooth deformation into three modes, namely, rotation, scaling, and shearing:



Globally, these modes induce different types of surface deformations. For instance, if all three modes are permitted then we get arbitrary smooth deformations, which provide great flexibility but make it difficult to preserve signal integrity. If shearing is not allowed then the angle between any pair of vectors is preserved and we obtain conformal deformations. If in addition we prohibit scaling then length is also preserved and we get *isometric* deformations, i.e., both length and angle are perfectly preserved. Finally, if all three modes are prohibited then we get *rigidity*: the shape looks exactly like the one we started with. Actually, in most cases isometric deformations are also rigid (up to global rotation and translation). For instance, Cauchy's rigidity theorem tells us that the boundary of any convex body (e.g., the unit sphere) cannot be deformed isometrically—in general, many of the objects we encounter in everyday life are likewise isometrically rigid.

Examining the list of possible motions, one is naturally inclined to ask: why not allow shearing but prohibit scaling? In other words, why not try to preserve area instead of angle? This idea certainly has a natural appeal, but one should be careful to understand what area preservation really implies. For instance, we can preserve area on the sphere by swirling its surface around in a fluid-like motion (computed here by advecting mesh vertices in a divergence-free velocity field):



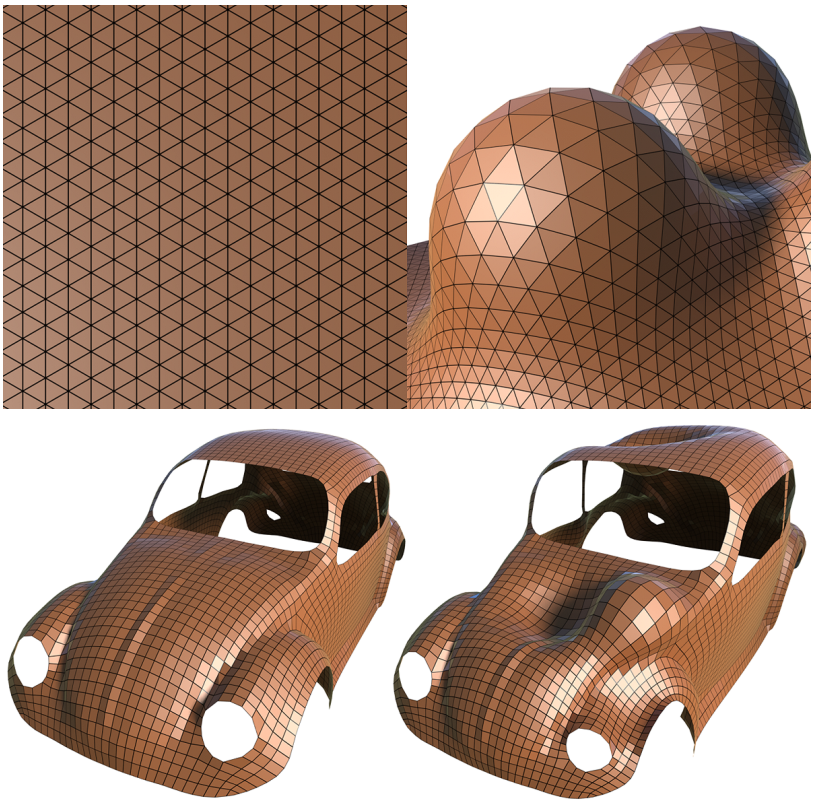
From the perspective of geometry processing, this type of motion is undesirable because it can lead to arbitrarily bad degeneration of mesh elements, and distortion of data associated with the surface. In other words, like smooth deformations, general area-preserving motions are "too flexible" to preserve signal integrity[1]. In contrast, angle-preserving motions are more rigid and hence better preserve the kinds of features we are interested in. Finally, one can try to preserve various properties of the metric only approximately—for instance, *elastic* energies are commonly used in geometry processing to find a deformation that is "as rigid as possible," i.e., as close as possible to isometric [SA07, CPSS10]. Later in this thesis we will describe a similar approach: preserve angles exactly, but try to preserve area as well as possible (Section 4.5.3.1).

The next question we must answer is: how does one manipulate a surface in space without distorting angles? Traditional methods consider maps into the complex plane, which are useful only for problems such as surface parameterization and planar shape deformation where the target surface is flat. We instead consider maps into the *quaternions* $\mathbb{H}$, which allows us to work directly with surfaces in $\mathbb{R}^3$ (Chapter 2). In particular, we introduce the *quaternionic Dirac operator* and use it to develop novel integrability conditions for conformal deformations (Chapter 4). A key feature of our theory is that a deformation can be expressed via a scalar function that prescribes the extrinsic curvature. Practically speaking, this setup allows us to "paint" a change in curvature on a surface and produce the corresponding conformal deformation (Chapter 6). For instance, a model that has already been carefully detailed (left) can be further altered without compromising texture or geometric detail (middle), whereas standard mesh editing tools do not respect these features (right):

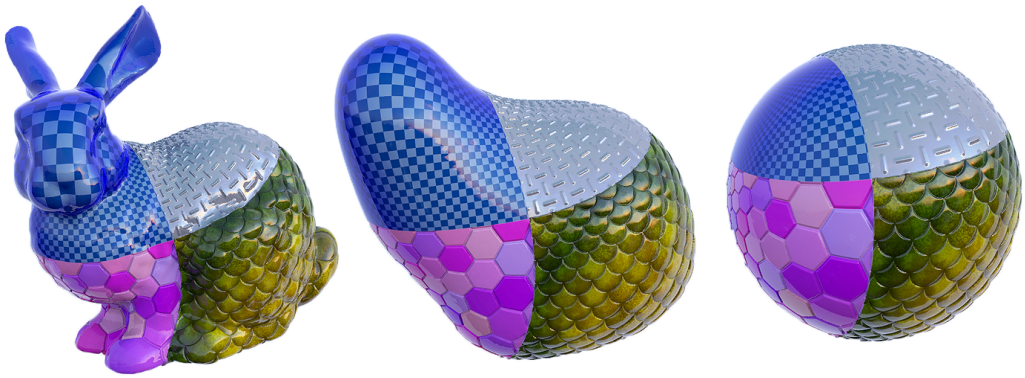
These transformations also preserve mesh quality—here a grid of equilateral triangles (top left) is substantially deformed while preserving triangle shape almost perfectly (top right); similarly, a carefully

[1]The present colloquial use of the term "area-preserving" or "volume-preserving" in geometry processing is perhaps more suggestive of volume-preserving *elastic* deformations, which also try to minimize angle distortion.

quadrangulated beetle (bottom left) becomes an old beat-up car (bottom right) while preserving the shape of quadrilaterals:



When applied to a task like *surface fairing*, where we wish to smooth out or regularize a given surface, our approach gracefully preserves texture and mesh quality throughout all stages of the flow:



Moreover, we can take integration time steps orders of magnitude larger than those seen in classical approaches (Section 6.4).

Our discretization yields sparse linear systems that are simple to build and can be efficiently solved using standard numerical techniques (Section 6.6). In particular, deformations are computed by solv-

ing an eigenvalue problem and a Poisson equation; matrices have the same sparsity as the standard cotangent discretization of the Laplace-Beltrami operator [Mac49, Duf59]. The linearity of this formulation is most unusual: if one instead wants to prescribe standard quantities (such as the induced metric, principal curvatures, etc.) more difficult *nonlinear* problems must be solved [ESP08].

Conformal maps have already proven invaluable in applications like computational medicine [GWC+04] and anatomy [LD11] because they are cheap to compute and preserve important geometric structure. For example, they preserve the numerical conditioning of discrete Poisson problems, and preserve the integrity of signals associated with a surface [CPS11]. Exceptional numerical methods for parameterization [GY03, MTAD08, SSP08] and uniformization [JKG07] have been developed by applying *intrinsic* tools from Riemann surface theory. However, conformal maps have thus far seen relatively little use in broader geometry processing applications due to the lack of a readily discretizable theory for surfaces immersed in three dimensions. The purpose of this thesis is to extend the applicability of conformal maps in geometry processing *beyond the traditional context of parameterization/uniformization.*

1.1 Contributions

The main outcome of this thesis is a clearly stated and consistent theory of conformal immersions from which one can develop a broad class of conformal geometry processing algorithms. In particular, let f be an immersion of a compact surface M into $\mathbb{R}^3$. We describe numerical procedures for constructing new immersions $\tilde{f}$ that, up to discretization error, (I) induce the same conformal structure on M and (II) satisfy additional application-specific criteria. For instance, to regularize noisy data one might define a fairing energy E (such as the Willmore energy) and require that $E(\tilde{f}) < E(f)$.

In pursuit of this goal, we introduce the *quaternionic Dirac operator* (Chapter 3), leading to simple integrability conditions for conformal transformations of smooth surfaces sitting in $\mathbb{R}^3$ (Section 4.3). Kamberov et al. [KPP98] investigate integrability in the case of surfaces with spherical topology; we complete the picture by establishing conditions on surfaces of arbitrary topological type. These conditions are discretized and applied to the computation of conformal deformations (Chapter 5). To our knowledge, our method is the first to produce general transformations of discrete surfaces that converge to perfectly conformal deformations in the limit of spatial and temporal refinement, as validated by numerical experiment (Section 5.8.3). We also show how changes in extrinsic curvature can be used to express a variety of mesh processing tasks (Chapter 6); in particular, we

establish the first numerical algorithm for Willmore flow that does not exhibit a time step restriction based on grid spacing (Section 6.4).

Much of the work in this thesis appears in the following publications:

- Keenan Crane, Ulrich Pinkall, and Peter Schröder, "Spin Transformations of Discrete Surfaces," *ACM Transactions on Graphics (SIGGRAPH)*, 30(4), 2011.
- Keenan Crane, Ulrich Pinkall, and Peter Schröder, "Robust Fairing via Conformal Curvature Flow," *ACM Transactions on Graphics (SIGGRAPH)*, 32(4), 2013.

A significant portion of the material (esp. Chapters 3 and 4) is previously unpublished, and appears on a private collaborative research blog shared by the above authors during the years 2010-2013.

1.2 Related Work

Differential representations are well-studied in digital geometry processing [BS08] but ignore the question of *integrability*: under what condition does a prescribed differential quantity come from an actual surface? The standard practice is to forgo this question and instead look for a surface that best approximates prescribed data in the least-squares sense, as first proposed by Yu et al. [YZX+04]. For conformal maps we need to be more careful since the closest surface may be far from conformal, even when the new differential comes from a similarity transformation at each point (see Section 5.9 for several comparisons).

Later work expresses edge vectors with respect to local coordinate frames at vertices to ensure rigid motion invariance [LSLCO05]. In these algorithms reconstruction is nonlinear since coordinate frames depend nonlinearly on vertex positions. Paries et al. [PDK07] also seek conformal deformations, encoding local frames as quaternions at vertices. Reconstruction entails a sequence of alternating least-squares problems: first to minimize the differences between local frames, then to find the closest surface that exhibits these frames; ultimately, however, distortion remains. Lipman et al. [LCOGL07] also minimize frame differences, with an additional nonlinear unit-norm constraint at each vertex to enforce *isometric* deformation. As before the nonlinear relationship between positions and frames is resolved with an iterative scheme and no integrability condition is considered.

Conformal deformations have also been studied in the context of *cage-based editing*, where coordinate functions on a coarse volumetric *cage* induce transformations of an enclosed surface [LLCO08, BCWG09]. The main difficulty is that the only conformal maps from $\mathbb{R}^3$ to itself are *Möbius transformations*, which are far too rigid for surface editing. As a result, cage-based methods are inherently limited to so-called *quasi-conformal* maps, which can still introduce significant distortion (Section 5.9). On the other hand, these methods are highly efficient since they need only evaluate simple (often closed-form) coordinate expressions. It may therefore make sense to use our method in *conjunction* with existing methods to produce a final, high-quality surface (see Sections 6.3 and 5.9).

Discrete conformal maps have been studied extensively in the case of *surface parameterization* where the target surface has constant curvature (such as the plane or the sphere). A common approach is to compute a pair of discrete harmonic functions that describe a map into the plane, requiring the solution of a linear system [Mer01, DMA02, LPRM02, GY03], or an eigenvalue problem [MTAD08]. Conformal transformations can also be constructed by prescribing values at vertices that directly control the rescaling of the metric [BCGB08, YKL$^+$08, SSP08]. However, a metric alone is not sufficient to describe an embedding—one must therefore restrict the scope of the problem to metrics of constant curvature in order to realize the final surface. We instead store values at vertices that encode both local rescaling and *rotation*, which is ultimately sufficient to describe the extrinsic geometry of a deformation (Section 4.3).

Several methods allow prescription of extrinsic curvatures [ESP08] and metric properties such as length [EP09]. However, the resulting energies are expressed in terms of vertex positions, making them highly nonlinear. Using higher-order differential quantities (such as our function ρ) as primary degrees of freedom may prove useful in this setting.

In contrast to previous approaches our setup ensures that transformations are conformal *by construction*—any failure to achieve integrability on a mesh is due purely to discretization error (Section 5.8.3).

Chapter 2

Background

2.1 Notation

The following notation is adopted throughout unless otherwise indicated [1]. Single angle brackets $\langle \cdot, \cdot \rangle$ and bars $|\cdot|$ denote the Euclidean inner product and norm (respectively) either on individual vectors in $\mathbb{R}^n$ or evaluated pointwise for vector-valued functions; double brackets $\langle\langle \cdot, \cdot \rangle\rangle$ and bars $||\cdot||$ denote the $\mathscr{L}^2$ inner product and norm on functions. The symbol $\times$ denotes the usual cross product on $\mathbb{R}^3$. When applied to vector fields, operators like $\langle \cdot, \cdot \rangle$ and $\times$ act pointwise. Vector space isomorphism is denoted by $V = W$ for any two vector spaces V, W. The symbol $\mathbf{1}$ denotes a function equal to one at every point, or a vector containing a one in each entry. The symbol id denotes the identity map.

If $\varphi(t)$ is a time-varying quantity, then $\dot{\varphi}$ denotes the time derivative of φ at time zero, i.e.,

$$\dot{\varphi} \coloneqq \frac{d}{dt}\varphi\bigg|_{t=0}.$$

For any quantity $\varphi(s)$ defined along a 1D curve, φ' denotes the spatial derivative:

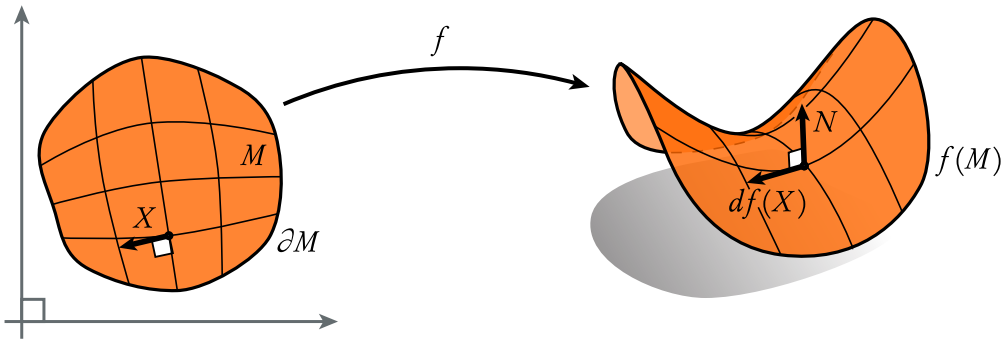
$$\varphi' \coloneqq \frac{d}{ds}\varphi.$$

Indices are typeset without serifs (e.g., i, j, k) to distinguish them from the basis for imaginary quaternions (i, j, and k). Superscripts x, y, and z denote components of vectors in $\mathbb{R}^3$.

[1] Note that symbols in most equations are *hyperlinked*, i.e., clicking on a symbol will take you to its definition

2.2 Curves and Surfaces

This section reviews standard concepts from the differential geometry of surfaces and outlines the notation we adopt in this thesis. Importantly, expressions in this section apply to general immersed surfaces, whereas in the remainder of the thesis we will adopt the convention that surfaces are described by a *conformal* immersion (see Section 2.3). *Caveat lector!*



The objects we consider in this thesis are 1- and 2-dimensional hypersurfaces immersed in Euclidean space, i.e., curves in the Euclidean plane $\mathbb{R}^2$ and surfaces in Euclidean $\mathbb{R}^3$. Explicitly, we describe the shape of a curve or surface via a map $f : M \to \mathbb{R}^{n+1}$ where M is a topological n-manifold with boundary ∂M. We assume throughout that M is compact, connected, and orientable, and will subsequently indicate when it is closed ($\partial M = \emptyset$) or simply connected ($\pi_1(M) = 0$). Since the dimension of M is no greater than two it has a uniquely defined smooth structure, hence a well-defined tangent bundle TM and differential $df : TM \to T\mathbb{R}^n = \mathbb{R}^n$. Tangent vector fields on M are typically denoted by capital Roman letters X, Y, Z, etc., and we use the notation $df(X)$ to denote the pushforward of X to the ambient space $\mathbb{R}^n$. For convenience we will sometimes abbreviate the pushforward with a caret, e.g.,

$$\hat{X} := df(X),$$

and will more generally use a caret to distinguish vector fields on the ambient space $\mathbb{R}^n$ from those on the domain M. Throughout we assume that f is an *immersion*, meaning that df maps nonzero vectors to nonzero vectors. (Note that immersions are not in general injective, i.e., $f(M)$ may have self-intersections.) Two immersions f and $\tilde{f}$ are *regularly homotopic* if there exists a continuous family of immersions $f_t : M \to \mathbb{R}^n, t \in [0, 1]$ such that $f_0 = f$ and $f_1 = \tilde{f}$.

Since f is an immersion we can use it to define an induced *Riemannian metric*

$$g(X, Y) := \langle df(X), df(Y) \rangle,$$

(also known as the *first fundamental form* $I(X,Y)$). When M is a surface, we let $\mathcal{A}$ denote the total area

$$\mathcal{A} := \int_M \sigma,$$

where σ is the induced *volume form*

$$\sigma(X,Y) := \langle N, df(X) \times df(Y) \rangle,$$

and N is the unit normal. More precisely, the unit normal or *Gauss map* $N : M \to S^{n-1} \subset \mathbb{R}^n$ is a smoothly varying unit vector field on M such that

$$\langle N, df(X) \rangle = 0$$

for all vector fields X. Our convention is to let N be the *outward* normal, i.e., normal vectors drawn on a closed compact surface are visible from a point at infinity. For a vector field $\hat{Z}$ on the immersed surface we use $^\top$ to denote the tangential part and $^\perp$ to denote the normal part, i.e.,

$$\hat{Z}^\top := \hat{Z} - \langle \hat{Z}, N \rangle N,$$

$$\hat{Z}^\perp := \langle \hat{Z}, N \rangle N.$$

Since N is a *unit* vector field, its differential dN must produce vectors orthogonal to N itself, i.e., $dN(X)$ is a vector field tangent to the surface. Hence, there must exist a tangent space endomorphism $S : TM \to TM$ such that

$$df(SX) = dN(X)$$

for all vector fields X. We adopt the somewhat unconventional nomenclature that dN is called the *Weingarten map* and S is called the *shape operator* (traditionally these terms are used interchangeably, but we wish to distinguish dN, which produces immersed vector fields, from S, which produces intrinsic ones). Using these operators, we can express the second fundamental form as

$$II(X,Y) := -\langle dN(X), df(Y) \rangle = -g(SX,Y)$$

and the normal curvature along X as

$$\kappa(X) := -\frac{\langle dN(X), df(X)\rangle}{|df(X)|^2} = \frac{II(X,X)}{g(X,X)}.$$

It is not hard to verify that S is self-adjoint with respect to g and hence II is symmetric in X and Y. On surfaces, we will use X_1, X_2 to denote unit vectors along the directions of principal curvature and κ_1, κ_2 to denote the corresponding curvatures, i.e.,

$$SX_i = \kappa_i X_i$$

and hence

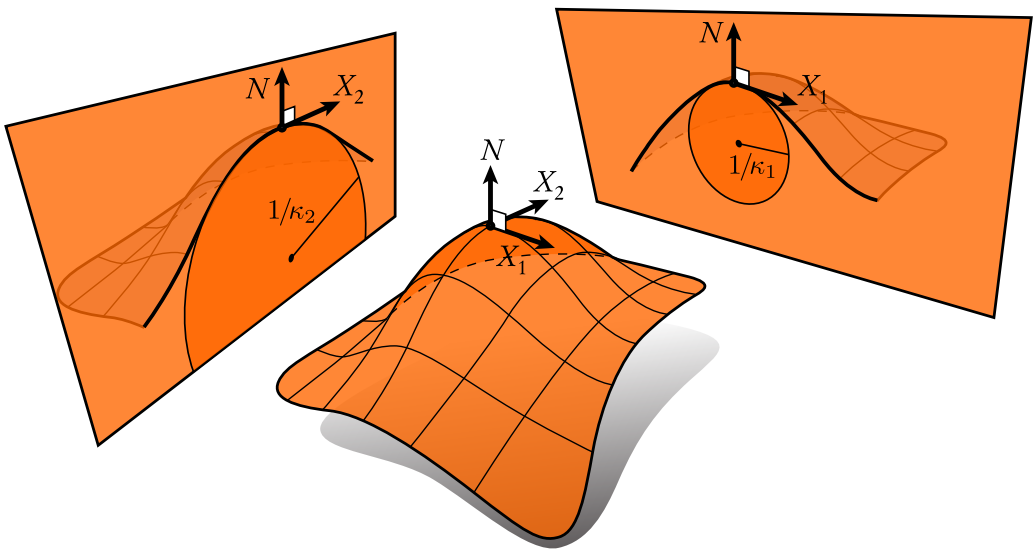
$$dN(X_i) = \kappa_i df(X_i).$$

Since S is self-adjoint with respect to g it follows that the principal directions are orthogonal with respect to the induced metric, i.e., $g(X_1, X_2) = 0$. We define the Gaussian and mean curvature of an immersed surface as

$$K = \kappa_1 \kappa_2,$$

$$H = \frac{\kappa_1 + \kappa_2}{2},$$

respectively.



For vector fields X, Y on Euclidean $\mathbb{R}^n$, the Levi-Civita connection $\hat{\nabla}$ can be expressed as

$$\hat{\nabla}_X Y = dY(X)$$

where d denotes the exterior derivative and Y is viewed as a vector-valued 0-form. In other words, to differentiate a vector field Y we simply take its usual directional derivative along X. If ∇ is the Levi-Civita connection on an immersed hypersurface $f : M \to \mathbb{R}^n$, then we can decompose $\hat{\nabla}$ as

$$\hat{\nabla}_{\hat{X}} \hat{Y} = df(\nabla_X Y) + II(X, Y)N,$$

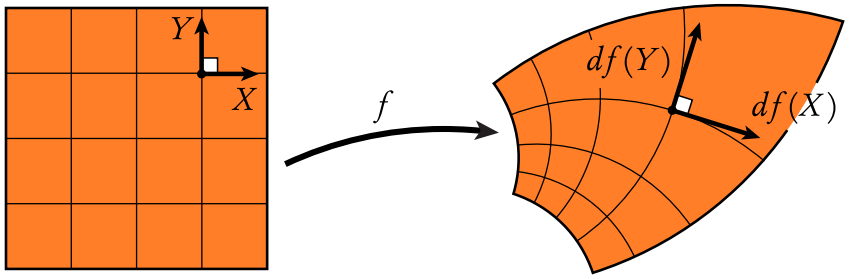
i.e., the derivative of a tangent vector field consists of a tangential part determined by the intrinsic covariant derivative and a normal part given by the second fundamental form. (Recall that $\hat{X} = df(X)$ and $\hat{Y} = df(Y)$; here we extend $\hat{X}$ and $\hat{Y}$ arbitrarily to smooth vector fields on $\mathbb{R}^n$.) We then have

$$df(\nabla_X Y) = (\hat{\nabla}_{\hat{X}} \hat{Y})^{\top},$$

$$II(X, Y) = \langle N, (\hat{\nabla}_{\hat{X}} \hat{Y})^{\perp} \rangle.$$

Throughout we use div, grad, curl, and Δ to denote the divergence, gradient, curl, and Laplace-Beltrami operators induced by f. The quantity $\frac{1}{2}\Delta f$ is called the *curvature normal*, equal to κN for curves and HN for surfaces.

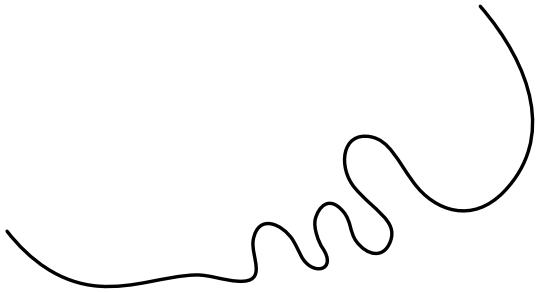
2.3 Conformal Immersions



When working with surfaces, we make the simplifying assumption that f is a *conformal immersion.* This idea will be made more precise in a moment, but roughly speaking conformal means that the notion of *angle* is preserved as we go from M to $\mathbb{R}^3$.

Why is it useful (or even reasonable) to work with conformal maps? For readers already familiar with geometry processing, the conformal assumption may sound like a "special" condition, tailored to very specific problems and application domains. As we will see, however, conformal immersions are a natural tool for analyzing general surfaces (i.e., when doing pen-and-paper derivations)—even if one has no interest whatsoever in *computing* conformal maps. Moreover, this convention puts absolutely no restriction on the type of geometry we are able to work with in practice.

To better motivate the use of conformal maps, consider the much simpler case of curves. In many situations a curve appears as just a subset of the plane, with no explicit parameterization: [2]



Conceptually, we can always imagine that this curve is the image $f(M)$ of some map $f : M \to \mathbb{R}^2$. In other words, as long as we know such a map exists, we can use it to *reason* about the curve without ever having to explicitly compute the function f itself. Moreover, since we do not have to go to the trouble of computing f, we may as well assume that it is as nice as possible. For curves, one particularly nice convention is to let f be an *arc-length* or *isometric* parameterization $f : [0, L] \to \mathbb{R}^2; s \mapsto f(s)$, which means that length is locally preserved: $|df(X)| = |X|$. In this case quantities like the *total* length of the curve can be expressed as just

$$\ell = \int_0^L ds.$$

For a general parameterization we would instead have to write something like

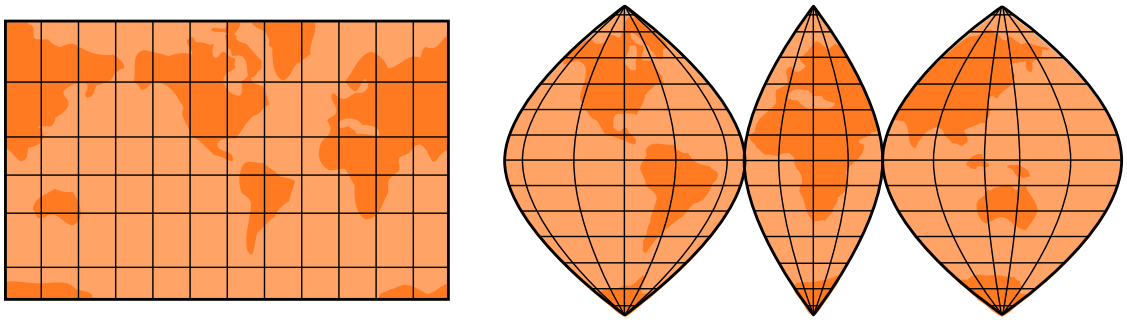
$$\ell = \int_0^L \sqrt{\left(\frac{\partial f^x}{\partial s}\right)^2 + \left(\frac{\partial f^y}{\partial s}\right)^2}\, ds \qquad \text{or equivalently} \qquad \ell = \int_0^L \sqrt{\det g}\, ds,$$

where f^x and f^y are the two coordinate functions associated with f, and g is the induced metric. In other words, we have to account for *metric distortion* as we go from M into $\mathbb{R}^2$, which often com-

[2] This curve, for instance, was extracted from the painting *"Yellow, Red, Blue"* by Kandinsky, who very likely had no parameterization in mind!

plicates our understanding of the curve. Conversely, an isometric parameterization provides simple algebraic expressions that convey a clear picture of the underlying geometry.

Like curves, surfaces often come without an explicit parameterization. Nowhere is this statement more true than in digital geometry processing, where a surface begins life as a big jumble of points or triangles floating in space! Unlike curves, however, we cannot always expect to find a parameterization free of metric distortion, which explains why cartographers have so many different projections of the globe:



On the other hand, we can always describe a surface using an angle-preserving or *conformal* map—*no matter what the surface looks like!* In other words, we do not have to make any special assumptions about the surface under consideration. As with curves, this convention tends to produce simple algebraic expressions that make it easy to understand the behavior of the geometry. As just one example, consider the *Laplace-Beltrami operator* Δ acting on a scalar function φ, which in the case of a general immersion $f : \mathbb{R}^2 \supset \Omega \to \mathbb{R}^3; (s_1, s_2) \mapsto f(s_1, s_2)$ is typically expressed as

$$\Delta\varphi = \sum_{i,j=1}^{2} \frac{1}{\sqrt{\det g}} \frac{\partial}{\partial s_i} \left(\sqrt{\det g}\, g^{ij} \frac{\partial}{\partial s_j} \varphi \right).$$

For a conformal immersion this expression simplifies to just

$$\Delta\varphi = \frac{1}{\sqrt{\det g}} \left(\frac{\partial^2}{\partial s_1^2} + \frac{\partial^2}{\partial s_2^2} \right) \varphi.$$

In the latter expression we still have to account for metric distortion, but overall the geometric meaning is easier to interpret: the Laplacian on the surface is just a scalar multiple of the usual sum of second derivatives in the plane. Again, we can use this insight to *understand* the geometry, without ever touching a computer or writing down an explicit formula for f.

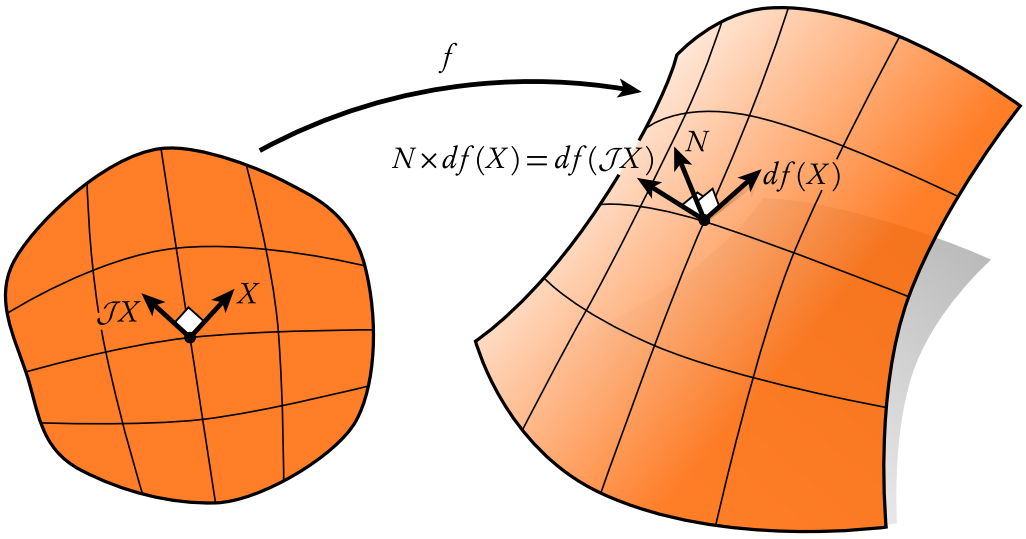
In general, we do not even need to imagine that the domain of f is the Euclidean plane (or any other metric space, for that matter). To be more precise, let M be a topological surface. A *complex structure*[3] on M is a vector bundle automorphism $\mathcal{J} : TM \to TM$ such that

$$\mathcal{J}^2 = -\mathrm{id}.$$

Intuitively $\mathcal{J}$ represents a 90-degree rotation in each tangent space, just as the imaginary unit i represents a 90-degree rotation in the complex plane. An immersion $f : M \to \mathbb{R}^3$ is then *conformal* if for all $X \in TM$

$$df(\mathcal{J}X) = N \times df(X). \tag{2.1}$$

Intuitively: an immersion is conformal if 90-degree rotations in the tangent space yield 90-degree rotations in the ambient space. One can easily verify that this definition agrees with the usual notion of a conformal immersion, namely that $|df(X)| = |df(\mathcal{J}X)|$ and $\langle df(X), df(\mathcal{J}X)\rangle = 0$ for all X. Note that we adopt the usual right-hand rule for the cross product, which means that $\mathcal{J}$ is effectively a *counter-clockwise* rotation.

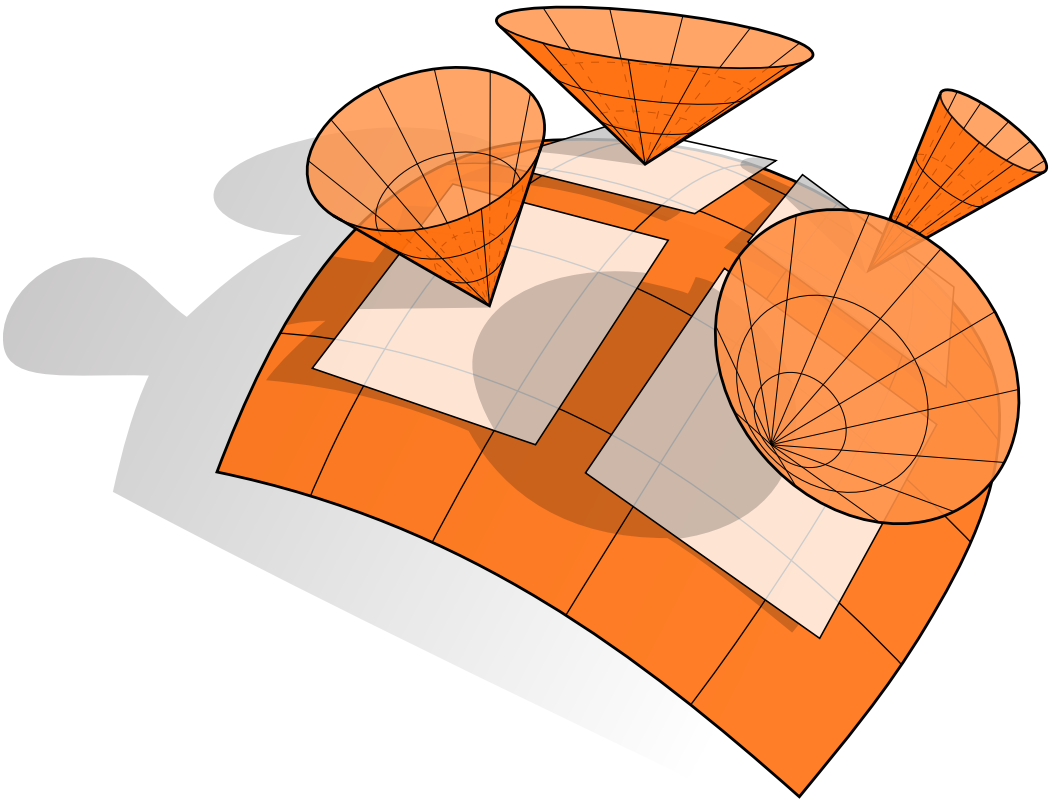


It is perhaps more intuitive to turn this story around: any immersion $f \to \mathbb{R}^3$ induces a natural complex structure on M, obtained by rotating each vector 90 degrees around its normal—in essence, the normal field of an immersed surface *defines* the complex structure. In other words, start with *any* immersion f and define $\mathcal{J}X$ as the unique vector such that Equation 2.1 holds. Then f is conformal

[3]Note that $\mathcal{J}$ is more typically called an *almost complex structure*, but for surfaces this data is sufficient to uniquely determine a complex structure in the usual sense.

by construction. Note that one does not need to invoke deeper results here (like uniformization): the existence of a conformal immersion is guaranteed by the fact that an immersed surface already has a smoothly varying Riemannian (and hence conformal) structure.

2.4 Half-Densities



Intuitively, a *half-density* is a function that maps each tangent vector to a scalar multiple of its length, where the scaling coefficient depends only on the basepoint. Since this function depends only on the length of a vector (and not its direction), its graph over each tangent space is a circular cone. A half-density on a surface can therefore be visualized as a collection of cones with varying aspect ratio, as depicted above.

More precisely, let M be a topological surface with complex structure $\mathcal{J}$, and let $f : M \to \mathbb{R}^3$ be any immersion compatible with this structure (i.e., $df(\mathcal{J}X) = N \times df(X)$ for all X). Every half-density on M can be expressed as a map

$$X \mapsto \varphi|df(X)|,$$

where φ is some real-valued function on M. (Notably, half-densities are not 1-forms since they are not linear in the argument X.) We will typically denote half-densities by expressions like $\varphi|df|$ to make explicit both the reference immersion f and the corresponding coefficient function φ. In other words,

$$\varphi|df|(X) := \varphi|df(X)|.$$

Note that for any other conformal immersion $\tilde{f}$ we can always find a function $\tilde{\varphi}$ such that $\tilde{\varphi}|d\tilde{f}| = \varphi|df|$, namely $\tilde{\varphi} := \varphi|df|/|d\tilde{f}|$. In other words, there is nothing special about the particular immersion we choose to represent a given half-density (other than the fact that it agrees with the conformal structure on M).

The fact that we can write all half-densities with respect to a single reference immersion f means that they have a natural vector space structure induced by the vector space structure of the associated coefficient functions, i.e., $\varphi_1|df|+\varphi_2|df| := (\varphi_1+\varphi_2)|df|$. This space also has a natural inner product

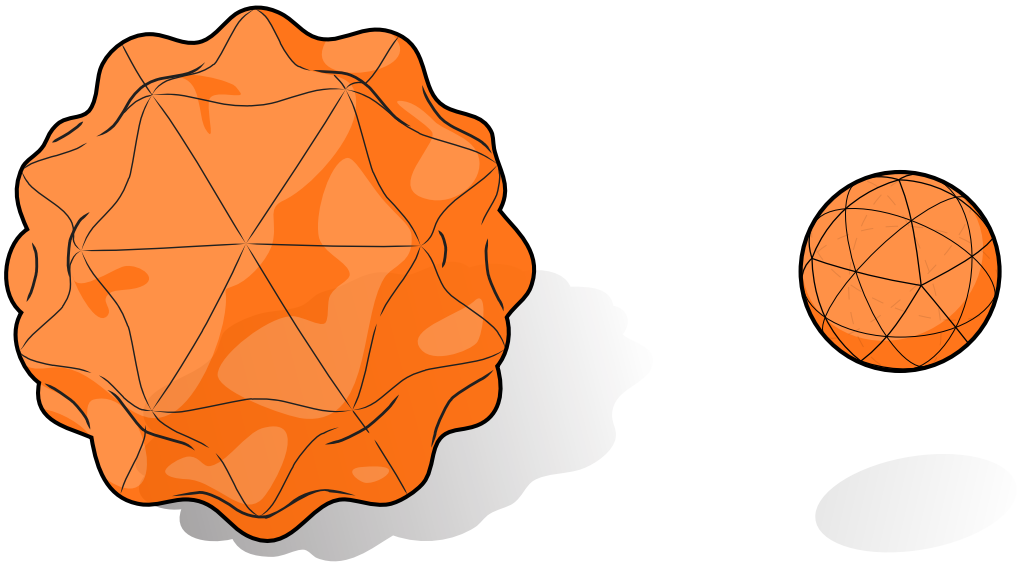
$$\langle\langle \varphi_1|df|, \varphi_2|df| \rangle\rangle := \int_M \varphi_1\varphi_2|df|^2,$$

i.e., we take the inner product of the two coefficient functions with respect to the volume form $|df|^2$ induced by the reference immersion. We will denote the space of half-densities on M by $\mathcal{H}(M)$, (or just $\mathcal{H}$ when the domain is understood.) An important property of half-densities is that their norm is preserved by *arbitrary* diffeomorphisms $\xi \in \mathrm{Diff}(M)$, not just volume-preserving diffeomorphisms $\xi \in \mathrm{SDiff}(M)$. This fact will come into play in Chapter 4.

A central object in our discussion of conformal immersions is the *mean curvature half-density*

$$\mu := H|df|$$

Note that μ is *scale-invariant*, since the mean curvature H is inversely proportional to the induced length element $|df|$. This quality is particularly convenient in the context of geometry processing, since one does not have to calibrate computations to account for global scale. In contrast, mean curvature alone does not convey much information about the global appearance of a surface. For example, the two surfaces below exhibit mean curvature whose magnitude is identical on average:



Likewise, a local increase in mean curvature H is easily mistaken for an increase in bending, but in reality an increase in mean curvature can also result from uniform shrinking and a simultaneous *decrease* in bending. In contrast, the effect of manipulating mean curvature half-density μ is unambiguous: increasing μ exaggerates bending; decreasing μ mitigates it.

2.5 Quaternions

The *quaternions* $\mathbb{H}$ are a number system well-suited to expressing three-dimensional geometric relationships; they also provide a natural algebraic language for conformally immersed surfaces [KNPP02]. Quaternions can be viewed as a four-dimensional real vector space with basis $\{1, i, j, k\}$ along with the associative, noncommutative *Hamilton product*, which is uniquely determined by the relationship

$$i^2 = j^2 = k^2 = ijk = -1.$$

The *imaginary quaternions* $\mathrm{Im}\mathbb{H}$ are elements of the three-dimensional subspace spanned by $\{i, j, k\}$. Throughout this thesis we use imaginary quaternions to represent vectors in $\mathbb{R}^3$ via the natural identification

$$(a, b, c) \mapsto ai + bj + ck.$$

We use Re and Im to denote the real and imaginary parts of a quaternion, respectively, i.e.,

$$\begin{aligned} \mathrm{Re}(a+bi+cj+dk) &= a, \\ \mathrm{Im}(a+bi+cj+dk) &= bi+cj+dk. \end{aligned}$$

We will sometimes express a quaternion q as the sum of a real scalar part $q_s \in \mathbb{R} \subset \mathbb{H}$ and an imaginary vector part $q_v \in \mathbb{R}^3 = \mathrm{Im}\mathbb{H} \subset \mathbb{H}$, writing

$$q = q_s + q_v$$

as an abbreviation for $q = q_s + q_v^x i + q_v^y j + q_v^z k$. In this case, the Hamilton product of two quaternions q, p becomes

$$qp = \underbrace{q_s p_s - \langle q_v, p_v \rangle}_{\text{real}} + \underbrace{q_s p_v + p_s q_v + q_v \times p_v}_{\text{imaginary}}.$$

In the case where q and p are purely imaginary we omit the subscript v and simply write

$$qp = q \times p - \langle q, p \rangle. \tag{2.2}$$

The *conjugate* of a quaternion $q = a + bi + cj + dk$ is the involution given by

$$\overline{q} := a - bi - cj - dk,$$

i.e., conjugation simply negates the imaginary part. In particular, $\overline{q} = -q$ whenever $q \in \mathrm{Im}\mathbb{H}$. Conjugation reverses the order of a product, i.e.,

$$\overline{qp} = \overline{p}\,\overline{q},$$

and can be used to concisely express the Euclidean inner product for any $p, q \in \mathbb{H} = \mathbb{R}^4$:

$$\langle p, q \rangle = \mathrm{Re}(\overline{p}q).$$

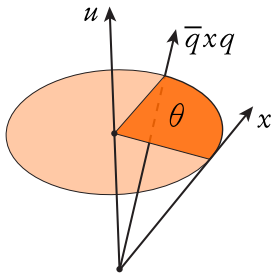
In particular,

$$|q|^2 = \overline{q}q.$$

Finally, if q is a *unit* quaternion then the action

$$q \mapsto \overline{q}xq$$

on vectors $x \in \mathrm{Im}\mathbb{H}$ represents a rotation of Euclidean 3-space, i.e., $\overline{q}xq$ is a rotated version of x. Explicitly, $q = \cos(\theta/2) - \sin(\theta/2)u$ represents rotation by an angle θ around the axis $u \in \mathbb{R}^3$. More generally, if $|q| \neq 1$ then this action represents a *similarity*, i.e., rotation and uniform scaling by a factor $|q|^2$. This representation will be especially important in our description of conformal deformations, since conformal maps are local similarity transformations.



2.6 Quaternionic Differential Forms

Quaternion-valued *differential forms* provide a convenient language for discussing conformal immersions. On an orientable surface M, these objects are quite easy to describe:

- *0-forms* are $\mathbb{H}$-valued functions on M;
- *1-forms* are real-linear maps taking a vector field on M to an $\mathbb{H}$-valued function;
- *2-forms* are rescalings of the volume form σ by an $\mathbb{H}$-valued function.

For example, the immersion f can be thought of as a quaternionic 0-form, the differential df can be thought of as a quaternionic 1-form (taking any vector field X to the corresponding $\mathbb{H}$-valued function $df(X)$), and the volume form σ itself can be thought of as a quaternionic 2-form. For a 1-form α, *real-linear* means that

$$\alpha(hX + Y) = h\alpha(X) + \alpha(Y)$$

for any real-valued function h and pair of vector fields X, Y.

The *wedge product* $\wedge$ of two 1-forms α, β yields a 2-form

$$\alpha\wedge\beta(X,Y) := \alpha(X)\beta(Y) - \alpha(Y)\beta(X),$$

and one can easily verify the basic identities

$$\overline{\alpha\wedge\beta} = -\overline{\beta}\wedge\overline{\alpha},$$

$$\alpha\wedge h\beta = \alpha h\wedge\beta$$

where h is again real-valued. To give a concrete example, the induced volume form of a general immersion f can be expressed as the quaternionic 2-form

$$\sigma = \frac{1}{2} df\wedge N df,$$

since then

$$\begin{aligned} 2\sigma(X,Y) &= df(X)Ndf(Y) - df(Y)Ndf(X) \\ &= -(df(X)df(Y) - df(Y)df(X))N \\ &= -2(df(X)\times df(Y))N \\ &= 2\langle N, df(X)\times df(Y)\rangle. \end{aligned}$$

In the special case where f is conformal, we will denote the volume form by the symbol

$$|df|^2.$$

This convention makes explicit the fact that the volume is induced by the immersion f, which is useful in contexts involving several different immersions. The notation $|\cdot|^2$ is motivated by the fact that σ is uniquely determined by the relationship $\sigma(X,\mathcal{J}X) = |df(X)|^2$. Also note that $g(X,X) = \langle df(X), df(X)\rangle = |df(X)|^2$; hence, one can also think of $|df|^2$ as the Riemannian metric induced by f. *Division* by $|df|^2$ identifies any 2-form ω with the corresponding scalar multiple of the volume form, i.e.,

$$\frac{\omega}{|df|^2} := \omega(X,\mathcal{J}X),$$

where X has unit length with respect to f, i.e., $|df(X)| = 1$. For readers familiar with exterior calculus, it may be useful to note that

- $|df|^2$ represents the Hodge star on 0-forms ($\star\varphi = \varphi|df|^2$),
- $\mathcal{J}$ represents the Hodge star on 1-forms ($\star d\alpha(X) = d\alpha(\mathcal{J}X)$), and
- $1/|df|^2$ represents the Hodge star on 2-forms ($\star\omega = \omega/|df|^2$).

In the particular case of the 1-form df, we can also use the unit normal to express the Hodge star, i.e., $\star df = N df$. Using the Hodge star, one can write the wedge product of 1-forms as

$$\alpha \wedge \beta = \alpha \star \beta - \star\alpha\beta,$$

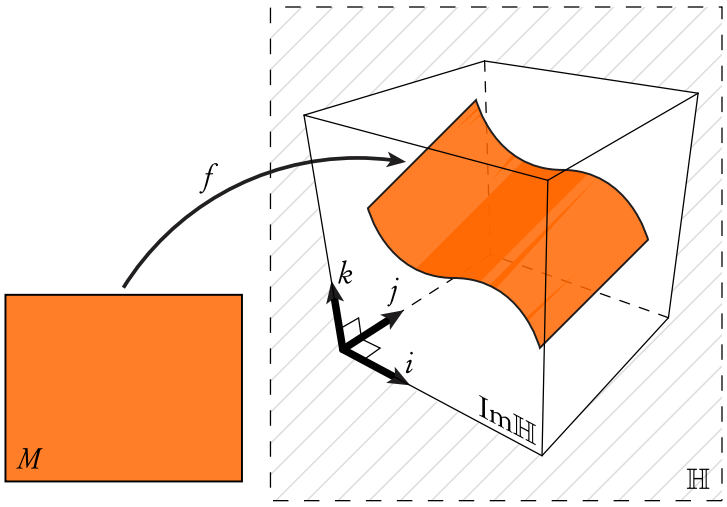
and the induced Laplace-Beltrami operator on 0-forms as

$$\Delta = \star d \star d.$$

Finally, the $\mathscr{L}^2$ inner product on quaternion-valued functions $\psi, \phi : M \to \mathbb{H}$ is given by

$$\langle\langle \psi, \phi \rangle\rangle := \int_M \overline{\psi}\phi |df|^2.$$

2.7 Geometry in the Quaternions

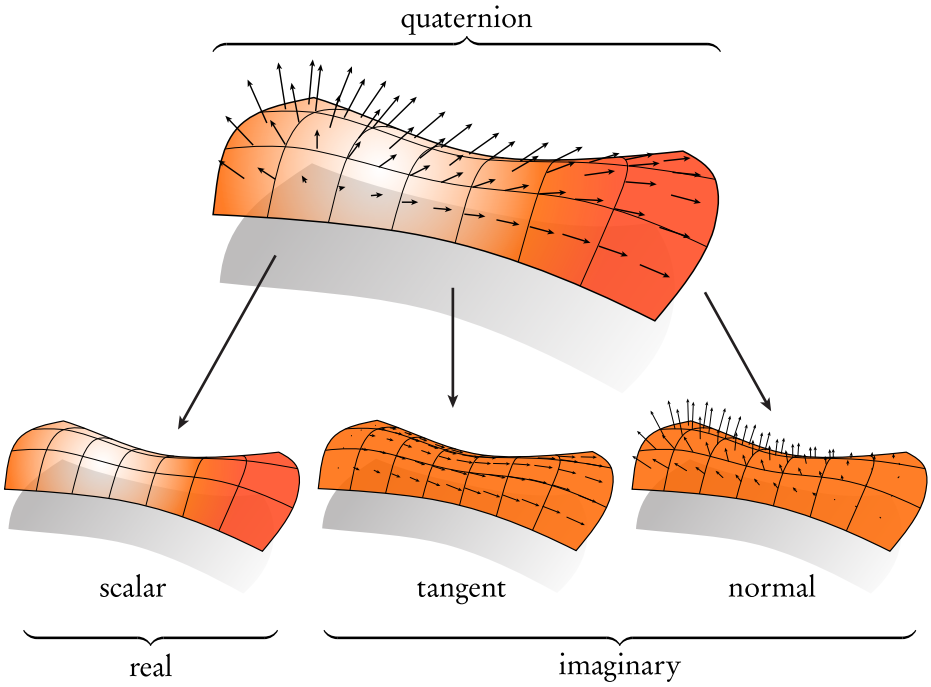


The ideas introduced in the past few sections lead naturally to the main convention adopted in this thesis: every surface in $\mathbb{R}^3$ can be viewed as a conformal immersion in the quaternions, i.e., $f : M \to \mathrm{Im}\mathbb{H} \subset \mathbb{H}$ [KPP98, KNPP02]. This description has nice algebraic consequences, since most geometric quantities of interest can be naturally identified with different components of a quaternion-valued

function. In particular, any quaternionic function $\psi : M \to \mathbb{H}$ can be decomposed as

$$\underset{\text{quaternion}}{\psi} = \underset{\text{real}}{a} + \underset{\text{tangent}}{df(Y)} + \underset{\text{normal}}{bN}$$

for some vector field Y and pair of real-valued functions a, b on M. In other words, it can be written as the sum of a real part, a tangential part, and a normal part, respectively, as depicted below.



When working with purely imaginary quantities $q, p \in \mathrm{Im}\mathbb{H}$ representing vectors in $\mathbb{R}^3$, we will repeatedly take advantage of the relationship

$$qp = q \times p - \langle q, p \rangle$$

to simplify calculations. For instance, for any tangent vector field $\hat{Y} = df(Y)$ and pair of normal vector fields $\hat{Z}_1 = b_1 N$, $\hat{Z}_2 = b_2 N$ we have

$$\hat{Z}_1 \hat{Z}_2 = \hat{Z}_2 \hat{Z}_1 \quad \text{and} \quad \hat{Y}\hat{Z} = -\hat{Z}\hat{Y}.$$

In other words: normal vector fields commute with each-other; normal and tangent vectors anti-commute. Similarly, for any pair of orthonormal tangent vector fields $df(X)$, $df(\mathcal{J}X)$ we have

$$df(X)df(\mathcal{J}X) = df(X) \times df(\mathcal{J}X) = N|df(X)|^2,$$

where the magnitude $|df(X)|^2$ is a consequence of the fact that f is a conformal immersion.

Chapter 3

The Quaternionic Dirac Operator

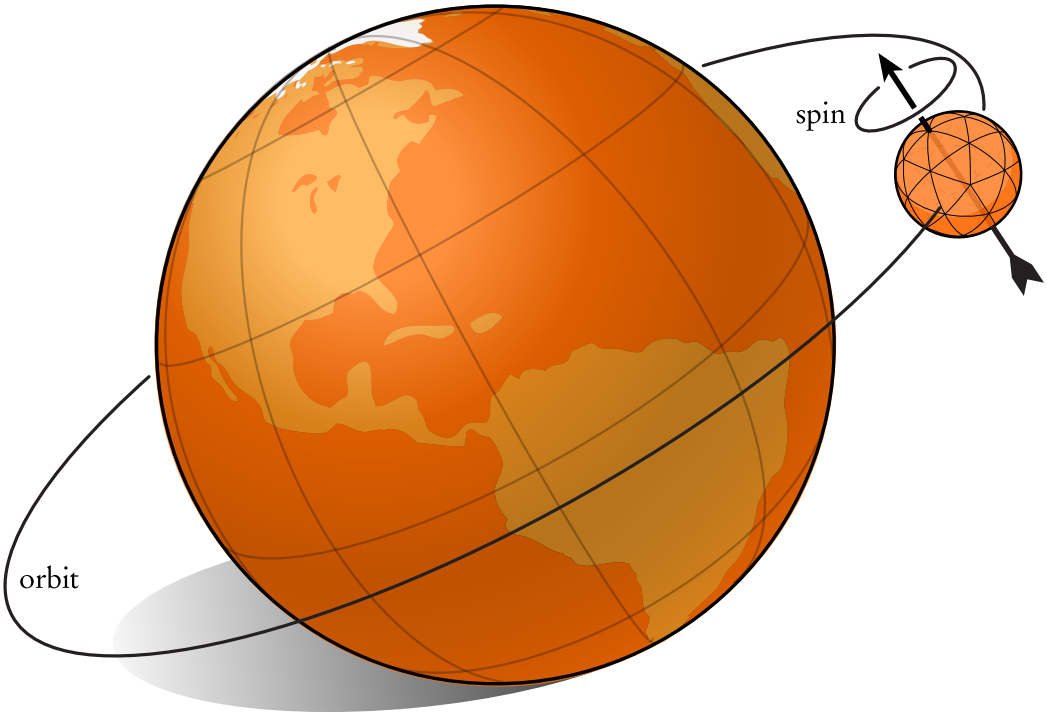
On any conformally immersed surface $f : M \to \mathrm{Im}\mathbb{H}$ we define a differential operator $\mathcal{D}$ on smooth quaternion-valued functions, which we call the *quaternionic Dirac operator*:

$$\mathcal{D}\psi := -\frac{df \wedge d\psi}{|df|^2}. \tag{3.1}$$

Very roughly speaking, the Dirac operator can be thought of as a generalized gradient operator: it describes the first-order derivatives of a given function. In the quaternionic setting, however, the derivative of a function is a function of the same type. For instance, a real-valued function and its gradient vector field can both be expressed in terms of quaternion-valued functions, as discussed in Section 2.7. For this reason, the operator $\mathcal{D}$ provides a unified treatment of differential operators on immersed surfaces, as will be discussed in Section 3.4. More generally, Dirac operators play an important role in physics and geometry, and will be central to our treatment of conformal deformations. In particular, we will be interested in the *time-independent Dirac equation*

$$(\mathcal{D} - \rho)\psi = \gamma\psi \tag{3.2}$$

where ρ is a real-valued function, ψ is a quaternion-valued function of unit norm ($||\psi|| = 1$), and γ is a real eigenvalue.



Physically, the Dirac equation can be thought of as a more accurate version of the more traditional *Schrödinger equation* which models a phenomenon known as *spin-orbit coupling*. To draw a classical analogy: if a Schrödinger equation were used to model the moon orbiting around the Earth, then the corresponding Dirac equation takes into account the fact that the moon is spinning. In both equations total angular momentum is conserved, but in the Dirac equation this momentum is transferred back and forth between two modes: *orbital angular momentum* (the moon traveling around the Earth) and *spin angular momentum* (the moon turning around its own axis). In our particular version of the Dirac equation (Equation 3.2) the function ρ plays the role of a scalar potential, the function ψ describes a steady-state wave function of a spin-1/2 particle interacting with this potential, and the eigenvalue γ represents the corresponding energy level. As with the Schrödinger equation, the magnitude $|\psi|^2$ describes the probability of discovering the particle over any given region of space. As demonstrated in Section 3.2, our quaternionic Dirac operator is locally equivalent to the quantum mechanical Dirac operator for a spin-1/2 particle—this interpretation is used in Section 6.5.1 to visualize (for the first time) relativistic electron wave functions as conformal immersions of the sphere.

3.1 Basic Properties

We first establish some basic properties of the quaternionic Dirac operator $\mathcal{D}$, where as usual we assume that $f : M \to \mathbb{R}^3$ is an immersion of a compact connected orientable surface M which inherits a complex structure $\mathcal{J}$ from the unit normal field on $f(M)$. On *closed* domains ($\partial M = \emptyset$), we show that $\mathcal{D}$ exhibits the behavior expected of any Dirac operator, namely that it is self-adjoint and (weakly) elliptic. These properties are desirable from an applications point of view, because on a compact domain any self-adjoint elliptic operator has an orthonormal eigenbasis and a discrete spectrum of real eigenvalues—hence, we can think of the operator $\mathcal{D}$ in much the same way as we think about a real symmetric matrix. (These properties become especially valuable when we analyze the configuration space of conformal immersions—see Chapter 4.) For surfaces with boundary, the behavior of $\mathcal{D}$ depends on the particular choice of boundary conditions. Motivated by applications described in this thesis, Bohle and Pinkall [BP13] provide a geometric characterization of boundary conditions for which the resulting Dirac operator is self-adjoint and elliptic—these results are summarized in Section 4.5.4.

Proposition 3.1.1. *Let M be closed (i.e., $\partial M = 0$). Then for any two differentiable functions $\psi, \phi : M \to \mathbb{H}$ we have*

$$\langle\langle \mathcal{D}\psi, \phi \rangle\rangle = \langle\langle \psi, \mathcal{D}\phi \rangle\rangle,$$

i.e., $\mathcal{D}$ is self-adjoint.

Proof. We have

$$\langle\langle \psi, \mathcal{D}\phi \rangle\rangle = -\int_M \overline{\psi} \frac{df \wedge d\phi}{|df|^2} |df|^2 = -\int_M \overline{\psi} df \wedge d\phi = \int_M d\overline{\psi} \wedge df\, \phi - \int_M d(\overline{\psi} df\, \phi).$$

By Stokes' theorem the latter term vanishes and we get

$$\int_M d\overline{\psi} \wedge df\, \phi = -\int_M \overline{\overline{\phi} df \wedge d\psi} = \overline{\langle\langle \phi, \mathcal{D}\psi \rangle\rangle} = \langle\langle \mathcal{D}\psi, \phi \rangle\rangle.$$

□

Proposition 3.1.2. *Let M be closed. Then $\mathcal{D}$ is (weakly) elliptic.*

Proof. Let (s_1, s_2) be local coordinates on M. Then for any quaternionic function ψ we have

$$\begin{aligned} df \wedge d\psi &= \left(\frac{\partial f}{\partial s_1} ds^1 + \frac{\partial f}{\partial s_2} ds^2\right) \wedge \left(\frac{\partial \psi}{\partial s_1} ds^1 + \frac{\partial \psi}{\partial s_2} ds^2\right) \\ &= \left(\frac{\partial f}{\partial s_1}\frac{\partial \psi}{\partial s_2} - \frac{\partial f}{\partial s_2}\frac{\partial \psi}{\partial s_1}\right) ds^1 \wedge ds^2, \end{aligned}$$

hence

$$\mathcal{D}\psi = \frac{\partial f}{\partial s_2}\frac{\partial \psi}{\partial s_1} - \frac{\partial f}{\partial s_1}\frac{\partial \psi}{\partial s_2}.$$

Locally, then, the symbol of $\mathcal{D}$ is a linear map $p_{\mathcal{D}} : \mathbb{R}^2 \to \mathbb{H}$ given by

$$p_{\mathcal{D}}(s_1, s_2) = \frac{\partial f}{\partial s_2} s_1 - \frac{\partial f}{\partial s_1} s_2$$

(i.e., we replace each partial derivative operator $\partial/\partial s_i$ with the corresponding coordinate variable s_i). Letting $X = (s_1, s_2)$, we can also write the symbol as

$$p_{\mathcal{D}}(X) = df(\mathcal{J}X).$$

But since f is an immersion, df is nondegenerate. Hence, $\mathcal{D}$ is weakly elliptic. □

3.2 Relationship to Spin Dirac Operator

The Dirac operator for a spin-1/2 particle in the plane is given by

$$-i(\sigma_x \partial_x + \sigma_y \partial_y),$$

where σ_x and σ_y are the *Pauli matrices*

$$\sigma_x = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \; \sigma_y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}.$$

Proposition 3.2.1. *In local coordinates, the quaternionic Dirac operator $\mathcal{D}$ is equivalent to the spin-Dirac operator described above.*

Proof. Note that any function $\psi : M \to \mathbb{H}$ can be written as

$$\psi = \phi_1 + j\phi_2$$

for some pair of $\mathbb{C}$-valued functions ϕ_1, ϕ_2. Define ∂ and $\overline{\partial}$ via

$$\partial\psi := \tfrac{1}{2}(\partial_x - i\partial_y)\psi, \quad \overline{\partial}\psi := \tfrac{1}{2}(\partial_x + i\partial_y)\psi.$$

At any given point, we can choose coordinates such that our immersion f maps to the j,k-plane, i.e., $f = jz$ for some $z : M \to \mathbb{C}$. We then have

$$\begin{aligned} df \wedge d\psi &= j\,dz \wedge ((\partial\phi_1 dz + \overline{\partial}\phi_1 d\overline{z}) + j(\partial\phi_2 dz + \overline{\partial}\phi_2 d\overline{z})) \\ &= (j\overline{\partial}\phi_1 + \partial\phi_2) dz \wedge d\overline{z} \\ &= -2(j\overline{\partial}\phi_1 + \partial\phi_2)(i|df|^2). \end{aligned}$$

Dividing through by $-1/|df|^2$ we get $\mathcal{D}\psi = 2(j\overline{\partial}\phi_1 + \partial\phi_2)i$, which means the matrix of $\mathcal{D}$ is

$$D = 2\begin{bmatrix} 0 & \partial \\ \overline{\partial} & 0 \end{bmatrix} i = i(\sigma_x\partial_x + \sigma_y\partial_y),$$

i.e., our quaternionic Dirac operator is locally the same as the spin Dirac operator up to sign. □

3.3 Relationship to Laplace-Beltrami

Dirac operators are often referred to as "square roots of the Laplacian." Here we make this relationship explicit for the quaternionic Dirac operator $\mathcal{D}$.

Proposition 3.3.1. *Let Δ be the Laplace-Beltrami operator induced by a conformal immersion f. Then*

$$\mathcal{D}^2\psi = \Delta\psi + \frac{dN \wedge d\psi}{|df|^2}$$

for any function $\psi : M \to \mathbb{H}$.

Proof. Recalling the definition of $\mathcal{D}$ (Equation 3.1), we have

$$\begin{aligned} & |df|^2\mathcal{D}\psi &&= -df \wedge d\psi \\ \iff \quad & -df\,df\,\mathcal{D}\psi &&= -df \star d\psi + \underbrace{\star df}_{-df\,N}\, d\psi \\ \iff \quad & df\,\mathcal{D}\psi &&= \star d\psi + N d\psi \end{aligned}$$

$\operatorname{div} X$	$=$	$\operatorname{Re}(N\mathcal{D}X)$	divergence
$\operatorname{grad}\varphi$	$=$	$-N\mathcal{D}\varphi$	gradient
$\operatorname{curl} X$	$=$	$\operatorname{Re}(\mathcal{D}X)$	curl
SX	$=$	$-(\mathcal{D}X)^{\top}$	shape operator
H	$=$	$-\frac{1}{2}N\mathcal{D}N$	mean curvature
K	$=$	$\frac{1}{2}N\operatorname{Im}(\mathcal{D}^2)N$	Gaussian curvature
N	$=$	$-\frac{1}{2}\mathcal{D}f$	unit normal
$I(X,Y)$	$=$	$-\operatorname{Re}(XY)$	first fundamental form
$II(X,Y)$	$=$	$\operatorname{Re}(Y\mathcal{D}X)$	second fundamental form
$\Delta\varphi$	$=$	$\operatorname{Re}(\mathcal{D}^2\varphi)$	scalar Laplacian
ΔX	$=$	$(\mathcal{D}(\mathcal{D}X)^{\perp})^{\top}$	vector Laplacian
$\nabla_X\varphi$	$=$	$\operatorname{Re}(XN\mathcal{D}\varphi)$	covariant derivative

Table 3.1: Standard differential operators expressed using the quaternionic Dirac operator $\mathcal{D}$. Here X and Y are tangent vector fields, φ is a scalar function, and N is the unit normal field.

where in the last step we take advantage of the fact that f is an immersion. Taking the derivative of both sides, we get

$$-df\wedge d(\mathcal{D}\psi) = d\star d\psi + dN\wedge d\psi.$$

Notice that the left-hand side of this expression has the same form as the right-hand side of the first line above, replacing ψ with $\mathcal{D}\psi$. Making this substitution, we get

$$|df|^2\mathcal{D}^2\psi = d\star d\psi + dN\wedge d\psi$$

or equivalently

$$\mathcal{D}^2\psi = \Delta\psi + \frac{dN\wedge d\psi}{|df|^2}.$$

□

Note that we can therefore obtain the Laplacian of a real function φ by extracting the real part of $\mathcal{D}^2\varphi$, i.e., $\Delta\varphi = \operatorname{Re}(\mathcal{D}^2\varphi)$ since $dN\wedge d\varphi$ is a tangent-valued 2-form.

3.4 Vector Analysis

One attractive feature of the quaternionic operator $\mathcal{D}$ is that it can be used to concisely express all other basic differential operators from the theory of surfaces, both intrinsic and extrinsic. This fact will prove useful in later analysis; it also suggests a potential approach to numerical discretization

(Chapter 7).

Consider any quaternionic function $\psi : M \to \mathbb{H}$. If we decompose this function into three components $\psi = a + df(Y) + bN$ (Section 2.7), then a concise way to summarize the effect of $\mathcal{D}$ on ψ is

$$\mathcal{D}\psi = \begin{bmatrix} 0 & -\text{curl} & 0 \\ \mathcal{J}\,\text{grad} & -S & \text{grad} \\ 0 & -\text{div} & 2H \end{bmatrix} \begin{bmatrix} a \\ Y \\ b \end{bmatrix} \tag{3.3}$$

Using this relationship we can express all of the standard differential operators on a surface, as summarized in Table 3.4. The remainder of this section serves to verify these facts. To obtain an expression for each component of the 0-form $\mathcal{D}\psi$ we simply evaluate the 2-form $-df \wedge d\psi$ on a pair of orthogonal vectors $X, \mathcal{J}X$ which have unit length with respect to the induced metric, i.e.,

$$|df(X)| = |df(\mathcal{J}X)| = 1.$$

Proposition 3.4.1. *Let $a : M \to \mathbb{R}$ be a real differentiable function on M. Then*

$$\mathcal{D}a = df(\mathcal{J}\,\text{grad}\, a),$$

i.e., applying the Dirac operator to a real function yields a 90-degree rotation of its gradient (as an immersed vector field).

Proof. We have

$$\begin{aligned} \mathcal{D}a &= -df \wedge da(X, \mathcal{J}X) \\ &= -(df(X)da(\mathcal{J}X) - df(\mathcal{J}X)da(X)) \\ &= Ndf(\mathcal{J}X)da(\mathcal{J}X) + Ndf(X)da(X) \\ &= Ndf(da(X)X + da(\mathcal{J}X)\mathcal{J}X) \\ &= Ndf(\text{grad}\, a) \\ &= df(\mathcal{J}\,\text{grad}\, a). \end{aligned}$$

□

Proposition 3.4.2. *Let Y be a vector field on M. Then*

$$\mathcal{D}\hat{Y} = -(\text{curl}\, Y + df(SY) + (\text{div}\, Y)N).$$

Proof. We have

$$-\mathcal{D}\hat{Y} = df \wedge d\hat{Y}(X, \mathcal{J}X) = df(X)d\hat{Y}(\mathcal{J}X) - df(\mathcal{J}X)d\hat{Y}(X).$$

Remembering that $d\hat{Y}(\mathcal{J}X)$ is just the ambient covariant derivative $\hat{\nabla}_{\widehat{\mathcal{J}X}}\hat{Y}$, the first term becomes

$$\hat{X}\nabla_{\widehat{\mathcal{J}X}}\hat{Y} = \hat{X}(\underbrace{df(\nabla_{\mathcal{J}X}Y)}_{=:\hat{Z}} + II(\mathcal{J}X, Y)N) = \hat{X}(\hat{Z} - g(S\mathcal{J}X, Y)N).$$

Since $\hat{X}$ and $\hat{Z}$ are both tangent vector fields we have $\hat{X}\hat{Z} = \hat{X} \times \hat{Z} - \langle \hat{X}, \hat{Z} \rangle = \langle N\hat{X}, \hat{Z} \rangle N - \langle \hat{X}, \hat{Z} \rangle$. We therefore get

$$\begin{aligned} & \langle N\hat{X}, \hat{Z} \rangle N - \langle \hat{X}, \hat{Z} \rangle - g(S\mathcal{J}X, Y)\hat{X}N \\ = \; & \langle Ndf(X), df(\nabla_{\mathcal{J}X}Y) \rangle N - \langle df(X), df(\nabla_{\mathcal{J}X}Y) \rangle - g(S\mathcal{J}X, Y)df(X)N \\ = \; & g(\mathcal{J}X, \nabla_{\mathcal{J}X}Y)N - g(X, \nabla_{\mathcal{J}X}Y) + g(S\mathcal{J}X, Y)df(\mathcal{J}X). \end{aligned}$$

Similarly, the second term of our initial expression for $\mathcal{D}\hat{Y}$ can be written as

$$df(\mathcal{J}X)d\hat{Y}(X) = -g(X, \nabla_X Y)N - g(\mathcal{J}X, \nabla_X Y) - g(SX, Y)df(X).$$

Taking the difference of these two terms we get a real part

$$g(\mathcal{J}X, \nabla_X Y) - g(X, \nabla_{\mathcal{J}X}Y) = (\text{curl } Y)N,$$

a tangential part

$$\begin{aligned} & g(SX, Y)df(X) + g(S\mathcal{J}X, Y)df(\mathcal{J}X) \\ = \; & df(g(SY, X)X + g(SY, \mathcal{J}X)\mathcal{J}X) \\ = \; & df(SY), \end{aligned}$$

and a normal part

$$(g(X, \nabla_X Y) + g(\mathcal{J}X, \nabla_{\mathcal{J}X}Y))N = (\text{div } Y)N.$$

All together, we get

$$-\mathcal{D}\hat{Y} = \text{curl } Y + df(SY) + (\text{div } Y)N.$$

□

Proposition 3.4.3. *Let N be the unit normal field on $f(M)$. Then*

$$\mathcal{D}N = -2HN.$$

Proof. Recall that the directions of principal curvature X_1, X_2 are orthonormal with respect to the induced metric, and are eigenvectors of the shape operator S with associated eigenvalues κ_1, κ_2, respectively. We therefore have

$$\begin{aligned} -\mathcal{D}N &= df \wedge dN(X_1, X_2) \\ &= df(X_1)dN(X_2) - df(X_2)dN(X_1) \\ &= df(X_1)df(SX_2) - df(X_2)df(SX_1) \\ &= \kappa_2 df(X_1)df(X_2) - \kappa_1 df(X_2)df(X_1) \\ &= 2H(df(X_1) \times df(X_2)) \\ &= 2HN. \end{aligned}$$

□

Proposition 3.4.4. *Let b be a real function on M. Then*

$$\mathcal{D}(bN) = df(\text{grad } b) + 2bHN.$$

Proof. This result follows immediately from Propositions 3.4.1 and 3.4.3. □

The propositions above effectively determine the 3×3 matrix form of $\mathcal{D}$ (Equation 3.3) and thereby verify the first five identities in Table 3.4; the remainder of this section serves to verify the rest.

Proposition 3.4.5. *The Gaussian curvature of a conformally immersed surface f can be expressed as*

$$K = \frac{1}{2} N(\text{Im}\mathcal{D}^2)N.$$

Proof. From Proposition 3.3.1, we have

$$\text{Im}(\mathcal{D}^2)N = \frac{dN \wedge dN}{|df|^2},$$

and evaluating the 2-form $dN \wedge dN$ on the principal directions X_1, X_2 yields

$$\begin{aligned} dN \wedge dN(X_1, X_2) &= dN(X_1)dN(X_2) - dN(X_2)dN(X_1) \\ &= 2dN(X_1)dN(X_2) \\ &= 2\kappa_1\kappa_2 df(X)df(\mathcal{J}X) \\ &= 2KN|df|^2. \end{aligned}$$

Multiplying through by $N/2|df|^2$ yields the result. □

Proposition 3.4.6. *The unit normal field N on a conformally immersed surface f can be expressed as*

$$N = -\frac{1}{2}\mathcal{D}f.$$

Proof. We have

$$df \wedge df(X, \mathcal{J}X) = df(X)df(\mathcal{J}X) - df(\mathcal{J}X)df(X) = 2df(X)df(\mathcal{J}X) = -2N|df(X)|^2.$$

Hence,

$$\mathcal{D}f = -\frac{2N|df|^2}{|df|^2} = -2N.$$

□

Proposition 3.4.7. *The first fundamental form of a conformally immersed surface f can be expressed as*

$$I(X, Y) = -\mathrm{Re}(\hat{X}\hat{Y}).$$

Proof. We have

$$\mathrm{Re}(\hat{X}\hat{Y}) = \mathrm{Re}(\hat{X} \times \hat{Y} - \langle \hat{X}, \hat{Y} \rangle) = -\langle df(X), df(Y) \rangle = -I(X, Y).$$

□

Proposition 3.4.8. *Let X and Y be any two vector fields on M. The second fundamental form induced by a conformal immersion f can be expressed as*

$$II(X, Y) = \mathrm{Re}(\hat{Y}\mathcal{D}\hat{X}).$$

Proof. From Proposition 3.4.2, we have

$$\mathcal{D}\hat{X} = -(\operatorname{curl} X + df(SX) + (\operatorname{div} X)N).$$

Since both $\hat{Y}(\operatorname{curl} X)$ and $\hat{Y}(\operatorname{div} X)N$ are tangent-valued, we are left with just

$$\begin{aligned} \operatorname{Re}(\hat{Y}\, df(SX)) &= \operatorname{Re}(\hat{Y} \times df(SX) - \langle \hat{Y}, df(SX)\rangle) \\ &= -\langle \hat{Y}, df(SX)\rangle \\ &= -\langle df(Y), dN(X)\rangle \\ &= II(X,Y). \end{aligned}$$

□

Proposition 3.4.9. *Let $\hat{X}$ be a tangent vector field on a conformally immersed surface f. Then*

$$\Delta\hat{X} = (\mathcal{D}(\mathcal{D}\hat{X})^{\perp})^{\top}$$

Proof. From Proposition 3.4.2 we get

$$(\mathcal{D}\hat{X})^{\perp} = (\operatorname{div} X)N,$$

and from Proposition 3.4.4 we get

$$\mathcal{D}((\operatorname{div} X)N)^{\top} = df(\operatorname{grad} \circ \operatorname{div} X) = \Delta\hat{X}.$$

□

Proposition 3.4.10. *Let φ be a real-valued function on M and let X be a tangent vector field. Then*

$$\nabla_X \varphi = \operatorname{Re}(\hat{X} N \mathcal{D}\varphi).$$

Proof. From Proposition 3.4.1 we have $\mathcal{D}\varphi = df(\mathcal{J}\operatorname{grad}\varphi)$, hence

$$\hat{X} N \mathcal{D}\varphi = \hat{X} N df(\mathcal{J}\operatorname{grad}\varphi) = -\hat{X} N df(\operatorname{grad}\varphi).$$

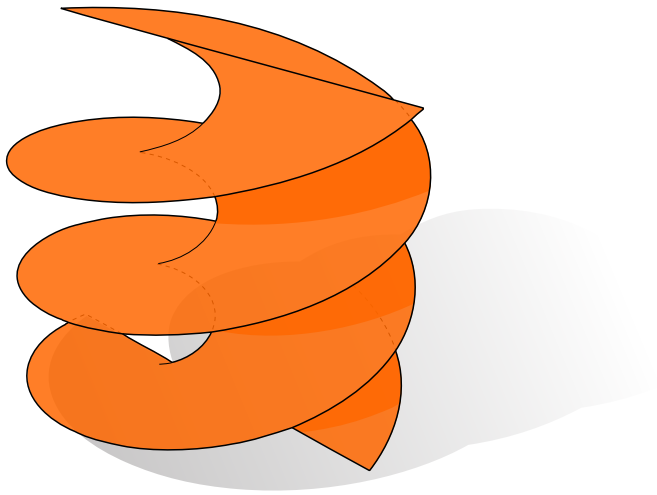
We then get

$$-\mathrm{Re}(\hat{X}\,N\mathcal{D}\varphi) = \langle \hat{X}, df(\mathrm{grad}\,\varphi)\rangle = \nabla_X \varphi.$$

□

Chapter 4

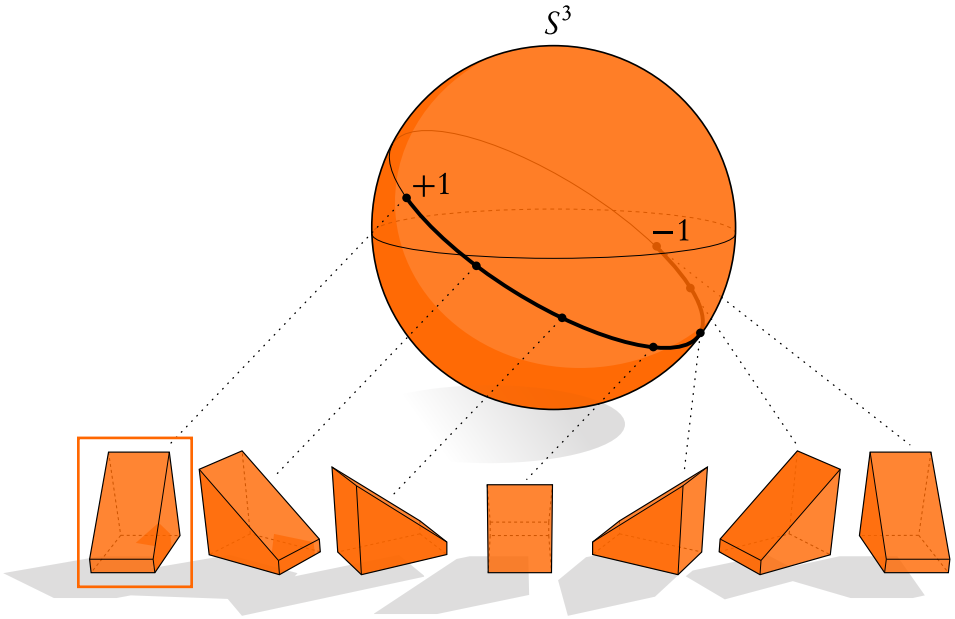
The Shape Space of Conformal Immersions



Given a surface in space, one can easily determine its curvature. A natural question is: given a curvature function, when can one uniquely determine the corresponding surface? This chapter investigates this question in the context of a conformal immersion f (Section 2.3) and its associated mean curvature half-density μ (Section 2.4). In particular, we show how to recover f by integrating μ, and describe the precise conditions under which μ must evolve in order to preserve integrability. The space of valid functions μ itself turns out to be (almost everywhere) an infinite dimensional Riemannian manifold which—in spirit—has the structure of the helicoid depicted above. We refer to this space as the *shape space of conformal immersions*. Understanding this manifold is essential to the development of algorithms because it allows one to successfully "navigate" the space, i.e., to find trajectories of curvature that induce the desired operations on the surface itself. In the following section we give a very informal discussion of shape spaces and how they relate to our framework for working with surfaces—for a more formal discussion, see the book by Younes [You10].

4.1 Shape Spaces

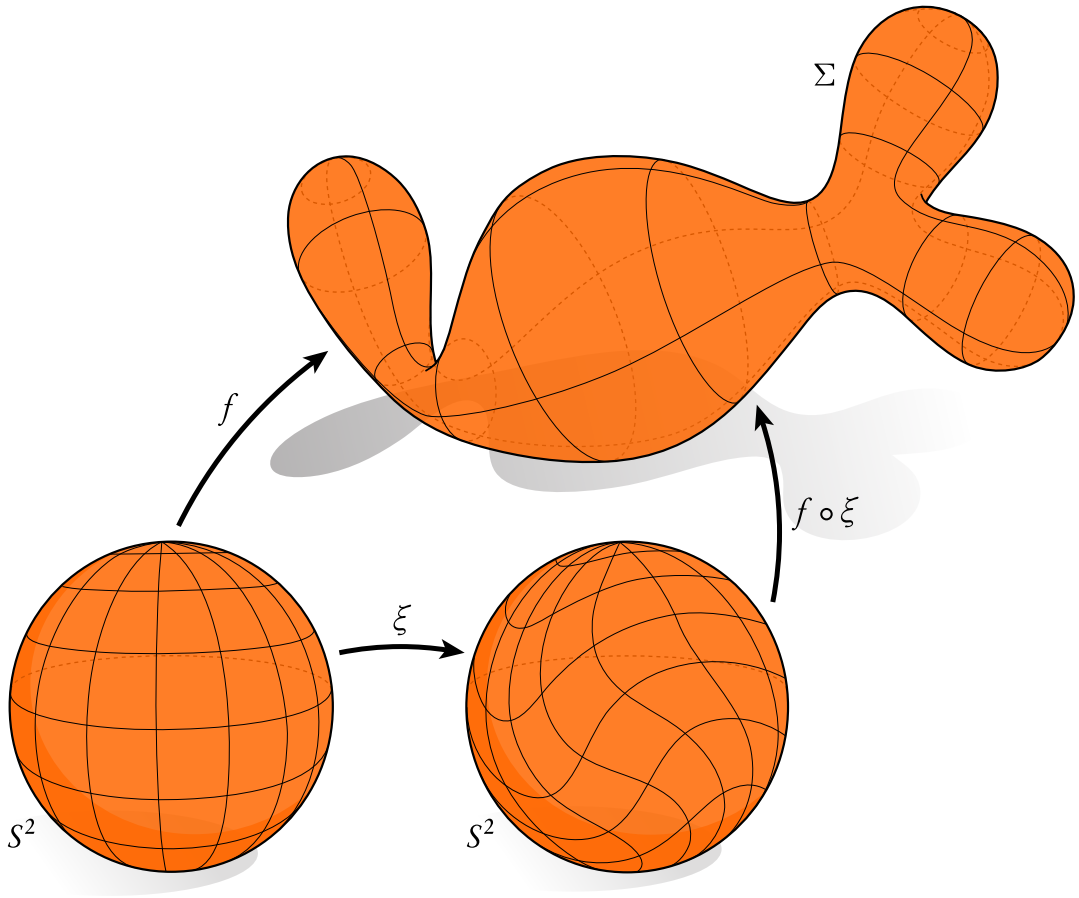
Shape spaces are, generally speaking, configuration spaces associated with a given piece of geometry, such as a curve in the plane or a surface in space. Shape spaces can be thought of in analogy with simple configuration spaces appearing in mechanics. For instance, the orientation of a rigid body spinning in space can be identified with a rotation relative to some initial, canonical configuration. To build an explicit description of this space, recall that every unit quaternion $q \in \mathbb{H} = \mathbb{R}^4$ determines a rotation $x \mapsto \overline{q}xq$ of vectors $x \in \mathrm{Im}\mathbb{H} = \mathbb{R}^3$ (Section 2.5). Geometrically, then, the configuration space of a spinning rigid body looks like the collection of unit vectors in $\mathbb{R}^4$—in other words, the *3-sphere* S^3:



The nice thing about this description is that S^3 has a concrete geometry, which allows one to talk clearly about things like "shortest paths of rotations" and "derivatives of rotations," etc. As emphasized in the cartoon above, however, using S^3 to represent rotations is in some sense redundant. In particular, any two quaternions q and $-q$ describe the same rotation: $\overline{q}xq = (-\overline{q})x(-q)$. To make this representation unique, we could "glue together" opposite points on the sphere, i.e., we could take the quotient $S^3/\pm 1$, yielding the *real projective space* $\mathbb{R}P^3$. This set is again a geometric space, endowed with natural notions of length, tangent vectors, etc. In practice, however, it is often more convenient to work with the "redundant" space S^3, which can be easily visualized and has a nice algebraic representation (namely, the quaternions).

The story about rigid bodies captures some of the essential features one considers when formulating a shape space for surfaces. For instance, consider a smooth submanifold $\Sigma \subset \mathbb{R}^3$ with spherical

topology. The subset Σ can always be expressed as the image of a general immersion $f : S^2 \to \mathbb{R}^3$, i.e., we can always find an f such that $f(S^2) = \Sigma$. And, just as quaternions were a redundant representation of orientations, immersions are a redundant representation of shape. For surfaces, the redundancy comes from the fact that we can first "swirl the sphere around" before mapping it into space. More precisely, if $\xi : S^2 \to S^2$ is any diffeomorphism of the sphere, then the map $f \circ \xi$ is another immersion describing Σ, i.e., since $\xi(S^2) = S^2$, we have $f(\xi(S^2)) = f(S^2) = \Sigma$:



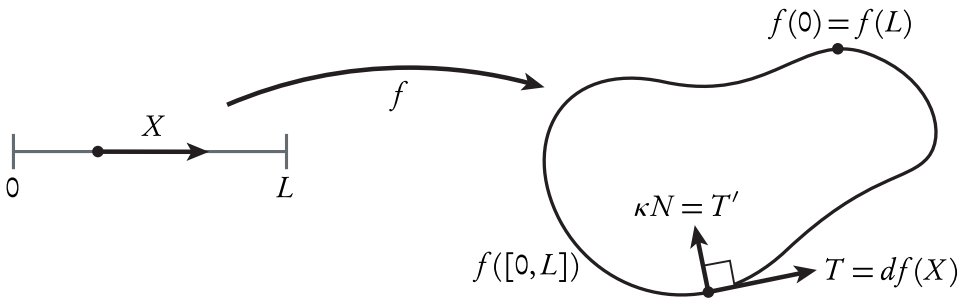
This description of surfaces exhibits a *reparameterization symmetry*: it does not matter which points on S^2 get mapped to which points of $\mathbb{R}^3$, as long as the whole sphere S^2 gets mapped to the whole surface Σ. Moreover, this representation exhibits something that, *very roughly*, resembles a vector space structure, i.e., one can often get away with pretending that the sum of two immersions is an immersion. This approach reflects a standard practice in geometry processing: surfaces are represented via Euclidean coordinates at mesh vertices, which get manipulated as though they are arbitrary $\mathbb{R}^3$-valued functions—there are typically no special constraints to ensure that the surface remains immersed, embedded, etc. The ability to work in a linear space is essential from an application perspective, because

it allows one to take advantage of efficient computational tools like numerical linear algebra.

As discussed in Section 2.3, we can also use a *conformal* immersion $f : M \to \mathbb{R}^3$ to describe any surface. This representation exhibits a much simpler reparameterization symmetry: if two conformal immersions describe the same surface, then they differ only up to a conformal diffeomorphism of the sphere, i.e., a *Möbius transformation*. Unlike the full space of diffeomorphisms $\mathrm{Diff}(S^2)$ which is infinite dimensional and hence quite challenging to discretize, the space of Möbius transformations is finite dimensional (six dimensional, in fact) and hence much more manageable from a numerical point of view. For instance, it is computationally feasible to compare two functions by searching over the Möbius group, as done, for example, by Lipman and Daubechies [LD11]. However, one important question remains: how does one represent the conformal immersions themselves? The immersion function f is an inconvenient choice, since the collection of such functions looks nothing like a vector space, i.e., in most cases the sum of two conformal immersions does not even come close to being conformal. (This fact is quite salient in the discrete setting: take the sum of two surfaces with "nice" texture coordinates and the result will not be so nice!)

An important observation made by Kamberov et al. [KPP98] is that, apart from a few exceptional cases, a global conformal immersion is uniquely determined by its *mean curvature half-density* $\mu = H|df|$. Yet unlike conformal immersions f, half-densities *do* have a vector space structure, as discussed in Section 2.4. This structure is the essential feature that allows us to work with conformal transformations in geometry processing: although we cannot "add together" two conformal immersions, we *can* add the corresponding half-densities and easily recover a surface that resembles both of them in a way that is highly intuitive. In this thesis, we provide a detailed analysis of the corresponding shape space $\mathcal{M}$ (Section 4.5). Notably, $\mathcal{M}$ also exhibits a conformal reparameterization symmetry (e.g., Möbius transformations in the case of S^2), an observation which may prove valuable in applications like shape matching. In most applications, however, it is more convenient to work in the "redundant" space of half-densities, which avoids the introduction of additional singularities in the quotient space.

4.2 Curves and Isometry: A Prelude to Surfaces



Before developing a shape space for surfaces, we examine the much simpler case of curves, which helps illustrate the general method for establishing integrability conditions in terms of curvature. Namely, we write down each relevant constraint in terms of positions or tangents, differentiate it in time, then convert it to a constraint on curvature using established relationships. The only difference in the case of surfaces is that these relationships become more intricate.

As discussed in Section 2.3, we view any planar curve as the image of an arc-length or *isometric* immersion

$$f : [0, L] \to \mathbb{R}^2$$

where $L \in \mathbb{R}$ is the total length of the curve, and isometry means that length is locally preserved, i.e.,

$$|df(X)| = |X|$$

for all tangent vectors X. Associated with this curve is a unit tangent field $T := f'$ and corresponding curvature normal $\kappa N = T'$ (see above); we use $d\ell$ to denote the usual length element on the real line.

Which curvature functions $\kappa : [0, L] \to \mathbb{R}$ describe a valid curve? In the case where f is an *open* curve (i.e., no constraints on the endpoints), it is fairly easy to see that any such function is valid: simply integrate κ to get the unit tangents T, then integrate T to get new positions $\tilde{f}$. More explicitly, if we let θ be the cumulative angle

$$\theta(s) := \theta_0 + \int_0^s \kappa d\ell,$$

then the new unit tangent field is given by $\tilde{T}(s) = (\cos\theta(s), \sin\theta(s))$. We can then recover the positions $\tilde{f}$ via

$$\tilde{f}(s) = \tilde{f}_0 + \int_0^s \tilde{T} d\ell.$$

Note that the curve $\tilde{f}$ is determined only up to a global rotation and translation, specified by values $\theta_0 \in \mathbb{R}$ and $f_0 \in \mathbb{R}^2$. By construction, this new curve will also be isometrically immersed, since we always integrate *unit* tangents $\tilde{T}$.

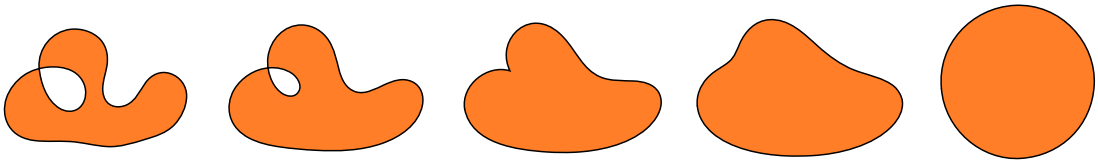
The situation becomes more interesting when we consider *closed, regular* curves. Closed means that the two endpoints meet:

$$f(0) = f(L) \tag{4.1}$$

and regular means that tangents agree at endpoints:

$$T(0) = T(L). \tag{4.2}$$

Regularity prohibits motions that develop sharp "kinks", as seen in the center image below:



The main task now is to re-express these conditions in terms of curvature. More precisely, suppose we start out with a valid curvature function κ (which we can obtain, for instance, by simply differentiating our initial curve). What then are the valid *changes* in curvature $\dot{\kappa}$ as our curve evolves in time? The latter condition (Equation 4.2) implies that the tangent turns around a whole number of times as we walk around the curve. Equivalently, it says that the total curvature is an integer multiple of 2π, i.e.,

$$\int_0^L \kappa d\ell = 2\pi \mathsf{k}$$

for some *turning number* $\mathsf{k} \in \mathbb{Z}$. If κ satisfies this condition at time $t = 0$, it will remain satisfied as long as the total curvature does not change, i.e., as long as

$$\int_0^L \dot{\kappa} d\ell = 0.$$

The first condition (Equation 4.1) is not quite as easy to reformulate, but a straightforward derivation shows that it can also be expressed as a simple linear condition on κ:

Proposition 4.2.1. *Let $\kappa(t)$ be the curvature of an isometrically immersed curve f, varying in time.*

Then

$$\int_0^L \dot{\kappa} f \, d\ell = 0.$$

Proof. The endpoints $f(0)$ and $f(L)$ agree as long as the unit tangent field T integrates to zero:

$$\int_0^L T \, d\ell = 0.$$

Since T is a unit vector at each point, both $\dot{T}$ and T' are orthogonal to T. Differentiating in time therefore yields

$$0 = \int_0^L \dot{T} \, d\ell = \int_0^L \langle JT, \dot{T} \rangle JT \, d\ell,$$

where here J denotes a 90-degree rotation in the plane. Recalling that $f' = T$ and $T' = \kappa N$, we get

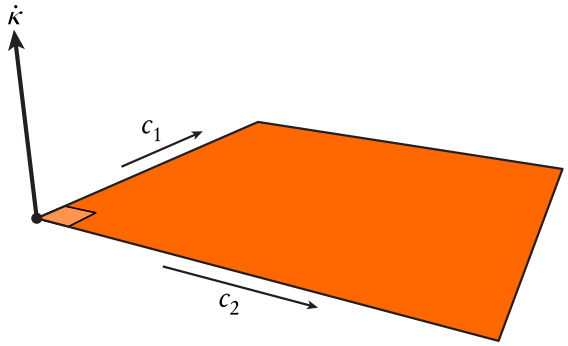
$$\begin{aligned} 0 = J \int_0^L \langle JT, \dot{T} \rangle f' \, d\ell &= -J \int_0^L (\cancelto{0}{\langle JT', \dot{T} \rangle} + \langle JT, \dot{T}' \rangle) f \, d\ell \\ &= -J \int_0^L (\langle JT, \dot{\kappa} N \rangle + \cancelto{0}{\langle JT, \kappa \dot{N} \rangle}) f \, d\ell = -J \int_0^L \dot{\kappa} f \, d\ell, \end{aligned}$$

where on the first line we apply integration by parts and the fundamental theorem of calculus. □

A concise way to write the whole collection of constraints is then

$$\langle\langle \dot{\kappa}, \mathbf{1} \rangle\rangle = \langle\langle \dot{\kappa}, f^x \rangle\rangle = \langle\langle \dot{\kappa}, f^y \rangle\rangle = 0,$$

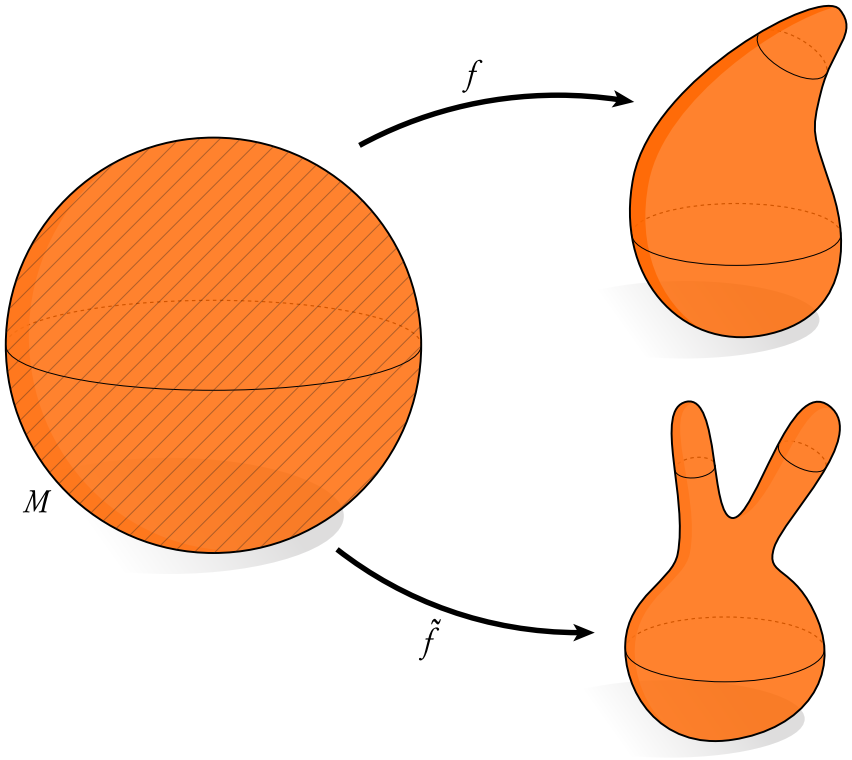
where $f^x, f^y : [0, L] \to \mathbb{R}$ are the current $x-$ and $y-$ coordinate functions of the curve, respectively, and $\langle\langle \cdot, \cdot \rangle\rangle$ is the $\mathscr{L}^2$ inner product on functions (Section 2.1).



In terms of shape spaces, this analysis allows us to adopt the following picture. Imagine that κ is a point in the space of all smooth functions on the interval $[0, L]$. We can move κ in any direction $\dot{\kappa}$,

except the three directions c_1, c_2, c_3 described by the three linear constraints above. Since the space of valid curvature functions *locally* looks like a linear space, one is naturally inclined to wonder whether these spaces are in fact *tangent spaces* of some global Riemannian manifold. This question will be considered in greater detail when we consider the case of conformally immersed surfaces (Section 4.5).

4.3 Spin Transformations



Consider two general immersions $f, \tilde{f} : M \to \mathrm{Im}\mathbb{H}$ of a topological surface M, as depicted above. These surfaces are said to be *spin equivalent* if their differentials $df, d\tilde{f}$ are related by a similarity transformation at each point, i.e., if there exists some non-vanishing function $\lambda : M \to \mathbb{H}$ such that

$$d\tilde{f}(X) = \overline{\lambda} df(X) \lambda \tag{4.3}$$

for all tangent vectors X. Importantly, the Riemannian metrics induced by two such immersions will be related by a positive scaling:

$$\begin{aligned}\tilde{g}(X,Y) &= \langle d\tilde{f}(X), d\tilde{f}(Y)\rangle \\ &= \langle \overline{\lambda} df(X)\lambda, \overline{\lambda} df(Y)\lambda\rangle \\ &= |\lambda|^4 \langle df(X), df(Y)\rangle \\ &= \underbrace{|\lambda|^4}_{>0} g(X,Y).\end{aligned}$$

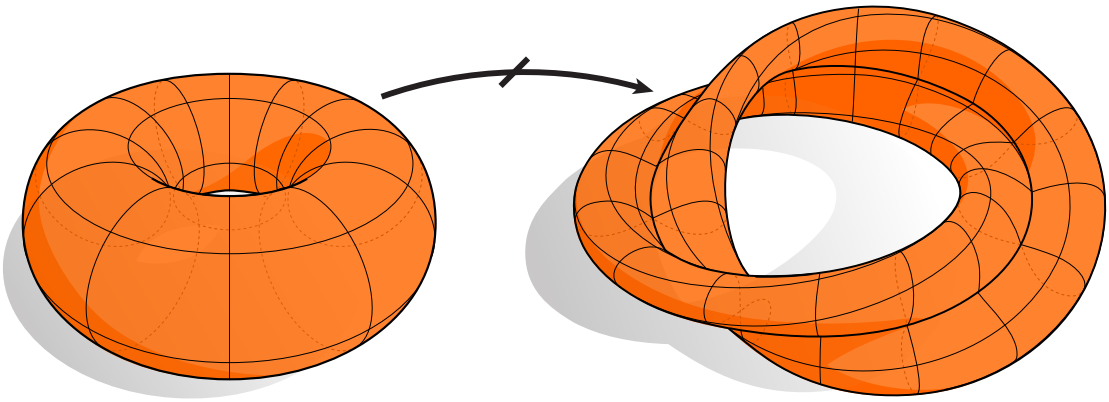
In other words, spin equivalence implies conformal equivalence. However, not every pair of conformally equivalent immersions are related by a spin transformation—in particular, Kamberov et al. [KPP98, Remark 2] make the following observations:

1. When M is simply connected, any two conformal immersions $f, \tilde{f}$ are spin equivalent.
2. When M is not simply connected, f and $\tilde{f}$ are spin equivalent if and only if they are regularly homotopic.

Inspired in part by numerical experiments done for this thesis (Section 6.5.4), Pinkall [Pin] makes a further conjecture:

Conjecture. *(Pinkall) Let $f, \tilde{f} : M \to \mathbb{R}^3$ be two regularly homotopic conformal immersions. Then f and $\tilde{f}$ are conformally homotopic in the sense that there exists a regular homotopy between f and $\tilde{f}$ that is conformal throughout.*

These facts help us understand what types of deformations can be achieved via spin transformations, and what cannot. For instance, one can achieve any conformal deformation of the sphere S^2 but *cannot* achieve certain conformal deformations of the torus T^2. However, these deformations correspond to large, global, non-injective "twists", which are not typically needed for geometry processing:



In practice, the most typical scenario is that we start with a surface f and wish to find a spin transformation $\tilde{f}$ that expresses a relatively small, local change in geometry (e.g., for editing, smoothing, etc.). However, an *arbitrary* transformation λ will not, in general, produce a 1-form $d\tilde{f}$ that is the differential of some surface $\tilde{f}$, i.e., $\overline{\lambda} df \lambda$ may not be *integrable*. In other words, we cannot construct the new surface $\tilde{f}$ unless λ satisfies the very special criterion expressed in Equation 4.3. Solving this equation directly appears intractable, since computationally it amounts to a large system of *quadratic* equations. Fortunately, there is an alterative, *linear* condition on λ that ensures integrability:

Proposition 4.3.1. *Let $f : M \to \mathrm{Im}\mathbb{H}$ be a conformal immersion of a simply connected domain M. Then the transformed differential $\beta := \overline{\lambda} df \lambda$ is integrable if and only if λ satisfies the condition*

$$(\mathcal{D} - \rho)\lambda = 0 \tag{4.4}$$

for some real-valued function $\rho : M \to \mathbb{R}$, where $\mathcal{D}$ is the quaternionic Dirac operator induced by f.

Proof. (This proof follows Kamberov et al. [KPP98, Lemma 2.2].) On a simply connected domain, β is exact (i.e., integrable) as long as it is closed, from which we get

$$0 = d(\overline{\lambda} df \lambda) = d\overline{\lambda} \wedge df \lambda - \overline{\lambda} df \wedge d\lambda = \overline{\overline{\lambda} df \wedge d\lambda} - \overline{\lambda} df \wedge d\lambda.$$

But for any quaternion $q \in \mathbb{H}$, $q - \overline{q} = 2\mathrm{Im}(q)$. The expression above therefore says that the 2-form $\overline{\lambda} df \wedge d\lambda$ is purely real, or in other words, $\overline{\lambda} df \wedge d\lambda = \hat{\rho} |df|^2$ for some real function $\hat{\rho}$. Equivalently, we can require that $-df \wedge d\lambda = \rho \lambda |df|^2$ for a different real-valued function ρ which is related to $\hat{\rho}$ by $\rho = -\hat{\rho}/|\lambda|^2$. Substituting $\mathcal{D}$ into this final expression yields $(\mathcal{D} - \rho)\lambda = 0$. □

At first glance, Equation 4.4 appears difficult to solve: we have to *simultaneously* find a function ρ and a similarity transformation λ. However, suppose we start with an arbitrary scalar function ρ. If we solve the eigenvalue problem

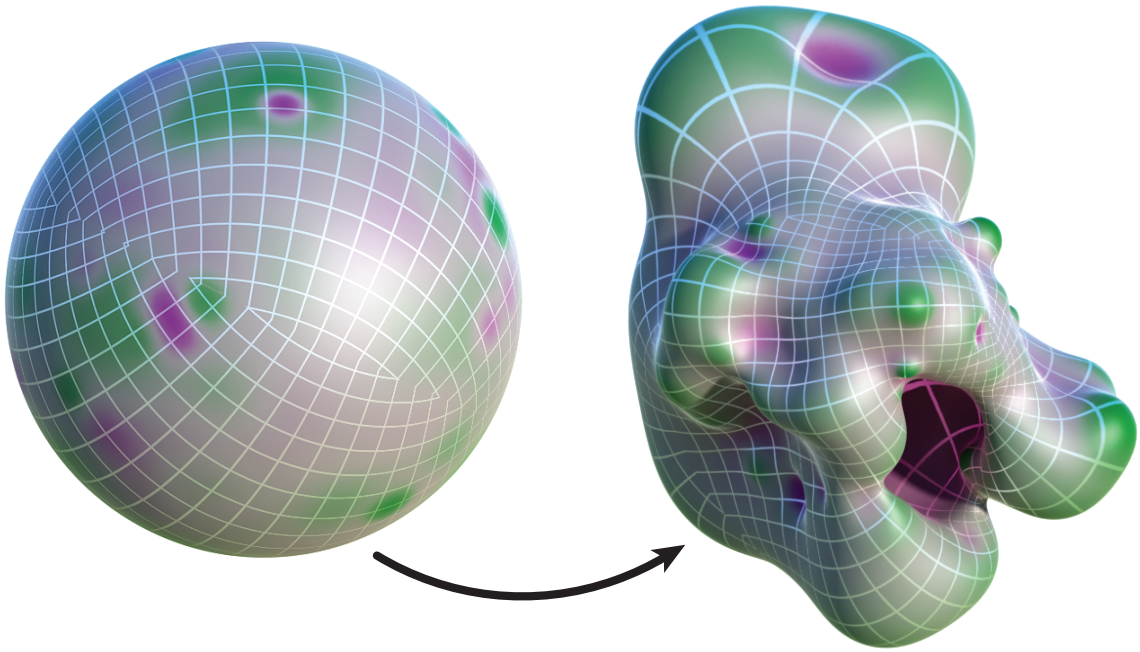
$$(\mathcal{D} - \rho)\lambda = \gamma\lambda \tag{4.5}$$

for any eigenvector λ and corresponding eigenvalue γ, it follows that the pair $(\lambda, \rho+\gamma)$ satisfies Equation 4.4, i.e., $(\mathcal{D} - (\rho + \gamma))\lambda = 0$. In other words, *any* scalar function ρ determines a conformal immersion f, as long as we allow ρ to be shifted by a global constant. Overall, then, the following procedure determines a spin transformation $\tilde{f}$ on a simply connected domain M:

1. Pick a scalar function ρ on M.
2. Solve an eigenvalue problem (Equation 4.5) for the similarity transformation λ.
3. Solve a Poisson equation (Equation 4.3) for the new surface $\tilde{f}$.

This procedure represents the main computational task in our conformal geometry processing algorithms—a numerical version is developed in Chapter 5. The nonsimply connected case is discussed in Section 4.5.2.

4.4 Curvature Potential



In the procedure above deformations are specified via the scalar function ρ. What is the geometric meaning of ρ? Recall the *mean curvature half-density* $\mu = H|df|$ induced by an immersion f (Section 2.4), which provides a scale-invariant description of the smoothness or roughness of the surface. The mean curvature half-density of a spin transformation $\tilde{f}$ is given by

$$\widetilde{H}|d\tilde{f}| = H|df| + \rho|df|, \tag{4.6}$$

where $\widetilde{H}$ is the mean curvature induced by $\tilde{f}$ (see Kamberov et al. [KPP98, Lemma 2.3]). Thus, the scalar function ρ controls the *change* in mean curvature half-density—we will call this function the *curvature potential*. As illustrated in the image above, the resulting deformations behave somewhat like normal displacements: positive values, indicated by green spots, cause the surface to bulge out; negative values, indicated by purple spots, cause the surface to bulge in. Unlike normal displacements, however, the conformal structure of the surface is preserved, as illustrated by the fact that grid lines remain orthogonal. Also note that the effect of applying a constant shift to the curvature potential ρ (as in Equation 4.5) is akin to adding a constant value to a grayscale image—the resulting appearance is largely the same.

4.5 Conformal Shape Space

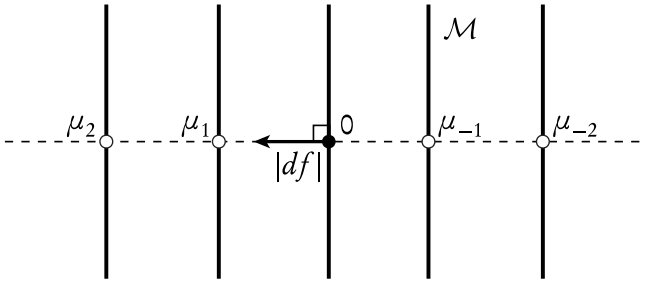
Definition 4.5.1. *Let $f : M \to \mathbb{R}^3$ be a conformal immersion of a closed surface M. The* conformal shape space $\mathcal{M}(f)$ *(or just $\mathcal{M}$ when the immersion is understood) is the collection of all mean curvature half-densities that can be achieved via some spin transformation $\tilde{f}$ of f:*

$$\mathcal{M}(f) := \left\{ \rho|df| \;\middle|\; \rho|df| = \widetilde{H}|d\tilde{f}| - H|df|, \tilde{f} : M \to \mathbb{R}^3 \right\}.$$

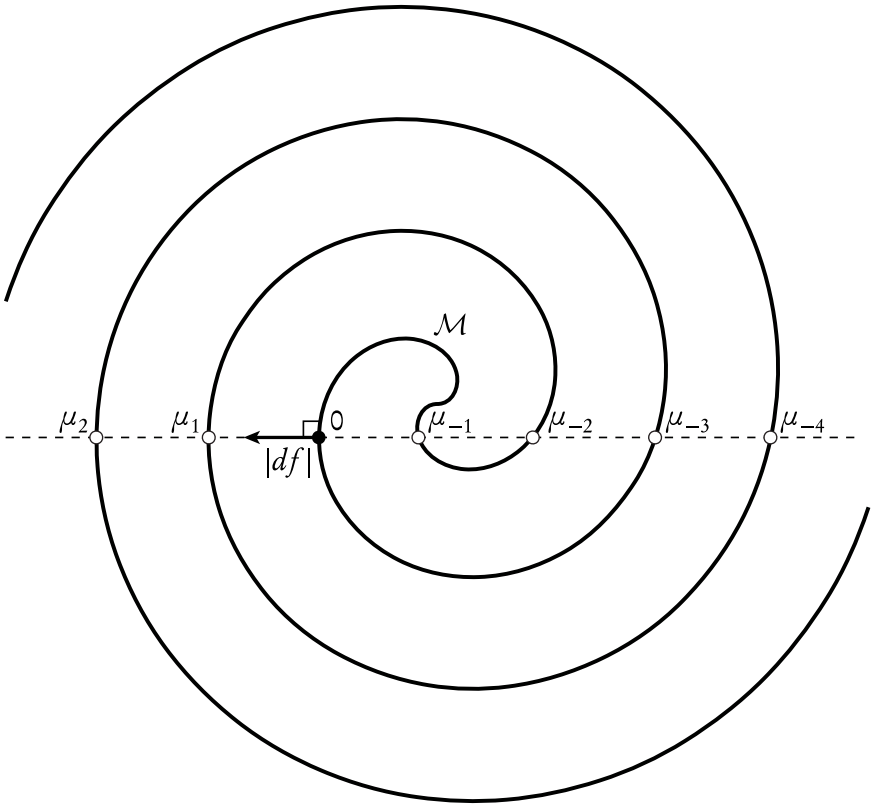
Note that since spin equivalence is in fact an equivalence relation, $\mathcal{M}$ depends only on the conformal structure and regular homotopy class induced by f, i.e., for any spin transformation $\tilde{f}$ of f we get $\mathcal{M}(\tilde{f}) = \mathcal{M}(f)$. In the case of the sphere S^2 we can drop the immersion f altogether, since all spheres are spin equivalent.

To better understand the structure of this space, it helps to consider special points $\rho|df| \in \mathcal{M}$. For instance, $\mathcal{M}(f)$ always contains the origin $\rho|df| = 0$, since $\mathcal{D}\lambda = 0$ for any constant function $\lambda \equiv q \in \mathbb{H}$. Geometrically, this statement reflects the fact that any global rigid rotation and uniform scale $\overline{q}dfq$ is a spin transformation.

More can be said about the global structure of shape space when the domain M is closed and simply connected, i.e., when $M = S^2$. In this case, Equation 4.5 indicates that if $\rho|df| \in \mathcal{M}$ then $(\rho + \gamma)|df| \in \mathcal{M}$ for any eigenvalue γ of $\mathcal{D}$. But as implied by Propositions 3.1.1 and 3.1.2, $\mathcal{D}$ has only a *discrete* collection of eigenvalues γ, which we index by the integers $\mathbb{Z}$, i.e., $\ldots, \gamma_{-1}, \gamma_0, \gamma_1, \ldots$. In terms of the shape space $\mathcal{M}$, we can then imagine that, starting out at our initial surface $\mu_0 = 0$ we can travel a finite distance γ_i along the constant direction $|df|$ to find another surface $\mu_i := \gamma_i|df|$. Locally, then, $\mathcal{M}$ must look like a discrete collection of "sheets:"



We also know that $\mathcal{M}$ must be connected, since all spheres are spin equivalent. One could therefore imagine that globally these sheets come together to form a single Euler spiral:



The pictures above are, of course, just "cartoons" of $\mathcal{M}$, meant only to provide some intuition about

conformal shape space (in general, for instance, sheets will not be evenly spaced). Nonetheless, this intuition strongly suggests that $\mathcal{M}$ itself will turn out to be a Riemannian manifold. A formal proof of this fact is likely quite technical and will not be pursued in the present thesis. However, the basic intuition is as follows. Let $\mathcal{D}_f$ denote the Dirac operator induced by f. Points in $\mathcal{M}$ correspond to half-densities $\rho|df|$ such that the operator $\mathcal{D}_f - \rho$ has a kernel (Equation 4.4), i.e., points where $\det(\mathcal{D}_f - \rho) = 0$ for a suitably defined determinant det. The function $\rho|df| \mapsto \det(\mathcal{D}_f - \rho)$ can be viewed as a differential map from $\mathcal{H}$ into $\mathbb{R}$; at points where $\mathcal{D}$ does not have repeated eigenvalues this map is a submersion and hence $\mathcal{M}$ is locally a hypersurface, i.e., a codimension-1 submanifold in the space of half-densities $\mathcal{H}$. Generically, $\mathcal{D}_f$ will not have repeated eigenvalues—this kind of degeneracy might corresponds to immersions f with a high degree of symmetry, for instance (Section 6.5.1). In this setting, $\mathcal{M}$ inherits the Riemannian metric induced by the inner product $\langle\langle , \rangle\rangle$ on $\mathcal{H}$.

4.5.1 Local Analysis

For applications where we consider, e.g., the continuous time evolution of a surface, it will also prove valuable to understand the *local* structure of $\mathcal{M}$, i.e., which local variations of the mean curvature half-density μ are admissible? In particular, we have the following characterization of the tangent space $T\mathcal{M}$:

Proposition 4.5.1. *Let λ, ρ be time-varying solutions to Equation 4.4 with $\lambda(0) = \mathbf{1}$ and $\rho(0) = 0$. Then*

$$\mathcal{D}\dot{\lambda} = \dot{\rho}. \tag{4.7}$$

Proof. We can rewrite Equation 4.4 as $\mathcal{D}\lambda = \rho\lambda$, and since $\mathcal{D}$ is constant with respect to t, differentiating this relationship in time yields just

$$\mathcal{D}\dot{\lambda} = \dot{\rho}\overset{1}{\cancel{\lambda}} + \overset{0}{\cancel{\rho}}\dot{\lambda} = \dot{\rho}.$$

□

Proposition 4.5.2. *Let M be a closed surface and let f be any immersion such that the induced Dirac operator $\mathcal{D}$ has a one-dimensional kernel. Then the tangent space of $\mathcal{M}$ at the origin consists of half-densities ξ with zero mean, i.e.,*

$$\frac{1}{\mathcal{A}}\langle\langle \mathbf{1}|df|, \xi \rangle\rangle = 0.$$

Proof. Any tangent vector can be viewed as the velocity $\xi = \dot{\rho}|df|$ of some curve $\rho(t)$ at time $t = 0$, where $\rho(0) = 0$. Integrating Equation 4.7 over M and applying Stokes' theorem, we get

$$\langle\langle \dot{\rho}, \mathbf{1} \rangle\rangle = \int_M \dot{\rho}|df|^2 = \int_M \mathcal{D}\dot{\lambda}|df|^2 = \int_M df \wedge d\,\dot{\lambda} = \int_M d(f\,d\,\dot{\lambda}) = \int_{\partial M} f\,d\,\dot{\lambda},$$

which equals zero since $\partial M = \emptyset$. □

Using this description of the tangent spaces, we can easily determine the normal direction on $\mathcal{M}$:

Proposition 4.5.3. *At regular points of $\mathcal{M}$, the unit normal is given by*

$$\nu := \frac{1}{\sqrt{\mathcal{A}}}|df|.$$

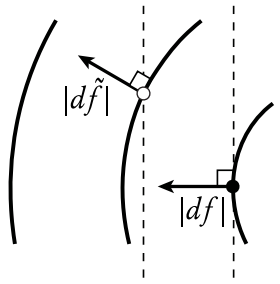
Proof. At regular points, $\mathcal{M}$ is a hypersurface with tangent spaces spanned by all half-densities ξ with zero mean, i.e.,

$$\langle\langle \mathbf{1}|df|, \xi \rangle\rangle = 0.$$

But since $\mathbf{1}|df|$ is orthogonal to all tangent vectors, it must be the unique normal direction; moreover, it has length

$$||\mathbf{1}|df|\,|| = \left(\int_M |df|^2 \right)^{1/2} = \sqrt{\mathcal{A}}.$$

□

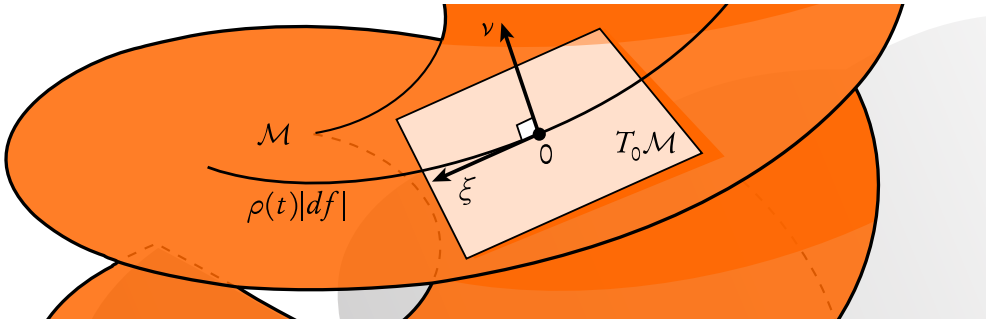


Note in particular that the normals of $\mathcal{M}$ corresponding to any two spin equivalent immersions $f, \tilde{f}$ are consistently oriented (i.e., they are contained within the same "half-space") since

$$\langle\langle \tilde{\nu}, \nu \rangle\rangle = \langle\langle |d\tilde{f}|, |df| \rangle\rangle = \langle\langle |\overline{\lambda} df\, \lambda|, |df| \rangle\rangle = \langle\langle |\lambda|^2 |df|, |df| \rangle\rangle = \int_M |\lambda|^2 |df|^2 > 0.$$

For this reason, a more appropriate cartoon for $\mathcal{M}$ than the Euler spiral (where normals can point in any direction) is the helicoid mentioned at the beginning of this chapter (where all normals point

"upwards"). Just as the 3-sphere provided a nice visualization for rotations, the helicoid provides a good mental model for the shape space of conformal immersions. In particular, to analyze the local behavior of shape space we will typically consider a time-varying curvature potential $\rho(t)$ with $\rho(0) = 0$, which determines a trajectory $\rho(t)|df|$ of half-densities passing through the origin:



At time $t = 0$, this curve corresponds to the initial immersion f since $\rho(0) = 0$, i.e., no change in curvature. At any other time t, we can associate the trajectory $\rho(t)$ with the induced similarity transformation $\lambda(t)$ and resulting spin transformation $\tilde{f}(t)$, i.e., λ satisfies the integrability relationship $(\mathcal{D} - \rho(t))\lambda(t) = 0$ (Equation 4.4) and $\tilde{f}$ satisfies the spin equivalence relationship $d\tilde{f}(t) = \overline{\lambda}(t)df\,\lambda(t)$. In particular, we have $\lambda(0) = 1$ and $\tilde{f}(0) = f$. From here we can analyze how a change $\xi := \dot{\rho}|df|$ in mean curvature half-density induces an infinitesimal change $\dot{\tilde{f}}$ in the resulting surface, or alternatively, how a desired motion of the surface can be expressed in terms of curvature. Importantly, we can *always* perform this analysis at the origin $\rho|df| = 0$, since (as discussed at the beginning of this section) we can always re-express $\mathcal{M}(f)$ with respect to any spin equivalent immersion $\tilde{f}$, i.e., we can "translate" shape space so that the immersion of interest sits at the origin. These conventions are worth understanding, since they will be adopted without further mention in later derivations—we also remind the reader that the notation $\dot{\varphi}$ is used to indicate the time derivative *at time zero* (Section 2.1).

The helicoid is also a good picture of shape space because it captures the fact that $\mathcal{M}$ is a *hypersurface* (i.e., a codimension-1 submanifold) which, despite being infinite dimensional, behaves in many ways like an ordinary surface in $\mathbb{R}^3$. As one example we compute the shape operator on $\mathcal{M}$:

Proposition 4.5.4. *At regular points of $\mathcal{M}$, the shape operator $\mathcal{S} : T\mathcal{M} \to T\mathcal{M}$ acts on any tangent vector $\xi = \dot{\rho}|df|$ via*

$$\mathcal{S}(\xi) = 2\mathcal{A}^{-1/2}\mathrm{Re}(\dot{\lambda}),$$

where $\dot{\lambda}$ is the corresponding change in λ, i.e., $\mathcal{D}\dot{\lambda} = \dot{\rho}$.

Proof. Recall from Proposition 4.5.3 that the unit normal is given by $\nu = \mathcal{A}^{-1/2}|df|$. The shape operator is simply the derivative of the unit normal along the direction of interest. In other words, imagine that $\nu(t)$ is the shape space normal associated with the trajectory $\rho(t)|df|$. Then

$$\mathcal{S}(\xi) = \dot{\nu} = \frac{d}{dt}\left(\mathcal{A}^{-1/2}|d\tilde{f}(t)|\right)\Big|_{t=0}.$$

Since ρ determines the immersion only up to global scale, we can normalize f such that $\mathcal{A}$ remains constant with respect to t. Recalling that $d\tilde{f} = \overline{\lambda} df \lambda$ (Equation 4.3) and hence $|d\tilde{f}| = |\lambda|^2|df|$, we get

$$\mathcal{S}(\xi) = \mathcal{A}^{-1/2}\left(\overset{\cdot}{\overline{\lambda}}\overset{1}{\lambda} + \overset{1}{\overline{\lambda}}\dot{\lambda}\right)|df| = 2\mathcal{A}^{-1/2}\mathrm{Re}(\dot{\lambda})|df|,$$

where in the final step we use the fact that $q + \overline{q} = 2\mathrm{Re}(q)$ for any $q \in \mathbb{H}$. □

The shape operator can be used to verify a fact about shape space that becomes particularly important in the context of Willmore flow (Section 6.4), namely that the round sphere is a stable critical point:

Theorem 4.5.1. *Let $f : S^2 \to \mathbb{R}^3$ describe the unit sphere in $\mathbb{R}^3$, i.e., for all $x \in f(S^2)$, $|x|^2 = 1$. Then $\mathcal{M}(f)$ is convex at the origin.*

Proof. Let $\{\phi_k^m\}$ be an orthonormal basis of Laplacian eigenfunctions on the unit sphere with corresponding eigenvalues $c_k = k(k+1)$; m is used to index eigenfunctions sharing the same eigenvalue. Since the eigenbasis spans all of $\mathscr{L}^2(S^2, \mathbb{R})$ (spectral theorem) and since any tangent vector in $T_0\mathcal{M}$ has zero mean (Proposition 4.5.2), we need only consider the action of $\mathcal{S}$ on each of the ϕ_k^m, *excluding* the constant function ϕ_0. Let $\dot{\lambda}$ be the change in the similarity transformation λ induced by motion along the direction ϕ_k^m, i.e., $\mathcal{D}\dot{\lambda} = \phi_k^m$. As discussed in Section 2.7, we can decompose $\dot{\lambda}$ as

$$\dot{\lambda} = a + df(Y) + bN,$$

where a and b are real functions and Y is a tangent vector field. From Equation 3.3, we then have

$$(\mathcal{D}\dot{\lambda})^\top = \mathcal{J}\,\mathrm{grad}\, a - df(SY) + \mathrm{grad}\, b.$$

Note that on the unit sphere $N = f$ and we get $df(SX) = dN(X) = df(X)$, i.e., the shape operator

is just $S = \mathrm{id}$. Since $0 = (\phi_k^m)^\top = (\mathcal{D}\dot{\lambda})^\top$, the previous equation becomes

$$\hat{Y} = \mathcal{J}\,\mathrm{grad}\, a + \mathrm{grad}\, b$$

and again applying Equation 3.3, we have

$$\phi_k^m = \mathrm{Re}(\mathcal{D}\dot{\lambda}) = -\mathrm{curl}\, Y = \mathrm{div}\, \mathcal{J}Y = \mathrm{div}\, \mathcal{J}^2 \mathrm{grad}\, a + \cancelto{0}{\mathrm{curl}\,\mathrm{grad}\, b} = -\mathrm{div}\,\mathrm{grad}\, a = \Delta a.$$

Hence,

$$\mathrm{Re}(\dot{\lambda}) = a = \Delta^{-1}\phi_k^m = \frac{1}{k(k+1)}\phi_k^m$$

and since the area of the unit sphere is $\mathcal{A} = 4\pi$, we get

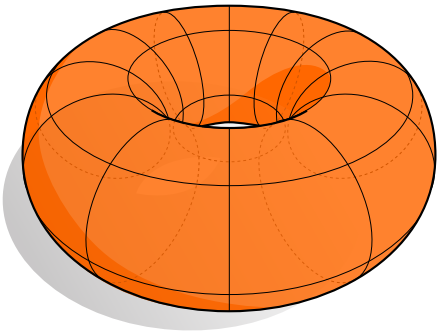
$$\mathcal{S}\phi_k^m = \frac{1}{k(k+1)\sqrt{\pi}}\phi_k^m.$$

In other words, the normal curvature associated with the direction ϕ_k^m is just

$$\kappa(\phi_k^m) = \frac{1}{k(k+1)\sqrt{\pi}},$$

since ϕ_k^m has unit norm with respect to the inner product on $\mathcal{H}$. But then curvature is positive in all directions, since $\kappa(\phi_k^m) > 0$ for all k, m. □

4.5.2 Nontrivial Topology



The integrability condition $(\mathcal{D} - \rho)\lambda = 0$ established in Section 4.3 fails when M is not simply connected (e.g., when M is a torus), since in general not every closed 1-form is exact. At present, there is no global characterization of the shape space $\mathcal{M}$ for nonsimply connected surfaces M. However, we can continue with the local analysis initiated in Section 4.5.1, i.e., starting with a mean curvature half-

density μ induced by an immersed surface, we can determine the local variations in μ that preserve integrability. In practice, this approach allows us to compute flows on surfaces of arbitrary topological type (Section 6.4); large deformations can then be achieved by flowing towards the desired curvature potential ρ. The key idea is to think of an exact 1-form as a closed 1-form with a vanishing harmonic component. One can then translate a condition on the transformed differential $\overline{\lambda} df \lambda$ to a constraint on the induced change in the curvature potential $\dot{\rho}$:

Proposition 4.5.5. *Let $\{\upsilon_i\}$ be a basis for harmonic vector fields on an immersed surface $f(M)$. Then any tangent vector $\dot{\rho}|df| \in T_0\mathcal{M}(f)$ satisfies*

$$\langle\langle \dot{\rho}, Z_i^x \rangle\rangle = \langle\langle \dot{\rho}, Z_i^y \rangle\rangle = \langle\langle \dot{\rho}, Z_i^z \rangle\rangle = 0$$

for all i*, where* $Z_i := \mathcal{D}^{-1}\upsilon_i$ *and* Z_i^x, Z_i^y, Z_i^z *denote the three imaginary components of* Z_i*. (Note that this statement is still vacuously true whenever M has trivial topology.)*

Proof. Recall Equation 4.3, which effectively says that a transformation λ is integrable as long as the resulting differential $\beta := \overline{\lambda} df \lambda$ is exact, i.e., $\beta = d\tilde{f}$ for some new immersion $\tilde{f}$. In particular, β must have a vanishing harmonic component. Let $\{\omega_i\}$ be an orthonormal basis for *real*-valued harmonic 1-forms on a closed surface M, and note that $\mathbb{H}$-linear combinations of these bases span all $\mathbb{H}$-valued harmonic 1-forms. To preserve integrability, it is therefore sufficient to ensure that $\langle\langle \dot{\beta}, \omega_i \rangle\rangle = 0$ for all i. Differentiating β with respect to time yields

$$\dot{\beta} = \dot{\overline{\lambda}} df + df \dot{\lambda} = 2\mathrm{Im}(df \dot{\lambda}),$$

which means that $\dot{\lambda}$ must satisfy $\langle\langle \mathrm{Im}(df \dot{\lambda}), \omega_i \rangle\rangle = 0$, or equivalently

$$\mathrm{Im} \int_M \star\omega_i \wedge df \dot{\lambda} = 0$$

for all i. Evaluating this integrand yields

$$\begin{aligned} \star\omega_i \wedge df(X, JX) &= \omega_i(JX) df(JX) + \omega_i(X) df(X) \\ &= df(\underbrace{\omega_i(X)X + \omega_i(JX)JX}_{=:Y_i}) = df(Y_i) =: \upsilon_i, \end{aligned}$$

where Y_i is a harmonic vector field on M and υ_i is its pushforward to the immersed surface. The

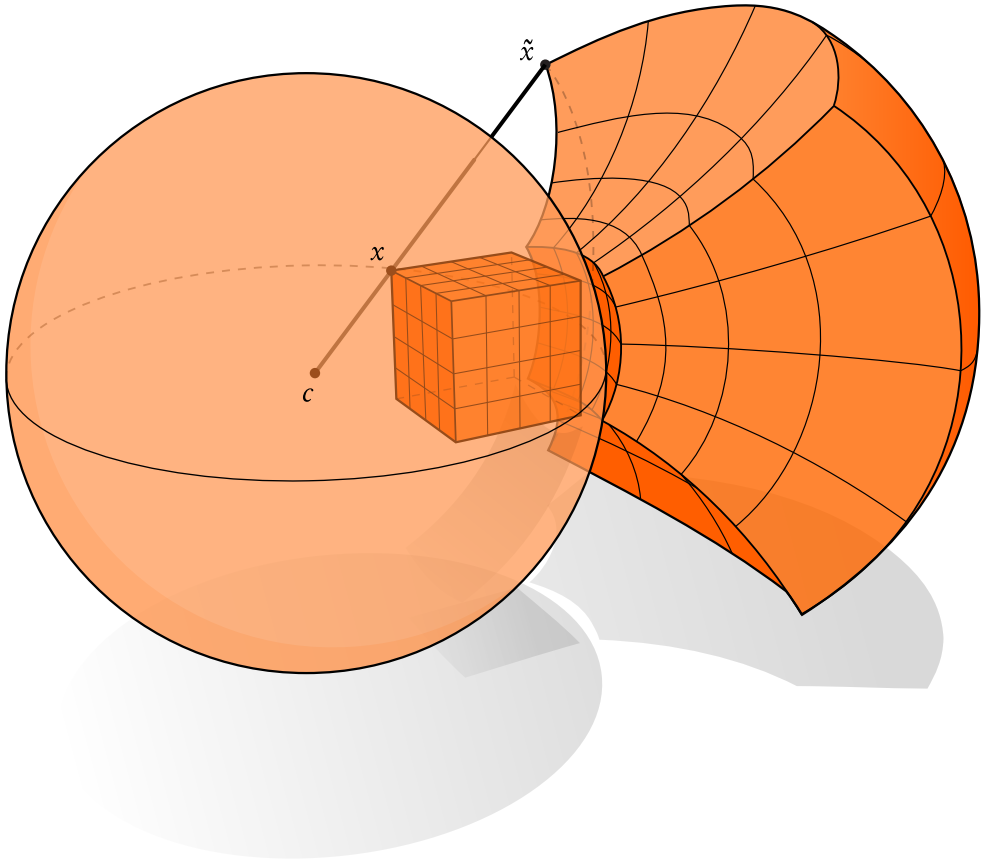
induced integrability condition on $\dot{\lambda}$ is then just $\mathrm{Im}\langle\langle \dot{\lambda}, \upsilon_i \rangle\rangle = 0$ for all harmonic bases υ_i.

To find the equivalent condition on $\dot{\rho}$, consider the solutions Z_i to $\mathcal{D}Z_i = \upsilon_i$. A quaternion has no imaginary component if its scalar product with all vectors $a \in \mathbb{R}^3$ is zero, hence our condition on $\dot{\lambda}$ becomes

$$\begin{aligned} 0 &= \langle a, \textstyle\int_M \dot{\lambda}\upsilon_i \rangle = -\int_M \langle \upsilon_i a, \dot{\lambda} \rangle = -\int_M \langle \mathcal{D}Z_i a, \dot{\lambda} \rangle \\ &= -\textstyle\int_M \langle Z_i a, \mathcal{D}\dot{\lambda} \rangle = -\int_M \langle Z_i a, \dot{\rho} \rangle = \int_M \langle Z_i, a \rangle \dot{\rho}. \end{aligned}$$

where we have used the self-adjointness of $\mathcal{D}$ and the fact that $\mathcal{D}\dot{\lambda} = \dot{\rho}$. In other words, $\dot{\rho}$ must be orthogonal to each of the three imaginary components of each of the functions Z_i. □

4.5.3 Sphere Inversions



Consider the unit sphere with center c in $\mathbb{R}^n$. An *inversion* in this sphere is the map

$$x \mapsto c + \frac{x - c}{|x - c|^2} =: \tilde{x},$$

i.e., relative to the center of the sphere, each vector $x - c$ gets mapped to a parallel vector of reciprocal length; note that points on the sphere get mapped to themselves. Sphere inversions of $\mathbb{R}^3$ are of interest in conformal geometry processing because (i) they induce a conformal deformation of any immersed surface, and (ii) because they preserve the *Willmore energy* of the surface (Section 6.4). In practice, we may wish to "factor out" sphere inversions from a given deformation, or alternatively, use them to reduce area distortion while preserving conformal structure (Section 6.4.4). To do so, we must understand how sphere inversions induce a change in mean curvature half-density.

Proposition 4.5.6. *Let $f : M \to \mathbb{R}^3$ be a conformal immersion and consider the "double inversion"*

$$f \mapsto (f^{-1} - c)^{-1}$$

where here we view f and c as quaternions, i.e., as points in $\mathrm{Im}\mathbb{H}$*. In other words, we first invert f in the unit sphere centered at the origin, then invert it in a sphere centered at c (which restores the original orientation). If the sphere center $c(t)$ varies in time, with $\dot{c} = a \in \mathbb{R}^3$ at time zero, then the induced change in the curvature potential at time zero is*

$$\dot{\rho} = -2\langle a, N\rangle$$

where N is the unit normal.

Proof. Applying the identity $\frac{d}{dt}q^{-1} = -q^{-1}\dot{q}q^{-1}$ (for any $q \in \mathbb{H}$), we get

$$\frac{d}{dt}(f^{-1} - c)^{-1}|_{t=0} = faf.$$

Hence

$$d\dot{f} = df(af) = (fa)df.$$

But since $df(t) = \overline{\lambda}(t)df(0)\lambda(t)$ and $\lambda(0) = 1$, we also have

$$d\dot{f} = df\,\dot{\lambda} + \dot{\overline{\lambda}}df.$$

Together these statements imply that $\dot{\lambda} = af$.

The resulting change in curvature $\dot{\rho}$ is determined via the relationship $\mathcal{D}\dot{\lambda} = \dot{\rho}$ (Equation 4.7). In

particular, let $b \in \mathbb{R}^3$ be the component of a orthogonal to N so that $a = b + \langle a, N\rangle N$. Then

$$\begin{aligned} df \wedge bdf(X, \mathcal{J}X) &= df(X)bdf(\mathcal{J}X) - df(\mathcal{J}X)bdf(X) \\ &= df(X)bNdf(X) - Ndf(X)bdf(X) \\ &= df(X)(bN + Nb)df(X) \\ &= 0, \end{aligned}$$

since $bN = -Nb$ (Equation 2.2). Hence $\mathcal{D}(bf) = 0$ and we find that any curvature change corresponding to an inversion has the form

$$\dot{\rho} = \mathcal{D}\dot{\lambda} = -\frac{df \wedge \langle a, N\rangle N df}{|df|^2} = \frac{2N|df|^2}{|df|^2}\langle a, N\rangle N = -2\langle a, N\rangle.$$

□

4.5.3.1 Möbius Balancing

By applying the result of Proposition 4.5.6, we can find the sphere inversion that moves a given surface towards a desired distribution of area, up to uniform global scaling. In particular, if dA is the desired metric, then we have $|df|^2 = e^{2u}dA$ for some conformal factor u. Since the metric of the transformed surface is

$$|d\tilde{f}|^2 = |\lambda|^4|df|^2,$$

the change in scale factor is

$$\dot{u} = \tfrac{1}{2}\tfrac{d}{dt}\log|\lambda|^4 = 2\mathrm{Re}(\dot{\lambda}).$$

For a sphere inversion, $\dot{\lambda} = af$ for some $a \in \mathbb{R}^3$ (Proposition 4.5.6). Hence, $\dot{u} = -2\langle a, f\rangle$. The optimal direction for a is found by solving

$$\max_{|a|_\mu = 1} \langle\langle \dot{u}, \mathrm{mean}(u) - u\rangle\rangle,$$

where $|\cdot|_\mu$ is the appropriate norm, described below. In other words, we want to move away from our current scale factors u and toward uniform factors $\mathrm{mean}(u)$ as quickly as possible. Letting $u^0 :=$

$u - \text{mean}(u)$, the objective can be expressed as

$$\langle\langle \langle a, f \rangle, u^0 \rangle\rangle = \int_M \langle a, f \rangle u^0 = \langle a, \underbrace{\int_M u^0 f}_{:= v \in \mathbb{R}^3} \rangle = \langle a, v \rangle,$$

where we omit a constant factor 2. Since the direction a determines a change in mean curvature half-density, the value of $|a|_\mu$ is given not by the usual Euclidean norm on $\mathbb{R}^3$ but rather the $\mathscr{L}^2$ norm of the induced function $\dot{\rho} = -2\langle a, N \rangle$. In particular,

$$|a|_\mu^2 := 4 \int_M \langle a, N \rangle^2 = \langle Ba, a \rangle,$$

where $B \in \mathbb{R}^{3\times 3}$ equals $B = 4 \int_M N \otimes N$. We then want to solve

$$\max_{a \in \mathbb{R}^3} \langle a, v \rangle \quad \text{s.t.} \quad \langle Ba, a \rangle = 1,$$

which amounts to solving $Ba = v$, then normalizing a w.r.t. B. A discrete version of this procedure will be applied in the context of Willmore flow (Section 6.4).

4.5.3.2 The Canonical Metric

In the previous section we looked at a curvature flow that attempts to achieve a given distribution of area distortion, via sphere inversions. In practice, how is this distribution determined? The answer depends largely on the application, but in general there are two attractive choices: we can either try to match the area distribution of some reference surface, or we can use the *canonical metric* [Hua06].

Definition 4.5.2. *Let M be a closed surface of nontrivial topology ($\pi_1(M) \neq 0$), and let $\{\omega_i\}$ be an orthonormal basis for real harmonic 1-forms on M, as in Proposition 4.5.5. The* canonical metric *on M is the positive-definite bilinear operator $g_0 : TM \times TM \to \mathbb{R}$ determined by*

$$g_M(X, X) = \sum_{i=1}^{2G} \omega_i(X)^2.$$

The notation g_M is suggestive of the fact that the canonical metric depends only on the global topology of M and not any particular immersion f. (Note, however, that this metric is distinct from the uniformized metric of constant curvature.) Nonetheless, it will be convenient to have an expression for g_M that can be easily evaluated for an immersed surface f:

Proposition 4.5.7. *Let g be the Riemannian metric induced by an immersion $f : M \to \mathbb{R}^3$ of a closed surface M of genus G, and let $y_k : M \to \mathbb{R}^3$ denote the tangent vector field on $f(M)$ representing the harmonic 1-form ω_k, i.e., $\omega_k(X) = \langle y_k, df(X)\rangle$ for all X. Then*

$$g_M = u\, g$$

where u is the positive function

$$u := \frac{1}{2}\sum_{i=1}^{2G} |y_i|^2.$$

In other words, g can be expressed as a conformal rescaling of g_M via the conformal factor u.

Proof. Suppose we order the y_k such that

$$y_{2k+1} = N \times y_{2k}.$$

Then we have

$$g_M(X,X) = \sum_{i=1}^{2G} \omega_i(X)^2 = \sum_{i=1}^{2G} \langle y_i, df(X)\rangle^2.$$

Since y_{2k} and y_{2k+1} are orthonormal, we have

$$\langle y_{2k}, df(X)\rangle^2 + \langle y_{2k+1}, df(X)\rangle^2 = |y_{2k}|^2 |df(X)|^2.$$

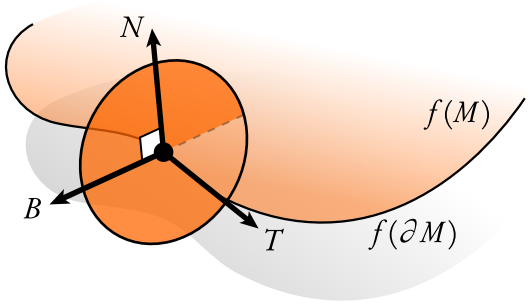
Hence,

$$g_M(X,X) = \sum_{i=1}^{G} |y_{2i}|^2 |df(X)|^2 = \frac{1}{2}\left(\sum_{i=1}^{2G} |y_i|^2\right) |df(X)|^2.$$

But note that $|df(X)|^2 = \langle df(X), df(X)\rangle = g(X,X)$. □

In Section 6.5.4 we use the canonical metric to generate canonical immersions of complex tori.

4.5.4 Boundary Conditions



So far we have considered the shape space $\mathcal{M}$ for surfaces M without boundary. Bohle and Pinkall describe boundary conditions for which the operator $\mathcal{D}$ is self-adjoint and elliptic, and hence for which Equation 4.4 still provides a meaningful notion of integrability [BP13]. In particular, let $f : M \to \mathbb{R}^3$ be an immersion of a surface M with boundary ∂M. Along the boundary curve $f(\partial M)$ there is a natural coordinate frame consisting of the unit normal N, unit vector T tangent to the boundary, and *binormal* $B := T \times N$, as depicted above. Let U be any linear combination of N, T, and B, and suppose we want to construct a spin transformation $\tilde{f}$ of f such that U gets mapped to a prescribed vector $\tilde{U}$. If θ is the angle between U and $\tilde{U}$ and w is the unit normal of their shared plane, then the function

$$\lambda_\partial := (\cos\tfrac{\theta}{2} - w \sin\tfrac{\theta}{2})(a - b\tilde{U})$$

maps U to $\tilde{U}$ for any pair of values $a, b \in \mathbb{R}$. Suppose, then, that in Equation 4.4 we adopt boundary conditions $\lambda|_{\partial M} = \lambda_\partial$, where $a, b : \partial M \to \mathbb{R}$ are undetermined functions. The operator appearing in this problem will be

- *elliptic* if U does not coincide with the directions $\pm N$, and
- *self-adjoint* if U is perpendicular to T.

Examples of surfaces constructed using these conditions are given in Section 6.5.3. In terms of the conformal shape space, we have the following characterization of $T\mathcal{M}$ for surfaces with prescribed binormals:

Proposition 4.5.8. *Let f be a conformal immersion of a topological disk M with prescribed binormal directions $\tilde{B}$ along the boundary. Then any tangent vector $\dot{\rho}|df| \in T_0\mathcal{M}(f)$ has zero mean.*

Proof. Along the boundary, we have

$$\dot{\lambda} = \dot{\alpha} + \dot{\beta}\tilde{B}$$

for real-valued time-varying functions α, β. In other words, the only admissible motion of λ is a gradual rotation around the prescribed binormal direction $\widetilde{B}$. Noting that $\mathcal{D}\dot{\lambda} = \dot{\rho}$ (Proposition 4.5.1), we have

$$\int_M \dot{\rho}|df|^2 = \int_M \mathcal{D}\dot{\lambda}|df|^2 = -\int_M df \wedge d\dot{\lambda} = \int_M d(df\,\dot{\lambda}) = \int_{\partial M} df\,\dot{\lambda}.$$

Note that $T = df(X)$ for any tangent vector X along ∂M; moreover, we have $T \times B = -N$. The integrand above therefore becomes $\dot{\alpha}T - \dot{\beta}N$. But since this quantity has no real component, $\dot{\rho}$ must integrate to zero. □

For surfaces f with prescribed *tangents* along the boundary, we have no such constraint: the operator $\mathcal{D}$ fails to be self-adjoint and therefore has a continuous (rather than discrete) spectrum. As a result (as long as the tangents $\tilde{T}$ themselves can be realized) $\mathcal{M}(f)$ consists of the entire Hilbert space $\mathcal{H}$, i.e., we can achieve any change in mean curvature half-density whatsoever. In this sense, a disk with prescribed tangents is the natural analog of curve with open endpoints (Section 4.2). However, not every collection of tangents is valid: for instance, it is impossible to construct a compact surface whose boundary tangents point in a constant direction (at some point the curve must "turn around").

Chapter 5

Discretization

In this chapter we discretize the operators used in Equation 4.5. Throughout we use sans serif characters to denote a matrix approximating a smooth linear operator—e.g., L approximates L.

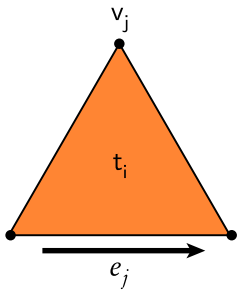
5.1 Quaternionic Matrices

Matrices $\mathsf{X} \in \mathbb{H}^{m \times n}$ with quaternion-valued entries provide a convenient representation for the operators used in our discretization. Although standard packages for numerical linear algebra do not support quaternionic matrices, they can easily be implemented by constructing a real matrix $\hat{\mathsf{X}} \in \mathbb{R}^{4m \times 4n}$ where each entry $q = a + bi + cj + dk$ is replaced by a 4×4 block

$$\begin{bmatrix} a & -b & -c & -d \\ b & a & -d & c \\ c & d & a & -b \\ d & -c & b & a \end{bmatrix}.$$

If performance or storage are an issue, one can alternatively provide a callback routine that applies the Hamilton product directly. Note that the real transpose $\hat{\mathsf{X}}^{\mathsf{T}}$ of such a matrix corresponds to the conjugate transpose X^{H} of the original quaternionic matrix.

5.2 Discrete Dirac Operator



To compute spin transformations of meshes, we need a discretization of the Dirac operator $\mathcal{D}$. Let $\mathcal{K} = \{V, E, F\}$ be a triangulation of M, and let $f_i \in \mathrm{Im}\mathbb{H}$, $\lambda_i \in \mathbb{H}$ denote the values of f and λ (respectively) at each vertex $v_i \in V$. Then the discrete Dirac operator $\mathsf{D} \in \mathbb{H}^{|F|\times|V|}$ is a sparse rectangular matrix whose nonzero entries are

$$\mathsf{D}_{ij} = -\frac{1}{2\mathcal{A}_i} e_j, \tag{5.1}$$

where $\mathcal{A}_i$ is the area of triangle t_i, j is the index of any vertex v_j of t_i, and e_j is the opposing edge vector (see above). By convention, edge vectors are oriented counter-clockwise around each triangle.

There are a number of ways to derive D, but the simplest is perhaps to observe that in the smooth setting $\mathcal{D}\lambda$ can be written as

$$\mathcal{D}\lambda = \frac{d(df\,\lambda)}{|df|^2}.$$

If we let f and λ be piecewise linear functions interpolating values at vertices, integrating the function $\mathcal{D}\lambda$ over a triangle t_l with vertices (i, j, k) gives us

$$\int_{t_l} \mathcal{D}\lambda |df|^2 = \int_{\partial t_l} df\,\lambda = \sum_{e_{ij}\in\partial t_l} \int_{e_{ij}} df\,\lambda = \sum_{e_{ij}\in\partial t_l} (f_j - f_i)\tfrac{\lambda_i+\lambda_j}{2}.$$

We can simplify this sum by making the substitution $e_k = f_j - f_i$ (and similarly for the other edges). Noting that $\mathcal{D}\lambda$ is constant in each face, we can then divide the integrated quantity by the total area to get

$$-\frac{1}{2\mathcal{A}_l}(e_i\lambda_i + e_j\lambda_j + e_k\lambda_k),$$

which is precisely the quantity expressed by our matrix D above.

5.3 Curvature Potential

To solve the time-independent Dirac equation (Equation 4.5) we also need to discretize the curvature potential ρ, which as an operator represents scalar multiplication by a real-valued function. We proceed as above by integrating over triangles—in particular, let ρ be a piecewise constant function with values $\rho_i \in \mathbb{R}$ in each face, and let λ be piecewise linear as before. We then have

$$\frac{1}{\mathcal{A}_i} \int_{t_i} \rho \lambda |df|^2 = \rho_i \left(\frac{1}{3} \sum_{v_j \in t_i} \lambda_j \right)$$

over any triangle t_i. For later derivations it will be convenient to express the corresponding discrete operator $\mathsf{R} \in \mathbb{H}^{|F| \times |V|}$ as

$$\mathsf{R} := \mathsf{P}\mathsf{B}$$

where $\mathsf{P} \in \mathbb{H}^{|F| \times |F|}$ is a diagonal matrix with entries $\mathsf{P}_{ii} = \rho_i$, and the averaging operator $\mathsf{B} \in \mathbb{H}^{|F| \times |V|}$ amounts to a triangle-vertex incidence matrix with nonzero entries $\mathsf{B}_{ij} = 1/3$ for each vertex v_j of triangle t_i. Finally, we let

$$\mathsf{A} := \mathsf{D} - \mathsf{R}$$

be the discretization of the operator $A := \mathcal{D} - \rho$ appearing in Equation 4.5.

5.4 Adjoint Matrices

For any matrix X, we use X^* to denote the adjoint with respect to the $\mathscr{L}^2$ inner product. More specifically, consider the diagonal mass matrices $\mathsf{M}_F \in \mathbb{H}^{|F| \times |F|}$ with diagonal entries equal to triangle areas $\mathcal{A}_i$, and $\mathsf{M}_V \in \mathbb{H}^{|V| \times |V|}$ with entries equal to one-third the area of triangles incident on each vertex. Then the adjoint of any matrix $\mathsf{X} \in \mathbb{H}^{|F| \times |V|}$ is given by $\mathsf{M}_V{}^{-1} \mathsf{X}^\mathsf{T} \mathsf{M}_F$. In all systems we consider, the term $\mathsf{M}_V{}^{-1}$ appears on both the left- and right-hand side—for convenience we therefore omit this term and adopt the notational convention

$$\mathsf{X}^* := \mathsf{X}^\mathsf{H} \mathsf{M}_F.$$

5.5 Dirac Equation

Consider the time-independent Dirac equation

$$(\mathcal{D} - \rho)\lambda = \gamma\lambda$$

introduced in Section 4.3. If we discretize this equation as simply $\mathsf{A}\lambda = \gamma\mathsf{B}\lambda$ the result is a *rectangular* eigenvalue problem, which is difficult to solve directly. One possibility is to make the system square by averaging the values $\mathsf{A}\lambda$ and $\mathsf{B}\lambda$ from faces back to vertices, but this local averaging introduces spurious modes in the null space that severely corrupt solutions.

We can instead obtain a square system in the following way. Consider that in the smooth setting any solution (γ, λ) to the problem $A\lambda = \gamma\lambda$ also yields a solution to the problem $A^2\lambda = \gamma^2\lambda$. However, the opposite is not true since positive and negative eigenspaces can "mix" when we square A. In other words, if (γ, λ^+) and $(-\gamma, \lambda^-)$ are solutions to the original problem then linear combinations of λ^+ and λ^- are eigenfunctions of A^2 but not of A. The solution is to use the *generalized* eigenvalue problem $A^2\lambda = \gamma A\lambda$, where the additional factor of A on the right-hand side now distinguishes between eigenvalues of different sign. This observation leads to the discretization

$$\mathsf{A}^*\mathsf{A}\lambda = \gamma\mathsf{B}^*\mathsf{A}\lambda, \tag{5.2}$$

which we solve for the smallest eigenvalue γ and its corresponding eigenvector $\lambda \in \mathbb{H}^{|V|}$. Solutions to this problem exhibit excellent agreement with smooth solutions (see Section 5.8).

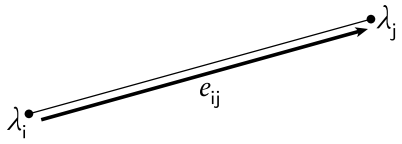
For surface editing tasks we are usually interested in the eigenvalue closest to zero and do not have to worry as much about mixing eigenspaces. In practice, then, we can often solve the problem $\mathsf{A}^*\mathsf{A}\lambda = \gamma\mathsf{M}_V\lambda$, or equivalently the *standard* eigenvalue problem

$$(\mathsf{AV})^*\mathsf{AV}\hat{\lambda} = \gamma\hat{\lambda}, \tag{5.3}$$

where $\mathsf{V} \coloneqq {\mathsf{M}_V}^{-1/2}$ and $\lambda = \mathsf{V}\hat{\lambda}$ recovers the final solution. In particular, we use this problem for the applications described in Chapter 6. Note that all matrices in Equation 5.2 and Equation 5.3 (including $\mathsf{B}^*\mathsf{D}$) are sparse and symmetric, and that operators appearing on the left-hand side are positive semi-definite. It is also easy to verify that these operators involve values only in the 1-ring of a given vertex. As a result, these systems can be efficiently solved using standard numerical libraries—see

Section 6.6 for further discussion.

5.6 Spin Transformations



After solving for the similarity transformation λ, we still need to determine the final vertex positions $\tilde{f}_i$ by solving $d\tilde{f} = \overline{\lambda} df \lambda$ (Equation 4.3). In the discrete setting, this equation says that each new edge vector $\tilde{e}_{ij} := \tilde{f}_j - \tilde{f}_i$ should equal the edge vector $e_{ij} := f_j - f_i$ from the original mesh scaled and rotated by λ. This transformation is discretized by integrating $\overline{\lambda} df \lambda$ along each edge, yielding

$$\tilde{e}_{ij} = \frac{1}{3}\overline{\lambda}_i e_{ij} \lambda_i + \frac{1}{6}\overline{\lambda}_i e_{ij} \lambda_j + \frac{1}{6}\overline{\lambda}_j e_{ij} \lambda_i + \frac{1}{3}\overline{\lambda}_j e_{ij} \lambda_j. \tag{5.4}$$

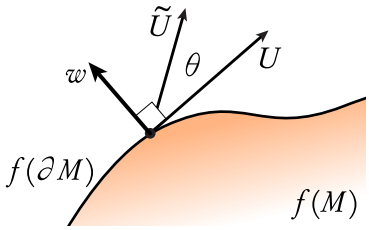
At this point we can solve the linear system $d\tilde{f} = \tilde{e}$ for vertex positions $\tilde{f}$. To do so we minimize the residual $r := |d\tilde{f} - \tilde{e}|^2$, which amounts to a standard Poisson problem

$$\Delta \tilde{f} = \nabla \cdot \tilde{e}. \tag{5.5}$$

Discretization of this problem yields (using notation from [DKT08]) the standard cotangent operator $\Delta_C := d_0^{\mathsf{H}} \star_1 d_0 \in \mathbb{H}^{|V| \times |V|}$ and divergence operator $\nabla \cdot = d_0^{\mathsf{H}} \star_1 \in \mathbb{H}^{|V| \times |E|}$ built as quaternionic matrices with purely real entries.

One might wonder why we need to solve this system in the least-squares sense—after all, in the smooth setting our integrability condition (Equation 4.4) guarantees that Equation 4.3 has an exact solution. As with linear discretizations of Cauchy-Riemann, however, this exactness does not carry over to the discrete setting. Nonetheless we can empirically verify that the magnitude of the residual r depends only on mesh resolution and vanishes under refinement (Section 5.8). More importantly, even at a coarse level we see substantial improvement over methods that do not consider integrability, which exhibit significant conformal distortion even under refinement (Section 5.9).

5.7 Boundary Conditions



Suppose that U and $\widetilde{U}$ are unit tangent vectors that specify initial and target directions at each boundary point, as discussed in Section 4.5.4. Let θ be the angle between U and $\widetilde{U}$ and let w be the unit vector orthogonal to both tangents. (In the case where $\widetilde{U} = \pm U$, let $\theta = 0$ and let w be any unit vector.) Then at each point along the boundary, the similarity transformation λ must have the form

$$\lambda = \left(\cos\tfrac{\theta}{2} - w \sin\tfrac{\theta}{2}\right)(a - b\,\widetilde{U})$$

for any $a, b \in \mathbb{R}$. The first term rotates from U to $\widetilde{U}$ and the second term describes subsequent scaling and rotation around $\widetilde{U}$.

In the discrete setting we store a pair of values $a, b \in \mathbb{R}$ at each boundary vertex (in lieu of a value $\lambda \in \mathbb{H}$) and build the block-diagonal matrix $\mathsf{C} \in \mathbb{R}^{4|\partial V| \times 2|\partial V|}$ where ∂V denotes the set of boundary vertices and each block has the form

$$\begin{bmatrix} 1 & 0 \\ 0 & \widetilde{U}^x \\ 0 & \widetilde{U}^y \\ 0 & \widetilde{U}^z \end{bmatrix},$$

where $\widetilde{U}^x, \widetilde{U}^y, \widetilde{U}^z$ are the components of $\widetilde{U}$. We also build the diagonal matrix $\mathsf{W} \in \mathbb{H}^{|V| \times |V|}$ where W_{ii} equals 1 for interior vertices and $\cos(\theta_i/2) + w_i \sin(\theta_i/2)$ for boundary vertices v_i. Finally, suppose that vertex indices are ordered such that interior vertices appear before boundary vertices. We can then build the matrix

$$\mathsf{U} \coloneqq \mathsf{W} \begin{bmatrix} \mathsf{I} & 0 \\ 0 & \mathsf{C} \end{bmatrix},$$

where I denotes the identity. This matrix maps both interior and boundary degrees of freedom to the appropriate values of λ. Hence, Equation 5.3 becomes

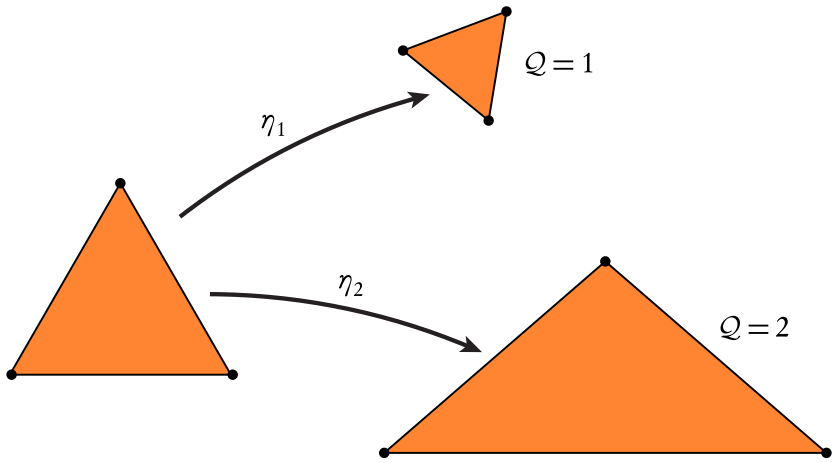
$$(\mathsf{AVU})^* \mathsf{AVU} \hat{\lambda} = \gamma \hat{\lambda}$$

and the final solution is recovered by evaluating $\lambda = \mathsf{V}\mathsf{U}\hat{\lambda}$.

5.8 Validation

In this section we examine the behavior of our discrete operator D relative to known properties of the smooth operator $\mathcal{D}$, and investigate numerical convergence of solutions to the time-independent Dirac equation (Equation 3.2).

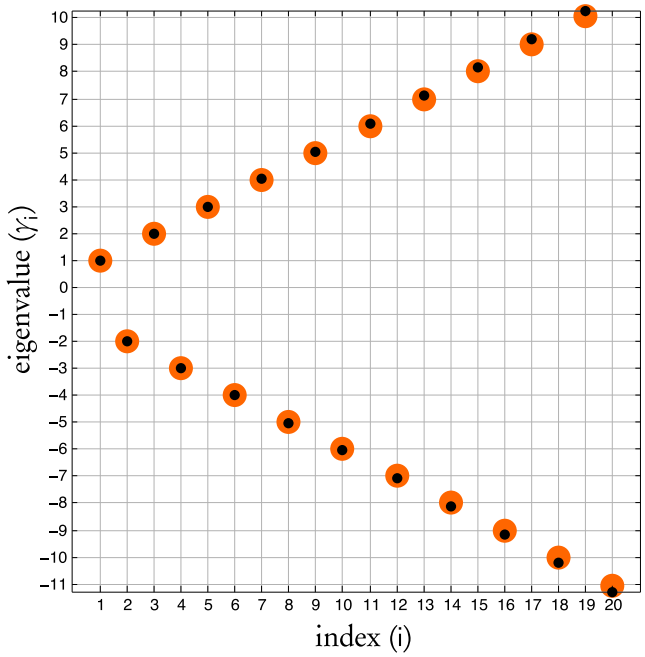
5.8.1 Quasi-Conformal Error



Throughout this chapter we measure the quality of any map $\eta : f(M) \to \tilde{f}(M)$ between immersed surfaces in terms of the associated *quasi-conformal error* $\mathcal{Q} : M \to \mathbb{R}$, defined as the ratio of largest to smallest singular value of $d\eta$. On a triangulated surface, quasi-conformal error is evaluated in each face, as described by Sander et al. [SSGH01]. An error of $\mathcal{Q} = \mathbf{1}$ is ideal, indicating that no shearing takes place (only uniform scaling and rotation). We use $\mathcal{Q}_{\text{mean}}$ and $\mathcal{Q}_{\text{max}}$ to denote the mean and maximum quasi-conformal error over the whole domain.

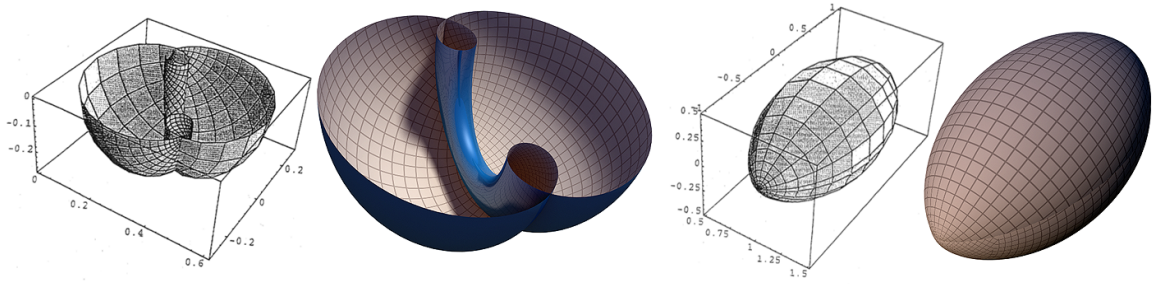
5.8.2 Eigenvalue Spectrum

On the unit sphere the eigenvalues of $\mathcal{D}$ are the integers $n \in \mathbb{Z}$, which appear with multiplicity $n + 1$ (note that $n = -1$ does not appear at all). Even on a coarse mesh of only 8k triangles, the spectrum of our discrete operator (black dots) matches the predicted spectrum (orange dots) exceptionally well:



Here we plot the first twenty distinct eigenvalues (out of 120 total) for the unit sphere; these eigenvalues appear with the correct multiplicity $n+1$.

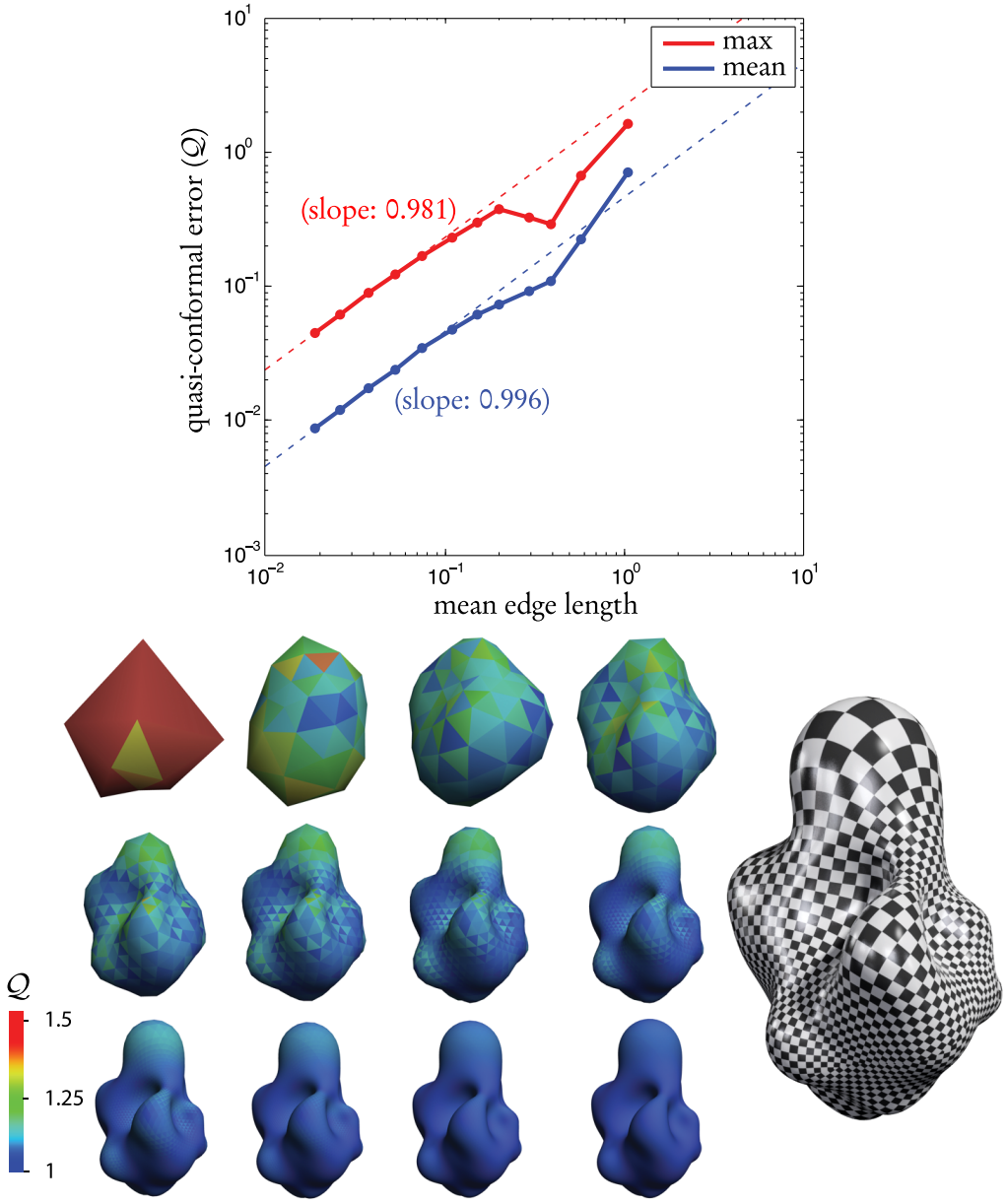
Eigenfunctions of $\mathcal{D}$ on the unit sphere are called the *spherical spinors* [Szm10], which can be used to compute fundamental solutions to the Dirac equation for a spin-1/2 particle such as an electron in a spherically symmetric potential (this relationship motivates the "spin equivalence" terminology). The corresponding spin transformations $\tilde{f}$ are called *Dirac spheres*, which can be used to validate our discretization since closed-form expressions are known and have previously been visualized for small values of n:



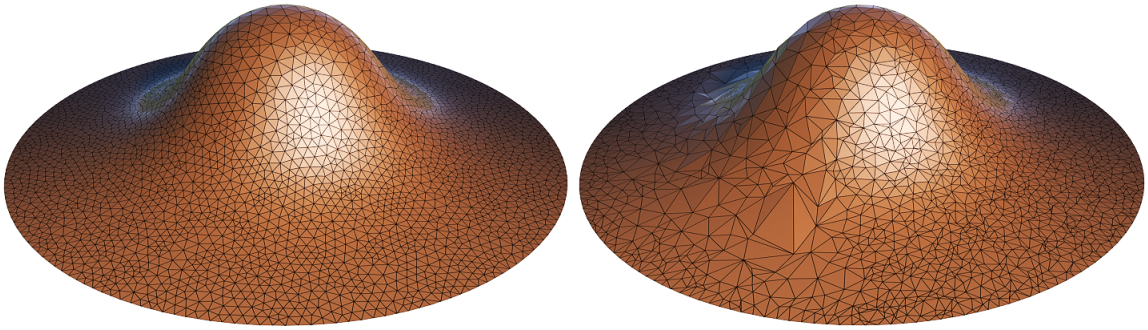
Boxed plots were produced by Richter [Ric97]; color plots are generated using our method as described in Chapter 5.

5.8.3 Convergence

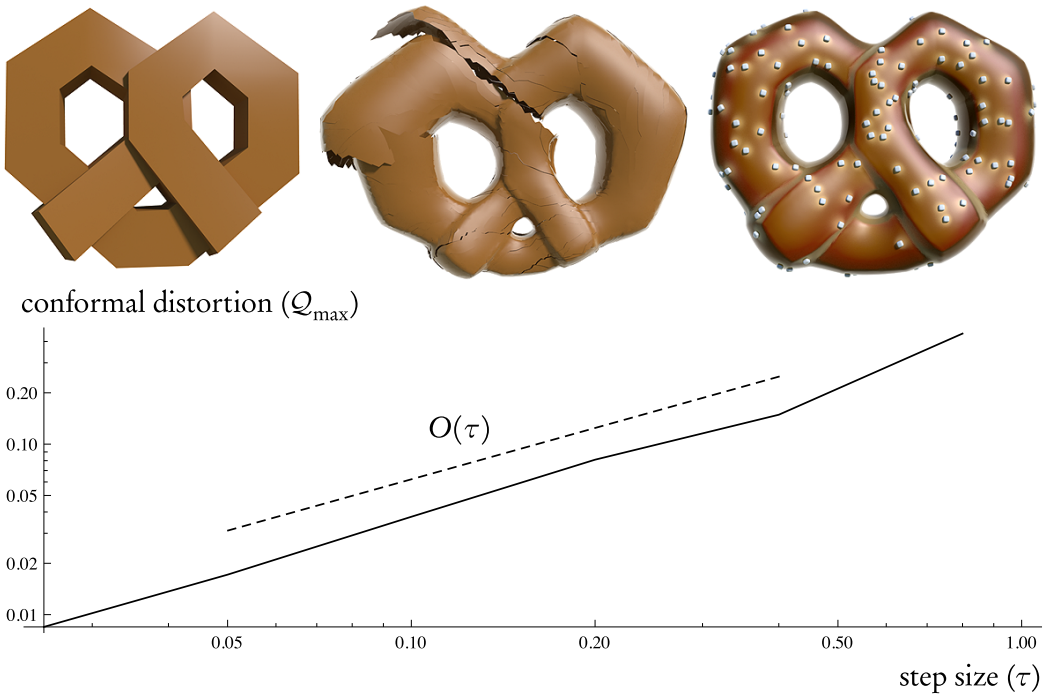
In the discrete setting, our integrability condition (Equation 4.4) will not be satisfied exactly due to spatial and temporal discretization error—the Poisson equation used to recover position Equation 5.5 effectively distributes this error over the domain. Even on coarse meshes, however, the resulting conformal distortion is slight, and converges linearly under temporal and spatial refinement. We examined the spatial convergence of our method by computing spin transformations of the unit sphere corresponding to an arbitrary but smooth curvature potential ρ, measuring quasi-conformal error as a function of mean edge length:



Colors indicate the spatial distribution of quasi-conformal error $\mathcal{Q}$. Note that $\mathcal{D}$ is a self-adjoint elliptic operator, which means that on a compact surface it has a discrete spectrum. Hence, asking for the *smallest* eigenvalue is a well-posed problem, and the observed linear convergence is precisely what we expect from a piecewise linear discretization. In this test we used a regular triangulation of the sphere obtained by applying regular 4-1 subdivision to a combinatorial icosahedron. However, further tests suggest that results do not depend critically on element quality. Below we let ρ be a smooth bump function on a flat circular disk, and obtain similar results for a uniform (left) and nonuniform (right) sampling of the domain:



In the case of surfaces with nontrivial topology, we also experience discretization error due to linearization of the infinitesimal constraints described in Section 4.5.2. In other words, a finite motion of our surface will "step off" the conformal shape space $\mathcal{M}$. We again observe, however, that any induced conformal distortion vanishes under temporal refinement:



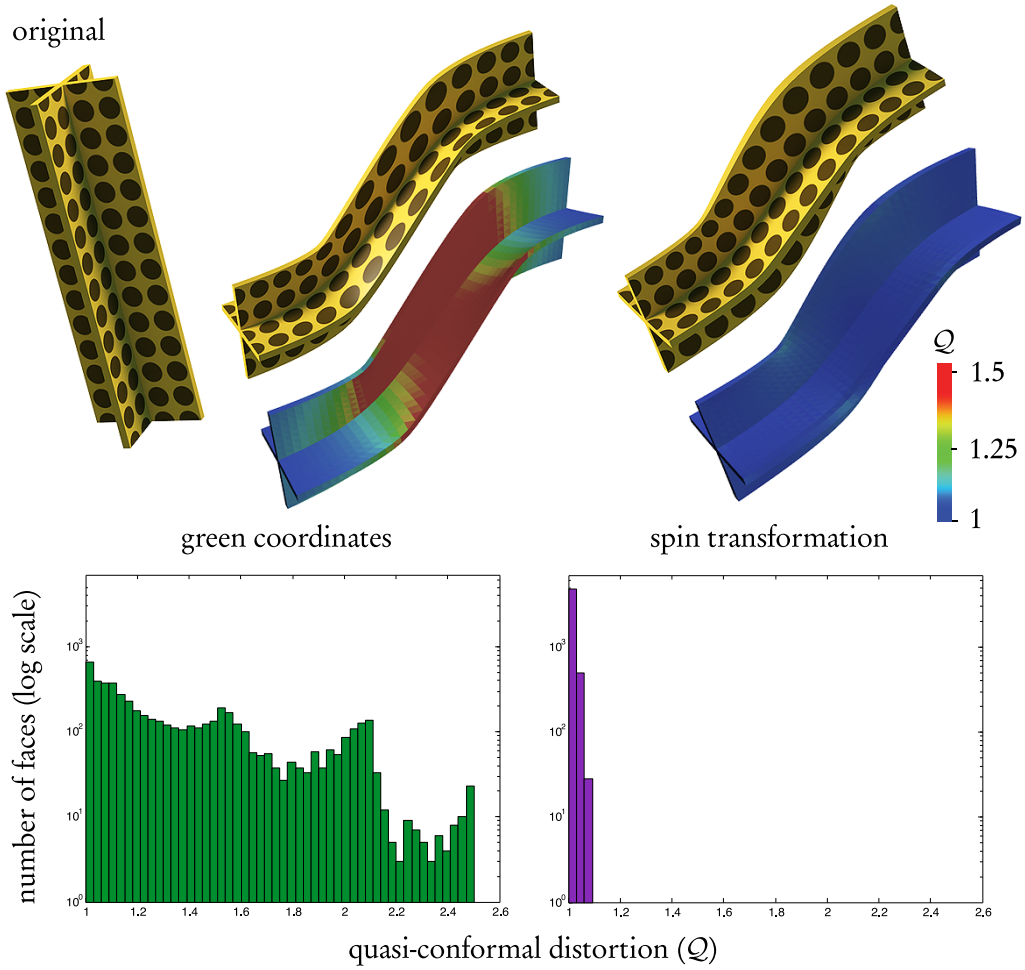
In this example we run Willmore flow (Section 6.4) on a surface of genus $G = 3$. The initial surface is depicted in the top left. In the top center, we visualize the failure of the transformed differential $\beta = \overline{\lambda} df \lambda$ to close by locally integrating the surface (rather than solving the Poisson problem described in Equation 5.5)—here we take an excessively large time step to exacerbate the effect of non-integrability. In the top right, we show the surface recovered using our method for a more reasonable time step (salt cubes are added for artistic effect). On the bottom we show a log-log plot of conformal distortion versus time step size τ; the dashed line corresponds to linear convergence. For all tests performed in this section we used the standard eigenvalue problem (Equation 5.3).

5.9 Comparison

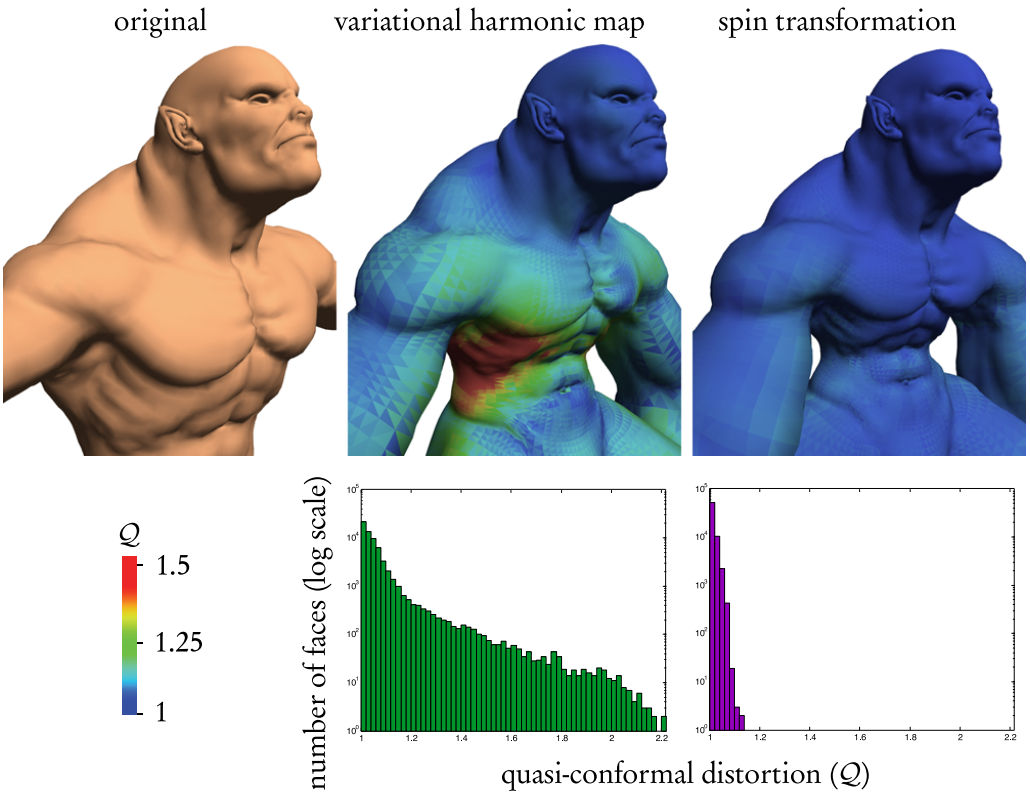
In this section we provide a comparison with methods described in Section 1.2. In each example we apply the procedure described in Section 6.3 to find a spin transformation that approximates the output of an existing approach. In each case, the quasi-conformal distortion $\mathcal{Q}$ incurred by the spin transformation is substantially smaller. Moreover, in contrast to previous methods, this error is expected to vanish under spatial refinement (see Section 5.8.3). Histograms plot the distribution of quasi-conformal distortion over mesh faces.

First, following Lipman et al. [LLCO08, Fig. 13], we examine a cross-shaped bar deformed using

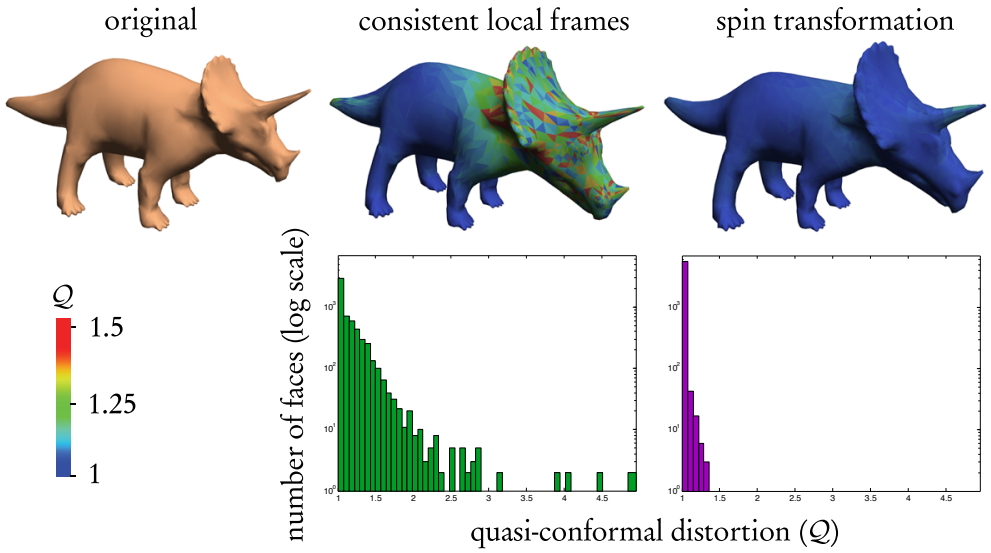
green coordinates:



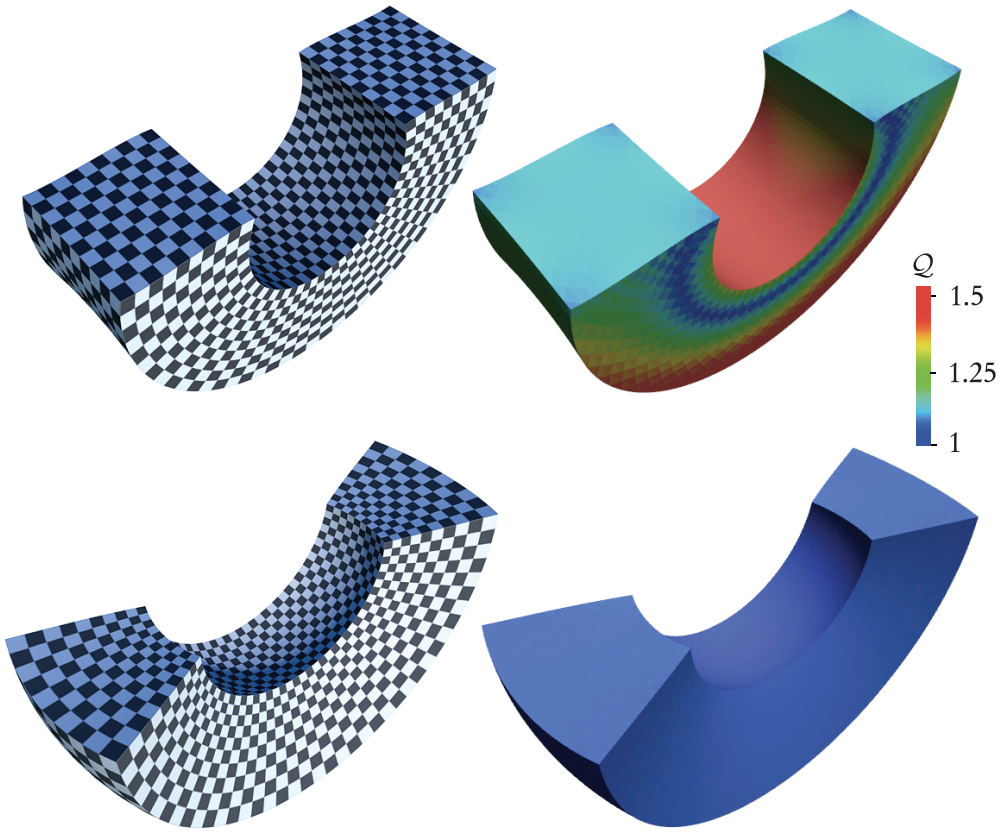
Notice that the spin transformation preserves circles perfectly. (See [LLCO08] for additional comparisons.) Next, we examine a deformation produced via a variational harmonic map [BCWG09]:



We next examine a deformation produced via the method of consistent local frames [PDK07]:



Finally, we apply a simple closed-form rotation λ to the differential df of a rectangular prism in order to obtain a bent block. In the top row we reconstruct the surface directly from the transformed differential $\beta = \overline{\lambda} df \lambda$; in the bottom row we find an integrable version of λ by first solving Equation 4.5:



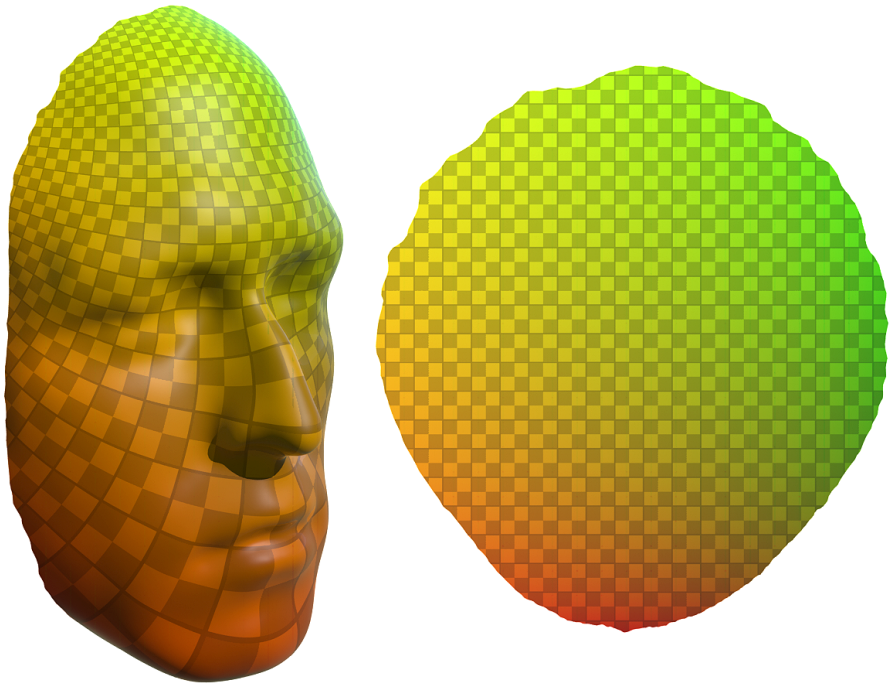
This example emphasizes the fact that global integrability is indeed crucial when computing conformal deformations. In other words, even though λ describes a perfect similarity transformation at each point, this local condition is not enough to preserve conformal structure globally.

Chapter 6

Applications

This chapter demonstrates how a variety of basic digital geometry processing tasks can be expressed in terms of mean curvature half-density.

6.1 Conformal Parameterization



The most traditional application of conformal maps in digital geometry processing is *conformal parameterization*: given an immersed disk $f : M \to \mathbb{R}^3$, find a map $z : M \to \mathbb{C}$ that induces the same conformal structure as f. For triangulated surfaces one typically tries to minimize conformal distortion (in the $\mathscr{L}^2$ sense, say) since in general there will be no piecewise linear map that perfectly preserves conformal structure. There are already a wide variety of excellent methods for solving this

problem (or variants of it), and at present we do not claim to improve on these methods from a numerical perspective. For completeness, however, we will demonstrate how this standard task can be easily implemented using our discrete Dirac operator.

Proposition 6.1.1. *Let $\psi = a - bN$ for real functions a, b on M. If $(\mathcal{D}\psi)^\top = 0$, then $z := a + bi$ is holomorphic.*

Proof. Applying Equation 3.3 we have

$$\mathcal{D}\psi = \begin{bmatrix} 0 & -\operatorname{curl} & 0 \\ \mathcal{J}\operatorname{grad} & -S & \operatorname{grad} \\ 0 & -\operatorname{div} & 2H \end{bmatrix} \begin{bmatrix} a \\ 0 \\ -b \end{bmatrix} = \begin{bmatrix} 0 \\ \mathcal{J}\operatorname{grad} a - \operatorname{grad} b \\ -2Hb \end{bmatrix}$$

which means the tangential part is just $(\mathcal{D}\psi)^\top = \mathcal{J}\operatorname{grad} a - \operatorname{grad} b$ and we recover the usual Cauchy-Riemann equations

$$\operatorname{grad} b = \mathcal{J}\operatorname{grad} a.$$

□

We can implement this system numerically as follows. Let $\mathsf{D} \in \mathbb{H}^{|F|\times|V|}$ be our usual discrete Dirac operator; let $\mathsf{E} \in \mathbb{R}^{4|V|\times 2|V|}$ be a block-diagonal matrix with blocks

$$\begin{bmatrix} 1 & 0 \\ 0 & N_{\mathsf{i}}^x \\ 0 & N_{\mathsf{i}}^y \\ 0 & N_{\mathsf{i}}^z \end{bmatrix}$$

and let $\mathsf{F} \in \mathbb{R}^{2|F|\times 4|F|}$ be block-diagonal with blocks

$$\begin{bmatrix} 0 & X_{\mathsf{k}}^x & X_{\mathsf{k}}^y & X_{\mathsf{k}}^z \\ 0 & Y_{\mathsf{k}}^x & Y_{\mathsf{k}}^y & Y_{\mathsf{k}}^z \end{bmatrix}.$$

The vector $N_{\mathsf{i}} \in \mathbb{R}^3$ is the normal of vertex i, and the vectors $X_{\mathsf{k}}, Y_{\mathsf{k}} \in \mathbb{R}^3$ are any orthonormal basis for face k. The matrix E maps real degrees of freedom a, b to quaternionic values $a - bN$ at each vertex; the matrix F extracts the tangential component of $(\mathcal{D}\psi)^\top$ from each face. The composite matrix $\mathsf{C} = \mathsf{FDE}$ can therefore be used to solve a least-squares problem for values $z = a + bi$ at each vertex,

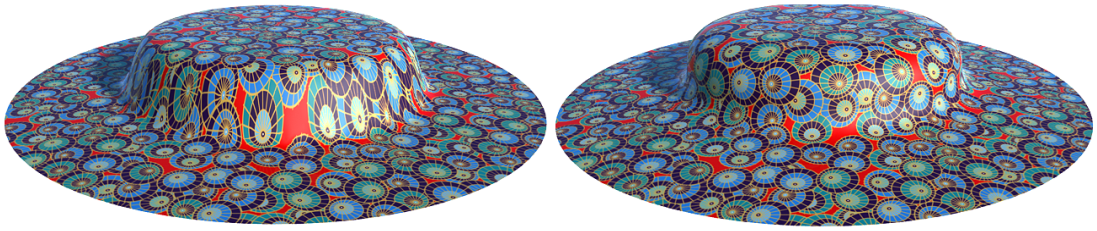
subject to appropriate boundary conditions or norm constraints. We adopt the approach of Mullen et al. [MTAD08], replacing their matrix L_C with our matrix C (otherwise, the numerical treatment is identical). The image above demonstrates a surface parameterized using this method. Note that our discretization *cannot* be the same as the standard least-squares approach [DMA02, LPRM02] since it depends on the choice of vertex normals N_i, though in this preliminary investigation we do not perform a detailed numerical comparison.

6.2 Filtering Curvature

The most straightforward way to specify a general deformation is to "paint" a function ρ on a surface, altering its curvature. Filtering this function achieves a variety of effects while preserving a conformal map to the original disk. Above we apply a low-pass filter (top right), high-pass filter (bottom left), and unsharp mask (bottom right). An important feature of our approach is that, unlike many methods for filtering surfaces, we do not have to derive special filters for triangulated surfaces—here we simply apply standard image filters to a regular grayscale image that defines the curvature

potential ρ.

Generally speaking, if h is a bump function specifying a displacement in the normal direction, setting $\rho = \Delta h$ produces a similar but conformal bump. This observation is based on the fact that for a flat surface undergoing a normal deformation $\tilde{f}(t) = f + t h N$, the change in $\widetilde{H} d\tilde{f}$ at time $t = 0$ equals Δh. Even for modest displacements, simple normal offsets can severely distort texture, whereas the corresponding spin transformation prevents distortion:



normal displacement spin transformation

Note that large, high-frequency spikes in ρ will not be well-represented unless the domain is sufficiently well-sampled. In other words, it is typically impossible to keep quasi-conformal distortion low when making large deformations to a small number of triangles (consider adding a large amount of curvature to a single vertex 1-ring).

6.3 Arbitrary Deformations

More generally, we can take a mesh that has been deformed arbitrarily and find a nearby spin transformation. We first compute "best-fit" similarity transformations in each face. In particular, let $\mathsf{E}_1, \mathsf{E}_2$ be rotation matrices bringing the source and target face into a common plane, and let F be the (orientation-preserving) map between the resulting planar triangles. If $\mathsf{F} = \mathsf{UY}$ is the polar decomposition of F (where U is the orthogonal part), then the best-fit similarity transformation is

$$\mathsf{S} = \sqrt{\det(\mathsf{Y})}\mathsf{E}_2^{-1}\mathsf{U}\mathsf{E}_1,$$

i.e., the geometric average of principal stretches times the rotational component of the map between triangles. These transformations are expressed as quaternions and averaged from faces to vertices to get a value μ_i at each vertex.

In general μ will not satisfy our integrability condition (Equation 4.4), but we can still find a change in curvature that closely approximates the target surface by minimizing $\|(\mathsf{D} - \mathsf{R})\mu\|^2$ with

respect to the values ρ_i. The optimal values can be computed locally in each face (i.e., no global solve is necessary), and are given explicitly by

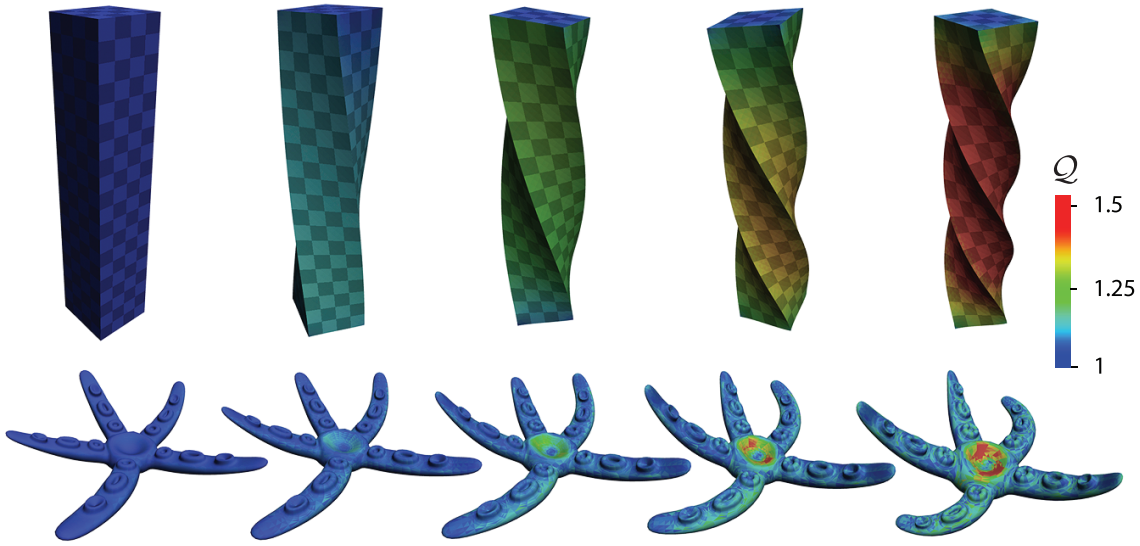
$$\rho_i = \frac{\mathrm{Re}((\overline{\mathsf{B}\mu})_i(\mathsf{D}\mu)_i)}{|(\mathsf{B}\mu)_i|^2}.$$

We then solve for λ and the new surface $\tilde{f}$ as usual.

Using this procedure, we can edit a mesh using a general-purpose 3D modeling tool and project the result onto a nearby spin transformation, preserving the appearance of associated textures:

The original surface is shown on the left; two different spin transformations are shown in the center and on the right where we achieve quasi-conformal distortion of $\mathcal{Q}_{\text{mean}} = 1.015$. As illustrated on the right, we can of course explicitly modify scale while preserving conformal structure—here we ask for a much larger head. We found that the generalized eigenvalue problem (Equation 5.2) proved more robust for this application.

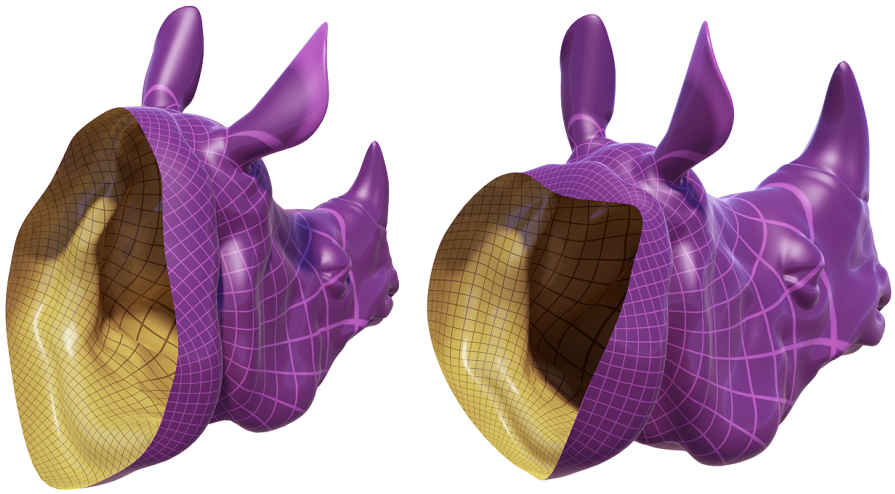
Deformations with a large amount of *shear* cannot be well-approximated by *any* conformal map, even in the smooth setting. Here we can make a tradeoff between approximation quality and conformality by interpolating between λ (far left) and μ (far right):



The top row shows an extreme example: the conformal "deformation" closest to a twisted beam (top right) is simply the original beam (top left). Similarly, local twisting of "suckers" on the starfish prohibit the arms from bending conformally.

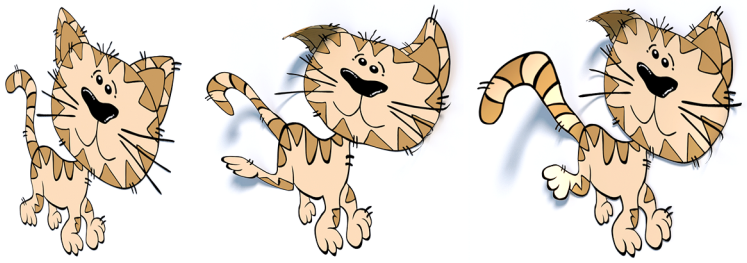
6.3.1 Boundary Prescription

We can also edit surfaces by manipulating boundary data. For instance, by setting $\rho = 0$ on the interior of the domain and prescribing new tangents $\widetilde{T}$ along the boundary (as described in Section 5.7) we can modify the boundary shape while preserving the surface curvature ($\mathcal{Q}_{\text{mean}} = 1.016$):



Notice that some features change scale, yet angles are almost perfectly preserved as illustrated by orthogonal grid lines.

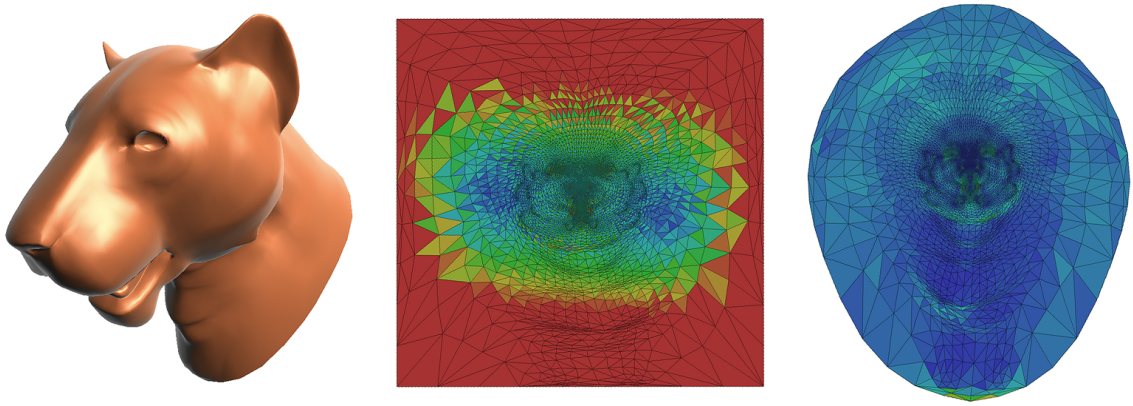
We can also prescribe a new boundary *and* new curvature. Here we lift a planar cartoon (left) into three dimensions (middle) and recover the closest conformally equivalent shape (right):



Note that the initially squashed leg, tail, and head, recover their initial shape ($\mathcal{Q}_{\text{mean}} = 1.011$). In these applications we discretize T and $\tilde{T}$ as simply the (unit) outgoing edge vector at each boundary vertex of the source and target mesh, respectively.

6.3.2 Position Constraints

So far, our treatment of boundary conditions allows one to prescribe the direction of vectors along the boundary, or equivalently, the boundary curvature. In many applications, however, it is desirable to enforce Dirichlet constraints on position, i.e., we want to prescribe the exact coordinates of $\tilde{f}$ at certain points. The reason we have not examined position constraints up to this point is that they are not, in general, conformal. For instance, if one tries to conformally map a surface to the plane while explicitly prescribing the boundary curve, the conformal distortion will be much larger that if one allows the boundary to evolve freely:

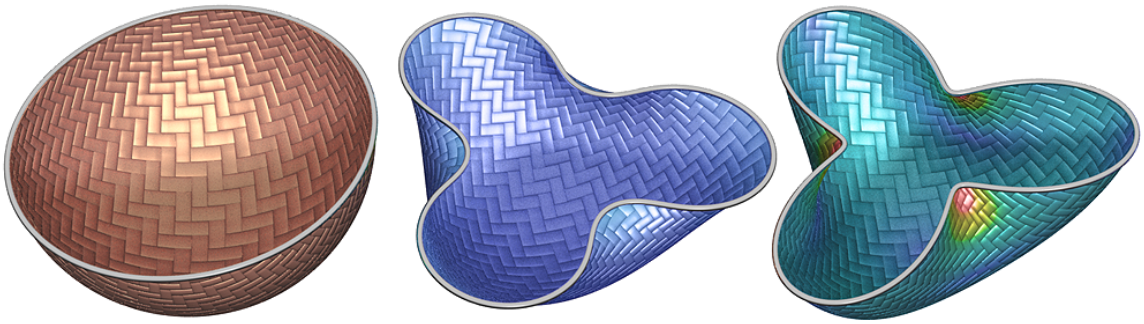


initial surface prescribed boundary free boundary

This behavior is unavoidable, independent of the particular numerical method used to compute the map (here, for example, we apply standard linear conformal parameterization [DMA02, LPRM02, MTAD08]). However, we can seek a deformation that minimizes conformal distortion in the $\mathscr{L}^2$ sense, as described below (one can also minimize distortion in the L^∞ sense [WMZ12], though it is not immediately obvious how to apply this technique for arbitrary conformal deformations in $\mathbb{R}^3$).

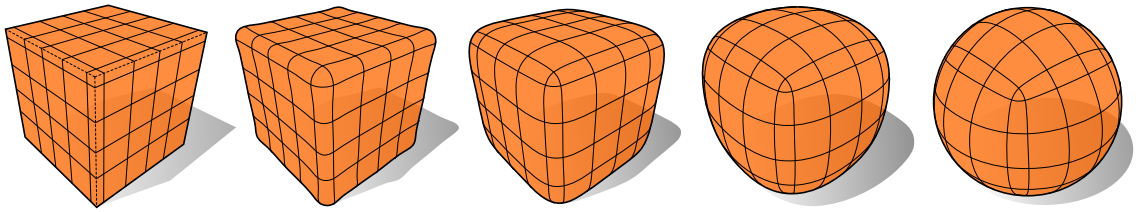
To force the boundary curve $f|_{\partial M}$ to match the boundary of a target surface $\hat{f}$, we find values λ_∂ along the boundary that (i) map initial tangents T to target tangents $\hat{T}$, and (ii) make the initial normals N parallel with target normals $\hat{N}$, precisely as described in Section 5.7. Discretely, we approximate T at vertices as the average of incident edge vectors; N is any reasonable vertex normal orthogonal to T. Minimizing the residual $||(\mathcal{D}-\rho)\lambda||^2$ subject to $\lambda|_{\partial M} = \lambda_\partial$ yields a standard linear least-squares problem—in particular, we build the same matrix X used for the eigenvalue problem (Section 6.6.1) with boundary columns moved to the right-hand side. We then recover the transformed surface by minimizing $||d\tilde{f} - \overline{\lambda} T \lambda||^2$ subject to $\tilde{f}|_{\partial M} = \hat{f}_{\partial M}$, amounting to a Poisson equation with fixed boundary values; here we use the usual cotangent operator (Section 6.6.2).

In the example below, the boundary of a hemisphere (left) is modified using constraints on either tangents (center) or positions (right):



In the latter case a perfectly conformal deformation does not exist—instead, conformal distortion is distributed over the domain. Overall we observe reasonably good preservation of conformal structure despite distortion near the boundary. Finally, although we consider only the domain boundary ∂M in this example, one could in principle enforce similar constraints for curves in the domain interior.

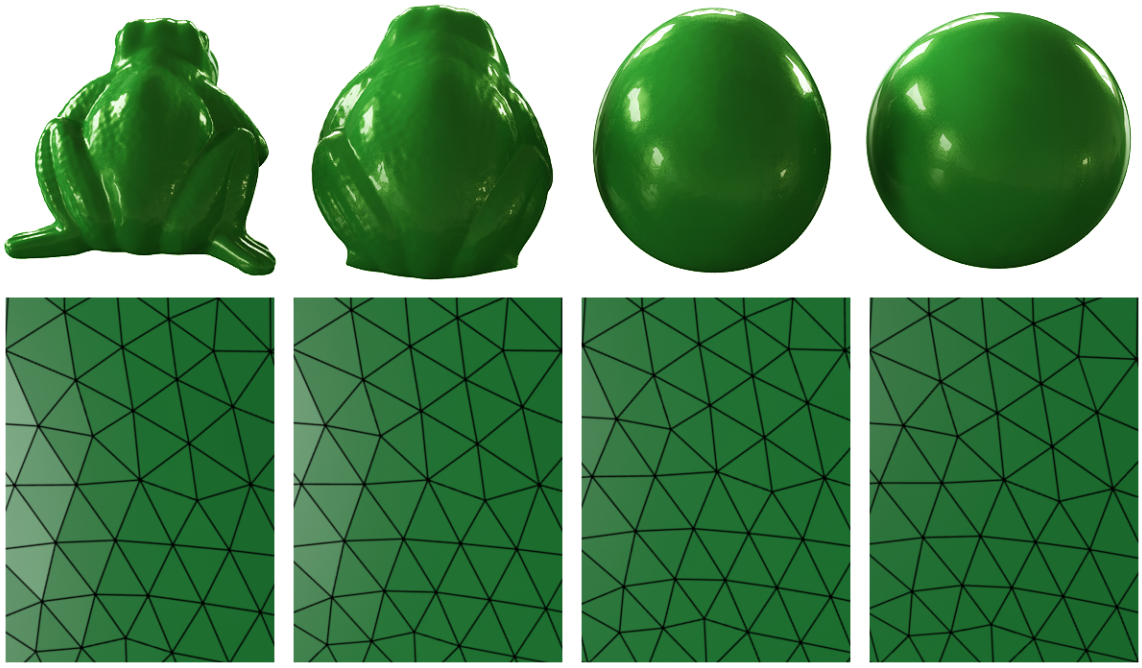
6.4 Willmore Flow



This section presents a formulation of Willmore flow for triangulated surfaces that permits extraordinarily large time steps and naturally preserves the quality of the input mesh. The main insight is that numerical algorithms for Willmore flow become remarkably stable when expressed in terms of mean curvature half-density (Section 2.4)—the practical outcome is a highly efficient algorithm that naturally preserves texture and does not require remeshing during the flow. We also present a new algorithm for length-preserving flow on planar curves, which provides a valuable analogy for the surface case.

At the most basic level, a curvature flow produces successively smoother approximations of a given piece of geometry (e.g., a curve or surface), typically by reducing a *fairing energy*. Such flows have far-ranging applications in fair surface design [CG91, WW94, SK01, YB02], inpainting [CDD+04], denoising [Tau95, DMSB99, BS05], and biological modeling [Hel73, Can70]; they are also the central object of study in mathematical problems such as the Willmore conjecture [PS87].

Numerical algorithms for curvature flow suffer from two principal difficulties: (I) a severe time step restriction, which often yields unacceptably slow evolution and (II) degeneration of mesh elements, which necessitates frequent remeshing or other corrective devices. We circumvent these issues by (I) using a curvature-based representation of geometry (Section 4.4), and (II) working only with *conformal* transformations (Section 2.3), which naturally preserve the aspect ratio of triangles throughout the flow. The resulting algorithm stably integrates time steps orders of magnitude larger than previous methods, resulting in substantially faster real-world performance (Section 6.4.6). For instance, in the example below a detailed frog flows to a round sphere in only three large, explicit integration steps—meanwhile, the quality of the triangulation is almost perfectly preserved:



6.4.1 Curvature Flow

There are a wide variety of surface fairing procedures for triangle meshes, which at first glance appear to be quite disparate. Yet most fairing algorithms can be viewed as numerical minimization of either the *membrane energy* E_A, or the *Willmore energy* E_W. To simplify discussion, we will assume that the surface M has no boundary—in this case, membrane energy is just the surface area

$$E_A(f) := \int_M \sigma,$$

and Willmore energy is the squared $\mathscr{L}^2$ norm of mean curvature

$$E_W(f) := \int_M H^2 \sigma.$$

Minimizing E_A via gradient descent leads to the popular *mean curvature flow* $\dot{f} = -HN$, where the surface evolves in the normal direction with speed proportional to curvature. Recalling that $\Delta f = 2HN$ (Section 2.2), we can also write mean curvature flow as

$$\dot{f} = -\tfrac{1}{2}\Delta f.$$

Similarly, we can write Willmore energy as

$$E_W(f) = \tfrac{1}{4}\langle\langle \Delta f, \Delta f \rangle\rangle = \tfrac{1}{4}\langle\langle \Delta^2 f, f \rangle\rangle.$$

If we ignore the dependence of Δ on f when taking the gradient of E_W, we get the *bi-Laplacian flow*

$$\dot{f} = -\tfrac{1}{2}\Delta^2 f,$$

which approximates the fully nonlinear *Willmore flow*

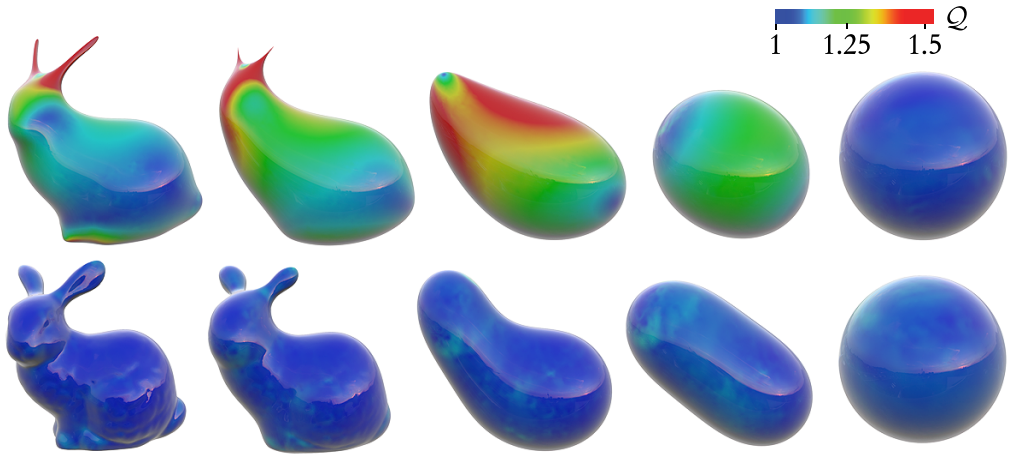
$$\dot{f} = -\nabla E_W(f).$$

A bi-Laplacian term also appears in *surface diffusion flow* [SK01]. Importantly, all these flows are *nonlinear* PDEs since the operator Δ is itself a function of the immersion f.

From this perspective, many algorithms for surface fairing arise from different choices of spatial and temporal discretization: Brakke's *Surface Evolver* [Bra92] minimizes membrane energy via explicit gradient descent, corresponding to mean curvature flow via the forward Euler method; Taubin's $\lambda|\mu$ algorithm [Tau95] amounts to bi-Laplacian flow via forward Euler when $\lambda = -\mu$; and the *implicit fairing* method of Desbrun et al. [DMSB99] corresponds to mean curvature flow via backward Euler. More recently, Bobenko and Schröder [BS05] and Wardetzky et al. [WBH+07] investigate discrete Willmore flow, using a semi-implicit quasi-Newton scheme to cope with nonlinearity. Common to all these methods are time step restrictions based on the smallest edge length h—at a practical level performance degrades rapidly as resolution increases or elements degenerate. *Explicit* methods typically exhibit restrictions of $O(h^2)$ and $O(h^4)$ for mean curvature and Willmore flow, respectively (see [OR09] for a proposal to ameliorate this restriction). Implicit integrators such as backward Euler improve the situation, but do not guarantee unconditional stability since flows are inherently nonlinear, again due to the dependence of Δ on f (cf. [DMSB99]). In light of this situation, it is rather remarkable to find a change of variables that evades this restriction.

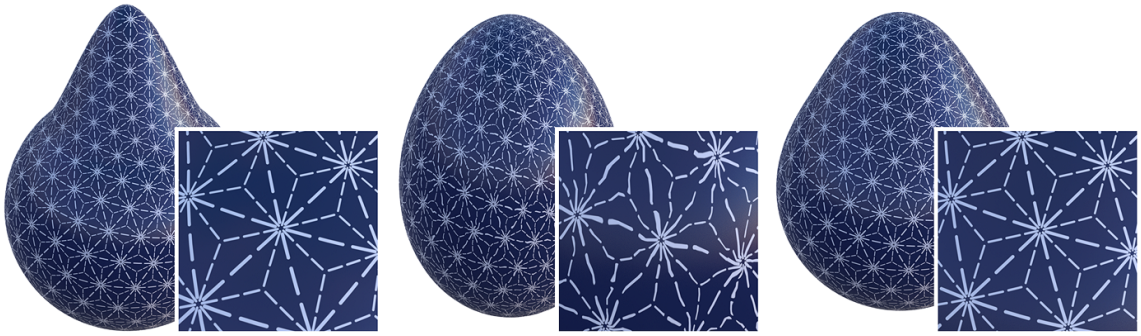
One can also compare geometric qualities of these methods. For example, mean curvature flow can develop sharp singularities (even in the continuous setting [CM09]) which undermine the fairing process. Kazhdan et al. propose a modification that helps avoid degeneracy but can still produce sharp features; in contrast, Willmore or bi-Laplacian flow tend to produce rounder, more aesthetic shapes. In this example, for instance, we compare Kazhdan's modified mean curvature flow (top) to

our conformal Willmore flow (bottom):



Fourth-order flows like Willmore also permit tangent constraints at the boundary, valuable for applications like geometric modeling [CG91, WW94].

The example above also illustrates that existing flows produce unwanted conformal distortion ($\mathcal{Q}$). For example, notice that the flow of Kazhdan et al. yields a conformal map to the sphere, yet exhibits conformal distortion at intermediate steps. This type of distortion not only degrades texture and mesh elements, but can also exacerbate numerical stability. Simple corrective devices do not quite work as desired. For instance, Laplacian-based tangential smoothing helps avoid degenerate elements, at the cost of distorting texture; it also suffers from the same stability issues as curvature flow itself [YB02]. In the example below we apply curvature flow to a bump representative of small surface detail (left):



Standard fairing (center) distorts texture, even when tangential smoothing is applied (here we apply the method of Yoshizawa & Belyaev [2002] to the flow of Desbrun et al. [1999]). Conformal fairing (right) nicely preserves texture while producing a pleasing geometric shape. Similarly, adaptive remeshing helps maintain element quality [dGV08], but neither prevents texture distortion nor improves asymptotic stability.

Another tempting idea is to project an existing flow onto the nearest angle-preserving deformation, yet in the discrete case this approach is far too rigid since neighboring triangles are forced to have identical scale. In the following example, for example, fairing an initial mesh (left) via standard methods such as implicit mean curvature flow (center) produces significant conformal distortion (qc):

Projecting onto an angle-preserving deformation (right) does not help, since in the discrete case such deformations are rigid and one simply recovers the initial surface. Here we apply the method of Bouaziz et al. [BDS+12] which very nearly restores the original surface—thereby reverting any fairing that may have occurred.

Conformal flows of the *metric* (e.g., Yamabe) have proven valuable for applications like surface parameterization—see [GZLY11] for further discussion. However, these methods work only with *intrinsic* (Gaussian) curvature which is insufficient to determine the geometry of an immersed surface. To date such flows have not been used for surface fairing. To our knowledge, ours is the first method for *extrinsic* conformal flow, making it naturally suited to surfaces embedded in $\mathbb{R}^3$.

Finally, one should be careful to note that our goal is *not* to develop integrators for the standard PDEs describing curvature flow. Classical flows are inherently non-conformal, which means that even an excellent numerical approximation will not provide the kinds of practical benefits we seek. Instead, our goal is to develop new PDEs for curvature flow that provide equal utility from an application perspective, but are far easier to integrate at the numerical level. (As a result, one should not attempt to use a pointwise comparison of our results with classical solutions as a figure of merit—doing so would most certainly be an "apples to oranges" comparison!)

6.4.2 Curvature Flow in Curvature Space

Our main point of departure from existing algorithms is that we manipulate curvature directly—for now, let u denote a generic curvature variable equal to either κ for curves or μ for surfaces. Fairness is measured via the quadratic energy

$$E(u) := ||u||^2,$$

leading to the simple gradient flow

$$\dot{u} = -2u. \tag{6.1}$$

However, we must be careful that curvature remains integrable as it evolves. As discussed in Chapter 4, this seemingly difficult condition can be expressed as a set of *linear* constraints $\langle\langle \dot{u}, c_i \rangle\rangle = 0$ for some collection of easily computed constraint functions c_i. In practice, we integrate Equation 6.1 using the forward Euler scheme

$$u^{k+1} = u^k - 2\tau u^k \tag{6.2}$$

for some fixed time step $\tau > 0$; constraints are enforced by building an orthonormal basis $\{\hat{c}_i\}$ via the Gram-Schmidt process and augmenting the initial flow direction $v := -2u$ via $v \leftarrow v - \sum_i \langle\langle v, \hat{c}_i \rangle\rangle \hat{c}_i$. To recover the final geometry we integrate curvature to get tangents, then integrate tangents to get positions, as discussed in Sections 6.4.3 and 6.4.4. The overall procedure is outlined in the following table:

Step	Description	Curve	Surface
I.	Evaluate curvature.	$\kappa \leftarrow \frac{1}{2}\langle N, \Delta f \rangle$	$H \leftarrow \frac{1}{2}\langle N, \Delta f \rangle$
II.	Pick a desired flow direction.	$\dot{\kappa} \leftarrow -\nabla E_C(\kappa)$	$\dot{\rho} \leftarrow -\nabla E_W(\rho)$
III.	Build a constraint basis.	Orthogonalize $\{1, f^x, f^y\}$	Orthogonalize $\{1, N^{x,y,z}, Z_i^{x,y,z}\}$
IV.	Project flow onto constraints.	$\dot{\kappa} \leftarrow \dot{\kappa} - \sum_i \langle\langle \dot{\kappa}, \hat{c}_i \rangle\rangle \hat{c}_i$	$\dot{\rho} \leftarrow \dot{\rho} - \sum_i \langle\langle \dot{\rho}, \hat{c}_i \rangle\rangle \hat{c}_i$
V.	Take an explicit Euler step.	$\kappa \leftarrow \kappa + \tau\dot{\kappa}$	$\rho \leftarrow \rho + \tau\dot{\rho}$
VI.	Recover tangents.	$\widetilde{T} \leftarrow$ Integrate κ	Solve $(\mathcal{D} - \rho)\lambda = \gamma\lambda$, $\widetilde{T} \leftarrow \overline{\lambda} T \lambda$
VII.	Recover positions.	Solve $\Delta \tilde{f} = \nabla \cdot \widetilde{T}$	Solve $\Delta \tilde{f} = \nabla \cdot \widetilde{T}$

Our formulation has a number of valuable consequences. First, since only curvatures are prescribed, we are free to reconstruct positions that preserve lengths or angles—in a sense we are free to take advantage of *reparameterization symmetry* (Chapter 4) to obtain more desirable numerical properties. Unlike constraint- or penalty-based methods, these quantities are preserved *by construction*.

Second, this scheme leads to greatly improved stability, chiefly because the flow we want to integrate (Equation 6.1) involves no spatial derivatives. Therefore, our one and only stability criterion is

that $|1-2\tau| < 1$, or equivalently, $\tau < 1$. The addition of constraints only *improves* stability, since projection onto the constraint set *reduces* the norm of the speed function. Experiments agree perfectly with this analysis—setting τ just above 1 yields an unstable flow; setting it just below 1 produces a stable flow, *independent of mesh quality or resolution*:

Here we apply our algorithm for Willmore flow to a highly irregular mesh (left) with minimum edge length less than 0.2% of the mesh diameter. The top row depicts the first four steps of an stable flow with $\tau = 1-\epsilon$; the bottom row depicts an unstable flow with $\tau = 1+\epsilon$, where $\epsilon = 3 \times 10^{-1}$. As predicted, the flow remains stable when $t < 1$, despite the presence of near-degenerate elements. One can also observe that curvature in the unstable flow exhibits oscillatory behavior characteristic of the forward Euler method.

6.4.2.1 Filtered Flows

The flow $\dot{u} = -2u$ gives an appearance different from traditional smoothing, since we take the gradient of E with respect to a nonstandard metric (namely, the $\mathscr{L}^2$ norm on curvature rather than position). In particular, large features shrink at the same rate as small bumps:

Proposition 6.4.1. *Consider an immersed surface $f : M \to \mathbb{R}^3$, which we can express locally as a time-varying height function $h(t)$ over each tangent plane. Integrating the flow $\dot{\rho} = -2H$ induces an exponential scaling down of height over time, namely,*

$$h(t) = e^{-t/2} h(0).$$

Proof. In a sufficiently small ball around any point $p \in M$ we can write the height function as

$$h_p(q) = \langle N(p), f(q) - f(p) \rangle.$$

Noting that $N(p)$ and $f(p)$ are constant with respect to q, we have

$$\Delta h_p = \langle N(p), \Delta f \rangle = 2H\langle N(p), N \rangle.$$

In particular, when $q = p$ we get

$$H(p) = \tfrac{1}{2}(\Delta h_p)(p).$$

Now consider an evolution of the surface that looks like $\dot{h}_p = -h_p$ at some point p, i.e., an exponential scaling down of height over the tangent plane. Then differentiating $H(p)$ in time yields

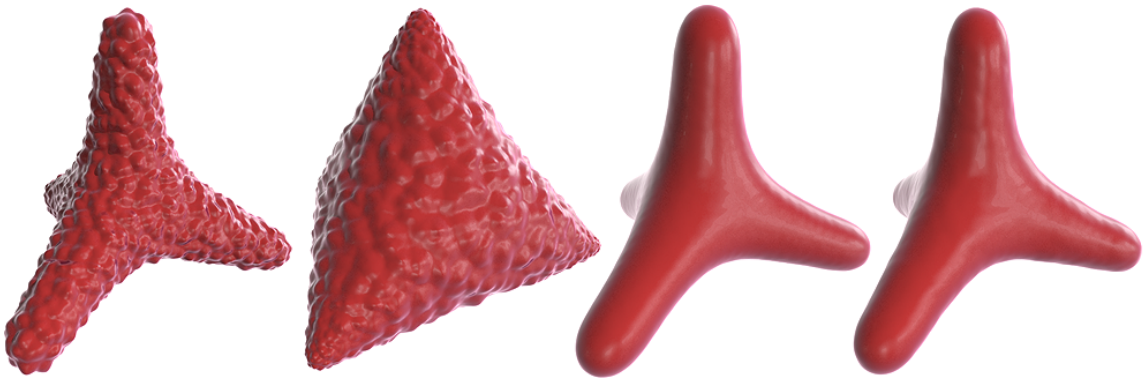
$$\dot{H}(p) = \tfrac{1}{2}(\Delta \dot{h}_p)(p) = -\tfrac{1}{2}(\Delta h_p)(p) = -\tfrac{1}{2}H(p),$$

i.e., the change in H must be proportional to H itself. This evolution is the same as the evolution $\dot{\rho} = -\frac{1}{2}H$, since a small normal deformation $\tilde{f} = f + \epsilon h N$ over a plane is isometric (hence conformal) up to first order. In particular, $dN = 0$ for a plane, hence

$$\tilde{g}(u,v) = \langle d\tilde{f}(u), d\tilde{f}(v) \rangle = g(u,v) + \epsilon(dh(u)\langle N, df(v)\rangle + dh(v)\langle N, df(u)\rangle) + O(\epsilon^2),$$

but $\langle N, df(u)\rangle = 0$ for all u. □

This observation can be confirmed experimentally—in this example, gradient flow with respect to curvature scales down features at all scales uniformly (center left), whereas filtering out low-frequency components of the flow (center right) closely approximates standard position-based Willmore flow (right):



To achieve this kind of behavior, one can simply filter the flow direction v—in particular, we use the

spectral filter

$$v \mapsto v - (\mathrm{id} - \sigma\Delta^k)^{-1} v,$$

which damps low-frequency motion by subtracting a regularized version of v from itself. The parameters $\sigma > 0$ and $k \in \mathbb{Z}$ control the degree of regularization and the filter shape, respectively; evaluating the filter amounts to solving a scalar Poisson equation (see Taubin [Tau95] for a more thorough discussion of this approach to filtering). For $k = 2$ this flow closely approximates traditional position-based Willmore flow, since Δ^{-2} approximates the norm on curvature induced by the $\mathscr{L}^2$ norm on position. The following example demonstrates the effect of parameters $k = 0, 1, 3$ (from left to right):

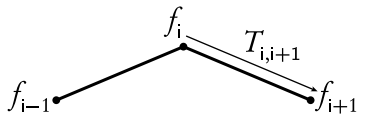
6.4.3 Isometric Curve Flows

Let $f : [0, L] \to \mathbb{R}^2$ an isometrically immersed curve with unit tangents $T = f'$ and curvature normal $\kappa N = T'$, as described in Section 4.2. A natural fairing energy is just the squared $\mathscr{L}^2$ norm of curvature

$$E_C(\kappa) := \int_0^L \kappa^2 d\ell = \|\kappa\|^2.$$

(Note that this energy would be rather difficult to express in terms of the positions f, involving second derivatives and a high degree of nonlinearity.) For open curves, the resulting flow $\dot{\kappa} = -2\kappa$ can be integrated without modification. In the case of a closed regular curve, however, $\dot{\kappa}$ must satisfy the linear conditions derived in Section 4.2, namely

$$\langle\langle \dot{\kappa}, 1 \rangle\rangle = \langle\langle \dot{\kappa}, f^x \rangle\rangle = \langle\langle \dot{\kappa}, f^y \rangle\rangle = 0.$$



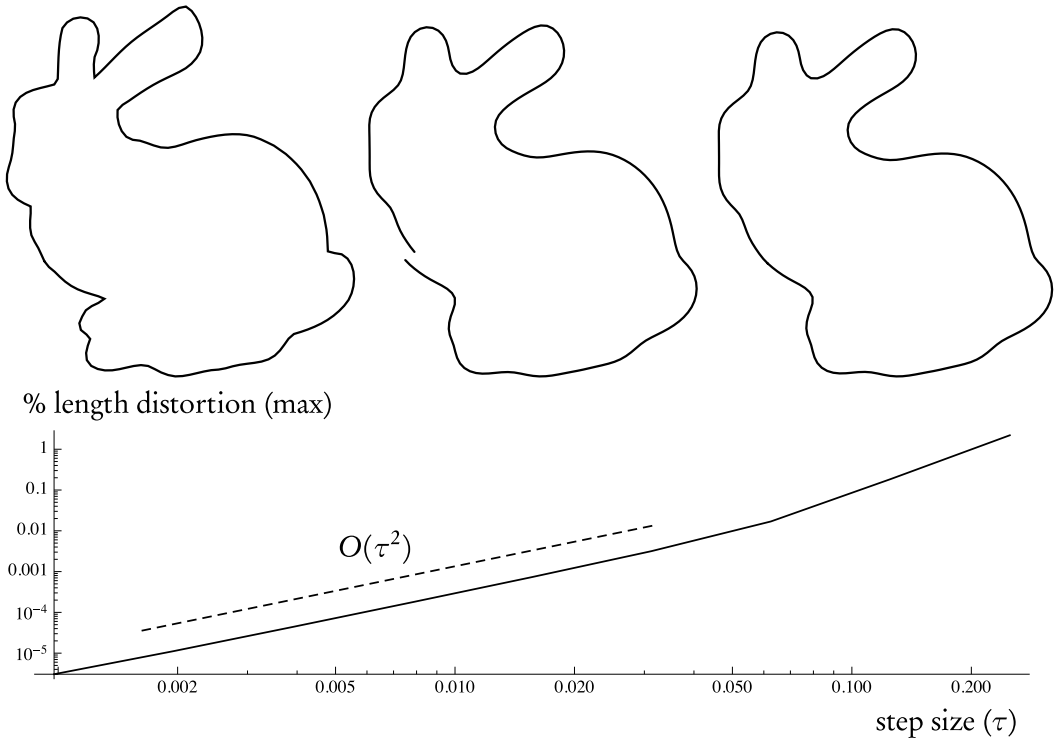
To numerically integrate this flow, we discretize the curve f as a collection of vertex coordinates $f_1, \ldots, f_m \in \mathbb{R}^2$, tangents $T_{ij} = f_j - f_i$ associated with edges, and curvatures $\kappa_i \in \mathbb{R}$ associated with vertices. On each iteration we apply the forward Euler method to the curvature values κ_i, enforcing constraints as described in Section 6.4.2. Letting $\ell_{ij} = |T_{ij}|$ denote edge lengths on the initial curve, we then recover tangents by computing the cumulative sums

$$\theta_k = \theta_0 + \sum_{i=1}^{k} \tfrac{1}{2}(\ell_{i-1,i} + \ell_{i,i+1})\kappa_i.$$

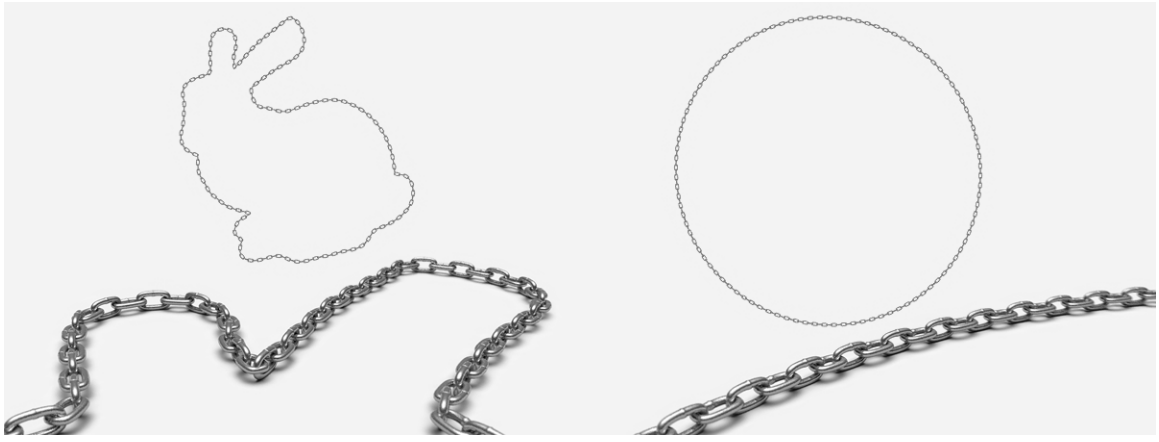
New tangent vectors are then given by $\tilde{T}_{ij} = \ell_{ij}(\cos\theta_i, \sin\theta_i)$.

In the smooth setting, length preservation follows from our isometry assumption. Numerically, however, we experience a small amount of discretization error, which we distribute uniformly by solving the optimization problem $\min_{\tilde{f}} |d\tilde{f} - \tilde{T}|^2$ for the vertex positions $\tilde{f}$ that best agree with our desired tangents $\tilde{T}$. Equivalently, we can solve the linear Poisson problem $\Delta\tilde{f} = \nabla \cdot \tilde{T}$. To recover positions in the discrete setting we solve the Poisson equation $L\tilde{f} = b$ where L is an $m \times m$ matrix with off-diagonal entries $L_{ij} = -1/\ell_{ij}$ for any edge (i,j) and diagonal entries $L_i = 1/\ell_{i-1,i} + 1/\ell_{i,i+1}$. The right-hand side b is the discrete divergence of the new tangent field, given by $b_i = \tilde{T}_{i,i+1}/\ell_{i,i+1} - \tilde{T}_{i-1,i}/\ell_{i-1,i}$ at each vertex. (Note that this system can be solved as a pair of *scalar* Poisson equations with either x- or y-components of $\tilde{T}$ on the right-hand side.)

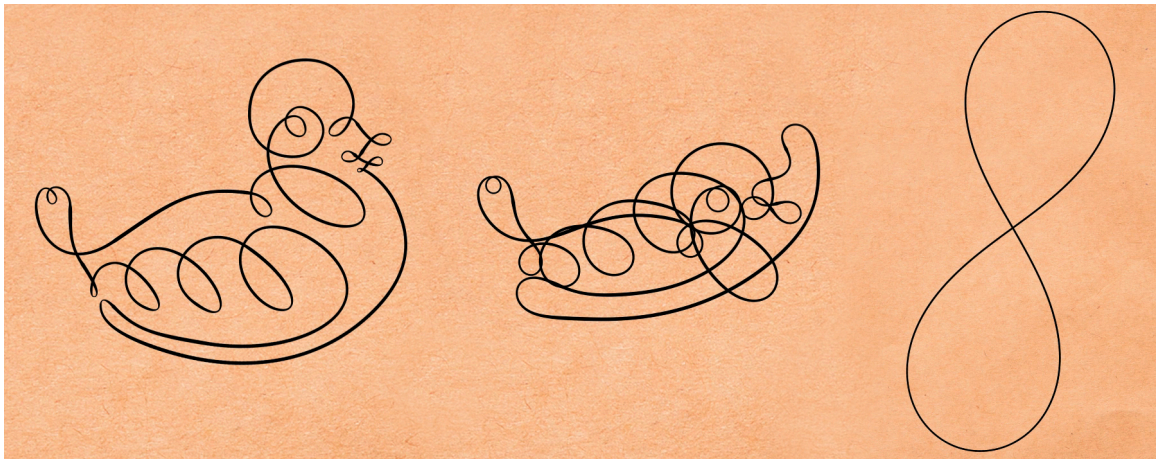
As demonstrated below, any remaining length distortion resulting from discretization is quite small even for large time steps, and converges quadratically under temporal refinement:



In this case we start with a polygon consisting of 200 edges (top left). Even after taking a large time step, the failure to close is slight (top center) and yields little length distortion when projected onto the nearest closed curve (top right). On the bottom we show a log-log plot of worst edge length distortion as a function of step size; the dashed line represents quadratic convergence. Visually, curves produced by our algorithm do not stretch as they evolve, as emphasized by the chain links used to represent curve edges in the following example:



Moreover, curves remain regular throughout the flow, ultimately flowing to the smoothest immersion of equal turning number:



In contrast, 1D mean curvature flow $\dot{f} = -2\kappa N$ develops sharp cusps that can be a source of numerical instability. Finally, the algorithm we describe is quite efficient, and remarkably stable: on a curve of 2k vertices, each integration step requires about 1.4ms of computation time (using a 2.4 GHz Core 2 Duo processor); a global minimum of the curvature energy E_C can be found in about six integration steps.

6.4.4 Conformal Surface Flows

Our treatment of surfaces closely parallels the curve case, except that instead of quantities f, T, and κ, we now work with the analogous quantities f, λ, and ρ, respectively. The immersion $f : M \to \mathbb{R}^3$ again describes the position of our surface in space; the map $\lambda : M \to \mathbb{H}$ describes how tangents *change* from one surface to the next; likewise the map $\rho : M \to \mathbb{R}$ describes a *change* in curvature (Section 4.3).

The key motivation for working with this setup is that Willmore energy now amounts to a simple quadratic function of mean curvature half-density:

$$E_W(\mu) = \int_M H^2 |df|^2 = \int_M \mu^2 = ||\mu||^2.$$

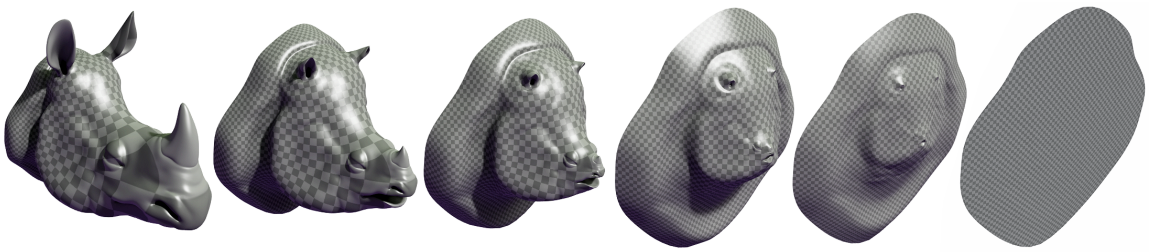
Gradient flow with respect to μ is then quite simply $\dot{\mu} = -2\mu$, or equivalently, $\dot{\rho} = -H$ (see Proposition 6.4.1). We again apply the forward Euler scheme

$$\rho^{k+1} = \rho^k - 2\tau H^k,$$

where H^k is the mean curvature of the current mesh computed via the cotan operator [DMSB99].

In practice we express ρ with respect to the previous surface in the flow, hence $\rho^k = 0$ and we simply solve the smallest eigenvalue problem $(\mathcal{D} + \tau H^m)\lambda = \gamma\lambda$ as described in Section 6.6. We solve these equations as described in Chapter 5, except that for convenience we now specify ρ via a single value at each vertex—Section 6.6 describes a simple facewise construction of the corresponding matrix.

For topological disks the flow $\dot{\rho} = -H$ can be integrated without further modification. Here, for instance, running conformal Willmore flow on a rhino head shrinks mean curvature as quickly as possible:



For more general surfaces, however, our flow direction $\dot{\rho}|df|$ must be projected onto the tangent space $T\mathcal{M}$ of the conformal shape space $\mathcal{M}$. In other words, it must satisfy the linear constraints derived in Section 4.5, which we summarize here:

- **Total Curvature**: As with curves, total curvature must remain constant, which is enforced via the constraint $\langle\langle \dot{\rho}, \mathbf{1} \rangle\rangle = 0$.

- **Exactness**: For non-simply connected surfaces (e.g., a torus or annulus) we must add the constraint $\langle\langle \dot{\rho}, Z_i^x \rangle\rangle = \langle\langle \dot{\rho}, Z_i^y \rangle\rangle = \langle\langle \dot{\rho}, Z_i^z \rangle\rangle = 0$ for Z_i solutions to $\mathcal{D}Z_i = \upsilon_i$, where $\{\upsilon_i\}$ is a basis for harmonic vector fields on the surface.

- **Inversion**: sphere inversions preserve both Willmore energy and conformal structure, but may produce area distortion [Bla29]. To avoid inversions, we apply the constraint $\langle\langle \dot{\rho}, N^x \rangle\rangle = \langle\langle \dot{\rho}, N^y \rangle\rangle = \langle\langle \dot{\rho}, N^z \rangle\rangle = 0$, where $N^{x,y,z}$ are the scalar components of the unit normal N.

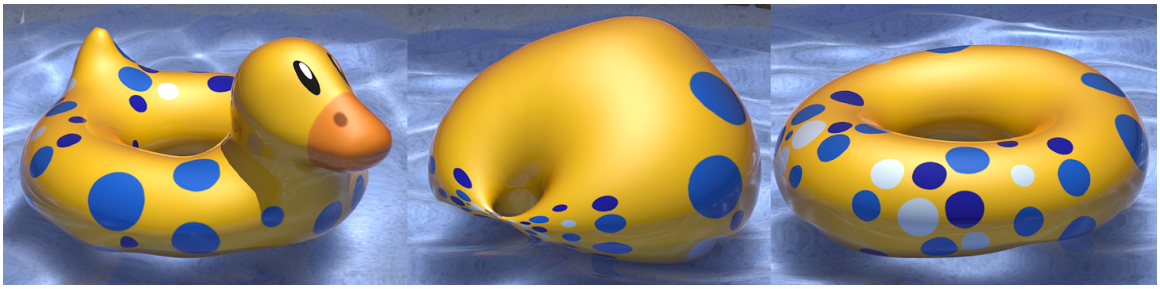
- **Möbius Balancing** (optional): We can also use sphere inversions (Section 4.5.3) to reduce area distortion, in particular by adding $\langle a, N \rangle$ to $\dot{\rho}$, where $a \in \mathbb{R}^3$ is computed as described in Section 6.4.5.

Overall we have a constraint basis

$$\{\mathbf{1}, N^{x,y,z}, Z_i^{x,y,z}\};$$

the corresponding constraints are enforced as described in Section 6.4.2. On a triangulated surface N is the vector of vertex normals; the functions Z_i are computed as described in Section 6.4.5. Note that Möbius balancing must be applied after the sphere inversion constraint and before any remaining constraints.

The following example demonstrates a flow on a surface of nontrivial topology with and without the exactness constraint—notice that this constraint is crucial not only for preserving texture, but also for preventing degenerate geometry:

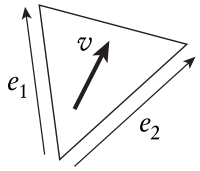


initial surface without constraint with constraint

The next example illustrates scale artifacts resulting from unnecessary sphere inversions (top), and the improved area distribution resulting from Möbius balancing (bottom):

6.4.5 Implementation

6.4.5.1 Exactness



On a triangulated surface, we compute the functions $Z_i \in \mathbb{H}^{|V|}$ as follows:

I. Pick $2G$ random discrete 1-forms $\alpha_i \in \mathbb{R}^{|E|}$.

II. Extract harmonic components ω_i via Hodge decomposition.

III. Orthogonalize $\{\omega_i\}$ via the Gram-Schmidt process.

IV. Construct corresponding vector fields $v_i \in \mathrm{Im}\mathbb{H}^{|F|}$.

V. Solve $\mathsf{X}Z_i = v_i$ for each harmonic vector field v_i.

For Hodge decomposition, we use the method of Desbrun et al. [DKT05]. The matrix X is given in Section 6.6, letting $\rho = 0$. To get the vector fields v_i, we construct appropriate vectors $v \in \mathrm{Im}\mathbb{H}$ in each triangle. Let e_1 and e_2 be two edge vectors (see inset), and let ω_i^1, ω_i^2 be the corresponding 1-form values. Then v must satisfy $\langle v, e_1 \rangle = \omega_i^1$ and $\langle v, e_2 \rangle = \omega_i^2$. If we express v as a linear combination $v = a_1 e_1 + a_2 e_2$, we get a 2×2 system

$$\begin{bmatrix} |e_1|^2 & \langle e_1, e_2 \rangle \\ \langle e_1, e_2 \rangle & |e_2|^2 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} \omega_i^1 \\ \omega_i^2 \end{bmatrix},$$

for the coefficients $a_1, a_2 \in \mathbb{R}$.

6.4.5.2 Möbius Balancing

On a mesh with vertex areas $\mathcal{V}_i$, desired vertex areas $\mathcal{V}_i^*$, and vertex normals N_i the balancing direction a is computed as follows:

I. Compute scale factors $u_i \leftarrow \frac{1}{2}\log(\mathcal{V}_i / \mathcal{V}_i^*)$ at each vertex.

II. Remove the mean via $u_i^0 \leftarrow u_i - \sum_{i=1}^{|V|} \mathcal{V}_i u_i / \sum_{i=1}^{|V|} \mathcal{V}_i$.

III. Compute the vector $v \leftarrow \sum_i \mathcal{V}_i f_i u_i^0$.

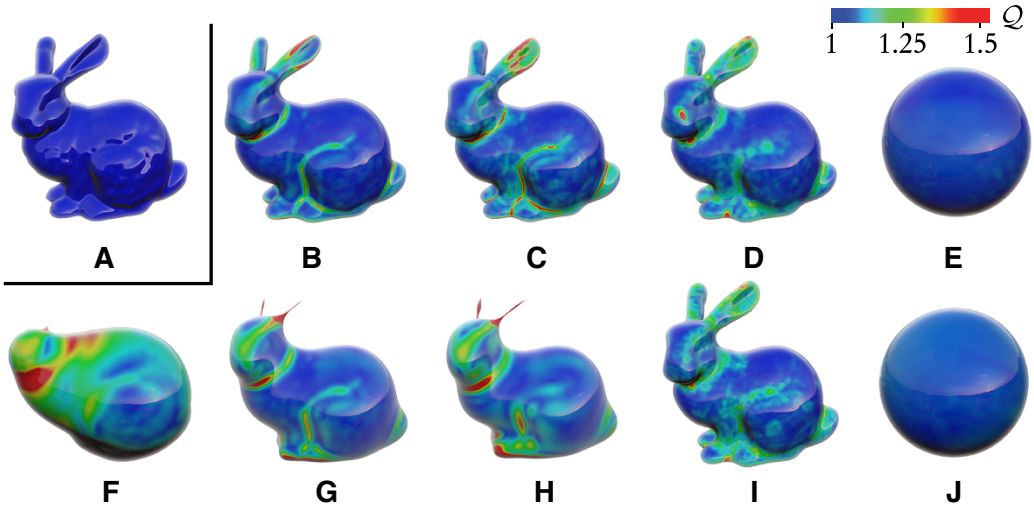
IV. Solve $Ba = v$ where $B = 4\sum_i \mathcal{V}_i N_i N_i^T$.

V. Normalize $a \leftarrow a / \langle Ba, a \rangle^{1/2}$.

6.4.6 Comparison

We compared our method to a wide variety of existing algorithms for surface fairing in terms of numerical stability and preservation of conformal structure. All methods were carefully optimized and run on the same 2.4 GHz Intel Core 2 Duo machine; for each method we used appropriate solvers from the SuiteSparse package [CDHR08], re-using both symbolic and numeric factorizations wherever possible. See Section 6.6 for further details about our implementation.

The figure below shows the result of running each flow for 10s of CPU time, using the maximum stable time step for each method. Here we use an isotropic mesh with near-uniform edge length [BK04], which represents a best-case scenario for traditional methods. Algorithms used were (**B**) explicit mean curvature flow [Bra92], (**C**) explicit bi-Laplacian flow [Tau95] using the cotangent Laplacian; (**D**) implicit discrete Willmore flow [BS05], (**E**) implicit modified mean curvature flow [KSBC12], (**F**) implicit volume-controlled mean curvature flow [EPT+07], (**G**) implicit mean curvature flow [DMSB99], (**H**) implicit bi-Laplacian flow via backward Euler, (**I**) implicit Willmore flow based on isometric bending [WBH+07], and (**J**) our conformal fairing flow. Notice that many of these methods do not make much progress, despite being very cheap to evaluate for an individual time step. Most methods also exhibit significant conformal distortion, especially in regions of high curvature. Notable exceptions are our method and the modified mean curvature flow of Kazhdan et al., both of which rapidly flow to the unit sphere. As noted before, however, the latter produces conformal distortion and sharp geometric features during intermediate steps; moreover, it applies only to surfaces with spherical topology.

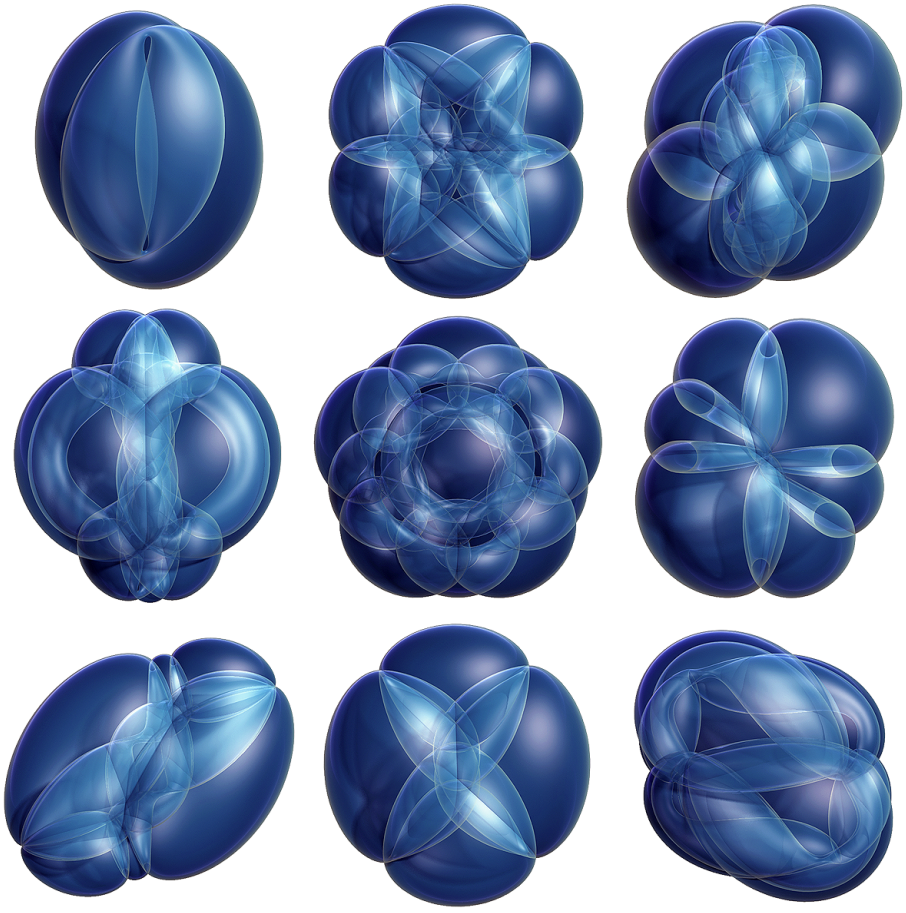


6.5 Special Surfaces

A numerical discretization of spin transformations allows us to compute special surfaces that are difficult or impossible to construct via closed-form expressions. Apart from a simple aesthetic appeal, these surfaces provide interesting visualizations of physical phenomena (Section 6.5.1), and can be valuable in domains such as geometric design or computational architecture (Section 6.5.3).

6.5.1 Dirac Spheres

In Section 5.8.2 we computed Dirac spheres corresponding to the eigenvalue $\gamma = 1$ to numerically validate our discretization of Equation 3.2. Here we visualize for the first time several Dirac spheres corresponding to higher energy states:



Note that when $\rho = 0$ (as in these examples) one must be careful to remove the constant component from each solution eigenvector, since constant functions are in the null space of the system $D^*D\lambda = \gamma B^*D\lambda$.

In the examples above we picked arbitrary solutions λ from several eigenspaces. To get a more structured depiction of the Dirac spheres, we can start with a canonical expression for the spherical spinors. In particular, let $(\theta, \phi) \in [0, \pi] \times [0, 2\pi)$ be polar coordinates on the unit sphere. Then for any nonzero integer κ and integer $-|\kappa| \leq s < |\kappa|$, the spherical spinors $\Omega_{\kappa,\mathsf{m}}(\theta, \phi)$ for $\mu = s + \frac{1}{2}$ can be expressed as quaternionic functions

$$\Omega_{\kappa,\mathsf{m}}(\theta, \phi) := \mathrm{Re}\psi_2 - \mathrm{Im}\psi_1 i - \mathrm{Re}\psi_1 j + \mathrm{Im}\psi_2 k,$$

where ψ_1, ψ_2 are complex-valued functions given by

$$\begin{aligned} \psi_1 &:= \mathrm{sgn}(-\kappa)\sqrt{\tfrac{\kappa-\mu+\frac{1}{2}}{2\kappa+1}}\, Y_{\ell,\mu-1/2}(\theta, \phi), \\ \psi_2 &:= \sqrt{\tfrac{\kappa+\mu+\frac{1}{2}}{2\kappa+1}}\, Y_{\ell,\mu-1/2}(\theta, \phi), \end{aligned}$$

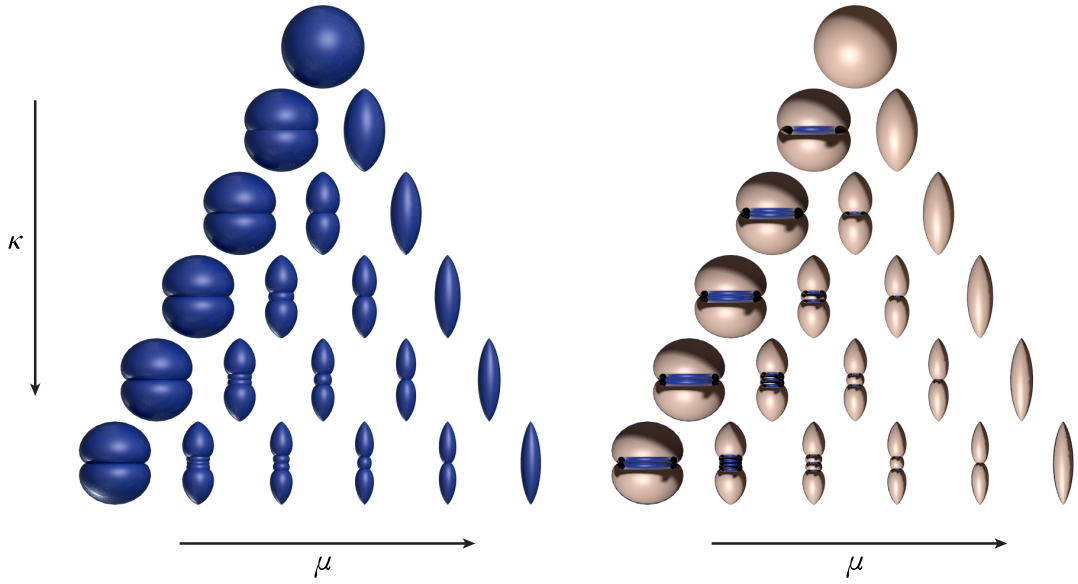
with $\ell := |\kappa + \frac{1}{2}| - \frac{1}{2}$ and $Y_{\ell,\mathsf{m}}$ is the complex-valued function

$$Y_{\ell,\mathsf{m}}(\theta, \phi) := \sqrt{\frac{2\ell+1}{4\pi}\frac{(\ell-\mathsf{m})!}{(\ell+\mathsf{m})!}} P_{\ell,\mathsf{m}}(\cos\theta) e^{im\phi}.$$

Here $P_{\ell,\mathsf{m}}(\xi)$ are the associated Legendre functions of the first kind, given by

$$P_{\ell,\mathsf{m}}(\xi) := \frac{(-1)^{\mathsf{m}}}{2^\ell \ell!}(1-\xi^2)^{\mathsf{m}/2}\frac{d^{\ell+\mathsf{m}}}{d\xi^{\ell+\mathsf{m}}}(\xi^2-1)^\ell.$$

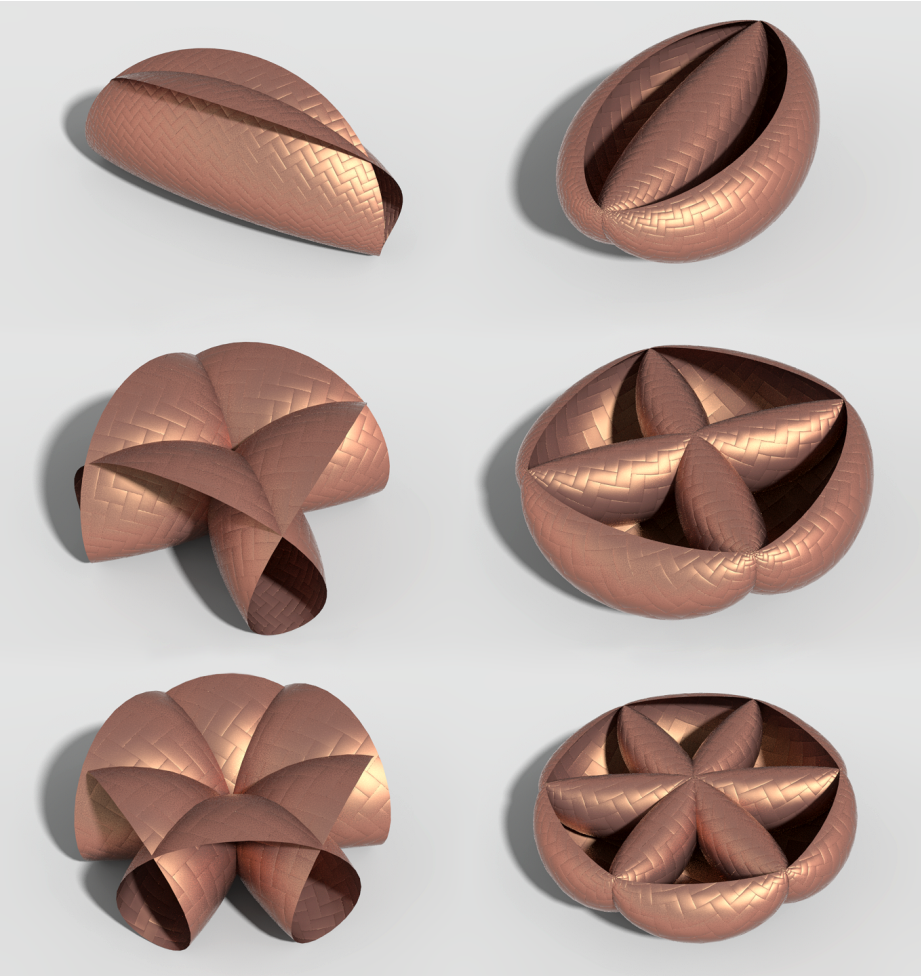
Letting $\lambda = \Omega_{\kappa,\mathsf{m}}$ and reconstructing the corresponding spin transformations $\tilde{f}$ yields the following table of conformally immersed surfaces; each row effectively represents a different energy level for an orbiting hydrogen electron:



Note that because of symmetry, we plot only surfaces corresponding to $\mu > 0$; the right half of the image shows a cross section of each surface.

6.5.2 Dirac Disks

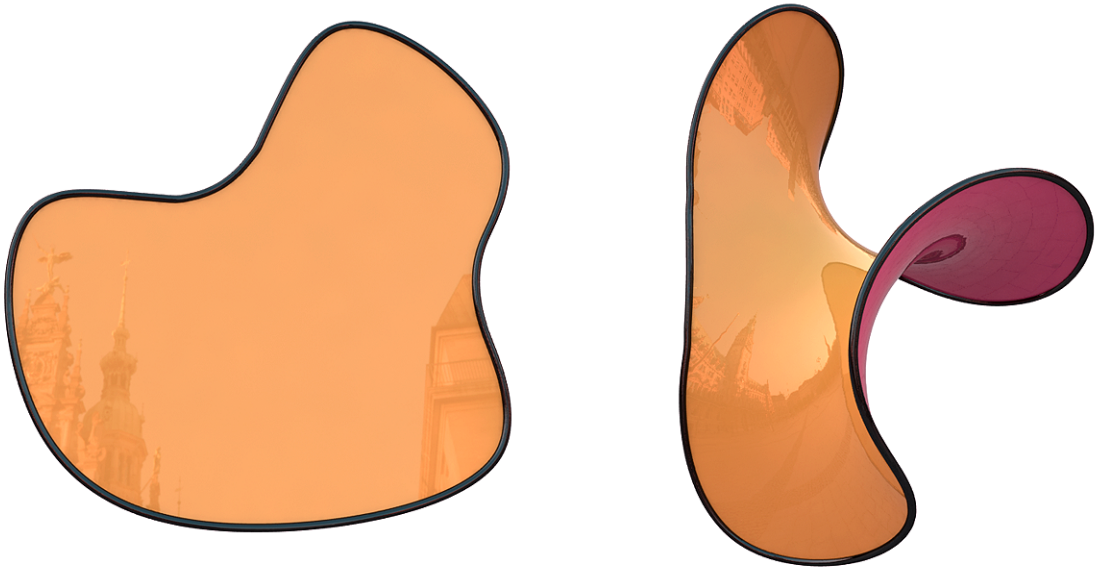
A similar experiment can be repeated when M is a topological disk and the immersion f takes M to a flat circular disk in the plane. Here we fix the direction of the binormals along ∂M (i.e., $\widetilde{B} = B$) by applying boundary conditions described in Section 5.7):



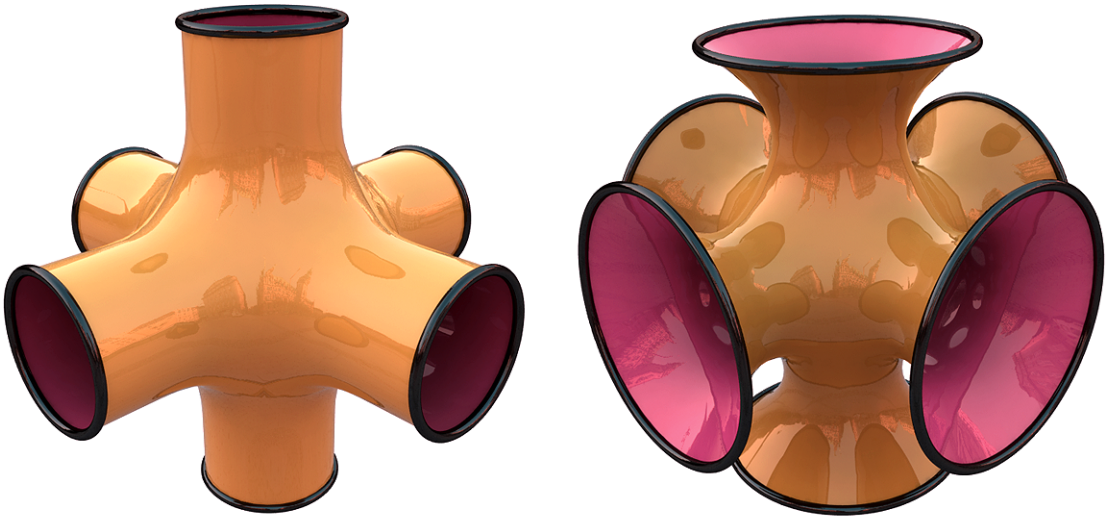
6.5.3 Constant Mean Curvature

Surfaces of constant mean curvature (CMC) are well-studied in surface theory; physically they can be thought of as describing the geometry of soap films. For instance, a soap film attached to a fixed boundary yields a *minimal surface* ($H = 0$); a simply connected soap film without boundary will buckle into a perfectly round sphere ($H > 0$).

Traditional numerical schemes for computing CMC surface proceed by iteratively minimizing an appropriate energy functional, e.g., via gradient descent. In contrast, the scheme described in Section 6.3.1 allows one to compute minimal surfaces "instantly", i.e., by solving a single eigenvalue problem. More explicitly, let M be a topological disk, and let $f : M \to \mathbb{R}^3$ be any immersion into the plane, i.e., $f(M) \subset \mathbb{R}^2 \subset \mathbb{R}^3$. Then setting the curvature potential to $\rho = 0$ while prescribing new boundary directions (e.g., $\widetilde{T}$ or $\widetilde{B}$) results in a surface of zero mean curvature half-density, and hence, zero mean curvature ($\mathcal{Q}_{\text{mean}} = 1.054$):



We can also construct CMC surfaces via a completely different procedure: start with a curved surface and flow to its Willmore minimizer (Section 6.4). Suppose that M is a topological disk with fixed tangents along the boundary. Then the Dirac operator $\mathcal{D}$ has a continuous spectrum and we can therefore achieve any change whatsoever in mean curvature half-density. In particular, we can remove the initial curvature (by setting $\rho = -H$) to again obtain a minimal surface. Empirically, this approach also appears to work for nonsimply connected surfaces exhibiting a high degree of symmetry ($\mathcal{Q}_{\text{mean}} = 1.070$):



As with boundary prescription, this surface was generated by solving a single eigenvalue problem, i.e., we did not need to run an iterative flow. Alternatively, by prescribing new binormals along the

boundary we obtain a CMC surface of *nonzero* curvature.

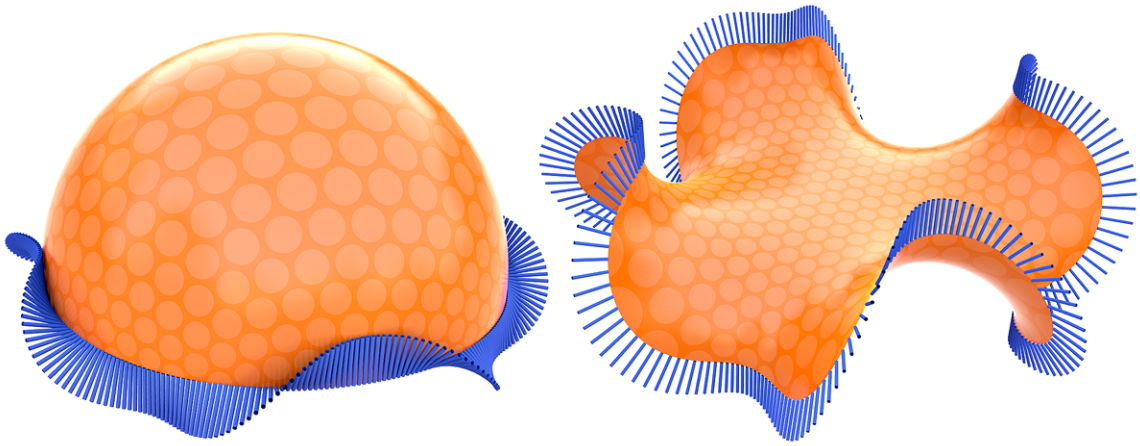
Proposition 6.5.1. *Let f be a conformally immersed disk with fixed binormals along the boundary. Then f is a stationary point of Willmore flow if and only if it has constant mean curvature.*

Proof. The steepest descent direction on Willmore energy is found by projecting the unconstrained gradient direction $\dot{\rho}|df| = -H|df|$ onto the tangent space $T_0\mathcal{M}$ (Section 6.4). Moreover, from Proposition 4.5.8 we know that $T_0\mathcal{M}$ consists of half-densities of mean zero whenever M is a topological disk with fixed binormals. But then f is a stationary point if and only if $\dot{\rho} - \text{mean}(\dot{\rho}) = 0$, i.e., if at every point

$$H = \text{mean}(H).$$

□

In this example, for instance, we prescribe oscillating binormals $\widetilde{B}$ along the boundary of a hemisphere:

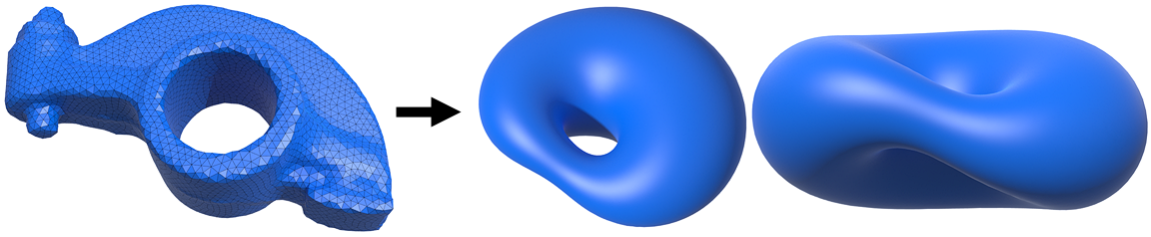


In addition to being efficient to compute, a notable fact about this approach is that (unlike existing methods for computing CMC surfaces) the conformal structure of the initial domain is naturally preserved, as illustrated by the circular pattern in the example above. This feature is valuable both from a numerical perspective (no remeshing is required) as well as an aesthetic/design perspective, e.g., in architectural applications one often seeks repeating patterns of similarly shaped elements [PAHK07].

6.5.4 Complex Tori

The *Willmore problem* asks the question: how smooth can one make a surface of a given genus? More precisely, what is the smallest Willmore energy E_W among all immersions f of a given surface M? It is

a widely held belief that the global minimizer of Willmore energy among all immersions of the torus T^2 is the *Clifford torus*, exhibiting a Willmore energy of $2\pi^2$ [Wil65]. However, little is known about so-called *constrained Willmore surfaces*, i.e., minimizers of Willmore energy within a given conformal class. The algorithm described in Section 6.4 allows us to examine this question experimentally. In particular, by flowing a torus with a non-rectangular conformal structure to its minimizer we find a surface with an appearance quite different from the Clifford torus:



Notably, this torus appears "twisted", owing to the fact that the generating lattice of its conformal structure is not rectangular. In the future we hope to continue such experiments in order to find the "smoothest complex tori" in each class, providing a concrete visualization of the often abstract notion of global conformal structure.

6.6 Numerics and Performance

The main computational cost when computing conformal deformations is solving the eigenvalue problem

$$(\mathcal{D} - \rho)\lambda = \gamma\lambda$$

for the smallest eigenvalue γ and corresponding eigenvector λ.

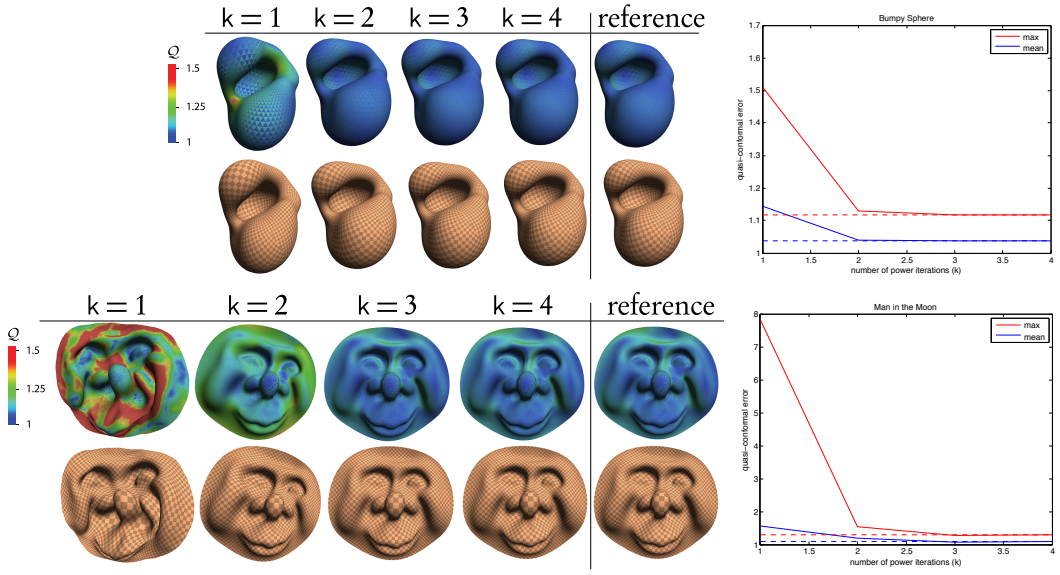
A fairly common misconception in geometry processing is that eigenvalue problems are expensive to solve, or require sophisticated numerical libraries (like ARPACK [LSY97]). In reality, however, the cost of finding the *smallest* eigenvalue/eigenvector pair is nearly identical to the cost of solving a single linear system; moreover, it can be found via a very simple algorithm: In our case X is the matrix used

Algorithm 1 The Inverse Power Method

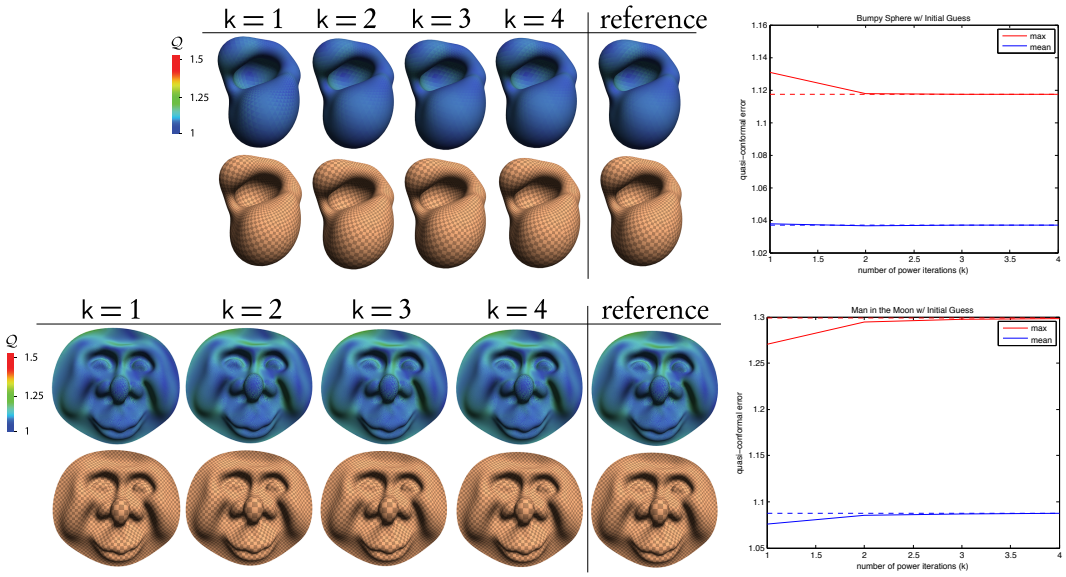
Require: Initial guess λ_0.
1: for $i = 1, \ldots, k$ do
2: Solve $X\lambda_i = \lambda_{i-1}$
3: $\lambda_i \leftarrow \lambda_i / ||\lambda_i||$
4: end for

in the standard eigenvalue problem (Equation 5.3), which is symmetric positive-semidefinite with roughly 7 nonzeros in each row/column. In each step of the inverse power method we must solve the same linear system, with different data on the right-hand side. We can therefore prefactor X and re-use this factorization in each iteration of the power method; the cost of applying backsubstitution in subsequent iterations is almost negligible in comparison. Moreover, for the geometry processing problems we are interested in solving this method typically converges to an acceptable solution after very few iterations (one or two), as discussed below.

The following figures depict two tests: `bumpy`, where we add random bumps to a sphere, and `moon`, where we paint a face on a disk. Starting from a random initial guess the inverse power method tends to produce acceptable results after only a few iterations:



A more intelligent initial guess sets $\lambda = \mathbf{1}$, which corresponds to the identity transformation. In this case we obtain a result virtually indistinguishable from the reference solution after only a single iteration:



In comparison, matrix factorization followed by a single application of backsubstitution (which requires an identical amount of computation) is currently among the most efficient ways to solve linear systems for geometry processing [BBK05].

In our experiments, we use the routines implemented in CHOLMOD [CDHR08] to perform sparse Cholesky factorization. On a 2.4 GHz Core 2 Duo laptop, a single iteration of the power method takes 0.3 seconds on a mesh with 8k faces (`bumpy`) and 1.26 seconds on a mesh with 33k faces (`moon`); this timing includes additional overhead due to calling CHOLMOD from within MATLAB.

6.6.1 Building the Eigenvalue System

In practice it is often convenient to build the matrix X directly, rather than building it up from constituent matrices as described in Section 5.5. The following algorithm provides a simple *facewise* construction of X: Here k is the index of the current triangle, $\mathcal{A}_\mathsf{k}$ is its area, and ρ_k is the desired change

Algorithm 2 Facewise Construction of Eigenvalue System

1: for $\mathsf{k} = 1, \ldots, |F|$ do
2: $\quad a \leftarrow -\frac{1}{4\mathcal{A}_\mathsf{k}}$
3: $\quad b \leftarrow \rho/6$
4: $\quad c \leftarrow \rho^2 \mathcal{A}_\mathsf{k}/9$
5: $\quad$ for all $(\mathsf{i},\mathsf{i}) \in \{1,2,3\} \times \{1,2,3\}$ do
6: $\quad\quad \mathsf{X}_{\mathsf{ij}} += a e_\mathsf{i} e_\mathsf{j} + b(e_\mathsf{j} - e_\mathsf{i}) + c$
7: $\quad$ end for
8: end for

in mean curvature half-density. The inner loop visits all ordered pairs of edges e_1, e_2, e_3 of the current

face. In this version of the algorithm we assume that ρ is expressed as a single value per face. If one instead wishes to associate ρ with vertices (as is done for Willmore flow), then Line 6 can be replaced with the expression

$$X_{ij} += -\frac{e_i e_j}{4\mathcal{A}_k} + \frac{1}{6}(\rho_i e_j - \rho_j e_i) + \frac{\mathcal{A}_k}{9}\rho_i \rho_j.$$

Listing 6.1 provides an implementation of this algorithm in C++ (with ρ stored on faces).

Listing 6.1: Facewise Construction of Eigenvalue System in C++

```
void buildEigenvalueProblem( const vector<Face>& faces,
                             const vector<Vertex>& vertices,
                             QuaternionSparseMatrix& E )
{
   // allocate a sparse |V|x|V| matrix
   int nV = vertices.size();
   E.resize( nV, nV );

   // visit each face
   for( size_t k = 0; k < faces.size(); k++ )
   {
      double   A = face[k].area();
      double rho = face[k].rho;

      // compute coefficients
      double a = -1. / (4.*A);
      double b = rho / 6.;
      double c = A*rho*rho / 9.;

      // get vertex indices
      int I[3] =
      {
         faces[k].vertex[0],
         faces[k].vertex[1],
         faces[k].vertex[2]
      };

      // compute edges across from each vertex
      Quaternion e[3];
      for( int i = 0; i < 3; i++ )
      {
         e[i] = vertices[ I[ (i+2) % 3 ]] -
                vertices[ I[ (i+1) % 3 ]] ;
      }

      // increment matrix entry for each ordered pair of vertices
      for( int i = 0; i < 3; i++ )
      for( int j = 0; j < 3; j++ )
      {
         E(I[i],I[j]) += a*e[i]*e[j] + b*(e[j]-e[i]) + c;
      }
   }
}
```

6.6.2 Building the Poisson Problem

The Poisson problem $\Delta \tilde{f} = \nabla \cdot \tilde{e}$ is solved using standard linear finite element discretization of the Laplacian [Mac49], which can be expressed using the well-known *cotangent formula*:

$$(\mathsf{L}\tilde{f})_i = \frac{1}{2} \sum_{j \in \mathcal{N}(i)} (\cot \alpha_j + \cot \beta_j)(\tilde{f}_i - \tilde{f}_j),$$

where $(\mathsf{L}\tilde{f})_i$ denotes the weak Laplacian of $\tilde{f}$ at vertex i, $\mathcal{N}(i)$ is the set of vertices adjacent to vertex i, and α_j and β_j are the two angles opposite edge (i, j).

Chapter 7

Conclusion

We have presented a starting point for developing general-purpose conformal geometry processing algorithms, but much work remains to be done. Most notably, our present discretization of spin transformations corresponds to a piecewise linear finite element approximation that does not preserve essential structures of the smooth theory—in the future, it may be useful to develop a truly "discrete" theory that captures phenomena like conformal equivalence classes, spin equivalence classes, etc., at the discrete level, á la Springborn et al. [SSP08]. Even with the current discretization, however, there are many interesting avenues to pursue. In particular, canonical visualizations of complex tori (as described in Section 6.5.4) could be obtained by applying conformal Willmore flow to *Hopf tori* [Pin85]; one is also tempted to produce a conformal *sphere eversion* in the vein of Francis et al. [FSH98]. From a more practical perspective, the vector analysis in Section 3.4 suggests an interesting approach to constructing discrete differential operators for various surface descriptions, since one can simply discretize the Dirac operator $\mathcal{D}$ and obtain most other differential operators via operations that are computationally quite trivial (e.g., taking the real part, multiplying by normals, etc.). This approach may provide a flexible framework for developing operators on more general polygonal surfaces, point clouds, higher-order surface descriptions, etc. Finally, the extraordinary stability exhibited by the curvature-space approach to Willmore flow (Section 6.4) may prove especially valuable in computer graphics, where meshes come from many disparate sources (e.g., simulation, artist modeling, etc.) that do not often provide guarantees on mesh quality. Note that although we have developed this machinery in the context of Willmore flow, it could be applied to other flows of interest by simply changing the flow direction $\dot{\rho}$. A promising avenue for future work is a more thorough investigation of curvature-based filtering, as already hinted at in Sections 6.2 and 6.4.2.1.

Bibliography

[BBK05] Mario Botsch, David Bommes, and Leif Kobbelt. Efficient linear system solvers for mesh processing. In *IMA Conference on the Mathematics of Surfaces*, pages 62–83. Springer, 2005.

[BCGB08] Mirela Ben-Chen, Craig Gotsman, and Guy Bunin. Conformal Flattening by Curvature Prescription and Metric Scaling. *Comp. Graph. Forum*, 27(2):449–458, 2008.

[BCWG09] Mirela Ben-Chen, Ofir Weber, and Craig Gotsman. Variational Harmonic Maps for Space Deformations. *ACM Trans. Graph.*, 28(3):34:1–34:11, 2009.

[BDS+12] Sofien Bouaziz, Mario Deuss, Yuliy Schwartzburg, Thibaut Weise, and Mark Pauly. Shape-Up: Shaping Discrete Geometry with Projections. *Proc. Symp. Geom. Proc.*, 31(5), 2012.

[BK04] Mario Botsch and Leif Kobbelt. A Remeshing Approach to Multiresolution Modeling. In *Proc. Symp. Geom. Proc.*, pages 185–192, 2004.

[Bla29] Wilhelm Blaschke. *Vorlesungen über Differentialgeometrie III*. Springer, 1929.

[BP13] Christoph Bohle and Ulrich Pinkall. Conformal Deformations of Immersed Discs in R^3 and Elliptic Boundary Value Problems. *ArXiv e-prints*, 2013.

[Bra92] Ken Brakke. The surface evolver. *Experiment. Math.*, 1(2):141–165, 1992.

[BS05] Alexander Bobenko and Peter Schröder. Discrete Willmore Flow. In *Proc. Symp. Geom. Proc.*, pages 101–110, 2005.

[BS08] Mario Botsch and Olga Sorkine. On Linear Variational Surface Deformation Methods. *IEEE Trans. Vis. Comp. Graph.*, 14(1):213–230, 2008.

[Can70] P. B. Canham. The Minimum Energy of Bending as a Possible Explanation of the Biconcave Shape of the Human Red Blood Cell. *J. Th. Bio.*, 26(1):61–81, 1970.

[CDD+04] U. Clarenz, U. Diewald, G. Dziuk, M. Rumpf, and R. Rusu. A Finite Element Method for Surface Restoration with Smooth Boundary Conditions. *Comput. Aided Geom. Des.*, 21(5):427–445, 2004.

[CDHR08] Yanqing Chen, Timothy A. Davis, William W. Hager, and Sivasankaran Rajamanickam. CHOLMOD, Supernodal Sparse Cholesky Factorization and Update/Downdate. *ACM Trans. Math. Softw.*, 35, 2008.

[CG91] George Celniker and Dave Gossard. Deformable Curve and Surface Finite-Elements for Free-Form Shape Design. *Comp. Graph. (Proc. of ACM/SIGGRAPH Conf.)*, 25(4):257–266, 1991.

[CM09] T. H. Colding and W. P. Minicozzi, II. Generic mean curvature flow I; generic singularities. *ArXiv e-prints*, 2009.

[CPS11] Keenan Crane, Ulrich Pinkall, and Peter Schröder. Spin Transformations of Discrete Surfaces. *ACM Trans. Graph.*, 30(4):104:1–104:10, 2011.

[CPSS10] Isaac Chao, Ulrich Pinkall, Patrick Sanan, and Peter Schröder. A simple geometric model for elastic deformations. In *Proc. ACM/SIGGRAPH*, 2010.

[dGV08] Fernando de Goes, Siome Goldenstein, and Luiz Velho. A simple and flexible framework to adapt dynamic meshes. *Comp. & Graph.*, 32(2):141–148, 2008.

[DKT05] Mathieu Desbrun, Eva Kanso, and Yiying Tong. Discrete Differential Forms for Computational Modeling. In *ACM SIGGRAPH 2005 Courses*, SIGGRAPH '05, 2005.

[DKT08] Mathieu Desbrun, Eva Kanso, and Yiying Tong. Discrete Differential Forms for Computational Modeling. In Alexander I. Bobenko, Peter Schröder, John M. Sullivan, and Günther M. Ziegler, editors, *Discrete Differential Geometry*, volume 38 of *Oberwolfach Seminars*, pages 287–324. Birkhäuser Verlag, 2008.

[DMA02] Mathieu Desbrun, Mark Meyer, and Pierre Alliez. Intrinsic Parameterizations of Surface Meshes. *Comp. Graph. Forum*, 21(3):209–218, 2002.

[DMSB99] Mathieu Desbrun, Mark Meyer, Peter Schröder, and Alan Barr. Implicit Fairing of Irregular Meshes using Diffusion and Curvature Flow. In *Proc. ACM/SIGGRAPH*, pages 317–324, 1999.

[Duf59] R. Duffin. Distributed and Lumped Networks. *J. Math. Mech.* [continued as *Indiana Univ. Math. J.*], 8:793–826, 1959.

[EP09] Michael Eigensatz and Mark Pauly. Positional, Metric, and Curvature Control for Constraint-Based Surface Deformation. *Comp. Graph. Forum*, 28(2):551–558, 2009.

[EPT+07] Ilja Eckstein, Jean-Philippe Pons, Yiying Tong, C.C. Jay Kuo, and Mathieu Desbrun. Generalized Surface Flows for Mesh Processing. In *Proc. Symp. Geom. Proc.*, pages 183–192, 2007.

[ESP08] Michael Eigensatz, Robert W. Sumner, and Mark Pauly. Curvature-Domain Shape Processing. *Comp. Graph. Forum*, 27(2):241–250, 2008.

[FSH98] George Francis, John M. Sullivan, and Chris Hartman. Computing sphere eversions. In Hans-Christian Hege and Konrad Polthier, editors, *Visualization and Mathematics*, pages 237–255. Springer Verlag, Heidelberg, 1998.

[GWC+04] Xianfeng Gu, Yalin Wang, Tony Chan, Paul Thompson, and Shing-Tung Yau. Genus zero surface conformal mapping and its application to brain surface mapping. *IEEE Trans. Med. Imag.*, 23(8), 2004.

[GY03] Xianfeng Gu and Shing-Tung Yau. Global Conformal Surface Parameterization. In *Proc. Symp. Geom. Proc.*, pages 127–137, 2003.

[GZLY11] Xianfeng Gu, Wei Zeng, Feng Luo, and Shing-Tung Yau. Numerical computation of surface conformal mappings. *Comp. Meth. & Fun. Theory.*, 11(2):747–787, 2011.

[Hel73] W. Helfrich. Elastic Properties of Lipid Bilayers: Theory and Possible Experiments. *Z. Naturf. C*, 28(11):693–703, 1973.

[Hua06] Zheng Huang. Asymptotics of the Gaussian Curvatures of the Canonical Metric on the Surface. *ArXiv Mathematics e-prints*, 2006.

[JKG07] Miao Jin, Junho Kim, and Xianfeng Gu. Discrete surface ricci flow: Theory and applications. In *Proceedings of the 12th IMA International Conference on Mathematics of Surfaces*, 2007.

[KNPP02] George Kamberov, Peter Norman, Franz Pedit, and Ulrich Pinkall. *Quaternions, Spinors and Surfaces*. Amer. Math. Soc., 2002.

[KPP98] George Kamberov, Franz Pedit, and Ulrich Pinkall. Bonnet Pairs and Isothermic Surfaces. *Duke Math. J.*, 92(3):637–644, 1998.

[KSBC12] Michael Kazhdan, Jake Solomon, and Mirela Ben-Chen. Can Mean-Curvature Flow Be Made Non-Singular? *Comp. Graph. Forum*, 2012.

[LCOGL07] Yaron Lipman, Daniel Cohen-Or, Ran Gal, and David Levin. Volume and Shape Preservation via Moving Frame Manipulation. *ACM Trans. Graph.*, 26(1), 2007.

[LD11] Yaron Lipman and Ingrid Daubechies. Conformal wasserstein distances: Comparing surfaces in polynomial time. *Advances in Mathematics*, 227(3), 2011.

[LLCO08] Yaron Lipman, David Levin, and Daniel Cohen-Or. Green Coordinates. *ACM Trans. Graph.*, 27, 2008.

[LPRM02] Bruno Lévy, Sylvain Petitjean, Nicolas Ray, and Jérome Maillot. Least squares conformal maps for automatic texture atlas generation. *ACM Trans. Graph.*, 21(3):362–371, July 2002.

[LSLCO05] Yaron Lipman, Olga Sorkine, David Levin, and Daniel Cohen-Or. Linear Rotation-Invariant Coordinates for Meshes. *ACM Trans. Graph.*, 24(3):479–487, 2005.

[LSY97] Richard Lehoucq, Daniel Sorensen, and Chao Yang. ARPACK Users Guide: Solution of Large Scale Eigenvalue Problems by Implicitly Restarted Arnoldi Methods., 1997.

[Mac49] Richard MacNeal. *The Solution of Partial Differential Equations by means of Electrical Networks*. PhD thesis, Caltech, 1949.

[Mer01] Christian Mercat. Discrete Riemann Surfaces and the Ising Model. *Comm. Math. Physics*, 218(1):177–216, 2001.

[MTAD08] Patrick Mullen, Yiying Tong, Pierre Alliez, and Mathieu Desbrun. Spectral Conformal Parameterization. *Comp. Graph. Forum*, 27(5):1487–1494, 2008.

[OR09] Nadine Olischläger and Martin Rumpf. Two Step Time Discretization of Willmore Flow. In *Mathematics of Surfaces XIII*, volume 5654/2009 of *Lecture Notes in Computer Science*, pages 278–292. Springer, 2009.

[PAHK07] Helmut Pottmann, Andreas Asperl, Michael Hofer, and Axel Kilian. *Architectural Geometry*. Bentley Institute Press, 2007.

[PDK07] Nikolas Paries, Patrick Degener, and Reinhard Klein. Simple and Efficient Mesh Editing with Consistent Local Frames. In *Proc. Pac. Graph.*, Comp. Graph. Appl., pages 461–464, 2007.

[Pin] Ulrich Pinkall. personal communication.

[Pin85] Ulrich Pinkall. Hopf tori in s^3. *Inventiones Mathematicae*, 81, 1985.

[PKK00] Mark Pauly, Thomas Kollig, and Alexander Keller. Metropolis light transport for participating media. In *Proceedings of the Eurographics Workshop on Rendering Techniques*, 2000.

[PS87] Ulrich Pinkall and Ivan Sterling. Willmore Surfaces. *Math. Intell.*, 9(2):38–43, 1987.

[Ric97] Jörg Richter. *Conformal Maps of a Riemann Surface into the Space of Quaternions*. PhD thesis, TU Berlin, 1997.

[SA07] Olga Sorkine and Marc Alexa. As-rigid-as-possible surface modeling. In *Proc. Symp. Geom. Proc.*, pages 109–116, 2007.

[SK01] Robert Schneider and Leif Kobbelt. Geometric Fairing of Irregular Meshes for Free-From Surface Design. *Comput. Aided Geom. Des.*, 18(4):359–379, 2001.

[SSGH01] Pedro V. Sander, John Snyder, Steven J. Gortler, and Hugues Hoppe. Texture Mapping Progressive Meshes. In *Proc. ACM/SIGGRAPH*, pages 409–416, 2001.

[SSP08] Boris Springborn, Peter Schröder, and Ulrich Pinkall. Conformal Equivalence of Triangle Meshes. *ACM Trans. Graph.*, 27(3), 2008.

[Szm10] Radoslaw Szmytkowski. Recurrence and differential relations for spherical spinors. *ArXiv e-prints*, 2010.

[Tau95] Gabriel Taubin. A Signal Processing Approach to Fair Surface Design. In *Proc. ACM/SIGGRAPH*, pages 351–358, 1995.

[WBH+07] Max Wardetzky, Miklós Bergou, David Harmon, Denis Zorin, and Eitan Grinspun. Discrete Quadratic Curvature Energies. *Comput. Aided Geom. Des.*, 24(8-9):499–518, 2007.

[Wil65] T.J. Willmore. Note on embedded surfaces, 1965.

[WMZ12] Ofir Weber, Ashish Myles, and Denis Zorin. Computing extremal quasiconformal maps. *Comp. Graph. Forum*, 31(5), August 2012.

[WW94] William Welch and Andrew Witkin. Free-Form Shape Design Using Triangulated Surfaces. *Comp. Graph. (Proc. of ACM/SIGGRAPH Conf.)*, 28:247–256, 1994.

[YB02] Shin Yoshizawa and Alexander G. Belyaev. Fair Triangle Mesh Generation with Discrete Elastica. In *Geo. Mod. & Proc.*, pages 119–123, 2002.

[YKL+08] Yongliang Yang, Junho Kim, Feng Luo, Shimin Hu, and David Gu. Optimal Surface Parameterization Using Inverse Curvature Map. *IEEE Trans. Vis. Comp. Graph.*, 14(5):1054–1066, 2008.

[You10] Laurent Younes. *Shapes and Diffeomorphisms*. Springer, 2010.

[YZX+04] Yizhou Yu, Kun Zhou, Dong Xu, Xiaohan Shi, Hujun Bao, Baining Guo, and Heung-Yeung Shum. Mesh Editing with Poisson-based Gradient Field Manipulation. *ACM Trans. Graph.*, 23(3):644–651, 2004.

Nomenclature

A Dirac operator with potential, page 65

$\mathcal{A}$ total area of an immersed surface, page 10

X^* adjoint of matrix X, page 65

$\hat{X}$ pushforward of vector field X by df, page 9

A discrete Dirac operator with potential, page 65

B vertex-face averaging operator, page 65

$\overline{q}$ conjugate of q, page 19

$\mathbf{1}$ constant vector/function, page 8

∇ Levi-Civita connection on immersed surface, page 12

$\hat{\nabla}$ Levi-Civita connection on ambient space, page 12

$\times$ cross product on $\mathbb{R}^3$, page 8

X^{H} matrix conjugate transpose, page 63

curl curl operator, page 12

$\mathcal{D}$ quaternionic Dirac operator, page 25

$|df|^2$ volume form induced by a conformal immersion f, page 21

Δ Laplace-Beltrami operator, page 12

df differential of the immersion f, page 9

div divergence operator, page 12

$|df|$ length element induced by f, page 17

dN Weingarten map, page 10

D discrete Dirac operator, page 64

E_A membrane energy, page 86

E_C curve fairing energy, page 93

e_{i} edge opposite vertex i, page 64

e_{ij} edge from vertex i to vertex j, page 67

E_W Willmore energy, page 86

f immersion, page 9

$\frac{1}{|df|^2}$ Hodge star on 2-forms induced by f, page 21

G topological genus, page 60

g induced Riemannian metric, page 9

g_M canonical metric on M, page 59

grad gradient operator, page 12

H mean curvature, page 11

$\mathbb{H}$ quaternions, page 18

I first fundamental form, page 9

id identity map, page 8

II second fundamental form, page 10

$\mathrm{Im}(q)$ imaginary part of q, page 19

$\mathrm{Im}\mathbb{H}$ imaginary quaternions, page 18

$\langle\langle , \rangle\rangle$ $\mathscr{L}^2$ inner product, page 8

$\langle \cdot, \cdot \rangle$ Euclidean inner product, page 8

$\mathcal{J}$ complex structure, page 15

K Gaussian curvature, page 11

κ normal curvature, page 11

κ_1 maximum principal curvature, page 11

κ_2 minimum principal curvature, page 11

L total length of a curve, page 41

λ local similarity transformation of differential, page 44

M domain of immersion, page 9

M_F triangle mass matrix, page 65

μ mean curvature half-density, page 17

M_V vertex mass matrix, page 65

N unit normal, page 10

$\| \cdot \|$ $\mathscr{L}^2$ norm, page 8

$| \cdot |$ Euclidean norm, page 8

ν unit normal on conformal shape space, page 51

∂M boundary of M, page 9

P diagonal matrix of ρ values, page 65

$\mathcal{Q}$ quasi-conformal error, page 69

$\mathrm{Re}(q)$ real part of q, page 19

ρ curvature potential, page 48

R discrete curvature potential, page 65

S shape operator, page 10

$\mathcal{S}$ shape operator on shape space, page 52

σ volume form induced by a general immersion, page 10

T unit tangent, page 41

τ time step, page 90

TM tangent bundle of M, page 9

$\top$ tangent part of an immersed vector field, page 10

X^{T} matrix transpose, page 63

υ harmonic vector field on $f(M)$, page 55

φ' spatial derivative of the quantity φ, page 8

V square root of vertex mass matrix, page 66

v_i vertex i, page 64

$\wedge$ wedge product of differential 1-forms, page 21

X_1 maximum principal direction, page 11

X_2 minimum principal direction, page 11

Z harmonic constraint function, page 55

$\mathcal{A}_i$ triangle area, page 64

t_i triangle i, page 64