

Discrete Laplacians on General Polygonal Meshes

Marc Alexa*
TU Berlin

Max Wardetzky†
Universität Göttingen

Abstract

While the theory and applications of discrete Laplacians on *triangulated* surfaces are well developed, far less is known about the general *polygonal* case. We present here a principled approach for constructing geometric discrete Laplacians on surfaces with arbitrary polygonal faces, encompassing non-planar and non-convex polygons. Our construction is guided by closely mimicking structural properties of the smooth Laplace–Beltrami operator. Among other features, our construction leads to an extension of the widely employed cotan formula from triangles to polygons. Besides carefully laying out theoretical aspects, we demonstrate the versatility of our approach for a variety of geometry processing applications, embarking on situations that would have been more difficult to achieve based on geometric Laplacians for simplicial meshes or purely combinatorial Laplacians for general meshes.

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Curve, surface, solid, and object representations, Geometric algorithms, languages, and systems.

Keywords: Discrete Laplace operator, generalized cotan formula, geometry processing with polygonal meshes.

1 Motivation

Among the geometric *atomic building blocks* of graphics, *triangles* have by far attracted the most attention—perhaps because triangles are the simplest geometric figures that are able to represent two-dimensional shapes, or perhaps even due to Plato’s foreshadowing work *Timaeus*, where he records that “every solid must necessarily be contained in planes; and every planar rectilinear figure is composed of triangles”. Yet, as favorable as the simplicity of triangles might appear from a purist’s perspective, exclusively restricting to triangles largely impedes artistic freedom. Ornaments, tilings, kaleidoscope pattern, cubist art, architecture, or design would be paltry without quadrilaterals, pentagons, hexagons, and in fact arbitrary polygonal shapes.

Likewise, in geometry processing, consider, for example, the clipping, trimming, and intersection of meshes, the insertion of meshes into spatial data structures, such as *kd*- or *BSP*-trees, or the reconstruction of meshes using marching cubes or Voronoi tessellations. Consider modeling and animating meshes with mixed quad-triangle control nets, such as they commonly appear in geometric design. Or consider the isolines of a surface parameterization, nets of curvature

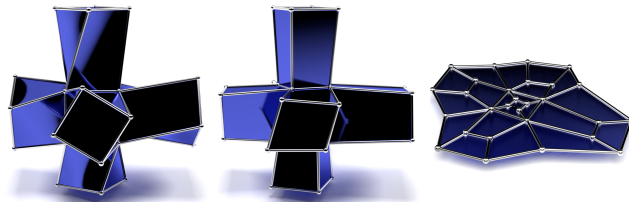


Figure 1: The non-planar faces of a polygonal mesh (left) can be planarized (middle) using a gradient flow of an energy that is based on our Laplacian. Likewise, the mesh can be conformally mapped to the plane with automatic boundary placement (right).

lines, or Morse complexes on surfaces. All of these innately lead to tessellations by general, non-triangular polygons.

Scrutinizing a parallel development, it seems fair to point out that the majority of contemporary geometry processing tools rely on, if only in the background, *discrete Laplace operators*, with the cotan operator perhaps being the most prominent representative. The use of discrete Laplacians spans mesh parameterization, fairing, denoising, manipulation, compression, shape analysis, and physical simulation. Accordingly, the theory of discrete Laplacians on triangle meshes is a far developed field. To date, this theory has not been extended to discrete surfaces with general *polygonal* faces.

Our work sets forth the missing development of discrete Laplacians on surfaces with arbitrary (including non-planar and non-convex) polygonal faces. We present a treatment that aims at maintaining core properties of Laplace–Beltrami operators on smooth Riemannian surfaces. While our approach initially requires some amount of theoretical development, the actual *implementation* is surprisingly simple. In principle, the tools that we develop here seamlessly allow for extending geometry processing and physical simulation applications that are based upon the cotan operator from the triangular to the general polygonal setting. Our approach, therefore, expands the artist’s creative freedom and takes a step toward facilitating the geometry pipeline, *without* sacrificing mathematical methodology.

2 Related Work

It would be impossible to do justice to the numerous publications in geometry processing and physical simulation relating to the *applications* of the cotan formula and its variants, as it would amount to citing several dozens of relevant works. Instead, we focus here on those developments that are concerned with the *theory* of the cotan Laplacian and that are relevant for our treatment.

Geometric discrete Laplacians For surfaces with triangular faces, the so-called *cotan formula* is attributed to [Pinkall and Polthier 1993], where the relation to discrete mean curvature and minimal surfaces was first brought to light. Earlier, Dziuk [1988] had presented a Finite Element approach that is equivalent to the cotan formula. It was later found that Duffin [1959] had already explicitly worked with the cotan representation.

The last decade or so has brought forward several parallel developments extending the cotan formula. By concurrently considering a

*e-mail: marc.alexa@tu-berlin.de

†e-mail: wardetzky@math.uni-goettingen.de

primal mesh together with a perpendicular dual, Discrete Exterior Calculus (DEC) [Desbrun et al. 2005] arrives at the cotan Laplacian via a discrete Hodge star operator. The attendant (dual) area weighting had previously been considered in [Meyer et al. 2003] but was later found to be problematic in some degenerate cases [Dyer and Schaefer 2009]. Gu and Yau [2003] constructed a discrete cotan-based Hodge star without explicitly referring to dual meshes. Using the cotan formula, Xiong et al. [2011] average locally over all possible triangulations to obtain a discrete Laplacian for quad meshes.

The primal-dual approach is also central to Mercat’s construction [2001] of a discrete Hodge star operator for studying discrete harmonic and holomorphic 1-forms. Bobenko and Springborn [2007] have extended the Voronoi–Delaunay duality to polyhedral surfaces, giving rise to an intrinsic cotan Laplacian with non-negative edge weights. Glickenstein [2007] has extended the cotan formula by considering weighted triangulations, leading to a generalization of Rippa’s theorem [1990] on the monotonicity of Delaunay edge flips. Recently, Chen et al. [2010] have found an elegantly short proof of Rippa’s theorem (in its original, unweighted formulation) based on a spectral approach.

Hildebrandt et al. [2006] have established convergence of the cotan formula in a *weak* sense, under some general, mild conditions on the shape of triangles. Xu [2004] showed pointwise convergence under special conditions (such as on the valence of vertices). In a recent development, based on discretizing the heat flow on triangle meshes, the approaches in [Belkin et al. 2008; Dey et al. 2010] construct discrete Laplacians that converge in a pointwise manner.

Mimetic Finite Differences & our approach In a series of articles based on the so-called MFD methodology, Brezzi and coworkers [2005a; 2005b] constructed numerical schemes for solving diffusion problems on *planar* domains subdivided into arbitrary *planar* polygons as well as spatial domains subdivided into arbitrary polyhedral elements. In an independent development, Perot and Subramanian [2007] considered discrete diffusion problems based on a mixed formulation, where the divergence of flux is assumed to be constant per polygonal element. Although these works are *not* directly concerned with the construction of discrete Laplacians, they have provided tremendous inspiration for our work. Indeed, on an algebraic level, the linear systems occurring in these works are similar to those considered in our exposition—in particular, in terms of the *barycentric (median) duals* constructed therein. To stress similarities, we have tried to adopt the notation of Brezzi et al. where possible. It bears noting that unlike Brezzi et al., Perot and Subramanian do not discuss the failure of positive definiteness of the resulting linear systems for non-triangular meshes.

What sets our approach apart from these works are three aspects: (i) We extend these approaches to two dimensional *manifolds* that can be comprised of general *non-planar* polygonal faces. (ii) We offer an independent derivation, based on *geometric* considerations that closely mimic the derivation of Laplace–Beltrami operators in the smooth case and the cotan Laplacian for triangular meshes. (iii) We treat applications that go beyond diffusion problems, such as conformal mesh parameterizations. Nonetheless, we certainly share with Brezzi et al. the observation that the resulting implementation is *inexpensive and easy-to-code*.

3 Discrete Laplacian Framework

Our setup is an *oriented* 2-manifold mesh M , possibly with boundary, with vertex set V , edge set E , and face set F . We allow for faces that are *simple*, but possibly non-planar, polygons in \mathbb{R}^3 . In our setup, faces are solely determined by their boundary edge loops. By *simple* we mean that each polygon forms a closed non self-

intersecting loop (which could be knotted). By *oriented* we mean that all faces are equipped with a consistent orientation in such a way that any two adjacent faces induce opposite orientations on their common edge. In order to distinguish between the resulting two orientations of each inner edge, we work with oriented half-edges. We distinguish between the set E_I of inner edges and the set E_B of boundary edges.

Algebraic approach to discrete Laplacians Our treatment of discrete Laplacians relies on an algebraic approach. It offers the advantages of (i) a unified treatment and (ii) a surprisingly simple implementation for Laplacians on meshes with polygonal faces of arbitrary degree, see Section 4. These advantages seem unique to our algebraic treatment. For example, the alternative of basing the construction on polynomial basis functions is not immediately viable: closed form expressions for the Dirichlet energy are cumbersome already for the simplest case of bilinear basis functions for convex planar quads [Shafiqul-Islam and Rathod 2006] and seem to be unavailable for general quads [Liu et al. 2008].

In the following we lay out our algebraic framework, deferring all proofs to the appendix. In analogy to standard notation in the smooth setting, we let Ω^k , $k \in \{0, 1\}$, be the linear space of discrete k -forms on M . In the discrete case 0-forms are real values associated with vertices and 1-forms are real values associated with half-edges. We require that $\alpha(e_{pq}) = -\alpha(e_{qp})$ for any 1-form α and an oriented half-edge e_{pq} from vertex p to q . In order to define discrete Laplacians acting on 0-forms (functions), we build on the smooth setting, where

$$\Delta = d^*d. \quad (1)$$

Here $d : \Omega^0 \rightarrow \Omega^1$ denotes Cartan’s exterior derivative and d^* is its (formal) adjoint with respect to the L^2 inner products induced on Ω^0 and Ω^1 by the Riemannian metric [Rosenberg 1997]. Notice that d is independent of the choice of a Riemannian metric.

In order to mimic the smooth setting, we consider the so-called coboundary operator $d : \Omega^0 \rightarrow \Omega^1$, defined as

$$(du)(e_{pq}) = u(q) - u(p), \quad (2)$$

where e_{pq} denotes the oriented half-edge from p to q . Analogous to the smooth case, d is independent of the choice of any metric.

In order to construct the adjoint operator d^* , we require inner products on the spaces of 0- and 1-forms, which we represent by a symmetric positive definite $|V| \times |V|$ matrix M_0 and a symmetric positive definite $(2|E_I| + |E_B|) \times (2|E_I| + |E_B|)$ matrix M_1 , respectively. Any concrete choice of inner products M_0 and M_1 then yields a discrete Laplacian of the form

$$\mathbb{L} = d^*d = M_0^{-1}L \quad \text{with} \quad L = d^T M_1 d. \quad (3)$$

For example, for meshes with triangular faces, one possible choice of inner products arises from using Finite Elements. Moreover, below we show that L becomes the *cotan* Laplacian for a natural choice of M_1 .

Dirichlet energy It bears pointing out the conceptual difference between \mathbb{L} and L . In the language of partial differential equations, \mathbb{L} corresponds to the *strong* (or pointwise) formulation of the Laplacian, whereas L represents the *weak* (or integrated) version. L gives rise to the *Dirichlet energy* of a function u defined via

$$E_D(u) = \frac{1}{2} u^T L u. \quad (4)$$

Observe that L is a symmetric matrix with respect to the standard inner product and, *equivalently*, \mathbb{L} is symmetric with respect to the inner product induced by M_0 .

Notation Throughout, we use the following notation. We denote 3-vectors by bold face letters, while we denote all matrices (except for d) by upper case letters. Furthermore, if f is a simple, possibly non-planar polygon with k_f vertices in \mathbb{R}^3 , then

- $X_f = (\mathbf{x}_1^f, \dots, \mathbf{x}_{k_f}^f)^T$ denotes the $k_f \times 3$ matrix of cyclically ordered vertices along the boundary ∂f ,
- $E_f = (\mathbf{e}_1^f, \dots, \mathbf{e}_{k_f}^f)^T$ denotes the $k_f \times 3$ matrix of oriented and cyclically ordered (half)edges along the boundary ∂f ,
- $B_f = (\mathbf{b}_1^f, \dots, \mathbf{b}_{k_f}^f)^T$ denotes the $k_f \times 3$ matrix of barycenters (midpoint positions) of each \mathbf{e}_i^f .

With these notations at hand, we can state our main objectives for discrete Laplacians.

3.1 Desiderata

While any choice of inner product matrices M_0 and M_1 on 0-forms and 1-forms, respectively, would in principle be conceivable, *not all choices are created equal* in terms of resembling and maintaining core structural properties of the smooth setting. In order to narrow the set down to *good* inner products, we are guided by requiring a number of important properties that hold for (and to some extent characterize) smooth Laplacians. Most of these properties were suggested in [Wardetzky et al. 2007b] for meshes with triangular faces. Here we offer an extension and adaptation for meshes whose faces are general polygons.

LOCALITY. The smooth Laplacian is a differential operator; therefore, its definition is *local*, i.e., it does not depend on properties of the underlying Riemannian manifold outside an arbitrarily small open neighborhood of its point of definition. In the discrete case, we maintain locality by only working with diagonal matrices M_0 and by requiring that M_1 is defined *per face* in the sense that

$$\alpha^T M_1 \beta = \sum_{f \in F} \alpha_{|f}^T M_f \beta_{|f} \quad (5)$$

for any pair (α, β) of discrete 1-forms. The sum runs over all faces and $\alpha_{|f}$ and $\beta_{|f}$, respectively, denote the restrictions of α and β to the boundary ∂f of f . For each $f \in F$, we work with a symmetric $k_f \times k_f$ matrix M_f . From an implementation point of view, locality corresponds to the desirable property of *sparsity*.

SYMMETRY. The symmetry of M_f (and thus of M_1) reflects the fact that on a Riemannian manifold without boundary, the smooth Laplacian is self-adjoint with respect to the L^2 inner product on 0-forms (functions). In the discrete case this translates to \mathbb{L} being self-adjoint with respect to the inner product structure defined by M_0 or, equivalently, to $L = L^T$.

POSITIVE SEMI-DEFINITENESS. The smooth Laplacian on a Riemannian manifold without boundary is positive semi-definite with one-dimensional kernel equal to the constants—a property that guarantees existence and uniqueness of the solutions to various variational problems. Likewise, in the discrete setting we require that M_0 and each M_f (and therefore M_1) is positive definite. The kernels of \mathbb{L} and L are then automatically one-dimensional since d has a one-dimensional kernel given by the constants.

LINEAR PRECISION. In the smooth case, if $M \subset \mathbb{R}^2$ is a planar domain and $u : \mathbb{R}^2 \rightarrow \mathbb{R}$ is a linear function, then $\Delta u = 0$. In the discrete case we require that if all vertices lie in a single plane then $(\mathbb{L}u)_p = 0$ at each interior vertex p and for each linear function u . In applications, this property is important for, e.g., mesh parameterization, where an already planar mesh should remain unaltered. Linear precision can be rewritten in a geometric way, which

requires some additional notation. Let

$$\star E_f := M_f E_f \quad (6)$$

be a $k_f \times 3$ matrix whose rows, $\star \mathbf{e}_i^f$, we refer to as *dual edges*. Consider an arbitrary but fixed vertex $p \in V$ and let $f \in F$ be an incident face to p . Let i and j be the row indices of E_f , respectively corresponding to the unique (half)edge \mathbf{e}_i^f of f pointing away from p and the unique (half)edge \mathbf{e}_j^f of f pointing towards p . Then, whenever M is a mesh in the plane, linear precision is *equivalent* to the *integrability condition*

$$0 = \sum_{f \ni p} (\star \mathbf{e}_j^f - \star \mathbf{e}_i^f) \quad (7)$$

where the sum runs over all faces f incident to p and the subscripts, i and j , may vary from face to face. Eq. (7) says that the (appropriately oriented) dual edges around each inner vertex form a *closed* polygon. To see the equivalence to linear precision, notice that the condition $(Lu)_p = 0$ for every linear function u and every inner vertex p is equivalent to $(LX)_p = 0$, where X is the $|V| \times 3$ matrix that encodes all vertex positions. By the definition of L , the last condition is equivalent to $(d\phi_p)^T M_f (dX) = 0$, where $\phi_p(q) = 1$ if $p = q$ and $\phi_p(q) = 0$ otherwise. It is then straightforward to check (7) by using (6) and noting that $dX_f = E_f$.

SCALE INVARIANCE. In dimension two, the (smooth) Dirichlet energy is a *conformal invariant*, i.e., it remains unchanged under conformal changes $g \mapsto \lambda g$ of the Riemannian metric g . In particular, Dirichlet energy is invariant under uniformly rescaling a smooth embedded surface. Therefore, we require that the weak discrete Laplacian, L , remains unchanged under uniformly rescaling a mesh, i.e., we require that M_f is invariant under uniform scaling.

CONVERGENCE. In order to obtain well-defined limits under mesh refinement, we require that $\mathbb{L}_n \rightarrow \Delta$ under appropriate refinement operations and with respect to appropriate norms. While we do not provide theoretical convergence results here, our experiments indicate good numerical convergence behavior of our construction.

It follows from the previous discussion that we can indeed satisfy *all* of the above desiderata—possibly apart from convergence—*provided that* we are able to construct, separately for each simple polygon f , a symmetric positive definite and scale invariant matrix M_f such that, using (6) as a definition for $\star E_f$, identity (7) is fulfilled at each inner vertex of a planar mesh.

Before describing the construction of a family of discrete Laplacians that satisfy our desiderata, we recall the notion of *vector area*, which extends the concept of triangle area to polygons.

3.2 Vector area and maximal projection

The *vector area*, $\mathbf{A}(\gamma)$, of a simple closed curve $\gamma \subset \mathbb{R}^3$ is given by the surface integral of the normal vector of a (sufficiently regular) surface with boundary γ . $\mathbf{A}(\gamma)$ only depends on the boundary curve, *not* on the choice of a particular surface spanning this curve:

$$\mathbf{A}(\gamma) = \frac{1}{2} \oint_{\gamma} \mathbf{x} \times d\mathbf{x} \quad ,$$

where \mathbf{x} denotes the position vector of γ , see [Sullivan 2008]. For polygonal curves we arrive at the following representation.

Lemma 1. *For a simple, possibly non-planar, polygon f in \mathbb{R}^3 ,*

$$A_f := E_f^T B_f \quad (8)$$

is a 3×3 skew-symmetric matrix of (maximal) rank 2 whose Darboux vector, $[A_f]$, is equal to the vector area of f .

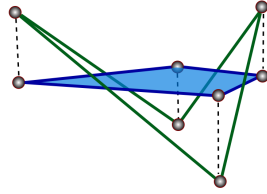
Recall that the *Darboux vector* of a skew-symmetric 3×3 matrix A is the unique vector $[A] \in \mathbb{R}^3$ satisfying $[A] \times \mathbf{x} = A\mathbf{x}$ for all $\mathbf{x} \in \mathbb{R}^3$. Observe further that the vector area changes sign when we change the orientation in which the boundary of a face is traversed.

The magnitude, $|f|$, of the vector area of a polygon f is the largest (signed) area over all orthogonal projections of f to planes in \mathbb{R}^3 . Accordingly, we call a planar polygon \bar{f} a *maximal projection* of a (possibly non-planar) polygon f if it is an orthogonal projection that has the *same* vector area as f . Note that if f is planar then $f = \bar{f}$. Vice-versa, consider a planar polygon \bar{f} with vertex set $\{\bar{\mathbf{x}}_1, \dots, \bar{\mathbf{x}}_k\}$ and face normal $\bar{\mathbf{n}}$. We say that a polygon f with vertex set $\{\mathbf{x}_1, \dots, \mathbf{x}_k\}$, $\mathbf{x}_i = \bar{\mathbf{x}}_i + h_i \bar{\mathbf{n}}$, has *height vector* $h = (h_1, \dots, h_k)^T \in \mathbb{R}^k$ over \bar{f} . Given a planar polygon \bar{f} , what are all possible height vectors such that \bar{f} is a maximal projection of f ? The following lemma provides a complete answer.

Lemma 2. *An orthogonal projection of a polygon f to a planar polygon \bar{f} is maximal if and only if the midpoint height vector, $h_b = \frac{1}{2}(h_1 + h_2, \dots, h_k + h_1)^T$, is in the kernel of $E_{\bar{f}}^T$.*

In particular, it follows from this lemma that if k is even, then any polygon f with height vector $(h_1, h_2, \dots, h_1, h_2)^T$ over a planar polygon \bar{f} has \bar{f} as its maximal projection.

Example Notice that the edge midpoints of any (possibly non-planar) quadrilateral f are always coplanar. Moreover, the plane spanned by these midpoints is the plane of the maximal projection of f . The incident figure provides an example of this fact, with the spatial polygon colored in green and its maximal projection depicted in blue.



Mean curvature The intrinsic Laplace–Beltrami operator of an embedded surface is intimately connected to mean curvature. Indeed, for smoothly embedded surfaces, the mean curvature *vector* is given by applying the Laplace–Beltrami operator to each component of the 3-valued surface positions, i.e., $\mathbb{H} = \Delta \mathbf{x}$. In the discrete case this translates to the (pointwise) mean curvature vector

$$\mathbb{H} = \mathbb{L}X,$$

i.e., \mathbb{H} is a $|V| \times 3$ matrix, whose rows correspond to the mean curvature 3-vectors that are associated with the vertices of the polygonal mesh. Moreover, in the smooth case, the mean curvature vector is the L^2 -gradient of the area functional, i.e., its negative determines the direction and speed of the surface flow that maximally decreases surface area. In the discrete case an analogous property holds for the cotan formula: As originally formulated, the cotan formula was derived precisely as the gradient of surface area of a triangulated surface mesh. Extending this approach to arbitrary polygons, we replace triangular area by polygonal *vector area* to arrive at the following result.

Lemma 3. *Let f be a (possibly non-planar) polygon in \mathbb{R}^3 with vector area of magnitude $|f|$. Then the gradient of $|f|$ with respect to varying a vertex \mathbf{x}_i of f satisfies*

$$\nabla_{\mathbf{x}_i} |f| = \left(\tilde{L}_f X_f \right)_i,$$

where $\tilde{L}_f := d^T \tilde{M}_f d$ and the symmetric $k_f \times k_f$ matrix \tilde{M}_f is

given by

$$\tilde{M}_f := \frac{1}{|f|} B_f B_f^T.$$

As before, B_f denotes the matrix of edge midpoint vectors.

We hasten to point out that, as given, our definition of \tilde{M}_f is not independent of the choice of origin of a coordinate system. Fortunately, for mean curvature and Laplacian computations, the choice of origin becomes irrelevant due to the presence of d . For implementation purposes, in order to obtain a numerically stable representation, we set the origin to the center of mass of the vertices spanning f , i.e., $0 = \sum_{i=1}^{k_f} \mathbf{x}_i$. However, unless explicitly stated otherwise, *all* of the following objects and results are independent of the choice of origin.

3.3 A family of discrete Laplacians

We are now in the position to outline a whole family of discrete Laplacians that satisfy our desiderata.

Inner products on 1-forms As a point of departure for constructing positive definite inner products M_f that satisfy our desiderata, we consider the matrices \tilde{M}_f constructed above.

Lemma 4. *The matrices \tilde{M}_f give rise to positive semi-definite inner products on 1-forms and yield pre-Laplacians $\tilde{L}_f = d^T \tilde{M}_f d$ that are local, linearly precise, and scale invariant.*

We note that the matrices \tilde{M}_f also appear (under different names) in the constructions of [Brezzi et al. 2005b; Perot and Subramanian 2007]. However, the properties of \tilde{M}_f do not suffice for our purposes. Indeed, while the matrices \tilde{M}_f are positive semi-definite by construction, they are unfortunately *not* (in general) positive definite, as would be required by our desiderata. In order to overcome this deficiency *without* giving up linear precision, we extend the construction proposed in [Brezzi et al. 2005b] from the case of planar to general non-planar polygons.

First notice that E_f has a non-trivial kernel if and only if f is planar. In particular, the Darboux vector $[A_f]$ is in the kernel of E_f if and only if f is planar. Hence, E_f^T has rank 2 if f is planar; otherwise, it has maximal rank 3.

Now let $C_{\bar{f}}$ be a $k_f \times (k_f - 2)$ matrix whose columns span the null space of $E_{\bar{f}}^T$, where \bar{f} is the maximal projection of f . We call any such matrix $C_{\bar{f}}$ *admissible*. Likewise, we call any $(k_f - 2) \times (k_f - 2)$ matrix $U_{\bar{f}}$ *admissible* if it is symmetric and positive definite. With these notations at hand, we obtain:

Theorem 1. *Let $C_{\bar{f}}$ and $U_{\bar{f}}$ be admissible. Then the matrices*

$$M_f := \tilde{M}_f + C_{\bar{f}} U_{\bar{f}} C_{\bar{f}}^T \quad (9)$$

give rise to positive definite inner products on 1-forms and yield Laplacians $L_f := d^T M_f d$ that are local and linearly precise.

Note that scale invariance is absent from the above list of properties. Indeed, scale invariance is an *extra* condition that needs to be imposed by the choice of admissible matrices $C_{\bar{f}}$ and $U_{\bar{f}}$.

Choosing C and U While various admissible choices would in principle be possible and lead to scale invariance, we restrict our attention to λ -simple choices: We let $U_{\bar{f}} := \lambda Id$ for some

$0 < \lambda \in \mathbb{R}$, and we let $C_{\bar{f}}$ be such that its columns are *orthonormal*. Clearly, our second choice does *not* uniquely pin down the matrices $C_{\bar{f}}$. However, it is straightforward to see that (together with our choice of $U_{\bar{f}}$) the expression $C_{\bar{f}}U_{\bar{f}}C_{\bar{f}}^T$ remains *invariant* under choosing different orthonormal bases for representing $C_{\bar{f}}$. Leaving a closer investigation of a broader range of choices of $C_{\bar{f}}$ and $U_{\bar{f}}$ for future work, we point out that our choice clearly leads to discrete Laplacians that are scale invariant.

In particular, our construction provides a generalization of the widely-employed cotan formula:

Theorem 2. *For triangle meshes, any admissible choices of $C_{\bar{f}}$ and $U_{\bar{f}}$ (λ -simple or not) lead to the cotan Laplace operator via $L = d^T M_1 d$.*

In order to finish our construction of discrete Laplacians, we are left with defining a positive inner product on 0-forms.

Inner products on 0-forms Due to our requirement of locality, we restrict our construction to diagonal matrices M_0 . We define

$$(M_0)_{pp} := \sum_{f \ni p} \frac{|f|}{k_f}, \quad (10)$$

i.e., each face f incident to a vertex p contributes $1/k_f$ of (the norm of) its vector area to the total mass of p .

Summary Summing up our construction, we work with positive inner products on 1-forms as in Theorem 1 for individual polygons, restricting to λ -simple choices for constructing U and C . We then use (5) in order to construct the inner product matrix M_1 on the entire polygonal mesh, and we work with the inner products M_0 on 0-forms described in the previous paragraph. With these tools at hand, the above results imply that the Laplacians \mathbb{L} and L defined in (3) are local, symmetric, positive semi-definite with one dimensional kernel, and linearly precise, while L is additionally scale invariant—in perfect analogy to the smooth setting.

3.4 Discussion

We note that our discrete Laplacians are comprised of two parts: a geometric and a combinatorial one. Indeed, for individual polygons, the matrix \tilde{M}_f encodes the polygon’s geometry, yielding, e.g., the *exact* Dirichlet energy for linear functions over planar polygons. The remaining part, $C_{\bar{f}}U_{\bar{f}}C_{\bar{f}}^T$, which guarantees that M_f becomes positive definite, can be interpreted as *combinatorial*. We leave the investigation of how to replace this combinatorial term by a more geometric one for future work. Below we discuss the different roles played by both parts for mean curvature computations.

Mean curvature revisited Given a polygonal face f , we define the local (integrated) mean curvature vector H_f (written as a $k_f \times 3$ matrix, with rows indexed by vertices) as

$$H_f = L_f X_f, \quad \text{and we let} \quad \tilde{H}_f = \tilde{L}_f X_f.$$

According to the discussion in Section 3.2, we use the notation *mean curvature vector* in equivalence to *area gradient* here. Indeed, by Lemma 3, \tilde{H}_f corresponds to the gradient of vector area of f . In order to better grasp the influence of the combinatorial part of our Laplacians, and in particular the influence of choosing λ , it thus remains to study the difference $H_f - \tilde{H}_f$.

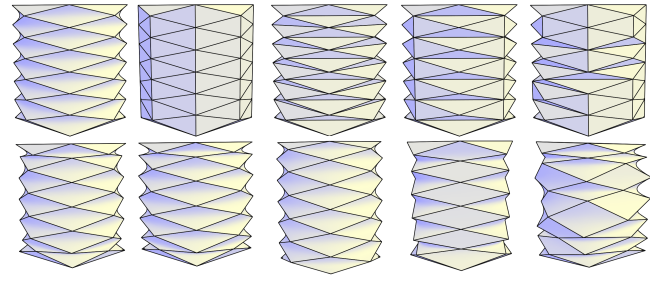


Figure 2: Different triangulations of the Schwarz lantern lead to drastically different results for mean curvature flow. Upper row: Original quadrangulation (left) and various triangulations of the exact same vertex set. Lower row: Results of mean curvature flow applied to the respective upper tessellations (with the same number of steps for each case), followed by removing the extra edges of the respective triangulation.

Lemma 5. *Let f be a non-planar polygon and \bar{f} be its maximal projection with height vector h_f . Then $\tilde{H}_f = H_{\bar{f}}$. Moreover, for any λ -simple choice of $C_{\bar{f}}$ and $U_{\bar{f}}$ we have*

$$H_f = H_{\bar{f}} + \lambda \left(d^T \bar{h}_e \right) \bar{\mathbf{n}}^T,$$

where \bar{h}_e is the orthogonal projection of $h_e = dh_f$ to the kernel of $E_{\bar{f}}^T$ and $\bar{\mathbf{n}}$ is the unit normal of \bar{f} .

One essential part (the *planar component*) of the mean curvature vector of a non-planar polygon f is thus already uniquely encoded by the mean curvature vector of its maximal projection \bar{f} . Moreover, the mean curvature vectors of f and \bar{f} differ by a term along the normal direction (with respect to \bar{f}) that incorporates the non-planarity of f in terms of the *combinatorial* second derivative $d^T \bar{h}_e$ of the height vector h_f . The choice of λ therefore determines how much we penalize deviation from planarity.

Relation to DEC Within the framework of DEC, Laplacians arise out of the construction of a discrete Hodge star operator, using that the Laplace–Beltrami operator on a Riemannian 2-manifold satisfies $\Delta = - * d * d$. Using *diagonal* Hodge star operators (as in DEC) or inner products on k -forms (as we do here) turns out to be equivalent for triangle meshes. For general polygonal meshes, however, our use of M_1 does in general no longer correspond to a *diagonal* Hodge star.

4 Implementation

Our algebraic approach allows for a simple and straightforward implementation. Even for the triangular case, we found this implementation to be somewhat simpler than the usual cotan approach. Our approach only requires the construction of three matrices: d , M_1 , and M_0 , with our discrete Laplacians then assembled using (3). Each of these three matrices is sparse and needs to be stored and multiplied efficiently—we use *Eigen* (see <http://eigen.tuxfamily.org/>) for this purpose.

The construction of M_0 is straightforward and the computation of the diagonal entries is given by (10). The coboundary matrix, d , as well as the inner product on 1-forms, M_1 , require an identification of half-edges with indices.

The entries of the $(2|E_I| + |E_B|) \times |V|$ coboundary matrix, d , are given by $d_{ep} = \pm 1$ if $e = \pm e_{qp}$ and $d_{ep} = 0$ otherwise. The values of M_1 is assembled per face: we compute the matrices M_f

and store the result in the respective entries of M_1 , following the above mentioned indexing scheme. The computation of each M_f follows the discussion in the preceding section and is summarized in the following pseudo-code.

```

COMPUTE  $M_f(\text{polygonal face } f, \text{parameter } \lambda)$ 
1   $B, E, \bar{E} \in \mathbb{R}^{\text{degree}(f) \times 3}$ 
2  for each vertex coordinate  $\mathbf{x}_i$  in  $f$ 
3       $E(i) = (\mathbf{x}_{i+1} - \mathbf{x}_i)^T$ 
4       $B(i) = \frac{1}{2}(\mathbf{x}_{i+1} + \mathbf{x}_i)^T$ 
5   $A = E^T B$ 
6   $\tilde{M} = \frac{\sqrt{2}}{\|A\|} B B^T$ 
7   $\bar{\mathbf{n}} = \text{normalize}(-A_{23}, A_{13}, -A_{12})^T$ 
8  for each vertex coordinate  $\mathbf{x}_i$  in  $f$ 
9       $\bar{\mathbf{x}}_i = \mathbf{x}_i - (\mathbf{x}_i \cdot \bar{\mathbf{n}}) \bar{\mathbf{n}}$ 
10 for each vertex coordinate  $\mathbf{x}_i$  in  $f$ 
11      $\bar{E}(i) = (\bar{\mathbf{x}}_{i+1} - \bar{\mathbf{x}}_i)^T$ 
12  $C = \text{orthonormal kernel of } \bar{E}^T$  (e.g., using LU, then SVD)
13  $U = \lambda \text{Id}$ 
14  $M_f = \tilde{M} + C U C^T$ 

```

5 Results & Applications

In principle, all geometry processing operations relying on geometric Laplacians are now applicable to polygonal meshes, simply by using the operator that we have developed in the preceding section. We demonstrate this fact for representative applications, such as fairing, parameterizing, and bending a surface mesh.

Unless stated otherwise, in the examples below we use $\lambda = 2$. In fact, we found that choosing $1 \leq \lambda \leq 3$ gives comparable results.

5.1 Implicit mean curvature flow

A common approach to the fairing of triangle meshes is provided by using a discrete mean curvature flow, where each vertex flows with speed and direction given by the (negative) pointwise mean curvature vector. Desbrun et al. [1999] have provided a stable and efficient implementation using an implicit time stepping scheme. This approach can seamlessly be carried over to the polygonal case by considering the flow

$$\frac{d}{dt} X = -\mathbb{L}X.$$

Figure 2 shows an example of this flow. Had we insisted on working with a triangulation of the polygonal mesh, observe the attendant strong dependence of the result on the choice of the triangulation.

5.2 Parameterization

For a polygonal surface mesh, we distinguish between interior (X_I) and boundary (X_B) vertices. For triangle meshes, various approaches to parameterization are based on the idea that the target boundary vertices Y_B are prescribed (i.e., $X_B \mapsto Y_B$), followed by solving an energy minimization problem in order to determine the positions of the target in-plane interior vertices Y_I . Sometimes it suffices (or is even natural) to prescribe part of the boundary only. *Least square conformal maps* (LSCM) [Lévy et al. 2002] and *discrete conformal parameterizations* (DCP) [Desbrun et al. 2002] are two prominent (in fact equivalent) examples of this paradigm. We next describe an extension of LSCM/DCP to polygonal meshes.

Conformal energy minimization Both for the case of smooth surfaces and for the case of triangle meshes, the expression

$$E_C(Y) = E_D(Y) - \mathcal{A}(Y) \quad (11)$$

constitutes a well-known way of formulating the *conformal energy* of a parameterization, where $\mathcal{A}(Y)$ is the area of the planar image and $E_D(Y)$ is the Dirichlet energy of the parameterization. E_C provides a measure of the deviation from the parameterization to be conformal, i.e., angle preserving. In the polygonal case, both \mathcal{A} and E_D have a well-defined meaning; thus, taking (11) as a *definition* yields a notion of conformal energy for polygonal meshes.

Working with this definition, it is straightforward to implement extensions of LSCM and DCP to the polygonal case. In order to do so, we require the gradient of E_C with respect to variations of target vertices. By (11), this gradient is the difference between the gradients of E_D and \mathcal{A} . The area gradient of a single *planar* polygon f with respect to variations of the position \mathbf{x}_i of its i th vertex can be computed explicitly by considering the triangle formed by the i th vertex and its two direct neighbors. Using the formula for area gradients of triangles (see [Desbrun et al. 2002]), we arrive at the following identity:

$$(L_f X_f)_i^T = \frac{1}{2} R^{90} (\mathbf{x}_{i+1} - \mathbf{x}_{i-1}), \quad (12)$$

where $(L_f X_f)_i$ denotes the i th row of the $k_f \times 2$ matrix $L_f X_f$ and R^{90} stands for the in-plane rotation by $\pi/2$.

The critical points of E_C are then essentially characterized by the linear system (12). To be precise, using Y instead of X and replacing L_f by the (weak) Laplacian L of the entire polygonal source mesh, the left side of (12) is nothing but the gradient of E_D (evaluated at vertex i), while the right side is the gradient of area for any *boundary* vertex i . The right side needs to be replaced by zero for inner vertices, which corresponds to the fact that the area gradient vanishes there.

Similar to the case of triangle meshes, two boundary vertices need to be prescribed in order to get a non-degenerate solution. As is well-known from the triangular case, the choice of such two vertices may drastically influence the shape of the solution. An interesting exception is the case where the domain mesh consists of a *single* (possibly non-planar) polygon f . In this case, *any* choice of two prescribed target vertices will lead to the (possibly scaled) *maximal projection* of f —a fact that nicely ties in with our discussion of vector area that motivated our construction of Laplacians.

Automatic boundary placement In order to overcome the dependence of the parameterizing on prescribing boundary targets, the authors of [Mullen et al. 2008] suggest the use of a (generalized) eigenvalue problem based upon the (generalized) Fiedler vector, i.e., the first non-trivial eigenvector of the Laplacian. To this end, consider a quadratic *objective function* for placing the boundary targets, which *only* “sees” the boundary, given by

$$g(Y_B) = \frac{1}{2} Y^T Q \bar{Y},$$

where we write the position vector $Y \in \mathbb{C}^{|V|}$ with complex entries, and we let Q be a Hermitian matrix such that $(Q)_{ij} = 0$ whenever i or j index an inner vertex. For example, Q might be chosen such that $g(Y_B) = \mathcal{A}(Y)$ or such that $g(Y_B) = Y_B^T \bar{Y}_B$.

The conformal parameterization problem with automatic boundary placement can then be formulated as follows:

$$\text{Minimize } E_C(Y) \quad \text{subject to } g(Y_B) = 1. \quad (13)$$

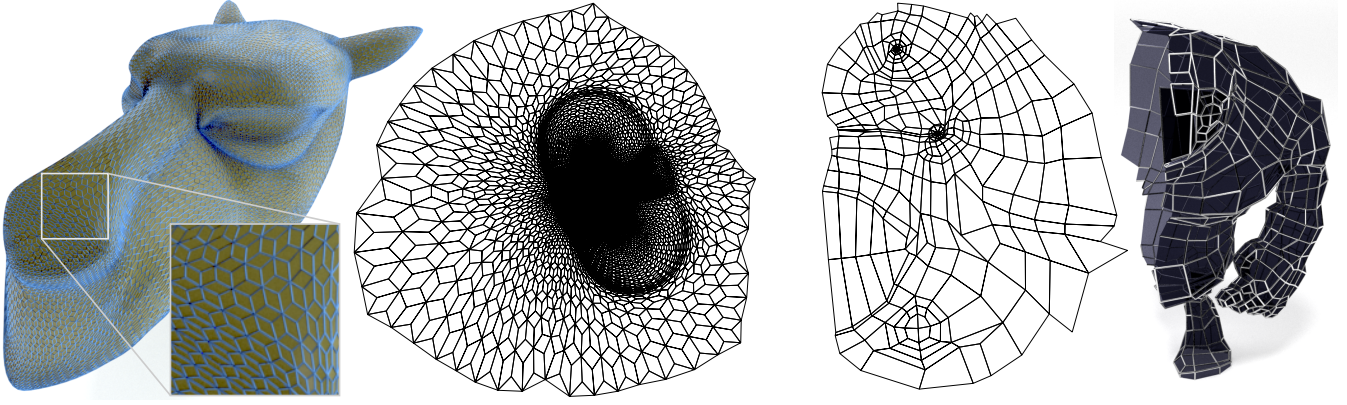


Figure 3: Results of our conformal parameterization with automatic boundary placement. The left mesh has been created by starting from a triangle mesh and then creating a quad mesh with distinguished shape for all quads. The resulting parametrization has smaller conformal energy than the corresponding triangle mesh and approximately preserves the shape of each quad. The right mesh demonstrates the ability of our approach to deal with anisotropic meshes.

In order to see that this approach indeed corresponds to an automatic boundary placement, observe that E_C is a *quadratic* energy. Hence, (13) leads to a generalized eigenvalue problem of the form

$$L_C Y = \mu Q Y, \quad (14)$$

where it follows from (11) that L_C is the Hermitian matrix given by the difference between our Laplacian L and the Hermitian form that gives area. The solution to (14) with *minimal nonzero eigenvalue* μ automatically determines the entire target, including the boundary, without the need of additionally specifying any boundary vertices.

Efficient solution The system (14) contains the $|V| \times |V|$ Hermitian matrix L_C , which might seem to call for an advanced eigenvector solver in order to guarantee efficiency for large meshes. However, we observe that the solution can be obtained by solving a smaller eigenvalue problem that *only* contains the boundary vertices. Assuming a consistent ordering of vertices, we can write

$$L_C = \begin{pmatrix} L_{II} & L_{IB} \\ L_{BI} & L_{BB} \end{pmatrix}, \quad L_{BI} = L_{IB}^T,$$

where I refers to inner vertices and B refers to boundary vertices of the polygonal mesh. Since Q only “sees” boundary vertices, we obtain that the solution to (13) satisfies $(L_C Y)_i = 0$ for any inner vertex i . Therefore, we can write the interior vertex positions as a linear function of the boundary vertices as $L_{II} Y_I = -L_{IB} Y_B$. This allows us to write the conformal energy of the parameterization as of function of the boundary values only:

$$E_C(Y) = \frac{1}{2} Y_B^T L_B Y_B,$$

where $L_B := L_{BB} - L_{IB}^T L_{II}^{-1} L_{IB}$. Note that computing the matrix L_B can be done efficiently without explicitly computing L_{II}^{-1} by solving the sparse linear system $L_{II} A = L_{IB}$.

In summary, we obtain that the eigenvalue problem (14) is *equivalent* to the problem

$$L_B Y_B = \mu Q_B Y_B, \quad (15)$$

where Q_B is the non-zero (lower right) part of Q . This system *only* contains the boundary vertices and is therefore more efficient to solve for than (14) for large meshes with small boundary. Figure 3 provides results of our implementation for $g(Y_B) = \mathcal{A}(Y)$.

We note that for triangle meshes, the results of our approach are equivalent to those of [Mullen et al. 2008] and it appears that their approach could also benefit from our efficient treatment. Likewise, we point out that there is no guarantee that flipped polygons can be avoided—due to a missing maximum principle of our Laplacians.

5.3 A planarizing flow

It follows from the discussion of Section 3.4 that the mean curvature vector of a non-planar polygon f and the corresponding mean curvature vector of its maximal projection, \bar{f} , differ by a term that measures the non-planarity of f . This term is *fully determined* by the CUC^T term of our Laplacian. Turning this observation around, we construct a “Laplacian” \mathcal{L} as before, except that we entirely drop \bar{M}_f from (9). A corresponding energy is given by

$$E_{plan}(X) = \frac{1}{2} \sum_{d=1}^3 X_d^T \mathcal{L} X_d,$$

where X_d denotes the $|V|$ -dimensional vector of the d th components of the mesh positions. Observe that this energy penalizes non-planarity only and vanishes if all polygons are planar.

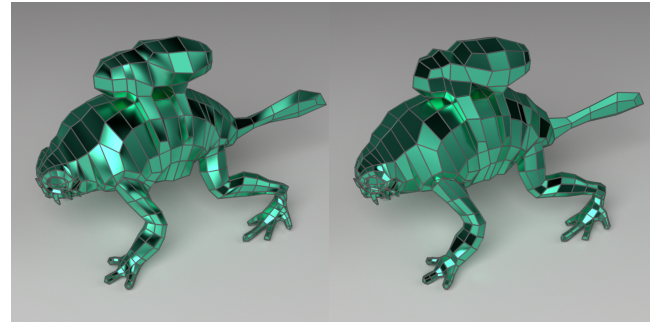


Figure 4: Left: base mesh for a Catmull-Clark surface as used in production, comprised of non-planar faces. Right: the result of applying our proposed planarizing flow.

Figure 4 shows the results of an unconstrained flow down the gradient of E_{plan} , using $\lambda = 2$ (for constructing U) and implemented with a quasi-Newton solver. We stress that this flow is not derived

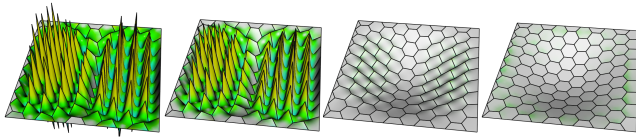


Figure 5: Results for uniform load bending experiments. From left-to-right, we depict the final shape for $\lambda = \{0, 0.005, 0.05, 0.5\}$. Non-gray color encodes (mean) curvature of individual polygons. In particular, choosing λ too small unfavorably affects the solution.

from a PDE, since CUC^T is a combinatorial term. Our experiments show that it is not beneficial to constrain the flow, for example such that the vertices stay close to their original positions or such that the original Laplacian is preserved (as often done in geometric modeling applications).

5.4 Thin plate bending

Since for *planar* meshes our construction is equivalent to that of Brezzi, we refer to the convergence analysis in [Brezzi et al. 2005a] for diffusion problems. These results do not, however, immediately imply convergence of the non-planar case, nor do they immediately imply convergence of thin plate bending scenarios, for which we provide numerical evidence here.

For constructing a thin plate bending energy, we follow the approach in [Wardetzky et al. 2007a], where an efficient quadratic energy (with attendant linear forces and constant Hessian) is proposed. Accordingly, we define bending energy as

$$E_{bend}(u) = \frac{1}{2} u^T \left(L^T M_0^{-1} L \right) u,$$

where u denotes the vertical deflection of a plate. In order to study convergence of minimizers of this energy, we compare the solution for a simply supported square plate, subject to a uniform load, to the analytic solution for this case. In our tests, we observe that the choice of λ (for constructing U) *does matter* for the quality of the solution. Indeed, setting $\lambda = 0$ leads to so-called *spurious modes*—functions u with zero bending energy. This is due to the rank-deficiency of L for polygons with degree greater than three. Although this rank-deficiency is immediately eliminated for *any* $\lambda > 0$, spurious modes can still have an effect on the solution if λ is chosen too small, as shown in Figure 5.

Vice-versa, choosing λ too large inhibits bending. In fact, as shown in Figure 6, the choice $1 \leq \lambda \leq 3$ appears to give the best results, although there exists no optimal choice that simultaneously works for all meshes. We plan to further investigate this limitation in future work.

Figure 7 shows convergence rates under refinement for largely differing polygonal mesh types, indicating linear convergence in the L^2 sense. Grinspun et al. [2006] have investigated the effect of different mesh types for bending problems using the cotan Laplacian. We share their observation that convergence behavior meliorates for irregular unstructured meshes and meshes with hexagonal symmetry, while it deteriorates for strongly anisotropic meshes.

Acknowledgments

We thank Ulrich Bauer, Eitan Grinspun, and Titus Laska for discussions and the anonymous reviewers for their valuable feedback and suggestions for improvement. This work was partially supported by the BMBF Research Project GEOMEC.

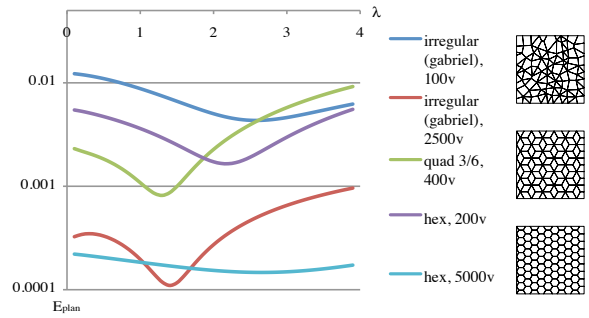


Figure 6: L^2 approximation error for a uniform load bending problem of a simply supported square plate. The plot depicts the error of the solution against different choices of λ for various mesh types.

References

- BELKIN, M., SUN, J., AND WANG, Y. 2008. Discrete Laplace operator on meshed surfaces. In *Proc. of the 24th annual Symposium on Computational Geometry*, 278–287.
- BOBENKO, A. I., AND SPRINGBORN, B. A. 2007. A discrete Laplace–Beltrami operator for simplicial surfaces. *Discrete and Computational Geometry* 38, 4, 740–756.
- BREZZI, F., LIPNIKOV, K., AND SHASHKOV, M. 2005. Convergence of mimetic finite difference methods for diffusion problems on polyhedral meshes. *SIAM J. Num. Anal.* 43, 1872–1896.
- BREZZI, F., LIPNIKOV, K., AND SIMONCINI, V. 2005. A family of mimetic finite difference methods on polygonal and polyhedral meshes. *Math. Models Methods Appl. Sci.* 15, 1533–1553.
- CHEN, R., XUA, Y., GOTSMAN, C., AND LIU, L. 2010. A spectral characterization of the Delaunay triangulation. *Computer Aided Geometric Design* 27, 4, 295–300.
- DESBRUN, M., MEYER, M., SCHRÖDER, P., AND BARR, A. H. 1999. Implicit fairing of irregular meshes using diffusion and curvature flow. In *Proceedings of ACM SIGGRAPH*, 317–324.
- DESBRUN, M., MEYER, M., AND ALLIEZ, P. 2002. Intrinsic parameterizations of triangle meshes. *Computer Graphics Forum (Proc. of Eurographics)*, 209–218.
- DESBRUN, M., HIRANI, A., LEOK, M., AND MARSDEN, J. E. 2005. Discrete exterior calculus. *arXiv:math.DG/0508341*.
- DEY, T., RANJAN, P., AND WANG, Y. 2010. Convergence, stability, and discrete approximation of Laplace spectra. In *Proceedings of SODA*, 650–663.
- DUFFIN, R. 1959. Distributed and Lumped Networks. *Journal of Mathematics and Mechanics* 8, 793–825.
- DYER, R., AND SCHAEFER, S., 2009. Circumcentric dual cells with negative area. Technical Report CMPT2009-6.
- DZIUK, G. 1988. Finite elements for the Beltrami operator on arbitrary surfaces. In *Partial Differential Equations and Calculus of Variations*, vol. 1357 of *Lec. Notes Math.* Springer, 142–155.
- GLICKENSTEIN, D. 2007. A monotonicity property for weighted Delaunay triangulations. *Discrete and Computational Geometry* 38, 651–664.

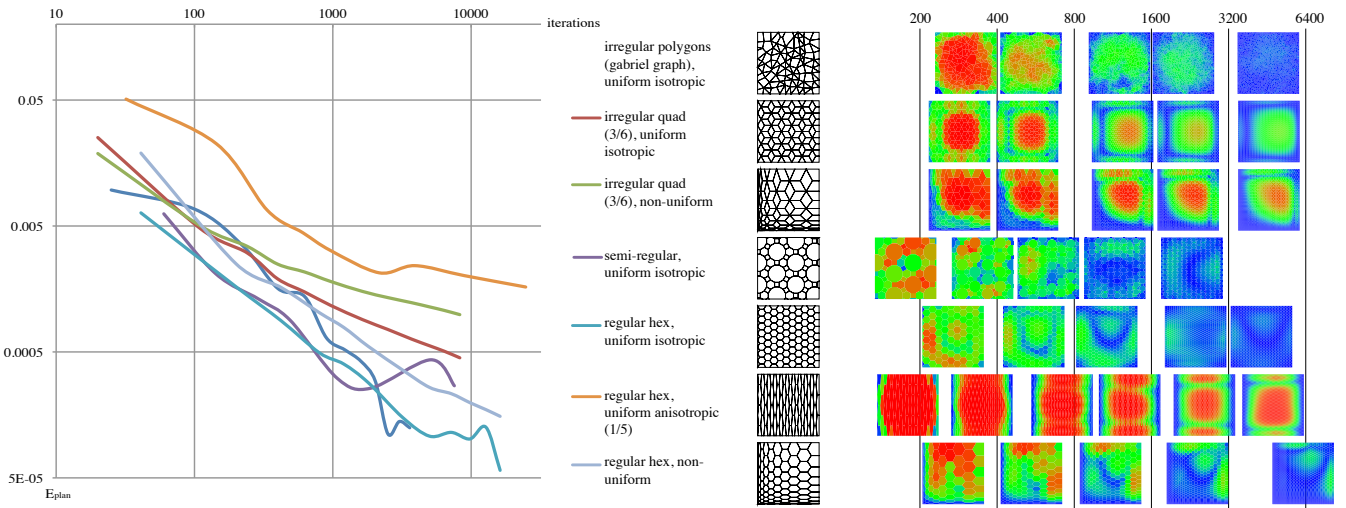


Figure 7: Error plots for the uniform load bending problem of a square plate for various mesh types and resolutions, using $\lambda = 1$. Left: L^2 error, indicating linear convergence. Right: pointwise error, color-coded from red (highest error) to blue (no error). The slight misalignment of the right colored squares reflects the different vertex counts of the various meshes.

GRINSUN, E., GINGOLD, Y., REISMAN, J., AND ZORIN, D. 2006. Computing discrete shape operators on general meshes. *Eurographics (Computer Graphics Forum)* 25, 547–556.

GU, X., AND YAU, S.-T. 2003. Global conformal surface parameterization. In *Sympos. Geom. Processing*, 127–137.

HILDEBRANDT, K., POLTHIER, K., AND WARDETZKY, M. 2006. On the convergence of metric and geometric properties of polyhedral surfaces. *Geometricae Dedicata* 123, 89–112.

LÉVY, B., PETITJEAN, S., RAY, N., AND MAILLOT, J. 2002. Least squares conformal maps for automatic texture atlas generation. *ACM Transactions on Graphics* 21, 3, 362–371.

LIU, D., XU, G., AND ZHANG, Q. 2008. A discrete scheme of Laplace–Beltrami operator and its convergence over quadrilateral meshes. *Computers and Mathematics with Applications* 55, 1081–1093.

MERCAT, C. 2001. Discrete Riemann surfaces and the Ising model. *Communications in Mathematical Physics* 218, 1, 177–216.

MEYER, M., DESBRUN, M., SCHRÖDER, P., AND BARR, A. H. 2003. Discrete differential-geometry operators for triangulated 2-manifolds. In *Visualization and Mathematics III*, H.-C. Hege and K. Polthier, Eds. Springer, 35–57.

MULLEN, P., TONG, Y., ALLIEZ, P., AND DESBRUN, M. 2008. Spectral conformal parameterization. *Computer Graphics Forum (Proc. of SGP)* 27, 5, 1487–1494.

PEROT, J., AND SUBRAMANIAN, V. 2007. Discrete calculus methods for diffusion. *Journal of Comp. Physics* 224, 1, 59–81.

PINKALL, U., AND POLTHIER, K. 1993. Computing discrete minimal surfaces and their conjugates. *Experim. Math.* 2, 15–36.

RIPPA, S. 1990. Minimal roughness property of the Delaunay triangulation. *Computer Aided Geometric Design* 7, 489–497.

ROSENBERG, S. 1997. *The Laplacian on a Riemannian manifold*. No. 31 in Student Texts. London Math. Soc.

SHAFIQU-ISMAM, M., AND RATHOD, H. 2006. Alternative approach of numerical integration for rational functions related to

linear convex quadrilateral finite elements. *Journal of App. Sciences Research* 2, 9, 533–540.

SULLIVAN, J. 2008. Curvatures of smooth and discrete surfaces. In *Discrete Differential Geometry*, A. I. Bobenko, J. M. Sullivan, P. Schröder, and G. Ziegler, Eds. Birkhäuser Basel, 175–188.

WARDETZKY, M., BERGOU, M., HARMON, D., ZORIN, D., AND GRINSUN, E. 2007. Discrete Quadratic Curvature Energies. *Computer Aided Geometric Design* 24, 499–518.

WARDETZKY, M., MATHUR, S., KÄLBERER, F., AND GRINSUN, E. 2007. Discrete Laplace operators: No free lunch. In *Siggraph/Eurographics Sympos. Geom. Processing*, 33–37.

XIONG, Y., LI, G., AND HAN, G. 2011. Mean Laplace–Beltrami operator for quadrilateral meshes. In *Transactions on Edutainment V*, Springer, 189–201.

XU, G. 2004. Discrete Laplace–Beltrami operators and their convergence. *Computer Aided Geometric Design* 21, 8, 767–784.

Appendix

We here provide the proofs of the various claims made in this paper. We additionally require the following result.

Lemma 6. Let f be polygon with height vector $h = (h_1, \dots, h_k)^T$ over its maximal projection, \bar{f} , and assume that the origin of \mathbb{R}^3 lies in the plane of \bar{f} . Then $\bar{M}_f - \bar{M}_{\bar{f}} = |f|^{-1} h_b \otimes h_b$, where $h_b = \frac{1}{2}(h_1 + h_2, \dots, h_k + h_1)^T$ denotes the midpoint height vector.

Proof of Lemma 6. The edge midpoint position matrices of f and \bar{f} satisfy the relation $B_f = B_{\bar{f}} + h_b \bar{n}^T$, where \bar{n} denotes to normal vector to \bar{f} . Then, since the origin is contained in the plane of \bar{f} by assumption and hence \bar{n} is orthogonal to every column of $B_{\bar{f}}$, we have $B_f B_f^T = B_{\bar{f}} B_{\bar{f}}^T + h_b \otimes h_b$. Using $|f| = |\bar{f}|$ then proves the claim. \square

Proof of Lemma 1. Let (e_1, \dots, e_k) be the cyclically ordered oriented edges of f and let (b_1, \dots, b_k) denote the corresponding

midpoint positions. Using a ccw ordering of the vertices, we then have $\mathbf{e}_i = \mathbf{x}_{i+1} - \mathbf{x}_i$ and $\mathbf{b}_i = (\mathbf{x}_{i+1} + \mathbf{x}_i)/2$. Therefore,

$$A_f = E_f^T B_f = \sum_{i=1}^k \mathbf{e}_i \otimes \mathbf{b}_i = \frac{1}{2} \sum_{i=1}^k (\mathbf{x}_{i+1} \otimes \mathbf{x}_i - \mathbf{x}_i \otimes \mathbf{x}_{i+1})$$

is clearly skew-symmetric, where \otimes denotes the outer product and the last sum is assumed to be cyclic. Moreover, one easily verifies that $\frac{1}{2}(\mathbf{x}_{i+1} \otimes \mathbf{x}_i - \mathbf{x}_i \otimes \mathbf{x}_{i+1})$ is a skew-symmetric matrix with Darboux vector equal to $\frac{1}{2}(\mathbf{x}_i \times \mathbf{x}_{i+1})$. The last term, in turn, is equal to the vector area of the triangle formed by 0, \mathbf{x}_i , and \mathbf{x}_{i+1} , where 0 is the origin. Hence the Darboux vector $[A_f]$ is equal to the vector area of the polyhedral surface with triangular faces and vertices 0, $\mathbf{x}_1, \dots, \mathbf{x}_k$. The claim follows since vector area only depends on the boundary curve, not on the spanning surface. \square

Proof of Lemma 2. By definition, \bar{f} is a maximal projection of f iff \bar{f} is an orthogonal projection and $A_f = A_{\bar{f}}$. According to the proof of Lemma 1 and using that $\mathbf{x}_i = \bar{\mathbf{x}}_i + h_i \bar{\mathbf{n}}$, we find that $A_f = A_{\bar{f}}$ is equivalent to

$$0 = \frac{1}{2} \sum_{i=1}^k (x_{i+1}(\bar{\mathbf{n}} \otimes \bar{\mathbf{x}}_i - \bar{\mathbf{x}}_i \otimes \bar{\mathbf{n}}) - x_i(\bar{\mathbf{n}} \otimes \bar{\mathbf{x}}_{i+1} - \bar{\mathbf{x}}_{i+1} \otimes \bar{\mathbf{n}})) ,$$

where the sum is assumed to be cyclic. Since $(\bar{\mathbf{n}} \otimes \bar{\mathbf{x}}_i - \bar{\mathbf{x}}_i \otimes \bar{\mathbf{n}})$ has Darboux vector $(\bar{\mathbf{x}}_i \times \bar{\mathbf{n}})$, we obtain the equivalent requirement

$$0 = \bar{\mathbf{n}} \times \frac{1}{2} \sum_{i=1}^k (h_i \bar{\mathbf{x}}_{i+1} - h_{i+1} \bar{\mathbf{x}}_i) = \bar{\mathbf{n}} \times \sum_{i=1}^k \frac{h_i + h_{i+1}}{2} \bar{\mathbf{e}}_i ,$$

which proves the claim. \square

Proof of Lemma 3. Using the proof of Lemma 1, it is straightforward to see that

$$(\nabla_{\mathbf{x}_i} |f|)^T = \frac{1}{|f|} \langle \nabla_{\mathbf{x}_i} [A_f], [A_f] \rangle = \frac{1}{|f|} A_f (\mathbf{b}_{i-1} - \mathbf{b}_i) .$$

Therefore,

$$(\nabla_{\mathbf{x}} |f|)^T = \frac{1}{|f|} A_f (B_f^T \mathbf{d}) = \frac{1}{|f|} E_f^T B_f B_f^T \mathbf{d} .$$

Using that $E_f = \mathbf{d} X_f$ then proves the claim. \square

Proof of Lemma 4. Let $\star E_f := \widetilde{M}_f E_f$. By Lemma 1, $|f| \star \mathbf{e}_i^f = [A_f] \times \mathbf{b}_i^f$ for the i th column of $\star E_f^T$. In particular, for planar meshes we obtain $\star \mathbf{e}_i^f = \mathbf{n} \times \mathbf{b}_i^f$, where \mathbf{n} is the plane's unit normal vector. Clearly, the edge midpoints \mathbf{b}_i^f satisfy (7) (using \mathbf{b}^f in place of $\star \mathbf{e}^f$) due to pairwise cancelations as we cyclically traverse the faces around an inner vertex. Therefore, for planar meshes, the vectors $\star \mathbf{e}_i^f = \mathbf{n} \times \mathbf{b}_i^f$ also satisfy (7). \square

Proof of Lemma 5. We first show that $\widetilde{H}_f = H_{\bar{f}}$. The definition of \widetilde{H}_f is independent of the choice of origin. We may therefore assume that the origin is contained in the plane of \bar{f} . Letting h_f denote the height vector of f over \bar{f} , with corresponding midpoint height vector h_b , Lemma 6 shows that $\widetilde{M}_f = \widetilde{M}_{\bar{f}} + |f|^{-1} h_b \otimes h_b$. Letting $h_e = \mathbf{d} f_f$, we find that $E_f = E_{\bar{f}} + h_e \bar{\mathbf{n}}^T$. From Lemma 2, we know that h_b is in the kernel of $E_{\bar{f}}^T$. Moreover, it is easy to see that $h_b^T h_e = 0$. Hence $(h_b \otimes h_b) E_f = 0$, so that $\widetilde{M}_f E_f = \widetilde{M}_{\bar{f}} E_f$. Using the definition of $\widetilde{M}_{\bar{f}}$ we find that $|\bar{f}| \widetilde{M}_{\bar{f}} E_f = B_{\bar{f}} B_{\bar{f}}^T (E_{\bar{f}} +$

$h_e \bar{\mathbf{n}}^T)$. It is easy to see that $E_{\bar{f}}^T h_b = 0$ (Lemma 2) is equivalent to $B_{\bar{f}}^T h_e = 0$. Hence $\widetilde{M}_{\bar{f}} E_{\bar{f}} = \widetilde{M}_{\bar{f}} E_f = \widetilde{M}_{\bar{f}} E_f$ and thus $\widetilde{H}_f = H_{\bar{f}}$. Finally, using that (i) $E_f = E_{\bar{f}} + h_e \bar{\mathbf{n}}^T$, (ii) the columns of $C_{\bar{f}}$ are an orthonormal basis spanning the kernel of $E_{\bar{f}}^T$, and (iii) $U_{\bar{f}} = \lambda \text{Id}$, we find that

$$(C_{\bar{f}} U_{\bar{f}} C_{\bar{f}}^T) E_f = (C_{\bar{f}} U_{\bar{f}} C_{\bar{f}}^T) h_e \bar{\mathbf{n}}^T = \lambda \bar{h}_e \bar{\mathbf{n}}^T ,$$

where \bar{h}_e is the orthogonal projection of h_e to $\ker E_{\bar{f}}^T$. The claim follows. \square

Proof of Theorem 1. With a few modifications, the original proof of [Brezzi et al. 2005b] carries over to the non-planar case. Indeed, the $k \times k$ matrix M_f is symmetric and positive semidefinite by construction. It remains to show that its kernel is trivial. Let $v \in \mathbb{R}^k$ with $M_f v = 0$. We need to show that $v = 0$. We have $v^T M_f v = 0$ and hence

$$0 = \frac{1}{|f|} \|B_f^T v\|^2 + \|U_{\bar{f}}^{1/2} C_{\bar{f}}^T v\|^2 ,$$

where $U_{\bar{f}}^{1/2}$ is a square root of the symmetric positive definite matrix $U_{\bar{f}}$. Hence $B_f^T v = 0$ and $C_{\bar{f}}^T v = 0$. In particular, since $\ker(E_{\bar{f}}^T) = \text{im}(C_{\bar{f}})$, we have that

$$v \in \ker(C_{\bar{f}}^T) = \{\text{im}(C_{\bar{f}})\}^\perp = \{\ker(E_{\bar{f}}^T)\}^\perp = \text{im}(E_{\bar{f}}) .$$

Therefore, there exists $\mathbf{u} \in \mathbb{R}^3$ with $v = E_{\bar{f}} \mathbf{u}$. Then $0 = B_f^T v = B_f^T E_{\bar{f}} \mathbf{u}$. Using that $E_f = E_{\bar{f}} + h_e \bar{\mathbf{n}}^T$, where $h_e = \mathbf{d} h_f$ and h_f is the height vector of f over its maximal projection \bar{f} , we obtain $0 = B_f^T E_{\bar{f}} \mathbf{u} = B_f^T E_f \mathbf{u}$. Indeed, the last equality follows from $B_f^T h_e = -E_f^T h_b = -E_{\bar{f}}^T h_b - \bar{\mathbf{n}} h_e^T h_b = 0$ by Lemma 2. Hence $A_f^T \mathbf{u} = -A_f \mathbf{u} = 0$ by Lemma 1. Since the kernel of A_f is spanned by $[A_f]$, which is parallel to $\bar{\mathbf{n}}$, we have $\mathbf{u} = \mu \bar{\mathbf{n}}$. It follows that $v = \mu E_{\bar{f}} \bar{\mathbf{n}} = 0$, since $\bar{\mathbf{n}}$ is orthogonal to the rows of $E_{\bar{f}}$. \square

Proof of Theorem 2. By construction, if f is a triangle, then C_f must be a 3-vector that is parallel to $(1, 1, 1)^T$, while U_f is a positive scalar. In particular, $C_f^T \mathbf{d} = 0$. Therefore, we have $L = \mathbf{d}^T M_f \mathbf{d} = \mathbf{d}^T \widetilde{M}_f \mathbf{d} = |f|^{-1} (\mathbf{d}^T B_f) (\mathbf{d}^T B_f)^T$. Moreover, since $B_f = (\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3)^T$ by definition, we obtain

$$B_f^T \mathbf{d} = (\mathbf{b}_3 - \mathbf{b}_1, \mathbf{b}_1 - \mathbf{b}_2, \mathbf{b}_2 - \mathbf{b}_3) = \frac{1}{2} (\mathbf{e}_2, \mathbf{e}_3, \mathbf{e}_1) .$$

Therefore, the entry (i, j) of $(\mathbf{d}^T B_f) (\mathbf{d}^T B_f)^T$ is given by the inner product $\frac{1}{4} \langle \mathbf{e}_{i+1}, \mathbf{e}_{j+1} \rangle$, where the subscripts are valid modulo 3. Since these inner products can be written using cosines, while the area, $|f|$, can be expressed using sines, we obtain that $L_{ij} = -\frac{1}{2} \cot \alpha_{ij}$ for $i \neq j$, where $0 < \alpha_{ij} < \pi$ is the angle between edges \mathbf{e}_{i+1} and \mathbf{e}_{j+1} , which is the cotan operator. \square