A Hands-On Introduction to Discrete Differential Operators on Polygon Meshes

S. D. Wagner[1], A. Bunge[2], and M. Botsch[1,3]

[1]TU Dortmund University, Germany
[2]AutoForm Engineering, Germany
[3]Lamarr Institute for Machine Learning and Artificial Intelligence, Germany

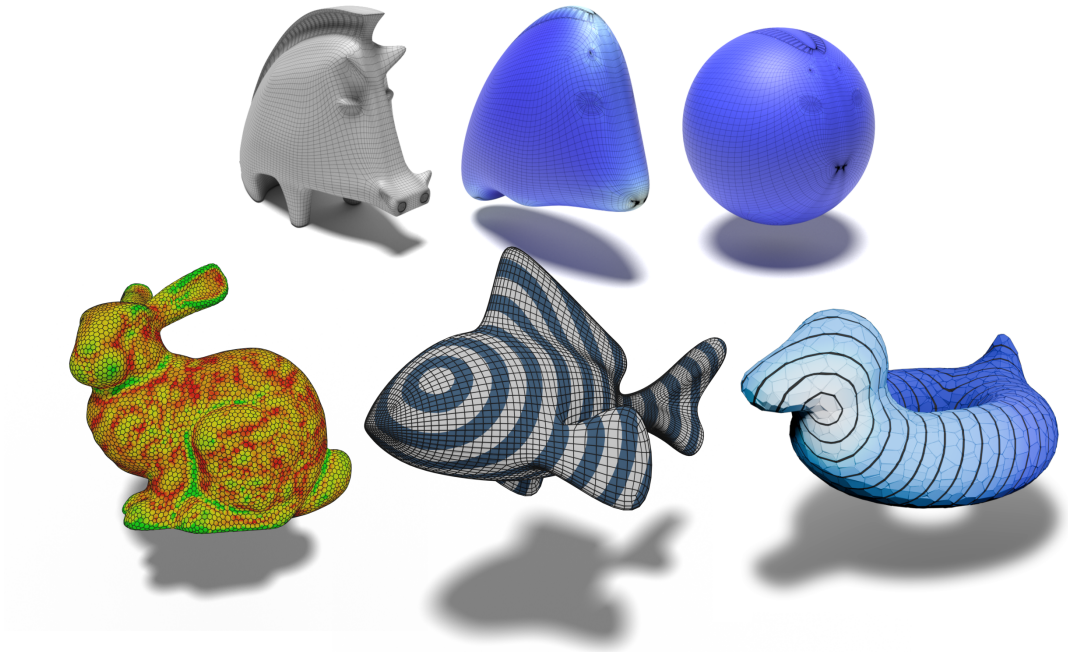


Figure 1: *A selection of applications of the Laplace operator on different polygon meshes.*

Abstract

Many applications in geometry processing involve the solution of partial differential equations on discrete surface meshes, with the Laplacian undoubtedly being the most ubiquitous operator in this context. Having discrete operators for gradient, divergence, and Laplacian at hand allows to solve many interesting geometry processing problems.
Unfortunately, many approaches or implementations require the mesh to be a well-behaved triangle mesh with good-quality elements, and severely degrade or completely fail if these conditions are not met. In this tutorial, we will present how to discretize (and implement) gradient, divergence, and Laplacian operators in a simple, flexible, and robust manner. The presented discrete differential operators can be applied to triangle meshes, quad meshes, or general polygon meshes, they work robustly even for low-quality or degenerate elements, and as such, they allow to generalize many geometry processing algorithms to a much wider range of mesh inputs. We also provide interactive HTML-based course notes at https://graphics.rocks/eg26DDG.

CCS Concepts

• ***Mathematics of computing*** → ***Discretization;*** • ***Computing methodologies*** → ***Mesh geometry models;***

1. Introduction

The discrete Laplace-Beltrami operator, or Laplacian for short, is a ubiquitous tool in geometry processing. It allows us to solve numerous partial differential equations on discrete surface meshes, which is essential for various computer graphics applications, like mesh smoothing, mesh parameterization or fairing, followed by many others. For triangulated surfaces, the discrete Laplacian based on the cotangent formula [PP93, MDSB03, DMSB99, Dzi88, BB23] is omnipresent in graphics and geometry processing.

However, despite its seemingly harmless definition (see Equations (12) and (13)), there are some factors that need to be considered with regard to the robustness of the cotangent Laplacian. As soon as meshes start to appear in a more degenerate form, numerical stability can become a significant challenge. At the same time, the growing demand for unsupervised processing of large-scale datasets makes the use of efficient and robust implementations of discrete operators more important than ever. Therefore, this course will follow the investigation of Wagner and Botsch [WB25] and analyze the cotangent implementation of modern general-purpose geometry processing libraries and their resilience to challenging tessellations. Furthermore, we will discuss approaches that go beyond linear FEM and review alternative, more robust Laplace discretizations [BS07, SC20, WB25, QKL*26].

Additionally, due to the growing needs in modeling and engineering applications, recent papers point out that the restriction to triangle meshes, while simple and convenient, is no longer sufficient. Many users have to fall back to more general shapes to be able to express geometric properties and features in their model. Applications benefiting from a more flexible range of elements are, for example, fracture modeling [TS08, Bis09, OSTLY12] or linear elasticity problems [TS06]. Additionally, since micro-structures of naturally occurring materials like bones can be described through polygonal domains, generalized differential operators are useful tools for the solid- and bio-mechanics community [TS06]. Furthermore, modeling artists predominantly use quad meshes.

In order to enable this flexibility, several papers within the graphics community developed strategies to generalize the Laplace-Beltrami operator to general polygon meshes. This presents several challenges. For instance, arbitrary polygons might not be inherently planar, potentially resulting in twisted surfaces in 3D. Coming from various backgrounds and inspirations, all introduced Laplacians address these difficulties in their own unique way, but it is not necessarily clear in which aspects the operators actually differ and what their various nuances imply. This tutorial, based on the state-of-the-art report by Bunge and Botsch [BB23] and course by Bunge et al. [BAB23], therefore intends to showcase occurring similarities between the presented methods that may not be apparent if considered individually.

However, the scope of this course is not limited to the Laplacian operator alone. In the smooth setting, the Laplacian of a function f is defined as

$$\Delta f = \operatorname{div} \nabla f. \tag{1}$$

Given their close relation, comparing suitable generalizations for discrete gradient and divergence is an almost equally essential affair focused on in this tutorial. In general, the papers we are going to discuss have all been inspired by different well-known numerical schemes commonly used for various discretization problems. Namely, the Finite Element Method (FEM), the Mimetic Finite Difference Method (MFD), and the Finite Volume Method (FV). Since all generalized Laplacians can be loosely sorted into one of these schemes, we will briefly explain their core principles at the beginning of each section and highlight the inspirational elements that influenced the respective papers.

We start by discussing the required properties a discrete Laplacian should fulfill based on the work presented by Wardetzky et al. [WMKG07]. Afterwards, we derive the cotangent Laplacian on standard triangle meshes and examine its numerical stability, followed by a discussion of more robust Laplace discretizations as presented by Wagner and Botsch [WB25]. We then extend the setting to polygonal meshes, introducing the necessary definitions required for constructing the corresponding operators. Building on this foundation, we provide a more detailed analysis of the involved polygonal operators, their different ideas, and the properties they fulfill. Finally, the different discretizations are evaluated in a variety of quantitative comparisons that address recurring debates within the original papers. In the end, the tutorial intends to provide the reader with an intuition to choose the optimal operator discretization for their given situation.

2. Basic Definitions

Consider a 2D polygon mesh $\mathcal{M} = (\mathcal{V}, \mathcal{E}, \mathcal{F})$ embedded in 3D, with vertices $\mathcal{V}$, edges $\mathcal{E}$, and faces $\mathcal{F}$. Each vertex $v_i \in \mathcal{V}$ has an associated 3D position $\mathbf{x}_i = (x_i, y_i, z_i)$ and each face f consists of n_f vertices. We define an additional set of oriented halfedges $\mathcal{H}$, where for each inner edge $e \in \mathcal{E}$ there exist two oppositely oriented halfedges, while each boundary edge has only one.

We define a discrete Laplace operator $\mathbb{L} \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$ as the product of the inverse of a so-called *mass matrix* $\mathbf{M} \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$ and *stiffness matrix* $\mathbf{L} \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$:

$$\mathbb{L} = \mathbf{M}^{-1}\mathbf{L}. \tag{2}$$

$\mathbb{L}$ is generally referred to as the strong form of the Laplacian and $\mathbf{L}$ is its integrated weak form. The exact conditions that are imposed on these matrices will be discussed in the next section. As for their construction, most of the upcoming methods focus on a local approach that builds the required matrices per face. We therefore define:

- The matrix $\mathbf{X} \in \mathbb{R}^{|\mathcal{V}|\times 3}$ encodes the vertex positions of the mesh in its rows.
- $\mathbf{X}_f = (\mathbf{x}_1^f, \dots, \mathbf{x}_{n_f}^f)^\mathsf{T}$ is the $n_f \times 3$ matrix containing in its rows the cyclically ordered vertex positions $\mathbf{x}_i^f$ of the face f.
- $\mathbf{E}_f = (\mathbf{e}_1^f, \dots, \mathbf{e}_{n_f}^f)^\mathsf{T}$ is the $n_f \times 3$ matrix containing in its rows the cyclically ordered edge vectors $\mathbf{e}_i^f = \mathbf{x}_{i+1}^f - \mathbf{x}_i^f$ of the face f.
- $\mathbf{B}_f = (\mathbf{b}_1^f, \dots, \mathbf{b}_{n_f}^f)^\mathsf{T}$ is the $n_f \times 3$ matrix containing in its rows the barycenters $\mathbf{b}_i^f = \frac{1}{2}\left(\mathbf{x}_{i+1}^f + \mathbf{x}_i^f\right)$ of each edge $\mathbf{e}_i^f$.

2.1. Properties of a Discrete General Laplace Operator

The smooth Laplace-Beltrami operator has a set of key structural properties that each discretization should be able to fulfill. The

correlation between these smooth properties and discrete Laplace operators has been discussed intensively for triangle meshes by Wardetzky et al. [WMKG07]. However, these requirements equally hold for general polygon meshes and are therefore important criteria for the numerical quality of a discrete Laplacian. Unfortunately, as pointed out by Wardetzky et al., most meshes do not allow for Laplacians to satisfy all discrete properties simultaneously, which coins the second part of their paper "No free lunch". In this section, we will reintroduce the individual definitions presented by Wardetzky et al. [WMKG07] in order to establish characteristics by which the quality of each presented polygon Laplacian operator can be assessed.

In the smooth setting, consider a single connected manifold Ω, possibly with boundary, that is equipped with a Riemannian metric. We define a function $u : \Omega \to \mathbb{R}$ and its discrete equivalent $\mathbf{u} \in \mathbb{R}^{|\mathcal{V}|}$, whose entries are the function values of u sampled at the vertices of the surface mesh $\mathcal{M}$. The strong Laplacian $\mathbb{L} \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$ defined on $\mathcal{M}$ is given as a $|\mathcal{V}| \times |\mathcal{V}|$ matrix pair $(\mathbf{M}, \mathbf{L})$ consisting of a sparse symmetric mass matrix $\mathbf{M}$ and the weak form of the Laplacian given by the sparse matrix $\mathbf{L}$.

Symmetry. Given two functions u and v that are sufficiently smooth and vanish along the boundary of Ω, the smooth Laplacian is self-adjoint with respect to the L^2 inner product of these functions, meaning

$$\langle \Delta u, v \rangle = \langle u, \Delta v \rangle \tag{3}$$

with $\langle u, v \rangle = \int_\Omega uv \, dA$. We therefore request the strong form $\mathbb{L}$ to be a self-adjoint operator with respect to the inner product induced by the symmetric mass matrix $\mathbf{M}$, meaning

$$(\mathbb{L}\mathbf{u})^\mathsf{T}\mathbf{M}\mathbf{v} = \mathbf{u}^\mathsf{T}\mathbf{M}(\mathbb{L}\mathbf{v}) \tag{4}$$

$$\Leftrightarrow \quad \mathbf{u}^\mathsf{T}\mathbf{L}^\mathsf{T}\mathbf{v} = \mathbf{u}^\mathsf{T}\mathbf{L}\mathbf{v} \tag{5}$$

for any $\mathbf{u}$ and $\mathbf{v}$.

Locality. The smooth Laplacian of a function u at a point $\mathbf{p}$ should only depend on the values $u(\mathbf{q})$ of other points $\mathbf{q}$ in an ϵ-ball around $\mathbf{p}$. This means that the discrete Laplacian should also operate locally in the 1-ring neighborhood of the respective vertex and should not be affected by distant vertices in the mesh.

Linear Precision. In the smooth setting, given a linear function u defined on Ω, the Laplacian of said function has to be zero in planar regions of the manifold. The discrete equivalent is similar: Given a *planar* mesh $\mathcal{M}$ and any linear function u, we require the strong version of the Laplacian $\mathbb{L}$ to satisfy

$$(\mathbb{L}\mathbf{u})_i = 0 \tag{6}$$

for each inner vertex v_i, where $(\cdot)_i$ denotes the i-th entry or row of the vector or matrix within the parentheses. Alternatively, we can omit the influence of the mass matrix and require the stiffness matrix to satisfy the equivalent condition

$$(\mathbf{L}\mathbf{X})_i = 0 \tag{7}$$

on the vertex positions $\mathbf{X}$.

Positive Semi-Definiteness and Null Space. In the smooth setting, the Dirichlet energy of a function u defined on the manifold Ω has to be greater than or equal to zero. The discrete version of the Dirichlet energy can be expressed with the help of the stiffness matrix as

$$\frac{1}{2}\mathbf{u}^\mathsf{T}\mathbf{L}\mathbf{u}. \tag{8}$$

Therefore, $\mathbf{L}$ has to be positive semi-definite in order for the energy to remain non-negative. Note that, depending on the definition, the Laplacian could alternatively be required to be negative semi-definite. A second aspect of this property addresses the kernel of the Laplacian. The smooth Dirichlet energy vanishes for constant functions. Therefore, the kernel of $\mathbf{L}$ has to be one-dimensional as well and can only contain constant functions. If the stiffness matrix can be expressed as

$$(\mathbf{L}\mathbf{u})_i = \sum_j w_{ij}(u_i - u_j), \tag{9}$$

the discrete Laplacian automatically fulfills this property [WMKG07].

Maximum Principle. The smooth maximum principle requires that harmonic functions ($\Delta u = 0$) have no local extremum at interior points of the manifold Ω. For example, this property assures that approximated solutions of diffusion problems flow from regions with higher potential to regions with lower potential, instead of the other way round. The discrete equivalent can be directly addressed through the entries of the stiffness matrix by the so-called *positive weight* property, which is a sufficient but not necessary condition for the discrete maximum principle. It demands that for each vertex v_i the entries $\mathbf{L}_{ij}$ have to be less than or equal to zero if $i \neq j$ and at least one entry per row has to be nonzero.

Convergence. The convergence property requires that approximate solutions involving the Laplace operator converge to the exact solution of the PDE under refinement of the mesh, which was analyzed by Hildebrandt et al. [HPW06] and Wardetzky [War08]. This property will not be proven for the upcoming operators, but analyzed empirically in the results section.

3. Laplacians on Triangle Meshes

For triangle meshes, the canonical choice of Laplacian discretization is the cotangent Laplacian. Furthermore, almost all of the later discussed polygon Laplacians reproduce it on triangle meshes. We will therefore revisit its definition based on the finite element discretization, discuss its behavior on (near-)degenerate elements, and describe ways to improve its robustness.

Given a triangle mesh $\mathcal{M}$, let $\{\varphi_1, \ldots, \varphi_{|\mathcal{V}|}\}$ be the piecewise linear Lagrange basis functions defined on $\mathcal{M}$, with

$$\varphi_i(\mathbf{x}_j) = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{otherwise.} \end{cases} \tag{10}$$

The mass and stiffness matrices $\mathbf{M}, \mathbf{L} \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$ of the Laplace operator are then discretized as

$$\mathbf{M}_{ij} = \int_{\mathcal{M}} \varphi_i \cdot \varphi_j = \begin{cases} \frac{|t_{ijk}|+|t_{jih}|}{12} & \text{if } j \in \mathcal{N}(i), \\ \sum_{k\in\mathcal{N}(i)} \mathbf{M}_{ik} & \text{if } j = i, \\ 0 & \text{otherwise,} \end{cases} \tag{11}$$

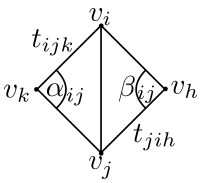
and

$$\mathbf{L}_{ij} = \int_{\mathcal{M}} \langle \nabla\varphi_i, \nabla\varphi_j \rangle = \begin{cases} -w_{ij} & \text{if } j \in \mathcal{N}(i), \\ \sum_{k\in\mathcal{N}(i)} w_{ik} & \text{if } j = i, \\ 0 & \text{otherwise,} \end{cases} \tag{12}$$

with

$$w_{ij} = \frac{\cot\alpha_{ij} + \cot\beta_{ij}}{2}. \tag{13}$$

Here $t_{ijk} = (v_i, v_j, v_k)$ and $t_{jih} = (v_j, v_i, v_h)$ denote the triangles adjacent to the edge e_{ij} between the vertices (v_i, v_j), with $|t_{ijk}|, |t_{jih}|$ describing their respective areas (see inset). The angles α_{ij} and β_{ij} lie in the opposite corners of the adjacent triangles and $\mathcal{N}(i)$ denotes the one-ring neighborhood surrounding v_i.



Emulating the smooth setting with the Laplacian being defined as the divergence of the gradient, one can express the gradient operator as a discrete matrix $\mathbf{G} \in \mathbb{R}^{3|\mathcal{F}|\times|\mathcal{V}|}$ consisting of local sub-matrices $\mathbf{G}_i \in \mathbb{R}^{3\times3}$ per triangle $f_i = t_{jkl}$. Each column of $\mathbf{G}_i$ is associated with the gradient of one of the respective vertices. For example, the first column referring to vertex v_j would be

$$\mathbf{G}_i(:,1) = \frac{(\mathbf{x}_l - \mathbf{x}_k)^\perp}{2\,|t_{jkl}|}, \tag{14}$$

where $\mathbf{v}^\perp$ denotes a counterclockwise rotation of the vector $\mathbf{v}$ by 90° in the triangle plane. The global matrix $\mathbf{G}$ is then assembled by placing the respective face gradients at the column entries of the individual vertices v_j and setting everything else to zero. This can further be used to discretize the divergence as

$$\mathbf{D} = \mathbf{G}^\mathsf{T}\hat{\mathbf{M}}, \tag{15}$$

with $\hat{\mathbf{M}} \in \mathbb{R}^{3|\mathcal{F}|\times3|\mathcal{F}|}$ being the diagonal mass matrix containing the area of the triangle i in the three consecutive diagonal entries associated with face i [BSPG06]. The product of $\mathbf{D}$ and $\mathbf{G}$ gives us the stiffness matrix $\mathbf{L}$, which is consistent with the continuous setting, but requires a concrete embedding of the mesh in contrast to the intrinsic formulation of $\mathbf{L}$ itself [Sha21].

3.1. Properties

Symmetry. Considering the individual entries of the stiffness matrix defined in Equation (12), the cotan operator is symmetric by construction.

Positive Semi-Definiteness and Kernel Dimension. Since $\mathbf{L}$ can be considered as the Gramian matrix of the gradients of the linear Lagrange basis functions, it is positive semi-definite by construction. Given that $\Delta u(\mathbf{x}_i)$ of a function u at vertex v_i can be expressed through the well known cotan formula [Mac49, PP93, DMSB99]

$$\Delta u(\mathbf{x}_i) = \frac{1}{2}\sum_{v_j\in\mathcal{N}(v_i)} (\cot\alpha_{ij} + \cot\beta_{ij})(u(\mathbf{x}_j) - u(\mathbf{x}_i)), \tag{16}$$

the operator satisfies the condition given in Equation (9) and therefore has a one dimensional kernel.

Locality. By construction, each row $(\mathbf{L})_i$ associated to vertex v_i has non-zero entries only in the columns associated to nodes in its immediate one-ring neighborhood.

Linear Precision. The area gradient of a triangle t_{ijk} with respect to the vertex v_i can be expressed as

$$\nabla_{\mathbf{x}_i} A = \frac{\cot\theta_k}{2}(\mathbf{x}_i - \mathbf{x}_j) + \frac{\cot\theta_j}{2}(\mathbf{x}_i - \mathbf{x}_k), \tag{17}$$

with θ_j and θ_k denoting the angles at the respective vertices v_j,v_k and $A = |t_{ijk}|$ the area of the triangle t_{ijk}. The cotan Laplace of the coordinate function $\mathbf{x}_i$ at vertex v_i can therefore be expressed as the sum of area gradients of its adjacent triangles [DMSB99], which, if the triangles all lie within the same plane, becomes zero.

Maximum Principle. This property is in general not satisfied, since the cotangent becomes negative for angles between 90 and 180 degrees, leading to positive entries $\mathbf{L}_{ij}$ in the stiffness matrix if the involved angles satisfy $\alpha_{ij} + \beta_{ij} > \pi$.

Convergence. The convergence behavior of the cotan Laplace was discussed by Hildebrand et al. [HPW06] and Wardetzky [War08]. They point out that pointwise convergence of refined meshes $\mathcal{M}$ to a smooth surface Ω is not sufficient to guarantee convergence for the cotan Laplace. However, if the meshes converge in Hausdorff distance and are normal graphs over Ω, then the Laplacian converges to its smooth solution.

3.2. Computing the Cotangent Laplacian

While calculating the corner angles of the triangles and applying the cotangent is the easiest way to compute the cotangent weights, as defined in Equation (13), several mathematical identities can make the calculations more performant and numerically stable. This section describes the different ways to calculate the cotangent Laplacian present in modern geometry processing libraries, as detailed by Wagner and Botsch [WB25].

Trigonometry. Trigonometry is still the most widely used method of calculation, but there are several different variations. OpenMesh [BSBK02], Geogram [L*25], VCGLib [Vis23], and CinoLib [Liv19] each first calculate the corner angles using the standard cosine identity

$$\alpha_i = \arccos\left(\frac{\langle \mathbf{e}_{ij}, \mathbf{e}_{ik}\rangle}{\|\mathbf{e}_{ij}\| \cdot \|\mathbf{e}_{ik}\|}\right) \tag{18}$$

at vertex v_i and triangle t_{ijk}, where $\mathbf{e}_{ij} = \mathbf{x}_j - \mathbf{x}_i$ is the edge vector from vertex v_i to v_j. Following this, there exist several variants to compute the cotangent using this angle. For instance, OpenMesh

and Geogram compute the cotangent as $\cot\alpha_i = 1/\tan\alpha_i$, while VCGLib uses the equivalent expression $\cot\alpha_i = \cos\alpha_i/\sin\alpha_i$. CinoLib, on the other hand, computes the cotangent via the identity $\cot\alpha_i = \tan(\pi/2 - \alpha_i)$.

Extrinsic Computation. Expanding the definitions of the trigonometric functions to vector calculus leads to

$$\cot\alpha_i = \frac{\langle \mathbf{e}_{ij}, \mathbf{e}_{ik} \rangle}{\left\| \mathbf{e}_{ij} \times \mathbf{e}_{ik} \right\|}, \tag{19}$$

which is the definition used by CGAL [The24] and geometry-central [SC*19] to calculate the cotangent weights.

Intrinsic Computation. As the cotangent Laplacian is an intrinsic operator, only depending on edge lengths and connectivity, the above equation can also be rewritten further:

$$\langle \mathbf{e}_{ij}, \mathbf{e}_{ik} \rangle = \frac{a^2 + b^2 - c^2}{2}, \tag{20}$$

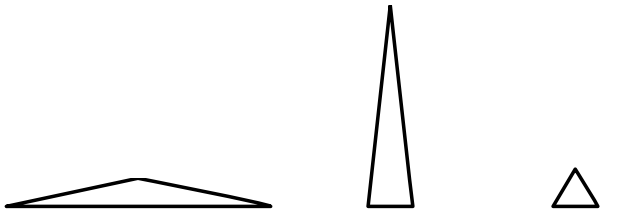
$$\left\| \mathbf{e}_{ij} \times \mathbf{e}_{ik} \right\| = 2|t_{ijk}| = 2\sqrt{s \cdot (s-a) \cdot (s-b) \cdot (s-c)}, \tag{21}$$

with $a = \left\| \mathbf{e}_{ij} \right\|$, $b = \|\mathbf{e}_{ik}\|$, $c = \left\| \mathbf{e}_{jk} \right\|$, and $s = \frac{a+b+c}{2}$. This computation is used by PMP [SB23], libigl [JP*18], and by geometry-central [SC*19] when explicitly using intrinsic meshes. To improve the numerical stability of the area computation, libigl further uses a sorted variant of Heron's formula (see Equation (21)), such that $a \geq b \geq c$ [Kah97].

Clamping. Although most libraries do not clamp any weights, several options exist to remove unwanted entries from the stiffness matrix. CGAL and PMP ignore cotangent computations whenever the area of the corresponding triangle is zero to avoid dividing by zero. This improves stability in the case of truly degenerate faces. In addition, PMP and CinoLib implement another approach, the clamping of negative weights, which is a sufficient condition to guarantee the maximum principle, but might result in the violation of other properties (e.g., linear precision) [WMKG07]. While PMP provides this optionally and per matrix entry, CinoLib clamps each cotangent computation to 10^{-10}, thus resulting in a more conservative and less correct result.

Robustness of Implementations. Wagner and Botsch [WB25] show in their evaluation that not all implementations offer the same amount of robustness to (near-)degenerate geometry. They recommend avoiding trigonometric computations, as they are by far the least robust due to explicitly and non-robustly computing the corner angles. In their tests, extrinsic computations performed better on meshes with many large angles, while intrinsic computations performed better on tessellations with many small angles and mixed unstructured meshes. Overall, there was no clear winner between the extrinsic and intrinsic approaches, but the trigonometric implementations were the clear losers. Furthermore, they recommend avoiding clamping of negative weights, while ignoring (near-)zero-area triangles can be advantageous.

However, all implementations suffer from problems caused by the overall discretization scheme, leading to many failures and high errors. Investigating the numerical properties of the cotangent Laplacian has been a key research interest, with several works proposing error bounds, quality measures [She02, SBMS23], and angle conditions [Zlá68, BA76] to estimate errors based on the shape of the underlying triangles. Generally, triangles with small angles (needles), large angles (caps), and very small areas (see Figure 2) should be avoided, as they are related to stiffness matrix conditioning, gradient error convergence [She02], and ensuring the maximum principle [WMKG07].



(a) *Cap* **(b)** *Needle* **(c)** *Small Area*

Figure 2: *Different problematic elements*

3.3. Robust Laplacians on Triangle Meshes

To improve the robustness of the regular cotangent Laplacian, several modifications have been proposed, related to improving the underlying triangulation [BS07, SC20] or making the Laplacian more robust to (near-)degenerate geometry [QKL*26, WB25].

3.3.1. Intrinsic Delaunay Laplacian

Bobenko and Springborn [BS07] propose building the cotangent Laplacian on the *intrinsic Delaunay triangulation (iDT)*, which guarantees the maximum principle but sacrifices locality in the process. This triangulation is built by flipping edges between triangles that do not satisfy the local Delaunay condition $\alpha_{ij} + \beta_{ij} \leq \pi$, which would result in $w_{ij} < 0$. However, instead of explicitly flipping the edge, which would result in changed geometry due to the linear nature of extrinsic edges, the edges are represented intrinsically, i.e., via edge lengths. For an intrinsic edge flip, the new edge lengths are instead calculated to preserve the original geometry, resulting in piecewise linear edges. Based on these edge lengths, the cotangent Laplacian can be built using Equations (20) and (21). While this formulation does result in a Laplacian without negative weights w_{ij}, it does not necessarily increase the robustness to (near-)degenerate geometry, as triangles satisfying the local Delaunay condition do not have to be well-shaped.

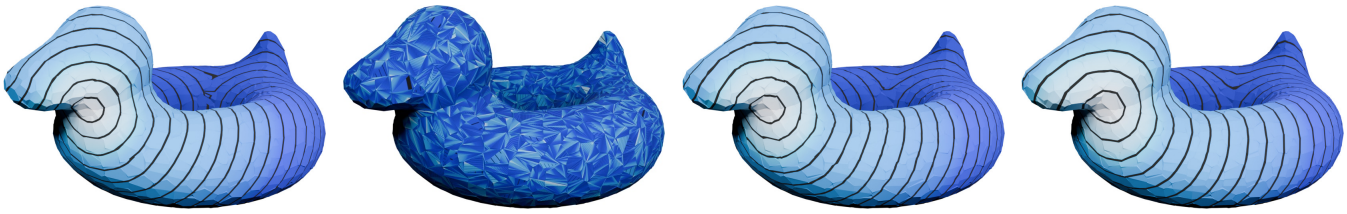
To improve upon this, Sharp and Crane [SC20] propose an *intrinsic mollification* strategy, which increases the length of every edge by a small amount $\varepsilon > 0$ such that the triangle inequality holds with significant inequality, i.e.,

$$\left\| \mathbf{e}_{ij} \right\| + \|\mathbf{e}_{ik}\| > \left\| \mathbf{e}_{jk} \right\| + \delta \tag{22}$$

for all the corners of triangle t_{ijk} and user-defined tolerance $\delta > 0$. As such, they choose

$$\varepsilon = \max_{ijk} \max\left(0, \delta - \left\| \mathbf{e}_{ij} \right\| - \|\mathbf{e}_{ik}\| + \left\| \mathbf{e}_{jk} \right\|\right). \tag{23}$$

This improves the intrinsic quality of the mesh, as both caps and needles only barely fulfill the triangle inequality.



(a) *Exact distances [MMP87]* **(b)** *Standard differential operators* **(c)** *iDT + Molli. [SC20]* **(d)** *D-TFEM [WB25]*

Figure 3: *Comparison of geodesic distances for a model generated with VoroMesh [MKO*23]: (a) the exact geodesic approach [MMP87] results in artifacts; (b–d) show geodesics in heat [CWW13] using (b) the standard differential operators, (c) iDT operators with mollification [BS07, SC20], and (d) D-TFEM [WB25]. While the standard operators fail, iDT with mollification and D-TFEM both obtain good results.*

3.3.2. Tempered Finite Element Method

The *tempered finite element method (TFEM)* by Quiriny et al. [QKL*26] takes a different approach by altering the discretization scheme itself. They argue that the regular cotangent Laplacian creates a locking phenomenon, where bands of caps that are (nearly) degenerate are only able to interpolate linearly across the band, thus losing the convergence property of the operator. To avoid this, they propose to use a mesh-dependent constant $C_{\mathcal{M}} \in \mathbb{R}$ to clamp the Jacobian determinant of the geometric map, which is equal to the double triangle area, e.g., the denominator in Equation (19). This results in the modified computation

$$\frac{\langle \mathbf{e}_{ij}, \mathbf{e}_{ik} \rangle}{\max\left(\left\|\mathbf{e}_{ij} \times \mathbf{e}_{ik}\right\|, C_{\mathcal{M}}\right)}. \tag{24}$$

This constant is chosen in relation to the mean edge length $\bar{h}$ of the mesh, where the authors propose $C_{\mathcal{M}} = C \cdot \bar{h}^3$, with $C \in \mathbb{R}$ being a mesh-independent constant. While this approach is more resilient to degenerate elements, it does not necessarily fulfill the linear precision property [WMKG07], which might be problematic in some applications.

However, Wagner and Botsch [WB25] argue that this scheme does not result in a scaling-invariant formulation, i.e., the clamping does not only depend on the individual triangle shape. They thus propose some modifications, such as using $\bar{h}^2$ instead of $\bar{h}^3$ in the definition of $C_{\mathcal{M}}$ and using the mean edge length $\bar{h}_t$ of the individual triangle $t \in \mathcal{F}$ instead of the mean edge length of the mesh. These changes result in their *D-TFEM* scheme, which only depends on the individual triangle shape, resulting in the modified computation

$$\frac{\langle \mathbf{e}_{ij}, \mathbf{e}_{ik} \rangle}{\max\left(\left\|\mathbf{e}_{ij} \times \mathbf{e}_{ik}\right\|, C_t\right)} \tag{25}$$

with

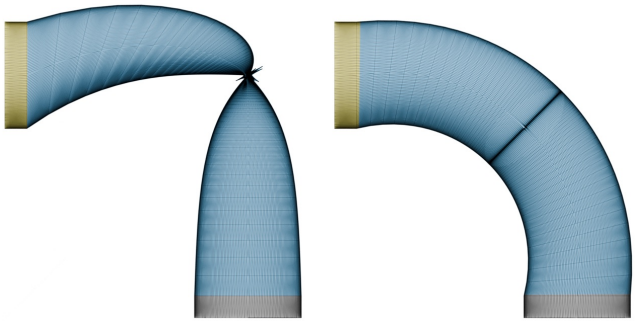
$$C_t = C \cdot \max\left(\bar{h}_t, 10^{-10}\right)^2, \tag{26}$$

where $C = 10^{-3}$. Furthermore, they also temper the mass matrix, as well as the gradient and divergence, by applying the clamping to the triangle areas in the definitions of $\mathbf{M}$, $\mathbf{G}$, and $\hat{\mathbf{M}}$ described in Section 3.

3.3.3. Comparison of Robust Triangle Laplacians

In their evaluation, Wagner and Botsch [WB25] show that all alternative Laplacian definitions exhibit significantly increased robustness to (near-)degenerate geometry, even yielding plausible results for meshes with fully-degenerate triangles. However, all approaches except D-TFEM perform notably worse when solving higher-order PDEs, due to the inversion of the mass matrix.

They furthermore investigate the robustness of the other differential operators, i.e., gradient and divergence, when computing geodesic distances using the geodesics in heat approach of Crane et al. [CWW13] (see Section 8.4), as shown in Figure 3. This qualitative comparison shows that the computation can fail for the standard differential operators on real-world examples, such as a mesh generated via VoroMesh [MKO*23]. While this also results in artifacts for the exact geodesics computation [MMP87], both the operators based on the mollified iDT and D-TFEM yield stable results.



(a) *Standard differential operators* **(b)** *D-TFEM [WB25]*

Figure 4: *Standard differential operators fail for (near-)degenerate elements, shown here for gradient-based bending [YZX*04] of a cylinder with degenerate mid-section (a). In contrast, the differential operators of D-TFEM work robustly (b).*

Lastly, because iDT-based operators are inherently intrinsic, they cannot be applied to applications that require extrinsic operators, such as gradient-based editing [YZX*04]. However, Wagner and Botsch show that their D-TFEM can also be applied to these use cases involving (near-)degenerate geometry, as seen in Figure 4.

3.4. Key Outcomes

In this section, we focused on the standard case of triangle meshes and discussed extensively how to optimize for robustness. There are many ways to compute a discrete Laplacian, and the recommendations of this section can improve results immensely. However, several authors have extended the definition of discrete Laplacians and related operators to more general surface meshes containing arbitrary polygons. These methods and their different mathematical backgrounds will be further explored in the upcoming sections.

4. Mimetic Finite Differences

The Mimetic Finite Difference method (MFD) [LMS14] is an approximation strategy whose main goal is to define discrete differential operators that try to preserve, or *mimic*, certain critical mathematical and physical properties of the underlying PDE. Its core principle lies in the definition of a so-called *primary* operator, typically gradient, divergence or curl, based on discrete vector and tensor calculus and various forms of Stokes' theorem. The other operators are then derived by using discrete analogs of Green's formulas in order to retain a duality relationship to the primary term. Several papers (e.g. [BLS05,BLSS07]) applied the MFD method to derive mimetic discretizations on polygonal and polyhedral meshes and stressed that one of the key components is the definition of an accurate mimetic inner product. This matrix is a vital part in some derivations of the discrete Laplacian. Although the MFD is not directly focused on the construction of this operator, therefore exceeding the scope of this tutorial, its theory influenced recent approaches in the graphics community that will be discussed in the following sections. As a disclaimer, some of the upcoming derivations require rather in-depth knowledge and may seem fast paced for readers that are not already familiar with these terms. However, we still chose to include these detailed explanations in the hope that they might provide some useful insights on the differences of the respective mathematical backgrounds that influenced each of the upcoming discrete polygonal Laplacians.

4.1. Mimetic Polygon Laplacian

Alexa and Wardetzky [AW11] rely on an algebraic approach to define their discrete Laplacian and extend the MFD-based inner product stabilization [BLS05] to two-dimensional manifolds that even allow for non-planar polygons. Given a polygon surface mesh $\mathcal{M}$ embedded in 3D, the only restrictions are that it has to be *oriented*, meaning that two adjacent faces have to be oppositely oriented on the shared edge, and that the faces are *simple*, meaning that they are not self-intersecting and have boundaries that form a closed loop.

4.1.1. Algebraic Framework

Let Γ^k, $k \in \{0,1\}$, be the linear function space of discrete k-forms on $\mathcal{M}$. A k-form can be thought of as a function that takes in k-surfaces and assigns them their integrated value as output, with a 0-surface being a node, a 1-surface an edge, a 2-surface a face and so on. Alexa and Wardetzky derive their polygon Laplacian for 0-forms from the Laplace-de Rahm operator, which for a scalar-valued function u is defined as

$$\Delta u = \mathrm{d}^* \mathrm{d} u. \tag{27}$$

In this context $\mathrm{d} : \Gamma^0 \to \Gamma^1$ is the exterior derivative and $\mathrm{d}^* : \Gamma^1 \to \Gamma^0$ the codifferential, which is defined as the adjoint of d with respect to the square integrable inner product [Ros97]. They use the so-called *coboundary* operator as a discrete version of the smooth exterior derivative, with

$$(\mathrm{d}u)(h_{ij}) = u(j) - u(i) \tag{28}$$

and h_{ij} being the oriented halfedge from vertex v_i to v_j. The definition of a suitable adjoint operator d^* requires inner products on the k-form function spaces and is therefore, in contrast to the exterior derivative, metric dependent. The inner products can be expressed as two symmetric positive definite matrices $\mathbf{M} \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$ and $\mathbf{M}_1 \in \mathbb{R}^{|\mathcal{H}|\times|\mathcal{H}|}$. Any choice of $\mathbf{M}$ and $\mathbf{M}_1$ gives us an expression for the discrete Laplacian

$$\mathbb{L} = \mathrm{d}^* \mathrm{d} = \mathbf{M}^{-1}\mathbf{L} \tag{29}$$

with

$$\mathbf{L} = \mathbf{d}^\mathsf{T}\mathbf{M}_1\mathbf{d}. \tag{30}$$

The matrix version of the coboundary operator $\mathbf{d} \in \mathbb{R}^{|\mathcal{H}|\times|\mathcal{V}|}$ is often referred to as the difference operator. Its k-th row associated with the k-th halfedge $h_{ij} \in \mathcal{H}$ can be expressed as

$$\mathbf{d}_{kl} = \begin{cases} -1 & l = i, \\ 1 & l = j, \\ 0 & \text{otherwise}, \end{cases} \tag{31}$$

which is only non-zero for the entries $\mathbf{d}_{ki}$ and $\mathbf{d}_{kj}$ associated with the vertices connected by the halfedge.

4.1.2. Choice Of Inner Product Matrices

Although in theory any choice for the two inner product matrices would be feasible, not all of them yield the same quality of results. Alexa and Wardetzky therefore motivate their chosen construction by fulfilling the desired criteria discussed in Section 2.1. The inner product matrix for 0-forms assigns each vertex a certain mass. In order to retain locality, the matrix $\mathbf{M}$ is given by

$$\mathbf{M}_{ii} = \sum_{f \ni v_i} \frac{|f|}{n_f}, \tag{32}$$

where $|f|$ denotes the magnitude of the polygons' vector area. As already mentioned, we also consider non-planar polygons in $\mathbb{R}^3$ that do not necessarily define a surface. Therefore, $|f|$ is defined as the area of the largest orthogonal projection of the polygon onto a plane and can be computed as the norm of the Darboux vector $\mathbf{a}_f \in \mathbb{R}^3$ of the skew symmetric (3×3) matrix

$$\mathbf{A}_f = \mathbf{E}_f^\mathsf{T}\mathbf{B}_f, \tag{33}$$

meaning

$$|f| = \|\mathbf{a}_f\| = \left\| \frac{1}{2} \sum_{v_i \in f} \mathbf{x}_i \times \mathbf{x}_{i+1} \right\|. \tag{34}$$

The cyclic vertex positions $(\mathbf{x}_i, \mathbf{x}_{i+1})$ depend on the orientation of the face, which is encoded in the previously defined matrix $\mathbf{X}_f$. It makes sense to look at the definition of the inner product for 1-forms from a local perspective per face and then later assemble the

individual matrices into the global representation, since the process can be repeated per element $f \in \mathcal{F}$. The starting point for the construction is the matrix $\tilde{\mathbf{M}}_f \in \mathbb{R}^{n_f \times n_f}$ given by

$$\tilde{\mathbf{M}}_f = \frac{1}{|f|}\mathbf{B}_f\mathbf{B}_f^\mathsf{T}, \tag{35}$$

which was previously defined by Brezzi et al. [BLS05] and is motivated by the Laplacian's connection to mean curvatures. However, while this choice of inner product matrix is generally positive semi-definite, in order for the Laplacian itself to fulfill this property, which is a desired criterion, the inner products have to be *positive* definite. Alexa and Wardetzky therefore add a stabilization term to extend Brezzi et al.'s definition to non-planar polygons and give rise to a positive definite inner product. The necessity stems from the fact that for general polygons with n_f vertices the transposed midpoint matrix $\mathbf{B}_f$ will have either rank 2 (planar) or 3 (non-planar), allowing for a kernel of dimension $n_f - \text{rank}(\mathbf{B}_f)$. Therefore, in order to fill up the kernel, Alexa and Wardetzky introduce the alternative inner product matrix

$$\mathbf{M}_f := \tilde{\mathbf{M}}_f + \mathbf{C}_{\bar{f}}\mathbf{U}_{\bar{f}}\mathbf{C}_{\bar{f}}^\mathsf{T}. \tag{36}$$

Here, $\bar{f}$ is the maximum orthogonal projection of the polygon f and $\mathbf{C}_{\bar{f}} \in \mathbb{R}^{n_f \times (n_f-2)}$ is a matrix whose columns span the kernel of $\mathbf{E}_{\bar{f}}^\mathsf{T}$. Combined with any choice of a symmetric positive definite matrix $\mathbf{U}_{\bar{f}} \in \mathbb{R}^{n_f \times n_f}$, the stabilization term will lead to a positive definite inner product $\mathbf{M}_f$, as proven in Theorem 1 of the original paper. That $\mathbf{C}_{\bar{f}}$ only has to span the kernel of $\mathbf{E}_{\bar{f}}^\mathsf{T}$ is motivated by the linear precision property. In order for $(\mathbf{LX})_i$ to vanish in a planar region surrounding vertex v_i, the stabilization term must also vanish. But since $\mathbf{E}_f = \mathbf{E}_{\bar{f}}$ for planar polygons, we get

$$\left(\mathbf{C}_f^\mathsf{T}\mathbf{d}_f\mathbf{X}_f\right)_i = \left(\mathbf{C}_f^\mathsf{T}\mathbf{E}_f\right)_i = \left(\mathbf{C}_f^\mathsf{T}\mathbf{E}_{\bar{f}}\right)_i \overset{!}{=} 0, \tag{37}$$

which is equivalent to

$$\left(\mathbf{E}_{\bar{f}}^\mathsf{T}\mathbf{C}_f\right)_i \overset{!}{=} 0. \tag{38}$$

Here, $\mathbf{d}_f$ refers to the local difference operator defined on the face f. Since all other properties are already accounted for, it is sufficient to require that $\mathbf{C}_f$ spans the kernel of $\mathbf{E}_{\bar{f}}^\mathsf{T}$. As for inner products in general, there are several choices for $\mathbf{C}_{\bar{f}}$ and $\mathbf{U}_{\bar{f}}$ that would satisfy the conditions, giving rise to a whole family of suitable matrices. However, Alexa and Wardetzky propose a special combination in order to achieve scale invariance as a property for the final Laplacian, meaning that the stiffness matrix $\mathbf{L}$ remains unchanged when the mesh is uniformly scaled. Using a parameter $0 < \lambda \in \mathbb{R}$, they choose the matrix $\mathbf{U}_{\bar{f}}$ as

$$\mathbf{U}_{\bar{f}} := \lambda\mathbf{I}_f, \tag{39}$$

with $\mathbf{I}_f$ being the n_f-dimensional identity matrix. They choose $\mathbf{C}_{\bar{f}}$ such that its columns are orthonormal, and the final inner product leads to a per-face Laplacian stiffness matrix

$$\mathbf{L}_f = \mathbf{d}_f^\mathsf{T}\mathbf{M}_f\mathbf{d}_f \tag{40}$$

that is not affected by scaling, is local and linearly precise. These local matrices are then assembled into the global stiffness matrix $\mathbf{L}$ by assigning each vertex v_i the i-th row and column of $\mathbf{L}$ in which the sum over their respective entries in the local matrices are collected.

4.2. Geometric Polygon Laplacian

Alexa and Wardetzky's focus lies solely on the definition of the Laplace-Beltrami and did not further investigate other operators. This was later addressed by de Goes et al. [dGBD20], who defined a variety of discrete differential polygon operators that also serve as a generalization of the MFD, but with a stabilization term for the inner product matrix on 1-forms inspired by the virtual element method (VEM) [BdVBM13]. The following section will further elaborate on their approach. The main focus of de Goes et al. [dGBD20] was a new linearly precise discretization of the gradient, which allows to define a consistent set of operators, including their own interpretation of the Laplacian.

4.2.1. Polygon Gradient

As in the previous section, the definition of the gradient will be applied locally per polygon $f \in \mathcal{F}$, but can be assembled into a global gradient matrix acting on the complete mesh. Given a scalar function u defined on f, we want to find a matrix $\mathbf{G}_f$ that simulates the behavior of the gradient ∇u on the polygon. For planar elements, this would normally be achieved by applying Stokes' theorem to ∇u and deriving a matrix discretization through the weak form of the resulting boundary integral. However, since the polygons of the given mesh are not necessarily planar it is not clear how to define the surface normal $\mathbf{n}(\mathbf{x})$ at the boundary points $\mathbf{x}$. Therefore, the standard approach cannot be used. De Goes et al. [dGBD20] circumvent this problem by evaluating the co-gradient operator

$$\nabla u^\perp(\mathbf{x}) := \mathbf{n}(\mathbf{x}) \times \nabla u(\mathbf{x}), \tag{41}$$

on which applying Stoke's theorem leads to

$$\iint_f \nabla u^\perp(\mathbf{x})\,\mathrm{d}A = \oint_{\partial f} u(\mathbf{x})\mathbf{t}(\mathbf{x})\,\mathrm{d}\mathbf{x}, \tag{42}$$

with $\mathbf{t}(\mathbf{x})$ being the unit tangent vector at boundary point $\mathbf{x}$. This expression is independent of the surface of the polygon and only requires the tangent vectors along the boundary, which are uniquely defined. For example, if we consider the boundary integral in Equation (42) for linear functions u, the integrated co-gradient can be evaluated exactly as a sum over the averaged function values along the polygon edges, multiplied with the respective edge vector:

$$\oint_{\partial f} u(\mathbf{x})\mathbf{t}(\mathbf{x})\,\mathrm{d}\mathbf{x} = \mathbf{E}_f^\mathsf{T}\mathbf{Avg}_f\mathbf{u}_f. \tag{43}$$

Here $\mathbf{Avg}_f \in \mathbb{R}^{n_f \times n_f}$ is a matrix that yields the average of consecutive vector entries, defined as

$$\left(\mathbf{Avg}_f\right)_{ij} = \begin{cases} \frac{1}{2} & \text{for } j = i \\ \frac{1}{2} & \text{for } j = (i+1) \bmod n_f \\ 0 & \text{otherwise.} \end{cases} \tag{44}$$

Additionally, in order to describe the co-gradient as matrix-vector multiplication, one can define the cross product as a map from a 3D

vector $\mathbf{p} = (p_1, p_2, p_3)^\mathsf{T}$ to a skew symmetric matrix $[\mathbf{p}] \in \mathbb{R}^{3\times3}$ with

$$[\mathbf{p}] = \begin{pmatrix} 0 & -p_3 & p_2 \\ p_3 & 0 & -p_1 \\ -p_2 & p_1 & 0 \end{pmatrix}, \tag{45}$$

such that $[\mathbf{p}]\mathbf{q} = \mathbf{p} \times \mathbf{q}$ for $\mathbf{q} \in \mathbb{R}^3$. If we consider $u : \mathbb{R}^3 \to \mathbb{R}$ to be a linear function, meaning $u(\mathbf{x}) = \mathbf{s}^\mathsf{T}\mathbf{x} + r$ with $\mathbf{s}, \mathbf{x} \in \mathbb{R}^3$ and $r \in \mathbb{R}$, its co-gradient would yield $\mathbf{n}(\mathbf{x}) \times \mathbf{s}$, which can then be expressed as $[\mathbf{n}(\mathbf{x})]\mathbf{s}$. This becomes useful if we consider

$$\iint_f \nabla u^\perp(\mathbf{x})\,\mathrm{d}\mathbf{x} = \oint_{\partial f} \left(\mathbf{s}^\mathsf{T}\mathbf{x} + r\right)\mathbf{t}(\mathbf{x})\,\mathrm{d}\mathbf{x} \tag{46}$$

$$= \mathbf{E}_f^\mathsf{T}\mathbf{A}\mathbf{v}\mathbf{g}_f\left(\mathbf{X}_f\mathbf{s} + \mathbf{1}_f r\right) \tag{47}$$

$$= \mathbf{E}_f^\mathsf{T}\mathbf{B}_f\mathbf{s}, \tag{48}$$

which equally gives

$$\mathbf{E}_f^\mathsf{T}\mathbf{B}_f\mathbf{s} = \left(\iint_f [\mathbf{n}(\mathbf{x})]\,\mathrm{d}\mathbf{x}\right)\mathbf{s}, \tag{49}$$

since the term $\mathbf{E}_f^\mathsf{T}\mathbf{A}\mathbf{v}\mathbf{g}_f\mathbf{1}_f = \mathbf{E}_f^\mathsf{T}\mathbf{1}_f$, which is the sum of edge vectors and therefore zero. Equation (49) implies that the surface integral of the matrix $[\mathbf{n}(\mathbf{x})]$ is equal to the previously mentioned area matrix $\mathbf{A}_f$ (see Equation (33)) related to the largest orthogonal projection of the face f and independent of the polygons' interpolated surface. Therefore, replacing the normal term $\mathbf{n}(\mathbf{x})$ with the constant face normal of the planar projection $\bar{f}$, given by

$$\mathbf{n}_f = \frac{\mathbf{a}_f}{|f|}, \tag{50}$$

the surface integral of the co-gradient can be changed to

$$\iint_f [\mathbf{n}_f]\nabla u(\mathbf{x})\,\mathrm{d}\mathbf{x}. \tag{51}$$

Considering that the co-gradient $\nabla u^\perp$ can now be interpreted as a local 90 degrees rotation of the gradient around the normal $\mathbf{n}_f$, one can apply a second rotation around the same normal to obtain $-\nabla u$. This leads to de Goes et al.'s definition of the the gradient matrix

$$\mathbf{G}_f = -\frac{1}{|f|}[\mathbf{n}_f]\mathbf{E}_f^\mathsf{T}\mathbf{A}\mathbf{v}\mathbf{g}_f \tag{52}$$

per polygon f, which is proven to be linearly precise. As for the stiffness matrix, the local gradient operators can be assembled into a global gradient operator per mesh $\mathbf{G} \in \mathbb{R}^{3|\mathcal{F}|\times|\mathcal{V}|}$.

4.2.2. Flat, Sharp And Projection Operator

Based on their definition of the gradient operator, de Goes et al. derive an alternative expression to Alexa's and Wardetzky's choice for the inner product matrix on 1-forms. Involved in the process are their discrete polygon extensions of the so-called *sharp* $\sharp$ and *flat* $\flat$ operators, both discretized as

$$\mathbf{V}_f^\sharp := \frac{1}{|f|}[\mathbf{n}_f]\left(\mathbf{B}_f^\mathsf{T} - \mathbf{c}_f\mathbf{1}_f^\mathsf{T}\right) \in \mathbb{R}^{3\times n_f}, \tag{53}$$

and

$$\mathbf{V}_f^\flat := \mathbf{E}_f\left(\mathbf{I} - \mathbf{n}_f\mathbf{n}_f^\mathsf{T}\right) \in \mathbb{R}^{n_f\times3} \tag{54}$$

respectively, with $\mathbf{c}_f$ being the face centroid and $\mathbf{1}_f \in \mathbb{R}^{n_f}$ a vector with only ones as entries. As pointed out in Lemma 2 of the original paper, this expression for the sharp operator yields

$$\mathbf{G}_f\mathbf{u}_f = \mathbf{V}_f^\sharp\mathbf{d}_f\mathbf{u}_f, \tag{55}$$

for any scalar function $\mathbf{u}_f$ and is therefore able to reproduce a discrete version of the smooth relation $\nabla u = (\mathrm{d}u)^\sharp$ between sharp and gradient operator. In the continuous setting, given a vector space V equipped with an inner product $\langle\cdot,\cdot\rangle$ represented by a matrix $\mathbf{K}$:

$$\langle\mathbf{x},\mathbf{y}\rangle = \mathbf{x}^\mathsf{T}\mathbf{K}\,\mathbf{y} \quad \forall\,\mathbf{x},\mathbf{y} \in V, \tag{56}$$

the flat operator $\flat$ is defined as

$$\mathbf{v}^\flat(\mathbf{u}) = \langle\mathbf{v},\mathbf{u}\rangle = \mathbf{v}^\mathsf{T}\mathbf{K}\,\mathbf{u}, \tag{57}$$

mapping a vector $\mathbf{v}$ from the vector space V to a functional $g(\mathbf{u}) = \langle\mathbf{v},\mathbf{u}\rangle$ in its dual space V^* [Lee97]. The flat and sharp operators form an isomorphism and we know from the previous assumption that for each linear functional $g \in \mathcal{V}^*$ there exists a unique $\mathbf{v} \in V$ such that $g(\mathbf{u}) = \langle\mathbf{v},\mathbf{u}\rangle$ for all $\mathbf{u} \in V$. Therefore, the sharp operator $\sharp$ forms the inverse of $\flat$ and can be considered as $g^\sharp = \mathbf{v}$. This is commonly referred to as *lowering* or *raising* an index. In de Goes' setting, the matrix $\mathbf{V}^\flat$ maps a 3D vector to its tangential part and then computes its counter-clockwise circulation along the edges of the polygon, giving us a discrete 1-form. The sharp operator $\mathbf{V}^\sharp$ inversely maps the values of a discrete 1-form defined on the polygon back to a single tangent vector per face. However, in contrast to the continuous setting, the operators defined by de Goes et al. are not the exact inverses of each other. The definition holds for any vector $\mathbf{v} \in \mathbb{R}^3$ that satisfies $\mathbf{v}^\mathsf{T}\mathbf{n}_f = 0$, meaning it is tangent to the polygon. If this vector is flattened to a 1-form and then inversely sharpened back, it regains its original form. However, first applying the sharpening operator and then lowering the resulting tangent vector back to its 1-form does not have the same effect due to the rank deficiency of $\mathbf{V}_f^\sharp$. It reduces the n_f values of a discrete 1-form to only a tangent vector on the polygon f [dGBD20] and therefore loses information. Inspired by the virtual element method [BdVBM13], they mitigate the effect by defining a so-called *projection* operator

$$\mathbf{P}_f := \mathbf{I}_f - \mathbf{V}_f^\flat\mathbf{V}_f^\sharp \quad \in \mathbb{R}^{n_f\times n_f} \tag{58}$$

that measures the error of $\mathbf{V}_f^\flat$ and $\mathbf{V}_f^\sharp$ being inverse to each other. Basically, by first sharpening a 1-form g to a tangent vector that is then flattened back to a representative 1-form $\hat{g}$, the projection operator eliminates the components of g that would result in a tangent vector after applying $\mathbf{V}_f^\sharp$.

4.2.3. Stiffness Matrix

Equipped with the previously defined operators, de Goes et al. [dGBD20] define their local inner product matrices acting on 1-forms as

$$\mathbf{M}_f := |f|\,\mathbf{V}_f^{\sharp\mathsf{T}}\mathbf{V}_f^\sharp + \lambda\,\mathbf{P}_f^\mathsf{T}\mathbf{P}_f, \tag{59}$$

which can be assembled into the global inner product matrix $\mathbf{M}_1$ acting on the whole mesh. The matrix $\mathbf{M}_f$ maps the involved 1-forms to their respective tangential vectors with the help of the sharpening operator $\mathbf{V}_f^\sharp$, resulting in their dot product. The potential rank deficiency is mitigated through the second correction term

regulated by a parameter $\lambda > 0$. As for Alexa and Wardetzky, this regularization is necessary to guarantee that the inner product matrix is strictly positive definite, which can then be used as before to define the local discrete Laplace-Beltrami operator

$$\mathbf{L}_f = \mathbf{d}_f^\mathsf{T} \mathbf{M}_f \mathbf{d}_f. \tag{60}$$

4.3. Gradient and Divergence Operator

Since the Laplacian of a function u is defined as the divergence of the gradient of said function

$$\Delta u = \operatorname{div}(\nabla u), \tag{61}$$

its discretization should be able to simulate the same behavior. This means that for each gradient matrix $\mathbf{G}_f \in \mathbb{R}^{3\times n_f}$ defined on a polygon f, there should exist a respective divergence matrix $\mathbf{D}_f \in \mathbb{R}^{n_f\times 3}$ with

$$\mathbf{L}_f = \mathbf{D}_f \mathbf{G}_f. \tag{62}$$

Typically, this divergence operator is defined as the adjoint of the gradient scaled with a mass matrix containing the faces areas. However, a possibility for both presented approaches [AW11, dGBD20] to satisfy Equation (62) would be to follow the discrete exterior calculus interpretation [DHLM05] of the operators. Here, the difference matrix $\mathbf{d}_f$ can be seen as a gradient operator acting on 0-forms and the divergence of a discrete 1-form on f is then defined as

$$\mathbf{D}_f := \mathbf{d}_f^\mathsf{T} \mathbf{M}_f, \tag{63}$$

which gives the required equality. Note that this interpretation of the gradient differs from de Goes' geometric operator in Equation (52). In fact, given that their sharp operator satisfies $\mathbf{V}_f^\sharp \mathbf{d}_f = \mathbf{G}_f$, we can interpret their $\mathbf{L}_f$ as

$$|f|\,\mathbf{G}_f^\mathsf{T}\mathbf{G}_f + \lambda\,\mathbf{d}_f \mathbf{P}_f^\mathsf{T} \mathbf{P}_f \mathbf{d}_f, \tag{64}$$

which means that their stiffness matrix is composed from the "traditional" interpretation of gradient and divergence, combined with the scaled stabilization term. This is reflected in the second part of their paper, where they define the divergence of face-based vector fields as the trace of the covariant derivative. Given a vector $\mathbf{u}_f$ associated with a face f, its divergence actually simplifies to

$$\mathbf{D}_f \mathbf{u}_f = |f|\,\mathbf{G}_f^\mathsf{T} \mathbf{u}_f, \tag{65}$$

leading to two interpretations of the divergence operator, one acting on 1-forms and the other on vector fields.

4.4. Differences between Inner Product Matrices

The previously presented Laplacians are closely related in their definitions of the inner product matrix for 1-forms. We will therefore shortly highlight some of the similarities and differences of the respective matrices. We already established that Alexa and Wardetzky's matrix $\tilde{\mathbf{M}}_f$ follows Brezzi et al.'s construction. $\tilde{\mathbf{M}}_f$ is dependent on the choice of origin if regarded individually, however, combined with the difference matrix $\mathbf{d}_f$ and its adjoint $\mathbf{d}_f^\mathsf{T}$, this dependency vanishes. De Goes et al.'s equivalent eliminates this dependency immediately by regarding the midpoints relative to the respective centroid of the face as $\mathbf{C}_f \in \mathbb{R}^{n_f\times 3}$, with row entries

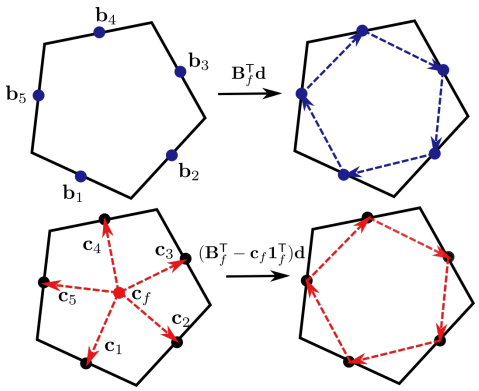


Figure 5: *Vectors involved in the inner product matrix for 1-forms for both Alexa and Wardetzky [AW11] and de Goes et al. [dGBD20] if computed on a planar polygon.*

$\mathbf{c}_i = \mathbf{b}_i - \mathbf{c}_f$. However, given a planar face, combining the matrices $\mathbf{B}_f^\mathsf{T}$ and $\mathbf{C}_f^\mathsf{T}$ with $\mathbf{d}_f$ actually yields the same result, as visualized in Figure 5. Without the respective stabilization terms, both methods would lead to identical inner products since the remaining part of de Goes et al.'s method, namely

$$-[\mathbf{n}_f]^2 = \left(\mathbf{I} - \mathbf{n}_f \mathbf{n}_f^\mathsf{T}\right), \tag{66}$$

has no effect in this situation. Therefore, at least for planar polygons, the biggest difference between the inner product matrices are the stabilization terms.

4.5. Key Outcomes

For the computer graphics community, the main achievement of the two presented polygon Laplacians is to extend the MFD-based inner product stabilization from Brezzi et al. [BLS05] to possible non-planar two manifolds embedded in 3D. While the individual mathematical derivations of the operators differ, they both introduce additional weighted stabilization terms in order to guarantee the crucial requirement of strict positive definiteness of the inner product matrices.

5. Finite Element Discretizations

The finite element method (FEM) is often used to approximate the solution u to a given PDE on a simplicial mesh with the help of a finite set of basis functions. The exact number depends on both the shape of the element and the order of the basis itself. In the linear case, we typically associate an individual shape function φ_i with the vertex $\mathbf{x}_i$, also commonly referred to as node. Now, instead of solving the PDE directly, the objective changes to finding suitable coefficients u_i, $i = 1,\dots,|\mathcal{V}|$, that approximate the unknown solution u of the PDE with

$$u(\mathbf{x}) = \sum_{i=1}^{|\mathcal{V}|} u_i \varphi_i(\mathbf{x}). \tag{67}$$

For example, a common problem solved with the finite element method is the Poisson equation $-\Delta u = f$ for a known function f.

Given a surface mesh, the discretized PDE leads to a linear system $\mathbf{Lu} = \mathbf{f}$ with a Laplace matrix $\mathbf{L}$ that is defined as the integrated dot product of the gradients of the basis functions:

$$\mathbf{L}_{ij} = \int_{\mathcal{M}} \langle \nabla \varphi_i, \nabla \varphi_j \rangle. \tag{68}$$

While a variety of different bases can be used to solve this system, for triangle meshes we focus on linear nodal shape functions that are defined piecewise per face and satisfy the Lagrange interpolation property already mentioned in Equation (10):

$$\varphi_i(\mathbf{x}_j) = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{otherwise.} \end{cases} \tag{69}$$

Furthermore, we want them to satisfy additional properties within each element of the mesh in order to guarantee convergence under refinement [Hug12]:

1. They have to be C^1 continuous within the element and C^0 across its boundaries.
2. For the basis to satisfy constant precision, the φ_i have to form a partition of unity
$$\sum_{i=1}^{n_f} \varphi_i(\mathbf{x}) = 1. \tag{70}$$
3. They have to fulfill the linear reproduction property
$$\sum_{i=1}^{n_f} \varphi_i(\mathbf{x}) \mathbf{x}_i = \mathbf{x} \tag{71}$$
on planar polyons.

A standard set of basis functions meeting all these requirements are the piecewise linear hat functions on triangle meshes, also known as barycentric coordinates. For general polygons, there exist a variety of *generalized* barycentric coordinates (GBC) [Flo03, JSW05, JMD*07, HS08, Bis14], which are based on the idea to express any point within the polygon as weighted sum over its boundary nodes. This defines local shape functions that can be used in the finite element analysis. Extensive surveys [Flo15, CG16] have already discussed the benefits and properties of these shape functions.

5.1. Harmonic Coordinates

Since this report is more focused on the explicit construction of a Laplacian operator, we will not discuss shape functions based on GBC in the same depth, but rather explain one representative case, named the harmonic coordinates. While other methods like the maximum entropy coordinates [HS08] are very present in the FEM analysis on polytopes, we still chose the harmonic shape functions due to their numerous natural mathematical properties that makes them so well suited for FEM. This includes smoothness, non-negativity, the mean-value property and minimization of the Dirichlet energy [MKB*08, CG16]. They can also be analyzed on arbitrary convex and non-convex polygons [WBG07]. The only real drawbacks of these shape functions are them not having a closed form, and therefore requiring costly numerical integration, and that they are only defined on planar elements. In this section, we will review the construction of polygonal finite element shape functions based on the work of Joshi et al. [JMD*07] and Martin et al. [MKB*08]. As with the other methods, the properties of the harmonic coordinates will be further described in Section 7.

5.1.1. Harmonic Shape Functions on Planar Polygon Meshes

Given a mesh $\mathcal{M}$ consisting of arbitrary *planar* polygons $\mathcal{F}$, shape functions $\varphi_i^f : f \to \mathbb{R}$, defined on a polygon $f \in \mathcal{F}$, are called *harmonic* if they satisfy $\Delta \varphi_i^f = 0$. In this case, they can be uniquely determined by specifying their function values b_i along the edges of the polygon as Dirichlet boundary conditions:

$$\begin{aligned} \Delta \varphi_i^f(\mathbf{x}) &= 0 \quad \text{for } \mathbf{x} \in f, \\ \varphi_i^f(\mathbf{x}) &= b_i(\mathbf{x}) \quad \text{for } \mathbf{x} \in \partial f. \end{aligned} \tag{72}$$

In the linear case, the required Lagrange interpolation property and C^0 continuity can be guaranteed by linearly interpolating the nodal values of φ_i^f along the boundary of the face. However, for polygons it is not possible to find a closed form for these shape functions and they have to be approximated numerically. This is why Martin et al. propose a scheme based on the method of fundamental solutions (MFS) [FK98] to determine the harmonic shape functions, although other methods are equally applicable. The core principle of MFS is to use an analytic fundamental solution ψ of the respective PDE, in our case the Laplace equation (72), and approximate the sought solution through a linear combination of ψ centered at different source points $\{\mathbf{k}_1, \ldots, \mathbf{k}_n\}$ of the ambient Euclidean space. In our case, the fundamental solution to the 2D Laplace equation would be the radial basis function

$$\psi(\|\mathbf{x}\|) = \log(\|\mathbf{x}\|), \tag{73}$$

which is well defined in $\mathbb{R}^2$ except for one singularity at the origin. Translating this function to the previously chosen source points, we can approximate shape functions ϕ_i^f with

$$\phi_i^f(\mathbf{x}) = \sum_{j=1}^{n} w_{ij} \psi\left(\left\|\mathbf{x} - \mathbf{k}_j\right\|\right), \tag{74}$$

which are then harmonic by construction.

The $\psi_j(\mathbf{x}) := \psi(\left\|\mathbf{x} - \mathbf{k}_j\right\|)$ are also often referred to as *kernels* and their centers $\mathbf{k}_j$ have to be placed outside of the face's domain (see Figure 6) due to the singularity at $\mathbf{x} = \mathbf{k}_j$. Martin et al. suggest a number of 3–5 kernels per edge distributed by a uniform sampling. However, in the current state, shape functions ϕ_i^f approximated via Equation (74) would not be able to exactly reproduce linear functions, violating the linear precision property. Therefore, Martin et al. add a linear polynomial

$$\varphi_i^f(\mathbf{x}) = \sum_{j=1}^{n} w_{ij} \psi(||\mathbf{x} - \mathbf{k}_j||) + \mathbf{s}_i^{\mathsf{T}} \mathbf{x} + r_i \tag{75}$$

to guarantee that the function space spanned by the shape functions contains all linear functions. Linear polynomials are always harmonic, so φ_i^f still satisfies Equation (72). In order to approximate the Dirichlet boundary constraints, we select $m = 3n$ uniformly sampled collocation points $\mathbf{c}_i$ along the edges e_{kl} of the polygon to minimize the discretized boundary integral over the L_2 error

$$\int_{\partial f} \left(\varphi_i^f(\mathbf{x}) - b_i(\mathbf{x})\right)^2 \approx \frac{1}{m} \sum_{j=1}^{m} \left(\varphi_i^f(\mathbf{c}_j) - b_i(\mathbf{c}_j)\right)^2. \tag{76}$$

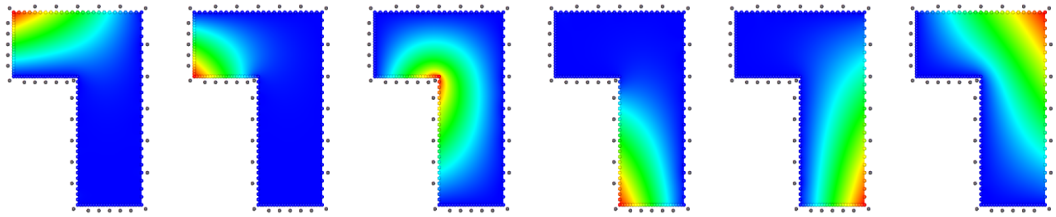


Figure 6: *Harmonic shape functions associated with the 6 nodes of a planar L-shaped polygon. The larger dots with the slight offset are the sampled center points* $\mathbf{k}_i$ *and the small dots on the boundary denote the collocation points* $\mathbf{c}_j$ *for Dirichlet boundary constraints. The figure was taken from [MKB*08].*

This can be solved with the help of the following linear system

$$\begin{pmatrix} \psi_{11} & \cdots & \psi_{1n} & \mathbf{c}_1^\top & 1 \\ \vdots & & \vdots & \vdots & \vdots \\ \psi_{m1} & \cdots & \psi_{mm} & \mathbf{c}_m^\top & 1 \end{pmatrix} \begin{pmatrix} w_{i1} \\ \vdots \\ w_{in} \\ \mathbf{s}_i \\ r_i \end{pmatrix} = \begin{pmatrix} b_i(\mathbf{c}_1) \\ \vdots \\ b_i(\mathbf{c}_m) \end{pmatrix}, \tag{77}$$

where $\psi_{ij} = \psi(||\mathbf{c}_i - \mathbf{k}_j||)$ and $b_i(\mathbf{c}_j)$ contains the function value of the respective basis function at this point. Since the system is overdetermined ($m > n+3$), it has to be solved for the least-squares solution. Martin et al. recommend to use a QR factorization or the SVD pseudo inverse [GVL96].

5.1.2. Stiffness and Mass matrix

Equipped with the shape functions described in the previous section, we are now able to express sought solutions of a PDE with the FEM interpolation scheme as described in Equation (67). The discretizations of both stiffness and mass matrix needed for the Laplacian can be obtained through

$$(\mathbf{L}_f)_{ij} = \int_f \langle \nabla \varphi_i^f, \nabla \varphi_j^f \rangle \, \mathrm{d}\mathbf{x}. \tag{78}$$

and

$$(\mathbf{M}_f)_{ij} = \int_f \varphi_i^f \cdot \varphi_j^f \, \mathrm{d}\mathbf{x}. \tag{79}$$

Note that this process requires numerical integration, since the gradients of the harmonic shape functions are not constant.

5.2. Linear Virtual Refinement Method

Given the problems one has to deal with while working on general polygons, a rather pragmatic solution would be to simply refine the mesh into triangles. However, this can potentially break intended symmetry structures of the original tessellation and increase the dimension of linear systems if new vertices have to be added. The method presented by Bunge et al. [BHKB20] took inspiration from the simplicity of the triangle refinement, but proposed an in-between approach that avoids the downsides of an explicit refinement of the mesh.

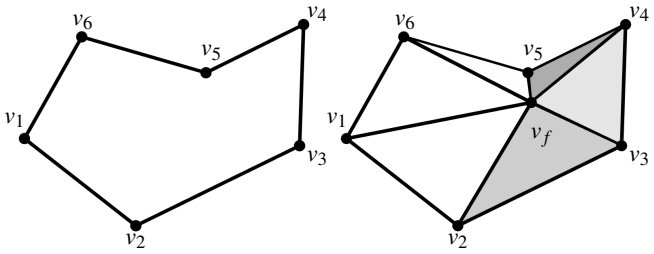


Figure 7: *Spanned triangle fan on the virtual mesh after inserting the vertex* v_f.

Given a polygon mesh $\mathcal{M}$, they introduce *virtual* vertices v_f into each (not necessarily planar) polygon f. These are expressed as affine combinations of the original faces' vertex positions

$$\mathbf{x}_f = \sum_{v_i \in f} w_i \, \mathbf{x}_i, \quad \text{with} \sum_{v_i \in f} w_i = 1. \tag{80}$$

The additional vertices allow Bunge et al. to construct a virtual triangle mesh $\mathcal{M}_\triangle$ by dividing each face into a triangle fan as shown in Figure 7. On this mesh, standard approaches like the cotan Laplacian (see Section 3) can be easily computed. However, in order to define operators working on the original mesh, Bunge et al. redistribute the values at the virtual vertices back to their associated polygon nodes. This is achieved by combining the affine weights $\mathbf{w}_f = (w_1, \ldots, w_{n_f})$ of each face into a local $(n_f+1) \times n_f$ prolongation matrix

$$\mathbf{P}_{ij}^f = \begin{cases} w_j & \text{for } i = n_f + 1 \\ \delta_{ij} & \text{otherwise,} \end{cases} \tag{81}$$

which can be assembled into a global matrix $\mathbf{P} \in \mathbb{R}^{(|\mathcal{V}|+|\mathcal{F}|)\times|\mathcal{V}|}$

$$\mathbf{P}_{ij} = \begin{cases} 1 & \text{if } i = j \text{ and } i \leq |\mathcal{V}| \\ w_{kj} & \text{if } i = |\mathcal{V}| + k \text{ and } v_j \in f_k \\ 0 & \text{otherwise,} \end{cases} \tag{82}$$

acting on the whole mesh. Using this matrix leads to a very easy refinement and coarsening process that allows Bunge et al. to define a polygon Laplacian, gradient and divergence operator on the original mesh: Given the global prolongation matrix $\mathbf{P}$, they construct the cotangent mass and stiffness matrices $\mathbf{M}_\triangle$ and $\mathbf{L}_\triangle$ on the

virtual triangle mesh $\mathcal{M}_\triangle$ (see Equations (11) and (12)) and define the matrices for the original polygon mesh as

$$\mathbf{L} = \mathbf{P}^\mathsf{T}\mathbf{L}_\triangle\mathbf{P} \tag{83}$$

and

$$\mathbf{M} = \mathbf{P}^\mathsf{T}\mathbf{M}_\triangle\mathbf{P}. \tag{84}$$

As for the Laplacian, they compute the gradient and divergence operators $\mathbf{G}_\triangle$ and $\mathbf{D}_\triangle$ on $\mathcal{M}_\triangle$ by using the simplicial definitions in Equations (14) and (15) to express the polygon operators as

$$\mathbf{G} = \mathbf{G}_\triangle\mathbf{P} \tag{85}$$

and

$$\mathbf{D} = \mathbf{P}^\mathsf{T}\mathbf{D}_\triangle = \mathbf{P}^\mathsf{T}\mathbf{G}_\triangle^\mathsf{T}\hat{\mathbf{M}}_\triangle. \tag{86}$$

Combining both polygon gradient and divergence leads once again to the stiffness matrix $\mathbf{L}$ and is therefore consistent with the previous discretization. The remaining question addressed by Bunge et al. [BHKB20] is the placement of the virtual vertex. Given that positions outside of the (planar) polygon's boundary would lead to flipped virtual triangles with bad numerical properties, they suggest the virtual vertex to be the unique minimizer of the sum of squared triangle areas of the refined face. This is motivated by the fact that for a planar star-shaped polygon the point almost always lies within the kernel of the polygon, leading to virtual triangles with positive areas. Finding the vertex position can be directly expressed as minimization problem over the weight vector $\mathbf{w}_f = (w_1, \dots, w_{n_f})^\mathsf{T} \in \mathbb{R}^{n_f}$ with

$$\mathbf{w}_f = \arg\min_{\mathbf{w}} \sum_{i=1}^{n_f} \text{area}\left(\mathbf{x}_i, \mathbf{x}_{i+1}, \sum_{j=1}^{n_f} w_j\,\mathbf{x}_j\right)^2 \tag{87}$$

$$\text{such that } \sum_{j=1}^{n_f} w_j = 1. \tag{88}$$

However, for faces with valence higher than three, this system is under-constrained and several sets of weights are able to represent the point. The authors therefore add the constraint that the weight vector should have minimal L_2 norm, which leads to a unique solution that encourages a more uniform distribution among the weights and can be solved with a linear system.

While these weights are simple to compute, they do not necessarily yield a well-shaped virtual triangle fan. To this end, Bunge et al. [BBW*24] propose an alternative vertex position and corresponding weights that minimize the *harmonic index* [Mus97, BS07], as a lower harmonic index can be linked to better-shaped triangles and more favorable stiffness matrix conditioning. However, it requires an iterative optimization strategy for the virtual vertex, which is less performant and more complex to implement.

Furthermore, both approaches still suffer from the regular cotangent Laplacian discretization, as (near-)degenerate polygons do not allow well-shaped triangle fans, independent of the virtual vertex. This can be alleviated by also applying the D-TFEM scheme (see Section 3.3.2), as demonstrated by Wagner and Botsch [WB25]. They show that even completely degenerate triangles in the virtual triangle fan can still yield decent results.

5.2.1. Finite Element Shape Functions

Drawing the connection to traditional FEM methods, the previously defined prolongation weights allow Bunge et al. [BHKB20] to define a set of local shape functions $\{\varphi_1^f, \dots, \varphi_{n_f}^f\}$ associated with the vertices of the polygon f: If $\varphi_i^\triangle$ are the (n_f+1) Lagrange basis functions (see Section 3) defined on the refined polygon, one can construct coarse shape functions φ_i^f associated with the polygon nodes as

$$\varphi_i^f = \varphi_i^\triangle + w_i\varphi_f^\triangle, \quad i = 1, \dots, n_f. \tag{89}$$

Here, $\varphi_f^\triangle$ refers to the Lagrange basis function associated with the virtual vertex v_f and w_i is the respective entry in the affine weight vector $\mathbf{w}_f$ previously used for the prolongation matrix. Integrating these shape functions over the polygon mesh as described in Section 5.1.2 would lead to the same discretized stiffness and mass matrices as defined in Equations (83) and (84). Given their construction, Bunge et al.'s shape functions are linear within each virtual triangle and can be integrated analytically, in contrast to the harmonic shape functions, which require more expensive numerical integration.

5.2.2. Similarities to DEC

Interestingly, combining the prolongation matrix $\mathbf{P}$ with the standard Lagrange basis functions $\{\varphi_1^\triangle, \dots \varphi_{|\mathcal{V}_\triangle|}^\triangle\}$ defined on the refined triangle mesh allows us to reinterpret the construction of the stiffness matrix $\mathbf{L}$ in Equation (83) as

$$\mathbf{L} = \mathbf{d}_\mathcal{E}^\mathsf{T} \star^1 \mathbf{d}_\mathcal{E}, \tag{90}$$

which follows the same structure as the operators presented in Section 4. Here $\star^1$ denotes a polygon equivalent of the so-called *Hodge star* operator acting on 1-forms, and the matrix $\mathbf{d}_\mathcal{E} \in \mathbb{R}^{|\mathcal{E}|\times|\mathcal{V}|}$ is the discrete differential operator

$$(\mathbf{d}_\mathcal{E})_{kl} = \begin{cases} -1 & l = i \\ 1 & l = j \\ 0 & \text{otherwise} \end{cases} \tag{91}$$

taking 0-forms to 1-forms acting on edges (in contrast to the previously used coboundary operator $\mathbf{d}$ that projects to halfedges). As in Equation (31), the indexing addresses the k-th row of the operator, this time associated with the k-th edge $e_{ij} \in \mathcal{E}$. Bunge et al. [BHKB20] define a suitable polygon Hodge star by first constructing the respective basis functions for 1-forms: Since the coarse polygon basis functions $\{\varphi_1, \dots \varphi_{|\mathcal{V}|}\}$ are associated with the vertices of the mesh $\mathcal{M}$, they form a set of 0-forms and can be expressed as

$$\varphi_j = \sum_{i=1}^{|\mathcal{V}_\triangle|} \mathbf{P}_{ij}\varphi_i^\triangle. \tag{92}$$

By construction, these bases form a partition of unity. Therefore, they can be used to define a set of polygon Whitney bases [Whi57, AFW06] for 1-forms, with

$$\varphi_{ij} = \varphi_i \cdot \mathbf{d}_\mathcal{E}\varphi_j - \varphi_j \cdot \mathbf{d}_\mathcal{E}\varphi_i = \sum_{l=1}^{|\mathcal{V}_\triangle|}\sum_{k=1}^{l} \mathbf{P}_{ki}\mathbf{P}_{lj}\varphi_{kl}^\triangle, \tag{93}$$

being a 1-form associated to the polygon edge $e_{ij} \in \mathcal{E}$. In order to define a prolongation operator that maps 1-forms from polygon edges to edges on the refined triangle mesh, Bunge et al. define a second prolongation matrix $\mathbf{P}^1 \in \mathbb{R}^{|\mathcal{E}_\triangle| \times |\mathcal{E}|}$ as

$$\mathbf{P}^1_{(ij)(kl)} = \mathbf{P}_{ik}\mathbf{P}_{jl}, \tag{94}$$

with (ij) indicating the row associated to the edge $e^\triangle_{ij} \in \mathcal{E}_\triangle$ on the refined mesh and (kl) the index of the respective coarse polygon edge $e_{kl} \in \mathcal{E}$. This matrix can be combined with the discrete Hodge star $\star^1_\triangle$ on the refined triangle mesh, giving us

$$\mathbf{M}_1 = \star^1 = \left(\mathbf{P}^1\right)^\mathsf{T} \star^1_\triangle \mathbf{P}^1 \tag{95}$$

and

$$\mathbf{L} = \mathbf{d}^\mathsf{T}_\mathcal{E} \left(\mathbf{P}^1\right)^\mathsf{T} \star^1_\triangle \mathbf{P}^1 \mathbf{d}_\mathcal{E}. \tag{96}$$

The question is if this inner product matrix $\mathbf{M}_1$ satisfies the same desiderata as for example the matrices presented by Alexa and Wardetzky [AW11] and de Goes et al. [dGBD20], which remains to be investigated.

5.3. Key Outcomes

The presented FEM methods can, just like the previously discussed Mimetic Polygon Laplacians, be applied to any type of polygon surface mesh. However, the harmonic shape functions are restricted to planar polygons, while the linear virtual refinement method is able to deal with non-planar elements. Furthermore, besides being able to construct all the operators we already described, having explicit shape functions allows us to interpolate any given function within the polygons if we know its values at the vertex positions. In relation to the MFD operators, since the virtual refinement method allows for a reinterpretation of the Laplacian in the same inner product structure, future analysis could further investigate the relation between FEM and MFD approaches and their different qualities.

6. Finite Volume Discretizations

The finite volume method (FV) was originally introduced by Dusinberre [Dus55, Dus61] for the heat equation and can be used on all differential equations that can be expressed through the divergence operator. It follows the idea that the integral of a differential over a small volume can be expressed as a surface integral of the fluxes over the boundary of the same cell [Rap17]. As the MFD, finite volume discretizations can be considered mimetic since they try to enforce balance equations for mass, momentum, and energy on each cell [LMS14], conservation properties that make them well suited for fluid mechanic problems. However, the basic derivation of the 2D Laplace operator with FV assumes a Delaunay triangle mesh, more specifically orthogonal dual and primal edges, in order to prevent negative coefficients. To avoid this restriction, Bunge et al. [BBA21] used a special polygonal variant of the FV, called Discrete Duality Finite Volume (DDFV) [Her00, Her09, DO05, CH11] and combined this technique with their previously described virtual triangle refinement to define a gradient, divergence and Laplacian operator for polygonal meshes.

6.1. Discrete Duality Finite Volume Method

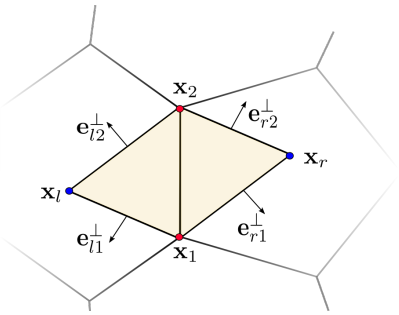


Figure 8: *A 2D DDFV diamond cell spanned between the primal edge $\mathbf{x}_1, \mathbf{x}_2$ and the virtual dual vertices $\mathbf{x}_l, \mathbf{x}_r$. The vectors $\mathbf{e}^\perp_{ij}$ are orthogonal to the diamond edges and involved construction of the gradient operator.*

We will shortly revisit the definition of a 2D gradient operator constructed with the DDFV approach. Given a planar polygonal surface mesh $\mathcal{M}$, we consider a second set of vertices, called *dual* vertices $\mathcal{V}^*$, associated with the barycenter of each face $f \in \mathcal{F}$. For each edge $e \in \mathcal{E}$, they are able to span a so-called *diamond* cell D, consisting of the vertex pair v_1, v_2 connected by e and the dual vertices v_l, v_r associated to the faces adjacent to the edge (see Figure 8). The diamond cell always forms a rectangular shape with edge vectors $\mathbf{e}_{ij} = \mathbf{x}_j - \mathbf{x}_i$ in $\mathbb{R}^2$ assigned to the edge tuples $(i, j) \in D$. On these diamonds, the DDFV method uses Stokes' theorem in order to define a local gradient for a function u in the following way

$$\iint_D \nabla u(\mathbf{x})\, \mathrm{d}\mathbf{x} = \int_{\partial D} u(\mathbf{x})\, \mathbf{n}(\mathbf{x})\, \mathrm{d}\mathbf{x} \tag{97}$$

$$= \sum_{(i,j)\in\partial D} \frac{\mathbf{e}^\perp_{ij}}{\|\mathbf{e}_{ij}\|} \int_0^1 \|\mathbf{e}_{ij}\| \left((1-t)u_i + t u_j\right) \mathrm{d}t \tag{98}$$

$$= \sum_{(i,j)\in\partial D} \mathbf{e}^\perp_{ij} \frac{u_i + u_j}{2}. \tag{99}$$

Here u_i denotes the function values associated to the vertex v_i and $\mathbf{n}(\mathbf{x}) \in \mathbb{R}^2$ is an outward pointing normal vector at point $\mathbf{x}$ along the boundary of the diamond. Therefore, the gradient operator can be solely expressed through the four outward rotated edge vectors of the respective diamond. One of the many benefits of the DDFV method is the discrete duality property between its discrete gradient and divergence operators, also known as Green's "integration-by-parts" formulas [ABHK12, LMS14]. This property is an essential part of the finite volume setting.

6.2. Diamond Laplace for Surface Meshes

Bunge et al. [BBA21] adapt the previously defined DDFV gradient operator in two ways to extend it to polygon surface meshes embedded in 3D:

- First, they reduce the systems' dimension by expressing the values at the dual vertices as affine combinations of the original primal vertices with the help of the previously introduced face prolongation matrix $\mathbf{P}$ (see Equation (82)). As in Section 5.2,

the position of the dual vertices is the minimizer of the sum of squared triangle areas of the refined face.
- Second, they define an *intrinsic* gradient operator associated with the (not necessarily planar) diamond cells, allowing them to directly apply the formula defined in Equation (99).

Given a diamond cell D with vertices (v_1, v_2, v_l, v_r) embedded in $\mathbb{R}^3$, the first objective is to isometrically unfold the planar triangles along their shared edge into a plane. The new 2D coordinates can then be used to construct the diamonds gradient operator $\mathbf{G}_D \in \mathbb{R}^{2\times 4}$. For each column i associated with a vertex it is defined as

$$\mathbf{G}_D(:,i) = \frac{1}{2|D|} \sum_{(i,j)\in\partial D} \tilde{\mathbf{e}}_{ij}^{\perp}, \tag{100}$$

where $|D|$ denotes the diamond's area and $\tilde{\mathbf{e}}_{ij}^{\perp} \in \mathbb{R}^2$ orthogonal vectors to the intrinsic 2D diamond edges. These local gradient matrices are then assembled into a global operator $\mathbf{G}_\diamond \in \mathbb{R}^{2|\mathcal{E}|\times(|\mathcal{V}|+|\mathcal{F}|)}$ defined on the refined triangle mesh $\mathcal{M}_\triangle$. Combined with the surface prolongation matrix $\mathbf{P}$ (see Equation (82)), we obtain a gradient operator for the original polygon mesh through

$$\mathbf{G} = \mathbf{G}_\diamond \mathbf{P}. \tag{101}$$

This operator maps function values u_i associated with the *primal* vertices $v_i \in \mathcal{V}$ to intrinsic gradient vectors $\nabla u|_D \in \mathbb{R}^2$ associated to the diamond cells spanned on the virtual triangle mesh $\mathcal{M}_\triangle$. Following the DDFV discretization of the divergence, Bunge et al. [BBA21] define their diamond divercence operator as

$$\mathbf{D} = \mathbf{P}^\mathsf{T} \mathbf{G}_\diamond^\mathsf{T} \hat{\mathbf{M}}_\diamond \tag{102}$$

where $\hat{\mathbf{M}}_\diamond \in \mathbb{R}^{2|\mathcal{E}|\times 2|\mathcal{E}|}$ is a diagonal matrix containing the diamond D_i's area $|D_i|$ in its diagonal entries with indices $2i$ and $2i+1$. The final stiffness matrix is then directly derived from its definition as the divergence of the gradient and given by

$$\mathbf{L} = \mathbf{D}\mathbf{G} = \mathbf{P}^\mathsf{T} \mathbf{G}_\diamond^\mathsf{T} \hat{\mathbf{M}}_\diamond \mathbf{G}_\diamond \mathbf{P}, \tag{103}$$

mapping from vertices to vertices. The diamond version of the mass matrix required for the strong formulation $\mathbb{L}$ of the Laplacian is defined as

$$\mathbf{M} = \mathbf{P}^\mathsf{T} \mathbf{M}_\diamond \mathbf{P}. \tag{104}$$

It is derived from the standard DDFV diagonal mass matrix $\mathbf{M}_\diamond \in \mathbb{R}^{(|\mathcal{V}|+|\mathcal{F}|)\times(|\mathcal{V}|+|\mathcal{F}|)}$ that distributes the diamond areas to the primal and dual vertices:

$$(\mathbf{M}_\diamond)_{ii} = \begin{cases} \sum_{D\ni v_i} \frac{|D|}{4} & \text{if } v_i \in \mathcal{V} \\ \sum_{D\ni v_i} \frac{|D|}{4} & \text{if } v_i \in \mathcal{V}^* \\ 0 & \text{otherwise.} \end{cases} \tag{105}$$

6.3. Key Outcomes

As for the FEM methods, the diamond Laplacian can be applied to surface meshes with possible non-planar faces. In contrast to the previous operators, the influence of the DDFV background causes its focus to lie more on the construction of a plausible gradient and divergence operator, which can both be constructed intrinsically. The diamond structure combined with the prolongation steps leads to a larger local neighborhood, which can yield more accurate results (see Section 8), but also causes denser and therefore more costly system matrices.

7. Properties of the Polygon Laplacians

In this section, we analyze each of the previously introduced polygon Laplacians based on the properties established by Wardetzky et al. [WMKG07] (see Section 2.1). Since all methods fail to satisfy the maximum principle, but are otherwise able to retain the remaining properties, we will structure the upcoming section accordingly. Therefore, this section will focus more on notable differences within the derivations and proofs of the individual properties instead of differences between the actual operators themselves.

7.1. Symmetry

Each of the presented Laplace operators fulfills symmetry by construction. As the products of individual symmetric or diagonal matrices, the respective inner product matrices $\mathbf{M}_f$ introduced in both [AW11] and [dGBD20] are symmetric as well. Therefore, given that the local stiffness matrix is defined as $\mathbf{d}_f^\mathsf{T}\mathbf{M}_f\mathbf{d}_f$, it and the globally assembled weak form fulfill this property.

Since the FEM stiffness matrix is defined as the integrated dot product of the shape functions' gradients, the property follows naturally for the operators introduced by [MKB*08] and [BHKB20].

In case of the diamond Laplacian, it follows from the definition of the weak form as

$$\mathbf{L} = \mathbf{D}\mathbf{G} = \mathbf{P}^\mathsf{T} \mathbf{G}_\diamond^\mathsf{T} \hat{\mathbf{M}}_\diamond \mathbf{G}_\diamond \mathbf{P} \tag{106}$$

and $\hat{\mathbf{M}}_\diamond$ being a diagonal matrix.

7.2. Locality

All methods, with the exception of [BBA21], define their involved matrices or shape functions locally per polygon. This implies that their respective influence is restricted to the boundary of the individual elements. Therefore, for operators discretized with linear degrees of freedom, the stencil of the Laplacian associated with a single vertex v_i involves all face vertices of the polygons where v_i itself is a part of, which leads to a local neighborhood. However, the diamond Laplacian has a larger stencil than the other operators. Since the diamond structure connects adjacent cells on the refined mesh and the prolongation process distributes values at virtual vertices back to all original face nodes, its neighborhood for a vertex v_i consists of the same vertices as the other methods, and additionally those that are part of a face sharing an edge with the primitives surrounding v_i.

7.3. Linear Precision.

In contrast to the previous two properties, linear precision does not automatically follow from each operator's construction and must be proven individually.

Alexa and Wardetzky. For the Laplacian introduced by Alexa and Wardetzky, similarly to the cotan Laplacian, the proof is based on the area gradient of the polygon with respect to its vertices. Consider the vertex v_i to lie in a planar neighborhood. Starting with Equation (7), we have to focus on two terms in order to show linear precision for Alexa and Wardetzky's operator:

$$\tilde{\mathbf{L}}_f \mathbf{X}_f = \mathbf{d}_f^\mathsf{T} \tilde{\mathbf{M}}_f \mathbf{d}_f \mathbf{X}_f, \tag{107}$$

where

$$\tilde{\mathbf{L}}_f := \mathbf{d}_f^\mathsf{T} \tilde{\mathbf{M}}_f \mathbf{d}_f \tag{108}$$

with $\tilde{\mathbf{M}}_f$ being the inner product matrix defined by Brezzi et al. [BLS05] (see Equation (35)), and

$$\mathbf{d}_f^\mathsf{T} \mathbf{C}_{\bar{f}} \mathbf{U}_{\bar{f}} \mathbf{C}_{\bar{f}}^\mathsf{T} \mathbf{d}_f \mathbf{X}_f. \tag{109}$$

As shown in Equation (37), the latter term vanishes for planar polygons, leaving us with the remaining part $\tilde{\mathbf{L}}_f \mathbf{X}_f$. As shown in Lemma 3 in the original paper, the i-th row of this matrix is equivalent to the polygon's area gradient at vertex v_i:

$$\nabla_{\mathbf{x}_i} |f| = \left(\tilde{\mathbf{L}}_f \mathbf{X}_f\right)_i. \tag{110}$$

Therefore, the expression $(\mathbf{LX})_i$ is equal to the sum over the area gradients of the adjacent faces with respect to the vertex v_i. Given that the neighborhood is planar, this sum becomes zero since moving the vertex in any direction in the plane would leave the area unchanged.

De Goes et al. For de Goes et al.'s Laplacian [dGBD20], we proceed in the same fashion as for Alexa and Wardetzky and consider the individual terms of the inner product matrix, starting with the projection:

$$\mathbf{P}_f \mathbf{d}_f \mathbf{X}_f = \mathbf{P}_f \mathbf{E}_f. \tag{111}$$

As mentioned by the authors, $\mathbf{P}_f$ actually eliminates all discrete 1-forms in the image of

$$\mathbf{V}_f^\flat = \mathbf{E}_f \left(\mathbf{I} - \mathbf{n}_f \mathbf{n}_f^\mathsf{T}\right). \tag{112}$$

Therefore, the projection term vanishes if applied to $\mathbf{E}_f$. As described in Section 4.4, since the vertex v_i is surrounded by planar polygons, the remaining inner product term is equal to $\tilde{\mathbf{L}}_f \mathbf{X}_f$ for each individual face f, allowing for the same area gradient interpretation.

Harmonic Coordinates. In the finite element theory, a common practice is to ensure that shape functions pass the so-called *patch test*, which verifies if they are able to fulfill the linear precision property we defined in Equation (71). A variant of the requirements the shape functions have to meet in order to pass the patch test are the exact same as our definition of the linear precision property for Laplacians [ZTZ13]. Therefore, given that the harmonic shape functions are linearly precise, Martin et al.'s Laplacian satisfies this property as well. In fact, as discussed by the authors in the original paper, the linear polynomial introduced in Equation (75) is crucial to guarantee *exact* linear precision, and therefore this property, independent of the number of chosen kernels.

Linear Virtual Refinement Method. The linear precision property is satisfied if the following expression

$$(\mathbf{Lu})_i = 0, \tag{113}$$

holds for all interior vertices v_i where the incident polygons $f_j \ni v_i$ are co-planar and the values of $\mathbf{u}$ in the one-ring of v_i are obtained by sampling a linear function on the plane. Given that the elements surrounding the interior vertex v_i are co-planar, the same must hold for the refined triangles of these primitives, since the position of the virtual vertex is an affine combination of the existing face vertices. The cotangent Laplacian has linear precision, so any linear function $\mathbf{u}$ sampled at the vertices and prolonged to the refined mesh $\mathbf{u}_\triangle = \mathbf{Pu}$ satisfies

$$(\mathbf{L}_\triangle \mathbf{u}_\triangle)_i = 0 \tag{114}$$

for the interior vertex v_i and all virtual vertices of its surrounding faces. These are exactly the values on the refined mesh that account for the value of the polygon Laplacian at v_i, since each value associated to the virtual vertices is redistributed to the original nodes of the mesh by applying $\mathbf{P}^\mathsf{T}$ to $\mathbf{L}_\triangle \mathbf{u}_\triangle$. Them all being zero gives us $(\mathbf{Lu})_i = 0$.

However, when also applying D-TFEM [WB25], the triangular Laplacian does not necessarily fulfill this property, as discussed in Section 3.3.2. Consequently, the polygon Laplacian also fails to satisfy the property.

Diamond Laplacian. Knowing that the DDFV Laplacian $\mathbf{G}_\diamond^\mathsf{T} \hat{\mathbf{M}}_\diamond \mathbf{G}_\diamond$ satisfies linear precision on the refined mesh [DO05, Her09, CLB*09], applying the full-rank prolongation matrices preserves this property.

7.4. Positive Semi-Definiteness and Null Space

For Martin et al.'s and Bunge et al.'s operators [MKB*08, BHKB20], given that their Laplacians follow the standard FEM approach, the respective stiffness matrices fulfill both properties by construction and do not require individual proofs.

In case of Alexa and Wardetzky [AW11] and de Goes et al. [dGBD20], the strict positive definiteness of their respective inner product matrices $\mathbf{M}_1$ is a fundamental building block in their derivation process, which leads to a kernel that contains only the zero vector. Furthermore, given its construction, the kernel of the coboundary operator $\mathbf{d}$ contains only constant functions. Combining these two matrices into $\mathbf{d}^\mathsf{T} \mathbf{M}_1 \mathbf{d}$ therefore yields a positive semi-definite stiffness matrix with a one-dimensional kernel consisting of constant functions.

For the diamond Laplacian, the diagonal matrix $\hat{\mathbf{M}}_\diamond$ is symmetric positive definite. The prolongation matrix $\mathbf{P}$ has full rank and therefore the constructed stiffness matrix is also positive semi-definite. However, the dimension of the null space requires further explanation. While it is obvious that constant functions, given its definition, lie in the kernel of the gradient, it remains to be shown that they are the only ones that do. As mentioned by Bunge et al. [BBA21], both the gradient of a minimal diamond and a 2D diamond cell can be interpreted as the gradient of an affine function fitted to the function values of the edge midpoints of the respective cell. The

gradient will therefore vanish if all function values associated to the midpoints are identical. However, this can only be satisfied if the function values at the dual and primal vertices of the respective diamond are identical as well. Furthermore, since the diamonds share the function values with their adjacent cells along the common boundary element, this requirement propagates through the complete mesh, leaving only room for constant functions.

7.5. Maximum Principle

As previously mentioned, all of the presented Laplacians are not able to fulfill the maximum principle for general meshes. In case of the Diamond Laplacian, failing to satisfy the maximum principle is a known limitation of the DDFV scheme [QSGBC20], which can not be rectified by applying the prolongation matrices. All other methods reproduce the cotan Laplacian on triangle meshes, which is also not able to retain this property as discussed in Section 3.

8. Evaluation

In this section, we compare the performance of the presented Laplace operators in a number of different computer graphics applications for surface meshes. We are interested in the influence of the respective parameters λ described by Alexa and Wardetzky [AW11] and de Goes et al. [dGBD20] and how they affect the quality of the inner product matrix. Therefore, we analyze a selection of values besides the recommended choices of the authors. Concerning the placement of the virtual vertices, we follow the recommendations of Bunge et al. [BHKB20, BBA21] and use the squared triangle area minimizer, respectively. For the harmonic shape functions [MKB*08], we noticed that the number of chosen kernels and control points strongly affects the results on our chosen test meshes (see Figure 9). We analyzed different numbers of kernels with $m = 4n$ collocation points instead of the recommended ratio of $m = 3n$ from the original paper, since it yielded slightly more accurate results in our setting. Based on this evaluation, we increased the number of edge kernel/collocation points for surface meshes to 20/90. The choice was also influenced by the numerical costs involved in using more samples. The triangles used for the numerical integration are the same as the virtual elements used for the virtual refinement method. Given that the integration of the shape functions is not exact, using a more elaborate tessellation technique could further improve the results.

8.1. Poisson Equation

We analyze the convergence behavior of the different Laplacians by solving the Poisson equation $-\Delta u = f$ with Dirichlet boundary conditions on various refined tessellations of the unit square. We chose the 2D Franke test function [Fra79] for the right hand side f and solve the discrete system

$$\mathbf{Lu} = \mathbf{Mb}, \tag{115}$$

with $\mathbf{b} \in \mathbb{R}^{|\mathcal{V}|}$ containing the values of the analytic Laplacian Δf of the test function sampled at the vertices. The solution $\mathbf{u}$ is then compared to the analytic values of f. The exact formula of the Franke test function is

$$\begin{aligned} f_{2D}(x,y) = &\frac{3}{4}e^{-\frac{(9x-2)^2+(9y-2)^2}{4}} + \frac{3}{4}e^{-\frac{(9x+1)^2}{49}-\frac{9y+1}{10}} \\ &+\frac{1}{2}e^{-\frac{(9x-7)^2+(9y-3)^2}{4}} - \frac{1}{5}e^{-(9x-4)^2-(9y-7)^2}. \end{aligned} \tag{116}$$

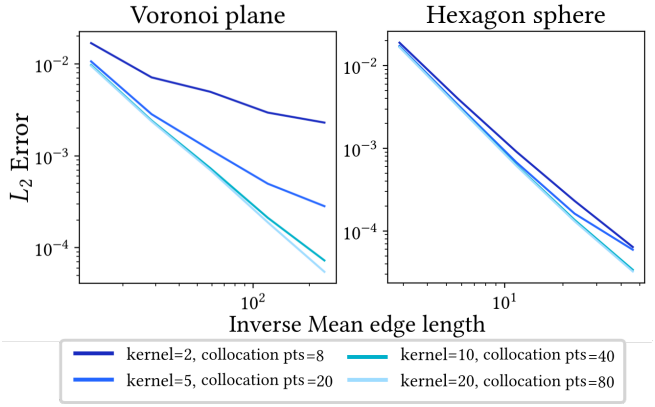


Figure 9: *Effect of the number of chosen kernels and collocation points on the convergence behavior of the harmonic shape functions on the Voronoi plane (left) and hexagon sphere (right). The shown L_2 errors refer to the Poisson systems solved for Franke's test function on planar grids and for the sphercial harmonic function Y_2^3 on unit spheres respectively.*

Figure 10 showcases the L_2 error rates from solving the Franke Poisson system on different surface meshes. Each Laplacian is able to reproduce the expected quadratic convergence rate across all tessellations. In terms of accuracy, both operators introduced by Alexa and Wardetzky [AW11] and de Goes et al. [dGBD20] are able to produce high quality results for $\lambda = 1$ on quad and Voronoi meshes. On concave surfaces, de Goes et al. with parameter $\lambda = 0.5$ yields the lowest errors, closely followed by the operator presented by Bunge et al. [BHKB20]. The Diamond Laplacian consistently maintains low error rates and since it is not reduced to the cotangent Laplacian, it also yields better results on the triangle grid.

8.2. Spherical Harmonics

The eigenfunctions of the Laplacian on the unit sphere $\mathcal{S}^2$ are called the spherical harmonics $Y_l^m : \mathcal{S}^2 \to \mathbb{R}$ with eigenvalues $-l(l+1)$. Using the fact that Y_l^m are eigenfunctions, we can solve for $\mathbf{u} \in \mathbb{R}^{|\mathcal{V}|}$:

$$\mathbf{u} = \mathbf{M}^{-1}\mathbf{L}\mathbf{y}_l^m \tag{117}$$

$$\Leftrightarrow \quad \mathbf{M}\mathbf{u} = \mathbf{L}\mathbf{y}_l^m \tag{118}$$

and rescale the solution with the respective eigenvalue. The entries of $\mathbf{y}_l^m \in \mathbb{R}^{|\mathcal{V}|}$ denote the function values of Y_l^m sampled at the vertices. We can measure the error of $\mathbf{u}$ being an eigenfunction to the presented Laplace operators by evaluating

$$\left\| \mathbf{y}_l^m + \frac{1}{l(l+1)}\mathbf{u} \right\|_{\mathbf{M}}^2 \tag{119}$$

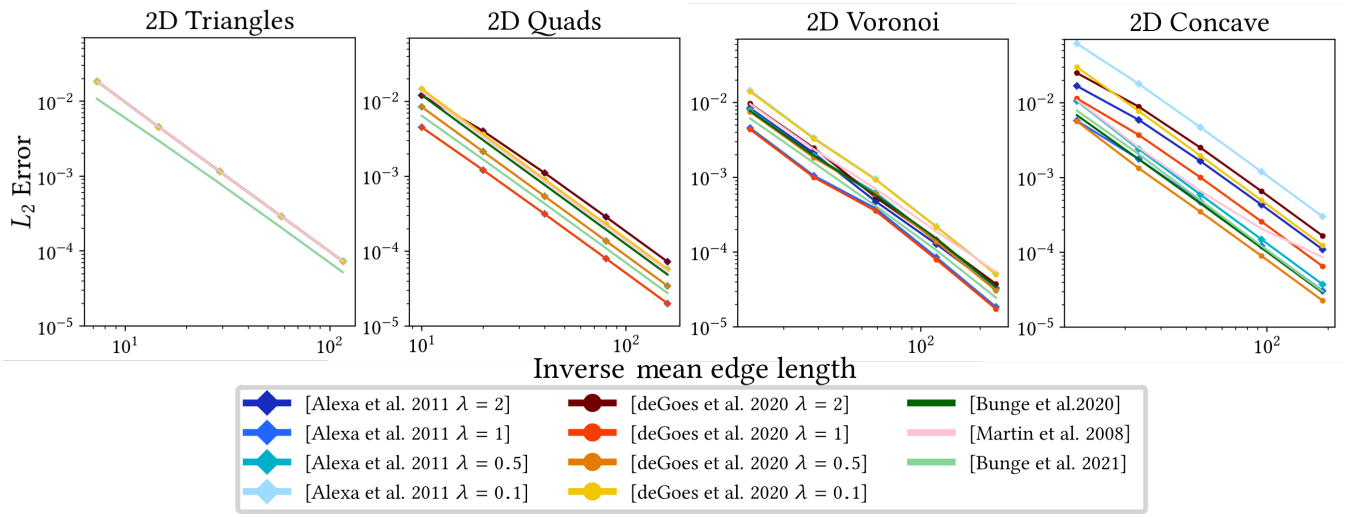


Figure 10: *L_2 error in log-log scale of the Poisson system solved for Franke's test function on planar grids with triangles (left), quads (center left), Voronoi cells (center right), and concave faces (right). Since all methods, with the exception of the Diamond Laplace, are equivalent to the cotangent Laplacian on triangle meshes, the lines overlap in the leftmost plot.*

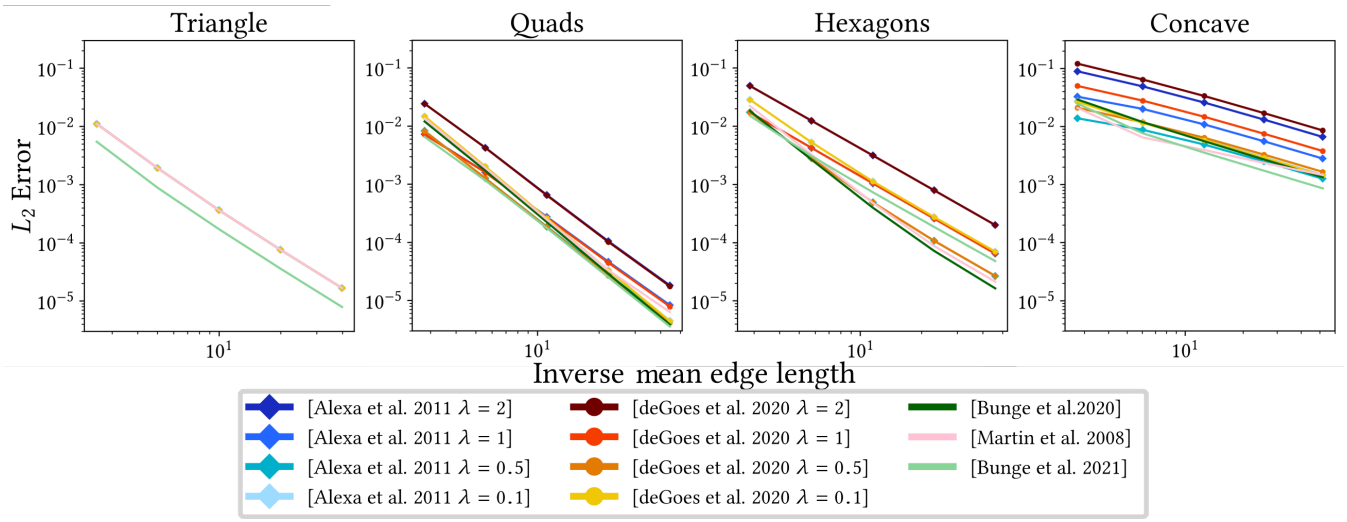


Figure 11: *L_2 error in log-log scale for the Poisson solve of the spherical harmonic function Y_2^3 with eigenvalue -12 on different tessellations of the unit sphere consisting of triangles (left), quads (center left), hexagons (center right) and concave faces (right). All methods, except the Diamond Laplace, are equivalent to the cotangent Laplacian on triangles. This leads to the overlapping lines for the leftmost plot.*

for a selected frequency with non-zero eigenvalue. The L^2 norm is computed with respect to the inner product induced by the mass matrix $\mathbf{M}$.

Figure 11 displays the deviation of the solution from the analytic function values of

$$Y_2^3(x,y,z) = \frac{1}{4}\sqrt{\frac{105}{\pi}}(x^2 - y^2)z \tag{120}$$

with eigenvalue -12. The linear virtual refinement method yields some of the lowest error rates especially for hexagons, while the harmonic shape functions yield very good results on the concave tessellation, which are only surpassed by the Diamond Laplacian.

As for the operators presented by Alexa and Wardetzky and de Goes et al., choosing lower λ for the stabilization term leads to the most accurate results, with $\lambda = 0.5$ being one of the most consistent options. The effect of chosen sample points on the harmonic shape functions and their possible numerical artefacts can be observed in Figure 9. We analyzed different kernel/collocation point samples for both the Franke as well as the spherical harmonics Poisson solve and observed the expected behavior of lower kernel numbers influencing both error and convergence rate.

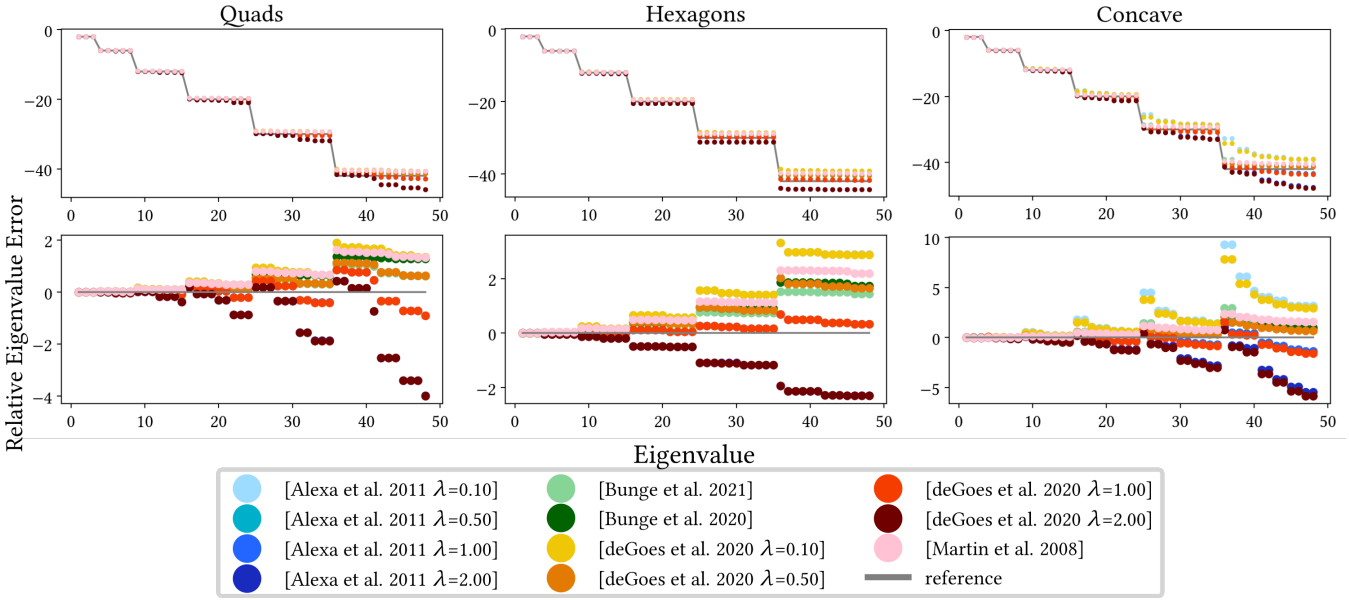


Figure 12: *The smallest 48 non-zero eigenvalues of the Laplacian on different unit spheres with quads (left), hexagons (center), and concave faces (right). The individual top plots shows the computed eigenvalues and the lower ones the relative deviation from the ground truth.*

8.3. Eigenvalues

Given that the spherical harmonics have an analytic expression for their eigenvalues, we can solve the generalized eigenvalue problem

$$\mathbf{L}\mathbf{u} = \tilde{\lambda}\mathbf{M}\mathbf{u} \tag{121}$$

for the eigenvalues $\tilde{\lambda}$ of the discrete Laplacians and compare them to the analytic values. Figure 12 shows the first eigenvalues of each polygon Laplacian obtained on different spherical meshes. For both Alexa and Wardetzky's and de Goes et al.'s operators, a value of $\lambda = 1$ leads to very accurate results. Using smaller values for the parameter leads to larger eigenvalues, while higher λ yield results that are lower than the correct solution. The other methods lead to relatively similar deviations with eigenvalues that display the expected "stair-like" appearance, but are too large in the higher frequencies. The highest and lowest values for λ start to lose the stair-like pattern for the eigenvalues.

8.4. Geodesics in Heat

In order to assess the quality of the divergence and gradient operators, we evaluate them in the context of the geodesics in heat method presented by Crane et al. [CWW13]. Given the i-th unit vector $\mathbf{e}_i \in \mathbb{R}^{|\mathcal{V}|}$, we can obtain the geodesic distances from a vertex v_i to all other vertices in the mesh in three steps: First we solve the heat flow with a fixed small time-step ϵ for the vector $\mathbf{u} \in \mathbb{R}^{|\mathcal{V}|}$:

$$\begin{aligned} &(\mathbf{I} - \epsilon\mathbb{L})\mathbf{u} = \mathbf{e}_i \\ \Leftrightarrow\ &(\mathbf{M} - \epsilon\mathbf{L})\mathbf{u} = \mathbf{M}\mathbf{e}_i. \end{aligned} \tag{122}$$

Then we compute the normalized gradients of the solution vector through

$$\mathbf{g}_j = \frac{(\mathbf{G}\mathbf{u})_j}{\|(\mathbf{G}\mathbf{u})_j\|}. \tag{123}$$

In the last step, we solve the Poisson equation

$$\mathbf{L}\mathbf{v} = \mathbf{D}\mathbf{g} \tag{124}$$

for the geodesic distances $\mathbf{v} \in \mathbb{R}^{|\mathcal{V}|}$ and shift the solution by the offset of the value associated with vertex v_i to zero. Note that, depending on the employed Laplacian, the number and dimension of the gradient vectors vary. For example, the methods introduced by Alexa and Wardetzky and de Goes et al. [AW11, dGBD20] obtain three-dimensional gradient vectors per polygon face, while Bunge et al. [BHKB20] associate their gradient vectors with the virtual triangles, so one per halfedge of the original mesh. On the other hand, the Diamond gradient operator [BBA21] leads to intrinsic two-dimensional gradients that are associated with the virtual diamond cells and therefore with the edges of the original mesh. Additionally, the normalization step in Equation (123) differs depending on the chosen method. While all methods with a geometrically motivated gradient can normalize the vectors by their respective Euclidean length, the method by Alexa and Wardetzky [AW11] needs an alternative approach. As pointed out by Crane et al. [CWW13], interpreting the coboundary operator $\mathbf{d}$ as gradient leads to discrete 1-forms associated with the halfedges, which cannot be directly normalized. However, since $\mathbf{M}_1$ (see Equation (36)) gives us an inner product matrix for 1-forms, they propose to use

$$\|\nabla u\|_f = \sqrt{\frac{\mathbf{u}_f^\mathsf{T}\mathbf{L}_f\mathbf{u}_f}{|f|}} \tag{125}$$

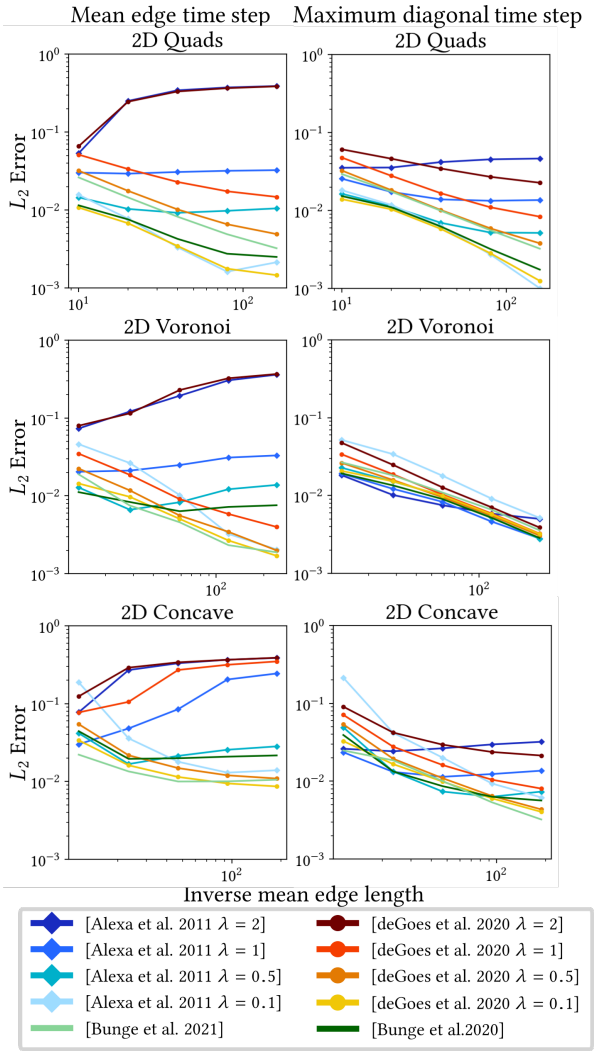


Figure 13: *L_2 error in log-log scale of the Geodesics in heat method on planar grids with quads (top), concave polygons (center) and Voronoi faces (bottom). For each refined planar mesh the selected vertex was the one with the least norm to the center of the plane.*

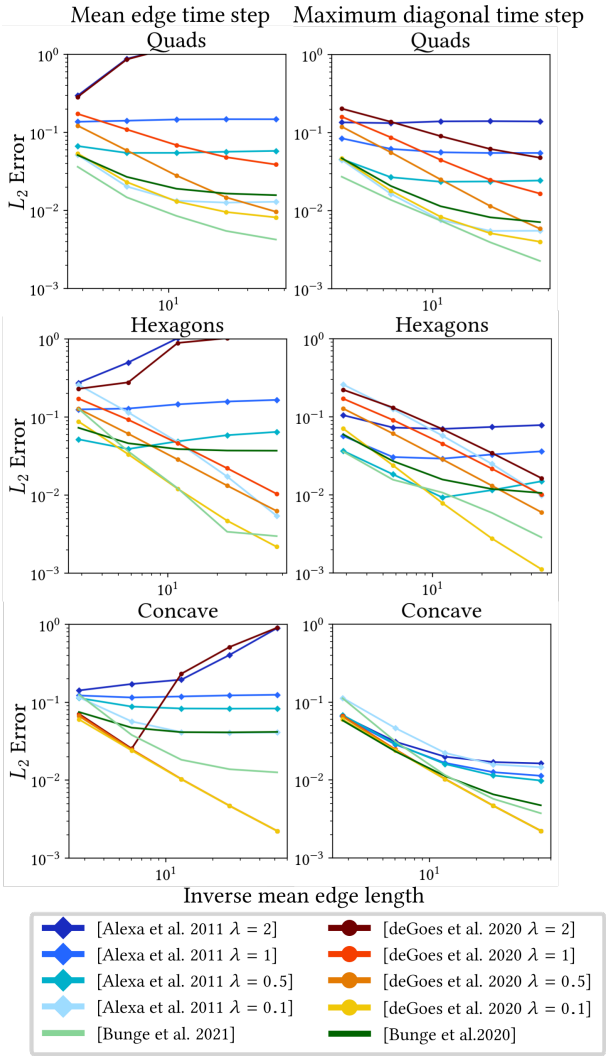


Figure 14: *L_2 error in log-log scale of the Geodesics in heat method on unit spheres with quad (top), hexagon (center) and concave faces (bottom).*

as normalization term by assuming that ∇u is constant over each face and therefore

$$\mathbf{u}_f^{\mathsf{T}} \mathbf{L}_f \mathbf{u}_f = \int_f \|\nabla u\|^2 \, \mathrm{d}A = \|\nabla u\|^2 \, |f| \,. \tag{126}$$

The time step ϵ involved in the first step of the heat method (Equation (122)) is a debated subject. As pointed out by Crane et al. [CWW13], the discrete setting does not follow the expected rule that smaller time steps necessarily lead to more accurate results. However, too large time steps lead to a smoothed approximation of the distances. We therefore compare the behavior of the two most common choices:

- The squared mean edge length of the mesh, as proposed by Crane et al. [CWW13] and used in Bunge et al. [BHKB20].
- The squared length of the longest face diagonal, as suggested by de Goes et al. [dGBD20, dGDMD16].

Figure 13 and 14 show the deviation of the obtained geodesic distances to the Euclidean distance in the plane and the great-circle distance on the unit sphere. Using the mean edge length as time step leads to larger error fluctuations for the methods introduced by Alexa and Wardetzky and de Goes et al., especially for progressively larger λ, while the Laplacians by Bunge et al. [BHKB20, BBA21] remain relatively unaffected. Using the maximum face diagonal stabilizes these deviations, but negatively affects the accuracy for several of the presented methods on some of the test meshes. In general, both the Diamond Laplace and de Goes et al.'s method for $\lambda = 0.1$ have the lowest error rates, in-

dependent of the chosen time step. Additionally, the definition of a geometric gradient operator greatly improves the accuracy of de Goes et al.'s method in comparison to the algebraic coboundary operator used for Alexa and Wardetzky's Laplacian. However, choosing larger values for λ affects both methods negatively. Still, given that λ controls the influence of the stabilization term for both methods, it can not be chosen to be indefinitely close to zero, since this would lead to Laplacians with too large kernels and therefore spurious modes.

8.5. Timings and Sparsity

In this section, we compare different statistics involved in the solution process of a Poisson problem for 2D meshes. Table 1 lists the respective timings to construct the stiffness matrix (build), the number of its non-zero entries (nnz), and the time it takes to solve the system (solve) with Eigen's SimplicialLLT solver [GJ*10]. The timings were measured on a standard workstation with a six-core Intel Xeon 3.6 GHz CPU. As discussed in the locality property section, all Laplacians, with the exception of the Diamond Laplacian, have the same sparsity pattern for their matrices, leading to roughly the same solving times. However, while the implementation of the respective methods has not been extensively optimized for efficiency, it is very apparent that the construction time of the harmonic shape functions by Martin et al. [MKB*08] exceeds the other operators by a tremendous amount. The time it takes to build the involved matrices makes the method not competitive, since its accuracy is on par with the linear virtual refinement method by Bunge et al. [BHKB20] and does not justify the large costs.

8.6. Mesh quality

An interesting sub-case of polygonal meshes are those with non-planar faces. They are in general more challenging than planar elements due to their twisted surfaces, but still occur frequently in computer graphics models. So far, we have only considered meshes with planar polygons in order to establish an intuition of the general effectiveness of each operator. However, in order to evaluate this more challenging setting of non-planar faces, we still have to retain a mesh with a known analytic solution to the given problem. We therefore added noise in tangential direction to the vertex positions of the previously mentioned hexagon spheres and projected them back onto the unit sphere. While this results in non-planar faces, note that the added noise also leads to less balanced hexagonal shapes. Equipped with these new test meshes, we repeat the spherical harmonics convergence test (see Figure 15). In general, each of the evaluated operators displays a higher error rate and slower convergence for the non-planar sphere, with the Laplacians presented by Bunge et al. [BHKB20, BBA21] being the least affected. In contrast to the planar case, choosing a lower hyperparameter λ for the operators of Alexa and Wardetzky [AW11] and de Goes et al. [dGBD20] leads to better results. Similar tendencies can be observed if we add progressively stronger noise to the faces of a single mesh, as depicted in Figure 16. All operators display higher errors for increasingly twisted faces, but in this setting, choosing the same hyperparameter $\lambda = 0.5$ as in the planar case leads to the lowest rates for the works presented by Alexa and Wardetzky and de Goes et al. However, with increasing non-planarity the error converges to similar values as for $\lambda = 0.1$, which appears to be more stable. As in the other test, the operators presented by Bunge et al. lead to the lowest errors. The harmonic shape functions presented by Martin et al. [MKB*08] were not included in these evaluations, since their definition only holds for planar polygons.

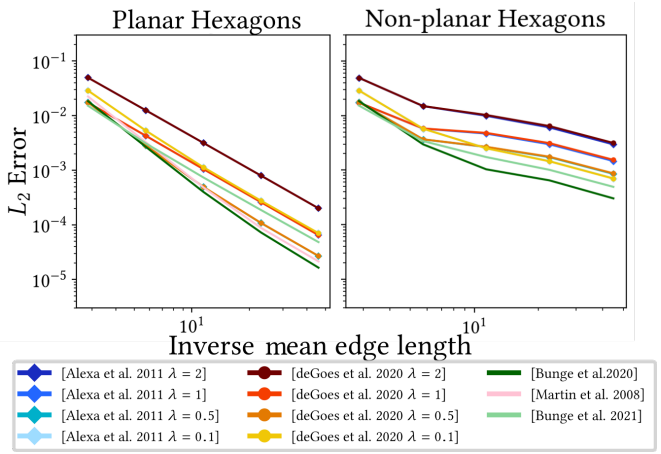


Figure 15: *L_2 error in log-log scale for the Poisson solve of the spherical harmonic function Y_2^3 with eigenvalue -12 on planar and non-planar hexagons. The non-planar elements affect both the convergence rate and accuracy of all methods. The harmonic shape functions are not included in this experiment, since their definition only holds for planar polygons.*

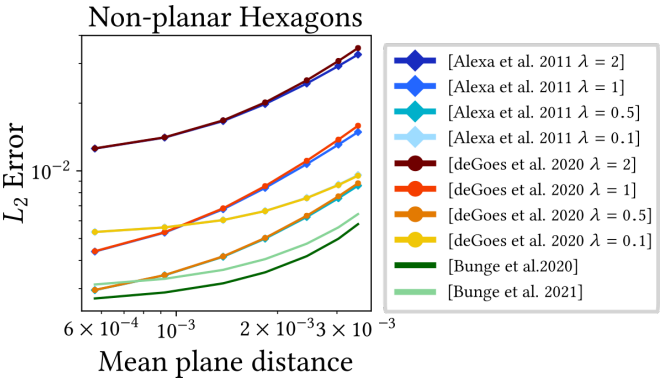


Figure 16: *L_2 error in log-log scale for the Poisson solve of the spherical harmonic function Y_2^3 on progressively non-planar hexagons. We added different magnitudes of tangential noise to the initial faces and projected the vertices back onto the unit sphere. In order to measure the "non-planarity" of the polygons, the x-axis depicts the mean distance of the face vertices to a fitted plane.*

Another aspect in which the underlying tessellation may directly affect the quality of the operators is the shape of the faces itself. For example, Delaunay triangles lead to more favorable results and properties for the Laplacian than meshes consisting of less ideal triangulations [HKA15]. A possible way to quantify the stability of

Table 1: *Different statistics involved in the solution process of a Poisson problem with the presented polygon Laplacians.*

Mesh	$\|\mathcal{V}\|$	[MKB*08]			[AW11]			[dGBD20]			[BHKB20]			[BBA21]		
		build	nnz	solve	build	nnz	solve	build	nnz	solve	build	nnz	solve	build	nnz	solve
Quads 2D	26k	92s	231k	8ms	44ms	231k	8ms	47ms	231k	8ms	10ms	231k	8ms	43ms	537k	21ms
Voronoi 2D	51k	78s	616k	25ms	94ms	616k	25ms	76ms	616k	25ms	29ms	616k	25ms	224ms	1723k	74ms

the operator with respect to the given tessellation is to analyze its condition number κ. It is defined as the ratio

$$\kappa = \frac{\lambda_{max}}{\lambda_{min}} \tag{127}$$

of the Laplacians largest and smallest non-zero eigenvalues λ_{max}

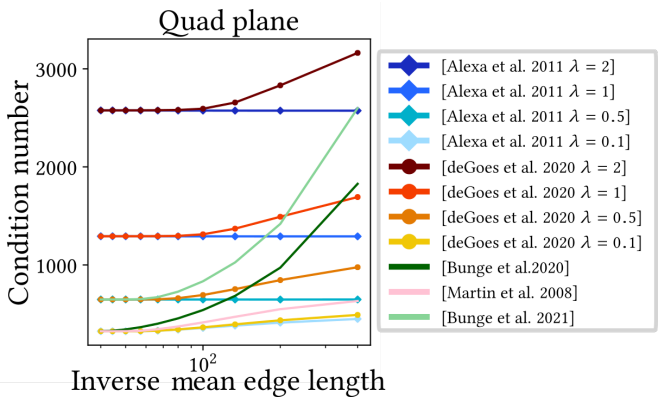


Figure 17: *Condition numbers (Equation (127)) of the different Laplace operators on an initially uniform quad plane. One edge within the grid is progressively shortened, resulting in four faces to become more and more distorted with each iteration. The diminishing quality of the polygons lead to higher condition numbers for most of the operators.*

and λ_{min}. Note that the actual smallest eigenvalue would be zero, since the Laplacian has a one-dimensional kernel. The condition number gives us a notion about the numerical quality of the stiffness and mass matrices and the operators potential to quickly converge to the correct solution of a given problem [KFS13]. We therefore evaluate the effect of decreasing polygon quality on the different Laplacians by incrementally collapsing a single edge of the previously considered quad plane, while measuring its effect on the respective condition numbers. The results can be seen in Figure 17. In general, all of the presented operators are not really affected by the first iterations of edge distortion. However, as the edge length approaches zero, the condition numbers of the Laplacians introduced by Bunge et al. [BHKB20, BBA21] begin to rise significantly due to the diminishing triangle quality of the virtual refinements. The numerical stability of the operators presented by Martin et al. [MKB*08] and de Goes et al. [dGBD20] also decreases, but with a lower magnitude. In contrast, the Laplacian presented by Alexa and Wardetzky [AW11] is not affected by the diminishing edge length and only slightly increases for the smallest hyperparameter $\lambda = 0.1$. However, both operators presented by Alexa and Wardetzky and de Goes et al. follow the pattern that a higher λ leads to generally larger condition numbers.

9. Summary and Recommendation

In light of the above, we want to summarize the presented results and give the reader a recommendation in which situation which operator should be used.

When only triangle meshes are concerned, the canonical choice is using the cotangent Laplacian. While most polygon Laplacians are identical to the cotangent Laplacian for triangle meshes, the direct implementation can be simpler and more performant. Furthermore, modifications, such as the intrinsic Delaunay triangulation [BS07], intrinsic mollification [SC20], or D-TFEM [WB25], can be added to significantly improve the robustness to (near-)degenerate geometry.

In the general polygon case, given their overall performance, the Laplacians presented by Alexa and Wardetzky and de Goes et al. lead to very favorable numerical results if the user is willing to adjust the stabilization parameter λ. Furthermore, de Goes et al. provide a list of other operators that go hand in hand with the matrices introduced in this paper and lead to a larger variety of problems that can be handled with their method.

If the reader is looking for a method that leads to accurate results without any adjustments, the Diamond Laplacian would be their method of choice. However, given its denser matrix pattern, this approach leads to longer solving times.

If this is a problem, a computationally more efficient but slightly less accurate choice would be the operator presented by Bunge et al. [BHKB20]. Given that many applications already work with the cotangent Laplacian, it can be easily integrated since the only missing piece is the prolongation matrix. Furthermore, in the case of (near-)degenerate elements, it can be combined with modifications to the triangle Laplacian, such as intrinsic mollification [SC20] or D-TFEM [WB25].

The harmonic shape functions are not competitive in comparison to the other methods due to their costly construction process. However, they are able to reproduce P1 and Q1 elements on triangles and quads, are C^0 continuous to P1/Q1 at the boundaries of polygons, and can therefore be seamlessly mixed with these standard elements.

10. Conclusion and Outlook

Throughout this tutorial, we attempted to give a comprehensive description of the recent advances achieved within the graphics com-

munity to construct robust discrete differential operators for any given manifold surface mesh. For triangle meshes, we observed that when computing the cotangent Laplacian, there is no clear winner when robust implementation is concerned. Both extrinsic and intrinsic formulations have individual strengths on different types of triangles, whereas purely trigonometric approaches generally perform poorly in comparison. Regardless, clamping of negative weights should be avoided, while ignoring (close-to-)zero-area triangles is often beneficial. Furthermore, using non-standard methods like mollification [SC20], intrinsic Delaunay triangulation [BS07] (in combination with mollification), or D-TFEM [WB25] can dramatically reduce errors and, in most cases, provide a solvable system across a wide range of meshes.

In the second part of the tutorial, we examined the numerical schemes used for the respective polygon Laplacians, providing context for the individual discretization strategies and motivating the problems involved in their generalization process. Furthermore, we analyzed the properties of the presented Laplacians and investigated similarities and parallels between the different methods. Additionally, we introduced a variety of discrete gradient and divergence operators and explained the relationship to their associated Laplacian. Finally, we provide a list of quantitative comparisons between the presented operators that highlight their individual strengths and weaknesses while simultaneously addressing recurring debates within the original papers.

Overall, we hope this tutorial encourages the use of more challenging triangle meshes and general tessellations within the graphics community, given that the necessary tools for many applications are already available. Furthermore, besides the Laplacian, gradient, and divergence, there exist a variety of other discrete differential operators that could be useful to the graphics community, as touched upon by de Goes et al. [dGBD20] and Lipnikov et al. [LMS14] in their survey on the MFD method.

Acknowledgements

We are grateful to Marc Alexa and Philipp Herholz for their help and support during the whole project. Thank you to Olga Sorkine-Hornung and Misha Kazhdan for all the valuable discussions about polygon shape functions and their properties. Finally, we are grateful to Fernando de Goes and Max Wardetzky for providing feedback on the implementation of the respective Laplacians. This research has been supported by the Federal Ministry of Education and Research of Germany and the state of North Rhine-Westphalia as part of the Lamarr Institute for Machine Learning and Artificial Intelligence.

References

[ABHK12] Andreianov B., Bendahmane M., Hubert F., Krell S.: On 3D DDFV discretization of gradient and divergence operators. I. Meshing, operators and discrete duality. *IMA Journal of Numerical Analysis 32*, 4 (2012), pp.1574–1603. doi:10.1093/imanum/drr046. 14

[AFW06] Arnold D. N., Falk R. S., Winther R.: Finite element exterior calculus, homological techniques, and applications. *Acta Numerica 15* (2006), 1–155. 13

[AW11] Alexa M., Wardetzky M.: Discrete Laplacians on general polygonal meshes. *ACM Transactions on Graphics 30*, 4 (2011), 102:1–102:10. 7, 10, 14, 15, 16, 17, 19, 21, 22

[BA76] Babuška I., Aziz A. K.: On the angle condition in the finite element method. *SIAM Journal on Numerical Analysis 13*, 2 (1976), 214–226. 5

[BAB23] Bunge A., Alexa M., Botsch M.: Discrete laplacians for general polygonal and polyhedral meshes. In *SIGGRAPH Asia 2023 Courses* (2023), pp. 1–49. 2

[BB23] Bunge A., Botsch M.: A survey on discrete laplacians for general polygonal meshes. *Computer Graphics Forum 42*, 2 (2023), 521–544. doi:10.1111/cgf.14777. 2

[BBA21] Bunge A., Botsch M., Alexa M.: The diamond laplace for polygonal and polyhedral meshes. *Computer Graphics Forum 40*, 5 (2021), 217–230. 14, 15, 16, 17, 19, 20, 21, 22

[BBW*24] Bunge A., Bukenberger D. R., Wagner S. D., Alexa M., Botsch M.: Polygon Laplacian made robust. *Computer Graphics Forum 43*, 2 (2024). 13

[BdVBM13] Beirão da Veiga L., Brezzi F., Marini L. D.: Virtual elements for linear elasticity problems. *SIAM J. Numer. Anal. 51*, 2 (2013), 794–812. 8, 9

[BHKB20] Bunge A., Herholz P., Kazhdan M., Botsch M.: Polygon Laplacian made simple. *Computer Graphics Forum 39*, 2 (2020), 303–313. 12, 13, 15, 16, 17, 19, 20, 21, 22

[Bis09] Bishop J.: Simulating the pervasive fracture of materials and structures using randomly close packed Voronoi tessellations. *Computational Mechanics 44*, 4 (2009), 455–471. 2

[Bis14] Bishop J.: A displacement-based finite element formulation for general polyhedra using harmonic shape functions. *International Journal for Numerical Methods in Engineering 97* (2014), 1–31. 11

[BLS05] Brezzi F., Lipnikov K., Simoncini V.: A family of mimetic finite difference methods on polygonal and polyhedral meshes. *Mathematical Models and Methods in Applied Sciences 15*, 10 (2005), 1533–1551. 7, 8, 10, 16

[BLSS07] Brezzi F., Lipnikov K., Shashkov M., Simoncini V.: A new discretization methodology for diffusion problems on generalized polyhedral meshes. *Computer Methods in Applied Mechanics and Engineering 196*, 37 (2007), 3682–3692. doi:10.1016/j.cma.2006.10.028. 7

[BS07] Bobenko A. I., Springborn B. A.: A discrete Laplace–Beltrami operator for simplicial surfaces. *Discrete & Computational Geometry 38*, 4 (2007), 740–756. 2, 5, 6, 13, 22, 23

[BSBK02] Botsch M., Steinberg S., Bischoff S., Kobbelt L.: OpenMesh: A generic and efficient polygon mesh data structure. In *OpenSG Symposium* (2002). https://graphics.rwth-aachen.de/software/openmesh. 4

[BSPG06] Botsch M., Sumner R., Pauly M., Gross M.: Deformation transfer for detail-preserving surface editing. In *Proc. of Vision, Modeling and Visualization* (2006), pp. 357–364. 4

[CG16] Chen R., Gotsman C.: On pseudo-harmonic barycentric coordinates. *Computer Aided Geometric Design 44* (2016), 15–35. doi:10.1016/j.cagd.2016.04.005. 11

[CH11] Coudière Y., Hubert F.: A 3D discrete duality finite volume method for nonlinear elliptic equations. *SIAM J. Sci. Comput. 33*, 4 (2011), 1739–1764. 14

[CLB*09] Chuang M., Luo L., Brown B. J., Rusinkiewicz S., Kazhdan M.: Estimating the Laplace-Beltrami operator by restricting 3d functions. *Computer Graphics Forum 28*, 5 (2009), 1475–1484. 16

[CWW13] Crane K., Weischedel C., Wardetzky M.: Geodesics in heat: A new approach to computing distance based on heat flow. *ACM Transactions on Graphics 32*, 5 (2013), 152:1–152:11. 6, 19, 20

[dGBD20] DE GOES F., BUTTS A., DESBRUN M.: Discrete differential operators on polygonal meshes. *ACM Transactions on Graphics 39*, 4 (2020), 110:1–110:14. 8, 9, 10, 14, 15, 16, 17, 19, 20, 21, 22, 23

[dGDMD16] DE GOES F., DESBRUN M., MEYER M., DEROSE T.: Subdivision exterior calculus for geometry processing. *ACM Transactions on Graphics 35*, 4 (2016), 133:1–133:11. 20

[DHLM05] DESBRUN M., HIRANI A. N., LEOK M., MARSDEN J. E.: Discrete exterior calculus. *arXiv: Differential Geometry* (2005). doi:10.48550/ARXIV.MATH/0508341. 10

[DMSB99] DESBRUN M., MEYER M., SCHRÖDER P., BARR A. H.: Implicit fairing of irregular meshes using diffusion and curvature flow. In *Proceedings of ACM SIGGRAPH* (1999), pp. 317–324. 2, 4

[DO05] DOMELEVO K., OMNÉS P.: A finite volume method for the Laplace equation on almost arbitrary two-dimensional grids. *Math. Model. Numer. Anal. (M2AN) 39*, 6 (2005), 1203–1249. 14, 16

[Dus55] DUSINBERRE G. M.: Heat transfer calculations by numerical methods. *Journal of the American Society for Naval Engineers 67*, 4 (1955), 991–1002. doi:10.1111/j.1559-3584.1955.tb03171.x. 14

[Dus61] DUSINBERRE G.: *Heat-transfer Calculations by Finite Differences*. International textbooks in mechanical engineering. International Textbook Company, 1961. 14

[Dzi88] DZIUK G.: Finite Elements for the Beltrami operator on arbitrary surfaces. In *Partial Differential Equations and Calculus of Variations* (1988), Hildebrandt S., Leis R., (Eds.), Springer Berlin Heidelberg, pp. 142–155. doi:10.1007/BFb0082865. 2

[FK98] FAIRWEATHER G., KARAGEORGHIS A.: The method of fundamental solutions for elliptic boundary value problems. *Advances in Computational Mathematics 9*, 1 (1998), 69–95. 11

[Flo03] FLOATER M. S.: Mean value coordinates. *Computer Aided Geometric Design 20*, 1 (2003), 19–27. 11

[Flo15] FLOATER M. S.: Generalized barycentric coordinates and applications. *Acta Numerica 24* (2015), 161–214. doi:10.1017/S0962492914000129. 11

[Fra79] FRANKE R.: *A critical comparison of some methods for interpolation of scattered data*. Tech. rep., Naval Postgraduate School, 1979. 17

[GJ*10] GUENNEBAUD G., JACOB B., ET AL.: Eigen v3. http://eigen.tuxfamily.org, 2010. 21

[GVL96] GOLUB G. H., VAN LOAN C. F.: *Matrix Computations*. Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, 1996. 12

[Her00] HERMELINE F.: A finite volume method for the approximation of diffusion operators on distorted meshes. *J. Comput. Phys. 160*, 2 (2000), 481–499. 14

[Her09] HERMELINE F.: A finite volume method for approximating 3D diffusion operators on general meshes. *J. Comput. Phys. 228*, 16 (2009), 5763–5786. 14, 16

[HKA15] HERHOLZ P., KYPRIANIDIS J. E., ALEXA M.: Perfect Laplacians for polygon meshes. *Computer Graphics Forum 34*, 5 (2015), 211–218. 21

[HPW06] HILDEBRANDT K., POLTHIER K., WARDETZKY M.: On the convergence of metric and geometric properties of polyhedral surfaces. *Geometriae Dedicata 123* (2006), 89–112. doi:10.1007/s10711-006-9109-5. 3, 4

[HS08] HORMANN K., SUKUMAR N.: Maximum entropy coordinates for arbitrary polytopes. *Computer Graphics Forum 27*, 5 (2008), 1513–1520. 11

[Hug12] HUGHES T.: *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. Dover Civil and Mechanical Engineering. Dover Publications, 2012. 11

[JMD*07] JOSHI P., MEYER M., DEROSE T., GREEN B., SANOCKI T.: Harmonic coordinates for character articulation. *ACM Trans. Graph. 26*, 3 (2007), 71–81. doi:10.1145/1276377.1276466. 11

[JP*18] JACOBSON A., PANOZZO D., ET AL.: libigl: A simple C++ geometry processing library. https://libigl.github.io, 2018. 5

[JSW05] JU T., SCHAEFER S., WARREN J.: Mean value coordinates for closed triangular meshes. *ACM Transactions on Graphics 24*, 3 (2005), 561–566. 11

[Kah97] KAHAN W.: *Miscalculating Area and Angles of a Needle-like Triangle*. Tech. rep., University of California, Berkeley, 1997. 5

[KFS13] KRISHNAN D., FATTAL R., SZELISKI R.: Efficient preconditioning of laplacian matrices for computer graphics. *ACM Transactions on Graphics 32*, 4 (2013). doi:10.1145/2461912.2461992. 22

[L*25] LÉVY B., ET AL.: Geogram. https://github.com/BrunoLevy/geogram, 2025. 4

[Lee97] LEE J.: *Riemannian Manifolds: An Introduction to Curvature*. Graduate Texts in Mathematics. Springer New York, 1997. 9

[Liv19] LIVESU M.: Cinolib: A generic programming header only C++ library for processing polygonal and polyhedral meshes. *Transactions on Computational Science XXXIV* (2019). https://github.com/mlivesu/cinolib/. doi:10.1007/978-3-662-59958-7_4. 4

[LMS14] LIPNIKOV K., MANZINI G., SHASHKOV M.: Mimetic finite difference method. *Journal of Computational Physics 257* (2014), 1163–1227. 7, 14, 23

[Mac49] MACNEAL R.: *The Solution of Partial Differential Equations by Means of Electrical Networks*. PhD thesis, California Institute of Technology, 1949. 4

[MDSB03] MEYER M., DESBRUN M., SCHRÖDER P., BARR A. H.: Discrete differential-geometry operators for triangulated 2-manifolds. In *Visualization and Mathematics III*, Hege H.-C., Polthier K., (Eds.). Springer-Verlag, 2003, pp. 35–57. 2

[MKB*08] MARTIN S., KAUFMANN P., BOTSCH M., WICKE M., GROSS M.: Polyhedral finite elements using harmonic basis functions. *Computer Graphics Forum 27*, 5 (2008), 1521–1529. 11, 12, 15, 16, 17, 21, 22

[MKO*23] MARUANI N., KLOKOV R., OVSJANIKOV M., ALLIEZ P., DESBRUN M.: VoroMesh: Learning watertight surface meshes with Voronoi diagrams. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)* (2023), pp. 14565–14574. 6

[MMP87] MITCHELL J. S., MOUNT D. M., PAPADIMITRIOU C. H.: The discrete geodesic problem. *SIAM Journal on Computing 16*, 4 (1987), 647–668. 6

[Mus97] MUSIN O. R.: Properties of the Delaunay triangulation. In *Proceedings of Symposium on Computational Geometry* (1997), pp. 424–426. doi:10.1145/262839.263061. 13

[OSTLY12] OOI E. T., SONG C., TIN-LOI F., YANG Z.: Polygon scaled boundary finite elements for crack propagation modelling. *International Journal for Numerical Methods in Engineering 91*, 3 (2012), 319–342. doi:10.1002/nme.4284. 2

[PP93] PINKALL U., POLTHIER K.: Computing discrete minimal surfaces and their conjugates. *Experim. Math. 2* (1993), 15–36. 2, 4

[QKL*26] QUIRINY A., KUČERA V., LAMBRECHTS J., MOËS N., REMACLE J.-F.: The tempered finite element method. *Journal of Computational Physics 549* (2026). 2, 5, 6

[QSGBC20] QUENJEL E. H., SAAD M., GHILANI M., BESSEMOULIN-CHATARD M.: Convergence of a positive nonlinear DDFV scheme for degenerate parabolic equations. *Calcolo 50*, 19 (2020). doi:10.1007/s10092-020-00367-5. 17

[Rap17] RAPP B. E.: Chapter 31 - finite volume method. In *Microfluidics: Modelling, Mechanics and Mathematics*, Micro and Nano

Technologies. Elsevier, Oxford, 2017, pp. 633–654. doi:10.1016/B978-1-4557-3141-1.50031-9. 14

[Ros97] ROSENBERG S.: *The Laplacian on a Riemannian Manifold*. London Mathematical Society Student Texts. Cambridge University Press, 1997, pp. 1–51. doi:10.1017/CBO9780511623783.002. 7

[SB23] SIEGER D., BOTSCH M.: The Polygon Mesh Processing Library. https://github.com/pmp-library/pmp-library, 2023. 5

[SBMS23] SORGENTE T., BIASOTTI S., MANZINI G., SPAGNUOLO M.: A survey of indicators for mesh quality assessment. *Computer Graphics Forum 42*, 2 (2023), 461–483. 5

[SC*19] SHARP N., CRANE K., ET AL.: GeometryCentral: A modern C++ library of data structures and algorithms for geometry processing. https://geometry-central.net, 2019. 5

[SC20] SHARP N., CRANE K.: A Laplacian for nonmanifold triangle meshes. *Computer Graphics Forum 39*, 5 (2020), 69–80. 2, 5, 6, 22, 23

[Sha21] SHARP N.: *Intrinsic Triangulations in Geometry Processing*. Phd thesis, Carnegie Mellon University, 12 2021. doi:10.1184/R1/17209181.v1. 4

[She02] SHEWCHUK J. R.: What is a good linear element? interpolation, conditioning, and quality measures. In *International Meshing Roundtable (IMR)* (2002). 5

[The24] THE CGAL PROJECT: *CGAL User and Reference Manual*, 6.0.1 ed. CGAL Editorial Board, 2024. URL: https://doc.cgal.org/6.0.1/Manual/packages.html. 5

[TS06] TABARRAEI A., SUKUMAR N.: Application of polygonal finite elements in linear elasticity. *International Journal of Computational Methods 03* (2006), 503–520. 2

[TS08] TABARRAEI A., SUKUMAR N.: Extended finite element method on polygonal and quadtree meshes. *Computer Methods in Applied Mechanics and Engineering 197* (2008), 425–438. doi:10.1016/j.cma.2007.08.013. 2

[Vis23] VISUAL COMPUTING LAB: The visualization and computer graphics library. https://github.com/cnr-isti-vclab/vcglib, 2023. 4

[War08] WARDETZKY M.: *Convergence of the Cotangent Formula: An Overview*. Birkhäuser Basel, Basel, 2008, pp. 275–286. doi:10.1007/978-3-7643-8621-4_15. 3, 4

[WB25] WAGNER S. D., BOTSCH M.: Robust discrete differential operators for wild geometry. In *Proc. of Vision, Modeling, Visualization* (2025). 2, 4, 5, 6, 13, 16, 22, 23

[WBG07] WICKE M., BOTSCH M., GROSS M.: A finite element method on convex polyhedra. *Computer Graphics Forum 26*, 3 (2007), 355–364. 11

[Whi57] WHITNEY H.: *Geometric Integration Theory*. Princeton University Press, 1957. 13

[WMKG07] WARDETZKY M., MATHUR S., KÄLBERER F., GRINSPUN E.: Discrete Laplace operators: No free lunch. In *Proceedings of Eurographics Symposium on Geometry Processing* (2007), pp. 33–37. 2, 3, 5, 6, 15

[YZX*04] YU Y., ZHOU K., XU D., SHI X., BAO H., GUO B., SHUM H.-Y.: Mesh editing with Poisson-based gradient field manipulation. In *Proceedings of ACM SIGGRAPH* (2004), pp. 644–651. 6

[Zlá68] ZLÁMAL M.: On the finite element method. *Numerische Mathematik 12*, 5 (1968), 394–409. 5

[ZTZ13] ZIENKIEWICZ O., TAYLOR R., ZHU J.: Chapter 8 - the patch test, reduced integration, and nonconforming elements. In *The Finite Element Method: its Basis and Fundamentals*, Zienkiewicz O., Taylor R., Zhu J., (Eds.), 7th edition ed. Butterworth-Heinemann, Oxford, 2013, pp. 257–284. doi:10.1016/B978-1-85617-633-0.00008-3. 16