Discrete Laplacians for General Polygonal and Polyhedral Meshes

ASTRID BUNGE, TU Dortmund University, Germany
MARC ALEXA, TU Berlin, Germany
MARIO BOTSCH, TU Dortmund University, Germany

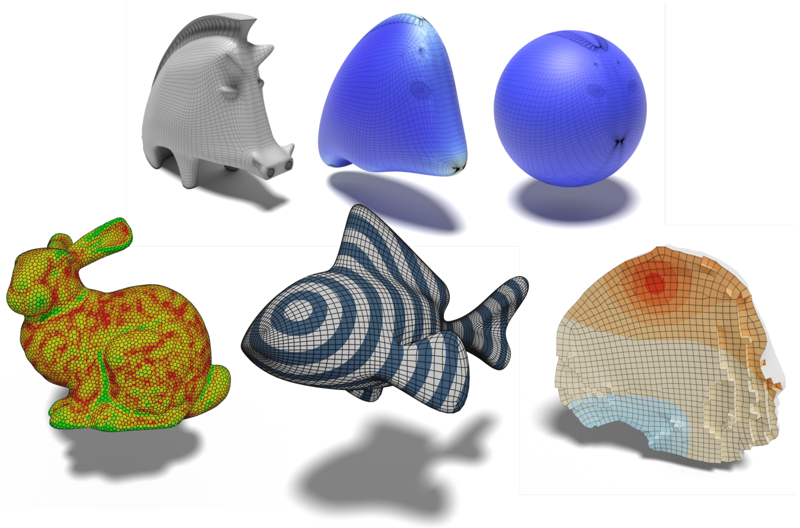


Fig. 1. A selection of applications of the Laplace operator on different polygonal and polyhedral meshes. Figures taken from [Bunge et al. 2021, 2020, 2022].

The Laplace-Beltrami operator is one of the essential tools in geometric processing. It allows us to solve numerous partial differential equations on discrete surface and volume meshes, which is a fundamental building block in many computer graphics applications. Discrete Laplacians are typically limited to standard elements like triangles or quadrilaterals, which severely constrains the tessellation of the mesh. But in recent years, several approaches were able to generalize the Laplace Beltrami and its closely related gradient and divergence operators to more general meshes. This allows artists and engineers to work with a wider range of elements which are sometimes required and beneficial in their field. This course, which extends the state-of-the-art report by Bunge and Botsch [2023], discusses the different constructions of these three ubiquitous differential operators on arbitrary polygons and polyhedra and analyzes their individual advantages and properties in common computer graphics applications.

ACM Reference Format:
Astrid Bunge, Marc Alexa, and Mario Botsch. 2023. Discrete Laplacians for General Polygonal and Polyhedral Meshes. In *SIGGRAPH Asia 2023 Courses (SA Courses '23), December 12-15, 2023.* ACM, New York, NY, USA, 49 pages. https://doi.org/10.1145/3610538.3614620

SA Courses '23, December 12-15, 2023, Sydney, NSW, Australia

ACM ISBN 979-8-4007-0309-6/23/12.
https://doi.org/10.1145/3610538.3614620

Contents

1 INTRODUCTION

The discrete Laplace-Beltrami operator, or Laplacian for short, is an ubiquitous tool in geometry processing. It allows us to solve numerous partial differential equations on discrete surface and volume 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 [Desbrun et al. 1999; Dziuk 1988; Meyer et al. 2003; Pinkall and Polthier 1993] is omnipresent in graphics and geometry processing.

However, due to the growing needs in modeling and engineering applications, recent papers point out that the restriction to triangle or tetrahedral 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 [Bishop 2009; Ooi et al. 2012; Tabarraei and Sukumar 2008] or linear elasticity problems [Tabarraei and Sukumar 2006]. Additionally, since microstructures 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 [Tabarraei and Sukumar 2006]. 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 course, based on the state-of-the-art report by Bunge and Botsch [2023], therefore intends to showcase occurring similarities between the presented methods that may not be apparent if considered individually. In order to achieve this, we summarize existing approaches that extend the Laplace operator to surface polygon meshes and, if possible, to volumetric polyhedral meshes.

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 course. 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. First, we will repeat necessary definitions required for the operator’s construction, followed by the more detailed explanations of the involved polygonal operators and their different ideas, including possible volumetric extensions. Afterwards, we will discuss the required properties a discrete Laplacian should fulfill based on the work presented by Wardetzky et al. [2007] and will analyze the respective operators in this context. The operators are evaluated in a variety of quantitative comparisons that address reoccurring debates within the original papers. The source code for the individual operators and experiments can be found at https://github.com/mbotsch/polyLaplace to enable researchers to experiment with the different discretizations. In the end, the course intends to provide the reader with an intuition to choose the optimal polygon Laplacian 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. Likewise, a 3D polyhedral mesh has the same structure with only one additional set consisting of the volumetric cells C.

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 or cell. We therefore define:

- The matrix $\mathbf{X} \in \mathbb{R}^{|\mathcal{V}|\times 3}$ encodes the vertex positions of the mesh in its rows.
- $\mathbf{X}_f = (\mathbf{x}_1^f, \ldots, \mathbf{x}_{n_f}^f)^\mathsf{T}$ is the $n_f \times 3$ matrix containing in its rows the cyclically ordered vertex positions $\mathbf{x}_i^f$ of the face f.
- $\mathbf{E}_f = (\mathbf{e}_1^f, \ldots, \mathbf{e}_{n_f}^f)^\mathsf{T}$ 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.
- $\mathbf{B}_f = (\mathbf{b}_1^f, \ldots, \mathbf{b}_{n_f}^f)^\mathsf{T}$ 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 must 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. [2007] and for tetrahedral meshes by Alexa et al. [2020]. However, these requirements equally hold for general polygon and polyhedral 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. [2007] 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 $u : \Omega \longrightarrow \mathbb{R}$ 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 $\langle u, v\rangle = \int_\Omega uv \,\mathrm{d}A$. 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 parenthesis. Alternatively, we can omit the influence of the mass matrix and require the stiffness matrix to satisfy

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

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 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 [Wardetzky et al. 2007].

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 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. [2006] and Wardetzky [2008]. This property will not be proven for the upcoming operators, but analyzed empirically in the result section.

3 COTANGENT LAPLACIAN ON TRIANGLE MESHES

One property shared by almost all of the discussed polygon Laplacians is that they reproduce the standard cotangent stiffness matrix on triangle surface meshes. We will therefore shortly revisit its definition based on the finite element discretization. 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

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

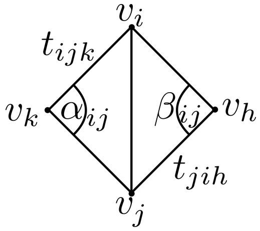
and

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

with

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

Here t_{ijk} and t_{jih} 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. Note that for better numerical robustness, the cotangent values are not obtained by using the inverse trigonometric function itself, but rather through the respective edge lengths and area of the involved triangle. Given a triangle t_{ijk} with edge lengths l_{ij}, l_{ik} and l_{jk} respectively, we can compute the cotangent of the corner angle θ_k at vertex v_k through



$$\cot\theta_k = \frac{l_{jk}^2 + l_{ik}^2 - l_{ij}^2}{4\,|t_{ijk}|}. \tag{14}$$

Therefore, the Laplacian itself can be constructed intrinsically [Sharp 2021]. 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

$$\mathbf{G}_i(:, 1) = \frac{(\mathbf{x}_l - \mathbf{x}_k)^{\perp}}{2\,|t_{jkl}|}. \tag{15}$$

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

$$\mathbf{D} = \mathbf{G}^{\mathsf{T}}\hat{\mathbf{M}}, \tag{16}$$

with $\hat{\mathbf{M}} \in \mathbb{R}^{3|\mathcal{F}| \times 3|\mathcal{F}|}$ being the diagonal mass matrix containing the area of the triangle i in the three consecutive diagonal entries associated with face i [Botsch et al. 2006]. 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 [Sharp 2021].

3.1 Cotangent Laplacian on Tetrahedral Meshes

In the 3-dimensional case, given a tetraherdal mesh $\mathcal{M}$, the $|\mathcal{V}| \times |\mathcal{V}|$ stiffness and mass matrices are discretized similarly to Equation (11) and Equation (12), but with the difference that the volumetric linear Lagrange basis is used instead. This leads to the 3D stiffness matrix

$$\mathbf{L}_{ij} = \begin{cases} -w_{ij} & \text{if } j \in \mathcal{N}(i), \\ \sum_{k \in \mathcal{N}(i)} w_{ik} & \text{if } j = i, \\ 0 & \text{otherwise,} \end{cases} \tag{17}$$

with

$$w_{ij} = \frac{1}{6} \sum_{t_{ijkl}} l_{kl} \cot \theta_{kl}^{ij}. \tag{18}$$

The sum is taken over all thetrahedra t_{ijkl} that include the edge connecting the vertices v_i and v_j. The edge length between v_k and v_l is denoted by l_{kl} and the angle θ_{kl}^{ij} is the respective interior angle between the adjacent triangles t_{ikl} and t_{jkl} [Crane 2019].

A commonly used alternative to the volumetric version of Equation (11) is the (lumped) diagonal mass matrix $\mathbf{M}$ [Alexa et al. 2020; Crane 2019] with

$$\mathbf{M}_{ii} = \frac{1}{4} \sum_{t_{ijkl}} \left| t_{ijkl} \right|. \tag{19}$$

The sum is taken over all tetrahedra containing the vertex v_i and $\left|t_{ijkl}\right|$ denotes the volume of the tetrahedron t_{ijkl}.

The discrete gradient operator for a tetrahedron $t_n = t_{ijkl}$ can be defined as follows [Alexa et al. 2020]: Given a vertex v_i within the tetrahedron, let $\mathbf{n}_{jkl}$ be the face normal of the opposite triangle t_{jkl} and $A_i = \left|t_{jkl}\right|$ its area. The local gradient matrix $\mathbf{G}_n \in \mathbb{R}^{3\times 4}$ can then be constructed column-wise for each vertex within the element. For example, with the first column referring to vertex v_i, it would be defined as

$$\mathbf{G}_n(:, 1) = \frac{A_i}{3\,|t_n|} \mathbf{n}_{jkl}. \tag{20}$$

The local sub-gradients are then assembled into the global gradient operator $\mathbf{G} \in \mathbb{R}^{3|C|\times|\mathcal{V}|}$. As in Equation (16), the volumetric divergence operator $\mathbf{D} \in \mathbb{R}^{|\mathcal{V}|\times 3|C|}$ is then given by

$$\mathbf{D} = \mathbf{G}^{\mathsf{T}} \hat{\mathbf{V}}, \tag{21}$$

with $\hat{\mathbf{V}} \in \mathbb{R}^{3|C|\times 3|C|}$ being the diagonal mass matrix containing the volumes of the tetrahedron i in the three consecutive diagonal entries associated with cell i.

3.2 Properties

Symmetry. Considering the individual entries of the stiffness matrices defined in Equation (12) and Equation (17), the cotan operator is symmetric by construction in both 2D and 3D.

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 $\Delta f(\mathbf{x}_i)$ of a function f at vertex v_i can be expressed through the well known cotan formula [Desbrun et al. 1999; MacNeal 1949; Pinkall and Polthier 1993]

$$\Delta f(\mathbf{x}_i) = \frac{1}{2} \sum_{v_j \in \mathcal{N}(v_i)} (\cot \alpha_{ij} + \cot \beta_{ij})(f(\mathbf{x}_j) - f(\mathbf{x}_i)), \tag{22}$$

the operator satisfies the condition given in Equation (9) and therefore has a one dimensional kernel. The same holds for the tetrahedral equivalent by using the weights defined in Equation (18).

Locality. By construction, for both triangles and tetrahedra, 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

$$\nabla_{\mathbf{x}_i} A = \frac{\cot \theta_k}{2}(\mathbf{x}_i - \mathbf{x}_j) + \frac{\cot \theta_j}{2}(\mathbf{x}_i - \mathbf{x}_k), \tag{23}$$

with θ_j and θ_k denoting the angles at the respective vertices v_j,v_k and $A = \left|t_{ijk}\right|$ 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 [Desbrun et al. 1999], which, if the triangles all lie within the same plane, becomes zero. In 3D we can derive the operator as the gradient of the volumes enclosed by the tetrahedra [Alexa et al. 2020; Crane 2019]. The gradient of the volume for a tetrahedral mesh in a 3-dimensonal subspace also vanishes for inner vertices.

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 $\alpha_{ij} + \beta_{ij} > \pi$. For tetrahedral meshes, all involved dihedral angles would have to be acute in order to retain negative weights, which can not be guaranteed [Alexa et al. 2020].

Convergence. The convergence behavior of the cotan Laplace was discussed by Hildebrand et al. [2006] and Wardetzky [2008]. 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.

4 MIMETIC FINITE DIFFERENCES

The Mimetic Finite Difference method (MFD) [Lipnikov et al. 2014] 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. [Brezzi et al. 2007, 2005]) 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 course, 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 [2011] rely on an algebraic approach to define their discrete Laplacian and extend the MFD-based inner product stabilization [Brezzi et al. 2005] 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 $\Gamma^k,\ k \in \{0, 1\}$, 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

$$\Delta u = \mathrm{d}^* \mathrm{d} u. \tag{24}$$

In this context $\mathrm{d} : \Gamma^0 \longrightarrow \Gamma^1$ is the exterior derivative and $\mathrm{d}^* : \Gamma^1 \longrightarrow \Gamma^0$ the codifferential, which is defined as the adjoint of d with respect to the square integrable inner product [Rosenberg 1997]. They use the so-called *coboundary* operator as a discrete version of the smooth exterior derivative, with

$$(\mathrm{d}u)(h_{ij}) = u(j) - u(i) \tag{25}$$

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

$$\mathbb{L} = \mathrm{d}^* \mathrm{d} = \mathbf{M}^{-1}\mathbf{L} \tag{26}$$

with

$$\mathbf{L} = \mathbf{d}^{\mathsf{T}} \mathbf{M}_1 \mathbf{d}. \tag{27}$$

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

$$\mathbf{d}_{kl} = \begin{cases} -1 & l = i, \\ 1 & l = j, \\ 0 & \text{otherwise}, \end{cases} \tag{28}$$

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

$$\mathbf{M}_{ii} = \sum_{f \ni v_i} \frac{|f|}{n_f}, \tag{29}$$

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

$$\mathbf{A}_f = \mathbf{E}_f^\mathsf{T}\mathbf{B}_f, \tag{30}$$

meaning

$$|f| = \left\|\mathbf{a}_f\right\| = \left\|\frac{1}{2}\sum_{v_i \in f} \mathbf{x}_i \times \mathbf{x}_{i+1}\right\|. \tag{31}$$

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

$$\tilde{\mathbf{M}}_f = \frac{1}{|f|}\mathbf{B}_f\mathbf{B}_f^\mathsf{T}, \tag{32}$$

which was previously defined by Brezzi et al. [2005] 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

$$\mathbf{M}_f := \tilde{\mathbf{M}}_f + \mathbf{C}_{\bar{f}}\mathbf{U}_{\bar{f}}\mathbf{C}_{\bar{f}}^\mathsf{T}. \tag{33}$$

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

$$\left(\mathbf{C}_f^\mathsf{T}\mathbf{d}_f\mathbf{X}_f\right)_i = \left(\mathbf{C}_f^\mathsf{T}\mathbf{E}_f\right)_i = \left(\mathbf{C}_f^\mathsf{T}\mathbf{E}_{\bar{f}}\right)_i \overset{!}{=} 0, \tag{34}$$

which is equivalent to

$$\left(\mathbf{E}_{\bar{f}}^{\mathsf{T}}\mathbf{C}_f\right)_i \overset{!}{=} 0. \tag{35}$$

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

$$\mathbf{U}_{\bar{f}} := \lambda \mathbf{I}_f, \tag{36}$$

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

$$\mathbf{L}_f = \mathbf{d}_f^{\mathsf{T}}\mathbf{M}_f\mathbf{d}_f \tag{37}$$

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. [2020], 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) [Beirão da Veiga et al. 2013]. The following chapter will further elaborate on their approach. The main focus of de Goes et al. [2020] 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. [2020] circumvent this problem by evaluating the co-gradient operator

$$\nabla u^{\perp}(\mathbf{x}) := \mathbf{n}(\mathbf{x}) \times \nabla u(\mathbf{x}), \tag{38}$$

on which applying Stoke's theorem leads to

$$\iint_f \nabla u^{\perp}(\mathbf{x})\, \mathrm{d}A = \oint_{\partial f} u(\mathbf{x})\mathbf{t}(\mathbf{x})\, \mathrm{d}\mathbf{x}, \tag{39}$$

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 (39) 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:

$$\oint_{\partial f} u(\mathbf{x})\mathbf{t}(\mathbf{x})\,\mathrm{d}\mathbf{x} = \mathbf{E}_f^\mathsf{T}\mathbf{Avg}_f\mathbf{u}_f. \tag{40}$$

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

$$\left(\mathbf{Avg}_f\right)_{ij} = \begin{cases} \frac{1}{2} & \text{for } j = i \\ \frac{1}{2} & \text{for } j = (i+1) \bmod n_f \\ 0 & \text{otherwise.} \end{cases} \tag{41}$$

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 $[\mathbf{p}] \in \mathbb{R}^{3\times 3}$ with

$$[\mathbf{p}] = \begin{pmatrix} 0 & -p_3 & p_2 \\ p_3 & 0 & -p_1 \\ -p_2 & p_1 & 0 \end{pmatrix}, \tag{42}$$

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

$$\iint_f \nabla u^\perp(\mathbf{x})\,\mathrm{d}\mathbf{x} = \oint_{\partial f} \left(\mathbf{s}^\mathsf{T}\mathbf{x} + r\right)\mathbf{t}(\mathbf{x})\,\mathrm{d}\mathbf{x} \tag{43}$$

$$=\mathbf{E}_f^\mathsf{T}\mathbf{Avg}_f\left(\mathbf{X}_f\mathbf{s} + \mathbf{1}_f r\right) \tag{44}$$

$$=\mathbf{E}_f^\mathsf{T}\mathbf{B}_f\mathbf{s}, \tag{45}$$

which equally gives

$$\mathbf{E}_f^\mathsf{T}\mathbf{B}_f\mathbf{s} = \left(\iint_f [\mathbf{n}(\mathbf{x})]\,\mathrm{d}\mathbf{x}\right)\mathbf{s}, \tag{46}$$

since the term $\mathbf{E}_f^\mathsf{T}\mathbf{Avg}_f\mathbf{1}_f = \mathbf{E}_f^\mathsf{T}\mathbf{1}_f$, which is the sum of edge vectors and therefore zero. Equation (46) 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 (30)) 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

$$\mathbf{n}_f = \frac{\mathbf{a}_f}{|f|}, \tag{47}$$

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

$$\iint_f [\mathbf{n}_f]\nabla u(\mathbf{x})\,\mathrm{d}\mathbf{x}. \tag{48}$$

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

$$\mathbf{G}_f = -\frac{1}{|f|}[\mathbf{n}_f]\mathbf{E}_f^\mathsf{T}\mathbf{Avg}_f \tag{49}$$

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

$$\mathbf{V}_f^\sharp := \frac{1}{|f|}[\mathbf{n}_f]\left(\mathbf{B}_f^\mathsf{T} - \mathbf{c}_f \mathbf{1}_f^\mathsf{T}\right) \in \mathbb{R}^{3\times n_f}, \tag{50}$$

and

$$\mathbf{V}_f^\flat := \mathbf{E}_f\left(\mathbf{I} - \mathbf{n}_f \mathbf{n}_f^\mathsf{T}\right) \in \mathbb{R}^{n_f\times 3} \tag{51}$$

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

$$\mathbf{G}_f \mathbf{u}_f = \mathbf{V}_f^\sharp \mathbf{d}_f \mathbf{u}_f, \tag{52}$$

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 $\mathbf{K}$:

$$\langle \mathbf{x}, \mathbf{y}\rangle = \mathbf{x}^\mathsf{T}\mathbf{K}\,\mathbf{y} \quad \forall\, \mathbf{x}, \mathbf{y} \in V, \tag{53}$$

the flat operator $\flat$ is defined as

$$\mathbf{v}^\flat(\mathbf{u}) = \langle \mathbf{v}, \mathbf{u}\rangle = \mathbf{v}^\mathsf{T}\mathbf{K}\,\mathbf{u}, \tag{54}$$

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^* [Lee 1997]. 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 [de Goes et al. 2020] and therefore loses information. Inspired by the virtual element method [Beirão da Veiga et al. 2013], they mitigate the effect by defining a so-called *projection* operator

$$\mathbf{P}_f := \mathbf{I}_f - \mathbf{V}_f^\flat \mathbf{V}_f^\sharp \quad \in \mathbb{R}^{n_f\times n_f} \tag{55}$$

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. [2020] define their local inner product matrices acting on 1-forms as

$$\mathbf{M}_f := |f|\,\mathbf{V}_f^{\sharp\mathsf{T}}\mathbf{V}_f^{\sharp} + \lambda\,\mathbf{P}_f^{\mathsf{T}}\mathbf{P}_f, \tag{56}$$

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

$$\mathbf{L}_f = \mathbf{d}_f^{\mathsf{T}}\mathbf{M}_f\mathbf{d}_f. \tag{57}$$

4.3 Gradient and Divergence Operator

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

$$\Delta u = \operatorname{div}(\nabla u), \tag{58}$$

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

$$\mathbf{L}_f = \mathbf{D}_f\mathbf{G}_f. \tag{59}$$

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 [Alexa and Wardetzky 2011; de Goes et al. 2020] to satisfy Equation (59) would be to follow the discrete exterior calculus interpretation [Desbrun et al. 2005] 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

$$\mathbf{D}_f := \mathbf{d}_f^{\mathsf{T}}\mathbf{M}_f, \tag{60}$$

which gives the required equality. Note that this interpretation of the gradient differs from de Goes' geometric operator in Equation (49). 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

$$|f|\,\mathbf{G}_f^{\mathsf{T}}\mathbf{G}_f + \lambda\,\mathbf{d}_f\mathbf{P}_f^{\mathsf{T}}\mathbf{P}_f\mathbf{d}_f, \tag{61}$$

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

$$\mathbf{D}_f\mathbf{u}_f = |f|\,\mathbf{G}_f^{\mathsf{T}}\mathbf{u}_f, \tag{62}$$

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

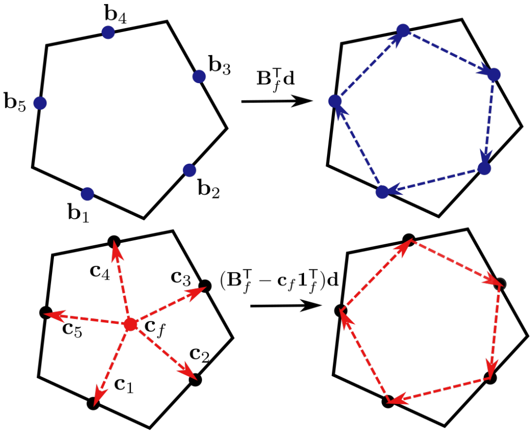


Fig. 2. Vectors involved in the inner product matrix for 1-forms for both Alexa and Wardetzky [2011] and de Goes et al. [2020] if computed on a planar polygon. The figure was taken from [Bunge and Botsch 2023].

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 $\mathbf{c}_i = \mathbf{b}_i - \mathbf{c}_f$. 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 2. 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

$$-[\mathbf{n}_f]^2 = \left(\mathbf{I} - \mathbf{n}_f\mathbf{n}_f^\mathsf{T}\right), \tag{63}$$

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. [2005] 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

$$u(\mathbf{x}) = \sum_{i=1}^{|\mathcal{V}|} u_i \varphi_i(\mathbf{x}). \tag{64}$$

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

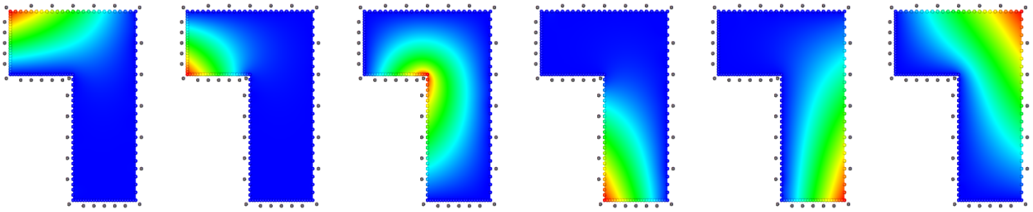


Fig. 3. Harmonic shape functions associated with the 6 nodes of a planar L-shaped polygon. The larger dots with the slight offset are the sampled center points $\mathbf{k}_i$ and the small dots on the boundary denote the collocation points $\mathbf{c}_j$ for Dirichlet boundary constraints. The figure was taken from [Martin et al. 2008].

$\mathbf{L}\mathbf{u} = \mathbf{f}$ with a Laplace matrix $\mathbf{L}$ that is defined as the integrated dot product of the gradients of the basis functions:

$$\mathbf{L}_{ij} = \int_{\mathcal{M}} \langle \nabla\varphi_i, \nabla\varphi_j \rangle. \tag{65}$$

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):

$$\varphi_i(\mathbf{x}_j) = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{otherwise.} \end{cases} \tag{66}$$

Furthermore, we want them to satisfy additional properties within each element of the mesh in order to guarantee convergence under refinement [Hughes 2012]:

(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

$$\sum_{i=1}^{n_f} \varphi_i(\mathbf{x}) = 1. \tag{67}$$

(3) They have to fulfill the linear reproduction property

$$\sum_{i=1}^{n_f} \varphi_i(\mathbf{x})\mathbf{x}_i = \mathbf{x} \tag{68}$$

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) [Bishop 2014; Floater 2003; Hormann and Sukumar 2008; Joshi et al. 2007; Ju et al. 2005], 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 [Chen and Gotsman 2016; Floater 2015] have already discussed the benefits and properties of these shape functions, which were also incorporated in polyhedral finite element methods [Manzini et al. 2014] for volume meshes.

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 [Hormann and Sukumar 2008] 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 [Chen and Gotsman 2016; Martin et al. 2008]. They can also be analyzed on arbitrary convex and non-convex polygons and polyhedra [Bishop 2014; Wicke et al. 2007]. 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 both the construction of polygonal and polyhedral finite element shape functions based on the work of Joshi et al. [2007] and Martin et al. [2008]. 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:

$$\begin{aligned} \Delta\varphi_i^f(\mathbf{x}) &= 0 \quad \text{for } \mathbf{x} \in f, \\ \varphi_i^f(\mathbf{x}) &= b_i(\mathbf{x}) \quad \text{for } \mathbf{x} \in \partial f. \end{aligned} \tag{69}$$

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) [Fairweather and Karageorghis 1998] 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 (69), 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

$$\psi(\|\mathbf{x}\|) = \log(\|\mathbf{x}\|), \tag{70}$$

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

$$\phi_i^f(\mathbf{x}) = \sum_{j=1}^{n} w_{ij}\psi\left(\left\|\mathbf{x} - \mathbf{k}_j\right\|\right), \tag{71}$$

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 have to be placed outside of the face's domain (see Figure 3), since their singularities lie at the centers $\mathbf{k}_j$. 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 (71) would not be able to exactly reproduce linear functions, violating the linear precision property. Therefore, Martin et al. add a linear polynomial

$$\varphi_i^f(\mathbf{x}) = \sum_{j=1}^{n} w_{ij}\psi(\|\mathbf{x} - \mathbf{k}_j\|) + \mathbf{s}_i^{\mathsf{T}}\mathbf{x} + r_i \tag{72}$$

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 (69). 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

$$\int_{\partial f} \left(\varphi_i^f(\mathbf{x}) - b_i(\mathbf{x})\right)^2 \approx \frac{1}{m} \sum_{j=1}^{m} \left(\varphi_i^f(\mathbf{c}_j) - b_i(\mathbf{c}_j)\right)^2. \tag{73}$$

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

$$\begin{pmatrix} \psi_{11} & \cdots & \psi_{1n} & \mathbf{c}_1^\mathsf{T} & 1 \\ \vdots & & \vdots & \vdots & \vdots \\ \psi_{m1} & \dots & \psi_{mm} & \mathbf{c}_m^\mathsf{T} & 1 \end{pmatrix} \begin{pmatrix} w_{i1} \\ \vdots \\ w_{in} \\ \mathbf{s}_i \\ r_i \end{pmatrix} = \begin{pmatrix} b_i(\mathbf{c}_1) \\ \vdots \\ b_i(\mathbf{c}_m) \end{pmatrix}, \tag{74}$$

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 [Golub and Van Loan 1996].

5.1.2 Harmonic Shape Functions on Polyhedral Meshes. From now on, additionally to the surface case, each of the introduced methods will have a volumetric extension to polyhedral meshes. As already mentioned, polyhedra allow for a variety of simulation applications, like fracture modeling [Bishop 2009], to be more flexible in their range of elements. However, boundary polygons in 3D are not necessarily planar and the increased connectivity between vertices lead to denser and more costly operators. Laplacians for volumetric meshes come therefore with their own individual set of challenges that have to be addressed.

The computation of a harmonic basis function for polyhedral meshes follows a similar approach as the previous section. The core principle to obtain a k-dimensional harmonic function is to impose $(k-1)$-dimensional harmonic coordinates as Dirichlet boundary conditions and proceed recursively. Therefore, given a polyhedron c with n_c vertices, we first approximate 2D shape functions for each of the polyhedron's faces with the system described in the previous section. Note that this requires the boundary faces of the polyhedron to be planar. These will give us the new boundary constraints b_i^c for the respective 3D shape function $\varphi_i^c : c \longrightarrow \mathbb{R}$ associated to vertex v_i, with kernels and collocation points uniformly sampled over the face of the polygonal cell c. The only remaining change to the linear system in Equation (74) is that the chosen fundamental solution for the 3D Laplace equation changes to

$$\psi^c(\|\mathbf{x}\|) = \frac{1}{\|\mathbf{x}\|}, \tag{75}$$

otherwise all steps remain the same.

5.1.3 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 (64). The discretizations of both stiffness and mass matrix needed for the Laplacian can be obtained through

$$(\mathbf{L}_f)_{ij} = \int_f \langle \nabla\varphi_i^f, \nabla\varphi_j^f \rangle \, \mathrm{d}\mathbf{x}. \tag{76}$$

and

$$(\mathbf{M}_f)_{ij} = \int_f \varphi_i^f \cdot \varphi_j^f \,\mathrm{d}\mathbf{x}. \tag{77}$$

Note that this process requires numerical integration, since the gradients of the harmonic shape functions are not constant. For volume meshes, the discretization process remains the same, but we integrate over the respective polyhedral cell c instead of a polygon face f.

5.2 Linear Virtual Refinement Method

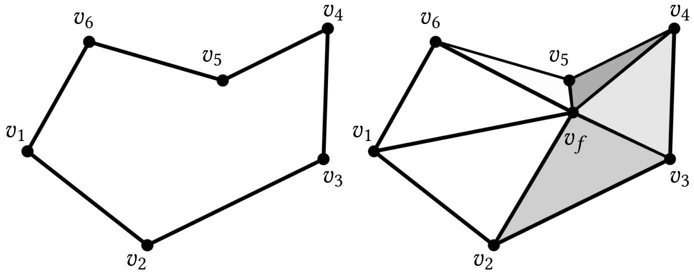


Fig. 4. Spanned triangle fan on the virtual mesh after inserting the vertex v_f. The figure was taken from [Bunge et al. 2020].

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. [2020] 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.

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

$$\mathbf{x}_f = \sum_{v_i \in f} w_i \,\mathbf{x}_i, \quad \text{with} \sum_{v_i \in f} w_i = 1. \tag{78}$$

The additional vertices allow Bunge et al. to construct a virtual triangle mesh $\mathcal{M}_\Delta$ by dividing each face into a triangle fan as shown in Figure 4. 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

$$\mathbf{P}_{ij}^f = \begin{cases} w_j & \text{for } i = n_f + 1 \\ \delta_{ij} & \text{otherwise,} \end{cases} \tag{79}$$

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

$$\mathbf{P}_{ij} = \begin{cases} 1 & \text{if } i = j \text{ and } i \leq |\mathcal{V}| \\ w_{kj} & \text{if } i = |\mathcal{V}| + k \text{ and } v_j \in f_k \\ 0 & \text{otherwise,} \end{cases} \tag{80}$$

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 $\mathbf{M}_\Delta$ and $\mathbf{L}_\Delta$ on the virtual triangle mesh $\mathcal{M}_\Delta$ (see Equation (11) and Equation (12)) and define the matrices for the original polygon mesh as

$$\mathbf{L} = \mathbf{P}^\mathsf{T}\mathbf{L}_\Delta\mathbf{P} \tag{81}$$

and

$$\mathbf{M} = \mathbf{P}^\mathsf{T}\mathbf{M}_\Delta\mathbf{P}. \tag{82}$$

As for the Laplacian, they compute the gradient and divergence operators $\mathbf{G}_\Delta$ and $\mathbf{D}_\Delta$ on $\mathcal{M}_\Delta$ by using the simplicial definitions in Equation (15) and (16) to express the polygon operators as

$$\mathbf{G} = \mathbf{G}_\Delta\mathbf{P} \tag{83}$$

and

$$\mathbf{D} = \mathbf{P}^\mathsf{T}\mathbf{D}_\Delta = \mathbf{P}^\mathsf{T}\mathbf{G}_\Delta^\mathsf{T}\hat{\mathbf{M}}_\Delta. \tag{84}$$

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. [2020] 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 is guaranteed to lie within 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

$$\mathbf{w}_f = \arg\min_{\mathbf{w}} \sum_{i=1}^{n_f} \text{area}\left(\mathbf{x}_i, \mathbf{x}_{i+1}, \sum_{j=1}^{n_f} w_j\ \mathbf{x}_j\right)^2 \tag{85}$$

$$\text{such that} \sum_{j=1}^{n_f} w_j = 1. \tag{86}$$

However, for faces with valence higher than 3 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.

5.2.1 Finite Element Shape Functions. Drawing the connecting to traditional FEM methods, the previously defined prolongation weights allow Bunge et al. 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 φ_i^Δ 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

$$\varphi_i^f = \varphi_i^\Delta + w_i\varphi_f^\Delta, \quad i = 1, \dots, n_f. \tag{87}$$

Here, φ_f^Δ 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 5.1.3 would lead to the same discretized stiffness and mass matrices as defined in Equation (81) and Equation (82). 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 $\{\varphi_1^\Delta, \dots \varphi_{|\mathcal{V}_\Delta|}^\Delta\}$ defined on the refined triangle mesh allows us to reinterpret the construction of the stiffness matrix $\mathbf{L}$ in Equation (81) as

$$\mathbf{L} = \mathbf{d}_\mathcal{E}^\mathsf{T} \star^1 \mathbf{d}_\mathcal{E}, \tag{88}$$

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

$$(\mathbf{d}_\mathcal{E})_{kl} = \begin{cases} -1 & l = i \\ 1 & l = j \\ 0 & \text{otherwise} \end{cases} \tag{89}$$

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 (28), 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. 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

$$\varphi_j = \sum_{i=1}^{|\mathcal{V}_\Delta|} \mathbf{P}_{ij} \varphi_i^\Delta. \tag{90}$$

By construction, these bases form a partition of unity. Therefore, they can be used to define a set of polygon Whitney bases [Arnold et al. 2006; Whitney 1957] for 1-forms, with

$$\varphi_{ij} = \varphi_i \cdot \mathbf{d}_\mathcal{E}\varphi_j - \varphi_j \cdot \mathbf{d}_\mathcal{E}\varphi_i = \sum_{l=1}^{|\mathcal{V}_\Delta|} \sum_{k=1}^{l} \mathbf{P}_{ki}\mathbf{P}_{lj}\varphi_{kl}^\Delta, \tag{91}$$

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 $\mathbf{P}^1 \in \mathbb{R}^{|\mathcal{E}_\Delta|\times|\mathcal{E}|}$ as

$$\mathbf{P}^1_{(ij)(kl)} = \mathbf{P}_{ik}\mathbf{P}_{jl}, \tag{92}$$

with (ij) indicating the row associated to the edge $e_{ij}^\Delta \in \mathcal{E}_\Delta$ 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 $\star_\Delta^1$ on the refined triangle mesh, giving us

$$\mathbf{M}_1 = \star^1 = \left(\mathbf{P}^1\right)^\mathsf{T} \star_\Delta^1 \mathbf{P}^1 \tag{93}$$

and

$$\mathbf{L} = \mathbf{d}_\mathcal{E}^\mathsf{T} \left(\mathbf{P}^1\right)^\mathsf{T} \star_\Delta^1 \mathbf{P}^1 \mathbf{d}_\mathcal{E}. \tag{94}$$

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 [2011] and de Goes et al. [2020], which remains to be investigated.

5.2.3 Laplacian on Volume Meshes. The previously described method can be intuitively extended to arbitrary polyhedral meshes, but instead of virtual triangles, the mesh $\mathcal{M}$ will be refined into virtual *tetrahedra*. The first steps are analogous to the surface case, meaning that all faces $f \in \mathcal{F}$ of a given polyhedron $c \in C$ are refined into triangles with virtual vertices placed at the point that minimizes the sum of squared triangle areas (see Equation (85)). To span the virtual tetrahedra, Bunge et al. [2021] introduce an additional vertex $\mathbf{x}_c$ inside of each cell, which is, similar to the surface case, the affine combination of the cells' refined faces vertex positions

$$\mathbf{x}_c = \sum_{v_i \in \mathcal{V}(\partial c)} w_i \, \mathbf{x}_i, \quad \text{with} \sum_{v_i \in \mathcal{V}(\partial c)} w_i = 1. \tag{95}$$

Here, $\mathcal{V}(\partial c)$ refers to the set of vertex indices that lie on the refined boundary of the polyhedron c. The position of $\mathbf{x}_c$ is defined as the minimizer of the sum of squared tetrahedron volumes

$$\sum_{t_{ijk} \in \partial c} \operatorname{vol}\left(\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k, \mathbf{x}_c\right)^2, \tag{96}$$

with t_{ijk} being the refined triangles along the cell's boundary. As for surfaces, this minimization problem can be expressed with respect to a set of affine weights $\mathbf{w}_c \in \mathbb{R}^{|\mathcal{V}(\partial c)|}$ and assembled into a local prolongation matrix $\mathbf{P}^c$. The only real difference of this approach is that the global prolongation $\mathbf{P}$ is now divided into a two-step process, with the "surface" prolongation matrix $\mathbf{P}_F$ inserting the virtual face points for all $f \in \mathcal{F}$ and $\mathbf{P}_C$ the cell points for all $c \in C$, respectively, giving us

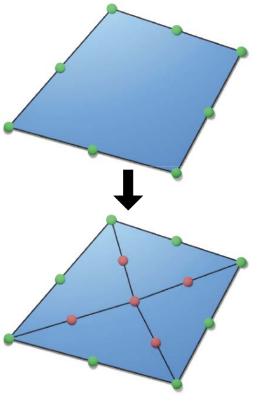
$$\mathbf{P} = \mathbf{P}_C \mathbf{P}_F. \tag{97}$$

The polyhedral stiffness and mass matrix are then obtained as in (81) and (82), with the slight change that the refined matrices are the volumetric discretizations of the cotan formula (see Section 3.1).

5.3 Quadratic Virtual Refinement Method

All of the previously discussed FEM methods share a common characteristic: They define piecewise linear basis functions. This means that the respective shape functions are associated with the vertex nodes of the given polygon. Using this kind of bases is common practice in geometry processing due to their balanced trade-off between accuracy and efficiency. Nevertheless, it is well-acknowledged that higher-order shape functions typically lead to more accurate results and faster convergence rates [Schneider et al. 2022]. However, this advantage comes at a cost, as these functions have additional degrees of freedom that result in denser, larger matrices, which consequently increase computational complexity. Nonetheless, several works investigated and utilized higher-order polynomials in the context of arbitrary polygonal and polyhedral meshes. For instance, Beirao et al. [2017] analyzed them in the context of the virtual element method for three-dimensional problems. Schneider et al. [2018], on the other hand, used a spline-based approach to achieve higher convergence rates on hex-dominant meshes. Furthermore, the 2D polygonal finite element basis introduced by Aurojyoti et al. [2019] achieves global C^1 smoothness by elevating the degree of generalized barycentric coordinates through Bernstein-Bezier functions. In the following, we will review the method introduced by Bunge et al. [2022], which extends the previously defined linear polygon shape functions (Section 5.2) to quadratic Lagrange elements on polygonal and polyhedral meshes.

5.3.1 Quadratic Basis function for Polygons. As in the linear setting, assume we are given a polygon $(\mathbf{x}_1, \ldots, \mathbf{x}_{n_f}) \subset \mathbb{R}^3$ and its corresponding virtual face vertex v_f, still defined as the minimizer of the sum of squared triangle areas. In order to define quadratic shape functions, it is no longer



sufficient to only consider basis functions associated with the original nodes of the polygon. Instead, one has to introduce additional degrees of freedom at the edges, whether real or virtual, as illustrated in the inset figure (taken from [Bunge et al. 2022]). Consequently, instead of the previous n_f vertices of the polygon, higher-order shape functions require $2n_f$ degrees of freedom at the polygon boundary, which the authors refer to as *coarse* nodes C (depicted in green). Furthermore, along with the virtual face vertex, they introduce n_f additional virtual nodes positioned on the interior edges, leading to $n_f + 1$ virtual degrees of freedom denoted by $\mathcal{K}$ (shown in red). The union of these sets is referred to as *fine* nodes $\mathcal{F} = C \bigcup \mathcal{K}$ and their locations are denoted by $\mathbf{x}_i \in \mathbb{R}^3$.

On the virtual triangulation, the linear Lagrange basis functions (Equation (10)) are exchanged for the quadratic Lagrange basis ψ_i associated with the fine nodes $\mathcal{F}$. Since the additional virtual edge nodes are involved in the construction of the basis functions, Bunge et al. defined $n_f + 1$ sets of weights $\mathbf{w}_j = (w_{ij}) \in \mathbb{R}^{n_f}$ to redistribute the virtual degrees of freedom j to coarse nodes i, forming local shape functions φ_i of the form

$$\varphi_i = \psi_i + \sum_{j \in \mathcal{K}} w_{ij} \psi_j \quad \text{for } i \in C. \tag{98}$$

The construction of the prolongation matrix is analogous to the linear case described in Equation (79). The only difference are the additional virtual vertices. The local prolongation matrix is for a face f is given by

$$\mathbf{P}^f_{ij} = \begin{cases} \delta_{ij} & i \in C, \\ w_{ji} & i \in \mathcal{K}, \end{cases}$$

and the per-face prolongation matrices are assembled into the global prolongation matrix $\mathbf{P}$.

Interpolation. The shape functions constructed according to (98) conform naturally to the Lagrange interpolation property $\varphi_i(\mathbf{x}_j) = \delta_{ij}$ for all coarse nodes $j \in C$, since the fine basis satisfies $\psi_i(\mathbf{x}_j) = 0$ for all $i \in \mathcal{K}$ and $j \in C$. This is important, since it ensures C^0 continuity across polygon edges: The function values for each shape function are either zero at the polygon edge or the unique quadratic function satisfying the Lagrange interpolation conditions at the three nodes along that edge. Consequently, the prolongation weights w_{ij} can be chosen independently for each polygon. Within each element, the shape functions φ_i are also able to retain C^0 continuity, since they are linear combinations of the C^0 functions ψ_i.

However, this is not the case for C^1 continuity. In general, the functions are not necessarily C^1 across the virtual edges connecting polygon vertices to the virtual vertex. This stems from the fact that the standard triangle quadratic shape functions ψ_i typically lack C^1 continuity across element edges. Left unresolved, as discussed in the original paper, this issue could present a problem. In order to guarantee the desired cubic convergence rate for the Laplacian discretized with these shape functions, it is necessary to maintain C^1 continuity within the element. Bunge et al. [2022] address this concern with a specific set of prolongation weights, which will be discussed next.

Variational energy minimization. Since the fine shape functions ψ_i are generally not C^1 across virtual edges, their linear combination is not guaranteed to be either. The new prolongation weights defined by Bunge et al. [2022] replace the L_2 minimizing constraint from their linear approach with the squared gradient difference integrated along all virtual edges, summed over all coarse

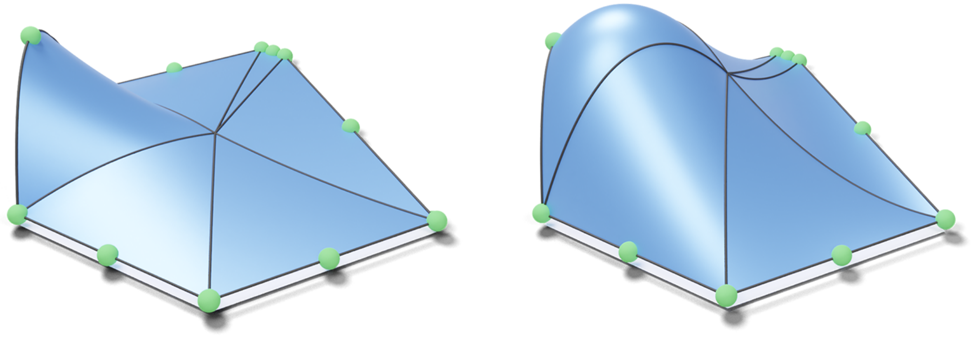


Fig. 5. Prolongation weights computed using a regularizer on their norm lead to very local basis functions at the cost of smoothness (left). The squared gradient difference energy explicitly leads to basis functions that prioritize smoothness across internal edges (right). The figure was taken from [Bunge et al. 2022].

basis functions. This energy helps to minimize the discontinuities between the shape functions' gradients along the virtual edges, as highlighted in Figure 5, leading to significantly smoother shape functions. They are obtained by solving the quadratic optimization problem

$$W = \arg\min_{W} \sum_{i \in C} \sum_{\sigma \in \mathcal{E}^*} \int_{\sigma} \|\nabla_{\sigma}^{+} \varphi_i - \nabla_{\sigma}^{-} \varphi_i\|^2 \, \mathrm{d}\sigma \tag{99}$$

with respect to the set of prolongation weights $W = \{w_{ij}\}$. Here $\mathcal{E}^*$ is the subset of edges in the virtual triangulation that are incident to the virtual vertex ($\mathcal{E}^* = \{(\mathbf{v}_0, \mathbf{v}_i)\}_{1 \le i \le n}$). The operator ∇_{σ}^{+} represents the gradient with respect to the right triangle of the edge and the operator ∇_{σ}^{-} with respect to the left one. Furthermore, two additional constraints are included in the system. Just as in the linear case, partition of unity

$$\sum_{i \in C} w_{ij} = 1 \quad \text{for } j \in \mathcal{K} \tag{100}$$

must be satisfied. Additionally, the positions of the virtual nodes have to be reproduced through affine combinations of the coarse nodes

$$\mathbf{x}_j = \sum_{i \in C} w_{ij} \mathbf{x}_i \quad \text{for } j \in \mathcal{K}. \tag{101}$$

In cases where a particular set of weights results in an energy of zero, all basis functions confined to the polygon attain C^1 continuity.

For higher-order shape functions, additionally to constant and linear precision (see Equation (67) and Equation (68)), the ability to accurately reproduce quadratic functions is another crucial criterion that has to be be met. All three conditions together ensure the desired faster convergence rate. The basis functions obtained with the alternative prolongation weights are able to satisfy all of these properties and reproduce the original quadratic Lagrange basis functions on triangles. However, we direct the reader to the original work [Bunge et al. 2022] for proofs and demonstrations.

5.3.2 Quadratic Basis Function for Polyhedra. As in Section 5.2.3, the energy described in Equation (99) can be extended to volume meshes by considering virtual tetrahedra instead of virtual triangles. The virtual face vertices on the polyhedron's boundary remain the respective minimizer of the sum of squared triangle areas: The position of the virtual polyhedron vertex minimizes the sum of squared tetrahedra volumes (see Equation (96)) of the resulting tessellation. The only difference

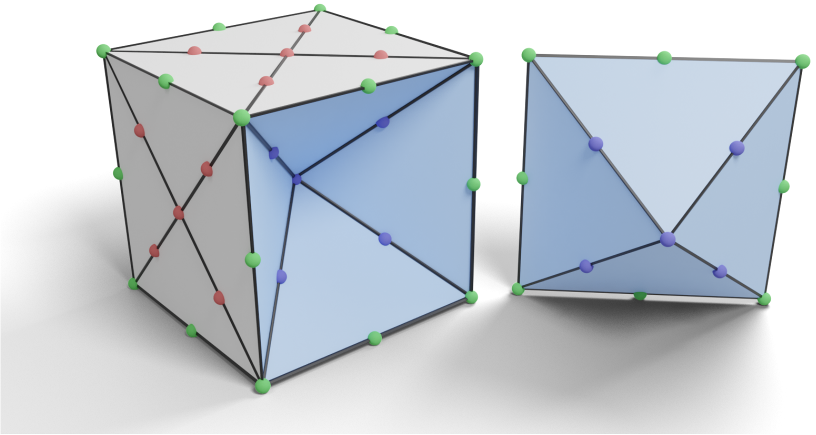


Fig. 6. Node positions for a refined cube. All faces are split into triangles using virtual vertices. The volume is decomposed into tetrahedra by introducing a central virtual vertex which is connected to all face triangles. Figure taken from [Bunge et al. 2022].

are additional midpoint nodes introduced at each edge of the polyhedron. As in the surface case, they are necessary degrees of freedom for the 3D quadratic Lagrange basis (see Figure 6). The authors distinguish between three types of nodes, as indicated by the colors in Figure 6. Level 1 nodes (green) are degrees of freedom that are defined with respect to the polyhedron itself, whereas level 2 (red) and level 3 (blue) nodes depend on virtual vertices. The final shape functions are defined at the coarse nodes (green), after the prolongation process redistributes the basis functions ψ_j associated with the virtual nodes (red and blue). The volumetric equivalent to the quadratic energy (99) is conceptually very similar to its surface counterpart. Rather than integrating along virtual edges, the squared gradient difference of the shape functions is integrated over virtual triangles shared by two tetrahedra. Consequently, triangles that tessellate the boundary of the polyhedron are not considered. The complete energy, including constraints, is defined as follows

$$\begin{gathered} W = \arg\min_{W} \sum_{i \in C} \sum_{\sigma \in \mathcal{T}^*} \int_{\sigma} \|\nabla_{\sigma}^{+}\varphi_i - \nabla_{\sigma}^{-}\varphi_i\|^2 \, \mathrm{d}\sigma \\ \text{s.t. constraints (100) and (101) are satisified.} \end{gathered} \tag{102}$$

In this context, $\mathcal{T}^*$ denotes a subset of triangles in the virtual tetrahedralization of the polyhedron, each of which is incident upon the virtual polyhedral vertex. Here, the symbols ∇_{σ}^{+} and ∇_{σ}^{-} represent the gradients of φ_i on the tetrahedron to the right and left of the shared face σ, respectively.

However, in contrast to the surface scenario, minimizing the system in its current state would result in shape functions that no longer satisfy C^0 continuity between adjacent polyhedral cells. The underlying cause for this are the shared virtual vertices situated on the respective boundary face of the cells. The weights that minimize Equation (102) redistribute the influence of virtual nodes by integrating the gradient mismatch over triangles *within a polyhedron*, while disregarding triangles located on the element's boundary. Consequently, the redistribution of the virtual face vertices is dependent on the specific cell over which the prolongation weights are computed. As a result, there is no inherent requirement for the redistribution to be identical between adjacent polyhedra. In the case of polygons, no virtual node shares a common edge with an adjoining element. Therefore, their associated fine basis functions are defined within a single polygon and the C^0 continuity between adjacent faces is not affected.

In order to address the problem, Bunge et al. [2022] split the prolongation into two steps:

(1) For each boundary face, initial prolongation weights are computed that satisfying Equation (99), distributing level 2 nodes to level 1 nodes.
(2) Then, Equation (102) is solved, where the prolongation weights obtained in the first step are included as hard constraints.

This amounts to first solving for a variational basis on the boundary of the polyhedron, and then adjusting the values of the basis functions *in the interior of the polyhedron* so as to minimize the cross-edge gradient difference there as well. By construction, the per-face prolongation weights computed in the first step, are defined by optimizing for C^1 continuity within boundary faces. Thus, two face-adjacent polyhedra necessarily distribute a level 2 node in the same way and the derived basis is guaranteed to be continuous. As in the 2D case, the quadratic Lagrange basis functions are reproduced when computed on arbitrary tetrahedra, independent of the choice of virtual vertex [Bunge et al. 2022].

5.4 Key Outcomes

One of the main benefits of the presented FEM methods, in contrast to the operators described in the Mimetic Polygon Laplacian section, is that they can all be applied to surface as well as volume meshes. However, the harmonic shape functions are restricted to planar polygons and boundary faces, while the linear and quadratic virtual refinement methods are able to deal with non-planar elements. Still, the extension to volume meshes and the option to use higher-order shape functions lead to an enhanced flexibility for the computer graphics community and its range of applications. 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 polytopes 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 [1961; 1955] 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 [Rapp 2017]. 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 [Lipnikov et al. 2014], 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. [2021] used a special polygonal variant of the FV, called Discrete Duality Finite Volume (DDFV) [Coudière and Hubert 2011; Domelevo and Omnés 2005; Hermeline 2000, 2009] and combined this technique with their previously described virtual triangle refinement to define a gradient, divergence and Laplacian operator for polygonal and polyhedral meshes.

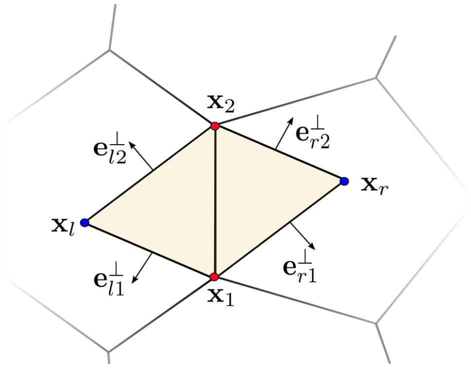

$\mathbf{x}_2$
$\mathbf{e}_{l2}^{\perp}$
$\mathbf{e}_{r2}^{\perp}$
$\mathbf{x}_r$
$\mathbf{x}_l$
$\mathbf{e}_{l1}^{\perp}$
$\mathbf{e}_{r1}^{\perp}$
$\mathbf{x}_1$

Fig. 7. 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 $\mathbf{e}_{ij}\perp$ are orthogonal to the diamond edges and involved construction of the gradient operator. The figure was taken from [Bunge et al. 2021].

6.1 Discrete Duality Finite Volume Method

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 7). 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

$$\iint_D \nabla u(\mathbf{x})\,\mathrm{d}\mathbf{x} = \int_{\partial D} u(\mathbf{x})\,\mathbf{n}(\mathbf{x})\,\mathrm{d}\mathbf{x} \tag{103}$$

$$= \sum_{(i,j)\in\partial D} \frac{\mathbf{e}_{ij}^{\perp}}{\left\|\mathbf{e}_{ij}\right\|} \int_0^1 \left\|\mathbf{e}_{ij}\right\| \left((1-t)u_i + tu_j\right)\,\mathrm{d}t \tag{104}$$

$$= \sum_{(i,j)\in\partial D} \mathbf{e}_{ij}^{\perp} \frac{u_i + u_j}{2}. \tag{105}$$

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 [Andreianov et al. 2012; Lipnikov et al. 2014]. This property is an essential part of the finite volume setting.

6.2 Diamond Laplace for Surface Meshes

Bunge et al. [2021] 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 (80)). 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 (105).

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

$$\mathbf{G}_D(:, i) = \frac{1}{2\,|D|} \sum_{(i,j)\in\partial D} \tilde{\mathbf{e}}_{ij}^{\perp}, \tag{106}$$

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 (80)), we obtain a gradient operator for the original polygon mesh through

$$\mathbf{G} = \mathbf{G}_\diamond \mathbf{P}. \tag{107}$$

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. [2021] define their diamond divercence operator as

$$\mathbf{D} = \mathbf{P}^\mathsf{T} \mathbf{G}_\diamond^\mathsf{T} \hat{\mathbf{M}}_\diamond \tag{108}$$

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

$$\mathbf{L} = \mathbf{D}\mathbf{G} = \mathbf{P}^\mathsf{T} \mathbf{G}_\diamond^\mathsf{T} \hat{\mathbf{M}}_\diamond \mathbf{G}_\diamond \mathbf{P}, \tag{109}$$

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

$$\mathbf{M} = \mathbf{P}^\mathsf{T} \mathbf{M}_\diamond \mathbf{P}. \tag{110}$$

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:

$$(\mathbf{M}_\diamond)_{ii} = \begin{cases} \sum_{D\ni v_i} \frac{|D|}{4} & \text{if } v_i \in \mathcal{V} \\ \sum_{D\ni v_i} \frac{|D|}{4} & \text{if } v_i \in \mathcal{V}^* \\ 0 & \text{otherwise.} \end{cases} \tag{111}$$

6.2.1 Diamond Laplace for Volume Meshes. As in Section 5.2.3, Bunge et al. [2021] define a Laplacian operator for general polyhedra by refining the given mesh $\mathcal{M}$ into a virtual tetrahedral mesh $\mathcal{M}_\triangle$. However, as for the surface case, they once again interpret the added virtual cell vertices as the required dual vertices $\mathcal{V}^*$ in order to divide the mesh into volumetric diamond cells $\mathcal{D}$. Since the refinement divides the original mesh into virtual tetrahedra, they can consider any combination of cells $c_\triangle \in C_\triangle$ as a region Ω bounded by a triangulated surface with $n_c^\triangle$ vertices. Therefore, Bunge

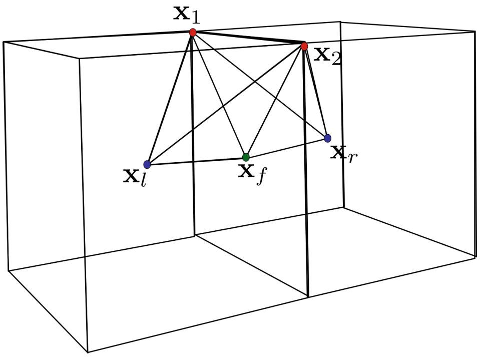


Fig. 8. A minimal diamond associated with the primal edge $\mathbf{x}_1, \mathbf{x}_2$ on the face f with virtual face vertex $\mathbf{x}_f$. The virtual cell vertices $\mathbf{x}_l$ and $\mathbf{x}_r$ of the adjacent polyhedra form the respective tips of the diamond. The figure was taken from [Bunge et al. 2021].

et al. [2021] can once again discretize the gradient of a function u over Ω with the help of Stokes' theorem:

$$\iiint_{\Omega} \nabla u(\mathbf{x})\,\mathrm{d}\mathbf{x} = \iint_{\partial\Omega} u(\mathbf{x})\mathbf{n}(\mathbf{x})\,\mathrm{d}\mathbf{x} \tag{112}$$

$$= \sum_{t_{ijk}\in\partial\Omega} \frac{\mathbf{a}_{ijk}}{\left\|\mathbf{a}_{ijk}\right\|} \int_0^1 \int_0^t \left\|\mathbf{a}_{ijk}\right\| \left((1-s-t)u_i + su_j + tu_k\right)\,\mathrm{d}s\,\mathrm{d}t \tag{113}$$

$$= \sum_{t_{ijk}\in\partial\Omega} \mathbf{a}_{ijk} \frac{u_i + u_j + u_k}{3}, \tag{114}$$

where

$$\mathbf{a}_{ijk} = \frac{1}{2}(\mathbf{x}_j - \mathbf{x}_i) \times (\mathbf{x}_k - \mathbf{x}_i) \tag{115}$$

denotes the outward pointing face normal of the boundary triangle t_{ijk} whose magnitude equals $\left|t_{ijk}\right|$. Therefore, the column-wise matrix representation of ∇u over Ω can be written as

$$\mathbf{G}_{\Omega}(:, i) = \frac{1}{3\,|\Omega|} \sum_{t_{ijk}\in\partial\Omega} \mathbf{a}_{ijk} \in \mathbb{R}^{3\times n_c^{\Delta}}, \tag{116}$$

which is consistent with the 2D discretization (see Equation (106)). The only remaining question is how to define suitable volumetric diamond cells. Here, Bunge et al. define a so called *minimal* diamond consisting of two adjacent virtual tetrahedra. The tips are formed by the virtual cell vertices v_l and v_r of the adjacent polyhedra and its base triangle $(\mathbf{x}_f, \mathbf{x}_i, \mathbf{x}_j)$ is defined by a shared edge e_{ij} and its connection to the virtual face vertex v_f of the shared face f (see Figure 8). This seems unintuitive on the first glance, since the larger region spanned by all edges of f connected to v_l and v_f equally forms an integrable region independent of the faces surface. However, the authors point out that this choice for the diamond cell could lead to *spurious modes*, which describes a Laplacian with a kernel that contains more than constant functions and can cause severe numerical artifacts. Spurious modes are a known limitation to some DDFV methods like [Hermeline 2009], as discussed by [Andreianov et al. 2013]. The focus on the null space of a discrete operator and avoiding numerically polluting modes is also a very important aspect of the MFD method [Lipnikov et al. 2014] and, as previously discussed, motivate the stabilization terms of the inner products introduced in the works of [Alexa and Wardetzky 2011] and [de Goes et al. 2020] explained in

Section 4.1.2 and Section 4.2.2. Using Equation (116), Bunge et al. define a local gradient $\mathbf{G}_D \in \mathbb{R}^{3\times 5}$ per minimal diamond cell D that can be assembled into a global gradient matrix $\mathbf{G}_\diamond$ defined on the refined mesh. Combined with the volume prolongation matrix $\mathbf{P}$ defined in Equation (97) we get a polyhedral gradient operator

$$\mathbf{G} = \mathbf{G}_\diamond \mathbf{P}, \tag{117}$$

its compatible divergence operator

$$\mathbf{D} = \mathbf{P}^\mathsf{T} \mathbf{G}_\diamond \hat{\mathbf{M}}_\diamond, \tag{118}$$

with $\hat{\mathbf{M}}_\diamond$ being a diagonal matrix containing the diamond volumes, and finally the generalized stiffness matrix

$$\mathbf{L} = \mathbf{D}\mathbf{G}. \tag{119}$$

The mass matrix $\mathbf{M}$ is obtained as in the surface case:

$$\mathbf{M} = \mathbf{P}^\mathsf{T} \mathbf{M}_\diamond \mathbf{P}, \tag{120}$$

but with $\mathbf{M}_\diamond$ now distributing the volumes of the minimal diamonds among the involved vertices and being defined as

$$(\mathbf{M}_\diamond)_{ii} = \begin{cases} \sum_{D \ni v_i} \frac{|D|}{6} & \text{if } v_i \in \mathcal{V} \\ \sum_{D \ni v_i} \frac{|D|}{6} & \text{if } v_i \in \mathcal{V}_\triangle \setminus (\mathcal{V} \cup \mathcal{V}^*) \\ \sum_{D \ni v_i} \frac{|D|}{4} & \text{if } v_i \in \mathcal{V}^* \\ 0 & \text{otherwise.} \end{cases} \tag{121}$$

6.3 Key Outcomes

As for the FEM methods, the diamond Laplacian can be applied to both surface and volume 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 and polyhedral Laplacians based on the properties established by Wardetzky et al. [2007] (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 [Alexa and Wardetzky 2011] and [de Goes et al. 2020] 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 [Martin et al. 2008], [Bunge et al. 2020] and [Bunge et al. 2022].

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

$$\mathbf{L} = \mathbf{D}\mathbf{G} = \mathbf{P}^{\mathsf{T}}\mathbf{G}_{\diamond}^{\mathsf{T}}\hat{\mathbf{M}}_{\diamond}\mathbf{G}_{\diamond}\mathbf{P} \tag{122}$$

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

7.2 Locality

All methods, with the exception of [Bunge et al. 2021], define their involved matrices or shape functions locally per polygon or polyhedron. 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/cell vertices of the polygons/polyhedrons where v_i itself is a part of, which leads to a local neighborhood. For the higher-order Lapalacian introduced by [Bunge et al. 2022], the neighborhood also includes the respective edge nodes. 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/cell 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 cell/face sharing a face/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:

$$\tilde{\mathbf{L}}_f\mathbf{X}_f = \mathbf{d}_f^{\mathsf{T}}\tilde{\mathbf{M}}_f\mathbf{d}_f\mathbf{X}_f, \tag{123}$$

where

$$\tilde{\mathbf{L}}_f := \mathbf{d}_f^{\mathsf{T}}\tilde{\mathbf{M}}_f\mathbf{d}_f \tag{124}$$

with $\tilde{\mathbf{M}}_f$ being the inner product matrix defined by Brezzi et al. [2005] (see Equation (32)), and

$$\mathbf{d}_f^{\mathsf{T}}\mathbf{C}_{\bar{f}}\mathbf{U}_{\bar{f}}\mathbf{C}_{\bar{f}}^{\mathsf{T}}\mathbf{d}_f\mathbf{X}_f. \tag{125}$$

As shown in Equation (34), 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:

$$\nabla_{\mathbf{x}_i}|f| = \left(\tilde{\mathbf{L}}_f\mathbf{X}_f\right)_i. \tag{126}$$

Therefore, the expression $(\mathbf{L}\mathbf{X})_i$ 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 [de Goes et al. 2020], 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:

$$\mathbf{P}_f \mathbf{d}_f \mathbf{X}_f = \mathbf{P}_f \mathbf{E}_f. \tag{127}$$

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

$$\mathbf{V}_f^\flat = \mathbf{E}_f \left(\mathbf{I} - \mathbf{n}_f \mathbf{n}_f^\mathsf{T}\right). \tag{128}$$

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 (68). 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 [Zienkiewicz et al. 2013]. 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 (72) is crucial to guarantee *exact* linear precision, and therefore this property, independent of the number of chosen kernels.

Linear Virtual Refinement Method. For volume meshes, the linear precision property is satisfied if for all linear functions $\mathbf{u}$ sampled on the mesh, the Laplacian $\mathbf{Lu}$ is exact at the interior vertices of the mesh [Alexa et al. 2020]. As for surfaces, an equivalent expression can be formulated as

$$(\mathbf{LX})_i = 0, \tag{129}$$

which has to be satisfied for all interior vertices v_i. For surface meshes, given that the elements surrounding the interior vertex v_i are planar, the refined triangles of these primitives are planar as well, 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

$$(\mathbf{L}_\triangle \mathbf{u}_\triangle)_i = 0 \tag{130}$$

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 $(\mathbf{Lu})_i = 0$. Given that the cotan Laplacian for volume meshes also satisfies linear precision, the same arguments apply for polyhedral meshes.

Quadratic Virtual Refinement Method. As proven in the original paper, the quadratic shape functions defined by Bunge et al. [2022] are linearly precise, which is a direct consequence of the reproduction property enforced as constraint (see Equation (101)). Therefore, Bunge et al.'s Laplacian satisfies this property as well.

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 [Chuang et al. 2009; Domelevo and Omnés 2005; Hermeline 2009], 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 [Bunge et al. 2020, 2022; Martin et al. 2008], 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 [2011] and de Goes et al. [2020], 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 for both surface and volume meshes. 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. [2021], 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 [Quenjel et al. 2020], which can not be rectified by applying the prolongation matrices. The same holds for the quadratic virtual refinement method. As stated by Höhn and Mittelmann [1981], quadratic Lagrange elements are only able to satisfy the discrete maximum principle under extremely restrictive assumptions on the mesh and can not be recovered by the prolongation step. 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 both surface and volume meshes. We are interested in the influence of the respective parameters λ described by Alexa and Wardetzky [2011] and de Goes et al. [2020] 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. [2021; 2020] and use the squared triangle area and squared tetrahedra volume minimizer, respectively. For the harmonic shape functions [Martin et al. 2008], 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 and used the recommended 3/9 points per edge and 10/30 per face for the volumetric tessellations.

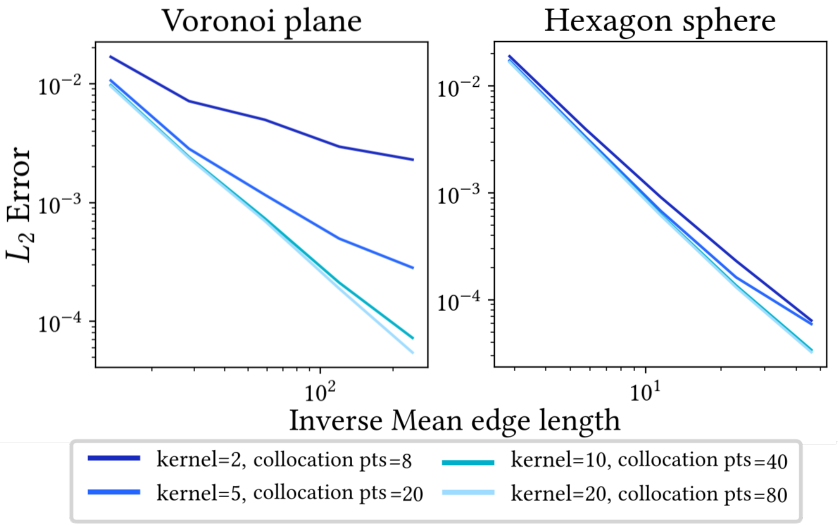


Fig. 9. Effect of the number of chosen kernels and collocation points on the convergence behavior of the harmonic shape functions on the Voronoi plane (left) and hexagon sphere (right). The shown L_2 errors refer to the Poisson systems solved for Franke's test function on planar grids and for the sphercial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

The choice was also influenced by the numerical costs involved in using more samples, explaining the lower sample sizes for volume meshes. The triangles and tetrahedra 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. The figures in the evaluation will feature certain labels for some methods to improve clarity. We will refer to the the harmonic shape functions by Martin et al. [2008] as "Harmonic". The linear and quadratic virtual refinement methods by Bunge et al. [2020; 2022] will be referred to as "Lin. Virt. Ref" and "Quad. Virt. Ref" respectively and the Diamond Laplacian by Bunge et al. [2021] will be called "Diamond".

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 and cube. We chose the 2D and 3D Franke test functions [Franke 1979] for the right hand side f and solve the discrete system

$$\mathbf{Lu} = \mathbf{Mb}, \tag{131}$$

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

$$\begin{aligned} f_{2\text{D}}(x,y) &= \frac{3}{4}\mathrm{e}^{-\frac{(9x-2)^2+(9y-2)^2}{4}} + \frac{3}{4}\mathrm{e}^{-\frac{(9x+1)^2}{49}-\frac{9y+1}{10}} \\ &+\frac{1}{2}\mathrm{e}^{-\frac{(9x-7)^2+(9y-3)^2}{4}} - \frac{1}{5}\mathrm{e}^{-(9x-4)^2-(9y-7)^2} \end{aligned} \tag{132}$$

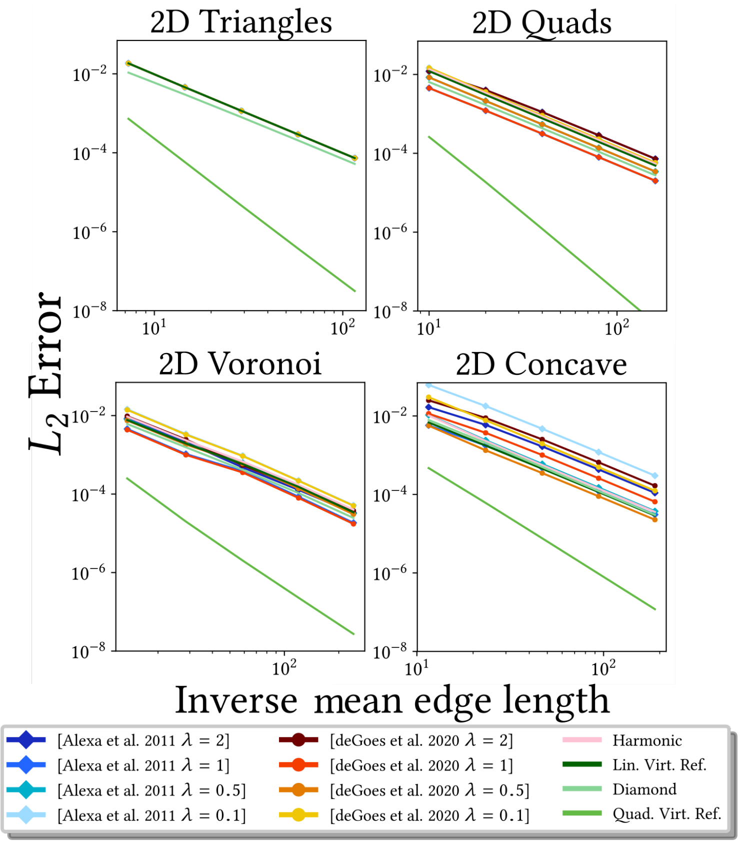


Fig. 10. L_2 error in log-log scale of the Poisson system solved for Franke's test function on planar grids with triangles (left), quads (center left), Voronoi cells (center right), and concave faces (right). Since all methods, with the exception of the Diamond Laplace, are equivalent to the cotangent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

and

$$\begin{aligned} f_{3D}(x,y,z) =& \frac{3}{4} e^{-\frac{(9x-2)^2+(9y-2)^2+(9z-2)^2}{4}} + \frac{3}{4} e^{-\frac{(9x+1)^2}{49} - \frac{9y+1}{10} - \frac{9z+1}{10}} \\ &+ \frac{1}{2} e^{-\frac{(9x-7)^2+(9y-3)^2+(9z-5)^2}{4}} \\ &- \frac{1}{5} e^{-(9x-4)^2-(9y-7)^2-(9z-5)^2}. \end{aligned} \tag{133}$$

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 or cubic convergence rate across all tessellations. In terms of accuracy, both operators introduced by Alexa and Wardetzky [2011] and de Goes et al. [2020] 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. [2020]. 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.

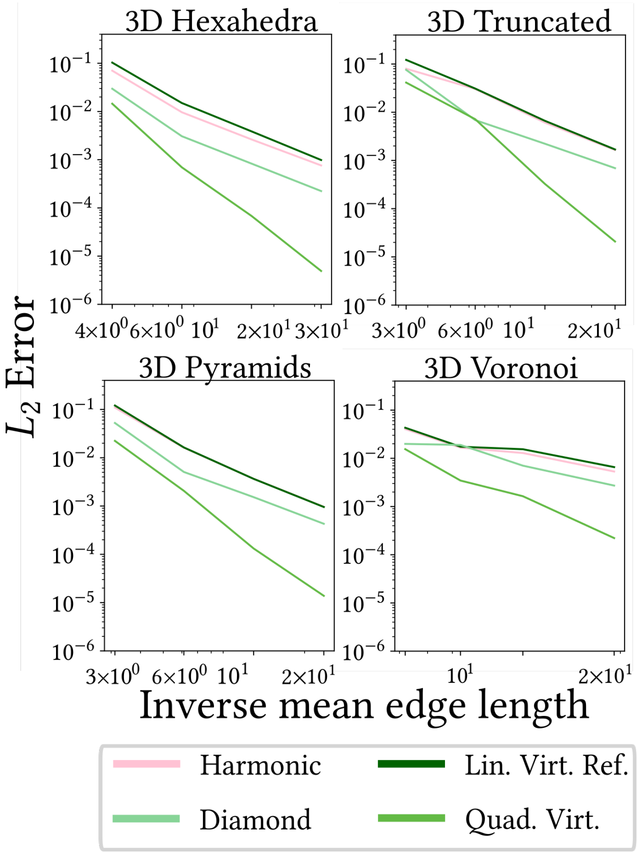


Fig. 11. L_2 error in log-log scale of the Poisson system solved for Franke's test function on unit cubes tessellated with hexahedra (left), pyramids (center left), truncated cells (center right), and Voronoi cells (right).

The error rates for the volumetric scenario are depicted in Figure 11. As before, the Diamond Laplacian is the most accurate operator with linear degrees of freedom, while both methods from Bunge et al. [2020] and Martin et al. [2008] yield qualitatively similar results. However, the harmonic shape functions are very expensive due to the solving process involved in their construction, especially for volume meshes, while the method from Bunge et al. [2020] is the fastest (see Section 8.5). Comparing the outcomes of all operators, the higher-order shape functions introduced by Bunge et al. [2022] yield the lowest error rates across all test meshes. This outcome is not unexpected, given that the major benefits of quadratic shape functions are higher accuracy and cubic convergence rate in exchange for larger and computationally more demanding systems.

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 $\mathbf{u} \in \mathbb{R}^{|\mathcal{V}|}$:

$$\mathbf{u} = \mathbf{M}^{-1}\mathbf{L}\mathbf{y}_l^m \tag{134}$$

$$\Leftrightarrow \quad \mathbf{M}\mathbf{u} = \mathbf{L}\mathbf{y}_l^m \tag{135}$$

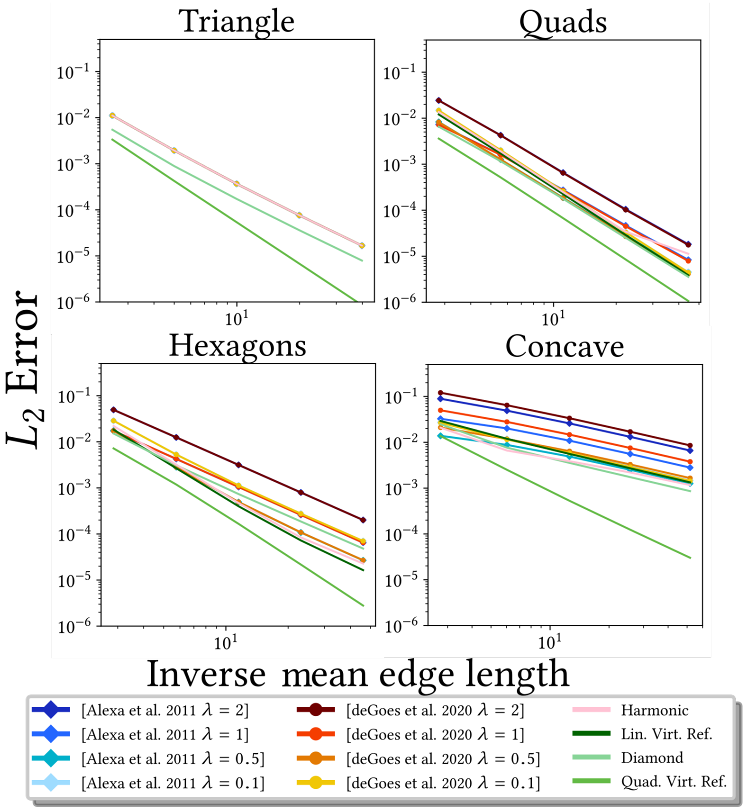


Fig. 12. L_2 error in log-log scale for the Poisson solve of the spherical harmonic function Y_2^3 with eigenvalue -12 on different tessellations of the unit sphere consisting of triangles (left), quads (center left), hexagons (center right) and concave faces (right). All methods, except the Diamond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

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

$$\left\| \mathbf{y}_l^m + \frac{1}{l(l+1)} \mathbf{u} \right\|_{\mathbf{M}}^2 \tag{136}$$

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 12 displays the deviation of the solution from the analytic function values of

$$Y_2^3(x, y, z) = \frac{1}{4}\sqrt{\frac{105}{\pi}}(x^2 - y^2)z \tag{137}$$

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 and the quadratic shape functions [Bunge et al. 2022]. 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

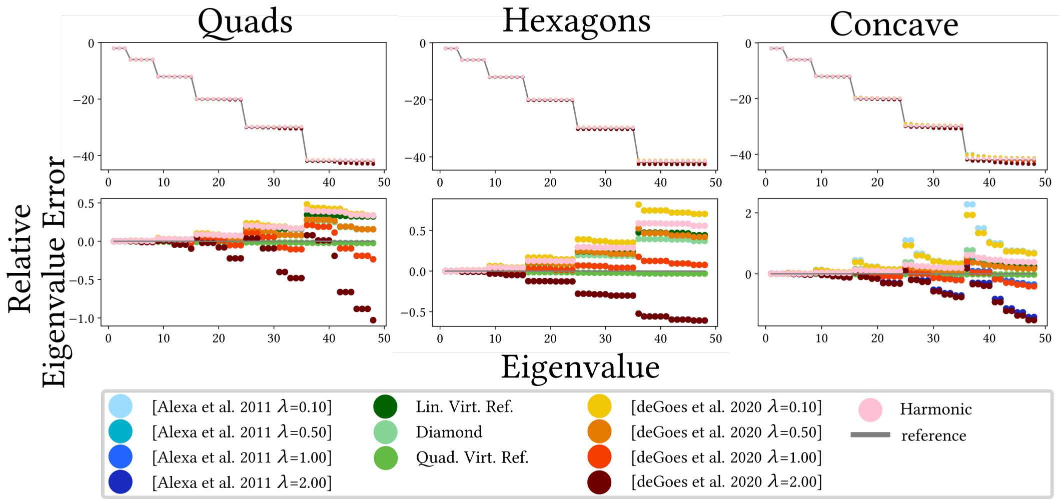


Fig. 13. The smallest 48 non-zero eigenvalues of the Laplacian on different unit spheres with quads (left), hexagons (center), and concave faces (right). The individual top plots shows the computed eigenvalues and the lower ones the relative deviation from the ground truth.

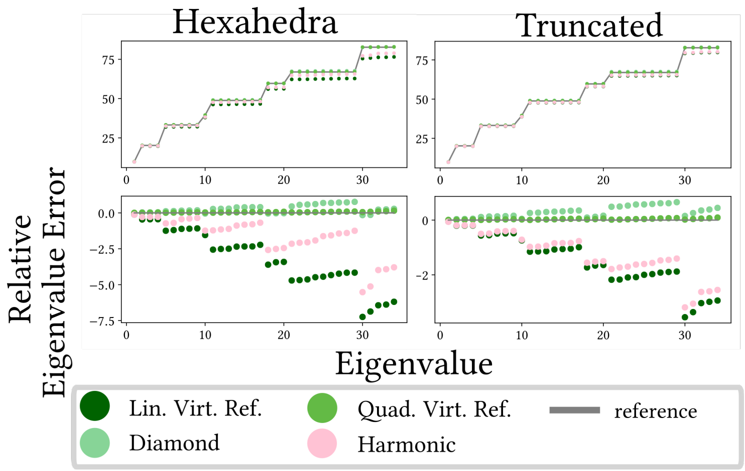


Fig. 14. The smallest 34 non-zero eigenvalues of the Laplacian on two unit balls consisting of hexhedra (left) and truncated cells (right). The individual top plots shows the computed eigenvalues and the lower ones the relative deviation from the ground truth.

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.

8.3 Eigenvalues and Eigenmodes

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

$$\mathbf{L}\mathbf{u} = \tilde{\lambda}\mathbf{M}\mathbf{u} \tag{138}$$

for the eigenvalues $\tilde{\lambda}$ of the discrete Laplacians and compare them to the analytic values. Figure 13 shows the first eigenvalues of each polygon Laplacian obtained on different spherical meshes. As before, the operator constructed with the higher-order shape functions by Bunge et al. [2022] yields the lowest error rates. A value of $\lambda = 1$ also leads to very accurate results for both Alexa and Wardetzky’s and de Goes et al.’s operators. 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 loose the stair-like pattern for the eigenvalues.

Similarly, given the volumetric unit 3-ball $\mathcal{B}^3$, the eigenfunctions u and eigenvalues λ of the Laplacian can be obtained with the help of the Helmholtz equation:

$$\Delta u = -\lambda u \text{ in } \mathcal{B}^3 \tag{139}$$

$$\text{s.t.} \quad u = 0 \text{ on } \partial\mathcal{B}^3. \tag{140}$$

The discrete solution can be expressed by the spherical Bessel functions, which allows us to solve the same generalized eigenvalue problem as in Equation (138) with the stiffness and mass matrix obtained on a polyhedral tessellation of $\mathcal{B}^3$. Figure 14 shows the results for the eigenvalues on the unit ball, where, after the higher-order discretization, the Diamond Laplacian has the most accurate results. Here, all methods display the desired constant eigenvalues for the respective frequencies, with only slight deviations.

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. [2013]. 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 $\mathbf{u} \in \mathbb{R}^{|\mathcal{V}|}$:

$$\begin{aligned} (\mathbf{I} - \epsilon\mathbb{L})\mathbf{u} &= \mathbf{e}_i \\ \Leftrightarrow \quad (\mathbf{M} - \epsilon\mathbf{L})\mathbf{u} &= \mathbf{M}\mathbf{e}_i. \end{aligned} \tag{141}$$

Then we compute the normalized gradients of the solution vector through

$$\mathbf{g}_j = \frac{(\mathbf{G}\mathbf{u})_j}{\left\|(\mathbf{G}\mathbf{u})_j\right\|}. \tag{142}$$

In the last step, we solve the Poisson equation

$$\mathbf{L}\mathbf{v} = \mathbf{D}\mathbf{g} \tag{143}$$

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. [Alexa and Wardetzky 2011; de Goes et al. 2020] obtain three-dimensional gradient vectors per polygon face, while Bunge et al. [2020] associate their gradient vectors with the virtual triangles, so one per halfedge of the original mesh. On the other hand, the Diamond gradient operator [Bunge et al. 2021] leads to intrinsic two-dimensional gradients that are associated with the virtual diamond

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

Mesh	$\lvert\mathcal{V}\rvert$	Harmonic			[Alexa and Wardetzky 2011]			[de Goes et al. 2020]			Lin. Virt. Ref.			Diamond		
		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
Hexahedra 3D	4913	465s	117k	3ms	—	—	—	—	—	—	190ms	117k	3ms	250ms	333k	6ms
Voronoi 3D	5183	482s	324k	5ms	—	—	—	—	—	—	140ms	324k	5ms	280ms	1497k	11ms

cells and therefore with the edges of the original mesh. Additionally, the normalization step in Equation (142) 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 [2011] needs an alternative approach. As pointed out by Crane et al. [2013], interpreting the coboundary operator **d** as gradient leads to discrete 1-forms associated with the halfedges, which cannot be directly normalized. However, since $\mathbf{M}_1$ (see Equation (33)) gives us an inner product matrix for 1-forms, they propose to use

$$\|\nabla u\|_f = \sqrt{\frac{\mathbf{u}_f^{\mathsf{T}} \mathbf{L}_f \mathbf{u}_f}{|f|}} \tag{144}$$

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

$$\mathbf{u}_f^{\mathsf{T}} \mathbf{L}_f \mathbf{u}_f = \int_f \|\nabla u\|^2 \, \mathrm{d}A = \|\nabla u\|^2 \, |f| \, . \tag{145}$$

The time step ϵ involved in the first step of the heat method (Equation (141)) is a debated subject. As pointed out by Crane et al. [2013], 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. [2013] and used in Bunge et al. [2020].
- The squared length of the longest face diagonal, as suggested by de Goes et al. [2020; 2016].

Figure 15 and 16 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. [2021; 2020] 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, independent 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.

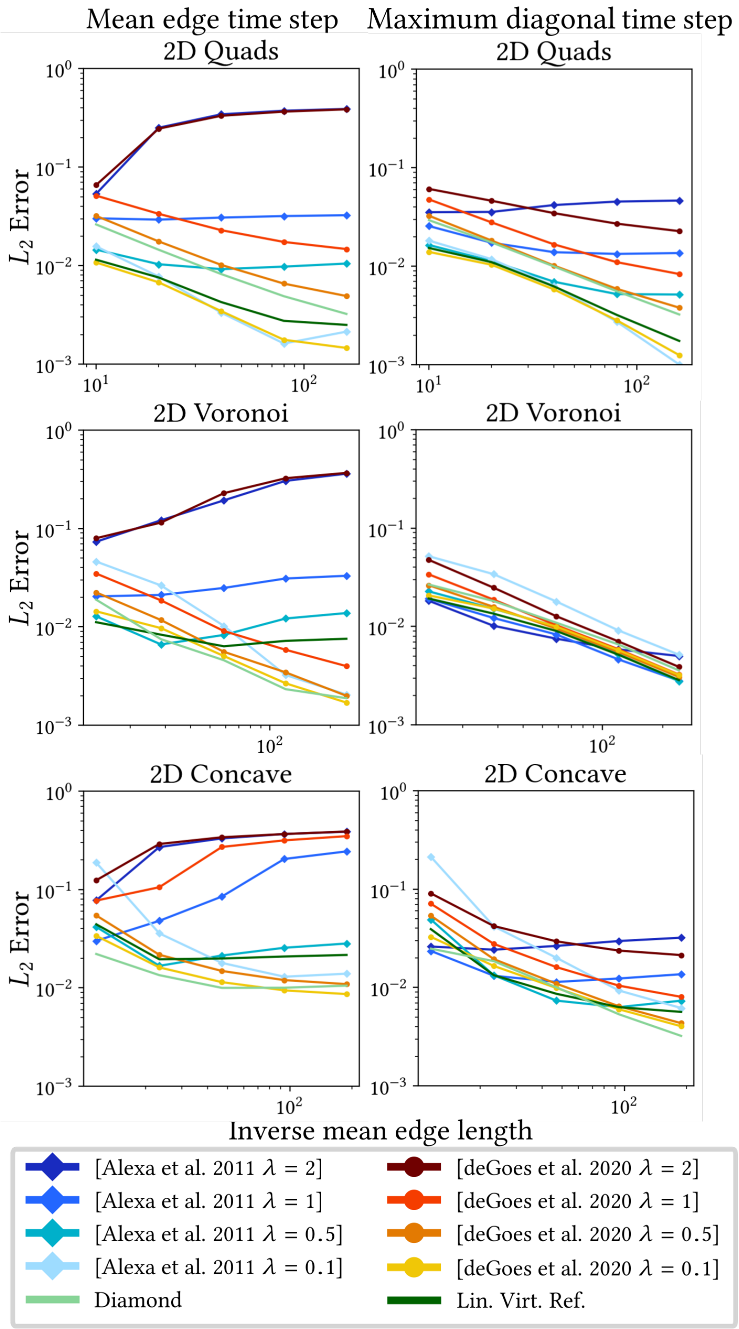


Fig. 15. L_2 error in log-log scale of the Geodesics in heat method on planar grids with quads (top), concave polygons (center) and Voronoi faces (bottom). For each refined planar mesh the selected vertex was the one with the least norm to the center of the plane.

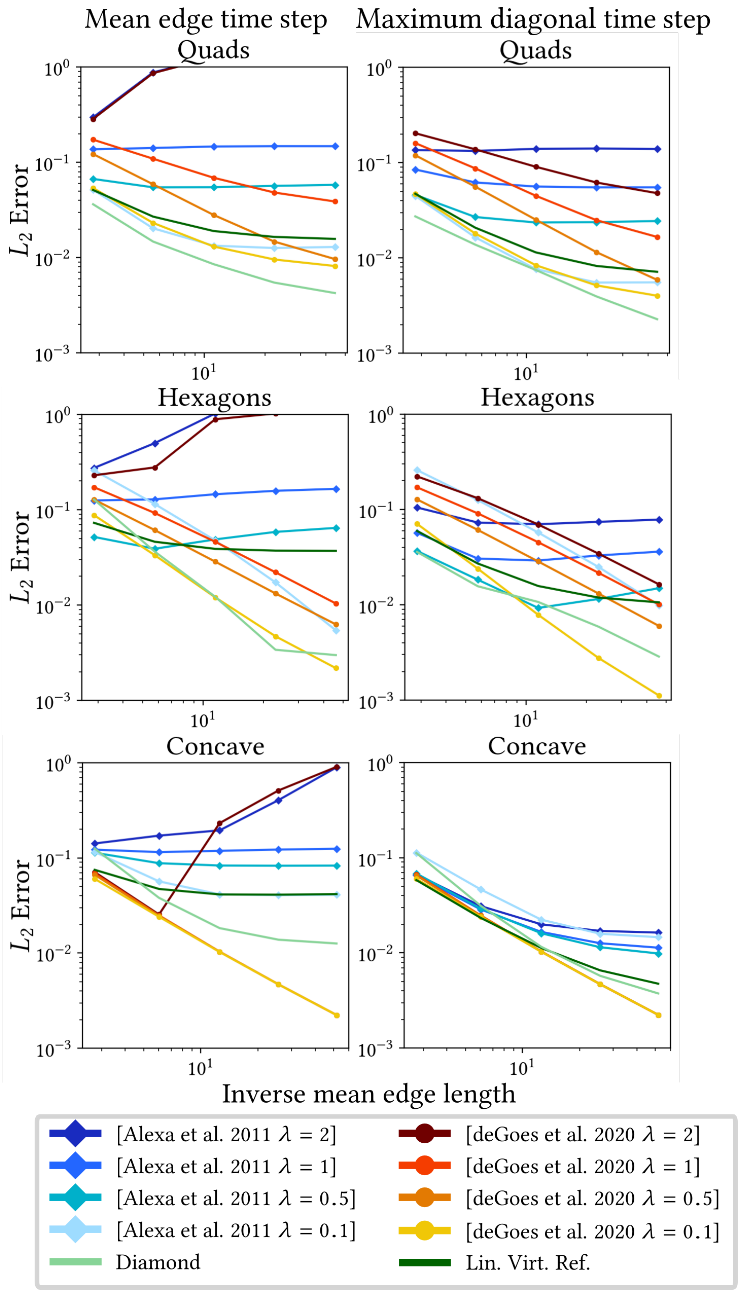


Fig. 16. L_2 error in log-log scale of the Geodesics in heat method on unit spheres with quad (top), hexagon (center) and concave faces (bottom).

Table 2. Statistics for the solution of a Poisson problem using different polygon and polyhedral Laplacians. The original values of this table were taken from [Bunge et al. 2022].

Mesh	$\lvert\mathcal{V}\rvert$	Lin. Virt. Ref.			Diamond			Quad. Virt. Ref.		
		solve	dof	nnz	solve	dof	nnz	solve	dof	nnz
Voronoi 2D	200k	0.73s	200k	2.6M	4.3s	200k	7.6M	7.7s	507k	12.1M
Voronoi 2D	800k	9.8s	800k	10.4M	21s	800k	29.6M	36.7s	2M	48M
Bunny 3D	80k	1.6s	80k	2M	6.2s	80k	5.9M	26s	316k	17.6M
Kong 3D	160k	4.44s	167k	4.3M	25.9s	167k	12M	97.6s	662k	38M

8.5 Timings and Sparsity

In this section, we compare different statistics involved in the solution process of a Poisson problem for both 2D and 3D 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 [Guennebaud et al. 2010] for the operators with linear degrees of freedom. Table 2 excludes the construction of the stiffness matrix (build), but lists its respective number of degrees of freedom (dof) which differs for higher-order shape functions. The solving times were obtained by using supernodal Cholesky decomposition and back substitution [Chen et al. 2008]. 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 Laplace and the operator discretized with the quadratic virtual refinement method, 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. [2008] exceeds the other operators by a tremendous amount. Especially for volume meshes, 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. [2020] and does not justify the large costs. Table 2 provides another comparison of statistics concerning the solution of a Poisson problem, now incorporating the quadratic shape functions introduced by Bunge et al. [2022]. As mentioned before, the time it takes to solve the system using a direct solver depends on the degrees of freedom (which manifest as number of rows and columns) and the number of non-zeros (nnz). The definition of quadratic basis functions introduces additional nodes, which leads to larger and denser matrices. The previously observed faster convergence and higher accuracy of Bunge et al.'s method therefore come at the cost of a numerically more demanding solve process compared to methods that only use vertex nodes. Since the timings in Table 2 were obtained with a Cholesky solver, discrete Laplacians with the same non-zero structure lead to identical solving times and can be treated equally in this setting. The Diamond Laplace of Bunge et al. [2021] introduces coefficients that establish connections between vertex nodes of neighboring elements, resulting in denser matrices and therefore justifying a separate column.

As expected, the solving times of the quadratic method are higher for all examples due to the larger number of degrees of freedom and denser matrices. However, using a multigrid construction presented in the original paper [Bunge et al. 2022] can lower the computational costs and is a common approach when using higher-order shape functions.

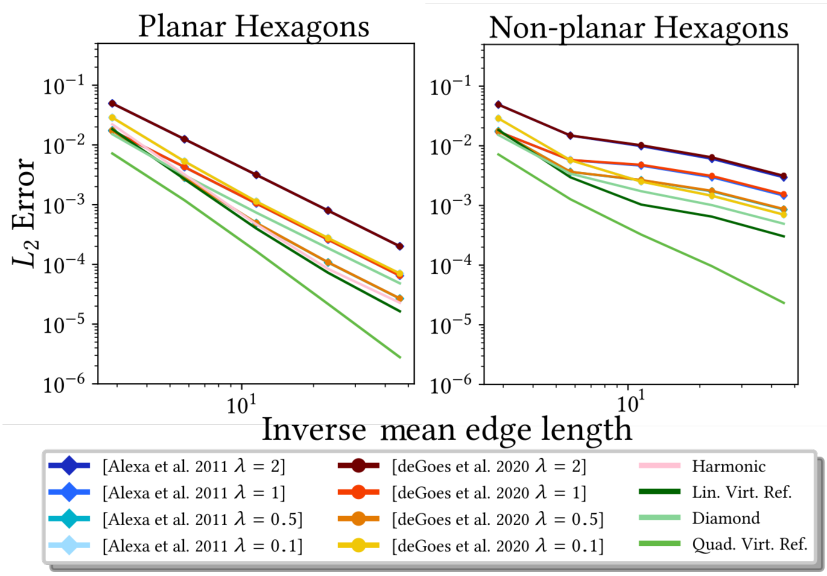


Fig. 17. L_2 error in log-log scale for the Poisson solve of the spherical harmonic function Y_2^3 with eigenvalue −12 on planar and non-planar hexagons. The non-planar elements affect both the convergence rate and accuracy of all methods. The harmonic shape functions are not included in this experiment, since their definition only holds for planar polygons.

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 both surface and volume 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 17). 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. [2021; 2020; 2022] being the least affected. In contrast to the planar case, choosing a lower hyperparameter λ for the operators of Alexa and Wardetzky [2011] and de Goes et al. [2020] 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 18. 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. [2008] were not included in these evaluations, since their definition only holds for planar polygons.

Another aspect in which the underlying tessellation may directly affect the quality of the operators

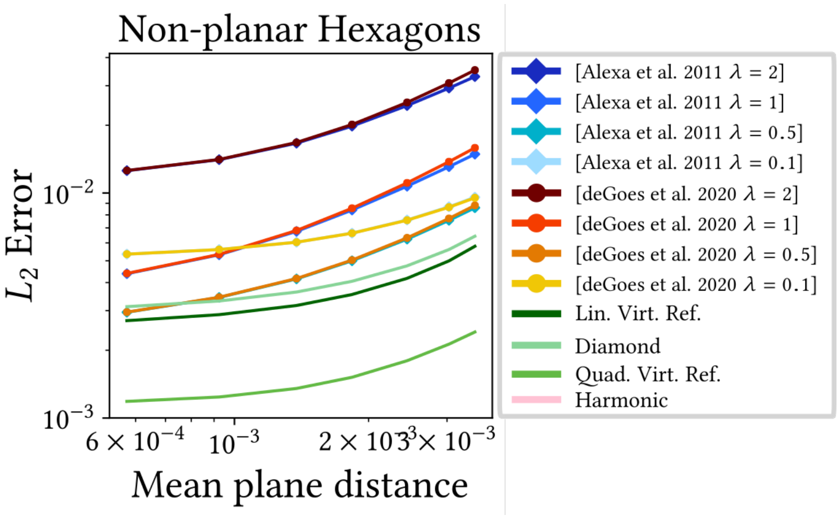


Fig. 18. L_2 error in log-log scale for the Poisson solve of the spherical harmonic function Y_2^3 on progressively non-planar hexagons. We added different magnitudes of tangential noise to the initial faces and projected the vertices back onto the unit sphere. In order to measure the “non-planarity” of the polygons, the x-axis depicts the mean distance of the face vertices to a fitted plane.

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 [Herholz et al. 2015]. A possible way to quantify the stability of the operator with respect to the given tessellation is to analyze its condition number κ. It is defined as the ratio

$$\kappa = \frac{\lambda_{max}}{\lambda_{min}} \tag{146}$$

of the Laplacians largest and smallest non-zero eigenvalues λ_{max} 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 [Krishnan et al. 2013]. 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 19. 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. [2021; 2020; 2022] begin to rise significantly due to the diminishing triangle quality of the virtual refinements. This trend is particularly pronounced in the context of the quadratic virtual refinement method, where the additional degrees of freedom and refined quadratic Lagrange elements result in considerably higher condition numbers. The numerical stability of the operators presented by Martin et al. [2008] and de Goes et al. [2020] also decreases, but with a lower magnitude. In contrast, the Laplacian presented by Alexa and Wardetzky [2011] 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.

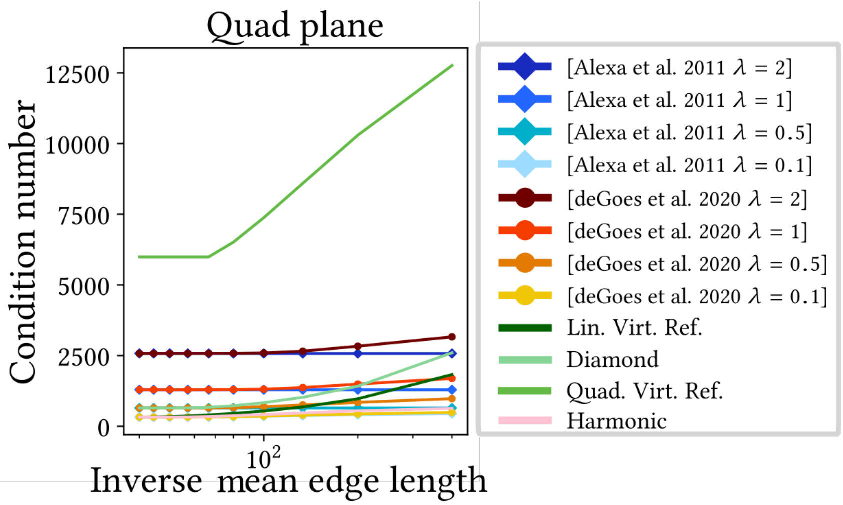


Fig. 19. Condition numbers (Equation (146)) of the different Laplace operators on an initially uniform quad plane. One edge within the grid is progressively shortened, resulting in four faces to become more and more distorted with each iteration. The diminishing quality of the polygons lead to higher condition numbers for most of the operators.

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.

In the linear setting, 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 works both on surface and volume meshes and leads to accurate results without any adjusting, 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. [2020]. It works on both surface and volume meshes and, given that many applications already work with the cotangent Laplacian, can be easily integrated since the only missing piece is the prolongation matrix.

Should the reader's primary goals revolve around achieving high accuracy and faster convergence, with limited regard for the computational complexities and potential numerical stability of the underlying system to be solved, then forsaking the linear setting entirely in favor of employing higher-order shape functions as defined by Bunge et al. [2022] would be the best choice.

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 polyhedra, and can therefore be seamlessly mixed with these standard elements.

10 CONCLUSION AND OUTLOOK

Throughout this course, we attempted to give a comprehensive description of the recent progress achieved within the graphics community to construct general polygonal and polyhedral Laplace operators. By highlighting the numerical schemes used for the respective Laplacians, we provide context for the individual discretization strategies and motivate the problems involved in their generalization process. Furthermore, we analyzed the properties of most Laplacians and investigated similarities and parallels between the presented methods. As a second aspect of this course, we introduced a variety of discrete gradient and divergence operators and explain 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 reoccurring debates within the original papers. The source code for these tests and the construction of the individual operators are publicly available.

We hope that this course helps to motivate the usage of more general tessellations within the graphics community, given that the necessary tools for many applications are already there. Furthermore, besides the Laplacian, 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. [2020] and [Lipnikov et al. 2014] in their survey on the MFD method.

Acknowledgements

We are grateful to Philipp Herholz for his 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.

REFERENCES

Marc Alexa, Philipp Herholz, Maximilian Kohlbrenner, and Olga Sorkine-Hornung. 2020. Properties of Laplace Operators for Tetrahedral Meshes. *Computer Graphics Forum* 39, 5 (2020).

Marc Alexa and Max Wardetzky. 2011. Discrete Laplacians on General Polygonal Meshes. *ACM Transactions on Graphics* 30, 4 (2011), 102:1–102:10.

Boris Andreianov, Mostafa Bendahmane, and Florence Hubert. 2013. On 3D DDFV discretization of gradient and divergence operators. II. Discrete functional analysis tools and applications to degenerate parabolic problems. *Computational Methods in Applied Mathematics* 13, 4 (2013), pp. 369–410. https://doi.org/10.1515/cmam-2013-0011

Boris Andreianov, Mostafa Bendahmane, Florence Hubert, and Stella Krell. 2012. 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. https://doi.org/10.1093/imanum/drr046

Douglas N. Arnold, Richard S. Falk, and Ragnar Winther. 2006. Finite element exterior calculus, homological techniques, and applications. *Acta Numerica* 15 (2006), 1–155.

Prusty Aurojyoti, Piska Raghu, Amirtham Rajagopal, and Jn Reddy. 2019. An n-sided polygonal finite element for nonlocal nonlinear analysis of plates and laminates. *Internat. J. Numer. Methods Engrg.* 120 (2019), 1071–1107.

Lourenço Beirão da Veiga, Franco Brezzi, and Luisa Donatella Marini. 2013. Virtual Elements for Linear Elasticity Problems. *SIAM J. Numer. Anal.* 51, 2 (2013), 794–812.

Lourenço Beirão da Veiga, Franco Dassi, and Alessandro Russo. 2017. High-order Virtual Element Method on polyhedral meshes. *Computers and Mathematics with Applications* 74 (2017), 1110–1122.

J.E. Bishop. 2009. Simulating the Pervasive Fracture of materials and Structures Using Randomly Close Packed Voronoi Tessellations. *Computational Mechanics* 44, 4 (2009), 455–471.

J.E. Bishop. 2014. A displacement-based finite element formulation for general polyhedra using harmonic shape functions. *Internat. J. Numer. Methods Engrg.* 97 (2014), 1–31.

Mario Botsch, Robert Sumner, Mark Pauly, and Markus Gross. 2006. Deformation Transfer for Detail-Preserving Surface Editing. In *Proc. of Vision, Modeling and Visualization.* 357–364.

Franco Brezzi, Konstantin Lipnikov, Mikhail Shashkov, and Valeria Simoncini. 2007. A new Discretization Methodology for Diffusion Problems on Generalized Polyhedral Meshes. *Computer Methods in Applied Mechanics and Engineering* 196, 37

(2007), 3682–3692. https://doi.org/10.1016/j.cma.2006.10.028

Franco Brezzi, Konstantin Lipnikov, and Valeria Simoncini. 2005. A Family of Mimetic Finite Difference Methods on Polygonal and Polyhedral Meshes. *Mathematical Models and Methods in Applied Sciences* 15, 10 (2005), 1533–1551.

Astrid Bunge and Mario Botsch. 2023. A Survey on Discrete Laplacians for General Polygonal Meshes. *Computer Graphics Forum* 42, 2 (2023), 521–544. https://doi.org/10.1111/cgf.14777

Astrid Bunge, Mario Botsch, and Marc Alexa. 2021. The Diamond Laplace for Polygonal and Polyhedral Meshes. *Computer Graphics Forum* 40, 5 (2021), 217–230.

Astrid Bunge, Philipp Herholz, Misha Kazhdan, and Mario Botsch. 2020. Polygon Laplacian Made Simple. *Computer Graphics Forum* 39, 2 (2020), 303–313.

Astrid Bunge, Philipp Herholz, Olga Sorkine-Hornung, Mario Botsch, and Michael Kazhdan. 2022. Variational Quadratic Shape Functions for Polygons and Polyhedra. *ACM Trans. Graph.* 41, 4, Article 54 (2022), 14 pages. https://doi.org/10.1145/3528223.3530137

Renjie Chen and Craig Gotsman. 2016. On pseudo-harmonic barycentric coordinates. *Computer Aided Geometric Design* 44 (2016), 15–35. https://doi.org/10.1016/j.cagd.2016.04.005

Yanqing Chen, Timothy A. Davis, William W. Hager, and Sivasankaran Rajamanickam. 2008. Algorithm 887: CHOLMOD, Supernodal Sparse Cholesky Factorization and Update/Downdate. *ACM Trans. Math. Softw.* 35, 3 (2008), 1–14.

Ming Chuang, Linjie Luo, Benedict J. Brown, Szymon Rusinkiewicz, and Michael Kazhdan. 2009. Estimating the Laplace-Beltrami Operator by Restricting 3D Functions. *Computer Graphics Forum* 28, 5 (2009), 1475–1484.

Yves Coudière and Florence Hubert. 2011. A 3D Discrete Duality Finite Volume Method for Nonlinear Elliptic Equations. *SIAM J. Sci. Comput.* 33, 4 (2011), 1739–1764.

Keenan Crane. 2019. The *n*-dimensional cotangent formula. https://www.cs.cmu.edu/~kmcrane/Projects/Other/nDCotanFormula.pdf.

Keenan Crane, Clarisse Weischedel, and Max Wardetzky. 2013. Geodesics in Heat: A New Approach to Computing Distance Based on Heat Flow. *ACM Transactions on Graphics* 32, 5 (2013), 152:1–152:11.

Fernando de Goes, Andrew Butts, and Mathieu Desbrun. 2020. Discrete Differential Operators on Polygonal Meshes. *ACM Transactions on Graphics* 39, 4 (2020), 110:1–110:14.

Fernando de Goes, Mathieu Desbrun, Mark Meyer, and Tony DeRose. 2016. Subdivision Exterior Calculus for Geometry Processing. *ACM Transactions on Graphics* 35, 4 (2016), 133:1–133:11.

Mathieu Desbrun, Anil N. Hirani, Melvin Leok, and Jerrold E. Marsden. 2005. Discrete Exterior Calculus. *arXiv: Differential Geometry* (2005). https://doi.org/10.48550/ARXIV.MATH/0508341

Mathieu Desbrun, Mark Meyer, Peter Schröder, and Alan H. Barr. 1999. Implicit Fairing of Irregular Meshes Using Diffusion and Curvature Flow. In *Proceedings of ACM SIGGRAPH*. 317–324.

K. Domelevo and P. Omnés. 2005. A finite volume method for the Laplace equation on almost arbitrary two-dimensional grids. *Math. Model. Numer. Anal. (M2AN)* 39, 6 (2005), 1203–1249.

G.M. Dusinberre. 1961. *Heat-transfer Calculations by Finite Differences.* International Textbook Company.

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

Gerhard Dziuk. 1988. Finite Elements for the Beltrami operator on arbitrary surfaces. In *Partial Differential Equations and Calculus of Variations*, Stefan Hildebrandt and Rolf Leis (Eds.). Springer Berlin Heidelberg, 142–155. https://doi.org/10.1007/BFb0082865

Graeme Fairweather and Andreas Karageorghis. 1998. The method of fundamental solutions for elliptic boundary value problems. *Advances in Computational Mathematics* 9, 1 (1998), 69–95.

Michael S. Floater. 2003. Mean value coordinates. *Computer Aided Geometric Design* 20, 1 (2003), 19–27.

Michael S. Floater. 2015. Generalized Barycentric Coordinates and Applications. *Acta Numerica* 24 (2015), 161–214. https://doi.org/10.1017/S0962492914000129

Richard Franke. 1979. *A critical comparison of some methods for interpolation of scattered data.* Technical Report. Naval Postgraduate School.

Gene H. Golub and Charles F. Van Loan. 1996. *Matrix Computations.* Johns Hopkins University Press.

Gaël Guennebaud, Benoît Jacob, et al. 2010. Eigen v3. http://eigen.tuxfamily.org.

Philipp Herholz, Jan Eric Kyprianidis, and Marc Alexa. 2015. Perfect Laplacians for Polygon Meshes. *Computer Graphics Forum* 34, 5 (2015), 211–218.

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

F. Hermeline. 2009. A Finite Volume Method for Approximating 3D Diffusion Operators on General Meshes. *J. Comput. Phys.* 228, 16 (2009), 5763–5786.

Klaus Hildebrandt, Konrad Polthier, and Max Wardetzky. 2006. On the Convergence of Metric and Geometric Properties of Polyhedral Surfaces. *Geometriae Dedicata* 123 (2006), 89–112. https://doi.org/10.1007/s10711-006-9109-5

W. Höhn and H. D. Mittelmann. 1981. Some remarks on the discrete maximum-principle for finite elements of higher order. *Computing* 27, 2 (1 June 1981), 145–154. https://doi.org/10.1007/BF02243548

K. Hormann and N. Sukumar. 2008. Maximum Entropy Coordinates for Arbitrary Polytopes. *Computer Graphics Forum* 27, 5 (2008), 1513–1520.

T.J.R. Hughes. 2012. *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis.* Dover Publications.

Pushkar Joshi, Mark Meyer, Tony DeRose, Brian Green, and Tom Sanocki. 2007. Harmonic Coordinates for Character Articulation. *ACM Trans. Graph.* 26, 3 (2007), 71–81. https://doi.org/10.1145/1276377.1276466

Tao Ju, Scott Schaefer, and Joe Warren. 2005. Mean Value Coordinates for Closed Triangular Meshes. *ACM Transactions on Graphics* 24, 3 (2005), 561–566.

Dilip Krishnan, Raanan Fattal, and Richard Szeliski. 2013. Efficient Preconditioning of Laplacian Matrices for Computer Graphics. *ACM Transactions on Graphics* 32, 4, Article 142 (2013), 15 pages. https://doi.org/10.1145/2461912.2461992

J.M. Lee. 1997. *Riemannian Manifolds: An Introduction to Curvature.* Springer New York.

Konstantin Lipnikov, Gianmarco Manzini, and Mikhail Shashkov. 2014. Mimetic finite difference method. *J. Comput. Phys.* 257 (2014), 1163–1227.

Richard MacNeal. 1949. *The Solution of Partial Differential Equations by Means of Electrical Networks.* Ph. D. Dissertation. California Institute of Technology.

G. Manzini, A. Russo, and N. Sukumar. 2014. New perspectives on polygonal and polyhedral finite element methods. *Mathematical Models and Methods in Applied Sciences* 24 (2014), 1665–1699.

Sebastian Martin, Peter Kaufmann, Mario Botsch, Martin Wicke, and Markus Gross. 2008. Polyhedral Finite Elements Using Harmonic Basis Functions. *Computer Graphics Forum* 27, 5 (2008), 1521–1529.

M. Meyer, M. Desbrun, P. Schröder, and A. H. Barr. 2003. Discrete Differential-Geometry Operators for Triangulated 2-Manifolds. In *Visualization and Mathematics III*, Hans-Christian Hege and Konrad Polthier (Eds.). Springer-Verlag, 35–57.

Ean Tat Ooi, Chongmin Song, Francis Tin-Loi, and Zhenjun Yang. 2012. Polygon scaled boundary finite elements for crack propagation modelling. *Internat. J. Numer. Methods Engrg.* 91, 3 (2012), 319–342. https://doi.org/10.1002/nme.4284

Ulrich Pinkall and Konrad Polthier. 1993. Computing discrete minimal surfaces and their conjugates. *Experim. Math.* 2 (1993), 15–36.

El Houssaine Quenjel, Mazen Saad, Mustapha Ghilani, and Marianne Bessemoulin-Chatard. 2020. Convergence of a positive nonlinear DDFV scheme for degenerate parabolic equations. *Calcolo* 50, 19 (2020). https://doi.org/10.1007/s10092-020-00367-5

Bastian E. Rapp. 2017. Chapter 31 - Finite Volume Method. In *Microfluidics: Modelling, Mechanics and Mathematics.* Elsevier, Oxford, 633–654. https://doi.org/10.1016/B978-1-4557-3141-1.50031-9

Steven Rosenberg. 1997. *The Laplacian on a Riemannian Manifold.* Cambridge University Press, 1–51. https://doi.org/10.1017/CBO9780511623783.002

Teseo Schneider, Yixin Hu, Jérémie Dumas, Xifeng Gao, Daniele Panozzo, and Denis Zorin. 2018. Decoupling Simulation Accuracy from Mesh Quality. *ACM Transactions on Graphics* 37, 6 (2018), 280:1–280:14.

Teseo Schneider, Yixin Hu, Xifeng Gao, Jeremie Dumas, Denis Zorin, and Daniele Panozzo. 2022. A Large Scale Comparison of Tetrahedral and Hexahedral Elements for Solving Elliptic PDEs with the Finite Element Method. *ACM Transactions on Graphics* 41, 3 (2022), 23:1–23:14.

Nicholas Sharp. 2021. *Intrinsic Triangulations in Geometry Processing.* PhD thesis. Carnegie Mellon University. https://doi.org/10.1184/R1/17209181.v1

Alireza Tabarraei and N. Sukumar. 2006. Application of Polygonal Finite Elements in Linear Elasticity. *International Journal of Computational Methods* 03 (2006), 503–520.

Alireza Tabarraei and N. Sukumar. 2008. Extended Finite Element Method on Polygonal and Quadtree Meshes. *Computer Methods in Applied Mechanics and Engineering* 197 (2008), 425–438. https://doi.org/10.1016/j.cma.2007.08.013

Max Wardetzky. 2008. *Convergence of the Cotangent Formula: An Overview.* Birkhäuser Basel, Basel, 275–286. https://doi.org/10.1007/978-3-7643-8621-4_15

Max Wardetzky, Saurabh Mathur, Felix Kälberer, and Eitan Grinspun. 2007. Discrete Laplace operators: No free lunch. In *Proceedings of Eurographics Symposium on Geometry Processing.* 33–37.

Hassler Whitney. 1957. *Geometric Integration Theory.* Princeton University Press.

Martin Wicke, Mario Botsch, and Markus Gross. 2007. A Finite Element Method on Convex Polyhedra. *Computer Graphics Forum* 26, 3 (2007), 355–364.

O.C. Zienkiewicz, R.L. Taylor, and J.Z. Zhu. 2013. Chapter 8 - The Patch Test, Reduced Integration, and Nonconforming Elements. In *The Finite Element Method: its Basis and Fundamentals* (7th edition ed.), O.C. Zienkiewicz, R.L. Taylor, and J.Z. Zhu (Eds.). Butterworth-Heinemann, Oxford, 257–284. https://doi.org/10.1016/B978-1-85617-633-0.00008-3