EUROGRAPHICS 2024 / A. Bermano and E. Kalogerakis
(Guest Editors)

COMPUTER GRAPHICS *forum*
Volume 43 (2024), Number 2

Polygon Laplacian Made Robust

A. Bunge[1†] D. R. Bukenberger[1,2†] S. D. Wagner[1] M. Alexa[3] M. Botsch[1]

[1]TU Dortmund University, Germany [2]TU Munich, Germany [3]TU Berlin, Germany

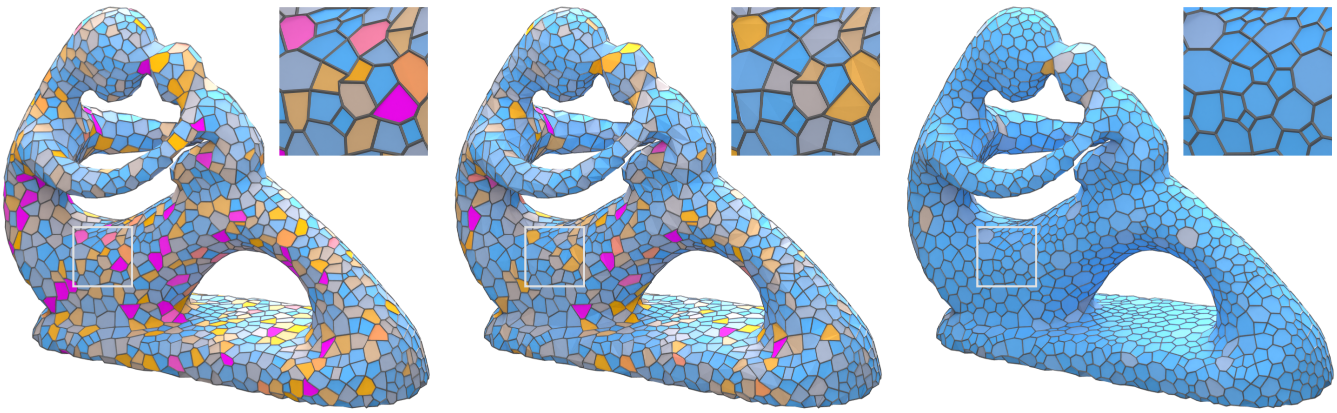


Figure 1: *We visualize the per-element stiffness matrix condition numbers of the discrete polygon Laplacian, with lower values* (blue) *indicating better conditioned local matrices than higher values* (magenta)*. The polygon Laplacian of Bunge et al. [BHKB20] suffers from low-quality elements* (left)*, which deteriorate the condition of the global Laplace matrix. Our proposed polygon Laplacian* (center) *and custom-tailored polygon mesh optimization* (right) *considerably improve numerical conditioning and robustness.*

Abstract

Discrete Laplacians are the basis for various tasks in geometry processing. While the most desirable properties of the discretization invariably lead to the so-called cotangent Laplacian for triangle *meshes, applying the same principles to* polygon *Laplacians leaves degrees of freedom in their construction. From linear finite elements it is well-known how the shape of triangles affects both the error and the operator's condition. We notice that shape quality can be encapsulated as the trace of the Laplacian and suggest that trace minimization is a helpful tool to improve numerical behavior. We apply this observation to the polygon Laplacian constructed from a virtual triangulation [BHKB20] to derive optimal parameters per polygon. Moreover, we devise a smoothing approach for the vertices of a polygon mesh to minimize the trace. We analyze the properties of the optimized discrete operators and show their superiority over generic parameter selection in theory and through various experiments.*

CCS Concepts

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

1. Introduction

The discrete Laplace-Beltrami operator is one of the most important tools in computer graphics and geometry processing, as it is employed for smoothing, fairing, parameterization, deformation, geodesic distances, and many more [BKP*10, SCV14]. Due to its importance, its properties have been thoroughly investigated and are well understood [WMKG07]. For triangle meshes the cotangent Laplacian [PP93, MDSB03, DMSB99, Dzi88] is used almost exclusively, since it satisfies crucial properties such as linear precision, symmetry, and negative semi-definiteness [WMKG07, War08].

While discrete operators for *triangle* meshes are well understood, this is not the case for general *polygon* meshes, despite the growing demand for the latter in many modeling and engineering applications. As noted by Peng et al. [PPW18], meshing algorithms commonly produce tessellations that incorporate non-triangular el-

[†] The first two authors contributed equally to this work.

ements, such as quad-dominant meshes, mixed tri/quad meshes, or Voronoi-based polygon meshes. Peng et al. further emphasize that these patterns have various applications in architecture, industrial design, and art. Polygon meshes (and differential operators on them) are used in the computer animation industry, as they effectively capture geometric features and facilitate both artistic design and fabrication processes [dGBD20]. Having a Laplacian that seamlessly works on all these meshes is a valuable tool, and several derivations for such operators have been suggested [dGBD20, BHKB20, BBA21], showing similar but slightly different performance in various applications [BB23].

Our central observation is that the approach of Bunge et al. [BHKB20] – using a *virtual triangulation of the polygons* – allows applying the knowledge from the linear Finite Element Method (FEM) about *triangle* shapes to construct an optimized *polygon* Laplace operator. From linear FEM it is known that (and how) the shape of the elements (i.e., triangles) affects accuracy and numerical stability of the "cotan" Laplacian [She02]: In general, the ratio of squared edge lengths to triangle areas should be small. We aim at creating virtual triangulations that minimize this ratio. Notably, this measure has appeared before as the *harmonic index* of a triangle and has been linked to the Delaunay triangulation [Mus97] and the discrete Laplace operator [BS07]. In particular, the sum of the harmonic indices across all triangles in the mesh corresponds to the *trace* of the cotangent stiffness matrix [Ale19]. Using the minimization of the trace as the guiding principle, we derive optimal positions for the virtual vertices within the polygons and optimal weights to express these points as affine combinations of the polygon vertices. Additionally, we derive a smoothing scheme that optimizes the vertex locations (for applications where this is permissible) based on the same principles.

We analyze how minimizing the trace affects the spectrum of the discrete Laplace operator and compare the optimized Laplacian in a variety of applications to the ones from the literature. Our experiments demonstrate that for low-quality polygon meshes our optimized operator consistently and significantly improves robustness and accuracy – with only negligible computational cost for constructing the discrete operator and no computational overhead for solving the involved linear systems. We integrated our robust Laplacian operator and polygon mesh smoothing into the source code of Bunge and Botsch [BB23], which can be found at https://github.com/mbotsch/polyLaplace.

2. Related Work

Polygon Laplacians Discrete differential operators typically emerge from applying methods such as the Finite Element Method (FEM), the Finite Volume Method (FVM), or Discrete Exterior Calculus (DEC) to continuous differential or integral equations. For general polygonal meshes, these different methods lead to distinct but related discretizations of the Laplace operator, on which the state-of-the-art report by Bunge and Botsch [BB23] provides an extensive background. We highlight the most prominent operators in this section, we refer to Bunge and Botsch for more details.

Alexa and Wardetzky [AW11] generalize to non-planar polygon meshes by projecting each polygon onto the plane maximizing its vector area. They introduce an inner product stabilization term influenced by mimetic finite differences [BLS05]. De Goes et al. [dGBD20] derive a range of DEC operators for polygon meshes. They expand on the concepts of Alexa and Wardetzky [AW11] and generalize the same planar inner product matrix [BLS05] using principles from the Virtual Element Method (VEM) [BdVBM13]. They define the gradient operator based on the co-gradient, from which the other operators are then derived. These DEC-based Laplacians require a stabilization term that regulates the dimension of the kernel, which may contain more than constants for two-manifolds embedded in 3D. The regularization is scaled by a hyperparameter λ, and while the methods work well depending on the chosen λ, this degree of freedom leaves little room to link the operator's properties to the underlying tessellation.

The finite element method discretizes a continuous surface into simple elements to approximate functions with the help of local basis functions. A possible way to obtain these shape functions for general polygons is to use generalized barycentric coordinates [HS17, Wac75, Flo03, JSW05, JMD*07, Suk04, HS08], which are typically defined for planar elements only. Another FEM approach that works for non-planar surface meshes is the linear virtual refinement method by Bunge et al. [BHKB20]. They divide each polygon into a virtual triangle fan by introducing a virtual vertex. Standard techniques like the cotangent Laplacian can then be used on the virtual triangulation. The virtual refinement process is concealed through prolongation and restriction matrices, introducing two degrees of freedom: the placement of the virtual vertex and the weights within the prolongation matrices.

The Diamond Laplacian [BBA21] is defined on polygonal and polyhedral meshes. It utilizes the Discrete Duality Finite Volume Method [Her09, CH11] to define a generalized gradient and divergence operator, which can be combined to obtain the Laplacian. This operator uses the same virtually refined mesh described by Bunge et al. [BHKB20] and, therefore, the same degrees of freedom. However, since the Laplacian is based on the FVM, which relies on two tessellations in the form of a dual and primal mesh, the analysis to uncover a geometric link to the original polygons would be increasingly convoluted in contrast to the original method by Bunge et al. [BHKB20] on which we will focus in this paper.

Numerical Accuracy and Robustness On Polygons The shape of a simplex can affect the quality of linear FEM discretizations for triangles and tetrahedra. Shewchuk [She02] analyzed various aspects of this phenomenon, which were later expanded on in the exhaustive survey by Sorgente et al. [SBMS23]. However, Sorgente et al. also noted that many concepts used to evaluate the geometric quality of lower-degree polytopes cannot be easily transferred to generic polygons. To address this issue, several works developed shape quality metrics specifically for surface and volume polytopes, such as Lipnikov [Lip13], Mu et al. [MWW15], and Gillette and Rand [GR17]. In a recent study, Attene et al. [ABB*21] investigated the correlation between the performance of VEM [BdVBM13] and the underlying polygonal tessellation. Our paper expands on this type of analysis and establishes a geometric link between polygons and the linear virtual refinement method [BHKB20].

Polygon Mesh Smoothing The ability to enhance the quality of a mesh by improving, or *smoothing* the shapes of the individual faces, is a valuable tool in geometry processing. The effectiveness of these algorithms has been extensively studied, particularly in 2D. This extends to meshes in 3D when the placement of (interior) vertices is less critical, e.g., tessellations of planes in CAD-like objects as long as sharp feature edges and corners are retained. Mesh smoothing impacts numerical stability and discretization error, a crucial consideration in computational simulations. Numerous approaches have been introduced for triangle meshes, with some extending to quads or polygonal meshes.

Zhou and Shimada approach [ZS00] focuses on optimizing the inner angles within mixed 2D meshes. It has been shown to be effective for triangles, as adjusting the angles generally improves the ratio of edge lengths to area (Section 5). The extension to quads is based on introducing virtual diagonals, and it is unclear how to extend this concept to general polygons. The works of Garimella et al. [GSK02, GSK04, GS04] aim for optimizing triangle, quad, mixed and polygon manifold surface meshes. They formulate an objective function to minimize the condition numbers of local Jacobian matrices, which then indirectly improve the global mesh quality. However, by construction, this approach can only incorporate direct vertex neighbors and, therefore, does not generalize well for larger polygons, limiting the effect on the condition of the global stiffness matrix. Further, Knupp et al. [KMS02] proposed mesh optimization using Reference Jacobian Matrices, which helps preserving mesh details but is not suitable for polygon meshes due to ambiguous local Jacobian definitions.

Vartziotis et al. [VAGW08] propose the Geometric Element Transformation Method (GETMe) to successively regularize low quality triangles by iteratively applying a geometric transformation until a user defined quality level is reached. Generalizations for general polygonal, tetrahedral [VW12] and later general polyhedral meshes are suggested, but the official GETMe implementation currently only supports smoothing 2D polygonal shapes with fixed boundary vertices. Our approach sets itself apart by optimizing the complete polygonal shape, and not only the interior angles at vertices. It is specifically tailored for the use of our proposed Laplace operator, taking into account the entire polygonal configuration in relation to the virtual vertex.

3. Laplacian on Triangle meshes

In this section, we provide a brief introduction to the concept of linear Lagrange basis functions on triangle meshes and revisit the definition of the cotangent Laplacian. Consider a triangle mesh $\mathcal{M}_T = (\mathcal{V}, \mathcal{T})$ defined by a set of vertices $\mathcal{V} = \{v_1, \dots, v_m\}$ and triangle faces $\mathcal{T}$, where each vertex $v_i \in \mathcal{V}$ has a 3D position $\mathbf{x}_i = (x_i, y_i, z_i)$. The linear Lagrange basis functions ψ_i are unique, C^0 continuous, piecewise linear functions associated with the vertices v_i. They satisfy the Lagrange interpolation property

$$\psi_i(\mathbf{x}_j) = \delta_{ij} := \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{otherwise}, \end{cases} \tag{1}$$

and are linear on each triangle $t \in \mathcal{T}$, making them piecewise linear on the mesh. The finite element method (FEM) uses these basis functions to interpolate a given function u with the degrees of freedom u_i associated with vertices:

$$u(\mathbf{x}) \approx \sum_{i=1}^{m} u_i \psi_i(\mathbf{x}).$$

Therefore, instead of solving a partial differential equation directly, FEM approaches solve for suitable coefficients u_i that approximate the unknown solution u. By construction, the linear Lagrange basis functions form a partition of unity and satisfy the linear precision property.

The Lagrange basis $\{\psi_1, \dots, \psi_m\}$ on the triangle mesh $\mathcal{M}_T$ allows to define the strong form of the cotangent Laplace operator

$$\mathbf{L} = -\left(\mathbf{M}^{\triangle}\right)^{-1} \mathbf{S}^{\triangle}, \tag{2}$$

using mass and stiffness matrices $\mathbf{M}^{\triangle}, \mathbf{S}^{\triangle} \in \mathbb{R}^{m \times m}$. Note that in the geometry processing literature, the term “cotangent Laplacian” often refers to the stiffness matrix $\mathbf{S}^{\triangle}$ only (the integrated weak form) and omits the mass matrix $\mathbf{M}^{\triangle}$. Both matrices are defined as

$$\mathbf{M}^{\triangle}_{ij} = \int_{\mathcal{M}_T} \psi_i \psi_j = \begin{cases} \dfrac{|t_{ijk}| + |t_{jih}|}{12} & \text{if } v_j \in \mathcal{N}(v_i) \\ \displaystyle\sum_{v_k \in \mathcal{N}(v_i)} \mathbf{M}^{\triangle}_{ik} & \text{if } j = i, \\ 0 & \text{otherwise}, \end{cases} \tag{3}$$

$$\mathbf{S}^{\triangle}_{ij} = \int_{\mathcal{M}_T} \langle \nabla\psi_i, \nabla\psi_j \rangle = \begin{cases} -\dfrac{\cot\alpha_{ij} + \cot\beta_{ij}}{2} & \text{if } j \in \mathcal{N}(v_i), \\ -\displaystyle\sum_{v_k \in \mathcal{N}(v_i)} \mathbf{S}^{\triangle}_{ik} & \text{if } j = i, \\ 0 & \text{otherwise}. \end{cases} \tag{4}$$

Here, the triangles t_{ijk} and t_{jih} are adjacent along the edge $e_{ij} = (v_i, v_j)$ and their areas are denoted by $|t_{ijk}|$ and $|t_{jih}|$. The respective interior angle of t_{ijk} and t_{jih} opposite to the edge e_{ij} are given by α_{ij} and β_{ij}, while $\mathcal{N}(v_i)$ refers to the one-ring vertex neighborhood around v_i. The cotangent of the corner angle θ_k at vertex v_k within a triangle t_{ijk} is given as the following ratio between squared triangle edge lengths and triangle area

$$\cot\theta_k = \frac{|e_{jk}|^2 + |e_{ik}|^2 - |e_{ij}|^2}{4\,|t_{ijk}|}. \tag{5}$$

The scalar $|e_{ij}|$ denotes the length of edge e_{ij}. Note that the definition of our stiffness matrix is *positive* semi-definite in contrast to the *negative* semi-definite Laplacian matrix $\mathbf{L}$. While the continuous Laplacian is a negative semi-definite operator and should be treated as such in the discrete setting, our choice for the sign of the stiffness matrix establishes that its eigenvalues are all greater or equal to zero, which makes them more convenient to handle.

The key principle of FEM is to divide a large, global problem into smaller and more manageable local equivalents.. In this spirit, we can obtain the same matrices as defined in Equations (3) and (4) by constructing local mass and stiffness matrices $\mathbf{M}_t^\triangle, \mathbf{S}_t^\triangle \in \mathbb{R}^{3\times 3}$ per triangle t:

$$\begin{aligned} \left(\mathbf{M}_t^\triangle\right)_{ij} &= \int_t \psi_i \psi_j, & (6)\\ \left(\mathbf{S}_t^\triangle\right)_{ij} &= \int_t \langle \nabla\psi_i, \nabla\psi_j \rangle . & (7)\end{aligned}$$

These local matrices are then assembled into $\mathbf{M}^\triangle$ and $\mathbf{S}^\triangle$ by assigning each vertex v_i the i-th row and column of the global matrix. The entries are then the sum over the respective values in the local triangle matrices in which the vertex v_i was involved.

4. Laplacian on Polygon Meshes

In this section we briefly introduce the polygon Laplacian based on virtual refinement proposed by Bunge et al. [BHKB20].

Consider a polygon mesh $\mathcal{M} = (\mathcal{V}, \mathcal{F})$ embedded in $\mathbb{R}^3$ with vertices $\mathcal{V}$ and faces $\mathcal{F}$. The virtual refinement method defines a set of polygonal shape functions $\{\varphi_1, \ldots, \varphi_m\}$ by introducing *virtual* vertices within each polygon of the mesh. These polygons can be non-planar and non-convex, but have to be star-shaped in order for the method to work as intended.

We explain the concept for a single polygon $f \in \mathcal{F}$ with n vertices $(v_1, \ldots, v_n)$. Inserting a virtual vertex v_f turns the polygon into a virtual triangle fan around v_f. The position of the virtual vertex is expressed as an affine combination of the polygon vertices:

$$\mathbf{x}_f = \sum_{i=1}^{n} w_i \mathbf{x}_i \quad \text{with} \quad \sum_{i=1}^{n} w_i = 1. \tag{8}$$

We define $\mathbf{w}_f = (w_1, \ldots, w_n) \in \mathbb{R}^n$ as the vector containing the affine weights w_i of each face f. For a planar polygon, the virtual vertex should be placed within the kernel of the face, which refers to the set of points from which the entire boundary of the polygon is visible. This prevents flipped virtual triangles around $\mathbf{x}_f$ and thus does not introduce negative areas. For arbitrary polygons, Bunge et al. suggest the minimizer of the sum of squared areas of the triangles around $\mathbf{x}_f$. This point is guaranteed to lie in the kernel of the polygon if it happens to be planar. Since the affine weights $\mathbf{w}_f$ that represent $\mathbf{x}_f$ according to (8) are underdetermined, they propose to use the solution with least L_2 norm.

Introducing the virtual point allows to define linear Lagrange basis functions ψ_i on the virtual triangle fan, which are combined into the final polygon shape functions as

$$\varphi_i = \psi_i + w_i \psi_f, \tag{9}$$

where the virtual degree of freedom at v_f is redistributed to the original vertices v_i through the chosen weights w_i. The polygon shape functions are then used to define mass and stiffness matrix in the same manner as described in Equations (3) and (4).

Alternatively, one can combine the affine weights of the polygon into a local $(n+1) \times n$ prolongation matrix

$$\mathbf{P}_{ij} = \begin{cases} \delta_{ij} & \text{for } 1 \le i \le n, \\ w_j & \text{otherwise.} \end{cases} \tag{10}$$

The local polygon Laplacian can then be obtained by constructing the cotangent mass and stiffness matrices $\mathbf{M}^\triangle$ and $\mathbf{S}^\triangle$ on the virtual triangle fan and "sandwiching" them with $\mathbf{P}$:

$$\begin{aligned} \mathbf{M}^\circ &= \mathbf{P}^\mathsf{T} \mathbf{M}^\triangle \mathbf{P}, & (11)\\ \mathbf{S}^\circ &= \mathbf{P}^\mathsf{T} \mathbf{S}^\triangle \mathbf{P}. & (12)\end{aligned}$$

Following Bunge et al.'s [BHKB20] recommendation, we use the diagonal (lumped) mass matrix.

5. Accuracy and Robustness on Triangle Meshes

For linear FEM, Shewchuk has analyzed how the shape of triangles affects various aspects of the discretization [She02]. He argues that for the interpolation error it is more important to control the error in the gradients and gives a smooth, *size-independent* measure of the quality of an element in this respect [She02, Tab. 4: scale-invariant (smooth) gradient interpolation quality measure.]

$$\frac{|t_{ijk}|}{\left(|e_{ij}|^2 \, |e_{jk}|^2 \, |e_{ik}|^2\right)^{\frac{1}{3}}}, \tag{13}$$

where larger ratios indicate better shaped triangles. For analyzing (or improving) the stiffness matrix' condition number, again with a smooth and scale-independent measure, he suggests [She02, Tab. 4: scale-invariant (smooth) conditioning quality measure.]

$$\frac{|t_{ijk}|}{\frac{1}{3}\left(|e_{ij}|^2 + |e_{jk}|^2 + |e_{ik}|^2\right)}. \tag{14}$$

The condition number κ of a quadratic matrix $\mathbf{A}$ is defined as the ratio between its largest and smallest eigenvalue

$$\kappa(\mathbf{A}) = \frac{\lambda_{max}(\mathbf{A})}{\lambda_{min}(\mathbf{A})} \tag{15}$$

and is a common measure to quantify numerical stability. Equation (14) is inversely proportional to the *harmonic index* of the triangle [Mus97, BS07, CXGL10]:

$$\eta\left(t_{ijk}\right) = \frac{|e_{ij}|^2 + |e_{jk}|^2 + |e_{ik}|^2}{|t_{ijk}|} = 4 \sum_{v_i \in t} \cot\theta_i. \tag{16}$$

Notice that the two quality measures differ only by how the average is taken over the squares of the three edge lengths. By the AM-GM inequality, the arithmetic mean in the denominator of the condition quality measure (14) is not smaller than the geometric mean in the denominator of the interpolation error measure (13). This means that *minimizing* the harmonic index $\eta(t_{ijk})$ improves the condition of the stiffness matrix w.r.t. the triangle t_{ijk}, and generally also improves the accuracy of the gradient in the solution. This is the case, because triangles with large angles contain points where the gradient interpolation error explodes [She02]. Since the minimization of the harmonic index avoids angles close to 0 and 180 degrees, these shapes do not occur, leading to well-behaved gradients.

The work of Alexa [Ale19] shows that the trace of the stiffness matrix (per element or for the whole mesh) is proportional to the sum of the harmonic indices of the triangle(s):

$$\begin{aligned} H(\mathcal{T}) &= \sum_{t_{ijk}\in\mathcal{T}} \eta(t_{ijk}) && (17)\\ &= 4\sum_{t_{ijk}\in\mathcal{T}} \cot\theta_i+\cot\theta_j+\cot\theta_k && (18)\\ &= 4\,\mathrm{tr}(\mathbf{S}^{\triangle}). && (19)\end{aligned}$$

In the following, we will use the idea of minimizing the trace of the triangle cotan operator for optimizing the operators for polygon meshes as well as improving vertex positions.

6. Accuracy and Robustness on Polygon Meshes

In Section 5 we established that a smaller trace of the cotangent stiffness matrix generally indicates a numerically preferable triangle mesh. In this section, we aim to investigate whether the same observation holds for the trace of the polygon stiffness matrix defined by Bunge et al. [BHKB20] and whether we can establish a similar connection to the geometry of the polygon.

When dealing with triangles, the cotangent Laplacian is solely based on the tessellation and cannot be altered unless the triangle mesh is modified. However, when working with polygons, we have two degrees of freedom to consider: the affine weights used in the prolongation matrix and the placement of the virtual vertex. As a result, we can optimize these parameters with respect to the trace of the polygon stiffness matrix *without* changing the polygon mesh and examine how it affects the Laplacian.

6.1. Optimal Prolongation Weights

We will start with the affine weights for the virtual vertex. As in Section 4, assume we are given a polygon $f\in\mathcal{F}$ with vertices $(v_1,\dots,v_n)$ and its corresponding virtual face vertex v_f. Let $\mathbf{S}^{\triangle}\in\mathbb{R}^{(n+1)\times(n+1)}$ denote the local cotangent stiffness matrix constructed on the virtual triangulation and $\mathbf{w}_f=(w_1,\dots,w_n)^{\mathsf{T}}\in\mathbb{R}^n$ the set of affine weights. We build the local prolongation matrix $\mathbf{P}\in\mathbb{R}^{(n+1)\times n}$ as defined in (10) and obtain the local polygon stiffness matrix $\mathbf{S}^{\circ}$ as established in (12).

The objective is to minimize the trace of $\mathbf{S}^{\circ}$ with respect to the individual prolongation weights $w_i\in\mathbf{w}_f$. We first rearrange the polygon stiffness matrix into a similar pattern as in Equation (9): We divide the matrix into the cotangent entries related to the existing polygon vertices and a matrix that redistributes the values associated with the virtual vertex v_f:

$$\mathbf{S}^{\circ} = \mathbf{P}^{\mathsf{T}}\mathbf{S}^{\triangle}\mathbf{P} = \mathbf{S}^{\triangle}_{\mathrm{sub}}+\mathbf{W}. \qquad (20)$$

The matrix $\mathbf{S}^{\triangle}_{\mathrm{sub}}$ is the symmetric $n\times n$ submatrix of $\mathbf{S}^{\triangle}$ excluding the row and column associated with the virtual vertex. In our setting, the cotangent weights associated with the virtual vertex are located at row/column $(n+1)$ of $\mathbf{S}^{\triangle}$. We will use the vector

$$\mathbf{s}=\left(\mathbf{S}^{\triangle}_{1,n+1},\dots,\mathbf{S}^{\triangle}_{n,n+1}\right)^{\mathsf{T}}\in\mathbb{R}^n$$

and its respective entries $(s_1,\dots,s_n)$ to refer to the first n entries of the virtual vertex column. The matrix $\mathbf{W}\in\mathbb{R}^{n\times n}$ can be defined as

$$\mathbf{W}_{ij}=\begin{cases} 2w_is_i-w_i^2\sum_{k=1}^{n}s_k & \text{if } i=j,\\ w_is_j+w_js_i-w_iw_j\sum_{k=1}^{n}s_k & \text{otherwise,}\end{cases} \qquad (21)$$

and redistributes the values of the virtual vertex onto the original n nodes of the polygon.

We minimize the trace of the polygon stiffness matrix with respect to the affine weights $w_1,\dots,w_n$ by setting the respective partial derivatives to zero. Here we only have to consider the trace of the matrix $\mathbf{W}$, since $\mathbf{S}^{\triangle}_{\mathrm{sub}}$ does not depend on the weights w_i:

$$\frac{\partial\,\mathrm{tr}(\mathbf{S}^{\circ})}{\partial w_i}=\frac{\partial\,\mathrm{tr}(\mathbf{W})}{\partial w_i}=2s_i-2w_i\sum_{j=1}^{n}s_j. \qquad (22)$$

Setting Equation (22) to zero leads to the trace-optimal weights

$$w_i=\frac{-(\cot\alpha_{i,n+1}+\cot\beta_{i,n+1})}{-\sum_{j=1}^{n}\cot\alpha_{j,n+1}+\cot\beta_{j,n+1}}=\frac{s_i}{\sum_{j=1}^{n}s_j}=:\frac{s_i}{\omega}, \qquad (23)$$

with $\omega\in\mathbb{R}$ referring to the negated $(n+1)$-th diagonal entry of $\mathbf{S}^{\triangle}$ associated with the virtual vertex.

These trace-minimizing weights are well-known as the *discrete harmonic coordinates* [PP93, EDD*95] and can be computed through Equation (23) without numerical optimization, in contrast to the least norm weights suggested in the original paper. Note that these weights are affine, since they by construction sum to one, and that the denominator ω is guaranteed to be non-zero as long as the virtual vertex v_f is positioned in the kernel of a star-shaped polygon.

We point out that using these weights in the prolongation step simplifies the entries of the matrix $\mathbf{W}$ to

$$\mathbf{W}_{ij}=\left\{\begin{matrix} s_i^2/\omega & \text{if } i=j\\ s_is_j/\omega & \text{otherwise}\end{matrix}\right\}=\frac{1}{\omega}\mathbf{s}\mathbf{s}^{\mathsf{T}}, \qquad (24)$$

which we will exploit in the upcoming section.

6.2. Eigenvalues of the Polygon Stiffness Matrix

In this section, we uncover a geometric link between the local eigenvalues of the polygon stiffness matrix and the polygon itself. Assume the local prolongation matrix $\mathbf{P}$ on the polygon f is obtained with the discrete harmonic weights established in the previous section. Given that $\mathbf{S}^{\circ}$, $\mathbf{S}^{\triangle}_{\mathrm{sub}}$, and $\mathbf{W}$ are all $n\times n$ Hermitian matrices and $\mathbf{S}^{\circ}=\mathbf{S}^{\triangle}_{\mathrm{sub}}+\mathbf{W}$, we can make the following observation: Let μ_i,ν_i and ρ_i be the respective eigenvalues of $\mathbf{S}^{\circ}$, $\mathbf{S}^{\triangle}_{\mathrm{sub}}$, and $\mathbf{W}$, ordered as follows:

$$\begin{aligned} \mathbf{S}^{\circ}:&\quad \mu_1\geq\cdots\geq\mu_{n-1}>\mu_n=0, && (25)\\ \mathbf{S}^{\triangle}_{\mathrm{sub}}:&\quad \nu_1\geq\cdots\geq\nu_{n-1}\geq\nu_n, && (26)\\ \mathbf{W}:&\quad \rho_1\geq\cdots\geq\rho_{n-1}\geq\rho_n. && (27)\end{aligned}$$

Applying Weyl's inequality [Wey12] yields

$$\nu_i+\rho_n\leq\mu_i\leq\nu_i+\rho_1,\quad i=1,\dots,n. \qquad (28)$$

which links the individual eigenvalues of the polygon stiffness matrix $\mathbf{S}^{\bigcirc}$ to the eigenvalues of the submatrix $\mathbf{S}^{\triangle}_{\text{sub}}$.

But what about the eigenvalues of $\mathbf{W}$? Using the discrete harmonic coordinates as prolongation weights allows us to directly determine the eigenvalues ρ_i. As an outer-product matrix (see Equation (24)) $\mathbf{W}$ has rank one and a kernel of dimension $n-1$, leading to $n-1$ vanishing eigenvalues. The only remaining non-zero eigenvalue is

$$\rho = \frac{\|\mathbf{s}\|^2}{\omega} \tag{29}$$

with the eigenvector $\mathbf{s}$, since

$$\mathbf{W}\mathbf{s} = \left(\frac{1}{\omega}\mathbf{s}\mathbf{s}^{\mathsf{T}}\right)\mathbf{s} = \frac{\|\mathbf{s}\|^2}{\omega}\mathbf{s}. \tag{30}$$

The question remains whether ρ is the largest or the smallest eigenvalue of $\mathbf{W}$. Since $\mathbf{S}^{\triangle}$ is positive semi-definite, ω is negative, which means that $\rho_n = \rho$ is negative as well and is therefore the smallest eigenvalue of $\mathbf{W}$. This allows us to simplify Equation (28) to

$$\nu_i + \frac{\|\mathbf{s}\|^2}{\omega} \le \mu_i \le \nu_i \quad \text{for } i = 1,\ldots,n. \tag{31}$$

Therefore, as a first conclusion, we can observe that the i-th eigenvalue μ_i of the polygon stiffness matrix is confined by the i-th eigenvalue ν_i of the submatrix $\mathbf{S}^{\triangle}_{\text{sub}}$.

Conveniently, the matrix $\mathbf{S}^{\triangle}_{\text{sub}}$ is an n-dimensional principal submatrix of the $(n+1)$-dimensional symmetric cotangent stiffness matrix $\mathbf{S}^{\triangle}$, allowing us to apply Cauchy's Interlace Theorem [Hwa04], which reveals the following eigenvalue relationship:

$$0 = \lambda_{n+1} \le \nu_n \le \lambda_n \le \cdots \le \lambda_2 \le \nu_1 \le \lambda_1, \tag{32}$$

with λ_i denoting the $n+1$ eigenvalues of the cotangent stiffness matrix $\mathbf{S}^{\triangle}$. Merging this inequality with the upper and lower bounds from Equation (31) results in the final bounds

$$\lambda_{i+1} + \frac{\|\mathbf{s}\|^2}{\omega} \le \nu_i + \frac{\|\mathbf{s}\|^2}{\omega} \le \mu_i \le \nu_i \le \lambda_i, \quad i = 1,\ldots,n. \tag{33}$$

We are now able to draw a powerful conclusion: All eigenvalues of the polygon stiffness matrix $\mathbf{S}^{\bigcirc}$ are less than or equal to the corresponding eigenvalues of the cotangent stiffness matrix $\mathbf{S}^{\triangle}$.

Therefore, the upper bound in Equation (33) directly establishes a geometric connection between the stiffness matrix and the polygon, albeit indirectly through the virtual triangle fan. Polygons that allow their virtual vertex to form a high-quality triangle fan (by the standards established in Section 5) can minimize the trace of the virtual cotangent stiffness matrix and, consequently, improve that of the polygon stiffness matrix. Not only that, but a good virtual triangulation also implies a cotan matrix of higher quality, which also improves the performance of the polygon Laplacian. Furthermore, the condition number of the polygon stiffness matrix is expected to improve proportionally to that of the cotan Laplacian.

6.3. Optimal Choice for Virtual Vertex Placement

The findings from the previous section establish a suitable objective function for the remaining degree of freedom: the placement of the virtual vertex. The point that minimizes (within the kernel of the polygon) the trace of the cotangent stiffness matrix will also minimize the harmonic index of the virtual triangle fan (see Equation (17)) and consequently further improve the quality of individual (virtual) triangles.

We start with the case of a *planar polygon* and consider the cotan stiffness matrix $\mathbf{S}^{\triangle}$ on the virtual triangles $\mathcal{T}_f$ of a polygon f with virtual vertex v_f. We want to minimize the following energy with respect to the virtual vertex position $\mathbf{x}_f$:

$$E(\mathbf{x}_f) = \operatorname{tr}\left(\mathbf{S}^{\triangle}\right) = \sum_{t\in\mathcal{T}_f}\sum_{v_j\in t}\cot(\theta_j). \tag{34}$$

We minimize this energy using the projected Newton method, which ensures a positive definite Hessian, and combine it with line search. In order to prevent virtual triangles from flipping, we start the minimization from the squared area minimizer of Bunge et al. [BHKB20] (which is guaranteed to lie in the polygon's kernel) and return an infinite cost during the line search in case of negative or degenerate triangle areas. This avoids artifacts caused by the flipped triangles and guarantees the validity of our prolongation weights, since ω cannot become zero. Combining Equation (34) and the penalty term leads to a virtual vertex position that minimizes the trace of the stiffness matrix within the polygon's kernel.

The discrete harmonic coordinates, as mentioned by Hormann and Floater [HF06], can represent points in the interior of a polygon as long as the resulting virtual triangle areas are positive and the edges connecting the virtual point to the polygon vertices remain non-degenerate. While a convex polygon satisfies these criteria for any point in its interior, star-shaped polygons guarantee these properties solely within the element's kernel. This condition is non-negotiable since reproducing the virtual vertex through the prolongation weights is necessary to retain the linear precision property for the polygon Laplacian [BHKB20].

The above approach, however, does not extend to polygon meshes embedded in $\mathbb{R}^3$ with potentially non-planar faces. First, we lose the notion of signed triangle area and flipped triangles. Second, for non-planar polygons only a limited set of points can be represented by discrete harmonic coordinates: Assuming that the virtual point $\mathbf{x}_f$ is represented by the weights w_i, and exploiting that the discrete harmonic weights are just the (normalized) cotangent weights for discretizing the Laplacian, we get

$$\sum_{j=1}^{n} w_j \mathbf{x}_j = \mathbf{x}_f \tag{35}$$

$$\Leftrightarrow \quad \sum_{j=1}^{n} \frac{s_j}{\omega}\mathbf{x}_j = \sum_{j=1}^{n} \frac{s_j}{\omega}\mathbf{x}_f \tag{36}$$

$$\Leftrightarrow \quad \sum_{j=1}^{n} s_j\left(\mathbf{x}_j - \mathbf{x}_f\right) = \mathbf{0} \tag{37}$$

$$\Leftrightarrow \quad \Delta\mathbf{x}_f = \mathbf{0}. \tag{38}$$

As the cotangent Laplacian corresponds exactly to the gradient of surface area [DMSB99], the last condition means that the virtual vertex has to minimize the sum of virtual triangle areas for the weights (23) to be able to reproduce the point. This is not an issue in the planar setting, where any point leads to the same area. For

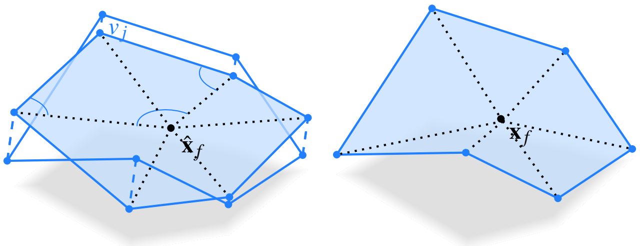


Figure 2: *Left: Non-planar polygon and its orthogonal projection with maximum area. The projection's virtual vertex position $\hat{\mathbf{x}}_f$ is the trace minimizer of the planar cotangent stiffness matrix. Right: The virtual vertex position $\mathbf{x}_f$ is obtained by multiplying the original vertices with the discrete harmonic weights of the projection.*

non-planar polygons, however, the trace minimizer and the area minimizer do in general not coincide, leading to a discontinuous energy when slightly deviating from a planar configuration.

We suggest an alternative approach that is able to retain linear precision: Given a non-planar polygon f, we consider its orthogonal planar projection with maximum surface area, as proposed by Alexa and Wardetzky [AW11]. To this end we project the polygon vertices $\mathbf{x}_i$ along the polygon's normal $\mathbf{n}_f$, which itself is defined in terms of the vector area $\mathbf{a}_f$:

$$\mathbf{a}_f = \frac{1}{2}\sum_{v_i \in f} \left(\mathbf{x}_i \times \mathbf{x}_{i+1}\right), \quad \mathbf{n}_f = \frac{\mathbf{a}_f}{\|\mathbf{a}_f\|}, \quad \hat{\mathbf{x}}_i = \mathbf{x}_i - \mathbf{n}_f \mathbf{n}_f^\mathsf{T} \mathbf{x}_i. \quad (39)$$

For the planar configuration $(\hat{\mathbf{x}}_1, \ldots, \hat{\mathbf{x}}_n)$ we minimize the trace of the stiffness matrix with respect to $\hat{\mathbf{x}}_f$ as described above. The resulting discrete harmonic weights $\hat{\mathbf{w}}_f$ are then used to define the virtual point for the original, non-planar polygon as $\mathbf{x}_f := \sum_j \hat{w}_j \mathbf{x}_j$.

For non-planar polygons that are still relatively close to a planar configuration, the virtual vertex will only slightly deviate from the trace minimizer of the orthogonal projection, as illustrated in Figure 2. This approach, therefore, avoids the discontinuity between the trace minimizer and the area minimizer. However, as the polygons become increasingly distorted, the results from this approach may become less favorable as the positions of the original and projected vertices increasingly differ. To further stabilize our approach, we integrate the fallback of comparing the trace of the optimized local polygon stiffness matrix to that of the original method. If ours is higher, we use the original point and weights instead. However, this is only necessary if the orthogonal projection of the polygon does not satisfy our initial assumption of being star-shaped, and almost never occurred in our experiments.

Although our energy minimizing requires an iterative approach, the implementation of is quite straightforward. Using the maximum orthogonal projection, we can express everything in an intrinsic 2D coordinate system, making the 2 × 2 Newton solver converge in just a few steps. The gradients and Hessian of Equation (34) can be derived analytically [Cra23], as well as the Hessian's eigenvalues required for the projection step.

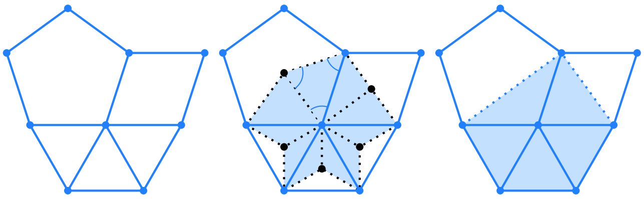


Figure 3: *In a mixed polygonal mesh (left) we also incorporate virtual vertices in the one-ring (center) whereas other methods often only consider real vertices (right) [Knu00, GS04, VAGW08].*

7. Polygon Mesh Smoothing

In the previous sections we explored the intricate relationship between virtual vertex positions, affine prolongation weights, and the eigenvalues of the polygon Laplacian. We established that reducing the trace of the stiffness matrix improves the numerical quality of the Laplacian. So far, however, we have only focused on the degrees of freedom of the discrete Laplace operator itself, not on the "real" non-virtual vertices of the polygon mesh. If we also take these into account when optimizing the trace of the stiffness matrix, we can improve the numerical conditioning much further.

Consider a polygon mesh $\mathcal{M} = (\mathcal{V}, \mathcal{F})$ with m vertices $\mathbf{x}_1, \ldots, \mathbf{x}_m$ and a virtual triangulation $\mathcal{M}^\triangle = (\mathcal{V}^\triangle, \mathcal{T})$ generated by inserting a virtual vertex into each polygonal face. Following the insights from Sections 5 and 6, we want to optimize the trace of the cotan Laplacian on the virtual triangulation $\mathcal{T}$ with respect to the positions of the polygon vertices $v_i \in \mathcal{V}$, as shown in Figure 3, center. The trace of the global cotan stiffness matrix can be formulated as the energy

$$E(\mathbf{x}_1, \ldots, \mathbf{x}_m) = \sum_{t \in \mathcal{T}} \sum_{v_j \in t} \cot(\theta_j), \quad (40)$$

which sums up local cotangent components of the corner angles θ_j at the vertices v_j within all virtual triangles $t \in \mathcal{T}$. We minimize this energy in a Newton-like manner, but decouple the relationship of real and virtual vertices in an alternating optimization scheme, such that each Newton iteration has these two steps:

1. Minimize trace (as in Equation (40)) with respect to all real vertex positions $\mathbf{x}_1, \ldots, \mathbf{x}_m$, while keeping (the affine weights of) virtual vertices fixed.
2. For each face f, minimize trace (as in Equation (34)) with respect to virtual vertex positions $\mathbf{x}_f$ and weights $\mathbf{w}_f$, while keeping the real vertices fixed.

This formulation not only simplifies the implementation, it also increases the robustness of the numerical optimization. We implement the smoothing (Step 1) using TinyAD [SBB*22].

When optimizing polygon meshes embedded in $\mathbb{R}^3$, we have to restrict vertex movement to the mesh surface to prevent deviation of the optimized mesh from the original geometry. This is typically done by restricting vertex movement to their tangent planes and/or re-projecting vertices to the original mesh after each iteration. For technical models with sharp features, however, this will not accurately preserve feature edges and corners.

We propose to instead adapt the quadric error metric [GH97] to our setting and to directly incorporate it into the optimization. At initialization time, each vertex v_i is assigned a 4 × 4 quadric $\mathbf{Q}_i$ built by summing up the quadrics of v_i's incident faces f. As proposed

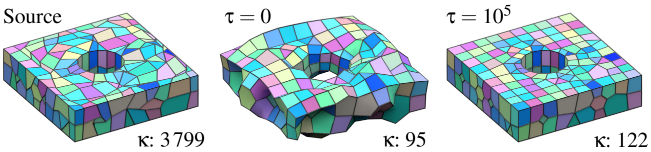


Figure 4: *Without constraints, surface meshes may get disfigured (center). Due to the quadrics energy, vertices move only on the surface and sharp features are preserved automatically (right). The condition number* κ *of the polygon stiffness matrix* $\mathbf{S}^{\varhexagon}$ *significantly improves on our smoothed tessellation compared to the input mesh.*

by Garland and Heckbert [GH97], the face quadrics are constructed from area-weighted normal vectors, i.e., from the vector areas $\mathbf{a}_f$. To optimize the trace of the stiffness matrix while keeping vertices on the surface and on the features, we minimize the energy

$$E(\mathbf{x}_1, \ldots, \mathbf{x}_m) = \sum_{t \in \mathcal{T}} \sum_{v_j \in t} \cot(\theta_j) + \tau \sum_{v_j \in \mathcal{V}} Q_j(\mathbf{x}_j), \qquad (41)$$

where

$$Q(\mathbf{x}) = (\mathbf{x}^{\mathsf{T}}, 1)\, \mathbf{Q} \begin{pmatrix} \mathbf{x} \\ 1 \end{pmatrix} \qquad (42)$$

denotes the quadric error of vertex position $\mathbf{x}$ with respect to the quadric $\mathbf{Q}$. As the τ parameter can be adjusted to fine-tune the optimization process, it is important to note that the sum of cotan values often tend to become relatively large. Consequently, to effectively balance the influence of cotan sums and quadric errors, higher τ values perform better and we used $\tau = 10^5$ for all results. While the quadrics could be updated after each Newton iteration, we typically keep them fixed to restrict the mesh optimization to rather local modifications.

Figure 4 compares the results of unconstrained ($\tau = 0$) and quadric-constrained ($\tau = 10^5$) optimization. While in the unconstrained case a lower condition number κ is achieved, this comes at the price of an unacceptable deviation from the original shape. Figure 5 compares to the polygon mesh smoothing of Garimella et al. [GS04]. Their energy formulation for the apex vertex has multiple equivalent minima on the lateral sides of the pyramid, which crumples down the pyramid in one iteration and would eventually flatten it completely. Our energy is minimal at the apex itself due to the symmetry of the constellation, thus the vertex is not moved (even without quadric constraints).

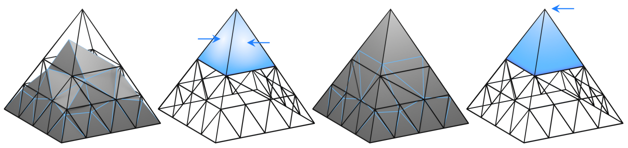


Figure 5: *Results of Garimella [GS04] (left) and ours (right). As highlighted, their energy function has multiple equivalent minima for the apex vertex on all four sides, thus crumples the pyramid, whereas ours remains straight as the minimum is exactly at the top.*

8. Results and Discussion

In this section we present experiments, comprehensive comparisons, and discussion of the results, highlighting the effectiveness of our approach on mixed polygon meshes.

8.1. Optimized Polygon Laplacian

We analyze the performance of our optimized polygon Laplacian in a set of computer graphics applications. We compare our results to those obtained using several other state-of-the-art operators. Our comparison includes the original linear virtual refinement method by Bunge et al. [BHKB20], as well as the DEC operators presented by Alexa and Wardetzky [AW11] and de Goes et al. [dGBD20], along with the FV Diamond Laplacian by Bunge et al. [BBA21]. We evaluate two choices for the respective stabilization parameter λ of the DEC operators. One option is the suggested hyper-parameter from the original paper, and the other is the parameter that yielded the best results in most of the test cases. For Alexa and Wardetzky, this corresponds to $\lambda = 2$ (suggested) and $\lambda = 0.5$ (optimal). In the case of de Goes et al., we consider $\lambda = 1$ (suggested) and $\lambda = 0.1$ (optimal). The results vary depending on the selected test mesh or application, suggesting that alternative hyper-parameter configurations tailored to specific mesh types or applications could further improve the results for the respective DEC operator. The implementation of the Laplacian operators and test applications is based on the source code for Bunge et al.'s survey paper [BB23]. Although we provide a concise summary of the test methods presented, we encourage readers to consult the survey for more details. Figure 16 illustrates a selection of the test meshes used in our evaluation.

Poisson Equation The *Poisson equation*, given by $-\Delta u = f$, is often used to analyze the convergence rate of a discrete Laplacian. On a polygon mesh $\mathcal{M}$ with m vertices, we consider the discretized Poisson equation for the Laplacian matrix $\mathbf{L} = -\mathbf{M}^{-1}\mathbf{S} \in \mathbb{R}^{m \times m}$

$$-\mathbf{L} \cdot \mathbf{u} = \mathbf{f} \quad \Leftrightarrow \quad \mathbf{S} \cdot \mathbf{u} = \mathbf{M} \cdot \mathbf{f} \qquad (43)$$

with Dirichlet boundary conditions. We solve for the discretization of the analytically calculated Laplacian of the 2D Franke test function [Fra79, BB23], sampled at the vertices. The convergence plots in Figure 6 show the L_2 errors rates on different planar tessellations. Our method consistently maintains low error rates and outperforms all other operators on several meshes. However, the accuracy of more regular shapes, like the Voronoi mesh, does not significantly differ from that of the original method. The operator of de Goes et al. [dGBD20] outperforms our method on two tessellations for the hyper-parameter $\lambda = 1$. However, on the other tessellations this parameter leads to less favorable results, while our method remains one of the most accurate.

Spherical Harmonics The analytic eigenfunctions of the smooth Laplacian on the unit sphere S^2 are referred to as the *spherical harmonics* $Y_l^m : S^2 \to \mathbb{R}$ with eigenvalues $\mu = -l(l+1)$ [PTVF07]. To compare different Laplacians on non-planar meshes, we discretize Y_3^2 with eigenvalue $\mu = -12$ to the vector $\mathbf{y}_3^2 \in \mathbb{R}^m$ on polygon tessellations of the unit sphere. The vector can then be used to solve for $\mathbf{f} \in \mathbb{R}^m$ according to the Poisson equation [BB23]. Figure 7 shows the obtained results. As for the planar meshes, our operator is able to outperform the other operators on several meshes. In the other cases, it is either on par with the optimized version of Alexa and Wardetzky's operator or outperforms the DEC Laplacians. The Diamond operator is able to yield the lowest error on Voronoi spheres.

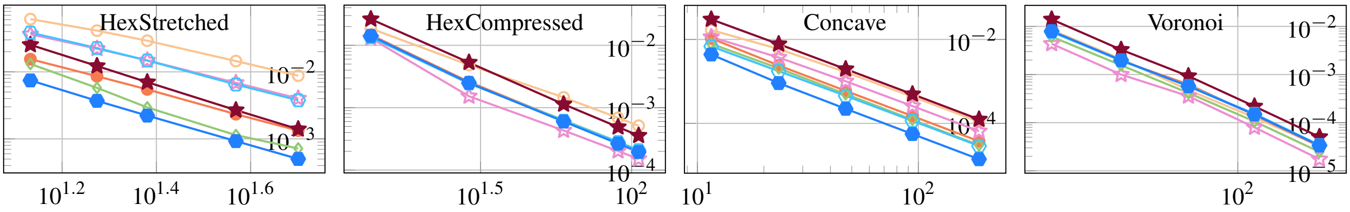


Figure 6: *Poisson solve for the 2D Franke test function in log-log scale. The errors were evaluated on different polygon tessellations of the unit plane. x-axis: inverse mean edge length, y-axis: L_2 error.*
Legend: [AW11] $\lambda =$ ○ 2, ● 0.5, [dGBD20] $\lambda =$ ☆ 1, ★ 0.1. [BBA21] ♢, [BHKB20] ⬡ default, ⬢ ours.

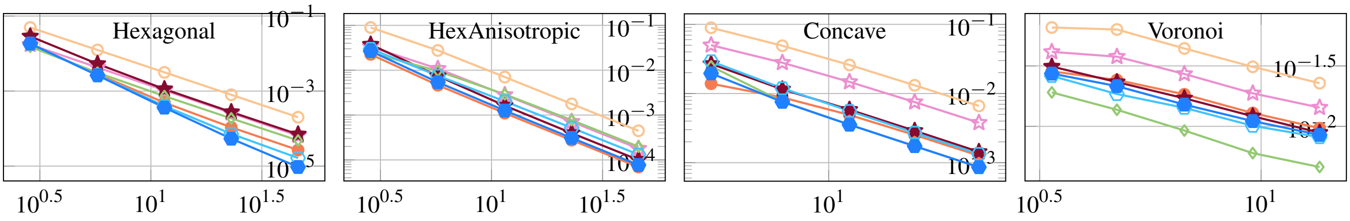


Figure 7: *Poisson solve for the spherical harmonic function Y_3^2 in log-log scale on different tessellations of the unit sphere. x-axis: inverse mean edge length, y-axis: L_2 error.*
Legend: [AW11] $\lambda =$ ○ 2, ● 0.5, [dGBD20] $\lambda =$ ☆ 1, ★ 0.1. [BBA21] ♢, [BHKB20] ⬡ default, ⬢ ours.

Geodesics in Heat The *geodesic distance* can be evaluated on polygon meshes and requires, besides the Laplacian, discrete divergence and gradient operators. We use the geodesics in heat method by Crane et al. [CWW13] to calculate the geodesic distances between one vertex v_i and all other vertices of the mesh. As suggested by de Goes et al. [dGBD20, dGDMD16] we use the squared norm of the longest face diagonal as time-step for the involved diffusion step. In order to provide a less biased evaluation, we computed the L_2 errors for 16^2 (plane) and 8^3 (spheres) uniformly sampled points and then report the mean of these errors. Additionally, we compare the results of the introduced polygon Laplacians to a triangle approach. We refine the mesh by triangulating polygons to minimize squared triangle areas [Lie03] and employ the Laplacian based on the intrinsic Delaunay triangulation [BS07], using the geodesic in heat implementation of libigl [JP*18]. The results are presented in Figure 8. Comparing the polygon Laplacians, our method and the two DEC operators with optimized hyper-parameters generally perform best, except in the case of the Voronoi Plane, where the Laplacians obtained with recommended DEC parameters outperform our method. However, these Laplacians had higher errors on other tessellations. The Diamond Laplacian also achieves competitive accuracy, especially for spherical meshes. All polygon approaches are bested by the intrinsic Delaunay approach on planar meshes but remain competitive on spherical tessellations. Comparing our approach to the original linear virtual refinement method, we found that our method consistently improves the accuracy. However, this improvement is more pronounced on planar meshes than on unit spheres.

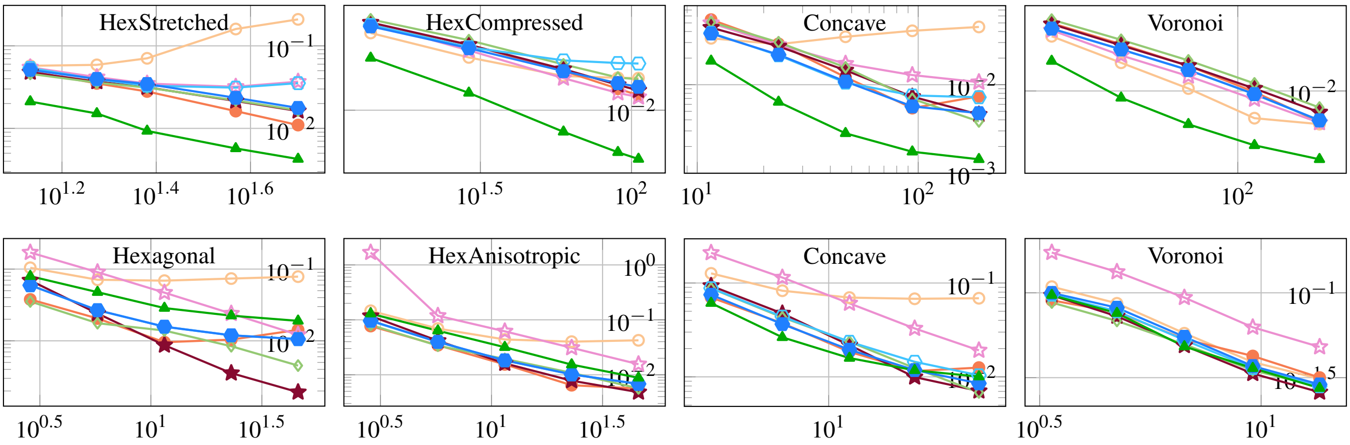


Figure 8: *Geodesics in heat method in log-log scale for planes (top) and spheres (bottom). x-axis: inverse mean edge length, y-axis: L_2 error.*
Legend: [AW11] $\lambda =$ ○ 2, ● 0.5, [dGBD20] $\lambda =$ ☆ 1, ★ 0.1. [BBA21] ♢, [BHKB20] ⬡ default, ⬢ ours, ▲ Intrinsic Delaunay.

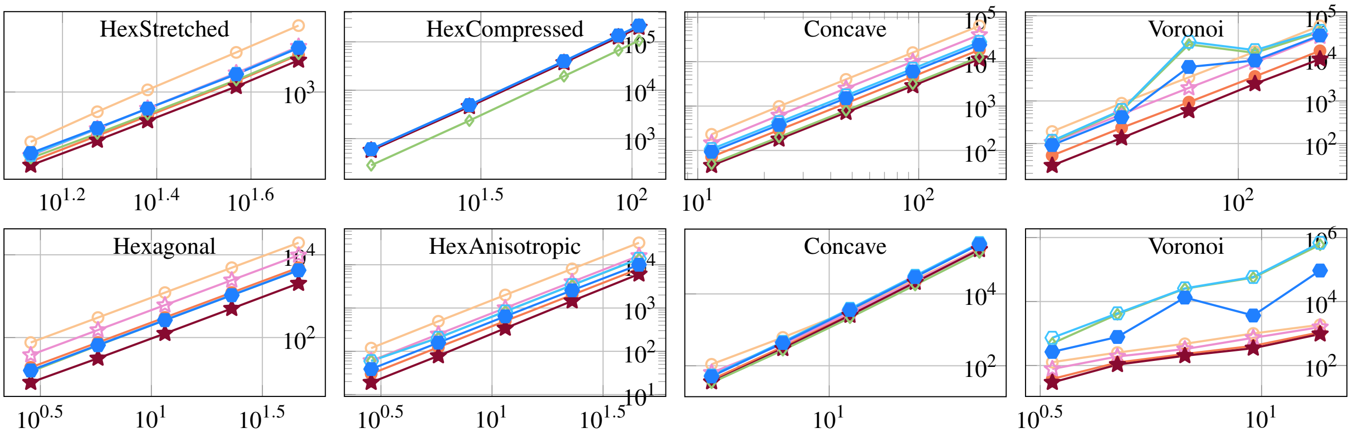


Figure 9: *The condition number $\kappa(\mathbf{S}^{\circ})$ of the respective polygon stiffness matrices on planar (top) and spherical (bottom) polygon meshes. x-axis: inverse mean edge length, y-axis: Condition Number.*
Legend: [AW11] $\lambda =$ ○ 2, ● 0.5, [dGBD20] $\lambda =$ ☆ 1, ★ 0.1. [BBA21] ◇, [BHKB20] ⬡ default, ⬢ ours.

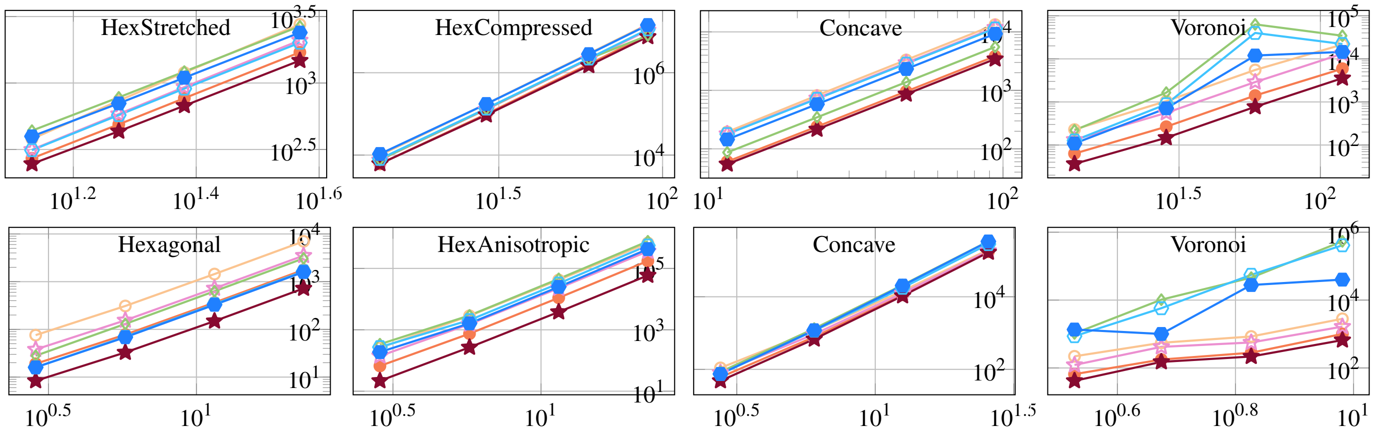


Figure 10: *The condition number $\kappa(\mathbf{L}^{\circ})$ of the respective point-wise polygon Laplacians on planar (top) and spherical (bottom) polygon meshes. x-axis: inverse mean edge length, y-axis: Condition Number.*
Legend: [AW11] $\lambda =$ ○ 2, ● 0.5, [dGBD20] $\lambda =$ ☆ 1, ★ 0.1. [BBA21] ◇, [BHKB20] ⬡ default, ⬢ ours.

Numerical Stability We evaluate the numerical stability of our proposed polygon Laplacian by analyzing the condition number of its involved components, defined in Equation (15). Note that the smallest eigenvalue of the stiffness and Laplacian matrices are zero due to their one-dimensional kernel, so we consider the minimal nonzero eigenvalue. Figure 9 displays the condition numbers of the respective stiffness matrices of the different Laplace operators on the previously presented test meshes. Figure 10 shows the condition numbers of the strong form of the respective Laplacians, which includes the individual mass matrices. Further examples regarding the condition number of our approach and the original linear virtual refinement method can be found in Figure 15. Our approach consistently improves the conditioning of the polygon stiffness matrix compared to the original method. However, consistent with the survey by Bunge et al. [BB23], we observe that the DEC operators yield lower condition numbers for lower hyper-parameter. An improvement in the conditioning of the *strong form* $\left(\mathbf{M}^{\circ}\right)^{-1}\mathbf{S}^{\circ}$ of the Laplacian cannot be guaranteed. While often observed, as shown in Figure 10, our optimization primarily focuses on angle quality. This can lead to smaller virtual triangles, for example, when the trace minimizer avoids acute angles at the virtual vertex by moving it closer to a polygon edge. These smaller triangle areas affect both the smallest eigenvalue of the stiffness matrix $\mathbf{S}^{\circ}$ and the eigenvalues of the mass matrix $\mathbf{M}^{\circ}$, which can lead to potentially increased condition numbers of the combined matrix $\left(\mathbf{M}^{\circ}\right)^{-1}\mathbf{S}^{\circ}$. Since we are using the *lumped* version of the mass matrix, the individual eigenvalues of $\mathbf{M}^{\circ}$ are the vertex areas on the diagonal, i.e., the sum of incident faces' areas.

The conditioning of the stiffness matrix is closely related to the convergence rate of iterative solvers like conjugate gradients [She02], meaning that well-conditioned matrices should cause a faster convergence. We verify this correlation by solving the Poisson equations mentioned above with the conjugate gradient method (without preconditioning) and compare the number of iterations until convergence with the condition numbers of the global stiffness matrices. We observed that when condition numbers improved by 10–50 % with the help of our new method in comparison to Bunge et al. [BHKB20], the number of iterations decreased by 8–15 %.

8.2. Polygon Mesh Smoothing

Figure 12 illustrates a comparison of smoothing results on a mixed polygon mesh, generated with Smart Laplacian and GETMe (both by Vartziotis et al. [VAGW08]), and our method. GETMe is considered a state-of-the-art method in mesh smoothing [Lo14,SBMS23], but the official implementation currently only supports 2D input. Further are the boundary vertices fixed during the optimization. Our approach easily allows for the integration of boundary adherence conditions, such that these vertices are allowed to move as well. For the optimization this additional degree of freedom can further improve the polygon constellations, shown by the condition numbers of our polygon stiffness matrix $\kappa(\mathbf{S}^{\circ})$, which we refer to as κ.

In Figure 13 we compare our methods to simple explicit Laplacian smoothing and the methods of Knupp [Knu00] and Garimella et al. [GS04]. While all of the other methods were also intended for polygons, the energy term for a respective vertex only incorporates its direct one-ring neighbors and neglects the rest of the polygon (see Figure 3). Especially on challenging inputs, featuring non-convex polygons, this often breaks the optimization for these methods. Therefore, the results for Knupp and Garimella had to be generated with a preceding step to repair non-convex faces.

A simpler approach on smoothing could be to apply established triangle smoothing algorithms on the virtual triangulation and unrefine the mesh afterwards. However, common techniques for triangles, such as the tangential smoothing approach by Dunyach et al. [DVBB13], can result in scenarios where vertices slide over sharp edges or high curvature features, as shown in Figure 11 (center). While the feature is preserved in the smoothed triangle mesh, unrefining it (removing the inserted vertices) leaves us with bent non-planar polygons. Increased non-planarity and ill-shaped polygons may also negatively impact the mesh conditioning, as shown with the condition numbers given in the bottom row of Figure 11.

In Figure 15, we present examples of our procedure's input-output pairs, complemented by bar plots that offer comparisons with alternative smoothing strategies. These comparisons encompass both the outcomes of the optimization processes and the condition numbers of the stiffness matrices of the original linear virtual refinement method [BHKB20] and our improved version. Notably, our mesh improvement strategy is meticulously tailored to complement our enhanced Laplace operator, resulting in substantial improvements when applied together.

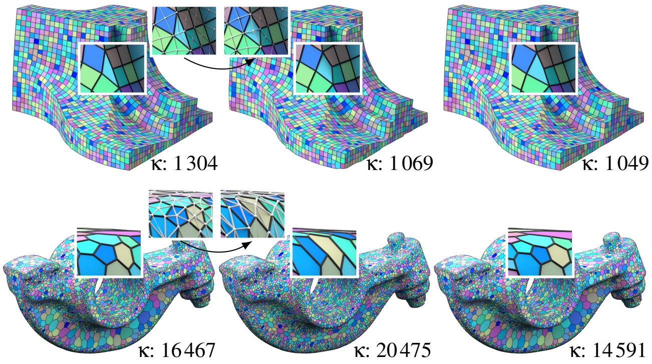


Figure 11: *The polygonal input mesh on the left and our smoothed result on the right. Applying tangential smoothing [DVBB13] on the virtual triangulation may lead to bent polygons (center).*

Level		$\kappa(\mathbf{S}^{\circ})$	$\kappa(\mathbf{L}^{\circ})$	L_2 error	Type
Voronoi 2	a	10 312	12 068	0.002 072 91	Plane
	b	2084	2432	0.002 134 34	
	c	226	358	0.002 019 59	
	a	4284	5776	0.020 494 60	Sphere
	b	782	1006	0.022 109 20	
	c	187	323	0.015 656 10	
Voronoi 3	a	43 091	53 099	0.000 584 99	Plane
	b	13 893	18 576	0.000 580 63	
	c	898	1806	0.000 549 82	
	a	25 877	55 648	0.015 578 80	Sphere
	b	13 372	27 876	0.017 322 00	
	c	317	350	0.007 819 03	

Table 1: *We analyze condition numbers and accuracy on two Voronoi planes and spheres (see Figure 16), respectively. Numbers are given on the unaltered mesh for the (a) virtual refinement method [BHKB20], (b) our new Laplacian and (c) our Laplacian on the optimized tessellation. Our optimization yields improvements in both condition numbers and accuracy.*

In Table 1, we show that in addition to improved condition numbers of the weak and strong form of the polygon Laplacian, the accuracy of Poisson solves on planar and spherical meshes is enhanced on the optimized mesh compared to the original tessellation.

8.3. Timings

With our current implementation being based on the code by Bunge et al. [BB23] we report the following performance: As elaborated in Section 6.3, the optimal position for our proposed virtual vertex is determined via a polygon projection and optimizing for a trace minimizing point. Compared to the original method [BHKB20], which solves for a squared area minimizer and least norm weights, our matrix assembly is on average about 10 % slower. Note however, that assembly costs are negligible (in the range of milliseconds) compared to solving times (several seconds), which stay the same. The time for the smoothing iteration strongly depends on the conditioning and size of the given input mesh. With our single-core implementation (using TinyAD [SBB*22]) the input meshes used in our tests, like Voronoi spheres, the horse, fertility or fandisk terminate with less than 50 iterations in 2 to 10 seconds. For the rocker-arm (Figure 11, bottom), 709 iterations finished in 1400 seconds. However, this can be improved by either adjusting the termination criterion, as the most improvement is achieved within the first few steps, or a dedicated parallelized implementation.

8.4. Limitations

While our method can handle non-planar and non-convex polygons and generally improves the stiffness matrix's accuracy and condition number, there are some limitations. First, the shape of the polygons is limited to those that result in star-shaped faces after the planar projection to avoid negative triangle areas. Challenging configurations are extremely disfigured saddle-shaped polygons that, when projected, result in tangled and self-intersecting shapes. This is, however, more of a theoretical issue as we have yet to find

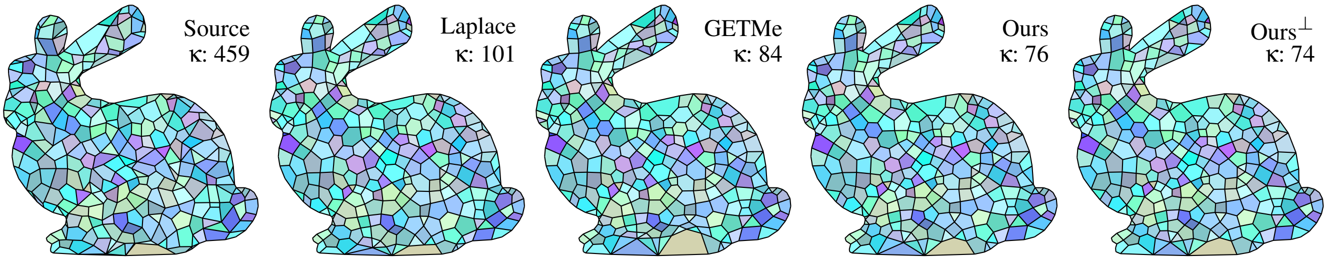


Figure 12: *2D smoothing results on a mixed polygon mesh using Smart Laplace [VAGW08], GETMe [VAGW08], and our method. Further included is a version (⊥) where boundary vertices are also allowed to move along the boundary (not possible with GETMe).*

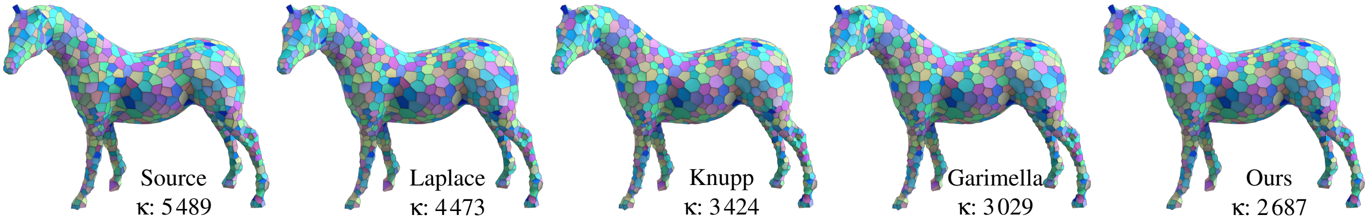


Figure 13: *2D manifold smoothing results on a mixed polygon mesh using Laplace smoothing, Knupp's [Knu00], Garimella's [GS04], and our method; all using surface Quadrics, respectively. GETMe is not included as the official implementation only supports 2D input.*

such shapes in the wild, and the example in Figure 14 is manually constructed. Secondly, negative discrete harmonic coordinates can cause negative mass matrix entries on ill-conditioned meshes. Notably, such occurrences are infrequent; we observed them only on a single face. While relatively large, this particular polygon contained an almost degenerating edge. The negative weight associated with the virtual vertex overpowered the area of the existing node, resulting in a negative entry despite the absence of flipped triangles. The mass matrix, and consequently the strong form of the Laplacian, are then no longer positive/negative (semi-)definite.

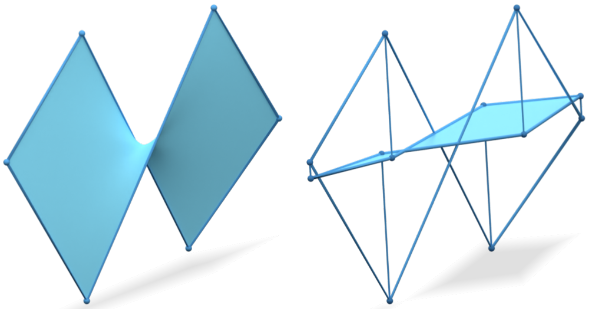


Figure 14: *Strongly disfigured polygons, as the saddle shape on the left, may induce a max-area projection (right) that is self-intersecting, thus non-star-shaped.*

9. Conclusion

This paper presents an approach that optimizes the numerical quality and accuracy of the polygon Laplacian based on the linear virtual refinement method by Bunge et al. [BHKB20]. We show that minimizing the trace of the polygon stiffness matrix establishes a direct link to the geometry of the polygon through the virtual triangle fan. This connection allows us to leverage existing knowledge of the finite element method regarding triangle shapes to find an optimized placement of the virtual vertex. Based on these insights, we present a smoothing algorithm that can further improve polygon meshes with regard to the trace of the polygon stiffness matrix. The combination of techniques offers a valuable tool set for improving the overall performance of numerical simulations on complex surface geometries.

The experiments show consistent improvement in both accuracy and condition numbers for the virtual refinement method. Compared to other existing operators, our methods offers an approach that consistently yields good results, whereas the other methods require tuning the available parameter to adapt to the task at hand.

While our current approach yields promising results on surface meshes, numerous avenues exist for future exploration and improvement. One direction for future research involves extending our findings to volume meshes and examining whether similar connections can be established. Concerning our smoothing approach, a promising addition would be to incorporate constraints into the energy function such that the optimization process preserves specific anisotropic mesh properties. We also aim at extending our analysis to other operators, such as the diamond Laplacian [BBA21] or the Discrete Exterior Calculus (DEC) methods [AW11, dGBD20].

References

[ABB*21] Attene M., Biasotti S., Bertoluzza S., Cabiddu D., Livesu M., Patanè G., Pennacchio M., Prada D., Spagnuolo M.: Benchmarking the geometrical robustness of a virtual element Poisson solver. *Mathematics and Computers in Simulation 190* (2021), 1392–1414. 2

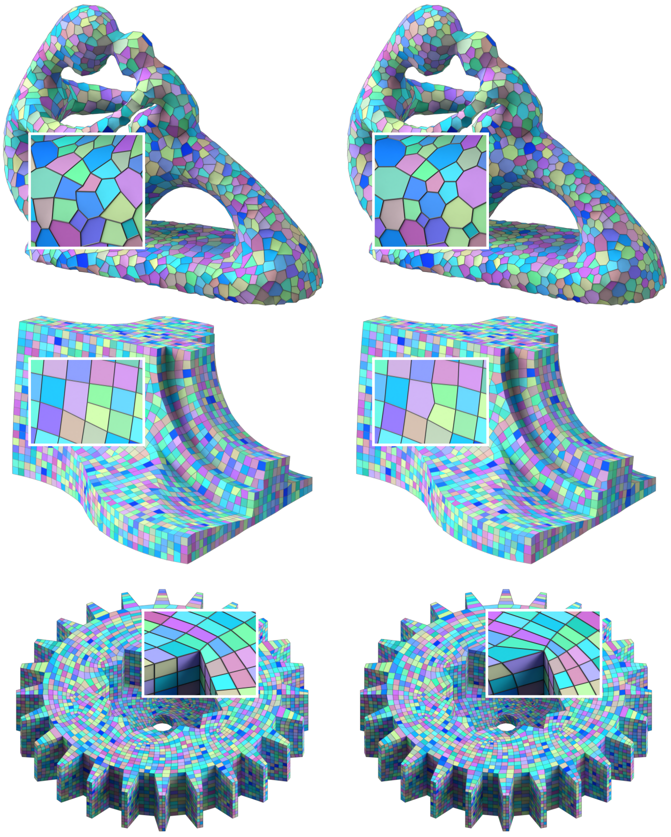
[Ale19] Alexa M.: Harmonic triangulations. *ACM Transactions on Graphics 38*, 4 (2019), 54:1–54:14. 2, 5

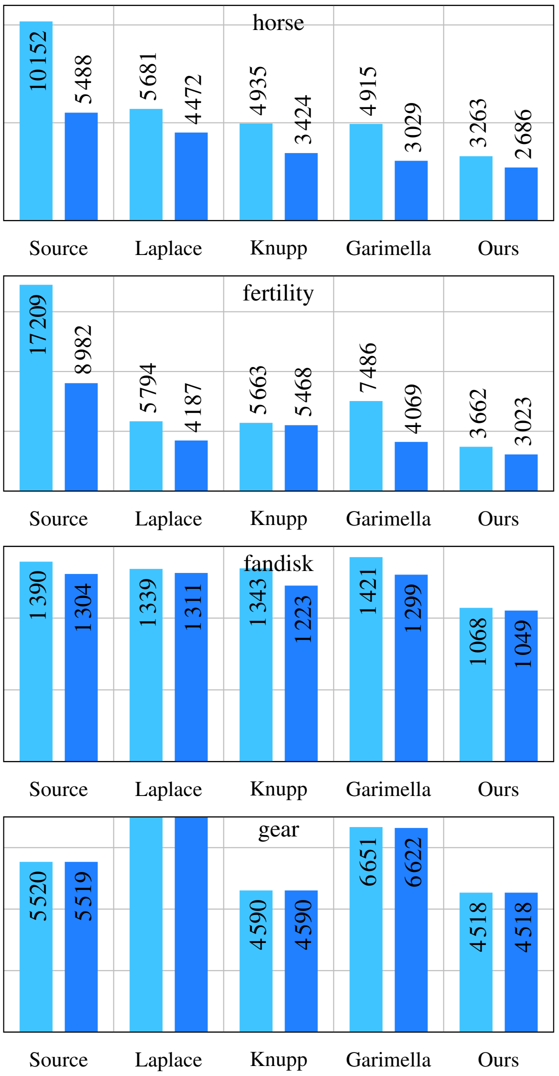
[AW11] Alexa M., Wardetzky M.: Discrete Laplacians on general polygonal meshes. *ACM Transactions on Graphics 30*, 4 (2011), 102:1–102:10. 2, 7, 8, 9, 10, 12

[BB23] Bunge A., Botsch M.: A survey on discrete Laplacians for general polygonal meshes. *Computer Graphics Forum 42*, 2 (2023), 521–544. 2, 8, 10, 11

[BBA21] Bunge A., Botsch M., Alexa M.: The Diamond Laplace for Polygonal and Polyhedral Meshes. *Computer Graphics Forum 40*, 5 (2021), 217–230. 2, 8, 9, 10, 12

Figure 15: *Sources are shown on the left and our results on the right. The horse (Fig. [13]) and fertility are mixed polygon meshes, the fandisk is a quad-dominant mesh and the gear wheel a random pick from the quad-mesh dataset of Pietroni et al. [PNA*21]. The plots on the right show condition numbers* κ *on results of different smoothing strategies for the Laplace operator [BHKB20] in default mode* 🟦 *and with our improvements* 🟦.





[BdVBM13] Beirão da Veiga L., Brezzi F., Marini L. D.: Virtual elements for linear elasticity problems. *SIAM Journal on Numerical Analysis 51*, 2 (2013), 794–812. 2

[BHKB20] Bunge A., Herholz P., Kazhdan M., Botsch M.: Polygon Laplacian made simple. *Computer Graphics Forum 39*, 2 (2020), 303–313. 1, 2, 4, 5, 6, 8, 9, 10, 11, 12, 13

[BKP*10] Botsch M., Kobbelt L., Pauly M., Alliez P., Levy B.: *Polygon Mesh Processing*. AK Peters, 2010. 1

[BLS05] Brezzi F., Lipnikov K., Simoncini V.: A family of mimetic finite difference methods on polygonal and polyhedral meshes. *Mathematical Models and Methods in Applied Sciences 15*, 10 (2005), 1533–1551. 2

[BS07] Bobenko A. I., Springborn B. A.: A discrete Laplace–Beltrami operator for simplicial surfaces. *Discrete & Computational Geometry 38*, 4 (2007), 740–756. 2, 4, 9

[CH11] Coudière Y., Hubert F.: A 3D discrete duality finite volume method for nonlinear elliptic equations. *SIAM Journal on Scientific Computing 33*, 4 (2011), 1739–1764. 2

[Cra23] Crane K.: *Discrete Differential Geometry: An Applied Introduction*. https://www.cs.cmu.edu/~kmcrane/Projects/DDG/paper.pdf, 2023, pp. 168–172. 7

[CWW13] Crane K., Weischedel C., Wardetzky M.: Geodesics in heat: A new approach to computing distance based on heat flow. *ACM Transactions on Graphics 32*, 5 (2013), 152:1–152:11. 9

[CXGL10] Chen R., Xu Y., Gotsman C., Liu L.: A spectral characterization of the Delaunay triangulation. *Computer Aided Geometric Design 27*, 4 (2010), 295–300. 4

[dGBD20] de Goes F., Butts A., Desbrun M.: Discrete differential operators on polygonal meshes. *ACM Transactions on Graphics 39*, 4 (2020), 110:1–110:14. 2, 8, 9, 10, 12

[dGDMD16] de Goes F., Desbrun M., Meyer M., DeRose T.: Subdivision exterior calculus for geometry processing. *ACM Transactions on Graphics 35*, 4 (2016), 133:1–133:11. 9

[DMSB99] Desbrun M., Meyer M., Schröder P., Barr A. H.: Implicit fairing of irregular meshes using diffusion and curvature flow. In *Proceedings of ACM SIGGRAPH* (1999), pp. 317–324. 1, 6

[DVBB13] Dunyach M., Vanderhaeghe D., Barthe L., Botsch M.: Adaptive remeshing for real-time mesh deformation. In *Proceedings of Eurographics Short Papers* (2013). 11

[Dzi88] Dziuk G.: Finite elements for the Beltrami operator on arbitrary surfaces. In *Partial Differential Equations and Calculus of Variations* (1988), pp. 142–155. 1

[EDD*95] Eck M., DeRose T., Duchamp T., Hoppe H., Lounsbery M., Stuetzle W.: Multiresolution analysis of arbitrary meshes. In *Proceedings of ACM SIGGRAPH* (1995), pp. 173–182. 5

[Flo03] Floater M. S.: Mean value coordinates. *Computer Aided Geometric Design 20*, 1 (2003), 19–27. 2

[Fra79] Franke R.: *A critical comparison of some methods for interpolation of scattered data*. Tech. rep., Naval Postgraduate School, 1979. 8

[GH97] Garland M., Heckbert P. S.: Surface simplification using quadric error metrics. In *Proceedings of ACM SIGGRAPH* (1997), pp. 209–216. 7, 8

[GR17] Gillette A., Rand A.: Shape quality for generalized barycentric interpolation. In *Generalized Barycentric Coordinates in Computer Graphics and Computational Mechanics*. CRC Press, 2017, pp. 23–42. 2

[GS04] Garimella R. V., Shashkov M. J.: Polygonal surface mesh optimization. *Engineering with Computers 20* (2004), 265–272. 3, 7, 8, 11, 12

[GSK02] Garimella R. V., Shashkov M. J., Knupp P. M.: Optimization of surface mesh quality using local parametrization. In *Proceedings of International Meshing Roundtable* (2002), pp. 41–52. 3

[GSK04] Garimella R. V., Shashkov M. J., Knupp P. M.: Triangular and quadrilateral surface mesh quality optimization using local parametrization. *Computer Methods in Applied Mechanics and Engineering 193*, 9–11 (2004), 913–928. 3

[Her09] Hermeline F.: A finite volume method for approximating 3D diffusion operators on general meshes. *Journal of Computational Physics 228*, 16 (2009), 5763–5786. 2

[HF06] Hormann K., Floater M. S.: Mean value coordinates for arbitrary planar polygons. *ACM Transactions on Graphics 25*, 4 (2006), 1424–1441. 6

[HS08] Hormann K., Sukumar N.: Maximum entropy coordinates for arbitrary polytopes. *Computer Graphics Forum 27*, 5 (2008), 1513–1520. 2

[HS17] Hormann K., Sukumar N.: *Generalized Barycentric Coordinates in Computer Graphics and Computational Mechanics*. Taylor & Francis, 2017. 2

[Hwa04] Hwang S.-G.: Cauchy's interlace theorem for eigenvalues of hermitian matrices. *The American Mathematical Monthly 111*, 2 (2004), 157–159. 6

[JMD*07] Joshi P., Meyer M., DeRose T., Green B., Sanocki T.: Harmonic coordinates for character articulation. *ACM Transactions on Graphics 26*, 3 (2007), 71–81. 2

[JP*18] Jacobson A., Panozzo D., et al.: libigl: A simple C++ geometry processing library, 2018. https://libigl.github.io/. 9

[JSW05] Ju T., Schaefer S., Warren J.: Mean value coordinates for closed triangular meshes. *ACM Transactions on Graphics 24*, 3 (2005), 561–566. 2

[KMS02] Knupp P., Margolin L. G., Shashkov M.: Reference Jacobian optimization-based rezone strategies for arbitrary Lagrangian Eulerian methods. *Journal of Computational Physics 176*, 1 (2002), 93–128. 3

[Knu00] Knupp P. M.: Achieving finite element mesh quality via optimization of the Jacobian matrix norm and associated quantities. Part I – a framework for surface mesh optimization. *International Journal for Numerical Methods in Engineering 48*, 3 (2000), 401–420. 7, 11, 12

[Lie03] Liepa P.: Filling holes in meshes. In *Proceedings of Symposium on Geometry Processing* (2003), pp. 200–205. 9

[Lip13] Lipnikov K.: On shape-regularity of polyhedral meshes for solving PDEs. In *Proceedings of International Meshing Roundtable* (2013). 2

[Lo14] Lo D. S.: *Finite Element Mesh Generation*. CRC press, 2014. 11

[MDSB03] Meyer M., Desbrun M., Schröder P., Barr A. H.: Discrete differential-geometry operators for triangulated 2-manifolds. In *Visualization and Mathematics III*. Springer-Verlag, 2003, pp. 35–57. 1

[Mus97] Musin O. R.: Properties of the Delaunay triangulation. In *Proceedings of Symposium on Computational Geometry* (1997), pp. 424–426. 2, 4

[MWW15] Mu L., Wang X., Wang Y.: Shape regularity conditions for polygonal/polyhedral meshes, exemplified in a discontinuous Galerkin discretization. *Numerical Methods for Partial Differential Equations 31*, 1 (2015), 308–325. 2

[PNA*21] Pietroni N., Nuvoli S., Alderighi T., Cignoni P., Tarini M.: Reliable feature-line driven quad-remeshing. *ACM Transactions on Graphics 40*, 4 (2021), 155:1–155:17. 13

[PP93] Pinkall U., Polthier K.: Computing discrete minimal surfaces and their conjugates. *Experimental Mathematics 2*, 1 (1993), 15–36. 1, 5

[PPW18] Peng C.-H., Pottmann H., Wonka P.: Designing patterns using triangle-quad hybrid meshes. *ACM Transactions on Graphics 37*, 4 (2018), 107:1–107:14. 1

[PTVF07] Press W. H., Teukolsky S. A., Vetterling W. T., Flannery B. P.: *Numerical Recipes: The Art of Scientific Computing*. Cambridge University Press, 2007. 8

[SBB*22] Schmidt P., Born J., Bommes D., Campen M., Kobbelt L.: TinyAD: Automatic differentiation in geometry processing made simple. *Computer Graphics Forum 41*, 5 (2022), 113–124. 7, 11

[SBMS23] Sorgente T., Biasotti S., Manzini G., Spagnuolo M.: A survey of indicators for mesh quality assessment. *Computer Graphics Forum 42*, 2 (2023), 461–483. 2, 11

[SCV14] Solomon J., Crane K., Vouga E.: Laplace-Beltrami: The swiss army knife of geometry processing. Symposium on Geometry Processing Graduate School, 2014. 1

[She02] Shewchuk J. R.: What is a good linear element? Interpolation, conditioning, and quality measures. In *Proceedings of International Meshing Roundtable* (2002), pp. 115–126. 2, 4, 10

[Suk04] Sukumar N.: Construction of polygonal interpolants: A maximum entropy approach. *International Journal for Numerical Methods in Engineering 61*, 12 (2004), 2159–2181. 2

[VAGW08] Vartziotis D., Athanasiadis T., Goudas I., Wipper J.: Mesh smoothing using the geometric element transformation method. *Computer Methods in Applied Mechanics and Engineering 197*, 45–48 (2008), 3760–3767. 3, 7, 11, 12

[VW12] Vartziotis D., Wipper J.: Fast smoothing of mixed volume meshes based on the effective geometric element transformation method. *Computer methods in applied mechanics and engineering 201–204* (2012), 65–81. 3

[Wac75] Wachspress E. L.: *A Rational Finite Element Basis*. Academic Press, 1975. 2

[War08] Wardetzky M.: Convergence of the cotangent formula: An overview. In *Discrete Differential Geometry*. Birkhäuser Basel, 2008, pp. 275–286. 1

[Wey12] Weyl H.: Das asymptotische Verteilungsgesetz der Eigenwerte linearer partieller Differentialgleichungen (mit einer Anwendung auf die Theorie der Hohlraumstrahlung). *Mathematische Annalen 71* (1912), 441–479. 5

[WMKG07] Wardetzky M., Mathur S., Kälberer F., Grinspun E.: Discrete Laplace operators: No free lunch. In *Proceedings of Symposium on Geometry Processing* (2007), pp. 33–37. 1

[ZS00] Zhou T., Shimada K.: An angle-based approach to two-dimensional mesh smoothing. In *Proceedings of International Meshing Roundtable* (2000), pp. 373–384. 3

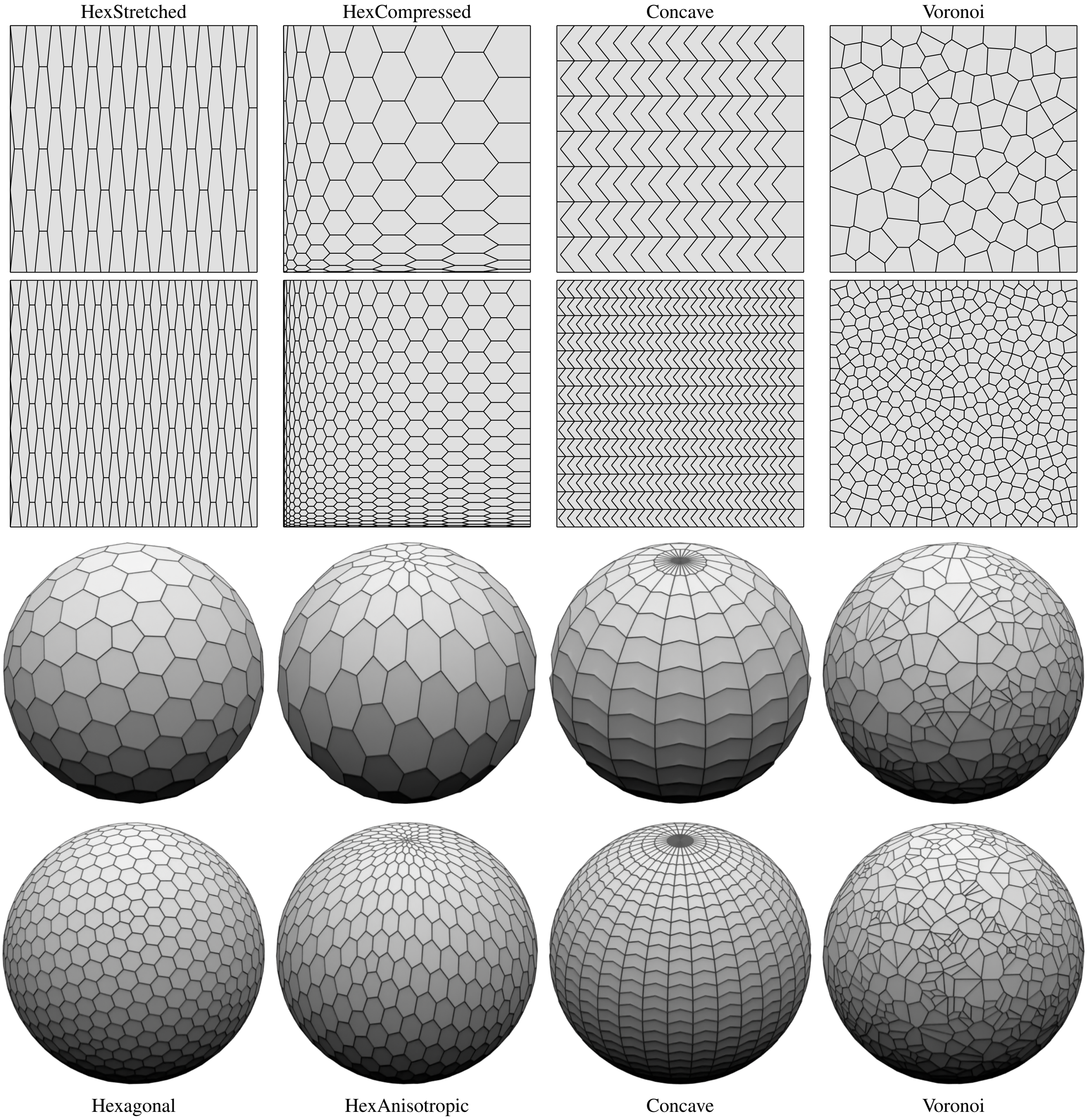


Figure 16: *The different mesh types used in our evaluation (Figures 7, 8 and 9). Shown here are resolutions 2 and 3 of the five levels used.*