DISCRETIZATION OF LAPLACE-BELTRAMI OPERATOR

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

ILGAZ ÇAKAR

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
MATHEMATICS

SEPTEMBER 2022

Approval of the thesis:

DISCRETIZATION OF LAPLACE-BELTRAMI OPERATOR

submitted by **ILGAZ ÇAKAR** in partial fulfillment of the requirements for the degree of **Master of Science in Mathematics Department, Middle East Technical University** by,

Prof. Dr. Halil KALIPÇILAR
Dean, Graduate School of **Natural and Applied Sciences** _____

Prof. Dr. Yıldıray OZAN
Head of Department, **Mathematics** _____

Assoc. Prof. Dr. İbrahim Ünal
Supervisor, **Mathematics, METU** _____

Examining Committee Members:

Prof. Dr. Aydın Gezer
Mathematics, Atatürk University _____

Assoc. Prof. Dr. İbrahim Ünal
Mathematics, METU _____

Prof. Dr. Yıldıray Ozan
Mathematics, METU _____

Date: 01.09.2022

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: Ilgaz Çakar

Signature :

ABSTRACT

DISCRETIZATION OF LAPLACE-BELTRAMI OPERATOR

Çakar, Ilgaz
M.S., Department of Mathematics
Supervisor: Assoc. Prof. Dr. İbrahim Ünal

September 2022, 69 pages

Discrete differential geometry studies the local properties of discrete shapes. Its main purpose is to translate the objects and tools such as curves, surfaces, curvature from smooth category to discrete category so that they can be easily used for computational purposes. One of these tools from smooth category is the Laplace-Beltrami operator whose discrete version is well-known for its applications in geometry processing such as surface smoothing, computing a vector field with prescribed singularities, or mesh parametrization. As the discrete form can be used in computers with more ease, the discretization of the Laplace operator is of utmost importance.

In this thesis, after examining two different approaches to discretize Laplacian on triangular meshes, we show that a discrete Laplacian can not preserve all the properties of its smooth counterpart. Finally, we generalize the discretization to general polygonal meshes which allows much more flexibility on applications.

Keywords: Laplacian, discretization, discrete differential geometry, mesh

ÖZ

LAPLACE-BELTRAMI OPERATÖRÜNÜN AYRIKLAŞTIRILMASI

Çakar, Ilgaz
Yüksek Lisans, Matematik Bölümü
Tez Yöneticisi: Doç. Dr. İbrahim Ünal

Eylül 2022 , 69 sayfa

Ayrık diferansiyel geometri ayrık şekillerin yerel özelliklerini inceler. Ana amacı eğriler, yüzeyler ve eğrilik gibi düzgün kategorideki araç ve nesneleri ayrık kategoriye tercüme etmektir. Ayrık versiyonu yüzey düzleştirme ve verilen bir tekillikteki vektör alanlarının hesaplanması gibi geometri işleme konularındaki uygulamalarıyla bilinen Laplace-Beltrami operatörü de bu araçlardan biridir. Bilgisayarlardaki kullanım kolaylığından dolayı Laplace operatörünün ayrıklaştırılması son derece önemlidir.

Bu tezde Laplace operatörünün üçgensel yüzeylerde iki farklı yöntemle nasıl ayrıklaştırıldığını inceledikten sonra, ayrık Laplace operatörünün düzgün eşdeğerinin tüm özelliklerine sahip olamayacağını gösteriyoruz. Son olarak, bu ayrıklaştırma işlemini uygulamalarda büyük esneklik sağlayan genel çokgensel yüzeylere genelleştireceğiz.

Anahtar Kelimeler: Laplace, ayrıklaştırma, ayrık diferansiyel geometri, ağ

To my loving family

ACKNOWLEDGMENTS

First, I want to express my sincere gratitude to my thesis advisor İbrahim Ünal for his reviews, guidence and fruitful discussions on the topic.

Also, I thank all my instructors Semra Pamuk, Yıldıray Ozan, Özcan Yazıcı, Ömer Küçüksakallı and Sergey Finashin for their great lectures in which I learned a lot about geometry, topology, real analysis, algebra and algebraic topology and I enjoyed my eductaion thanks to them.

I have my parents Fulya and Rahmi Çakar to give my special thanks for their endless support, encoruagement and patience during my education.

I also want to thank to my friends Ali Osman Yıldız, Su Saydam and Burcu Yayla for their support.

Finally, I would like to thank TUBITAK for granting me a full scholarship (2210-A) for my study.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

2D	2 Dimensional
3D	3 Dimensional
DDG	Discrete Differential Geometry
DEC	Discrete Exterior Calculus
FEM	Finite Elements Method
PDE	Partial Differential Equation
SVD	Singular Value Decomposition

CHAPTER 1

INTRODUCTION

The word 'geometry' comes from the Greek word 'geometron', geo- meaning earth and -metron meaning measurement. Geometry arose as the mathematical field dealing with spatial relationships. Geometry was one of the very early fields of premodern mathematics, where the other is the study of numbers, namely arithmetic. Classical geometry focused on compass and straight edge constructions. Introducing the axiomatic method, Euclid made a revolution in the field. His book *The Elements* is the base of what we know as the *Euclidean geometry* today, and is considered the most influential book of all time [2].

In modern times, geometric concepts are generalized to more abstract and complex constructions. They have been exposed to calculus and abstract algebra to give rise to many different mathematical fields which can be barely recognized as the descendants of early geometry (e.g. algebraic geometry, topology, discrete and combinatorial geometry, arithmetic geometry, etc.). In this text, we are going to go through a similar process and set a new destination for modern geometry to computational fields. We are going to discretize an important object from differential geometry, the *Laplace-Beltrami operator*, and see how it is used in geometry processing applications. First, we need to understand what is Discrete Differential Geometry (DDG), and how does it work.

Discrete differential geometry, similar to differential geometry, is all about the local properties of shape. The main difference between differential geometry and discrete differential geometry is that in the discrete case one does not have infinity! So derivatives and infinitesimals are not available in DDG. One uses finite number of angles and lengths to describe the shape. However, this dramatic change in the toolkit does

not lead to loss of information on the object.

Use of less tools leads to an easier understanding of the properties. Also computations via computers gets easier since computers have limited memory to keep a computation going, where the use of infinity or infinitesimals does not help very much. So DDG can be called the modern language for geometric computing. Therefore DDG has a serious effect on modern technology and science in 21st century. The ultimate goal of DDG is to translate the knowledge of smooth differential geometry to DDG.

There is a common method to translate the information:

1. Write down known **equivalent** definitions for an object in the smooth setting.
2. Apply those smooth definitions to a discrete object.
3. Determine which properties are unchanged by resulting **inequivalent** discrete definitions.

Generally no single discrete definition keeps all properties of its smooth counterpart. Indeed, we discuss why a discrete Laplacian does not inherit all properties of the smooth Laplacian in Chapter 4.

Here is the outline of this thesis. We start in Chapter 2 by applying the three-step discretization process to discretize curvature of plane curves. We see that four equivalent curvature definitions lead us to four inequivalent discrete definitions. Then we discretize the curvature of a surface and solve the problem of defining normal vectors on the vertices by using area and volume gradients. In Chapter 3, we focus on discretizing the Laplace-Beltrami operator on triangular meshes. We do it by using Finite Element Method [3, 4], originally based on the work of Brezzi et.al. [5]. Alternative to finite element approach, we use Discrete Exterior Calculus to obtain the same result before. After reviewing the properties of the smooth Laplacian in Chapter 4, we state the discrete counterparts of these properties, namely symmetry, locality, linear precision, positive weights, positive semi-definiteness and convergence. As is shown in the Table 4.1, not all discrete Laplacians satisfies even the first four properties we mentioned before [6]. We prove that no discrete Laplacian can satisfy these four

properties simultaneously unless we work on a regular triangulation, that is a triangulation consisting of only equilateral triangles [7]. Finally, we generalize the idea of Chapter 3 in Chapter 5, and define discrete Laplacians on general polygonal meshes by defining inner products on 0- and 1-forms as matrices [1]. We also generalize the properties of discrete Laplacians using a more general notation. Furthermore, we see that polygons in our mesh are not necessarily planar since planar polygons may violate positive-definiteness. On the other hand, by taking non-planar polygons, we sacrifice linear precision [5, 8]. We give implicit mean curvature flow [9] and a planarizing flow [3, 4] briefly as applications to discrete Laplace operator on general polygonal meshes.

CHAPTER 2

PRELIMINERIES

As a warming up, let us see how this method works by translating the concept of curvature from the smooth setting to the discrete setting. We are going to consider four equivalent definitions of smooth curvature, and obtain four inequivalent definitions for discrete curvature as in [3, 4].

2.1 Discrete Curvature of a Plane Curve

2.1.1 Turning Angle

Curvature of a given curve tells basically how much the curve bends. For a smooth arc-length parametrized curve $\gamma : [0, L] \to \mathbb{R}^2$, the curvature κ can be calculated by the second derivatives. Let γ has a unit tangent $T := \dfrac{d}{ds}\gamma(s)$ and a unit normal N at a point s, then the curvature of γ is the inner product of its unit normal and second derivative, that is first derivative of T, which at the end yields the length of the second derivative of γ since N is a unit vector orthogonal to T.

$$\kappa(s) := \left\langle N, \frac{d}{ds}T \right\rangle = \left| \frac{d}{ds}T \right| = |\gamma''(s)| \tag{2.1}$$

The equality on the middle of the Equation 2.1 makes the curvature loose some information about the curve. Inner product with the unit normal gives the curvature a direction but the length of the second derivative does not have a direction. So one must be aware of the situation, whether the curvature is needed to be a signed quantity or a scalar one.

Figure 2.1: A smooth (continuous) curve and a discrete curve

For the discrete setting, let us use a discrete (polygonal) curve on the plane instead as in the right side of the Figure 2.1. Let $\gamma_1, \ldots, \gamma_n \in \mathbb{R}^2$ be its vertices. Now, one faces with the most elementary problem of DDG: our discrete curve is not differentiable at its vertices, so the standard curvature definition (2.1) does not apply. Thus one need a different characterization to calculate the curvature in a natural way. The curvature is simply the rate at which the tangent turns. Let Ψ be the angle from the horizontal line to the unit tangent T. Then,

$$\int_a^b \kappa \, ds = \Psi(b) - \Psi(a) \quad mod\, 2\pi,$$

where a and b are the points on the curve.

This definition can be applied to the discrete curve in Figure 2.1. The change of the angle along any edge is zero and on the vertices it is the turning angle $\theta_i := \Psi_{i,i+1} - \Psi_{i-1,i}$ which is the angle between the edges $\Psi_{i-1,i}$ and $\Psi_{i,i+1}$. Thus our first notion of discrete curvature is found:

$$\kappa_i^A = \theta_i \in (-\pi, \pi) \tag{2.2}$$

2.1.2 Length Variation

Let us start with a useful fact about curvature from the smooth setting [10]:

The fastest way to decrease (or increase) the length of a curve whose end points are at

a fixed position is to move its points in the normal direction, with speed proportional to its curvature.

Formally, let an *arbitrary* change in the curve γ is given by a function $\eta : [0, L] \to \mathbb{R}^2$ with $\eta(0) = \eta(L) = 0$. Then the amount of displacement of the curve γ is given by the formula

$$\frac{d}{d\epsilon}|_{\epsilon=0}\text{length}(\gamma + \epsilon\eta) = -\int_0^L \langle \eta(s), \kappa(s)N(s)\rangle \, ds$$

Hence the motion that most quickly decreases the length is $\eta = \kappa N$ since the derivative on the left hand side would be zero in this case.

This method is much easier in the discrete case since it is enough to take the gradient of length with respect to vertex positions as vertices are the only variables. At a vertex i, we get:

$$\partial\gamma_i L = \frac{\gamma_i - \gamma_{i-1}}{|\gamma_i - \gamma_{i-1}|} - \frac{\gamma_{i+1} - \gamma_i}{|\gamma_{i+1} - \gamma_i|} := T_{i-1,i} - T_{i,i+1} \tag{2.3}$$

i.e, the difference of unit tangent vectors along two consecutive edges, see Figure 2.2.

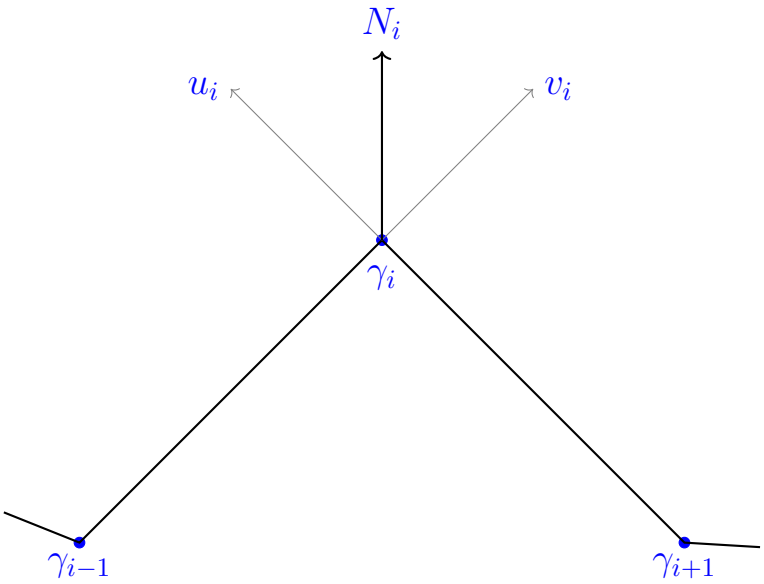


Figure 2.2: Gradient of length for a discrete curve

If N_i is the unit angle bisector at vertex i, this difference can also be written in terms of N_i as

$$\kappa_i^B N_i := 2\sin(\theta_i/2)N_i$$

which yields another definition for discrete curvature

$$\kappa_i^B := 2\sin(\theta_i/2) \tag{2.4}$$

2.1.3 Steiner Formula

A very similar process to length variation is the Steiner's idea of considering how the length of a curve changes if it is displaced by an amount of arbitrarily small constant ϵ in the normal direction [11]. As Steiner observed, the new length of a smooth curve can be expressed as

$$\text{length}(\gamma \pm \epsilon N) = \text{length}(\gamma) \pm \epsilon \int_0^L \kappa(s)ds \tag{2.5}$$

Since this method is applicable for a small piece of the curve, it can be used to determine the curvature for each point on the curve. However one encounters a problem at vertices of a discrete curve, where the unit normal vector N can not be defined. A solution to this situation is to break the curve into individual edges, move each edge along its unit normal vector by the amount of ϵ, and then somehow fill the blank spaces between each consecutive edges.

Different ways of filling the gap are to:
(A) use a circular arc of radius ϵ,
(B) use a line segment between two vertices, and
(C) extend the edges until they meet. (Figure 2.3)

Calculating the lengths of these new curves, one gets:

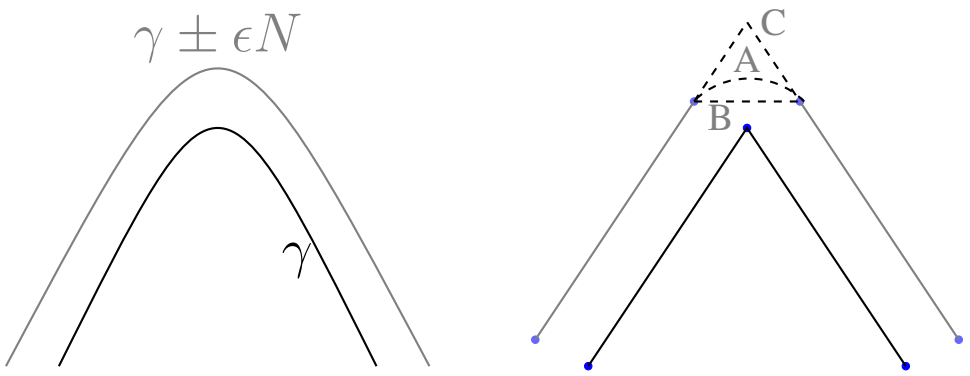


Figure 2.3: Steiner's method used on a smooth and a discrete curve.

$$\begin{aligned} length_A &= length(\gamma) \pm \epsilon \textstyle\sum_{i=2}^{n-1} \theta_i, \\ length_B &= length(\gamma) \pm \epsilon \textstyle\sum_{i=2}^{n-1} 2\sin(\theta_i/2), \\ length_C &= length(\gamma) \pm \epsilon \textstyle\sum_{i=2}^{n-1} 2\tan(\theta_i/2). \end{aligned}$$

So as in the smooth setting, change in the discrete length affects the discrete curvature. One can observe that methods A and B are already obtained in previous sections, see Equation 2.1 and Equation 2.4, respectively. On the other hand the method C provides another definition for the discrete curvature:

$$\kappa_i^C := 2\tan(\theta_i/2) \tag{2.6}$$

2.1.4 Osculating Circle

In the smooth setting, it is well known that the curvature of a curve is equal to the inverse of the radius of its osculating circle. The natural way to define a osculating circle for a discrete curve is to let the circle pass through one vertex and its two neighbor vertices. So one has a triangle and its circumcircle (Figure 2.4), from which it is easy to calculate the radius of the circle using the side lengths of the triangle and *sine* rule. So the discrete curvature obtained from there is

$$\kappa_i^D := 2\sin(\theta_i)/w_i \tag{2.7}$$

where $w_i := |\gamma_{i+1} - \gamma_{i-1}|$.

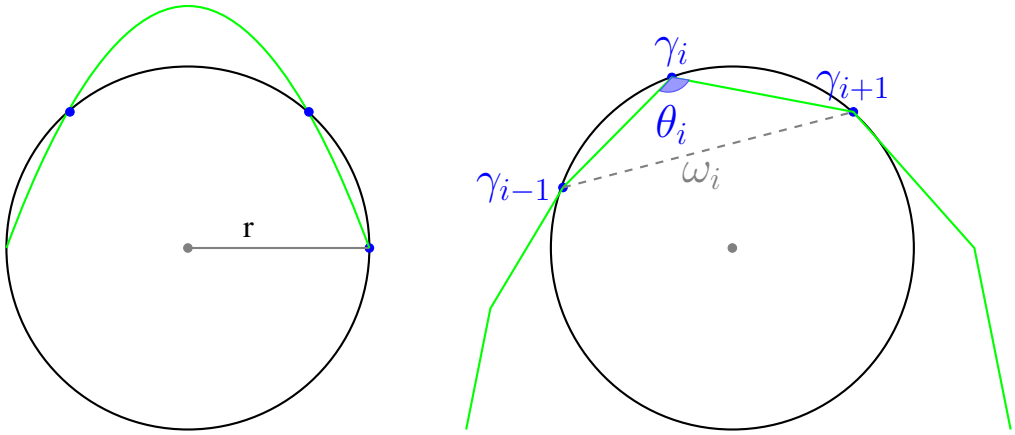


Figure 2.4: Osculating circles of a smooth and a discrete curve.

At the end of the day, equivalent definitions of smooth curvature had led us to inequivalent definitions in the discrete setting. One can observe the inequivalence by just applying these definitions to a single curve. Treated carefully, each discrete curvature converges to the smooth curvature as the discrete curve converges to smooth curve. However neither of the discrete curvatures preserves all properties of the shape as in the smooth case. So there is no unique definition for discrete curvature. Each definition obtained above and others which are not mentioned here must be chosen carefully for the job.

2.2 Curvature of a Discrete Surface

As usual, in order to define curvature we need vectors normal to the surface. Normal direction can be easily defined for smooth surfaces in $\mathbb{R}^3$, it is just the direction of vectors perpendicular to all tangent vectors. It is, however, not that simple for discrete surfaces. If we consider a mesh with planar faces, the normal is easy to find; yet if we have a non-planar polygon, or the normal at a vertex is asked, then some problems arise.

Before considering various ways to define the normal direction for discrete surfaces, some other geometric facts should be established.

2.2.1 Vector Area

In order to evaluate a polygonal area in the plane, one adds an extra point q among the vertices $p_1, \ldots, p_n$ to form triangles with vertices q, p_i, p_{i+1}, and then sum up the areas of those triangles (See Figure 2.5)

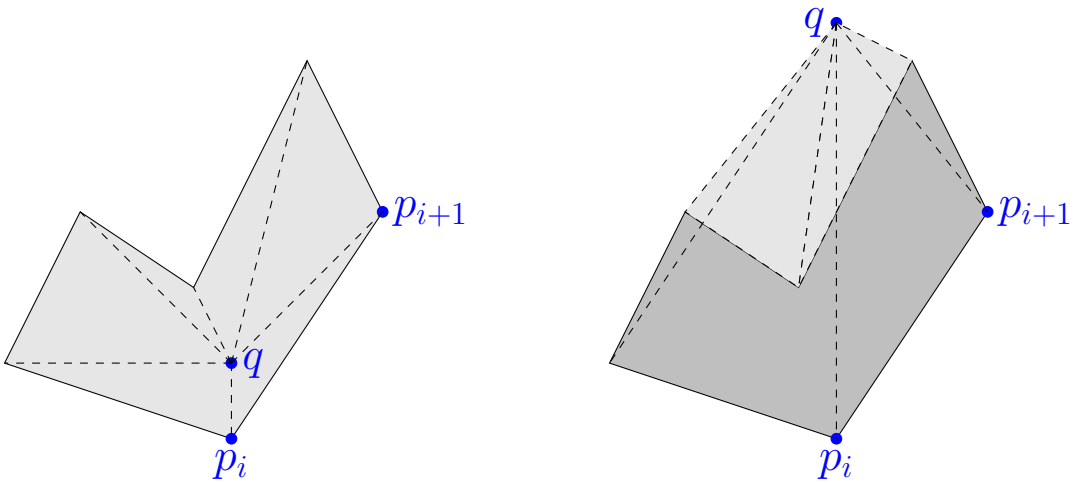


Figure 2.5: Area of a polygon

If q is inside the polygon, the process is simple. Otherwise, one should use the signed areas.

Let A be the area of a planar polygon P, then A can be written as

$$A = \int_P dx \wedge dy.$$

Keeping in mind that $dx \wedge dy = d(x \wedge dy) = -d(y \wedge dx)$, and using the *Stoke's theorem*, one gets

$$A = \frac{1}{2} \int_P d(x \wedge dy) - d(y \wedge dx) = \frac{1}{2} \int_{\partial P} x \wedge dy - y \wedge dx.$$

Now let the polygon P has vertices $p_i = (x_i, y_i)$. So the boundary integral is the sum of the integrals over each edge

$$\int_{\partial P} x \wedge dy - y \wedge dx = \sum_i \int_{e_{ij}} x \wedge dy - y \wedge dx.$$

Since the coordinate functions x and y are linear along each edge e_{ij}, their differentials are constant. So the integral can be written as

$$\begin{aligned}\sum_i \int_{e_{ij}} x \wedge dy - y \wedge dx &= \sum_i \frac{x_i + x_j}{2}(y_j - y_i) - \frac{y_i + y_j}{2}(x_j - x_i) \\ &= \frac{1}{2}\sum_i (p_i + p_j) \times (p_j - p_i) \\ &= \frac{1}{2}\sum_i p_i \times p_j.\end{aligned}$$

Hence the area can be interpreted as

$$A = \frac{1}{2}\sum_i p_i \times p_j.$$

A more general version of this is the *vector area* of a surface patch $f : M \to \mathbb{R}^3$, it is defined as follows

$$N_\nu := \int_M N dA.$$

The vector area depends only on the shape of the boundary ∂M. So, two completely different surfaces may have equal vector areas. From the physical point of view, the reason for this is that the vector area is used for measuring the total flux through the boundary curve.

For a flat region, its normal is constant everywhere, so the vector area is the usual area multiplied by the normal vector. However, in the case of a non-flat surface, situation is different. For instance consider a circular band (Figure 2.6), the vector area is zero since opposite normals cancels each other.

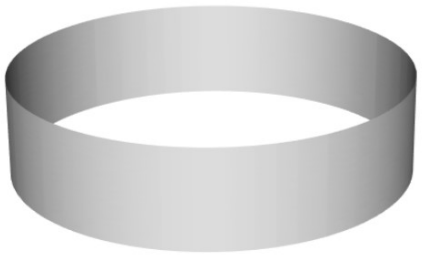


Figure 2.6: Circular Band

2.2.2 Area Gradient

There are necessarily two definitions coming from the smooth structure: the *area* and the *volume gradients*.

The area gradient is used for determining the direction in which when the surface is pushed along, the total area A increases the most, fixing the boundary of the surface to its original position. But this "pushing", in fact, is just a translation, and hence an isometry under which lengths and areas are invariant by its very definition. The only thing to raise the area of a surface is to push the surface along in its normal direction. So, the vertex normal must be defined to be the gradient of area with respect to a given vertex.

The area gradient of the total area A with respect to a point p in a triangle σ is given by

$$\nabla_p A_\sigma = \frac{1}{2} u^\perp,$$

where $u^\perp$ is the normal vector from the edge opposite to p, pointing towards p (see Figure 2.7). So the area gradient of the whole surface is

$$\nabla_p A = \sum_\sigma \nabla_p A_\sigma$$

Since this calculation is done on the vertex p, it only covers the areas of triangles incident to p. So the area gradients can be summed up over a small collection of triangles.

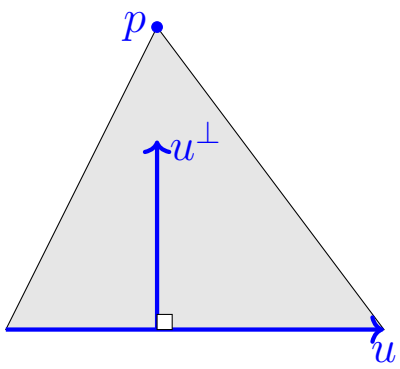


Figure 2.7: A triangle with base vector u, and a normal vector $u^\perp$.

The gradient of surface area with respect to vertex p_i is given by the formula so called the *cotan formula*:

$$\nabla_{p_i} A = \frac{1}{2} \sum_j (\cot \alpha_j + \cot \beta_j)(p_j - p_i), \tag{2.8}$$

where p_j is the j^{th} neighbor of p_i, and α_j and β_j are the angles across from edge (p_i, p_j) shown in the Figure 2.8. For more details see [12].

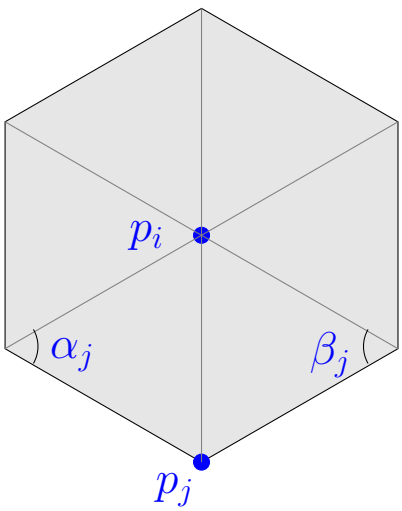


Figure 2.8: Neighbors of vertex p

2.2.2.1 Mean Curvature Vector

The Equation 2.8 for the area gradient is being encountered all over discrete differential geometry. The same formulation shows up in [12] when approaching differently to find normals at vertices by using the *mean curvature vector* HN.

Theorem 1. *For a smooth surface $f : M \to \mathbb{R}^3$, if we denote the mean curvature of the surface by H, and the unit surface normal by N, then the* Laplace-Beltrami *operator, Δ, acting on f is given by*

$$\Delta f = 2HN.$$

Hence, if we are able to discretize the Laplace operator, we would have a way to define vertex normals on a discrete surface. We are going to do this discretization for triangular meshes in Chapter 3, and for general polygonal meshes in Chapter 5. The most natural way of discretizing the Laplacian brings up the *cotan formula*. So the vertex normals we obtained by examining the mean curvature vector are exactly the same ones coming from the area gradient.

If g is the metric, the standard coordinate formula for the Laplace operator is given by Rosenberg in [13] as

$$\Delta\phi = \frac{1}{\sqrt{|g|}} \frac{\partial}{\partial x_i} \left(\sqrt{|g|} g^{ij} \frac{\partial}{\partial x^j} \phi \right).$$

Not only that this expression looks frightening, but also it is not very useful for discretization since g does not have a coordinate representation for a triangle mesh.

So, let us use a tool useful for discretization processes, *conformal parametrization*. Recall that if f is a conformal map, then the lengths on M are scalar multiples of the lengths on $f(M)$, where the scaling factor is e^u. That is, $|df(X)| = e^u|X|$ for some real valued function u on M. Since a conformal parametrization always exists, there is no need for restrictions on the geometry. Conformal coordinates comes in handy when working on the Laplace-Beltrami operator, because Δ can be written as a re-scaling of the standard Laplace operator in the pane.

Theorem 2. *If X and Y denote unit orthogonal directions, the Laplacian of a conformal map ϕ is given by*

$$\Delta\phi = \frac{d(d\phi(X))(X) + d(d\phi(Y))(Y)}{e^{2u}},$$

Recall that the second derivative of a real valued function gives the curvature of its graph. Since Δ contains second derivatives as a sum, it is closely related to the mean curvature, again leading us to Theorem 1.

2.2.3 Volume Gradient

Looking at the volume gradient is another way to find normals. If our surface has a total volume V, then similar to the area case, the fastest method to increase (or decrease) this volume is pushing the surface along its normal direction. In discrete setting, the volume gradient gives a different definition for vertex normals than the area gradient. To express the volume gradient, one needs a 3D versions of the earlier geometric facts.

Firstly, in order to compute the volume of a polyhedron, one needs to add an extra point q and partition the polyhedron into tetrahedra, as the area was partitioned into triangles as in Figure 2.9.

Next, we know that the volume of a tetrahedron is $V = \frac{1}{3}Ah$, where A is the area of the base triangle and h is the altitude from the opposing vertex of the base triangle, say p. Using a similar geometric argument for the triangle case, the volume gradient of a single tetrahedron is found as

$$\nabla_p V = \frac{1}{3}AN,$$

where N is the unit normal of the base triangle with area A pointing towards the point p (see Figure 2.10).

Finally, summing up all the gradients related to vertex p, one gets

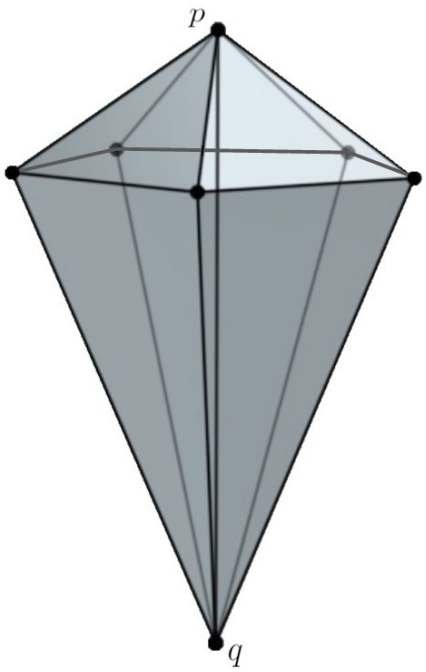


Figure 2.9: Calculation of the volume of a polyhedron using an extra point q.

$$\nabla_p V = \sum_i V_i = \frac{1}{3} \sum_i A_i N_i$$

The surface normals N_i have almost nothing to do with the geometry, yet it appears in the expression in which it increases the complexity. To prevent this, choosing $p = q$, which we already know we can do from the area calculation, eliminates all unnecessary surface normals and leaves only the ones on our original surface.

At the end, volume gradient and vector area N_ν are in the same direction, that is, they are the same up to a constant.

Since now we have a basic intuition about how DDG works, we can jump into the Laplace-Beltrami operator, which needs more elegant approach to discretize compared to the curvature.

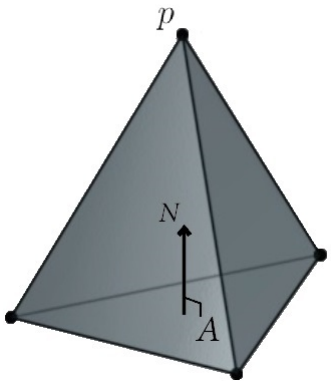


Figure 2.10: Tetrahedron with base area A, and a normal vector pointing towards the vertex p.

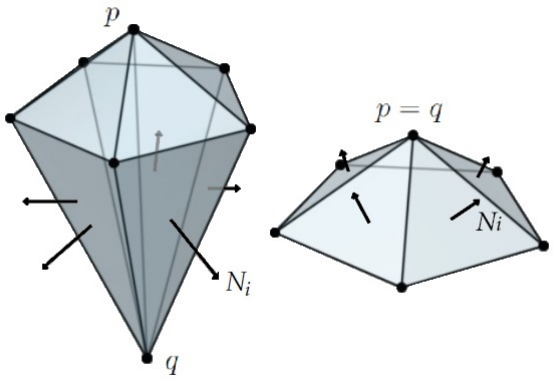


Figure 2.11: Eliminating unnecessary surface normals by simply choosing $p = q$.

CHAPTER 3

DISCRETE LAPLACIANS ON TRIANGULAR MESHES

The Laplace-Beltrami operator (or shortly the Laplacian) is a fundamental tool in mathematics and physics. In this chapter, we will use the Laplacian to discretize Poisson equation for triangulated surfaces. This chapter will be based on the *Chapter 6. The Laplacian* on the lecture series [3, 4]. We will encounter the famous cotan formula while investigating the problem in different ways, one of which is using *test functions* (also known by the name *Galerkin projection*), and the other is by *discrete exterior calculus*.

3.1 Basic Properties of the Laplacian

Firstly, we will examine some basic facts about the smooth Laplacian Δ and a closely related problem, namely the standard *Poisson problem*:

$$\Delta\Phi = \rho$$

Note that the homogeneous Poisson equation is what we know as the *Laplace equation*. Poisson equations can be seen in almost any field, e.g, in physics ρ might represent a mass density in which case the solution Φ, up to suitable constants, would give the corresponding gravitational potential. Or, if ρ describes a charge density, then Φ gives the corresponding electric potential. Moreover, in geometry processing, plenty of applications, such as surface smoothing, and computing a vector field with prescribed singularities or the geodesic distance on a surface, can be done by solving a Poisson equation.

Often we will want to solve Poisson equation on a compact surface M without boundary.

Definition 3.1.1 (Harmonic function)**.** A twice differentiable function $\Phi : M \to \mathbb{R}$ is called *harmonic* if its Laplacian is zero, that is $\Delta\Phi = 0$.

Theorem 3. *In a compact and connected domain without boundary, the only harmonic functions are constant functions.*

Theorem 3 is an important result for the solutions of the Poisson equation since

$$\begin{aligned} \Delta\Phi &= \rho \\ \Rightarrow \Delta(\Phi + c) &= \Delta(\Phi) + \Delta(c) \\ &= \Delta\Phi + 0 \\ &= \Delta\Phi \\ &= \rho \\ \therefore \Delta(\Phi + c) &= \rho. \end{aligned} \tag{3.1}$$

Thus, we have the following theorem:

Theorem 4. *If we have a domain without boundary, which is also compact, we can not obtain constant functions by applying Δ to a function, that is, there is no $c \in \mathbb{R}$ such that $\Delta\Phi = c$.*

This theorem tells whether or not a Poisson equation has a solution. For example, if ρ has a constant component, then the problem is not well-posed. In some cases, getting rid of the constant term may be useful. That is, instead of solving $\Delta\Phi = \rho$, one can solve $\Delta\Phi = \rho - \bar{\rho}$, where

$$\bar{\rho} := \int_M \rho \frac{dV}{|M|}$$

and $|M|$ is the total volume of M.

We need inner products between functions when we work with PDE's like the Poisson equation. The L^2 inner product is one of commonly used ones:

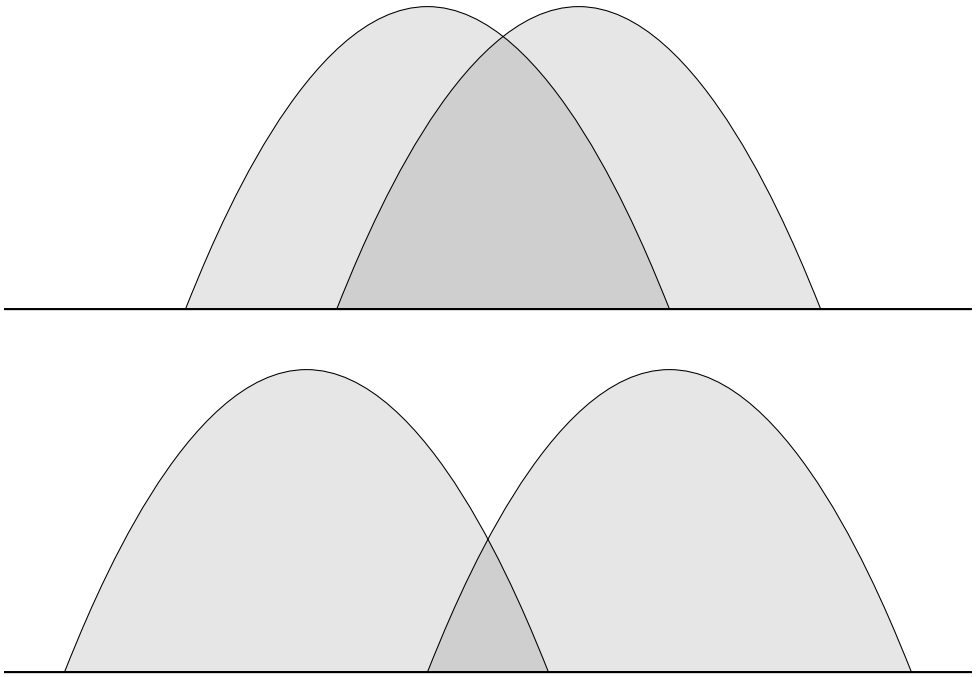


Figure 3.1: L^2 inner product shows how well two functions line up together.

$$\langle f, g \rangle = \int_\Omega f(x)g(x)dx$$

Geometrically, this product corresponds to the dot product in $\mathbb{R}^n$, that is, it measures how well these functions are lined up together. For instance, in the Figure 3.1, the upper functions have larger inner product since they line up well, and conversely the lower ones have smaller inner product.

Similarly, L^2 inner product of two vector fields at the point x is defined as

$$\langle X, Y \rangle = \int_\Omega X(x)Y(x)dx,$$

which, again, measures how well two fields line up at each point. Using this product, we can establish an important result:

Theorem 5 (Green's first identity)**.** *If u and v are enough differentiable functions, N is the outward unit normal, and $\langle -, - \rangle_\partial$ is the inner product on the boundary, then we have*

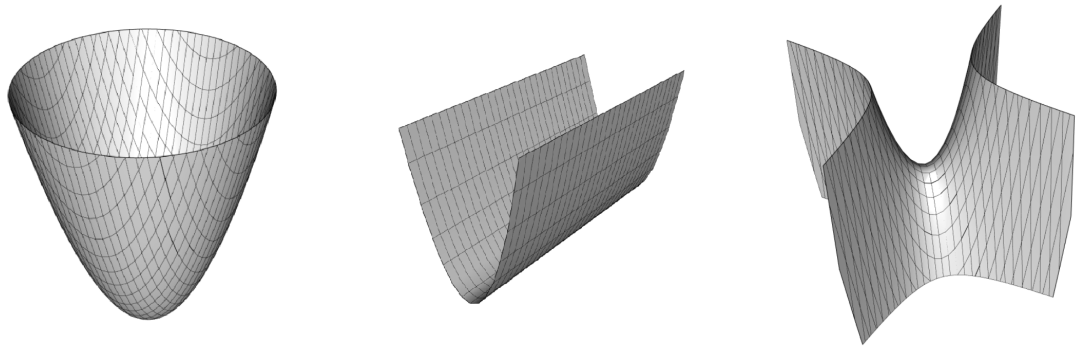


Figure 3.2: Φ depending on its definiteness type. *Left*: $x^2 + y^2$, definite. *Middle*: x^2, semi-definite. *Right*: $x^2 - y^2$, indefinite.

$$\langle \Delta u, v \rangle = - \langle \nabla u, \nabla v \rangle + \langle N \cdot \nabla u, v \rangle_{\partial} .$$

It is worth noting that the Laplacian is *positive semi-definite*, that is $\langle \Delta \Psi, \Psi \rangle \geq 0$ for all functions Ψ.

For example, consider a functions $\Psi(x, y) = ax^2 + bxy + cy^2$ in the plane for some constants $a, b,$ and c. Functions of that form can be given as a matrix:

$$\Psi(x, y) = ax^2 + bxy + cy^2 = \begin{bmatrix} x & y \end{bmatrix} \cdot \begin{bmatrix} a & b/2 \\ b/2 & c \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix}$$

Define $x^T := \begin{bmatrix} x & y \end{bmatrix}$ and $A := \begin{bmatrix} a & b/2 \\ b/2 & c \end{bmatrix}$. So we can define positive semi-definiteness of A. If A is positive semi-definite (i.e., $x^T A x \geq 0$), Ψ looks like a half cylinder, and if A is indefinite ($x^T A x$ is positive or negative depending on x), Ψ looks like a saddle (see Figure 3.2).

3.2 Discretization via Finite Elements Method

The solutions of a physical or geometric problem is often given by a function. However; the set of all functions is too big to be defined and used in a computer which has

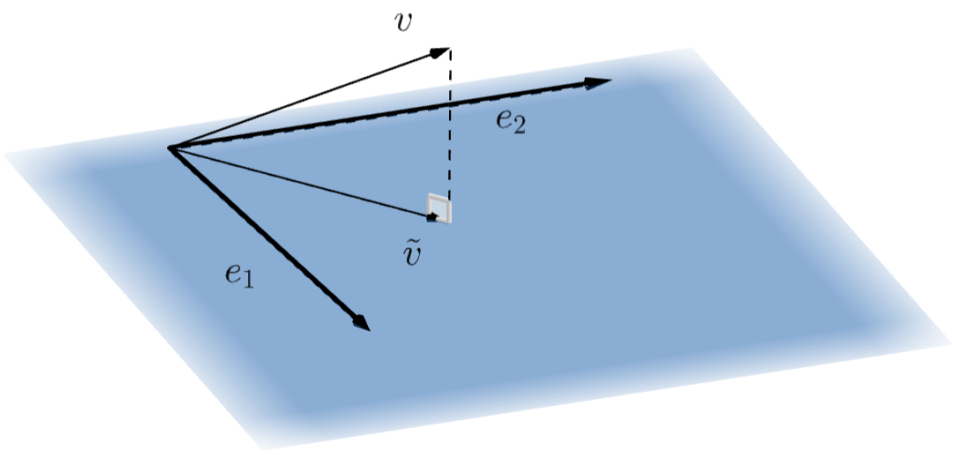


Figure 3.3: Approximating $\tilde{v}$ within a plane.

a finite memory to keep track of all the functions. What *finite element method (FEM)* does is to determine a smaller space of functions and try to find the most suitable solution from that space. Explicitly, if u is the precise solution to a problem and $\{\Phi_i\}$ is some collection of basis functions, then we look for the linear combinations

$$\tilde{u} = \sum_i x_i \Phi_i, \text{ for some } x_i \in \mathbb{R},$$

so that, with respect to some norm, the difference $||\tilde{u} - u||$ takes the smallest possible value.

Start with an easy problem: Let $v \in \mathbb{R}^3$. Find the best approximation $\tilde{v}$ of v in a plane spanned by two basis vectors $e_1, e_2 \in \mathbb{R}^3$ (Figure 3.3).

Since $\tilde{v}$ must be in the plane, the only error that can be tolerated is in the direction of the normal vector of the spanned plane. That is, the difference $(\tilde{v} - v)$ must be perpendicular to vectors $\{e_1, e_2\}$:

$$(\tilde{v} - v) \cdot e_1 = 0$$
$$(\tilde{v} - v) \cdot e_2 = 0$$

So we have a system of two linear equations with two variables, which can be easily solved to find the optimal vector $\tilde{v}$.

Now, let us consider a harder question of solving a standard Poisson problem:

$$\Delta u = f$$

We need to check if a given solution $\tilde{u}$ is the best solution. The basic picture still applies, yet now we are dealing with functions, not with finite dimensional vectors, and also standard inner product is changed to L^2 inner product. Since we do not know what the correct solution u looks like (otherwise we were done), instead of the error $\tilde{u} - u$, we check the quantity $\Delta\tilde{u} - f$, called the *residual*, which measures how well our approximation $\tilde{u}$ satisfies the equation. Our aim is to *test* the residual to see if it becomes zero on every basis direction Φ_j:

$$\langle \Delta\tilde{u} - f, \Phi_j \rangle = \langle \Delta\tilde{u}, \Phi_j \rangle - \langle f, \Phi_j \rangle = 0,$$

which is again a system of linear equations. This way, we make sure that the approximate solution $\tilde{u}$ and the true solution behaves just as the same over a large collection of possible 'measurements'.

Now, let us carry our work on triangulated surfaces. A canonical choice for basis functions are the piece-wise linear hat functions Φ_i, which equals to 1 at their associated vertex, and 0 on any other vertex (see Figure 3.4). So,

$$\Phi_i(v_j) = \delta_{ij}, \tag{3.2}$$

where δ_{ij} is the Kronecker delta which takes the value 1 if $i = j$, and 0 otherwise.

Despite the intuition that second derivative (Δ) of these functions must be zero, Green's identity (Theorem 5) comes in handy in this situation. From now on, we will use u instead of $\tilde{u}$ for the sake of simplicity in notation. Applying Green's identity to the mesh by dividing the integral over triangles σ, we get

$$\begin{aligned}
\langle \Delta u, \Phi_j \rangle &= \sum_k \langle \Delta u, \Phi_j \rangle_{\sigma_k} \\
&= -\sum_k \langle \nabla u, \nabla \Phi_j \rangle_{\sigma_k} + \sum_k \langle N \cdot \nabla u, \Phi_j \rangle_{\partial\sigma_k} .
\end{aligned}$$

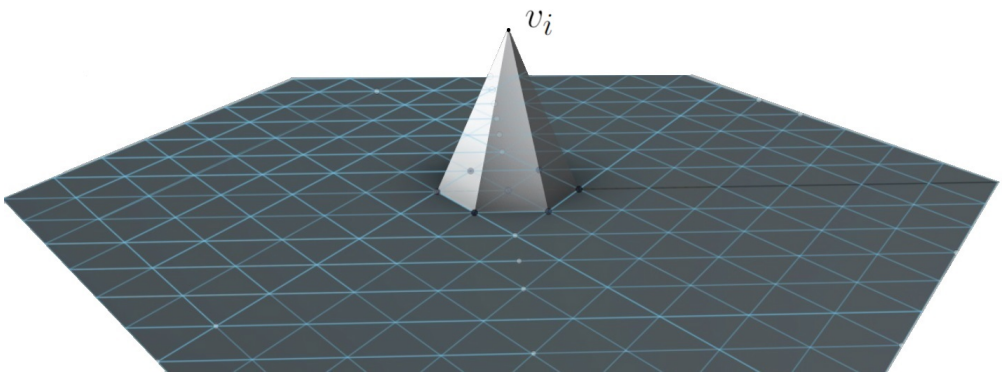


Figure 3.4: Domain of the hat function Φ_i

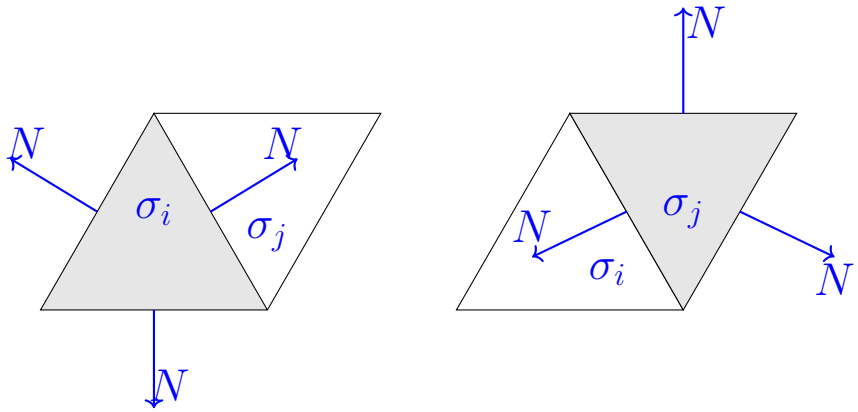


Figure 3.5: Applying Green's identity to a triangle mesh

The normals N of consecutive triangles are oriented in opposing directions. So, if the mesh has no boundary, the boundary integrals cancel each other along shared edges (Figure 3.5).

Thus, what we have now is

$$\langle \nabla u, \nabla \Phi_j \rangle$$

in each triangle σ_k. So, provided that we can compute the gradients of both the candidate solution u, and each basis function Φ_j, we can "test" Δu. On the other hand, u was already a linear combination of the base functions Φ_i, so we have

$$\langle \nabla u, \nabla \Phi_j \rangle = \left\langle \nabla \sum_i x_i \Phi_i, \nabla \Phi_j \right\rangle = \sum_i x_i \langle \nabla \Phi_i, \nabla \Phi_j \rangle .$$

Now our job becomes the computation of the quantity $\langle \nabla \Phi_i, \nabla \Phi_j \rangle$ in each triangle. If we can compute these, then we can simply build the matrix

$$A_{ij} := \langle \nabla \Phi_i, \nabla \Phi_j \rangle ,$$

and solve the linear equation

$$Ax = b$$

for the coefficients x, where the entries on the right-hand side are given by $b_i = \langle f, \Phi_i \rangle$.

In order to keep going further, we need some geometric facts.

Theorem 6. *For an arbitrary triangle (Figure 3.6), we have the following ratio:*

$$\frac{w}{h} = \cot \alpha + \cot \beta.$$

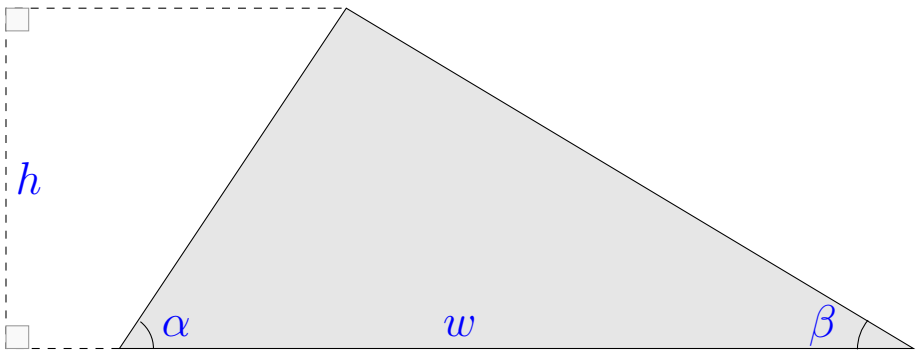


Figure 3.6: A triangle with base length w, height h, and internal angles α and β.

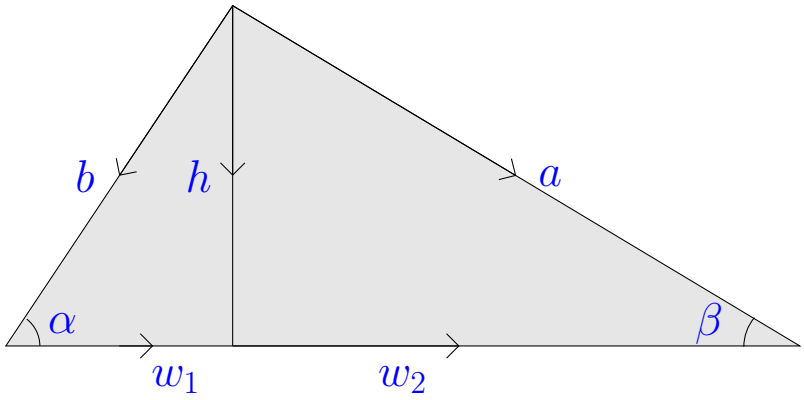
Proof. Consider the figure below, where the base length $w_1 + w_2 = w$ and the edges

are oriented. Let D^{90} denote the 90° rotation in the positive direction. Then we have

$$\begin{aligned} \cot\alpha.h &= -D^{90}w_1 \\ \cot\beta.h &= -D^{90}w_2 \\ a &= h + w_2 \\ b &= h - w_1 \end{aligned}$$

Adding the first two lines together, we get $h.(\cot\alpha + \cot\beta) = -D^{90}(w_1 + w_2) = -D^{90}w$. Since $-D^{90}w = h$, we get our result

$$\frac{w}{h} = \cot\alpha + \cot\beta.$$



□

Theorem 7. *Consider a triangle with the edge vector e along its base. Then the gradient of Φ, the hat function corresponding to the opposite vertex, is given by*

$$\nabla\Phi = \frac{e^\perp}{2A},$$

on the interior of the triangle, where $e^\perp$ is obtained by rotating the vector e by 90° in the counter-clockwise direction, and A is the area of the triangle.

Theorem 8. *For any hat function Φ associated with a given vertex we have*

$$\langle\nabla\Phi, \nabla\Phi\rangle = \frac{1}{2}(\cot\alpha + \cot\beta)$$

within a given triangle, where α and β are the interior angles at the remaining two vertices.

Theorem 9. *Let i and j be distinct vertices of a triangle, and Φ_i and Φ_j be the corresponding hat functions, respectively. Then we have*

$$\langle \nabla \Phi_i, \nabla \Phi_j \rangle = -\frac{1}{2} \cot \theta,$$

where θ is the angle between the opposing edge vectors.

Now, combining these theorems altogether we get a discretization of the Laplacian in terms of famous *cotan formula*:

$$(\Delta u)_i = \frac{1}{2} \sum_j (\cot \alpha_j + \cot \beta_j)(u_j - u_i),$$

where i and j are direct neighbors, and u_i and u_j are the function values of the function u at the vertices i and j, respectively (see Figure 2.8).

3.3 Discretization via Discrete Exterior Calculus

The finite element method is a common method for the discretization of partial differential equations. Now, we will use a different approach, based on discrete exterior calculus. It is worth to mention that although nothing is similar between these two approaches, when finished, we will get the same formulation for the discrete Laplacian.

We will again try solving the standard Poisson equation $\Delta u = f$. It can also be expressed by using the Hodge star as

$$\star d \star du = f$$

To obtain this representation, we first consider a 0-form u specified as a number u_i at each vertex i (see Figure 3.7).

Next, we *integrate* its derivative along each edge in order to compute du:

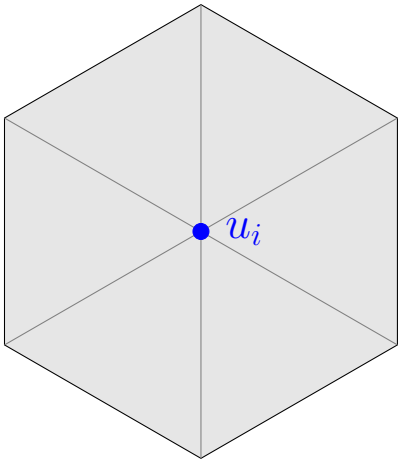


Figure 3.7

$$(du)_{ij} = \int_{e_{ij}} du = \int_{\partial e_{ij}} u = u_j - u_i.$$

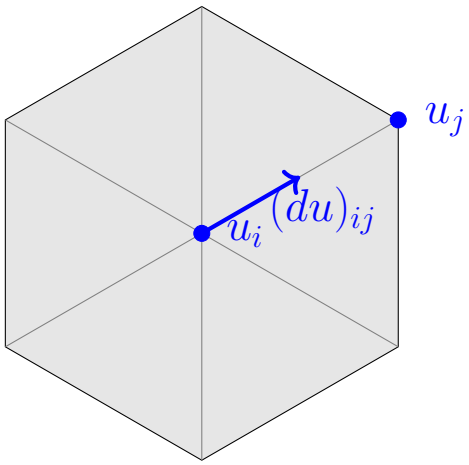


Figure 3.8

Note that the boundary of the edge e_{ij} is just the two end points u_i and u_j. The Hodge star converts a circulation along the edge e_{ij} into the flux through the corresponding dual edge $e^\star_{ij}$. In particular, we take the *total circulation* along the primal edge, divide it by the edge length to get the *average point-wise circulation*, then multiply by the dual edge length to get the *total flux* through the dual edge:

$$(\star du)_{ij} = \frac{|e^\star_{ij}|}{|e_{ij}|}(u_j - u_i). \text{ (see Figure 3.9)}$$

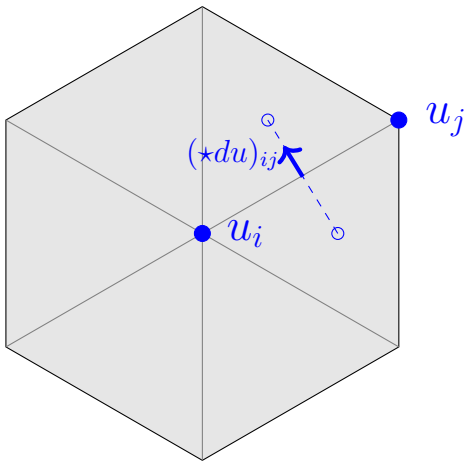


Figure 3.9: $\star du$ is the edge du rotated 90 degrees along the counter-clockwise direction

Now we apply d to $\star du$ and integrate it over C_i:

$$(d \star du)_i = \int_{C_i} d \star du = \int_{\partial C_i} \star du = \sum_j \frac{|e_{ij}^\star|}{|e_{ij}|}(u_j - u_i). \text{ (see Figure 3.10)}$$

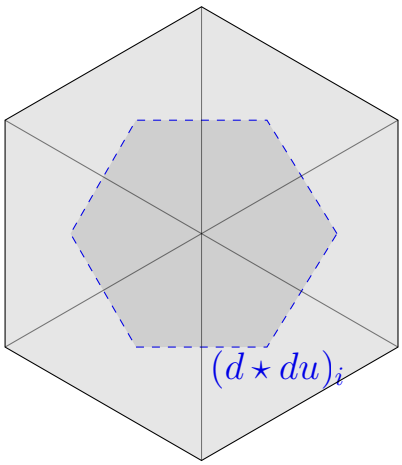


Figure 3.10: $d \star du$

The final Hodge star is used for dividing this quantity by the area of the hexagon C_i to get the average value over the cell. So we get a system of linear equations:

$$(\star d \star du)_i = \frac{1}{|C_i|} \sum_j \frac{|e_{ij}^\star|}{|e_{ij}|}(u_j - u_i) = f_i$$

where f_i is the value of the summation at vertex i. In practice, however, we move the area factor $|C_i|$ to the other side of the equation so that the system

$$(d \star du)_i = \sum_j \frac{|e_{ij}^\star|}{|e_{ij}|}(u_j - u_i) = |C_i| f_i \tag{3.3}$$

we obtained can be represented by a *symmetric* matrix. Symmetric matrices are generally easier to do calculations on and they lead to apter algorithms. Another way to see this transformation is to think as if we discretized the system

$$d \star du = \star f.$$

That is, we converted an equation in terms of 0-forms into an equation in terms of n-forms. When working with surfaces, the operator $d \star d$ is sometimes called *conformal Laplacian* as it does not change when we apply our surface to a conformal transformation. Or equivalently, we can think of $d \star d$ as an operator that gives the value of the Laplacian integrated over each dual cell of the mesh instead of the point-wise value.

Finally, we finish our discretization process with the following theorem.

Theorem 10. *Consider a simplicial surface and suppose we place the vertices of the dual mesh at the circumcenters of the triangles (i.e., the center of the unique circle containing all three vertices) as in the Figure 3.11*

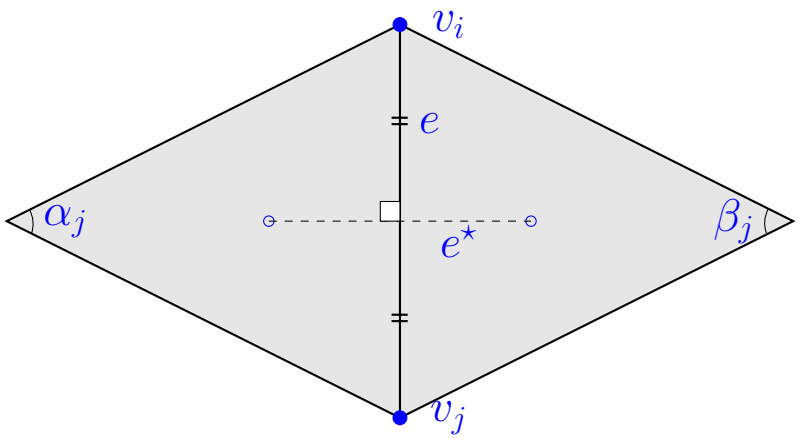


Figure 3.11: $e^\star$ is the dual edge between the two circumcenters

The dual edge $e^\star$ bisects the primal edge e, and the following cotan formula *holds:*

$$\frac{|e^\star_{ij}|}{|e_{ij}|} = \frac{1}{2}(\cot \alpha_j + \cot \beta_j).$$

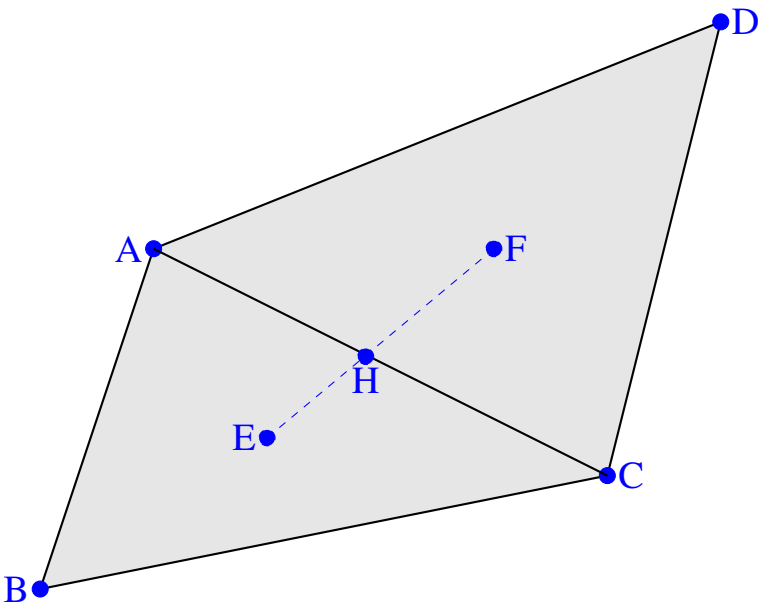


Figure 3.12: E and F are circumcenters of respective triangles. Edge AC corresponds e, and EF corresponds $e^\star$. The interior angles at B and D are α and β, respectively.

Proof. We start with a general setup, where ABC and ACD are triangles with mutual edge AC (Figure 3.12). EF bisects the edge AC since both E and F are intersections of edge bisectors.

If we draw edges AE and AF, forming two triangles AEH and AHF (Figure 3.13), we can see that the interior angles of these triangles at vertices E and F are α and β, respectively because these angles sees the half of the arcs seen by vertices B and D. Moreover, since AE and AF are the radii of the respective circumcircles, their length can be calculated by the *sine rule* as $|AE| = \dfrac{|AC|}{2\sin\alpha}$ and $|AF| = \dfrac{|AC|}{2\sin\beta}$.

Now, we see that

$$\cot\alpha = \frac{|EH|}{|AH|} = \frac{|EH|}{\frac{1}{2}|AC|} = \frac{2|EH|}{|AC|}$$

and

$$\cot\beta = \frac{|HF|}{|AH|} = \frac{|HF|}{\frac{1}{2}|AC|} = \frac{2|HF|}{|AC|}.$$

Therefore,

$$\frac{|EF|}{|AC|} = \frac{1}{2}\left(\cot\alpha + \cot\beta\right).$$

Finally, for the entire triangulation, we have again the same result

$$(\Delta u)_i = \frac{1}{2}\sum_j (\cot\alpha_j + \cot\beta_j)(u_j - u_i).$$

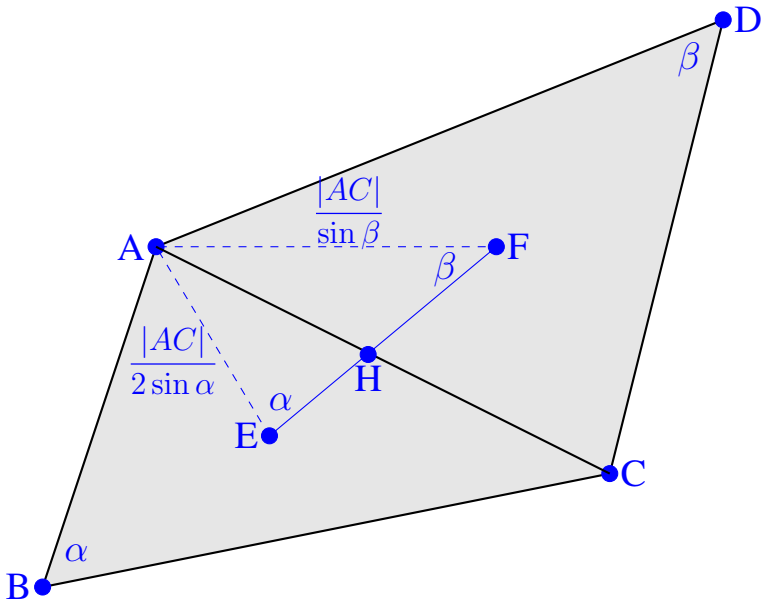


Figure 3.13: Drawing extra edges to form right-angled triangles.

□

Hence, when we put this result in the Equation 3.3, we see that the discrete exterior calculus discretization gives exactly the same result as the Galerkin (i.e., FEM) discretization.

Discrete Laplacians on triangular meshes are used for several geometry processing applications such as parametrization, pose transfer, mesh filtering, re-meshing, compression, and interpolation via barycentric coordinates [14, 15, 16, 17].

CHAPTER 4

NO PERFECT LAPLACIAN

4.1 Introduction

Discrete Laplacians should satisfy some certain properties such as linear precision, symmetry, sparsity, convergence, and positivity. These properties, in fact, properties satisfied by the smooth Laplacian, and we want the discrete counterpart to satisfy them, too. However, we are not that lucky. Discrete Laplacians, in general, do not satisfy all the properties of its smooth version. We will prove this later on this chapter. To do so, we will use an old theorem of Cremona and Maxwell [18, 19]. Finally, we will explain why are there so many different Laplace operators in discrete spaces, and examine their differences.

4.1.1 Properties of smooth Laplacians

Let S be a smooth surface, with or without boundary, equipped with a Riemannian metric. Let us denote the intrinsic L^2 inner product of functions u and v on S as $(u, v)_{L^2} = \int_S uv dA$, and let $\Delta = -div\nabla$ be the intrinsic smooth Laplace-Beltrami operator [13]. This smooth Laplacian has the following properties:

- **(NULL):** If u is a constant function, then $\Delta u = 0$.
- **(SYM) Symmetry:** If u and v are sufficiently smooth functions vanishing on ∂S, then $(\Delta u, v)_{L^2} = (u, \Delta v)_{L^2}$. This also means that the Laplacian is a self-adjoint operator.
- **(LOC) Local support:** For any pair of distinct points p and q, $\Delta u(p)$ does not

depend on $u(q)$. That is, changing the function value at a further point does not affect the action of the Laplacian locally.

- **(LIN) Linear precision:** If S is a part of the Euclidean plane (i.e., S is flat), and u is a linear function on S, then $\Delta u = 0$.

- **(MAX) Maximum principle:** If u is a harmonic function, i.e., $\Delta u = 0$ in the interior, then u does not yield a local maximum or minimum at the interior of S.

- **(PSD) Positive semi-definite:** The *Dirichlet energy*, $E_D(u) = \int_S ||\nabla u||^2 dA$, is non-negative. With the sign choice for Δ before, if u is a smooth enough function identically zero on the boundary of S, then $E_D(u) = (\Delta u, u)_{L^2} \geq 0$.

So, we need at least some of the above properties in applications.

4.2 Discrete Laplacians

Let Γ be a triangular surface mesh with sets V, E and F which contains vertices, edges and faces of the mesh, respectively. A *discrete Laplacian* L on Γ acts linearly on functions u defined for each vertex as

$$(Lu)_i = \sum_j \omega_{ij}(u_i - u_j), \tag{4.1}$$

where i and j are vertex numbers. We have already seen the case where ω_{ij} given in terms of *cotans.* That Laplacian is called *the cotan Laplacian*. It is straightforward to see that (4.1) satisfies (NULL). Moreover, if L were any linear operator acting on function values at vertices, that is $(Lu)_i = \sum_j l_{ij}u_j$, which is zero on constants, then, it satisfies $\sum_j l_{ij} = 0$. Therefore, it can be written in the form of (4.1) with $\omega_{ij} = -l_{ij}$. So, every such linear operator is of the form (4.1). In this case the coefficient matrix $l_{ij} = -\omega_{ij}$ is called the *weight* of the operator L. In the case of L being a discrete Laplacian, we call ω_{ij} as the *Laplacian weight.*

4.2.1 Desired properties for discrete Laplacians

Now, we will give natural properties for discrete Laplacians similar to those from the smooth setting, and whenever possible, we try to give some geometric and physical insight for it.

- **(SYM) Symmetry:** $\omega_{ij} = \omega_{ji}$. Symmetric matrices with real entries have real eigenvalues, and eigenvectors corresponding to these eigenvalues are orthogonal.
- **(LOC) Locality:** The Laplacian weights ω_{ij} becomes zero whenever i and j do not have a common edge connecting them. If there is no common edge between i and j, the action of the Laplacian $(Lu)_i$ will remain invariant under changing the function value of u_j. This is motivated by the fact that smooth Laplacians administer diffusion processes with $u_t = -\Delta u$.
- **(LIN) Linear precision:** If Ω is an embedding of a straight-line into the plane, and u is a linear map on that plane whose domain is V, then $(Lu)_i = 0$ at interior vertices. Equivalently we require that

$$0 = (Lx)_i = \sum_j \omega_{ij}(x_i - x_j) \tag{4.2}$$

  for each interior vertex label i, where $x \in \mathbb{R}^{2|V|}$ gives the vector position of each vertex. In graphics applications, we need (4.2) for de-noising, where we need to remove normal noise only but not to go further [9], parametrization [16], where we need invariance of planar regions under parametrizations, and plate bending energies that should become zero for flat configurations [20].
- **(POS) Positive weights:** $\omega_{ij} \geq 0$ if i and j are distinct vertices. Also, we need at least one vertex j with $\omega_{ij} > 0$ for any vertex i. We need this for several reasons:
  1. (POS) is a sufficient condition for our Laplacian to satisfy a discrete maximum principle.
  2. If we have a diffusion problem such as $u_t = -\Delta u$, (POS) guarantees that flow goes from regions of higher potential to regions of lower potential, not the other way around.

3. (POS) forms a relation to barycentric coordinates by setting

$$\lambda_{ij} = \frac{\omega_{ij}}{\sum_{j \neq i} \omega_{ij}} \qquad \text{so that} \qquad \sum_{j \neq i} \lambda_{ij} = 1.$$

In fact, the function u is discrete harmonic, i.e. $(Lu)_i = 0$ at interior vertices, if and only if u_i is a convex combination of its neighbors, i.e. $u_i = \sum_{j \neq i} \lambda_{ij} u_j$.

4. Having the properties (LOC), (LIN), and (POS) is related to Tutte's embedding theorem for planar graphs [21, 22]: positive weights associated to edges give a straight-line embedding of an abstract planar graph. Moreover, if the boundary vertices are fixed, this embedding is unique, and satisfies (LIN) by its construction.

- **(PSD) Positive semi-definiteness:** L is symmetric positive semi-definite with respect to the standard inner product, and its kernel is one-dimensional.

- **(CON) Convergence:** $L_n \to \Delta$, in the sense that solutions to the discrete Dirichlet problem converge to the solution of the smooth Dirichlet problem under appropriate conditions and in appropriate norms [23]. (CON) is basically required to find approximate solutions of PDEs.

Examples We shortly review some Laplacians used in computer graphics. Purely *combinatorial* Laplacians [15], such as the umbrella operator, where $\omega_{ij} = 1$ if and only if i and j are on the same edge, and the Tutte Laplacian, $\omega_{ij} = 1/d_i$, where d_i is the valence of vertex i (i.e., the number of edges having that vertex), fail to be geometric, that is, they do not satisfy (LIN). Moreover, Floater's *mean value weights* and Wachpress coordinates, mostly used for mesh parametrization [16], violate (SYM) and (CON) simultaneously. The famous *cotan weights* [24] and their variants, generally used for mesh de-noising, violate (POS) on general meshes.

We summarize this situation in the Table 4.1. According to that, none of the Laplacians we mentioned satisfies all desired properties. Moreover, none of them satisfy even the first four properties. Now we will show that this situation is not a coincidence.

Table 4.1: Properties of different discrete Laplacians

	SYM	LOC	LIN	POS	PSD	CON
MEAN VALUE	×	✓	✓	✓	×	×
DELAUNAY	✓	×	✓	✓	✓	?
COMBINATORIAL	✓	✓	×	✓	✓	×
COTAN	✓	✓	✓	×	✓	✓

4.3 No Perfect Laplacian

In this section, we are going to prove the following statement: "Not all meshes admit Laplacians satisfying properties (SYM), (LOC), (LIN), and (POS) simultaneously."

As mentioned before, to prove this statement, we are going to extend a theorem of Maxwell and Cremona on the study of discrete Laplacians and barycentric coordinates in graphics. We will use already-known tools to develop the obstacle preventing the existence of 'perfect' discrete Laplacians satisfying all the properties from the previous section.

We first give a relation between the properties (SYM) + (LOC) + (LIN) and *orthogonal (reciprocal) dual graphs*, based on the Maxwell-Cremona theorem. After that, we prove that orthogonal dual satisfying (POS) correspond to *regular triangulations*. Finally, since not every triangulation is regular, we state that on general meshes, Laplacians can not satisfy (SYM) + (LOC) + (LIN) + (POS) at the same time

4.3.1 Geometric Laplacians and Orthogonal Dual Graphs

Maxwell-Cremona view We can see the weights ω_{ij} as stresses on a planar framework, where positive weight implies pulling stress, and negative weight implies pushing stress. Then, when all boundary vertices fixed, (4.2) gives the Euler-Lagrange equation of the equilibrium state of the framework. The Maxwell-Cremona theorem says that the framework is in equilibrium *if and only if* there is an orthogonal (reciprocal) dual framework.

Orthogonal duals Consider a planar graph Γ, embedded into the plane with non-crossing straight edges. An orthogonal dual is a realization of the dual graph $\Gamma^* = (V^*, E^*, F^*) = (F, E, V)$ in the plane, whose edges are perpendicular to primal edges (see Figure 4.1-*left*).

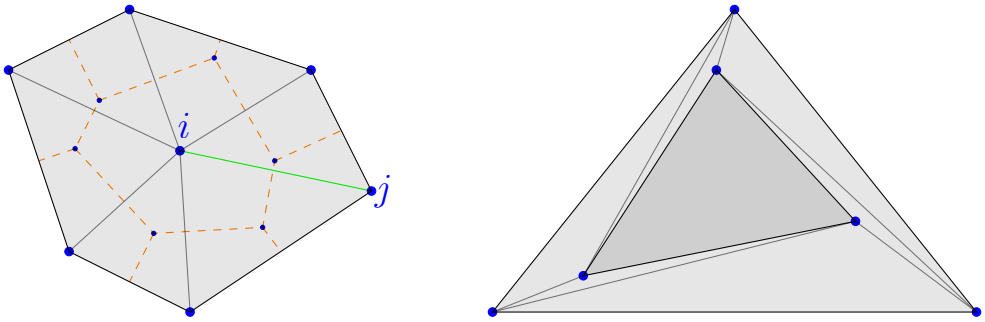


Figure 4.1: Left: Primal graph and orthogonal dual(dashed lines), with edge e_{ij} and its dual colored in green. Right: The projection of the Schönhardt polytope is not regular, so it does not allow for a discrete Laplacian satisfying (SYM) + (LOC) + (LIN) + (POS).

To make a connection between orthogonal duals and our properties, we start with a Laplacian on Γ satisfying (SYM) + (LOC) + (LIN). We define a corresponding *dual* edge to each primal edge e_{ij} of Γ, viewing it as a vector in the plane, by

$$\star e_{ij} = R^{90}(\omega_{ij}, e_{ij}),$$

where R^{90} is the 90 degrees rotation in the plane. Although, in general, dual edges does not have to form a cycle around an interior primal vertex, it does in our situation. Thus, we get a realization of the dual graph in the plane with its edges being *perpendicular* to the dual edges.

Similarly, consider a pair (Γ, Γ^*) of a primal graph and its orthogonal dual, both embedded into the plane with straight edges. Weights per primal edge are obtained by

$$\omega_{ij} := \frac{|\star e_{ij}|}{|e_{ij}|}. \tag{4.3}$$

Here, $|e_{ij}|$ denotes the usual Euclidean length, but $|\star e_{ij}|$ denotes the *signed* Euclidean length of the dual edge. The dual edge $|\star e_{ij}|$ connects the vertices $\star f_1$ and $\star f_2$, corresponding to the primal faces f_1 and f_2. The sign of $|\star e_{ij}|$ is *positive*, if along

the direction of the ray form $\star f_1$ through $\star f_2$, the primal face f_1 comes before f_2. It is *negative* otherwise. With this sign convention, one can check that (4.3) implies (4.2). In fact, this follows from Theorem 10 of Chapter 3. Hence, we obtained a Laplacian with properties (SYM) + (LOC) + (LIN).

Examples Without emphasizing the *equivalence* to (SYM) + (LOC) + (LIN) in planar case, discrete Laplacians came from orthogonal duals on arbitrary (possibly non-planar) triangular meshes were introduced in [25]. A noticeable example of orthogonal duals are the cotan weights [24], which arise from assigning dual vertices to *circumcenters* of primal triangles [26].

4.3.2 Positive Laplacians and regular triangulations

At last, we can present the actual obstacle: Laplacians on a triangulation of the plane satisfies (SYM) + (LOC) + (LIN) + (POS) if and only if that triangulation is regular.

There are several equivalent definitions for *regularity*, but we will use an observation of Aurenhammer [7] for our purposes: a straight-line triangulation of the plane is regular if and only if it allows for a *positive* orthogonal dual, i.e., a dual with positive weights ω_{ij}. However, an arbitrary mesh Γ does not have to be regular. For an example see Figure 4.1-*right*. Therefore, we proved our premise, that is, there is no 'perfect' discrete Laplacian for triangle meshes satisfying (SYM) + (LOC) + (LIN) + (POS).

Our results explains the reason behind the variety of discrete Laplacians used in graphics. Each application using discrete Laplacians must choose which properties are more important for their requirements, and choose the appropriate discrete Laplacian accordingly.

4.3.3 Examples of Imperfect Discrete Laplacians

Let us examine some examples of discrete Laplacians, and consider which requirements they fail to satisfy.

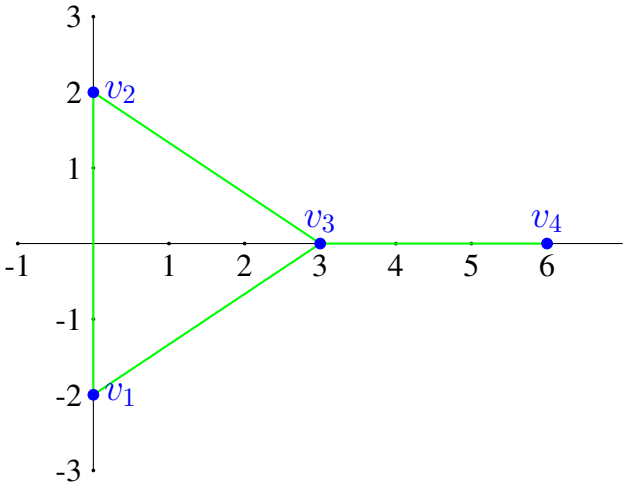


Figure 4.2: A simple graph with 4 vertices

Purely combinatorial Laplacians, such as umbrella operator, $\omega_{ij} = 1$ if and only if i and j share an edge, or Tutte Laplacian, $\omega_{ij} = \frac{1}{d_i}$, where d_i is the number of edges emerging from vertex i (the valence of i), fails to satisfy (LIN) [15]. Recall that (LIN) corresponds to the equality $(L\mathbf{x})_i = \sum_j \omega_{ij}(\mathbf{x}_i - \mathbf{x}_j) = 0$, where $\mathbf{x}$ denotes the vector position of the corresponding vertex. For example, let us consider the umbrella operator whose weight matrix is given as

$$\mathcal{L}(G)_{ij} = 1 \qquad \text{if i and j shares an edge,}$$

and 0 otherwise, where G is a simple graph.

Suppose we have the simple graph G in Figure 4.2. Then the combinatorial Laplacian matrix with respect to the umbrella operator is given by

$$\mathcal{L}_G = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

If we check (LIN) condition, we get

$$\begin{aligned}(Lv)_1 &= 2v_1 - v_2 - v_3\\ (Lv)_2 &= -v_1 + 2v_2 - v_3\\ (Lv)_3 &= -v_1 - v_2 + 3v_3\\ (Lv)_4 &= -v_3 + v_4.\end{aligned}$$

None of these vector positions is 0, thus this combinatorial Laplacian on the given simple graph G is not linearly precise.

Let's consider the discrete Laplacian obtained by using mean value weights [27]. Mean value weights derived from the necessity to generalize barycentric coordinates. It is defined as follows

$$\lambda_{ij} = \frac{\omega_{ij}}{\sum_{n=1}^{k} \omega_n}, \qquad \omega_i = \frac{\tan\frac{\alpha_{i-1}}{2} + \tan\frac{\alpha_i}{2}}{||v_i - v_j||}, \tag{4.4}$$

where the angles and vertices can be seen in Figure 4.3. So, mean value Laplacian is $(Lu)_j = \sum_i \lambda_{ij}(u_j - u_i)$. Notice the change in notation. Usually ω is used for Laplacian weight, but this time λ is used instead. In order this discrete Laplacian to be symmetric, we must have $\lambda_{ij} = \lambda_{ji}$, however we do not always have

$$\tan\frac{\alpha_{i-1}}{2} + \tan\frac{\alpha_i}{2} = \tan\frac{\beta_{i-1}}{2} + \tan\frac{\beta_i}{2}.$$

Therefore, mean value weights does not always give a symmetric discrete Laplace operator.

We are familiar with the cotan Laplacian [24] derived from using cotan weights:

$$(Lu)_i = \frac{1}{2}\sum_j (\cot\alpha_j + \cot\beta_j)(u_j - u_i),$$

where the angles and edges were already given in Figure 3.11 in Chapter 3. Here, the (POS) condition $\omega_{ij} = \cot\alpha_j + \cot\beta_j > 0$ is satisfied only when $\alpha_j + \beta_j > \pi$, which is not always the case. Thus, infamous cotan Laplacian is not always positive definite.

In order to overcome the violation of (POS), Bobenko [28] used the *intrinsic Delaunay triangulation*, yet this time (LOC) became a problem. Even if one redefined locality so that it fits with the intrinsic Delaunay triangulation, a generalized locality

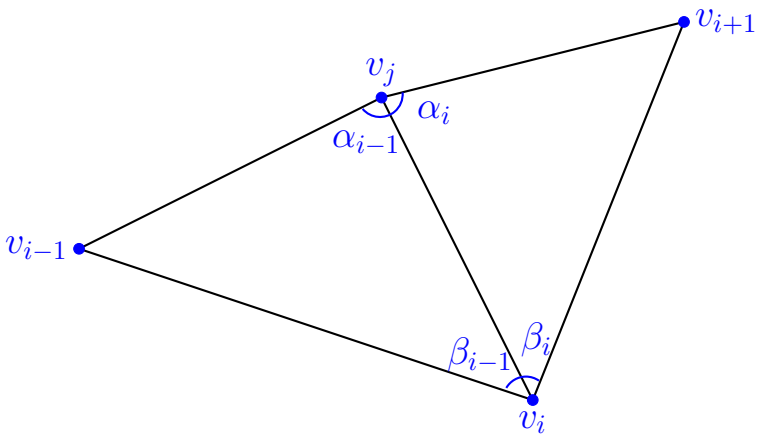


Figure 4.3: A triangular mesh on which mean value weight is defined.

condition would not be satisfied. This generalized locality by Wardetzky [6] suggests the same statement as standard locality not only on *one edge long* neighborhood of a vertex, but on *k-edges long* neighborhood in the mesh for a universal, input independent integer k. On the other hand, Crane and Sharp [29] questions the necessity to use this *combinatorial* notion of locality. Combinatorial locality is to define neighborhoods in terms of the number of edges. They suggest that, from a geometrical point of view, neighborhood relation on the input mesh has nothing special. Like many different atlases can describe the same manifold, various triangulations can define the same mesh. There is no positive effect of using combinatorial locality instead of a geometric one since the goal is to approximate smooth solutions as accurately as possible. They add that changing the mesh combinatorics may result in severe computational costs, for instance we do not even know how to fix intrinsic Laplacian when we move a single vertex, other than rebuilding the intrinsic Delaunay triangulation from scratch.

CHAPTER 5

DISCRETE LAPLACIANS ON GENERAL POLYGONAL MESHES

5.1 Motivation

Triangle is the simplest polygon on a plane. Thus, as Plato suggested in his *Timaeus* as "every solid must necessarily be contained in planes; and every planar rectilinear figure is composed of triangles", triangles span every planar shape. Moreover, algebraic topology uses triangular meshes, i.e., abstract simplicial complexes, to represent and work on surfaces. Although the simplicity of using triangles is tempting, it sure restricts artistic freedom. Design and architecture would be so shallow and too simple if there were no quadrilaterals, pentagons, and any other polygonal shapes.

A fairly large amount of geometry processing tools rely on *discrete Laplace operators* whose most common representative is being the cotan operator. Discrete Laplace operators are mostly used for mesh parametrization, shape analysis, de-noising, manipulation, compression, and physical simulation. In this thesis, we will develop the theory behind discrete Laplace operators on a general polygonal mesh so that we can extend the applications of Laplacian to a bigger scale. While doing so, we try to maintain some core properties of smooth Laplacian. Our approach needs some theoretical development, but, at the end, implementation will be surprisingly easy. In principal, tools we will develop extend the applications of the Laplacian from the triangular setting to the general polygonal setting. For instance, physical simulation and geometry processing have their roots on cotan operator defined for triangular meshes, and they will be extended on general polygonal meshes.

5.2 Discrete Laplacian Framework

In this text, we work with an *oriented* 2-manifold mesh M with or without boundary, whose vertices, edges and faces are stored in the sets V, E and F, respectively. Faces of our surface will be *simple*, but not necessarily planar, polygons in $\mathbb{R}^3$. A *simple* face means that each polygon is a closed, non-self-intersecting loop of edges. *Oriented* means that all faces have a consistent orientation so that any two adjacent face would have opposite orientations on their shared edge. Since some edges will have two opposite orientations, we will use oriented half-edges in order to distinguish between them. Moreover, we separate inner and boundary edges with the sets E_I and E_B, respectively.

Approaching discrete Laplacians algebraically We will approach discrete Laplacians algebraically as in [1]. So we will have utilities of a unified treatment and a simple implementation for Laplace operators defined on polygonal meshes whose faces are of arbitrary degree.

In the following construction, we will use standard notation in analogy to smooth setting. Let Ω^k be the linear space of discrete $k-$forms on M. In the discrete case, we associate 0-forms with vertices and 1-forms with half-edges. Any 1-form β is required to satisfy $\beta(e_{pq}) = -\beta(e_{qp})$, where e_{pq} is an oriented half-edge from the vertex p to q. Construction of discrete Laplacians on 0-forms (functions) mainly depends on the construction from smooth setting as in [13].

We know that in $\mathbb{R}^n$, the smooth Laplacian is defined as

$$\Delta = -\left(\frac{\partial^2}{\partial(x^1)^2} + \cdots + \frac{\partial^2}{\partial(x^n)^2}\right) = -div\nabla,$$

where div is the divergence operator and ∇ is the gradient. The gradient ∇ in local coordinates is given by $\nabla f = g^{ij}\partial_i f \partial_j$, where $\partial_i = \partial_{x^i} = \frac{\partial}{\partial x^i}$, and g^{ij} is a Riemannian metric. Here, and from now on whenever it is convenient, we will use Einstein's summation convention, that is, $a_i x^i := \sum_{i=0}^{n} a_i x^i$.

As for divergence, when $f \in C_c^\infty(\mathbb{R}^n)$, an infinitely many differentiable function on

$\mathbb{R}^n$ with compact support, is being acted on by integration by parts, it gives

$$-\int_{\mathbb{R}^n} \partial_i X^i \cdot f = \int_{\mathbb{R}^n} \partial_i f \cdot X^i$$

for functions X^i such that $X = X^i \partial_i$ on $\mathbb{R}^n$, the divergence is characterized by the equality of the following inner products

$$\langle -divX, f \rangle = \langle X, \nabla f \rangle , \tag{5.1}$$

where $\langle -, - \rangle$ is the global inner product on functions and vector fields induced by the standard dot product. Thus, $-div$ is the (formal) adjoint to ∇.

Now, assume that an operator $divX$ satisfying (5.1) exists. Let U be a coordinate patch on M. For any function $f \in C_c^\infty(U)$ and vector field $X = X^i \partial_i \in TM$, we have

$$\begin{aligned}
\langle X, \nabla f \rangle &= \int_M \langle X, \nabla f \rangle \, vol \\
&= \int_U \left\langle X^i \partial_i, g^{kj} \partial_k f \partial_j \right\rangle vol \\
&= \int_U X^i (\partial_k f) g^{ki} g_{ij} \sqrt{\det g} dx^1 \dots dx^n \\
&= \int_U X^i (\partial_i f) \sqrt{\det g} dx^1 \dots dx^n \\
&= -\int_U \frac{1}{\sqrt{\det g}} f \cdot \partial_i \left(X^i \sqrt{\det g} \right) \sqrt{\det g} dx^1 \dots dx^n \\
&= \left\langle f, -\frac{1}{\sqrt{\det g}} \partial_i \left(X^i \sqrt{\det g} \right) \right\rangle .
\end{aligned}$$

Hence, if $divX$ exists, it must satisfy

$$divX = \frac{1}{\sqrt{\det g}} \partial_i \left(X^i \sqrt{\det g} \right) .$$

Under the assumption of this expression being coordinate-independent, we can define

$\Delta = -div\nabla$. In local coordinates, we get

$$\begin{aligned}\Delta f &= -\frac{1}{\sqrt{\det g}}\partial_j\left(g^{ij}\sqrt{\det g}\partial_i f\right)\\ &= -g^{ij}\partial_j\partial_i f + \text{ (lower order terms)}.\end{aligned}$$

So, Δ is determined by the Riemannian metric g.

Before proceeding further, recall that a finite dimensional vector space V with an inner product $\langle -, -\rangle$ is naturally isomorphic to the dual vector space V^* under the map $\alpha : V \to V^*$, where $\alpha(v) = v^*$ satisfies $v^*(w) = \langle v, w\rangle$ for $v, w \in V$. We will also use α to denote the bundle isomorphism $\alpha : TM \to T^*M$.

Now, we rewrite (5.1) in terms of 1-forms. Since α is trivially an isometry, it is easy to check that at each point of M, we have $g(\alpha(X), df) = g(X, \nabla f)$ for any tangent vector X and any function f. Set $d^* : \Omega^1(T^*M) \to C^\infty(M)$ by $d^*(w) = -div(\alpha^{-1}(w))$, so that d^* is "the same" as div up to the isomorphism $\alpha : TM \to T^*M$. Then d^* is characterized by

$$\langle d^*w, f\rangle = \langle w, df\rangle,$$

for all $w \in C_c^\infty(\Omega^1)$, $f \in C_c^\infty$. Now we have

$$d^*(w) = -\frac{1}{\sqrt{\det g}}\ \partial_i\left(g^{ij}\sqrt{\det g}\ w\right),$$

where $w = w_i\, dx^i$, and this expression is independent of the choice of the local coordinates.

Finally, we can define the Laplacian as

$$\Delta = d^*d$$

This is the same as the previous one since

$$d^*df = (-div\alpha^{-1})(\alpha\nabla f) = -div\nabla f.$$

Here, for our purposes, we will see $d : \Omega^0 \to \Omega^1$ as Cartan's exterior derivative, and d^* as its (formal) adjoint with respect to the L^2 inner products induced on Ω^0 and Ω^1

by the Riemannian metric g. Independence of d from the choice of a Riemannian metric is worth noting down.

In order to copy the smooth setting, we define the co-boundary operator $d : \Omega^0 \to \Omega^1$ to be

$$(du)(e_{pq}) = u(q) - u(p). \tag{5.2}$$

In order to construct the adjoint operator d^*, we need inner product $\langle -, - \rangle_k$ on the linear space of k-forms. Now we will build such inner products. Given a fixed choice of inner products on discrete k-forms, the coboundary operator is defined by requiring that

$$\langle d\alpha, \beta \rangle_{k+1} = \langle \alpha, d^*\beta \rangle_k \tag{5.3}$$

for all k-forms α and $(k+1)$-forms β. Thus, *strongly defined* discrete Laplacian acting on k-forms is given by

$$\mathbb{L} := dd^* + d^*d. \tag{5.4}$$

This is the point of view of *discrete exterior calculus* [4, 26], where, with an abuse of notation, inner products are referred to as "discrete Hodge stars". Strong Laplacians $\mathbb{L}$ are self-adjoint with respect to inner products $\langle -, - \rangle_k$ on discrete k-forms since

$$\langle \mathbb{L}\alpha, \beta \rangle_k = \langle d\alpha, d\beta \rangle_{k+1} + \langle d^*\alpha, d^*\beta \rangle_{k-1} = \langle \alpha, \mathbb{L}\beta \rangle_k .$$

Moreover, $\mathbb{L}$ is always positive semi-definite since

$$\langle \mathbb{L}\alpha, \alpha \rangle_k = \langle d\alpha, d\alpha \rangle_{k+1} + \langle d^*\alpha, d^*\alpha \rangle_{k-1} \geq 0.$$

Also, a discrete k-form is harmonic, i.e., $\mathbb{L}u = 0$, if and only if $d\alpha = d^*\alpha = 0$, as in the smooth setting.

Every strongly defined Laplacian, as in the Equation 5.4, has a weakly defined counterpart L which acts on discrete functions u. The weak Laplacian is defined on each vertex i as follows

$$(Lu)_i := \langle \mathbb{L}u, \Phi_i \rangle_0 = \langle d^*du, \Phi_i \rangle_0 = \langle du, d\Phi_i \rangle_1 , \tag{5.5}$$

where Φ_i is the linear hat function we defined on Section 3.2. From the Equation 5.5 we get two identities:

$$(Lu)_i = \langle du, d1_i \rangle_1 \tag{5.6}$$

$$(Lu)_i = \langle \mathbb{L}u, 1_i \rangle_0 . \tag{5.7}$$

Now, let M_0 be a $|V| \times |V|$ matrix which is symmetric, positive definite, and M_1 be a $(2|E_I| + |E_B|) \times (2|E_I| + |E_B|)$ matrix which is symmetric, positive definite representing inner products on 0-forms and 1-forms, respectively:

$$\langle u, v \rangle_0 = u^T M_0 v \tag{5.8}$$

$$\langle \alpha, \beta \rangle_1 = \alpha^T M_1 \beta \tag{5.9}$$

Now, using Equation 5.9, Equation 5.6 becomes

$$\begin{aligned} (Lu)_i &= \langle du, d1_i \rangle_1 \\ &= (du)^T M_1 d1_i \\ \therefore L &= d^T M_1 d \end{aligned} \tag{5.10}$$

Similarly, using Equation 5.8, Equation 5.7 becomes

$$\begin{aligned} (Lu)_i &= \langle \mathbb{L}u, 1_i \rangle_0 \\ &= (\mathbb{L}u)^T M_0 (1_i)^T = (\mathbb{L}u)^T M_0 \\ \therefore L &= \mathbb{L}^T M_0 \Rightarrow \mathbb{L}^T = L M_0^{-1} \\ &\Rightarrow \mathbb{L} = (M_0^{-1})^T L^T = M_0^{-1} L \\ \therefore \mathbb{L} &= M_0^{-1} L \end{aligned} \tag{5.11}$$

since both M_0 and L are symmetric.

To sum up this construction, we have

$$\mathbb{L} = M_0^{-1} L \qquad \text{with} \qquad L = d^T M_1 d. \tag{5.12}$$

A well-known example of such inner products for meshes with triangular faces comes from Finite Elements. Also, for a specific but natural choice of M_1, we will see that L turns out to be the *cotan* Laplacian .

Since we said that both $\mathbb{L}$ and L gives Laplacians, we should mention their differences. Under the spotlight of PDE's, $\mathbb{L}$ is called the *strong* (or *pointwise*) *Laplacian*, whereas L is called the *weak* (or *integrated*) *Laplacian*. Then we define the *Dirichlet energy* E_D of a function u in terms of L as

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

This comes from

$$\begin{aligned} E_D(u) &= \frac{1}{2} \langle \Delta u, u \rangle_L \\ &= \frac{1}{2} \langle u, u \rangle_L \qquad (\text{ as } \Delta \text{ is self-adjoint}) \\ &= \frac{1}{2} u^T L u, \end{aligned}$$

where $\langle -, - \rangle_L$ is the inner product induced by the matrix L.

It is worth noting that L and $\mathbb{L}$ are symmetric matrices with respect to the standard inner product and, the inner product induced by M_0, respectively.

Notation Throughout this text, we embrace the following notation. Bold face letters denote 3-vectors, and upper case letters denote all matrices with the only exception of d. Also, if f is a simple and not necessarily planar polygon with k_f vertices in $\mathbb{R}^3$, then

- $X_f = (\mathbf{x}_1^f, \cdots, \mathbf{x}_k^f)^T$ denotes the $k_f \times 3$ matrix of cyclically ordered vertices along the boundary ∂f,

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

With these notations at hand, let us decribe the desired properties for discrete Laplacians.

5.2.1 Desiderata

Whichever inner product matrices M_0 and M_1 on 0-forms and 1-forms we choose, they would work nicely. However, we want our discrete Laplacian to mimic some core properties of the smooth Laplacian. So we need to apply a number of restrictions to these inner products. Most of these properties were suggested and examined in Chapter 4 and also in [6] for triangular meshes. We will extend these properties to general polygonal meshes.

LOCALITY The definition of the smooth Laplacian is *local* since it is a differential operator. It depends on properties of the underlying Riemannian manifold only inside of a small neighborhood of a point it is defined on. In discrete case, locality means that the change in the value of an operator on some vertex does not affect its value on a neighboring vertex. So in order to keep locality property, we need to work with only diagonal matrices M_0 so that applying M_0 does not influence neighbor vertices. On the other hand, we define M_1 separately for each face of the mesh. So if α and β is any pair of discrete 1-forms, then the inner product matrix M_1 is defined as:

$$\alpha^T M_1 \beta = \sum_{f \in F} \alpha_{|f}^T M_f \beta_{|f}, \tag{5.13}$$

Here $\alpha_{|f}$ and $\beta_{|f}$ are standard restrictions to the boundary ∂f of f. So, for each face $f \in F$, M_f is a symmetric $k_f \times k_f$ matrix. In terms of implementation, locality and *sparsity* corresponds to each other. The term sparsity has no formal definition, but it corresponds to amount of zero entries of a matrix. In some applications, sparsity of

a matrix is calculated by the ratio of the number of non-zero entries to number of all entries of a matrix, that is, the more zeros a matrix has, the more sparse it gets. For calculations, it is better for a matrix to have more and more zero entries.

SYMMETRY The symmetry of M_f, and hence the symmetry of M_1, corresponds to smooth Laplacian being self-adjoint with respect to the L^2 inner product on 0-forms on a Riemannian manifold without boundary. This is equivalent, in the discrete case, to $\mathbb{L}$ being self-adjoint with respect to the inner product defined by the matrix M_0 , that is $L = L^T$.

POSITIVE SEMI-DEFINITENESS On a Riemannian manifold without boundary, the smooth Laplacian is positive semi-definite whose kernel is one-dimensional and equal to the constants. This property is required for the existence and uniqueness of the solutions to various variational problems. Similarly, in the discrete case, each inner product matrix M_0, M_f, and thus M_1, should be positive definite. So the kernels of $\mathbb{L}$ and L are automatically one-dimensional since d has a one-dimensional kernel spanned by constants.

LINEAR PRECISION In the smooth case, for a planar domain $M \subset \mathbb{R}^2$ and for any linear map $u : \mathbb{R}^2 \to \mathbb{R}$, we have $\Delta u = 0$. Similar to this, in the discrete case, if a single plane contains all the vertices, then for any linear map u, we have $(\mathbb{L}u)_p = 0$ at each interior vertex. This property is essential in applications like mesh parametrization, where a planar mesh should remain unchanged after parametrization is applied. If we define an additional notation, we can rewrite linear precision geometrically. Let

$$\star E_f = M_f E_f \tag{5.14}$$

be a $k_f \times 3$ matrix whose rows, $\star \mathbf{e}_i^f$, are called *dual edges*. Now, let $f \in F$ be a face that contains a fixed vertex $p \in V$. Also let i and j be the row indices of E_f, respectively corresponding to the unique half-edge $\mathbf{e}_i^f$ pointing away from p, and the unique half-edge $\mathbf{e}_j^f$ pointing towards p (see Figure 5.1). Then, the property of linear precision is equivalent to the *integrability condition*:

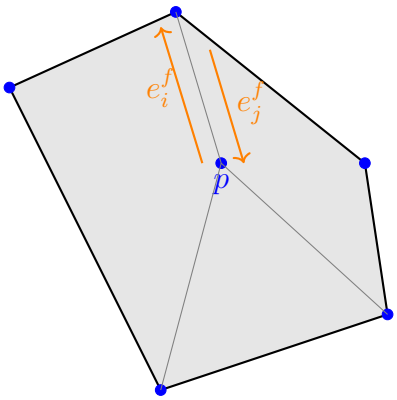


Figure 5.1: Half-edges e_i^f and e_j^f on the face f.

$$0 = \sum_{f \ni p} \left(\star \mathbf{e}_j^f - \star \mathbf{e}_i^f \right), \tag{5.15}$$

where the sum is taken over all faces f containing p, and i and j varies between faces. The Equation 5.15 suggests that appropriately oriented dual edges around each inner vertex form a *closed* polygon. Since M is in the plane, resulting closed polygon is planar, and hence $(\mathbb{L}u)_p = 0$ for each interior vertex p.

SCALE INVARIANCE The smooth Dirichlet energy is a *conformal invariant* in two dimensions. That is, it does not change when the Riemannian metric g is changed conformally, $g \mapsto \lambda g$. Particularly, Dirichlet energy does not change under uniformly re-scaling a smooth embedded surface. So, the weak discrete Laplacian L must remain unaltered under uniformly re-scaling a mesh, that is, M_f must be invariant under uniform scaling.

CONVERGENCE Under suitable operations and with respect to appropriate norms, we should have $\mathbb{L}_n \to \Delta$ so that we can get well-defined limits under mesh refinement. We do not provide any convergence proofs in this text, but one should know that experiments give satisfactory numerical convergence results for our construction [1].

Then, for each simple polygon f, if we can build a positive definite, symmetric and scale invariant matrix M_f such that $\star E_f$ is defined as (5.14) and the condition (5.15)

is satisfied at each inner vertex of a planar mesh, we can guarantee that all of the above desiderata is satisfied, except possibly *convergence*.

Before constructing discrete Laplacians satisfying our desiderata, we need to generalize the area of a triangle to a polygon. We do that by remembering the notion of the *vector area*.

5.2.2 Vector Area and Maximal Projection

Let $\gamma \subset \mathbb{R}^3$ be a simple closed curve, which is also the boundary of a sufficiently regular surface. Then the *vector area*, $\mathbf{A}(\gamma)$, of γ is given by the surface integral of the normal vector of the surface with boundary γ. The vector area is constructed as follows: Let M be a disk-like region whose boundary ∂M is our closed curve γ, $\mathbf{x}$ be the position vector (0-form) of γ, and N be the unit normal vector field on M induced by $\mathbf{x}$. If dA represents the standard volume form on M, we have

$$df \wedge df(u,v) = df(u) \times df(v) - df(v) \times df(u) = 2df(u) \times df(v) = 2NdA(u,v)$$

for any pairs of vectors $u, v \in \mathbb{R}^2$. Then, for the vector area, we get

$$\mathbf{A}(\gamma) = \int_M N dA = \frac{1}{2}\int_M d\mathbf{x} \wedge d\mathbf{x} = \frac{1}{2}\int_M d(\mathbf{x} \wedge d\mathbf{x}) = \frac{1}{2}\int_{\partial M=\gamma} \mathbf{x} \wedge d\mathbf{x} \quad (5.16)$$

by applying the Stoke's theorem on the last step. $\mathbf{A}(\gamma)$ depends only on the boundary curve γ, *not* on the choice of a particular surface spanning this curve [30]. For polygonal curve, we have the following representation:

Lemma 11. *Let $f \in \mathbb{R}^3$ be a simple and not necessarily planar polygon. Then*

$$A_f := E_f^T B_f \quad (5.17)$$

is a 3×3 matrix which is skew-symmetric, has maximal rank 2 and its Darboux vector, $[A_f]$, is equal to the vector area $\boldsymbol{A}(f)$ of f.

Proof. Let the face f has cyclically ordered oriented edges $(\mathbf{e}_1, \cdots, \mathbf{e}_k)$ with the corresponding barycenters $(\mathbf{b}_1, \cdots, \mathbf{b}_k)$. If we use a counter clock-wise rotation on the

vertices, we get $\mathbf{e}_i = \mathbf{x}_{i+1} - \mathbf{x}_i$ and $\mathbf{b}_i = (\mathbf{x}_{i+1} + \mathbf{x}_i)/2$. Hence,

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

is skew-symmetric, where $\otimes$ is the outer product and we assume that the last sum is cyclic. Also, it is easy to see that $\frac{1}{2}(\mathbf{x}_{i+1} \otimes \mathbf{x}_i - \mathbf{x}_i \otimes \mathbf{x}_{i+1})$ is a skew-symmetric matrix whose Darboux vector is equal to $\frac{1}{2}(\mathbf{x}_i \times \mathbf{x}_{i+1})$. So, the last term turns out to be equal to the vector area of the triangle with vertices $0, \mathbf{x}_i$ and $\mathbf{x}_{i+1}$, where 0 is the origin. Hence the Darboux vector $[A_f]$ is equal to the vector area of the polyhedral surface with triangular faces and vertices $0, \mathbf{x}_1, \cdots, \mathbf{x}_k$ since the vector area depends only on the boundary curve, not the spanning surface. □

Recall that if A is a skew-symmetric 3×3 matrix, its *Darboux vector* is the unique vector $[A] \in \mathbb{R}^3$ with $[A] \times \mathbf{x} = A\mathbf{x}$ for all $\mathbf{x} \in \mathbb{R}^3$. Also note that if the orientation of the boundary is changed, then the sign of the vector area also changes.

The largest signed area over all orthogonal projections of a polygon f to any plane in $\mathbb{R}^3$ is the magnitude $|f| = ||\mathbf{A}(\gamma)||$ of the vector area of f. Then, an orthogonal projection $\bar{f}$ of a possibly non-planar polygon f is called a *maximal projection* if f and $\bar{f}$ has the same vector area. The face f is the maximal projection of itself if it is planar. Let $\bar{f}$ be a planar polygon with vertex set $\{\bar{\mathbf{x}}_1, \cdots, \bar{\mathbf{x}}_k\}$ and face normal $\bar{\mathbf{n}}$. The polygon f is said to have *height vector* $h = (h_1, \cdots, h_k)^T \in \mathbb{R}^k$ over $\bar{f}$ if the vertex set $\{\mathbf{x}_1, \cdots, \mathbf{x}_k\}$ of f satisfies the equality $\mathbf{x}_i = \bar{\mathbf{x}}_i + h_i \bar{\mathbf{n}}$. Now we have the question of finding all possible height vectors so that a given planar polygon $\bar{f}$ is a maximal projection of f. A precise answer to this question is given in the following lemma:

Lemma 12. *Let f be a (not necessarily planar) polygon. Its orthogonal projection onto a planar polygon $\bar{f}$ is maximal if and only if the height vector $h_b = \frac{1}{2}(h_1 + h_2, \cdots, h_k + h_1)^T$ from the midpoints is in the null space of $E_{\bar{f}}^T$.*

Proof. We know that $\bar{f}$ is a maximal projection of f if and only if $\bar{f}$ is an orthogonal projection of f and $A_f = A_{\bar{f}}$. By using $\mathbf{x}_i = \bar{\mathbf{x}}_i + h_i \bar{\mathbf{n}}$ and the proof of Lemma 11,

we observe that $A_f = A_{\bar{f}}$ is the same as

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

where we assume the sum to be cyclic. Since $(\bar{\mathbf{n}} \otimes \bar{\mathbf{x}}_i - \bar{\mathbf{x}}_i \otimes \bar{\mathbf{n}})$ has Darboux vector $(\bar{\mathbf{x}}_i \times \bar{\mathbf{n}})$, we arrive at the following requirement proving the claim:

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

□

Mean curvature and the Laplacian of an embedded surface are closely related objects. To get the mean curvature *vector* $\mathbb{H}$, one applies the Laplacian to every component of the surface positions $\mathbf{x}$, that is $\mathbb{H} = \Delta\mathbf{x}$. This formulation gives the point-wise mean curvature vector $\mathbb{H}$

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

In other words, $\mathbb{H}$ is a matrix of dimension $|V| \times 3$, whose rows are corresponding to 3-vectors from $\mathbb{H}$ which are correspondent with the vertices of the polygonal mesh. In the smooth setup, $\mathbb{H}$ is given by taking the L^2 gradient of the area functional, so $(-\mathbb{H})$ works as a gradient, that is, it contains the surface flow's direction and speed in which the surface area decreases maximally. Analogously, the same holds in the discrete setup with the cotan formula which was derived precisely as the gradient of surface area of a triangulated surface mesh in Chapter 3. Our main purpose in this chapter is to generalize this idea to arbitrary polygons, so we use polygonal vector area instead of triangle area to get the lemma below:

Lemma 13. *Let f be a not-necessarily planar polygon in $\mathbb{R}^3$ whose vector area is $|f|$. Then we have the following equality for the gradient of $|f|$ with respect to varying vertex $\boldsymbol{x}_i$ of f*

$$\nabla_{x_i}|f| = \left(\widetilde{L}_f X_f\right),$$

where $\widetilde{L}_f := d^T\widetilde{M}_f d$ and $\widetilde{M}_f$, which is a symmetric $k_f \times k_f$ matrix, is given by

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

Proof. Again consulting the proof of Lemma 11, we can see that

$$(\nabla_{x_i}|f|)^T = \frac{1}{|f|} A_f(\mathbf{b}_{i-1} - \mathbf{b}_i).$$

Hence,

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

Then putting $E_f = dX_f$ we get

$$\begin{aligned}
(\nabla_{x_i}|f|)^T &= \frac{1}{|f|}(dX_f)^T B_f B_f^T d \\
\Rightarrow \nabla_{x_i}|f| &= \frac{1}{|f|} d^T B_f B_f^T dX_f \\
\therefore \nabla_{x_i}|f| &= d^T \widetilde{M}_f d = \left(\widetilde{L}_f X_f\right)_i .
\end{aligned}$$

□

Despite the fact that $\widetilde{M}_f$ depends on the choice of origin of a coordinate system, calculation of Laplacian and mean curvature makes the choice of origin irrelevant due to presence of d. For implementation purposes, we set the origin to be the center of mass of the vertices spanning f, that is $0 = \sum_{i=1}^{k_f} \mathbf{x}_i$. This choice leads to numerically stable representations. But, *all* of the following objects and results will be independent of the choice of origin, unless otherwise stated.

5.2.3 Discrete Laplacians

We can finally give Laplacians satisfying our desiderata with the tools we obtained in Section 5.2.2. Recall that we defined the inner product on 1-forms with a matrix M_1, and since the inner product should be local, we redefined it for each face f of the mesh as M_f. To construct these matrices, we first consider the matrices $\widetilde{M}_f$ defined above.

Lemma 14. *The matrices $\widetilde{M}_f$ results in producing positive semi-definite inner products on 1-forms and give pre-Laplacians $\widetilde{L}_f = d^T \widetilde{M}_f d$ that are local, linearly precise, and scale invariant.*

Proof. Let $\star E_f := \widetilde{M}_f E_f$. We know that by Lemma 11, $|f| \star \mathbf{e}_i^f = [A_f] \times \mathbf{b}_i^f$ for the ith column of the matrix $\star E_f^T$. Specifically, we obtain $\star \mathbf{e}_i^f = \mathbf{n} \times \mathbf{b}_i^f$ for planar

meshes, where $\mathbf{n}$ is the unit normal vector of the plane. If we use $\mathbf{b}^f$ instead of $\star\mathbf{e}^f$, the barycenters $\mathbf{b}_i^f$ satisfy the Equation 5.15 because of pairwise cancellations as we cyclically move on the faces around an inner vertex. Thus, the vectors $\star\mathbf{e}_i^f = \mathbf{n} \times \mathbf{b}_i^f$ also satisfy the Equation 5.15 for planar meshes. □

One may question that why do we consider non-planar polygons instead of planar ones. The matrices $\widetilde{M}_f$ are also considered in the constructions of [5, 8]. But, although $\widetilde{M}_f$ is positive semi-definite due to its construction, it is not positive definite in general as our desiderata require it to be. To solve this problem without sacrificing linear precision, we need to generalize the construction of Brezzi et al. [5] from the case of planar polygons to non-planar polygons.

Note that E_f has a non-trivial kernel if and only if f is planar. Specifically, the Darboux vector $[A_f]$ is in the kernel of E_f if and only if f is planar. So, if f is planar, rank of E_f^T is 2; otherwise its maximal rank is 3.

Let $C_{\bar{f}}$ be a $k_f \times (k_f - 2)$ matrix whose columns span the kernel of $E_{\bar{f}}^T$, where $\bar{f}$ is the maximal projection of f. Such a matrix C is called *admissible*. Similarly, any symmetric and positive definite $(k_f - 2) \times (k_f - 2)$ matrix $U_{\bar{f}}$ is called *admissible*.

Theorem 15. *Let $C_{\bar{f}}$ and $U_{\bar{f}}$ be admissible matrices. Then the matrices*

$$M_f := \widetilde{M}_f + C_{\bar{f}} U_{\bar{f}} C_{\bar{f}}^T \tag{5.18}$$

define a positive definite inner products on 1-forms and gives local and linearly precise Laplacians $L_f := d^T M_f d$.

Proof. The original proof of Brezzi [5] applies to the non-planar case with a few modifications. The $k_f \times k_f$ matrix M_f is already symmetric and positive semi-definite by its construction. So we only need to show that its kernel is trivial. That is, if $M_f v = 0$ for any $v \in \mathbb{R}^k$, showing that $v = 0$ is enough. Since $v^T M_f v = 0$, we have

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

where $U_{\bar{f}}^{1/2}$ is a square root of the matrix $U_{\bar{f}}$. So $B_f^T v = C_{\bar{f}}^T v = 0$. Since $ker(E_{\bar{f}}^T) = im(C_{\bar{f}})$, we have

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

Thus, there exists $\mathbf{u} \in \mathbb{R}^3$ with $v = E_{\bar{f}}\mathbf{u}$. Then we get $0 = B_f^T v = B_f^T E_{\bar{f}}\mathbf{u}$. Having $E_f = E_{\bar{f}} + h_e \bar{\mathbf{n}}^T$ at hand, where $h_e = dh_f$ and h_f is the height vector of f over its maximal projection $\bar{f}$, we obtain $0 = B_f^T E_{\bar{f}}\mathbf{u} = B_f^T E_f \mathbf{u}$. This last equality follows from $B_f^T H_e = -E_f^T H_b = -E_{\bar{f}}^T h_b - \bar{\mathbf{n}} h_e^T h_b = 0$ by Lemma 12. So, by Lemma 11, $A_f^T \mathbf{u} = -A_f \mathbf{u} = 0$. Since the kernel of A_f is spanned by its Darboux vector $[A_f]$ which is parallel to $\bar{\mathbf{n}}$, we have $\mathbf{u} = \mu\bar{\mathbf{n}}$. Hence, since $\bar{\mathbf{n}}$ is orthogonal to the rows of $E_{\bar{f}}$, we have $v = \mu E_{\bar{f}}\bar{\mathbf{n}} = 0$. □

Scale invariance from our desiderata is missing from the properties of M_f we listed above. In fact, scale invariance is an *extra* condition which will be supplemented by the choice of admissible matrices $U_{\bar{f}}$ and $C_{\bar{f}}$.

Choosing C and U There are several possible admissible choices which can lead us to scale invariance. But we will focus on λ-simple choices, that is, we let $U_{\bar{f}} := \lambda\mathrm{Id}$ for some $0 < \lambda \in \mathbb{R}$, and $C_{\bar{f}}$ be such that its columns are orthonormal. Although the choice of $C_{\bar{f}}$ is not very specific, together with our choice of $U_{\bar{f}}$, the expression $C_{\bar{f}} U_{\bar{f}} C_{\bar{f}}^T$ remains invariant under the choice of orthonormal bases to represent $C_{\bar{f}}$. Although studying a larger range of choices of $C_{\bar{f}}$ and $U_{\bar{f}}$ is left for future studies [1], we should note that our choice leads to scale invariant discrete Laplacians. Moreover, our construction generalizes the well-known cotan formula:

Theorem 16. *For triangular meshes, any admissible choice of $C_{\bar{f}}$ and $U_{\bar{f}}$, not necessarily λ-simple, leads to the cotan Laplacian via $L = d^T M_1 d$.*

Proof. From our construction, in the case of f being a triangle, C_f should be a 3-vector parallel to $(1,1,1)^T$, and U_f is a positive scalar. In particular, $C_f^T d = 0$. Thus, we have $L = d^T M_f d = d^T \widetilde{M}_f d = |f|^{-1}(d^T B_f)(d^T B_f)^T$. Also, since $B_f = (\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3)^T$ by definition, we get

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

Hence, the entry (i,j) of $(d^T B_f)(d^T B_f)^T$ is given by the inner product $\frac{1}{4}\langle \mathbf{e}_{i+1}, \mathbf{e}_{j+1} \rangle$ whose indexes are considered modulo 3. Since inner products have cosine, and the area $|f|$ can be calculated via the sine formula, we get $L_{ij} = -\frac{1}{2}\cot\alpha_{ij}$ for distinct

i and j, where $0 < \alpha_{ij} < \pi$ is the angle between edges $\mathbf{e}_{i+1}$ and $\mathbf{e}_{j+1}$, which is our usual cotan operator. □

This finishes the construction of a suitable inner product on 1-forms. Now, to finish constructing discrete Laplacians, we, finally, need to specify matrices M_0 which define positive inner products on 0-forms. Since discrete Laplacian must be local, our construction is restricted to diagonal matrices M_0. We define

$$(M_0)_{pp} := \sum_{f \ni p} \frac{|f|}{k_f}, \tag{5.19}$$

that is, each f incident to a vertex p adds $\frac{1}{k_f}$ of the norm of its vector area to the total mass of p.

To sum up Starting from writing the Laplacian as $\Delta = d^*d$, we used d as standard co-boundary operator and then, defined its formal adjoint d^* in terms of inner products M_0 and M_1 on 0- and 1-forms, respectively. Since smooth Laplacian is a local operator, we required that M_0 and M_1 to be local too via setting M_0 as a diagonal matrix, and defining M_1 on each face f separately as matrices M_f. Hence, combining these definitions and Theorem 15, we get Laplacians in (5.12) satisfying our desiderata.

5.2.4 Discussion

Our Laplacian consists of a geometric and a combinatorial part. For any polygon f, $\widetilde{M}_f$ characterizes the geometry of f. As an example, one can think about the *exact* Dirichlet energy for linear functions over planar polygons. The $C_{\bar{f}} U_{\bar{f}} C_{\bar{f}}^T$ part is the combinatorial part, which makes M_f positive definite. Finding out a geometric term to replace this combinatorial term is left for future research [1].

Mean curvature revisited If f is a polygonal face, then the local mean curvature vector H_f is defined to be a $k_f \times 3$ matrix whose rows are indexed by vertices satis-

fying

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

As in a previous discussion, the use of the notation *mean curvature vector* is the same as *area gradient* here. In fact, by Lemma 13, $\widetilde{H}_f$ coincides with the gradient of vector area of f. In order to understand the effect of the combinatorial part of our Laplacians, and especially the effect of choosing λ, we need to look at the difference $H_f - \widetilde{H}_f$.

Lemma 17. *Let f be a not necessarily planar polygon with maximal projection $\bar{f}$ whose height vector is denoted by h_f. Then $\widetilde{H}_f = H_{\bar{f}}$. Also, if $C_{\bar{f}}$ and $U_{\bar{f}}$ are chosen to be λ-simple, we have*

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

In order to prove Lemma 17, we first need another lemma:

Lemma 18. *Let $h = (h_1, \cdots, h_k)^T$ be the height vector of a polygon f whose corresponding maximal projection is $\bar{f}$, and assume that $\bar{f}$ contains $(0,0,0) \in \mathbb{R}^3$. Then, $\widetilde{M}_f - \widetilde{M}_{\bar{f}} = |f|^{-1} h_b \otimes h_b$, where $h_b = 1/2(h_1 + h_2, \cdots, h_k + h_1)^T$ denotes the midpoint height vector.*

Proof of Lemma 18. We have $B_f = B_{\bar{f}} + \bar{h}_b \bar{\mathbf{n}}^T$, where $\bar{\mathbf{n}}$ is defined as before. Moreover, $B_f^T = B_{\bar{f}}^T + \bar{h}_b \bar{\mathbf{n}}$. Since $(0,0,0) \in \bar{f}$, $\bar{\mathbf{n}}$ is orthogonal to each column of $B_{\bar{f}}$. Thus we have

$$\begin{aligned} B_f B_f^T &= B_{\bar{f}} B_{\bar{f}}^T + B_{\bar{f}} \bar{h}_b \bar{\mathbf{n}} + \bar{h}_b \bar{\mathbf{n}}^T B_{\bar{f}}^T + (\bar{h}_b \bar{\mathbf{n}}^T) \otimes (\bar{h}_b \bar{\mathbf{n}}) \\ &= B_{\bar{f}} B_{\bar{f}}^T + \bar{h}_b \otimes \bar{h}_b. \end{aligned}$$

Since $|f| = |\bar{f}|$, if we divide each side of the above equation by $|f|$, we get the desired result.

$$\widetilde{M}_f - \widetilde{M}_{\bar{f}} = |f|^{-1} h_b \otimes h_b.$$

□

Now we can prove Lemma 17:

Proof of Lemma 17. We start with showing that $\widetilde{H}_f = H_{\bar{f}}$. Since the definition of $\widetilde{H}_f$ is independent of the choice of origin, we can safely assume that $\bar{f}$ contains the origin. Let h_f be the height vector of f over its maximal projection $\bar{f}$ with corresponding midpoint height vector h_b. Then Lemma 18 asserts that $\widetilde{M_f} - \widetilde{M_{\bar{f}}} = |f|^{-1} h_b \otimes h_b$. Since $h_e = dh_f$, we have $E_f = E_{\bar{f}} + h_e \bar{\mathbf{n}}^T$. We know that h_b is in the null space of $E_{\bar{f}}^T$ by Lemma 12, so $h_b^T h_e = 0$. Thus, $(h_b \otimes h_b) E_f = 0$ which yields the result $\widetilde{M_f} E_f = \widetilde{M_{\bar{f}}} E_f$. By using the definition of $\widetilde{M_{\bar{f}}}$, we get $|\bar{f}| \widetilde{M_{\bar{f}}} E_f = B_{\bar{f}} B_{\bar{f}}^T (E_{\bar{f}} + h_e \bar{\mathbf{n}}^T)$. So, now $E_{\bar{f}}^T h_b = 0$ (Lemma 12) is the same as $B_{\bar{f}}^T = 0$. Therefore, $\widetilde{M_{\bar{f}}} E_{\bar{f}} = \widetilde{M_{\bar{f}}} E_f = \widetilde{M_f} = E_f$, thus $\widetilde{H}_f = H_{\bar{f}}$.

Finally, combining the facts that $E_f = E_{\bar{f}} h_e \bar{\mathbf{n}}^T$, the columns of $C_{\bar{f}}$ form an orthonormal basis spanning the null space of $E_{\bar{f}}^T$, and $U_{\bar{f}} = \lambda \mathrm{Id}$, we arrive at

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

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

The mean curvature vectors of f and $\bar{f}$ uniquely specify the planar component of the mean curvature vector of f. Additionally, mean curvature vectors of these polygons differ from each other by a term along the normal direction with respect to the latter one. This difference term also shows how non-planar f is in terms of the *combinatorial* second derivative $d^T \bar{h}_e$ of the height vector h_f. Therefore, the choice of λ suggests how far f is from being planar.

Relation to DEC We know that the Laplacian on a Riemannian 2-manifold satisfies $\Delta = -\star d \star d$, where $\star$ is the Hodge star operator [13]. Similarly, discrete Laplacians come from a discrete Hodge star. On triangle meshes, using *diagonal* Hodge star operators as in DEC, and inner products on k-forms as we do in this text are equivalent to each other. However, using M_1 inner product on general polygonal meshes does not correspond to a *diagonal* Hodge star in general.

5.3 Implementation

As mentioned before, our algebraic approach to Laplacian leads to a straightforward implementation. Interestingly, our approach is simpler than the usual cotan approach on triangular meshes. We only need to construct three matrices d, M_1, and M_0. Then we combine these matrices according to result in (5.12).

The construction of M_0 is trivial and given in (5.19). However, the co-boundary matrix d and the inner product M_1 on 1-forms needs us to describe half-edges with indices.

The entries of the $(2|E_I| + |E_B|) \times |V|$ matrix d are $d_{ep} = \pm 1$ if $e = \pm e_{qp}$ and $d_{ep} = 0$ otherwise. We compute M_1 on each face separately: first we compute the matrices M_f, then keep the values in the corresponding entries of M_1 using the given index scheme. Computation of each M_f is discussed in the previous section, and is summarized as a pseudo-code [1]:

procedure COMPUTE M_f(polygonal face f, parameter λ)
Require: $B\,, E\,, \bar{E} \in \mathbb{R}^{\deg(f)\times 3}$
for each vertex coordinate $\mathbf{x}_i$ in f **do**
$E(i) = (\mathbf{x}_{x+1} - \mathbf{x}_i)^T$
$B(i) = \frac{1}{2}(\mathbf{x}_{i+1} + \mathbf{x}_i)^T$
end for
$A = E^T B$
$\tilde{M} = \frac{\sqrt{2}}{||A||} B B^T$
$\bar{\mathbf{n}}$ = normalize$(-A_{23}\,, A_{13}\,, -A_{12})^T$
for each vertex coordinate $\mathbf{x}_i$ in f **do**
$\bar{\mathbf{x}}_i = \mathbf{x}_i - (\mathbf{x}_i \cdot \bar{\mathbf{n}})\bar{\mathbf{n}}$
$\bar{E}(i) = (\bar{\mathbf{x}}_{x+1} - \bar{\mathbf{x}}_i)^T$
end for
$C =$ orthogonal kernel of $\bar{E}^T$ (e.g., using LU, then SVD)
$U = \lambda$Id
return $M_f = \tilde{M} + CUC^T$
end procedure

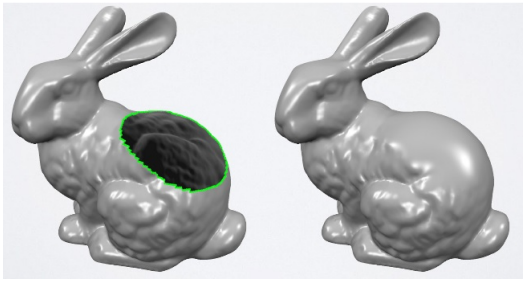


Figure 5.2: Smoothing a surface using Laplace operator [1].

5.4 Results & Applications

From now on, when we use our construction of the Laplace operator, we know that all geometry processing operations using geometric Laplacians are valid on general polygonal meshes. We show this fact for some of the applications of Laplacian. In the below examples, we take $\lambda = 2$ unless stated otherwise. As a matter of fact, choosing $\lambda \in [1, 3]$ gives comparable results.

5.4.1 Implicit mean curvature flow

A main use of Laplace operator in geometry processing is mesh fairing, or surface smoothing. A common way to do so is by using a discrete mean curvature flow. In this way each vertex flows with speed and direction given by the point-wise mean curvature vector. Desbrun et al. [9] gives a stable implementation using an implicit time stepping scheme. His approach is for triangular meshes, however it can be generalized to the general polygonal setup in view of the flow

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

Here X is a polygonal mesh. When we integrate both sides of this identity, it turns out to be a smoothing operator. In Figure 5.2, a portion of a surface can be seen to be smoothed by implementing this mean curvature flow several times.

5.4.2 A planarizing flow

From the discussion from a previous section, we know that the mean curvature vectors of f, and its maximal projection $\bar{f}$'s disagree by a term determining how far f from being a planar polygon. This difference term is completely given by the combinatorial term CUC^T of our Laplacian. With this fact in mind, we can define a new 'Laplacian' $\mathcal{L}$ as before, but we drop $\widetilde{M}_f$ from the original construction. Then, we have an energy, $E_{plan}(X)$, on the mesh X, punishing non-planarity, defined as

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

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

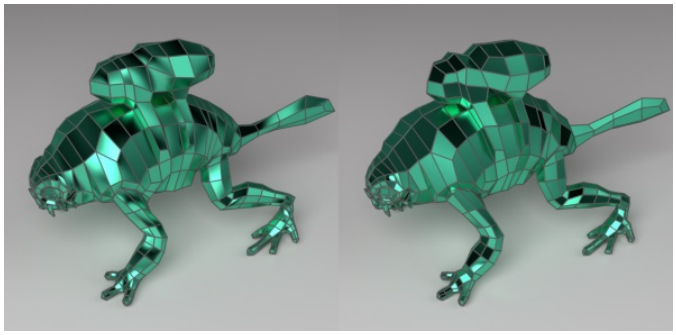


Figure 5.3: Left: A mesh with non-planar faces. Right: The same mesh after applying the above planarizing flow [1].

The Figure 5.3 shows that how this flow affects a non-planar mesh. We know that this flows does not come from a PDE as CUC^T is a combinatorial term.

REFERENCES

[1] M. Alexa and M. Wardetzky, "Discrete laplacians on general polygonal meshes," *ACM Trans. Graph.*, vol. 30, July 2011.

[2] H. Eves, *An introduction to the history of mathematics*. Saunders College Publishing, 1983.

[3] K. Crane, "Discrete differential geometry: An applied introduction." `http://www.cs.cmu.edu/~kmcrane/Projects/DGPDEC/paper.pdf?cv=1`, 4 2020.

[4] E. Grinspun, M. Desbrun, K. Polthier, P. Schröder, and A. Stern, "Discrete differential geometry: an applied introduction," *ACM Siggraph Course*, vol. 7, 2006.

[5] F. Brezzi, K. Lipnikov, and V. Simoncini, "A family of mimetic finite difference methods on polygonal and polyhedral meshes," *Mathematical Models and Methods in Applied Sciences*, vol. 15, pp. 1533–1551, 2005.

[6] M. Wardetzky, S. Mathur, F. Kälberer, and E. Grinspun, "Discrete laplace operators," *ACM SIGGRAPH ASIA 2008 courses on - SIGGRAPH Asia '08*, 2008.

[7] F. Aurenhammer, "A criterion for the affine equivalence of cell complexes in $\mathbb{R}^d$ and convex polyhedra in $\mathbb{R}^{d+1}$," *Discrete & Computational Geometry*, vol. 2, pp. 49–64, 1987.

[8] J. B. Perot and V. Subramanian, "Discrete calculus methods for diffusion," *Journal of Computational Physics*, vol. 224, pp. 59–81, 2007.

[9] M. Desbrun, M. Meyer, P. Schröder, and A. H. Barr, "Implicit fairing of irregular meshes using diffusion and curvature flow," *Proceedings of the 26th annual conference on Computer graphics and interactive techniques - SIGGRAPH '99*, 1999.

[10] S. Lang, *Curvature and the Variation Formula*, pp. 294–321. New York, NY: Springer New York, 1999.

[11] H. Federer, "Curvature measures," *Transactions of the American Mathematical Society*, vol. 93, no. 3, pp. 418–491, 1959.

[12] J. M. Sullivan, "Curvature measures for discrete surfaces," *ACM SIGGRAPH 2006 Courses*, pp. 10–13, 2006.

[13] S. Rosenberg, *The Laplacian on a Riemannian Manifold*. Cambridge University Press, 10 1997.

[14] G. Taubin, "Geometric signal processing on polygonal meshes," *Proceedings of EUROGRAPHICS*, 2000.

[15] H. Zhang, "Discrete combinatorial laplacian operators for digital geometry processing," *Proceedings of SIAM Conference on Geometric Design and Computing. Nashboro Press*, pp. 575–592, 2004.

[16] M. S. Floater and K. Hormann, "Surface parameterization: a tutorial and survey," *Advances in multiresolution for geometric modelling*, pp. 157–186, 2005.

[17] O. Sorkine, "Laplacian mesh processing," *Eurographics (State of the Art Reports)*, pp. 53–70, 2005.

[18] J. C. Maxwell, "Xlv. on reciprocal figures and diagrams of forces," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 27, pp. 250–261, 1864.

[19] L. Cremona, *Graphical statics: two treatises on the graphical calculus and reciprocal figures in graphical statics*. Clarendon Press, 1890.

[20] M. Wardetzky, M. Bergou, D. Harmon, D. Zorin, and E. Grinspun, "Discrete quadratic curvature energies," *Computer Aided Geometric Design*, vol. 24, pp. 499–518, 2007.

[21] W. T. Tutte, "How to draw a graph," *Proceedings of the London Mathematical Society*, vol. 3, pp. 743–767, 1963.

[22] S. J. Gortler, C. Gotsman, and D. Thurston, "Discrete one-forms on meshes and applications to 3d mesh parameterization," *Computer Aided Geometric Design*, vol. 23, pp. 83–112, 2006.

[23] K. Hildebrandt, K. Polthier, and M. Wardetzky, "On the convergence of metric and geometric properties of polyhedral surfaces," *Geometriae Dedicata*, vol. 123, pp. 89–112, 2006.

[24] U. Pinkall and K. Polthier, "Computing discrete minimal surfaces and their conjugates," *Experimental Mathematics*, vol. 2, 10 1993.

[25] D. Glickenstein, "Geometric triangulations and discrete laplacians on manifolds," *arXiv preprint math/0508188*, 2005.

[26] M. Desbrun, A. N. Hirani, M. Leok, and J. E. Marsden, "Discrete exterior calculus," *arXiv preprint math/0508341*, 2005.

[27] M. S. Floater, "Mean value coordinates," *Computer aided geometric design*, vol. 20, pp. 19–27, 2003.

[28] A. Bobenko and B. A. Springborn, "A discrete laplace–beltrami operator for simplicial surfaces," *Discrete & Computational Geometry*, vol. 38, 10 2007.

[29] N. Sharp and K. Crane, "A laplacian for nonmanifold triangle meshes," in *Computer Graphics Forum*, vol. 39, pp. 69–80, Wiley Online Library, 2020.

[30] J. M. Sullivan, *Curvatures of smooth and discrete surfaces*. Springer, 2008.