Lecture 12: Discrete Laplacian

Scribe: Tianye Lu

Our goal is to come up with a discrete version of Laplacian operator for triangulated surfaces, so that we can use it in practice to solve related problems. We are mostly interested in the standard *Poisson problem*:

$$\Delta f = g$$

We will first introduce some basic facts and then talk about discretization. By the end, we will be able to derive a discretized linear system from poisson problem and calculate the numerical solutions.

1 Facts and Tools

Laplacian Operator: Δ

Laplacian Operator is a linear functional on $C^\infty(M)$, i.e. $\Delta : f(x) \in C^\infty(M) \to \Delta f(x) \in C^\infty(M)$.

If $M = \mathbb{R}^n$, it can be explicitly expressed as a derivative operator[1]: $\Delta = -\sum\limits_i \frac{\partial^2}{\partial x_i^2}$. That is to say, if we have a region $\Omega \in \mathbb{R}^n$ and a function $f(x) \in C^\infty(\Omega)$, then $\Delta f(x) = -\sum\limits_i \frac{\partial^2}{\partial x_i^2} f(x)$.

If M is a surface, which is more often the case, one standard definition of Laplacian operator is: $\Delta f = \nabla \cdot \nabla f$, where $\nabla \cdot$ and ∇ are divergence and gradient operator respectively. However, this definition is not straigtforward for discretization. Alternatively, we will avoid this problem with Galerkin's approach.

Before we go on, let's briefly talk about why Laplacian operator is important. Laplacian operator is used in many important partial differential equations, which are the keys to many mathematical and physical models. Here are some examples:

- The heat equation $\frac{\partial u}{\partial t} = -\Delta u$ describes the distribution of heat in a given region over time.
- The eigenfunctions of Δ (Recall that a matrix is a linear operator defined in a vector space and has its eigenvectors in the space; similarly, the Laplacian operator is a linear operator defined in a function space, and also has its eigenfunctions in the function space) are the solutions to the equation $\Delta u = \lambda u$. If S is a surface and $u \in C^\infty(S)$, the eigenfunctions describe the vibration modes of the surface S.

Galerkin's Approach

Given a function defined on M, i.e. $f : M \to \mathbb{R}$, its $\mathcal{L}^2$ dual $\mathcal{L}_f$ is defined on the function space $\mathcal{L}^2(M)$.

$$\mathcal{L}_f : \mathcal{L}^2(M) \to \mathbb{R}$$

$$\mathcal{L}_f[g] = \int_M fg dA, \ \forall g \in \mathcal{L}^2(M)$$

$\mathcal{L}^2(M)$ is all the square integrable functions on M. More rigorously, $\mathcal{L}^2(M) = \{f : M \to \mathbb{R} : \int_M f^2 < \infty\}$.

The function g is often called a *test function.* We will only deal with the following set of test functions in our discussion:

$$\{g \in C^\infty(M) : g|_{\partial M} = 0\}$$

Often, we are interested in a compact surface without boundary, so $\partial M = \emptyset$.

[1] Note that the sign of Laplacian operator is inconsistent in different literature.

Note that we can also recover the function f from its dual $\mathcal{L}_f$. Figure 1 gives an intuition about this process. We can take g to be the square function and as g approaches a single point when it gets narrow, $\mathcal{L}_f[g]$ approaches the value of f at that particular point.

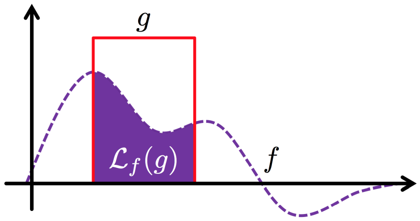


Figure 1: $\mathcal{L}_f \to f$

We apply $\mathcal{L}^2$ dual to Laplacian using the test functions stated above (note that the test functions vanish on the boundary), and use integration by parts:

$$\begin{array}{rcl} \mathcal{L}^2_{\Delta f}[g] & = & \int_M g \Delta f dA \\ & = & \textit{boundary terms} - \int_M \nabla g \cdot \nabla f dA \\ & = & -\int_M \nabla g \cdot \nabla f dA \end{array}$$

Notice that we have used Laplacian without actually evaluating it. This is an example of *Galerkin's approach*, which is a class of methods for converting a continuous operator problem to a discrete problem (e.g. for discretizing and solving differential equations). In this approach we need to decide on a function space (where function f come from) and a test function space (where function g come from; we can often apply boundary conditions by choosing test functions). We'll see this method in practice in the next section.

2 Discretization

In this section, we will first pose a different representation of the Poisson problem and then use the tools from the previous section to derive a discretization.

Weak Solution

Consider Poisson Equation $\Delta f = g$. As stated before, it is hard to directly derive a good discretization for the equation. Therefore, we seek a different representation with the concept of weak solution. The weak solutions for the Poisson Equation are functions f that satisfy the following set of equations[2]:

$$\int_M \phi \Delta f dA = \int_M \phi g dA, \forall \; test \; functions \; \phi$$

Recall that according to Galerkin's method, we need to choose the basis function for f, g and test function ϕ.

First Order Finite Element

Since we are focusing on the Poisson Equation on a surface M, f and g should be functions defined on the surface. Therefore we need a set of basis function on M. For triangulated surface, the most natural choice of basis functions are the piecewise linear *hat functions* h_i, which equal one at their associated vertex and zero at all other vertices, as shown in Figure 2.

[2] The notation here is different from the last section. Specifically, Δf and g are functions we want to discretize and ϕ is test function.

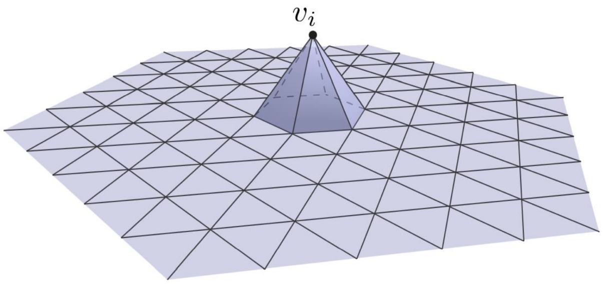


Figure 2: One hat function per vertex

Therefore, if we know the value of $f(x)$ on each vertex, $f(v_i) = a_i$, we can approximate it with:

$$f(x) = \sum_i a_i h_i(x)$$

Since $h_i(x)$ are all fixed, we can store f with only a single array $\vec{a} \in \mathbb{R}^{|V|}$. Similarly, we can have $g(x) = \sum\limits_i b_i h_i(x)$.

With the discretization of f and g, using h_i as test functions, recall the $\mathcal{L}^2$ dual of Δf and the equations for weak solutions of Poisson Equation, we can now pose the original Poisson problem as a set of $|V|$ equations:

$$\int_M h_i \Delta f dA = \int_M h_i g dA, \forall i \in \{1, 2, \ldots, |V|\}$$

For the left hand side:

$$\begin{aligned} \textstyle\int_M h_i \Delta f dA &= -\textstyle\int_M \nabla h_i \cdot \nabla f dA \\ &= -\textstyle\int_M \nabla h_i \cdot \nabla(\sum\limits_j a_j h_j) dA \\ &= -\sum\limits_j a_j \textstyle\int_M \nabla h_i \cdot \nabla h_j dA \end{aligned}$$

Suppose matrix $L = \{L_{ij}\}_{|V|\times|V|}$, $L_{ij} = \int_M \nabla h_i \cdot \nabla h_j dA$. Then the left hand side of the set of equations is: $\begin{pmatrix} \int_M h_1 \Delta f dA \\ \int_M h_2 \Delta f dA \\ \cdots \\ \int_M h_{|V|} \Delta f dA \end{pmatrix} = \begin{pmatrix} \sum\limits_j L_{1j} a_j \\ \sum\limits_j L_{2j} a_j \\ \cdots \\ \sum\limits_j L_{|V|j} a_j \end{pmatrix} = L\vec{a}$

For the right hand side:

$$\begin{aligned} \textstyle\int_M h_i g dA &= \textstyle\int_M h_i \cdot (\sum\limits_j b_j h_j) dA \\ &= \sum\limits_j b_j \textstyle\int_M h_i \cdot h_j dA \end{aligned}$$

Similarly, suppose matrix $A = \{A_{ij}\}_{|V|\times|V|}$, $A_{ij} = \int_M h_i \cdot h_j dA$. The right hand side of the set of equations is $A\vec{b}$.

Now we derive a linear problem $L\vec{a} = A\vec{b}$, we only need to calculate the matrices L and A in order to solve $\vec{a}$.

Cotan Laplacian

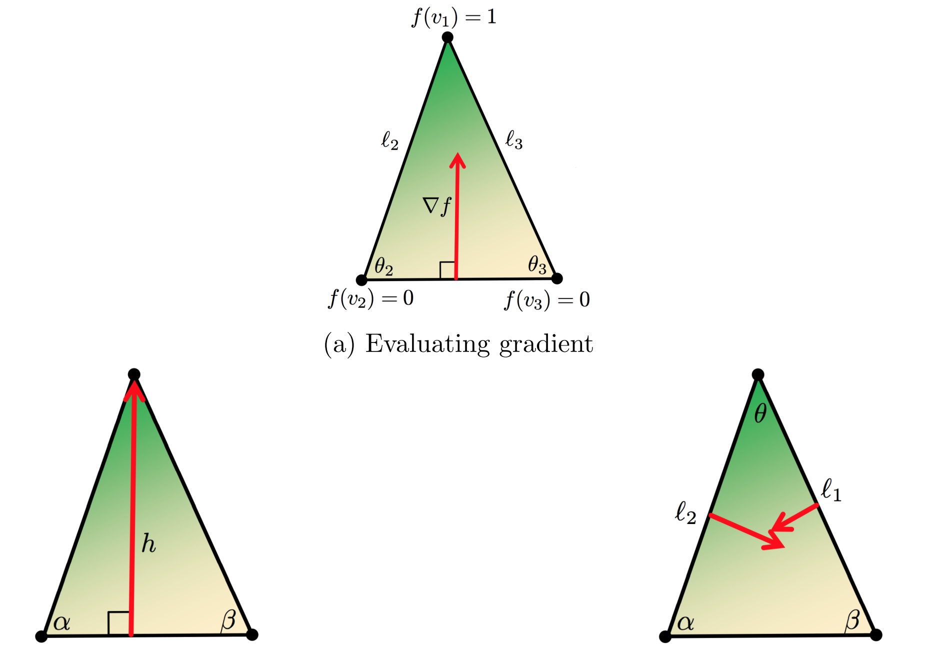
We now try to calculate the matrix L by examining its element $L_{ij} = \int_M \nabla h_i \cdot \nabla h_j dA$. Since h_i are piecewise linear functions on a triangular face, ∇h_i is a constant vector on a face, and thus

$\nabla h_i \cdot \nabla h_j$ yields one scaler per face. Therefore, to calculate the integral above, for each face we multiply the scalar $\nabla h_i \cdot \nabla h_j$ on that face by the area of the face and then sum the results.

Let's first evaluate the gradient. Now consider a linear function f defined on a triangle so that $f(v_1) = 1, f(v_2) = f(v_3) = 0$, as shown in Figure 3a. For a linear function, we have $f(x) = f(x_0) + \nabla f|_{x_0} \cdot (x - x_0)$. Let $x_0 = v_1$, $x = v_2$ and v_3 respectively, and notice that ∇f lies within the triangle face, we get:

$$\begin{aligned} \nabla f \cdot (v_1 - v_3) &= 1 \\ \nabla f \cdot (v_1 - v_2) &= 1 \\ \nabla f \cdot n &= 0 \end{aligned}$$

This yields: $\nabla f \cdot (v_2 - v_3) = 0$. Therefore, ∇f is perpendicular to the edge $v_2 v_3$, as shown in Figure 3a.



(a) Evaluating gradient

(b) Case 1: Same vertex

(c) Case 2: Different vertices

Figure 3

Now we know the direction of the gradient, we will go on to evaluate its magnitude.

$$\begin{aligned} 1 &= \nabla f \cdot (v_1 - v_3) \\ &= ||\nabla f|| l_3 \cos(\tfrac{\pi}{2} - \theta_3) \\ &= ||\nabla f|| l_3 \sin \theta_3 \\ ||\nabla f|| &= \tfrac{1}{l_3 \sin \theta_3} \\ &= \tfrac{1}{h} \end{aligned}$$

where h is the height of the triangle corresponding to edge $v_2 v_3$. Recall that triangle area $A = \frac{1}{2} |v_2 v_3| \cdot h$, thus

$$\nabla f = \frac{e_{23}^{\perp}}{2A}$$

$e_{23}^{\perp}$ is the vector from v_2 to v_3 rotated by a quarter turn in the counter-clockwise direction.

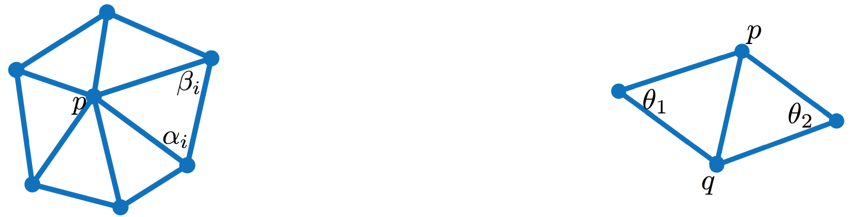
Now we have the gradient vectors on a face, we need to take dot products of them to get the scalar associated with each face. There are two different cases. Let's still look at a single triangle face and functions defined on it for now. The first case is when two functions are defined on the same vertex.

$$\begin{array}{rcl}\int_T \langle \nabla f, \nabla f \rangle dT & = & A||\nabla f||^2 = \frac{A}{h^2} = \frac{b}{2h} \\ & = & \frac{(h\cot\alpha + h\cot\beta)}{2h} \\ & = & \frac{1}{2}(\cot\alpha + \cot\beta)\end{array}$$

The second case is when two functions are defined on different vertices (but of the same edge).

$$\begin{array}{rcl}\int_T \langle \nabla f_\alpha, \nabla f_\beta \rangle dT & = & A\langle \nabla f_\alpha, \nabla f_\beta \rangle = \frac{1}{4A}\langle e_{31}^\perp, e_{12}^\perp \rangle = \frac{-l_1 l_2 \cos\theta}{4A} \\ & = & \frac{-(h/\sin\beta)(h/\sin\alpha)\cos\theta}{2bh} \\ & = & \frac{-h\cos\theta}{2(h\cot\alpha + h\cot\beta)\sin\alpha\sin\beta} \\ & = & \frac{-\cos\theta}{2(\cos\alpha\sin\beta + \cos\beta\sin\alpha)} \\ & = & \frac{-\cos\theta}{2\sin(\alpha+\beta)} = \frac{-\cos\theta}{2\sin\theta} \\ & = & -\frac{1}{2}\cot\theta\end{array}$$

Now we can apply these results on hat functions $\{h_i\}$ by simply summing around each vertex.



(a) Case 1: Same vertex

(b) Case 2: Different vertices

Figure 4: Summing around each vertex

$$\langle \nabla h_p, \nabla h_p \rangle = \frac{1}{2}\sum_i (\cot\alpha_i + \cot\beta_i)$$

$$\langle \nabla h_p, \nabla h_q \rangle = -\frac{1}{2}(\cot\theta_1 + \cot\theta_2)$$

Finally, we get the cotangent Laplacian matrix L:

$$L_{ij} = \begin{cases} \frac{1}{2}\sum\limits_{i\sim j}(\cot\alpha_j + \cot\beta_j), & \text{if } i = j. \\ -\frac{1}{2}(\cot\alpha_j + \cot\beta_j), & \text{if } i \sim j. \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

$i \sim j$ means that vertex i and vertex j are adjacent.

Mass Matrix

For the right hand side, we need to calculate the matrix A, which is often called the *mass matrix*. As it involved the product of h_i and h_j, the result would be quadratic. There are several approaches to deal with it. A straightforward and consistent approach is to just do the quadratic integral.

Similarly to the approach to calculate L, we fist integrate on a single triangle.

$$A_{ij}^{triangle} = \begin{cases} area/6 & \text{if } i = j \\ area/12 & \text{if } i \neq j \end{cases} \tag{2}$$

Then we can sum up around each vertex. There are two cases as shown in Figure 4.

$$A_{ij} = \begin{cases} \frac{one-ring\ area}{6} & \text{if } i = j \\ \frac{adjacent\ area}{12} & \text{if } i \neq j \end{cases} \tag{3}$$

Some properties of the mass matrix constructed this way: each row sums to one third of the one-ring area of the vertex corresponding to that row; to construct the mass matrix it involves only vertex and its neighbors; it partitions surface area as the weight to assign to each vertex. The matrix can be used for calculating integration (notice that $\sum_i h_i = 1$):

$$\begin{aligned} \int_M f &= \int_M \sum_j a_j h_j \\ &= \int_M \sum_j a_j h_j (\sum_i h_i) \\ &= \sum_{ij} A_{ij} a_j \\ &= \vec{1}^T A \vec{a} \end{aligned}$$

Setting $\vec{a} = \vec{1}$ will give us the surface area.

However, there is one drawback of this approach. The mass matrix constructed is not diagonal which means it is often hard to manipulate it. We can turn it into a diagonal matrix by introducing certain approximations.

From the previous example, we have seen that the mass matrix is actually integrating a function on the surface. Therefore, we can try to find different ways to do the integration. For example, the *lumped mass matrix* finds the dual cells of each vertex and approximate the diagonal of A with the areas of each cell:

$$a_{ii} = Area(cell\ i)$$

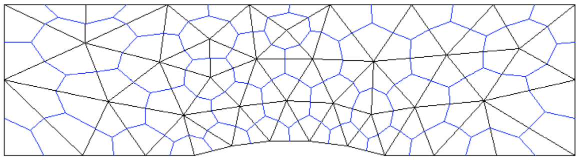


Figure 5: Lumped Mass Matrix with dual cells

Such approximation won't make a difference for smooth functions. Intuitively, the function values of adjacent vertices are very close for a smooth functions, so the result won't change a lot if we add them all to the diagonal. Meanwhile, as the mesh gets more and more refined, we can argue that the result would converge.

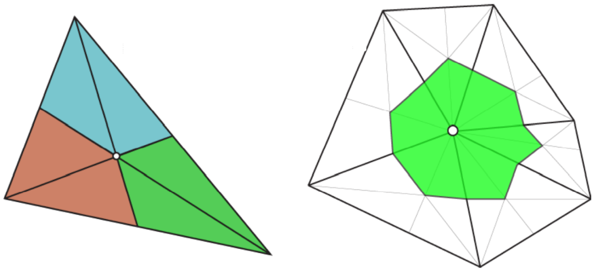


Figure 6: Barycentric Lumped Mass Matrix

There are many ways to choose the dual cell. One simple solution is the *barycentric lumped mass matrix*. We assign the dual of each face to be the barycentric of the triangle. Therefore, each vertex has a third of its one-ring area assigned to it. The resulting cells are likely to be of irregular shapes. One alternative approach is to take Voronoi Cells and use the areas accordingly.