

## Weekly Exercises 4

Room: 02.09.023

Wed, 14.06.2017, 14:00-16:00

Submission deadline: Tue, 13.06.2017, 23:59 to laehner@in.tum.de

### Mathematics: Triangle Meshes

**Exercise 1** (4 points). Recap the hat functions on triangles, the first fundamental form and the derivation of the mass matrix in 2D from the lecture.

1. Derive the mass matrix for 3D triangular meshes with the hat functions  $\phi_i$  as a basis. Each entry can be calculated by taking  $M_{ij} = \int_{\mathcal{M}} \phi_i(p)\phi_j(p)dp$ . It will help to use a reference triangle with vertices  $(0, 0)$ ,  $(1, 0)$ ,  $(0, 1)$  and deconstruct the integral for each triangle. The final matrix should look like this:

$$M_{ij} = \begin{cases} \frac{\text{area}(T_1)+\text{area}(T_2)}{12} & (i, j) \text{ is an edge} \\ \sum_{T_k \in N(i)} \frac{\text{area}(T_k)}{6} & i = j \\ 0 & \text{otherwise} \end{cases}$$

$T_1, T_2$  are triangles that share the edge  $(i, j)$ .  $N(i)$  is the set of triangles adjacent to vertex  $i$ .

2. Show that the mass matrix is symmetric and positive definite. With this properties we know that it defines an inner product  $\langle \mathbf{f}, \mathbf{g} \rangle_{\mathbf{M}} = \mathbf{f}^T \mathbf{M} \mathbf{g}$  (and therefore also induces a norm).

**Solution.** 1. Since our parametrization consists of triangles, we need to deconstruct the integral into separate integrals for each triangle and sum these up:

$$M_{jk} = \int_S \psi_j(p)\psi_k(p)dp = \sum_{i=1}^m \int_{T_i} \psi_j(p)\psi_k(p)dp$$

Notice that the integral is zero if neither vertex  $j$  nor  $k$  is adjacent to  $T_i$ . This leaves two cases:  $j = k$  or  $j, k$  are connected by an edge. We solve for both separately. First, we introduce a reference triangle with vertices  $(0, 0)$ ,  $(1, 0)$ ,  $(0, 1)$  and a coordinate mapping  $x_k : \text{Ref} \rightarrow T_k$  that maps each vertex of the reference triangle to a vertex of  $T_k$ . Let  $u = (0, 1)$  and  $v = (1, 0)$  then we can write any point in the reference triangle as a convex combination of  $u, v$ . The same holds for any point  $T_k$  that can be represented by a convex

combination of the vectors  $l, k$  which are the mappings of  $u, v$  by  $Dx_k$ . As a formula:

$$\begin{aligned} x_k(u, v) &= u \cdot l + v \cdot k \\ Dx_k(u, v) &= \begin{pmatrix} l_1 & k_1 \\ l_2 & k_2 \\ l_3 & k_3 \end{pmatrix} \\ g_{x_k} &= Dx_k^\top Dx_k = \begin{pmatrix} \|l\|_2^2 & \langle k, l \rangle \\ \langle k, l \rangle & \|k\|_2^2 \end{pmatrix} \end{aligned}$$

Since we need it later we already derive  $\sqrt{\det(g_{x_k})} = 2\text{area}(T_k)$ . A lot of trigonometry is involved. We denote the angle between  $l$  and  $k$  with  $\alpha$ .

$$\begin{aligned} \sqrt{\det(g_{x_k})} &= \sqrt{\|l\|_2^2 \|k\|_2^2 - \langle k, l \rangle^2} \\ &= \sqrt{\|l\|_2^2 \|k\|_2^2 \cdot \underbrace{(1 - \cos(\alpha))^2}_{=\sin(\alpha)^2}} \\ &= 2 \cdot \text{area}(T_k) \end{aligned}$$

Finally, we handle each case ( $j = k$ ,  $(j, k)$  edge) separately:

$$\begin{aligned} M_{jj} &= \sum_{k \in \mathcal{N}_j} \int_{T_k} \psi_j(p)^2 dp \\ \int_{T_k} \psi_j(p)^2 dp &= \int_0^1 \int_0^{1-u} \psi_j(x_k(u, v))^2 \cdot \sqrt{\det(Dx_k^\top Dx_k)} \, dv \, du \\ &= 2 \cdot \text{area}(T_k) \int_0^1 \int_0^{1-u} (1 - u - v)^2 \, dv \, du \\ &= 2 \cdot \text{area}(T_k) \int_0^1 \int_0^{1-u} (u^2 + v^2 + 2uv - 2u - 2v + 1) \, dv \, du \\ &= 2 \cdot \text{area}(T_k) \int_0^1 [u^2v + 1/3v^3 + uv^2 - 2uv - v^2 + v]_0^{1-u} \, du \\ &= 2 \cdot \text{area}(T_k) \int_0^1 \frac{1}{3}(1 - u)^3 \, du \\ &= 2 \cdot \text{area}(T_k) \left[ -\frac{1}{12}(1 - u)^4 \right]_0^1 \\ &= 2 \cdot \text{area}(T_k) \cdot \frac{1}{12} = \frac{1}{6} \text{area}(T_k) \end{aligned}$$

In the second case it is important to remember that the functions are one at

different vertices.

$$\begin{aligned}
 M_{ij} &= \sum_{k \in \mathcal{N}_i \cap \mathcal{N}_j} \int_{T_k} \psi_j(p) \psi_i(p) dp \\
 \int_{T_k} \psi_j(p) \psi_i(p) dp &= \int_0^1 \int_0^{1-u} \psi_j(x_k(u, v)) \cdot \psi_i(x_k(u, v)) \cdot \sqrt{\det(Dx^\top Dx)} \, dv du \\
 &= 2 \cdot \text{area}(T_k) \int_0^1 \int_0^{1-u} (1 - u - v) \cdot u \, dv du \\
 &= 2 \cdot \text{area}(T_k) \int_0^1 \left( \frac{1}{2} u^3 - u^2 + \frac{1}{2} u \right) du \\
 &= 2 \cdot \text{area}(T_k) \frac{1}{24} = \frac{1}{12} \cdot \text{area}(T_k)
 \end{aligned}$$

2. For symmetric we have to show that  $M_{ij} = M_{ji}$  which automatically follows from

$$M_{ij} = \frac{1}{12} \sum_{k \in \mathcal{N}_i \cap \mathcal{N}_j} \text{area}(T_k)$$

For positive definiteness we can argue by the construction of  $M$

$$\mathbf{f}^\top M \mathbf{f} = \int_S f(x)^2 dx > 0 \quad \forall \mathbf{f} \neq 0$$

or use the result that real, symmetric, weak diagonal dominant matrices with positive entries on the diagonal are positive definite (Carl D. Meyer, Matrix Analysis and Applied Linear Algebra).

## Programming: Also Triangle Meshes

Download the supplementary material from the homepage. It contains two files describing a 3D triangular mesh.

**Exercise 2** (3 points). We represent 3D shapes as triangle meshes  $\mathcal{M} = (\mathcal{V}, \mathcal{F})$  embedded in  $\mathbb{R}^3$ . They consist of vertices  $\mathcal{V} = \{v_1, \dots, v_n\}$  and triangles  $\mathcal{F} \subset \mathcal{V} \times \mathcal{V} \times \mathcal{V}$  connecting them to a closed surface. The coordinates of each vertex are denoted by  $p(v_i) \in \mathbb{R}^3$ . Note that the vertices of a triangle  $t = (u, v, w) \in \mathcal{F}$  are ordered. We define triangles to be identical if they can be transformed into each other by a cyclic permutation, i.e.  $(u, v, w) = (v, u, v) = (w, u, v)$  but  $(u, v, w) \neq (w, v, u)$ . The order defines the direction of the normal and is also called orientation.

1. There are different formats to store triangle meshes. The easiest one is the vert-tri format consisting of two files, one containing the 3D coordinates and the other one the triangles. The file `name.vert` contains  $n$  rows with three double numbers separated by tabs. Each row represents one 3D coordinate. The file `name.tri` (note that the file name has to be the same as for the .vert) contains  $m$  rows with three positive integer values  $x \leq n$ . Each  $x$  refers

to a row/vertex from the `.vert` file. The order of  $x$  in each row defines the orientation of the triangle. Write a function `read_vert_tri.m` that takes a filename and returns a struct `M` with fields `M.VERT` a  $n \times 3$  array and `M.TRIV` a  $m \times 3$  array. You can try your function on the `cat0` files from the supplementary material (and see if you did everything correctly in the next part).

2. Triangulated surfaces can be plotted in Matlab with `trisurf(TRI, X, Y, Z)`. You need to give each column of `M.VERT` separately to the function. The colors you see on the surface represent the height function.
3. Implement a function `facearea` that takes a triangle mesh struct as an input and returns a  $\mathbb{R}^m$  vector containing the area of each triangle. You can plot your function by using `trisurf(TRI, X, Y, Z, f)` where  $f$  is a vector with as many entries as the numbers of vertices or faces. (Of course the entries in the vector must be ordered in the same way as the vertices or faces they correspond to.)
4. Implement a function `mass_matrix.m` that takes a triangle mesh and returns a `sparse` (that's a Matlab type) mass matrix as in Exercise 1. (No need to do anything we it, we will use it in later exercise sheets.)