912702055 6.838 Shape Analysis
Congyue Deng **Homework 2** March 20, 2019

# Problem 1

(a) *Proof.* The norm of $f(u,v)$ is

$$\begin{aligned}\|f(u,v)\| &= \frac{\sqrt{(2u)^2+(2v)^2+(u^2+v^2-1)^2}}{u^2+v^2+1} \\ &= \frac{\sqrt{(2u)^2+(2v)^2+(u^2+v^2)^2-2(u^2+v^2)+1}}{u^2+v^2+1} \\ &= \frac{\sqrt{(u^2+v^2)^2+2(u^2+v^2)+1}}{u^2+v^2+1} = \frac{\sqrt{(u^2+v^2+1)^2}}{u^2+v^2+1} = 1.\end{aligned}$$

Thus $f(u,v)$ is always on the unit sphere. □

(b) By chain rule, we have

$$\mathrm{d}f_{\mathbf{p}}(\mathbf{w}) = (f\circ\gamma)'(0) = f_i(\gamma^i)'(0) = f_i(\mathbf{p})w^i = \left(\frac{\partial f}{\partial u}, \frac{\partial f}{\partial v}\right)(\mathbf{p})\begin{pmatrix} w^1 \\ w^2\end{pmatrix}.$$

Thus the differential of $f$ is

$$\mathrm{d}f_{\mathbf{p}} = \left(\frac{\partial f}{\partial u}, \frac{\partial f}{\partial v}\right)(\mathbf{p}) = \frac{1}{(u_0^2+v_0^2+1)^2}\begin{pmatrix} 2(-u_0^2+v_0^2+1) & -4u_0v_0 \\ -4u_0v_0 & 2(u_0^2-v_0^2+1) \\ 4u_0 & 4v_0\end{pmatrix}.$$

(c) Under the orientation $\left[\frac{\partial f}{\partial u}, \frac{\partial f}{\partial v}\right]$, the Guass map is

$$n(\mathbf{p}) = \frac{\frac{\partial f}{\partial u}\times\frac{\partial f}{\partial v}}{\left\|\frac{\partial f}{\partial u}\times\frac{\partial f}{\partial v}\right\|}(\mathbf{p}) = -\frac{1}{u_0^2+v_0^2+1}\left(2u_0, 2v_0, u_0^2+v_0^2+1\right).$$

Or just note that $n = f$ or $-f$ since $f(u,v)$ is always on the unit sphere.

(d) Since $n(\mathbf{p}) = -f(\mathbf{p})$

$$\mathrm{d}n_{\mathbf{p}} = \left(\frac{\mathrm{d}n}{\mathrm{d}u}, \frac{\mathrm{d}n}{\mathrm{d}v}\right)(\mathbf{p}) = -\mathrm{d}f_{\mathbf{p}} = -\frac{1}{(u_0^2+v_0^2+1)^2}\begin{pmatrix} 2(-u_0^2+v_0^2+1) & -4u_0v_0 \\ -4u_0v_0 & 2(u_0^2-v_0^2+1) \\ 4u_0 & 4v_0\end{pmatrix}.$$

# Problem 2

(a) *Proof.* Since $\mathbf{t}_\theta = \mathbf{t}_{\min}\cos\theta + \mathbf{t}_{\max}\sin\theta$ is always orthogonal to $n(\mathbf{p})$, $\mathbf{t}_\theta^T n(\mathbf{p}) = 0$, which yields

$$M_{\mathbf{p}} n(\mathbf{p}) = \frac{1}{2\pi}\int_{-\pi}^{\pi}\kappa_\theta \mathbf{t}_\theta \mathbf{t}_\theta^T n(\mathbf{p})\mathrm{d}\theta = 0.$$

Hence $n(\mathbf{p})$ is an eigenvector of $M_{\mathbf{p}}$ of eigenvalue 0. □

(b) *Proof.* Since $\mathbf{t}_{\min} \cdot \mathbf{t}_{\min} = \mathbf{t}_{\max} \cdot \mathbf{t}_{\max} = 1$ and $\mathbf{t}_{\min} \cdot \mathbf{t}_{\max} = 0$, we have

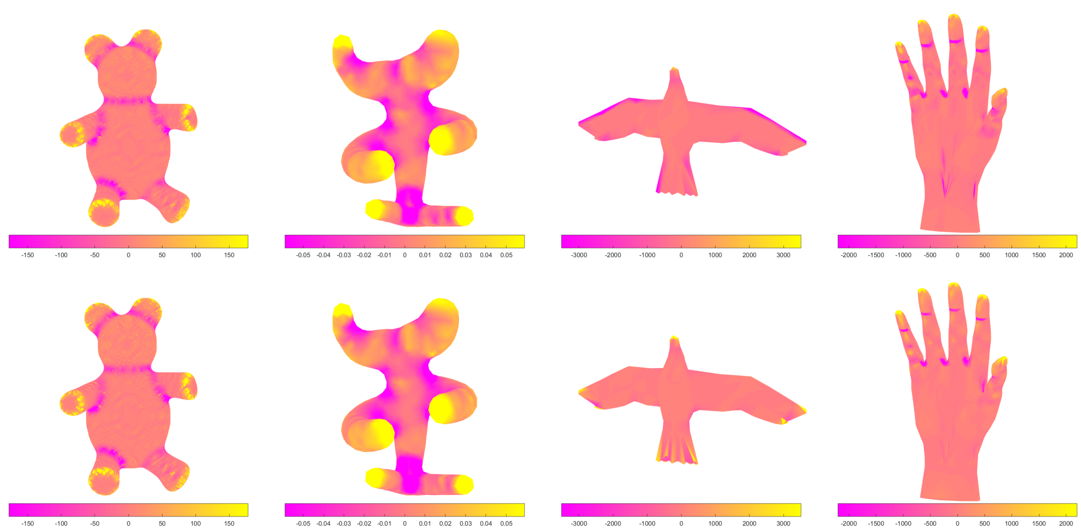
$$\begin{aligned}
\mathbf{t}_{\min}^T M_{\mathbf{p}} \mathbf{t}_{\min} &= \frac{1}{2\pi} \int_{-\pi}^{\pi} \left(\kappa_{\min} \cos^2\theta + \kappa_{\max} \sin^2\theta\right) \cos^2\theta \mathrm{d}\theta \\
&= \frac{\kappa_{\min}}{2\pi} \int_{-\pi}^{\pi} \cos^4\theta \mathrm{d}\theta + \frac{\kappa_{\max}}{2\pi} \int_{-\pi}^{\pi} \sin^2\theta \cos^2\theta \mathrm{d}\theta \\
&= \frac{\kappa_{\min}}{2\pi} \frac{1}{32} (12\theta + 8\sin 2\theta + \sin 4\theta)\Big|_{-\pi}^{\pi} + \frac{\kappa_{\max}}{2\pi} \frac{1}{32} (4\theta - \sin 4\theta)\Big|_{-\pi}^{\pi} \\
&= \frac{3}{8}\kappa_{\min} + \frac{1}{8}\kappa_{\max}
\end{aligned}$$

and similarly

$$\mathbf{t}_{\max}^T M_{\mathbf{p}} \mathbf{t}_{\max} = \frac{1}{2\pi} \int_{-\pi}^{\pi} \left(\kappa_{\min} \cos^2\theta + \kappa_{\max} \sin^2\theta\right) \sin^2\theta \mathrm{d}\theta = \frac{1}{8}\kappa_{\min} + \frac{3}{8}\kappa_{\max}.$$

Therefore $\mathbf{t}_{\min}$ and $\mathbf{t}_{\max}$ are eigenvectors of $M_{\mathbf{p}}$, of eigenvalues $\frac{3}{8}\kappa_{\min} + \frac{1}{8}\kappa_{\max}$ and $\frac{1}{8}\kappa_{\min} + \frac{3}{8}\kappa_{\max}$ respectively. $\square$

# Problem 3



22176 triangles 2086 triangles 462 triangles 3110 triangles

Figure 1: Discrete Gaussian curvature given by the two methods. Top: Estimation of the Gaussian curvature using the Taubin matrix. Bottom: Structure-preserving Gaussian curvature given by Gauss-Bonnet theorem.

First, we can estimate the Gaussian curvature using the Taubin matrix $M_\mathbf{p}$ in Problem 2. Let the two non-zero eigenvalues of $M_\mathbf{p}$ be $\lambda_1 \leqslant \lambda_2$. Then the Gaussian curvature is defined by $K(\mathbf{p}) = \kappa_{\min}\kappa_{\max}$, where $\kappa_{\min}$ and $\kappa_{\max}$ can be solved via

$$\lambda_1 = \frac{3}{8}\kappa_{\min} + \frac{1}{8}\kappa_{\max}, \quad \lambda_2 = \frac{1}{8}\kappa_{\min} + \frac{3}{8}\kappa_{\max}.$$

We can also compute the per-vertex Gaussian curvature using $K(\mathbf{v}) = \left(2\pi - \sum_j \theta_j\right)/A$ (the structure-preserving Gaussian curvature given by Gauss-Bonnet theorem), where $\theta_j$'s are the interior angles at vertex $\mathbf{v}$, and $A$ is barycentric area associated to $\mathbf{v}$.

Figure 2: Performance of the two methods on shape features of the mesh. The left ones are computed by Taubin's method, while the right ones are the structure-preserving curvatures.

Some results are shown in Figure 1. The two methods show similar performances, especially when the triangulation is fine. However, the structure-preserving Gaussian curvature is more alert to sharp features than the Taubin's estimating method, especially when the triangulation is coarse (see Figure 2).

# Problem 4

(a) Surface area of the meshes are as follows.

| mesh | `teddy` | `moomoo` | `bird` | `hand` |
|---|---|---|---|---|
| Area | 4.153460 | 3796.168137 | 0.381569 | 0.852252 |

(b) $L$ can be computed by

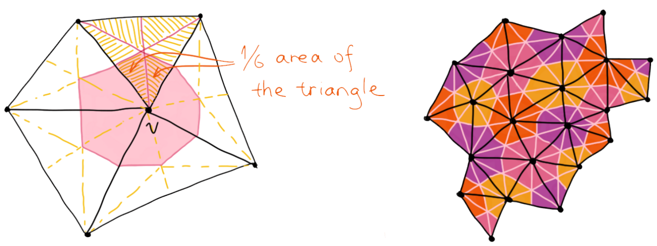
$$L_{ij} = \begin{cases} \sum_{k\sim i}(\cot\alpha_{ik} + \cot\beta_{ik}) & \text{if } i = j \\ -(\cot\alpha_{ij} + \cot\beta_{ij}) & \text{if } i \sim j \\ 0 & \text{otherwise} \end{cases}$$

where $\alpha_{ij}, \beta_{ij}$ are the angles opposite to edge $i - j$ in the two triangles containing $i - j$ respectively.

(c) Taking $h = 0.1$, $0.01$ and $0.001$ and computing the Frobenius norm $\left\|\nabla_{\mathbf{p}} A - \frac{1}{2} L\mathbf{p}\right\|$ respectively, we get the following results.

| mesh | `teddy` | `moomoo` | `bird` | `hand` |
|---|---|---|---|---|
| $h = 0.1$ | $3.041231 \times 10^{-1}$ | $3.381429 \times 10^{-2}$ | $6.201921 \times 10^{-1}$ | $4.612075 \times 10^{-1}$ |
| $h = 0.01$ | $4.420590 \times 10^{-2}$ | $3.384059 \times 10^{-4}$ | $1.527964 \times 10^{-1}$ | $1.785730 \times 10^{-1}$ |
| $h = 0.001$ | $4.937617 \times 10^{-4}$ | $3.384082 \times 10^{-6}$ | $1.580669 \times 10^{-3}$ | $1.424051 \times 10^{-2}$ |

(d) The cell around a vertex consists of exactly 1/3 area of each triangle adjacent to this vertex (Figure 3a). Thus the coefficients in the front of the barycentric area is 1/3.




(a) (b)

Figure 3: Dual cells around vertices on triangle mesh.

Since all the 1/3 parts of each triangle are uniquely assigned to a vertex and there is no overlap (see Figure 3b), the sum of barycentric areas over all vertices is exactly the surface area. Or equivalently, we may use the fact that the barycentric area associated to a vertex is 1/3 times the sum of triangle areas adjacent to that vertex, and each triangle has three vertices.

(e) Combining the previous parts, we compute the per-vertex pointwise mean curvature and the results are shown in Figure 4.

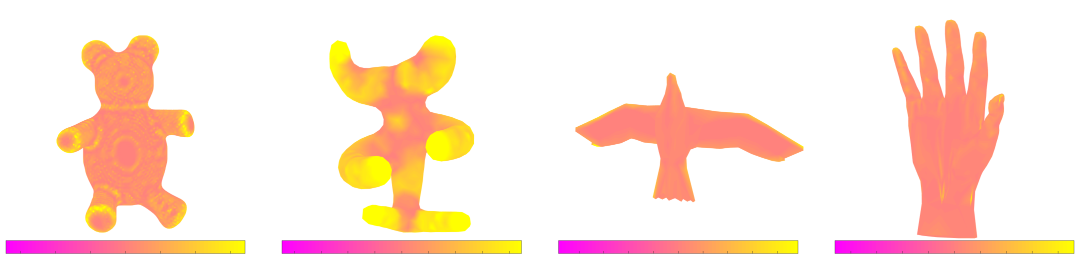



Figure 4: Per-vertex pointwise mean curvature.

# Problem 5

(a) Since the Laplacian operator acting on the parameterization of the surface gives the mean curvature normal (*i.e.* $\Delta_g \varphi = g^{ij}\varphi_{ij} - g^{ij}\Gamma^k_{ij}\varphi_k = H\mathbf{n}$), which is equal to the variational derivative of surface area, the mesh represented by $\mathbf{p}$ will be smoothened and its area will be decreasing over time following this ODE.

(b) The results of the explicit integrator are shown in Figure 5 and Figure 6 (top rows). If $\tau$ is too large, the algorithm will not converge.

(c) The results of the semi-implicit integrator are shown in Figure 7 and Figure 8 (top rows). The semi-implicit method seems to be more stable and less likely to result in singular points than the explicit method. If $\tau$ is too large, the algorithm will not converge.

(d) We do not use the fully-implicit integrator that often because it yields a non-linear equation on $\mathbf{p}(t + \tau)$, while the previous two are both linear.

# Problem 6

According to the paper, instead of requiring that the Laplacian be computed anew at each time-step (as is required by traditional mean-curvature flow), the modified flow simply re-uses the Laplacian from time t = 0. The results of this modification are shown Figure 5, 6, 7 and 8 (bottom rows). Compared with the original method (see the top rows of Figure 5, 6, 7, 8), the modified algorithm successfully avoids the singularities and increases the smoothness of the mesh at each time step.

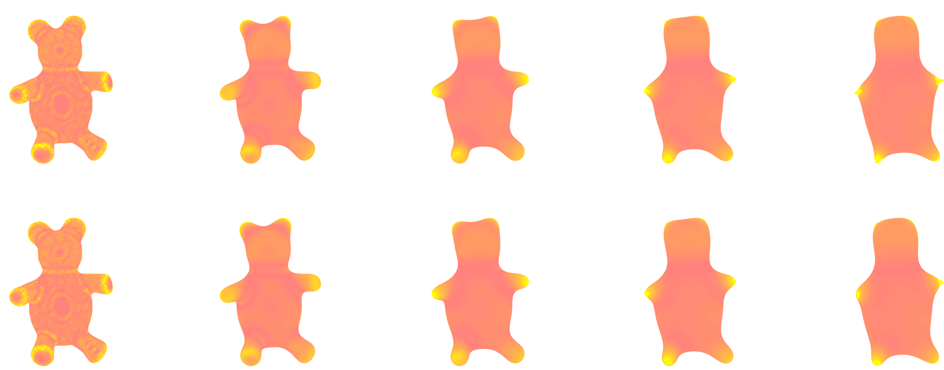


Figure 5: Example of explicit mean curvature flow on `teddy`. Top: initial algorithm. Bottom: modified algorithm which avoids singularities.

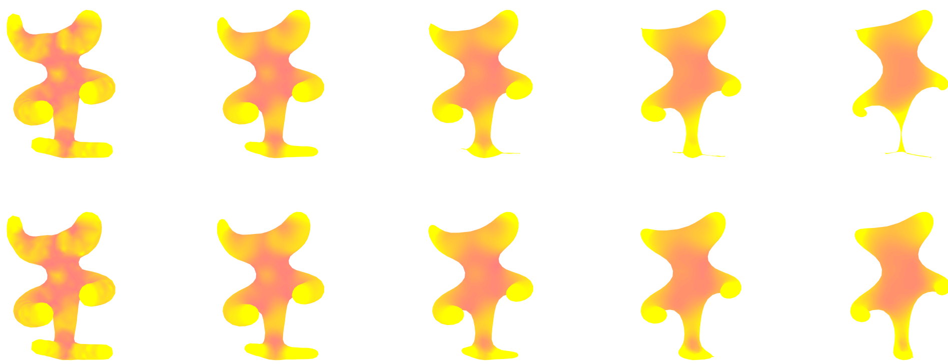


Figure 6: Example of explicit mean curvature flow on `moomoo`. Top: initial algorithm. Bottom: modified algorithm which avoids singularities.

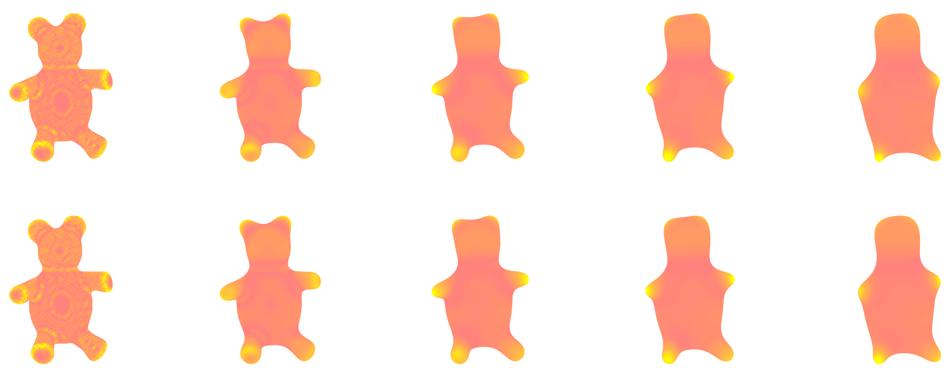


Figure 7: Example of semi-implicit mean curvature flow on `teddy`. Top: initial algorithm. Bottom: modified algorithm which avoids singularities.

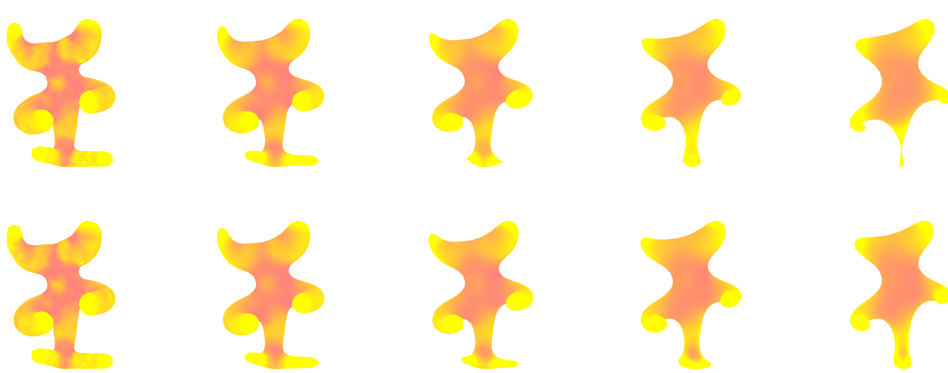


Figure 8: Example of semi-implicit mean curvature flow on `moomoo`. Top: initial algorithm. Bottom: modified algorithm which avoids singularities.