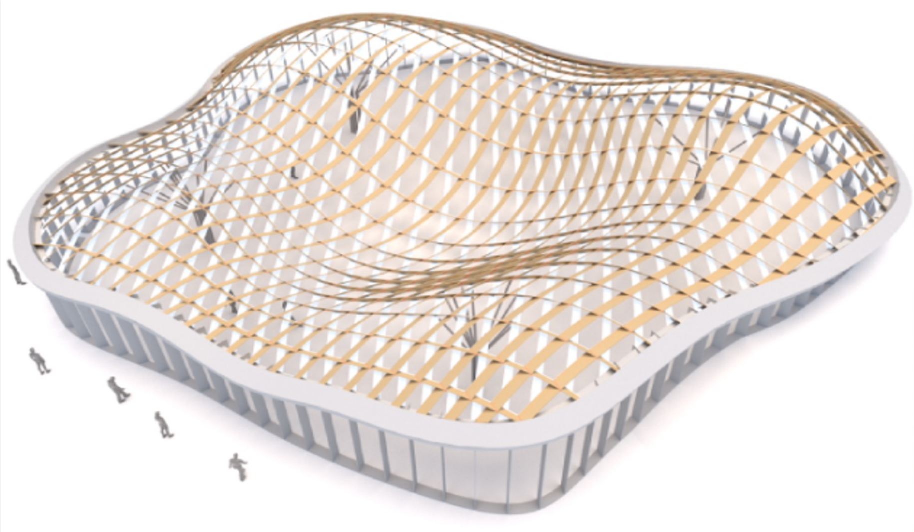
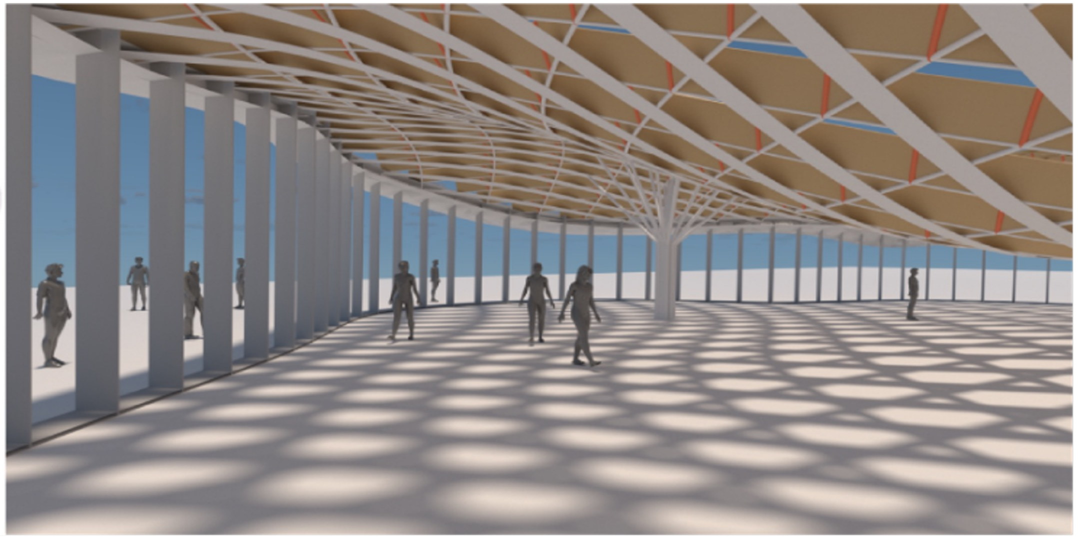
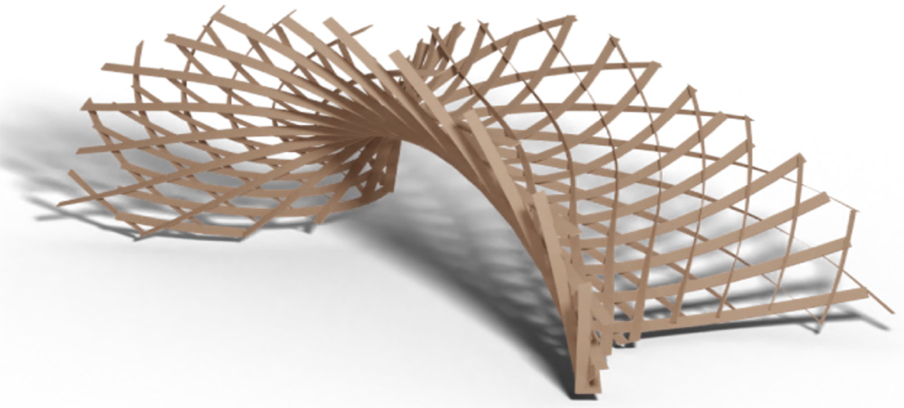
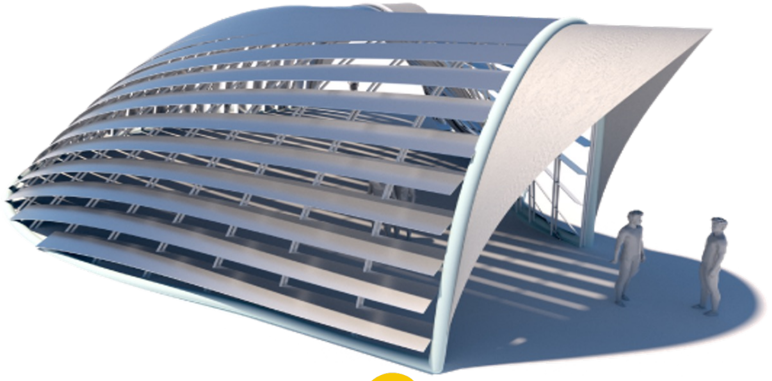
Rectifying Strip Patterns



Bolun Wang[1], Hui Wang[1], Eike Schling[2], Helmut Pottmann[1], Siggraph Asia 2023

KAUST 1

2

Motivations: Gridshell Structures

Eike Schling, et al., 2022

Motivations: Gridshell Structures

Eike Schling, et al., 2022

Motivations: Gridshell Structures

Eike Schling, et al., 2022

Motivations: Gridshell Structures

Strips tangential to the surface

[Natterer et al. 2000]

Motivations: Gridshell Structures

Strips tangential to the surface

Strips orthogonal to the surface

[Natterer et al. 2000]

[Eike et al. 2022]

Motivations: Gridshell Structures

Strips tangential to the surface

[Natterer et al. 2000]

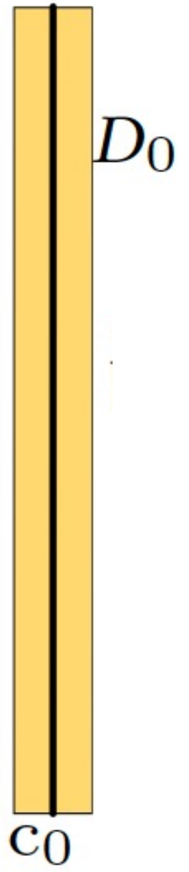
Strips orthogonal to the surface

[Eike et al. 2022]

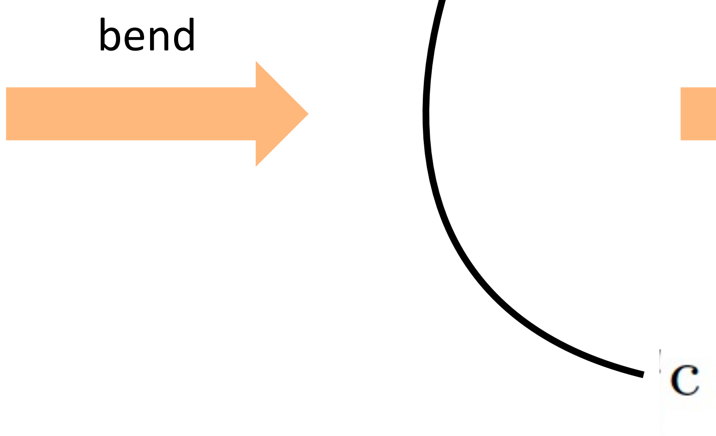
Strips holding a constant angle to the surface

[Mesnil and Baverel 2023]

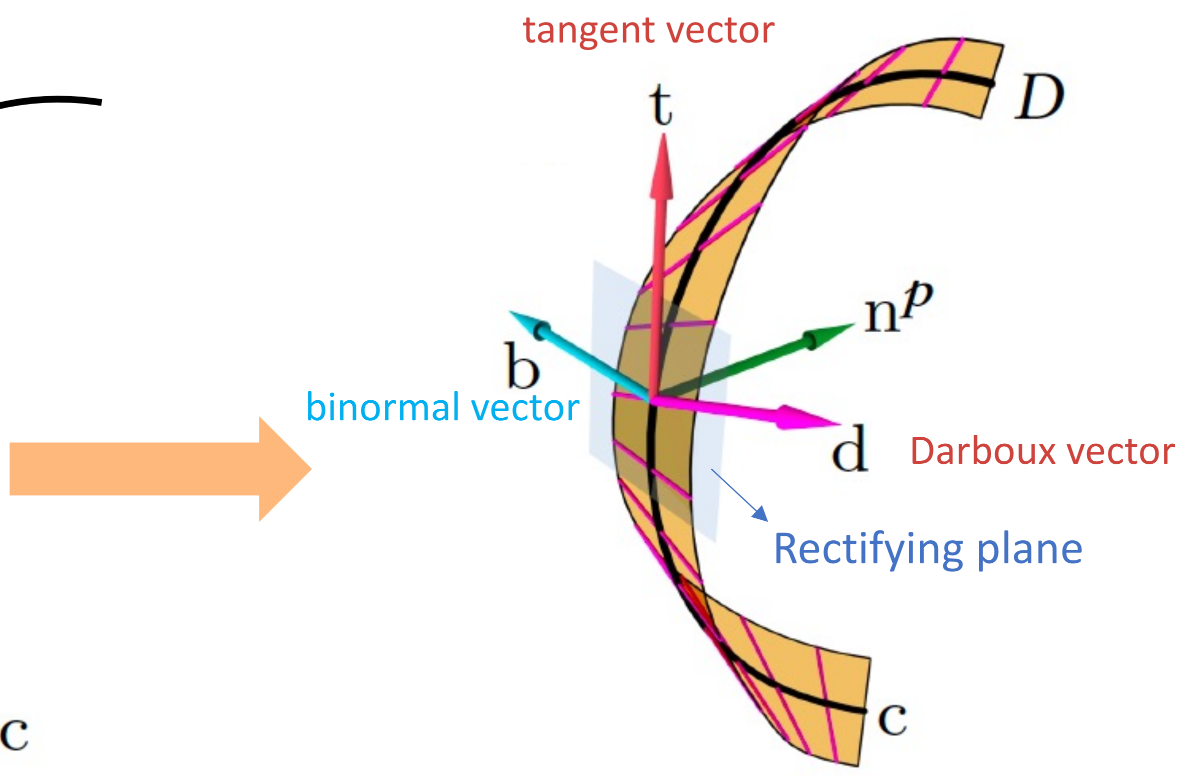
Rectifying Strips in Differential Geometry



Straight Flat Strip

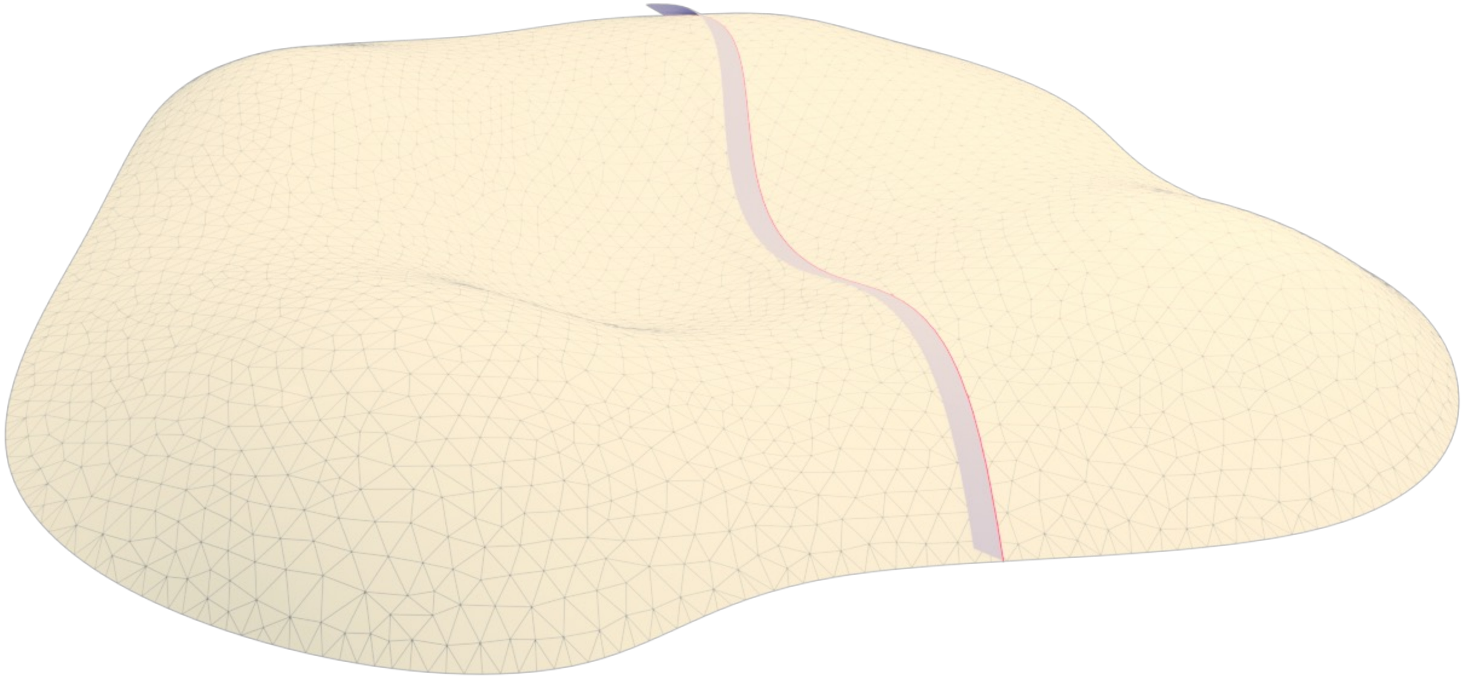


Arbitrary Curve

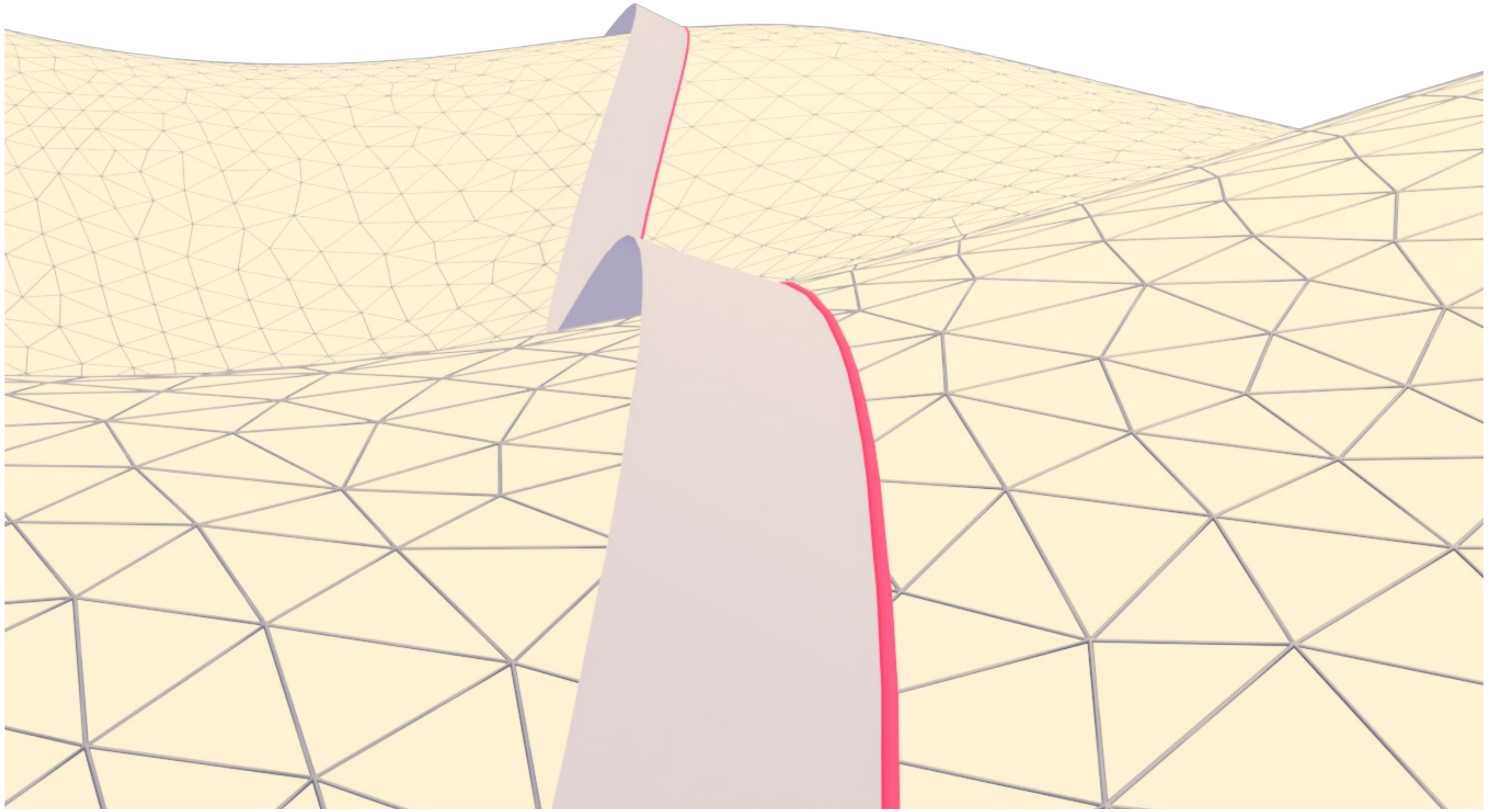


Rectifying Strip

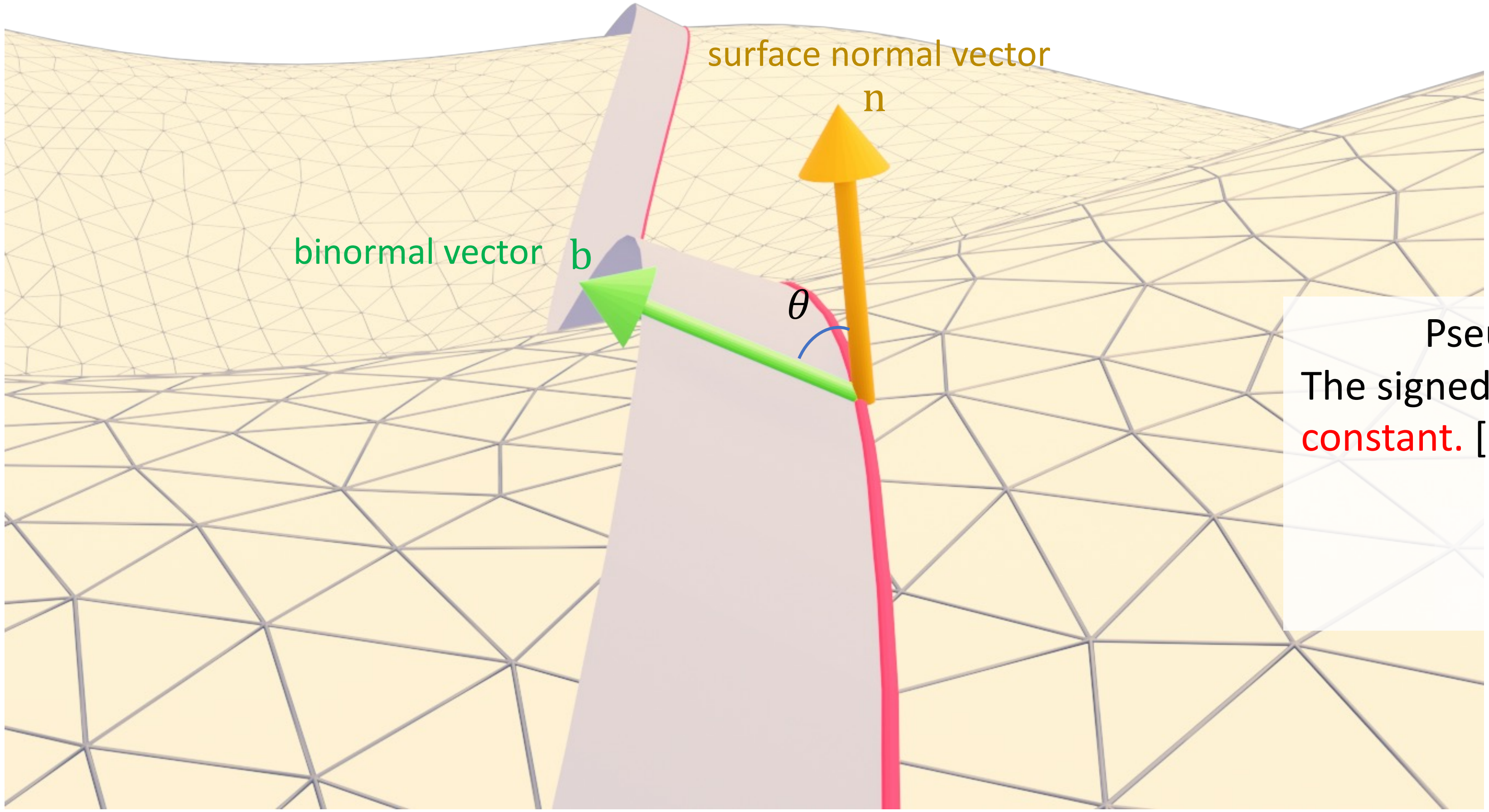
Attaching Rectifying Strips on the Surface



Attaching Rectifying Strips on the Surface



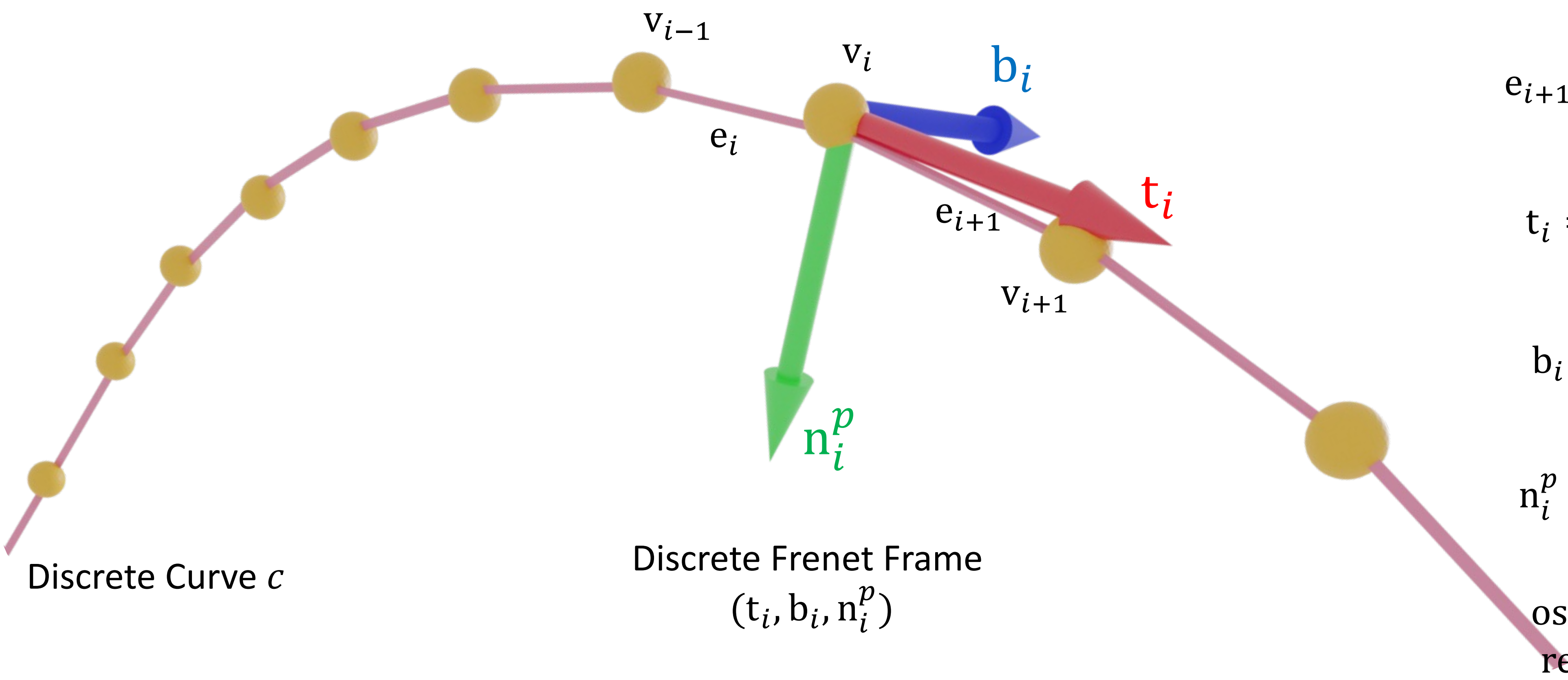
Attaching Rectifying Strips on the Surface



Pseudo-geodesic curves:
The signed angle θ between b and n is constant. [*W. Wunderlich, 1950*]

$\theta = 90°$, geodesic
$\theta = 0°$, asymptotic

Discrete Frenet Frame



$$\mathrm{e_i} = \frac{\mathrm{v}_i - \mathrm{v}_{i-1}}{||\mathrm{v}_i - \mathrm{v}_{i-1}||},$$

$$\mathrm{e}_{i+1} = \frac{\mathrm{v}_{i+1} - \mathrm{v}_i}{||\mathrm{v}_{i+1} - \mathrm{v}_i||},$$

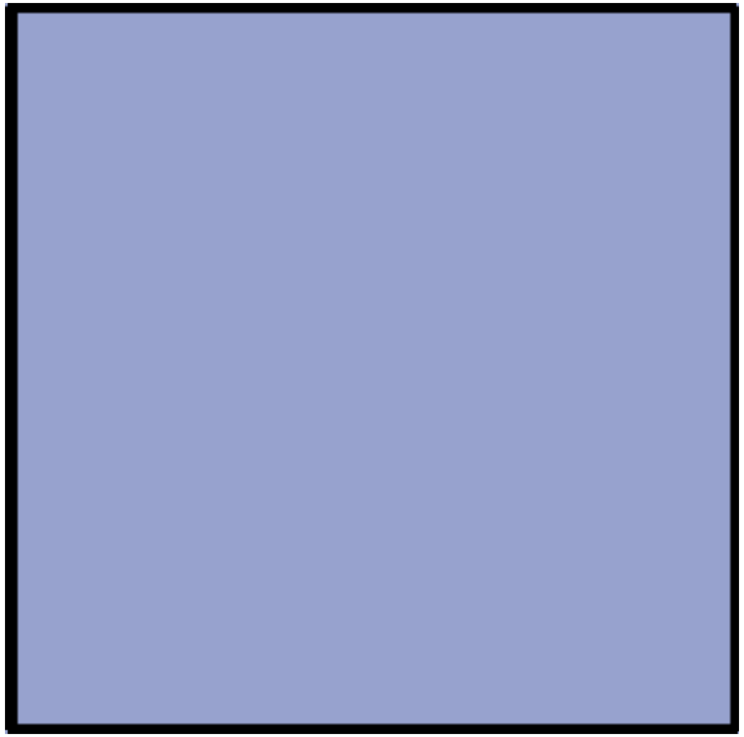
$$\mathrm{t}_i = \frac{\mathrm{e}_i + \mathrm{e}_{i+1}}{||\mathrm{e}_i + \mathrm{e}_{i+1}||},$$

$$\mathrm{b}_i = \frac{\mathrm{e}_{i+1} \times \mathrm{e}_i}{||\mathrm{e}_{i+1} \times \mathrm{e}_i||},$$

$$\mathrm{n}_i^p = \frac{\mathrm{e}_{i+1} - \mathrm{e}_i}{||\mathrm{e}_{i+1} - \mathrm{e}_i||},$$

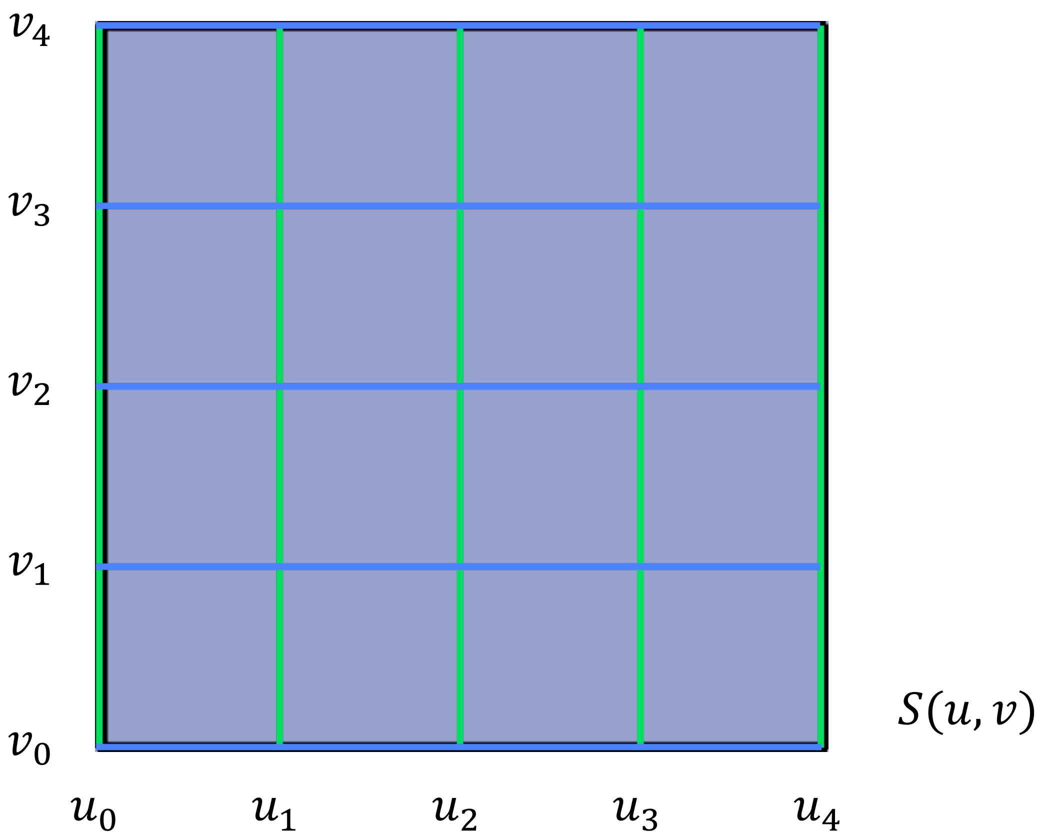
osculating plane,
rectifying plane

Geometry of Gridshells

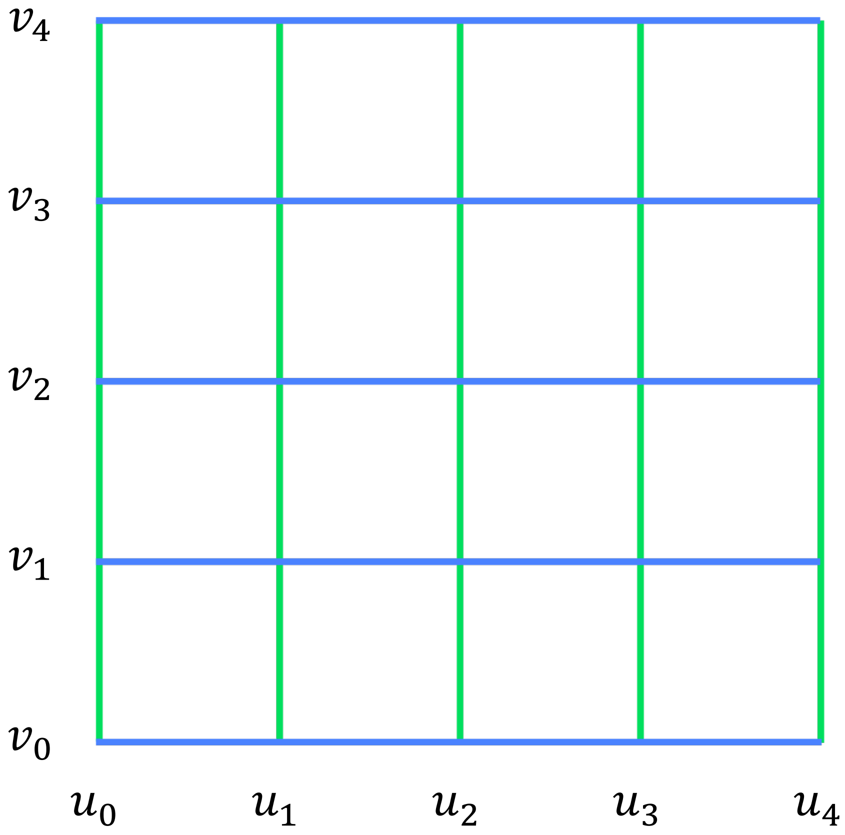


$S(u, v)$

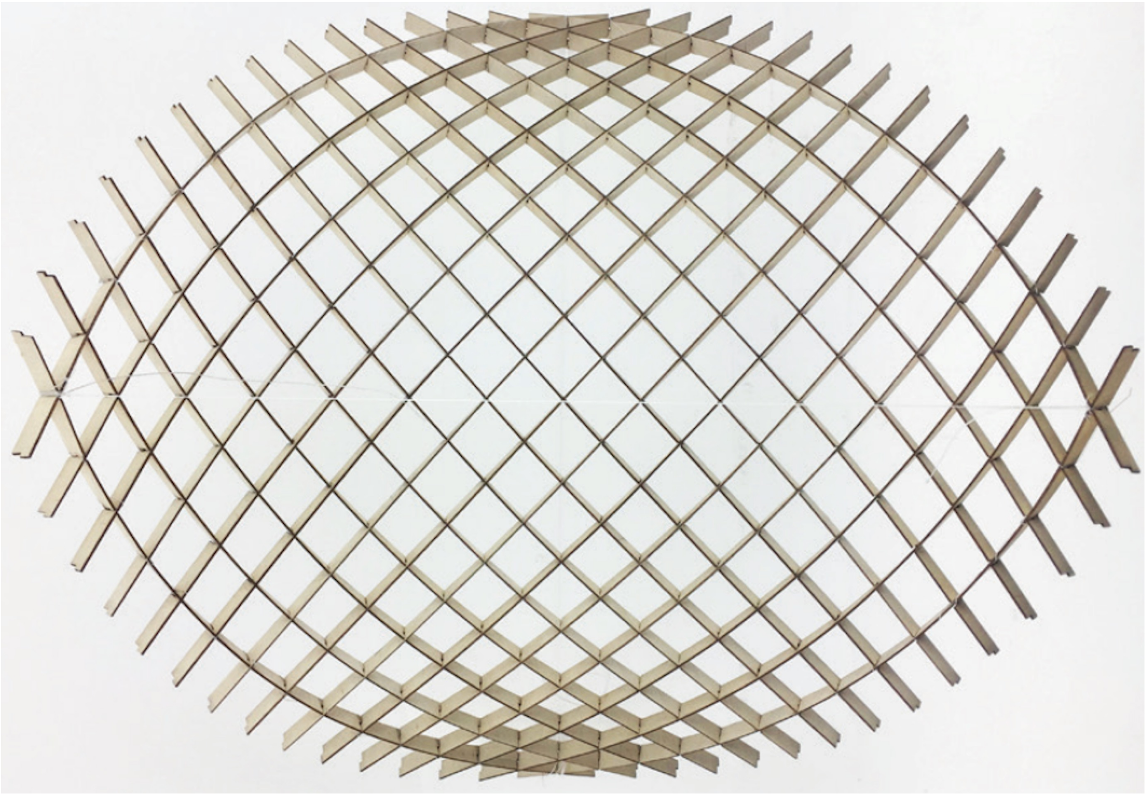
Geometry of Gridshells



Geometry of Gridshells

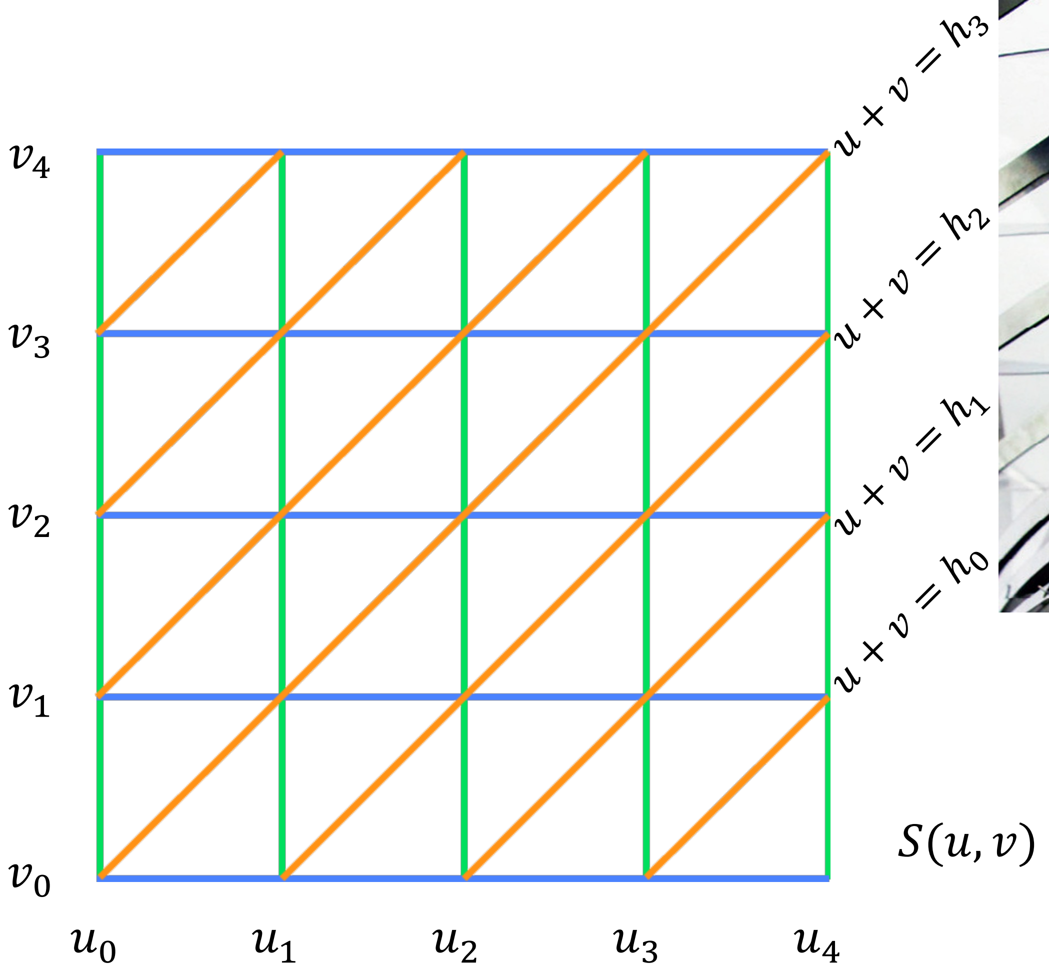


Net



$S(u,v)$

Geometry of Gridshells

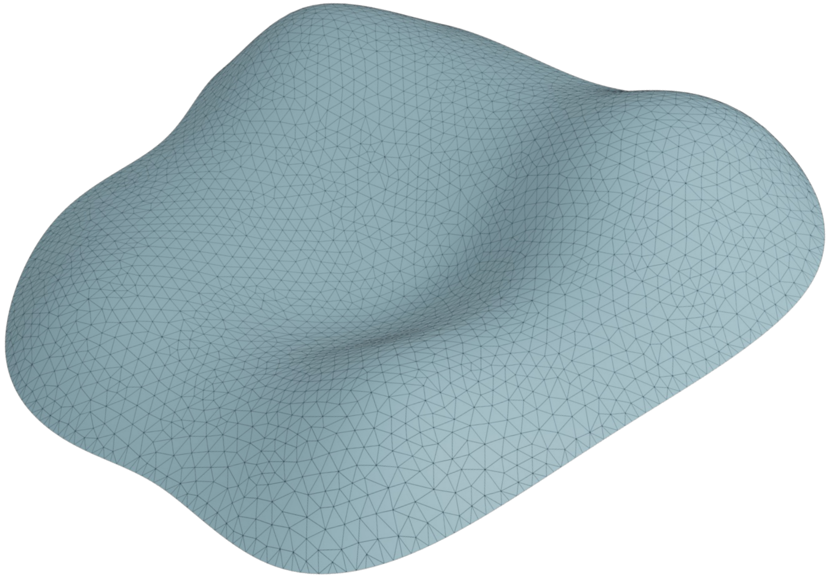


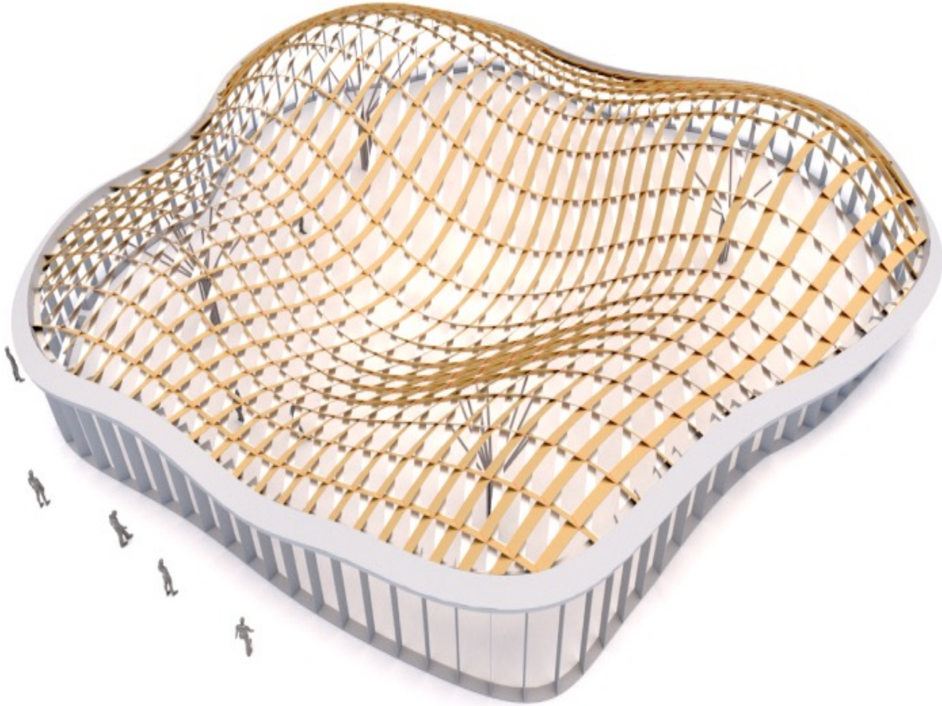
3-Web

Geometry of webs [Blaschke and Bol 1938]: The 3-web on S can be represented by the level-sets of 3 functions F, G and H that satisfy

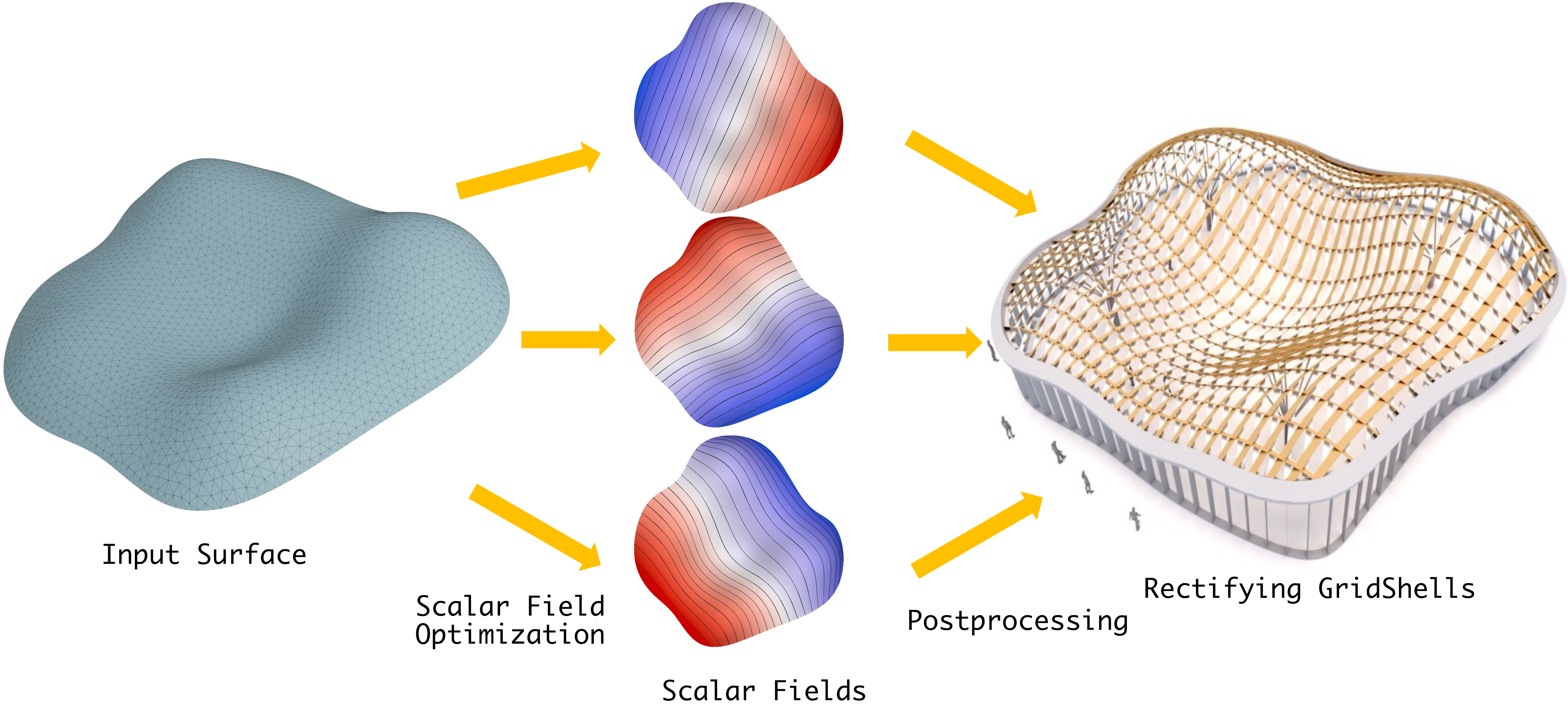
$$F + G + H = 0$$

Motivations: Gridshell Structures

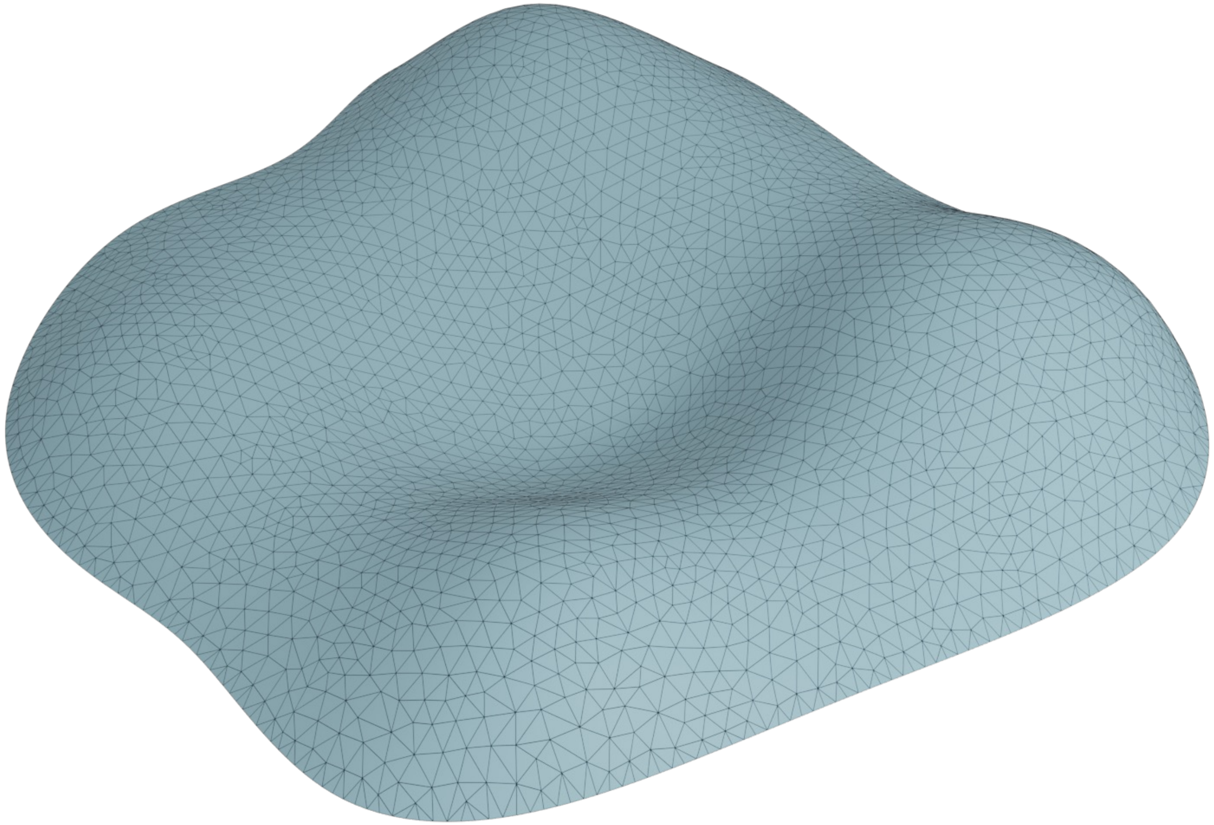




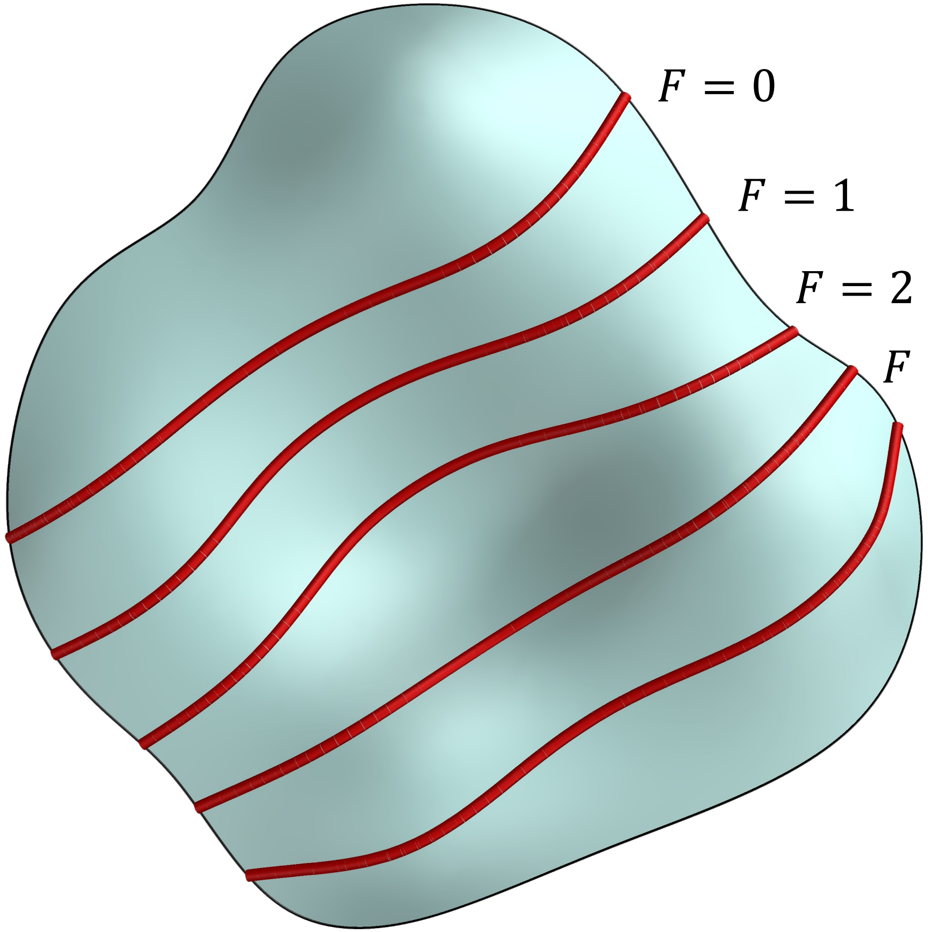
Method: A Level-Set Based Framework



Initialization



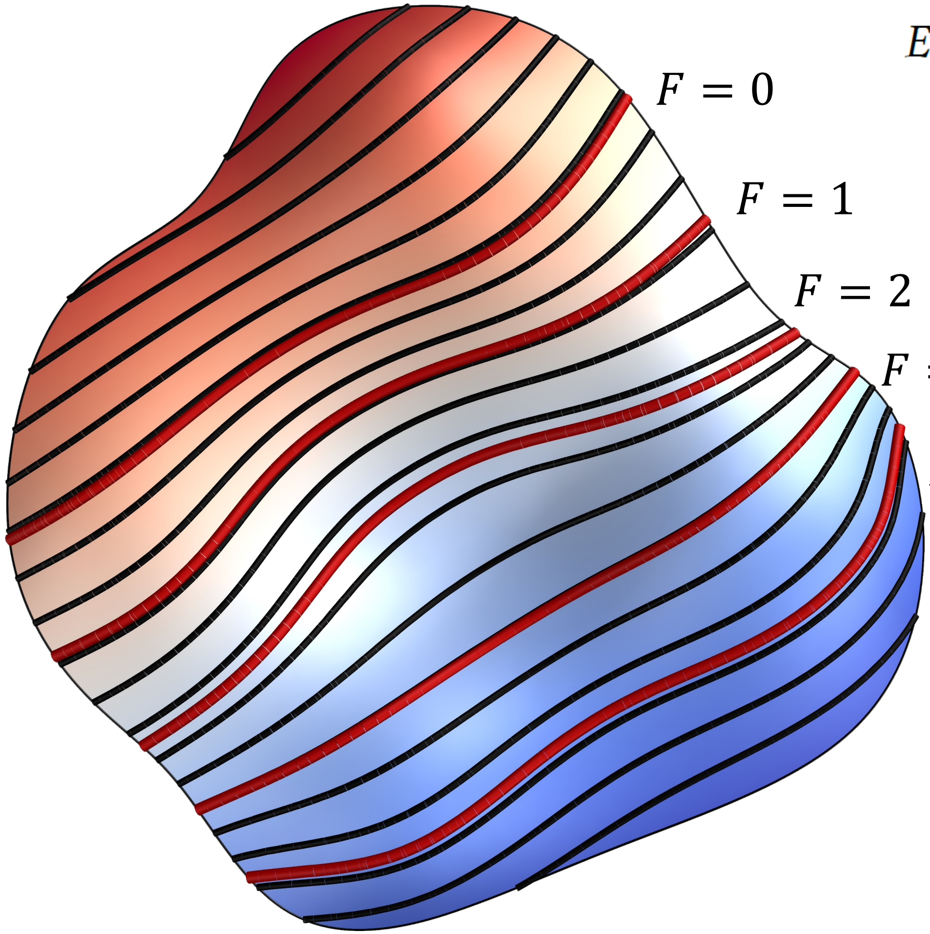
Initialization



A robust version of [Jiang et al. 2019]'s tracing algorithm.

```
Assign function values for
the curves
```

Initialization



$$E_{\text{trace}} = \sum_{c^i \in C_t} \sum_{p \in c^i} \left(\frac{F^i - F_0}{F_1 - F_0} - \frac{\|p - v_0\|}{\|v_1 - v_0\|} \right)^2 .$$ (Linear interpolation)

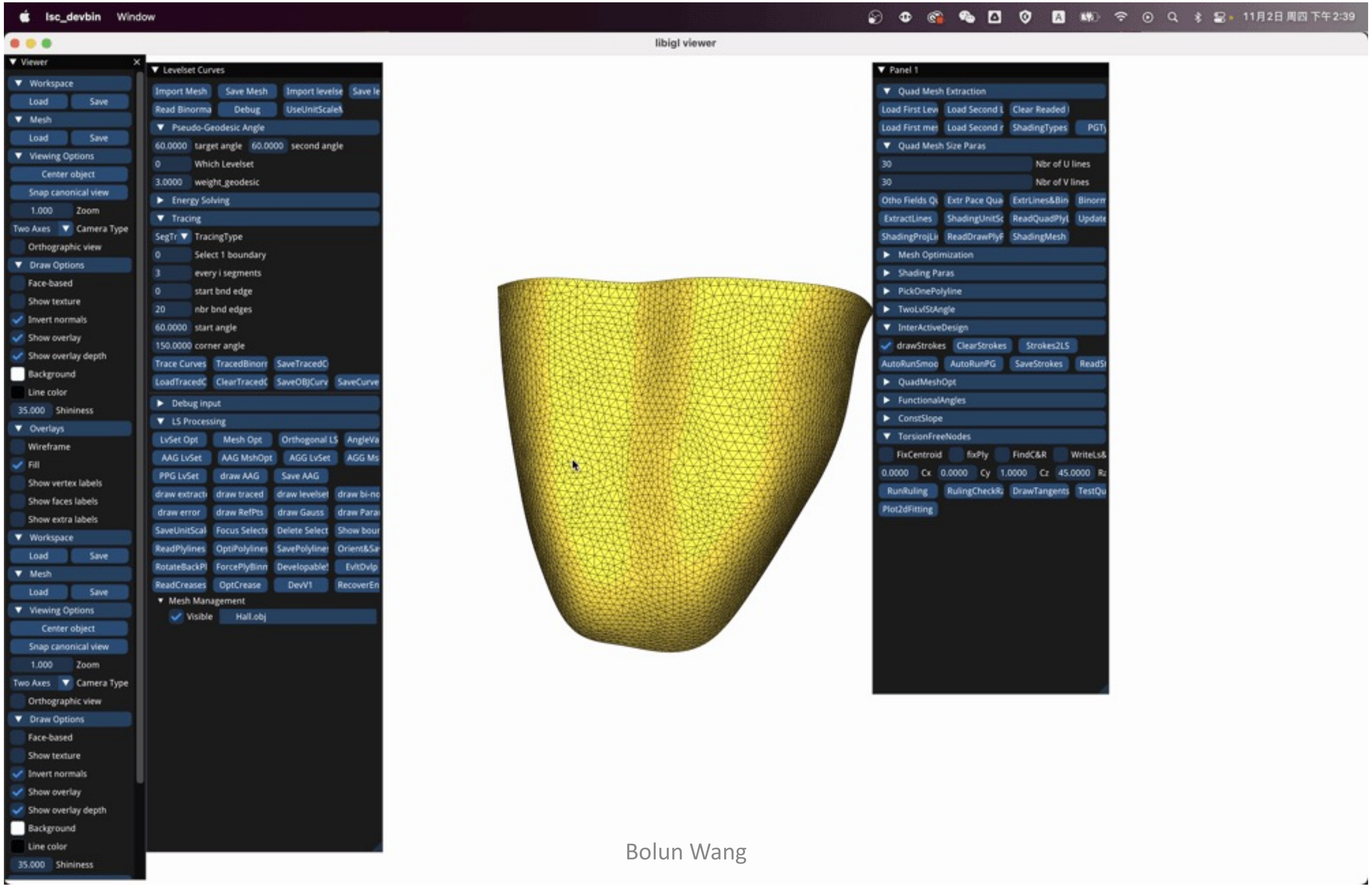
$$E_{\text{fair}} = \sum_{v \in \mathcal{V}} \|H(v)\|^2 \mathcal{A}(v).$$ (Fairness)

$$\min E_{\text{init}} = \lambda_0 E_{\text{trace}} + \lambda_1 E_{\text{fair}},$$

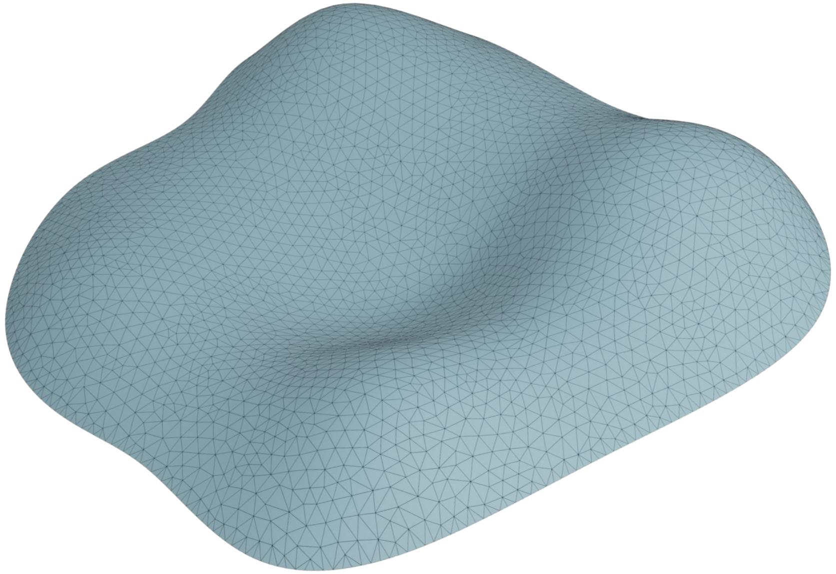
Optimize the level sets to fit the curves.

Initialization

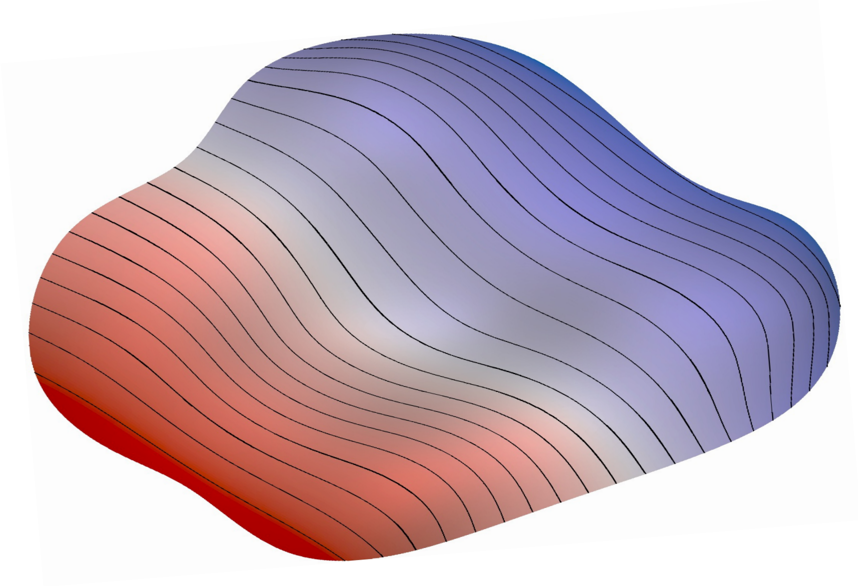
An Optional Initialization: An Interactive Method:



Optimizing Pseudo-Geodesics



Angle
Constraints



Input Surface + Target Angle θ

Pseudo-Geodesics of Angle θ

Optimizing Pseudo-Geodesics

~~Control the Inclination Angles: Optimizing Curvatures~~

The normal curvature and geodesic curvature[Pottmann et. al., 2010]

Second fundamental form

$$\kappa_n = \frac{\mathrm{II}(J\nabla F)}{||\nabla F||^2}, \quad \kappa_g = \operatorname{div}\left(\frac{\nabla F}{||\nabla F||}\right),$$

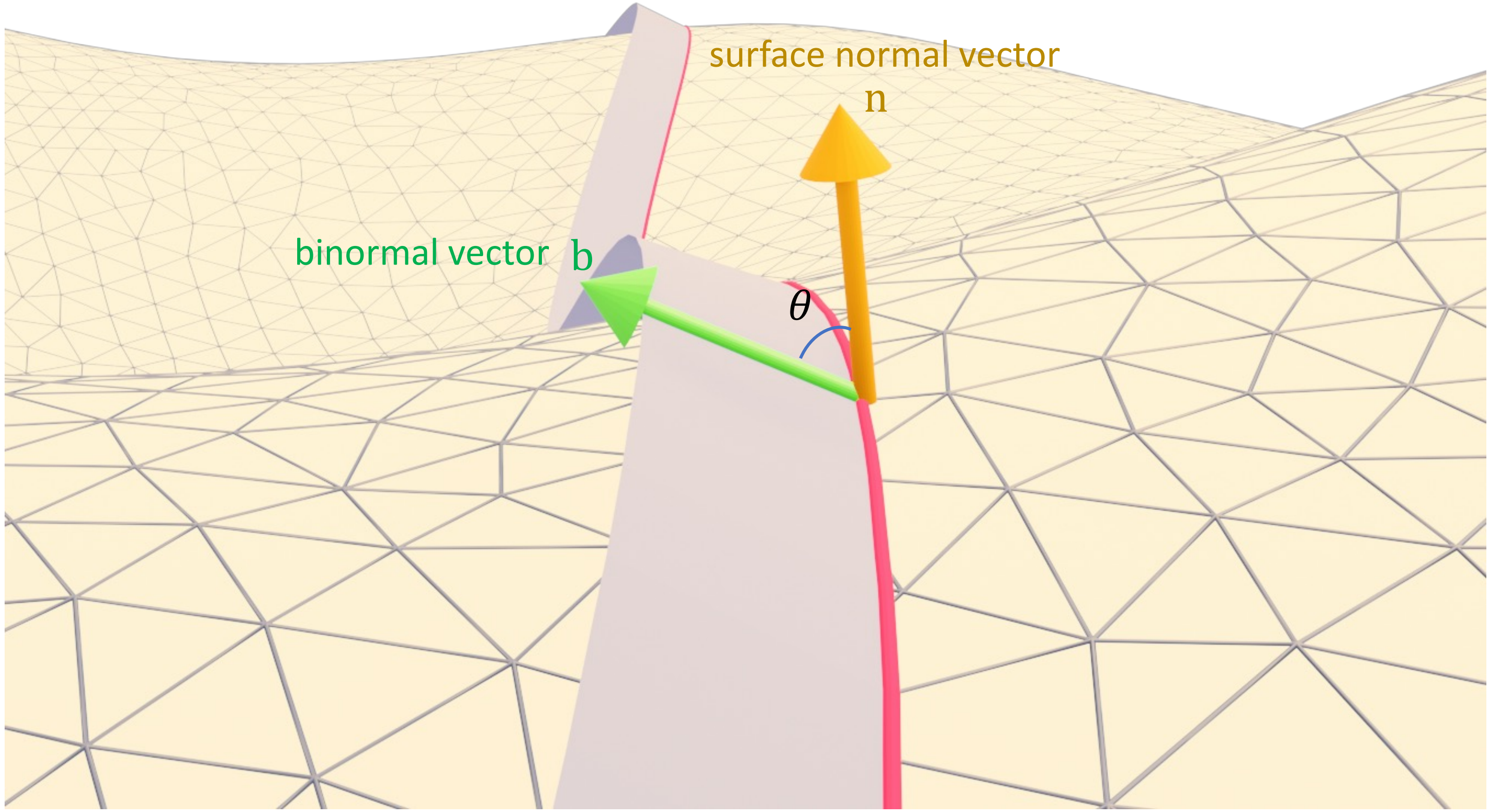
Rotate by 90°

$$\theta = \operatorname{acot}(\kappa_n/\kappa_g)$$

Need to compute I and II, and high-order derivatives.

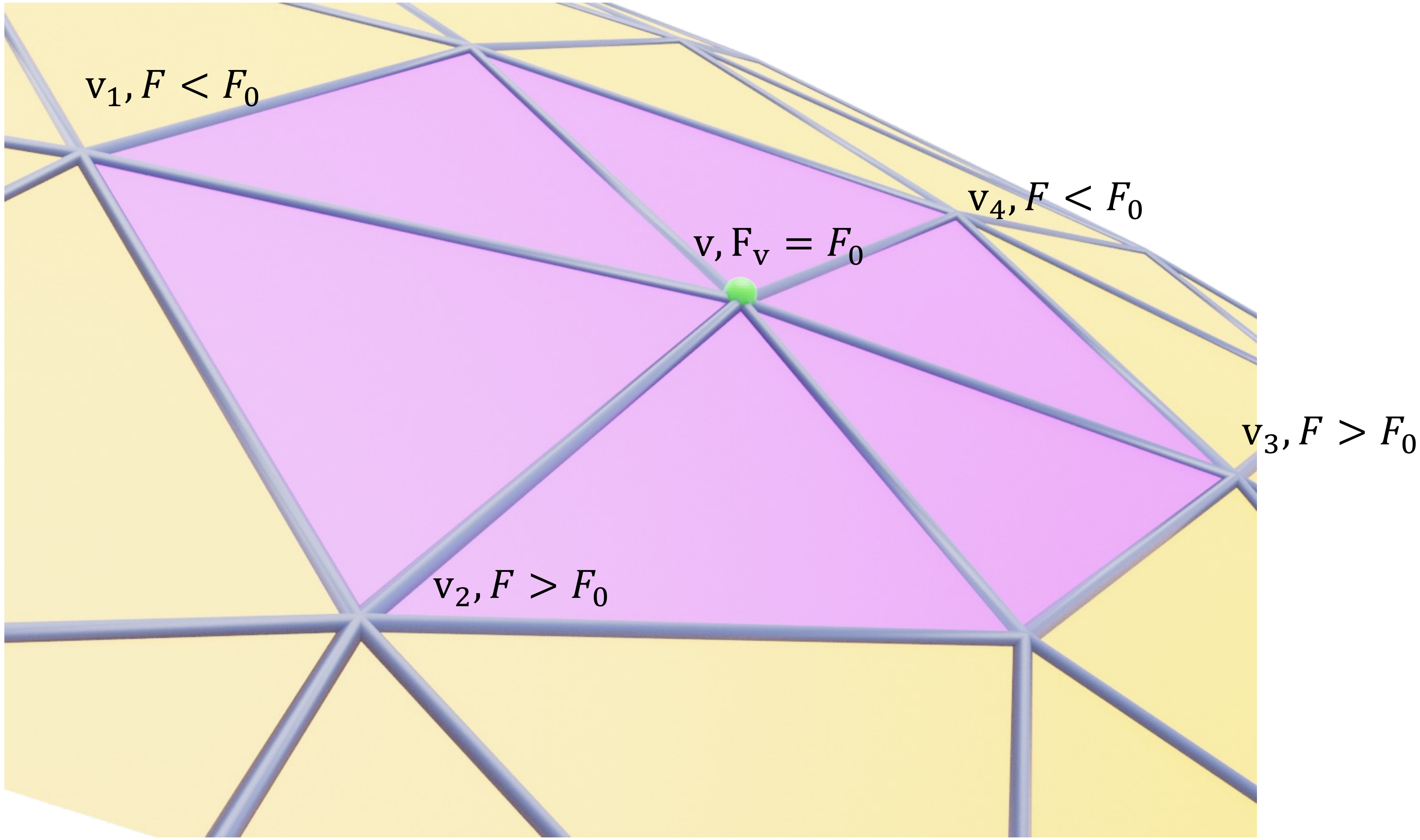
Optimizing Pseudo-Geodesics

Control the Inclination Angles: Controlling the Binormals on Each Vertex Star



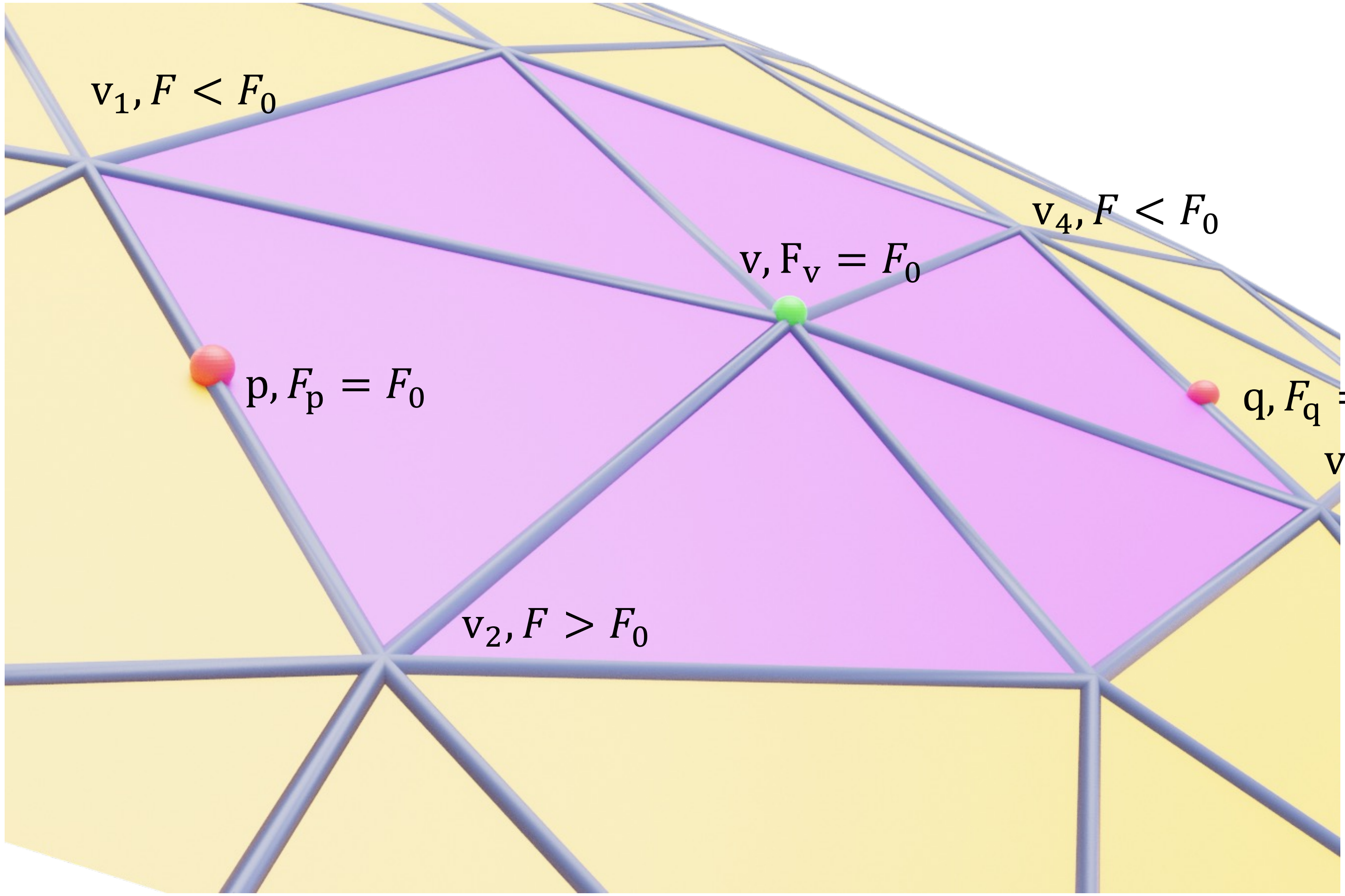
Optimizing Pseudo-Geodesics

Control the Inclination Angles: Controlling the Binormals on Each Vertex Star



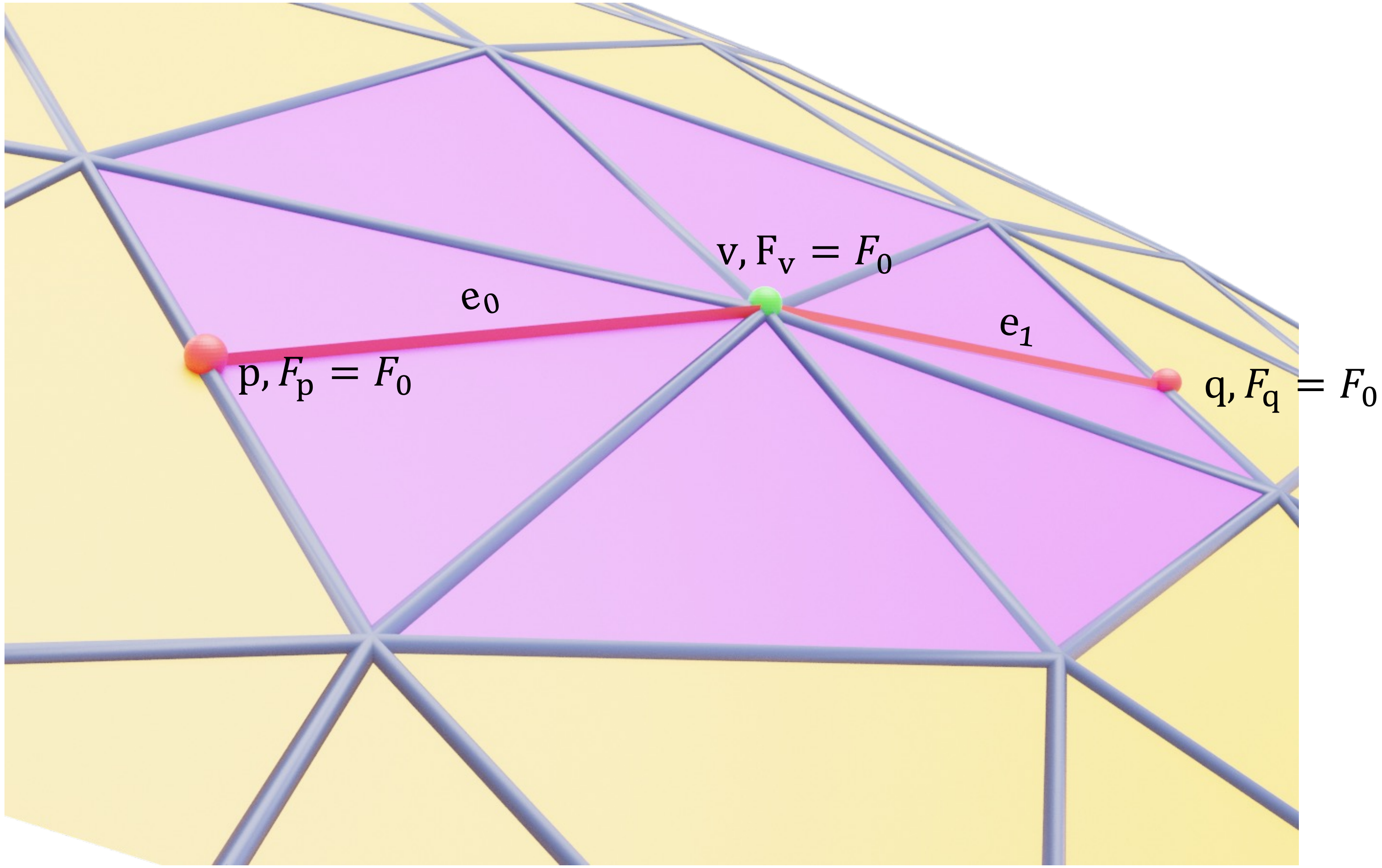
Optimizing Pseudo-Geodesics

Control the Inclination Angles: Controlling the Binormals on Each Vertex Star



Optimizing Pseudo-Geodesics

Control the Inclination Angles: Controlling the Binormals on Each Vertex Star

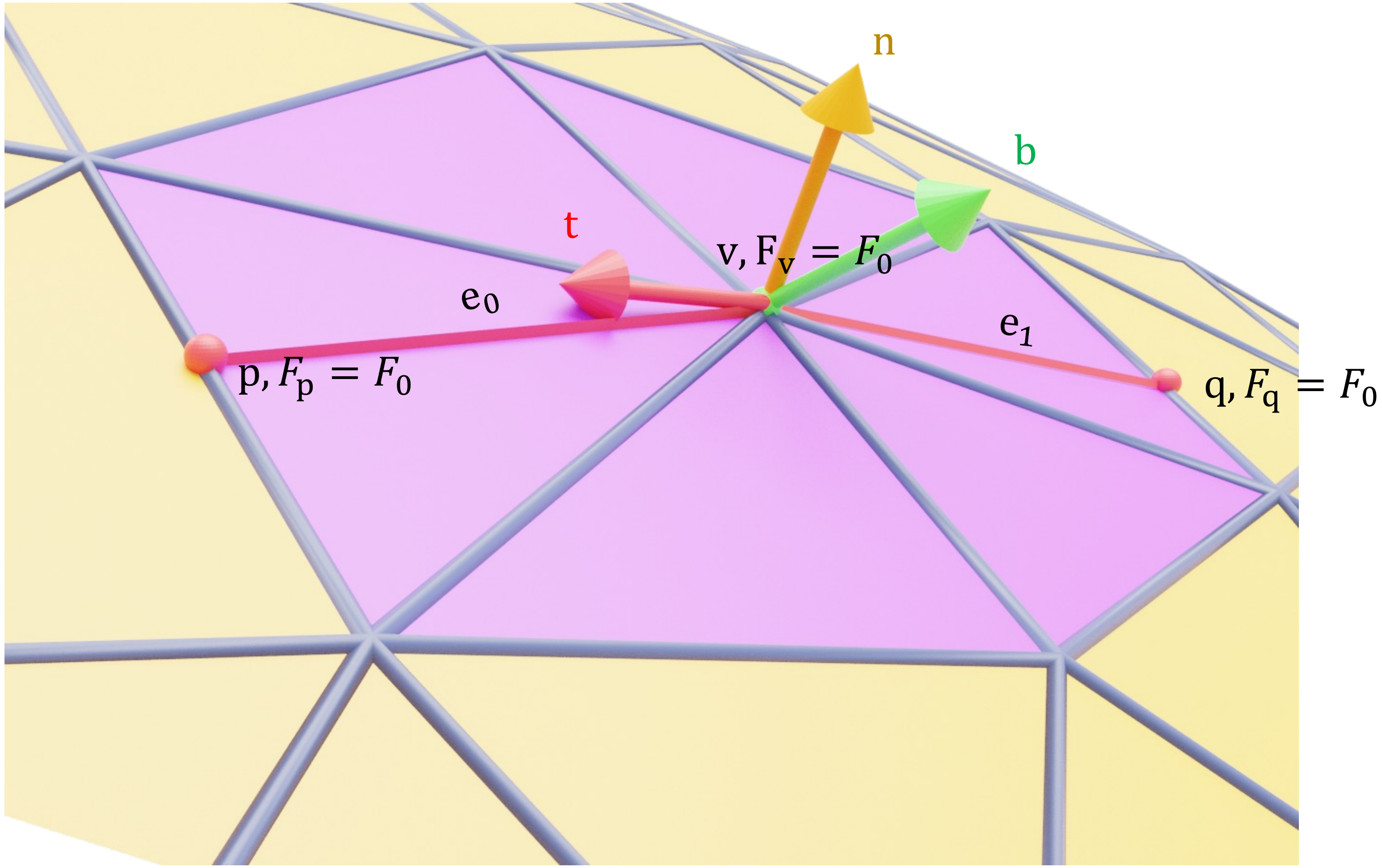


$$\mathrm{e}_0 = \frac{\mathrm{p} - \mathrm{v}}{||\mathrm{p} - \mathrm{v}||},$$

$$\mathrm{e}_1 = \frac{\mathrm{v} - \mathrm{q}}{||\mathrm{v} - \mathrm{q}||},$$

Optimizing Pseudo-Geodesics

Control the Inclination Angles: Controlling the Binormals on Each Vertex Star



$$e_0 = \frac{p - v}{||p - v||},$$

$$e_1 = \frac{v - q}{||v - q||},$$

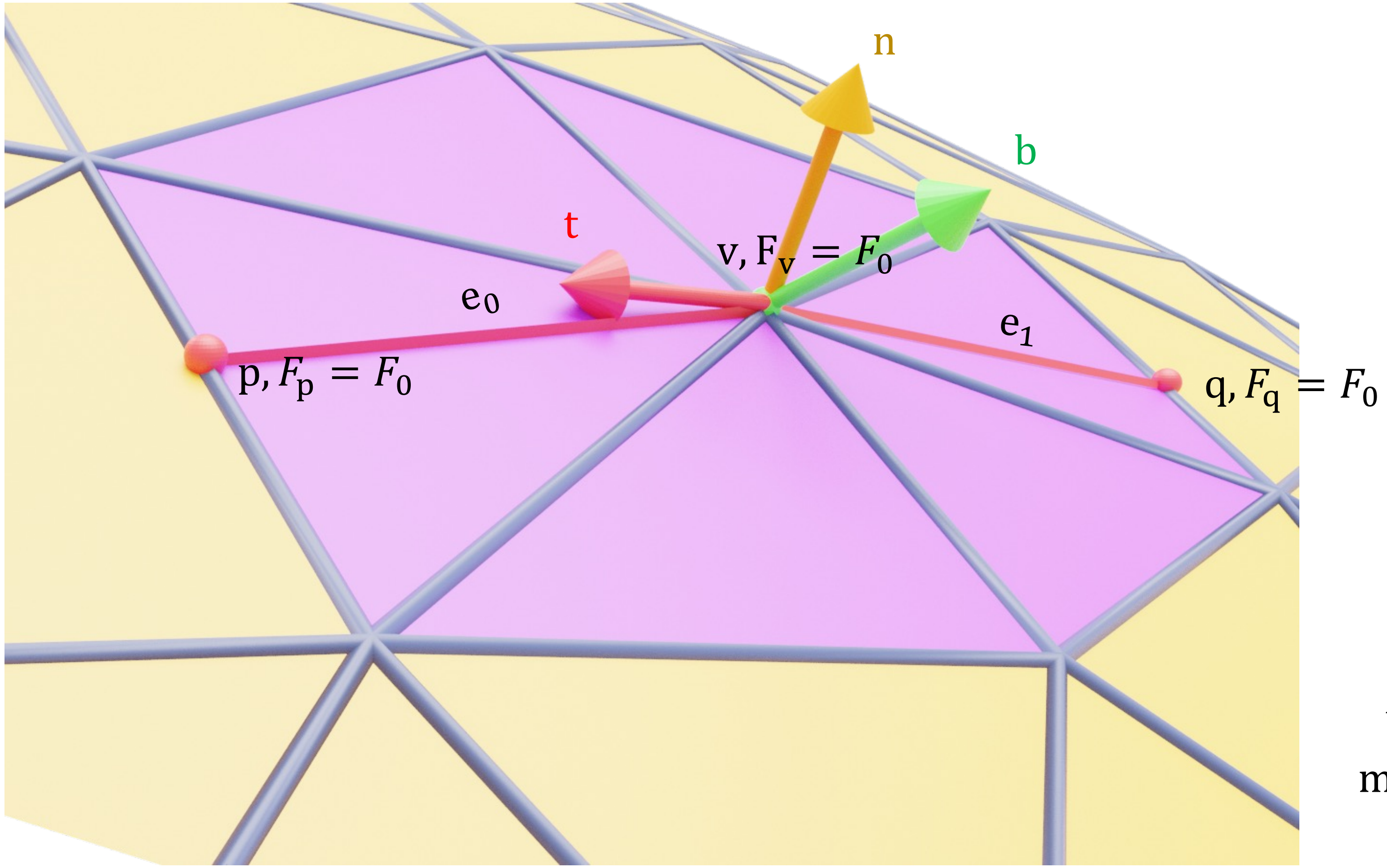
$$b = \frac{e_0 \times e_1}{||e_0 \times e_1||}$$

Angle Constraint:

$$b \cdot n - \cos\theta = 0$$

Optimizing Pseudo-Geodesics

Control the Inclination Angles: Controlling the Binormals on Each Vertex Star



$$e_0 = \frac{p - v}{||p - v||},$$

$$e_1 = \frac{v - q}{||v - q||},$$

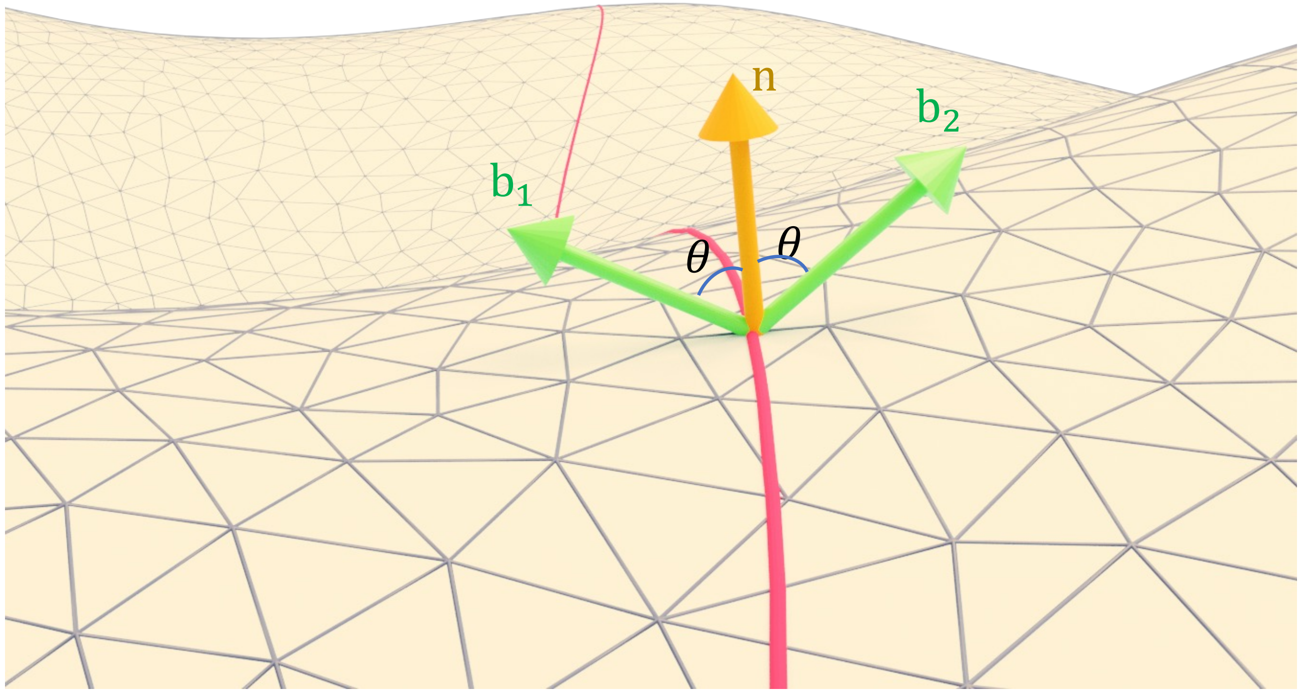
$$b = \frac{e_0 \times e_1}{||e_0 \times e_1||}$$

Angle Constraint:

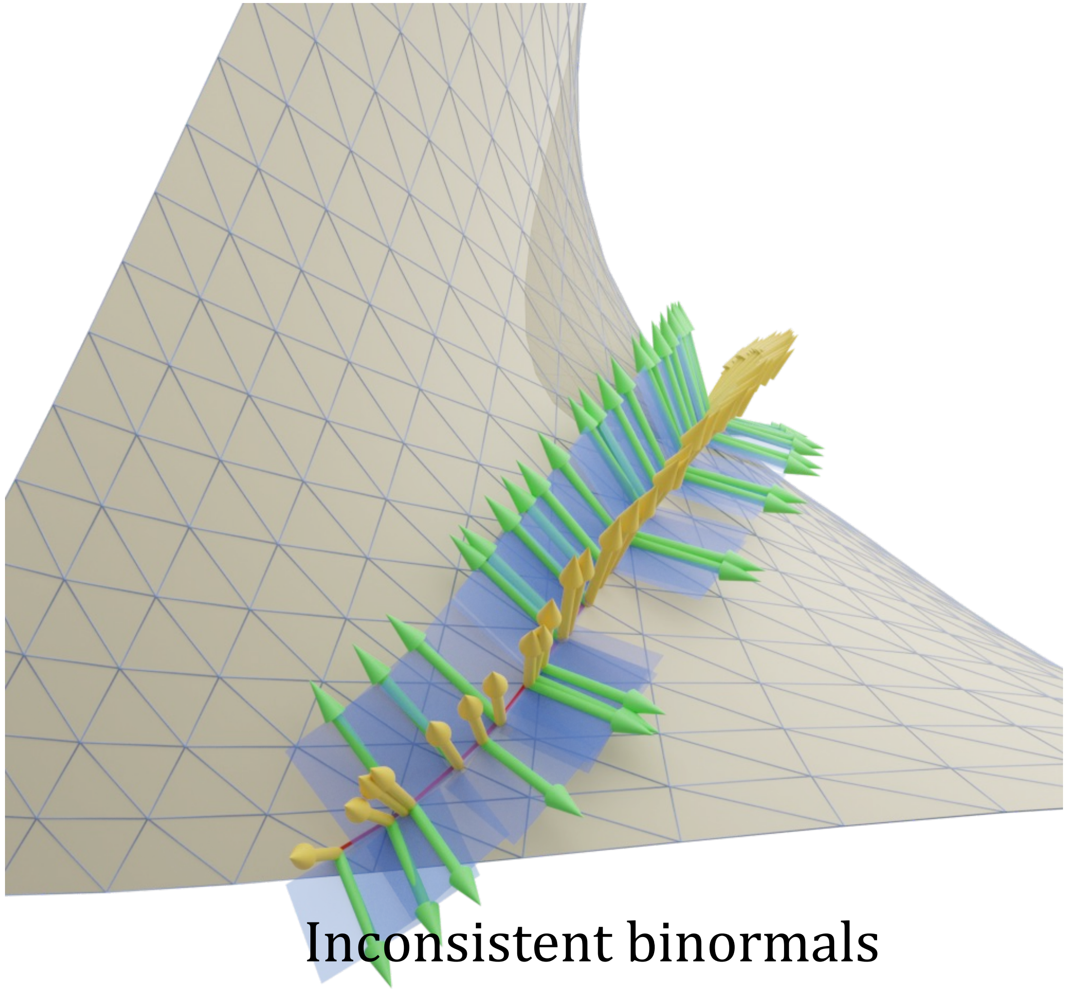
$$\min\,(b \cdot n - \cos\theta)^2$$

Optimizing Pseudo-Geodesics

Control the Inclination Angles: Controlling the Binormal Vectors



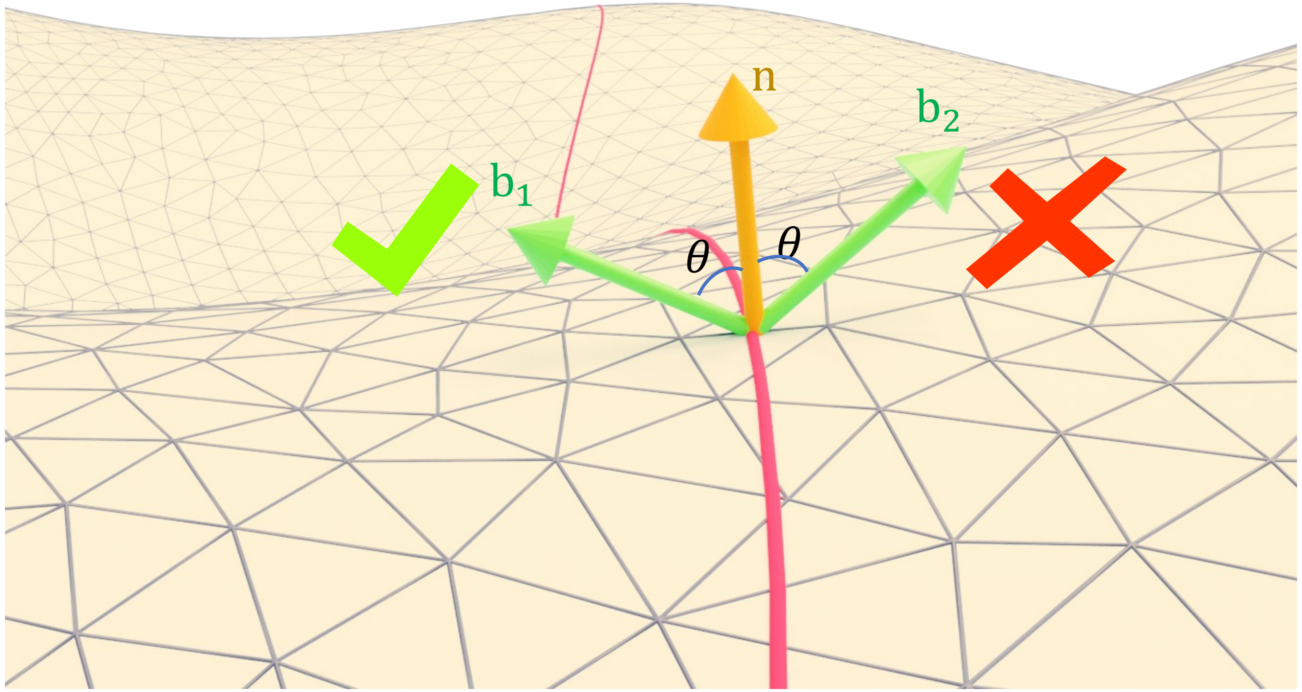
$b \cdot n - \cos\theta = 0$ cannot distinguish b_1 from b_2

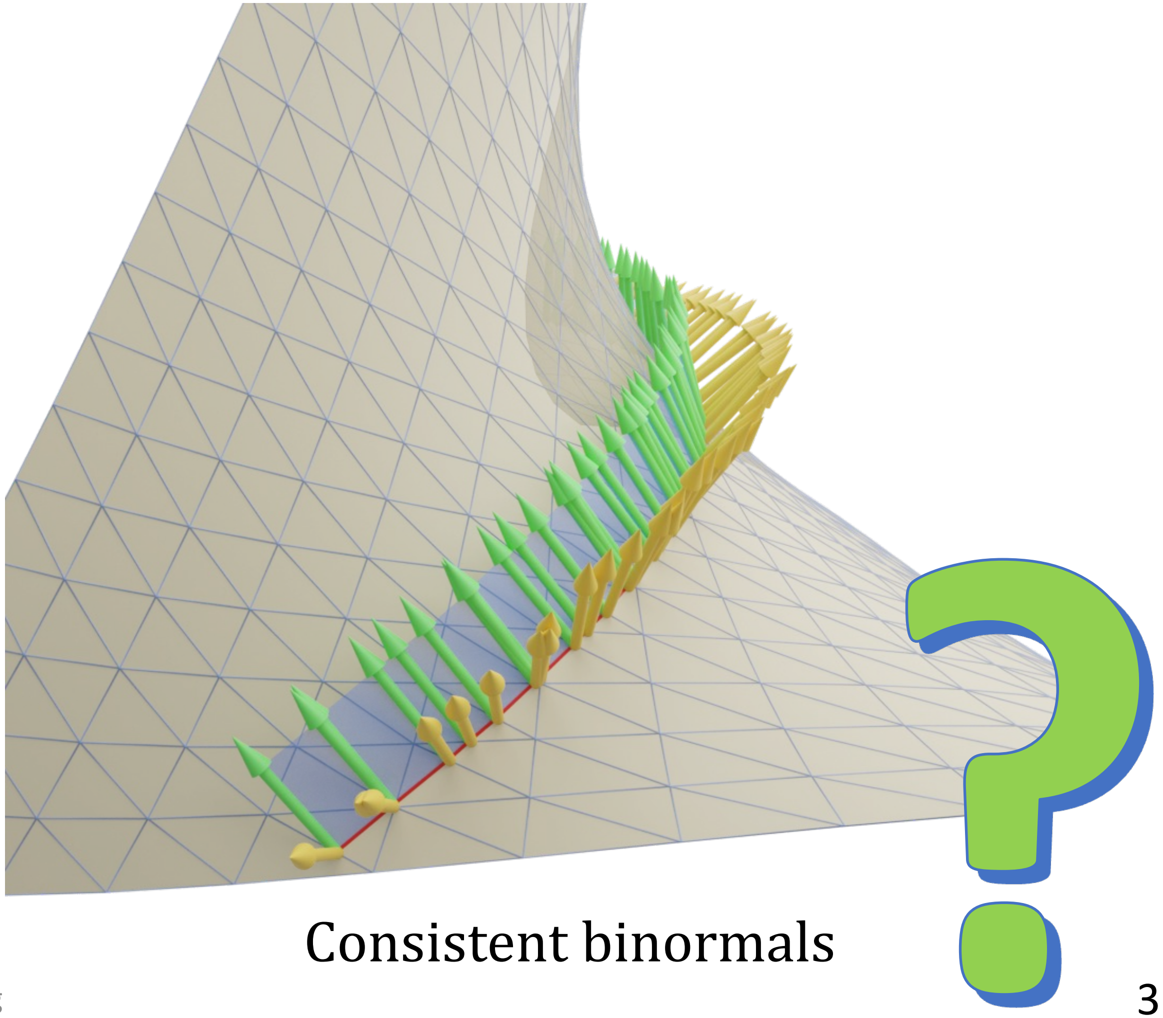


Inconsistent binormals

Optimizing Pseudo-Geodesics

Control the Inclination Angles: Controlling the Binormal Vectors



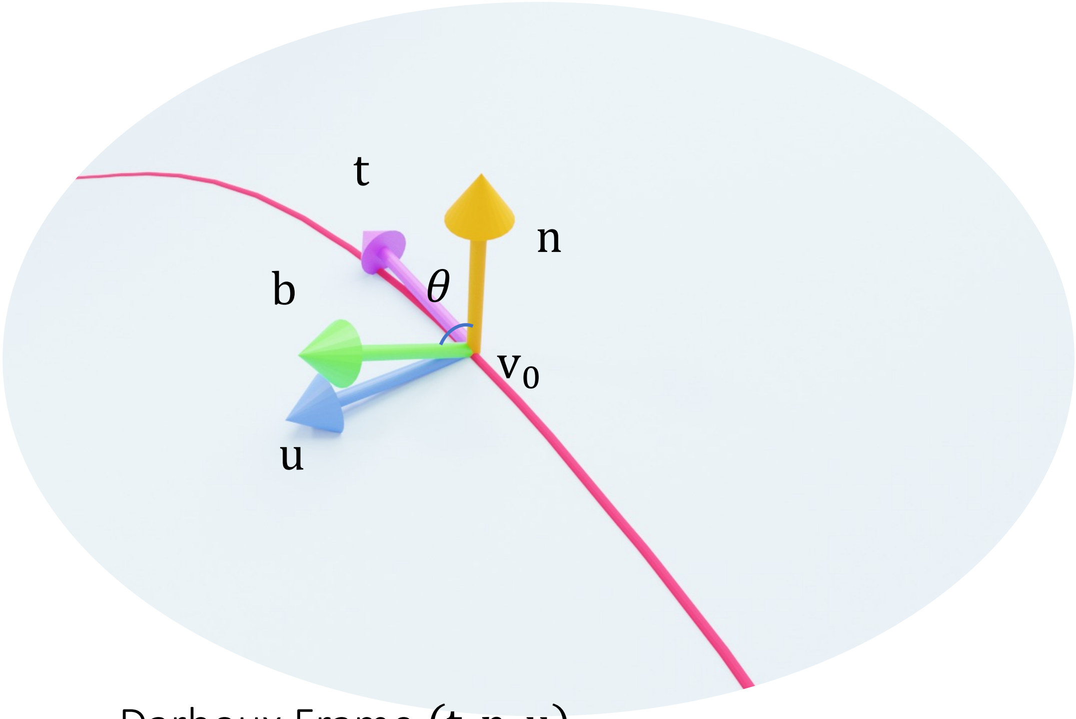


$b \cdot n - \cos\theta = 0$ cannot distinguish b_1 from b_2

Consistent binormals

Optimizing Pseudo-Geodesics

Control the Inclination Angles: Controlling the Binormal Vectors

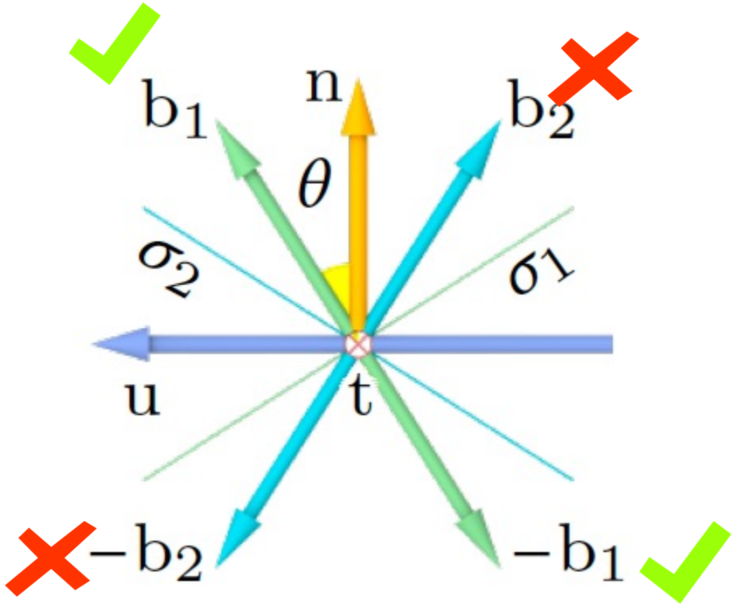


Darboux Frame $(\mathrm{t}, \mathrm{n}, \mathrm{u})$

- Unit tangent vector u,
- Unit normal vector n,
- Side vector $\mathrm{u} = \mathrm{n} \times \mathrm{t}$

The angle constraints:

$$\mathrm{b} \cdot \mathrm{n} - \cos\theta = 0 \Rightarrow \begin{aligned} (\mathrm{b} \cdot \mathrm{n})^2 - \cos^2\theta &= 0, \\ (\mathrm{b} \cdot \mathrm{n})(\mathrm{b} \cdot \mathrm{u}) - \cos\theta\sin\theta &= 0, \end{aligned}$$



Optimizing Pseudo-Geodesics

(Angle Constraints) $$E_{\text{angle}} = \sum_{v \in \mathcal{V}} ((b \cdot n)^2 - \cos^2\theta)^2 \mathcal{A}(v) + \sum_{v \in \mathcal{V}} ((b \cdot n)(b \cdot u) - \sin\theta\cos\theta)^2 \mathcal{A}(v),$$

(Preventing Vanishing Gradients) $$E_{\text{grad}} = \sum_{f \in \mathcal{F}} (\|\nabla F(f)\| - r)^2 \mathcal{A}(f),$$

(fairness) $$E_{\text{fair}} = \sum_{v \in \mathcal{V}} \|H(v)\|^2 \mathcal{A}(v).$$

$$\min E_{\text{pg}} = \lambda_{\text{fair}} E_{\text{fair}} + \lambda_{\text{grad}} E_{\text{grad}} + \lambda_{\text{angle}} E_{\text{angle}}.$$

- Why containing the area $\mathcal{A}(v)$ of the Voronoi cell?

  The error $E_{\text{angle}} = \int (Error\ on\ Point) d\mathcal{A}$

- Two simplified constraints for asymptotics and geodesics

$$E_{\text{geo}} = \sum_{v \in \mathcal{V}} \left(\frac{\det(\mathbf{n}, \mathbf{v}_0 - \mathbf{p}, \mathbf{v}_0 - \mathbf{q})}{\|\mathbf{v}_0 - \mathbf{p}\| \|\mathbf{v}_0 - \mathbf{q}\|} \right)^2 \mathcal{A}(\mathbf{v}),$$

$$E_{\text{asy}} = \sum_{v \in \mathcal{V}} \left(\left(\frac{\mathbf{n} \cdot (\mathbf{v}_0 - \mathbf{p})}{\|\mathbf{v}_0 - \mathbf{p}\|} \right)^2 + \left(\frac{\mathbf{n} \cdot (\mathbf{v}_0 - \mathbf{q})}{\|\mathbf{v}_0 - \mathbf{q}\|} \right)^2 \right) \mathcal{A}(\mathbf{v}).$$

- The E_{grad}:

  Prevent optimization failures, make the curves uniform (Pottmann et.al., 2010, Geodesic Patterns)

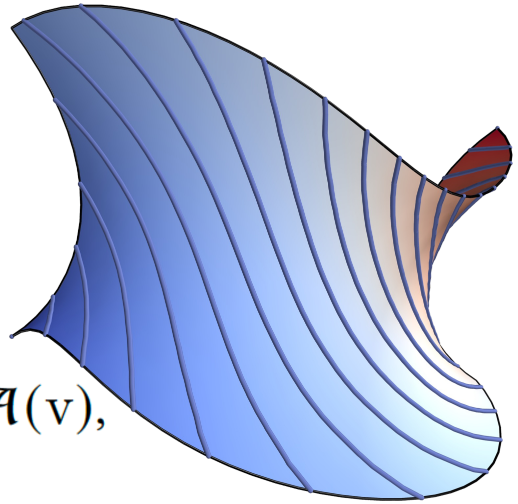
Optimizing Pseudo-Geodesics

(Angle Constraints) $$E_{\text{angle}} = \sum_{\mathrm{v}\in\mathcal{V}} ((\mathrm{b}\cdot\mathrm{n})^2 - \cos^2\theta)^2 \mathcal{A}(\mathrm{v}) + \sum_{\mathrm{v}\in\mathcal{V}} ((\mathrm{b}\cdot\mathrm{n})(\mathrm{b}\cdot\mathrm{u}) - \sin\theta\cos\theta)^2 \mathcal{A}(\mathrm{v}),$$

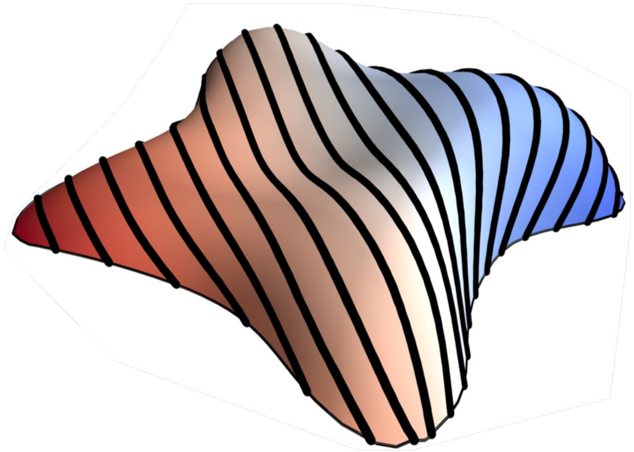
(Preventing Vanishing Gradients) $$E_{\text{grad}} = \sum_{\mathrm{f}\in\mathcal{F}} (\|\nabla F(\mathrm{f})\| - r)^2 \mathcal{A}(\mathrm{f}),$$

(fairness) $$E_{\text{fair}} = \sum_{\mathrm{v}\in\mathcal{V}} \|H(\mathrm{v})\|^2 \mathcal{A}(\mathrm{v}).$$

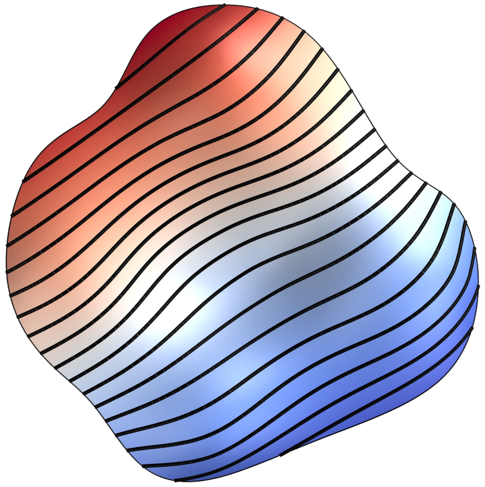
$$\min E_{\text{pg}} = \lambda_{\text{fair}} E_{\text{fair}} + \lambda_{\text{grad}} E_{\text{grad}} + \lambda_{\text{angle}} E_{\text{angle}}.$$



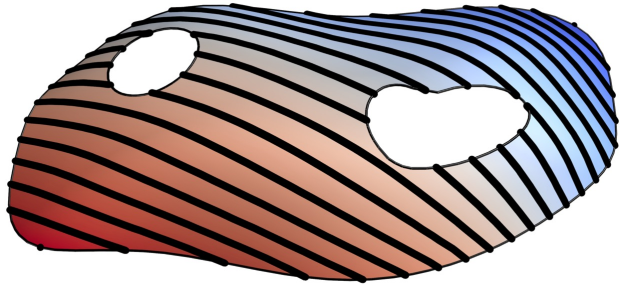
$\theta = 0°$ (Asymptotic)



$\theta = 90°$ (Geodesic)



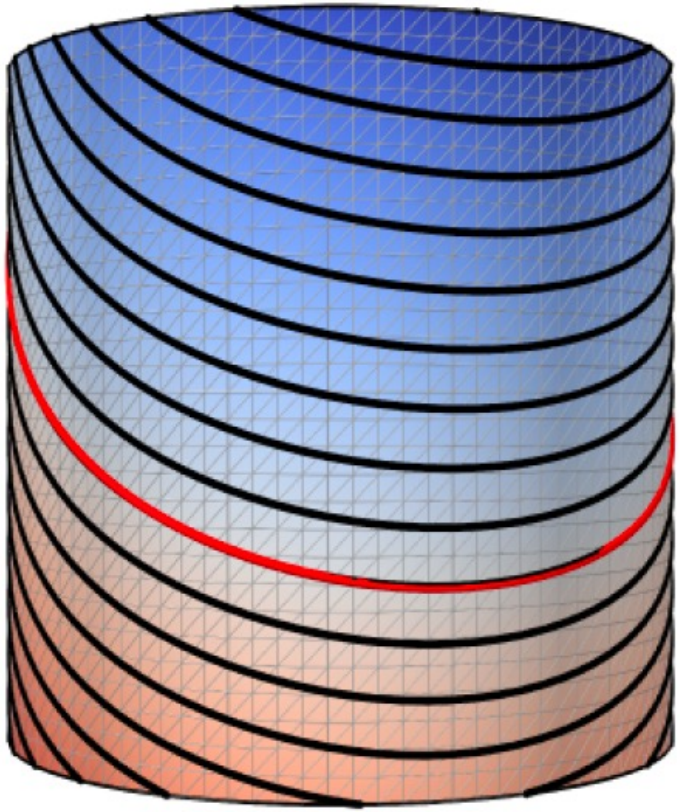
$\theta = 60°$ (Pseudo-Geodesic)



$\theta = 75°$ (Pseudo-Geodesic)

Optimizing Pseudo-Geodesics

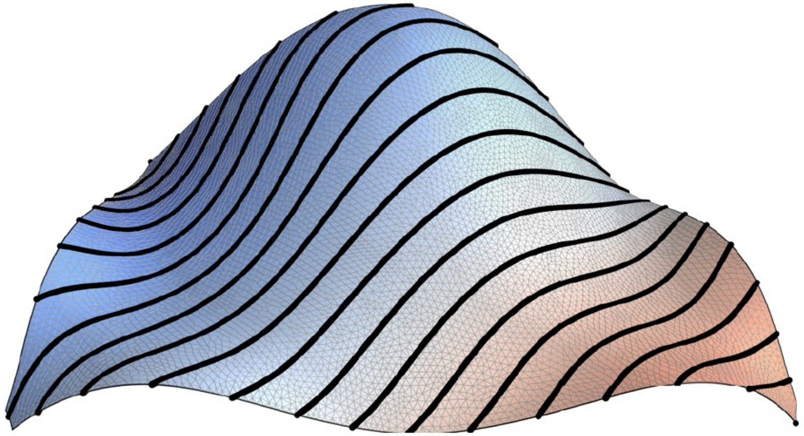
`Validation: comparing with an exact pseudo-geodesic`



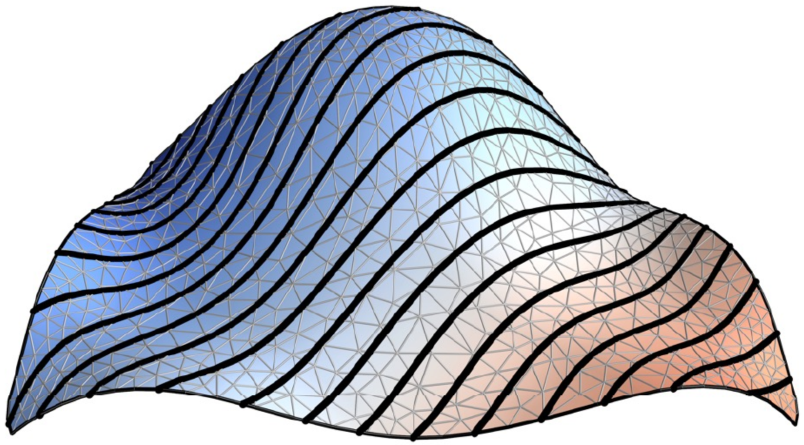
$\theta = 60°$. Red: continuous pseudo-geodesic curve. Black: level sets.
Error: 0.47% of the curve length

Optimizing Pseudo-Geodesics

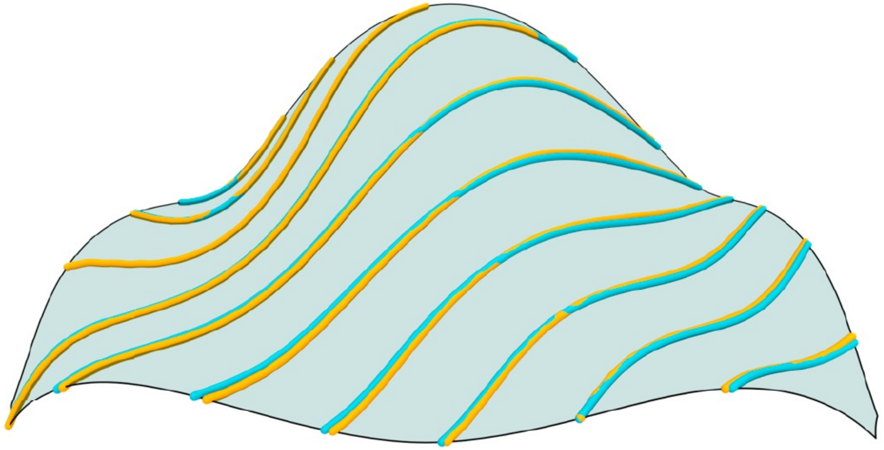
Validation: stability under remeshing



8960 vertices



657 vertices



Error is 0.76% of bbd

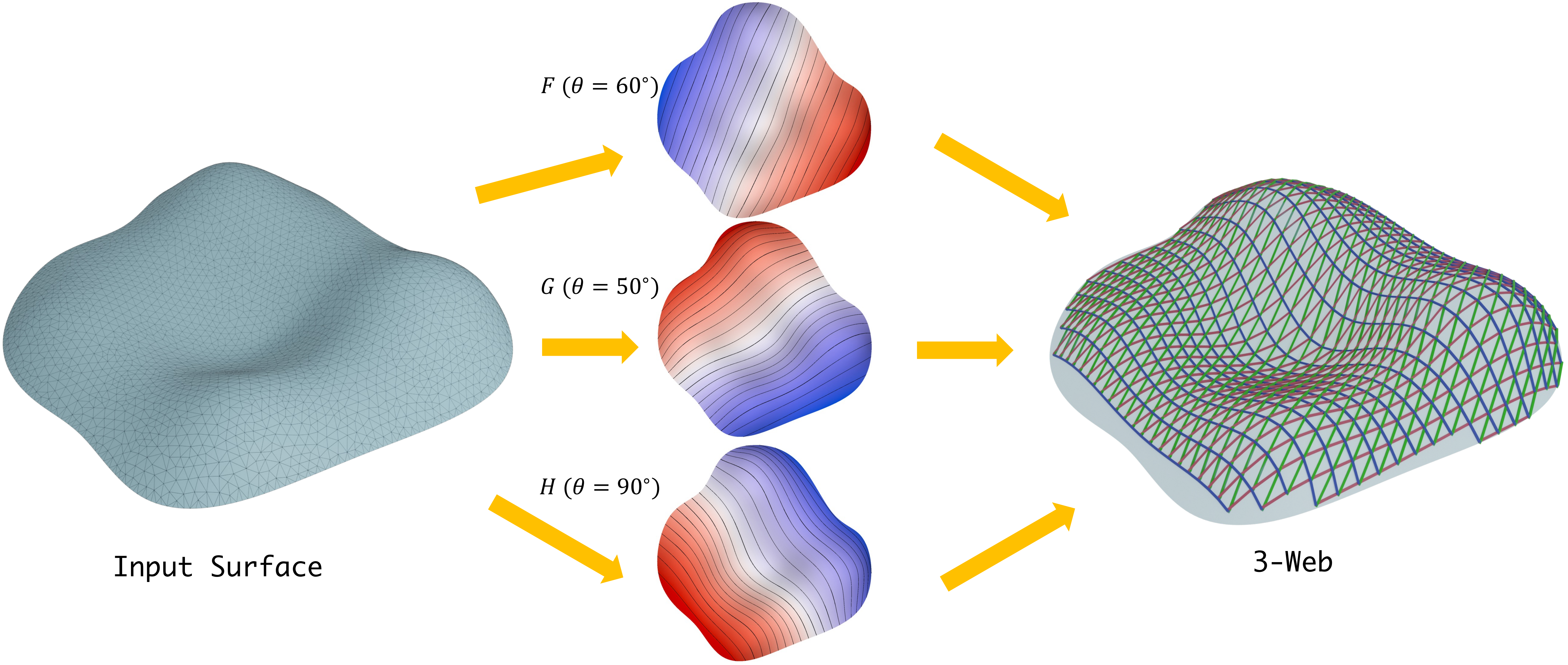
Optimizing Pseudo-Geodesics

Co-optimizing Reference Surface and Level Sets

$\theta = 60^\circ$ $\theta = 45^\circ$ $\theta = 75^\circ$ $\theta = 90^\circ$

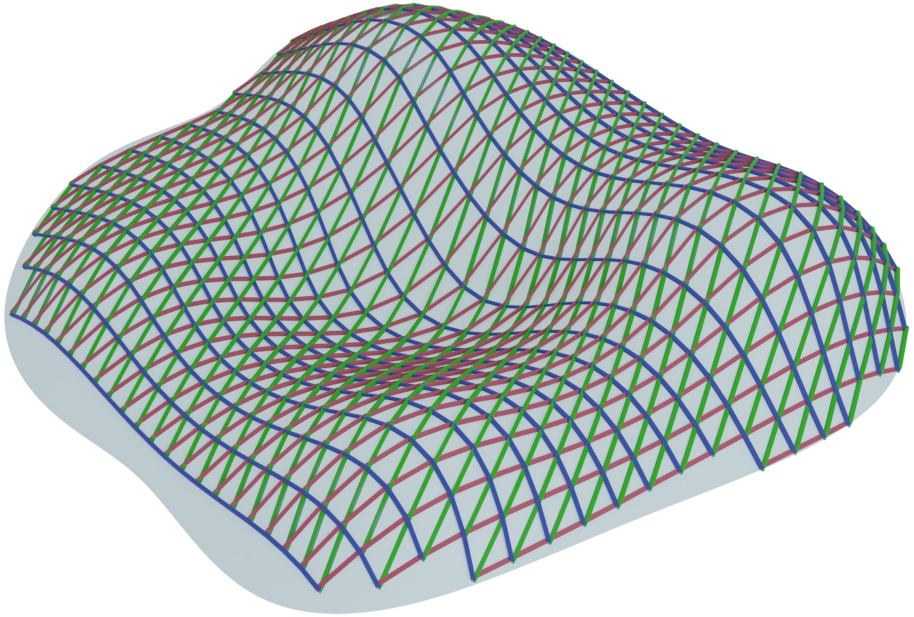
Keeping F fixed and optimize the underlying surface

Optimizing Pseudo-Geodesics

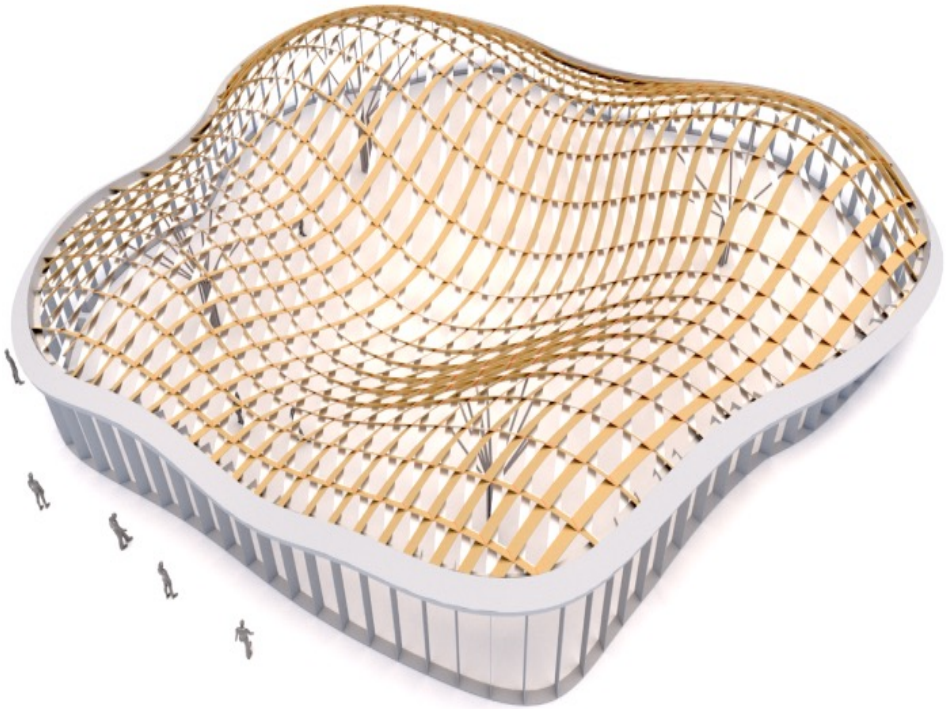


Angle Constraints + Geometry of Webs $(F + G + H = 0)$

Postprocessing



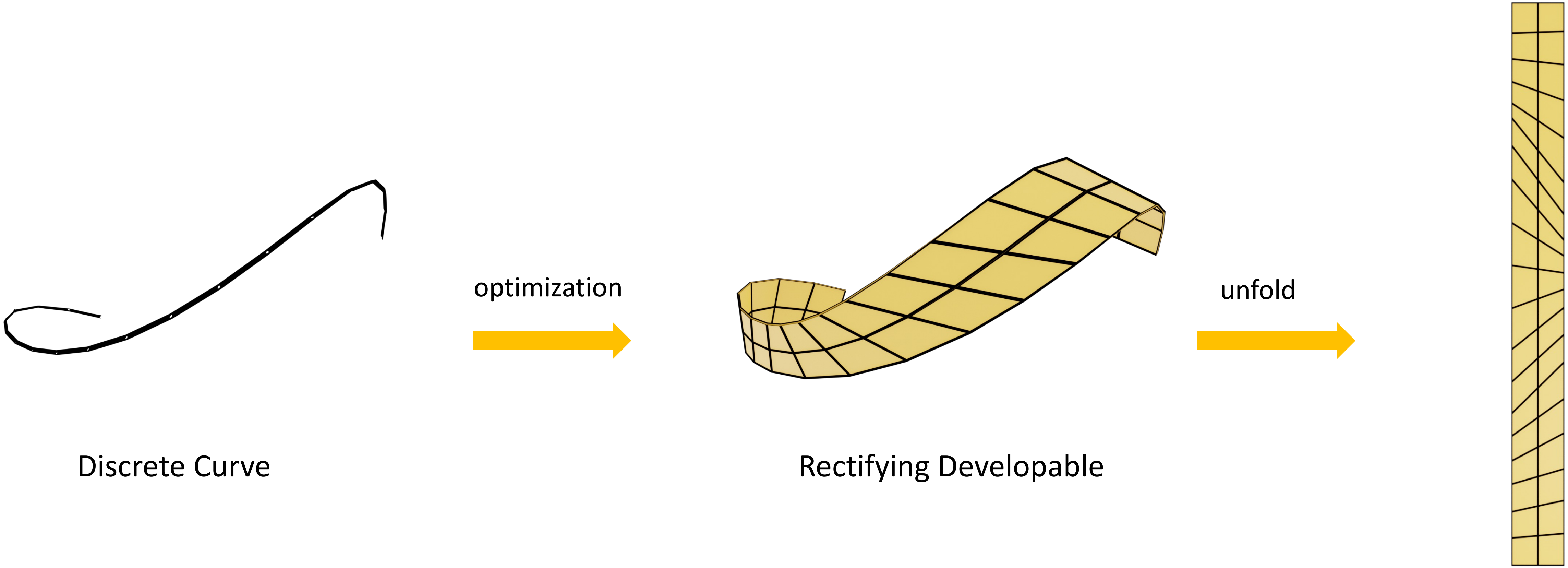
3-Web



Rectifying Strip Structure

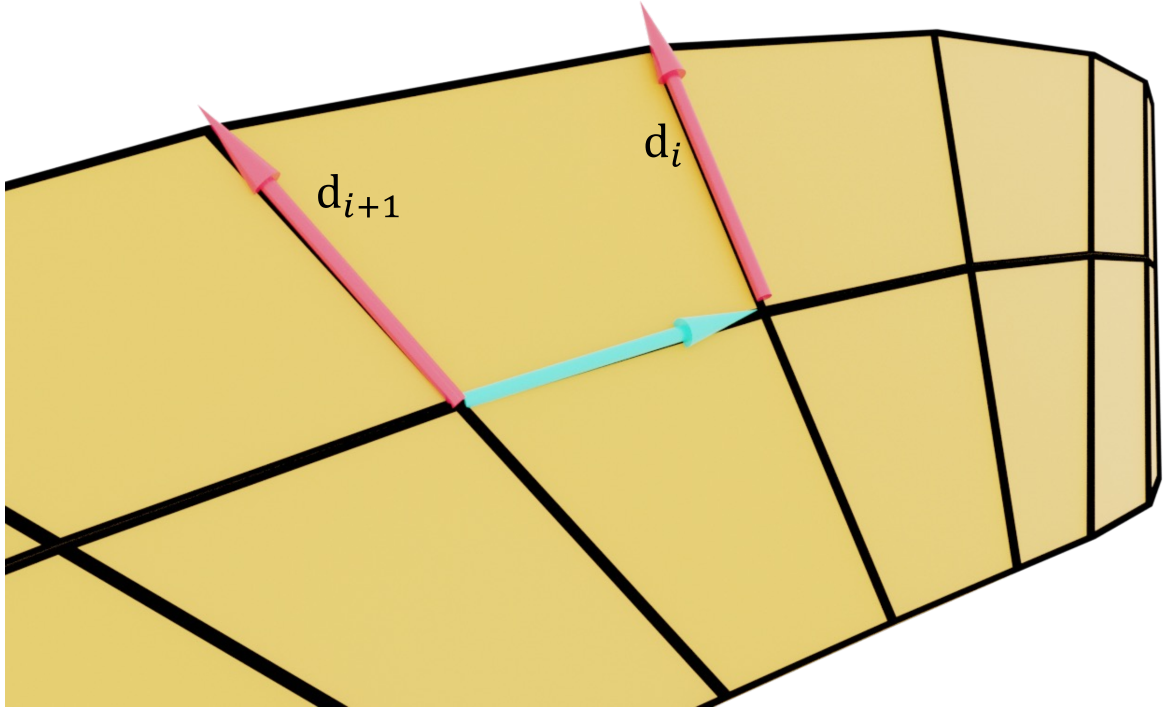
Postprocessing

Discrete Rectifying Developable Optimization



Postprocessing

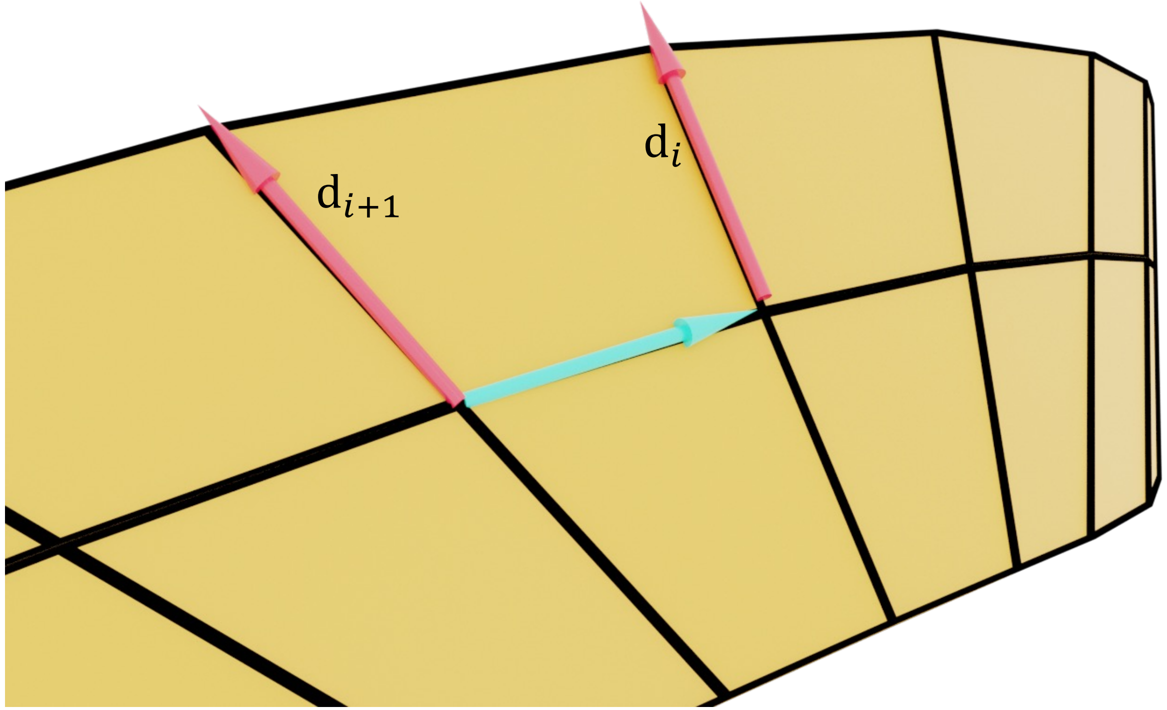
Discrete Rectifying Developable Optimization

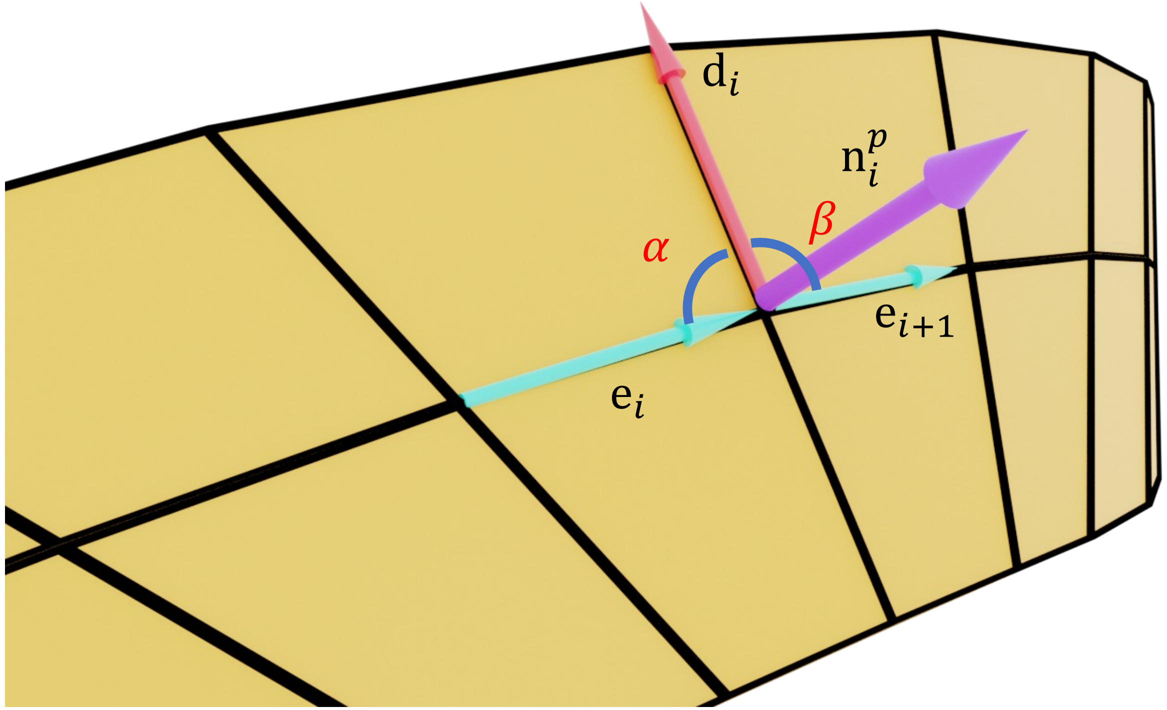


Flat: The rulings d_i, d_{i+1} are coplanar

Postprocessing

Discrete Rectifying Developable Optimization



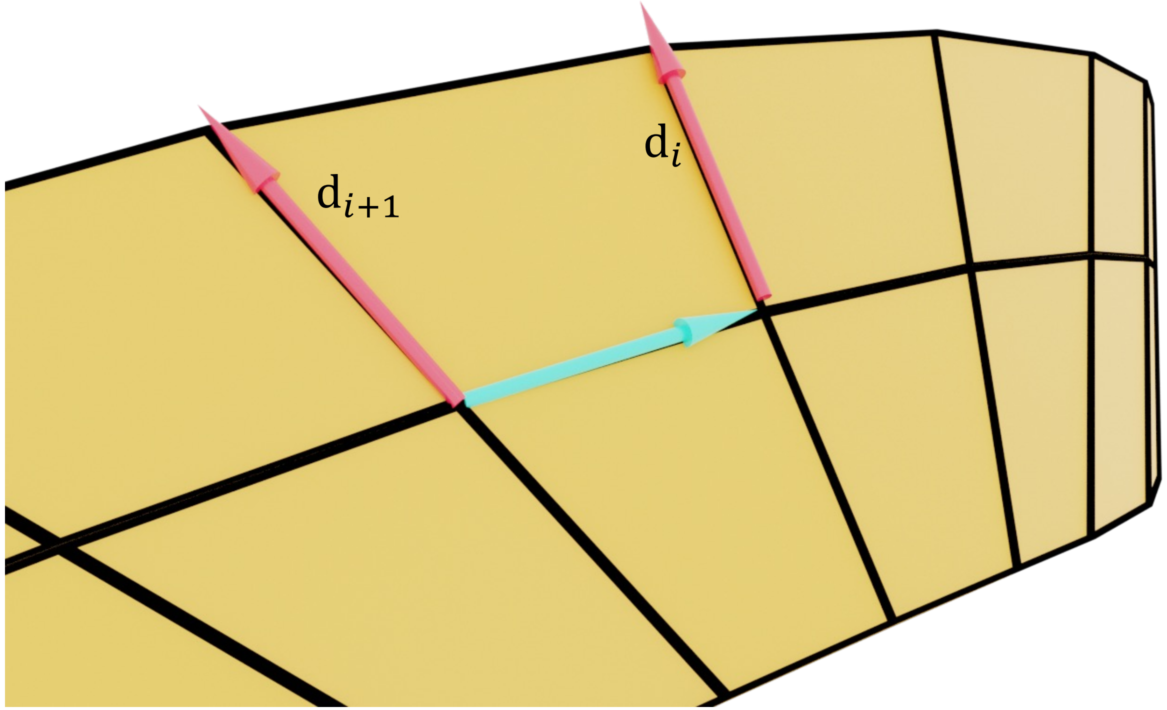


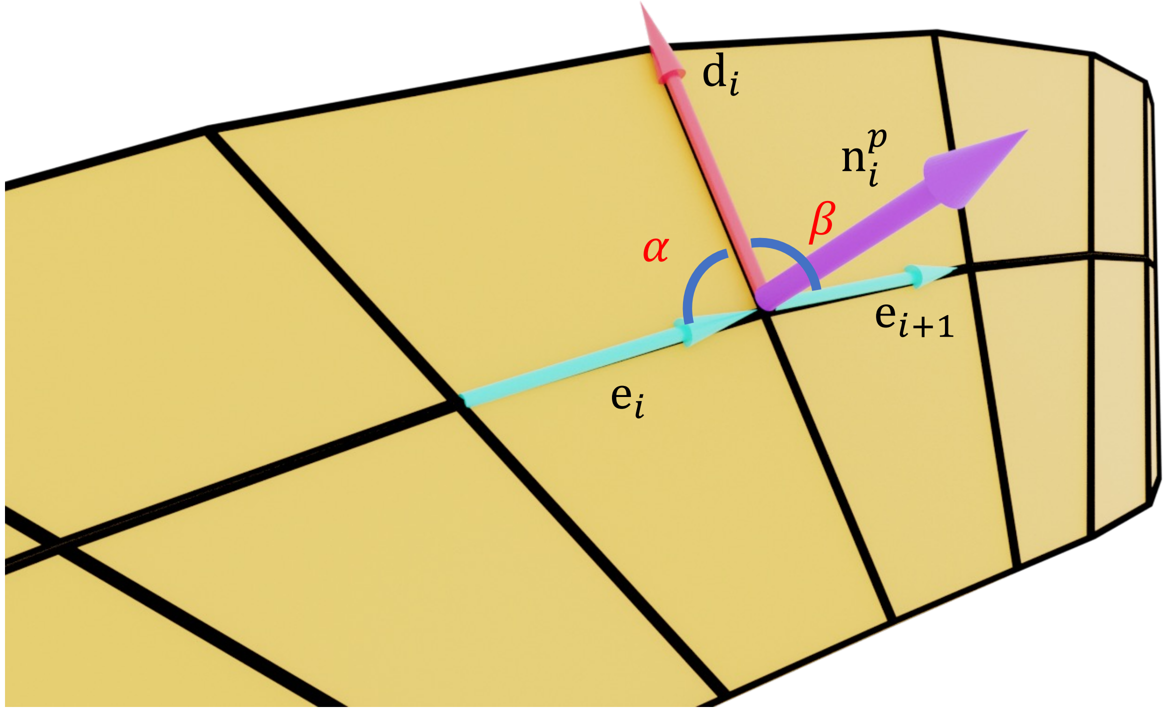
Flat: The rulings d_i, d_{i+1} are coplanar

Straight: $\alpha + \beta = \pi$

Postprocessing

Discrete Rectifying Developable Optimization



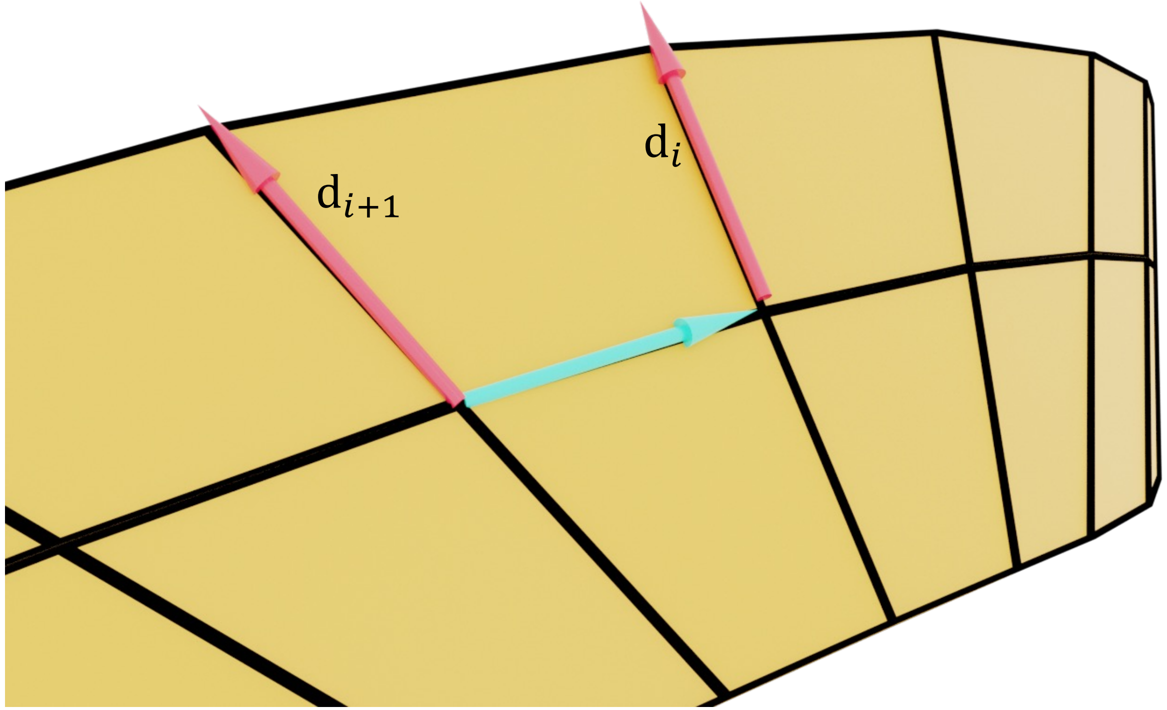


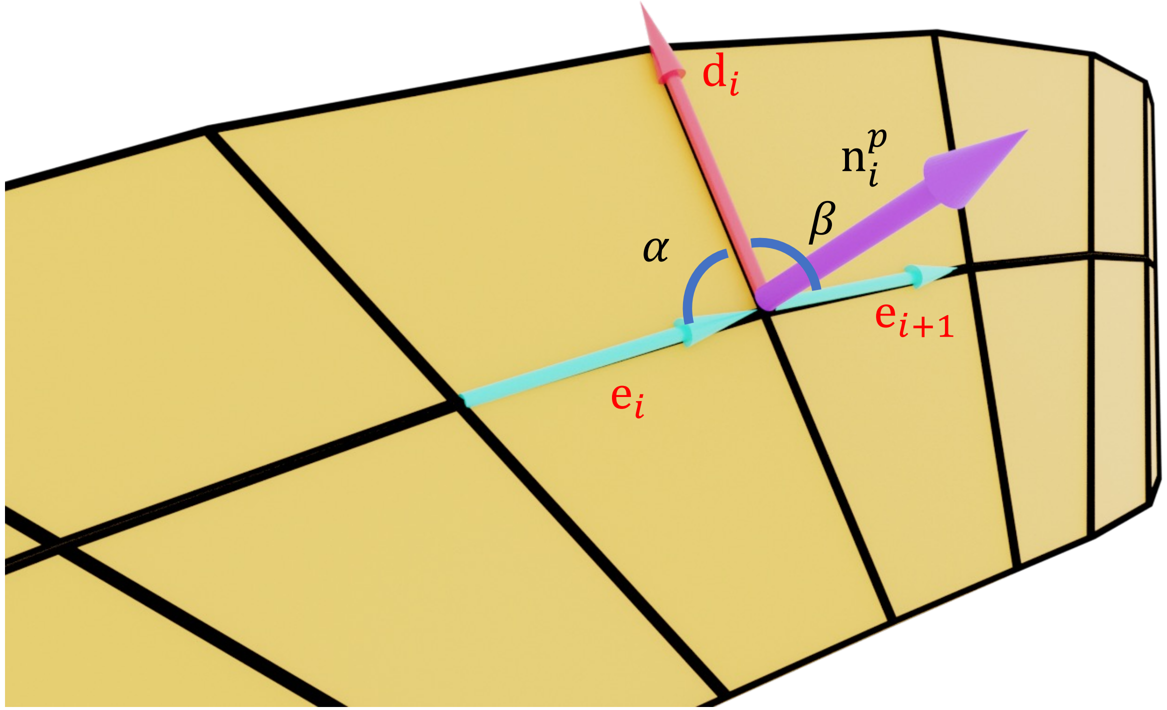
Flat: The rulings d_i, d_{i+1} are coplanar

Straight: $\cos(\alpha) + \cos(\beta) = 0$

Postprocessing

Discrete Rectifying Developable Optimization



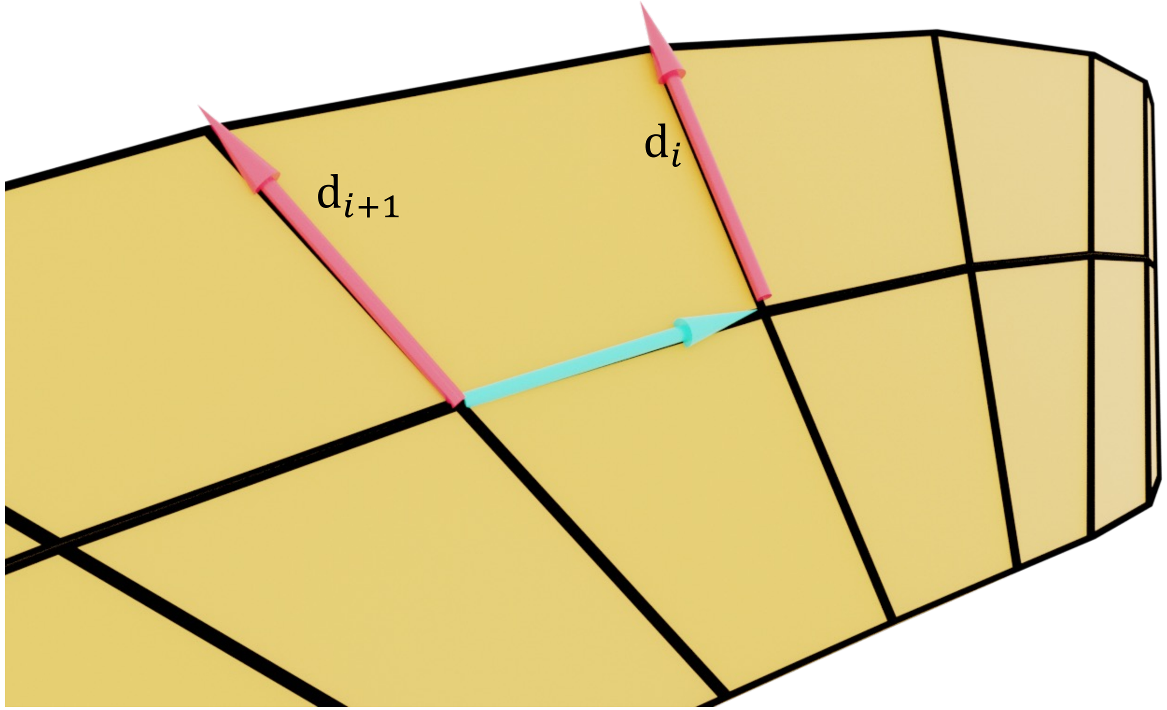


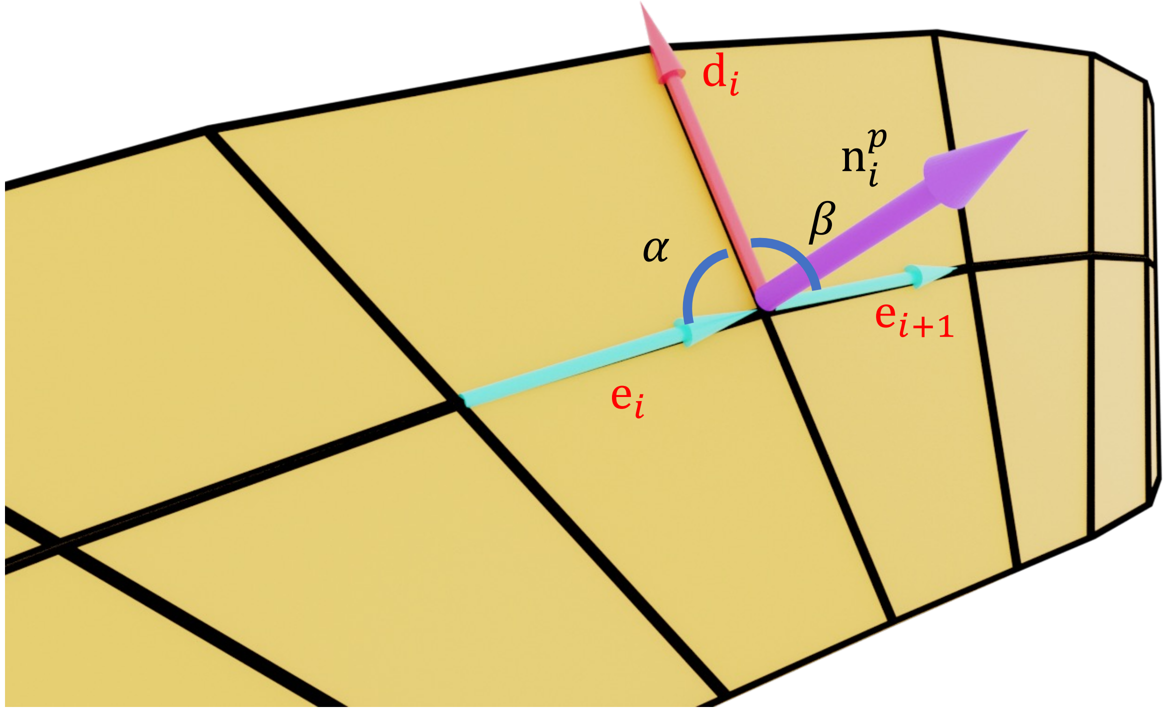
Flat: The rulings d_i, d_{i+1} are coplanar

Straight: $(e_i, d_i) - (e_{i+1}, d_i) = 0$

Postprocessing

Discrete Rectifying Developable Optimization



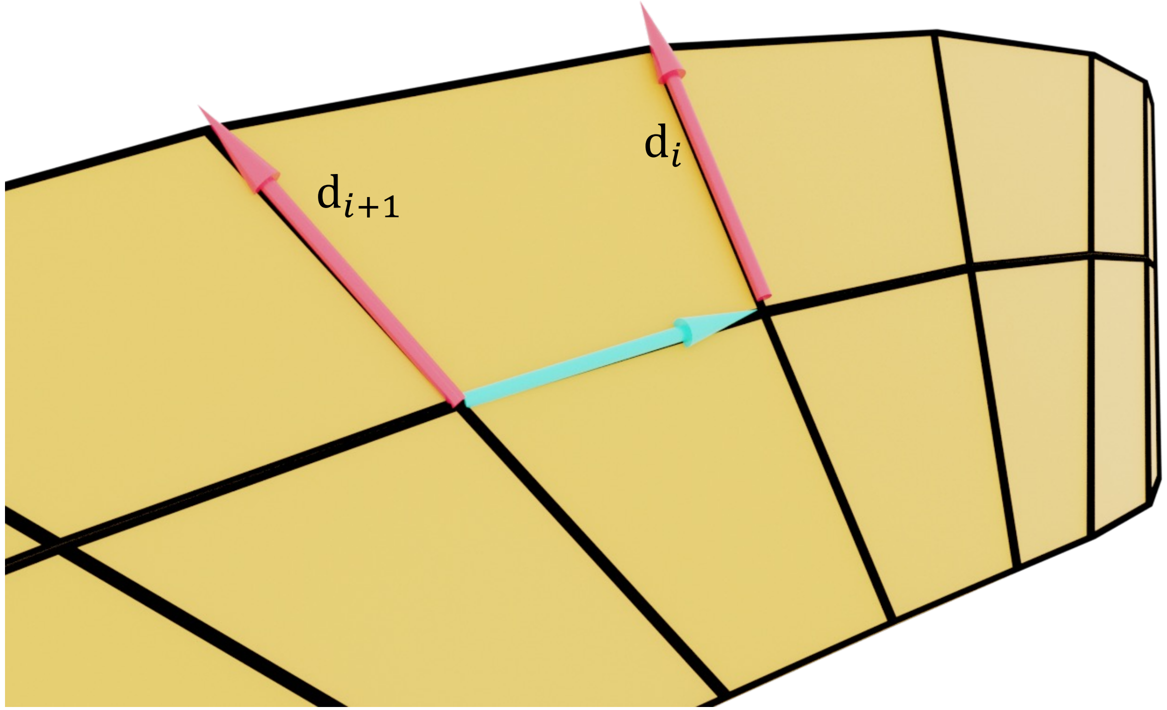


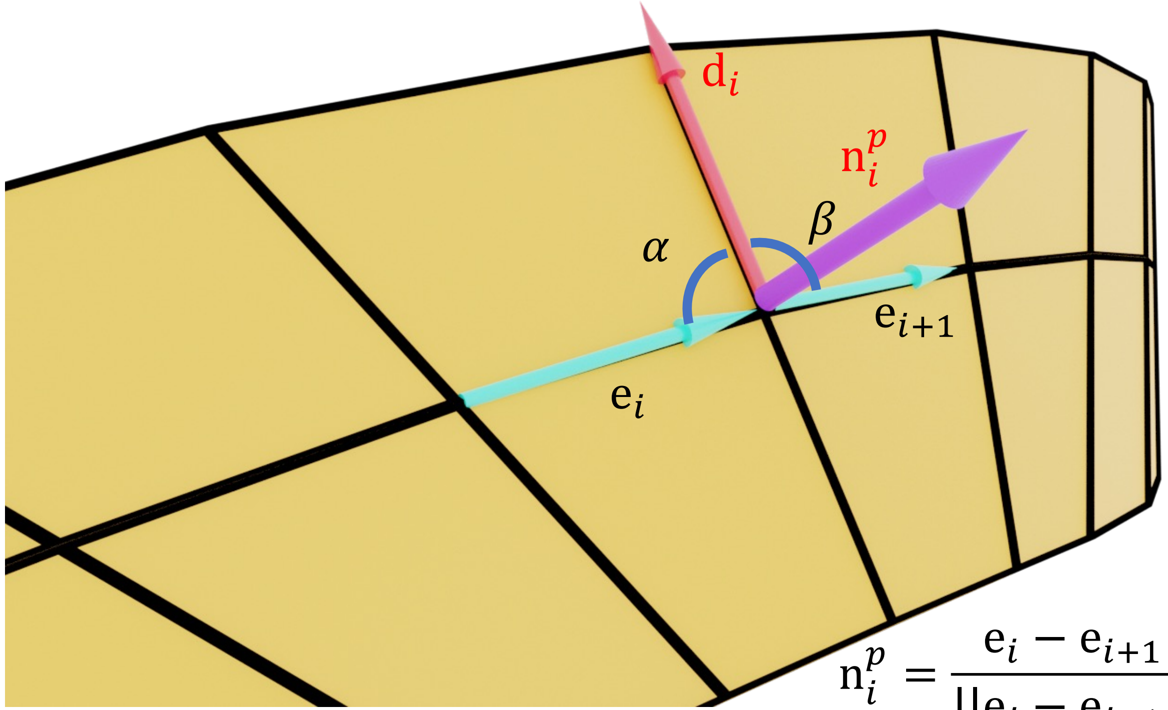
Flat: The rulings d_i, d_{i+1} are coplanar

Straight: $(e_i - e_{i+1}) \perp d_i$

Postprocessing

Discrete Rectifying Developable Optimization



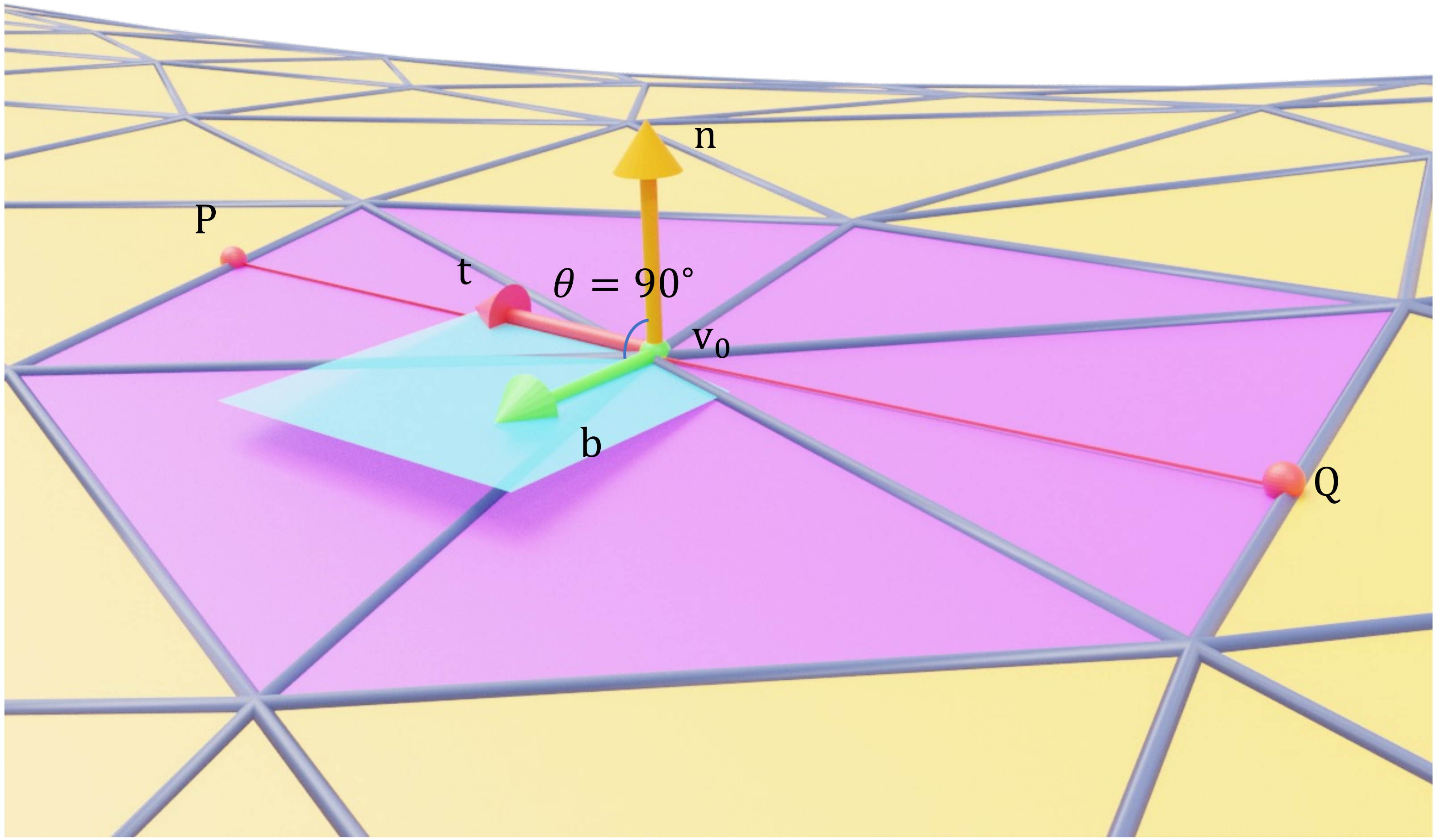


$$n_i^p = \frac{e_i - e_{i+1}}{||e_i - e_{i+1}||}$$

Flat: The rulings d_i, d_{i+1} are coplanar

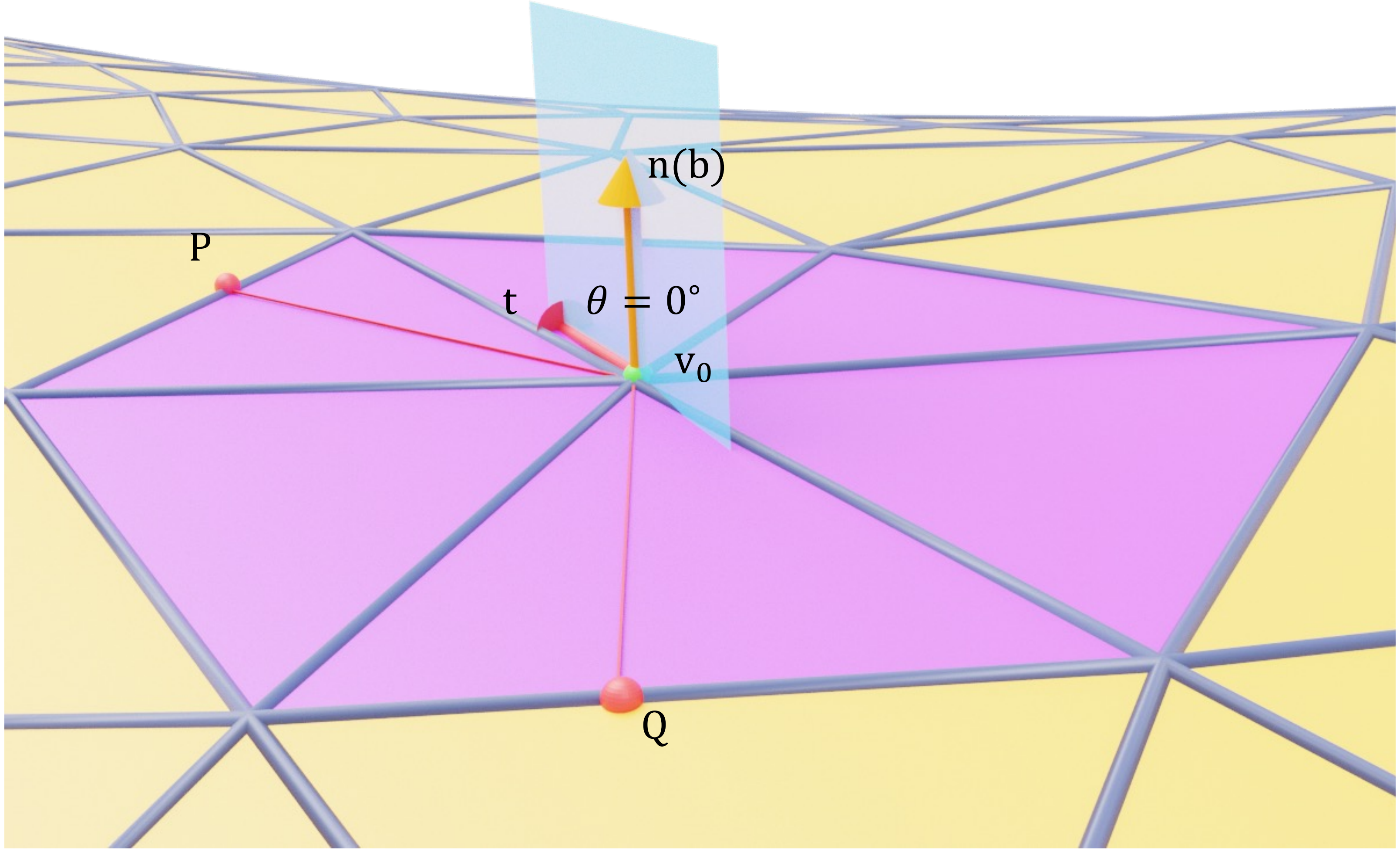
Straight: $n_i^p \perp d_i$

Applications: gridshells



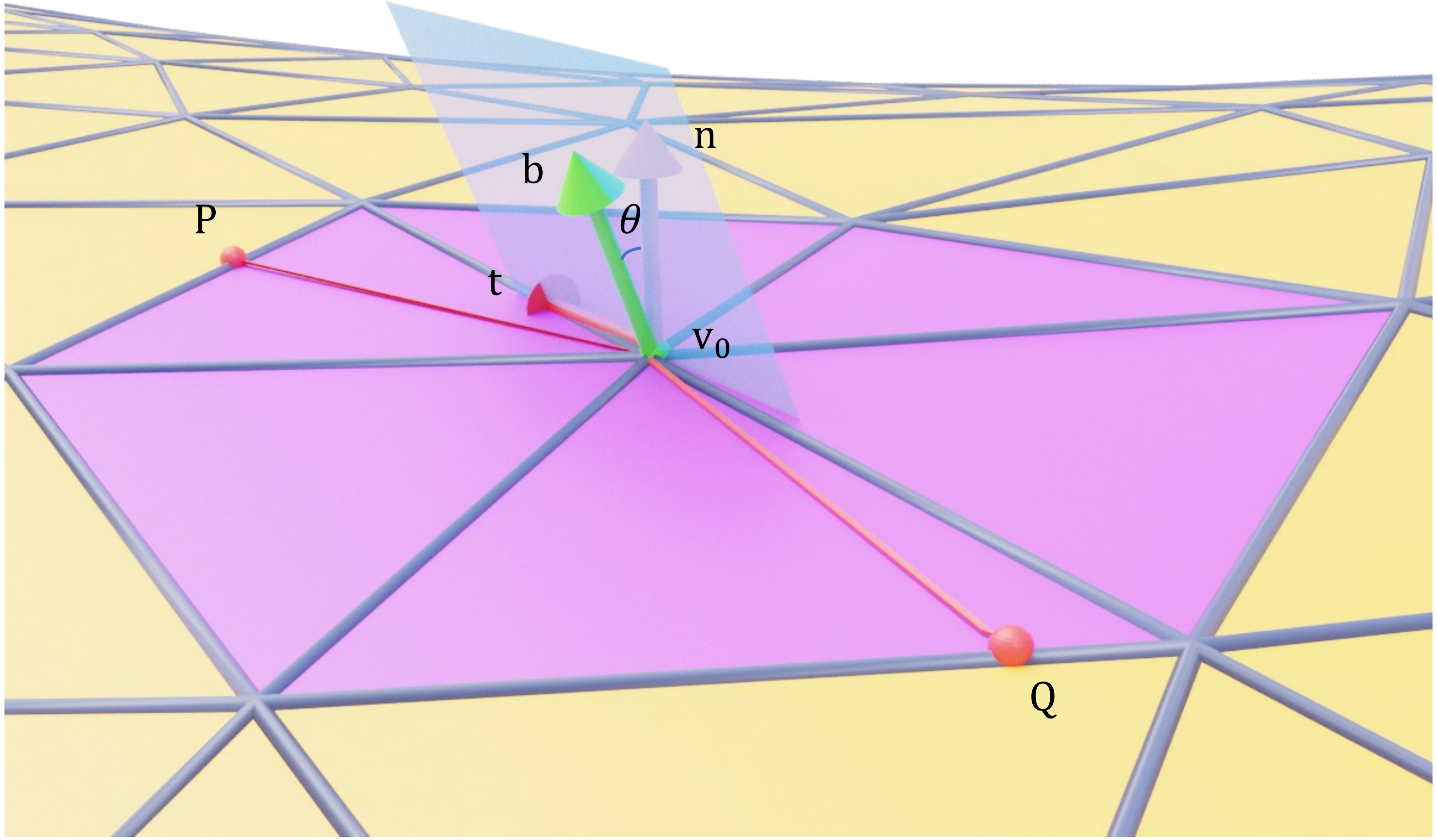
G: Geodesic Strips

Applications: gridshells



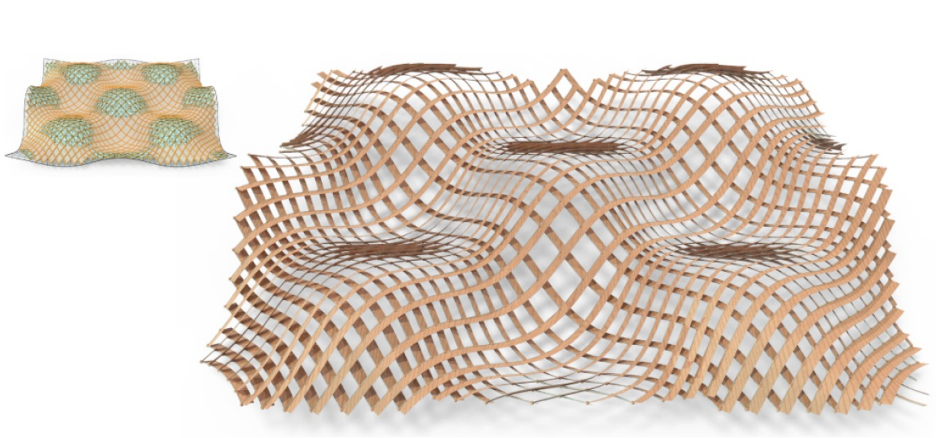
A: Asymptotic Strips

Applications: gridshells

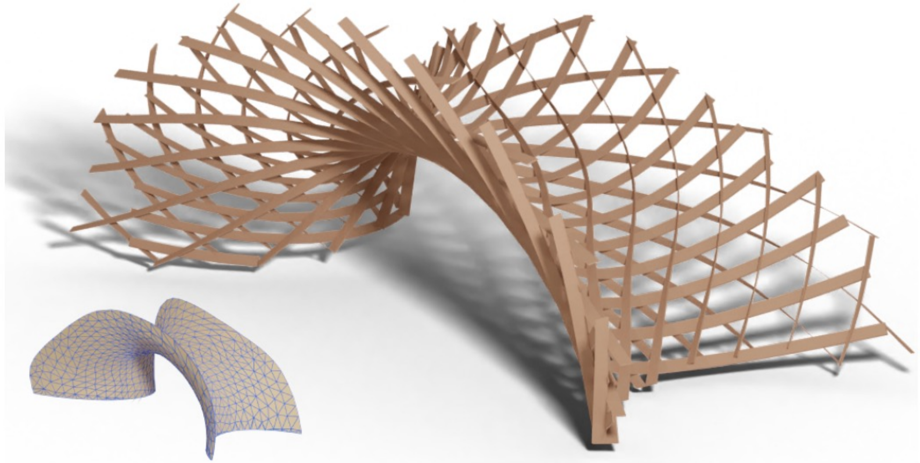


P: Pseudo-Geodesic Strips ($\theta \neq 0, \pi/2$)

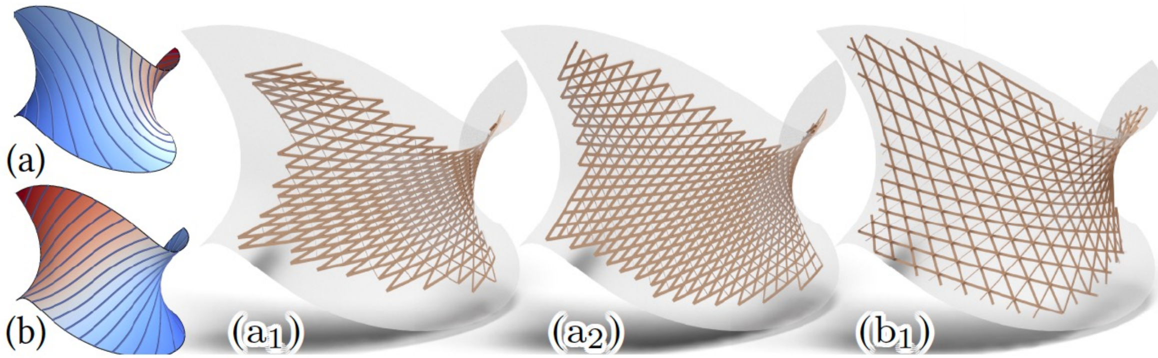
Applications: gridshells



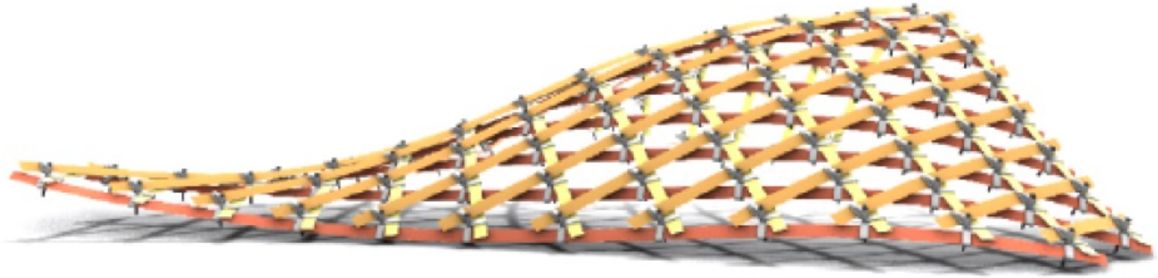
PP-Net $\theta_1 = \theta_2 = 60°$



AAG-Web



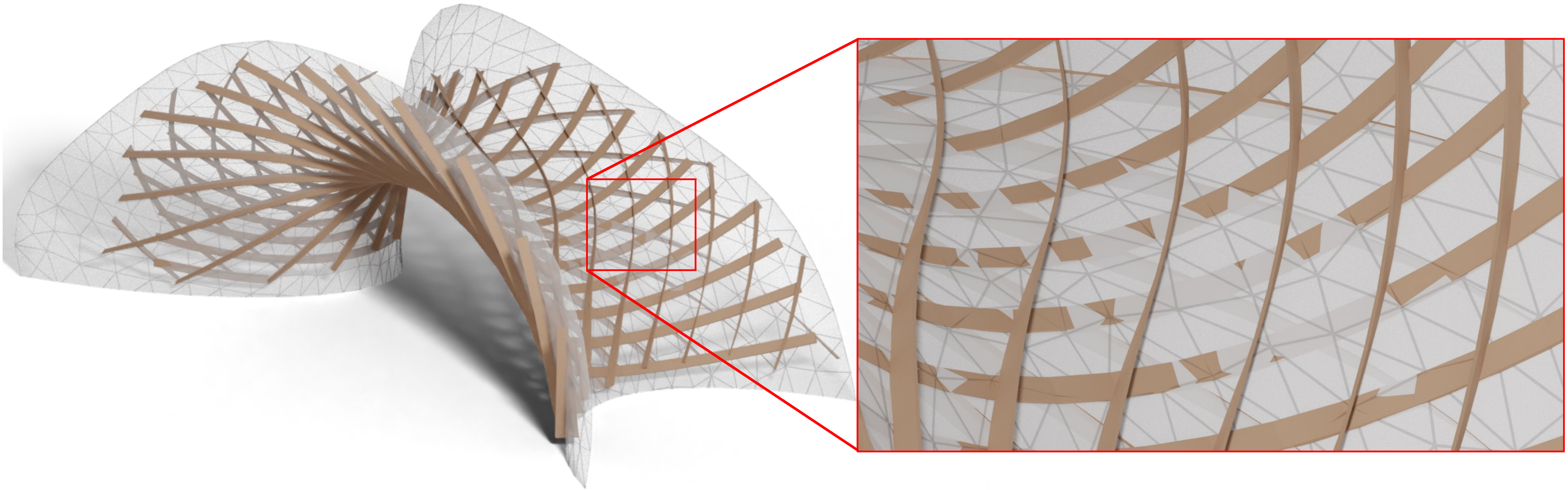
AGG-Webs



PPG-Web $\theta_1 = \theta_2 = 60°$

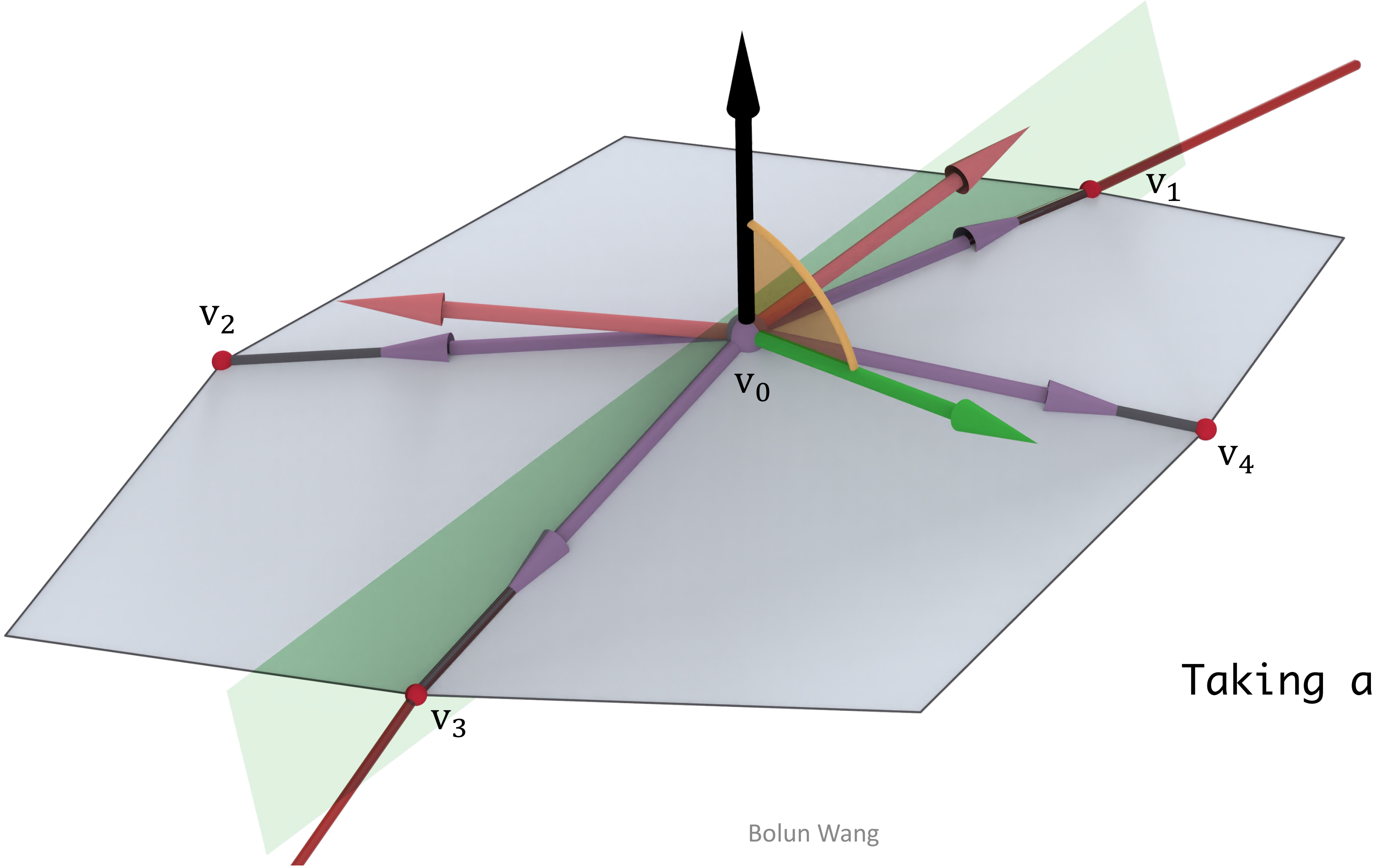
Applications: gridshells

Changing the Underlying Surface for More Accurate Results



Applications: gridshells

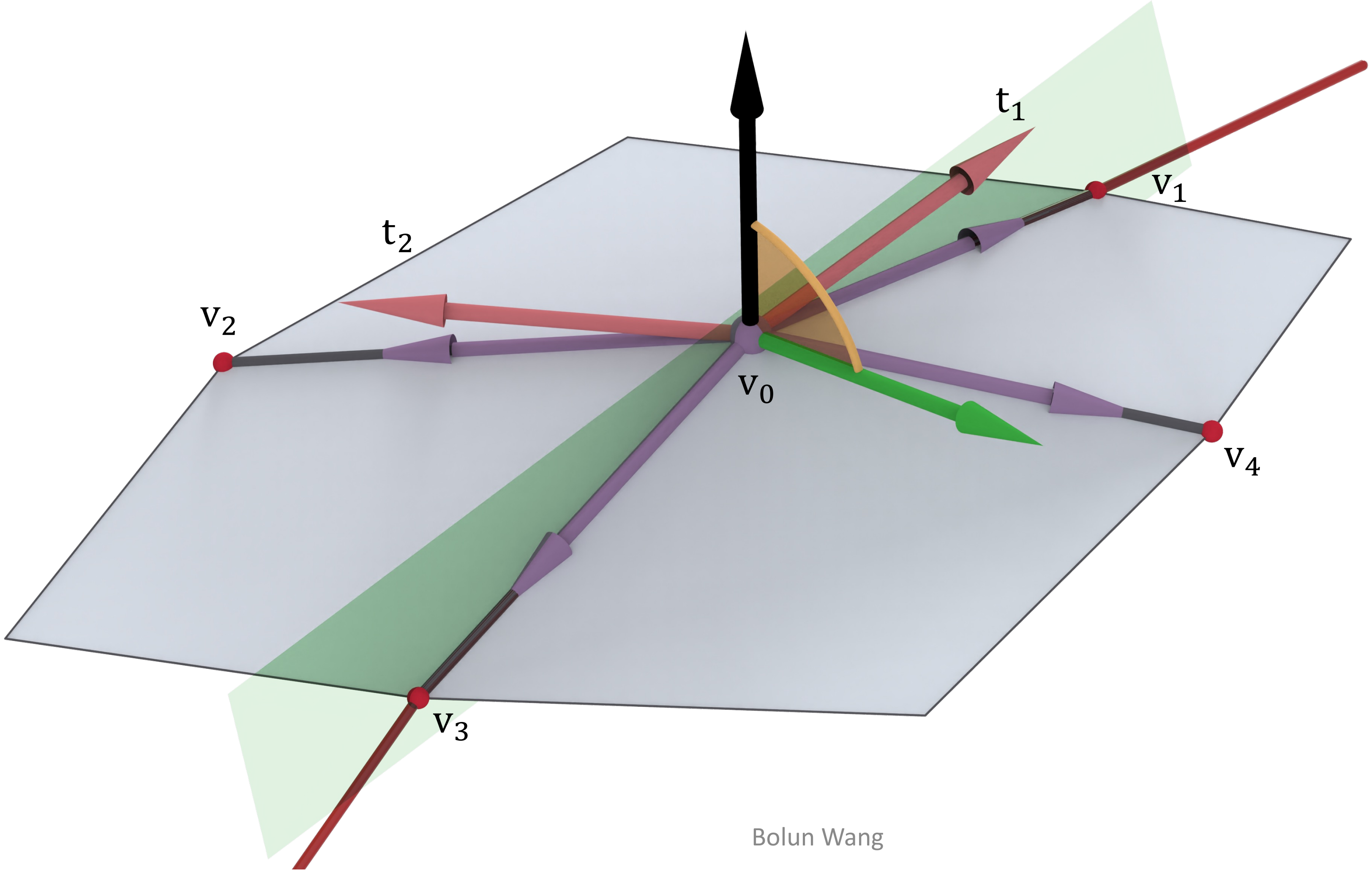
Changing the Underlying Surface for More Accurate Results



Taking a net as an example

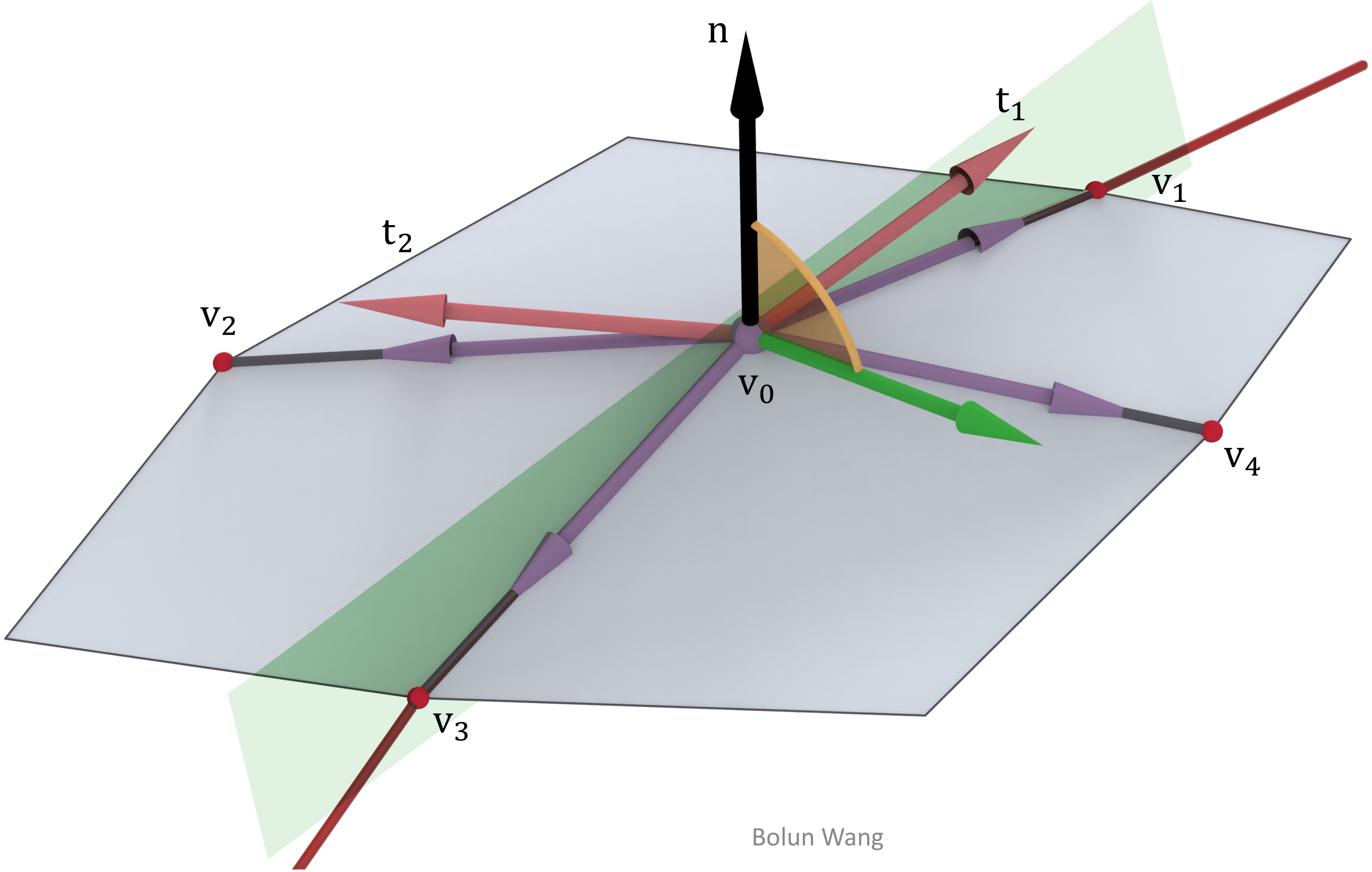
Applications: gridshells

Changing the Underlying Surface for More Accurate Results



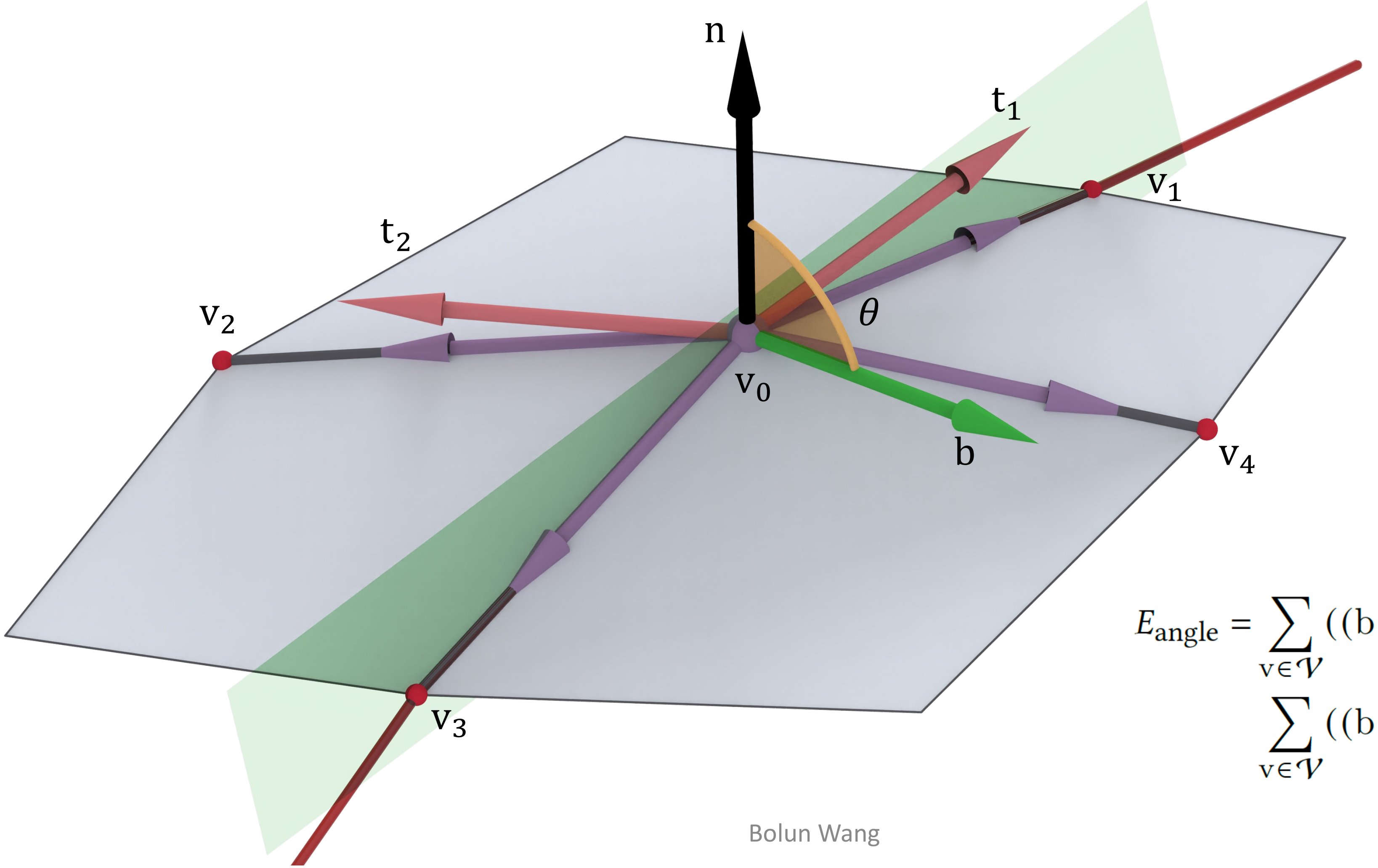
Applications: gridshells

Changing the Underlying Surface for More Accurate Results



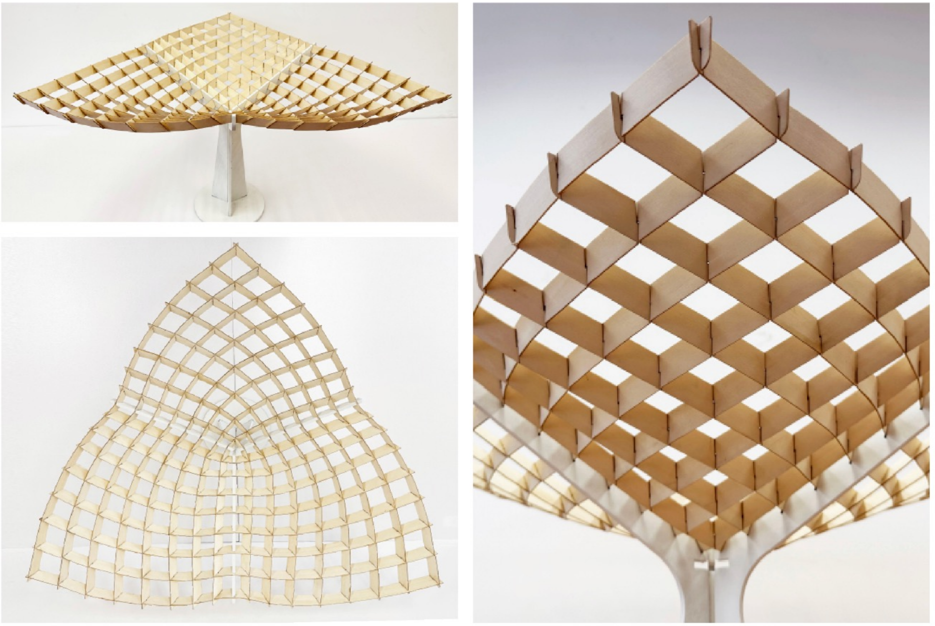
Applications: gridshells

Changing the Underlying Surface for More Accurate Results

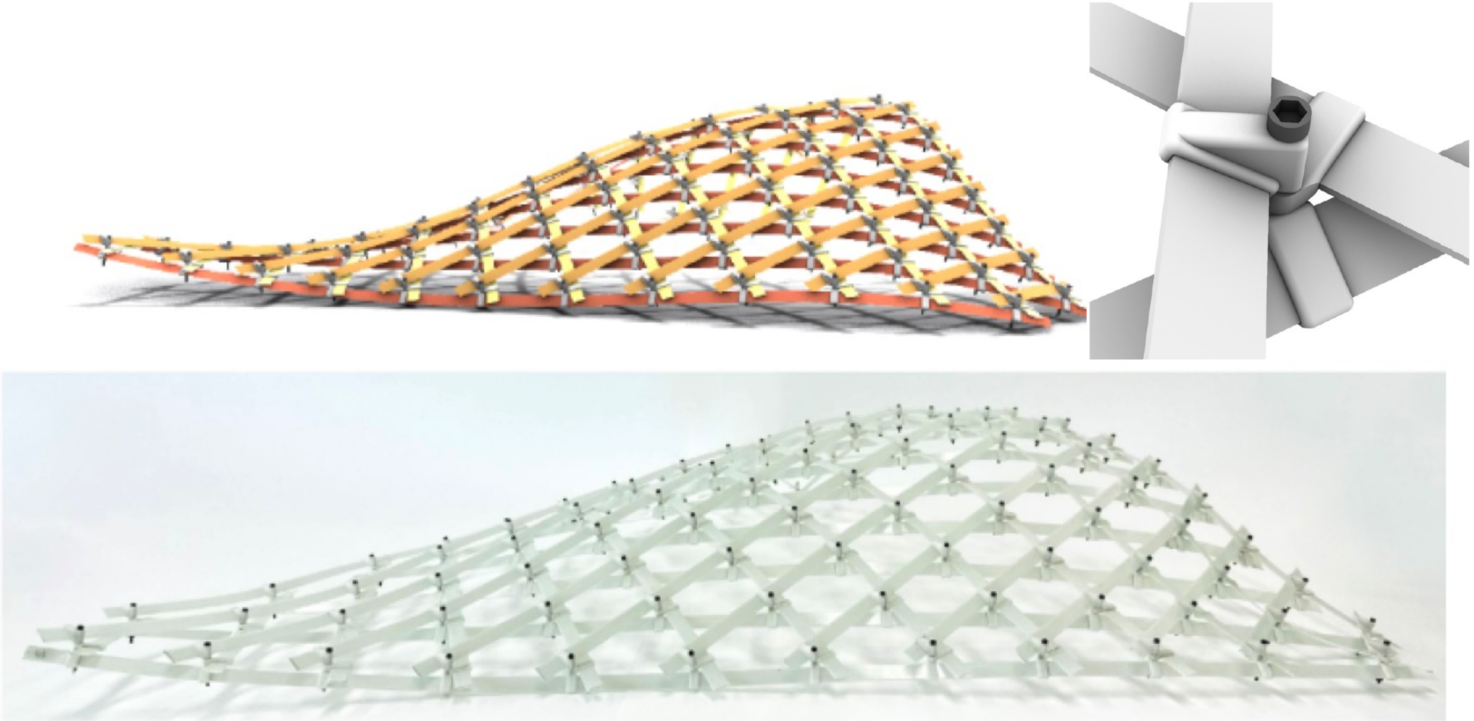


$$E_{\text{angle}} = \sum_{v \in \mathcal{V}} ((b \cdot n)^2 - \cos^2\theta)^2 \mathcal{A}(v) + \sum_{v \in \mathcal{V}} ((b \cdot n)(b \cdot u) - \sin\theta\cos\theta)^2 \mathcal{A}(v),$$

Applications: gridshells



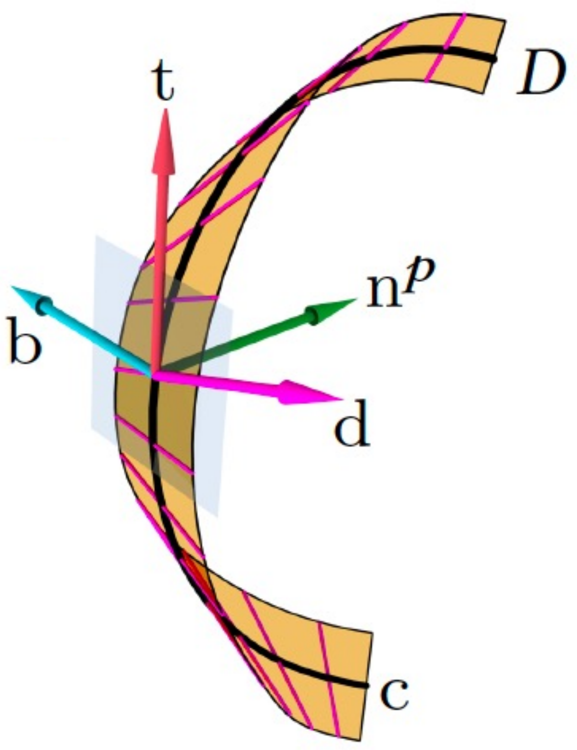
Physical Model: PP-Net, $\theta_1 = \theta_2 = 50^\circ$



Physical Model: PPG-Web, $\theta_1 = \theta_2 = 60^\circ$

Applications: gridshells

Torsion-free rectifying strip structures



The Darboux vector $d = \tau t + \kappa b$ is the ruling of the rectifying developable

$$\frac{d\mathbf{T}}{ds} = \kappa \mathbf{N}$$

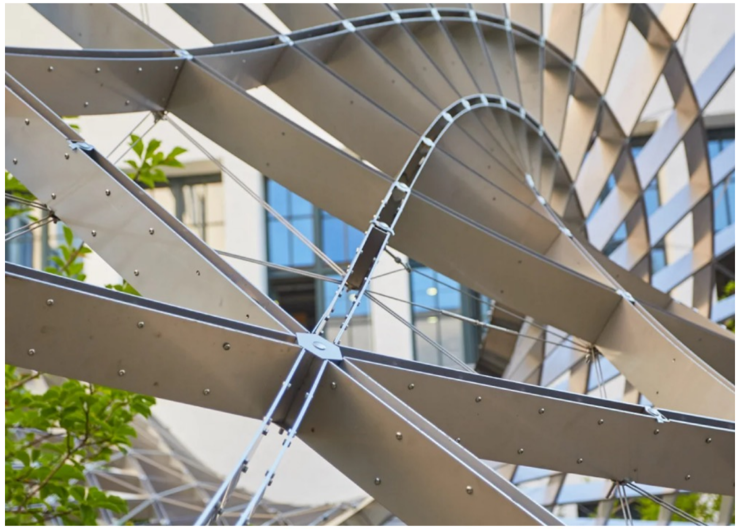
$$\frac{d\mathbf{N}}{ds} = -\kappa \mathbf{T} + \tau \mathbf{B}$$

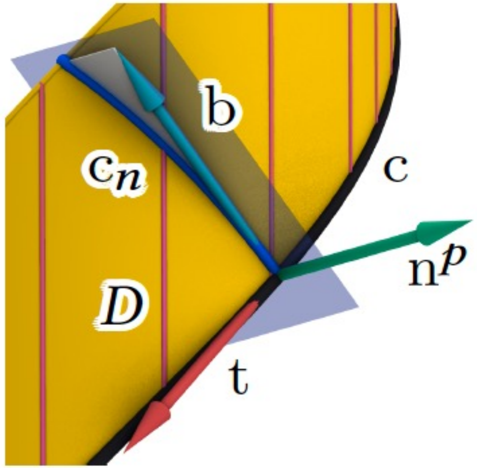
$$\frac{d\mathbf{B}}{ds} = -\tau \mathbf{N}$$

The ruled surface $S(u, v) = c(u) + v\mathrm{d}$,
$S_u = t + v(\tau\kappa n^p - \kappa\tau n^p) = t$,
$S_v = d = \tau t + \kappa b$,
$\mathrm{span}(S_u, S_v) = \mathrm{span}(t, b)$

Applications: gridshells

Torsion-free rectifying strip structures





- Using binormal vectors as node axes is not accurate!

- The curvature along the direction of b is NOT 0 if the torsion $\tau \neq 0$!

- Proposition 1. The ruled surface $\mathrm{B}(u,v) = \mathrm{c}(u) + v \cdot \mathrm{b}(u)$ has Gaussian curvature

$$K(u,v) = -\left(\frac{\tau}{1+\tau^2 v^2}\right)^2.$$

- Proposition 2. The first normal curvature $\kappa_1 = 0$ along the Darboux vector d. The second normal curvature κ_2 and normal curvature $\kappa_n(\mathrm{b})$ in direction of b are

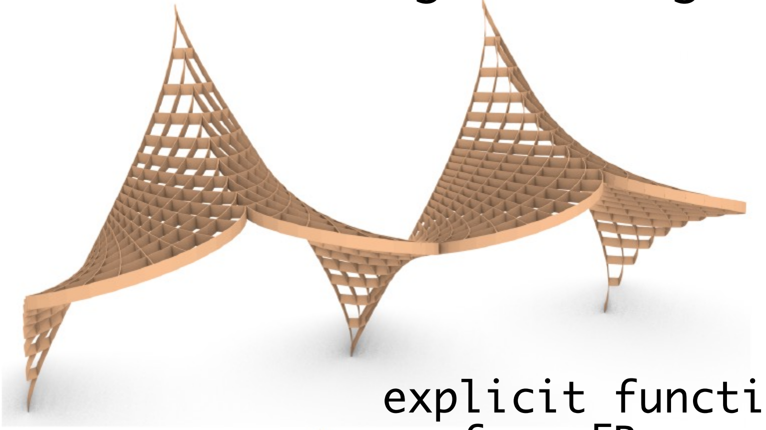
$$\kappa_2 = \kappa(1+k^2), \kappa_n(\mathrm{b}) = \kappa k^2,$$

where $k := \tau/k$.

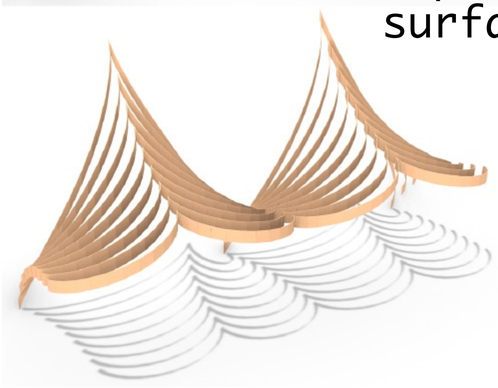
Applications: gridshells

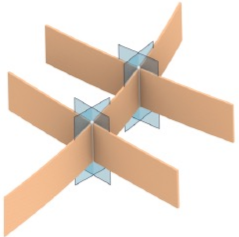
Torsion-free rectifying strip structures

Torsion-free node: a node where two developable strips intersect along a straight line segment

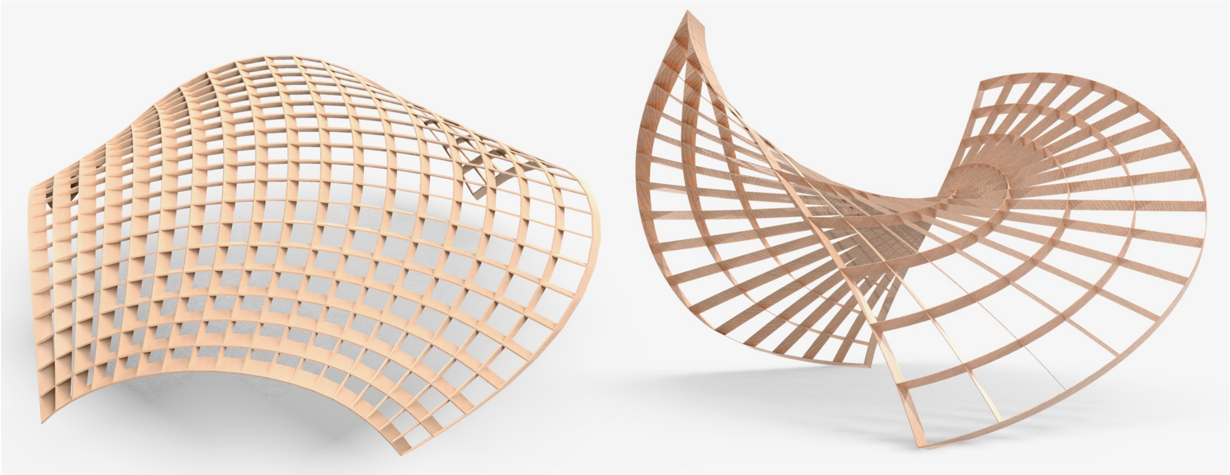


explicit functional surface [Brauner 1968]





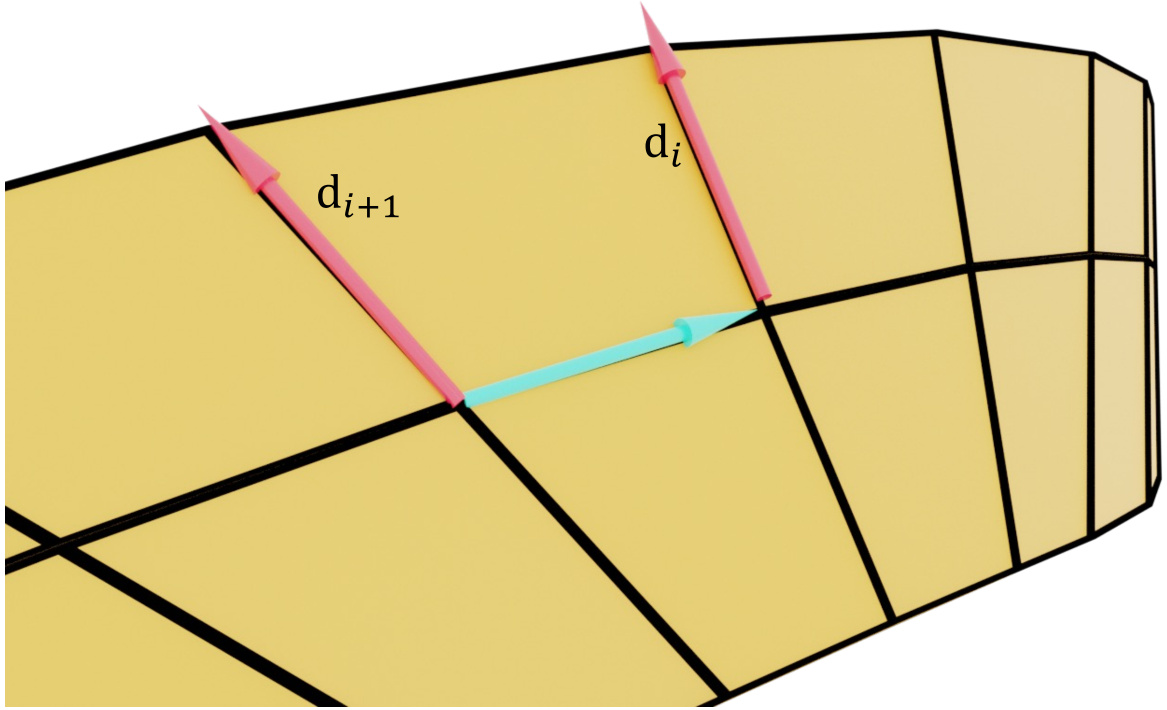
Vertical node axes

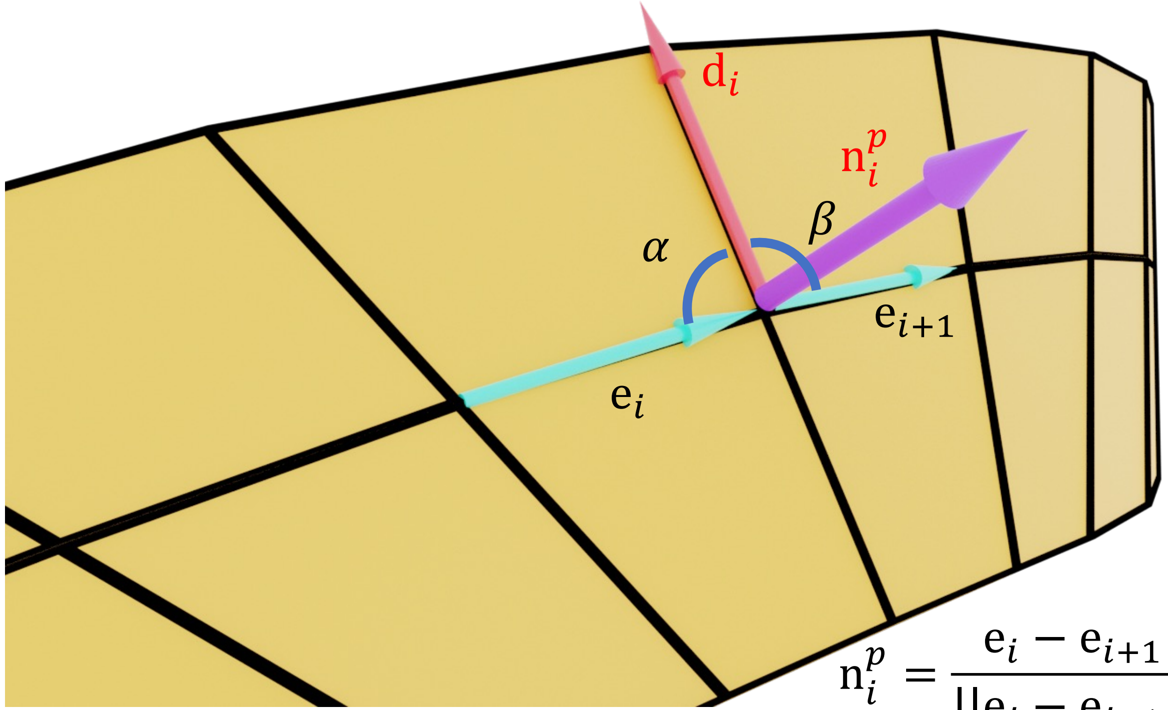


mesh optimization

Postprocessing

Discrete Rectifying Developable Optimization





$$n_i^p = \frac{e_i - e_{i+1}}{||e_i - e_{i+1}||}$$

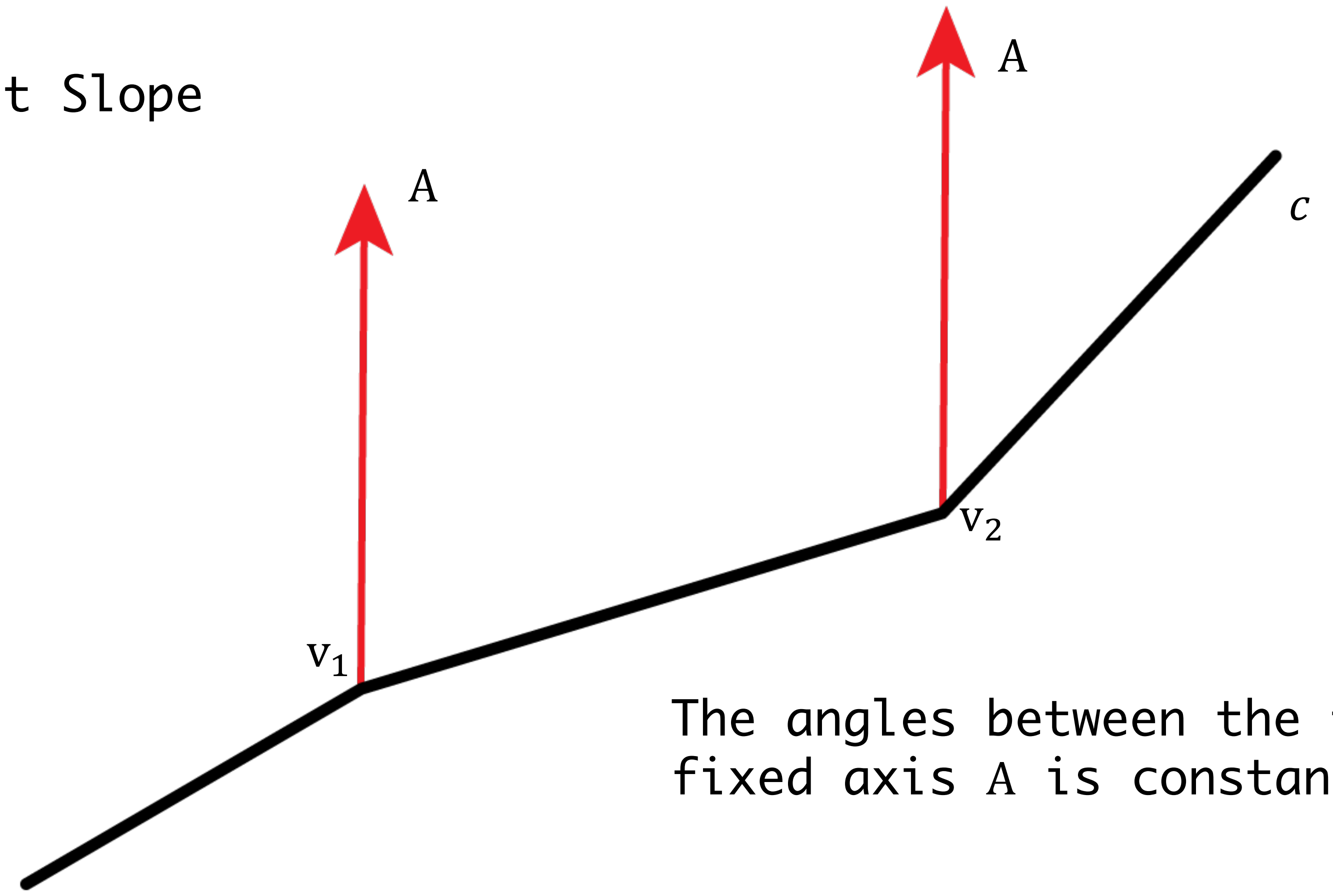
Flat: The rulings d_i, d_{i+1} are coplanar

Straight: $n_i^p \perp d_i$

Applications: gridshells

Torsion-free rectifying strip structures [Pottmann and Wallner 2001].

Curves of Constant Slope

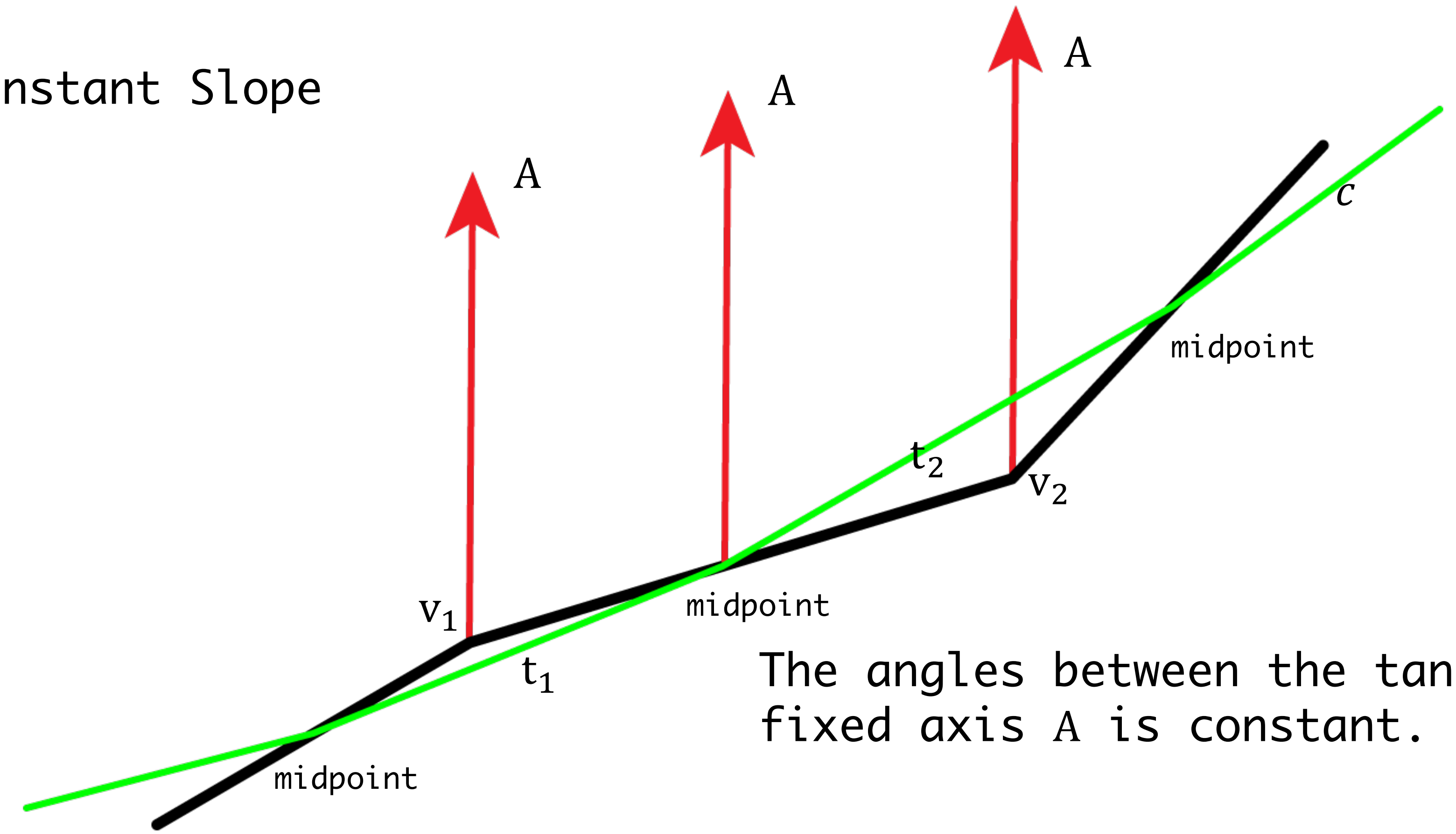


The angles between the tangent vectors t and a fixed axis A is constant.

Applications: gridshells

Torsion-free rectifying strip structures

Curves of Constant Slope

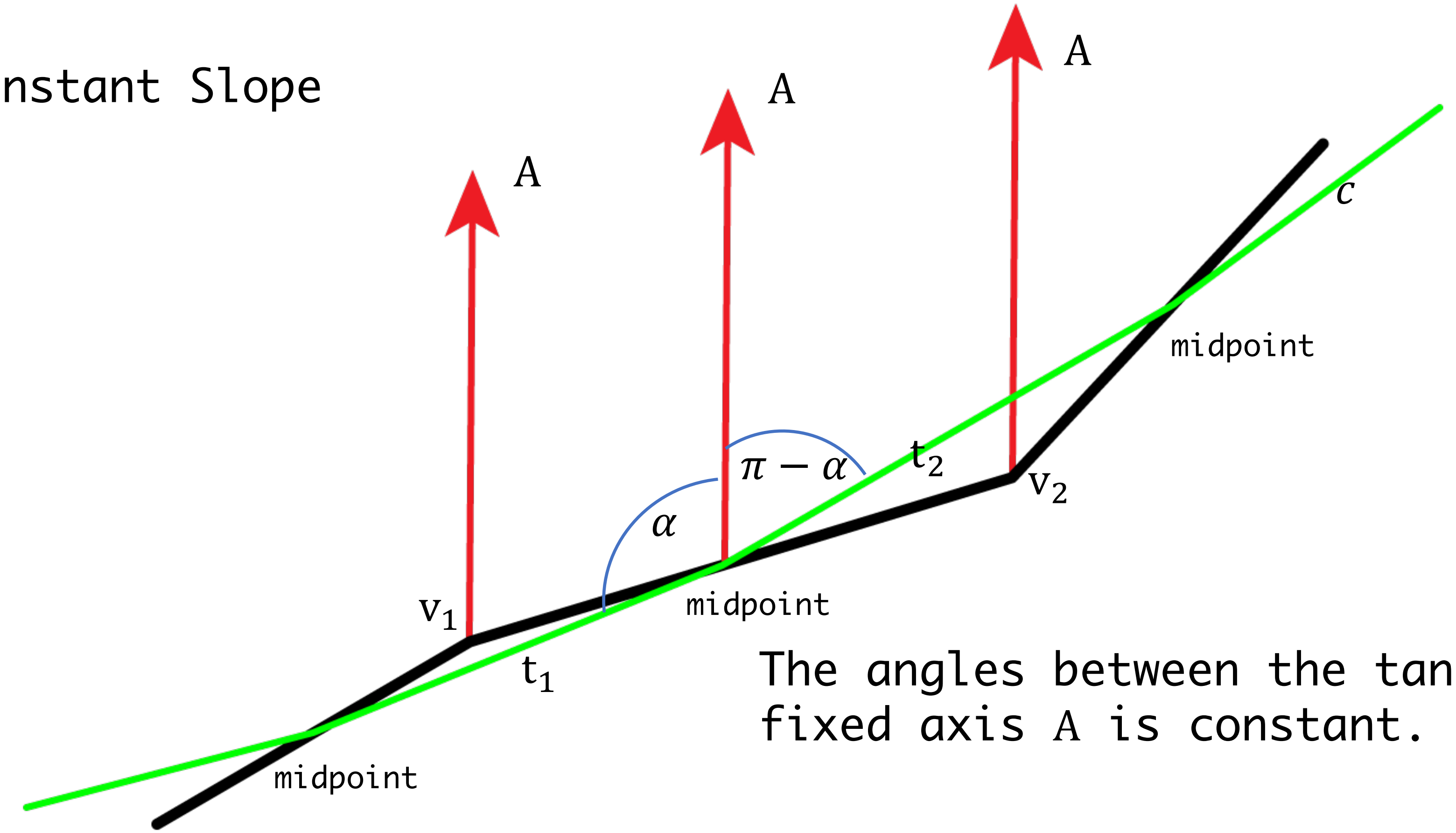


The angles between the tangent vectors t and a fixed axis A is constant.

Applications: gridshells

Torsion-free rectifying strip structures

Curves of Constant Slope

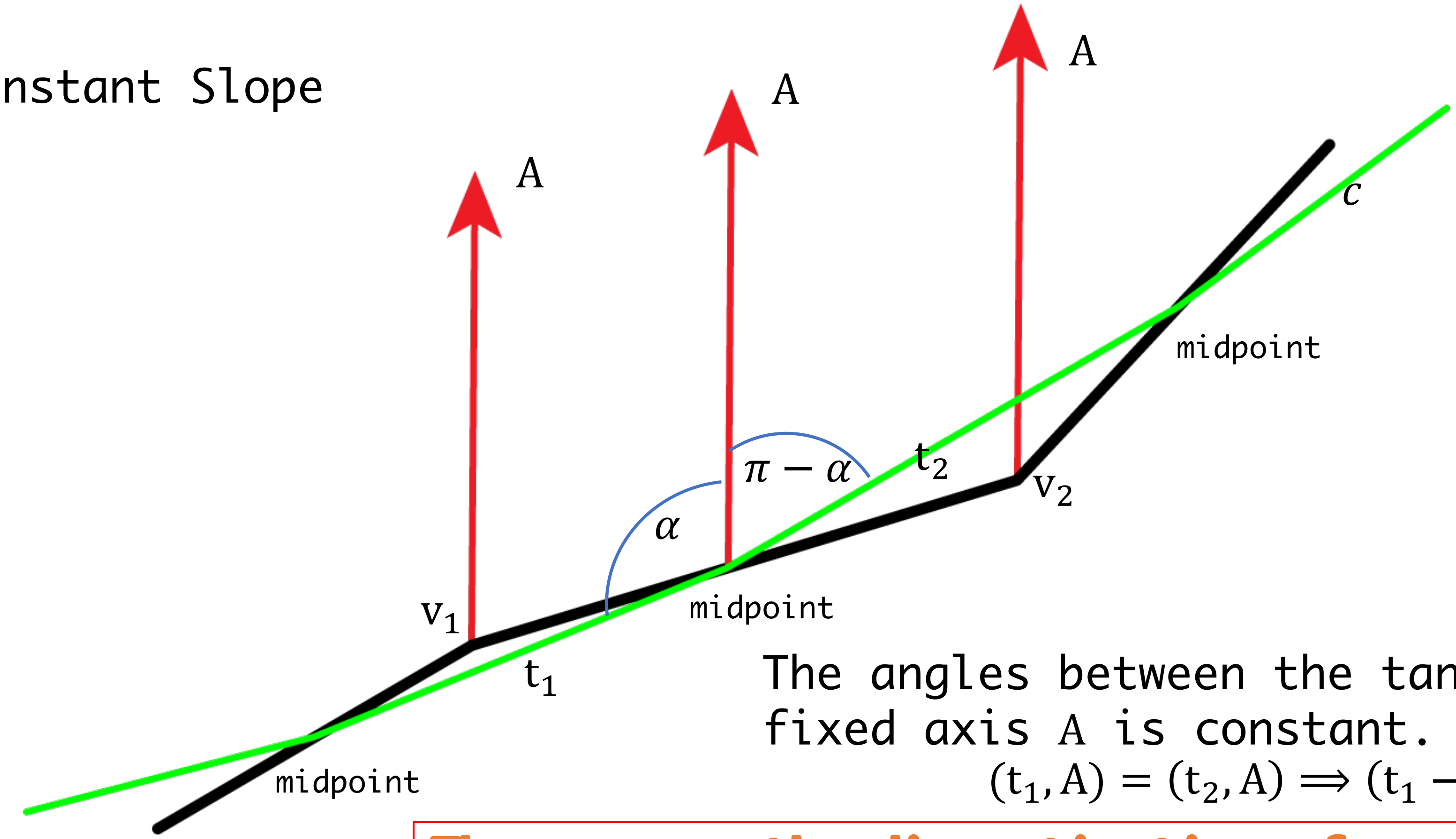


The angles between the tangent vectors t and a fixed axis A is constant.

Applications: gridshells

Torsion-free rectifying strip structures

Curves of Constant Slope



The angles between the tangent vectors t and a fixed axis A is constant.

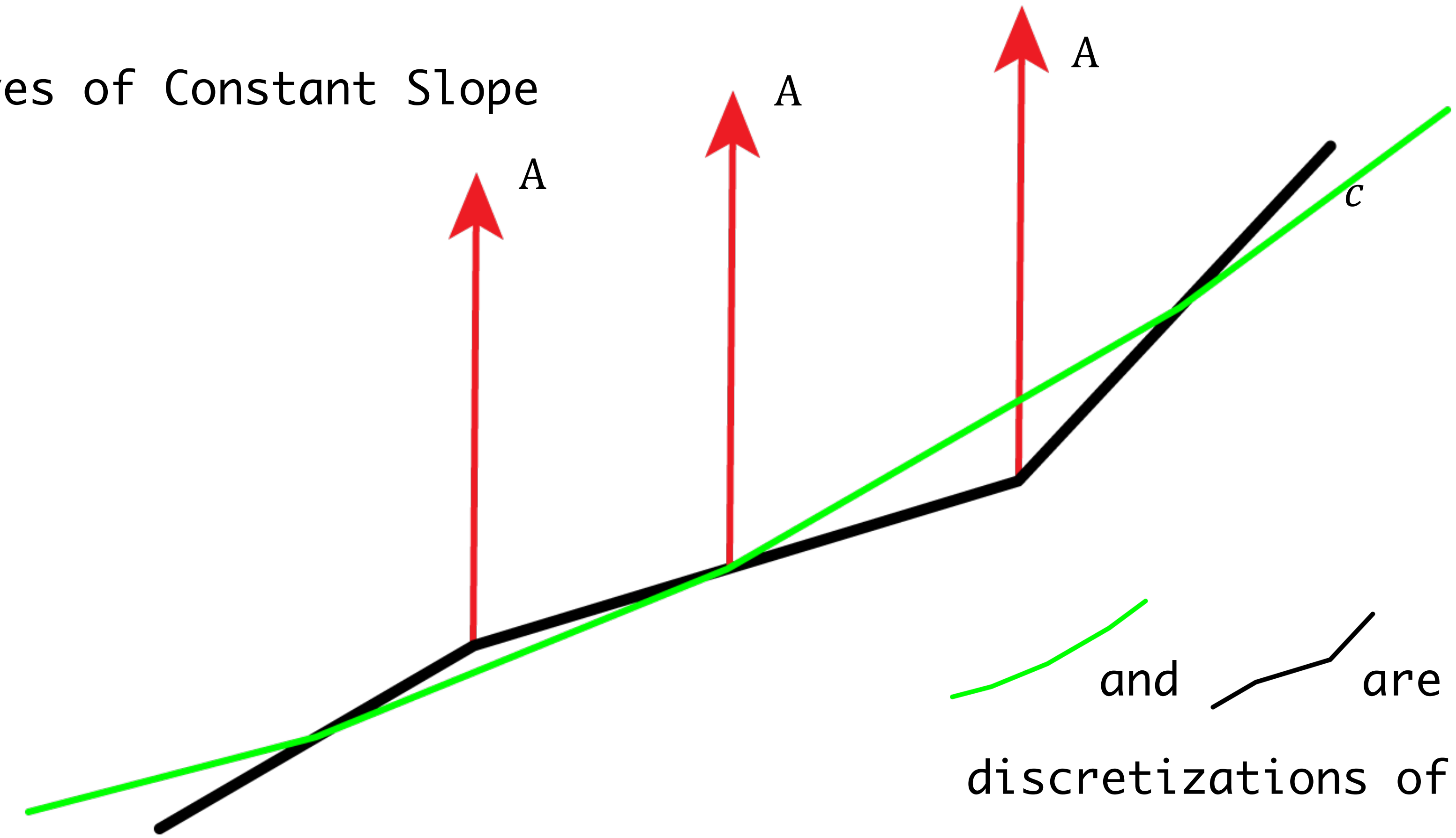
$$(t_1, A) = (t_2, A) \Longrightarrow (t_1 - t_2, A) \Longrightarrow (n^p, A) = 0$$

$\boxed{\text{The same as the discretization of rectifying developable}}$

Applications: gridshells

Torsion-free rectifying strip structures

Curves of Constant Slope

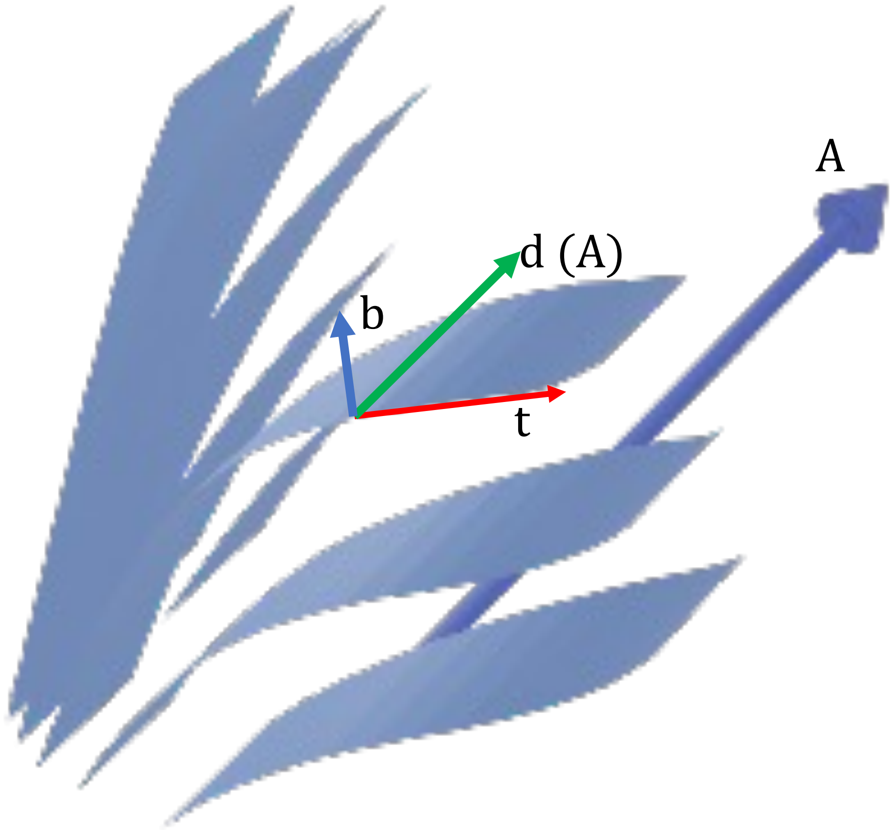


and are the different discretizations of the same curve

Applications: gridshells

Torsion-free rectifying strip structures

Curves of Constant Slope



Property:

A rectifying strip of a COCS takes the axis as the rulings, thus is a cylinder.

Proof:

$(\mathrm{t}, \mathrm{A}) = \mathrm{const} \Longrightarrow (\dot{\mathrm{t}}, \mathrm{A}) = \kappa(\mathrm{n}^p, \mathrm{A}) = 0.$

Assume that $\kappa \neq 0$, $(\mathrm{n}^p, \mathrm{A}) = 0$.

$\left(\dot{\mathrm{n}^p}, \mathrm{A}\right) = 0 \Longrightarrow (-\kappa \mathrm{t} + \tau \mathrm{b}, \mathrm{A}) = 0,$

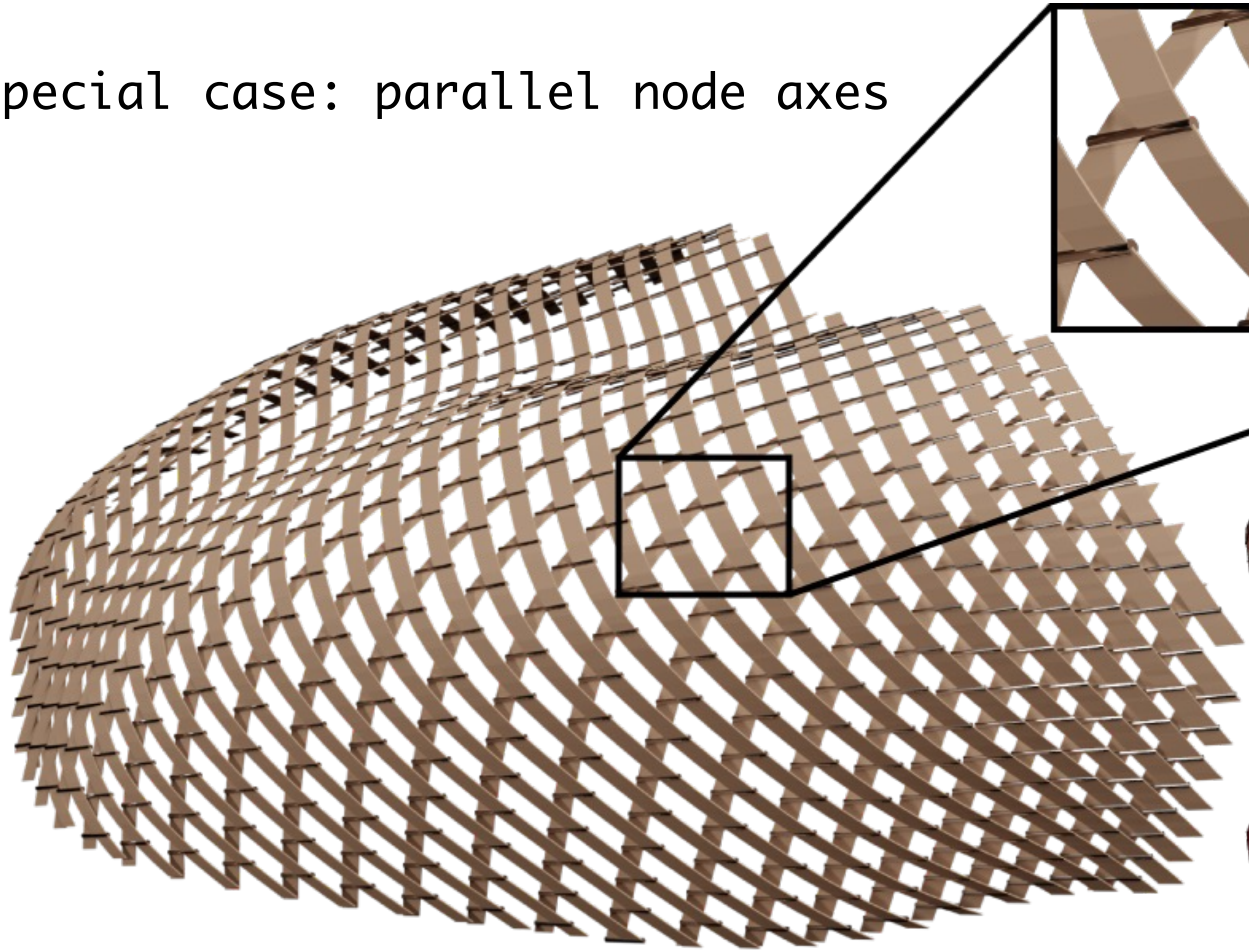
A is in the direction of $\mathrm{d} = \kappa \mathrm{t} + \tau \mathrm{b}$, which is the ruling of the rectifying developable.

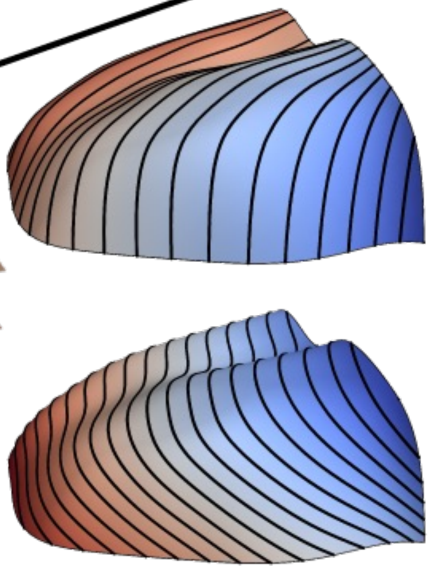
- Parallel rulings + developable = cylinder

Applications: gridshells

Torsion-free rectifying strip structures

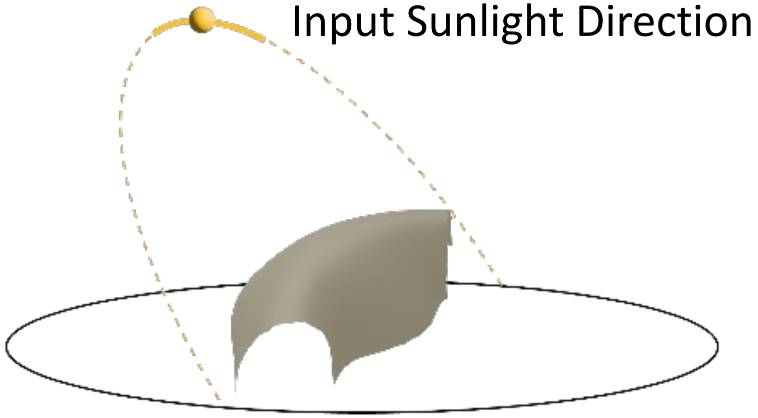
A special case: parallel node axes



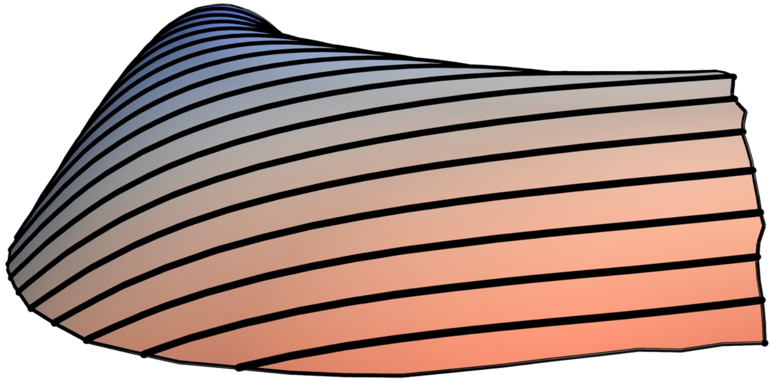


Different slopes of the same axis

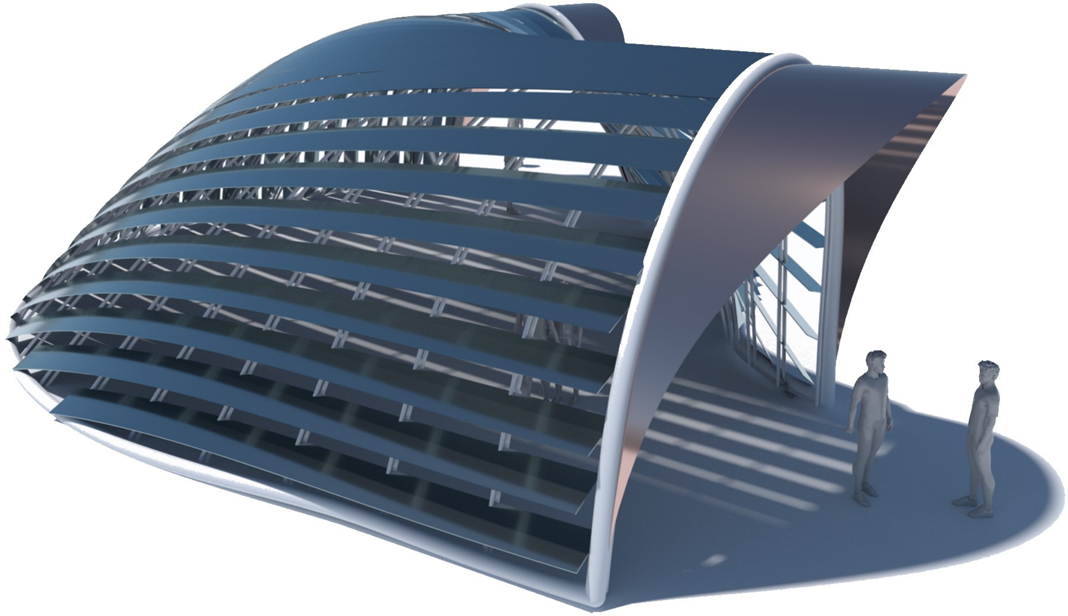
Applications: Shading Systems



Input Surface

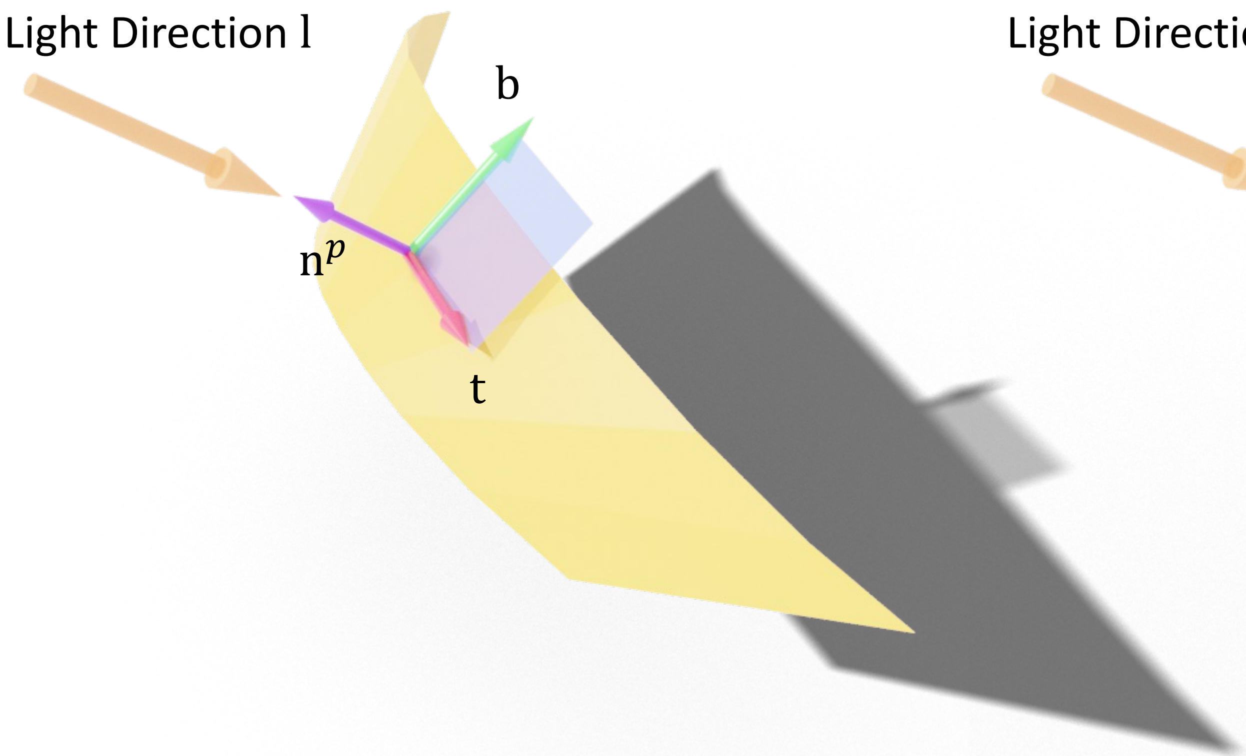


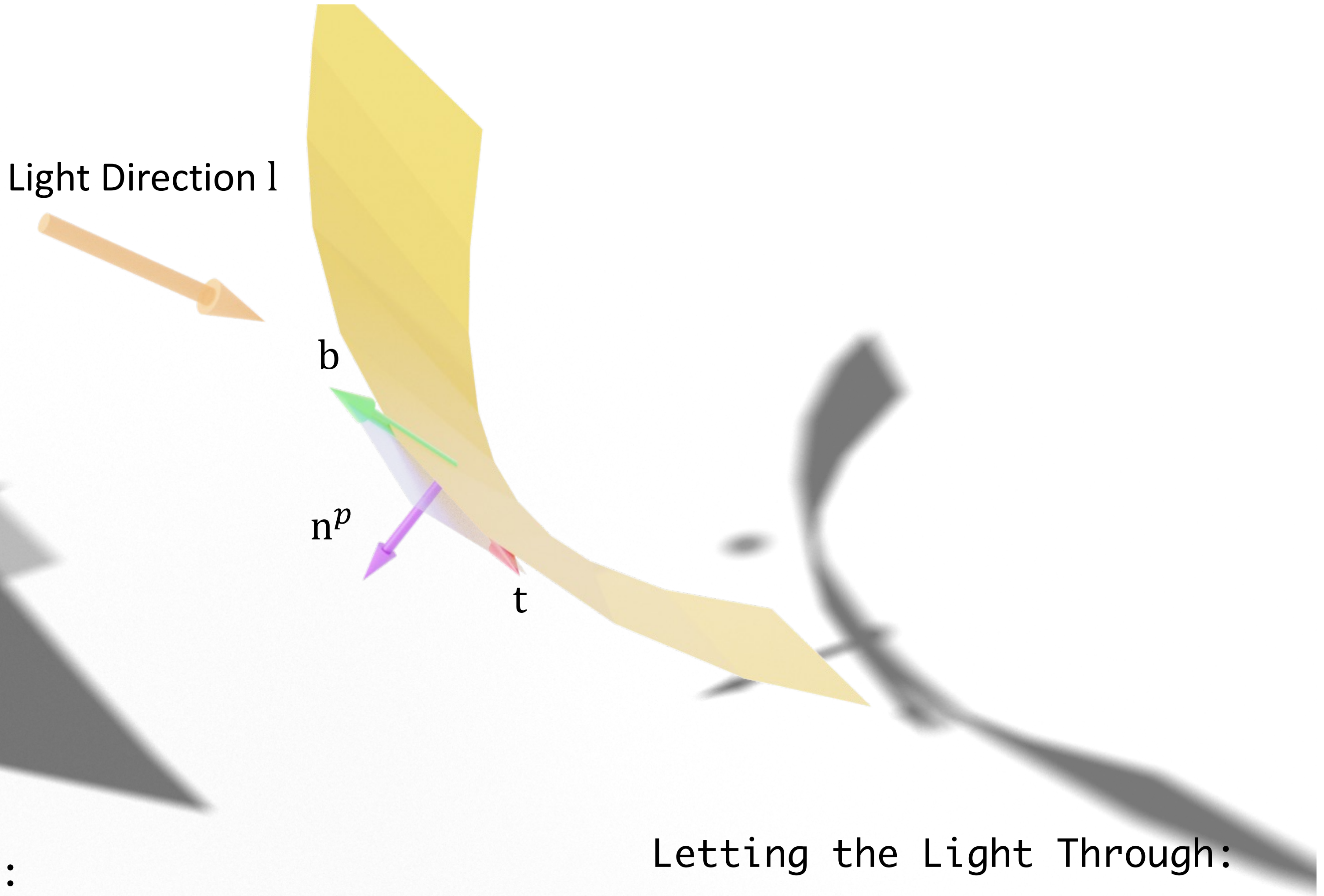
Scalar Field Optimization



Shading System

Applications: Shading Systems





Blocking the Light:

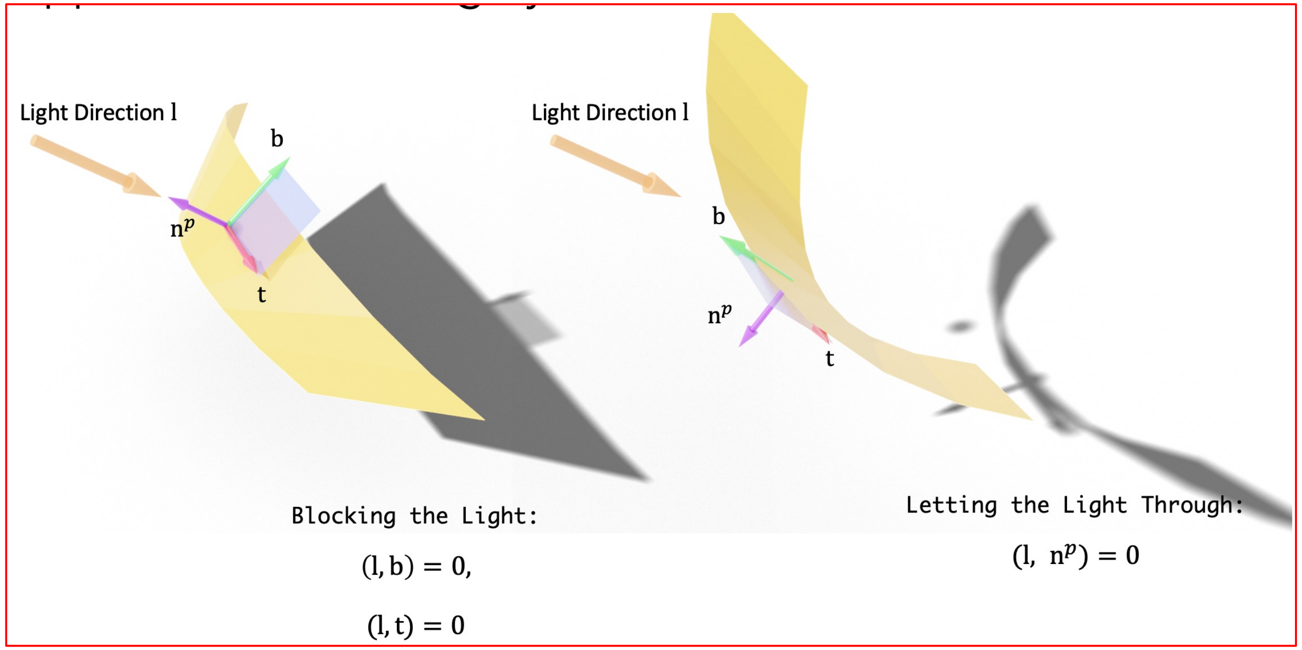
$$(l, b) = 0,$$

$$(l, t) = 0$$

Letting the Light Through:

$$(l,\ n^p) = 0$$

Applications: Shading Systems



The constraints need to be soft penalties, because:

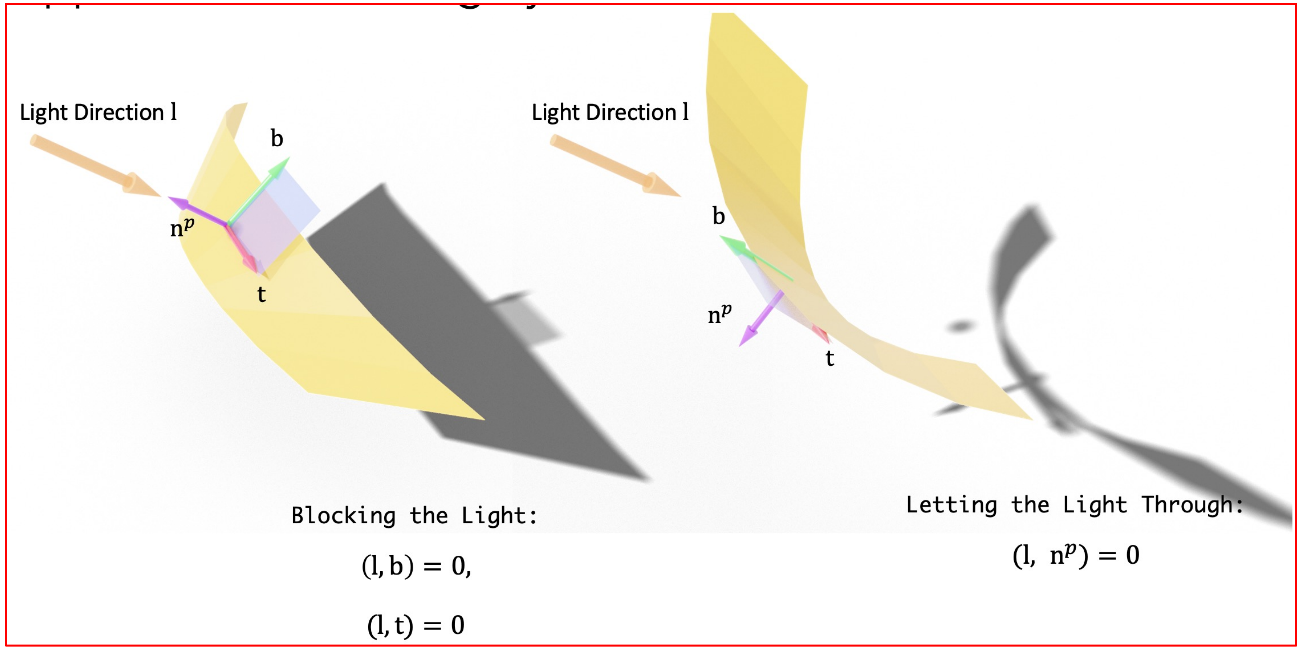
- Only straight strips fulfill "**Blocking the light**" condition
- Only strips of COCS fulfill "**Letting the Light Through**" condition

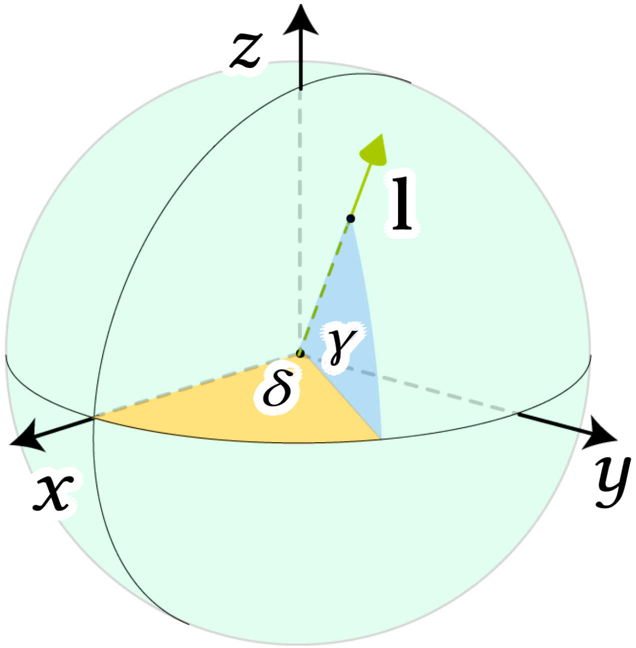
Light l is always orthogonal to the rectifying planes
$\Longrightarrow$
Strip is straight

COCS: $(t_1, A) = (t_2, A) \Longrightarrow (t_1 - t_2, A) \Longrightarrow (n^p, A) = 0$

$\Longleftarrow$ $\Longleftarrow$ (reverse the COCS condition)

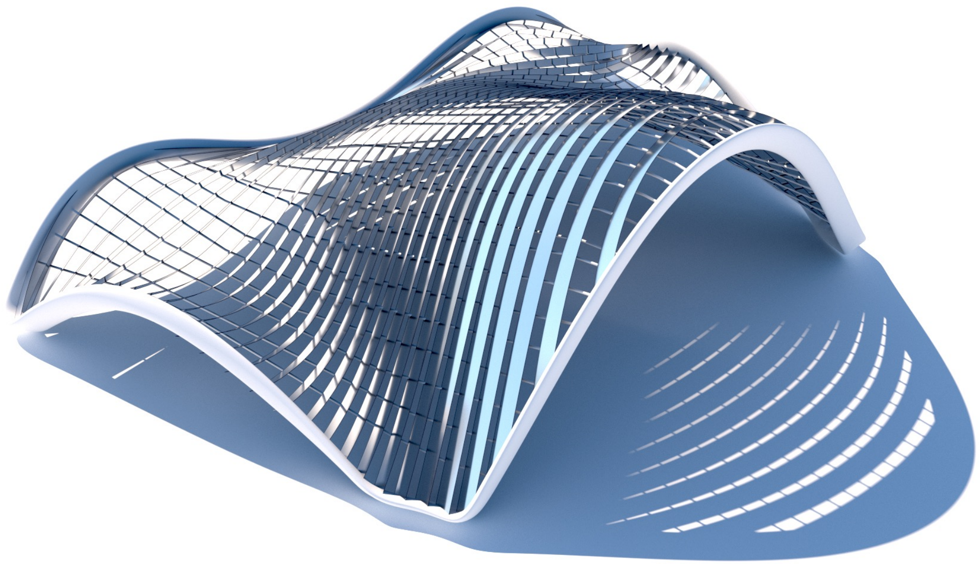
Applications: Shading Systems



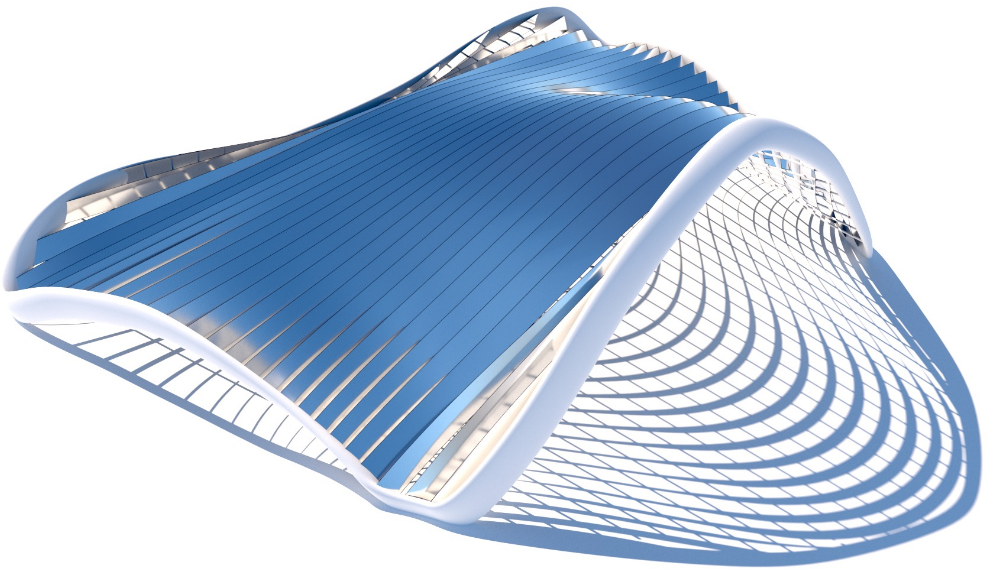


$$\gamma \in [\gamma_{\min}, \gamma_{\max}] \iff \mathbf{l}_z \in [z_{\min}, z_{\max}],$$
$$\delta \in [\delta_{\min}, \delta_{\max}] \iff l_y / l_x \in [g_{\min}, g_{\max}],$$

Applications: Shading Systems



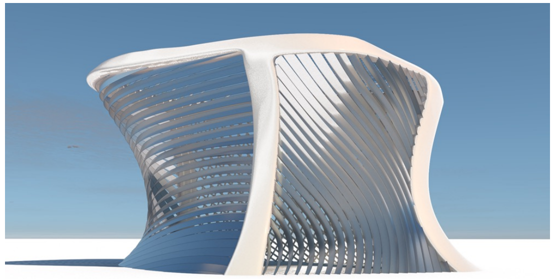
Blocking the Light



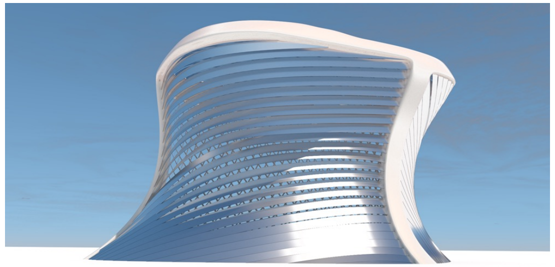
Letting the Light Through

Makkah, 12:00, Dec 1st.

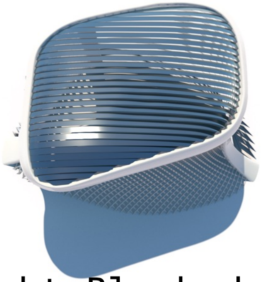
Applications: Shading Systems



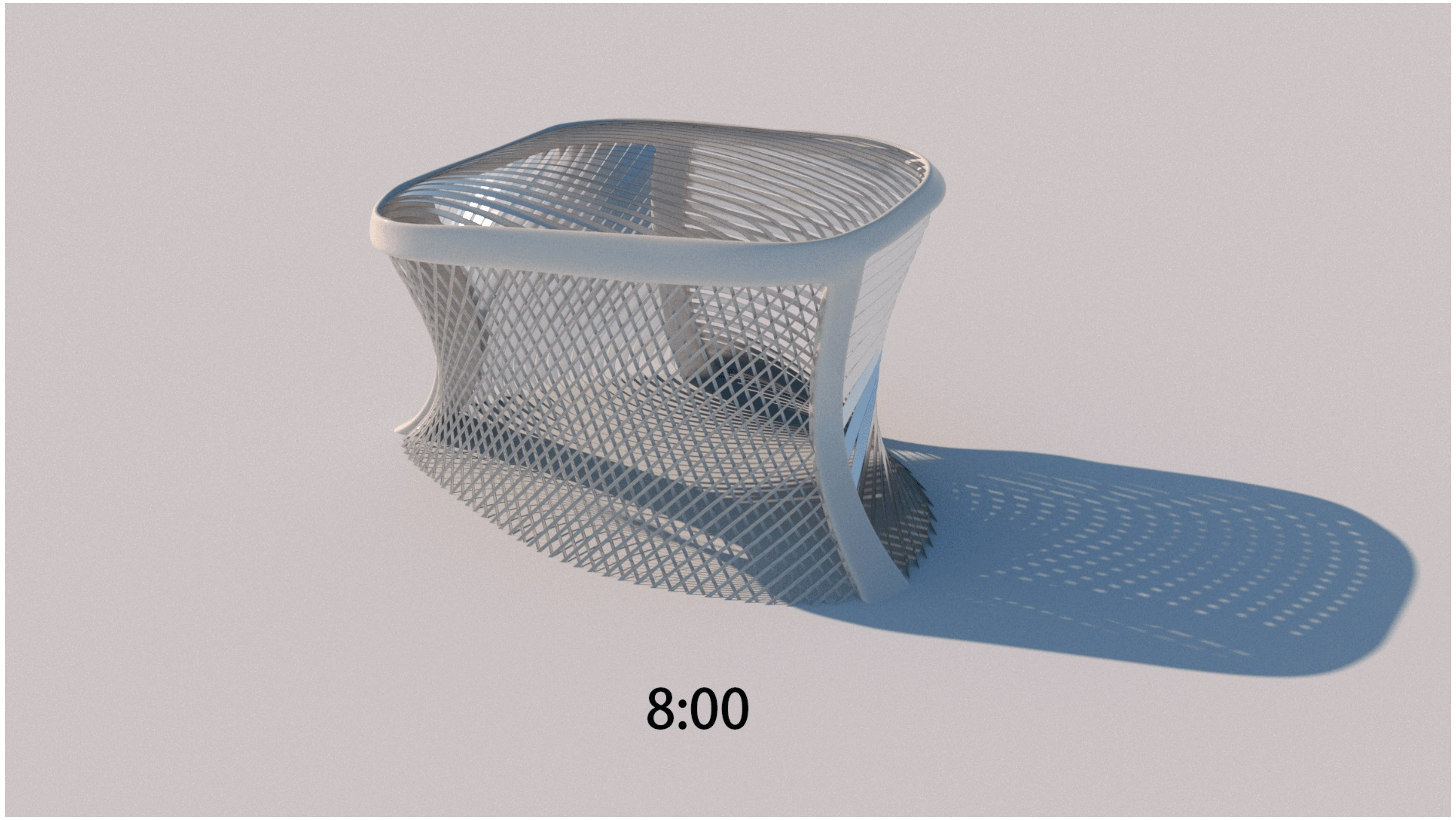
Light Through



Light Blocked

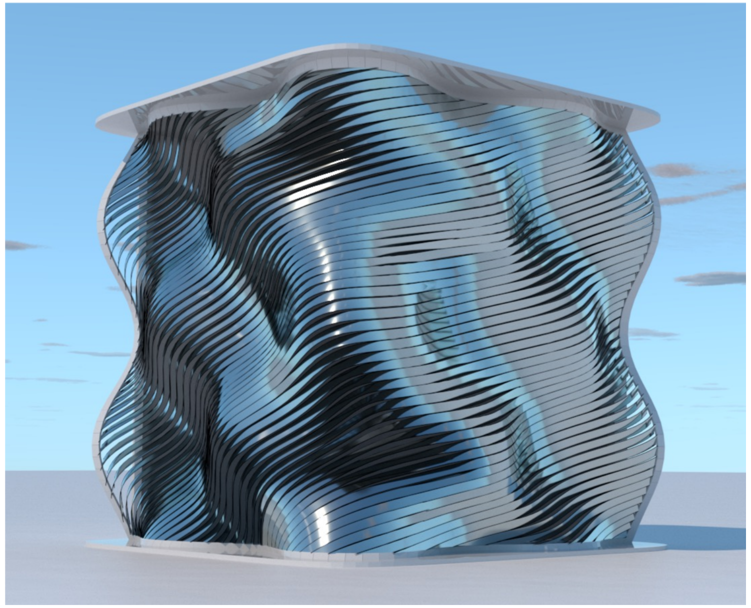


Light Blocked

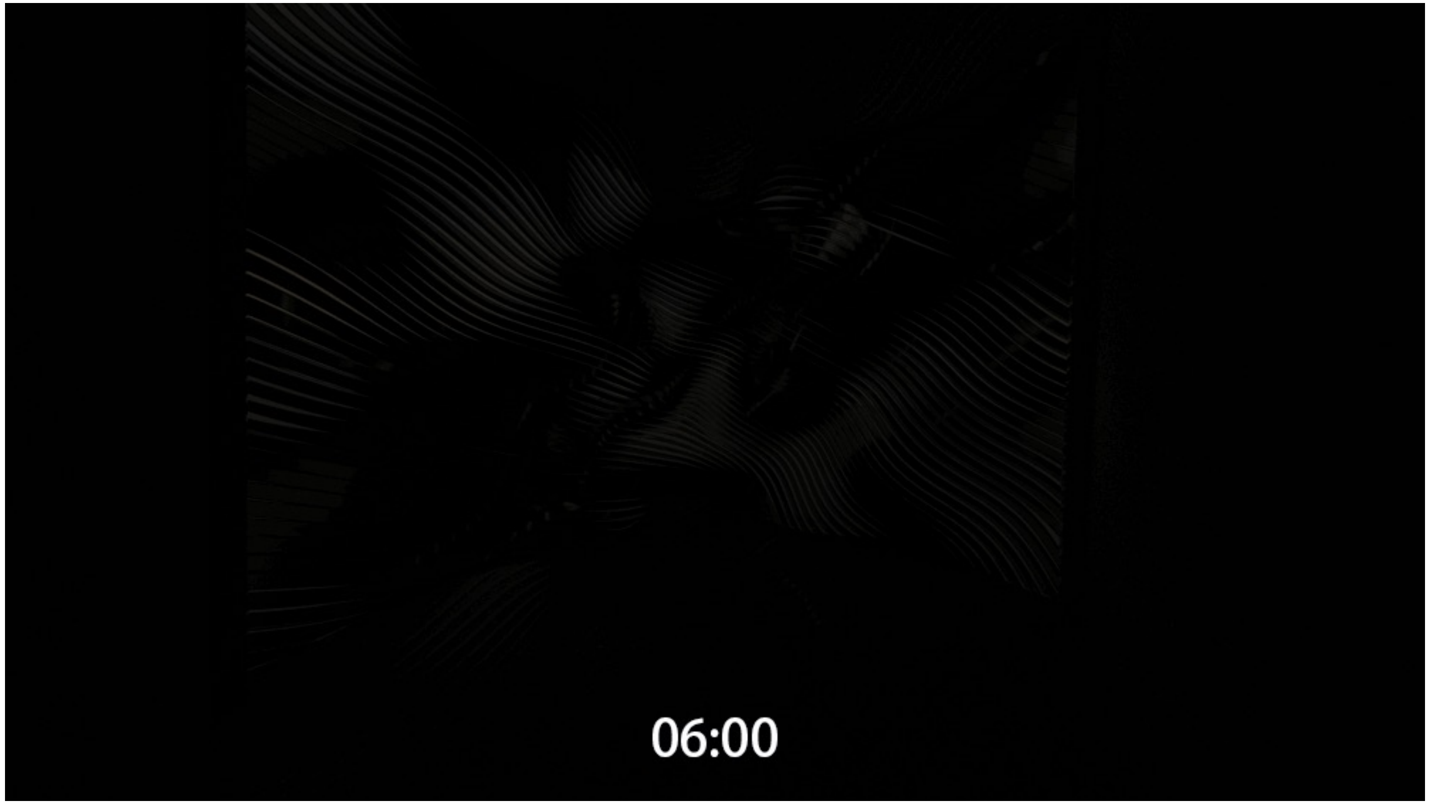


Vienna, Aug 1st.

Applications: Shading Systems

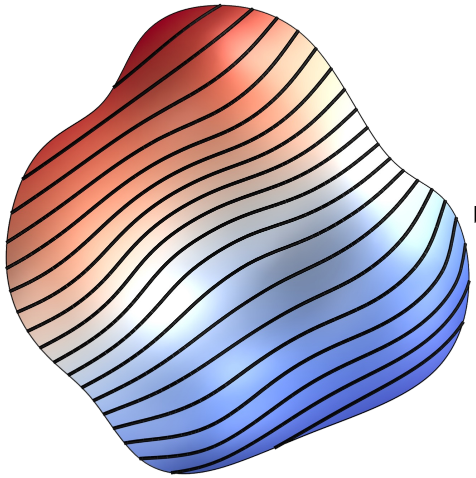


Let the sunlight through in the morning, and block the sunlight in the afternoon.

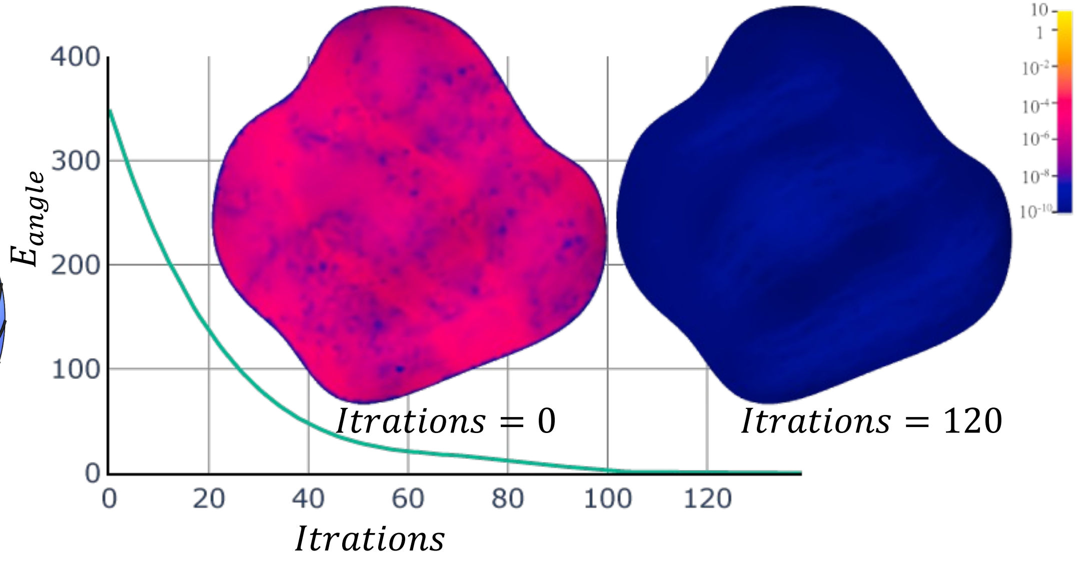


London, Aug 15th.

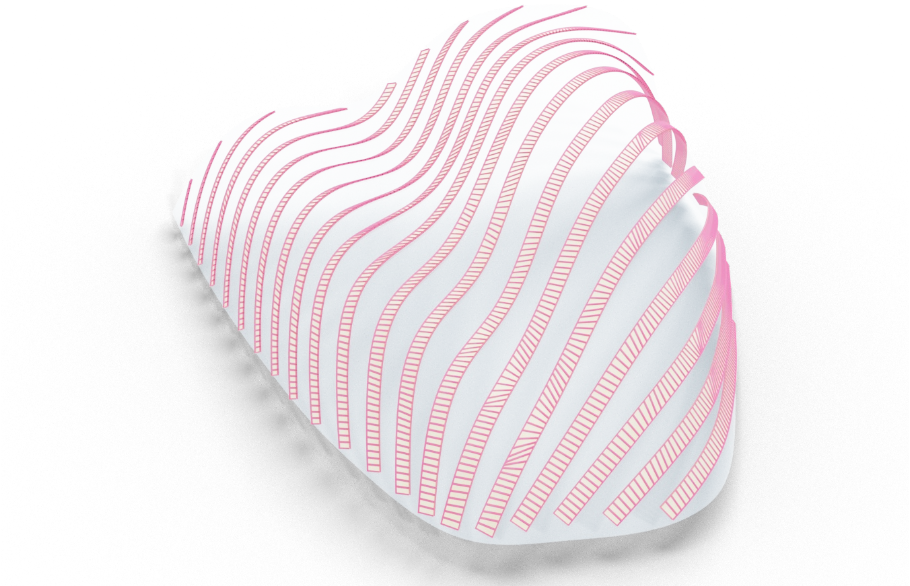
Result Evaluations



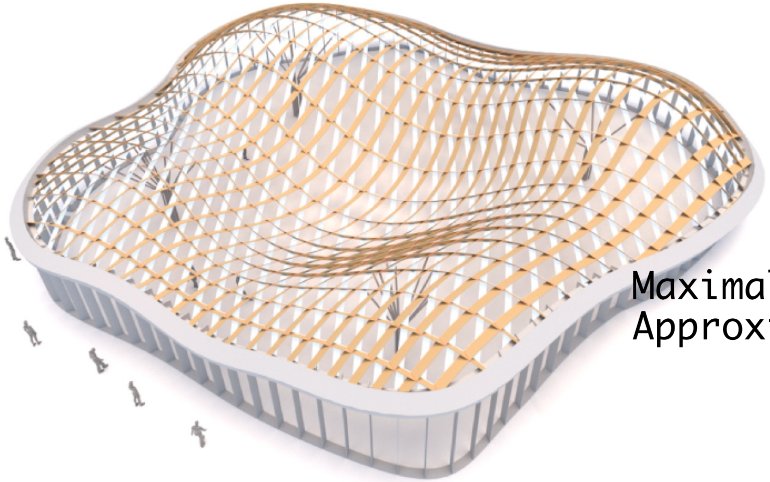
$\theta = 60^{\circ}$



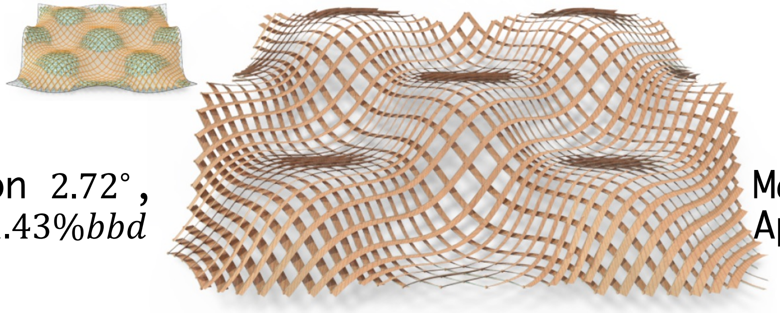
Convergence of scalar field optimization ($E_{angle} \approx 3e-6$)



$\theta = 72^{\circ}$, maximal angle deviation 1.59°
Approximation error: $1.5\% bbd$



Maximal angle deviation 2.72°,
Approximation error: $1.43\% bbd$



Maximal angle deviation 1.59°,
Approximation error: $0.14\% bbd$

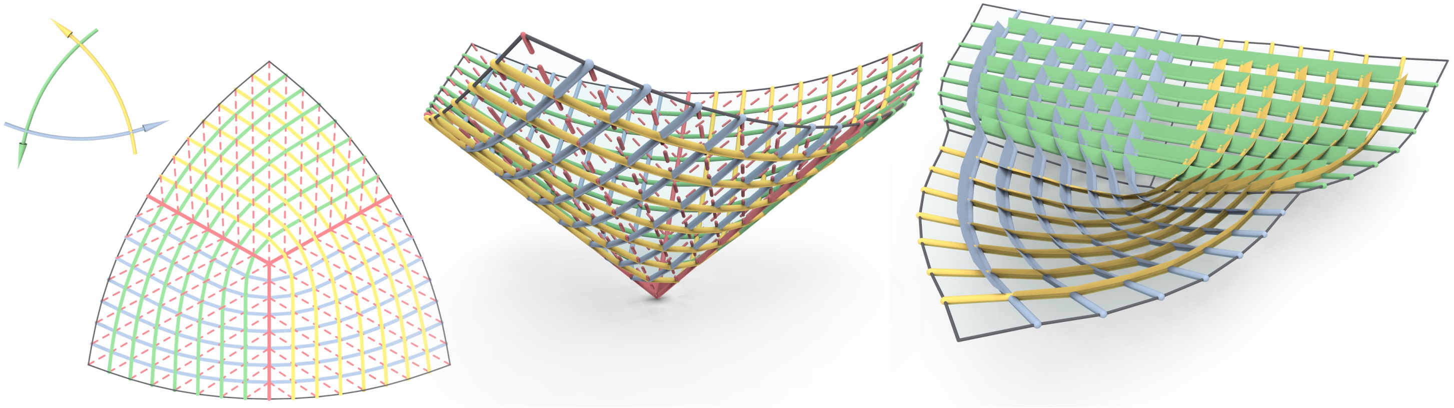
bbd: the length of the Axis-Aligned Bounding Box diagonal

Conclusion

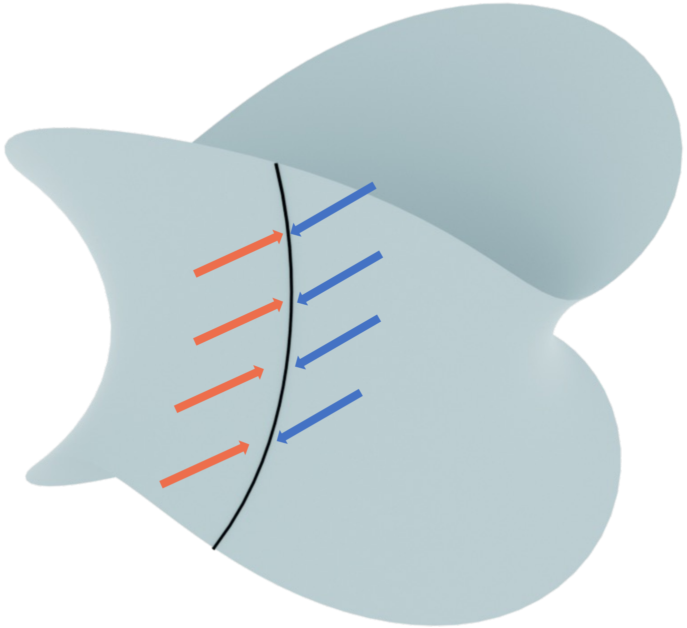
Limitations:

Avoid singularities: limited by the fundamental geometry nature of level sets.

Topology: need to be a topological disk.



Composition of Regular parts

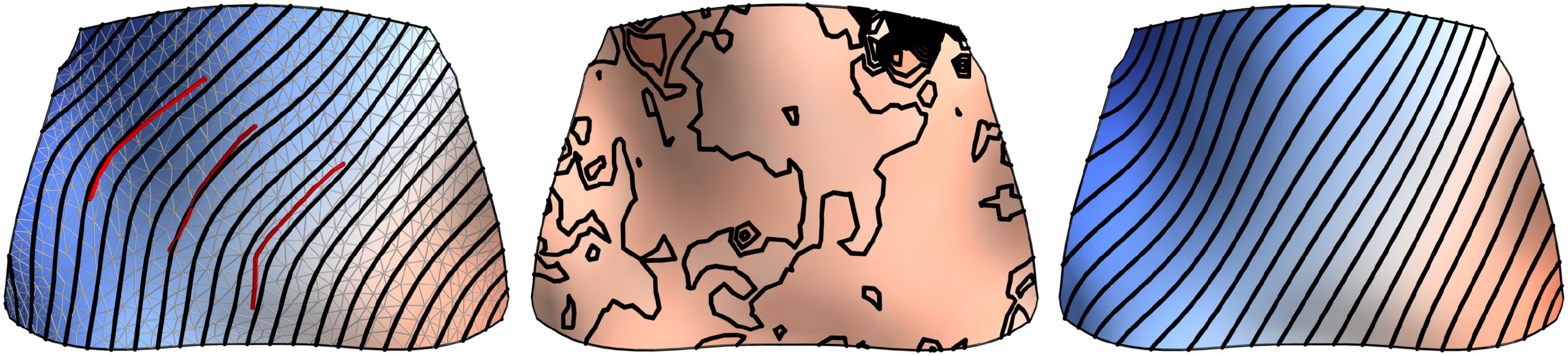


Stitch Level Sets on the Boundaries
(Future Work)

Conclusion

Limitations:

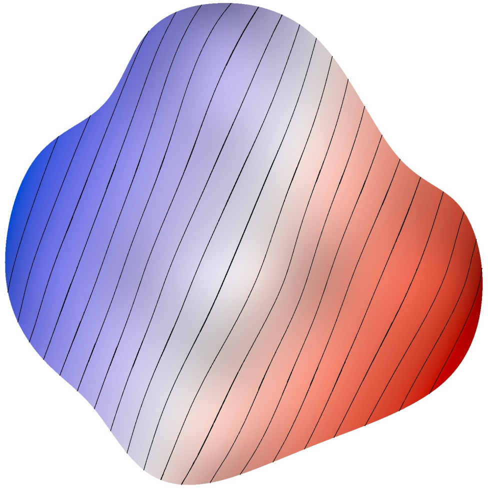
Non-linear optimization is not fully automatic

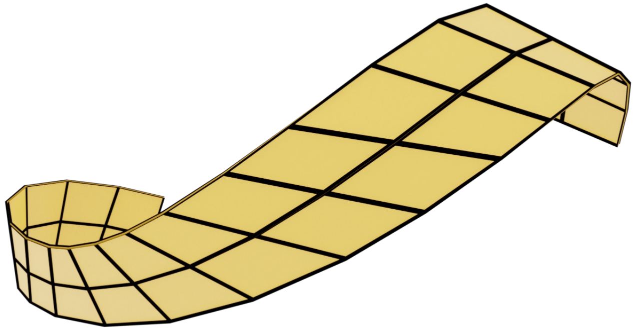


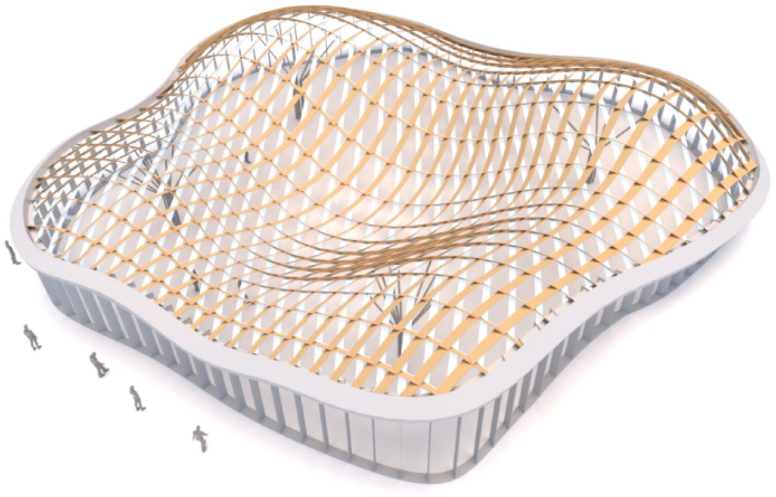
Initialization Using Default Parameters Choosing Parameters Manually

Conclusion

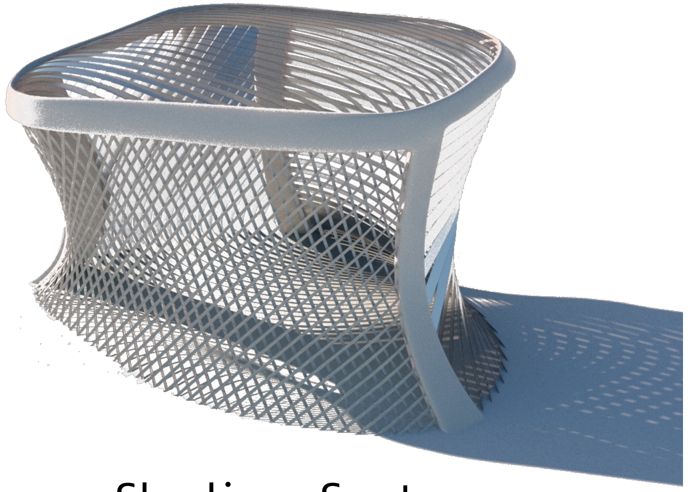
- Straight Flat Strips
- Controllable Inclinations
- Gridshell Design
- Shading System Design



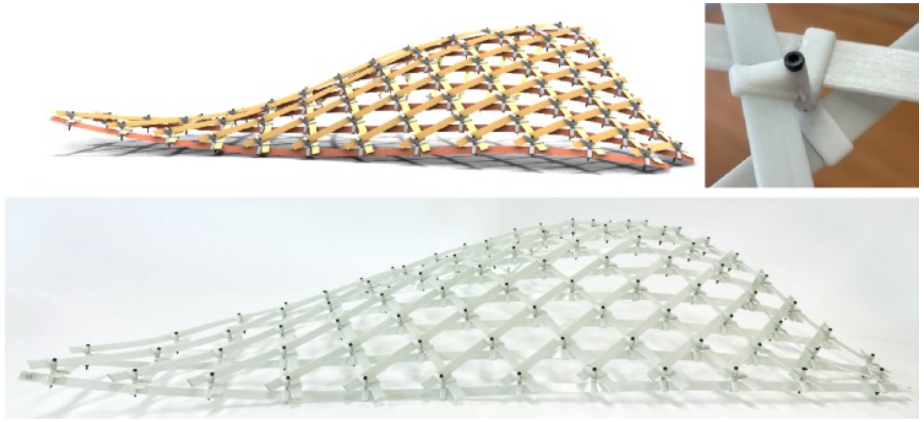




PPG-Gridshell ($\theta_1 = 45°, \theta_2 = 60°$)



Shading Systems



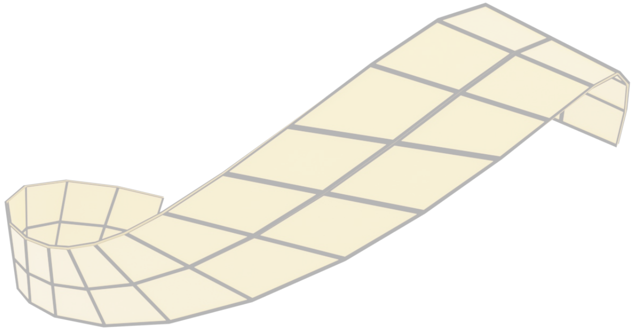
Physical Models

Open source project: https://github.com/wangbolun300/RectifyingStripPatterns

Conclusion

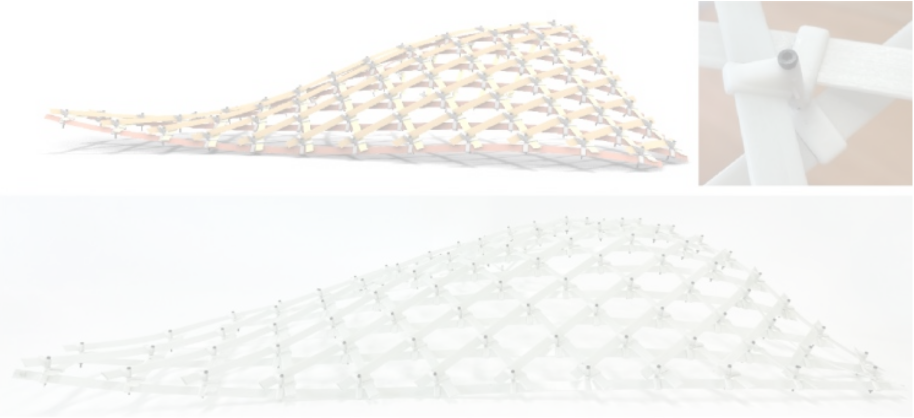
- Straight Flat Strips
- Controllable Inclinations
- Gridshell Design
- Shading System Design

Thanks



PPG-Gridshell ($\theta_1 = 45°, \theta_2 = 60°$)

Shading Systems

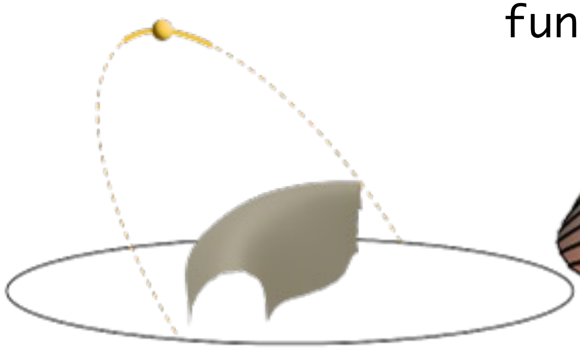


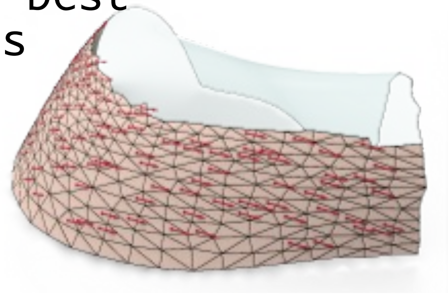
Physical Models

Open source project: https://github.com/wangbolun300/RectifyingStripPatterns

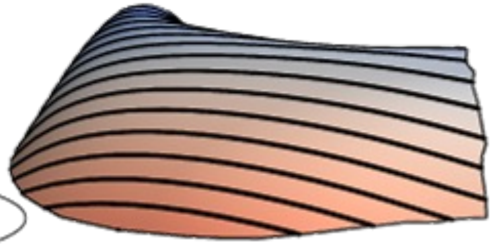
Applications: Shading Systems

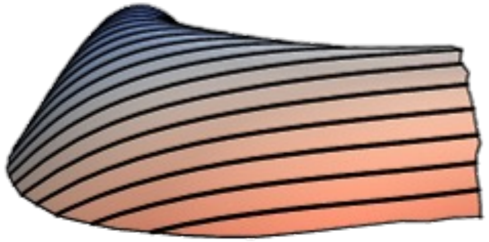
1. Initialize the best shading directions



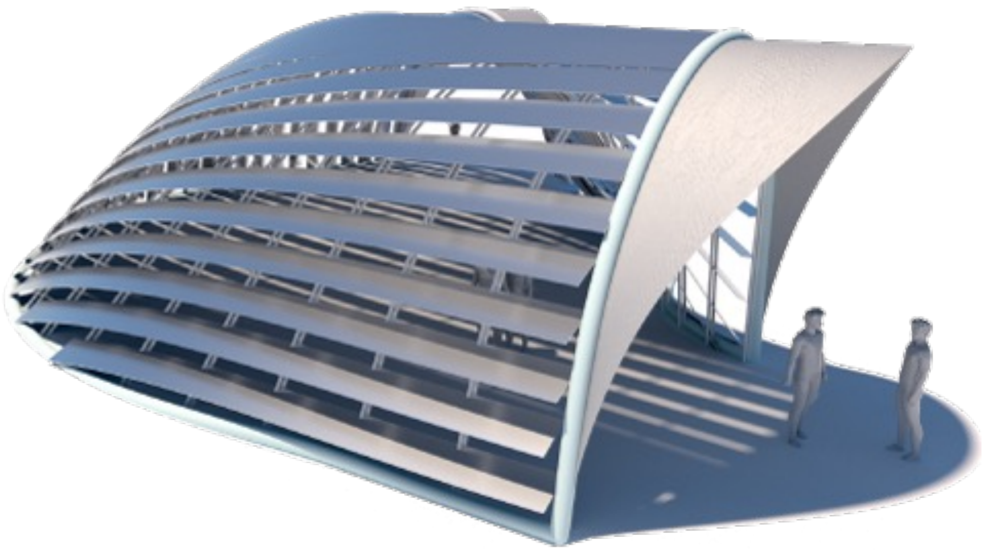


2. Initialize the scalar function F





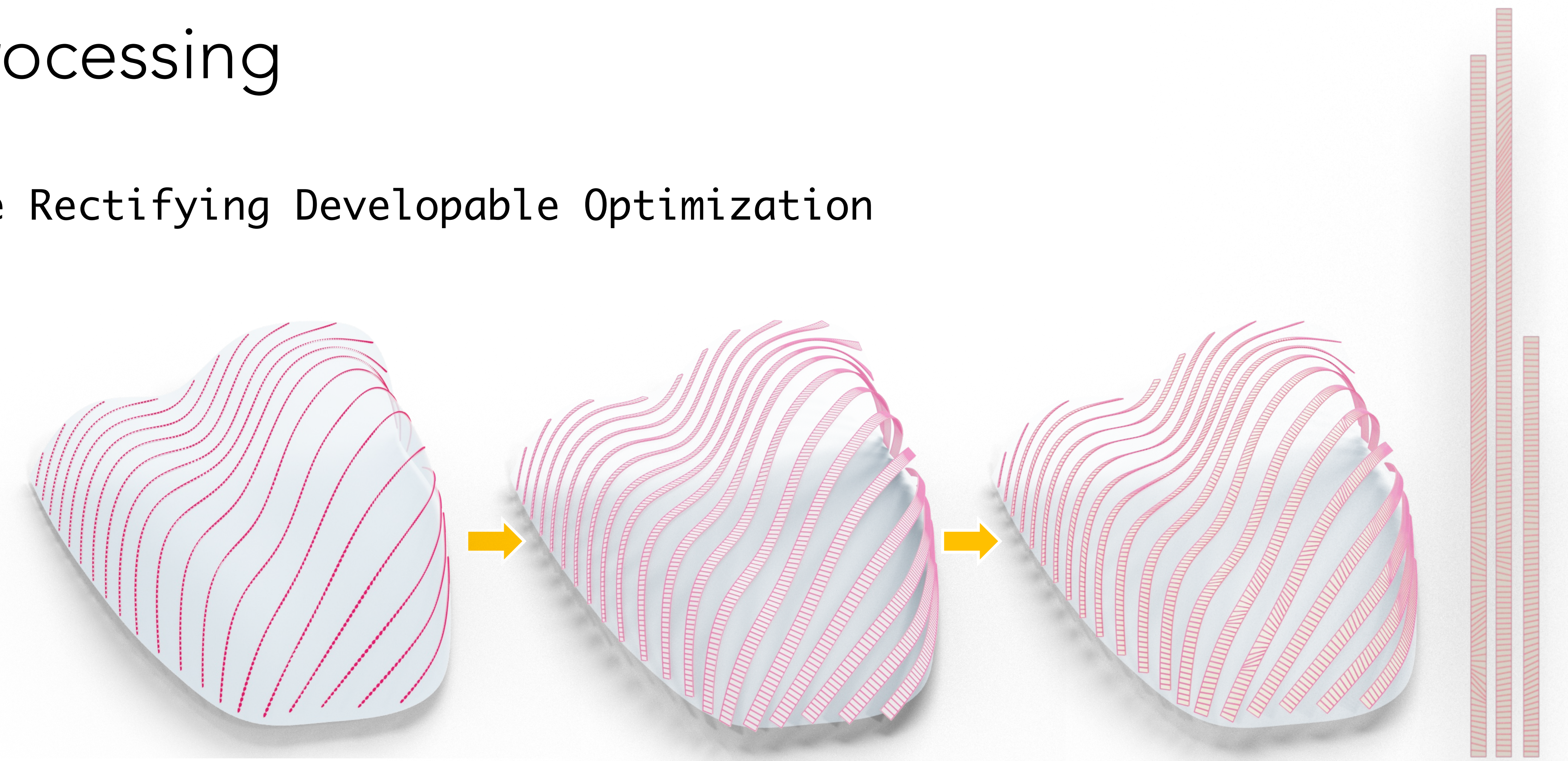
3. Optimize the level sets



4. Extract rectifying strips

Postprocessing

Discrete Rectifying Developable Optimization



Extract level sets

Resample and optimize for accurate binormals

Optimization for rectifying developables