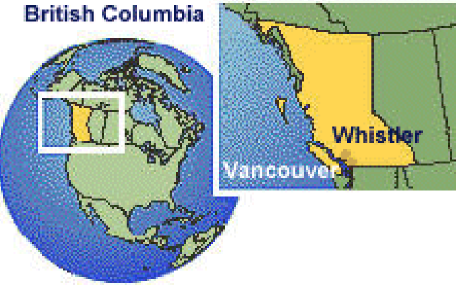
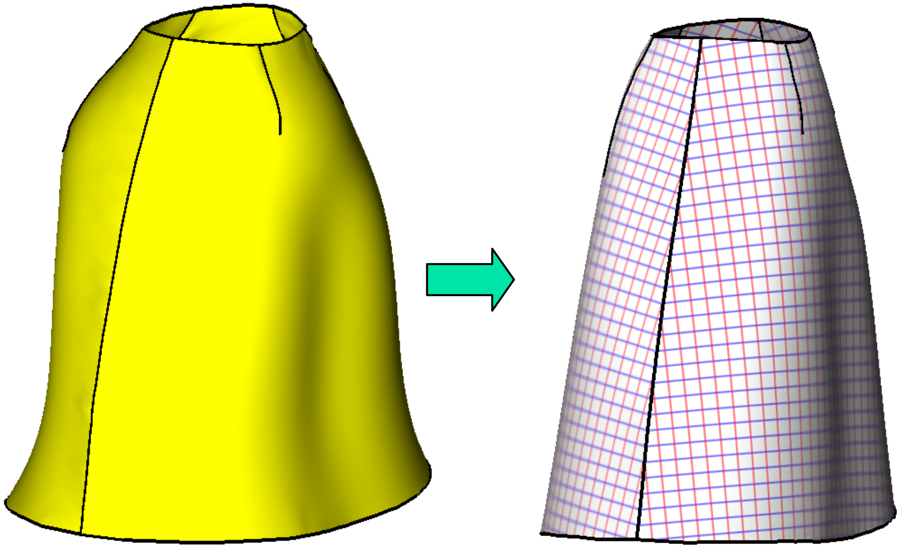


University of British Columbia

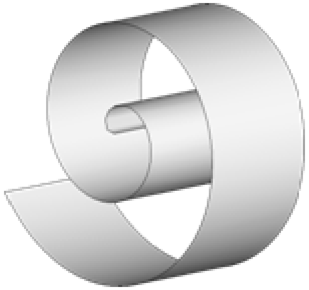
Approximate and Exact Modeling of General Developable Surfaces

Alla Sheffer



Motivation

- Developable Surfaces
  - Unfolded to plane without distortion

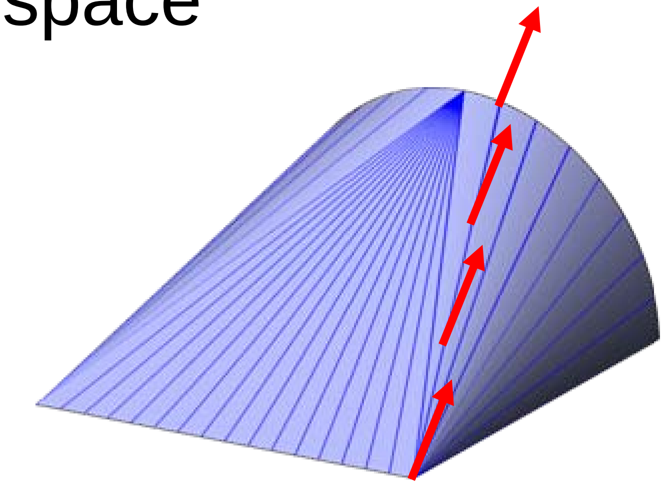


- Ubiquitous

- Difficult to model with existing tools

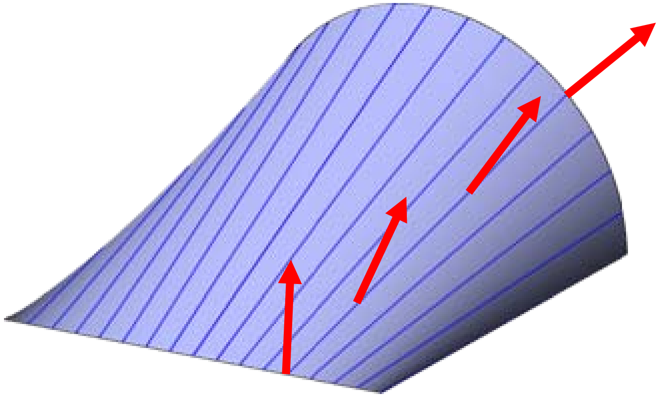
Background: Differential Geometry

Ruled Surface: Generated by continuous motion of straight line in space



Developable

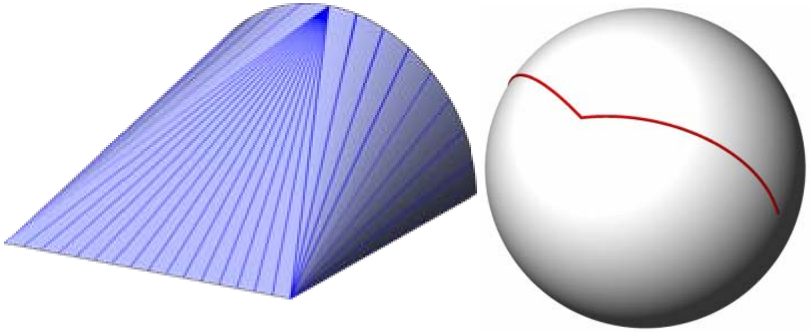
Ruled surface with constant normals along rulings



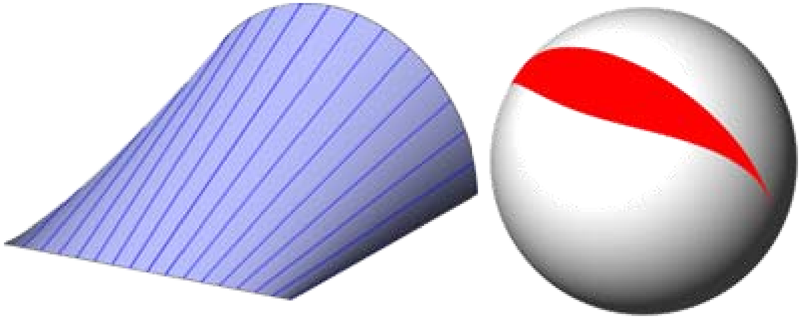
Warped

Background

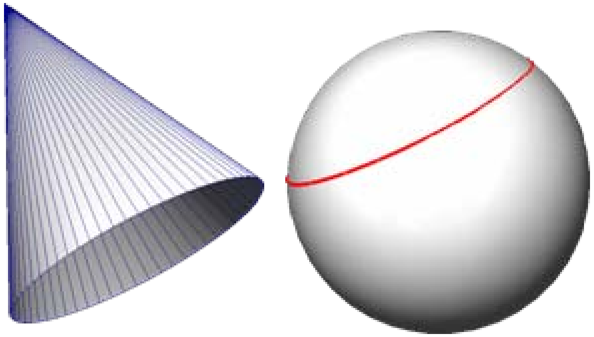
- Gauss map of developable surface is 1D



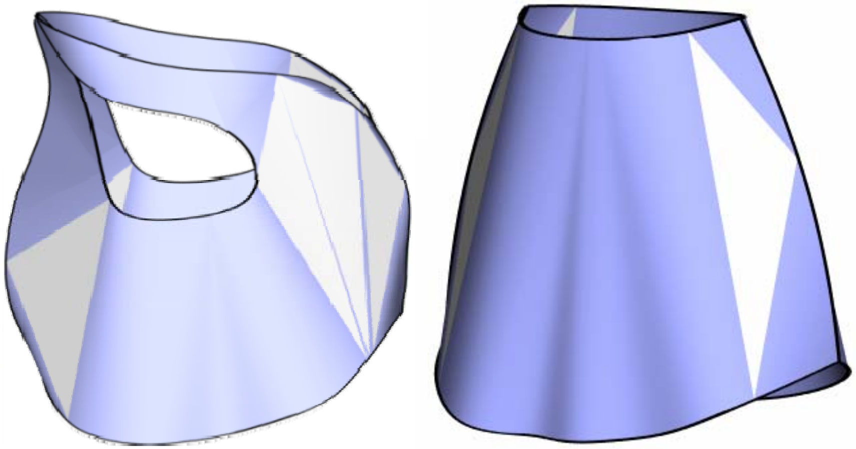
Developable



Not Developable



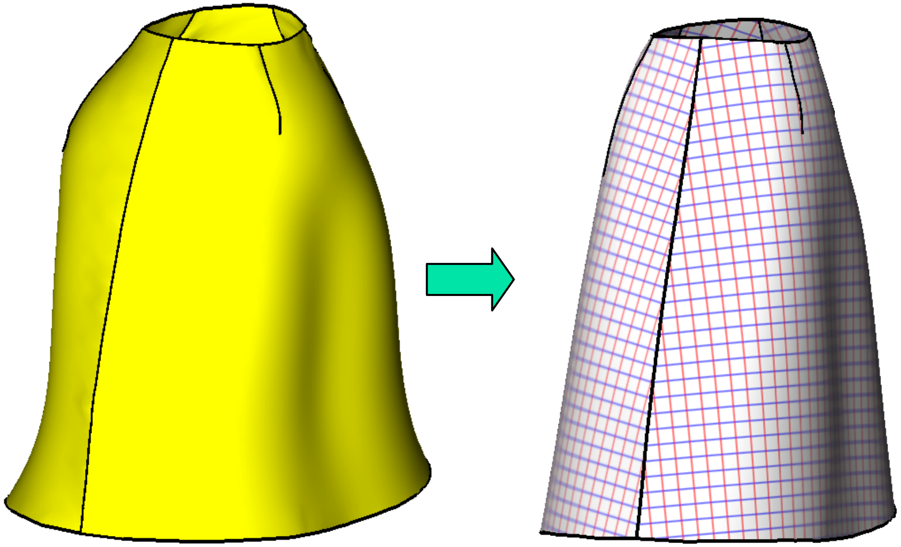
Torsal Ruled Developable



General Developable

Approximation

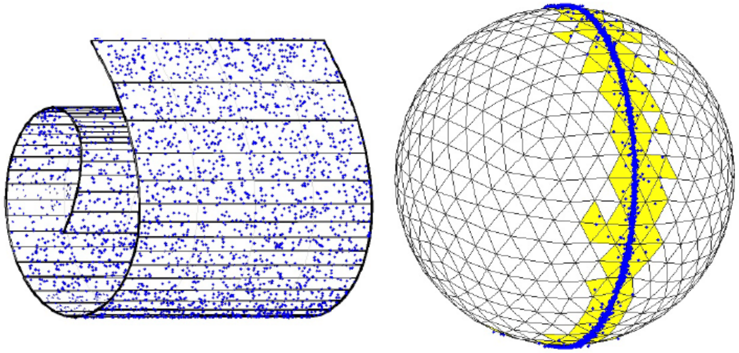
- Input: Non-developable surface
- Output: (Nearly-)Developable approximation



Related work

Torsal developables

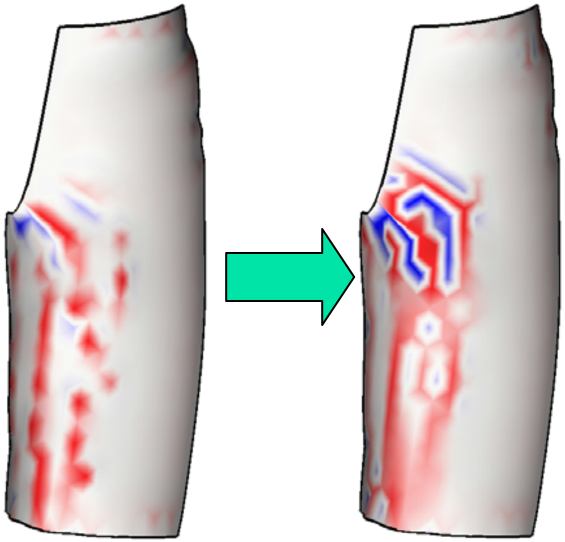
- Dual space



[Pottmann & Randrup:98, Chen:99,Peternell:04]

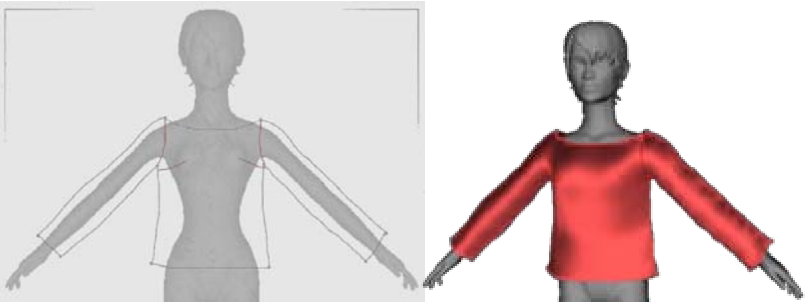
General developables

- Gauss curvature minimization

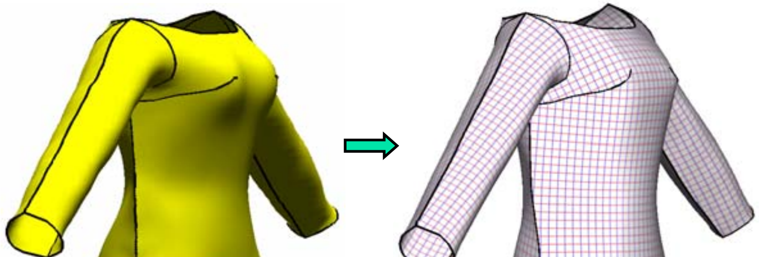


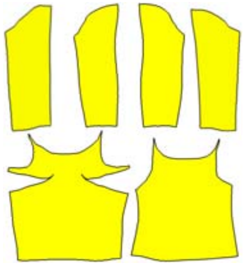
[Wang & Tang ’04]

Context – Garment Modeling [Decaudin'06]



1. 3D from sketch
   - Output: non-developable surface + seams

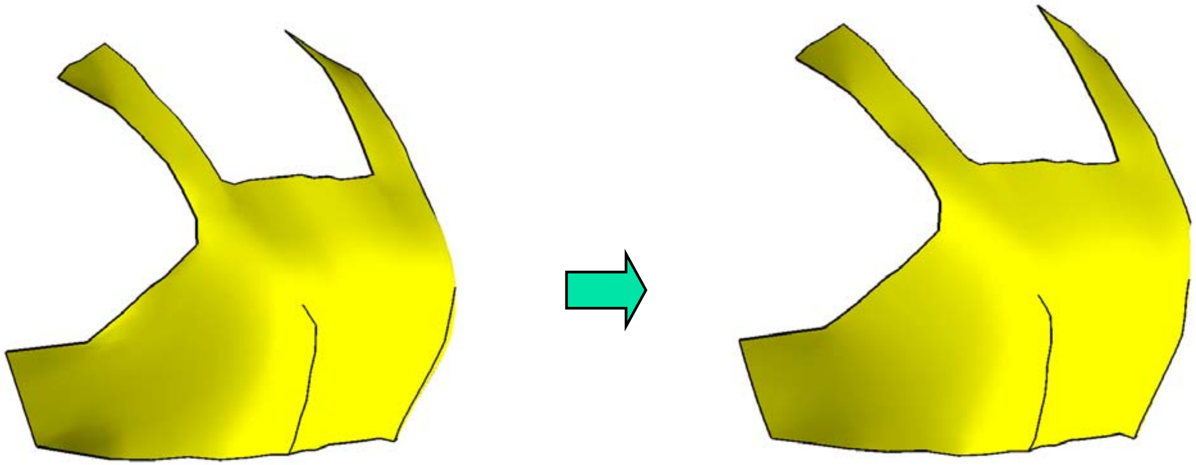




2. Piecewise developable approximation
   - Keep parts connected
   - Output: "paper garment" + 2D patterns

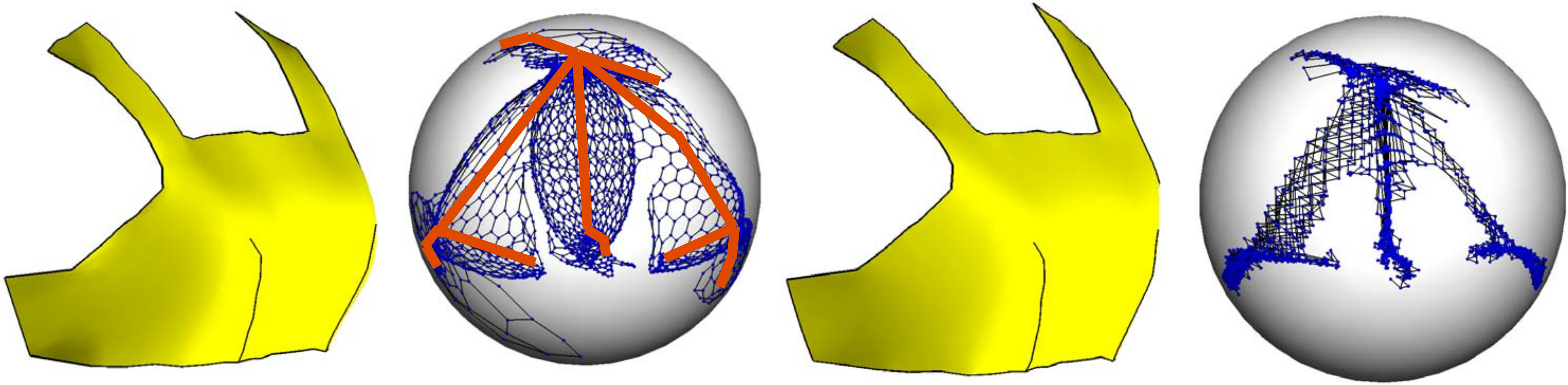
3. Introducing Physics
   - Procedural

Developable Approximation - Idea



- Use dual space
- BUT do not assume normal map/Blaschke image is curve
- Use local approximation ONLY

Local Approximation in Dual Space

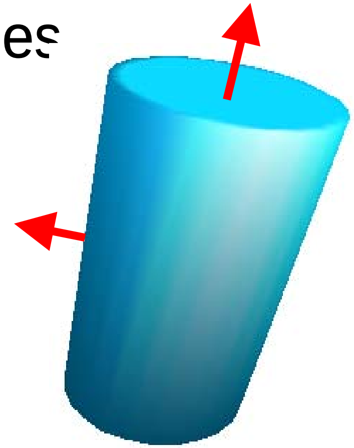


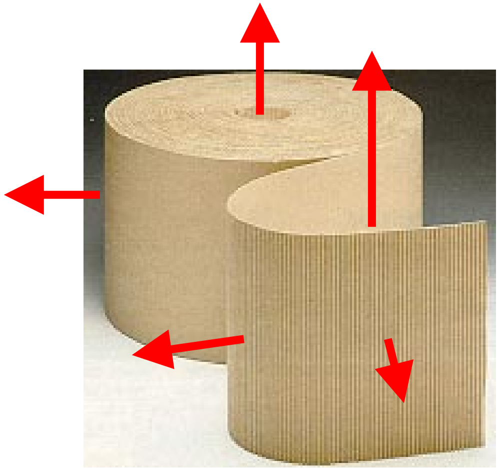
- Locally approximate normal map
  - Inspired by MLS
- Project normals to approximation
- Adjust vertex positions to satisfy new normals

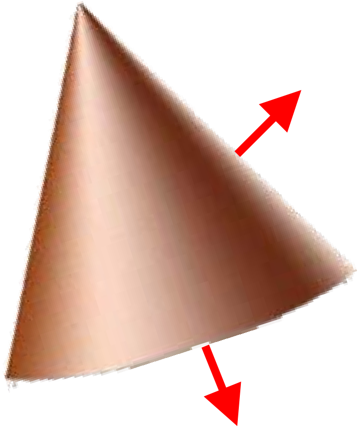
Dual Space Approximation

- ***Observation:***
  Curve network on unit sphere is locally approximated by arcs + branching points = developable surface locally approximated by *developable of constant slope* (DCS)
- DCS - constant angle between surface normal and axis
  - Planes
  - Generalized cylinders
  - Cones

Defined by normal and angle <*N*,*θ*>







Method overview

- Input: mesh (with seams/darts)
- Output: developed mesh

- While not developable (enough)
  - For each triangle
    - Compute best locally approximating DCS
      - Use neighbourhood
        - Bound by normal deviation/radius
        - Anisotropic
    - Rotate to fit DCS
  - Reconcile vertices

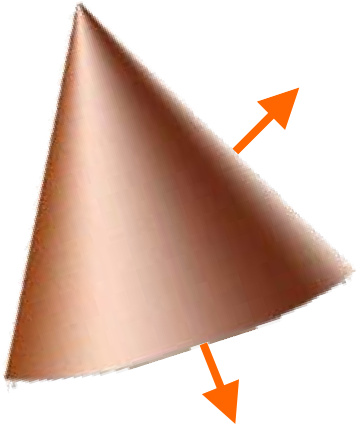
Per-Triangle Processing

- Compute approximating DCS

$$\min_{N,\theta}\sum_{n_j}\left[\frac{\left(n_j\cdot N-\cos\theta\right)}{2}\right]^2 subj.to\,\|N\|=1,|\cos\theta|<.5$$

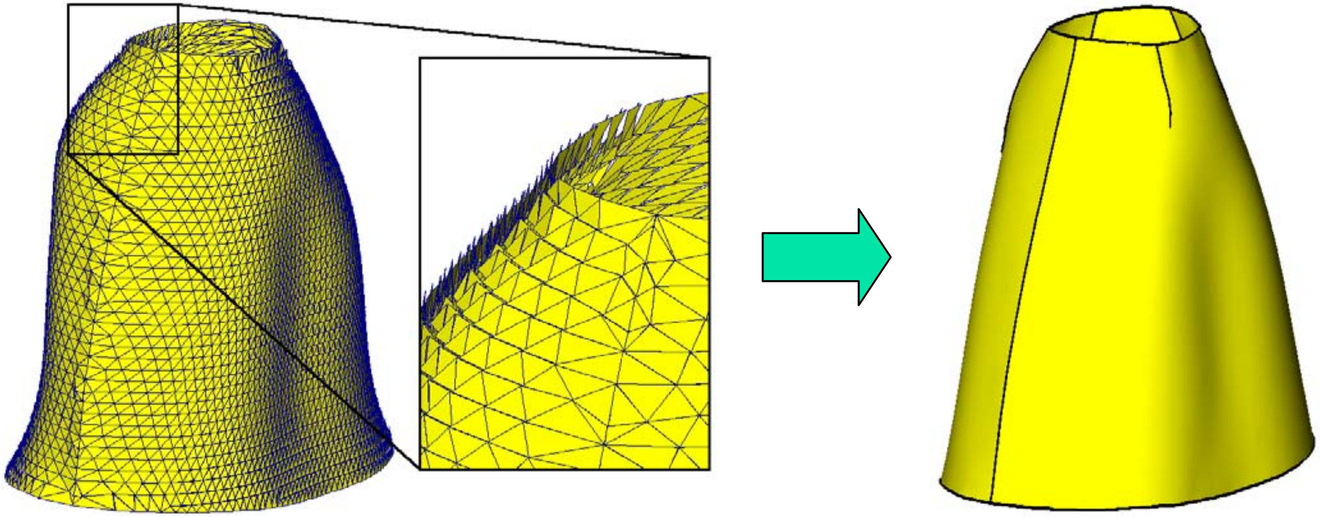
- Rotate to proxy - define new normal $\tilde{n}$ & move vertices

$$\min_{v_i',d}\sum_{i=1}^{3}\|v_i-v_i'\|^2 subj.to\,v_i'\cdot\tilde{n}-d=0$$



  - Support positional constraints

Reconcile vertices

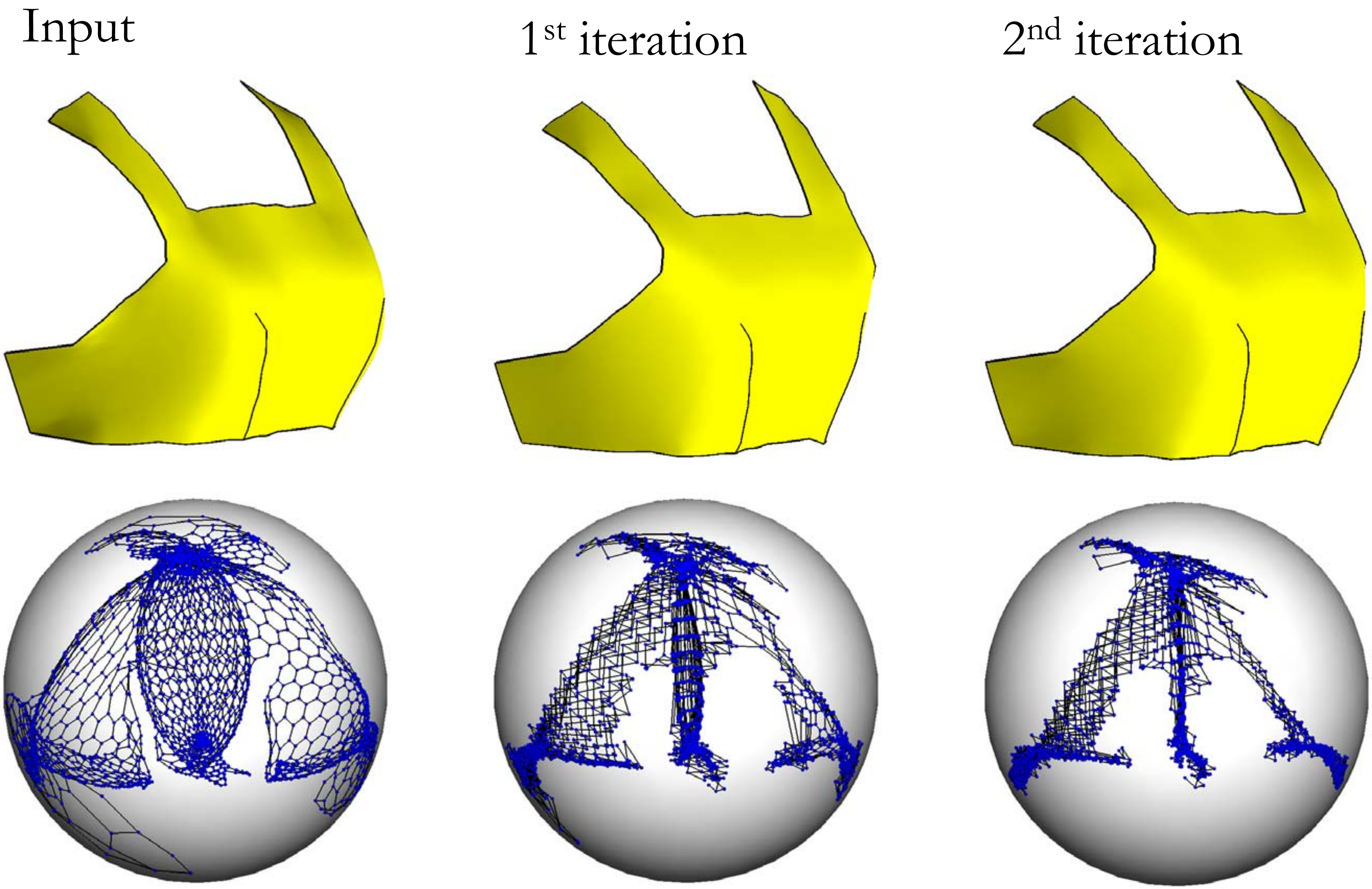


- Keep orientation as much as possible
  - Similar to [Sumner & Popovic:04, Yu:04]
- Solve

$$\min_{\tilde{v}} \sum_{j=1}^{T} \left\| \tilde{V}_j V_j^{-1} - I \right\|_F^2$$

  - Global - preserve seams intact

Iterate

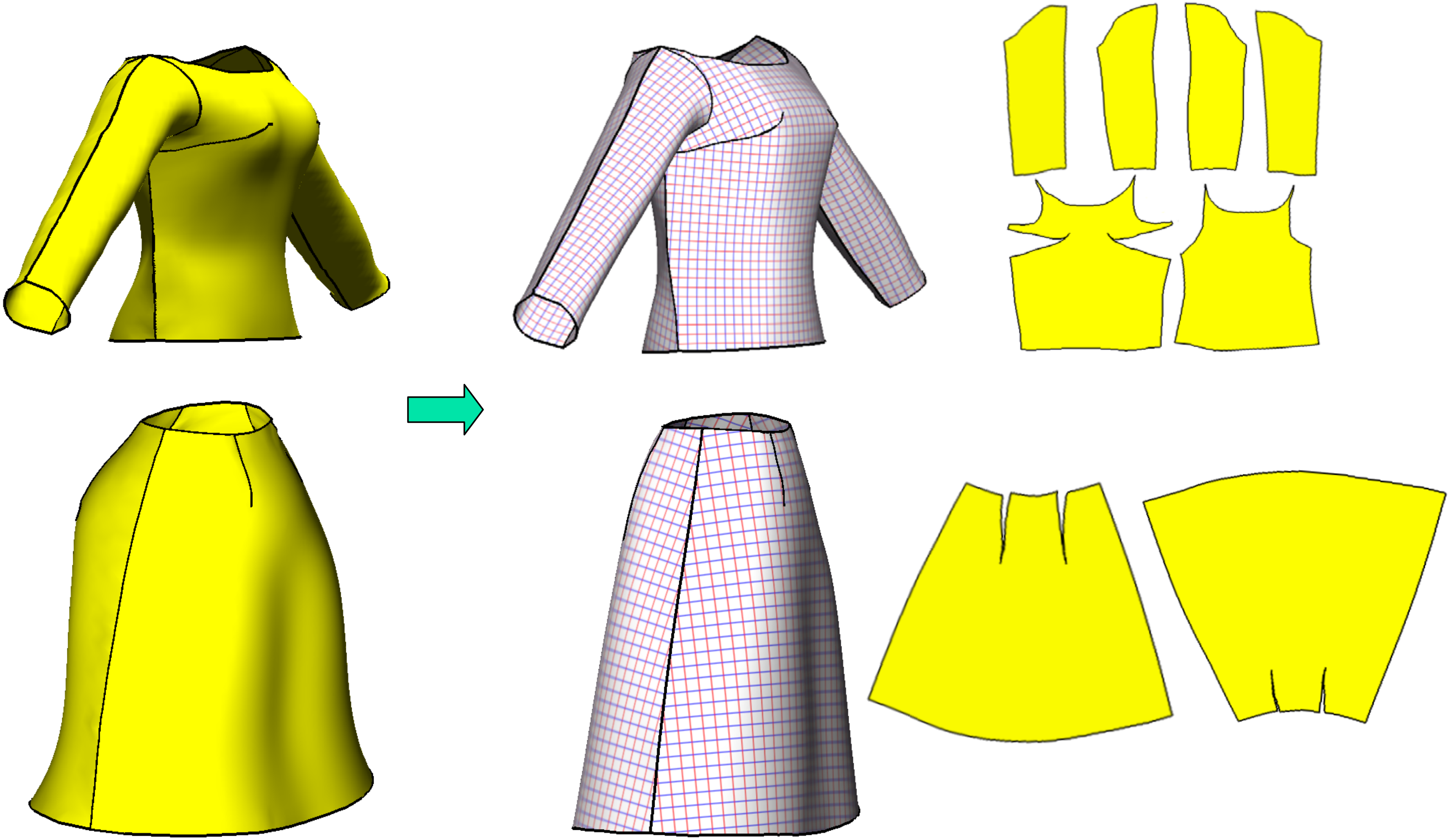


Input

1st iteration

2nd iteration

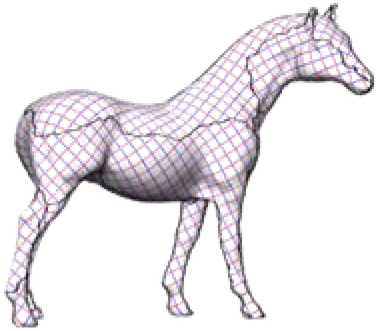
Results



Limitations

- Must start from “fairly” developable surface
- How much developable is developable enough?

L^2 stretch: 1.01



[Julius et al. 2005]

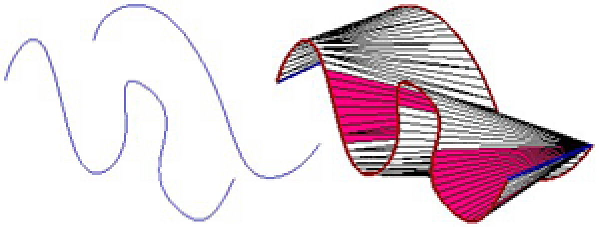
Direct Modeling

- Model developable surface “from scratch”
  - Based on user intent

Related Work

Torsal developables

(require ruling direction)



[Wang & Tang:05]

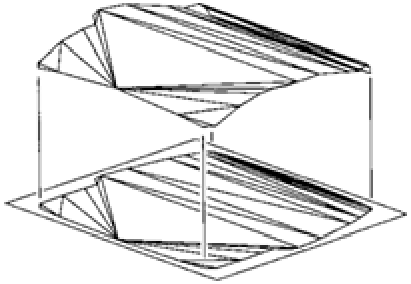
[Aumann:04]

[Chu & Sequin:02]

[Pottmann & Wallner:01]

[Wang:07]

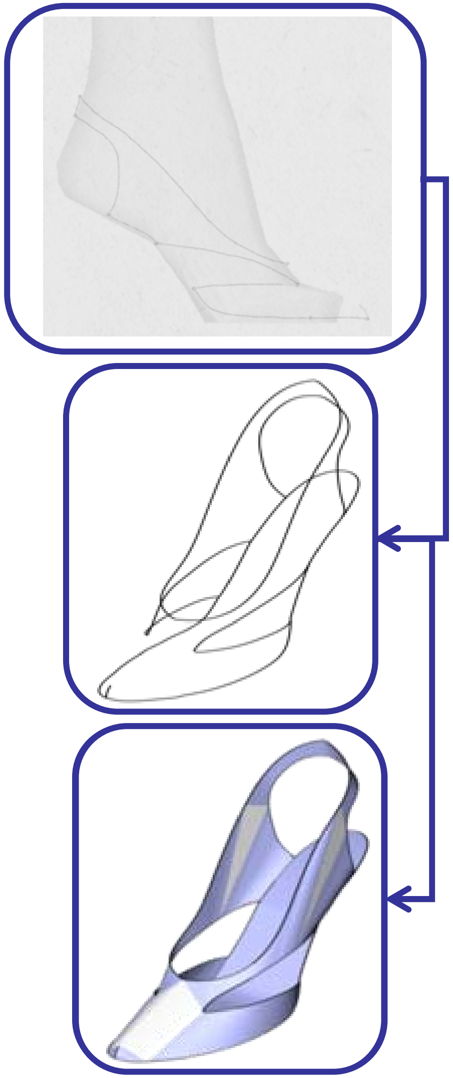
General height-field developables



[Frey:01]

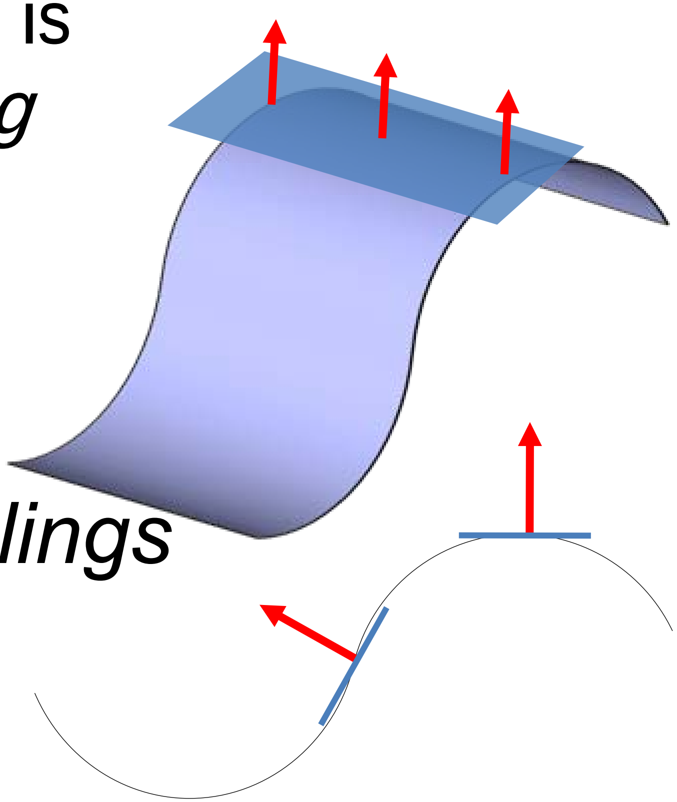
Modeling from Boundaries

- Algorithm to compute developable surfaces interpolating given boundaries
- Control surface characteristics
  - Allows search space navigation
- Combine with simple sketching interface
  - Creating boundaries
  - Modeling hints



Key Observations: Developability and Convexity

- On developable surfaces, tangent plane along ruling is almost always a *supporting* plane [LayL72]

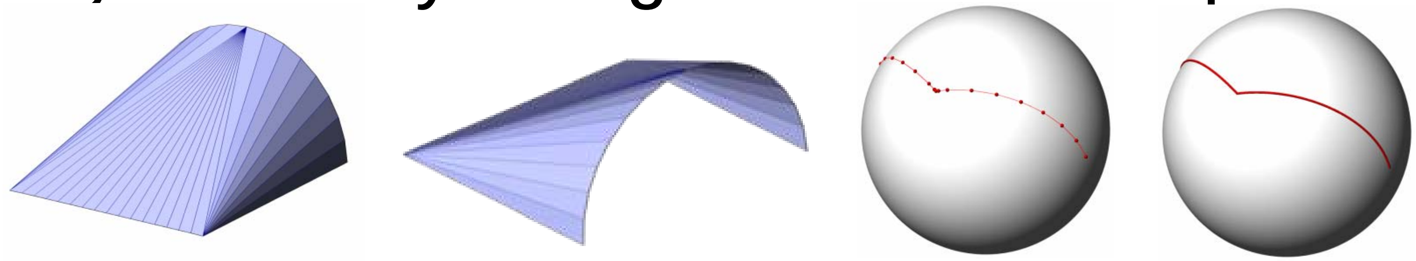


- *Developable* ⇔ *nearly all rulings lie* **on** *local convex hull*

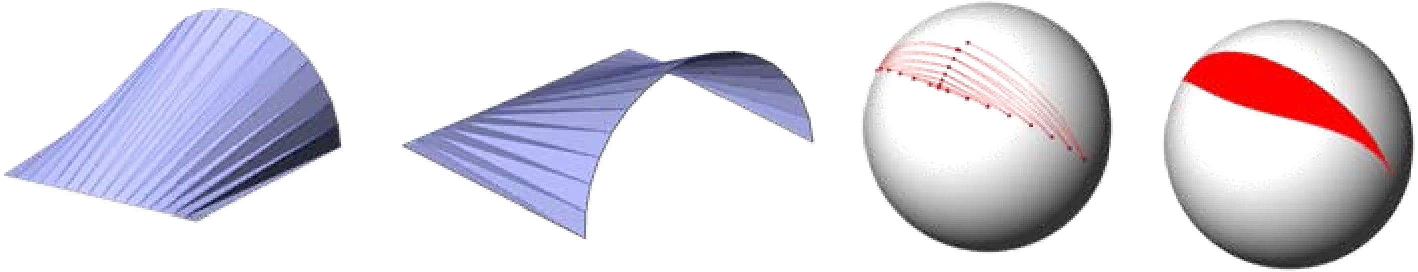
- Warped Ruled ⇔ nearly all rulings lie **inside** local convex hull

Developable Triangulations

- Every boundary triangulation is developable



Triangulation approximating smooth developable surface

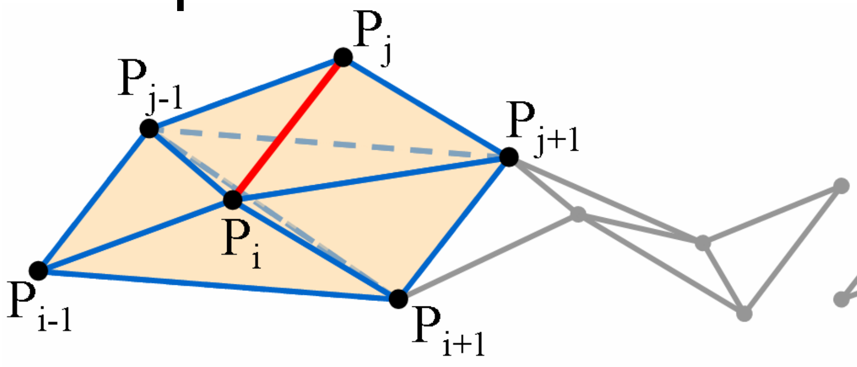


Triangulation approximating warped surface

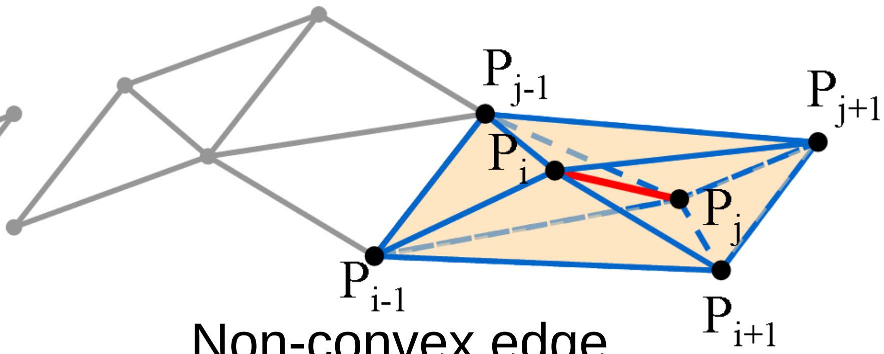
- Want boundary triangulations that approximate *smooth* developable surfaces

Developable Triangulations

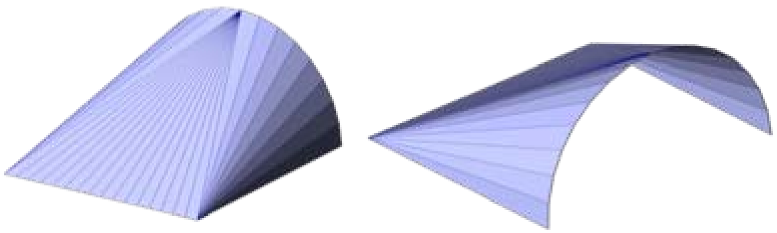
Approximates smooth developable surface ⟺ Nearly all internal edges are **locally convex**



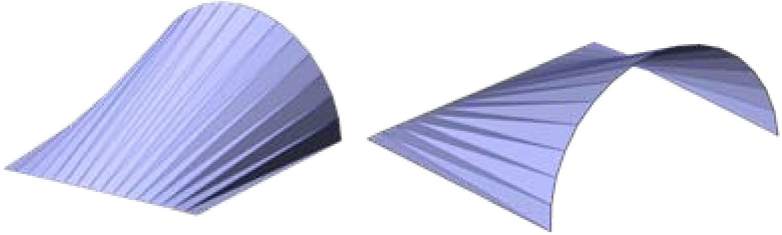
Locally convex edge



Non-convex edge



Developable Triangulation

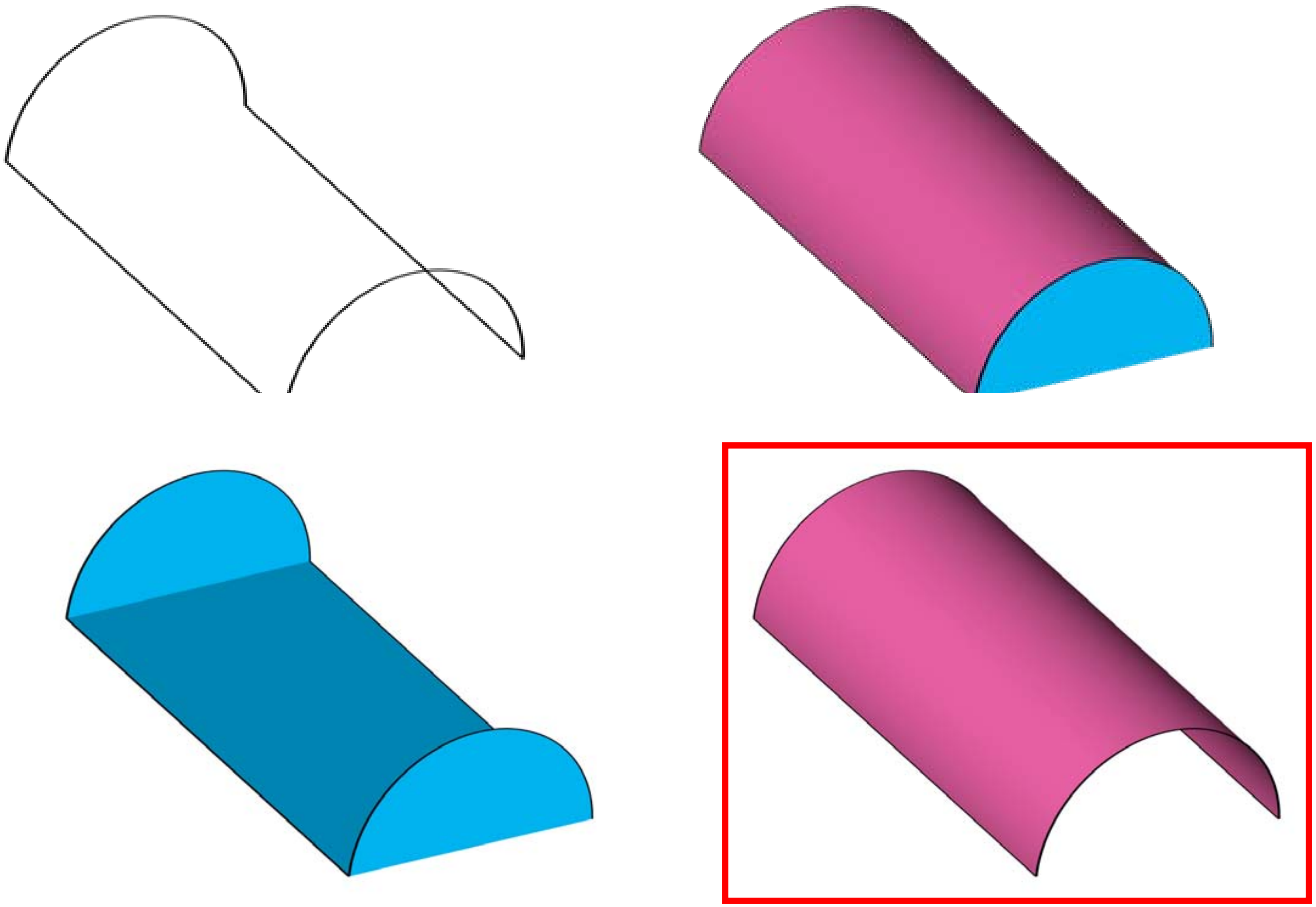


Warped Triangulation

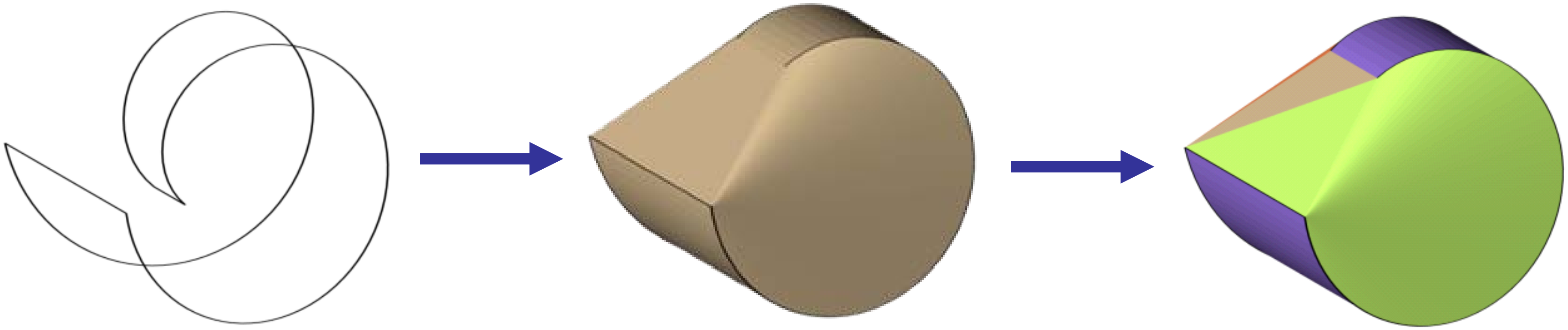
Convex Hulls & Developable Triangulations

- On polyline CH, all edges are locally convex
- Smooth ruled surface contained by its boundary's CH
- CH of smooth curve consists of torsal developable surfaces and planar regions [Sedykh:86]
- Curve that lies on CH separates it into two developable envelopes [Sedykh:86]
- **Key Idea** - Look for interpolating developable regions on boundary CH

Base Case

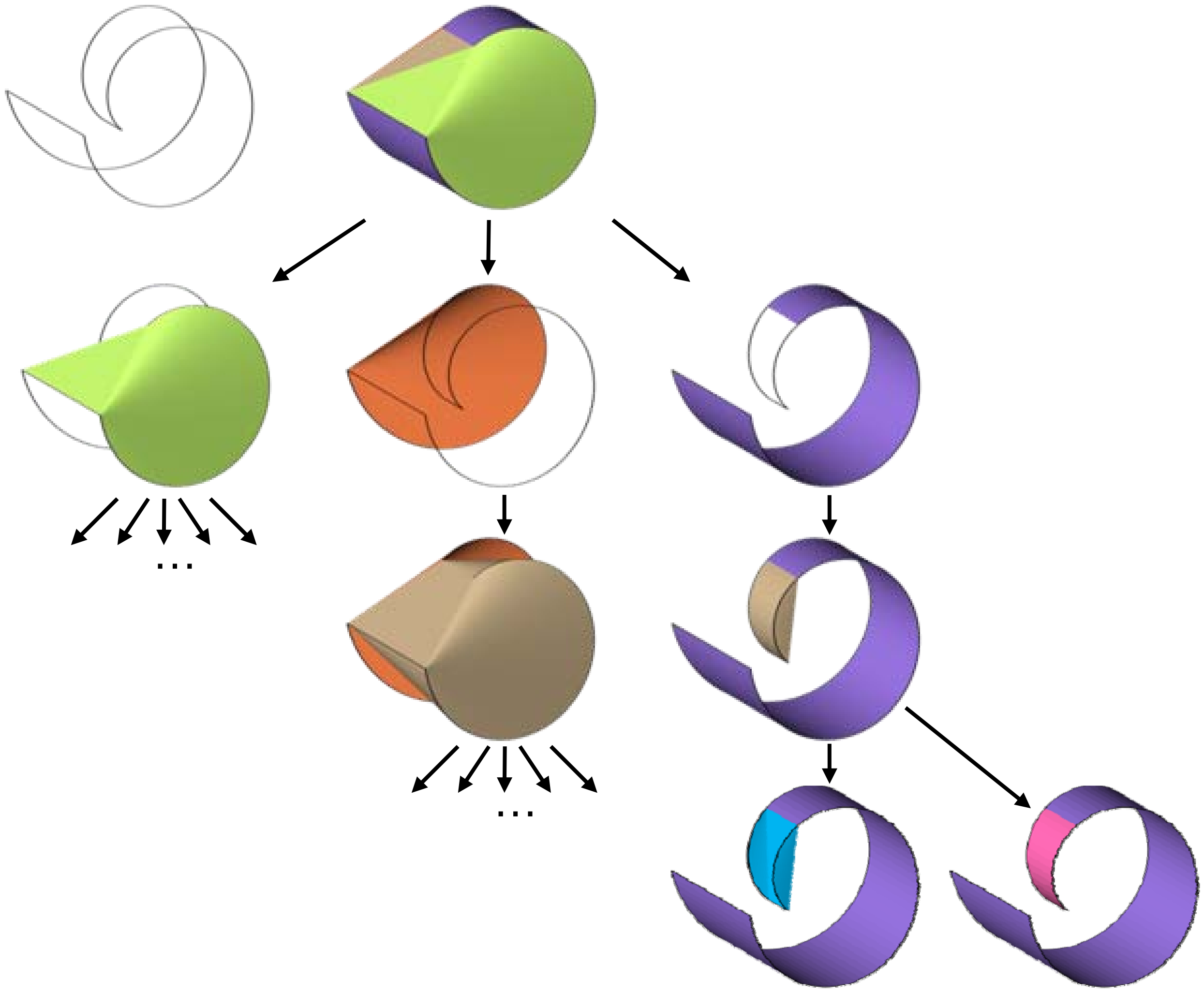


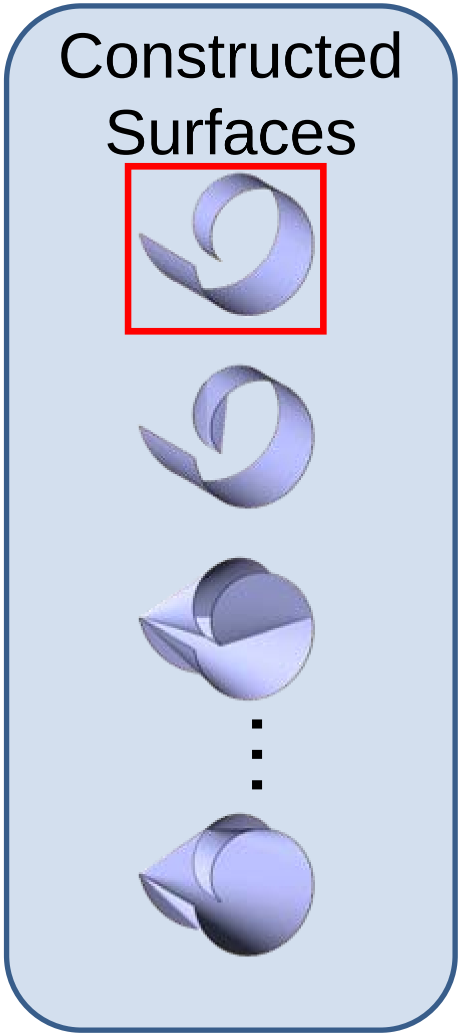
Recursion



- Identify charts that correspond to torsal surfaces
    - Sequences of triangles with one or more boundary edges

Recursion




Constructed Surfaces

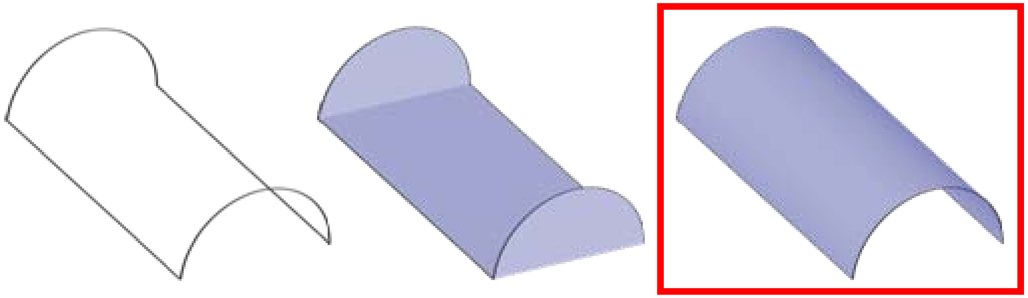
Desirable Triangulation Properties

Developable

Nearly all edges locally convex - “a given”

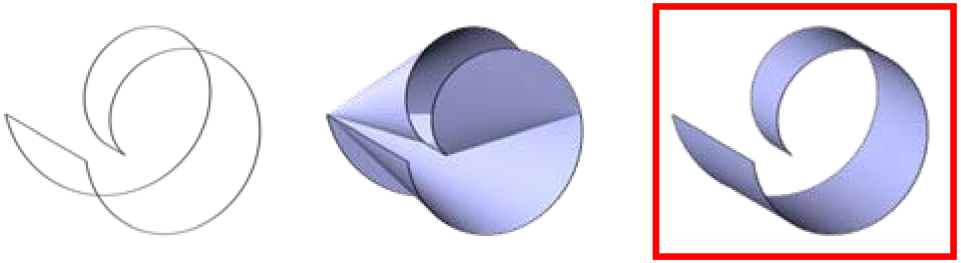
Smooth

No sharp angles



Predictable

Less internal triangles



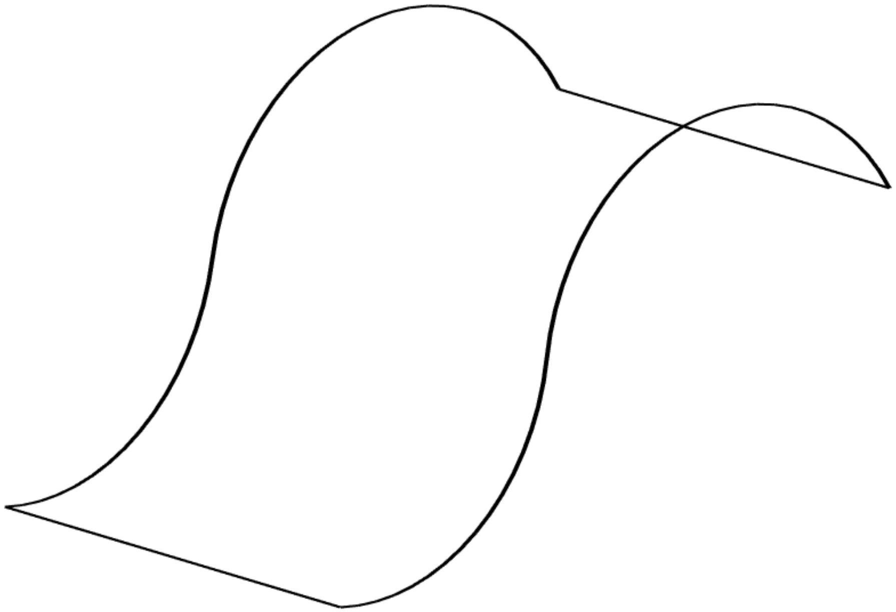
Fair

Lower sum of squared dihedral angles

Branch and Bound Algorithm

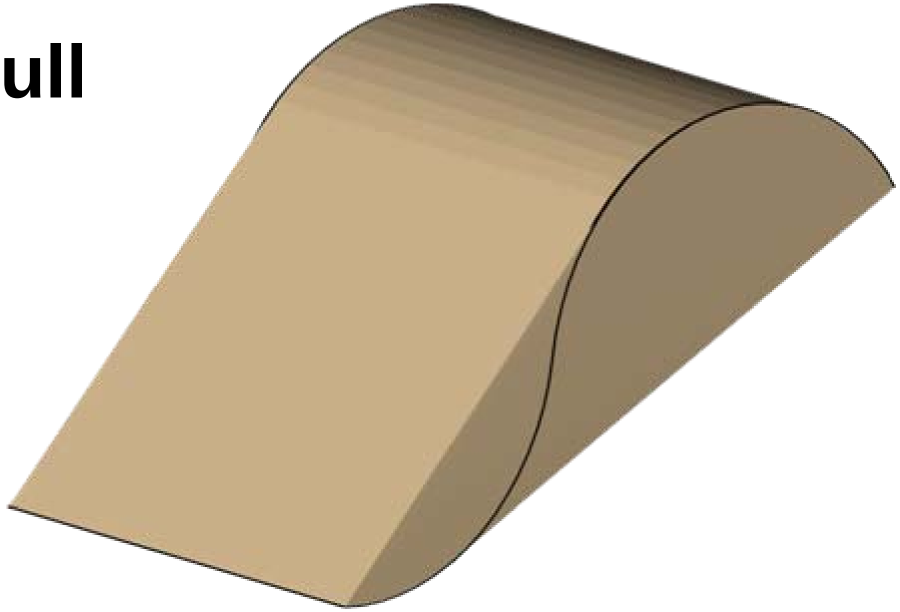
1. **Expand** Search Space
   - Always segment CH into charts
   - Discard charts with sharp dihedral angles
2. **Navigate** Search Space
   - Compute triangulations which are as predictable and fair as possible
   - Use branch and bound:
     - Store partial solutions (covers)
     - At each step, process most promising cover
     - Discard not-promising or redundant covers

Branch and Bound Algorithm



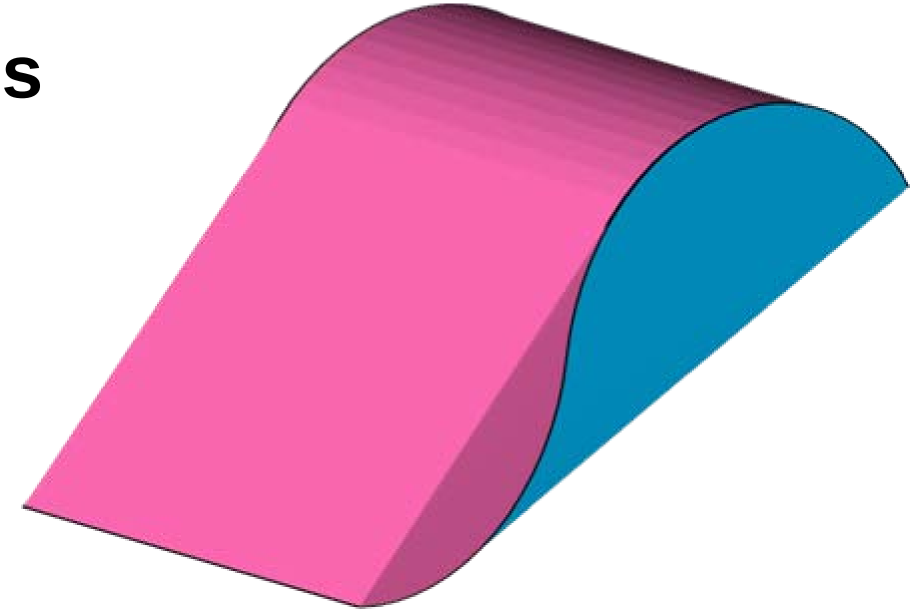
Branch and Bound Algorithm

Convex Hull



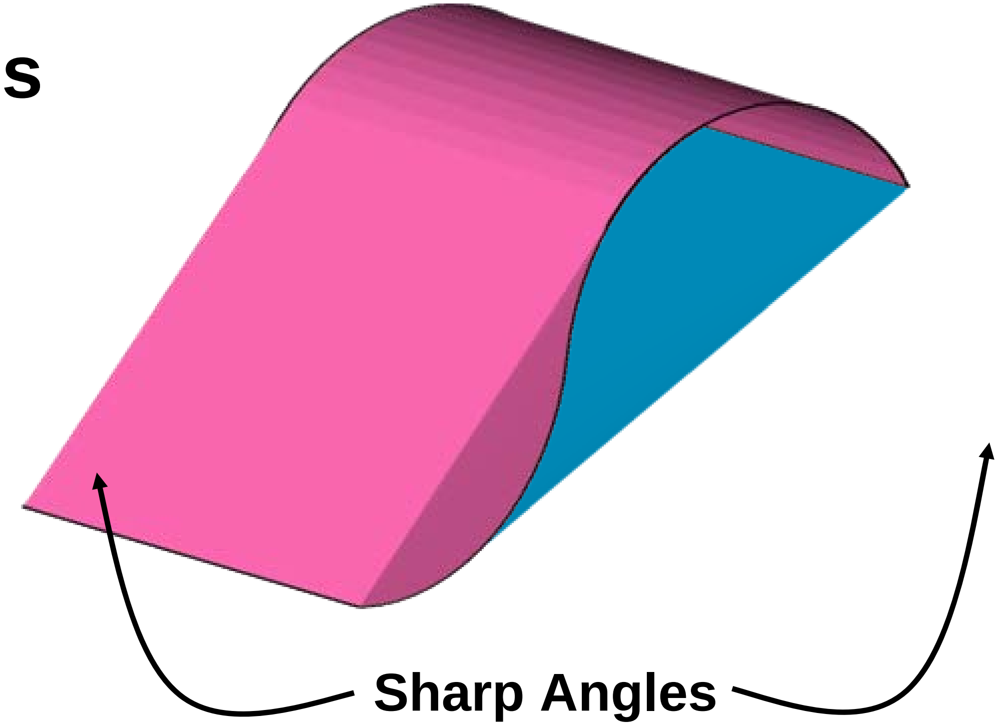
Branch and Bound Algorithm

Envelopes



Branch and Bound Algorithm

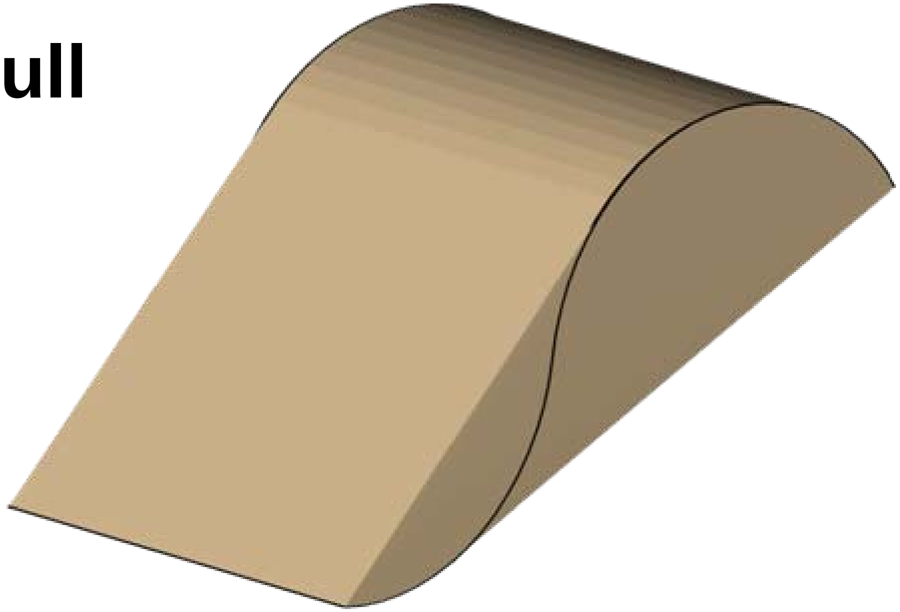
Envelopes



Sharp Angles

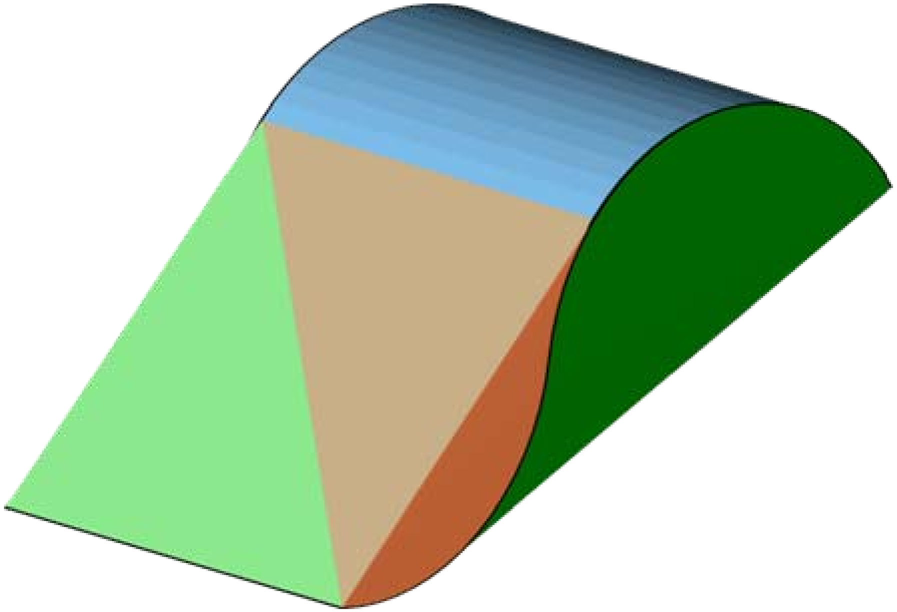
Branch and Bound Algorithm

Convex Hull



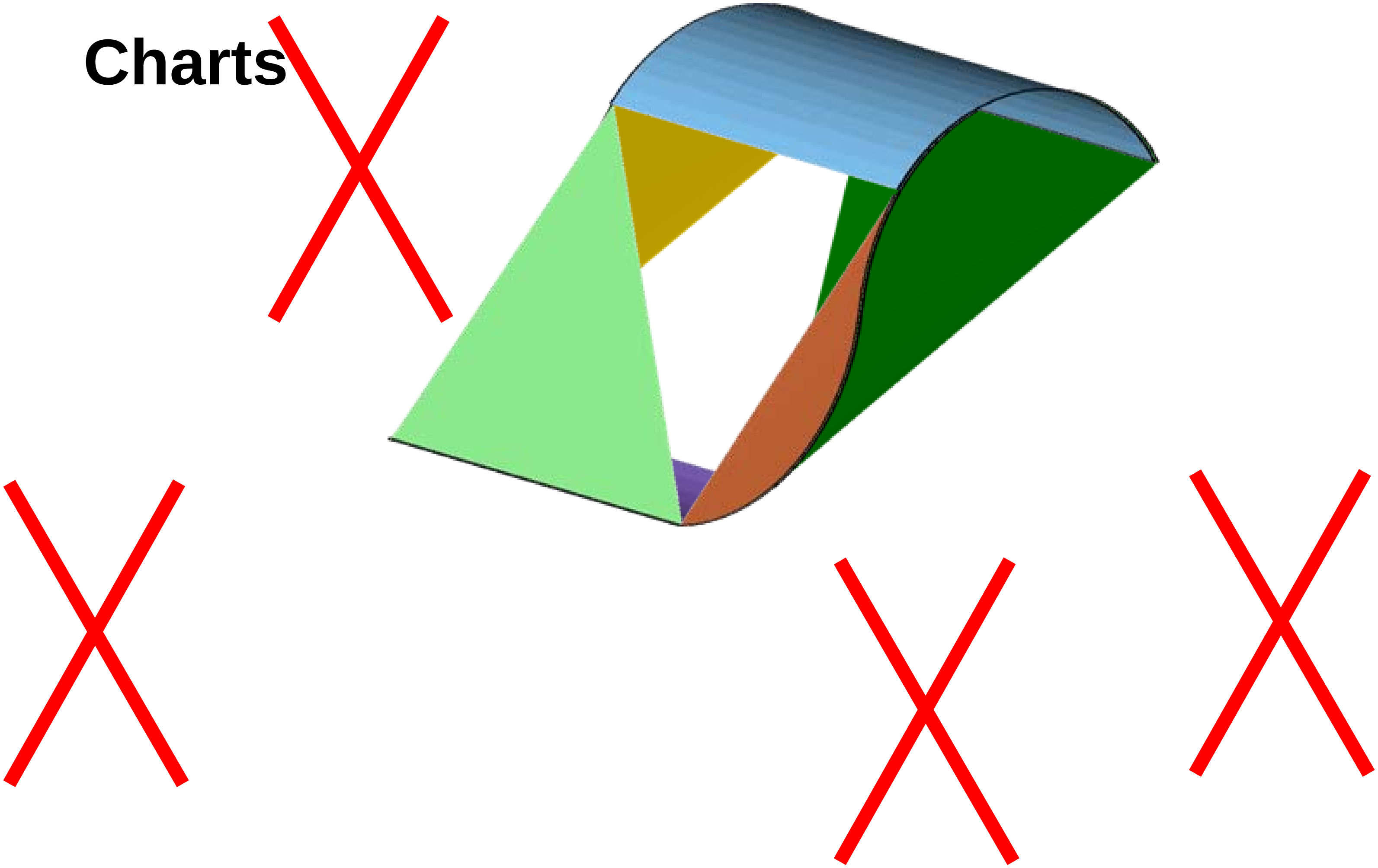
Branch and Bound Algorithm

Charts

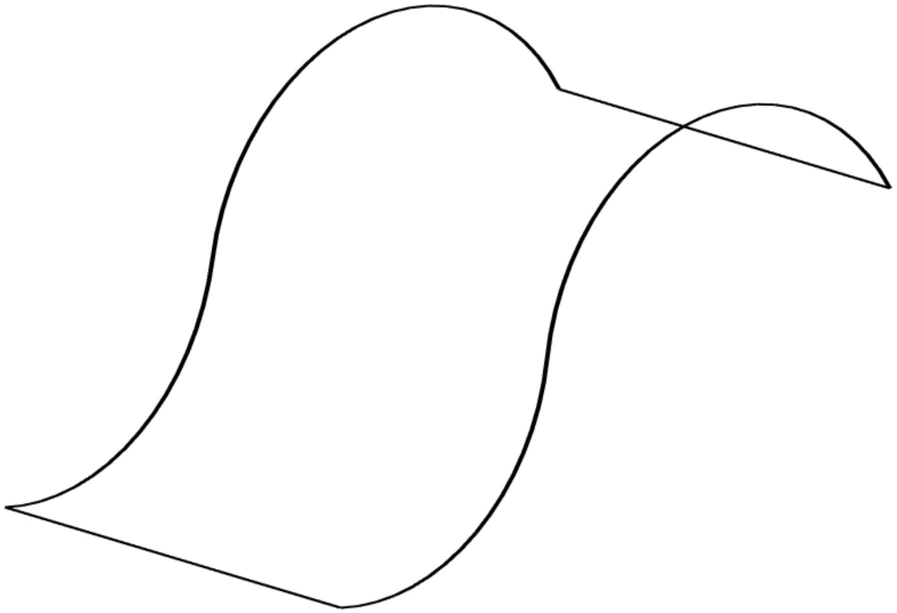


Branch and Bound Algorithm

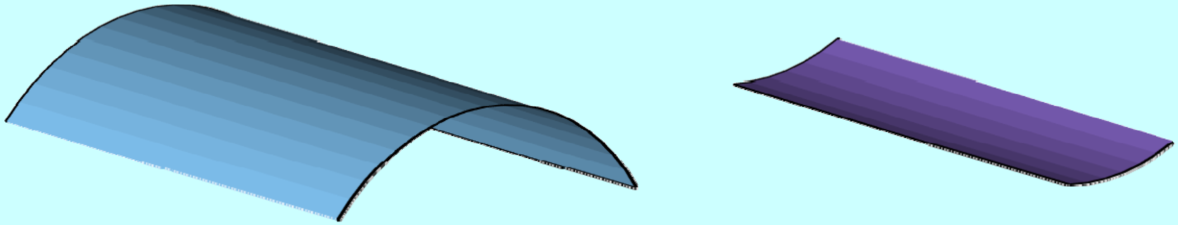
Charts



Branch and Bound Algorithm

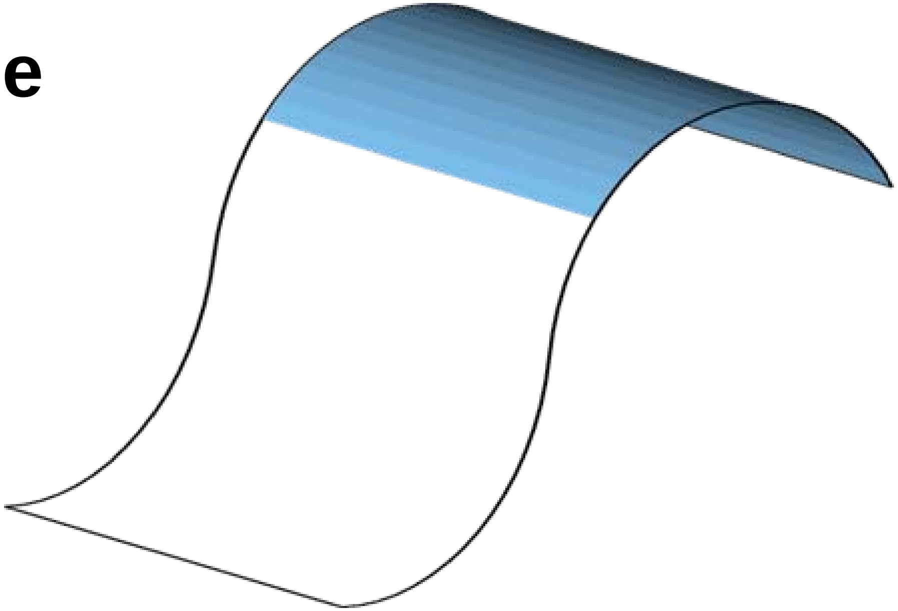


Priority Queue

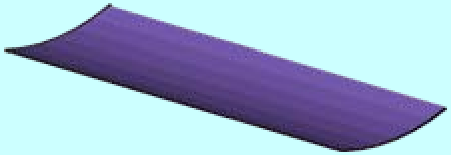


Branch and Bound Algorithm

Pop Queue

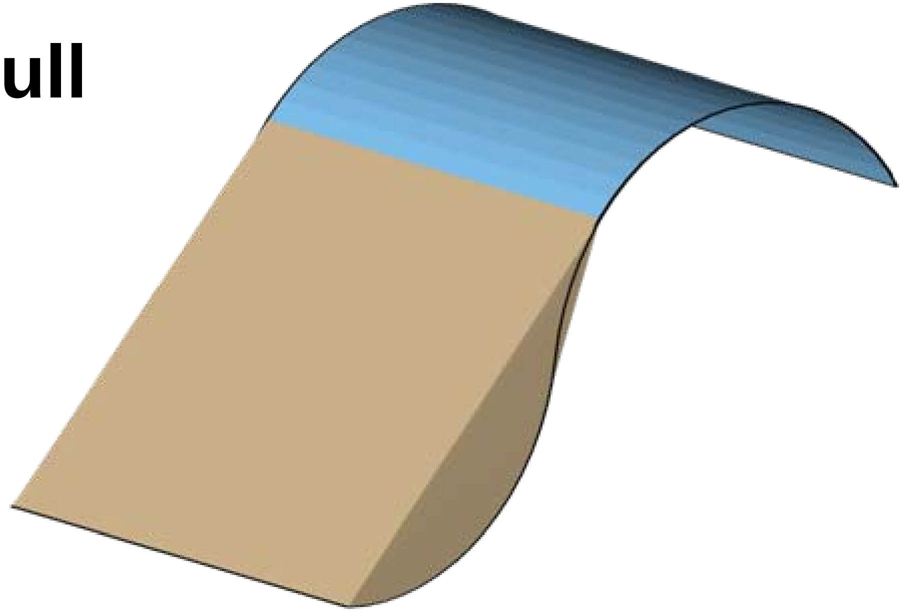


Priority Queue

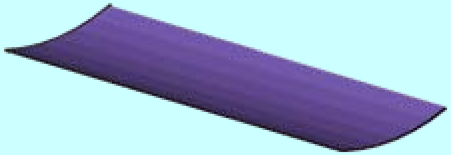


Branch and Bound Algorithm

Convex Hull

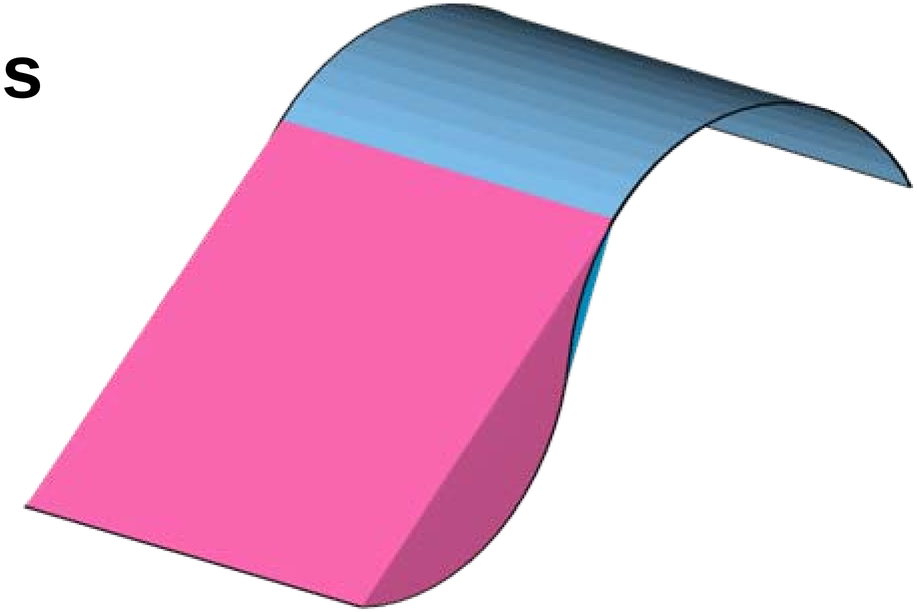


Priority Queue

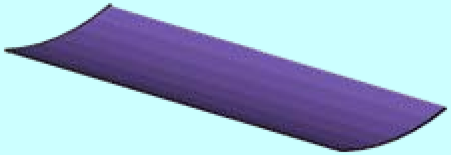


Branch and Bound Algorithm

Envelopes

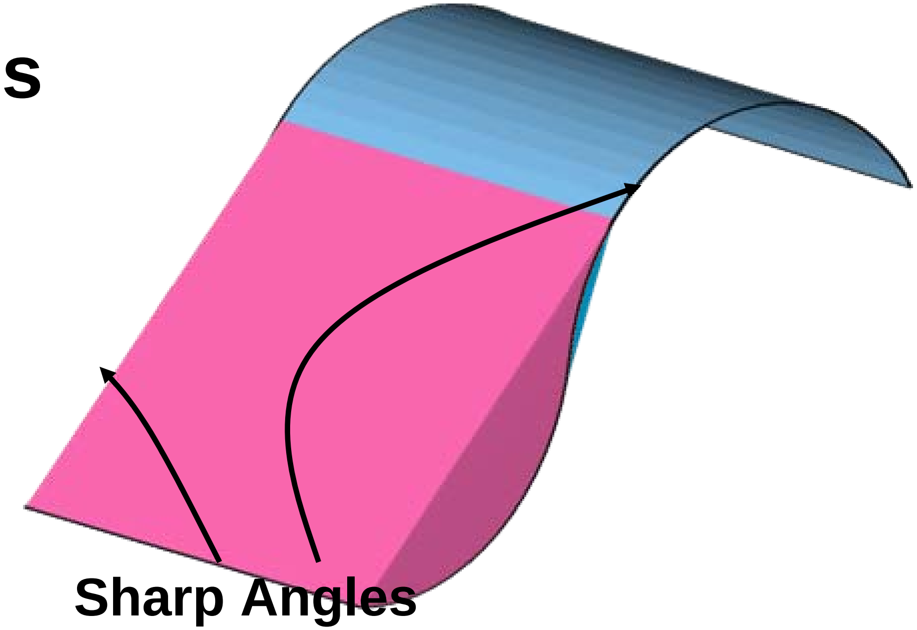


Priority Queue

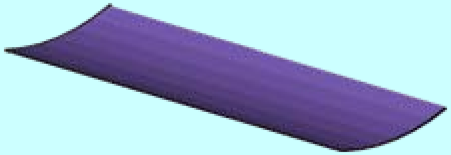


Branch and Bound Algorithm

Envelopes

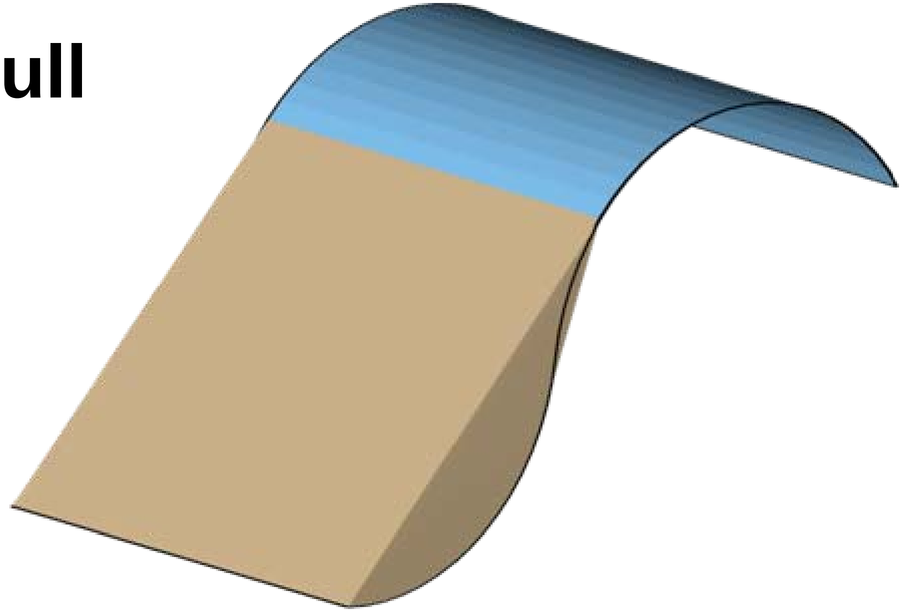


Priority Queue

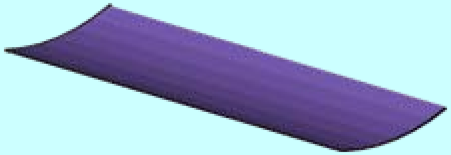


Branch and Bound Algorithm

Convex Hull

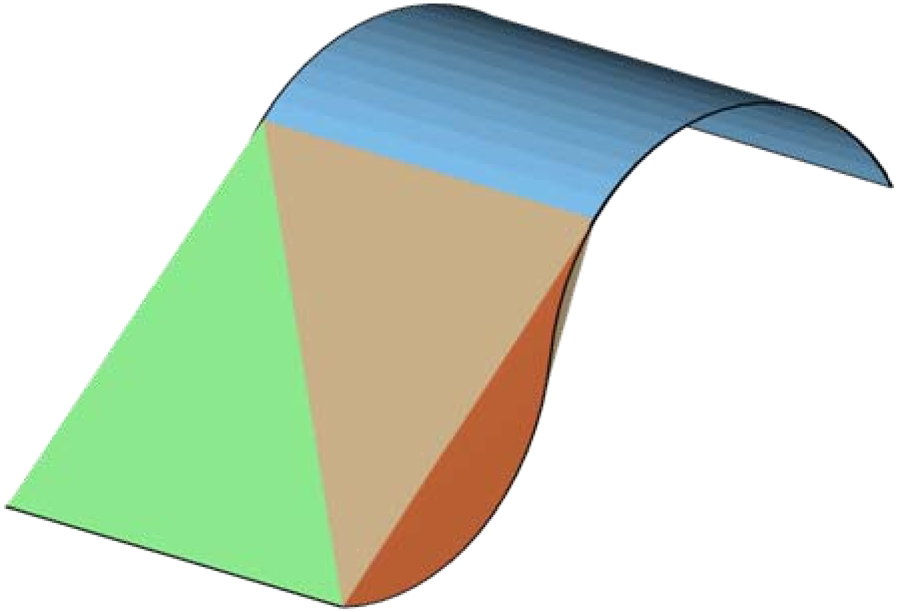


Priority Queue

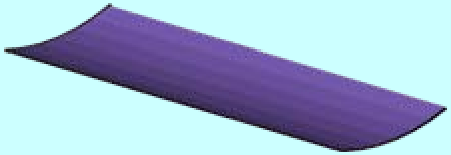


Branch and Bound Algorithm

Charts

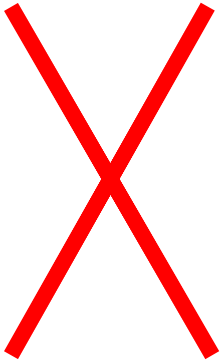


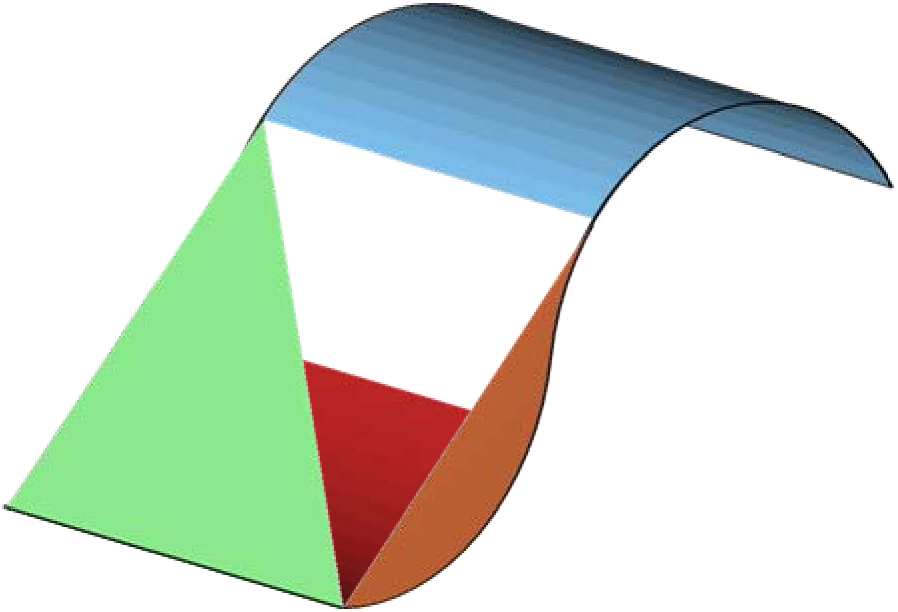
Priority Queue

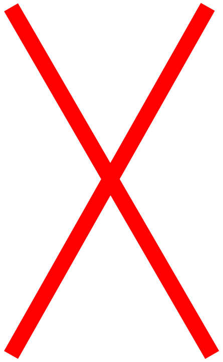


Branch and Bound Algorithm

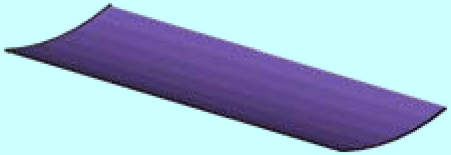
Charts





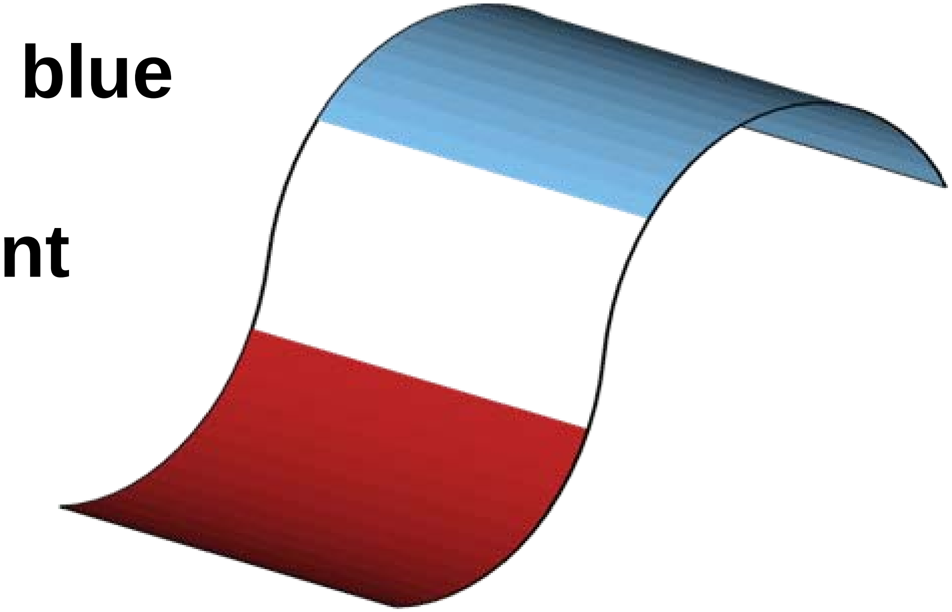


Priority Queue

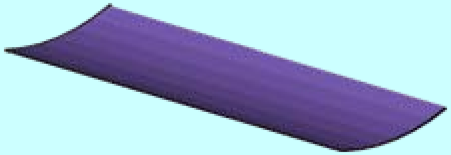


Branch and Bound Algorithm

Red and blue cover is redundant

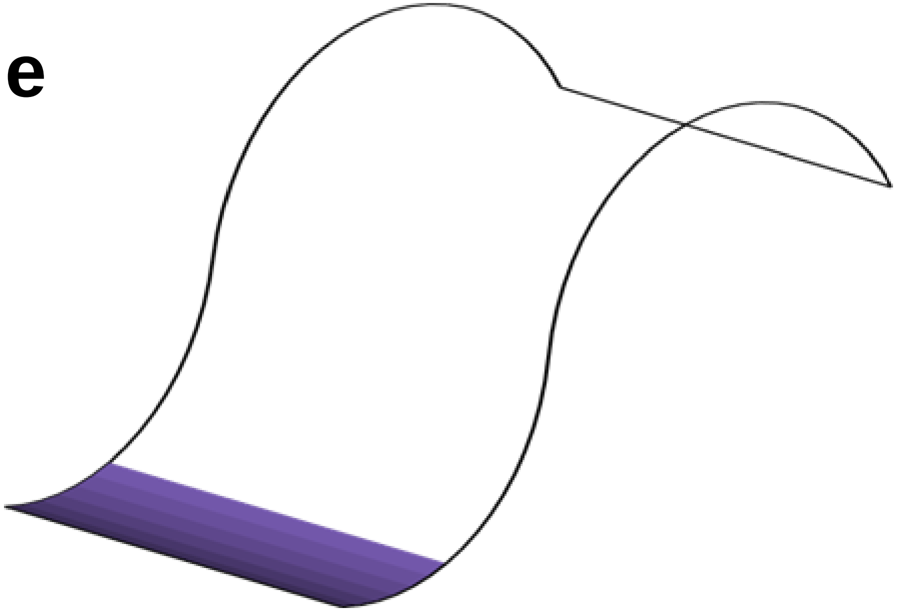


Priority Queue



Branch and Bound Algorithm

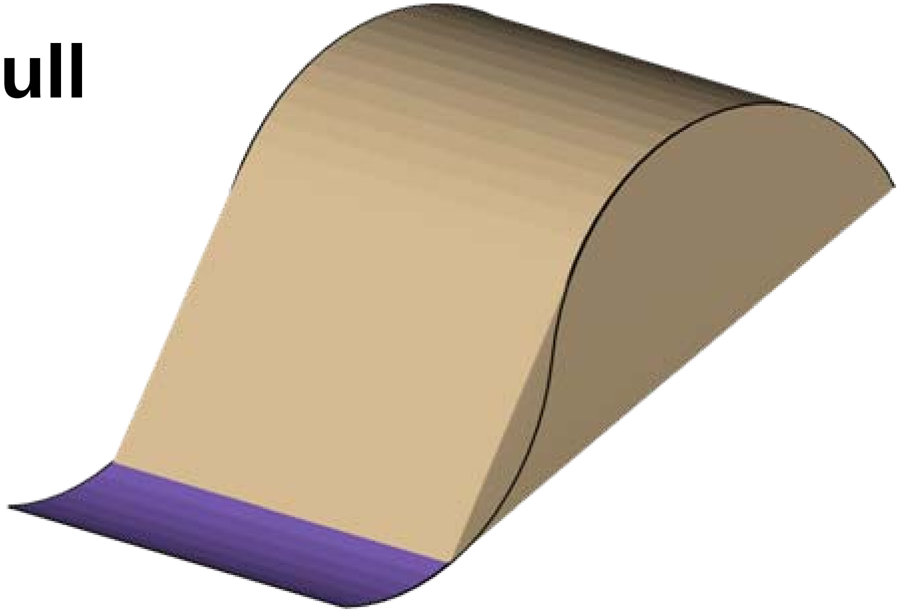
Pop Queue



Priority Queue

Branch and Bound Algorithm

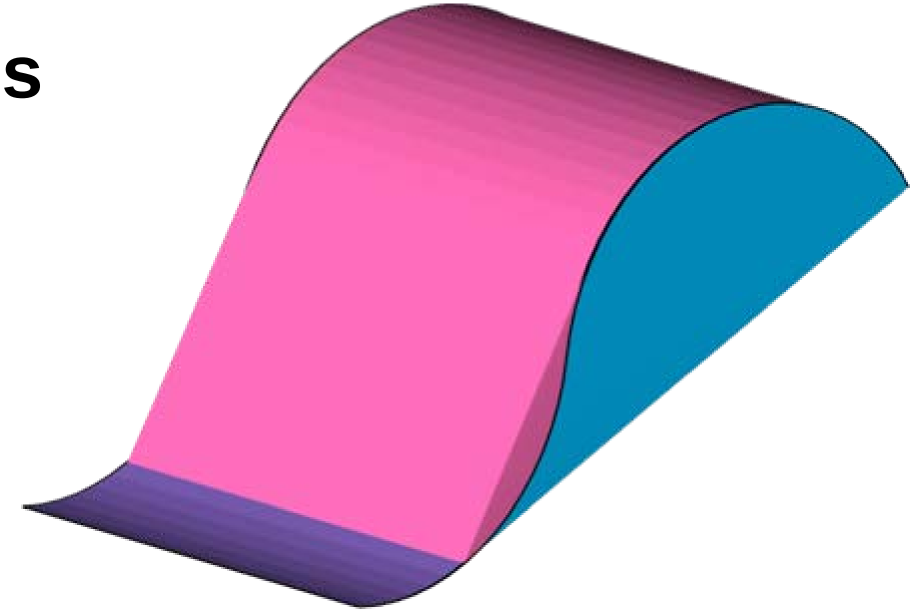
Convex Hull



Priority
Queue

Branch and Bound Algorithm

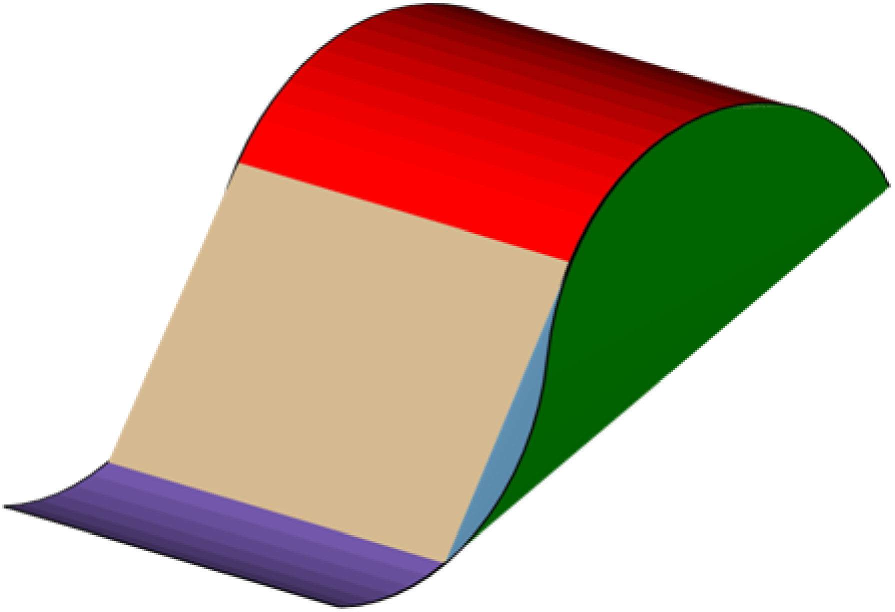
Envelopes



Priority Queue

Branch and Bound Algorithm

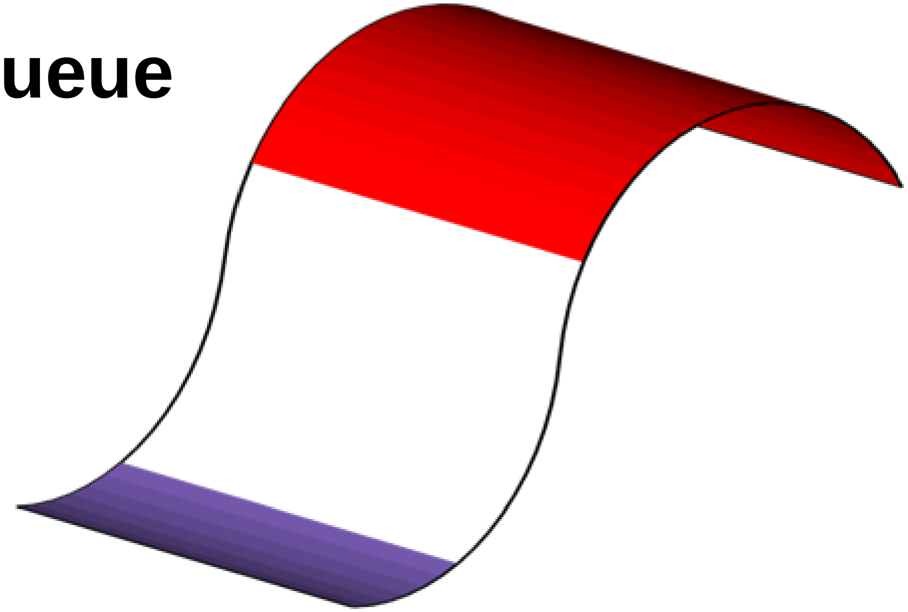
Charts



Priority Queue

Branch and Bound Algorithm

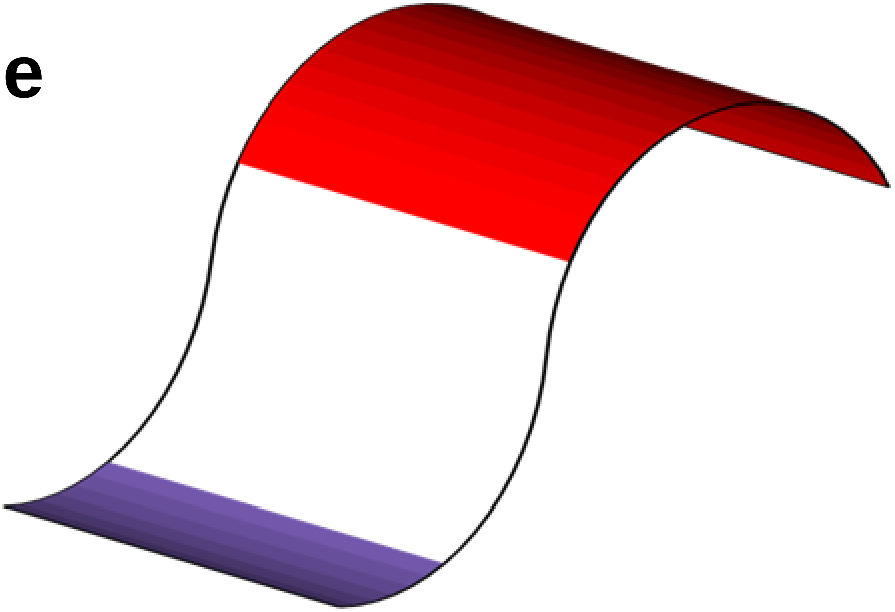
Add To Queue



Priority Queue

Branch and Bound Algorithm

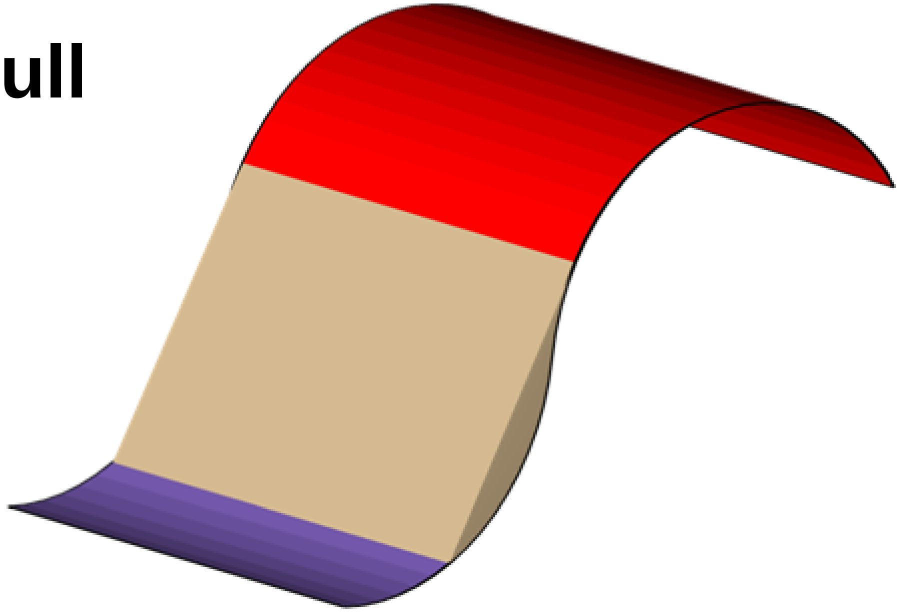
Pop Queue



Priority Queue

Branch and Bound Algorithm

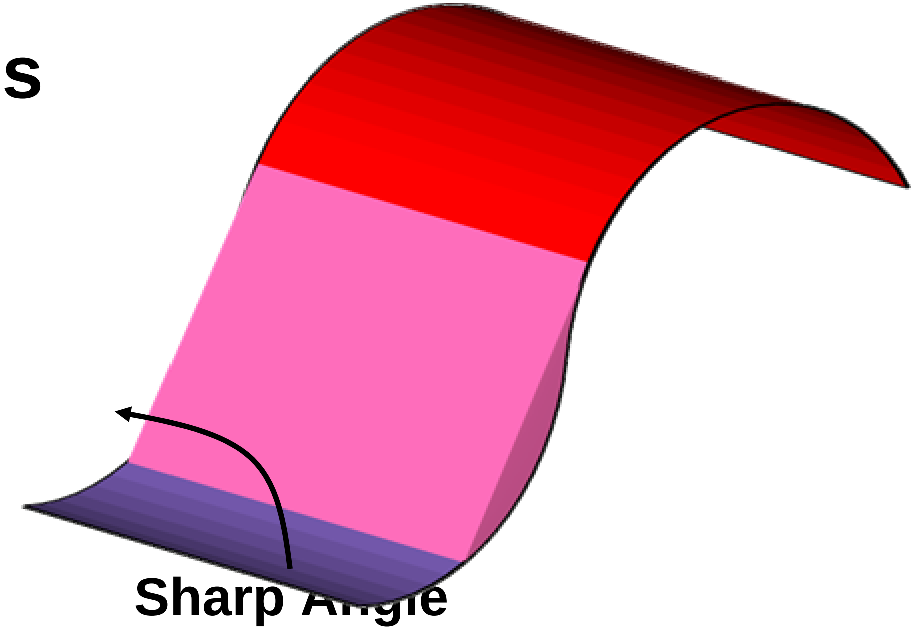
Convex Hull



Priority Queue

Branch and Bound Algorithm

Envelopes

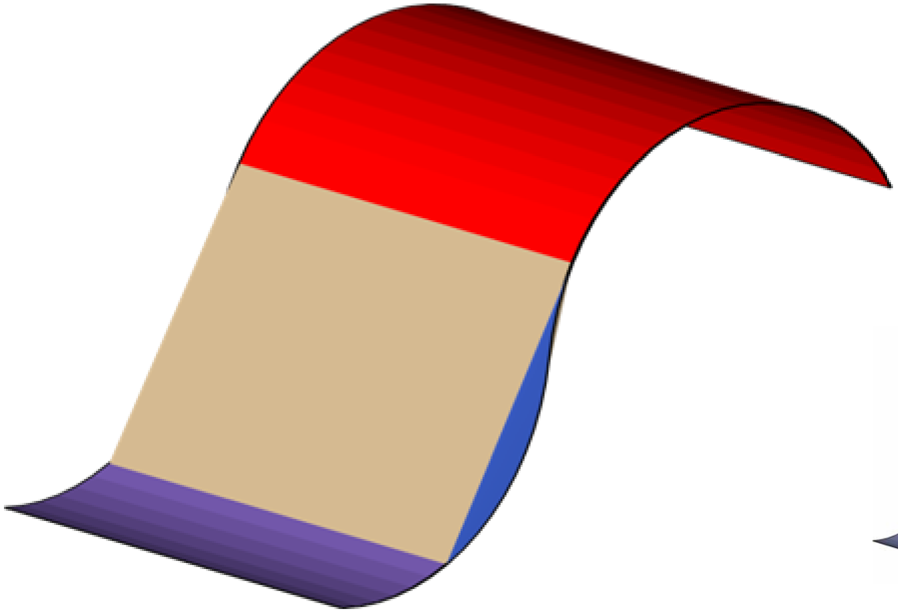


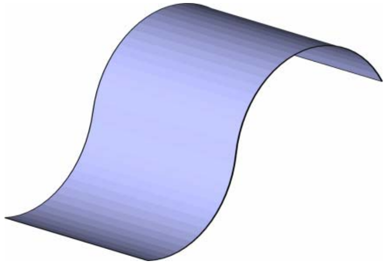
Sharp Angle

Priority Queue

Branch and Bound Algorithm

Charts



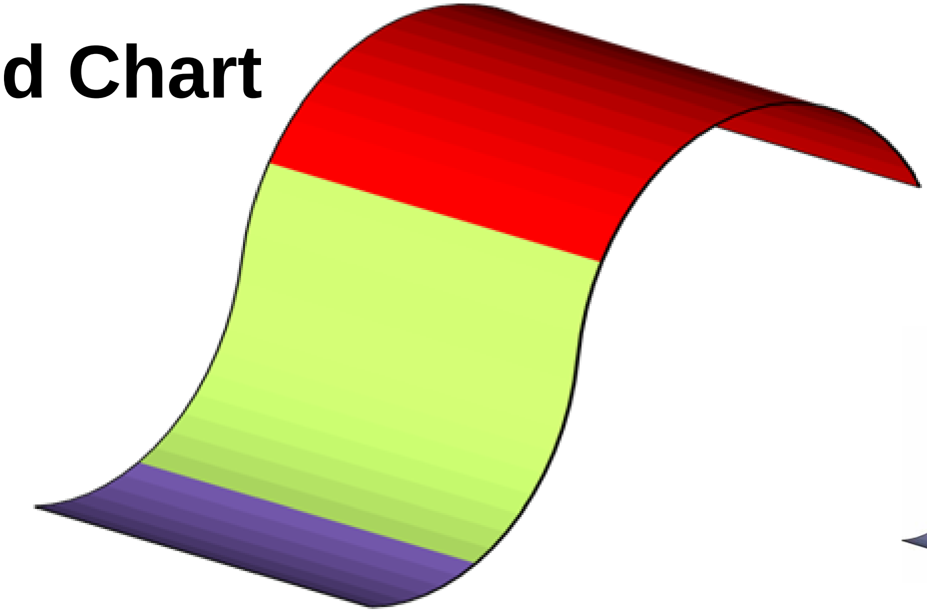


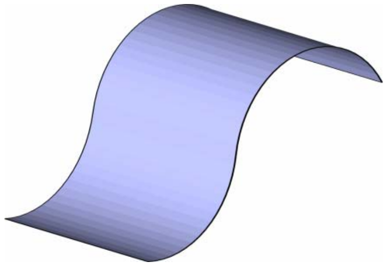
Best Triangulation

Priority Queue

Branch and Bound Algorithm

Only Valid Chart



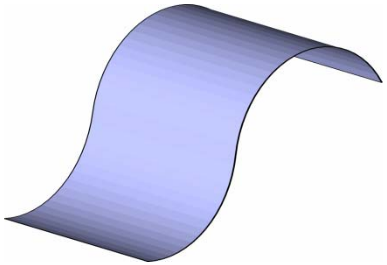


Best Triangulation

Priority
Queue

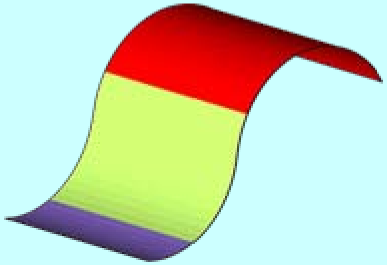
Branch and Bound Algorithm

Add To Queue



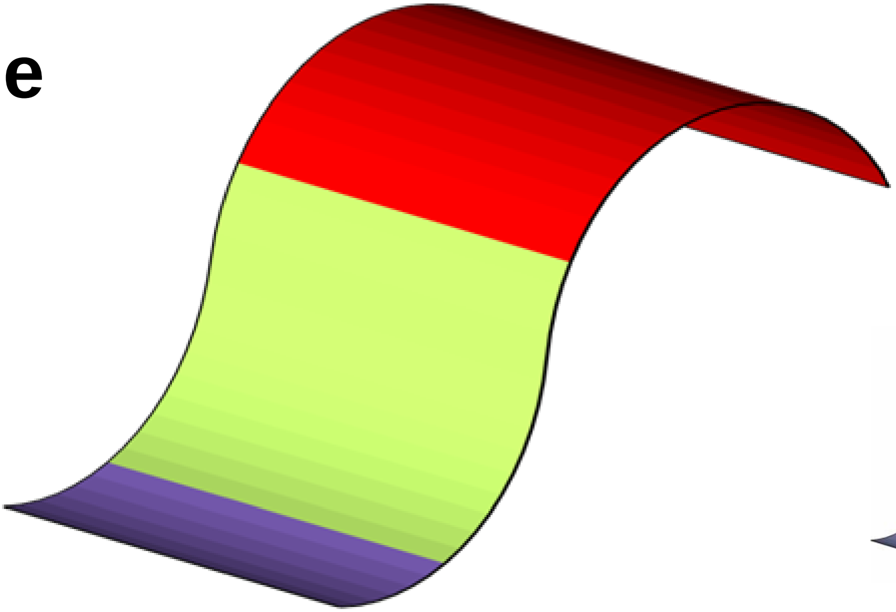
Best Triangulation

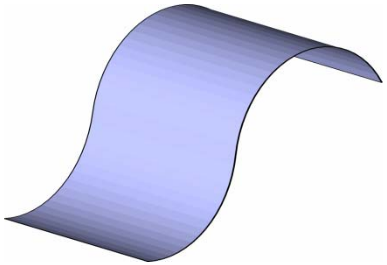
Priority Queue



Branch and Bound Algorithm

Pop Queue



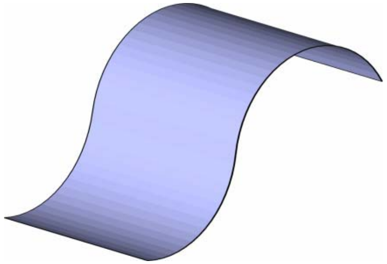


Best Triangulation

Priority
Queue

Branch and Bound Algorithm

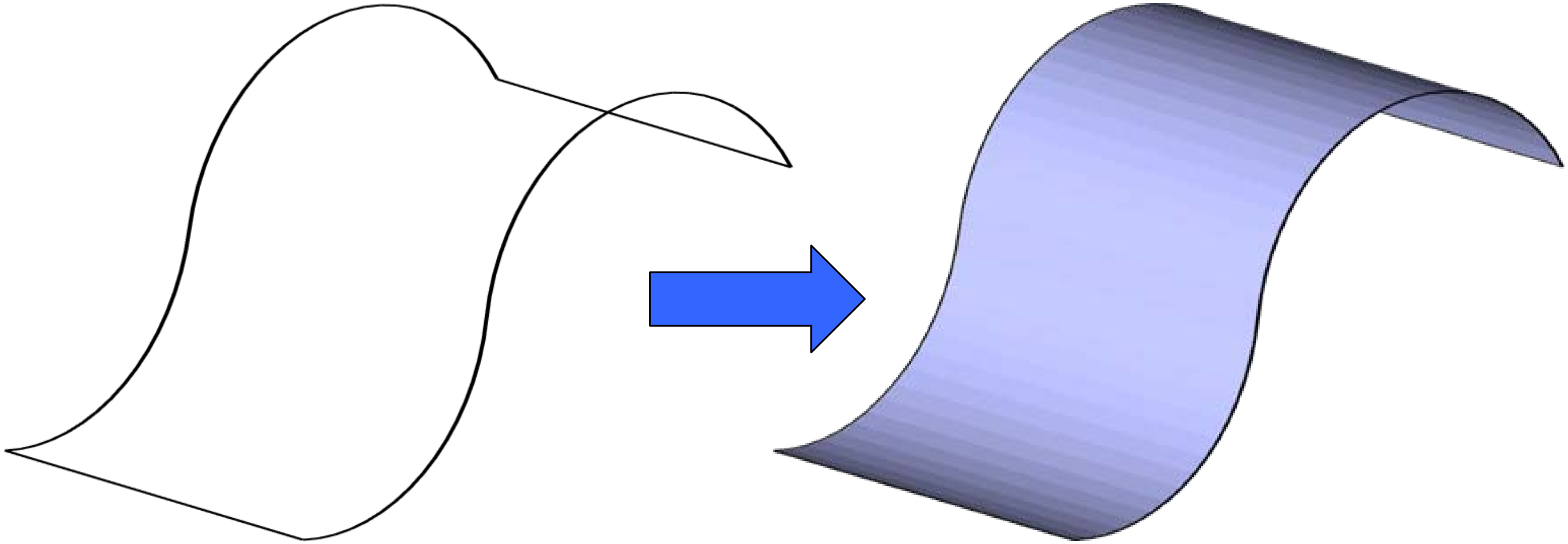
Termination



Best Triangulation

Priority
Queue

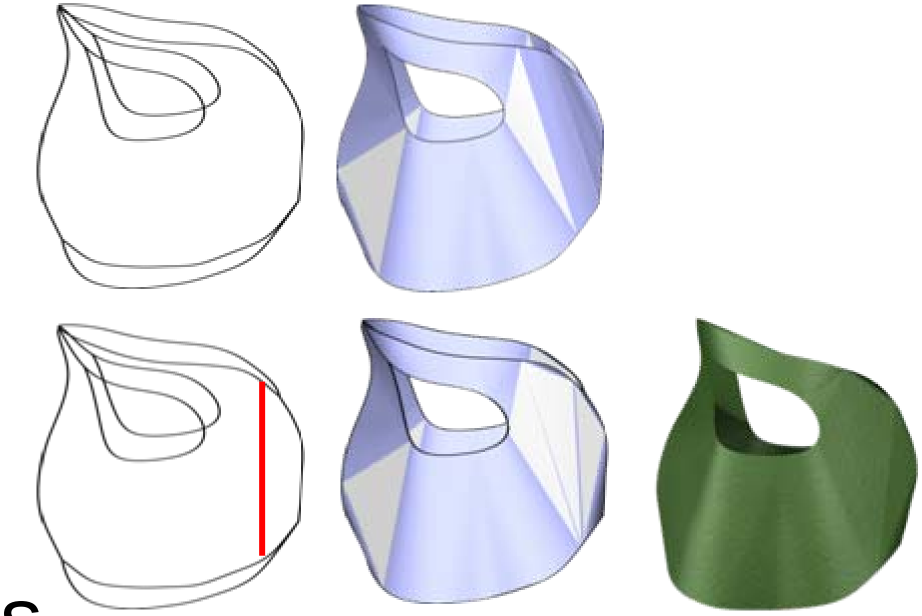
Branch and Bound Algorithm



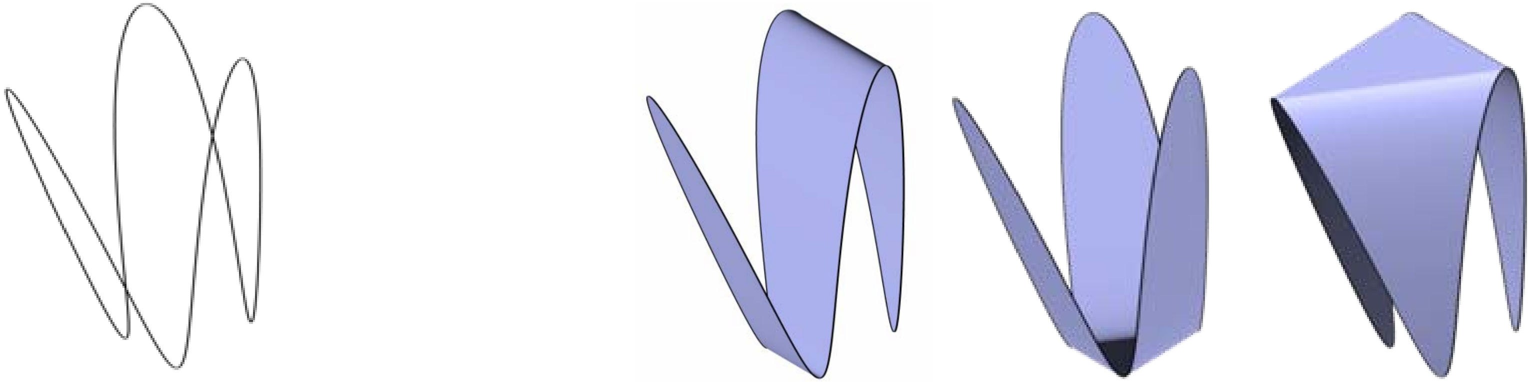
Priority Queue

Additional User Control

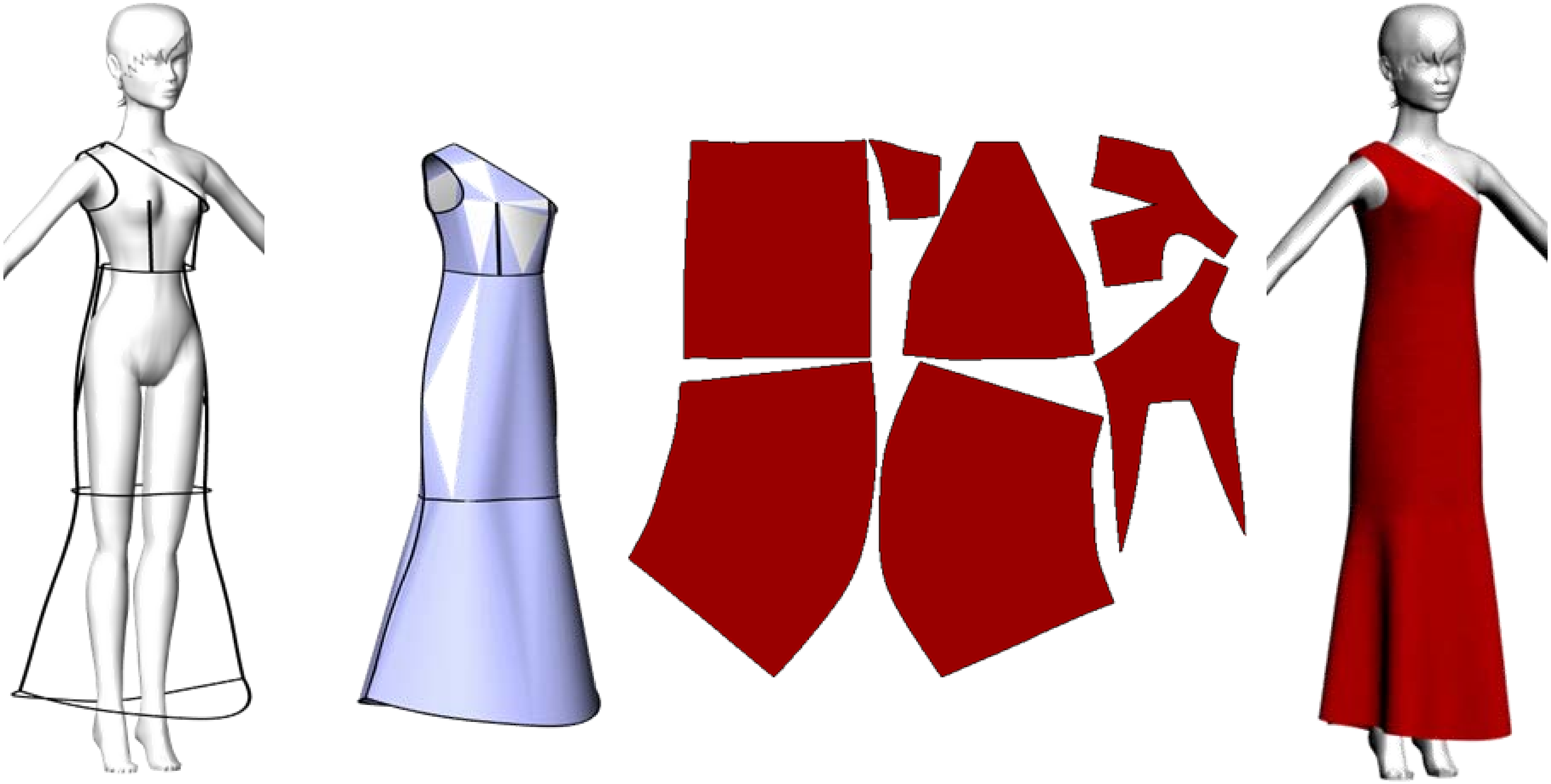
- Specify rulings



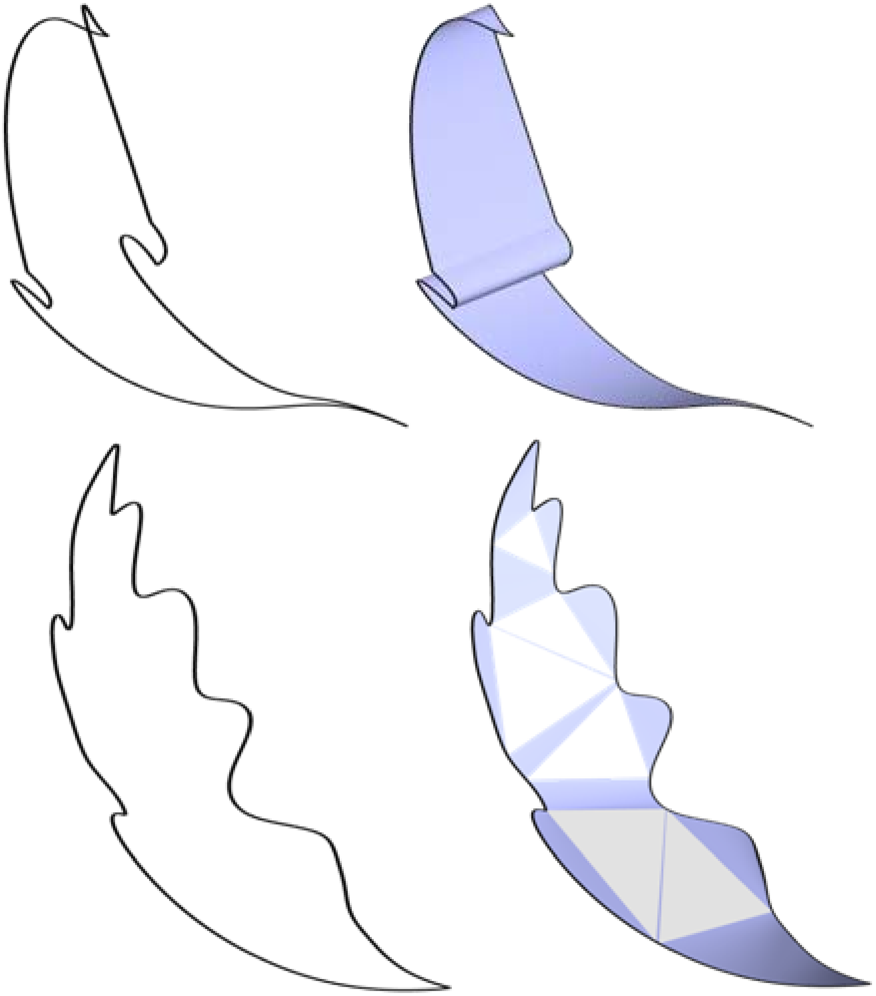
- Select other solutions

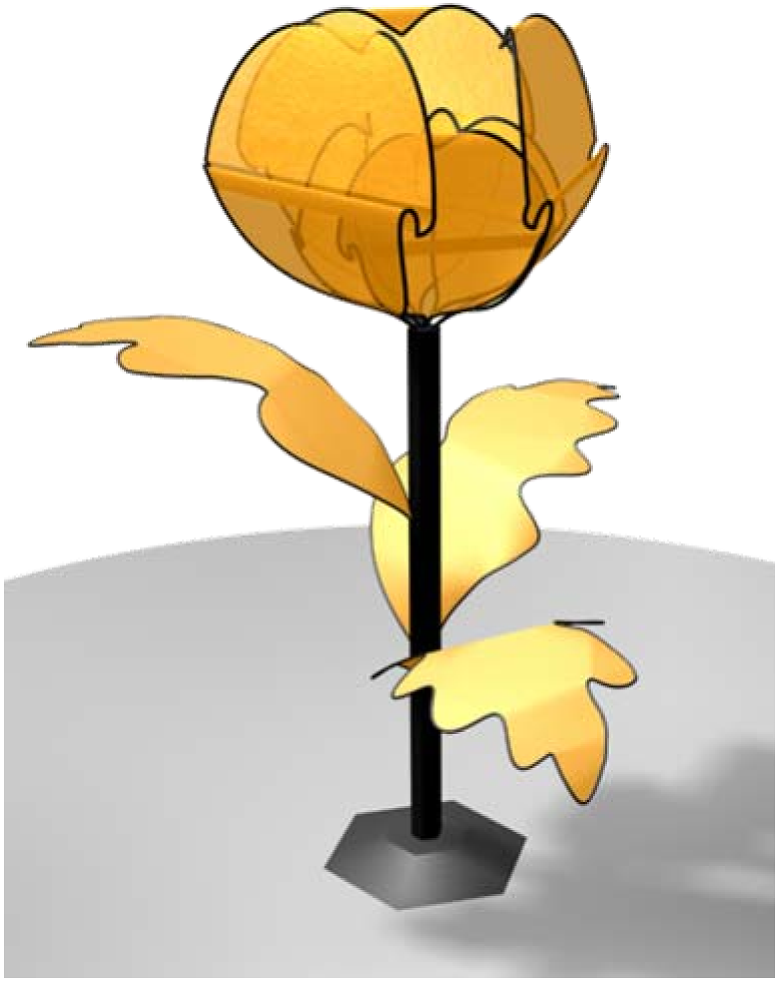


Results - Dress

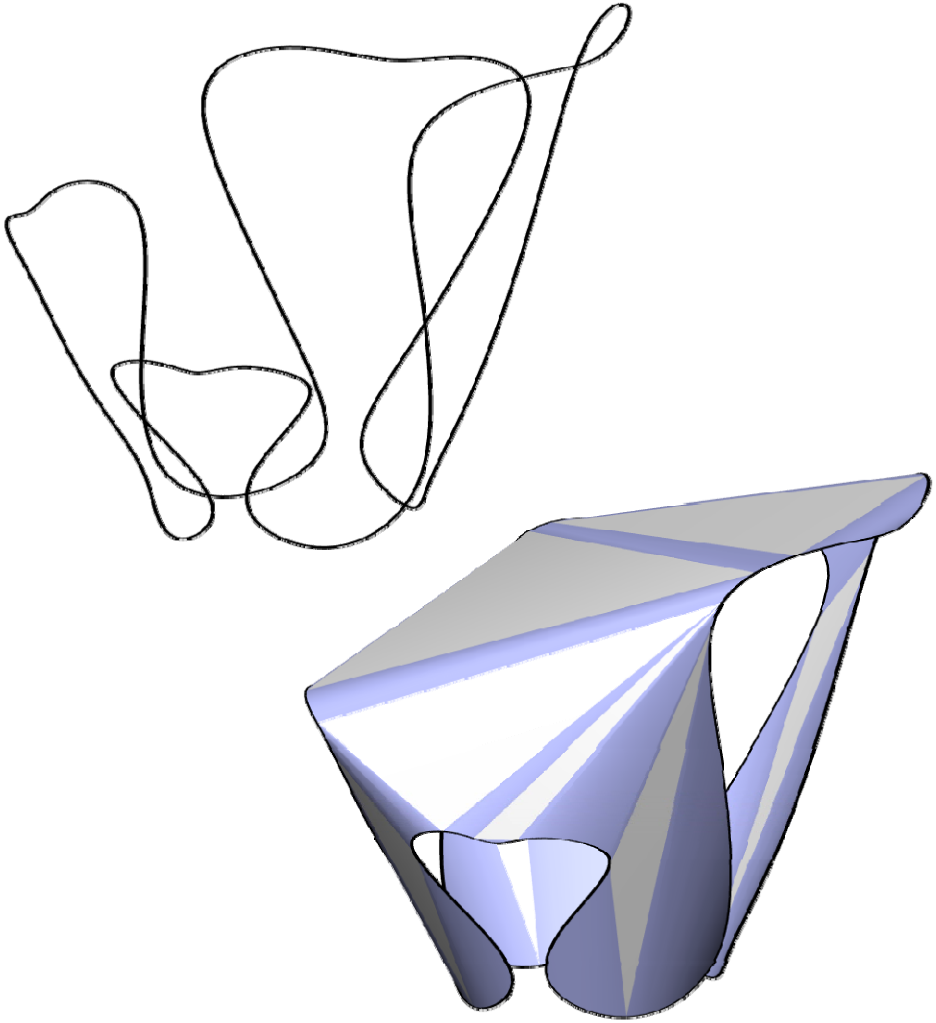


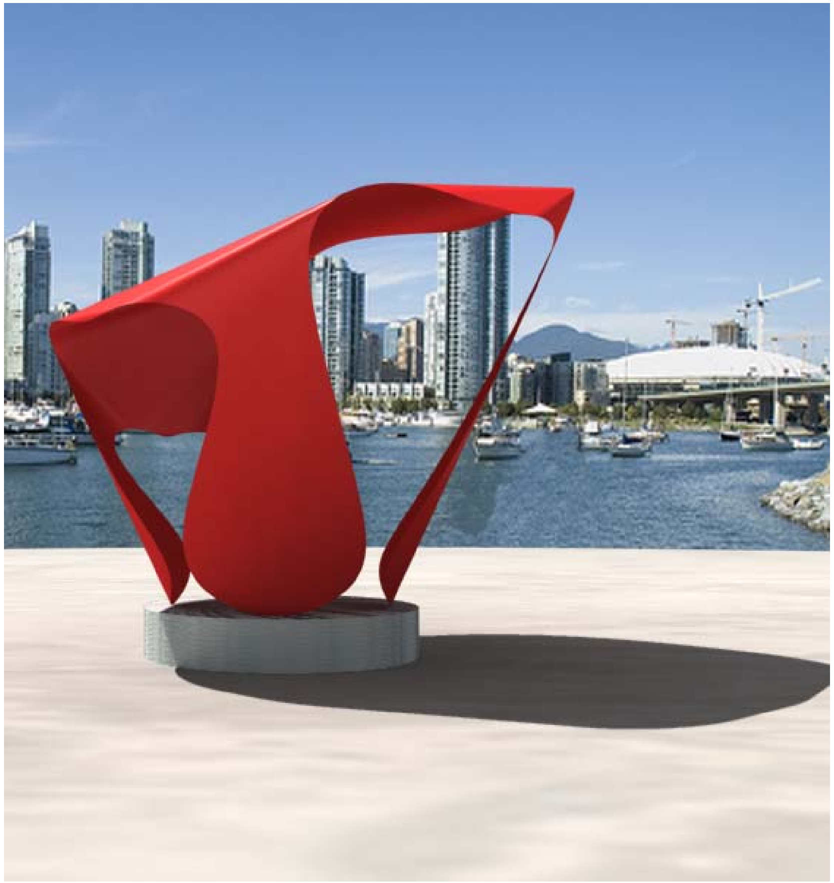
Results – Art Noveau Paper & Gold Leaf Lamp



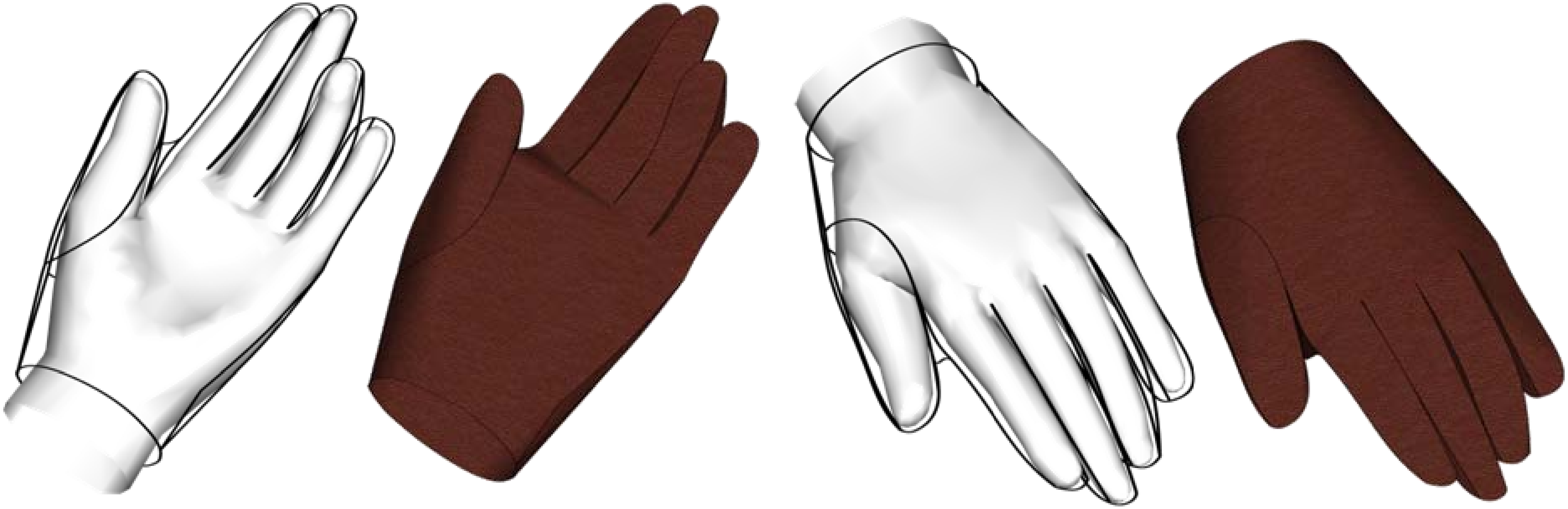


Results - Pavillion





Results – Gloves



Summary

- First robust and easy to use system for modeling general developable surfaces
  - No user expertise required
- Allows user interaction and optimization of different properties
- Based on connection between developable surfaces and CH of their boundary

Future Work

- Approximate v.s. Exact
- Further investigation of linkage between developables and convex hulls
- Understanding singularities

Questions?