CPSC424: Tutorial 7

March 3, 2025

TA: Jinfan Yang (yangjf@cs.ubc.ca)

Credits

Tutorial this time is mostly a copy of the 2019 tutorial material prepared by Jerry.

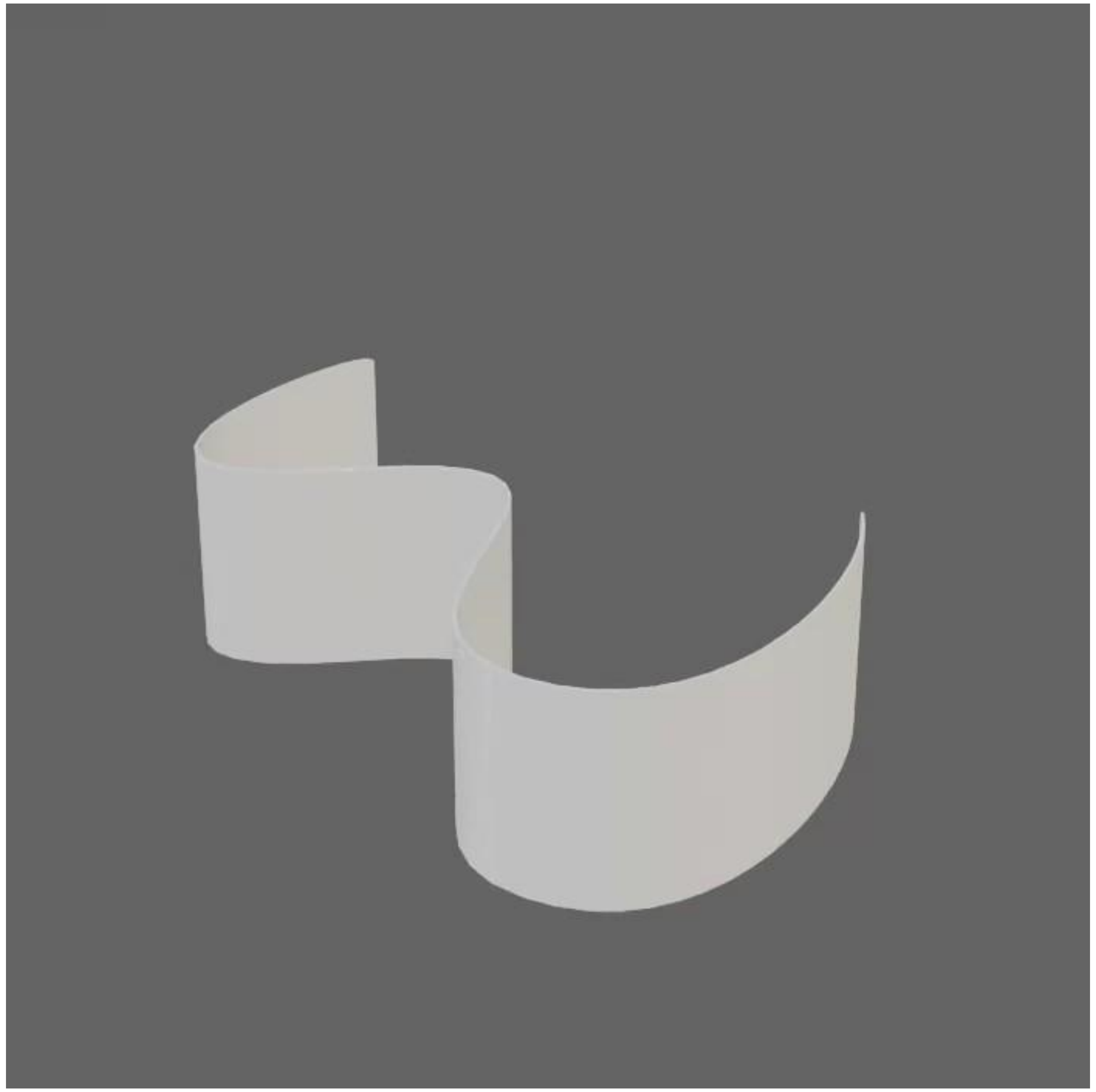
Modelling Surfaces

Surfaces can be derived from curves by simple operations:

- Extrusion
- Revolution
- Sweep
- Ruling

Modelling Surfaces

Describe the process being used to create each of the surfaces below. Be as specific as possible (*i.e.* if one process is a special case of another process, state the special case).



Modelling Surfaces

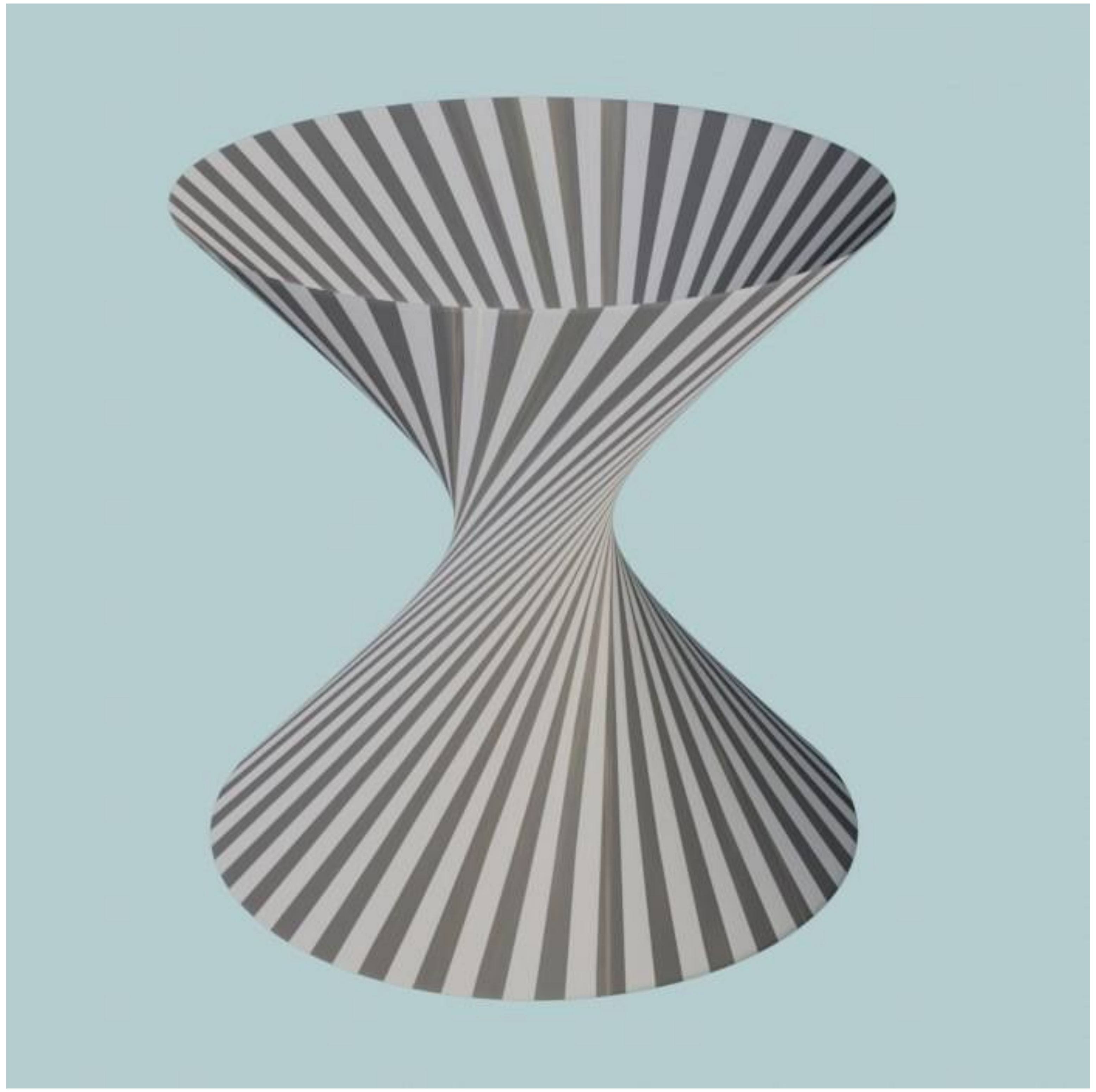
Describe the process being used to create each of the surfaces below. Be as specific as possible (*i.e.* if one process is a special case of another process, state the special case).

Modelling Surfaces

Describe the process being used to create each of the surfaces below. Be as specific as possible (*i.e.* if one process is a special case of another process, state the special case).

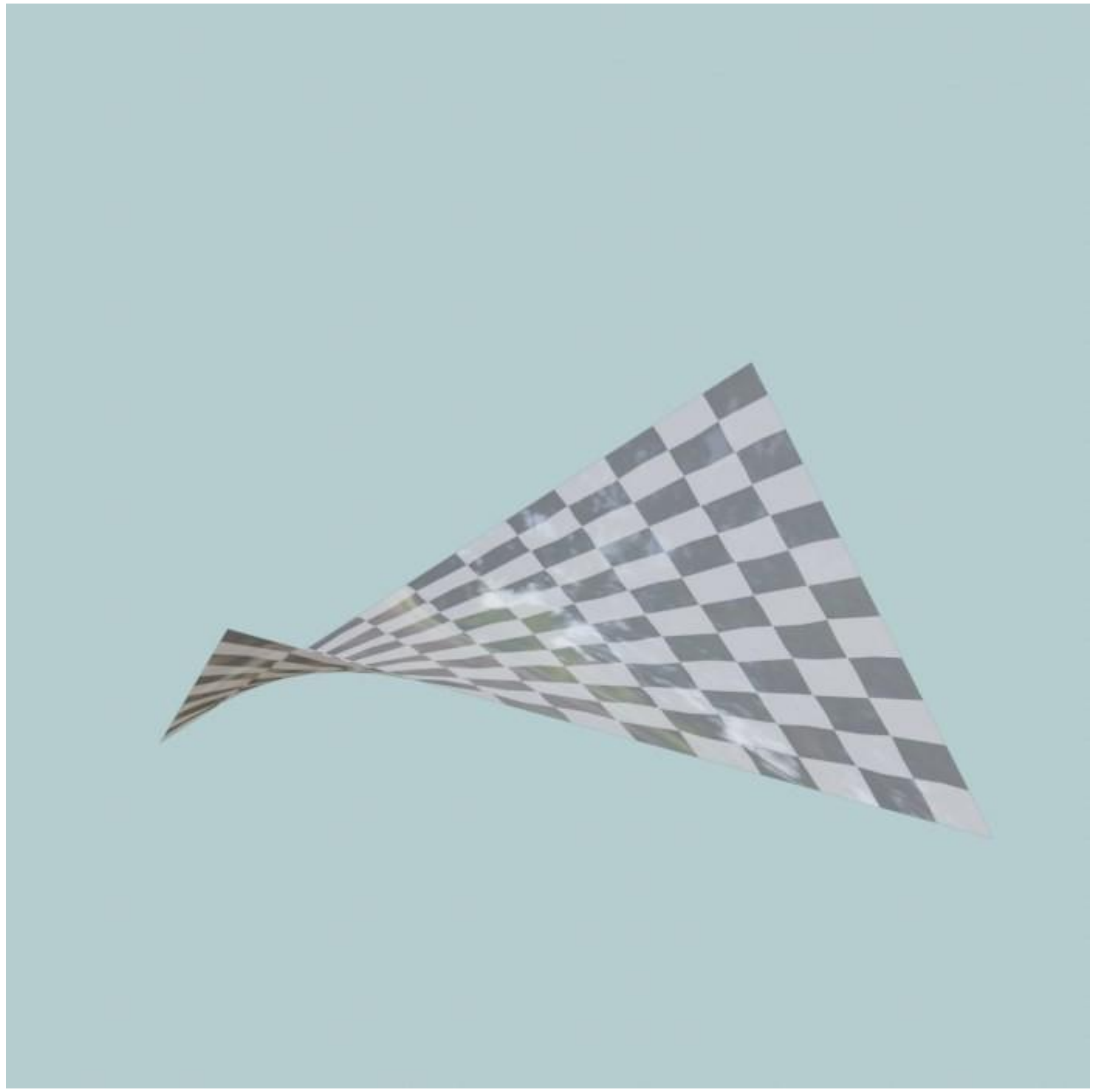
Modelling Surfaces

Describe the type of each surface below. Be as specific as possible.



Modelling Surfaces

Describe the type of each surface below. Be as specific as possible.



Tensor Product Surfaces

An online Bezier patch demo.

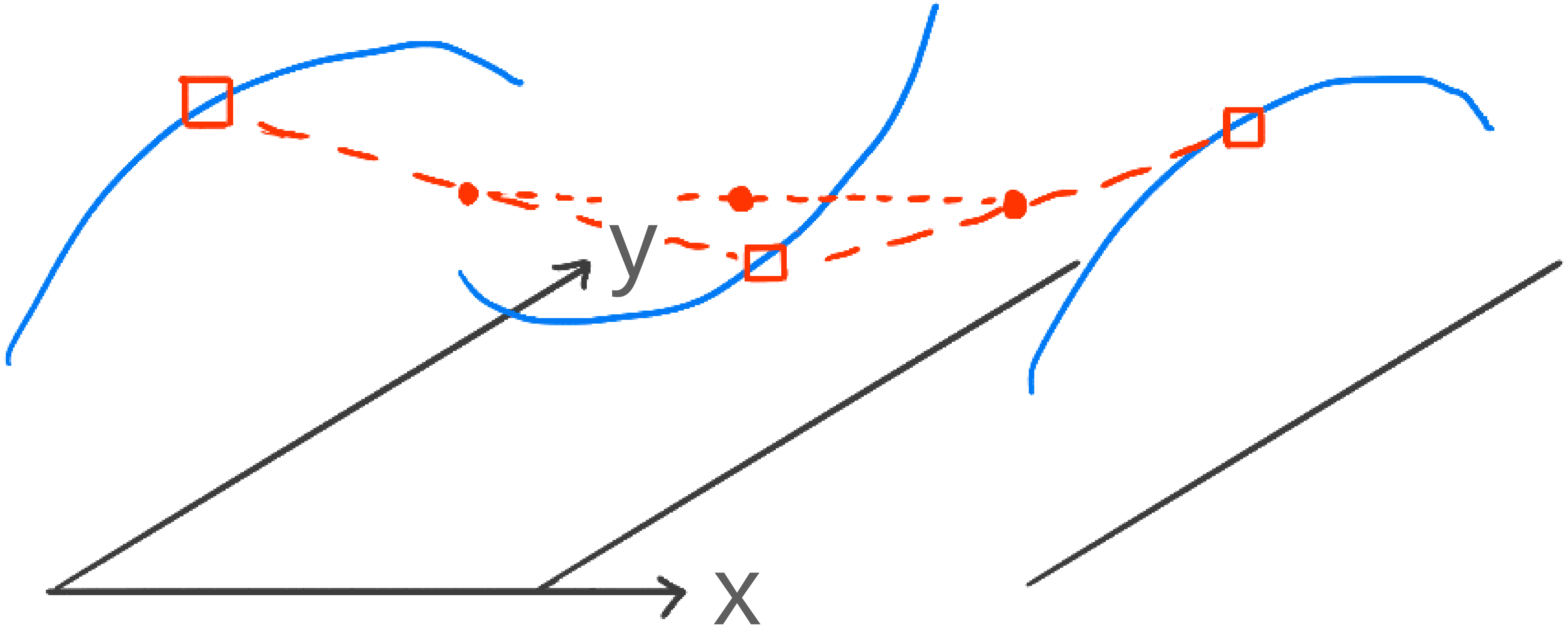
https://kovacsv.github.io/JSModeler/documentation/examples/bezier.html

Tensor Product Surfaces

Given the following nine control points which define a quadratic Bézier patch,

$$\begin{aligned}
\mathbf{b}_{0,0} &= (0, 0, 0), & \mathbf{b}_{0,1} &= (1, 0, 3), & \mathbf{b}_{0,2} &= (2, 0, 2),\\
\mathbf{b}_{1,0} &= (0, 1, 2), & \mathbf{b}_{1,1} &= (1, 1, 1), & \mathbf{b}_{1,2} &= (2, 1, 4),\\
\mathbf{b}_{2,0} &= (0, 2, 4), & \mathbf{b}_{2,1} &= (1, 2, 3), & \mathbf{b}_{2,2} &= (2, 2, 2),
\end{aligned}$$

use the de Casteljau algorithm to find the point at (s, t) = (0.5, 0.5) (the middle of the patch).



Tensor Product Surfaces

Given the following nine control points which define a quadratic Bézier patch,

$$
\begin{aligned}
\mathbf{b}_{0,0} &= (0, 0, 0), & \mathbf{b}_{0,1} &= (1, 0, 3), & \mathbf{b}_{0,2} &= (2, 0, 2), \\
\mathbf{b}_{1,0} &= (0, 1, 2), & \mathbf{b}_{1,1} &= (1, 1, 1), & \mathbf{b}_{1,2} &= (2, 1, 4), \\
\mathbf{b}_{2,0} &= (0, 2, 4), & \mathbf{b}_{2,1} &= (1, 2, 3), & \mathbf{b}_{2,2} &= (2, 2, 2),
\end{aligned}
$$

use the de Casteljau algorithm to find the point at (s, t) = (0.5, 0.5) (the middle of the patch).

First treat each column as a separate Bézier curve. We find the midpoints $\mathbf{c}_i$ of each Bézier curve as usual.

$$
\begin{aligned}
c_0 &= \frac{1}{4}(0, 0, 0) + \frac{1}{2}(0, 1, 2) + \frac{1}{4}(0, 2, 4) = (0, 1, 2) \\
c_1 &= \frac{1}{4}(1, 0, 3) + \frac{1}{2}(1, 1, 1) + \frac{1}{4}(1, 2, 3) = (1, 1, 2) \\
c_2 &= \frac{1}{4}(2, 0, 2) + \frac{1}{2}(2, 1, 4) + \frac{1}{4}(2, 2, 2) = (2, 1, 3)
\end{aligned}
$$

Tensor Product Surfaces

Given the following nine control points which define a quadratic Bézier patch,

$$
\begin{aligned}
\mathbf{b}_{0,0} &= (0, 0, 0), & \mathbf{b}_{0,1} &= (1, 0, 3), & \mathbf{b}_{0,2} &= (2, 0, 2),\\
\mathbf{b}_{1,0} &= (0, 1, 2), & \mathbf{b}_{1,1} &= (1, 1, 1), & \mathbf{b}_{1,2} &= (2, 1, 4),\\
\mathbf{b}_{2,0} &= (0, 2, 4), & \mathbf{b}_{2,1} &= (1, 2, 3), & \mathbf{b}_{2,2} &= (2, 2, 2),
\end{aligned}
$$

use the de Casteljau algorithm to find the point at (s, t) = (0.5, 0.5) (the middle of the patch).

c_0, c_1, c_2 form the control points for a Bézier curve in the other direction. Thus the middle of the patch is

$$
\begin{aligned}
F\,(0.5, 0.5) &= \frac{1}{4}c_0 + \frac{1}{2}c_1 + \frac{1}{4}c_2\\
&= \frac{1}{4}(0, 1, 2) + \frac{1}{2}(1, 1, 2) + \frac{1}{4}(2, 1, 3) = \left(1, 1, \frac{9}{4}\right).
\end{aligned}
$$

Curve Network

Flow Aligned Surfacing of Curve Networks

Hao Pan[1], Yang Liu[2], Alla Sheffer[3], Nicholas Vining[3], Changjian Li[1], Wenping Wang[1]

[1]The University of Hong Kong, [2]Microsoft Research Asia, [3]University of British Columbia

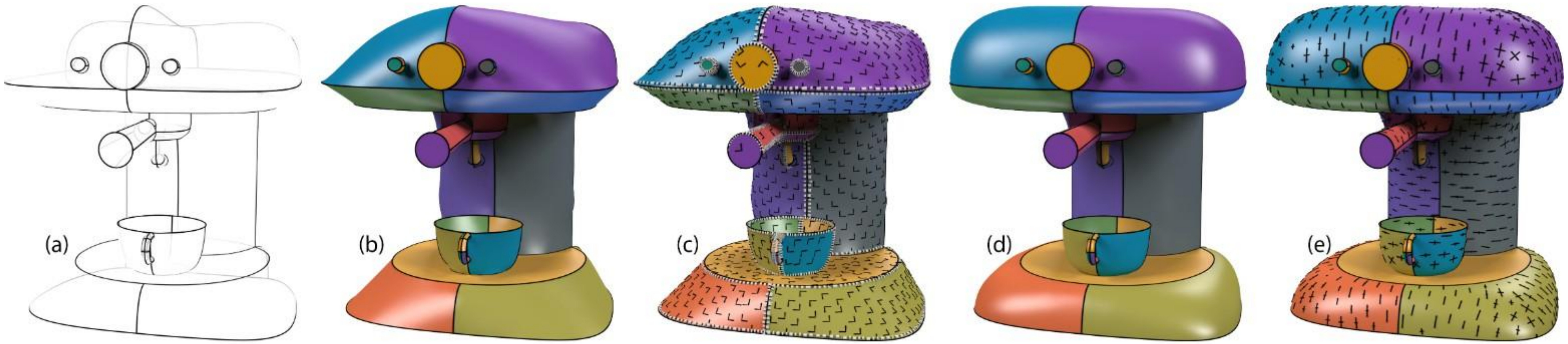


Figure 1: Surfacing an artist created curve network: (a) an input curve network of the coffee maker model; (b) initial surface; (c) the computed flow field aligned with network flow lines denoted by dashed white lines on the initial surface; (d) final surface whose curvature directions (scaled by principal curvatures) (e) are well aligned with the representative flow lines in the curve network.

Curve Network

Design-Driven Quadrangulation of Closed 3D Curves

Mikhail Bessmeltsev[1] Caoyu Wang[1] Alla Sheffer[1] Karan Singh[2]

[1]University of British Columbia [2]University of Toronto

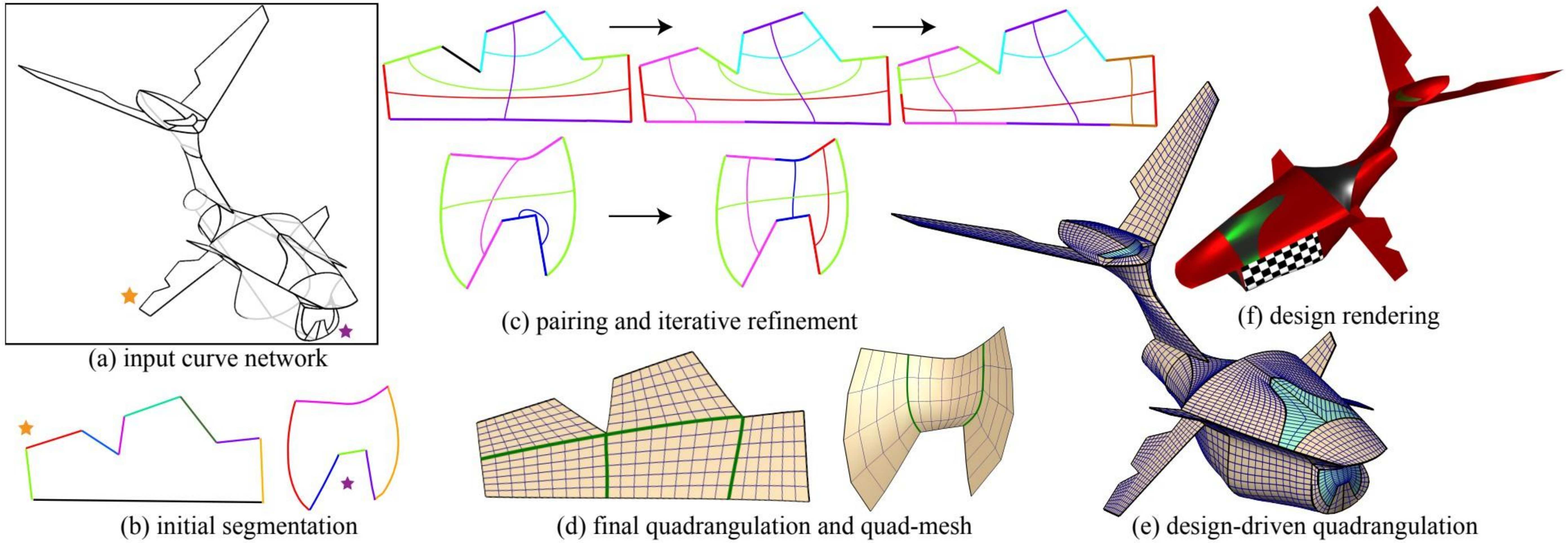


Figure 1: *Steps to quadrangulating a design network of closed 3D curves (a) : Closed curves are independently segmented (b) and iteratively paired and refined to capture dominant flow-lines as well as overall flow-line quality (c); final quadrangulation in green and dense quad-mesh (d); quadrangulations are aligned across adjacent cycles to generate a single densely sampled mesh (e), suitable for design rendering and downstream applications (f).*