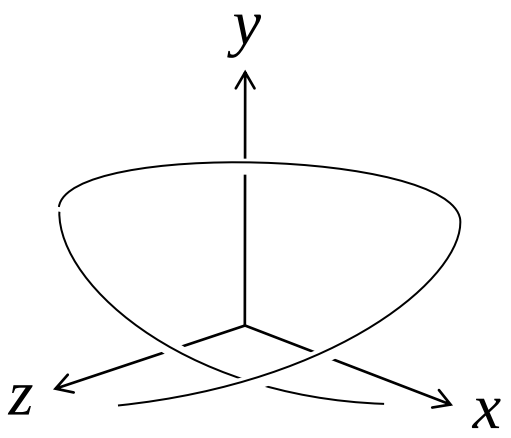
Parametric Surfaces

CS 318

Intro to Computer Graphics

John C. Hart

Space Curves



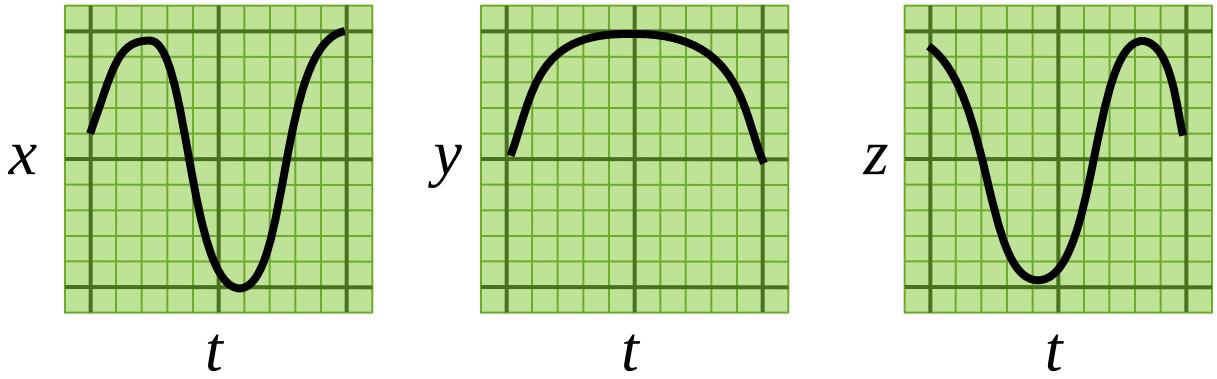
Separate into three coordinate functions

$$\mathbf{p}(t) = (x(t), y(t), z(t))$$

$$x(t) = (1\text{-}t)^3 x_0 + 3t(1\text{-}t)^2 x_1 + 3t^2(1\text{-}t) x_2 + t^3 x_3$$

$$y(t) = (1\text{-}t)^3 y_0 + 3t(1\text{-}t)^2 y_1 + 3t^2(1\text{-}t) y_2 + t^3 y_3$$

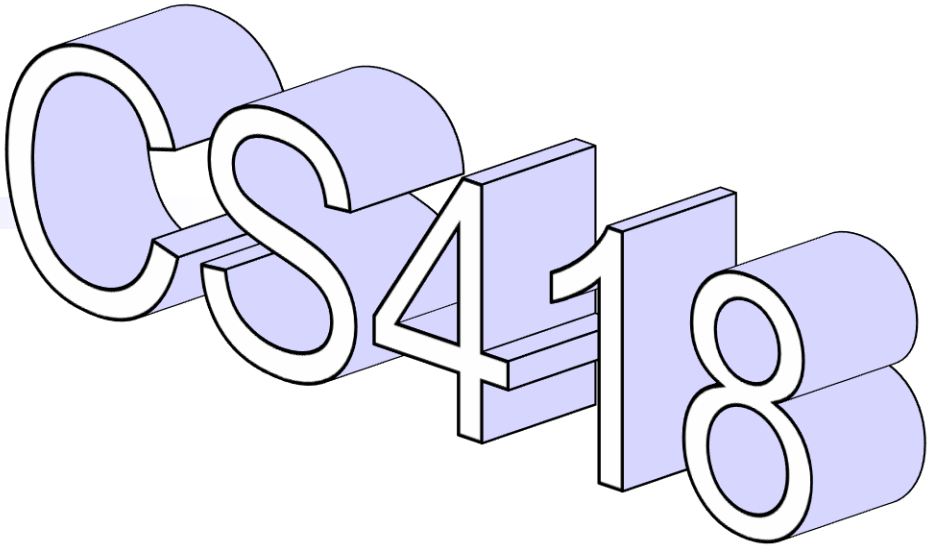
$$z(t) = (1\text{-}t)^3 z_0 + 3t(1\text{-}t)^2 z_1 + 3t^2(1\text{-}t) z_2 + t^3 z_3$$



Make your own roller-coaster ride

- Camera position along space curve
- Look at point is next position along space curve (tangent)
- Up direction is cross product of vector to next positon with vector to previous position (binormal)

Extrusion

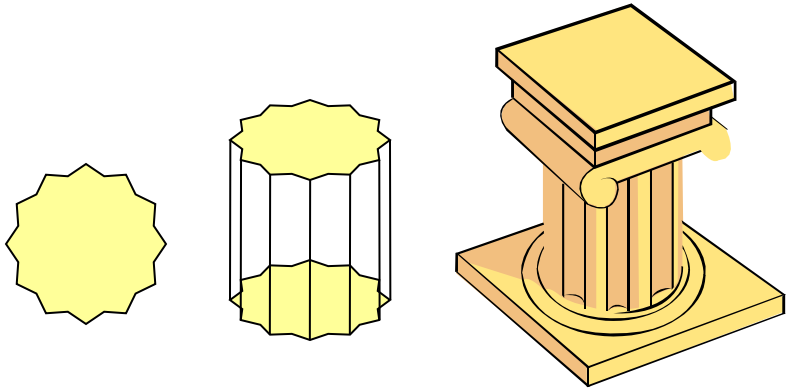


- Two 3-D copies of each 2-D curve

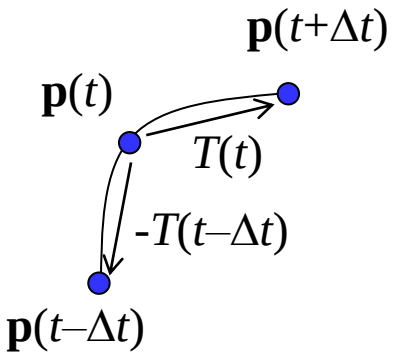
$$\mathbf{p}_0(t) = (x(t), y(t), 0)$$
$$\mathbf{p}_1(t) = (x(t), y(t), 1)$$

- Create a mesh of quads (or tri-strip)

$$\mathbf{p}_0(t),\ \mathbf{p}_1(t),\ \mathbf{p}_0(t+\Delta t),\ \mathbf{p}_1(t+\Delta t)$$



Generalized Cylinder



- Construct a 2-D profile curve

$$\mathbf{q}(s) = (a(s), b(s))$$

- Construct a space curve

$$\mathbf{p}(t) = (x(t), y(t), z(t))$$

- Construct a Frenet frame at each point along space curve

$$T(t) = \mathbf{p}(t+\Delta t) - \mathbf{p}(t)$$

$$B(t) = T(t) \times -T(t - \Delta t)$$

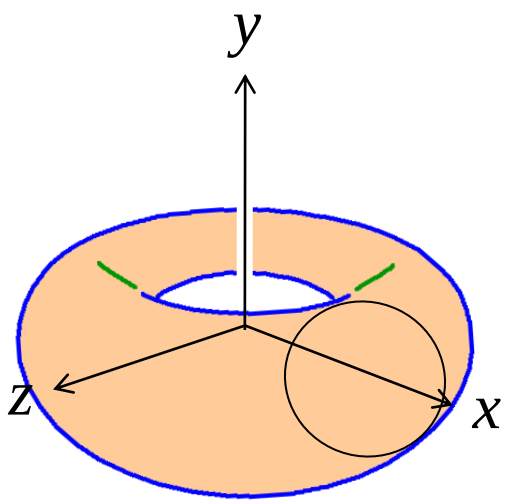
$$N(t) = B(t) \times T(t)$$

(all normalized)

- Plot 2-D curve in (N,B) space

$$\mathbf{gc}(s,t) = \mathbf{p}(t) + a(s)\, N(t) + b(s)\, B(t)$$

Revolution

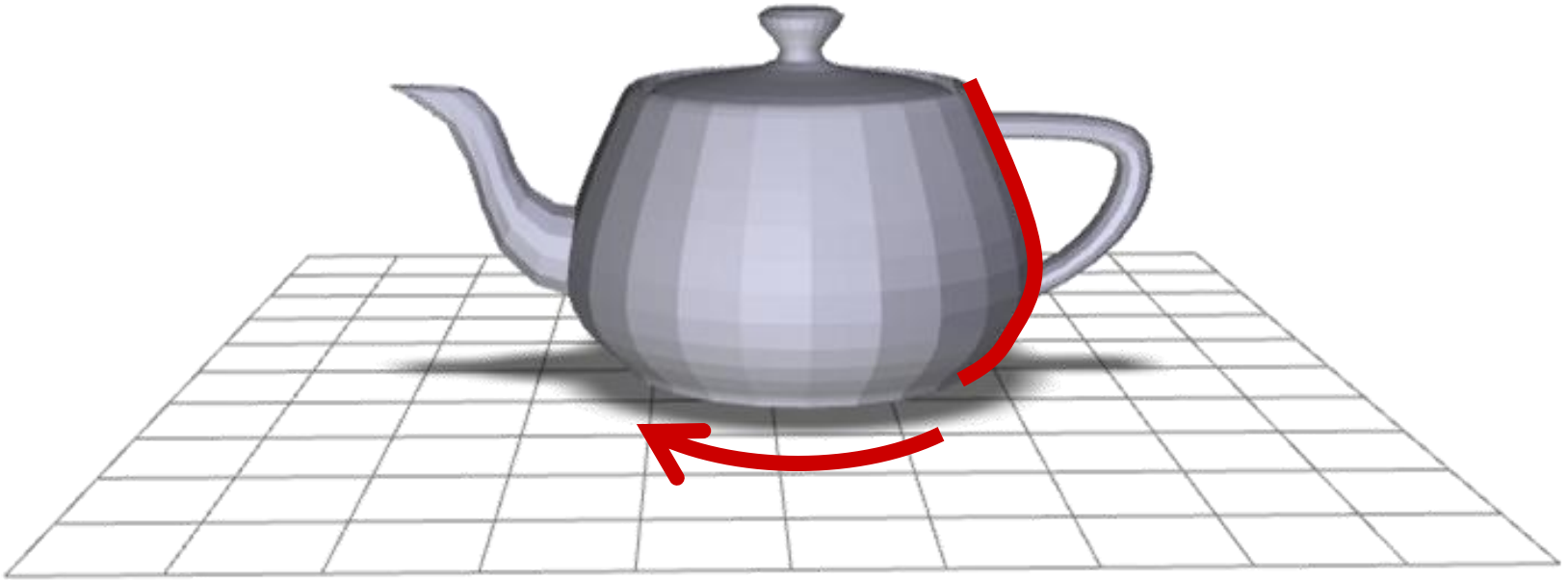


- Construct a 2-D profile curve

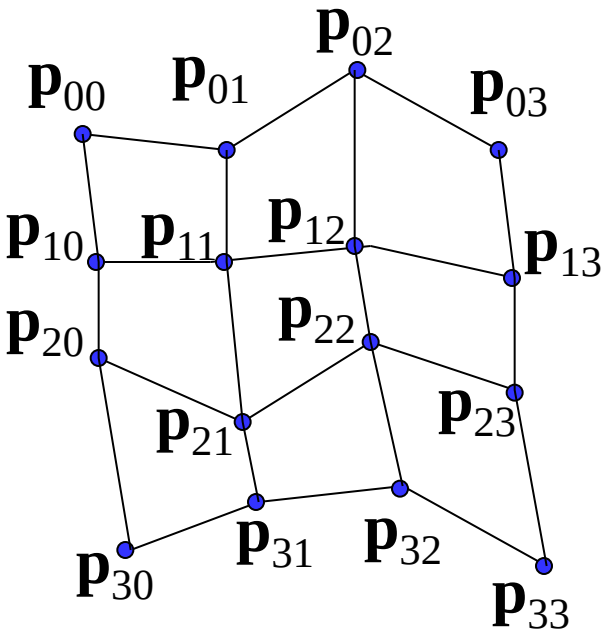
$$\mathbf{q}(t) = (x(t), y(t))$$

- Rotate about y axis

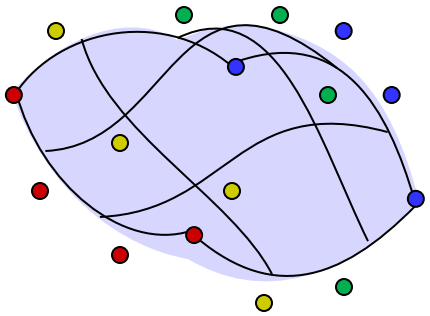
$$\mathbf{p}(s,t) = (x(t)\cos 2\pi s,\ y(t),\ x(t)\sin 2\pi s)$$



Bezier Patches



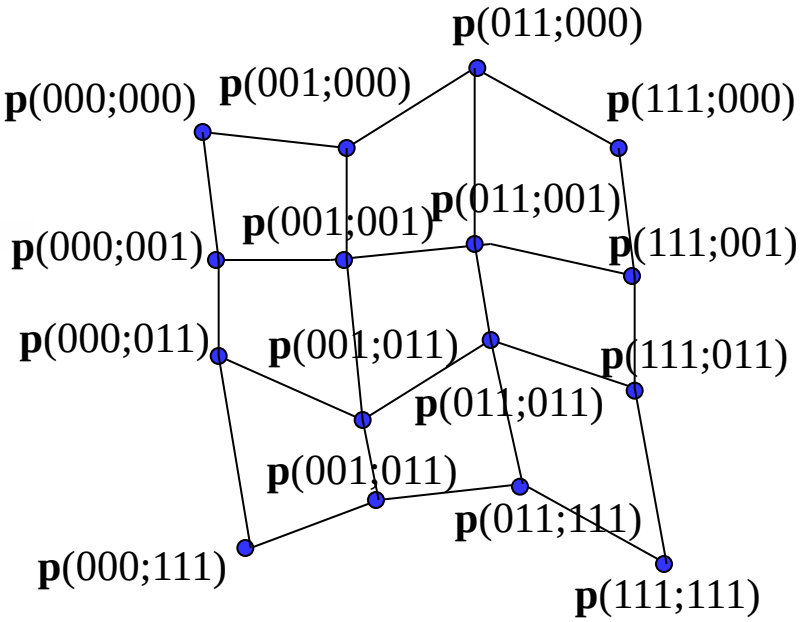
- Bezier patch
  - Tensor product of two Bezier curves
$$p(s,t) = \sum_{j=1}^{n}\sum_{i=1}^{n} B_j^n(s)B_i^n(t)\mathbf{p}_{ij}$$
  - Product of Bernstein polynomials
$$p(s,t) = \sum_{j=1}^{n}\sum_{i=1}^{n} \left(B_j^n(s)B_i^n(t)\right)\mathbf{p}_{ij}$$
  - Bernstein interpolation of Bernstein polynomials
$$p(s,t) = \sum_{j=1}^{n} B_j^n(s)\left(\sum_{i=1}^{n} B_i^n(t)\left(\mathbf{p}_i\right)\right)_j$$

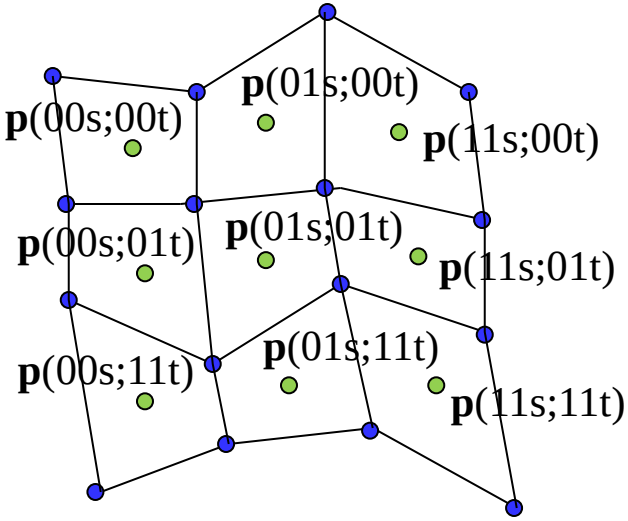


- Works same way for B-splines

Blossoming Patches

- Curves: p(t) → p(t,t,t)
- Patches: p(s,t) → p(s,s,s;t,t,t)
- Variables not allowed to cross the semicolon
- In patches, bilinear interpolation replaces linear interpolation in curves

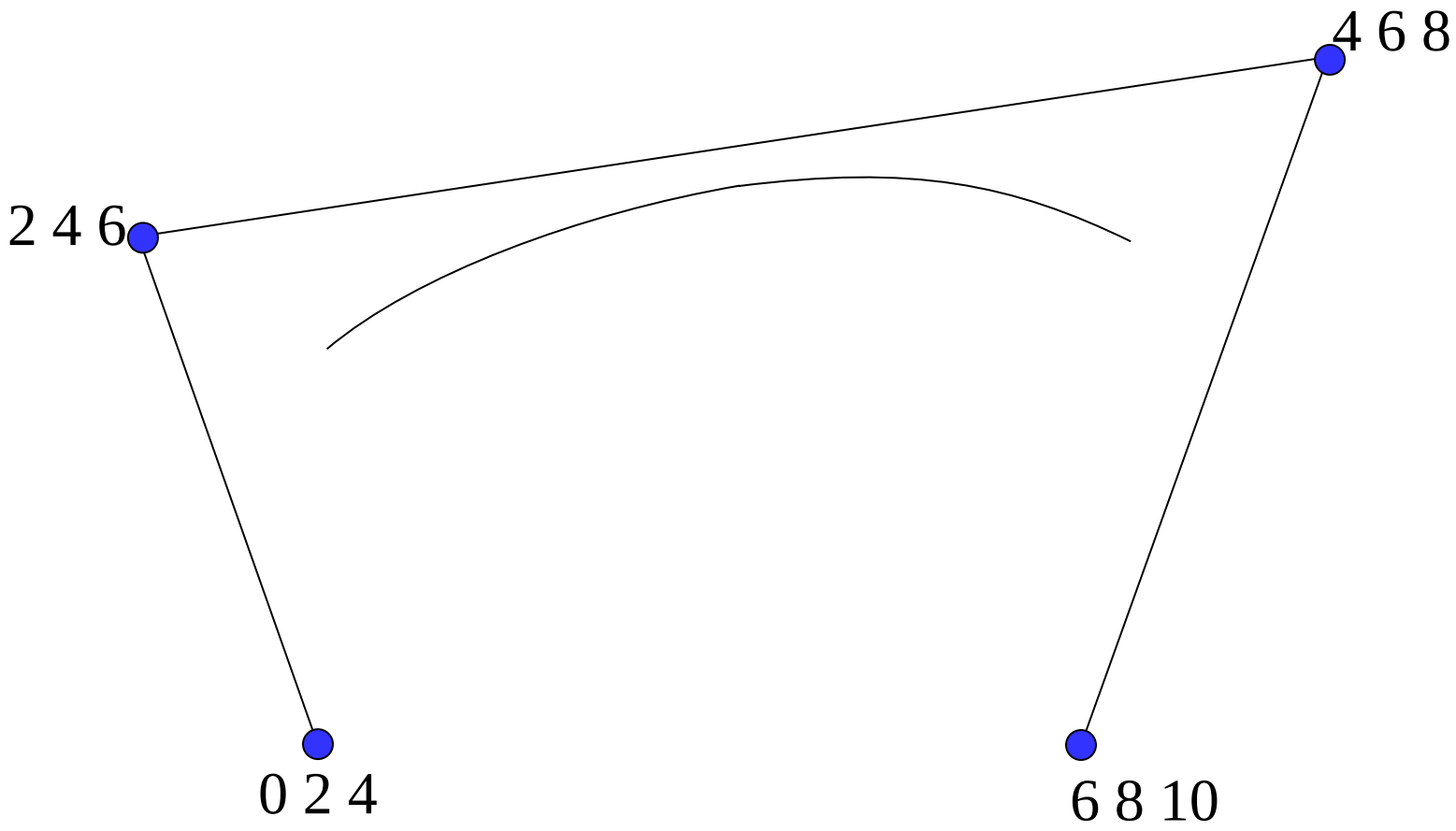




… etc.

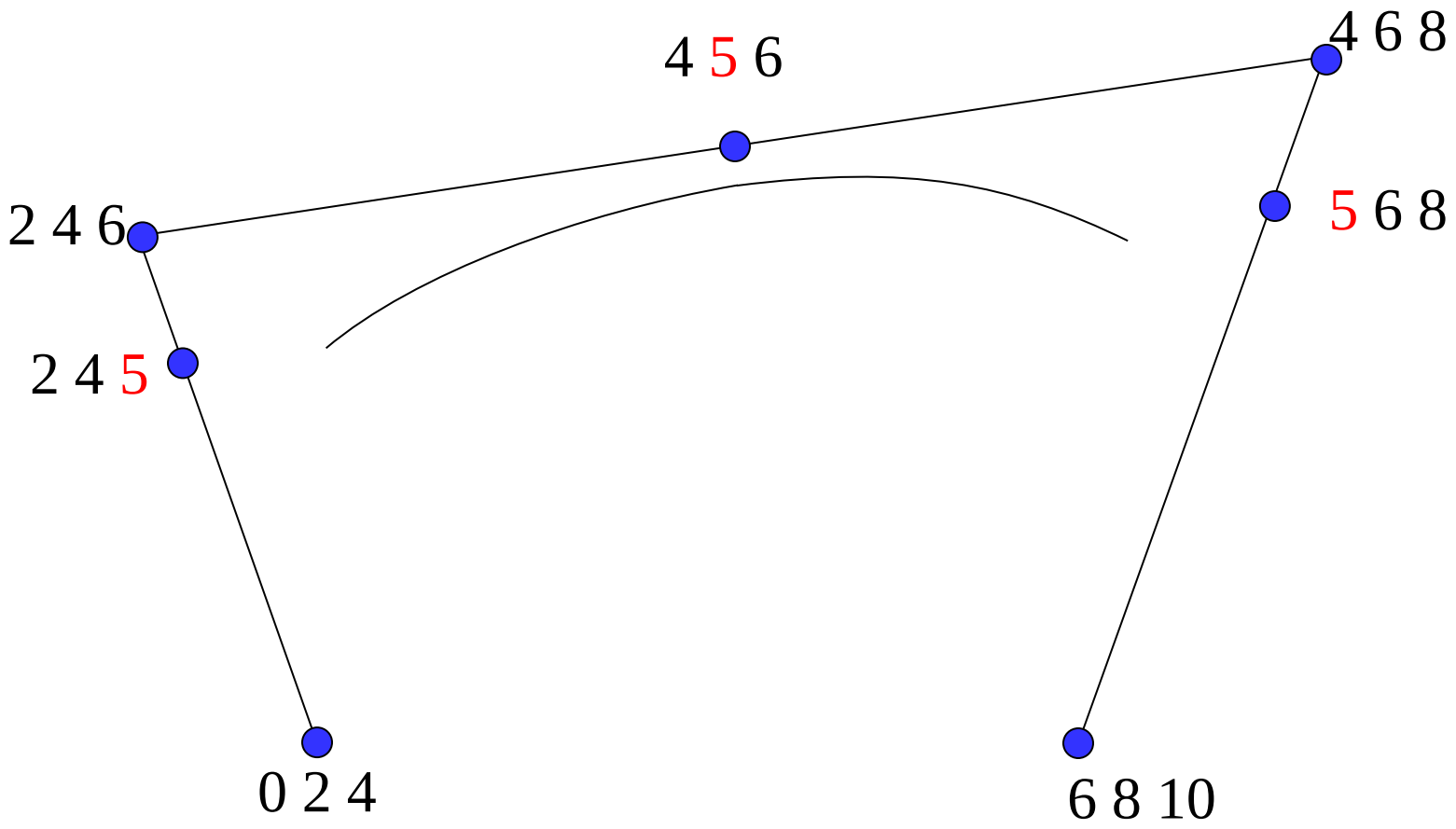
Knot Insertion

[0 2 4 6 8]



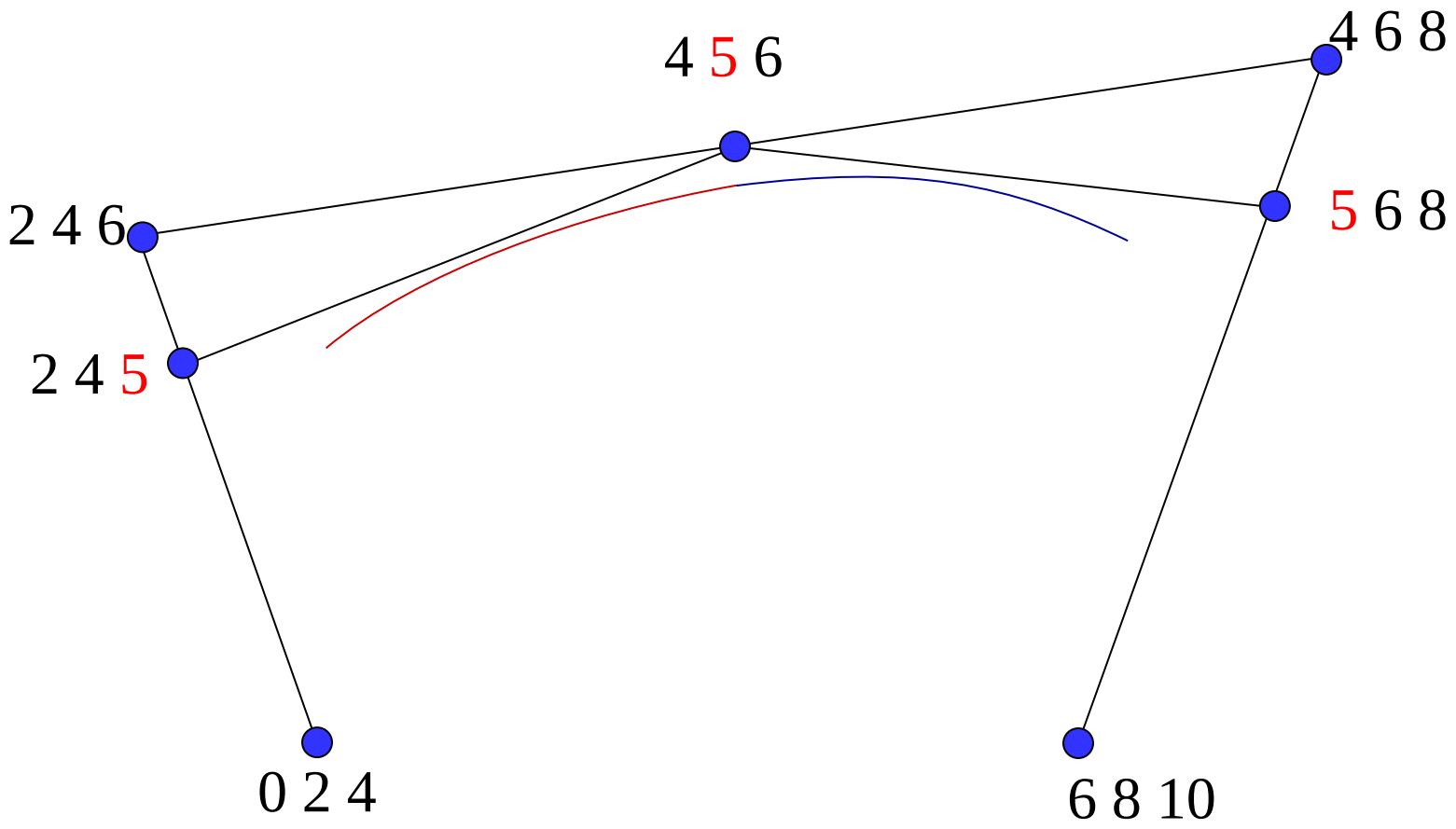
Knot Insertion

[0 2 4 5 6 8 10]

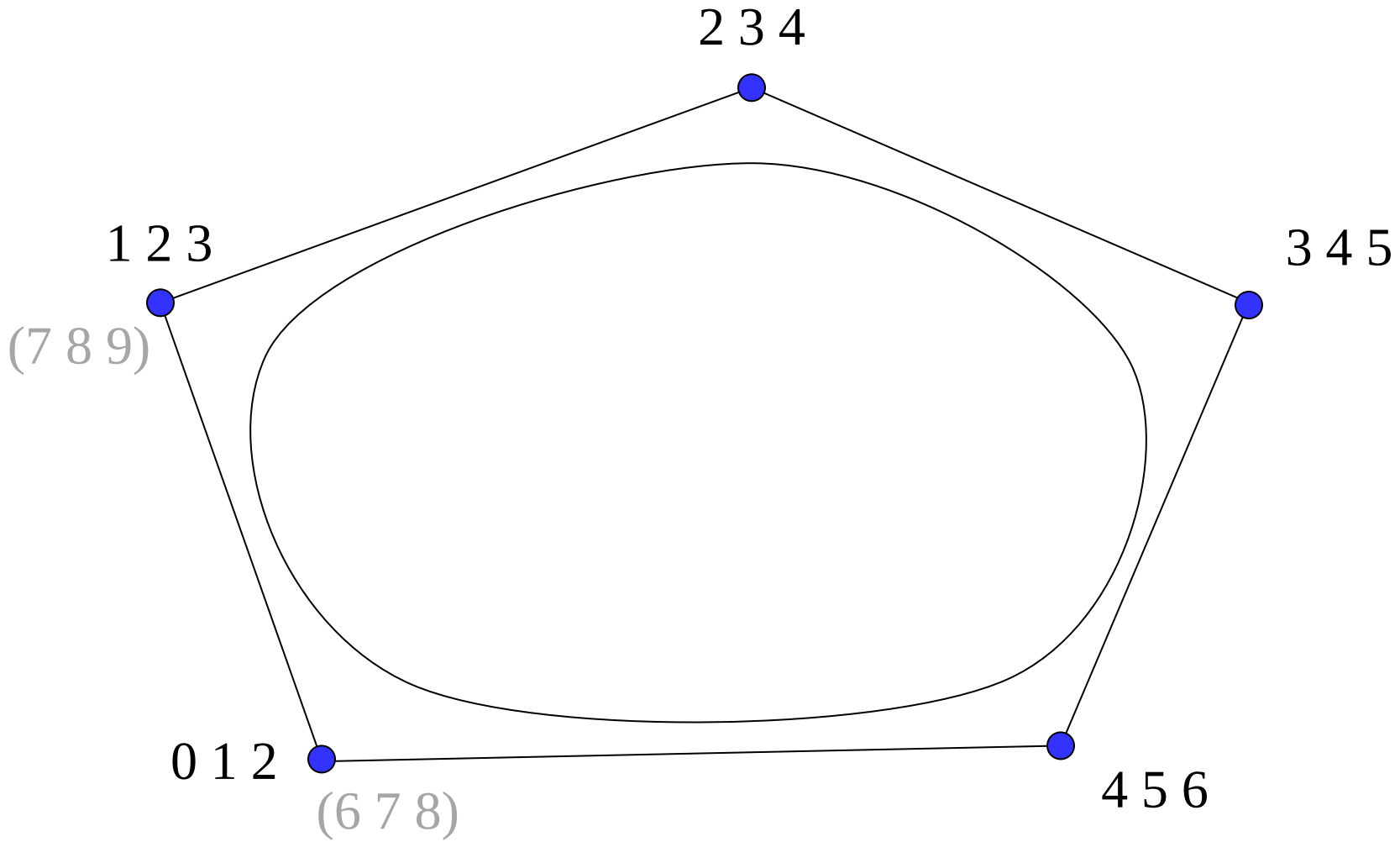


Knot Insertion

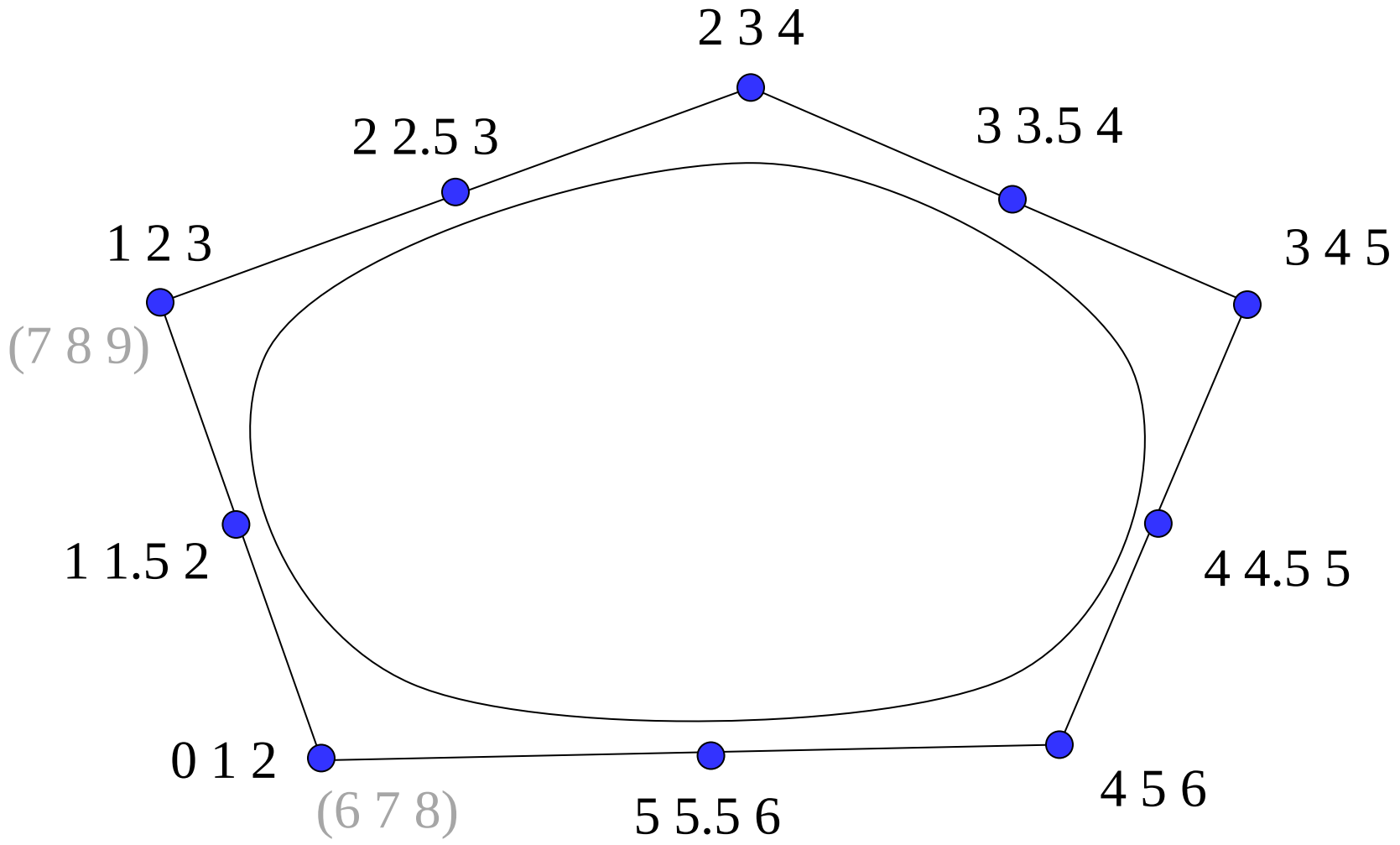
[0 2 4 5 6 8 10]



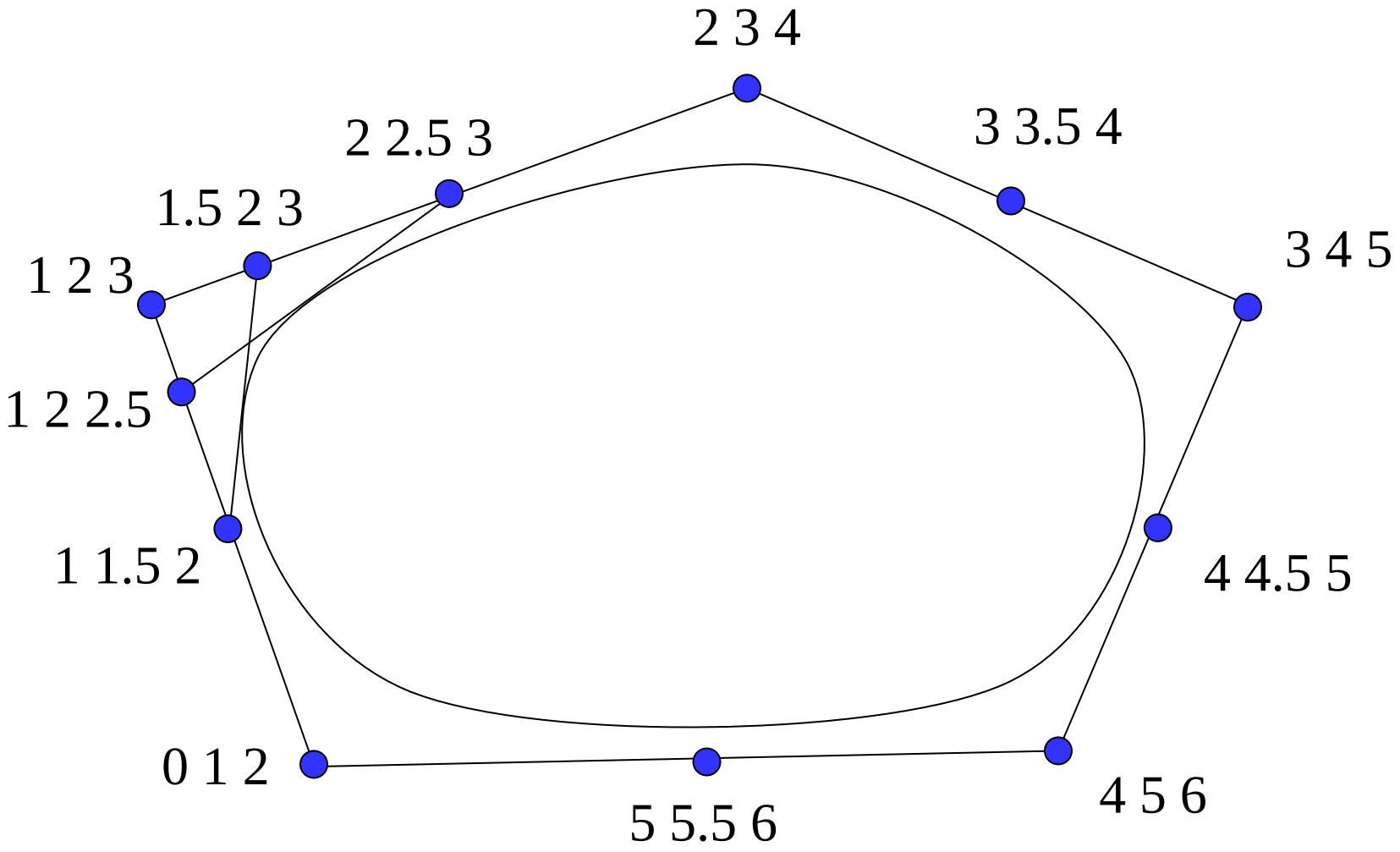
Loop Knot Insertion



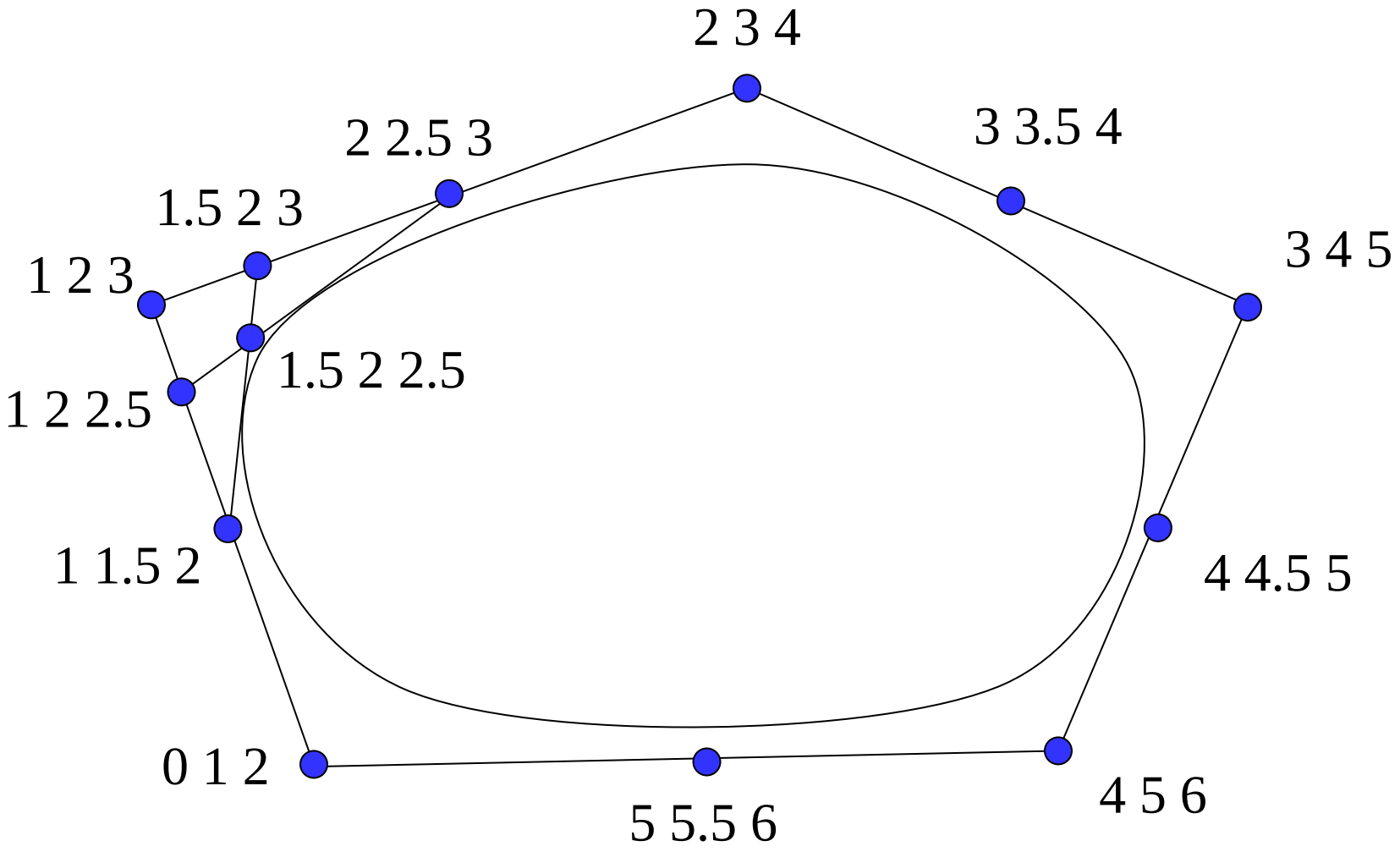
Loop Knot Insertion



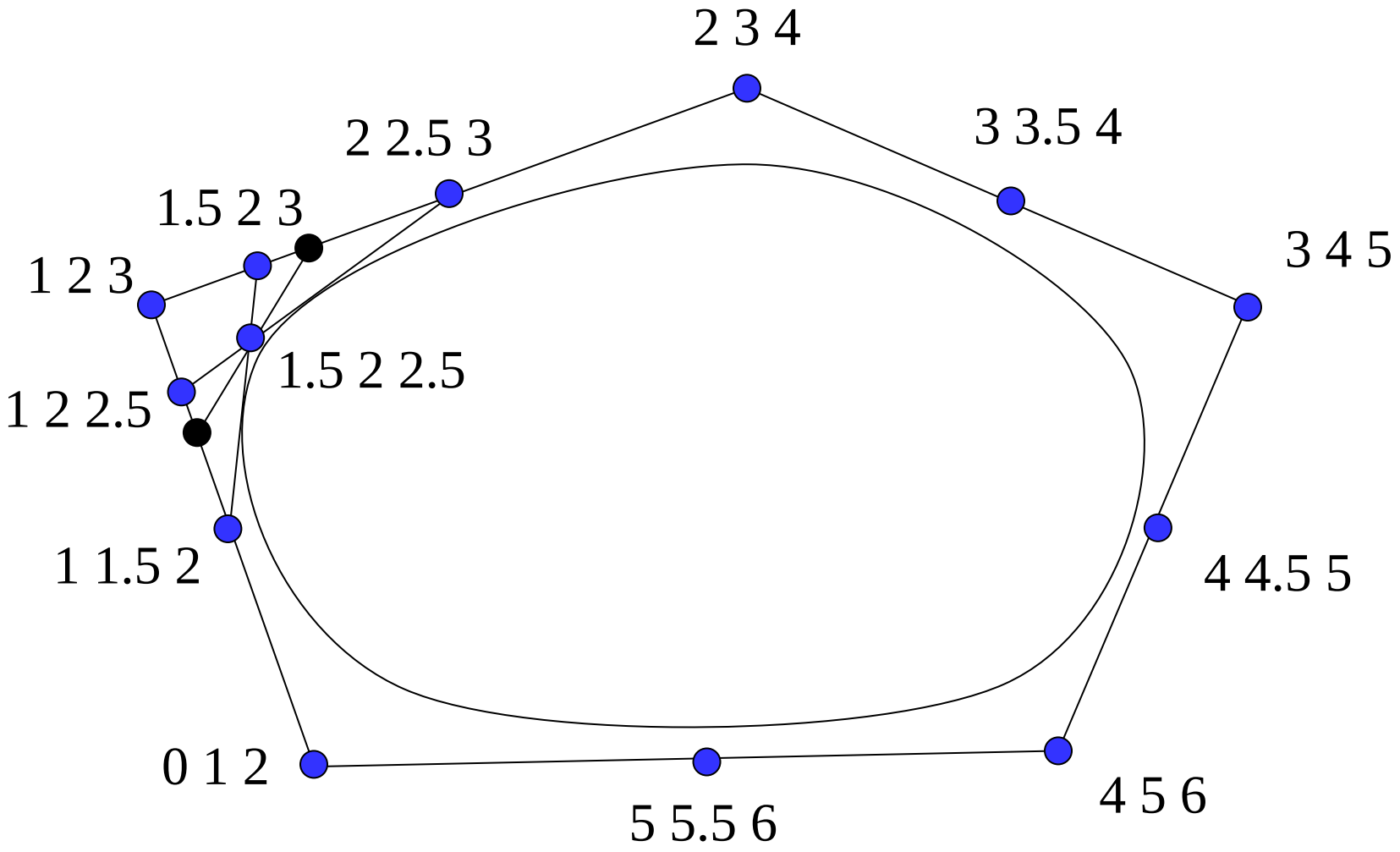
Loop Knot Insertion



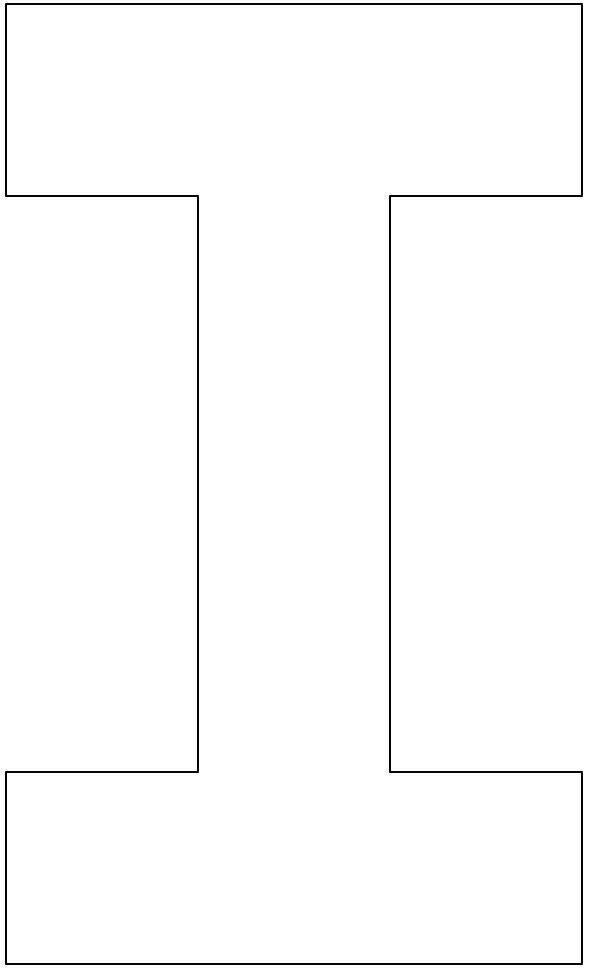
Loop Knot Insertion



Loop Knot Insertion

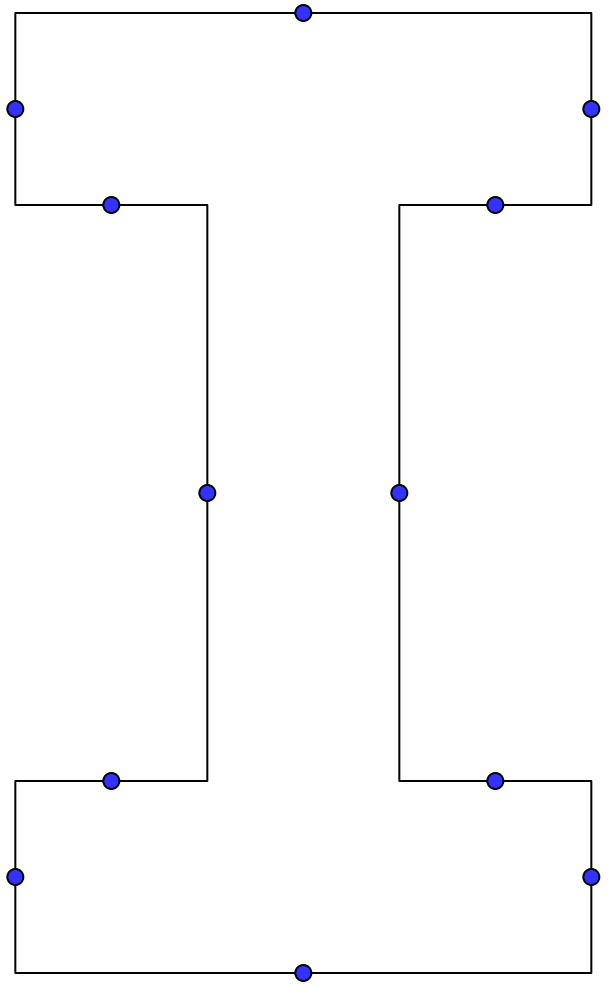


Smoothing a Polygon



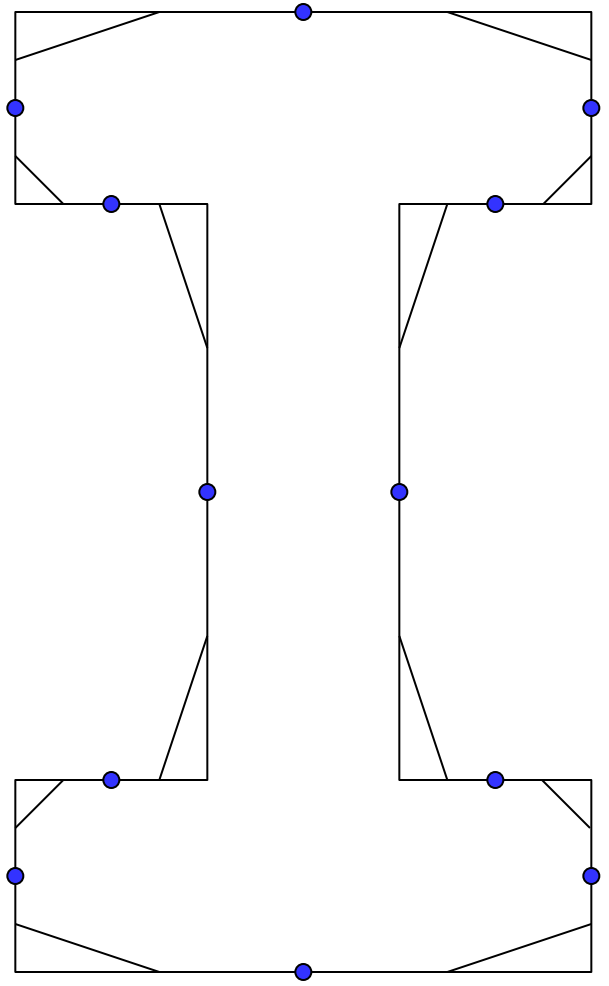
Smoothing a Polygon

1. Add edge midpoints



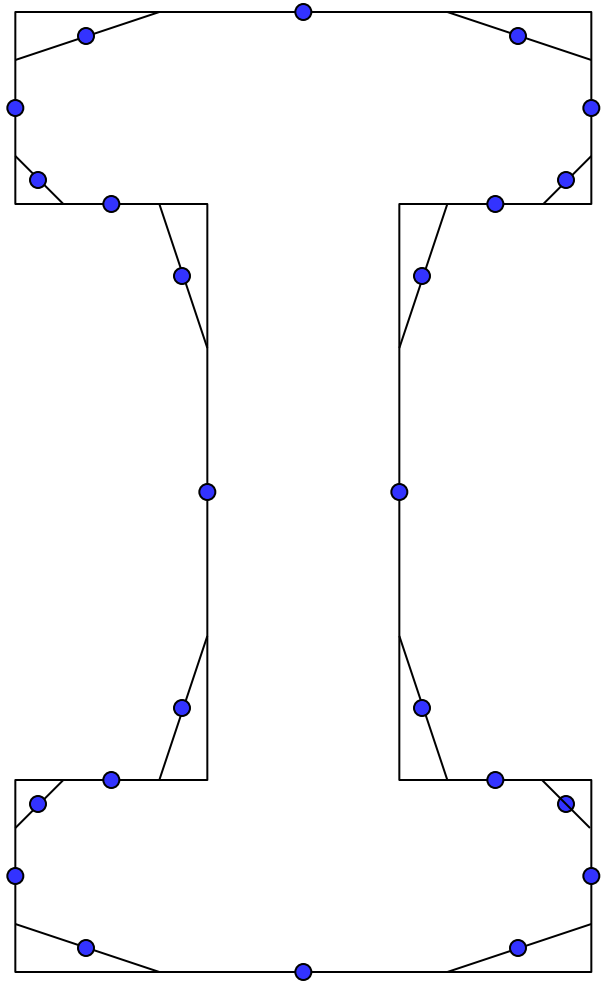
Smoothing a Polygon

1. Add edge midpoints
2. Add struts
   - Struts connect midpoints of segments from vertices to edge midpoints
   - One strut per vertex



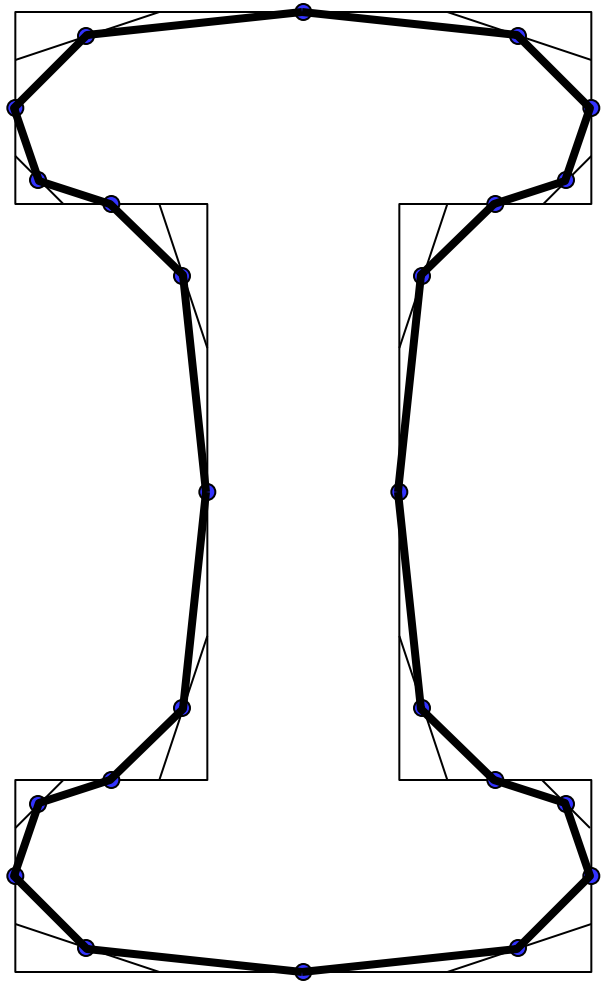
Smoothing a Polygon

1. Add edge midpoints
2. Add struts
   - Struts connect midpoints of segments from vertices to edge midpoints
   - One strut per vertex
3. Add strut midpoints



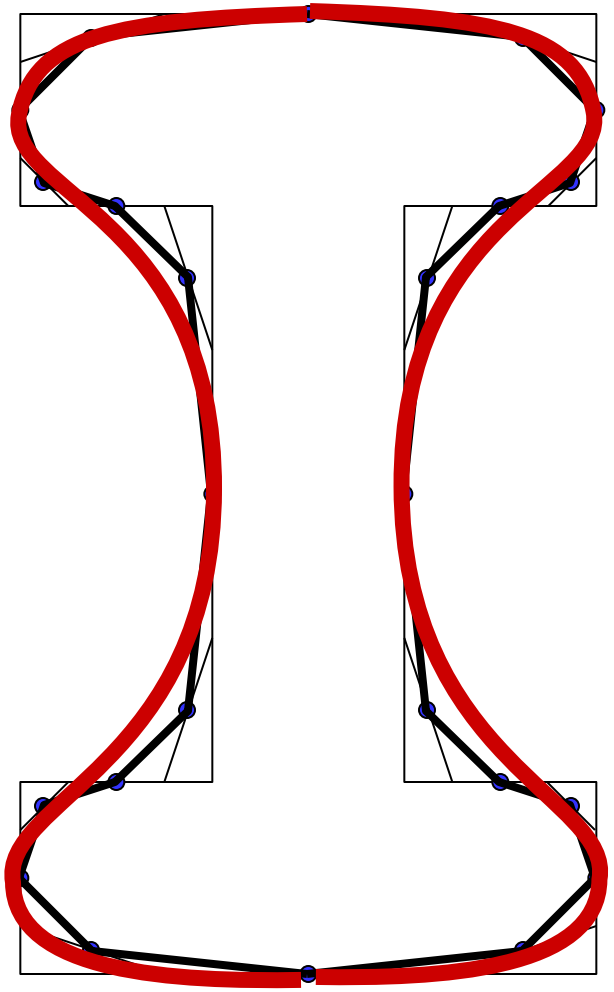
Smoothing a Polygon

1. Add edge midpoints
2. Add struts
   - Struts connect midpoints of segments from vertices to edge midpoints
   - One strut per vertex
3. Add strut midpoints
4. Connect



Smoothing a Polygon

1. Add edge midpoints
2. Add struts
   - Struts connect midpoints of segments from vertices to edge midpoints
   - One strut per vertex
3. Add strut midpoints
4. Connect
5. Repeat

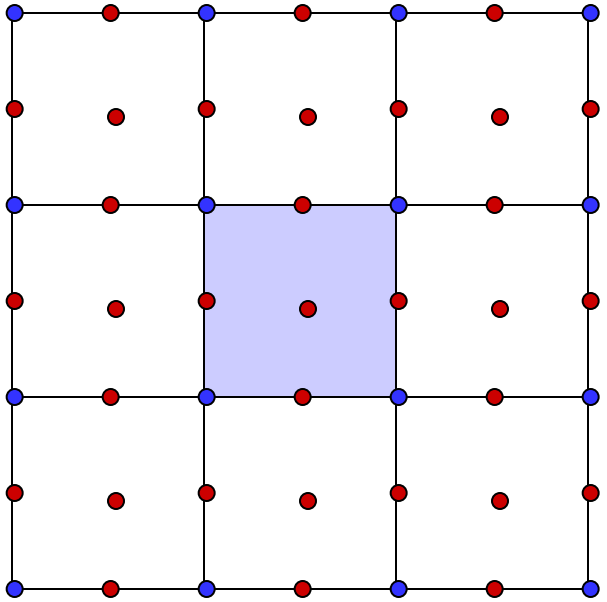


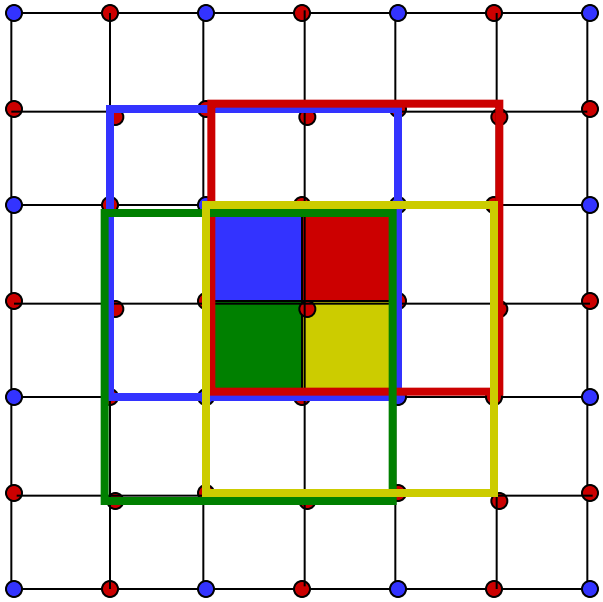
B-Spline Patches

- Tensor product of two curves

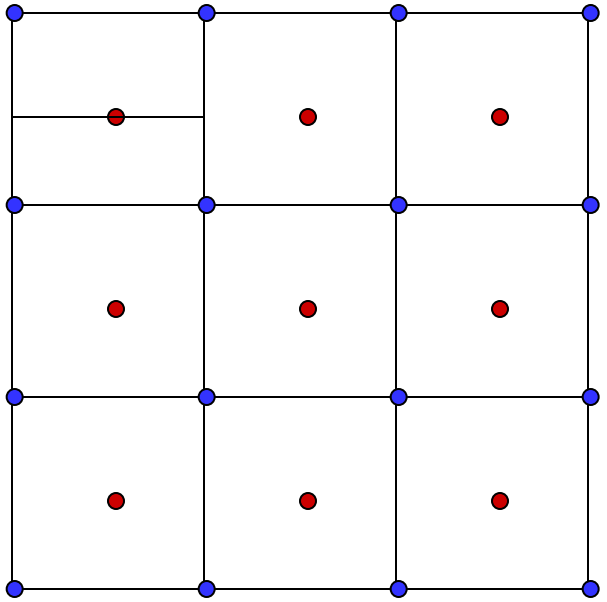
$$\mathbf{p}(s,t) = \sum_{j=0}^{n} \sum_{i=0}^{n} N_j^n(s) N_i^n(t) \mathbf{p}_{ij}$$

- Need to subdivide control points to create four sub-patches
- Need to generate new control points
  - vertex points (replacing control points)
  - edge points
  - face points





Face Points



- Approximate edge points as midpoint of control points

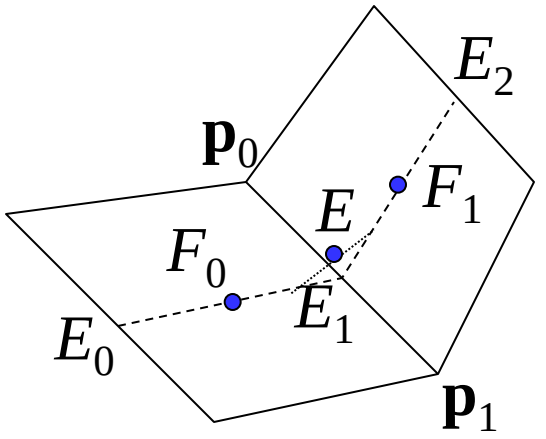
  E = 1/2 **p** + 1/2 **p**

- Face point is midpoint of approximate edge points

  F = 1/2 E + 1/2 E

  = 1/4 **p** + 1/4 **p** + 1/4 **p** + 1/4 **p**

Edge Points



- Face points are midpoints between approx. edge points
- Approx. edge point is midpoint between control points
- Actual edge point is midpoint between midpoints between approx edge point and face points

$$
\begin{aligned}
E = \; & 1/2\,(1/2\,(1/2\,E_0 + 1/2\,E_1) + 1/2\,E_1) + \\
& 1/2\,(1/2\,E_1 + 1/2\,(1/2\,E_1 + 1/2\,E_2)) \\
= \; & 1/2\,(1/2\,F_0 + 1/2\,(1/2\,\mathbf{p}_0 + 1/2\,\mathbf{p}_1)) + \\
& 1/2\,(1/2\,(1/2\,\mathbf{p}_0 + 1/2\,\mathbf{p}_1) + 1/2\,F_1) \\
= \; & 1/4\,(F_0 + \mathbf{p}_0 + \mathbf{p}_1 + F_1)
\end{aligned}
$$

Vertex Points

$$V_0 = 1/4\, E_0 + 1/2\, \mathbf{p}_0 + 1/4\, E_1$$

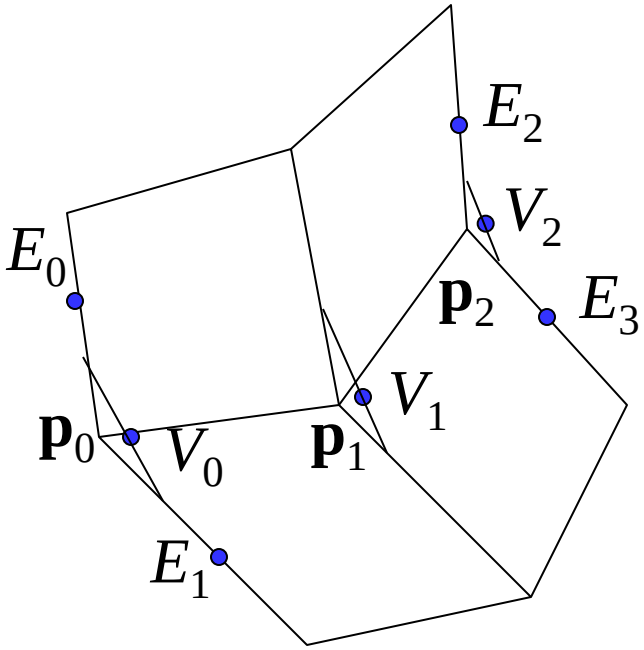
$$V_2 = 1/4\, E_2 + 1/2\, \mathbf{p}_2 + 1/4\, E_3$$

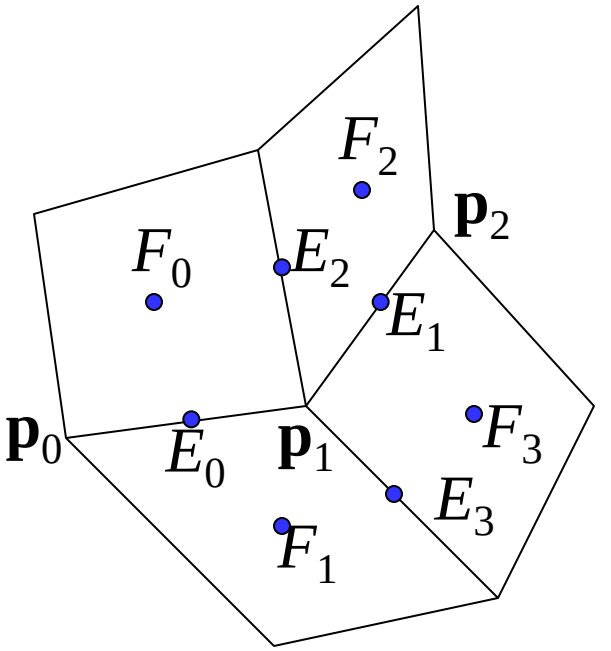
$$V = 1/2\,(1/2\,(1/2\,V_0 + 1/2\,V_1) + 1/2\,V_1) + 1/2\,(1/2\,V_1 + 1/2\,(1/2\,V_1 + 1/2\,V_2)$$

$$= 1/4\,(1/4\,(F_0 + F_1 + \mathbf{p}_0 + \mathbf{p}_1) + 1/4\,(F_2 + F_3 + \mathbf{p}_1 + \mathbf{p}_2) + 2\,V_1)$$

$$= 1/4\,(1/4\,(F_0 + F_1 + F_2 + F_3) + 1/4\,(\mathbf{p}_0 + 2\,\mathbf{p}_1 + \mathbf{p}_2) + 2/4\,(E_2 + E_3 + 2\,\mathbf{p}_1))$$

$$= 1/16(F_0 + F_1 + F_2 + F_3 + 2E_0 + 2E_1 + 2E_2 + 2E_3 + 4\mathbf{p}_1)$$





Catmull-Clark Subdiv

- Add new “face” vertex at each face centroid

  centroid = average of face’s vertices

- Add new “edge” vertex at the average of each edge’s endpoints and adjacent face centroids

- Move each vertex to a new position that is…

$1 \times$ The average of its adjacent face centroids

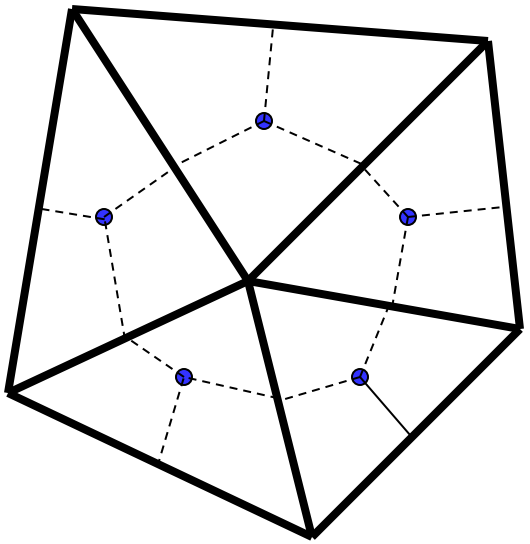
+

$2 \times$ the average of its adjacent edge midpoints

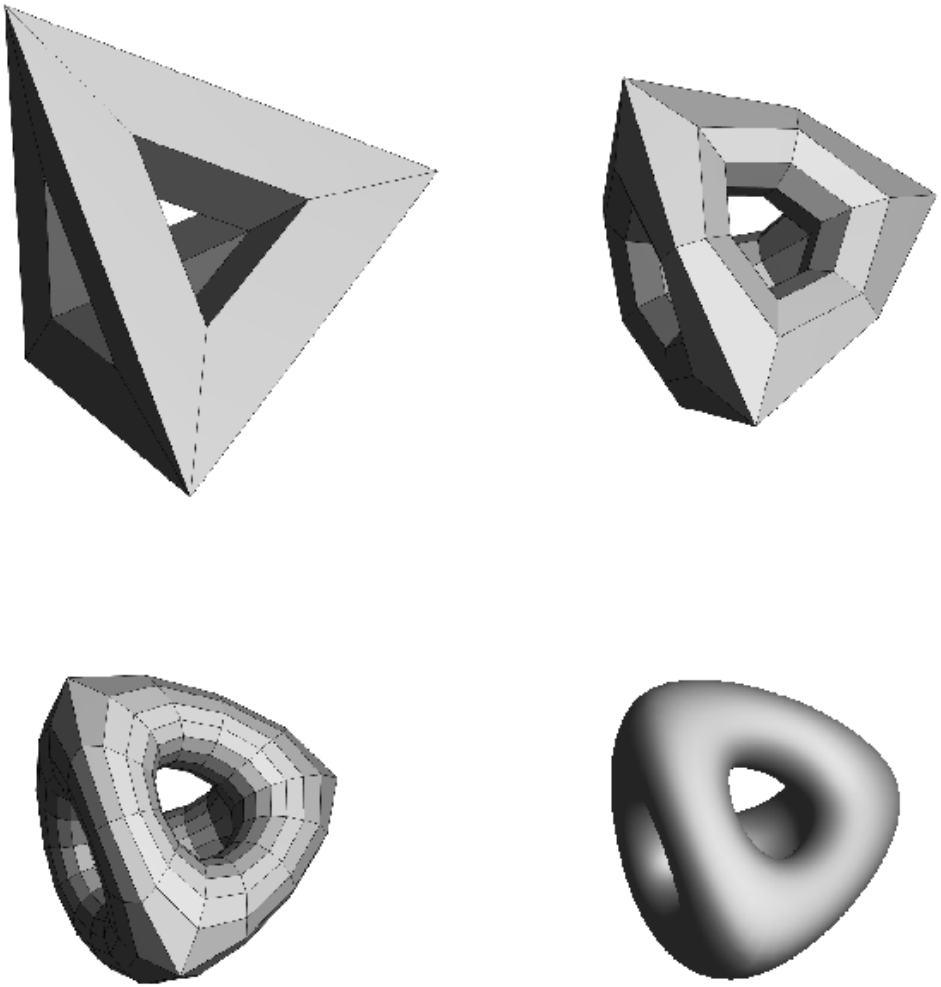
+

$(n-3) \times$ the current vertex position

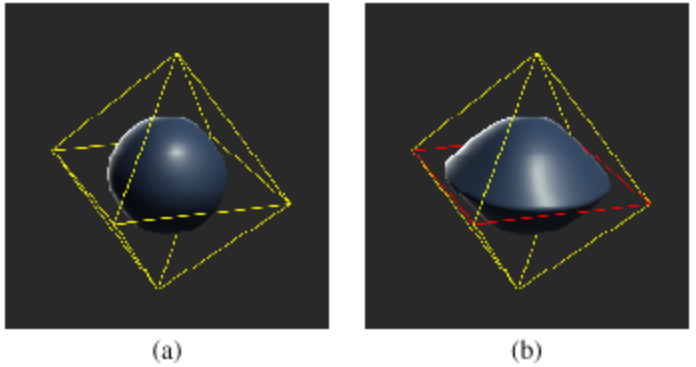
where n is the valence of the vertex (# of neighboring edges, also # of adjacent faces)

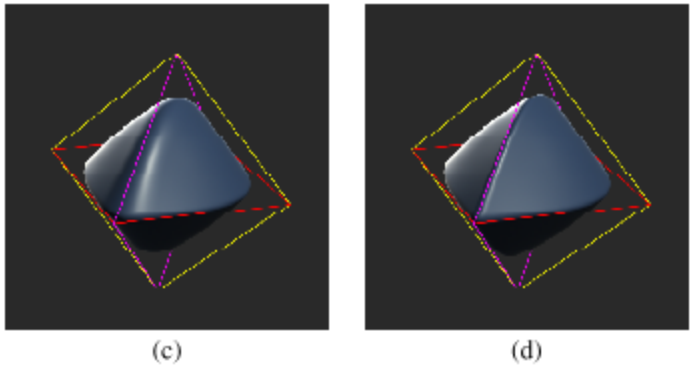


Example



Creases





$$f^{i+1}_{j} = \text{Centroid of polygon}$$

$$e^{i+1}_{j} = (v^i + e^i_j)/2$$

- Dart vertex (one sharp edge):

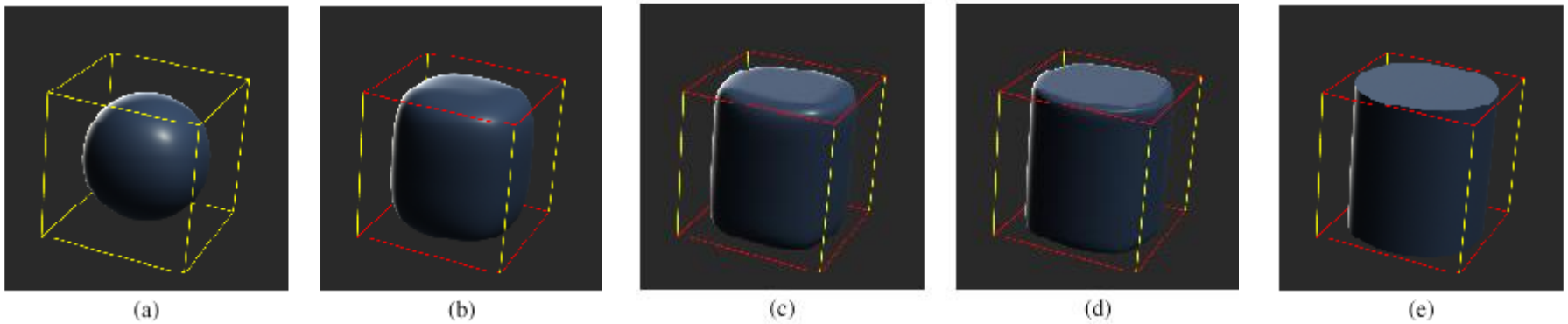
$$v^{i+1} = (n\text{-}2)/n\, v^i + 1/n^2\, \Sigma_j\, e^i_j + 1/n^2\, \Sigma_j\, f^{i+1}_{j}$$

- Crease vertex (two sharp edges):

$$v^{i+1} = (e^i_j + 6v^i + e^i_k)/8$$

- Corner vertex (three or more sharp edges)

$$v^{i+1} = v^i$$



Another Example

Success?