TIEA311
Tietokonegrafiikan perusteet
kevät 2019

(“Principles of Computer Graphics” – Spring 2019)

TIEA311 Tietokonegrafiikan perusteet – kevät 2019 (“Principles of Computer Graphics” – Spring 2019)

Adapted from: *Wojciech Matusik*, and *Frédo Durand*: 6.837 Computer Graphics. Fall 2012. Massachusetts Institute of Technology: MIT OpenCourseWare, https://ocw.mit.edu/.

Frontpage of the local course version, held during Spring 2019 at the Faculty of Information technology, University of Jyväskylä:
`http://users.jyu.fi/~nieminen/tgp19/`

TIEA311 - Today in Jyväskylä

Plan for today:

- Usual warm-up.
- Continue from yesterday
- Go through theory
- Remember to have a break!
- The teacher will try to remember and make use of the fact that we have groups of 2-3 students with pen and paper.

TIEA311 - Local plan for today

- Maybe some things I forgot to mention yesterday?
- Very brief recap of what went on previously.
- Then forward, with full speed!

6.837 Computer Graphics

Curve Properties & Conversion, Surface Representations

TIEA311

We now enter topics that require some more (simple) math.

We need to **know what we are looking at** when we see one of the following:

- Dot product $\vec{a} \cdot \vec{b}$ (or with coordinates: $\mathbf{a}^T\mathbf{b}$)
- Inner product notation $< a, b >$ when a and b are Euclidean space vectors.
- Norm $\|\vec{a}\|$
- Normalization $\frac{\vec{a}}{\|\vec{a}\|}$
- Cross product $\vec{a} \times \vec{b}$
- Determination of surface normal using $\vec{a} \times \vec{b}$
- The algorithm that makes orthonormal directions from any 2 linearly independent ones.
- Homogenization $[x/w, y/w, z/w, w/w]^T$
- The C++ interface for dot, cross, normalization, and homogenization in our “vecmath”.

Let us open the Game of Internet!

General Spline Formulation

$$Q(t) = \mathbf{GBT(t)} = \text{Geometry } \mathbf{G} \cdot \text{Spline Basis } \mathbf{B} \cdot \text{Power Basis } \mathbf{T(t)}$$

- Geometry: control points coordinates assembled into a matrix (P1, P2, …, Pn+1)
- Power basis: the monomials 1, *t*, *t*2, ...
- Cubic Bézier:

$$\mathrm{P(t)} = \begin{pmatrix} x(t) \\ y(t) \end{pmatrix} =$$

$$\begin{pmatrix} x_1 & x_2 & x_3 & x_4 \\ y_1 & y_2 & y_3 & y_4 \end{pmatrix} \begin{pmatrix} 1 & -3 & 3 & -1 \\ 0 & 3 & -6 & 3 \\ 0 & 0 & 3 & -3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ t \\ t^2 \\ t^3 \end{pmatrix}$$

Linear Transformations & Cubics

- What if we want to transform each point on the curve with a linear transformation **M**?

$$P'(t) = \mathbf{M}\left[\begin{pmatrix} P_{1,x} & P_{2,x} & P_{3,x} & P_{4,x} \\ P_{1,y} & P_{2,y} & P_{3,y} & P_{4,y} \end{pmatrix} \begin{pmatrix} 1 & -3 & 3 & -1 \\ 0 & 3 & -6 & 3 \\ 0 & 0 & 3 & -3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ t \\ t^2 \\ t^3 \end{pmatrix} \right]$$

Linear Transformations & Cubics

- What if we want to transform each point on the curve with a linear transformation **M**?
  - Because everything is linear, it is the same as transforming only the control points

$$P'(t) = \mathbf{M}\left(\begin{pmatrix} P_{1,x} & P_{2,x} & P_{3,x} & P_{4,x} \\ P_{1,y} & P_{2,y} & P_{3,y} & P_{4,y} \end{pmatrix} \begin{pmatrix} 1 & -3 & 3 & -1 \\ 0 & 3 & -6 & 3 \\ 0 & 0 & 3 & -3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ t \\ t^2 \\ t^3 \end{pmatrix} \right)$$

$$= \left(\mathbf{M} \begin{pmatrix} P_{1,x} & P_{2,x} & P_{3,x} & P_{4,x} \\ P_{1,y} & P_{2,y} & P_{3,y} & P_{4,y} \end{pmatrix} \right) \begin{pmatrix} 1 & -3 & 3 & -1 \\ 0 & 3 & -6 & 3 \\ 0 & 0 & 3 & -3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ t \\ t^2 \\ t^3 \end{pmatrix}$$

Affine Transformations

- Homogeneous coordinates also work
  - Means you can translate, rotate, shear, etc.
  - Note though that you need to normalize P' by $1/w'$

$$P'(t) = \mathbf{M}\left(\begin{pmatrix} P_{1,x} & P_{2,x} & P_{3,x} & P_{4,x} \\ P_{1,y} & P_{2,y} & P_{3,y} & P_{4,y} \\ 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & -3 & 3 & -1 \\ 0 & 3 & -6 & 3 \\ 0 & 0 & 3 & -3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ t \\ t^2 \\ t^3 \end{pmatrix} \right)$$

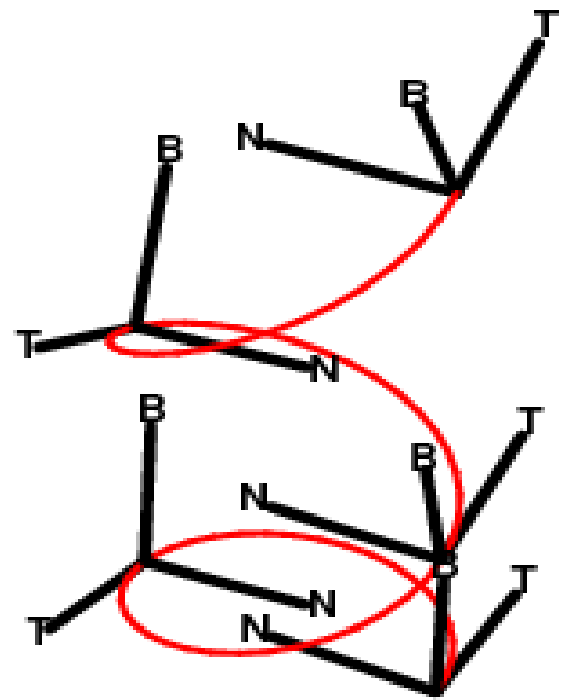
$$= \left(\mathbf{M} \begin{pmatrix} P_{1,x} & P_{2,x} & P_{3,x} & P_{4,x} \\ P_{1,y} & P_{2,y} & P_{3,y} & P_{4,y} \\ 1 & 1 & 1 & 1 \end{pmatrix} \right) \begin{pmatrix} 1 & -3 & 3 & -1 \\ 0 & 3 & -6 & 3 \\ 0 & 0 & 3 & -3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ t \\ t^2 \\ t^3 \end{pmatrix}$$

The Plan for Today

- Differential Properties of Curves & Continuity
- B-Splines
- Surfaces
  - Tensor Product Splines
  - Subdivision Surfaces
  - Procedural Surfaces
  - Other

Differential Properties of Curves

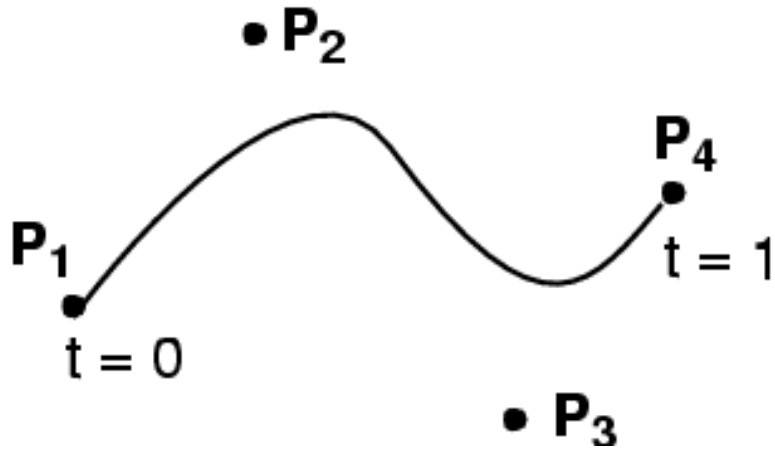
- Motivation
  - Compute normal for surfaces
  - Compute velocity for animation
  - Analyze smoothness



Velocity

- First derivative w.r.t. *t*
- Can you compute this for Bezier curves?

$$P(t) = (1-t)^3 P1 + 3t(1-t)^2 P2 + 3t^2(1-t) P3 + t^3 P4$$

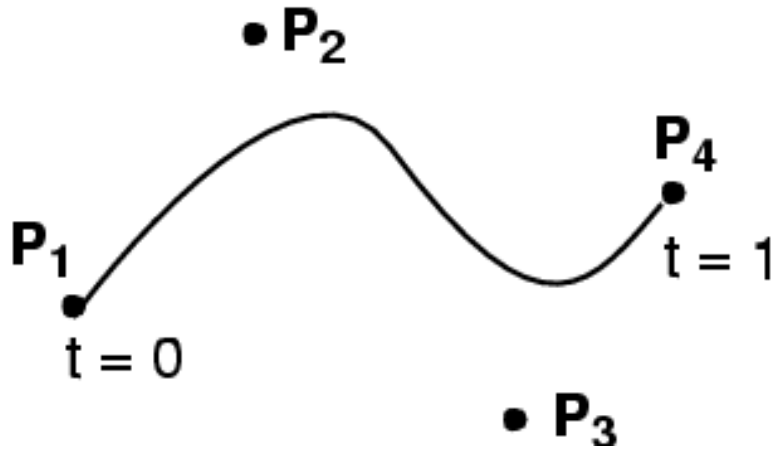


- You know how to differentiate polynomials...

Velocity

- First derivative w.r.t. *t*
- Can you compute this for Bezier curves?

$$P(t) = (1-t)^3 P1 + 3t(1-t)^2 P2 + 3t^2(1-t) P3 + t^3 P4$$



Sanity check: t=0; t=1

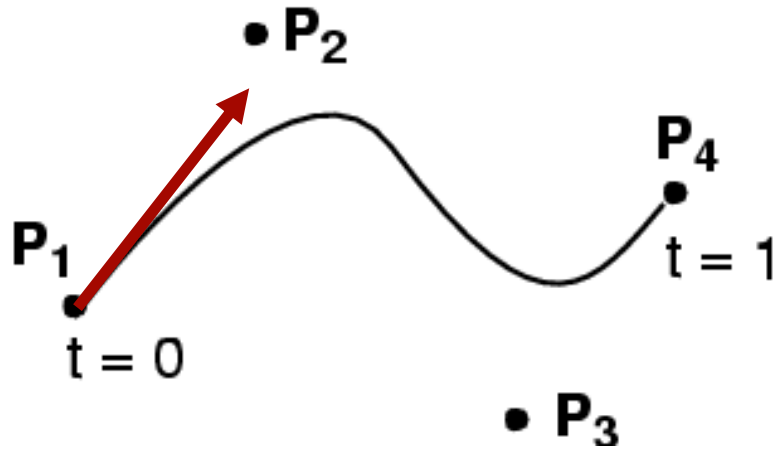
- **$P'(t) = -3(1-t)2 P1 + [3(1-t)\,2\,-6t(1-t)] P2 + [6t(1-t)-3t\,2] P3 + 3t\,2 P4$**

Linearity?

- Differentiation is a linear operation
  - $(f+g)' = f' + g'$
  - $(af)' = a f'$
- This means that the derivative of the basis is enough to know the derivative of any spline.
- Can be done with matrices
  - Trivial in monomial basis
  - But get lower-order polynomials

Tangent Vector

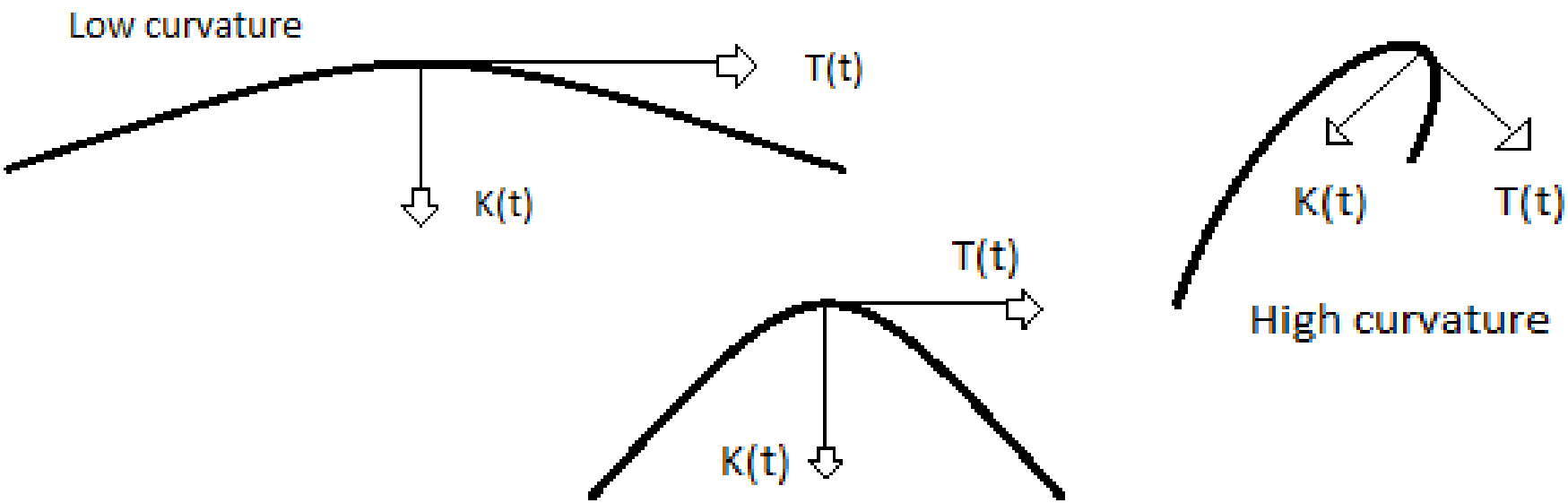
- The tangent to the curve P(t) can be defined as $T(t)=P'(t)/||P'(t)||$
  - normalized velocity, $||T(t)|| = 1$
- This provides us with one orientation for swept surfaces later



Courtesy of Seth Teller.

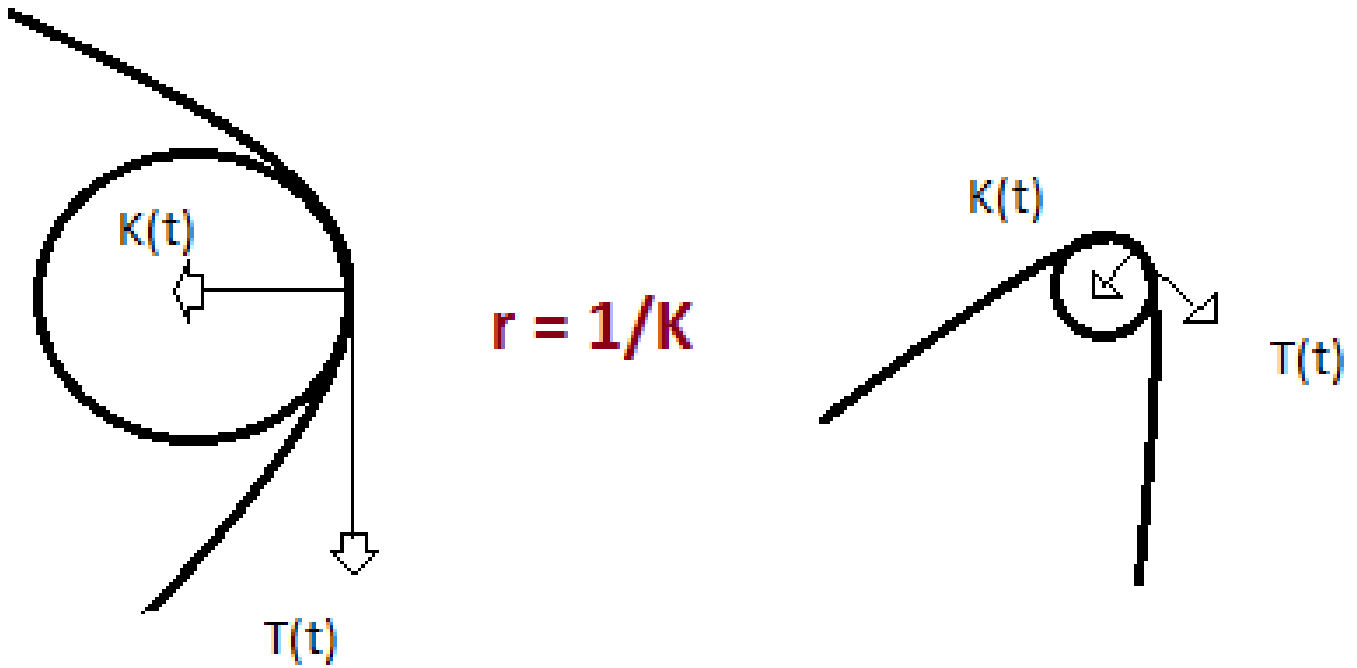
Curvature Vector

- Derivative of unit tangent
  - K(t)=T'(t)
  - Magnitude ||K(t)|| is constant for a circle
  - Zero for a straight line
- Always orthogonal to tangent, ie. $K \cdot T = 0$



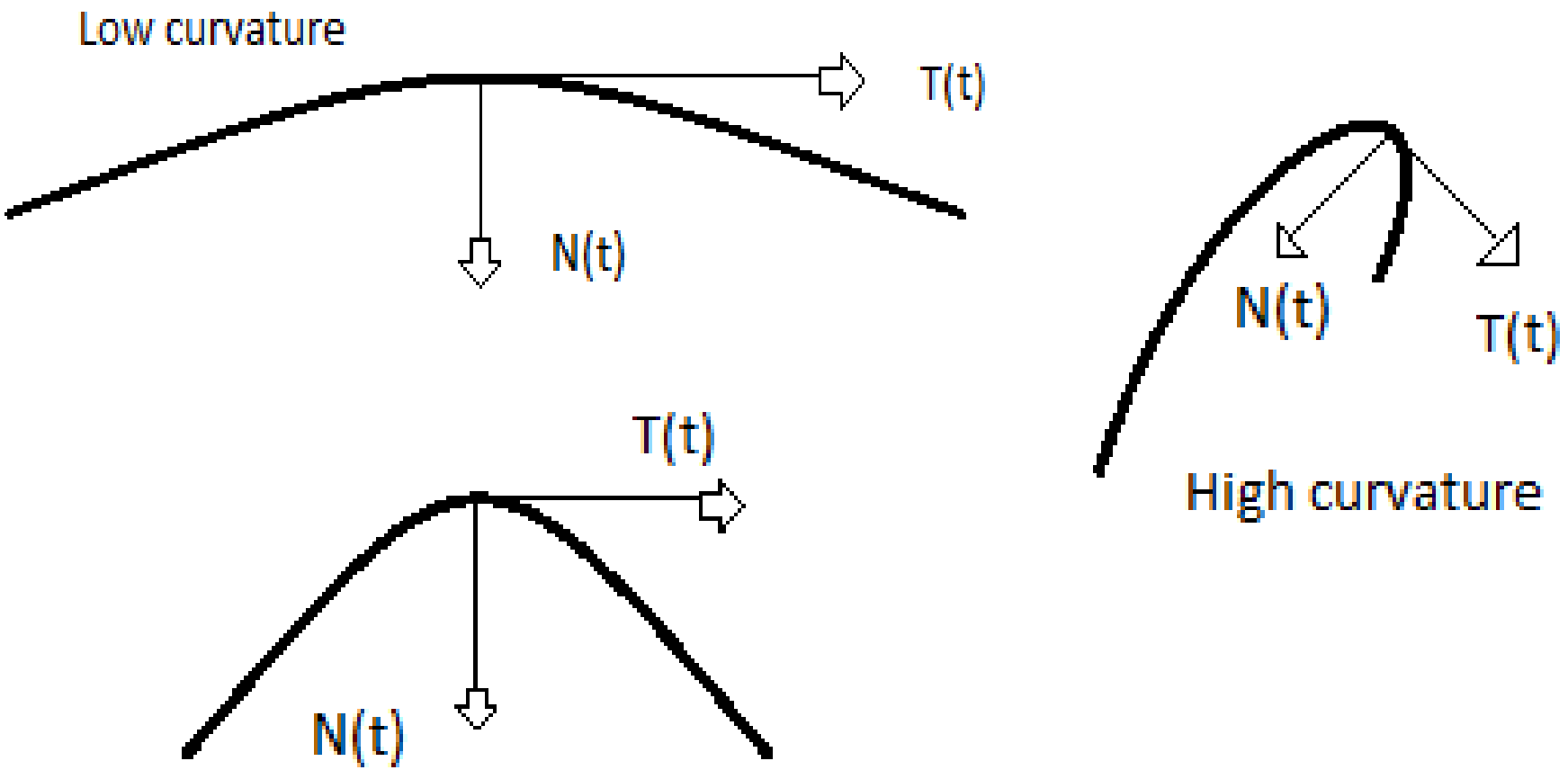
Geometric Interpretation

- K is zero for a line, constant for circle
  - What constant? 1/r
- 1/||K(t)|| is the radius of the circle that touches P(t) at *t* and has the same curvature as the curve



Curve Normal

- Normalized curvature: T’(t)/||T’(t)||



TIEA311 - Today in Jyväskylä

The time allotted for this week's graphics lectures is now over.

Next lecture happens in 6 days and 4 hours.

The teacher will now tell his view about what **could be useful activities** for you during that time period.

$\rightarrow$ see lecture video.

Make notes, if you have to.

Even if he forgets to say it, **remember to rest, too**!

TIEA311

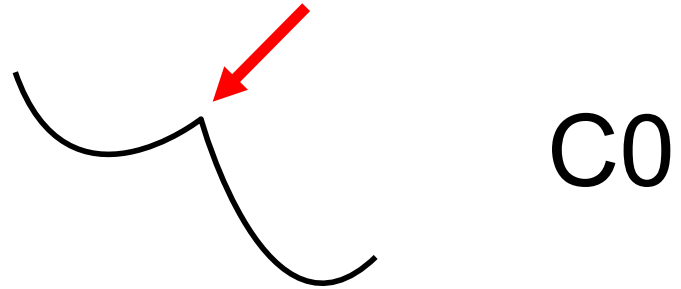
The following slides were not shown on the lecture (yet).

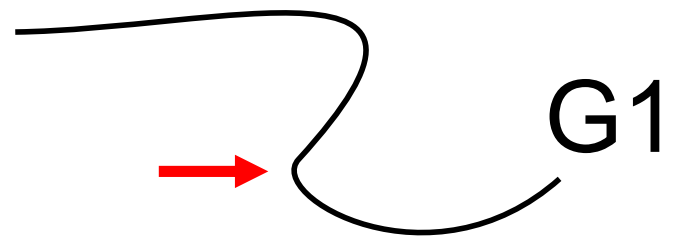
They are a **preview** of what we will talk about next, very soon.

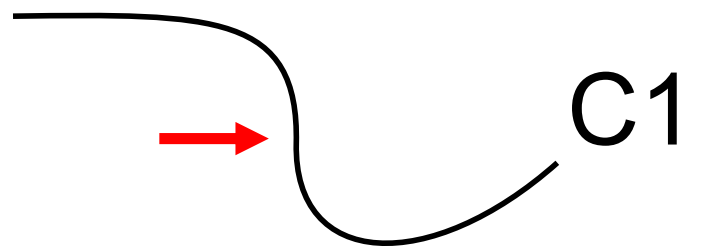
Questions based on your preview will be **much appreciated** when we meet on the next lecture!

Orders of Continuity

- C0 = continuous
  - The seam can be a sharp kink
- G1 = geometric continuity
  - Tangents **point to the same direction** at the seam
- C1 = parametric continuity
  - Tangents **are the same** at the seam, implies G1
- C2 = curvature continuity
  - Tangents and their derivatives are the same

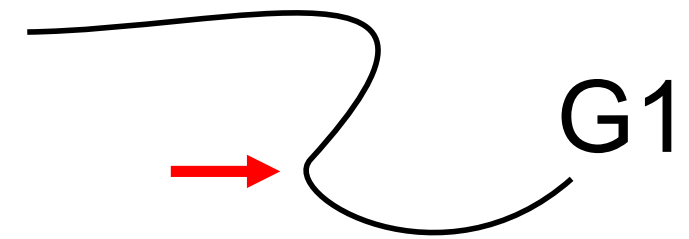


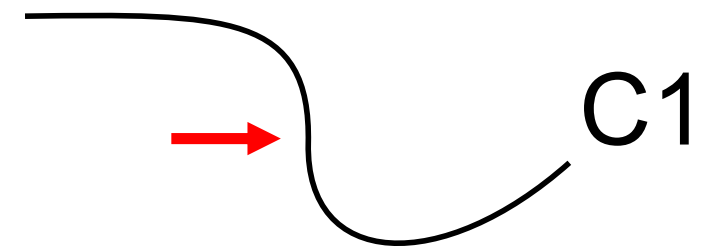




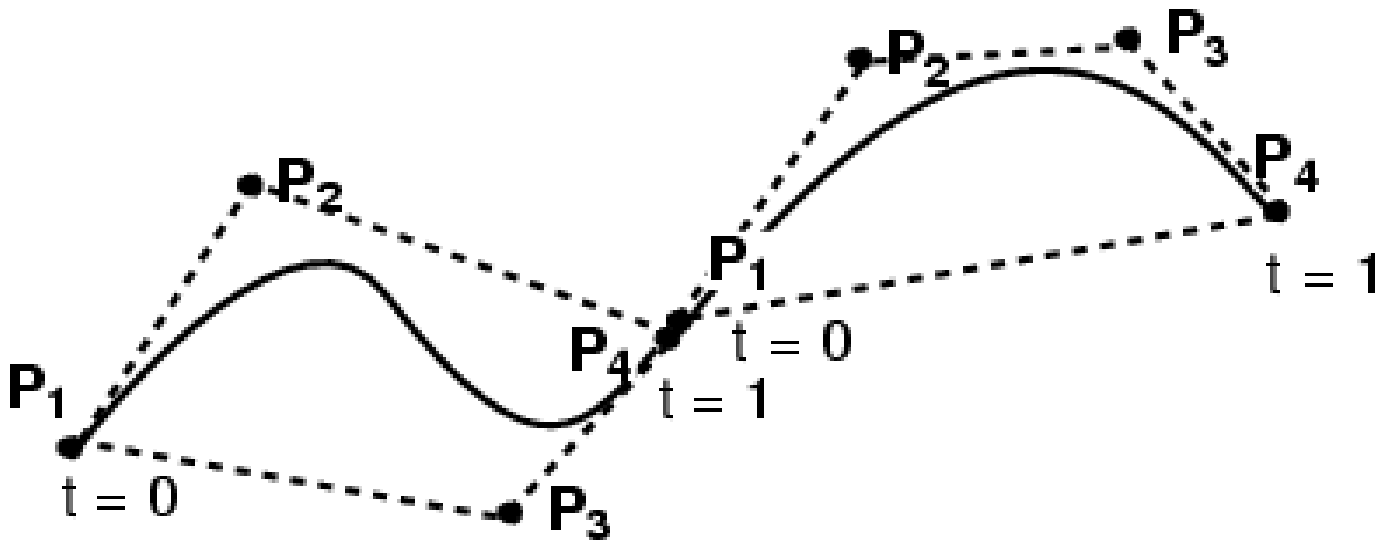
Orders of Continuity

- G1 = geometric continuity
  - Tangents **point to the same direction** at the seam
  - good enough for modeling
- C1 = parametric continuity
  - Tangents **are the same** at the seam, implies G1
  - often necessary for animation



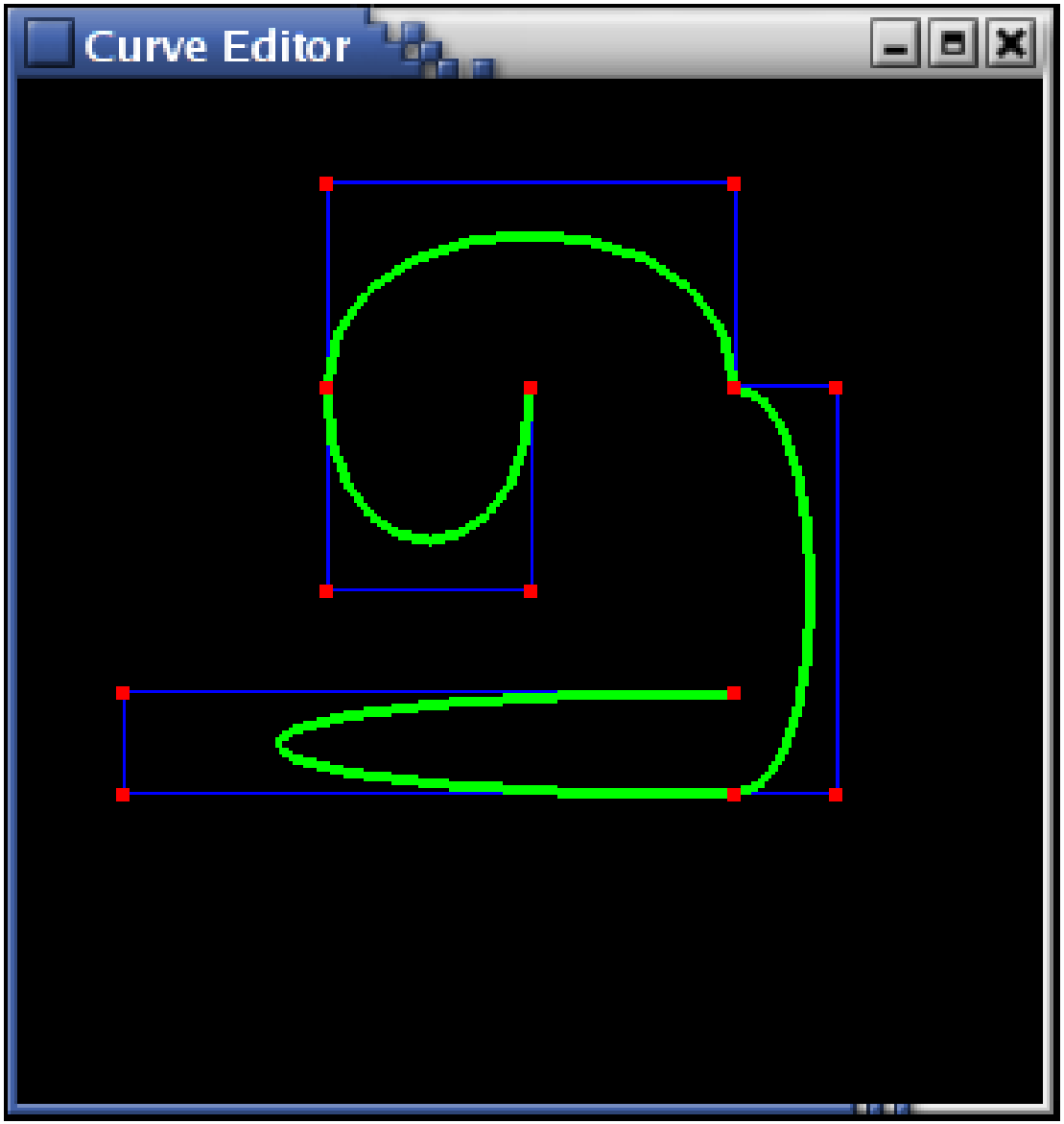


Connecting Cubic Bézier Curves



- How can we guarantee C0 continuity?
- How can we guarantee G1 continuity?
- How can we guarantee C1 continuity?
- C2 and above gets difficult

Connecting Cubic Bézier Curves



- Where is this curve
  - C0 continuous?
  - G1 continuous?
  - C1 continuous?
- What's the relationship between:
  - the # of control points, and the # of cubic Bézier subcurves?

Questions?

TIEA311 - Local plan for today

Oh, wait! Let us get one thing out of the way. . .

The teacher will now communicate to you at least three things simultaneously:

- How to pass our T2 with points 1/5, towards course grade 1/5.
- Some philosophy behind the definition of grade 1/5.
- Practically, a spoken-out definition of grade 1/5 on this course.
- One possible option of some steps that would need to be taken first in order to progress towards the higher grades, or in general “being a pro” in IT stuff.

$\rightarrow$ Live coding and thinking aloud. See lecture video.

TIEA311

OK.

That went smoothly.

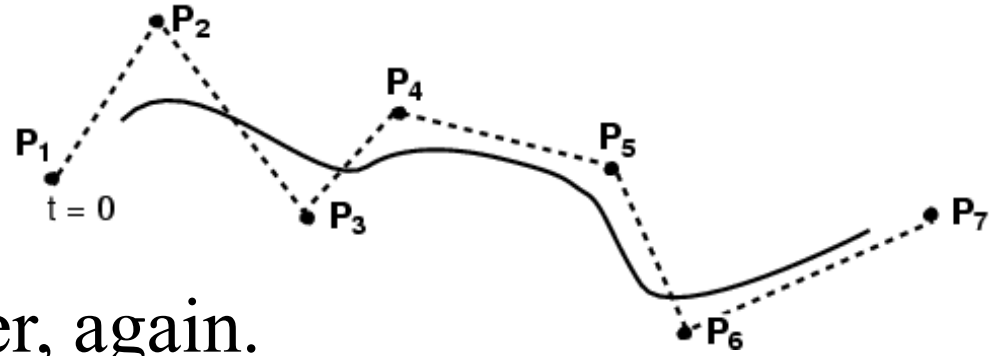
Like a smooth curve!

With score 1/5 secured, we can feel warm and happy, and stop worrying about “passing or not passing” T2 on TIEA311! That question about passing is foul in all contexts, anyway! Ugh!

This **enables us** to relax and spend some nice **focused learning time** in order to **understand more** about all this, and move towards not (just) copy-pasting, but at least copy-paste-modifying-with-some-idea-why. . .

Cubic B-Splines

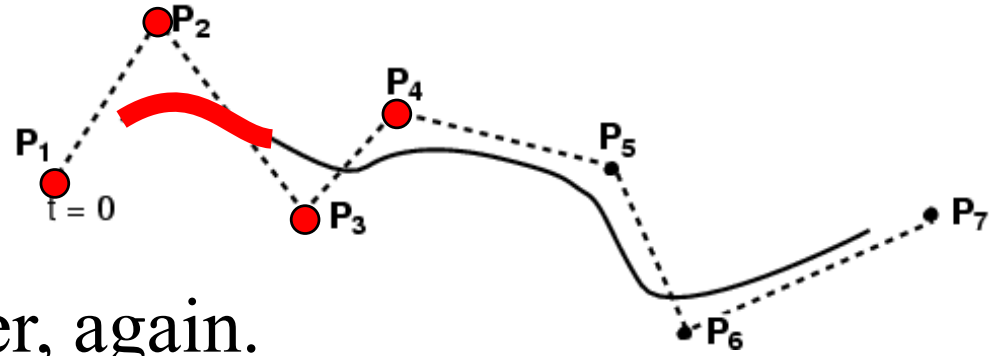
- ≥ 4 control points
- Locally cubic
  - Cubics chained together, again.



Courtesy of Seth Teller.

Cubic B-Splines

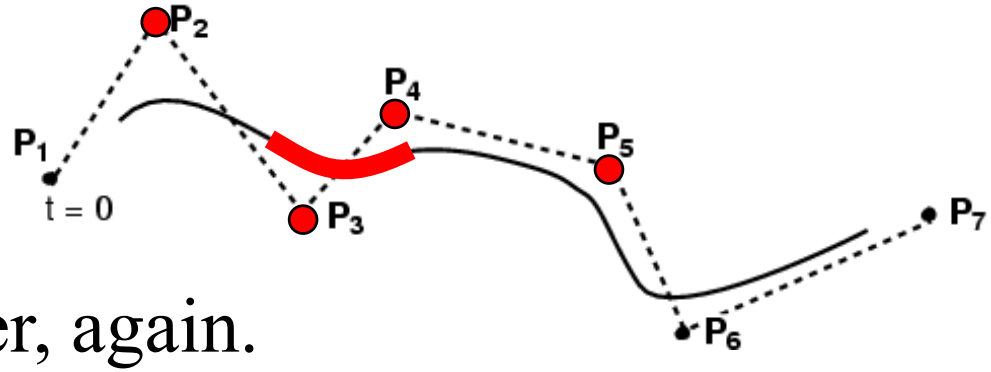
- ≥ 4 control points
- Locally cubic
  - Cubics chained together, again.



Courtesy of Seth Teller.

Cubic B-Splines

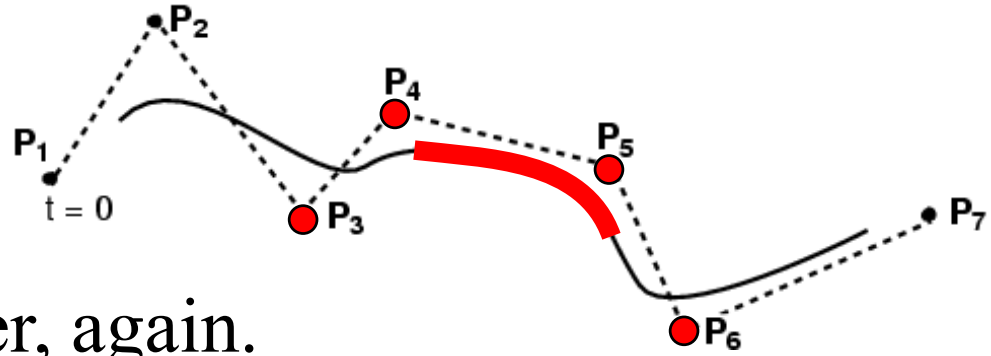
- ≥ 4 control points
- Locally cubic
  - Cubics chained together, again.



Courtesy of Seth Teller.

Cubic B-Splines

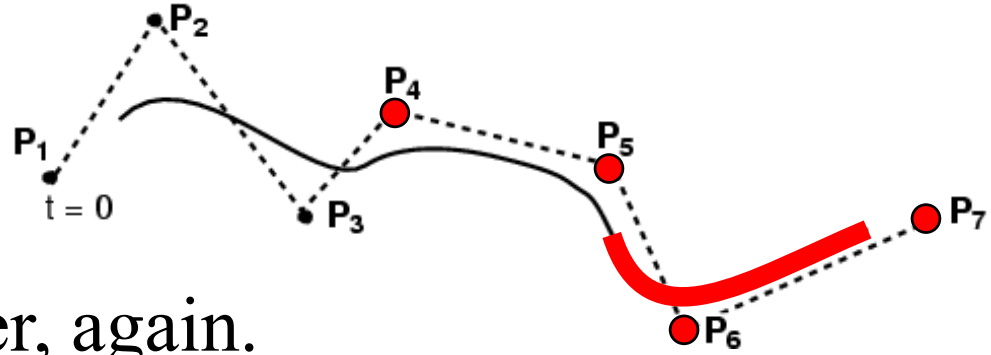
- ≥ 4 control points
- Locally cubic
  - Cubics chained together, again.



Courtesy of Seth Teller.

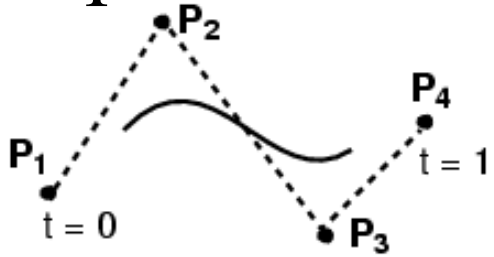
Cubic B-Splines

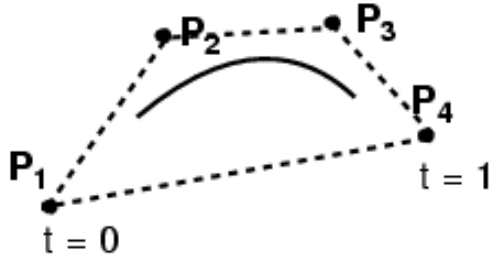
- ≥ 4 control points
- Locally cubic
  - Cubics chained together, again.

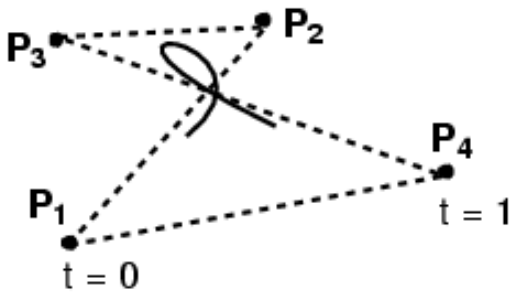


Courtesy of Seth Teller.

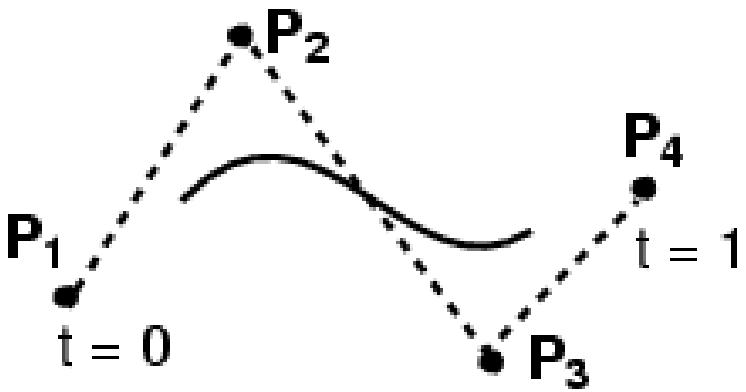
- Curve is not constrained to pass through any control points



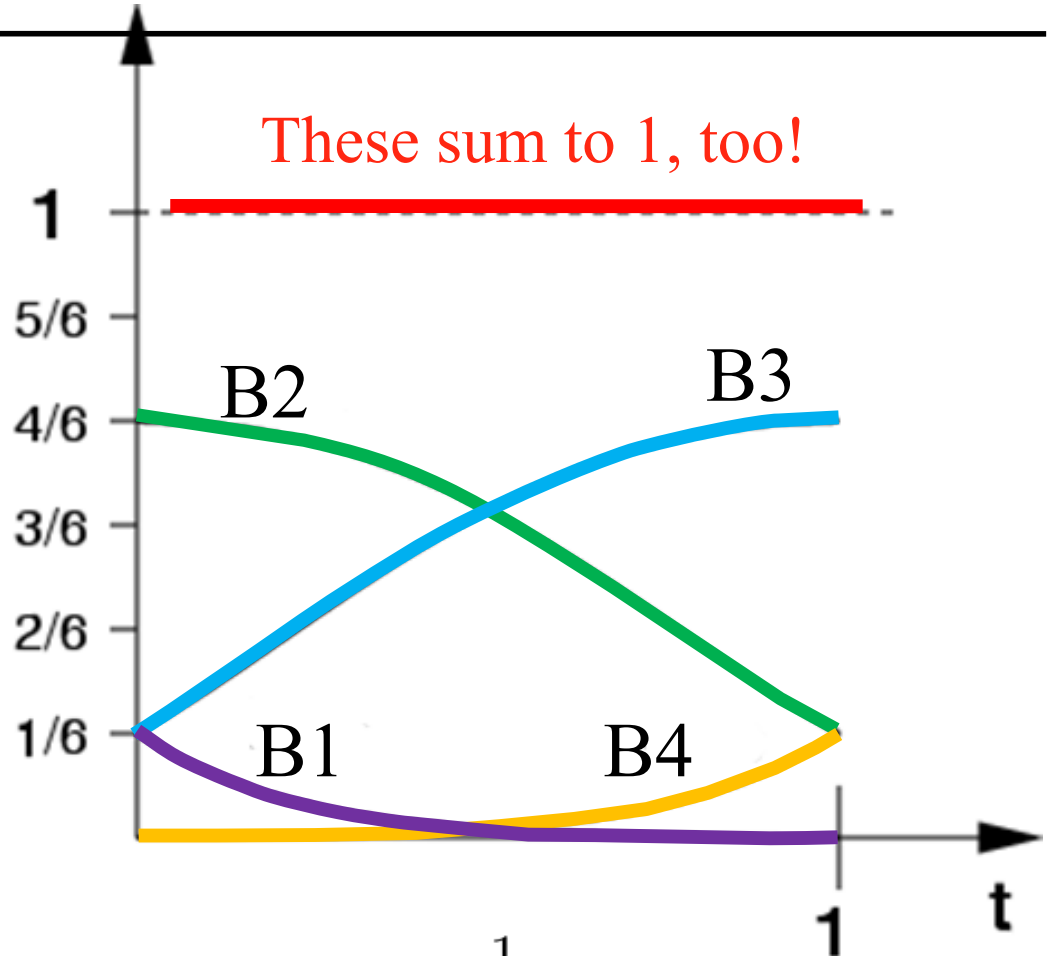




Cubic B-Splines: Basis



A B-Spline curve is also bounded by the convex hull of its control points.



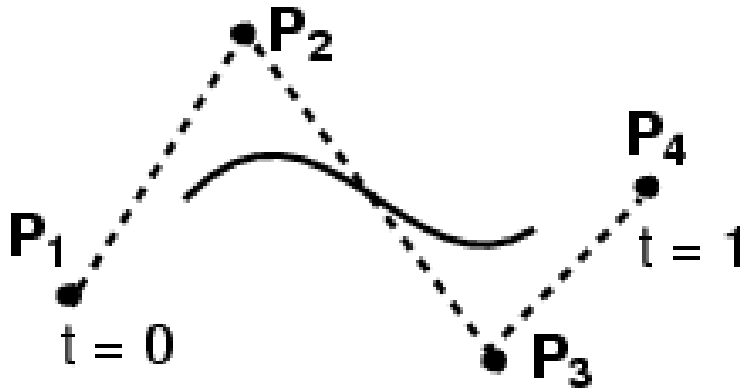
$$B_1(t) = \frac{1}{6}(1-t)^3$$

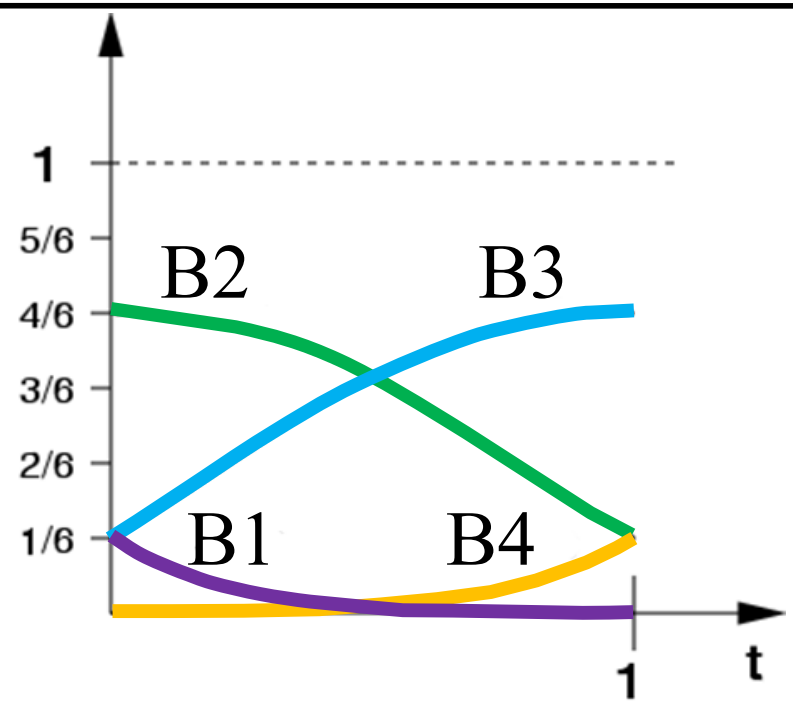
$$B_2(t) = \frac{1}{6}(3t^3 - 6t^2 + 4)$$

$$B_3(t) = \frac{1}{6}(-3t^3 + 3t^2 + 3t + 1)$$

$$B_4(t) = \frac{1}{6}t^3$$

Cubic B-Splines: Basis





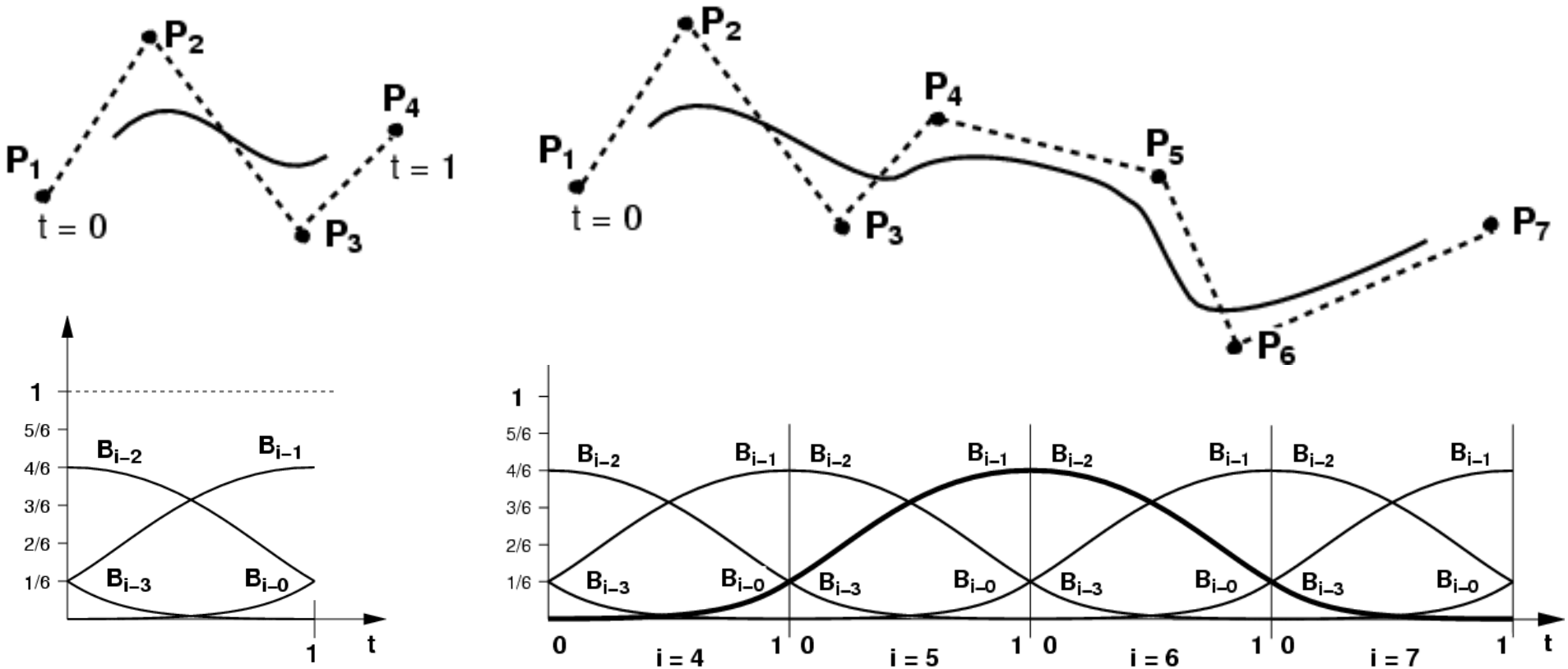
$$Q(t) = \frac{(1-t)^3}{6}P_1 + \frac{3t^3 - 6t^2 + 4}{6}P_2 + \frac{-3t^3 + 3t^2 + 3t + 1}{6}P_3 + \frac{t^3}{6}P_4$$

$$Q(t) = \mathbf{GBT(t)}$$

$$B_{B-Spline} = \frac{1}{6}\begin{pmatrix} 1 & -3 & 3 & -1 \\ 4 & 0 & -6 & 3 \\ 1 & 3 & 3 & -3 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

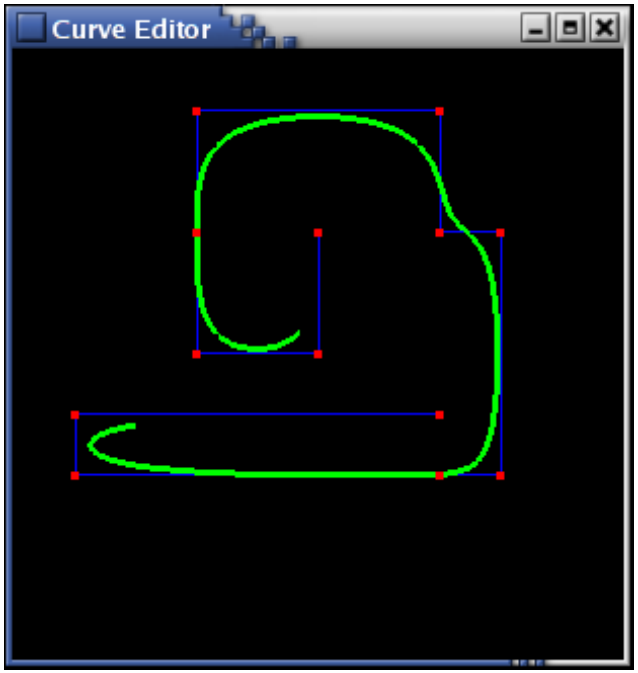
Cubic B-Splines

- Local control (windowing)
- Automatically C2, and no need to match tangents!

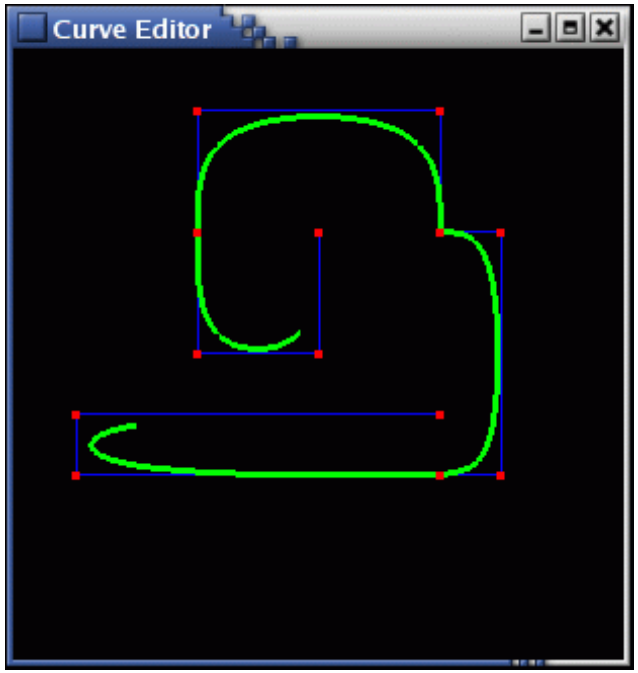


Courtesy of Seth Teller. Used with permission.

B-Spline Curve Control Points

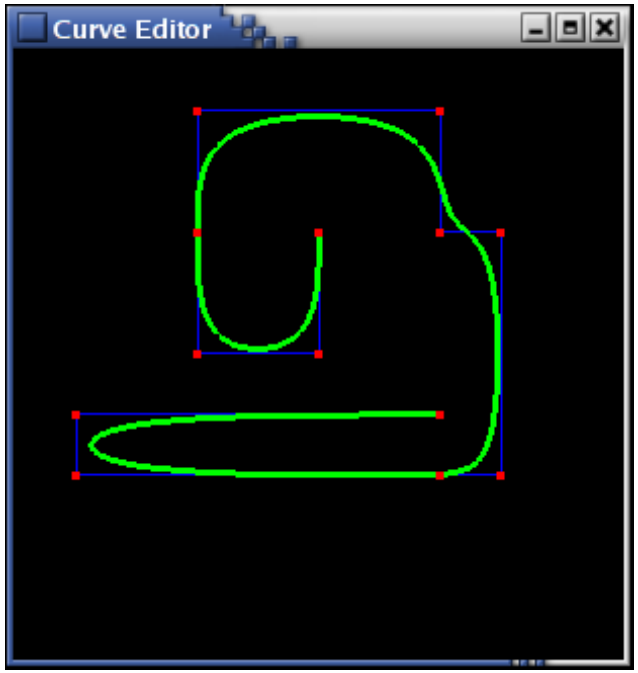


Default B-Spline



B-Spline with derivative discontinuity

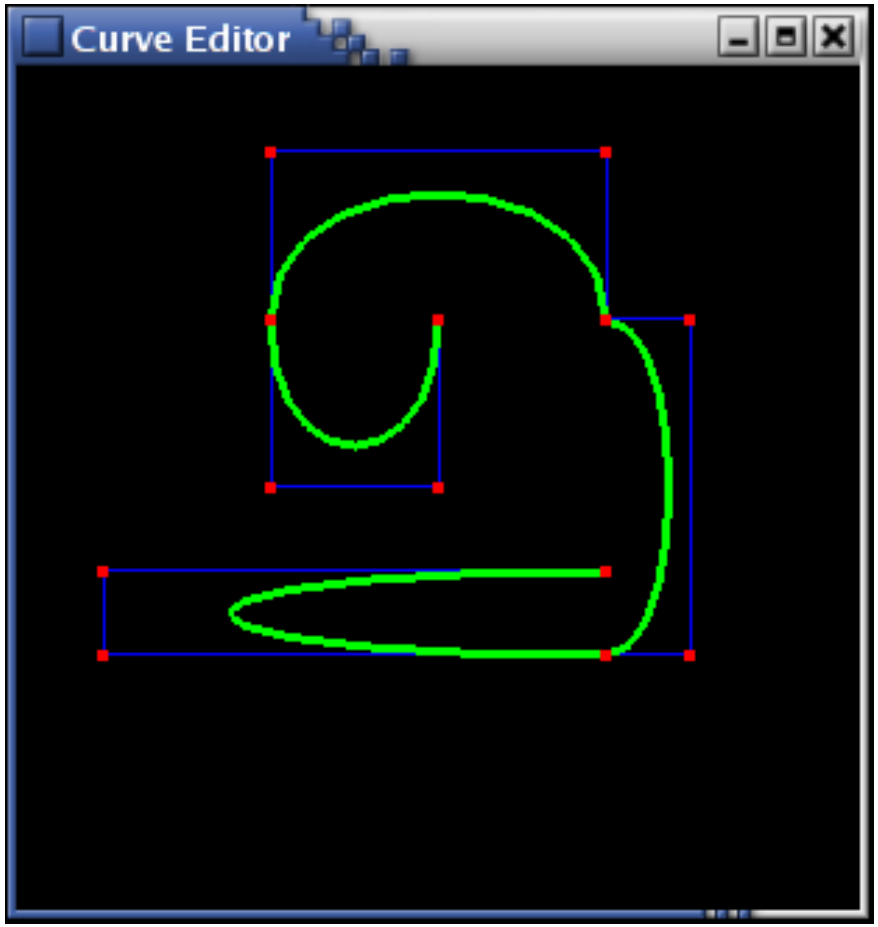
Repeat interior control point



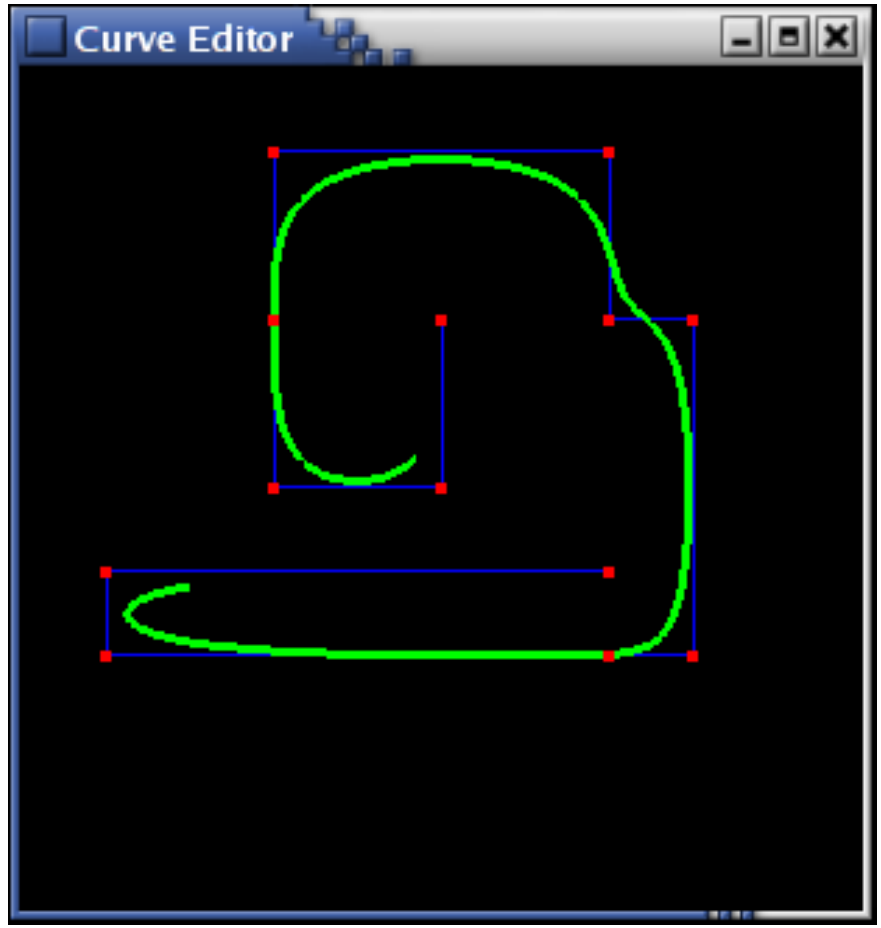
B-Spline which passes through end points

Repeat end points

Bézier ≠ B-Spline



Bézier



B-Spline

But both are cubics, so one can be converted into the other!

Converting between Bézier & BSpline

$Q(t) = \mathbf{GBT(t)}$ = Geometry $\mathbf{G}$ · Spline Basis $\mathbf{B}$ · Power Basis $\mathbf{T(t)}$

- Simple with the basis matrices!
  - Note that this only works for a single segment of 4 control points

$$B_{Bezier} = \begin{pmatrix} 1 & -3 & 3 & -1 \\ 0 & 3 & -6 & 3 \\ 0 & 0 & 3 & -3 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- P(t) = **G B**1 **T**(t) = **G B**1 **(B2-1B2)** **T**(t)= (**G B**1 **B**2-1) **B**2 **T**(t)

$$B_{B-Spline} = \frac{1}{6}\begin{pmatrix} 1 & -3 & 3 & -1 \\ 4 & 0 & -6 & 3 \\ 1 & 3 & 3 & -3 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- **G B**1 **B**2-1 are the control points for the segment in new basis.

In the previous slide, the minor inconvenience of misprinted subscripts and superscripts is especially harmful. The equation should read as:

$$\begin{aligned} P(t) &= GB_1T(t) \\ &= GB_1(B_2^{-1}B_2)T(t) \\ &= (GB_1B_2^{-1})B_2T(t) \end{aligned}$$

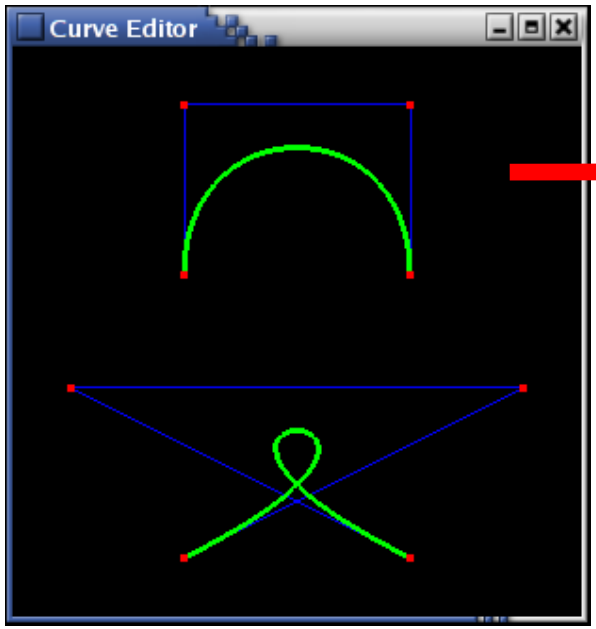
Then, we end up with $(GB_1B_2^{-1})$ as new control points.

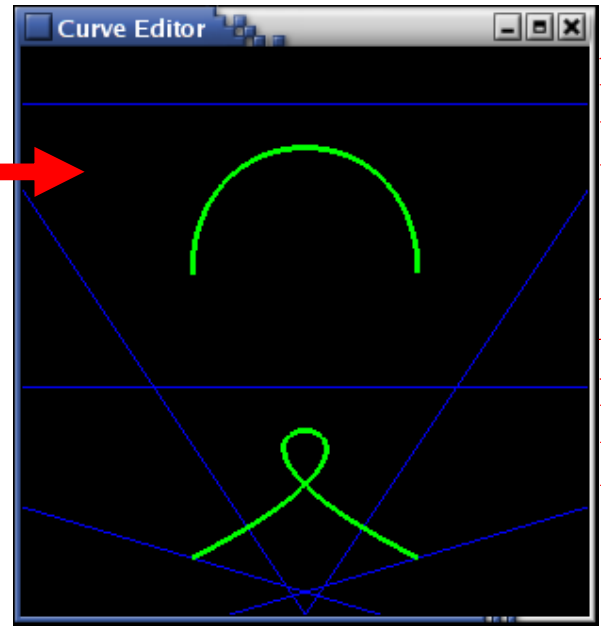
“Unfortunately”, you will need to do similar re-interpretation of many of the equations in the OpenCourseware slides to fully understand them.

“Fortunately”, **doing this will actually make you understand each equation better** :). Pen and paper are your friends!

Converting between Bézier & B-Spline

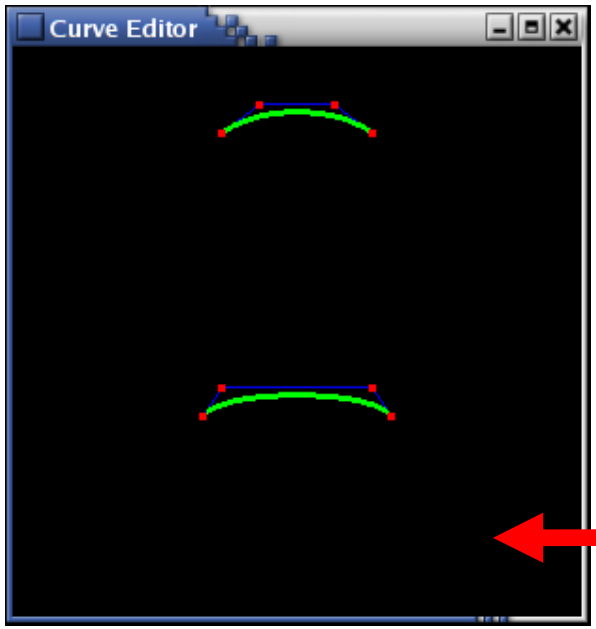
original control points as Bézier

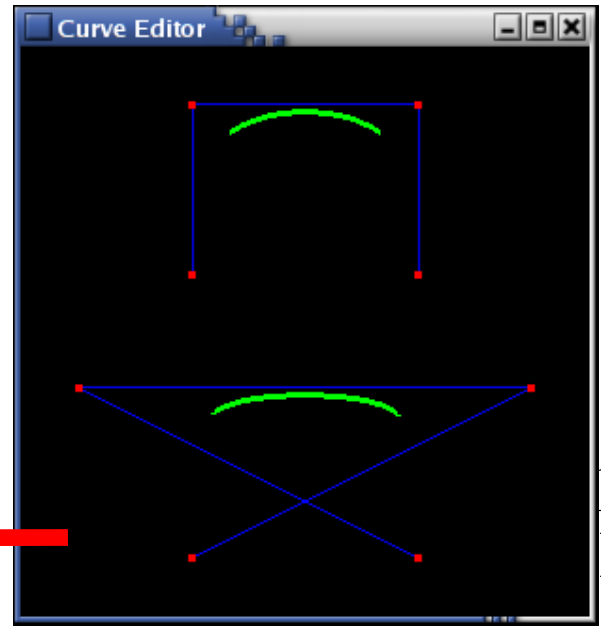




new BSpline control points to match Bézier

new Bézier control points to match B-Spline





original control points as B-Spline

NURBS (Generalized B-Splines)

- Rational cubics
  - Use homogeneous coordinates, just add *w* !
    - Provides an extra weight parameter to control points

- NURBS: Non-Uniform Rational B-Spline
  - **non-uniform** = different spacing between the blending functions, a.k.a. "knots"
  - **rational** = ratio of cubic polynomials (instead of just cubic)
    - implemented by adding the homogeneous coordinate *w* into the control points.

Demo

Questions?

Representing Surfaces

- Triangle meshes
  - Surface analogue of polylines, this is what GPUs draw
- **Tensor Product Splines**
  - Surface analogue of spline curves
- **Subdivision surfaces**
- **Implicit surfaces, e.g.** f(x,y,z)=0
- **Procedural**
  - e.g. surfaces of revolution, generalized cylinder
- From volume data (medical images, etc.)