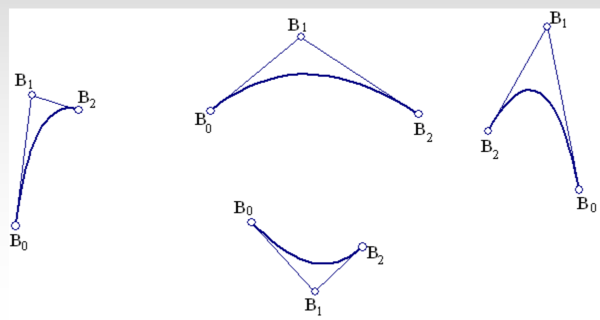
CPSC 424
Affine Combinations, de Casteljau Algorithm



Syllabus

Curves in 2D and 3D

- Implicit vs. Explicit vs. Parametric curves
- **Bézier curves, de Casteljau algorithm**
- Continuity
- B-Splines
- Subdivision Curves

Properties of Curves and Surfaces

Surfaces

Bézier Curves

Definition:

- Bézier curve is a polynomial curve that uses **Bernstein polynomials** as basis

$$F(t) = \sum_{i=0}^{m} \mathbf{b}_i B_i^m(t) \qquad \boldsymbol{t} \in [\mathbf{0}, \mathbf{1}]$$

- b_i are called control points of Bézier curve

- Control polygon obtained by connecting control points with line segments

Bernstein Polynomials

$$B_i^m(t) := \binom{m}{i} t^i (1-t)^{m-i}; i = 0..m; t \in [0,1],$$

$$\binom{m}{i} = \frac{m!}{(m-i)!\,i!}$$

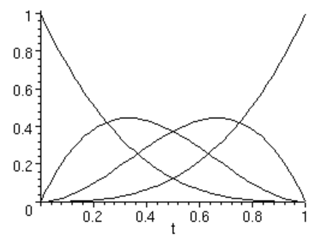
Bézier Curves - Polynomials

Bernstein Polynomials

$$B_i^m(t) := \binom{m}{i} t^i (1-t)^{m-i}; i = 0..m; t \in [0,1]$$

- Graph for degree m=3:

```
> plot({seq(binomial(3,i)*t^i*(1-t)^(3-i),i=0..3)},
  t=0..1,color=black);
```



Bernstein Polynomials

$$B_i^m(t) := \binom{m}{i} t^i (1-t)^{m-i}; i = 0..m; t \in [0,1]$$

Properties:

- $B_i^m(t)$ is a polynomial of degree m
- $B_i^m(t) \geq 0 \text{ for } t \in [0,1]; B_0^m(0) = 1; B_i^m(0) = 0 \text{ for } i \neq 0$
- $B_i^m(t) = B_{m-i}^m(1-t)$

Bernstein Polynomials

$$B_i^m(t) := \binom{m}{i} t^i (1-t)^{m-i}; i = 0..m; t \in [0,1]$$

Properties:

- $B_i^m(t)$ has exactly one maximum in the interval 0..1. It is at t=i/m (proof: compute derivative…)
- W/o proof: all (m+1) functions B_i^m are linearly independent
  - *Thus they form a basis for all polynomials of degree ≤ m*

Bernstein Polynomials

More properties

- $\sum_{i=0}^{m} B_i^m(t) = (t + (1-t))^m \equiv 1$
  - *(proof: apply Binomial Theorem to definition)*
- $B_i^m(t) = t \cdot B_{i-1}^{m-1}(t) + (1-t) \cdot B_i^{m-1}(t)$
  - *(proof on board)*
- Important (later) for fast evaluation algorithm of Bézier curves (de Casteljau algorithm)

Bézier Curves

Definition:

- A Bézier curve is a polynomial curve that uses the Bernstein polynomials as a basis

$$F(t) = \sum_{i=0}^{m} \mathbf{b}_i B_i^m(t)$$

- b_i are called control points of Bézier curve
- Control polygon obtained by connecting control points with line segments

- Examples: https://www.ibiblio.org/e-notes/Splines/bezier.html

Clicker Question

For a Bezier curve with 4 control points positioned along a horizontal line. If I move the first point up, will the curve between two last points

A. Move up

B. Move down

C. Stay where it was

D. No idea

Bézier Curves

Definition:

- A Bézier curve is a polynomial curve that uses the Bernstein polynomials as a basis

$$F(t) = \sum_{i=0}^{m} \mathbf{b}_i B_i^m(t)$$

Advantage of Bézier curves:

- Control points & control polygon have clear geometric meaning and are intuitive to use

Properties of Bézier Curves (Pierre Bézier, Renault, about 1970)

Easy to see:

- Endpoints b_0 and b_m of control polygon interpolated & corresponding parameter values are t=0 and t=1

Less easy: Curve is affine invariant

- Affine invariant: invariant under linear transformations + translation
- Transforming control points = Transforming curve

Convex Combination

Convex Combination

$$\sum_i \lambda_i \mathbf{x}_i, \quad \sum_i \lambda_i = 1, \quad \lambda_i \geq 0$$

Special case

$$tx_0 + (1-t)x_1$$

Convex Combination

Convex Combination

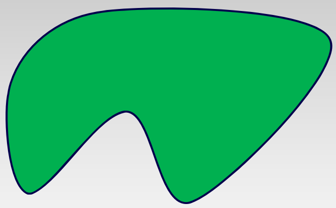
$$\sum_i \lambda_i \mathbf{x}_i, \quad \sum_i \lambda_i = 1, \quad \lambda_i \geq 0$$

Def: Convex Hull

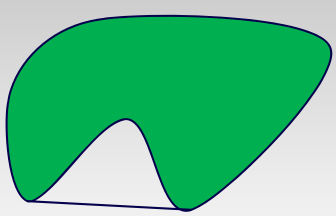
- The convex hull $[x_i]$ of a set of points x_i is the set of points that can be expressed as convex combinations of the x_i:

$$[x_i] := \left\{ \sum_i \lambda_i \mathbf{x}_i \mid \sum_i \lambda_i = 1; \quad \lambda_i \geq 0 \right\}$$

Convex Hulls



Convex Hulls



Clicker

Which of the following are convex: sphere, torus, cube, cow?

A. All

B. None

C. Sphere and torus

D. Sphere and cube

E. Cow

Back to Bézier Curves

$$F(t) = \sum_{i=0}^{m} \mathbf{b}_i B_i^m(t)$$

Recall:

- Bernstein polynomials have values between 0 and 1 for t∈[0,1], and $\sum_{i=0}^{m} B_i^m(t) \equiv 1$
  - *Therefore: every point on Bézier curve is convex combination of control points*
  - *Therefore: Bézier curve lies completely within convex hull of control points*

Properties of Bézier Curves (Pierre Bézier, Renault, about 1970)

Without proof for the moment (will be easier to show later):

- Bézier curve is tangential to control polygon at endpoints
- There is a fast, recursive evaluation algorithm

Lagrange v.s. Bezier

Lagrange

Basis: *Lagrange Polynomials*

$$L_i^m(t) := \prod_{j=0..m, j \neq i} \frac{t - t_j}{t_i - t_j}; i = 0...m$$

Spline:

$$F(t) = \sum_{i=0}^{m} \mathbf{b}_i L_i^m(t_j)$$

Interpolates control points

Hard to control ("wiggles")

Bezier

Basis: *Bernstein Polynomials*

$$B_i^m(t) := \binom{m}{i} t^i (1-t)^{m-i}; i = 0..m; t \in [0,1]$$

Spline:

$$F(t) = \sum_{i=0}^{m} \mathbf{b}_i B_i^m(t)$$

Approximates control points

Easy to control (more or less)

De Casteljau Algorithm

Recall:

- Recursive formula for Bernstein polynomials (what about i=0 or i=m?):

$$B_i^m(t) = t \cdot B_{i-1}^{m-1}(t) + (1-t) \cdot B_i^{m-1}(t)$$

Plug into Bézier curve definition (on board):

$$F(t) = \sum_{i=0}^{m} \mathbf{b}_i \left(t \cdot B_{i-1}^{m-1}(t) + (1-t) \cdot B_i^{m-1}(t) \right)$$

$$= t \cdot \sum_{i=1}^{m} \mathbf{b}_i B_{i-1}^{m-1}(t) + (1-t) \cdot \sum_{i=0}^{m-1} \mathbf{b}_i B_i^{m-1}(t)$$

De Casteljau Algorithm

Consequence:

- Every point $F(t_0)$ on a Bézier curve of degree m is the convex combination of two points $G(t_0)$ and $H(t_0)$ that lie on Bézier curves of degree m-1.
- The control points of $G(t)$ are the first m control points of $F(t)$
- The control points of $H(t)$ are the last m control points of $F(t)$

De Casteljau Algorithm

Recursion:

- Every point on a Bézier curve can be generated through successive convex combinations of the degree 0 Bézier curves
- Degree 0 Bézier curves are the control points!

$$F(t) = \sum_{i=0}^{0} \mathbf{b}_i B_i^0(t) = \mathbf{b}_i \cdot 1 \equiv \mathbf{b}_i$$

De Casteljau Algorithm

After working out the math we get:

$$F(t) = \mathbf{b}_0^m(t) \;\; \text{; where}$$

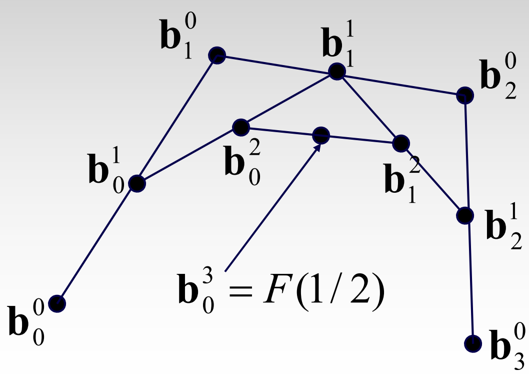
$$\mathbf{b}_i^0(t) := \mathbf{b}_i; \quad i = 0 \ldots m$$

$$\mathbf{b}_i^l(t) := (1-t) \cdot \mathbf{b}_i^{l-1}(t) + t \cdot \mathbf{b}_{i+1}^{l-1}(t)$$

De Casteljau Algorithm

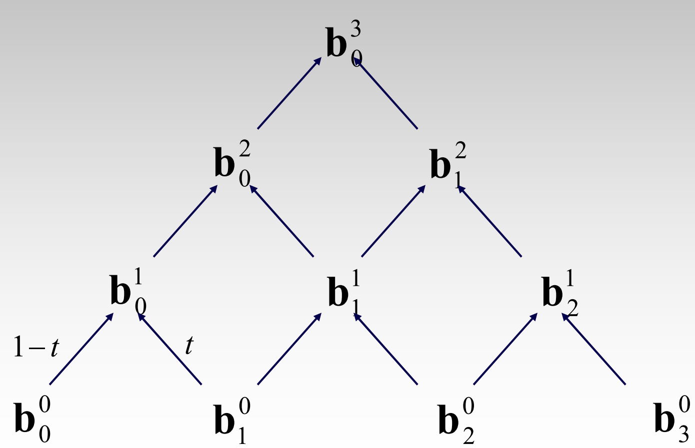
Graphical Interpretation:

- Determine point *F(1/2)* for the cubic Bézier curve given by the following four points:



De Casteljau Algorithm

Evaluation scheme (cubic case):



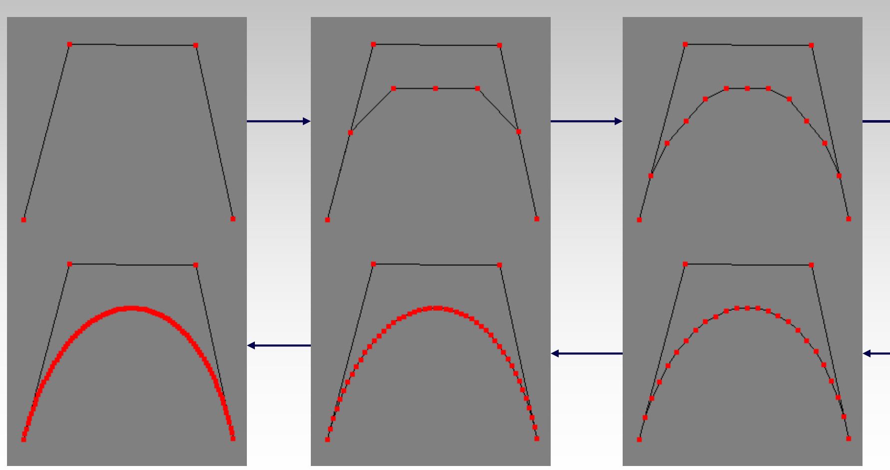
Observation

De Casteljau generates 2 new control polygons!

- For parameter interval [0,1/2], and [1/2,1]
- Can be used to recursively subdivide control polygon
- Can you prove it?

Example

Cubic case:



Subdivision Algorithm

Algorithm:

- Recursively subdivide control polygon at center of parameter interval
- Resulting control polygons converge to actual curve

Theorem (proof in book, Section 5.2):

- Convergence is very fast
  - *for n subdivision steps, the error (maximum distance between control polygons and curve) is*

$$\varepsilon < \frac{c}{2^n}$$

for some constant c

Degree Elevation

Motivation:

- Sometimes necessary to view curve of degree m as curve of degree m+1 or higher
- Control points of degree m Bézier curve can be geometrically converted into degree (m+1) control points for **same** curve

Degree Elevation

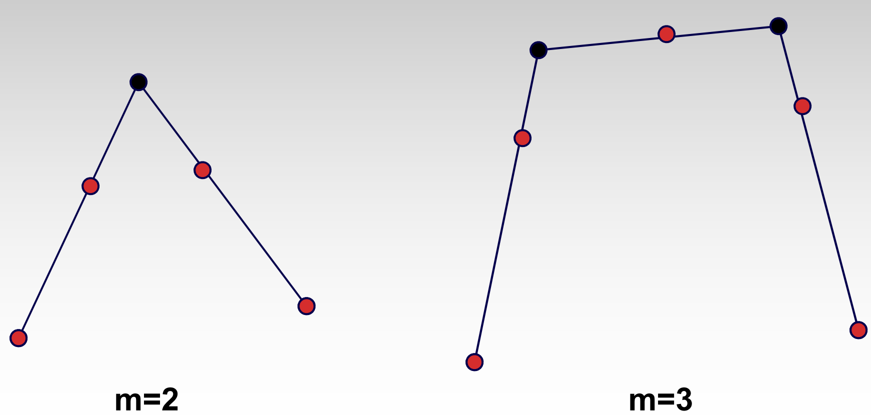
Replace degree m polynomial (m+1 control points) with degree m+1 polynomial (m+2 control points):

- New control points $\mathbf{b}'_i$:

$$\mathbf{b}_i' = \frac{1}{m+1}\left[i \cdot \mathbf{b}_{i-1} + (m+1-i)\mathbf{b}_i\right]$$

Degree Elevation

Examples:



Derivatives of Bézier Curves

Theorem (proof on board):

- The derivative of a Bézier curve

$$F(t) := \sum_{i=0}^{m} B_i^m(t) \cdot \mathbf{b}_i$$

is given as

$$F'(t) := m \cdot \sum_{i=0}^{m-1} B_i^{m-1}(t) \cdot (\mathbf{b}_{i+1} - \mathbf{b}_i)$$