CPSC 424
Affine Combinations,
de Casteljau Algorithm

Syllabus

Curves in 2D and 3D
- Implicit vs. Explicit vs. Parametric curves
- Bézier curves, de Casteljau algorithm, Polar forms
- Continuity
- B-Splines
- Subdivision Curves

Properties of Curves and Surfaces

Surfaces
Bézier Curves - Polynomials

Bernstein Polynomials

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

- Graph for degree \( m = 3 \):

Bernstein Polynomials

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

Properties:

- \( B_i^m(t) \) is a polynomial of degree \( m \)

- \( B_i^m(t) \geq 0 \) for \( t \in [0,1] \); \( B_0^m(0) = 1; B_i^m(0) = 0 \) for \( i \neq 0 \)

- \( B_i^m(t) = B_{m-i}^m(1-t) \)
Bernstein Polynomials

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

**Properties:**

- \( B^m_i(t) \) has exactly one maximum in the interval 0..1. It is at \( t = \frac{i}{m} \) (proof: compute derivative…)

- W/o proof: all \((m+1)\) functions \( B^m_i \) are linearly independent
  - Thus they form a basis for all polynomials of degree \( \leq m \)

---

**More properties**

- \( \sum_{i=0}^{m} B^m_i(t) = (t + (1-t))^m \equiv 1 \)
  - (proof: apply Binomial Theorem to definition)

- \( B^m_i(t) = t \cdot B^{m-1}_{i-1}(t) + (1-t) \cdot B^{m-1}_i(t) \)
  - (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
  \[ F(t) = \sum_{i=0}^{m} b_i B_i^m(t) \]
- \( b_i \) are called control points of Bézier curve
- Control polygon obtained by connecting control points with line segments

**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 \( t=1 \)

**Without proof for the moment (will be easier to show later):**
- Bézier curve is tangential to control polygon at endpoints
- Curve lies within convex hull of control points
- Curve is affine invariant
- There is a fast, recursive evaluation algorithm
Lagrange v.s. Bezier

\textbf{Lagrange}

\textit{Basis:} Lagrange Polynomials

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

\textbf{Spline:}

\[ F(t) = \sum_{i=0}^{m} b_i L_i^m(t_j) \]

- \textit{Interpolates control points}
- \textit{Hard to control (“wiggles”)}

\textbf{Bezier}

\textit{Basis:} Bernstein Polynomials

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

\textbf{Spline:}

\[ F(t) = \sum_{i=0}^{m} b_i B_i^m(t) \]

- \textit{Approximates control points}
- \textit{Easy to control (more or less)}

\section*{Convex Combination}

\textbf{Convex Combination}

\[ \sum_i \lambda_i x_i, \quad \sum_i \lambda_i = 1, \quad \lambda_i \geq 0 \]

\textbf{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 \(x_i\):

\[ [x_j] := \left\{ \sum_i \lambda_i x_i \mid \sum_i \lambda_i = 1; \quad \lambda_i \geq 0 \right\} \]
Back to Bézier Curves

\[ F(t) = \sum_{i=0}^{m} b_i B_i^m(t) \]

 Recall:

• Bernstein polynomials have values between 0 and 1 for \( t \in [0,1] \), and \( \sum_{i=0}^{m} B_i^m(t) = 1 \)

  – Therefore: every point on Bézier curve is convex combination of control points
  – Therefore: Bézier curve lies completely within convex hull of control points

De Casteljau Algorithm

Also recall:

• Recursive formula for Bernstein polynomials:

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

Plug into Bézier curve definition:

\[ F(t) = \sum_{i=0}^{m} b_i \left( t \cdot B_{i-1}^{m-1}(t) + (1-t) \cdot B_i^{m-1}(t) \right) \]

\[ = t \cdot \sum_{i=1}^{m} b_i B_{i-1}^{m-1}(t) + (1-t) \cdot \sum_{i=0}^{m-1} b_i B_i^{m-1}(t) \]
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 $F(t)$.

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!

$$F(t) = \sum_{i=0}^{0} b_i B_i^0(t) = b_i \cdot 1 \equiv b_i$$
De Casteljau Algorithm

After working out the math we get:

\[ F(t) = b_0^m(t) \hspace{1cm} \text{where} \]

\[ b_i^0(t) := b_i, \hspace{1cm} i = 0 \ldots m \]

\[ b_i^l(t) := (1-t) \cdot b_i^{l-1}(t) + t \cdot b_{i+1}^{l-1}(t) \]

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
**Example**

**Cubic case:**

- 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

**Replace degree m polynomial (m-1 control points) with degree m polynomial (m control points):**
- New control points $b'_i$:

$$b'_i = \frac{1}{m+1} \left[ i \cdot b_{i-1} + (m+1-i) b_i \right]$$
Degree Elevation

Examples:

\[ m=2 \]
\[ m=3 \]

Derivatives of Bézier Curves

**Theorem (proof on board):**

- The derivative of a Bézier curve

\[
F(t) := \sum_{i=0}^{m} B_i^m(t) \cdot b_i
\]

is given as

\[
F'(t) := m \cdot \sum_{i=0}^{m-1} B_i^{m-1}(t) \cdot (b_{i+1} - b_i)
\]