Remainder of the Course

Splines (Today)

- **Linear and Hermite Interpolation**
- **Cubic Basis Functions**
- **Bezier Curves**

Animation

Hidden Surface Algorithms

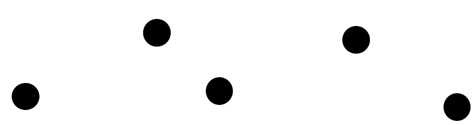
Reflection Models

Lighting (Ray Tracing and Radiosity)

Texture Mapping

Rendering Tricks

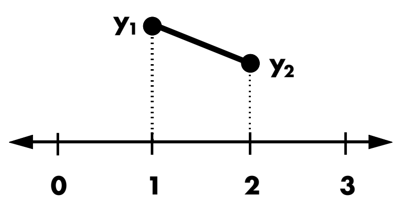
Splines

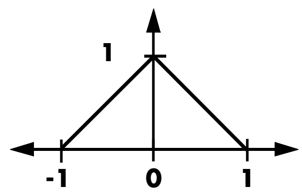


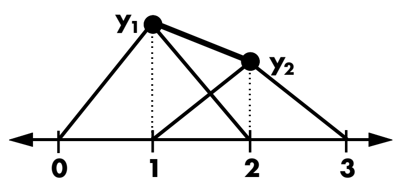
Functions that interpolate/approximate

1. **Filtering and reconstruction for images**
2. **Keyframes and inbetweens for animation**
3. **Curves and surfaces for modeling**

Linear Interpolation = Triangle Basis



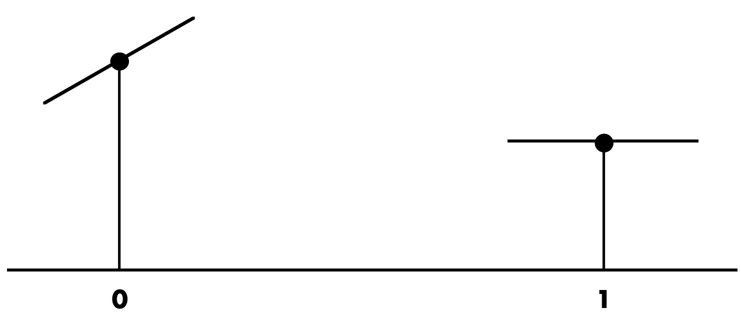




$$P(t) = y_1 T_1(t) + y_2 T_2(t)$$

$$T(t) = \begin{cases} 0 & t < -1 \\ 1+t & -1 < t < 0 \\ 1-t & 0 < t < 1 \\ 0 & t > 1 \end{cases}$$

Cubic Hermite Interpolation



Cubic Hermite Interpolation

$$\begin{aligned} \text{Given}: &\quad P(0), P(1), P'(0), P'(1) \\ \text{Compute}: &\quad P(t) = at^3 + bt^2 + ct + d \quad \textbf{(Cubic Polynomial)} \\ \text{Derivative}: &\quad P'(t) = 3at^2 + 2bt + c \end{aligned}$$

$$\begin{aligned} P(0) &= h_0 = d \\ P(1) &= h_1 = a + b + c + d \\ P'(0) &= h_2 = c \\ P'(1) &= h_3 = 3a + 2b + c \end{aligned}$$

Solve

$$\begin{aligned} a &= 2h_0 - 2h_1 + h_2 + h_3 \\ b &= -3h_0 + 3h_1 - 2h_2 - h_3 \\ c &= h_2 \\ d &= h_0 \end{aligned}$$

Matrix Representation of Cubic Curves

$$P(t) == \begin{bmatrix} a & b & c & d \end{bmatrix} \begin{bmatrix} t^3 \\ t^2 \\ t \\ 1 \end{bmatrix} = \sum_{i=0}^{3} b_i B_i^3(t) = \begin{bmatrix} b_0 & b_1 & b_2 & b_3 \end{bmatrix} \begin{bmatrix} B_0^3(t) \\ B_1^3(t) \\ B_2^3(t) \\ B_3^3(t) \end{bmatrix}$$

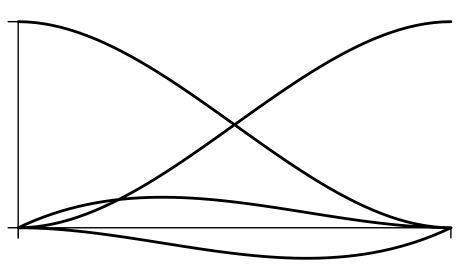
$$= \sum_{i=0}^{3} h_i H_i^3(t) = \begin{bmatrix} h_0 & h_1 & h_2 & h_3 \end{bmatrix} \begin{bmatrix} H_0^3(t) \\ H_1^3(t) \\ H_2^3(t) \\ H_3^3(t) \end{bmatrix}$$

Hermite Basis Matrix

$$P(t) == \begin{bmatrix} a & b & c & d \end{bmatrix} \begin{bmatrix} t^3 \\ t^2 \\ t \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} a & b & c & d \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 & 3 \\ 0 & 1 & 0 & 2 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix}}_{\begin{bmatrix} h_0 & h_1 & h_2 & h_3 \end{bmatrix}} \underbrace{\begin{bmatrix} 2 & -3 & 0 & 1 \\ -2 & 3 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} t^3 \\ t^2 \\ t \\ 1 \end{bmatrix}}_{\begin{bmatrix} H_0^3(t) \\ H_1^3(t) \\ H_2^3(t) \\ H_3^3(t) \end{bmatrix}}$$

$$= \begin{bmatrix} h_0 & h_1 & h_2 & h_3 \end{bmatrix} \begin{bmatrix} H_0^3(t) \\ H_1^3(t) \\ H_2^3(t) \\ H_3^3(t) \end{bmatrix} = \sum_{i=0}^{3} h_i H_i^3(t)$$

Hermite Basis Functions



$$H_0(t) = 2t^3 + 3t^2 + 1$$
$$H_1(t) = -2t^3 + 3t^2$$
$$H_2(t) = t^3 - 2t^2 + t$$
$$H_3(t) = t^3 - t^2$$

Chaiken's Algorithm

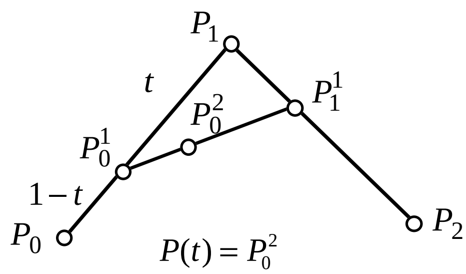
Step 1

$$P_0^1 = (1-t)P_0 + tP_1$$

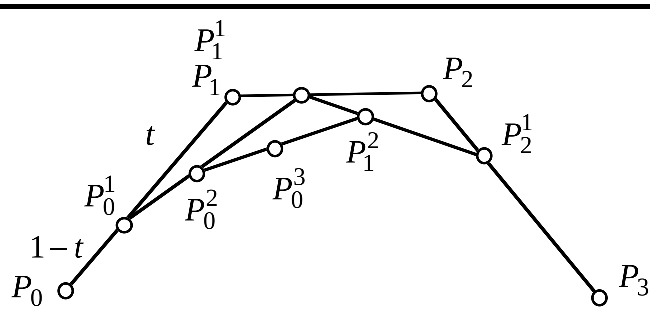
$$P_1^1 = (1-t)P_1 + tP_2$$

Step 2

$$P_0^2 = (1-t)P_0^1 + tP_1^1$$



Bezier Curves



Properties

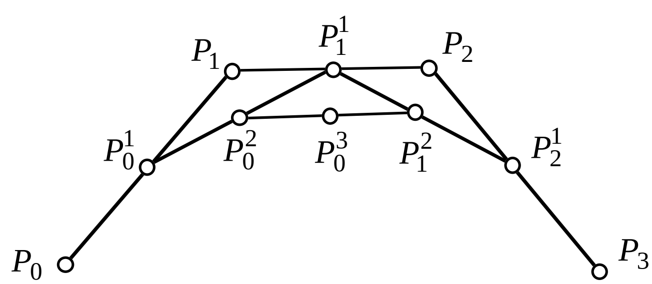
Property 1: Interpolate end points

Property 2: Tangents

Property 3: Convex hull property

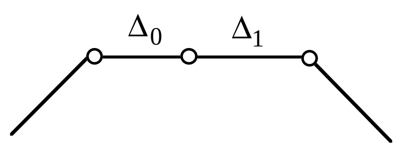
Property 4: Symmetry

Subdivision and Extrapolation Algorithm

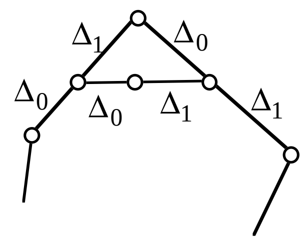


Piecewise Curves

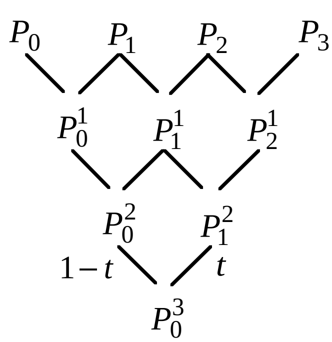
C¹ Continuity



C² Continuity



Bernstein Polynomials



$$P(t) = \sum_{i=0}^{n} P_i B_i^n(t)$$

$$B_i^n(t) = \binom{n}{i} t^i (1-t)^{n-i}$$

Bezier Curves = Bernstein Polynomials

$$B_i^n(t) = \binom{n}{i} t^i (1-t)^{n-i}$$

$$B_0^3(t) = \binom{3}{0} t^0 (1-t)^3 = -t^3 + 3t^2 - 3t + 1$$

$$B_1^3(t) = \binom{3}{1} t^1 (1-t)^2 = \frac{3!}{2!}\left(t(1-2t+t^2)\right) = 3t^3 - 6t^2 + 3t$$

$$B_2^3(t) = \binom{3}{2} t^2 (1-t)^1 = \frac{3!}{2!}\left(t^2(1-t)\right) = -3t^3 + 3t^2$$

$$B_2^3(t) = \binom{3}{3} t^3 (1-t)^0 = t^3$$

$$\boxed{\begin{bmatrix} B_0^3 \\ B_1^3 \\ B_2^3 \\ B_3^3 \end{bmatrix} = \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} t^3 \\ t^2 \\ t^1 \\ t^0 \end{bmatrix}}$$