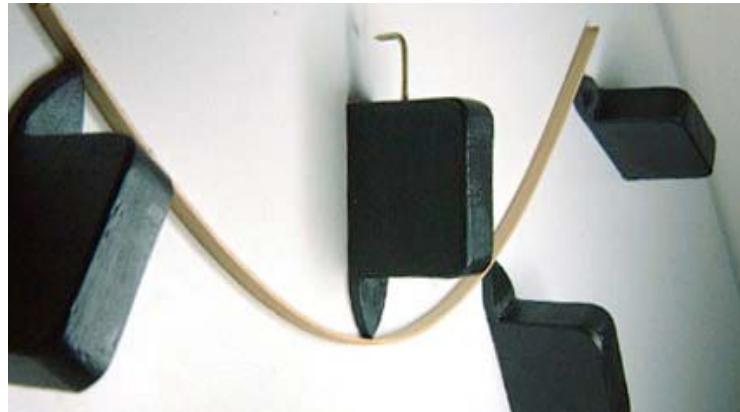


# Splines and Interpolation



## Topics

---

**Splines**

**Interpolation**

**Basis functions**

**Linear interpolation; triangular basis functions**

**Cubic basis functions**

**Bezier Curves**

# Splines

---

Convert discrete values/points to continuous function/curve

## Applications

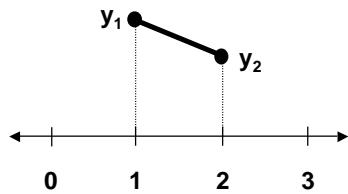
- Interpolating across a triangle
  - Interpolating between vertices
- Filtering and reconstruction images
  - Interpolating between pixels/texels
- Generating motion
  - Interpolating between keyframes
- Curves and surfaces
  - Interpolating between control points

# Basis Functions

## Linear Interpolation

---

$$y(t) = (1 - t) y_1 + t y_2$$



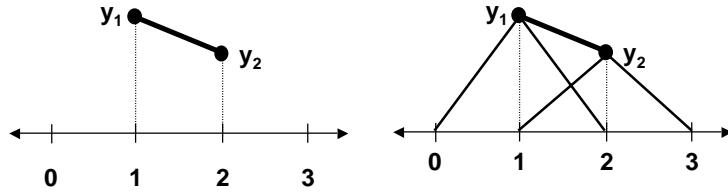
CS148 Lecture 6

Pat Hanrahan, Winter 2007

## Linear Interpolation = Triangle Basis

---

$$y(t) = (1 - t) y_1 + t y_2 \quad y(t) = y_1 T_1(t) + y_2 T_2(t)$$



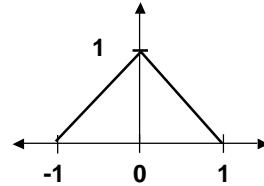
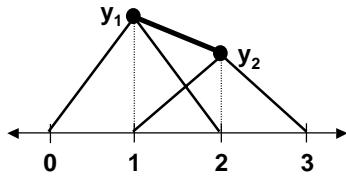
CS148 Lecture 6

Pat Hanrahan, Winter 2007

## Triangle Basis

---

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



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

CS148 Lecture 6

Pat Hanrahan, Winter 2007

## Lerp

---

### Lerp

$$\begin{aligned}\text{lerp}(t, v0, v1) &= (1-t)*v0 + t*v1 \\ &= v0 + t*(v1-v0)\end{aligned}$$

CS148 Lecture 6

Pat Hanrahan, Winter 2007

## Bilerp

### Lerp

```
lerp(t,v0,v1) = (1-t)*v0+t*v1  
= v0 + t*(v1-v0)
```

### Bilerp



```
bilerp(s,t,v0,v1,v2,v3)  
{  
    v01 = lerp(t,v0,v1);  
    v23 = lerp(t,v2,v3);  
    v = lerp(s,v01,v23);  
    return v;  
}
```

CS148 Lecture 6

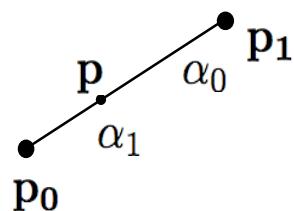
Pat Hanrahan, Winter 2007

## Barycentric Interpolation

### Edge

$$\mathbf{p} = \alpha_0 \mathbf{p}_0 + \alpha_1 \mathbf{p}_1$$

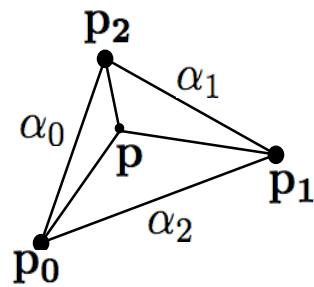
$$\alpha_0 + \alpha_1 = 1$$



### Triangle

$$\mathbf{p} = \alpha_0 \mathbf{p}_0 + \alpha_1 \mathbf{p}_1 + \alpha_2 \mathbf{p}_2$$

$$\alpha_0 + \alpha_1 + \alpha_2 = 1$$



CS148 Lecture 6

Pat Hanrahan, Winter 2007

## Basis Functions

---

**Discrete to continuous**

**Function**

$$y(t) = \sum_{i=0}^n y_i B_i(t)$$

**Curve**

$$\mathbf{p}(t) = \sum_{i=0}^n \mathbf{c}_i B_i(t)$$

**Control point:**  $\mathbf{c}_i$

**Basis function:**  $B_i(t)$

CS148 Lecture 6

Pat Hanrahan, Winter 2007

## Interpolating Function

---

**Conditions**

$$B_i(0) = 1$$

$$B_i(j) = 0$$

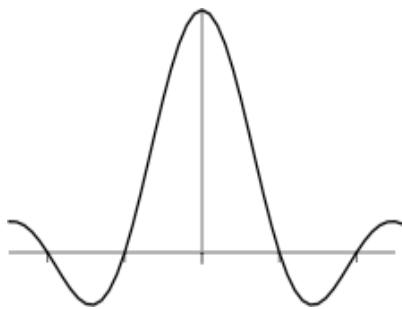
$$y(t) = \sum_{i=0}^n y_i B_i(t)$$

CS148 Lecture 6

Pat Hanrahan, Winter 2007

## The “Sinc” Function

---



$$\text{sinc } 0 = \lim_{x \rightarrow 0} \frac{\sin \pi x}{\pi x} = 1$$

$$\text{sinc } 1 = \frac{\sin \pi}{\pi} = \frac{0}{1} = 0$$

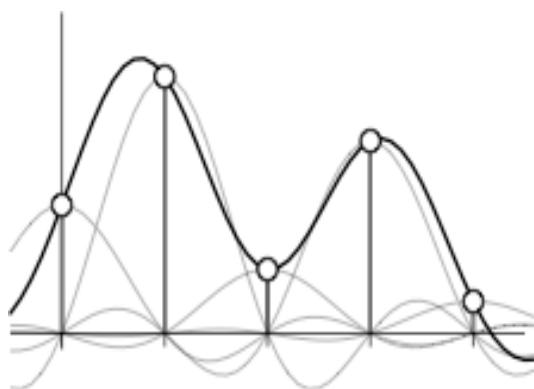
$$\text{sinc } x = \frac{\sin \pi x}{\pi x}$$

CS148 Lecture 6

Pat Hanrahan, Winter 2007

## Interpolating with sinc

---



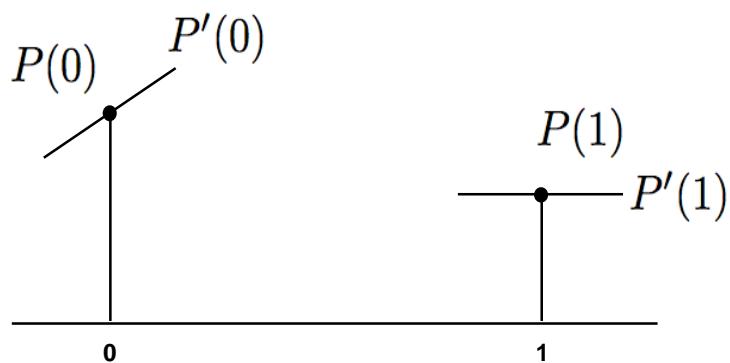
CS148 Lecture 6

Pat Hanrahan, Winter 2007

## Cubic Hermite Interpolation

### Cubic Hermite Interpolation

---



**Given: values and derivatives**

## Cubic Hermite Interpolation

Given :  $P(0), P(1), P'(0), P'(1)$

Compute :  $P(t) = at^3 + bt^2 + ct + d$     (**Cubic Polynomial**)

Derivative :  $P'(t) = 3at^2 + 2bt + c$

$$P(0) = h_0 = d$$

$$P(1) = h_1 = a + b + c + d$$

$$P'(0) = h_2 = c$$

$$P'(1) = h_3 = 3a + 2b + c$$

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

Solve

CS148 Lecture 6

Pat Hanrahan, Winter 2007

## Hermite Basis Matrix

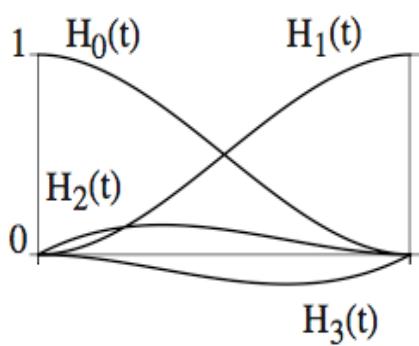
$$\begin{aligned} P(t) &= [a \quad b \quad c \quad d] \begin{bmatrix} t^3 \\ t^2 \\ t \\ 1 \end{bmatrix} = [a \quad b \quad c \quad d] \underbrace{\begin{bmatrix} 0 & 1 & 0 & 3 \\ 0 & 1 & 0 & 2 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix}}_{[h_0 \quad h_1 \quad h_2 \quad h_3]} \underbrace{\begin{bmatrix}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

CS148 Lecture 6

Pat Hanrahan, Winter 2007

## Hermite Basis Functions

---



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

$$H_1(t) = -2t^3 + 3t^2$$

$$H_2(t) = t^3 - 2t^2 + t$$

$$H_3(t) = t^3 - t^2$$

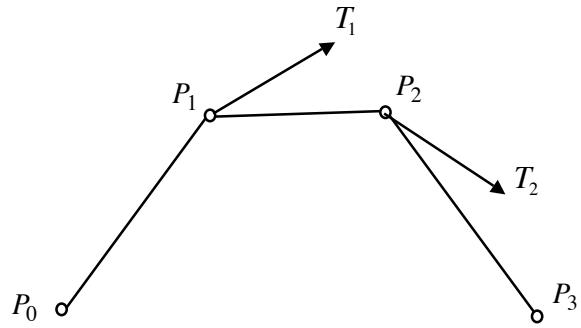
$$\mathbf{M}_H = \begin{bmatrix} 2 & -3 & 0 & 1 \\ -2 & 3 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix}$$

CS148 Lecture 6

Pat Hanrahan, Winter 2007

## Cubic Splines

## Hermite Spline

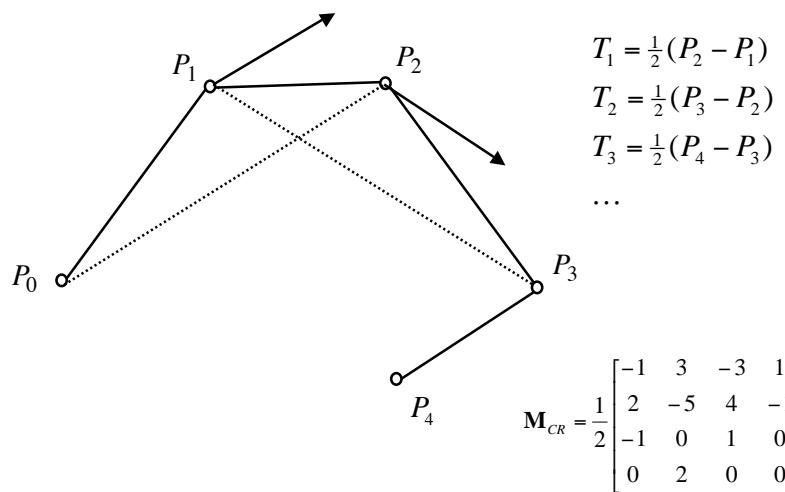


$$\mathbf{M}_H = \begin{bmatrix} 2 & -3 & 0 & 1 \\ -2 & 3 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix}$$

CS148 Lecture 6

Pat Hanrahan, Winter 2007

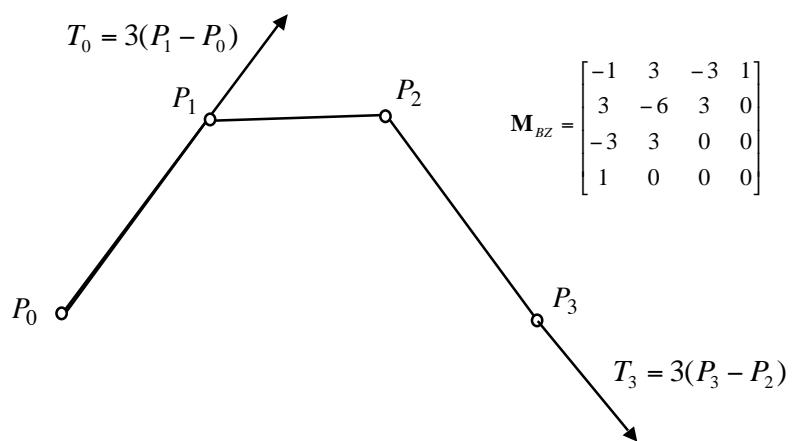
## Catmull-Rom Spline



CS148 Lecture 6

Pat Hanrahan, Winter 2007

## Bezier Curve



$$\mathbf{M}_{BZ} = \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

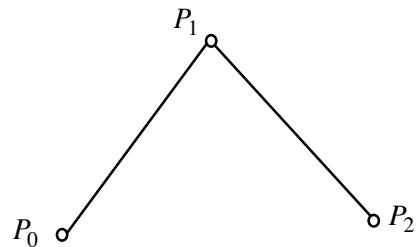
CS148 Lecture 6

Pat Hanrahan, Winter 2007

## Demo Bezier Curves

## Chaiken's Algorithm

---



CS148 Lecture 6

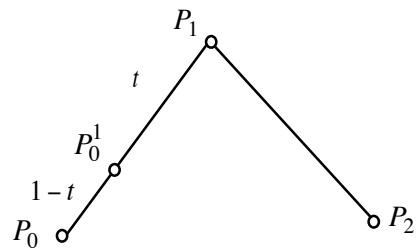
Pat Hanrahan, Winter 2007

## Chaiken's Algorithm

---

**Step 1**

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

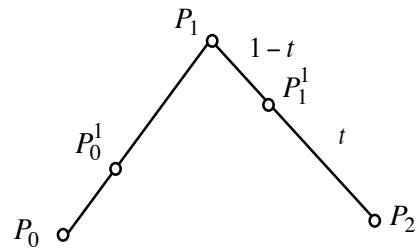


CS148 Lecture 6

Pat Hanrahan, Winter 2007

## Chaiken's Algorithm

**Step 1**  
 $P_0^1 = (1 - t)P_0 + tP_1$   
 $P_1^1 = (1 - t)P_1 + tP_2$



CS148 Lecture 6

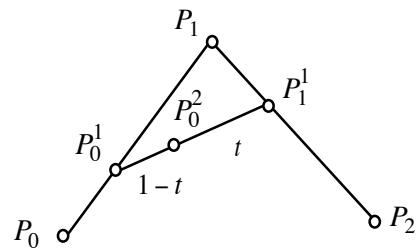
Pat Hanrahan, Winter 2007

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

$$P(t) = P_0^2$$

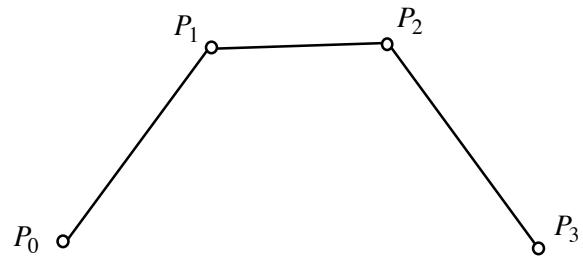


CS148 Lecture 6

Pat Hanrahan, Winter 2007

## Bezier Curves

---

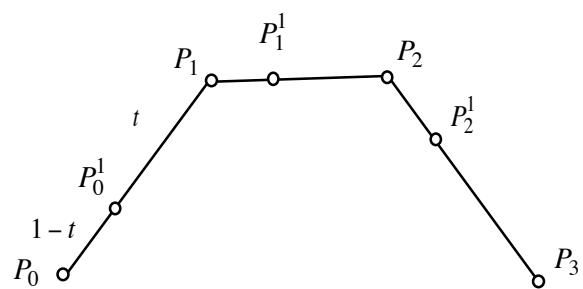


CS148 Lecture 6

Pat Hanrahan, Winter 2007

## Bezier Curves

---

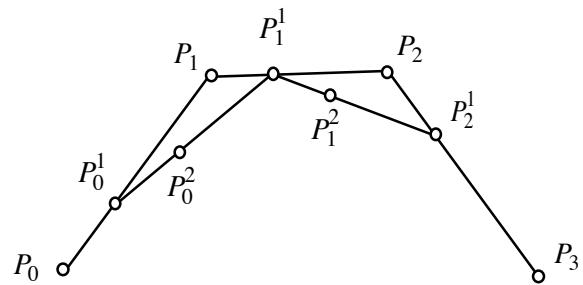


CS148 Lecture 6

Pat Hanrahan, Winter 2007

## Bezier Curves

---

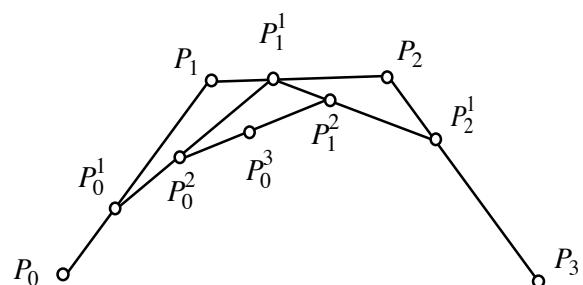


CS148 Lecture 6

Pat Hanrahan, Winter 2007

## Bezier Curves

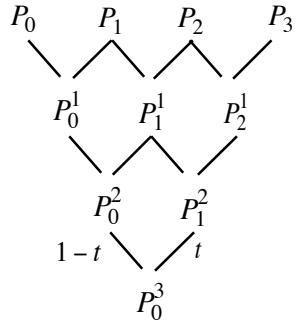
---



CS148 Lecture 6

Pat Hanrahan, Winter 2007

## Bernstein Polynomials



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

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

CS148 Lecture 6

Pat Hanrahan, Winter 2007

## Bernstein Polynomials

$$B_i^n(t) = \binom{n}{i} t^i (1-t)^{n-i}$$
$$\left| \begin{array}{l} B_0^3(t) = \binom{3}{0} t^0 (1-t)^3 = -t^3 + 3t^2 - 3t + 1 \\ B_1^3(t) = \binom{3}{1} t^1 (1-t)^2 = \frac{3!}{2!} (t(1-2t+t^2)) = 3t^3 - 6t^2 + 3t \\ B_2^3(t) = \binom{3}{2} t^2 (1-t)^1 = \frac{3!}{2!} (t^2(1-t)) = -3t^3 + 3t^2 \\ B_3^3(t) = \binom{3}{3} t^3 (1-t)^0 = t^3 \end{array} \right.$$

Bezier curves = Bernstein polynomials as basis

CS148 Lecture 6

Pat Hanrahan, Winter 2007

## Properties

---

**Property 1: Interpolate end points**

**Property 2: Tangents**

**Property 3: Convex hull property**

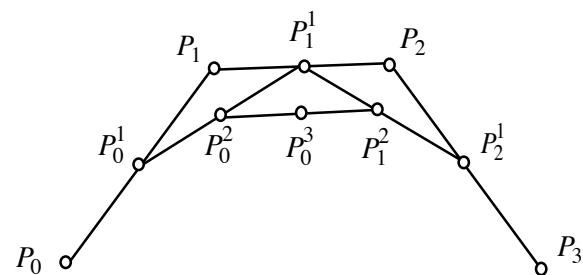
**Property 4: Symmetry**

CS148 Lecture 6

Pat Hanrahan, Winter 2007

## Subdivision and Extrapolation

---



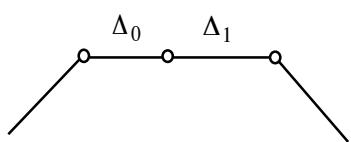
CS148 Lecture 6

Pat Hanrahan, Winter 2007

## Piecewise Curves

---

**C<sup>1</sup> Continuity**



**C<sup>2</sup> Continuity**

