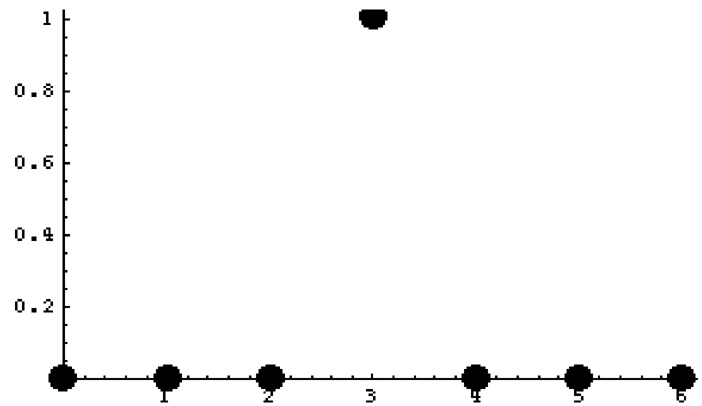
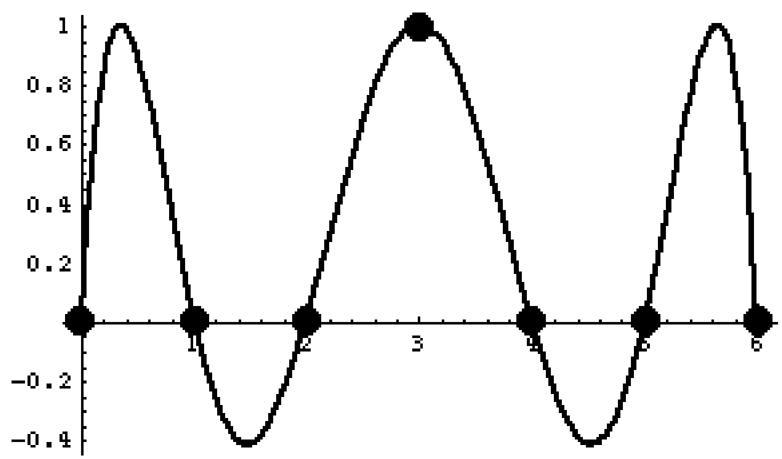
Bezier Approximation and De Casteljau's Algorithm

Ron Goldman
Department of Computer Science
Rice University

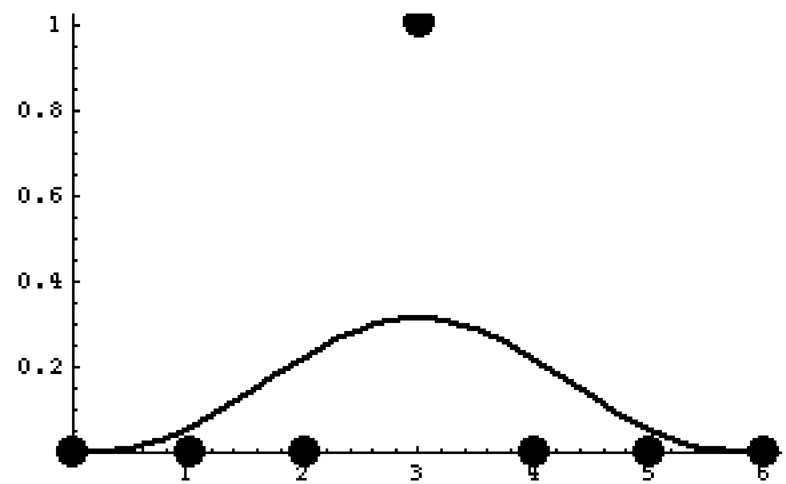
Interpolation and Approximation



Data

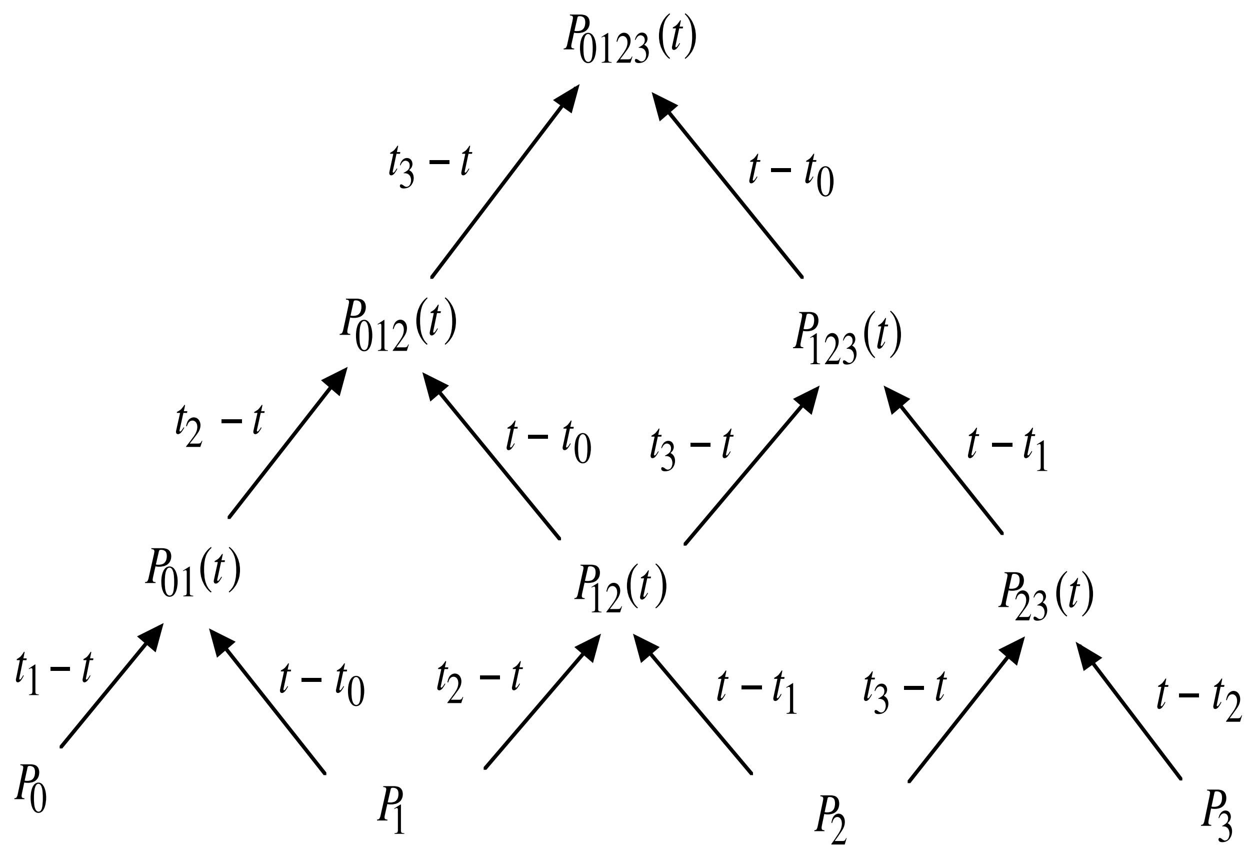


Interpolation of Data



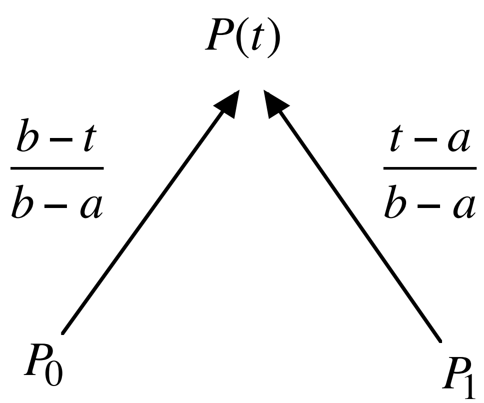
Approximation of Shape

Neville's Algorithm

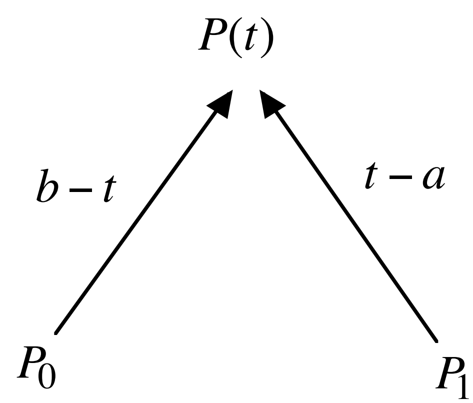


$$P_{0123}(t_k) = P_k$$

Linear Interpolation



Normalized

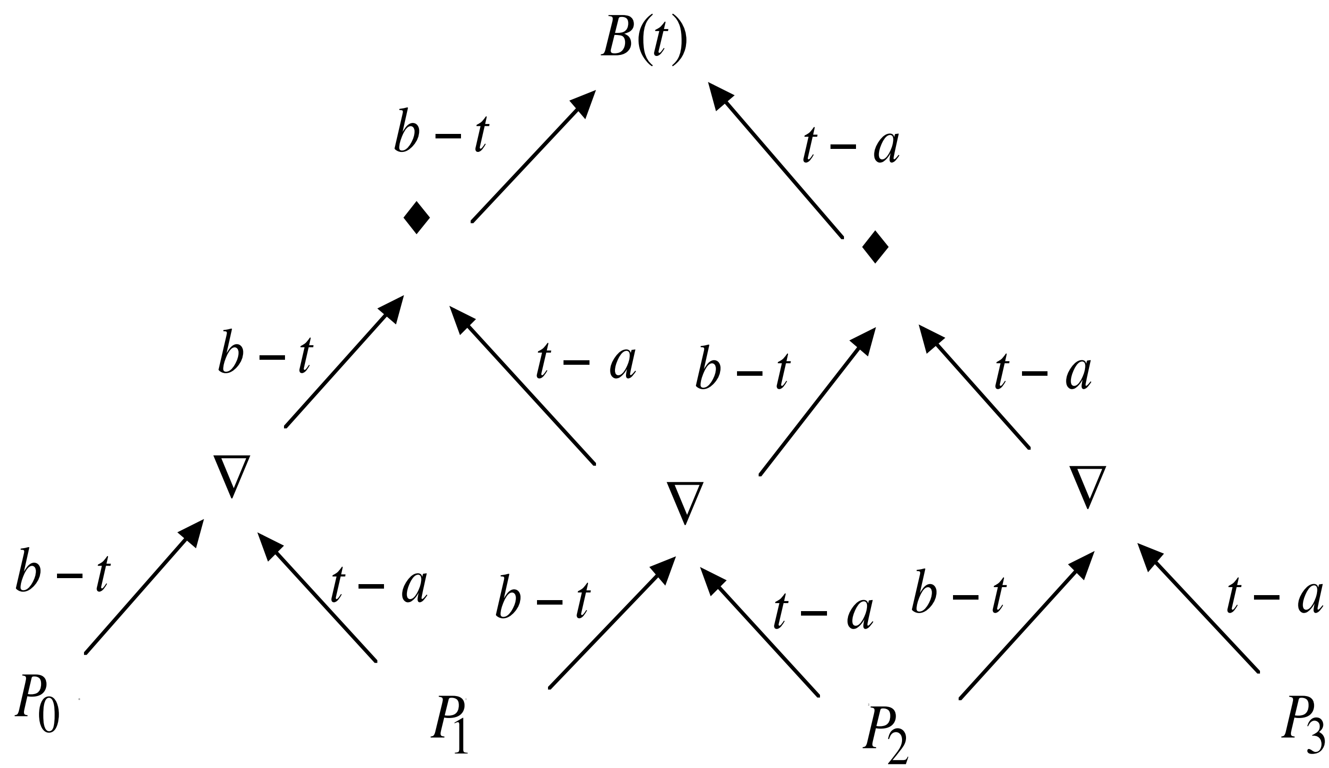


Unnormalized

$$P(t) = \frac{b-t}{b-a} P_0 + \frac{t-a}{b-a} P_1$$

$$P(a) = P_0 \qquad\qquad P(b) = P_1$$

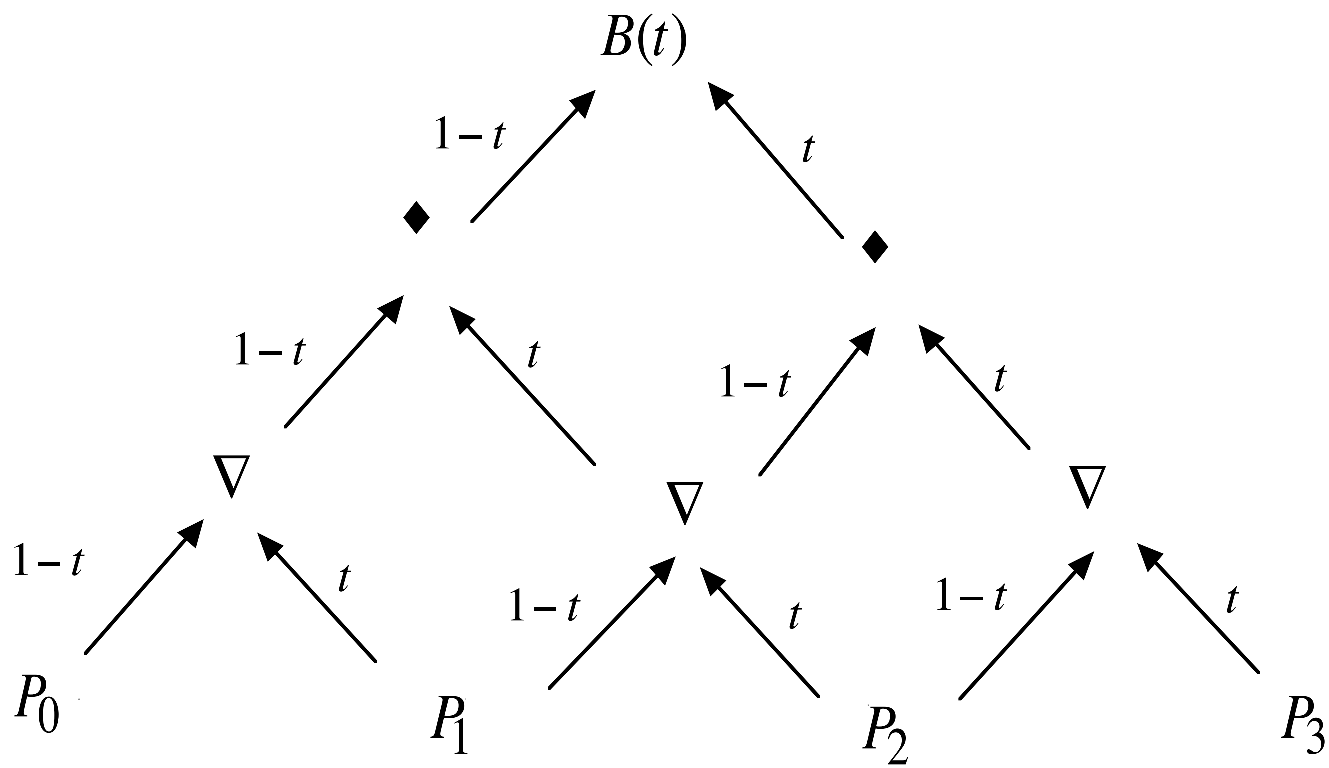
De Casteljau's Algorithm



$B(t)$ = Bezier Curve $\quad a \le t \le b$

$P_0, \ldots, P_n$ = Control Points

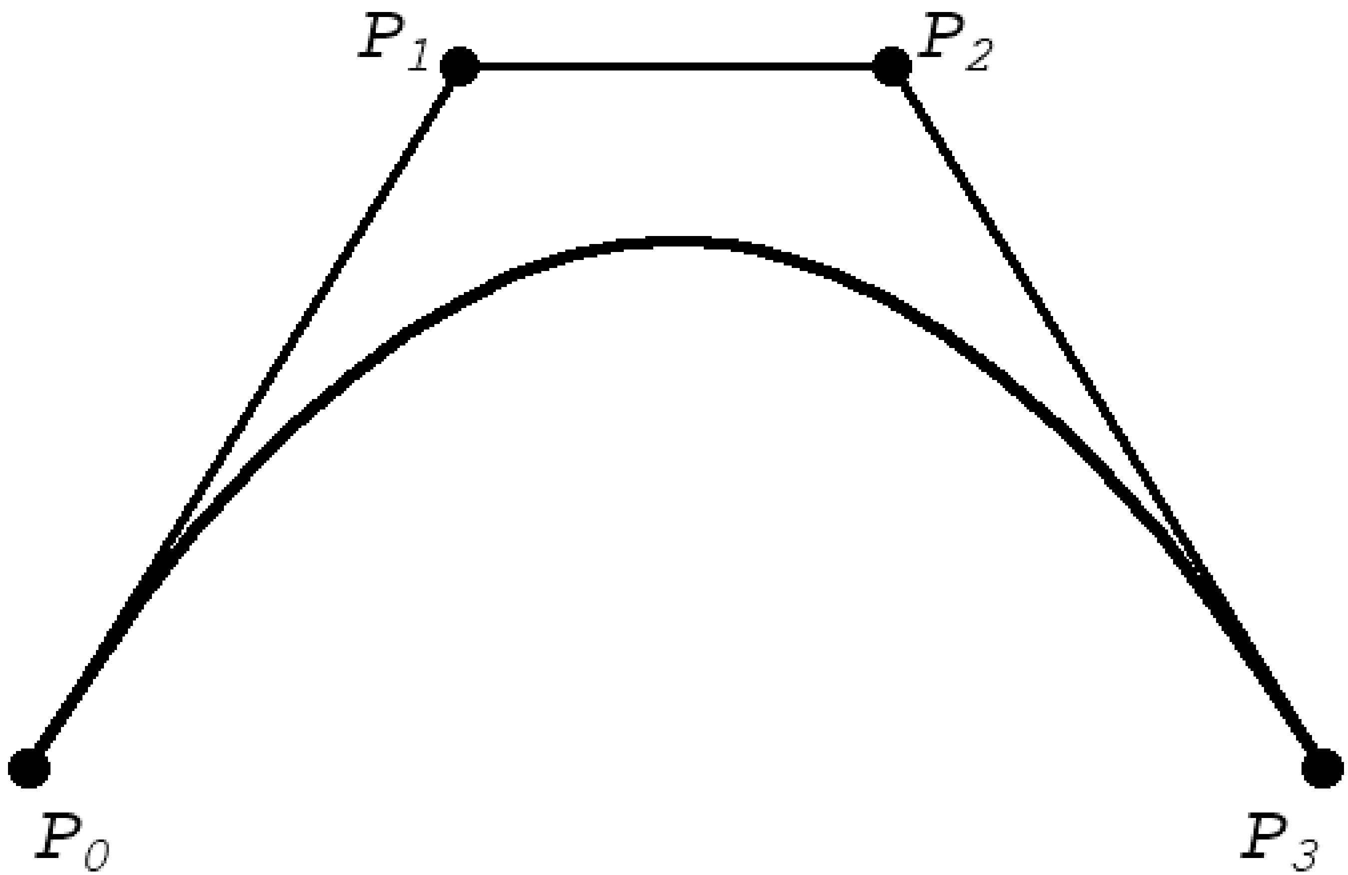
De Casteljau's Algorithm



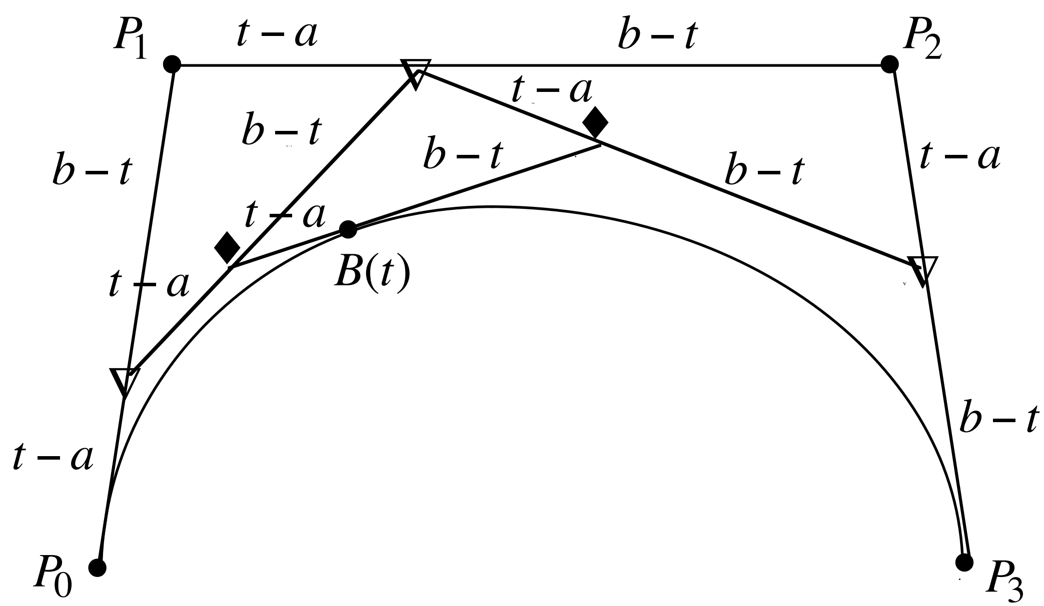
$B(t)$ = Bezier Curve $\quad 0 \le t \le 1$

$P_0, \ldots, P_n$ = Control Points

Cubic Bezier Curve



De Casteljau's Algorithm -- Geometric Interpretation



Elementary Properties of Bezier Curves

1. Polynomial

2. Translation Invariant -- Affine Invariant

3. Lie in Convex Hull of their Control Points

4. Symmetry

5. Interpolation of End Points

Translation Invariance

Linear Interpolation

$$L(t) = \left(\frac{b-t}{b-a}\right)P_0 + \left(\frac{t-a}{b-a}\right)P_1$$

$$L(t) + v = \left(\frac{b-t}{b-a}\right)(P_0 + v) + \left(\frac{t-a}{b-a}\right)(P_1 + v)$$

Bezier Curves

$$B\lfloor P_0, \ldots, P_n \rfloor(t) + v = B\lfloor P_0 + v, \ldots, P_n + v \rfloor(t)$$

Consequences

- Bezier Curves Depend Only on their Control Points
- Bezier Curves are Independent of the Choice of the Origin of the Coordinate System -- Also True for Lagrange Interpolation

Translation Invariance

$$
\begin{tikzcd}
 & & & B(t)+v & & & \\
 & & \blacklozenge+v \arrow[ur, "b-t"] & & \blacklozenge+v \arrow[ul, "t-a"'] & & \\
 & \nabla+v \arrow[ur, "b-t"] & & \nabla+v \arrow[ul, "t-a"'] \arrow[ur, "b-t"] & & \nabla+v \arrow[ul, "t-a"'] & \\
P_0+v \arrow[ur, "b-t"] & & P_1+v \arrow[ul, "t-a"'] \arrow[ur, "b-t"] & & P_2+v \arrow[ul, "t-a"'] \arrow[ur, "b-t"] & & P_3+v \arrow[ul, "t-a"']
\end{tikzcd}
$$

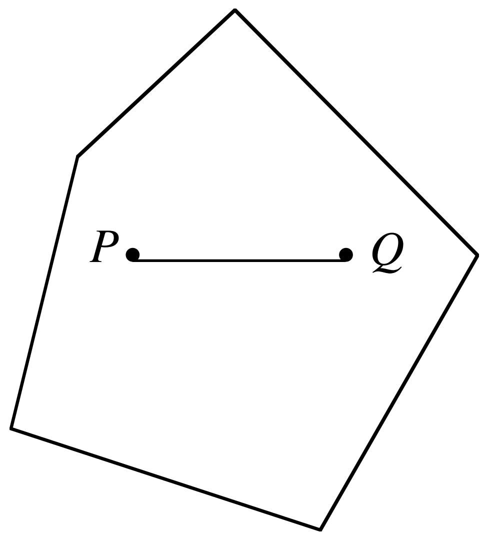
Affine Invariance

$$\begin{tikzcd}
 & & & \left(B(t),1\right) \ast M & & & \\
 & & (\blacklozenge,1) \ast M \arrow[ur, "b-t"] & & (\blacklozenge,1) \ast M \arrow[ul, "t-a"'] & & \\
 & (\nabla,1) \ast M \arrow[ur, "b-t"] & & (\nabla,1) \ast M \arrow[ul, "t-a"'] \arrow[ur, "b-t"] & & (\nabla,1) \ast M \arrow[ul, "t-a"'] & \\
(P_0,1) \ast M \arrow[ur, "b-t"] & & (P_1,1) \ast M \arrow[ul, "t-a"'] \arrow[ur, "b-t"] & & (P_2,1) \ast M \arrow[ul, "t-a"'] \arrow[ur, "b-t"] & & (P_3,1) \ast M \arrow[ul, "t-a"']
\end{tikzcd}$$

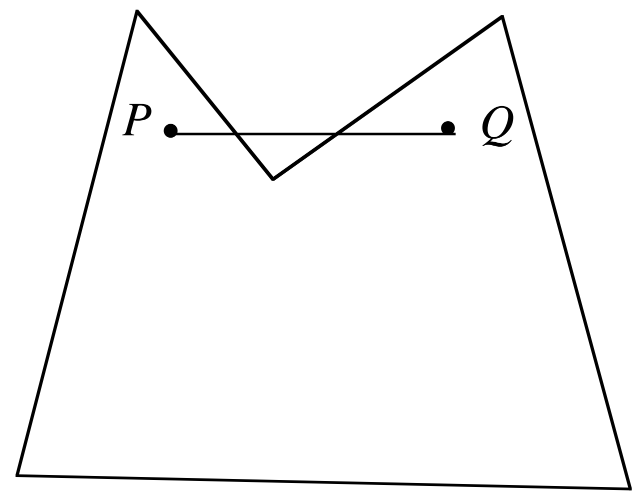
$$L(t) = \left(\frac{b-t}{b-a}\right)P_0 + \left(\frac{t-a}{b-a}\right)P_1$$

$$(L(t),1) \ast M = \left(\frac{b-t}{b-a}\right)\left(P_0,1\right) \ast M + \left(\frac{t-a}{b-a}\right)\left(P_1,1\right) \ast M$$

Convexity



(a) Convex Set



(b) Non-Convex Set

Definition

A set is *convex* if and only if for every two points in the set the line segment joining the two points lies entirely inside the set.

<u>**Convex Hull Property**</u>

Definition

Convex Hull$\{P_0,\ldots,P_n\}$ = Smallest Convex Set Containing $P_0,\ldots,P_n$

Theorem

Bezier Curves Lie in the Convex Hull of their Control Points.

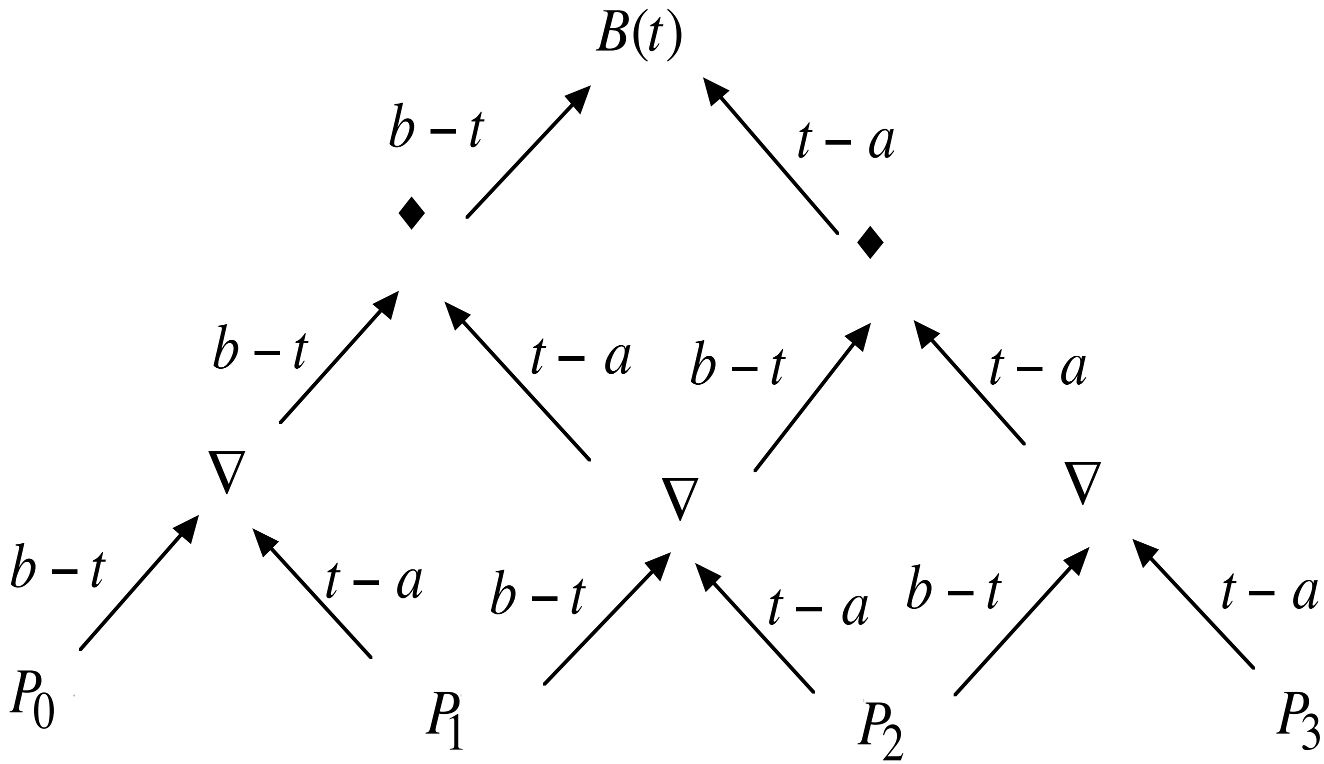
Proof

From the de Casteljau Algorithm by Induction on Degree.

Consequence

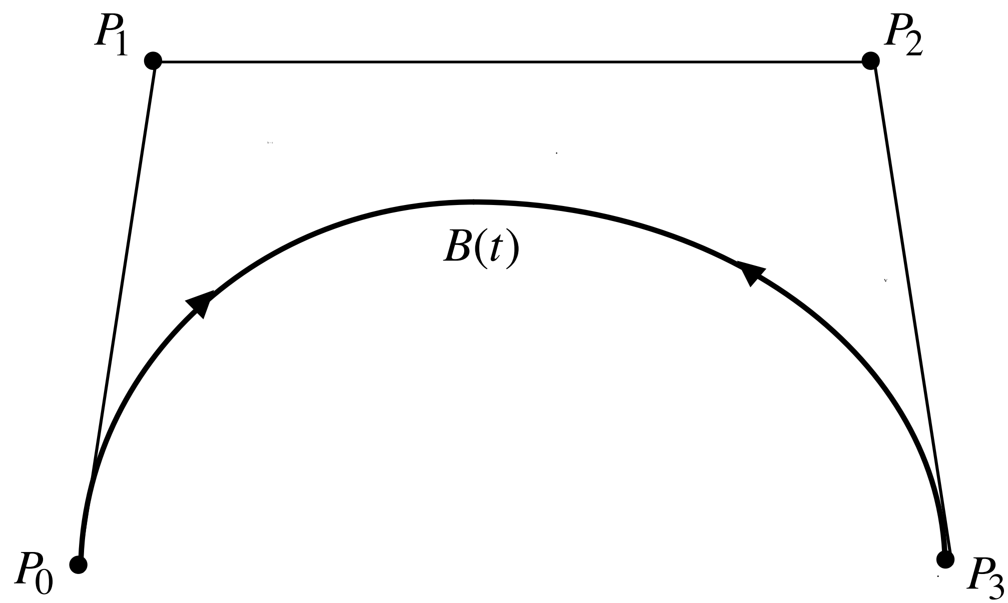
- If the Control Points are Visible, the Entire Bezier Curve is Visible.
- Not True for Lagrange Interpolation.

De Casteljau's Algorithm



$B(t)$ = Bezier Curve $\quad a \leq t \leq b$

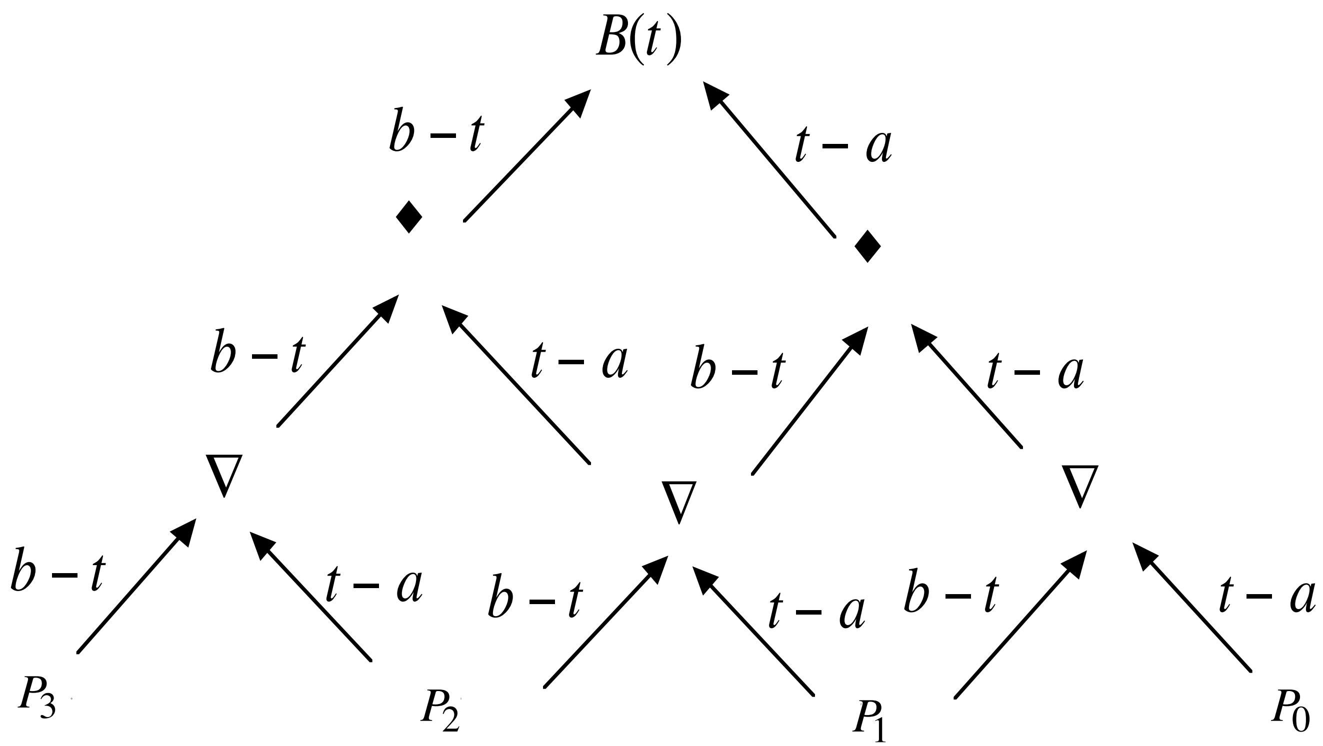
Symmetry



$$B\lfloor P_n, \ldots, P_0 \rfloor(t) = B\lfloor P_0, \ldots, P_n \rfloor(a + b - t)$$

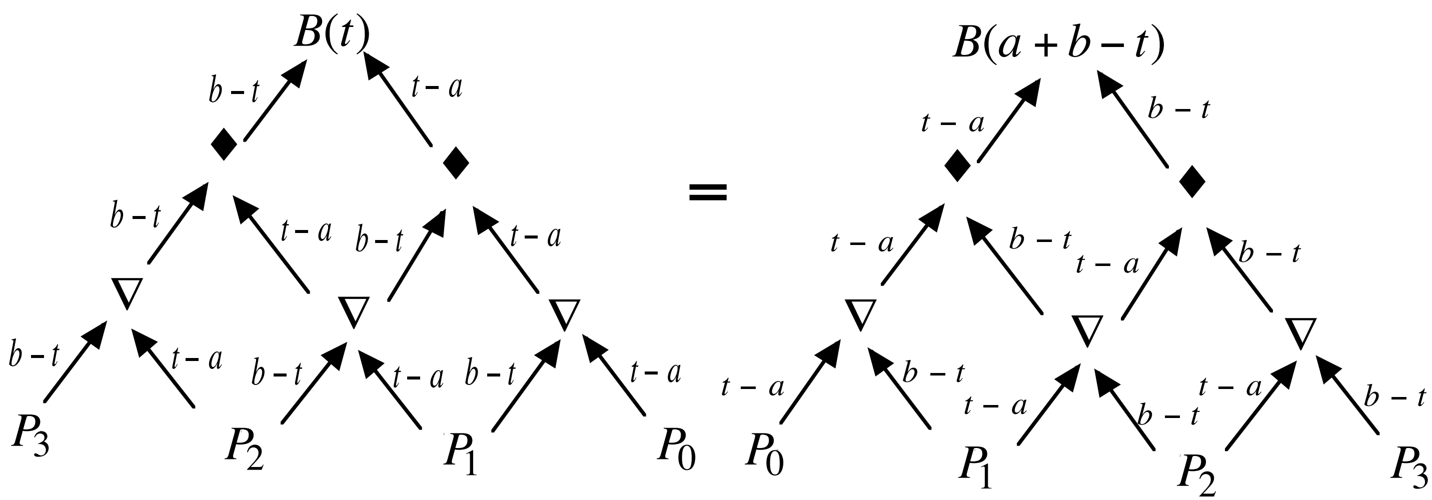
Reversing the order of the control points, reverses the orientation of the Bezier curve.

Symmetry



$$B\lfloor P_n,\dots,P_0\rfloor(t) = B\lfloor P_0,\dots,P_n\rfloor(a+b-t)$$

<u>Symmetry</u>



$$B\lfloor P_n, \ldots, P_0 \rfloor(t) = B\lfloor P_0, \ldots, P_n \rfloor(a + b - t)$$

Consequence

- Reversing the order of the control points, reverses the orientation of the Bezier curve.

Interpolation of End Points

Theorem

Bezier Curves Interpolate their First and Last Control Points.

In particular, $B(a) = P_0$ *and* $B(b) = P_n$

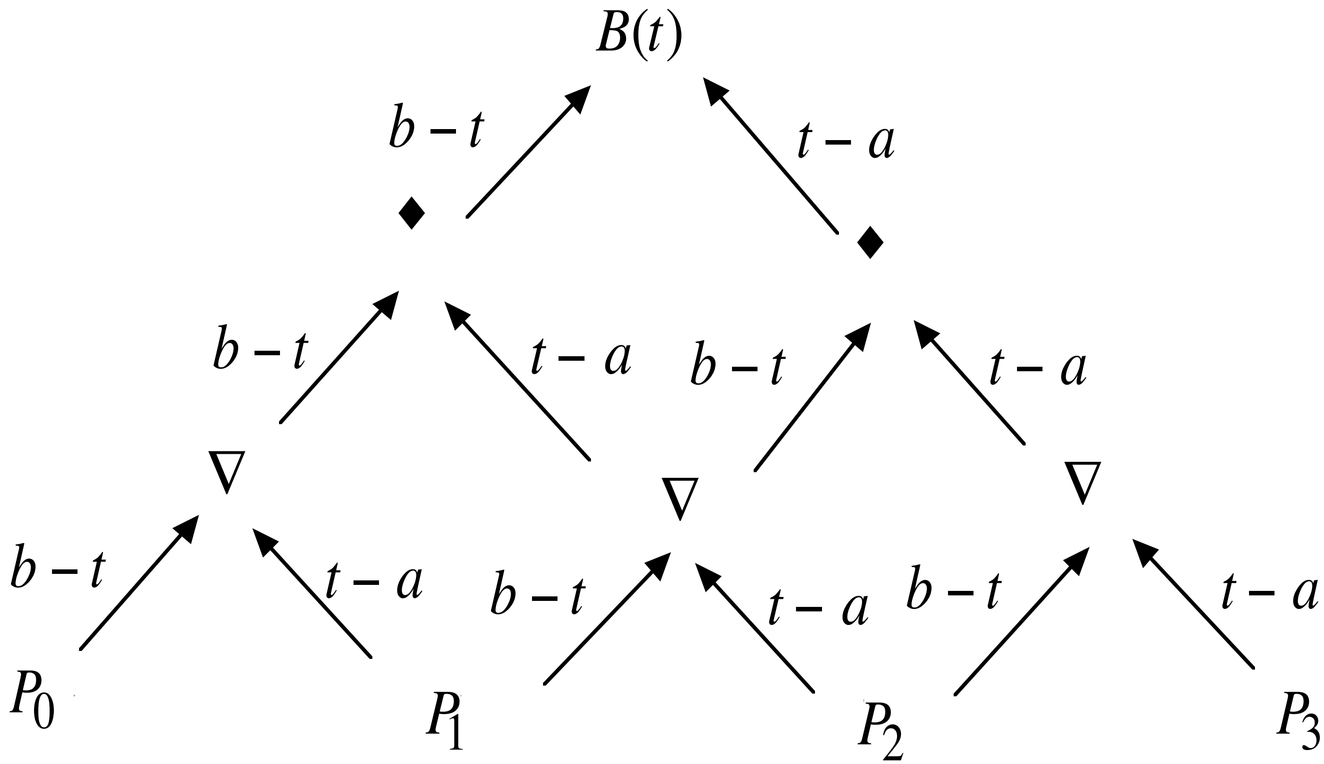
Proof

Examine the De Casteljau Diagram.

Consequence

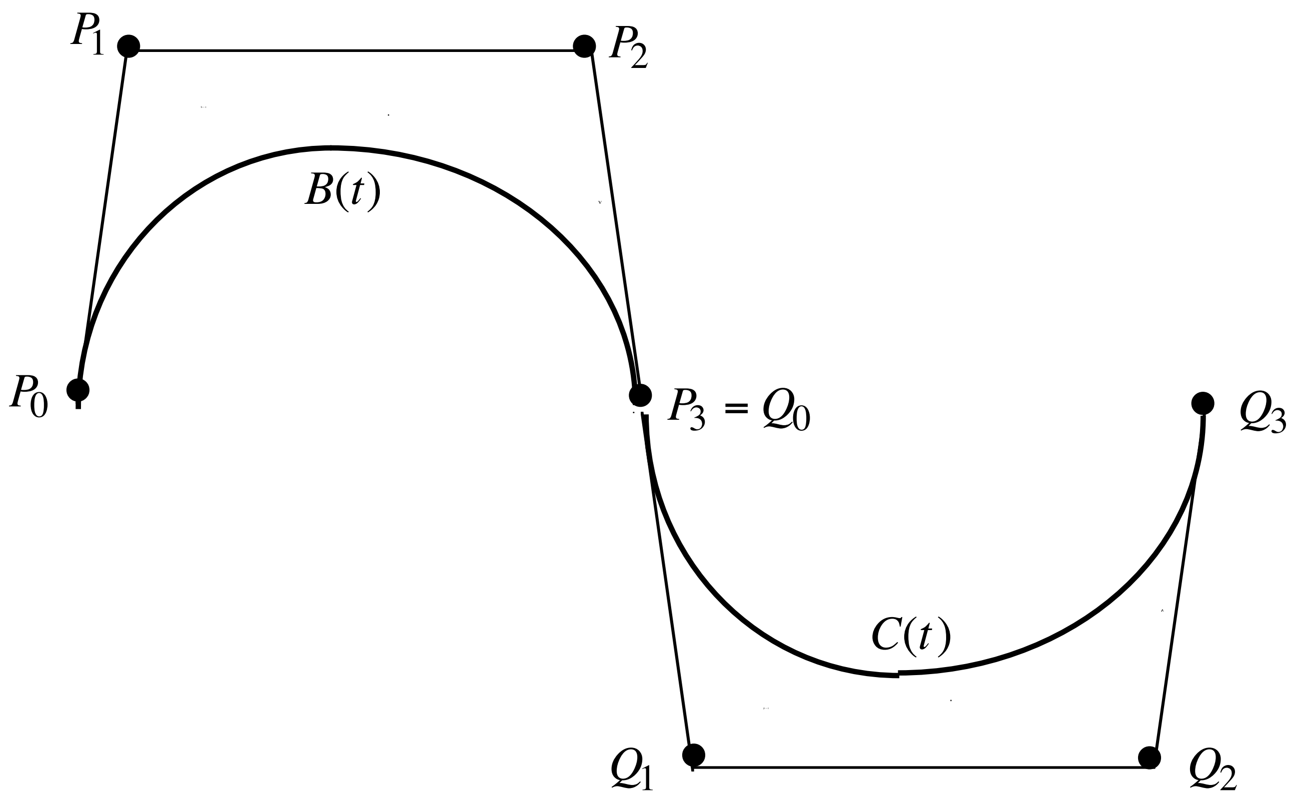
Easy to String Together Bezier Curves in a Continuous Manner.

Interpolation of End Points



$B(a) = P_0$ and $B(b) = P_3$

Piecewise Bezier Curve



Advantages of de Casteljau's Algorithm

1. *Numerically Stable*
   - Uses the Control Points Directly
   - Computes Only Convex Combinations

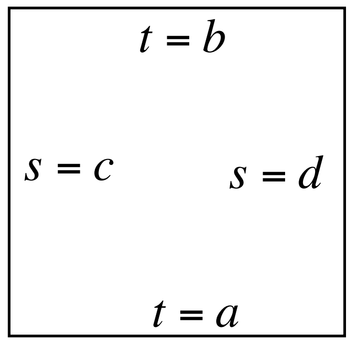
2. *Fast*
   - Dynamic Programming
   - $O(n^2)$ vs. $O(2^n)$

3. *Simple Structure*
   - All Computations are Identical

4. *Basic Properties of Bezier Curves*
   - Polynomial
   - Translation Invariant
   - Lie in Convex Hull of their Control Points
   - Symmetric
   - Interpolate First and Last Control Points

Tensor Product Bezier Surfaces

Setup



Domain -- Rectangle

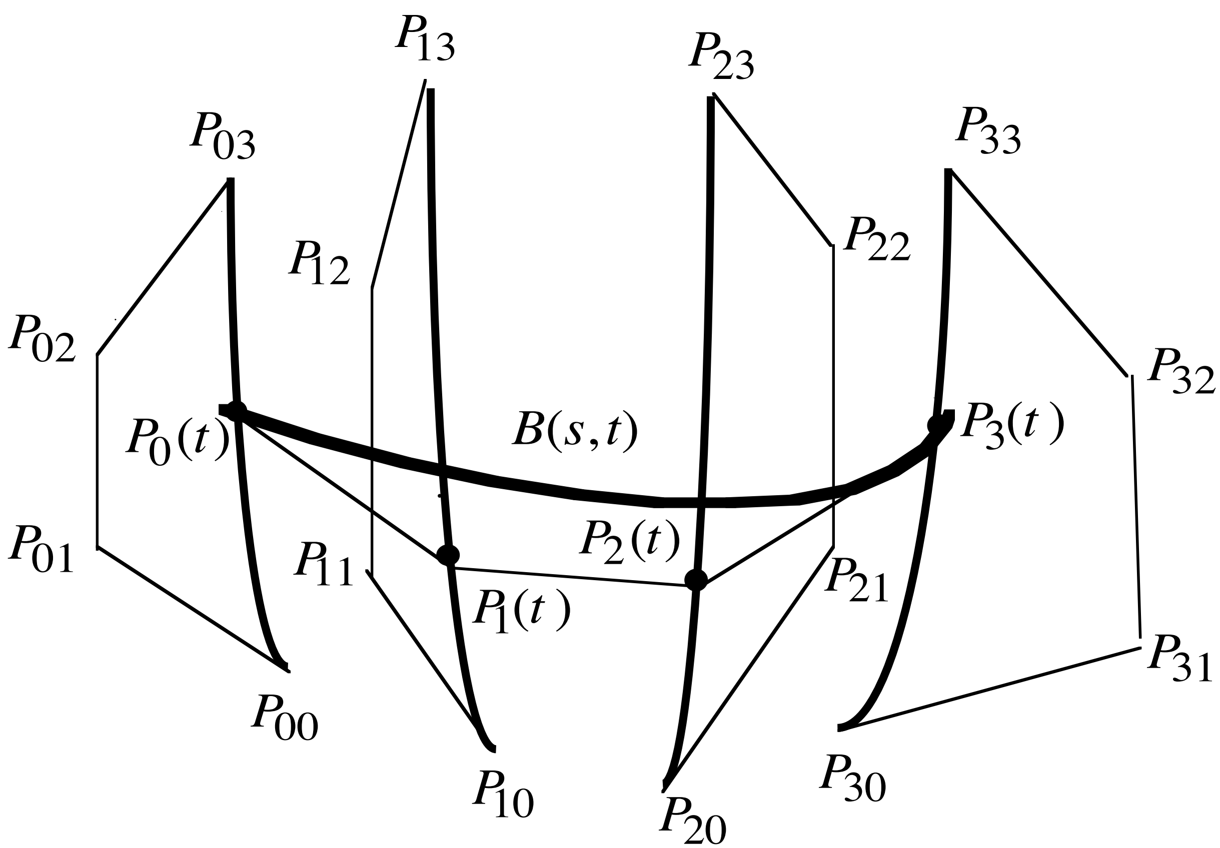
$$\begin{array}{cccc}
P_{03} & P_{13} & P_{23} & P_{33} \\
P_{02} & P_{12} & P_{22} & P_{32} \\
P_{01} & P_{11} & P_{21} & P_{31} \\
P_{00} & P_{10} & P_{20} & P_{30}
\end{array}$$

Range -- Rectangular Array of Points

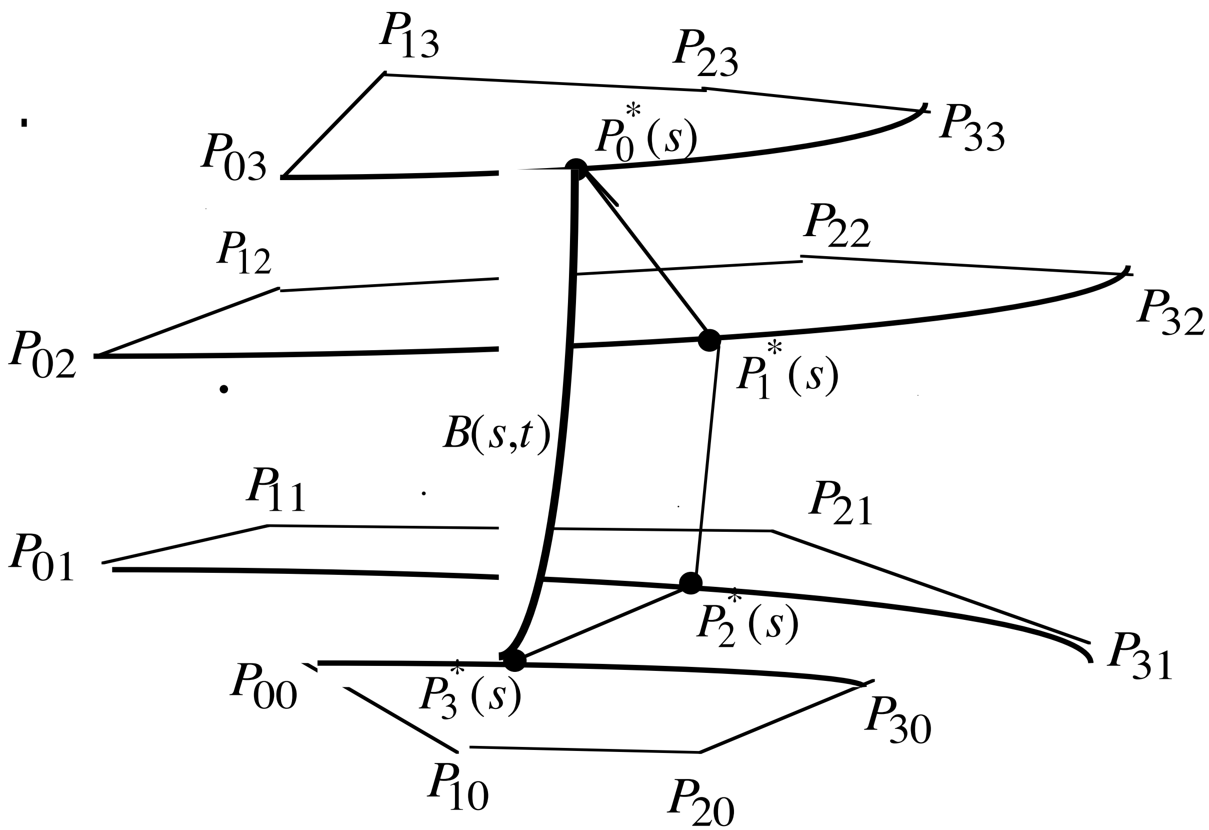
Problem

Find a <u>smooth</u> surface $B(s,t)$ that approximates the shape defined by the control points.

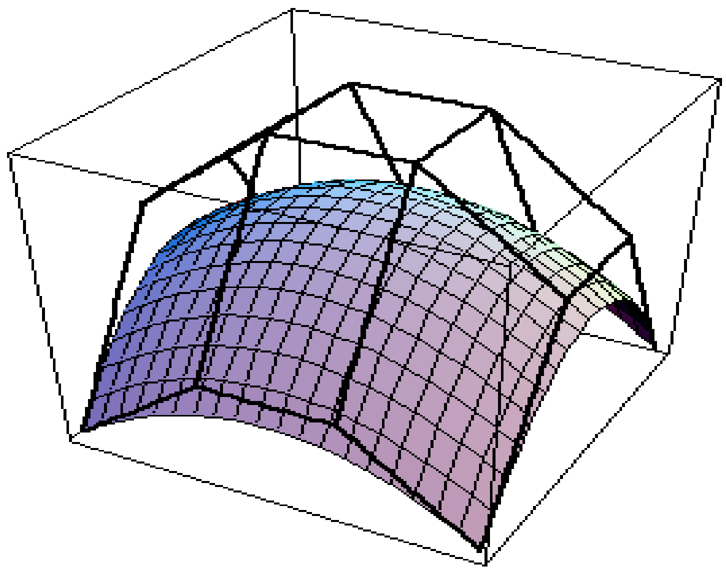
Tensor Product Bezier Surface

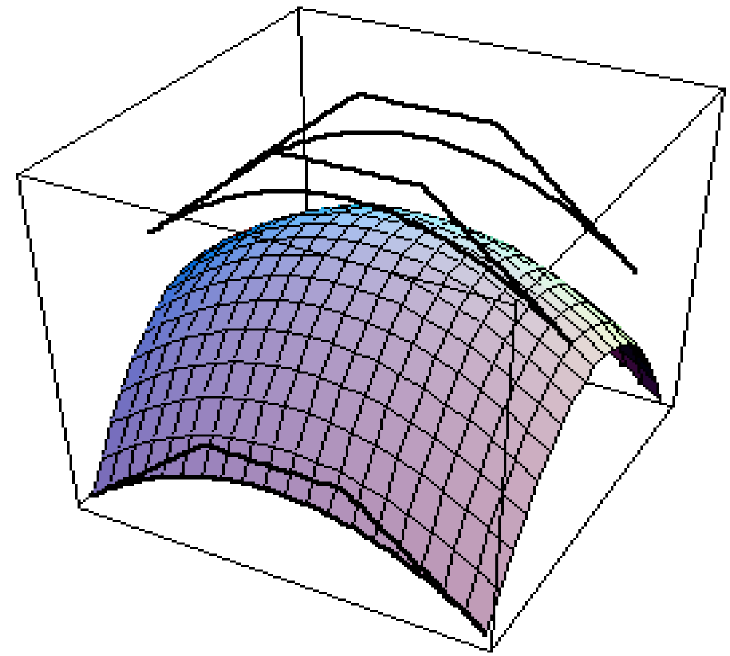


Tensor Product Bezier Surface

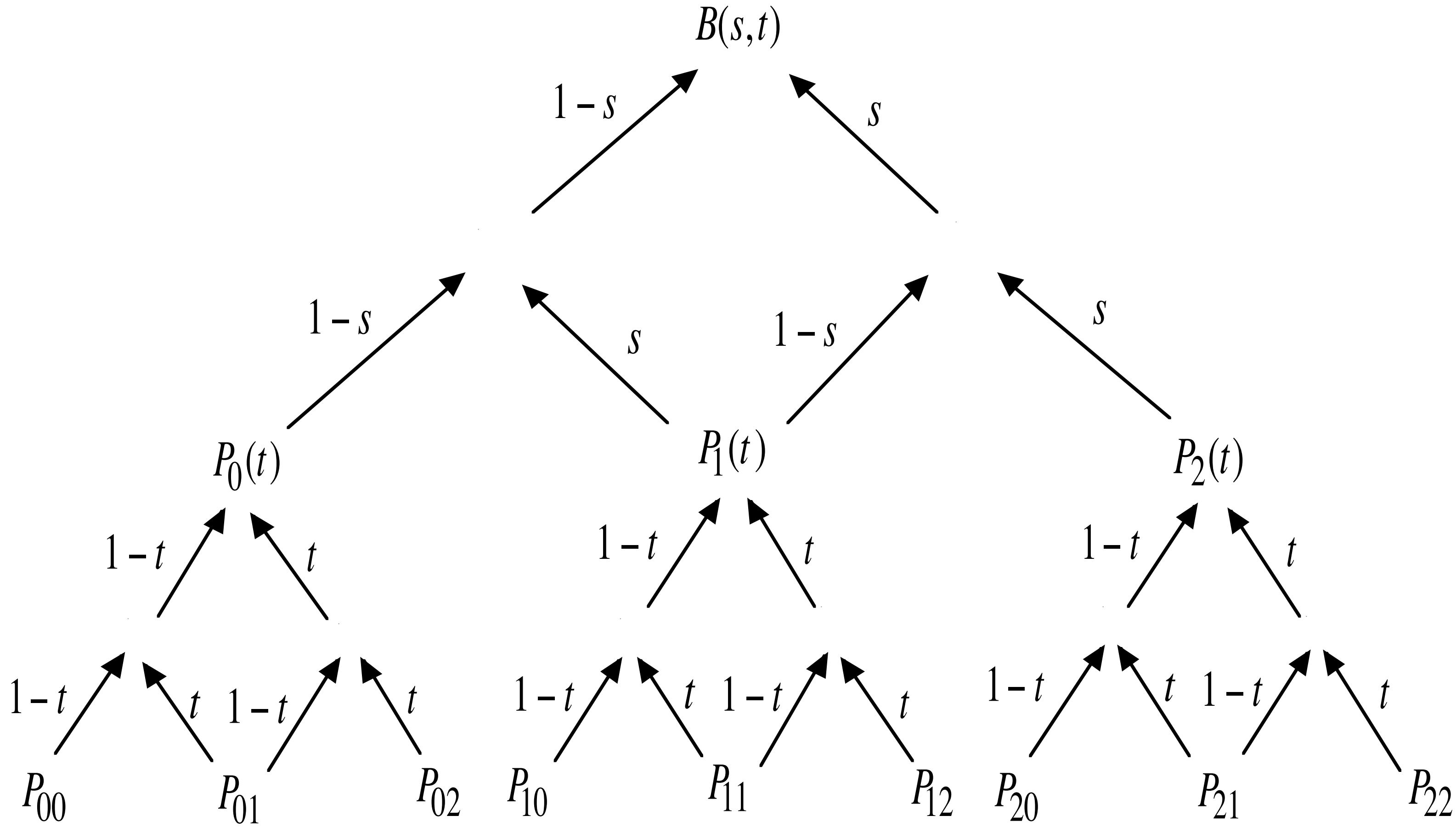


Tensor Product Bezier Surface

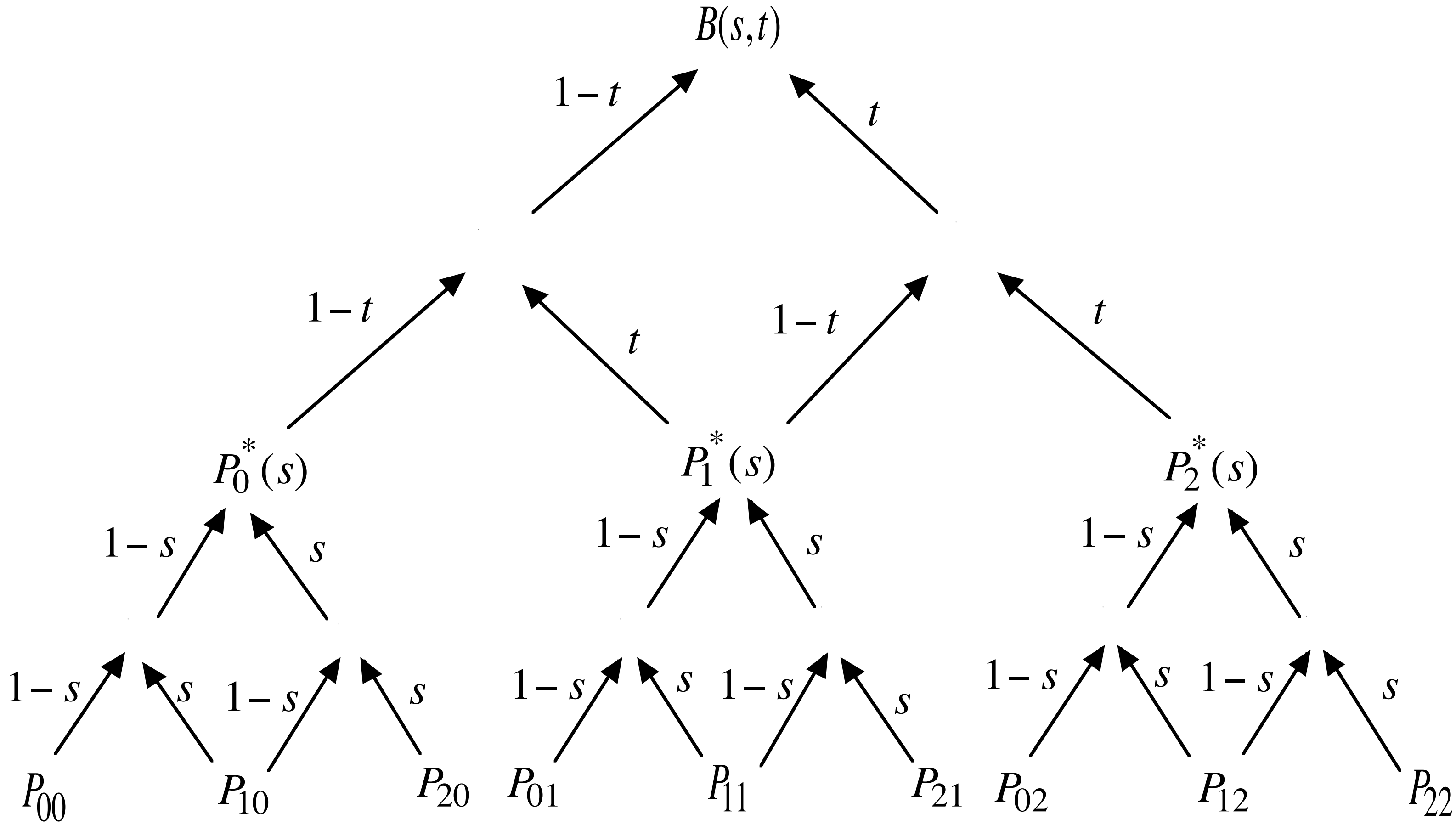




<u>De Casteljau's Algorithm for Tensor Product Bezier Surfaces</u>



De Casteljau's Algorithm for Tensor Product Bezier Surfaces



Elementary Properties of Tensor Product Bezier Surfaces

1. Polynomial

2. Translation Invariant

3. Lie in Convex Hull of their Control Points

4. Symmetry

5. Interpolation of Boundary Bezier Curves Defined by Boundary Control Points

Summary

Key Ideas

- de Casteljau's Algorithm
- Basic Properties of Bezier Curves
- Extension to Tensor Product Bezier Surfaces