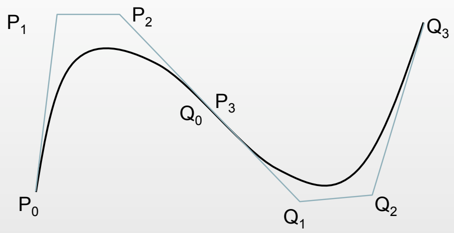
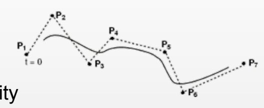
B-Splines

Bézier Curves – Joining Curve Segments



- Can only achieve C^1-continuity
- Need to pay attention to it explicitly!
  - Desirable: basis functions with continuity built-in

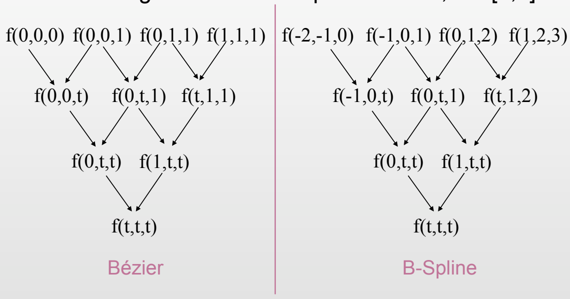
B-Splines



- Polynomial curves
- C^{k-1} continuity
  - Cubic B-spline: C^2 continuity
- Main idea:
  - Use blossom form as seen with Bézier
  - Instead of using fixed control points (for each segment) f(0,0,0), f(0,0,1), f(0,1,1), f(1,1,1)
  - Allow “sliding window” of control points, defined by knot vector, e.g.,
    f(-2,-1,0), f(-1,0,1), f(0,1,2), f(1,2,3)
    with knot vector [-2,-1,0,1,2,3]

Example

- Evaluating Bézier and B-Spline Curves, t in [0,1]



Knots

- A sequence of scalar values $t_1, \ldots, t_{2k}$ with $t_i \neq t_j$ if $i \neq j$, and $t_i < t_j$ for $i<j$ (k = degree of polyn.)
- If t_i chosen at uniform interval (such as 1,2,3, …), than it is a uniform knot sequence

Control Points

- We can define a unique k-degree polynomial F(t) with blossom f, such that $v_i = f(t_{i+1}, t_{i+2}, \ldots, t_{i+k})$
- The sequence of v_i for i [0,k] are the control points of a B-spline
- Evaluation of a point on a curve with f(t,t,t,…)
- Remark: no control points will lie on the curve!

Example: Degree 2 (k=2)

- Knots: t_1,t_2,t_3,t_4
- Control points
  $v_0 = f(t_1,t_2) = a_0 + a_1 (t_1+t_2)/2 + a_2 t_1t_2$
  $v_1 = f(t_2,t_3) = a_0 + a_1 (t_2+t_3)/2 + a_2 t_2t_3$
  $v_2 = f(t_3,t_4) = a_0 + a_1 (t_3+t_4)/2 + a_2 t_3t_4$

Example: Degree 3 (k=3)

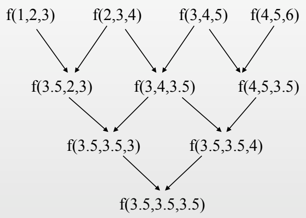
- Knots: t_1,t_2,t_3,t_4,t_5,t_6
- Control points
  $v_0 = f(t_1,t_2, t_3)$
  $v_1 = f(t_2,t_3, t_4)$
  $v_2 = f(t_3,t_4,t_5)$
  $v_3 = f(t_4,t_5,t_6)$

Concrete Example for Degree 3

- For the knot sequence 1,2,3,4,5,6
- Control points
  $v_0 = f(1,2,3)$
  $v_1 = f(2,3,4)$
  $v_2 = f(3,4,5)$
  $v_3 = f(4,5,6)$
- Find the point on the curve for t = 3.5

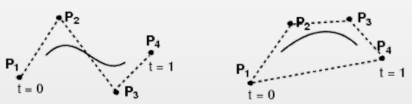
Concrete Example for Degree 3

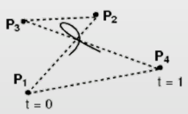
- For t = 3.5   (t must lie in [3,4])



Cubic B-Spline Example

- Curve is **not** constrained to pass through any control points





A B-Spline curve is also bounded by the convex hull of its control points.

More than One Segment

- Promised earlier on that there is automatic continuity
- Let’s see how this is achieved…

Definition

- Given a sequence of knots, $t_1,\ldots,t_{2k}$
- For each interval $[t_i, t_{i+1}]$, there is a k^{th}-degree parametric curve F(t) defined with corresponding B-spline control points

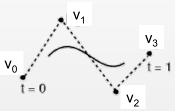
$$v_{i-k}, v_{i-k+1}, \ldots, v_i$$

(sliding window)

Definition

- If f() is the k-parameter blossom associated to the curve F(t) on $[t_i,t_{i+1}]$, then
  - The control points are defined by $v_j=f(t_{j+1}, \ldots, t_{j+k})$, j=i-k, i-k+1, …, i
  - The evaluation of the point on the curve at $t \in [t_i,t_{i+1}]$ is given by $F(t) = f(t,t,\ldots,t)$
  - Aside: the k-th degree Bézier curve corresponding to this curve has the control points:
    $p_j=f(\underbrace{t_i, t_i, \ldots, t_i}_{k-j},\underbrace{t_{i+1},t_{i+1},\ldots, t_{i+1}}_{j})$, j = 0, 1, …, k

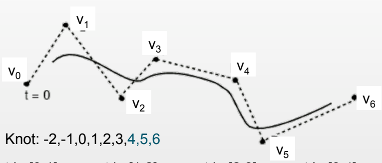
Example (Cubic B-Splines, k=3)



Knot: -2,-1,0,1,2,3

t in $[t_3,t_4]$=[0,1]:
v0 = f(-2,-1,0)
v1 = f(-1,0,1)
v2 = f(0,1,2)
v3 = f(1,2,3)

One Segment



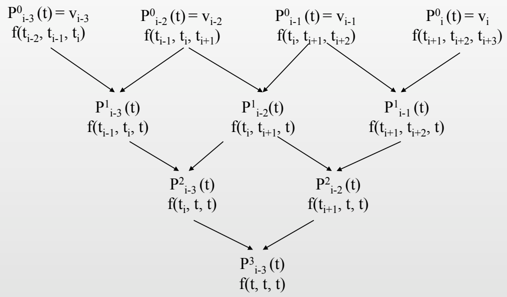
Knot: -2,-1,0,1,2,3,4,5,6

t in [0,1]:	t in [1,2]:	t in [2,3]:	t in [3,4]:
v0 = f(-2,-1,0)	v1 = f(-1,0,1)	v2 = f(0,1,2)	v3 = f(1,2,3)
v1 = f(-1,0,1)	v2 = f(0,1,2)	v3 = f(1,2,3)	v4 = f(2,3,4)
v2 = f(0,1,2)	v3 = f(1,2,3)	v4 = f(2,3,4)	v5 = f(3,4,5)
v3 = f(1,2,3)	v4 = f(2,3,4)	v5 = f(3,4,5)	v6 = f(4,5,6)

Multiple Segments

Computation – De Boor Algorithm

- Problem: computing a point on the B-spline for $t \in [t_i, t_{i+1}]$



De Boor Algorithm

- Recursion formula:

$$P_j^0(t) = v_j, j = i-k, i-k+1, \ldots, i$$

$$P_j^r(t) = \left(\frac{t_{k+1+j} - t}{t_{k+1+j} - t_{r+j}}\right) P_j^{r-1}(t) + \left(\frac{t - t_{r+j}}{t_{k+1+j} - t_{r+j}}\right) P_{j+1}^{r-1}(t), r = 1,2,\ldots,k, j = i-k, i-k+1, \ldots, i-r$$

- The required point on the curves is $P_{i-k}^k(t)$

Remarks for Cubic B-Splines

- For control points $v_0, v_1, \ldots, v_n$, the required knot sequence is $t_1, t_2, \ldots, t_{n+3}$
- The curve is defined over the range $t \in [t_3, t_{n+1}]$
- There will be n-2 curve segments altogether, since each interval $[t_i, t_{i+1}]$, i=3, 4, …, n defines a curve segment

B-Spline Basis

- For a k[th] degree B-spline

$$F(t) = \sum_{i=0}^{n} N_{k,i}(t) v_i$$

- Where the basis functions are

$$N_{0,i}(t) = \begin{cases} 1 & \text{if } t \in [t_i, t_{i+1}] \\ 0 & \text{otherwise} \end{cases}$$

$$N_{r,i}(t) = \left(\frac{t - t_i}{t_{i+r} - t_i}\right) N_{r-1,i}(t) + \left(\frac{t_{i+r+1} - t_i}{t_{i+r+1} - t_{i+1}}\right) N_{r-1,i+1}(t), \; t \in [t_i, t_{i+r+1}]$$

B-Spline Basis – Example Degree 1

- For k =1, $v_i = f(t_{i+1})$, $v_{i-1} = f(t_i)$, and the segment $t \in [t_i, t_{i+1}]$

$$F(t) = f(t) = \frac{t_{i+1} - t}{t_{i+1} - t_i} v_{i-1} + \frac{t - t_i}{t_{i+1} - t_i} v_i$$

- For k =1, $v_i = f(t_{i+1})$, $v_{i+1} = f(t_{i+2})$, and the segment $t \in [t_{i+1}, t_{i+2}]$

$$F(t) = f(t) = \frac{t_{i+2} - t}{t_{i+2} - t_{i+1}} v_i + \frac{t - t_{i+1}}{t_{i+2} - t_{i+1}} v_{i+1}$$

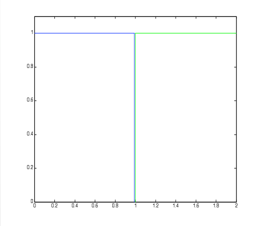
B-Spline Basis – Example Degree 1, contd.

- Then

$$N_{1,i}(t) = \left(\frac{t - t_i}{t_{i+1} - t_i}\right) N_{0,i}(t) + \left(\frac{t_{i+2} - t_i}{t_{i+2} - t_{i+1}}\right) N_{0,i+1}(t), \; t \in [t_i, t_{i+2}]$$

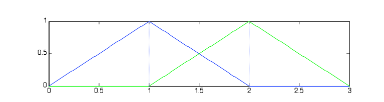
B-Spline Basis Functions – Visually

- $N_{0,i}$:

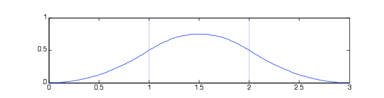


B-Spline Basis Functions – Visually

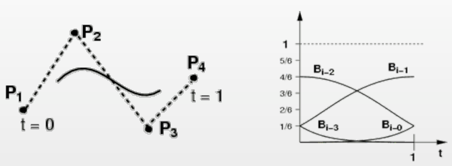
- $N_{1,i}$:



- $N_{2,i}$:



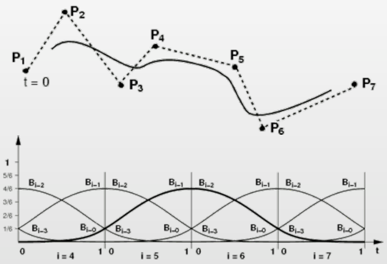
Cubic B-Splines – Basis Functions Visually



$$Q(t) = \frac{(1-t)^3}{6}P_{i-3} + \frac{3t^3 - 6t^2 + 4}{6}P_{i-2} + \frac{-3t^3 + 3t^2 + 3t + 1}{6}P_{i-1} + \frac{t^3}{6}P_i$$

Cubic B-Splines – Basis Functions Visually

- Chained together

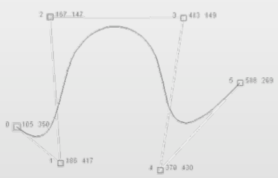


Knot Insertion

- Inserting new knots in the sequence while maintaining the B-spline curves can be used for
  - Rendering
  - Adding greater flexibility to the curve shape

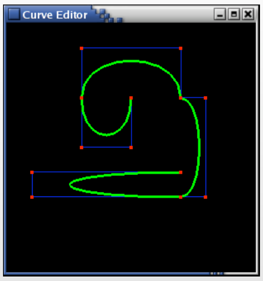
Multiple Knots

- Duplicating knots can force curve to go through a control point
- Clamped B-Spline goes through start/end point (multiplicity k+1 for start/end knot)
- Example:
  - Cubic B-Spline
  - Knot vector: [0 0 0 0 1 2 3 3 3 3]

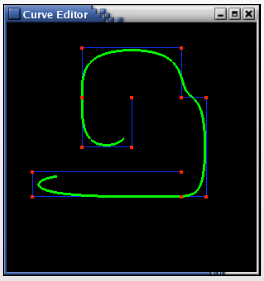


Bezier vs. B-Splines

Bézier is not the same as BSpline



Bézier

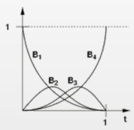


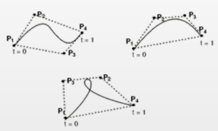
B-Spline

Bézier is not the same as BSpline

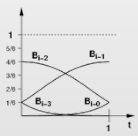
- Relationship to the control points is different

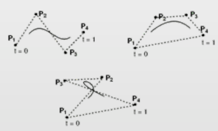
Bézier





BSpline



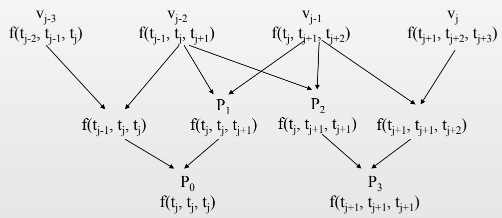


B-splines or Bézier curves?

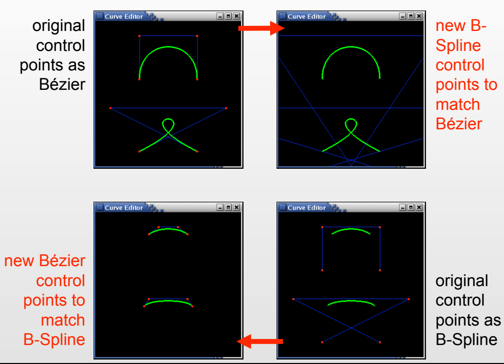
- Bézier curves are B-splines!
- But the control points are different
- You can find the Bézier control points from the B-spline control points
- In the case of a quadratic B-Spline:
  P_0 is an interpolation between v_{i-2} and v_{i-1},
  $P_1 = v_{i-1}$
  P_2 is an interpolation between v_{i-1} and v_i

Relation between 3rd-degree Bézier curves and B-Splines

- Constructing Bézier points from B-Spline points



Converting between Bézier & B-Spline



Advantages of B-Splines over Bézier curves

- The convex hull based on m control points is smaller than for Bézier curve
- There is a better local control
- The control points give a better idea of the shape of the curve