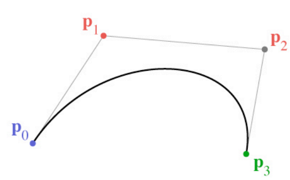
CS4620/5620: Lecture 28

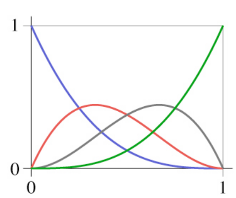
Splines

Announcements

- PPA2
  - Due on Friday Nov 18

Bézier basis



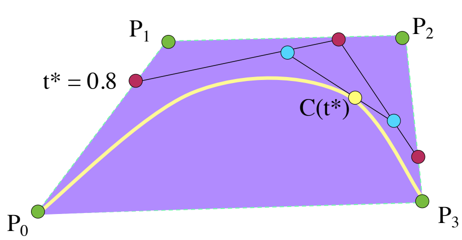


Chaining spline segments

- Hermite curves are convenient because they can be made long easily
- Bézier curves are convenient because their controls are all points and they have nice properties
  - and they interpolate every 4th point, which is a little odd
- We derived Bézier from Hermite by defining tangents from control points
  - a similar construction leads to the interpolating *Catmull-Rom* spline

Interpolation property

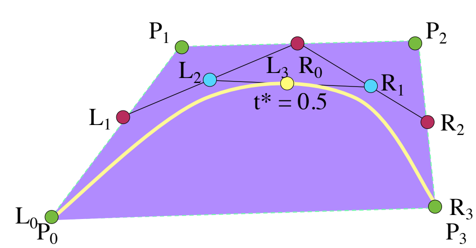
- $C(t^*)$ can be evaluated using interpolation
- $C(t) = (1-t)^3 P_0 + 3\,t\,(1-t)^2 P_1 + 3\,t^2 (1-t) P_2 + t^3 P_3$



9

De Casteljau algorithm

- Adaptive subdivision!

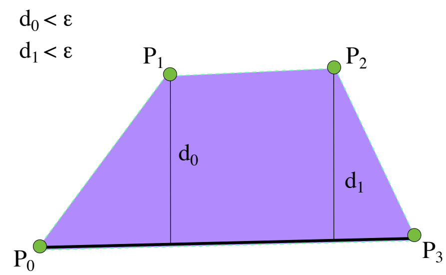


12

Recursive algorithm: de Casteljau algorithm

```
void DrawRecBezier (float eps) {
  if Linear (curve, eps)
    DrawLine (curve);
  else
    SubdivideCurve (curve, leftC, rightC);
    DrawRecBezier (leftC, eps);
    DrawRecBezier (rightC, eps);
}
```


Test for Linearity



Cubic Bézier splines

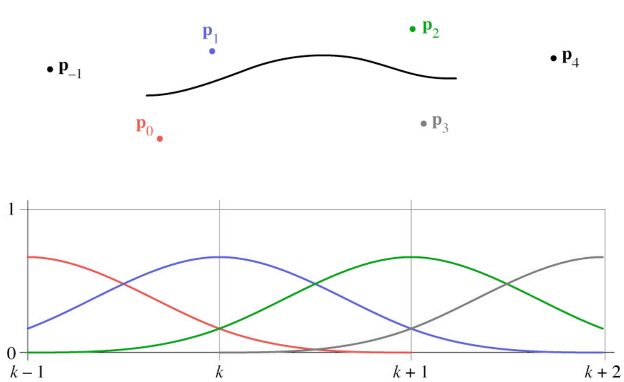
- Very widely used type, especially in 2D
  - e.g. it is a primitive in PostScript/PDF
- Can represent C^1 and/or G^1 curves with corners
- Can easily add points at any position

- Disadvantage
  - Special points
  - Only C^1

B-splines

- We may want more continuity than C^1
  - We may not need an interpolating spline
- B-splines are a clean, flexible way of making long splines with arbitrary order of continuity

Cubic B-spline basis



Deriving the B-Spline

- Approached from a different tack than Hermite-style constraints
  - Want a cubic spline; therefore 4 active control points
  - Want C^2 continuity
  - Turns out that is enough to determine everything

Efficient construction of any B-spline

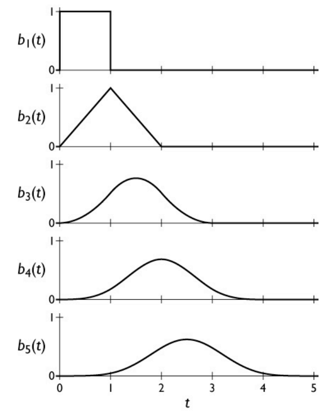
- B-splines defined for all orders
  - order *d*: degree *d* – 1
  - order *d*: *d* points contribute to value
- One definition: Cox-deBoor recurrence

$$b_1 = \begin{cases} 1 & 0 \le u < 1 \\ 0 & \text{otherwise} \end{cases}$$

$$b_d = \frac{t}{d-1} b_{d-1}(t) + \frac{d-t}{d-1} b_{d-1}(t-1)$$

B-spline construction, alternate view

- Recurrence
  - ramp up/down
- Convolution
  - smoothing of basis fn
  - smoothing of curve

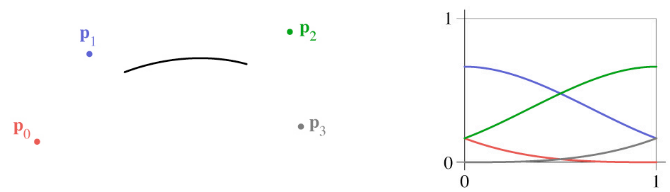


Cubic B-spline matrix

$$\mathbf{p}(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \cdot \frac{1}{6} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{p}_{k-1} \\ \mathbf{p}_k \\ \mathbf{p}_{k+1} \\ \mathbf{p}_{k+2} \end{bmatrix}$$

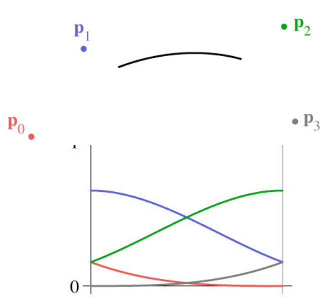
Cubic B-spline curves

- Treat points uniformly
- C^2 continuity
- $C(t) = [(1-t)^3 P_{i-3} + (3t^3 - 6t^2 + 4)P_{i-2} + (-3t^3 + 3t^2 + 3t + 1) P_{i-1} + t^3 P_i]/6$
- Notice blending functions still add to 1



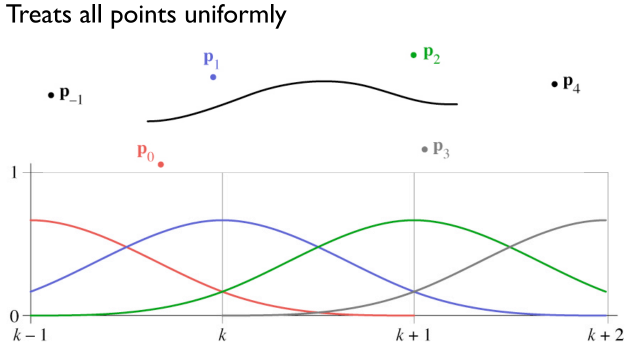
Cubic B-spline basis

- B-spline from each 4-point sequence matches previous, next sequence with C^2 continuity!
- Treats all points uniformly



Cubic B-spline basis

- B-spline from each 4-point sequence matches previous, next sequence with C^2 continuity!
- Treats all points uniformly

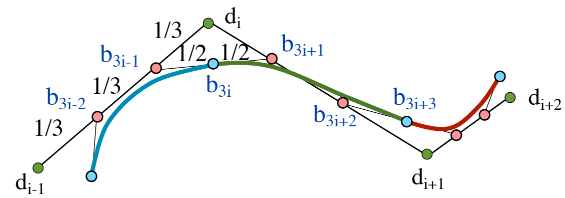


Evaluating splines for display

- Need to generate a list of line segments to draw
  - generate efficiently
  - use as few as possible
  - guarantee approximation accuracy
- Approaches
  - recursive subdivision (adaptive)
  - uniform sampling (easy to implement)

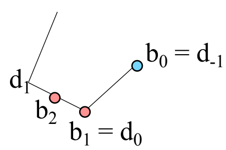
Rendering the spline-curve

- Given B-spline points $d_{-1}, \ldots d_{L+1}$
- Compute Bézier points $b_0, \ldots b_{3L}$
- Use De Casteljau algorithm to render



Equations and boundary conditions

- Equations
- $b_{3i} = (b_{3i-1}+b_{3i+1})/2$
- $b_{3i-1} = d_{i-1}/3 + 2d_i/3$
- $b_{3i-2} = 2d_{i-1}/3 + d_i/3$



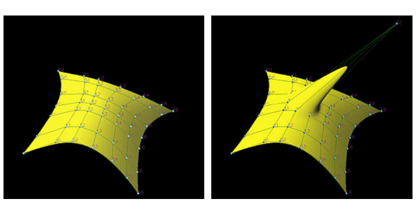
- Boundary conditions
  - $b_0 = d_{-1}$, $b_1 = d_0$, $b_2 = (d_0+d_1)/2$
  - $b_{3L} = d_{L+1}$, $b_{3L-1} = d_L$, $b_{3L-2} = (d_{L-1}+d_L)/2$

Other types of B-splines

- Nonuniform B-splines
  - discontinuities not evenly spaced
  - allows control over continuity or interpolation at certain points
  - e.g. interpolate endpoints (commonly used case)
- Nonuniform Rational B-splines (NURBS)
  - ratios of nonuniform B-splines: $x(t)/w(t)$, $y(t)/w(t)$
  - key properties:
    - invariance under perspective
    - ability to represent conic sections exactly

Surfaces

- Generalize by product of basis functions in 2 dimensions



Summary

- Splines
  - Smoothness, continuity (C0, C1, C2)
- Hermite
  - No convex hull property
  - 2 points and 2 tangents
- Bezier
  - Convex hull property
  - De Casteljau evaluation
  - Invariant to affine transformations
- B splines
  - Non-interpolation
  - C2