Exercise

$$B_{i,1}(u) = \begin{cases} 1 & u_i \le u \le u_{i+1} \\ 0 & \text{otherwise} \end{cases}$$

$$B_{i,d}(u) = \frac{u-u_i}{u_{i+d-1}-u_i} B_{i,d-1}(u) + \frac{u_{i+d}-u}{u_{i+d}-u_{i+1}} B_{i+1,d-1}(u)$$

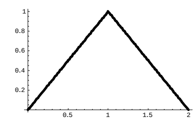
Compute $B_{0,2}$ if
a) the knot vector is $[0,1,2,3,4]$
b) the knot vector is $[0,0,1,2,3]$
c) is there a sense in tripling knot 0?

Solution (a)

$[u_0, u_1, u_2, u_3, u_4] = [0,1,2,3,4]$

Note that the degree of B is 1 ($=d-1$)

$$\begin{aligned} B_{0,2} &= \frac{u-u_0}{u_1-u_0} B_{0,1}(u) + \frac{u_2-u}{u_2-u_1} B_{1,1}(u) \\ &= uB_{0,1}(u) + (2-u)B_{1,1}(u) \\ &= u\mathbf{1}_{0\le u\le 1} + (2-u)\mathbf{1}_{1\le u\le 2} \\ &= \begin{cases} u & 0 \le u \le 1 \\ 2-u & 1 \le u \le 2 \end{cases} \end{aligned}$$

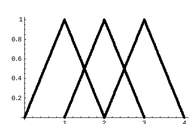


Solution (a)

Note that for this degree (1) and this knot vector ([0,1,2,3,4]) we have a uniform B-spline

Each point of the curve is linearly weighted between two points $\mathbf{P}$, which are interpolated

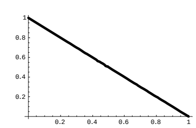
Valid u-range is [1,3]



Solution (b-c)

$[u_0, u_1, u_2, u_3, u_4] = [0,0,1,2,3]$

$$\begin{aligned} B_{0,2} &= \frac{u-u_0}{u_1-u_0} B_{0,1}(u) + \frac{u_2-u}{u_2-u_1} B_{1,1}(u) \\ &= 0 + (1-u)B_{1,1}(u) \\ &= (1-u)\mathbf{1}_{0\le u\le 1} \\ &= 1-u \qquad 0 \le u \le 1 \end{aligned}$$



Tripling the 0 knot will make $B_{0,2}$ zero, so no weight for $\mathbf{P}_0$

B-splines: recap of computation

$$\mathbf{Q}(u) = \sum_{k=0}^{n} \mathbf{P}_k B_{k,d}(u) \qquad u_{\min} \le u \le u_{\max}$$

n+1 # control points
d-1 degree of polynomials (d=4 for cubic)
B blending functions

- Knot vector $[u_i]$ is used to control interpolation properties of B-spline. Used to form the blending functions:

$$B_{i,1}(u) = \begin{cases} 1 & u_i \le u \le u_{i+1} \\ 0 & \text{otherwise} \end{cases}$$

$$B_{i,d}(u) = \frac{u-u_i}{u_{i+d-1}-u_i} B_{i,d-1}(u) + \frac{u_{i+d}-u}{u_{i+d}-u_{i+1}} B_{i+1,d-1}(u)$$

Non-uniform B-splines: summary

- Curve follows control point shape
- Curve lies in convex hull of control polygon
- Affine transformation of the curve can be done by affine transformation of the control points (invariance)
- Local control
- In general no point interpolation
- C^2 continuity if no multiple knots
- Degree independent of # control points

Like Bézier
Unlike Bézier

Interpolating splines

- Up until now, we have treated only *approximating* splines, that do not (in general) pass through the control points
- Interpolating cubic splines can be constructed by solving

$$\mathbf{Q}(u) = \begin{bmatrix} u^3 & u^2 & u & 1 \end{bmatrix} \begin{bmatrix} a_x & a_y & a_z \\ b_x & b_y & b_z \\ c_x & c_y & c_z \\ d_x & d_y & d_z \end{bmatrix}$$

$0 \le u \le 1$

Or

$$\begin{aligned} x(u) &= a_x u^3 + b_x u^2 + c_x u + d_x \\ y(u) &= a_y u^3 + b_y u^2 + c_y u + d_y \\ z(u) &= a_z u^3 + b_z u^2 + c_z u + d_z \end{aligned}$$

Between each pair of control points

- Coefficients follow from setting boundary conditions

61

Exercise

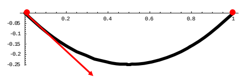
- Compute the 2D quadratic spline segment through the points $(0,0)$ and $(1,0)$ with boundary condition

$$\mathbf{Q}'(0) = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

62

Solution

$$\mathbf{Q}(u) = \begin{pmatrix} x(u) \\ y(u) \end{pmatrix} = \begin{pmatrix} a_x u^2 + b_x u + c_x \\ a_y u^2 + b_y u + c_y \end{pmatrix}$$



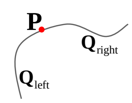
$$\mathbf{Q}'(u) = \begin{pmatrix} 2a_x u + b_x \\ 2a_y u + b_y \end{pmatrix}$$

$$\left.\begin{aligned} x(0) = 0 &\Rightarrow c_x = 0 \\ y(0) = 0 &\Rightarrow c_y = 0 \\ x'(0) = 1 &\Rightarrow b_x = 1 \\ y'(0) = -1 &\Rightarrow b_y = -1 \\ x(1) = 1 &\Rightarrow a_x + b_x + c_x = 1 \Rightarrow a_x = 0 \\ y(1) = 0 &\Rightarrow a_y + b_y + c_y = 0 \Rightarrow a_y = 1 \end{aligned}\right\} \mathbf{Q}(u) = \begin{pmatrix} u \\ u^2 - u \end{pmatrix}$$

Note that for four boundary conditions, a cubic spline is needed!

63

Interpolating splines: natural cubic splines



- $n+1$ control points, n curve sections → $4n$ coefficients
- At each of the n-1 internal points $\mathbf{P}$ we have 4 boundary conditions:
  - $\mathbf{Q}_{\text{left}}$ and $\mathbf{Q}_{\text{right}}$ pass through $\mathbf{P}$
  - $\mathbf{Q}_{\text{left}}$ and $\mathbf{Q}_{\text{right}}$ touch, and have identical first and second derivatives at $\mathbf{P}$
- Gives us $4n$-4 equations → we need 4 more
- Interpolation of $\mathbf{P}_0$ and $\mathbf{P}_\text{n}$ gives us 2
- Last 2 from

64

Cubic Hermite splines

- Natural splines have no local control and are expensive to compute
- If for each segment -apart from the end points- the *derivatives* (tangents) at the end points are known, we can compute each segment based only on these conditions: *Hermite* splines
- → local control and computation needs only one matrix multiplication per segment

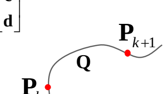
65

Hermite splines

$$\mathbf{Q}(u) = \begin{bmatrix} u^3 & u^2 & u & 1 \end{bmatrix} \begin{bmatrix} a_x & a_y & a_z \\ b_x & b_y & b_z \\ c_x & c_y & c_z \\ d_x & d_y & d_z \end{bmatrix} = \begin{bmatrix} u^3 & u^2 & u & 1 \end{bmatrix} \begin{bmatrix} \mathbf{a} \\ \mathbf{b} \\ \mathbf{c} \\ \mathbf{d} \end{bmatrix} = \mathbf{a}u^3 + \mathbf{b}u^2 + \mathbf{c}u + \mathbf{d}$$

$0 \le u \le 1$

$$\mathbf{Q}'(u) = 3\mathbf{a}u^2 + 2\mathbf{b}u + \mathbf{c} = \begin{bmatrix} 3u^2 & 2u & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{a} \\ \mathbf{b} \\ \mathbf{c} \\ \mathbf{d} \end{bmatrix}$$



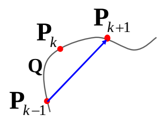
Fill in $\mathbf{P}_k$ (u=0) and $\mathbf{P}_{k+1}$ (u=1) and derivatives:

$$\begin{bmatrix} \mathbf{P}_k \\ \mathbf{P}_{k+1} \\ \mathbf{P}_k' \\ \mathbf{P}_{k+1}' \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{a} \\ \mathbf{b} \\ \mathbf{c} \\ \mathbf{d} \end{bmatrix} \Rightarrow \begin{bmatrix} \mathbf{a} \\ \mathbf{b} \\ \mathbf{c} \\ \mathbf{d} \end{bmatrix} = \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{P}_k \\ \mathbf{P}_{k+1} \\ \mathbf{P}_k' \\ \mathbf{P}_{k+1}' \end{bmatrix}$$

66

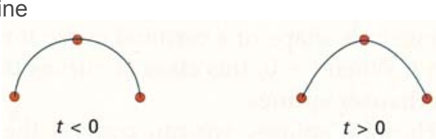
Cardinal splines

- Hermite splines require derivatives to be known $\longrightarrow$ often awkward
- *Cardinal* splines estimate the derivative from neighboring points:



$$\mathbf{P}_k' = \tfrac{1}{2}(1-t)(\mathbf{P}_{k+1} - \mathbf{P}_{k-1})$$

- t is a tension parameter. t=0 gives a *Catmull-Rom* spline



67

Exercise

- Give the expression for the cardinal spline segment $\mathbf{Q}$ between $\mathbf{P}_1$ and $\mathbf{P}_2$ for t=3 if

$\mathbf{P}_0 = (0,0)$
$\mathbf{P}_1 = (0,2)$
$\mathbf{P}_2 = (1,3)$
$\mathbf{P}_3 = (3,4)$

$$\mathbf{Q}(u) = \mathbf{a}u^3 + \mathbf{b}u^2 + \mathbf{c}u + \mathbf{d}$$

$$\mathbf{P}_k' = \tfrac{1}{2}(1-t)(\mathbf{P}_{k+1} - \mathbf{P}_{k-1})$$

$$\begin{bmatrix}\mathbf{a}\\ \mathbf{b}\\ \mathbf{c}\\ \mathbf{d}\end{bmatrix} = \begin{bmatrix}2 & -2 & 1 & 1\\ -3 & 3 & -2 & -1\\ 0 & 0 & 1 & 0\\ 1 & 0 & 0 & 0\end{bmatrix}\begin{bmatrix}\mathbf{P}_k\\ \mathbf{P}_{k+1}\\ \mathbf{P}_k'\\ \mathbf{P}_{k+1}'\end{bmatrix}$$

68

Solution

$$\mathbf{P}_k' = \tfrac{1}{2}(1-t)(\mathbf{P}_{k+1} - \mathbf{P}_{k-1})$$

$$\mathbf{P}_1' = \tfrac{1}{2}(-2)\begin{pmatrix}1\\3\end{pmatrix} = \begin{pmatrix}-1\\-3\end{pmatrix}$$

$$\mathbf{P}_2' = \tfrac{1}{2}(-2)\begin{pmatrix}3\\2\end{pmatrix} = \begin{pmatrix}-3\\-2\end{pmatrix}$$

$$\begin{bmatrix}\mathbf{a}\\ \mathbf{b}\\ \mathbf{c}\\ \mathbf{d}\end{bmatrix} = \begin{bmatrix}2 & -2 & 1 & 1\\ -3 & 3 & -2 & -1\\ 0 & 0 & 1 & 0\\ 1 & 0 & 0 & 0\end{bmatrix}\begin{bmatrix}\begin{pmatrix}0\\2\end{pmatrix}\\ \begin{pmatrix}1\\3\end{pmatrix}\\ \begin{pmatrix}-1\\-3\end{pmatrix}\\ \begin{pmatrix}-3\\-2\end{pmatrix}\end{bmatrix}$$

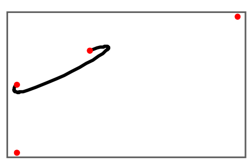
$$\mathbf{a} = 2\begin{pmatrix}0\\2\end{pmatrix} - 2\begin{pmatrix}1\\3\end{pmatrix} + \begin{pmatrix}-1\\-3\end{pmatrix} + \begin{pmatrix}-3\\-2\end{pmatrix} = \begin{pmatrix}-6\\-7\end{pmatrix}$$

$$\mathbf{b} = \begin{pmatrix}8\\11\end{pmatrix}$$

$$\mathbf{c} = \begin{pmatrix}-1\\-3\end{pmatrix}$$

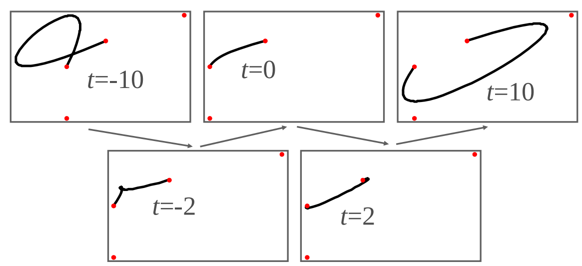
$$\mathbf{d} = \begin{pmatrix}0\\2\end{pmatrix}$$

$$\mathbf{Q}(u) = \mathbf{a}u^3 + \mathbf{b}u^2 + \mathbf{c}u + \mathbf{d}$$



Solution

For different tension values t:



Useful values are mostly $-1 \le t < 1$

70

Kochanek-Bartels splines

- Also known as TCB-splines
- Lightwave, 3ds Max (?)
- Extensions of Cardinal splines
- Derivative formula is altered:
  - Extra parameters *bias* and *continuity* next to tension
  - Derivative on the left and right of a control point need not match anymore
- Turns can be both smooth and sharp (useful for specifying animation paths)

71