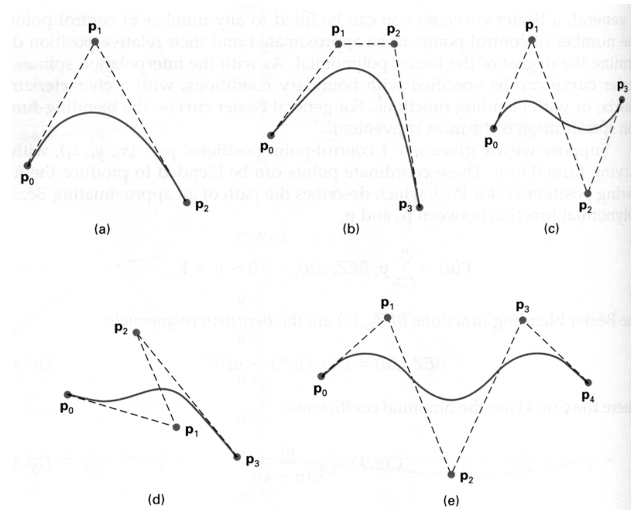
Bézier (H&B, page 432)

- Curve goes through two of the control points
- Curve is adjusted by moving two (cubic case) **other** control points
- Tangent at endpoints is in direction of adjacent control point
- Curve lies in convex hull of all 4 (cubic case) control points.

Example Bézier Curves



Bézier

- Geometry matrix based on Hermite case where
  - First two columns are endpoints
  - Next two are like derivatives from the Hermite case, but are now defined by

$$R_1 = 3(P_2 - P_1)$$
$$R_2 = 3(P_4 - P_3)$$

  - Note that this gives our condition on endpoint tangents
  - Factor of 3 gives good "balance" in control point effect, and is needed to be consistent with other derivations (e.g., Bernstein polynomials, subdivision, etc).

$$R_1 = 3(P_2 - P_1)$$
$$R_2 = 3(P_4 - P_3)$$

Means that

$$[P_1 \quad P_4 \quad R_1 \quad R_2] = [P_1 \quad P_2 \quad P_3 \quad P_4] \begin{bmatrix} 1 & 0 & -3 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & -3 \\ 0 & 1 & 0 & 3 \end{bmatrix}$$

Translation to Hermite case

M_{HB}

Recall Hermite

$$Q(t)=\begin{bmatrix}x(t)\\y(t)\\z(t)\end{bmatrix}=\begin{bmatrix}P_1 & P_4 & R_1 & R_2\end{bmatrix}M_H\begin{bmatrix}t^3\\t^2\\t\\1\end{bmatrix}$$

From previous slide

$$\begin{bmatrix}P_1 & P_4 & R_1 & R_2\end{bmatrix}=\begin{bmatrix}P_1 & P_2 & P_3 & P_4\end{bmatrix}M_{HB}$$

So, for Bézier

$$Q(t)=\begin{bmatrix}x(t)\\y(t)\\z(t)\end{bmatrix}=\begin{bmatrix}P_1 & P_2 & P_3 & P_4\end{bmatrix}M_{HB}M_H\begin{bmatrix}t^3\\t^2\\t\\1\end{bmatrix}$$

Want M_B in $Q(t)=\begin{bmatrix}x(t)\\y(t)\\z(t)\end{bmatrix}=\begin{bmatrix}P_1 & P_2 & P_3 & P_4\end{bmatrix}M_B\begin{bmatrix}t^3\\t^2\\t\\1\end{bmatrix}$

$$M_B=M_{HB}M_H$$

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

$$=\begin{bmatrix}-1 & 3 & -3 & 1\\3 & -6 & 3 & 0\\-3 & 3 & 0 & 0\\1 & 0 & 0 & 0\end{bmatrix}$$

Bézier in standard form (summary)

$$Q(t)=\begin{bmatrix}x(t)\\y(t)\\z(t)\end{bmatrix}=\begin{bmatrix}P_1 & P_2 & P_3 & P_4\end{bmatrix}M_B\begin{bmatrix}t^3\\t^2\\t\\1\end{bmatrix}$$

$$M_B=\begin{bmatrix}-1 & 3 & -3 & 1\\3 & -6 & 3 & 0\\-3 & 3 & 0 & 0\\1 & 0 & 0 & 0\end{bmatrix}$$

Skipped in 06

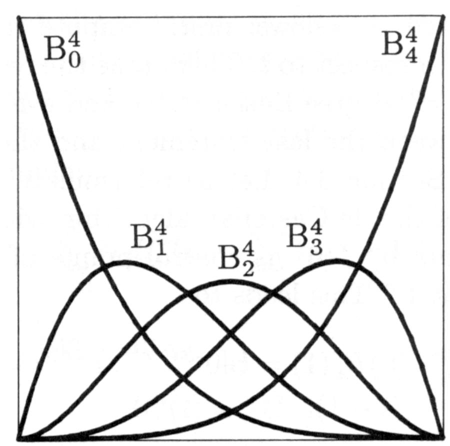
Bézier curves - II

- The blending functions are the Bernstein polynomials

$$c(t)=\sum_{i=0}^{n}p_iB_i^n(t)$$

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

Note, n is degree, was 3 before, now we are looking at the general case.



Skipped in 06

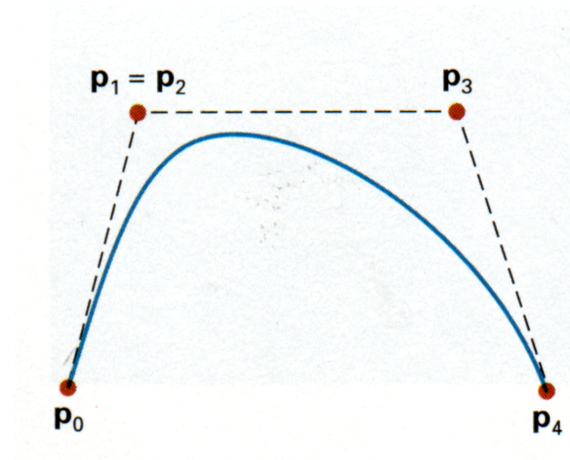
Bézier curves - III

- Bernstein polynomials have several important properties (over (0,1)).
  - they are postive and sum to 1, hence curve lies within convex hull of control points
  - curve interpolates its endpoints
  - curve's tangent at start lies along the vector from p_0 to p_1
  - tangent at end lies along vector from p_{n-1} to p_n

Skipped in 06

Bézier curve tricks - I

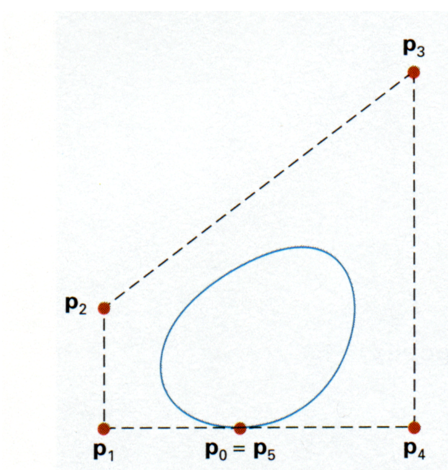
- "Pull" a curve toward a control point by doubling the control point



Skipped in 06

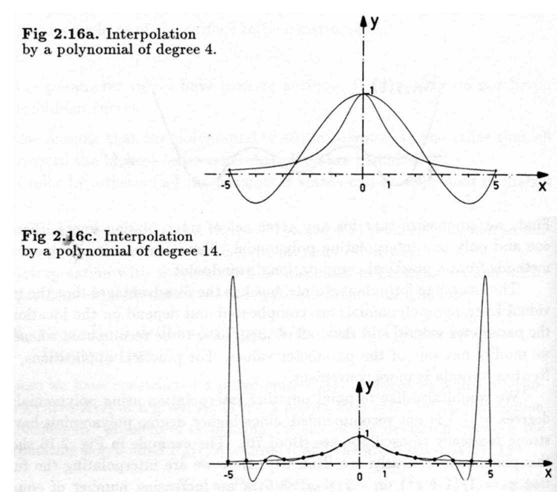
Bézier curve tricks-II

- Close the curve by making last point and first point coincident
  - curve has continuous tangent if first segment and last segment are collinear



Interpolating Splines (H&B page 420)

- Key idea:
  - high degree interpolates are badly behaved->
  - construct curves out of low degree segments



Interpolating Splines - II

- n+1 points;
- write derivatives X'
- X_i is spline for interval between P_i and P_{i+1}

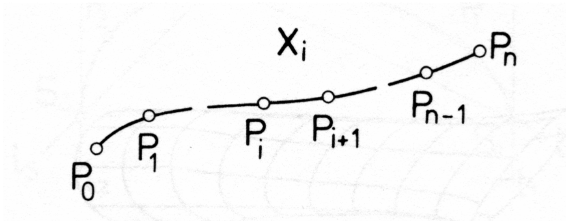


Fig. 3.11. The spline segment X_i.

Skipped in 06

Interpolating Splines - II

- Bolt together a series of Hermite curves with derivatives matching at joints (Knots).
- But where are the derivative values to come from?
  - Measurements
  - Combination of points (see cardinal splines--next topic)
  - Continuity considerations
  - Conventions for enpoints
- Cardinal splines

Equation Optional

$$P'_k = \left(\frac{1}{2}\right)(1-t)(P_{k+1} - P_{k-1})$$

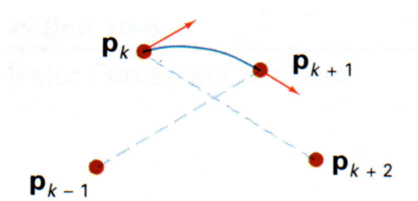
  - t is "tension"
  - still need to specify endpoint tangents
    - or use difference between first two, last two points

(Don't confuse t with parameter!)

Skipped in 06

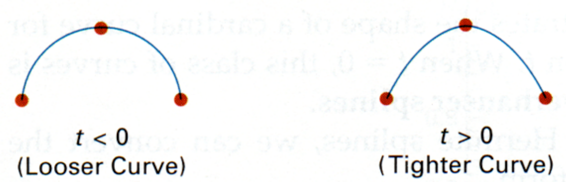
Details Optional

Tension



- larger values of tension give tighter curves (limit (as t-->1) is linear interpolate).

(Don't confuse t with parameter!)



Interpolating Splines

- Intervals:

$$a = t_0 < t_1 < t_2 < \cdots < t_{N-1} < t_N = b.$$

  - t values often called "knots"

$$\Delta t_i := t_{i+1} - t_i.$$

- Spline form:

$$\boldsymbol{X}_i(t) := \boldsymbol{A}_i(t-t_i)^3 + \boldsymbol{B}_i(t-t_i)^2 + \boldsymbol{C}_i(t-t_i) + \boldsymbol{D}_i, \quad t \in [t_i, t_{i+1}], \quad i = 0(1)N\text{–}1,$$

Continuity

- Require at endpoints:
  - endpoints equal
  - 1'st derivatives equal
  - 2'nd derivatives equal
- Now we get extra information from continuity (instead of tension equation, tangent measurements, etc)

$$\begin{aligned} \boldsymbol{X}_i(t_i) &= \boldsymbol{X}_{i-1}(t_i) \quad \text{or} \quad \boldsymbol{X}_i(t_{i+1}) = \boldsymbol{X}_{i+1}(t_{i+1}), \\ \boldsymbol{X}'_i(t_i) &= \boldsymbol{X}'_{i-1}(t_i) \quad \text{or} \quad \boldsymbol{X}'_i(t_{i+1}) = \boldsymbol{X}'_{i+1}(t_{i+1}), \\ \boldsymbol{X}''_i(t_i) &= \boldsymbol{X}''_{i-1}(t_i) \quad \text{or} \quad \boldsymbol{X}''_i(t_{i+1}) = \boldsymbol{X}''_{i+1}(t_{i+1}). \end{aligned}$$

Skipped in 06

Details Optional

- From endpoint and 1'st derivative:

$$\begin{aligned} \boldsymbol{X}_i(t_i) &= \boldsymbol{P}_i = \boldsymbol{D}_i, \quad \boldsymbol{X}_i(t_{i+1}) = \boldsymbol{P}_{i+1} = \boldsymbol{A}_i\Delta t_i^3 + \boldsymbol{B}_i\Delta t_i^2 + \boldsymbol{C}_i\Delta t_i + \boldsymbol{D}_i, \\ \boldsymbol{X}'_i(t_i) &= \boldsymbol{P}'_i = \boldsymbol{C}_i, \quad \boldsymbol{X}'_i(t_{i+1}) = \boldsymbol{P}'_{i+1} = 3\boldsymbol{A}_i\Delta t_i^2 + 2\boldsymbol{B}_i\Delta t_i + \boldsymbol{C}_i, \end{aligned}$$

- So that
- Yielding:

$$\begin{aligned} \boldsymbol{A}_i &= \frac{1}{(\Delta t_i)^3}[2(\boldsymbol{P}_i - \boldsymbol{P}_{i+1}) + \Delta t_i(\boldsymbol{P}'_i + \boldsymbol{P}'_{i+1})], \\ \boldsymbol{B}_i &= \frac{1}{(\Delta t_i)^2}[3(\boldsymbol{P}_{i+1} - \boldsymbol{P}_i) - \Delta t_i(2\boldsymbol{P}'_i + \boldsymbol{P}'_{i+1})]. \end{aligned}$$

$$\begin{aligned} \boldsymbol{X}_i(t) &= \\ \boldsymbol{P}_i &\left(2\frac{(t-t_i)^3}{(\Delta t_i)^3} - 3\frac{(t-t_i)^2}{(\Delta t_i)^2} + 1\right) + \boldsymbol{P}_{i+1}\left(-2\frac{(t-t_i)^3}{(\Delta t_i)^3} + 3\frac{(t-t_i)^2}{(\Delta t_i)^2}\right) \\ &+ \boldsymbol{P}'_i\left(\frac{(t-t_i)^3}{(\Delta t_i)^2} - 2\frac{(t-t_i)^2}{\Delta t_i} + (t-t_i)\right) + \boldsymbol{P}'_{i+1}\left(\frac{(t-t_i)^3}{(\Delta t_i)^2} - \frac{(t-t_i)^2}{\Delta t_i}\right) \end{aligned}$$

Skipped in 06

Details Optional

- Second Derivative:

$$\begin{aligned} X''_i(t) = 6P_i &\left(\frac{2(t-t_i)}{(\Delta t_i)^3} - \frac{1}{(\Delta t_i)^2}\right) + 6P_{i+1}\left(-2\frac{(t-t_i)}{(\Delta t_i)^3} + \frac{1}{(\Delta t_i)^2}\right) \\ &+ 2P'_i\left(3\frac{(t-t_i)}{(\Delta t_i)^2} - \frac{2}{\Delta t_i}\right) + 2P'_{i+1}\left(\frac{3(t-t_i)}{(\Delta t_i)^2} - \frac{1}{\Delta t_i}\right). \end{aligned}$$

- Want:

$$\boldsymbol{X}''_{i-1}(t_i) = \boldsymbol{X}''_i(t_i)$$

- Yielding:

$$\begin{aligned} \Delta t_i \boldsymbol{P}'_{i-1} &+ 2(\Delta t_{i-1} + \Delta t_i)\boldsymbol{P}'_i + \Delta t_{i-1}\boldsymbol{P}'_{i+1} \\ &= 3\frac{\Delta t_{i-1}}{\Delta t_i}(\boldsymbol{P}_{i+1} - \boldsymbol{P}_i) + 3\frac{\Delta t_i}{\Delta t_{i-1}}(\boldsymbol{P}_i - \boldsymbol{P}_{i-1}). \end{aligned}$$

Missing equations

- Recurrence relations represent d(n-1) equations in d(n+1) unknowns (d is dimension)
- We need to supply the derivative at the start and at the finish (or two equivalent constraints)
- Options:
  - second derivatives vanish at each end (natural spline)
  - give slopes at the boundary
    - vector from first to second, second last to last
    - parabola through first three, last three points
  - third derivative is the same at first, last knot

B-splines - I (H&B page 442)

- Now consider stitching together curves which do not necessarily pass through the control points.
- Stitching --> Local control
- Blending functions are non-zero over limited range--thus they are like "switches"
- In the simplest case of uniformly spaced control points, the blending functions will be shifted versions of the same function.

B-splines - II

- Curve (general case):

$$X(t) = \sum_{k=0}^{n} P_k B_{k,d}(t)$$

- The "degree parameter" d is:

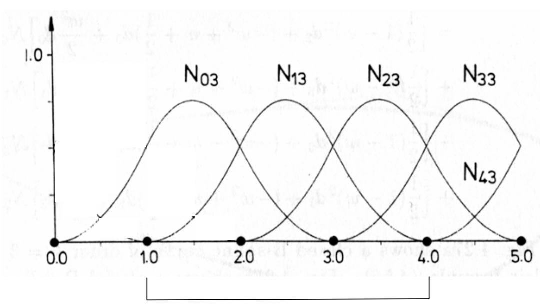
$$2 \le d \le n+1$$

- The "degree" of the polynomial is **d-1** (not d).
- We assume we have n+1 control points, and n+1 blending functions.
- The most common case is d=4 (cubic).

B-Spline Blending Functions

- Knots
  - parameter values where curve segments meet
- An "degree" d blending function blends d control points
- Hence each one has d+1 knots including the endpoints

d=3



B-Spline Blending Functions

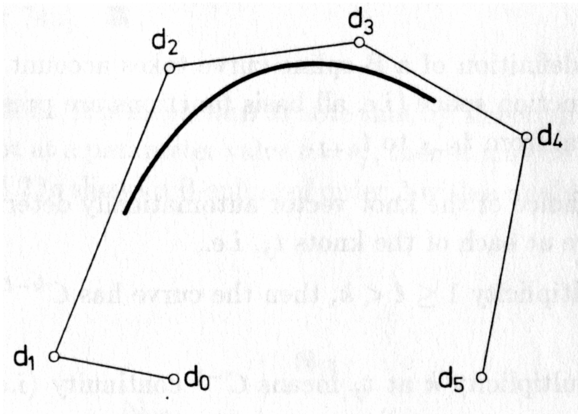
- One control point means d knots. .
- n+1 control points means n+d+1 knots

$$(t_0, t_1, \ldots, t_{n+d})$$

where $t_0 \le t_1 \le \ldots \le t_{n+d}$

- But we only use the sections where we blend d control points (ignore some begining and end knots).
- Thus the B-spline is defined for the range

$$(t_{d-1}, \ldots, t_{n+1})$$



http://www.siggraph.org/education/materials/HyperGraph/modeling/splines/demoprog/curve.html

Skipped in 06

Details Optional

B-Spline Blending Functions

- Blending functions (recursive definition)

$$B_{k,1}(t)=\begin{cases}1 & t_k \le t \le t_{k+1}\\ 0 & \text{otherwise}\end{cases}$$

$$B_{k,d}(t)=\left(\frac{t-t_k}{t_{k+d-1}-t_k}\right)B_{k,d-1}(t)+\left(\frac{t_{k+d}-t}{t_{k+d}-t_{k+1}}\right)B_{k+1,d-1}(t)$$

- If knots are repeated we use 0/0=0

Skipped in 06

Details Optional

B-Spline Blending Functions

$$B_{k,d}(t)=\left(\frac{t-t_k}{t_{k+d-1}-t_k}\right)B_{k,d-1}(t)+\left(\frac{t_{k+d}-t}{t_{k+d}-t_{k+1}}\right)B_{k+1,d-1}(t) \qquad B_{k,1}(t)=\begin{cases}1 & t_k \le t \le t_{k+1}\\ 0 & \text{otherwise}\end{cases}$$

So, assuming uniformly spaced knots,

$$B_{k,2}(t)= \quad ?$$

Skipped in 06

Details Optional

B-Spline Blending Functions

$$B_{k,d}(t)=\left(\frac{t-t_k}{t_{k+d-1}-t_k}\right)B_{k,d-1}(t)+\left(\frac{t_{k+d}-t}{t_{k+d}-t_{k+1}}\right)B_{k+1,d-1}(t)$$

$$B_{k,1}(t)=\begin{cases}1 & t_k \le t \le t_{k+1}\\ 0 & \text{otherwise}\end{cases}$$

So, assuming uniformly spaced knots,

$$B_{k,2}(t)=\begin{cases}\left(\dfrac{t-t_k}{\Delta t}\right) & t_k \le t \le t_{k+1}\\ \left(\dfrac{t_k-t}{\Delta t}\right) & t_{k+1} \le t \le t_{k+2}\\ 0 & \text{otherwise}\end{cases}$$

Skipped in 06

Details Optional

B-Spline Blending Functions

$$B_{k,d}(t)=\left(\frac{t-t_k}{t_{k+d-1}-t_k}\right)B_{k,d-1}(t)+\left(\frac{t_{k+d}-t}{t_{k+d}-t_{k+1}}\right)B_{k+1,d-1}(t)$$

$$B_{k,2}(t)=\begin{cases}\left(\dfrac{t-t_k}{\Delta t}\right) & t_k \le t \le t_{k+1}\\ \left(\dfrac{t_k-t}{\Delta t}\right) & t_{k+1} \le t \le t_{k+2}\\ 0 & \text{otherwise}\end{cases}$$

So, assuming uniformly spaced knots,

$$B_{k,3}(t)= \ ?$$

Skipped in 06

Details Optional

B-Spline Blending Functions

$$B_{k,d}(t)=\left(\frac{t-t_k}{t_{k+d-1}-t_k}\right)B_{k,d-1}(t)+\left(\frac{t_{k+d}-t}{t_{k+d}-t_{k+1}}\right)B_{k+1,d-1}(t)$$

$$B_{k,2}(t)=\begin{cases}\left(\dfrac{t-t_k}{\Delta t}\right) & t_k \le t \le t_{k+1}\\ \left(\dfrac{t_k-t}{\Delta t}\right) & t_{k+1} \le t \le t_{k+2}\\ 0 & \text{otherwise}\end{cases}$$

So, assuming uniformly spaced knots,

$$B_{k,3}(t)=\begin{cases}\text{quadratic function} & t_k \le t \le t_{k+1}\\ \text{quadratic function} & t_{k+1} \le t \le t_{k+2}\\ \text{quadratic function} & t_{k+2} \le t \le t_{k+3}\\ 0 & \text{otherwise}\end{cases}$$

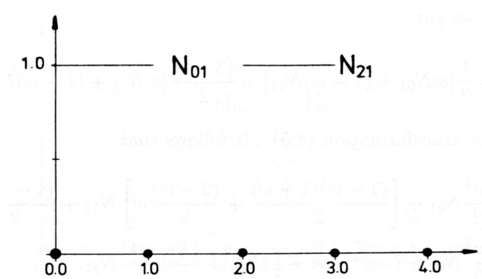


Fig. 4.22c. The B-splines N_{01}, N_{21}.

These figures show blending functions with a uniform knot vector, knots at 0, 1, 2, etc. Note that N is the same as our B

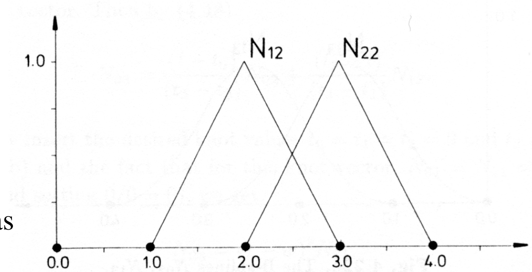


Fig. 4.22d. The B-splines N_{12}, N_{22}.

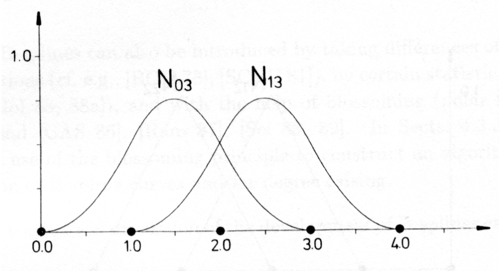
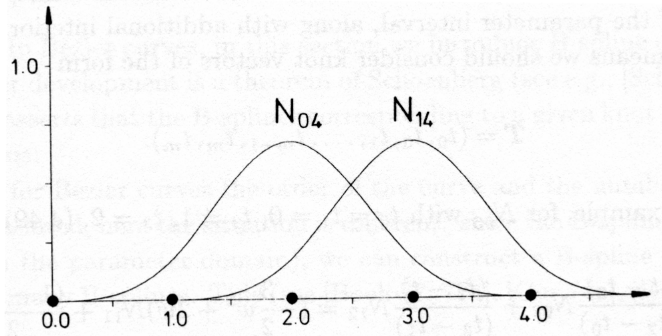


Fig. 4.22e. The B-splines N_{03}, N_{13}.



Skipped in 06

Details Optional

Matrix form of Uniform Cubic B-Spline Blending Functions

$$M_B = \frac{1}{6}\begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix}$$

Closed B-Splines

- Extend the control points and the knots to “wrap around”.

$$P_{n+1} = P_0$$
$$t_{n+1} = t_0$$

Fig. 4.26a.

Fig. 4.26b.

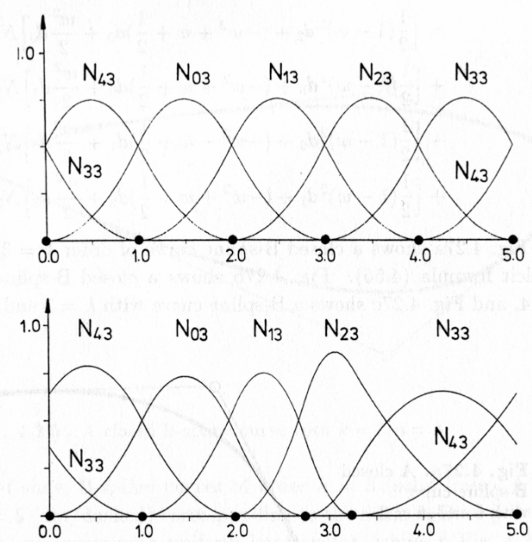


Fig. 4.26. B-splines with uniform and non-uniform knot vectors for a closed B-spline curve.

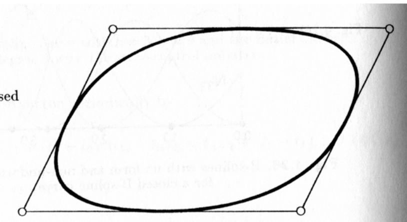


Fig. 4.27a. A closed B-spline curve with $k = 3, n = 3$.

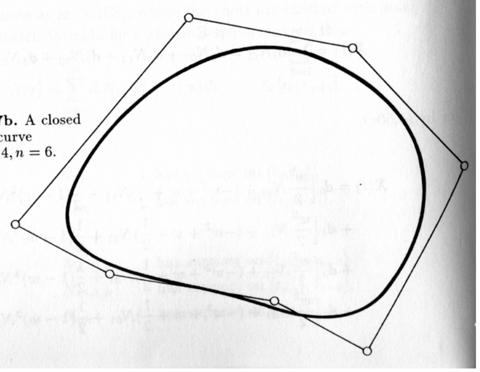


Fig. 4.27b. A closed B-spline curve with $k = 4, n = 6$.

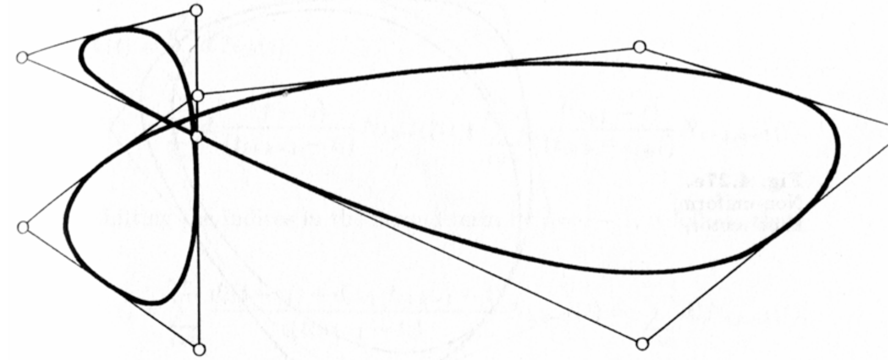
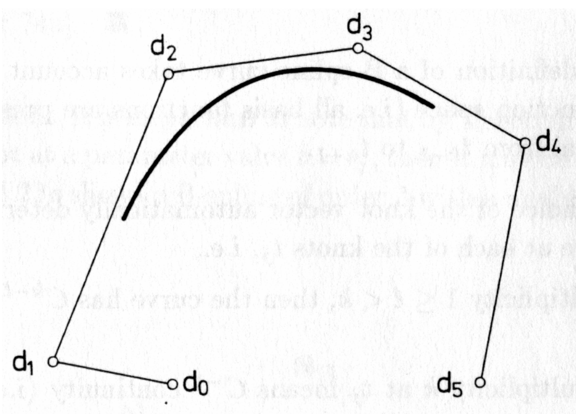


Fig. 4.27c. A closed B-spline curve with $k = 3, n = 8$.



Recall that each curve section is a blend of d control points.

What if we want to interpolate the endpoints?

Repeated knots

- Definition works for repeated knots (if we are understanding about 0/0)
- Repeated knot reduces continuity. A B-spline blending function has continuity C^{d-2}; if the knot is repeated m times, continuity is now C^{d-m-1}
- e.g. -> quadratic B-spline (i.e. order 3) with a double knot

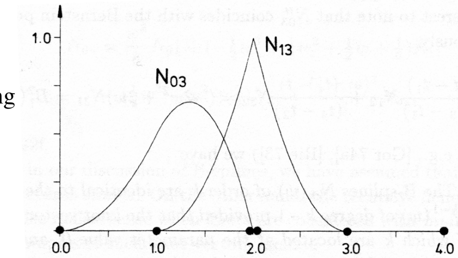


Fig. 4.22g. A quadratic B-spline with a double knot.

Most useful case

- Select the first d and the last d knots to be the same
  - we then get the first and last points lying on the curve
  - also, the curve is tangent to the first and last segment
- E.g. cubic case below
- Notice that a control point influences at most d parameter intervals - **local control**

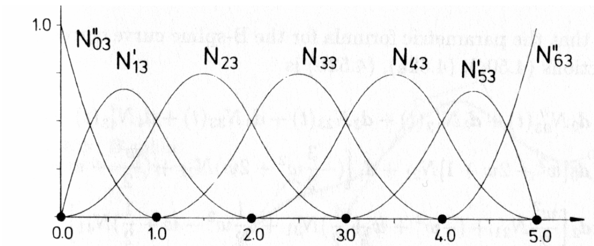


Fig. 4.24a. B-splines for an open B-spline curve with uniform knot vector.

Fig. 4.25a. B-spline curve with $k = 3$, $n = 5$.

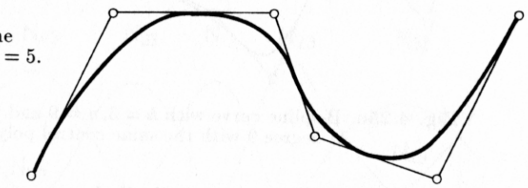
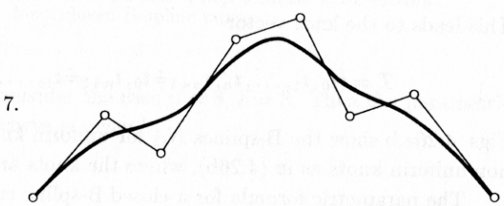


Fig. 4.25b. B-spline curve with $k = 4$, $n = 7$.



k is our d - top curve has order 3, bottom order 4

Example of blending function with repeated knots at the endpoints and non-uniform spacing of interior knots

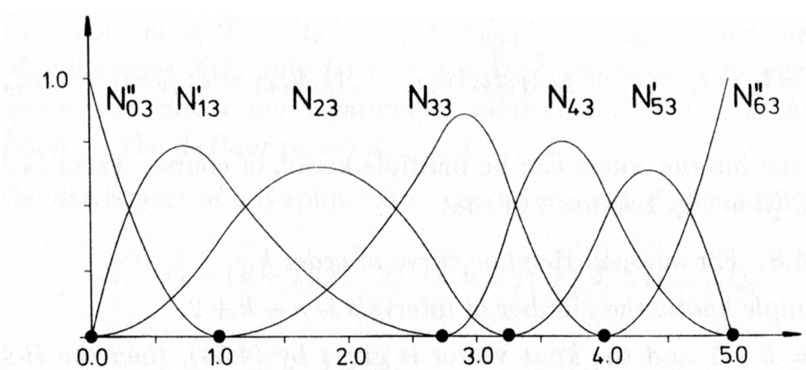


Fig. 4.24b. B-splines for an open B-spline curve with non-uniform knot vector.

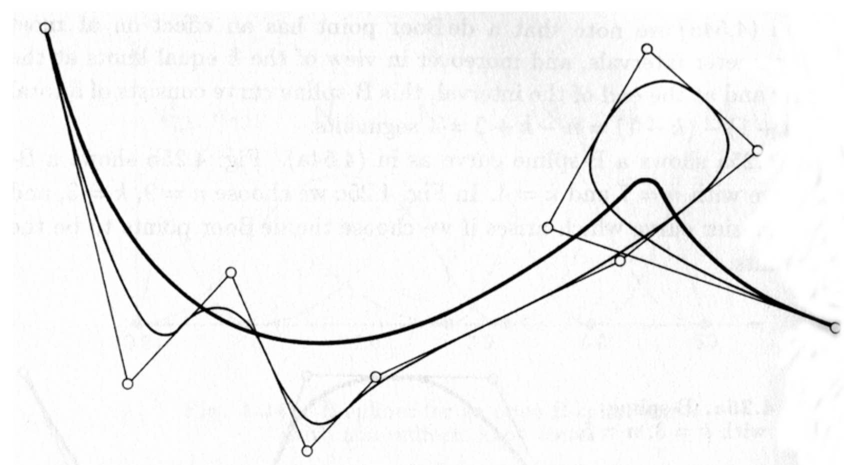


Fig. 4.25c. B-spline curve with $k = 3, n = 9$ and the Bézier curve of degree 9 with the same control polygon.

Bézier curve is the heavy curve

B-Spline properties

- For a B-spline curve of order d
  - if m knots coincide, the curve is C^{d-m-1} at the corresponding point
  - if d-1 consecutive* points of the control polygon are collinear, then the curve is tangent to the polygon
  - if d consecutive* points of the control polygon are collinear, then the curve and the polygon have a common segment
  - if d-1 points coincide, then the curve interpolates the common point and the two adjacent sides of the polygon are tangent to the curve
  - each segment of the curve lies in the convex hull of the associated d points

*The fish shaped curve a few slides back have 4 collinear points (d=3), but they are not consecutive so the condition does not hold.