Central issues in modelling

- Construct families of curves, surfaces and volumes that
  - can represent common objects usefully;
  - are easy to interact with; interaction includes:
    - manual modelling;
    - fitting to measurements;
  - support geometric computations
    - intersection
    - collision

- Main topics:
  - curves
  - surfaces
  - volumes
  - deformation
- Simple curves
- Simple surfaces
- Continuity and splines
- Bezier surfaces and spline surfaces
- Volume models
- Animation

Parametric vs Implicit

- A parametric curve is given as a function of parameters Examples:
  - circle as (cos t, sin t)
  - twisted cubic as (t, t*t, t*t*t)
- A parametric surface is given as a function of parameters. Examples:
  - sphere as (cos s cos t, sin s cos t, sin t)
- Advantage - easy to compute normal, easy to render, easy to put patches together.
- Disadvantage - ray tracing is hard

- An implicit curve is given by the vanishing of some functions
  - circle on the plane, x*x +y*y-r*r=0
  - twisted cubic in space, x*y-z=0, x*z-y*y=0, x*x-y=0
- An implicit surface is given by the vanishing of some functions
  - sphere in space x*x+y*y +z*z-r*r=0
  - plane a x+ b y + c z+d=0

Interpolation

- Construct a parametric curve that passes through (interpolates) a set of points.
- Lagrange interpolate:
  - give parameter values associated with each point
  - use Lagrange polynomials (one at the relevant point, zero at all others) to construct curve
  - curve is:

$$\sum_{i \in \text{points}} p_i \phi_i^{(l)}(t)$$

- degree is (#pts-1)
  - e.g. line through two points
  - quadratic through three.
- Functions phi are known as "blending functions"

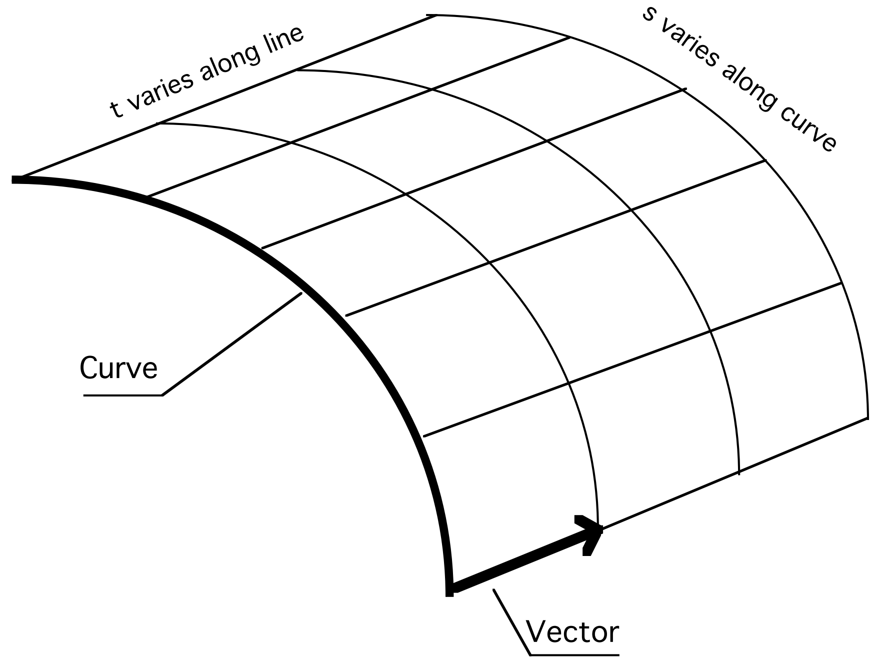
Hermite curves

- Hermite interpolate
  - give parameter values and derivatives associated with each point
  - curve passes through given point and the given derivative at that parameter value
  - curve is:
- use Hermite polynomials to construct curve
  - one at some parameter value and zero at others or
  - derivative one at some parameter value, and zero at others

$$\sum_{i \in \text{points}} p_i \phi_i^{(h)}(t) + \sum_{i \in \text{points}} v_i \phi_i^{(hd)}(t)$$

Extruded surfaces

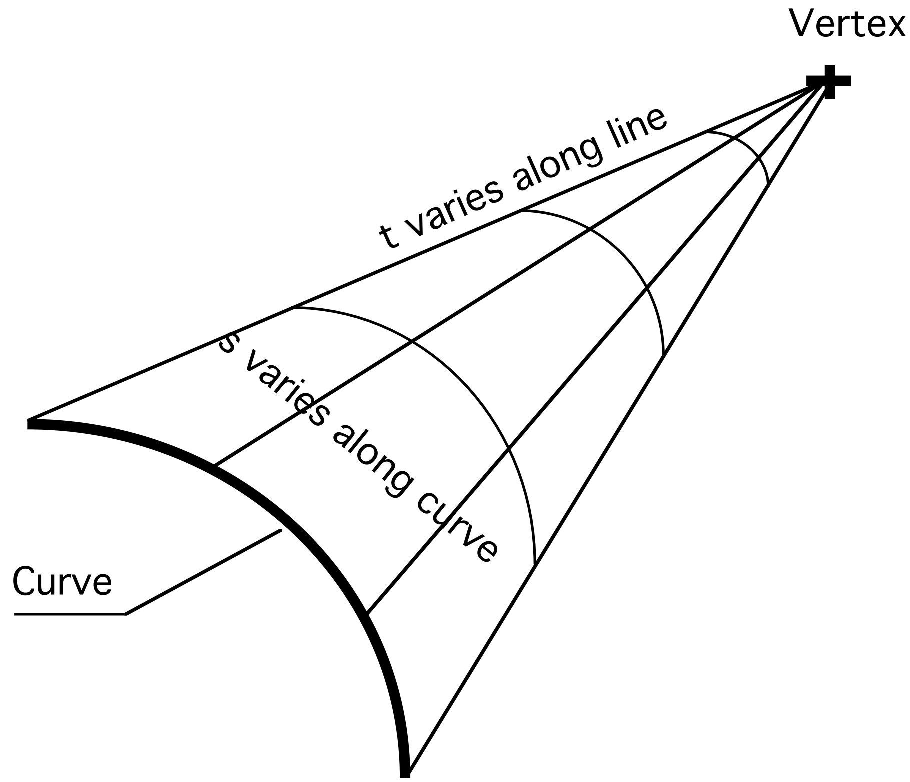
- Geometrical model - Pasta machine
- Take curve and "extrude" surface along vector
- Many human artifacts have this form - rolled steel, etc.



$$(x(s,t), y(s,t), z(s,t)) = (x_c(s), y_c(s), z_c(s)) + t(v_0, v_1, v_2)$$

Cones

- From every point on a curve, construct a line segment through a single fixed point in space - the vertex
- Curve can be space or plane curve, but shouldn't pass through the vertex



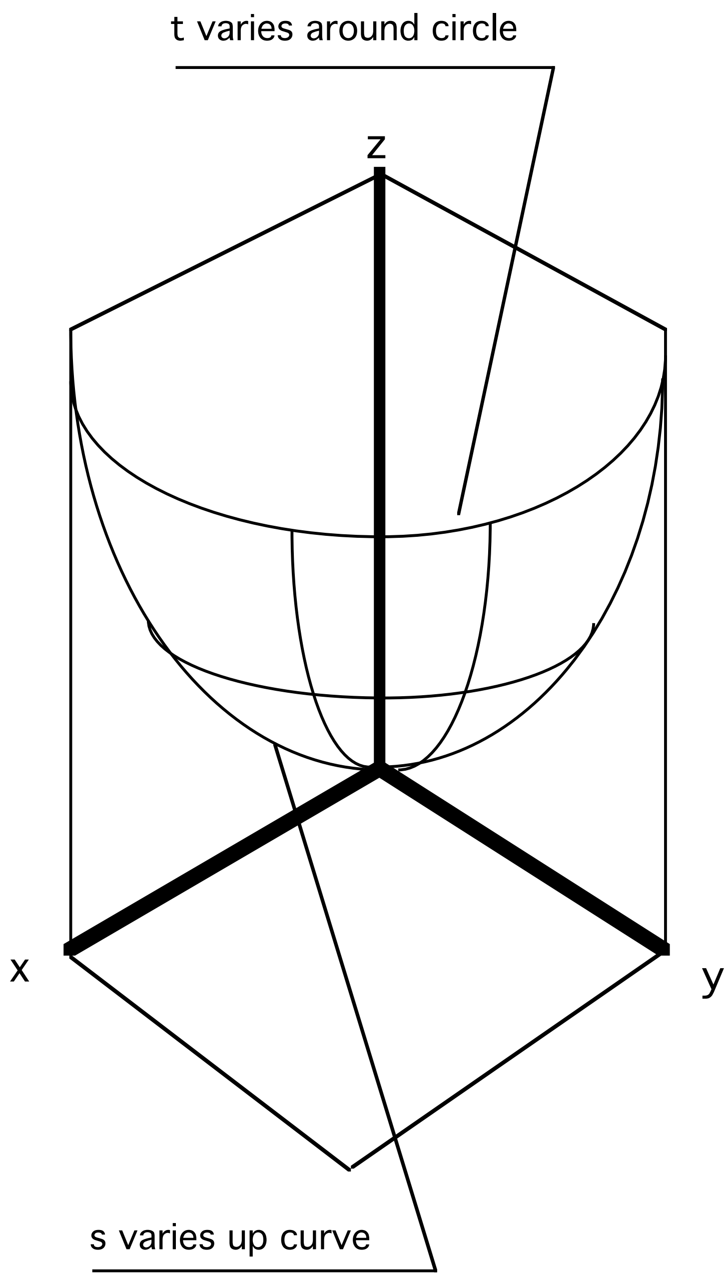
$$(x(s,t), y(s,t), z(s,t)) = (1-t)(x_c(s), y_c(s), z_c(s)) + t(v_0, v_1, v_2)$$

Surfaces of revolution - 1

- Plane curve + axis
- "spin" plane curve around axis to get surface
- Choice of plane is arbitrary, choice of axis affects surface
- In this case, curve is on x-z plane, axis is z axis.

$$(x(s,t), y(s,t), z(s,t)) =$$

$$(x_c(s)\cos(t), x_c(s)\sin(t), z_c(s))$$



Surfaces of revolution -2

Many artifacts are SOR's, as they're easy to make on a lathe.

Controlling is quite easy - concentrate on the cross section.

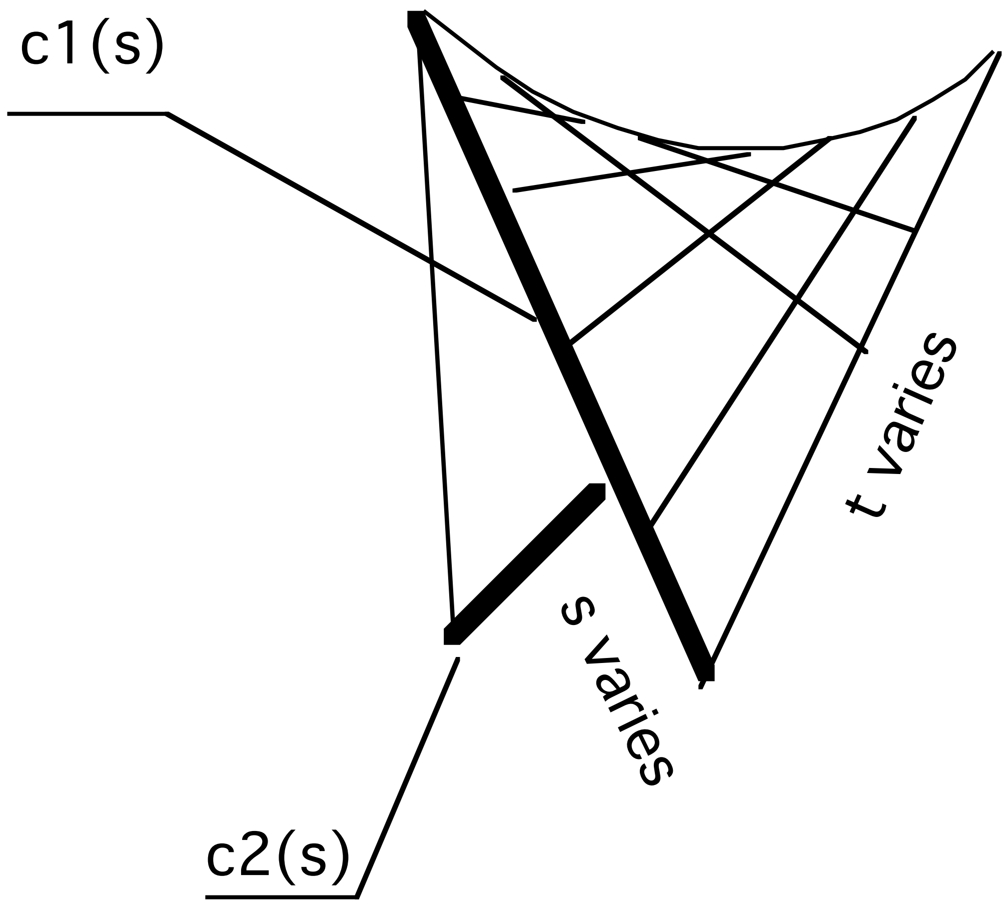
Axis crossing cross-section leads to ugly geometry.

Ruled surfaces -1

- Popular, because it's easy to build a curved surface out of straight segments - eg pavilions, etc.
- Take two space curves, and join corresponding points - same s - with line segment.
- Even if space curves are lines, the surface is usually curved.

$$(x(s,t), y(s,t), z(s,t)) =$$
$$(1-t)(x_1(s), y_1(s), z_1(s)) +$$
$$t(x_2(s), y_2(s), z_2(s))$$

Ruled Surfaces - 2

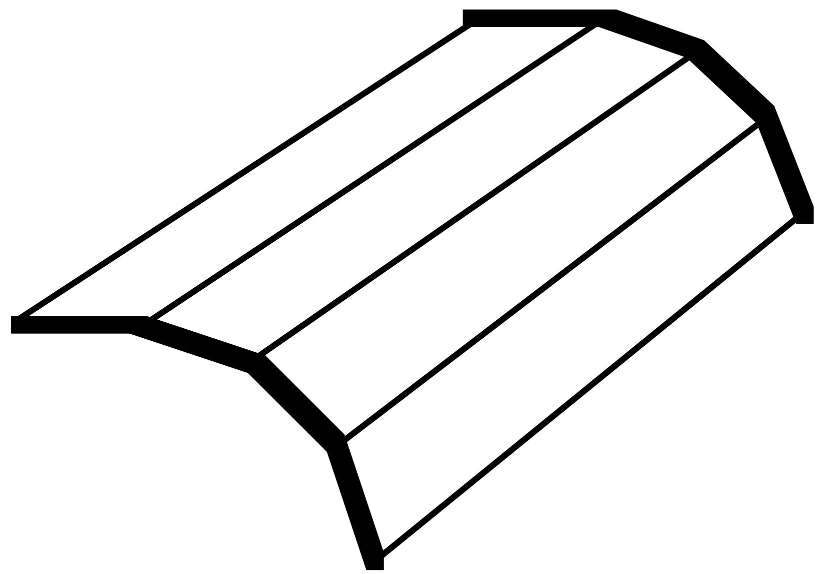


Normals

- Recall: normal is cross product of tangent in t direction and s direction.
- Cylinder: normal is cross-product of curve tangent and direction vector
- SOR: take curve normal and spin round axis

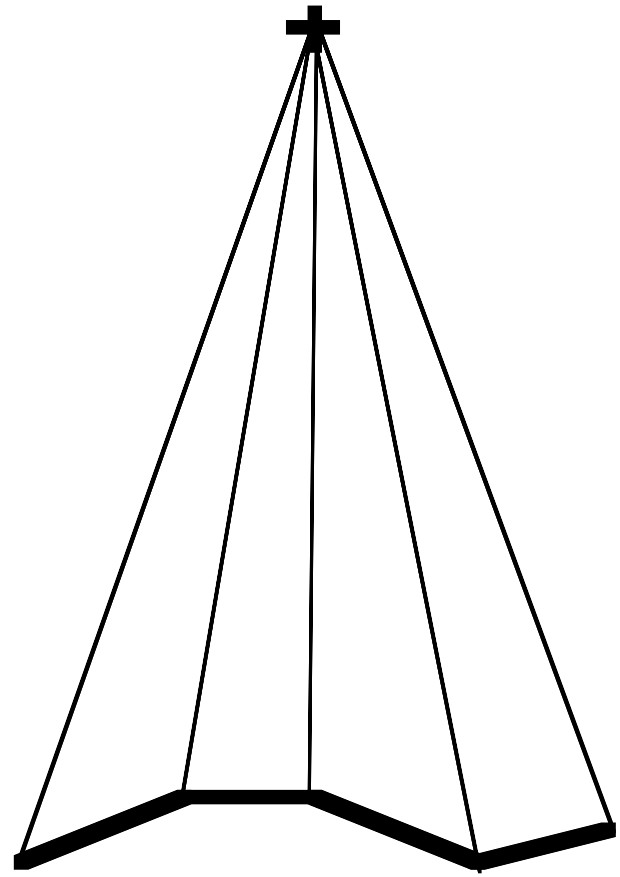
Rendering

- Cylinders: small steps along curve, straight segments along t generate polygons; exact normal is known.



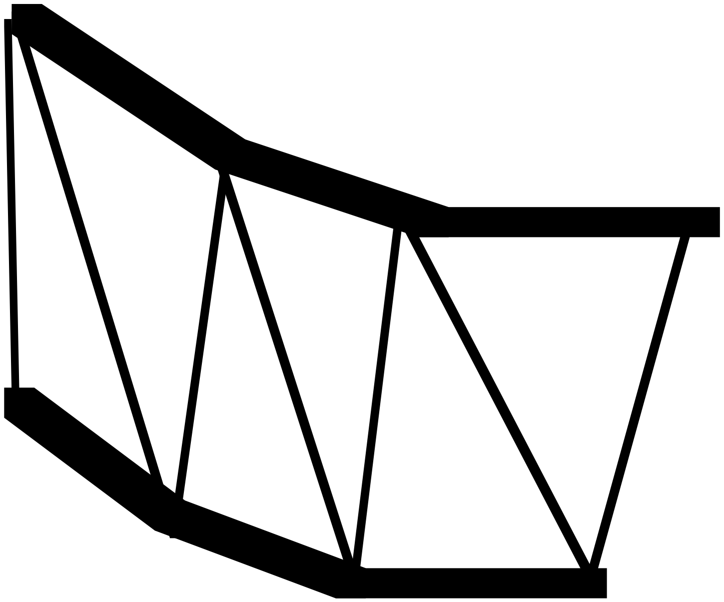
Rendering

- Cone: small steps in s generate straight edges, join with vertex to get triangles, normals known exactly except at vertex.



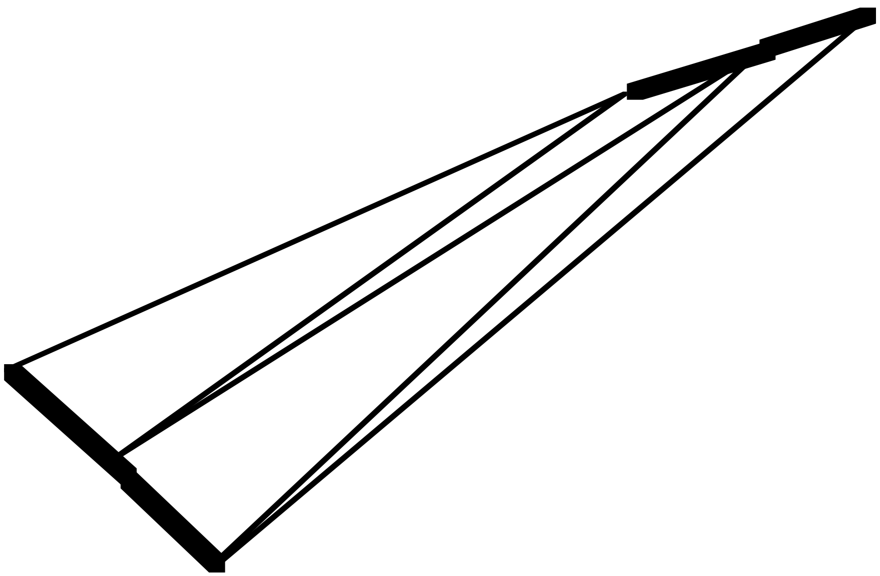
Rendering

- SOR: small steps in s generate strips, small steps in t along the strip generate edges; join up to form triangles. Normals known exactly.

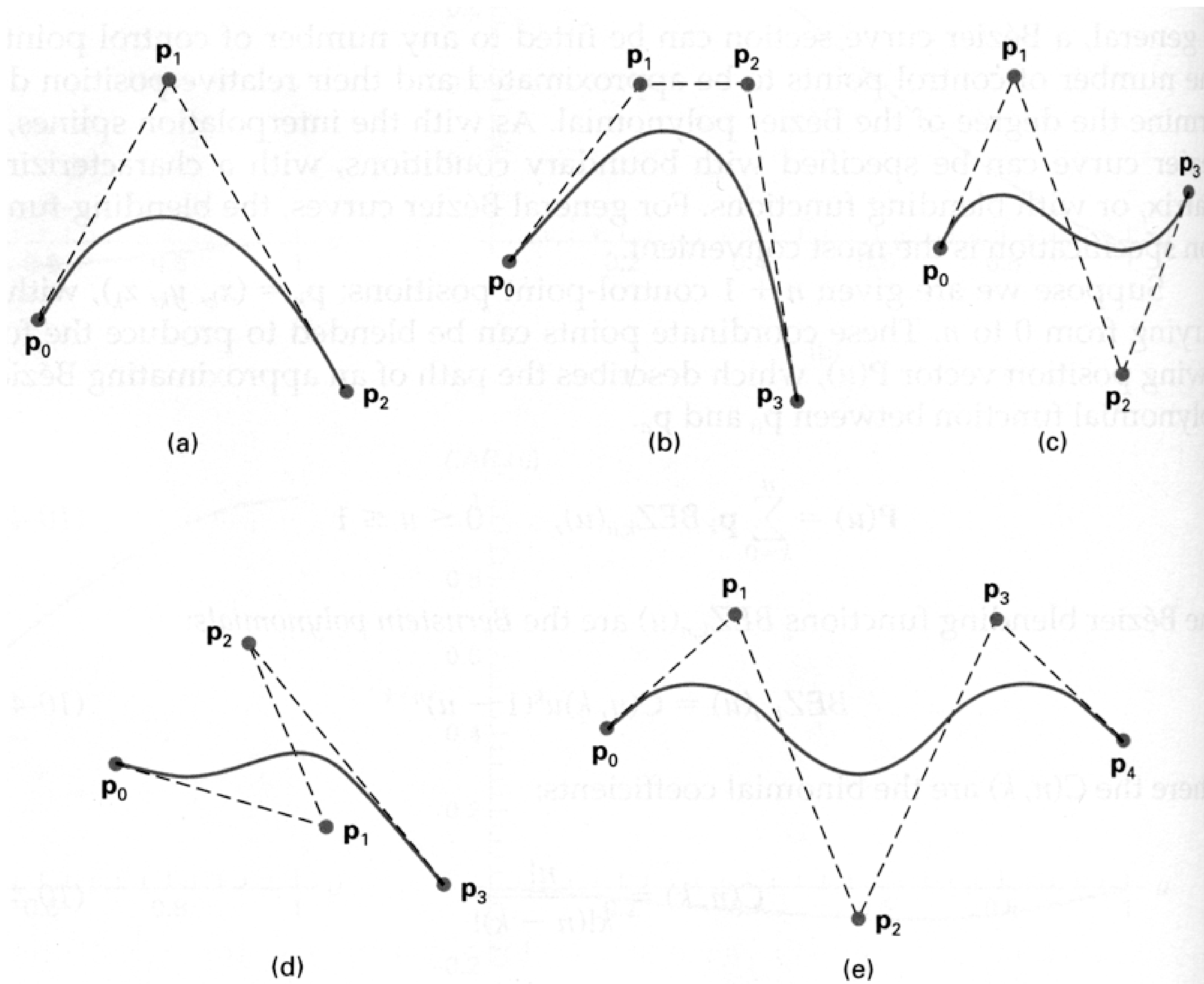


Rendering

- Ruled surface: steps in s generate polygons, join opposite sides to make triangles - otherwise “non planar polygons” result. Normals known exactly.



Bezier curves-1



- obtained by iterated linear interpolation
- process is known as DeCasteljau's algorithm

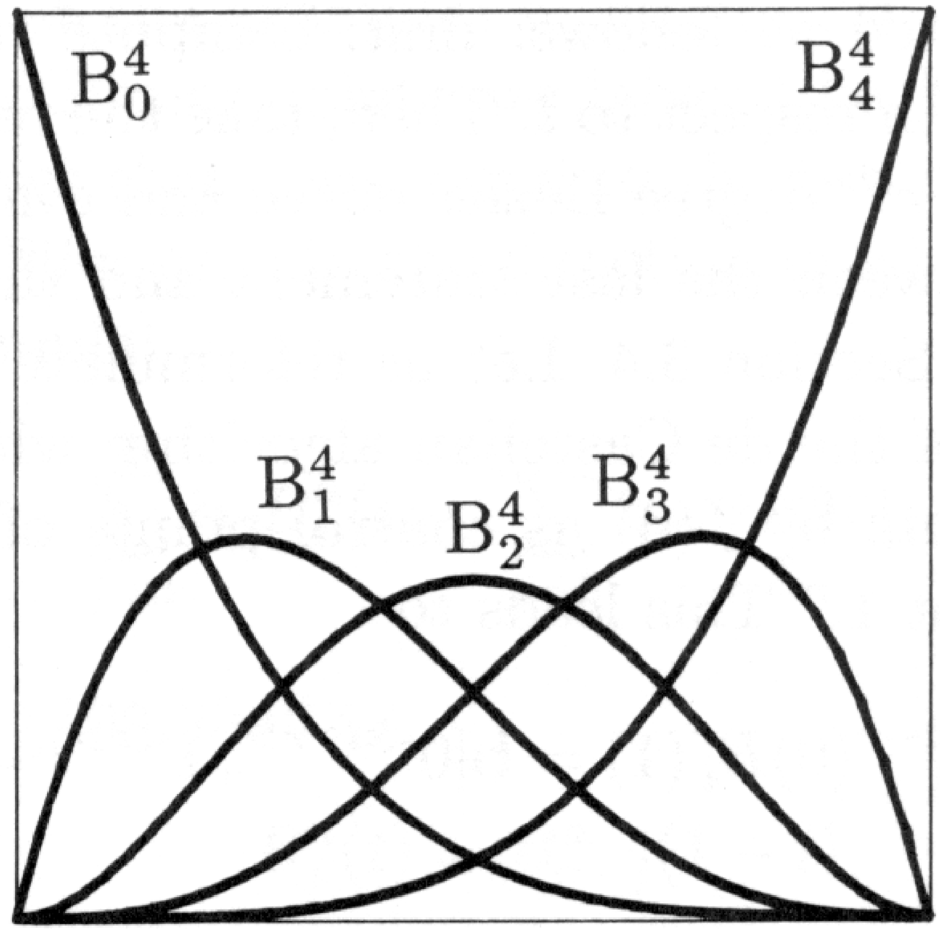
Bezier curves - II

- Blending functions are the Bernstein polynomials

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

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

- e.g. two points

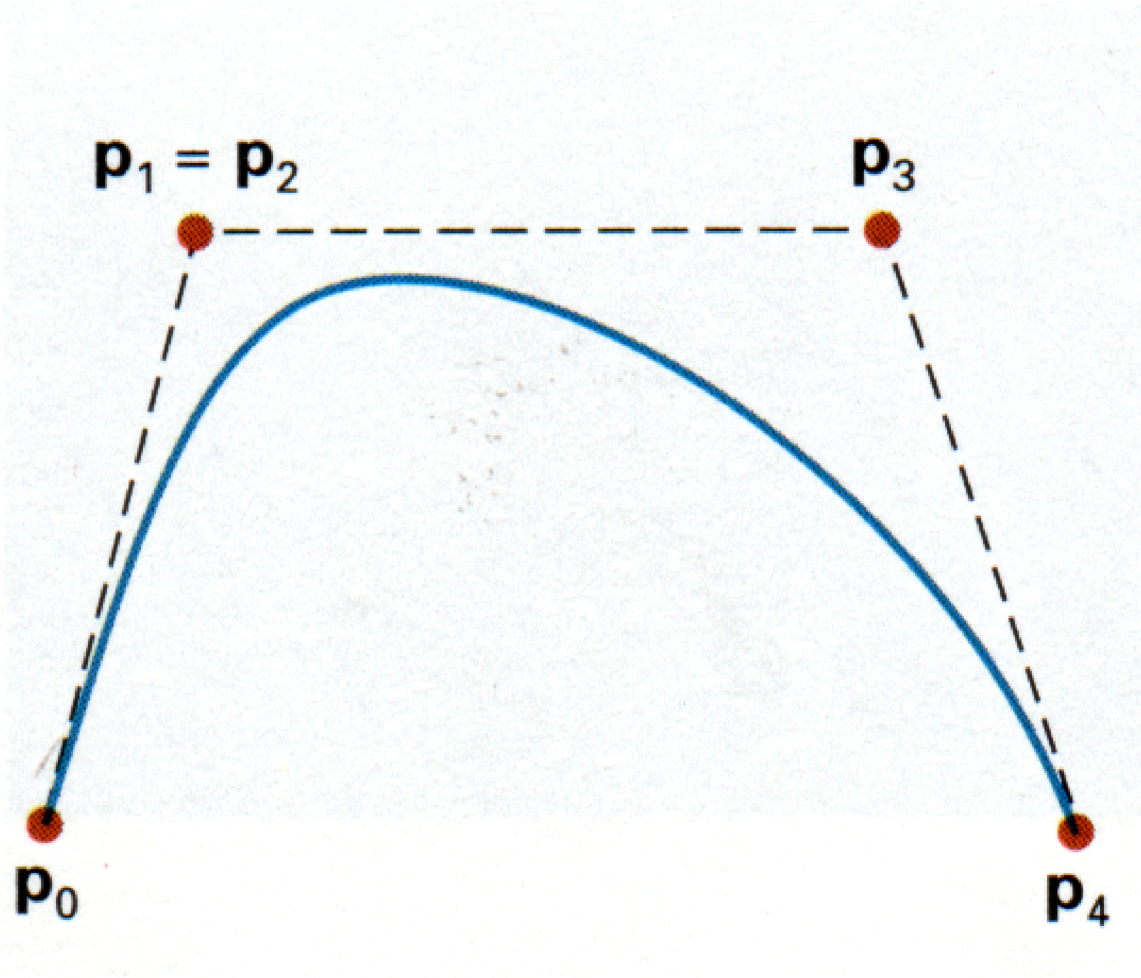


Bezier curves - III

- Bernstein polynomials have several important properties
  - they 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 p0 to p1
  - tangent at end lies along vector from pn-1 to pn

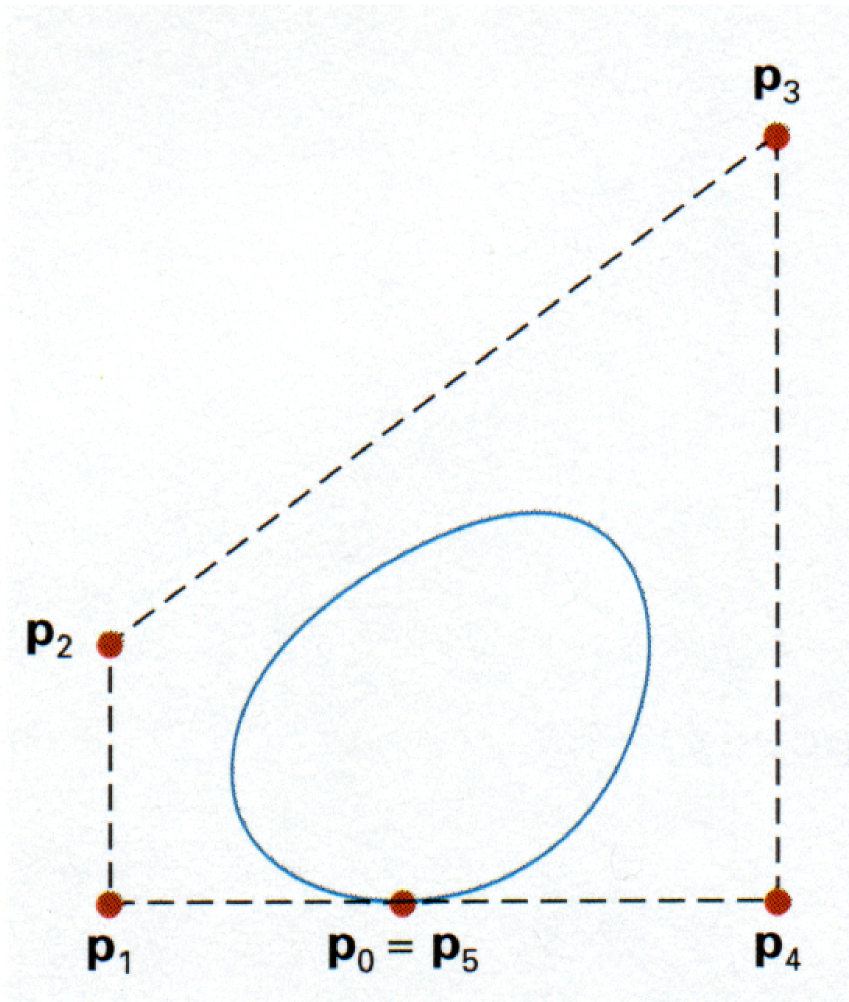
Bezier curve tricks - I

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



Bezier 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



Subdivision for Bezier curves

- Use De Casteljau (repeated linear interpolation) to identify points.
- Points as marked in figure give two control polygons, for two Bezier curves, which lie on top of the original.
- Repeated subdivision leads to a polygon that lies very close to the curve
- Limit of subdivision proces is a curve

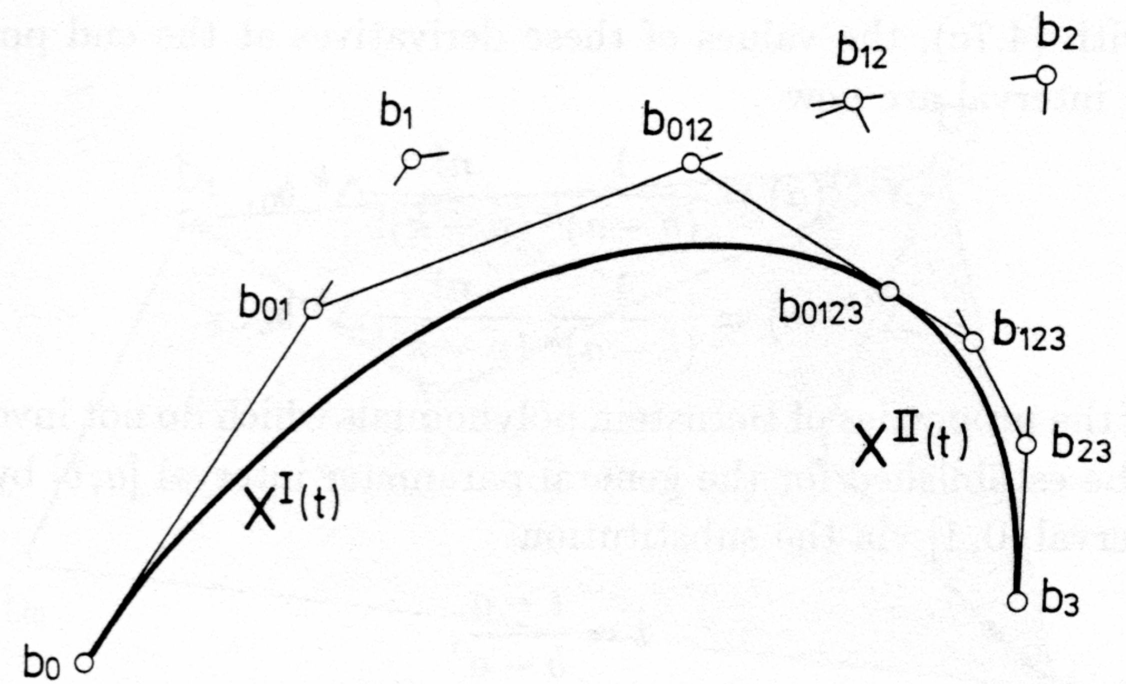


Fig. 4.5. Decomposition of a Bézier curve into two C^3 continuous curve segments (cf. Fig. 4.4).

de Casteljau Algorithm

- Cascading lerps

  $\mathbf{p}_{01} = (1\text{-}t)\ \mathbf{p}_0 + t\ \mathbf{p}_1$

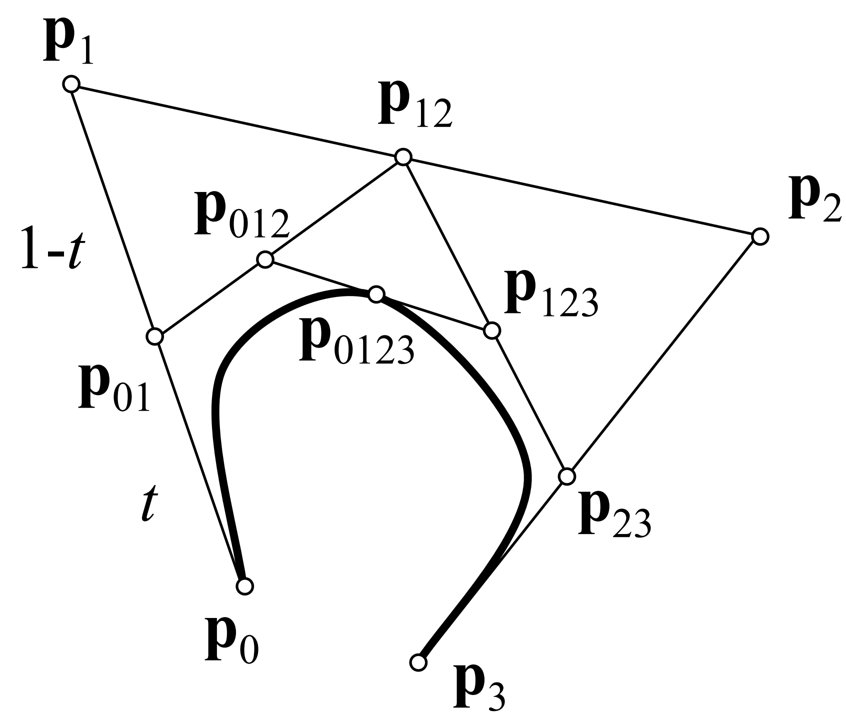
  $\mathbf{p}_{12} = (1\text{-}t)\ \mathbf{p}_1 + t\ \mathbf{p}_2$

  $\mathbf{p}_{23} = (1\text{-}t)\ \mathbf{p}_2 + t\ \mathbf{p}_3$

  $\mathbf{p}_{012} = (1\text{-}t)\ \mathbf{p}_{01} + t\ \mathbf{p}_{12}$

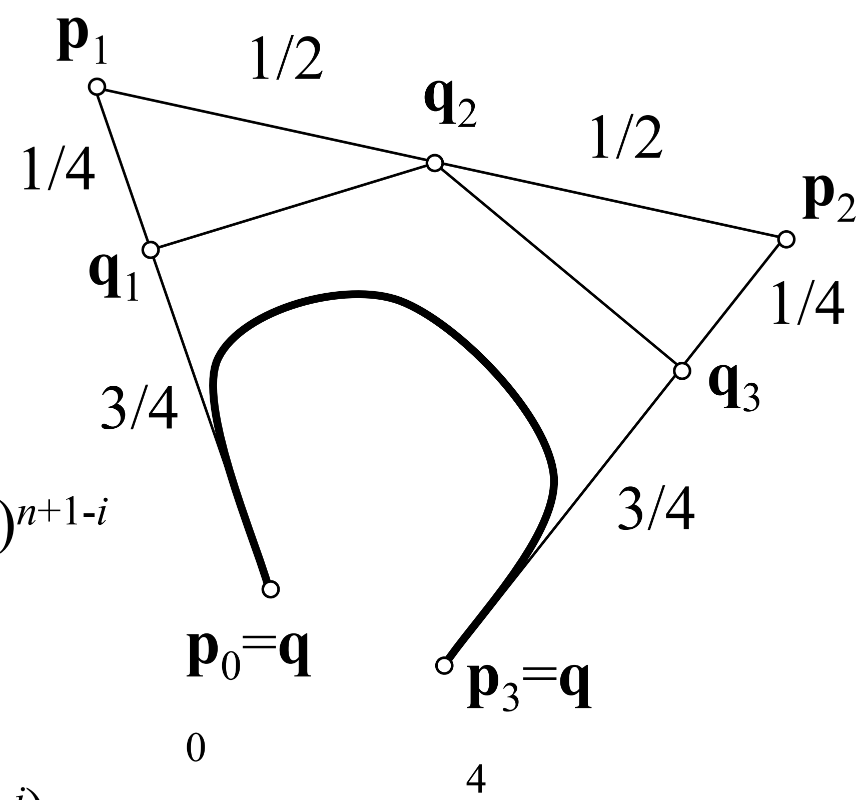
  $\mathbf{p}_{123} = (1\text{-}t)\ \mathbf{p}_{12} + t\ \mathbf{p}_{23}$

  $\mathbf{p}_{0123} = (1\text{-}t)\ \mathbf{p}_{012} + t\ \mathbf{p}_{123}$
- Subdivides curve at $\mathbf{p}_{0123}$
  - $\mathbf{p}_0\ \mathbf{p}_{01}\ \mathbf{p}_{012}\ \mathbf{p}_{0123}$
  - $\mathbf{p}_{0123}\ \mathbf{p}_{123}\ \mathbf{p}_{23}\ \mathbf{p}_3$
- Repeated subdivision converges to curve



coordinate free!

Degree Elevation



- Used to add more control over a curve
- Start with

$$\Sigma\, \mathbf{p}_i \binom{n}{i} t^i (1-t)^{n-i} = \Sigma\, \mathbf{q}_i \binom{n+1}{i} t^i (1-t)^{n+1-i}$$

- Now figure out the $\mathbf{q}_i$

$$(t+(1-t))\, \Sigma\, \mathbf{p}_i \binom{n}{i} t^i (1-t)^{n-i}$$

$$= \Sigma\, \mathbf{p}_i \binom{n}{i} \left(t^i (1-t)^{n+1-i} + t^{i+1} (1-t)^{n-i}\right)$$

- Compare coefficients

$$\mathbf{q}_i \binom{n+1}{i} = \mathbf{p}_i \binom{n}{i} + \mathbf{p}_{i-1} \binom{n}{i-1}$$

$$\mathbf{q}_i = (i/(n+1))\mathbf{p}_{i-1} + (n+1-i/(n+1))\mathbf{p}_i$$

- Repeated elevation converges to curve

Interpolating Splines

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

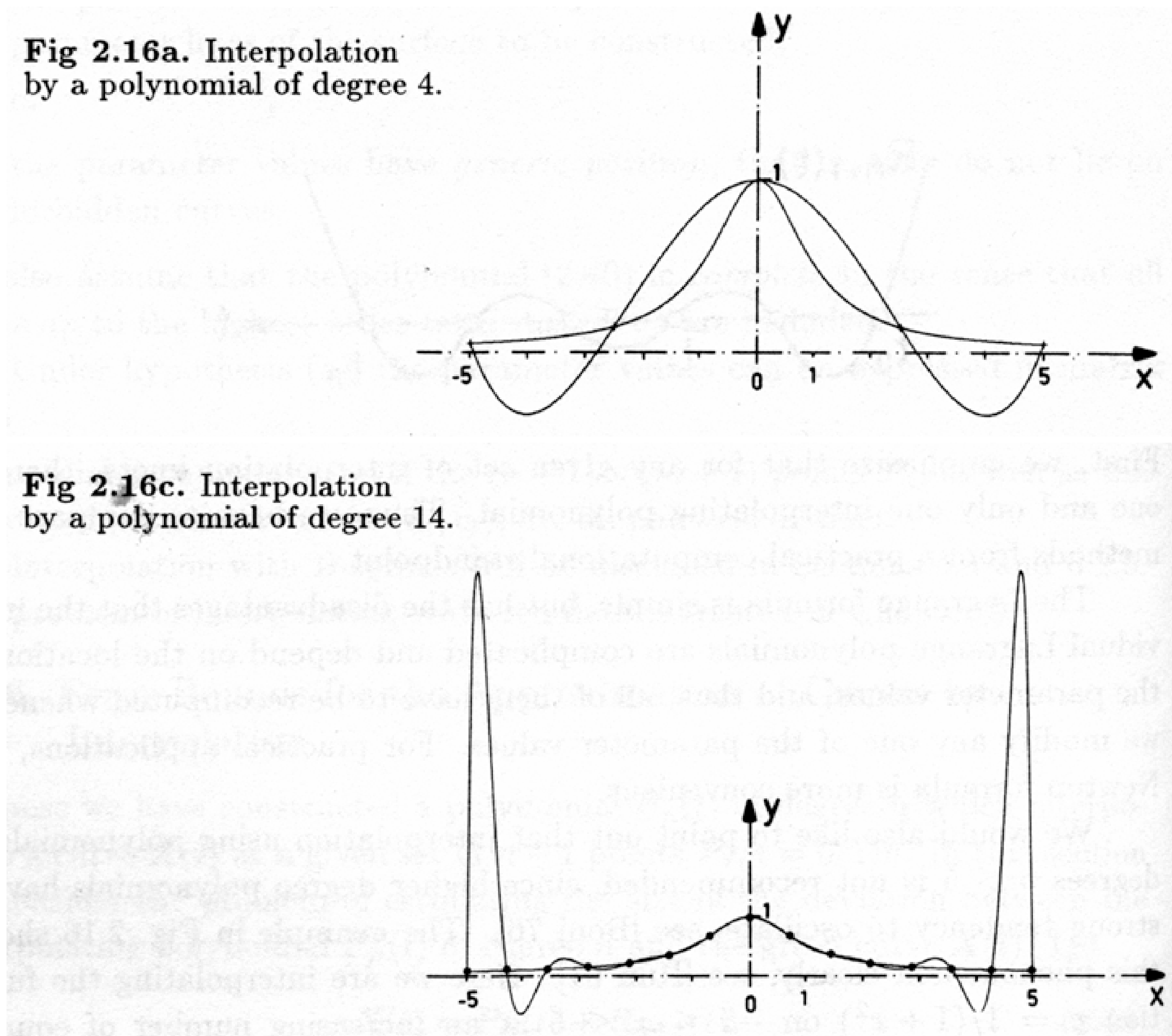


Fig 2.16a. Interpolation by a polynomial of degree 4.

Fig 2.16c. Interpolation by a polynomial of degree 14.

Interpolating Splines - II

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

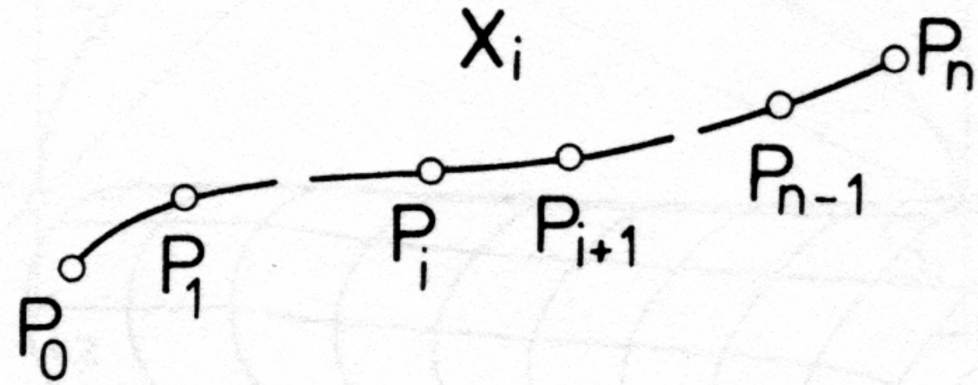


Fig. 3.11. The spline segment $\boldsymbol{X}_i$.

Interpolating Splines - II

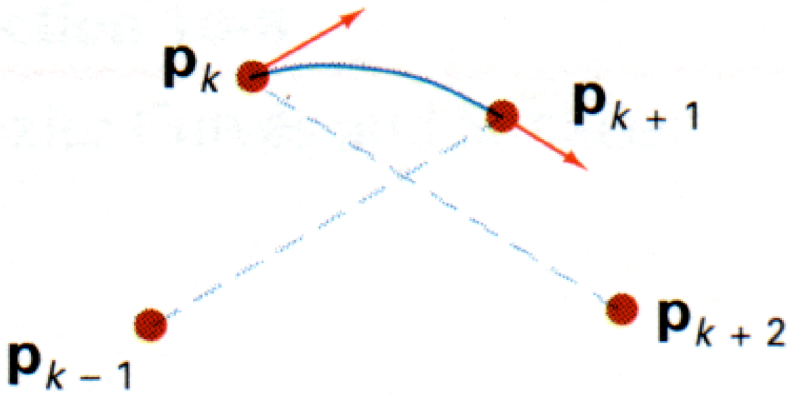
- bolt together a series of Hermite curves with equivalent derivatives.
- But where are the derivative values to come from?
  - Measurements
  - Combination of points
  - Continuity considerations

- Cardinal splines
  - average points
  - t is “tension”
  - specify endpoint tangents
    - or use difference between first two, last two points

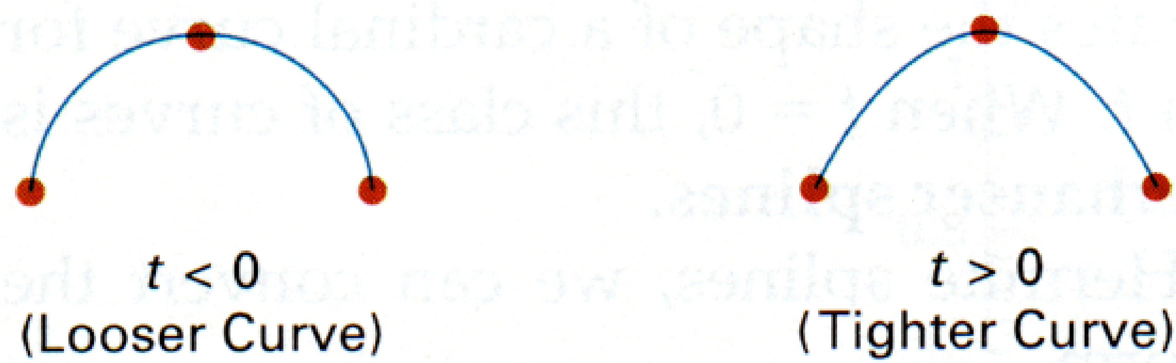
$$P'_k = \left(\frac{1}{2}\right)(1-t)\left(P_{k+1} - P_{k-1}\right)$$

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

Tension



- t=0 gives derivatives as <-
- different values of tension give longer/shorter tangents



Interpolating Splines

- Intervals:

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

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

  - t values often called “knots”

- Spline form:

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

Continuity

- Require at endpoints:
  - endpoints equal
  - 1'st derivatives equal
  - 2'nd derivatives equal

$$\boldsymbol{X}_i(t_i) = \boldsymbol{X}_{i-1}(t_i) \quad \text{or} \quad \boldsymbol{X}_i(t_{i+1}) = \boldsymbol{X}_{i+1}(t_{i+1}),$$

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

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

- From endpoint and 1'st derivative:

$$X_i(t_i) = P_i = D_i, \qquad X_i(t_{i+1}) = P_{i+1} = A_i \Delta t_i^3 + B_i \Delta t_i^2 + C_i \Delta t_i + D_i,$$
$$X_i'(t_i) = P_i' = C_i, \qquad X_i'(t_{i+1}) = P_{i+1}' = 3A_i \Delta t_i^2 + 2B_i \Delta t_i + C_i,$$

- So that

$$A_i = \frac{1}{(\Delta t_i)^3}[2(P_i - P_{i+1}) + \Delta t_i(P_i' + P_{i+1}')],$$
$$B_i = \frac{1}{(\Delta t_i)^2}[3(P_{i+1} - P_i) - \Delta t_i(2P_i' + P_{i+1}')].$$

- Yielding:

$$X_i(t) =$$
$$P_i\left(2\frac{(t-t_i)^3}{(\Delta t_i)^3} - 3\frac{(t-t_i)^2}{(\Delta t_i)^2} + 1\right) + P_{i+1}\left(-2\frac{(t-t_i)^3}{(\Delta t_i)^3} + 3\frac{(t-t_i)^2}{(\Delta t_i)^2}\right)$$
$$+ P_i'\left(\frac{(t-t_i)^3}{(\Delta t_i)^2} - 2\frac{(t-t_i)^2}{\Delta t_i} + (t-t_i)\right) + P_{i+1}'\left(\frac{(t-t_i)^3}{(\Delta t_i)^2} - \frac{(t-t_i)^2}{\Delta t_i}\right)$$

- Second Derivative:

$$\boldsymbol{X}_i''(t) = 6\boldsymbol{P}_i\left(\frac{2(t-t_i)}{(\Delta t_i)^3} - \frac{1}{(\Delta t_i)^2}\right) + 6\boldsymbol{P}_{i+1}\left(-2\frac{(t-t_i)}{(\Delta t_i)^3} + \frac{1}{(\Delta t_i)^2}\right)$$
$$+ 2\boldsymbol{P}_i'\left(3\frac{(t-t_i)}{(\Delta t_i)^2} - \frac{2}{\Delta t_i}\right) + 2\boldsymbol{P}_{i+1}'\left(\frac{3(t-t_i)}{(\Delta t_i)^2} - \frac{1}{\Delta t_i}\right).$$

- Want:

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

- Yielding:

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

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

Parametric vs Geometric Continuity

- **Parametric continuity**:
  - The curve and derivatives up to k are continuous *as a function of parameter value*
    - C^k
  - Useful for (for example) animation
  - e.g. the interpolating spline from above
- **Geometric continuity**
  - curve, derivatives up to k'th are the same for equivalent parameter values
  - i.e. there exists a reparametrisation that would achieve parametric continuity
    - D^k
  - Useful, because we often don't require parametric continuity,
  - e.g. take two Hermite curves, both parametrised by [0, 1], identify endpoints and derivatives

More on Geometric Continuity

- Tangent direction is invariant to translation and parametrisation - so we can use this to get G1 continuity.
- G2 - use curvature
  - property of a curve that is invariant to rotation and translation, and also reparametrisation
  - (1/radius) for best fitting circle
    - the circle whose 2nd derivative is the same as the curve's
    - (equivalent) a circle that intersects the curve in three points arbitrarily close
  - Formula
    - $(x'' y' - y'' x')/(x'^2 + y'^2)^{(3/2)}$
    - $dN/ds=kN$ for N the unit normal

Keep in mind

- Lagrange and Hermite interpolates of the same degree are the same families of curves
  - they just have different control structures
- The interpolating cubic spline is equivalent to a bunch of Hermite cubics, with a different control structure
  - we got the derivatives from the second derivative constraint
- The line of reasoning for interpolating cubic splines works for higher degrees, too
  - but we must either use more derivatives, or supply more information
  - Cubic is the most important case, because cubic splines (rather roughly) look like wooden splines
- We chose parameter values for the interpolating curve
  - different choices lead to different curves

Spline blending functions

- “Switches” turn blending functions on and off
- E.g. a piecewise cubic spline obtained by attaching two Hermite curves to one another
  - In principle, there are 8 blending functions (4 points and 4 derivatives)
  - Actually, two points and two derivatives are the same
    - 6 blending functions
    - these are piecewise cubic, easily sketched
  - The properties of the blending functions are what’s important
- Now let’s consider splines that don’t interpolate, by concentrating on the blending functions

B-splines - I

- We obtain a set of blending functions by a recursive definition, with “switches” at the base of the recursion
- Curve:

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

- where d (called the “order”) is:

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

B-Spline Blending Functions

- Knots
  - idea: parameter values where curve segments meet, as in Hermite example

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

$$\text{where} \quad t_0 \leq t_1 \leq \ldots \leq t_{n+d}$$

- Blending functions

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

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

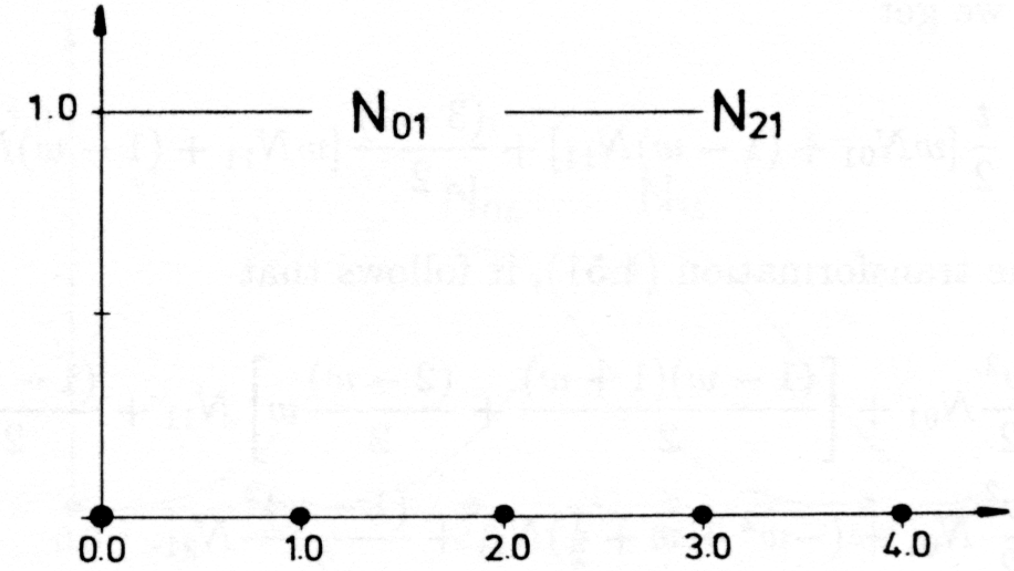


Fig. 4.22c. The B-splines N_{01}, N_{21}.

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

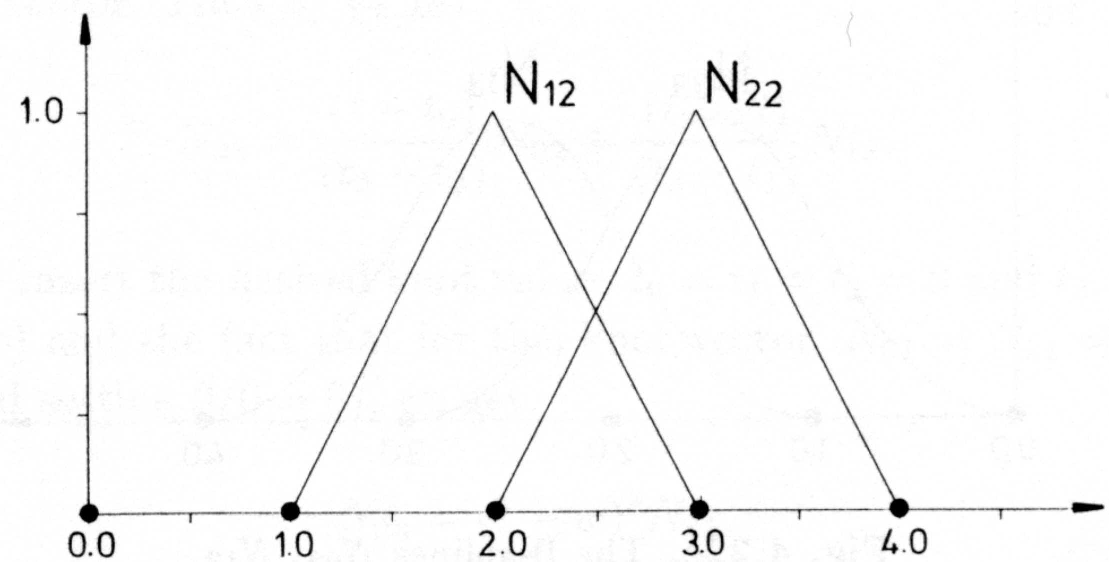


Fig. 4.22d. The B-splines N_{12}, N_{22}.

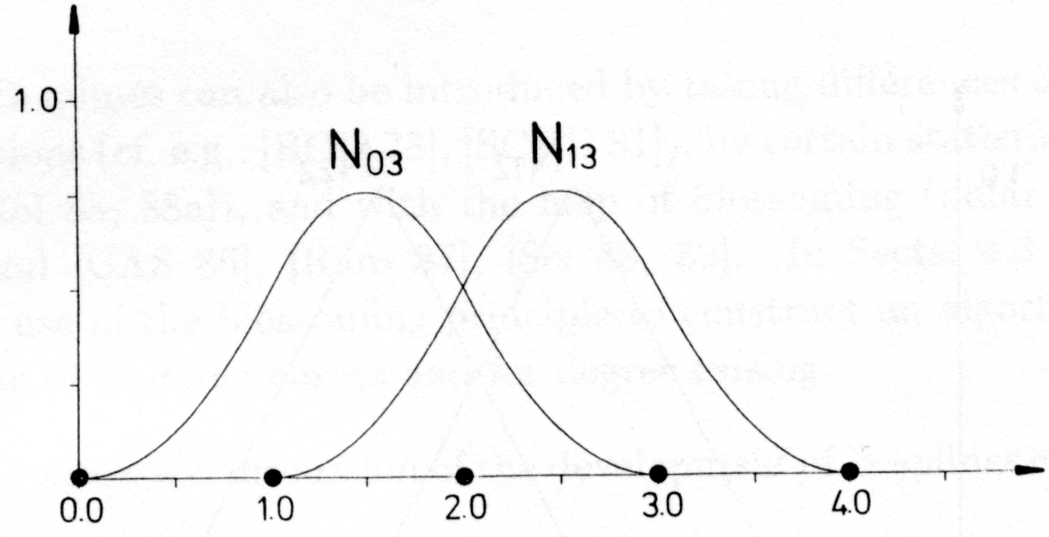
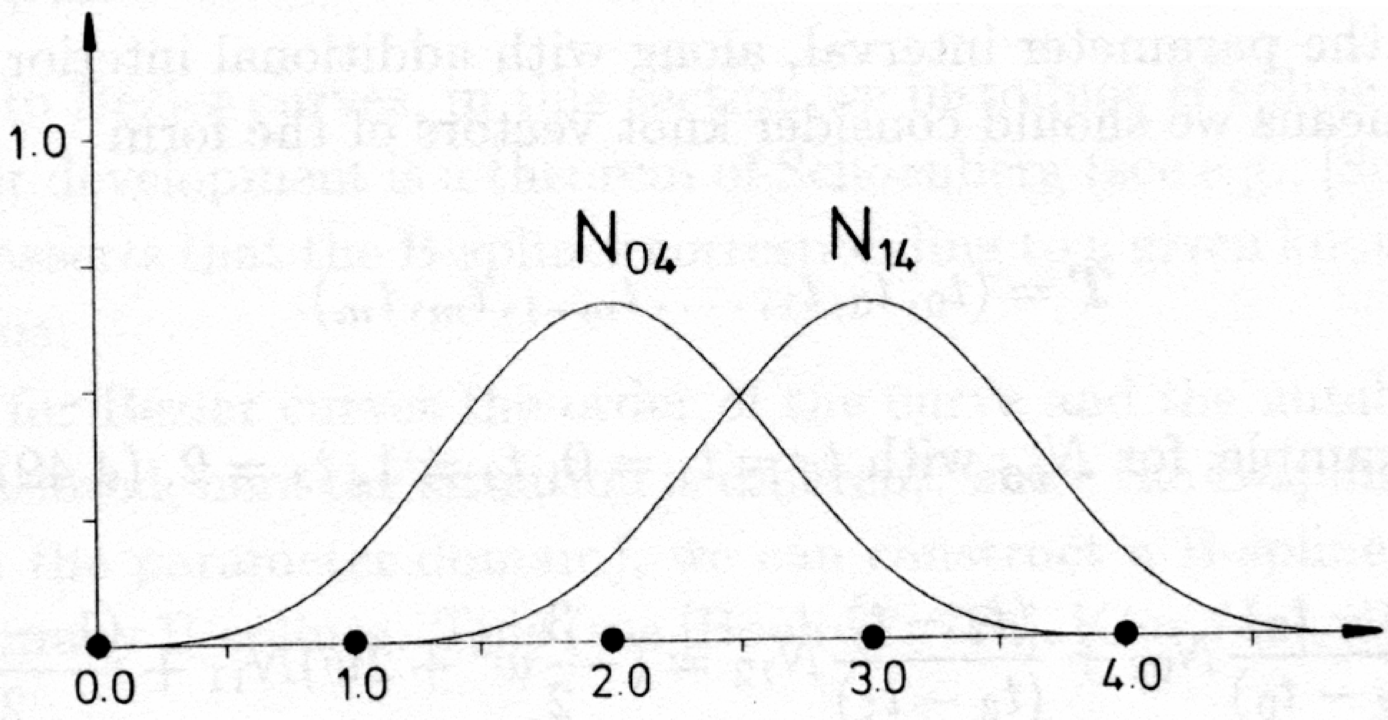


Fig. 4.22e. The B-splines N_{03}, N_{13}.



Closed B-Splines

- Periodically extend the control points and the knots

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

- etc

Fig. 4.26a.

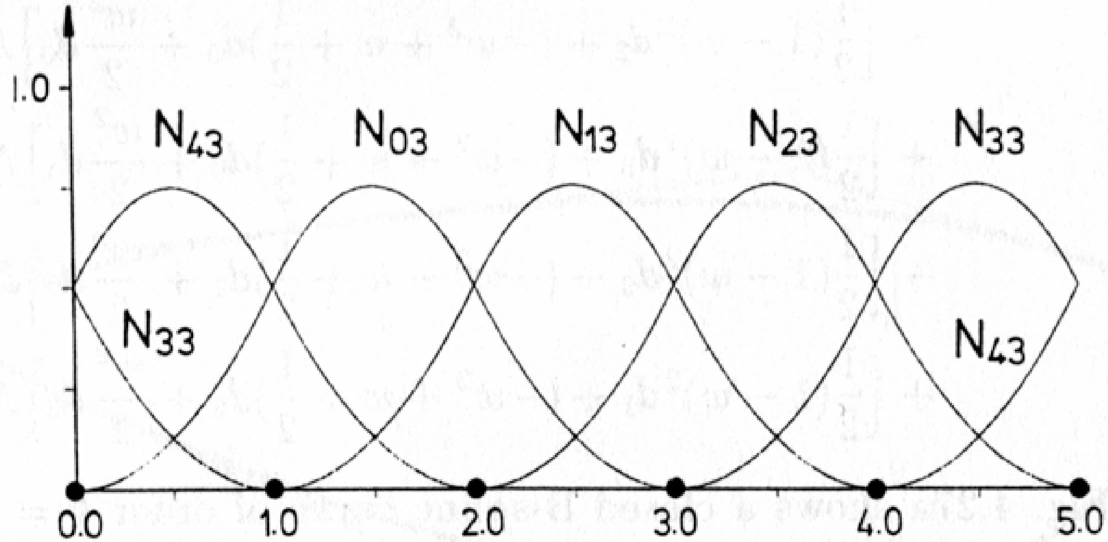


Fig. 4.26b.

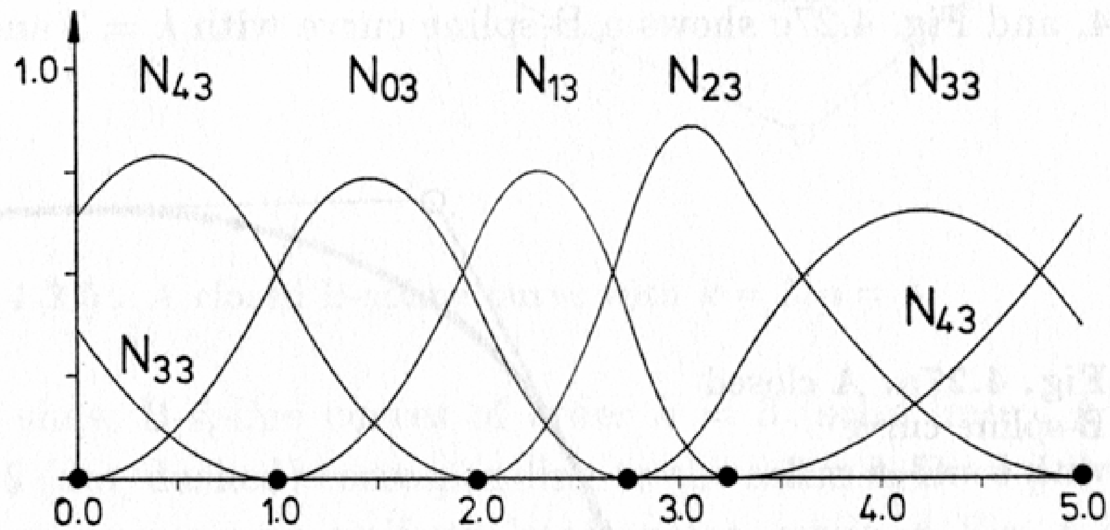


Fig. 4.26. B-splines with uniform and non-uniform knot vectors for a closed B-spline curve.

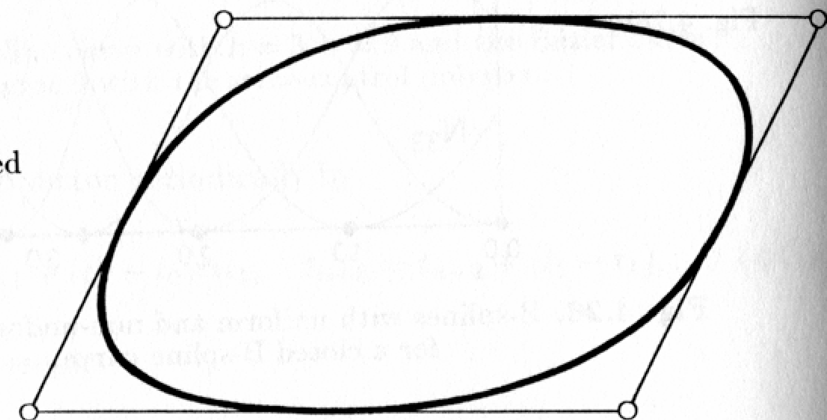


Fig. 4.27a. A closed B-spline curve with $k = 3, n = 3$.

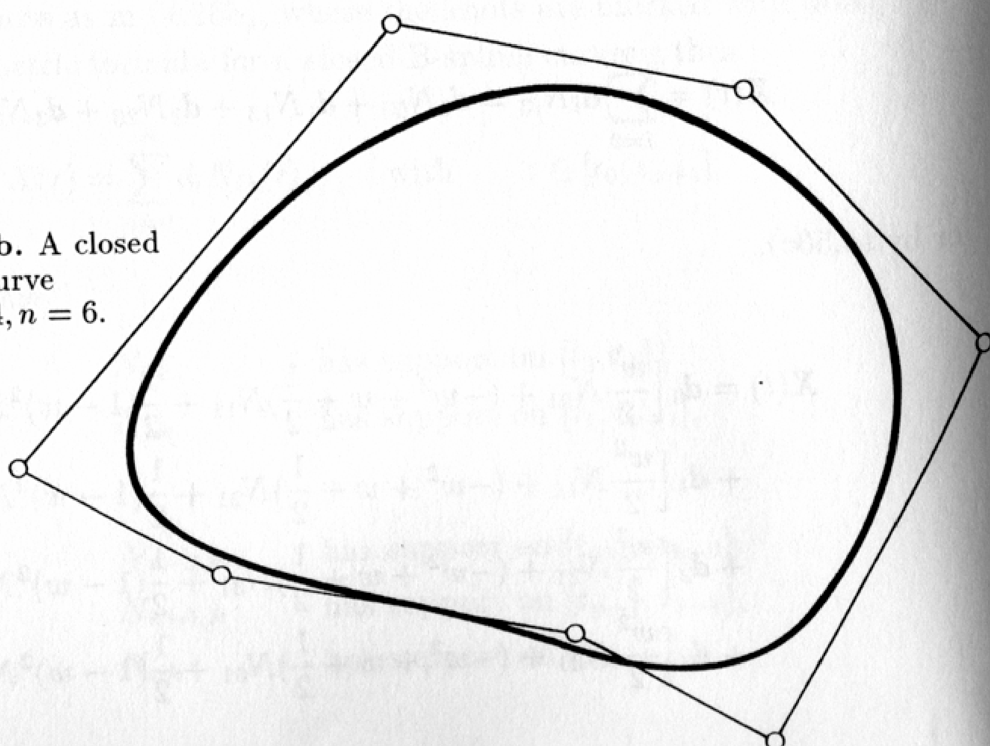


Fig. 4.27b. A closed B-spline curve with $k = 4, n = 6$.

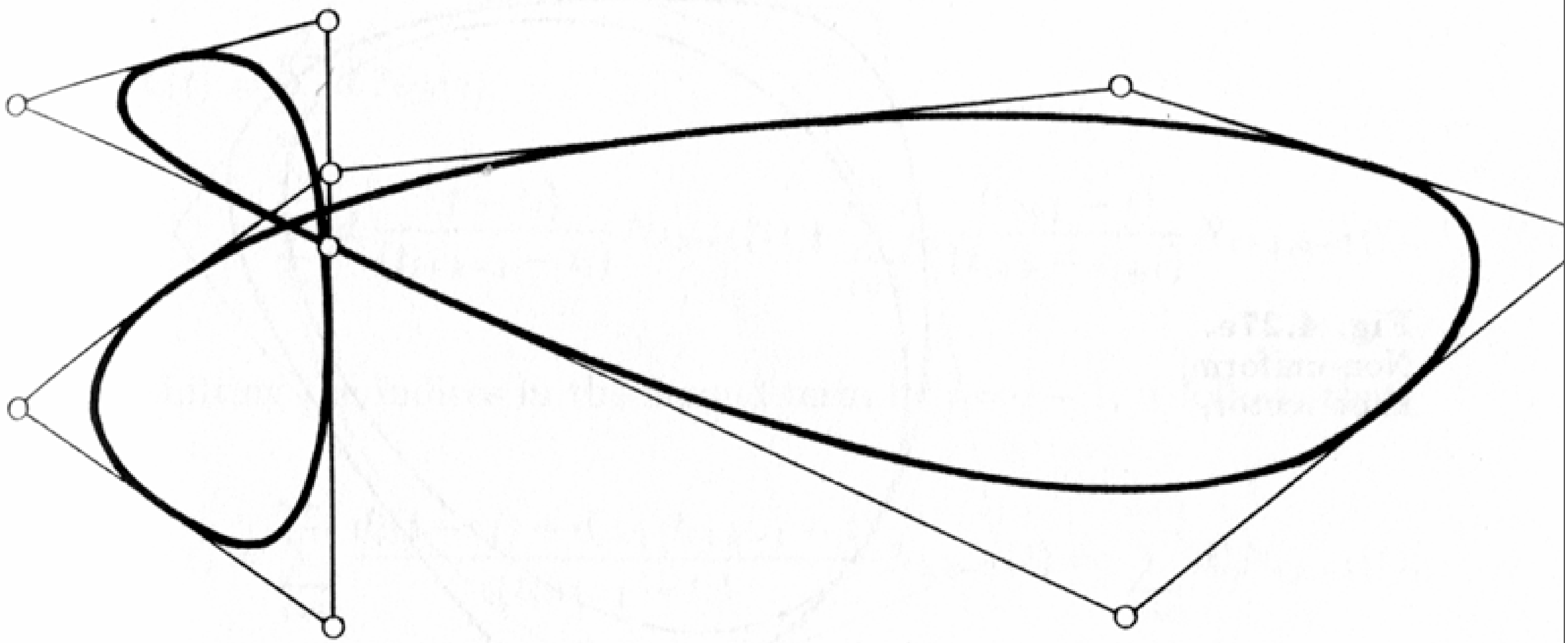
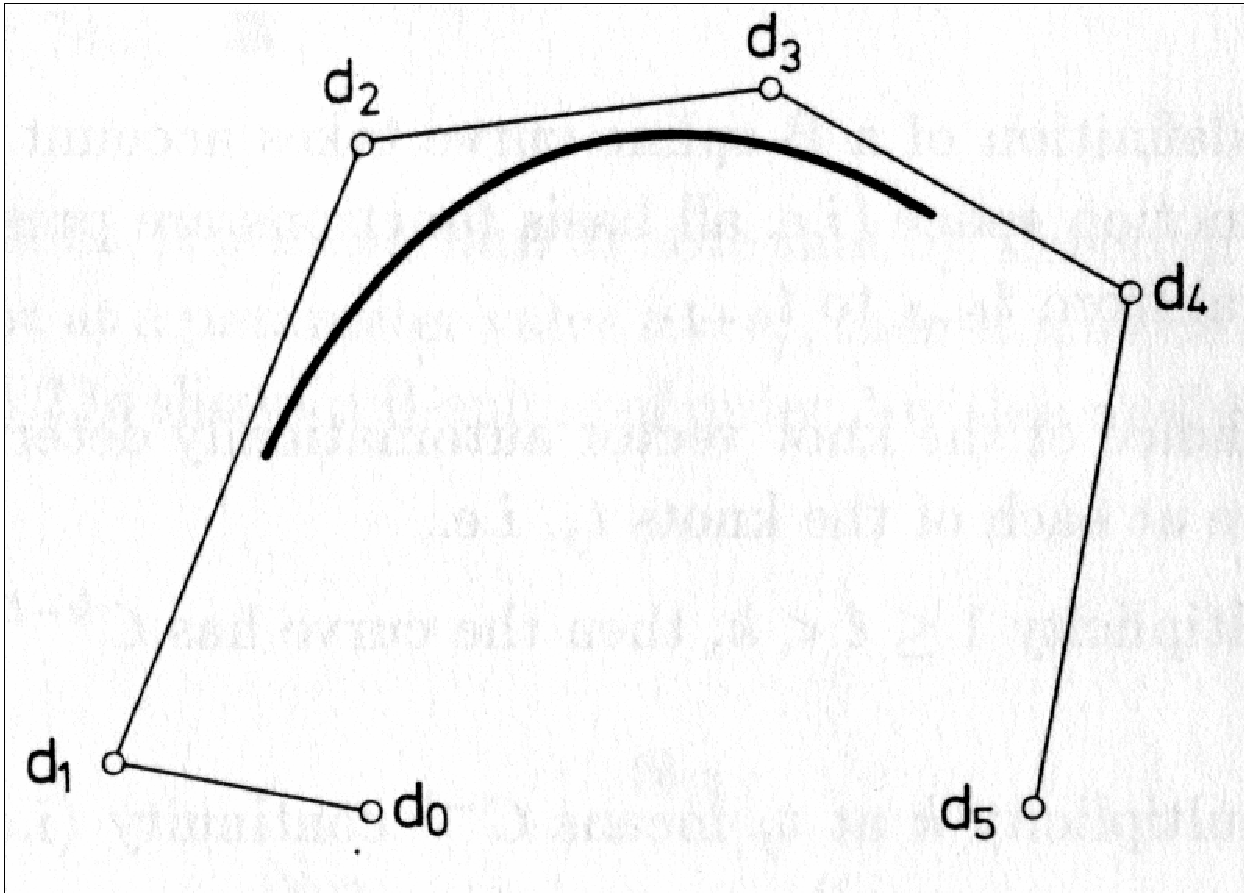


Fig. 4.27c. A closed B-spline curve with $k = 3, n = 8$.



A B-spline curve, with knots at 0,1,... and order 5

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

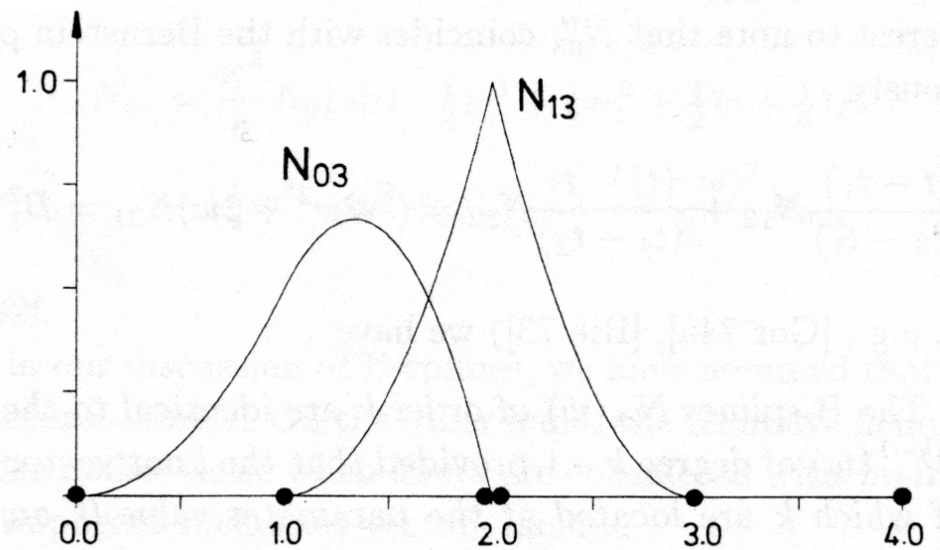


Fig. 4.22g. A quadratic B-spline with a double knot.

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 se
- e.g. cubic case below
- Notice that a control point influences at most d parameter intervals - **local control**

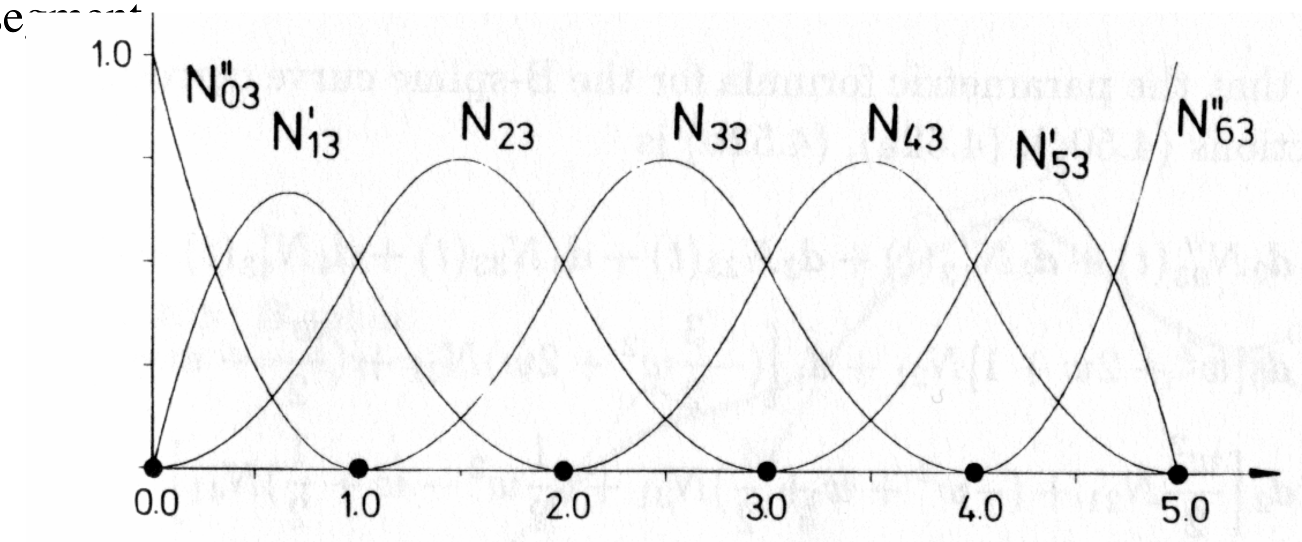


Fig. 4.24a. B-splines for an open B-spline curve with uniform knot vector.

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

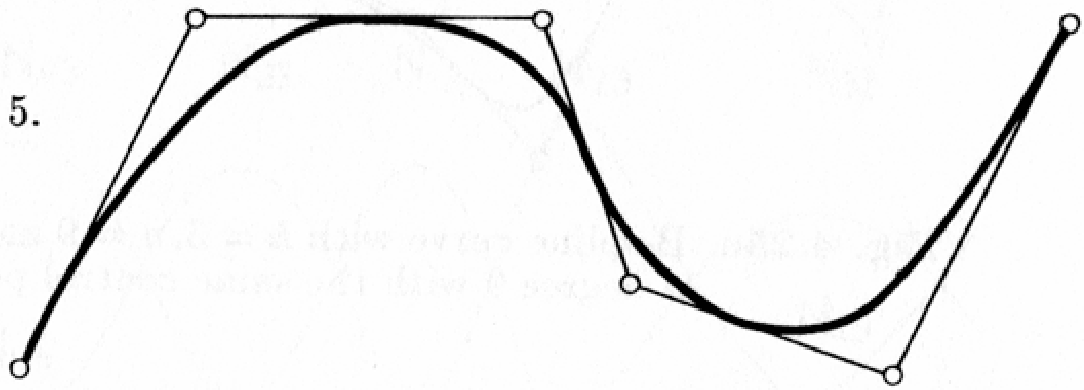
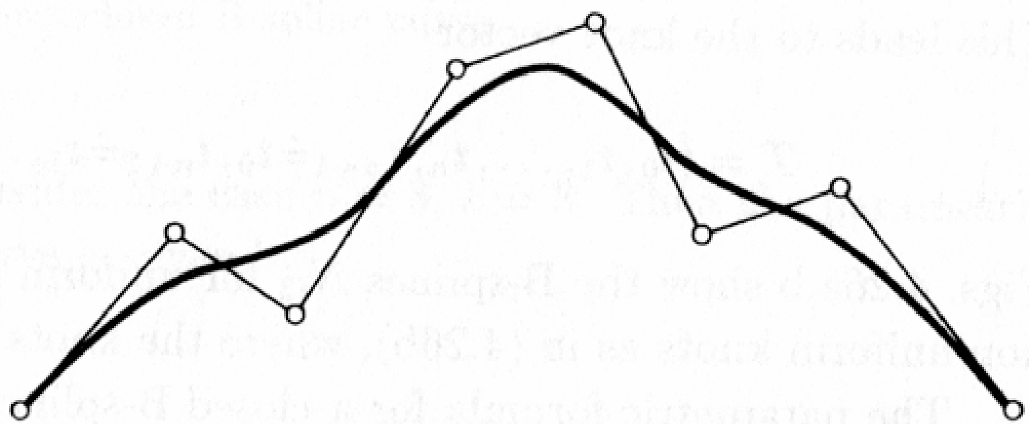


Fig. 4.25b. B-spline curve with $k = 4$, $n = 7$.



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

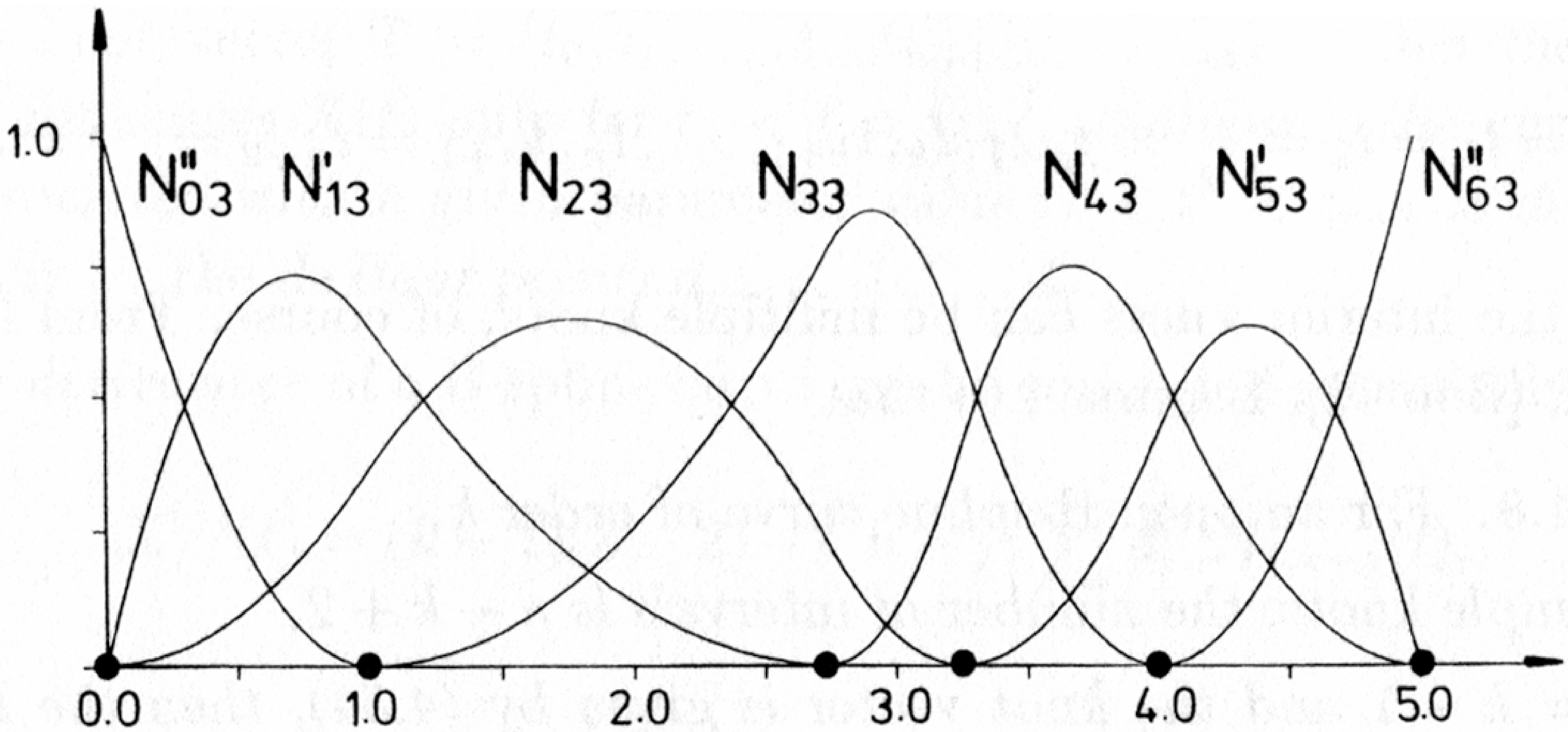


Fig. 4.24b. B-splines for an open B-spline curve with non-uniform knot vector.

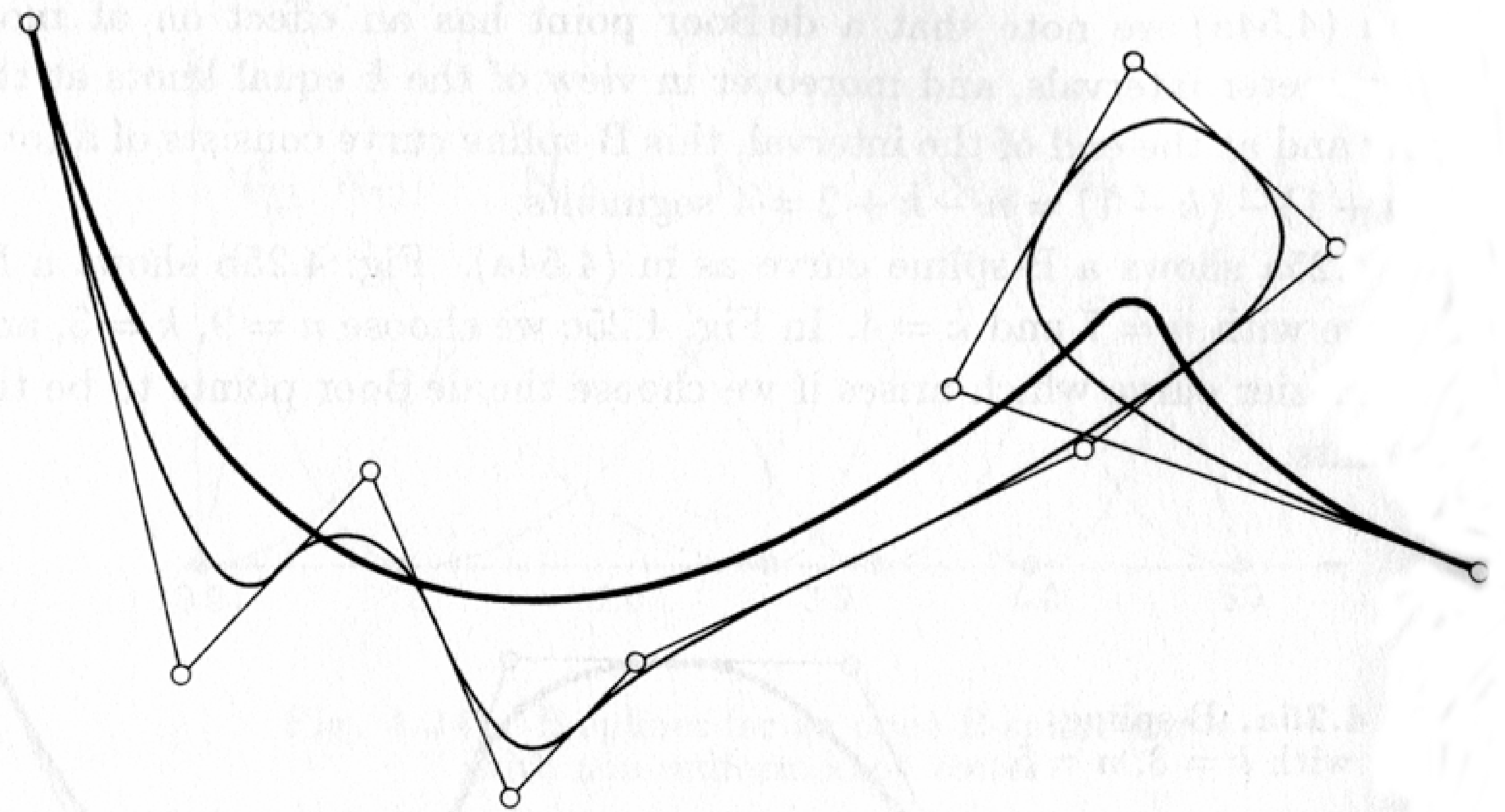


Fig. 4.25c. B-spline curve with $k = 3, n = 9$ and the Bézier curve of degree 9 with the same control polygon.

Bezier 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 points of the control polygon are collinear, then the curve is tangent to the polygon
  - if d 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