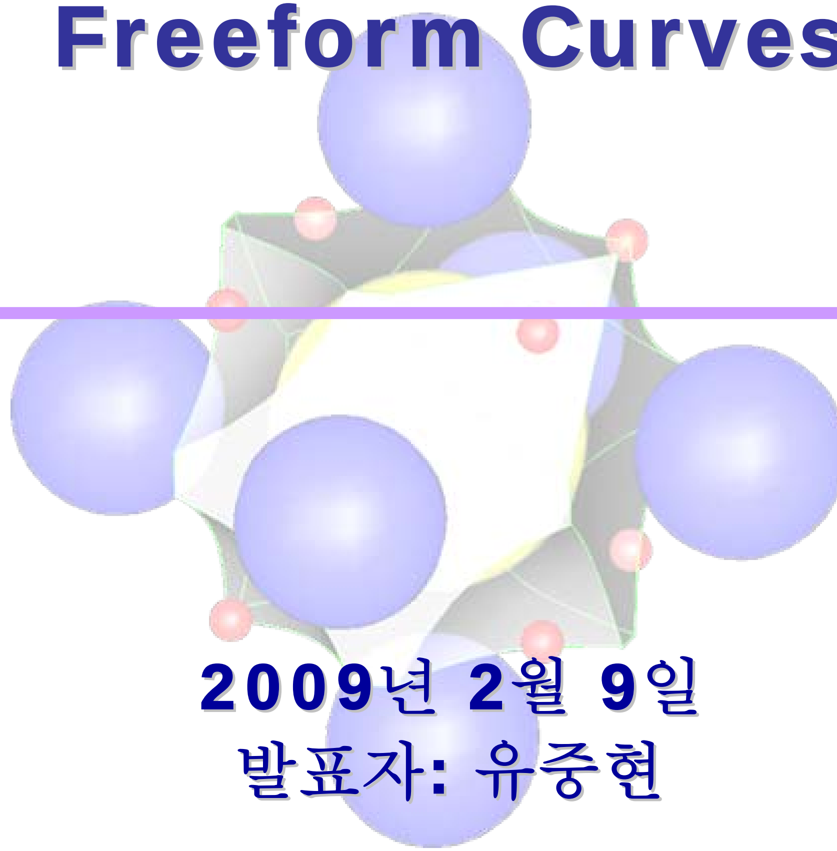
Architectural geometry: Freeform Curves

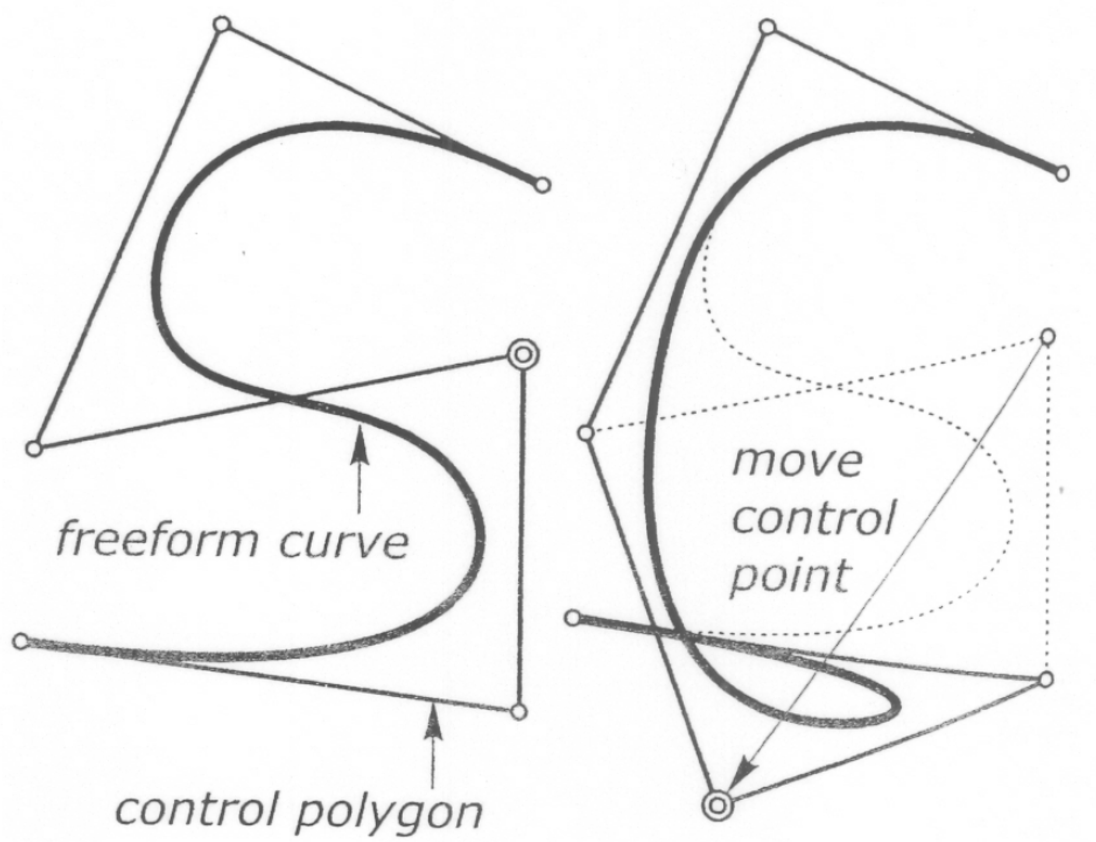


2009년 2월 9일

발표자: 유중현

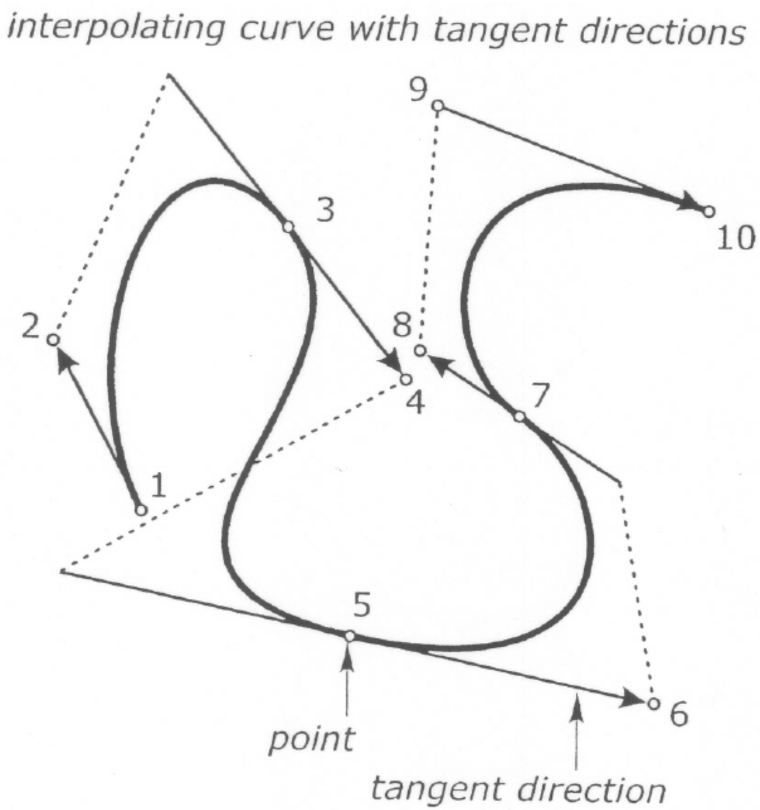
Freeform curves

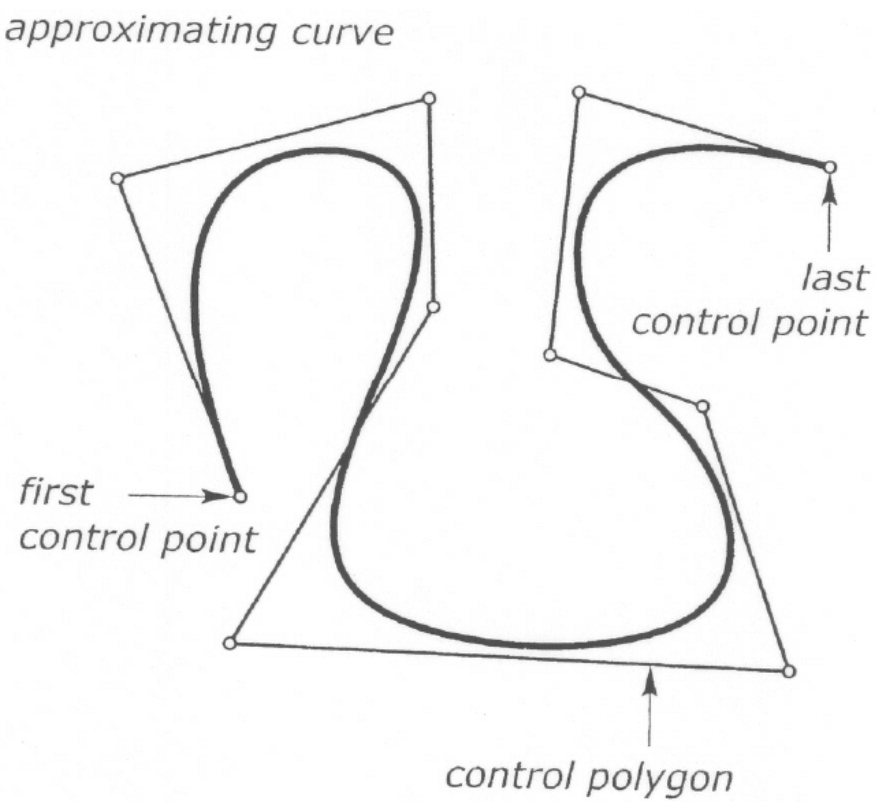
- **Control polygon (points)**
  - **Construction of freeform curve**
  - **Modification of freeform curve**



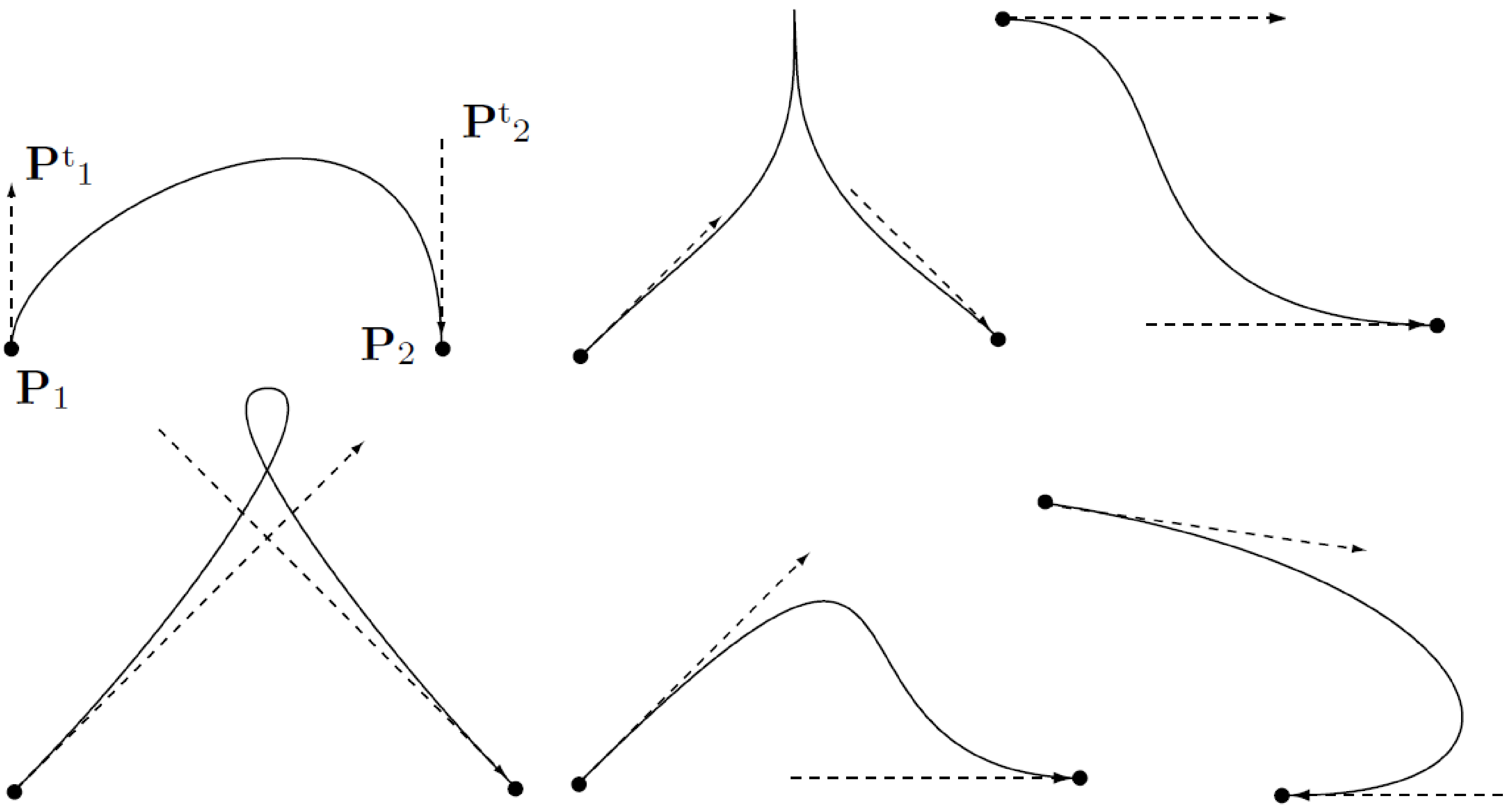
Freeform curves

- **Interactive curve design**
  - Interpolation
  - Approximation





Hermite interpolation



Parametric curve

$$C(t) = \sum_i BF_{i,n}(t) P_i$$

BF: basis function

P : control point

n : the degree of curve

Parametric curve

- **Representation is not unique!**

$$C(t) = \sum_{i=0}^{n} B_{i,n}(t) P_i \ (0 \le t \le 1)$$
$$= \sum_{i=0}^{n} PB_{i,n}(t) Q_i \ (0 \le t \le 1)$$

$B_{i,n}(t) = \frac{n!}{i!(n-i)!} t^i (1-t)^{n-i}$: Bernstein basis function

P : control point

n : the degree of curve

$PB_{i,n}(t) = t^i$: Power basis function

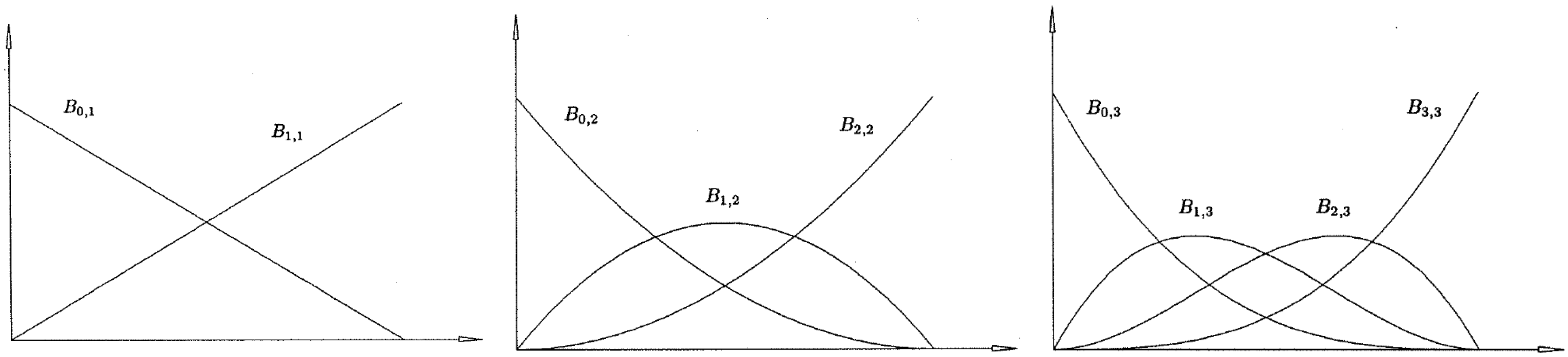
Q: appropriate coefficient

Basis conversion

$$
\begin{aligned}
C(t) &= \sum_{i=0}^{2} B_{i,2}(t)P_i \\
&= (1-t)^2 P_0 + 2t(1-t)P_1 + t^2 P_2 \\
&= t^2(P_0 - 2P_1 + P_2) + t(-2P_0 + 2P_1) + P_0 \\
&= \sum_{i=0}^{2} PB_{i,2}(t)Q_i
\end{aligned}
$$

where $Q_0 = P_0$, $Q_1 = -2P_0 + 2P_1$ and $Q_2 = P_0 - 2P_1 + P_2$

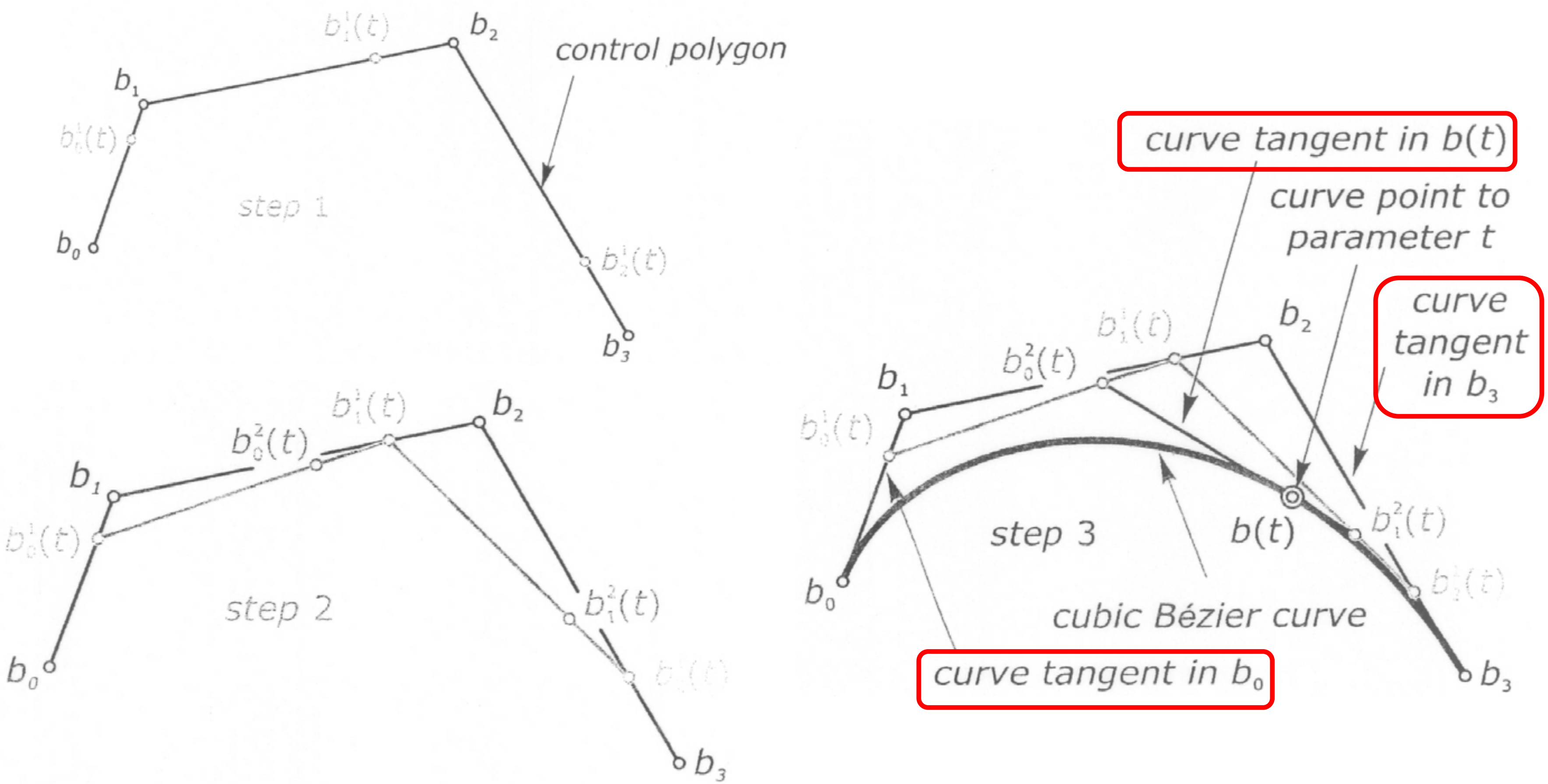
Bernstein basis functions



Bernstein basis (polynomial) function for $n = 1$, $n = 2$ and $n = 3$

Bézier curves

- **de Casteljau algorithm (n = 3;cubic case)**



Bézier curves

- **de Casteljau algorithm (n = 3;cubic case)**

Given: $\mathbf{b}_0, \mathbf{b}_1, \ldots, \mathbf{b}_n \in \mathbb{E}^3$ and $t \in \mathbb{R}$,

Set:

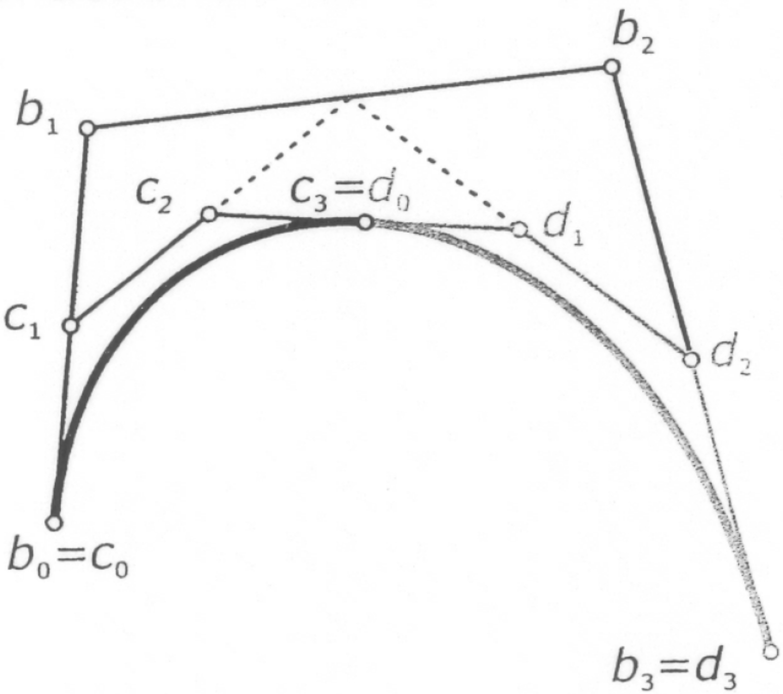
$$\mathbf{b}_i^r(t) = (1-t)\mathbf{b}_i^{r-1}(t) + t\mathbf{b}_{i+1}^{r-1}(t) \qquad \begin{cases} r = 1, \ldots, n \\ i = 0, \ldots, n-r \end{cases} \tag{3.2}$$

and $\mathbf{b}_i^0(t) = \mathbf{b}_i$. Then $\mathbf{b}_0^n(t)$ is the point with parameter value t on the *Bézier curve* $\mathbf{b}^n$.

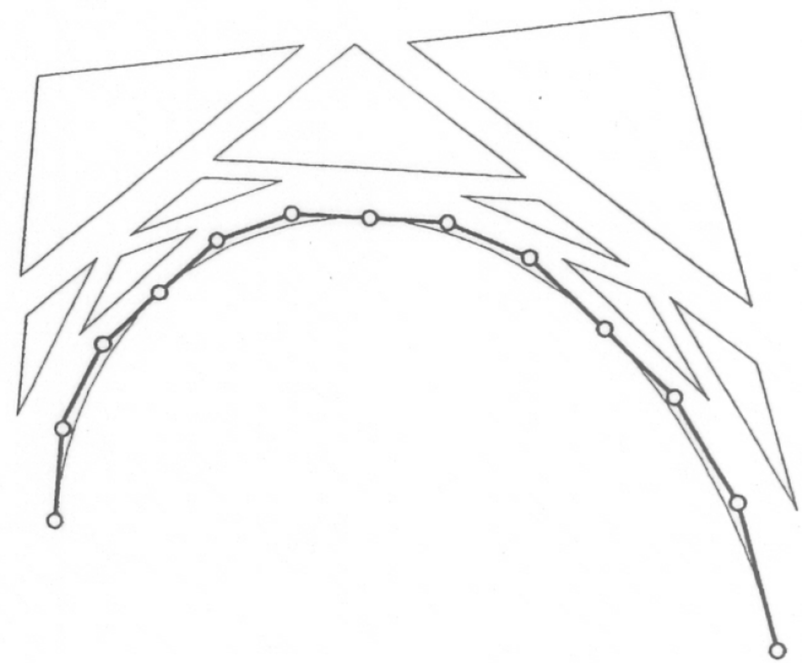
$$\begin{array}{llll} b_0 & & & \\ b_1 & b_0^1(t) & & \\ b_2 & b_1^1(t) & b_0^2(t) & \\ b_3 & b_2^1(t) & b_1^2(t) & b(t) \end{array}$$

Subdivision of Bézier curves

subdivision porperty of Bézier curves

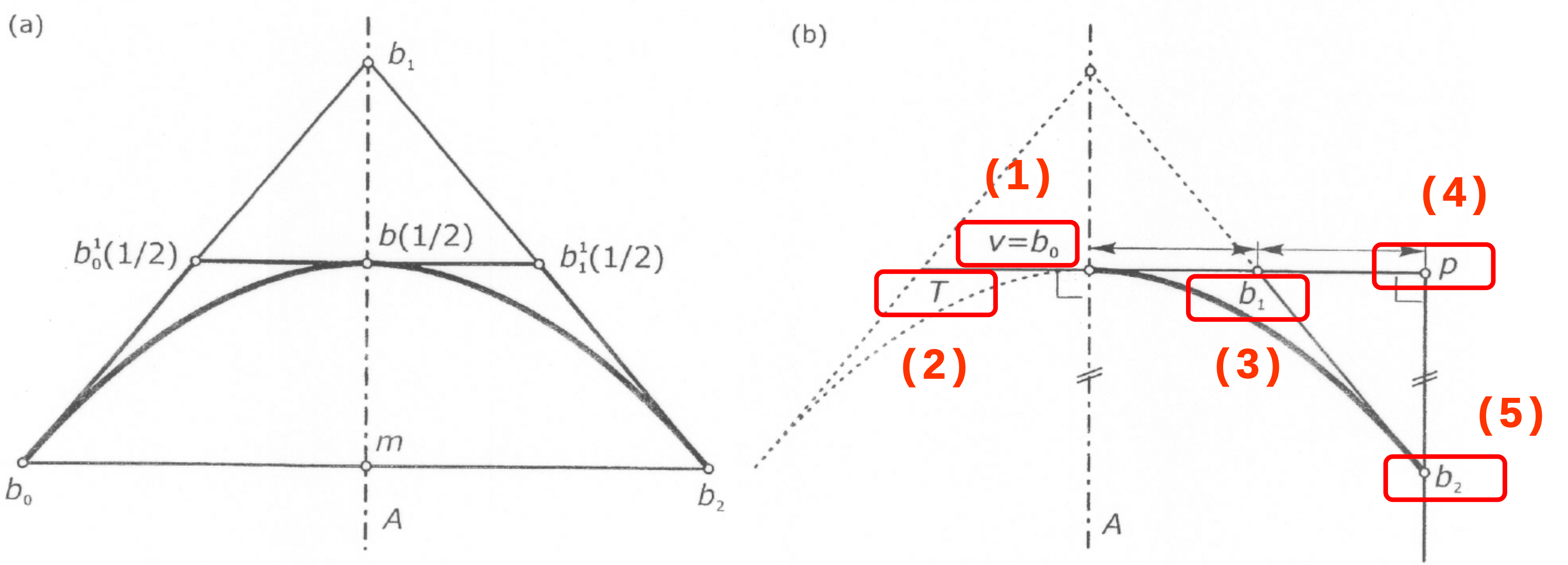


corner cutting



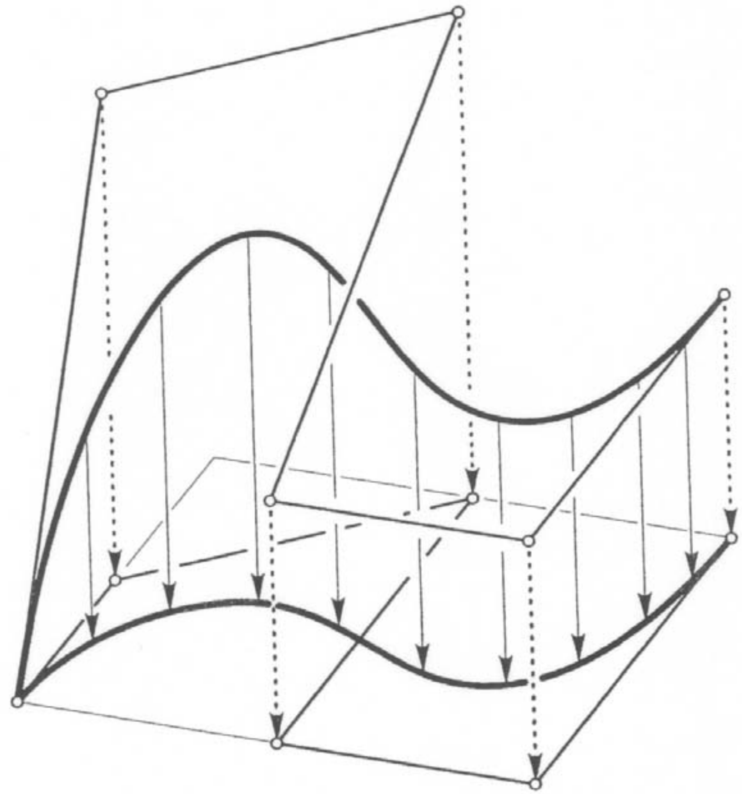
Parabolic arc via Bézier curve

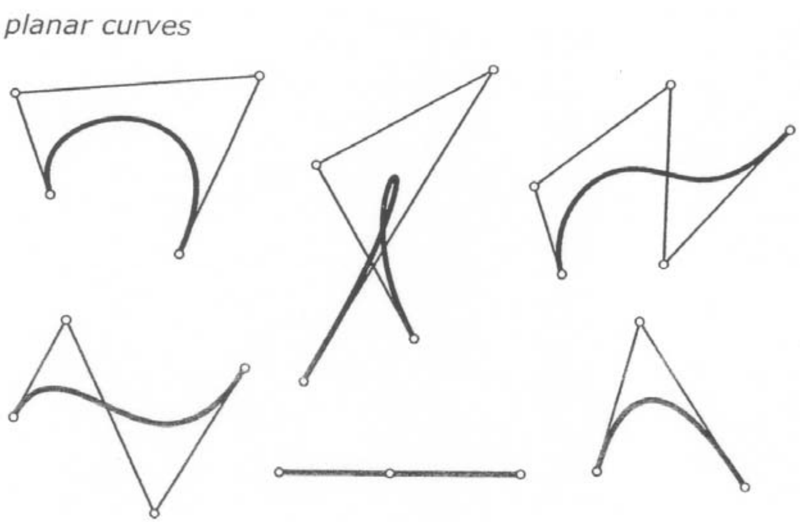
- A parabolic arc with axis A as quadratic Bézier curve
- Construction of a parabolic arc with a vertex and an axis

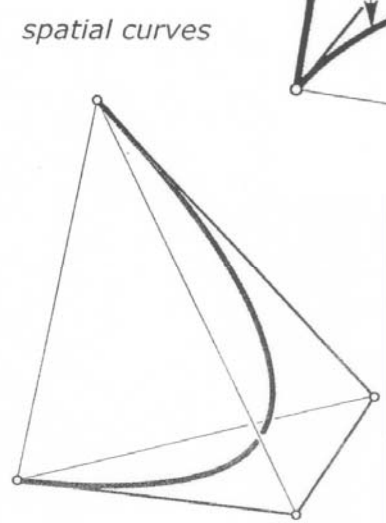


Properties of Bézier curves

- Convex hull property
  - Linear precision
- Affine invariance
- End point interpolation



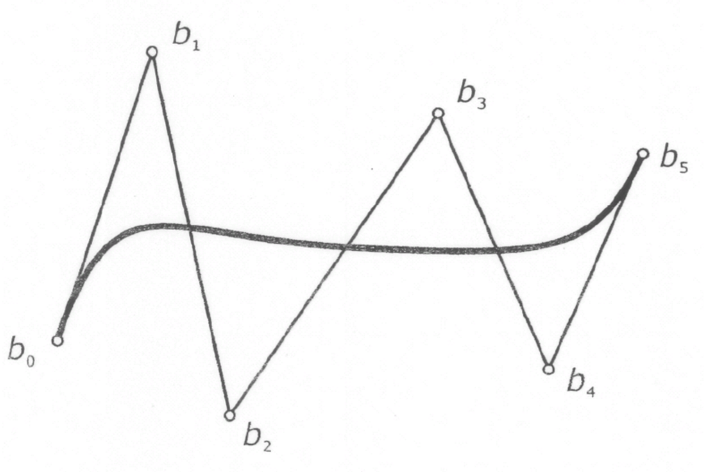


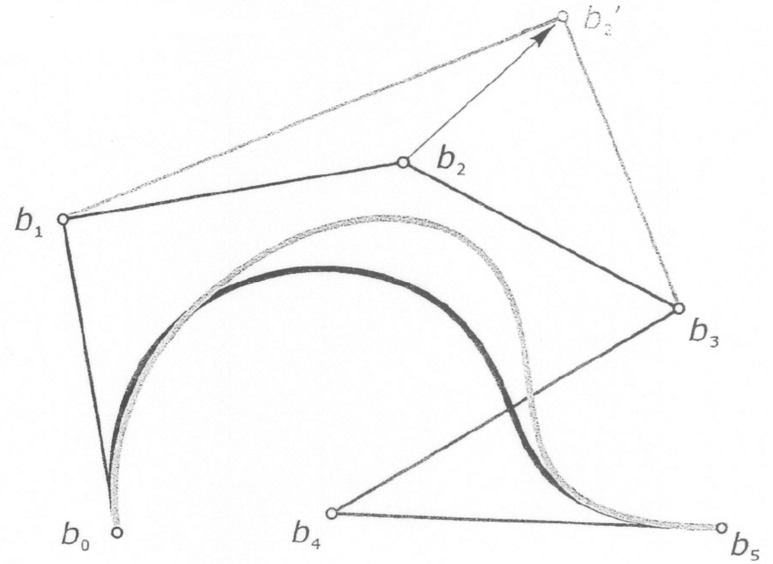


Limitations of Bézier curves

- A large number of control points is impractical.
  - Required for satisfying a large number of constraints (ex: (n-1)-degree necessary for Bézier curve through n data points)

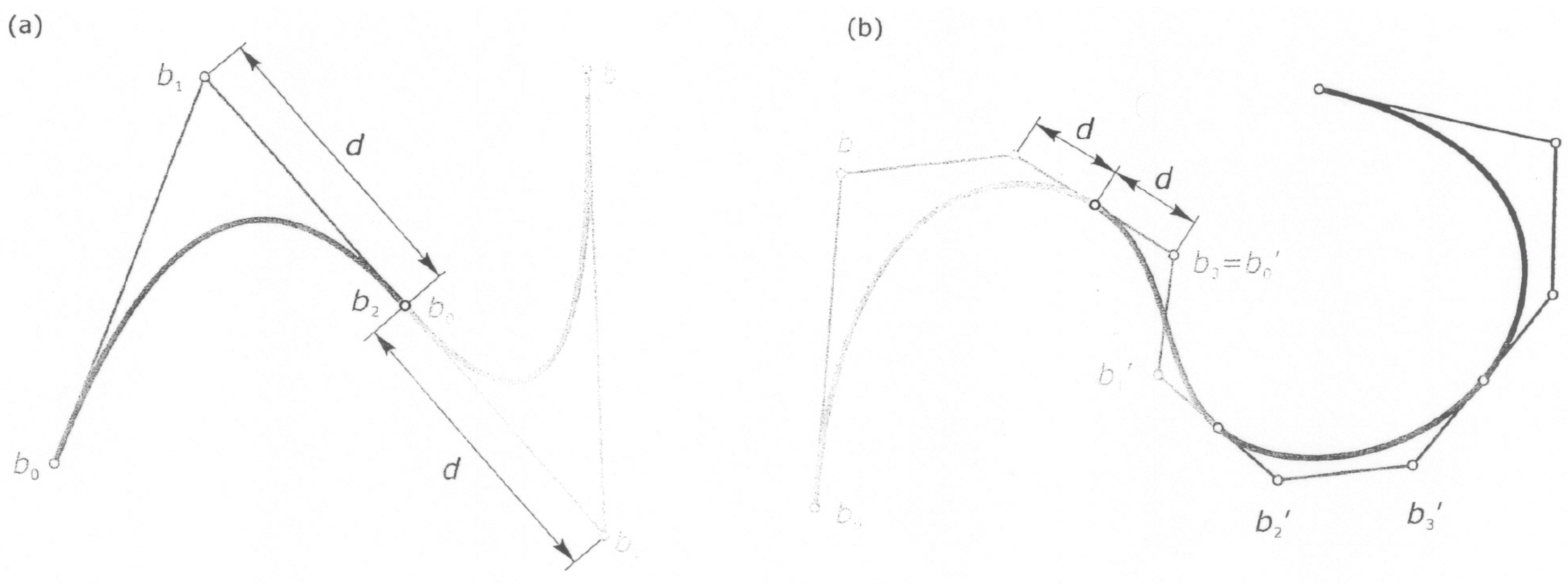
- The control points have global control on the shape of the curve.





Piecewise Bézier curves

- How to select control points in order to satisfy
  - Tangent continuity
  - Curvature continuity

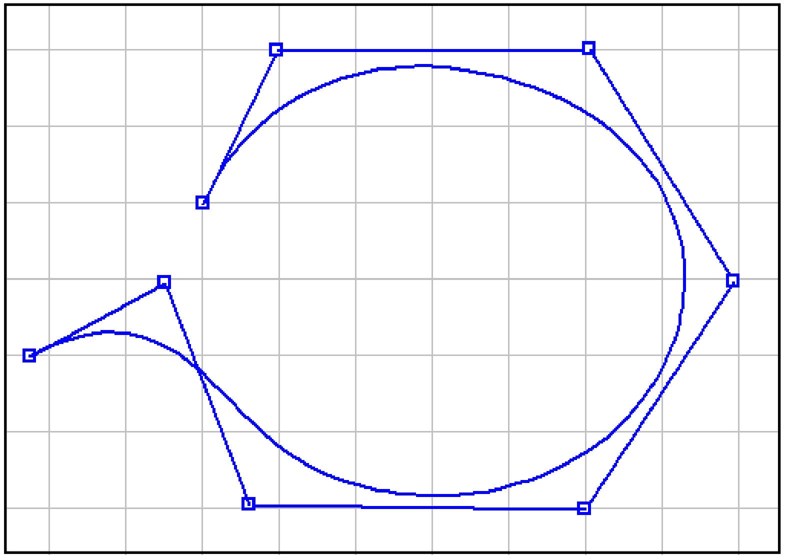


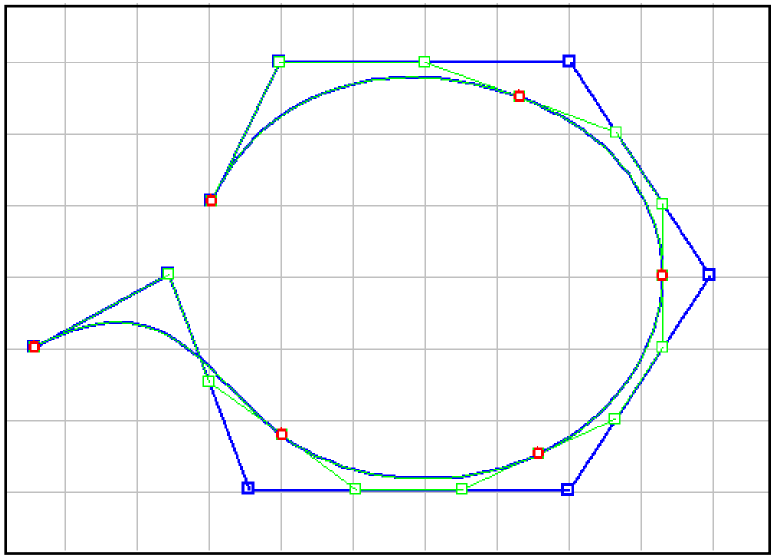
B-Spline curves

- A set of **piecewise Bézier curve segments**
  - Control points: (m + 1)
  - Degree: n
  - Knot vector with size (m + 1 + n + 1) = k
  - # of curve segments: (k – 2 * n – 1) = (m – n + 1)

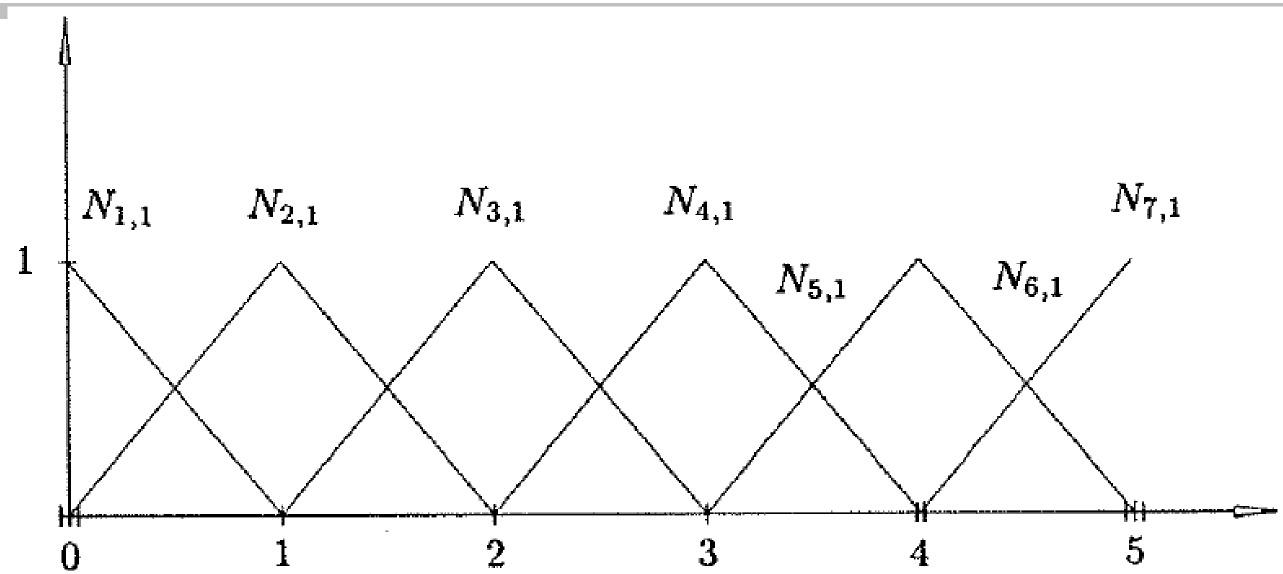
$$U = \{\underbrace{0,\cdots,0}_{n+1},\cdots,\underbrace{1,\cdots,1}_{n+1}\}$$

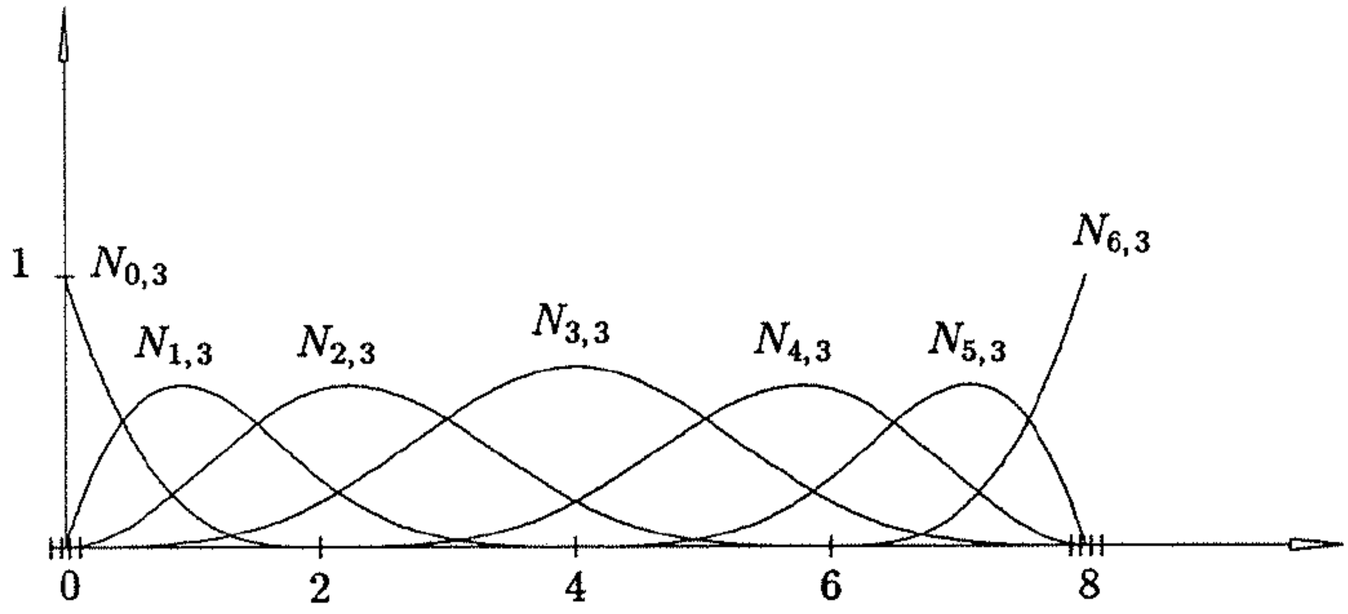
m = 7, n = 3, k = 12, **5** curve segments



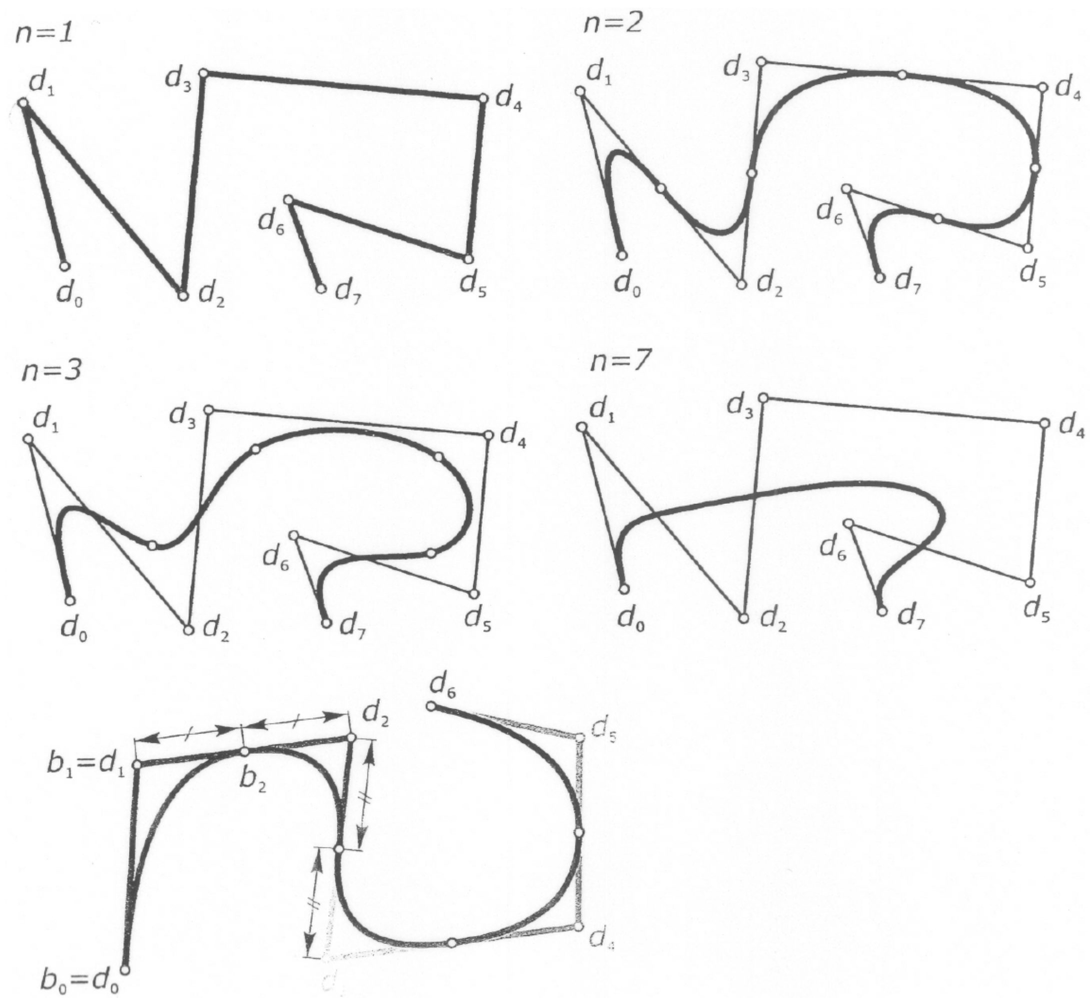


B-Spline basis functions



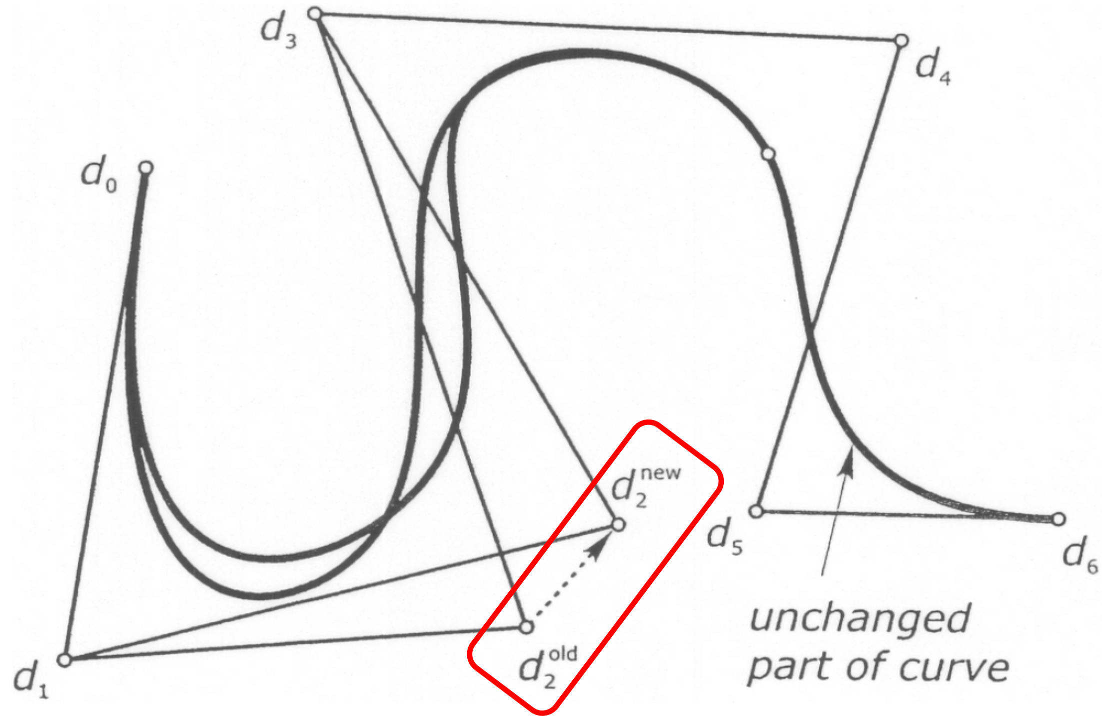


The influence of the degree



Local control on the shape

- **Modification of a control point**
  - **change only (n + 1) curve segments**



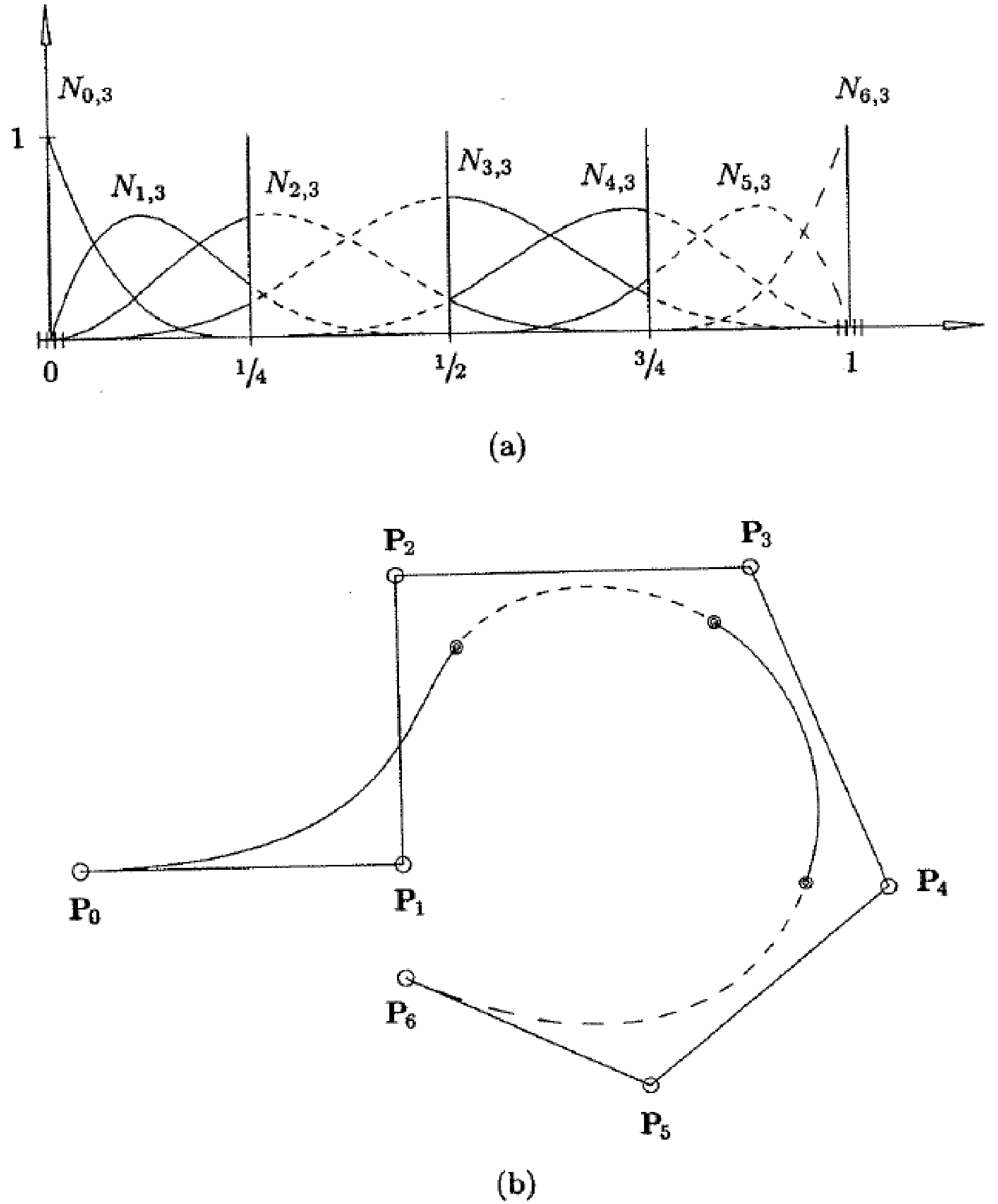
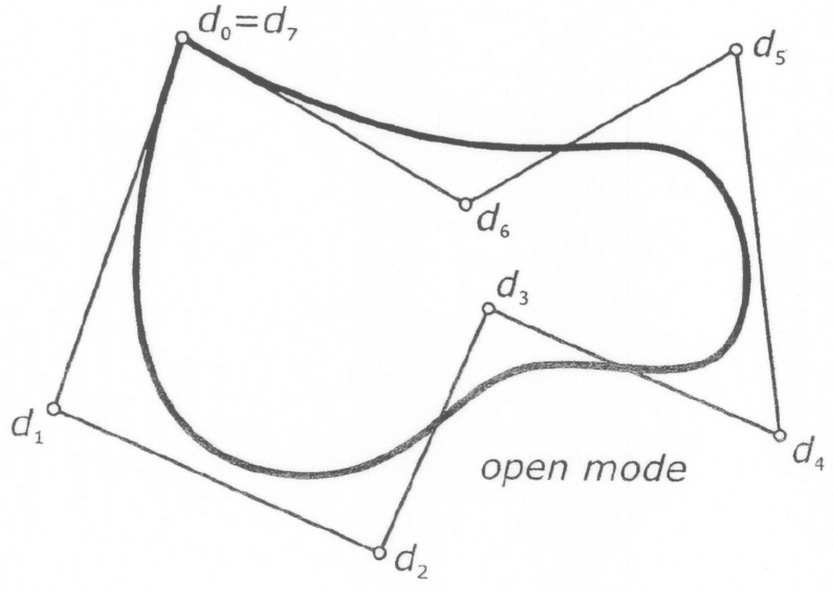
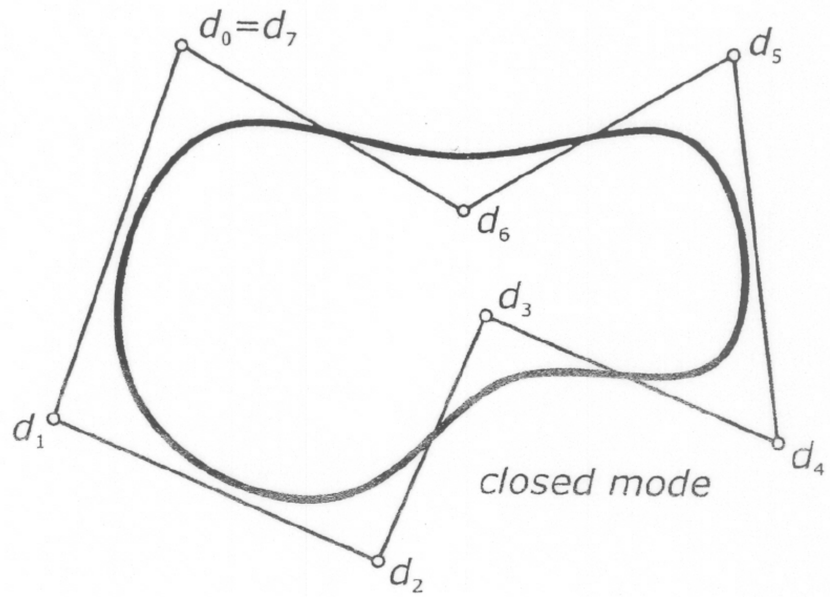


Figure 3.2. (a) Cubic basis functions $U = \{0,0,0,0,1/4,1/2,3/4,1,1,1,1\}$; (b) a cubic curve using the basis functions of Figure 3.2a.

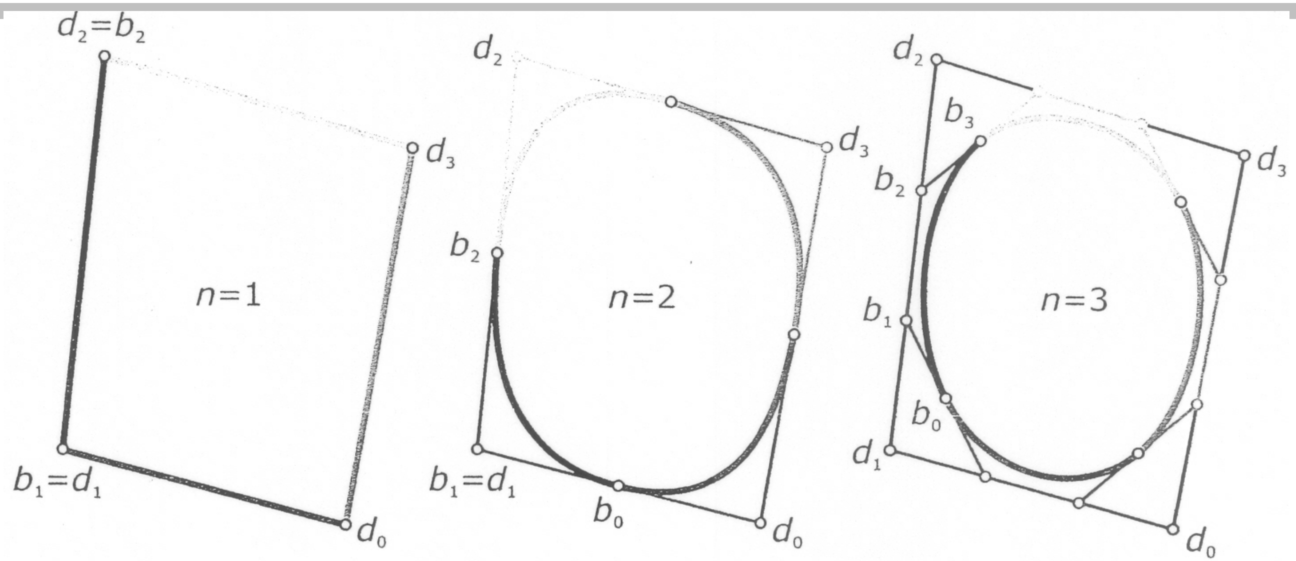
Open and closed B-Spline curves

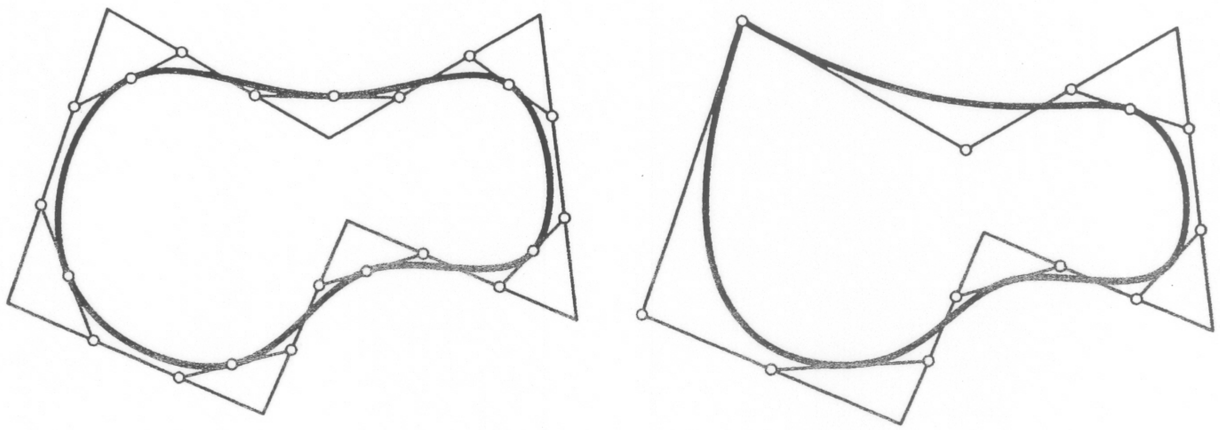
- Closed B-Spline curves
  - Control points: (m + 1)
  - Degree: n
  - # of curve segments: (m + 1)





Open and closed B-Spline curves





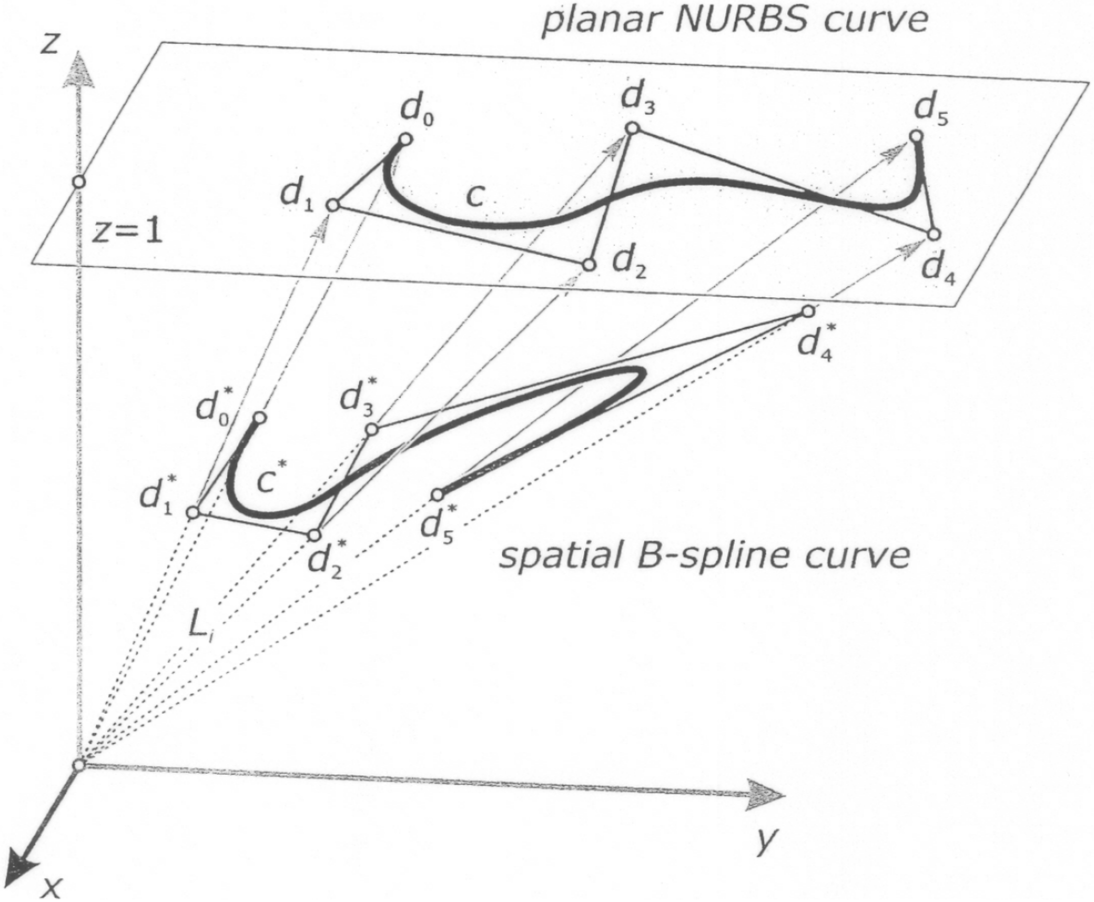
Drawback of B-Spline curves

- While a parabola can be reprsented by B-Spline curve, B-Spline curve can not represent a circle, an ellipse, and a hyperbola.
  - Rational (polynomial) is required to represent a circle, an ellipse, and a hyperbola

NURBS curves

$$C(t)=\frac{\sum_{i=0}^{n} N_{i,k}(t)w_i d_i}{\sum_{i=0}^{n} N_{i,k}(t)w_i}$$

w_i : weights



The influence of the weight

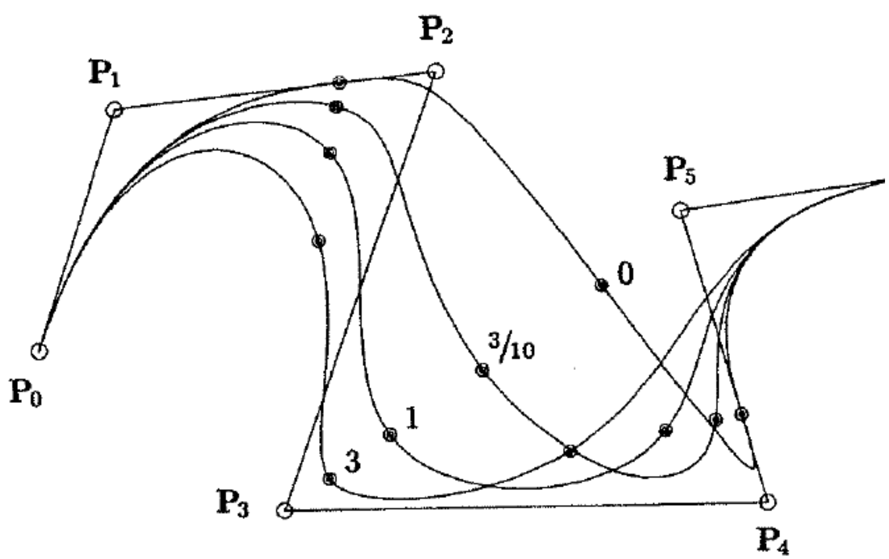


Figure 4.2. Rational cubic B-spline curves, with w_3 varying.

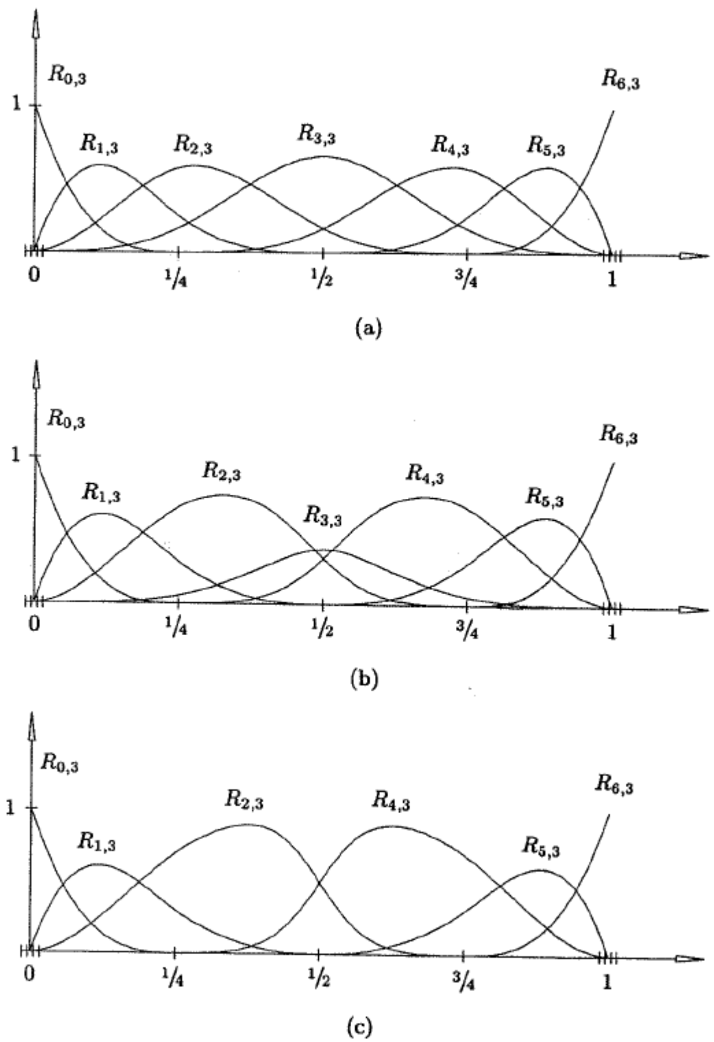
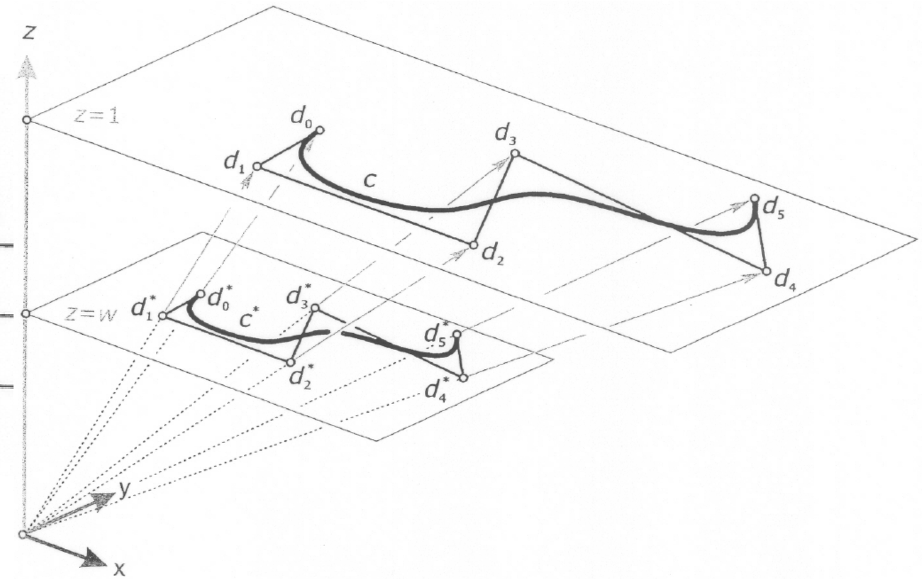


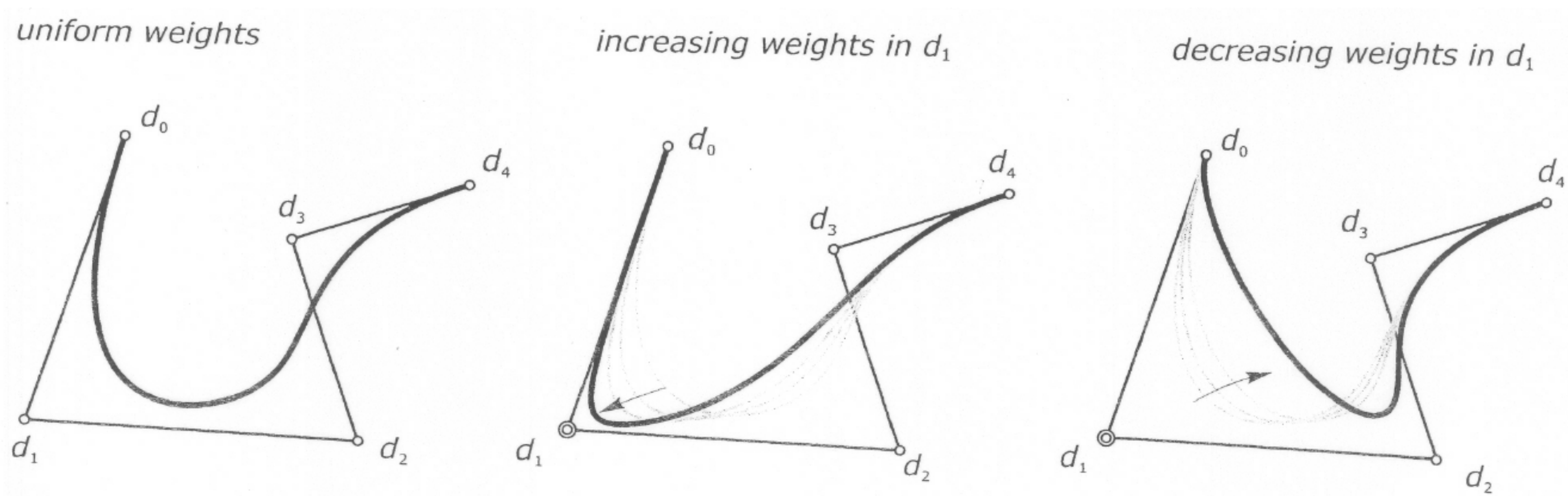
Figure 4.3. The cubic basis functions for the curves of Figure 4.2. (a) $w_3 = 1$; (b) $w_3 = {}^3/_{10}$; (c) $w_3 = 0$.

NURBS curves

Table 8.1 Design handles for freeform curves.

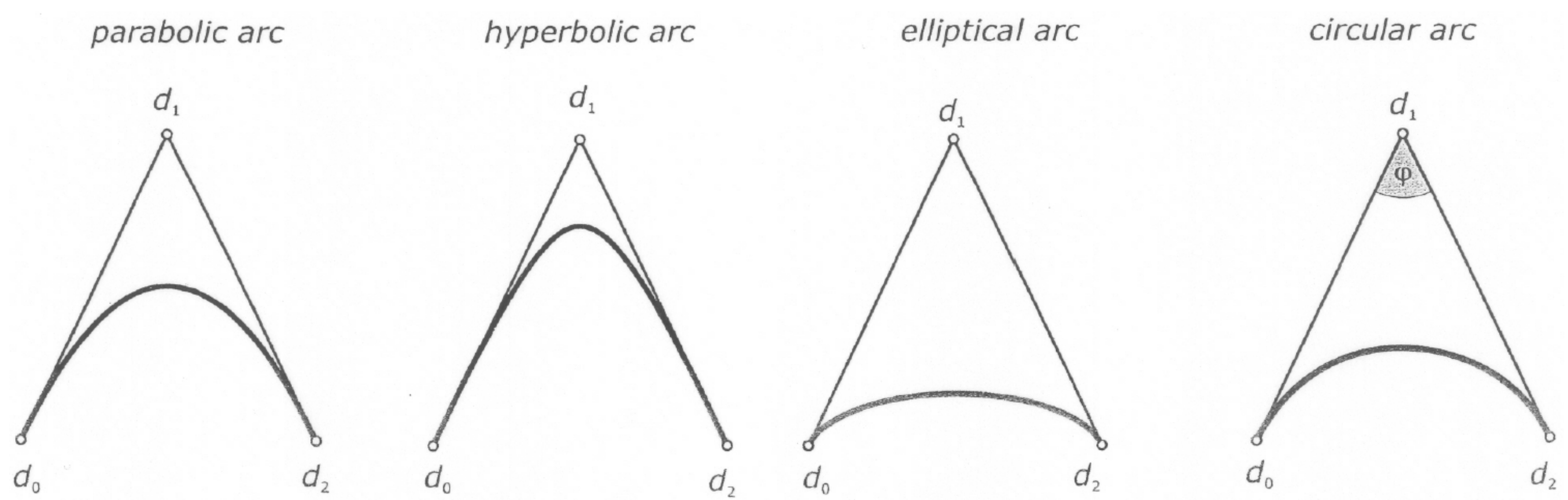
	control points	degree	weights
Bézier	✓		
B-spline	✓	✓	
NURBS	✓	✓	✓





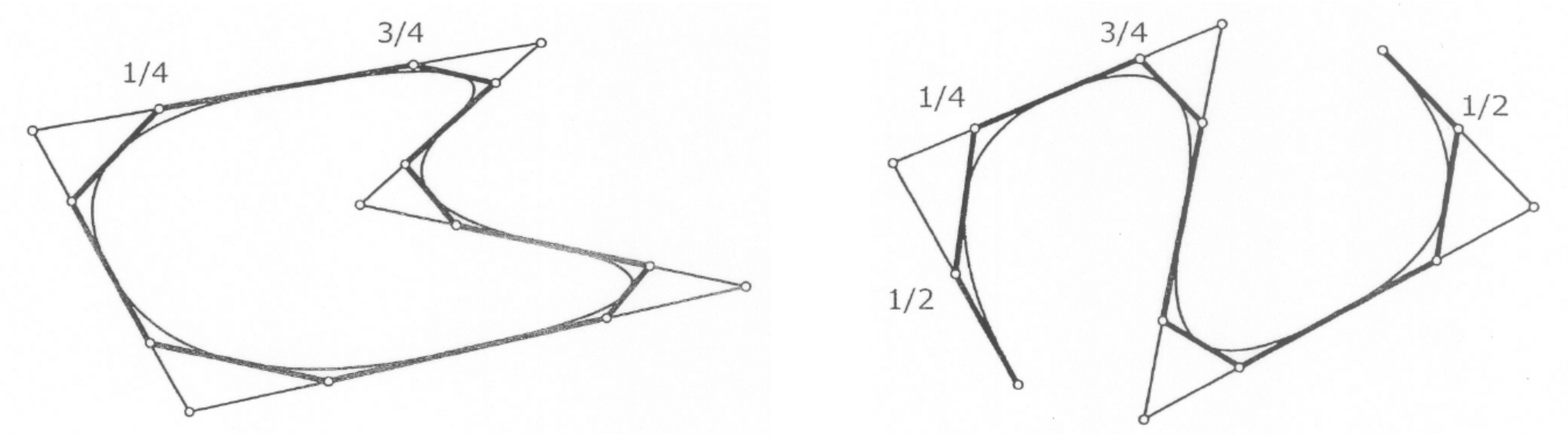
Conic sections as NURBS curves

	w_0	w_1	w_2
Parabola	1	1	1
Hyperbola	1	> 1	1
Ellipse	1	< 1	1
Circle	1	$\sin(\varphi/2)$	1



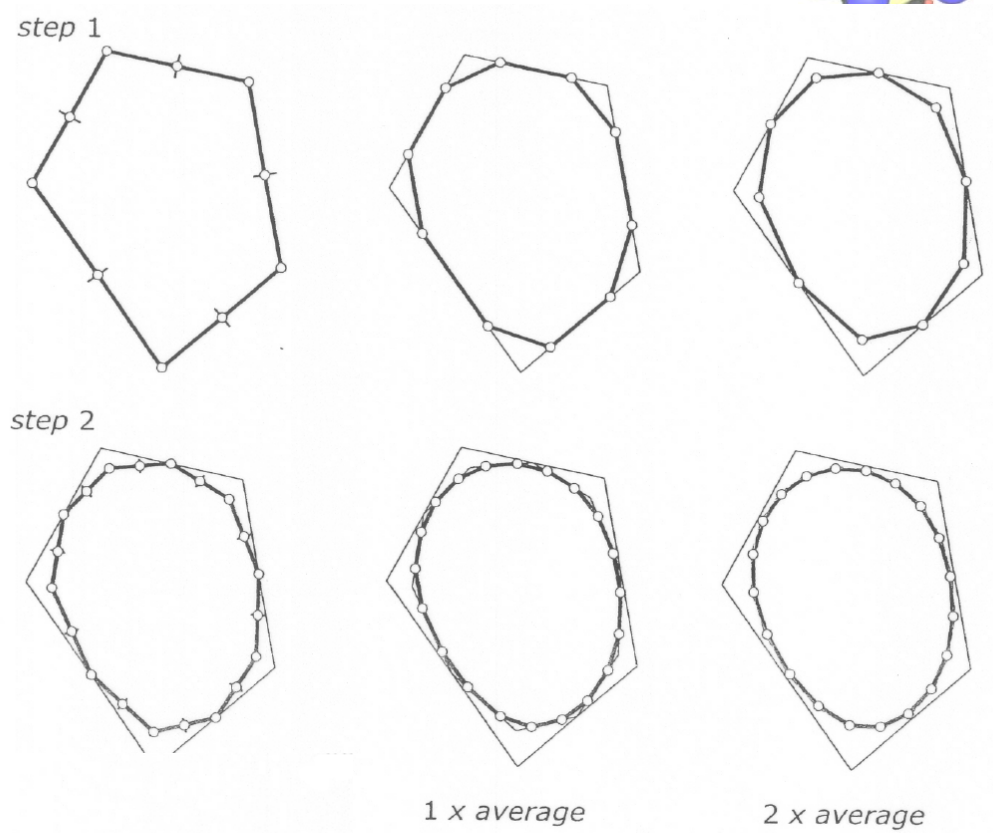
Subdivision curves

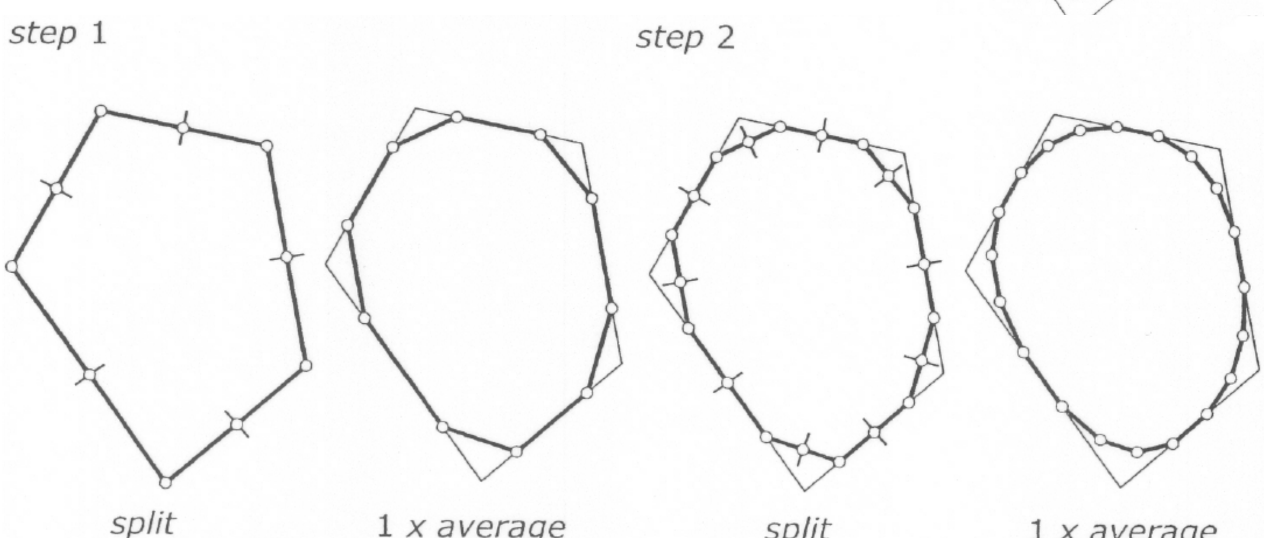
- Iteratively subdivide (refine) initial control polygon
  - Control polygon (point)
  - Subdivision (refinement) level
- Chaikin's corner-cutting algorithm
  - Uniform quadratic B-Spline curve
  - Closed curve / open curve



Subdivision curves

- **Lane-Riesenfeld algorithm**
  - **Split and *n* x average**
  - **Uniform B-Spline curve of degree (*n* + 1)**





Subdivision curves

- **The four-point scheme**

$$p_i^{new} = -w \cdot p_{i-1} + (1/2 + w) \cdot p_i + (1/2 + w) \cdot p_{i+1} - w \cdot p_{i+2}$$