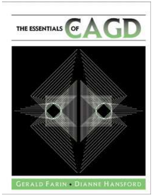
The Essentials of CAGD

Chapter 4: Bézier Curves: Cubic and Beyond

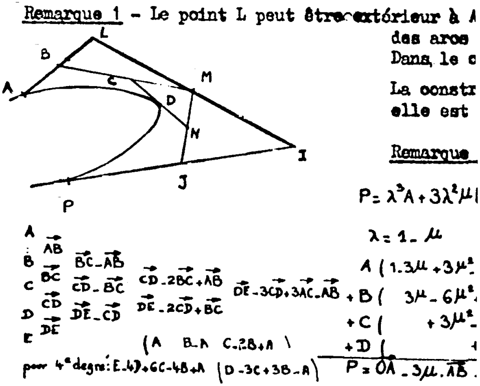
Gerald Farin & Dianne Hansford

CRC Press, Taylor & Francis Group, An A K Peters Book
www.farinhansford.com/books/essentials-cagd



Outline

Introduction to Bézier Curves: Cubic and Beyond

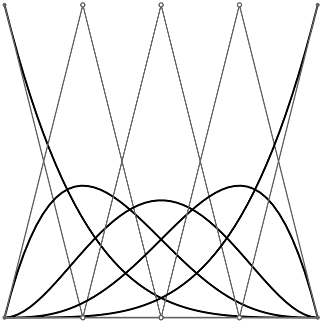


An excerpt from P. de Casteljau's writings

Bézier curves are not restricted to cubics

– Here we explore these more general curves

Bézier Curves



$B_i^4(t)$ over $[0,1]$

$(1-t)^4 \qquad 4(1-t)^3 t \qquad 6(1-t)^2 t^2 \qquad 4(1-t) t^3 \qquad t^4$

A Bézier curve of degree n

$$\mathbf{x}(t) = \mathbf{b}_0 B_0^n(t) + \mathbf{b}_1 B_1^n(t) + \ldots + \mathbf{b}_n B_n^n(t)$$

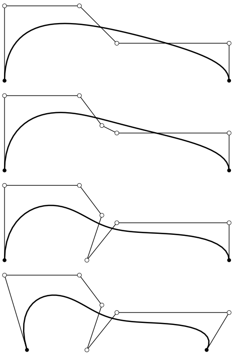
$B_i^n(t)$ are Bernstein polynomials

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

Binomial coefficients:

$$\binom{n}{i} = \begin{cases} \frac{n!}{i!(n-i)!} & \text{if} \quad 0 \le i \le n \\ 0 & \text{else} \end{cases}$$

Bézier Curves



Several examples of higher degree Bézier curves

User might influence the shape by

– Adding more control points

– Moving control points

Properties from the cubic case carry over

Derivatives Revisited

$$\dot{\mathbf{x}}(t) = n[\Delta\mathbf{b}_0 B_0^{n-1} + \ldots + \Delta\mathbf{b}_{n-1} B_{n-1}^{n-1}] \quad \text{where} \quad \Delta\mathbf{b}_i = \mathbf{b}_{i+1} - \mathbf{b}_i$$

$\Rightarrow$ A degree $n-1$ Bézier curve with *vector* coefficients

The k^{th} derivative

$$\frac{\mathrm{d}^k \mathbf{x}(t)}{\mathrm{d}t^k} = \frac{n!}{(n-k)!}[\Delta^k \mathbf{b}_0 B_0^{n-k}(t) + \ldots + \Delta^k \mathbf{b}_{n-k} B_{n-k}^{n-k}(t)]$$

Δ^k is the forward difference operator – recursively defined by

$$\Delta^k \mathbf{b}_i = \Delta^{k-1}\mathbf{b}_{i+1} - \Delta^{k-1}\mathbf{b}_i \quad \text{where} \quad \Delta^0 \mathbf{b}_i = \mathbf{b}_i$$

Examples:

$$\begin{aligned} k = 2: &\quad \mathbf{b}_{i+2} - 2\mathbf{b}_{i+1} + \mathbf{b}_i \\ k = 3: &\quad \mathbf{b}_{i+3} - 3\mathbf{b}_{i+2} + 3\mathbf{b}_{i+1} - \mathbf{b}_i \\ k = 4: &\quad \mathbf{b}_{i+4} - 4\mathbf{b}_{i+3} + 6\mathbf{b}_{i+2} - 4\mathbf{b}_{i+1} + \mathbf{b}_i \end{aligned}$$

Derivatives Revisited

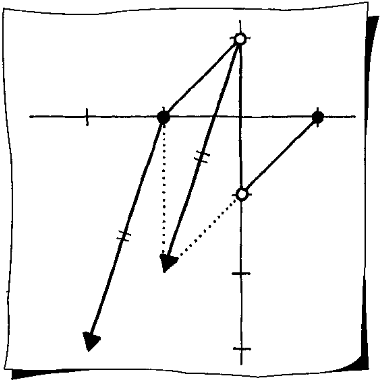
At the endpoints the derivative calculations simplify
(Abbreviated notation for the k^{th} derivative)

$$\mathbf{x}^{(k)}(0) = \frac{n!}{(n-k)!}\Delta^k \mathbf{b}_0$$

$$\mathbf{x}^{(k)}(1) = \frac{n!}{(n-k)!}\Delta^k \mathbf{b}_{n-k}$$

One nice feature of Bézier curves:
Simple geometric interpretation of the first and second derivatives at the endpoints

Derivatives Revisited



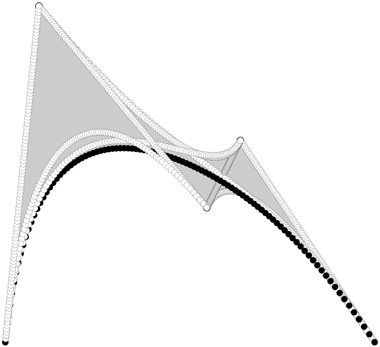
Example

$$\mathbf{x}(t) = (1-t)^3 \begin{bmatrix} -1 \\ 0 \end{bmatrix} + 3(1-t)^2 t \begin{bmatrix} 0 \\ 1 \end{bmatrix} + 3(1-t)t^2 \begin{bmatrix} 0 \\ -1 \end{bmatrix} + t^3 \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$\ddot{\mathbf{x}}(0) = 6\Delta^2 \mathbf{b}_0 = 6(\mathbf{b}_2 - 2\mathbf{b}_1 + \mathbf{b}_0)$$

$$\ddot{\mathbf{x}}(0) = 6\left(\begin{bmatrix} 0 \\ -1 \end{bmatrix} - 2\begin{bmatrix} 0 \\ 1 \end{bmatrix} + \begin{bmatrix} -1 \\ 0 \end{bmatrix}\right) = \begin{bmatrix} -6 \\ -18 \end{bmatrix}$$

The de Casteljau Algorithm Revisited



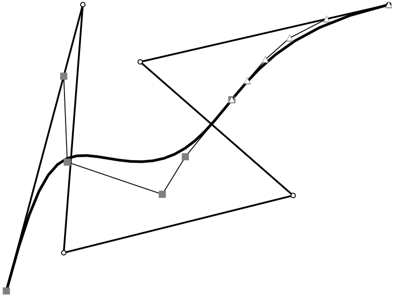
Evaluation of a degree n Bézier curve via the de Casteljau algorithm
for $r = 1, \ldots, n$
  for $i = 0, \ldots, n - r$

$$\mathbf{b}_i^r(t) = (1 - t)\mathbf{b}_i^{r-1} + t\mathbf{b}_{i+1}^{r-1}$$

Point on the curve:
$\mathbf{x}(t) = \mathbf{b}_0^n(t)$

Several de Casteljau algorithm evaluations of a degree four Bézier curve
Note locus of each $\mathbf{b}_i^r(t)$

The de Casteljau Algorithm Revisited



Quintic curve subdivided at $t = 3/4$

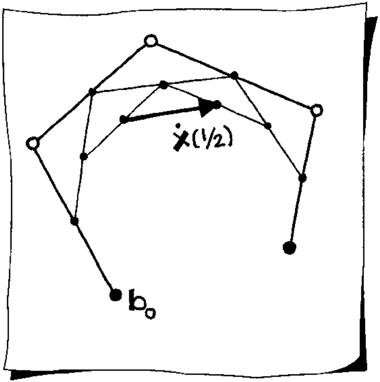
The de Casteljau algorithm *subdivides* the curve into a "left" and a "right" segment

$$\mathbf{b}_0, \mathbf{b}_0^1, \ldots, \mathbf{b}_0^n$$

$$\mathbf{b}_0^n, \mathbf{b}_1^{n-1}, \ldots, \mathbf{b}_n$$

Recall: these are points along diagonal and base of the schematic triangular diagram

The de Casteljau Algorithm Revisited



First derivative of a quartic curve at $t = 1/2$

The de Casteljau algorithm provides a way for computing the first derivative

$$\dot{\mathbf{x}}(t) = n[\mathbf{b}_1^{n-1} - \mathbf{b}_0^{n-1}]$$

Difference of the points in the next to last step

The de Casteljau Algorithm Revisited

Second derivative can also be extracted from the de Casteljau algorithm

$$\ddot{\mathbf{x}}(t) = n(n-1)[\mathbf{b}_2^{n-2} - 2\mathbf{b}_1^{n-2} + \mathbf{b}_0^{n-2}]$$

A scaling of the second difference of the $(n-2)^{\mathrm{nd}}$ column in the schematic triangular diagram

The Matrix Form and Monomials Revisited

Sometimes convenient to write a Bézier curve in *matrix form*

Generalizing the cubics
Define two vectors N and $\mathbf{B}$ by

$$N = \begin{bmatrix} B_0^n(t) \\ \vdots \\ B_n^n(t) \end{bmatrix} \qquad \mathbf{B} = \begin{bmatrix} \mathbf{b}_0 \\ \vdots \\ \mathbf{b}_n \end{bmatrix}$$

then the Bézier curve becomes

$$\mathbf{x}(t) = N^{\mathrm{T}}\mathbf{B}$$

This notation will be useful for dealing with surfaces

The Matrix Form and Monomials Revisited

Matrix notation useful for converting between the Bernstein and monomial basis functions

Recall for cubics:

$$\mathbf{b}(t) = \begin{bmatrix} \mathbf{b}_0 & \mathbf{b}_1 & \mathbf{b}_2 & \mathbf{b}_3 \end{bmatrix} \begin{bmatrix} 1 & -3 & 3 & -1 \\ 0 & 3 & -6 & 3 \\ 0 & 0 & 3 & -3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ t \\ t^2 \\ t^3 \end{bmatrix}$$

$$\begin{bmatrix} \mathbf{a}_0 & \mathbf{a}_1 & \mathbf{a}_2 & \mathbf{a}_3 \end{bmatrix} = \begin{bmatrix} \mathbf{b}_0 & \mathbf{b}_1 & \mathbf{b}_2 & \mathbf{b}_3 \end{bmatrix} \begin{bmatrix} 1 & -3 & 3 & -1 \\ 0 & 3 & -6 & 3 \\ 0 & 0 & 3 & -3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Columns $\Rightarrow$ scaled forms of the derivative at $t = 0$

$$\mathbf{a}_0 = \mathbf{b}_0 \qquad \text{and} \qquad \mathbf{a}_i = \binom{n}{i} \Delta^i \mathbf{b}_0 \quad \text{for } i = 1 \ldots n$$

The Matrix Form and Monomials Revisited

The Bernstein form is more appealing from a geometric point of view

– Curve defined by control points which mimic the shape of the curve

Monomial form defined in terms of its derivatives

Bernstein form numerically more stable than the monomial form

Degree Elevation

A degree n polynomial is also one of degree $n+1$
– Leading monomial form coefficient is zero

A quadratic Bézier curve to demonstrate

$$\mathbf{x}(t) = (1-t)^2\mathbf{b}_0 + 2(1-t)t\mathbf{b}_1 + t^2\mathbf{b}_2$$

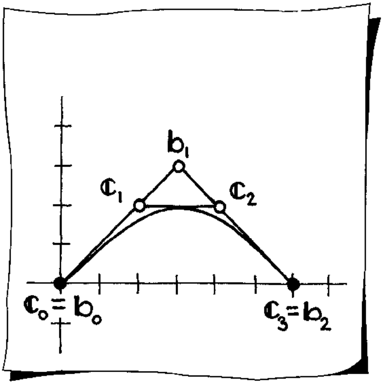
Trick: multiply the quadratic expression by $[t + (1-t)]$

Results in a cubic curve with control points

$$\mathbf{x}(t) = B_0^3\mathbf{b}_0 + B_1^3[\frac{1}{3}\mathbf{b}_0 + \frac{2}{3}\mathbf{b}_1] + B_2^3[\frac{2}{3}\mathbf{b}_1 + \frac{1}{3}\mathbf{b}_2] + B_3^3\mathbf{b}_2$$

Trace of the cubic form of curve identical to original quadratic

Degree Elevation



Curve drawn incorrectly

Max y-value occurs at $\begin{bmatrix} 3 \\ 3/2 \end{bmatrix}$

Example: Quadratic Bézier curve

$$\mathbf{b}_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \mathbf{b}_1 = \begin{bmatrix} 3 \\ 3 \end{bmatrix} \quad \mathbf{b}_2 = \begin{bmatrix} 6 \\ 0 \end{bmatrix}$$

Degree elevation results in

$$\mathbf{x}(t) = \mathbf{c}_0 B_0^3 + \mathbf{c}_1 B_1^3 + \mathbf{c}_2 B_2^3 + \mathbf{c}_3 B_3^3$$

$$\mathbf{c}_0 = \mathbf{b}_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\mathbf{c}_1 = [\frac{1}{3}\mathbf{b}_0 + \frac{2}{3}\mathbf{b}_1] = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$$

$$\mathbf{c}_2 = [\frac{2}{3}\mathbf{b}_1 + \frac{1}{3}\mathbf{b}_2] = \begin{bmatrix} 4 \\ 2 \end{bmatrix}$$

$$\mathbf{c}_3 = \mathbf{b}_2 = \begin{bmatrix} 6 \\ 0 \end{bmatrix}$$

Degree Elevation

Degree n Bézier curve with control polygon $\mathbf{b}_0, \ldots, \mathbf{b}_n$

Degree elevate to Bézier curve with control polygon $\mathbf{c}_0, \ldots, \mathbf{c}_{n+1}$

$$\begin{aligned}
\mathbf{c}_0 &= \mathbf{b}_0 \\
&\vdots \\
\mathbf{c}_i &= \frac{i}{n+1}\mathbf{b}_{i-1} + (1 - \frac{i}{n+1})\mathbf{b}_i \\
&\vdots \\
\mathbf{c}_{n+1} &= \mathbf{b}_n
\end{aligned}$$

Degree Elevation

Written as a matrix operation

$$\begin{bmatrix} 1 & & & & & \\ \star & \star & & & & \\ & \star & \star & & & \\ & & & \vdots & \vdots & \\ & & & & \star & \star \\ & & & & & 1 \end{bmatrix} \begin{bmatrix} \mathbf{b}_0 \\ \vdots \\ \mathbf{b}_n \end{bmatrix} = \begin{bmatrix} \mathbf{c}_0 \\ \vdots \\ \mathbf{c}_{n+1} \end{bmatrix}$$

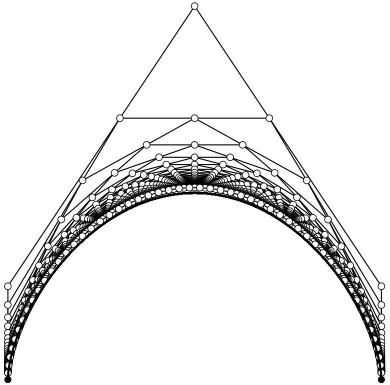
Abbreviated:

$$D\mathbf{B} = \mathbf{C} \qquad D \text{ is } (n+2) \times (n+1)$$

Example:

$$\begin{bmatrix} 1 & 0 & 0 \\ 1/3 & 2/3 & 0 \\ 0 & 2/3 & 1/3 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{b}_0 \\ \mathbf{b}_1 \\ \mathbf{b}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{c}_0 \\ \mathbf{c}_1 \\ \mathbf{c}_2 \\ \mathbf{c}_3 \end{bmatrix}$$

Degree Elevation



Degree elevation may be applied repeatedly

Resulting sequence of control polygons will converge to the curve

Convergence is too slow for practical purposes

Degree Reduction

Some CAD systems allow up to degree 40 and others use degree 3 only

Reducing a degree 40 curve to a cubic is not trivial
– In practice several degree 3 curves needed
  $\Rightarrow$ Interplay between subdivision and degree reduction

Degree Reduction

Must *approximate* a degree $n+1$ curve by degree n curve

Recall degree elevation

$$\begin{bmatrix} 1 & & & & \\ \star & \star & & & \\ & \star & \star & & \\ & & \vdots & \vdots & \\ & & & \star & \star \\ & & & & 1 \end{bmatrix} \begin{bmatrix} \mathbf{b}_0 \\ \vdots \\ \mathbf{b}_n \end{bmatrix} = \begin{bmatrix} \mathbf{c}_0 \\ \vdots \\ \mathbf{c}_{n+1} \end{bmatrix} \qquad D\mathbf{B} = \mathbf{C}$$

Degree reduction: Given $\mathbf{C}$ then find $\mathbf{B}$

Problem: D not a square matrix $\rightarrow$ cannot invert

Solution: multiply both sides by D^{T}

$$D^{\mathrm{T}} D\mathbf{B} = D^{\mathrm{T}}\mathbf{C}$$

Degree Reduction

Example: Revisit degree elevation example

$$\begin{bmatrix} 1 & 0 & 0 \\ 1/3 & 2/3 & 0 \\ 0 & 2/3 & 1/3 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{b}_0 \\ \mathbf{b}_1 \\ \mathbf{b}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{c}_0 \\ \mathbf{c}_1 \\ \mathbf{c}_2 \\ \mathbf{c}_3 \end{bmatrix}$$

$$D^{\mathrm{T}} D = \frac{1}{9} \begin{bmatrix} 10 & 2 & 0 \\ 2 & 8 & 2 \\ 0 & 2 & 10 \end{bmatrix} \qquad D^{\mathrm{T}} C = \frac{1}{3} \begin{bmatrix} 2 & 2 \\ 12 & 8 \\ 22 & 2 \end{bmatrix}$$

First column of $D^{\mathrm{T}} C$ corresponds to the $x-$components
Second column corresponds to the $y-$components
– x and y sent separately to linear system solver

$$B = \begin{bmatrix} 0 & 0 \\ 3 & 3 \\ 6 & 0 \end{bmatrix} \qquad \Rightarrow \mathbf{b}_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \mathbf{b}_1 = \begin{bmatrix} 3 \\ 3 \end{bmatrix} \quad \mathbf{b}_2 = \begin{bmatrix} 6 \\ 0 \end{bmatrix}$$

Degree Reduction

Degree reduction via solution to

$$D^{\mathrm{T}} D\mathbf{B} = D^{\mathrm{T}}\mathbf{C}$$

In general will not pass through the original curve endpoints $\mathbf{c}_0$ and $\mathbf{c}_{n+1}$

Could be enforced after solving the linear system
$\Rightarrow$ *endpoint interpolation*

In the example above:
Automatic because the cubic was a degree elevated quadratic!

Bézier Curves over General Intervals

Associate a Bézier curve with the parameter interval $[a, b]$ rather than $[0, 1]$

Let u be the *global parameter* associated with $[a, b]$

Parameter transformation

$$t = \frac{u - a}{b - a}$$

$\Rightarrow$ *Local parameter t*

Local parameter needed for de Casteljau algorithm evaluation
– Obtain the curve segment that runs from $\mathbf{b}_0$ to $\mathbf{b}_n$
– Trace of Bézier curve same regardless of the parameter interval associated with it

Plugging global parameter into the de Casteljau algorithm results in *extrapolation*

Functional Bézier Curves

Recall graphs of functions versus parametric curves in Chapter 3
– Parametric curve is more general than a functional curve
– Graph of a functional curve can be thought of as a parametric curve

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} t \\ g(t) \end{bmatrix}$$

$\Rightarrow$ One dimension restricted to be a linear polynomial

Another name for a functional curve: nonparametric curve

Functional Bézier Curves

Write a (polynomial) functional curve in Bézier form

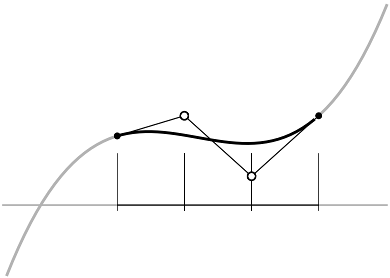
For now: $t \in [0,1]$

$$g(t) = b_0 B_0^n + \ldots + b_n B_n^n$$

b_i are scalar values: Bézier ordinates

Remains to write the linear polynomial t as a degree n polynomial
– Match the degree of $g(t)$

Functional Bézier Curves



Bézier curves have *linear precision*

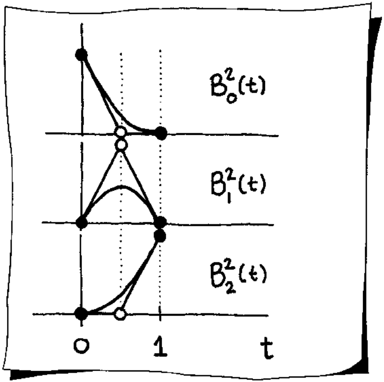
Degree n linear interpolant requires evenly spaced control points
$\Rightarrow$ Abscissa values evenly spaced

$$\mathbf{b}(t) = \begin{bmatrix} 0 \\ b_0 \end{bmatrix} B_0^n + \ldots + \begin{bmatrix} j/n \\ b_j \end{bmatrix} B_j^n + \ldots + \begin{bmatrix} 1 \\ b_n \end{bmatrix} B_n^n$$

Function with $t \in [a, b]$
$\Rightarrow$ abscissa values

$$a + j\frac{(b-a)}{n} \quad j = 0, \ldots, n$$

More on Bernstein Polynomials



Quadratic Bernstein polynomials
(See earlier slide for degree 4 figure)

Look a little closer at B_i^n to understand the behavior of Bézier curves

de Casteljau algorithm generally preferred for the evaluation

To plot the Bernstein polynomials:
– Note they are functions
– Formulate them as Bézier curves

More on Bernstein Polynomials

The binomial coefficients $\binom{n}{i}$ look complicated

Pascal's triangle

$$\begin{array}{ccccccccccc}
 & & & & & 1 & & & & & \\
 & & & & 1 & & 1 & & & & \\
 & & & 1 & & 2 & & 1 & & & \\
 & & 1 & & 3 & & 3 & & 1 & & \\
 & 1 & & 4 & & 6 & & 4 & & 1 & \\
1 & & 5 & & 10 & & 10 & & 5 & & 1
\end{array}$$

Each element in a row generated by adding the two elements in the previous row that lie above the element

For degree n: take from the $(n+1)^{\mathrm{st}}$ row

$$B_i^4: \qquad (1-t)^4 \qquad 4(1-t)^3 t \qquad 6(1-t)^2 t^2 \qquad 4(1-t)t^3 \qquad t^4$$

More on Bernstein Polynomials

The Bernstein polynomials also called Bernstein basis functions
– Monomials are another example of basis functions

A set of polynomials of degree n that form a basis allow you to write any polynomial of degree less than or equal to n in terms of a unique combination of the basis functions

More on Bernstein Polynomials

Partition of unity property:
For any particular value of t

$$B_0^n(t) + \ldots + B_n^n(t) = 1$$

Useful identity to keep in mind when debugging a program!

Each Bernstein polynomial is *nonnegative* within the interval $[0, 1]$

Nonnegative property and partition of unity property
$\Rightarrow$ Bézier curves have the *convex hull property*

More on Bernstein Polynomials

Symmetry in the Bernstein polynomials:

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

$\Rightarrow$ Bézier curves have the symmetry property

Bézier points numbered from "left to right" or "right to left"
$\Rightarrow$ Same curve geometry