## 3.2 Cubic Bézier Curves

Ex 3.3

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

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

The polynomial curve is expressed in terms of a combination of points. We may compute $\mathbf{x}(0.5) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$

A cubic Bézier curve is defined by

$$\mathbf{x}(t) = (1-t)^3 \mathbf{b}_0 + 3(1-t)^2 t\, \mathbf{b}_1 + 3(1-t)t^2 \mathbf{b}_2 + t^3 \mathbf{b}_3,$$

where $\mathbf{b}_i$, the Bézier control points, form the Bézier polygon of the curve.

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

↳ the cubic Bernstein polynomials.

Properties of Cubic Bézier Curves

1. Endpoint interpolation: $\mathbf{x}(0) = \mathbf{b}_0$, $\mathbf{x}(1) = \mathbf{b}_3$.
2. Symmetry: Two polygons $\mathbf{b}_0, \mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3$ and $\mathbf{b}_3, \mathbf{b}_2, \mathbf{b}_1, \mathbf{b}_0$ describe the same curve; but the directions are different
3. Invariance under rotations:
4. Invariance under affine maps: If an affine map is applied to the control polygon, the curve is mapped by the same map.
5. Convex hull property: For $t \in [0,1]$, the point $\mathbf{x}(t)$ is in the convex hull of the control polygon.
6. Linear precision: If $\mathbf{b}_1 = \frac{2}{3}\mathbf{b}_0 + \frac{1}{3}\mathbf{b}_3$ and $\mathbf{b}_2 = \frac{1}{3}\mathbf{b}_0 + \frac{2}{3}\mathbf{b}_3$, then the curve $\mathbf{x}(t) = (1-t)\mathbf{b}_0 + t\,\mathbf{b}_3$ : linear interpolation.

o. For $t \notin [0,1]$, $\mathbf{x}(t)$ may not stay within the convex hull of the control polygon.

## 3.3 Derivatives

$$\mathbf{x}'(t) = -3(1-t)^2\mathbf{b}_0 + [3(1-t)^2 - 6(1-t)t]\mathbf{b}_1 + [6(1-t)t - 3t^2]\mathbf{b}_2 + 3t^2\mathbf{b}_3$$

$$= (1-t)^2 \cdot 3[\mathbf{b}_1 - \mathbf{b}_0] + 2(1-t)t \cdot 3[\mathbf{b}_2 - \mathbf{b}_1] + t^2 \cdot 3[\mathbf{b}_3 - \mathbf{b}_2]$$

$$= (1-t)^2 \cdot 3\Delta\mathbf{b}_0 + 2(1-t)t \cdot 3\Delta\mathbf{b}_1 + t^2 \cdot 3\Delta\mathbf{b}_2$$

↪ the forward difference

$$= 3\left(B_0^2(t)\cdot\Delta\mathbf{b}_0 + B_1^2(t)\cdot\Delta\mathbf{b}_1 + B_2^2(t)\cdot\Delta\mathbf{b}_2\right)$$

↪ the quadratic Bernstein basis functions

- For parametric curves, "derivative curves" produce vectors rather than points.
- The coefficients are the difference vectors of the polygon, scaled by 3, the degree of the curve.
- $\mathbf{x}'(0) = 3\Delta\mathbf{b}_0$, $\mathbf{x}'(1) = 3\Delta\mathbf{b}_2$

## 3.4 The de Casteljau Algorithm

$$\begin{cases} \mathbf{b}_0^1(t) = (1-t)\mathbf{b}_0 + t\,\mathbf{b}_1 \\ \mathbf{b}_1^1(t) = (1-t)\mathbf{b}_1 + t\,\mathbf{b}_2 \\ \mathbf{b}_2^1(t) = (1-t)\mathbf{b}_2 + t\,\mathbf{b}_3 \end{cases} \Rightarrow \begin{cases} \mathbf{b}_0^2(t) = (1-t)\mathbf{b}_0^1(t) + t\,\mathbf{b}_1^1(t) \\ \mathbf{b}_1^2(t) = (1-t)\mathbf{b}_1^1(t) + t\,\mathbf{b}_2^1(t) \end{cases}$$

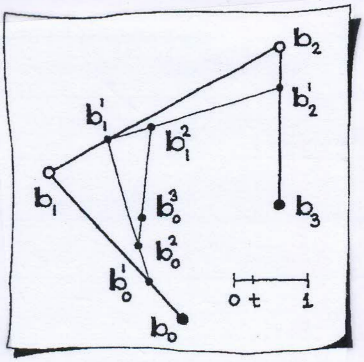
$$\Rightarrow \underbrace{\mathbf{b}_0^3(t) = (1-t)\mathbf{b}_0^2(t) + t\,\mathbf{b}_1^2(t)}_{\mathbf{x}(t)}.$$

### A convenient schematic tool for the algorithm

$$\begin{array}{llll} \mathbf{b}_0 & & & \\ & \mathbf{b}_0^1(t) & & \\ \mathbf{b}_1 & & \mathbf{b}_0^2(t) & \\ & \mathbf{b}_1^1(t) & & \mathbf{b}_0^3(t) \\ \mathbf{b}_2 & & \mathbf{b}_1^2(t) & \\ & \mathbf{b}_2^1(t) & & \\ \mathbf{b}_3 & & & \end{array}$$

- In the implementation, $\mathbf{b}_0^1$ is calculated and loaded into $\mathbf{b}_0$ since $\mathbf{b}_0$ is never needed.
  ⇒ 1D array of control points is sufficient!

- $\mathbf{x}'(t) = 3[\mathbf{b}_1^2(t) - \mathbf{b}_0^2(t)]$ :
  The derivative is essentially a byproduct of point evaluation!




**Sketch 27.**
The de Casteljau algorithm.

## 3.5 Subdivision

- Two polygons $\mathbb{b}_0, \mathbb{b}_0^1, \mathbb{b}_0^2, \mathbb{b}_0^3$ and $\mathbb{b}_0^3, \mathbb{b}_1^2, \mathbb{b}_2^1, \mathbb{b}_3$ define two segments corresponding to $[0,t]$ and $[t,1]$, respectively.
- Subdivision at $t=0.5$ splits the curve at the parameter midpoint, but the two arcs are not of equal length.
- Subdivision may be repeated:
  Each of the two new control polygons may be subdivided, ...
  The resulting sequence of control polygons converges to the curve.
- Another application is the intersection of a curve with a line.
  1. Find the AABB (axis aligned bounding box) of the polygon.
  2. If no intersection between the AABB and the line, EXIT.
  3. Else if the AABB is smaller than $\varepsilon$, report the center of the AABB.
  4. Else, subdivide the curve at $t=0.5$ into two and repeat the same procedure to each segment.
- The curve-curve intersection can be done in a similar way. In this case, the curve with a bigger AABB is subdivided.

## 4.5 Degree Elevation

o. a Bézier curve of degree $n$ may be represented as a Bézier curve of degree $(n+1)$.

Ex

$\mathbb{X}(t) = (1-t)^2 \mathbb{b}_0 + 2(1-t)t\, \mathbb{b}_1 + t^2 \mathbb{b}_2$ : a quadratic curve.

Multiplying by $1 = [(1-t)+t]$, we get

$$\mathbb{X}(t) = [(1-t)^3 + (1-t)^2 t]\,\mathbb{b}_0 + 2[(1-t)^2 t + (1-t)t^2]\,\mathbb{b}_1 + [(1-t)t^2 + t^3]\,\mathbb{b}_2$$

$$= (1-t)^3 \mathbb{b}_0 + 3(1-t)^2 t \left[\tfrac{1}{3}\mathbb{b}_0 + \tfrac{2}{3}\mathbb{b}_1\right] + 3(1-t)t^2 \left[\tfrac{2}{3}\mathbb{b}_1 + \tfrac{1}{3}\mathbb{b}_2\right] + t^3 \mathbb{b}_2 .$$

$$\therefore \mathbb{X}(t) = B_0^3(t)\,\mathbb{b}_0 + B_1^3(t)\left[\tfrac{1}{3}\mathbb{b}_0 + \tfrac{2}{3}\mathbb{b}_1\right] + B_2^3(t)\left[\tfrac{2}{3}\mathbb{b}_1 + \tfrac{1}{3}\mathbb{b}_2\right] + B_3^3(t)\,\mathbb{b}_2 .$$

Ex 4.2

$\mathbb{X}(t) = (1-t)^2 \mathbb{b}_0 + 2(1-t)t\, \mathbb{b}_1 + t^2 \mathbb{b}_2$,

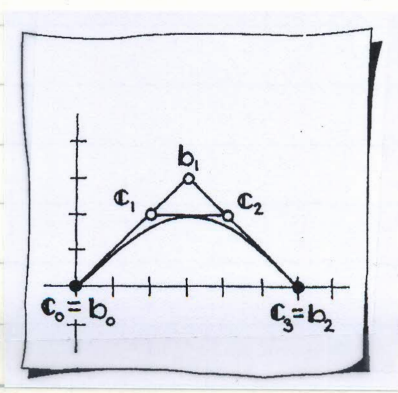
where $\mathbb{b}_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, $\mathbb{b}_1 = \begin{bmatrix} 3 \\ 3 \end{bmatrix}$, $\mathbb{b}_2 = \begin{bmatrix} 6 \\ 0 \end{bmatrix}$.

Degree elevation will produce

$\mathbb{X}(t) = (1-t)^3 \mathbb{c}_0 + 3(1-t)^2 t\, \mathbb{c}_1 + 3(1-t)t^2 \mathbb{c}_2 + t^3 \mathbb{c}_3$,

where $\mathbb{c}_0 = \mathbb{b}_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, $\mathbb{c}_3 = \mathbb{b}_2 = \begin{bmatrix} 6 \\ 0 \end{bmatrix}$

$\mathbb{c}_1 = \tfrac{1}{3}\mathbb{b}_0 + \tfrac{2}{3}\mathbb{b}_1 = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$, $\mathbb{c}_2 = \tfrac{2}{3}\mathbb{b}_1 + \tfrac{1}{3}\mathbb{b}_2 = \begin{bmatrix} 4 \\ 2 \end{bmatrix}$.




→ 그림 틀렸음!

Degree elevation of a Bézier curve of degree n to a Bézier curve of degree (n+1):

$$\begin{cases} \mathbb{c}_0 = \mathbb{b}_0 \\ \vdots \\ \mathbb{c}_i = \frac{i}{n+1}\mathbb{b}_{i-1} + \left(1 - \frac{i}{n+1}\right)\mathbb{b}_i \\ \vdots \\ \mathbb{c}_{n+1} = \mathbb{b}_n \end{cases}$$

$$\Rightarrow \begin{bmatrix} 1 & & & \\ * & * & & \\ & & \ddots & \\ & & * & * \\ & & & 1 \end{bmatrix} \begin{bmatrix} \mathbb{b}_0 \\ \vdots \\ \mathbb{b}_n \end{bmatrix} = \begin{bmatrix} \mathbb{c}_0 \\ \vdots \\ \mathbb{c}_{n+1} \end{bmatrix}$$

↳ $(n+2) \times (n+1)$ matrix

$$\underline{DB = \mathbb{C}}$$

∘ The process of degree elevation may be repeated. The resulting sequence of control polygons converges to the curve. However, convergence is too slow.

<u>Ex 4.3</u>

For $n = 2$,

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

# 4.6 Degree Reduction

- The inverse process of degree reduction is more important. Some CAD systems allow degrees up to 40, others use cubic curves. Reducing a curve of degree 40 to cubic is non-trivial. In practice, several cubic segments will be needed, involving an interplay between subdivision and degree reduction.
- Degree reduction approximates a curve of degree $(n+1)$ by a curve of degree $n$. To approximate the solution of $D\mathbb{B} = \mathbb{C}$, we solve $\underline{D^T D \mathbb{B} = D^T \mathbb{C}}$.
- The matrix $D^T D$ is independent of the given data, but dependent only on $n$. To solve many degree reduction problems, we store the LU factorization of $D^T D$.

Ex 4.4

For $n+1 = 3$, $D^T D = \frac{1}{9}\begin{bmatrix} 10 & 2 & 0 \\ 2 & 8 & 2 \\ 0 & 2 & 10 \end{bmatrix}$

For the degree-elevated cubic curve $\mathbb{X}(t)$ with $\mathbb{c}_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, $\mathbb{c}_1 = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$, $\mathbb{c}_2 = \begin{bmatrix} 4 \\ 2 \end{bmatrix}$, $\mathbb{c}_3 = \begin{bmatrix} 6 \\ 0 \end{bmatrix}$, we get

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

corresponds to the y-components

corresponds to the x-components

$$\Rightarrow B = (D^T D)^{-1}(D^T C) = \begin{bmatrix} 0 & 0 \\ 3 & 3 \\ 6 & 0 \end{bmatrix}$$

$\therefore \mathbb{b}_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, $\mathbb{b}_1 = \begin{bmatrix} 3 \\ 3 \end{bmatrix}$, $\mathbb{b}_2 = \begin{bmatrix} 6 \\ 0 \end{bmatrix}$.

- In general, the degree-reduced curve may not pass through the original curve endpoint $\mathbb{c}_0$ and $\mathbb{c}_{n+1}$.