MAT300 CURVES AND SURFACES

Julia Sánchez Sanz

DigiPen Institute of Technology Europe

julia.sanchez@digipen.edu

Spring 2020

Bezier curves

1. The Bezier curve in standard basis
2. Subdivision of curves

Apply change of basis Bernstein-Standard

$$\gamma(t) = \sum_{i=0}^{n} P_i B_i^n(t), \qquad t \in [0,1]. \tag{1}$$

We can obtain the standard representation of the Bezier curve by doing a change of basis Bernstein-Standard.

Let $T : \mathbb{R}^{n+1} \to \mathbb{R}^{n+1}$ with $T(\vec{x}) = M\vec{x}$ be the transformation for a change of basis Bernstein-Standard, then

$$\gamma(t) = (1, t, t^2, \ldots, t^n) M \begin{pmatrix} P_0 \\ P_1 \\ \vdots \\ P_n \end{pmatrix} \tag{2}$$

Example

Control points:

$$P_0 = (1, -1),\ P_1 = (2, 0),\ P_2 = (3, -1),\ P_3 = (2, -2) \text{ and } P_4 = (1, -2)$$

Bernstein polynomials:

$$B_0^4(t) = \binom{4}{0} (1-t)^4 t^0 = 1 - 4t + 6t^2 - 4t^3 + t^4 \Rightarrow (1, -4, 6, -4, 1)_S$$

$$B_1^4(t) = \binom{4}{1} (1-t)^3 t^1 = 4t - 12t^2 + 12t^3 - 4t^4 \Rightarrow (0, 4, -12, 12, -4)_S$$

$$B_2^4(t) = \binom{4}{2} (1-t)^2 t^2 = 6t^2 - 12t^3 + 6t^4 \Rightarrow (0, 0, 6, -12, 6)_S$$

$$B_3^4(t) = \binom{4}{3} (1-t)^1 t^3 = 4t^3 - 4t^4 \Rightarrow (0, 0, 0, 4, -4)_S$$

$$B_4^4(t) = \binom{4}{4} (1-t)^0 t^4 = t^4 \Rightarrow (0, 0, 0, 0, 1)_S$$

$$(1, t, t^2, t^3, t^4) \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ -4 & 4 & 0 & 0 & 0 \\ 6 & -12 & 6 & 0 & 0 \\ -4 & 12 & -12 & 4 & 0 \\ 1 & -4 & 6 & -4 & 1 \end{pmatrix} \begin{pmatrix} 1 & -1 \\ 2 & 0 \\ 3 & -1 \\ 2 & -2 \\ 1 & -2 \end{pmatrix} =$$

$$(1, t, t^2, t^3, t^4) \begin{pmatrix} 1 & -1 \\ 4 & 4 \\ 0 & -12 \\ -8 & 8 \\ 4 & -1 \end{pmatrix} =$$

$$(1 + 4t - 8t^3 + 4t^4, -1 + 4t - 12t^2 + 8t^3 - t^4), \quad t \in [0, 1]$$

Review of the De Casteljau algorithm

Given $P_0,\ P_1,\ \ldots,\ P_n$ points in $\mathbb{R}^2$ or $\mathbb{R}^3$ the Bezier curve

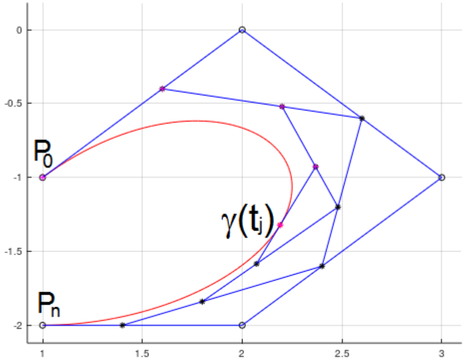
$\gamma(t) = \sum_{i=0}^{n} P_i B_i^n(t),\ t \in [0,1]$ can be computed as follows:

- Construct a mesh of $m+1$ nodes in $[0,1]$, for instance $t_j = \frac{j}{m}$ for $j = 0, 1, \ldots, m$.
- For each node apply recursively linear interpolation to obtain $\gamma(t_j)$

$$P_i^k(t_j) = t_j P_{i+1}^{k-1}(t_j) + (1-t_j) P_i^{k-1}(t_j) \tag{3}$$

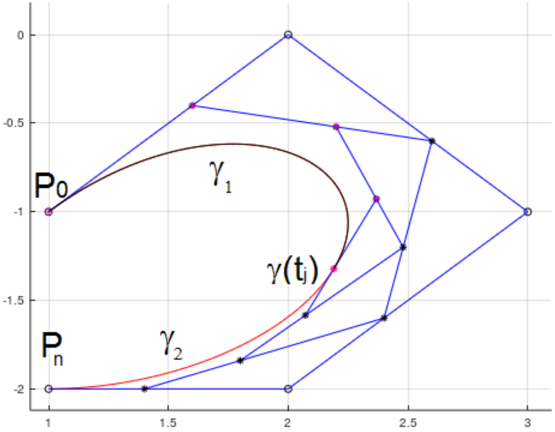
  for $k = 1, \ldots, n$ and $i = 0, \ldots, n-k$.
- Store the values of γ in an array for displaying the plot.

We now revise what is going on in the recursion process.



We can split the Bezier curve into two curves:

- Curve γ_1 starting at P_0 and ending at $\gamma(t_j)$
- Curve γ_2 starting at $\gamma(t_j)$ and ending at P_n



How can we express γ_1 as a Bezier curve?

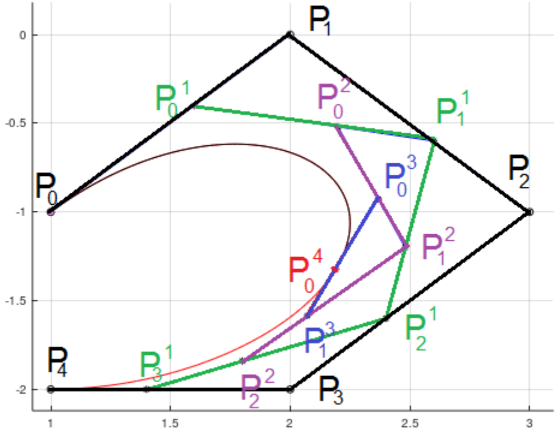
How can we express γ_2 as a Bezier curve?

Construction through recursion

When evaluating $\gamma(t_j)$ through recursion we did

$$
\begin{array}{cccccc}
P_0 & & & & & \\
 & P_0^1 & & & & \\
P_1 & & P_0^2 & & & \\
 & P_1^1 & & P_0^3 & & \\
P_2 & & P_1^2 & & & \\
 & P_2^1 & & \vdots & \ddots & \\
P_3 & \vdots & \vdots & \vdots & \dots & P_0^n = \gamma \\
\vdots & \vdots & \vdots & P_{n-3}^3 & & \\
P_{n-1} & & P_{n-2}^2 & & & \\
 & P_{n-1}^1 & & & & \\
P_n & & & & &
\end{array}
$$

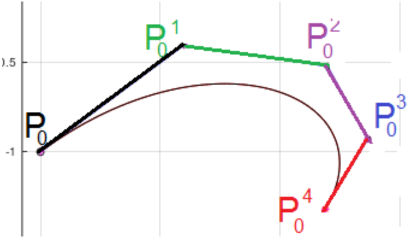
Lets have a look to the intermediate control points



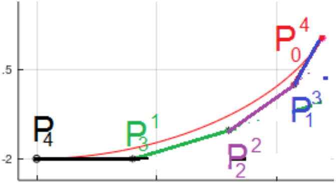
We see that γ_1 starts at P_0 and ends at P_0^4

We see that γ_2 starts at P_0^4 and ends at P_4

Moreover, γ_1 is convex and lies in the hull of P_0, P_0^1, P_0^2, P_0^3 and P_0^4



Similarly, γ_2 is convex and lies in the hull of P_0^4, P_1^3, P_2^2, P_3^1 and P_4



Both curves are generated by linear interpolation using the intermediate control points,

$$
\begin{array}{cccccc}
P_0 & & & & & \\
 & P_0^1 & & & & \\
P_1 & & P_0^2 & & & \\
 & P_1^1 & & P_0^3 & & \\
P_2 & & P_1^2 & & & \\
 & P_2^1 & & \vdots & \ddots & \\
P_3 & \vdots & \vdots & \vdots & \dots & P_0^n \\
\vdots & \vdots & \vdots & P_{n-3}^3 & & \\
P_{n-1} & & P_{n-2}^2 & & & \\
 & P_{n-1}^1 & & & & \\
P_n & & & & &
\end{array}
$$

so taking the above intermediate control points

$$\gamma_1(t) = \sum_{i=0}^{n} P_0^i(t_j) B_i^n(t), \quad t \in [0,1] \tag{4}$$

$$
\begin{array}{cccccc}
P_0 & & & & & \\
 & P_0^1 & & & & \\
P_1 & & P_0^2 & & & \\
 & P_1^1 & & P_0^3 & & \\
P_2 & & P_1^2 & & & \\
 & P_2^1 & & \vdots & \ddots & \\
P_3 & \vdots & \vdots & \vdots & \dots & \color{blue}{P_0^n} \\
\vdots & \vdots & \vdots & \color{blue}{P_{n-3}^3} & & \\
P_{n-1} & & \color{blue}{P_{n-2}^2} & & & \\
 & \color{blue}{P_{n-1}^1} & & & & \\
\color{blue}{P_n} & & & & &
\end{array}
$$

and taking the below intermediate control points

$$\gamma_2(t) = \sum_{i=0}^{n} P_i^{n-i}(t_j) B_i^n(t), \quad t \in [0,1] \tag{5}$$

By applying subdivision we obtain two Bezier curves. By applying subdivision to the resuliting curves 4 curves, then 8, 16, and so on.

Midpoint subdivision

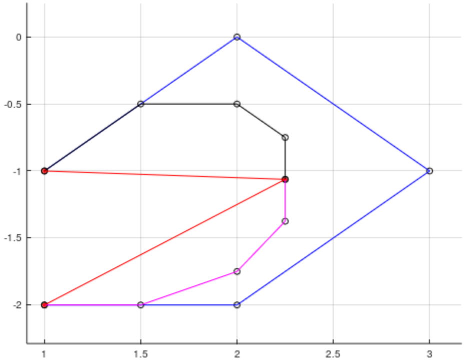
If we take the subdivision at $t = \frac{1}{2}$ for the previous example we obtain

$$
\begin{array}{lllll}
P_0 = (1,-1) & & & & \\
& P_0^1 = (\frac{3}{2}, -\frac{1}{2}) & & & \\
P_1 = (2,0) & & P_0^2 = (2, -\frac{1}{2}) & & \\
& P_1^1 = (\frac{5}{2}, -\frac{1}{2}) & & P_0^3 = (\frac{9}{4}, -\frac{3}{4}) & \\
P_2 = (3,-1) & & P_1^2 = (\frac{5}{2}, -1) & & P_0^4 = (\frac{9}{4}, -\frac{17}{16}) \\
& P_2^1 = (\frac{5}{2}, -\frac{3}{2}) & & P_1^3 = (\frac{9}{4}, -\frac{11}{8}) & \\
P_3 = (2,-2) & & P_2^2 = (2, -\frac{7}{4}) & & \\
& P_3^1 = (\frac{3}{2}, -2) & & & \\
P_4 = (1,-2) & & & &
\end{array}
$$

The points P_0, P_0^4 and P_4 belong to the curve γ.

By taking the above and below intermediate control points we can generate new Bezier curves.

Midpoint subdivision 1st iteration



The red polygonal curve passing through P_0, P_0^4 and P_4 is the approximation to the Bezier curve. The black and magenta lines are the control points for γ_1 and γ_2

Midpoint subdivision 2nd iteration

We apply now the midpoint subdivision for the two curves γ_1 and γ_2.

For γ_1

$$P_0 = (1, -1)$$
$$P_1 = (\tfrac{3}{2}, -\tfrac{1}{2})$$
$$P_2 = (2, -\tfrac{1}{2})$$
$$P_3 = (\tfrac{9}{4}, -\tfrac{3}{4})$$
$$P_4 = (\tfrac{9}{4}, -\tfrac{17}{16})$$

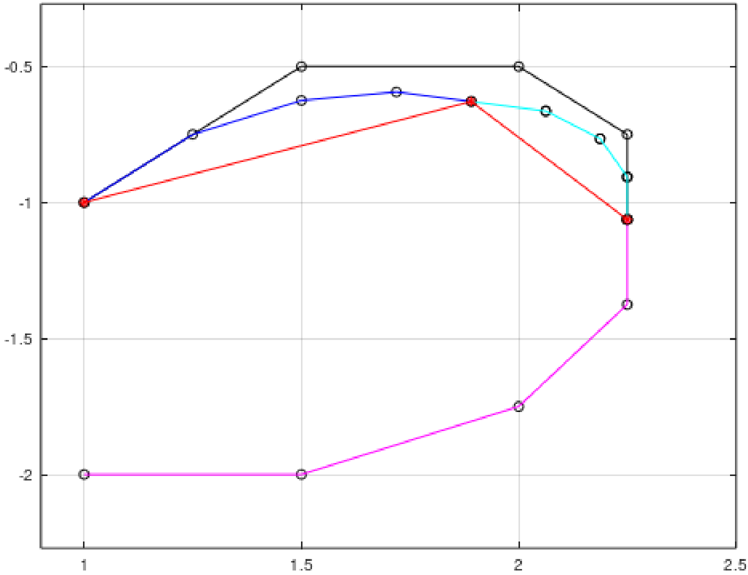
$$P_0^1 = (\tfrac{5}{4}, -\tfrac{3}{4})$$
$$P_1^1 = (\tfrac{7}{4}, -\tfrac{1}{2})$$
$$P_2^1 = (\tfrac{17}{8}, -\tfrac{5}{8})$$
$$P_3^1 = (\tfrac{9}{4}, -\tfrac{29}{32})$$

$$P_0^2 = (\tfrac{3}{2}, -\tfrac{5}{8})$$
$$P_1^2 = (\tfrac{31}{16}, -\tfrac{9}{16})$$
$$P_2^2 = (\tfrac{35}{16}, -\tfrac{49}{64})$$

$$P_0^3 = (\tfrac{55}{32}, -\tfrac{19}{32})$$
$$P_1^3 = (\tfrac{33}{16}, -\tfrac{85}{128})$$

$$P_0^4$$

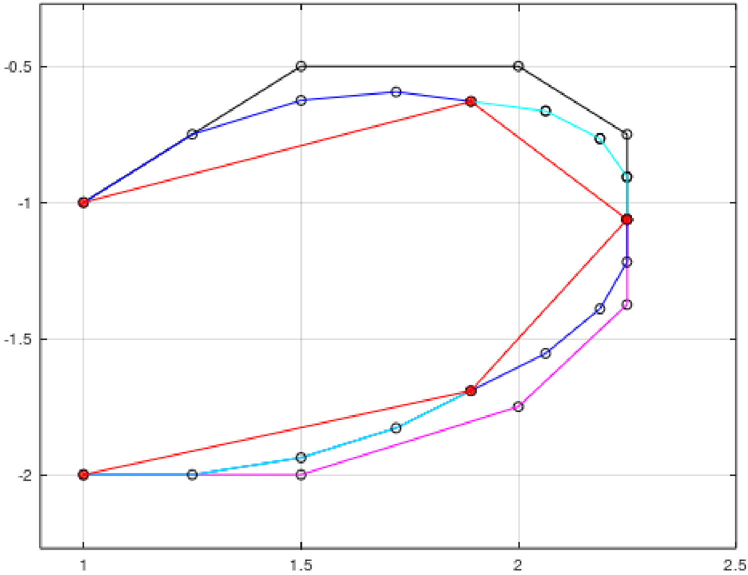
with $P_0^4 = (\frac{121}{64}, -\frac{161}{256})$


-0.5
-1
-1.5
-2
1
1.5
2
2.5

For γ_2

$$
\begin{array}{lllll}
P_0 = (\frac{9}{4}, -\frac{17}{16}) & & & & \\
 & P_0^1 = (\frac{9}{4}, -\frac{39}{32}) & & & \\
P_1 = (\frac{9}{4}, -\frac{11}{8}) & & P_0^2 = (\frac{35}{16}, -\frac{89}{64}) & & \\
 & P_1^1 = (\frac{17}{8}, -\frac{25}{16}) & & P_0^3 = (\frac{33}{16}, -\frac{199}{128}) & \\
P_2 = (2, -\frac{7}{4}) & & P_1^2 = (\frac{31}{16}, -\frac{55}{32}) & & P_0^4 \\
 & P_2^1 = (\frac{7}{4}, -\frac{15}{8}) & & P_1^3 = (\frac{55}{32}, -\frac{117}{64}) & \\
P_3 = (\frac{3}{2}, -2) & & P_2^2 = (\frac{3}{2}, -\frac{31}{16}) & & \\
 & P_3^1 = (\frac{5}{4}, -2) & & & \\
P_4 = (1, -2) & & & &
\end{array}
$$

with $P_0^4 = (\frac{121}{64}, -\frac{433}{256})$


-0.5
-1
-1.5
-2
1
1.5
2
2.5

In the second iteration we approximated our Bezier curve by 5 points:

- the initial and end points of the curve,
- the midpoint of the first iteration,
- the midpoints of the curves in the second iteration.

For the third iteration we select the above and below control points for each piece, therefore we obtain four new Bezier curves, and so on.

This process for obtaining the approximation of a Bezier curve is called midpoint subdivision.

- 1st iteration: 1 curve, 3 points
- 2nd iteration: 2 curves, 5 points
- 3rd iteration: 4 curves, 9 points
- 4th iteration: 8 curves, 17 points
- 5th iteration: 16 curves, 33 points
- $\vdots$
- kth iteration: 2^{k-1} curves, $2 + \sum_{i=0}^{k-1} 2^i = 2^k + 1$ points.

Midpoint subdivision algorithm

Given $P_0,\ P_1,\ \ldots,\ P_n$ points in $\mathbb{R}^2$ or $\mathbb{R}^3$ the Bezier curve

$\gamma(t) = \sum_{i=0}^{n} P_i B_i^n(t),\ t \in [0,1]$ can be computed as follows:

- Store P_0 and P_n.
- For j=1:k do:
  - For m=1:2^{j-1} do:
    - apply the midpoint subdivision
    - store midpoint
- Plot the resulting curve

Remark: one of the key points is how to store the midpoints to plot the curve in the correct order!