NURB

2025 年 5 月 18 日

1 Bernstein Polynomial

Bernstein basis polynomial of degree n are

$$B_{i,n}(t) = \binom{n}{i} t^i (1-t)^{n-i} \qquad , (t \in [0,1] , \quad i = 1, \dots, n)$$

Bernstein polynomial of degree n is

$$B_n(t) = \sum_{i=0}^{n} \beta_i B_{i,n}(t)$$

where β_i is called Bernstein or Bezier coefficient.

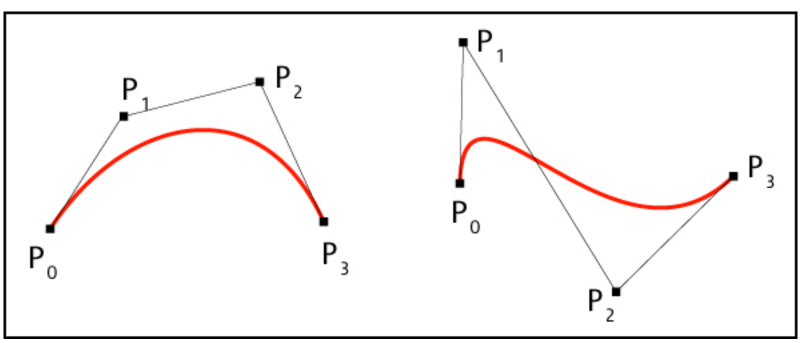
Bernstein polynomial can approximate any continuous function on $[0, 1]$.

$$B_n(f)(t) = \sum_{i=0}^{n} f\left(\frac{i}{n}\right) B_{i,n}(t)$$

and

$$\lim_{n\to\infty} B_n(f)(t) \xrightarrow{\text{uniformly}} f(t)$$

2 Bezier Curve



Given $n+1$ control points, $\boldsymbol{P}_1, \dots, \boldsymbol{P}_n$, a bezier curve is defined by

$$C(t) = \sum_{i=0}^{n} \boldsymbol{P}_i B_{i,n}(t)$$

$C(t)$ always passes through the first and the last control point and its tangent at $C(t_0)$ and $C(t_n)$ are $\overrightarrow{P_0 P_1}$ and $\overrightarrow{P_{n-1} P_n}$ respectively.

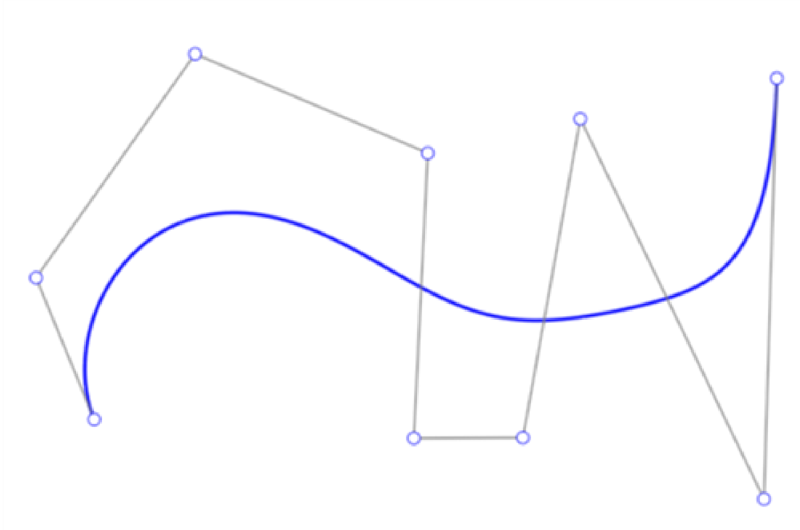


图 **1:** Bezier curve with more control points

3 B-Spline

B-Spline is a generalization of Bezier curve. Given a knot vector

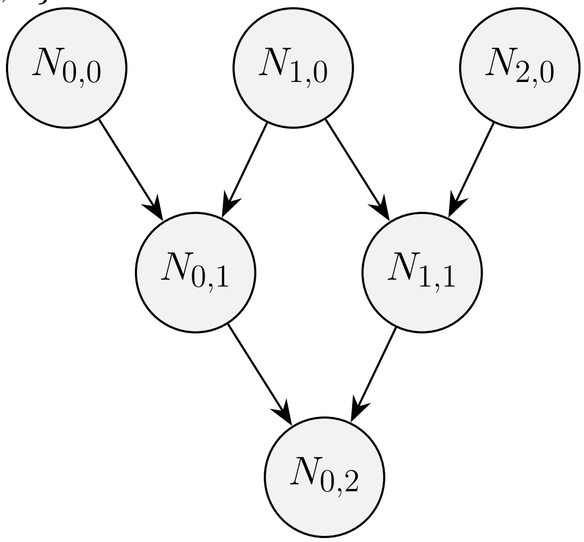
$$T = \{t_0, \ldots, t_m \quad | \quad t_i \in [0,1], t_i \leq t_{i+1}\}$$

We can build B-Spline basis $N_{i,j}$ by bottom up recursion defined by

$$N_{i,0}(t) = \begin{cases} I_{[t_i, t_{i+1}]}(t) & \text{, if } t_i \neq t_{i+1} \\ 0 & \text{, otherwise} \end{cases}$$

$$N_{i,j}(t) = \frac{t - t_i}{t_{i+j} - t_i} N_{i,j-1}(t) + \frac{t_{i+j+1} - t}{t_{i+j+1} - t_{i+1}} N_{i+1,j-1}(t)$$

For example, if $T = \{0, 1, 2, 3\}$, we can build the basis from bottom up recursively



We ended up with

$$
\begin{aligned}
N_{0,1} &= tI_{[0,1]} + (2-t)I_{[1,2]} \\
N_{1,1} &= (t-1)I_{[1,2]} + (3-t)I_{[2,3]} \\
N_{0,2} &= \frac{t^2}{2}I_{[0,1]} + \frac{(6t-2t^2-3)}{2}I_{[1,2]} + \frac{(3-t)^2}{2}I_{[2,3]}
\end{aligned}
$$

With control points $C_0, \ldots, C_n$, we can define the degree of B-spline $D = m - n - 1$. And the B-spline is defined by the basis as

$$C(t) = \sum_{i=0}^{D} C_i N_{i,D}(t)$$

Often times, control points are not given, instead we need to fit a B-spline to points $(P_0, t_0), \ldots, (P$
Instead we solve for C_i,

$$P_i = C(t_i) = \sum_{j=0}^{D} C_j N_{j,D}(t_i) \quad \Rightarrow \quad P = CN$$