三、Bezier曲线的生成

生成一条Bezier曲线实际上就是要求出曲线上的点。下面介绍两种曲线生成的方法:

1、根据定义直接生成Bezier曲线

绘制Bezier曲线主要有以下步骤:

$$p(t)=\sum_{i=0}^{n} P_i B_{i,n}(t) \qquad t\in[0,1]$$

$$B_{i,n}(t) = \frac{n!}{i!(n-i)!} t^i (1-t)^{n-i} = C_n^i t^i (1-t)^{n-i} \qquad (i = 0, 1, \ldots, n)$$

① 首先给出 C_n^i 的递归计算式：

$$C_n^i = \frac{n!}{i!\left(n-i\right)!} = \frac{n-i+1}{i} C_n^{i-1} \qquad n \geq i$$

② 将 $p(t) = \sum_{i=0}^{n} P_i B_{i,n}(t)$ 表示成分量坐标形式：

$$x(t) = \sum_{i=0}^{n} x_i B_{i,n}(t)$$

$$y(t) = \sum_{i=0}^{n} y_i B_{i,n}(t) \qquad t \in [0,1]$$

$$z(t) = \sum_{i=0}^{n} z_i B_{i,n}(t)$$

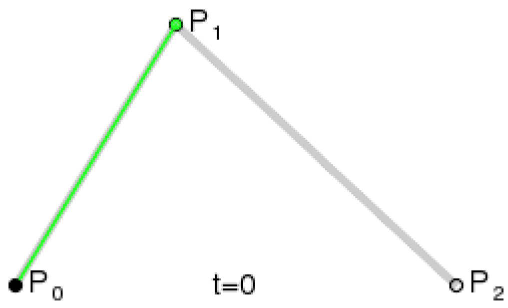
根据以上的公式可以直接写出绘制Bezier曲线的程序

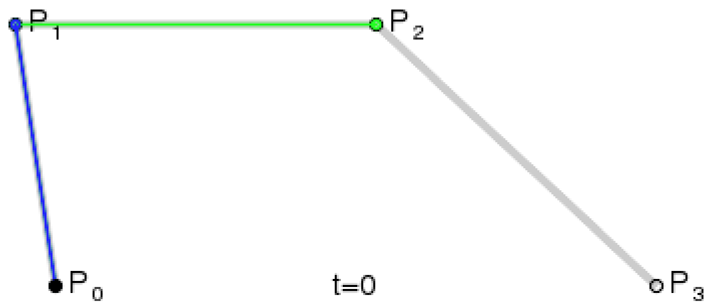
$$C_n^i = \frac{n!}{i!(n-i)!} = \frac{n-i+1}{i} C_n^{i-1} \qquad \mathrm{n} \geq \mathrm{i}$$

2、Bezier曲线的递推(de Casteljau)算法

根据Bezier曲线的定义确定的参数方程绘制Bezier曲线,因其计算量过大,不太适合在工程上使用

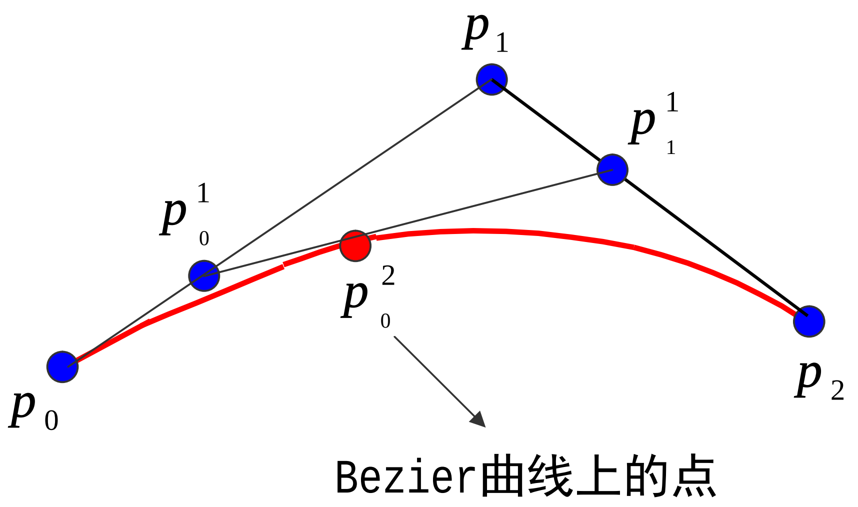
de Casteljau提出的递推算法则要简单得多



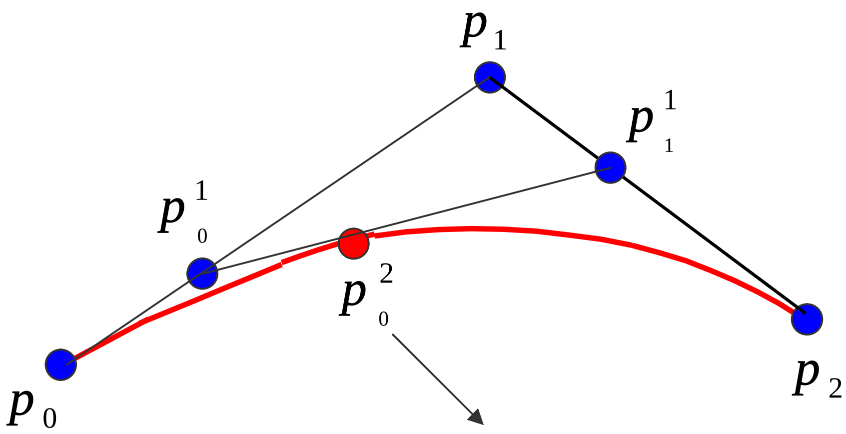


Bezier曲线上的任一个点(t)，都是其它相邻线段的同等比例(t)点处的连线，再取同等比例(t)的点再连线，一直取到最后那条线段的同等比例(t)处，该点就是Beizer曲线上的点(t)

以二次Bezier曲线为例，求曲线上t=1/3的点：



$$P_0^1 = (1-t)P_0 + tP_1$$
$$P_1^1 = (1-t)P_1 + tP_2$$
$$P_0^2 = (1-t)P_0^1 + tP_1^1$$



t从0变到1，第一、二式就分别表示控制二边形的第一、二条边，它们是两条一次Bezier曲线。将一、二式代入第三式得：

$$P_0^2 = (1-t)^2 P_0 + 2t(1-t)P_1 + t^2 P_2$$

$$P_0^2 = (1-t)^2 P_0 + 2t(1-t)P_1 + t^2 P_2$$

当t从0变到1时，它表示了由三顶点P_0、P_1、P_2三点定义的一条二次Bezier曲线

二次Bezier曲线P_0^2可以定义为分别由前两个顶点(P_0,P_1)和后两个顶点(P_1,P_2)决定的一次Bezier曲线的线性组合

由(n+1)个控制点P_i(i=0,1,...,n)定义的n次Bezier曲线P_0^n可被定义为分别由前、后n个控制点定义的两条(n-1)次Bezier曲线P_0^{n-1}与P_1^{n-1}的线性组合:

$$P_0^n = (1-t)P_0^{n-1} + tP_1^{n-1} \qquad t \in [0,1]$$

由此得到Bezier曲线的递推计算公式:

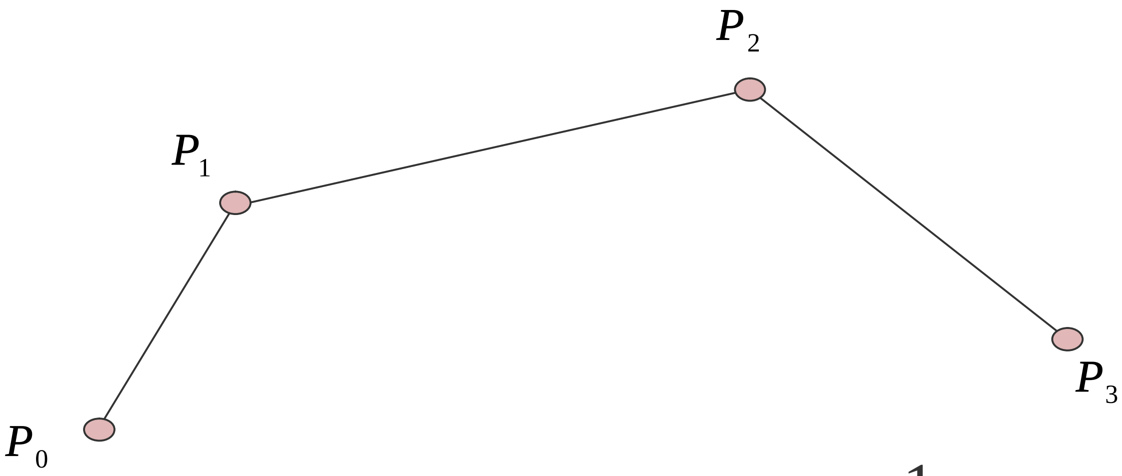
$$P_i^{\,k} = \begin{cases} P_i & k = 0 \\ (1-t)P_i^{\,k-1} + tP_{i+1}^{\,k-1} & k = 1,2,\ldots,n, i = 0,1,\ldots,n-k \end{cases}$$

这便是著名的de Casteljau算法。用这一递推公式，在给定参数下，求Bezier曲线上一点P(t)非常有效

de Casteljau算法稳定可靠，直观简便，可以编出十分简捷的程序，是计算Bezier曲线的基本算法和标准算法。

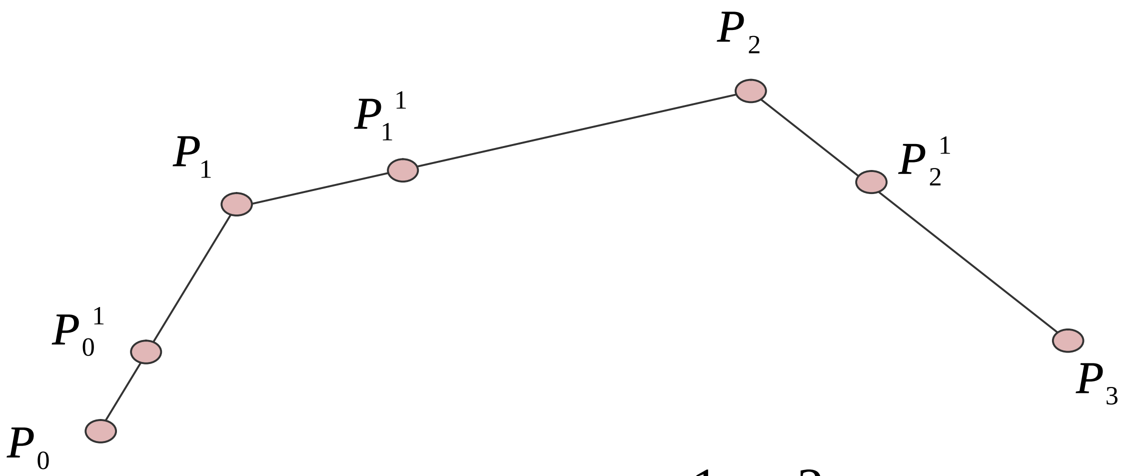
3、de Casteljau算法几何作图

这一算法可用简单的几何作图来实现



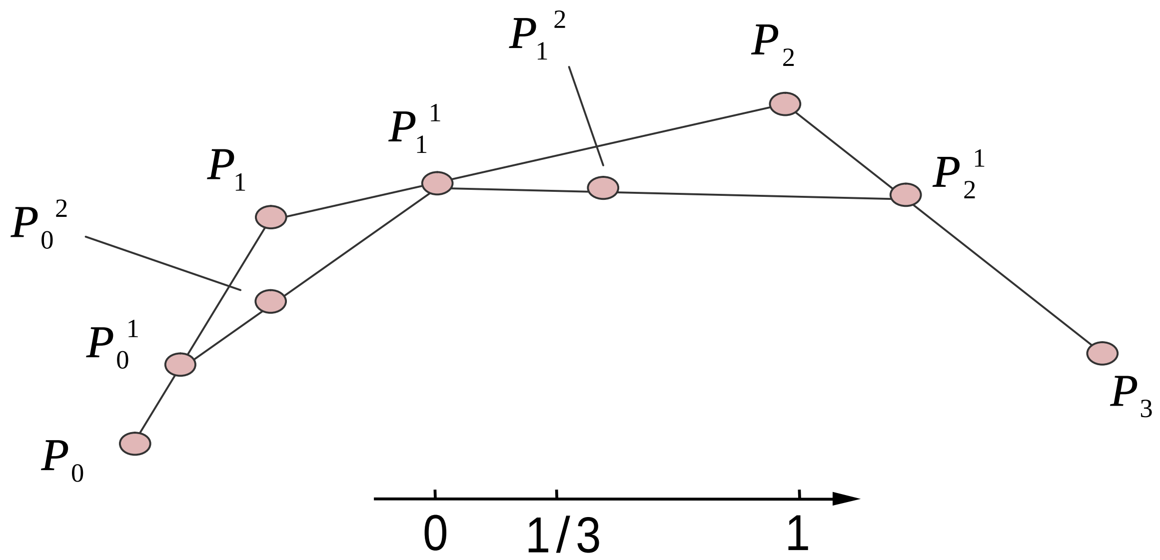
$t \in [0,1]$ 假设 $t = \dfrac{1}{3}$

（1）依次对原始控制多边形每一边执行相同的定比分割，所得分点就是由第一级递推生成的中间顶点 $P_i^1 (i=0,1,\cdots,n-1)$

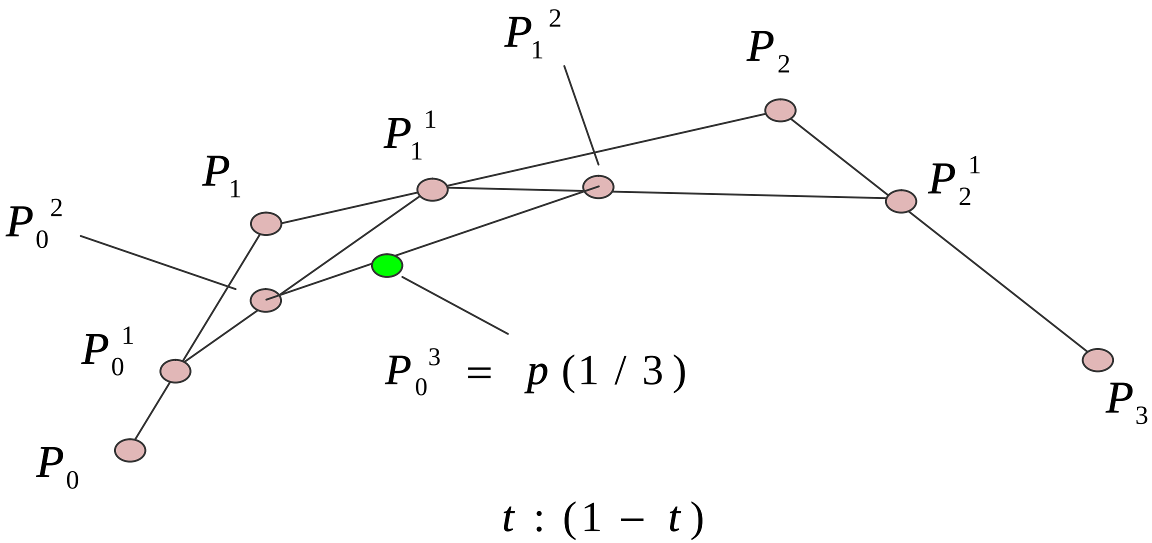


$$t:(1-t) \qquad \frac{1}{3}:(\frac{2}{3})$$

（2）对这些中间顶点构成的控制多边形再执行同样的定比分割，得第二级中间顶点：$P_i^2(i=0,1,\mathbf{L},n-2)$

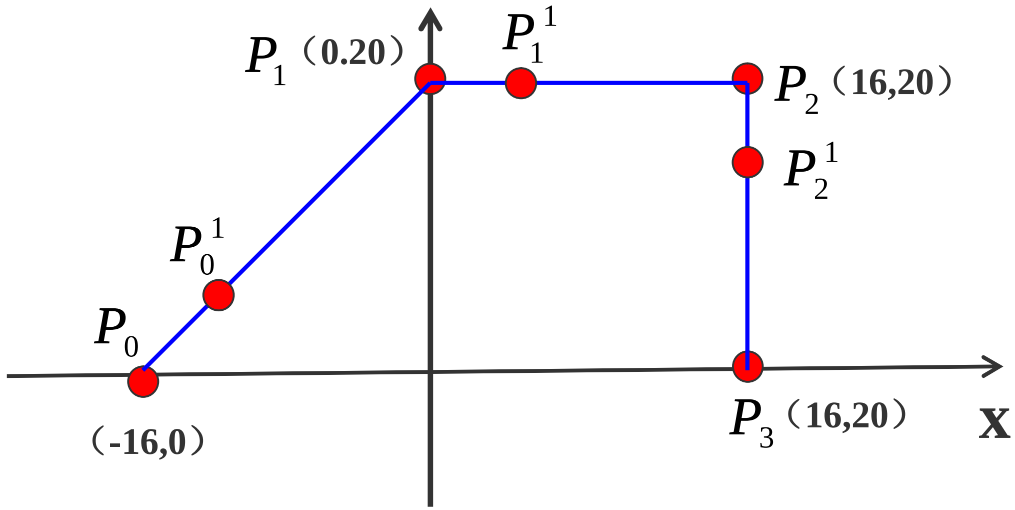


（3）重复进行下去，直到n级递推得到一个中间顶点 $P(t)$，即为所求曲线上的点



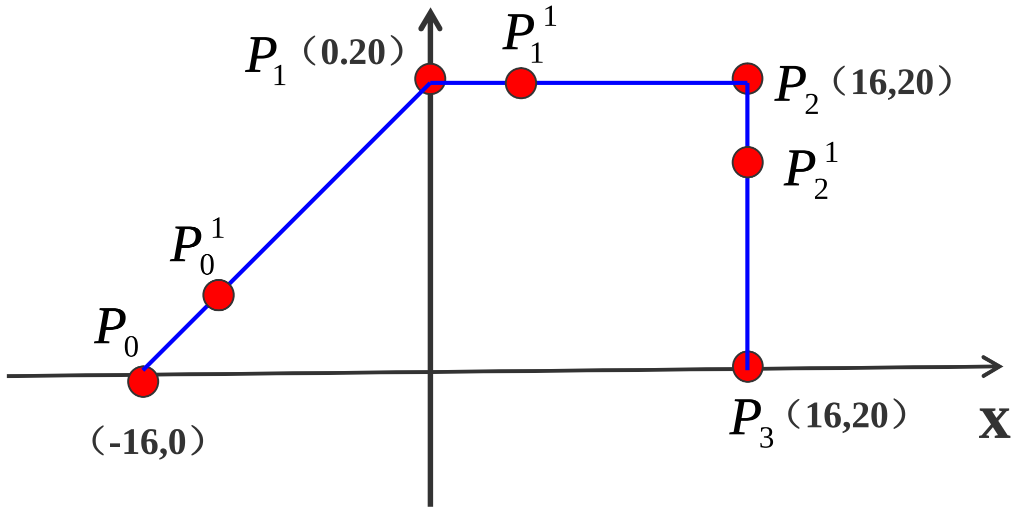
Bezier曲线计算举例

计算参数为 t=1/4 的P值



$$P_i^k = \begin{cases} P_i & k = 0 \\ (1-t)P_i^{k-1} + tP_{i+1}^{k-1} & k = 1,2,\ldots, n, i = 0,1,\ldots, n-k \end{cases}$$

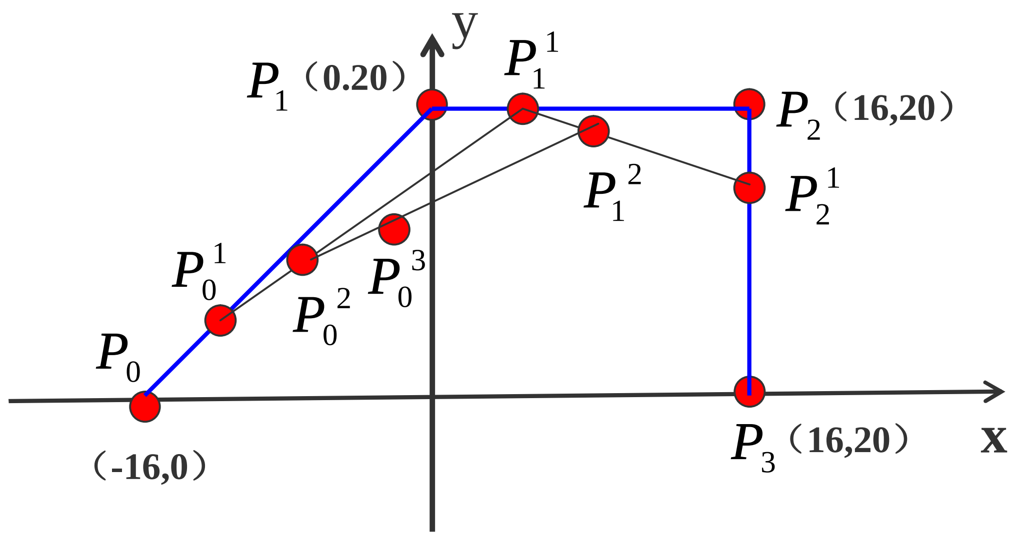
$$P_0^1 = (1-t)P_0 + tP_1 = (1-\frac{1}{4})p_0 + \frac{1}{4}p_1 = \frac{3}{4}[-16,0] + \frac{1}{4}[0,20] = [-12,5]$$



$$P_0^1 = (1-t)P_0 + tP_1 = (1-\frac{1}{4})p_0 + \frac{1}{4}p_1 = \frac{3}{4}[-16,0] + \frac{1}{4}[0,20] = [-12,5]$$

$$P_1^1 = (1-t)P_1 + tP_2 = (1-\frac{1}{4})p_1 + \frac{1}{4}p_2 = \frac{3}{4}[0,20] + \frac{1}{4}[16,20] = [4,20]$$

$$P_2^1 = (1-t)P_2 + tP_3 = (1-\frac{1}{4})p_2 + \frac{1}{4}p_3 = \frac{3}{4}[16,20] + \frac{1}{4}[16,0] = [16,15]$$



$$P_0^2 = (1-t)P_0^1 + tP_1^1 = (1-\frac{1}{4})P_0^1 + \frac{1}{4}P_1^1 = \frac{3}{4}[-12,5] + \frac{1}{4}[4,20] = [-8,8.75]$$

$$P_1^2 = (1-t)P_1^1 + tP_2^1 = (1-\frac{1}{4})P_0^1 + \frac{1}{4}P_1^1 = \frac{3}{4}[-12,5] + \frac{1}{4}[4,20] = [-8,8.75]$$

$$P_0^3 = (1-t)P_0^2 + tP_1^2 = (1-\frac{1}{4})P_0^2 + \frac{1}{4}P_1^2 = \frac{3}{4}[-8,8.75] + \frac{1}{4}[7,18.75]$$

$$P_0^1 = (1-t)P_0 + tP_1 = (1-\frac{1}{4})p_0 + \frac{1}{4}p_1 = \frac{3}{4}[-16,0] + \frac{1}{4}[0,20] = [-12,5]$$

$$P_1^1 = (1-t)P_1 + tP_2 = (1-\frac{1}{4})p_1 + \frac{1}{4}p_2 = \frac{3}{4}[0,20] + \frac{1}{4}[16,20] = [4,20]$$

$$P_2^1 = (1-t)P_2 + tP_3 = (1-\frac{1}{4})p_2 + \frac{1}{4}p_3 = \frac{3}{4}[16,20] + \frac{1}{4}[16,0] = [16,15]$$

$$P_0^2 = (1-t)P_0^1 + tP_1^1 = (1-\frac{1}{4})P_0^1 + \frac{1}{4}P_1^1 = \frac{3}{4}[-12,5] + \frac{1}{4}[4,20] = [-8,8.75]$$

$$P_0^3 = (1-t)P_0^2 + tP_1^2 = (1-\frac{1}{4})P_0^2 + \frac{1}{4}P_1^2 = \frac{3}{4}[-8,8.75] + \frac{1}{4}[7,18.75]$$

$$= [-4.25, 11.25] = P(1/4)$$