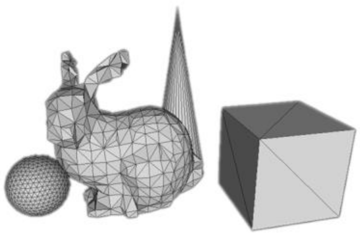
Introduction to Computer Graphics



Geometry
几何

第八章 几何

- 几何的表示形式
  - 隐式表示
  - 显式表示
- 参数化曲线曲面
  - 贝塞尔曲线
  - 贝塞尔曲面
- 点云表示
- 符号距离场表示
- 网格表示
- 网格处理

几何的表示方式

- 隐式几何表示（Implicit Representation）
  - 几何的空间位置不直接给出，仅给出其满足的关系
  - $X^2+Y^2+Z^2 = 1$

- 显式几何表示（ Explicit Representation ）
  - 直接给出几何的空间位置
  - X=(1,2,3)， Y=(2,3,4)， Z=(1, 1, 1)

参数表示

- 曲线曲面的参数表示是指将其上各点的坐标变量显式地表示成参数的函数形式。若取参数为 t，则曲线的参数表示为

$$\boldsymbol{P}(t)=(x(t),y(t),z(t)) \qquad t\in[0,1]$$

其中 $x(t)$, $y(t)$和 $z(t)$分别为 t的显式函数，

即每一个t对应空间一个点 $(x(t),y(t),z(t))$

- 通常将参数区间规范化为[0, 1]。参数方程中的参数可以代表多种不同的量，如时间、角度等。
- 连接 $\boldsymbol{P}_0(x_0,y_0)$和 $\boldsymbol{P}_1(x_1,y_1)$两点的直线段的参数方程可写为

$$\boldsymbol{P}(t)=\boldsymbol{P}_0+(\boldsymbol{P}_1-\boldsymbol{P}_0)t \Leftrightarrow \begin{cases} x=x_0+(x_1-x_0)t, \\ y=y_0+(y_1-y_0)t, \end{cases} \qquad t\in[0,1]$$

参数表示的优越性

- 参数方程的形式不依赖于坐标系的选取，具有形状不变性
- 对参数表示的曲线、曲面进行平移、放缩和旋转等几何变换比较容易
- 用参数表示的曲线曲面的交互能力强，参数表示式中系数的几何意义明确，并提高了自由度，便于控制形状

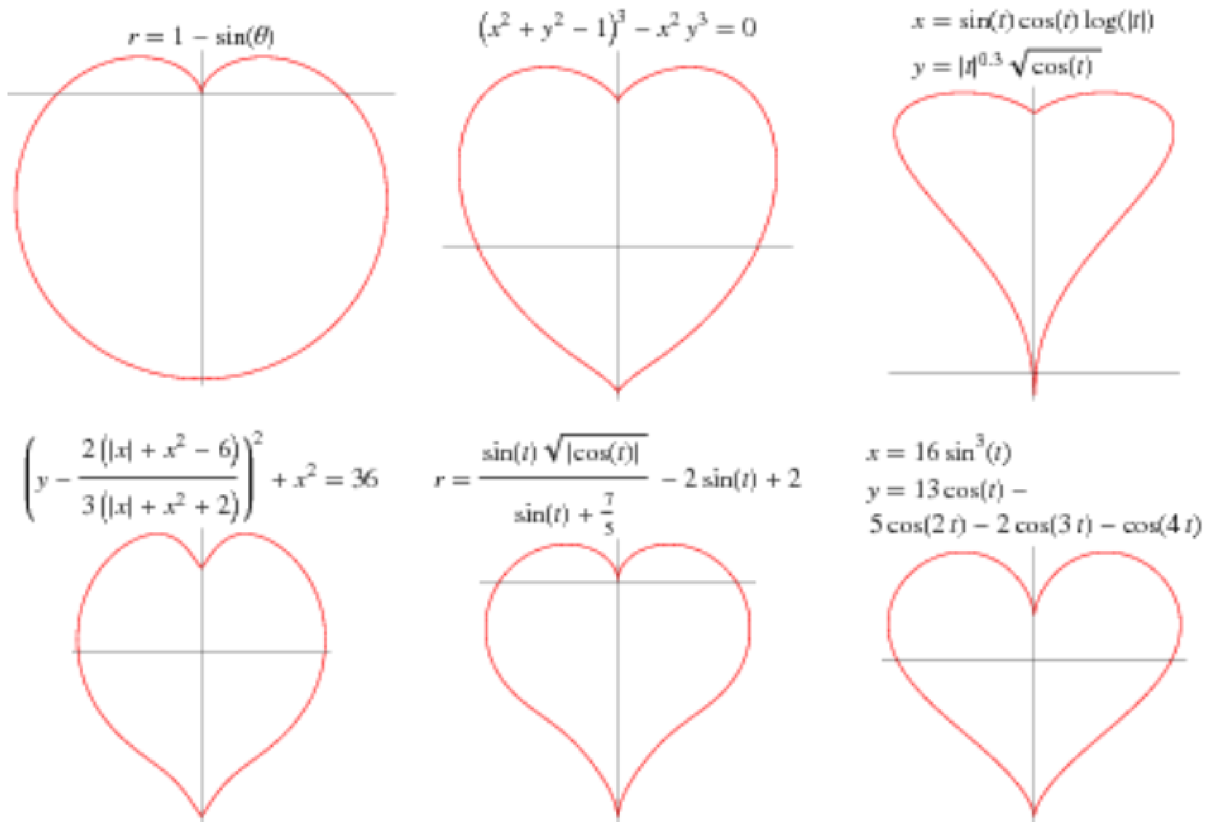
参数化曲线曲面

- 字体设计、Logo设计等

- 怎么用函数表达?

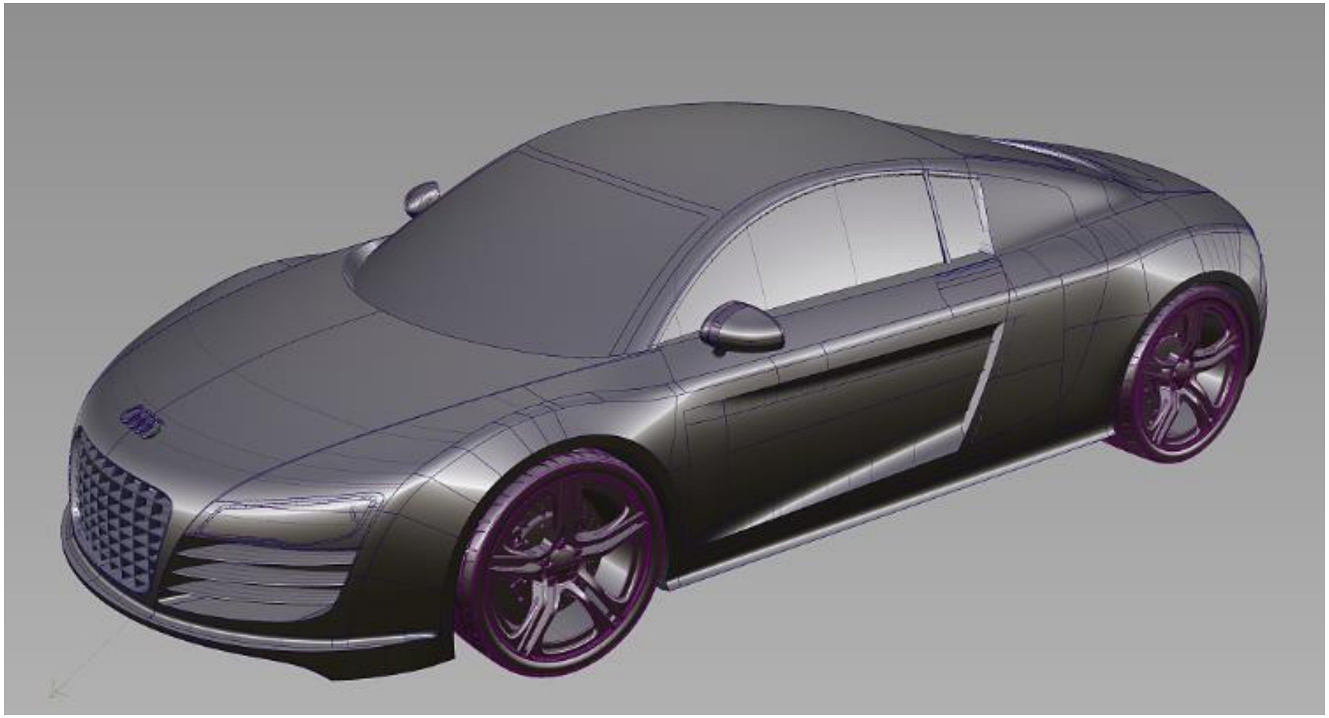
参数化曲线曲面

- 人工设计函数式->困难
  - 哪些是隐式，哪些是显式？



参数化曲线曲面

- 先点出控制点，再基于控制点生成曲线？
  - 贝塞尔曲线、B样条曲线、NURBS曲线等



贝塞尔曲线

- 贝塞尔曲线（Bézier Curve）是一段n次多项式曲线，是构造自由曲线曲面的重要和基本方法之一。
- 它具有许多优点，诸如保凸性，凸包性，曲线形状不依赖于坐标系的选择，人机交互手段灵活等。由于它能满足几何造型对曲线曲面的要求，因此在理论和应用上均得到了极大的重视和发展。

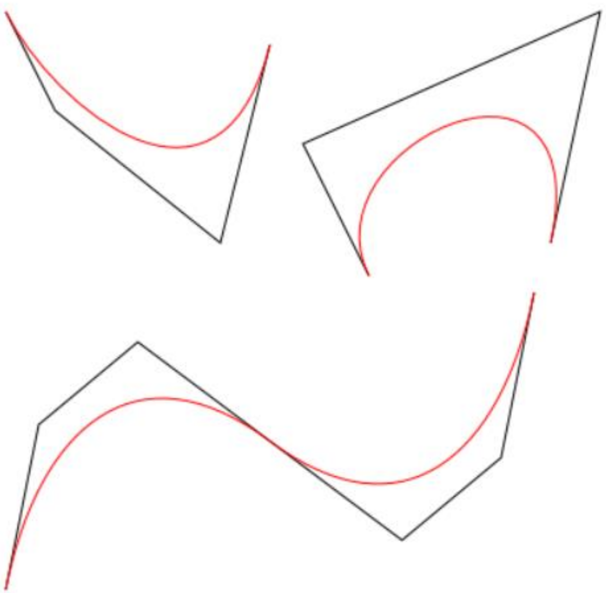
贝塞尔曲线

- 贝塞尔曲线于1962，由法国工程师皮埃尔·贝塞尔（Pierre Bézier）所广泛发表，他运用贝塞尔曲线来为汽车的主体进行设计
- 贝塞尔曲线最初由Paul de Casteljau于1959年运用de Casteljau演算法开发，以稳定数值的方法求出贝塞尔曲线。

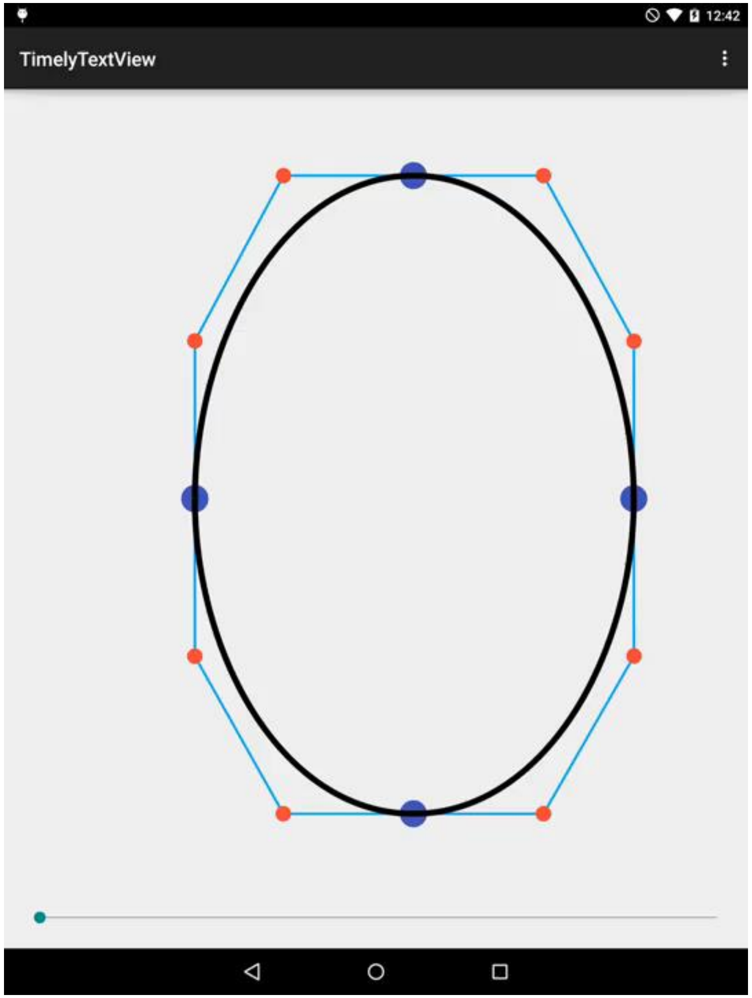

Paul de Casteljau

贝塞尔曲线

- 先点出控制点，再基于控制点生成曲线？控制点对曲线的影响？
  - 控制点控制曲线的基本形状
  - 要求曲线通过第0个和最后一个控制点
  - 曲线起始点和终止点处的切线与对应控制点切线一致



贝塞尔曲线



线性贝塞尔曲线

- 给定点P0、P1两个控制点，线性贝塞尔曲线只是一条两点之间的直线。这条线由下式给出：

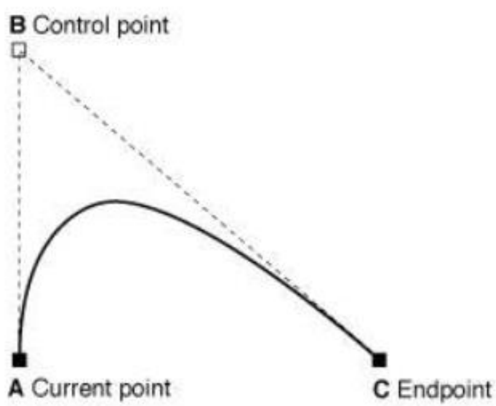
$$\mathbf{B}(t) = \mathbf{P}_0 + (\mathbf{P}_1 - \mathbf{P}_0)t = (1-t)\mathbf{P}_0 + t\mathbf{P}_1 \ , \ t \in [0,1]$$

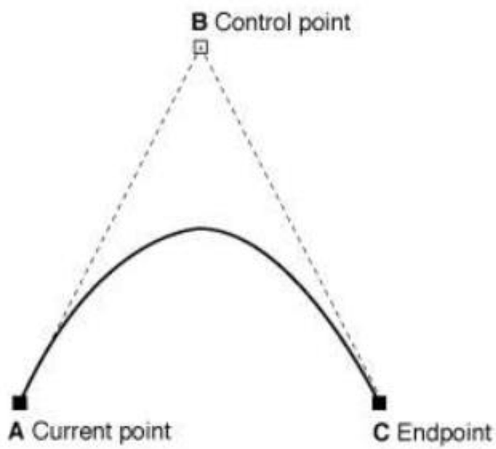
- 等同于线性插值

二次方贝塞尔曲线

- 二次方贝塞尔曲线的路径由给定点**P0**、**P1**、**P2**三个控制点的函数**B**（t）得到：

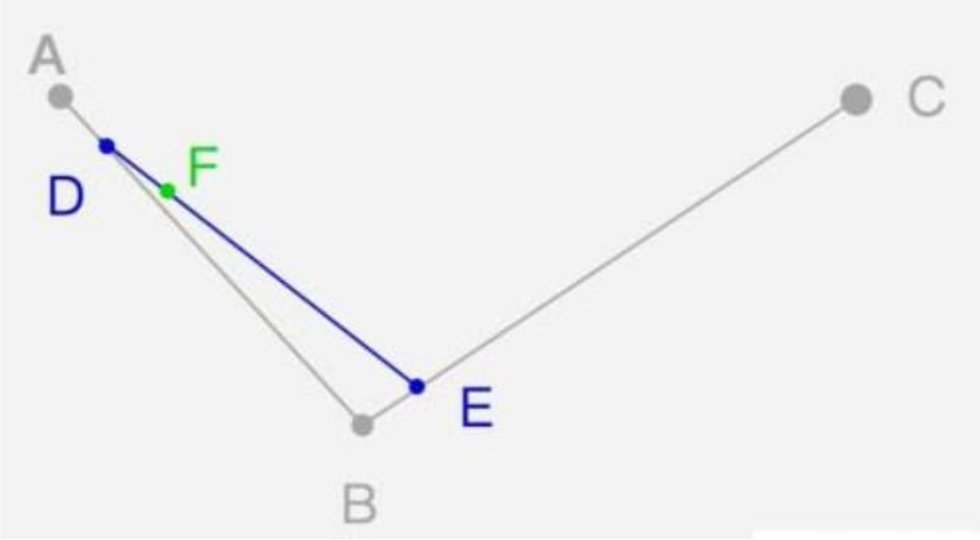
$$\mathbf{B}(t) = (1-t)^2\mathbf{P}_0 + 2t(1-t)\mathbf{P}_1 + t^2\mathbf{P}_2 \ , t \in [0,1]$$





二次方贝塞尔曲线

- 二次方贝塞尔曲线与线性贝塞尔曲线的关系：
- 如下图：
  - D=(1-t)A+tB, E=(1-t)B+tC
  - F=(1-t)D+tE

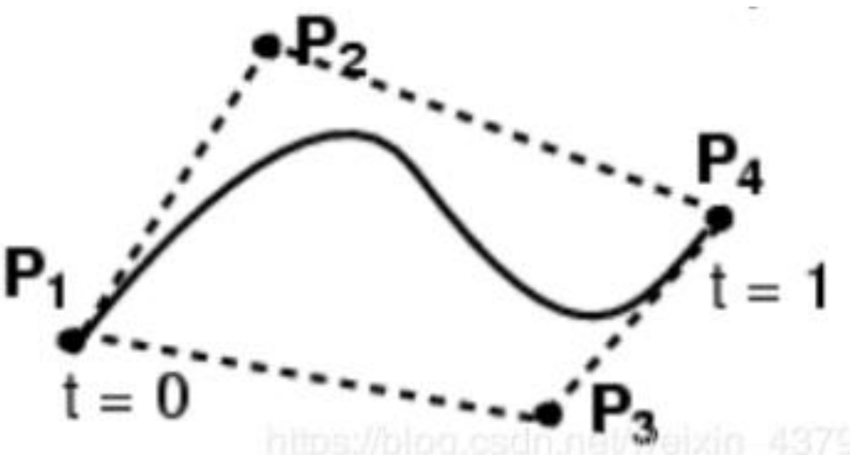


二次方贝塞尔曲线

- 二次方贝塞尔曲线与线性贝塞尔曲线的关系：
- 可以把二次方贝塞尔曲线看成是线性贝塞尔曲线的嵌套
- 可以看成是控制点之间的线性组合
- 观察组合系数有什么规律？
- 组合系数之和为1！

三次方贝塞尔曲线

- 三次方贝塞尔曲线有几个控制点呢？
  - 4个
- 如何定义？
  - 类似二次方贝塞尔曲线，嵌套定义



三次方贝塞尔曲线

- 给定四个点
- 首先通过线性插值，得到三个新的控制点
- 再对三个新控制点，进行线性插值，得到两个控制点
- 对两个控制点，进行线性插值，得到曲线上的点

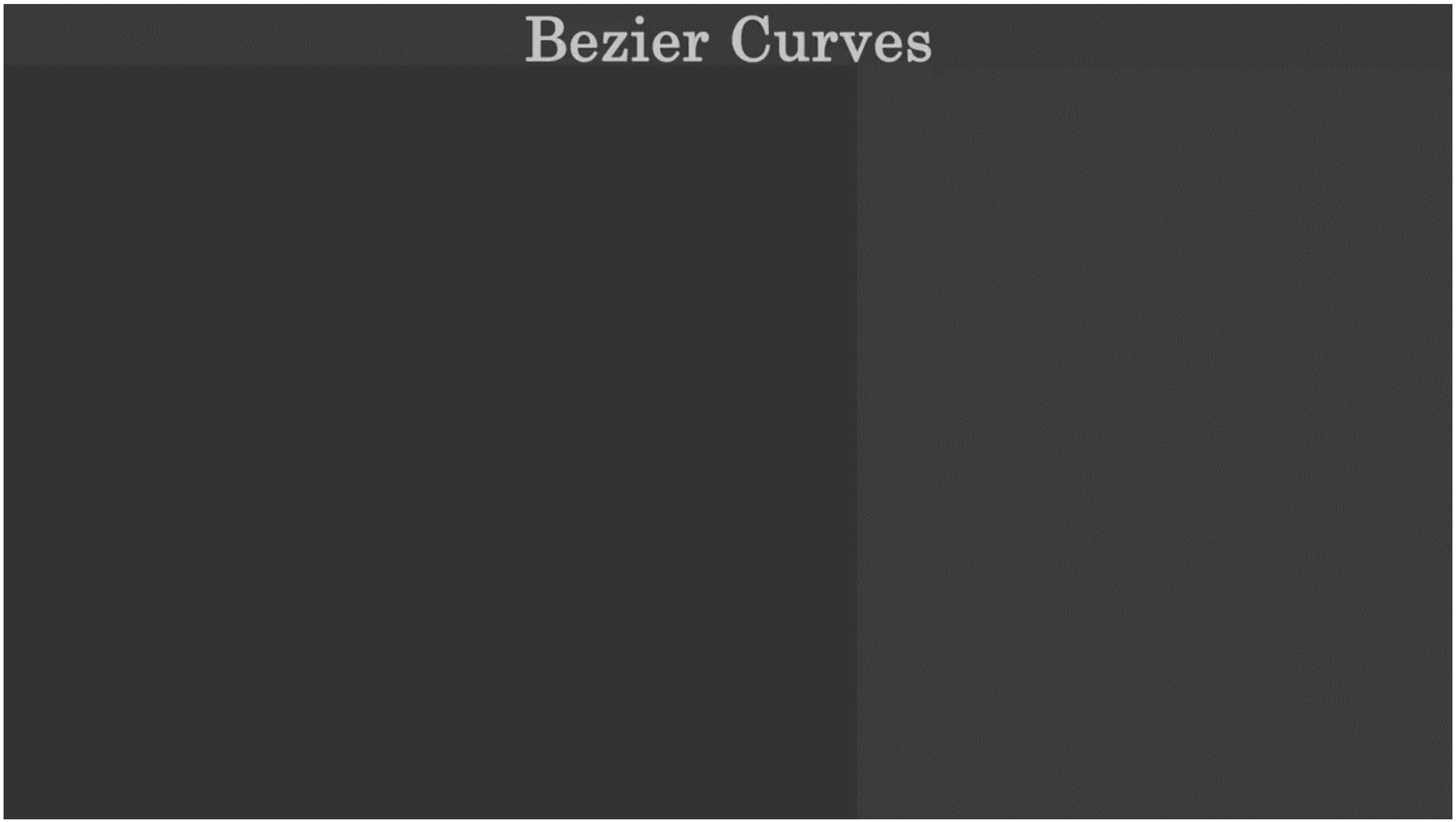
- 进行三次线性插值

三次方贝塞尔曲线

- **P**0、**P**1、**P**2、**P**3四个控制点在平面或在三维空间中定义了三次方贝塞尔曲线。曲线起始于**P**0走向**P**1，并从**P**2的方向来到**P**3。一般不会经过**P**1或**P**2；这两个点只是在那里提供方向信息。**P**0和**P**1之间的间距，决定了曲线在转而趋进**P**3之前，走向**P**2方向的“长度有多长”。

$$\mathrm{B}(t)=\mathrm{P}_0(1-t)^3+3\mathrm{P}_1t(1-t)^2+3\mathrm{P}_2t^2(1-t)+\mathrm{P}_3t^3\ ,\ t\in[0,1]$$

三次方贝塞尔曲线

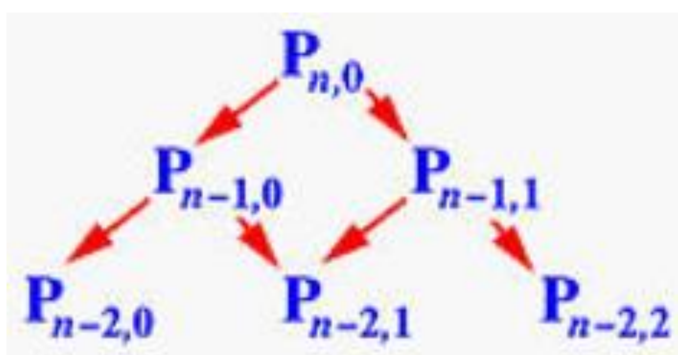


de Casteljau算法

- 对于n+1个控制点，可生成n次贝塞尔曲线
- 采用递归算法

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

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

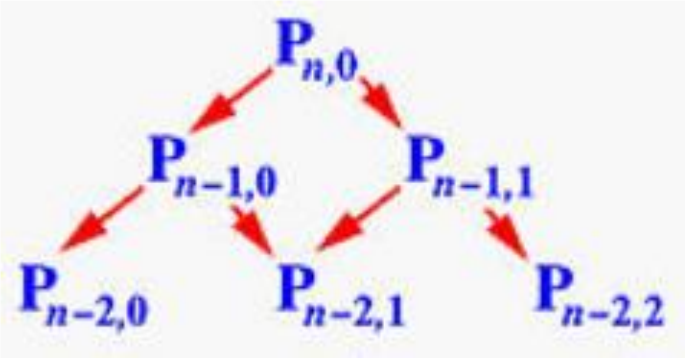


de Casteljau算法

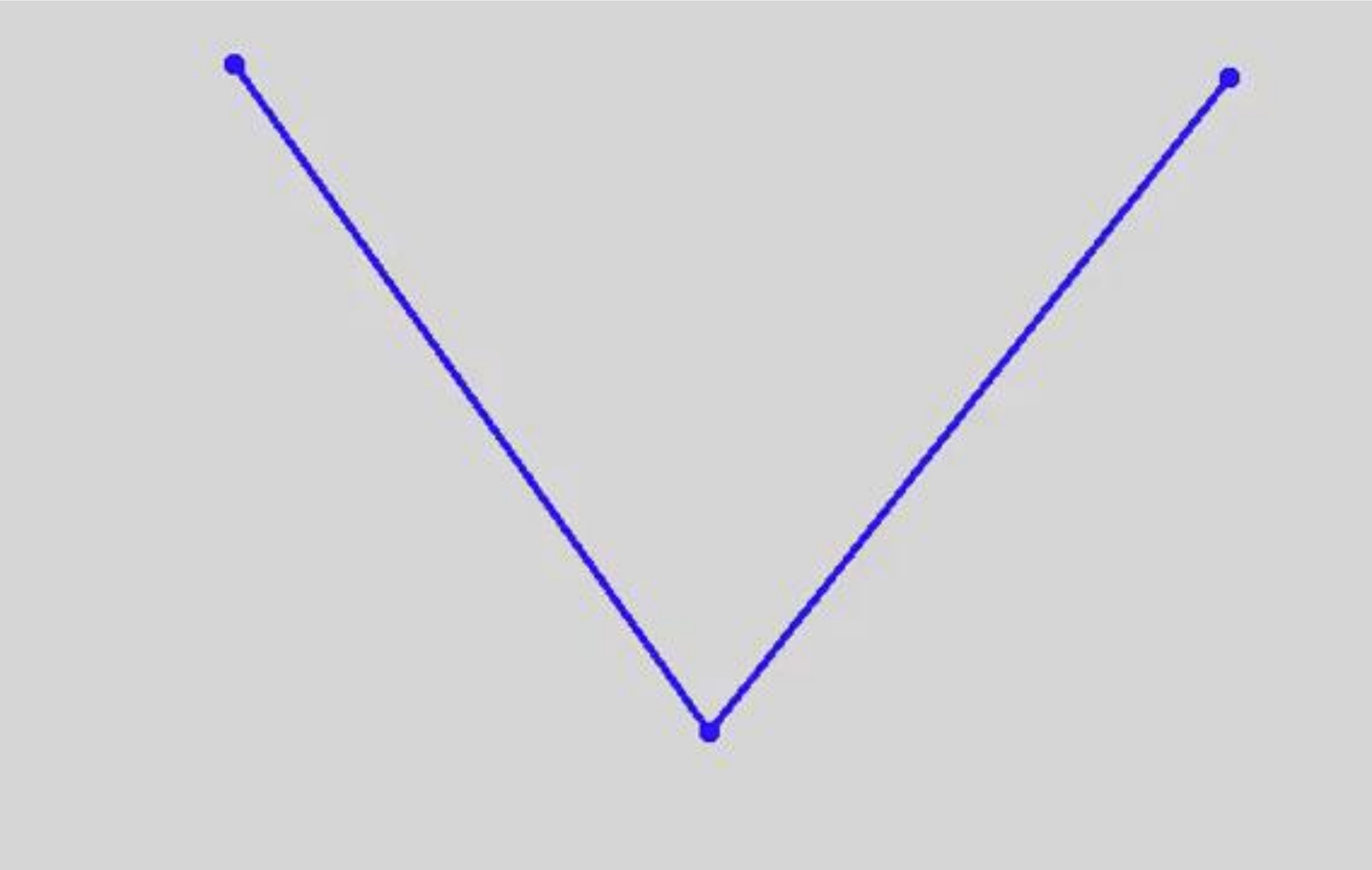
- 对于n+1个控制点，可生成n次贝塞尔曲线
- 采用递归算法

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

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



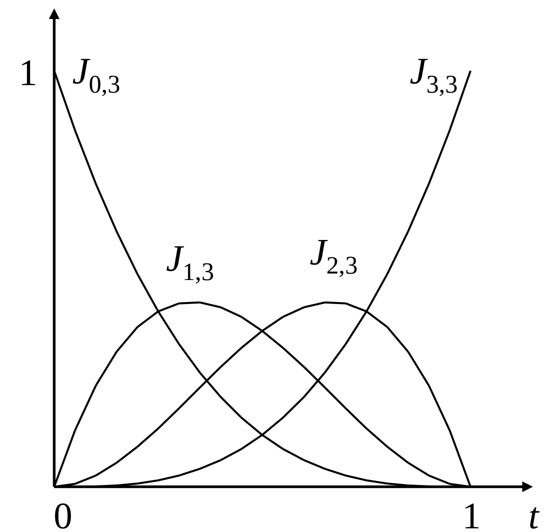
de Casteljau算法



贝塞尔曲线的数学定义

- 在空间给定$n+1$个点 $\boldsymbol{P}_0,\boldsymbol{P}_1,\cdots,\boldsymbol{P}_n$ ，称下列参数多项式曲线为n次Bézier曲线

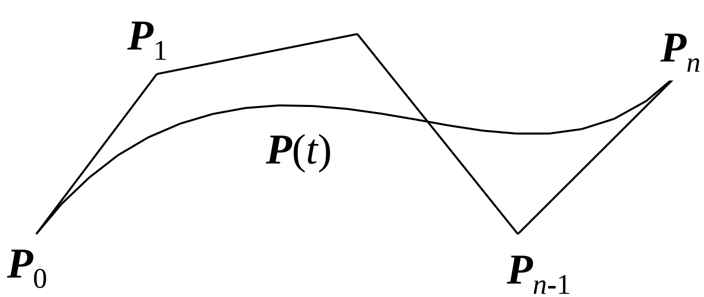
$$\boldsymbol{P}(t)=\sum_{i=0}^{n}\boldsymbol{P}_i J_{i,n}(t) \qquad 0\le t\le 1$$



四个3次Bernstein基函数

其中 $J_{i,n}(t)$ 是Bernstein基函数：

$$J_{i,n}(t)=\frac{n!}{i!(n-i)!}t^i(1-t)^{n-i} \qquad i=0,1,\cdots,n$$



n次Bézier曲线及控制多边形

贝塞尔曲线的性质

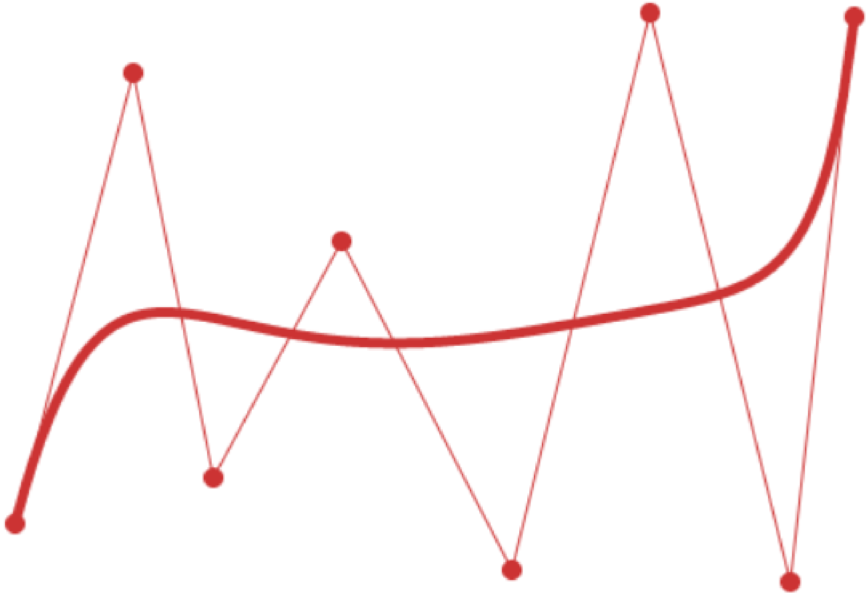
- 经过起始点和终点（未必经过中间控制点）
- 在起点和终点处，其切线方向与首段控制线和末段控制线方向一致
- 仿射变换不变性
  - 何为仿射变换?
  - 平移、旋转、缩放、错切
- 凸包性
  - 何为凸包

贝塞尔曲线的性质

- 经过起始点和终点（未必经过中间控制点）
- 在起点和终点处，其切线方向与首段控制线和末段控制线方向一致
- 仿射变换不变性
  - 何为仿射变换?
  - 平移、旋转、缩放、错切
- 凸包性
  - 何为凸包

贝塞尔曲线的拼接

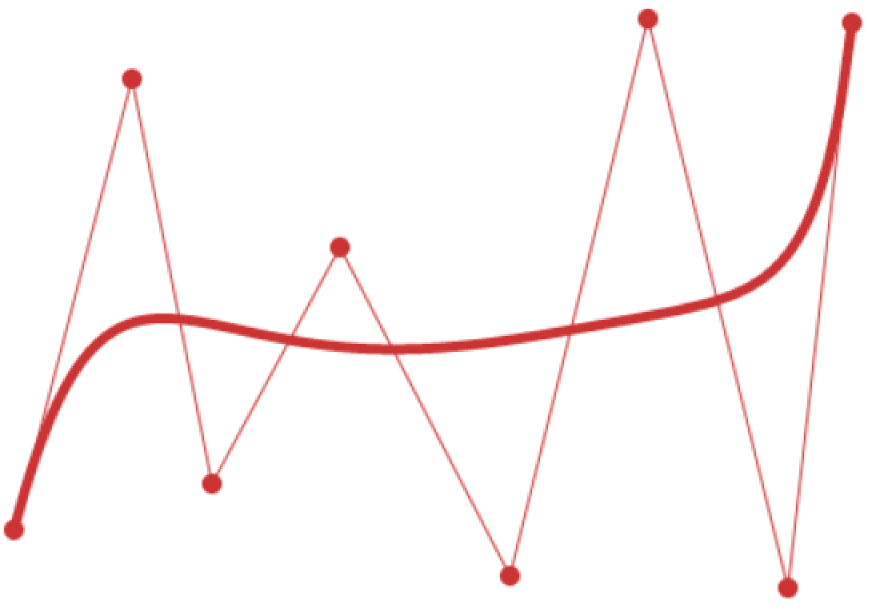
- 控制点过多，自由度差
- 无法反映控制点真实走向



http://bezierdemo.appspot.com/

贝塞尔曲线的拼接

- 分解成多个三次贝塞尔曲线，进行拼接
- 曲线的连续性
  - C0,C1,C2连续



贝塞尔曲面

- 思考 曲线和曲面的关系？
- 点动成线，线动成面？
- 先生成贝塞尔控制线，再根据控制线上面的点生成面

贝塞尔曲面

- 给定3*3的控制点Pi,j (i=0,1,2, j =0,1,2)
- 根据de Casteljau算法，对同一行的点生成贝塞尔曲线
- 再根据de Casteljau算法，对生成的贝塞尔曲线上的点，再生成贝塞尔曲线。

贝塞尔曲面

- 曲面生成公式

  B(u,v) = B(v)(B(u)(P00,P01,P02) , B(u)(P10,P11,P12) , B(u)(P20,P21,P22))

贝塞尔曲面