第8讲：几何

上次课程内容

- 纹理的应用
  - 环境贴图
  - 凹凸贴图
  - 位移贴图
- 几何概述
  - 几何无处不在
  - 几何的多种表示形式
    - ✓ 隐式
    - ✓ 显式

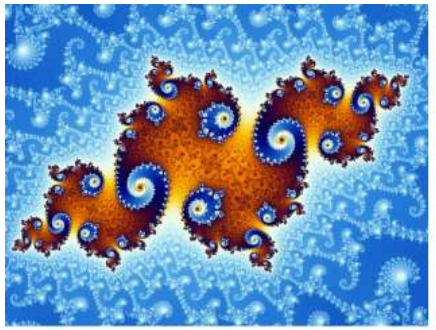
本次课程内容

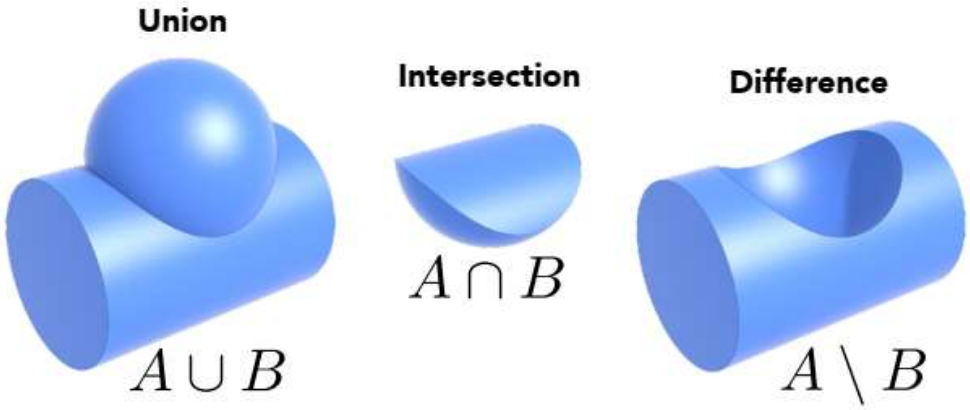
- 几何的多种表示形式
  - 隐式
  - 显式
- 曲线（Curves）
- 曲面(Surfaces)
- 学习体验调查

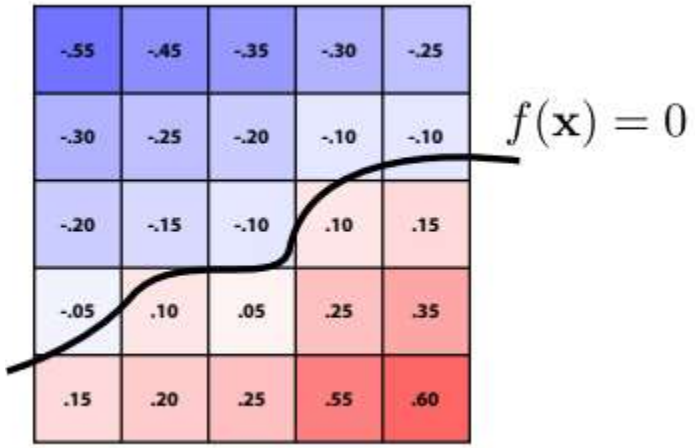
几何的表示形式

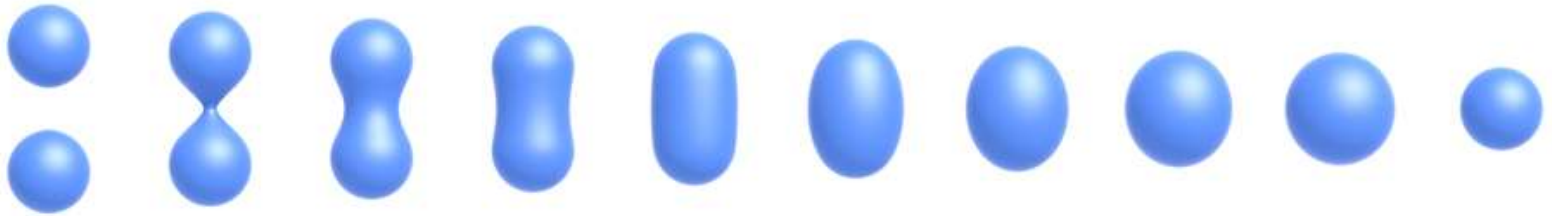
- 隐式（Implicit）
  - 代数曲面
  - 构造实体几何
  - 距离函数
  - 水平集
  - 分形
  - ……

$$(R-\sqrt{x^2+y^2})^2+z^2=r^2$$









中国海洋大学 OCEAN UNIVERSITY OF CHINA

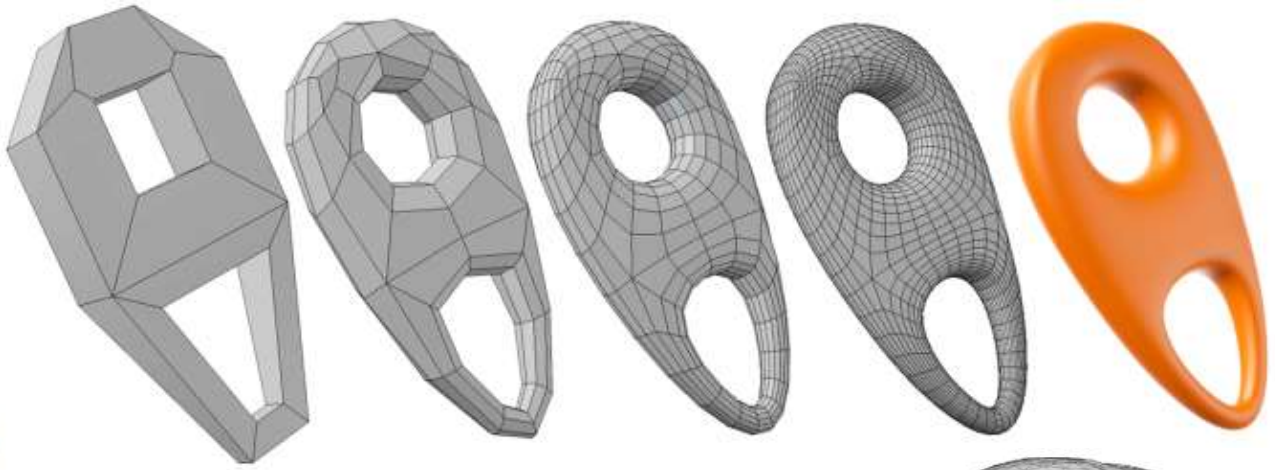
几何的隐式表示

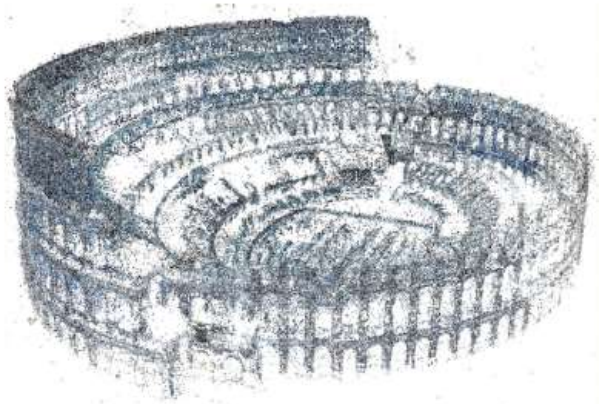
- 优点
  - 简洁的描述（例如：利用函数）
  - 某些查询很容易（例如：点在不在内部，点到表面的距离）
  - 适合做光线与表面的求交
  - 对于简单形状，描述准确，不存在采样误差
  - 易于处理拓扑变化（例如：流体）

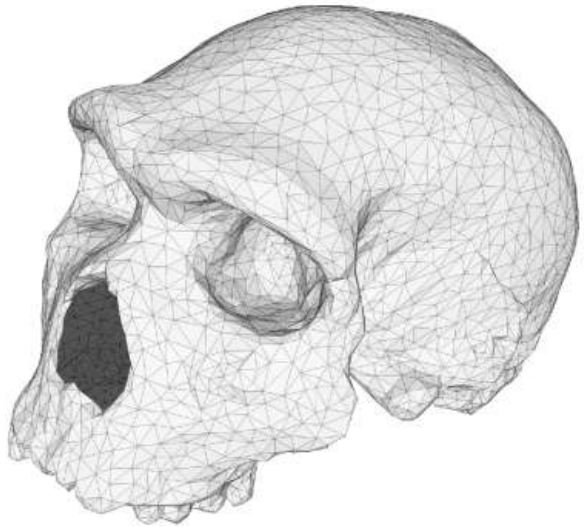
- 缺点
  - 复杂形状建模困难

几何的表示形式

- 显式（Explicit）
  - 三角网格
  - Bezier曲面
  - 细分曲面
  - NURBS
  - 点云
  - ……

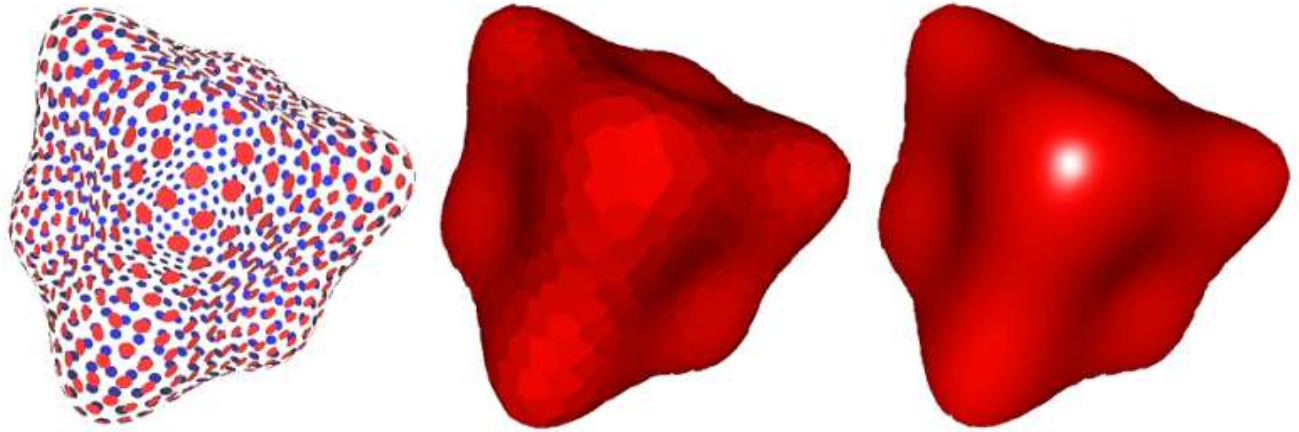






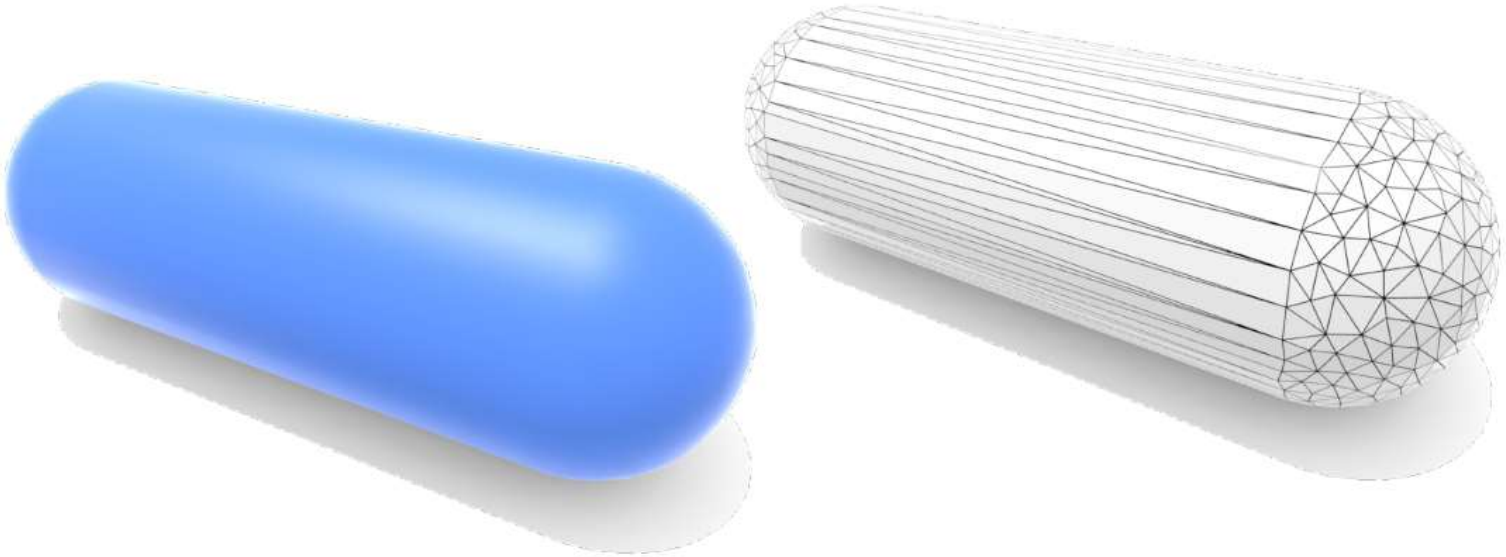
几何的显式表示

- 点云（Point Cloud）
  - 最简单的表示形式：点（x,y,z）的列表
  - 可以方便的表示各种几何类型
  - 适用于大数据集（>>1 point/pixel）
  - 通常需要转换成多边形网格
  - 在欠采样区域难以绘制



几何的显式表示

- 多边形网格（Polygon Mesh）
  - 存储顶点和多边形（常用：三角形、四边形）
  - 易于进行处理/模拟、自适应采样
  - 相较于点云，需要更复杂的数据结构
  - 图形学中最常用的表示形式



几何的显式表示

- OBJ文件（The Wavefront Object File）

➢ 指定了顶点、法线、纹理坐标及其连接关系的文本文件

```
# This is a comment

v 1.000000 -1.000000 -1.000000
v 1.000000 -1.000000 1.000000
v -1.000000 -1.000000 1.000000
v -1.000000 -1.000000 -1.000000
v 1.000000 1.000000 -1.000000
v 0.999999 1.000000 1.000001
v -1.000000 1.000000 1.000000
v -1.000000 1.000000 -1.000000

vt 0.748573 0.750412
vt 0.749279 0.501284
vt 0.999110 0.501077
vt 0.999455 0.750380
vt 0.250471 0.500702
vt 0.249682 0.749677
vt 0.001085 0.750380
vt 0.001517 0.499994
vt 0.499422 0.500239
vt 0.500149 0.750166
vt 0.748355 0.998230
vt 0.500193 0.998728
vt 0.498993 0.250415
vt 0.748953 0.250920

vn 0.000000 0.000000 -1.000000
vn -1.000000 -0.000000 -0.000000
vn -0.000000 -0.000000 1.000000
vn -0.000001 0.000000 1.000000
vn 1.000000 -0.000000 0.000000
vn 1.000000 0.000000 0.000001
vn 0.000000 1.000000 -0.000000
vn -0.000000 -1.000000 0.000000

f 5/1/1 1/2/1 4/3/1
f 5/1/1 4/3/1 8/4/1
f 3/5/2 7/6/2 8/7/2
f 3/5/2 8/7/2 4/8/2
f 2/9/3 6/10/3 3/5/3
f 6/10/4 7/6/4 3/5/4
f 1/2/5 5/1/5 2/9/5
f 5/1/6 6/10/6 2/9/6
f 5/1/7 8/11/7 6/10/7
f 8/11/7 7/12/7 6/10/7
f 1/2/8 2/9/8 3/13/8
f 1/2/8 3/13/8 4/14/8
```

Q&A

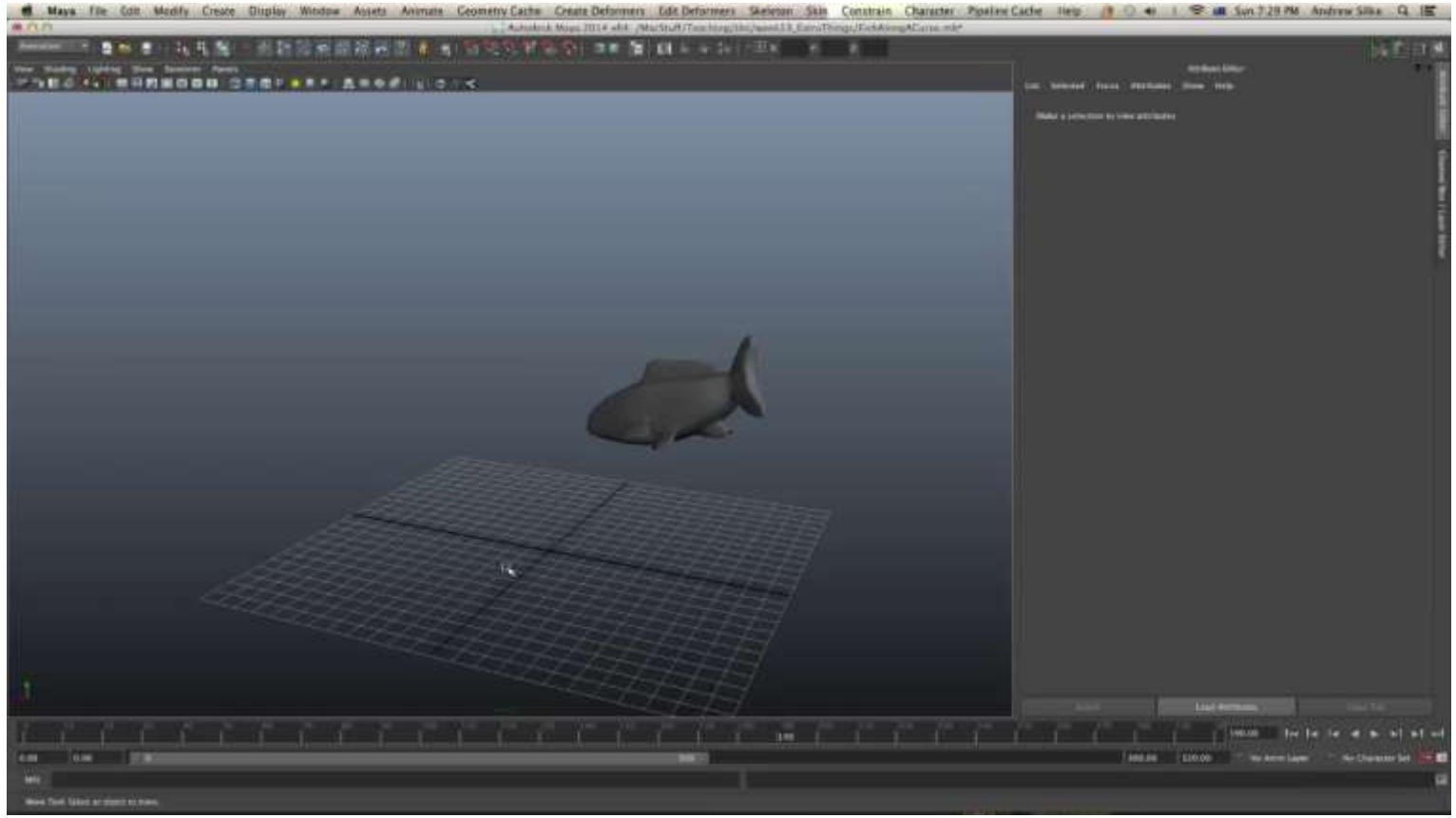
曲线的应用

- 相机路径

Flythrough of the future Perth City Link https://youtu.be/rIJMuQPwr3E

曲线的应用

- 动画曲线



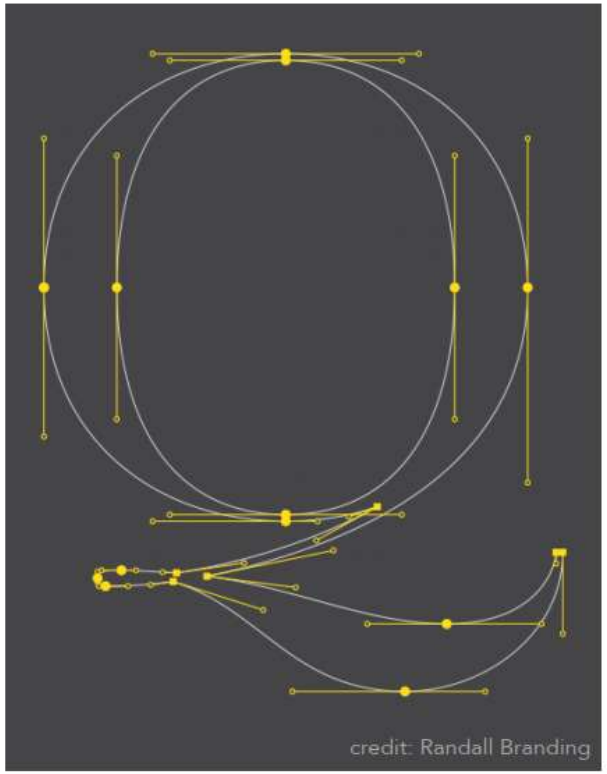
Maya Animation Tutorial https://youtu.be/b-o5wtZlJPc

曲线的应用

- 定义字体

The Quick Brown
Fox Jumps Over
The Lazy Dog

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz 0123456789



Baskerville font - represented as piecewise cubic Bézier curves

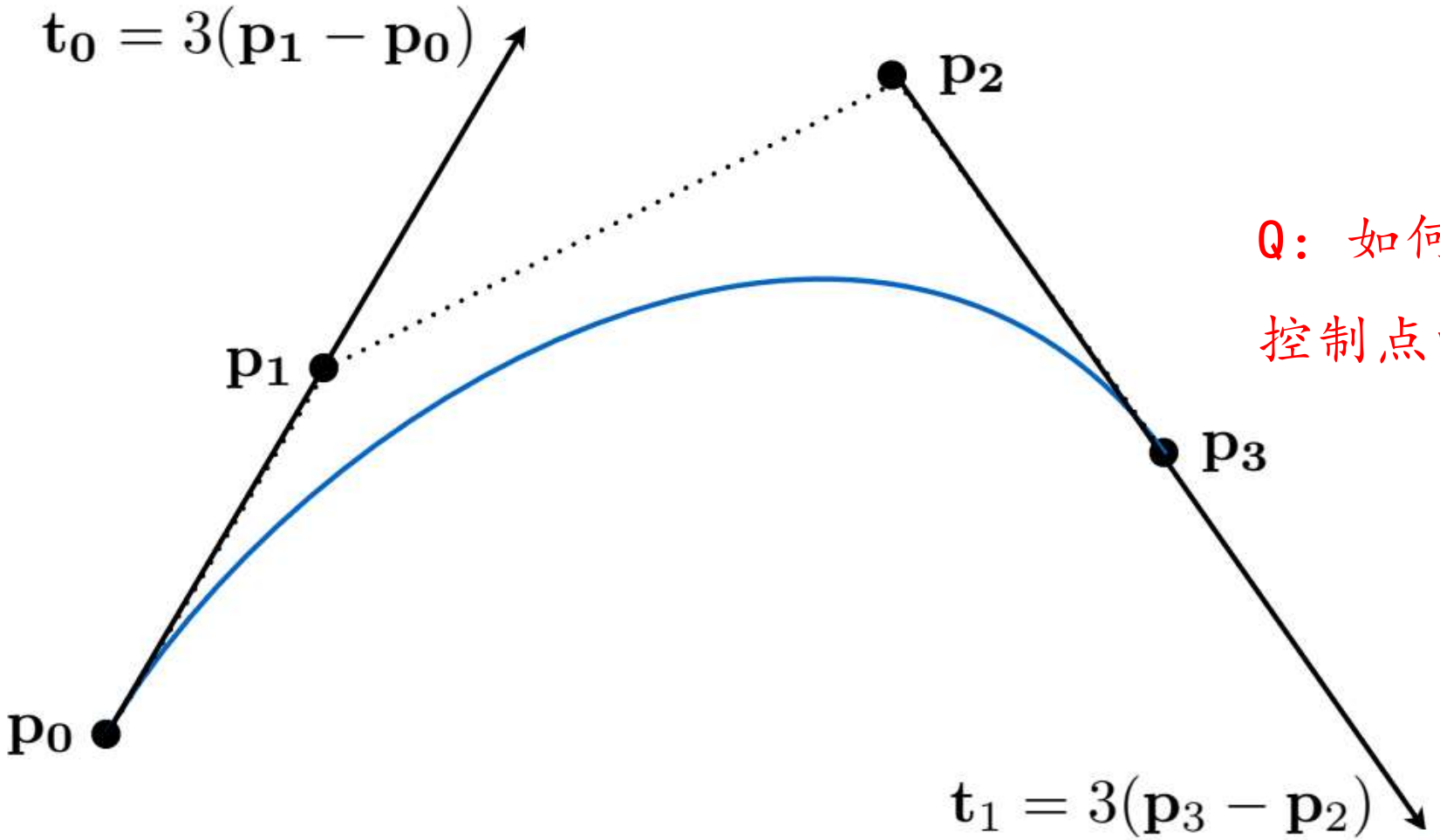
贝塞尔曲线（Bezier Curve）

- 1962年，法国雷诺汽车公司，P.E.Bezier工程师提出一种新的基于“逼近”思想的参数曲线。以此方法为基础完成一种自由型曲线和曲面的设计系统UNISURF，1972年在雷诺汽车公司正式使用。
- 设计者先大致给出设计曲线的初始轮廓，然后可以很直观地以交互的方式通过选择和调整这个初始轮廓来改变曲线的形状，直到获得满意的形状，这种曲线易于计算机实现，称为Bezier曲线。

Pierre Bézier
1910 – 1999

Q：贝塞尔曲线是隐式表示还是显式表示？

贝塞尔曲线



Q：如何画一条有n个控制点的贝塞尔曲线？

de Casteljau算法

- 递推算法

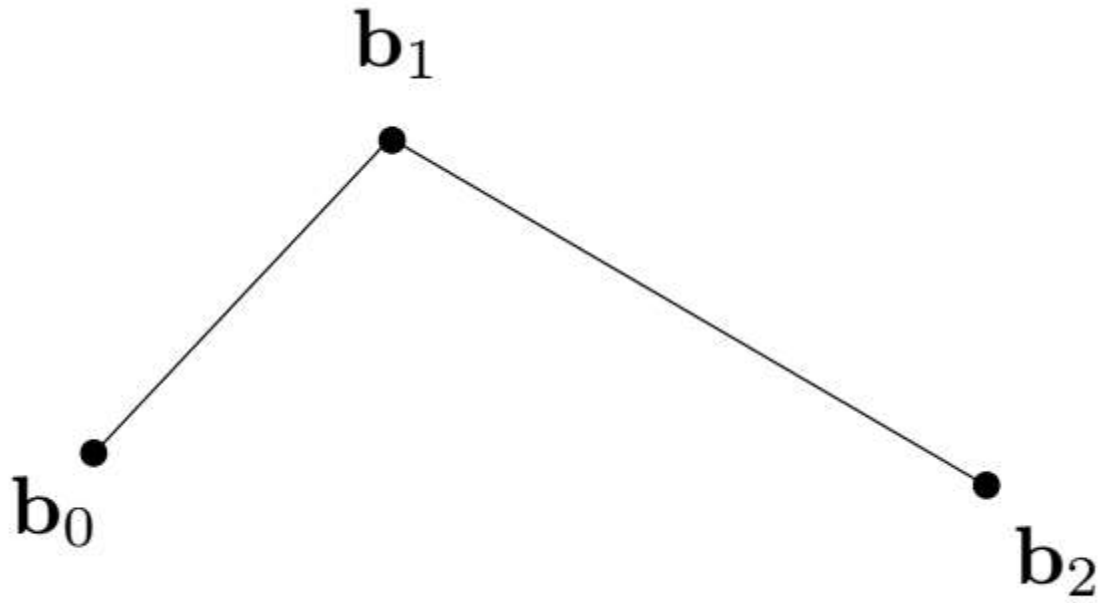
➢ 基于分割递推的任意阶次Bezier曲线的离散生成算法

Q:为什么要这么做?

Paul de Casteljau
b. 1930

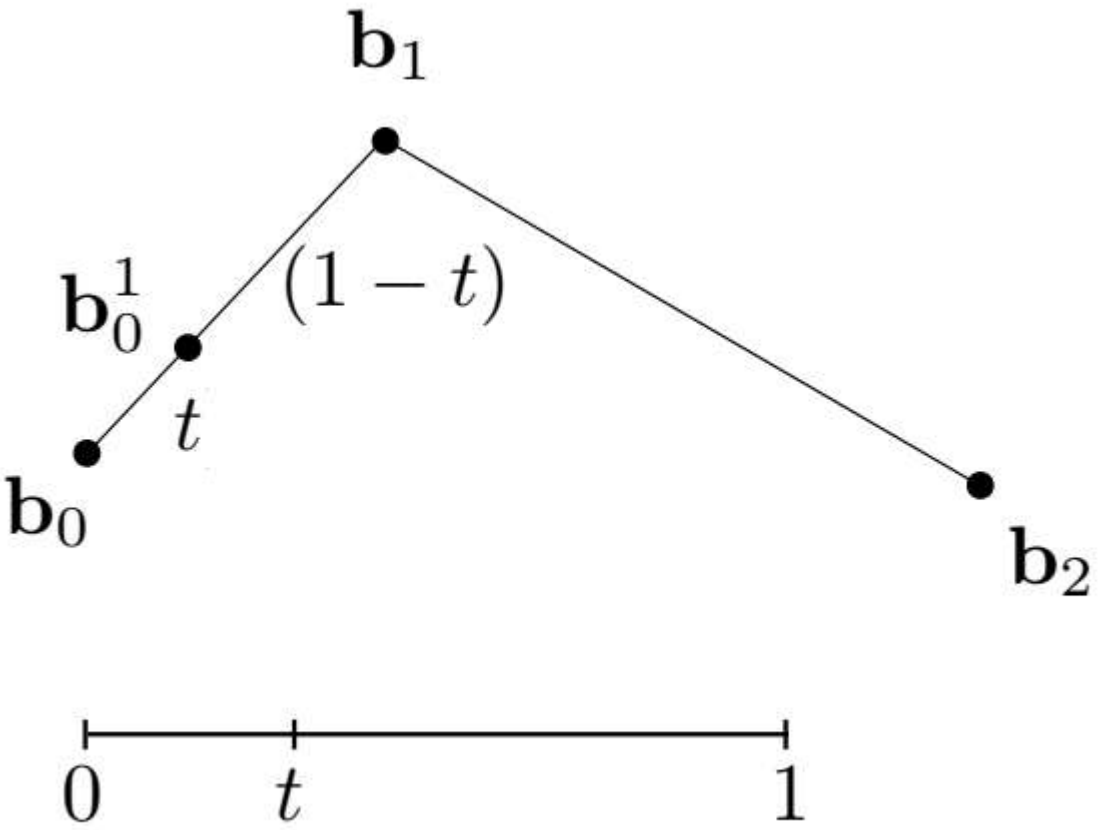
de Casteljau算法

- 考虑三个控制点（二次Bezier曲线）



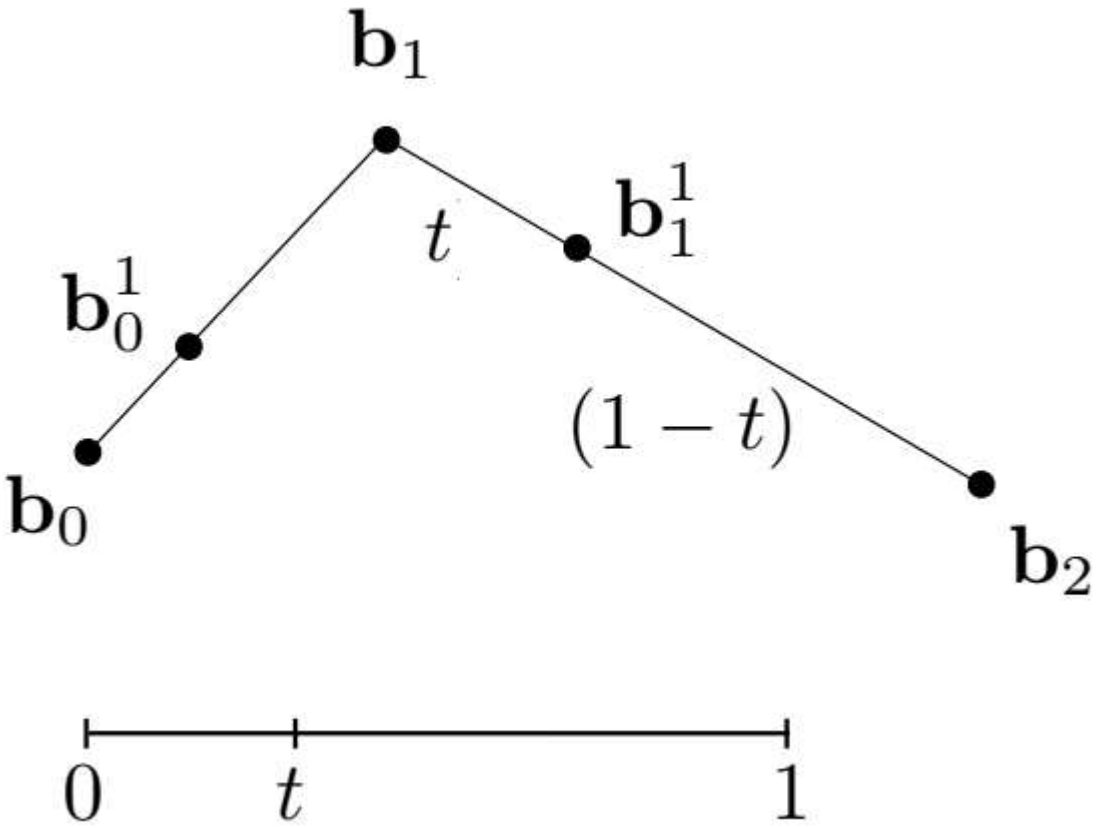
de Casteljau算法

- 利用线性插值插入一个点



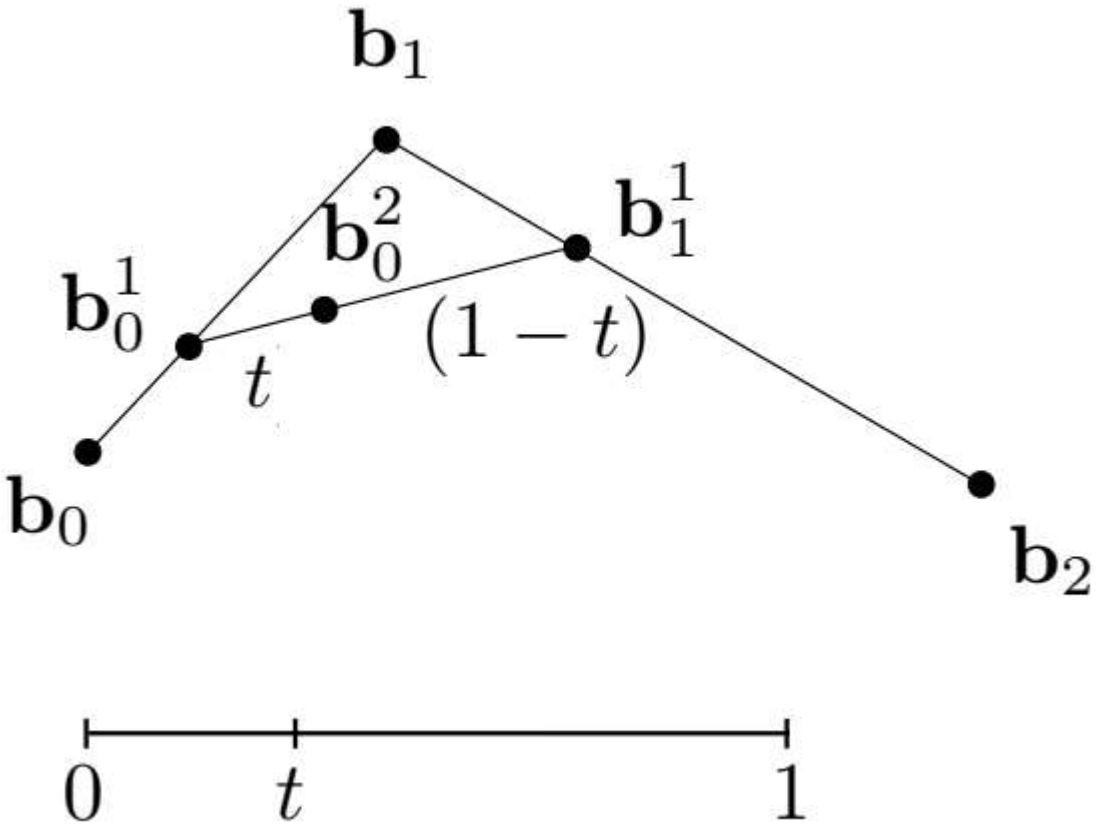
de Casteljau算法

- 同理插入另一个点



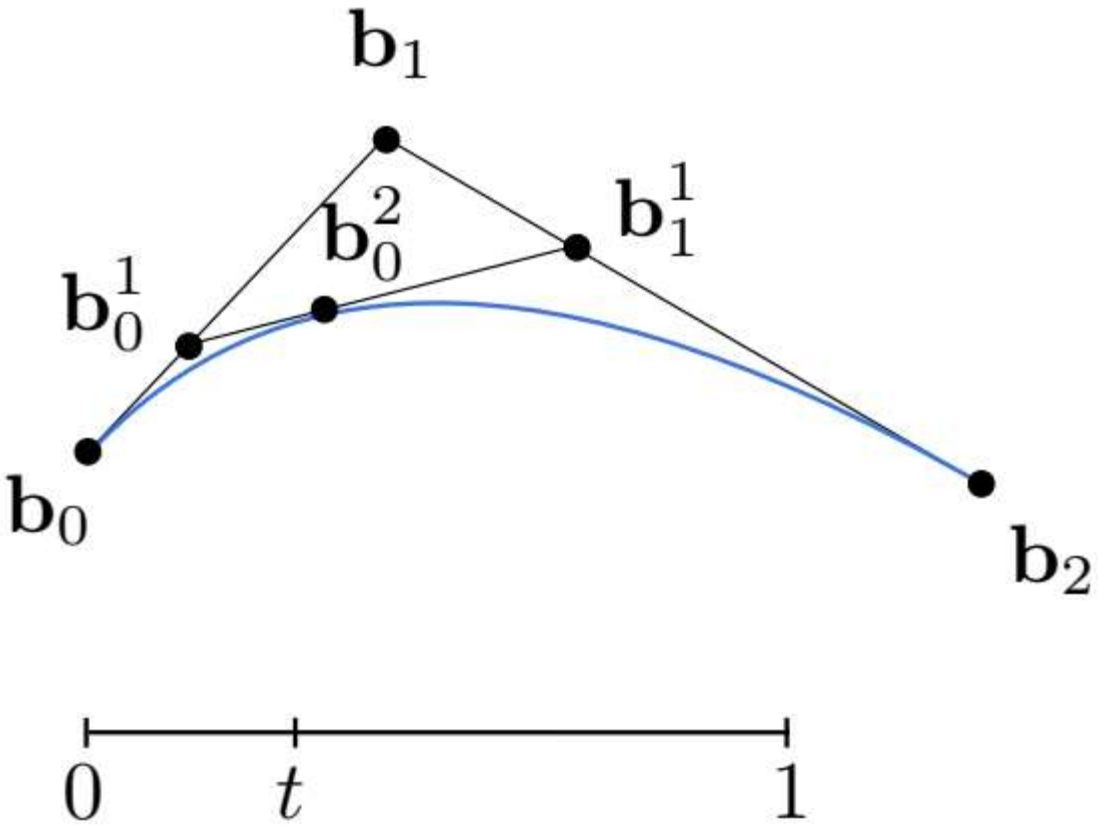
de Casteljau算法

- 递归的继续求解



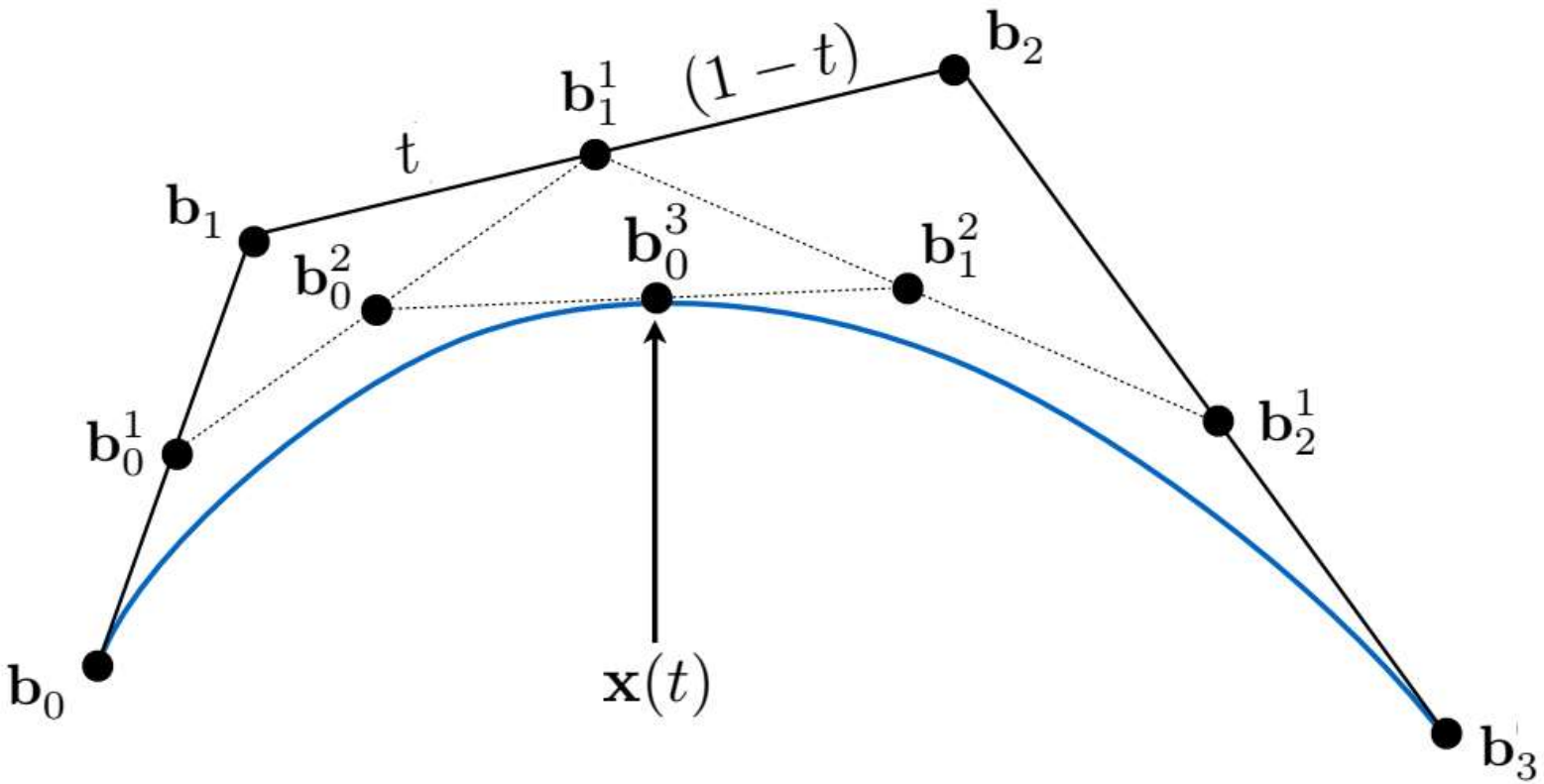
de Casteljau算法

- 对[0,1]区间中的每一个t,做同样的计算



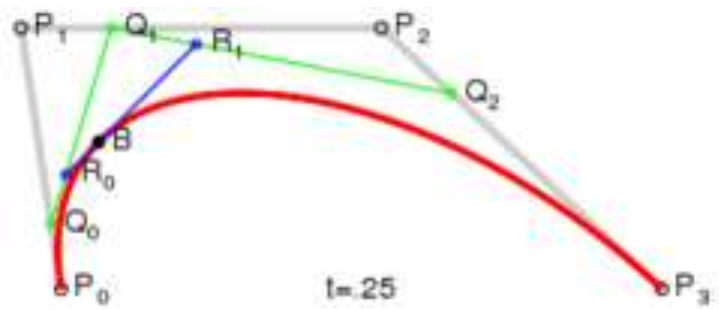
de Casteljau算法

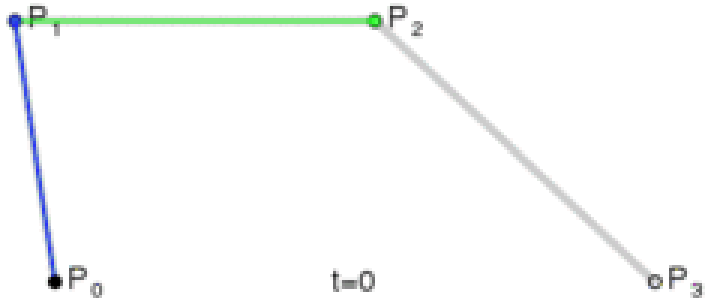
- 考虑四个控制点（三次Bezier曲线）



de Casteljau算法

- 考虑四个控制点（三次Bezier曲线）



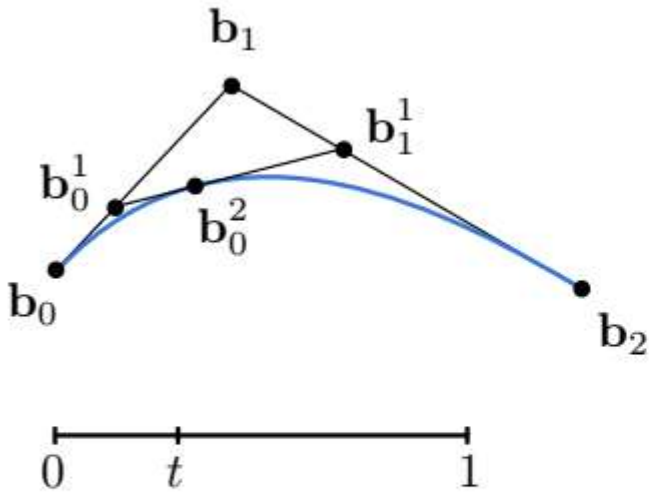


https://acko.net/files/fullfrontal/fullfrontal/wdcode/online.html

Q&A

贝塞尔曲线的代数公式

- 考虑三个控制点（二次Bezier曲线）



$$\mathbf{b}_0^1(t) = (1-t)\mathbf{b}_0 + t\mathbf{b}_1$$

$$\mathbf{b}_1^1(t) = (1-t)\mathbf{b}_1 + t\mathbf{b}_2$$

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

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

贝塞尔曲线的代数公式

- n次Bezier曲线

$$\mathbf{b}^n(t) = \mathbf{b}_0^n(t) = \sum_{j=0}^{n} \mathbf{b}_j B_j^n(t)$$

Bézier curve order n
(vector polynomial of degree n)

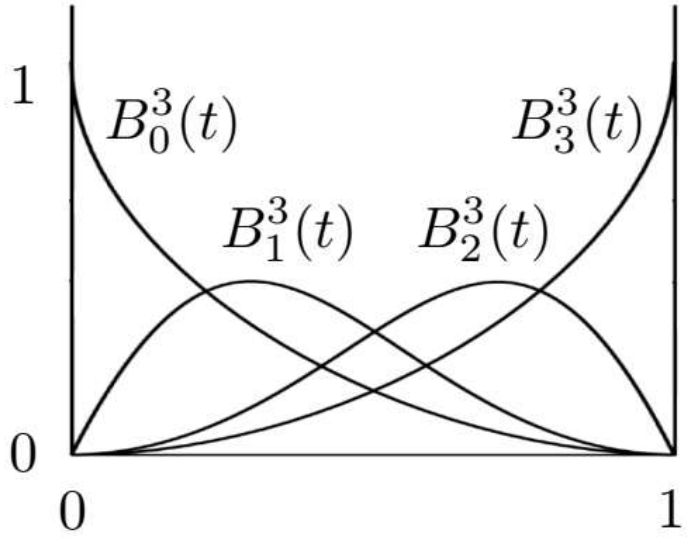
Bernstein polynomial
(scalar polynomial of degree n)

Bézier control points
(vector in R^N)

贝塞尔曲线的代数公式

- Bernstein多项式（Bezier基函数）： $B_i^n(t) = \binom{n}{i} t^i (1-t)^{n-i}$
- 三次Bezier曲线

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



Sergei N. Bernstein
1880 – 1968

贝塞尔曲线的性质

- 端点位置

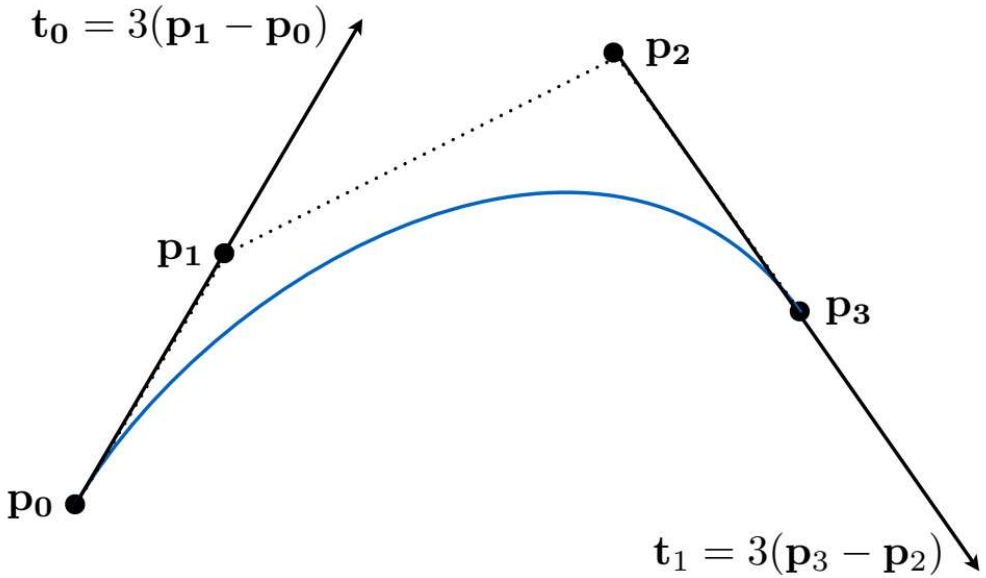
$$P(t)|_{t=0} = P_0$$

$$P(t)|_{t=1} = P_n$$

- 端点切向量

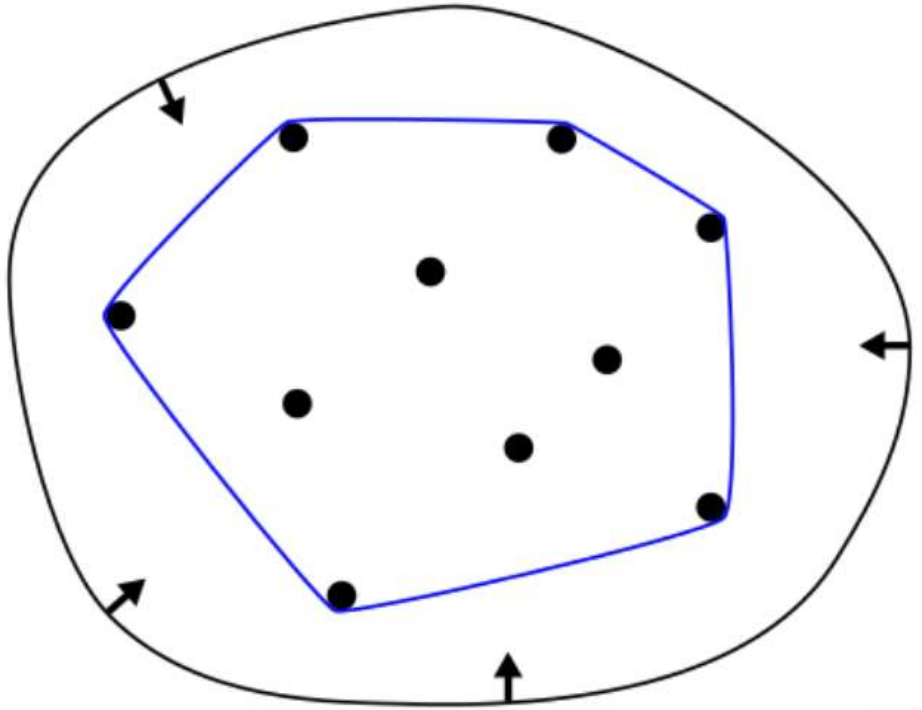
$$P'(t)|_{t=0} = n(P_1 - P_0)$$

$$P'(t)|_{t=1} = n(P_n - P_{n-1})$$



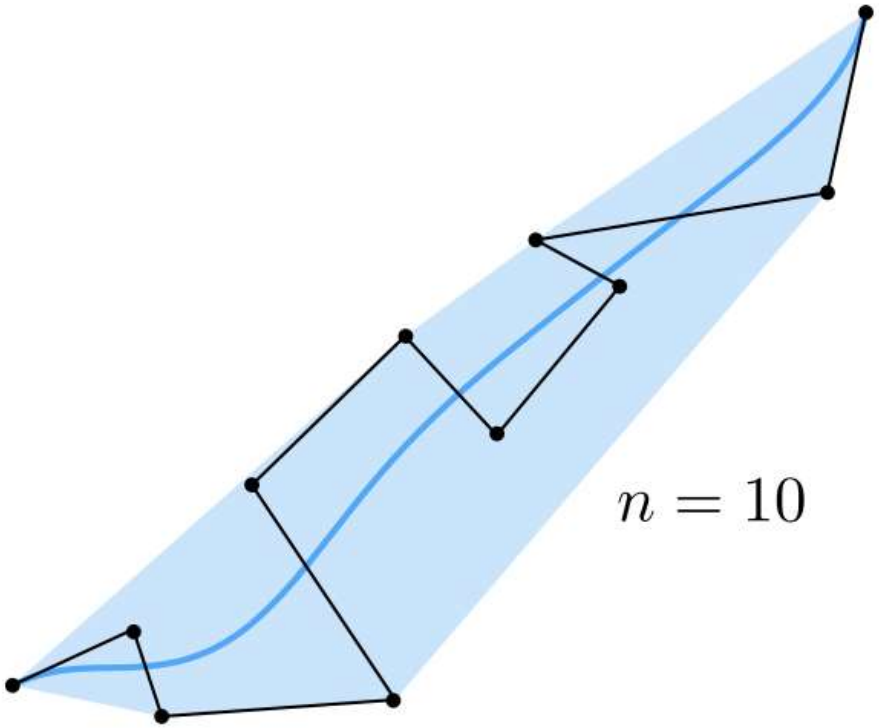
贝塞尔曲线的性质

- 仿射变换性质
  - ➢ 通过变换控制点达到变换曲线的目的
- 凸包性质
- 曲线位于控制点的凸包内



高阶贝塞尔曲线

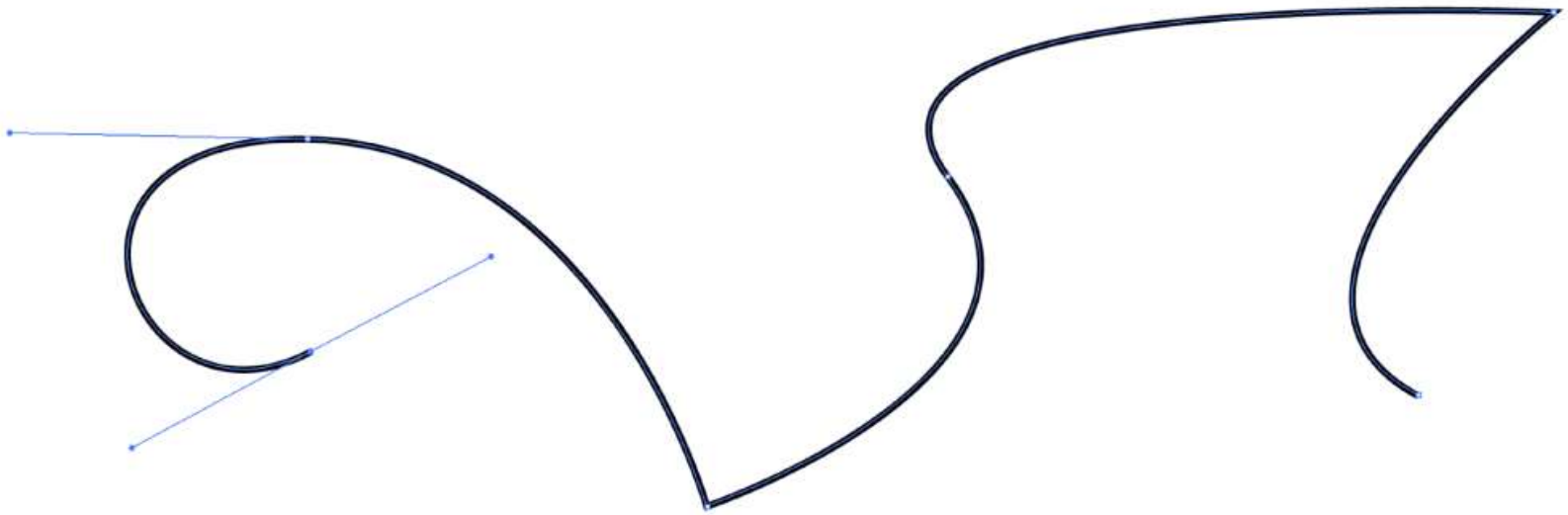
Q：高阶Bezier曲线存在什么问题？如何解决？



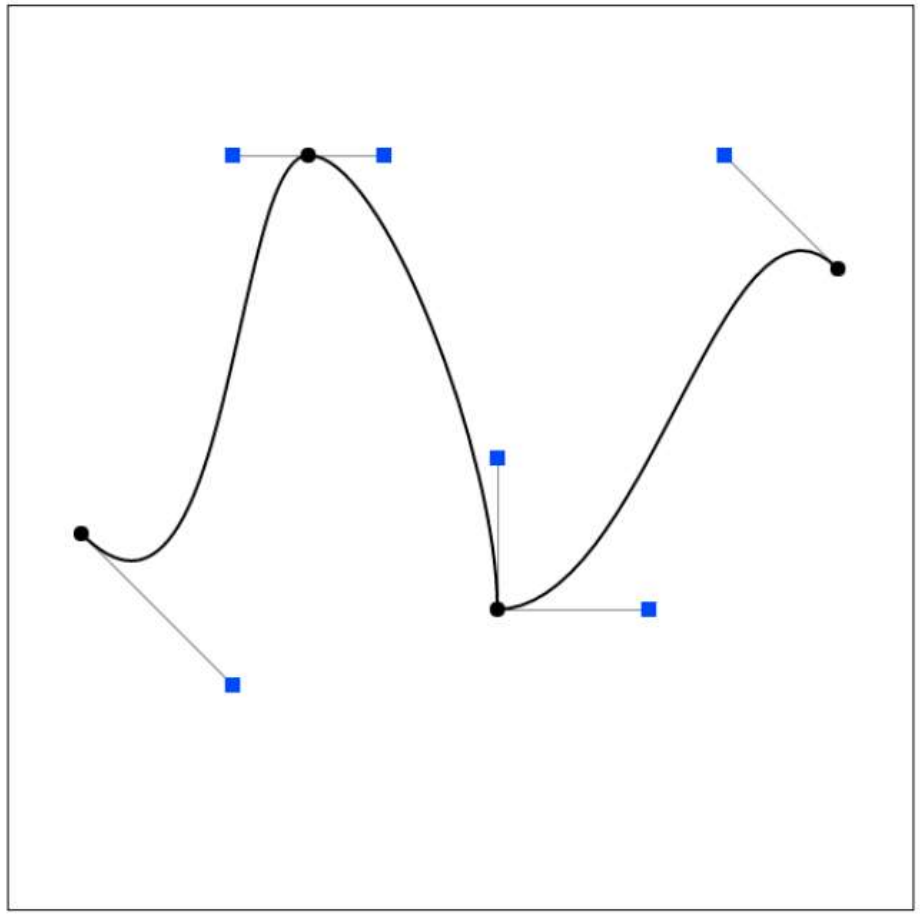
不直观，难以控制！并不常用！

分段贝塞尔曲线

- 通过连接多段低阶Bezier曲线实现
- 分段三次Bezier曲线最为常见



分段贝塞尔曲线



David Eck http://math.hws.edu/eck/cs424/notes2013/canvas/bezier.html

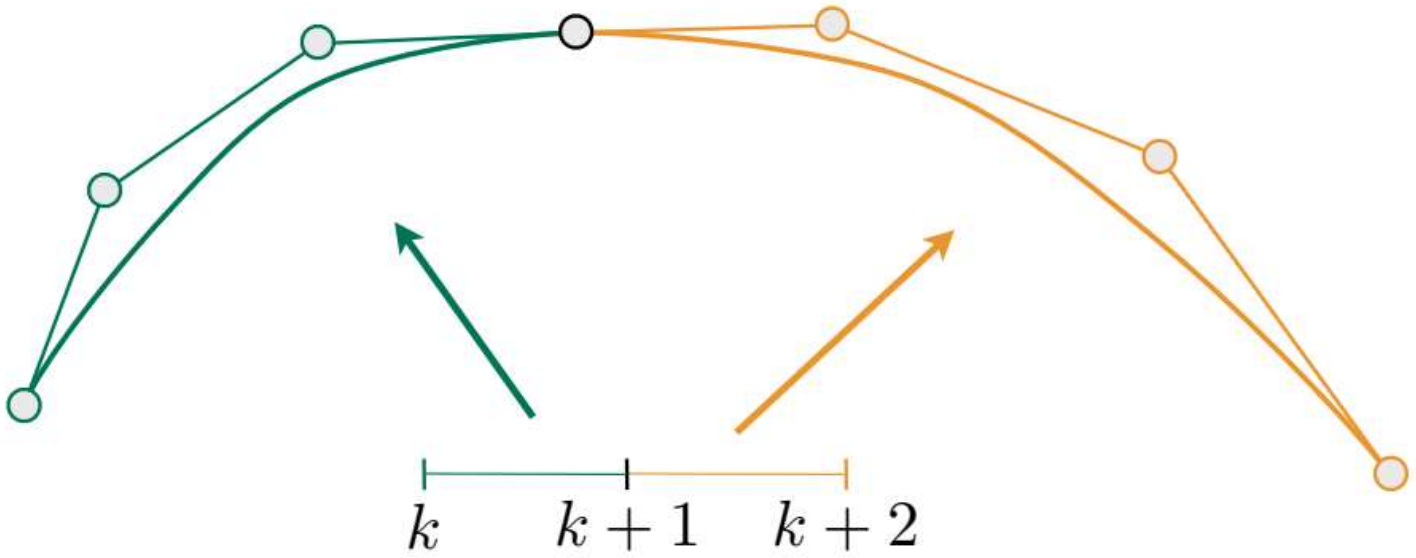
分段贝塞尔曲线的连续性

Two Bézier curves

$$\mathbf{a} : [k, k+1] \to \mathbb{R}^N$$

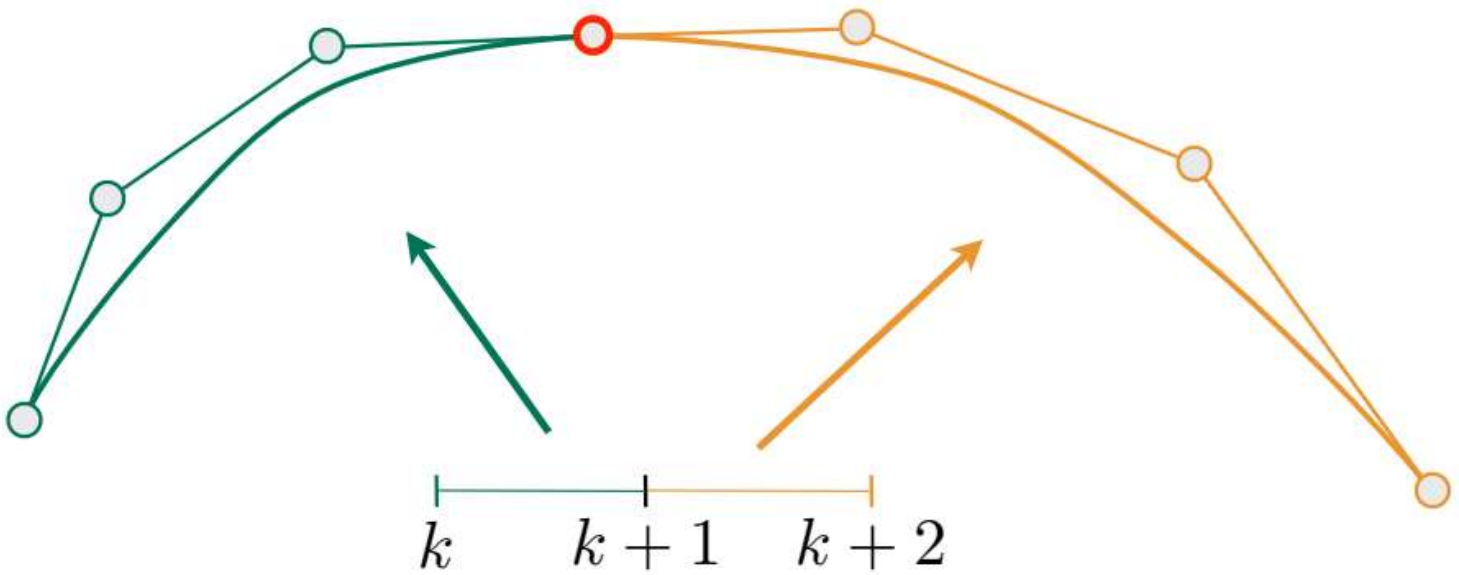
$$\mathbf{b} : [k+1, k+2] \to \mathbb{R}^N$$

Assuming integer partitions here,
can generalize



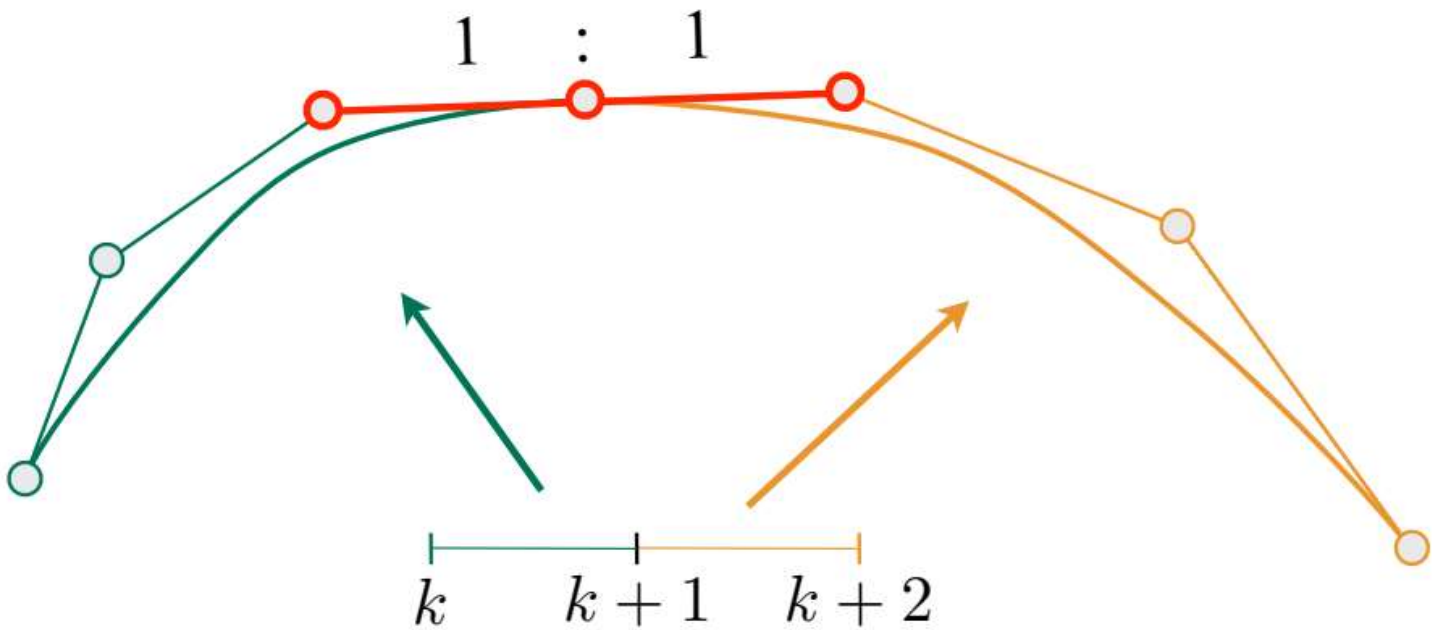
分段贝塞尔曲线的连续性

C^0 continuity: $\mathbf{a}_n = \mathbf{b}_0$



分段贝塞尔曲线的连续性

C^1 continuity: $\mathbf{a}_n = \mathbf{b}_0 = \frac{1}{2}(\mathbf{a}_{n-1} + \mathbf{b}_1)$



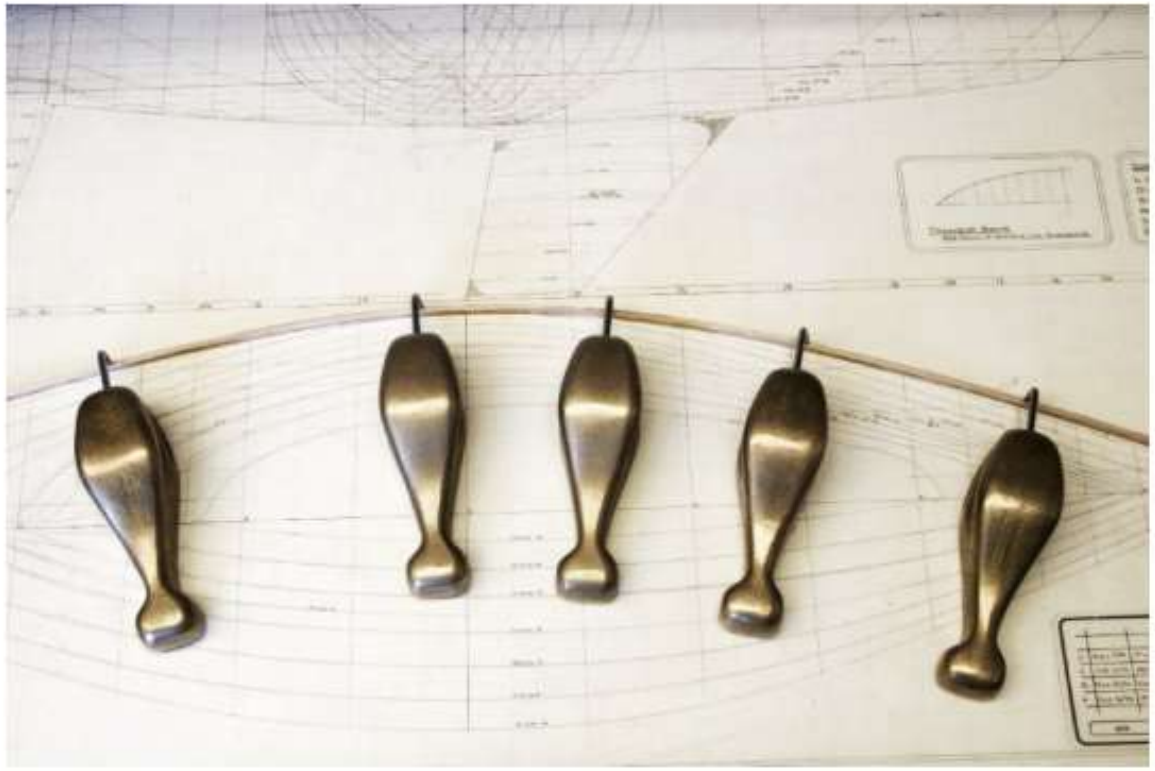
贝塞尔曲线

- 优点
  - 形状控制直观，设计灵活，应用较为广泛

- 缺点
  - 控制顶点数较多时，多边形对曲线的控制能力减弱
  - 控制顶点数增多时，生成曲线的阶数也增高
  - 曲线拼接需要附加条件，不太灵活
  - 局部控制能力弱：因为Bezier基函数的值在（0,1）开区间内均不为零，曲线上任意一点都是所有给定顶点的加权平均

样条（Spline）

- 优于Bezier曲线之处
  - 局部修改能力强
  - 易于拼接
  - ……
- B样条
- NURBS
- ……

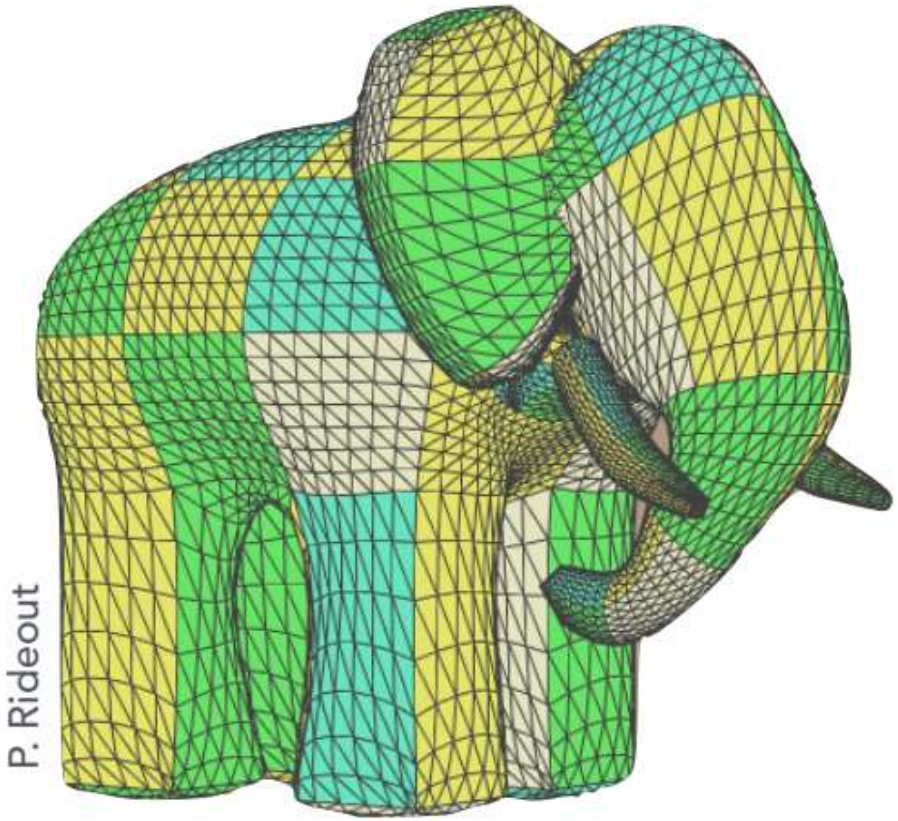


A Real Draftsman’s Spline

http://www.alatown.com/spline-history-architecture/

Q&A

贝塞尔曲面

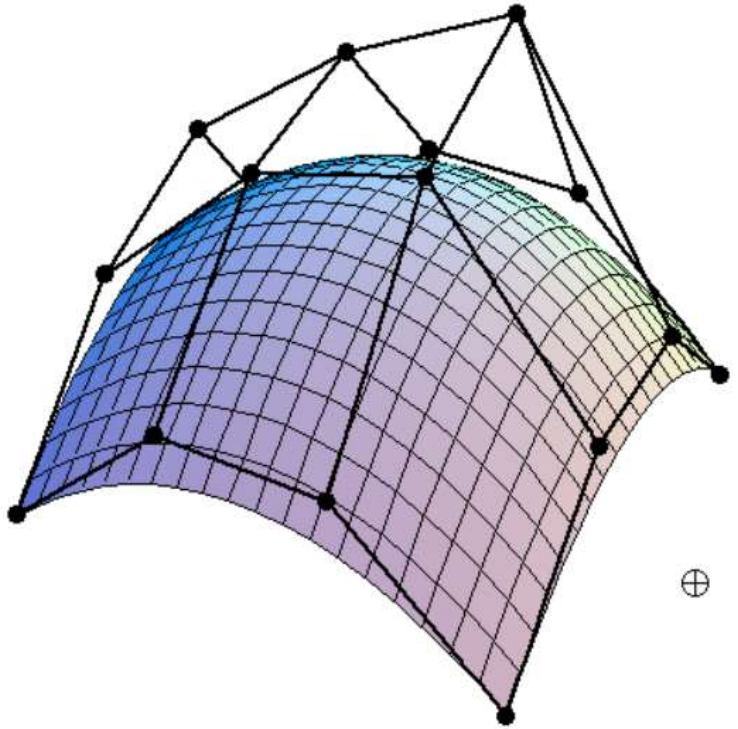


Ed Catmull's "Gumbo" model

Utah Teapot

贝塞尔曲面

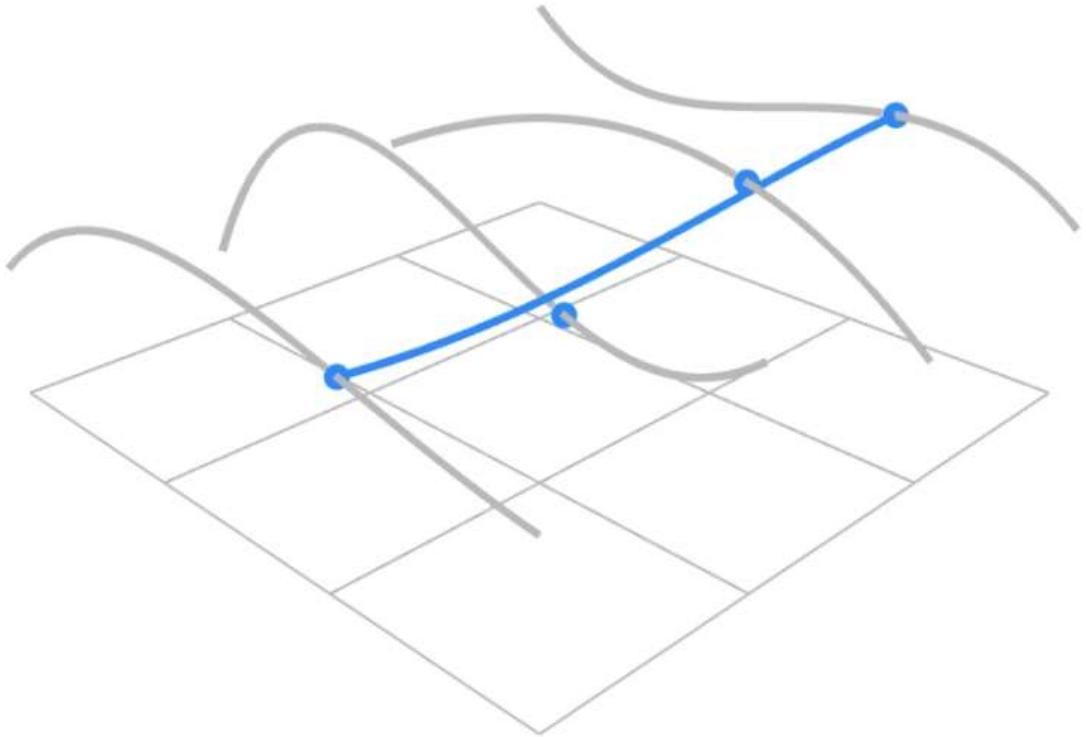
- 双三次Bezier曲面片



Bezier surface and 4 x 4 array of control points

贝塞尔曲面

- 双三次Bezier曲面片

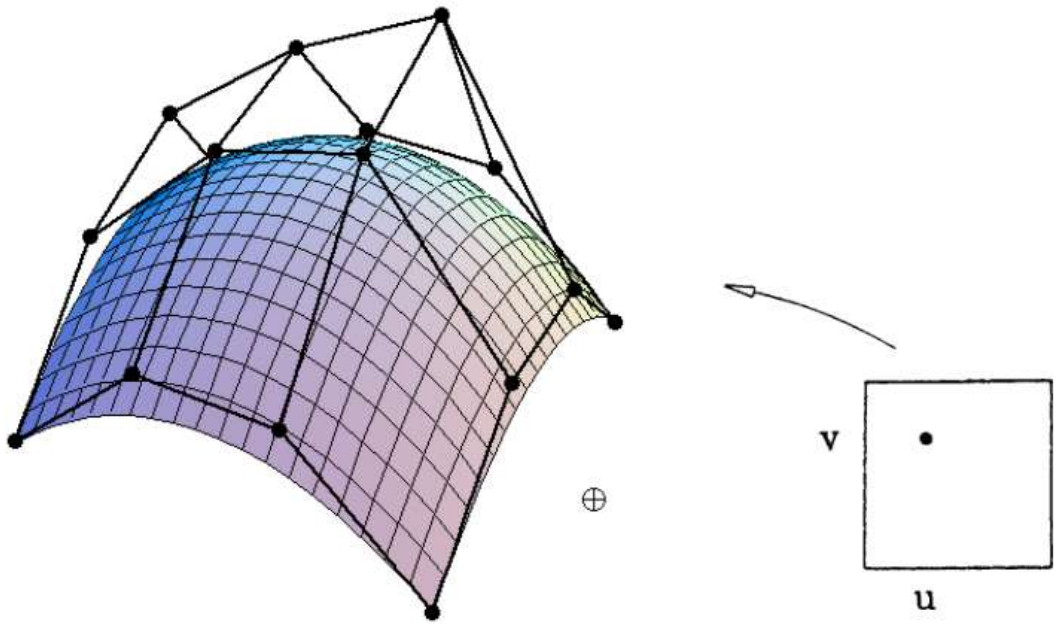


https://acko.net/files/fullfrontal/fullfrontal/wdcode/online.html

贝塞尔曲面

- 双三次Bezier曲面片的计算

Input: 4x4 control points
Output is 2D surface parameterized by (u,v) in $[0,1]^2$



贝塞尔曲面

- 双三次Bezier曲面片的计算

Goal: Evaluate surface position corresponding to (u,v)

(u,v)-separable application of de Casteljau algorithm

- Use de Casteljau to evaluate point u on each of the 4 Bezier curves in u. This gives 4 control points for the "moving" Bezier curve
- Use 1D de Casteljau to evaluate point v on the "moving" curve