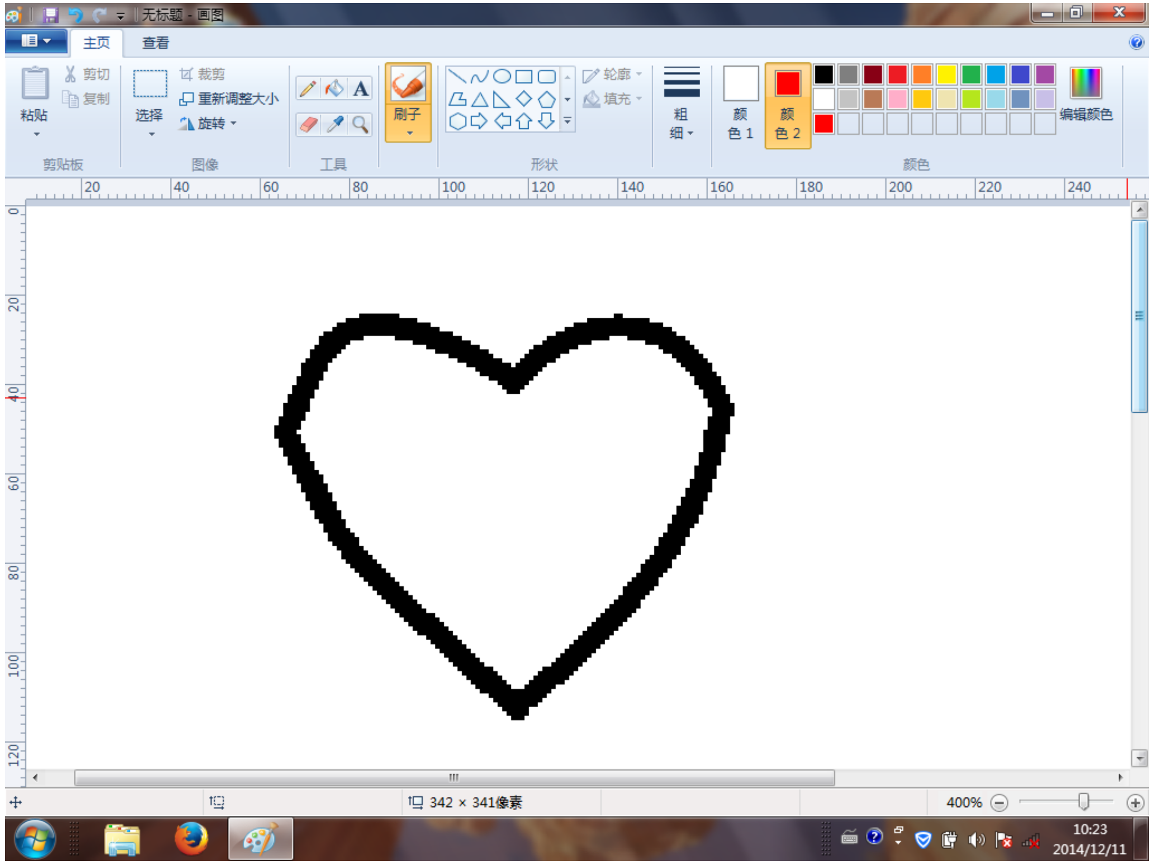
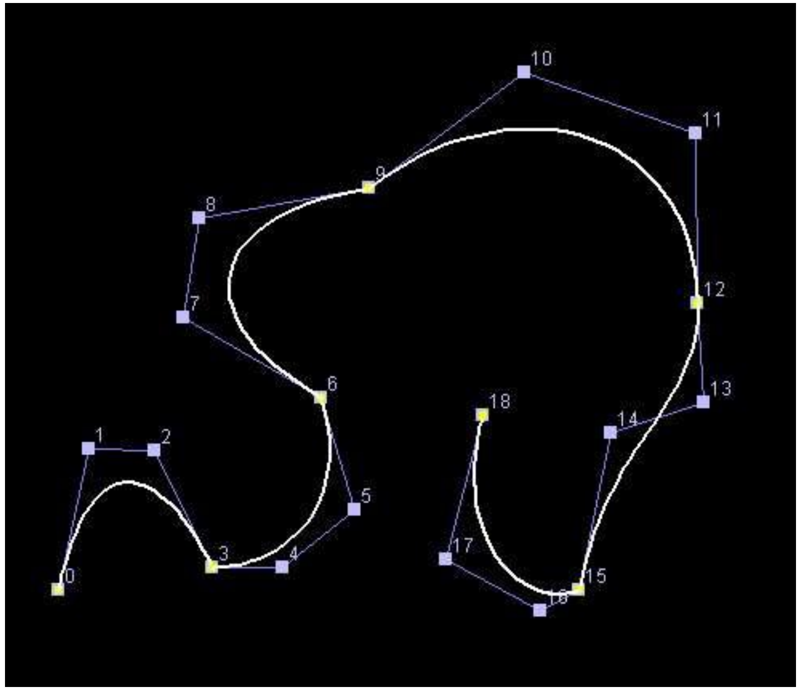
Your Shape: Please exercise it!!!





历史

曲线曲面历史：

在近20年来CAGD在多变量曲线插值、Coons曲面、Bezier曲线曲面、B样条、有理曲线、曲面、多边形曲面片等有了长远发展。

1963年，美国波音(Boeing)飞机公司的弗格森(Ferguson)首先提出了将曲线曲面表示为参数的矢函数方法。他最早引入参数三次曲线，构造了组合曲线和由四角点的位置矢量及两个方向的切矢定义的弗格森双三次曲面片。

1964年，麻省理工学院(MIT)的孔斯(Coons)发表了一个具有一般性的曲面描述方法，给定围成封闭曲线的四条边界就可定义一块曲面片。目前在CAGD中得到广泛应用的是它的特殊形式——孔斯双三次曲面片。它与弗格森双三次曲面片一样，都存在形状控制与连接的问题。

由舍恩伯格(Schoenberg) 1964年提出的样条函数提供了解决连接问题的一种技术。用于自由曲线曲面描述的样条方法是它的参数形式，即参数样条曲线、曲面。样条方法用于解决插值问题，在构造整体达到某种参数连续阶（指可微性）的插值曲线、曲面是很方便的，但不存在局部形状调整的自由度，样条曲线和曲面的形状难以预测。

历史

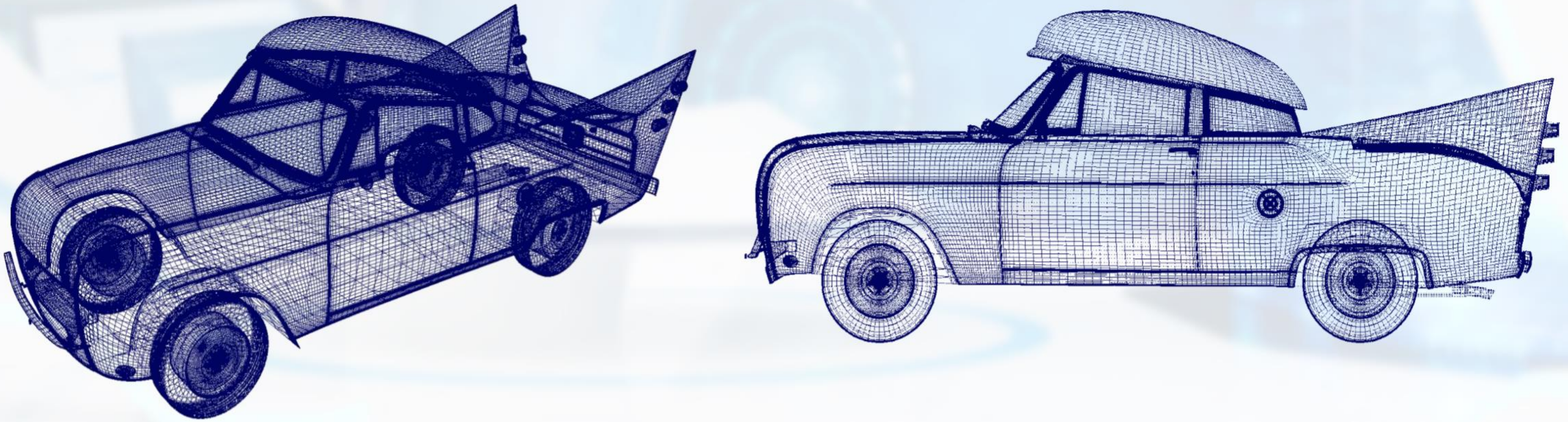
法国雷诺(Renault)汽车公司的贝齐尔(Bezier) 以逼近为基础研究了曲线曲面的构造，于1971年提出了一种由控制多边形定义曲线的方法。设计者只要移动控制顶点就可方便地修改曲线的形状，而且形状的变化完全在预料之中。Bezier方法简单易用，又漂亮地解决了整体形状控制问题，至今一些著名软件如UGII、UNISURF、DUCT等仍保留着Bezier曲线、曲面。但它也还存在连接问题和局部修改问题。

德布尔(de Boor)1972年给出了关于B样条的一条标准算法。

美国通用汽车公司的Gordon和Riesenfeld1974年将B样条理论应用于自由曲线曲面描述，提出了B样条曲线曲面。B样条克服了Bezier方法的不足之处，较成功地解决了局部控制问题，又轻而易举地在参数连续基础上解决了连接问题。具有局部修改方便、形态控制灵活、直观优良特性，成为构造曲线、曲面的主要工具。

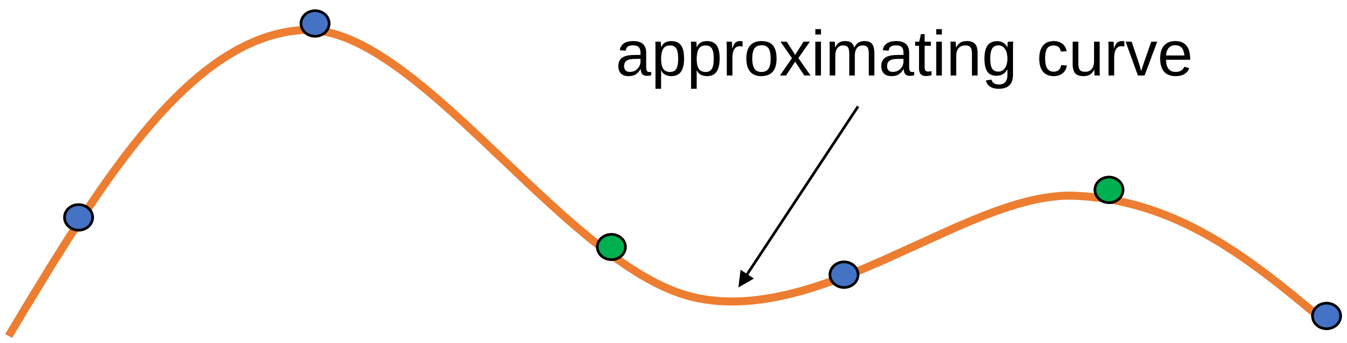
应用

样条的典型计算机辅助设计（CAD）应用包括：汽车车身设计、飞机和航天飞机表面的设计、船体设计及家庭应用。



样条spline

- 样条(spline)一词来源于造船工程师设计船体形状的可弯曲的木条或者金属条。
- 每一个样条用钉子固定在恰当的位置，近似出一条曲线。



样条用于设计曲线和曲面形状，将绘制的图形数字化及指定场景中对象的动画路径或照相机位置。

样条曲线（ spline curve ）指由多项式曲线段连接而成的曲线，在每段的边界处满足特定的连续性条件。

内容提要

- Curves and Surfaces Representation
    - **参数化**多项式表示
    - 分段及光滑性连接条件
- Curves and Surfaces Design
- Curves and Surfaces Render

1)Explicit Representation显式表示

- Most familiar form of curve in 2D

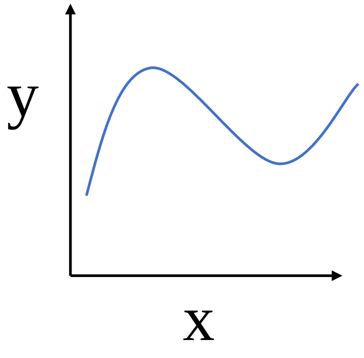
  y=f(x), y=mx+b

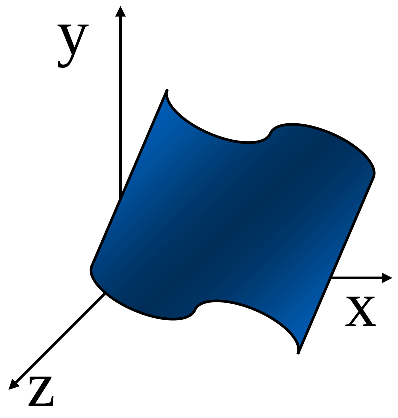
- Extension to 3D
  - y=f(x), z=g(x) define a curve
  - **z = f(x,y) defines a surface**

- Cannot represent all curves

  Vertical lines~垂线

  Circles~圆





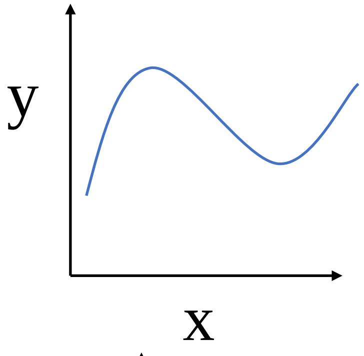
2)Implicit Representation隐式表示

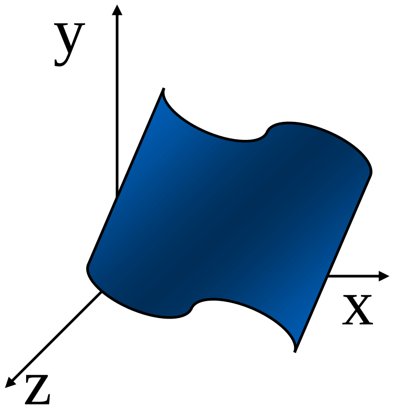
✓Two dimensional curve(s)

$g(x,y)=0$

✓Three dimensions $g(x,y,z)=0$ defines a surface

✓Intersect two surface to get a curve





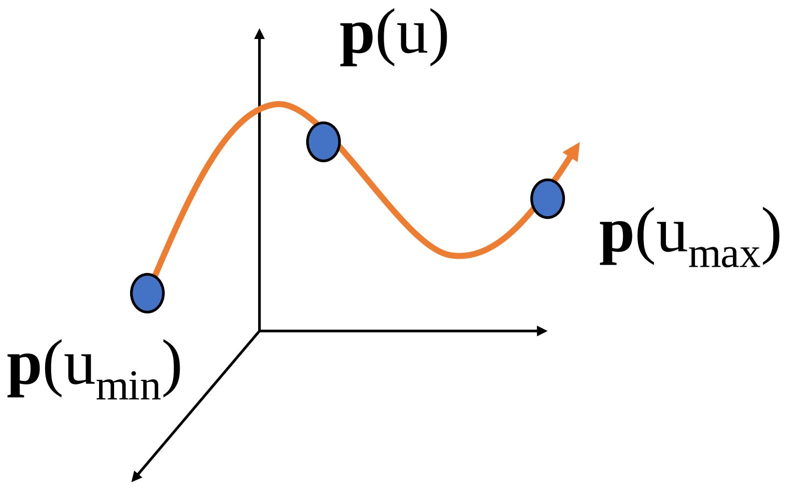
- Much more robust
  - All lines $ax+by+c=0$
  - Circles $x^2+y^2-r^2=0$

3)Parametric Representation参数表示

- Separate equation for each spatial variable

$$x=x(u)$$
$$y=y(u)$$
$$z=z(u)$$

$$\mathbf{p}(u)=[x(u), y(u), z(u)]^T$$

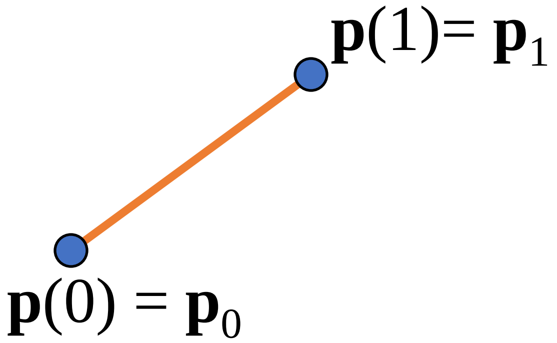


$u_{max} \geq u \geq u_{min}$

a.参数直线Parametric Lines

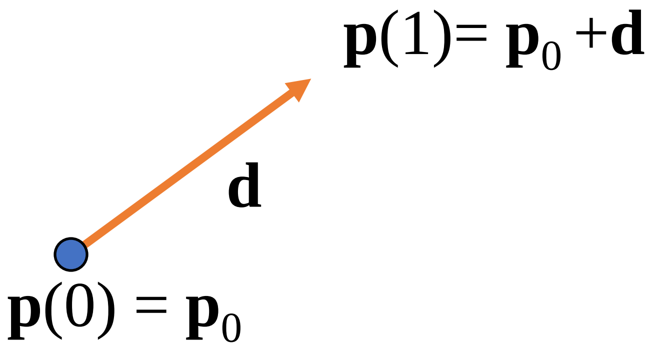
Line connecting two points $\mathbf{p}_0$ and $\mathbf{p}_1$

$$\mathbf{p}(u)=(1-u)\mathbf{p}_0+u\mathbf{p}_1$$



Ray from $\mathbf{p}_0$ in the direction $\mathbf{d}$

$$\mathbf{p}(u)=\mathbf{p}_0+u\mathbf{d}$$



- normalize u to be over the interval (0,1)

b.Parametric Planes参数平面

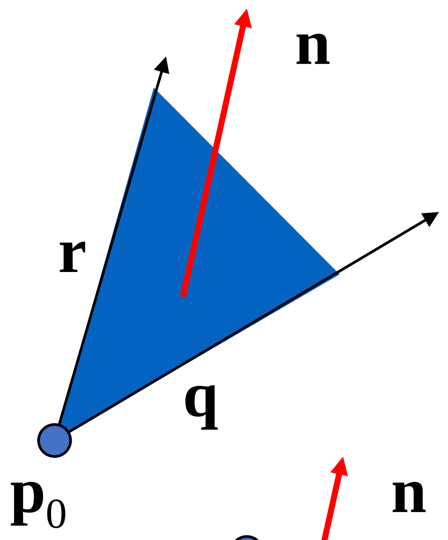
point-vector form

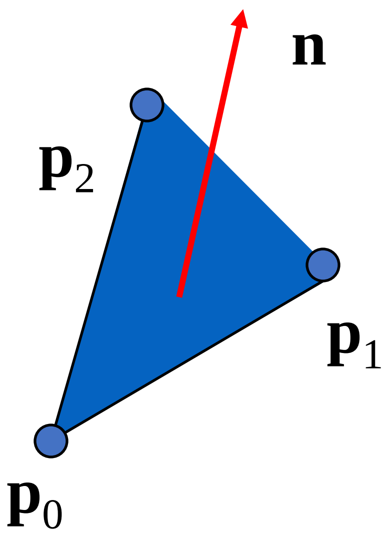
$$\mathbf{p}(u,v)=\mathbf{p}_0+u\mathbf{q}+v\mathbf{r}$$

three-point form

$$\mathbf{q} = \mathbf{p}_1 - \mathbf{p}_0$$
$$\mathbf{r} = \mathbf{p}_2 - \mathbf{p}_0$$
$$\mathbf{n} = \mathbf{q} \times \mathbf{r}$$

$$P(u,v)=p0+u(p1-p0)+v(p2-p0)$$

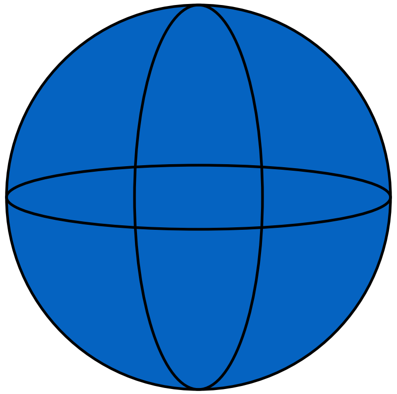




c.Parametric Sphere参数球面

$$x(\theta, \phi) = r \cos\theta \sin\phi$$
$$y(\theta, \phi) = r \sin\theta \sin\phi$$
$$z(\theta, \phi) = r \cos\phi$$

$$360 \geq \theta \geq 0$$
$$180 \geq \phi \geq 0$$



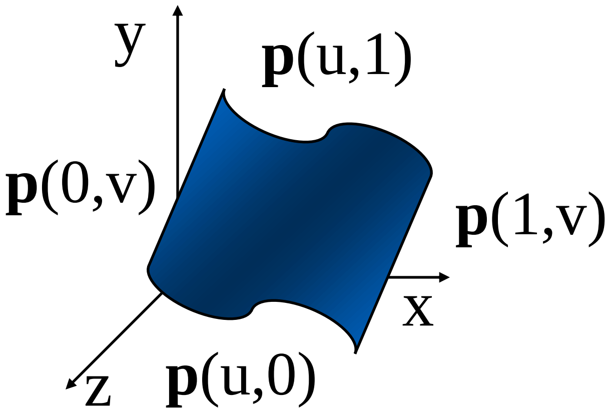
θ constant: circles of constant longitude

ϕ constant: circles of constant latitude

Parametric Surfaces参数曲面

- Surfaces require 2 parameters

$$x=x(u,v)$$
$$y=y(u,v)$$
$$z=z(u,v)$$
$$\mathbf{p}(u,v) = [x(u,v), y(u,v), z(u,v)]^T$$



- Want same properties as curves:
  - Smoothness平滑的
  - Differentiability可微的(求导）
  - Ease of evaluation易于求值的

设计原则Design criteria

- **There are <u>many ways</u> to represent curves and surfaces，**

  Want a representation that is

  - Stable稳定
  - Easy to rendering易于绘制
  - Local control of shape 形状的局部控制
  - Ability to evaluate derivative 可导性
  - Smooth and continuity平滑性

Parametric Polynomial Curves参数多项式曲线

$$x(u) = \sum_{i=0}^{N} c_{xi} u^i \quad y(u) = \sum_{j=0}^{M} c_{yj} u^j \quad z(u) = \sum_{k=0}^{L} c_{zk} u^k$$

- If N=M=L =K, we need to determine 3(N+1) coefficients
- Equivalently we need 3(N+1) independent conditions

$$\mathrm{p}(u) = \sum_{k=0}^{L} c_k u^k$$

After normalizing u（规范化U）, each curve is written

p(u)=[x(u), y(u), z(u)]^T, （$1 \geq u \geq 0$）

Why Polynomials（为什么用多项式函数）

$$p(u) = \sum_{k=0}^{L} c_k u^k$$

- Easy to evaluate (求值)
- Continuous（连续） and differentiable（微分） everywhere

Parametric Surfaces Normals曲面求导

- **一阶导-切矢量**

  P'(t)=dP/dt=[dx/dt dy/dt dz/dt]
  t∈[0,1]

  切矢量反映曲线对t的变化速度。

  方向趋向于该点的切线方向。

$$P'(t) = \frac{dP}{dt}$$

二阶导-曲率

P''(u)=[x y z]=[x''(u) y''(u) z''(u)]   u∈[0,1]
曲率=|P''(u)|。单位切矢量对弧长的转动率。
曲率反映了曲线对参数t的变化的加速度。

$$P''(t) = \frac{d^2P}{dt^2}$$

$$P^k(t) = \frac{d^kP}{dt^k}$$

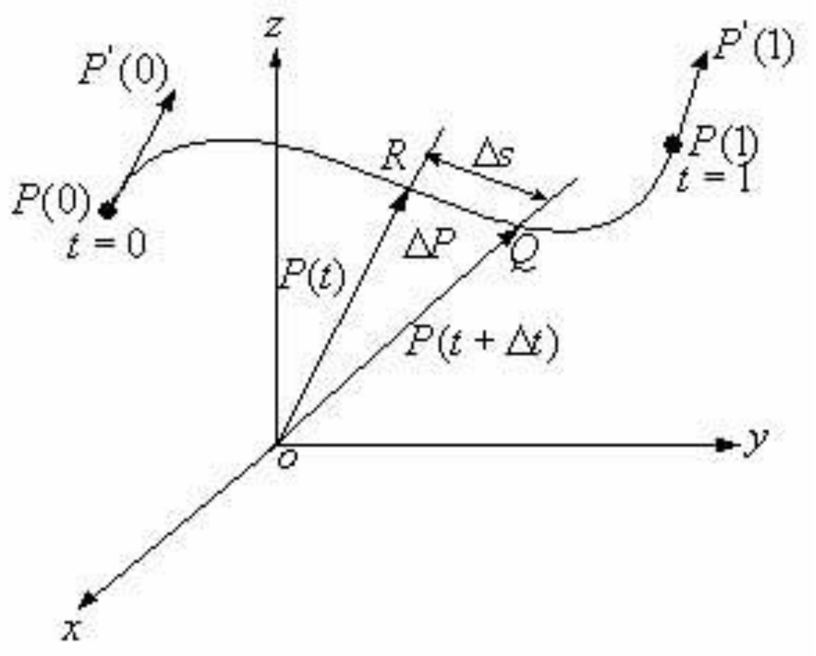


图3.1.1 表示一条参数曲线的有关矢量

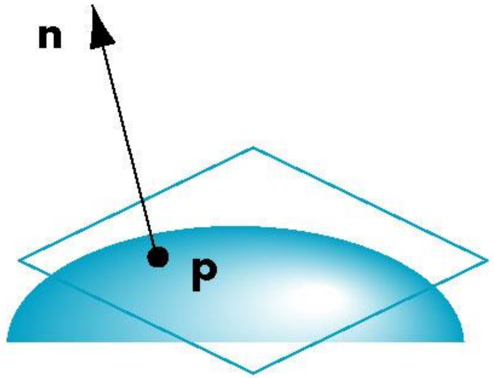
Parametric Surfaces Normals曲面法向量

differentiate with respect to $\mathbf{u}$ and $\mathbf{v}$ to obtain the normal at any point $\mathbf{p}$

(对u和v求导得到任意点p处的法线)

$$\frac{\partial \mathbf{p}(u,v)}{\partial u} = \begin{bmatrix} \partial x(u,v)/\partial u \\ \partial y(u,v)/\partial u \\ \partial z(u,v)/\partial u \end{bmatrix} \qquad \frac{\partial \mathbf{p}(u,v)}{\partial v} = \begin{bmatrix} \partial x(u,v)/\partial v \\ \partial y(u,v)/\partial v \\ \partial z(u,v)/\partial v \end{bmatrix}$$

$$\mathbf{n} = \frac{\partial \mathbf{p}(u,v)}{\partial u} \times \frac{\partial \mathbf{p}(u,v)}{\partial v}$$

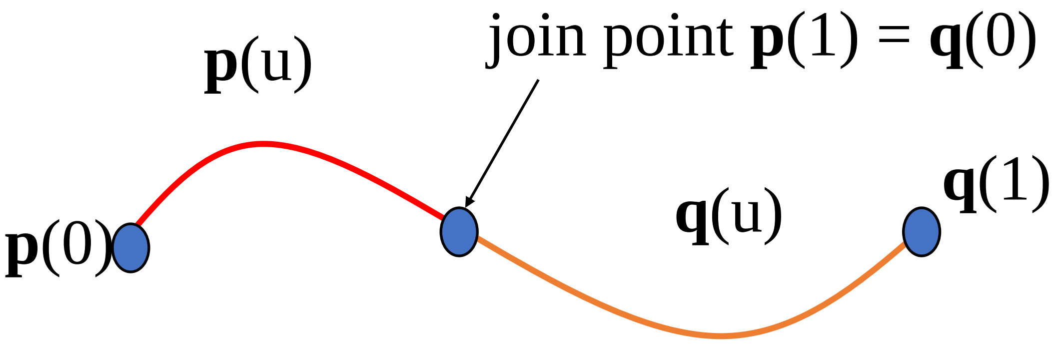


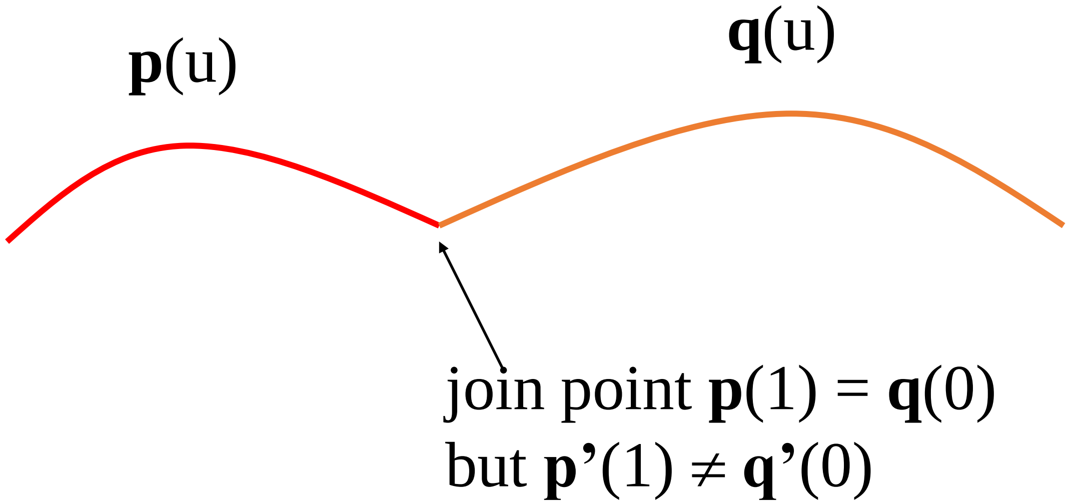
内容提要

- Curves and Surfaces Representation
  - **参数化**多项式表示
  - 分段及光滑性连接条件
- Curves and Surfaces Design
- Curves and Surfaces Render

Curve Segments and Continuity（分段和连续性）

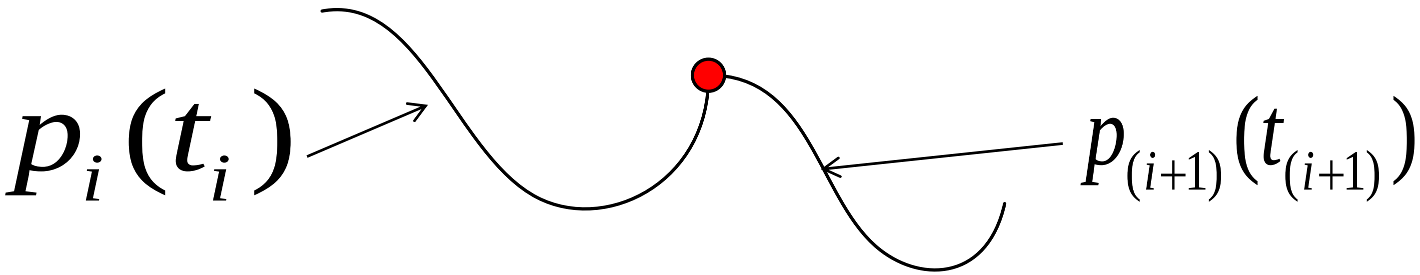
- *In classical numerical methods, we design a single global curve*
- *In computer graphics and CAD, it is better to design small connected curve segments*





Smooth and continuity（光顺和连续性）

- **曲线光顺/光滑：**不存在多余拐点和奇异点（一阶导数）并且曲率变化较小（二阶导数）.即在连接点处平滑。

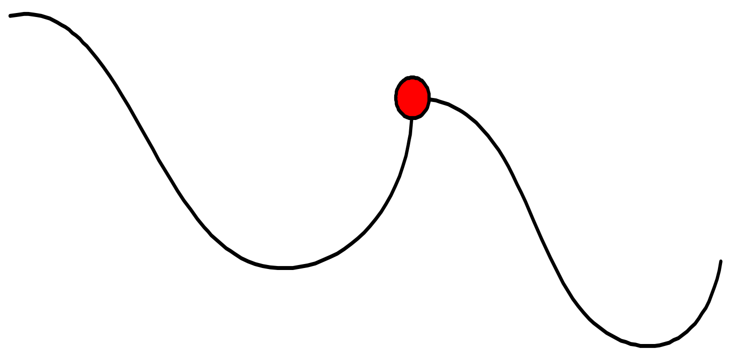


$$p_i = p_i(t_i) \quad t_i \in [t_{i0}, t_{i1}]$$

参数曲线段以参数形式p_i进行描述，i取不同值代表不同的曲线。

0阶-参数连续性

记作C^0连续性，是指曲线的几何位置连接（重合）

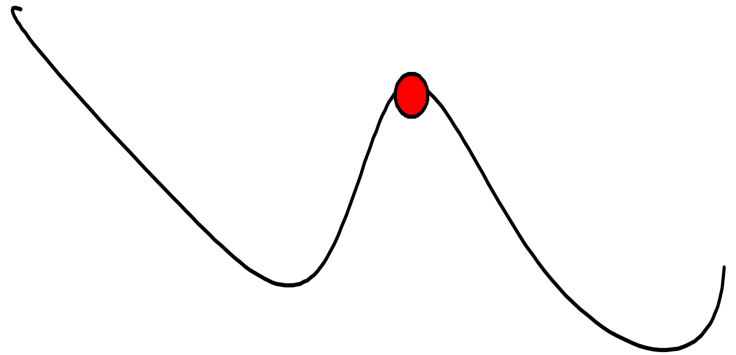


$$p_i(t_{i1}) = p_{(i+1)}(t_{(i+1)0})$$

1阶-参数连续性

记作C1连续性　两个相邻曲线段在相交点重合并有相同的一阶导数

一阶连续性对数字化绘图及一些设计应用已经足够。



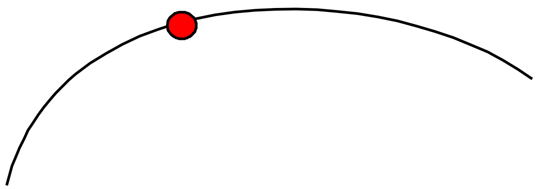
$$p_i(t_{i1}) = p_{(i+1)}(t_{(i+1)0})$$

$$\text{且} p_i'(t_{i1}) = p_{(i+1)}'(t_{(i+1)0})$$

2阶-参数连续性

记作C^2连续性，*两个相邻曲线段交点处重合，并具有相同的一阶和二阶导数。*

一二阶连续性对电影中的动画路径和精密CAD需求有用。



(c)2阶连续性

$$p_i(t_{i1}) = p_{(i+1)}(t_{(i+1)0})$$

$$\text{且}\quad p'_i(t_{i1}) = p'_{(i+1)}(t_{(i+1)0})$$

$$\text{且}\quad p''_i(t_{i1}) = p''_{(i+1)}(t_{(i+1)0})$$

几何连续性

0阶几何连续性，G^0，与0阶参数连续性的定义相同，满足连接点处位置相等。

1阶几何连续性，G^1，指一阶导数在相邻段的交点处成比例，即连接点处位置相等，一阶导数成比列，即方向相同，大小不同。

2阶几何连续性，G^2，指相邻曲线段在连接点处位置重合，一阶和二阶导数均成比例。

- C^n连续保证G^n连续，但反之不然。
- C^n连续的条件比G^n连续的条件要苛刻

连续性练习

证明：下面的两条参数曲线段达到了 C^1 和 G^1 连续并求出达到 C^1 和 G^1 连续的连接点。

$P(t)=(t^2-2t,t)$ $\quad t\in[0,1]$

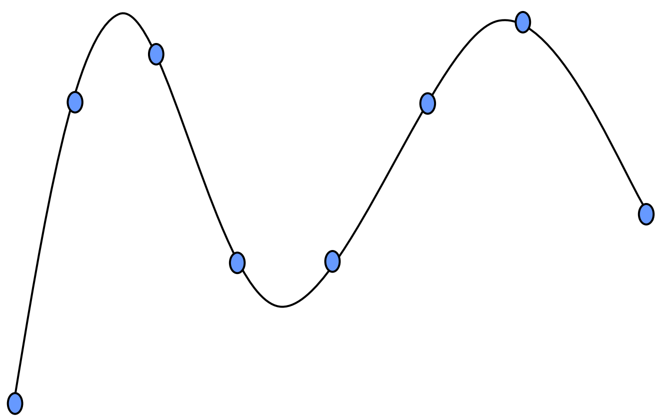
$Q(u)=(u^2-1, u+1)$ $\quad u\in[0,1]$

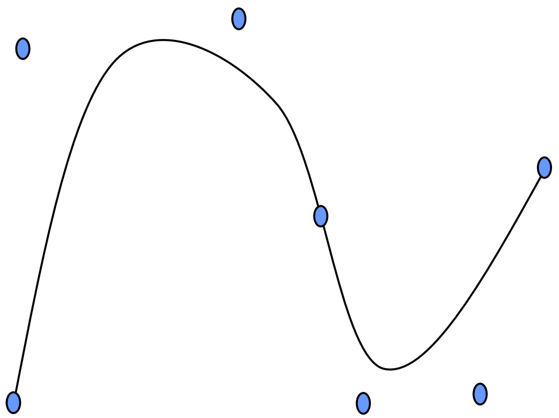
内容提要

- Curves and Surfaces Representation
  - **参数化**多项式表示
  - 分段表示及光滑性连接条件
- Curves and Surfaces Design
  - 曲线构造方法 – 插值和逼近
  - 插值样条 （Natural,Hermite ,Cardinal ）
  - 逼近样条 （Bezier, B-spline, NURBS）
- Curves and Surfaces Render

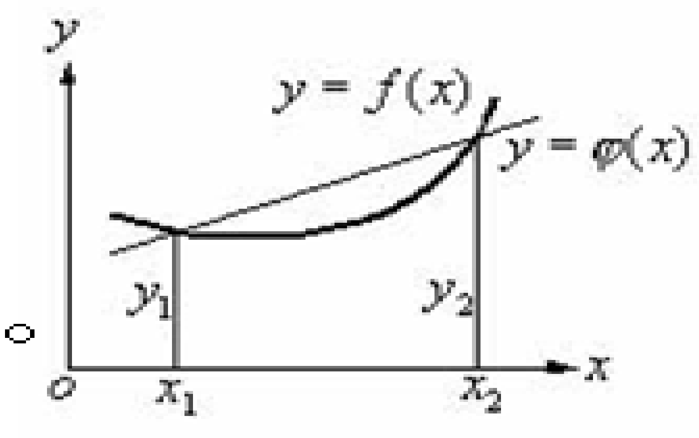
曲线模型构造方法

- **构造样条曲线的基本思路：拟合**
- 给定一些曲线上的点（型值点）或接近曲线的点（控制点），***插值或逼近***建立模型。

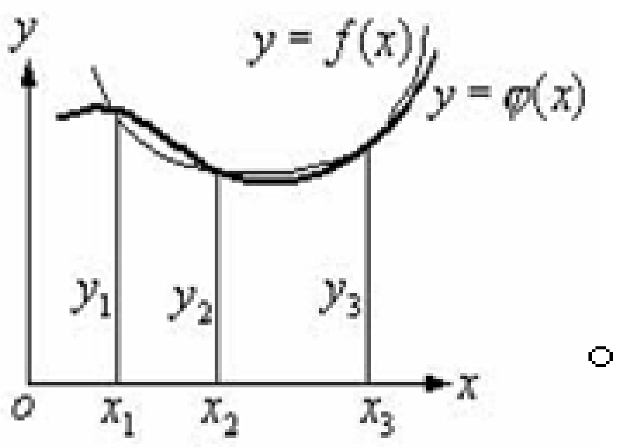




插值



(a)



(b)

1）线性插值：$\phi(t)=A_1t+A_0$, ， $t\in[0,1]$
p0和p1是已知线段的端点, $\phi(t)=(1-t)P0+tP1=(P1-P0)\,t+P0$

2)抛物线插值（二次插值）： $\phi(t)=A_2t^2+A_1t+A_0$， $t\in[0,1]$
已知f(t)在三个互异点（型值点）有三个值(t1, f(t1)), (t2,f(t2)), (t3,f(t3))，代入$\phi(t)$可求出系数Ai

逼近

- **曲线曲面的逼近**：选择一个次数较低的函数,在某种意义上最佳逼近给定点(控制点). 该函数不必通过所有控制点。

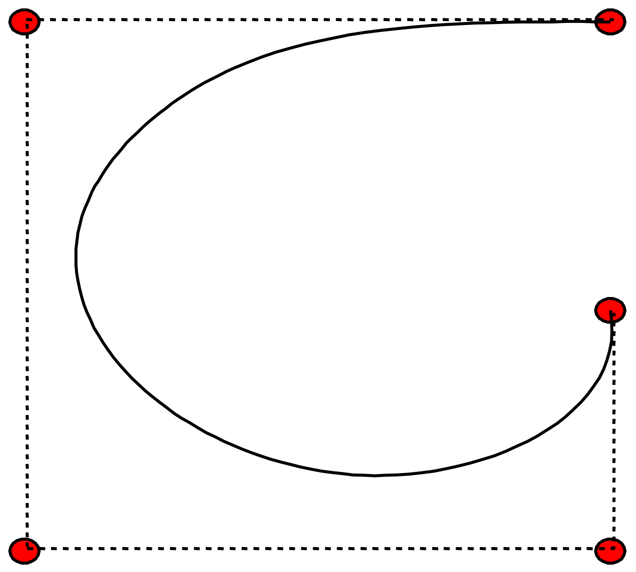


图　逼近的拟合曲线

三次参数曲线 Cubic Parametric Curves

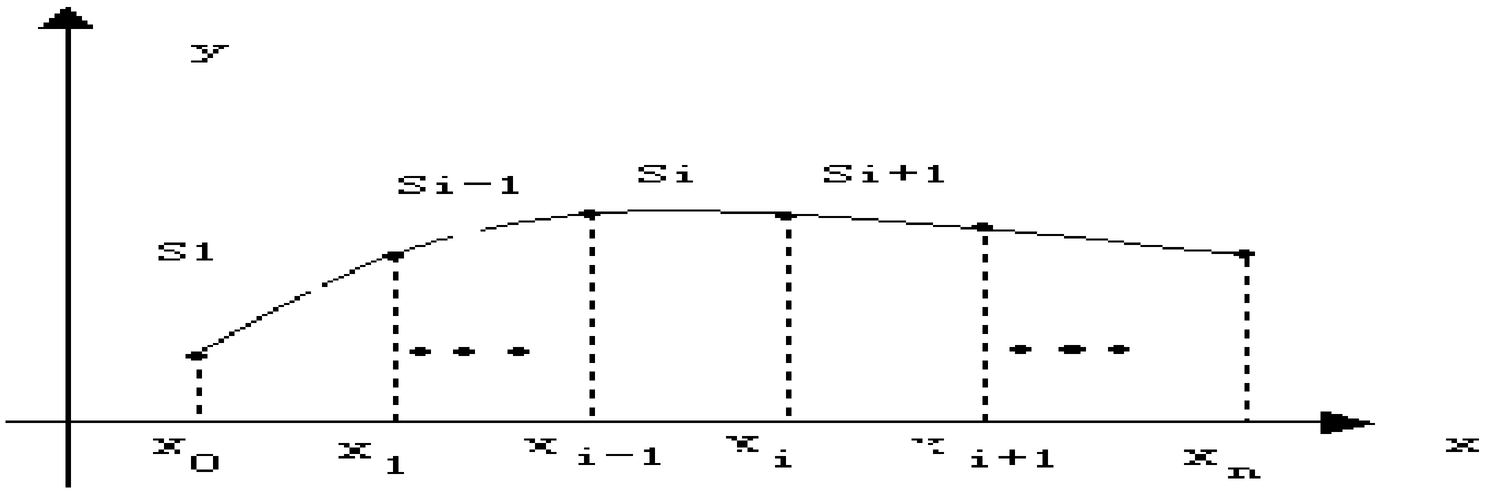
- PC曲线

form： $P(t)= A_3t^3+A_2t^2+A_1t+A_0 \quad t\in[0,1]$

拟合曲线的次数过高：过拟合
拟合曲线的次数太低：欠拟合

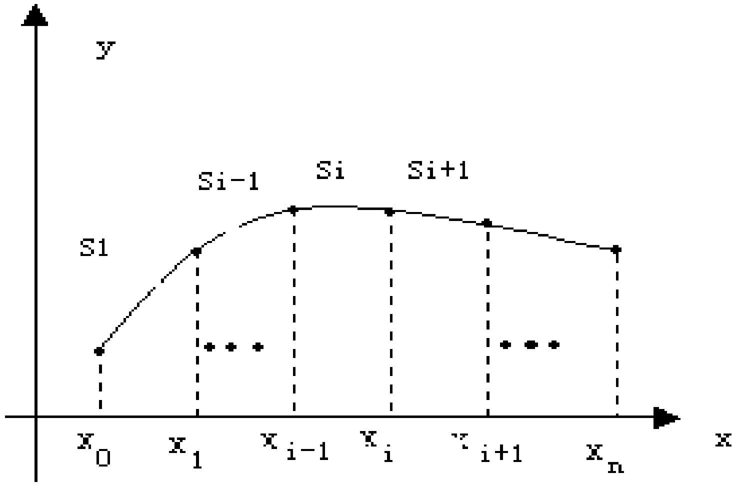
插值法构造曲线

- n+1个型值点，坐标为（x_k, y_k, z_k)，　k=0,1,2,…,n
- 定义多段三次插值样条曲线$S_i(t)$：$S_i(t)= A_{i3}t_i^3+A_{i2}t_i^2+A_{i1}t_i+A_{i0}$,
- 每一段都需要确定**4个系数值**（矢量）**A3, A2, A1, A0**（矢量）
- 通过在两条曲线段的“***交点***”处设置足够的边界条件来解决该问题，求所有样条曲线的系数值



1)Natural Cubic Spline自然三次样条（插值法）

在所有曲线段的连接处均具有零阶、一阶和二阶导数的连续性，曲线有C^2连续性。



N+1个型值点，N段样条曲线段，每段是M次多项式：

1) **总共待求系数　(m＋1)*n=4n　(m=3)**

2) **为解出系数，找方程：**

(1) S(x)经过这n+1个有序点，可建立n+1个方程

(2) 根据连续性条件，在整个区间[a,b]上，Si(t)有0、1、2阶连续性，可建立(n-1)*3个方程.

3) **添加2个补齐条件**

$$4n-((n+1)+(n-1)*3)=2$$

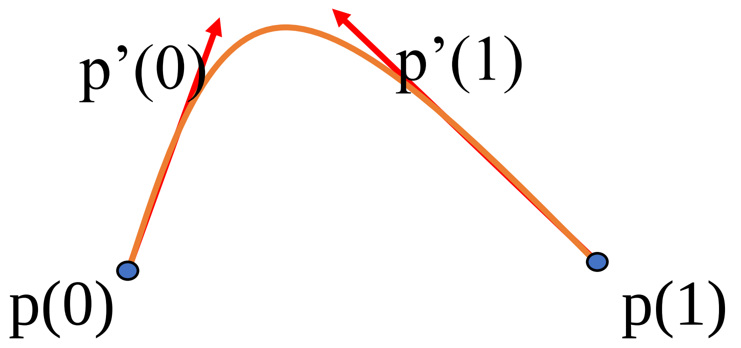
- 方法之一是在P_0和P_n处设其二阶导数为0。
- 方法之二是增加二个“隐含”控制点，各自位于控制点序列的两端。即增加P_{-1}和P_{n+1}控制点，这时，所有的控制点都成为了内点 。

缺点：“全局控制”：如果控制点中的任何一个改动，则整个曲线都受影响。即不允许“局部控制”

2)Hermite Interpolation（插值法）

$$P(t)= TA=A_3t^3+A_2t^2+A_1t+A_0$$

$$P'(t)=2tA_3+tA_2+A_1$$



已知：矢量P_k、P_{k+1}、R_k和R_{k+1}，

$$p(0)=P_k\,,\; p(1)=P_{k+1}$$
$$p'(0)=R_k\,,\; p'(1)=R_{k+1}$$

$P_k=P(0)=A_0$

$P_{k+1}=P(1)=A_3+A_2+A_1+A_0$

$R_k=P'(0)=A_1$

$R_{k+1}=P'(1)=2A_3+A_2+A_1$

$A3=2P_k-2P_{k+1}+R_k+R_{k+1}$

$A2=-3P_k+3P_{k+1}-2R_k-R_{k+1}$

$A1=R_{k+1}$

$A0=P_k$

2)Hermite Interpolation（续）

- **解得系数**

$$
\mathbf{A}= \begin{matrix} A3 \\ A2 \\ A1 \\ A0 \end{matrix} = \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} P_k \\ P_{k+1} \\ R_k \\ R_{k+1} \end{bmatrix} = M_h \times G_h
$$

M_h称为Hermite矩阵

$$
P(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} P_k \\ P_{k+1} \\ R_k \\ R_{k+1} \end{bmatrix}
$$

代数形式 $p(t)=T\bullet A$ **A:代数系数**

几何形式 $p(t)=H(t)\bullet G_h$ **Gh：几何系数**

H(t)是**Hermite基函数**是t的函数，有四个分量。

$$p(t)=P_kH_0(t)+P_{k+1}H_1(t)+R_kH_2(t)+R_{k+1}H_3(t)$$

$$H(t)=\begin{bmatrix} H_0(t) \\ H_1(t) \\ H_2(t) \\ H_3(t) \end{bmatrix}=\begin{bmatrix} 2t^3-3t^2+1 \\ -2t^3+3t^2 \\ t^3-2t^2+t \\ t^3-t^2 \end{bmatrix} \qquad G_h=\begin{bmatrix} P_k \\ P_{k+1} \\ R_k \\ R_{k+1} \end{bmatrix}$$

$$p(t) = P_k H_0(t) + P_{k+1} H_1(t) + R_k H_2(t) + R_{k+1} H_3(t)$$

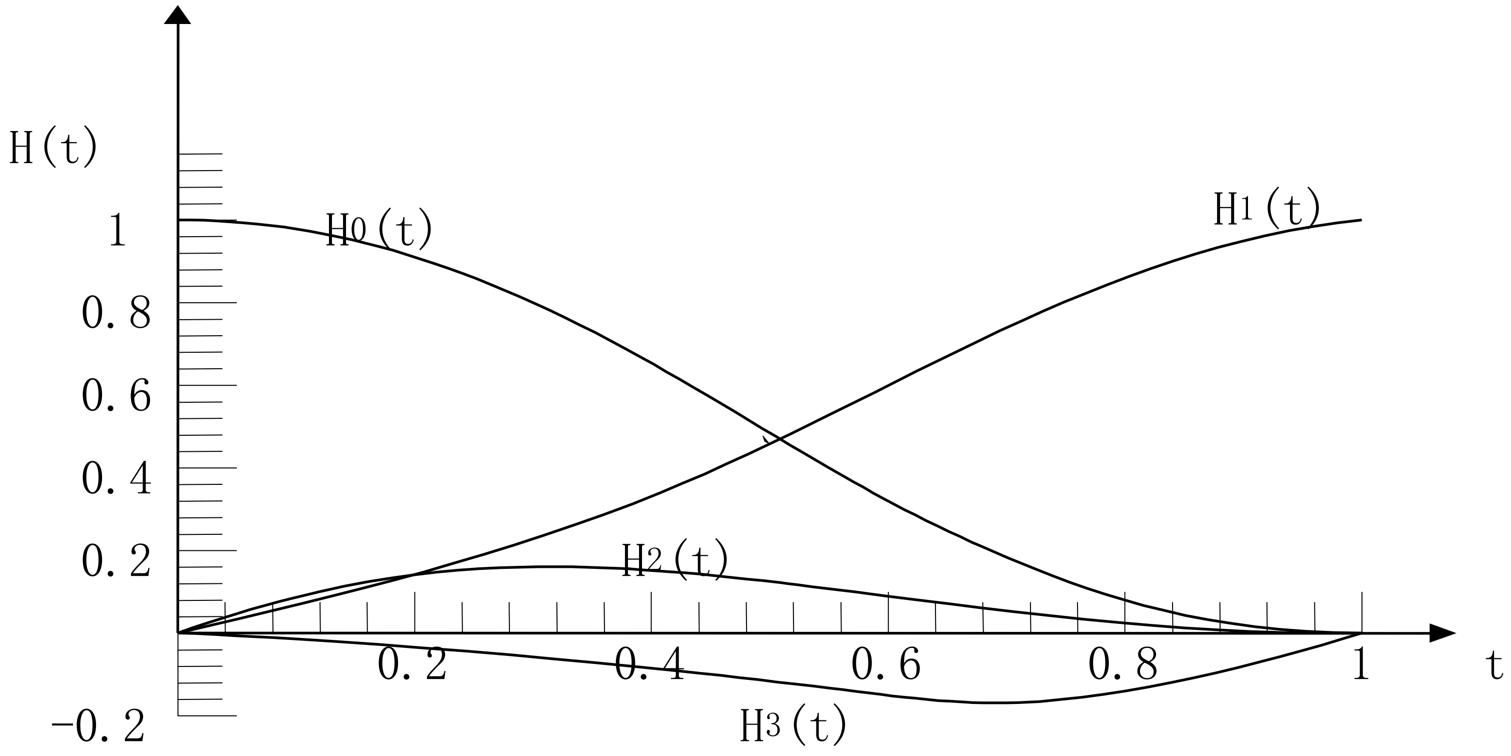
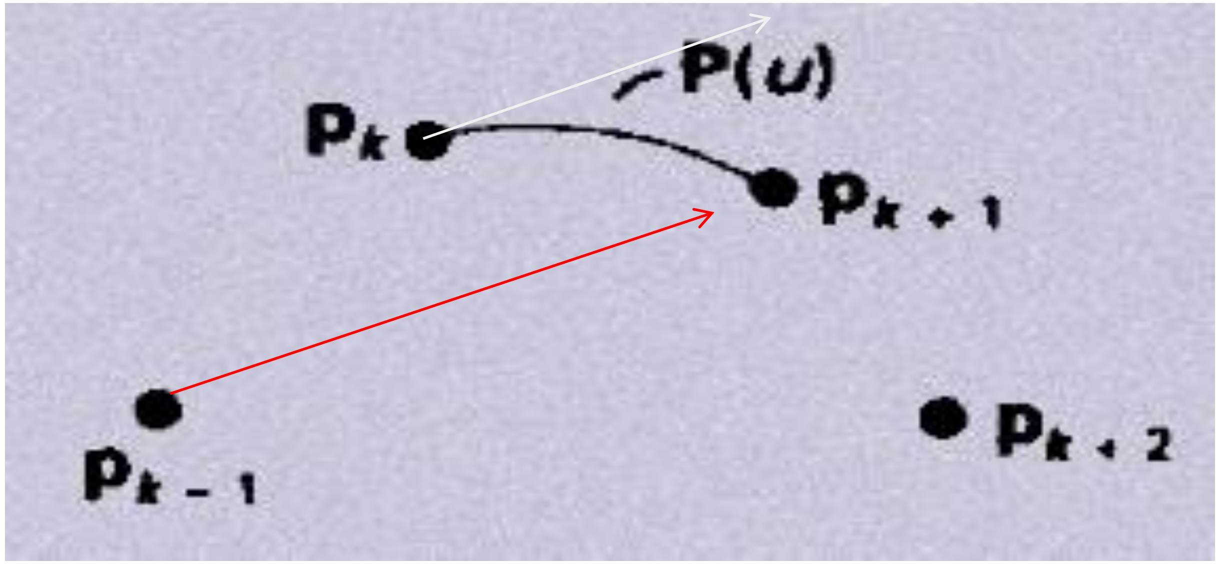


图　Hermite blend function（基函数/调和函数）

3.Cardinal Spline

- 在Cardinal样条中，控制点处斜率值可由两个相邻控制点坐标计算



思考：P(u)的端点切导

$P'(0)=(P_{k+1}-P_{k-1})*1/2$

$P'(1)=(P_{k+2}-P_{k})*1/2$

- Cardinal样条段P(u)的边界条件

  $P(0)=P_k$

  $P(1)=P_{k+1}$

  $P'(0)=(1-t)(P_{k+1}-P_{k-1})/2$

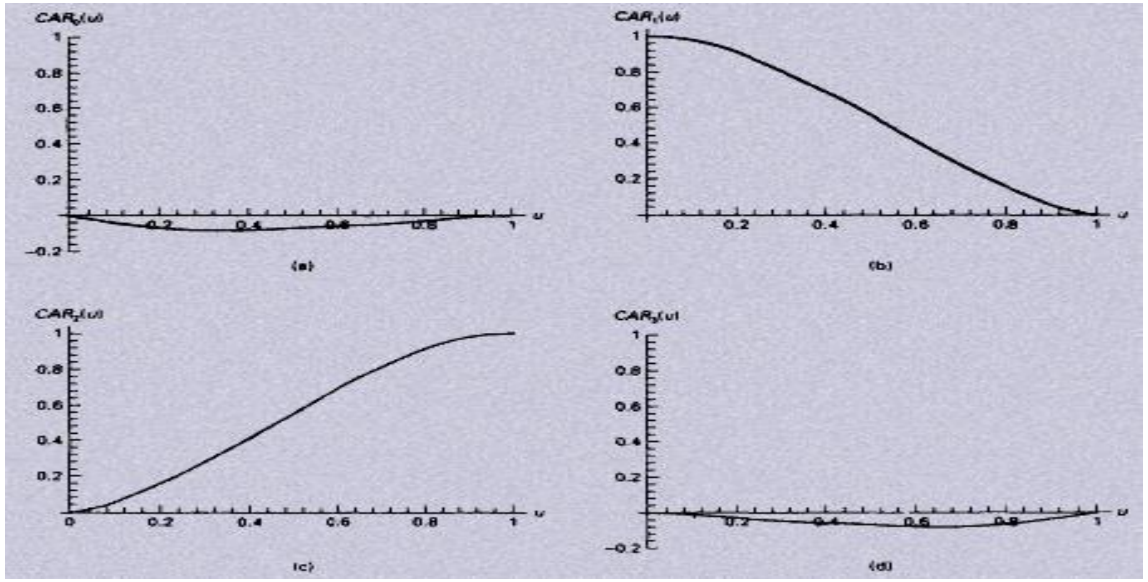
  $P'(1)=(1-t)(P_{k+2}-P_{k})/2$　　//t是一个可调参数，用来控制端点切导大小

- 用类似Hermite样条中的方法求解出代数系数后，得Cardinal曲线表示

$$p(t)=\begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix}\begin{bmatrix} -s & 2-s & s-2 & s \\ 2 & (s-3)s & 3-2s & -s \\ -s & 0 & s & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}\begin{bmatrix} P_{k-1} \\ P_k \\ P_{k+1} \\ P_{k+2} \end{bmatrix}, \text{t} \in [0,1]$$

$$= T \cdot M_{be} \cdot G_{be}$$

- *s=(1-t)/2，　可调参数的简记，t=0,s=1/2*



当t=0时，Cardinal混合函数

$P(t)=P_{k-1}CAR_0(t)+P_kCAR_1(t)+P_{k+1}CAR_2(t)+P_{k+2}CAR_3(t)$

$CAR_0(t)=-st^3+2st^2-st$

$CAR_1(t)=(2-s)t^3+(s-3)st^2+1$

$CAR_2(t)=(s-2)t^3+(3-2s)t^2+st$

$CAR_3(t)=st^3-st^2$

插值样条曲线表示小结

构造方法：

根据几何条件G-(型值点)，用某些端点条件，用插值方法确定系数，得到一组基函数H(t)来构造样条曲线段。

曲线是一组基函数的线性组合（几何表示）

$$p(t) = P_0 H_0(t) + P_1 H_1(t) + P_2 H_2(t) + P_3 H_3(t)$$

4.逼近曲线~Bezier曲线曲面

- 1962法国雷诺汽车公司P.E.Bezier构造了一种以**逼近**为基础的参数曲线。
- 完成一种曲线和曲面的设计系统UNISURF,并在1972 年在该公司应用。

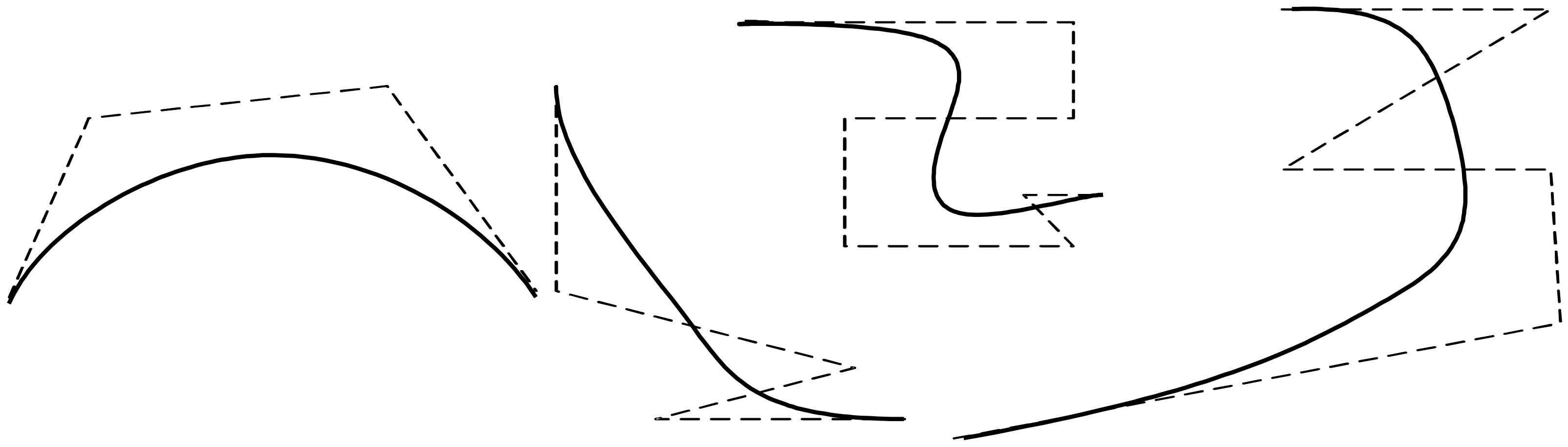
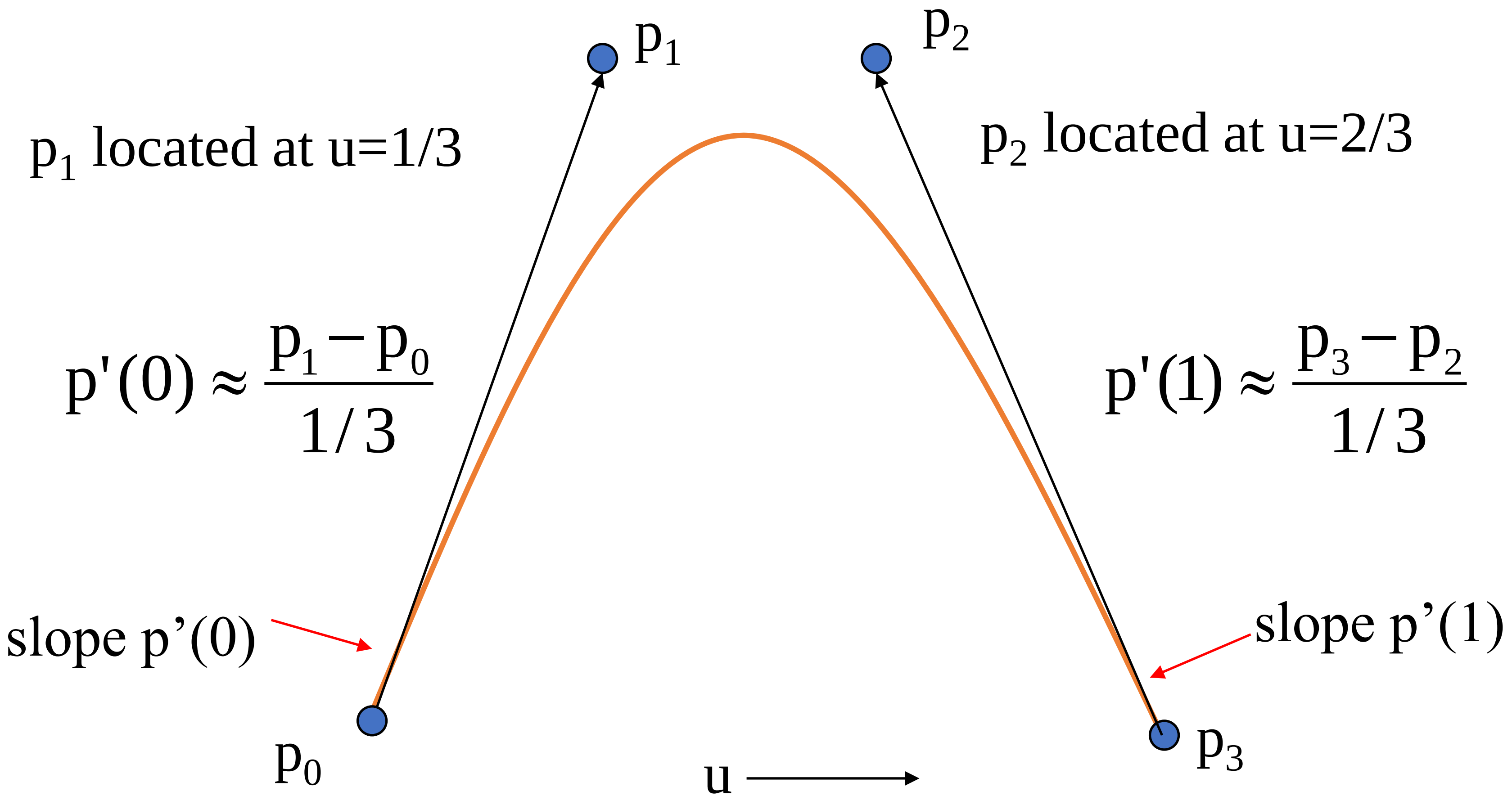


图　Bezier曲线

Bezier's Idea

- In graphics and CAD, we do not usually have derivative data（**通常没有导数数据**）
- Bezier suggested using the same 4 data points as with the cubic interpolating curve to approximate the derivatives in the Hermite form (Approximating Derivatives（**近似导数**）)



Bezier曲线定义：

$$p(t) = \sum_{k=0}^{n} P_k BEN_{k,n}(t) \qquad t \in [0, 1]$$

Bernstein基函数具有如下形式：

$$BEN_{k,n}(t) = \frac{n!}{k!(n-k)!} t^k (1-t)^{n-k} = C_n^k t^k (1-t)^{n-k}$$

$$k = 0, 1, \cdots, n$$

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

注意：当k=0，t=0时，t^k=1，k!=1。即0^0=1,0!=1

三次Bezier曲线

$$p(t) = \sum_{k=0}^{3} P_k BEN_{k,3}(t)$$

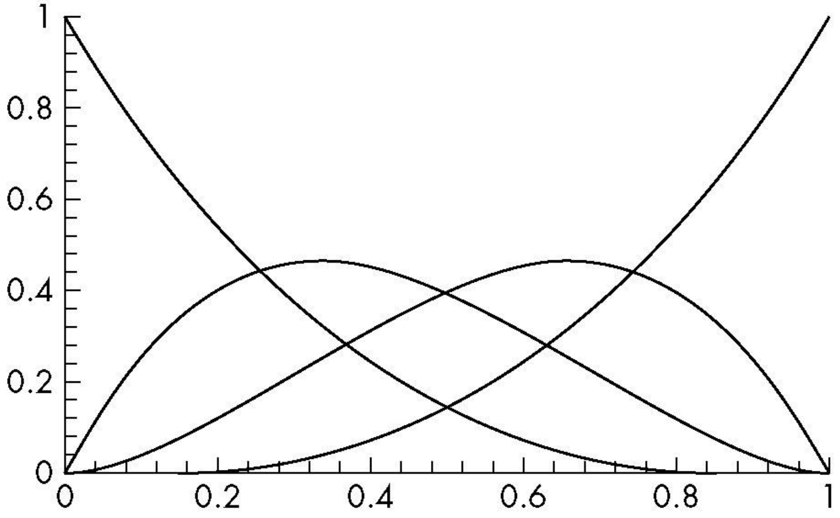
$$= (1-t)^3 P_0 + 3t(1-t)^2 P_1 + 3t^2(1-t)P_2 + t^3 P_3 \qquad t \in [0,1]$$

$$p(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{bmatrix} \qquad t \in [0,1]$$

$$= T \cdot M_{be} \cdot G_{be}$$

三次Bezier曲线的混合函数

$$\mathbf{b}(u) = \begin{bmatrix} (1-u)^3 \\ 3u(1-u)^2 \\ 3u^2(1-u) \\ u^3 \end{bmatrix}$$



Note that all zeros are at 0 and 1 which forces the functions to be smooth over (0,1)

Bezier曲线的性质

1．两端点是控制点

$$p(0)=\sum_{k=0}^{n}P_k BEN_{k,n}(0)$$

$$=P_0 BEN_{0,n}(0)+P_1 BEN_{1,n}(0)+\cdots+P_n BEN_{n,n}(0)$$

$$=P_0$$

$$p(1)=\sum_{k=0}^{n}P_k BEN_{k,n}(1)$$

$$=P_0 BEN_{0,n}(1)+P_1 BEN_{1,n}(1)+\cdots+P_n BEN_{n,n}(1)$$

$$=P_n$$

Bezier曲线的性质

2．**端点一阶导数**由控制点计算得到

$$p'(t) = n\sum_{k=0}^{n} P_k(BEN_{k-1,n-1}(t) - BEN_{k,n-1}(t))$$

$$= n((P_1 - P_0)BEN_{0,n-1}(t) + (P_2 - P_1)BEN_{1,n-1}(t)$$

$$+ \cdots + (P_n - P_{n-1})BEN_{n-1,n-1}(t))$$

$$= n\sum_{k=1}^{n}(P_k - P_{k-1})BEN_{k-1,n-1}(t)$$

$$p'(0) = n(P_1 - P_0)$$

$$p'(1) = n(P_n - P_{n-1})$$

三次bezier端点导数：

$$p'(0) = 3(P_1 - P_0)$$

$$p'(1) = 3(P_3 - P_2)$$

Bezier曲线的性质

3．Bezier曲线段端点的**二阶导数易于计算**

$$p''(0) = n(n-1)((P_2 - P_1) - (P_1 - P_0))$$

$$p''(1) = n(n-1)((P_{n-2} - P_{n-1}) - (P_{n-1} - P_n))$$

三次Bezier曲线段端点的二阶导数：

$$p''(0) = 6(P_0 - 2P_1 + P_2)$$

$$p''(1) = 6(P_1 - 2P_2 + P_3)$$

Bezier曲线的性质

4．对称性：$B_{i,n}(t)=B_{n-i,n}(1-t)$，即基函数具有对称性。

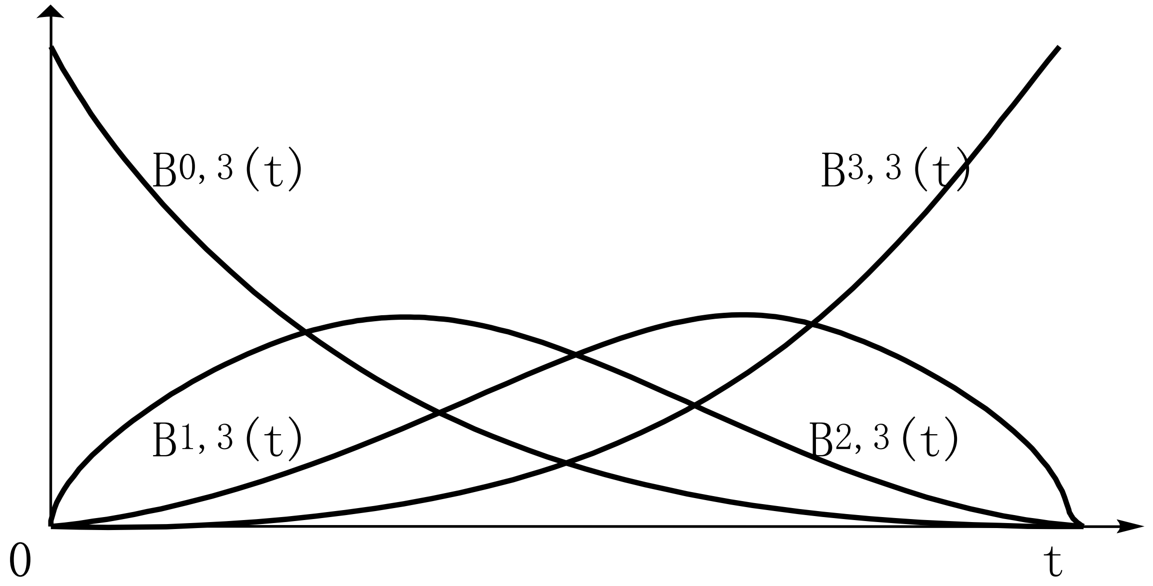


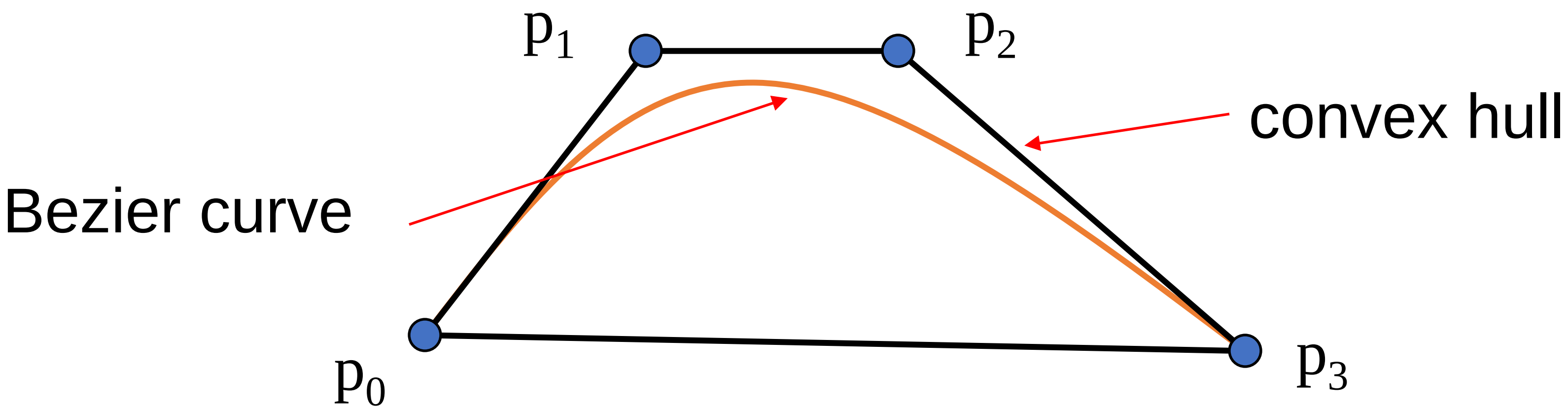
图　三次Bezier曲线四个Bezier基函数(调和函数)

Bezier曲线的性质

5．凸包性

$$BEN_{k,n}(t)=\frac{n!}{k!(n-k)!}t^k(1-t)^{n-k}\geq 0$$

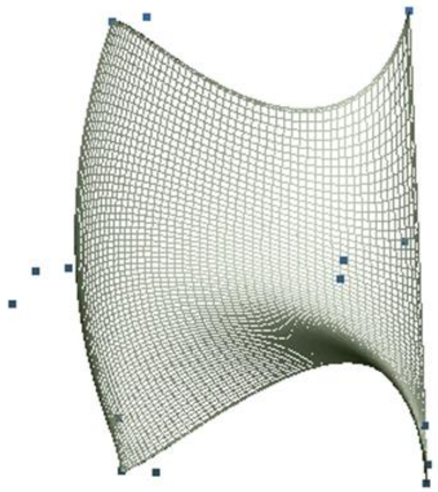
$$\sum_{k=0}^{n}BEN_{k,n}(t)=\sum_{k=0}^{n}\frac{n!}{k!(n-k)!}t^k(1-t)^{n-k}=((1-t)+t)^n\equiv 1$$



Bezier曲面

$$p(u,v)=\sum_{i=0}^{m}\sum_{j=0}^{n}P_{i,j}BEN_{i,m}(u)BEN_{j,n}(v)$$

$$(\mathrm{u},\mathrm{v})\in[0,1]\times[0,1]$$



BENi,m(t)与BENj,n(v)是**Bernstein基函数**:

$$BEN_{i,m}(u)=C_m^i\cdot u^i\cdot(1-u)^{m-i}$$

$$BEN_{j,n}(v)=C_n^j\cdot v^j\cdot(1-v)^{n-j}$$

Bezier**曲面**-双线性Bezier曲面($m=n=1$)

$$p(u,v)=\sum_{i=0}^{1}\sum_{j=0}^{1}P_{i,j}BEN_{i,1}(u)BEN_{j,1}(v)$$

$$(\mathrm{u},\mathrm{v})\in[0,1]\times[0,1]$$

$$p(u,v)=(1-u)(1-v)P_{0,0}+(1-u)vP_{0,1}$$

$$+u(1-v)P_{1,0}+uvP_{1,1}$$

Bezier**曲面**-双二次Bezier曲面(*m*=*n*=2)

$$p(u,v)=\sum_{i=0}^{2}\sum_{j=0}^{2}P_{i,j}BEN_{i,2}(u)BEN_{j,2}(v)$$

$$(\mathrm{u},\mathrm{v})\in[0,1]\times[0,1]$$

Bezier**曲面**-双三次Bezier曲面($m=n=3$)

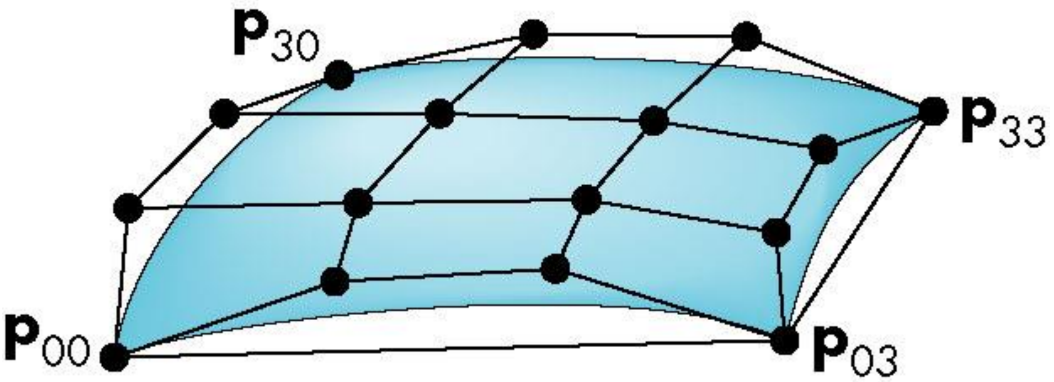
$$p(u,v) = \sum_{i=0}^{3}\sum_{j=0}^{3} P_{i,j} BEN_{i,3}(u) BEN_{j,3}(v)$$

$$(\mathrm{u},\mathrm{v}) \in [0,1]\times[0,1]$$

Using same data array $\mathbf{P}=[p_{ij}]$ as with interpolating form

$$p(u,v) = \sum_{i=0}^{3}\sum_{j=0}^{3} b_i(u)\, b_j(v)\, p_{ij} = u^T \mathbf{M}_B \mathbf{P} \mathbf{M}_B^T v$$

Patch lies in convex hull



$$p(u,v) = UM_{be}PM_{be}^{T}V^{T}$$

$$U = \begin{bmatrix} u^3 & u^2 & u & 1 \end{bmatrix}$$

$$V = \begin{bmatrix} v^3 & v^2 & v & 1 \end{bmatrix}$$

$$M_{be} = \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

$$P = \begin{bmatrix} P_{0,0} & P_{0,1} & P_{0,2} & P_{0,3} \\ P_{1,0} & P_{1,1} & P_{1,2} & P_{1,3} \\ P_{2,0} & P_{2,1} & P_{2,2} & P_{2,3} \\ p_{3,0} & P_{3,1} & P_{3,2} & P_{3,3} \end{bmatrix}$$

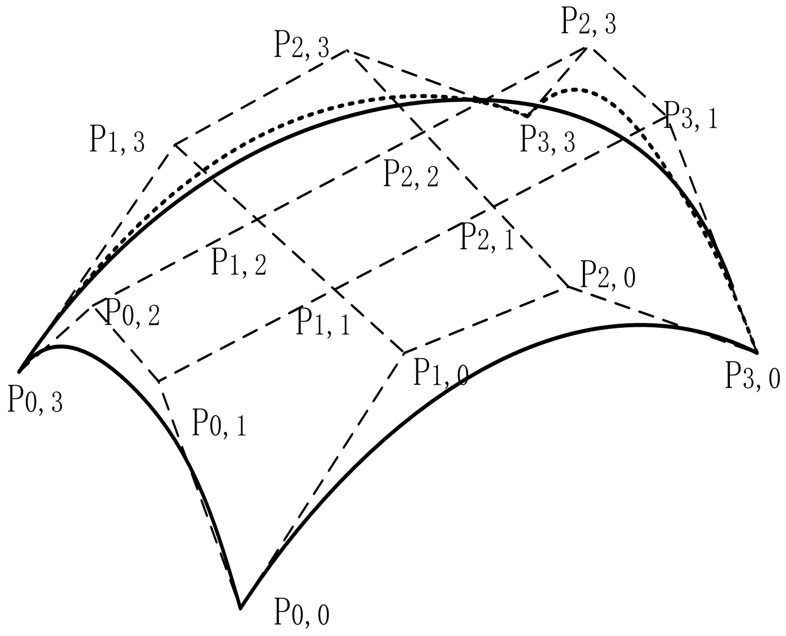


图 双三次Bezier曲面及其控制网格

Bezier曲线的拼接

- 两段三次Bezier曲线之间间得到G^1连续性，则有：

$$p'(1) = 3(P_3 - P_2)$$

$$Q'(0) = 3(Q_1 - Q_0) \qquad Q_1 - Q_0 = \alpha \cdot (P_3 - P_2)$$

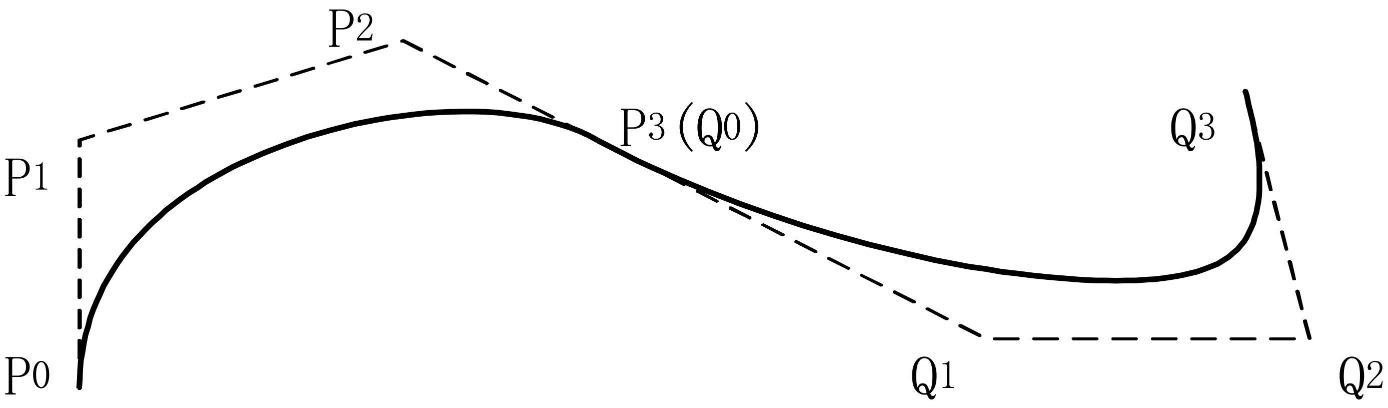


图 两段三次Bezier曲线的连接

Bezier曲线的拼接

- 两段三次Bezier曲线间得到G^2连续性：

$$p_2''(0) = \beta \cdot p_1''(1)$$

$$(Q_0 - 2Q_1 + Q_2) = \beta \cdot (P_1 - 2P_2 + P_3)$$

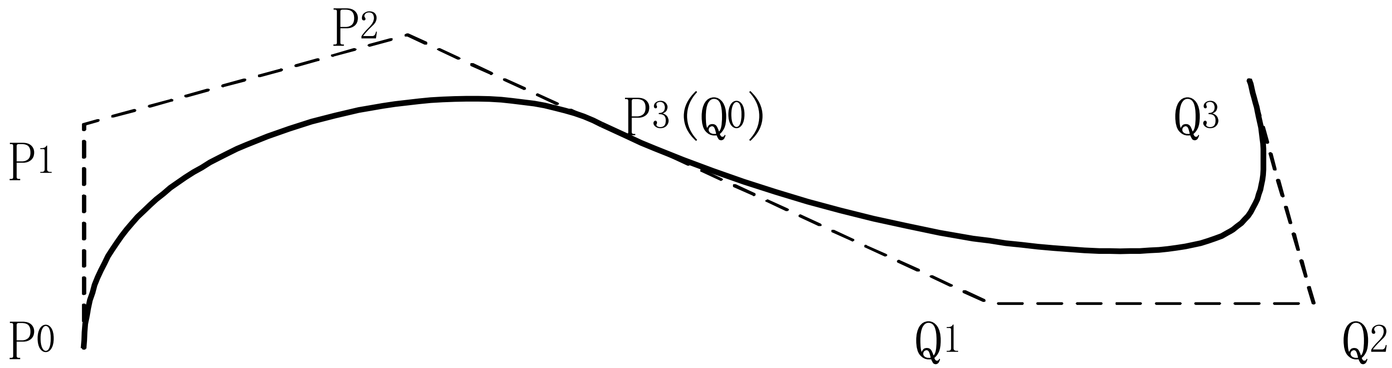


图 两段三次Bezier曲线的连接

PC曲线总结

$$p(t) = P_0 H_0(t) + P_1 H_1(t) + P_2 H_2(t) + P_3 H_3(t)$$

$$p(t) = \sum_{k=0}^{n} H_{k,n}(t) P_k \qquad t \in [0,1]$$

- PC曲线：三次多项式参数表示
- 构造：是给定一组控制点或型值点，采用插值或逼近的方法构造一组基函数，并以基函数作权和控制点作几何系数，二者相乘得到。

内容提要

- Curves and Surfaces Representation
- Curves and Surfaces Design
- Curves and Surfaces Render

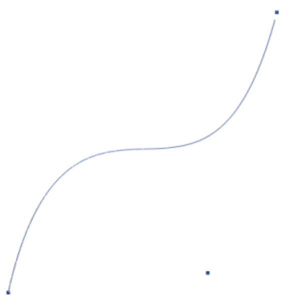
样条曲线绘制方法

- to render a polynomial curve is to evaluate the polynomial at many points and form an **approximating polyline**

曲线:多个参数点上求多项式值，并形成近似折线

- For surfaces we can form an **approximating mesh** of triangles or quadrilaterals

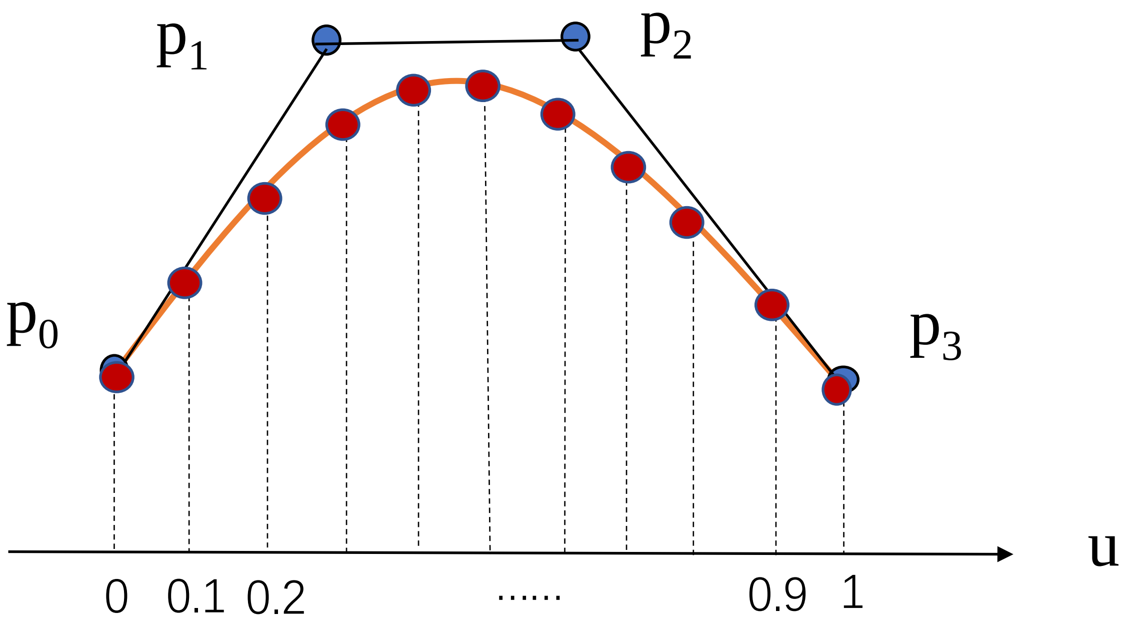
曲面:两组参数点上求值,形成三角形或四边形的近似网格



1）直接求值

思路：通过等间距的一组参数值计算曲线上的点，然后连接这些点，使用折线来近似表示曲线。

特点：计算量大。



$$p(t)=\sum_{k=0}^{3}P_k BEN_{k,3}(t)$$
$$=(1-t)^3P_0+3t(1-t)^2P_1+3t^2(1-t)P_2+t^3P_3 \qquad t\in[0,1]$$

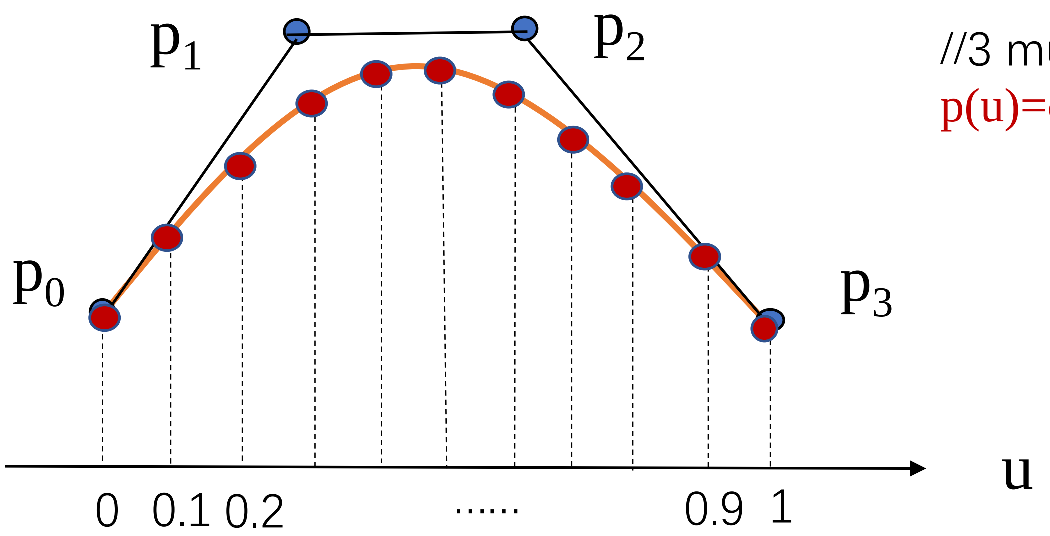
如：等间距划分参数区间[0,1] t=[0, 0.1，0.2，0.3,0.4,0.5,0.6,0.7,0.8,0.9,1]

对每个参数取值t，计算基函数和控制点P的点乘得曲线上的点（红点）

2）多项式求值/ Horner’s method

思路：通过等间距的一组参数值计算曲线上的点，然后连接这些点，使用折线来近似表示曲线。

特点：计算有优化，但是只适合均匀网络，容易累积数值误差



//3 multiplications/evaluation for cubic

$p(u)=c_0+u(c_1+u(c_2+uc_3))$

如：等间距划分参数区间[0,1] t=[0, 0.1，0.2，0.3,0.4,0.5,0.6,0.7,0.8,0.9,1]

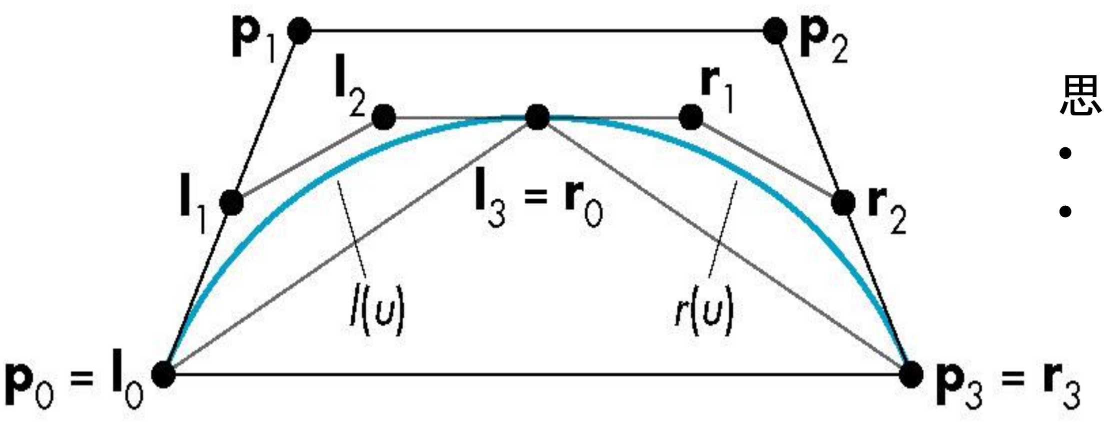
对每个参数取值t，计算基函数和控制点P的点乘得曲线上的点（红点）

3) 递归细分方法De Casteljau's algorithm

Splitting a Cubic Bezier

use the convex hull（凸包） property of Bezier curves to obtain **an efficient recursive method**

- does not require any function evaluations（不需要函数求值 /计算多项式）
- Uses only the values at the control points（只需计算控制点的值）



思想：

- 不断将曲线一分为二，重新计算所分更短两段的控制点，
- 当小段的控制点凸包“充分靠近”曲线时，就用首末两点线段表示该曲线段。

算法描述

- 初始，将p(u)的凸包控制点(P0,P1,P2,P3)分成两组控制点（l0,l1,l2,l3）,(r0,r1,r2,r3)

➢ 判断左边组控制点的扁平性（l_1l_2到l_0l_3的距离）或l_0l_3足够小）

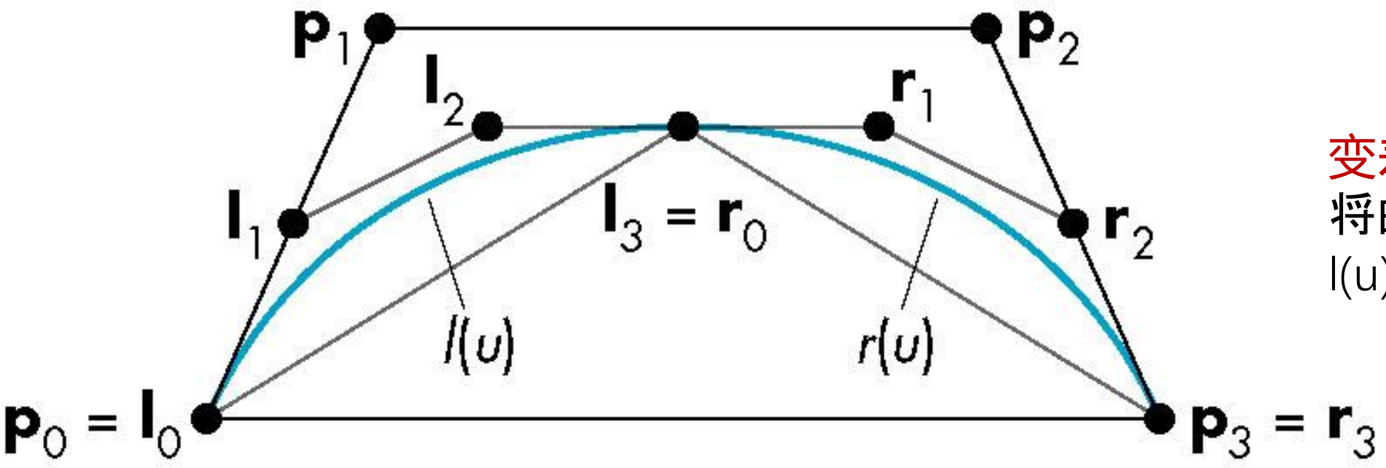
如果距离足够小，则可用线段(l_0l_3)代替该段曲线；

否则，可将控制点组分成两部分并且继续判断两个新凸包的扁平性（递归调用）

➢ 判断右边边组控制点的扁平性（ r_1r_2到r_0r_3的距离） 或r_0r_3足够小

如果距离足够小，则可用线段(r_0r_3)代替该段曲线；

否则，可将控制点组分成两部分并且继续判断两个新凸包的扁平性（递归调用）

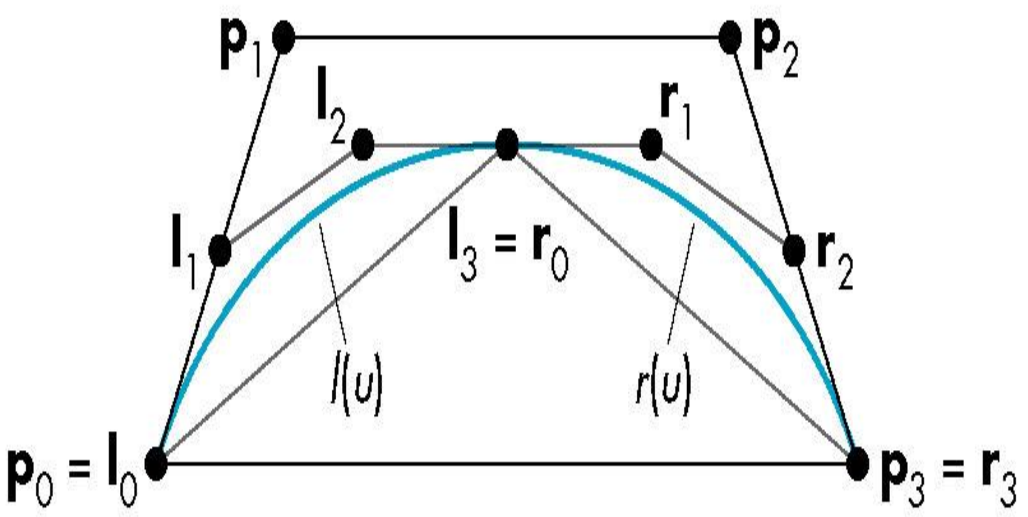


变差缩减性the *variation diminishing property*
将曲线P(u)分割成两个独立的多项式l(u),r(u),
l(u)和r(u)的凸包都将位于P（u）的凸包内。

Equations 计算控制点

Start with Bezier equations $p(u)=\mathbf{u}^T M_B \mathbf{p}$
$l(u)$ must interpolate $p(0)$ and $p(1/2)$

$l(0) = l_0 = p_0$
$l(1) = l_3 = p(1/2) = 1/8(\, p_0 + 3\, p_1 + 3\, p_2 + p_3\,)$



Matching slopes, taking into account that $l(u)$ and $r(u)$ only go over half the distance as $p(u)$

$l'(0) = 3(l_1 - l_0) = p'(0) = 3/2(p_1 - p_0\,)$
$l'(1) = 3(l_3 - l_2) = p'(1/2) = 3/8(-\, p_0 - p_1 + p_2 + p_3)$

对于$r(u)$同理得到四个计算方程。

Efficient Form有效计算形式

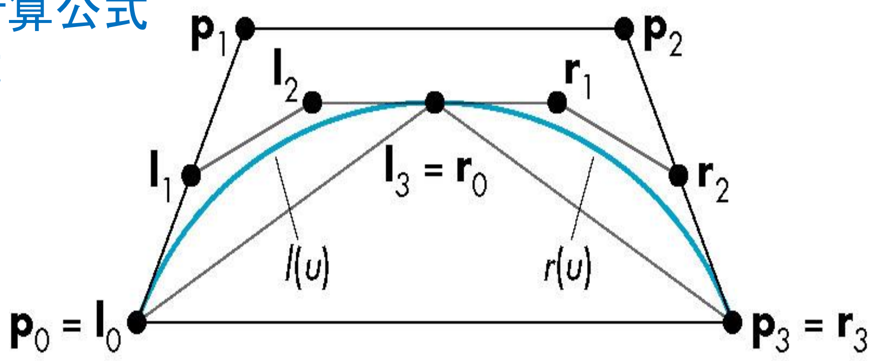
上页四个方程联立求解可得到控制点(l0,l1,l2,l3)的计算公式
同理对称得到r(u)的控制点值(r0,r1,r2,r3)的计算公式

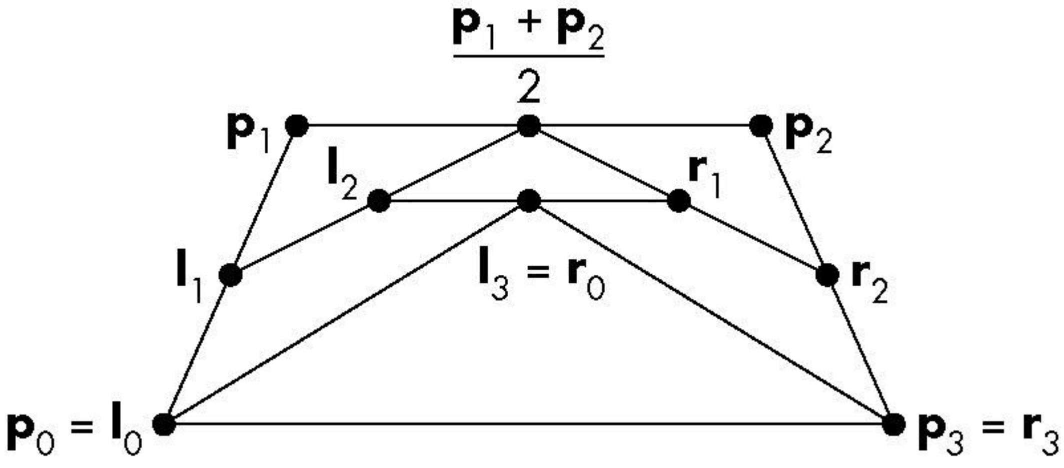
$$l_0 = p_0$$
$$r_3 = p_3$$

$$l_1 = ½(p_0 + p_1)$$
$$r_1 = ½(p_2 + p_3)$$

$$l_2 = ½(l_1 + ½(p_1 + p_2))$$
$$r_1 = ½(r_2 + ½(p_1 + p_2))$$

$$l_3 = r_0 = ½(l_2 + r_1)$$





Requires only shifts and adds!

总结递归样条细分法

- Recursive spline-subdivision 高效的算法
- 作用1：重复二分细分曲线，直到控制点逼近曲线路径，然后将控制点坐标作为曲线位置而绘制出整个曲线。
- 作用2：也位绘制曲线造型生成了更多的控制点-附加的控制点，这样可以使用这些附加的控制点，对曲线的某些小段进行精细的调整。