



计算机图形学基础

胡事民

清华大学计算机科学与技术系



B ézier曲线及曲面

- 参数曲线及曲面: 一些概念
- B ézier曲线: 概念及性质
- B ézier曲面: 矩形域和三角域
- 矩形域与三角域B ézier曲面的转换



引入

- 在计算机图形学的应用中，经常需要设计三维模型或三维场景，然后通过渲染方程产生具有真实感的图像
- 因此有必要引入几何造型（ **Geometric Modeling** ）技术来完成三维建模的任务



几何造型的历史

- 曲面造型（Surface Modeling）
 - 法国雷诺汽车公司工程师**Pierre Bézier**于1962年提出了一种新的曲线表达方式，并最终于1972年开发了用于汽车表面设计的**UNISURF**系统





- **Pierre Etienne B ézier**于1910年出生在法国巴黎，其父亲和爷爷都是工程师
- 他于1930年取得了机械工程学士学位，并于1931年获得了电子工程的第二学位，46年后的1977年，他获得了巴黎大学数学博士学位
- 1933年，时年23岁的**B ézier**进入雷诺公司并为其工作了42年
- **B ézier**的学术生涯是从1968-1979年，在此期间他担任国立巴黎工艺美术学院生产工程系教授



- 实体造型（Solid Modeling）：
 - 1973年，剑桥大学的Ian Braid开发了用于设计工程部件的实体造型系统
 - 该工作在他的论文“**Designing with volumes**”中予以介绍，并在**BUILD-1**系统中演示



- **BUILD-1系统的三位发明人获2008年Bezier奖。**

Solid Modeling Association

Join SMA Home Contact Us

SPM Conferences Awards SMA Forums Events & News Relevant Links Credits

Bézier Award ■ I. Braid, A. Grayer and C. Lang

Ian Braid, Alan Grayer and Charles Lang, the 2008 Pierre Bézier Award Recipients

This group of three people made many fundamental contributions to practical solid modelling, and their work has had a profound influence on today's commercial solid modelling systems. They commenced working together in the CAD Group at the Computer Laboratory, Cambridge University. The Group was set up by Charles Lang under Prof. Maurice Wilkes' direction in 1965 to undertake research on tools for building mechanical CAD/CAM systems, with an emphasis on software system components, computer graphics and computational geometry. Initial experiments in solid modelling were made in 1969. Also in 1969 Ian Braid joined the Group where, under Charles Lang's supervision, he developed the BUILD boundary representation modeller, the most advanced such system of its day. Whereas other systems used faceting to avoid the problems of calculating intersections between non-planar surfaces, the BUILD team tackled such problems head-on. Ian was awarded his PhD in 1973.

Alan Grayer joined the group in 1971 and, also under Charles' supervision, developed algorithms for the automatic machining of prismatic parts modelled in BUILD. These were machined on a model making machine, built by the Group in 1971 following an inspirational visit to Bézier at Renault in Paris. Alan was awarded his PhD in 1977. Ian then developed a completely new solid modeller, BUILD 2, which was a significant advance as it made a clear separation of geometry and topology in both its data structures and algorithms. This made it possible to implement generalised boolean operations and to systematically extend the geometric coverage and the functionality of the modeller with operations such as blending. Subsequently other PhD theses supervised by Ian and based on the BUILD modellers included Dimensions and Tolerances (Hillyard 1978), Feature Recognition (Kyprianou 1980), Automatic 2D and 3D Mesh Generation (Márdenauer 1982) and Surface Intersections (Solomon 1986). These theses were some of the earliest contributions to solid modelling and to



怎样表示曲线?

- 三种表示方式:

- 显式表示(Explicit representation): 2维空间中曲线的显式表示是以自变量来表达因变量的值, 在x-y空间中, 可以写成

$$y = f(x)$$

- 对于直线, 我们通常写成

$$y = mx + h$$



- 隐式表示(**Implicit representation**): 在二维空间中，一个隐式曲线可以通过下式表达

$$f(x, y) = 0$$

- 对于直线

$$ax + by + c = 0$$

- 对于圆

$$x^2 + y^2 - r^2 = 0$$



- 参数形式(Parametric form): 一个曲线的参数形式是通过一个自变量(参数) t 的形式来表达曲线点的每一个空间变量
- 在3维空间中，我们三个显式函数：

$$x = x(t)$$

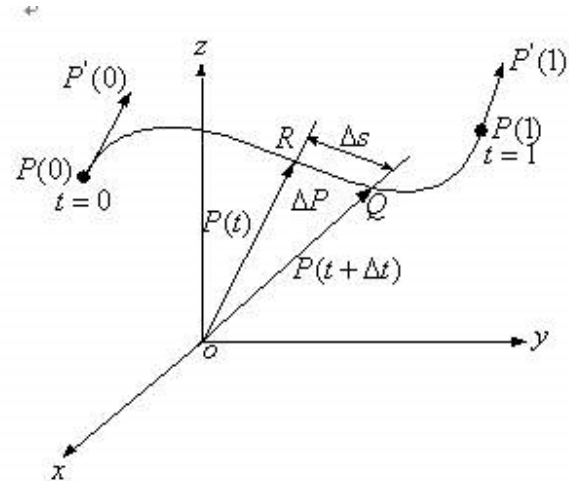
$$y = y(t)$$

$$z = z(t)$$



- 参数形式的一个优点是在不同维度的空间中具有统一的表达形式，例如只需增加关于 z 的方程，就可以从二维空间推广到三维空间
- 参数形式也便于进行可视化，它可以被看作是随着 t 的变化产生的点的轨迹 (locus of points)

$$P(t) = [x(t), y(t), z(t)]^T$$





– 我们可以将 $P(t)$ 的导数

$$\frac{dP(t)}{dt} = \begin{bmatrix} \frac{dx(t)}{dt} \\ \frac{dy(t)}{dt} \\ \frac{dz(t)}{dt} \end{bmatrix}$$

的模长视为曲线的绘制速度，该导数指向曲线的切向



参数多项式曲线

- 参数多项式曲线是指以 t 的多项式函数表达的参数形式，例如3阶多项式参数曲线具有如下形式：

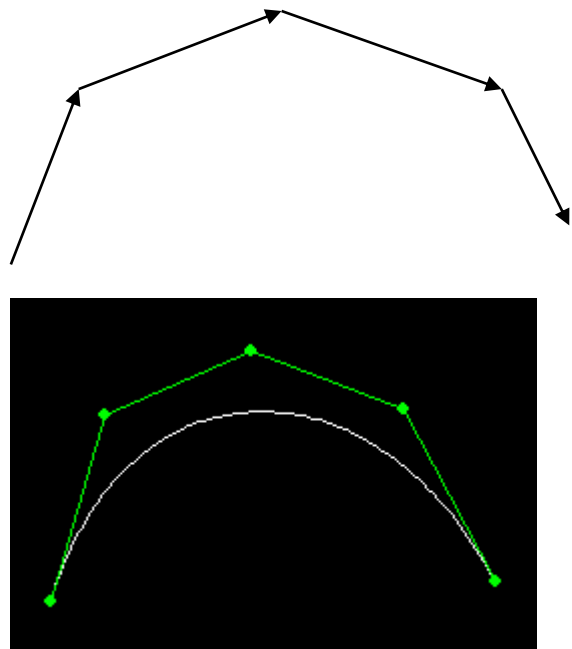
$$P(t) = at^3 + bt^2 + ct + d$$

- 它被称为Ferguson曲线，曾被用于美国早期的飞机设计中
- 但是Ferguson曲线并不直观：给定系数 a, b, c, d 之后，想象曲线的形状很困难



B ézier曲线: 概念及性质

- Pierre B ézier提出了一种通过连接向量 (connected vectors) 的加权求和来表示曲线的方法



$$V(t) = \sum_{i=0}^n f_{i,n}(t) A_i$$

$$f_{i,n}(t) = \begin{cases} 1, & i = 0, \\ \frac{(-t)^i}{(i-1)!} \frac{d^{i-1}}{dt^{i-1}} \left(\frac{(1-t)^{n-1} - 1}{t} \right) \end{cases}$$



- 曲线可以交互式的设计吗？
 - 曲线根据控制网 (连接向量) 生成
 - 我们可以通过改变连接向量来修改曲线
 - 示例: [curvesystem.exe](#)



- 然而，这样的定义是难以理解的

$$V(t) = \sum_{i=0}^n f_{i,n}(t) A_i$$
$$f_{i,n}(t) = \begin{cases} 1, & i = 0, \\ \frac{(-t)^i}{(i-1)!} \frac{d^{i-1}}{dt^{i-1}} \left(\frac{(1-t)^{n-1} - 1}{t} \right) \end{cases}$$



- 1972年, Forrest在Computer Aided Design杂志上发表了他的著名论文
 - 他指出 B ézier曲线可以借助于伯恩斯坦 (Bernstein)多项式被定义在点集上
 - 中国学者的贡献: 梁友栋, 刘鼎元, 常庚哲
 - B ézier教授于1999年去世, CAGD杂志在2001年为他出版了一个特刊(全球征集、7篇论文)



ELSEVIER

Computer Aided Geometric Design 18 (2001) 667–671

COMPUTER
AIDED
GEOMETRIC
DESIGN

www.elsevier.com/locate/comaid

Conversion between triangular and rectangular Bézier patches

Shi-Min Hu

Department of Computer Science and Technology, Tsinghua University, Beijing 100084, PR China

Received September 2000; revised May 2001

In memory of P. Bézier

Abstract

This paper presents an explicit formula that converts a triangular Bézier patch of degree n to a degenerate rectangular Bézier patch of degree $n \times n$ by reparametrization. Based on this formula, we develop a method for approximating a degenerate rectangular Bézier patch by three nondegenerate Bézier patches; more patches can be introduced by subdivision to meet a user-specified error tolerance. © 2001 Elsevier Science B.V. All rights reserved.

Keywords: Bézier surfaces; Degree elevation; Subdivision; Conversion



定义

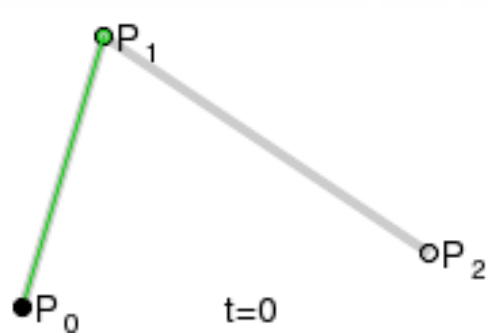
- 给定控制顶点 P_0, P_1, \dots, P_n , Bézier曲线可以被定义为:

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

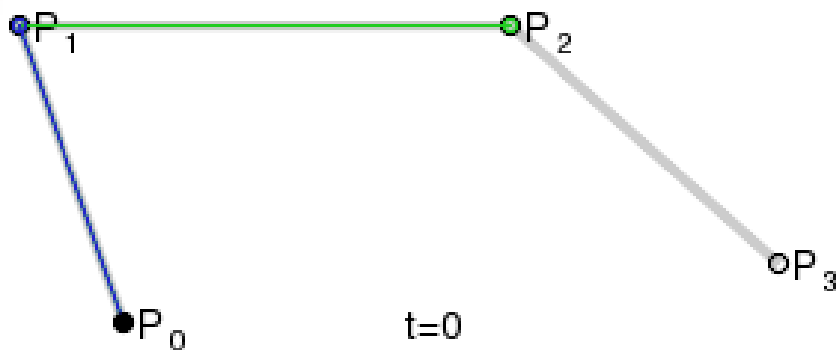
其中: $B_{i,n}(t)$ 是第 i 个 n 阶Bernstein多项式

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

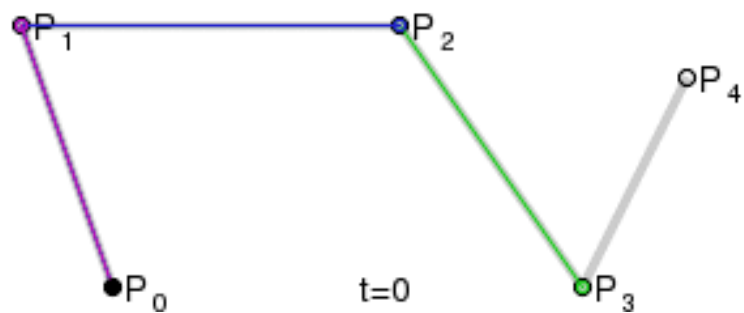
$(i = 0, 1, \dots, n)$



二阶



三阶



四阶

不同阶数Bézier曲线



伯恩斯坦(*Bernstein*)多项式的性质

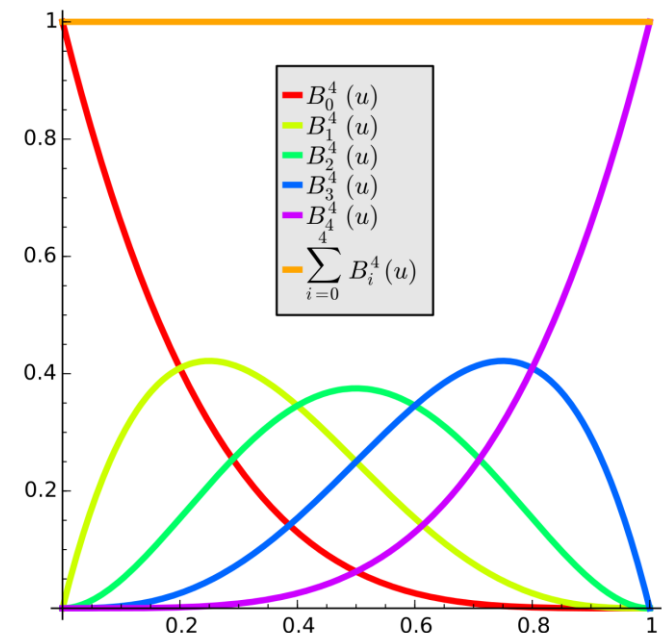
- 非负 (Non-negative)

$$B_{i,n}(t) = \begin{cases} = 0 & t = 0, 1 \\ > 0 & t \in (0, 1), i = 1, 2, \dots, n-1; \end{cases}$$

- 端点(End point)

$$B_{i,n}(0) = \begin{cases} 1 & (i = 0) \\ 0 & \textit{otherswise} \end{cases}$$

$$B_{i,n}(1) = \begin{cases} 1 & (i = n) \\ 0 & \textit{otherswise} \end{cases}$$





- 归一性(Unity)

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

– 证明: 根据二项式定理(*Binomial Theorem*), 可以得到

$$\sum_{i=0}^n B_{i,n}(t) = \sum_{i=0}^n C_n^i t^i (1-t)^{n-i} = [(1-t) + t]^n \equiv 1$$



- 对称性 (Symmetry)

$$B_{i,n}(1-t) = B_{n-i,n}(t)$$

– 证明:

$$\begin{aligned} B_{n-i,n}(t) &= C_n^{n-i} [1 - (1-t)]^{n-(n-i)} \cdot (1-t)^{n-i} \\ &= C_n^i t^i (1-t)^{n-i} = B_{i,n}(1-t) \end{aligned}$$



- 递归性 (Recursive)

$$B_{i,n}(t) = (1-t)B_{i,n-1}(t) + tB_{i-1,n-1}(t),$$
$$(i = 0, 1, \dots, n)$$

– 这个式子表明n阶*Bernstein*多项式是n-1阶*Bernstein*多项式的线性组合

$$\begin{aligned} B_{i,n}(t) &= C_n^i t^i (1-t)^{n-i} = (C_{n-1}^i + C_{n-1}^{i-1}) t^i (1-t)^{n-i} \\ &= (1-t) C_{n-1}^i t^i (1-t)^{(n-1)-i} + t C_{n-1}^{i-1} t^{i-1} (1-t)^{(n-1)-(i-1)} \\ &= (1-t) B_{i,n-1}(t) + t B_{i-1,n-1}(t) \end{aligned}$$



- 导数 (Derivation)

$$B'_{i,n}(t) = n[B_{i-1,n-1}(t) - B_{i,n-1}(t)],$$
$$i = 0, 1, \dots, n;$$

- 最大值

- $B_{i,n}(t)$ 在 $[0,1]$ 之间的 $t = i/n$ 处有唯一的局部极大值



- 升阶(degree raising)公式

$$(1-t)B_{i,n}(t) = \left(1 - \frac{i}{n+1}\right)B_{i,n+1}(t)$$

$$tB_{i,n}(t) = \frac{i+1}{n+1}B_{i+1,n+1}(t)$$

$$B_{i,n}(t) = \left(1 - \frac{i}{n+1}\right)B_{i,n+1}(t) + \frac{i+1}{n+1}B_{i+1,n+1}(t)$$



- 积分(Integral)

$$\int_0^1 B_{i,n}(t) = \frac{1}{n+1}$$



B ézier曲线的性质

- 端点(End point)性质

- 端点位置

- 根据*Bernstein*多项式的端点位置性质，可以得到

$$P(0) = P_0, \quad P(1) = P_n,$$

- 因此， B ézier曲线的起点和终点和控制多边形的起点和终点重合



- 切向量 (Tangent Vector)

- 因为

$$P'(t) = n \sum_{i=0}^{n-1} B_{i,n-1}(t)(P_{i+1} - P_i)$$

- 所以

$$P'(0) = n(P_1 - P_0), \quad P'(1) = n(P_n - P_{n-1})$$

- 这意味着起点和终点处曲线的切向量方向与控制多边形第一个及最后一个边的方向一致



- 二阶导数(Second Derivative)

$$P''(t) = n(n-1) \sum_{i=0}^{n-2} (P_{i+2} - 2P_{i+1} + P_i) B_{i,n-2}(t)$$

因此

$$P''(0) = n(n-1) (P_2 - 2P_1 + P_0)$$

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

由曲率公式可得

$$k(0) = \frac{n-1}{n} \cdot \frac{|(P_1 - P_0) \times (P_2 - P_1)|}{|P_1 - P_0|^3}$$

$$k(1) = \frac{n-1}{n} \cdot \frac{|(P_{n-1} - P_{n-2}) \times (P_n - P_{n-1})|}{|P_n - P_{n-1}|^3}$$



- k 阶导数的差分形式

$$P^k(t) = \frac{n!}{(n-k)!} \sum_{i=0}^{n-k} \Delta^k P_i B_{i,n-k}(t) \quad t \in [0,1]$$

– 高阶前向差分向量可以通过低阶前向差分向量递归定义:

$$\Delta^0 P_i = P_i$$

$$\Delta^k P_i = \Delta^{k-1} P_{i+1} - \Delta^{k-1} P_i$$



- 对称 (Symmetry)

- 控制顶点为 $P_i^* = P_{n-i}, (i = 0, 1, \dots, n)$ 的曲线保留了曲线 $P(t)$ 的形状，但方向与之相反

$$\begin{aligned} C^*(t) &= \sum_{i=0}^n P_i^* B_{i,n}(t) = \sum_{i=0}^n P_{n-i} B_{i,n}(t) = \sum_{i=0}^n P_{n-i} B_{n-i,n}(1-t) \\ &= \sum_{i=0}^n P_i B_{i,n}(1-t), \quad t \in [0, 1] \end{aligned}$$



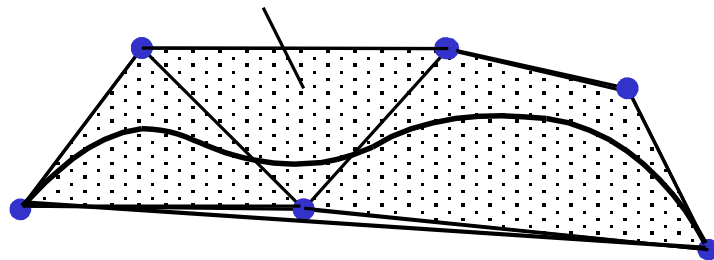
- 凸包性质

- 根据

$$\sum_{i=0}^n B_{i,n}(t) \equiv 1$$

$$0 \leq B_{i,n}(t) \leq 1 (0 \leq t \leq 1, i = 0, 1, \dots, n)$$

- 曲线P(t)在所有控制点的凸包内部



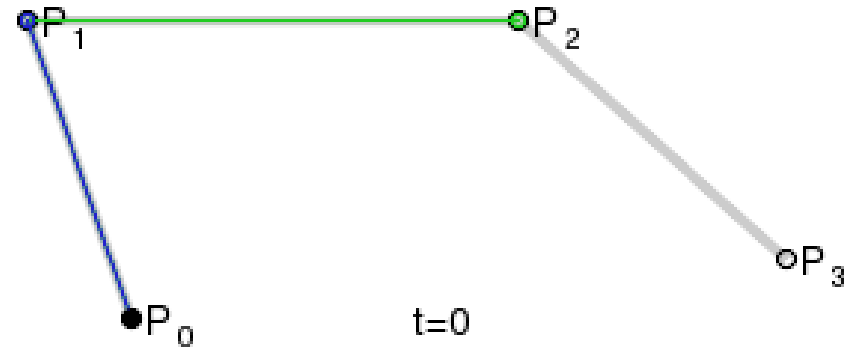


- 几何不变性 (Geometric invariance)
 - 这意味着，一些几何性质不随坐标系变化
 - B ézier曲线的位置和形状依赖于控制多边形的顶点 P_i ，而不是坐标系



de Casteljau算法

- 在工业应用中，经常需要计算曲线上参数 t 位置的点
- 我们并不通过Bézier曲线方程式来求解，而是通过一个递归的数值稳定的 (numerical stable) 算法





- 请注意:

$$\begin{aligned} P(t) &= \sum_{i=0}^n P_i B_{i,n}(t) = \sum_{i=0}^n P_i \left[(1-t) B_{i,n-1}(t) + t B_{i-1,n-1}(t) \right] \\ &= \sum_{i=0}^n \left[(1-t) P_i + t P_{i+1} \right] B_{i,n-1}(t) \end{aligned}$$



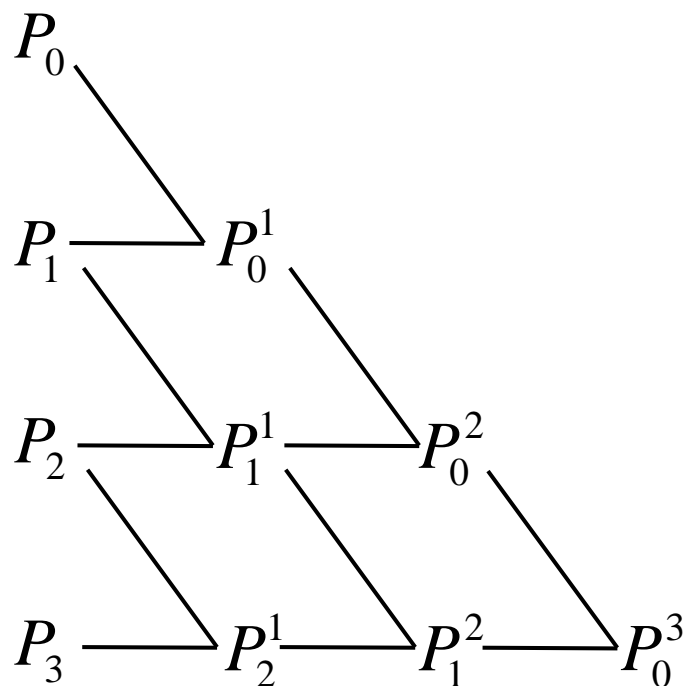
- 我们可以得到Bézier曲线的递推计算公式:

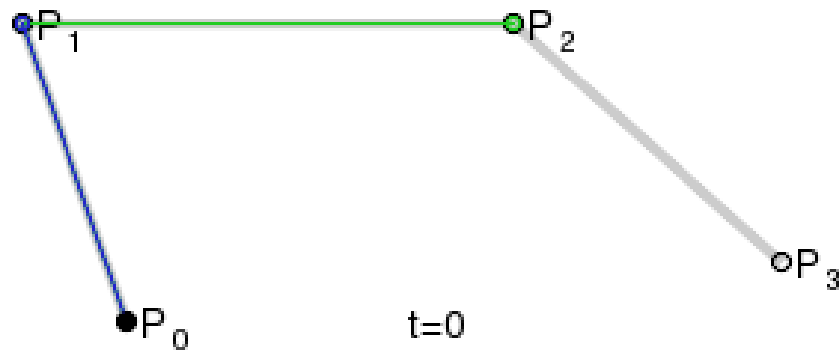
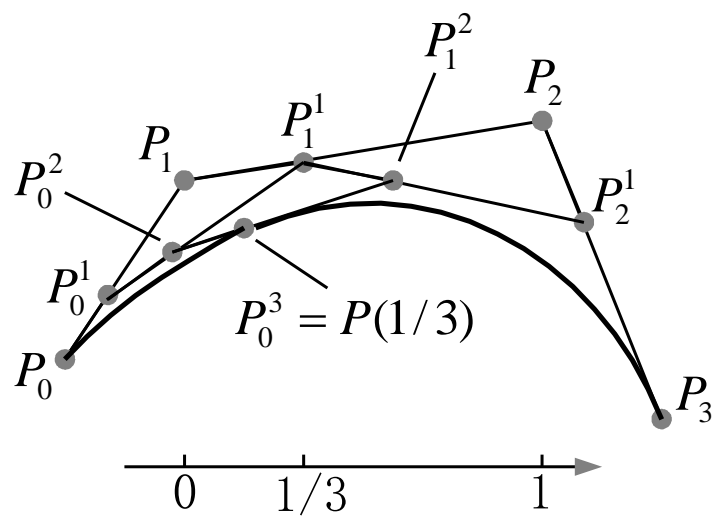
$$P(t) = P_0^n$$

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



- 当 $n = 3$ 时，递推过程如下图所示：





[Curvesystem.exe](#)

de Casteljau的历史贡献



- de Casteljau
 - “The award committee has chosen this year to honor the creator of at least **three fundamental ideas** important in surface modeling. The first idea was the use of **Bernstein polynomials as basis functions** for parametric polynomial curves and surfaces. The second was the use of **multilinear polynomials as a representation for these polynomials**, providing a fundamental tool for understanding, and working with, splines. His third contribution was using this **multilinear form to give an efficient and stable evaluation algorithm.**”



计算Bezier曲线上一点的值

一条3次Bezier曲线顶点为 $(10,0)$, $(30,60)$, $(130,60)$, $(190,0)$, 请问 $t=1/2$ 处曲线的值为?

- ☐ A $(85,30)$
- ☐ B $(100,45)$
- ☒ C $(85,45)$
- ☐ D $(100,30)$

提交



几何连续性

- 在CAD应用中，通常不鼓励通过高阶B ézier曲线来表达曲线
- 通常使用多段光滑连接的低阶B ézier曲线
- 我们能不能用传统概念上的连续性？

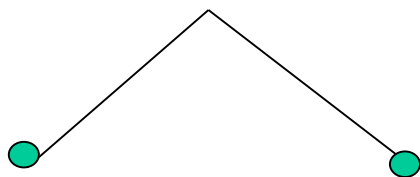
$$\Phi(t) = \begin{cases} V_0 + \frac{V_1 - V_0}{3}t, & 0 \leq t \leq 1 \\ V_0 + \frac{V_1 - V_0}{3} + (t-1)\frac{2(V_1 - V_0)}{3}, & 1 \leq t \leq 2 \end{cases}$$

– 该曲线是什么形状？

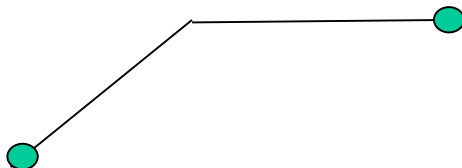


该曲线的形状为

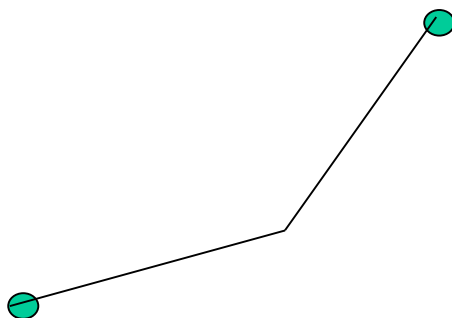
A



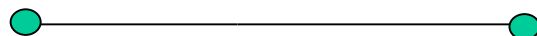
B



C



D



提交



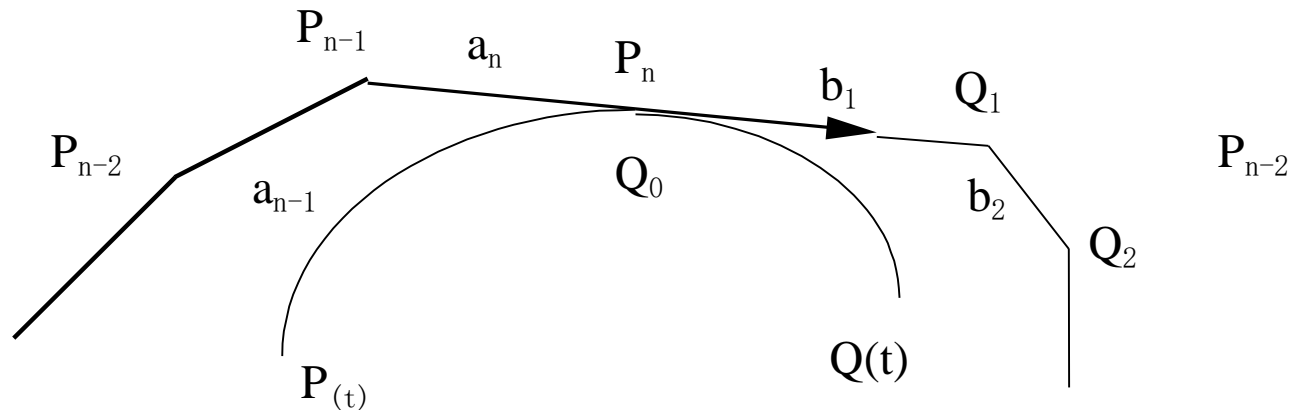
几何连续性 vs. 函数连续性

- 请注意 $\Phi'(1^-) = \frac{1}{3}(V_1 - V_0)$ $\Phi'(1^+) = \frac{2}{3}(V_1 - V_0)$
- 这意味着该导函数 $\Phi'(t)$ 在 1 处不是连续的，即
 - $\|\Phi'(1^-)\|^2 \neq \|\Phi'(1^+)\|^2$
- 但该方程事实上是一条直线
- 这种现象意味着关于连续性的传统观念并不适合用于描述 CAD 和图形学中形状的连续性
- 这就是为什么我们采用“几何连续性” (Geometric continuity) 的原因
 - 刘鼎元和梁友栋



- 两个分别由控制点 $P_i (i = 0, 1, \dots, n)$ 和 $Q_j (j = 0, 1, \dots, m)$ 定义的 Bézier 曲线
- 两条曲线共享端点，且：

$$a_i = P_i - P_{i-1}, b_j = Q_j - Q_{j-1}$$





G^n 几何连续性

– G^0 连续 $P_n = Q_0$

– G^1 连续

$Q_0 - Q_1 = \alpha(P_n - P_{n-1})$ 即 $P_{n-1}, P_n = Q_0, Q_1$ 共线

– G^2 连续 (曲率连续)

$$Q_2 = \left(\alpha^2 + 2\alpha + \frac{\beta}{n-1} + 1 \right) P_n - \left(2\alpha^2 + 2\alpha + \frac{\beta}{n-1} \right) P_{n-1} + \alpha^2 P_{n-2}$$

$$Q''(0) = \alpha^2 P''(1) + \beta P'(1)$$



升阶 (Degree raising/elevation)

- 升阶意味着通过增加控制点的方式来增加 **B ézier** 曲线的阶数，但保持曲线的形状和方向不变
 - 升阶可以增加形状控制的灵活性
 - 升阶之后，控制点会发生变化
 - 怎样增加多项式的阶？怎样增加 **B ézier** 曲线的阶？



- 请注意

$$\begin{aligned} P(t) &= \sum_{i=0}^n P_i B_{i,n}(t) = \sum_{i=0}^n P_i ((1-t) + t) B_{i,n}(t) \\ &= \sum_{i=0}^{n+1} \left(\frac{n+1-i}{n+1} P_i + \frac{i}{n+1} P_{i-1} \right) B_{i,n+1}(t) \end{aligned}$$

- 我们可以得到升阶公式:

$$P_i^* = \frac{i}{n+1} P_{i-1} + \left(1 - \frac{i}{n+1} \right) P_i \quad (i = 0, 1, \dots, n+1)$$



- 上述公式说明
 - 新的控制顶点是旧的控制顶点的系数为 $i/(n+1)$ 的线性组合
 - 新的控制多边形在旧控制多边形的凸包内部
 - 新的控制多边形更加靠近曲线
- 演示: [curve-system](#)



降阶

- 降阶是升阶的逆过程
 - 我们可以在降低多项式阶数的同时保持其形状不变吗？
 - 降阶的目的是寻找一组新的控制点定义的曲线使得误差最小

$$P_i^* (i = 0, 1, \dots, n-1)$$



– 假设 P_i 是 P_i^* 升阶的结果:

$$P_i = \frac{n-i}{n} P_i^* + \frac{i}{n} P_{i-1}^*$$

– 我们可以得到两个递推公式

$$P_i^\# = \frac{nP_i - iP_{i-1}^\#}{n-i} \quad i = 0, 1, \dots, n-1$$

$$P_{i-1}^* = \frac{nP_i - (n-i)P_i^*}{i} \quad i = n, n-1, \dots, 1$$



- 因此我们有两种降阶方案

- Forrest (1972)

$$\hat{P}_i = \begin{cases} P_i^{\#}, & i = 0, 1, \dots, \left\lfloor \frac{n-1}{2} \right\rfloor \\ P_i^*, & i = \left\lfloor \frac{n-1}{2} \right\rfloor + 1, \dots, n-1 \end{cases}$$

- Farin (1983)

$$\hat{P}_i = \left(1 - \frac{i}{n-1}\right) P_i^{\#} + \frac{i}{n-1} P_i^*$$



- 另一个思路：退化的曲线，可以精确降阶。
- 约束优化方法
 - 每个顶点给一个扰动 ε_i
 - 扰动后的曲线满足退化条件

$$\sum_{i=0}^n (-1)^i \binom{n}{i} (P_i + \varepsilon_i) = 0$$

- 求解约束优化问题，使得：

$$\sum_{i=0}^n \|\varepsilon_i\|^2 = \min$$



- 精确降阶的相关参考文献

- M. A. Watkins and A. J. Worsey, Degree reduction of Bézier curves, Computer Aided Design, **20**(7), 1988, 398-405, 177 cites
- 胡事民、孙家广、金通洸等, **Approximate degree reduction of Bézier curves, Tsinghua Science and Technology, No.2, 1998, 997-1000. (was reported in national CAGD conference, 1993), 60 cites**
- 雍俊海、胡事民、孙家广等, Degree reduction of B-spline curves, Computer Aided Geometric Design, 2001, Vol. 13, NO. 2, 2001, 117-127. 49 cites



B ézier曲面： 矩形域曲面

- 矩形域B ézier曲面

- 假设 $P_{ij}(0,1,\cdots,m; j=0,1,\cdots,n)$ 是 $(m+1)\times(n+1)$ 的控制点集, $m\times n$ 阶矩形域B ézier曲面可以通过张量积的形式定义

$$P(u,v) = \sum_{i=0}^m \sum_{j=0}^n P_{ij} B_{i,m}(u) B_{j,n}(v) \quad u,v \in [0,1]$$

- 其中 $B_{i,m}(u) = C_m^i u^i (1-u)^{m-i}$ 和 $B_{j,n}(v) = C_n^j v^j (1-v)^{n-j}$ 是**Bernstein**多项式



– Bézier曲面的矩阵表达

$$P(u, v) = \begin{bmatrix} B_{0,n}(u) & B_{1,n}(u) & \cdots & B_{m,n}(u) \end{bmatrix} \begin{bmatrix} P_{00} & P_{01} & \cdots & P_{0m} \\ P_{10} & P_{11} & \cdots & P_{1m} \\ \cdots & \cdots & \cdots & \cdots \\ P_{n0} & P_{n1} & \cdots & P_{nm} \end{bmatrix} \begin{bmatrix} B_{0,m}(v) \\ B_{1,m}(v) \\ \cdots \\ B_{n,m}(v) \end{bmatrix}$$



- 矩形域Bézier曲面的性质

- Bézier曲面保持了Bézier曲线相似的性质:

- 控制网格的四个顶点同样是Bézier曲面的顶点

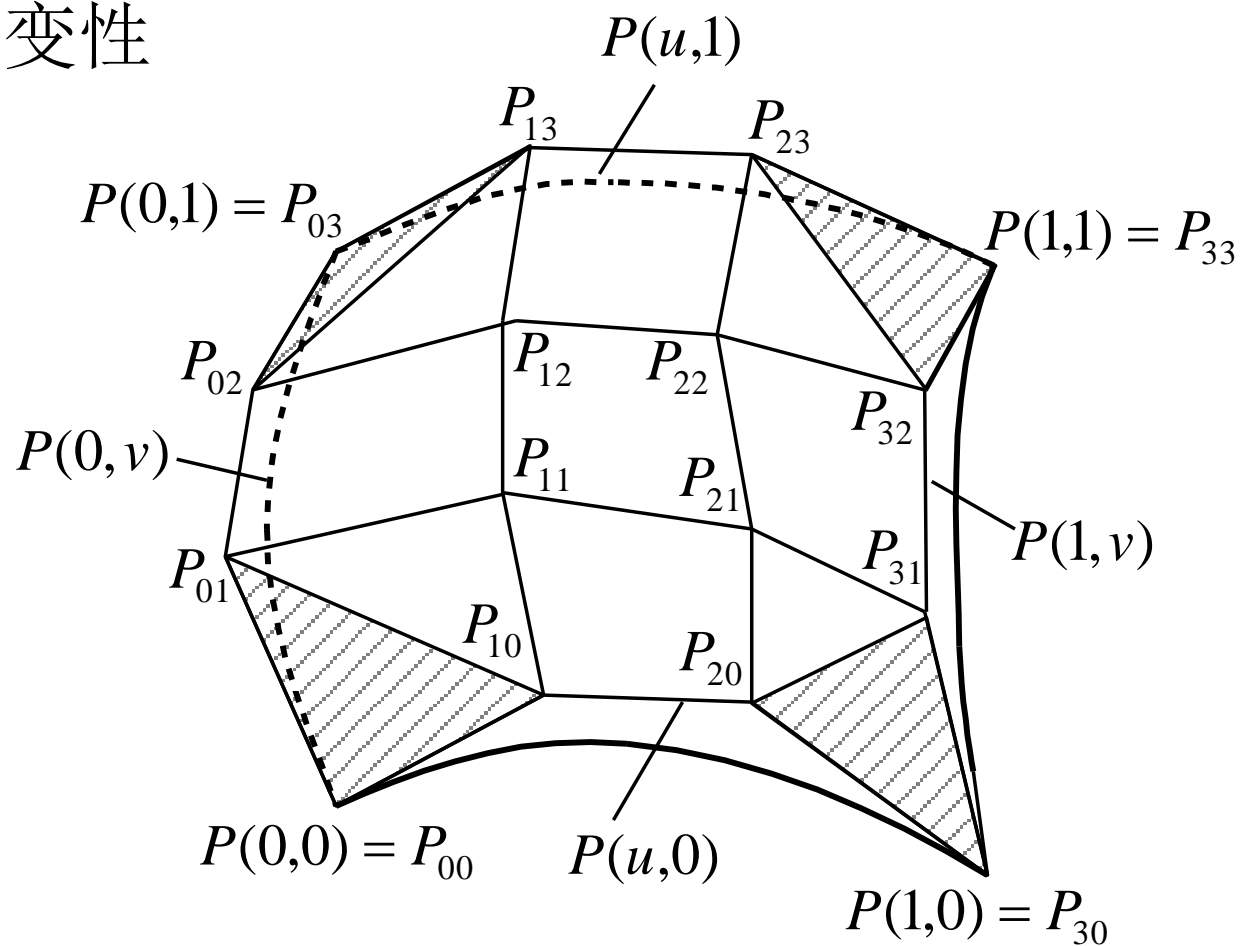
$$P(0, 0) = P_{00} \quad P(1, 0) = P_{m0} \quad P(0, 1) = P_{0n} \quad P(1, 1) = P_{mn}$$

- 以下三角形指定了四个角处的切平面

$$P_{00}P_{10}P_{01} \quad P_{0n}P_{1n}P_{0,n-1} \quad P_{mn}P_{m,n-1}P_{m-1,n} \quad P_{m0}P_{m-1,0}P_{m1}$$



- 几何不变性
- 对称性
- 凸包





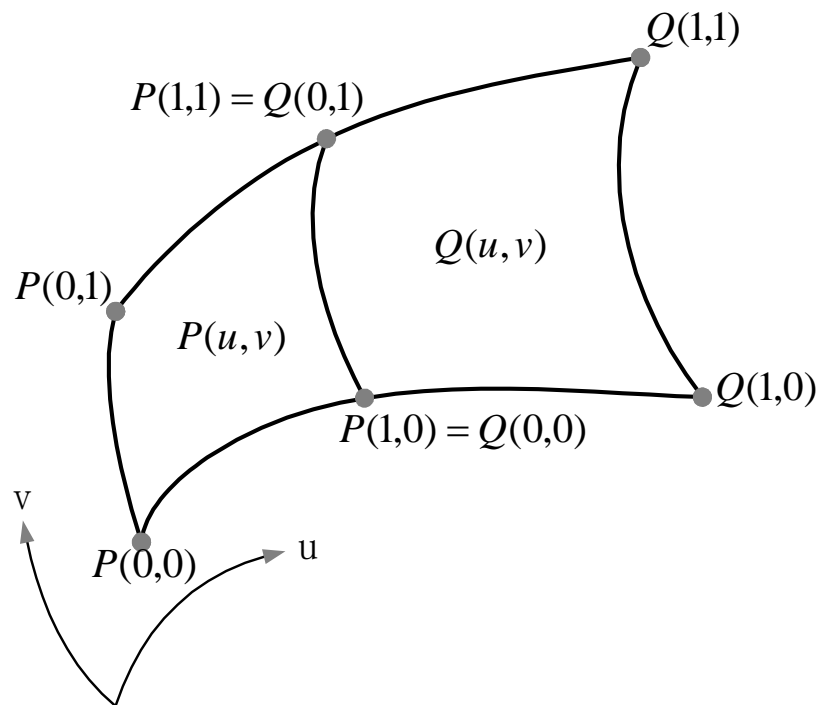
- 几何连续性

- 给定2个 $m \times n$ 阶控制点为 P_{ij} 和 Q_{ij} 的Bézier曲面:

$$P(u, v) = \sum_{i=0}^m \sum_{j=0}^n P_{ij} B_{i,m}(u) B_{j,n}(v)$$

$$Q(u, v) = \sum_{i=0}^m \sum_{j=0}^n Q_{ij} B_{i,m}(u) B_{j,n}(v)$$

$$u, v \in [0, 1]$$





– G^0 连续的条件:

$$P(1, v) = Q(0, v)$$

$$\text{i.e., } P_{ni} = Q_{0i}, \quad (i = 0, 1, \dots, m)$$

– G^1 连续的条件:

$$Q_u(0, v) = \alpha(v)P_u(1, v) + \beta(v)P_v(1, v)$$

任意系数



- de Casteljau算法

- B ézier曲线的de Casteljau算法可以被推广到曲面上，给定控制点集 $P_{ij} (i = 0, 1, \dots, m; j = 0, 1, \dots, n)$ 和参数 (u, v) ，我们有

$$P(u, v) = \sum_{i=0}^{m-k} \sum_{j=0}^{n-l} P_{i,j}^{k,l} B_{i,m}(u) B_{j,n}(v) = \dots = P_{00}^{m,n}$$

$$u, v \in [0, 1]$$



$P_{i,j}^{k,l}$ 通过以下递归公式定义

$$P_{i,j}^{k,l} = \begin{cases} P_{ij} & (k = l = 0) \\ (1-u)P_{ij}^{k-1,0} + uP_{i+1,j}^{k-1,0} & (k = 1, 2, \dots, m; l = 0) \\ (1-v)P_{0,j}^{m,l-1} + vP_{0,j+1}^{m,l-1} & (k = m, l = 1, 2, \dots, n) \end{cases}$$

或

$$P_{ij}^{k,l} = \begin{cases} P_{ij} & (k = l = 0) \\ (1-v)P_{ij}^{0,l-1} + vP_{i,j+1}^{0,l-1} & (k = 0; l = 1, 2, \dots, n) \\ (1-u)P_{i0}^{k-1,n} + uP_{i+1,0}^{k-1,n} & (k = 1, 2, \dots, m; l = n) \end{cases}$$



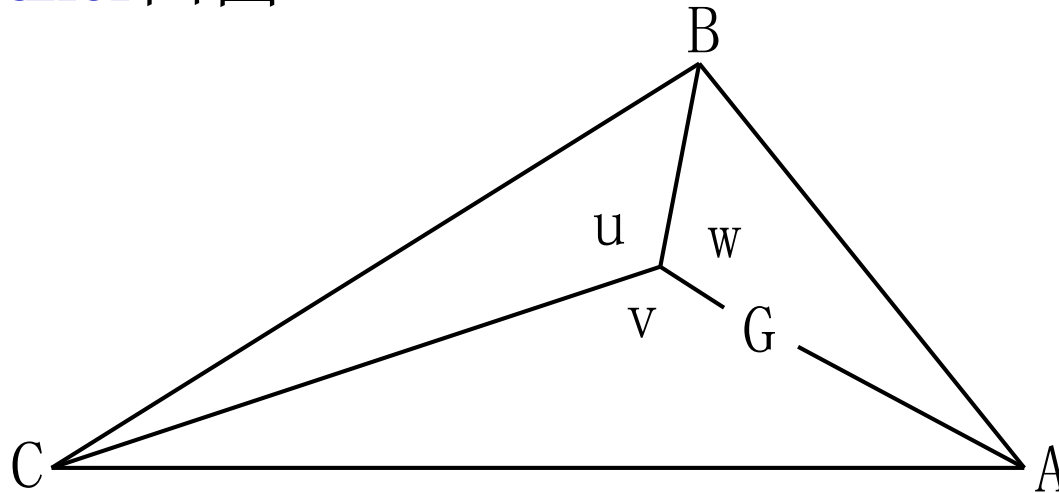
- de Casteljau算法直观的解释:

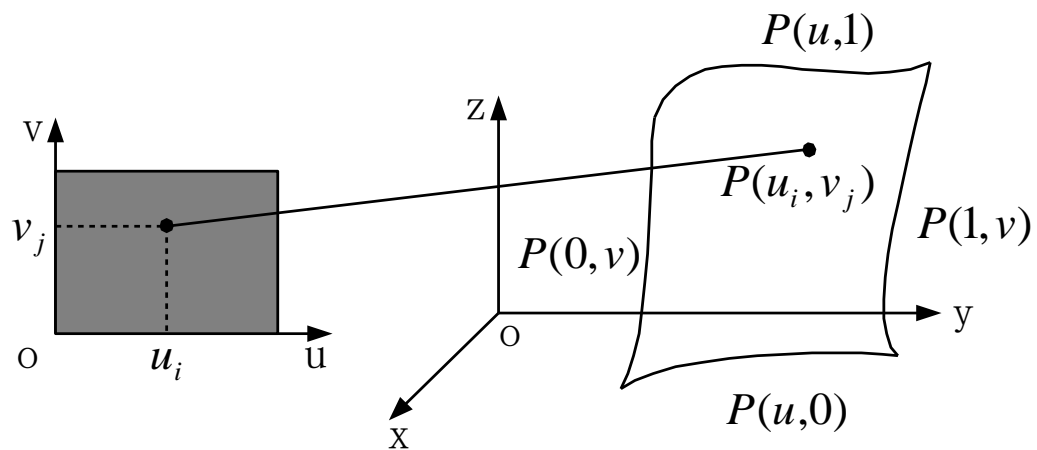
$$\begin{aligned} P(u, v) &= \sum_{i=0}^m \sum_{j=0}^n P_{ij} B_{i,m}(u) B_{j,n}(v) \\ &= \sum_{i=0}^m \left(\sum_{j=0}^n P_{ij} B_{j,n}(v) \right) B_{i,m}(u) \end{aligned}$$



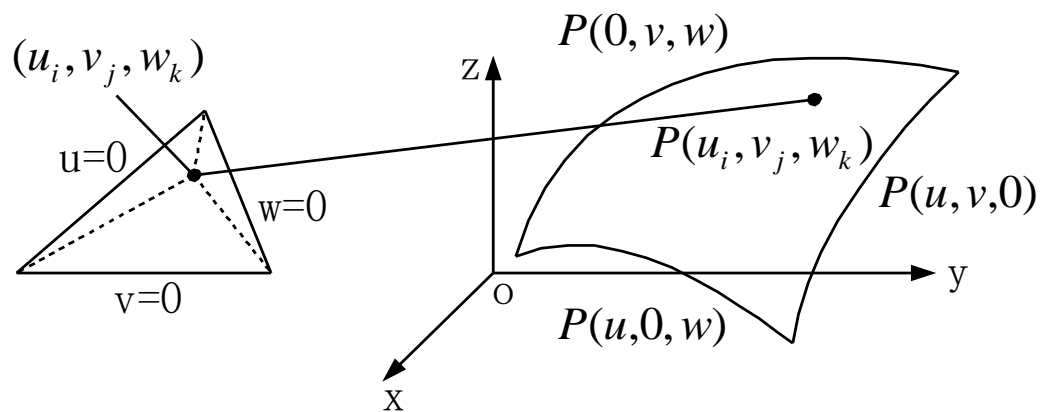
三角域Bézier曲面

- 三角域Bézier曲面定义在三角形域，而不是矩形域
- 用Barycentric坐标(u, v, w)来定义三角域Bézier曲面





(a) 矩形域曲面



(b) 三角域曲面

图3. 1. 17



- Bernstein基函数:

$$B_{i,j,k}(u,v,w) = \frac{n!}{i!j!k!} u^i v^j w^k \quad u, v, w \in [0,1]$$

其中 $i + j + k = n$ 且 $i, j, k \geq 0$.

– 有 $\frac{1}{2}(n+1)(n+2)$ 个 n 阶 Bernstein 基函数

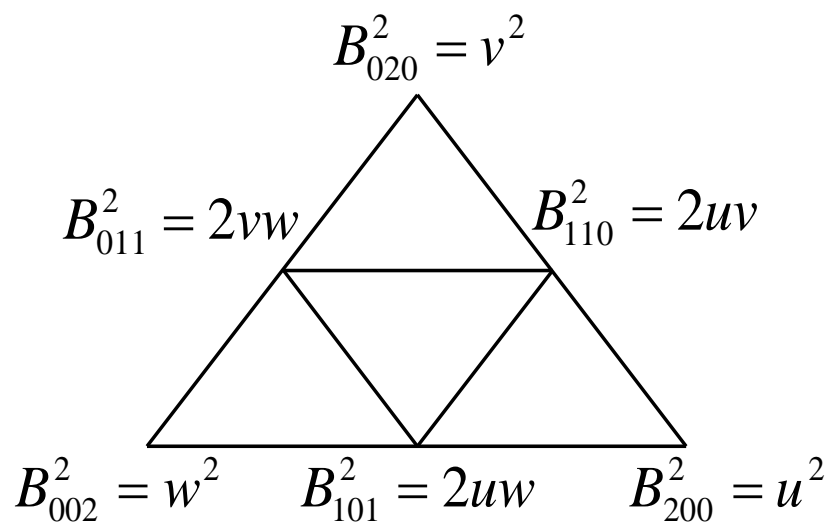


图3.1.19 二次Bernstein基的三角阵列

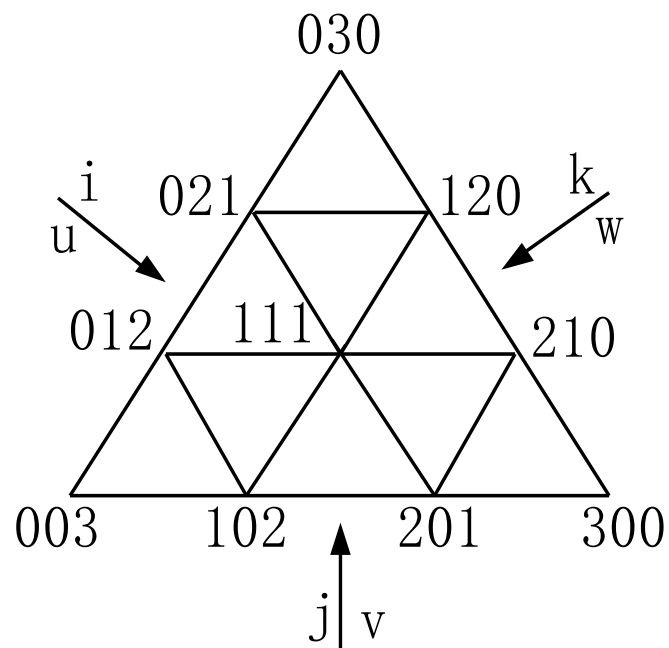


图3.1.20 $n=3$ 时三角域各节点指标



– 非负性(non-negative)和归一性(unity)

$$B_{i,j,k}^n(u, v, w) \geq 0$$

$$\sum_{i+j+k=n} B_{i,j,k}^n(u, v, w) = 1$$

– 递归:

$$\begin{aligned} B_{i,j,k}^n(u, v, w) &= uB_{i-1,j,k}^{n-1}(u, v, w) + vB_{i,j-1,k}^{n-1}(u, v, w) \\ &\quad + wB_{i,j,k-1}^{n-1}(u, v, w) \end{aligned}$$



- 三角域Bézier曲面的定义

$$\begin{aligned} P(u, v, w) &= \sum_{i+j+k=n} P_{i,j,k} B_{i,j,k}^n(u, v, w) \\ &= \sum_{i=0}^n \sum_{j=0}^{n-i} P_{i,j,k} B_{i,j,k}^n(u, v, w) \end{aligned}$$

$$u, v, w \geq 0, u + v + w = 1$$

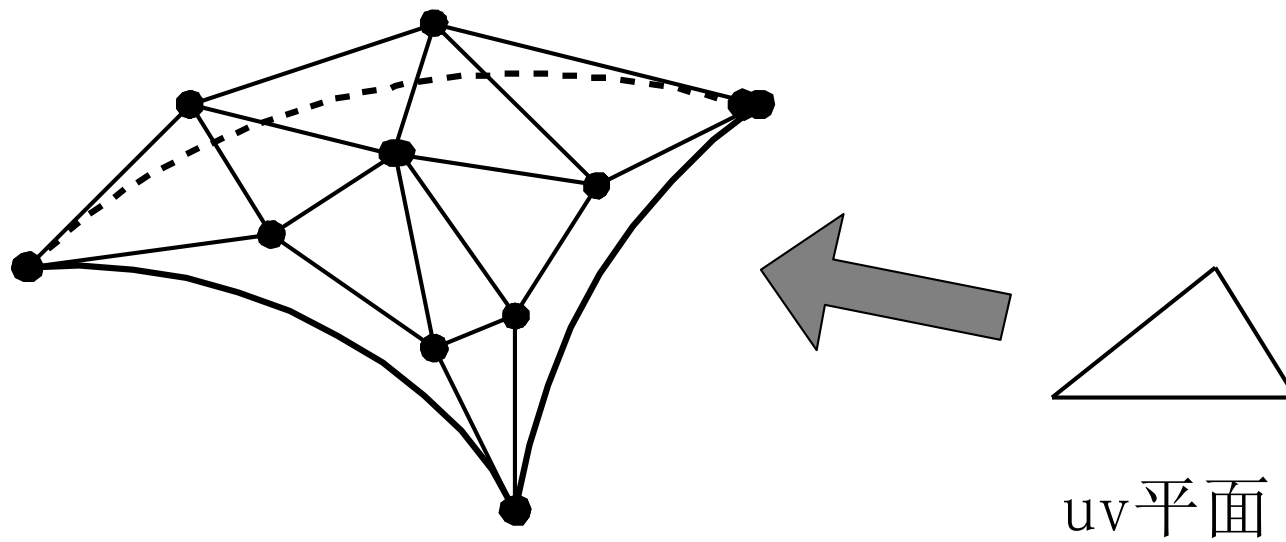
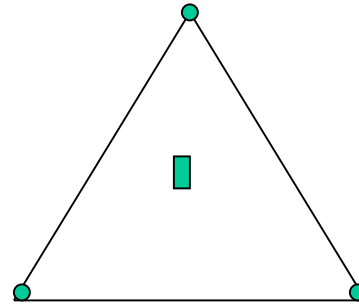
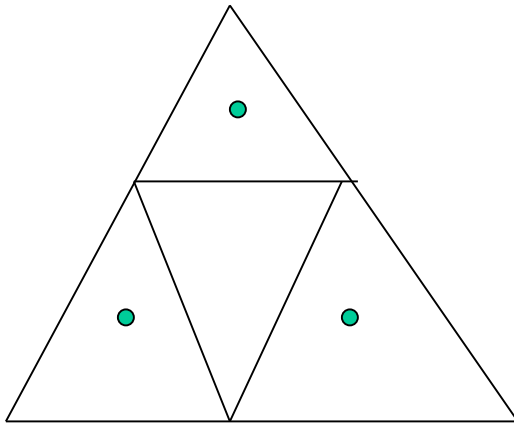


图3.1.21 三边Bezier曲面片



- de Casteljau算法

$$P_{i,j,k} = uP_{i+1,j,k} + vP_{i,j+1,k} + wP_{i,j,k+1}$$





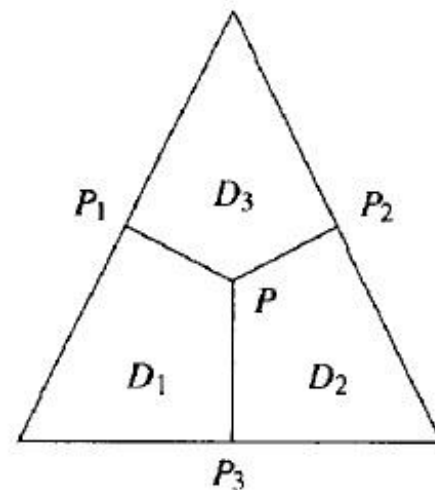
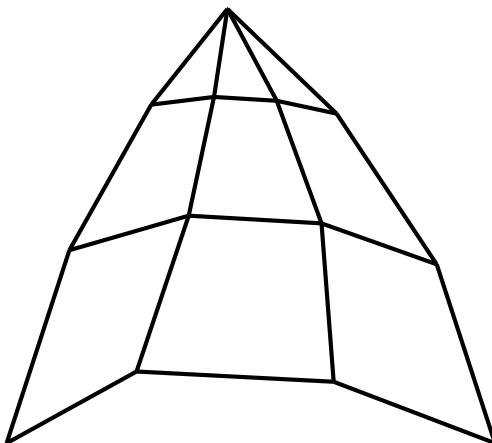
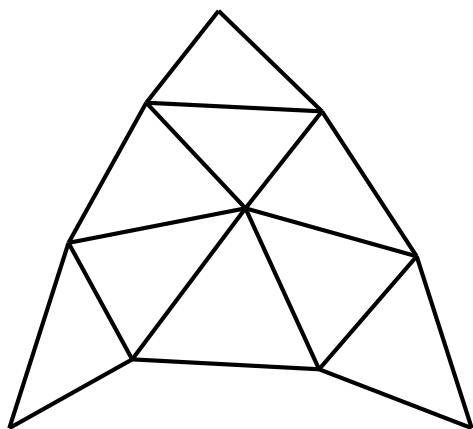
矩形域与三角域Bézier曲面的转换

- 由于矩形域与三角域使用不同的基函数，它们在同一个CAD系统中使用的时候会存在一些问题，因此有必要研究它们之间的转换
 - 三角域曲面 \rightarrow 矩形域曲面
 - 矩形域曲面 \rightarrow 三角域曲面

矩形域与三角域Bézier曲面的转换



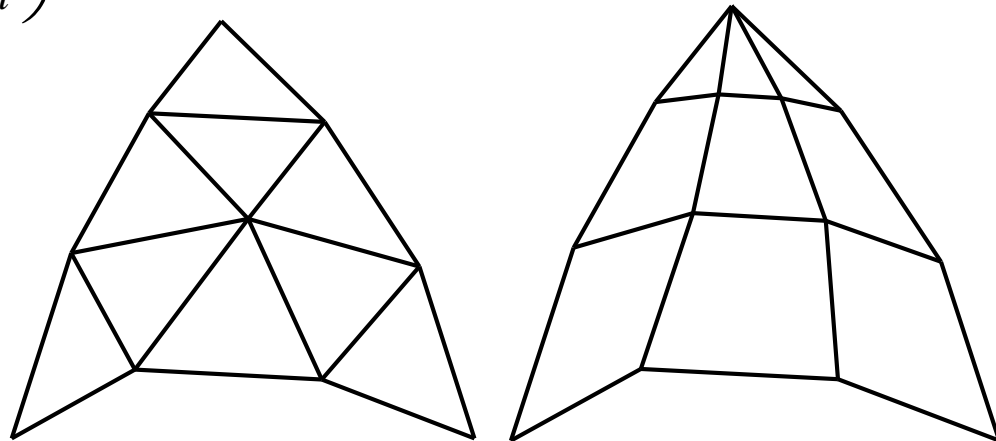
- 一个三角域曲面可以被表示为一个退化的矩形域曲面或者三个非退化矩形域曲面





- 转化为一个退化的矩形域曲面

$$\begin{pmatrix} P_{i0} \\ P_{i1} \\ \vdots \\ P_{in} \end{pmatrix} = A_1 A_2 \dots A_i \begin{pmatrix} T_{i0} \\ T_{i1} \\ \vdots \\ T_{i,n-i} \end{pmatrix}, \quad i = 0, 1, \dots, n,$$





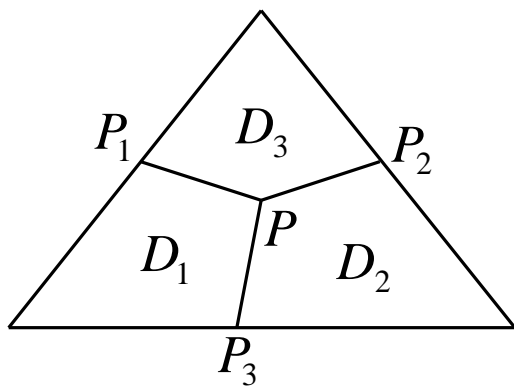
$A_i (i = 0, 1, \dots, n)$ 是一个升阶算子:

$$A_k = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 & 0 \\ 1 & \frac{n-k}{n+1-k} & 0 & \dots & 0 & 0 \\ \frac{1}{n+1-k} & \frac{n-k}{n+1-k} & 0 & \dots & 0 & 0 \\ 0 & \frac{2}{n+1-k} & \frac{n-k-1}{n+1-k} & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & \frac{n-k}{n+1-k} & \frac{1}{n+1-k} \\ 0 & 0 & 0 & \dots & 0 & 1 \end{pmatrix}_{(k+1) \times k}$$

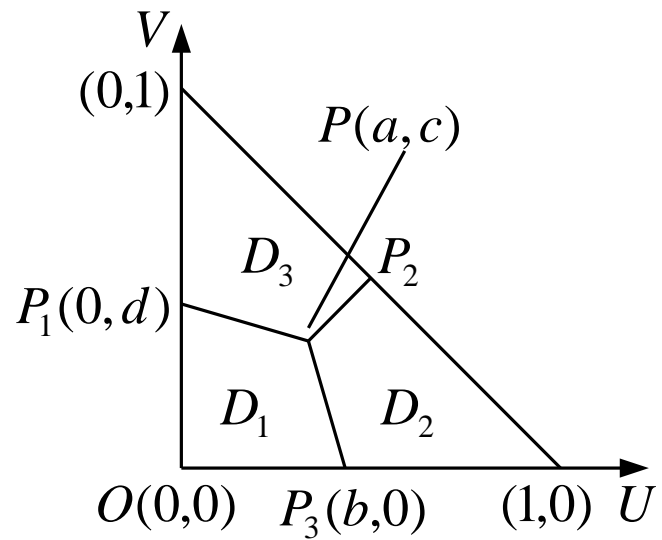
Shi-Min Hu, Conversion between triangular and rectangular Bézier patches, Computer Aided Geometric Design, 2001, 18(7), 667-671.



- 转化为三个矩形域曲面
– 区域分解



(a)



(b)

三角定义域的分解



- 一些算子 (operator)
 - 单位算子: $I : IT_{ij} = T_{ij}$
 - 移位算子: $E_i : E_1 T_{ij} = T_{i+1,j}, E_2 T_{ij} = T_{i,j+1}$
 - 差分算子: $\Delta_i : \Delta_1 T_{ij} = T_{i+1,j} - T_{ij}, \Delta_2 T_{ij} = T_{i,j+1} - T_{ij}$
- 利用这些算子, 我们可以将三角域Bézier曲面重写为:

$$\begin{aligned} T(u, v) &= (uE_1 + vE_2 + (1-u-v)I)^n T_{00} \\ &= (\Delta_1 u + \Delta_2 v + I)^n T_{00} \end{aligned}$$



- 定义在 D_1 上的控制顶点可以通过下式获得

$$P_{ij} = \sum_{\substack{k=0 \\ k+l=j}}^i \sum_{l=0}^{n-i} \binom{i}{k} \frac{\binom{n-i}{l}}{\binom{n}{j}} Q_{kl}^{(i)} \quad 0 \leq i, j \leq n, \quad (5)$$

in which for $0 \leq i \leq n, 0 \leq k \leq i, 0 \leq l \leq n-i,$

$$Q_{kl}^{(i)} = (aE_1 + cE_2 + (1-a-c)I)^k (bE_1 + (1-b)I)^{i-k} (dE_2 + (1-d)I)^l T_{00} \quad (6)$$



ELSEVIER

Computer Aided Geometric Design 13 (1996) 219–226

**COMPUTER
AIDED
GEOMETRIC
DESIGN**

Conversion of a triangular Bézier patch into three rectangular Bézier patches¹

Shi-Min Hu

Department of Applied Mathematics, Zhejiang University, Hangzhou, 310027, People's Republic of China

Received April 1994; revised February 1995

Abstract

In this paper, we give an explicit formula for converting a triangular Bézier patch into three non-degenerate rectangular Bézier patches of the same degree. The use of certain operators simplifies the formulation of such a decomposition. The formula yields a stable recursive algorithm for computing the control points of the rectangular patches. We also illustrate the formula with an example.



今日人物: Christoph M. Hoffmann

- Christoph M. Hoffmann: Purdue University
发表论文313+篇, 引用10000+次, H-Index: 46+
 - 1974, Wisconsin 大学获博士学位
 - 1984-1986, Cornell University,
和**John Hopcroft** 合作
 - 主要学术贡献:
 - 几何计算的鲁棒性
 - 几何约束求解
 - 2011年获B ézier奖





什么是几何约束求解（GCS）？

- CAD的基础：几何造型内核+GCS
- Hopcroft的经典论文
 - *John E. Hopcroft and Richard M. Karp, An $n^{5/2}$ Algorithm for Maximum Matchings in Bipartite Graphs, SIAM J. Comput., 1973, 2(4), 225–231.*
- Hoffmann的一个示意图

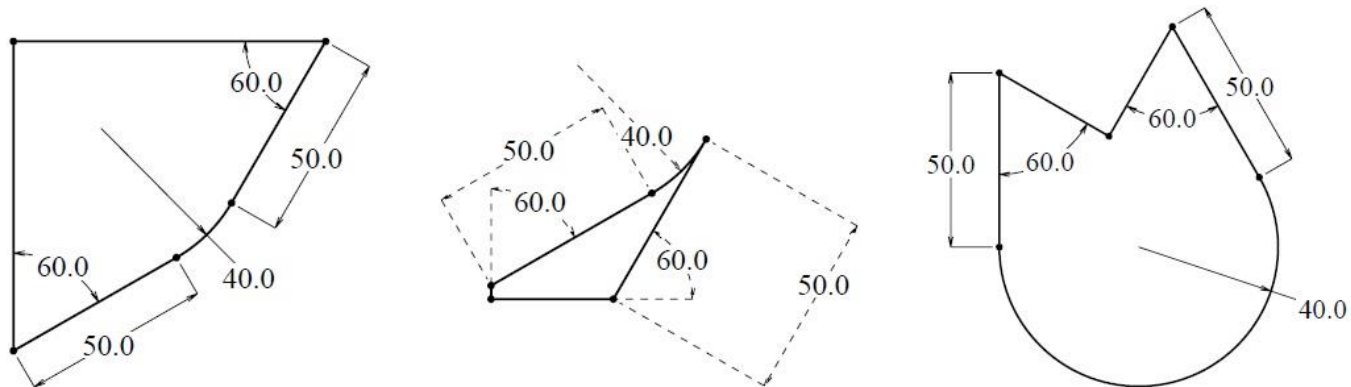


Figure 5: A well-constrained sketch may yield fundamentally different solutions.



谢谢！