

## 第二章 计算机动画技术背景

### 2.1 三维空间变换

2.1.1 变换的数学基础

2.1.2 几何变换

2.1.3 坐标变换

2.1.4 投影变换

2.1.5 图形的显示流程

2.1.6 OpenGL中的变换

### 2.2 曲线与曲面

2.2.1 基础知识

2.2.2 B é zier曲线/曲面

2.2.3 B样条曲线/曲面

2.2.4 NURBS曲线/曲面

2.2.5 Coons曲面

2.2.6 三维物体的描述方法

## 2.2 曲线与曲面

### 2.2.1 基础知识

### 2.2.2 B é z i e r 曲线/曲面

### 2.2.3 B样条曲线/曲面

### 2.2.4 NURBS曲线/曲面

### 2.2.5 Coons曲面

### 2.2.6 三维物体的描述方法

## 2.2.1 曲线曲面基础知识

- 从卫星的轨道、导弹的弹道，到汽车和飞机等的外形，直至日常生活中的图案和花样设计

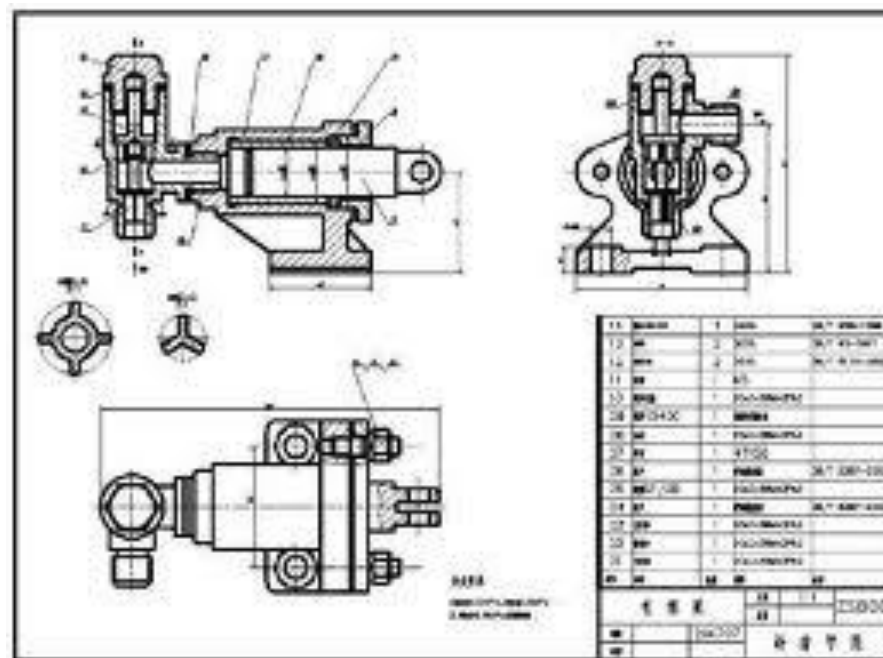




第一类**曲线**是可以用一个标准的**解析式**来表示，称为曲线的方程等。

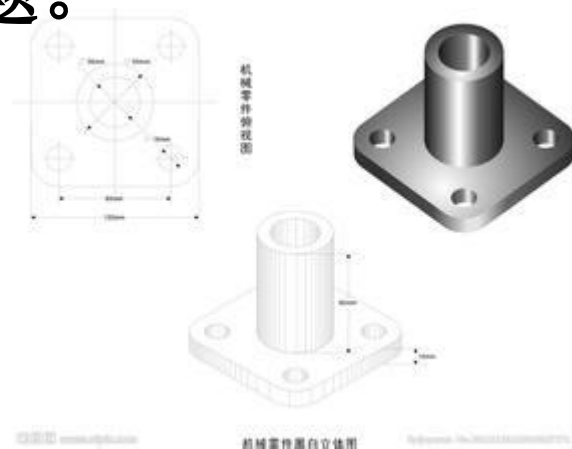
第二类**曲线**的特点是不能确切给出描述整个曲线的方程，它们往往是由一些从实际测量得到的一系列**离散数据点**来确定。这些数据点也称为**型值点**。

## 早期手工绘图



# 工业产品的形状大致上可分为两类：

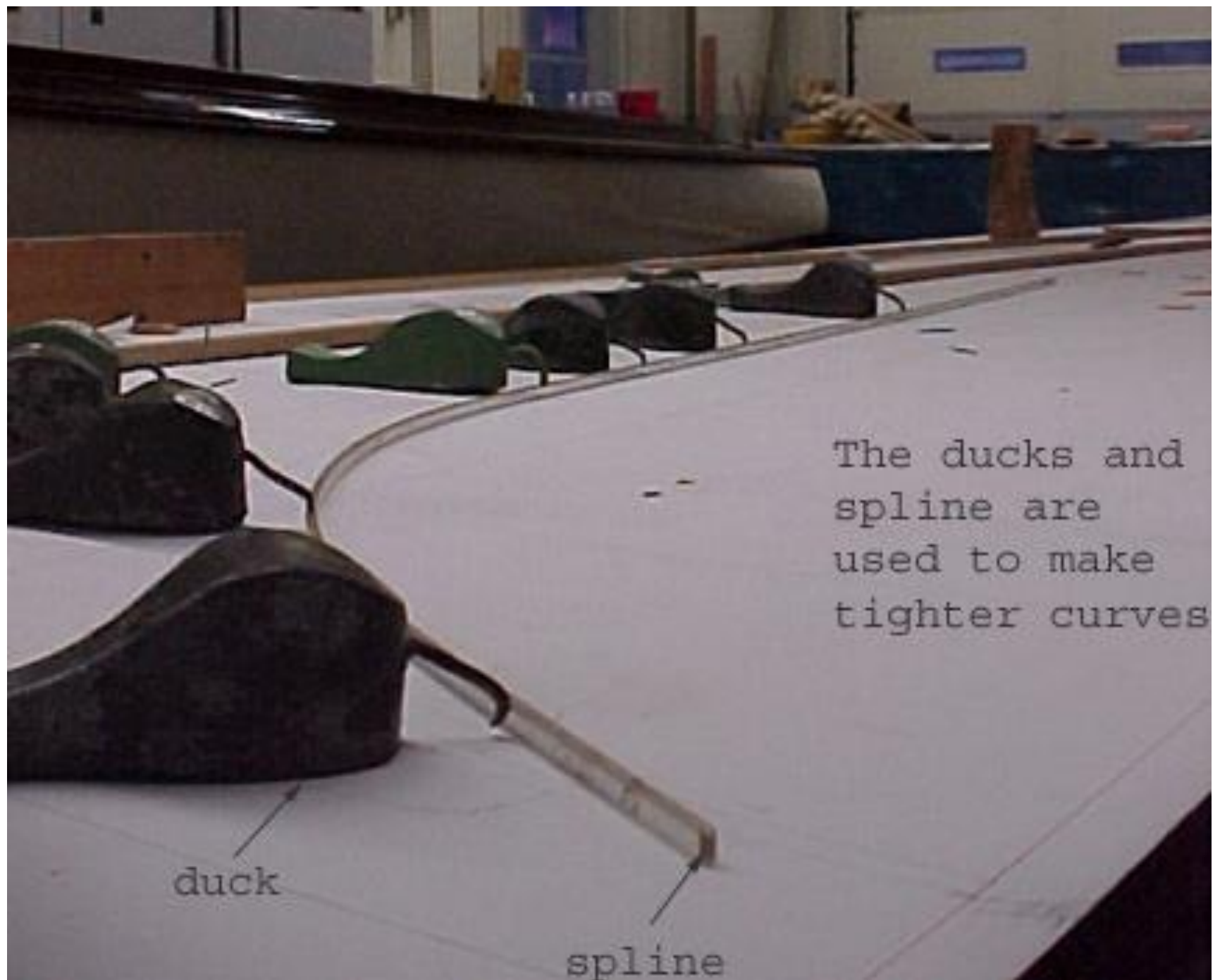
- 第一类是仅由初等解析曲面例如平面、圆柱面、圆锥面、球面、圆环面等组成，大多数机械零件属于这一类。
- 第二类以复杂方式自由地变化的曲线曲面即所谓自由曲线曲面组成，如飞机、汽车、船舶的外形零件。自由曲线曲面因不能由画法几何与机械制图表达清楚，成为摆在工程师面前首要解决的问题。



## 对于复杂曲线和曲面的绘制方法

先确定一些满足条件的、位于曲线上的坐标点，然后借用**曲线板**把这些点分段光滑地连接成曲线。绘出的曲线的精确程度，则取决于所选择的数据点的精度和数量，坐标点的精度高，点的数量取得多，则连成的曲线愈接近于理想曲线。





The ducks and  
spline are  
used to make  
tighter curves

duck

spline



# 存在的问题

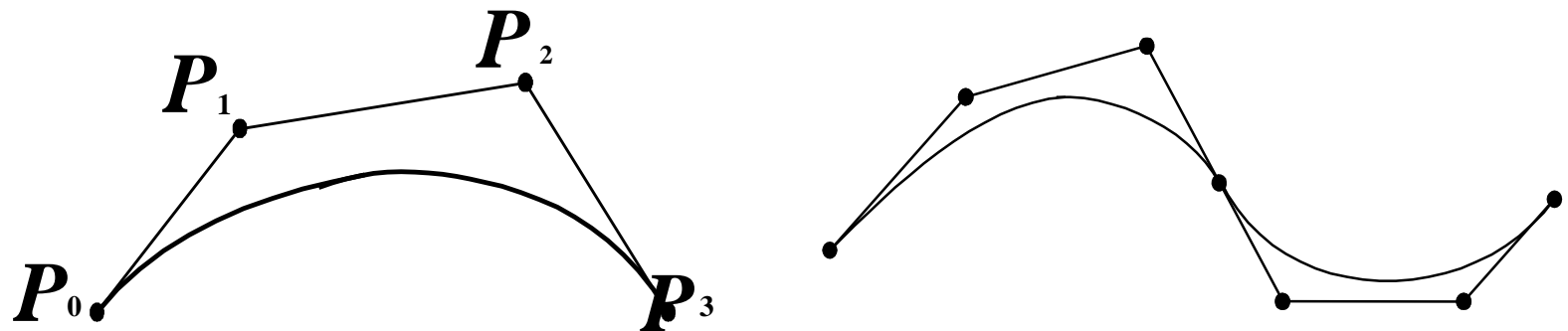
- 不灵活
- 不易修改
- 耗费时间

## 解决办法

利用计算机设计曲线和曲面

- 1963年，美国波音(Boeing)飞机公司的Ferguson将曲线曲面表示成参数矢量函数形式，从此，**参数形式**成为自由曲线曲面数学描述的标准形式。
- 1964年 MIT的孔斯（Coons）用封闭曲线的四条边界定义一块曲面；
- 1971年 雷诺汽车公司的贝塞尔（Bezier）发表了一种用控制多边形定义曲线和曲面的方法；同期，法国雪铁龙（Citroen）汽车公司的德卡斯特里奥（de Casteljau）也独立地研究出与Bezier类似的方法；
- 1972年 德布尔（DeBoor）给出了B样条的标准计算方法；

- 1974年，美国通用汽车公司的戈登（Gorden）和里森费尔德（Riesenfeld）将B样条理论用于形状描述，提出了B样条曲线和曲面；
- 1975年，美国Syracuse大学的佛斯普里尔（Versprill）提出了有理B样条方法；
- 80年代后期皮格尔（Piegl）和蒂勒（Tiller）将有理B样条发展成非均匀有理B（NURBS）样条方法，并已成为当前自由曲线和曲面描述的最广为流行的技术。



# 曲线、曲面包括

- Bézier曲线/曲面
- B样条曲线/曲面
- NURBS曲线/曲面
- Coons曲面

# 曲线、曲面包括表示方法

- 两种表示方法
  - ✓ 参数表示
  - ✓ 非参数表示（两种）
    - 显式
    - 隐式

# 显式表示

- 显式表示  
平面曲线

$$y = f(x)$$

- 每一个x值只对应一个y值，所以显式方程不能表示封闭或者多值曲线，例如圆。

# 隐式表示

- 隐式表示  
平面曲线

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

$$ax^2 + 2bxy + cy^2 + 2dx + 2ey + f = 0$$

- 隐式表示的优点是易于判断函数 $f(x,y)$ 是否大于、小于或等于零，也就易于判断点是落在所表示曲线上或在曲线的哪一侧。

# 隐式表示

- 三维空间曲线的隐式表示

$$\begin{cases} f(x, y, z) = 0 \\ g(x, y, z) = 0 \end{cases}$$



# 曲线的非参数表示存在问题

- 与坐标系相关
- 会出现斜率为无穷大的情况(如垂线)
- 非平面曲线难用常系数的非参数化函数表示
- 不利于计算和编程

# 参数表示

- 将曲线上各点的坐标变量显式地表示成参数  $t$  的函数形式

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

- 其中  $x(t)$ ,  $y(t)$  和  $z(t)$  分别是参数  $t$  的显式函数

$$x = x(t)$$

$$y = y(t)$$

$$z = z(t)$$



- 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任何量，如时间、角度等
- 连接  $P_0(x_0, y_0)$  和  $P_1(x_1, y_1)$  两点的直线段的参数方程可写为

$$P = P_0 + (P_1 - P_0) * t \Leftrightarrow \begin{cases} x = x_0 + (x_1 - x_0) * t, \\ y = y_0 + (y_1 - y_0) * t, \end{cases} \quad t \in [0, 1]$$

# 参数表示比非参数表示更优越

- 更大的自由度
- 参数方程的形式不依赖于坐标系的选取，具有形状不变性；
- 在参数表示中，变化率以切矢量表示，不会出现无穷大的情况；
- 对参数表示的曲线、曲面进行平移、比例、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，参数的系数几何意义明确，并提高了自由度，便于控制形状。

# 一些概念

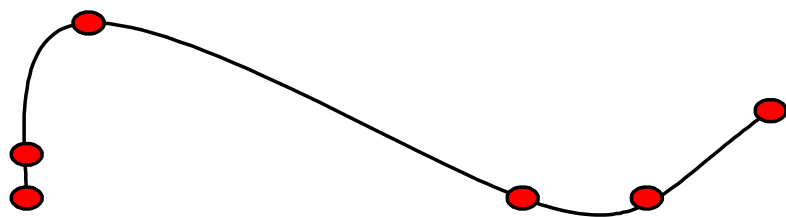
- 插值

给定一组有序的数据点 $P_i$ ,  $i=0, 1, \dots, n$ , 构造一条曲线**顺序通过这些数据点**, 称为对这些数据点进行插值, 所构造的曲线称为**插值曲线**。

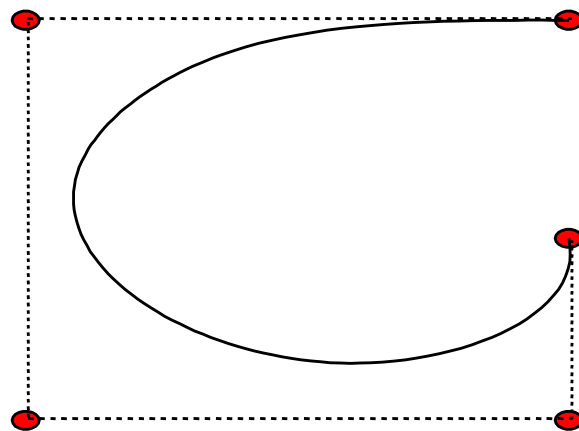
# 一些概念

## • 逼近

构造一条曲线使之在某种意义下**最接近**给定的数据点，称为对这些数据点进行**逼近**，所构造的曲线为**逼近曲线**。**插值**则称为**拟合**。

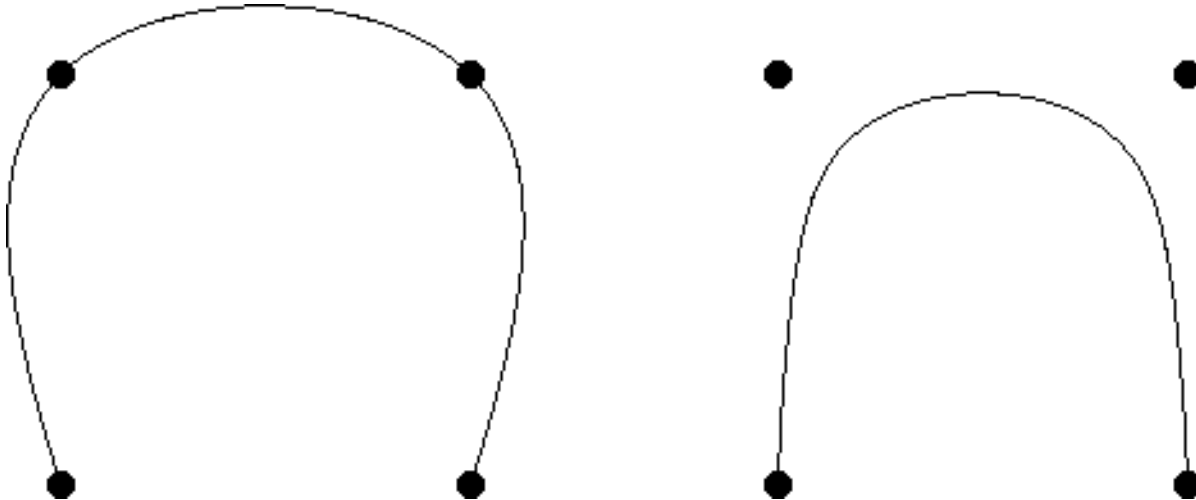


曲线的拟合



曲线的逼近

曲线/曲面的**拟合**：当用一组型值点来指定曲线/曲面的形状时，形状完全通过给定的型值点列。



曲线/曲面的**逼近**：当用一组控制点来指定曲线/曲面的形状时，求出的形状不必通过控制点列。但受其影响。



## 2.2 曲线与曲面

2.2.1 基础知识

2.2.2 Bézier曲线/曲面

2.2.3 B样条曲线/曲面

2.2.4 NURBS曲线/曲面

2.2.5 Coons曲面

2.2.6 三维物体的描述方法

## 2.2.2 Bézier 曲线/曲面

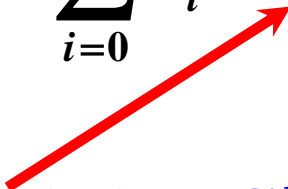
Bezier 曲线

# Bezier曲线生成

- **Bezier定义**

- 在空间给定 $n+1$ 个点 $P_0, P_1, P_2, \dots, P_n$ , 称下列参数曲线为 $n$ 次的Bezier曲线

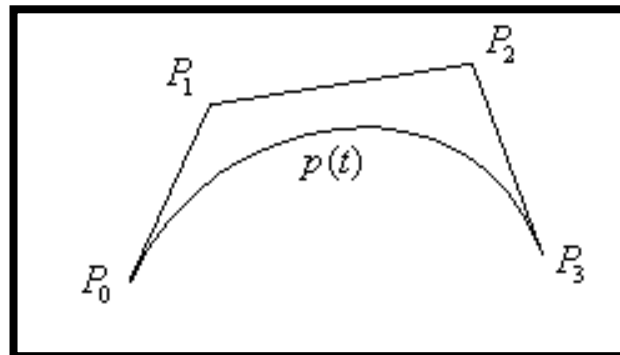
$$P(t) = \sum_{i=0}^n P_i * B_{i,n}(t), \quad 0 \leq t \leq 1$$


$$B_{i,n}(t) = C_n^i t^i (1-t)^{n-i}, \quad t \in [0, 1]$$

$$C_n^i = \frac{n!}{i!(n-i)!}$$

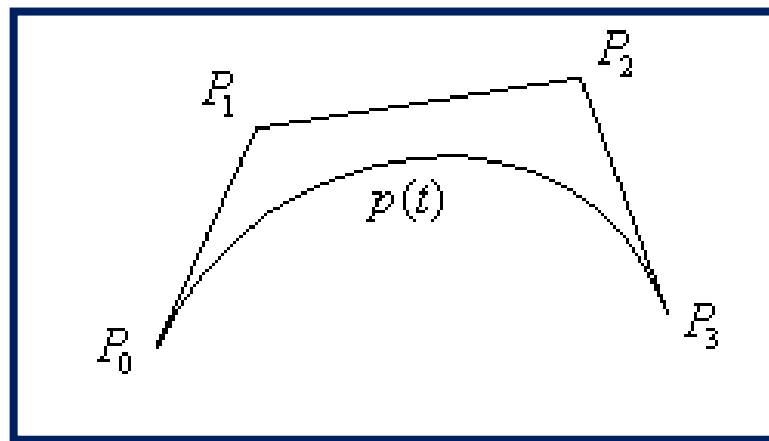
$$P(t) = \sum_{i=0}^n P_i * B_{i,n}(t), \quad 0 \leq t \leq 1$$

- 一般称折线 $P_0P_1P_2...P_n$ 为 $P(t)$ 的控制多边形；称 $P_0, P_1, P_2, ..., P_n$ 各点为 $P(t)$ 的**控制顶点**。
- Bezier曲线 $P(t)$ 与其控制多边形的关系可以这样认为：**控制多边形** $P_0P_1P_2...P_n$ 是 $P(t)$ 的大致形状的勾画； **$P(t)$** 是对 $P_0P_1P_2...P_n$ 的**逼近**



# Bezier曲线的特点

- 曲线的起点与终点和特征多边形的起点与终点重合，且多边形的第一条边和最后一条边表示了曲线在起点和终点处的切矢方向。而且由于曲线的形状趋向于控制多边形的形状，所以可以通过调整顶点的位置来控制曲线的形状。



# Bezier曲线的性质

- 端点特性

$$\text{当 } t = 0 \text{ 时, } P(0) = \frac{n!}{1 * n!} * 0^0 * (1 - 0)^n * P_0 = P_0$$

$$\text{当 } t = 1 \text{ 时, } P(1) = P_n$$

- 对称性

控制顶点次序颠倒，则新的曲线与原曲线的形状相同，仅曲线的走向相反。

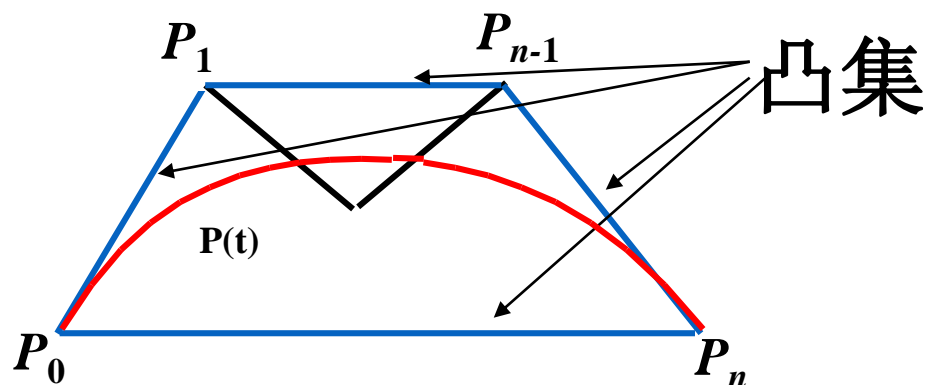
$$P(t) = \sum_{i=0}^n P_i * B_{i,n}(t), \quad 0 \leq t \leq 1$$

# Bezier曲线的性质（续）

- 凸包性

点集的凸包是指包含这些点的最小凸集

Bézier曲线位于其控制顶点的凸包之内



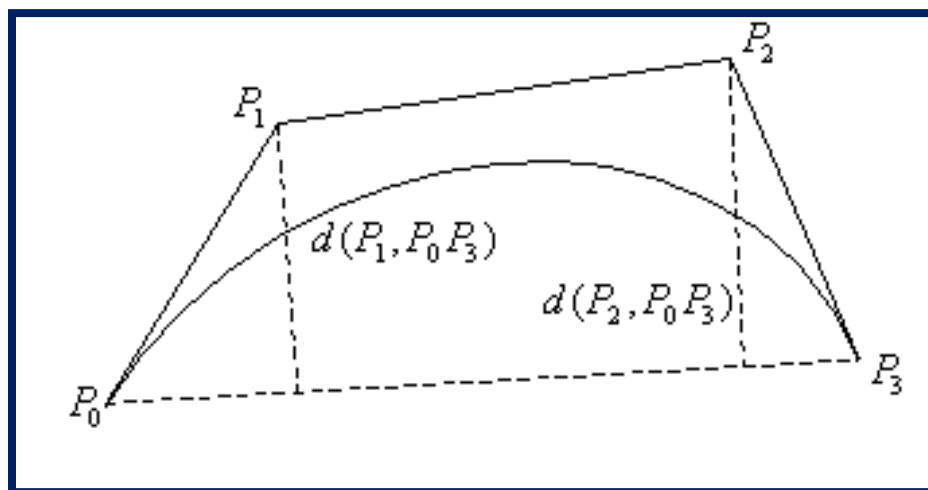
Bézier曲线的凸包性



# Bezier曲线的性质（续）

由于计算曲线与控制多边形端点连线比较麻烦，我们利用Bezier曲线的凸包性质得到：

- $d(P(t), P_0P_n) \leq \max(d(P_1, P_0P_n), \dots, d(P_{n-1}, P_0P_n))$



# Bezier曲线的性质（续）

- 几何不变性

曲线的形状仅与其控制顶点有关，而与具体坐标系的选择无关。

- 曲线的可分割性

采用德卡斯特里奥算法进行分割

如何进行分割

# Bezier 曲线分割

- 分割定理

- 一条多项式曲线被分割成两段，得到的两段曲线仍是多项式曲线

$$P=Q+R$$

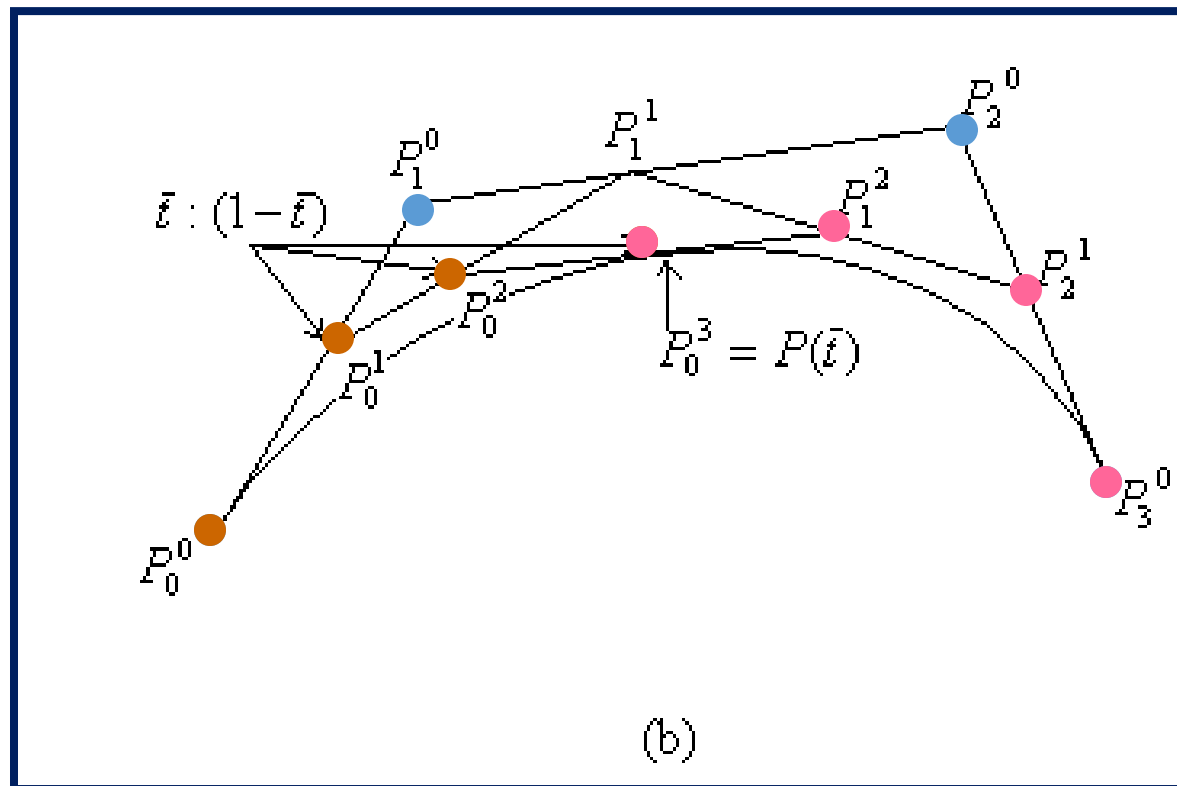
$$\left\{ \begin{array}{l} Q : P(t) = \sum_{i=0}^n P_i \cdot B_{i,n}(t) = \sum_{i=0}^n P_0^i \cdot B_{i,n}\left(\frac{t}{\bar{t}}\right) \quad t \in [0, \bar{t}] \\ R : P(t) = \sum_{i=0}^n P_i \cdot B_{i,n}(t) = \sum_{i=0}^n P_i^{n-i} \cdot B_{i,n}\left(\frac{t-\bar{t}}{1-\bar{t}}\right) \quad t \in [\bar{t}, 1] \end{array} \right.$$

# Bezier曲线分割（续）

- 将分割成的两段曲线继续不断地分割下去，所生成的控制顶点序列将收敛于曲线 $P(t)$ ，这一点称为Bezier曲线的收敛性。
- Bezier曲线具有的收敛性质，保证了在适当次数的分割之后，分得的每一段曲线都能由其两端点的连线所代替。所以可以利用Bezier曲线的分割性和收敛性来生成Bezier曲线。

# Bezier曲线分割（续）

- 几何解释



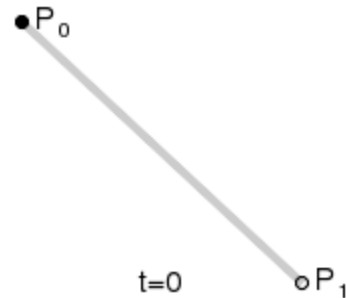
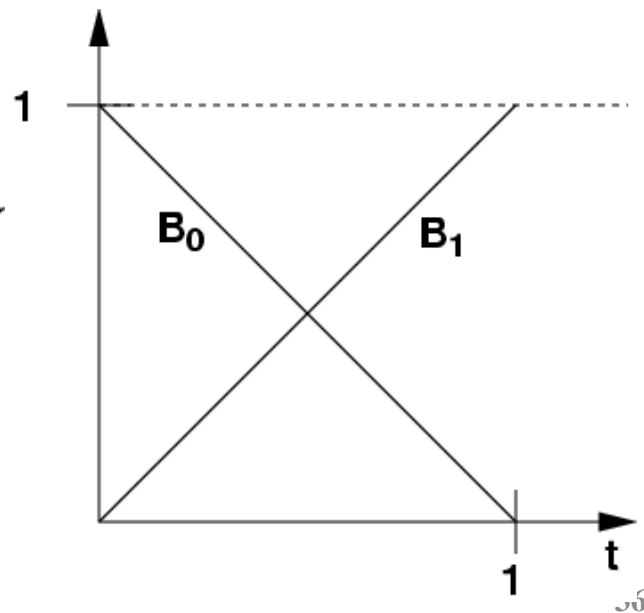
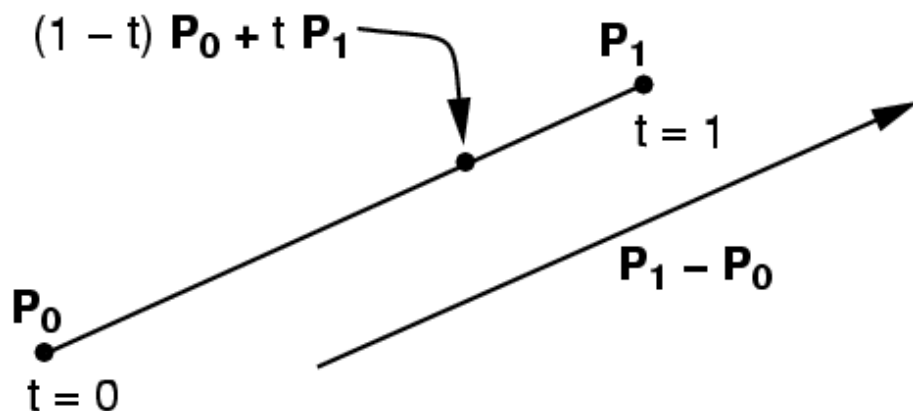
# 一次Bezier曲线

- $P_0, P_1$

$$P = (1 - t)P_0 + tP_1, \quad (0 \leq t \leq 1)$$

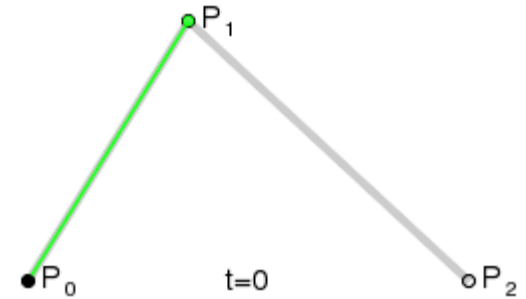
- $t=0$

- $t=1$



# 二次Bezier曲线

- $P_0, P_1, P_2$



$$1^2 = (t + (1 - t))^2$$

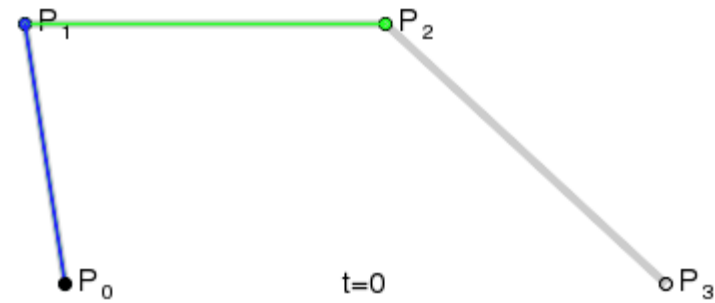
$$1 = t^2 + 2 * t * (1 - t) + (1 - t) * (1 - t),$$

$$(0 \leq t \leq 1)$$

$$P = (1 - t)^2 P_0 + 2 * (1 - t) t P_1 + t^2 P_2 \quad (0 \leq i \leq 1)$$

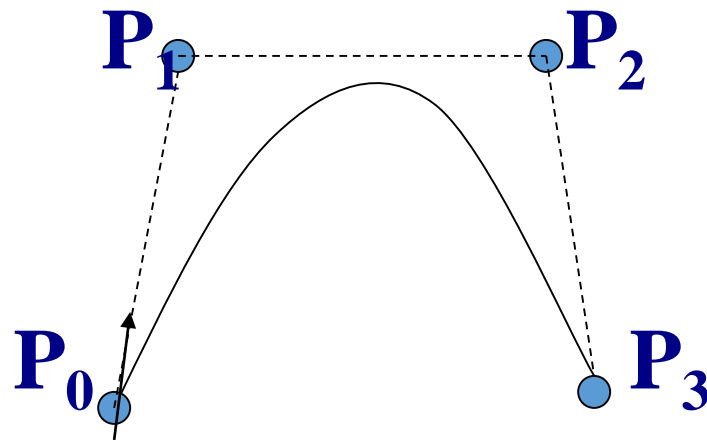


# 三次Bezier曲线



- $1^3 = (t + (1 - t))^3$   
 »  $1 = (t + (1 - t)) \cdot (t^2 + 2t(1 - t) + (1 - t)^2)$   
 »  $1 = t^3 + 3t^2(1 - t) + 3t(1 - t)^2 + (1 - t)^3$

- $B_1(t) = t^3$   
 $B_2(t) = 3t^2(1 - t)$   
 $B_3(t) = 3t(1 - t)^2$   
 $B_4(t) = (1 - t)^3$



$$P(t) = (1-t)^3 P_1 + 3t(1-t)^2 P_2 + 3t^2(1-t) P_3 + t^3 P_4$$

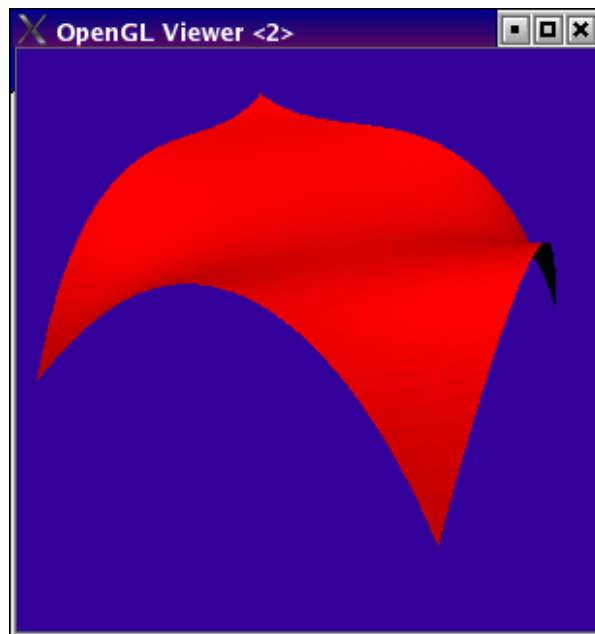
# Bezier曲线的矩阵表示

$$P(t) = (t^2 \ t \ 1) \begin{pmatrix} 1 & -2 & 1 \\ -2 & 2 & 0 \\ 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} P_0 \\ P_1 \\ P_2 \end{pmatrix}$$

$$P(t) = (t^3 \ t^2 \ t \ 1) \begin{pmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{pmatrix}$$

## 2.2.2 Bézier 曲线/曲面

### Bezier 曲面



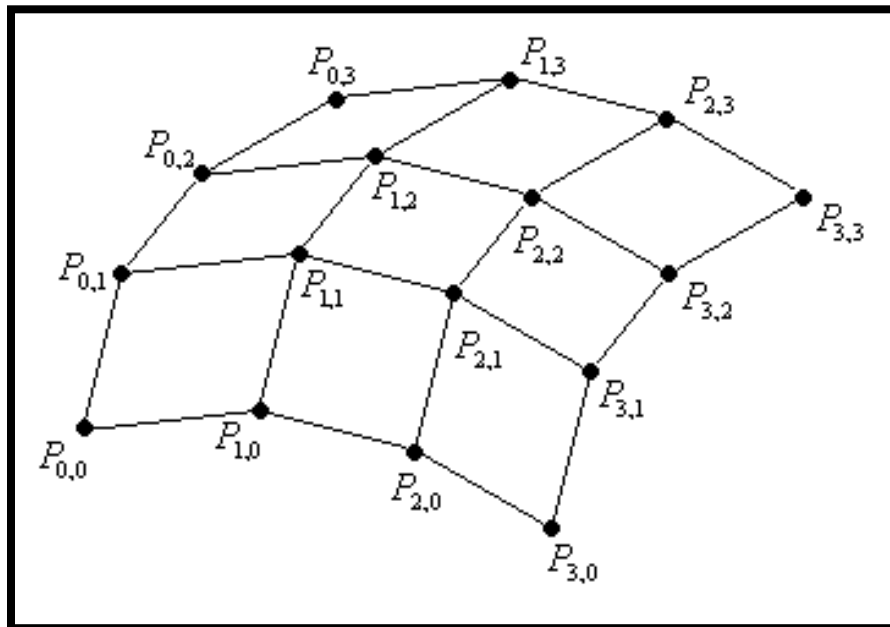
# Bezier曲面

- Bezier曲面的定义

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

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

- 控制点  $P_{i,j}$
- 控制网格  $\{P_{i,j}\}_{i,j=0}^{m,n}$



# Bezier 曲面

- Bezier曲面的定义

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

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

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

# Bezier曲面

- Bezier曲面的性质

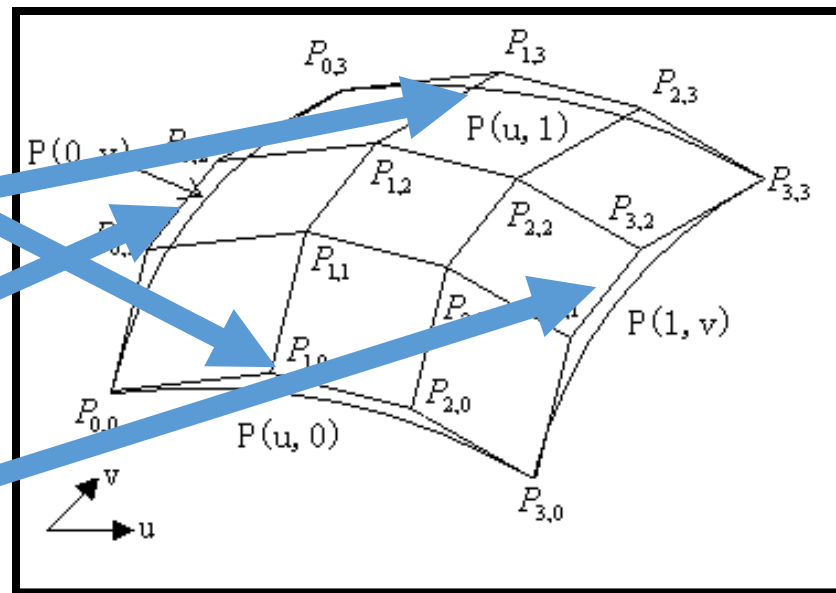
- 边界线

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

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

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

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



# Bezier 曲面

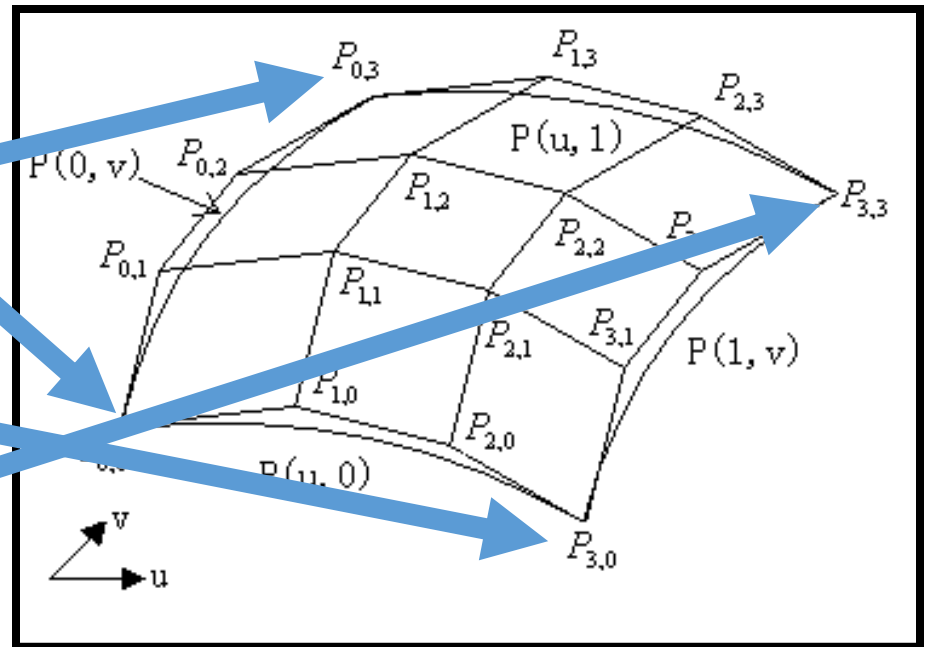
- 角点位置

$$P(0,0) = P_{0,0}$$

$$P(0,1) = P_{0,n}$$

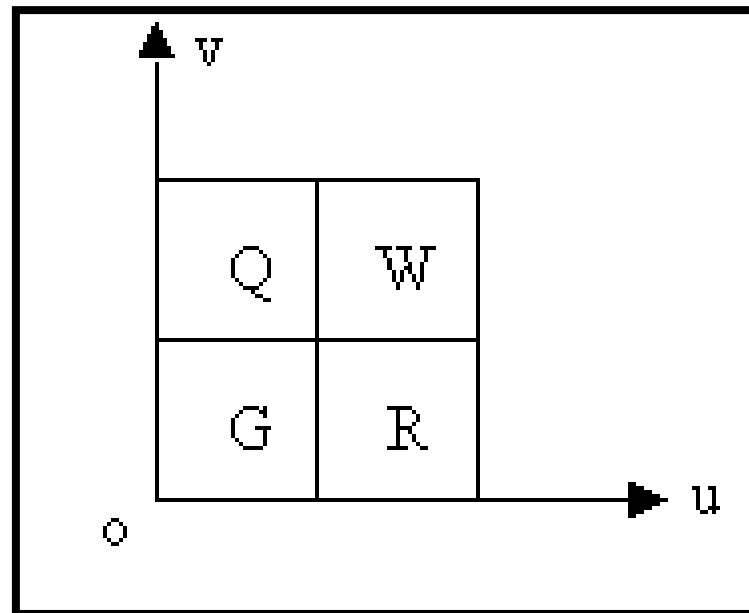
$$P(1,0) = P_{m,0}$$

$$P(1,1) = P_{m,n}$$



# Bezier 曲面

$$\begin{cases} P_{i,j}^{0,0} = P_{i,j} & i = 0, 1, \dots, m, j = 0, 1, \dots, n \\ P_{i,j}^{r,s} = (1-u_0)P_{i,j}^{r-1,s} + u_0P_{i+1,j}^{r-1,s} & r = 1, \dots, m, i = 0, 1, \dots, m-r \\ P_{i,j}^{r,s} = (1-v_0)P_{i,j}^{r,s-1} + v_0P_{i,j-1}^{r,s-1} & s = 1, \dots, n, j = 0, 1, \dots, n-s \end{cases}$$





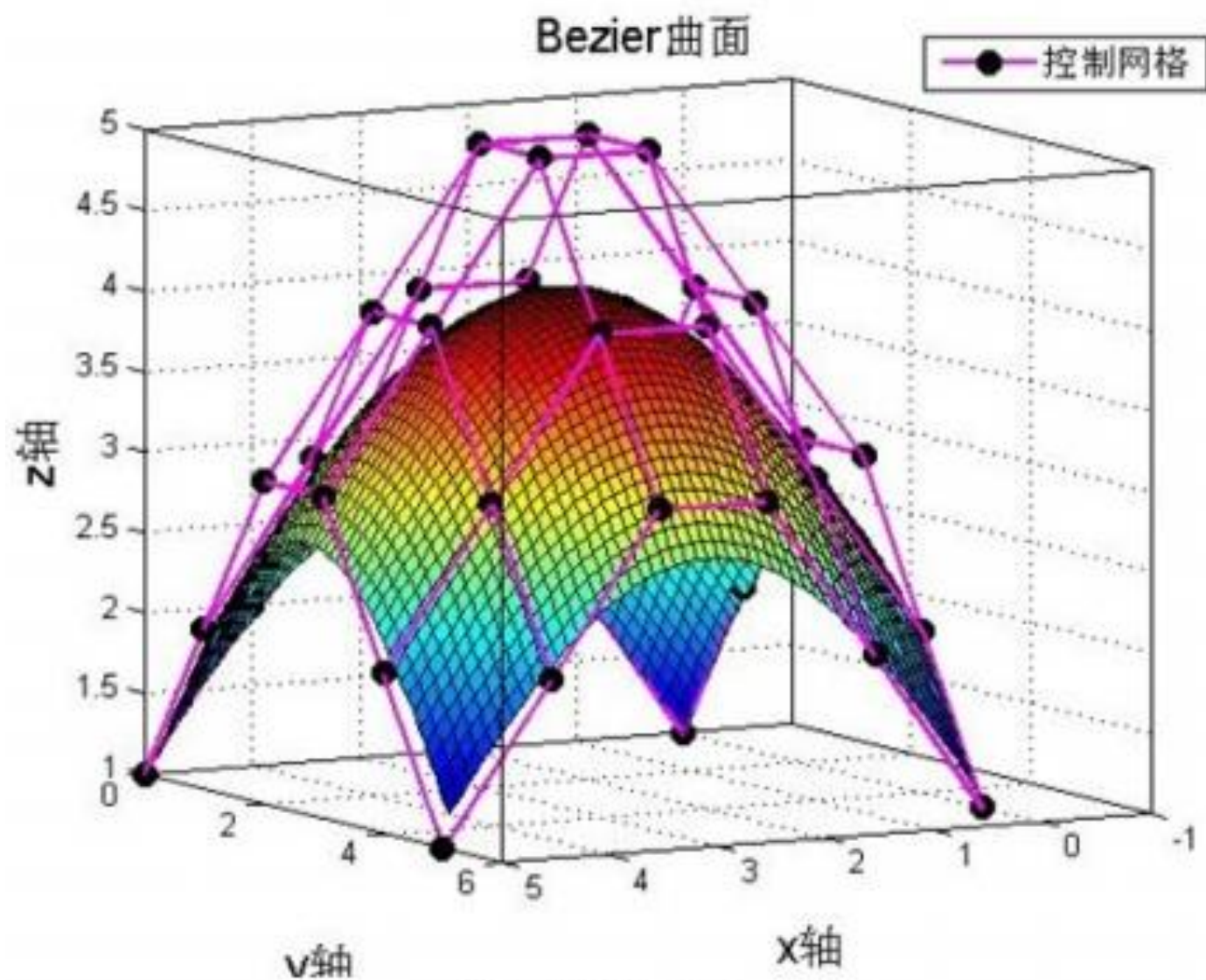


图 3-1 双五次 Bezier 曲面

## 2.2 曲线与曲面

2.2.1 基础知识

2.2.2 B é zier曲线/曲面

2.2.3 B样条曲线/曲面

2.2.4 NURBS曲线/曲面

2.2.5 Coons曲面

2.2.6 三维物体的描述方法

## 2.2.3 B样条曲线/曲面

### B样条曲线

# B样条曲线

## 1. 从 Bezier 曲线到 B 样条曲线

Bezier 曲线在应用中的不足:

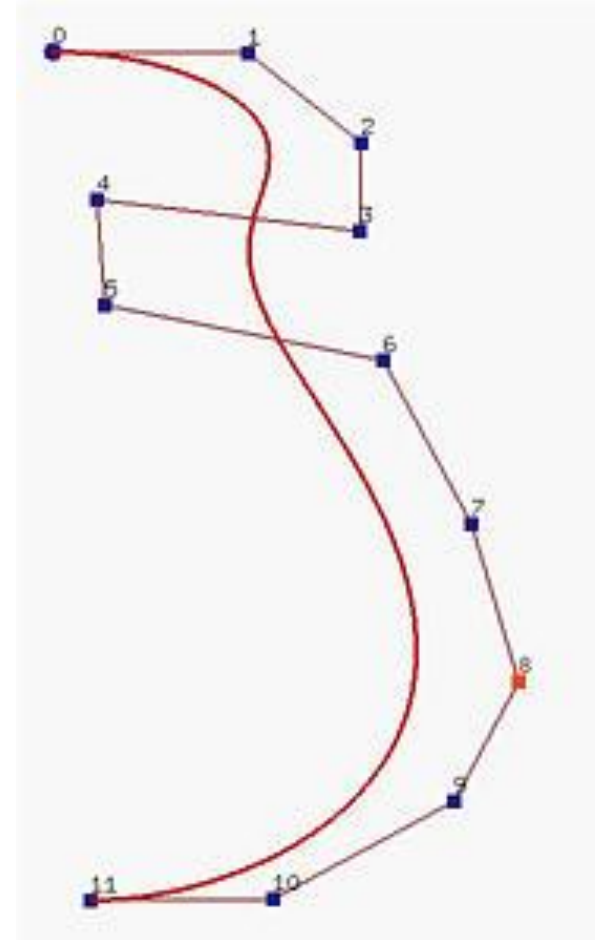
**缺乏灵活性** 一旦确定了特征多边形的顶点数( $m$ 个), 也就决定了曲线的阶次( $m-1$ 次), 无法更改;

**控制性差** 当顶点数较多时, 曲线的阶次将较高, 此时, 特征多边形对曲线形状的控制将明显减弱;

# B-Spline

## ■原因 (对比Bezier曲线)

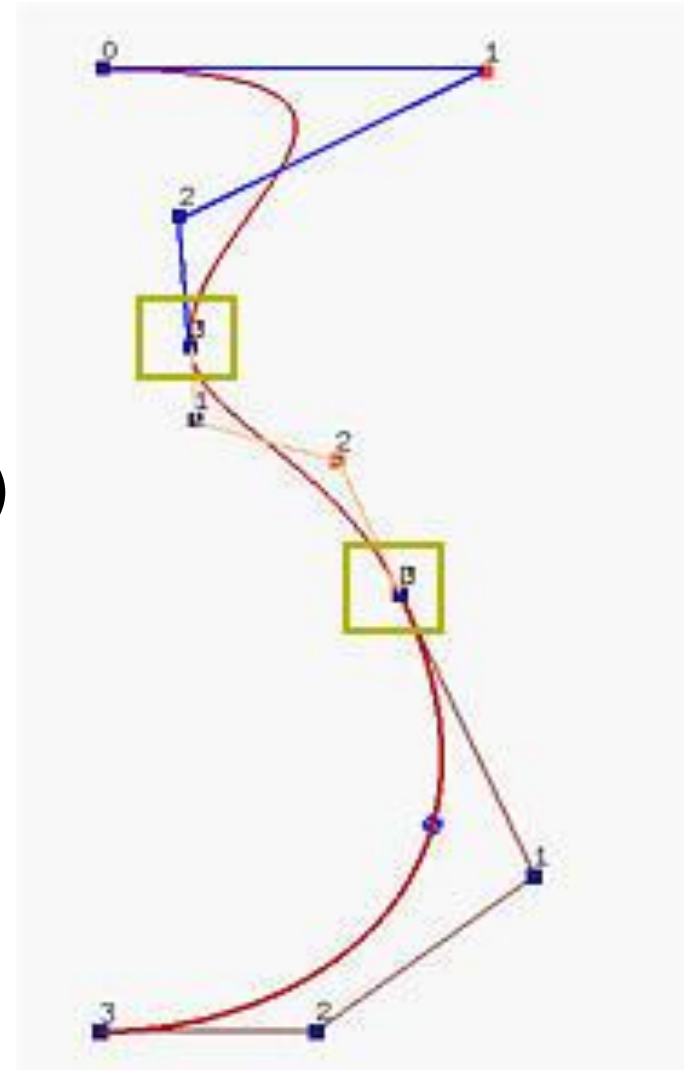
- Bezier 曲线的次数由控制点决定
- 例如 $P_4P_5$ 线段不易弯曲.



# B-Spline

## ■ 复杂曲线如何用Bezier曲线表示

- 多段Bezier
- 连接点的光滑(导数相同)比较难

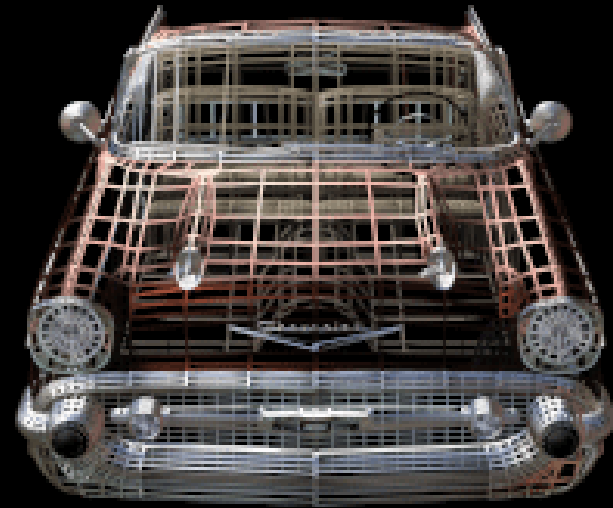
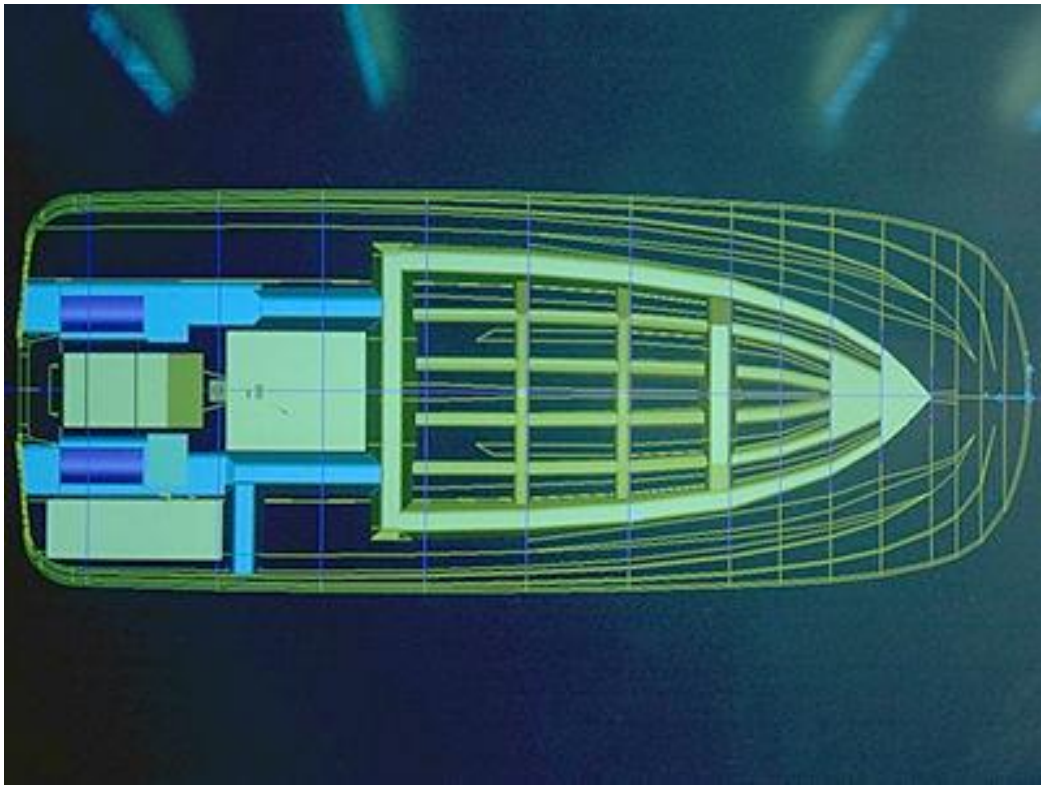


# B-Spline

为了克服 Bezier 曲线存在的问题，Gordon 等人拓展了 Bezier 曲线，就外形设计的需求出发，希望新的曲线

要：  
易于进行局部修改；  
更逼近特征多边形；  
是低阶次曲线。

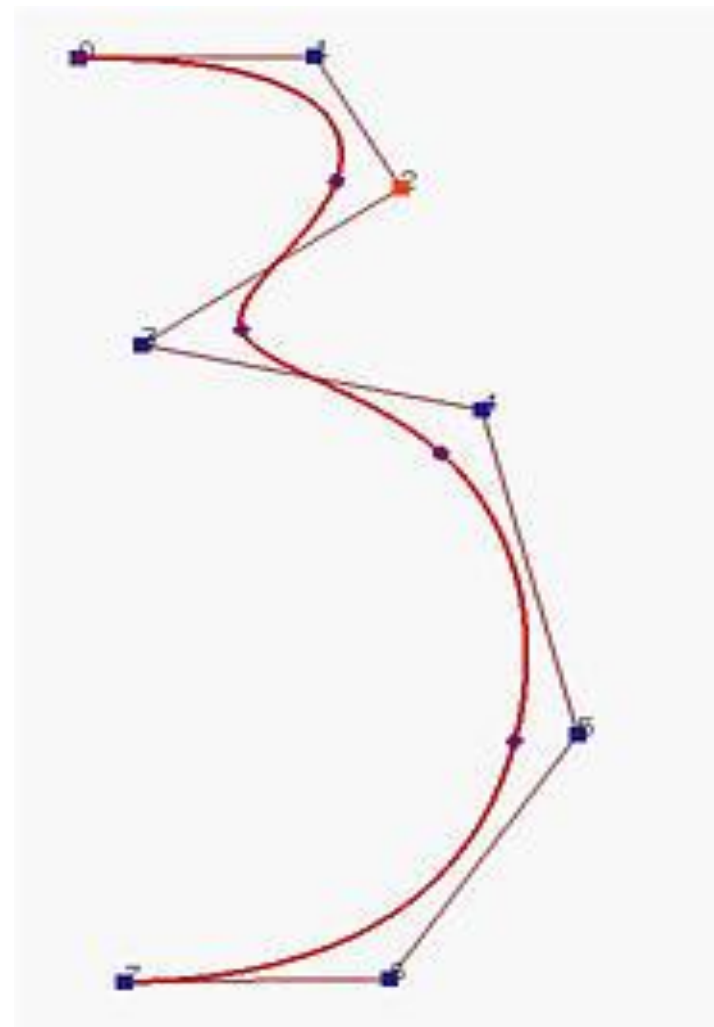
于是，用  $n$  次 B 样条基函数替换了伯恩斯坦基函数，构造了称之为 B 样条曲线的新型曲线。





# B-Spline

- 灵活
- 次数和控制点个数无关
- 右图 8个控制点，3次



## 2. B 样条曲线的数学表达式

B 样条曲线的数学表达式为：

$$P(t) = \sum_{i=0}^n P_i \cdot B_{i,k}(t)$$

$$B_{i,1}(t) = \begin{cases} 1, & t_i \leq t < t_{i+1} \\ 0, & \text{其他} \end{cases}$$

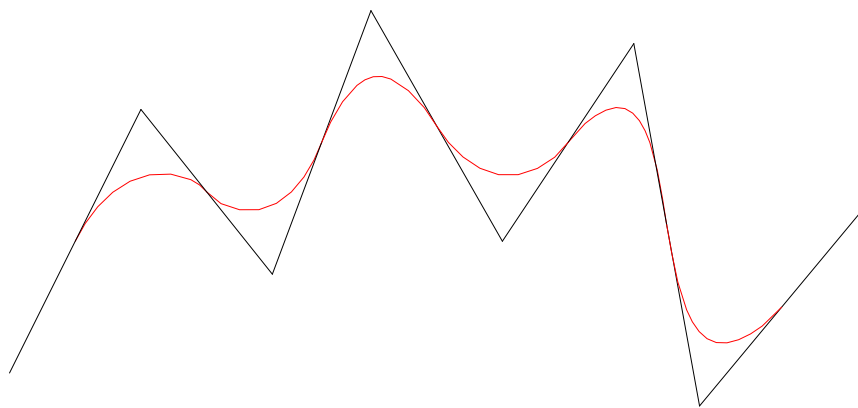
$$B_{i,k}(t) = \frac{t - t_i}{t_{i+k-1} - t_i} B_{i,k-1}(t) + \frac{t_{i+k} - t}{t_{i+k} - t_{i+1}} B_{i+1,k-1}(t), \quad -\infty < t < +\infty$$

$$B_{i,1}(t) = \begin{cases} 1, & t_i \leq t < t_{i+1} \\ 0, & \text{其他} \end{cases}$$

$$B_{i,k}(t) = \frac{t - t_i}{t_{i+k-1} - t_i} B_{i,k-1}(t) + \frac{t_{i+k} - t}{t_{i+k} - t_{i+1}} B_{i+1,k-1}(t), \quad -\infty < t < +\infty$$

- 给定参数  $t$  轴上的一个分割  $t_i$  ( $t_i \leq t_{i+1}$ ,  $i=0, \pm 1, \pm 2 \cdots$ )。由递推关系所定义的  $B_{i,k}(t)$  称为  $k$  阶(或  $k-1$  次) B 样条基函数。并约定  $0/0=0$ 。

在上式中,  $0 \leq t \leq 1$ ;  $i = 0, 1, 2, \dots, m$  所以可以看出: B样条曲线是分段定义的。如果给定  $m+n+1$  个顶点  $P_i$  ( $i=0, 1, 2, \dots, m+n$ ), 则可定义  $m+1$  段  $n$  次的参数曲线。



## 一次均匀B样条曲线

- 空间 $n+1$ 个顶点 ( $i = 0, 1, \dots, n$ ) 定义 $n$ 段一次 ( $k=1$ , 二阶) 均匀B样条曲线, 即每相邻两个点可构造一曲线段 $P_i(u)$ , 其定义表达为:

$$P_i(u) = \begin{bmatrix} u & 1 \end{bmatrix} \begin{bmatrix} -1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{P}_{i-1} \\ \mathbf{P}_i \end{bmatrix} \quad i = 1, \dots, n; \quad 0 \leq u \leq 1$$

$$\begin{aligned} &= (1-u) * \mathbf{P}_{i-1} + u * \mathbf{P}_i \\ &= B_{0,1}(u) * \mathbf{P}_{i-1} + B_{1,1}(u) * \mathbf{P}_i \end{aligned}$$

## 二次 B 样条曲线

在二次 B 样条曲线中， $n=2$ ， $k=0,1,2$   
其基函数形式为：

$$B_{0,2}(t) = \frac{1}{2}(t-1)^2$$

$$B_{1,2}(t) = \frac{1}{2}(-2t^2 + 2t + 1)$$

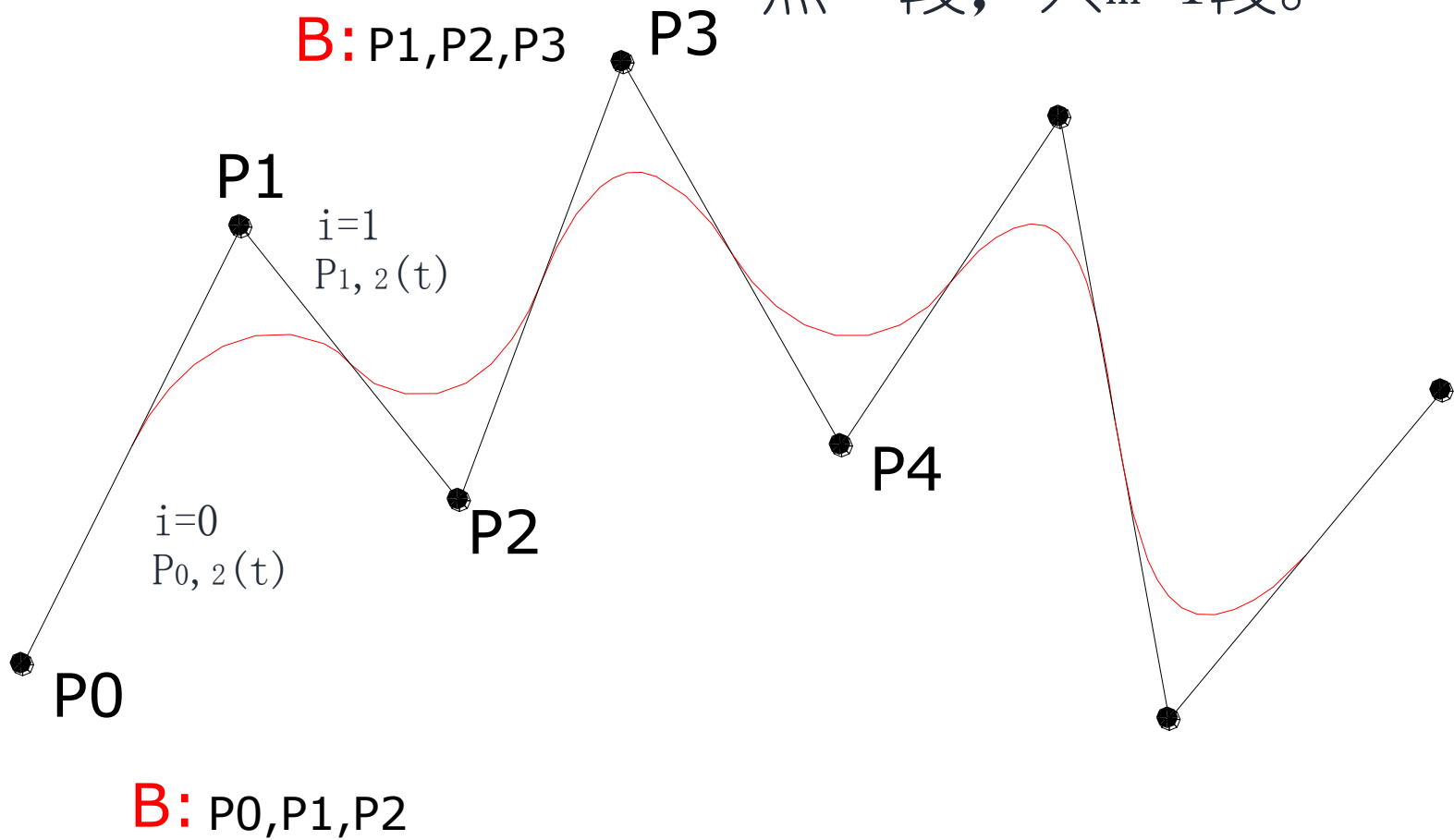
$$B_{2,2}(t) = \frac{1}{2}t^2$$

有了基函数，因此可写出二次 B 样条曲线的分段表达式为：

$$P_i(t) = B_{0,2}(t) \cdot P_i + B_{1,2}(t) \cdot P_{i+1} + B_{2,2}(t) \cdot P_{i+2}$$

(  $i = 0, 1, 2, \dots, m$  )     $m+1$  段

$n=2$ , 二次B样条曲线  
 $m+n+1$ 个顶点, 三  
点一段, 共 $m+1$ 段。





## 三次 B 样条曲线

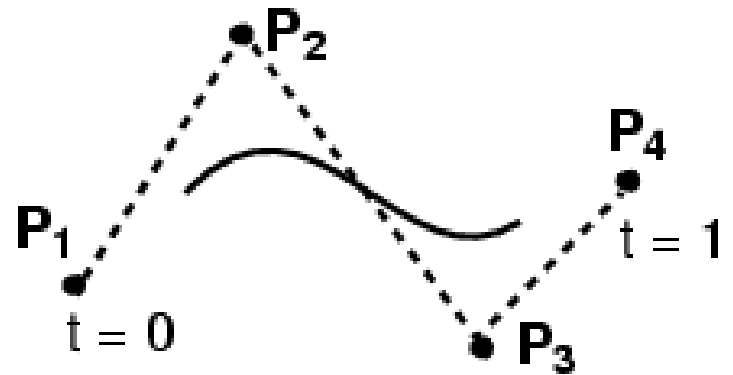
分段三次 B 样条曲线由相邻四个顶点定义，  
其表达式为：

$$P(t) = B_{0,3}(t) \cdot P_0 + B_{1,3}(t) \cdot P_1 + B_{2,3}(t) \cdot P_2 + B_{3,3}(t) \cdot P_3 \\ (0 < t < 1)$$

可见，由  $n$  个顶点定义的完整的三次 B 样条曲线  
是由  $n-3$  段分段曲线连接而成的。

- 当参数均匀分割时，空间  $n+1$  个顶点 ( $i=0,1,2,3\cdots n$ ) 定义的三次B样条曲线称为三次均匀B样条曲线。由相邻四个顶点定义的三次均匀B样条曲线及其性质。

- $B_{0,4}(u) = 1/6 (1-t)^3$
- $B_{1,4}(u) = 1/6 (3t^3 - 6t^2 + 4)$
- $B_{2,4}(u) = 1/6 (-3t^3 + 3t^2 + 3t + 1)$
- $B_{3,4}(u) = 1/6 t^3$



# 三次 B 样条曲线

$$Q(t) = \frac{(1-t)^3}{6}P_{i-3} + \frac{3t^3 - 6t^2 + 4}{6}P_{i-2} + \frac{-3t^3 + 3t^2 + 3t + 1}{6}P_{i-1} + \frac{t^3}{6}P_i$$

$$\mathbf{P}(t) = \sum_{i=j-3}^j \mathbf{P}_i B_{i-j+3}(t)$$

$$B_{B-Spline} = \frac{1}{6} \begin{pmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{pmatrix}$$

# B样条曲线的计算

可采用不同的方法计算B样条曲线上任何一点的值。

de Boor算法的描述如下：

将 $t$ 固定在区间  $[t_j, t_{j+1})$  ( $k-1 \leq j \leq n$ )上，将 $P(t)$  简化如下：

$$\begin{aligned} P(t) &= \sum_{i=0}^n P_i B_{i,k}(t) = \sum_{i=j-k+1}^j P_i B_{i,k}(t) \\ &= \sum_{i=j-k+1}^j P_i \left[ \frac{t-t_i}{t_{i+k-1}-t_i} B_{i,k-1}(t) + \frac{t_{i+k}-t}{t_{i+k}-t_{i+1}} B_{i+1,k-1}(t) \right] \\ &= \sum_{i=j-k+2}^j \left[ \frac{t_{i+k-1}-t}{t_{i+k-1}-t_i} P_{i-1} + \frac{t-t_i}{t_{i+k-1}-t_i} P_i \right] B_{i,k-1}(t) \quad t_j \leq t < t_{j+1} \end{aligned}$$

$$\mathbf{P}_i^{[l]}(t) = \begin{cases} \mathbf{P}_i, & l = 0, \quad i = j - k + 1, j - k + 2, \dots, j \\ \frac{t_{i+k-l} - t}{t_{i+k-l} - t_i} \mathbf{P}_{i-1}^{[l-1]}(t) + \frac{t - t_i}{t_{i+k-l} - t_i} \mathbf{P}_i^{[l-1]}(t) \\ l = 1, 2, \dots, k - 1; \quad i = j - k + l + 1, \dots, j \end{cases}$$

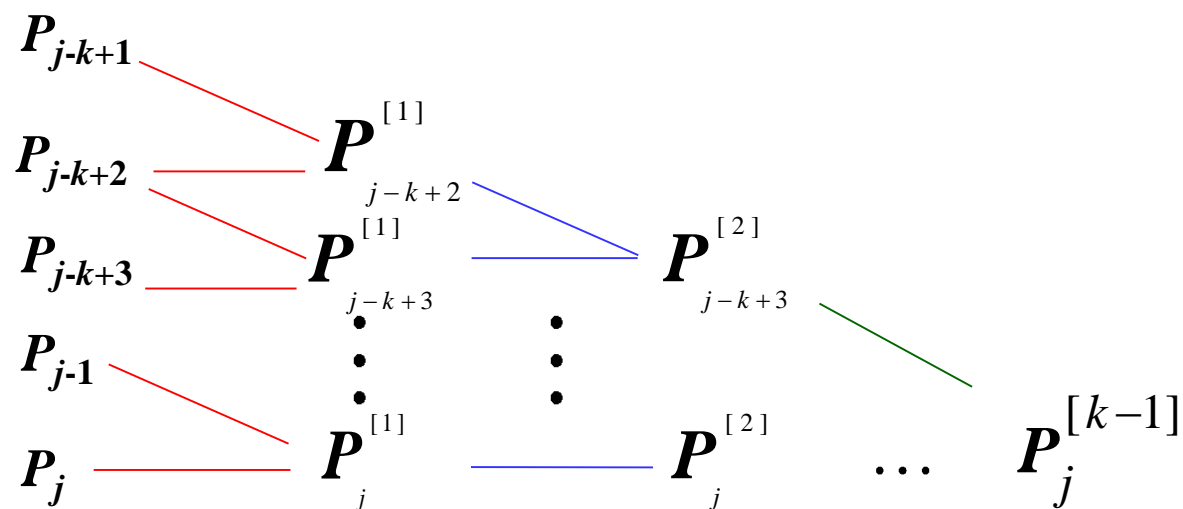
则可表示成

$$\mathbf{P}(t) = \sum_{i=j-k+2}^j \mathbf{P}_i^{[l]}(t) B_{i,k-1}(t)$$

上式将同一条曲线  $\mathbf{P}(t)$  由  $k$  阶 B 样条表示成  $k-1$  阶 B 样条。反复运用此公式，最终将得到

$$\mathbf{P}(t) = \mathbf{P}_j^{[k-1]}(t)$$

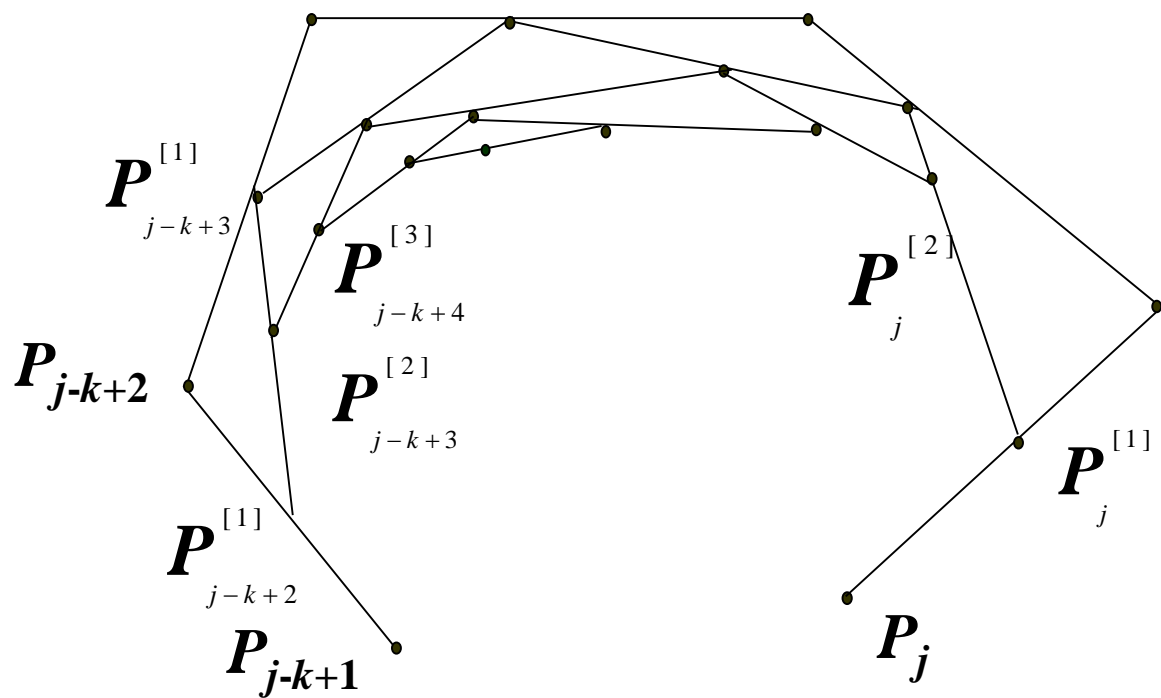
于是, $P(t)$ 的值可以通过de Boor算法的递推关系求得。下图反映的是用de Boor算法从 $P_0, P_1, P_2, \dots, P_n$ 得到 $P_j^{[k-1]}(t)$ 的递推过程。



de Boor算法的递推关系

de Boor算法的几何意义:

以线段 $P_i^{[r]}P_{i+1}^{[r]}$ 割去角 $P_i^{[r-1]}$ ，从多边形 $P_{j-k+1}P_{j-k+2}\cdots P_j$ 开始经过 $k-1$ 层的割角，最后得到 $P(t)$ 上的点 $P_j^{[k-1]}(t)$ 。这个割角过程比Bézier曲线的割角过程复杂。

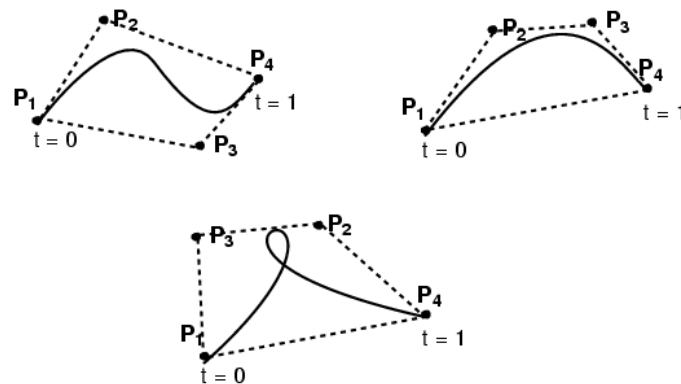
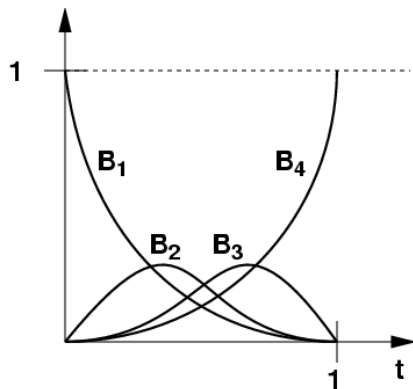


**B样条曲线的deBoor算法**

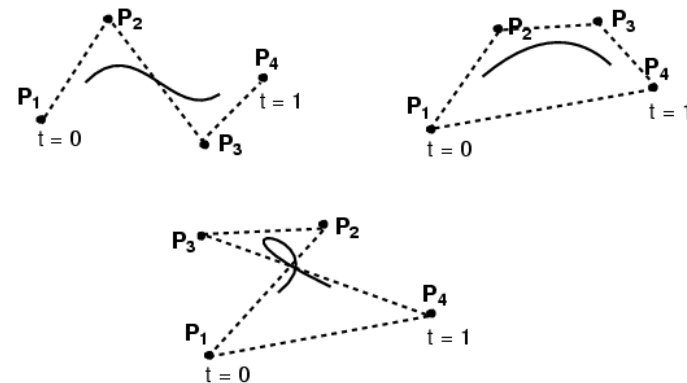
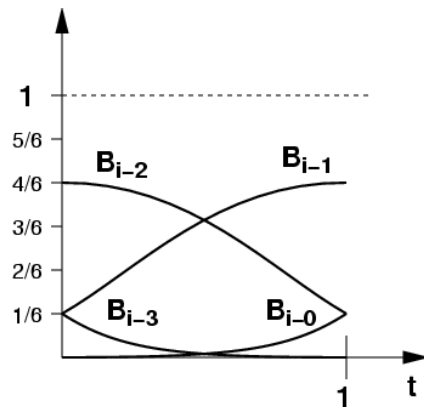


# 区别

**Bézier**



**BSpline**



# BSpline

B样条曲线是一种非常灵活的曲线，曲线的局部形状受相应顶点的控制很直观。这些顶点控制技术如果运用得好，可以使整个B样条曲线在某些部位满足一些特殊的技术要求。

## 2.2.3 B样条曲线/曲面

### B样条曲面

# B样条曲面

- B样条曲面可以通过一个特征多边形网格来定义，其数学表达式如下：

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

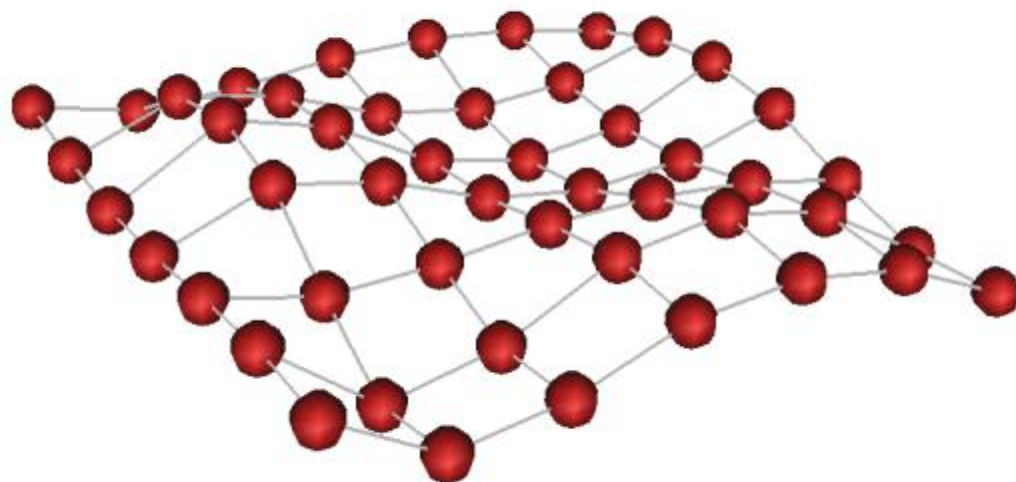
$$u_k \leq u \leq u_{n+1}, v_k \leq v \leq v_{m+1}$$

其中 $P_{ij}$ 是空间中给定的 $n \times m$ 个点； $B_{i,k}(u)$ ， $B_{j,h}(v)$ 分别是关于节点向量 $U, V$ 的 $k$ 阶和 $h$ 阶的B样条基函数。

## B样条曲面的优点

- B样条曲面除具有凸包性、保凸性、易交互性等优点外，还具有局部性，即如果变动其某一个控制顶点，曲面只有与其相关的一小部分发生变化，其余部分保持不变，这就为设计曲面时修改某一局部的形状带来了很大的便利。
- 目前，非均匀节点有理B样条已成为曲面造型中的标准。

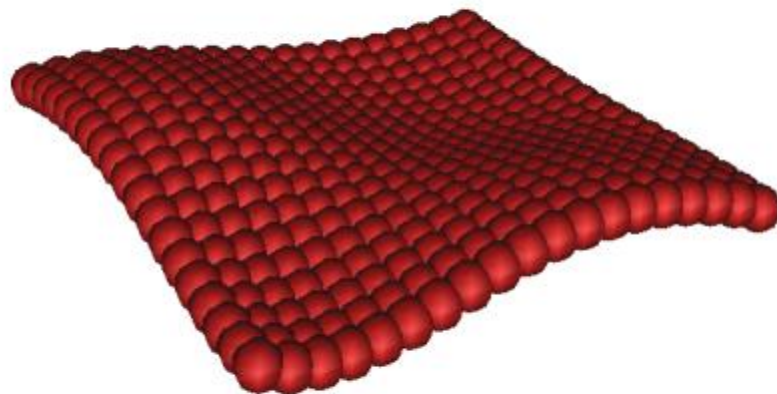
# B样条曲面



# B样条曲面



# B样条曲面





## 2.2 曲线与曲面

2.2.1 基础知识

2.2.2 B é zier曲线/曲面

2.2.3 B样条曲线/曲面

2.2.4 NURBS曲线/曲面

2.2.5 Coons曲面

2.2.6 三维物体的描述方法

## 2.2.4 NURBS曲线/曲面

NURBS (非均匀有理B样条)

# NURBS (非均匀有理B样条)

- B样条曲线可以按节点向量中节点的分布情况进行分类: 均匀B样条采用沿参数轴均匀分布, 非均匀B样条曲线的节点向量可以任意分布。

$$P(t) = \frac{\sum_{i=0}^n \omega_i P_i B_{i,k}(t)}{\sum_{i=0}^n \omega_i B_{i,k}(t)}$$

# 非均匀有理B样条曲线的特点

- B样条曲线所有优点
- 透视不变性
- 圆、球面的精确表示
- 更多的形状控制自由度

- 以三维空间中的NURBS曲线为例，设其控制顶点为  $P_i = [x_i \ y_i \ z_i]$ ，权因子为  $w_i$ ，则它是四维空间中以四维点  $[w_i P_i \ W_i] = [w_i x_i \ w_i y_i \ w_i z_i \ w_i]$  为控制顶点的B样条曲线在  $w=1$  的超平面上的中心投影，以上  $i=1, 2, \dots, n$ ，且两条曲线具有相同的节点向量。中心投影H定义如下：

$$H\{[X \ Y \ Z \ w]\} = [x \ y \ z] = \begin{bmatrix} \frac{X}{w} & \frac{Y}{w} & \frac{Z}{w} \end{bmatrix}$$

若 $w=0$ ，表示为无穷远点  $\begin{bmatrix} X & Y & Z \\ w & w & w \end{bmatrix}$

若 $w \neq 0$ ，可以得到NURBS曲线上点的计算方法：

- 构造与 $n$ 维空间中的NURBS曲线相应的 $n+1$ 维空间中的B样条曲线；
- 用deBoor算法求出 $n+1$ 维空间中的点 $[P \ w]$ ；
- 将此点中心投影到 $w=1$ 的超平面上，得到 $n$ 维空间中的相应点 $[P/w]$ 。

## 2.2.4 NURBS曲线/曲面

非均匀有理B样条(NURBS)曲面

# 非均匀有理B样条(NURBS)曲面

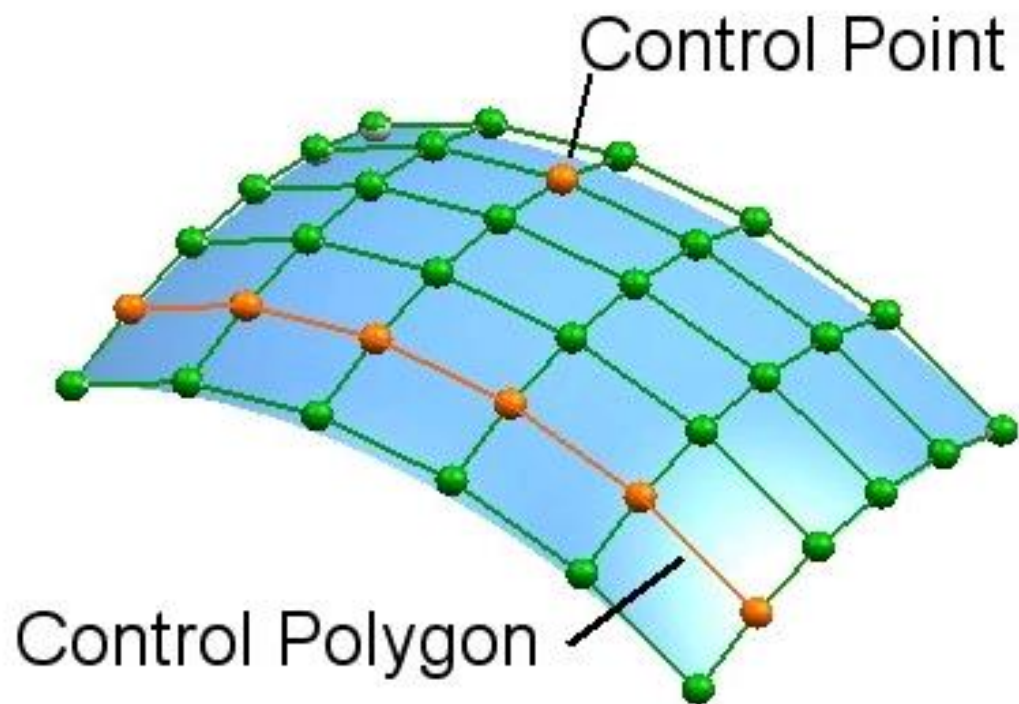
- 一张 $k \times h$ 阶的NURBS曲面定义如下:

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

其中 $P_{ij}$ 是空间中给定的 $n \times m$ 个点,  $W_{ij}$ 是与 $P_{ij}$ 相联系的权因子。节点向量 $U=\{u_i\}$ ,  $V=\{v_i\}$ , 分别是对参数 $uv$ 平面上的 $u$ 轴和 $v$ 轴的分割,  $B_{i,k}(u)$ ,  $B_{i,h}(v)$ 分别是关于节点向量 $U, V$ 的 $k$ 阶和 $h$ 阶的B样条基函数。



- NURBS曲面:



## 2.2 曲线与曲面

2.2.1 基础知识

2.2.2 B é zier曲线/曲面

2.2.3 B样条曲线/曲面

2.2.4 NURBS曲线/曲面

2.2.5 Coons曲面

2.2.6 三维物体的描述方法

## 2.2.5 Coons 曲面

孔斯 (Coons) 曲面

# 孔斯 (Coons) 曲面

- $H(u,v)$ 是在区域上给定的曲面，现要求作一曲面  $P(u,v)$ 使得

$$\left\{ \begin{array}{ll} P(u,v) = H(u,v), & u = 0,1, \quad v = 0,1 \\ P'_u(u,v) = H'_u(u,v), & u = 0,1, \quad v = 0,1 \\ P'_v(u,v) = H'_v(u,v), & u = 0,1, \quad v = 0,1 \\ P''_{uv}(u,v) = H''_{uv}(u,v), & u = 0,1, \quad v = 0,1 \end{array} \right.$$

- 双三次孔斯函数

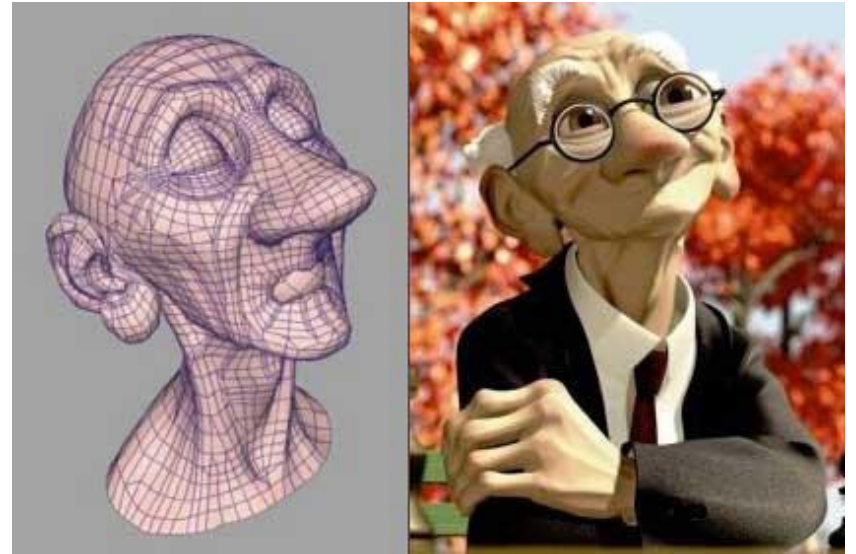
$$P(u, v) = [F_0(u), F_1(u), G_0(u), G_1(u)] C \begin{bmatrix} F_0(v) \\ F_1(v) \\ G_0(v) \\ G_1(v) \end{bmatrix}$$

$$C = \begin{bmatrix} H(0,0) & H(0,1) & H'_v(0,0) & H'_v(0,1) \\ H(1,0) & H(1,1) & H'_v(1,0) & H'_v(1,1) \\ H'_u(0,0) & H'_u(0,1) & H''_{uv}(0,0) & H''_{uv}(0,1) \\ H'_u(1,0) & H'_u(1,1) & H''_{uv}(1,0) & H''_{uv}(1,1) \end{bmatrix} \begin{cases} F_0(t) = 2t^3 - 3t^2 + 1 \\ F_1(t) = -2t^3 + 3t^2 \\ G_0(t) = t^3 - 2t^2 + t \\ G_1(t) = t^3 - t^2 \end{cases}$$

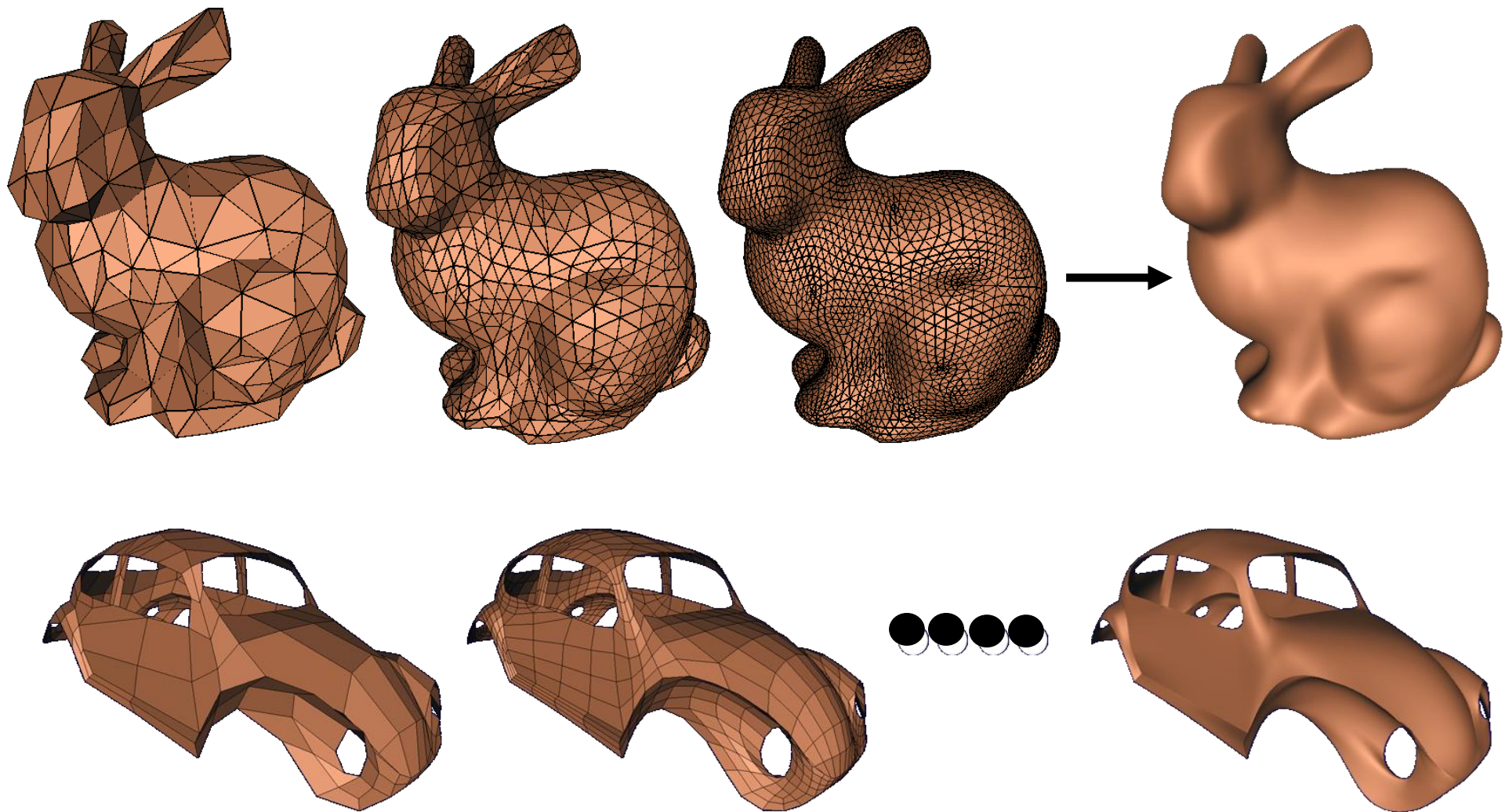
# 曲面细分



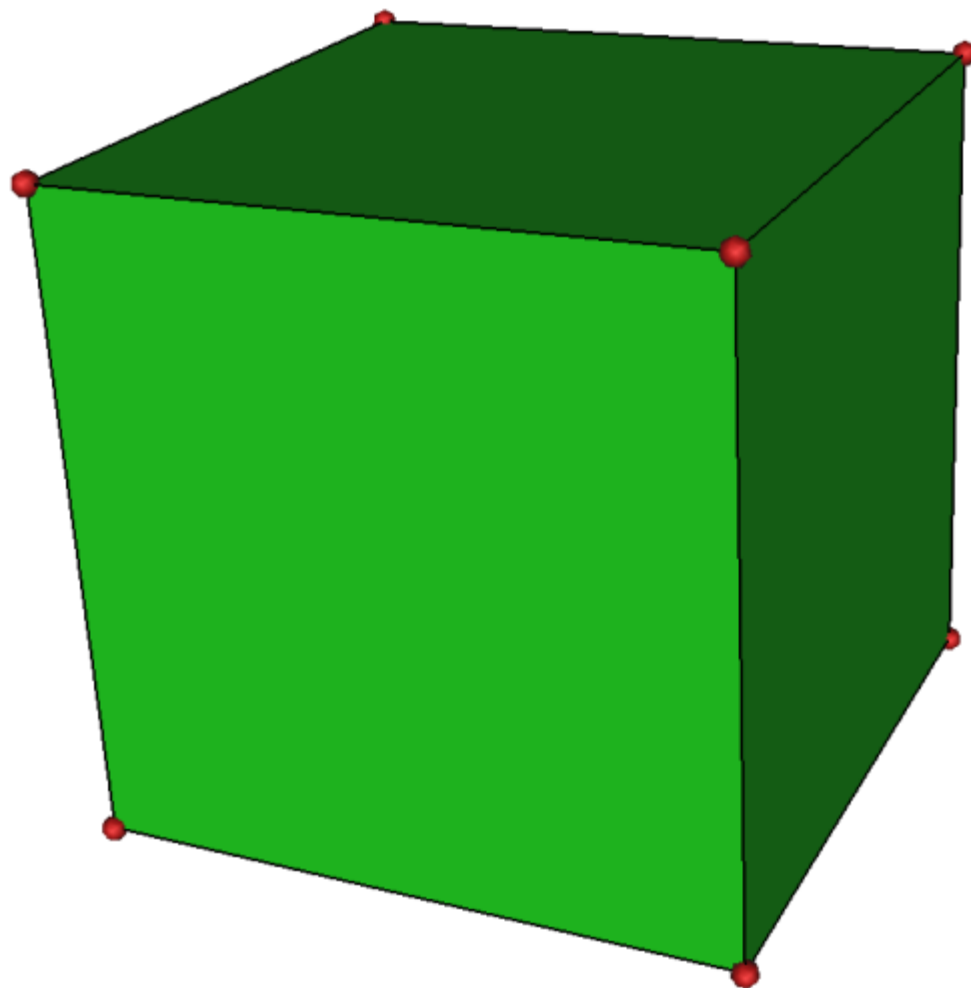
Toy Story © Disney / Pixar



Geri's Game © Pixar Animation Studios

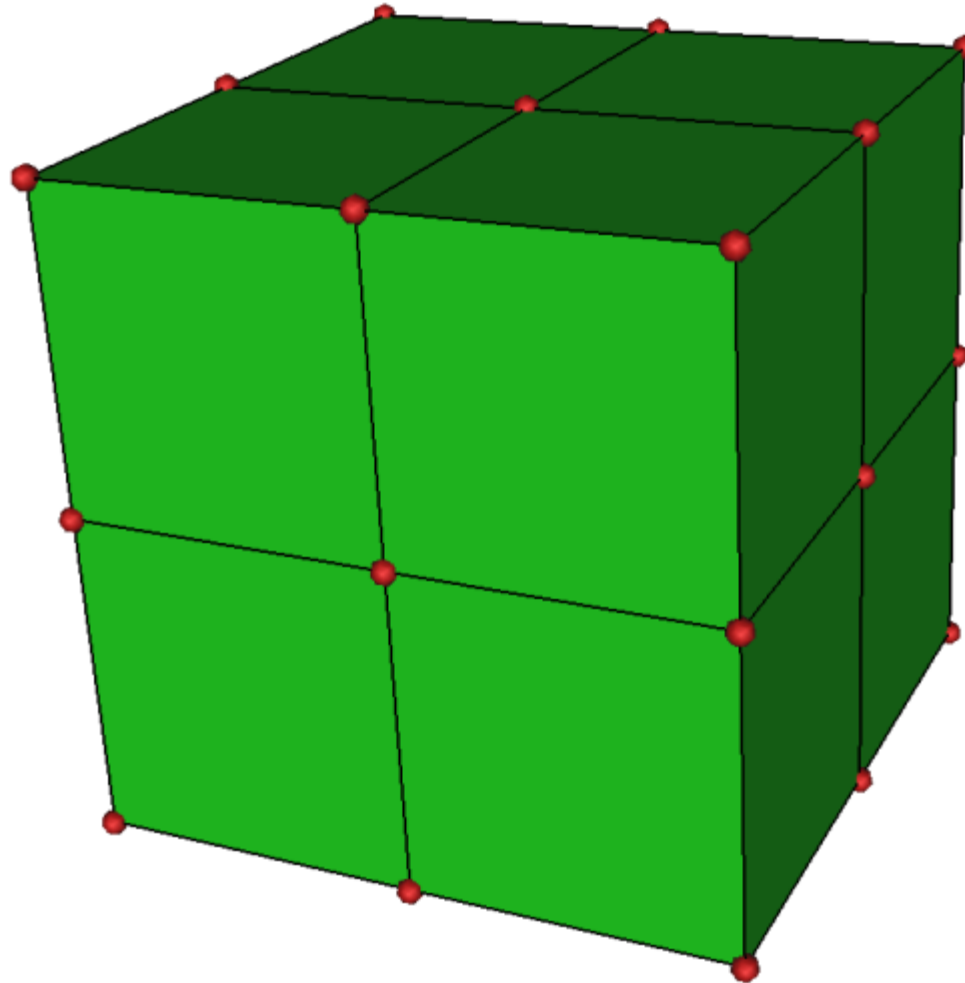


# Subdivision Surfaces – Examples

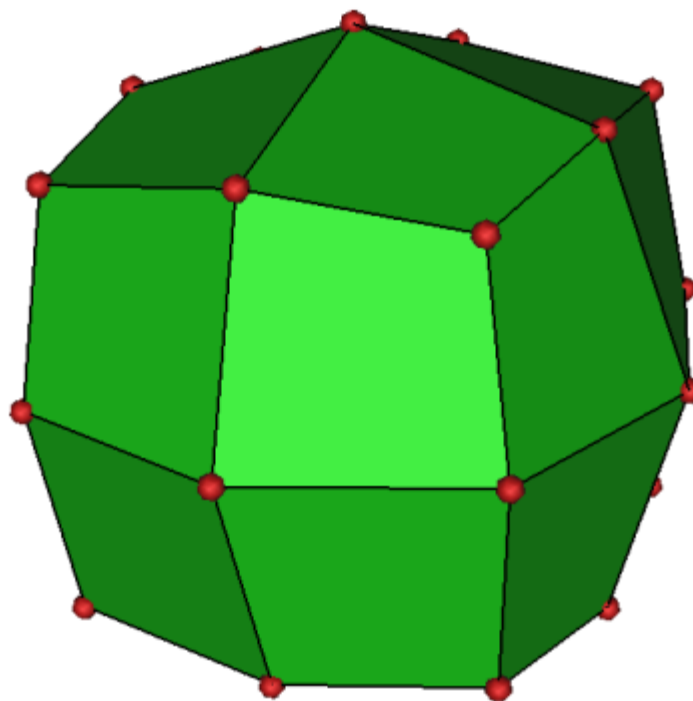




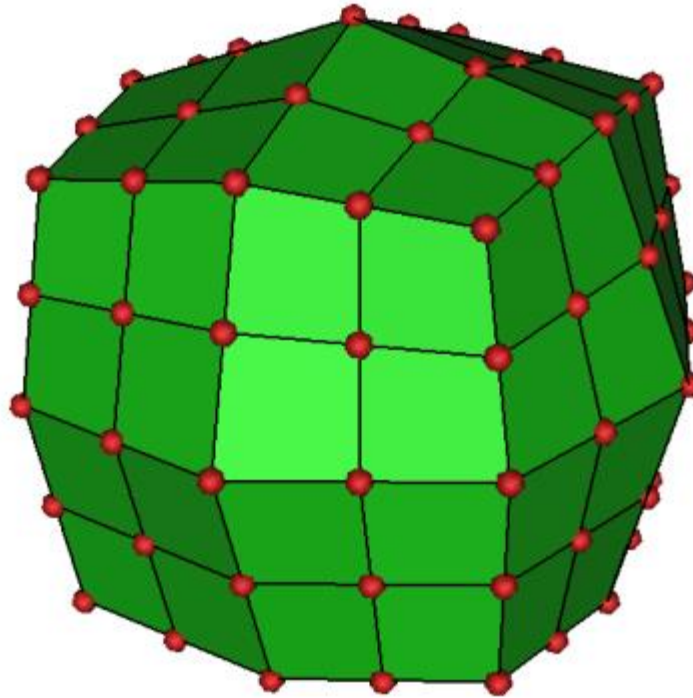
# Subdivision Surfaces – Examples



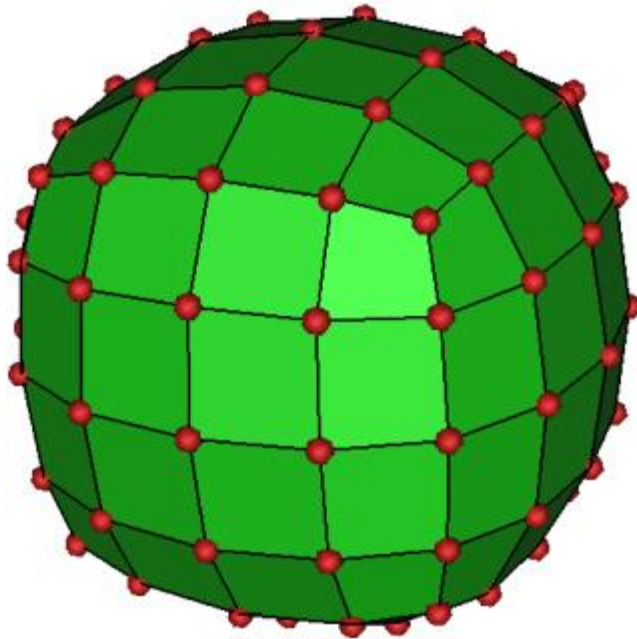
# Subdivision Surfaces – Examples



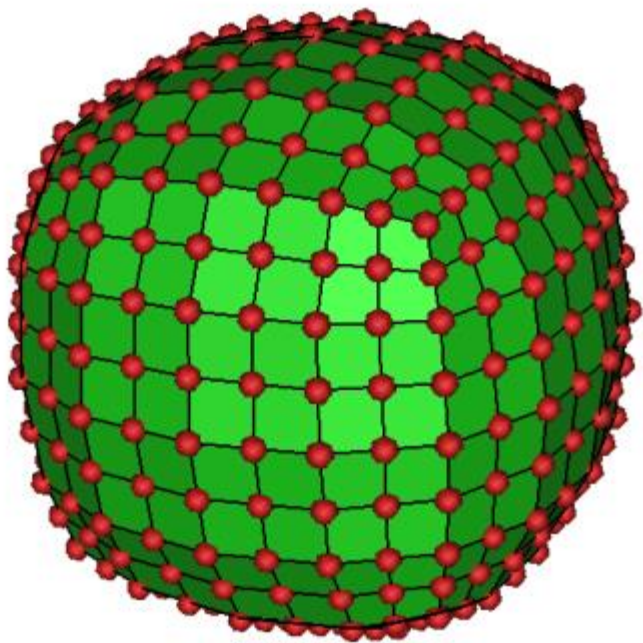
# Subdivision Surfaces – Examples



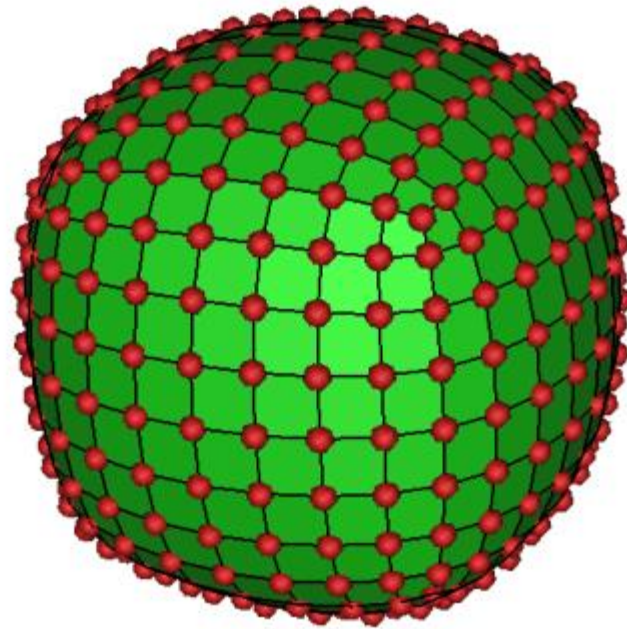
# Subdivision Surfaces – Examples



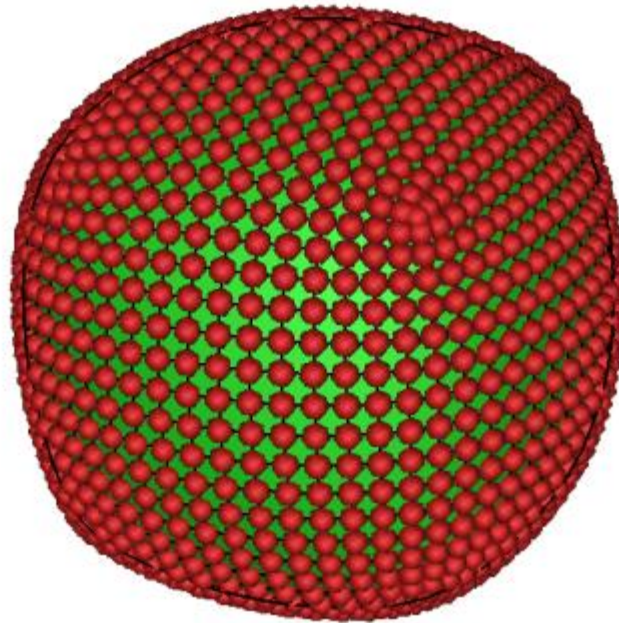
# Subdivision Surfaces – Examples



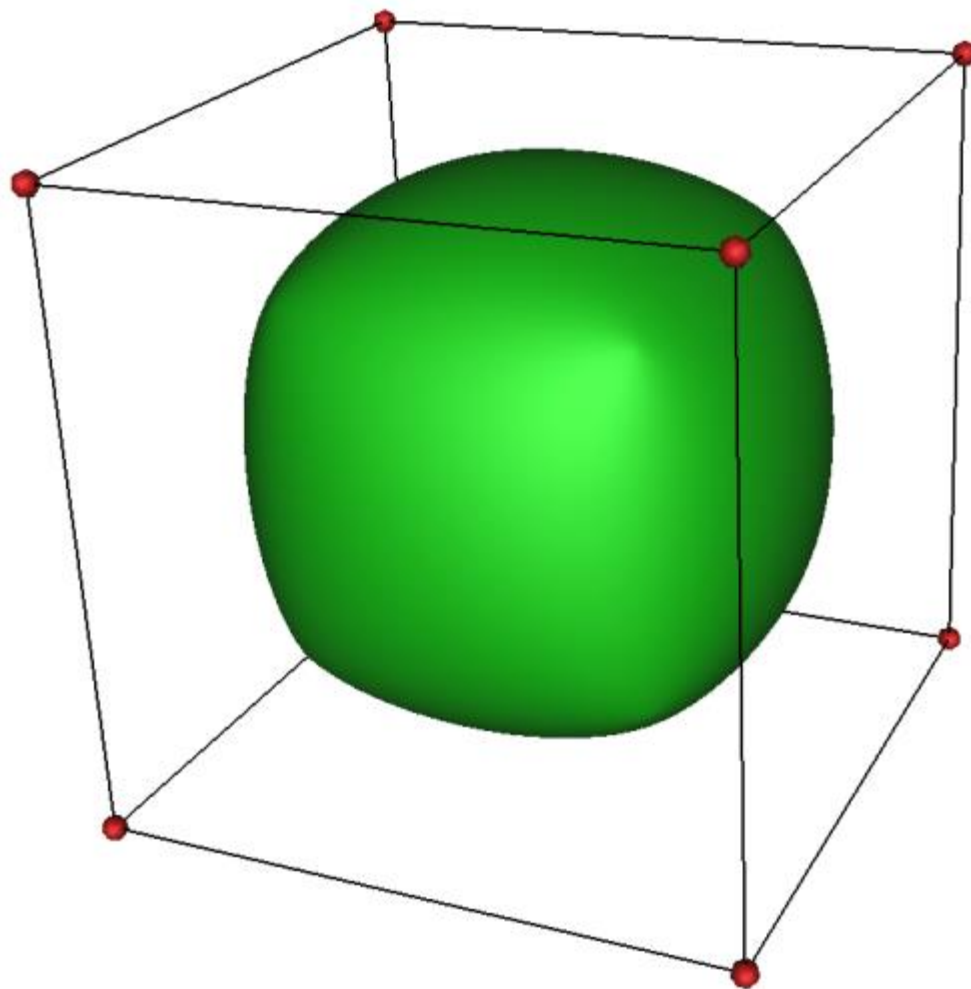
# Subdivision Surfaces – Examples



# Subdivision Surfaces – Examples



# Subdivision Surfaces – Examples





## 2.2 曲线与曲面

2.2.1 基础知识

2.2.2 B é zier曲线/曲面

2.2.3 B样条曲线/曲面

2.2.4 NURBS曲线/曲面

2.2.5 Coons曲面

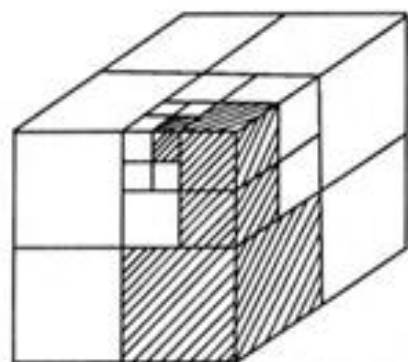
2.2.6 三维物体的描述方法

## 2.2.6 三维物体的描述方法

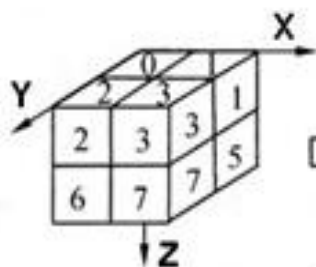
即采用什么样的方法建立三维物体的几何模型，特别是曲面物体的几何模型。

### 一、八叉树表示法 (Octree Representation)

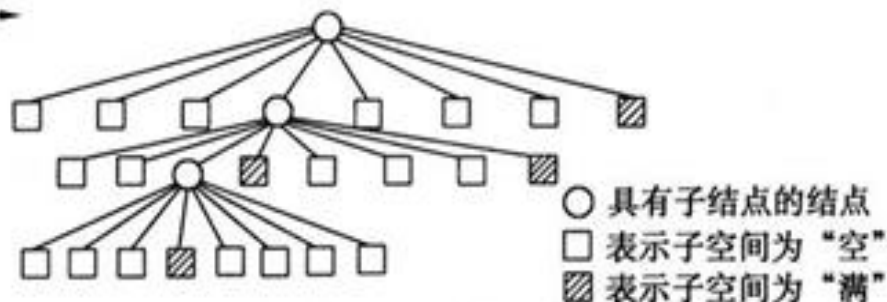
它是一种具有层次结构的空间占有计数法，见图。它在精确表示物体几何形状上不够理想，但在快速确定物体之间碰撞、相交计算、图形显示上都有重要的应用。



(a) 用八叉树法表示的空间物体



(b) 子空间的编码方法

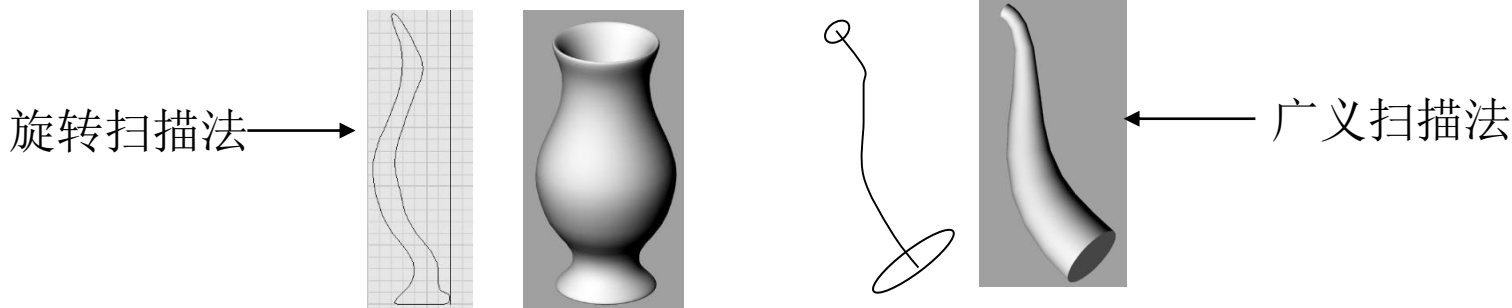
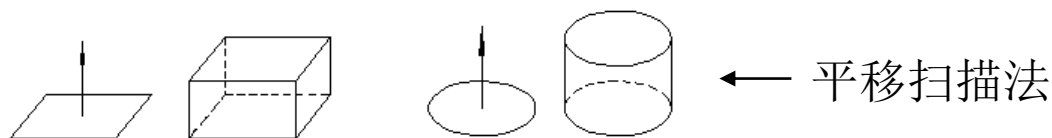


(c) 用八叉树结构记录的图 (a) 中的物体

## 2.2.6 三维物体的描述方法（续）

### 二、扫描表示法 (Sweep Representation)

**原理：** 即当空间中的一个点、一条边或一个面沿着某一路径扫描时，所形成的轨迹将定义一维的线、二维的面或三维的物体。在三维物体的表示中，应用最多的是平移扫描体和旋转扫描体。

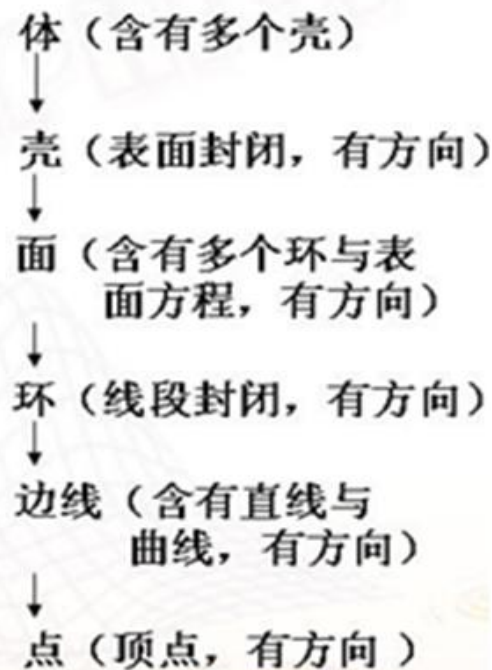


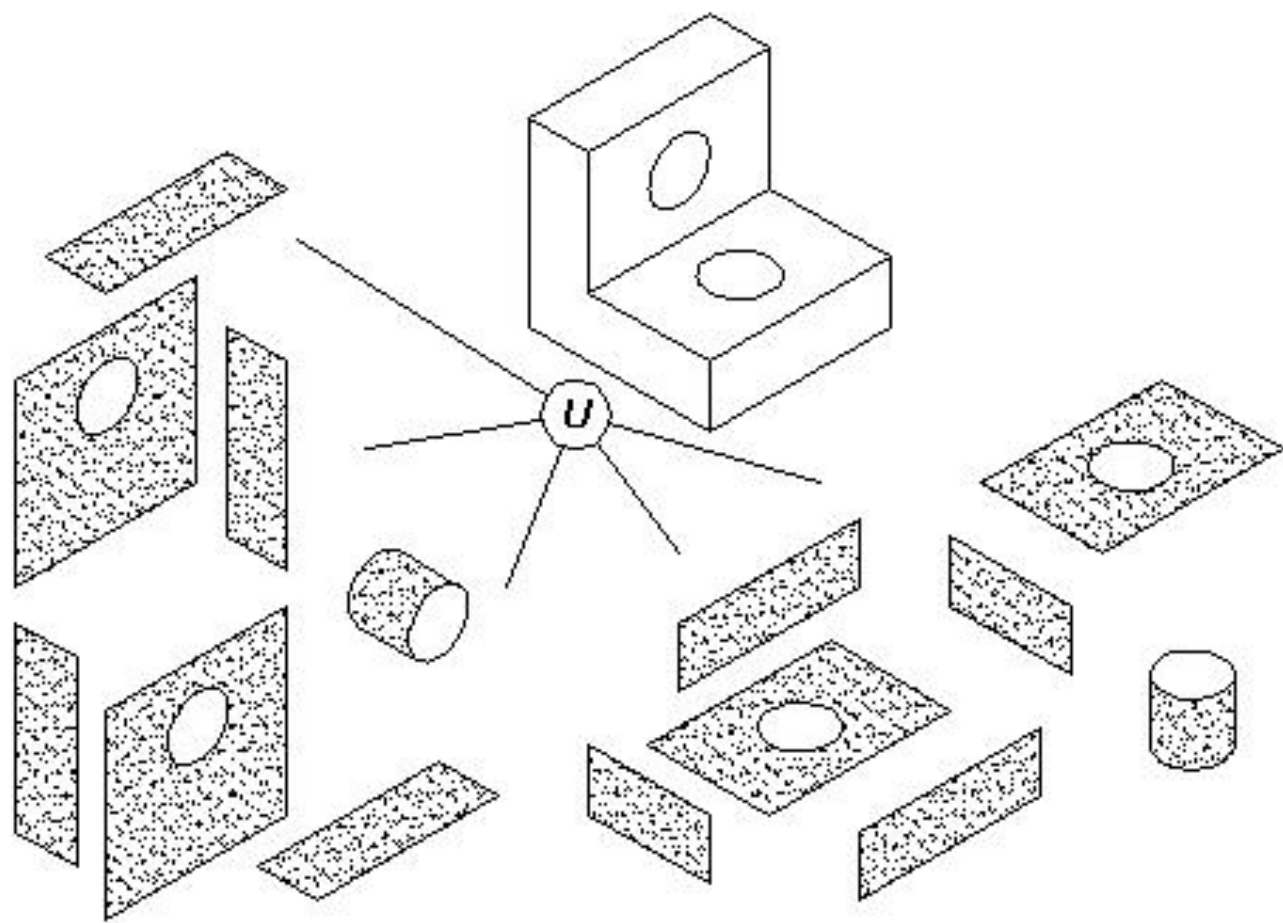
## 2.2.6 三维物体的描述方法（续）

### 三、边界表示法(Boundary Representation)

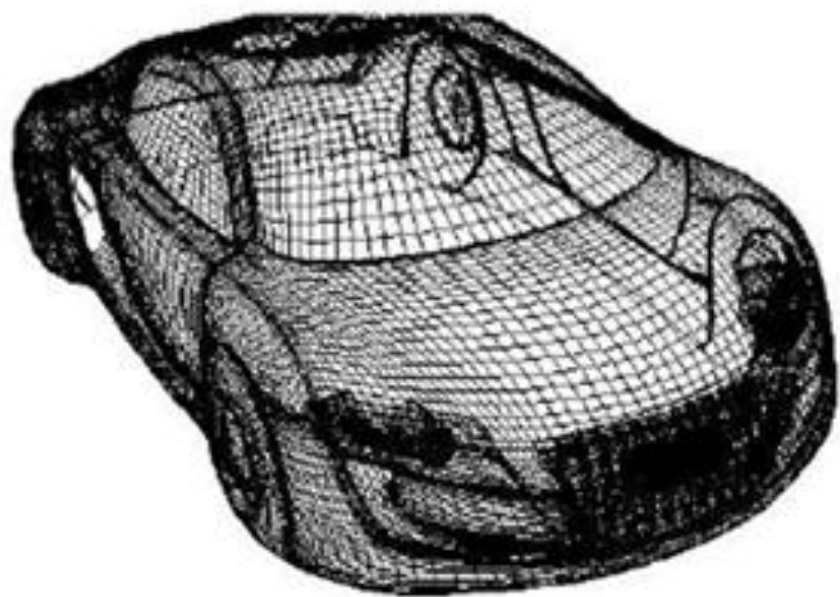
**边界表示法**以物体的表面为基础的描述物体外形的一种方法，它给出了一个物体几何形状的完整和显示的表面函数表达式。

**缺点**数据量大、运算处理复杂；  
**优点**描述物体的能力强，它与人们观察物体的习惯相一致，使用非常广泛。

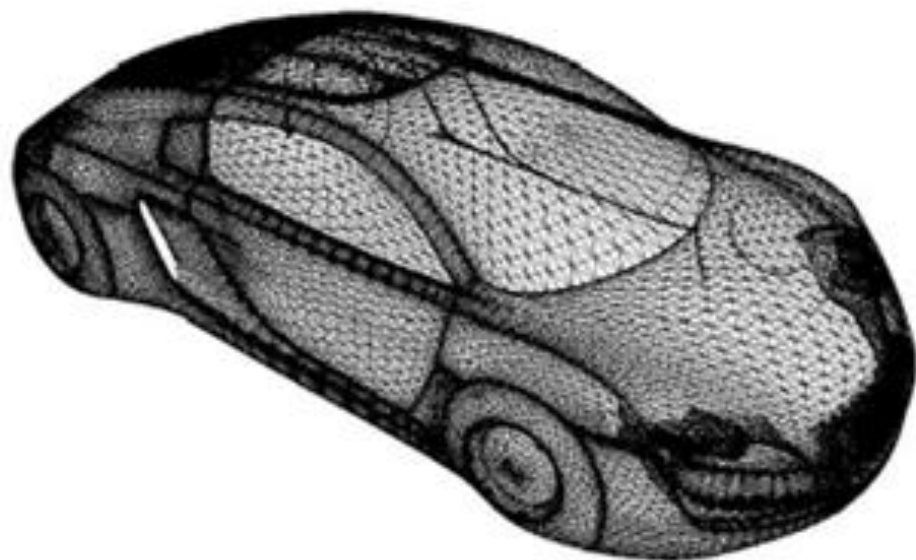




边界表示法

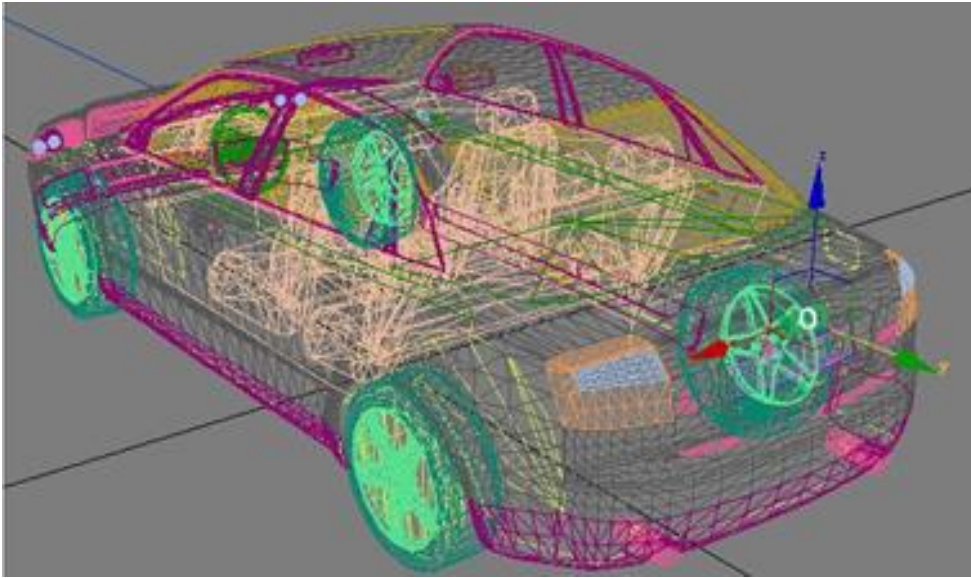


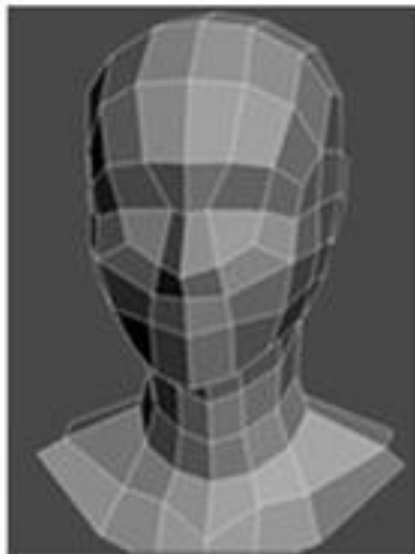
用多边形网格表示曲面



用三角面片表示曲面





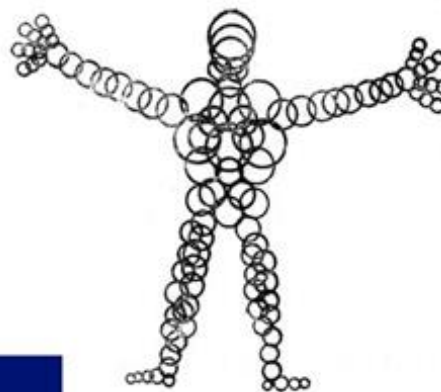




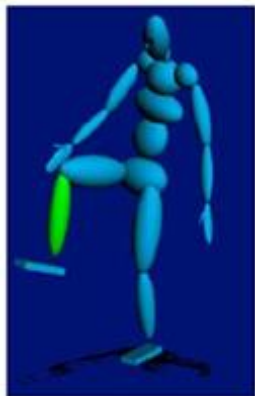
## 2.2.6 三维物体的描述方法（续）

### 四、圆球表示法

圆球具有描述方法简单的特点，用它表示人体模型见图。



用球体构成人体模型



## 思考题：

- NURBS的全称是什么？
- 请总结Bezier/B样条/Nurbs曲线与曲面的优缺点？
- 三维物体的描述方法有哪些？