第 5 章

样 条 曲 面

曲线与曲面是计算机图形学中重要的研究对象，是计算机绘图与动画技术的核心要素。

在计算机中，可以用离散的点来描述曲线曲面，也可以用直线段或者小平面片拼接在一起表示曲线曲面。平面曲线可以用一个一维数组描述，数组下标作为横坐标，数组元素的值作为纵坐标；空间曲线可以用三个一维数组描述，每个数组分别为 X、Y、Z 值，这便是参数方程表示方法；曲面可以用三维数组表示，也可以用一些三角形或者四边形的顶点数组表示。

5.1 三维空间样条曲线

在第 2 章中介绍了二维贝塞尔曲线等，现在讨论三维空间的样条曲线，主要介绍贝塞尔曲线与 B 样条曲线。

5.1.1 三维空间贝塞尔曲线

第 2 章中介绍的各种曲线都可以扩展到三维空间中来。例如，2.3 节中的式 2-8 表示的曲线是平面贝塞尔曲线，如果增加一个表达式，如式 5-1 所示，那么就表示三维空间中的贝塞尔曲线。

$$\begin{cases} x(t)=\sum_{i=0}^{n} x_i B_{i,n}(t) \\ y(t)=\sum_{i=0}^{n} y_i B_{i,n}(t) \\ z(t)=\sum_{i=0}^{n} z_i B_{i,n}(t) \end{cases} \tag{5-1}$$

式 5-1 中的 n 就是曲线的次数，当 $n=3$ 时，式 5-1 就表示空间三次贝塞尔曲线。

$B_{i,n}(t)=C_n^i t^i (1-t)^{n-i}$ 是伯恩斯坦多项式，这与二维情形相同。

式 2-12 是平面贝塞尔曲线的通用矩阵表示形式，在三维空间中，这个表示形式不变，只是 $\boldsymbol{P}(t)=\begin{bmatrix} x(t) \\ y(t) \\ z(t) \end{bmatrix}$，$\boldsymbol{P}_0=(x_0,y_0,z_0),\cdots,\boldsymbol{P}_3=(x_3,y_3,z_3)$。

例如，当 $\boldsymbol{P}[3][4]=\{\{1\quad 2\quad 5\quad 6\},\{1.5\quad 7\quad 11\quad 5\},\{2\quad 3\quad 6\quad 5.5\}\}$，即 4 个控制

点依次是(1,1.5,2),(2,7,3),(5,11,6),(6,5,5.5)时,其三维空间中的贝塞尔曲线如图 5-1 所示。

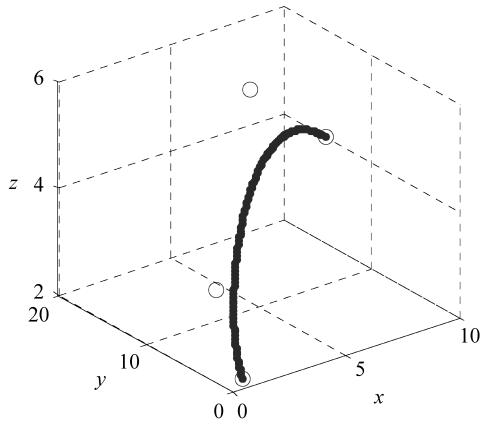


图 5-1　空间三次贝塞尔曲线

空间贝塞尔曲线的两个端点即是两端的控制点,其他控制点用来控制曲线的形状,图 5-1 中有 4 个控制点,生成的贝塞尔曲线是三次贝塞尔曲线。

与空间三次贝塞尔曲线类似,让式 2-17 中的 $P(t)=[x(t),y(t),z(t)]$,就可以表示空间三次 B 样条曲线段。

上面介绍的各种曲线本质上都是多项式曲线,只因为给定的初始条件不同,或者曲线对控制点的要求不同,所以产生了区别。一个曲线的多种参数方程是可以互相转换的。

5.1.2　曲线的拼接

两条曲线端点处的函数值决定是否能实现拼接,函数值相同才可以拼接到一起。如果端点处的切线斜率(一阶导数)也相同就可以实现平滑连接,没有折起的棱角。有时根据具体问题,要求连接时函数值相同、一阶导数相同、二阶导数也相同,如果是三维空间曲线连接,还要求导数的方向也相同,这是一种更高要求的连接。

2.4.2 节中的 Hermite 曲线给出了端点坐标值以及端点处的切向量,所以该方法绘制的曲线容易实现平滑连接。

如果两段贝塞尔曲线要实现光滑连接(一阶导数相同),那么需要满足一定的条件。

下面的例题研究的是两段三次贝塞尔曲线连接问题。

【例 5-1】 给定 4 个控制点 P_1、P_2、P_3、P_4,绘制出的三次贝塞尔曲线记为 C_1,再根据另外 4 个控制点 P_4、P_5、P_6、P_7 可以绘制出三次贝塞尔曲线 C_2。问这 7 个顶点满足什么条件时两条曲线在 P_4 点光滑连接(导数相等)?

由贝塞尔曲线的性质:贝塞尔曲线起点处切线与终点处切线分别是特征多边形的第一条边和最后一条边所在直线。所以,当 P_3、P_4、P_5 在一条直线上时,两条空间三次贝塞尔曲线在 P_4 点光滑连接。

5.1.3 三维空间B样条曲线

B样条曲线也属于一种由基函数乘以系数构造的参数曲线，其基函数与贝塞尔曲线不同，所以具有不同的性质。

【例5-2】 给定4个控制点 P_1、P_2、P_3、P_4 绘制出的三次 B 样条曲线段记为 C_1；再给定一点 P_5，绘制出由 P_2、P_3、P_4、P_5 控制的三次B样条曲线段记为 C_2，观察曲线段 C_1 与 C_2 的关系。

三次B样条曲线的基函数如下。

$$F_{0,3}(t)=\frac{1}{6}(-t^3+3t^2-3t+1)$$

$$F_{1,3}(t)=\frac{1}{6}(3t^3-6t^2+4)$$

$$F_{2,3}(t)=\frac{1}{6}(-3t^3+3t^2+3t+1)$$

$$F_{3,3}(t)=\frac{1}{6}t^3$$

三次B样条曲线段的参数方程如下。

$$\begin{cases}x(t)=F_{0,3}(t)x_0+F_{1,3}(t)x_1+F_{2,3}(t)x_2+F_{3,3}(t)x_3\\y(t)=F_{0,3}(t)y_0+F_{1,3}(t)y_1+F_{2,3}(t)y_2+F_{3,3}(t)y_3\\z(t)=F_{0,3}(t)z_0+F_{1,3}(t)z_1+F_{2,3}(t)z_2+F_{3,3}(t)z_3\end{cases}$$

建立单文档项目，在视图文件中OnDraw()函数的前面写入函数Draw3()如下。

```
void Draw3(float k[4][3],int color,CDC * p)
{
    float x,y,z;
    float f3t[4],kk[4][3]={0};
    for(int i=0;i<4;i++)
        for(int j=0;j<3;j++)
            kk[i][j]=k[i][j];
    for(float t=0;t<1;t=t+0.001)
    {
        f3t[0]=1/6.0 * (-pow(t,3)+3 * pow(t,2)-3 * t+1);
        f3t[1]=1/6.0 * (3 * pow(t,3)-6 * pow(t,2)+4);
        f3t[2]=1/6.0 * (-3 * pow(t,3)+3 * pow(t,2)+3 * t+1);
        f3t[3]=1/6.0 * pow(t,3);
        x=f3t[0] * kk[0][0]+f3t[1] * kk[1][0]+f3t[2] * kk[2][0]+f3t[3] * kk[3][0];
        y=f3t[0] * kk[0][1]+f3t[1] * kk[1][1]+f3t[2] * kk[2][1]+f3t[3] * kk[3][1];
        z=f3t[0] * kk[0][2]+f3t[1] * kk[1][2]+f3t[2] * kk[2][2]+f3t[3] * kk[3][2];
        p->SetPixel((int)x+100,(int)y+100,color);
    }
}
```

在 OnDraw()函数中写入调用函数 Draw3()的代码，写入后 OnDraw()函数的代码如下(其中粗体是加入的语句)。

```
void CBbbView::OnDraw(CDC * pDC)
{
    CBbbDoc * pDoc = GetDocument();
    ASSERT_VALID(pDoc);
    //TODO: add draw code for native data here
    Float xyz1[4][3]={{20,100,110},{70,120,10},{200,60,20},{250,20,160}};
    Draw3(xyz1,0,pDC);
    Float xyz2[4][3]={{70,120,10},{200,60,20},{250,20,160},{150,16,178}};
    Draw3(xyz2,0,pDC);
}
```

编译运行，绘制出如图 5-2(a)所示的投影图形。

把语句

```
p->SetPixel((int)x+100,(int)y+100,color)
```

修改为：

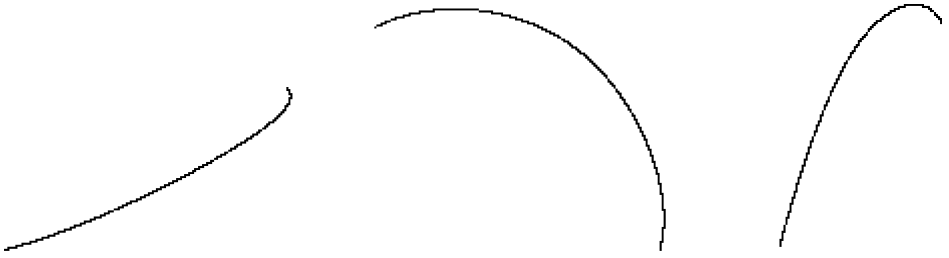
```
p->SetPixel((int)x+100,(int)z+100,color)
```

绘制出如图 5-2(b)所示的投影图形。

把这个语句再修改为：

```
p->SetPixel((int)y+100,(int)z+100,color)
```

绘制出如图 5-2(c)所示的投影图形。



(a) *XY*面投影　　(b) *XZ*面投影　　(c) *YZ*面投影

图 5-2　三次 B 样条曲线在三个坐标轴上的投影(0＜*t*＜1)

该程序绘制的曲线是三维空间中的曲线，在三个坐标平面上的投影都是样条曲线。本来是绘制两段曲线，但是这两段曲线(“无缝地”)连接在一起了。

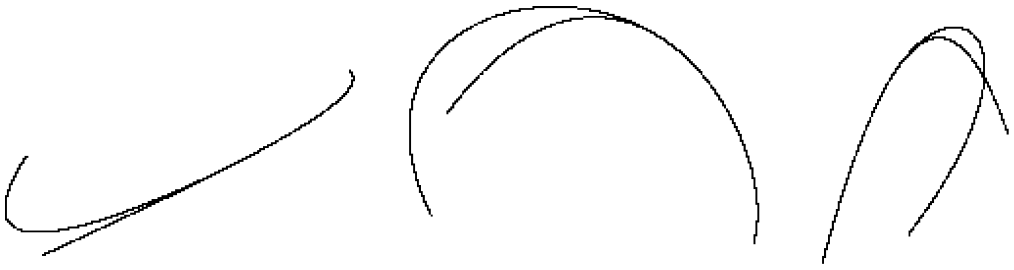
为了更好地观察连接情况，把语句

```
for(float t=0;t<1;t=t+0.001)
```

修改为

```
for(float t=-1;t<1;t=t+0.001)
```

结果绘制出的对应各个坐标平面的投影图形如图 5-3 所示。从图上可以看到两段不同的曲线在某个点处光滑的连接过渡情形。



(a) *XY*面投影　　(b) *XZ*面投影　　(c) *YZ*面投影

图 5-3　三次 B 样条曲线在三个坐标轴上的投影(−1＜*t*＜1)

【注】 该例题的坐标系是 VC++ 默认的坐标系,左上角为(0,0)点,横轴正方向向右,纵轴正方向向下。

【思考题】 在例 5-2 给出的条件下,为什么两个 B 样条曲线段能够实现光滑连接?参数 t 的范围为什么限制在 0～1 就可以无缝光滑连接?

与贝塞尔曲面类似,绘制 B 样条曲线也有快速算法。正是因为有快速绘制方法,所以这类曲线曲面才在图形绘制中被广泛使用。

5.1.4　三维空间分段插值曲线

给定三维空间中的一些离散点,然后求(或者绘制)满足一定条件的一条曲线,使得该曲线过这些离散点,这就是三维空间曲线插值问题。

与贝塞尔曲线以及 B 样条曲线不同的是,插值曲线必须过插值点。

给定三维插值点后,实现三维空间曲线插值有很多方法,其中,三次样条插值是比较常用的一种方法。

下面以一个具体例子说明三维空间中的三次样条插值问题。

给定 5 个点(1,1,1),(2,2,3),(3,6,5),(6,3,2),(9,6,1),求三维空间内顺次经过这 5 个点的两段三次样条插值曲线,使得在(3,6,5)点连续并且在该点的一阶导数与二阶导数相等,在点(1,1,1)处斜率向量为(−1,−1,−1),在点(9,6,1)处的斜率向量为(1,1,1)。

空间曲线可以视为两个曲面的交线,由式 5-2 确定,也可以由式 5-3 参数方程确定。

$$\begin{cases} F_1(x,y,z)=0 \\ F_2(x,y,z)=0 \end{cases} \tag{5-2}$$

$$\begin{cases} x=x(t) \\ y=y(t) \\ z=z(t) \end{cases} \tag{5-3}$$

所谓切向量，如果使用式5-3，就是指$\left(\frac{\mathrm{d}x}{\mathrm{d}t},\frac{\mathrm{d}y}{\mathrm{d}t},\frac{\mathrm{d}z}{\mathrm{d}t}\right)$。

对于上面的三维空间三次样条插值问题，使用待定系数法可以计算出三次样条插值曲线（段）。

5.2 贝塞尔曲面

在图形学中，贝塞尔曲面是常用的曲面之一。在前面绘制曲面的过程中，使用了方程进行绘制。对于计算机图形学中的一些曲面，例如贝塞尔曲面也有参数方程，可以使用方程绘制。不过，贝塞尔曲面以及B样条曲面除了可以使用方程绘制外，还可以使用递归快速算法进行绘制。使用贝塞尔曲面可以轻松绘制出高次多项式曲面，同时可以控制该曲面的大致形状。诸多的优点使得计算机图形学教材或者著作中都介绍这类曲面。

5.2.1 贝塞尔曲面的定义

下面给出贝塞尔曲面的参数表达形式：

在三维空间中，给定$(n+1)\times(m+1)$个点$P_{ij}=(x_{ij},y_{ij},z_{ij})$，$i=0,1,2,\cdots,n$；$j=0,1,2,\cdots,m$。

$$\begin{cases}X=f(u,v)=\sum\limits_{i=0}^{n}\sum\limits_{j=0}^{m}x_{ij}B_{i,n}(u)B_{j,m}(v)\\Y=g(u,v)=\sum\limits_{i=0}^{n}\sum\limits_{j=0}^{m}y_{ij}B_{i,n}(u)B_{j,m}(v)\\Z=h(u,v)=\sum\limits_{i=0}^{n}\sum\limits_{j=0}^{m}z_{ij}B_{i,n}(u)B_{j,m}(v)\end{cases}\quad 0\leqslant u\leqslant 1,0\leqslant v\leqslant 1 \tag{5-4}$$

其中，$B_{i,n}(u)=C_n^i u^i(1-u)^{n-i}$，$B_{j,m}(v)=C_m^j v^j(1-v)^{m-j}$是伯恩斯坦基函数，组合数$C_n^i=\frac{n!}{i!(n-i)!}$，$C_m^j=\frac{m!}{j!(m-j)!}$。

从式5-4看出，贝塞尔曲面的构造方式与贝塞尔曲线类似，基函数是相同的；只是贝塞尔曲面有三个变量，也就有三个参数方程构成方程组。当u,v在一个平面区域变化时，三维点(X,Y,Z)便形成了曲面。

5.2.2 双一次贝塞尔曲面

在式5-4中，如果$n=m=1$，那么得到的曲面称为双一次贝塞尔曲面。

【例5-3】 把$n=m=1$代入式5-4，求出双一次贝塞尔曲面的参数方程，分析双一次贝塞尔曲面片的特征。

把$n=m=1$代入后得：

$$\begin{cases} X=\sum_{i=0}^{1}\sum_{j=0}^{1}x_{ij}B_{i,1}(u)B_{j,1}(v) \\ Y=\sum_{i=0}^{1}\sum_{j=0}^{1}y_{ij}B_{i,1}(u)B_{j,1}(v) \\ Z=\sum_{i=0}^{1}\sum_{j=0}^{1}z_{ij}B_{i,1}(u)B_{j,1}(v) \end{cases} \Rightarrow \begin{cases} \boldsymbol{X}=[B_{0,1}(u)B_{1,1}(u)]\begin{bmatrix}x_{00} & x_{01}\\ x_{10} & x_{11}\end{bmatrix}\begin{bmatrix}B_{0,1}(v)\\ B_{1,1}(v)\end{bmatrix} \\ \boldsymbol{Y}=[B_{0,1}(u)B_{1,1}(u)]\begin{bmatrix}y_{00} & y_{01}\\ y_{10} & y_{11}\end{bmatrix}\begin{bmatrix}B_{0,1}(v)\\ B_{1,1}(v)\end{bmatrix} \\ \boldsymbol{Z}=[B_{0,1}(u)B_{1,1}(u)]\begin{bmatrix}z_{00} & z_{01}\\ z_{10} & z_{11}\end{bmatrix}\begin{bmatrix}B_{0,1}(v)\\ B_{1,1}(v)\end{bmatrix} \end{cases}$$

其中，基函数：

$$B_{0,1}(u)=1-u,\quad B_{1,1}(u)=u$$
$$B_{0,1}(v)=1-v,\quad B_{1,1}(v)=v$$

所以有

$$\begin{cases} X=(1-u)(1-v)x_{00}+u(1-v)x_{10}+(1-u)vx_{01}+uvx_{11} \\ Y=(1-u)(1-v)y_{00}+u(1-v)y_{10}+(1-u)vy_{01}+uvy_{11} \\ Z=(1-u)(1-v)z_{00}+u(1-v)z_{10}+(1-u)vz_{01}+uvz_{11} \end{cases} \tag{5-5}$$

写成矩阵表示形式为

$$\boldsymbol{P}(u,v)=[u\quad 1]\begin{bmatrix}-1 & 1\\ 1 & 0\end{bmatrix}\begin{bmatrix}\boldsymbol{P}_{00} & \boldsymbol{P}_{01}\\ \boldsymbol{P}_{10} & \boldsymbol{P}_{11}\end{bmatrix}\begin{bmatrix}-1 & 1\\ 1 & 0\end{bmatrix}\begin{bmatrix}v\\ 1\end{bmatrix} \tag{5-6}$$

分析式 5-5 参数方程(组)，该方程组有两个参数，各项的最高次数为 2；方程中一共有 4 个控制点的 12 个坐标值；3 个方程的形式是相同的，只是分别使用了控制点的 x、y 与 z 坐标值。使 u 与 v 在 0～1 变化，对应每一个(u，v)都能得到一个空间点(X，Y，Z)。

当 $u=0$、$v=0$、$u=1$、$v=1$ 其中一个成立时，便得到一条边界直线段；所以一共有 4 条边界直线段。当 u 或者 v 中一个不变，改变另外一个，得到的是一条直线段。

可以证明，双一次贝塞尔曲面是双曲抛物面。

下面编写程序绘制双一次贝塞尔曲面。

【例 5-4】 绘制双一次贝塞尔曲面。

修改例 5-2 中相关语句如下。

```
double cv[4][3]={{20,10,110},{100,60,20},{90,20,10},{5,30,80}};
for(double u=0;u<1;u=u+0.05)
{   n=0;
    for(double v=0;v<1;v=v+0.05)
    {   x[m][n][0]=(1-u) * (1-v) * cv[0][0]+u * (1-v) * cv[1][0]+(1-u) * v * cv[2]
[0]+u * v * cv[3][0];
        x[m][n][1]=(1-u) * (1-v) * cv[0][1]+u * (1-v) * cv[1][1]+(1-u) * v * cv[2]
[1]+u * v * cv[3][1];
        x[m][n][2]=(1-u) * (1-v) * cv[0][2]+u * (1-v) * cv[1][2]+(1-u) * v * cv[2]
[2]+u * v * cv[3][2];
    n++;
}
```

绘制出的图形如图 5-4(a)所示。

如果把控制点赋值语句修改为如下所示，每次运行，或者调整运行后文档窗口，都可以重新绘制出新的曲面，图 5-4(b)和图 5-4(c)就是随机生成的两个曲面。

```
double cv[4][3];
for(int i=0;i<4;i++)
    for(int j=0;j<3;j++)
        cv[i][j]=rand()%300;        //控制点坐标介于 0~300
```

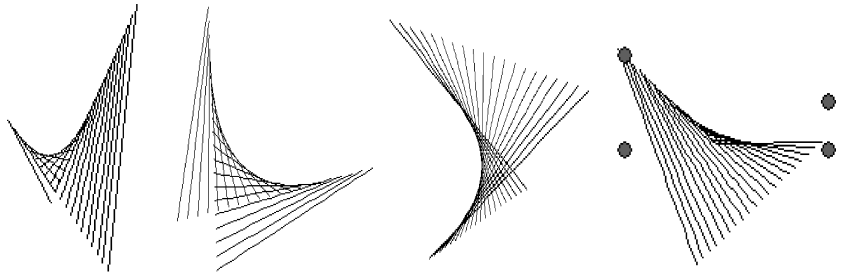


图 5-4　双一次贝塞尔曲面

【注】 空间中的一些直线也可以构成曲面。

【例 5-5】 绘制双一次贝塞尔曲面，并绘制出其 4 个控制点。

在例 5-4(从例 5-2 修改而来)中的函数 DrawCurve()的最后加入下面的语句。

```
CBrush b(RGB(255,0,0));
pDC->SelectObject(&b);
for(int r=0;r<4;r++)
{
    xx[r]=(cv[r][0]-p[0]/p[2]*cv[r][2])+100;
    yy[r]=(cv[r][1]-p[1]/p[2]*cv[r][2])+100;
    pDC->Ellipse(xx[r],yy[r],xx[r]+10,yy[r]+10);
}
```

运行程序，除了绘制出双一次贝塞尔曲面外，还绘制出 4 个控制点。为了清楚，用小椭圆代替了点。

5.2.3　双二次贝塞尔曲面

在式 5-4 中，令 $n=m=2$，得到的曲面称为双二次贝塞尔曲面。

当 $n=m=2$ 时，给定 $(n+1)\times(m+1)=9$ 个点 $P_{ij}=(x_{ij},y_{ij},z_{ij})$，其中，$i=0,1,2$；$j=0,1,2$。代入式 5-7：

$$P(u,v)=\sum_{i=0}^{2}\sum_{j=0}^{2}P_{ij}B_{i,2}(u)B_{j,2}(v)\quad 0\leqslant u\leqslant 1,0\leqslant v\leqslant 1 \tag{5-7}$$

其中，基函数：

$$B_{0,2}(u)=u^2-2u+1,\quad B_{1,2}(u)=-2u^2+2u,\quad B_{2,2}(u)=u^2$$

$$B_{0,2}(v)=v^2-2v+1,\quad B_{1,2}(v)=-2v^2+2v,\quad B_{2,2}(v)=v^2$$

把如式 5-7 所示双二次贝塞尔曲面的参数方程整理成矩阵表示形式为

$$\boldsymbol{P}(u,v)=\begin{bmatrix}u^2 & u & 1\end{bmatrix}\begin{bmatrix}1 & -2 & 1\\ -2 & 2 & 0\\ 1 & 0 & 0\end{bmatrix}\begin{bmatrix}\boldsymbol{P}_{00} & \boldsymbol{P}_{01} & \boldsymbol{P}_{02}\\ \boldsymbol{P}_{10} & \boldsymbol{P}_{11} & \boldsymbol{P}_{12}\\ \boldsymbol{P}_{20} & \boldsymbol{P}_{21} & \boldsymbol{P}_{22}\end{bmatrix}\begin{bmatrix}1 & -2 & 1\\ -2 & 2 & 0\\ 1 & 0 & 0\end{bmatrix}\begin{bmatrix}v^2\\ v\\ 1\end{bmatrix} \tag{5-8}$$

其中，顶点矩阵实际上是 3 个矩阵，即 9 个控制点的 $\boldsymbol{X}$，$\boldsymbol{Y}$，$\boldsymbol{Z}$ 矩阵。即矩阵

$$\begin{bmatrix}\boldsymbol{P}_{00} & \boldsymbol{P}_{01} & \boldsymbol{P}_{02}\\ \boldsymbol{P}_{10} & \boldsymbol{P}_{11} & \boldsymbol{P}_{12}\\ \boldsymbol{P}_{20} & \boldsymbol{P}_{21} & \boldsymbol{P}_{22}\end{bmatrix}$$

相当于下面三个矩阵：

$$\begin{bmatrix}x_{00} & x_{01} & x_{02}\\ x_{10} & x_{11} & x_{12}\\ x_{20} & x_{21} & x_{22}\end{bmatrix},\begin{bmatrix}y_{00} & y_{01} & y_{02}\\ y_{10} & y_{11} & y_{12}\\ y_{20} & y_{21} & y_{22}\end{bmatrix},\begin{bmatrix}z_{00} & z_{01} & z_{02}\\ z_{10} & z_{11} & z_{12}\\ z_{20} & z_{21} & z_{22}\end{bmatrix} \tag{5-9}$$

对于每一个固定的(u,v)，对于式 5-9 中的每一个矩阵(给定数值后)，代入式 5-8，都可以得到一个数值；三个矩阵得到三个数值，便是一个空间点的坐标(x,y,z)。

式 5-9 中的数值就是贝塞尔曲面的控制点的坐标。

图 5-5 是借助于 MATLAB 软件绘制的双二次贝塞尔曲面。

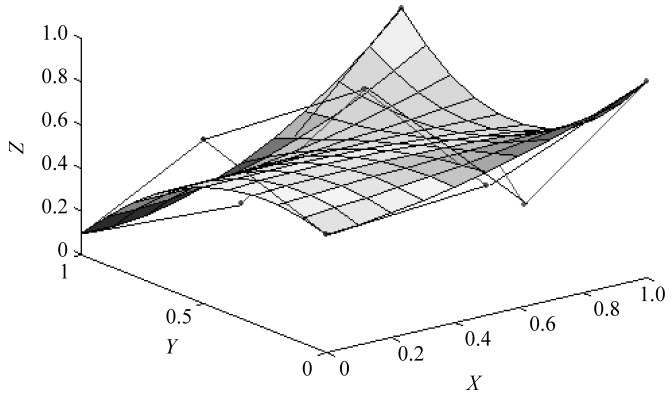


图 5-5　9 个控制点的双二次贝塞尔曲面及其控制网格

其 9 个控制点的坐标的 X、Y、Z 值如下。

$X=0$	0.5000	1.0000	0	0.5000	1.0000	0	0.5000	1.0000
$Y=0$	0	0	0.5000	0.5000	0.5000	1.0000	1.0000	1.0000
$Z=0.5468$	0.5978	0.9053	0.7596	0.8144	0.1088	0.0963	0.0580	0.7884

其中有 4 个控制点在曲面(的 4 个角)上。

5.2.4 双三次贝塞尔曲面的16个控制点

当 $n=m=3$ 时，式5-4表示的曲面称为双三次贝塞尔曲面。

当 $n=m=3$ 时，给定 $(n+1)\times(m+1)=16$ 个点 $P_{ij}=(x_{ij},y_{ij},z_{ij})$，其中，$i=0,1,2,3$；$j=0,1,2,3$。代入式5-10：

$$\boldsymbol{P}(u,v)=\sum_{i=0}^{3}\sum_{j=0}^{3}P_{ij}B_{i,3}(u)B_{j,3}(v)\quad 0\leqslant u\leqslant 1,\ 0\leqslant v\leqslant 1 \tag{5-10}$$

其中，基函数：

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

把式5-10整理成矩阵形式为

$$\boldsymbol{P}(u,v)=\begin{bmatrix}u^3 & u^2 & u & 1\end{bmatrix}\begin{bmatrix}-1 & 3 & -3 & 1\\ 3 & -6 & 3 & 0\\ -3 & 1 & 0 & 0\\ 0 & 0 & 0 & 0\end{bmatrix}\begin{bmatrix}\boldsymbol{P}_{00} & \boldsymbol{P}_{01} & \boldsymbol{P}_{02} & \boldsymbol{P}_{03}\\ \boldsymbol{P}_{10} & \boldsymbol{P}_{11} & \boldsymbol{P}_{12} & \boldsymbol{P}_{13}\\ \boldsymbol{P}_{20} & \boldsymbol{P}_{21} & \boldsymbol{P}_{22} & \boldsymbol{P}_{23}\\ \boldsymbol{P}_{30} & \boldsymbol{P}_{31} & \boldsymbol{P}_{32} & \boldsymbol{P}_{33}\end{bmatrix}\begin{bmatrix}-1 & 3 & -3 & 1\\ 3 & -6 & 3 & 0\\ -3 & 1 & 0 & 0\\ 0 & 0 & 0 & 0\end{bmatrix}\begin{bmatrix}v^3\\ v^2\\ v\\ 1\end{bmatrix}$$

通常在实际工作中，一般使用三次以下的贝塞尔曲面进行物体建模与图形设计。三次贝塞尔曲面已经具有足够的形状变化。

图5-6就是一个16个控制点控制的三次贝塞尔曲面，同时（为了清晰）只绘制出了 Y 轴方向的网格线。

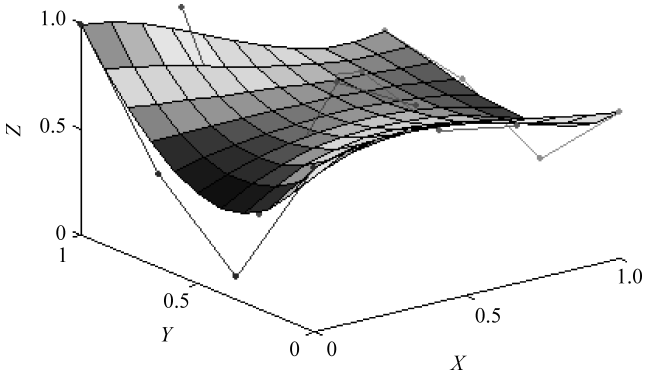


图5-6 16个控制点的双三次贝塞尔曲面及其Y轴方向上的控制网格

网格（控制）点的坐标点 X、Y、Z 分量如下。

$X=$

0	0.3333	0.6667	1.0000
0	0.3333	0.6667	1.0000
0	0.3333	0.6667	1.0000
0	0.3333	0.6667	1.0000