# 基于MATLAB语言的图形学实验教程

钟宝江　编著

# 前 言

图形学是计算机图形学是利用计算机研究图形的表示、生成、处理和显示的一门重要的计算机学科分支，它是计算机科学中最活跃的分支之一。近年来，随着计算机及互联网技术的迅速发展，计算机图形学正越来越深入我们的生活，它在图形视频处理，工业建模，游戏制作，生物信息，医药医疗等各行各业都有着及其重要的作用。

图形学具有完备和系统的数学理论基础，但毫无疑问，这也是一门需要重视实践的学科。为此，我们编写了这本实验教程。本教程最明显的特色是基于 MATLAB 语言平台来完成系列实验。在作者从事教学和科研中，可以说不断感受到 MATLAB 语言的发展和强大。以图形学中的曲线曲面构造为例，MATLAB 能够提供数十种相关的函数命令用于解决不同性质的问题。另一方面，MATLAB 虽然本身并不是直接面向工程开发的语言，然而即使是工程技术人员，也会经常使用 MATLAB 对某个思路方案进行实验性调研，取得具体结果后再进行实际编程开发。究其原因是因为 MATLAB 能够让使用者跳过比较繁琐的底层开发任务，直接面对问题来考虑解决方案。相比之下，作者以为在图形学教授和学习过程中，底层开发技术更适合让学生自学，而深层次的理论和方法需要在课堂上做更好的引导。从这个意义上来说，《基于 MATLAB 的计算机图形学实验教程》能够帮助学习者迅速提高图形学的实际应用能力。

本书作为《计算机图形学》课程的实验教材，适用于讲授 16-20 学时。具体内容分为 9 个章节，安排了 9 个具体的实验项目。每个实验项目又分为四个部分：实验目的；程序思想；算法代码样例；作业题。四部分的内容和功能有机搭配，比如算法代码样例中会给出完成作业可以借鉴的基础算法代码，而作业题则侧重于考察学生对多个知识点的综合和灵活运用。

本书同时附带了两个附录。一是实验报告的标准格式。在科学研究领域，有“三分实验、七分写作”的说法。而写作的第一原则是遵照标准格式来撰写报告内容。附录二中是作者实际讲授本实验课程时学生提交的报告，可以作为教程使用者的参考。

# 目 录

# 实验一 曲线多项式插值与龙格现象

## 一） 实验目的

代数多项式因为便于求导和求积分等运算，是用于曲线插值和拟合的首选。最初人们尝试直接使用一个多项式来近似曲线，但低次多项式一般无法取得良好的逼近效果。如果增加多项式的次数，则容易出现著名的龙格（Runge）现象，即在插值区间的某些区域(特别是两端)因多项式曲线剧烈波动产生较大的误差。

作为本课程第一次实验，本实验的目的包括两个方面。一是以前没有MATLAB 语言基础的同学通过本次实验进一步熟悉 MATLAB 语言的使用；二是让同学们理解我们在科研研究中会常常遇到的困境：一个很好的想法往往在开始时无法取得理想的效果；同时，大家也可以借此理解图形处理领域得到“第一桶金”（即随后的样条插值）的艰辛。

## 二） 程序思想

考虑如下一个看似简单的函数

$$f(x)=\frac{1}{1+25x^2},$$

我们首先利用 MATLAB 语言的绘图功能绘制出其在区间$[-1,1]$上的函数曲线，并在上面等距截取 11 个点作为数据采样点，如图 1.1 所示。

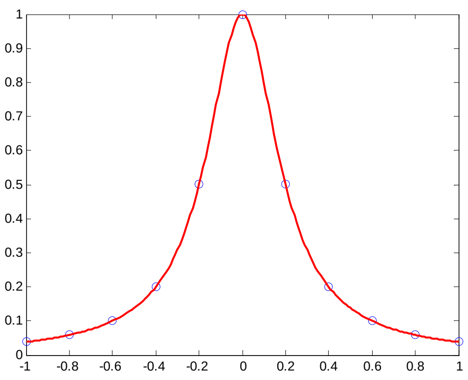



图 1.1 待插值的函数与插值节点

要强调的是，在实际的工程问题中待插值的函数及其曲线往往是未知的，数据采样点则可以通过测量等工程手段获取。

首先应用低次多项式进行插值，以下以二次多项式为例。二次插值需要 3 个插值节点，以等距对称方法从以上的数据采样点中选取。MATLAB 语言有多种实现插值的途径，这里我们选取 polyfit 命令来实现。该命令可以实现对曲线

的多项式拟合，而当拟合多项式的次数 $m$ 与数据采样点个数 $n$ 满足关系式 $m = n - 1$ 时，拟合与插值是一致的。对上述函数曲线的二次插值结果如图 1.2 所示，可见插值效果并不理想。

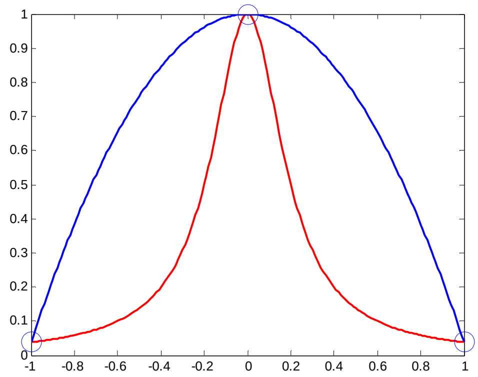



图 1.2 二次多项式插值效果

图 1.3 给出了十次多项式插值的效果。可以看出，虽然插值多项式在更多点上与原函数有相等的函数值，但总体插值效果变得更加不理想：多项式曲线在区间两端出现了剧烈的波动，此现象即为著名的龙格现象。

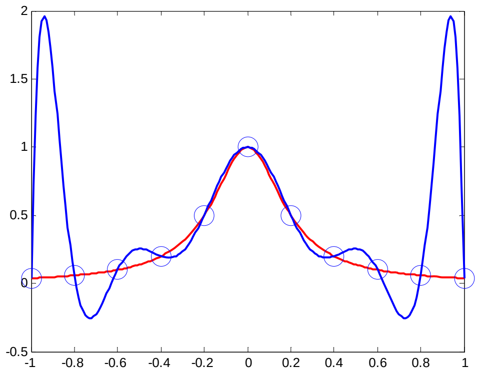



图 1.3 十次多项式插值效果，出现龙格现象

## 三） 算法代码样例

如下代码可以用于验证函数的各次多项式插值效果，其中 m 为多项式次数，n 根据 m 的取值自适应调整从而确定插值节点的数量。

```
clc; clear all; close all;
x = -1:0.01:1;
y = 1./(1+25*x.^2);
figure; plot(x,y,'r','linewidth',2);
m = 2;
n = m+1;
```

```
xp = -1:2/(n-1):1;
yp = 1./(1+25*xp.^2);
hold on
plot(xp,yp,'o','markersize',15);
p = polyfit(xp,yp,m);
ys = polyval(p,x);
plot(x,ys,'linewidth',2)
hold off
```

## 四） 作业

选择一个函数（不能是偶函数，最好也不是奇函数），演示多项式插值效果（可以尝试使用其它插值实现函数），要求龙格现象比较明显。观察和分析结果，完成实验报告。

# 实验二 曲线的样条插值

## 一） 实验目的

鉴于对函数曲线应用多项式进行整体插值无法实现良好的近似效果，科学工作者们随后提出了分段低次插值的思路。但简单的分段低次插值在分段点处会出现不光滑的现象，无法满足很多工程问题的要求。经过进一步“艰辛”的研究，人们提出了“样条”插值思想。古代造船业中应用坚韧的竹条获得光滑的船体，其思想和“样条”插值是想通的。样条函数的研究始于 20 世纪中叶，到了 60 年代它与计算机辅助设计相结合，在外形设计方面得到成功的应用。样条理论已成为函数逼近的有力工具。它的应用范围也在不断扩大，不仅在数据处理、数值微分、数值积分、微分方程和积分方程数值解等数学领域有广泛的应用，而且与最优控制、变分问题、统计学、计算几何与泛函分析等学科均有密切的联系。

本实验的目的是让大家重走分段低次插值到样条这条科研发展之路，理解图形学中最重要的工具之一“样条插值”的妙处。

## 二） 程序思想

仍然以实验一中给出的函数

$$f(x)=\frac{1}{1+25x^2}$$

为例，首先根据 11 个等距数据点将曲线分段为十段进行线性插值（一次多项式插值），效果如图 2.1 所示。与图 1.2（二次多项式插值）及图 1.3（十次多项式插值）相比，可以看出分段线性插值效果要好得多。实际上，对于某些工程任务如数值积分，这样的分段低次插值也可以用于实际计算。

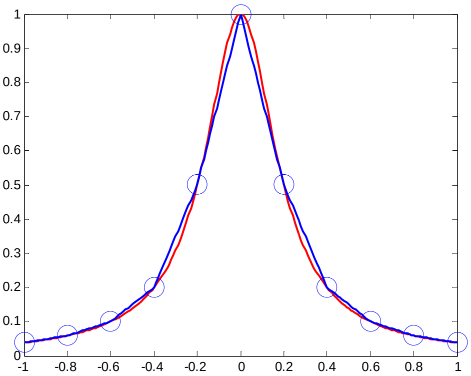



图 2.1 分段线性插值

然而对于更多的工程任务如产品设计，我们希望所设计出来的产品具有流线

型的效果，上面的分段线性插值就无法满足这样的要求了。为此，可以考虑其它分段低次插值，比较常用的是分段三次插值。在不同的分段区间上插值多项式一般也不相同，单它们可以在插值节点上不仅具备相等的函数值，也具备相等的一阶与二阶导数值。这样，插值曲线在整体上能够具有二阶光滑性，满足了大部分实际工程问题的需求。

我们通过 MATLAB 的一维插值函数 interp1 来实现样条插值. 该函数形如 YI = INTERP1(X,Y,XI,METHOD)，其中的 METHOD 参数取为'linear'时即可得上面的分段线性插值结果，取值为'spline'时便是样条插值效果，如图 2.2 所示。

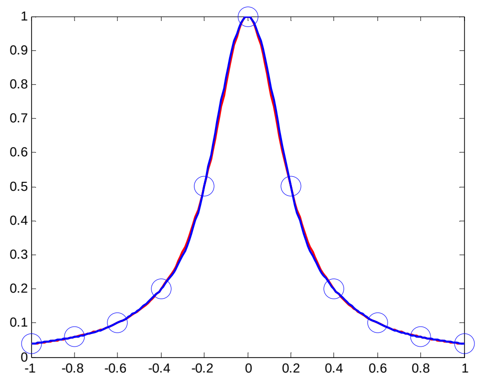



图 2.2 三次样条插值

可以看出，样条插值具有良好的近似效果。由此同学们淘得了图形学这门课程的“第一桶金”，此后的样条曲面和样条曲线的思想是相同的。以后的学习中大家还可能遇到有理样条等更复杂的样条近似方法，都是以这里的样条插值方法进一步发展而来的。

## 三） 算法代码样例

```
x = -1:0.01:1;
y = 1./(1+25*x.^2);
figure; plot(x,y,'r','linewidth',2);
xp = -1:0.2:1;
yp =    1./(1+25*xp.^2);
hold on
plot(xp,yp,'o','markersize',15);
ys = interp1(xp,yp,x,'spline');
plot(x,ys,'linewidth',2);
hold off
```

## 四） 作业

MATLAB 语言实现了更为强大的三次样条插值的函数 cscvn（可以实现对封闭曲线的近似，从而实现更丰富的几何造型）。 作为练习，要求大家熟悉和掌握该函数的使用。

（1） 调用 MATLAB 的 help 命令，学习 cscvn 的使用方法。使用 fnplt(C) 命令绘制出样条插值曲线。

（2）应用 C = cscvn(B); coefs = fnbrk(C,’coef’); 语句段查看各段多项式的系数。并将分段多项式系数组合成多项式，分别绘制出曲线，与（1）中所得到的曲线进行对比，验证是否一致。

# 实验三 Matlab 图形动画实现与控制

## 一） 实验目的

对于一些复杂的图形学设计任务，仅在三维空间中显示设计结果有时尚显不足，这次需要增加一个时间维度即应用动画技术来观察设计效果。从这个意义上来说，动画技术是图形学的重要工具之一。本次实验首先学习动画技术的实现，掌握 MATLAB 中生成 avi 格式动画文件的方法，并通过参数控制动画的速度。最后通过 GUI 的按钮，实现不同动画速度的控制。

## 二） 程序思想

MATLAB 中生成 avi 文件的基本代码如下:

```
aviobj=avifile(’mymovie.avi’,’fps’,20);     %定义一个 avi 文件
for i=1:n
      XXXXXXX; %在当前窗体上生成一帧图像
      frame=getframe(gca); %获得一帧图像
      aviobj=addframe(aviobj,frame); %并加到动画视频文件中
end
aviobj=close(aviobj); %关闭文件，结束数值仿真模拟过程.
```

在上面的代码段中，首先生成一个 mymovie.avi 文件，然后一般通过循环语句得到动画视频文件的各帧内容，最后关闭文件句柄。

avifile(’mymovie.avi’,’fps’,20) 命令中’fps’后的参数具体含义为每秒钟播放的帧数，通过参数设置便可以控制动画播放速度。如果通过 GUI 来实现动画，可以考虑使用暂停函数 pause(t)并通过设置不同的暂停时间 t 来实现对动画速度的控制。

## 三） 算法代码样例

以下样例实现对曲线的平滑去噪过程的动画，随着平滑过程的进行，曲线变得愈来愈光滑。在代码中，我们在窗口的适当位置显示帧数信息。图 3.1 给出了最后一帧的结果。

```
aviobj=avifile('mymovie.avi','fps',5);
A=load('map.txt');
```

```
v=size(A(:,1));
n=v(1);
x=A(:,1)+(rand(n,1)-0.5)*20;
y=A(:,2)+(rand(n,1)-0.5)*20;
for k = 1 : 100
    xn(1)=0.25*x(n)+0.5*x(1)+0.25*x(2);
    yn(1)=0.25*y(n)+0.5*y(1)+0.25*y(2);
    xn(n)=0.25*x(n-1)+0.5*x(n)+0.25*x(1);
    yn(n)=0.25*y(n-1)+0.5*y(n)+0.25*y(1);
    for i = 2 : n-1
        xn(i) = 0.25*x(i-1) + 0.5*x(i) + 0.25*x(i+1);
        yn(i) = 0.25*y(i-1) + 0.5*y(i) + 0.25*y(i+1);
    end
    for i = 1 : n
        x(i) = xn(i);
        y(i) = yn(i);
    end
    drawnow;
    plot(x,y,'k','linewidth',2)
    axis equal
    axis([280,480,220,420])
    numofpic=num2str(k);
    text(440,380,['第  ', numofpic, '  帧']);
    frame=getframe(gca);
    aviobj=addframe(aviobj,frame);
end
aviobj=close(aviobj);
```

## 四） 作业

1）将课堂演示过的动画效果制作成 avi 文件，并随堂演示（至少两个动画文件）。

2） 设计一个 GUI，用按钮控制一段动画（以前编写的）的播放。

3） 在 GUI 上加上另一个按钮，控制动画播放的速度。

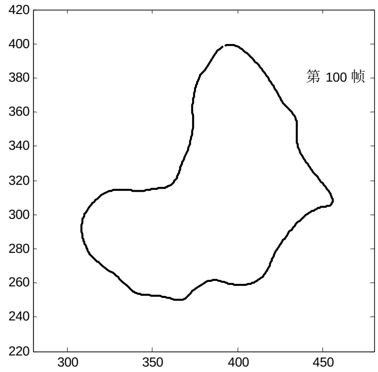

第 100 帧
420
400
380
360
340
320
300
280
260
240
220
300
350
400
450

图 3.1 动画最后一帧

# 实验四 Bezier 曲线与 B 样条曲线的实现与拼接

## 一） 实验目的

Bezier 曲线是计算机图形学中相当重要的参数曲线，在一些比较成熟的位图软件中也有 Bezier 曲线工具，如 PhotoShop 等。Bezier 曲线依据四个位置任意的点坐标绘制出的一条光滑曲线，其有趣之处在于它的“皮筋效应”，也就是说，随着点有规律地移动，曲线将产生皮筋伸引一样的变换，带来视觉上的冲击。1962 年，法国数学家 Pierre Bézier 第一个研究了这种矢量绘制曲线的方法，因此绘制出来的曲线就用他的姓氏来命名是为贝塞尔曲线。

B 样条是样条曲线一种特殊的表示形式。它是 B 样条基曲线的线性组合。B-样条是 Bezier 曲线的一种一般化，可以进一步推广为非均匀有理 B 样条（NURBS），使得我们能给更多一般的几何体建造精确的模型。B 样条曲线具有几何不变性、凸包性、保凸性、变差减小性、局部支撑性等许多优良性质，是目前 CAD 系统常用的几何表示方法，因而基于测量数据的参数化和 B 样条曲面重建是反求工程的研究热点和关键技术之一。

本实验在课堂学习以上两类曲线的基本理论与方法的基础上，实现对空间 Bezier 曲线与 B 样条曲线拼接技术，从而达到巩固课堂基本知识的目的，并通过实验结果进一步理解这两类曲线的相关性质。

## 二） 程序思想

三次 Bezier 曲线的计算公式形如

$$\begin{cases} x(t) = (1-t)^3 x_0 + 3t(t-1)^2 x_1 + 3t^2(1-t)x_2 + t^3 x_3 \\ y(t) = (1-t)^3 y_0 + 3t(t-1)^2 y_1 + 3t^2(1-t)y_2 + t^3 y_3 \end{cases}$$

给定 4 个控制点 P1、P2、P3、P4，绘制出的三次 Bezier 曲线记为 C1，再根据另外 4 个控制点 P4、P5、P6、P7 可以绘制出三次 Bezier 曲线 C2。关键问题是这 7 个顶点满足什么条件时两条曲线在 P4 点光滑连接？

由 Bezier 曲线的性质：Bezier 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 P3、P4、P5 在一条直线上时，两条三次 Bezier 曲线在 P4 点光滑连接。

三次 B 样条曲线的矩阵形式如下：

$$P(t) = \frac{1}{6}[t^3 \quad t^2 \quad t \quad 1]\begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix}\begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{bmatrix}$$

给定 4 个控制点 P1、P2、P3、P4，绘制出的三次 B 样条曲线记为 C1，再根据 4 个控制点 P2、P3、P4、P5 可以绘制出三次 B 样条曲线 C2。B 样条曲线段的连接比较自然，可以验证以上两条具有三个共同控制顶点的两条 B 样条曲线可以自然实现光滑拼接。

此外，Bezier 曲线中的每个控制点都会影响整个曲线的形状(缺点: 不容易做局部修改; 拼接比较复杂)，而 B 样条中的控制点只会影响整个曲线的一部分，显然 B 样条提供了更多的灵活性。

## 三） 算法代码样例

以下是绘制空间三次 Bezier 曲线的基础代码，运行结果如图 4.1 所示。

```
P=[1    2    5    6; 1.5    7    11    5; 2    3    6    5.5];
plot3(P(1,:),P(2,:),P(3,:),'o')
hold on
B=[-1 3 -3 1;3 -6 3 0;-3 3 0 0;1 0 0 0];
for t=0:0.01:1
    T= [t^3    t^2    t    1];
    P1=T*B*P(1,:)'
    P2=T*B*P(2,:)'
    P3=T*B*P(3,:)'
    plot3(P1,P2,P3,'.')
    hold on
end
hold off
```

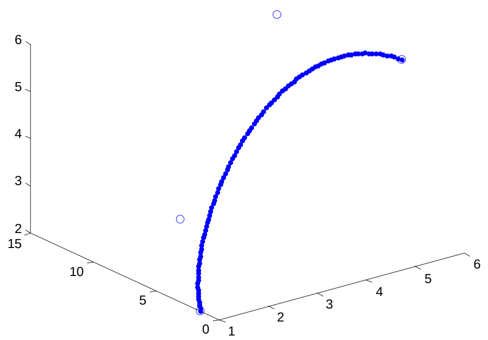



图 4.1  空间三次 Bezier 曲线

以下是代码实现二次 B 样条曲线的拼接，结果如图 4.2 所示。

```
P=[1    2    5    5.5;    1.5    7    11    6; 3    5    6.8    2.1];
plot3(P(1,1:3),P(2,1:3),P(3,1:3),'o')
hold on
plot3(P(1,2:4),P(2,2:4),P(3,2:4),'+')
hold on
B=[1    -2    1;-2    2    0;1    1    0];
for t=0:0.01:1
T= [t^2    t    1];
P1=1/2*T*B*P(1,1:3)';
P2=1/2*T*B*P(2,1:3)';
P3=1/2*T*B*P(3,1:3)';
plot3(P1,P2,P3,'r.')
hold on
Q1=1/2*T*B*P(1,2:4)';
Q2=1/2*T*B*P(2,2:4)';
Q3=1/2*T*B*P(3,2:4)';
plot3(Q1,Q2,Q3,'b*')
end
hold off
```

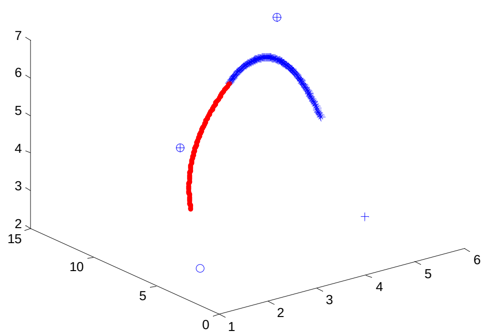



图 4.1 二次 B 样条曲线的拼接

借鉴以上两个代码段，可以完成作业任务。

## 四） 作业

1） 基于曲线拼接技术，在二维平面上分别绘制出闭合的 Bezier 曲线和闭合的 B 样条曲线。

2） 将样例代码中的绘制空间 Bezier 曲线和空间 B 样条曲线的部分改写为子函数 Bezier2by2(X,Y,Z)和 Bspline2by2(X,Y,Z)

3） 设计程序，分别完成三次 Bezier 曲线（空间曲线）的拼接与 B 样条曲线（空间曲线）的拼接。

附加题：

4） Bezier 曲线几何意义的动画效果。

5） 实现在二维平面上能够通过鼠标移动 Bezier 曲线和 B 样条曲线控制点的程序。

6） 移动一个控制点，比较三次 Bezier 与 B 样条曲线的变化性质。

# 实验五 Bezier 曲面与 B 样条曲面的实现与拼接

## 一） 实验目的

Bezier 曲线与 B 样条曲线的理论和方法可以进一步推广到三维空间中，由此形成 Bezier 曲面与 B 样条曲面。本实验在课堂学习这两类曲面的基本理论与方法的基础上，实现对 B 样条曲面拼接技术，并尝试完成对 Bezier 曲面的拼接技术（选做，需要检索文献资料，理解光滑拼接的条件），从而达到巩固课堂基本知识的目的，并通过实验结果进一步理解这两类曲面的相关性质。

## 二） 程序思想

与 B 样条曲线类似，B 样条曲面天生具备能够光滑拼接的优势：两片 B 样条曲面，若有 12 个共同控制点，则自然能够光滑拼接。Bezier 曲面的光滑拼接相对复杂，必要条件是需要有 4 个共同控制点，充要条件需要检索相关的文献进行尝试性了解和掌握。

## 三） 算法代码样例

以下是绘制三次 B 样条曲面的子函数。

```
function Bspline3by3(X,Y,Z)
M = 1/6 * [
    -1   3 -3   1
     3 -6   3   0
    -3   0   3   0
     1   4   1   0
     ];
  B(:,:,1) = X;
  B(:,:,2) = Y;
  B(:,:,3) = Z;
  for   i = 1:3
      Q(:,:,i) = M*B(:,:,i)*M';
      for u = 0:10
          for v = 0:10
              u3 = (u/10)^3; u2 = (u/10)^2; u1 = u/10;
              v3 = (v/10)^3; v2 = (v/10)^2; v1 = v/10;
```

```
                    S(u+1,v+1,i) = [u3 u2 u1 1] * Q(:,:,i) * [v3 v2 v1 1]';
                end
            end
        end
        surf(S(:,:,1),S(:,:,2),S(:,:,3))
```

以下代码调用该子函数，实现对 B 样条曲面的绘制，效果如图 5.1 所示。

```
[X,Y] = meshgrid([0:1/3:1]);
Z = sin(X+Y)+hilb(4)*1.5;
plot3(X,Y,Z)
hold on
plot3(Y,X,Z)
plot3(X,Y,Z,'o')
Bspline3by3(X,Y,Z)
hold off
```

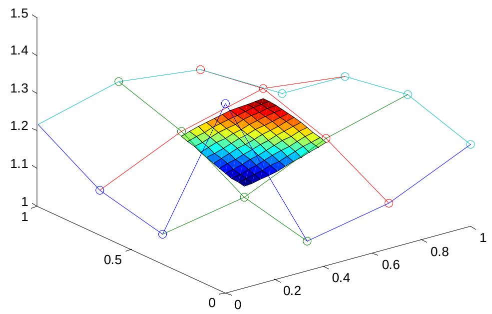



图 5.1 三次 B 样条曲面绘制

以下是绘制三次 Bezier 曲面的子函数。

```
function Bezier3by3(X,Y,Z)
M = [
    -1   3 -3   1
     3 -6   3   0
    -3   3   0   0
     1   0   0   0
     ];
 B(:,:,1) = X;
 B(:,:,2) = Y;
```

```
B(:,:,3) = Z;

for   i = 1:3

      Q(:,:,i) = M*B(:,:,i)*M';

      for u = 0:10
            for v = 0:10
                  u3 = (u/10)^3; u2 = (u/10)^2; u1 = u/10;
                  v3 = (v/10)^3; v2 = (v/10)^2; v1 = v/10;
                  S(u+1,v+1,i) = [u3 u2 u1 1] * Q(:,:,i) * [v3 v2 v1 1]';
            end
      end
end
surf(S(:,:,1),S(:,:,2),S(:,:,3))
```

以下代码调用该子函数，实现对 Bezier 曲面的绘制，效果如图 5.2 所示。

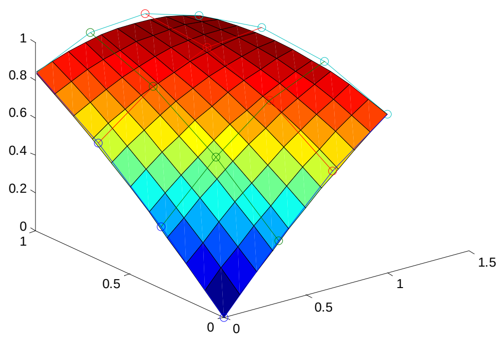



图 5.2  三次 Bezier 样条曲面绘制

基于以上代码，可以完成作业。

## 四） 作业

设计程序，完成三次 B 样条曲面的拼接；尝试完成三次 Bezier 曲面的拼接。

# 实验六 几何造型

## 一） 实验目的

研究模型的目的是为了更好的实现计算机存储、传输与重新绘制。模型构造是把基本几何单元作为基本要素，采用一些有效的方法把这些要素组合起来，以更好的实现几何体在计算机内存储、传输、变换与显示等。

几何造型的主要三种模型为线框模型、表面模型和实体模型。本实验主要了解和掌握三种模型的建立和着色。

## 二） 程序思想

线框(wireframe)模型是用顶点与邻边表示形体的一种模型。线框模型是计算机图形学领域最早用来表示形体的模型，目前也被广泛使用。这种表示方法结构简单、易于理解，又是表面和实体模型的基础。对于多面体，因为主要靠棱边表达图形信息，所以适合于使用线框模型。

表面(surface)模型是用棱边围成的部分来定义形体表面，由面的集合来定义形体。表面模型是在线框模型的基础上，增加了有关面边的信息以及表面特征等。表面模型的一般用表来描述（例见表 6.1）。

表 6.1 一个具体的表面模型结构与数据

| 面编号 | 棱编号 | | | 棱编号 | 定点编号 | | 定点 | 坐标值 | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 一 | 1 | 2 | 3 | 1 | [1] | [2] | [1] | 1 | 1 | 1 |
| 二 | 1 | 5 | 4 | 2 | [1] | [3] | [2] | 1 | 5 | 1 |
| 三 | 3 | 4 | 6 | 3 | [2] | [3] | [3] | 3 | 3 | 1 |
| 四 | 2 | 5 | 6 | 4 | [1] | [4] | [4] | 1.5 | 1.5 | 3 |
| | | | | 5 | [2] | [4] | | | | |
| | | | | 6 | [3] | [4] | | | | |

表面模型确切的描述了表面信息，但是表面模型有时还不能明确表面的哪一侧存在实体。在一些实际应用领域（例如 CAD\CAM 领域），完整的表达实体信息是非常重要的。目前，几何体的实体造型技术已经发展成熟，成熟的实体造型技术就是建立在实体模型基础上。实体模型完整的定义了三维实体，能够在计算机上进行准确的处理。

三维实体模型的最简单形式是逐点描述方法，就是把该物体在空间上占据的点记载下来（点之间的距离满足视觉需要），存储再现。但是，由于需要记载的点太多，这种方法是不可行的。所以，要研究各种模型，使模型尽量减少描述物

体的工作量。

### 三） 算法代码样例

以下代码首先绘制 8 个顶点的六面体的线框图，进行表面着色得到表面模型。程序运行结果在图 6.1 中给出。

```
V = [
    -2 -2 -2
    -2  2 -2
    -2 -2  2
    2 -2 -2
    2  2 -2
    2 -2  2
    -2 2  2
    2  2  2
    ];

line([V(1,1) V(2,1)],[V(1,2) V(2,2)], [V(1,3) V(2,3)]);
line([V(1,1) V(3,1)],[V(1,2) V(3,2)], [V(1,3) V(3,3)]);
line([V(1,1) V(4,1)],[V(1,2) V(4,2)], [V(1,3) V(4,3)]);
line([V(2,1) V(5,1)],[V(2,2) V(5,2)], [V(2,3) V(5,3)]);
line([V(2,1) V(7,1)],[V(2,2) V(7,2)], [V(2,3) V(7,3)]);
line([V(3,1) V(6,1)],[V(3,2) V(6,2)], [V(3,3) V(6,3)]);
line([V(3,1) V(7,1)],[V(3,2) V(7,2)], [V(3,3) V(7,3)]);
line([V(4,1) V(5,1)],[V(4,2) V(5,2)], [V(4,3) V(5,3)]);
line([V(4,1) V(6,1)],[V(4,2) V(6,2)], [V(4,3) V(6,3)]);
line([V(5,1) V(8,1)],[V(5,2) V(8,2)], [V(5,3) V(8,3)]);
line([V(6,1) V(8,1)],[V(6,2) V(8,2)], [V(6,3) V(8,3)]);
line([V(7,1) V(8,1)],[V(7,2) V(8,2)], [V(7,3) V(8,3)]);

for n = 1:8
```

```
    hold on
    plot3(V(n,1),V(n,2),V(n,3),'ro','Linewidth',2);
    text(V(n,1),V(n,2),V(n,3),['    ' num2str(n)]);
end
view(3)
axis off

S1 = [V(1,:)' V(2,:)' V(5,:)' V(4,:)' ];
fill3(S1(1,:),S1(2,:),S1(3,:),[0,0.2,0.5]);
S2 = [V(3,:)' V(7,:)' V(8,:)' V(6,:)' ];
fill3(S2(1,:),S2(2,:),S2(3,:),[0.5,0.1,0.3]);
S3 = [V(1,:)' V(3,:)' V(7,:)' V(2,:)' ];
fill3(S3(1,:),S3(2,:),S3(3,:),[0.4,0.4,0.4]);
S4 = [V(2,:)' V(7,:)' V(8,:)' V(5,:)' ];
fill3(S4(1,:),S4(2,:),S4(3,:),[0.6,0.6,0.6]);

hold off
```

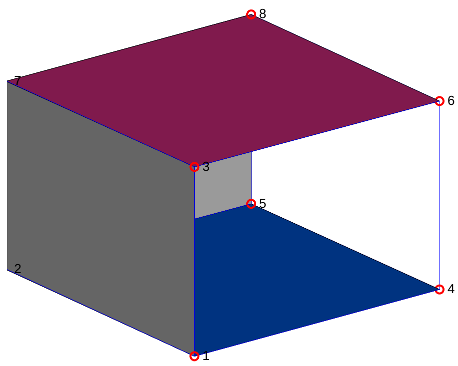



图 6.1 六面体的线框描述与表面模型

以下代码通过平移扫描造型法实现一个实体模型，程序运行结果在图 6.2 中给出。

```
x = 0:0.1:3;

y = sin(x);
```

```
for i = 1:50
    z = ones(1,31)*i;
    plot3(x,y,z)
    hold on
end
 hold off
```

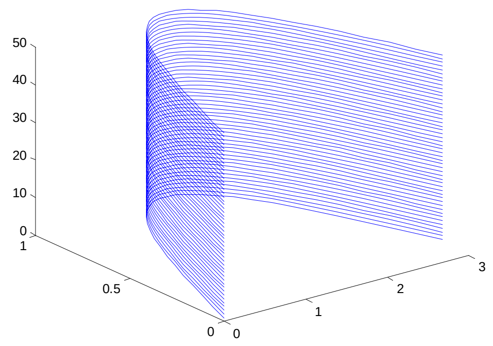



图 6.2 平移扫描法建立实体模型

## 四） 作业

构造一个五面体，应用 patch 命令对五面体涂色；参考表 6.1，写出该五面体的结构与数据。

# 实验七 隐藏面处理

## 一） 实验目的

经过几何造型以后，物体的空间位置以及形状等数据存储在计算机中，这些数据唯一的代表着这个物体。但是，要把这个物体绘制出来，显示在二维平面上却有多种显示方法。在屏幕上绘制物体只能绘制出该物体的一个投影面，绘制出物体的哪个（投影面）部分与设定的观察者所在位置有关。我们把观察者所在的位置称为视点。

三维物体显示在二维平面上，一定有一部分没有被显示出来，没有被显示出来的表面叫做被隐藏面。本实验了解和掌握隐藏面的处理效果，通过在 GUI 中对相关参数的控制实现对隐藏面显示度的控制。

## 二） 程序思想

在几何造型中，实体模型边界表示法为每个面规定了一个正方向，定义垂直于该面并且背离物体的方向为该面的正方向。

对于单个凸多面体，可以按如下步骤计算不可见面。

1）求一个面所在平面的法向量 n；

2）求这个平面的视线向量 v ；

3）计算视线向量与法向量的数量积；

根据 n 与 v 数量积的符号判断该面是否被遮挡，符号为正，被遮挡；符号为负，没有被遮挡。

在 MATLAB 中，隐藏面的处理计算可通过相关函数中的参数来设置，比如 mesh 函数，其具体形式如下：

h=mesh(Z,'FaceAlpha',alpha)

alpha = 0， 则显示隐藏面；

alpha=0.5，隐藏线的显示度为 0.5；

alpha=1，默认设置,隐藏线不显示。

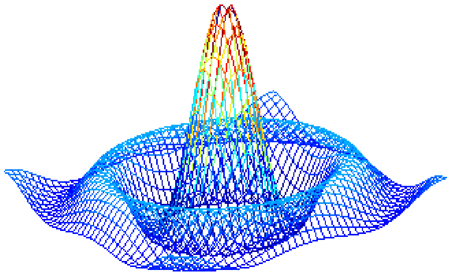
## 三） 算法代码样例

以下代码利用 mesh 函数绘制一个曲面网格，对其中的参数 alpha 进行控制，可实现对隐藏线显示度的控制。

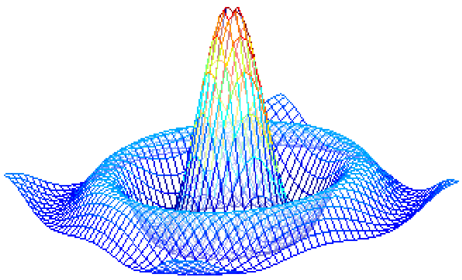
```
[X,Y]=meshgrid(-10:.5:10);
```

```
R=sqrt(X.^2+Y.^2);
Z=sin(R)./R;
alpha = 1;
h=mesh(Z,'FaceAlpha',alpha)
axis off
```

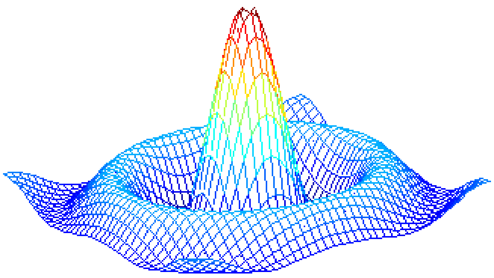
图 7.1 给出了三个不同参数设置下隐藏线的显示效果。



(a) alpha=0，显示隐藏线



（b）alpha=0.5，隐藏线的显示度为 0.5



(c) alpha=1，默认设置,隐藏线不显示

图 7.1 隐藏线效果

## 四） 作业

设计一个 GUI，通过滚动条控制参数 alpha 取值，演示隐藏面处理效果。

# 实验八 光照效果

## 一） 实验目的

人之所以能够看见物体，就是因为物体表面对光进行反射，反射光进入人眼。为了使物体的图形具有真实感，需要给物体加上细腻的光照效果。现在常见的光照效果一般由 Phong 光照模型生成，该模型包含如下三个成分。

（1） 镜面反射光

镜面反射光遵循反射定律，入射角等于反射角。产生镜面反射的条件是入射光一般是平行光，并且物体表面比较光滑。事实上，绝对光滑的物体是没有的，所以，反射光一般散布在反射方向周围的小范围内。

（2） 漫反射光

不光滑的物体在接受光照时，向多个方向反射，这样的反射叫做漫反射。漫反射光的强度可以用朗伯(Lambert)定律来计算。

（3） 环境反射光

环境反射光是由周围物体（包括墙面）多次反射后照射在该物体上的光。白天，没有被太阳直射的物体上接受的光就可以近似称为环境光。物体可见表面上得到的环境光是一样的。

本实验的目的是通过给几何造型依次添加镜面反射光、漫反射光和环境反射光，从而熟悉和掌握 Phong 光照模型。

## 二） 程序思想

以三棱柱为例，平行光光源在[1，1，1]方向，视点在默认方向，计算三棱柱上各个可见面的反射光亮度。首先，绘制出没有光照效果的三棱柱。假设光源是平行光源，在 [1，1，1]方向。视点方向为默认的 v=[0.5272,0.6871, -0.5000]（方位角-37.5 度，仰角 30 度）。

设入射光的亮度为[0.9, 0.9, 0.9],物体光滑程度为 10，镜面反射系数为 1。先计算面 1 的镜面反射方向。取面 1 的 3 个顶点（-0.5000，-0.8660，0）、（1，0，0）、（1，0，1），根据这 3 个顶点计算法线方向。 如面 1 的法线方向为 v=[0.8660，-1.5000，0]。计算出在面 1 上的反射向量，然后求与视向量的夹角，再计算出该面的镜面反射光亮度。同理，计算出各个面的反射光亮度。

为了添加漫反射光照效果。取面 1 的 3 个顶点（-0.5000，-0.8660，0）、（1，0，0）、（1，0，1），根据这 3 个顶点，计算法线方向。然后计算该面的漫反射光亮度。最后把漫反射光的亮度绘制上去。

环境反射光由于成分单一，容易操作，比如把各面亮度加上 0.1 即可绘制出。

### 三） 算法代码样例

以下代码段生成一个三棱柱，见图 8.1 所示。

```
[X,Y,Z]=cylinder(1,3);
h=surf(X,Y,Z,'FaceColor',[0.5,0.5,0.5])
```

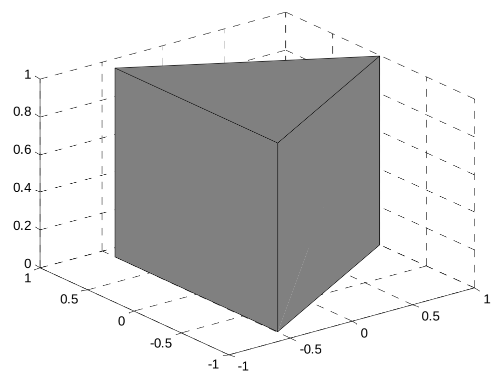



图 8.1 需添加光照效果的三棱柱

取面 1 的 3 个顶点，使用下面程序计算法线方向。由此计算出在面 1 上的反射向量，然后求与视向量的夹角，从而计算出该面的镜面反射光亮度。

```
[X,Y,Z]=cylinder(1,3);
p1=[X(1,3),Y(1,3),Z(1,3)];
p2=[X(1,4),Y(1,4),Z(1,4)];
p3=[X(2,4),Y(2,4),Z(2,4)];
v=cross(p3-p2,p2-p1)
```

输出结果为

```
v =     -0.8660     1.5000             0
```

取面 1 的 3 个顶点，使用下面程序计算法线方向。

```
[X,Y,Z]=cylinder(1,3);
for i=1:3
p1=[X(1,i),Y(1,i),Z(1,i)];
p2=[X(1,i+1),Y(1,i+1),Z(1,i+1)];
p3=[X(2,i+1),Y(2,i+1),Z(2,i+1)];
v(i,:)=cross(p2-p1,p3-p2);
end
```

输出结果如下。

```
v =
```

| | | |
|---|---|---|
| 0.8660 | 1.5000 | 0 |
| -1.7321 | 0.0000 | 0 |
| 0.8660 | -1.5000 | 0 |

## 四） 作业

实现光照效果一个 GUI，通过文本框输入相关的参数（入射光方向 $L$、入射光亮度 $I_p$、表面漫反射系数 $K_d$、环境光亮度 $I_a$、环境光反射系数 $K_a$）。以三棱柱几何造型，添加

1) 漫反射光照
2) 环境反射光
3) 附加题：添加镜面反射光（添加相关参数文本框）；添加几何体可旋转效果（添加旋转角度）。

# 实验九 不规则物体建模

## 一） 实验目的

计算机图形学擅长于创建规则物体。不规则物体在自然界是大量存在的，如山、水、树木、云、火等；不规则物体在日常生活中也是无处不在的，如衣服等。图形学绘制不规则物体最常使用的方法是随机的方法。目前，正在研究使用图像绘制图形，充分的利用图像的不规则特性。

关于不规则物体建模，分形是一个相对成熟的图形学分支，非真实感图形绘制是一个新的图形学分支。在数学上，分形一般定义为具有分数维数的点的集合；在有的学科领域，分形定义为混沌吸引子；在图形学中，一般把分形定义为具有自相似特性的图形点集。可以使用线性迭代函数系统绘制分形图，例如，绘制树、山等。本实验目的是熟悉和掌握通过分形技术实现不规则物体建模技术。

## 二） 程序思想

迭代函数系统简记为 IFS，是英文 Iterated Function Systems 的缩写。线性迭代函数系统的每个迭代函数表达式都是线性的，所以称为线性迭代表达式。关于线性迭代函数的迭代，一般使用下面的迭代算法。

1.设定一个起始点（x0，y0）及总的迭代步数。

2.以概率 P 选取仿射变换 W，形式为

　　X1=a x0+b y0 +e

　　Y1=c x0+d y0+f

3.以 W 作用点（x0，y0），得到新坐标（x1，y1）。

4.令 x0=x1，y0=y1。

5.在屏幕上打出（x0，y0）。

6.重返第 2 步，进行下一次迭代，直到迭代次数大于总步数为止。

## 三） 算法代码样例

以下 IFS 算法代码运行后可以生成一颗树（见图 9.1）。

```
a=[0.195 -0.488 0.344 0.433 0.4431 0.2452 0.25;
    0.462 0.414 -0.252 0.361 0.2511 0.5692 0.25;
    -0.058 -0.07 0.453 -0.111 0.5976 0.0969 0.25;
    -0.035 0.07 -0.469 -0.022 0.4884 0.5069 0.2;
    -0.637 0 0 0.501 0.8562 0.2513 0.05];
x0=1;y0=1;
for i=1:10000
    r=rand;
    if r<=0.25
        x1=a(1,1)*x0+a(1,2)*y0+a(1,5);
        y1=a(1,3)*x0+a(1,4)*y0+a(1,6);
    end
    if r>0.25 & r<=0.5
        x1=a(2,1)*x0+a(2,2)*y0+a(2,5);
        y1=a(2,3)*x0+a(2,4)*y0+a(2,6);
    end
    if r>0.5 & r<=0.75
        x1=a(3,1)*x0+a(3,2)*y0+a(3,5);
        y1=a(3,3)*x0+a(3,4)*y0+a(3,6);
    end
    if r>0.75 & r<=0.95
        x1=a(4,1)*x0+a(4,2)*y0+a(4,5);
        y1=a(4,3)*x0+a(4,4)*y0+a(4,6);
    end
    if r>0.95 & r<=1
        x1=a(5,1)*x0+a(5,2)*y0+a(5,5);
        y1=a(5,3)*x0+a(5,4)*y0+a(5,6);
    end
    x0=x1;y0=y1;
    plot(x1,y1);hold on;
end
```

## 四） 作业

1)　修改代码样例中的参数，绘制另一颗树来。

2)　改动程序，制作出一个树叶摇动的动画。

3)　附加题，在三维空间显示一幅图像的图形效果，并尝试在 XOY 平面放置对应的图像（挑战：放置一幅彩色图像）。

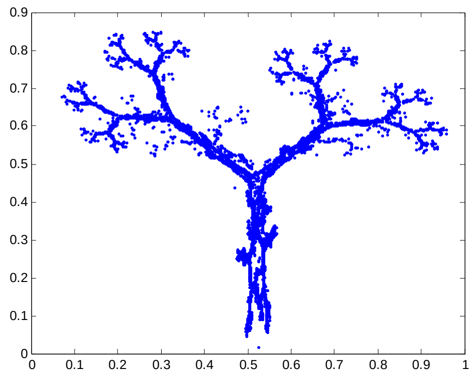



图 9.1　分形树

# 参考文献

[1] 于万波．基于 MATLAB 的计算机图形学与动画技术．清华大学出版社, 2007.

[2] 倪明田，吴良芝．计算机图形学．北京大学出版社，2012.

[3] http://cg.cs.tsinghua.edu.cn/course/resource_main.htm (清华大学计算机图形学教学网站)

附录 1：实验报告格式

# 实验名称（黑 2 号）

姓名，学号　宋体(小四)

1. 实验目的(黑小四)

   宋体(小四),行间距 20 磅

2. 程序思想(黑小四)

   宋体(小四),行间距 20 磅

3. 实验结果(黑小四)

   宋体(小四),行间距 20 磅