

在 matlab 中实现旋转曲面的动画设计

向修栋¹, 付云芝^{2,3}

(1. 中国石油大学(华东) 数学与计算科学学院, 山东 东营 257062;

2. 海南大学 材料与化工学院, 海南 海口 570228;

3. 中国科学院化学研究所胶体与界面国家重点实验室, 北京 100082)

摘 要: 旋转曲面是工程设计与曲面造型中最为常见的曲面之一, 在现代工业中, 许多产品的形状都是由旋转曲面生成, 直观动态演示这些旋转曲面工件造型就显得尤为重要, 特别是考古学中计算机模拟复原技术中也用到旋转曲面。随着计算机技术高度发展, 各种设计软件就应运而生, 其中 matlab 已成为当今国际上科学界最具影响力、最有活力的软件。文中通过利用 matlab 强大的绘图功能和程序设计技巧, 依据 matlab 矩阵处理问题的特点, 以旋转曲面为研究对象, 探讨了 matlab 表现旋转曲面动画的几种方式方法。给出旋转曲面生成过程的动画程序, 并导出了任一曲线绕任意轴旋转的坐标间几何变换公式。通过运行程序, 结果表明运用 matlab 可以得到生动、完美、逼真的旋转曲面动画。特别地, 程序 3 可以单独绘制一条曲线的绕轴旋转的旋转曲面动画, 亦可绘制组合曲线旋转曲面动画。程序 1-3 都可嵌入到 GUI 界面演示系统中作为其中一部分。

关键词: 旋转曲面; 几何变换; 矩阵; matlab; 动画

中图分类号: TP391.72

文献标识码: A

文章编号: 1673-629X(2011)03-0052-04

Making Animation of Revolution Surface in Matlab

XIANG Xiu-dong¹, FU Yun-zhi^{2,3}

(1. College of Mathematics and Computational Science, China University of Petroleum, Dongying 257062, China;

2. Chemical and Material College, Hainan University, Haikou 570228, China;

3. Key Lab. of Colloids and Surfaces, Institute of Chemistry, Chinese Academy of Sciences, Beijing 100082, China)

Abstract: Revolution surface is one of the most familiar surfaces in the engineering design and surface modeling. Many product shapes are generated by the rotating surface in modern industry. Dynamic visual presentation of these rotating surface is particularly important in the work-piece shape. Especially rotating surface is also used to archaeological recovery techniques in computer simulation. With the high speed development of computer technology, the multifarious design soft-ware emerged immediately, and matlab which has become the most scientific, most influential and the most vivid software in the international scientific community all over world. By powerful drawing-graphics function, program design technique and data processing techniques in matlab. The some animation methods of coming into being revolution surfaces are discussed in this paper. The animation programs of the making revolution surface is compiled, and a general coordinate formula of geometric transformation is deduced about an arbitrary revolution curve around a rotation axis. The programs running shows that the vivid perfect realistic animations of rotating surfaces can be realized based on matlab. In particular, Procedure 3 can draw a rotation surface animation of a curve around an axis, and it can draw surface of revolution animation of combined curves around an axis too. The 1-3 Procedures can be embedded into the GUI interface as a part of the demonstration system.

Key words: rotation surface; geometry; matrix; matlab; animation

0 引 言

matlab 语言是以矩阵运算为基础, 把计算、可视

化、程序设计融合到一个简单易用的交互式工作环境中, 可实现工程计算、算法研究、符号运算、建模和仿真、原型开发、数据分析及可视化、科学和工程绘图、应用程序设计等功能^[1]。在 matlab 应用中, 特别是数据的可视化给人们的研究带来方便, matlab 制作的精确、逼真动画也备受关注。我们利用 matlab 探讨了极限动画的多种表达方法, 非常直观地展示了极限的渐进动态过程^[2]。通过对 matlab 不断地深入探讨和研究, 依据 matlab 矩阵运算的特点, 利用 matlab 的 axis, hold

收稿日期: 2010-07-26; 修回日期: 2010-10-28

基金项目: 海南省自然科学基金(807029); 海南省教育厅基金(Hjkj2009-13)

作者简介: 向修栋(1967-), 男, 山东汶上人, 硕士, 研究方向为应用数学及图形处理; 付云芝, 博士, 副教授, 硕士生导师, 研究方向为纳米技术和化学信息学研究。

on 和 for...end 等基本命令就可以实现 matlab 动画。hold on 是确保画图在同一坐标轴内实现,而 for...end 是实现动画关键,这些都只是实现动画的基本构架,要实现内容丰富更为逼真的动画,就必须拥有翔实的数据和必要的方式方法。旋转曲面是工程设计与曲面造型中最为常见的曲面之一,在现代工业中,许多产品的形状都是由旋转曲面生成,旋转曲面一直备受关注^[3-6]。动画表现手法丰富了传统静态视觉,各种方法层出不穷,其中文献[7]在 C# 中研究动态图。文中就以旋转曲面动画生成作为研究对象,在 matlab 中作进一步探讨。

1 探究旋转曲面动画表达方式

1.1 依次画出曲面矩阵的各列——真实再现旋转曲面形成过程

所谓旋转曲面就是一曲线绕一固定的轴旋转所得的曲面^[8],所以要想真实再现旋转曲面生成过程,就要把旋转过程中的每一位置的曲线画出来(见图1)。而这样的每一条曲线都对应着描述旋转曲面的 matlab 矩阵中的相应一列数据,故只要依次把矩阵的一列列数据地画出来,就可以很生动地展现旋转曲面生成过程。如在 $[-4, 4, -4, 4]$ 上作出椭圆抛物面 $9z = x^2 + y^2$ 生成过程。它的参数方程: $x = 3r \cos t; y = 3r \sin t; z = r^2, 0 \leq t \leq 2\pi; 0 \leq r \leq 3/4\sqrt{2}$,其动画程序(1)如下:

```
a=sqrt(2)*4/3;r=0:0.1:a;t=0:0.1:2*pi;[T,R]=
meshgrid(t,r);
X=3*cos(T).*R;Y=3*sin(T).*R;Z=ones(size
(T)).*R.^2;
axis([-10,10,-10,10,0,6]);hold on;n=size(Z,2);
for i=2:n
    plot3(X(:,i),Y(:,i),Z(:,i));drawnow;pause(0.3);
end
```

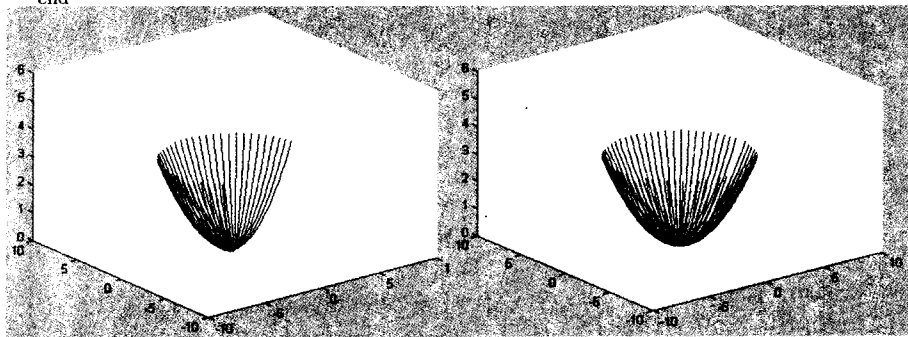


图1 旋转曲线在转动过程中的两幅截图

程序(1)虽然简单,旋转曲面的具体形成过程却很直观,程序运行速度亦非常快。由于不可能把旋转曲面上的每一条母线都画出来,所以画出的曲面是由有间隔的曲线构成的,显得不尽完美。即使增加曲线的密集程度,这种缺憾依然存在,但对于一般的演示说明还是有一定直观意义的。为了弥补以上不足,最直观的想法就是把画出的曲线间的间隙填实,具体操作上把用 plot3 画旋转曲线改为用 surf 或 mesh 画旋转曲面片,就能避免这类问题的发生。

1.2 依次绘制曲面矩阵 A 中相邻两列 A(:,i-1:i)组成曲面片——展示旋转曲面形成

在 matlab 中 surf 或 mesh 把描述曲面的矩阵的相邻数据点以曲面片形式画出来。如果把描述旋转曲面矩阵中的每相邻两列所描述的曲面片在同一坐标系里依次画出来,这样画出的旋转曲面动画更为逼真,仍以上例,将程序(1)主要改动如下语句: $t=0:0.1:2 \cdot \pi + 0.1$; $\text{surf}(X(:,i-1:i), Y(:,i-1:i), Z(:,i-1:i))$;将会得到如图2所示动画图形。

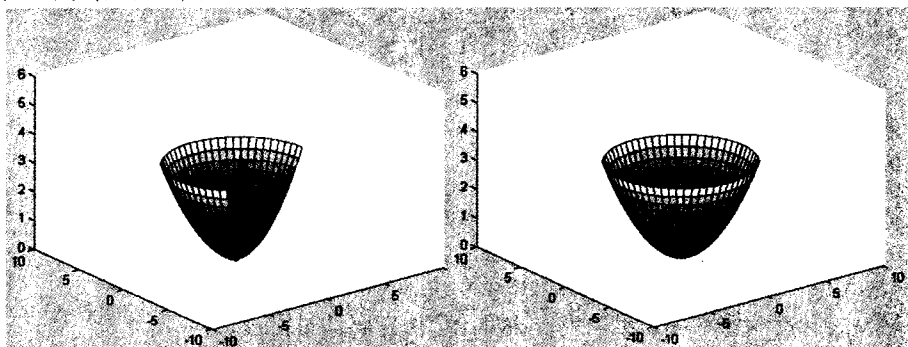


图2 画旋转曲面片过程中的两幅截图

由图2知,它比图1描述旋转曲面形成动画更接近实际,效果更好。由于 $A(:,i-1:i)$ 与 $A(:,[i-1:i])$ 都表示曲面矩阵 A 中相邻的两列,所以上述程序中的语句中的 $\text{surf}(X(:,i-1:i), Y(:,i-1:i), Z(:,i-1:i))$ 可改为 $\text{surf}(X(:,[i-1,i]), Y(:,[i-1,i]), Z(:,[i-1,i]))$,仍达到同样的效果。

1.3 依次将曲面矩阵的第 i 列后赋值为 NaN——实现旋转曲面动画效果

NaN 是 matlab 中非数值特殊量,它一般是在有正负无穷大参与的加减、乘法、除法的 $0/0, \infty/\infty$ 等式中将会产生 NaN^[9]。如果将表示曲面矩阵的数据的某一部分改为 NaN, matlab 的作图函数将会忽略 NaN 这些数据点,因而在曲面上就会出现孔洞。这样,依次将旋转曲面矩阵的第 i 列后

的各列赋值为 NaN, 并依次画出图形, 就会获得生动的旋转曲面动画效果 (见图 3)。如画出曲线 $x=0, y=5\sqrt{1+z^2/36}, -40/3 \leq z \leq 40/3$ 绕 z 轴旋转的旋转曲面 $36(x^2+y^2)-25z^2=900$ 动画, 程序 (2) 如下:

```
z=-15:0.1:15;y=5*sqrt(1+z.^2/36);[x1,y1,z1]=cylinder(y,40);z1=15*z1-7.5;
axis([-20,20,-20,20,-10,10]);hold on;n=size(z1,2);
for i=1:n;z11=z1;z11(:,i+1:n)=NaN;mesh(x1,y1,z11);drawnow;pause(0.2);
end
```

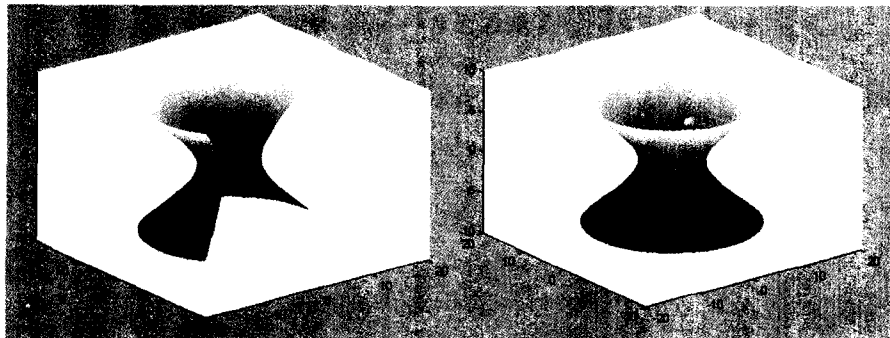


图 3 利用 NaN 实现旋转曲面动画的两幅截图

由于在 matlab 中, inf 是无穷大量, 绘图时, 画图函数同样也会将等于 inf 数据点忽略掉, 所以上述程序中的 NaN 可用 inf 代替。

1.4 利用几何变换——画出曲线绕任意旋转轴所形成的旋转曲面动画

为了程序的广适性, 探究曲线绕任意直线作为旋转轴旋转动画非常必要的。对于此类作图, 只要从最初的数据获得旋转后的曲面上点的坐标数据。为此, 将图像平移、旋转, 在此过程中, 空间点的坐标采用齐次形式^[10-12]。若曲线 $l_1: F(x, y, z) = 0, G(x, y, z) = 0$, 绕直线 $l_2: \frac{x-x_0}{l} = \frac{y-y_0}{m} = \frac{z-z_0}{n}$ 旋转, 要得到旋转曲面的坐标, 具体过程如下:

(1) 将 l_1 与 l_2 平移, 使 $P(x_0, y_0, z_0)$ 移至原点, 变换矩阵为 $T = [1 \ 0 \ 0 \ 0; 0 \ 1 \ 0 \ 0; 0 \ 0 \ 1 \ 0; -x_0 \ -y_0 \ -z_0 \ 1]$ 。

(2) 继续将新位置的 l_1 与 l_2 绕 x 轴旋转 α 度角, 使 l_2 落在 xoz 面内; 进一步将 l_1 与 l_2 绕 y 轴旋转 β 度角, 使 l_2 与 z 轴重合, 变换矩阵分别为: $R_x = [1 \ 0 \ 0 \ 0; 0 \ \cos \alpha \ -\sin \alpha \ 0; 0 \ \sin \alpha \ \cos \alpha \ 0; 0 \ 0 \ 0 \ 1]$, $R_y = [\cos \beta \ 0 \ -\sin \beta \ 0; 0 \ 1 \ 0 \ 0; \sin \beta \ 0 \ \cos \beta \ 0; 0 \ 0 \ 0 \ 1]$, 其中 $\cos \alpha = n/\sqrt{m^2 + n^2}$, $\sin \alpha = m/\sqrt{m^2 + n^2}$; $\cos \beta = n/\sqrt{l^2 + m^2 + n^2}$, $\sin \beta = -l/\sqrt{l^2 + m^2 + n^2}$ 。

(3) 再将最后位置的 l_2 绕 z 轴旋转 γ 度角, 变换矩阵 $A = R_z = [\cos \gamma \ \sin \gamma \ 0 \ 0; -\sin \gamma \ \cos \gamma \ 0 \ 0; 0 \ 0 \ 1 \ 0; 0 \ 0 \ 0 \ 1]$ 。

(4) 将旋转后所得曲面与 l_2 一块移动, 使 l_2 恢复

到最初的位置, 其中用到逆变换 R_y^{-1}, R_x^{-1} 以及 T^{-1} 。

(5) 若曲线 l_1 各坐标以列向量 X, Y, Z 表示, 则旋转 γ 后的曲面上的曲线坐标列向量 $X1, Y1, Z1$, 则有

$[X1, Y1, Z1] = [X, Y, Z]TR_xR_yA R_y^{-1}R_x^{-1}T^{-1}$, 这样曲线绕任意直线旋转所得的旋转曲面的动画一般程序 (3) 就可以写出了:

```
p=input('输入旋转轴上定点 p[x0,y0,z0]=');l=input('输入旋转轴方向矢量[l,m,n]=');
lim=input('输入旋转轴作图范围[t1,t2]=');
t=lim(1):0.1:lim(2);x1=t*l(1)+p(1)*ones(1,length(t));y1=t*l(2)+p(2)*ones(1,length(t));z1=t*l(3)+p(3)*ones(1,length(t));
plot3(x1,y1,z1,'k');
a=input('输入坐标轴取值范围[x1,x2,y1,y2,z1,z2]=');
axis([a(1),a(2),a(3),a(4),a(5),a(6)]);hold on;
T=[1,0,0,0;0,1,0,0;0,0,1,0;0,0,0,1];
Rx=[1,0,0,0;0,1,0,0;0,0,1,0;0,0,0,1];
Ry=[1,0,0,0;0,0,1,0;0,1,0,0;0,0,0,1];
Tr=T*Rx*Ry;alpha=pi/60;A=[cos(alpha),sin(alpha),0,0;-sin(alpha),cos(alpha),0,0;0,0,1,0;0,0,0,1];
n=input('请输入旋转曲线条数(1或2)=');
if n==1
    Lim=input('输入旋转曲线作图范围[s1,s2]=');s=Lim(1):0.1:Lim(2);
    u=input('输入旋转曲线参数 s 表达式[x(s);y(s);z(s)]=');u=u';I=ones(length(s),1);V=[u,I];
    for i=1:120;
        U=V*Tr*A*inv(Tr);
        surf([V(:,1),U(:,1)],[V(:,2),U(:,2)],[V(:,3),U(:,3)]);colormap(gray);
        shading interp;V=U;drawnow;pause(0.2);
    end
else
    L1=input('输入旋转曲线 1 作图范围[s1,s2]=');s=L1(1):0.1:L1(2);
    u1=input('输入旋转曲线 1 参数 s 表达式[x1(s);y1(s);z1(s)]=');u1=u1';I1=ones(length(s),1);V1=[u1,I1];
    L2=input('输入旋转曲线 2 作图范围[s3,s4]=');s=L2(1):0.1:L2(2);
    u2=input('输入旋转曲线 2 参数 s 表达式[x2(s);y2(s);z2(s)]=');u2=u2';I2=ones(length(s),1);V2=[u2,I2];
    for i=1:120;
```

```

U1 = V1 * Tr * A * inv (Tr); U2 = V2 * Tr * A * inv
(Tr);
surf([ V1(:,1),U1(:,1)], [ V1(:,2),U1(:,2)],
[ V1(:,3),U1(:,3)]]);colormap(gray);
surf([ V2(:,1),U2(:,1)], [ V2(:,2),U2(:,2)],
[ V2(:,3),U2(:,3)]]);colormap(gray);
shading interp; V1 = U1; V2 = U2; drawnow; pause(0.
2);
end
end

```

例如将椭圆 $l_5: y^2/16 + (z+3)^2/9 = 1, x=0$ 绕 z 轴旋转,执行程序(3)将得到椭球生成的动画;再次运行程序(3),绘制 $l_6: z = y^2, x=0$ 与直线 $l_7: z = y + 6, x=0$ 绕 z 轴旋转动画,注意其中输入旋转曲线条数为2(见图4)。

在文献[13]中,对于不同旋转曲面,需要重新编制自定义函数 generatrix(),执行主程序时,需调用该函数,这给使用带来不便,而且程序中旋转轴是固定

的,并且程序对于画双叶双曲面甚至分段函数动画,都无能为力。在此,推出了绕任轴旋转的几何变换坐标的一般公式,而且在使用程序时,只需要把描述旋转轴的定点 p 和方向矢量以及转曲线的参数方程,按照程序的提示输入即可,使用方便。特别地,运行程序(3)不但能够绘制两曲线或2-分段曲线同时旋转动画,而且还可以绘制多曲线依次旋转组合动画,只要连续多次运行程序即可。

2 结束语

依据 matlab 处理数据的特点,以旋转曲面为实例,通过画旋转曲线、曲面片、给曲面矩阵赋值 NaN 或 inf 以及利用齐次坐标和几何变换等形式给出动画程序的,这些方法对于其它曲面动画作图,也有普遍的指导意义。以上所有程序亦可嵌入到适当 gui 交互界面中使用,可以 m 文件或 m 函数形式保存,以便长期运用。

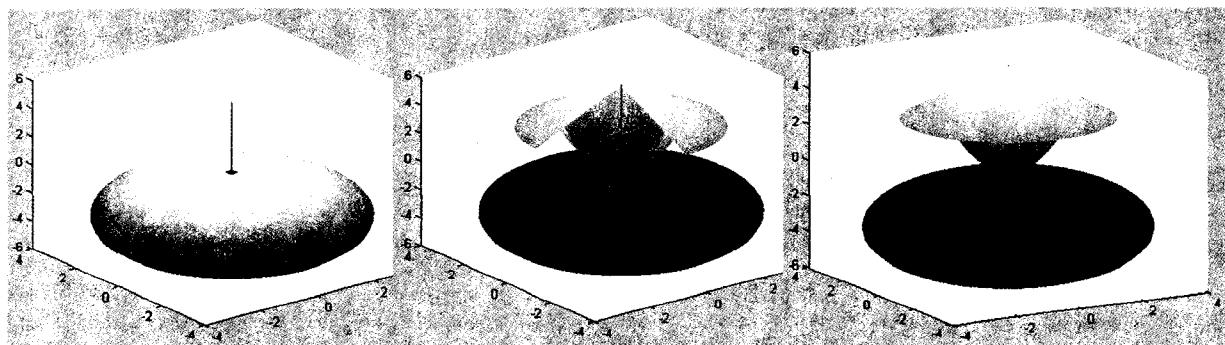


图4 l_5 旋转生成椭球和曲线 l_6 与 l_7 同时绕 z 轴旋转的动画截图

参考文献:

- [1] 胡 华. 用 Matlab 解决数学分析中的图形问题[J]. 西南民族大学学报, 2003, 29(6): 766-771.
- [2] 向修栋, 付云芝. 在 matlab 中实现极限动画效果[J]. 现代计算机, 2009(1): 151-153.
- [3] 张先波, 杨文颖, 宋来忠, 等. 给定主曲率函数的旋转曲面设计[J]. 计算机工程与设计, 2009, 30(4): 1030-1035.
- [4] Colombo C, Comanducci D, Bimbo A, et al. 3D Database Population from Single Views of Surfaces of Revolution[J]. Lecture Notes in Computer Science, 2005, 3617: 834-841.
- [5] Kenmotsu K. Surfaces of Revolution with periodic mean curvature[J]. Osaka J. Math., 2003, 40: 687-696.
- [6] Morin G, Warren J, Weimer H A. Subdivision Scheme for Surfaces of Revolution[J]. Computer Aided Geometric Design, 2001, 18(5): 483-502.
- [7] 闫宇哈, 常 鑫. 在 C# 中用 GDI 实现图形动态显示[J]. 计算机技术与发展, 2006, 16(12): 117-119.
- [8] 戴春来, 丁运亮, 陆 信. 多截面形状要求的广义旋转曲面[J]. 机械科学与技术, 2002, 21(3): 537-539.
- [9] 段俊生, 安建业, 徐 立. MATLAB 曲面绘制中的挖补方法[J]. 大学数学, 2006, 22(4): 36-39.
- [10] Buss S R. 3-D Computer Graphics A Mathematical Introduction with OpenGL[M]. New York: Cambridge University Press, 2003: 17-58.
- [11] 何援军. 图形变换的几何化表示—论图形变换和投影的若干问题之一[J]. 计算机辅助设计与图形学学报, 2005, 17(4): 723-728.
- [12] Zhang Jinhuan, Ma Chunsheng, Bai Yuanli, et al. Airbag Mapped Mesh Auto-Flattening Method[J]. Tsinghua Science and Technology, 2005, 10(3): 387-390.
- [13] 郑凤川. 用 Matlab 动画演示几何图形的几种方法[J]. 内科, 2008, 10: 164-165.