

# 实验教程：函数的可视化与 Matlab 作图

## 1 实验与观察：函数的可视化

### 1.1 Matlab 二维绘图命令

人们很难从一大堆原始数据中感受到它们的含义，数据图形恰能使人们直接感受到数据的许多内在本质。同样的，对于数学课程中所涉及的一元或多元函数，了解它们的图形将给予我们更多的直观印象，从而能更好地理解课程的内容。数据可视化是人们研究科学、认识世界不可缺少的手段。Matlab 提供了强大的绘图工具，可以给出数据和函数的二维、三维、乃至四维的图形表现；并通过对图形的线型、立面、色彩、渲染、光线和视角的属性处理，可以将数据的特征表现得淋漓尽致。

#### (1) 周期函数与线性 $p$ -周期函数

我们首先从周期函数入手，利用 Matlab 的图形表现功能，探讨一些有趣的规律。

周期函数是我们熟悉的函数，它们描述一再重复的现象，比如太阳的升起、弹簧的振动以及声波的运动等等。

称函数  $f$  是  $p$  周期的，只要存在一个正实数使得对每个实数  $x$  均有  $f(x+p)=f(x)$  成立；使  $f$  是  $p$  周期的最小正数  $p$  称为  $f$  的周期。

例如正弦函数和余弦函数是以  $2\pi$  为周期的，我们可以把三角函数当作砖块，用加、乘和复合的手段构造出新的周期函数。

- ◆ 观察：(1)  $f(x) = \cos(x + \sin(x))$  是周期函数吗，它的图象是什么？  
(2)  $g(x) = 0.1x + \cos(x + \sin(x))$  是周期的吗？它的图象又是什么？

运行观察程序 **program\_1.m**

```
【 clf, x=linspace(0,8*pi,100);
F=inline('sin(x+cos(x+sin(x)))');
y1=sin(x+cos(x+sin(x)));
y2=0.2*x+sin(x+cos(x+sin(x)));
plot(x,y1,'k:',x,y2,'k-') ,
set(gca,'fontsize',16);
legend('sin(x+cos(x+sin(x)))','0.2x+sin(x+
cos(x+sin(x)))',2) 】
```

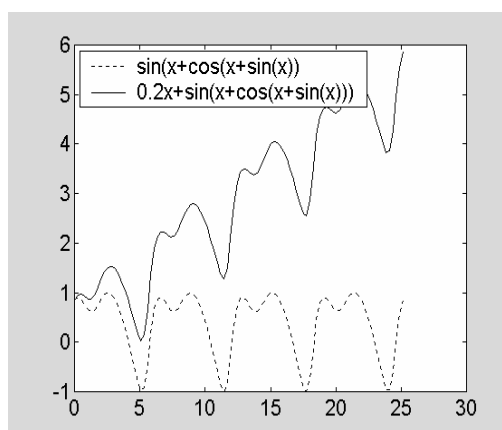


图1 周期和线性周期函数

得到图1。现在这两个函数直观地显示在你的面前，你看到它们有什么共同的特点？又有什么区别吗？

我们将结合上面的例子，逐步介绍Matlab的绘图、文字的标注及其绘图属性的设置等内容。

## (2) plot 指令：绘制直角坐标的二维曲线

绘制二维曲线的最常用的指令是plot，基本用法为：

★plot(x,y)：以x为横坐标、y为纵坐标绘制函数或数据图形。x,y是同维数的向量；

★plot(y)：绘制一个向量的图形，相当于x=[1,2,⋯,length(y)]时情形。

★在同一图形窗口中绘制多个曲线：有下面的四种方法做到这一点：

(1) plot(x,[y1;y2;⋯]),其中 x 是横坐标向量, [y1;y2;...]是由几个函数的纵坐标拼成的矩阵,这时 Matlab 将用不同颜色的曲线表示不同的函数,读者可运行 plot(x,[y1;y2])观察其绘图效果；

(2) 使用 hold on 命令：在画完一张图后,用 hold on 命令保持住,然后再画其它图形,结束画图后,可用 hold off 命令取消连续画图命令；

(3) 使用 plot(x,y1,x,y2,⋯)命令,如程序 program\_1 所做的那样；

(4) 用 plotyy 命令,plotyy 命令设置了两个坐标系,用于绘制不同尺度的函数。例如,

【plotyy(x,y1,x,y2),grid on】

得到图 2,注意左边纵轴尺度和右边的尺度是不同的。

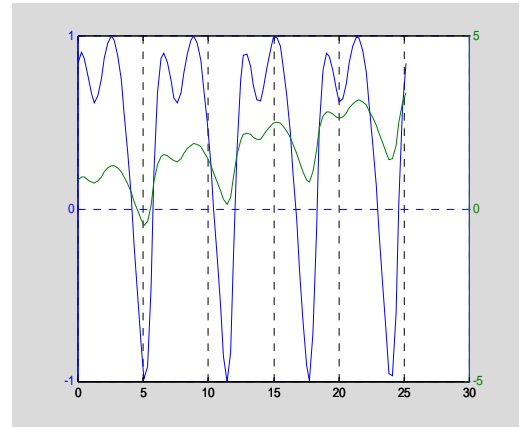


图2 用plotyy命令绘图的效果

## (3) 图形的属性设置和屏幕控制

常常需要改变曲线的线型、点的大小和形状、坐标系的范围等等,以便更清楚地进行观察。Matlab绘图提供了这样的功能,使用者可以方便的改变各种图形对象的属性。

★线型、点形和颜色的控制。

可用下面的方法设置图形对象的属性：

plot(x,y,'颜色+线型+点形');

plot(x,y,'颜色+线型+点形',x,y,'颜色+线型+点形',⋯)

另一种改变图形对象属性的方式是采用句柄图形的方式,并用 set 命令改变属性的值,可套用如下命令格式:

h=plot(x,y), set(h,'属性',属性值,'属性',属性值,⋯)

也可用 plot(x,y,'属性','属性值')设置图形对象的属性。

表 1 给出了常用的属性值选项,更多的属性值可在 Matlab 命令窗口菜单中点击 help→help desk→Matlab Function .by index→axes 找到它们的详细列表(axes 是用来产生图形对象的命令)。

表 1 (a) 颜色的标识符

标识符	y	r	c	m	g	b	w	k
颜色	黄	红	青	紫	绿	蓝	白	黑

表 1 (b) 点型和线型的标识符

标识符	-	--	:	-.	.	o	x	+	*	square
线型和点型	实线	破折线	虚线	点划线	点	圆圈	×号	十号	*号	方形单词

这样，就可以对线型（LineStyle）、线宽（LineWidth）、点的大小（MarkerSize）、颜色（color）等等属性进行设置，取得满意的绘图效果。

◆ 观察：改变绘图的线型和颜色。用 `grid on` 指令为图形窗口加上 网格线，并改变网格的线型和字体的大小。

```
【h=plot([0:0.1:2*pi],sin([0:0.1:2*pi])); grid on
    set(h,'LineWidth',5,'color','red'); set(gca,'GridLineStyle','-','fontSize',16) 】
```

运行结果见图 3。

这里 `grid on` 是给图形加上网格，默认网格线是虚线，在上面程序中用 `set` 命令设置 '`GridLineStyle`' = '-', 所以图形中网格线为实线；`gca` 是轴对象句柄，程序改变了它的默认设置，如字体大小和网格线的属性等等。

注：另一种画函数图的命令为：

```
fplot('函数运算式',[xmin xmax])
```

该命令可以画出函数在区间 `[xmin xmax]` 上的图形。



图 3 用 `set` 改变图形的属性

★`axis` 指令。

用 `axis` 指令可以设定坐标轴的范围：

`axis([xmin xmax ymin ymax])`：设定二维图形的 `x` 和 `y` 坐标的范围；

`axis([xmin xmax ymin ymax zmin ymax])`：设定三维图形的坐标范围；

其中 `xmin < x < xmax`，`ymin < y < ymax`，`zmin < z < zmax`。

★改变坐标刻度。

`gca` 为当前轴对象句柄，用 `set(gca,'属性',属性值,...)` 可改变字体大小、坐标刻度等轴对象的内容。例如指令 `set(gca,'ytick',[-1 -0.5 0 0.5 1])` 表示将 `y` 坐标按向量 `[-1 -0.5 0 0.5 1]` 给出的刻度分成 4 格；而指令 `set(gca,'yticklabel','a|b|c|d|e')` 则改变 `y` 坐标刻度的说明：

◆ 设置 `y` 坐标的刻度并加以说明，并改变字体的大小。

```
【h=plot([0:0.1:2*pi],sin([0:0.1:2*pi]));grid
on,
```

```
    set(gca,'ytick',[-1 -0.5 0 0.5 1]),
```

```
    set(gca,'yticklabel','a|b|c|d|e'),
```

```
    set(gca,'fontSize',20)      】
```

运行结果见图 4。你可以把程序中的指令 `set(gca,'yticklabel','a|b|c|d|e')` 删去看看效果，了解改变坐标刻度说明的意思；其它坐标轴刻度也可以类似处理。

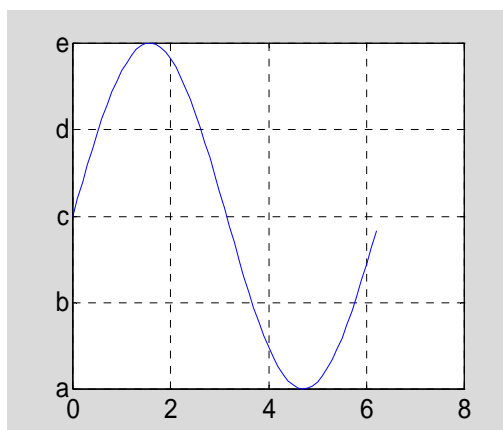


图 4 改变坐标刻度

可用 `get` 指令获得当前坐标轴属性，在命令窗口键入 `get(gca)`，可以查看轴对象 `gca` 的各项内容。

#### (4) 文字标注指令

在很多情形下，我们要对图形进行某些必要的文字说明，这可以使用如下的文字标注指令完成：

★`title('图形标题')`；`xlabel('x轴名称')`；`ylabel('y轴名称')`；`zlabel('z轴名称')`；`text('说明文字')`：创建说明文字；`gtext('说明文字')`：用鼠标在特定位置输入文字。

下面是进行文字标注的常用符号：

`\pi` ( $\pi$ )、`\alpha` ( $\alpha$ )、`\beta` ( $\beta$ )、`\leftarrow` (左箭头)、  
`\rightarrow` (右箭头)、`\bullet` (点号)

可以在前面的程序 `program_1.m` 中加入某些标注指令进行必要的文字标注，对图形文字的标注可以套用下面程序的格式。

◆观察：假设我们已经运行了程序 `program_1.m`，这时在工作变量空间中就存储了所需要的变量，并得到图形窗口如图 1 所示。重新画图（可在命令窗口直接操作，也可以在 M 文件编辑器中直接对程序进行修改），为图 1 加入适当文字标注，如图形名称、坐标说明和曲线说明等等。

```
plot(x,y1,'b',x,y2,'k-');
set(gca,'fontsize',15,'fontname','times New Roman'), %设置轴对象的字体为 times
% New Roman, 字体的大小为 15
title('\it{Peroid and linear peroid function}'); %加标题，注意文字用单引号''加上
%斜杠'\后可输入不同的设置，例如it{...}表示花括号里的文字为斜体；如果有多项设置，
%则可用\...\...\连续输入。
xlabel('x from 0 to 8*pi \it{t}'); ylabel('\it{y}'); %说明坐标轴
text(x(49),y1(50)-0.4,'fontsize{15}\bullet\leftarrow The period function {\it{f(x)}}');
%在坐标(x(49),y1(50)-0.4) 处作文字说明， 各项设置用\"隔开。
%\fontsize{15}\bullet\leftarrow的意义依次是：\字体大小=15\画圆点\左箭头
text(x(14),y2(50)+1,'fontsize{15} The
linear period function {\it{g(x)}}\rightarrow
\bullet') %与上一语句类似，用右箭头  】
```

结果见图 5。

在对图形进行文字标注时，可套用上面程序的命令格式，运行后进行观察，逐步加以熟悉。

命令 `gtext('...')` 和 `Text` 的作用是一样的，前者是鼠标选择文字标注的位置，比后者更为方便；事实上，画好图后，可在图形窗口使用 Matlab 图形窗

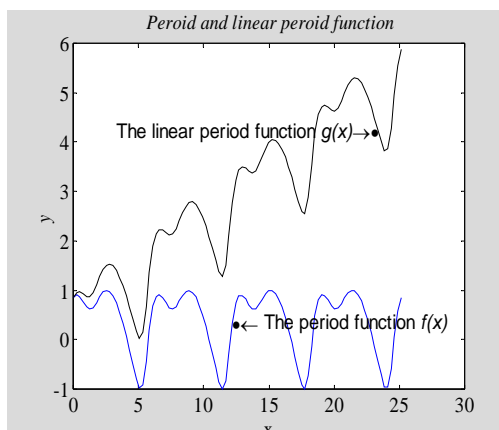


图 5 文字标注

口的编辑菜单进行包括文字标注在内的各项编辑，读者可以自行摸索掌握。

★**legend** 指令。

**legend** 可用于说明不同函数所使用的线型，如 **program\_1** 所做的那样。格式为：

**legend** ('string1','string2', ...)

其中 **string1** 为第 1 条曲线的说明文字、**string2** 为第 2 条曲线的说明文字，等等。

★**num2str** 指令。

**num2str** (**b**) 把数字 **b** 转化为字符串，如果希望在图形上显示函数值，则可用此命令并结合 **text** 做到这一点，这时应按如下格式使用 **text** 命令：

**text**(...,['string1',**num2str**(**b**),'string2'],...)

◆观察指令 **legend** 和 **num2str** 的用法：在同一张图上画出  $y = \sin(t)$  和  $y = 3e^{-0.5t}$ ，这里  $t \in [0, 3\pi]$ ，并进行适当的标注。

**program\_2.m**

```

【  clf, t=0:0.1:3*pi; alpha=0:0.1:3*pi;
    plot(t,sin(t),'r-'); hold on; plot(alpha, 3*exp(-0.5*alpha), 'k:');
    set(gca, 'fontsize', 15, 'fontname', 'times New Roman'),
    xlabel('\it{t(deg)}'); ylabel('\it{magnitude}');
    title('\it{sine wave and {\it{Ae}}^{-\alpha t} wave}'); %注意\alpha的意义
    text(6, sin(6), '\fontsize{15}The Value \it{sin(t)} at {\itt}=6\rightarrow\bullet',
    'HorizontalAlignment', 'right'),
    %上面的语句是一整行，如果要写成两行，必须使用续行号 ...，例如要在“bullet'，”
    %后换行，需写“bullet', ...”后才能换行。
    % 'HorizontalAlignment', 'right' 表示箭头所指的曲线对象在文字的右边。
    text(2, 3*exp(-0.5*2), '\fontsize{15}\bullet\leftarrow The Value of \it{3e}^{-0.5}
    \it{t}}=', num2str(3*exp(-0.5*2)), ' at \it{t} = 2 ');
    %num2str的用法：['string1', num2str, 'string2']，注意方括号的使用。
    %legend('\itsin(t)', '\it{Ae}^{-\alpha t}') %请结合图形观察此命令的使用
】

```

运行结果如图 6 所示。

到此我们对图形的标注和其属性的修改有了大致的解，对于三维作图和其它的作图，这些命令也是适用的。

★**clf**、**shg**、**clear**、续行号...

**clf** 指令是清除图形窗口已有的内容，否则常会遇到绘制的新图与已有旧图迭在一起的情况。

**shg**：显示图形窗口。

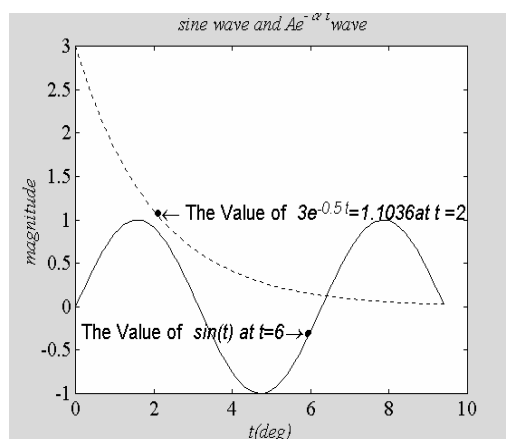


图 6 **legend** 命令和 **num2str** 命令的用法

**clear:** 清除工作空间的已有变量。

有时, 你可能处理了一个 10 维向量  $x$ , 希望用 `plot(x)` 指令它的图形, 但图形窗口却显示一个 100 维向量。出错的原因可能是在前面你已经运行了一个程序, 其中含有一个 100 维向量  $x(1:100)$ , Matlab 工作空间里存储着你已经运行的所有变量, 用 `clear` 指令可以全部清除它们; 用 `clear x` 则只清除变量  $x$ , 再运行程序就可显示 10 维向量。

当一句过长的语句在一行写不下的时候, 必须使用续行号...才能另起一行写下去, 本书所有过长的语句都省略了这一续行号, 读者在编辑运行这些程序时要注意这一点。

## (5) 图形窗口的创建和分割

★ `subplot(m,n,k)` 命令。

如果希望在一个图形区域中显示多个图形窗口, 可以使用 `subplot` 指令, 其格式为: `subplot(m,n,k)`,  $m$  为上下分割数,  $n$  为左右分割数,  $k$  为分割后第  $k$  个子图编号。

◆观察: 将一个图形分为 9 个子图, 在第  $k$  个子图画函数  $\sin(kx)$  的图象, 进行比较。

```
【 clf, b=2*pi;x=linspace(0,b,50);
   for k = 1:9
       y=sin(k*x);
       subplot(3,3,k),plot(x,y),axis([0,2*pi,-1,1])
   end   】
```

运行结果如图 7 所示。在第 8~9 个子图中函数有些失真, 因为相对于这些频率较高的函数, 采样点较少。有专门的采样定理讨论这一问题, 由于在实际中只能得到连续信号的离散数据, 所以如何确定采样数据的间隔才能不失真地恢复原信号是非常重要的问题。有兴趣的读者可查阅有关信号处理的书籍。

★ `figure(n)` 指令

如果想开新的图形窗口, 可使用命令 `figure(n)`, 这里  $n$  为自然数, 表示第  $n$  个图形窗口, 例如 `figure(1)`、`figure(2)`、`figure(3)` 等, 读者可试试这些命令。

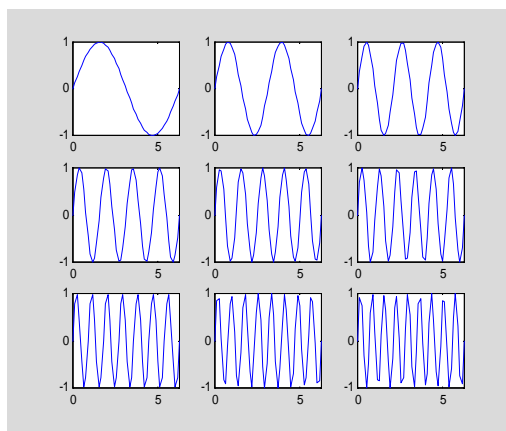


图7 subplot指令的用法

## 1.2 多元函数的可视化与空间解析几何（三维图形）

本节通过高等数学的几个例子观察 Matlab 的三维绘图功能和技巧。

### (1) 绘制二元函数

◆观察: 绘制  $z = f(x, y) = (1-x)^{-\frac{1}{2}} \ln(x-y)$  的图象, 作定义域的裁剪。

### ★meshgrid 指令

meshgrid指令用来形成二维网格点，生成这些网格点后，可以计算在这些网格点上的函数值，为绘制图形做准备。

◆（1）观察meshgrid指令的效果。

```
【 a=-0.98;b=0.98;c=-1;d=1;n=10;
    x=linspace(a,b,n); y=linspace(c,d,n);
    [X,Y]=meshgrid(x,y);
    plot(X,Y,'+') 】
```

运行结果见图 8。由此可见，meshgrid(x,y)根据向量 x,y 生成了二维网格点[X,Y]，用 whos 指令查看变量维数：

Name	Size	Bytes	Class
x	1x10	80	double array
y	1x10	80	double array
X	10x10	800	double array
Y	10x10	800	double array

所以网格点是  $(X(i,j), Y(i,j))$ ， $i, j=1, 2, \dots, 10$ ，共有100个网格点。

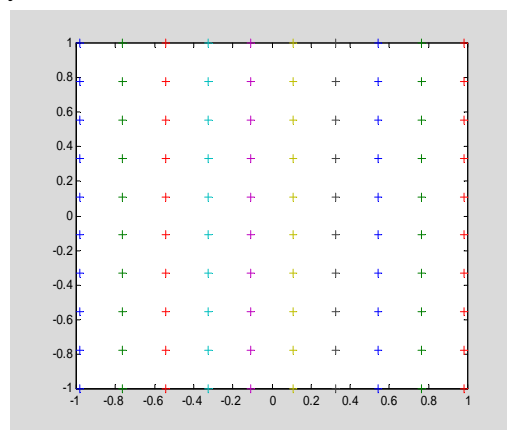


图 8 用 meshgrid 产生二维网格

### ★计算函数值和定义域的裁剪。

下一步是计算函数在这些网格点上的值，这只需直接计算函数值  $z = f(X, Y)$ ；但由于函数  $z = f(x, y) = (1-x)^{-\frac{1}{2}} \ln(x-y)$  的定义域为  $x \leq 1$  &  $y > x$ ，所以  $f(X, Y)$  的有些函数值在实数范围内无意义，需将这些无意义的值裁剪掉。

有两种可行的处理方法：一种是计算函数值时将非定义域内的点的函数值赋为非数 NaN，如下面程序中所做的一样；另一种是先计算函数值，由于 Matlab 一般是处理复数，因此当函数值  $f(x_0, y_0)$  的虚部不为零时，则  $(x_0, y_0)$  在定义域之外，这时令  $f(x_0, y_0) = \text{NaN}$  就将无定义的点裁剪掉了。Matlab 只绘制  $f(X, Y)$  的数值部分，而将其非数 (NaN) 部分裁掉，这样我们就可以很容易地处理函数的图象了。

### ★三维绘图指令 mesh、meshc、surf。

在下面的观察程序中 program\_4.m，采用四种绘制三维图指令，显示它们不同绘图效果。

mesh(X,Y,z)：在三维空间中绘出由(X,Y,z)表示的曲面，见图 9 (a)；

meshz(X,Y,z)：除了具有 mesh 的功能外，还画出上下高度线，如图 9 (c) 所示；

meshc(X,Y,z)：除了具有 mesh 的功能外，还在曲面的下方画出函数  $z=f(x,y)$  的等值线图，如图 9 (d) 所示；

surf(X,Y,z)：也是三维绘图指令，与 mesh 的区别在于：mesh 绘出彩色的线，surf 绘出彩色的面，如图 9 (b) 所示。



◆ (2) 做函数的定义域裁剪，观察上述三维绘图指令的效果。

运行观察程序 `program_4.m`，结果如图 9 所示，在图 9 各图中，我们绘出了定义域的边界  $y=x$ 。

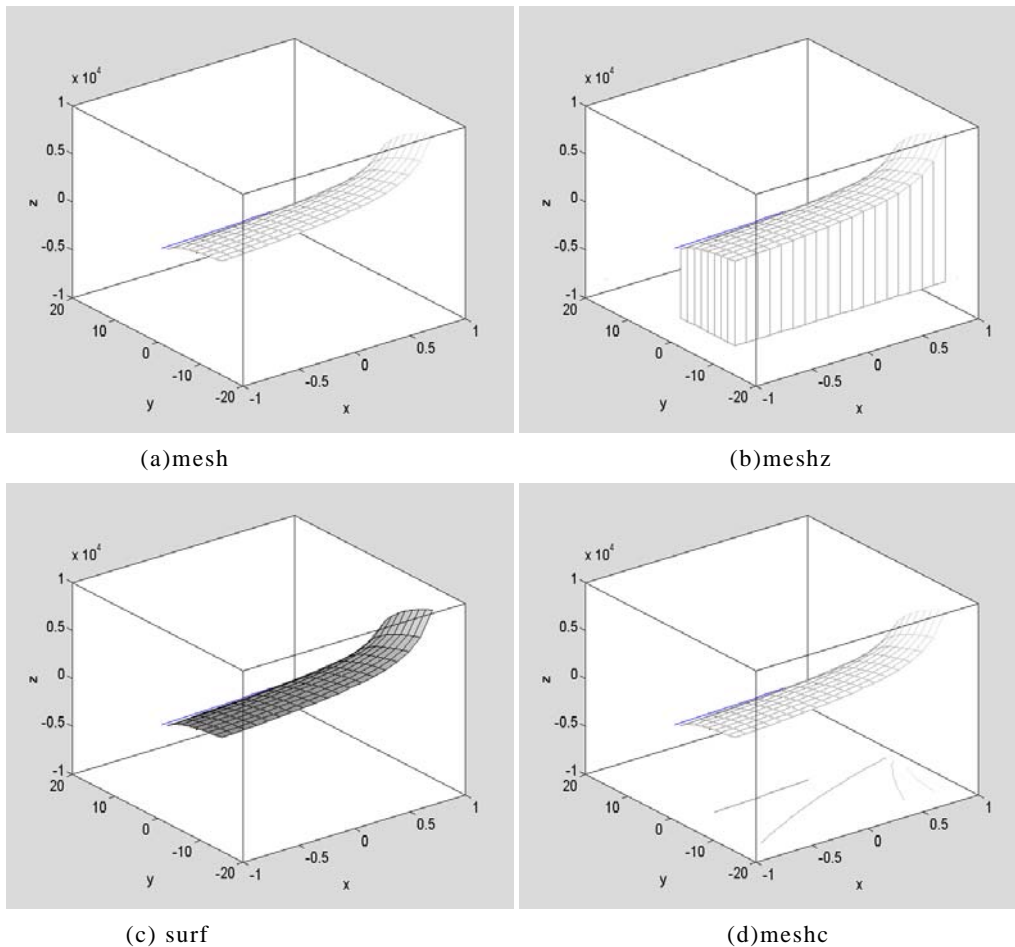


图9 几种三维绘图指令的不同视觉效果

程序 `program_3.m`

```

【 clear,clf,
a=-1;b=1;c=-15;d=15;n=20;eps1=0.01;
x=linspace(a,b,n);y=linspace(c,d,n);
[X,Y]=meshgrid(x,y);
for i=1:n
    %计算函数值z，并作定义域裁剪
    for j=1:n
        if (1-X(i,j))<eps1|X(i,j)-Y(i,j)<eps1 %if语句这样用
            z(i,j)=NaN; %作定义域裁剪，定义域以外的函数值为NaN
        else
            z(i,j)=1000*sqrt(1-X(i,j))^-1.*log(X(i,j)-Y(i,j));
        end
    end
end
end
end

```



```
zz=-20*ones(1,n);plot3(x,x,zz),grid off,hold on    %画定义域的边界线
mesh(X,Y,z)    %绘图，读者可用meshz, surf, meshc在此替换之
xlabel('x'),ylabel('y'),zlabel('z'), box on    %把三维图形封闭在箱体里】
```

★绘制等值线的 contour 指令。

表现二维函数的图形的另一种方式是绘制等值线图，Matlab 有具有几个不同效果的等值线绘制指令，主要是：

**contour(X,Y,z,n):** 将空间图形水平切割所得的等高线投影到在x-y平面上，切割次数为n，参数n可缺省；

**contourf(X,Y,z,n):** 用法与contour相同，但在等值线间用不同的颜色填满，从而具有更好的视觉效果；

**contour3(X,Y,z,n):** 在三维空间画出等值线图，即不把等值线向x-y平面投影。

◆运行了program\_3.m 以后，有关向量存储在工作空间中，在此基础上，观察上述等值线绘制指令的运行效果。

【 clf,contour(X,Y,z,40),colorbar 】 结果见图10（a）。

★colorbar指令：在图中，Matlab以不同色彩绘制了40条等值线，colorbar指令将颜色与函数值对应起来显示在图中，使人明了函数值的变化情形。由于印刷是非彩色的，在书中只能显示灰度不同。

【 contourf(X,Y,z,40),colorbar 】 结果见图10（b）。

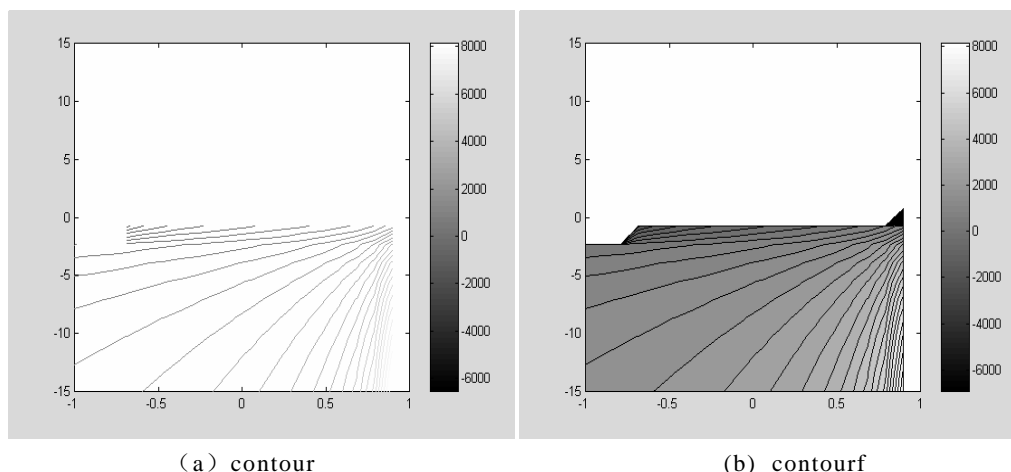
【 contour3(X,Y,z,40),colormap([0,0,0]) 】 结果见图10（c）。

注：在上面的语句中，考虑到印刷是非彩色的，为使图形能比较清楚地表现出来，放弃了默认色彩，使用命令colormap([0,0,0])将当前图形的颜色设置为黑色。

还可以通过 clabel 指令为等值线标上函数值，用法可套用下面程序的格式：

【 [cs,h]=contour(X,Y,z,15); clabel(cs,h,'labelspacing',244) 】

结果如图10(d)所示，程序中labelspace是数值标记之间相隔的宽度，默认值为144。这里取了244，其详细用法可参见Matlab的帮助。



(a) contour

(b) contourf

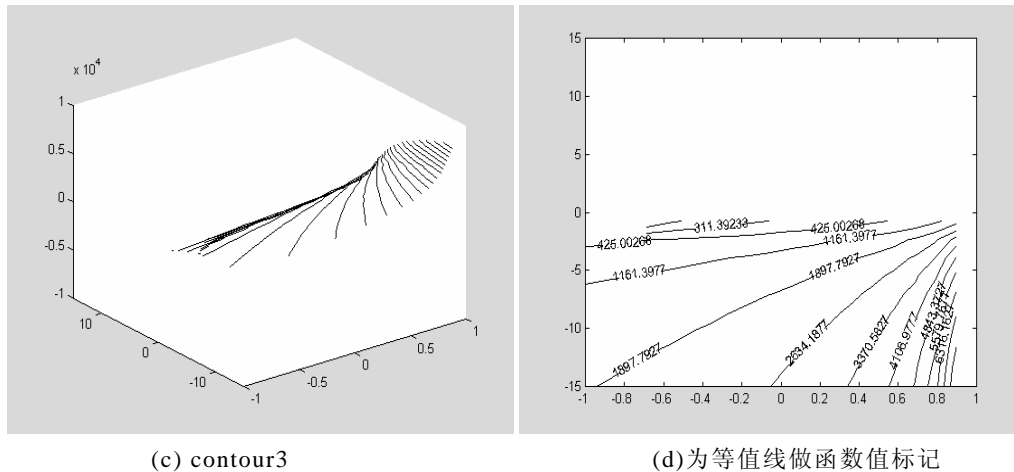


图 10 绘制等值线

## (2) 三元函数可视化: slice 指令

为了实现三元函数  $w = f(x, y, z)$  的可视化表现, Matlab 提供了一个绘制三维物体切片图指令 slice 及与之配合的三维坐标生成指令。指令 slice 的调用格式为:

`slice(X,Y,Z,w,xi,yi,zi)`

其中  $X,Y,Z$  为三维坐标网格点,  $w=f(X,Y,Z)$  为这些网格节点上的函数值矩阵。 $xi,yi,zi$  分别为垂直于  $x,y,z$  轴的切片位置向量, 它们的维数可以不同。

◆ 观察: 绘制三元函数  $w = x^2 + y^2 + z^2$  的可视化图形。

```

【   clf,x=linspace(-2,2,40); y=x; z=x;
      [X,Y,Z]=meshgrid(x,y,z); w=X.^2+Y.^2+Z.^2;
      slice(X,Y,Z,w,[-1,0,1],[-1,0,1],[-1,0,1]),colorbar   】

```

运行结果如图 11 所示。在图中, 函数值的变化通过颜色的差异来表现, 所以应配合 `colorbar` 指令来使用。

也可类似绘制其它函数的切片图, 如

$$w = \sin(x^2 + y^2 + z^2)$$

## (3) 空间曲线及其运动方向的表现: plot3 和 quiver 指令

在空间解析几何中, 我们需要考虑空间曲线及其切矢量曲线的可视化表现。一条空间曲线可以用矢量函数表示为  $\Gamma: r(t) = \{x(t), y(t), z(t)\}$ ,  $t \in [\alpha, \beta]$ , 而它的速度矢量可由对  $r(t)$  求

导而得, 表现为曲线的切矢量, 即速度矢量为  $v(t) = \frac{dr}{dt} = \left( \frac{dx(t)}{dt}, \frac{dy(t)}{dt}, \frac{dz(t)}{dt} \right)$ 。

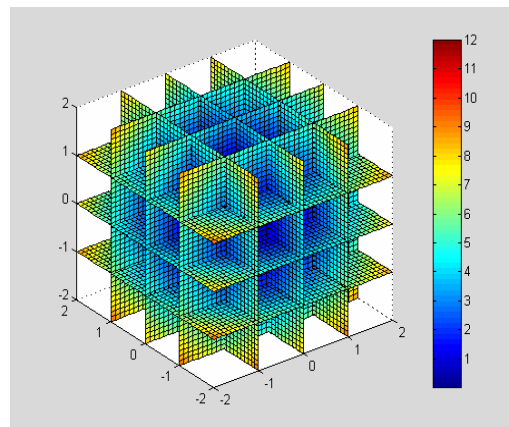


图 11 slice 指令, 用颜色表示函数值

◆观察：一架飞机沿某空间曲线飞行，飞行速度为  $\frac{dx}{dt}=2t, \frac{dy}{dt}=2t^2, \frac{dz}{dt}=6t^3-t^2$ 。设飞机的初始位置在坐标原点，绘制飞机飞行的轨迹，并标明速度矢量。

很显然，曲线方程为  $r(t) = \int_0^t (2t, 2t^2, 6t^3 - t^2) dt = (t^2, (2/3)t^3, 3t^4 - (1/3)t^3)$ ，用 `plot3` 和 `quiver3` 指令可很画出飞机飞行的轨迹并且标明飞行方向。

★ `plot3(x,y,z)`：在三维空间中绘制曲线，用法和 `plot` 类似。

★ `quiver(X,Y,u,v)`：用于绘制二维矢量，即在坐标矩阵点 `[X,Y]` 处绘制矢量 `[u,v]`，其中 `u` 为矢量的 `x` 坐标，`v` 为矢量的 `y` 坐标，它们的维数要大于等于 2。

★ `quiver3(X,Y,Z,u,v,w)`：用于绘制三维矢量，用法与 `quiver` 类似。

★ `gradient` 指令，由于 `quiver` 常被用来表现函数的梯度场，所以常和计算数值梯度的指令 `gradient` 一起配合使用，`gradient(F)` 等效为  $\nabla F(x)$ ，例如若 `F` 为三元函数，则指令 `[Fx,Fy,Fz]=gradient(F)` 将输出 `F` 的数值梯度，其中 `Fx`、`Fy`、`Fz` 为  $\nabla F$  的三个分量。对一元、二元函数的情形是类似的。可以通过参数设置来调整差分的步长，可进一步查阅 `Matlab` 的帮助。

下面的观察程序说明了上述指令的用法，可以套用。

```

【 clf, t=0:0.1:1.5;
Vx=2*t;Vy=2*t.^2;Vz=6*t.^3-t.^2;
x=t.^2;y=(2/3)*t.^3;z=(6/4)*t.^4-(1/3)*t.^3; %由速度得到曲线
plot3(x,y,z,'r-'),hold on                    %画飞行轨迹
%算数值梯度，也就是重新计算数值速度矢量，这只是为了编程的方便，不是必须的
Vx=gradient(x);Vy=gradient(y);Vz=gradient(z);
quiver3(x,y,z,Vx,Vy,Vz),grid on               %画速度矢量图
xlabel('x'),ylabel('y'),zlabel('z')          】

```

运行结果见图 12。

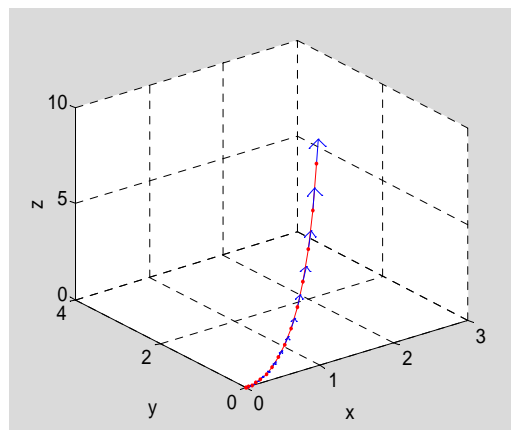


图 12 飞机的飞行轨迹与方向

## 2 应用、思考和练习

### 2.1 线性 $p$ 周期函数

◆思考 1. 在图 1 中，显然  $f(x) = \cos(x + \sin(x))$  是周期函数，它的周期  $p = 2\pi$ ，而另一个函数  $g(x) = 0.1x + \cos(x + \sin(x))$  好像是一个倾斜的周期函数，我们的问题是：如何刻画这种倾斜的周期函数？

如果一个函数的周期是  $p$ ，那么  $f(x + p) = f(x)$ ，把它换一种写法：

$$f(x+p) - f(x) = 0$$

即  $f(x+p) - f(x)$  的差值永远是 0。基于这种理解，可以把周期函数的定义进行扩充：假定这个差值不是 0 而是非零的常数，我们就可以把这个函数看成是线性周期的。也就是说，可以提出下面的定义。

**定义：** 假设  $p > 0$ ，当且仅当存在一个实常数  $M$ ，使得对所有的  $x$  都有

$$f(x+p) = f(x) + M$$

则称  $f(x)$  是线性  $p$  周期的，常数  $M$  被称为  $f$  的  $p$  变换常数。

◆思考2. 由这一定义， $p$  周期函数是一种特殊的线性  $p$  周期函数，为  $M=0$  的情形。容易验证  $g(x) = 0.1x + \cos(x + \sin(x))$  是线性  $2\pi$  周期函数。在  $g(x)$  中，第一项  $0.1x$  是线性函数，第二项  $\cos(x + \sin(x))$  是以  $p = 2\pi$  的周期函数，这是不是线性  $p$  周期函数的一般特征呢？也就是说，线性  $p$  周期函数是不是一个线性函数同一个  $p$  周期函数之和呢？

为考察这一问题，可以选择一些特殊的函数进行观察，归纳出某些结论，然后进行必要的理论证明。让我们给定一个  $p$  周期函数  $h(x)$ ，考察下面的问题。

◆问题1: 若  $h(x)$  是一个  $p$  周期函数，上限积分函数  $S(x) = \int_a^x h(t)dt$  的图象有什么特点？

我们可取一个周期函数来进行观察，例如取  $h(x) = \sin(x + \cos(x))$ ，并在区间  $[a, b] = [-8, 12]$  上计算  $S(x) = \int_a^x h(t)dt$  并作出上限积分函数的图象。积分可用 Matlab 指令

`quad('f', 积分下限, 积分上限)`

来计算，其中  $f$  是被积函数，需编制一个函数子程序，其基本格式为：

**program\_4\_f.m**

```
【 function f=program_4_f(x)
    f=sin(x+cos(x));
    】
```

在上面的程序中，`function` 是必须的，在 M 文件编辑器中编好这一文件存盘后将生成名为 `program_4_f.m` 的 M-文件。然后调用 `quad('program_4_f', -8, xx)` 就可以计算数值积分

$\int_{-8}^{xx} h(t)dt$  了，不断改变  $xx$  的值就得到上限函数  $S(xx)$ ，但由于数值的奇异性可能产生计算错误，请注意下面程序的处理。

**program\_4.m**

```
【 clear, clf
    a=-8;b=12;n=300;xx=linspace(a,b,n);
    h=program_4_f(xx);
    S(1)=0;
    for i=2:n
        S(i)=S(i-1)+quad('program_4_f',xx(i-1),xx(i));
    end
    】
```

```
subplot(1,2,1),plot(xx,S,'k-'),axis([a,b,-1.5,9])
```

```
subplot(1,2,2),plot(xx,[h;zeros(1,length(xx))],'k-'),axis([a,b,-1.5,1.5])  】
```

运行结果如图 8 所示。和图 1 一样，该曲线向上倾斜，似乎沿着一条倾斜的直线方向进行周期波动。

◆练习： 请你为图 13 加上标题、坐标说明，曲线说明和适当的文字标注。

◆问题2: 你能把这条直线找出来吗？ 即，能否能找到一条直线，使函数  $S(x)$  与该直线的差恰为某个周期函数？ 如果找到了这一条直线，则  $(S(x) - \text{该直线})$  的图形应该是水平的周期函数，就是说我们把图 13 中的倾斜函数变成水平的了，试用图形把这种情形反映出来。

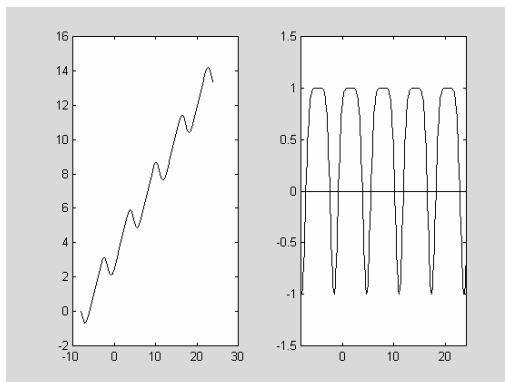


图 13 上限函数的线性周期性观察

提示：找一直线的思路有：

(1) 试探法：选取线性函数  $y = k(x - a)$ ，调整  $k$  使差函数的图象变为水平；请你选择一个周期函数（例如  $h(x) = \sin(x + \cos(x))$ ），用 Matlab 计算它的上限积分函数，并编制程序，用试探法找到这条直线。

(2) 理论分析法：取  $k = \frac{1}{p} \int_0^p h(t) dt$  试试，观察其结果， $k$  可以作为这条直线的斜率吗？请你加以分析，给出一个合理的说明。

观察周期函数的上限积分函数的曲线形状，是因为两者有密切的关系，事实上有下面的结论。

**定理：** 一个连续函数  $y = h(x)$  是  $p$  周期的充分必要条件是 its 上限积分函数  $S(x) = \int_a^x h(t) dt$  是线性  $p$  周期的。

定理 1 的必要性是容易验证的。

下面讨论它的充分性，因为  $S(x+p) - S(x) = \int_x^{x+p} h(t) dt$ ，我们可以用 Matlab 观察函数  $\int_x^{x+p} h(t) dt$  的图象（请读者绘制）。如果它是常数，根据线性  $p$  周期函数的定义，定理的充分性条件就应该成立，理论上证明也并不困难。

进一步，对于连续可微的线性  $p$  周期函数  $S(x)$ ，如果能够确定问题 2 中的直线斜率  $k$ ，则有  $S(x) = k(x - a) + \int_a^x (S'(t) - k) dt$ ，其中易证  $(S'(t) - k)$  为  $p$  周期函数。由此可见，把线性  $p$  周期函数校平的作用是找到  $k$  使得函数  $\int_a^x (S'(t) - k) dt$  变成  $p$  周期的。这样，我们就可以把线性  $p$  周期函数分解为线性函数和一个  $p$  周期函数之和了。

问题1和问题2是有趣的，它告诉我们如何扩展自己的思维。将周期函数的定义换一个写法  $f(x+p)-f(x)=0$  并将0换成常数M，就扩充了周期函数的定义，线性p周期函数以周期函数作为特例，用Matlab可以帮助我们观察它应具有图形。从这些观察中，归纳成某种猜想并加以分析论证，这种观察—猜想—论证的过程是数学实验的一种模式。当我们试图去发现某些规律时，特殊例子的观察往往能引导人们走向正确的研究途径，计算机可以帮助我们进行观察，但是却代替不了我们的思维。真正的成果还需要积极思考，大胆假设，小心求证。关键在于能否提出有意义的问题，请读者解决问题2中所提出的问题。

◆ 能否继续扩充周期函数的定义呢？把六个三角函数当作积木块，可以制造出许多新的周期函数。你可以用它们进行各种复合，考察复合函数的图形。观察正弦和余弦的复合函数的形式，例如  $y = \sin(x \cos(x \sin \cos(x + \cos(x))))$ 。注意复合后函数的周期，你也可以改变复合方式，这种复合后的函数的周期会是怎样的？是否不同的复合产生不同的周期？你乐意研究它们吗？你可以预言这种复合函数生成的线性p周期函数性质吗？

◆练习：使用不同的绘图风格(如图形窗口的分割，标注、线型和颜色等等)绘制下面的函数图形：

(1) 绘制幂函数  $y = x^\alpha$  的图形，对不同的  $\alpha$  值，请处理好幂函数的定义域。注意 Matlab 是以复数形式给出数的形式，当虚部为零是变为实数，当虚部不为零时，只绘制实部。

(2) 绘制指数、对数、三角和三角函数的图形，观察它们的变化。

## 2.2 平面截割法和曲面交线的绘制

在多元微积分中，总会遇到各种各样的空间曲面，用 Matlab 绘图功能将它们可视化，有助于加强我们的空间想象能力和教材内容的理解。现在你可以随时在计算机上将在高等数学中所遇到的函数图形绘制出来进行观察思考。也就是说，你已经初步掌握了一种数学实验工具和简单的实验方法。

但是，数学实验并不总是简单的，如果你遇到的问题难度增加，实验的难度也就随之增加。即使绘制一个函数的图形，在实验中有时也会遇到麻烦，需要你仔细分析，找出原因所在，改进实验方法，在实践中不断提高自身的数学修养和水平。

让我们来讨论一个高等数学的例子：平面截割法是分析曲面形状的一个有用方法，该方法的要点是，对给定的空间曲面（例如二次曲面），选定垂直于  $z$  轴的一组平面对曲面进行截割，通过观察各截割面与曲面的交线形状，分析整个曲面的形状。同理可选垂直  $x$  轴或  $y$  轴的平面进行分析，以取得对曲面更完整的了解。

◆用平行截面法讨论由曲面  $z = x^2 - 2y^2$  构成的马鞍面形状。

这样的问题显然比直接画出一个函数的图象要复杂一些，你必须理解平行截面法的原理。在此基础上设计一个好的实验方案，便于观察。

用平行于  $x-y$  的平面族  $z=a$  截割曲面，平面  $z=a$  与曲面交线是双曲线  $x^2 - 2y^2 = a$ ，可以设计一个实验，画出了马鞍面、截割平面以及截割平面和马鞍面的交线；如果能将截割所得的双曲线显示在另一图形中，则能更好的表现平行截割法的内涵。

**program\_5.m** （平行截割法示例，本程序的绘制两曲面交线方法可以套用）

```

【 clf, a=-20;eps0=1;
[x,y]=meshgrid(-10:0.2:10); %生成平面网格
v=[-10 10 -10 10 -100 100]; %设定空间坐标系的范围
colormap(gray) %将当前的颜色设置为灰色
z1=(x.^2-2*y.^2)+eps; %计算马鞍面函数z1=z1(x,y)
z2=a*ones(size(x)); %计算平面 z2=z2(x,y)
r0=abs(z1-z2)<=eps0;
%计算一个和z1同维的函数r0,当abs(z1-z2)<=eps时r0=1; 当abs(z1-z2)>eps0时, r0=0。
%可用mesh(x,y,r0)语句观察它的图形，体会它的作用，该方法可以套用。
zz=r0.*z2;xx=r0.*x;yy=r0.*y; %计算截割的双曲线及其对应的坐标
subplot(2,2,2), %在第2图形窗口绘制双曲线
h1=plot3(xx(r0~=0),yy(r0~=0),zz(r0~=0),'+');
set(h1,'markersize',2),hold on,axis(v),grid on
subplot(2,2,1), %在第一图形窗口绘制马鞍面和平面
mesh(x,y,z1);grid,hold on;mesh(x,y,z2);
h2=plot3(xx(r0~=0),yy(r0~=0),zz(r0~=0),'.'); %画出二者的交线
set(h2,'markersize',6),hold on,axis(v),
for i=1:5 %以下程序和上面是类似的，通过循环绘制一系列的平面去截割马鞍面
a=70-i*30; %在这里改变截割平面
z2=a*ones(size(x)); r0=abs(z1-z2)<=1; zz=r0.*z2;yy=r0.*y;xx=r0.*x;
subplot(2,2,3),
mesh(x,y,z1);grid,hold on;mesh(x,y,z2);hidden off
h2=plot3(xx(r0~=0),yy(r0~=0),zz(r0~=0),'.'); axis(v),grid
subplot(2,2,4),
h4=plot3(xx(r0~=0),yy(r0~=0),zz(r0~=0),'o'
);
set(h4,'markersize',2),hold
on,axis(v),grid on
end
】

```

结果由图14给出。由图可见， $a$ 变化到某一临界值 $a_0$ 处交换双曲线的虚轴和实轴，请对其作理论分析，确定出这一临界值。

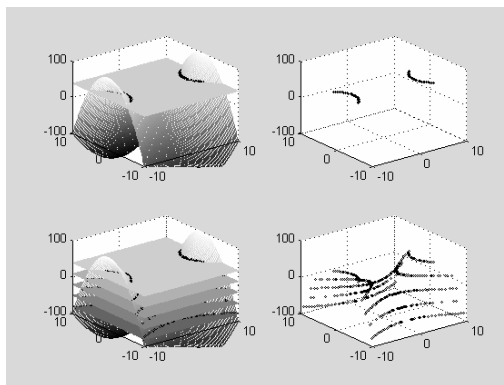


图14 平面截割法分析曲面图形



◆练习

(1) 对马鞍面  $z = x^2 - 2y^2$ ，采用平面  $x = h$  或  $y = h$  截割曲面，观察其图形的变化。

(2) 选取不同的参数  $a, b, c$ ，用平面截割法绘制二次曲面  $\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} = d$  的图形，并讨论参数  $a, b, c$  对其形状的影响。

(3) 绘制曲面  $z = x^2 - 2y^2$ ， $z = 2x - 3y$  的交线。

(4) 在计算二重和三重积分时，常常需要绘制积分区域的图象，例如用三重积分计算由旋转抛物面  $x = \sqrt{y - x^2}$ 、抛物柱面  $\sqrt{y}/2 = x$  及平面  $y = 1$  在空间区域围成的三维立体的体积；

或者用二重积分计算抛物柱面  $z = 4 - x^2$ 、坐标面和平面  $2x + y = 4$  在第一卦限围成的立体体积。你能够绘制出这些立体的图形吗？能够绘出这些立体在某个坐标平面的投影吗？

(5) 对于二重积分，积分指示线方法是很有用的，当然你首先得了解一下什么是积分指示线法，请查阅高等数学相关的内容，然后设计一个数学实验，然后用 Matlab 的绘图工具表现这一方法。

## 2.3 微分方程的斜率场

从高等数学里知道，一阶微分方程具有下面的形式：

$$\frac{dy}{dx} = f(x, y), \quad y(x_0) = y_0$$

如果方程的解存在，则上述方程确定了一条解曲线  $y = y(x, y_0)$ ，这一曲线与初始值  $y_0$  有关。设解曲线  $y = y(x, y_0)$  在点  $(x, y(x))$  处与  $x$  轴的夹角为  $\alpha(x, y(x))$ ，则  $dy/dx = \tan(\alpha(x, y))$  是解曲线  $y = y(x, y_0)$  的斜率，由方程知  $\alpha(x, y(x)) = \arctan(f(x, y))$ 。

一阶微分方程的斜率场，就是在平面上的每一点  $(x, y)$  处，绘制出切线矢量：

$$(\cos(\alpha(x, y)), \sin(\alpha(x, y))), \quad (x, y) \in D \subseteq \mathbb{R}^2$$

从而构成一个矢量场，根据斜率场可以大致看到从某个初始点  $(x_0, y_0)$  出发式，微分方程的解轨线的走向。

◆ 绘制微分方程  $\frac{dy}{dx} = xy, y(0) = 0.4$  的斜率场，并将解曲线画在图中，观察斜率场和解曲线的关系。

用 Matlab 绘制一阶微分方程的斜率场是方便的，下面的参考程序使用了 `quiver`（画向量）指令和 `gradient`（求数值梯度）指令供读者套用。可直接算出方程的解的表达式再画图；也可以用 `ode45` 指令直接求出方程的数值解。

**program\_6.m** (绘制一阶微分方程的斜率场和解曲线)

```

【   clf,clear      %清除当前所有图形窗口的图像，清除当前工作空间的内存变量。
    a=0;b=4;c=0;d=4;n=15;
    [X,Y]=meshgrid(linspace(a,b,n),linspace(c,d,n)); %生成区域中的网格。
    z=X.*Y;                                     %计算斜率函数。
    Fx=cos(atan(X.*Y));Fy=sqrt(1-Fx.^2); %计算切线斜率矢量。
    quiver(X,Y,Fx,Fy,0.5),hold on,axis([a,b,c,d])
    %在每一网格点画出相应的斜率矢量，0.5是控制矢量大小的控制参数，可以调整。
    [x,y]=ode45('program_5f',[0,4],0.4); %求解微分方程。
    %program_5f.m是方程相应函数f(x,y)的程序，单独编制；[x0,xs]=[0,4]为求解区间；
    %y0=0.4为初始值；输出变量x,y分别为解轨
    线自变量和因变量数组。
    plot(x,y,'r.-') %画解轨线  】

```

**program\_5f.m** (微分方程的函数子程序)

```

【   function dy=program_5f(x,y)
    dy=x.*y;      】

```

结果见图15。

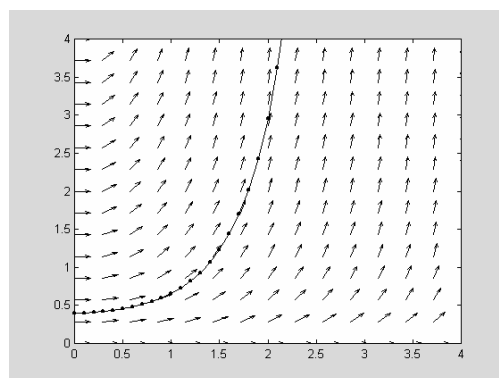


图 15 微分方程的斜率场和解曲线

#### ◆练习

(1) 绘制  $z = \frac{1}{x^2 - 2x + 4} + \frac{1}{y^3 - 2y + 4}$  的等值线图和梯度场。

(2) 绘制  $\frac{dy}{dx} = \sin(x)\sin(y)$  的斜率场并选取初始值，在斜率场中画出解轨线。你遇到什么问题，如何解决？

## 2.4 颜色控制和渲染及特殊绘图指令

### (1) 地球表面的气温分布 (sphere 指令)

◆ 地球表面的气温差异很大，而且随时间变化，要绘制气温分布图绝不是一件容易的事情。但是赤道温度最高，而在两级最冷，在中间地带则是过渡带。所以可粗略将这种气温分布情况用图形表现出来，试绘制地球表面的气温分布示意图。

用一个球体表示地球，用不同颜色表示不同的气温，这样就可用色图表现地球的气温分布。为了有好的视觉效果，可进行色彩渲染。运行下面的程序：

```

【 [a,b,c]=sphere(40);t=max(max(abs(c)))-abs(c);surf(a,b,c,t);
    axis('equal'),colormap('hot'), shading flat,colorbar  】

```

结果见图16。

★生成单位球面 sphere。

[X,Y,Z]=sphere(N)用来生成 3 个(N+1) × (N+1) 矩阵，存放单位球面的 x,y,z 坐标，再用 surf(X,Y,Z)可绘出单位球面。

★色彩控制 surf、colormap。

surf(X,Y,Z,C) 绘制由4个矩阵变量定义的色彩图，三个坐标轴刻度由X、Y、Z的取值方位确定，色轴的取值由C的范围或当前函数Caxis的设置决定，C缺省时相当于C=Z；刻度颜色值被用到当前函数colormap的索引中；渲染模式由函数shading设置。

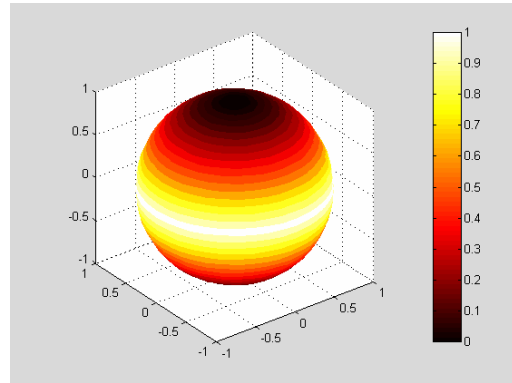


图 16 地球表面气温的分布示意图

表3 色图函数表

MAP	色图性质	MAP	色图性质
hsv	色度饱和度色图（默认）	colorcube	增强立方色图
hot	黑—红—黄—白	jet	hsv的变异图
gray	线性灰度色图	prism	棱镜色图
bone	蓝色色调	cool	青色和洋红阴影
copper	线性纯铜色	autumn	红黄阴影
pink	淡阴影粉红色	spring	洋红黄色阴影
white	全白	winter	蓝色绿色阴影
flag	交互的红白蓝黑	summer	绿色黄色阴影
lines	线性色图		

colormap为色彩调制函数，用来设置当前图窗内色图的性质：colormap(MAP) 设置当前色图为MAP。因为色彩由红绿蓝三色构成，所以色图矩阵可以有任何数目的行数，但列数必须为3，如MAP=[a,b,c]，其中第一个元素a 用来制定红色的强度，第二值b制定绿色的强度，第三值c为蓝色强度值。a、b、c均在[0,1]区间中取值：如[1 0 0]、[0 0 0]、[1 0.62 0.40]分别为纯红、黑、纯铜色等等。也可用色图函数来生成色图，见表3。colormap('default')为系统默认色图hvs，用map=colormap指令可查看当前色图。

★色彩的渲染 shading

色彩渲染主要由shading函数来控制，渲染模式有下面 3 种

shading falt：平坦式渲染；shading interp：插补式渲染；shading faceted：渲染模式为小面式（系统默认）。

★坐标轴控制（axis）： axis equal 设置坐标纵横比相同，这样才能使图16看起来像一个球，否则由于变形看起来像椭球。

## (2) 旋转曲面的生成：柱面指令 cylinder 和光照控制指令 surf

◆一条曲线绕轴旋转可生成旋转曲面，试在三维空间中画出曲线  $z = \frac{1}{x^3 - 2x + 4}$ ，并在同一图形窗口绘制它绕 z 轴旋转得到的旋转曲面，作颜色的渲染和光照控制，并加以适当的标注。

可参考如下程序完成这一练习：

```
【 x=0:0.1:10;z=x;y=1./(x.^3-2.*x+4);
[u,v,w]=cylinder(y);surfl(u,v,w,[45,45]);
shading interp 】
```

效果如图17。

★**cylinder** 指令： 可用来产生柱体， 格式为： **[X,Y,Z] = cylinder(r)**， 其中r为在每个高度处的半径值， 使用时必须配合**mesh**、**surf**或光照控制指令**surfl**才能画出图形。

★**surfl(x,y,z,s)**指令： 三维图形表面的光照图函数， 其中 **x,y,z** 和 **surf** 指令要求的完全相同， 只不过前者是绘制一有光源的光照图， **s** 是用来设定光源的位置向量。在设定光源坐标时， 可采用直角坐标系 **s=[sx,sy,sz]**； 也可以采用球坐标系 **s=[az,el]**， 即光线投射的角度。其默认值为光源的方位角 **az** 在观察点逆时针方向的 40 度， 在程序中**[az,el]=[45 45]**（角度）。可改变**[az,el]**的值， 观察因光源变化产生的不同效果， 例如令 **s=[45 120]**或者 **s=[25 45]**等等。

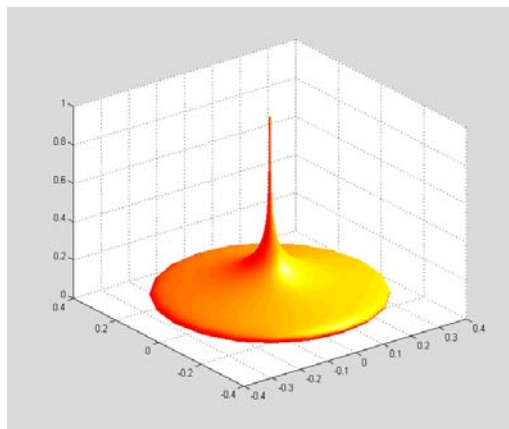


图 17 旋转曲面生成和光照控制

### (3) 若干特殊图形

Matlab 有许多特殊图形绘制指令， 下面列举几个常用的， 更多的绘图指令可参见本书附录中的命令列表， 各个指令的详细用法则可查阅帮助或参阅有关参考书。

**bar(x,y)**： 条形图， 绘制对应于x 的 y的高度； **hist(y,x)**： 直方图， 绘制数据y在以x为中心的区间中分布的频数直方图； **stairs(x,y)**： 阶梯图， 绘制 y 对应于 x 的阶梯图； **stem(x,y)**： 火柴杆图， y 对应于 x 的火柴杆图； **pie(vector,explode)**： 饼图， 显示各统计项目百分比（vector）分布的饼图， 同时用 **explode**凸出显示其中某一项。

★**comet3**： 彗星图， 动画显示质点沿曲线运动的情况， 这是一个很有用的指令。

◆ 运行下面程序， 了解各指令的用法和效果。

```
【 x=[1:10]; y=[5 6 3 4 8 1 10 3 5 6];
subplot(2,3,1),bar(x,y),axis([1 10 1 11])
subplot(2,3,2),hist(y,x),axis([1 10 1 4])
subplot(2,3,3),stem(x,y,'k'),axis([1 10 1 11])
subplot(2,3,4),stairs(x,y,'k'), axis([1 10 1 11])
subplot(2,3,5), x = [1 3 0.5 5];explode = [0 0 0 1];pie(x,explode)
subplot(2,3,6),z=0:0.1:100; x=sin(z);y=cos(z).*10;
comet3(x,y,z) 】
```

效果如下图18。

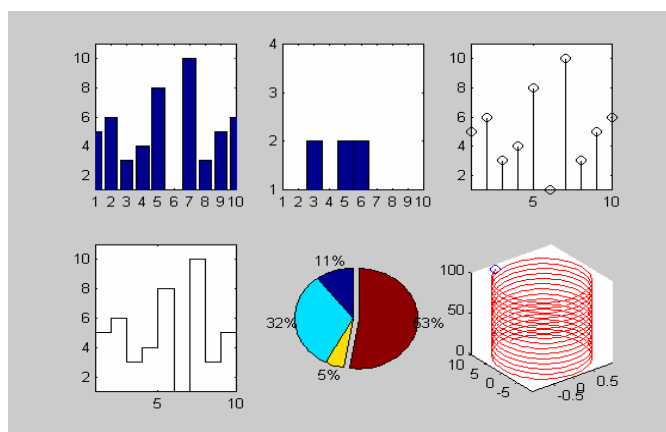


图18几个特殊绘图指令的效果示意