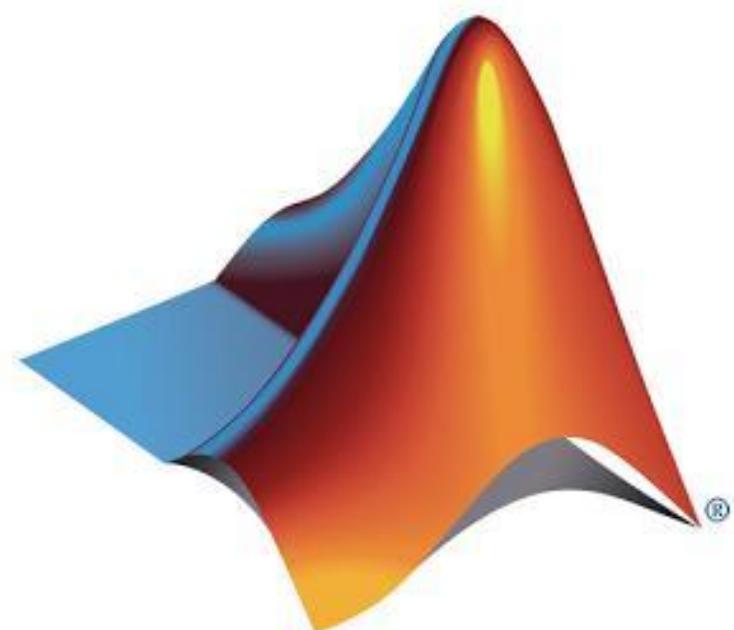
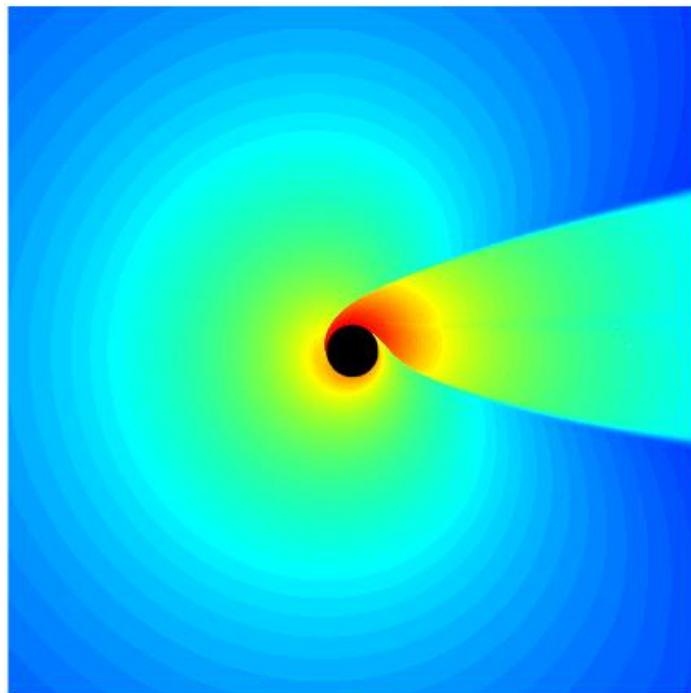


数学实验

Mathematical Experiments



实验一：

函数与数据可视化实验

Visualize functions and data

- 复杂的函数图像难以想象和准确手工绘制
 - 人们很难从一大堆原始数据中感受到它们的含义
-
- 数学的很多现象和原理可以用图像直观生动地表达出来。
 - 图形能帮我们直观地理解函数性质、发现数据的本质规律。

实验1.1 一元函数的可视化

■ 考虑 $f(x) = \sin(\cos(x + \sin(x)))$

$$g(x) = 0.2x + \sin(\cos(x + \sin(x)))$$

■ $f(x), g(x)$ 是周期函数吗？观察它们的图象。

■ 程序

```
clf, x=linspace(0,8*pi,100);
y1=sin(x+cos(x+sin(x)));
y2=0.2*x+sin(x+cos(x+sin(x)));
plot(x,y1,'k:',x,y2,'k-');
legend('sin(x+cos(x+sin(x))','0.2x+sin(x+cos(x+sin(x))))')
```

绘制平面曲线(**plot**指令)

- **plot(x,y):**

- 以**x**为横坐标、**y**为纵坐标绘制二维图形
- **x,y**是同维数的向量；

- **plot(y):**

- 相当于**x=[1,2,...,length(y)]**时情形。

绘制平面曲线（绘制多个图形）

- 1. **plot(x,[y1;y2;...]),**
 - x是横坐标向量， [y1;y2;...]是由若干函数的纵坐标拼成的矩阵
- 2. **plot(x,y1), hold on, plot(x,y2), hold off**
- 3. **plot(x,y1,x,y2,...)**
- 4. **plotyy**
 - 两个坐标系，用于绘制不同尺度的函数。

绘制平面曲线（绘制多个图形）

在数值比较过程中，有时会遇到双纵坐标(即双**y**轴坐标系)显示的要求，为解决该问题，**MATLAB**软件提供了双纵坐标绘制二维图的函数**plotyy**。调用格式：

- **plotyy(X1,Y1,X2,Y2,fun1,fun2)**

表示以**fun1**方式绘制(**X1, Y1**)，以**fun2**方式绘制(**X2, Y2**)。**fun1**、**fun2**可以选择的方式为**plot**、**semilogx**、**semilogy**、**loglog**等。

- 例： 使用双纵坐标绘制曲线 $y = \sin[f(x)]$ 与 $y = 3^x$ 。

绘制平面曲线 (线型、点形和颜色的控制)

- **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',...)**
- 句柄图形和**set**命令改变属性值，可套用：
 - **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。

绘制平面曲线 (属性变量和属性值)

- 线宽: **LineWidth**
- 点的大小: **MarkerSize**
- 线型: **LineStyle**
- 颜色: **color**

表 2-1 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 2-2 (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	+号	*号	方形单词

图形元素设定

颜色	标记	线型		
b g r c m y k w	蓝 (默认) 绿 红 青 洋红 黄 黑 白	无标记 (默认) 点 圈 叉 十字 星 方块 菱形 下三角形 上三角形 左三角形 右三角形 五角形 六角形	- : -. --	实线 (默认) 虚线 点画线 划线

绘制平面曲线（例）

- 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。
- 程序

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

绘制平面曲线 (坐标轴的控制)

- **axis**指令
- **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 zmax]):**
设定三维图形的坐标范围；
- 其中**xmin<x<xmax**, **ymin<y<ymax** ,
zmin<z<zmax。

绘制平面曲线 (**gca**属性控制)

- 改变当前轴对象句柄**gca**属性
- 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：
 - **set(gca,'ytick',[-1 -0.5 0 0.5 1])**
将 **y** 坐标按向量**[-1 -0.5 0 0.5 1]** 将刻度分成**4**格；
 - **set(gca,'yticklabel',{'a','b','c','d','e'})**
改变**y**坐标刻度的说明。

绘制平面曲线（**gca**属性控制,例）

- 设置**y**坐标的刻度并加以说明，并改变字体的大小。
- 程序

```
plot([0:0.1:2*pi],sin([0:0.1:2*pi]),'k.-'); grid on,  
axis([0 6.3 -1.1 1.1]),  
set(gca,'ytick',[-1 -0.5 0 0.5 1]),  
set(gca,'yticklabel',{'a','b','c','d','e'}),  
set(gca,'fontsize',20)  
get(gca)
```

运行结果

绘制平面曲线（文字标注）

- **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。
- 文字标注常用符号:
- **\pi (π) ; \alpha (α) ; \beta (β) ;**
- **\leftarrow (左箭头) \rightarrow (右箭头) ;**
- **\bullet (点号)**

绘制平面曲线 (程序讲解)

```
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','time  
s New Roman'),  
 xlabel('\it{t(deg)}');  
 ylabel('\it{magnitude}');  
 title(' \it{sine wave and }{\it{Ae}}^{{-}}  
 {\it{alpha}}{\it{itt}}{\it{wave}}');
```

绘制平面曲线 (程序讲解)

```
text(6,sin(6),'\fontsize{15}The Value
\it{sin(t)} at {\itt}=6\rightarrow\bullet',
'HorizontalAlignment','right'),
text(2,3*exp(-0.5*2),...
['\bfs{15}\bullet\leftarrow The Value
of \it{3e}^{-0.5} \it{t}=',...
num2str(3*exp(-0.5*2)), ' at \it{t} =2 ']);
legend('\it{sin(t)}','{\it{Ae}}^{-\alpha}')
```

- 注1: **num2str**:
['string1' ,num2str,'string2'], 用方括号
- 注2: **legend** 请结合图形观察此命令的使用

图形窗口的创建和分割

- **subplot(m,n,k)**命令。
- 在图形区域中显示多个图形窗口。
- **m**为上下分割数，**n**为左右分割数，**k**为第**k**子图编号。
- 例：将一个图形分为**9**个子图，在第**k**个子图画**sin(kx)**的图象。

• 程序：

```
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
```

若干有用的指令

- **clf:** 清除图形窗口已有的内容.
- **figure(n):** 打开第**n**个图形窗口
- ... : 续行号

实验1.2 正弦函数的叠加

■ 分别画出区间[0,10*pi] 上的函数

$$y = \sin x + \frac{1}{3} \sin 3x,$$

$$y = \sin x + \frac{1}{3} \sin 3x + \frac{1}{5} \sin 5x, \dots$$

$$y = \sum_{k=1}^m \frac{1}{2k-1} \sin(2k-1)x$$

其中m可以试验从小到大， 观察有什么规律？

- 子图
- 收敛？（什么意义下）
- 声音

- 声音是由振动产生的.乐音有三要素：音调、响度、音色。音调反映声音的高低，由振动频率决定.响度反映声音的强弱，由振幅决定。
- 音色反映什么呢?比如笛子的声音与提琴的声音，即使音调和响度都相同，仍能听出明显的差别.不同的人、不同的物体发出的声音也都是这样，除了音调和响度外还有它们各自的特点，这个特点就是音色.
- 音色是由什么决定的呢?

- 作为振动，乐音也可以用周期函数来描述，并且也可以作Fourier分解：分解为不同频率的正弦、余弦函数的合成，其中**频率最低的称为基音**，其余的正、余弦函数的频率都是基音的整倍数，称为**泛音**。

傅里叶级数, sine-cosine 形式

$$\frac{a_0}{2} + \sum_{n=1}^N \left(a_n \cos\left(\frac{2\pi nx}{P}\right) + b_n \sin\left(\frac{2\pi nx}{P}\right) \right)$$

- 乐音的音色就是由这个分解式中各系数的相对比例决定的，也就是各泛音的响度与基音的响度的相对比例决定的。电子琴之所以能模拟不同乐器（钢琴、小提琴、大提琴、长笛等）的声音，就是通过改变各泛音与基音响度的比例来实现的。

实验1.3 无极限的例子

- 在区间[-1,1]上作出函数 $y = \sin \frac{1}{x}$ 的图像，并观察 $x \rightarrow 0$ 时的变化情况。

请课后尝试： $y = x \sin \frac{1}{x}$?

实验1.3 无极限的例子

- 在以上曲线 $y = \sin \frac{1}{x}$ 取一部分点，比如令

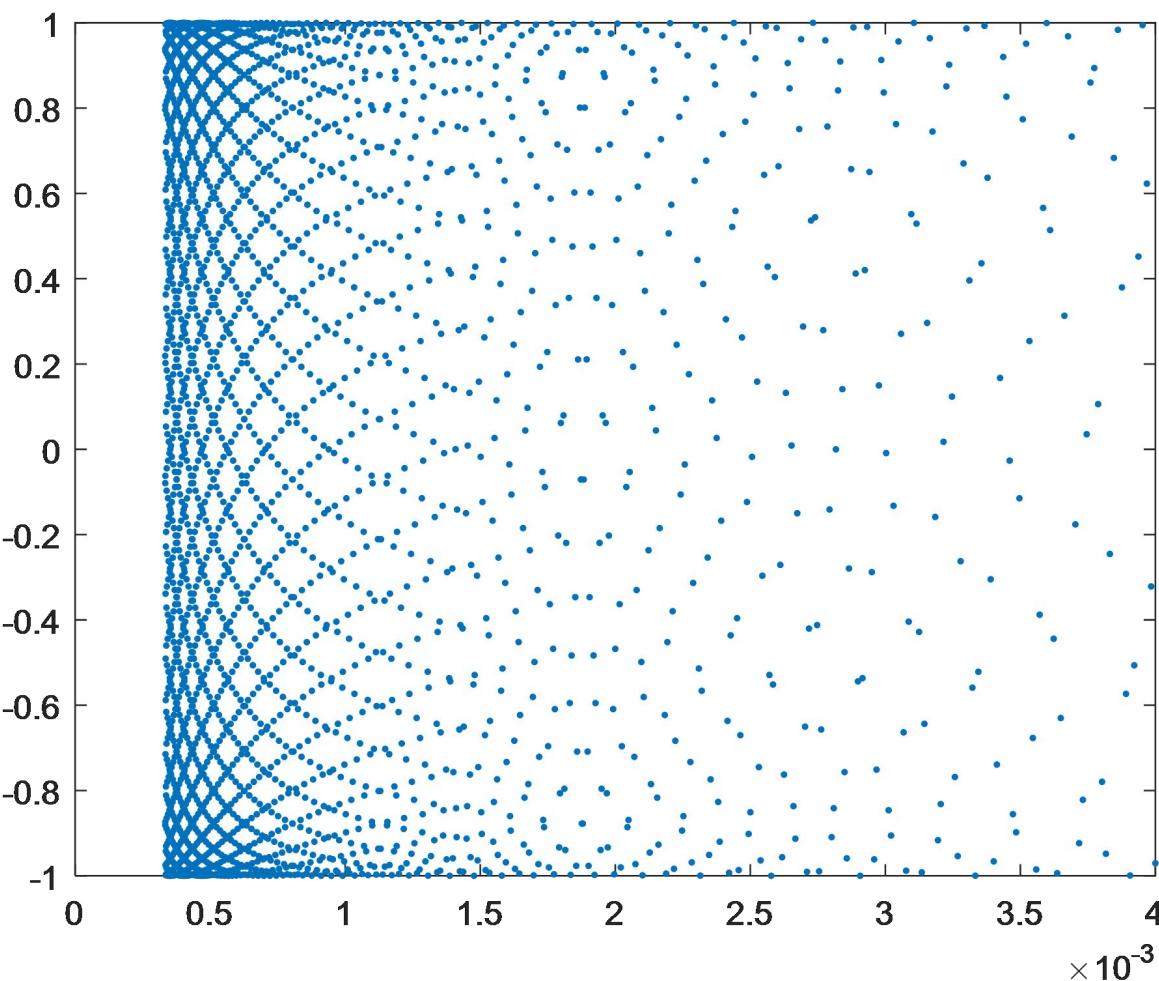
$$x = \frac{1}{k} \quad (\mathbf{k=1,2,...,3000})$$

则当k增加时， x向0趋近， 相应的y的值为
sin1, sin2, ... sin3000。

- 这样就在曲线上取出了3000个点 $(\frac{1}{k}, \sin k)$ 。
- 将这3000个点画在同一个坐标系中，看看它们组成的图形是什么样子？

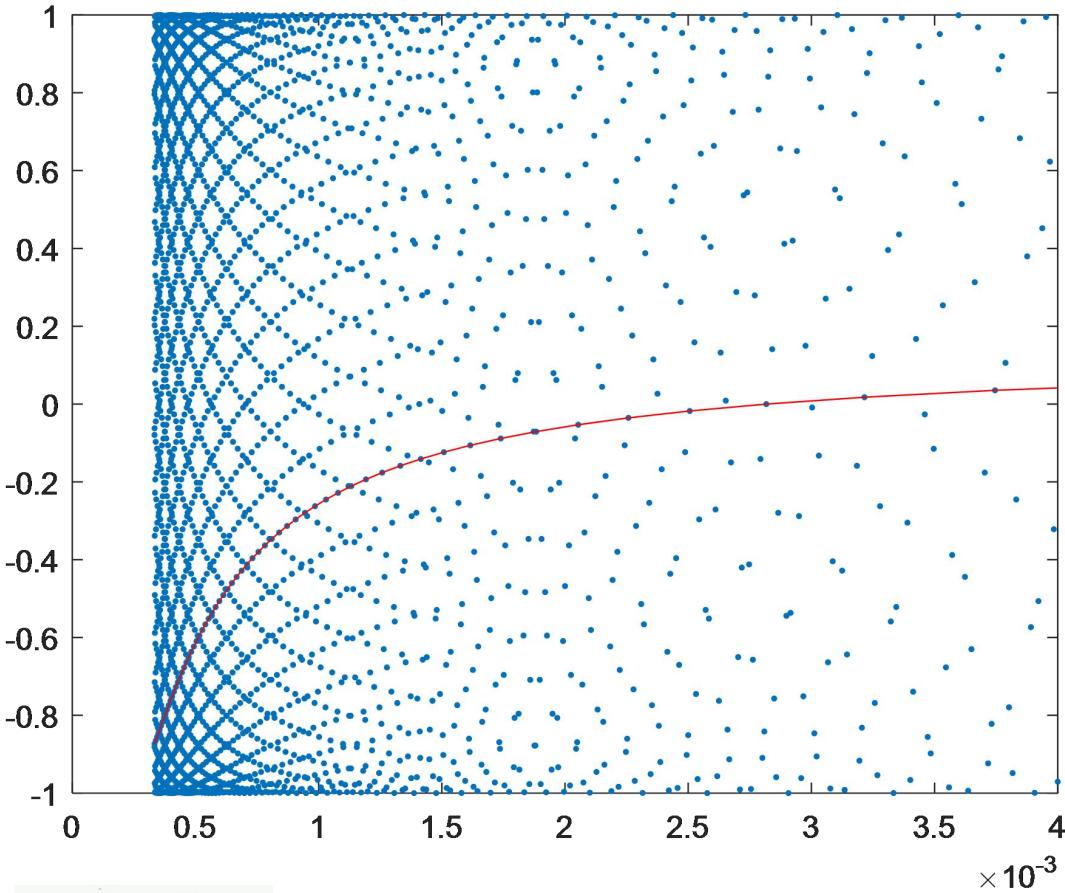
实验1.3 无极限的例子

令人意外的是：这个图形不仅不是一塌糊涂、杂乱无章，反而很有规律，呈现出一些美丽的曲线组成一张网！



- 图中呈现出来的是什么曲线？
- 能把其中某一条曲线（找出）单独画出来吗？
- 即能否分辨出这300个点当中哪些点组成我们看到的曲线

实验1.3 无极限的例子



$$A_k \left(\frac{1}{k}, \sin k \right)$$

$$A_m \left(\frac{1}{m}, \sin m \right)$$



约率: $\frac{22}{7}$, 密率: $\frac{355}{113}$

$$d(k, m) = \sqrt{\left(\frac{1}{k} - \frac{1}{m}\right)^2 + (\sin k - \sin m)^2}$$

实验1.4 符号函数画图

MATLAB系统提供了ezplot与fplot两个符号函数画图函数，以实现显函数、隐函数及参数方程绘图。调用格式如下：

□ `ezplot('f',[xmin,xmax])`

表示绘制函数f在 $xmin < x < xmax$ 内的图形，当区间[a,b]默认时，默认区间是 $[-2\pi, 2\pi]$ ；

□ `ezplot('f',[xmin,xmax,ymin,ymax])`

表示绘制函数f在 $xmin < x < xmax$ 与 $ymin < y < ymax$ 区域内的图形，当“[xmin, xmax, ymin, ymax]”项默认时，默认x与y的范围都是 $[-2\pi, 2\pi]$ ；

□ `ezplot(x,y,[tmin,tmax])`

表示绘制参数方程 $x=x(t)$ 与 $y=y(t)$ 在区间 $tmin < t < tmax$ 内的图形；

□ `fplot('fun',[xmin,xmax])`

表示绘制字符串fun 所指定的函数在 $xmin < x < xmax$ 内的图形，fun必须是M文件的函数名或是独立变量x的字符串。

实验1.4 符号函数画图

例1 使用ezplot函数， 分别绘制下述曲线：

(1) $y = \sin[x], x \in [-4\pi, 4\pi];$

(2) $x^2/4 - y^2/9 = 1, x \in [-5, 5], y \in [-3, 3];$

(3) $x^2 + x \cdot e^{(y^2)} - \sin[x] + y \cdot e^{(x^2)} = 0, x \in [-3, 3], y \in [-3, 3];$

(4) $x = \cos^3[t], y = \sin^3[t], x \in [0, 2\pi], y \in [-1, 1].$

```
>>ezplot('sin(x)',[-4*pi,4*pi]);
```

```
>>ezplot('x^2/4-y^2/9=1',[-5,5,-3,3]);
```

```
>>ezplot('x^2+x*exp(y^2)-y*sin(x)+y*exp(x^2)',[-3,3,-3,3]);
```

```
>>ezplot('cos(t)^3','sin(t)^3',[0,2*pi,-1,1]);
```

实验1.4 符号函数画图

例2 使用fplot函数， 分别绘制下述曲线：

(1) $y_1 = \sin[x], y_2 = \cos[x], x \in [-2\pi, 2\pi]$;

(2) $y = e^x + \sin[x], x \in [-1, 1]$ 。

```
>> fplot('sin(x),cos(x)',[-2*pi,2*pi]);
```

```
>> title('y1=sin(x),y2=cos(x)');
```

```
>> fplot('exp(x)+sin(x)',[-1,pi]);
```

```
>> title('y=e^x+sin(x)');
```

实验1.5 对数坐标画图

在考察两变量之间的函数关系时，某些问题通过对数据进行对数转换可以更清晰地看出数据的某些特征。

- MATLAB系统提供对数转换的方式有双对数坐标转换和单轴对数坐标转换两种。
- `loglog`函数可以实现双对数坐标转换
- `semilogx`、`semilogy`函数可分别实现x、y轴单轴对数坐标转换。

实验1.5 对数坐标画图

例1 分别使用直角坐标系与对数坐标系，
绘制变量x与y曲线图：

$$(1) y = 2^x; (2) y = \ln x.$$

实验1.6 极坐标图

MATLAB软件提供了polar函数绘制极坐标系下二维曲线，该函数的使用格式：
`polar(theta,rho,'s')`

其中theta为弧度表示的角度向量； rho为对应的极径； s用于控制线型与颜色。

例 使用polar函数绘制以下曲线： (1) $y = 2(1 - \cos\theta)$;
(2) $y = 3 \cos(4\theta)$

实验1.7 散点图

分析两变量间函数关系，常常需要绘制散点图。MATLAB软件提供了scatter、scatter3命令分别用于绘制二维、三维散点图，使用格式：

scatter(X, Y, S, C): 用于绘制二维散点图，表示在向量 X 与 Y 的指定位置显示标识点， X 与 Y 必须大小相同， S 用于控制标识点的大小， C 用于控制标识点的形状；

scatter3(X, Y, Z, S, C): 用于绘制三维散点图，表示在向量 X 、 Y 、 Z 的指定位置显示标识点， X 、 Y 、 Z 必须大小相同， s 与 C 的用法同上。

例 使用MATLAB生成二维与三维数据点，并绘制的散点图。

常用的作用图命令

函数/命令	意义	函数/命令	意义
plot	基本二维图形	ylabel	y轴说明
fplot	一元函数图像	zlabel	z轴说明
plot3	空间曲线	clabel	等高线高度标志
meshgrid	网格数据生成	grid	格栅
mesh	网面图	hold	图形保持
surf	曲面图	axis	定制坐标轴
contour	等高线图	view	改变视点
comet	二维动画	subplot	子图
contour3	三维等高线图	figure	新图形窗口
comet3	三维动画	clf	清除图形
title	标题	close	关闭图形窗口
xlabel	x轴说明	ndgrid	n维网格数据生成

其他特殊函数图

名称	说明	名称	说明
area	区域填充图	quiver(quiver3)	二维矢量图（三维）
bar(bar3)	条形图（三维）	stairs	阶梯图
barh(bar3h)	水平条形图（三维）	meshc	带等高线网格图
comet(comet3)	彗星图（三维）	meshz	带垂帘线网格图
errorbar	误差带图	surf	带等高线着色图
feather	箭号图	trimesh	三角形网格图
fill	多边形填充图	trisurf	三角形表面图
hist	统计直方图	waterfall	瀑布图
pie(pie3)	饼图（三维）	cylinder	柱面图
stem(stem3)	火柴杆图（三维）	sphere	球面图

实验1.8 其他图形举例

使用**MATLAB**生成二维与三维数据点，并绘制以下图形：

- (1) 条形图；
- (2) 箭号图；
- (3) 统计直方图；
- (4) 饼图；
- (5) 火柴杆图；
- (6) 矢量图；
- (7) 多边形填充图；
- (8) 区域填充图。

实验1.9 绘制二元函数

- 基本步骤：

- 1.**生成二维网格点
- 2.**计算函数在网格点上的值
- 3.**绘制函数图形

三维绘图 (meshgrid 指令：生成网格点)

- 观察**meshgrid**指令的效果。
- 程序：
- **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,'+')**
- 观察结果

三维绘图 (例子)

- 画如下函数图像

$$f(x, y) = \frac{\sin \sqrt{x^2 + y^2}}{\sqrt{x^2 + y^2}}$$

讲解**Exp1_9b.m**

三维绘图 (绘图指令)

- **mesh(X,Y,Z)** :
 - 在三维空间中绘出由**(X,Y,z)**表示的曲面;
- **meshz(X,Y,Z)**:
 - 除了具有**mesh**的功能外，还画出上下高度线，
- **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，
- **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，
 - 运行**Exp1_9c.m**, 观察效果，比较，设置...

三维绘图 (等值线指令, 续)

MATLAB软件提供了绘制二维和三维等高线图的函数**contour** 和 **contour3**，使用格式：
contour(X,Y,Z,n/V)

表示绘制二维等高线图。其中变量**Z**必须为一数值矩阵，变量**X**、**Y**可省略；**n/V**为选择输入参数，若输入正整数**n**，表示绘制等高线的条数为**n**；若输入向量**V**，等高线的条数为向量**V**的长度，并且等

高线的值为对应向量元素的值；若**n/V**省略，等高线的条数为预设值**10**。若对等高线进行数值标注，可使用**clabel**函数。**contour3**使用格式与**contour**类似

例 绘制下述图形：

(1) **peaks**函数曲面图；

(2)**peaks**二维等高线图**n=15**；

(3)**peaks**三维等高线图**n=15**；

(4)对**peaks**二维等高线图**n=5**进行数值标注。

三维绘图（等值线指令）

- 表现二维函数的图形的另一种方式是绘制等值线图。
- **contour(X,Y,z,n):**
 - **n**条等高线， **n**可缺省；
- **contourf(X,Y,z,n):**
 - 等值线间用不同的颜色填满，有更好的视觉效果；
- **contour3(X,Y,z,n):**
 - 在三维空间画出等值线图
- **colorbar:**
 - 将颜色与函数值对应起来显示在图中。

三维绘图 (等值线指令, 显示效果)

- **clf,contour(X,Y,z,40),colorbar**
- **contourf(X,Y,z,40),colorbar**
- **contour3(X,Y,z,40),colormap([0,0,0])**
- 为等值线标上函数值:
 - 可套用下面程序的格式.
 - **[cs,h]=contour(X,Y,z,15);**
 - **clabel(cs,h,'labelspacing',244)**
- **labelspace**是数值标记之间相隔的宽度, 默认值为**144**, 这里取了**244**,

三维绘图 (pcolor指令)

- 表现二维函数的图形的另一种方式是绘制纹影图（伪色图）。

pcolor(X,Y,C);

- 使用矩阵 **C** 中的值创建一个伪彩图。伪彩图以彩色单元（称为面）阵列形式显示矩阵数据。矩阵 **C** 指定顶点处的颜色。**C** 的大小必须与 **x-y** 坐标网格的大小匹配。例如，如果 **X** 和 **Y** 定义一个 $m \times n$ 网格，则 **C** 必须为 $m \times n$ 矩阵。
- shading ('interp')** 可以实现对不平滑的图像进行平滑
- 运行**Exp1_9e.m**, 比较效果。

实验1.10空间曲线和运动方向

- 一条空间曲线可以用矢量函数表示为

$$\Gamma: \quad r(t) = \{x(t), y(t), z(t)\}, \quad t \in [\alpha, \beta]$$

- 它的速度矢量表现为曲线的切矢量：

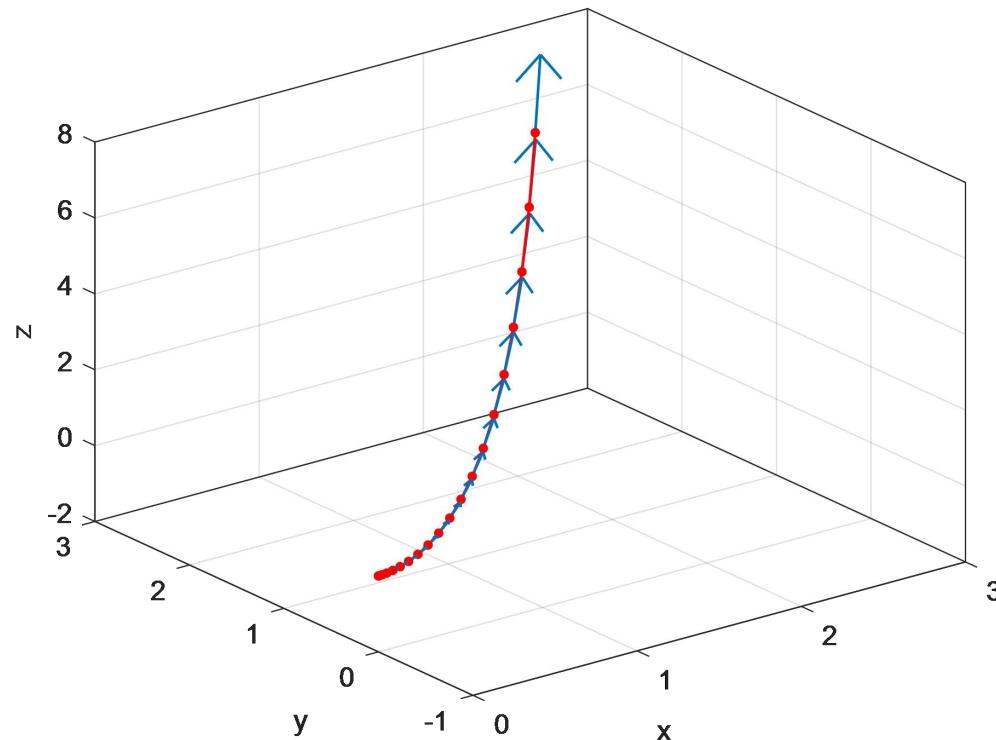
$$v(t) = \frac{dr}{dt} = \left(\frac{dx(t)}{dt}, \frac{dy(t)}{dt}, \frac{dz(t)}{dt} \right)$$

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。

实验1.10空间曲线和运动方向

■很显然飞行曲线方程为：

$$r(t) = \int_0^t (2t, 2t^2, 6t^3 - t^2) dt = (t^2, (2/3)t^3, \frac{3}{2}t^4 - \frac{1}{3}t^3)$$



绘制空间曲线（指令）

- **plot3(x,y,z):**
 - 绘制三维空间曲线，用法和**plot**类似。
- **quiver(X,Y,u,v):** 绘制二维矢量，
 - 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，其维数不小于**2**。
- **quiver3 (X,Y,Z,u,v,w) :**
 - 绘制三维矢量，用法与**quiver**类似。
- **Gradient:**
 - **[Fx,Fy,Fz]=gradient(F)**为函数**F**数值梯度

绘制空间曲线（程序讲解）

```
clf, t=linspace(0,1.5,20);%:0.1:1.5;
x=t.^2;y=(2/3)*t.^3;z=(6/4)*t.^4-(1/3)*t.^3;
Vx=2*t;Vy=2*t.^2;Vz=6*t.^3-t.^2;
plot3(x,y,z,'r-
 $\cdot$ ,'linewidth',1,'markersize',10),hold on
h=quiver3(x,y,z,Vx,Vy,Vz);set(h,'linewidth',1),
grid on
xlabel('x'),ylabel('y'),zlabel('z')
box on
• 运行程序Exp1_10.m
```

向量场和相图

- 除了quiver指令， streamslice指令也可用于绘制向量场和相图。
- 运行程序Exp1_10b.m，观察，并学习存eps矢量图文件

实验1.11 平面截割法和曲面交线的绘制

- 用平行截面法讨论由曲面

$$z = x^2 - 2y^2$$

构成的马鞍面形状。

- 程序Exp1_11.m，观察和思考

实验1.12 动画

- 方式一：以质点运动轨迹的方式显示

使用**comet**、**comet3**函数，前者是二维，后者是三维

comet (**y**) 显示质点绕向量**y**，**comet** (**x**, **y**) 显示质点绕向量**y**与**x**，
comet(x,y,p)，其中为轨迹尾巴的长度
以**comet** (**x**, **y**) 为例：

实验1.12 动画

- 方式一：以质点运动轨迹的方式显示
显示平抛运动

$$\begin{cases} x = v_x t \\ y = -\frac{1}{2} g t^2 \end{cases}$$

vx = 40;
t = 0:0.001:10;
x = vx*t;
y = -9.8*t.^2/2;
comet(x,y)

实验1.12 动画

- 方式一：以质点运动轨迹的方式显示

显示导弹发射

$$\begin{cases} x = v_x t \\ y = v_y t - \frac{1}{2} g t^2 \end{cases}$$

`vx = 100*cos(1/4*pi);`

`vy = 100*sin(1/4*pi);`

`t = 0:0.001:15;`

`x = vx*t;`

`y = vy*t-9.8*t.^2/2;`

`comet(x,y)`

实验1.12 动画

- 方式一：以质点运动轨迹的方式显示匀速圆周运动

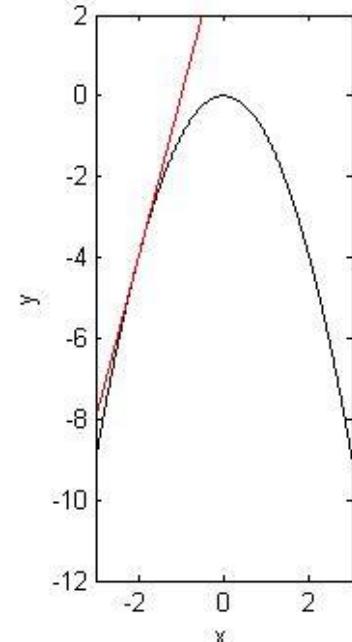
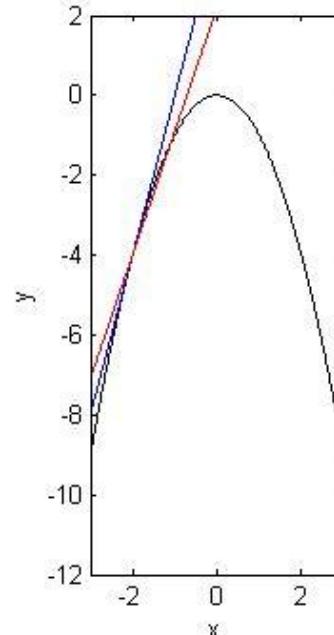
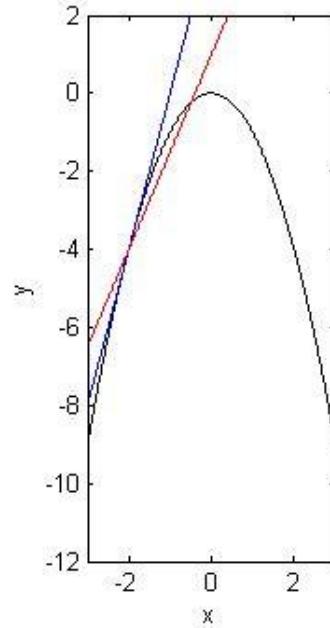
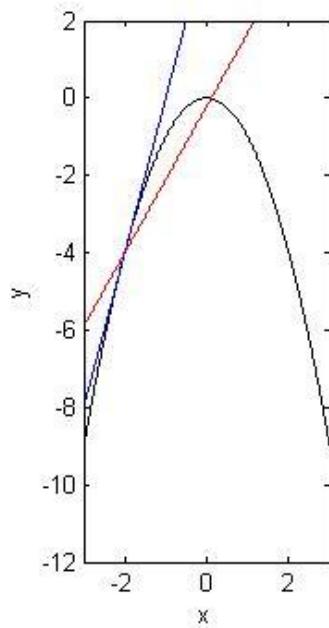
$$\begin{cases} x = r \cos(\theta) \\ y = r \sin(\theta) \end{cases}$$

```
theta = 0:0.0001:2*pi;  
r = 10;  
x=r*cos(theta);  
y=r*sin(theta);  
comet(x,y)
```

实验1.12 动画

- 方式二：以电影播放的方式显示
割线、切线与导数的实验（编写程序动态展示）

$$f'(x_0) = \lim_{\Delta x \rightarrow 0} \frac{\Delta y}{\Delta x} = \lim_{\Delta x \rightarrow 0} \frac{f(x_0 + \Delta x) - f(x_0)}{\Delta x}$$

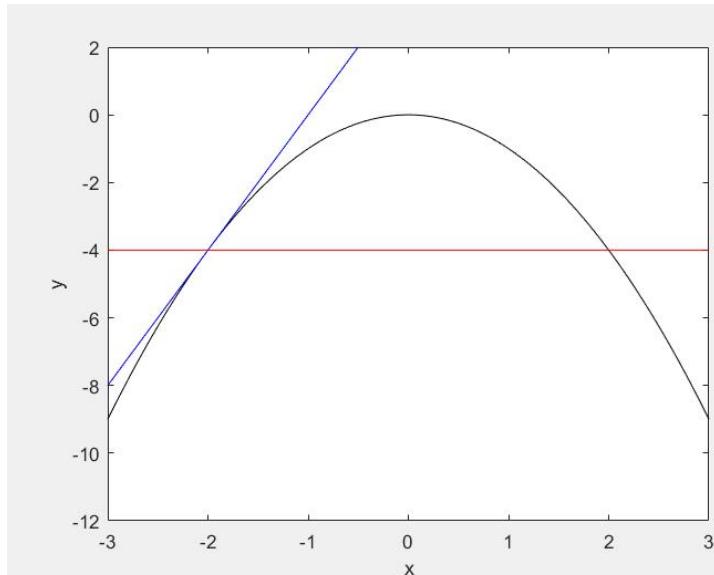


实验1.12 动画

- 方式二：以电影播放的方式显示割线、切线、与导数的实验

$$f'(x_0) = \lim_{\Delta x \rightarrow 0} \frac{\Delta y}{\Delta x} = \lim_{\Delta x \rightarrow 0} \frac{f(x_0 + \Delta x) - f(x_0)}{\Delta x}$$

```
for n=1:100  
frame(n)=getframe  
end  
movie(frame)
```



实验1.12 动画

- 方式三：以**gif**格式存储
割线、切线、与导数的实验

$$f'(x_0) = \lim_{\Delta x \rightarrow 0} \frac{\Delta y}{\Delta x} = \lim_{\Delta x \rightarrow 0} \frac{f(x_0 + \Delta x) - f(x_0)}{\Delta x}$$

```
for n=1:100
frame(n)=getframe(gcf);
end
printgif('test1_11b.gif',frame,0.1);
%printgif 非内置函数
```

实验1.12 动画

- 方式三：以**gif**格式存储
割线、切线、与导数的实验

$$f'(x_0) = \lim_{\Delta x \rightarrow 0} \frac{\Delta y}{\Delta x} = \lim_{\Delta x \rightarrow 0} \frac{f(x_0 + \Delta x) - f(x_0)}{\Delta x}$$

```
for n=1:100
frame(n)=getframe(gcf);
end
printgif('test1_11b.gif',frame,0.1);
%printgif 非内置函数
```

实验1.12 动画

- 摆线动画实验

摆线及其方程：一个圆在直线上滚动而不滑动时，圆周上的一点**M**所绘出的曲线称为摆线，**r**为圆的半径，**t**为转动角度。

已知摆线的参数方程如下：

$$\begin{cases} x = r(t - \sin t) \\ y = r(1 - \cos t) \end{cases}$$

当**t**从**0**变到 **2π** 时， 点**M**的轨迹称为摆线的一拱。

- 实验任务：请用动画表示摆线一拱的构造过程 (**$0 \leq t \leq 2\pi$**) 。

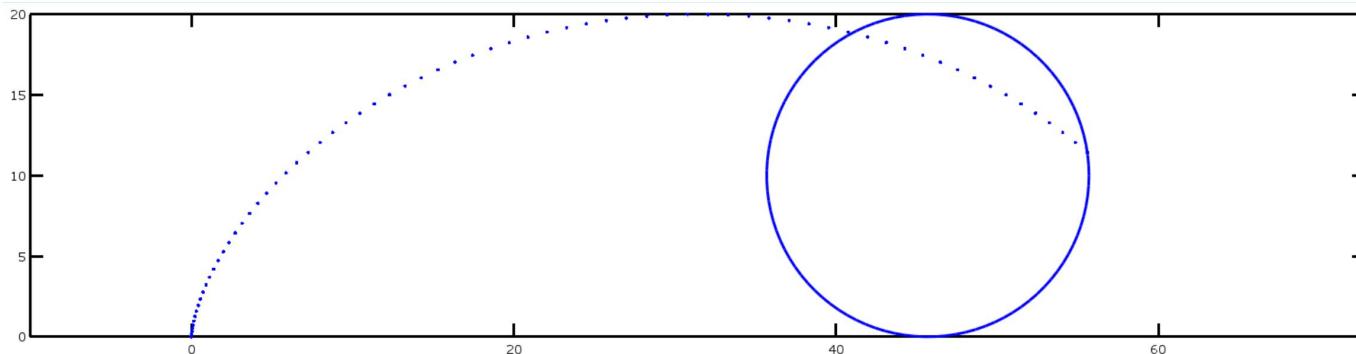
实验1.12 动画

- 摆线动画实验

分析：

1.为了用动画表示摆线构造过程，需要将**t**的取值区间**[0,2π]**离散化，绘制当**t=T**时的如下图像：

点**M**随转动角度**t**从**0**变化到**T**的运动轨迹；
正在运动的圆



实验1.12 动画

- 摆线动画实验

分析：

2. 绘制在 $t=T$ 时圆：需要先确定给定 t 时圆的方程。

设圆的转动角度为 t 时，点M坐标为(X,Y)，设运动圆的圆心坐标 (x_0, y_0) 。

可知 $x_0=X+a \sin t, y_0=r$ ，得圆的方程 $(x - x_0)^2 + (y - r)^2 = r^2$.

也可表示出该圆的参数方程： $\begin{cases} x = x_0 + r \cos \theta \\ y = r + r \sin \theta \end{cases}$

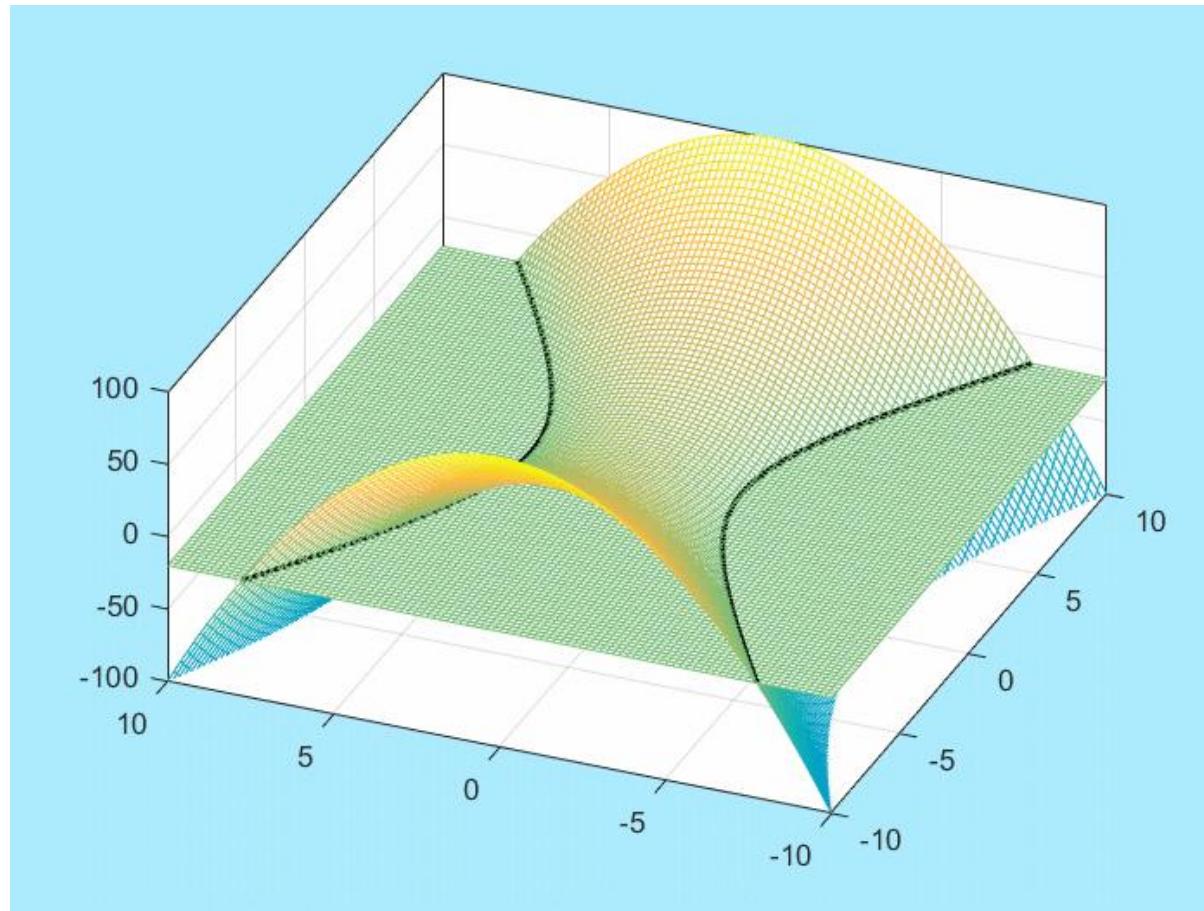
实验1.12 动画

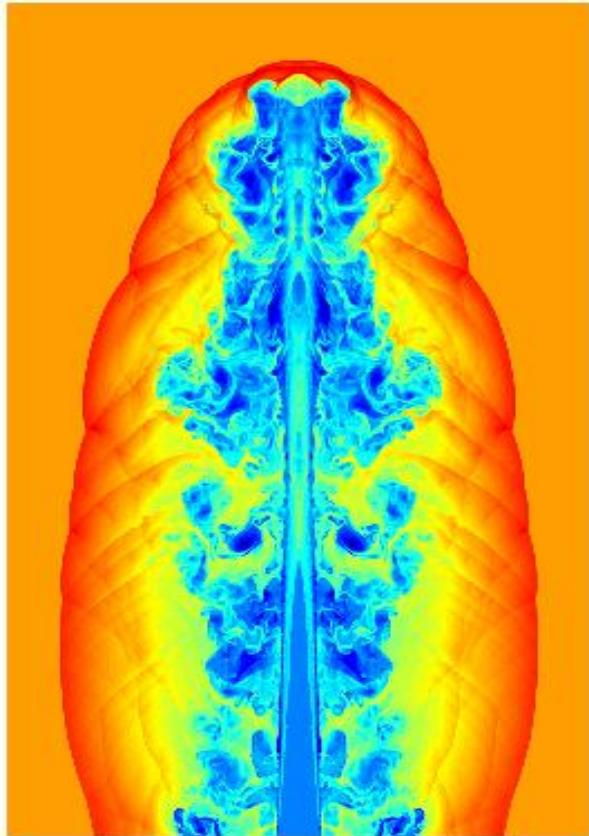
- 摆线动画实验（程序解释）

```
r=10;t=linspace(0,2*pi,100);%转动角度离散化  
X=r*(t-sin(t));Y=r*(1-cos(t));%计算摆线一拱离散点的坐标  
for k=1:length(t)  
    hold off  
    plot(X(1:k),Y(1:k),'b-','linewidth',1)%绘制摆线上离散点  
    x0=X(k)+r*sin(t(k));y0=r;%圆心  
    x1=x0+r*cos(t);y1=y0+r*sin(t);%圆的坐标(x1,y1)  
    hold on,plot(x1,y1,'r','linewidth',3)%绘制圆  
    h=plot(X(k),Y(k),'bo','MarkerFaceColor','blue') %绘制摆  
线上离散点  
    axis([-r,(2*pi*r+r),0,2*r]),axis equal  
    set(gca,'fontsize',14,'linewidth',3),pause(0.05),  
end
```

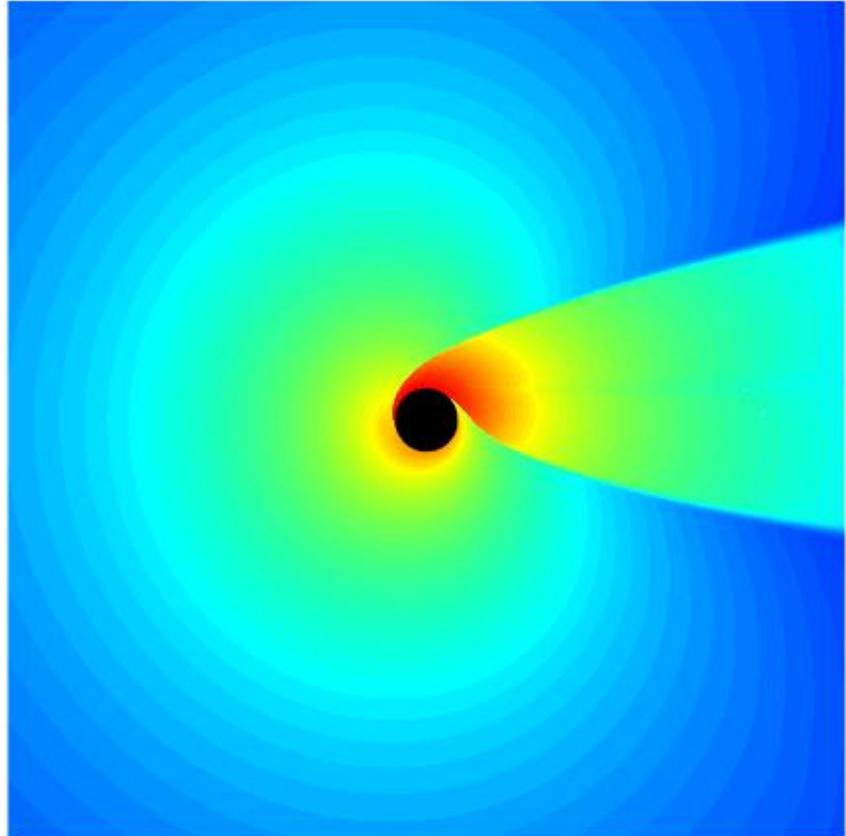
实验1.12 动画

- 思考：如何将前面画的三维图变成以下动画？





高速射流



黑洞吸积

随堂实验

函数与数据可视化实验

- 实验1： 分别用 plot, ezplot, fplot, 画出函数 $y = x \sin \frac{1}{x}$ 及其导函数在区间 $[-1, 1] \setminus \{0\}$ 上的图像，并使用 xlabel, ylabel, title 等命令。通过观察图像简要说明该函数的性质。
- 实验2：写一个M文件函数，实现一元二次方程求解：
 - 函数输入为 a, b, c ；
 - 输出为方程 $ax^2 + bx + c = 0$ 的两个解；
 - 注意考虑特殊情况和细节。鼓励采用多种不同方法。
- 实验3：寻找拐点问题（详见下一页）

寻找拐点的问题

如果我们已经知道连续函数的解析表达式，则可以利用拐点的定义寻找出该函数的拐点。根据拐点定义及判别方法可知：如果函数在一个点两侧二阶导数异号，则该点对应曲线上的点即为拐点。现实问题中，往往没有这种已知条件较为充足、理想的情况。例如，如果知道一个函数的某些离散节点的函数值，能否找出函数的拐点。

问题(寻找拐点问题)已知一元函数 $y=y(x)$ 在若干个点的函数值，具体数据如表1所示。请找出函数在 $[0, 12]$ 区间上的所有可能的拐点。

表 函数在若干节点的函数值

k	xk	yk	k	xk	yk	k	xk	yk
1	0	2.4051	21	4	0.6131	41	8	1.6234
2	0.2	2.8759	22	4.2	0.5258	42	8.2	1.4945
3	0.4	3.4072	23	4.4	0.4668	43	8.4	1.3263
4	0.6	3.969	24	4.6	0.4302	44	8.6	1.1387
5	0.8	4.5147	25	4.8	0.4125	45	8.8	0.9503
6	1	4.9844	26	5	0.4114	46	9	0.7756
7	1.2	5.3149	27	5.2	0.4263	47	9.2	0.6233
8	1.4	5.4541	28	5.4	0.4575	48	9.4	0.4971
9	1.6	5.3752	29	5.6	0.5064	49	9.6	0.3965
10	1.8	5.0849	30	5.8	0.5748	50	9.8	0.3188
11	2	4.6224	31	6	0.6648	51	10	0.2605
12	2.2	4.0482	32	6.2	0.7776	52	10.2	0.2178
13	2.4	3.4297	33	6.4	0.9129	53	10.4	0.1875
14	2.6	2.826	34	6.6	1.0671	54	10.6	0.167
15	2.8	2.2793	35	6.8	1.2324	55	10.8	0.1546
16	3	1.8125	36	7	1.3961	56	11	0.149
17	3.2	1.4322	37	7.2	1.5416	57	11.2	0.1496
18	3.4	1.1336	38	7.4	1.6501	58	11.4	0.1565
19	3.6	0.9059	39	7.6	1.7052	59	11.6	0.1701
20	3.8	0.7364	40	7.8	1.6966	60	11.8	0.1913

实验4

- 将课堂上画的三维图变成类似下图的动画，并保存为**gif**图片

