

MATLAB与信号处理

“基于MATLAB的数字信号处理”是配合 “数字信号处理” 等专业基础课，面向研究生开出的实验课，课程不仅包含MATLAB程序设计的内容，而且涉及数字信号处理、数字图像处理和神经网络等内容，重点是掌握用MATLAB实现算法和验证所学理论，应用性强。



课程内容

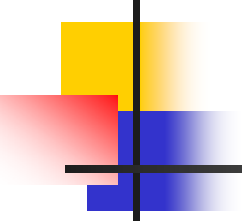
- MATLAB
 - 1、简介
 - 2、程序设计
 - 3、图形功能（数据可视化）
 - 4、界面和调用
- 基于MATLAB的数字信号处理
- 基于MATLAB的语音信号处理
- 基于MATLAB的数字图像处理
- 基于MATLAB的神经网络设计

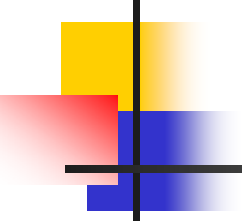


1. MATLAB简介

◆ Mathwork公司

- 科学研究需要大量的数学计算，用传统的编程语言（C/C++，Basic，Fortran等）实现复杂计算时，需熟练地掌握所用语言的语法及编程技巧，对多数科学工作者而言，编制程序非常繁杂，不仅消耗人力与物力，而且影响工作进程和效率。
- 为克服上述困难，美国Mathwork公司于1984年推出了商用的“Matrix Laboratory”（缩写为Matlab）软件包，其内核采用C语言编写，并不断更新和扩充。是一种功能强、效率高便于进行科学和工程计算的交互式软件包。

- 
- ◆ MATLAB 语言是当今国际上科学界最具影响力、也是最有活力的软件。它提供了强大的科学运算、灵活的程序设计流程、高质量的图形可视化与界面设计、便捷的与其他程序和语言接口的功能。MATLAB 语言在各国高校与研究单位起着重大的作用。



在美国和欧洲大学中，九十年代将MATLAB正式列入了电子工程专业研究生和本科生的教学计划，MATLAB是必须掌握的基本工具。

在设计研究单位和工业界，MATLAB也成为工程师们应该掌握的一种工具，是被认作进行高效研究、开发的首选软件工具。



◆ MATLAB的组成

- 开发环境：集成的工作空间
- 语言：
- 图形处理系统：二维、三维图形、图像处理和动画显示、GUI
- 丰富数学库函数：从基本运算，如加法、正弦等，到复杂算法，如矩阵求逆、贝塞尔函数和快速傅里叶变换等。
- 应用程序接口：使Matlab与其他语言进行交互的函数库
- 丰富工具箱(toolbox)：不同应用领域，修改函数和源代码
- Simulink：模型化图形输入与仿真工具，实现系统建模和仿真



◆ MATLAB的特点

- 用户使用方便：稿纸似的编程，无需编译连接；
- 高效方便的矩阵和数组运算；
- 具有丰富的数学功能：

矩阵各种运算。如：正交变换、三角分解、特征值等。

各种特殊函数。如：贝塞尔函数、勒让德函数、伽码函数、贝塔函数、椭圆函数等。

各种数学运算功能。如：数值微分、数值积分、插值、求极值、方程求根、FFT、常微分方程的数值解等。



MATLAB的特点

- 具有若干功能强大的应用工具箱：
Communication、Control system、SIGNAL 等
- 方便的绘图功能：绘图函数很丰富；
二维、三维图形、图形用户界面，图形用户
界面**GUI**制作工具，方便制作用户菜单和控件。
- 具有很好的帮助功能：提供十分详细的帮助文件、
联机查询指令**help** 指令。



◆ MATLAB编程举例

- 矩阵运算

```
>> A = [1 2 0; 2 5 -1; 4 10 -1]
```

```
A =
```

1	2	0
2	5	-1
4	10	-1

```
>> B = A'
```

```
B =
```

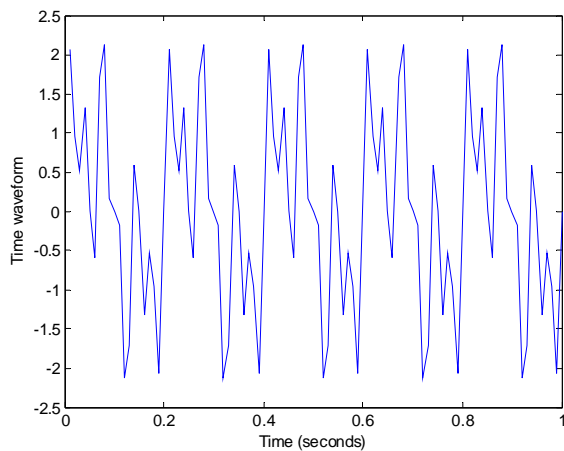
1	2	4
2	5	10
0	-1	-1

```
>> C = A * B
```

```
C =
```

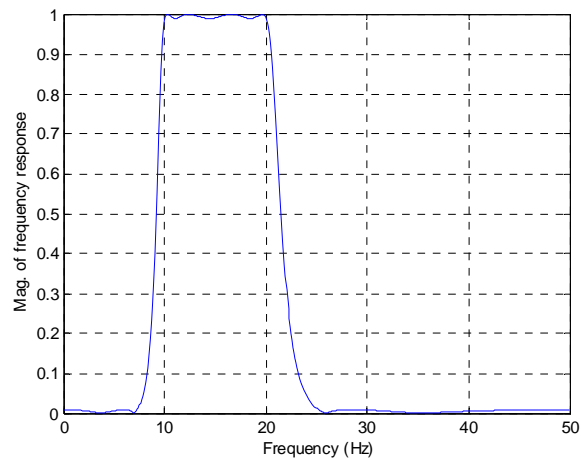
5	12	24
12	30	59
24	59	117

一维信号处理：信号滤波



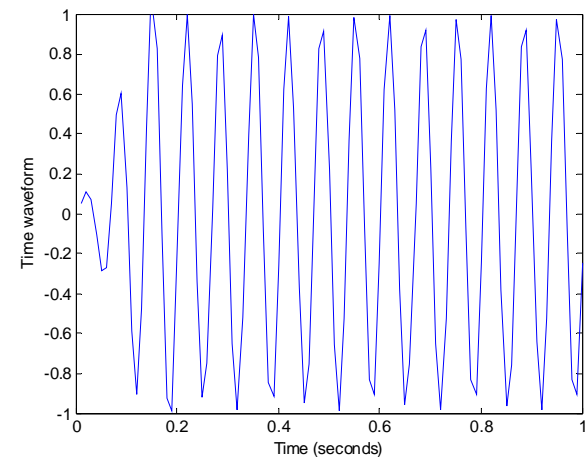
信号:

```
s1 = sin(2*pi*t*5);  
s2=sin(2*pi*t*15);  
s3=sin(2*pi*t*30);  
  
S=s1+s2+s3
```



滤波器:

```
[b,a] = ellip(4,0.1,40,[10 20]*2/Fs);  
[H,w] = freqz(b,a,512);
```

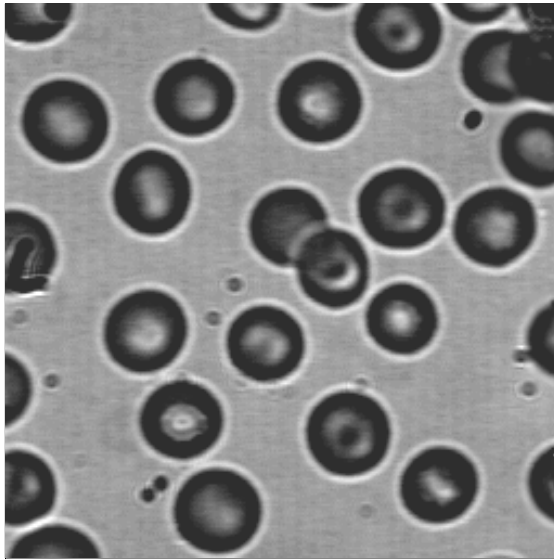


滤波后的结果

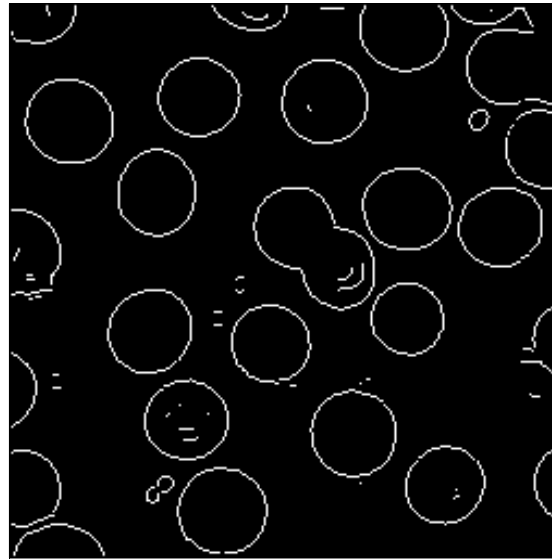
```
sf = filter(b,a,s);
```

• 二维信号处理：图像处理

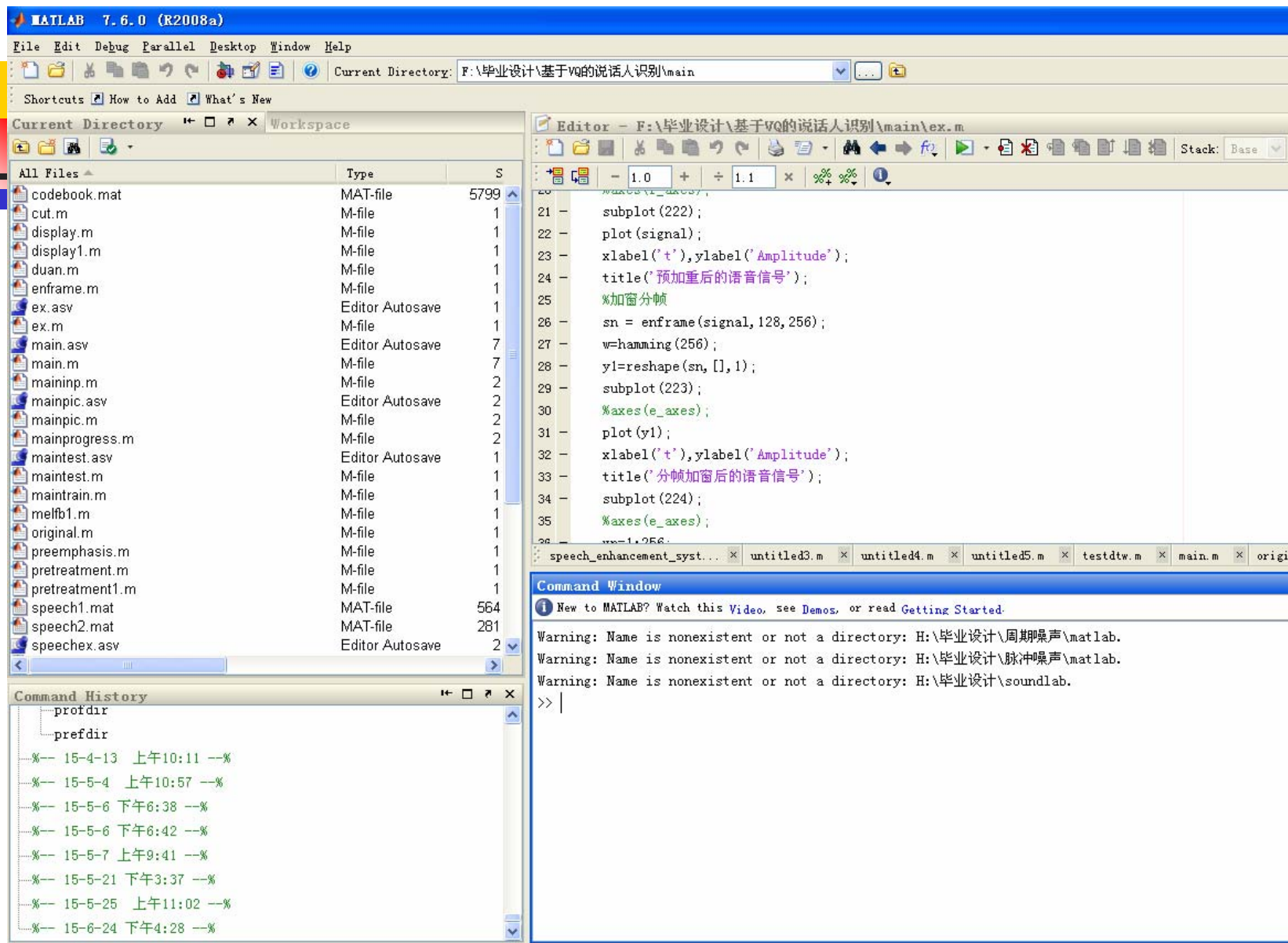
Original Blood Image



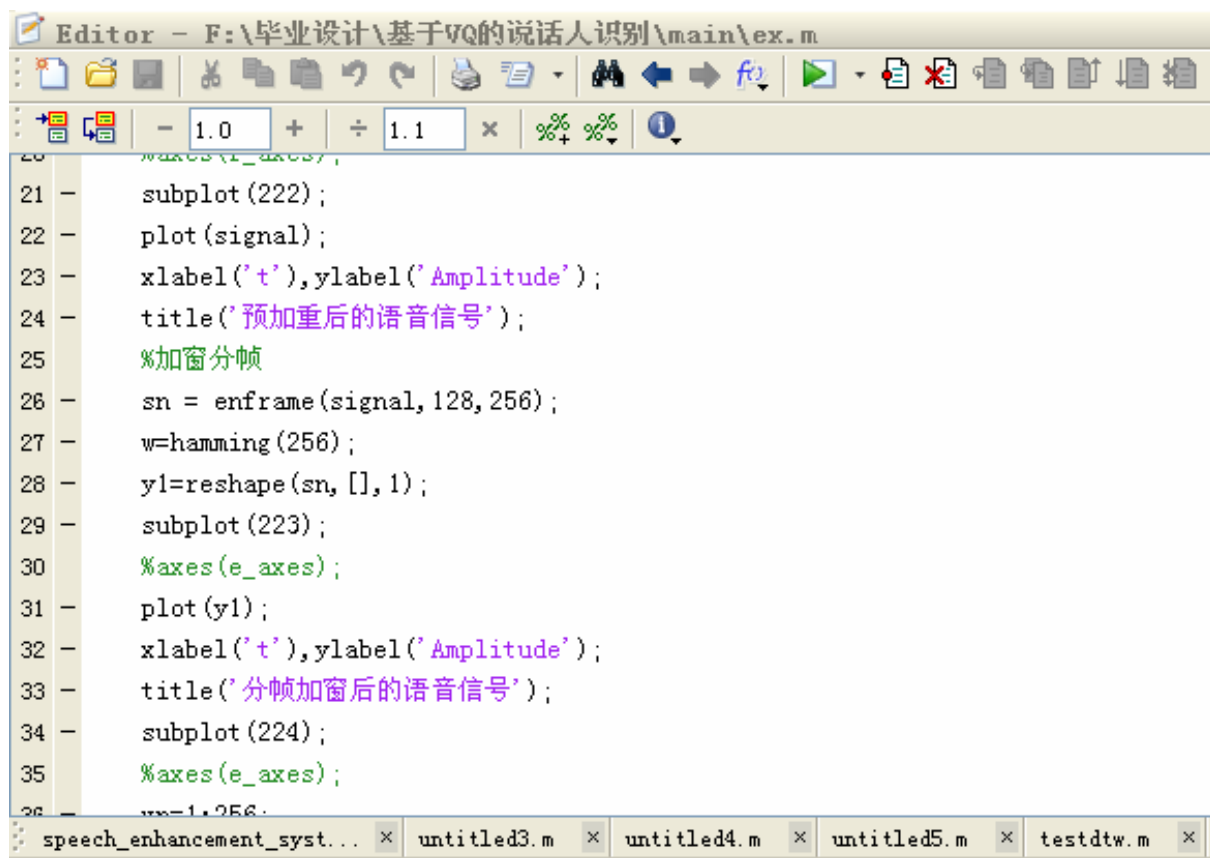
Edge Map



◆ MATLAB的窗口（开发环境）



● 源程序编辑器



```
20 - %axes(e_axes);
21 - subplot(222);
22 - plot(signal);
23 - xlabel('t'),ylabel('Amplitude');
24 - title('预加重后的语音信号');
25 - %加窗分帧
26 - sn = enframe(signal,128,256);
27 - w=hamming(256);
28 - y1=reshape(sn,[],1);
29 - subplot(223);
30 - %axes(e_axes);
31 - plot(y1);
32 - xlabel('t'),ylabel('Amplitude');
33 - title('分帧加窗后的语音信号');
34 - subplot(224);
35 - %axes(e_axes);
36 - %w=1.256;
```

speech_enhancement_syst... x untitled3.m x untitled4.m x untitled5.m x testdtw.m x

➤MATLAB的源程序由ASCII构成，任何文本编辑器都可以用来编写MATLAB源程序（如写字板等）。

➤MATLAB的源程序文件名为*.m，在命令窗中直接输入文件名，并回车，既可执行该文件。

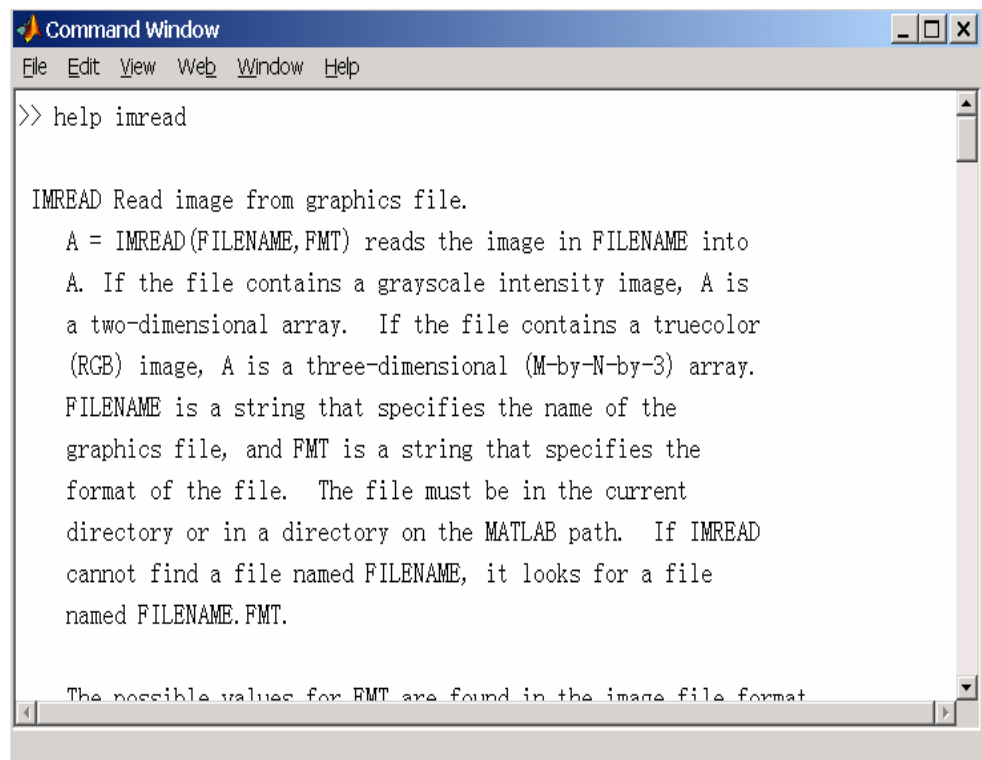
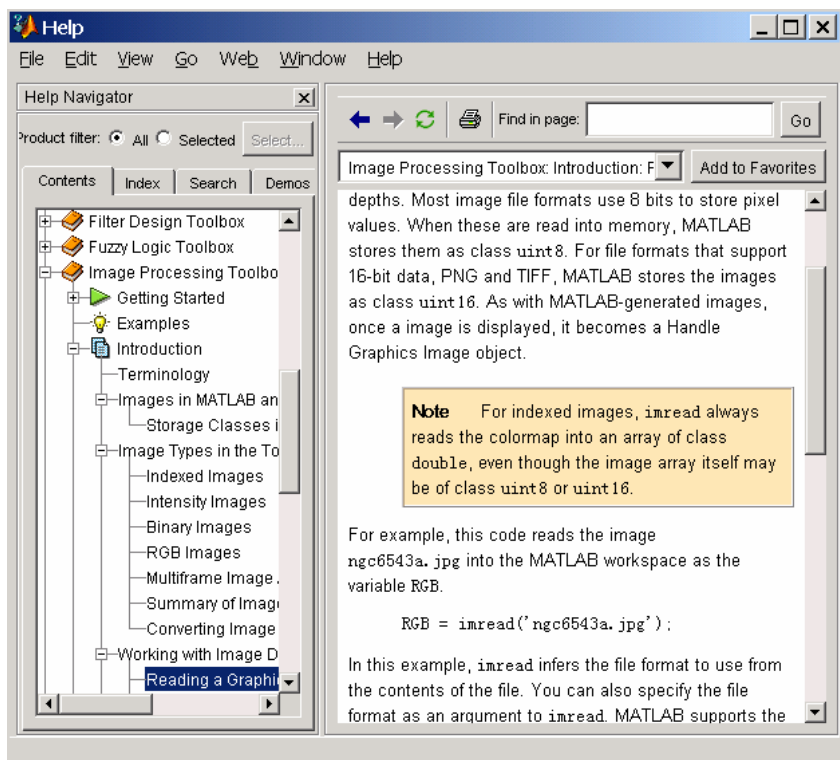
➤源程序编辑器编写、修改、运行程序，设断点调试程序。

➤在调试MATLAB程序时，如果遇到错误信号，命令窗中会给出错误信息，并指出错误在源程序的行号。

• MATLAB的联机帮助

点击Help菜单

Help 命令或函数名





2 MATLAB语言程序设计

- 2.1 数据类型
- 2.2 计算功能
- 2.3 MATLAB语言的控制语句
- 2.4 数据的I/O及保存和装载
- 2.5 M文件与M函数
- 2.6 MATLAB一些使用技巧



2.1 数据类型

5种基本数据类型：

双精度、整型、复数、字符串数组、逻辑类型
元胞数组、构架数组

1) 双精度数据

- MATLAB最常用默认的数据类别是双精度；
- 在MATLAB中变量名区分大小写！

2.1 数据类型

- MATLAB的赋值语句

- 直接赋值语句：变量=表达式或数值

例： $x=1/3$; $V=[1\ 2\ 3]$;

- 函数调用语句：

[返回变量列表]=函数名（输入变量列表）

例： $[x,y,z]=\text{mypro}(a,b,c,d)$;

- MATLAB中的保留常量

- inf——无穷大
- NaN——不定式，如 $0/0$, inf/inf 等
- pi——圆周率

2.1 数据类型

• 矩阵的表示

- 矩阵运算是MATLAB的灵魂！
- MATLAB中矩阵的表示非常容易：

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 0 \end{bmatrix} \longrightarrow A = [1,2,3; 4,5,6; 7,8,0]$$

$$V = [1 \quad 2 \quad 3] \longrightarrow V = [1 \ 2 \ 3]$$

$$V = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \longrightarrow V = [1; 2; 3]$$



2.1 数据类型

- 复杂矩阵的表示

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

$$B = \begin{bmatrix} 5 & 6 & 7 & 8 \\ 9 & 0 & 1 & 2 \end{bmatrix}$$

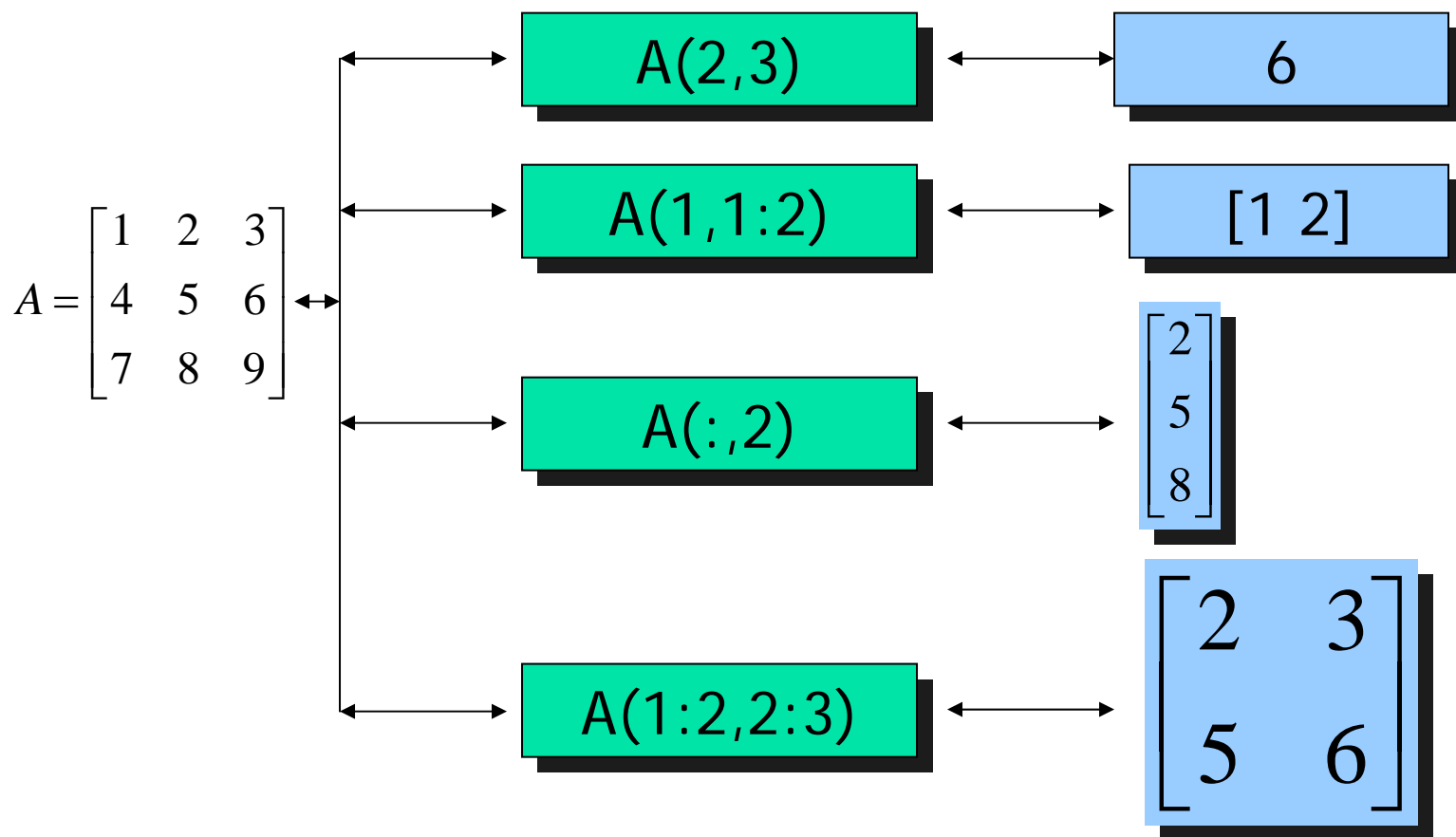
↓

$$C = [A \ B];$$

↓

$$C = \begin{bmatrix} 1 & 2 & 5 & 6 & 7 & 8 \\ 3 & 4 & 9 & 0 & 1 & 2 \end{bmatrix}$$

• 矩阵元素的引用与赋值



2.1 数据类型

2) 整型数据

- 在Matlab7.X及R2016a中，支持8、16、32和64位有符号和无符号整数数据类型。
- 需指定整型类型
int8,16,32,64
uint8,16,32,64
- 有的版本可直接运算、有的需转换为浮点型。如 double(x1)

```
>> x1=int16(20)
x1 =
    20
>> intmin('int16')
ans =
   -32768
>> intmax('int16')
ans =
    32767
>> class(x1)
ans =
int16
>> y1=35
y1 =
    35
>> class(y1)
ans =
double
```

2.1 数据类型

3) 复数

- 在Matlab7.X中，能够直接在复数域上进行运算，不需要任何特殊操作。
- 与数学中的复数表达方式相同， i, j
- 复数的直角坐标形式和极坐标形式的转换
 $\text{real}(z)$, $\text{image}(z)$, $\text{abs}(z)$, $\text{angle}(z)$

```
>> z1=5+4i
z1 =
    5.0000 + 4.0000i
>> z2=3+5*i
z2 =
    3.0000 + 5.0000i
>> z3=2*exp(i*pi/3)
z3 =
    1.0000 + 1.7321i
>> c0=complex(1,2)
c0 =
    1.0000 + 2.0000i
>> c1=sqrt(-3)
c1 =
         0 + 1.7321i
>> c2=9+sin(0.5)i
??? c2=9+sin(0.5)i
Error: Unexpected MA
>> c2=9+sin(0.5)*i
c2 =
    9.0000 + 0.4794i
```

2.1 数据类型

4) 字符串及处理

字符串是用单引号括起来的简单文本

- 赋值: **s='Hello World!'**

字符串中的每个字符都是数组中的一个元素, 以ASCII码存储, char。

```
>> s='Hello World!'
```

```
s =
```

```
Hello World!
```

```
>> size(s)
```

```
ans =
```

```
1    12
```

```
>> |
```



2.1 数据类型

- 截取: `t='This is a character string.'`
`u=t(11:20)` `character`
- 合并: `S=[s1, s2, s3];`
- 字符串与双精度数的转换
`s='MATLAB';`
`double(s)` 77 65 84 76 65 66
`D=77`
`Char(D)` M



2.1 数据类型

5) 单元（元胞）数组（Cell Array）

将相异但相关的数据集成为一个数组（变量）

- 元胞数组的基本组分或元素称为元胞（Cell），以下标来区分。
- 元胞可以存放任何类型、任何大小的数据
- 同一个元胞数组中各元胞的类型可以不同。
- 两种创建方法：
 - 通过赋值语句直接创建；
 - 利用cell创建； `C=cell(2,3)`



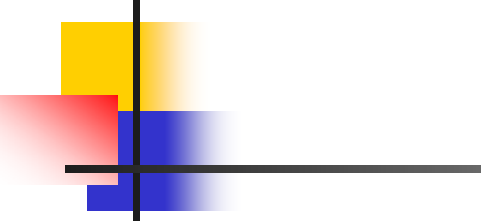
➤ 创建单元数组

```
>> A(1,1)={ [1 2 3;4 5 6;7 8 9] };  
>> A(1,2)={2+3i};  
>> A(2,1)={'A character string'};  
>> A(2,2)={12:-2:0}
```

A =

[3x3 double]	[2.0000+ 3.0000i]
'A character string'	[1x7 double]

单元索引，也可内容寻址（花括号移到左边）



```
>> celldisp(A)
```

```
A{1,1} =
```

```
      1      2      3
      4      5      6
      7      8      9
```

```
A{2,1} =
```

```
A character string
```

```
A{1,2} =
```

```
2.0000 + 3.0000i
```

```
A{2,2} =
```

```
      12      10      8      6      4      2      0
```



➤ 单元数组处理

整体赋值、建更大单元数组、提取子集等

```
C={'x',[1;3;6];10,pi}
```

➤ 获得单元数组内容（用花括号）

```
>> A{1,1}
```

```
ans =
```

1	2	3
4	5	6
7	8	9

```
>> A{2,1}
```

```
ans =
```

```
A character string
```



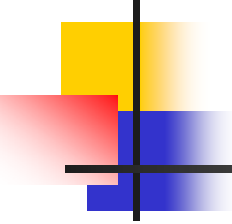
2.1 数据类型

6) 结构（构架）数组（**Structure Array**）

也能存放各类数据。

该数组的基本组分是构架（ **Structure** ），
构架必须在划分“域”后才能使用，
数据不能存放于构架，只能存放在域中，
构架的域可以存放任何类型、任何大小的数组，
不同构架的同名域中存放的内容可不同。

例：创建结构



```
>> circle.radius=2.5;  
>> circle.center=[0,1];  
>> circle.linestyle='--';  
>> circle.color='red'
```

直接赋值

```
circle =
```

```
    radius: 2.5000  
    center: [0 1]  
linestyle: '--'  
    color: 'red'
```

创建函数struct

circle=struct('radius',2.5,'center',[0 1],'linestyle','--','color','red')



2.2 计算功能

1) 基本运算

- 数组运算（点运算）：

. * 乘法, . \ 左除, ./ 右除, . ^ 幂,

A.*B 两个数组中的元素对应相乘



2.2 计算功能

- 矩阵运算

转置:

A' 共轭

加减法:

$A+B$ $A-B$

乘法:

$A*B$

除法:

$A \setminus B$ $AX=B$ 的解

B/A $XA=B$ 的解

乘方:

A^m



2.2 计算功能

例：已知 t 的采样数据是 $(n \times m)$ 维数组。
要计算 $y = e^{-2t} \sin(5t)$ 。

对一般的计算语言来说，必须采用两层循环
MATLAB处理这类问题则简洁快捷得多，它只需直截了当的一条指令

$$\mathbf{y} = \mathbf{exp}(-2*\mathbf{t}).*\mathbf{sin}(5*\mathbf{t})$$

就可获得同样是 $(n \times m)$ 维的 y 数组。

```
t=rand(10,10);  
y=exp(-2*t).*sin(5*t);
```



2.2 计算功能

- 标准数组的产生

`A=eye(100)`

产生100维的单位阵

```
>> eye(3)
```

```
ans =
```

```
1    0    0
0    1    0
0    0    1
```



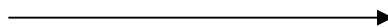
2.2 计算功能

ones(n)
ones(n,m)



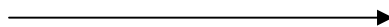
产生全1数组

zeros(n)
zeros(n,m)



产生全0数组

rand(n)
rand(n,m)



产生随机数组



2.2 计算功能

- 矩阵元素的数据变换

取整:

Round(A) 四舍五入取整

截尾:

Fix(A) 把元素按离0近的方向取整



2.2 计算功能

2) 关系运算和逻辑运算

- 关系运算符:

< > <= >= == ~=

- 逻辑运算符:

& | ~

- 如: `A=1:9;`

`tf=A>4`

0 0 0 0 1 1 1 1 1



2.3 MATLAB语言的控制语句

- 循环结构

```
for 循环变量=s1:s2:s3  
    循环体语句组  
end
```

```
while 逻辑变量  
    循环体语句组  
end
```



2.3 MATLAB语言的控制语句

- 条件语句

```
if 逻辑变量  
    条件块语句组  
end
```

```
if 逻辑变量  
    条件块语句组1  
else  
    条件块语句组2  
end
```



2.3 MATLAB语言的控制语句

- 开关结构

switch 开关表达式

case 表达式1

语句段1

case 表达式2

语句段2

otherwise

语句段n

End



2.3 MATLAB语言的控制语句

- 试探式语句结构

```
try  
    语句段1  
catch  
    语言段2  
end
```

首先试探性的执行语句段1，如果发生错误，则执行语句段2。



2.4 数据的I/O及保存和装载

1) 数据的输入/输出

- 输入 (**input**)

```
>> x=input('please input a number:')  
please input a number:33
```

```
x =
```

```
33
```

```
>> x=input('please input a string :','s')  
please input a string :MATLAB
```

```
x =
```

```
MATLAB
```



2.4 数据的I/O及保存和装载

- 输出：disp自由格式，fprintf 格式化

```
>> disp([11 22 33;44 55 66;77 88 99])  
11      22      33  
44      55      66  
77      88      99
```

```
>> fprintf('%8.5f\n',12.5566789)  
12.55668  
.
```



2.4 数据的I/O及保存和装载

2) 保存与装载

- save: 将工作区变量保存在文件中.mat
save filename或save filename variable
- load: 从文件中装载数据
load filename或load filename variable
- 可以:
fopen fclose fread fwrite fscanf fprintf



2.4 数据的I/O及保存和装载

文件打开与读写

打开文件

`Fid=fopen('文件名', 允许级别)`

关闭文件

`Fclose(fid)`

读文件

`A=fscanf(fid, '格式', N)`

`A=fread(fid,num,precision)`

写文件

`fprintf(fid, '格式', 输出变量)`

`fwrite(fid, array,precision)`



2.5 M文件与M函数

- MATLAB程序：M文件（M-script）
M函数（M-function）
- M文件：包含一组MATLAB支持的语句；
直接运行。
- M函数：是MATLAB程序设计的主流；
调用。



2.5 M文件与M函数

- 函数的基本结构

Function [返回变量列表]=函数名(输入变量列表)

%注释语句

函数体语句

End

注意：

文件名最好与函数名一致

```
function [x,y,z]=mytest(a,b)
```

```
% This is a sample function in matlab.
```

```
x=a+b;
```

```
y=a-b;
```

```
z=a*b;
```

```
end
```



2.5 M文件与M函数

- 函数间数据传递

- 1) 输入参数列表和返回参数列表
- 2) 全局变量 `global x`
- 3) 利用.MAT文件



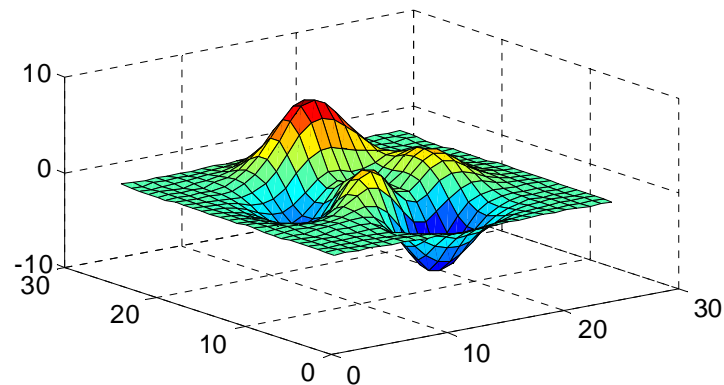
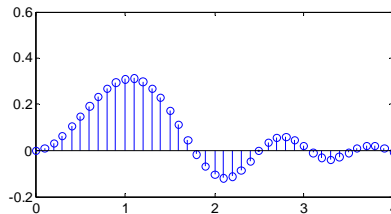
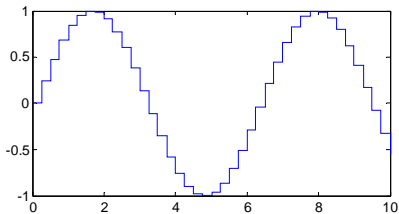
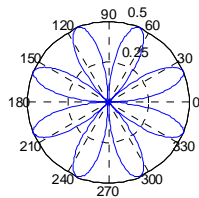
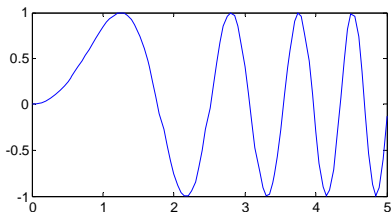
2.6 MATLAB一些使用技巧

加快Matlab函数执行速度的方法

- 尽量避免使用循环：把循环操作变为矩阵操作
- 大型矩阵预先定维
- 优先考虑内在函数
- 采用有效的算法

3 MATLAB图形功能

Matlab能被人们接受的另一个重要原因就是它具有强大的绘图功能





3.1 MATLAB二维图形绘制

- 二维图形绘制

`plot(x,y,'参数')`

X为横坐标，y为纵坐标

`plot(x1,y1,'参数1',
x2,y2,'参数2')`

绘制两条曲线

`plot(y,'参数')`

点数为横坐标，y为纵坐标

`plotyy(x,y1,x,y2)`

把两条曲线绘制在同一窗口内，纵轴两边均有刻度



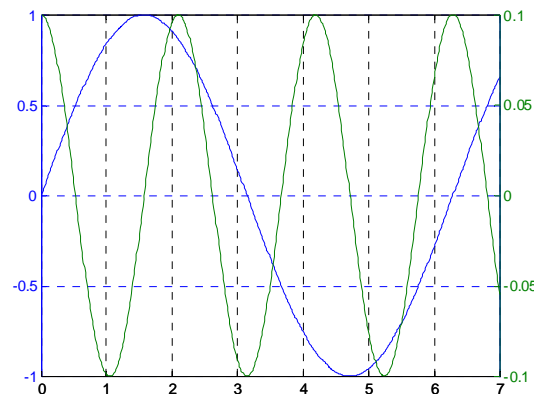
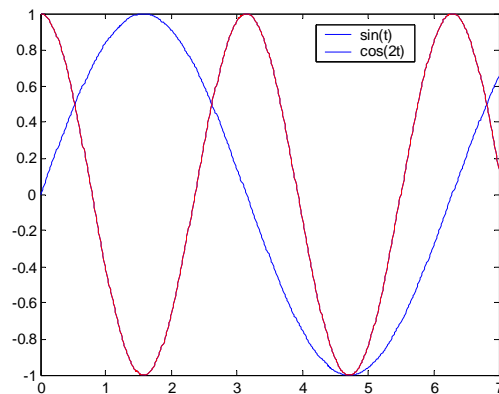
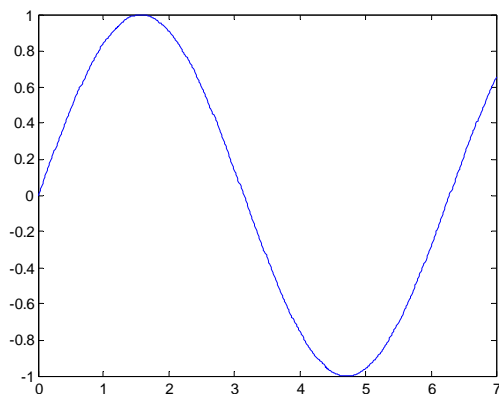
3.1 MATLAB二维图形绘制

- 绘图参数

曲线类型		曲线颜色		标记符号	
-	solid	b	Blue	*	*
:	dotted	r	Red	o	圆
-.	dashdot	g	Green	x	x
--	dashed	y	Yellow	+	+
		k	Black	.	点

3.1 MATLAB二维图形绘制

• 例子



```
t=0:0.01:7;  
y=sin(t);  
Plot(t,y);
```

```
Hold on  
y2=cos(2*t);  
Plot(t,y2,'r');
```

```
t=0:0.01:7;  
y1=sin(t);  
y2=0.1*cos(3*t)  
Plotyy(t,y1,t,y2);  
grid;
```

3.1 MATLAB二维图形绘制

- 二维曲线的标注方法

`xlabel('标注语言')`

标注横坐标

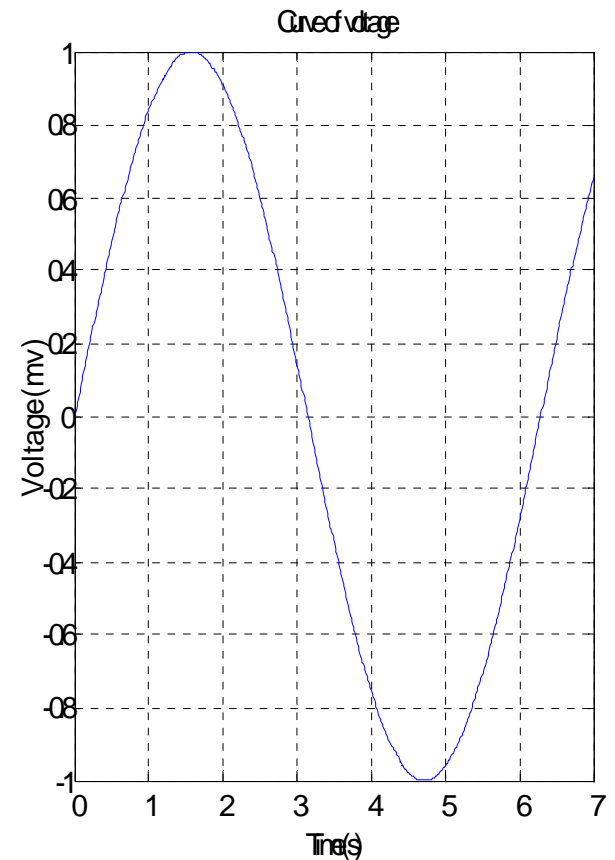
`ylabel('标注语言')`

标注纵坐标

`title('标注语言')`

标注标题

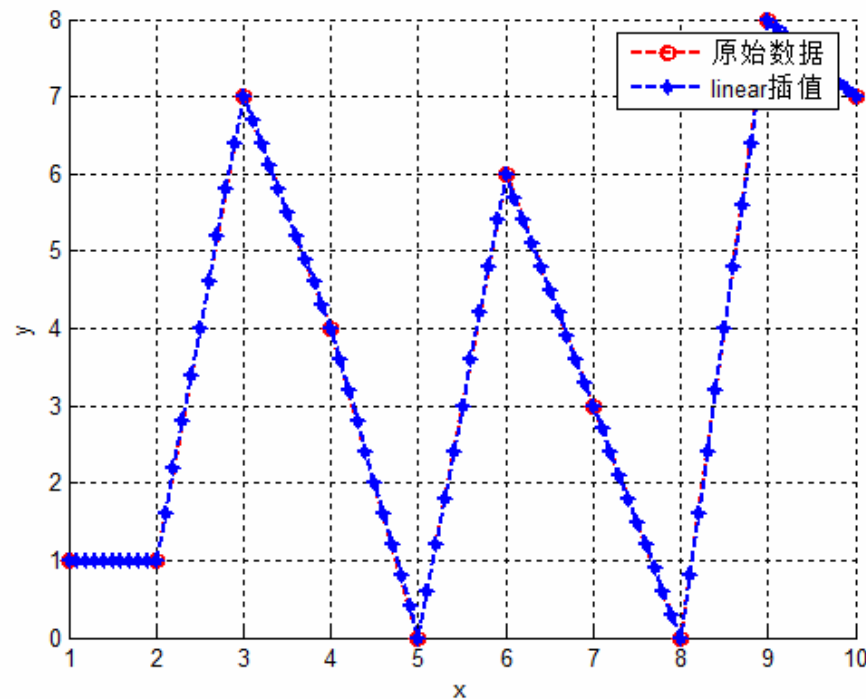
```
xlabel('time(s)');  
Ylable('Voltage(mv)');  
Title('Curve of Vol');
```



3.1 MATLAB二维图形绘制

- 二维曲线的标注方法

```
x = [1:10];  
y = [1, 1, 7, 4, 0, 6, 3, 0, 8, 7];  
figure('color',[1,1,1])  
plot(x,y,'ro--','linewidth',2)  
xi = 1:0.1:10;  
yi = interp1(x,y,xi);  
hold on  
grid on  
plot(xi,yi,'b*--','linewidth',2);  
xlabel('x'); ylabel('y');  
legend('原始数据','linear插值')
```

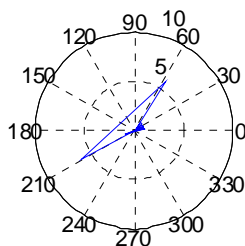
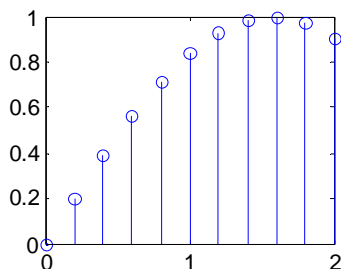
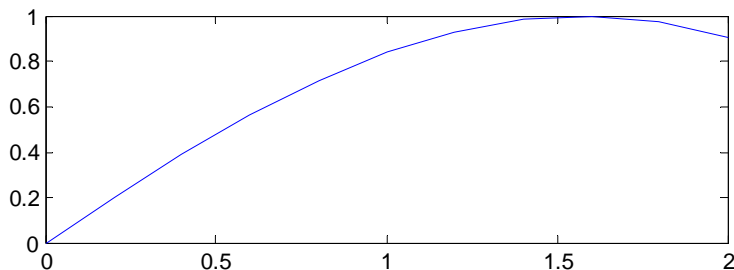


3.1 MATLAB二维图形绘制

• 图形窗口的分割

Subplot(n,m,k)

把图形窗口分为 $n*m$ 个子窗口，当前窗口是第 k 个。



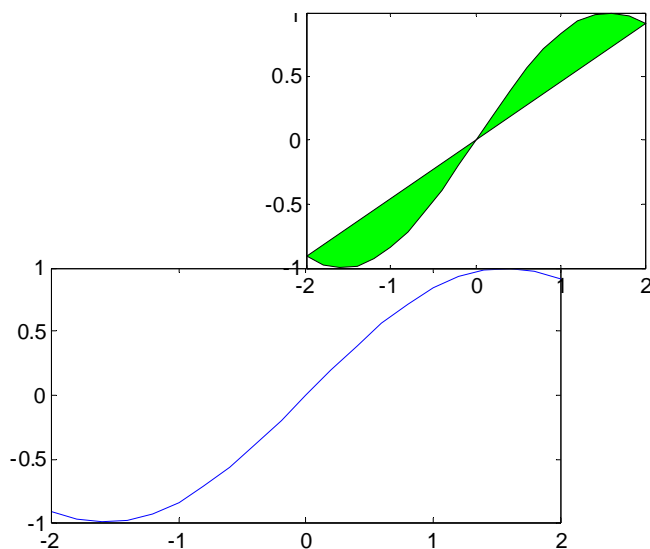
```
x=0:0.2:2;y=sin(x);  
subplot(2,1,1); plot(x,y);  
subplot(2,2,3); stem(x,y);  
subplot(2,2,4); rose(x,y);
```


3.1 MATLAB二维图形绘制

- 任意图形窗口分割

```
h=axes('pos',[x,y,dx,dy])
```

x,y 位置 dx,dy 增量, 为 $(0, 1)$ 之间的数, $(0, 0)$ 表示左下方, $(1, 1)$ 表示右上方。 h 是句柄。

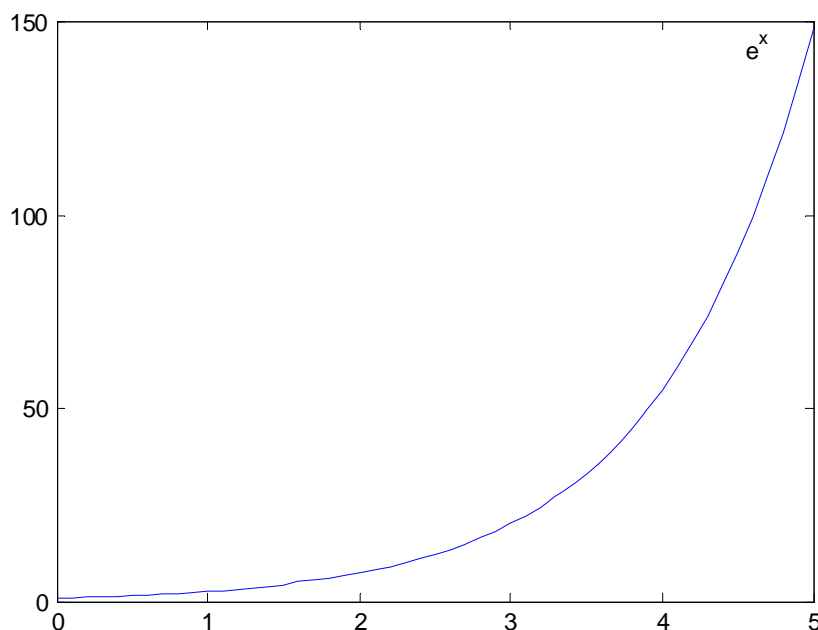


```
x=-2:0.2:2; y=sin(x);  
h1=axes('pos',[0.2, 0.2, 0.6,0.4]);  
plot(x,y);  
h1=axes('pos',[0.5, 0.6, 0.4,0.4]);  
plot(x,y);  
fill(x,y,'g');
```

3.1 MATLAB二维图形绘制

- 给图形上加文字标注

MATLAB中允许用户用给图像上加文字标注



```
t=['e^x'];  
gtext(t);
```

`text(x,y,'字符串')`——在坐标
(x,y)处标注说明文字
`gtext('字符串')`——用鼠标在特
定处标注说明文字

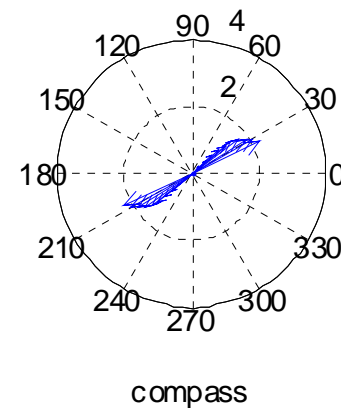
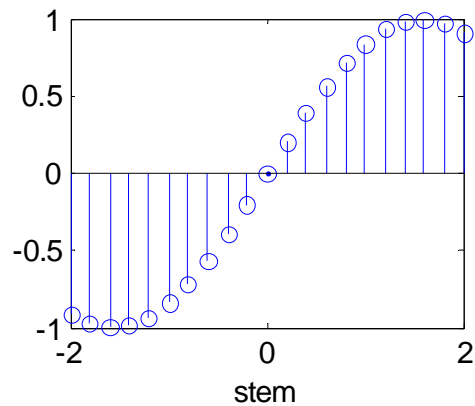
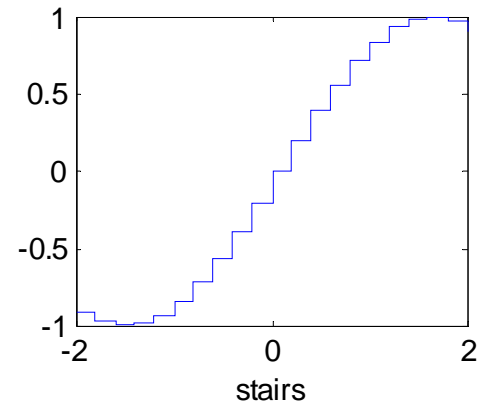
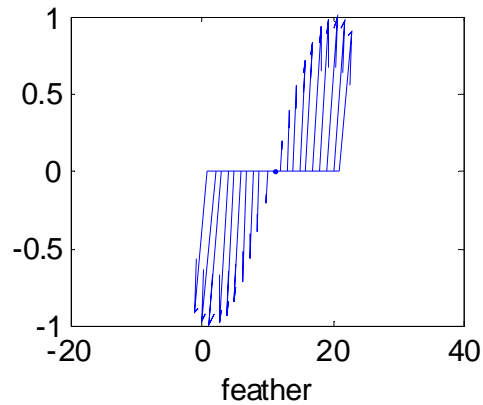
3.1 MATLAB二维图形绘制

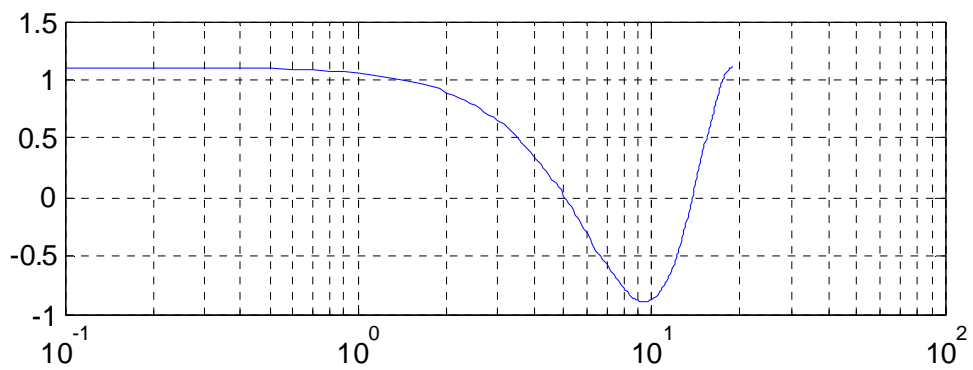
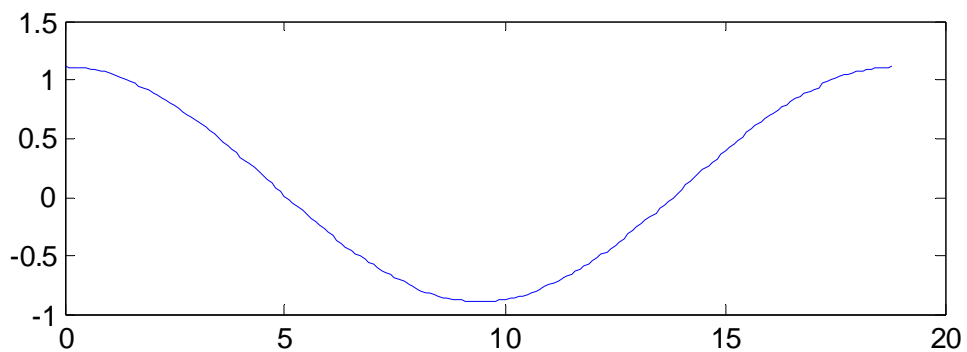
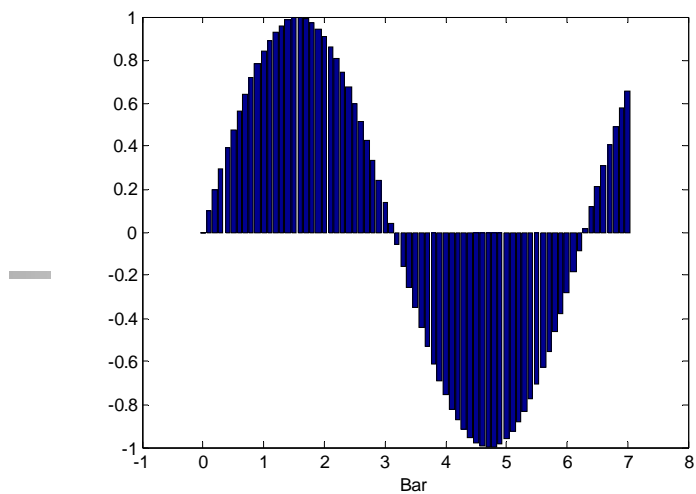
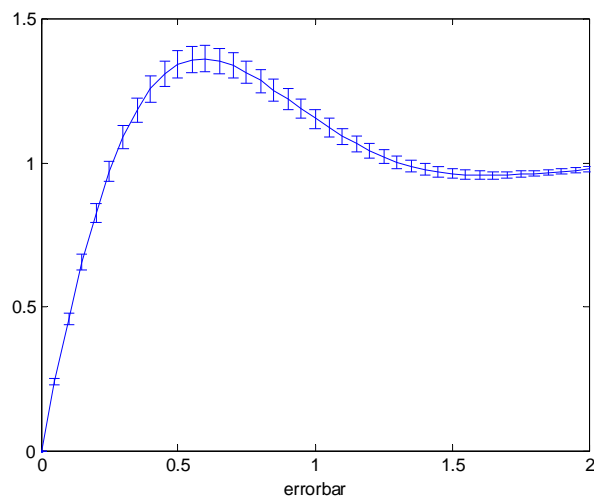
• 特殊图形函数绘制

函数名	意义	函数名	意义
bar(x,y)	二维条形图	fill(x,y,c)	填充图
comet(x,y)	彗星状轨迹图	hist(y,n)	直方图
compass(x,y)	罗盘图	loglog(x,y)	对数图
errorbar(x,y,l,u)	误差限图	polar(x,y)	极坐标图
feather(x,y)	羽毛状图	stairs(x,y)	阶梯图
stem(x,y)	火柴杆图	semilogx(x,y)	半对数图
Rose(x,y)	极坐标累计图		

3.1 MATLAB二维图形绘制

举例

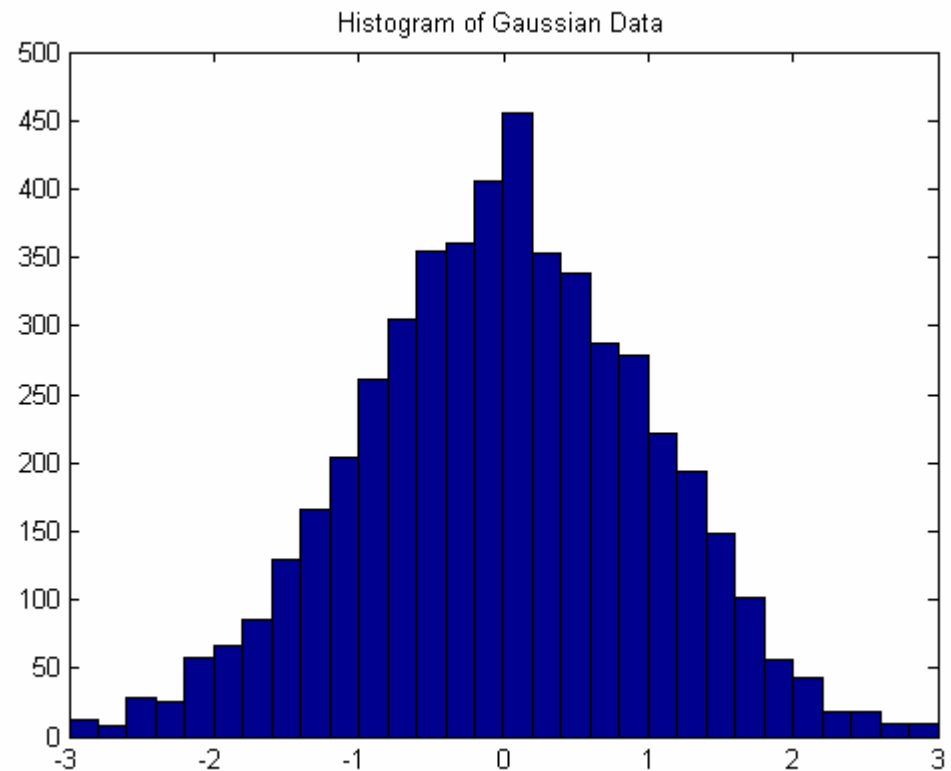




3.1 MATLAB二维图形绘制

➤ 直方图:

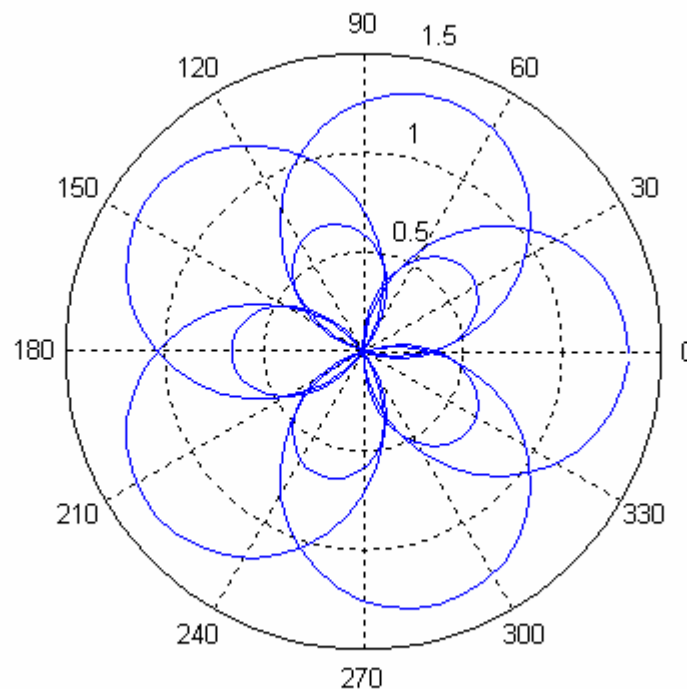
```
x=-2.9:0.2:2.9;  
y=randn(5000,1);  
hist(y,x);  
title('Histogram of  
Gaussian Data')
```



3.1 MATLAB二维图形绘制

- `polar(r,t)`极坐标绘制
R:为幅值 t:为角度

```
t=0:.1:8*pi;  
r=cos(5*t/4)+1/3;  
polar(t,r);
```



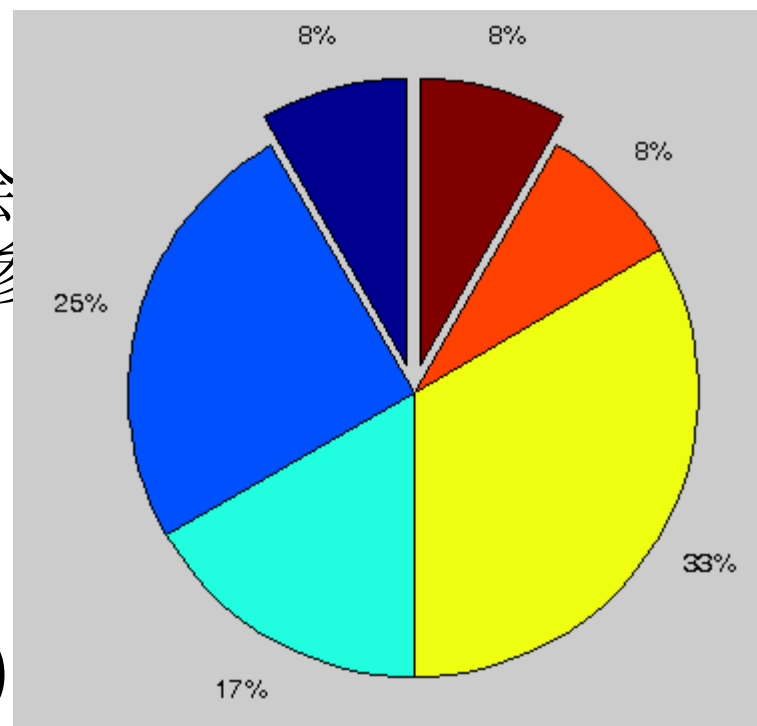
3.1 MATLAB二维图形绘制

➤ 饼图:

`pie(x,explode)`

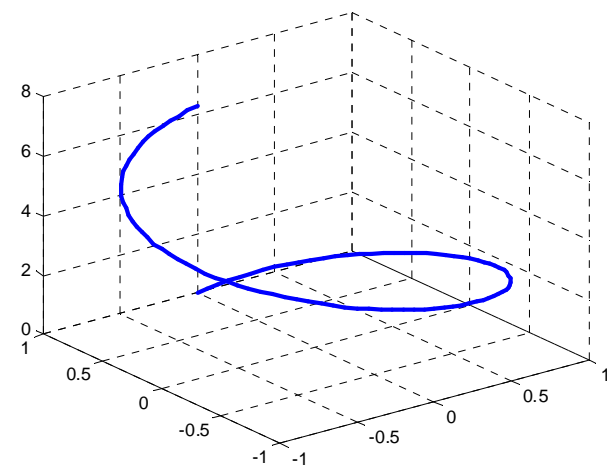
根据矩阵或者向量x绘制占sum(x)的百分比。扇形图是否从整个饼图中分离出来。

```
x=[1 2 1;3 4 1];  
pie(x,[1 0 0;0 0 1])
```

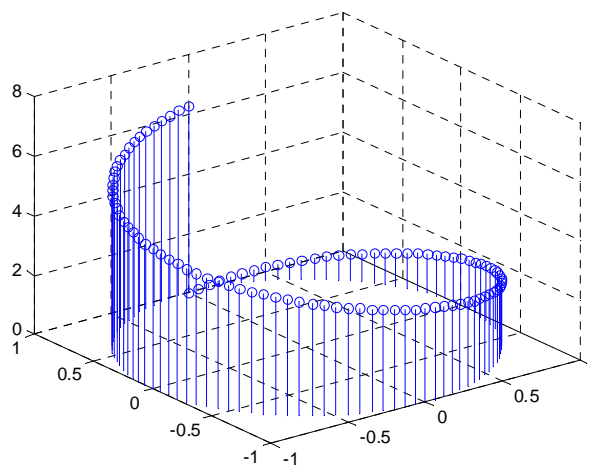


3.2 MATLAB三维图形绘制

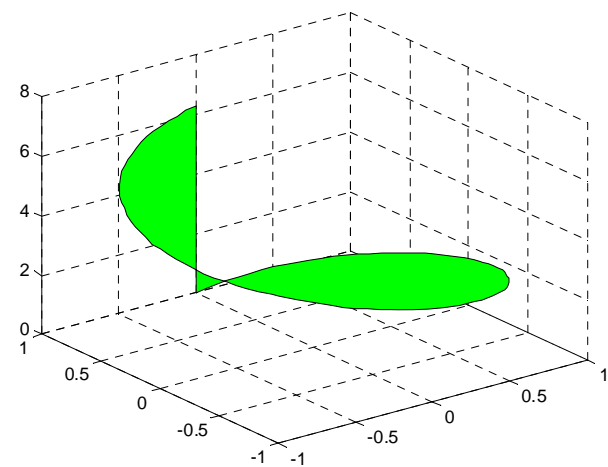
• 基本的三维曲线绘制



`plot3(x,y,z)`



`stem3(x,y,z)`



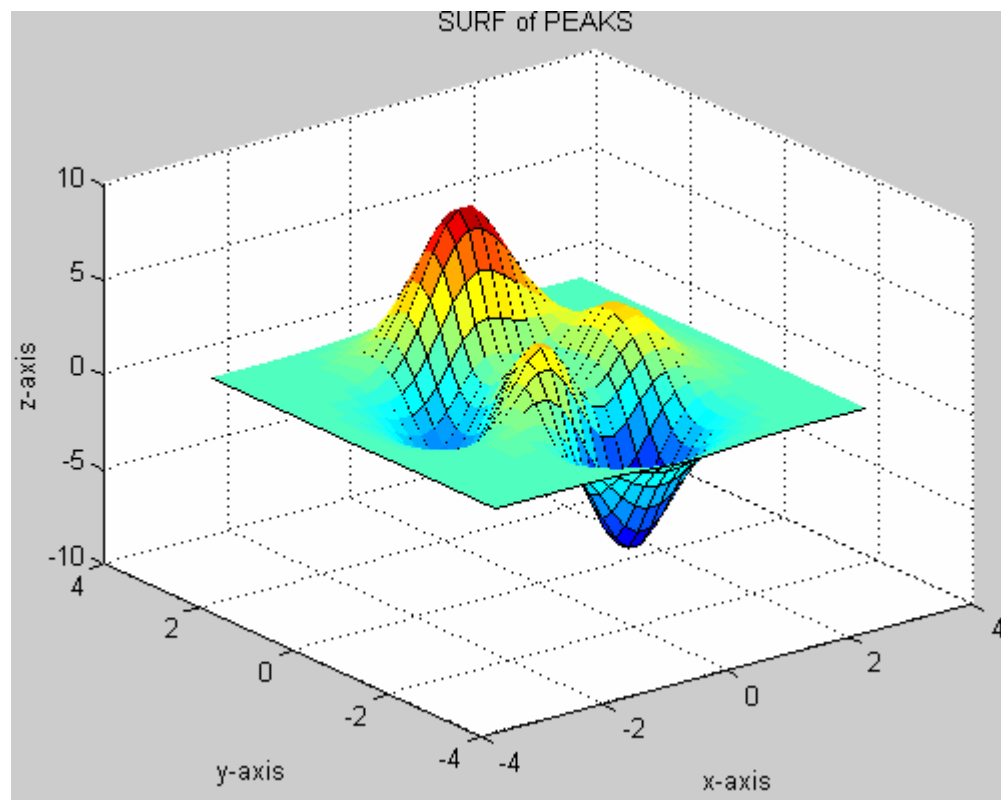
`fill3(x,y,z,'color')`

```
t=0:pi/50:2*pi; x=sin(t); y=cos(t); z=t;
```

3.2 MATLAB三维图形绘制

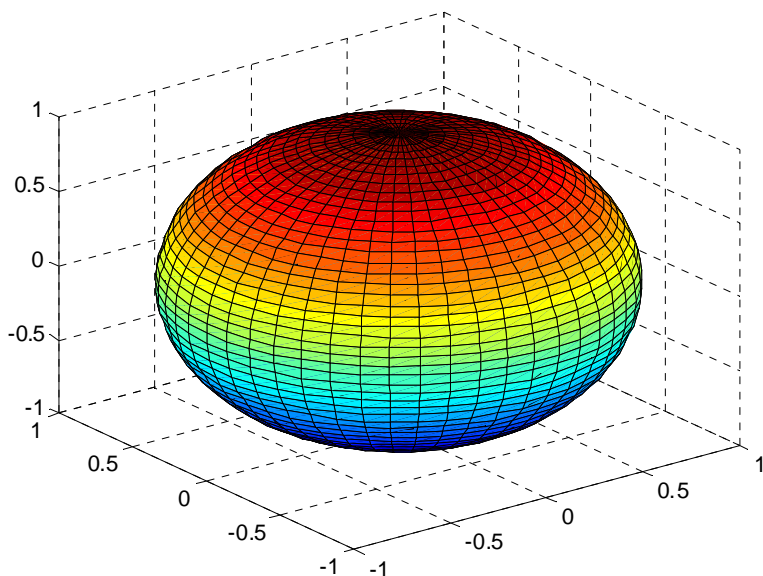
- 三维曲面图

```
[x, y, z]=peaks(30)
surf(x, y, z);
%grid;
xlabel(' x-axis ');
ylabel(' y-axis ');
zlabel(' z-axis ');
title(' SURF of PEAKS ')
end
```

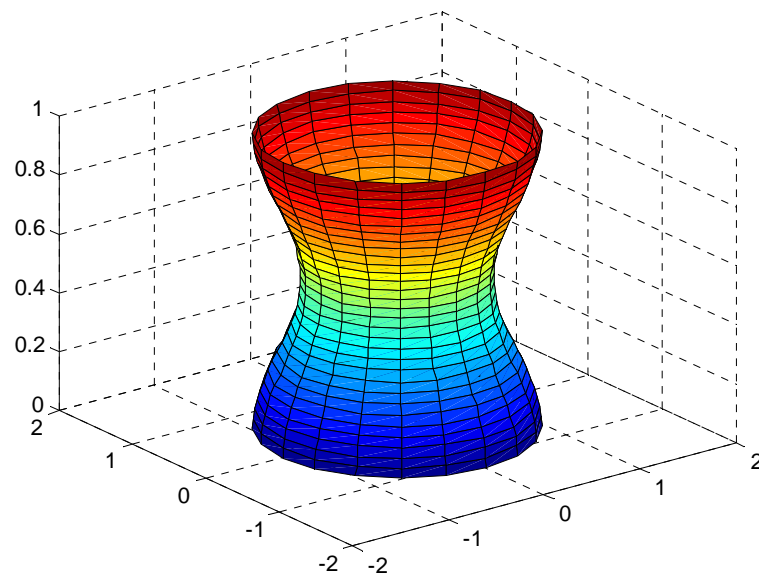


3.2 MATLAB三维图形绘制

- 实体三维曲面绘制



```
[x, y, z]=sphere(n)
```



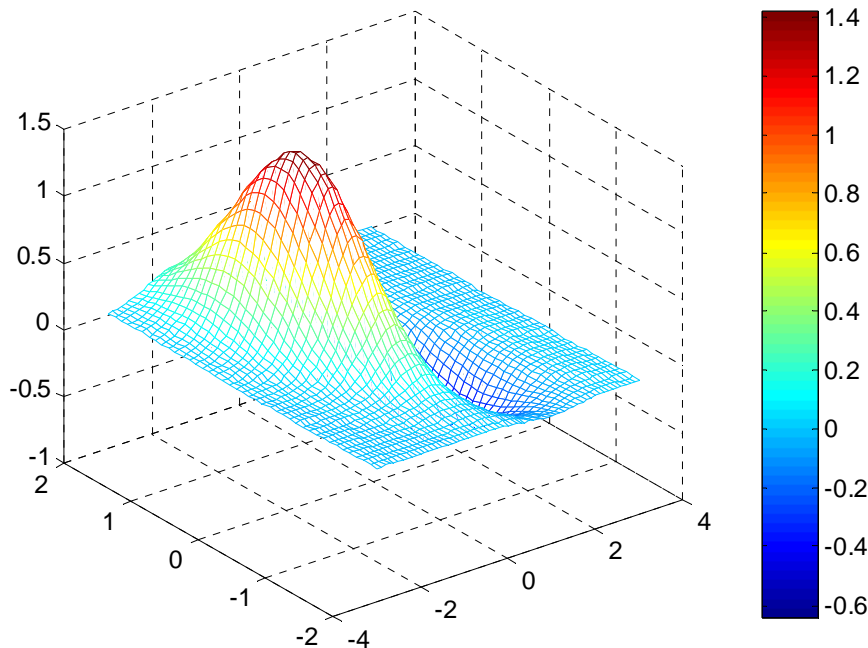
```
[x, y, z]=cylinder(R,n)
```

3.2 MATLAB三维图形绘制

• 三维网格图

`mesh(x, y, z)`

绘制3维表面网格图 $z=f(x,y)$



```
[x,y]=meshgrid(-3:0.1:3, -2:0.1:2);  
z=(x.^2-2*x).*exp(-x.^2-y.^2-x.*y);  
mesh(x,y,z);  
colorbar;
```

$$z = f(x, y) = (x^2 - 2x)e^{-x^2 - y^2 - xy}$$

3.2 MATLAB三维图形绘制

- 三维图形控制
 - 视角
 - 光照
- 动画