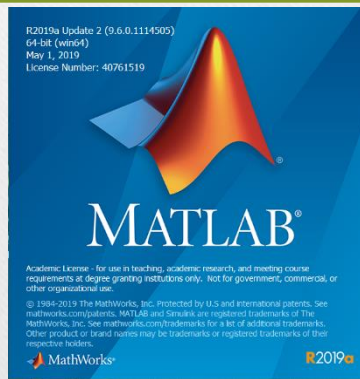


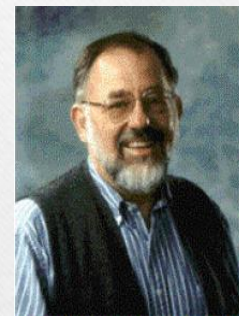
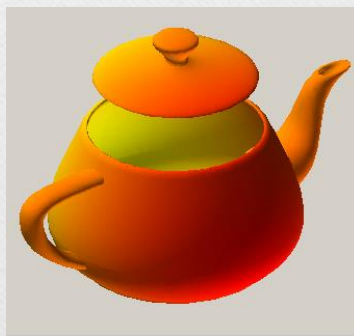
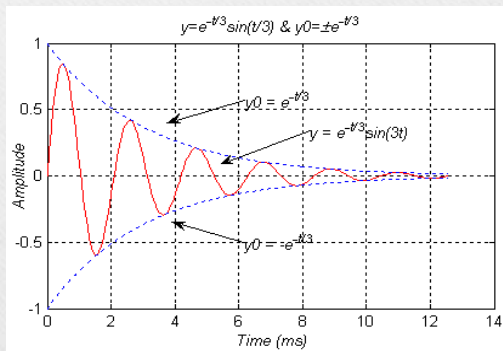
# Matlab简介



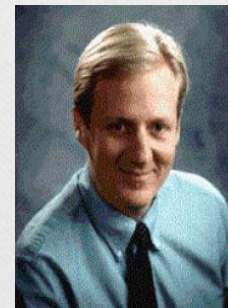
# MATLAB=Matrix Lab

矩阵实验室 美国MathWorks公司出品

- 功能强大
  - 数值运算优势：矩阵运算，数值分析
  - 符号运算优势(Maple)
  - 强大的2D、3D数据可视化功能，图形处理
  - 许多具有算法自适应能力的功能函数



Cleve Moler



Jack Little



# MATLAB的主要组成部分

- **开发环境：**一组工具和组件的集成，图形化用户接口：**MATLAB** 桌面、命令窗口、命令历史窗口、编辑调试窗口及帮助信息、工作空间、Simulink等；
- **MATLAB 数学函数库：**基本函数：求和、正弦、余弦和复数运算等；特殊函数：矩阵求逆、矩阵特征值、贝塞尔函数和快速傅里叶变换等；
- **M (ATLAB) 语言：**一种高级编程语言，包括控制流的描述、函数、数据结构、输入输出及面向对象编程；
- **句柄图形：**定制图形的外观，建立完整的图形界面的应用程序；
- **应用程序接口：**MATLAB的应用程序接口允许用户使用C或FORTRAN语言编写程序与MATLAB

# MATLAB工具箱

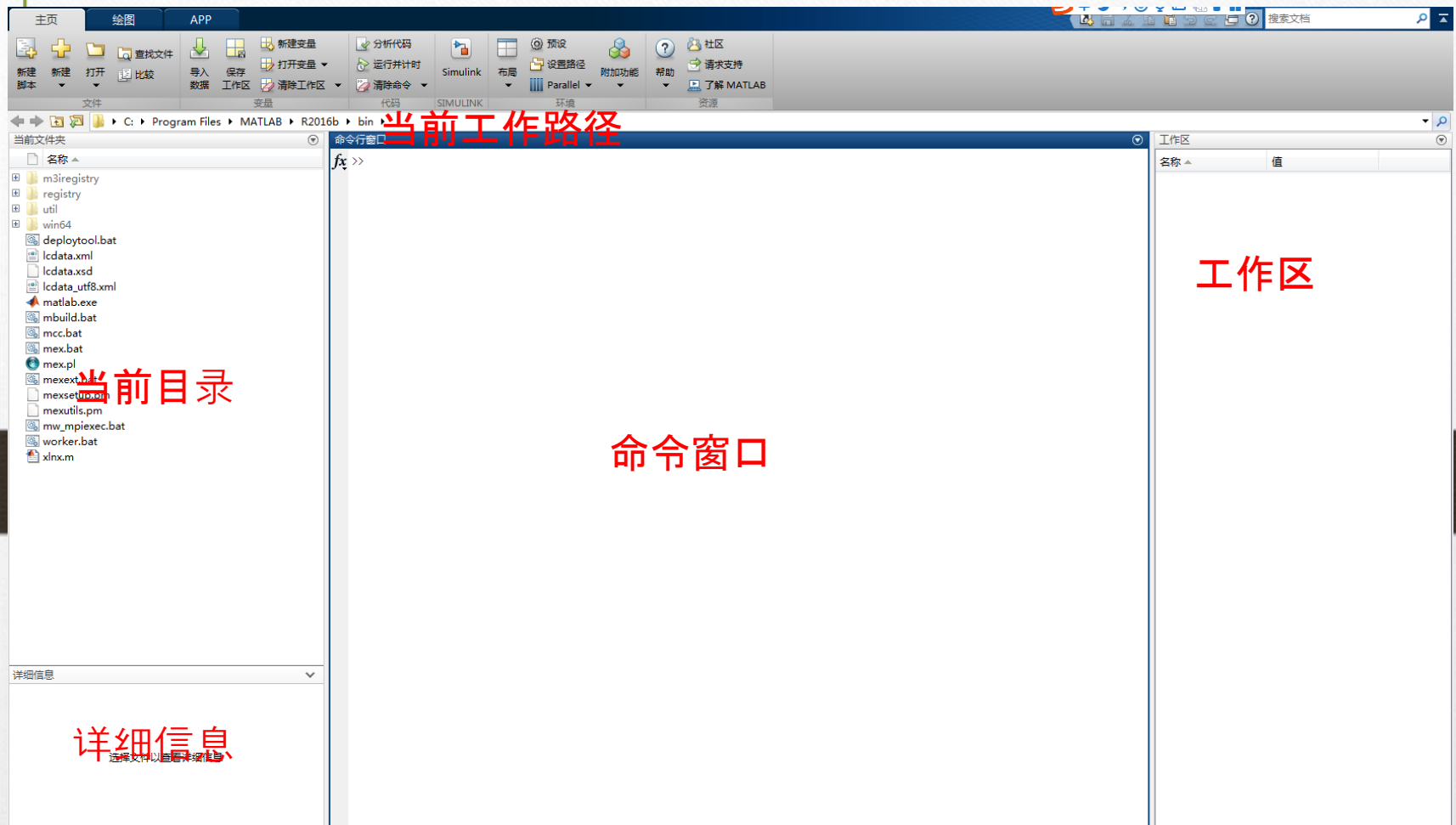
- 控制系统工具箱 (control systems toolbox) ——作者 Alan Laub、John Little
- 信号处理工具箱 (signal processing toolbox) —— John Little、Loren Shure
- 系统辨识工具箱 (system identification toolbox) ——Leonard Ljung
- 鲁棒控制工具箱 (robust control toolbox) ——Richard Chiang、Michael Sofanov
- 分析与综合工具箱 (analysis and synthesis toolbox) ——Richard Chiang、Michael Sofanov
- 神经网络工具箱 (neural network toolbox) ——Howard Demuth、Mark Beale
- 最优化工具箱 (optimization toolbox) ——Peter Fleming、Andrew Grace
- 多变量频域设计工具箱 (multivariable frequency design toolbox) ——Jan Maciejowski

◦ ◦ ◦ ◦ ◦ ◦ ◦ ◦



# MATLAB的系统开发环境

- 操作桌面
  - 桌面布局：5 个窗口
    - 命令窗口 (Command Window )
    - 工作空间窗口 (Workspace )
    - 当前目录浏览器 (Current Directory )
    - 命令历史窗口 ( Command History)
    - 启动平台 (Launch Pad)



Matlab 2016b

# 命令窗口

---

- 用于输入MATLAB 命令、函数、数组、表达式等信息，并显示图形以外的所有计算结果
- `>>A=[1 2 3; 4 5 6; 7 8 9]` ; •不显示结果
- 显示:

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



# 命令窗口

---

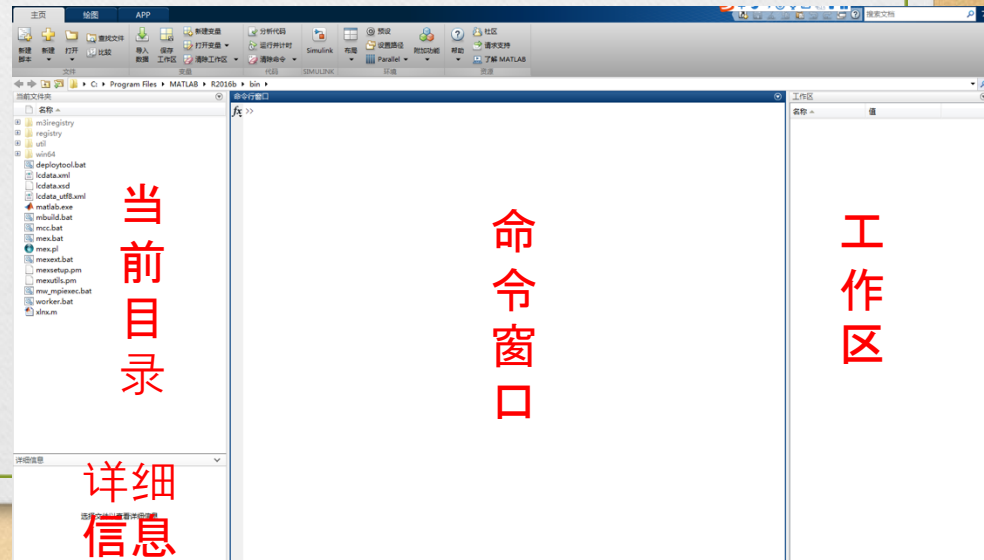
- 高性能计算
- 例：计算 $18 + (5\sin \pi / 6) / (2 + \cos \pi / 6)$
- `>>18+(5*sin(pi/6))/(2+cos(pi/6))`
- 结果显示：

```
ans=  
18.8723
```



# 工作空间窗口

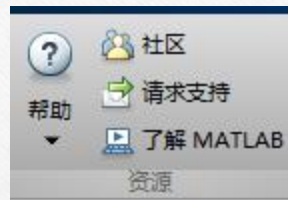
- 用于储存各种变量和结果的空间，显示变量的名称、大小、字节数及数据类型，对变量进行观察、编辑、保存和删除。临时变量不占空间。
- whos （显示存在工作空间全部变量的名称、大小、数据类型等信息）
- who （只显示变量名）



# Matlab帮助

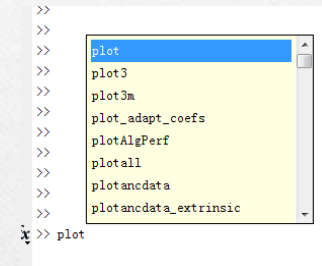
- 提供方便快捷的帮助信息获取途径和图文并茂的帮助内容

- 帮助浏览器



- 模糊查询

输入命令的前几个字母  
然后键入Tab 键



- help 命令 + 函数名

```
>> help exp
exp    Exponential.
exp(X) is the exponential of the elements of X, e to the X.
For complex Z=X+i*Y, exp(Z) = exp(X)*(COS(Y)+i*SIN(Y)).

See also expmi, log, log10, expm, expint.

exp 的参考页
名为 exp 的其他函数
```

fx >>

- lookfor命令+功能

```
>> lookfor integral
dblquad
integral
integral2
integral3
quad
quad2d
quadgk
quadl
triplequad
ellipke
expint
integralBoxFilter
integralBoxFilter3
integralImage
integralImage3
assema
ellipk
- Numerically evaluate double integral over a rectangle.
- Numerically evaluate integral.
- Numerically evaluate double integral.
- Numerically evaluate triple integral.
- Numerically evaluate integral, adaptive Simpson quadrature.
- Numerically evaluate double integral over a planar region.
- Numerically evaluate integral, adaptive Gauss-Kronrod quadrature.
- Numerically evaluate integral, adaptive Lobatto quadrature.
- Numerically evaluate triple integral.
- Complete elliptic integral.
- Exponential integral function.
- 2-D box filtering of integral images
- 3-D box filtering of 3-D integral images
- Compute upright or rotated integral image.
- Compute upright 3-D integral image.
- Assembles area integral contributions in a PDE problem.
- Complete elliptic integral of first kind.
```



# Matlab基本操作

MATLAB以矩阵为基本运算对象，标量可看作 $1 \times 1$ 的矩阵，矢量看作 $n \times 1$ 或 $1 \times n$ 的矩阵

## ● 矩阵的输入方式

- 以直接列出元素的形式输入：把矩阵元素直接排列到方括号[ ]中，每行内的元素用逗号或空格分开，行与行之间用分号隔开

$A=[1,2,3; 4,5,6; 7,8,9]$

- 通过语句和函数产生：矩阵  $x$  已存在，利用  $y=\sin(x)$  产生新矩阵  
语句  $s1:s2:s3$ ;  $\text{zeros}(m,n)$ ;  $\text{eye}(m,n)$   $\text{ones}(m,n)$ ;  $\text{rand}(m,n)$
- 在  $m$  文件中创建矩阵
- 从外部的数据文件中装入： $\text{load}$  或  $\text{fread}$  命令

# 矩阵元素

- 可以用任何形式的表达式来充当矩阵元素

```
>>x=[-1.3 sqrt(3) (1+2+3)*4/5]
```

- 可以用小矩阵构成大矩阵 - 矩阵合并

```
>>a=[1 2 3;4 5 6;7 8 9];
```

```
>>c=[a;[10, 11, 12]]
```

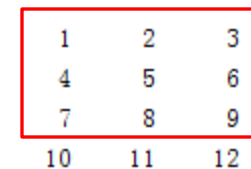
- 可使用 ” : ” 从大矩阵中抽取小矩阵

```
>>d=c(1:3, :) % 意为抽取一到三行的各列
```

- 矩阵转置  $B=A'$

```
>> a=[1 2 3;4 5 6;7 8 9];  
>> c=[a;[10, 11, 12]]
```

c =



1	2	3
4	5	6
7	8	9
10	11	12

```
>> c(1:3, :)
```

ans =

1	2	3
4	5	6
7	8	9



# 语句与变量

## ■ MATLAB 语言的变量名规则:

- 由一个字母引导，后面可以为其他字符;
- 区分大小写.

## ■ MATLAB 语句的常用格式: 变量=表达式 (;) 或简化为: 表达式 (;)

- ① 表达式可以由运算符、特殊字符、函数名、变量名组成，表达式的结果为一矩阵，它赋给左边的变量。
- ② 如省略变量名和 “=”，则自动产生一个名为ans的变量，并显示在屏幕上。
- ③ 如语句以分号结束，则结果不显示。
- ④ 表达式较长时可利用续行符 “...”

```
>> s=1-1/2+1/3-1/4+1/5-1/6+1/7-...  
1/8+1/9-1/10+1/11-1/12
```

```
s =
```

```
0.6532
```

# Matlab永久变量

变量	取值	变量	取值
<b>ans</b>	默认变量名	<b>nargin</b>	函数的输入变量数目
<b>pi</b>	圆周率 ( $\pi=3.1415926\dots$ )	<b>nargout</b>	函数的输出变量数目
<b>i或j</b>	基本虚数单位	<b>realmin</b>	系统所能表示的最小数值
<b>inf或Inf</b>	无限大, 如 $1/0$	<b>realmax</b>	系统所能表示的最大数值
<b>nan或NaN</b>	不定量, 如 $0/0, \infty/\infty, 0*\infty$	<b>lasterr</b>	存放最新的错误信息
<b>eps</b>	浮点相对精度	<b>lastwarn</b>	存放最新的警告信息



# 算数运算符

算术运算符	意义	算术运算符	意义
+	加	\	左除
-	减	/	右除
*	乘	^	幂

- MATLAB 采用10 进制惯例，表示10 的幂次用符号e或E ：

3<sup>-99</sup> 0.001 9.456 1.3e-3 4.5E21

# 复数和复数矩阵

---

复数用特殊字符 $1i$  表示:

$$1i = \text{sqrt}(-1)$$

MATLAB 中复数的语句生成法:

$$z = a + b * 1i \quad \text{或} \quad z = r * \exp(1i * \theta)$$

其中 $r$ 为复数的模,  $\theta$  为复数辐角的弧度数。

例: 复数矩阵的两种输入方法:

```
>>a=[1 2;3 4]+i*[5 6;7 8]
```

或

```
>>a=[1+5i 2+6i; 3+7i 4+8i]
```



# 算术运算

---

## ■ 加、减运算

运算对象必须是同阶矩阵。标量可以和矩阵进行加减运算但应对矩阵的每个元素施加运算

## ■ 乘除运算

矩阵在进行乘除运算时与通常的运算符号相同，

$$A \setminus B = \text{inv}(A) * B, \quad B / A = B * \text{inv}(A)$$

# 算术运算

## ■ 点乘、点除 两个元素大小一致

```
>> x=[1 2 3];  
>> y=[4 5 6];  
>> z=x.*y
```

```
z =  
  
    4    10    18
```

```
>> z=x.\y
```

```
z =  
  
    4.0000    2.5000    2.0000
```

```
>> z=x.^y
```

```
z =  
  
    1    32    729
```

## ■ 乘方运算

$$A^P$$

P 为正整数，A 矩阵自乘P 次；

P 为负整数，A 自乘 P 次，再求逆。

P 是一个分数，例如  $P = m \setminus n$ ，其中n 和m 均为整数，先将A 矩阵自乘n 次，对结果再开m 次方



# 数学函数和矩阵函数

## ■ 数学函数

(a) 基本函数：三角函数、指数函数、复数函数、取整和求余函数。

(b) 特殊函数：特殊数学函数、数理函数、坐标变换

Bessel 函数、beta 函数、gamma 函数、椭圆积分等

## ■ 矩阵函数

矩阵分析、线性方程组、特征值和特征矢量、矩阵函数、因式分解等

```
>> a=[1 4; 9 16];  
>> r1=sqrt(a)
```

```
r1 =
```

```
1    2  
3    4
```

```
>> r2=sqrtm(a) | %r2*r2=a
```

```
r2 =
```

```
0.4662 + 0.9359i    0.8860 - 0.2189i  
1.9935 - 0.4924i    3.7888 + 0.1152i
```

# 常用的矩阵运算

- 矩阵的行列式 `det(A)`
- 矩阵的迹 `trace(A)`
- 矩阵的秩 `rank(A)`
- 矩阵的范数 `norm(A)`
- 矩阵的特征多项式、特征方程与特征根  
`p=poly(A), v=roots(p)`
- 多项式的求值 `C=polyval(a, x)`
- 矩阵的特征值 `[V, D]=eig(A)`
- 矩阵的条件数 `cond(A)`

- 多项式：向量表示多项式系数

$$p(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n$$

$$P = [a_0, a_1, a_2, \dots, a_n];$$



# 举例

例：求  $A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$  的特征多项式、特征根

```
>>A=[1 2 3; 4 5 6; 7 8 9];
```

```
>>p=poly(A)
```

```
>>r=roots(p)'
```

```
>> A=[1 2 3; 4 5 6; 7 8 9];  
>> p=poly(A)  
  
p =  
  
    1.0000   -15.0000   -18.0000    -0.0000  
  
>> r=roots(p)'  
  
r =  
  
   16.1168   -1.1168   -0.0000
```

# 关系运算

## 关系运算

条件：对于两个矩阵的关系运算，**两边的矩阵必须具有同样尺寸。**

关系运算符：< 小于、<= 小于等于、> 大于、>= 大于等于、  
== 等于、~= 不等于。

结果“1”表示“真”，“0”表示“假”。

```
>> a=[0 -1 2];  
>> b=[-3 1 2];  
>> a<b  
  
ans =  
  
1×3 logical 数组  
  
0 1 0
```



# 逻辑运算

逻辑运算符:

& 与 (AND )

```
>> a=[1 2 3; 4 5 6];  
>> b=[-1 0 0; 0 0.5 0];  
>> a&b
```

ans =

2×3 logical 数组

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

| 或 (OR )

```
>> a|b
```

ans =

2×3 logical 数组

```
1 1 1  
1 1 1
```

~ 非 (NOT )

```
>> ~b
```

ans =

2×3 logical 数组

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

# 图形处理功能

## ■ 二维图形

基本绘图函数: Plot, semilogx, semilogy, loglog, polar, plotyy

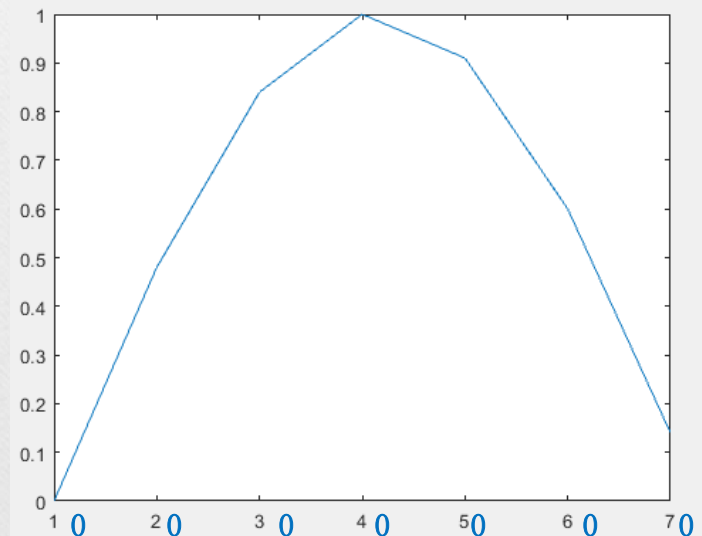
### (1) 单矢量绘图: plot(y)

矢量y的元素与y元素下标之间在线性坐标下的关系曲线。

```
>>y=[0 0.48 0.84 1 0.91 0.6 0.14];  
>>plot(y)
```

```
>>x=10:10:70;  
y=[0 0.48 0.84 1 0.91 0.6 0.14];  
>>plot(x, y)
```

注: 图形加网格和标注可以方便地在图形窗口中完成

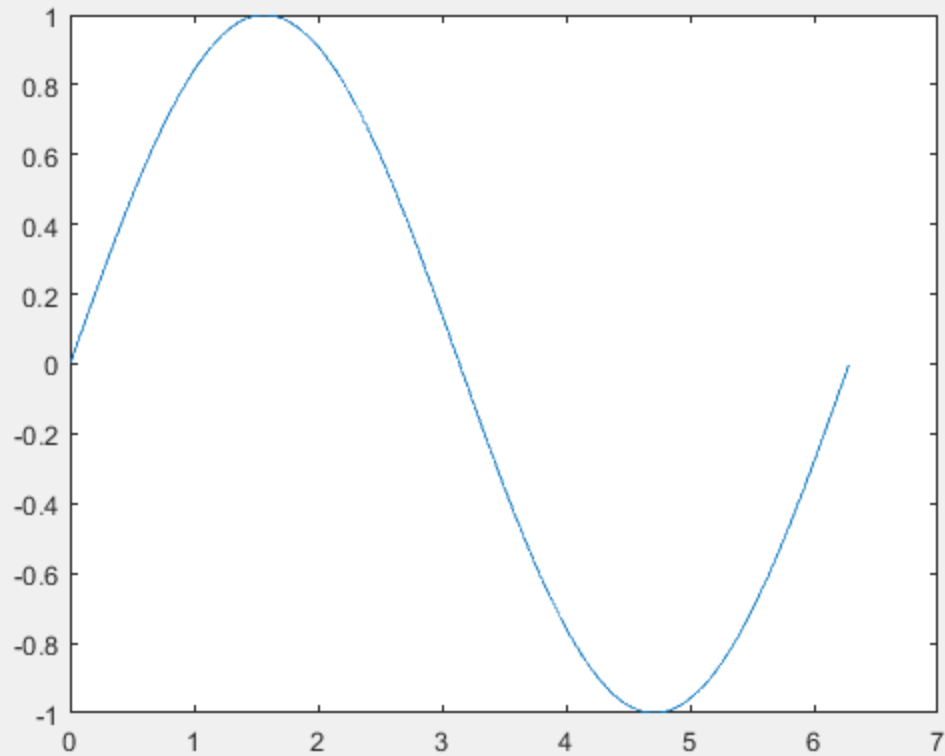




# 图形处理功能

(2) 双矢量绘图 `plot(x, y)`

```
>>x=0:0.01:2*pi;  
>>y=sin(x);  
>>plot(x, y)
```

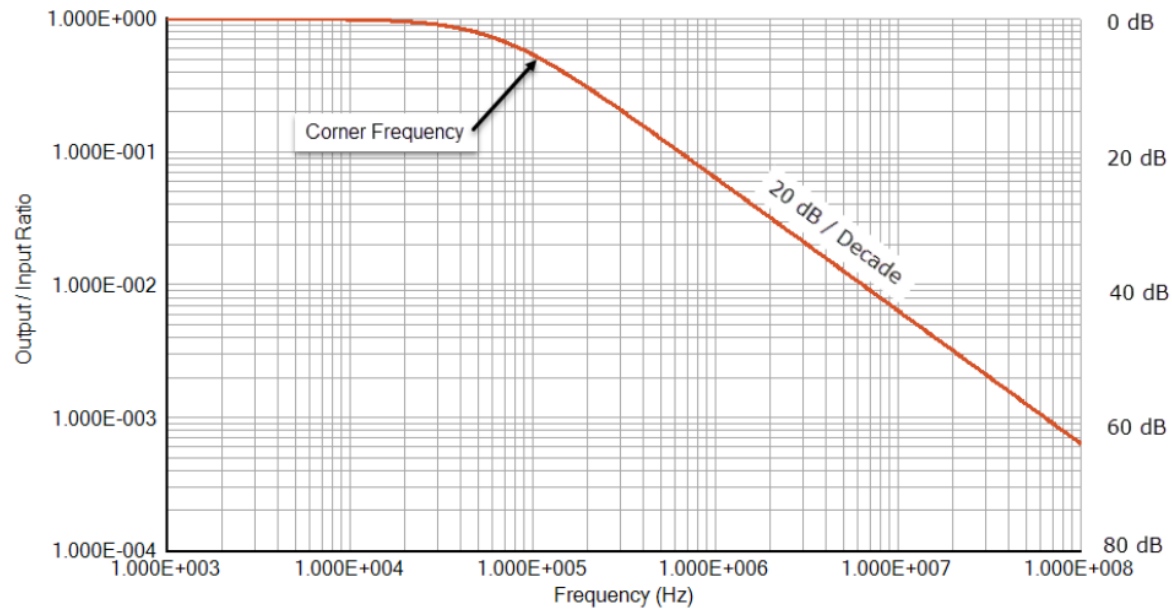


# 图形处理功能

## ■ 对数坐标

Bode图

[bode\(sys\)](#)



Typical Bode Plot

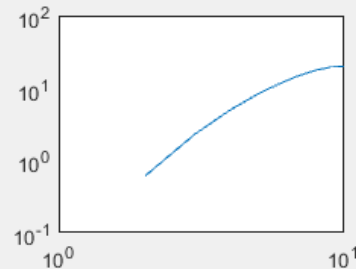
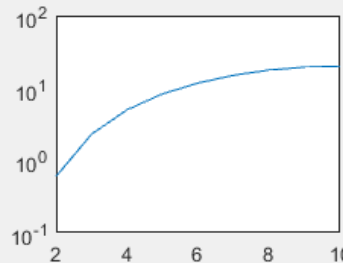
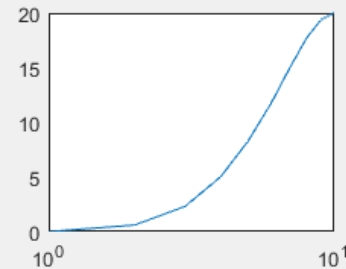
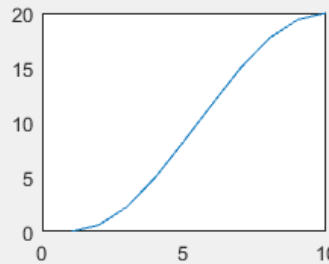


# 图形处理功能

## (3) 对数坐标绘图

x轴对数semilogx, y轴对数semilogy, 双对数loglog,

```
>>y=[0 0.6 2.3 5 8.3 11.7 15 17.7 19.4 20];  
>>subplot(2,2,1)  
>>plot(y)  
>>subplot(2,2,2)  
>>semilogx(y)  
>>subplot(2,2,3)  
>>semilogy(y)  
>>subplot(2,2,4)  
>>loglog(y)
```



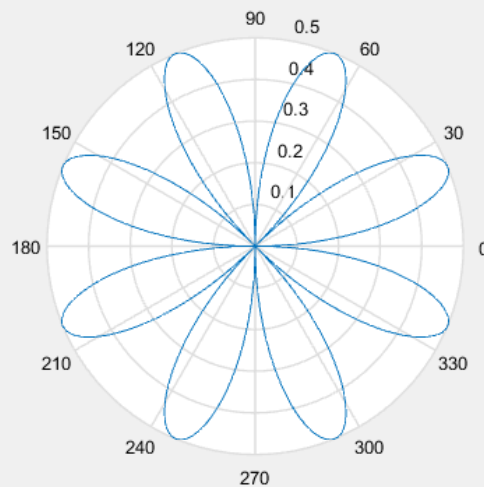
# 图形处理功能

## (4) 极坐标绘图

`polar(theta, rho)` theta——角度, rho——半径

例: 建立简单的极坐标图形。

```
>>t=0:.01:2*pi;  
>>polar(t, sin(2*t).*cos(2*t))
```





# 多重曲线绘图

一组变量绘图  $\text{plot}(x, y)$

- ①  $x$  为矢量,  $y$  为矩阵时  $\text{plot}(x, y)$  用不同的颜色绘制  $y$  矩阵中各行或列对应于  $x$  的曲线。
- ②  $x$  为矩阵,  $y$  为矢量时绘图规则与①的类似
- ③  $x$  和  $y$  是同样大小的矩阵时,  $\text{plot}(x, y)$  绘制  $y$  矩阵中各列对应于  $x$  各列的图形。
- ④ 如果  $y$  是矩阵, 则  $\text{plot}(y)$  绘出  $y$  中各列相对于行号的图形, 对于  $n$  行矩阵,  $x$  轴的坐标为  $[1:n]$ 。

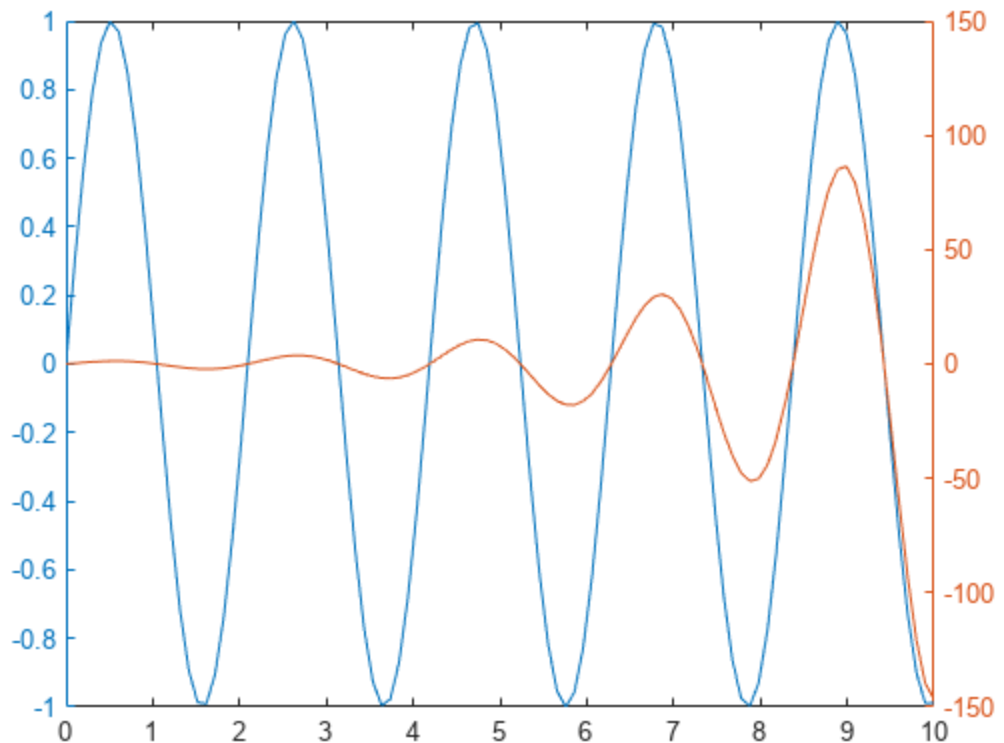
# 多重曲线绘图

## (2) 多组变量绘图

```
plot(x1,y1,x2,y2)
>>x=0:pi/50:2*pi;
>>y1=sin(x);
>>y2=0.6*sin(x);
>>y3=0.3*sin(x);
>>plot(x,y1,x,y2,x,y3)
```

## (3) 双y轴绘图

```
plotyy is no
```





# 图线形式和颜色

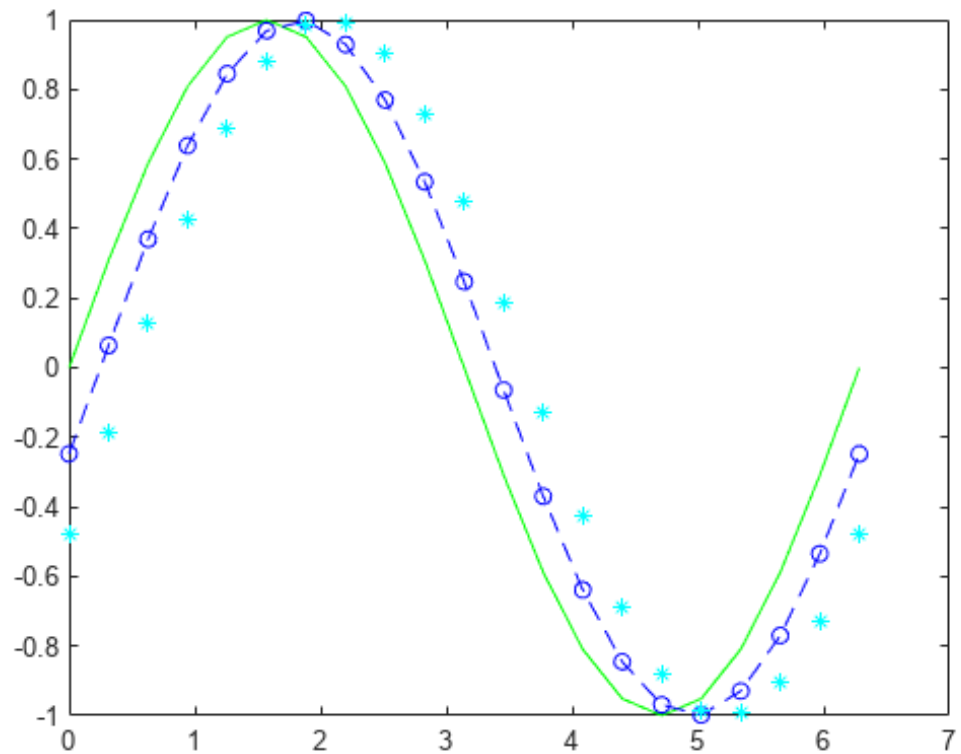
图线的形式：MATLAB 提供的四种线形，

实线— 虚线—

标记点类型： . , +  
d ( 或  
pentag

图线的颜色： MATLAB  
红r , 绿

图线的其他属性： 可  
色、标记点的大小等



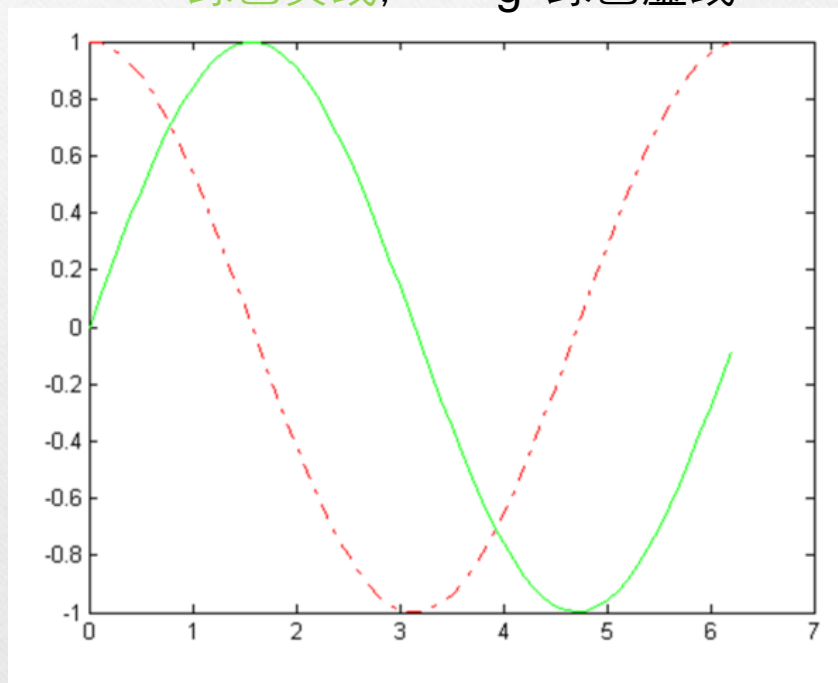
# 图线形式和颜色

例: `>>x=0:0.1:2*pi;`

`>>plot(x, sin(x), ' -g' , x, cos(x), ' -.r' )`

绿色实线, '--g' 绿色虚线

红色点划线





# 图形的控制与表现

---

axis: 人工选择坐标轴尺寸.

clf : 清图形窗口.

ginput: 利用鼠标的十字准线输入.

hold: 保持图形. 分次画多个图形, 不覆盖

shg : 显示图形窗口.

subplot: 将图形窗口分成N 块子窗口。

figure(n): 函数用于为当前的绘图创建图形窗口

subplot(m, n, p): 把窗口分成 $m \times n$  个小窗口, 并把第p个窗口当作当前窗口。

text(x, y, ‘字符串’) 图中加注文本

# 特殊图形

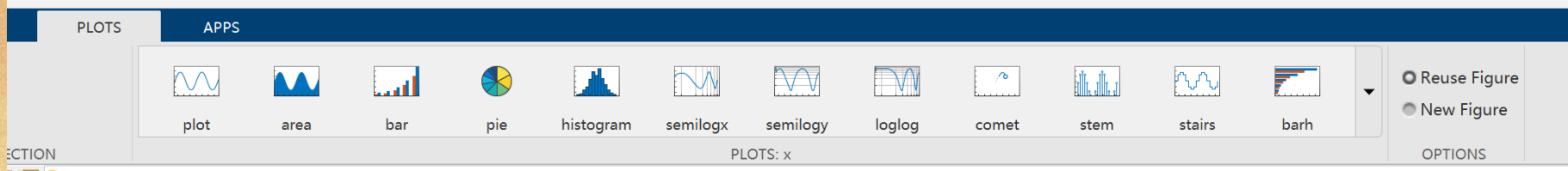
条形图: `bar(y)`, `bar(x,y)`, `barh`, `bar3(y)`, `bar3(x,y)`, `bar3h(x,y)`

饼图: 函数 `pie`, `pie3`

直方图: `hist`

杆状图: `stem(x,y)`

阶梯图: `stairs(x,y)`



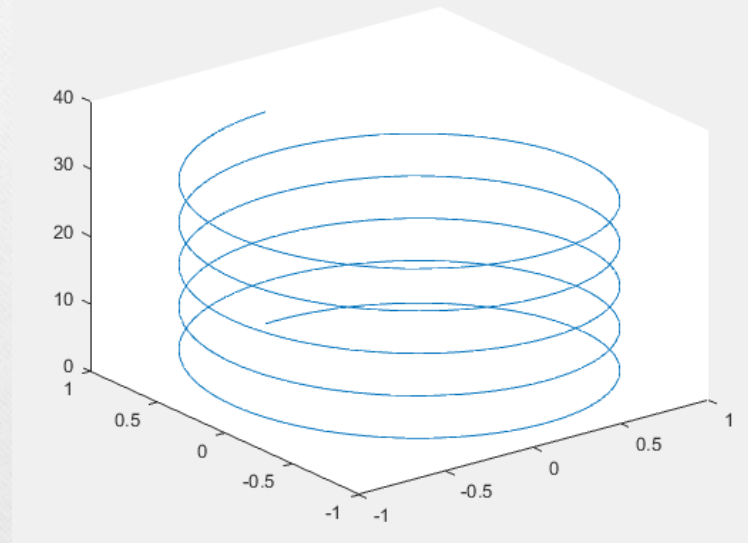


# 三维图形

三维曲线图 `plot3(x, y, z)`

当 $x, y, z$ 是同样长度的矢量时，绘出的是一条在三维空间贯穿的曲线。

例： `>>t=0:pi/50:10*pi;`  
`>>plot3(sin(t), cos(t), t)`



# 三维曲面图

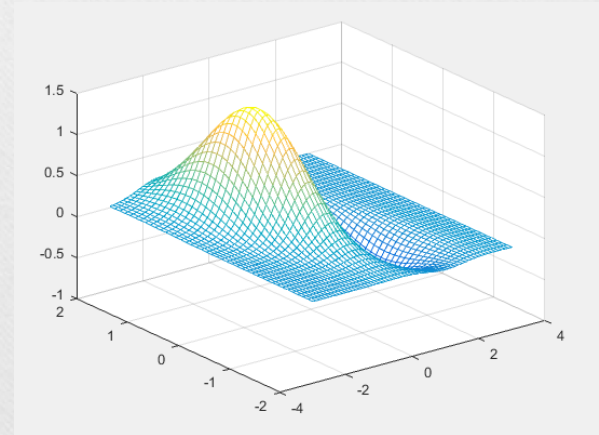
`meshgrid(x, y)` : 在绘制三维图之前, 先生成网格矩阵数据 `x` 和 `y` 。

`mesh(x, y, z)` : 绘制三维表面网格图。

`surf(x, y, z)` : 绘制三维表面着色图。

例: 绘制二元函数的曲线

```
>> [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)
```





# MATLAB程序设计基础

---

## 数据及数据文件

1. 数据类型：字符数组、数值数组、单元数组、结构数组、Java 类和函数句柄等。

## 2. 数据文件

- ① 二进制数据文件：以.mat 为扩展名。是标准的MATLAB 数据文件。
- ② ASCII 码数据文件：扩展名为.txt, .dat 等，可以是MATLAB 环境下存储，也可能是其他软件的计算结果。
- ③ 图象文件：扩展名为.bmp, .jpg .tif 等，用于图形图象处理。
- ④ 声音文件：扩展名为.wav

# M文件

MATLAB 的两种工作方式：

- (1) 交互式命令操作方式— 通过命令窗口进行交互式操作；
- (2) M文件的编程工作方式。

M 文件编程：

可用普通文本编辑器编制MATLAB 文件，文件由纯ASC II 字符组成，确定文件名后加.m 扩展名，称为M文件。运行M文件时，只需在命令窗口键入文件名即可。

注：

- (1) ‘%’ 引导注释行，不予执行；
- (2) 不需要用 “end” 作为M 文件的结束标志；
- (3) 若文件存放在自己的目录上，在运行文件前，应先将自己的目录设置为当前工作目录。最简单方法：在当前目录浏览器中设置。运行后存放在工作空间的变量可以用工作空间浏览器查看



# 函数文件

---

函数文件：第一个可执行行以function 开始

函数文件的基本组成部分：

函数定义行：如：

```
function[x, y, z]=sphere(theta, phi, rho)
```

也可以没有输出参数，如：function printresults(x)

nargin, nargout：输入和返回变量的个数

H1 行：帮助文本的第一行，是供lookfor 查询时使用的；

帮助文本：主要是为自己的函数文件建立在线查询信息；

函数体：包含全部的用于完成计算及给输出参数赋值等工作的语句；

注释：以 % 起始到行尾结束部分的说明文字。可放置在程序中的任何部位。

# 局部变量与全局变量

- 通常每个函数均有各自的局部变量
- 函数间的局部变量是相互独立的
- 如果将特定的变量名称声明为全局变量，则它们都共享该变量

用global 就可以把一个变量定义为全局变量。MATLAB 中变量名是区分大小写的，习惯上常将大写字母定为全局变量。

如：

```
global A B C
```

```
global x
```

函数1

```
function r1=func1  
global x  
r1 = x; %调用全局变量  
X =666 ; % 改变全局变量的值
```

函数2

```
function r2=func2  
global x  
r2 = x;
```



# 可变输入输出个数

varargin, varargout

例：可以计算两个多项式的积，

用varargin 实现任意多个多项式的积

```
function a=convsv(varargin)
%Compute the multiplication of variable multinomials

a=1;
for i=1:length(varargin)
    a=conv(a, varargin{i});
end
```

```
>> P=[1 2 4 0 5];
>> Q=[1 2];
>> F=[1 2 3];
```

```
>> D=convsv(P, Q, F)
```

```
D =
```

```
1    6   19   36   45   44   35   30
```

# 程序结构

8 种控制程序流程的语句: for, while, if, switch, try, continue, break, return

## 1. 循环语句

(1) for 语句:

for v= 表达式 % 通常为一个矢量: m:s:n语句体  
end

例:

```
>>n=10
>>for i=1:n
>> x(i)=(i+1).^2;
>>end
>>x
x =
4 9 16 25 36 49 64 81 100 121
```



# 程序结构

---

(2) While 语句：为条件循环语句。循环不确定次数，只要表达式的结果非零，语句体就重复执行，直到循环条件不成立为止

While 表达式  
语句体  
end

例：while 循环，求解  $n!$  达到100 位数的第一个  $n$

```
>>n=1;  
>>while prod(1:n)<1e100  
>> n=n+1;  
>>end  
>>n  
n =  
70
```

# 程序结构

---

## 2. 条件语句

(1) if—end 语句  
if 表达式  
语句体  
end

(2) if—else—end 语句  
if 表达式  
语句体1;  
else  
语句体2;  
end

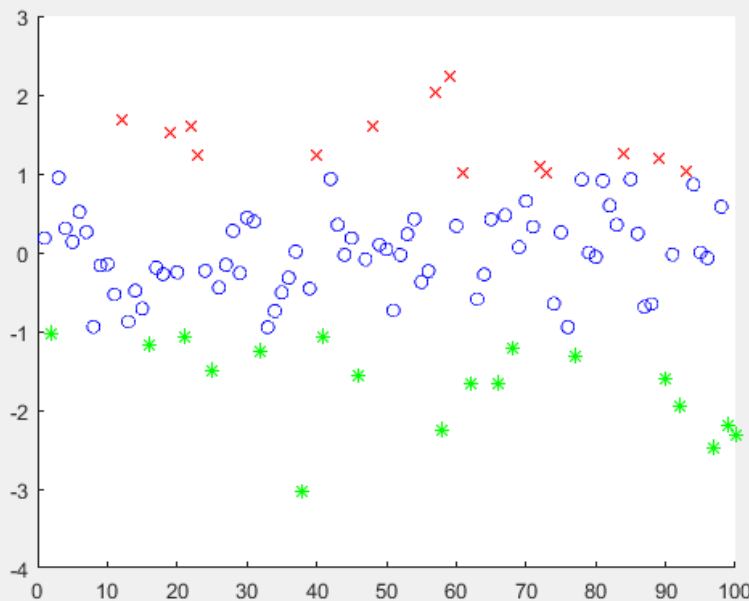
(3) if—elseif—end 语句  
if 表达式 1  
语句体1;  
elseif 表达式2  
语句体2;  
else  
语句体3;  
end



# 程序结构

例： 用色彩区分数据点的范围

```
>>n=100;  
>>x=1:n;  
>>y=randn(1,n);  
>>hold on  
>>for i=1:n  
>> if y(i)<-1  
>>   plot(x(i),y(i),'*g')  
>> elseif y(i)>=-1 & y(i)<=1  
>>   plot(x(i),y(i),'ob')  
>> elseif y(i)>=1  
>>   plot(x(i),y(i),'xr')  
>> end  
>>end  
>>hold off
```



# 程序结构

## 3. 分支语句：switch—case—end

通过对某个变量值的比较做各种不同的执行选择。

switch 表达式（数字或字符串）

case 数字或字符串1

语句体 1;

case 数字或字符串2

语句体 2;

.....

otherwise

语句体 n;

end

```
month=3;
switch month
    case {3,4,5}
        season='spring'
    case {6,7,8}
        season='summer'
    case {9,10,11}
        season='autumn'
    otherwise
        season='winter'
end
```



# 程序结构

## 4. 检测语句： 用于检测错误并改变流程

形式：

```
try  
语句体 1  
catch  
语句体2  
end
```

```
a=5;  
try  
    a=a+5;  
    disp(num2str(a)) %最后显示a数值  
catch  
    disp('Error')  
    a=nan  
end
```

```
a=5;  
try  
    a=a+q %不对, 没有变量q  
    disp(num2str(a)) %最后显示a数值  
catch  
    disp('Error')  
    a=nan  
end
```

# 程序结构

---

## 5. 其他流程控制语句:

(a) continue 语句: 用于控制for 循环和while 循环跳过某些执行语句

(b) break 语句: 用于终止for 循环和while循环的执行.

(c) return 语句: 用于终止当前的命令序列, 并返回到调用的函数或键盘

```
for i=1:100
    if i==10
        continue;
        j=1000;
    end
end
disp(j)
```



# MATLAB的符号运算

MATLAB 的符号运算是通过集成在MATLAB 中的符号数学工具箱（ Symbolic MathToolbox ）来实现的，它可完成几乎所有的符号运算功能。

## 一. 符号表达式的生成

### 1. 用单引号生成符号表达式

```
>>fun=' sin(x)'
```

### 2. 用函数sym( ) 建立符号 数组

```
>> A=sym(' [sin(x) b;c d]'
```

### 3. 用命令syms 生成符号 函数

```
>>syms K t T;fun=K*(exp(-t/T))
```

注：利用 函数symvar( ) 可知道符号表达式中哪些变量为符号变量。

# 符号表达式的基本运算

---

## 1 . 符号表达式的提取分子/分母运算

$[nem, den]=numden(f)$

## 2 . 符号表达式与数值表达式的相互转换

符号 表达式转换成数值 表达式:

$eval()$

数值 表达式转换成符号 表达式:

$sym()$



# 实例

---

例：

```
>>f= 'abs(-1)+sqrt(1)/2' , % 符号表达式
```

```
>>p=eval(f),n=sym(p) % 符号&数值转换
```

运行结果：

```
f =
```

```
abs(-1)+sqrt(1)/2
```

```
p =
```

```
1.5000
```

```
n =
```

```
3/2
```

# 符号表达式的化简

---

`simple()` % 将符号表达式化简成最简形式  
`pretty()` % 将符号表达式转换成公式的形式  
`collect()` % 将符号表达式的同类项合并  
`factor()` % 将符号表达式进行因式分解  
`expand()` % 将符号表达式进行展开



# 实例

例：求Vandermonde 矩阵  $A = \begin{bmatrix} 1 & 1 & 1 \\ a & b & c \\ a^2 & b^2 & c^2 \end{bmatrix}$  的行列式。

```
>>syms a b c
>>A=[1 1 1;a b c;a^2 b^2 c^2];
>>det(A) %求矩阵行列式 - 符号表达式
>>factor(ans) % 行列式因式分解
```

运行结果：

```
ans =
-a^2*b + a^2*c + a*b^2 - a*c^2 - b^2*c + b*c^2
ans =
[-1, b - c, a - c, a - b]
```

分解质因数  $f = \text{factor}(200)$

```
f = 1×5
    2 2 2 5 5
```

# 符号表达式的微积分

---

符号微分:

`y=diff(f)` % 求符号表达式f 对独立变量的 微分

`y=diff(f,n)` % 求符号表达式f 对独立变量的n 次微分

`y=diff(f,'x')` % 求符号表达式f 对变量x 的 微分

`y=diff(f,'x',n)` % 求符号表达式f 对变量x 的n 次微分

符号积分:

`y=int(f)` % 求符号表达式f 对独立变量的 不定积分

`y=int(f,'x')` % 求符号表达式f 对变量x 的 不定积分

`y=int(f,a,b)`

% 求符号表达式f 对独立变量从a 到b 的 定积分

`y=int(f,'x',a,b)`

% 求符号表达式f 对变量x 从a 到b 的 定积分



# 符号表达式的变换

---

## 1. Laplace 变换及其反变换

$$F=\text{laplace}(f, t, s), \quad f=\text{ilaplace}(F, s, t)$$

## 2. Z 变换及其反变换

$$F=\text{ztrans}(f, n, z), \quad f=\text{iztrans}(F, z, n)$$

## 3. Fourier 变换及其反变换

$$F=\text{fourier}(f, t, w), \quad f=\text{ifourier}(F, w, t)$$

# 符号表达式的求解

---

## 1. 符号代数方程求解

符号代数线性方程、符号代数非线性方程以及符号超越方程均可利用函数`solve()` 对其求解。

$$[x, y, z, \dots] = \text{solve}('eq1', 'eq2', 'eq3', \dots, 'a', 'b', 'c', \dots)$$

## 2. 符号微分方程可利用函数`dsolve()` 对其进行求解。

$$[y1, y2, \dots] = \text{dsolve}('eq1', 'eq2', \dots, 'cond1', 'cond2', \dots, 'x')$$



# 求常微分方程的解析解

---

$$\frac{d^4 y(t)}{dt^4} + 11 \frac{d^3 y(t)}{dt^3} + 41 \frac{d^2 y(t)}{dt^2} + 61 \frac{d y(t)}{dt} + 30 y(t) = e^{-6t} \cos 5t$$

```
>> syms t y;  
>> Y=dsolve('D4y+11*D3y+41*D2y+61*Dy+30*y= exp(-6*t)*cos(5*t)')  
>> pretty(Y)
```

# 结果

```
>> syms t y;
>>
>> Y=dsolve('D4y+11*D3y+41*D2y+61*Dy+30*y= exp(-6*t)*cos(5*t)')

Y =

(exp(-6*t)*(4*cos(5*t) - 5*sin(5*t)))/123 - (exp(-6*t)*(3*cos(5*t) - 5*sin(5*t)))/136 - (exp(-6*t)*(5*cos(5*t) - 5*sin(5*t)))/400 +

>> pretty(Y)
exp(-6 t) (cos(5 t) 4 - #1)   exp(-6 t) (cos(5 t) 3 - #1)   exp(-6 t) (cos(5 t) 5 - #1)   exp(-6 t) (cos(5 t) - #1)
----- + -----
          123                136                400                624

+ C1 exp(-3 t) + C3 exp(-2 t) + C4 exp(-t) + C2 exp(-5 t)

where

#1 == sin(5 t) 5
```