


R2016a (9.0.0.341360)  
64-bit (win64)  
February 11, 2016  
License Number: 123456

**MATLAB**<sup>®</sup>

Professional License

© 1984-2016 The MathWorks, Inc. Protected by U.S. and international patents. See mathworks.com/patents. MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See mathworks.com/trademarks for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.

 MathWorks

**R2016a**

# MATLAB 与工程计算

自动化学院

二教322

方璐 13588492201

[fanglu@hdu.edu.cn](mailto:fanglu@hdu.edu.cn)



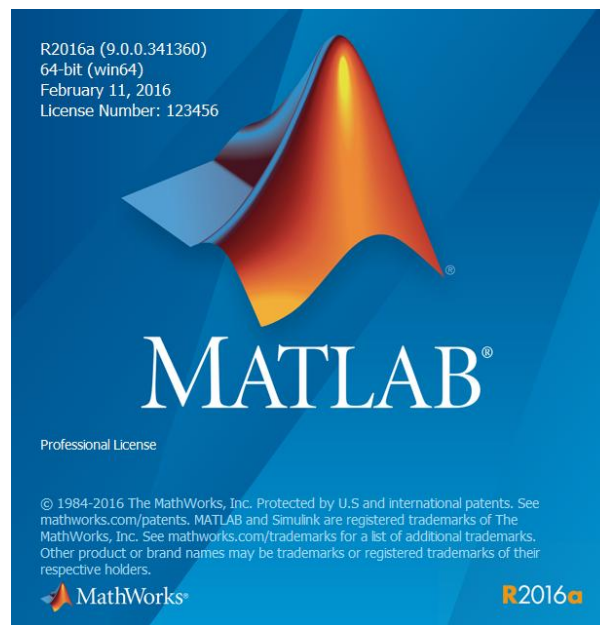
# MATLAB基础知识

- Matlab 是一个可视化的计算程序，被广泛地应用在科学运算领域里。它具有功能强大、使用简单等特点，内容包括：  
数值计算、符号计算、数据拟合、图形图像处理、系统模拟和仿真分析等功能。此外，用Matlab还可以进行动画设计、有限元分析等。

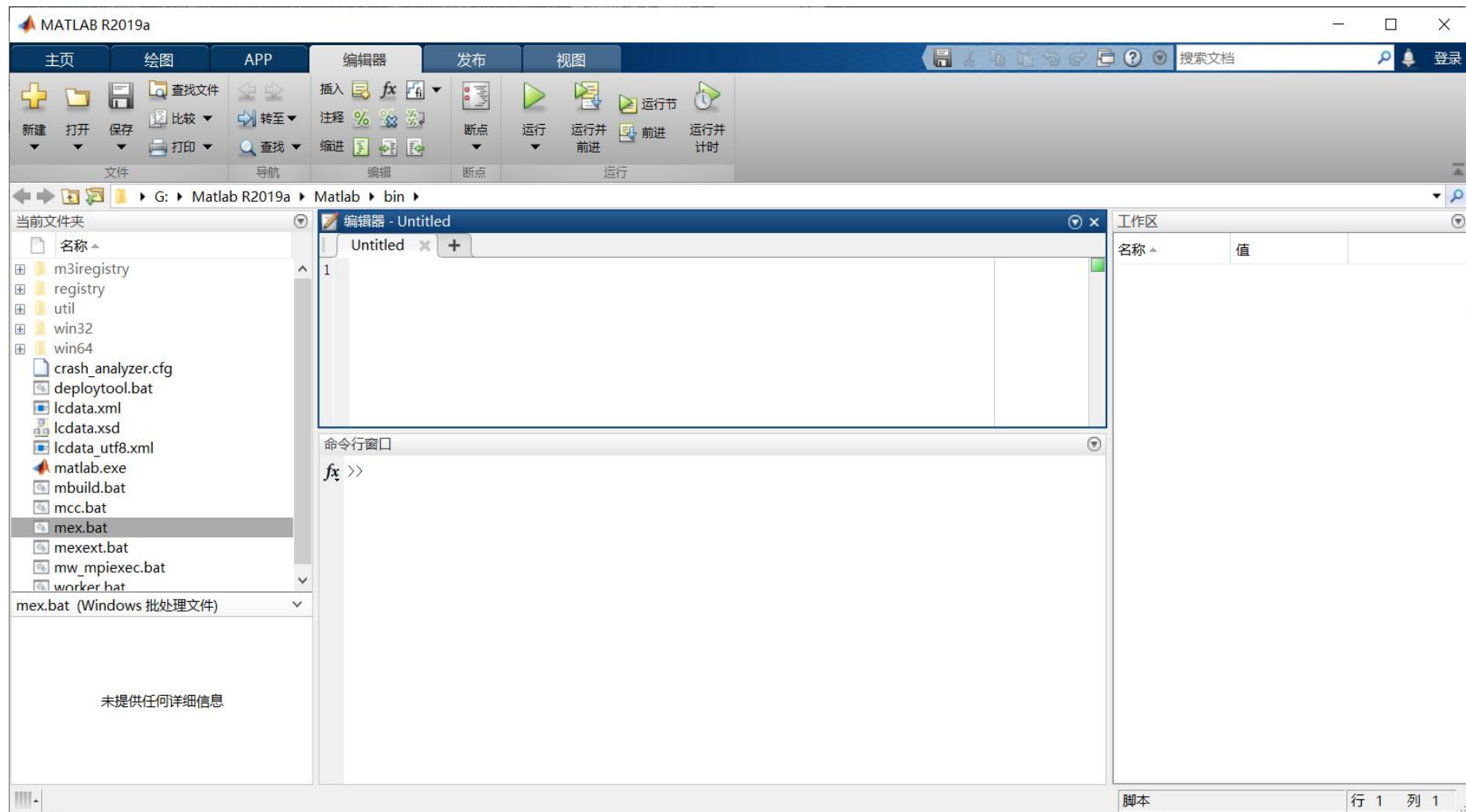
# MATLAB基础知识

MATLAB将计算、可视化和编程等功能集于一个易于使用的环境，并具有如下特点：

功能强大  
简单易学  
编程效率高



# MATLAB工作环境



# 命令窗口

- 激活命令窗口
- “>>”与闪烁的光标一起表明系统就绪，等待输入
- 命令窗口脱离MATLAB桌面

【例2.2-1】计算  $[12 + 2 \times (7 - 4)] \div 3^2$

(1) 在MATLAB命令窗口输入  
以下内容：

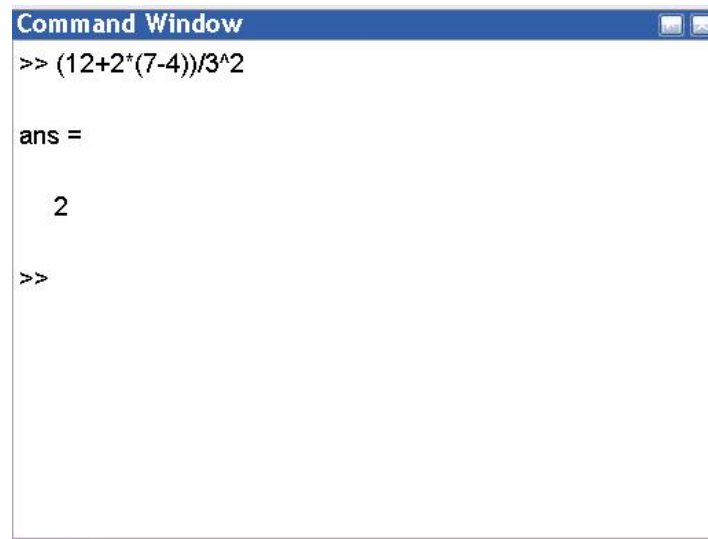
**$>> (12 + 2 * (7 - 4)) / 3^2$**

(2) 按【Enter】键，指令执行。

(3) 返回的计算结果：

ans=

2

A screenshot of the MATLAB Command Window. The title bar is blue with the text "Command Window" and standard window control buttons. The main area is white and contains the following text: the prompt ">>" followed by the expression "(12+2\*(7-4))/3^2", the prompt "ans =", the result "2", and the prompt ">>" at the bottom.

```
Command Window
>> (12+2*(7-4))/3^2

ans =

    2

>>
```

## 〔说明〕

- 在命令窗口 **【Enter】** 键提交命令执行。
- Matlab所用运算符（如+、-、^等）是各种计算程序中常见的。
- 计算结果中的“ans”是英文“answer”的一种缩写，其含义就是“运算答案”。ans是Matlab的一个预定义变量。

# 命令窗口

## ➤ 简单计算

【例2.2-2】 计算  $\sin(45^\circ)$

```
>> sin(45*pi/180)
```

```
ans=
```

```
0.7071
```

- **Matlab**中正弦函数**sin**就是常见的正弦函数。
- 它的参数值是以“弧度”为单位的。
- **pi**也是**Matlab**的预定义变量。
- **pi=3.14159...**
- **Matlab**对字母大小写是敏感的。

【例2.2-3】 计算  $\sqrt{2e^{x+0.5} + 1}$

其中  $x=4.92$

```
>> sqrt(2*exp(4.92+0.5)+1)
```

```
ans=
```

```
21.2781
```

- **Matlab**中开平方——**sqrt(x)**，是英文**square root**的缩写。
- **Matlab**中指数函数**exp(x)**，常见的表达方式。

# 命令窗口

➤ “clc”清除窗口显示内容的命令。

【例2.2-4】 计算  $y = \frac{2\sin(0.3\pi)}{1+\sqrt{5}}$  的值。

```
>>y=2*sin(0.3*pi)/(1+sqrt(5))
```

```
y=  
0.5000
```

【例2.2-5】 计算  $y = \frac{2\cos(0.3\pi)}{1+\sqrt{5}}$  的值。

```
>>y=2*cos(0.3*pi)/(1+sqrt(5))
```

```
y=  
0.3633
```

➤ 命令行编辑

➤ “↑”键调回已输入过命令。

➤ 修改。





# 命令窗口

【例2.2-5】 计算半径为5.2m的圆的周长和面积。

```
>>radius=5.2; %圆的半径
```

```
>>area=pi*5.2^2, circle_len=2*pi*5.2
```

```
area =
```

```
84.9487
```

```
circle_len =
```

```
32.6726
```

- 以上两例，命令行中用到了等号“=”。
- 计算结果不再赋给“ans”，而是赋给用户指定的变量y、**area**、**circle\_len**。
- 无论是预定义变量还是用户自定义变量都被存储在系统的工作空间内，即系统定义的一个存储窗口变量的内存空间。
- **Who**、**whos**命令用来显示工作空间的变量
- **clear**命令用来清除工作空间的变量。



# 命令窗口

---

**>>who**

**Your variables are:**

**ans          circle\_len          y**  
**area          radius**

**>>whos**

<b>Name</b>	<b>Size</b>	<b>Bytes</b>	<b>Class</b>
<b>ans</b>	<b>1x1</b>	<b>8</b>	<b>double array</b>
<b>area</b>	<b>1x1</b>	<b>8</b>	<b>double array</b>
<b>circle_len</b>	<b>1x1</b>	<b>8</b>	<b>double array</b>
<b>radius</b>	<b>1x1</b>	<b>8</b>	<b>double array</b>
<b>y</b>	<b>1x1</b>	<b>8</b>	<b>double array</b>

**Grand total is 5 elements using 40 bytes**

---



# 命令窗口

---

```
>>clear y
```

```
>>who
```

**Your variables are:**

**ans          circle\_len**

**area          radius**

```
>>clear ans area
```

```
>>whos
```

**Your variables are:**

<b>Name</b>	<b>Size</b>	<b>Bytes</b>	<b>Class</b>
<b>ans</b>	<b>1x1</b>	<b>8</b>	<b>double</b>
<b>area</b>	<b>1x1</b>	<b>8</b>	<b>double</b>
<b>circle_len</b>	<b>1x1</b>	<b>8</b>	<b>double</b>
<b>radius</b>	<b>1x1</b>	<b>8</b>	<b>double</b>

---



# 命令窗口

## ➤ 数值显示格式设置

- 缺省显示格式：简洁的短（short g）格式
- 窗口命令及语法格式：format 显示格式关键字

如：format long %15位数字显示

## 常见通用命令

命令

clc

clear

who或whos

dir

cd

type

help或doc

quit或exit

含义

清除命令窗口的显示内容

清除Matlab工作空间中保存的变量

显示Matlab工作空间中的变量信息

显示当前工作目录的文件和子目录清单

显示或设置当前工作目录

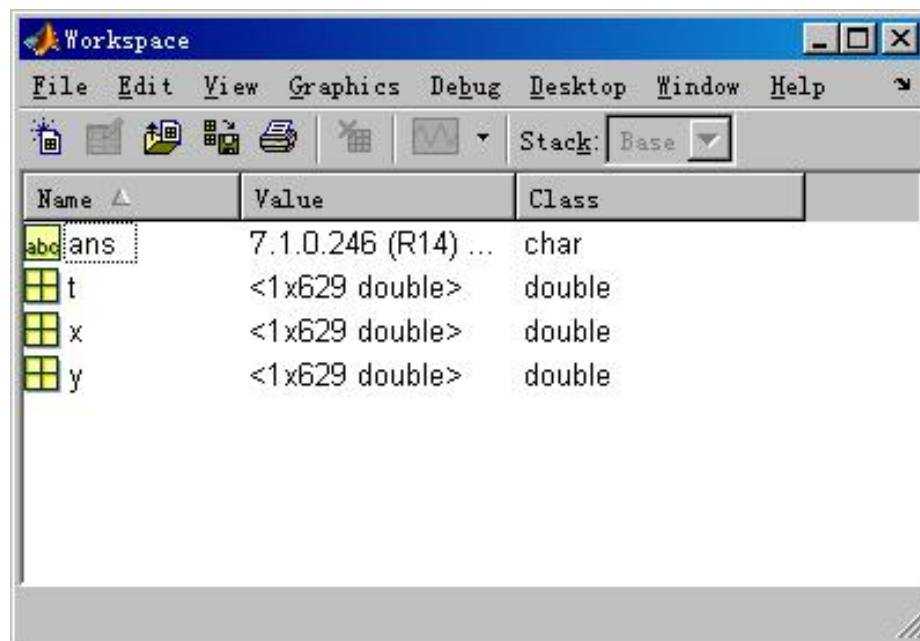
显示指定m文件的内容

获取在线帮助

关闭/推出MATLAB

# 工作空间

- 查看工作空间内存变量，可以由who、whos。
- 命名新变量。
- 修改变量名
- 删除变量
- 绘图
- 保存变量数据
- 装入数据





# 历史窗口

---

- 历史窗口：
  - 首先记录每次启动时间
  - 并记录在命令窗口输入命令，此次运行期间，输入的所有命令被记录为一组，并以此次启动时间为标志。
- 使用历史窗口：
  - 可以查看命令窗口输入过的命令或语句
  - 可以选择一条或多条命令执行拷贝、执行、创建M文件等。

要清除历史记录，可以选择**Edit**菜单中的**Clear Command History** 命令

---



# 当前目录窗口和搜索路径

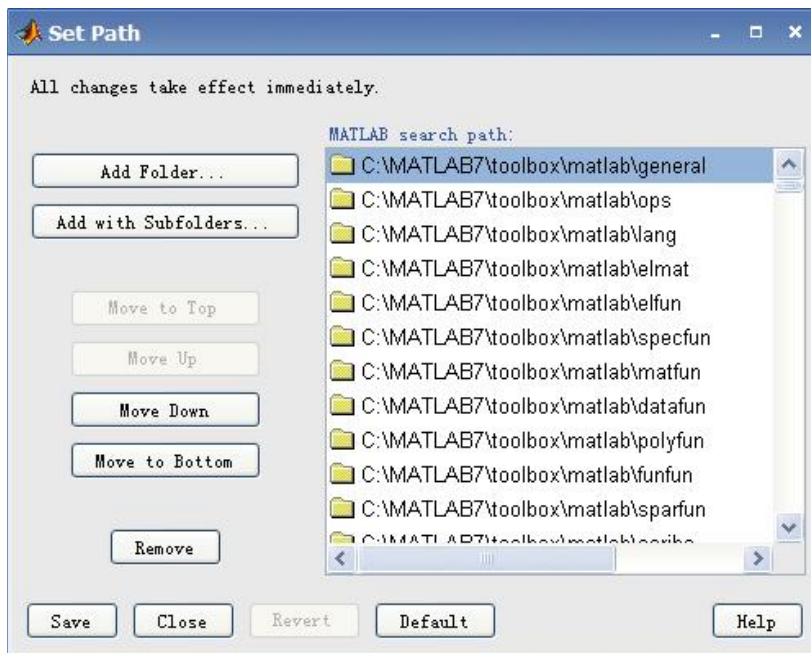
---

- **当前目录窗口：指Matlab运行时的工作目录。**
  - 只有在当前目录和搜索路径下的文件、函数才可以被运行和调用。
  - 如果没有特殊指明，数据文件也将存放在当前目录下；
  - 用户可以将自己的工作目录设置成当前目录，从而使所有操作都在当前目录中进行。

# 当前目录窗口和搜索路径

**搜索路径：**指**Matlab**执行过程中对变量、函数和文件进行搜索的路径。

在File菜单中选择Set Path命令或在命令窗口输入pathtool命令，出现搜索路径设置对话框：



！修改完搜索路径后，需要进行保存。





# 获取在线帮助

---

- **MATLAB**提供的帮助信息有两类
  - 简单纯文本帮助信息
    - **help**
    - **lookfor**（条件比较宽松）例：**inverse**
  - 窗口式综合帮助信息（文字、公式、图形）
    - **doc**
    - **helpwin**



# 数值表示、变量及表达式

## ➤ 数值的记述

➤ Matlab的数只采用习惯的十进制表示，可以带小数点和负号;其缺省的数据类型为双精度浮点型（double）

➤ 例如：3      -10      0.001      1.3e10      1.256e-6

## ➤ 变量命令规则

➤ 变量名、函数名对字母的大小写是敏感的，如myVar与myvar表示两个不同的变量

➤ 变量名第一个字母必须是英文字母

➤ 变量名可以包含英文字母、下划线和数字

➤ 变量名不能包含空格、标点

➤ 变量名最多可包含63个字符

---

# 数值表示、变量及表达式

## ➤ Matlab预定义的变量

变量名	意义
<b>ans</b>	最近的计算结果的变量名
<b>eps</b>	<b>MATLAB</b> 定义的正的极小值= <b>2.2204e-16</b>
<b>pi</b>	圆周率 $\pi$
<b>inf</b>	$\infty$ 值, 无限大
<b>i或j</b>	虚数单元, <b>sqrt(-1)</b>
<b>NaN</b>	非数, <b>0/0</b> 、 $\infty/\infty$

### 【说明】

- 每当MATLAB启动完成, 这些变量就被产生。MATLAB中, 被0除不会引起程序中断, 给出报警的同时用inf或NaN给出结果。
- 用户只能临时覆盖这些预定义变量的值, Clear或重启MATLAB可恢复其值。

# 数值表示、变量及表达式

## ➤ 运算符和表达式

运算	数学表达式	MATLAB运算符	MATLAB表达式
加	$a+b$	+	$a+b$
减	$a-b$	-	$a-b$
乘	$a \times b$	*	$a*b$
除	$a/b$ 或 $a \backslash b$	/ 或 \	$a/b$ 或 $a \backslash b$
幂	$a^b$	^	$a^b$

### 〔说明〕

- Matlab用“\”和“/”分别表示“左除”和“右除”。对标量而言，两者没有区别。对矩阵产生不同影响。MATLAB表达式的书写规则与“手写方式”几乎完全相同。
- 表达式按与常规相同的优先级自左至右执行运算。优先级：指数运算级别最高，乘除次之，加减最低。括号改变运算的次序。



# 计算演示：求根

---

求方程  $2x^5 - 3x^3 + 71x^2 - 9x + 13 = 0$  的全部根

**p = [2,0,-3,71,-9,13];** %建立多项式系数向量

**x = roots(p);** 求根

**x =**

**-3.4914**

**1.6863 + 2.6947i**

**1.6863 - 2.6947i**

**0.0594 + 0.4251i**

**0.0594 - 0.4251i**

---



## 计算演示：求解线性方程组

$$\begin{cases} 2x + 3y - z = 2 \\ 8x + 2y + 3z = 4 \\ 45x + 3y + 9z = 23 \end{cases}$$

**a** = [2,3,-1;8,2,3;45,3,9]; %建立系数矩阵**a**

**b** = [2;4;23]; %建立列向量**b**

**x** = inv(**a**)\***b**

**x** =

0.5531

0.2051

-0.2784



# 计算演示：求解线性方程组

---

**syms x y z** %建立符号变量

**[x,y,z]=solve(2\*x+3\*y-z-2,8\*x+2\*y+3\*z-4,45\*x+3\*y+9\*z-23)**

**x =**

**151/273**

**y =**

**8/39**

**z =**

**-76/273**



# 计算演示：求解定积分

求解  $I = \int_0^1 x \ln(1+x) dx$

```
quad('x.*log(1+x)',0,1)
```

```
ans =
```

```
0.250
```

或

```
syms x
```

```
int(x*log(1+x),0,1)
```

```
ans =
```

```
1/4
```

---



# 计算演示：多项式曲线拟合

考虑如下 x-y 一组实验数据：

$x=[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]$

$y=[1.2, 3, 4, 4, 5, 4.7, 5, 5.2, 6, 7.2]$

注：  $y(x) = x^3 - 2x^2 - 5$       In MATLAB     $y=[1 \quad -2 \quad 0 \quad -5]$

➤ 一次多项式拟合：

$p1 = \text{polyfit}(x,y,1)$  ➡

➤ 三次多项式拟合：

$p3 = \text{polyfit}(x,y,3)$  ➡

➤ plot 原始数据、一次拟合曲线和三次拟合曲线

$x2=1:0.1:10;$

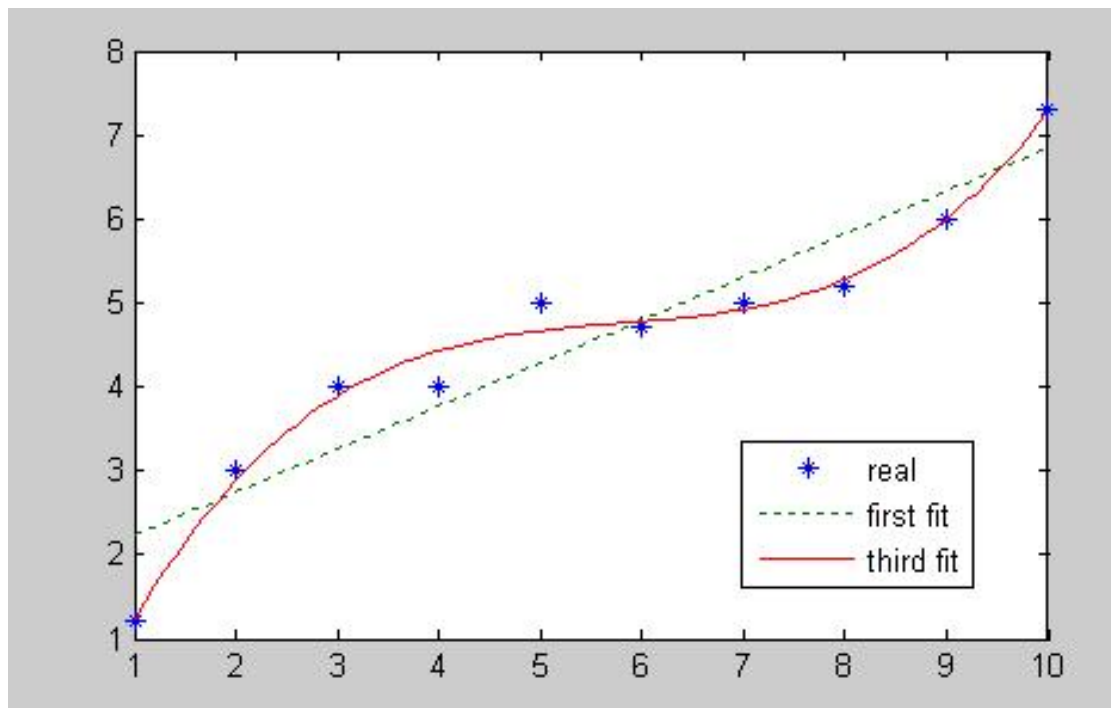
$y1=\text{polyval}(p1,x2)$

$y3=\text{polyval}(p3,x2)$

$\text{plot}(x, y, '*', x2, y1, ':', x2, y3)$

# 计算演示：多项式曲线拟合

拟合曲线图



由图可见，三次拟合结果较好



# 二维曲线绘图的基本操作

## ➤ **plot**指令的基本调用格式

### (1) **plot(x)**

- **x**为向量时，以该元素的下标为横坐标、元素值为纵坐标绘出曲线
- **x**为实数二维数组时，则按列绘制每列元素值相对其下标的曲线，曲线数等于**x**数组的列数。
- **x**为复数二维数组时，则按列分别以数组的实部和虚部为横、纵坐标绘制多条曲线

### (2) **plot(x, y)**

- **x**、**y**为同维数组时，绘制以**x**、**y**元素为横纵坐标的曲线
- **x**为向量，**y**为二维数组、且其列数或行数等于**x**的元素数时，绘制多条不同颜色的曲线
- **x**为二维数组，**y**为向量时，情况与上相同，只是**y**仍为纵坐标。

### (3) **plot(x1, y1, x2, y2, ...)**

- 绘制以**x1**为横坐标、**y1**为纵坐标的曲线1，以**x2**为横坐标、**y2**为纵坐标的曲线2，等等。
- 
- 其中**x**为横坐标，**y**为纵坐标，绘制 **$y=f(x)$** 函数曲线。

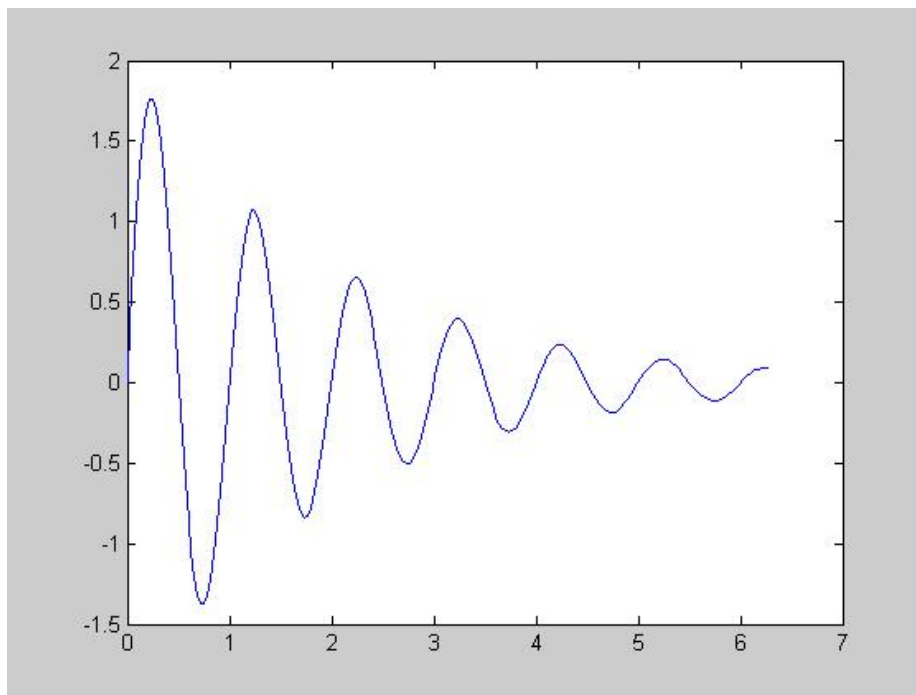
# 使用直角坐标系绘制曲线

在 $[0, 2\pi]$ 区间内，绘制曲线  $y = 2e^{-0.5x} \sin(2\pi x)$

```
x = 0:pi/100:2*pi;
```

```
y = 2*exp(-0.5*x).*sin(2*pi*x);
```

```
plot(x,y)
```



# 使用方程参数绘制曲线

## 绘制曲线

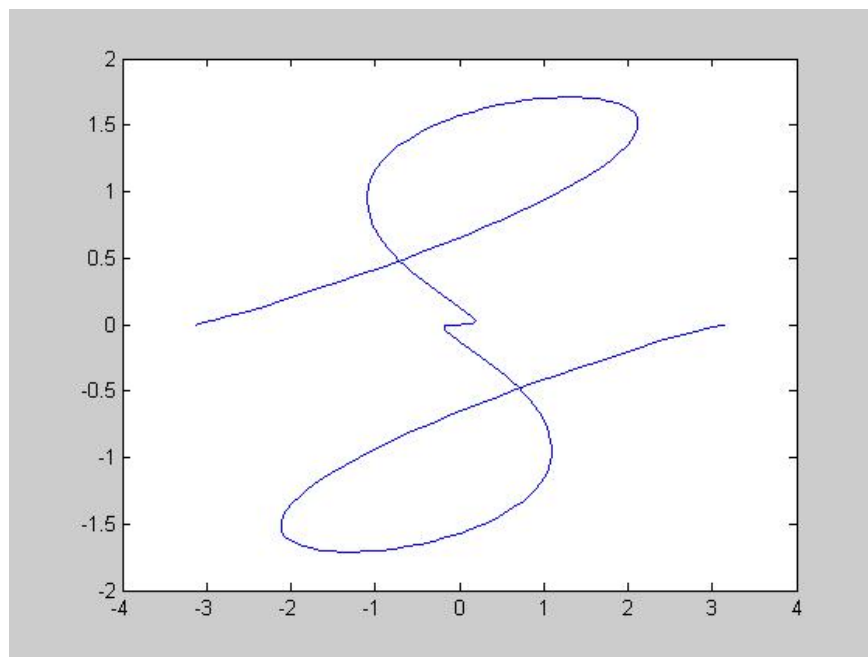
$$\begin{cases} x = t \cos(3t) \\ y = t \sin^2 t \end{cases}, -\pi \leq t \leq \pi$$

**t = -pi:pi/100:pi;**

**x = t.\*cos(3\*t);**

**y = t.\*sin(t).^2;**

**plot(x,y)**





# 多次叠绘、双纵坐标和多子图

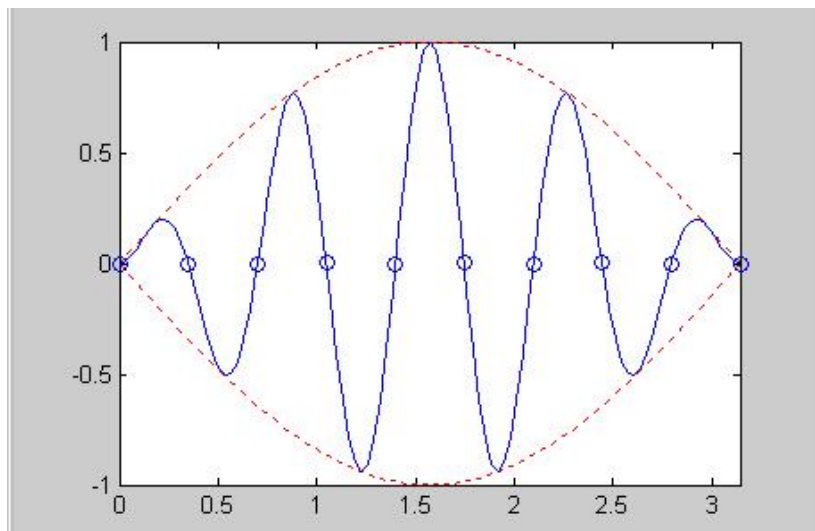
## ➤ 多次叠绘

- 多次调用**plot**命令在一幅图上绘制多条曲线，需要**hold**指令的配合。
- **hold on** 保持当前坐标轴和图形，并可以接受下一次绘制。
- **hold off** 取消当前坐标轴和图形保持，这种状态下，调用**plot**绘制完全新的图形，不保留以前的坐标格式、曲线。

# 多次叠绘

重绘波形  $y = \sin(t)\sin(9t)$  及其包络线

```
t=(0:pi/100:pi)';    %长度为101的时间采样列向量
y1=sin(t)*[1,-1];    %包络线函数值, 是(101x2)的矩阵
y2=sin(t).*sin(9*t); %长度为101的调制波列向量
t3=pi*(0:9)/9;
y3=sin(t3).*sin(9*t3);
plot(t,y1,'r:')
hold on
plot(t,y2,'b')
plot(t3,y3,'bo')
axis([0,pi,-1,1])
hold off
```



# 双纵坐标

➤ **plotyy**指令调用格式:

**plotyy(x1, y1, x2, y2)**

**x1-y1**曲线y轴在左, **x2-y2**曲线y轴在右。

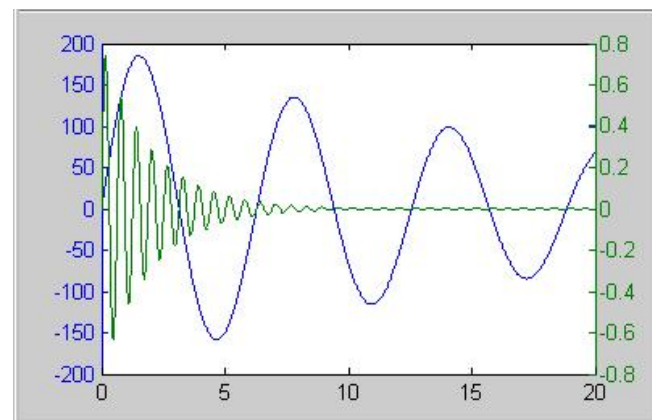
例:

**x = 0:0.01:20;**

**y1 = 200\*exp(-0.05\*x).\*sin(x);**

**y2 = 0.8\*exp(-0.5\*x).\*sin(10\*x);**

**plotyy(x,y1,x,y2);**







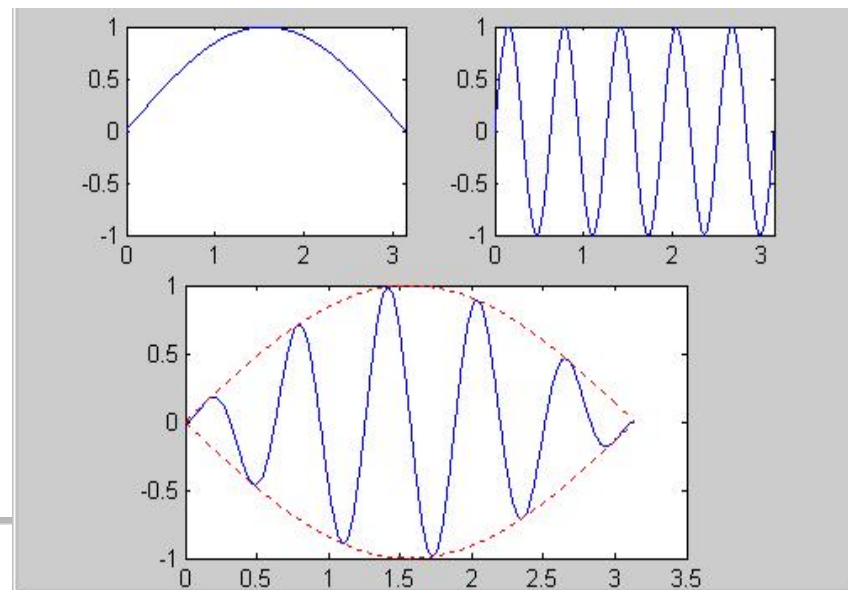
# 多子图

---

- **MATLAB**允许在同一图形窗口布置几幅独立的子图。  
具体指令：
- **subplot(m, n, k)**  
使 (**m****x****n**) 幅子图中第**k**个子图成为当前图
- **subplot('position', [left, bottom, width, height])**  
在指定的位置上开辟子图并成为当前图

# 多子图

```
t=(pi*(0:1000)/1000)';  
y1=sin(t);y2=sin(10*t);y12=sin(t).*sin(10*t);  
subplot(2,2,1),plot(t,y1);axis([0,pi,-1,1])  
subplot(2,2,2),plot(t,y2);axis([0,pi,-1,1])  
subplot('position',[0.2,0.05,0.6,0.45])  
plot(t,y12,'b-',t,[y1,-y1],'r:');
```



# 三维绘图

## ➤ 三维线图指令plot3

➤ 三维绘图指令中，**plot3**最易于理解，它的使用格式与**plot**十分相似，只是对应第3维空间的参量。

```
t=(0:0.02:2)*pi;
```

```
x=sin(t);
```

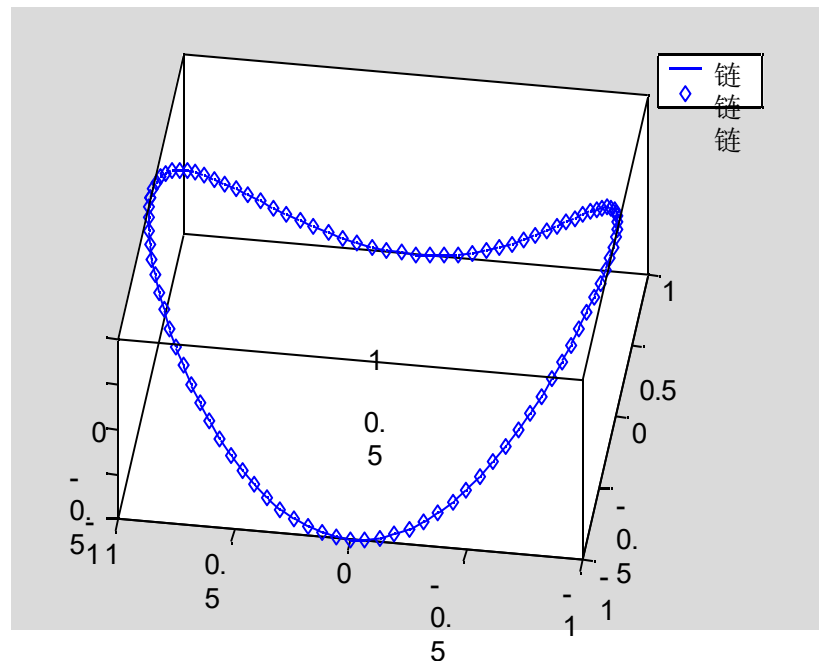
```
y=cos(t);
```

```
z=cos(2*t);
```

```
plot3(x,y,z,'b-',x,y,z,'bd');
```

```
view([-82,58]);
```

```
box on legend('链  
, '宝石')
```





# 图像文件的读写与图像显示

➤ **imread**指令

读取图像文件（ **BMP, GIF , PNG, JPEG, and TIFF**）

➤ **imshow**指令

显示图像

➤ **imwrite**指令

保存图像

➤ 例：读取图像文件

```
img1=imread('mudan.jpg');
```

```
% Load image
```

```
data
```

```
img2=imread('eight.tif');
```

```
whos  img1 img2
```



# 图像文件的读写与图像显示

---

Name	Size	Bytes	Class
img1	750x553x3	1244250	uint8 array
img2	242x308	74536	uint8 array

显示图像：

```
imshow(img1); % Display image
```

# 图像文件的读写与图像显示

Name	Size	Bytes	Class
img1	750x553x3	1244250	uint8 array
img2	242x308	74536	uint8 array

显示图像:

```
imshow(img1); % Display image
```





# 图像文件的读写与图像显示

## ➤ 简单图像处理

```
lighter = 2 * img1;
```

```
subplot(1,2,1);
```

```
imshow(img1);
```

```
title('Original'); % Display image
```

```
subplot(1,2,2);
```

```
imshow(lighter);
```

```
title('Lighter'); % Display image
```

---

# 图像文件的读写与图像显示

## 图像处理前后的比较

Original



Lighter







# 图像文件的读写与图像显示

---

➤ 保存图像

```
>> imwrite(lighter, 'mysaved.jpg')
```

➤ 查看保存结果

```
>> dir mysaved.*
```

```
mysaved.jpg
```



# Matlab命令的执行方式

## ➤ 交互式命令执行方式（命令窗口）

- 逐条输入，逐条执行，操作简单、直观，但速度慢，执行过程不能保留。

## ➤ M文件的程序执行方式

- 将命令编成程序存储在一个文件中（M文件），依次运行文件中的命令，可以重复进行。
- Matlab程序设计有传统高级语言的特征，又有自己独特的特点，可以利用数据结构的优点，使程序结构简单，编程效率高。



# M文件的分类

---

- 用Matlab语言编写的程序，称为M文件。
  - 是由若干Matlab命令组合在一起构成的，它可以完成某些操作，也可以实现某种算法。
- M文件根据调用方式的不同分为两类：
  - 命令文件（Script File）
  - 函数文件（Function File）
- 它们的扩展名都是.m



# 命令文件和函数文件的区别

- 命令文件没有输入参数，也不返回输出参数；
- 函数文件可以带输入参数，也可以返回输出参数。
- 命令文件对工作空间中的变量进行操作，文件中所有命令的执行结果也返回工作空间中；
- 函数文件中定义的变量为局部变量，当函数文件执行完毕时，这些变量也被清除。
- 命令文件可以直接运行；
- 函数文件不能直接运行，要以函数调用的方式来调用它。



# 命令文件

---

命令文件：

```
clear;
```

```
a = 1:10;
```

```
b = [11,12,13,14;15,16,17,18];
```

```
c = a; a = b; b = c;
```

a

b

将文件保存为exch，并在命令窗口执行。 执行结果：

```
a =  
11    12    13    14  
15    16    17    18
```

```
b =  
1     2     3     4     5     6     7     8     9    10
```



# 函数文件

---

fexch.m

```
function [a,b] = exch(a,b)
```

```
c = a; a = b; b = c;
```

然后在命令窗口调用该函数文件：

```
clear;
```

```
x = 1:10;
```

```
y = [11,12,13,14;15,16,17,18];
```

```
[x,y] = fexch(x,y)
```

输出结果为：

```
a =  
    11    12    13    14  
    15    16    17    18
```

```
b =  
     1     2     3     4     5     6     7     8     9    10
```

函数参数a, b, c未保留在工作空间中, x, y保留在工作空间中。



# 函数文件

---

## 1、函数名与函数文件名

函数文件名通常由函数名再加上扩展名.m组成。当函数文件名与函数名不同时，Matlab将忽略函数名而确认文件名因此调用时使用函数文件名。

## 2、函数调用的一般格式是：

[输出实参表] = 函数名(输入实参表)

注意：函数调用时，各实参出现的顺序、个数，应与函数定义时相同。