

# MATLAB基础

- > • MATLAB窗口环境
- > • MATLAB程序设计语言基础
- > • 矩阵运算和操作
- > • 脚本与函数
- > • 流程控制
- > • 绘图

# MATLAB介绍

- 它的首创者是美国新墨西哥大学计算机系的系主任Cleve Moler博士，他在教授线性代数课程发现其他语言很不方便，便构思开发了MATLAB。最初采用FORTRAN语言编写，20世纪80年代后出现了MATLAB的第二版，全部采用C语言编写。
- 1984年Moler博士和一批数学家及软件专家创建了MathWorks公司，专门开发MATLAB。
- 1993年出现了微机版，到2003年是6.5版
- 目前已经有matlab2016b版

# MATLAB语言的特点：

- 1) 语言简洁紧凑，使用方便灵活，库函数极其丰富

MATLAB程序书写形式自由，利用起丰富的库函数避开繁杂的子程序编程任务，压缩了一切不必要的编程工作。由于库函数都由本领域的专家编写，用户不必担心函数的可靠性。可以说，用MATLAB进行科技开发是站在专家的肩膀上。

## ■ 2) 运算符丰富

- 由于MATLAB是用C语言编写的，MATLAB提供了和C语言几乎一样多的运算符，灵活使用MATLAB的运算符将使程序变得极为简短。

- 
- 3) MATLAB既具有结构化的控制语句（如**for**循环，**while**循环，**break**语句和**if**语句），又有面向对象编程的特性。
-

- 
- 4) 程序限制不严格，程序设计自由度大
  - 例如，在**MATLAB**里，用户无需对矩阵预定义就可使用。
-

- 
- 5) 程序的可移植性很好，基本上不做修改就可以在各种型号的计算机和操作系统上运行。

- 
- 6) MATLAB的图形功能强大。
  - 在FORTRAN和C语言里，绘图都很不容易，但在MATLAB里，数据的可视化非常简单。MATLAB还具有较强的编辑图形界面的能力。
-



7) 功能强大的工具箱是MATLAB的另一特色。

- MATLAB包含两个部分：
- 核心部分：有数百个核心内部函数
- 各种可选的工具箱
- 工具箱又分为两类：功能性工具箱
- 学科性工具箱

功能性工具箱主要用来扩充其符号计算功能、图示建模仿真功能、文字处理功能以及与硬件实时交互功能，功能性工具箱用于多种学科。

学科性工具箱是专业性比较强的，面向专门的学科领域。如control toolbox, signal processing toolbox, communication toolbox等。这些工具箱都是由该领域内学术水平很高的专家编写的，所以用户无需编写自己学科范围内的基础程序，而可直接进行高、精、尖的研究。

- 符号数学工具箱
- **SIMULINK**仿真工具箱
- 控制系统工具箱
- 信号处理工具箱
- 图像处理工具箱
- 通讯工具箱
- 系统辨识工具箱
- 小波分析工具箱
- 神经网络工具箱
- 金融工具箱

## ■ 8) 源程序的开放性

- 开放性也许是**MATLAB**最受人们欢迎的特点。
- 除内部函数以外，所有**MATLAB**的核心文件和工具箱文件都是可读可改的源文件，用户可通过对源文件的修改以及加入自己的文件构成新的工具箱。

---

- 9) **MATLAB**的缺点:

- 和其他高级程序相比，程序的执行速度较慢。由于**MATLAB**的程序不用编译等预处理，也不生成可执行文件，程序为解释执行，所以速度较慢。
-

# **MATLAB语言的功能：**

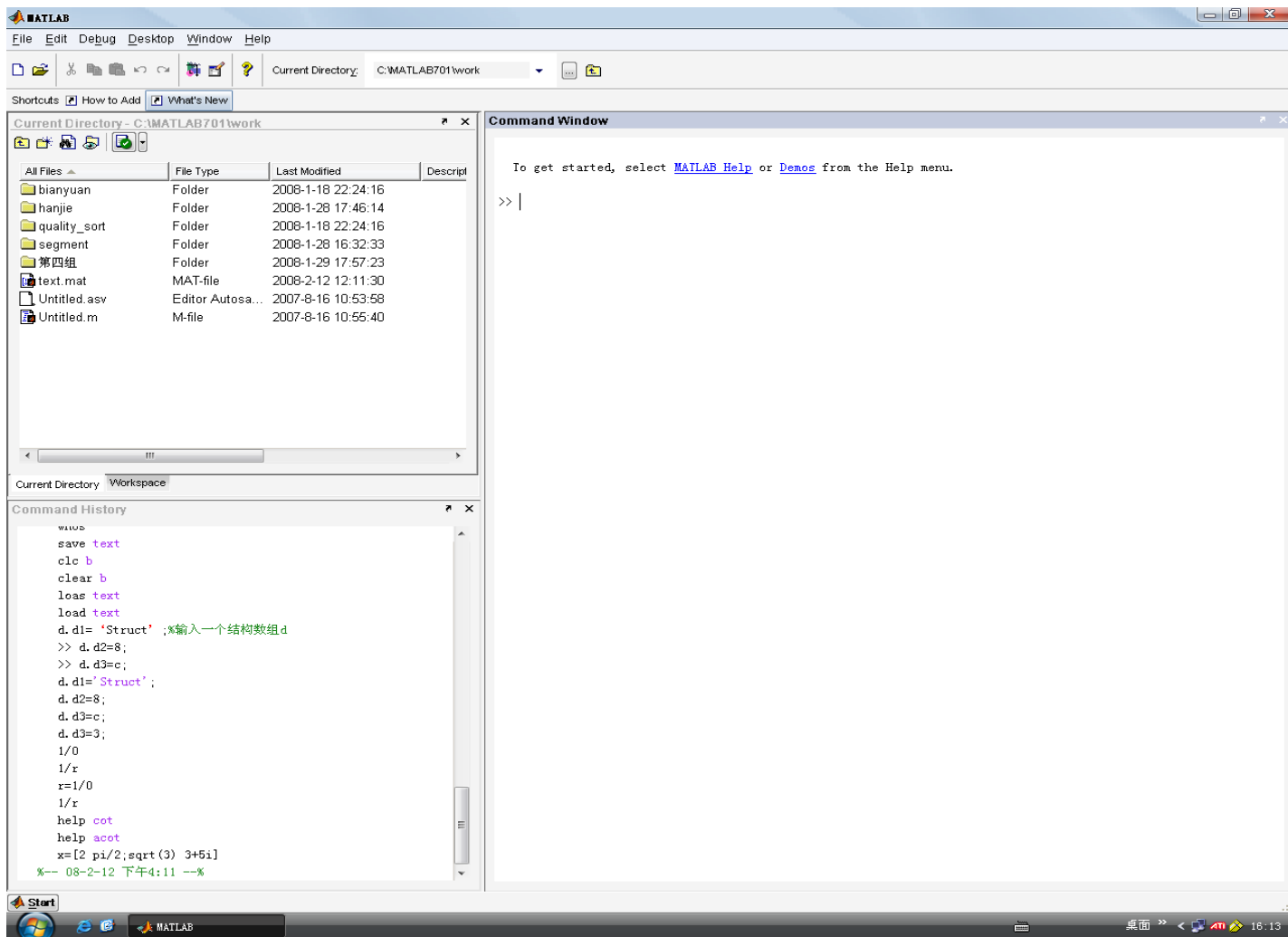
- 强大的数值（矩阵）运算功能
- 广泛的符号运算功能
- 计算结果的可视化功能
- 可靠的容错功能
- 应用灵活的兼容与接口功能
- 信息量丰富的联机检索功能

# Unit 1 MATLAB的窗口环境

---

# 1.1 Matlab窗口环境

- Command Window
  - Command History
  - Current Directory
  - Workspace
  - Help
-





# 1.命令窗口（command window）

- **Matlab**的主要操作界面，大部分操作命令和结果在此进行
- 命令提示符 **>>**
- 显示方式 字体、大小、颜色设置  
**files/preferences**

# 命令窗口常用控制指令

指 令	含 义	指 令	含 义
cd	设置当前工作目录	exit/quit	退出MATLAB
clf	清除图形窗	open	打开文件
clc	清除指令窗中显示的内容	md	创建目录
clear	清除内存变量	more	使显示内容分页显示
dir	列出指定目录的文件清单	type	显示M文件的内容
edit	打开M文件编辑器	which	指出文件所在目录

## 2. 历史命令窗口 (command history)

- 记录用户在Matlab命令窗口中输入的所有的命令
- 包括每次启动Matlab的时间和每次启动所有的命令行
- 对历史命令的编辑（复制/运行/创建m文件/快捷方式/profile code）

### 3. 当前目录窗口（current directory）

- 当前目录浏览器管理Matlab中大量的函数和数据库
- 包括菜单栏、当前目录设计区，工具菜单和文件详细菜单
- 对文件详细菜单区域中文件可以方便地进行编辑或运行

# 如何设置当前目录呢？

- 启动Matlab时，默认路径为
- `MATLAB701\work`
- Windows 创建目录，可以自动添加
- `Files/ set path`
- 利用`path`命令来查看matlab的路径信息

## 4.工作空间浏览器（workspace browser）

- 保存内存变量，从而可以方便地对其进行查阅和编辑等等（**双击**浏览数组编辑器或**右键**其他编辑）
- 可在命令窗口利用操作命令在查阅和删除内存变量

## 1.2 MATLAB命令窗口

### ■ 启动MATLAB命令窗口

安装**MATLAB**后，双击**MATLAB**图标，即可进入命令窗口，可以在命令窗口中直接输入命令语句。

### ■ MATLAB语句形式

**>> 变量=表达式;**

当键入回车时，该语句被执行。语句执行之后，窗口自动显示出语句执行的结果。如果希望结果不被显示，则只要在语句之后加上一个分号（；）即可。此时尽管结果没有显示，但它依然被赋值并在**MATLAB**工作空间中分配了内存。

## 1.2 MATLAB命令窗口

### ■ 命令行编辑器

(1) ↑:回调上一行命令; ↓:回调下一行命令

(2) 多行命令 (...)

如果命令语句超过一行或者太长希望分行输入，则可以使用多行命令继续输入。

例:  $S=1-12+13+4+\dots$   
9-4-18;



## 1.3 帮助的使用

- **Help 命令**: 已知命令不知道用法
- **Lookfor 命令**: 知道命令的关键词
- **Help Desk**: 命令查找, 索引, 说明书
- **Matlab**电子版手册
- **Demo**

# MATLAB的帮助系统

- 纯文本帮助
- 在MATLAB中，所有执行命令或者函数的M源文件都有详细的注释，用纯文本的形式来表示的，一般包括函数的条用格式或输入参数、输出结果的含义等等

## 基本帮助指令

在工作窗口输入：

`helpwin` 显示帮助窗口；

`helpdesk` 显示html格式的帮助内容

`demo` 显示Matlab Demo Window的窗口，选择感兴趣的内容进行演示

- 联机查询功能
- 两种方式：`helpwin`指令  
`help`帮助命令

# help帮助命令：

## 1. Help 指令+待查询的函数

命令窗口输入：**help exp** 得到指数函数命令的详细信息， **help sum** 加法函数

- 2.lookfor指令
- **help**需要事先知道函数准确的名字，如果不能确定时，可用**lookfor** +完整或不完整的关键词，进行搜索
- 如：**lookfor fft** ；检索出各种与傅立叶有关的函数

---

## Unit 2

# MATLAB程序设计语言基础

---

## 2.1 变量和常量

### ■ 变量的命名

- ❑ 变量的名字必须以字母开头（不能超过**19**个字符），之后可以是任意字母、数字或下划线；
- ❑ 变量名区分字母的**大小写**；
- ❑ 变量中不能包含有标点符号。

**例：**MYvar12, My\_var12, MyVar12\_均为有效的变量名

\_MyVar12, **1\_a** , d,minute为无效的变量名

ABC和ABc表达的是不同的变量

MATLAB命令通常是用小写字母书写。MATLAB中变量使用之前，不需要指定变量的数据类型，也不必事先声明变量。

## 2.1 特殊变量 (MATLAB默认的预定义变量)

预定义变量在MATLAB启动时由系统自动生成。用户在编写指令和程序时，应尽量避免使用下列预定义变量，以免混淆。

特殊变量	取值
<b>ans</b>	用于结果的缺省变量名
<b>pi</b>	圆周率
<b>eps</b>	计算机的浮点运算误差限 $2.2204 \times 10^{-16}$
<b>inf</b>	无穷大, 如 $1/0$
<b>nan</b>	不定式, 如 $0/0$
<b>i, j</b>	$i=j=\sqrt{-1}$
<b>nargin</b>	函数的输入变量数目
<b>nargout</b>	函数的输出变量数目
<b>realmin</b>	最小的可用正实数
<b>realmax</b>	最大的可用正实数

## 2.2 复数

- 复数可以表示为： $a=10-9i$
- 复数运算不需要特殊处理，可以直接进行

例：常用复数转换指令      `real`、`imag`、`abs`、`angle`

```
>> z1=3+4i                      %输入一个复数z1
```

```
z1 =  
    3.0000 + 4.0000i
```

```
>> a=real(z1)                    %求复数z1的实部
```

```
a =  
    3
```

```
>> b=imag(z1)                    %求复数z1的虚部
```

```
b =  
    4
```

```
>> r=abs(z1)                    %求复数z1的模
```

```
r =  
    5
```

```
>> theta=angle(z1)              %求复数z1的相角
```

```
theta =  
    0.9273
```



## 2.3 基本数学运算

### 1、常用的数学运算符

+, -, \* (乘), / (左除), \ (右除), ^ (幂)

在运算式中, **MATLAB**通常不需要考虑空格; 多条命令可以放在一行中, 它们之间需要用分号隔开; 逗号告诉**MATLAB**显示结果, 而分号则禁止结果显示。

### 2、常用数学函数

abs, sin, cos, tan, asin, acos, atan, sqrt, exp, imag, real, sign, log, log10, conj (共轭复数) 等

### 3. 指令行中常用的标点符号

标点在MATLAB指令中的作用极其重要。为了保证指令的正确执行，  
标点符号必须在英文状态下输入

名 称	标点	作 用
空格		分隔输入量；分隔数组元素
逗号	,	作为要显示结果的指令的结尾；分隔输入量；分隔数组元素
黑点	.	小数点
分号	;	命令结束；作为不显示结果的指令的结尾；分隔数组中的行
冒号	:	用作生成一维数组；用作下标时表示该维上的所有元素
注释号	%	其后内容为注释内容
单引号	' '	其内容为字符串
圆括号	()	用作数组标识；表示函数输入向量列表时用
方括号	[]	输入数组时用；表示函数输出向量列表时用
花括号	{ }	用作元胞数组标识
下连符	_	用在变量、函数和文件名中
续行号	...	将长指令行分成两行输入，保持两行的逻辑连续。

## 例 在MATLAB中输入矩阵

■ `>> a=[2,3,4;3,5,7;8,9,10]`      %显示结果

■ `a =`

■      2      3      4

■      3      5      7

■      8      9      10

■ `% [ ]`表示构成矩阵,分号分隔行,空格分隔元素

■ `>> a=[2,3,4;3,5,7;8,9,10];`    %不显示结果

只要是赋过值的变量，不管是否在屏幕上显示过，都存储在工作空间中，以后可随时显示或调用。变量名尽可能不要重复，否则会覆盖。

- ..... 续行命令
- >> B=1+25-36+.....
- 37-58
- B =
- -31

>> %两个数组点乘

C=[1,2,3].\*[4,5,6]

>> C =

4    10    18

%, 点乘

## ■ 冒号的作用

❶ 用于生成等间隔的向量，默认间隔为1.例如: $A=1:3; B=1:0.5:3;$

❷ 用于选出矩阵指定行、列及元素。例如: $A=B(i,:)$

❸ 循环语句 例如: $\text{for } i=1:3$

# 运算符和表达式

MATLAB表达式的规则与一般手写算式基本相同。

- a) 表达式由变量名、运算符和函数名组成。
- b) 表达式按优先级自左向右运算，括号可改变优先级顺序。
- c) 优先级顺序由高到底为：指数运算、乘除运算、加减运算。
- d) 表达式中赋值符为“=”。

## 2.4 数值显示

1. 在缺省情况下，当结果为整数，作为整数显示；当结果为实数，以小数后4位的精度近似显示。
2. 如果结果中的有效数字超出了这一范围，以科学计数法显示结果。
3. `format`只是影响结果的显示，不影响其计算与存储；MATLAB总是以双字长浮点数（双精度，值阈的近似范围 $[-1.7 \times 10^{308}, 1.7 \times 10^{308}]$ ）来执行所有的运算。

## 2.4 数值显示

`format short`: 短格式（**5**位定点数）

`format long`: 长格式（**15**位定点数）

`format short e`: 短格式**e**方式

`format long e`: 长格式**e**方式

`format bank`: **2**位十进制

`format hex`: 十六进制格式

**`format` 只改变变量的输出格式，  
但不会影响变量的值！**



# 各种 format 格式

格式	解释	例
format	短格式（缺省显示格式），同short	3.1416
format short	短格式（缺省显示格式），只显示5位	3.1416
format long	长格式，双精度数15位，单精度数7位	3.14159265358979
format short e	短格式e方式（科学计数格式）	3.1416e+000
format long e	长格式e方式	3.141592653589793e+000
format short g	短格式g方式	3.1416
format long g	长格式g方式	3.14159265358979
format compact	压缩格式	
format loose	自由格式	
format bank / format rat / format hex （银行/有理数/十六进制）		

## 2.5 文件、变量管理

- **Who:** 显示当前工作空间中所有变量的一个简单列表
- **Whos:** 列出变量的大小、数据格式等详细信息
- **Type:** Display contents of file
- **What:** List MATLAB files in current directory
- **Which:** Locate functions and files
- **Clear:** Removes all variables from the workspace
- **Clear variable1:** Removes variable1 from the workspace
- **Save:** Save workspace variables on disk
- **Load:** Load workspace variables from disk

# 内存变量的操作命令

## 1. 内存变量的查阅、删除

- 1) **who, whos** 指令查询内存变量  
**who**指令获取当前所有内存变量的名称列表。  
**whos**指令获取详细的内存变量列表，包含变量类型、大小等信息。
- 2) **clear** 指令删除内存变量  
**clear**指令清除所有内存变量  
**clear v1 v2**清除内存变量v1 v2，被删变量间用空格分隔
- 3) 打开**工作空间浏览器**( **Workspace** )，可以看到所有内存变量的详细说明，和**whos**指令效果相同。并且可以通过右键点击变量引出现场菜单，对该变量进行编辑、删除、重命名、保存、绘图等操作。

## 数据文件的存取

1) 利用指令save、load实现数据文件(\*.mat)的存取。

save/load filename %保存/打开文件

filename.mat

save/load filename v1 v2 %保存/打开变量v1 v2

2) 打开工作空间浏览器 (WorkSpace), 可以在右键现场菜单中选择[Save Workspace as]保存所有内存变量, 或[Save Selection as]保存指定变量。

[例]

>> a=[1 2 3;4 5 6;7 8 9]; %输入一个3\*3

数组a

b='This is a test'; %输入一个字符

串数组

c=3;

d=0.2;

---

- >> who

- Your variables are:

- a b c d

- >> whos

■	Name	Size	Bytes	Class
■	a	3x3	72	double array
■	b	1x14	28	char array
■	c	1x1	8	double array
■	d	1x1	8	double array

- Grand total is 25 elements using 116 bytes

- >> clear c

- >> save filename a b

- >> clear

- >> load filename

- >> who

- Your variables are:

- a b

---

# Unit 3

## 基本矩阵运算

# 3.1 矩阵输入

## 1. 从键盘输入

```
>> a=1; b=2; c=3;  
>> x=[5 b c; a*b a+c c/b]  
x=  
    5.000 2.000 3.000  
    2.000 4.000 1.500
```

```
>> y=[2, 4, 5  
      3 6 8]  
y=  
    2 4 5  
    3 6 8
```

矩阵生成不但可以使用纯数字（含复数），也可以使用变量和表达式；矩阵的元素直接排列在方括号内，行与行之间用分号隔开，每行内的元素使用空格或逗号隔开。大的矩阵可以用分行输入，回车键代表分号。

# 3.1 矩阵输入

## 2. 语句生成

(1) 用线性等间距生成向量矩阵 (start:step:end)

```
>> a=[1:2:10]
a=
    1    3    5    7    9
```

(2) `a=linspace(n1,n2,n)`

在线性空间上，行矢量的值从n1到n2，数据个数为n，缺省n为100。

```
>> a=linspace(1,10,10)
a=
    1    2    3    4    5    6    7    8    9   10
```



## 3.1 矩阵输入

### 2. 语句生成

(3) `a=logspace(n1,n2,n)`

在对数空间上，行矢量的值从 $10^{n1}$ 到 $10^{n2}$ ，数据个数为 $n$ ，缺省 $n$ 为50。

```
>> a=logspace(1,3,3)
```

```
a=
```

```
10  100 1000
```

# Matlab中用于产生特殊矩阵的函数

- **Matlab**提供了大量的函数来创建一些常用的特殊矩阵，例如对角阵、单位阵和零矩阵
- `zeros(m,n)` ——  $m \times n$  的零矩阵，全部元素都为0
- `ones(m,n)` —— 全部元素都为1的  $m \times n$  的矩阵
- `rand(m,n)` —— 随机矩阵
- `eye(m,n)` —— 单位矩阵
- `magic(n)` ——  $n$  维 magic 方阵
- 空阵 `[]` —— **matlab** 允许输入空阵，当一项操作无结果时，返回空阵。
- 对角矩阵： 对角元素向量  $V=[a_1,a_2,\dots,a_n]$      $A=\text{diag}(V)$
- 随机矩阵： `rand(m,n)` 产生一个  $m \times n$  的均匀的随机矩阵

# 用matlab函数创建和修改矩阵

- [例]: 0-1分布的随机矩阵, 利用rand函数产生任意行列的随机矩阵
- `>> a=rand(5,5)`

## [例] 利用diag产生对角阵

diag (M) M为矩阵或向量，对于矩阵，取对角元产生一个列向量；对于向量则产生一个对角阵

```
>> a=rand(5,5)
a =
    0.7027    0.7948    0.9797    0.1365    0.6614
    0.5466    0.9568    0.2714    0.0118    0.2844
    0.4449    0.5226    0.2523    0.8939    0.4692
    0.6946    0.8801    0.8757    0.1991    0.0648
    0.6213    0.1730    0.7373    0.2987    0.9883
```

```
>> d=diag(a) %矩阵产生列向量
```

```
d =
    0.7027
    0.9568
    0.2523
    0.1991
    0.9883
```

```
>> D=diag(d) %向量产生矩阵
D =
```

```
    0.7027         0         0         0         0
         0    0.9568         0         0         0
         0         0    0.2523         0         0
         0         0         0    0.1991         0
         0         0         0         0    0.9883
```

# 语句生成矩阵举例：

```
>> eye(2,3)
```

```
ans=
```

```
1 0 0
```

```
0 1 0
```

```
>> zeros(2,3)
```

```
ans=
```

```
0 0 0
```

```
0 0 0
```

```
>> ones(2,3)
```

```
ans=
```

```
1 1 1
```

```
1 1 1
```

```
>> V=[5 7 2]; A=diag(V)
```

```
A=
```

```
5 0 0
```

```
0 7 0
```

```
0 0 2
```

```
>> eye(2)
```

```
ans=
```

```
1 0
```

```
0 1
```

```
>> zeros(2)
```

```
ans=
```

```
0 0
```

```
0 0
```

```
>> ones(2)
```

```
ans=
```

```
1 1
```

```
1 1
```

如果已知A为方阵，则 $V=\text{diag}(A)$ 可以提取A的对角元素构成向量V。

## 3.2 矩阵运算

**转置：**对于实数矩阵用（'）符号或（. '）求转置结果是一样的；然而对于含复数的矩阵，则（'）将同时对复数进行共轭处理，而（. '）则只是将其排列形式进行转置。

```
» a=[1 2 3;4 5 6]'
```

```
a =
```

```
1    4  
2    5  
3    6
```

```
» a=[1 2 3;4 5 6].'
```

```
a =
```

```
1    4  
2    5  
3    6
```

```
» b=[1+2i, 2-7i]'
```

```
b =
```

```
1.0000 - 2.0000i  
2.0000 + 7.0000i
```

```
» b=[1+2i, 2-7i].'
```

```
b =
```

```
1.0000 + 2.0000i  
2.0000 - 7.0000i
```

## 3.2 矩阵运算-四则运算与幂运算

运算符:  $+$   $-$   $*$   $\backslash$   $/$   $^$ ;  $.*$   $.\backslash$   $./$   $.^$

例:  $a=[1, 2; 3, 4];$   
 $b=[3, 5; 5, 9]$

```
» a+b=[4 7; 8 13]
» a-b=[-2 -3; -2 -5]
» a*b=[13 23; 29 51]
» a/b=[-0.50 0.50; 3.50 -1.50]
» a\b=[-1 -1; 2 3]
» a^3=[37 54; 81 118]
```

```
» a.*b=[3 10; 15 36]
» a./b=[0.33 0.40; 0.60 0.44]
» a.\b=[3.00 2.50; 1.67 2.25]
» a.^3=[1 8; 27 64]
```

只有维数相同的矩阵才能进行加减运算。

只有当两个矩阵中前一个矩阵的列数和后一个矩阵的行数相同时,才可以进行乘法运算。 $a \backslash b$ 运算等效于求 $a*x=b$ 的解;而 $a/b$ 等效于求 $x*b=a$ 的解。只有方阵才可以求幂。

点运算是两个维数相同矩阵对应元素之间的运算,在有的教材中也定义为数组运算。

	MATLAB表达式
加	$a+b$
减	$a-b$
乘	$a*b$
除	$a/b$ 或 $a\b$
幂	$a^b$

## 矩阵加、减（+，-） 运算规则：

- ① 相加、减的两矩阵必须有相同的行和列两矩阵对应元素相加减。
- ② 允许参与运算的两矩阵之一是标量。标量与矩阵的所有元素分别进行加减操作。



# 矩阵乘（\*）运算

规则：

- A矩阵的列数必须等于B矩阵的行数

[例]  $a = [1 \ 2 \ 3; 4 \ 5 \ 6; 7 \ 8 \ 0]; b = [1; 2; 3]; c = a * b$

$c = 14$

$32$

$23$

- 标量可与任何矩阵相乘

[例]  $d = [-1; 0; 2]; f = \pi * d$

$f = -3.1416$

$0$

$6.2832$

# 矩阵除运算

在matlab中矩阵除的运算有两种矩阵除运算

MATLAB用 “/” 代表右除(常用除法)和 “\”左除运算。

这两种运算的差别如下：

例如：2/5 表示0.4， 2\5 表示2.5；

对于矩阵，

$A/B$  表示 $AB^{-1}$ ，即 $A*\text{inv}(B)$ ；

$B\backslash A$  表示 $B^{-1}A$ ，即 $\text{inv}(B)*A$ 。

矩阵除法可以看作矩阵乘法的逆运算

## [例]左除解方程组 $ax=b$ 的解 : $x=a\backslash b$

- `>> a=rand(5); %产生 (5*5)的均匀分布随机矩阵`
- `>> b=ones(5,1); %产生全为1的5元列向量`
- `>> x1=inv(a)*b`

■ `x1 =`

- `-0.1734`
- `0.5573`
- `0.0309`
- `0.7208`
- `0.8147`

■ `>> x2=a\b`

■ `x2 =`

- `-0.1734`
- `0.5573`
- `0.0309`
- `0.7208`
- `0.8147`

矩阵乘方—— $a^n, a^p, p^a$

$a^p$  ——  $a$  自乘 $p$ 次幂

方阵

$>1$ 的整数

对于 $p$ 的其它值,计算将涉及特征值和特征向量, 如果 $p$ 是矩阵,  $a$ 是标量  
 $a^p$ 使用特征值和特征向量自乘到 $p$ 次幂; 如 $a, p$ 都是矩阵,  $a^p$ 则无意义。

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

```
ans = 30    36    42
```

```
        66    81    96
```

```
       102   126   150
```

※当一个方阵有复数特征值或负实特征值时，非整数幂是复数阵。

---

$a^{0.5}$

ans =

**0.4498 + 0.7623i   0.5526 + 0.2068i   0.6555 - 0.3487i**

**1.0185 + 0.0842i   1.2515 + 0.0228i   1.4844 - 0.0385i**

**1.5873 - 0.5940i   1.9503 - 0.1611i   2.3134 + 0.2717i**

---

# 矩阵的一些特殊操作

- 矩阵的变维

`a=[1:12];b=reshape(a,3,4)`

- `c=zeros(3,4);c(:)=a(:)`

- `b =`     1     4     7     10

- 2     5     8     11

- 3     6     9     12

- 矩阵的变向

`rot90`:旋转; `fliplr`:左右翻转; `flipud`:上下翻转

- 矩阵的抽取

`diag`:抽取主对角线;`tril`: 抽取主下三角;

`triu`:抽取主上三角

# 矩阵的其它运算

- `inv` —— 矩阵求逆
- `size` —— 矩阵的大小（行列数）
- `det` —— 行列式的值
- `eig` —— 矩阵的特征值
- `rank` —— 矩阵的秩
- `'` —— 矩阵转置
- `sqrt` —— 矩阵开方
- `svd` —— 矩阵的奇异值分解



## 3.2 矩阵运算-逆矩阵与行列式

### 逆矩阵与行列式计算

- 求逆:  $\text{inv}(A)$ ;
- 求行列式:  $\det(A)$
- \* 要求矩阵必须为方阵

```
» a=[1 2 3; 4 5 6; 2 3 5];  
» b=inv(a)  
b =  
    -2.3333    0.3333    1.0000  
     2.6667    0.3333   -2.0000  
    -0.6667   -0.3333    1.0000  
» det(a) = -3
```

## 3.3 矩阵的超越函数

- MATLAB中sqrt、sin、cos等命令可以直接使用在矩阵上，这种运算只是定义在矩阵的单个元素上，即分别对矩阵的每个元素进行运算。MATLAB中也提供了基本的三角函数。
- 注意其中的取整

函数名	含义
<b>abs</b>	绝对值或者复数模
<b>sqrt</b>	平方根
<b>real</b>	实部
<b>imag</b>	虚部
<b>conj</b>	复数共轭
<b>round</b>	4舍5入到整数
<b>fix</b>	舍入到最接近0的整数
<b>floor</b>	舍入到最接近 $-\infty$ 的整数
<b>ceil</b>	舍入到最接近 $\infty$ 的整数

## 3.3 矩阵的超越函数

函数名	含义
<b>sign</b>	符号函数
<b>sin</b>	正弦
<b>cos</b>	余弦
<b>tan</b>	正切
<b>asin</b>	反正弦
<b>acos</b>	反余弦
<b>atan</b>	反正切

函数名	含义
<b>exp</b>	自然指数
<b>log</b>	自然对数
<b>log10</b>	以10为底的对数

## 3.4 矩阵操作

### 矩阵下标

MATLAB通过确认矩阵下标，可以对矩阵进行**插入子块**，**提取子块**和**重排子块**的操作。

- **A(m,n):** 提取第m行，第n列元素
- **A(:,n):** 提取第n列元素
- **A(m,:):** 提取第m行元素
- **A(m1:m2,n1:n2):** 提取第m1行到第m2行和第n1列到第n2列的所有元素（提取子块）。
- **A(:):** 得到一个长列矢量，该矢量的元素按矩阵的列进行排列。
- **矩阵扩展:** 如果在原矩阵中一个不存在的地址位置上设定一个数（赋值），则该矩阵会自动扩展行列数，并在该位置上添加这个数，而且在其他没有指定的位置补零。
- **消除子块:** 如果将矩阵的子块赋值为空矩阵[]，则相当于消除了相应的矩阵子块。

## 3.4 矩阵操作

### 矩阵的大小

- **[m,n]=size(A)**: 返回矩阵的行数m与n。
- **length(A)=max(size(A))**: 返回行数或列数的最大值。
- **rank(A)**: 求矩阵的秩

```
» a=[1 2 3;3 4 5];
```

```
» [m,n]=size(a)
```

```
m = 2, n = 3
```

```
» length(a)
```

```
ans = 3
```

```
» max(size(a))
```

```
ans = 3
```

```
» rank(a)
```

```
ans = 2
```

# 求最大值和最小值

**MATLAB**提供的求数据序列的最大值和最小值的函数分别为**max**和**min**

## 1. 求**向量**的最大值和最小值

求一个向量**X**的最大值的函数有两种调用格式，分别是：

- (1) **y=max(X)**：返回向量**X**的最大值存入**y**，如果**X**中包含复数元素，则按模取最大值。
- (2) **[y,I]=max(X)**：返回向量**X**的最大值存入**y**，最大值的序号存入**I**，如果**X**中包含复数元素，则按模取最大值。

求向量**X**的最小值的函数是**min(X)**，用法和**max(X)**完全相同。

例 求向量x的最大值。

命令如下：

- >> x=[-43, 72, 9, 16, 23, 47];
- y=max(x) %求向量x中的最大值
- [y, 1]=max(x) %求向量x中的最大值及其该元素的位置
- y =
- 72
- y =
- 72
- 1 =
- 2

## 2. 求矩阵的最大值和最小值

求矩阵A的最大值的函数有3种调用格式，分别是：

- (1)  $y = \max(A)$ ：返回一个行向量，向量的第 $i$ 个元素是矩阵A的第 $i$ 列上的最大值。
- (2)  $[Y, U] = \max(A)$ ：返回行向量Y和U，Y向量记录A的每列的最大值，U向量记录每列最大值的行号。
- (3)  $\max(A, [], \text{dim})$ ：dim取1或2。dim取1时，该函数和 $\max(A)$ 完全相同；dim取2时，该函数返回一个列向量，其第 $i$ 个元素是A矩阵的第 $i$ 行上的最大值。

求最小值的函数是min，其用法和max完全相同。



## 例 分别求 $3 \times 4$ 矩阵中各列和各行元素中的最大值

```
■ >> a=[9,6,7;20,9,2;15,13,0;3,4,6]
■ a =
■     9     6     7
■    20     9     2
■    15    13     0
■     3     4     6
■ >> y=max(a)           %每列的最大值
■ y =
■    20    13     7
■ >> [y,u]=max(a)       %每列的最大值及下标
■ y =
■    20    13     7
■ u =
■     2     3     1
■ >> max(a,[],2)        %dim=2, 每行的最大值
■ ans =
■     9
■    20
■    15
■     6
```

### 3. 两个向量或矩阵对应元素的比较

函数`max`和`min`还能对两个同型的向量或矩阵进行比较，调用格式为：

(1)  $U = \max(A, B)$ ：A, B是两个同型的向量或矩阵，结果U是与A, B同型的向量或矩阵，U的每个元素等于A, B对应元素的较大者。

(2)  $U = \max(A, n)$ ：n是一个标量，结果U是与A同型的向量或矩阵，U的每个元素等于A对应元素和n中的较大者。

`min`函数的用法和`max`完全相同。

例 求两个 $2 \times 3$ 矩阵x, y所有同一位置上的较大元素构成的新矩阵p。

```
>> a=[9,6,7;20,9,2]
```

```
■ a =
```

```
■      9      6      7
```

```
■     20      9      2
```

```
■ >> b=[15,13,0;3,4,6]
```

```
■ b =
```

```
■     15     13      0
```

```
■      3      4      6
```

```
■ >> u=max(a,b)    %找出同一位置的最大值
```

```
■ u =
```

```
■     15     13      7
```

```
■     20      9      6
```

---

# 求和与求积

数据序列求和与求积的函数是`sum`和`prod`，其使用方法类似。  
设`X`是一个向量，`A`是一个矩阵，函数的调用格式为：

`sum(X)`：返回向量`X`各元素的和。

`prod(X)`：返回向量`X`各元素的乘积。

`sum(A)`：返回一个行向量，其第`i`个元素是`A`的第`i`列的元素和。

`prod(A)`：返回一个行向量，其第`i`个元素是`A`的第`i`列的元素乘积。

`sum(A,dim)`：当`dim`为1时，该函数等同于`sum(A)`；当`dim`为2时，返回一个列向量，其第`i`个元素是`A`的第`i`行的各元素之和。

`prod(A,dim)`：当`dim`为1时，该函数等同于`prod(A)`；当`dim`为2时，返回一个列向量，其第`i`个元素是`A`的第`i`行的各元素乘积。

在MATLAB中，使用`cumsum`和`cumprod`函数能方便地求得向量和矩阵元素的累加和与累乘积向量，调用格式同上

例 求矩阵A的每行元素的乘积和全部元素的乘积。

```
■ >> a=[1,2,3;4,5,6]
```

```
■ a =
```

```
■     1     2     3
```

```
■     4     5     6
```

```
■ >> a_s=sum(a)           %每列之和
```

```
■ a_s =
```

```
■     5     7     9
```

```
■ >> a_s2=sum(a,2) %dim=2, 每行之和
```

```
■ a_s2 =
```

```
■     6
```

```
■    15
```

```
■ >> a_p=prod(a)         %每列之积
```

```
■ a_p =
```

```
■     4    10    18
```

```
■ >> a_p2=prod(a,2)      %每行之积
```

```
■ a_p2 =
```

```
■     6
```

```
■    120
```

%累加和举例cumsum函数

```
>> a=[1,2,3;4,5,6;3,4,5]
```

```
a =
```

```
     1     2     3
```

```
     4     5     6
```

```
     3     4     5
```

```
>> a_cs=cumsum(a) %求每列的累加和
```

```
a_cs =
```

```
     1     2     3
```

```
     5     7     9
```

```
     8    11    14
```

# 求平均值和标准方差、中值与相关系数

求数据序列平均值的函数是mean

求数据序列标准方差的函数是std

求数据序列中值的函数是median

corrcoef函数可求出数据的相关系数矩阵

---

例 生成满足正态分布的 $10000 \times 5$ 随机矩阵，然后求各列元素的均值和标准方差，再求这5列随机数据的相关系数矩阵

命令如下：

**`X=randn(10000,5);`**

**`M=mean(X)`**

**`D=std(X)`**

**`R=corrcoef(X)`**

---

## 3.5 矩阵的查找和排序（1）

- 子矩阵的查找使用**find**命令完成，它返回关系表达式为真的下标。例如：

```
>> a=[10: 20];  
>> find(a>15)  
ans =  
      7      8      9     10     11
```



## 3.5 矩阵的查找和排序（2）

- 矩阵的排序使用**sort**函数，它将矩阵按照升序排列。

**sort**函数也可以对矩阵A的各列或各行重新排序，其调用格式为：

**[Y,I]=sort(A,dim)**

其中**dim**指明对A的列还是行进行排序。若**dim=1**，则按列排；若**dim=2**，则按行排。**Y**是排序后的矩阵，而**I**记录**Y**中的元素在**A**中位置。

```
>> A = [15:-3:1]
```

```
A =
```

```
15  12   9   6   3
```

```
>> B = sort(A)
```

```
B =
```

```
3   6   9  12  15
```

## 例 对二维矩阵做各种排序

```
■ >> a=[10,2,3;41,25,6;3,1,5]
```

```
■ a =
```

```
■    10    2    3  
■    41   25    6  
■     3    1    5
```

```
■ >> a_s=sort(a)
```

%按列进行从小到大排序

```
■ a_s =
```

```
■     3    1    3  
■    10    2    5  
■    41   25    6
```

```
■ >> [a_s,l]=sort(a)
```

% l给出排序后各元素在原始矩阵中所在的行

```
■ l =
```

```
■     3    3    1  
■     1    1    3  
■     2    2    2
```

```
■ >> a_s=sort(a,2)
```

%按行进行排序

```
■ a_s =
```

```
■     2    3   10  
■     6   25   41  
■     1    3    5
```

## 3.6 关系运算符

MATLAB的运算符有三种类型：算术运算符、关系运算符、逻辑运算符。它们的处理顺序依次为算术运算符、关系运算符、逻辑运算符。

### 关系运算符

假设有：A=[1 2 -1 -5]    B=[0 2 3 1]

< 小于            A<B    ans=[0 0 1 1]; A < 1    ans=[0 0 1 1]

> 大于            A>B    ans=[1 0 0 0]; A > 1    ans=[0 1 0 0]

<= 小于等于      A<=B    ans=[0 1 1 1]

>= 大于等于      A>=B    ans=[1 1 0 0]

== 等于            A=B    ans=[0 1 0 0]; A = 1    ans=[1 0 0 0]

— ~= 不等于        A~=B    ans=[1 0 1 1]; A ~=1    ans=[0 1 1 1] —

# 关系运算

- **运算法则**
- 两个变量是标量**a**和**b**时，则如果关系成立，结果为1，否则为0；
- 两个维数相同的数组**A**和**B**时，比较相同位置的元素，按标量运算规则逐个进行；
- 维数相同的数组**A**和标量**b**时，把标量**b**与数组中的每一个元素逐个比较；
- 优先级：高到低为算术运算、关系运算、逻辑运算
- **<**、**<=** 和 **>**、**>=**仅对变量的**实部**进行比较，而**==** 和 **~=** 则同时对**实部和虚部**进行比较。

# 例:

- `a=[2 3 4 5 6 7 8];`

- `b=[6 5 4 3 2 1 0];`

- `>> t=a>4`

- `t =`

- `0 0 0 1 1 1 1`

- `>> t1=(a==b)`

- `t1 =`

- `0 0 1 0 0 0 0`

- 说明: `=`和`==`的不同: `==`比较两个变量, 相等时返回1, 否则, 返回0; `=`示将结果赋给一个变量

## 3.6 逻辑运算符

注意：在处理逻辑运算时，运算元只有两个值即0和1，所以如果指定的数为0，MATLAB认为其为0，而任何数不等于0，则认为是1。

设有：A=[5 -4 0 -0.5]    B=[0 1 0 9]

& 与

$$A \& B = [0 \ 1 \ 0 \ 1]; \quad A \& 1 = [1 \ 1 \ 0 \ 1]$$

| 或

$$A | B = [1 \ 1 \ 0 \ 1]; \quad A | 1 = [1 \ 1 \ 1 \ 1]$$

~ 非

$$\sim A = [0 \ 0 \ 1 \ 0]; \quad \sim 1 = 0$$

xor 异或

$$\text{xor}(A, B) = [1 \ 0 \ 0 \ 0]$$

# MATLAB关系与逻辑函数、测试函数

## ❖ 部分常用逻辑函数和测试函数

**all函数**：当某列的元素都为真时，返回值为1，否则返回0。最终运算结果为一个0-1行向量

**any函数**：当向量中至少有一个元素为真时，返回值为1，否则返回0。最终运算结果为一个0-1行向量

**find函数**：用于查找向量中的真元素的下标，返回由所有真元素下标构成的列向量。

# Unit 4

## 脚本与函数



## 4.1 MATLAB的程序类型

**MATLAB**的程序类型有三种，一种是在命令窗口下执行的**M**文件；另外一种是可以存取的**M**文件，也即程序文件；最后一种是函数（**function**）文件。

### 1、直接执行**M**文件

在命令窗口中输入并执行，它所用的变量都要在工作空间中获取，不需要输入输出参数的调用，退出**MATLAB**后就释放了。

### 2、程序文件

- 以**.m**格式进行存取，包含一连串的**MATLAB**指令和必要的注解。没有输入参数，也不会返回参数。
- 执行方式: 在**Command Window**中键入其名称并回车。

## 4.1 举例

例: 求出  $\sum_{i=1}^m i > 10000$  的最小m值

编写程序如下:

```
s=0; m=0;
```

```
while (s<=10000)
```

```
    m=m+1;
```

```
    s=s+m;
```

```
end
```

```
[s, m]
```

```
s=10011
```

```
m=141
```

## 4.1 MATLAB的程序类型（续）

### 3、函数文件

- 函数定义行（关键字function）

`function[out1,out2,...]=filename(in1,in2,...)`

输入和输出（返回）变量的个数分别由nargin和nargout两个MATLAB保留的变量来给出, 只要进入该函数, Matlab就将自动生成这两个变量。

- 函数体说明及有关注解

以（%）开头，只起注释作用, 不执行; 使用help命令可以显示出注释语句的内容

如果不希望显示某段信息，可在它的前面加空行

- 函数体语句

## 4.2 M文件的编辑及工作路径的设置

- 进入**MATLAB**的**Editor/Debugger**窗口来编辑程序
- 在编辑环境中，文字的不同颜色显示表明文字的不同属性。**绿色**：注解；**黑色**：程序主体；**蓝色**：控制流程关键字；**紫色**：属性值的设定。
- 在运行程序之前，必须设置好**MATLAB**的工作路径，使得所要运行的程序及运行程序所需要的其他文件处在当前目录之下，否则可能导致程序无法执行。
- 通过**cd**指令和路径浏览器（**path browser**）可以更改、显示当前工作路径。

## 4.3 声明子程序（函数程序）变量

- 1、子程序与主程序之间的数据是通过参数进行传递的，子程序应用主程序传递来的参数进行计算后，将结果返回主程序。
- 2、在一个函数内，除输入、输出变量外，其他在函数内部产生的所有变量都是局部变量，在函数调用结束后，这些变量将消失。如果两个或多个函数共用一个变量（或者说在子程序中也要用到主程序中的变量，**注意不是参数**），那么可以用**global**来将它声明为全局变量。

全局变量的使用可以减少参数传递，合理利用全局变量可以提高程序执行的效率。

# Unit 5

## 流程控制

# 流程控制

- 循环语句 **for, while**
- 条件转移 **if end, if elseif else end**
- 开关语句 **switch case**
- 注释语句 **%**
- 中断语句 **break**
- 暂停语句 **pause**
- 回显语句 **echo on/off**

# 5.1 For循环

## 基本格式

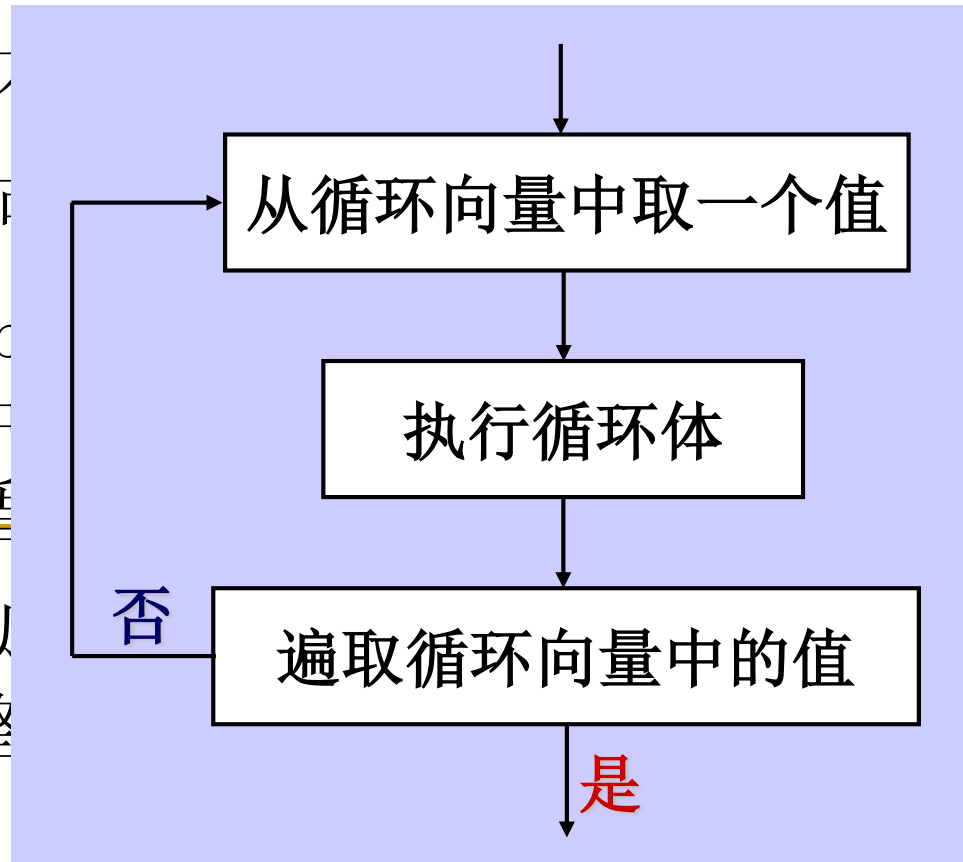
```
for 循环变量=起始值: 步长: 终止值  
    循环体  
end
```

- 步长缺省值为1，可以在正实数或负实数范围内任意指定。对于正数，循环变量的值大于终止值时，循环结束；对于负数，循环变量的值小于终止值时，循环结束。循环结构可以嵌套使用。



# For循环应当注意的一些方面

1. for循环不
2. for循环可
3. 在Matlab  
实际编程中  
要采用循
4. 循环可以不跳出整



量而终止;

是, 所以在实  
问题, 尽量不  
效率。

在的循环,

```
for i=1:5
    for j=1:5
        a(i, j)=1/(i+j-1);
    end
end
```

end

a=

1.0000	0.5000	0.3333	0.2500	0.2000
0.5000	0.3333	0.2500	0.2000	0.1667
0.3333	0.2500	0.2000	0.1667	0.1429
0.2500	0.2000	0.1667	0.1429	0.1250
0.2000	0.1667	0.1429	0.1250	0.1111

## 5.2 while循环

### 基本格式

```
while 表达式  
    循环体  
end
```

- 若表达式为真，则执行循环体的内容，执行后再判断表达式是否为真，若不为真，则跳出循环体，向下继续执行。

while循环和for循环的区别：

- ◆ while循环结构的循环体被执行的次数是不确定的(基于判断语句执行)；
- ◆ for结构中循环体的执行次数是确定的。

用**while**循环语句编写一个计算**1000**以内的  
**Fibonacci**数:

```
f=[1 1];
```

```
i=1;
```

```
while f(i)+f(i+1)<1000
```

```
    f(i+2)=f(i)+f(i+1);
```

```
    i=i+1;
```

```
end
```

```
f=
```

```
1  1  2  3  5  8 13 21 34 55 89 144 233  
377 610 987
```

## 5.3 if-else-elseif 语句

```
(1) if 逻辑表达式  
    执行语句  
end
```

```
(2) if 逻辑表达式  
    执行语句1  
    else 执行语句2  
end
```

```
(3) if 逻辑表达式1  
    执行语句1  
    elseif 逻辑表达式2  
        执行语句2  
    ...  
    else  
    end
```

当逻辑表达式的值为真时，执行该结构中的执行语句，执行完之后继续向下进行；若为假，则跳过结构中的内容，向下执行。

## 5.4 switch语句

格式:

```
switch  表达式  
case    值1  
        语句1  
case    值2  
        语句2  
        ....  
otherwise  
        语句3  
end
```

执行方式:

表达式的值和哪种情况(case)的值相同, 就执行哪种情况中的语句, 如果不同, 则执行otherwise中的语句。格式中也可以不包括otherwise, 这时如果表达式的值与列出的各种情况都不相同, 则继续向下执行。

## 5.5 try-catch试探式结构\*

格式:

try

可能出错的语句

catch

错误处理

end

功能:

可以对错误进行处理，提高计算的可靠性

# MATLAB程序的基本组成结构

%说明

清除命令：清除workspace中的变量和图形（**clear,close**）

定义变量：包括全局变量的声明及参数值的设定

逐行执行命令：指**MATLAB**提供的运算指令或工具箱

... .. 提供的专用命令

控制循环： 包含**for,if then,switch,while**等语句

逐行执行命令

... ..

**end**

绘图命令：将运算结果绘制出来

- 当然更复杂程序还需要调用子程序，或与**simulink**以及其他应用程序结合起来。



# Unit 6 绘图

## 6.1 绘图简介

**MATLAB**提供了丰富的绘图功能

- ❑ `help graph2d`可得到所有画二维图形的命令
- ❑ `help graph3d`可得到所有画三维图形的命令

## 6.2 二维图形绘制

### 1、基本的绘图命令

**plot (x1,y1,option1,x2,y2,option2,...)**

x1, y1给出的数据分别为x, y轴坐标值，option为选项参数，定义了图形曲线的颜色、线型及标示符号，它由一对单引号括起来。

这是plot命令的完全格式，在实际应用中可以根据需要进行简化。比如：

plot(y); plot(x,y); plot(x,y,option)

绘图指令选项参数表

曲线线型		曲线颜色				标记符号			
选项	意义	选项	意义	选项	意义	选项	意义	选项	意义
'-'	实线	'b'	蓝色	'k'	黑色	'*'	星号	'pentagram'	五角星
'--'	虚线	'g'	绿色	'r'	红色	'.'	点号	'o'	圆圈
'::'	点线	'm'	紫色	'y'	黄色	'x'	叉号	'square'	□
'-.'	点划线	'w'	白色	'c'	青色	'v'	▽	'diamond'	◇
'none'	无线					'^'	△	'hexagram'	六角星
						'>'	▷	'<'	◁

也可通过**help plot**命令进行查询

## 2、选择图像

`figure (1) ; figure (2) ; ... ; figure(n)`

打开不同的图形窗口，以便绘制不同的图形。

**3、 grid on:** 在所画出的图形坐标中加入栅格

**grid off:** 除去图形坐标中的栅格

**4、 hold on:** 把当前图形保持在屏幕上不变，同时允许在这个坐标内绘制另外一个图形。

**hold off:** 使新图覆盖旧的图形

## 5、设定轴的范围

**axis** ([xmin xmax ymin ymax])

**axis('equal')**: 将x坐标轴和y坐标轴的单位刻度大小调整为一样。

## 6、文字标示

**text** (x,y,'字符串' )

在图形的指定坐标(x, y)处, 标示单引号括起来的字符串。

**get**('字符串' )

利用鼠标在图形的某一位置标示字符串。

**title** ('字符串' )

在所画图形的最上端显示说明该图形标题的字符串。

**xlabel**('字符串' ), **ylabel**('字符串' )

设置x, y坐标轴的名称。

— 输入特殊的文字需要用反斜杠 (\) 开头。

## 7、**legend** (‘字符串1’,‘字符串2’,...,‘字符串n’)

- 在屏幕上开启一个小视窗，然后依据绘图命令的先后次序，用对应的字符串区分图形上的线。

## 8、**subplot** (mnk)：分割图形显示窗口

m:上下分割个数，n:左右分割个数，k:子图编号

9、**semilogx**: 绘制以x轴为对数坐标（以10为底），y轴为线性坐标的半对数坐标图形。

**semilogy**: 绘制以y轴为对数坐标（以10为底），x轴为线性坐标的半对数坐标图形。

## 10、应用型绘图指令:

可用于数值统计分析或离散数据处理

**bar (x,y)**: 条形图

**hist (y,x)**: 直方图

**stairs (x,y)**: 阶梯形图

**stem (x,y)**: 火柴杆图



## 11、一点补充说明

- 对于图形的属性编辑同样可以通过在图形窗口上直接进行。
- 但图形窗口关闭之后编辑结果不会保存

## 6.3 三维图形绘制

### 1、三维曲线绘制

**plot3 (x, y, z)**

**plot3 (x<sub>1</sub>,y<sub>1</sub>,z<sub>1</sub>,选项1, x<sub>2</sub>,y<sub>2</sub>,z<sub>2</sub>,选项2, ..., x<sub>m</sub>,y<sub>m</sub>,z<sub>m</sub>,选项m )**

其中，选项与二维曲线绘制的完全一致。

## 6.3 三维图形绘制

### 2、三维曲面绘制

```
[x, y]=meshgrid(v1, v2)    %生成网格数据  
z = ..., 如 z = x*y          %计算二元函数的z矩阵  
surf(x,y,z)或mesh(x,y,z)    %mesh绘制网格图, surf绘制表面图
```

# 二维数据曲线图

## 绘制单根二维曲线

### plot函数

基本调用格式为: **plot(x,y)**

其中**x**和**y**为长度相同的向量, 分别用于存储x坐标和y坐标数据。

# 对给定数据绘制图形

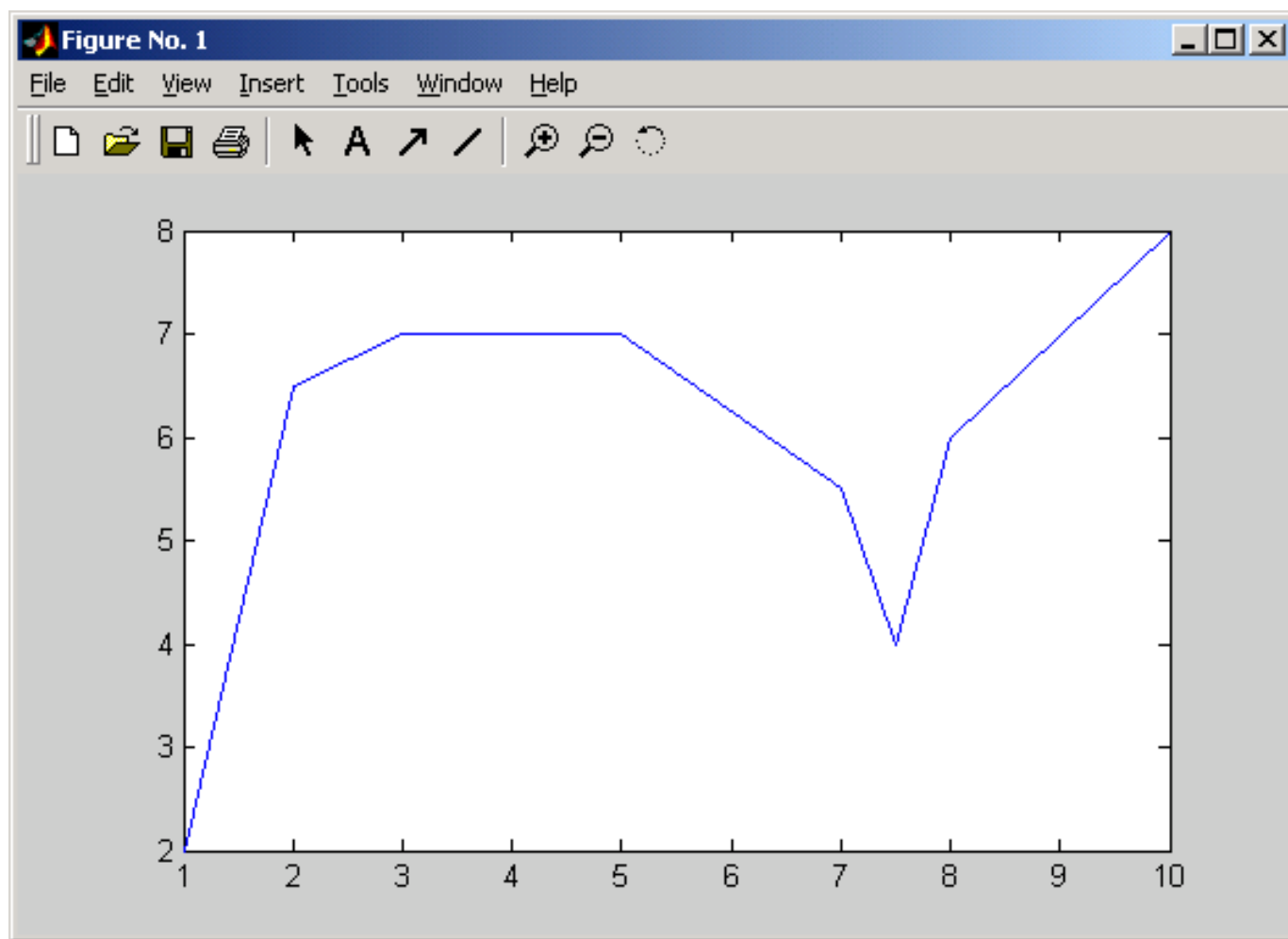
**例6-1：** 给定数据:

x	1	2	3	5	7	7.5	8	10
y	2	6.5	7	7	5.5	4	6	8

程序如下：

```
>> x=[1 2 3 5 7 7.5 8 10];  
>> y=[2 6.5 7 7 5.5 4 6 8];  
>> plot(x,y)
```

一旦命令得以执行，图形窗口随绘图结果打开



## 线条属性的修改

**plot** 命令可以添加线条属性参数来确定:

- 线条的类型.
- 线条的颜色.
- 线条的标记 (markers)

```
plot(x,y,'line specifiers')
```

图形颜色、标记和线型参数表

色彩字符	所定颜色	线型字符	线型格式	标记符号	数据点形式	标记符号	数据点形式
y	黄	-	实线（默认）	.	点	<	左三角形
m	紫	:	点线	o	圆	s	方形
c	青	-.	点划线	x	叉号	d	菱形
r	红	--	虚线	+	加号	h	六角星
g	绿			*	星号	p	五角星
b	蓝			v	下三角形		
w	白			^	上三角形		
k	黑			>	右三角形		



## Plot命令的线型属性修改

- 线型修改选项以字符串形式输入 `plot (x,y,'r')`
- 颜色，线型，标记可以以任何顺序输入 `plot(x,y,'r--d')/plot(x,y,'--r')`
- 所有选项都是可选的，即`plot`命令中可以同时含有一个、两个或三个选项

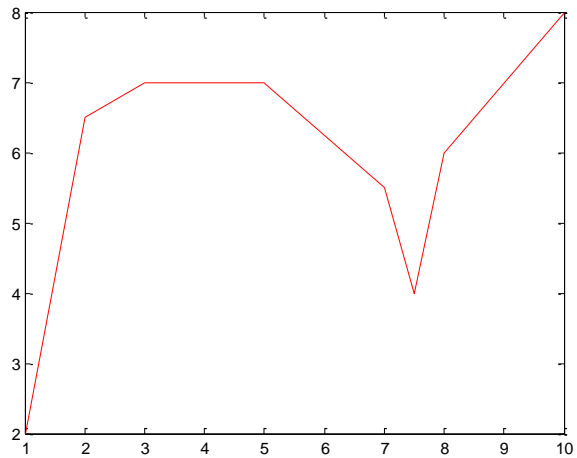
### 例6-2: 修改线型

```
>> x=[1 2 3 5 7 7.5 8 10];  
>> y=[2 6.5 7 7 5.5 4 6 8];  
>> plot(x,y)
```

`plot(x,y)`

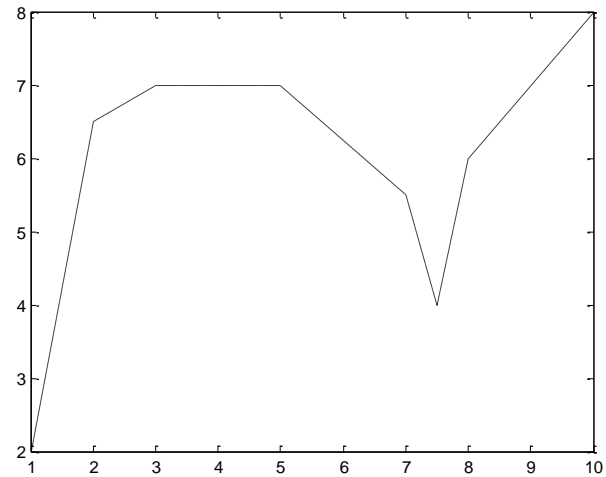
蓝色实线（默认）

**plot(x,y,'r')** 红色实线

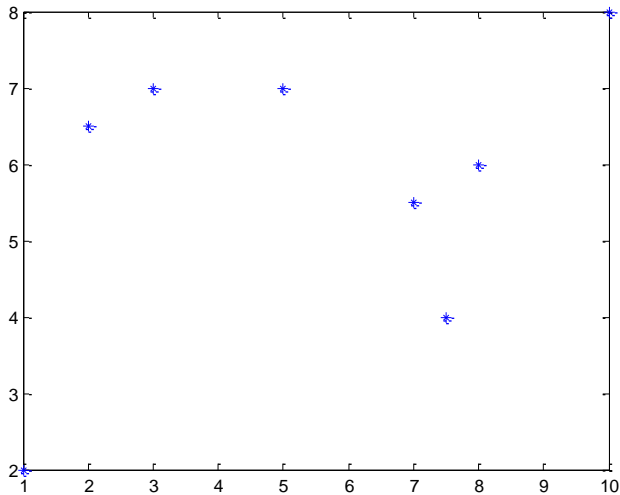


**plot(x,y,'--k')**

黑色虚线.

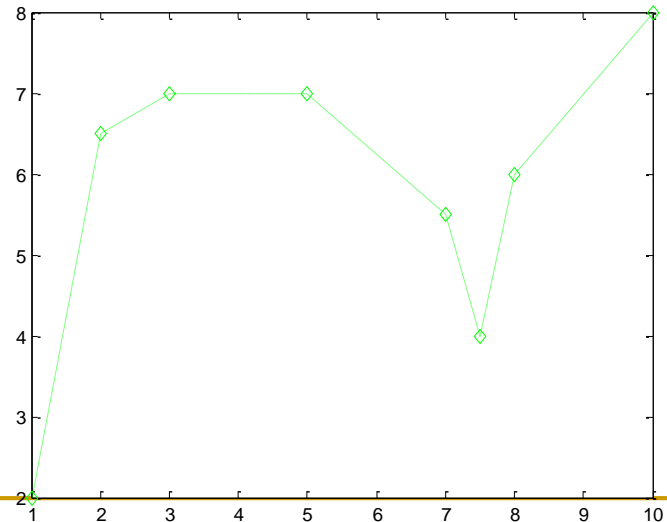


**plot(x,y,'\*')** 用 “\*” 号标记的点，中间无连线



**plot(x,y,'g:d')**

菱形标记的绿色点线



---

**例6-3 在 $0 \leq x \leq 2\pi$ 区间内，绘制曲线**

$$y = 2e^{-0.5x} \cos(4\pi x)$$

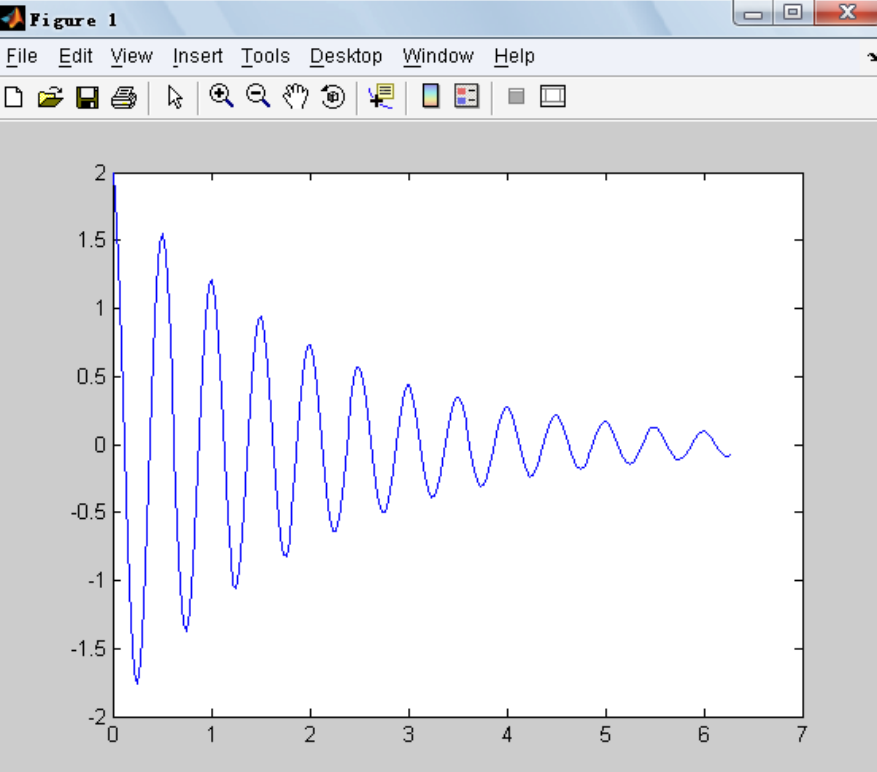
程序如下：

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

```
y=2*exp(-0.5*x).*cos(4*pi*x);
```

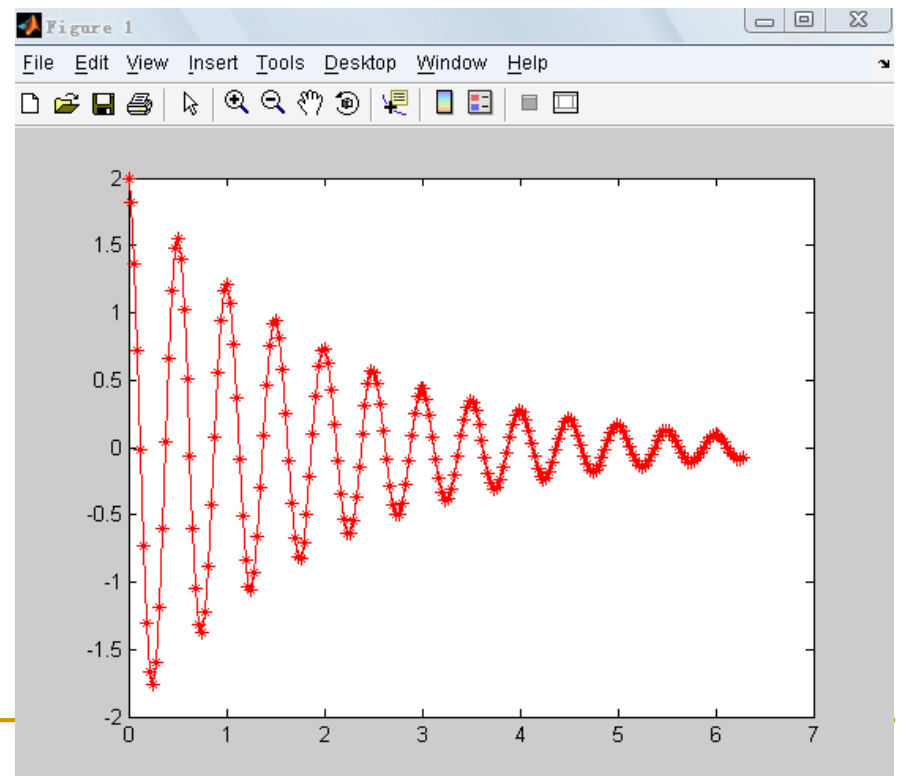
```
plot(x,y)
```

---



**plot(x,y)**

**plot(x,y,'r-\*')**



---

具有两个纵坐标标度的图形

在MATLAB中，如果需要绘制出具有不同纵坐标标度的两个图形，可以使用`plotyy`绘图函数。

调用格式为：`plotyy(x1,y1,x2,y2)`

其中`x1,y1`对应一条曲线，`x2,y2`对应另一条曲线。横坐标的标度相同，纵坐标有两个，左纵坐标用于`x1,y1`数据对，右纵坐标用于`x2,y2`数据对。

---

**例6-6** 用不同标度在同一坐标内绘制曲线 $y_1=0.2e^{-0.5x}\cos(4\pi x)$  和 $y_2=2e^{-0.5x}\cos(\pi x)$ 。

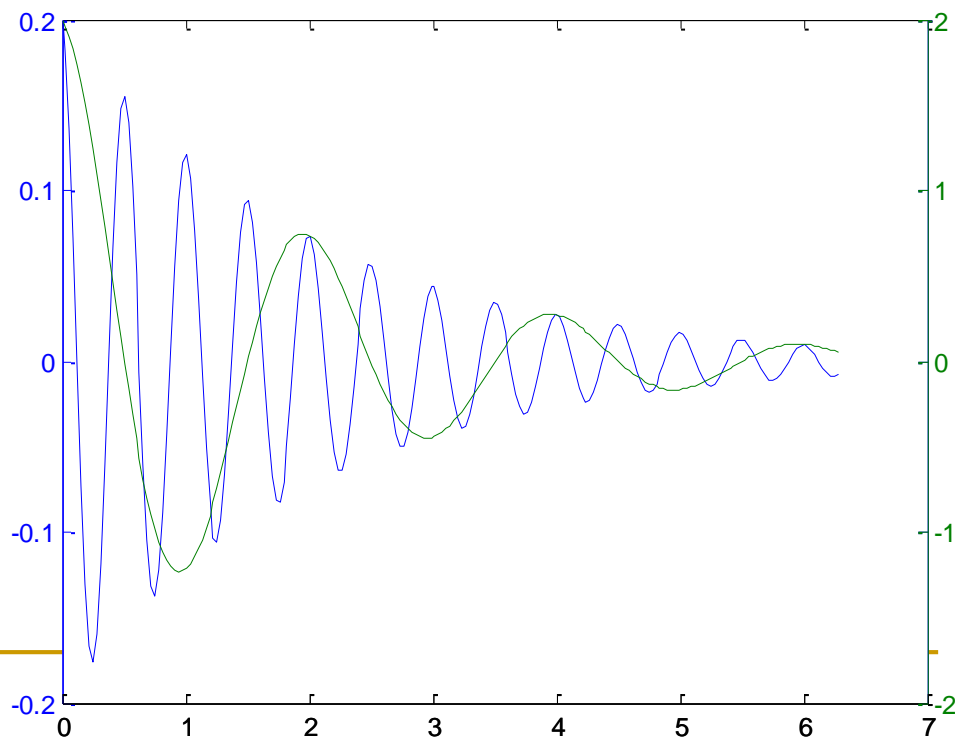
程序如下：

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

```
y1=0.2*exp(-0.5*x).*cos(4*pi*x);
```

```
y2=2*exp(-0.5*x).*cos(pi*x);
```

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



---

## 图形保持

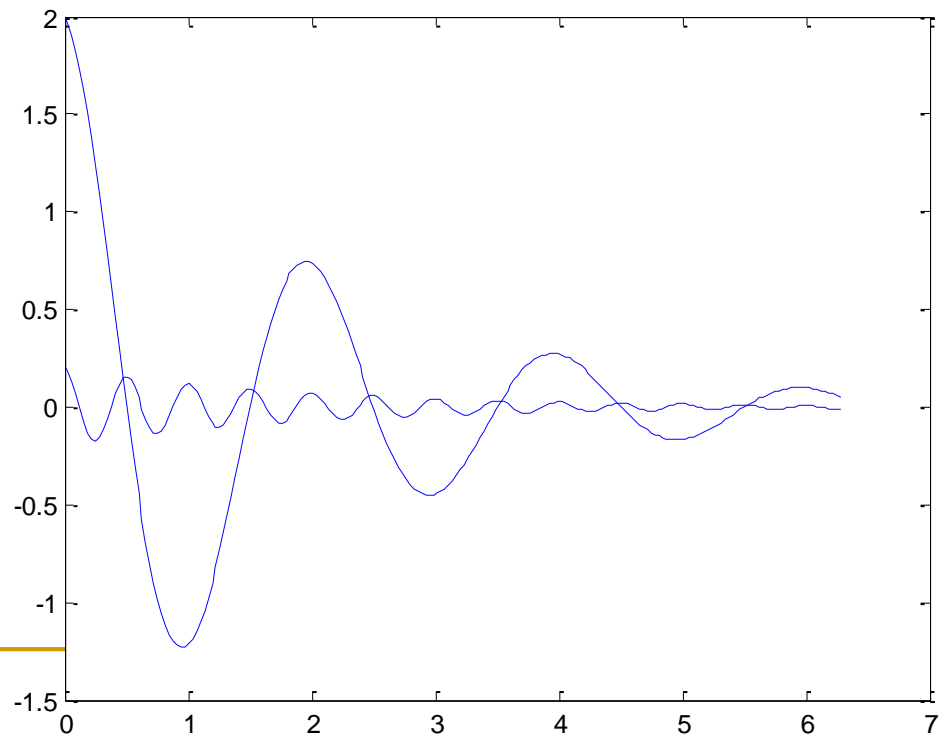
**hold on/off**命令控制是保持原有图形还是刷新原有图形，不带参数的**hold**命令在两种状态之间进行切换。

---

例6-7 采用图形保持，在同一坐标内绘制曲线 $y_1=0.2e^{-0.5x}\cos(4\pi x)$ 和 $y_2=2e^{-0.5x}\cos(\pi x)$ 。

程序如下：

```
x=0:pi/100:2*pi;  
y1=0.2*exp(-0.5*x).*cos(4*pi*x);  
plot(x,y1)  
hold on  
y2=2*exp(-0.5*x).*cos(pi*x);  
plot(x,y2);  
hold off
```





# 图形标注与坐标控制

## 1. 图形标注

有关图形标注函数的调用格式为：

**title(‘图形名称’)**

**xlabel(‘x轴说明’)**

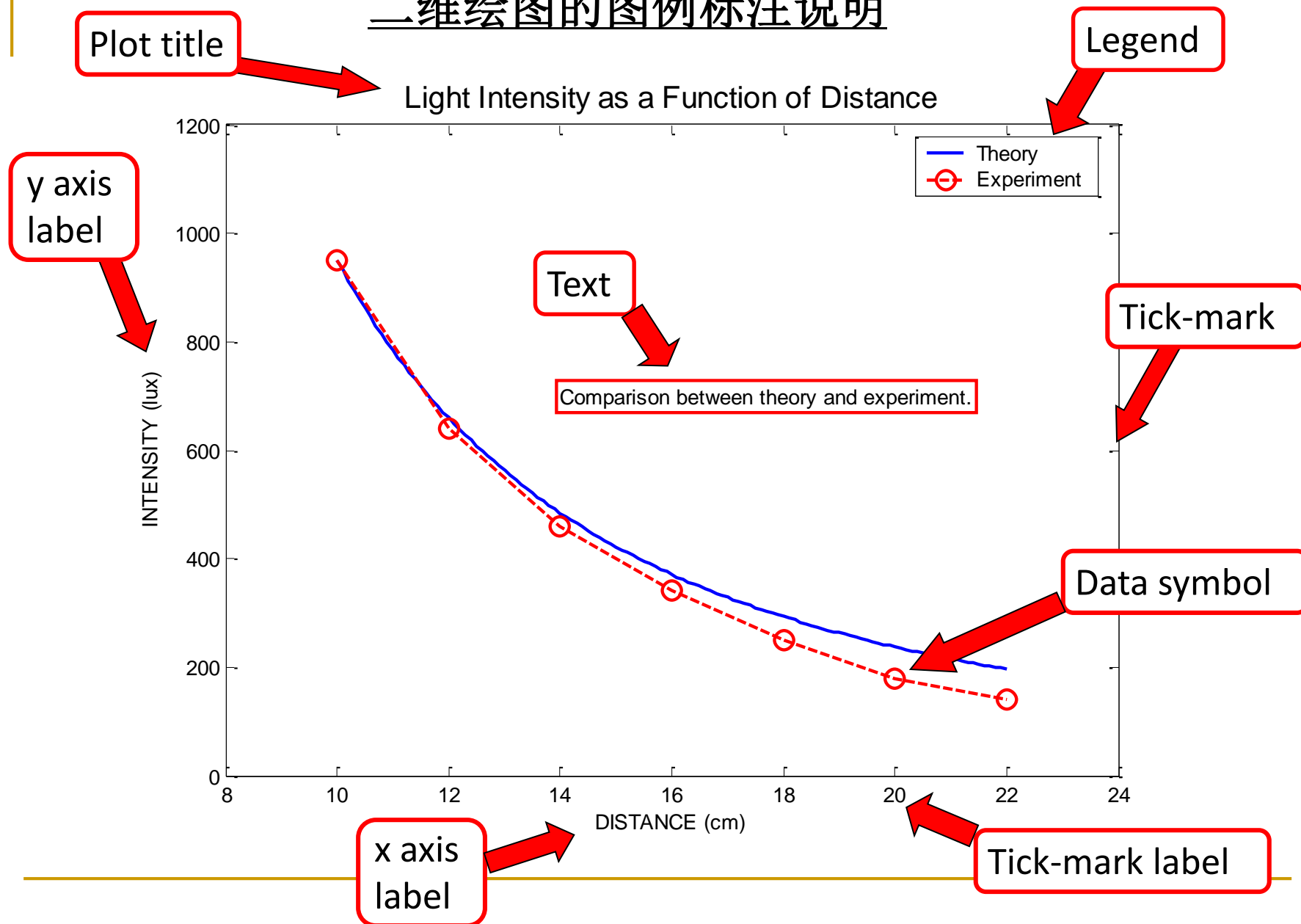
**ylabel(‘y轴说明’)**

**text(x,y, ‘图形说明’)**

**gtext** —— 将标注加到图形任意位置

**legend(‘图例1’, ‘图例2’,...)**

## 二维绘图的图例标注说明

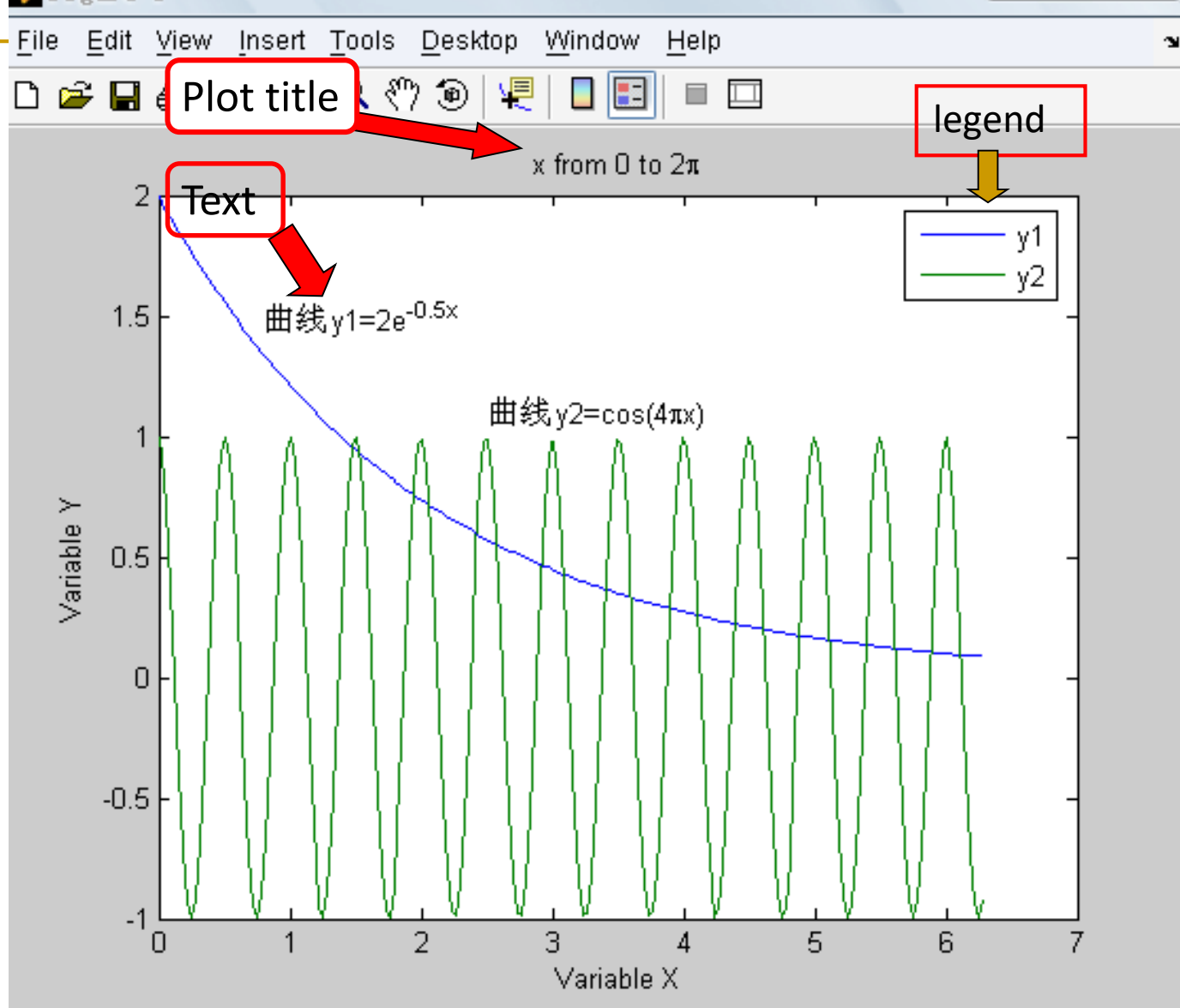


函数中的说明文字，除使用标准的ASCII字符外，还可使用LaTeX格式的控制字符，这样就可以在图形上添加希腊字母、数学符号及公式等内容。例如，  
`text(0.3,0.5,'sin(\omega t+\beta)')`将得到标注效果 $\sin(\omega t+\beta)$ 。

**例6-9** 在 $0 \leq x \leq 2\pi$ 区间内，绘制曲线 $y_1=2e^{-0.5x}$ 和 $y_2=\cos(4\pi x)$ ，并给图形添加图形标注。

程序如下：

```
x=0:pi/100:2*pi;  
y1=2*exp(-0.5*x);  
y2=cos(4*pi*x);  
plot(x,y1,x,y2)  
title('x from 0 to 2{\pi}');           %加图形标题  
xlabel('Variable X');                   %加X轴说明  
ylabel('Variable Y');                   %加Y轴说明  
text(0.8,1.5,'曲线y1=2e^{-0.5x}');    %在指定位置添加图形说明  
text(2.5,1.1,'曲线y2=cos(4{\pi}x)');  
legend('y1',' y2')                     %加图例
```



**gtext** 利用鼠标，交互操作放置文本说明

## 2. 坐标控制

**axis**函数的调用格式为：

**axis([xmin xmax ymin ymax zmin zmax])**

**axis**函数功能丰富，常用的格式还有：

**axis equal**：纵、横坐标轴采用等长刻度。

**axis square**：产生正方形坐标系(缺省为矩形)。

**axis auto**：使用缺省设置。

**axis off**：取消坐标轴。

**axis on**：显示坐标轴。

## 网格线和边框线

给坐标加网格线用**grid**命令来控制。**grid on/off**命令控制是画还是不画网格线，不带参数的**grid**命令在两种状态之间进行切换。

给坐标加边框用**box**命令来控制。**box on/off**命令控制是加还是不加边框线，不带参数的**box**命令在两种状态之间进行切换。

---

## 图形窗口的分割

**subplot函数的调用格式为:**

**subplot(m,n,p)**

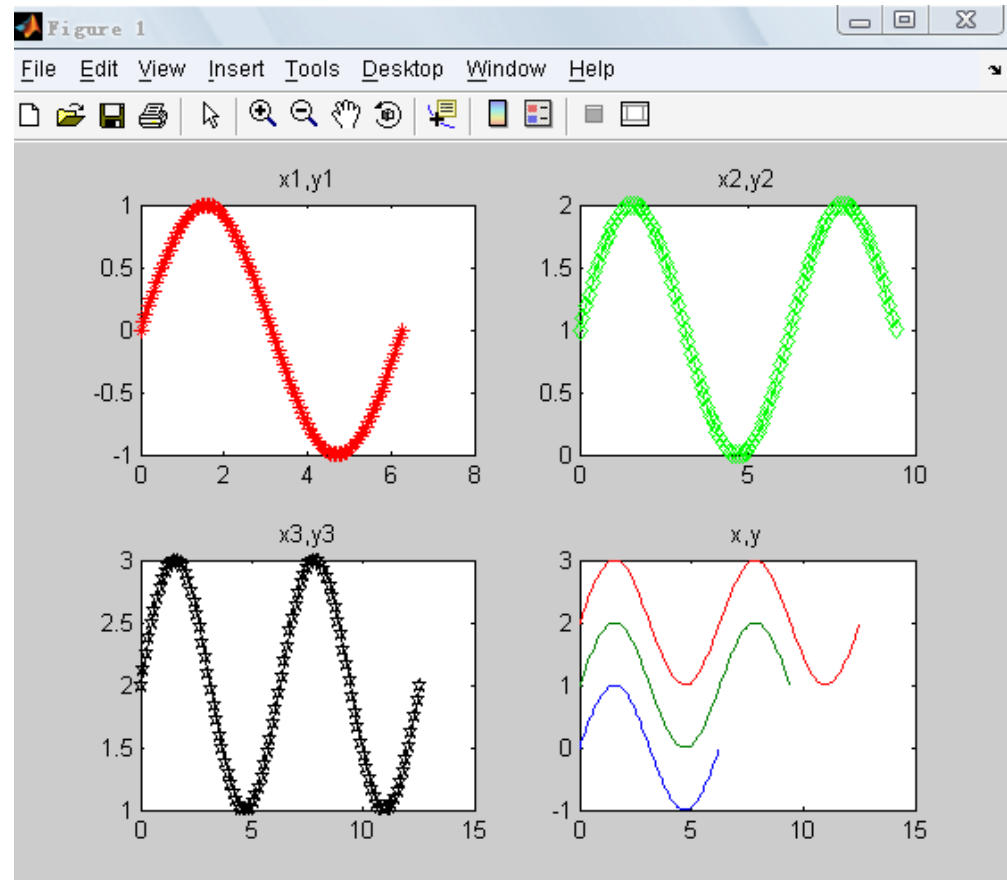
该函数将当前图形窗口分成 $m \times n$ 个绘图区，即每行 $n$ 个，共 $m$ 行，区号按行优先编号，且选择第 $p$ 个绘图区作为当前活动区。在每一个绘图区允许以不同的坐标系单独绘制图形。

**例5-10** 在图形窗口中，以子图形式同时绘制多根曲线。



### 例6-11 分析下列程序绘制的曲线。

```
x1=linspace(0,2*pi,100);  
x2=linspace(0,3*pi,100);  
x3=linspace(0,4*pi,100);  
y1=sin(x1);  
y2=1+sin(x2);  
y3=2+sin(x3);  
x=[x1;x2;x3]';  
y=[y1;y2;y3]';  
subplot(2,2,1),plot(x1,y1,'r-*')  
subplot(2,2,2),plot(x2,y2,'g-*')  
subplot(2,2,3),plot(x3,y3,'k-*')  
subplot(2,2,4),plot(x,y)
```



还可利用**figure**命令开新的图形窗口

## 其他二维图形

### 其他坐标系下的二维数据曲线图

#### 1. 对数坐标图形

MATLAB提供了绘制对数和半对数坐标曲线的函数，调用格式为：

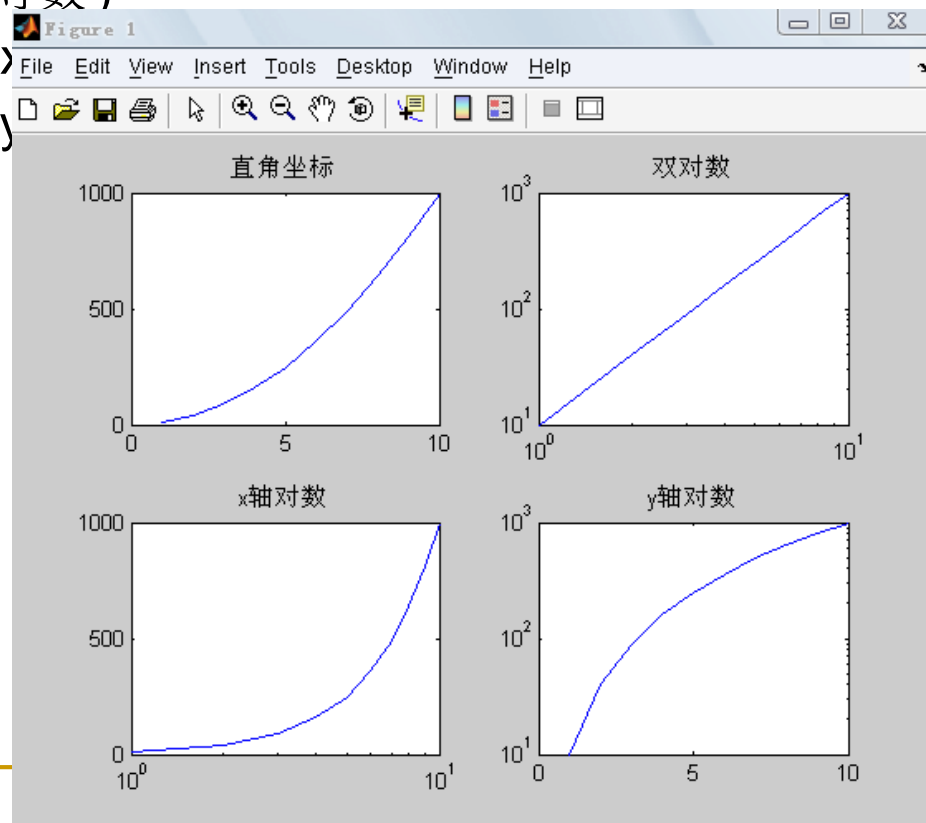
**semilogx**(x1,y1,选项1,x2,y2,选项2,...)

**semilogy**(x1,y1,选项1,x2,y2,选项2,...)

**loglog**(x1,y1,选项1,x2,y2,选项2,...)

## 例6-12 绘制 $y=10x^2$ 的对数坐标图并与直角坐标图进行比较。

- $x=1:10$ ;
- $y=10*(x.^2)$ ;
- `subplot(2,2,1), plot(x,y), title('直角坐标')`
- `subplot(2,2,2), loglog(x,y), title('双对数')`
- `subplot(2,2,3), semilogx(x,y), title('x轴对数')`
- `subplot(2,2,4), semilogy(x,y), title('y轴对数')`



## 2. 极坐标图

**polar**函数用来绘制极坐标图，其调用格式为：

**polar(theta,rho,选项)**

其中**theta**为极坐标极角，**rho**为极坐标矢径，选项的内容与**plot**函数相似。

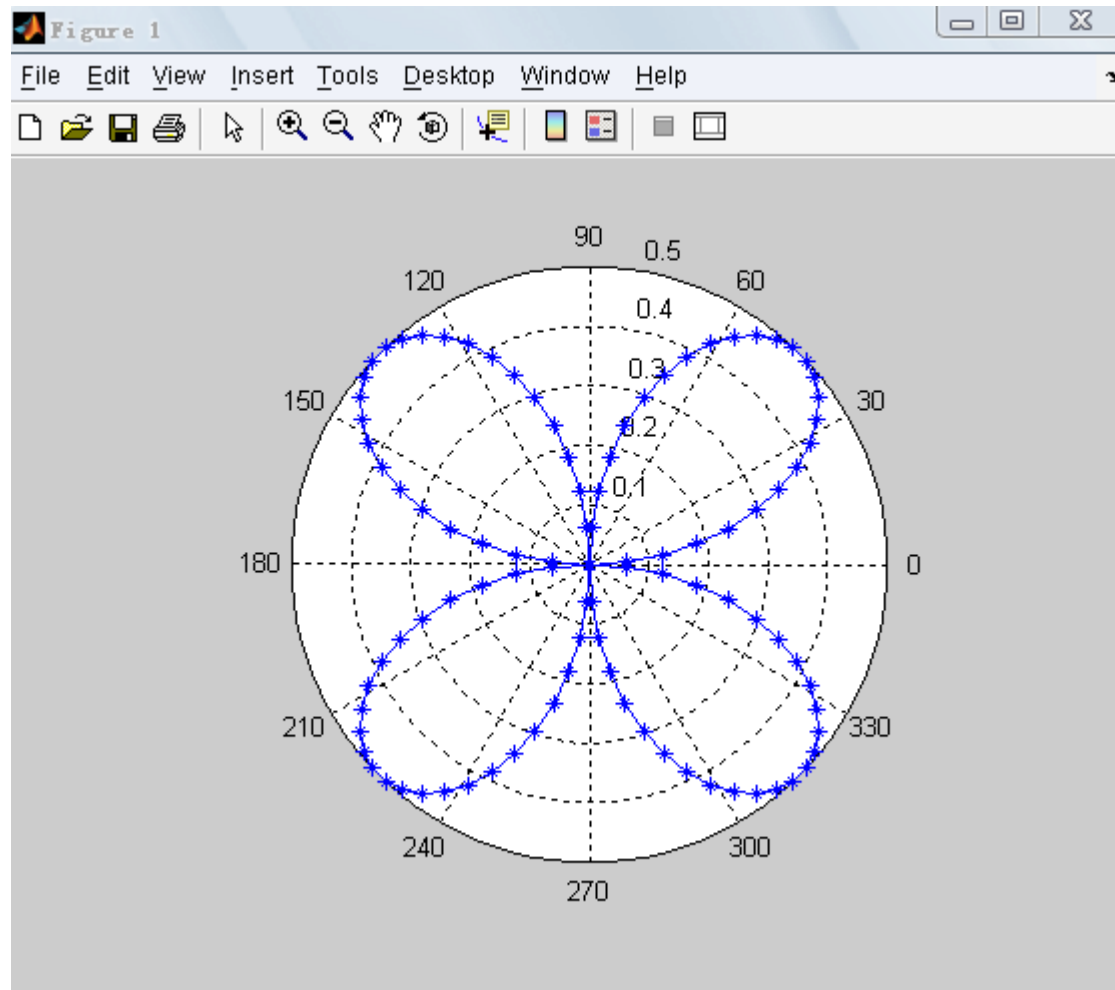
例6-13 绘制 $r=\sin(t)\cos(t)$ 的极坐标图，并标记数据点。

程序如下：

```
t=0:pi/50:2*pi;
```

```
r=sin(t).*cos(t);
```

```
polar(t,r,'-*');
```



## 二维统计分析图

在MATLAB中，二维统计分析图形很多，常见的有条形图、阶梯图、杆图和填充图等，

所采用的函数分别是：

**bar(x,y,选项)**

**stairs(x,y,选项)**

**stem(x,y,选项)**

**fill(x1,y1,选项1,x2,y2,选项2,...)**

**例6-14** 分别以条形图、阶梯图、杆图和填充图形式绘制曲线  $y=2\sin(x)$ 。

程序如下：

```
x=0:pi/10:2*pi;
```

```
y=2*sin(x);
```

```
subplot(2,2,1);bar(x,y,'g');
```

```
title('bar(x,y,'g')');axis([0,7,-2,2]);
```

```
subplot(2,2,2);stairs(x,y,'b');
```

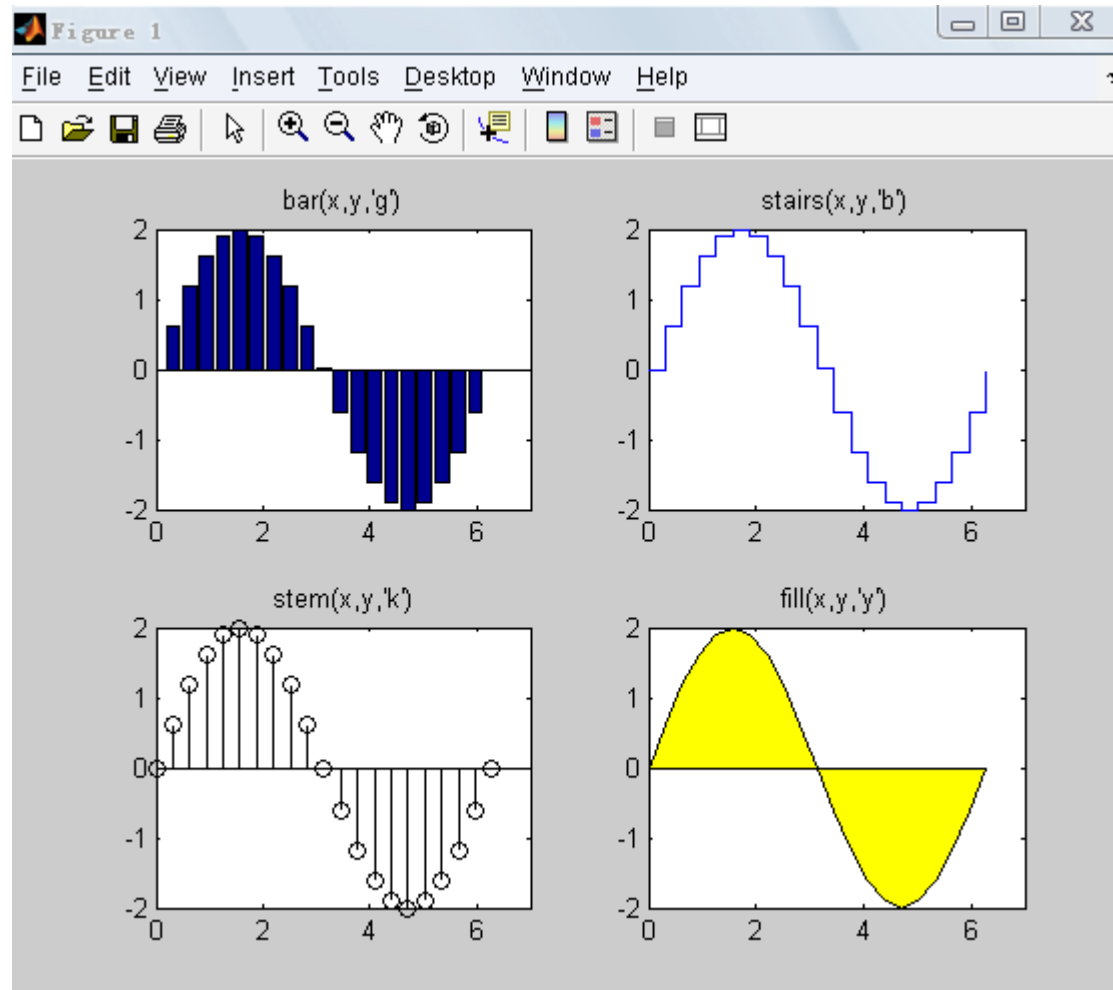
```
title('stairs(x,y,'b')');axis([0,7,-2,2]);
```

```
subplot(2,2,3);stem(x,y,'k');
```

```
title('stem(x,y,'k')');axis([0,7,-2,2]);
```

```
subplot(2,2,4);fill(x,y,'y');
```

```
title('fill(x,y,'y')');axis([0,7,-2,2]);
```





**MATLAB**提供的统计分析绘图函数还有很多，例如，用来表示各元素占总和的百分比的饼图、复数的相量图等等。

### 例6-15 绘制图形：

- (1) 某企业全年各季度的产值(单位：万元)分别为：2347,1827,2043,3025，试用饼图作统计分析。
- (2) 绘制复数的相量图： $7+2.9i$ 、 $2-3i$ 和 $-1.5-6i$ 。

程序如下：

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

```
pie([2347,1827,2043,3025]);
```

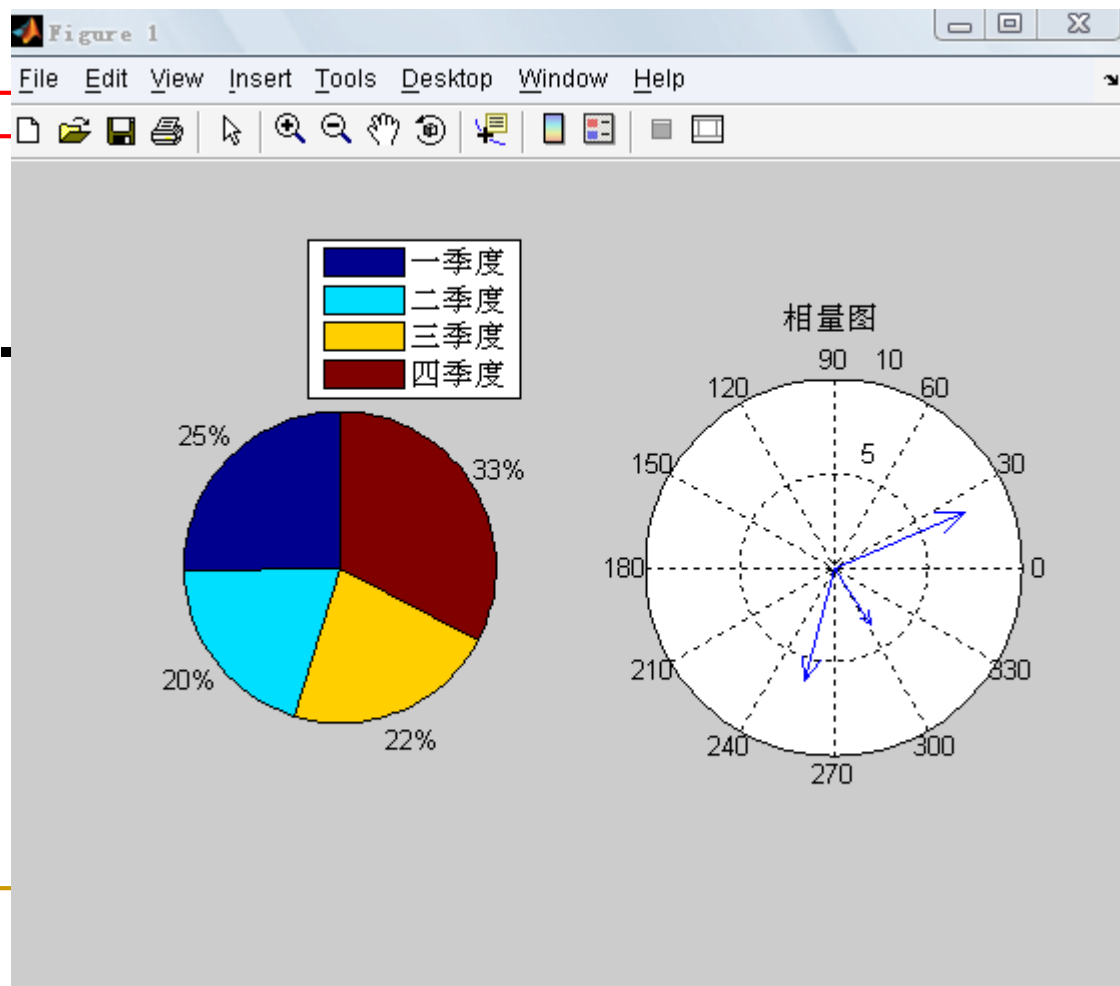
```
title('饼图');
```

```
legend('一季度','二
```

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

```
compass([7+2.9i,2-
```

```
title('相量图');
```



## 符号函数绘图

MATLAB提供了一个`ezplot`函数绘制符号函数图形

**`ezplot(F, [a,b])`**

作函数 $F$ 在 $[a,b]$ 上的图， $F$ 可以是显函数、隐函数或参变量函数，其表示可以是字符串、函数句柄、`Inline` 函数或匿名函数。

$[a,b]$  缺省为 $[-2*\pi, 2*\pi]$

**`ezplot(F, [xmin,xmax,ymin,ymax]):`**

在区间 $x_{\min} < x < x_{\max}$ 和 $y_{\min} < y < y_{\max}$ 绘制 $f(x,y)$  图形。

---

例6-16 符号函数绘图应用举例。

程序如下：

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

```
ezplot('x^2+y^2-9');axis equal
```

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

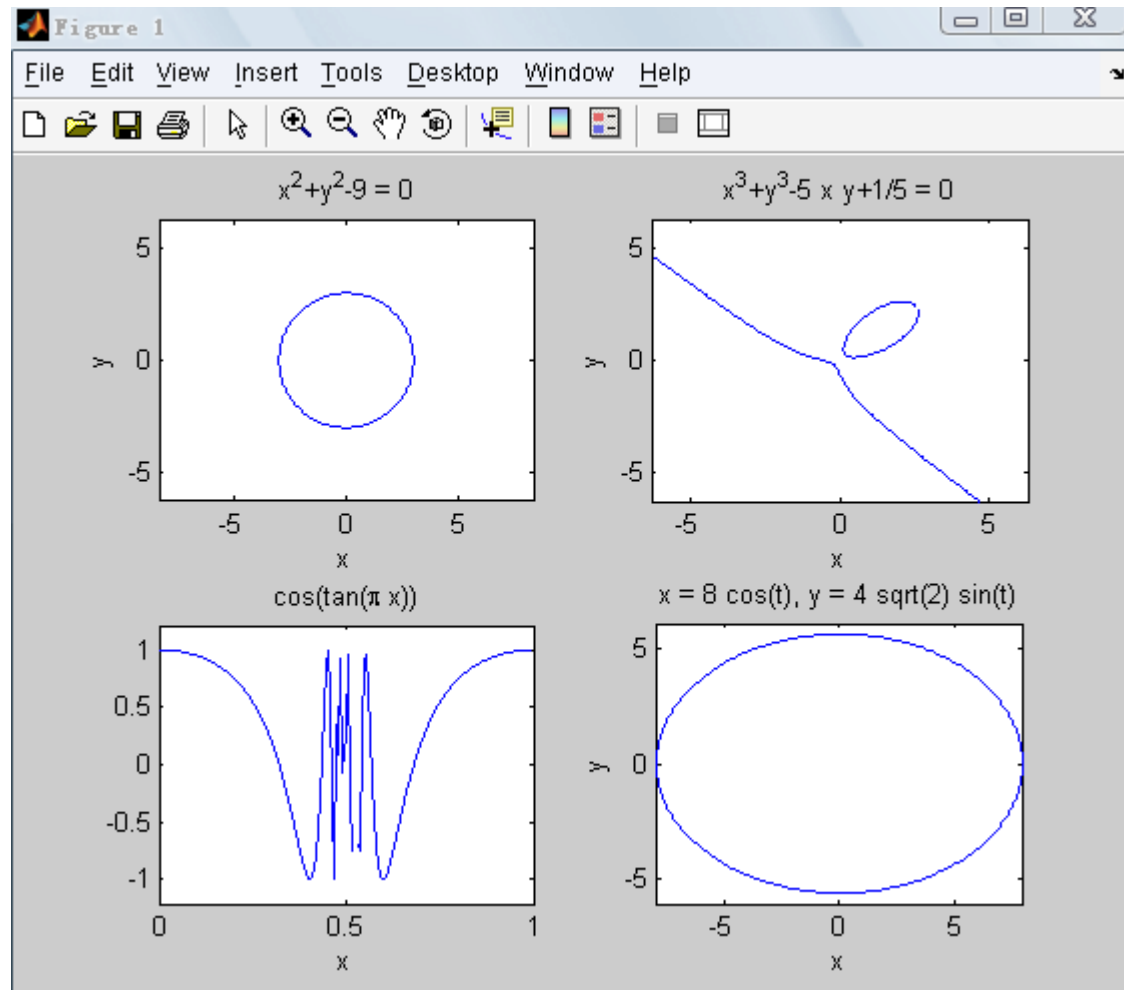
```
ezplot('x^3+y^3-5*x*y+1/5')
```

```
subplot(2,2,3);
```

```
ezplot('cos(tan(pi*x))',[ 0,1])
```

```
subplot(2,2,4);
```

```
ezplot('8*cos(t)','4*sqrt(2)*sin(t)',[0,2*pi])
```



# 三维图形

## 三维曲线

**plot3**函数与**plot**函数用法十分相似，其调用格式为：

**plot3(x1,y1,z1,选项1,x2,y2,z2,选项2,...,xn,yn,zn,选项n)**

其中每一组 $x,y,z$ 组成一组曲线的坐标参数，选项的定义和**plot**函数相同。当 $x,y,z$ 是同维向量时，则 $x,y,z$ 对应元素构成一条三维曲线。当 $x,y,z$ 是同维矩阵时，则以 $x,y,z$ 对应列元素绘制三维曲线，曲线条数等于矩阵列数。

## 例6-17 绘制三维曲线。

程序如下：

```
t=0:pi/100:20*pi;
```

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

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

```
z=t.*sin(t).*cos(t);
```

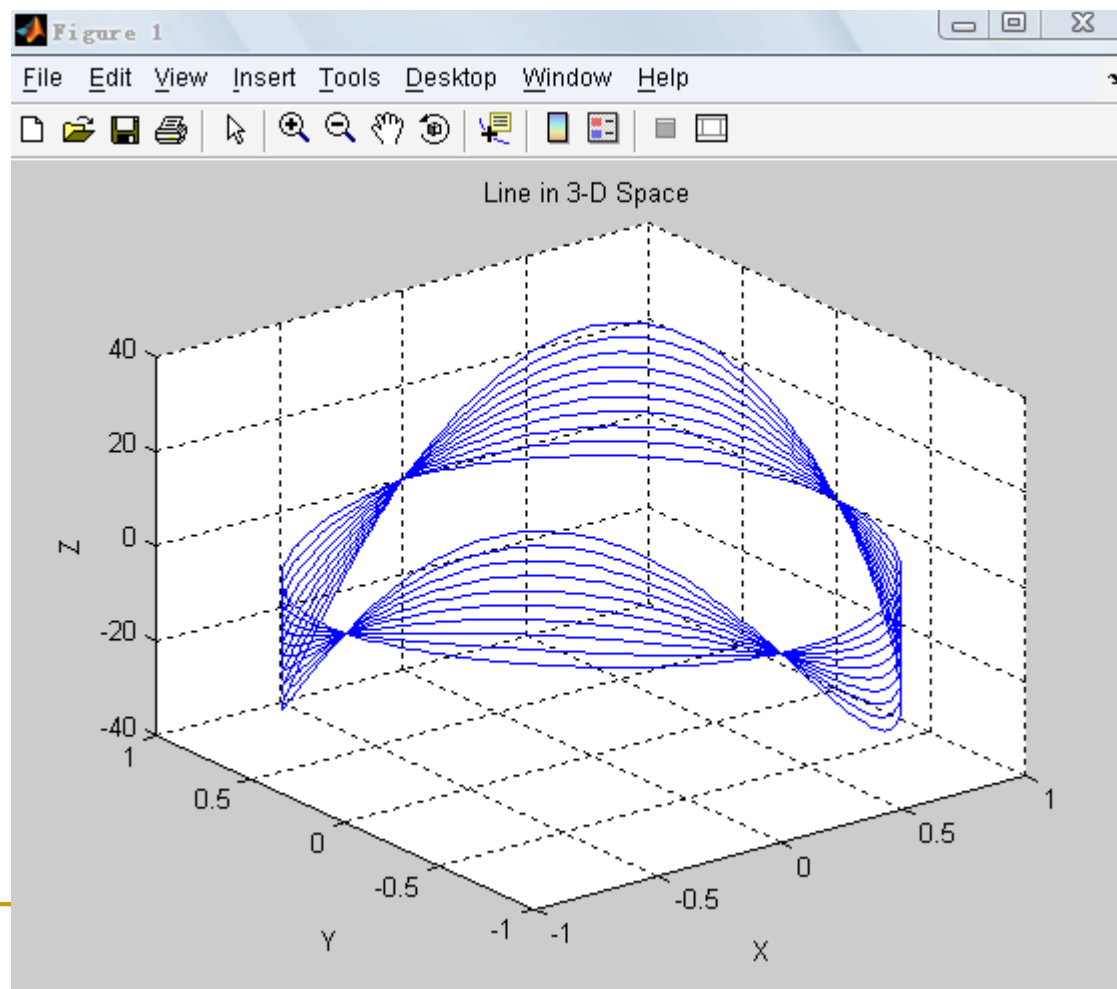
```
plot3(x,y,z);
```

```
title('Line in 3-D Space');
```

```
xlabel('X');ylabel('Y');
```

```
zlabel('Z');
```

```
grid on;
```



## 6.4.2 三维曲面

### 1. 产生三维数据

在MATLAB中，利用meshgrid函数产生平面区域内的网格坐标矩阵。其格式为：

**`x=a:d1:b; y=c:d2:d;`**

**`[X,Y]=meshgrid(x,y);`**

语句执行后，矩阵X的每一行都是向量x，行数等于向量y的元素个数，矩阵Y的每一列都是向量y，列数等于向量x的元素个数。



## 例

- `x=[1,2,3];`
- `y=[4,5,6,7];`
- `[X,Y]=meshgrid(x,y)`

**X =**

<b>1</b>	<b>2</b>	<b>3</b>
<b>1</b>	<b>2</b>	<b>3</b>
<b>1</b>	<b>2</b>	<b>3</b>
<b>1</b>	<b>2</b>	<b>3</b>

**Y =**

<b>4</b>	<b>4</b>	<b>4</b>
<b>5</b>	<b>5</b>	<b>5</b>
<b>6</b>	<b>6</b>	<b>6</b>
<b>7</b>	<b>7</b>	<b>7</b>

## 2. 绘制三维曲面的函数

### **surf**函数和**mesh**函数

**mesh(x,y,z)**画**网格曲面**， $x, y, z$ 是数据矩阵，分别表示数据点的横坐标、纵坐标和函数值，该命令将数据点在空间中描出，并连成网格

- **surf(x,y,z)**画**完整曲面**， $x, y, z$ 是数据矩阵，分别表示数据点的横坐标、纵坐标和函数值，该命令将数据点所表示曲面画出

**例6-18 绘制三维曲面图 $z=\sin(x+\sin(y))-x/10$ 。**

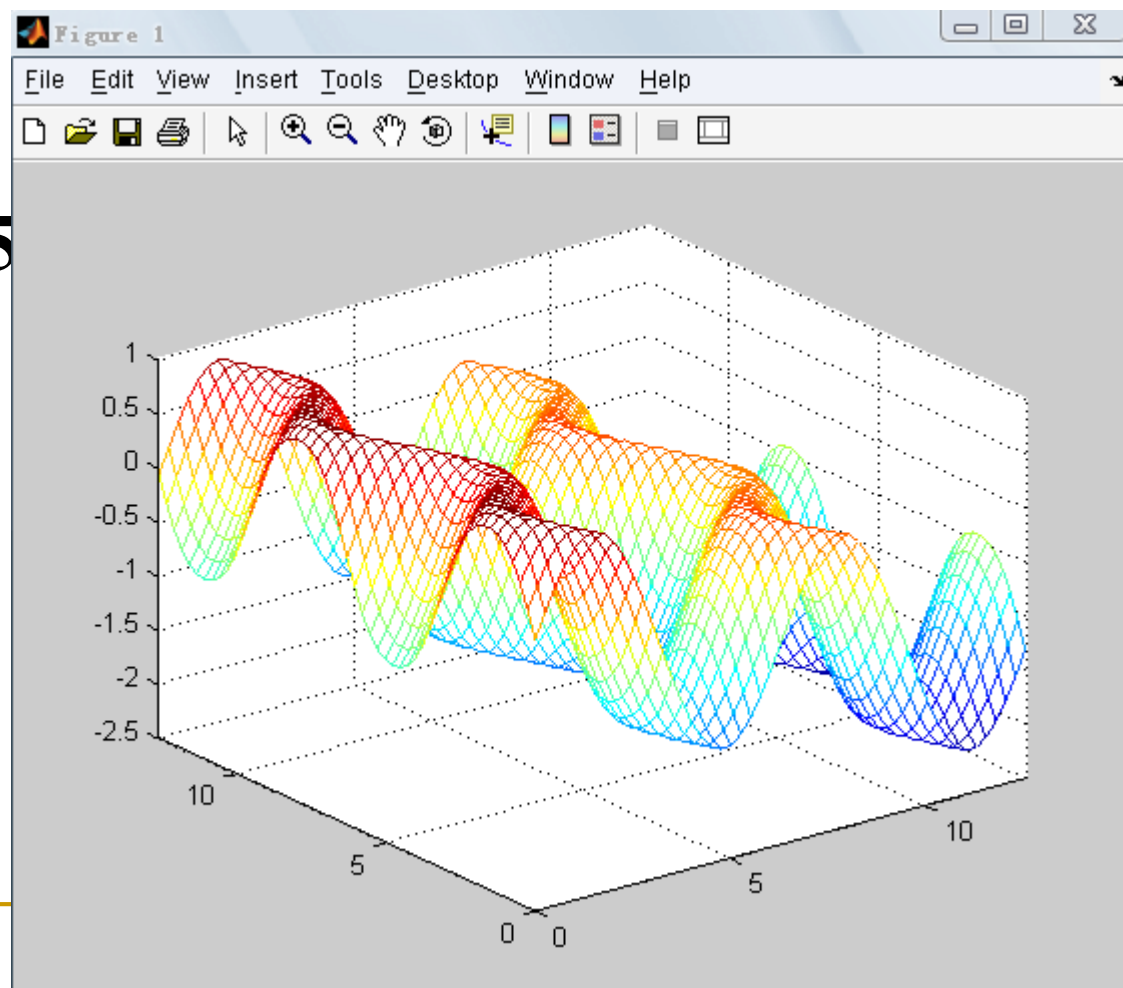
**程序如下：**

**`[x,y]=meshgrid(0:0.25:4*pi);`**

**`z=sin(x+sin(y))-x/10;`**

**`mesh(x,y,z);`**

**`axis([0 4*pi 0 4*pi -2.5`**



---

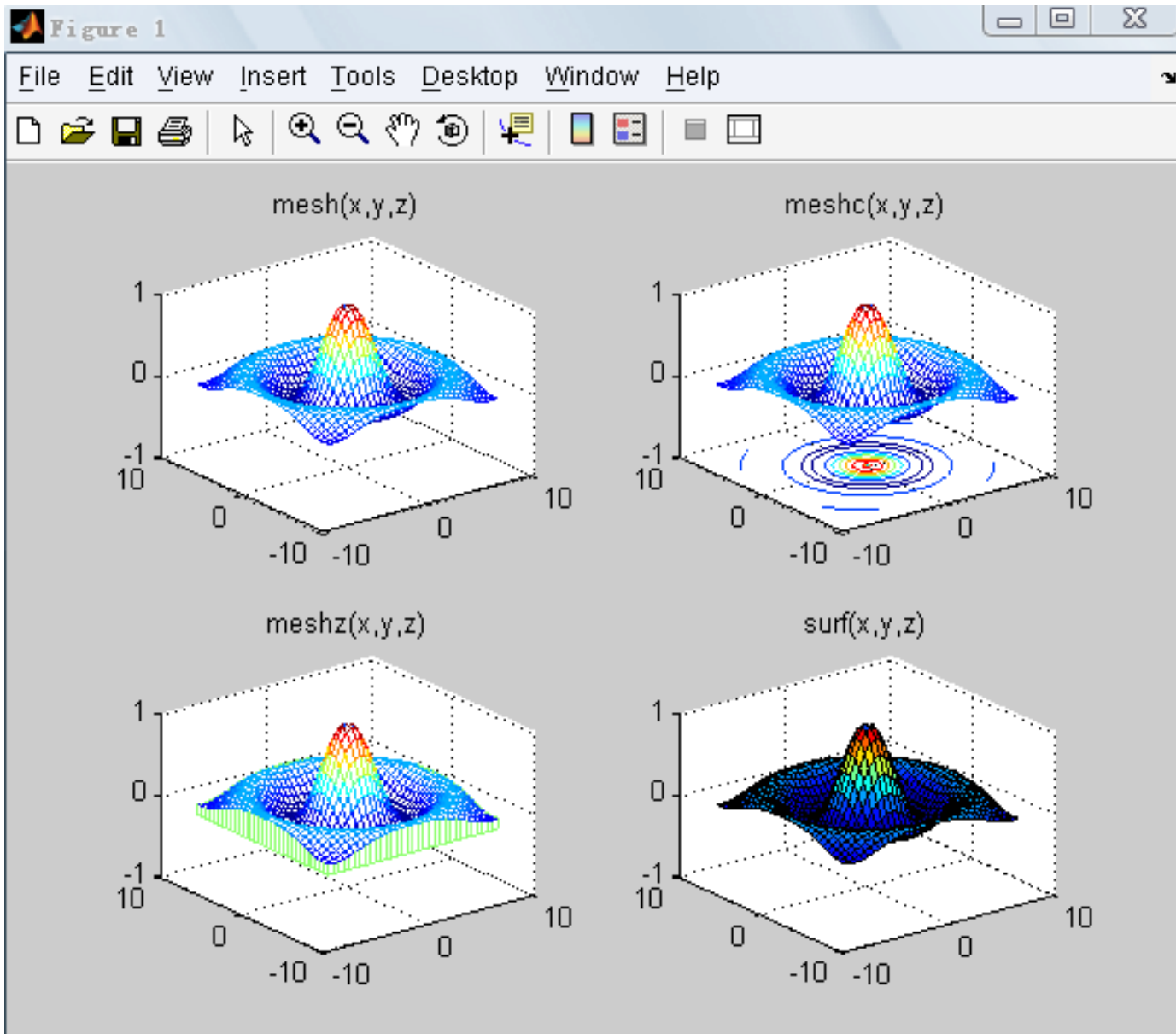
此外，还有带等高线的三维网格曲面函数 **meshc** 和带底座的三维网格曲面函数 **meshz**。  
其用法与 **mesh** 类似，  
不同的是 **meshc** 还在 **xy** 平面上绘制曲面在 **z** 轴方向的等高线  
**meshz** 还在 **xy** 平面上绘制曲面的底座。

---

**例6-19** 在xy平面内选择区域 $[-8,8] \times [-8,8]$ ，绘制4种三维曲面图。

程序如下：

```
[x,y]=meshgrid(-8:0.5:8);  
z=sin(sqrt(x.^2+y.^2))./sqrt(x.^2+y.^2+eps);  
subplot(2,2,1);  
mesh(x,y,z);title('mesh(x,y,z)')  
subplot(2,2,2);  
meshc(x,y,z);title('meshc(x,y,z)')  
subplot(2,2,3);  
meshz(x,y,z);title('meshz(x,y,z)')  
subplot(2,2,4);  
surf(x,y,z);title('surf(x,y,z)')
```



### 3. 标准三维曲面

球面sphere函数的调用格式为：

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

三维柱面 cylinder函数的调用格式为：

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

MATLAB还有一个peaks 函数，称为多峰函数，常用于三维曲面的演示。

例6-20 绘制标准三维曲面图形。

程序如下：

```
t=0:pi/20:2*pi;
```

```
[x,y,z]=cylinder(2+sin(t),30);
```

```
subplot(2,2,1);surf(x,y,z);
```

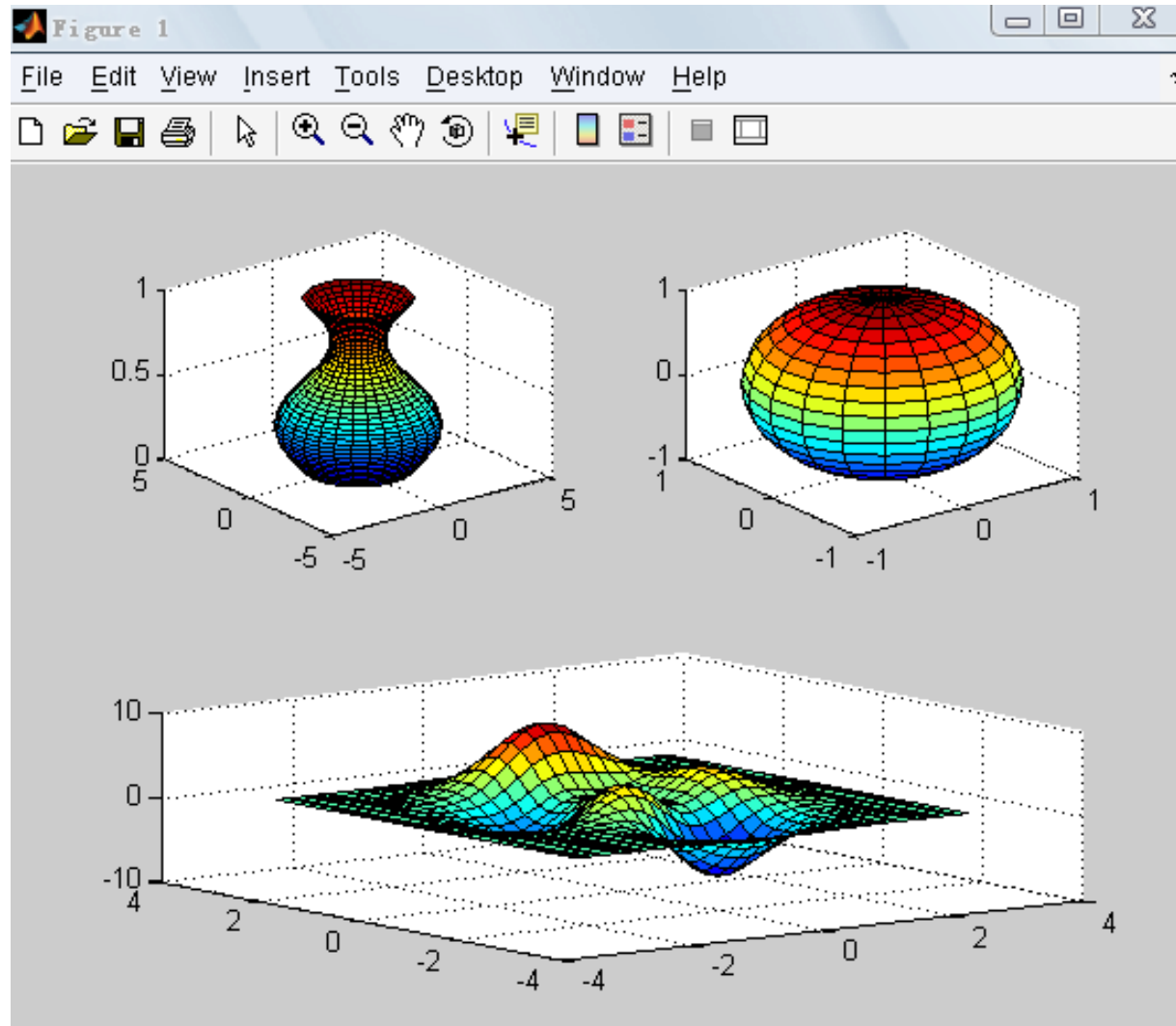
```
[x,y,z]=sphere;
```

```
subplot(2,2,2);surf(x,y,z);
```

```
[x,y,z]=peaks(30);
```

```
subplot(2,1,2);surf(x,y,z);
```





### 6.4.3 其他三维图形

在介绍二维图形时，曾提到条形图、杆图、饼图和填充图等特殊图形，它们还可以以三维形式出现，使用的函数分别是**bar3**、**stem3**、**pie3**和**fill3**。

**bar3**函数绘制三维条形图，常用格式为：

**bar3(y)**

**bar3(x,y)**

---

**stem3**函数绘制离散序列数据的三维杆图，常用格式为：

**stem3(z)**

**stem3(x,y,z)**

**pie3**函数绘制三维饼图，常用格式为：

**pie3(x)**

**fill3**函数等效于三维函数**fill**，可在三维空间内绘制出填充过的多边形，常用格式为：

**fill3(x,y,z,c)**

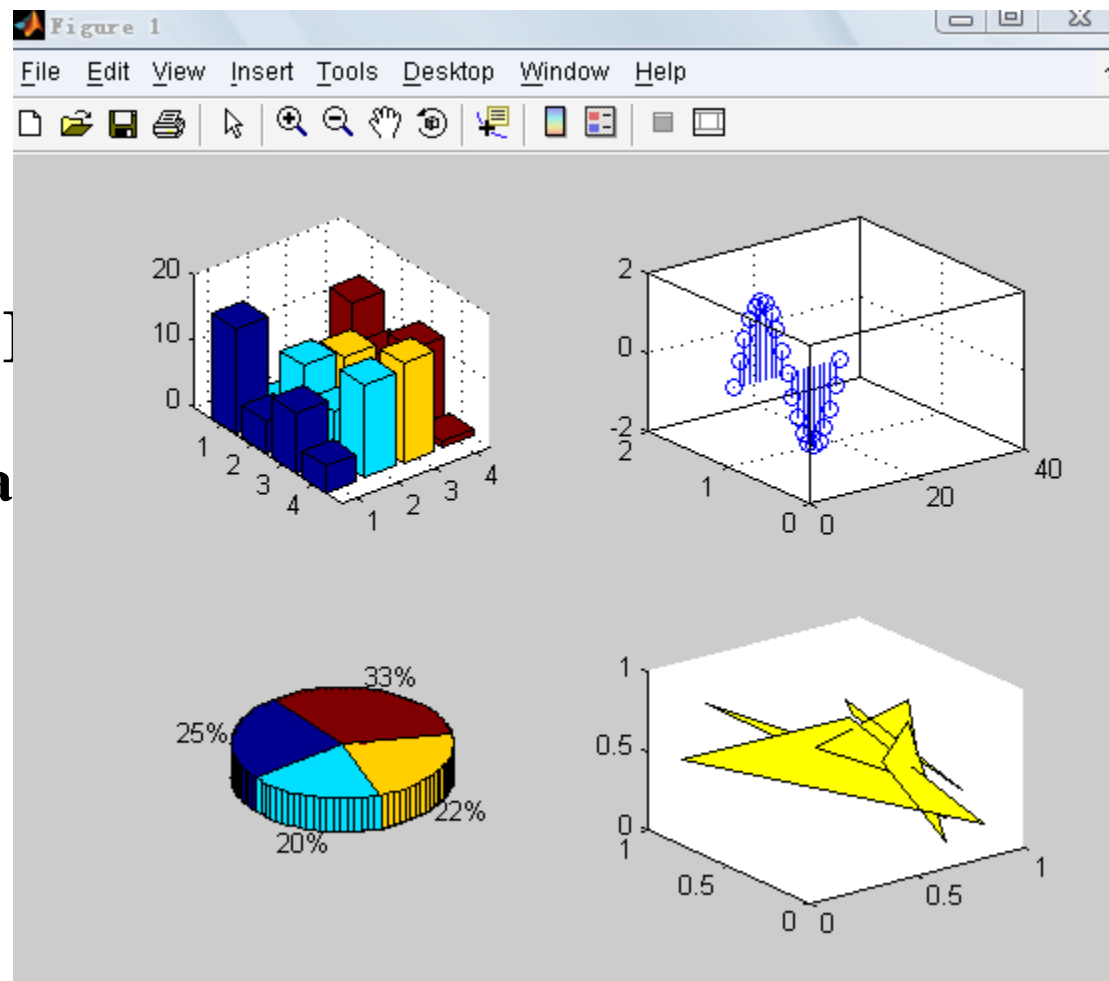
---

### 例6-21 绘制三维图形：

- (1) 绘制魔方阵的三维条形图。
- (2) 以三维杆图形式绘制曲线 $y=2\sin(x)$ 。
- (3) 已知 $x=[2347,1827,2043,3025]$ ，绘制饼图。
- (4) 用随机的顶点坐标值画出五个黄色三角形。

程序如下：

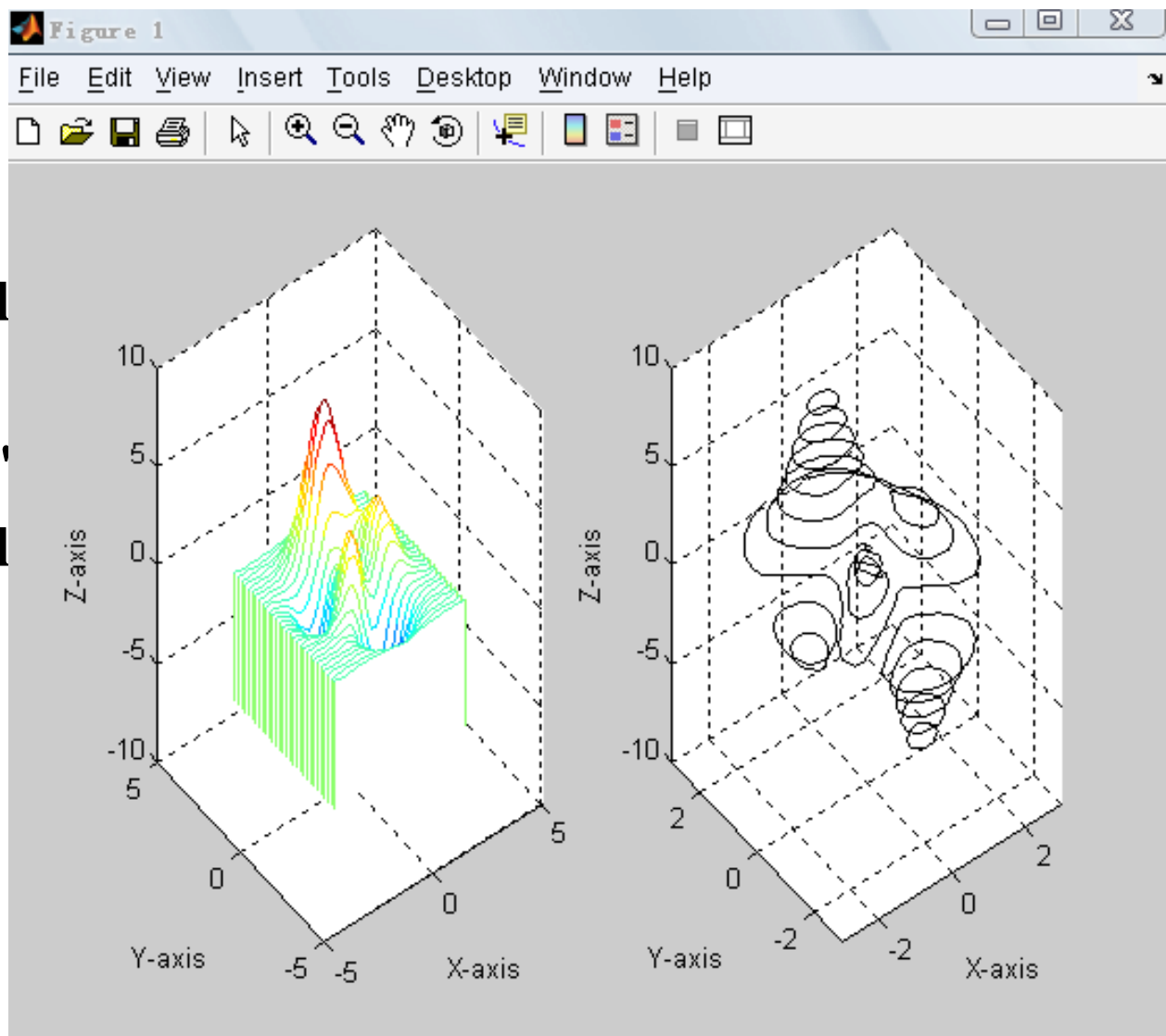
```
subplot(2,2,1);  
bar3(magic(4))  
subplot(2,2,2);  
y=2*sin(0:pi/10:2*pi);  
stem3(y);  
subplot(2,2,3);  
pie3([2347,1827,2043,3025])  
subplot(2,2,4);  
fill3(rand(3,5),rand(3,5),ra
```



## 例6-22 绘制多峰函数的瀑布图和等高线图。

程序如下：

```
subplot(1,2,1);  
[X,Y,Z]=peaks(30);  
waterfall(X,Y,Z)  
xlabel('X-axis'),ylabel  
subplot(1,2,2);  
contour3(X,Y,Z,12,'k')  
xlabel('X-axis'),ylabel
```



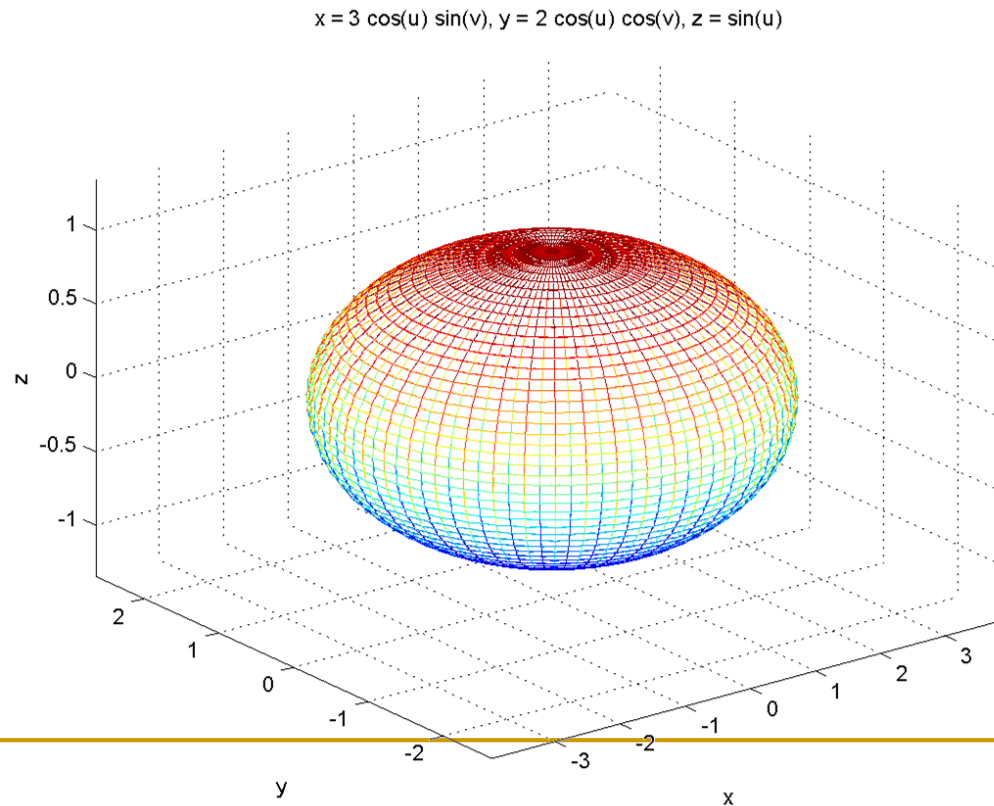
## ■ 符号表达式进行三维绘图

● ezmesh, ezsurf

■ 例6-23 绘制椭球面

$$\begin{cases} x = 3 \cos u \sin v \\ y = 2 \cos u \cos v \\ z = \sin u \end{cases}$$

```
u=-2*pi:0.1:2*pi;  
v=-2*pi:0.1:2*pi;  
ezmesh('3*cos(u)*sin(v)','2*cos(u)*cos(v)','sin(u)',[-2*pi,2*pi,-2*pi,2*pi])
```

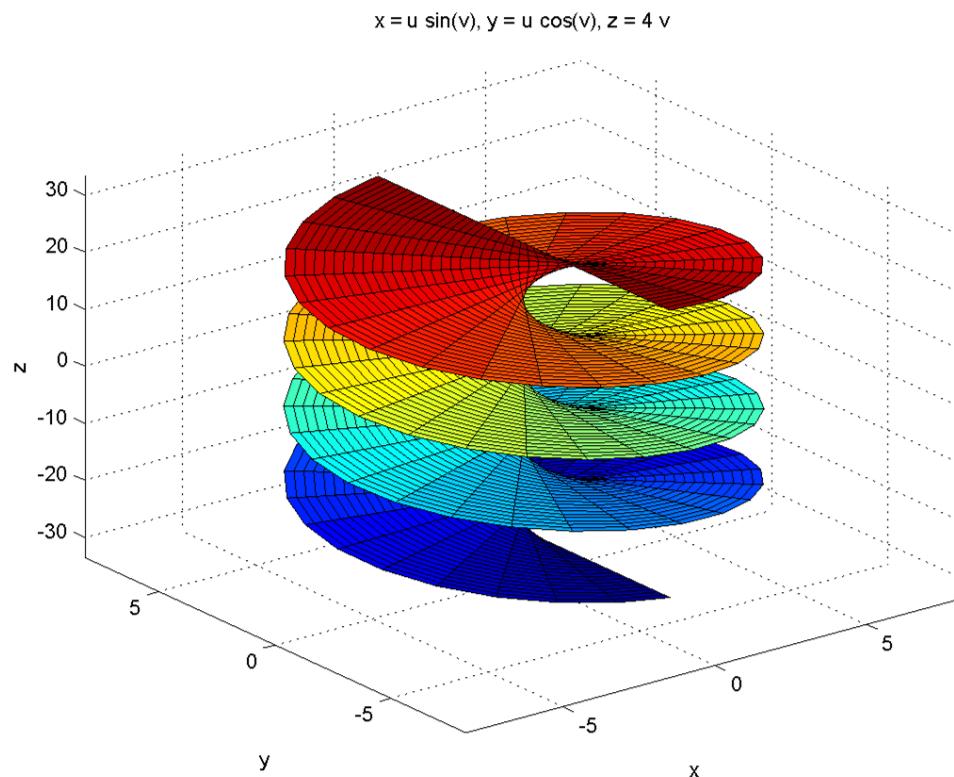




■ 例6-24: 绘制正螺面

$$\begin{cases} x = u \sin v \\ y = u \cos v \\ z = 4v \end{cases}$$

**u,v在-2pi至2pi  
命令为ezsurf**

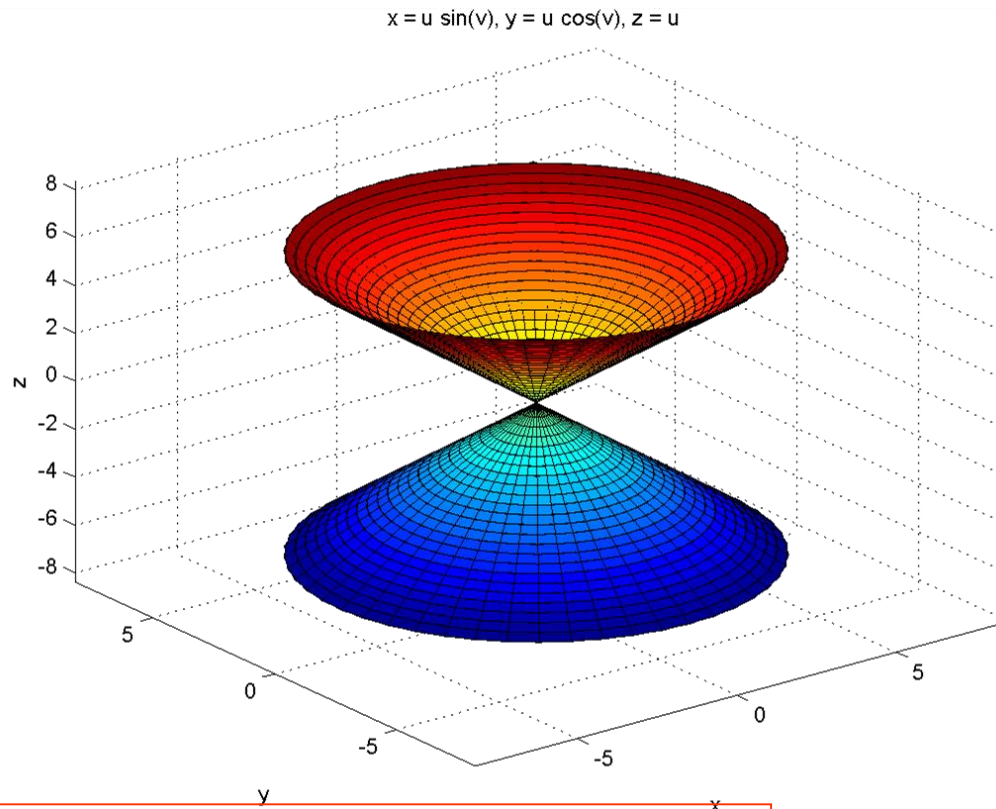


**ezsurf('u\*sin(v)', 'u\*cos(v)', '4\*u', [-2\*pi, 2\*pi, -2\*pi, 2\*pi])**

■ 例6-25: 绘制圆锥面

$$\begin{cases} x = u \sin v \\ y = u \cos v \\ z = u \end{cases}$$

**u,v在-2pi至2pi  
命令为ezsurf**



**ezsurf('u\*sin(v)', 'u\*cos(v)', 'u', [-2\*pi, 2\*pi, -2\*pi, 2\*pi])**

# 小结：MATLAB程序的基本设计原则

- 1、%后面的内容是程序的注解，要善于运用注解使程序更具可读性。
- 2、养成在主程序开头用clear指令清除变量的习惯，以消除工作空间中其他变量对程序运行的影响。但注意在子程序中不要用clear。
- 3、参数值要集中放在程序的开始部分，以便维护。要充分利用MATLAB工具箱提供的指令来执行所要进行的运算，在语句行之后输入分号使其及中间结果不在屏幕上显示，以提高执行速度。

# 小结：MATLAB程序的基本设计原则

- 4、程序尽量模块化，也就是采用主程序调用子程序的方法，将所有子程序合并在一起执行全部的操作。
- 5、充分利用Debugger来进行程序的调试（设置断点、单步执行、连续执行），并利用其他工具箱或图形用户界面（GUI）的设计技巧，将设计结果集成到一起。
- 6、设置好MATLAB的工作路径，以便程序运行。