《数值计算方法》

MATLAB软件介绍

讲授人：杨 程

时　间：2021-2022学年
秋季学期

Email: yang_cheng@nun.edu.cn

目录

MATLAB软件介绍

PART 1

MATLAB的概述

1、简介

Matlab软件是一个功能非常强大的数学软件。 包括：科学计算、符号计算、图形处理等功能。

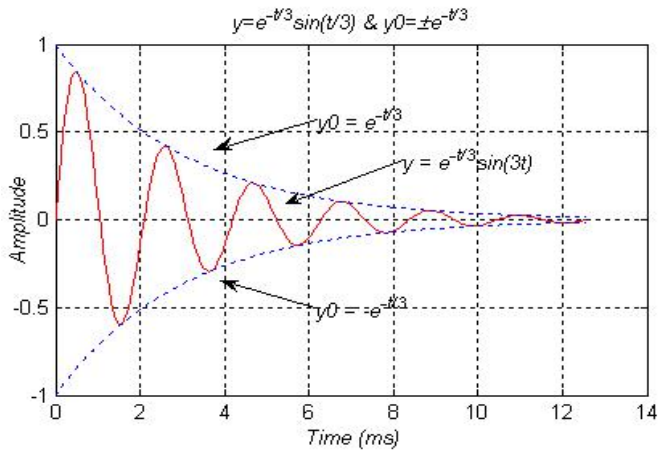
Matlab是一种类似于Fortran和C的一种语言。虽然Matlab的计算远慢于Fortran和C，但是它方便且易学易用。

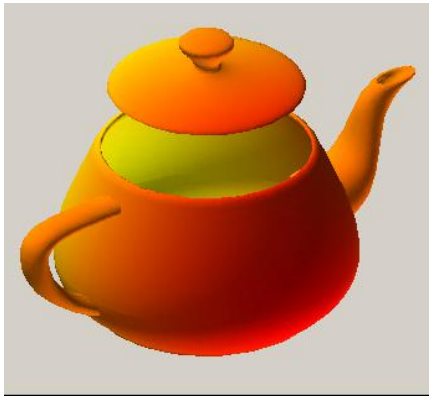
2、MATLAB编程语言的特点

- 语法规则简单。尤其内定的编程规则，与其他编程语言（如C、Fortran等）相比更接近于常规数学表示。对于数组变量的使用，不需类型声明，无需事先申请内存空间。
- MATLAB基本的语言环境提供了数以千计的计算函数，极大的提高了用户的编程效率。如，一个fft函数即可完成对指定数据的快速傅里叶变换，这一任务如果用C语言来编程实现的话，至少要用几十条C语言才能完成。
- MATLAB是一种脚本式的解释型语言，无论是命令、函数或变量，只要在命令窗口的提示符下键入，并“回车（Enter）”，MATLAB都予以解释执行。
- 平台无关性（可移植性）。MATLAB软件可以运行在很多不同的计算机系统平台上，如Windows Me/NT/2000/XP、很多不同版本的UNIX以及Linux。无论你在哪一个平台上编写的程序都可以运行在其它平台上，对于MATLAB数据文件也一样，是平台无关的。极大保护了用户的劳动、方便了用户。其绘图功能也是平台无关的。无论任何系统平台，只要MATLAB能够运行，其图形功能命令就能正常运行。

因此，MATLAB是一个简单易用、功能强大的高效编程语言。

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





- 语言简单、内涵丰富
  - 语言及其书写形式非常接近于常规数学书写形式；
  - 其操作和功能函数指令就是常用的计算机和数学书上的一些简单英文单词表达的，如：help、clear等；
  - 完备的帮助系统，易学易用。
- 扩充能力、可开发能力较强
  - MATLAB完全成了一个开放的系统
  - 用户可以开发自己的工具箱
  - 可以方便地与Fortran、C等语言接口
- 编程易、效率高
  - Matlab以数组为基本计算单元
  - 具有大量的算法优化的功能函数

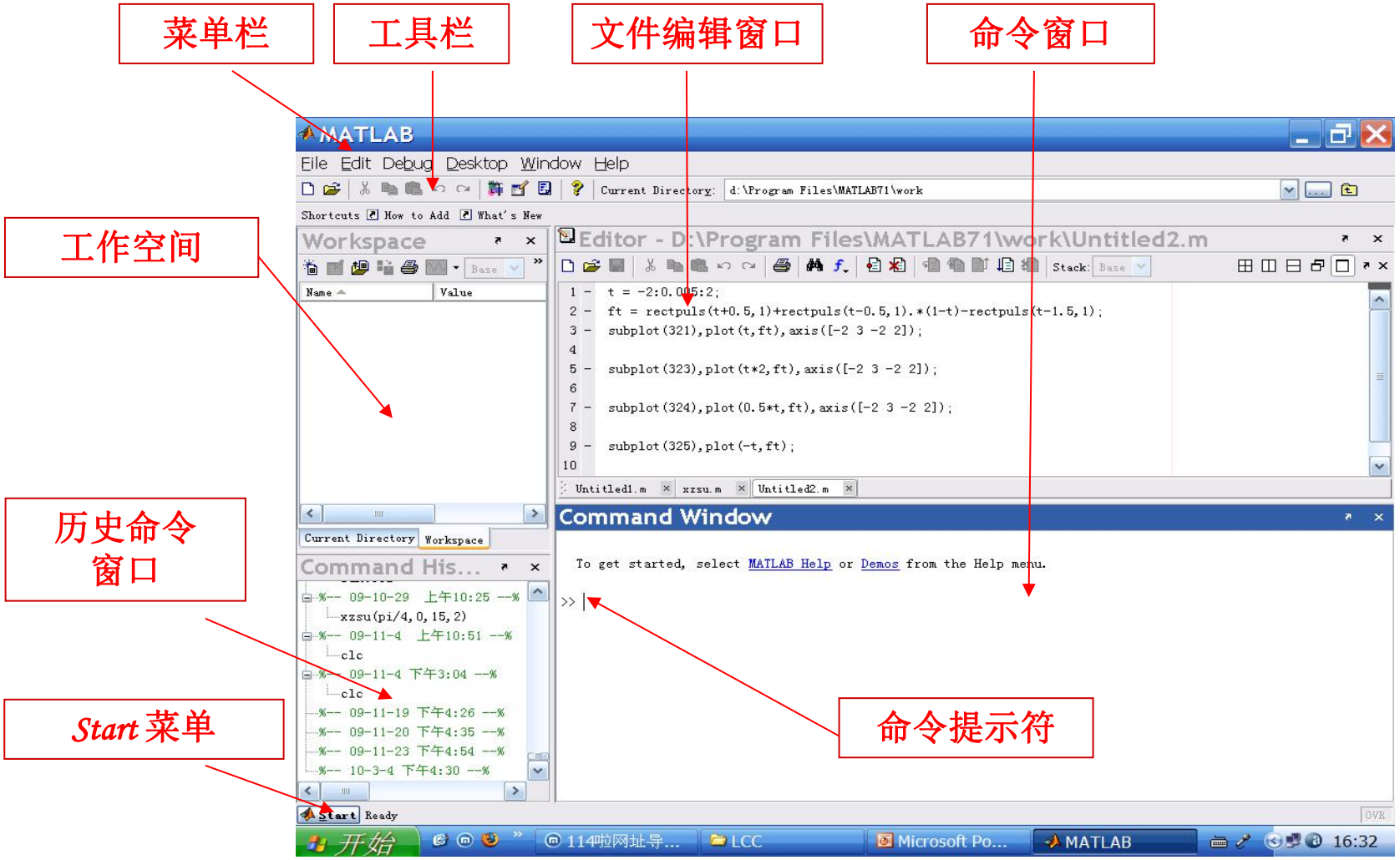
PART 2

MATLAB程序设计基础

1、MATLAB的桌面环境及入门知识

1.1 启动与退出MATLAB

- 启动MATLAB
  - 直接用鼠标双击桌面上MATLAB图标
  - 或Windows桌面的“开始”—〉“所有程序”—〉“MATLAB”—〉“MATLAB”。
- 退出MATLAB
  - 关闭MATLAB桌面
  - 在命令窗口执行quit或exit命令
- MATLAB缺省桌面（见下页）


菜单栏
工具栏
文件编辑窗口
命令窗口
工作空间
历史命令窗口
Start 菜单
命令提示符
MATLAB
File Edit Debug Desktop Window Help
Current Directory: d:\Program Files\MATLAB71\work
Shortcuts How to Add What's New
Workspace
Name Value
Current Directory Workspace
Command His...
%-- 09-10-29 上午10:25 --%
xzsu(pi/4,0,15,2)
%-- 09-11-4 上午10:51 --%
clc
%-- 09-11-4 下午3:04 --%
clc
%-- 09-11-19 下午4:26 --%
%-- 09-11-20 下午4:35 --%
%-- 09-11-23 下午4:54 --%
%-- 10-3-4 下午4:30 --%
Start Ready
Editor - D:\Program Files\MATLAB71\work\Untitled2.m
1 - t = -2:0.005:2;
2 - ft = rectpuls(t+0.5,1)+rectpuls(t-0.5,1).*(1-t)-rectpuls(t-1.5,1);
3 - subplot(321),plot(t,ft),axis([-2 3 -2 2]);
4
5 - subplot(323),plot(t*2,ft),axis([-2 3 -2 2]);
6
7 - subplot(324),plot(0.5*t,ft),axis([-2 3 -2 2]);
8
9 - subplot(325),plot(-t,ft);
10
Untitled1.m xzsu.m Untitled2.m
Command Window
To get started, select MATLAB Help or Demos from the Help menu.
>>
开始 114啦网址导... LCC Microsoft Po... MATLAB 16:32

1.2 命令窗口的使用

- 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。

- 简单计算

【例1.2-1】计算 $\left[12+2\times\left(7-4\right)\right]\div3^2$

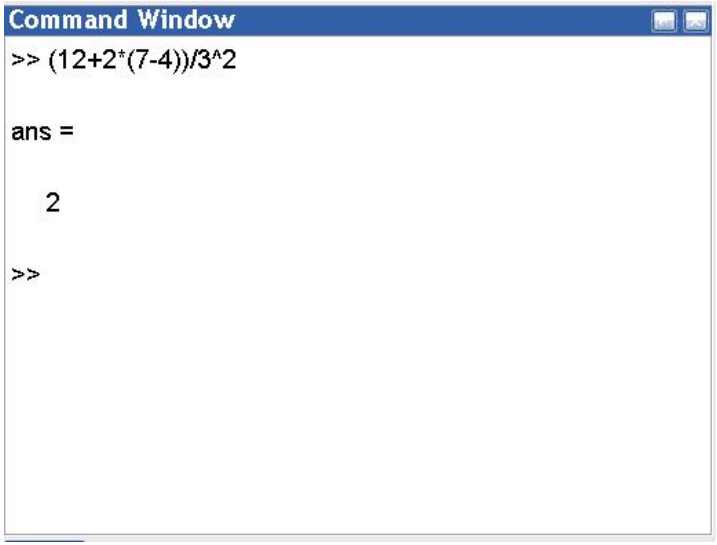
（1）在MATLAB命令窗口输入以下内容:

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

（2）按【Enter】键，指令执行。

（3）返回的计算结果:

```
ans=
     2
```



〖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。

- 简单计算
- 【例1.2-2】计算$\sin(45^\circ)$

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

ans=
    0.7071
```

- **Matalb**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**Matalb**的预定义变量。
- **pi=3.14159…**
- **Matlab**对字母大小写是敏感的。

【例1.2-3】计算$\left(\sqrt{2e^{x+0.5}+1}\right)$的值，

```
>>sqrt(2*exp(4.92+0.5)+1)
 ans=
    21.2781
```

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

“clc”清除窗口显示内容的命令。

【例1.2-4】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1. 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。

【例1.2-5】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。

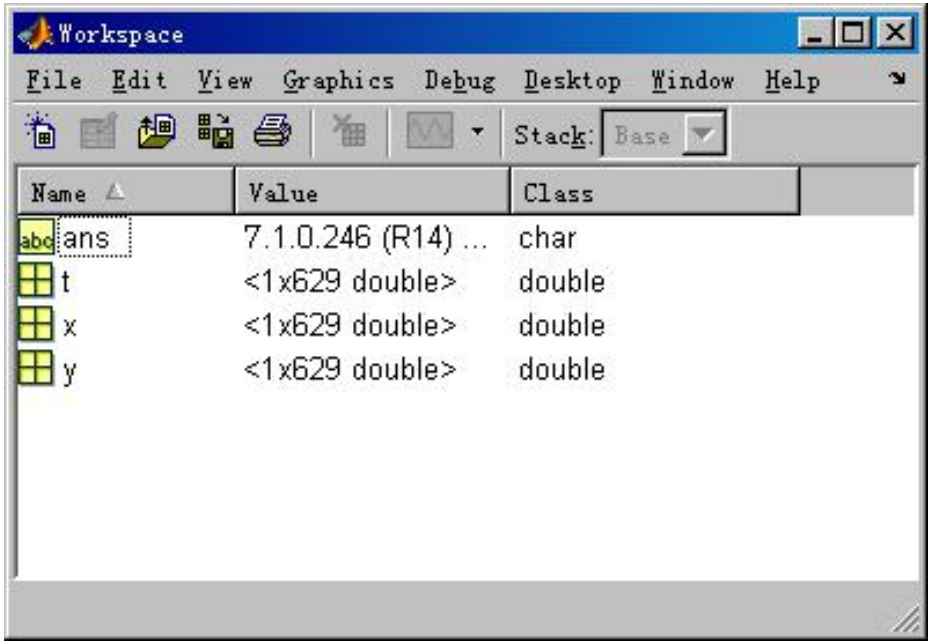
- 数值显示格式设置
  - 缺省显示格式：简洁的短（short g）格式
  - 窗口命令及语法格式：format 显示格式关键字

    如：format long %15位数字显示
- 常见通用命令

命令	含义
clc	清除命令窗口的显示内容
clear	清除Matlab工作空间中保存的变量
who或whos	显示Matlab工作空间中的变量信息
dir	显示当前工作目录的文件和子目录清单
cd	显示或设置当前工作目录
type	显示指定m文件的内容
help或doc	获取在线帮助
quit或exit	关闭/推出MATALB

1.3 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



1.4 历史空间

- 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

1.5 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

【功能演示】

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
```

【功能演示】

求解线性方程组
$$\begin{cases} 2x_1 + 3x_2 - x_3 = 2 \\ 8x_1 + 2x_2 + 3x_3 = 4 \\ 45x_1 + 3x_2 + 9x_3 = 23 \end{cases}$$

```
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
```

【功能演示】　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 = polyfit(x,y,1) ⇒

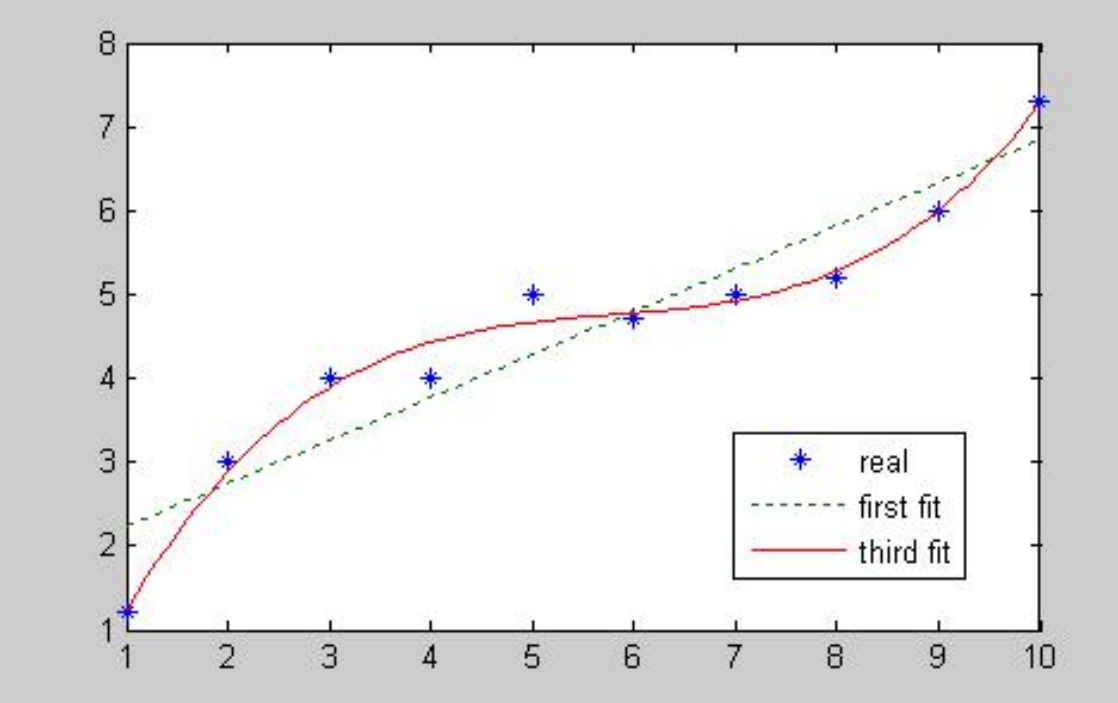
- 三次多项式拟合：

  p3 = 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=polyval(p1,x2)
y3=polyval(p3,x2)
plot( x, y, '*', x2, y1, ':', x2, y3)
```

拟合曲线图



由图可见，三次拟合结果较好。

2、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（6.5及以后的版本）。

- Matlab预定义的变量

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

〖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。*

• 运算符和表达式

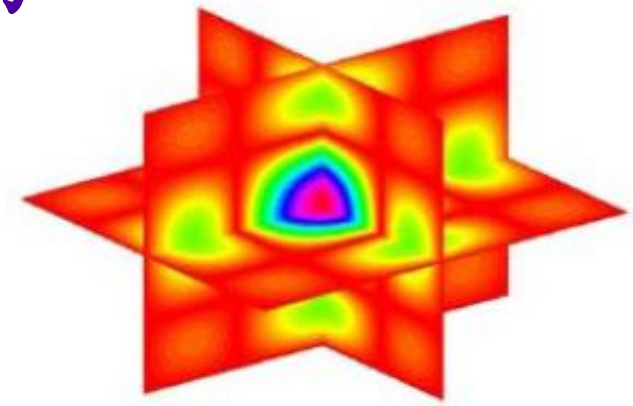
运算	数学表达式	MATLAB运算符	MATLAB表达式
加	a+b	+	a+b
减	a-b	-	a-b
乘	axb	*	a*b
除	a/b或a\b	/或\	a/b或a\b
幂	a^b	^	a^b

〖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。

3、Matlab矩阵(数组)的表示

- 数组的概念
- 一维数组变量的创建
- 二维数组变量的创建
- 数组元素的标识与寻访
- 数组运算
- 多维数组



3.1 数组(array)的概念

- **数组定义：**

  按行(row)和列(column)顺序排列的实数或复数的有序集，被称为数组。

  数组中的任何一个数都被称为这个数组的元素，由其所在的行和列标识，这个标识也称为数组元素的下标或索引。Matlab将标量视为1×1的数组。

  对m行、n列的2维数组a：

  计为m×n的数组a；

  *行标识、列标识均从1开始；

  行标识从上到下递增；

  列标识从左到右递增。

a=

1	2	3	4	5
2	22	23	24	25
3	32	33	34	35
4	42	43	44	45

a(3, 4)=34 **row is first**

- 数组的分类
  - 一维数组，也称为向量(vector) 。
    - 行向量(row vector)、列向量(column vector)。
  - 二维数组(矩阵matrix)。
  - 多维数组。

*有效矩阵：每行元素的个数必须相同，每列元素的个数也必须相同。

数组 (array)	大小(size)	
$a=\begin{bmatrix}1 & 2\\ 3 & 4\\ 5 & 6\end{bmatrix}$	3×2	
$b=\begin{bmatrix}1 & 2 & 3 & 4\end{bmatrix}$	1×4	行向量
$c=\begin{bmatrix}1\\ 2\\ 3\end{bmatrix}$	3×1	列向量

a(2,1)=3　　a(1,2)=2　　b(3)=3　　c(2)=2

3.1.1 创建一维数组变量

- 第一种方法：使用方括号“[ ]”操作符

【例】创建数组(行向量)a=[1 3 pi 3+5i]

```
>>a=[1 3 pi 3+5*i]  %or a=[1, 3, pi, 3+5*i]
a= 1.0000     3.0000     3.1416     3.0000 + 5.0000i
```

所有的向量元素必须在操作符“[ ]”之内；
向量元素间用空格或英文的逗点“,”分开。

- 第二种方法：使用冒号“:”操作符

【例】创建以1~10顺序排列整数为元素的行向量b。

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

【例】键入并执行c=1:2:10和d=1:2:9

```
>> c=1:2:10
    c=1 3 5 7 9
>>d=1:2:9
    d= 1 3 5 7 9
```

利用冒号“:”操作符创建行向量的基本语法格式:

x=Start:Increment:End

- Start表示新向量x的第一个元素;
- 新向量x的最后一个元素不能大于End ;
- Increment可正可负，若负，则必须Start>End；若正，则必须Start<End，否则创建的为空向量。
- 若Increment=1,则可简写为：x=Start:End。

- **第三种方法：利用函数linspace**

  函数linspace的基本语法

  x= linspace(x1, x2, n)

  - 该函数生成一个由n个元素组成的行向量；
  - x1为其第一个元素；
  - x2为其最后一个元素；
  - x1、x2之间元素的间隔=(x2-x1)/(n-1)。
  - 如果忽略参数n，则系统默认生成100个元素的行向量。

  【例】键入并执行x= linspace(1,2,5)

  x=1.0000　1.2500　1.5000　1.7500　2.0000

  同学们可以在实验时察看x= linspace(1,2)执行结果。

- 第四种方法：利用函数logspace

  通过实验认识该函数的功能。

- 列向量的创建
  - 使用方括号“[ ]”操作符，使用分号“;”分割行。

  【例】键入并执行x= [1; 2; 3]

  X=1
  2
  3

  - 使用冒号操作符

  【例】键入并执行x= (1:3)’ % “ ’ ”表示矩阵的转置

3.1.2 创建二维数组变量

- **第一种方法：使用方括号“[ ]”操作符**

  使用规则

  - 数组元素必须在“[ ]”内键入；
  - 行与行之间须用分号“;”间隔，也可以在分行处用回车键间隔；
  - 行内元素用空格或逗号“,”间隔。

  【例】键入并执行a2=[1 2 3;4 5 6;7 8 9]

  a2=

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

  【例】键入并执行a2=[1:3;4:6;7:9] %结果同上

【例】由向量构成二维数组。

```
>>a=[1 2 3]; b=[2 3 4];
>>c=[a;b];
>>c1=[a b];
```

- 第二种方法：函数方法

函数ones(生成全1矩阵)、zeros (生成全0矩阵) 、reshape

☞“help elmat”获得基本的矩阵生成和操作函数列表

【例】创建全1的3x3数组。

```
>>ones(3)
```

【例】创建全1的3x4数组。

```
>>ones(3,4)
```

【例】 reshape的使用演示

```
>>a=-4:4
    a=
      -4 -3 -2 -1 0 1 2 3 4
>>b=reshape(a, 3, 3)
    b=
    -4  -1  2
    -3   0  3
    -2   1  4
```

☞数组元素的排列顺序，从上到下按列排列，先排第一列，然后第二列，…

☞要求数组的元素总数不变。

4、数组元素的标识与寻访

- 数组元素的标识
  - “全下标（index）”标识

    经典数学教科书采用“全下标”标识法：每一维对应一个下标。
    - 如对于二维数组，用“行下标和列下标”标识数组的元素，a(2,3)就表示二维数组a的“第2行第3列”的元素。
    - 对于一维数组，用一个下标即可，b(2)表示一维数组b的第2个元素，无论b是行向量还是列向量。
  - “单下标”（linear index）标识

    所谓“单下标”标识就是用一个下标来表明元素在数组的位置。
    - 对于二维数组，“单下标”编号：设想把二维数组的所有列，按先后顺序首尾相接排成“一维长列”，然后自上往下对元素位置执行编号。
  - 两种“下标”标识的变换：sub2ind、ind2sub

【例】单下标的使用

```
>>a=zeros(2, 5); %设置一个存储空间
>>a(:)=-4:5  %矩阵a元素的取值是-4,-3,…,3,4,5
```

```
a =

    -4    -2     0     2     4

    -3    -1     1     3     5
```

注意数组的排列顺序。

- 元素与子数组的寻访与赋值

【例】一维数组元素与子数组的寻访与赋值

```
>>a=linspace(1,10,5)
    a =
      1.0000    3.2500    5.5000    7.7500   10.0000
>>a(3)      %寻访a的第3个元素
    ans =
      5.5000
>>a([1 2 5])  %寻访a的第1、2、5个元素组成的子数组
    ans =
      1.0000    3.2500   10.0000
```

```
>>a(1:3)   %寻访前3个元素组成的子数组
      ans =
         1.0000    3.2500    5.5000
>>a(3:-1:1)  %由前3个元素倒序构成的子数组
      ans =
         5.5000    3.2500    1.0000
>>a(3:end)
      ans =
         5.5000    7.7500   10.0000
>>a(3:end-1)
      ans =
         5.5000    7.7500
```

```
>>a([1 2 3 5 5 3 2 1])
    ans =
      1.0000    3.2500    5.5000   10.0000   10.0000    5.5000
      3.2500    1.0000
```

数组元素可以被任意重复访问，构成长度大于原数组的新数组。

```
>>a(6)
    ??? Index exceeds matrix dimensions.
```

下标值超出了数组的维数，导致错误

```
>>a(2.1)
    ??? Subscript indices must either be real positive integers or
      logicals.
```

下标值只能取正整数或逻辑值

```
>>a(3)=0          %修改数组a的第3元素值为0
    a =
       1.0000    3.2500    0    7.7500   10.0000
>>a([2 5])=[1 1]
    a =
       1.0000    1.0000    0    7.7500   1.0000
```

- 可以修改指定数组元素的值
- 一次可以修改多个数组元素的值
- 要修改的数组元素的个数应与送入数组的元素个数相同

【例】二维数组元素与子数组的寻访与赋值

```
>>a_2=zeros(2, 4) %创建2x4的全0数组
    a_2 =
        0    0    0    0
        0    0    0    0
>>a_2(:)=1:8
    a_2 =
        1    3    5    7
        2    4    6    8
>>a_2([2 5 8])   %单下标方式寻访多个元素
    ans =
        2    5    8
```

```
>> a_2([2 5 8]) =[10 20 30]
    a_2 =
       1    3   20    7
      10    4    6   30
>>a_2(:,[2 3])=ones(2)  %双下标方式寻访并修改
    a_2 =
       1    1    1    7
      10    1    1   30
```

- 二维数组可以“单下标”方式或“全下标”方式访问、赋值；
- “单下标”方式赋值时，等号两边涉及的元素个数必须相等；
- “全下标”方式赋值时，等号右边数组的大小必须等于原数组中涉及元素构成的子数组的大小。

```
>>a_2(:,end)
    ans =
        7
       30
>>a_2(:,end-1)
    ans =
        1
        1
>>a_2(:, end:-1:3)
    ans =
        7     1
       30     1
```

```
>>a_2(end,:)
    ans =
       10     1     1    30
>>a_2(end,[2:4])
    ans =
        1     1    30
>>a_2 ([4 6])=6:7
    a_2 =
        1     1     1     7
       10     6     7    30
>>a_2(end,[2:end-1])
```

What is the result?

【例】 size、length函数

```
>>a=ones(4,6)*6
>>m=size(a)
>>len=length(a)
>>b=1:5;
>>length(b)
>>c=b'
>>length(c)
```

- size函数返回变量的大小，即变量数组的行列数
- length函数返回变量数组的最大维数

5、数组的算术运算

- MATLAB数组支持线性代数中所有的矩阵运算。
- 建立特有的数组运算符，如：“.*”、“./”等。

MATLAB数组运算符列表

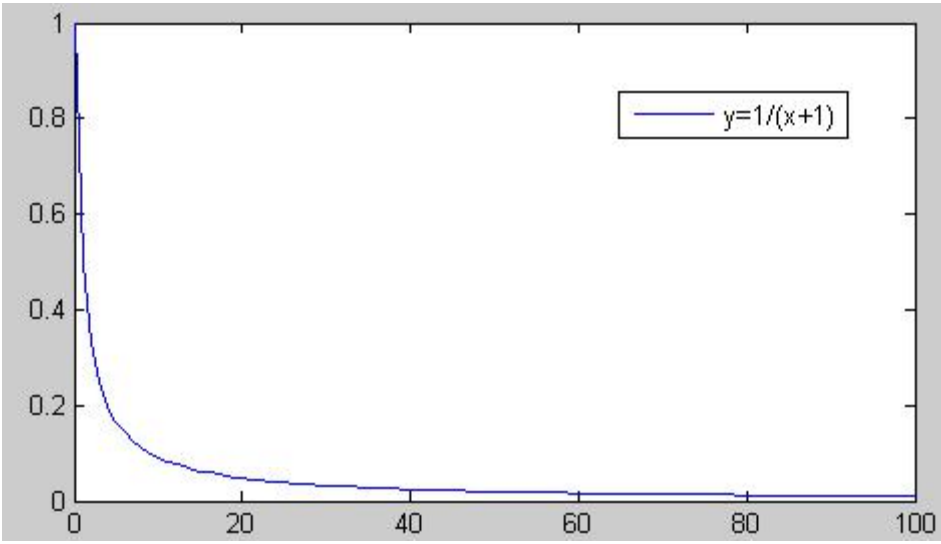
运算	运算符	含义说明
加	+	相应元素相加
减	-	相应元素相减
乘	*	矩阵乘法
点乘	.*	相应元素相乘
幂	^	矩阵幂运算
点幂	.^	相应元素进行幂运算
左除或右除	\或/	矩阵左除或右除
左点除或右点除	.\或./	A的元素被B的对应元素除

【例】数组加减法

```
>>a=zeros(2, 3);
>>a(:)=1:6;
>>b=a+2.5
b =
    3.5000    5.5000    7.5000
    4.5000    6.5000    8.5000
>>c=b-a
    c =
    2.5000    2.5000    2.5000
    2.5000    2.5000    2.5000
```

【例】画出y=1/(x+1)的函数曲线，x∈[0, 100]。

```
x=0:100;
y=1./(x+1);
plot(x, y);
legend('y=1/(x+1)');
```



【例】点幂“.^”举例

```
>>a=1:6
    a =
        1   2   3   4   5   6

>>b=reshape(a,2,3)
    b =
        1     3     5
        2     4     6
```

```
>>a=a.^2
    a =
        1   4   9   16   25   36

>>b=b.^2
    b =
        1     9     25
        4     16    36
```

6、关系运算

Matlab提供了6种关系运算符:

<、>、<=、>=、==、~=（不等于）

关系运算符的运算法则:

1、当两个标量进行比较时，直接比较两数大小。若关系成立，结果为1，否则为0。

2、当两个维数相等的矩阵进行比较时，其相应位置的元素按标量关系进行比较，并给出结果，形成一个维数与原来相同的0、1矩阵。

3、当一个标量与一个矩阵比较时，该标量与矩阵的各元素进行比较，结果形成一个与矩阵维数相等的0、1矩阵。

7、逻辑运算

Matlab提供了3种逻辑运算符:

&（与）、|（或）、~（非）

逻辑运算符的运算法则:

1、在逻辑运算中，确认非零元素为真（1），零元素为假（0）。

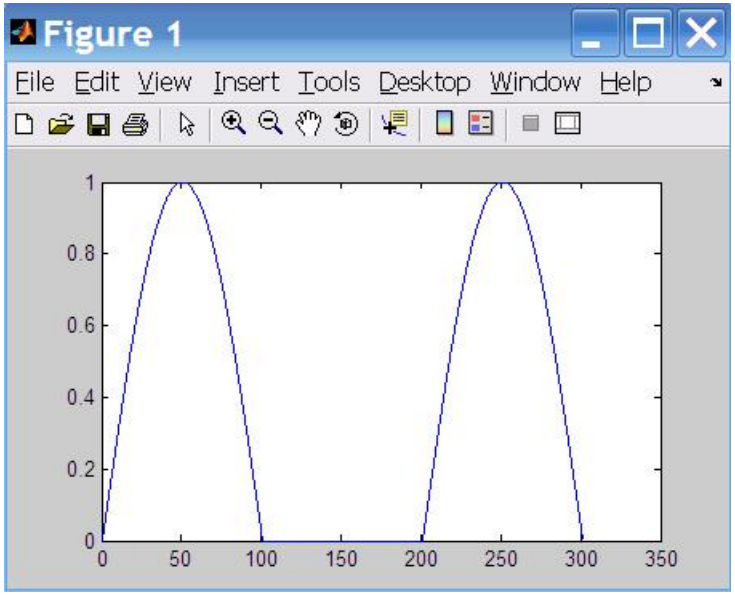
2、当两个维数相等的矩阵进行比较时，其相应位置的元素按标量关系进行比较，并给出结果，形成一个维数与原来相同的0、1矩阵;

3、当一个标量与一个矩阵比较时，该标量与矩阵的各元素进行比较，结果形成一个与矩阵维数相等的0、1矩阵;

4、算术运算优先级最高，逻辑运算优先级最低。

【例】在[0，3π]区间，求 $y=\sin x$ 的值。要求消去负半波，即(π,2π)区间内的函数值置零。

```
x = 0:pi/100:3*pi;
y = sin(x);
y1 = (y>=0).*y;  %消去负半波
```



【例】建立矩阵A，找出在[10，20]区间的元素的位置。

```
A = [4,15,-45,10,6;56,0,17,-45,0];
find(A>=10 & A<=20) %找到[10，20]区间元素的位置
```

```
A =
4    15   -45    10     6
56    0    17   -45     0
ans =
3
6
7
```

8、数据分析与统计

■ 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，两个函数的调用格式和操作过程类似。

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】求向量的最大值

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

```
>>x=[-43,72,9; 16,23,47];
>>y=max(x)     %求矩阵x中每列的最大值
y =
    16  72  47
>>[y,l]=max(x) %求矩阵x中每列的最大值及其该元素的位置
y =
   16  72  47
l =
    2  1  2
>>max(x, [],1), max(x, [],2) %求矩阵中每行的最大值
```

■ 求和与求积

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。

■ 平均值与中值

求数据序列平均值的函数是mean，求数据序列中值的函数是median。两个函数的调用格式为：

mean(X)：返回向量X的算术平均值。

median(X)：返回向量X的中值。

mean(A)：返回一个行向量，其第i个元素是A的第i列的算术平均值。

median(A)：返回一个行向量，其第i个元素是A的第i列的中值。

mean(A,dim)：当dim为1时，该函数等同于mean(A)；当dim为2时，返回一个列向量，其第i个元素是A的第i行的算术平均值。

median(A,dim)：当dim为1时，该函数等同于median(A)；当dim为2时，返回一个列向量，其第i个元素是A的第i行的中值。

9、M文件

- 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

fexch.m

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

function 输出形参表 = 函数名（输入形参表）

注释说明部分

函数体语句

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)
```

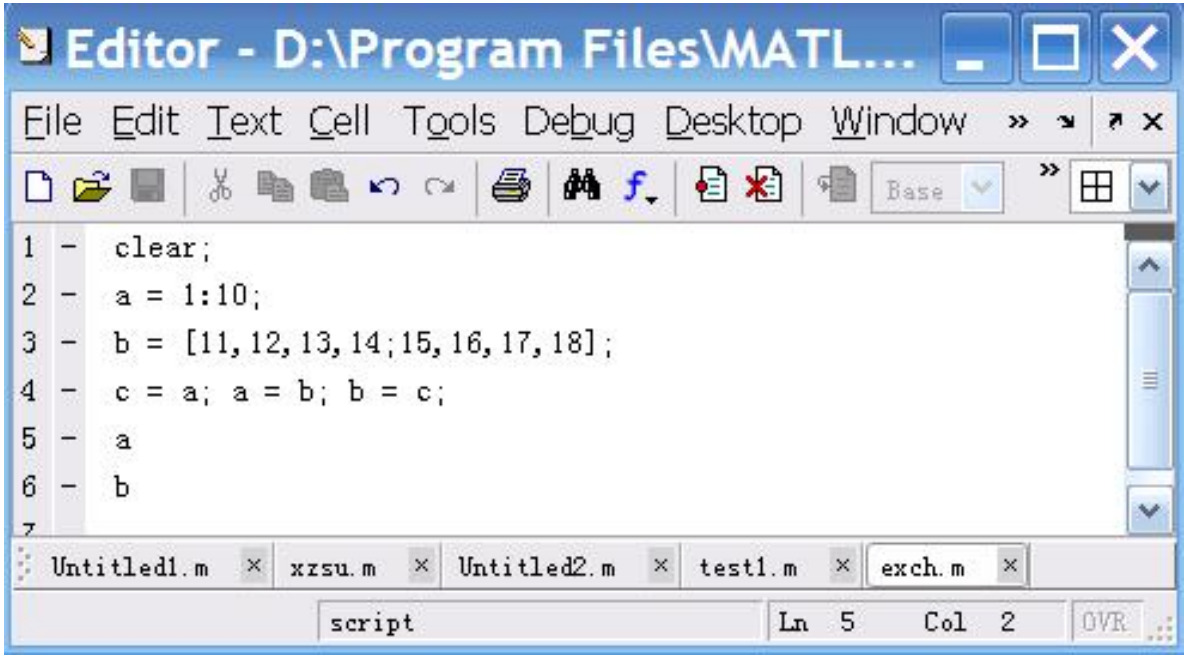
输出结果为：

```
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。

M文件是一个文本文件，可以用任何编辑程序来建立和编辑，一般最常用的是使用Matlab提供的文本编辑器。

该编辑器是一个集编辑和调试于一体的工作环境。



10、程序控制结构

顺序结构、选择结构、循环结构

任何复杂的程序都可以由这3种基本结构构成。

例 求一元二次方程 $ax^2+bx+c=0$ 的根。

由于Matlab能进行复数运算，所以不需要判断方程的判别式，可直接根据求根公式求根。

程序如下：

```
a = input('a=?');
b = input('b=?');
c = input('c=?');
d = b*b-4*a*c;
x = [(-b+sqrt(d))/(2*a),(-b-sqrt(d))/(2*a)];
disp(['x1=',num2str(x(1)),',x2=',num2str(x(2))]);
```

程序输出为：

```
a=? 4
b=? 78
c=? 54
x1=-0.7188,x2=-18.7812
```

选择结构是根据给定的条件成立或不成立，分别执行不同的语句。

Matlab用于实现选择结构的语句有if语句，switch语句和try语句。

1. if语句：在Matlab中，if语句有3种格式。

(1)单分支if语句

语句格式：

```
if 条件
   语句组
end
```

例如：当x是整数矩阵时，输出x的值

```
if fix(x)==x
   disp(x);
end
```

(2)双分支if语句

语句格式：

if 条件

语句组 1

else

语句组 2

end

当条件成立时，执行语句组1，否则执行语句组2，然后再执行if语句的后续语句。

(3)多分支if语句

语句格式：

```
if 条件1
   语句组 1
elseif  条件2
   语句组  2
…
elseif  条件m
   语句组  m
else
   语句组n
end
```

循环是指按照给定的条件，重复执行指定的语句，Matlab提供了两种

实现循环结构的语句：for语句和while语句。

1、for语句

for语句的格式为：

for 循环变量 =表达式1：表达式2：表达式3

循环体语句

end

其中表达式1的值为循环变量的初值，表达式2的值为步长，表达式3的值为循环变量的终值。步长为1时，表达式2可以省略。

已知 $y=\frac{1}{1^2}+\frac{1}{2^2}+\frac{1}{3^2}+\cdots+\frac{1}{n^2}$ ，当n=100时，求y的值。

程序如下：

```
y = 0;n = 100;
for i=1:n
   y = y+1/i/i;
end
y
```

输出结果为：

```
y =
    1.6350
```

利用Matlab的特点，常用向量运算来代替循环操作，程序可以如下：

```
n = 100;
i = 1:n;
f = 1./i.^2;
y = sum(f)
```

PART 3

MATLAB在数值计算中的应用

1、函数的数值导数

- 导数定义为：

$$\frac{dy}{dx}=\lim_{h\to 0}\frac{f(x+h)-f(x)}{(x+h)-(x)}$$

- 则y=f(x)的导数可近似为：

$$\frac{dy}{dx}\approx\frac{f(x+h)-f(x)}{(x+h)-(x)}$$

这里h>0

它是y的有限差分除以x的有限差分。

- **MATLAB中没有直接提供数值导数的函数，只有计算向前差分的函数diff，其调用格式为：**

DX = diff(X)　计算向量X的向前差分

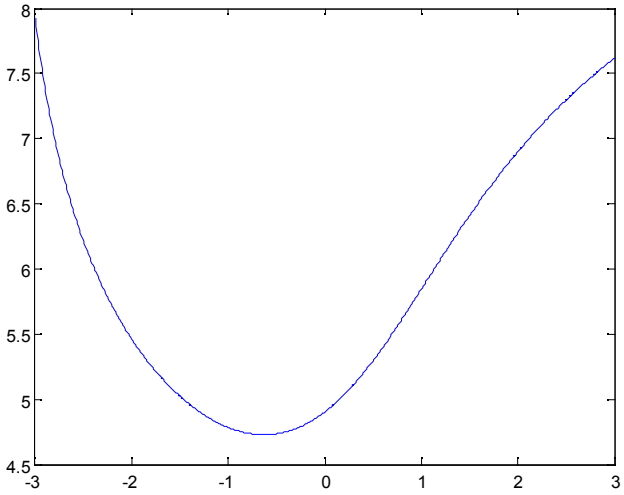
DX = diff(X，n)　计算向量X的n阶向前差分

例：设 $f(x)=\sqrt{x^3+2x^2-x+12}+\sqrt[6]{x+5}+5x+2$

在[-3,3]区间内以0.01为步长求数值导数。并画出导函数图像。

程序如下：

```
f = inline('sqrt(x.^3+2*x.^2-x+12)+(x+5).^(1/6)+5*x+2'); %内联函数
x = -3:0.01:3;
dx = diff(f([x,3.01]))/0.01;  %根据定义式求导数
plot(x,dx)
```



2、数值积分

一元函数的数值积分

- 常用积分指令：quad和quadl。

  一般说来，quadl比quad更有效。

- 具体调用格式如下：

  q = quadl(fun,a,b)

  q = quadl(fun,a,b,tol)

  q = quadl(fun,a,b,tol,trace)

  [q,fcnt] = quadl(fun,a,b,...)

  - 输入量fun为被积函数的句柄。
  - 输入量a, b分别是积分的下限、和上限，都必须是确定的数值;
  - 前3个输入参数是调用积分指令所必须的，其他可以缺省;
  - 输入量tol是一个标量，控制绝对误差;
  - 输入量trace为非0值时，将随积分的进程逐点画出被积分函数;
  - 输出参数fcnt返回函数的执行次数。

  Note：quad的调用格式与quadl相同

- 举例：求定积分 $I=\int_0^1 e^{-x^2}dx$

MATLAB指令quad和quadl求积分

```
>>fun=inline('exp(-x.*x)','x'); %数组乘符号.*的采用是必须的
>>Isim=quad(fun,0,1),  I8=quadl(fun,0,1)
```

```
Isim =  0.7468
I8 =  0.7468
```

3、数值插值

在工程测量和科学实验中，所得到的数据通常是离散的，要得到这些离散点以外的其他点的数值，就需要根据已知的数据进行插值。

插值函数一般由**线性函数、多项式、样条函数或这些函数的分段函数**充当。

一维数据插值：被插值函数有一个单变量。

采用的方法有：**线性方法、最近方法、三次样条和三次插值**。

在Matlab中实现这些插值的函数是interp1，其调用格式如下：

Y1 = interp1(X,Y,X1,method）

函数根据X，Y的值，计算函数在X1处的值。

X,Y是两个等长的已知向量，分别描述采样点和样本值；

X1是一个向量或标量，描述欲插值的点；

Y1是一个与X1等长的插值结果。

method是插值方法，允许的取值为：

（1） ‘linear’：线性插值。默认的插值方式。它是把插值点靠近的两个数据点用直线连接，然后在直线上选取对应插值点的数据。

（2） ‘nearest’：最近点插值。根据已知插值点与已知数据点的远近程度进行插值。插值点优先选择较近的数据点进行插值。

（3） ‘cubic’：3次多项式插值。根据已知数据求出一个3次多项式，然后根据该多项式进行插值。

（4） ‘spline’：3次样条插值。指在每个分段内构造一个3次多项式，使其满足插值条件外，在各节点处具有光滑的条件。

例：给出概率积分数据表如下，用不同的插值方法计算f(0.472)。

x	0.46	0.47	0.48	0.49
f(x)	0.4846555	0.4937542	0.5027498	0.5116683

命令如下：

```
x = 0.46:0.01:0.49;
f = [0.4846555,0.4937542,0.5027498,0.5116683];
format long
interp1(x,f,0.472)
ans =
 0.49555332000000
interp1(x,f,0.472,'nearest')
ans =
0.49375420000000
interp1(x,f,0.472,'spline')
ans =
0.49556073600000
interp1(x,f,0.472,'cubic')
ans =
0.49556111971206
```

其中，3次样条和3次多项式的插值结果优于最近点插值方法和线性插值方法，但插值方法的好坏依赖于被插值函数，没有一种对所有函数都是最好的插值方法。

4、数值拟合

数值插值要求逼近函数在采样点与被逼近函数相等，但由于测量误差，所获得的数据不一定准确，如果强求逼近显然不够合理。

曲线拟合不要求逼近函数通过各采样点，但要尽量的接近这些点，使误差在某种意义上达到最小。

曲线拟合的实现：

在matlab中，用polyfit函数来求得最小二乘拟合多项式的系数，再用polyval函数按所得的多项式计算所给出点上的函数近似值。

polyfit函数的调用格式为：

[P,S] = polyfit(X,Y,m)

函数根据采样点X和采样点函数值Y，产生一个m次多项式P及其在采样点的误差向量S。其中X、Y是两个等长的向量，P是一个长度为m+1的向量，P的元素是多项式系数。

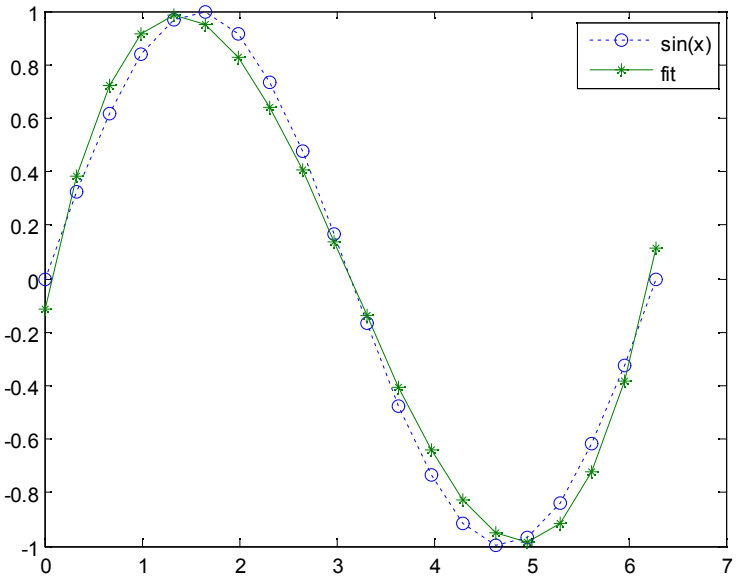
polyval函数的功能是按多项式的系数计算x点多项式的值。

例：用一个三次多项式在区间[0,2π]内逼近函数sinx。

在给定区间内，均匀的选择20个采样点，并计算采样点的函数值，然后利用3次多项式逼近。

命令如下：

```
x = linspace(0,2*pi,20);
y = sin(x);
p = polyfit(x,y,3)
y1 = polyval(p,x)
plot(x,y,':o',x,y1,'-*')
legend('sin(x)','fit')
```



PART 4

MATLAB的图形处理

1、二维图形及相关函数

❖ 基本的绘图命令

绘制二维图形最常用的函数是plot函数。其调用格式：

- plot(Y)

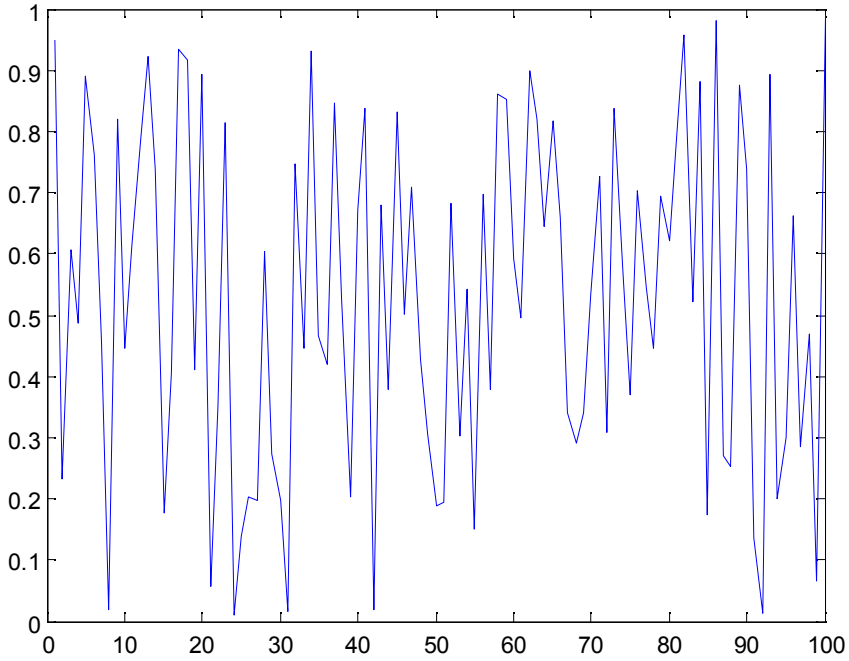
若**Y为向量**，则绘制的图形以向量索引为横坐标值，以向量分量值为纵坐标值。

若**Y为矩阵**，则绘制Y的列向量对其坐标索引的图形。

若**Y为一复向量（矩阵）**，则plot(Y)相当于plot(real(Y),imag(Y))。而在其他形式的函数调用中，元素的虚部将被忽略。

例如：

```
>> y=rand(100,1);
>> plot(y)
```



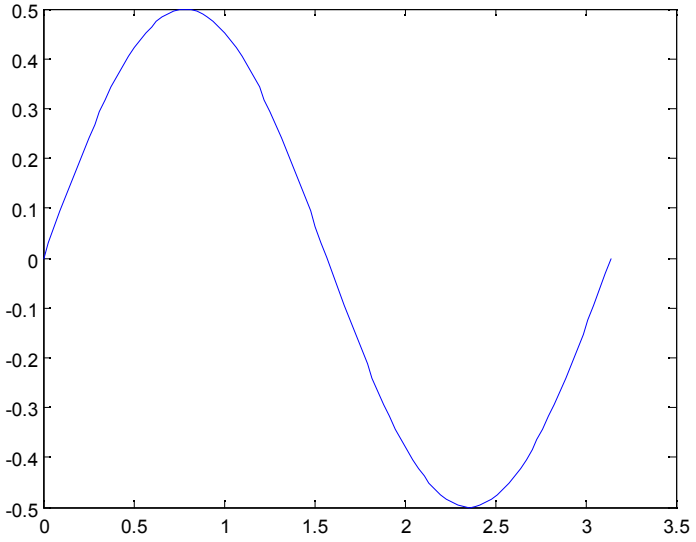
- plot(X,Y) X为横坐标，Y为纵坐标。

注意：向量X和Y必须是同维数的，也必须同是行向量或列向量。

当变量X和Y是同阶矩阵时，将按矩阵的行或列进行操作。特别的，变量Y可以包含多个符合要求的向量，这时将在同一幅图中绘制所有的图形。

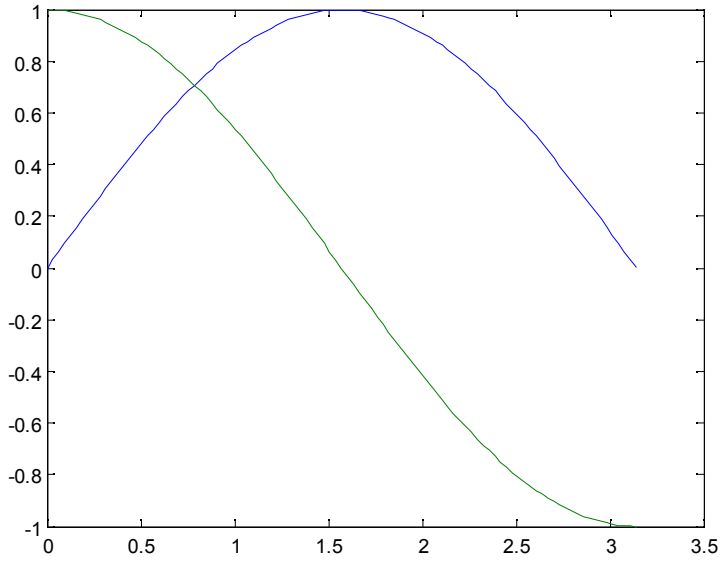
例：x，y为同维向量时

```
x=0:0.01*pi:pi;
y=sin(x).*cos(x);
plot(x,y)
```



例：

```
x=0:0.01*pi:pi;
y=[sin(x‘),cos(x’)]; %这化成y=[sin(x)’,cos(x)’]行不？
plot([x',x'],y)
```



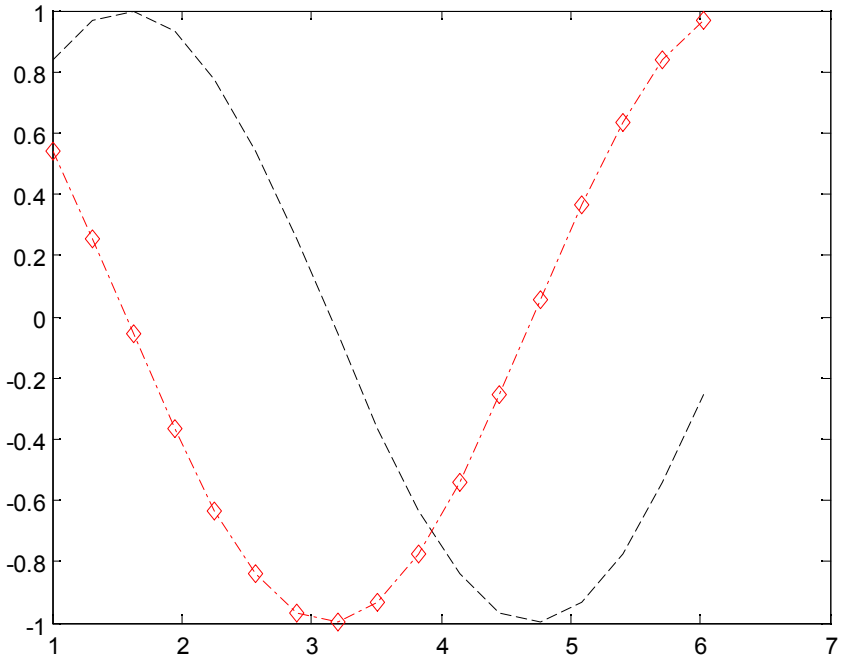
- plot(X,Y,s) X为横坐标，Y为纵坐标。s为一字符，可以代表不同的线型、点标、颜色。

  下表是s图形设置选项：当选项多于一个时，各项直接相连即可。

选项	说明	选项	说明	选项	说明
-	实线	g	绿色	x	x符号
:	点线	b	蓝色	+	+号
-.	点划线	w	白色	*	星号
--	虚线	k	黑色	s	方形
y	黄色	.	点	d	菱形
r	红色	o	圆	p	正五边形

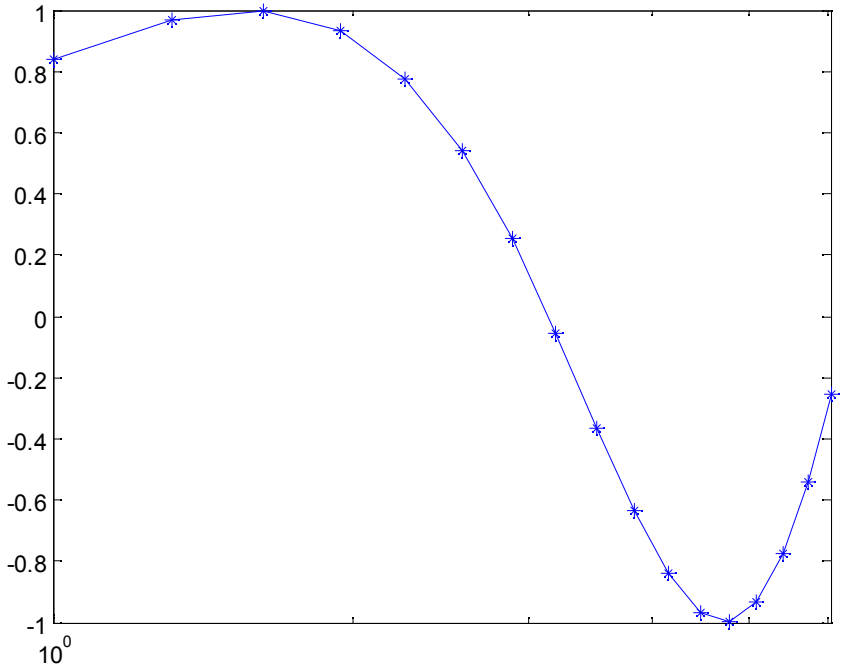
例如：

```
x=1:0.1*pi:2*pi;
y=sin(x);
z=cos(x);
plot(x,y,‘--k’,x,z,‘-.rd’) %结果怎样？
```



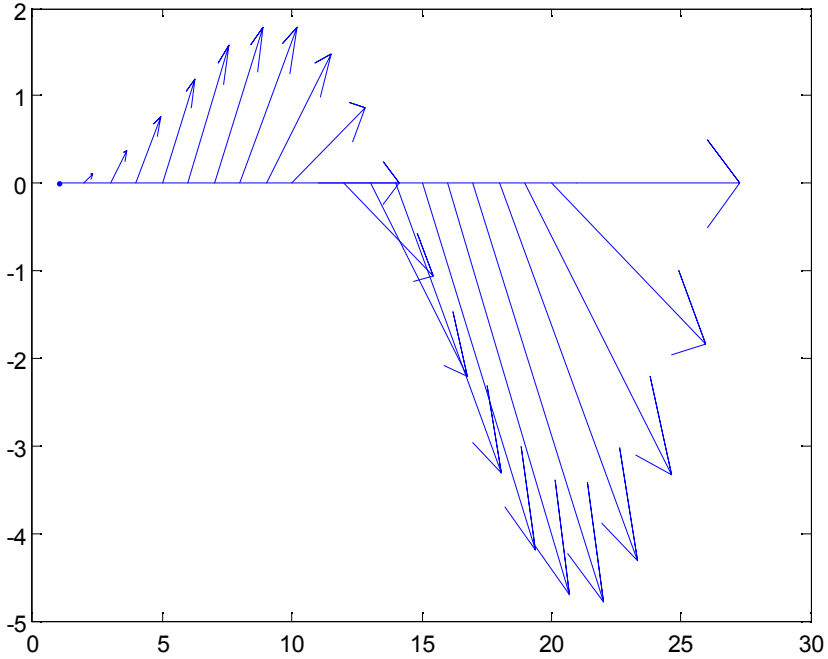
例如：

```
>> x=1:0.1*pi:2*pi;
>> y=sin(x);
>> semilogx(x,y,'-*')
```



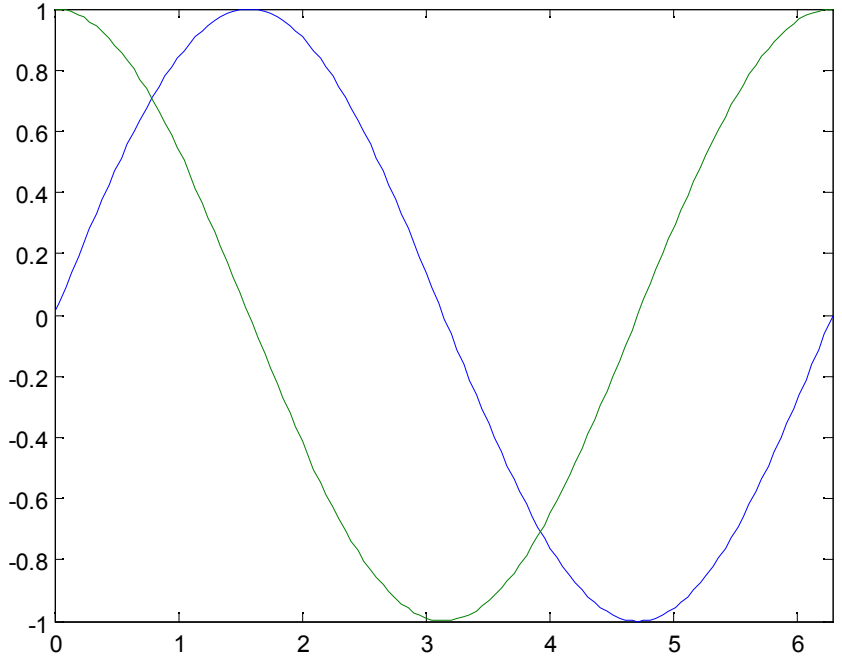
%绘制矢量图

```
x=0:0.1*pi:2*pi;
y=sin(x).*x;
feather(x,y)
```



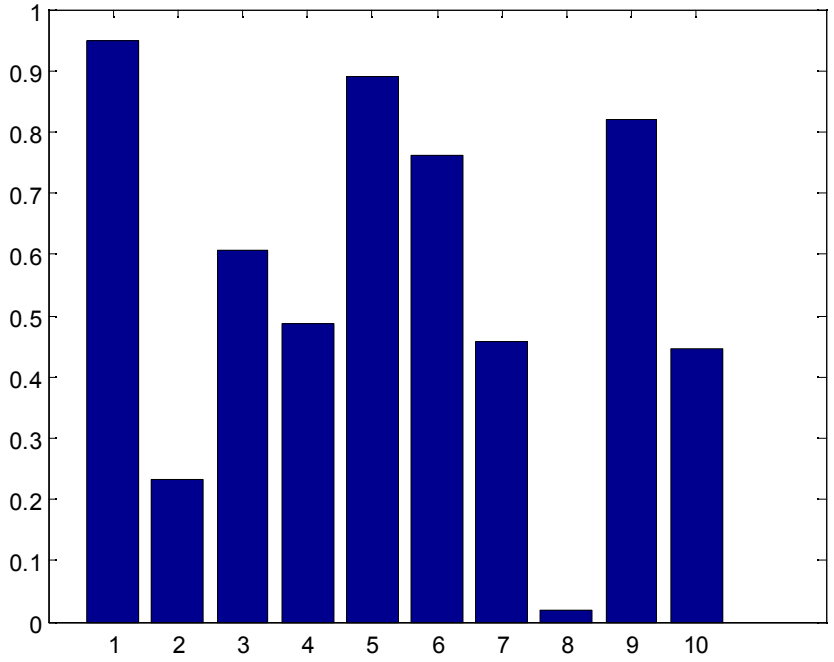
%绘制函数图形

```
lim=[0,2*pi,-1,1];
fplot('[sin(x),cos(x)]',lim)
```

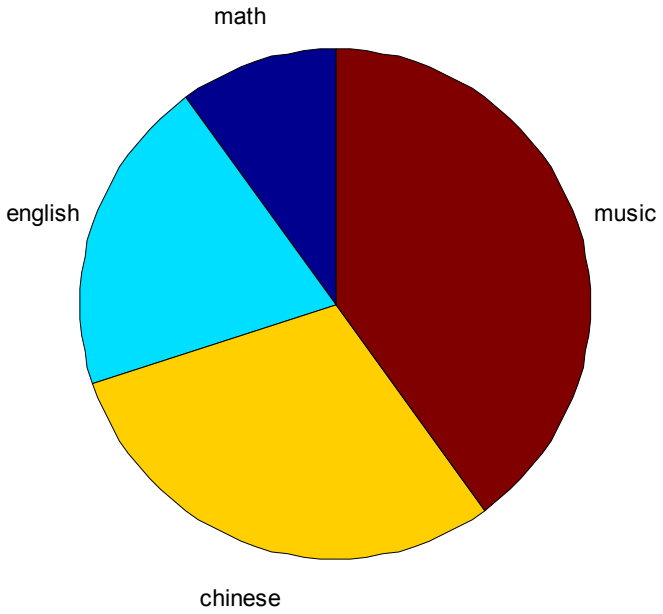


例如：

```
%绘制条形图
x=1:10;
y=rand(10,1)
bar(x,y)
```




```
%绘制饼状图形
x=[2,4,6,8];
pie(x,{'math','english','chinese','music'})
```



2、图形处理的基本技术

❖ 图形的控制

函数基本包括：坐标轴控制函数（axis）、坐标轴放缩函数（zoom）、平面图形的坐标网格函数（grid）以及坐标轴封闭函数（box）。

- 坐标轴的控制函数——axis

最简单的调用形式：axis(V) 其中V为一个用于存储坐标轴的坐标范围的数组。

对于二维图形：V=[xmin,xmax,ymin,ymax]

对于三维图形：V=[xmin,xmax,ymin,ymax,zmin,zmax]

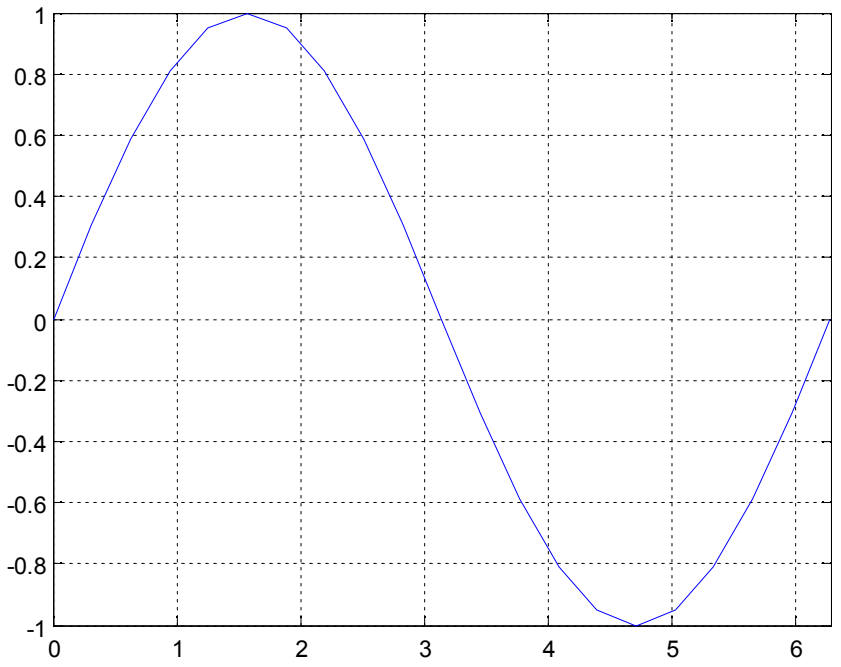
- 平面坐标的网格函数——grid

  具体的调用格式：grid on/off

  grid on 表示在图形中绘制坐标网格，grid off 表示取消坐标网格

单独的grid函数将实现grid on与grid off状态间的转换。

```
x=0:0.1*pi:2*pi;
y=sin(x);
plot(x,y)
axis([0,2*pi,-1,1])
grid on
```



图形的标注

- 坐标标注

常用的函数有：title、xlabel、ylabel等。

上述三个函数的调用格式大同小异，以xlabel为例：

xlabel(‘标注’,’属性1’,属性值1,’属性2’,属性值2,……)

这里属性为文本属性，包括：字体大小、字体名、字体粗细等。

```
x=0:0.1*pi:2*pi;
y=sin(x);
plot(x,y)
axis([0,2*pi,-1,1])
xlabel('x(0-\pi)','fontweight','bold');
ylabel('y=sin(x)','fontweight','bold');
title('正弦函数','fontsize',12)
```

- 子图——subplot(m,n,p)

subplot函数将把一个图形窗口分割成m*n个子绘图区域，用户柯通过参数p调用各子绘图区域进行操作。

例如：把图形窗口分割成2x2的子绘图区域。在各个绘图子域分别绘制sin(x)，cos(x),sin(x).*cos(x),sin(x)+cos(x),并给出各子图的标题。

20XXPOWERPOINT

感谢观看 THANGKS!

单击此处添加您的副标题文字