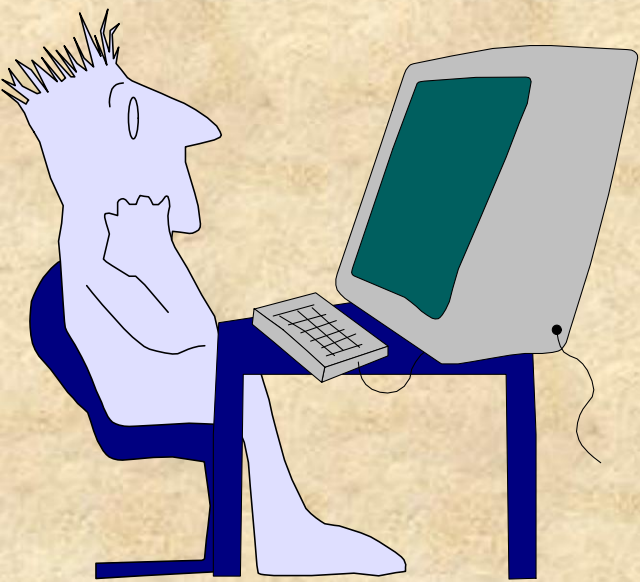


数学建模与数学实验

MATLAB入门



数学建模 (Mathematical Modelling)

- 数学建模

- 把现实世界中的实际问题加以提炼，抽象为数学模型，求出模型的解，验证模型的合理性，并用该数学模型所提供的解答来解释现实问题。数学知识的这一应用过程称为数学建模。

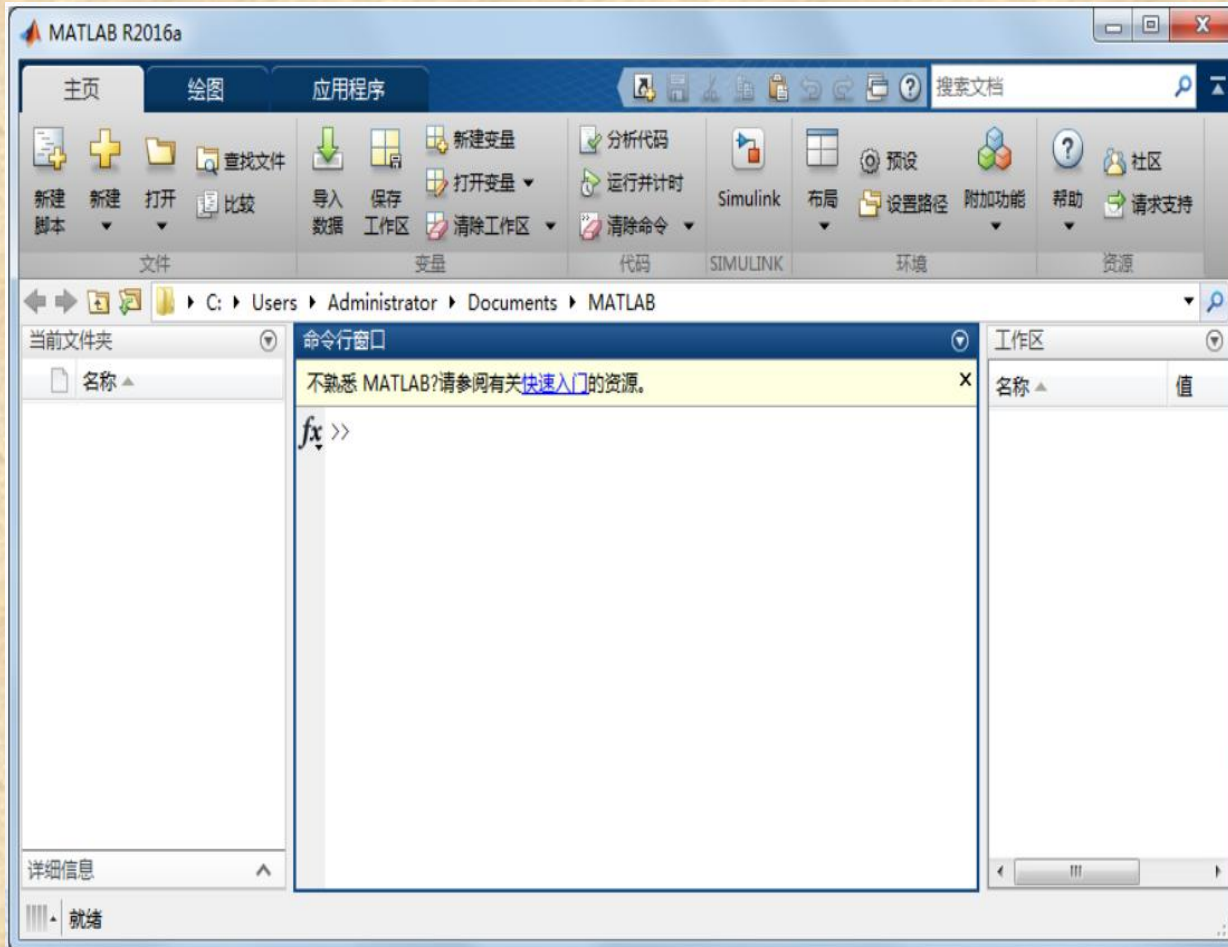
常用数学软件

- 综合型:通常功能比较全面,国际上有影响的主要有MATLAB、Maple 、 Mathematica等
- 专业型: 通常功能比较单一
 - 统计软件 (SPSS SAS SPLUS)
 - 数据处理和作图软件 (Origin)
 - 优化软件 (Lingo)
- 排版结合型:通常将科学排版和科学计算想融合,如Mathcad、 Scientific等

- MATLAB作为线性系统的一种分析和仿真工具，是理工科大学生应该掌握的技术工具，它作为一种编程语言和可视化工具，可解决**工程**、**科学计算**和**数学学科**中许多问题.
- MATLAB建立在**向量**、**数组**和**矩阵**的基础上，使用方便，人机界面直观，输出结果可视化.
- **矩阵**是MATLAB的核心.
- MATLAB的进入与运行方式（两种）.

MATLAB R2016a 窗口

❖ MATLAB的版本不断更新，图1展示的是MATLAB R2016a的界面，从中了解一下MATLAB操作界面的构成。



打开MATLAB R2016a，默认打开的窗口包括：(1)命令行窗口；(2)当前文件夹窗口；(3)工作区窗口。

命令行窗口

- 命令行窗口用于输入命令并显示命令的执行结果。其中>>为命令提示符。

❖ >> a=[1 2 3];

❖ >> b=[4 5 6];

❖ >> c=a+b

❖ 得到运算结果为

❖ c =

❖ 5 7 9

- 在MATLAB中变量默认为矩阵类型。
- 命令行后面加; 表示不显示结果

表1 MATLAB 常用命令

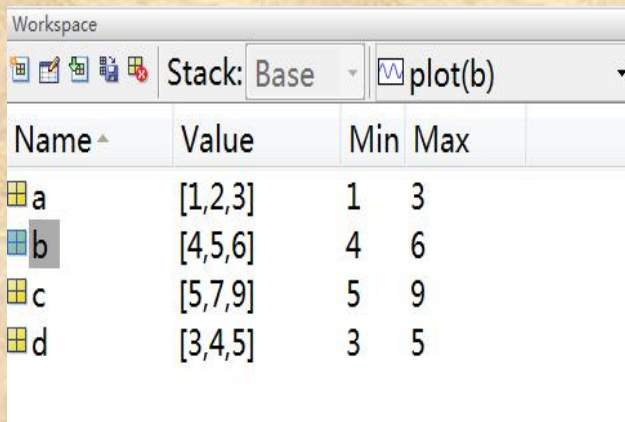
命令	功能	命令	功能
who	显示内存变量	whos	显示内存变量的详细信息
clear	清除工作空间中的变量	clc	清除命令窗口显示的内容
help	获得帮助信息	demo	获得demo演示帮助信息
figure	打开新图形窗口	clf	清除图形窗口
type	显示M文件的内容	which	显示文件所在文件夹
cd	设置当前工作文件夹	md	创建文件夹
dir	显示目录清单	edit	打开M文件编辑器
exit	退出MATLAB	quit	退出MATLAB

当前文件夹窗口

- ❖ 在程序设计时，如果不特别指明存放数据和文件的路径，MATLAB默认把数据和文件存放在当前文件夹中，为使用方便最好把用户文件夹设置为当前文件夹。有两种方式设置当前文件夹：
- ❖ （1）在当前文件夹工具栏或当前文件夹窗口中选择某文件夹为当前文件夹
- ❖ （2）使用cd命令
- ❖ 在命令行窗口输入cd e:\matlabexers 命令就将e:\matlabexers文件夹设置为当前文件夹。建立文件时会自动存入当前文件夹中。如果MATLAB需要和多个文件夹交换信息，则可以把这些文件夹设置为MATLAB的搜索路径中。使用path命令可以添加搜索路径，例如命令
- ❖ >> path(path,'e:\bai')
- ❖ 将E:\bai文件夹添加到搜索路径中。

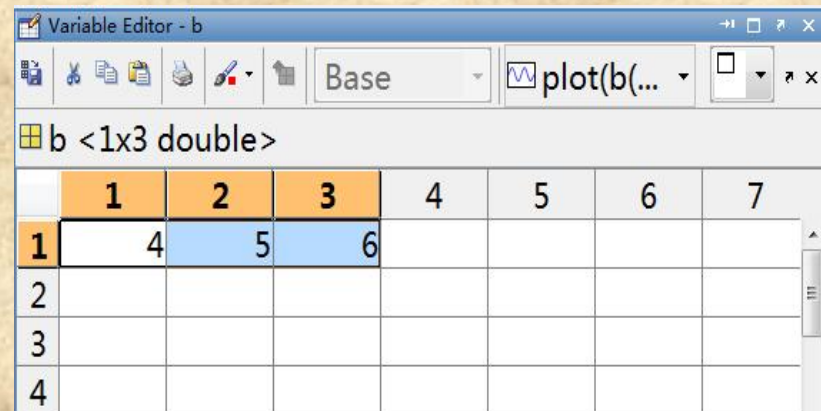
工作区（Workspace）窗口

- ❖ 工作区中可以看到当前计算机内存中MATLAB变量的名称、值、数学结构、该变量的字节数及其类型等，在默认设置下，工作区管理窗口自动显示于MATLAB界面中。
- ❖ 鼠标双击某个变量后可以打开变量编辑器窗口，方便修改变量的值，选中数据b可以单击或右键plot(b)，得到函数图像，下拉菜单中还有更多选择。如图所示。



The screenshot shows the MATLAB Workspace window. It has a toolbar with icons for saving, deleting, and other workspace actions. Below the toolbar, there's a 'Stack' dropdown menu set to 'Base' and a 'plot(b)' button. The main area is a table with columns 'Name', 'Value', 'Min', and 'Max'.

Name	Value	Min	Max
a	[1,2,3]	1	3
b	[4,5,6]	4	6
c	[5,7,9]	5	9
d	[3,4,5]	3	5



The screenshot shows the MATLAB Variable Editor window for variable 'b'. It has a toolbar with icons for saving, deleting, and other variable editing actions. Below the toolbar, there's a 'Base' dropdown menu and a 'plot(b(...))' button. The main area is a grid showing the values of variable 'b'.

	1	2	3	4	5	6	7
1	4	5	6				
2							
3							
4							

MATLAB入门

一、变量与函数

二、数组

三、矩阵

四、MATLAB编程

五、实验作业

一、变量与函数

1. 变量

MATLAB中变量的命名规则是：

- (1) 变量名必须是不含空格的单个词；
- (2) 变量名区分大小写，标准函数名以及命令名必须用小写字母；
- (3) 变量名最多不超过63个字符；
- (4) 变量名必须以字母打头，之后可以是任意字母、数字或下划线，变量名中不允许使用标点符号。

变量赋值语句有两种形式：

(1) 变量=表达式

(2) 表达式

练习

- ❖ 以下哪项是MATLAB的合法变量名
- ❖ A. 1a
- ❖ B. a#1
- ❖ C. a_1
- ❖ D. 平均

练习

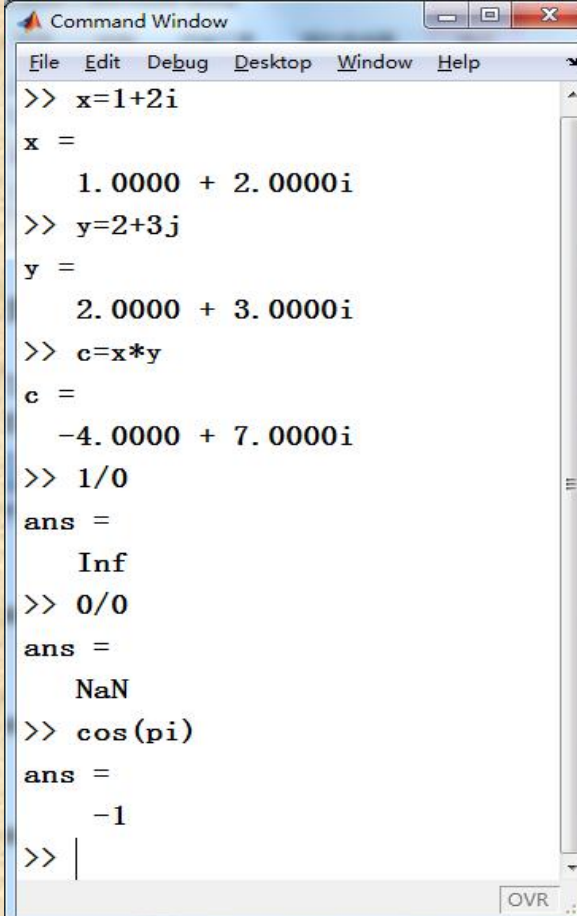
- ❖ 以下哪项是MATLAB的合法变量名
- ❖ A. 1a
- ❖ B. a#1
- ❖ C. a_1
- ❖ D. 平均

答案： C

预定义变量

❖ 预定义变量

- 预定义变量是在MATLAB工作空间中驻留，由系统本身定义的变量。例如，
 - π 代表圆周率，
 - ans代表默认变量名
 - NaN代表非数，
 - i和j代表虚数单位，
 - inf或Inf代表无穷大等。
 - MATLAB可以直接面向复数进行运算。



```
Command Window
File Edit Debug Desktop Window Help
>> x=1+2i
x =
    1.0000 + 2.0000i
>> y=2+3j
y =
    2.0000 + 3.0000i
>> c=x*y
c =
   -4.0000 + 7.0000i
>> 1/0
ans =
    Inf
>> 0/0
ans =
    NaN
>> cos(pi)
ans =
    -1
>> |
```


特殊变量表

特殊变量	取 值
ans	用于结果的缺省变量名
pi	圆周率
eps	计算机的最小数，和 1 相加时产生一个比 1 大的数
flops	浮点运算数
inf	无穷大，如 1/0
NaN	不定量，如 0/0
i, j	$i=j=\sqrt{-1}$
nargin	所用函数的输入变量数目
nargout	所用函数的输出变量数目
realmin	最小可用正实数
realmax	最大可用正实数

变量的管理

- ❖ 内存变量的删除与修改可以在工作区窗口进行。使用who或whos命令可以查看内存变量。
- ❖ 内存变量文件用于保存MATLAB工作区变量的文件，其扩展名为.mat，也叫MAT文件，save命令用于创建内存变量文件。
- ❖ load命令用于装入内存变量文件。使用方式如下：
 - >> save mydata a x
 - >> load mydata

练习

- ❖ 在命令行窗口输入下列命令后，a的值为
- ❖ `>>clear`
- ❖ `>>a=i*j`
- ❖ A. 0
- ❖ B. 不确定
- ❖ C. $i*j$
- ❖ D. -1

练习

❖ 在命令行窗口输入下列命令后，a的值为

❖ `>>clear`

❖ `>>a=i*j`

❖ A. 0

❖ B. 不确定

❖ C. $i*j$

❖ D. -1

❖ 思考题：如果没有clear这句，结果又如何？

答案：D

答案：B

2. 数学运算符号及标点符号

+	加法运算，适用于两个数或两个同阶矩阵相加.
—	减法运算
*	乘法运算
.*	点乘运算
/	除法运算
./	点除运算
^	乘幂运算
.^	点乘幂运算
\	反斜杠表示左除.

- (1) MATLAB的每条命令后，若为**逗号**或**无标点**符号，
则显示命令的结果；若命令后为**分号**，则禁止显示结果.
- (2) “%” 后面所有文字为注释.
- (3) “...”表示续行.

3. 数学函数

函 数	名 称	函 数	名 称
<code>sin(x)</code>	正弦函数	<code>asin(x)</code>	反正弦函数
<code>cos(x)</code>	余弦函数	<code>acos(x)</code>	反余弦函数
<code>tan(x)</code>	正切函数	<code>atan(x)</code>	反正切函数
<code>abs(x)</code>	绝对值	<code>max(x)</code>	最大值
<code>min(x)</code>	最小值	<code>sum(x)</code>	元素的总和
<code>sqrt(x)</code>	开平方	<code>exp(x)</code>	以 e 为底的指数
<code>log(x)</code>	自然对数	<code>log₁₀(x)</code>	以 10 为底的对数
<code>sign(x)</code>	符号函数	<code>fix(x)</code>	取整
<code>abs(x)</code>	求绝对值、复数的模及字符串的 ASCII 值	<code>rem(x,n)</code>	求 x 除以 n 的余数

- ❖ 三角函数有以弧度为单位的函数和以角度为单位的函数，如果是以角度为单位的函数，就在函数名后面加“d”，以示区别，
 - 例如`sin(pi/2)`和`sind(90)`分别是弧度和角度的函数用法，计算的值一样。

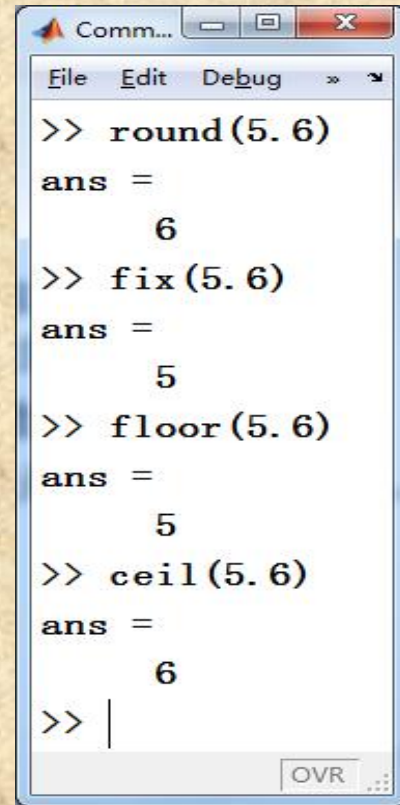
数学函数

```
Comm...
File Edit Debug
>> abs(-5)
ans =
    5
>> abs(3+4i)
ans =
    5
>> abs('a')
ans =
    97
>> sin(pi/2)
ans =
    1
>> sind(90)
ans =
    1
>>
```

```
Command Window
File Edit Debug Desktop Window Help
>> %求三位数m的个位、十位、百位
>> m=456;
>> d1=rem(m, 10)
d1 =
    6
>> d2=rem(fix(m/10), 10)
d2 =
    5
>> d3=fix(m/100)
d3 =
    4
>>
```


取整函数

- ❖ 用于取整的函数有fix、floor、ceil、round
- ❖ round函数是按照四舍五入的规则来取整；
- ❖ ceil是向上取整，取大于等于这个数的第一个整数。
- ❖ floor是向下取整，取小于等于这个数的第一个整数。
- ❖ fix是固定取靠近0的那个整数，也就是舍去小数取整

A screenshot of the MATLAB Command Window. The window has a title bar with 'Comm...' and standard minimize, maximize, and close buttons. The menu bar includes 'File', 'Edit', and 'Debug'. The command history shows four function calls: 'round(5.6)', 'fix(5.6)', 'floor(5.6)', and 'ceil(5.6)'. Each call is followed by 'ans =' and the result. The results are 6, 5, 5, and 6 respectively. The prompt '>>' is visible at the end of the last line.

```
>> round(5.6)
ans =
    6
>> fix(5.6)
ans =
    5
>> floor(5.6)
ans =
    5
>> ceil(5.6)
ans =
    6
>> |
```

矩阵运算函数

- ❖ **MATLAB**提供了矩阵计算的有关函数，可以求矩阵的维数、长度、行列式值、矩阵的秩、矩阵的迹、矩阵的范数和条件数等。
 - **size(A)**:求矩阵的各维的大小。
 - **length(A)**:求矩阵的长度，
 - 如果是多维矩阵，取各维长度的最大值。

求矩阵大小和长度

❖ `>> A=[1 2 3 4 5; 6 7 8 9 0];`

❖ `>> size(A)`

❖ `ans =`

❖ `2 5`

❖ `>> length(A)`

❖ `ans =`

❖ `5`

❖ `>> length(A')`

❖ `ans =`

❖ `5`

练习

- ❖ 假设a是一个2行3列的矩阵，执行以下命令后b的值是
- ❖ `>> b=zeros(size(a))`
- ❖ A. 3行2列的全0矩阵
- ❖ B. 2行3列的全0矩阵
- ❖ C. 2行1列的全0矩阵
- ❖ D. 1行3列的全0矩阵

练习

- ❖ 假设a是一个2行3列的矩阵，执行以下命令后b的值是
- ❖ `>> b=zeros(size(a))`
- ❖ A. 3行2列的全0矩阵
- ❖ B. 2行3列的全0矩阵
- ❖ C. 2行1列的全0矩阵
- ❖ D. 1行3列的全0矩阵

答案： B

矩阵函数

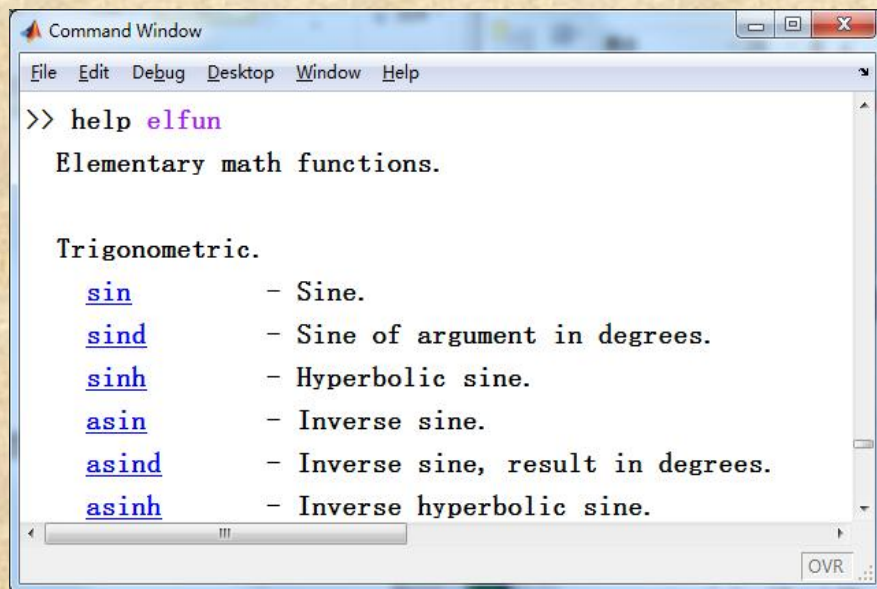
- ❖ $\det(A)$: 求方阵 A 所对应的行列式的值。
- ❖ $\text{rank}(A)$: 求矩阵 A 的秩，矩阵线性无关的行数或列数称为矩阵的秩。
- ❖ $\text{trace}(A)$: 求矩阵 A 的迹，矩阵的迹等于矩阵的对角线元素之和，也等于矩阵的特征值之和。
- ❖ 矩阵或向量的范数用来度量矩阵或向量在某种意义下的长度。
- ❖ 在 MATLAB 中，求向量范数的函数为：
- ❖ $\text{norm}(V)$ 或 $\text{norm}(V,2)$: 计算向量 V 的2范数。
- ❖ $\text{norm}(V,1)$: 计算向量 V 的1范数。
- ❖ $\text{norm}(V,\text{inf})$: 计算向量 V 的 ∞ 范数。

- ❖ 矩阵 A 的条件数等于 A 的范数与 A 的逆矩阵的范数的乘积。条件数越接近于1，矩阵的性能越好，反之，矩阵的性能越差。
- ❖ 在MATLAB中，计算矩阵 A 的3种条件数的函数是：
- ❖ $\text{cond}(A,1)$ ：计算 A 的1范数下的条件数。
- ❖ $\text{cond}(A)$ 或 $\text{cond}(A,2)$ ：计算 A 的2范数下的条件数。
- ❖ $\text{cond}(A,\text{inf})$ ：计算 A 的 ∞ 范数下的条件数。

帮助函数

❖ MATLAB提供了帮助命令可以查询个函数的用法，例如 help可以列出全部的帮助主题；

- help elfun命令可以查阅各基本函数的用法；
- help elmat命令可以查阅初等矩阵和矩阵运算的相关函数。

A screenshot of the MATLAB Command Window. The title bar says "Command Window". The menu bar includes "File", "Edit", "Debug", "Desktop", "Window", and "Help". The command prompt shows ">> help elfun". The output text is as follows:

```
Elementary math functions.  
  
Trigonometric.  
    sin          - Sine.  
    sind         - Sine of argument in degrees.  
    sinh         - Hyperbolic sine.  
    asin         - Inverse sine.  
    asind        - Inverse sine, result in degrees.  
    asinh        - Inverse hyperbolic sine.
```

At the bottom right of the window, there is a button labeled "OVR" and a small icon.

4. M文件

MATLAB的内部函数是有限的，有时为了研究某一个函数的各种性态，需要为MATLAB定义新函数，为此必须编写函数文件. 函数文件是文件名后缀为M的文件，这类文件的**第一行必须是一特殊字符function开始**，格式为：

function 因变量名=函数名（自变量名）

函数值的获得必须通过具体的运算实现，并赋给因变量.

M文件建立方法：

1. 在MATLAB中，点:File→New → M-file
2. 在编辑窗口中输入程序内容
3. 点File → Save，存盘，M文件名必须与函数名一致.

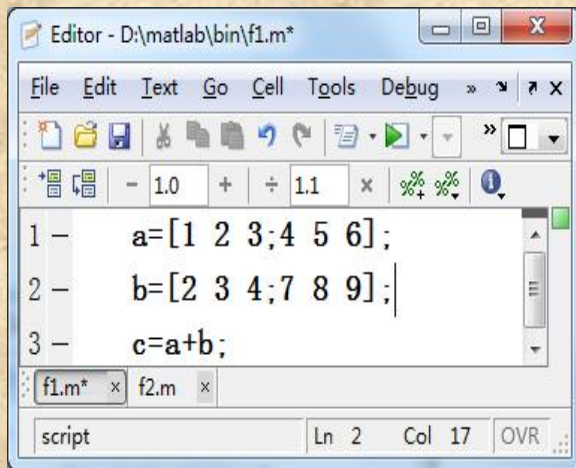
MATLAB的应用程序也以M文件保存.

❖ **M文件**可分为两大类，**M脚本文件**和**M函数文件**，这两种**Matlab**程序代码所编写的文件通常都是以“**.m**”为扩展名，因此都统称为**M文件**。

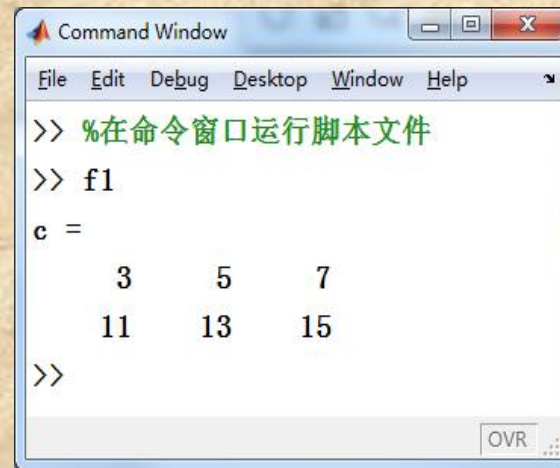
- 在**MATLAB**窗口的文件菜单**File**中**new**再选**script**，即可打开脚本文件的编辑器
- 在**File**菜单中**new**下再选**function**即可打开函数文件的编辑器。函数文件的标志就是以**function**开头
- **MATLAB**的不同版本菜单和操作稍有差异。

脚本文件

- ❖ 脚本文件可在命令行窗口直接执行的文件，也叫命令文件，直接输入文件名即可执行。
- ❖ 例如，文件名为**f1.m**，则在命令窗口直接输入**f1**即可执行这个脚本文件，得到运行结果，其变量定义在工作空间中。



```
Editor - D:\matlab\bin\f1.m*
File Edit Text Go Cell Tools Debug » » »
+ + - 1.0 + ÷ 1.1 x % % %
1 - a=[1 2 3;4 5 6];
2 - b=[2 3 4;7 8 9];
3 - c=a+b;
f1.m* x f2.m x
script Ln 2 Col 17 OVR
```

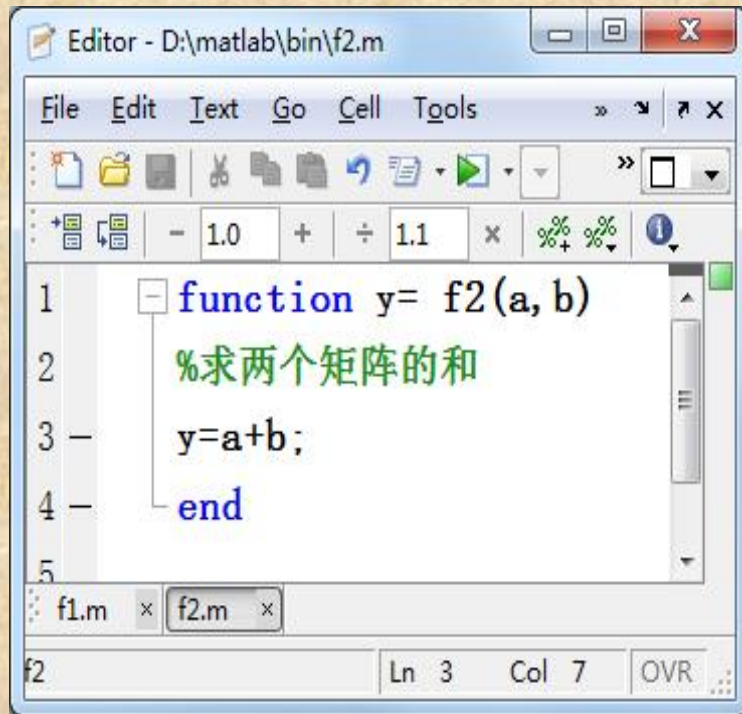


```
Command Window
File Edit Debug Desktop Window Help »
>> %在命令窗口运行脚本文件
>> f1
c =
     3     5     7
    11    13    15
>>
OVR
```


函数文件

- ❖ 函数文件是定义一个函数（**function**），不能在编辑器窗口直接执行，而必须以函数调用的方式来执行它。函数文件在保存时需要文件名要和函数名一致。当函数文件名与函数名不相同时，**MATLAB**将忽略函数名，调用时使用函数文件名。例如，函数名为**f2**，那么保存的文件名也应该为**f2**，在命令窗口调用函数时，
- ❖ 使用如下的一般调用形式：
 - 函数名（实际参数）

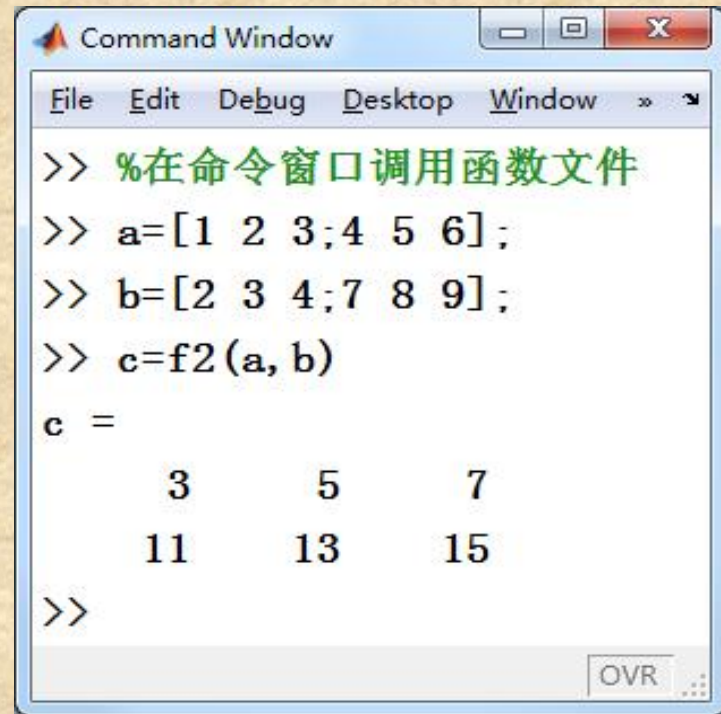
$f2(a,b)$, 函数的定义和调用



The image shows the MATLAB Editor window for a file named f2.m. The window has a menu bar (File, Edit, Text, Go, Cell, Tools) and a toolbar with various icons. The code is as follows:

```
1 function y= f2(a, b)
2     %求两个矩阵的和
3     y=a+b;
4 end
```

The status bar at the bottom indicates the current line is 3 and the column is 7.



The image shows the MATLAB Command Window. The user has entered the following commands:

```
>> %在命令窗口调用函数文件
>> a=[1 2 3;4 5 6];
>> b=[2 3 4;7 8 9];
>> c=f2(a, b)
```

The output shows the value of c:

```
c =
     3     5     7
    11    13    15
```

The status bar at the bottom indicates the current line is 3 and the column is 7.

例：定义函数 $f(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$

1. 建立M文件： **fun.m**

```
function      f=fun(x)  
f=100*(x(2)-x(1)^2)^2+(1-x(1))^2
```

2. 可以直接使用函数 **fun.m**

例如：计算 $f(1,2)$, 只需在MATLAB命令窗口键入命令：

```
x=[1 2]
```

```
fun(x)
```

MATLAB(fun)

返回

二、数 组

MATLAB(shuzu1)

1. 创建简单的数组

`x=[a b c d e f]` 创建包含指定元素的行向量.

`x=first: last`

创建从**`first`**开始, 加1计数, 到**`last`**结束的行向量.

`x=first: increment: last`

创建从**`first`**开始, 加**`increment`**计数, 到**`last`**结束的行向量.

`x= linspace(first, last, n)`

创建从**`first`**开始, 到**`last`**结束, 有**`n`**个元素的行向量.

`x=logspace(first, last, n)`

创建从**`first`**开始, 到**`last`**结束, 有**`n`**个元素的对数分隔行向量.

2. 数组元素的访问

(1) **访问一个元素**: $\mathbf{x}(\mathbf{i})$ 表示访问数组 \mathbf{x} 的第 \mathbf{i} 个元素.

(2) **访问一块元素**: $\mathbf{x}(\mathbf{a} : \mathbf{b} : \mathbf{c})$ 表示访问数组 \mathbf{x} 的第 \mathbf{a} 个元素开始, 以步长 \mathbf{b} 到第 \mathbf{c} 个元素(但不超过 \mathbf{c}), \mathbf{b} 可以为负数, \mathbf{b} 缺省时为1.

(3) **直接使用元素编址序号**. $\mathbf{x}([\mathbf{a} \ \mathbf{b} \ \mathbf{c} \ \mathbf{d}])$ 表示提取数组 \mathbf{x} 的第 \mathbf{a} 、 \mathbf{b} 、 \mathbf{c} 、 \mathbf{d} 个元素构成一个新的数组
 $[\mathbf{x}(\mathbf{a}) \ \mathbf{x}(\mathbf{b}) \ \mathbf{x}(\mathbf{c}) \ \mathbf{x}(\mathbf{d})]$.

3. 数组的方向

前面例子中的数组都是一行数，是行方向分布的。称之为**行向量**。数组也可以是**列向量**，它的数组操作和运算与行向量是一样的，唯一的区别是结果以列形式显示。

产生列向量有两种方法：

直接产生 例 $c = [1; 2; 3; 4]$

转置产生 例 $b = [1 \ 2 \ 3 \ 4]; \ c = b'$

说明：以**空格**或**逗号**分隔的元素指定的是不同列的元素，而以**分号**分隔的元素指定了不同行的元素。

4. 数组的运算

(1) 标量-数组运算

数组对标量的加、减、乘、除和平方运算，是指数组的每个元素对该标量施加相应的加、减、乘、除、平方运算. 设: $a=[a_1, a_2, \dots, a_n]$, c 是标量.

则: $a+c=[a_1+c, a_2+c, \dots, a_n+c]$

$a.*c=[a_1*c, a_2*c, \dots, a_n*c]$

$a./c=[a_1/c, a_2/c, \dots, a_n/c]$ (右除)

$a.\backslash c=[c/a_1, c/a_2, \dots, c/a_n]$ (左除)

$a.^c=[a_1^c, a_2^c, \dots, a_n^c]$

$c.^a=[c^{a_1}, c^{a_2}, \dots, c^{a_n}]$

MATLAB(shuzu3)

(2) 数组-数组运算

当两个数组有相同维数时，加、减、乘、除、幂运算可按**元素对元素**方式进行，不同大小或维数的数组是不能进行运算的。

设： $a=[a_1, a_2, \dots, a_n]$, $b=[b_1, b_2, \dots, b_n]$

则： $a+b= [a_1+b_1, a_2+b_2, \dots, a_n+b_n]$

$a.*b= [a_1*b_1, a_2*b_2, \dots, a_n*b_n]$

$a./b= [a_1/b_1, a_2/b_2, \dots, a_n/b_n]$

$a.\backslash b= [b_1/a_1, b_2/a_2, \dots, b_n/a_n]$

$a.^b= [a_1^{b_1}, a_2^{b_2}, \dots, a_n^{b_n}]$

MATLAB(shuzu4)

返回

三、矩 阵

1. 矩阵的建立

逗号或空格用于分隔某一行的元素，分号用于区分不同的行. 除了分号，在输入矩阵时，按Enter键也表示开始新一行. 输入矩阵时，严格要求所有行有相同的列.

例 $m = [1 \ 2 \ 3 \ 4 \ ; \ 5 \ 6 \ 7 \ 8; \ 9 \ 10 \ 11 \ 12]$

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

创建矩阵

- ❖ (1) 直接输入
- ❖ (2) 利用已建好的矩阵连接为更大的矩阵:
- ❖ (3) 冒号操作符
- ❖ (4) 利用`linspace`产生行向量

(1) 直接输入

❖ 将矩阵的元素用中括号括起来，按矩阵行的顺序输入各元素，同一行的各元素之间用逗号或空格分隔，不同行的元素之间用分号分隔或直接回车。

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

- `a =`

- `1 2 3`

- `4 5 6`

- `7 8 9`

(2) 利用已建好的矩阵连接为更大的矩阵

- ❖ 一个大矩阵可以由已经建立好的小矩阵拼接而成。
 - 水平方向连接用`horzcat(a,b)`或`[a,b]`或`[a b]`;
 - 垂直方向连接用`vertcat(a,b)`或`[a;b]`。
 - `c=[a a;b b]`就是由`a`矩阵和`b`矩阵拼接成一个更大的矩阵。

矩阵的拼接

❖ `>> a=[1 2 3;4 5 6;7 8 9]`

❖ `a =`

❖ `1 2 3`

❖ `4 5 6`

❖ `7 8 9`

❖ `>> b=11:16`

❖ `b =`

❖ `11 12 13 14 15 16`

❖ `>> c=[a a;b ;b]`

`c =`

`1 2 3 1 2 3`

`4 5 6 4 5 6`

`7 8 9 7 8 9`

`11 12 13 14 15 16`

`11 12 13 14 15 16`

(3) 冒号操作符

❖ 冒号表达式的一般格式为：

- `e1:e2:e3`
 - 其中e1为初始值，e2为步长，e3为终止值，省略步长e2，则步长为1。生成一个从e1到e3，以步长e2自增的行向量。
- `>> x=1:5`
- `x =`
- `1 2 3 4 5`
- `>> y=1:2:10`
- `y =`
- `1 3 5 7 9`

(4) 利用linspace产生行向量

❖ 使用线性等分函数linspace，其一般形式为：

- `linspace(a,b,n)`

- 其中a为第一个元素，b为最后一个元素，n为元素个数。当n省略时，自动产生100个元素。这样产生的向量的元素成等差数列。

- `>> z=linspace(1,9,5)`

- `z =`

- `1 3 5 7 9`

2. 矩阵中元素的引用

- (1) 通过下标来引用矩阵元素
- (2) 通过序号来引用
- (3) 利用冒号表达式获得子矩阵

(1) 通过下标来引用矩阵元素

- $a(1,2)$ 表示 a 矩阵第1行第2列的元素，用赋值语句可以改变其值， $a(1,2)=100$ 。
- 注意：如果给出的行下标或列下标大于原来矩阵的行数和列数，那么MATLAB将自动扩展原来的矩阵，并将扩展后没有赋值的矩阵元素置为0。

❖ `>> a=[1 2 3;4 5 6;7 8 9]`

❖ `a =`

❖ `1 2 3`

❖ `4 5 6`

❖ `7 8 9`

`>> a(4,5)=100`

`a =`

`1 2 3 0 0`

`4 5 6 0 0`

`7 8 9 0 0`

`0 0 0 0 100`

(2) 通过序号来引用

- ❖ 在MATLAB中，矩阵元素按列存储，
- ❖ 即首先存储矩阵的第1列元素，然后存储第2列元素，...，一直到矩阵的最后一列元素。
- ❖ 矩阵元素的序号就是矩阵元素在内存中的排列顺序。序号与下标是一一对应的
- ❖ 以 $m \times n$ 矩阵A为例，矩阵元素A(i,j)的序号为 $(j-1) \times m + i$ 。当a为之前例子中的4*5矩阵时，第5个元素的值为2。

❖ >> a(5)

❖ ans =

❖ 2

1	2	3	0	0
4	5	6	0	0
7	8	9	0	0
0	0	0	0	100

(3) 利用冒号表达式获得子矩阵

- ❖ 子矩阵是指由矩阵中的一部分元素构成的矩阵。
- ❖ 包括以下几种方式：
- ❖ $a(i,:)$ 表示 a 矩阵第 i 行的全部元素；
- ❖ $a(:,j)$ 表示 a 矩阵第 j 列的全部元素；
- ❖ $a(i:i+d,k:k+d)$ 表示 a 矩阵第 $i \sim i+d$ 行内且在第 $k \sim k+d$ 列中的所有元素；
- ❖ $a(i:i+d,:)$ 表示 a 矩阵第 $i \sim i+d$ 行的全部元素；
- ❖ end 运算符：表示某一维的末尾元素下标。
 - $a(\text{end},:)$ 表示最后一行。

(3) 利用冒号表达式获得子矩阵

❖ >> a=[1 2 3 4;5 6 7 8;9 10 11 12]

❖ a =

❖ 1 2 3 4

❖ 5 6 7 8

❖ 9 10 11 12

❖ >> a(2,:)

❖ ans =

❖ 5 6 7 8

>> a(:,2:3)

ans =

2 3

6 7

10 11

>> a(end,:)

ans =

9 10 11 12

练习 以下命令序列输出结果是：

❖ `>> a=[1 2 3 4;5 6 7 8;9 10 11 12]`

❖ `a =`

❖ `1 2 3 4`

❖ `5 6 7 8`

❖ `9 10 11 12`

❖ `>> a(:,3)`

A. 3

B. 9 10 11 12

C. 3

7

11

D. 1 2 3

5 6 7

9 10 11

答案：C

3.利用空矩阵删除矩阵元素

❖ 空矩阵是指没有任何元素的矩阵。

- $x=[]$ ， x 就是一个空矩阵。 $a(:,2:3)=[]$ 表示把 a 矩阵的第2列到第3列删除

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

- $a =$

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

`>> a(:,2:3)=[]`

$a =$

1	4
5	8
9	12

4. 改变矩阵的形状

- ❖ 单撇号'或`transpose`函数可实现矩阵的转置，即行列互换。
- ❖ 冒号:可将矩阵变为一维列向量。
- ❖ `reshape(A,m,n)`: 在矩阵总元素保持不变的前提下，将矩阵A重新排成 $m \times n$ 的二维矩阵。
 - 注意，`reshape`函数只是改变原矩阵的行数和列数，但**并不改变原矩阵元素个数及其存储顺序**。

改变形状

❖ `>> x=1:9`

❖ `x =`

❖ `1 2 3 4 5 6 7 8 9`

❖ `>> a=reshape(x,3,3)`

❖ `a =`

❖ `1 4 7`

❖ `2 5 8`

❖ `3 6 9`

将矩阵变为列向量

❖ $A(:)$ 可将矩阵 A 的每一列元素堆叠起来，成为一个列向量。

- `>> c=[1 2; 3 4]`

- `c =`

- `1 2`

- `3 4`

- `>> d=c(:)`

- `d =`

- `1`

- `3`

- `2`

- `4`

矩阵中元素的操作

(1) 矩阵A的第r行: $A(r, :)$

(2) 矩阵A的第r列: $A(:, r)$

(3) 依次提取矩阵A的每一列, 将A拉伸为一个列向量: $A(:)$

(4) 取矩阵A的第i1~i2行、第j1~j2列构成新矩阵: $A(i1:i2, j1:j2)$

(5) 以逆序提取矩阵A的第i1~i2行, 构成新矩阵: $A(i2:-1:i1, :)$

(6) 以逆序提取矩阵A的第j1~j2列, 构成新矩阵: $A(:, j2:-1:j1)$

(7) 删除A的第i1~i2行, 构成新矩阵: $A(i1:i2, :) = []$

(8) 删除A的第j1~j2列, 构成新矩阵: $A(:, j1:j2) = []$

(9) 将矩阵A和B拼接成新矩阵: $[A \ B]; [A; B]$

MATLAB(matrix2)

5.利用函数创建特殊矩阵

创建矩阵的函数包括eye、ones、zeros、rand、randn、diag、magic、meshgrid和ndgrid等。

- ❖ eye产生一个单位阵；
- ❖ ones产生一个全一的矩阵，
- ❖ zeros产生一个全零的矩阵，
- ❖ rand产生一个随机阵，数值介于0~1之间，
- ❖ randn产生一个随机矩阵，数据呈标准正态分布。
- ❖ diag产生一个以一个向量为主对角线的对角阵。
- ❖ magic产生一个魔方阵。各函数的参数如果只有一个就是产生一个方阵，如果给两个参数（m,n）则产生一个m行n列的矩阵。
- ❖ [x,y]=meshgrid(1:10,1:10)可生成3-D图形所需要的多维数据。

特殊矩阵的建立:

c=ones (m, n) 产生一个m行n列的元素全为1的矩阵

b=zeros (m, n) 产生一个m行n列的零矩阵

a=[] 产生一个空矩阵，当对一项操作无结果时，返回空矩阵，空矩阵的大小为零

d=eye (m, n) 产生一个m行n列的单位矩阵

MATLAB(matrix1)

产生特殊矩阵

❖ >> a=eye(3)

❖ a =

❖ 1 0 0

❖ 0 1 0

❖ 0 0 1

❖ >> b=ones(2,4)

❖ b =

❖ 1 1 1 1

❖ 1 1 1 1

❖ >> c=zeros(2,2)

❖ c =

❖ 0 0

❖ 0 0

❖ >> d=magic(3)

❖ d =

❖ 8 1 6

❖ 3 5 7

❖ 4 9 2

产生特殊矩阵

❖ >> e=rand(2,5)

❖ e =

❖ 0.1576 0.9572 0.8003 0.4218 0.7922

❖ 0.9706 0.4854 0.1419 0.9157 0.9595

❖ >> f=randn(2,4)

❖ f =

❖ 0.6715 0.7172 0.4889 0.7269

❖ -1.2075 1.6302 1.0347 -0.3034

产生特殊矩阵

❖ `>> g=diag([1 2 3])`

❖ `g =`

❖ $\begin{matrix} 1 & 0 & 0 \end{matrix}$

❖ $\begin{matrix} 0 & 2 & 0 \end{matrix}$

❖ $\begin{matrix} 0 & 0 & 3 \end{matrix}$

例 分别建立 3×3 、 3×2 和与矩阵A同样大小的零矩阵。

(1) 建立一个 3×3 零矩阵。

`zeros(3)`

(2) 建立一个 3×2 零矩阵。

`zeros(3,2)`

(3) 设A为 2×3 矩阵，则可以用`zeros(size(A))`建立一个与矩阵A同样大小零矩阵。

`A=[1 2 3;4 5 6];` %产生一个 2×3 阶矩阵A

`zeros(size(A))` %产生一个与矩阵A同样大小的零矩阵

例 建立随机矩阵:

(1) 在区间[20,50]内均匀分布的5阶随机矩阵。

(2) 均值为0.6、方差为0.1的5阶正态分布随机矩阵。

命令如下:

```
x=20+(50-20)*rand(5)
```

```
y=0.6+sqrt(0.1)*randn(5)
```

此外，常用的函数还有`reshape(A,m,n)`，它在矩阵总元素保持不变的前提下，将矩阵A重新排成 $m \times n$ 的二维矩阵。

练习 以下哪个命令序列不能得到3*3的矩阵

:

- ❖ A. `>> a=[1 2 3; 4 5 6 ;7 8 9]`
- ❖ B. `>> a=1:9`
- ❖ `>> b=reshape(a,3,3)`
- ❖ C. `>> a=1:16`
- ❖ `>>b=reshape(a,4,4)`
- ❖ `>>c=b(1:3,2:4)`
- ❖ D. `>>a=ones(2,3)`

答案： D

MATLAB的基本运算

❖ 1. 算术运算

- (1) 基本算术运算
 - +(加)、-(减)、*(乘)、/(右除)、\ (左除)、^(乘方)
- (2) 点运算
 - .*、./、.\和.^

❖ 2. 关系运算

- <(小于)、<=(小于或等于)、>(大于)、>=(大于或等于)
- ==(等于)、~==(不等于)。

❖ 3. 逻辑运算

- &(逻辑与)、|(逻辑或)和~(逻辑非)

基本算术运算

- ❖ 基本算术运算符包括+(加)、-(减)、*(乘)、/(右除)、\ (左除)、^(乘方)。
- ❖ **MATLAB**的算术运算对象默认是矩阵类型的。单个数据的算术运算只是矩阵运算的一种特例。
- ❖ 加减运算时，若两矩阵同型，则运算时两矩阵的相应元素相加减。
- ❖ 若两矩阵不同型，则**MATLAB**将给出错误信息。
- ❖ 一个标量也可以和矩阵进行加减运算，这时把标量和矩阵的每一个元素进行加减运算。

矩阵的运算

(1) 标量-矩阵运算

标量-数组运算相同.

(2) 矩阵-矩阵运算

[1] 元素对元素的运算，同数组-数组运算.

[2] 矩阵运算：

矩阵加法： $\mathbf{A}+\mathbf{B}$

矩阵乘法： $\mathbf{A}*\mathbf{B}$

方阵的行列式： $\det(\mathbf{A})$

MATLAB(matrix3)

方阵的逆： $\text{inv}(\mathbf{A})$

方阵的特征值与特征向量： $[\mathbf{V}, \mathbf{D}]=\text{eig}[\mathbf{A}]$

返回

- 在MATLAB中，有两种矩阵除法运算：右除/和左除\。如果A矩阵是非奇异方阵，则 B/A 等效于 $B*\text{inv}(A)$ ， $A\backslash B$ 等效于 $\text{inv}(A)*B$ 。
- 乘方运算时，一个矩阵的乘方运算可以表示成 A^x ，要求A为方阵，x为标量。

点运算

- 点运算符包括 `.*`、`./`、`.\`和`.^`。
- 两矩阵进行点运算是指它们的对应元素进行相关运算，要求两矩阵同型。


```
Command Window
File Edit Debug Desktop >>
>> s=[1 2 3;2 3 4]
s =
     1     2     3
     2     3     4
>> t=[1 2;3 4;5 6]
t =
     1     2
     3     4
     5     6
>> w=s*t
w =
    22    28
    31    40
>> q=t*s
q =
     5     8    11
    11    18    25
    17    28    39
>> p=2*w
p =
    44    56
    62    80
>>
```

```
Command Window
File Edit Debug Desktop Window Help >>
>> a=[2 2 3;4 7 7;-2 4 5]
a =
     2     2     3
     4     7     7
    -2     4     5
>> b=[3;1;-7]
b =
     3
     1
    -7
>> x=a\b
x =
    2.0000
   -2.0000
    1.0000
>> y=a/b
??? Error using ==> mrdivide
Matrix dimensions must agree.
>> z=inv(a)*b
z =
     2
    -2
     1
>>
```

```
Command Window
File Edit Debug Desktop Window Help >>
>> e=[1 2 3]
e =
     1     2     3
>> f=[4 5 6]
f =
     4     5     6
>> g=e*f
??? Error using ==> mtimes
Inner matrix dimensions must agree.
>> h=e.*f
h =
     4    10    18
>> k=e+f
k =
     5     7     9
>> m=e^3
??? Error using ==> mpower
Inputs must be a scalar
and a square matrix.
To compute elementwise
POWER, use POWER (.^ )
instead.
>> m=e.^3
m =
     1     8    27
>>
```

关系与逻辑运算

1. 关系操作符

关系操作符	说明
<	小于
<=	小于或等于
>	大于
>=	大于或等于
= =	等于
~=	不等于

- ❖ 当两个比较量是标量时，直接比较两数的大小。若关系成立，关系表达式的值为1，否则值为0。
- ❖ 当参与比较的量是两个同型的矩阵时，比较时对两矩阵相同位置的元素按标量关系运算规则逐个进行，最终的关系运算的结果是一个与原矩阵同型的矩阵，它的元素由0或1组成。
- ❖ 当参与比较的一个是标量，而另一个是矩阵时，则把标量与矩阵的每一个元素按标量关系运算规则逐个比较，最终的关系运算的结果是一个与原矩阵同型的矩阵，它的元素由0或1组成。

建立2行4列的矩阵，判断哪些元素是偶数。

❖ `>> a=[23 46 65 78;21 34 56 99]`

❖ `a =`

❖ `23 46 65 78`

❖ `21 34 56 99`

❖ `>> b=rem(a,2)==0`

❖ `b =`

❖ `0 1 0 1`

❖ `0 1 1 0`

❖ `b`中元素值为1的对应`a`矩阵中的元素即为偶数。

2. 逻辑运算符

逻辑操作符	说明
&	与
	或
~	非

- ❖ 设 a 、 b 为参与逻辑运算的是两个标量，那么运算规则为：
 - $a \& b$: a 、 b 全为非零时，运算结果为1，否则为0。
 - $a | b$: a 、 b 中只要有一个为非零时，运算结果为1。
 - $\sim a$: 当 a 为零时，运算结果为1；当 a 为非零时，运算结果为0。
- ❖ 若参与逻辑运算的是两个同型矩阵，那么将对矩阵相同位置上的元素按标量规则逐个进行运算，最终运算结果是一个与原矩阵同型的矩阵，其元素由1或0组成。
- ❖ 若参与逻辑运算的两个运算对象一个是标量，另一个是矩阵，那么将在标量与矩阵中的每个元素之间按标量规则逐个进行运算，最终运算结果是一个与矩阵同型的矩阵，其元素由1或0组成。

❖ >> a=[1 2 3 4;5 6 7 8;9 10 11 12]

❖ a =

❖ 1 2 3 4

❖ 5 6 7 8

❖ 9 10 11 12

❖ >>a>6

❖ a =

❖ 0 0 0 0

❖ 0 0 1 1

❖ 1 1 1 1

顺序结构程序设计

- ❖ 1、数据的输入
- ❖ 2、数据的输出
- ❖ 3、程序的暂停

1、数据的输入

- ❖ 从键盘输入数据，则可使用input函数来实现，其格式：
- ❖ `A=input(提示信息, 选项);`
 - 其中，提示信息为字符串，用于提示用户输入什么样的数据。例如：
- ❖ `A=input('请输入A矩阵:');`
 - 如果在input函数调用时采用's'选项，则允许用户输入一个字符串。例如，想输入一个人的姓名，可采用命令：
- ❖ `xm = input ('What"s your name?', 's')`

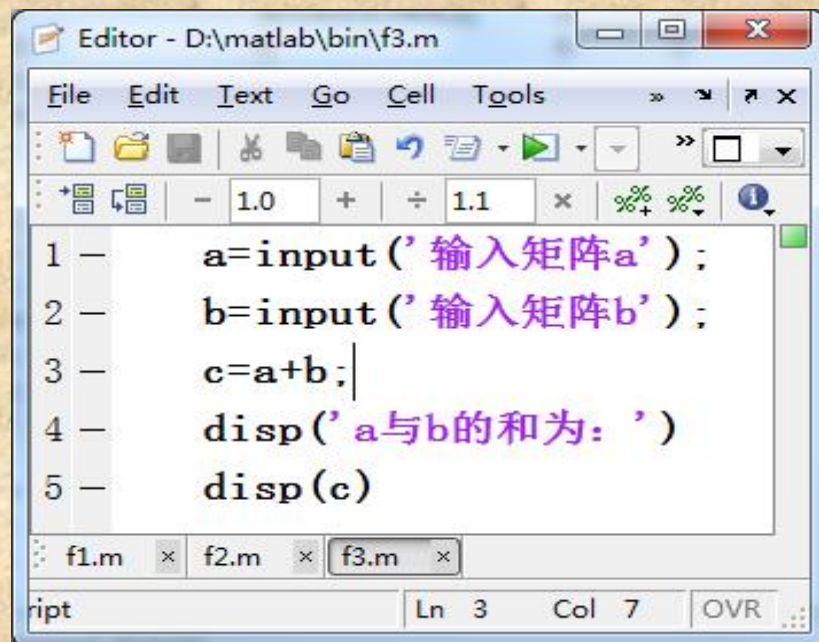
2、数据的输出

- ❖ Matlab提供的命令窗口输出函数主要有disp函数和fprintf函数。

(1) disp函数

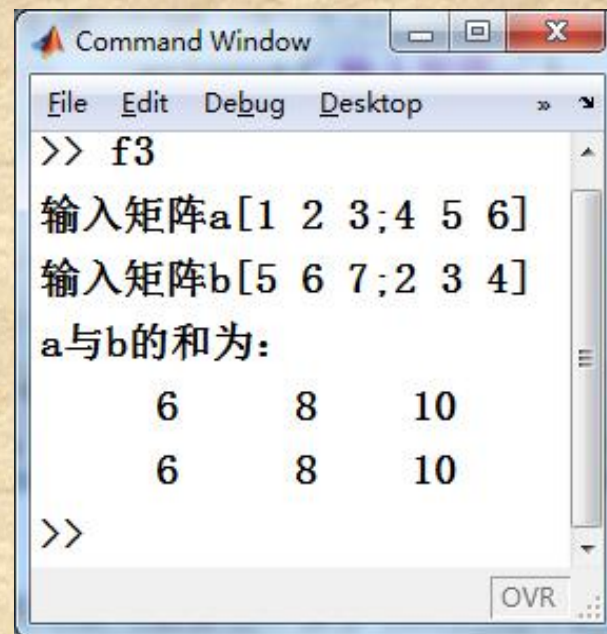
- ❖ (1) disp函数将数据输出到matlab的命令窗口
 - disp函数的调用格式为:
 - disp(输出项)
 - 其中，输出项既可以是字符串、也可以是矩阵。
 - 用disp函数显示矩阵时将不显示矩阵的名字，而且其格式更紧密，且不留任何没有意义的空行。

输入和输出举例



The image shows the MATLAB Editor window with the file 'f3.m' open. The script contains five lines of code: line 1: `a=input('输入矩阵a');`, line 2: `b=input('输入矩阵b');`, line 3: `c=a+b;`, line 4: `disp('a与b的和:')`, and line 5: `disp(c)`. The status bar at the bottom indicates the cursor is at line 3, column 7.

```
1 — a=input('输入矩阵a');  
2 — b=input('输入矩阵b');  
3 — c=a+b;  
4 — disp('a与b的和:');  
5 — disp(c)
```



The image shows the MATLAB Command Window with the command `>> f3` entered. The output displays the prompts for matrices 'a' and 'b', followed by the sum 'c'. The sum is shown as a 2x3 matrix with values 6, 8, and 10 in each row.

```
>> f3  
输入矩阵a[1 2 3;4 5 6]  
输入矩阵b[5 6 7;2 3 4]  
a与b的和为:  
      6      8     10  
      6      8     10  
  
>>
```


(2) fprintf函数

- ❖ (2) 用fprintf函数格式化输出数据到文件中
- ❖ fprintf函数显示带有相关文本的一个或多个值，允许程序员控制显示数据的方式。它在命令行窗口打印一个数据的一般格式如下：
- ❖ fprintf(文件句柄fid,格式format,数据data)
 - 其中fid表示由fopen函数打开的文件句柄，如果fid省略，则直接输出在屏幕上，format用于表示一个描述打印数据方式的字符串，data代表要打印的一个或多个标量或数组。format包括两方面的内容，一方面是打印文本内容；另一方面是打印内容中的数据格式。

fprintf 举例

❖ `>> fprintf('the value of pi is%6.2f\n',pi)`

❖ the value of pi is 3.14

- 打印的结果为the value of pi is 3.14，后面带有一个换行符。
- 转义序列%6.2f代表在本函数中的第一个数据项将占有6个字符宽度，小数点后有2位小数。

❖ fprintf函数有一个重大的局限性，只能显示复数的实部。当我们的计算结果是复数时，这个局限性将会产生错误。在这种情况下，最好用disp显示数据。

format命令中格式符的意义:

表 format命令中的格式符

格式符	功能
%d	把值作为整数来处理
%e	用科学记数法来显示数据
%f	用于格式化浮点数，并显示这个数
%g	用科学记数格式，或浮点数格式，根据长度最短的显示
%n	换行符

3、程序的暂停

❖ Matlab中程序暂停函数的调用格式:

❖ `pause`(延时秒数)

- 若省去延时秒数，直到用户按任意键程序继续执行，按Ctrl+C强行中止程序的执行。

选择结构程序设计

- ❖ MATLAB中可以用3种语句实现选择结构
- ❖ if语句
- ❖ switch语句
- ❖ try语句

if语句

- ❖ 单分支
- ❖ 双分支
- ❖ 多分支

(1) 单分支if语句

❖ 格式:

- if 条件
- 语句组
- end
- 如果条件成立就执行语句组，条件通常是关系表达式或逻辑表达式，当条件结果为标量时，非零表示条件成立，零表示条件不成立；
- 当条件结果为矩阵时，如果矩阵为非空，且不包含零元素，条件成立，否则不成立。
- [1,2;0,3]表示条件时，条件不成立；[1,2;3,4]表示条件时，条件成立。

(2) 双分支if语句

❖ (2) 双分支if语句

❖ 格式:

- if 条件
- 语句组1
- else
- 语句组2
- end

❖ 如果条件成立则执行语句组1，否则执行语句组2。

(3) 多分支if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

【例】 编写程序完成以下分段函数，要求输入x的值，输出相应的y值。

```
Editor - E:\matlabexers\l7sd.m
File Edit Text Go Cell Tools Debug Desktop
1 %符号函数
2 x=input('输入x');
3 if x>0
4     y=1;
5 elseif x==0
6     y=0;
7 else
8     y=-1;
9 end
10 disp(['x=', num2str(x), ', y=', num2str(y)])
```

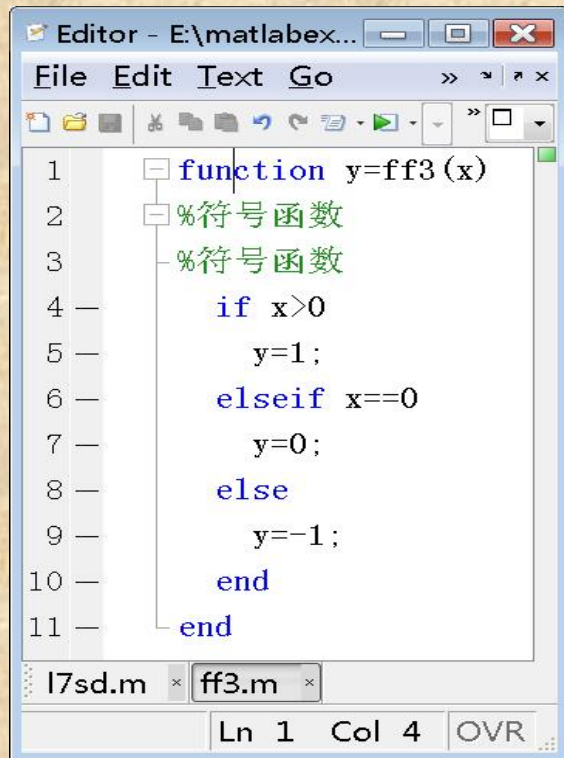
```
Co...
File Edit
>> l7sd
输入x6
x=6, y=1
>> l7sd
输入x-3
x=-3, y=-1
>> l7sd
输入x0
x=0, y=0
>> |
```

$$y = \begin{cases} 1 & (x > 0) \\ 0 & (x = 0) \\ -1 & (x < 0) \end{cases}$$

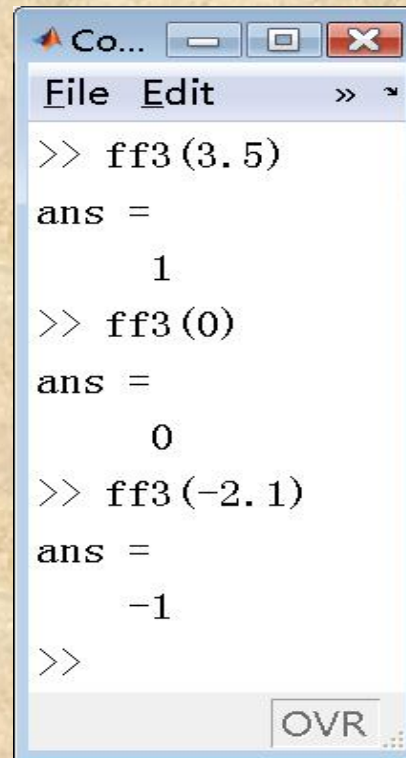
【例】编写函数，完成符号函数的功能

- ❖ 首先要定义函数，计算符号函数时，需要已知 x ，因此函数要有一个形式参数 x ，再给函数一个名字，这里用ff3，这样，函数首部即可确定为function y=ff3(x)，
- ❖ 当 x 为形参时，它的值是调用函数时传递过来的，所以在函数中无需也不能再输入 x ，即当 x 为已知的，
- ❖ 求完函数值也不在函数内打印，而是由函数名带回到调用它的位置。

【例】编写函数，完成符号函数的功能



```
Editor - E:\matlabex...
File Edit Text Go
1 function y=ff3(x)
2 %符号函数
3 %符号函数
4 if x>0
5     y=1;
6 elseif x==0
7     y=0;
8 else
9     y=-1;
10 end
11 end
l7sd.m x ff3.m x
Ln 1 Col 4 OVR
```



```
Co...
File Edit
>> ff3(3.5)
ans =
     1
>> ff3(0)
ans =
     0
>> ff3(-2.1)
ans =
    -1
>>
OVR
```


例 设 $f(x) = \begin{cases} x^2 + 1 & x > 1 \\ 2x & x \leq 1 \end{cases}$, 求 $f(2), f(-1)$

MATLAB命令窗口输入 **fun1 (2) , fun1 (-1)** 即可.

MATLAB(fun1)

例 设 $f(x) = \begin{cases} x^2 + 1 & x > 1 \\ 2x & 0 < x \leq 1 \\ x^3 & x \leq 0 \end{cases}$, 求 $f(2), f(0.5), f(-1)$

先建立M文件**fun2.m**来定义函数 **$f(x)$** , 再在
MATLAB命令窗口输入**fun2 (2) , fun2 (0.5) ,**
fun2 (-1) 即可.

MATLAB(fun2)

2、switch语句

- switch语句是根据表达式的取值不同，分别执行不同的语句，其语句格式：

- **switch** 表达式
- **case** 表达式1
- 语句组1
- **case** 表达式2
- 语句组2
-
- **case** 表达式m
- 语句组m
- **otherwise**
- 语句组n
- **end**

Switch语句举例

- ❖ 输入某个学生的成绩 g （假设 $0 \leq g \leq 100$ ）。如果 $g \geq 90$ ，输出“A”； $80 \leq g < 90$ ，输出“B”； $70 \leq g < 80$ ，输出“C”； $60 \leq g < 70$ ，输出“D”； $g < 60$ ，输出“E”。
- ❖ 建立脚本文件，取名swi1.m，程序代码如下：

成绩等级

```
❖ x=input('输入整数成绩');  
❖ switch(fix(x/10))  
❖     case {9,10}  
❖         disp('A')  
❖     case 8  
❖         disp('B')  
❖     case 7  
❖         disp('C')  
❖     case 6  
❖         disp('D')  
❖     otherwise  
❖         disp('E')  
❖ end
```

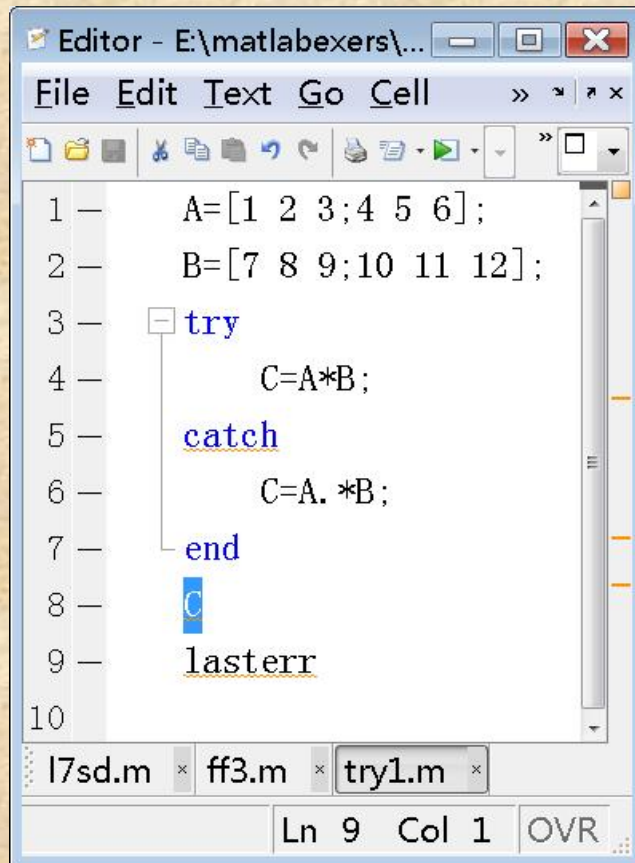

3、try语句

❖ try语句是一种试探性执行语句，其语句格式：

- try
- 语句组1
- catch
- 语句组2
- end

❖ try语句先试探执行语句组1，如果语句组1在执行过程中出现错误，则将错误信息保留在**lasterr**变量中，并转去执行语句组2。

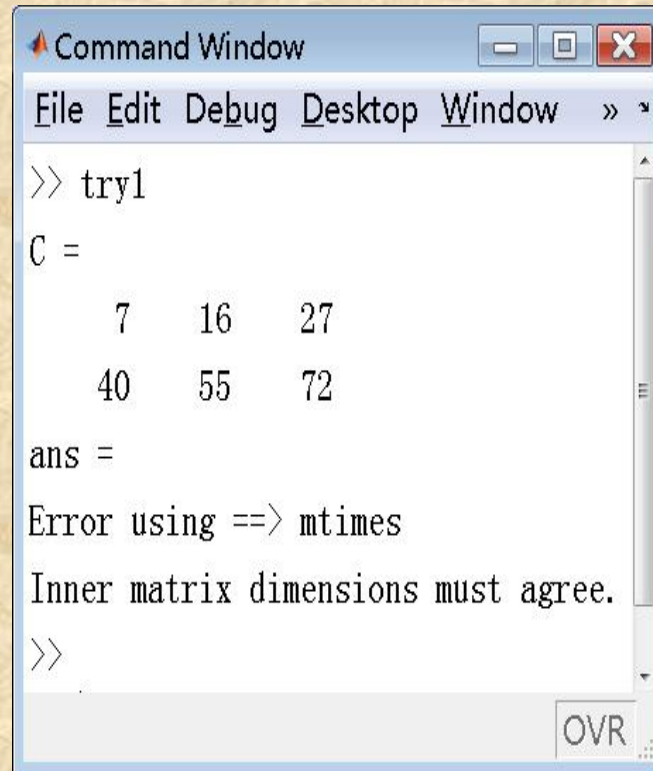
Try语句举例



The image shows a MATLAB Editor window with the following code:

```
1 A=[1 2 3;4 5 6];  
2 B=[7 8 9;10 11 12];  
3 try  
4     C=A*B;  
5 catch  
6     C=A.*B;  
7 end  
8 C  
9 lasterr  
10
```

The status bar at the bottom indicates the cursor is at Line 9, Column 1.



The image shows the MATLAB Command Window with the following output:

```
>> try1  
C =  
  
     7     16     27  
    40     55     72  
  
ans =  
Error using ==> mtimes  
Inner matrix dimensions must agree.  
>>
```

The status bar at the bottom right indicates the cursor is at the end of the line.

3.4 循环结构程序设计

- ❖ 1、for语句
- ❖ 2、while语句
- ❖ 3、break语句和continue语句

for语句

❖ 1、for语句

❖ (1) 简单格式:

❖ for 循环变量=循环初值: 步长: 终值

❖ 循环体语句

❖ end

1. **for**循环: 允许一组命令以固定的和预定的次数重复

```
for x=array  
    {commands}  
end
```

在**for**和**end**语句之间的命令串{**commands**}按数组(**array**)中的每一列执行一次. 在每一次迭代中, **x**被指定为数组的下一列, 即在第**n**次循环中, **x=array(:, n)**

例 对**n=1,2,...,10**,求 $x_n = \sin \frac{n \cdot \pi}{10}$ 的值.

MATLAB(for1)

2、while语句

- ❖ while语句的一般格式为：
- ❖ while (条件)
- ❖ 循环体语句
- ❖ end
- ❖ 其执行过程为：若条件成立，则执行循环体语句，执行后再判断条件是否成立，如果不成立则跳出循环

2. While循环

与for循环以固定次数求一组命令相反，while循环以不定的次数求一组语句的值。

```
while (expression)
    { commands }
end
```

只要在表达式 (expression) 里的所有元素为真，就执行while和end语句之间的命令串 { commands }。

例 设银行年利率为11.25%.将10000元钱存入银行，问多长时间会连本带利翻一番？

MATLAB(while1)

3、 **break**语句和**continue**语句

- ❖ **break**语句用来跳出循环体，结束整个循环。
- ❖ **continue**语句用来结束本次循环，接着进行下一次是否执行循环的判断。该语句一般与**if**语句配合使用。
- ❖ 在多重循环中，**break**只能使程序跳出包含它的最内层的那个循环。

【例】 求 [100, 200]之间第一个能被33整除的整数

❖ for n=100:200

❖ if rem(n,33)~=0

❖ continue

❖ end

❖ n

❖ break

❖ end

■ 运行结果为:

■ n=

■ 132

实验作业

对以下问题,编写M文件:

(1)用起泡法对10个数由小到大排序.即将相邻两个数比较,将小的调到前头.

(2)有一个 4×5 矩阵,编程求出其最大值及其所处的位置.

(3)编程求 $\sum_{n=1}^{20} n!$

(4)一球从100米高度自由落下,每次落地后反跳回原高度的一半,再落下.求它在第10次落地时,共经过多少米?第10次反弹有多高?

(5)有一函数 $f(x, y) = x^2 + \sin xy + 2y$,写一程序,输入自变量的值,输出函数值.

谢谢！

