

MATLAB编程指导

《计算方法》课程实验



HITSZ 实验与创新实践教育中心
Education Center of Experiments and Innovations, HITSZ

目录

- 一、Matlab的工作环境
- 二、Matlab的基础操作
- 三、Matlab的程序设计

1.1 Matlab的概述

1.2 Matlab的启动退出

1.3 Matlab主界面

1.4 Matlab通用命令

1.5 Matlab常用标点符号

1.6 Matlab帮助系统

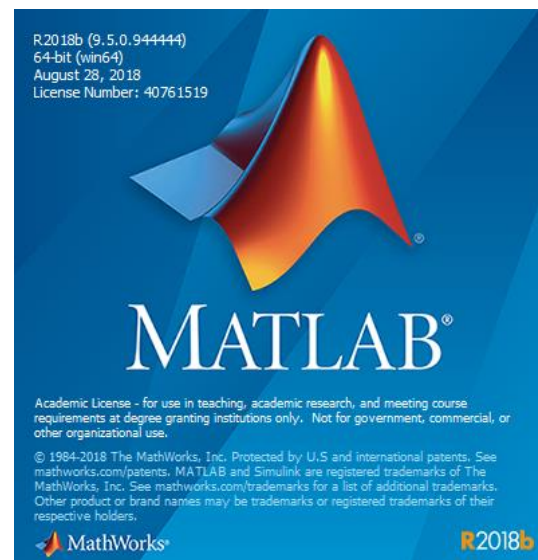
一、Matlab的工作环境

1.1 Matlab的概述

Matlab是Matrix Laboratory(矩阵实验室)的缩写，是由美国MathWorks公司于1984年推出的一套**以矩阵计算为基础的**、适合多学科、多种工作平台的功能强劲的大型软件。

Matlab的基本功能有：**数学计算功能**、**图形化显示功能**、**M语言编程功能**、编译功能、图形用户界面开发功能、并行计算功能、Simulink建模仿真功能等等。

Matlab版本：Matlab2018b版本（实验室环境）



一、Matlab的工作环境

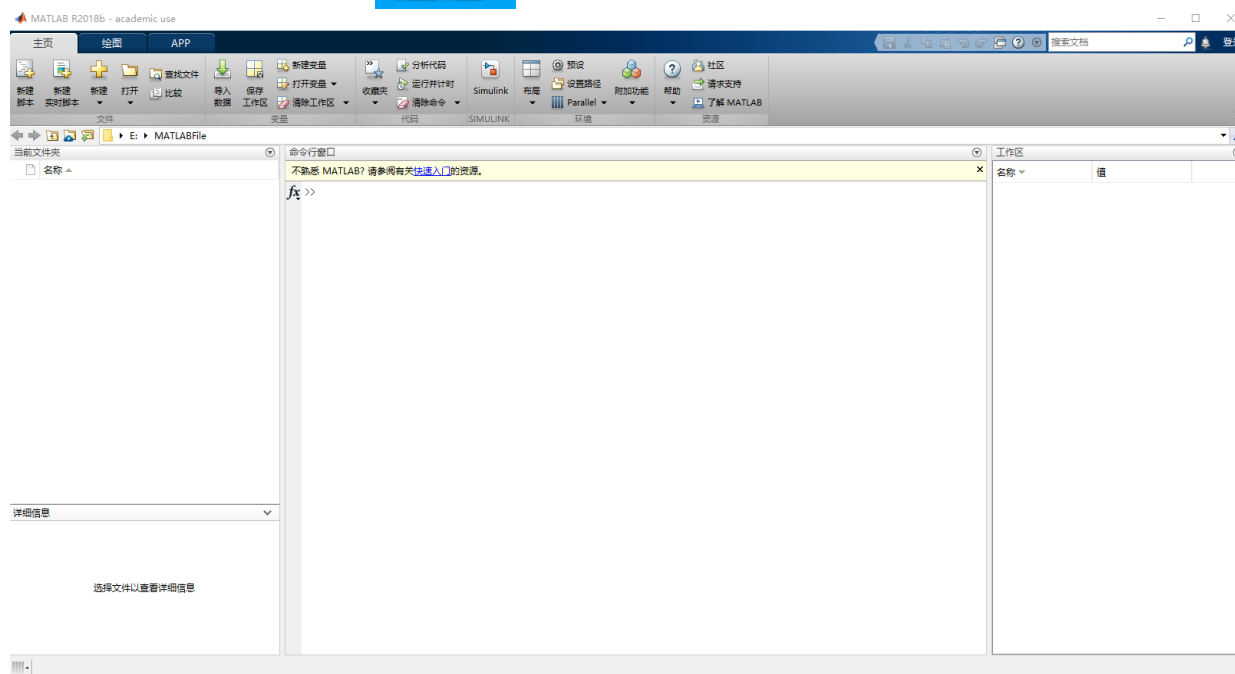
1.2 Matlab的启动和退出

➤ 启动


双击电脑桌面上的MATLAB图标



, 进入Matlab程序;

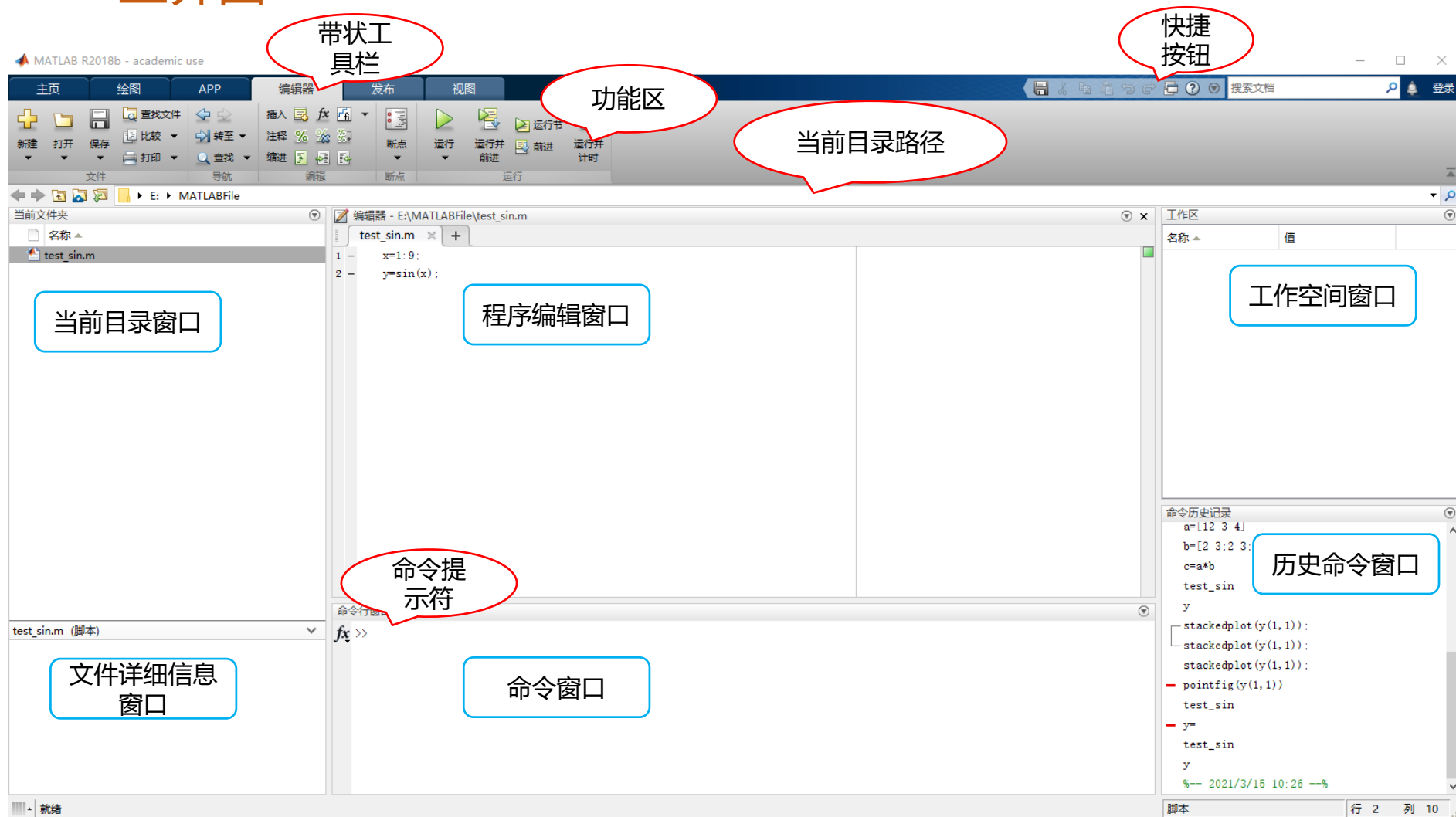


➤ 退出

单击窗口右上角的关闭图标  , 退出Matlab程序。

一、Matlab的工作环境

1.3 Matlab主界面



一、Matlab的工作环境

(1) 工具栏



(2) 快捷按钮

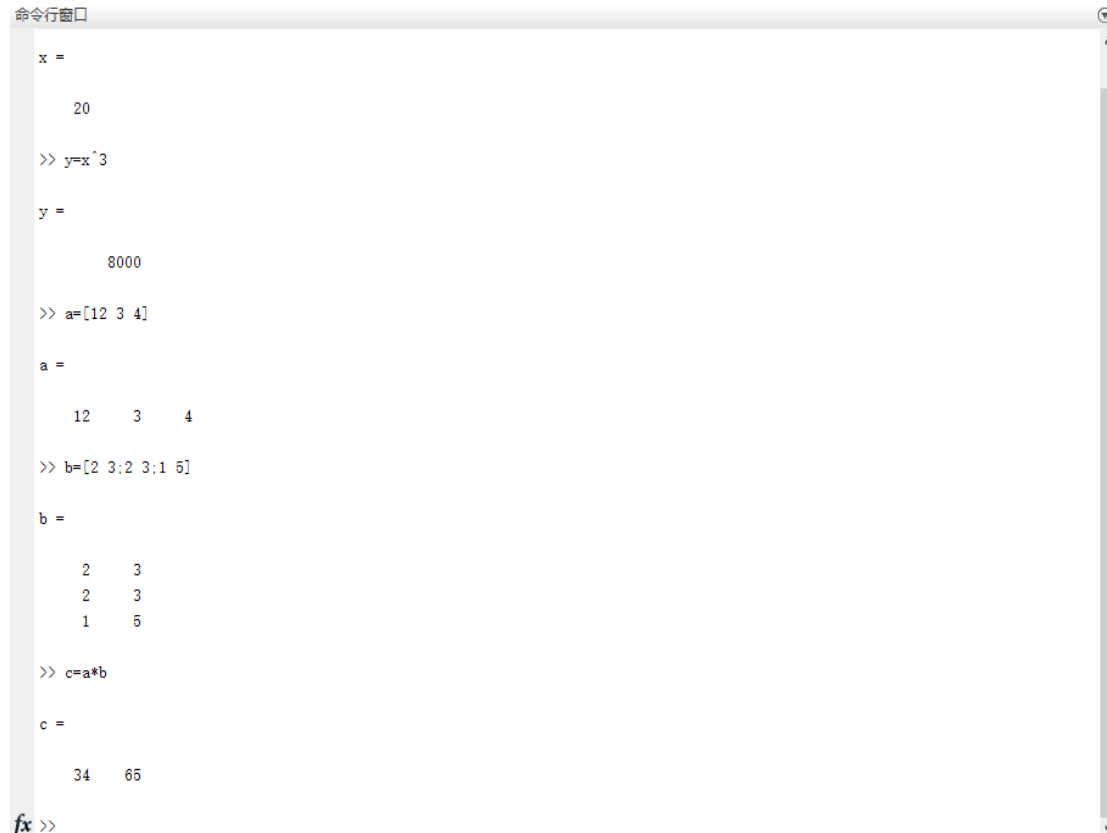


分别对应的功能为“保存”，“剪切”，“复制”，“粘贴”，“撤销”，“重做”，“切换窗口”以及“帮助”。

一、Matlab的工作环境

(3) 命令窗口

用户可以直接在Matlab命令窗口中输入Matlab命令，实现其相应的功能。



```
命令窗口
x =
    20

>> y=x^3

y =
    8000

>> a=[12 3 4]

a =
    12     3     4

>> b=[2 3:2 3:1 5]

b =
     2     3
     2     3
     1     5

>> c=a*b

c =
    34    65

fx >>
```

命令窗口中常用的快捷键：

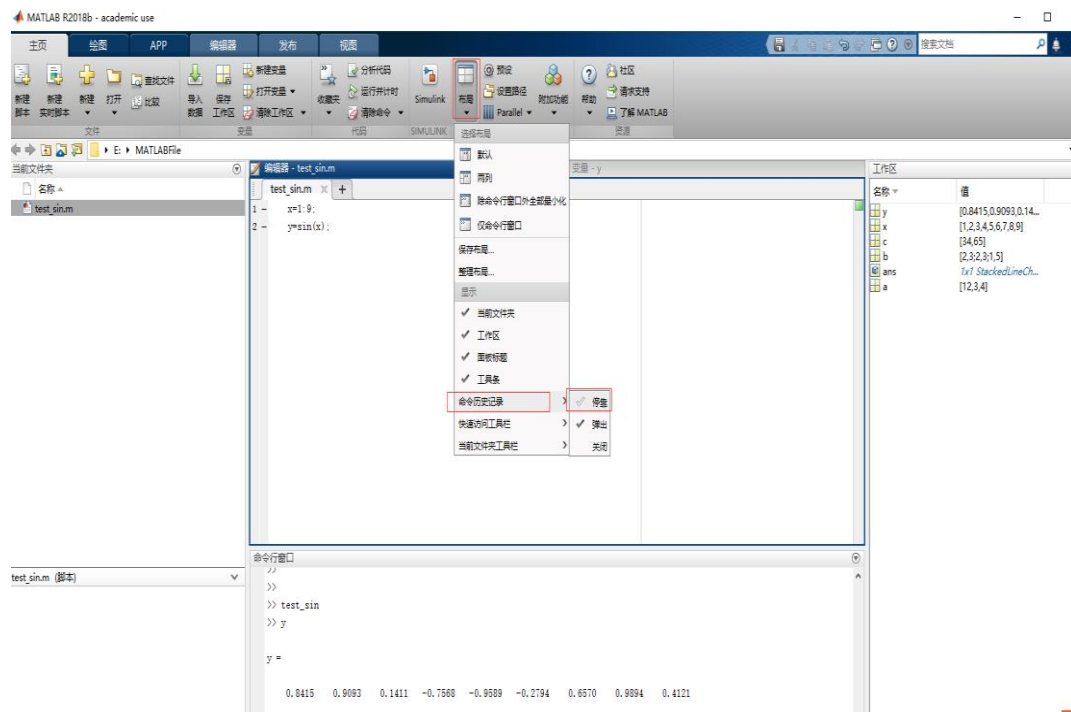
快捷键	说 明
方向键↑	调出历史命令中的前一个命令
方向键↓	调出历史命令中的后一个命令
Tab 键	输入命令的前几个字符，然后按 Tab 键，会弹出前面包含这几个字符的所有命令，方便查找所需命令
Ctrl+C	中断程序的运行，用于耗时过长程序的紧急中断

一、Matlab的工作环境

(4) 历史命令窗口

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的时间和运行的所有命令行。

Matlab默认是不显示历史命令窗口的，可以从工具栏中的“布局”选项进入，设置显示历史命令窗口，如下操作：



结果如下：

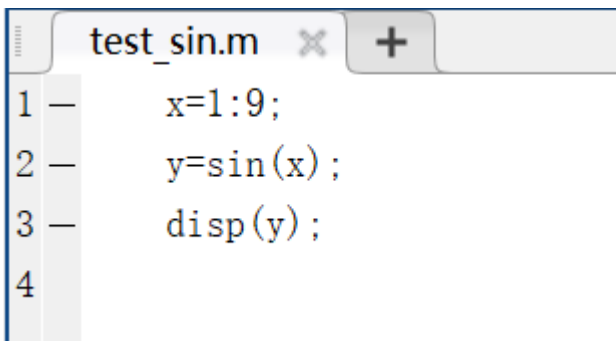


另外单击右键，可以对历史命令进行编辑（剪切/复制/运行/创建m文件/等）。

一、Matlab的工作环境

(5) 程序编辑窗口

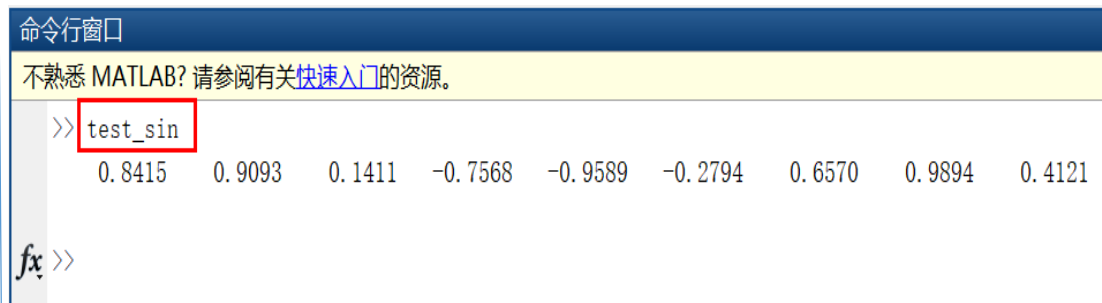
扩展名为.m的文件，称之为M文件。M文件通常在程序编辑窗口（或称脚本编辑窗口）中编写，也可在记事本、写字板等文本编辑工具中编写，只需保存成M文件即可。



```
test_sin.m x +
1 — x=1:9;
2 — y=sin(x);
3 — disp(y);
4
```

M文件的调用：

在命令窗口**输入定义好的函数名**，如test_sin，**回车**，就能得到结果。

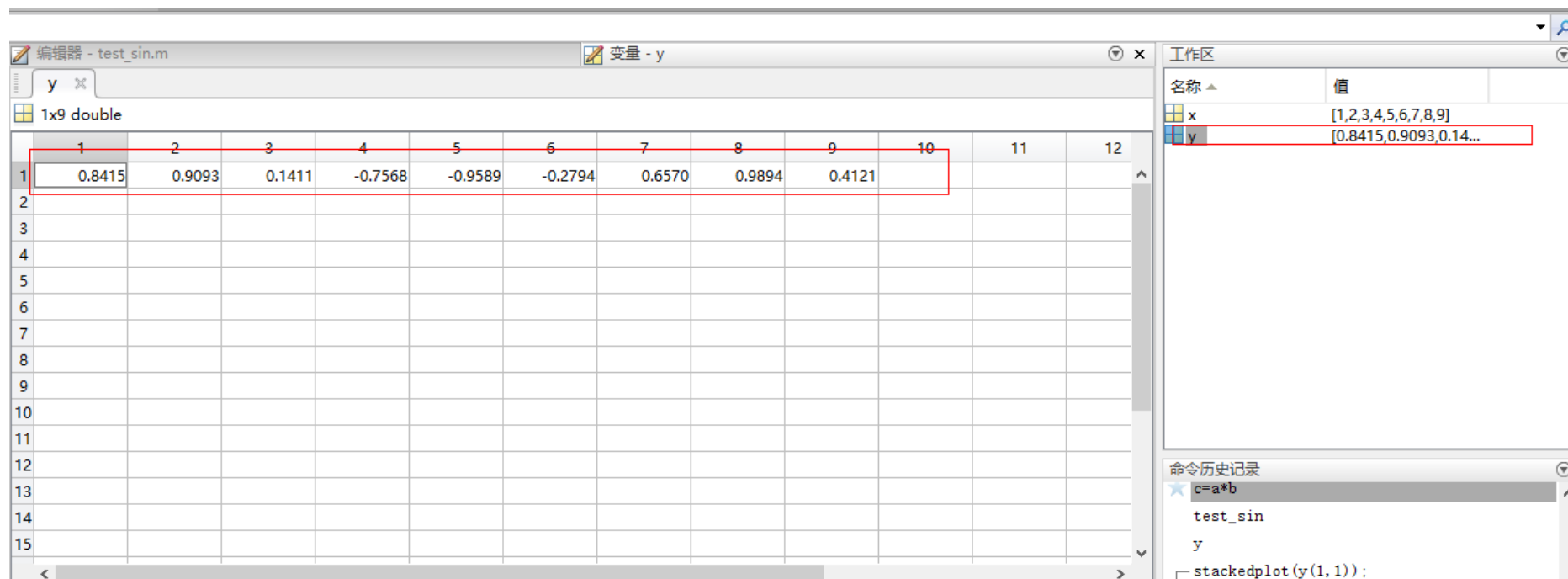


```
命令行窗口
不熟悉 MATLAB? 请参阅有关快速入门的资源。
>> test_sin
    0.8415    0.9093    0.1411   -0.7568   -0.9589   -0.2794    0.6570    0.9894    0.4121
fx >>
```

一、Matlab的工作环境

(6) 工作空间窗口

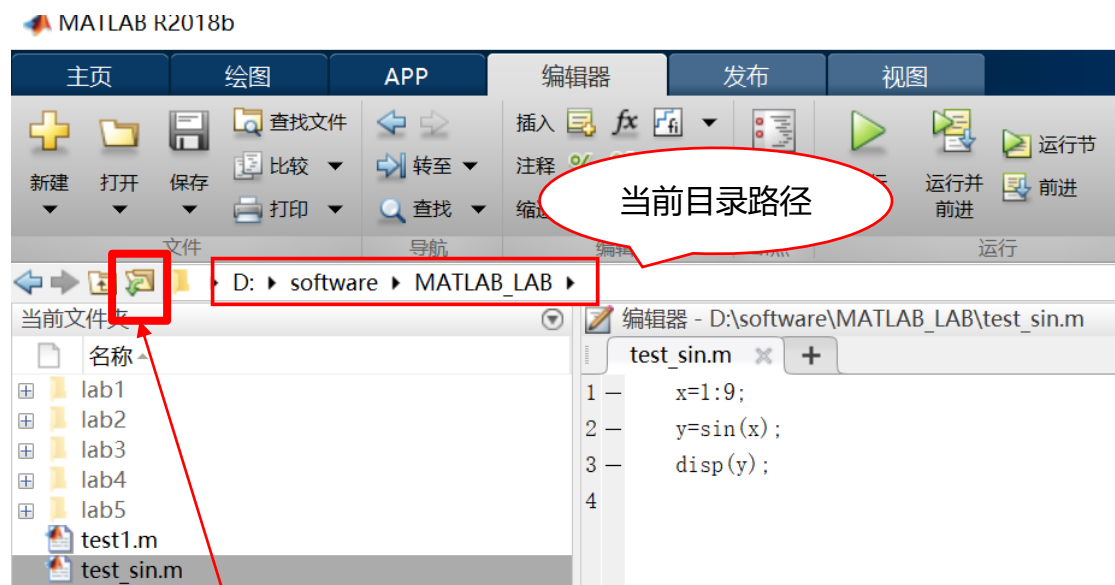
功能：查询和编辑已定义变量；通过右键菜单进行编辑或绘图等相关操作。



一、Matlab的工作环境

重点!

(7) 当前目录路径



更新路径的方法：选中图标 ，在弹出的对话框中，选择新文件夹路径，为当前工作目录路径。

一、Matlab的工作环境

重点!

1.4 Matlab通用命令

命令	说明	命令	说明
cd	改变当前目录	!	调用DOS命令
dir或ls	列出当前文件夹下的文件	edit	打开M文件编辑器
clc	清除命令行窗口的内容	mkdir	创建目录
type	显示文件内容	pwd	显示当前工作目录
clear	清除工作空间中的变量	what	显示当前目录下的M文件、MAT和MEX文件
disp	显示文字内容	which	函数或文件的位置
exit或quit	关闭MATLAB	help	获取函数的帮助信息
save	保存变量到磁盘	pack	收集内存碎片
load	从磁盘调入数据变量	path或genpath	显示搜索路径
who	列出工作空间中的变量名	clf	清除图形窗口的内容
whos	显示变量的详细信息	delete	删除文件

一、Matlab的工作环境

1.5 Matlab常用标点符号

名 称	标 点	功能说明
空格		数组元素或输入量之间的分隔符
逗号	,	数组元素或输入量之间的分隔符
黑点	.	小数点；结构体数组的字段标识符；点运算标识符。
分号	;	定义数组时，作为行间分隔符；用在某条命令的“结尾”，不显示计算结果
冒号	:	作为冒号运算符，用来生成一维数组；作为数组单下标引用时，表示将数组按列拉长为长向量；作为数组多下标引用时，表示该维上的所有元素
注释号	%	注释内容引导符
单引号对	' '	字符串标记符
圆括号	()	用来访问数组元素；用来标记运算作用域；定义函数时用来标记输入变量列表
方括号	[]	用来定义数组；定义函数时用来标记输出变量列表
下连符	_	作为变量、函数或文件名中的连字符；图形对象中下脚标前导符
续行号	...	由三个以上连续黑点构成。它把其下的命令行看作该行的延续，以构成一个“较长”的完整命令
“@”号	@	放在函数名前，形成函数句柄；匿名函数前导符；放在目录名前，形成“用户对象”类目录。

一、Matlab的工作环境

1.6 Matlab帮助系统

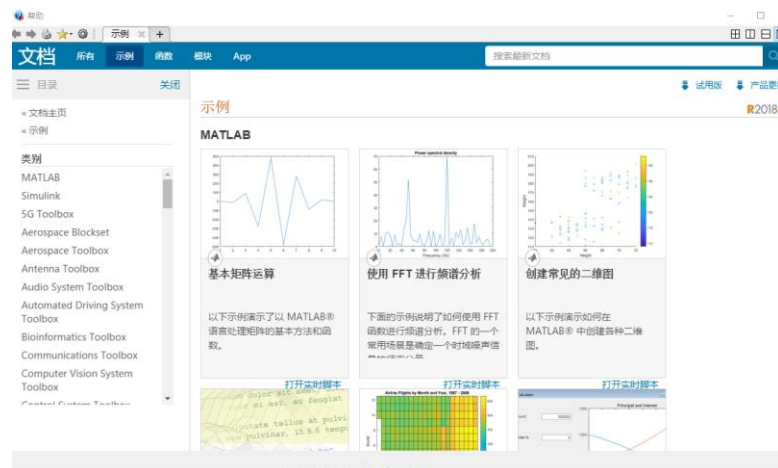
➤ Matlab命令窗口帮助系统

Matlab中提供了help、helpbrowser、helpwin、doc、docsearch和lookfor等函数，用来在命令窗口中查询函数的帮助信息。如：



➤ Help帮助浏览器

在工具栏中选择“帮助”按钮，进入后选择“示例”选项，点击进入Help帮助浏览器。



目录

- 一、Matlab的工作环境
- 二、Matlab的基础操作
- 三、Matlab的程序设计

2.1 变量的定义与数据类型

2.2 常用函数

2.3 数组运算

2.4 矩阵运算

二、Matlab的基础操作

2.1 变量的定义与数据类型

(1) 变量的定义与赋值

➤ 变量命名规则

- ✓ 可由任意的**字母、数字或下划线**组成，但必须**以字母打头**；
- ✓ 变量名区分字母大小写；
- ✓ 变量名不超过63个字符。

注：不要用内部函数名作为变量名。

➤ 赋值语句，举例：

```
>> x=1
x =
    1
>> y=1+2+sqrt(9)
y =
    6
>> z='Hello World!!!'
z =
'Hello World!!!'
```


二、Matlab的基础操作

(2) Matlab中的常量

➤ Matlab中的特殊函数或常量列表

在变量名缺省的情况下，计算结果被赋给变量ans，ans是一个内部函数。Matlab中提供了一些特殊函数，它们的返回值是一些有用的常量。

特殊函数（常量）	意义
ans	用于存储计算结果的默认变量名
pi	圆周率 π (= 3.1415926...)
i 或 j	虚数单位 ($\sqrt{-1}$)
inf 或 Inf	无穷大 (∞)，正数除以0 的结果
NaN 或 nan	非数（或不定量），0/0、inf/inf 或inf-inf
eps	浮点运算的相对精度， $\varepsilon = 2^{-52}$
realmin	最小的正浮点数 2^{-1022}
realmax	最大的正浮点数 $(2 - \varepsilon) 2^{1023}$
version	MATLAB 版本信息字符串，例如9.5.0.944444 (R2018b)

➤ 清除变量和恢复内部函数

```
>> pi % 查看圆周率的值
>> pi = 1 % 对变量pi重新赋值
>> clear pi % 清除变量pi
>> pi
ans = 3.1416
```

二、Matlab的基础操作

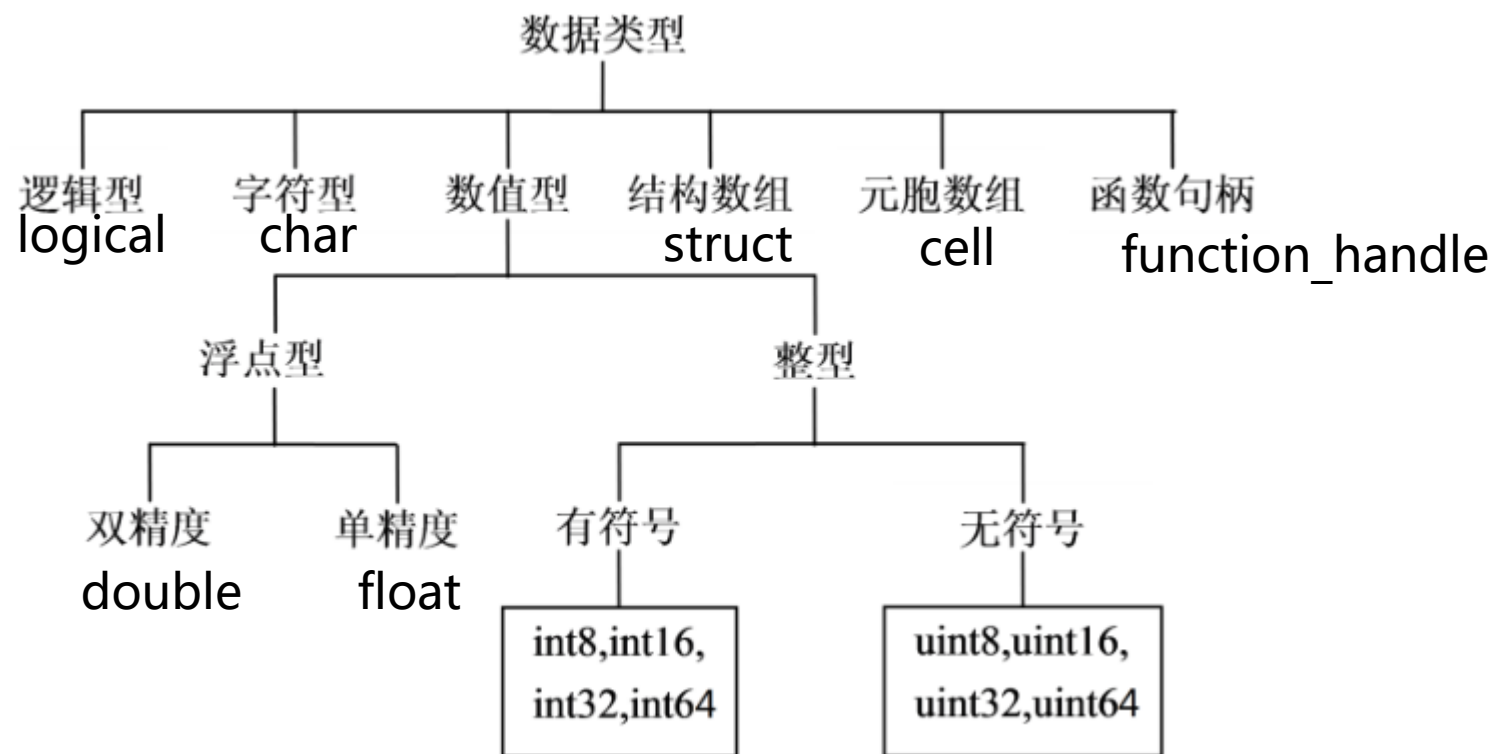
(3) Matlab中的关键字

作为一种编程语言，Matlab中为编程保留了一些关键字：`break`、`case`、`catch`、`classdef`、`continue`、`else`、`elseif`、`end`、`for`、`function`、`global`、`if`、`otherwise`、`parfor`、`persistent`、`return`、`spmd`、`switch`、`try`、`while`，这些关键字在程序编辑窗口中会以蓝色显示，它们是**不能作为变量名的**，否则会出现错误。

二、Matlab的基础操作

(4) 数据类型

Matlab中有15种基本的数据类型，有逻辑型、字符型、整型、浮点型、结构数组、元胞数组以及函数句柄等。其中整型又分为有符号整型和无符号整型，8位整型、16位整型、32位整型和64位整型，浮点型又分为单精度浮点型和双精度浮点型。具体可以通过Matlab中自带的isa函数查看。



二、Matlab的基础操作

(5) 数据输入

Matlab中数据输入：**input函数**，其格式：A=input(提示信息，选项) 具体有两种应用：

x = **input(prompt)** 输入的东西当成数字或者矩阵

str = **input(prompt, 's')** 输入的东西当成字符串存起来

举例：

```
>> a=input('请输入矩阵a=')
```

```
请输入矩阵a=[1,2,3;4,5,6]
```

```
a =
```

```
1    2    3
```

```
4    5    6
```

```
>> b=input('请输入数据b=')
```

```
请输入数据b=20220317
```

```
b =
```

```
20220317
```

```
>> c=input('请输入姓名c=','s')
```

```
请输入姓名c=匡老师
```

```
c =
```

```
'匡老师'
```

二、Matlab的基础操作

(6) 数据输出

Matlab中数据输出：**disp函数**和**fprintf函数**。其格式：

◆ disp(输出项) （注意：输出项既可以是字符串、也可以是矩阵）

例如：

```
f3.m x +
1 — a=input('请输入矩阵a=');
2 — b=input('请输入矩阵b=');
3 — c=a+b;
4 — disp('矩阵a与b的和为: ')
5 — disp(c)
6
```

命令行窗口

不熟悉 MATLAB? 请参阅有关[快速入门](#)的资源。

```
>> f3
请输入矩阵a=[1, 2; 3, 4]
请输入矩阵b=[1, 2; 3, 4]
矩阵a与b的和为:
     2     4
     6     8
```

fx >>

二、Matlab的基础操作

(6) 数据输出

Matlab中数据输出：**disp函数和fprintf函数**。其格式：

◆ fprintf(文件句柄fid, 格式format, 数据data)

参数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，

参数 format用于表示一个描述打印数据方式的字符串，

参数data代表要打印的一个或多个标量或数组

例如：

```
>> fprintf('the value of pi is%6.2f\n',pi)
the value of pi is  3.14
fx >>
```

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

注意：fprintf函数只能显示复数的实部，在这种情况下，最好用disp显示数据。

二、Matlab的基础操作

重点!

2.2 常用函数

函数名	说明	函数名	说明
abs	绝对值或复数的模	sqrt	平方根函数
exp	指数函数	log	自然对数
log2	以2为底的对数	log10	以10为底的对数
round	四舍五入到最接近的正整数	ceil	向正无穷方向取整
floor	向负无穷方向取整	fix	向0方向取整
rem	求余函数	mod	取模函数
sin	正弦函数	cos	余弦函数
tan	正切函数	cot	余切函数
asin	反正弦函数	acos	反余弦函数
atan	反正切函数	acot	反余切函数
real	求复数实部	imag	求复数虚部
angle	求相位角	conj	求共轭复数
mean	求均值	std	求标准差
max	求最大值	min	求最小值
var	求方差	cov	求协方差

举例:

```
>> x=[1 -1.65 2.2 -3.1];
>> y1=abs(x)
y1 =
    1.0000    1.6500    2.2000    3.1000
>> y2=sin(x)
y2 =
    0.8415   -0.9969    0.8085   -0.0416
>> y3=round(x)
y3 =
     1     -2     2     -3
>> y4=floor(x)
y4 =
     1     -2     2     -4
>> y5=ceil(x)
y5 =
     1     -1     3     -3
>> y6=min(x)
y6 =
   -3.1000
>> y7=mean(x)
y7 =
   -0.3875
```

二、Matlab的基础操作

重点!

2.2 常用函数（举例）

(1) **sym**函数：用来创建单个符号量，
如`sym x`;

(2) **syms**函数：可以在一条语句中定义多个符号变量，如`syms x,y,z`;

(3) **subs**函数：用于实现变量的替换，
如`subs(f,[x,y],[1,2])`

(4) **diff**函数：对原函数求导数

.....

计算某一定点 x_0 处 $f(x_0)$ 和导数 $df(x_0)$ 的值，有两种方式，举例：

```
>> syms x;  
>> f=cos(x)-x  
  
f = cos(x) - x  
  
>> x0=pi/4;  
>> fx0=subs(f,symvar(f),x0)  
  
fx0 = 2^(1/2)/2 - pi/4  
  
>> df=diff(f)  
  
df = - sin(x) - 1  
  
>> df0=subs(df,x,x0)  
  
df0 = - 2^(1/2)/2 - 1
```

```
>> syms x;  
>> f(x)=cos(x)-x  
  
f(x) =cos(x) - x  
  
>> x0=pi/4;  
>> f(x0)  
  
ans =2^(1/2)/2 - pi/4  
  
>> df(x)=diff(f(x))  
  
df(x)=- sin(x) - 1  
  
>> df(x0)  
df(x0) =- 2^(1/2)/2 - 1
```


二、Matlab的基础操作

重点!

2.3 数组运算

(1) 定义向量

➤ 逐个输入向量元素

✓ $x = [x1, x2, x3, \dots]$ % 定义行向量

```
>> x=[1,0,2,-3,5]
x =
     1     0     2    -3     5
```

✓ $x = [x1; x2; x3; \dots]$ % 定义列向量

注：行向量各元素之间用逗号或空格分隔；列向量各元素之间用分号分隔。

➤ 规模化定义向量

✓ 通过冒号运算符构造等间隔向量，如 $x = \text{初值} : \text{步长} : \text{终值}$

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

✓ 调用linspace函数生成等间隔向量，如

$x = \text{linspace}(\text{初值}, \text{终值}, \text{向量长度})$

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

二、Matlab的基础操作

重点!

(2) 定义矩阵

- 按行方式输入矩阵元素，如定义矩阵 $A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$

➤ 矩阵与向量的互相转换

$x = A(:)$ % 矩阵转为列向量

$A = \text{reshape}(x, [m, n])$ % 将向量x转为m行n列的矩阵

➤ 访问矩阵元素

双下标访问，如 $x = A(i, j)$ % 访问矩阵A的第i行第j列的元素

单下标访问，如 $x = A(k)$ % 访问矩阵A的第k个元素

注：必须从**下标1**开始访问!!!

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

```
A =
```

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

```
>> x=A(:)
```

```
x =
```

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

```
>> A(0)
```

数组索引必须为正整数或逻辑值。

```
>> A(1)
```

```
ans =
```

```
     1
```

二、Matlab的基础操作

(3) 特殊矩阵

- 零矩阵: zeros
- 一矩阵: ones
- 单位阵: eye
- 对角阵: diag
- 随机阵: rand
- 魔方阵: magic

```
>> A=zeros(3)
A =
    0    0    0
    0    0    0
    0    0    0
>> B=ones(3,5)
B =
    1    1    1    1    1
    1    1    1    1    1
    1    1    1    1    1
>> c=eye(3,5)
c =
    1    0    0    0    0
    0    1    0    0    0
    0    0    1    0    0
>> D=diag([1 2 3])
D =
    1    0    0
    0    2    0
    0    0    3
```

```
>> E=diag(D)
E =
    1
    2
    3
>> F=rand(3)
F =
    0.8147    0.9134    0.2785
    0.9058    0.6324    0.5469
    0.1270    0.0975    0.9575
>> G=magic(3)
G =
    8    1    6
    3    5    7
    4    9    2
```

二、Matlab的基础操作

2.4 矩阵运算——算数运算

(1) 矩阵的加减

```
>> A=[1 2; 3 4];
>> B=[5 6; 7 8];
>> C=A+B
C =
     6     8
    10    12
>> D=A-B
D =
    -4    -4
    -4    -4
```

(2) 矩阵的乘法

矩阵的乘法包括乘 ($A*B$) 和点乘 ($A.*B$) 两种。其中 $A*B$ 要求A的列数等于B的行数; $A.*B$ 则表示同型矩阵A和B的对应元素相乘。

```
>> A=[1 2 3;4 5 6];
>> B=[1 1 1 1;2 2 2 2;3 3 3 3];
>> C=A*B
C =
    14    14    14    14
    32    32    32    32
>> D=[1 1 1;2 2 2];
>> E=A.*D
E =
     1     2     3
     8    10    12
```

(3) 矩阵的除法

矩阵的除法包括左除 ($A\backslash B$)、右除 (A/B) 和点除 ($A./B$) 三种。其中 $x = A\backslash b$ 是方程组 $A*x = b$ 的解, 而 $x = b/A$ 是方程组 $x*A = b$ 的解, $x = A./B$ 表示同型矩阵A和B对应元素相除。

```
>> A=[2 3 8;1 -2 -4;-5 3 1];
>> b=[-5;3;2];
>> x=A\b
x =
    1.0000
    3.0000
   -2.0000
>> B=A;
>> C=A./B
C =
     1     1     1
     1     1     1
     1     1     1
```

二、Matlab的基础操作

2.4 矩阵运算——关系运算

矩阵的关系运算：通过比较两个同型矩阵的对应元素的大小关系，或者比较一个矩阵的各元素与某一标量之间的大小关系，返回一个逻辑矩阵（1表示真，0表示假）。

关系运算的运算符有： < (小于)、<= (小于或等于)、> (大于)、>= (大于或等于)、== (等于)、~= (不等于)6种。例子如下：

```
>> A=[1 2;3 4];  
>> B=[2 3;2 2];  
>> C1=A>B  
C1 =  
    0    0  
    1    1  
>> C2=A~=B  
C2 =  
    1    0  
    1    1  
>> C3=A>=2  
C3 =  
    0    1  
    1    1
```

二、Matlab的基础操作

2.4 矩阵运算——逻辑运算

矩阵的逻辑运算包括：

- ✓ 逻辑“或”运算，运算符为“|”。 $A | B$ 表示同型矩阵A和B的或运算，若A和B的对应元素至少有一个非0，则相应的结果元素值为1，否则为0；
- ✓ 逻辑“与”运算，运算符为“&”。 $A \& B$ 表示同型矩阵A和B的与运算，若A和B的对应元素均非0，则相应的结果元素值为1，否则为0；
- ✓ 逻辑“非”运算，运算符为“~”。 $\sim A$ 表示矩阵A的非运算，若A的元素值为0，则相应的结果元素值为1，否则为0；
- ✓ 逻辑“异或”运算， $\text{xor}(A, B)$ 表示同型矩阵A和B的异或运算，若A和B的对应元素均为0或均非0，则相应的结果元素值为0，否则为1。例子如右：

```
>> A=[0 0 1 2];  
>> B=[0 -2 0 1];  
>> C1=A|B  
C1 =  
    0    1    1    1  
>> C2=A&B  
C2 =  
    0    0    0    1  
>> C3=~A  
C3 =  
    1    1    0    0  
>> C4=xor(A,B)  
C4 =  
    0    1    1    0
```

二、Matlab的基础操作

2.4 矩阵运算——其他常用运算

(1) 矩阵的转置与共轭转置

矩阵的转置包括转置 (A') 和共轭转置 (A') 两种。

```
>> A = [1 2 3; 4 5 6; 7 8 9]
A =
1 2 3
4 5 6
7 8 9
>> B = A' % 或 B = A.'
B =
1 4 7
2 5 8
3 6 9
```

(2) 矩阵的翻转

```
>> A = [1 2 3; 4 5 6; 7 8 9];
>> B1 = flipud(A)
B1 =
7 8 9
4 5 6
1 2 3
>> B2 = fliplr(A)
B2 =
3 2 1
6 5 4
9 8 7
```

(3) 调用det函数计算方阵的行列式

$d = \det(A)$ % 计算方阵A的行列式

```
>> A = [1 2; 3 4];
>> d1 = det(A)
d1 = -2
```

二、Matlab的基础操作

(4) 逆矩阵与广义伪逆矩阵

```
>> A = [1 2; 3 4];
>> Ai = inv(A)
Ai =
-2.0000 1.0000
1.5000 -0.5000

>> C = [1 2 3; 4 5 6];
>> Cpi = pinv(C)
Cpi =
-0.9444 0.4444
-0.1111 0.1111
0.7222 -0.2222

>> D = C * Cpi * C
D =
1.0000 2.0000 3.0000
4.0000 5.0000 6.0000
```

(5) 方阵的特征值与特征向量

```
>> A = [5 0 4; 3 1 6; 0 2
3];
>> d = eig(A)
d =
-1.0000
3.0000
7.0000

>> [V, D] = eig(A)
V =
-0.2857 0.8944 0.6667
-0.8571 0.0000 0.6667
0.4286 -0.4472 0.3333
D =
-1.0000 0 0
0 3.0000 0
0 0 7.0000
```

(6) 矩阵的迹和矩阵的秩

```
>> A = [1 2 3; 4 5 6; 7 8 9];
>> t = trace(A)
t =
15

>> r = rank(A)
r =
2
```


目录

➤ 一、Matlab的工作环境

➤ 二、Matlab的基础操作

➤ 三、Matlab的程序设计

3.1 MATLAB语言的流程结构

3.2 M文件的编写与调试

三、Matlab的程序设计

重点!

3.1 Matlab语言的流程结构

MATLAB作为一种程序设计语言，它提供了选择语句结构和循环语句结构，其中
选择语句结构包括：if/else 条件转移语句结构， switch开关语句结构， try...catch试探语句结构；
循环语句结构包括：for循环和while循环语句结构。
此外，MATLAB还提供了 continue、break、return和pause等流程控制函数。

(1) 选择结构

➤ if/else 条件转移语句结构

格式一： if 条件
 语句组
 end

格式二： if 条件
 语句组1
 else
 语句组2
 end

格式三： 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m+1
 end

三、Matlab的程序设计

例子：输入3个实数，判断以这3个数为边长能否构成三角形，若构成三角形，请利用海伦公式求其面积。

% 交互式输入一个包含三个元素的向量

```
A = input('请输入三角形的三条边: ');
```

```
if A(1) + A(2) > A(3) & A(1) + A(3) > A(2) & A(2) + A(3) > A(1)
```

```
    p = (A(1) + A(2) + A(3)) / 2;
```

% 用海伦公式求三角形面积

```
s = sqrt(p*(p - A(1))*(p - A(2))*(p - A(3)));
```

```
disp(['该三角形面积为: ' num2str(s)]);
```

```
else
```

```
    disp('不能构成一个三角形。')
```

```
end
```

运行：请输入三角形的三条边： [4 5 6]
9.9216

三、Matlab的程序设计

重点!

➤ switch/case开关语句结构

switch语句根据变量或表达式的取值不同，分别执行不同的语句。其格式为：

```
switch 表达式
case 值1
    语句组1
case 值2
    语句组2
.....
case 值m
    语句组m
otherwise
    语句组m+1
end
```

例子：根据变量 num 的值来决定显示的内容。

```
num=input('请输入一个数: '); % 交互式输入一个数
switch num % 根据num的不同取值显示不同的信息
    case -1
        disp('I am a teacher. ');
    case 0
        disp('I am a student. ');
    case 1
        disp('You are a teacher. ');
    otherwise
        disp('You are a student. ');
end
```

三、Matlab的程序设计

➤ try...catch试探语句结构

```
try
    语句组1
catch
    语句组2
end
```

程序首先运行try和catch之间的“语句组1”，
如果发生错误，则执行catch和end之间的“语句组2”，
然后继续执行end后的程序。反之，没有
错误时，直接继续执行end后的程序。

例子：

%程序：

```
a = rand(3,4);
b = magic(5);
try
    c=a*b %执行该语句段出现错误，转去执行catch之后的语句段
    disp(c)
catch
    disp(size(a))
    disp(size(b))
end
```

三、Matlab的程序设计

重点!

(2) 循环结构

➤ for 循环

格式: for 循环变量 = 向量
 循环体语句
 end

注: 循环中, 循环变量依次从向量中取值。

➤ while 循环

格式: while 条件
 循环体语句
 end

例子: 令 $y = \sum_{i=1}^n i^2$, 求使得 $y \leq 2000$ 的最大的正整数 n 和相应的 y 值。

%程序1:

```
y = 0;
for i = 1:100
    y = y + i^2;
    if y > 2000
        break; %跳出循环
    end
end
n = i - 1
y = y - i^2
```

运行结果得
 $n = 17$
 $y = 1785$

%程序2:

```
y = 0;
i = 0;
while y <= 2000
    i = i + 1;
    y = y + i^2;
end
n = i - 1
y = y - i^2
```

三、Matlab的程序设计

(3) continue、break、return和pause函数

- ✓ **continue**函数能用在for或while循环结构的循环体语句中，它的功能是**跳过当步循环直接执行下一次循环**，通常与if语句结合使用；
- ✓ **break**函数另能用在for或while循环结构的循环体语句中，它的功能是**跳出break函数所在层循环**，通常与if语句结合使用；
- ✓ **return**函数的用法比较灵活，通常用在某个函数体里面，根据需要可以用在函数体的任何地方，其功能是**跳出正在调用的函数**，通常与if语句结合使用；
- ✓ **pause**函数用来**实现暂停功能**

pause %暂停程序的执行，等待用户按任意键继续

pause(n) %暂停程序的执行，n秒后继续，n为非负实数

pause on %开启暂停功能，使后续pause和 pause(n)指令可以执行

pause off %关闭暂停功能，不执行后续 pause和pause(n)指令

三、Matlab的程序设计

重点!

3.2 M文件的编写与调试

(1) M文件的分类

M文件可分为**脚本文件**和**函数文件**。

脚本文件是包含多条MATLAB命令的文件。如右图；

函数文件可以包含输入变量，并把结果传送给输出变量。

函数文件的格式：

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

注释说明部分 (%号引导的行)

函数体

```
end
```

其中out1, out2, ...为输出参数列表, in1, in2, ...为输入参数列表, funname为函数名。

脚本文件的例子：

```
编辑器 - F:\test.m
test.m  x  +
1 - A = input('请输入三角形的三条边: '); % 交互式输入一个包含三个元素的向量
2 - if (A(1) + A(2) > A(3)) & (A(1) + A(3) > A(2)) & (A(2) + A(3) > A(1))
3 -     p = (A(1) + A(2) + A(3)) / 2; % 用海伦公式求三角形面积
4 -     s = sqrt(p*(p - A(1))*(p - A(2))*(p - A(3)));
5 -     disp(['该三角形面积为: ' num2str(s)]);
6 - else
7 -     disp('不能构成一个三角形。');
8 - end
```

函数文件的例子：

```
sumlm.m  x  +
1 - function [n, y] = sumlm(m)
2 - % 求n和y, 使得1^2+2^2+...+n^2<=m
3 - y = 0;
4 - i = 0;
5 - while y <= m
6 -     i = i + 1;
7 -     y = y + i^2;
8 - end
9 - n = i - 1;
10 - y = y - i^2;
```


三、Matlab的程序设计

(2) M文件的编写

- **步骤1**：创建一个新m文件

单击快捷按钮

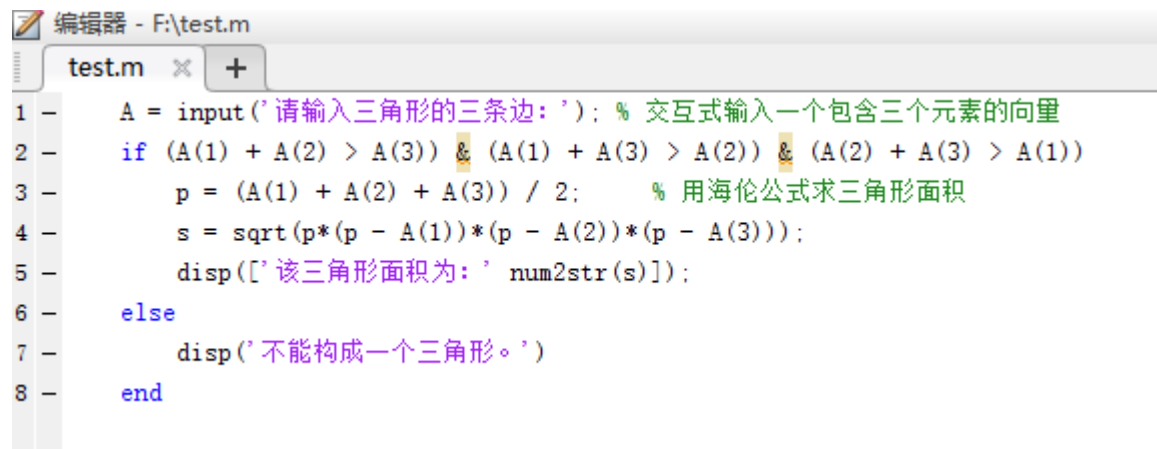


,打开如下脚本编辑窗口



- **步骤2**：编写代码

编写三角形判断的代码至**脚本文件**中



- **步骤3**：点击保存按钮



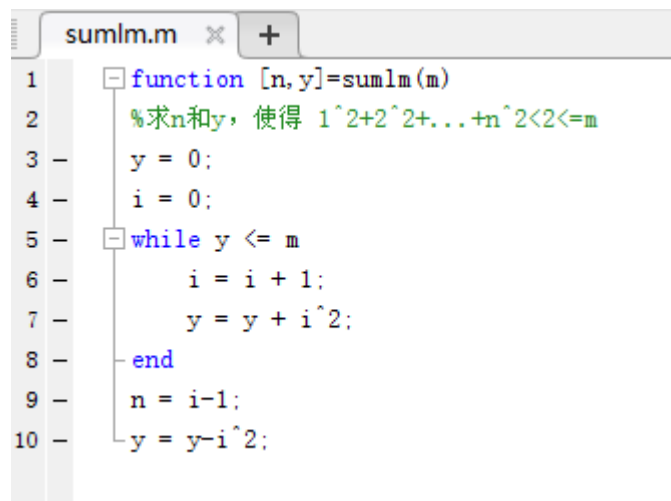
, 在弹出的对话框中选择保存路径, 以及命名文件名为test

三、Matlab的程序设计

- **步骤4**：M文件的使用。回到Matlab的**命令窗口**，输入**函数名称**test，进行调用，输出数据，得到结果如下：

```
>> test
请输入三角形的三条边: [4 5 6]
该三角形面积为: 9.9216
```

- **步骤5**：若是编写的文件为**函数文件**，那么保存的文件名即为函数名，如下sumlm



```
sumlm.m  x  +
1  function [n,y]=sumlm(m)
2      %求n和y, 使得 1^2+2^2+...+n^2<=m
3      y = 0;
4      i = 0;
5      while y <= m
6          i = i + 1;
7          y = y + i^2;
8      end
9      n = i-1;
10     y = y-i^2;
```

那回到Matlab的**命令窗口**，输入**函数名称(参数)**，才能调用成功，如sumlm(2000)

```
>> sumlm
输入参数的数目不足。

出错 sumlm (line 5)
while y <= m

>> sumlm(2000)

ans =

17
```

三、Matlab的程序设计

(3) M文件的调试 (debug)

➤ 常用的调试方法:

- (1) **设置或清除断点**: 使用快捷键F12
- (2) **执行**: 使用快捷键F5
- (3) **单步执行**: 使用快捷键F10
- (4) **step in**: 当遇见函数时, 进入函数内部, 使用快捷键F11
- (5) **step out**: 执行流程跳出函数, 使用快捷键Shift + F11
- (6) 执行到光标所在位置
- (7) 观察变量或表达式的值: 将鼠标放在要观察的变量上停留片刻, 就会显示出变量的值, 当矩阵太大时, 只显示矩阵的维度
- (8) **退出调试模式**: Shift + F5

➤ 常用的显示出数据的方法:

- ❑ disp函数 (注意: disp的参数是一个数组, 以及联合num2str或Int2str来打印出信息)
- ❑ fprintf函数 (格式为: fprintf(format,data))

The background features several circles of varying sizes, some solid blue and others light gray with a 3D shadow effect. A large light gray circle with a shadow is positioned in the upper left, containing the word 'THANKS' in red. Other circles are scattered around the slide, including a medium blue circle in the lower left and a large light gray circle in the lower center.

THANKS

同学们，
请开始实验吧！