

# 第1章 初识MATLAB

---

本章内容：

1.1 MATLAB入门

1.2 数据运算

1.3 矩阵

1.4 关系运算与逻辑运算

1.5 例题讲解

学时： 4

# 1.1 MATLAB入门

---

- 1.1.1 操作桌面
- 1.1.2 帮助系统
- 1.1.3 数据类型

## 1.2.1 操作桌面

- **命令行窗口 (Command Window) :**  
在命令行中输入命令 (由提示符 (>>) 表示) 。
- **工作区 (Workspace)**  
浏览用户创建或从文件导入的数据。
- **当前文件夹 (Current Directory) :** 访问文件
- **命令历史记录 (Command History) :**  
记录用户输入过的命令

工具栏

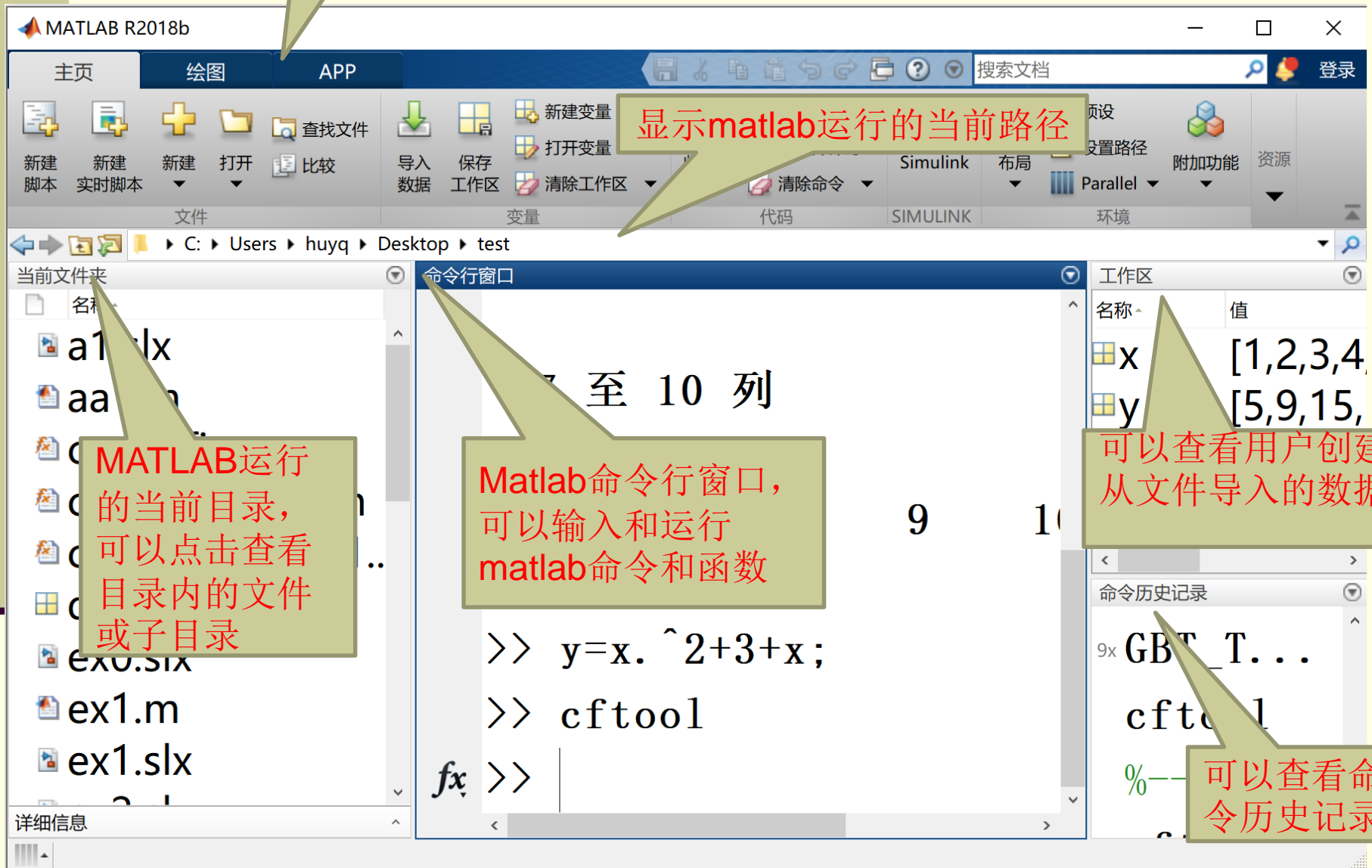
显示matlab运行的当前路径

MATLAB运行的当前目录，可以点击查看目录内的文件或子目录

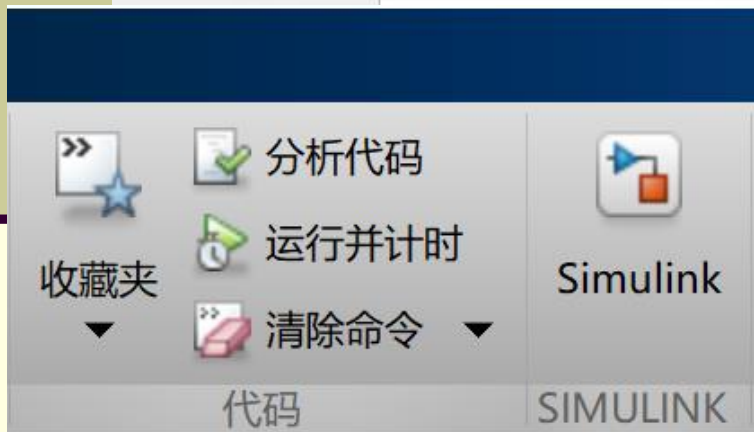
Matlab命令行窗口，可以输入和运行matlab命令和函数

可以查看用户创建或从文件导入的数据。

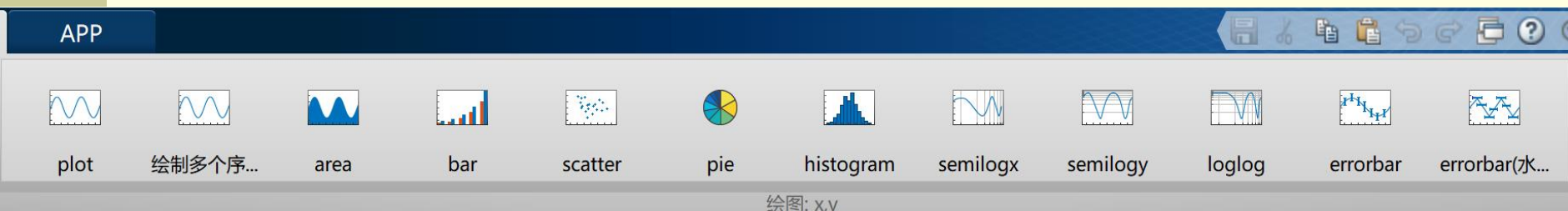
可以查看命令历史记录



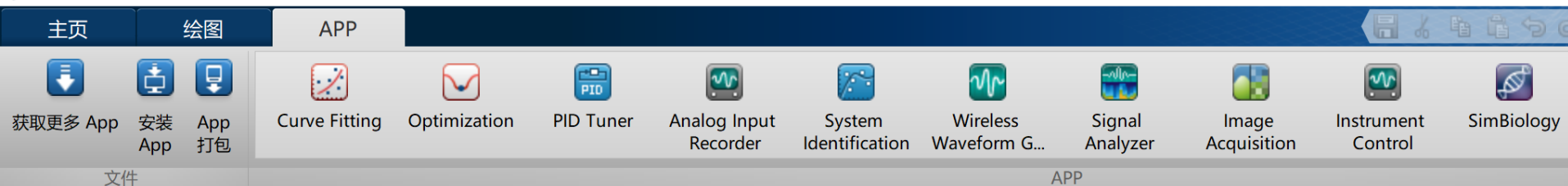
# 1.2.1 操作桌面



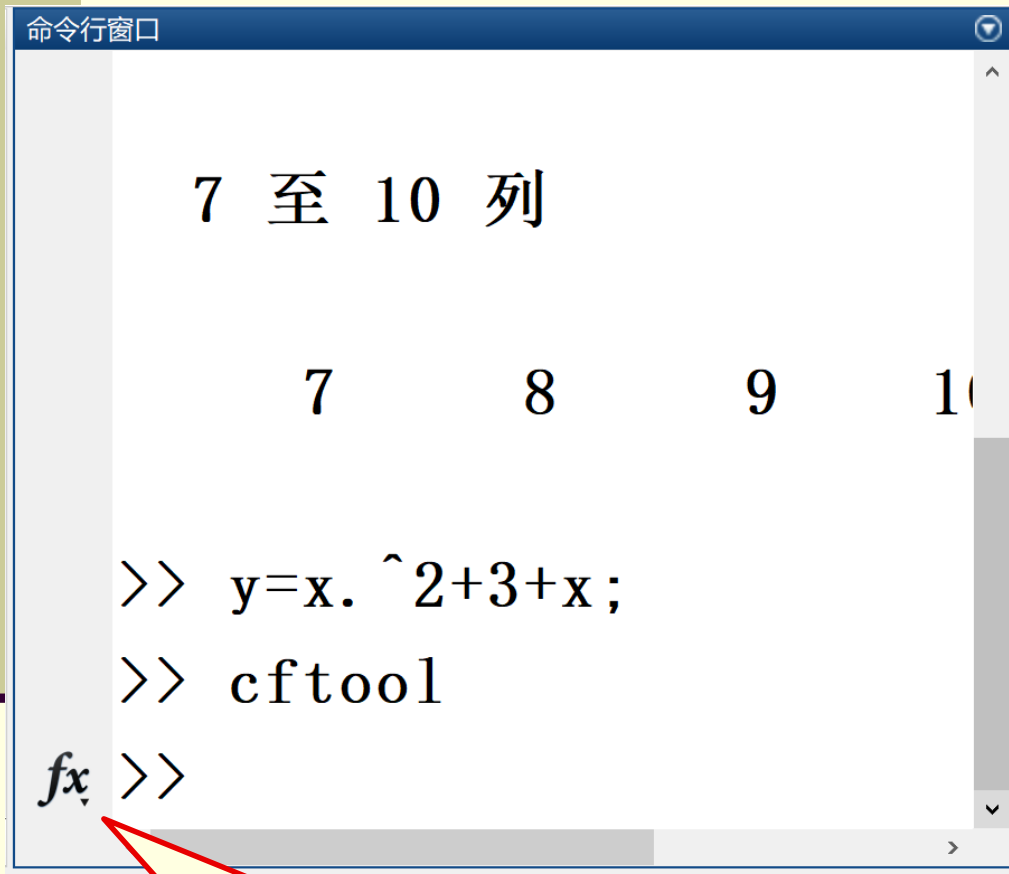
# 1.2.1 操作桌面



MATLAB R2018b



## 1.2.1 操作桌面——命令行窗口



- 命令窗口用于输入 MATLAB 命令、函数、矩阵、表达式等信息，并显示除图形以外的所有计算结果，是 MATLAB 的主要交互窗口。
- 当命令窗口出现提示符 `>>` 时，表示 MATLAB 已准备好，可以输入命令、变量或函数，回车后就可执行。

## 1.2.1 操作桌面——命令行窗口

可发出创建变量和调用函数的命令。

```
>> a=1
```

```
>> b=2
```

```
>> c = a + b
```

```
>> d = cos(a)
```

```
>> sin(a) %如果未指定输出变量，将使用ans来存储计算结果。
```

说明1：百分号“%”后面的语句为注释语句

说明2：指令不能包含中文符号



## 1.2.1 操作桌面——命令行窗口

比较如下两行的区别：

```
>> e=sin(2*a)    %从键盘输入，回车
```

```
>> E=sin(2*a);   %从键盘输入，回车
```

说明1：如果语句以“;”结束，MATLAB 会执行计算，但不在命令行窗口中显示输出。

说明2：MATLAB对字母的大小写是敏感的，变量e与变量E表示两个不同的变量。

说明3：MATLAB所有的命令和函数都必须用小写。

说明4：一个命令行也可以输入若干条命令，以逗号或分号分隔。例如

```
>> e=sin(2*a); E=sin(2*a);
```

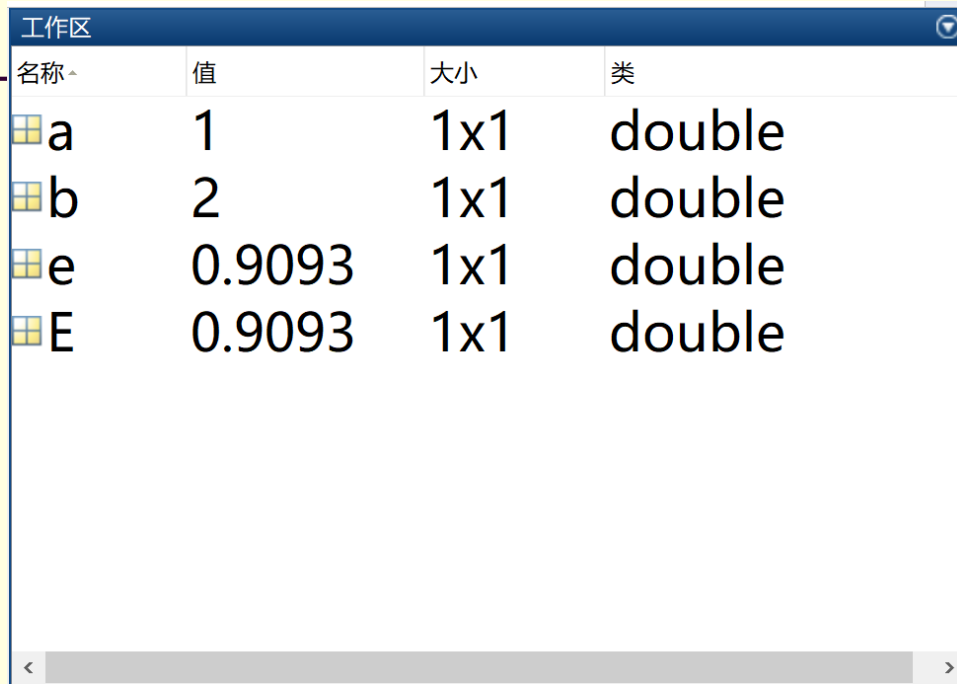
## 1.2.1 操作桌面——命令行窗口

(1) 在空白命令行中按向上 **【↑】** 或向下箭头键 **【↓】**，然后按 **【Enter】** 键可以重新调用以前的命令。

(2) 在键入命令的前几个字符之后按箭头键。可以重新调用以前的命令。

```
>> clc          % 清除命令空间
```

## 1.2.1 操作桌面——工作区



The image shows a screenshot of the MATLAB 'Workbench' (工作区) window. It contains a table with four columns: 'Name' (名称), 'Value' (值), 'Size' (大小), and 'Class' (类). The table lists four variables: 'a' with value 1, 'b' with value 2, 'e' with value 0.9093, and 'E' with value 0.9093. All variables are of size 1x1 and of type double. Each variable name in the first column is preceded by a small icon representing its class (double).

名称	值	大小	类
a	1	1x1	double
b	2	1x1	double
e	0.9093	1x1	double
E	0.9093	1x1	double

工作区是用于存储各种变量和结果的内存空间。

包含在 **MATLAB** 中创建或从数据文件或其他程序导入的变量。

`>> whos`

%查看工作区的内容。

`>> save myfile`

%使用 **save** 命令保存数据以供将来使用

`>> clear`

%清除工作空间

`>> load myfile`

%使用 **load** 命令将 **.mat** 文件中的数据还

%原到工作区

说明：关闭**MATLAB**时工作空间自动清除。

## 1.2.1 操作桌面——工作区

---

例：工作区的清除

```
>> 1+1                % 计算1+1
```

```
>> clear              % 清除命令空间
```

说明：百分号“%”后面的语句为注释语句

## 1.2.1 操作桌面——当前文件夹

---

当前目录是指**MATLAB**运行文件时的工作目录，只有在当前目录或搜索路径下的文件及函数可以被直接运行或调用，如果没有特殊指明，数据文件也将存储在当前目录下。

通常我们需要建立自己的工作目录，以便于文件和数据的管理。运行文件前要将该文件所在的目录设置为当前目录。

## 1.2.1 操作桌面——当前文件夹

### 例：工作目标的设置

1. 在桌面上建立一个文件夹 **ex**
2. 文件夹 **ex**下面新建一个名为**ex1**的**txt**文档，写入几行数字
3. 更改**MATLAB**的工作目录到**ex**
4. 观察当前目录浏览器的变化
5. 在当前目录浏览器中双击打开名为**ex1**的**txt**文档

## 1.2.1 操作桌面——命令历史窗口

---

命令历史窗口记录已经运行过的命令、函数、表达式等信息，可以进行命令历史的查找、检查等工作。

可以在该窗口中对命令历史进行复制、删除等操作。

窗口中除了保留了输入的命令外，还记录了每次打开系统的时间。

可以清除掉这些记录。

## 1.2.1 操作桌面——命令历史窗口

### 例：命令历史窗口的使用

1. 在命令窗口输入几行指令
2. 在命令历史窗口查看刚才输入的指令
3. 双击其中一行指令，再次运行该指令
4. 选中其中的几行指令，建立一个名为**ex2**的**M**文件
5. 将**M**文件保存到**ex**文件夹
6. 在当前目录浏览器名为**ex2**的**M**文件，双击打开运行



## 1.2.2 帮助系统

- MATLAB强大的帮助系统是学习MATLAB知识的最权威资料。
- 帮助系统本身就是一个优秀的MATLAB教程，掌握帮助系统的使用方法至关重要。

■ 打开方式：

菜单栏->Help->Product Help

>> help                    % 显示帮助总览

>> help elfun            %显示基本函数信息

>> help sin              %显示具体函数的详细信息



### 帮助中心

≡ 目录

« 文档主页

类别

MATLAB 快速入门

语言基础知识

数据导入和分析

数学

图形

编程

App 构建

软件开发工具

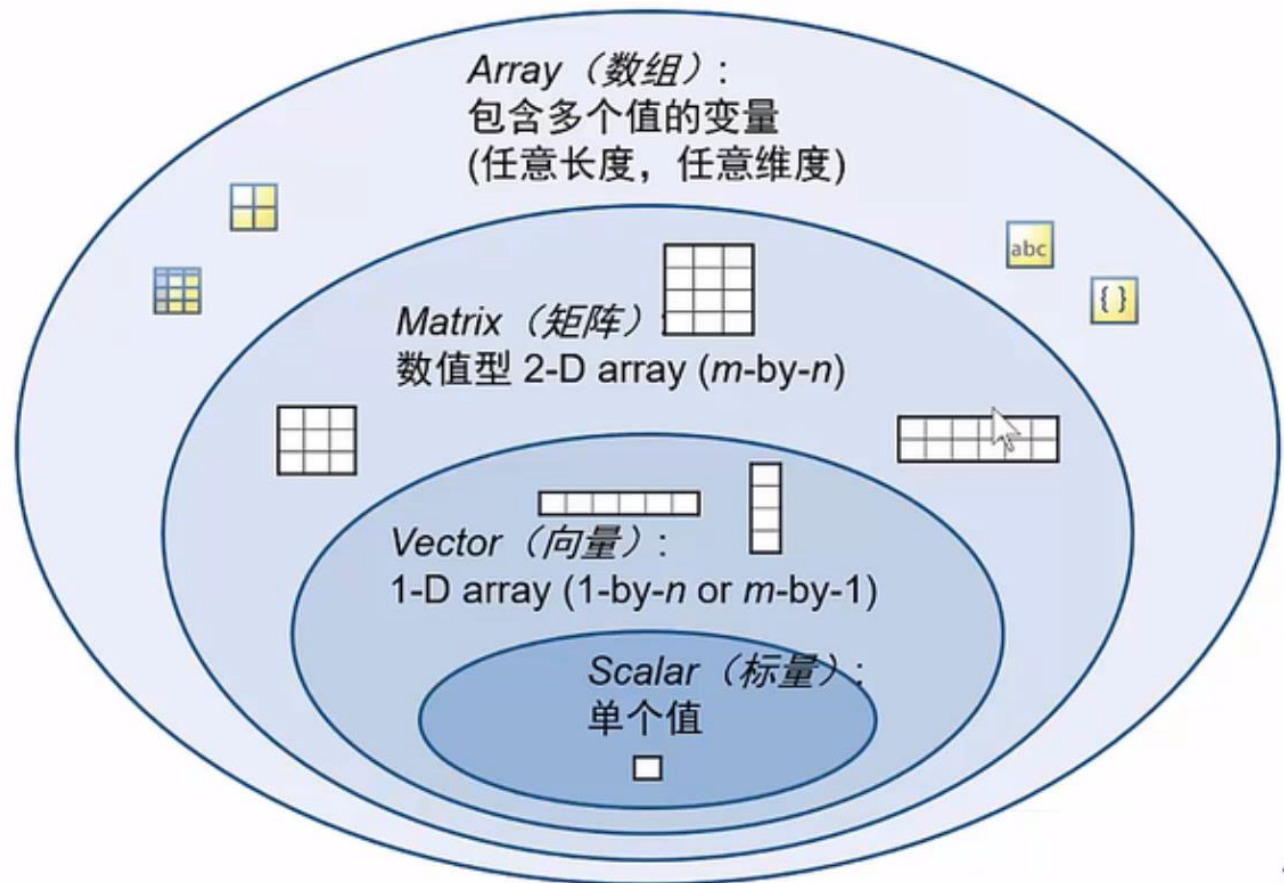
外部语言接口

环境和设置

## 1.2.3 数据类型



### MATLAB 中的数据结构



## 1.2.3 数据类型

类型名称	函数	举例	说明
字符型	<b>char</b>	'A'、'happy'	字符型数组每个字符占2个字节，即16位
整型（有符号）	<b>int8、int16、 int32</b>	<b>int8(156)</b>	8位、16位、32位的整数数组，常用于表示信号
整型（无符号）	<b>uint8、 uint16、 uint32</b>	<b>uint8(2)</b>	
单精度	<b>single</b>	<b>single(32.3)</b>	单精度数值数组所需的存储空间较小，占4个字节，可以表示小数，但精度差，数值范围小，能用于数学运算
双精度	<b>double</b>	<b>32 double(44.5)</b>	双精度数值数组，占8个字节，精度高，数值范围大，能用于数学运算，是默认的MATLAB变量类型
稀疏矩阵	<b>sparse</b>	<b>sparse(6)</b>	稀疏双精度矩阵，稀疏矩阵只存储少数的非零元素，较常规矩阵的存储节约了大量的存储空间
单元数组	<b>cell</b>	<b>{10,'h',3.4}</b>	单元数组，单元数组元素的尺寸、性质可以不同
结构数组	<b>struct</b>	<b>g=struct('name','LiXin', 'number','441')</b>	结构数组，结构数组包括域名，域中可以包括其他数组，与单元数组类似

## 1.2.3 数据类型

### a. 矩阵和数组

矩阵是指含有M行、N列数据的矩形结构。

$$\begin{bmatrix} 1 & 4 & 2 \\ 2 & 0 & 0 \end{bmatrix}$$

数组运算是元素对元素的运算，也就是说无论什么运算，对数组中的元素都是平等进行的；

矩阵运算是强调整体的运算，采用线性代数的运算方法。

## 1.2.3 数据类型

---

MATLAB 是一种基于矩阵的计算环境。在 MATLAB 中无论是数值、字符还是逻辑 **true** 或 **false** 数据，都会以矩阵或多维数组的形式进行存储。

```
>> A = 100;
```

```
>> B='Hello World';
```

```
>> whos A B
```

# 1.3 数据运算

---

- 1.3.1 变量
- 1.3.2 常用数学函数
- 1.3.3 数据操作

## 1.3.1 变量

### a. 变量的命名

在MATLAB中，变量名可以由字母、数字或下划线组成的字符序列，最多可包含63个字符，但第一个字符必须是字母。

例如：myfile13、ab\_1cd、EXAMPE等均为合法变量名，而3dat、\_mydat、123.4等都不是合法变量名

注：不能使用MATLAB的关键字作为变量名，如if, while, for等

### b. 赋值语句

赋值语句的格式：变量名=表达式

表达式    %表达式的值赋给预定义变量ans

## c. 预定义变量

预定义变量名	含义	预定义变量名	含义
ans	计算结果的缺省赋值变量	i, j	虚数单位
eps	容差变量，定义为1.0到最近浮点数的距离在pc机上，等于 $2^{-52}$	inf, Inf	正无穷大，定义为 $(1/0)$
pi	圆周率 $\pi$ 的近似值	NaN, nan	非数。在IEEE运算规则中，它产生于 $0/0$ 、 $0 \times \infty$ 等的结果
realmax	最大正实数	nargin	函数输入参数个数
realmin	最小正实数	nargout	函数输出参数个数
lasterr	存放最新的错误信息	lastwarn	存放最新的警告信息



#### d. 变量的清除

例: `n=3;`  
`clear n`

#### e. 变量名的修改

例: `n=3;`  
`m=n;`  
`clear n`

#### f. 变量的查看 `whos` 或者 `whos` 变量名

#### g. 变量的保存和读取

**save** 函数将工作空间的数据读存入磁盘。

**load** 函数将磁盘上的数据读入到工作空间。

## 函数格式:

- (1) `save` %将所有变量存入文件matlab.mat
- (2) `save dfile.mat` %将所有变量存入文件dfile.mat
- (3) `save dfile.mat a b` %将指定变量a,b存入文件dfile.mat

- (1) `load` %载入文件matlab.mat中的所有变量
- (2) `load dfile .mat` %载入文件dfile.mat中的所有变量
- (3) `load dfile .mat a b` %从文件dfile.mat中提取指定变量a,b

## 例：数据的保存与读入

- `>> a=2;`
- `>> b=4;`
- `>> c=6;`
- `>> save mydata1.mat`
- `>> save mydata2.mat a b`
- `>> clear`
- `>> load mydata2.mat a`

## 1.3.2 运算符和表达式

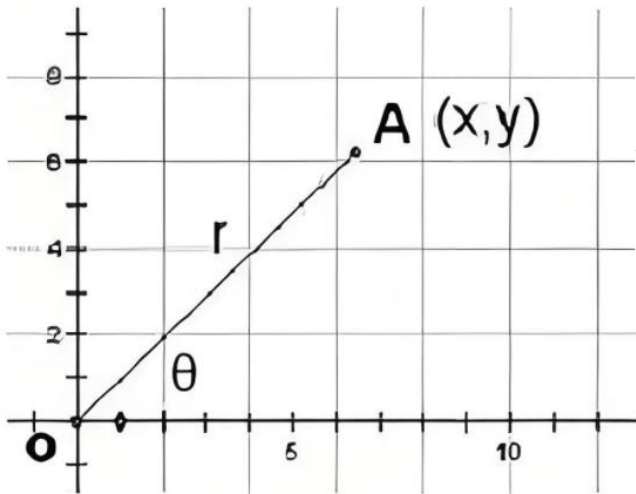
### ■ 运算符和表达式

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

- Matlab用 “\” 和 “/” 分别表示 “左除” 和 “右除” 。比较 $3/4$   $3 \setminus 4$ 。
- 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。

## 1.3.2 常用数学函数

任意角三角函数



$$\sin \theta = \frac{y}{r}$$

$$\tan \theta = \frac{y}{x}$$

$$\sec \theta = \frac{r}{x}$$

$$\cos \theta = \frac{x}{r}$$

$$\cot \theta = \frac{x}{y}$$

$$\csc \theta = \frac{r}{y}$$

函数类别	函数	功能	函数类别	函数	功能
三角函数	sin/sind	正弦	反三角函数	asin/asind	反正弦
	cos/cosd	余弦		acos/acosd	反余弦
	tan/tand	正切		atan/atand	反正切
	cot/cotd	余切		acot/acotd	反余切
	sec/secd	正割（余弦倒数）		asec/asecd	反正割
	csc/cscd	余割（正弦倒数）		acsc/acscd	反余割

## 1.3.2 常用数学函数

函数类别	函数	功能	函数类别	函数	功能
对数函数	<b>log2</b>	以2为底的对数	幂函数	<b>pow2(x)</b>	$2^x$
	<b>log10</b>	自然对数（以10为底）		<b>sqrt</b>	开平方
	<b>log</b>	以e为底的对数		<b>power(x,y)</b>	$x^y$
指数函数	<b>exp</b>	以e为底的指数	其他	<b>gcd(a,b)</b>	最大公约数
取整函数	<b>fix</b>	朝零方向取整		<b>lcm(a,b)</b>	最小公倍数
	<b>floor</b>	朝负无穷方向取整	与复数有关的函数	<b>real</b>	复数的实部
	<b>ceil</b>	朝正无穷方向取整		<b>imag</b>	复数的虚部
	<b>round</b>	四舍五入到最近的整数		<b>conj</b>	复数共轭运算
	<b>abs</b>	绝对值		<b>abs</b>	绝对值

例1：比较exp(1); log(2.7183)

例2：比较 power(3,2); power(2,3); pow2(3)

例3： x=1+2i, y=conj(x), z=abs(x)

## 1.3.2 常用数学函数

例1: 计算 
$$\frac{\left[\sqrt{2\pi} + \cos(45^\circ)\right]^3}{1 + \sqrt{5}}$$

```
>> (sqrt(2*pi)+cosd(45))^3/(1+sqrt(5))
```

```
ans =
```

```
10.2568
```

## 1.3.2 常用数学函数

■ 计算

$$\frac{5 + \cos 47^\circ}{1 + \sqrt{7} - \sin 0.7}$$

■ ans =

■ 1.8930

■ `(5+cosd(47))/(1+sqrt(7)-sin(0.7))`



## 1.3.2 复数及其运算

复数的表达:  $\mathbf{z=a+bi}$ ,  $\mathbf{z=a+b*i}$  其中a、b为实数。

提取实部、虚部:  $\mathbf{real(z)}$   $\mathbf{imag(z)}$

复数的模和幅角:  $\mathbf{abs(z)}$   $\mathbf{angle(z)}$  %返回区间  $[-\pi, \pi]$  中的相位角

【例】复数  $z_1=3+4i$ ,  $z_2=1+2i$ ,  $z_3=2e^{\frac{\pi}{6}i}$   $z = \frac{z_1 z_2}{z_3} = ?$

- $\mathbf{z1=3+4i;}$  % 4和i之间不能有空格
- $\mathbf{z2=1+2*i;}$
- $\mathbf{z3=2*exp((pi/6)*i);}$
- $\mathbf{z=z1*z2/z3}$

$\mathbf{z =}$

$\mathbf{0.3349 + 5.5801i}$

## 1.4 矩阵

---

- 1.4.1 矩阵的建立
- 1.4.2 矩阵的基本计算
- 1.4.3 矩阵的操作

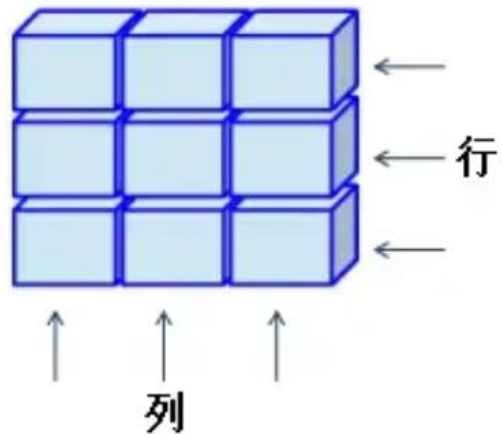
## 1.4.1 数组的概念

向量vector    矩阵matrix    数组array

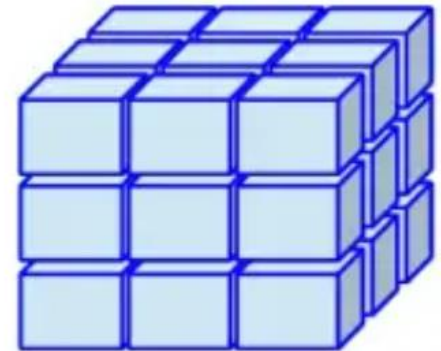
向量



矩阵



数组



## 1.4.1 矩阵的建立

在MATLAB中，不需要对矩阵的维数和类型进行说明，MATLAB会根据用户所输入的内容进行配置。

创建矩阵有以下三种方法：

- a. 使用[ ]直接输入
- b. 利用冒号:创建行向量
- c. 利用函数法创建矩阵


## 1.4.1 矩阵的建立

### 方法一. 使用[]直接输入

规则1: 矩阵维数不必预先定义, 矩阵元素必须在 “[ ]” 内

规则2: 同一行的元素之间用空格或“,” 隔开

```
>> x = [19 13 7 11 2 17 5 3]
```



19	13	7	11	2	17	5	3
----	----	---	----	---	----	---	---

只有一行的  
矩阵称为  
行向量

规则3: 行与行之间用“;” 隔开

```
>> x = [19;13;7;11;2;17;5;3]
```



19
13
7
11
2
17
5
3

只有一列的  
矩阵称为  
列向量

规则4: 矩阵元素可以是数值、变量、表达式或函数。

## 1.4.1 矩阵的建立

例：

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

A=

1	2	3
4	5	6
7	8	9

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

要串联的数组的维度不一致。

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。

```
>> B=[2.5,sqrt(2);sind(30),exp(2)]
```

```
>> 创建复数矩阵
```

```
C = [3+4i, 4+3j; -i, 10j]
```

## 1.4.1 矩阵的建立

练习：使用[]直接输入法创建矩阵

数组 (array)	大小(size)
$a = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$	$3 \times 2$
$b = [1 \ 2 \ 3 \ 4]$	$1 \times 4$
$c = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$	$3 \times 1$

## 1.3.2 常用数学函数

练习：计算角度 $x=30,45,60$ 的正弦、余弦、正切和余切

```
>> x=[30,45,60]
```

```
>> sind(x)
```

```
>> cosd(x)
```

```
>> tand(x)
```

```
>> cotd(x)
```



## 1.4.1 矩阵的建立

方法二. 利用 “:” 创建行向量

其格式为:

向量名=初值:增量:终值

说明: 向量是从初值开始, 以增量为步长, 直到不超过终值的所有元素所构成的序列。步长可缺省, 默认为 “1”。

例: 建立一个10以内的奇数矩阵

```
>>A=1:2:10
```

A=

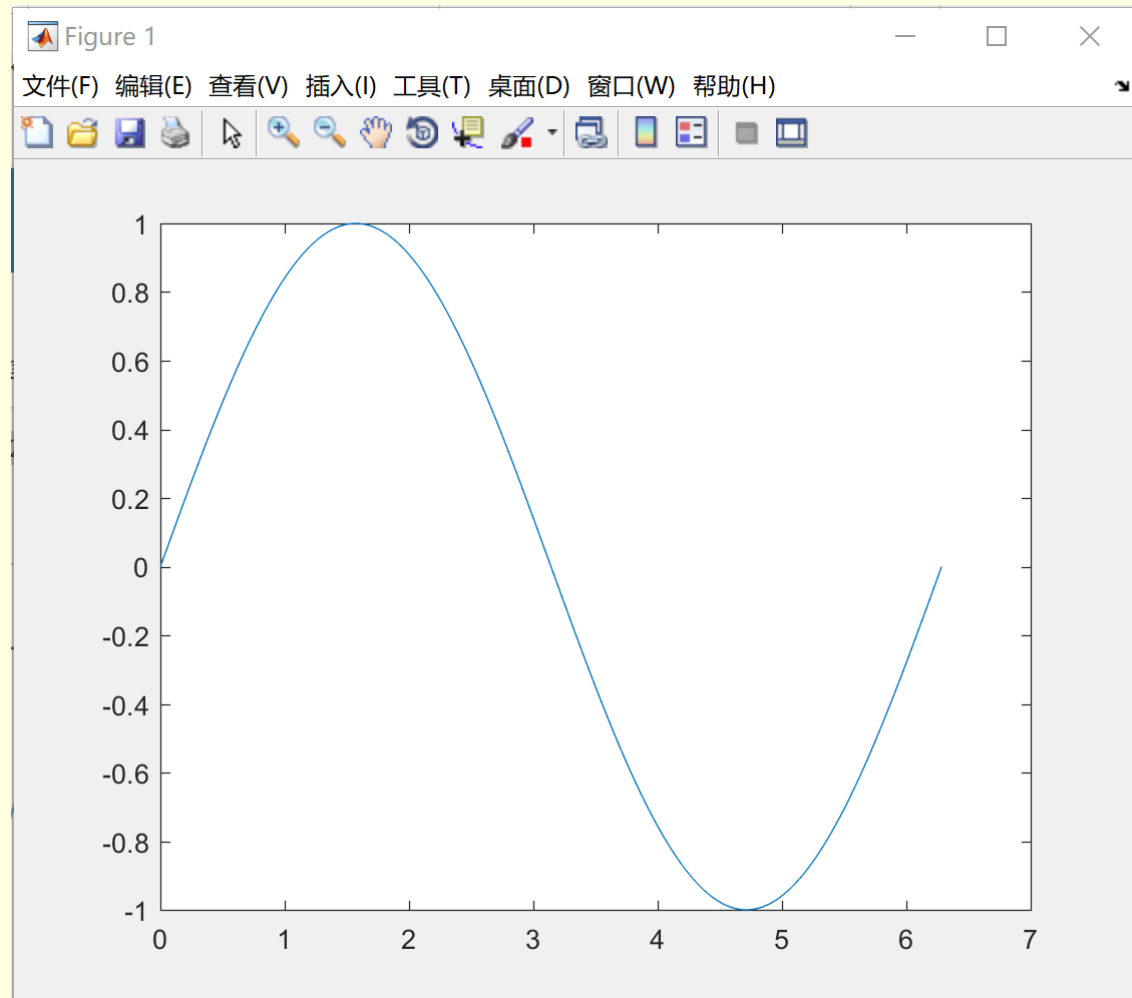
1 3 5 7 9

## 1.4.1 矩阵的建立

例：使用冒号运算符创建从 0 至  $2\pi$  的  $x$  值向量，计算这些值的正弦，并绘制结果。

```
x = 0:pi/100:2*pi;  
y = sin(x);  
plot(x,y)
```

%plot(x,y) 生成  
y 对 x 的图形。



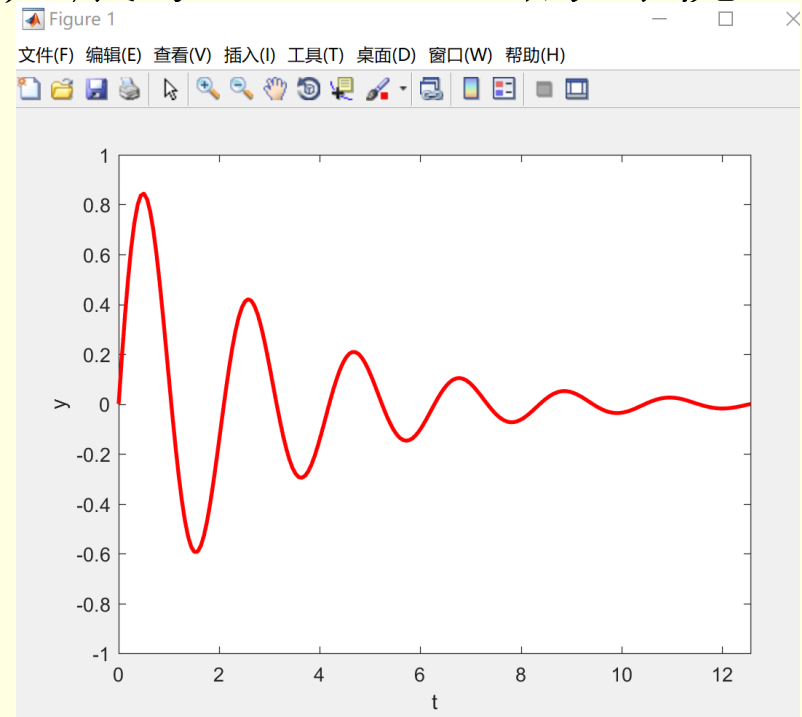
## 1.4.1 矩阵的建立

例：画出衰减振荡曲线

$$y = e^{-\frac{t}{3}} \sin 3t \quad (0 \leq t \leq 4\pi)$$

本例展示数组运算的优点；展示MATLAB的可视化能力。

- **`t=0:pi/50:4*pi;`**
- **`y=exp(-t/3).*sin(3*t);`**
- **`plot(t,y,'-r','LineWidth',2)`**
- **`axis([0,4*pi,-1,1])`**
- **`xlabel('t'),ylabel('y')`**



## 1.4.1 矩阵的建立

方法三. 定数线性采样法

格式: 向量名=`linspace(a,b, n)`

说明: **a**为第一个元素;

**b**为最后一个元素;

**n**为总元素个数, 默认值为100。

该指令等效于 向量名=`a:(b-a)/(n-1):b`

```
>> x1=linspace(2,4,6)
```

```
x1 =
```

```
2.0000    2.4000    2.8000    3.2000    3.6000    4.0000
```

## 1.4.1 矩阵的建立

### 方法四. 函数法创建矩阵

函数	说明	函数	说明
[ ]	空矩阵	<b>zeros(m,n)</b>	全部元素都为0的矩阵
<b>eye(m,n)</b>	单位矩阵	<b>magic(n)</b>	魔方矩阵
<b>ones(n)</b>	全部元素都为1的 常数矩阵	<b>randperm(n)</b>	随机排列整数矩阵
<b>rand(m,n)</b>	元素服从0和1之间 均匀分布的随机矩 阵	<b>randn(m,n)</b>	元素服从零均值单位方 差正态分布的随机矩阵

## 1.4.1 矩阵的建立

例：利用函数法创建随机矩阵

```
X = rand %返回一个在区间 (0,1) 内均匀分布的随机数。  
X = rand(n) %生成一个n×n的介于 0 和 1 之间均匀分布的随机矩阵。  
X = rand(sz1,...,szN) %返回由随机数组成的 sz1×...×szN 数组。  
X = rand(sz)  
X = rand(___,typename)  
X = rand(___, 'like', p)
```

示例：

```
>> a = rand + 1i*rand %生成一个实部和虚部位于区间 (0,1) 内的随机复数。  
>> r = rand(5) %生成一个5×5的介于 0 和 1 之间均匀分布的随机矩阵。  
>> r = -5 + (5+5)*rand(10,1) %生成区间 (-5,5) 内的 10个随机数构成的列向量。
```

## 1.4.2 矩阵的基本计算

### 矩阵运算符

表达式使用大家熟悉的算术运算符和优先法则。

+	加法
-	减法
*	乘法
/	除法
\	左除
^	幂
'	复共轭转置
( )	指定计算顺序

## 1.4.2 矩阵的基本计算

### 数组运算符

如果矩阵不用于线性代数运算，则成为二维数值数组的乘法数组运算表示法中包含点，也就是小数点。

运算符列表包括

+	加法
-	减法
.*	逐元素乘法
./	逐元素除法
.\	逐元素左除
.^	逐元素幂
.'	非共轭数组转置



## 1.4.2 矩阵的基本计算

### a. 矩阵与标量的运算

矩阵与标量运算是矩阵的每个元素对该标量的运算。

例如：

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

```
>>b=a+1
```

```
>>b=a-1
```

```
>>b=a*2
```

```
>>b=a/2
```

```
>>b=a^2
```

## 1.4.2 矩阵的基本计算

### b. 矩阵与矩阵的运算

#### (1) 加减运算

两个矩阵的维数必须完全相同。如果两个矩阵的维数不相等，则MATLAB将给出错误信息，提示两个矩阵的维数不相等。

```
>> a=ones(3)
```

```
>> b=zeros(3)
```

```
>> c=rand(4)
```

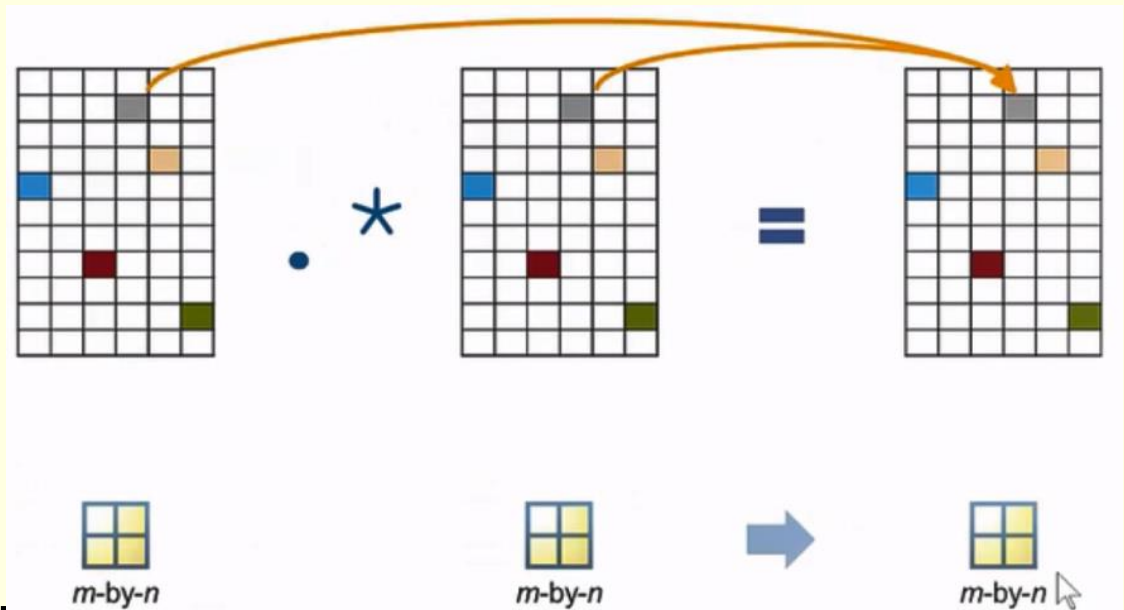
```
>> d=a+b
```

```
>> e=a+c
```

## 1.4.2 矩阵的基本计算

### (2) 点运算 “.” “.” “.” “.”

点运算是按照数组运算规则计算，矩阵的对应元素直接运算。要求矩阵大小必须相同。



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

```
>> B=[10,20,30;40,50,60];
```

```
>> C=A.*B; D=A./B; E=B.\A
```

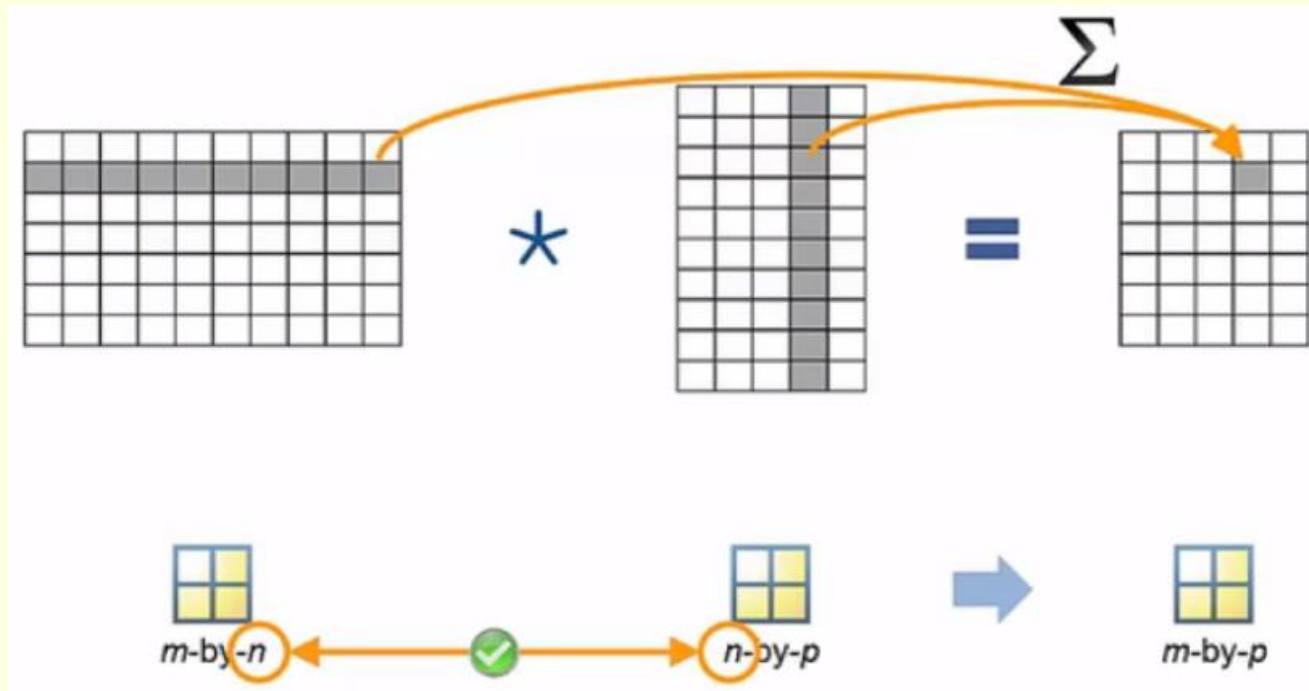
```
>> F=A.^2;
```

## 1.4.2 矩阵的基本计算

### (3) 乘法运算 $A*B$

执行标准矩阵乘法，**计算行与列之间的内积。**

**要求**A的列数=B的行数



## 1.4.2 矩阵的基本计算

---

(4) 除法运算 （包括右除 “/” 和左除 “\” ）

左除：  $A \setminus B = \text{inv}(A) * B$ ，A为方矩阵

右除：  $A / B = A * \text{inv}(B)$ ，B为方矩阵

## 1.4.2 矩阵的基本计算

### c. 矩阵的行列式、秩、逆矩阵

#### (1) 行列式

命令: `det(A)`

`det`是英文单词determinant的缩写

#### (2) 秩

`rank(A)`

#### (3) 命令: `inv(A)`

或者:  $A^{-1}$

`inv`是英语单词inverse(逆向)的缩写

## 1.4.2 矩阵的基本计算

请练习：

$a = [1 \ 2 \ 3; 4 \ 5 \ 6; 7 \ 8 \ 10]$

$a + 10$

$a'$

$p = a * \text{inv}(a)$

$p = a .* a$

$p = a * a$

$a.^3$

## 1.4.2 矩阵的基本计算

■ 求解线性方程组

$$\begin{bmatrix} 2 & 3 & -1 \\ 8 & 2 & 3 \\ 45 & 3 & 9 \end{bmatrix} \begin{Bmatrix} x \\ y \\ z \end{Bmatrix} = \begin{Bmatrix} 2 \\ 4 \\ 23 \end{Bmatrix} \quad \begin{cases} 2x + 3y - z = 2 \\ 8x + 2y + 3z = 4 \\ 45x + 3y + 9z = 23 \end{cases}$$

■  $x =$

■ 0.5531

■ 0.2051

■ -0.2784

■ `a=[2,3,-1;8,2,3;45,3,9];`

■ `b=[2;4;23];`

■ `x=inv(a)*b`



## 1.4.2 矩阵的基本计算

### d. 矩阵的特征值与特征向量

如果  $Ax = \lambda x$  ,那么  $\lambda$  为特征值,  $x$  为特征向量。

命令:  $[X,D]=\text{eig}(A)$

其中D为全部特征值构成的对角矩阵,  
X的各列为与特征值对应的特征向量。

例:  $A=[1,1,0.5;1,1,0.25;0.5,0.25,2];$   
 $[X,D]=\text{eig}(A)$

## 1.4.3 矩阵的操作

a. 元素引用:  $A(i,j)$  %A矩阵的第i行第j列元素

```
>> A=[1,2,3,4,5; 6,7,8,9,10;11,12,13,14,15;16,17,18,19,20]
```

```
>> A(1,2) %通过下标引用
```

```
>> A(5) %通过索引引用，矩阵元素按列存储。
```

b. 提取子块:  $A(i:i+m,j:j+n)$

% A矩阵的第i~ i+m行内且在第j~j+n列的全部元素

```
>> B=A(1:3,:)
```

```
>> C=A(:,1:2)
```

```
>> D=A(2:3,1:2:5)
```

c. 矩阵合并

```
>> E=[A;B]
```

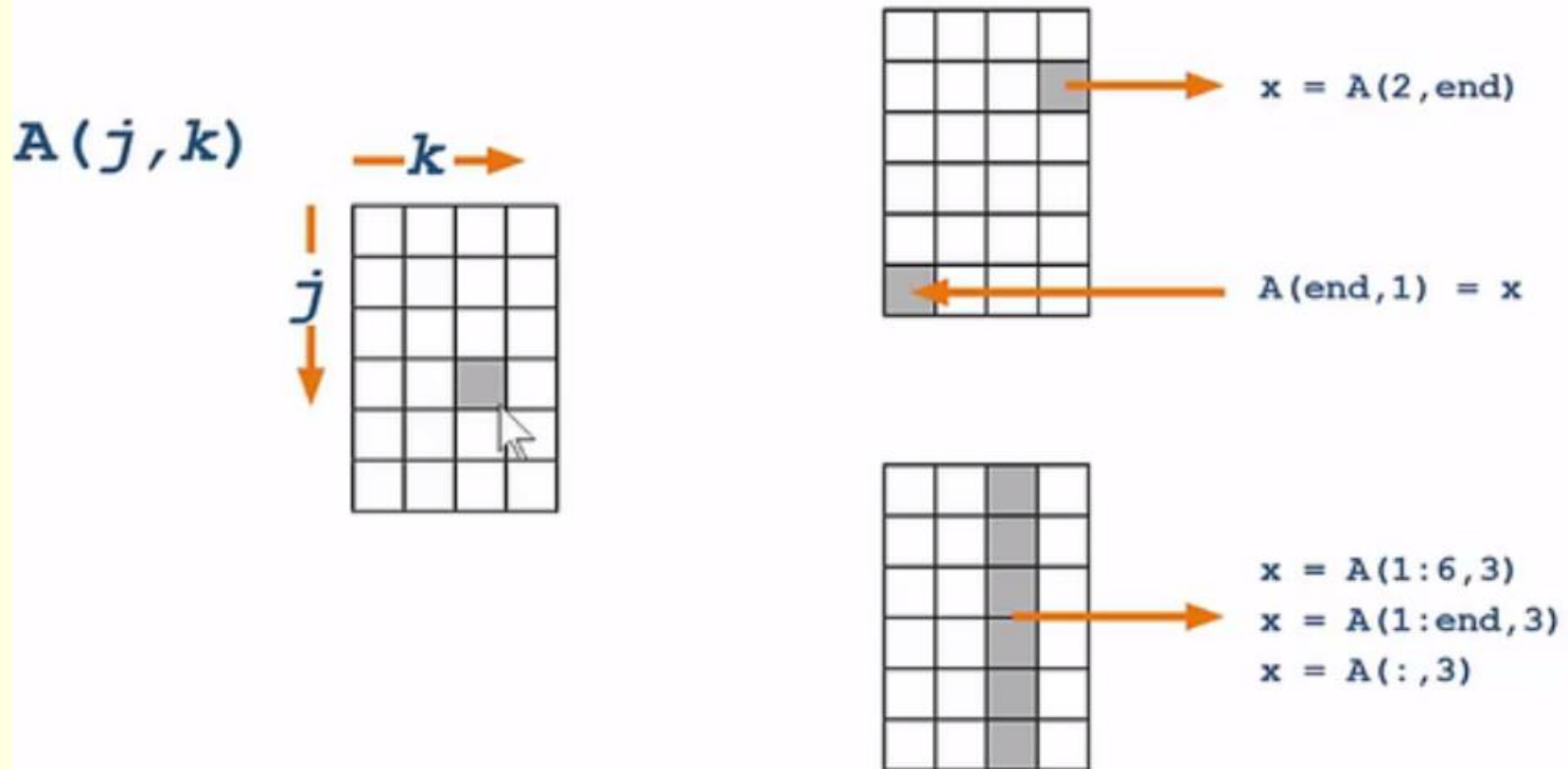
```
>> F=[A,B]
```

d. 矩阵的转置

```
>> G=A'
```

## 1.4.3 矩阵的操作

### ■ 矩阵化的数据访问



## 1.4.3 矩阵的操作

$$A = \begin{bmatrix} 12 & 34 & -4 \\ 34 & 7 & 87 \\ 3 & 65 & 7 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 3 & -1 \\ 2 & 0 & 3 \\ 3 & -2 & 7 \end{bmatrix}$$

求下列表达式的值：

(1)  $A+6*B$  和  $A-B$       (2)  $A*B$  和  $A.*B$

(3)  $A^3$  和  $A.^3$       (4)  $A/B$  及  $B \setminus A$

(5)  $A$  的第 1 行与第 2 行对调

(6)  $B$  的第 1 行的第 1 个元素+第 2 行的第 3 个元素

## 1.4.3 矩阵的操作

- $A=[12,34,-4;34,7,87;3,65,7];$
- $B=[1,3,-1;2,0,3;3,-2,7];$
- $a1=A+6*B$
- $a1=A*B$
- $a2=A.*B$
- $a1=A^3$
- $a2=A.^3$
- $a1=A/B$
- $a1=B\backslash A$
- $A1=[A(2,:);A(1,:);A(3,:)]$
- $a1= A(1,1)+A(2,3)$

# 1.5 关系运算与逻辑运算

---

- 1.5.1 关系运算符

- 1.5.2 逻辑运算符

- 1.5.3 其他关系与逻辑函数

## 1.5.1 关系运算符

运算符	说明	运算符	说明
<	小于	<=	小于或等于
>	大于	>=	大于或等于
注意:	等于	~=	不等于

**==**是关系运算符，相等时返回**1**，不相等时返回**0**；

**=**是赋值运算符，将右侧表达式的结果赋给左侧的变量。

## 1.5.2 逻辑运算符

名称	运算符	说明
与运算	&	两个元素同为非零时，结果为1；否则为0。
或运算		两个元素同为零时，结果为0；否则为1。
非运算	~	单目运算符。元素为零，结果为1；元素为非零，结果为0。

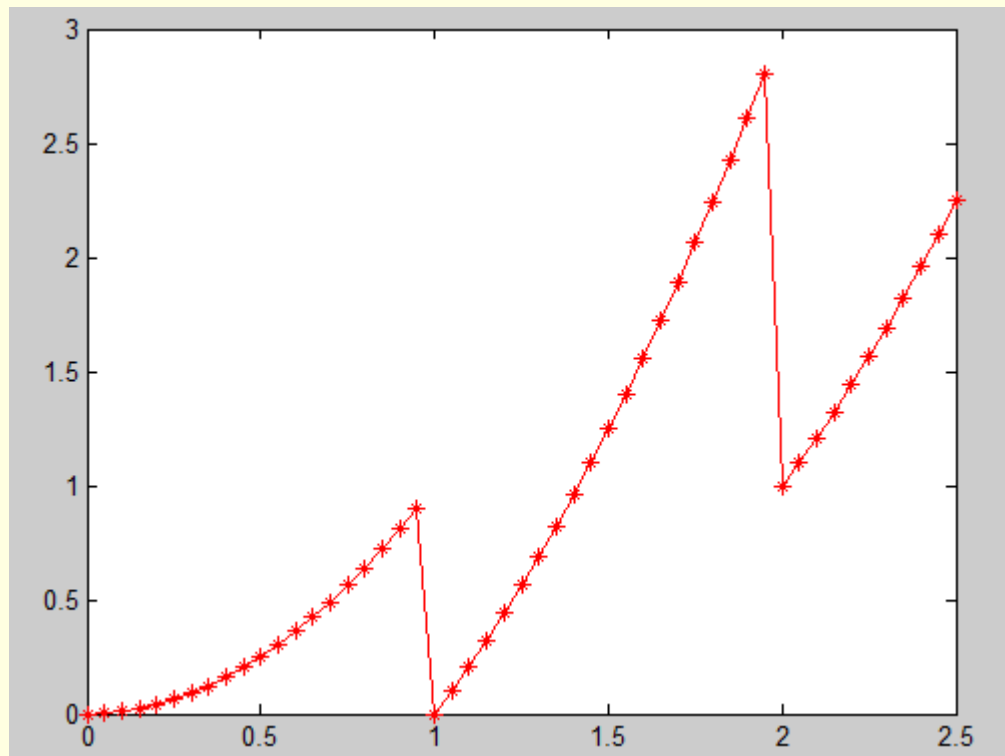


## 1.5.2 逻辑运算

- 已知 $A=0:9$ ,
- $B=10:-1:1$
- 计算:
- $A==B$
- $A \leq 5$
- $A > 3 \& A < 7$

## 1.5.2 逻辑运算

$$z_4 = \begin{cases} t^2 & 0 \leq t < 1 \\ t^2 - 1 & 1 \leq t < 2 \\ t^2 - 2t + 1 & 2 \leq t < 3 \end{cases}, \text{ 其中 } t=0:0.05:2.5$$



## 1.4.2、逻辑运算符

$$z_4 = \begin{cases} t^2 & 0 \leq t < 1 \\ t^2 - 1 & 1 \leq t < 2 \\ t^2 - 2t + 1 & 2 \leq t < 3 \end{cases}, \text{ 其中 } t=0:0.05:2.5$$

- `t=0:0.05:2.5;`
- `a=(t>=0&t<1).*(t.^2) ;`
- `b=(t>=1&t<2).*(t.^2-1);`
- `c=(t>=2&t<3) .*(t.^2-2*t+1);`
- `z4=a+b+c;`
- `plot(t,z4);`

## 1.5.3 其他关系与逻辑函数

函数格式	说明
<b>xor(x, y)</b>	异或运算。 <b>x</b> 和 <b>y</b> 相同，即都是零（假）或都是非零（真）结果为 <b>0</b> ； <b>x</b> 或 <b>y</b> 不同则结果为 <b>1</b>
<b>any(x)</b>	如果 <b>x</b> 是一个向量，含有一个或一个以上的非零元素，结果为 <b>1</b> ，否则为零；如果 <b>x</b> 是一个矩阵，结果是一个行向量，向量中的 <b>1</b> 对应矩阵中含有非零元素的列
<b>all(x)</b>	如果 <b>x</b> 是一个向量，所有元素均为非零时，结果为 <b>1</b> ；如果 <b>x</b> 是一个矩阵，结果是一个行向量，向量中的 <b>1</b> 对应矩阵中所有元素均非零的列