

● MATLAB 作为线性系统的一种分析和仿真工具，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 入门

一、变 量

二、数

组

三、矩

阵

四、 MATLAB 编程

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；
- (3) 变量名最多不超过 19 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

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，当和 1 相加就产生一个比 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

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、数学函数

函 数	名 称	函 数	名 称
$\sin(x)$	正弦函数	$\text{asin}(x)$	反正弦函数
$\cos(x)$	余弦函数	$\text{acos}(x)$	反余弦函数
$\tan(x)$	正切函数	$\text{atan}(x)$	反正切函数
$\text{abs}(x)$	绝对值	$\text{max}(x)$	最大值
$\text{min}(x)$	最小值	$\text{sum}(x)$	元素的总和
$\text{sqrt}(x)$	开平方	$\text{exp}(x)$	以 e 为底的指数
$\log(x)$	自然对数	$\log_{10}(x)$	以 10 为底的对数
$\text{sign}(x)$	符号函数	$\text{fix}(x)$	取整

二、数

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 **first : increment : last**

创建从 first 开始，加 increment 计数，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

2、 数组元素的访问

(1) **访问一个元素:** $x(i)$ 表示访问数组 x 的第 i 个元素

(2) **访问一块元素:** $x(a : b : c)$ 表示访问数组 x 的从第 a 个元素开始, 以步长为 b 到第 c 个元素 (但不超过 c), b 可以为负数, b 缺损时为 1.

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

$[x(a) \ x(b) \ x(c) \ x(d)]$.

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

(1) 标量 - 数组运算

数组对标量的加、减、乘、除、乘方是数组的每个元素对该标量施加相应的加、减、乘、除、乘方运算。设： $a=[a_1, a_2, \dots, a_n]$, c = 标量

$$\text{则: } a+c=[a_1+c, a_2+c, \dots, a_n+c]$$

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

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

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

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

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

(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**方式进行的，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： $a+b=[a_1+b_1,a_2+b_2,\dots,a_n+b_n]$

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

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

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

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

三、 矩 阵、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 键也表示开始一新行。输入矩阵时，严格要求所有行有相同的列。

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

$p = [1\ 1\ 1\ 1$

$2\ 2\ 2\ 2$

$3\ 3\ 3\ 3]$

特殊矩阵的建立： .

`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

`b=zeros(m , n)` 产生一个 `m` 行、 `n` 列的零矩阵

`c=ones(m , n)` 产生一个 `m` 行、 `n` 列的元素全为 1 的矩阵

`d=eye(m , n)` 产生一个 `m` 行、 `n` 列的单位矩阵

2、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 $i_1 \sim i_2$ 行、第 $j_1 \sim j_2$ 列构成新矩阵: $A(i_1:i_2, j_1:j_2)$

(5) 以逆序提取矩阵 A 的第 $i_1 \sim i_2$ 行, 构成新矩阵: $A(i_2:-1:i_1, :)$

(6) 以逆序提取矩阵 A 的第 $j_1 \sim j_2$ 列, 构成新矩阵: $A(:, j_2:-1:j_1)$

(7) 删除 A 的第 $i_1 \sim i_2$ 行, 构成新矩阵: $A(i_1:i_2, :) = []$

(8) 删除 A 的第 $j_1 \sim j_2$ 列, 构成新矩阵: $A(:, j_1:j_2) = []$

(9) 将矩阵 A 和 B 拼接成新矩阵: $[A \ B]; [A;$

B]

矩阵的抽取

1. 对角元素抽取函数：

`diag (A)` 抽取矩阵 A 的主对角线元素向量。

2. 上三角矩阵和下三角矩阵的抽取：

`tril (A)` 提取矩阵 A 的主下三角部分。

`tril (A , k)` 提取矩阵 A 的第 k 条对角线下面的部分 。

`triu (A)` 提取矩阵 A 的主上三角部分。

`triu (A , k)` 提取矩阵 A 的第 k 条对角线上面的部分

3、矩阵的运算

(1) 标量 - 矩阵运算

同标量 - 数组运算。

(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： $A+B$ 矩阵乘法： $A*B$

方阵的行列式： $\det(A)$

方阵的逆： $\text{inv}(A)$

方阵的特征值与特征向量： $[V, D]=\text{eig}[A]$

方阵的条件数： $\text{cond}(A,n)$

矩阵的范数： $\text{norm}(A,n)$ 矩阵的秩： $\text{rank}(A)$

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

Matlab 的应用程序也以 M 文件保存。 与函数名一致。

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)
```

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

2、逻辑运算符

逻辑操作符	说明
&	与
	或
~	非

控制流

MATLAB 提供三种决策或控制流结构：

for 循环、while 循环、if-else-end 结构。

这些结构经常包含大量的 MATLAB 命令，故经常出现在 MATLAB 程序中，而不是直接加在 MATLAB 提示符下。

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 $n=1,2,\dots,10$, 求 $\sin \frac{n \cdot \pi}{10}$ 的值

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}.

3 、 If-Else-End 结构

(1) 有一个选择的一般形式是:

```
if expression  
    {commands}  
end
```

如果在表达式 (expression) 里的所有元素为真, 就执行 if 和 end 语句之间的命令串 {commands}.

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

先建立 M 文件 fun1.m 定义函数 f (x)
， 再在 Matlab 命令窗口输入 fun1(2),fun1(-1) 即可。

2) 有三个或更多的选择的一般形式是:

```
if expression( )  
    statements;  
elseif expression( )  
    statements;  
.....  
else  
    statements;  
end
```


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** 定义函数 **f (x)**
，再在 Matlab 命令窗口输入 **fun2(2),fun2(0.5), fun2(-1)** 即可。

五 绘 图

1. 二维绘图调用格式:

`plot(x,y,s)` x,y 表示横坐标和纵坐标

2. 三维绘图调用格式:

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