

教育部直属国家211工程重点大学



大学数学基础实验

--大学数学实践教学系列课件--

内容提要



1

变量与数据操作

2

向量与矩阵

3

基础运算

4

字符串与元胞数组



MATLAB中的数据类型

❖ 共**16**个基础类

❖ **7**种基本数据类型，分别为

❖ 数值（**numeric**）、字符和字符串（**char and strings**）、逻辑（**logical**）、结构(**struct**)、细胞(**cell**)、表格（**table**）和函数句柄（**function handle**）。

2.1 变量与数据操作



2.1.1 变量与赋值

1) 变量命名

- ◆ 以字母开头
- ◆ 后面可以跟 任意字母、数字 或者 下划线
- ◆ 长度不超过 63 个字符
- ◆ 变量名 区分字母的**大小**写，变量中不能含有标点符号和空格符。



2) 赋值语句

(1) 变量=表达式

(2) 表达式

其中表达式是用运算符将有关运算量连接起来的式子，其结果是一个矩阵。

- 
- 逗号、分号的意义
 - 行内编辑、注释、运算规则

在运算式中，**MATLAB**通常不需要考虑空格；多条命令可以放在一行中，它们之间需要用分号隔开；逗号告诉**MATLAB**显示结果（屏幕打印），而分号则禁止结果显示。



例2-1 计算表达式的值，并显示计算结果。

在MATLAB命令窗口输入命令：

$x=1+2i$;

$y=3-\text{sqrt}(17)$;

$z=(\cos(\text{abs}(x+y))-\sin(78*\text{pi}/180))/(x+\text{abs}(y))$

其中pi和i都是MATLAB预先定义的变量，分别代表圆周率 π 和虚数单位。

输出结果是：

$z =$

$-0.3488 + 0.3286i$



2.1.2 预定义变量

在MATLAB工作空间中，还驻留几个由系统本身定义的变量。

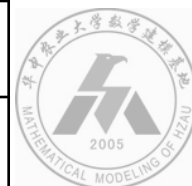
例如，用 **pi** 表示圆周率 π 的近似值，用 **i**, **j** 表示虚数单位 $\sqrt{-1}$ 。

预定义变量有特定的含义。

在使用时，应尽量避免给系统预定义变量重新赋值！（意思是说不能使用与预定义变量相同的变量名定义其它变量）

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
flops	浮点运算数
inf	无穷大 如1/0
nan	不等量 如0/0
i j	i=j=虚数单位
nargin	函数的输入变量数目
nargout	函数的输出变量数目
realmin	最小的可用正实数
realmax	最大的可用正实数



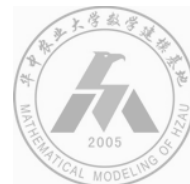


关键字

关键字是MATLAB程序设计中常用到的流程控制变量，共有20个，前面提到了不建议使用关键字作为变量的名字

在命令行窗口输入命令iskeyword，即可查询到：

```
>> iskeyword
```





复数的表示

不需要特殊的处理。复数可以表示为： $a=10-9i$
复数运算不需要特殊处理，可以直接进行。

2.1.3 内存变量的管理



1) 内存变量的删除与修改

MATLAB 工作空间窗口专门用于内存变量的管理。在工作空间窗口中可以显示所有内存变量的属性。

当选中某些变量后，再单击 Delete按钮，就能删除这些变量。

当选中某些变量后，再单击 Open 按钮，将进入变量编辑器。通过变量编辑器可以直接观察变量中的具体元素，也可修改变量中的具体元素。



clear命令用于删除MATLAB工作空间中的变量。

who和**whos**这两个命令用于显示在MATLAB工作空间中已经驻留的变量名清单。

who命令只显示出驻留变量的名称。

whos在给出变量名的同时，还给出它们的大小、所占字节数及数据类型等信息。

应用程序

查找文件 比较 导入数据 保存工作区 新建变量 打开变量 清除工作区 分析代码 运行和计时 清除命令 Simulink 库 布局 预设 设置路径 Parallel 帮助 社区 请求支持 附加功能

D:\Documents\MATLAB

命令窗口

```
>> whos
```

Name	Size	Bytes	Class	Attributes
A	2x2	32	double	
a	1x7	14	char	
f	1x16	32	char	
f1	1x11	22	char	
f2	1x13	26	char	
f3	1x8	16	char	

fx >> |

工作区

名称	值
a	'[55,56]'
A	[0.3333,2.5000;1.42..
f	'sin(x)+5*a*x^2+b'
f1	'a*x^2+b*x+c'
f2	'a*x^2+b*x+c=0'
f3	'Dy+y^2=1'



2) 内存变量文件

利用 MAT文件可以把当前MATLAB工作空间中的一些有用变量长久地保留下来，扩展名是 **.mat**。

MAT文件的生成和装入由save和load命令来完成。常用格式为：

save 文件名 [变量名表] [-append][-ascii]

load 文件名 [变量名表] [-ascii]



其中，文件名可以带路径，但不需带扩展名.mat，命令隐含一定对.mat文件进行操作。

变量名表中的变量个数不限，只要内存或文件中存在即可，变量名之间以空格分隔。当变量名表省略时，保存或装入全部变量。

-ascii选项使文件以ASCII格式处理，省略该选项时文件将以二进制格式处理。

save命令中的 -append 选项控制将变量追加到MAT文件中。

2.1.4 MATLAB常用数学函数



MATLAB 提供了许多数学函数，函数的自变量规定为矩阵变量，运算法则是将函数逐项作用于矩阵的元素上，因而运算的结果是一个与自变量同维数的矩阵。



`sin、cos、tan、cot、sec、csc、...`

`asin、acos、atan、acot、asec、acsc、...`

`exp、log、log2、log10、sqrt`

`abs、conj、real、imag、sign`

`fix、floor、ceil、round、mod、rem`

`max、min、sum、mean、sort、fft`

`norm、rank、det、inv、eig、lu、qr、svd`

`.....`



函数使用说明：

(1) 三角函数以弧度为单位计算。

(2) `abs` 函数可以求实数的绝对值、复数的模、字符串的ASCII码值。

(3) 用于取整的函数有 `fix`、`floor`、`ceil`、`round`，要注意它们的区别。

(4) `rem`与 `mod`函数的区别。`rem(x,y)`和 `mod(x,y)` 要求 `x`, `y` 必须为相同大小的实矩阵或为标量。

2.1.5 数据的输出格式



MATLAB用十进制数表示一个常数，具体可采用日常记数法和科学记数法两种表示方法。

在一般情况下，**MATLAB**内部每一个数据元素都是用双精度数来表示和存储的。显示规则：

在缺省情况下，当结果为整数，作为整数显示；当结果为实数，以小数后4位的精度近似显示。

如果结果中的有效数字超出了这一范围，以科学计数法显示结果。

数据输出时用户可以用**format**命令设置或改变数据输出格式。**format**命令的格式为：

format 格式符

其中格式符决定数据的输出格式。

各种 format 格式

格式	解释	例
format	短格式（缺省显示格式），同short	3.1416
format short	短格式（缺省显示格式），只显示5位	3.1416
format long	长格式，双精度数15位，单精度数7位	3.14159265358979
format short e	短格式e方式（科学计数格式）	3.1416e+000
format long e	长格式e方式	3.141592653589793e+000
format short g	短格式g方式	3.1416
format long g	长格式g方式	3.14159265358979
format compact	压缩格式	
format loose	自由格式	
format + / format bank / format rat / format hex （详情查看联机帮助）		

2.2 MATLAB中的向量与矩阵



2.2.1 向量与矩阵的生成

◆ 向量的生成

✓ 直接输入: $\mathbf{a}=[1,2,3,4]$

✓ 冒号运算符

例: $\mathbf{a}=[1:4] \Rightarrow \mathbf{a}=[1, 2, 3, 4]$

$\mathbf{b}=[0:\pi/3:\pi] \Rightarrow \mathbf{b}=[0, 1.0472, 2.0944, 3.1416]$

$\mathbf{c}=[6:-2:0] \Rightarrow \mathbf{c}=[6, 4, 2, 0]$

✓ 从矩阵中抽取行或列



◆ 矩阵的生成

✓ 直接输入: **$A=[1, 2, 3; 4, 5, 6; 7, 8, 9]$**

✓ 由向量生成

例: `>> x=[1,2,3]; y=[2,3,4];`

`>> A=[x,y], B=[x;y]`

✓ 由函数生成

例: `>> C=magic(3)`

✓ 通过编写m文件生成

2.2.2 矩阵的拆分



1. 矩阵元素

通过下标引用矩阵的元素，例

$$A(3,2)=200$$


采用矩阵元素的序号来引用矩阵元素。矩阵元素的序号就是相应元素在内存中的排列顺序。在 **MATLAB** 中，矩阵元素按列存储，先第一列，再第二列，依次类推。例如

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

$$A(3)$$

ans =

2



显然，序号(Index)与下标(Subscript)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 $(j-1)*m+i$ 。其相互转换关系也可利用sub2ind和ind2sub函数求得。

2. 矩阵拆分

(1) 利用冒号表达式获得子矩阵

① $A(:,j)$ 表示取A矩阵的第j列全部元素； $A(i,:)$ 表示A矩阵第i行的全部元素； $A(i,j)$ 表示取A矩阵第i行、第j列的元素。

② $A(i:i+m,:)$ 表示取A矩阵第i~i+m行的全部元素； $A(:,k:k+m)$ 表示取A矩阵第k~k+m列的全部元素， $A(i:i+m,k:k+m)$ 表示取A矩阵第 i~i+m行内，并在第k~k+m列中的所有元素。

此外，还可利用一般向量和end运算符来表示矩阵下标，从而获得子矩阵。end表示某一维的末尾元素下标。如 $A([1,3,4])$ $A(2:end)$

(2) 利用空矩阵删除矩阵的元素

在MATLAB中，定义[]为空矩阵。给变量X赋空矩阵的语句为X=[]。

如 $A(2,:)=[]$ %可以删去矩阵第二行的内容

注意， $X=[]$ 与 `clear X` 不同，`clear` 是将X从工作空间中删除，而空矩阵则存在于工作空间中，只是维数为0。

2.2.3 一些常见矩阵生成函数



1) 通用的特殊矩阵

常用的产生通用特殊矩阵的函数有：

zeros: 产生全0矩阵(零矩阵)。

ones: 产生全1矩阵(幺矩阵)。

eye: 产生单位矩阵。

rand: 产生0~1间均匀分布的随机矩阵。

randn: 产生均值为 0，方差为 1 的标准正态分布随机矩阵。

zeros(m,n)	生成一个 m 行 n 列零矩阵, $m=n$ 时可简写 zeros(n)
ones(m,n)	生成一个 m 行 n 列的元素全为 1 的矩阵, $m=n$ 时可写为 ones(n)
eye(m,n)	生成一个主对角线全为 1 的 m 行 n 列矩阵, $m=n$ 时可简写为 eye(n) , 即为 n 维单位矩阵
diag(X)	if X 是矩阵, 则 diag(X) 为 X 的主对角线向量; if X 是向量, diag(X) 是以 X 为主对角线对角矩阵
tril(A)	提取一个矩阵的下三角部分
triu(A)	提取一个矩阵的上三角部分
rand(m,n)	产生 0~1 均匀分布的随机矩阵 $m=n$ 时简写为 rand(n)
randn(m,n)	产生均值为0, 方差为1的标准正态分布随机矩阵 $m=n$ 时简写为 randn(n)



例2-3 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2-4 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。



2) 用于专门学科的特殊矩阵

(1) 魔方矩阵

魔方矩阵有一个有趣的性质，其每行、每列及两条对角线上的元素和都相等。对于 n 阶魔方阵，其元素由 $1, 2, 3, \dots, n^2$ 共 n^2 个整数组成。

MATLAB提供了求魔方矩阵的函数`magic(n)`，其功能是生成一个 n 阶魔方阵。



例2-5 将101~125等25个数填入一个5行5列的表格中，使其每行每列及对角线的和均为565。

$$\mathbf{M=100+magic(5)}$$



(2) 范得蒙矩阵

范得蒙(Vandermonde)矩阵最后一列全为1，倒数第二列为一个指定的向量，其他各列是其后续列与倒数第二列的点乘积。可以用一个指定向量生成一个范得蒙矩阵。

在MATLAB中，函数`vander(V)`生成以向量`V`为基础向量的范得蒙矩阵。

例如，`A=vander([1;2;3;5])`即可得到上述范得蒙矩阵。

(3) 托普利兹矩阵

托普利兹(Toeplitz)矩阵除第一行第一列外，其他每个元素都与左上角的元素相同。

生成托普利兹矩阵的函数是`toeplitz(x,y)`，它生成一个以`x`为第一列，`y`为第一行的托普利兹矩阵。这里`x,y`均为向量，两者不必等长。`toeplitz(x)`用向量`x`生成一个对称的托普利兹矩阵。

例如 `T=toeplitz(1:6)`



(4) 帕斯卡矩阵

我们知道，二次项 $(x+y)^n$ 展开后的系数随 n 的增大组成一个三角形表，称为杨辉三角形。

由杨辉三角形表组成的矩阵称为帕斯卡(Pascal)矩阵。

函数`pascal(n)`生成一个 n 阶帕斯卡矩阵。



例2-6 求 $(x+y)^5$ 的展开式。

在MATLAB命令窗口，输入命令：

pascal(6)

矩阵次对角线上的元素1,5,10,10,5,1即为展开式的系数。



2.2.4 矩阵分析

1. 对角阵与三角阵

1) 对角阵

只有对角线上有非 0 元素的矩阵称为对角矩阵，对角线上的元素相等的对角矩阵称为数量矩阵，对角线上的元素都为 1 的对角矩阵称为单位矩阵。

(1) 提取矩阵的对角线元素

设 A 为 $m \times n$ 矩阵， $\text{diag}(A)$ 函数用于提取矩阵 A 主对角线元素，产生一个具有 $\min(m,n)$ 个元素的列向量。

$\text{diag}(A)$ 函数还有一种形式 $\text{diag}(A,k)$ ，其功能是提取第 k 条对角线的元素。

(2) 构造对角矩阵 设 V 为具有 m 个元素的向量， $\text{diag}(V)$ 将产生一个 $m \times m$ 对角矩阵，其主对角线元素即为向量 V 的元素。

$\text{diag}(V)$ 函数也有另一种形式 $\text{diag}(V,k)$ ，其功能是产生一个 $n \times n$ ($n=m+|k|$ ， k 为一整数) 对角阵，其第 k 条对角线的元素即为向量 V 的元素。



例2.7 先建立 5×5 矩阵A，然后将A的第一行元素乘以1，第二行乘以2，...，第五行乘以5。

```
>>A=[17,0,1,0,15;23,5,7,14,16;4,0,13,0,22;10,12,19,21,  
3;11,18,25,2,19];
```

```
D=diag(1:5);
```

```
D*A           %用D左乘A，对A的每行乘以一个  
指定常数
```

请大家动手操作这几条语句



2) 三角阵

三角阵分为上三角阵和下三角阵。

上三角阵，即矩阵的对角线以下的元素全为 0 的一种矩阵，

下三角阵，则是对角线以上的元素全为 0 的一种矩阵。

(1) 上三角矩阵

求矩阵A的上三角阵的MATLAB函数是`triu(A)`。

`triu(A)` 函数也有另一种形式 `triu(A,k)`，其功能是求矩阵A的第k条对角线以上的元素。例如，

$$B = \text{triu}(A, 2)$$

即提取矩阵A的第2条对角线以上的元素，形成新的矩阵B。

(2) 下三角矩阵

在MATLAB中，提取矩阵A的下三角矩阵的函数是`tril(A)`和`tril(A,k)`，其用法与提取上三角矩阵的函数`triu(A)`和 `triu(A,k)`完全相同。



2. 矩阵的转置与旋转

1) 矩阵的转置

转置运算符是单撇号(')。

2) 矩阵的旋转

利用函数`rot90(A,k)`将矩阵A旋转 90° 的k倍，当k为1时可省略。



3) 矩阵的左右翻转

对矩阵实施左右翻转是将原矩阵的第一列和最后一列调换，第二列和倒数第二列调换，...，依次类推。

MATLAB对矩阵**A**实施左右翻转的函数是 **fliplr(A)**。

4) 矩阵的上下翻转

MATLAB对矩阵**A**实施上下翻转的函数是 **flipud(A)**。



3. 矩阵的逆与伪逆

1) 矩阵的逆

对于一个方阵**A**，如果存在一个与其同阶的方阵**B**，使得：

$$\mathbf{A} \cdot \mathbf{B} = \mathbf{B} \cdot \mathbf{A} = \mathbf{I} \text{ (I为单位矩阵)}$$

则称**B**为**A**的逆矩阵，当然，**A**也是**B**的逆矩阵。

求一个矩阵的逆是一件非常烦琐的工作，容易出错，但在MATLAB中，求一个矩阵的逆非常容易。

求方阵**A**的逆矩阵可调用函数`inv(A)`。



例2.8 用求逆矩阵的方法解线性方程组。

$$\mathbf{Ax}=\mathbf{b}$$

其解为：

$$\mathbf{x}=\mathbf{A}^{-1}\mathbf{b}$$

命令为：

$$\mathbf{a}=[2,-3,1;8,3,2;45,1,-9];$$

$$\mathbf{b}=[4;2;17];$$

$$\mathbf{x}=\text{inv}(\mathbf{a})*\mathbf{b}$$

2) 矩阵的伪逆

如果矩阵A不是一个方阵，或者 A是一个非满秩的方阵时，矩阵A没有逆矩阵，但可以找到一个与A的转置矩阵 A'同型的矩阵B，使得：

$$\mathbf{A} \cdot \mathbf{B} \cdot \mathbf{A} = \mathbf{A}$$

$$\mathbf{B} \cdot \mathbf{A} \cdot \mathbf{B} = \mathbf{B}$$

此时称矩阵B为矩阵A的伪逆，也称为广义逆矩阵。
在MATLAB中，求一个矩阵伪逆的函数是pinv(A)。



4. 方阵的行列式

把一个方阵看作一个行列式，并对其按行列式的规则求值，这个值就称为矩阵所对应的行列式的值。

在MATLAB中，求方阵A所对应的行列式的值的函数是 $\det(A)$ 。



5. 矩阵的秩与迹

1) 矩阵的秩

矩阵线性无关的行数与列数称为矩阵的秩。在 **MATLAB** 中，求矩阵秩的函数是 **rank(A)**。

2) 矩阵的迹

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。在 **MATLAB** 中，求矩阵的迹的函数是 **trace(A)**。

2.3 MATLAB的基础运算



1. 基本算术运算

MATLAB的基本算术运算有：

+(加)、-(减)、
*(乘)、/(右除)、\ (左除)、
^(乘方)。

注意，运算是在矩阵意义下进行的，单个数据的算术运算只是一种特例。



(1) 矩阵加减运算

假定有两个矩阵**A**和**B**，则可以由 **$A+B$** 和 **$A-B$** 实现矩阵的加减运算。

运算规则是：若**A**和**B**矩阵的维数相同，则可以执行矩阵的加减运算，**A**和**B**矩阵的相应元素相加减。如果**A**与**B**的维数不相同，则**MATLAB**将给出错误信息，提示用户两个矩阵的维数不匹配。

(2) 矩阵乘法

假定有两个矩阵**A**和**B**，若**A**为 **$m \times n$** 矩阵，**B**为 **$n \times p$** 矩阵，则 **$C=A*B$** 为 **$m \times p$** 矩阵。


(3) 矩阵除法

在MATLAB中，有两种矩阵除法运算： \backslash 和 $/$ ，分别表示左除和右除。

如果A矩阵是非奇异方阵，则 $A\backslash B$ 和 B/A 运算可以实现。

$A\backslash B$ 等效于A的逆左乘B矩阵，也就是 $\text{inv}(A)*B$ ，而 B/A 等效于A矩阵的逆右乘B矩阵，也就是 $B*\text{inv}(A)$ 。

注：若n阶方阵A的行列式不为零，即 $|A| \neq 0$ ，则称A为非奇异矩阵或满秩矩阵。



对于含有标量的运算，两种除法运算的结果相同，如 $3/4$ 和 $4\backslash 3$ 有相同的值，都等于0.75。又如，设 $a=[10.5, 25]$ ，则 $a/5=5\backslash a=[2.1000 \ 5.0000]$ 。

对于矩阵来说，左除和右除表示两种不同的除数矩阵和被除数矩阵的关系。对于矩阵运算，一般 $A\backslash B \neq B/A$ 。

(4) 矩阵的乘方

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。

2. 点运算

在MATLAB中，有一种特殊的运算，因为其运算符是在有关算术运算符前面加点，所以叫点运算。

点运算符有 $.*$ 、 $./$ 、 $.\backslash$ 和 $.^$ 。两矩阵进行点运算是指它们的对应元素进行相关运算，要求两矩阵的维数相同。

```
>>A=[1,2;3,4];B=[4,3;2,1];  
A.*B  
A./B  
A.^2
```



3. 关系运算

MATLAB提供了6种关系运算符:

$<$ (小于)、 \leq (小于或等于)、 $>$ (大于)、
 \geq (大于或等于)、 $==$ (等于)、 \sim (不等于)。

它们的含义不难理解，但要注意其书写方法与数学中的不等式符号不尽相同。



关系运算符的运算法则为：

(1) 当两个比较量是标量时，直接比较两数的大小。若关系成立，关系表达式结果为1，否则为0。

(2) 当参与比较的量是两个维数相同的矩阵时，比较是对两矩阵相同位置的元素按标量关系运算规则逐个进行，并给出元素比较结果。最终的关系运算的结果是一个维数与原矩阵相同的矩阵，它的元素由0或1组成。



(3) 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，并给出元素比较结果。最终的关系运算的结果是一个维数与原矩阵相同的矩阵，它的元素由0或1组成。



例2.9 产生5阶随机方阵 A，其元素为 [10,90]区间的随机整数，然后判断A的元素是否能被3整除。

(1) 生成5阶随机方阵A:

$A=\text{fix}((90-10+1)*\text{rand}(5)+10)$

(此处fix函数是取整)

(2) 判断A的元素是否可以被3整除:

$P=\text{rem}(A,3)==0$

其中，**rem(A,3)**是矩阵A的每个元素除以3的余数矩阵。此时，**0**被扩展为与A同维数的零矩阵，**P**是进行等于(==)比较的结果矩阵。

请大家动手输入上面的语句，并参看演示结果。




4. 逻辑运算

MATLAB提供了3种逻辑运算符：

$\&$ (与)、 $|$ (或)和 \sim (非)。

逻辑运算的运算法则为：

(1) 在逻辑运算中，确认非零元素为真，用1表示，零元素为假，用0表示。（其实就是二值运算符号，或称为布尔代数）



(2) 设参与逻辑运算的是两个标量 a 和 b ，那么：

$a \& b$ a, b 全为非零时，运算结果为1，否则为0。

$a | b$ a, b 中只要有一个非零，运算结果为1。

$\sim a$ 当 a 是零时，运算结果为1；
当 a 非零时，运算结果为0。

(3) 若参与逻辑运算的是两个同维矩阵，那么运算将对矩阵相同位置上的元素按标量规则逐个进行。最终运算结果是一个与原矩阵同维的矩阵，其元素由 1或 0组成。



(4) 若参与逻辑运算的一个是标量，一个是矩阵，那么运算将在标量与矩阵中的每个元素之间按标量规则逐个进行。最终运算结果是一个与矩阵同维的矩阵，其元素由 1 或 0 组成。

(5) 逻辑非是单目运算符，也服从矩阵运算规则。

(6) 在算术、关系、逻辑运算中，算术运算优先级最高，逻辑运算优先级最低。



例2.10 建立矩阵A，然后找出大于4的元素的位置。

(1) 建立矩阵A:

$A=[4,-65,-54,6;56,0,67,-45]$

(2) 找出大于4的元素的位置:

`find(A>4)`

请大家动手实践一下，然后再看看课件操作

2.4 字符串与元胞数组



2.4.1 字符串

在MATLAB中，字符串是用单撇号(')括起来的字符序列。

MATLAB 将字符串当作一个行向量，每个元素对应一个字符，其标识方法和数值向量相同。也可以建立多行字符串矩阵。



字符串是以**ASCII**码形式存储的。**abs**和**double**函数都可以用来获取字符串矩阵所对应的**ASCII**码数值矩阵。

相反，**char**函数可以把**ASCII**码矩阵转换为字符串矩阵。



例2.11 建立一个字符串向量，然后对该向量做如下处理：

- (1) 取第1~5个字符组成的子字符串。
- (2) 将字符串倒过来重新排列。
- (3) 将字符串中的小写字母变成相应的大写字母，其余字符不变。
- (4) 统计字符串中小写字母的个数。



命令如下：

```
ch='ABc123d4e56Fg9';  
subch=ch(1:5)           %取子字符串  
revch=ch(end:-1:1)      %将字符串倒排  
k=find(ch>='a'&ch<='z'); %找小写字母的位置  
ch(k)=ch(k)-('a'-'A'); %将小写字母变成相应的大写字母  
char(ch)  
length(k)                %统计小写字母的个数
```



与字符串有关的另一个重要函数是**eval**，其调用格式为：

eval(t)

其中**t**为字符串。它的作用是把字符串的内容作为对应的**MATLAB**语句来执行。

如：

```
t='sin(1)';  
eval(t)
```



2.4.2 元胞数组

- ❖ **元胞数组**是**MATLAB**的一种特殊数据类型，可以将元胞数组看做一种无所不包的通用矩阵(广义矩阵)。
- ❖ 组成元胞数组的元素可以是任何一种数据类型的常数或者常量，每一个元素也可以具有不同的尺寸和内存占用空间，每一个元素的内容也可以完全不同，所以元胞数组的元素叫做**元胞**（**cell**）。
- ❖ 和一般的数值 矩阵一样，元胞数组的内存空间也是动态分配的。



❖ 元胞数组（**cell**）可以将浮点型、字符型、结构数组等不同类型的数据放在同一个存储单元中。

创建元胞数组



❖ 使用 {} 运算符或 cell 函数创建元胞数组。

(1) 使用元胞数组构造运算符 {} 创建该数组。

```
>>myCell = {1, 2, 3; 'text',  
rand(5,10,2), {11; 22; 33}}
```

myCell = 2x3 cell array

{[1]}

{[2]}

{[3]}

{'text'} {5x10x2 double} {3x1 cell}

元胞数组也是矩阵，每一行中具有相同的元胞数。

使用 {} 运算符创建一个空的 **0×0** 元胞数组

```
>>C = {}
```



(2)使用 **cell** 函数创建一个空的 **N** 维数组

如要随时间推移或要循环向元胞数组添加值，可使用 **cell** 函数先创建一个空的 **N** 维数组。

```
>>emptyCell = cell(3,4,2)
```

创建一个由空矩阵构成的3阶元胞数组。

```
>>emptyCell3 = cell(3)
```

元胞数组的基本操作



1、采用“()”或“{}”形式访问元胞数组的元素。在命令行窗口输入：

```
>>A={'I am a student ',pi;{1,101},rand(3)}
```

创建元胞数组A。

注意观察A(2,1)和A{2,1}区别：

```
>> A(2,1)
ans = {1x2 cell}
>> A{2,1}
ans = [1] [101]
```

可以看到，A(2,1)返回的是元胞数组A在（2,1）位置上的元胞，而A{2,1}返回的是元胞数组A在（2,1）位置上的元胞中的数据。

元胞数组的基本操作



2、创建字符串元胞数组cellstr，在命令行窗口接着输入：

```
>> B=char('apple','pear','watermelon','rice');  
>> C=cellstr(B)
```

回车返回如下结果：

C =

'apple'

'pear'

'watermelon'

'rice'

元胞数组的基本操作

3、显示元胞数组的内容celldisp, 在命令行窗口接着输入:

```
>> celldisp(A)
```

```
>> celldisp(A)
```

```
A{1,1} =
```

```
I am a student
```

```
A{2,1}{1} =
```

```
1
```

```
A{2,1}{2} =
```

```
101
```

```
A{1,2} =
```

```
3.1416
```

```
A{2,2} =
```

0.8147	0.9134	0.2785
0.9058	0.6324	0.5469
0.1270	0.0975	0.9575

fx



❖ 本讲内容到此为止，谢谢！