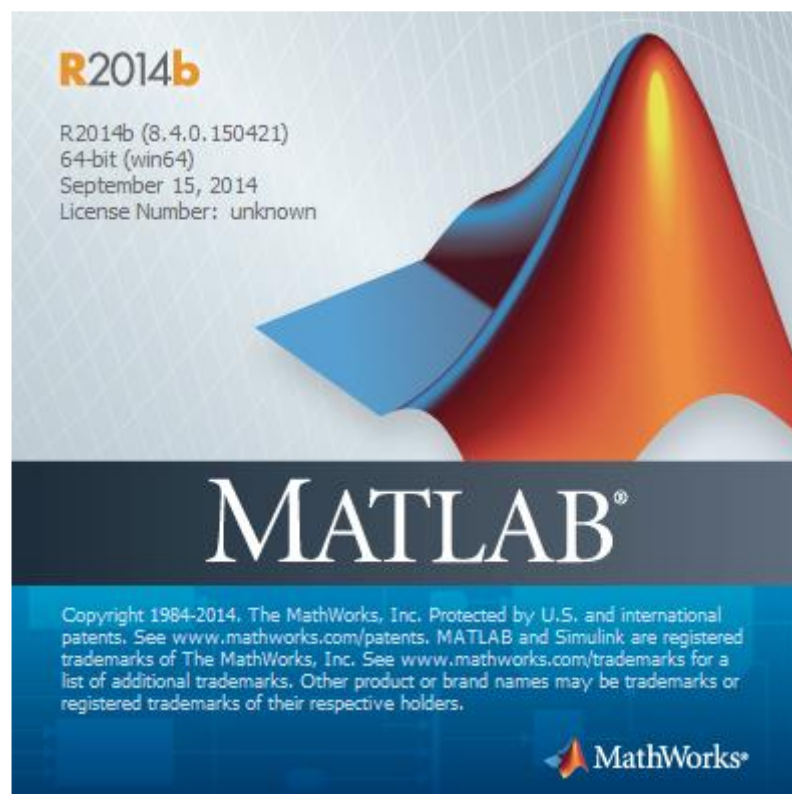


第2章a-20200227更新

MATLAB软件与应用



你更期望用哪个平台上课

- ☒ **A 雨课堂**
- ☐ **B 腾讯会议**
- ☐ **C 其他**

提交

前言

授课：雨课堂**GAXZUT**+腾讯会议128479845

备用：学校课程平台(下载课件自学)

课后交流：QQ群**1046970565**

第二章 矩阵运算

- 常量，变量，数据类型
 - 矩阵的类型与定义方式
 - 矩阵的基本操作与运算
-

- 字符串、元胞与结构

常量，变量，数据类型

常量

常量可用于对变量进行赋值，或参与构成表达式。

数值: 10 , 10.0%两种写法是相同的
 0.3 , .3 %两种写法是相同的
 -123e-2 %即1.23
 2-3.5i %3.5与i之间不能有空格
 1+25e-1j %即1.0000 + 2.5000i

字符 : 'A'

字符(串) : 'ABC'

特殊常量

MATLAB预先定义了一些特殊常量（实质上是变量，用户可以对其重新赋值，但不建议这样操作）

变量名	意义
ans	最近一次计算结果的默认变量名
eps	预设的正极小值=2.2204e-16
pi	圆周率 π
Inf	∞ 值，无限大
i或j	虚数单元， $\text{sqrt}(-1)$
NaN	非数，0/0、 ∞/∞
nargin	M函数入口参数，记录实际输入参数数量
nargout	M函数出口参数，记录实际输出参数数量
realmin	最小的正浮点数2.2251e-308
realmax	最大的正浮点数1.7977e+308
bitmax	最大的正整数9.0072e+015

变量命名规则

□ 变量命名原则

- ◆ 以字母开头
- ◆ 后面可以跟 字母、数字 和 下划线
- ◆ 长度不超过 63 位（31字符）
- ◆ 区分字母的大小写
- ◆ 避免使用系统的特殊保留字（for, end, if 等）
- ◆ 避免同函数名、M文件名重名（例如内置函数 sin 等）

□ 赋值语句的通常形式

变量 = 表达式

例 A=20; a=10;
 b='ok'
 c=zeros(3)
 d=[1,2]

变量与数据类型

Matlab数据类型非常丰富，各种数据以**矩阵**为基本形式存在。

常用的数据类型如下表：

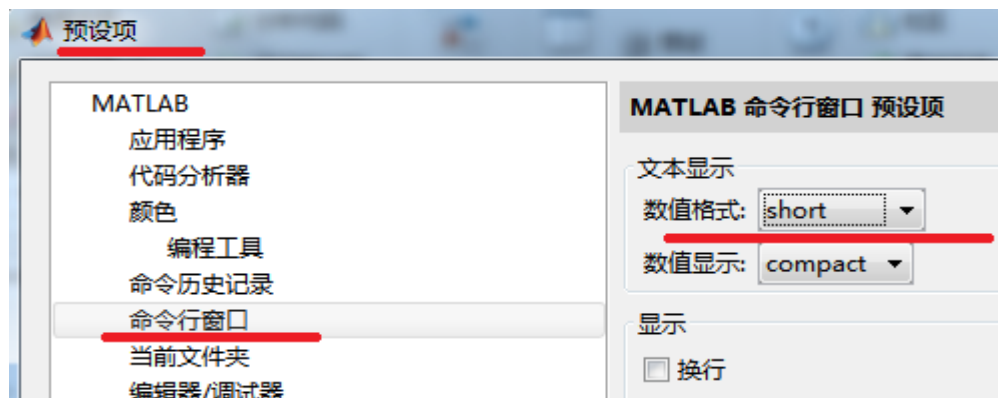
数值数据

类型	声明函数
整数	int8, int16, int32, int64 unit8, unit16, unit32, unit64
浮点数	double , single
逻辑型	logical
字符型	char
符号型	sym 或 syms
结构	struct
元胞	cell
时间日期型，函数句柄型等 ...	

数值的显示形式

Matlab命令	例：数值 2/3 的显示	说 明
format short (默认)	0.6667	短浮点
format long	0.6666666666666667	长浮点
format short e	6.6667e-01	短科学计数
format long e	6.666666666666666e-01	长科学计数
format hex	3fe5555555555555	十六进制
format bank	0.67	银行计数（元角分）
format rat	2/3 (π 显示为 355/113)	分数近似（近似有理数表示）
其他: format short g, format long g, format compact, format loose, format +		

可以在菜单中
修改默认显示格式



赋值即定义原则

Matlab变量一般不需要专门定义和声明，在变量赋值的同时系统会判断赋值的方式并自动设置变量类型。**对变量再次赋值会刷新其数据类型。**

$x1=20$, $x2=20.0$, $x3=0.5$ %自动成为double浮点数

$y1='c'$ %y1自动成为字符型

$y2=\text{int16}(3.5)$ %y2自动成为int16整型

有时也需要一些特定处理

$\text{syms } y3$ % 定义y3为符号型

$x=y3+2$ % x被赋值并自动成为符号型

$x=5$ % x重新赋值并自动成为double浮点型

赋值即定义原则在多种场合被广泛运用，

【例】 如果A是2x2 double型矩阵(每个元素是double型)，那么

$B=A*2$ %B自动成为2x2矩阵，元素都是double浮点型

数据类型的转换

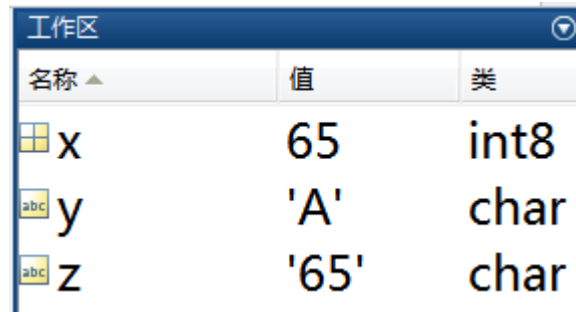
◆ 数据类型可以进行转换

`x=int8(65);` %转换为8位整数65

`y=char(65);` %转换为65号ASCII码代表的字符A

`z=num2str(65);` %转换为字符串65，相当于`z='65'`

工作空间显示的结果是



The image shows a screenshot of the MATLAB 'Workspace' window. It contains a table with three columns: '名称' (Name), '值' (Value), and '类' (Class). The table lists three variables: 'x' with value 65 and class int8, 'y' with value 'A' and class char, and 'z' with value '65' and class char.

名称	值	类
x	65	int8
y	'A'	char
z	'65'	char

再观察以下命令的执行结果,并用**whos**命令查看。

`w1=sym('b+c');` %字符串'b+c'转换为符号型

`w2=sym('c*2-b');` %字符串'c*2-b'转换为符号型

`w3=w1+w2` %变量**w3**为符号型，其值为 $3*c$

复数

浮点数定义在复数域上，复数的输入

$z=3+4i$ %4 与 i 之间不能有空格, i需小写

$z=3+4* i$

Matlab中复数运算遵守数学上的一般运算方法

$z1=1+2*i, \quad z2=2+2*j, \quad z=z1+z2$

还提供了一些专门函数对复数进行操作

$z_real=real(z), \quad z_image=imag(z)$

$z_angle=angle(z), \quad z_length=abs(z)$

变量的存储和载入

□ 存储当前工作区中的变量

`save` 所有变量存入默认文件 `matlab.mat`
`save abc` 所有变量存入文件 `abc.mat`
`save abc.mat` 所有变量存入文件 `abc.mat`

默认存储位置
是当前目录

`save c:\abc.mat` 所有变量存入文件 `c:\abc.mat`

指定了存储位置

□ 存储指定的变量

`save` 文件名 变量名列表

变量名列表中各变量之间用空格分隔

`save abc.mat A x z` 存储了三个变量

变量的存储和载入

❑ 将数据文件中的变量载入工作空间

`load` 文件名 变量名列表

`load abc.mat` 载入数据文件中的所有变量

`load abc.mat A x` 从数据文件中提取指定变量

❑ 清除当前工作空间中的变量

`clear` 清除工作空间中的所有变量

`clear A x` 清除工作空间中两个指定变量

变量的存储和载入

- 还可以按照**指定格式**进行变量的存储和载入

```
save ('文件名', '格式', '变量名')
```

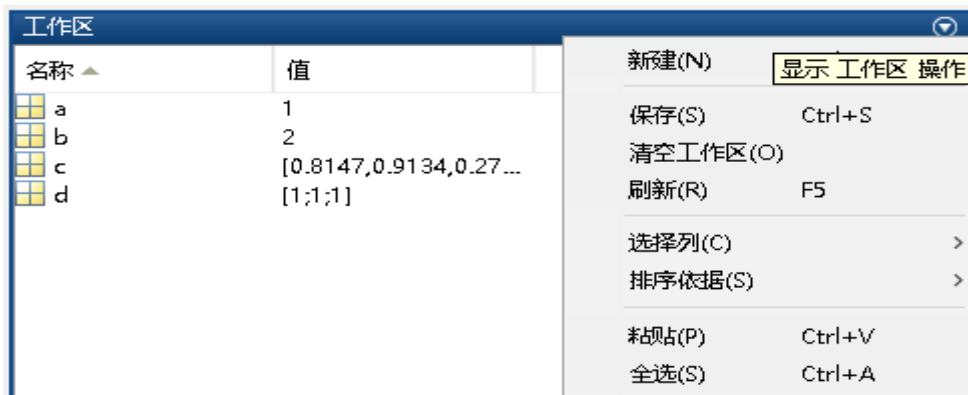
```
load ('文件名', '格式', '变量名')
```

```
save ('abc.mat', '-ascii', 'A', 'B')
```

将变量A以8位文本格式存入文件 **abc.mat**

关于格式的用法等详细说明，可参考系统自带文档。

- 还可以通过**工作空间窗口**进行交互式数据存取



矩阵的类型与定义方式

矩阵（数组）的基本定义

按行(row)和列(column)顺序排列的实数或复数的有序集，被称为矩阵或数组，matlab对矩阵/数组的概念进行了扩展，不仅可处理数值，还能处理其他多种数据类型，不仅有一维、二维，还有多维形式。

矩阵中任何一个数都被称为元素，由其所在的行和列来标识，也称为标识、下标或索引。

一个数量(标量)视为 1×1 矩阵；

m行、n列的矩阵a，计为 $m \times n$ 矩阵a；

行下标、列下标均从1开始；

行下标从上到下递增，列下标从左到右递增。

矩阵（数组）的基本定义

矩阵的一个重要属性是其形状（尺寸size），如下表所示

- 一维矩阵/数组，也称为向量(vector)。

行向量(row vector)

列向量(column vector)

- 二维矩阵/数组

每行元素的个数须相同

每列元素的个数须相同

- 多维矩阵/数组

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

矩阵的创建 利用操作符 []

□ 在[]内依次输入矩阵的所有元素

创建4行3列矩阵A，按行依次输入

```
A = [1 2 3; 4 5 6; 7 8 9; 11 12 13]
```

- 同一行中元素之间用 空格 或 逗号 分隔 (建议优先使用逗号)
- 矩阵行与行之间用 分号 分隔
- 行之间的分号也可以用 回车 或 shift+回车 代替
- 也可创建复数矩阵 $A=[2, 3+2*i]$
- $a=(1,2)$ 是不对的，要用中括号

以下创建 **A1 A2 A3** 是相同的

```
A1=[1 2 3; 4,5,6; 7 8 9; 11 12 13]
```

```
A2=[1 2 3; 4,5,6; 7 8 9 这里回车  
    11, 12, 13]
```

```
A3=[1,2 3; 4,5,6; 7 8 9; 11 ...  
    12 13]
```

矩阵的创建

利用冒号：等差数列

A=1:2:10

%得到行向量 1 3 5 7 9, 注意这里没有中括号

B=[1:2:9]

%得到行向量 1 3 5 7 9, 也可以有中括号

C=(1:3)'

%得到列向量, 等价于C=[1;2;3]或C=[1:3]'

D=[1:3 , 2:4]

%得到行向量 1 2 3 2 3 4

E=[1:3 ; 2:4]

%得到 2x3 矩阵, 这里不能改成小括号

F=[2.4:-0.5:1]

%得到行向量 1.0 1.5 2.0

利用冒号：操作符创建行向量的基本语法格式：

x=Start : Increment : End

- 新向量x的最后一个元素不能大于End ;
- Increment可正可负，若负，则必须Start>End;
若正，则必须Start<End，否则创建的为空向量;
- 若Increment=1,则可简写为：x=Start:End;
- Increment可以是整数或实数。

矩阵的创建 一般方法

可用已有矩阵的全部、部分、或其运算式来创建新矩阵。

【例】 **a=[1, 3-1, 3]** %注意表达式3-1
 b=[a, 3, 4] % a作为组件参与矩阵b的创建
 c=[0, a, 3]
 d=[b+c ; a,5,6]
 e=b(1:2:5) % 提取矩阵b的一部分
 f=b([1 5]) % 提取b的第一,四元素

结果为

a =	1	2	3		
b =	1	2	3	3	4
c =	0	1	2	3	3
d =	1	3	5	6	7
	1	2	3	5	6
e =	1	3	4		
f =	1	4			

矩阵的创建 利用专门函数

linspace 函数（用于建立行向量，类似的还有 **logspace**）

x= linspace(x1, x2, n)

- 生成一个由 n 个元素 组成的行向量；
- x1为其第一个元素，x2为其最后一个元素；
- x1、x2之间元素的间隔为 $(x2-x1)/(n-1)$ ；
- 如果忽略参数n，则系统默认生成100个元素的行向量。

【例】 **x= linspace(2,3,3)**

将区间2等分，得到包含边界在内的3个数构成向量。

x= 2.0000 2.5000 3.0000

【例】 **y= logspace(2,3,3)**

同**linspace**类似，但将等分所得数值作为指数以10为底进行计算

y= 100 316.2278 1000

矩阵的创建

生成特殊矩阵

zeros(m,n)	生成 m 行 n 列的零矩阵, $m=n$ 时可简写为 <code>zeros(n)</code> <code>zeros(m1,m2,m3,...)</code> 可生成高维矩阵。
ones(m,n)	同zeros类似, 但元素都为1
eye(m,n)	生成主对角线全为 1 的 m 行 n 列矩阵, $m=n$ 时可简写为 <code>eye(n)</code> , 即为 n 维单位矩阵
diag(X)	若 X 是矩阵, 则 <code>diag(X)</code> 为 X 的主对角线向量 若 X 是向量, <code>diag(X)</code> 产生以 X 为主对角线的对角矩阵
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 间均匀分布的随机矩阵 $m=n$ 时简写为 <code>rand(n)</code>
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 <code>randn(n)</code>
其它特殊矩阵生成函数: magic 、 hilb 、 pascal ... 可用 help elmat 查看更多的特殊矩阵生成函数	

矩阵的创建 利用Excel表格文件

xlsread函数，读取Excel表格文件并形成矩阵

num = xlsread ('filename', 'sheet', 'range')

- filename 为文件名，如'myfile.xls'
- sheet为表格页名称，如'sheet2'，或者直接给表格页编号
- range为数据范围，如'C1:C5'，其中冒号是必须的
- 用 **lookfor xls** 查看关于Excel读、写的更多操作方法（xlsread, xlswrite 等）

矩阵的创建 特殊情况----空矩阵

- 有一维长度是0的矩阵即为空矩阵
- 空矩阵不占据存储空间，空矩阵不是全0元素矩阵
- 最简单的空矩阵：0 x 0的矩阵
- 复杂的空矩阵： 0 x 5 或 10 x 0矩阵

`a=[]; b=ones(0,5);`

察看空矩阵 `a, b % or whos a b`

结果 `a = []`

`b =` 空矩阵: 0×5

其他特殊情况还有 NaN矩阵等。

执行指令 `a = [1,2,3; 3,2,1];`
变量 `a` 同以下哪些指令的结果相同?

A

`a = [1,2,3, 3,2,1];`

B

`a = [1 2 3; 3 2 1];`

C

`a = 1:3; a=[a; 3,2,1];`

D

`a = 1:3; a=[a; 3:1];`

提交

矩阵的基本运算与操作

引用矩阵元素(标识并寻访)

□ 用全下标引用单个元素

$x(i)$: 向量 x 中的第 i 个元素

$A(i, j)$: 矩阵 A 中的第 i 行, 第 j 列元素

□ 多个元素的引用, 冒号的用法(本质上还是等差数列)

灵活运用冒号可提取矩阵的整行、整列或一部分

【例】 $A = [1, 2, 3, 4, 5; 10, 20, 30, 40, 50]$

$A(1, :)$ %引用第1行

$A(:, 1:3); A(:, [1\ 3])$ %这2个相同吗?

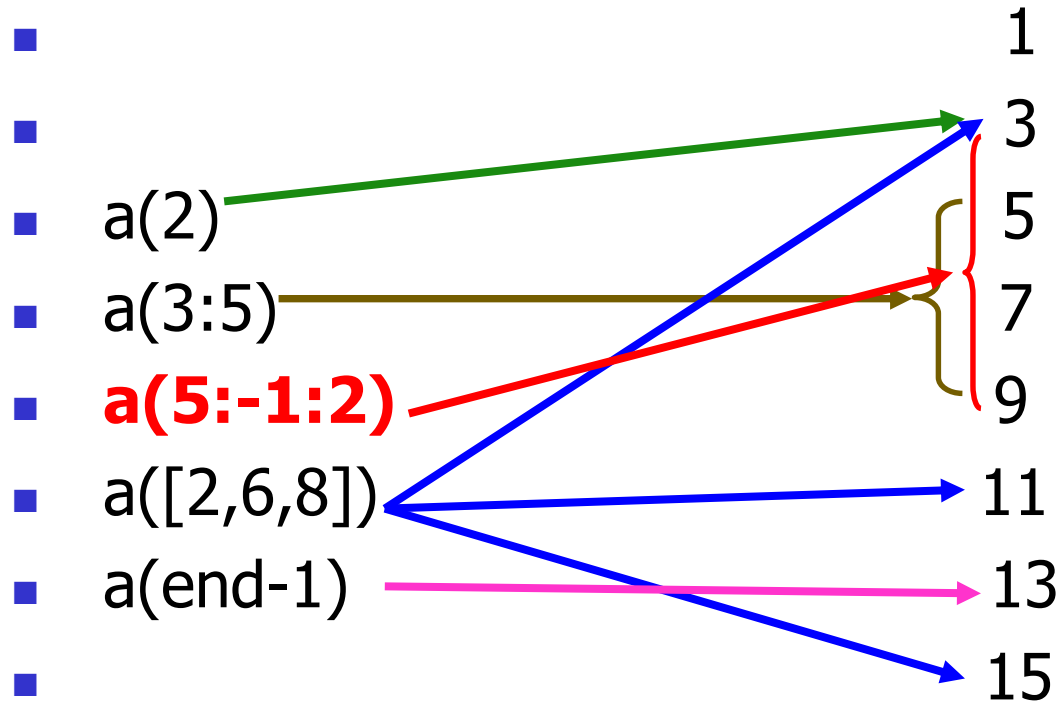
$A(:, :)$ %引用全部行和列

$A(1, 2:3)$ %引用第1行的2, 3列

$B = A(1:2, 1:2:5)$ %引用第1, 2行的第1, 3, 5列形成2×3矩阵²⁹

引用矩阵元素(标识并寻访)

【例】 $a = 1:2:16$



注意 $a(5:-1:2)$ 是9,7,5,3

$a([2,2,5])=?$ 是3,3,9, 元素可被多次重复引用

$a(2.1)$ 和 $a(10)$ 是错误的(超界)

矩阵操作 引用矩阵元素(标识并寻访)

【例】修改数组中一个或多个指定元素

$a=[1:5]$ % a 是 1,2,3,4,5

$a(3)=0$ % 修改数组 a 的第3元素值为0

$a([2, 5])=[1, 1]$ % a 再次变成1,1,0,4,1

- 可以修改指定数组元素的值
- 一次可以修改多个数组元素的值
- 要修改的数组元素的个数应与送入数组的元素个数相同

矩阵元素赋值

- 矩阵元素可以是数值表达式(也允许字符等其他类型)

```
x=[-1.3, sqrt(3), (1+2+3)*4/5]
```

```
y=['abc', '65'] %如果把'65'的两个引号去掉,会有什么结果
```

- 矩阵元素单独赋值,也能以此方法来扩展矩阵

```
x=[-1.3, sqrt(3), (1+2+3)*4/5];
```

```
x(5)=abs(x(1)) %系统自动将向量x的长度扩展到5,  
%并将未赋值部分(第4元素)置零
```

结果是 x =

```
-1.3000    1.7321    4.8000    0    1.3000
```

```
a=[1,2]; a(3,3)=3, b(2,2)=2 %什么结果
```


矩阵元素赋值

□ 大矩阵可以把小矩阵作为其片段，进行拼接

【例】 $x = [1 \quad 1 \quad 1]$
 $y = [11 \quad 12 \quad 13]$
 $z1 = [x, y]$ %z1 z2相同
 $z2 = [x \quad y]$
 $z3 = [x; y]$ %z3与z1不同
 $x = [x; 2, 2, 2]$ %通过拼接可以扩展自身

【例】 $a = [10, 20, 30], b = a(3:-1:2)$
 $c = [a(1:2:3); b]$

c的结果是2x2矩阵

10	30
30	20

单下标与全下标

□ 用单下标引用二维矩阵元素

Matlab的矩阵在内存中按列进行存储

对于二维数组，将二维数组的所有列，按先后顺序首尾相接排成“一长队”，然后自上往下对元素位置进行编号形成单下标，用来确定元素位置。

该方法也能用于高维矩阵，按照左侧下标优先变化的方式形成“一长队”并形成单下标。

【例】 $A=[10:10:30; 40,50,60]$

结果 $A=$

10	20	30
40	50	60

$A(3)=?$

10	40	20	50	30	60	一长队
(1,1)	(2,1)	(1,2)	(2,1)	(1,3)	(2,3)	

单下标与全下标

【例】单下标的使用

```
a = zeros(2, 5);
```

```
a(:) = -4:5
```

结果为

a =

-4 -2 0 2 4

-3 -1 1 3 5

注意数组的排列顺序。a(2)或a(2,1)都是 - 3

单下标与全下标

【例】

`a=zeros(2,4)` %创建2x4的全0数组

`a(:)=1:8` %对其赋值

`a([2 5 8])` %单下标寻访多个元素

`a([2 5 8])=[10 20 30]` %单下标修改多元素

`a(:,[2 3])=ones(2)` %全下标修改多元素

a为

1	3	5	7
2	4	6	8

a变为

1	3	20	7
10	4	6	30

a再变为

1	1	1	7
10	1	1	30

- 二维数组可以单下标或全下标方式访问、赋值；
- 单下标方式赋值时，等号两边涉及的元素个数必须相等；
- 全下标方式赋值时，等号右边数组的大小必须等于原数组中涉及元素构成的子数组的大小。

单下标与全下标

【例】高维情况

```
a=zeros(2,3,4)
```

```
a(10)=5    %相当于a(2,2,2)=5
```

```
a=zeros(2, 4)
```

```
a(:,end)
```

```
a(:,end-1)
```

```
a(:, end:-1:3)
```

```
a=zeros(2, 4)
```

```
a(end,:)
```

```
a(end,[2:4])
```

```
a ([4 6])=6:7
```

```
a(end,[2:end-1])
```

以上各命令行，标识了数组的哪些元素？

有关矩阵的一些常用函数

size、length 矩阵形状，最长一维的长度

```
a=ones(4,6)*6
```

```
m=size(a)    % 结果是4 6，表示形状是4行6列
```

```
n=length(a) % 结果是6
```

```
b=1:6
```

```
x=length(b)    % 结果是6
```

```
c=b'
```

```
y=length(c)    % 结果是6
```

有关矩阵的一些常用函数

sub2ind 全下标转换为单下标 (高维)

```
A = [17 24 1 8; 2 22 7 14; 4 6 13 20];
```

```
A(:,:,2) = A - 10  %在3*4二维矩阵基础上又扩了一维
```

```
A(2,1,2)
```

```
n=sub2ind(size(A),2,1,2)  % n=14
```

```
A(n)
```

A(:,:,1)					A(:,:,2)			
17	24	1	8		7	14	-9	-2
2	22	7	14	→	-8	12	-3	4
4	6	13	20		-6	-4	3	10

有关矩阵的一些常用函数

ind2sub 单下标转换为全下标

```
b = zeros(3);
```

```
b(:) = 10:10:90
```

```
IND = [3 4 5 6] %这里的数字将作为单下标
```

```
[I,J] = ind2sub(size(b),IND) %函数可有多多个结果
```

结果 b = 10 40 70
 20 50 80
 30 60 90

I =

3 1 2 3 %这里数字是行下标

J =

1 2 2 2 %这里数字是列下标

有关矩阵的一些常用函数

reshape 改变矩阵形状

a = -4:4

b = reshape(a, 3, 3)

结果为 b =

-4 -1 2

-3 0 3

-2 1 4

- 数组元素的排列顺序，**从上到下按列排列**，先排第一列，然后第二列，...
- 要求数组的**元素总数不变**

类似的还有**repmat**函数，可对矩阵进行重复拼接得到新矩阵，请查阅帮助文档。

矩阵基本运算

- 矩阵置空（用于删除某行或列，空和等于零不同）
 - `a = [1,2,3;4,5,6;7,8,9];`
 - `a(:,2) = [];` %将第2列删掉
- 矩阵元素的点运算 +加 -减 .*乘 ./除(还有反除.\) .^幂
- 矩阵比较和逻辑运算
- 矩阵乘法以及乘方 a^2
- 矩阵除法（逆） $A \backslash b = \text{inv}(A) * b$

矩阵基本运算

- 纯数量同矩阵的**点运算** **.*** **./** **.** **.^** **+** **-**

矩阵所有元素和数量进行运算，结果保持原矩阵形状

A=[1, 4, 8]

B=A.*3 %等价于B=[A(1)*3 ,A(2)*3 ,A(3)*3] 或3*A

C=A.^2 % 1 16 65, A^2是有问题的

D=A./10 % 0.1 0.4 0.8

E=A.\2 % 2 0.5 0.25

F=A+1

G=A-1 %不用 **.-** **.-**

矩阵基本运算

■ 两矩阵对应元素的点运算 $+$ $-$ $.*$ $./$ $.\backslash$ $.^$

A=[1, 4]

B=[1, 2]

A.*B %等价于 $[A(1)*B(1), A(2)*B(2)]$

%类似的还有

A./B, A.\B, A.^B, A+B, A-B

注意，可以对矩阵的一部分进行运算，参与运算的部分需形状一致，例如：

C=[4, 3, 1, 2]

A.*C(4:-2:1) %等价于 $[A(1)*C(4), A(2)*C(2)]$

矩阵基本运算

■ 比较运算(关系运算) < <= == > >= ~=

- 1、若关系成立时结果为1，否则为0。
- 2、当一个数量与一个矩阵比较时，该数量与矩阵的各元素进行比较，结果形成一个与矩阵维数相等的0、1矩阵。
- 3、当两个维数相等的矩阵进行比较时，其相应位置的元素按数量关系进行比较，并给出结果，形成一个维数与原来相同的0、1矩阵。

A=[-1, 4], B=[-1, 2]

C=[4, 3, 1, 2, 5]

D1=A>=B

D2=A~=B

D3=A<2 %也可以和纯数量进行运算

矩阵基本运算

【例】建立5阶方阵A，判断其元素能否被3整除。

A = [24, 35, 13, 22, 63; 23, 39, 47, 80, 80; ...

90, 41, 80, 29, 10; 45, 57, 85, 62, 21; 37, 19, 31, 88, 76]

A =

24 35 13 22 63

23 39 47 80 80

90 41 80 29 10

45 57 85 62 21

37 19 31 88 76

P = rem(A,3) == 0 %rem用来求余数

P =

1 0 0 0 1

0 1 0 0 0

1 0 0 0 0

1 1 0 0 1

0 0 0 0 0

矩阵基本运算

■ 矩阵逻辑运算 **&**与 **|**或 **~**非 **XOR**异或

- 1、在逻辑运算中，非零元素为真 1，否则为假 0
- 2、算术运算优先级最高，逻辑运算优先级最低
- 3、对任一矩阵都可以进行 **~** 非运算

【例】

a = [-2; -1; 0; 0; 1; 2]

b = [-2; 1; -1; 0; 0; 2]

x = ~a **%[0; 0; 1; 1; 0; 0]**

y = ~b **%[0; 0; 0; 1; 1; 0]**

z = a&b **%[1; 1; 0; 0; 0; 1]**

矩阵基本运算

■ 矩阵逻辑运算 $\&$ 与 $|$ 或 \sim 非 **XOR** 异或

【例】

$A=[2, 2], B=[1, 0], C=[2, -1]$

$D1= A\&B, D2= A|B, D3=A\&C, D4=\sim C, D5=B\&5$

结果为

$A = \quad 2 \quad 2$

$B = \quad 1 \quad 0$

$C = \quad 2 \quad -1$

$D1 = \quad 1 \quad 0$

$D2 = \quad 1 \quad 1$

$D3 = \quad 1 \quad 1$

$D4 = \quad 0 \quad 0$

$D5 = \quad ???$

1 0

矩阵基本运算

MATLAB 运算符的优先次序

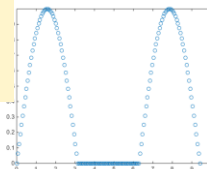
优先次序	运 算 符
最 高	'(转置共轭)、^(矩阵乘幂)、!(转置)、.^(数组乘幂)
	~(逻辑非)
	、/(右除)、\ (左除)、.(数组乘)、./(数组右除)、.\(数组左除)
	+, -
	:(冒号运算)
	<, <=, >, >=, ==(恒等于), ~= (不等于)
	&(逻辑与)
	(逻辑或)
	&&(先决与)
最 低	(先决或)

矩阵基本运算

【例】在 $[0, 3\pi]$ 区间，求 $y=\sin(x)$ 的值。要求消去负半波，即 $(\pi, 2\pi)$ 区间内的函数值置零。

```
x = 0:pi/50:3*pi; %产生一系列x坐标  
y = sin(x);        %计算一系列y坐标  
y1 = (y>=0).*y;    %y消去负半波，注意运算优先级，如果这  
                    %里的括号（）删掉，将是什么结果？  
  
plot(x,y1,'o')
```

Matlab的许多内置函数例如sin输入参数不仅可以是数量，还允许使用矩阵，相当于将矩阵元素分别进行sin运算，计算结果保持原矩阵形状。



矩阵基本运算

【例】建立矩阵A，找出在[10, 20]闭区间的元素的位置。

A = [4,15,-45,10,6; 56,0,17,-45,0]

结果是 A =

4	15	-45	10	6
56	0	17	-45	0

%得到符合条件元素的单下标

%元素15,17,10分别对应单下标 3 6 7

find(A>=10 & A<=20)

结果是 ans =

3
6
7

%得到符合条件元素的全下标

% x,y 分别记录行与列下标

[x,y]=find(A>=10 & A<=20)

结果是 x = y =

1	2
2	3
1	4

关于矩阵乘和反除

- 矩阵乘法 $*$
- 矩阵除法 \backslash (实际上是反除 不是 $./$ **matlab**特有)

$M1 \times N$ 矩阵 $*$ $N \times M2$ 矩阵 = $M1 \times M2$ 矩阵

回顾线性代数，复习矩阵相乘的法则

特殊的: $1 \times N * N \times 1 = 1 \times 1$ (数量)

特殊的: $A \backslash b = \text{inv}(A) * b$ $A \backslash b$ 同 b/A 并不等价

$N \times N \backslash N \times 1$

矩阵除法可用于解线性方程组

$$A * x = b \quad \rightarrow \quad x = \text{inv}(A) * A * x = \text{inv}(A) * b = A \backslash b$$

关于矩阵乘和反除

【例】

`a=[1,2; 3,4]`

`b=[1; 2]`

`a*b`

`a\b`

`b*a` %不能运算，why?

`b/a` %也不能运算，why? 与`a\b`并不等价

原因是矩阵维度不满足要求

有变量 $a = [1, 2, 3; 4, 5, 6]$;
以下哪些指令能将其中的3与4变为0,

- ☒ **A** $a(1,3)=0; a(2,1)=0$
- ☒ **B** $a(2)=0; a(5)=0$
- ☒ **C** $a([2\ 5])=0$
- ☒ **D** $a=a-a.*(a==3|a==4)$

矩阵分析常用函数

最大值和最小值 **max**, **min**

1、求向量的最大值和最小值 有两种调用格式，分别是：

(1) $y=\max(X)$ ：返回向量 X 的最大值存入 y ，如果 X 中包含复数元素，则按模取最大值；

(2) $[y,I]=\max(X)$ ：返回向量 X 的最大值存入 y ，最大值的序号存入 I ，如果 X 中包含复数元素，则按模取最大值。

求向量 X 的最小值的函数是 $\min(X)$ ，用法和 $\max(X)$ 完全相同。

矩阵分析常用函数

最大值和最小值 **max, min**

2. 求矩阵的最大值和最小值

求矩阵A的最大值的函数有3种调用格式，分别是：

- (1) **max(A)**: 返回一个行向量，向量的第i个元素是矩阵A的第i列上的最大值；
- (2) **[Y,U]=max(A)**: 返回行向量Y和U，Y向量记录A的每列的最大值，U向量记录每列最大值的行号；
- (3) **max(A,[],dim)**: dim取1或2。dim取1时，该函数和max(A)完全相同；dim取2时，该函数返回一个列向量，其第i个元素是A矩阵的第i行上的最大值。

矩阵分析常用函数

【例】 求矩阵的最大值

```
x=[-43,72,9; 16,23,47];
```

```
y=max(x)    %求矩阵x中每列的最大值
```

```
y =
```

```
    16    72    47
```

```
[y,l]=max(x) %求矩阵x中每列的最大值及其该元素的位置
```

```
y =
```

```
    16    72    47
```

```
l =
```

```
     2     1     2
```

```
max(x, [],1) %求矩阵中每列的最大值， 等同于max(x)
```

```
max(x, [],2) %求矩阵中每行的最大值
```

矩阵分析常用函数

sum(X): 返回向量X各元素的和。

prod(X): 返回向量X各元素的乘积。

sum(A): 返回一个行向量，其第i个元素是A的第i列的元素和。

prod(A): 返回一个行向量，其第i个元素是A的第i列的元素乘积。

sum(A,dim): 当dim为1时，该函数等同于sum(A)；当dim为2时，返回一个列向量，其第i个元素是A的第i行的各元素之和。

prod(A,dim): 当dim为1时，该函数等同于prod(A)；当dim为2时，返回一个列向量，其第i个元素是A的第i行的各元素乘积。

矩阵分析常用函数

mean(X) 返回向量X的算术平均值。

median(X) 返回向量X的中值。

mean(A) 返回一个行向量，第i个元素是A的第i列的算术平均值。

median(A) 返回一个行向量，第i个元素是A的第i列的中值。

mean(A,dim) 当dim为1时，等同于mean(A)；当dim为2时，返回一个列向量，其第i个元素是A的第i行的算术平均值。

median(A,dim) 当dim为1时，等同于median(A)；当dim为2时，返回一个列向量，其第i个元素是A的第i行的中值。

关于max sum mean等函数的一些小技巧

【例】 `x=[1,2,3; 4,5,6];`

`max(x)` %按列求最大元素

`ans = 4 5 6`

`max(max(x))` %两层调用max得到整个矩阵的最大元素

`ans = 6`

`sum(x)`

`ans = 5 7 9`

`sum(sum(x))`

`ans = 21`

`mean(x)`

`ans = 2.5 3.5 4.5`

`mean(mean(x))`

`ans = 3.5`

矩阵分析常用函数

help matfun 列出矩阵分析的常用函数（很多）

Matrix functions - numerical linear algebra.

-
- Matrix analysis.
- norm - Matrix or vector norm.
- normest - Estimate the matrix 2-norm.
- rank - Matrix rank.
- det - Determinant
- trace - Sum of diagonal elements.
- null - Null space.
- orth - Orthogonalization.
- rref - Reduced row echelon form.
- subspace - Angle between two subspaces.

矩阵分析常用函数

Linear equations.

- **inv** - **Matrix inverse**
- rcond - LAPACK reciprocal condition estimator
- cond - Condition number with respect to inversion.
- condest - 1-norm condition number estimate.
- normest1 - 1-norm estimate.
- chol - Cholesky factorization.
- cholinc - Incomplete Cholesky factorization.
- **lu** - **LU factorization.**
- luinc - Incomplete LU factorization.
- qr - Orthogonal-triangular decomposition.
- lsqnonneg - Linear least squares with nonnegativity constraints.
- pinv - Pseudoinverse.
- lscov - Least squares with known covariance.

矩阵分析常用函数

Eigenvalues and singular values.

- `eig` - Eigenvalues and eigenvectors.
- `svd` - Singular value decomposition.
- `gsvd` - Generalized singular value decomposition.
- `eigs` - A few eigenvalues.
- `svds` - A few singular values.
- `poly` - Characteristic polynomial.
- `polyeig` - Polynomial eigenvalue problem.
- `condeig` - Condition number with respect to eigenvalues.
- `hess` - Hessenberg form.
- `qz` - QZ factorization for generalized eigenvalues.
- `schur` - Schur decomposition.

矩阵分析常用函数

Matrix functions.

- expm - Matrix exponential.
- logm - Matrix logarithm.
- sqrtm - Matrix square root.
- funm - Evaluate general matrix function.

Factorization utilities

- qrdelete - Delete a column or row from QR factorization.
- qrinsert - Insert a column or row into QR factorization.
- rsf2csf - Real block diagonal form to complex diagonal form.
- cdf2rdf - Complex diagonal form to real block diagonal form.
- balance - Diagonal scaling to improve eigenvalue accuracy.
- planerot - Givens plane rotation.
- cholupdate - rank 1 update to Cholesky factorization.
- qrupdate - rank 1 update to QR factorization.

本节结束
谢谢