

MATLAB 语言及应用

MATLAB Language and Its Applications

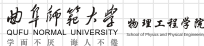
## 第二章 矩阵和数组

### Ch.2 Matrices and Arrays

刘世东

Shidong Liu

School of Physics and Physical Engineering  
Qufu Normal University



May 24, 2020

# 本章目录

## 1. 数据类型

### 1.1 数值类型

### 1.2 逻辑类型

### 1.3 字符与字符数组

### 1.4 函数句柄

### 1.5 结构体类型

### 1.6 数组类型

### 1.7 胞元数组

## 2. 矩阵和数组的概念

## 3. 数组的创建与寻访

### 3.1 一维向量的创建

### 3.2 矩阵的创建

## 4. 数组元素的标志与寻访

### 4.1 数组大小, 维度, 元素数

### 4.2 数组的标志与寻访

## 5. 数组的操作

### 5.1 数组的扩充与收缩

### 5.2 数组的其他常用操作

## 6. 数组的运算

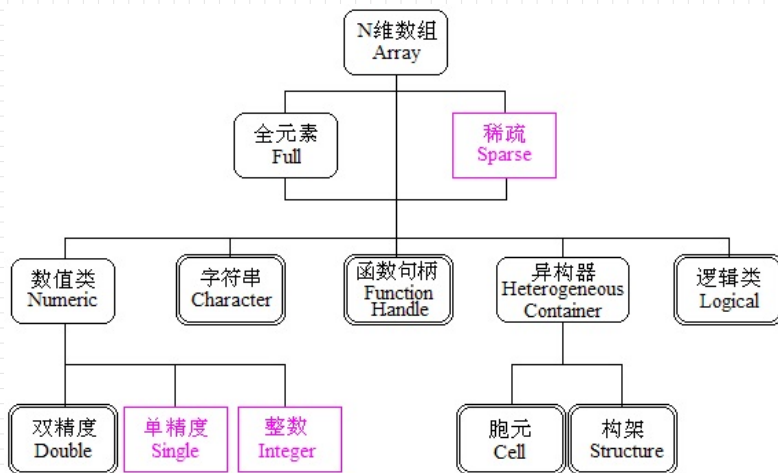
### 6.1 数组的运算规则

## 7. 矩阵的运算

### 7.1 矩阵的运算规则

### 7.2 矩阵的其他常规操作

# MATLAB 将所有的数据看作数组 Array



# MATLAB 内的数据类型

MATLAB 中的主要数据类型可以通过其帮助文档查看: `doc Data Types`

我们主要了解:

数值类型,  
逻辑类型,  
字符与字符串,  
函数句柄,  
结构体类型,  
胞元数组.

# 数值类型

基本的数值类型主要有整数、单精度浮点数和双精度浮点数 3 类:

数据格式	示 例	说 明
int8, uint8 int16, uint16 int32, uint32 int64, uint64	int32(820)	有符号和无符号的整数类型 相同数值的整数类型占用比浮点类型更少的内存 除了 int64 和 uint64 类型外的所有整数类型，都可以进行数学运算
single	single(128.1)	单精度浮点类型 相同数值的单精度浮点类型比双精度浮点类型占用更少内存 精度与能够表示的数值范围比双精度浮点类型小
double	333.77 1.000-1.000i	双精度浮点类型，MATLAB 中默认的数值类型

在未加说明与特殊定义时，MATLAB 对所有数值按照双精度浮点数类型进行存储和操作。

我们对此不再赘述，多年使用经验，几乎没有遇到过数据类型转换。

## 逻辑类型

除了传统的数学运算之外，MATLAB 还支持关系和逻辑运算。逻辑类型的数据是指布尔类型的数据及数据之间的逻辑关系。

关系	含义	逻辑类型	逻辑	含义
<	小于			
<=	小于或等于		&	与
>	大于			或
>=	大于或等于		~	非
==	等于			
~=	不等于			

MATLAB 提供短逻辑关系：**与** &&, **或** || 以更快实现逻辑运算。

## 字符与字符数组

**MATLAB 将文本当作字符数组**(有些教材也叫字符串)。字符数组能够显示在屏幕上，也可以用来构成一些命令，这些命令在其他的命令中用于求值或者被执行。

一个字符数组是存储在一个**行向量中的文本**，这个行向量中的每一个元素代表一个字符，**空格也是一个字符**。

实际上，**元素中存放的是字符的内部代码**，即 ASCII 码。当在屏幕上显示字符变量的值时，显示出来的是文本，而不是 ASCII 数字。我们可以通过它的下标对字符串中的任何一个元素进行访问。

```
Str = 'Every good boy does fun.'; %字符数组的创建用/单引号/, 切切  
size(Str) % 结果 1 24
```

注意, 新版本中对字符串 (数组) 和字符串数组有不同的定义...(略)

```
c = 'AB'; %字符串AB  
c2n = abs(c) % abs将字符转换为ASCII码, 也可以用double  
% 运行结果  
c2n =  
65    66
```

## 字符串 (数组) 与字符串数组的异同

MORE: docsearch Characters and Strings. 以下来自 MATLAB 帮助文档

Character arrays and string arrays provide storage for text data in MATLAB.

A **character array** is a sequence of characters, just as a numeric array is a sequence of numbers. A typical use is to store **short** pieces of text as character vectors, such as `c = 'Hello World'`;

A **string array** is a container for pieces of text. String arrays provide a set of functions for working with text as data (*Starting from R2016b*). To convert text to string arrays, use the `string` function. MATLAB displays strings in string arrays with double quotes.



# 函数句柄

在 MATLAB 中，对函数的调用方法分为直接调用法和间接调用。

直接调用函数，被调用的函数通常被称为子函数。但是子函数只能被与其 M 文件同名的主函数或在 M 文件中的其他函数所调用，同时在一个文件中只能有一个主函数。

使用函数句柄对函数进行调用则避免了上述问题。函数句柄实际上提供了一种间接调用函数的方法。创建函数句柄需要用到操作符 @。对 MATLAB 库函数中提供的各种 M 文件中的函数和自己编写的程序中的内部函数，都可以创建函数句柄。

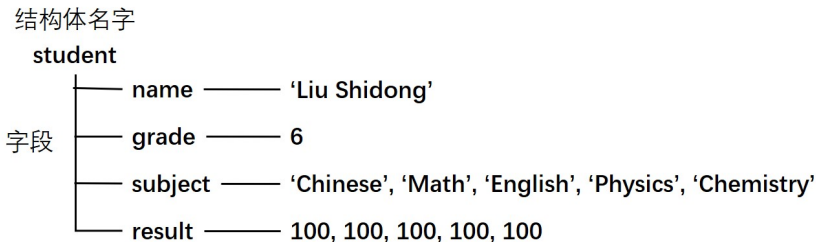
创建函数句柄的一般句法格式为：`Funh = @Funname;`

Funh 变量保存了这一函数句柄，并在后续的运算中作为数据流进行传递。

例如，`FHandle=@cos` 创建了 MATLAB 内部函数 cos 的句柄，并将其保存在 FHandle 变量中，之后就可以通过 `FHandle(x)` 来实现 `cos(x)` 的功能。

## 结构体类型

MATLAB 中的结构体<sup>1</sup>与 C 语言中的结构体类似，一个结构体可以通过字段存储多个不同类型的数据。因此，结构体相当于一个数据容器，把多个相关联的不同类型的数据封装在一个结构体对象中。



创建结构体对象的方法有两种，

- ① 直接通过赋值语句给结构体的字段赋值
- ② 使用结构体创建函数 `struct`

M 语法: `s = struct(field1,value1, ... ,fieldN,valueN)`

<sup>1</sup>有的教材也称之为构架数组

# 创建结构体

## 一. 直接赋值

```
Student.name = 'Liu Shiong';
Student.grade = 6;
Student.subject = {'Chinese', 'Math', 'English', 'Physics', 'Chemistry'};
Student.result = {100,100,100,100,100}; % 胞元数组
Student % 运行结果如下:
Student =
struct with fields:
name: 'Sam'
grade: 6
subject: {'Chinese' 'Math' 'English' 'Physics' 'Chemistry'}
result: {[100] [100] [100] [100] [100]}
```

运行 `whos` 查看 student 的信息

```
whos
Name          Size          Bytes   Class      Attributes
Student       1x1             1946    struct
```

## 二. struct 函数创建

% 创建日程结构体

```
Schedule(2)=struct('Day','Thursday','Time','15:00','Number',18)
```

Schedule(2) 表明创建了一个结构体组.

命令窗口内运行

```
Schedule(1)
```

%运行结果, 返回空结构体

```
ans =
```

```
struct with fields:
```

```
Day: []
```

```
Time: []
```

```
Number: []
```

# 数组类型

在 MATLAB 中进行运算的所有数据类型，都是按照数组及矩阵的形式进行存储和运算的，而二者在 MATLAB 中的基本运算性质不同，数组强调元素对元素的运算，而矩阵则采用线性代数的运算方式。关于它们的运算等后面详说。

数组的属性及数组之间的逻辑关系，是编写程序时非常重要的两个方面。在 MATLAB 平台上，数组的定义是广义的，数组的元素可以是任意的数据类型，例如可以是数值、字符串等。

例如数值数组 `a = [1 2 3 3];`

我们在本门课程中主要涉及的是数值数组。

# 胞元数组

胞元 (Cell) 数组<sup>2</sup>是一种无所不包的广义矩阵。组成单元数组的每一个元素称为一个单元。每一个单元可以包括一个任意数组，如数值数组、字符串数组、结构体数组或另外一个单元数组，因而每一个单元可以具有不同的尺寸和内存占用空间。

MATLAB 中使用单元数组的目的在于，它可以把不同类型的数据归并到一个数组中。

如何创建请参看 `doc cell`。

---

<sup>2</sup>有的教材称之单元数组

## 矩阵和数组的概念

MATLAB is an abbreviation for "matrix laboratory." While other programming languages mostly work with numbers one at a time, MATLAB is designed to operate primarily on whole matrices and arrays.

- 一般数组的第一维称之为行, 第二维称之为列, 第三维通常称之为页 (一般是这样叫, 随你).
- MATLAB 中将二维数组称之为矩阵.
- 一维数组指行数或者列数为 1 数组, 有时称之为向量;
- 矩阵指行数和列数都大于 1 的数组.

矩阵是数学概念, 数组是程序概念.

数组可分为数值数组、字符数组、胞元数组、结构数组等各种类别。

# 一维向量的创建

MATLAB 中向量是用方括号表征的. 实际应用中有很多种创建方法.

**一. 直接输入法:** 一般用于没有明显特征规律的向量创建. 如

## ① 行向量

```
a = [0, 0.11, 0.45, 0.89, 0.92]
```

```
a = [0 0.11 0.45 0.89 0.92]
```

每个数组元素之间, 可以用逗号或空格分隔 (MATLAB 没有规定或建议用哪种, 随你, 多么友好, 多么宽松...).

## ② 列向量

```
a = [0; 0.11; 0.45; 0.89; 0.92]
```

列向量的创建用分号间隔 (也可以用换行, 但是不推荐 (这是我写的)).

NOTE: `a = []` 表示一个空矩阵. (`isempty(a)` 返回真.)

⋮

(续) ...



## 二. 冒号生成法: 等步长递增或者递减的数列 (是等差数列), 生成的是行向量.

① M 语句格式: `x = a:inc:b`

a 表示数组第一个元素, b 是最后一个元素<sup>3</sup>.

inc 是步长, 可正可负 (缺省是默认为 1): 正表示递增 (要求  $b > a$ ), 负表示递减 (要求  $a > b$ )  
方括号可以省略.

② `x=0:0.2:1.2`  $\rightarrow$  0 0.2000 0.4000 0.6000 0.8000 1.0000 1.2000

③ `x=0:-0.2:-1.2`  $\rightarrow$  0 -0.2000 -0.4000 -0.6000 -0.8000 -1.0000 -1.2000

④ `x=2:8`  $\rightarrow$  2 3 4 5 6 7 8

⑤ `x=2.1:8`  $\rightarrow$  2.1000 3.1000 4.1000 5.1000 6.1000 7.1000

生成的行向量第一个元素一定是 a, 但是最后一个元素不一定是 b.

将行向量转置即可变成列向量, 转置运算是 `.'` 或者函数 `transpose`.<sup>4</sup>

<sup>3</sup>并不一定...

<sup>4</sup>我们后面会讲<sup>1</sup> 或者函数 `ctranspose` 的区别.

**三. 函数生成法:** 生成特定规律的数组. 常用的函数为 `linspace` 和 `logspace`, 可以通过 doc 查看函数并练习用法.

#### ① M 语言格式

`x = linspace(a,b,n)` 生成以  $a$ ,  $b$  为初末元素的  $n$  个等间隔数组. ( $n$  缺省默认为 100)

`x = logspace(a,b,n)` 生成以  $10^a$  和  $10^b$  为初末元素的  $n$  个等对数间隔数组 ( $n$  缺省默认为 50).

② `x=linspace(-5,5,7)` → -5.0000 -3.3333 -1.6667 0 1.6667 3.3333 5.0000

③ `x=logspace(1,5,7)` →  $1.0e+05 \times (0.0001 \ 0.0005 \ 0.0022 \ 0.0100 \ 0.0464 \ 0.2154 \ 1.0000)$

NOTE: `logspace` 还有一个特殊的用法: `y = logspace(a,pi,n)` 生成以  $10^a$  和  $\pi$  为初末元素的等对数间隔数组, 数字信号处理中用的较多.

还有很多其他函数, 在高维数组的创建中讲.....

# 矩阵的创建

## 一. 直接输入法: 小规模矩阵

- ① 元素用方括号包围.
- ② 行与行之间用分号间隔<sup>5</sup>.
- ③ 每一行的元素之间用逗号或者空格.

```
x=[0.8 0.9 0.2 0.9; 0.9 0.6 0.9 0.1; 0.1 0 0.5 0.7]
```

线性代数中常用于系数矩阵生成...

---

<sup>5</sup>建议不要用 enter 键

# 矩阵的创建

二. 函数生成法: 特性规律的大规模矩阵. MATLAB 中有很多这样的函数, 常用的有

① eye: 单位矩阵 `doc eye` 查看帮助

`eye` 产生 1.

`eye(n)`  $n \times n$  的单位矩阵.

② ones: 全 1 矩阵 `doc ones` 帮助

`ones(n,m)`  $n \times m$  的全 1 矩阵.

③ zeros: 全零矩阵同上

④ rand: 0 和 1 之间的随机矩阵 `doc rand` 查看

帮助, 注意 see also

`rand` 产生一个随机数

`rand(n,m)` 产生  $n \times m$  随机矩阵.

`ones`和`zeros`常用于预分配内存.

$$e = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$x = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}$$

同理全零...

## 高维数组的创建

一般用函数生成, 如 `rand(3,2,5)`, 也可以用下标生成.

MATLAB 显示三维以上的数组, 不直观...

本课程中不涉及, 从略...

## 数组的大小

数组的大小 (size) 数组的行数和列数的数值大小, 用命令 `size` 可以查看数组大小. 如

### 数组的大小 `size`

```
x = rand(2,3)
[hangShu, lieShu] = size(x)
y = rand(3,6,2)
[H,L,Y] = size(y)
```

还有一个函数 `length`, 可以查看数组的大小, 实际给出的是数组的行数和列数的最大值. 对于一维数组或向量, 给出的是向量的长度, 是一个非常有用的函数. 如

`length(x)` 给出的结果是 3; `length(y)` 给出的是 6.

对于数组 A, `length` 和 `size` 函数的联系是  $\text{length}(A) = \max(\text{size}(A))$ .

`doc size` 和 `length` 查看函数帮助文档.

## 数组的维数

数组的维数是指数组有几维. 可以用 `ndims` 函数查看. 矩阵的维数为 2.

标量和向量的维数也都是 2. 标量相当于  $1 \times 1$  的矩阵, 向量相当于  $1 \times n$  或者  $n \times 1$  的矩阵.

### 矩阵的维数

```
ab = 42;  
nab = ndims(ab)  
nx = ndims(x)  
ny = ndims(y) % 接上一页的x和y
```

结果分别是 2, 2, 3.

对于一个数组 A, `ndims(A) = length(size(A))` `doc ndims` 查看帮助文档.

## 数组元素的个数

数组内元素的个数, 用 `numel` 查看, 实际上就是行数和列数的乘积, 即 `numel(A) = prod(size(A))`.<sup>6</sup>

`doc numel` 查看帮助文档.

---

<sup>6</sup>`prod` 表示 (连) 乘积



## 数组的标志与寻访方法

对于任意矩阵  $A$ , 其元素的标志方法主要有三种

- ① 全下标法: 格式  $A(r,c)$ , 表示  $A$  矩阵第  $r$  行第  $c$  列的元素.
- ② 单下标法: 格式  $A(s)$ , 表示  $A$  矩阵第  $s$  个元素, 多用于向量.
- ③ 逻辑标志: 格式  $A(L)$ , 给出  $A$  矩阵中逻辑数列  $L$  中为真的元素, 一般用于向量.

魔方矩阵  $A$

$$\begin{pmatrix} 35 & 1 & 6 & 26 & 19 & 24 \\ 3 & 32 & 7 & 21 & 23 & 25 \\ 31 & 9 & 2 & 22 & 27 & 20 \\ 8 & 28 & 33 & 17 & 10 & 15 \\ 30 & 5 & 34 & 12 & 14 & 16 \\ 4 & 36 & 29 & 13 & 18 & 11 \end{pmatrix}$$

### 单元素寻访示例

```
A = magic(6) % magic returns an n-by-n ...
               matrix constructed from the integers ...
               1 through n^2 with equal row and ...
               column sums. The sum is (1+2+3+n^2)/n
A23 = A(2,3) % return 7
As8 = A(8) % return 32
```

对于逻辑寻访, 须先生成逻辑数组, 再进行寻访, 通常在条件判断环境中使用..... 后讲...

除了可以单元素寻访, MATLAB 可以同时提取多个元素, 只要将  $r$ ,  $c$ ,  $s$  变成数组即可. 例如

- ① 提取 A 中的第 2 行中, 第 3,4,5 列的元素

`A(2,[3 4 5])` 或者 `A(2,3:5)`

- ② 提取 A 中任意第  $r$  行或者第  $c$  列的元素

`A(3,:)`

`A(:,c)`

- ③ 最后一个元素, 利用 `end`, `end` 表示数组中的最有一个元素的标志

`A(end)`

= 提取 A 的元素时, 可以将所提取的元素复制给其他变量, 例如

`Bs = A(2,3)`      `Bm = A(2,[1 4 6])`

= 可以给 A 的任意元素重新赋值, 如

`A(2,3) = 100`

`A(:)` 表示什么意思???

除了可以单元素寻访, MATLAB 可以同时提取多个元素, 只要将  $r$ ,  $c$ ,  $s$  变成数组即可. 例如

- ① 提取  $A$  中的第 2 行中, 第 3,4,5 列的元素

`A(2,[3 4 5])` 或者 `A(2,3:5)`

- ② 提取  $A$  中任意第  $r$  行或者第  $c$  列的元素

`A(3,:)`

`A(:,c)`

- ③ 最后一个元素, 利用 `end`, `end` 表示数组中的最有一个元素的标志

`A(end)`

= 提取  $A$  的元素时, 可以将所提取的元素复制给其他变量, 例如

`Bs = A(2,3)`      `Bm = A(2,[1 4 6])`

= 可以给  $A$  的任意元素重新赋值, 如

`A(2,3) = 100`

`A(:)` 表示什么意思???

提取  $A$  的全部元素, 同时将矩阵  $A$  按元素 (单) 标志顺序变成列向量.

## 数组的扩充与收缩

一般只涉及向量的扩充和收缩. 扩充和收缩有两层含义:

- ① 维度: 3 维变成 2 维
- ② 大小:  $1 \times 3$  数组变成  $1 \times 7$  数组.

创建数组: `a=1:5; b=6:10; c=11:15;`

### 数组的扩充

```
a_b=[a,b] %horizontally concatenates arrays a and b. horzcat(a,b)
abc=[a;b;c] % vertically concatenates arrays a, b, and c, vertcat(a,b,c)
```

a_b =	1	2	3	4	5	6	7	8	9	10
-------	---	---	---	---	---	---	---	---	---	----

abc =					
1	2	3	4	5	
6	7	8	9	10	
11	12	13	14	15	

## 数组的收缩: 空数组 `[]`

```
a_b(2:5)=[] %delete the 2nd to 5th elements of array, a_b
abc(:,3)=[] %delete all the elements of the 3rd column of array, abc
```

a_b =	1	6	7	8	9	10
-------	---	---	---	---	---	----

abc =				
1	2	4	5	
6	7	9	10	
11	12	14	15	

## `repmat`函数: 扩充块数组

```
A=repmat(a,[2,1])
B=repmat(b,[1,2])
```

A =				
1	2	3	4	5
1	2	3	4	5

B =						
6	7	8	9	10	6	7 ...
	8	9	10			

## 数组常用的操作函数

MATLAB 内置了很多函数对数组进行操作, 如变形, 排序, 旋转, 最值, 平均值等. 对于实际特定情况, 往往还需要通过元素寻访达到预定要求.

### 一. 排序 `sort` 函数升序排列: 多用于向量排序.

```
A = [9 0 -7 5 3 8 -10 4 2];
```

```
B = sort(A) % ascending order, 结果如下
```

```
B = -10    -7         0         2         3         4         5         8         9
```

```
B = sort(A, 'descend') % descending order, 结果如下
```

```
B = 9         8         5         4         3         2         0        -7       -10
```

NOTE: 若将 `sort` 函数用于矩阵, 则默认按列进行排列<sup>7</sup>

```
A = [3 6 5; 7 -2 4; 1 0 -9]
```

```
B = sort(A) % try B = sort(A,2)
```

See also `issorted`

<sup>7</sup>这里需要指出, MATLAB 的函数是列优先处理的...

## 二. 最值, 求和, 平均值等

`A = 1:9;`

① 最大值: `max(A)`

② 最小值: `min(A)`

③ 求和: `sum(A)`

④ 平均值: `mean(A)`

⑤ 中值: `median(A)`

⑥ 标准差: `std(A)`

⑦ 连乘: `prod(A)`

## 三. 变形操作 `reshape`

M 语句格式 `reshape(A, r, c)`, 要求  $r \times c$  等于 A 的元素总数

`A = 1:24`

```
B = reshape(A,4,6)
```

B =

1	5	9	13	17	21
2	6	10	14	18	22
3	7	11	15	19	23
4	8	12	16	20	24

```
% try B = reshape(A,3,8)
```

---

方差计算公式  $\sigma = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (A_i - \mu)^2}$

---

旋转操作 `rot90`, `flip` and see also.

## 自建函数产生特殊要求的数组

对于实际问题, MATLAB 内置的函数往往不能满足我们要求的数组 (向量或者矩阵). 此时, 只能通过自建函数 (一般不会手动输入, 因为自建函数可以满足一类要求, 手动只能满足特定要求).

**一. 克拉兹数列:** 第一个元素  $n$  若为奇数, 则下一个元素为  $3n+1$ , 否则为  $n/2$ , 依次类推, 直到最后的元素 1 (若无法达到 1 怎么办? 目前的认知, 没有这种可能:-)).

例如, 对于 3, 则数列为 [3 10 5 16 8 4 2 1].

**二. 折线矩阵:** 矩阵为  $n$  阶方阵, 元素为 1 到  $n^2$  之间的整数, 元素排列要求按行成 Z 自行排列.

例如对于  $n=4$ , 则折线矩阵为

**三. blabla ... :** 放到实验里... :-)

1	2	3	4
8	7	6	5
9	10	11	12
16	15	14	13



# 接上一页幻灯片给出程序代码

## 一. 克拉兹数列

```
function clzS = collatz(n)
clzS=n;
while n ~=1
if mod(n,2)==0
n=n/2;
else
n=3*n+1;
end
clzS=[clzS ,n];
end
end
```

## 二. 折线矩阵

```
function b = zmat(n)
b=transpose(reshape(1:n^2,n,n));
b(2:2:n,:)=fliplr(b(2:2:n,:));
%for k=2:2:n
%b(k,:)=sort(b(k,:), 'descend');
%end
end
```

首要问题是要把问题弄清楚, 然后想算法, 最后写代码. 算法部分: 如果 MATLAB 有相关函数且知道, 或者有能力利用网络查到, 此算法代码会简洁, 如果 MATLAB 没有相关函数或者有但是不知道, 则算法代码需要自己利用 M 语言完成.

算法千万个...

## 数组的运算规则

数组的运算规则是针对数组元素定义的运算.

- ① 标量 b-数组 s 运算: b 与 s 的每个元素之间进行运算.
- ② 数组 s1-数组 s2 运算: 要求 s1 和 s2 必须同为同大小, 它们之间的运算时相同位置的元素之间的运算.
- ③ 函数-数组运算: 函数对数组的逐个元素进行作用, 函数范围的值个数与数组元素个数相等..

MATLAB 中, 数组运算符加减乘除和乘方需要加上一个英文点, 即

数学符号	+	-	×	÷ 或 /	上标或^
M 语言	+	-	.*	./ 或 .\	.^

注意: 左边的**黑点**; MATLAB 中定义了左除...

可以在帮助文档中搜索关键词 (Array Operation) 查询数组运算规则.

## 标量-数组运算

数组与标量的加、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.

设数组  $a = [a_1, a_2, \dots, a_n]$ , 标量为  $c$

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

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

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

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

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

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

例如: `a=1:3;` `c=2;`

MATLAB 命令窗口中分别输入 `a+c`, `a-c`, `a.*c`, `a./c`, `a. \ c`, `a.^c`, `c.^a`

## 数组-数组运算

当两个数组有**相同维数** (通常指  $n \times m$  相同) 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.

设数组  $a = [a_1, a_2, \dots, a_n]$   $b = [b_1, b_2, \dots, b_n]$

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

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

$$a ./ b = [a_1 / b_1, a_2 / b_2, \dots, a_n / b_n] \quad \text{左除}$$

$$a . \backslash b = [b_1 / a_1, b_2 / a_2, \dots, b_n / a_n] \quad \text{右除}$$

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

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

例: 略.

**咳咳咳:** `a = 1:3; b = (3:6).'`, MATLAB 内运算 `a+b` 是否可以, 若可以观察结果.

## 函数-数组运算

第一章中涉及的所有函数均服从数组运算, 关系运算和逻辑运算均是如此.

`x = 0:30:360;` `y = sind(x)`

```
y =
0      0.5000      0.8660      1.0000      0.8660      0.5000      0      -0.5000      -0.8660      ...
      -1.0000     -0.8660     -0.5000      0
```

`yd0=y>=0`

```
yd0 =
1×13 logical array
1      1      1      1      1      1      1      0      0      0      0      0      1
```

`yh0=y或0` (竖杠无法显示...)

```
yh0 =
1×13 logical array
0      1      1      1      1      1      0      1      1      1      1      1      0
```

## 向量的点乘和叉乘

向量的点乘和叉乘在物理学中经常用到.

向量点乘得到的是标量, 向量叉乘得到的是是向量.

向量  $\mathbf{F} = [1, 2, 3]$ ,  $\mathbf{r} = [0, 1, 4]$

二者的点乘数学上表示为  $W = \mathbf{F} \cdot \mathbf{r}$ , MATLAB 语言

```
W = dot(F,r) % equal to sum(F.*r)
```

叉乘表示为  $\mathbf{M} = \mathbf{F} \times \mathbf{r}$ , MATLAB 语言

```
M = cross(F,r)
```

# 矩阵的运算规则

MATLAB 中矩阵的运算服从线性代数规则. 两矩阵之间的大小维度等取决于它们之间的运算符. 矩阵运算结果往往不同于数组运算结果.

- ① 同维度同大小的矩阵可以相加减, 结算结果与数组运算一样.
- ② 矩阵相乘时, 内维大小要相等, 即  $A_{mn}$  与  $B_{nk}$  是可以相乘的.  
矩阵与标量的乘法和数组与标量的乘法结果相同.
- ③ 只有方阵可以进行乘方运算.

# 矩阵的加减法

矩阵的行数和列数相等时, 可以相加减

```
A = [5  1; 0  9];
```

```
B = [2  -2; 1  1];
```

```
A + B
```

```
ans =
```

```
7      -1
```

```
1      10
```

```
A - B
```

```
ans =
```

```
3      3
```

```
-1     8
```

相应位置的元素进行加减.



# 矩阵的乘法

设两个矩阵  $A = \begin{bmatrix} 2 & 1; 1 & 2 \end{bmatrix}$ ;  $B = \begin{bmatrix} 3 & 4; 5 & 6 \end{bmatrix}$ ;

## 数组乘法

```
A.*B
ans =
6      4
5     12
```

设矩阵  $A = \begin{bmatrix} 1 & 4; 8 & 0; -1 & 3 \end{bmatrix}$ ;  $B = \begin{bmatrix} -1 & 7 & 4; 2 & 1 & -2 \end{bmatrix}$ ;

## 数组乘法

```
A.*B
Matrix dimensions must agree.
```

数与矩阵相乘等于数与每个元素相乘 ←

## 矩阵乘法

```
A*B
ans =
11     14
13     16
```

## 矩阵乘法

```
A*B
ans =
7      11     -4
-8     56     32
7      -4    -10
```

## 矩阵的除法

矩阵的出发经常用在解线性方程组.

一般地, 矩阵的除法要求两矩阵拥有相同的行数 (左除) 或者具有相同的列数 (右除).

- ① 对于方程组  $Ax = B$ , 则解  $x = A \setminus B$

```
A = [1 2 0; 0 4 3]; b = [8; 18];
x = A\b
x =
    0
 4.0000
 0.6667
```

- ② 对于方程组  $xA = B$ , 则解  $x = B/A$

```
A = [1 1 3; 2 0 4; -1 6 -1]; B = [2 19 8];
x = B/A
x =
 1.0000    2.0000    3.0000
```

# 矩阵的除法解方程组

$$5x + 2y - 9z = -18$$

$$-9x - 2y + 2z = -7$$

$$6x + 7y + 3z = 29$$

系数矩阵  $A = [5 \ 2 \ -9; -9 \ -2 \ 2; 6 \ 7 \ 3]$ ,  $b = [-18; -7; 29];$

则方程的解为

```
A \ b
ans =
1.0000
2.0000
3.0000
```

# 矩阵的转置, 行列式, 逆矩阵, 秩

## ① 转置操作 `'`

```
A = [-1 2 0; 6 4 1]
```

```
A'
```

矩阵含复数元素, 转置操作会自动计算复数的共轭值. 若只需转置, 用`.'`

## ② 行列式 (方阵, 与线性方程有关) `det`

```
A = [1 -2 4; -5 2 0; 1 0 3]
```

```
det(A)
```

```
ans = -32
```

## ③ 矩阵的逆 (方阵) `inv` 或 `^(-1)`

```
A = [1 0 2; -1 5 0; 0 3 -9];
```

```
A^(-1) % or inv(A)
```

```
ans =
```

```
0.8824    -0.1176     0.1961
```

```
0.1765     0.1765     0.0392
```

```
0.0588     0.0588    -0.0980
```

## 轻微感受一下矩阵运算的便利 (强大?) 之处.....

其实绪论时已经强烈感受了以下.

按以下步骤操作:

- ① 创建  $2 \times 12$  数组 `X = [-6 -6 -7 0 7 6 6 -3 -3 0 0 -6; -7 2 1 8 1 2 -7 -7 -2 -2 -7 -7];` 这一操作与矩阵运算无关:-)
- ② 画图 `plot (X(1,:), X(2,:), '.-', 'markersize',18,'linewidth',2)`, 这一操作与矩阵运算亦无关:-)
- ③ 继续执行 `axis(10*[-1 1 -1 1]), axis square` 这一操作与矩阵运算还是无关:-)
- ④ 留空他用

## 轻微感受一下矩阵运算的便利 (强大?) 之处.....

其实绪论时已经强烈感受了以下.

按以下步骤操作:

- ① 创建  $2 \times 12$  数组 `X = [-6 -6 -7 0 7 6 6 -3 -3 0 0 -6; -7 2 1 8 1 2 -7 -7 -2 -2 -7 -7];` 这一操作与矩阵运算无关:-)
- ② 画图 `plot (X(1,:), X(2,:), '.-', 'markersize',18, 'linewidth',2),` 这一操作与矩阵运算亦无关:-)
- ③ 继续执行 `axis(10*[-1 1 -1 1]), axis square` 这一操作与矩阵运算还是无关:-)
- ④ 留空他用
- ⑤ 创建矩阵 `a = [0.5 0; 0 1],` 重复 1, 执行 `X=a * X;`, 重复 2 和 3.
- ⑥ 重复 5, 只是 `a = [1 0; 0 0.5]`
- ⑦ 重复 6, 只是 `a = [0 1; 0.5 0]`
- ⑧ 重复 7, 只是 `a = [0.5 0; 0 -1]`
- ⑨ 重复 8, 只是 `a = [cos(t) -sin(t); sin(t) cos(t)],` 其中  $t$  可以是  $[0 \ 2\pi]$  间的任意数值.
- ⑩ 重复 9, 只是 `a = 2*rand(2)-1,` 多运行此步骤看效果.

# 致谢

本课程参考资料包括但不限于 (排名不分先后):

- ① MATLAB 帮助文档
- ② Experiments with MATLAB, C.Moler
- ③ MATLAB 高效编程技巧与应用: 25 个案例分析, 吴鹏 (rocwoods)
- ④ 精通 MATLAB R2011a, 张志勇
- ⑤ MATLAB 2014a 完全自学一本通, 刘浩/韩晶
- ⑥ Elementary Mathematical and Computational Tools for Electrical and Computer Engineers using MATLAB, Jamal T.Manassah
- ⑦ Physical Modeling in MATLAB, Allen B.Downey
- ⑧ 一些网络资源 (知乎, MATLAB 中文论坛等)
- ⑨ ...