



关于矩阵及其操作

第2章 矩阵及其操作

- 2.1 数据类型
- 2.2 变量及其操作
- 2.3 矩阵基础
- 2.4 矩阵运算
- 2.5 矩阵的基本操作

2.1 数据类型

- MATLAB有15种基本数据类型，每种基本数据类型均以数组/矩阵的形式出现。
 - 1. 数值类型
 - 2. 逻辑类型
 - 3. 字符和字符串类型
 - 4. 结构体类型

1. 数值类型

- (1) 整数
- (2) 浮点数
- (3) 复数
- (4) Inf
- (5) NaN

(1) 整数类型

- MATLAB支持1、2、4和8字节的有符号整数和无符号整数。

名称	表示范围	转换函数	名称	表示范围	转换函数
有符号 1 字节整数	$-2^7 \sim 2^7 - 1$	int8()	无符号 1 字节整数	$0 \sim 2^8 - 1$	uint8()
有符号 2 字节整数	$-2^{15} \sim 2^{15} - 1$	int16()	无符号 2 字节整数	$0 \sim 2^{16} - 1$	uint16()
有符号 4 字节整数	$-2^{31} \sim 2^{31} - 1$	int32()	无符号 4 字节整数	$0 \sim 2^{32} - 1$	uint32()
有符号 8 字节整数	$-2^{63} \sim 2^{63} - 1$	int64()	无符号 8 字节整数	$0 \sim 2^{64} - 1$	uint64()

(2) 浮点数类型

- MATLAB有单精度和双精度两种浮点数。

名称	存储空间	表示范围	转换函数
单精度浮点数	4 字节	$-3.40282 \times 10^{38} \sim 3.40282 \times 10^{38}$	single()
双精度浮点数	8 字节	$-1.79769 \times 10^{308} \sim 1.79769 \times 10^{308}$	double()

(3) 复数类型

- 复数包含实部和虚部，用i或者j表示虚部。
- 生成复数有两种方法：

```
>> z=3+4i
```

```
z =
```

```
3.0000 + 4.0000i
```

```
>> complex(3, 4)
```

```
ans =
```

```
3.0000 + 4.0000i
```

```
>> help complex
```

(4) Inf和NaN

- Inf和-Inf分别表示正无穷大和负无穷。
- NaN (Not a Number) 表示一个既不是实数也不是复数的值。

2. 逻辑类型

- 在MATLAB中逻辑类型包含true和false，分别由1和0表示。函数logical将任何非零的数值转换为true（即1），将数值0转换为false（即0）。

3. 字符和字符串类型

- 在MATLAB中，数据类型（char）表示一个字符；
- 一个char类型的 $1 \times n$ 数组称为字符串string

例 在命令窗口用“单引号对”表示字符串'I am a great person'，具体代码如下：

```
str='I am a great person'
```

例 在命令窗口用函数 char()构造字符串'AB'，具体代码如下：

```
str=char([65 66])
```

4. 结构体类型

- 结构体类型是一种由若干属性（field）组成的MATLAB数组，其中的每个属性可以是任意数据类型。
- 结构体数组的创建及操作将在第4章里面进行详细的介绍。

2.2 变量及其操作

○ 变量

- 变量名以字母开头，后接字母、数字或下划线的字符序列；
- 变量名区分字母的大小写。

○ 赋值

○ 变量=表达式

```
>> num_students = 25  
num_students =  
25
```

```
>> x=1+2i, y=3-sqrt(17),  
z=(cos(abs(x+y))-sin(78*pi/180))/(x+abs(y))  
x =  
1.0000 + 2.0000i  
y =  
-1.1231  
z =  
-0.3488 + 0.3286i
```

特殊变量(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。预定义变量有特定的含义，在使用时，应尽量避免对这些变量重新赋值。

pi	3.14159265...
i	Imaginary unit, $\sqrt{-1}$
j	Same as i
eps	Floating-point relative precision, $\epsilon = 2^{-52}$
realmin	Smallest floating-point number, 2^{-1022}
realmax	Largest floating-point number, $(2 - \epsilon)2^{1023}$
Inf	Infinity
NaN	Not-a-number

内存变量的管理

- 利用MATLAB工作空间窗口可实现对内存变量的查看、修改、保存、删除、导出及画图等操作。
- 利用clear命令可删除工作空间中的变量。
- 利用who和whos命令可分别用于显示在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详细信息。

内存变量的保存与载入

mat文件

- 利用mat文件可以把当前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文件的生成和装入分别由save和load命令来完成。
 - save 文件名 变量名表
 - load 文件名 变量名表
- ```
>> save data x y z
>> clear
>> load data z
```

# 对load和save命令的一点说明

---

- save 文件名 变量名表
  - load 文件名 变量名表
- ¡ 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。
- ¡ 更多内容，请
- >> help save
  - >> help load

## 2.3 矩阵基础

---

- 在MATLAB中，所有的数据均以二维、三维或高维矩阵的形式存储，每个矩阵的单元可以是数值类型、逻辑类型、字符类型或者其他任何数据类型。
  - 对于标量，可以用 $1 \times 1$ 矩阵来表示；
  - 对于一组 $n$ 个数据，可以用 $1 \times n$ 矩阵来表示；
  - 对于多维数组，可以用多维矩阵来表示。

## 2.3 矩阵基础

### 1. 矩阵的索引

---

- $A(i, j)$  — 第 $i$ 行、第 $j$ 列的元素
- $A(i, :)$  — 第 $i$ 行的全部元素
- $A(:, j)$  — 第 $j$ 列全部元素
- $A(i:i+m, :)$  — 第 $i \sim i+m$ 行的全部元素
- $A(:, k:k+m)$  — 第 $k \sim k+m$ 列的全部元素
- $A(i:i+m, k:k+m)$  — 第 $i \sim i+m$ 行内，并在第 $k \sim k+m$ 列中的所有元素
- 还可利用一般向量和`end`运算符来表示矩阵下标，`end`表示某一维的末尾元素下标。

## 2. 矩阵的创建

- 在命令窗口直接输入矩阵的各个元素

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

```
A =
```

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

- 通过load命令载入数据文件

```
>> load mymatrix.txt
```

```
>> mymatrix
```

```
mymatrix =
```

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

- Matlab内在函数（特殊矩阵）
- 利用m文件创建

# 特殊矩阵生成函数

| 函数名   | 函数用途                           | 基本调用格式                                        |                                                    |
|-------|--------------------------------|-----------------------------------------------|----------------------------------------------------|
| ones  | 产生矩阵元素全为 1 的矩阵                 | $A=\text{ones}(n)$<br>$A=\text{ones}(m, n)$   | 产生 $n \times n$ 的全 1 矩阵<br>产生 $m \times n$ 的全 1 矩阵 |
| zeros | 产生矩阵元素全为 0 的矩阵                 | $A=\text{zeros}(n)$<br>$A=\text{zeros}(m, n)$ | 产生 $n \times n$ 的全 0 矩阵<br>产生 $m \times n$ 的全 0 矩阵 |
| eye   | 产生单位矩阵, 即主对角线上的元素为 1, 其他元素全为 0 | $A=\text{eye}(n)$                             | 产生 $n \times n$ 的单位矩阵                              |
| diag  | 将向量转化为对角矩阵                     | $A=\text{diag}(v)$                            | 把向量 $v$ 转换为一个对角矩阵                                  |

# 特殊矩阵生成函数

| 函数名      | 函数用途                    | 基本调用格式                                       |                                                                         |
|----------|-------------------------|----------------------------------------------|-------------------------------------------------------------------------|
| magic    | 产生魔方矩阵，即每行、每列之和相等的矩阵    | $A=\text{magic}(n)$                          | 产生 $n \times n$ 的魔方矩阵                                                   |
| rand     | 产生 0~1 均匀分布的随机数         | $A=\text{rand}(n)$<br>$A=\text{rand}(m,n)$   | 产生 $n \times n$ 的随机数矩阵，其中，随机数服从 0~1 的均匀分布（下同）<br>产生 $m \times n$ 的随机数矩阵 |
| randn    | 产生均值为 0 且方差为 1 的高斯分布随机数 | $A=\text{randn}(n)$<br>$A=\text{randn}(m,n)$ | 产生 $n \times n$ 的随机数矩阵，其中，随机数服从标准高斯分布（下同）<br>产生 $m \times n$ 的随机数矩阵     |
| randperm | 产生整数 1~ $n$ 的随机排列       | $A=\text{randperm}(n)$                       | 产生整数 1~ $n$ 的随机排列                                                       |

# 特殊矩阵生成函数

```
>> B=magic(4)
```

```
B =
```

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

```
>> A=B(:,[1 3 2 4])
```

```
A =
```

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

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

# 向量的创建

- 利用冒号表达式产生行向量，调用格式： $e1:e2:e3$

```
>> 1:10
ans =
 1 2 3 4 5 6 7 8 9 10
>> 100:-7:50
ans =
 100 93 86 79 72 65 58 51
```

- 用linspace函数产生行向量，调用格式：

$\text{linspace}(a, b, n)$

```
>> linspace(0, 100, 6)
ans =
 0 20 40 60 80 100
```

- 用logspace函数产生行向量，调用格式：

$\text{logspace}(a, b, n)$

```
>> logspace(-2, 2, 6)
ans =
 0.0100 0.0631 0.3981 2.5119 15.8489 100.0000
```

## 3. 矩阵的合并

---

- 把两个或者两个以上的矩阵连接成一个新的矩阵。矩阵构造符  $[]$  可用于构造矩阵，并可以作为一个矩阵合并操作符。
  - $C=[A\ B]$  在水平方向合并矩阵  $A$  和  $B$ ;
  - $C=[A;B]$  在竖直方向合并矩阵  $A$  和  $B$ 。

# 3.矩阵的合并

具有相同行数的两个矩阵，合并为一个新矩阵

|   |   |
|---|---|
| 1 | 2 |
| 3 | 4 |
| 5 | 6 |

 + 

|    |    |    |
|----|----|----|
| 4  | 5  | 6  |
| 7  | 8  | 9  |
| 10 | 11 | 12 |

 = 

|   |   |    |    |    |
|---|---|----|----|----|
| 1 | 2 | 4  | 5  | 6  |
| 3 | 4 | 7  | 8  | 9  |
| 5 | 6 | 10 | 11 | 12 |

<sup>3×2</sup>不具有相同行数的两个矩阵，<sup>3×3</sup>不允许合并为<sup>3×5</sup>一个新矩阵

|   |   |
|---|---|
| 1 | 2 |
| 3 | 4 |
| 5 | 6 |

 + 

|   |   |   |
|---|---|---|
| 4 | 5 | 6 |
| 7 | 8 | 9 |

 ≠ 

|   |   |   |   |   |
|---|---|---|---|---|
| 1 | 2 | 4 | 5 | 6 |
| 3 | 4 | 7 | 8 | 9 |
| 5 | 6 |   |   |   |

3×2                      2×3

# 3.矩阵的合并

## 矩阵合并函数

| 函数名     | 函数描述           | 基本调用格式                       |                               |
|---------|----------------|------------------------------|-------------------------------|
| cat     | 在指定的方向合并矩阵     | cat(1, A, B)<br>cat(2, A, B) | 与[A;B]用途一致<br>与[A B]用途一致      |
| horzcat | 在水平方向合并矩阵      | horzcat(A, B)                | 与[A B]用途一致                    |
| vertcat | 在垂直方向合并矩阵      | vertcat(A, B)                | 与[A;B]用途一致                    |
| repmat  | 通过复制矩阵来构造新矩阵   | repmat(A, M, N)              | 得到 $M \times N$ 块矩阵, 其中每块都为 A |
| blkdiag | 用已知矩阵来构造块对角化矩阵 | blkdiag(A, B)                | 得到以矩阵 A 和 B 为对角块的矩阵           |

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

### 3. 矩阵的合并

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

```
>> B = [A A+32; A+48 A+16]
```

```
B =
```

|    |    |    |    |    |    |    |    |
|----|----|----|----|----|----|----|----|
| 16 | 3  | 2  | 13 | 48 | 35 | 34 | 45 |
| 5  | 10 | 11 | 8  | 37 | 42 | 43 | 40 |
| 9  | 6  | 7  | 12 | 41 | 38 | 39 | 44 |
| 4  | 15 | 14 | 1  | 36 | 47 | 46 | 33 |
| 64 | 51 | 50 | 61 | 32 | 19 | 18 | 29 |
| 53 | 58 | 59 | 56 | 21 | 26 | 27 | 24 |
| 57 | 54 | 55 | 60 | 25 | 22 | 23 | 28 |
| 52 | 63 | 62 | 49 | 20 | 31 | 30 | 17 |

## 4. 矩阵的扩展

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

```
>> B=A; B(4,5)=17
```

```
B =
```

```
16 3 2 13 0
 5 10 11 8 0
 9 6 7 12 0
 4 15 14 1 17
```

```
>> A(end+1,:)=18
```

```
A =
```

```
16 3 2 13 0
 5 10 11 8 0
 9 6 7 12 0
 4 15 14 1 17
 18 18 18 18 18
```

## 5. 删除矩阵的行、列

```
>> A(end, :)=[]
```

```
A =
```

```
 16 3 2 13
 5 10 11 8
 9 6 7 12
 4 15 14 1 17
```

```
A =
 16 3 2 13 0
 5 10 11 8 0
 9 6 7 12 0
 4 15 14 1 17
 18 18 18 18 18
```

```
>> A(:, end)=[]
```

```
A =
```

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

## 6. 改变矩阵结构

| 函数名        | 函数描述          | 基本调用格式                                            |                                                             |
|------------|---------------|---------------------------------------------------|-------------------------------------------------------------|
| reshape    | 按照长列向量的顺序重排元素 | $B = \text{reshape}(A, m, n)$                     | 把 $A$ 重新排列为 $m \times n$ 的矩阵                                |
| rot90      | 旋转矩阵          | $B = \text{rot90}(A)$<br>$B = \text{rot90}(A, k)$ | 逆时针旋转矩阵 $90^\circ$<br>逆时针旋转矩阵 $k \times 90^\circ$ , $k$ 为整数 |
| fliplr     | 以垂直方向为轴做镜像    | $B = \text{fliplr}(A)$                            |                                                             |
| flipud     | 以水平方向为轴做镜像    | $B = \text{flipud}(A)$                            |                                                             |
| flipdim    | 以指定的轴做镜像      | $B = \text{flipdim}(A, dim)$                      | $dim=1$ 以水平方向为轴做镜像<br>$dim=2$ 以垂直方向为轴做镜像                    |
| transpose  | 矩阵的转秩         | $B = \text{transpose}(A)$                         | 相当于 $B=A'$                                                  |
| ctranspose | 矩阵的共轭转秩       | $B = \text{ctranspose}(A)$                        | 相当于 $B=A'$                                                  |

# 7. 基于列的操作规则

- 在MATLAB中，矩阵元素按列存储，先第一列，再第二列，依此类推。
- 阵列中的每列数据代表一个变量，每一行代表一个观察者，第  $(i, j)$  个要素是第  $i$  个观察者的第  $j$  个变量。
- 对5个人的3个身体指标数据进行记录

| 姓名 \ 指标 | 心率/次 | 体重/千克 | 锻炼时间/小时 |
|---------|------|-------|---------|
| 张某      | 72   | 134   | 3.2     |
| 王某      | 81   | 201   | 3.5     |
| 李某      | 69   | 156   | 7.1     |
| 赵某      | 82   | 148   | 2.4     |
| 周某      | 75   | 170   | 1.2     |

```
Data = [72 134 32
 81 201 35
 69 156 71
 82 148 24
 75 170 12
];
```

# 7. 基于列的操作规则

## ○ 例子

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

```
A =
```

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

```
>> a3=A(3),a5=A(5)
```

```
a3 =
```

```
9
```

```
a5 =
```

```
3
```

- 序号 (Index) 与下标 (Subscript ) 一一对应，以  $m \times n$  矩阵A为例，矩阵元素  $A(i, j)$  的序号为  $(j-1)*m+i$ 。其相互转换关系可利用 `sub2ind` 和 `ind2sub` 函数求得。

## 7. 基于列的操作规则

```
>> sum(Data)
ans =
 3790 8090 174
```

```
>> mean(Data)
ans =
 758 1618 34800
```

```
>> max(Data)
ans =
 820 2010 7.1
```

```
Data = [72 134 32
 81 201 35
 69 156 71
 82 148 24
 75 170 12];
```

| 姓名 \ 指标 | 心率/次 | 体重/千克 | 锻炼时间/小时 |
|---------|------|-------|---------|
| 张某      | 72   | 134   | 3.2     |
| 王某      | 81   | 201   | 3.5     |
| 李某      | 69   | 156   | 7.1     |
| 赵某      | 82   | 148   | 2.4     |
| 周某      | 75   | 170   | 1.2     |

# 基于列操作规则的函数

## 基本操作

- max - 最大值
- min - 最小值
- mean - 平均值
- median - 中值
- std - 标准差
- var - 方差
- sort - 升序排列
- sortrows - 按行的升序排列
- sum - 求和
- prod - 求积
- hist - 直方图
- histc - 直方图计数
- trapz - 梯形数值积分
- cumsum - 元素的累积求和
- cumprod - 元素的累积求积
- cumtrapz - 累计梯形数值积分

## 有限差分

- diff - 微分和导数
- gradient - 梯度
- del2 - 离散拉普拉斯算

子

## 相关性分析

- corrcoef - 相关系数
- cov - 协方差矩阵
- subspace - 子空间的夹角

## 滤波和卷积

- filter - 一维数字滤波器
- filter2 - 二维数字滤波器
- conv - 卷积和多项式乘法
- conv2 - 二维卷积
- convn - N维卷积
- deconv - 反卷积和多项式除法运

算

- detrend - 去除线性趋势

# 基于列操作规则的函数

## · 傅里叶变换

|                        |               |
|------------------------|---------------|
| <code>fft</code>       | - 离散傅里叶变换     |
| <code>fft2</code>      | - 二维离散傅立叶变换   |
| <code>fftn</code>      | - N维离散傅里叶变换   |
| <code>ifft</code>      | - 逆离散傅立叶变换    |
| <code>ifft2</code>     | - 二维逆离散傅立叶变换  |
| <code>ifftn</code>     | - N维离散傅里叶逆变换  |
| <code>fftshift</code>  | - 移零频率分量的频谱中心 |
| <code>ifftshift</code> | - 逆FFTSHIFT   |

## 8.矩阵的下标引用

---

通过矩阵下标来存取矩阵元素。

1) 访问单个元素

2) 线性引用元素

3) 访问多个元素

# 1) 访问单个元素

---

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

A (i, j)

i——行号;

j——列号。

```
>> A(1,4) + A(2,4) + A(3,4) + A(4,4)
ans =
 34
```

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/635102230002012003>