

第 2 章 MATLAB 矩阵及其运算

Lecturer: 白煌

杭州师范大学
信息科学与技术学院

2022.9.23



本章要点

- MATLAB 数据类型、变量和数据操作
- MATLAB 矩阵的操作、运算与矩阵分析
- MATLAB 字符串、结构数据和单元数据
- MATLAB 稀疏矩阵及其操作



目录

① 2.1 MATLAB 数据类型

② 2.2 变量和数据操作

③ 2.3 MATLAB 矩阵

④ 2.4 MATLAB 运算



整型数据

MATLAB 数据类型较为丰富，既有数值型、字符串等基本数据类型，又有结构（`structure`）、单元（`cell`）等复合数据类型。

在 MATLAB 中，没有专门的逻辑型数据，而以数值 1（非零）表示“真”，以数值 0 表示“假”。



整型数据

MATLAB 数据类型较为丰富，既有数值型、字符串等基本数据类型，又有结构（`structure`）、单元（`cell`）等复合数据类型。

在 MATLAB 中，没有专门的逻辑型数据，而以数值 1（非零）表示“真”，以数值 0 表示“假”。

整型数据是不带小数的数，有带符号整数和无符号整数之分。表 2-1 列出了各种整型数据的取值范围 and 对应的转换函数。



浮点型数据

浮点型数据有单精度（single）和双精度（double）之分：

- 单精度型实数在内存中占用 4 个字节
- 双精度型实数在内存中占用 8 个字节，双精度型的数据精度更高



浮点型数据

浮点型数据有单精度（single）和双精度（double）之分：

- 单精度型实数在内存中占用 4 个字节
- 双精度型实数在内存中占用 8 个字节，双精度型的数据精度更高

在 MATLAB 中，数据默认为双精度型。single 函数可以将其他类型的数据转换为单精度型，double 函数可以将其他类型的数据转换为双精度型。



复型数据

复型数据包括实部和虚部两个部分，实部和虚部默认为双精度型。在 MATLAB 中，虚数单位用 i 或 j 表示。例如， $6+5i$ 与 $6+5j$ 表示的是同一个复数，也可以写成 $6+5*i$ 或 $6+5*j$ ，这里将 i 或 j 看作一个运算量参与表达式的运算。



2.2.1 变量与赋值

- 变量名是以字母开头，后接字母、数字或下划线的字符序列
- 最多 63 个字符
- 变量名区分字母的大小写



2.2.1 变量与赋值

MATLAB 赋值语句有两种格式:

- 变量 = 表达式
- 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，其结果可以是单个的数，也可以是一个矩阵。



2.2.1 变量与赋值

MATLAB 赋值语句有两种格式:

- 变量 = 表达式
- 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，其结果可以是单个的数，也可以是一个矩阵。

在 MATLAB 语句后面可以加上注释，用于解释或说明语句的含义，对语句处理结果不产生任何影响。注释以 % 开头，后面是注释的内容。



2.2.1 变量与赋值

例 2-1: 计算下式表达式的值

$$\frac{\cos |x + y| - \sin 78^\circ}{x + |y|}$$

其中 $x = 1 + 2i$, $y = 3 - \sqrt{17}$ 。将结果赋给变量 z , 并显示计算结果。



2.2.1 变量与赋值

例 2-1: 计算下式表达式的值

$$\frac{\cos |x + y| - \sin 78^\circ}{x + |y|}$$

其中 $x = 1 + 2i$, $y = 3 - \sqrt{17}$ 。将结果赋给变量 z , 并显示计算结果。

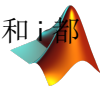
在 MATLAB 命令行窗口输入命令并得到输出结果:

```
>> x = 1+2i;
```

```
>> y = 3-sqrt(17);
```

```
>> z = (cos(abs(x+y))-sin(78*pi/180))/(x+abs(y))
```

其中 sqrt、cos、abs、sin 均是 MATLAB 提供的数学函数, pi 和 i 都是 MATLAB 预先定义的变量。



2.2.2 预定义变量

在 MATLAB 工作空间中，还驻留有几个由系统本身定义的变量。它们有特定的含义，在使用时，应尽量避免对这些变量重新赋值。

表 2-2 列出了一些常用的预定义变量。



2.2.3 内存变量的管理

1. 内存变量的删除与修改:



2.2.3 内存变量的管理

1. 内存变量的删除与修改:

- MATLAB 工作区窗口专门用于内存变量的管理



2.2.3 内存变量的管理

1. 内存变量的删除与修改:

- MATLAB 工作区窗口专门用于内存变量的管理
- 在工作区窗口中可以显示所有内存变量的属性



2.2.3 内存变量的管理

1. 内存变量的删除与修改:

- MATLAB 工作区窗口专门用于内存变量的管理
- 在工作区窗口中可以显示所有内存变量的属性
- 选择右键快捷菜单中的“删除”命令，就能清除选中变量



2.2.3 内存变量的管理

1. 内存变量的删除与修改:

- MATLAB 工作区窗口专门用于内存变量的管理
- 在工作区窗口中可以显示所有内存变量的属性
- 选择右键快捷菜单中的“删除”命令，就能清除选中变量
- 双击变量或选择右键快捷菜单中的“打开所选内容”命令，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，也可以修改变量中的具体元素



2.2.3 内存变量的管理

1. 内存变量的删除与修改:

- MATLAB 工作区窗口专门用于内存变量的管理
- 在工作区窗口中可以显示所有内存变量的属性
- 选择右键快捷菜单中的“删除”命令，就能清除选中变量
- 双击变量或选择右键快捷菜单中的“打开所选内容”命令，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，也可以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



2.2.3 内存变量的管理

1. 内存变量的删除与修改:

- MATLAB 工作区窗口专门用于内存变量的管理
- 在工作区窗口中可以显示所有内存变量的属性
- 选择右键快捷菜单中的“删除”命令，就能清除选中变量
- 双击变量或选择右键快捷菜单中的“打开所选内容”命令，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，也可以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 用于显示 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 还给出它们的大小、所占字节数及数据类型等信息



2.2.3 内存变量的管理

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。MAT 文件是 MATLAB 保存数据的一种标准格式的二进制文件, 扩展名一定是 .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]



2.2.3 内存变量的管理

2. 内存变量文件:

- 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



2.2.4 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大小的矩阵。



2.2.4 MATLAB 常用数学函数

表 2-3 列出了一些常用数学函数。函数使用说明：

- 三角函数以弧度为单位计算
- `abs` 函数可以求实数的绝对值、复数的模、字符串的 ASCII 码值
- 用于取整的函数有 `fix`、`floor`、`ceil`、`round`，要注意它们的区别
- `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
- 符号函数 `sign`



2.2.5 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。
- MATLAB 内部每一个数据元素都是用双精度数来表示和存储的。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，各种格式符及其含义见表 2-4。



2.3.1 矩阵的建立

1. 直接输入法建立矩阵:

最简单的建立矩阵的方法是从键盘直接输入矩阵的元素。具体方法如下：将矩阵的元素用方括号括起来，按矩阵行的顺序输入各元素，同一行的各元素之间用空格或逗号分隔，不同行的元素之间用分号分隔。



2.3.1 矩阵的建立

2. 利用冒号表达式建立一个向量:

- 冒号表达式可以产生一个行向量，一般格式是

`e1:e2:e3`

其中 `e1` 为初始值，`e2` 为步长，`e3` 为终止值。



2.3.1 矩阵的建立

2. 利用冒号表达式建立一个向量:

- 冒号表达式可以产生一个行向量，一般格式是

$e1:e2:e3$

其中 $e1$ 为初始值， $e2$ 为步长， $e3$ 为终止值。

- 还可以用 `linspace` 函数产生行向量。其调用格式为

`linspace(a,b,n)`

其中 a 和 b 是生成向量的第一个和最后一个元素， n 是元素总数。

显然，`linspace(a,b,n)` 与 `a:(b-a)/(n-1):b` 等价。



2.3.1 矩阵的建立

3. 利用已建好的矩阵建立更大的矩阵:

大矩阵可由已建好的小矩阵拼接而成。注意行列尺寸!



2.3.1 矩阵的建立

3. 利用已建好的矩阵建立更大的矩阵:

大矩阵可由已建好的小矩阵拼接而成。注意行列尺寸!

补: 工作区生成矩阵。



2.3.2 矩阵的拆分

1. 矩阵元素的引用方式:



2.3.2 矩阵的拆分

1. 矩阵元素的引用方式:

- 通过下标引用矩阵的元素，例如

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



2.3.2 矩阵的拆分

1. 矩阵元素的引用方式:

- 通过下标引用矩阵的元素，例如

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

- 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

$$A=[1,2,3;4,5,6] \Rightarrow A(3)=2$$



2.3.2 矩阵的拆分

1. 矩阵元素的引用方式:

- 通过下标引用矩阵的元素，例如

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

- 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

$$A=[1,2,3;4,5,6] \Rightarrow A(3)=2$$

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。



2.3.2 矩阵的拆分

2. 利用冒号表达式获得子矩阵:



2.3.2 矩阵的拆分

2. 利用冒号表达式获得子矩阵:

- $A(:,j)$ 表示取 A 矩阵第 j 列的全部元素



2.3.2 矩阵的拆分

2. 利用冒号表达式获得子矩阵:

- $A(:,j)$ 表示取 A 矩阵第 j 列的全部元素
- $A(i,:)$ 表示取 A 矩阵第 i 行的全部元素



2.3.2 矩阵的拆分

2. 利用冒号表达式获得子矩阵:

- $A(:,j)$ 表示取 A 矩阵第 j 列的全部元素
- $A(i,:)$ 表示取 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



2.3.2 矩阵的拆分

2. 利用冒号表达式获得子矩阵:

- $A(:,j)$ 表示取 A 矩阵第 j 列的全部元素
- $A(i,:)$ 表示取 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
- $A(i:i+m,:)$ 表示取 A 矩阵第 $i \sim i+m$ 行的全部元素



2.3.2 矩阵的拆分

2. 利用冒号表达式获得子矩阵:

- $A(:,j)$ 表示取 A 矩阵第 j 列的全部元素
- $A(i,:)$ 表示取 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
- $A(i:i+m,:)$ 表示取 A 矩阵第 $i \sim i+m$ 行的全部元素
- $A(:,k:k+m)$ 表示取 A 矩阵第 $k \sim k+m$ 列的全部元素



2.3.2 矩阵的拆分

2. 利用冒号表达式获得子矩阵:

- $A(:,j)$ 表示取 A 矩阵第 j 列的全部元素
- $A(i,:)$ 表示取 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
- $A(i:i+m,:)$ 表示取 A 矩阵第 $i \sim i+m$ 行的全部元素
- $A(:,k:k+m)$ 表示取 A 矩阵第 $k \sim k+m$ 列的全部元素
- $A(i:i+m,k:k+m)$ 表示取 A 矩阵第 $i \sim i+m$ 行内，并在第 $k \sim k+m$ 列中的所有元素



2.3.2 矩阵的拆分

2. 利用冒号表达式获得子矩阵:

- $A(:,j)$ 表示取 A 矩阵第 j 列的全部元素
- $A(i,:)$ 表示取 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
- $A(i:i+m,:)$ 表示取 A 矩阵第 $i \sim i+m$ 行的全部元素
- $A(:,k:k+m)$ 表示取 A 矩阵第 $k \sim k+m$ 列的全部元素
- $A(i:i+m,k:k+m)$ 表示取 A 矩阵第 $i \sim i+m$ 行内，并在第 $k \sim k+m$ 列中的所有元素
- $A(:)$ 表示将 A 矩阵每一列元素堆叠起来成为一个列向量



2.3.2 矩阵的拆分

2. 利用冒号表达式获得子矩阵:

- $A(:,j)$ 表示取 A 矩阵第 j 列的全部元素
- $A(i,:)$ 表示取 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
- $A(i:i+m,:)$ 表示取 A 矩阵第 $i \sim i+m$ 行的全部元素
- $A(:,k:k+m)$ 表示取 A 矩阵第 $k \sim k+m$ 列的全部元素
- $A(i:i+m,k:k+m)$ 表示取 A 矩阵第 $i \sim i+m$ 行内, 并在第 $k \sim k+m$ 列中的所有元素
- $A(:)$ 表示将 A 矩阵每一列元素堆叠起来成为一个列向量
- 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



2.3.2 矩阵的拆分

3. 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=[]`



2.3.2 矩阵的拆分

3. 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=[]`
- `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, 只是尺寸为 0



2.3.2 矩阵的拆分

4. 改变矩阵的形状:

`reshape(A,m,n)` 函数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。



2.3.2 矩阵的拆分

4. 改变矩阵的形状:

`reshape(A,m,n)` 函数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。

注意: 在 MATLAB 中, 矩阵元素按列存储, 即首先存储矩阵的第 1 列元素, 然后存储第 2 列元素, 一直到矩阵的最后一列元素。`reshape` 函数只是改变原矩阵的行数和列数, 即改变其逻辑结构, 但并不改变原矩阵元素个数及其存储结构。



2.3.3 特殊矩阵

1. 通用的特殊矩阵:

- **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) 区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



2.3.3 特殊矩阵

2. 用于专门学科的特殊矩阵:

- 魔方矩阵
- 范得蒙矩阵
- 希尔伯特矩阵
- 托普利兹矩阵
- 伴随矩阵
- 帕斯卡矩阵



2.4.1 算术运算

1. 基本算术运算:

MATLAB 的基本算术运算有: $+$ (加)、 $-$ (减)、 $*$ (乘)、 $/$ (右除)、 \backslash (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。



2.4.1 算术运算

(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 矩阵的尺寸相同，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 的尺寸不相同，则 MATLAB 将给出错误信息，提示用户两个矩阵的大小不匹配。



2.4.1 算术运算

(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 矩阵的尺寸相同，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 的尺寸不相同，则 MATLAB 将给出错误信息，提示用户两个矩阵的大小不匹配。

一个标量也可以和矩阵进行加减运算，这时把标量和矩阵的每一个元素进行加减运算。



2.4.1 算术运算

(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。



2.4.1 算术运算

(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。

在 MATLAB 中，还可以进行矩阵和标量相乘，标量可以是乘数也可以是被乘数。矩阵和标量相乘是矩阵中的每一个元素与此标量相乘。



2.4.1 算术运算

(3) 矩阵除法

有两种矩阵除法运算：\ 和 /，分别表示左除和右除。



2.4.1 算术运算

(3) 矩阵除法

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 $B * \text{inv}(A)$ 。



2.4.1 算术运算

(3) 矩阵除法

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 $B * \text{inv}(A)$ 。

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。



2.4.1 算术运算

(3) 矩阵除法

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 $B * \text{inv}(A)$ 。

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 $a/5 = 5 \backslash a = [2.1000 \ 5.0000]$ 。



2.4.1 算术运算

(3) 矩阵除法

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 $B * \text{inv}(A)$ 。

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 $A \backslash B \neq B/A$ 。



2.4.1 算术运算

(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。



2.4.1 算术运算

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 `.*`、`./`、`.\` 和 `.^`。两矩阵进行点运算是指它们的对应元素进行相关运算，要求两矩阵尺寸相同。



2.4.1 算术运算

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 `.*`、`./`、`.\` 和 `.^`。两矩阵进行点运算是指它们的对应元素进行相关运算, 要求两矩阵尺寸相同。

点运算是 MATLAB 很有特色的一个运算符, 在实际应用中起着很重要的作用。



2.4.2 关系运算

MATLAB 提供了 6 种关系运算符：<（小于）、<=（小于或等于）、>（大于）、>=（大于或等于）、==（等于）、~=（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。



2.4.2 关系运算

关系运算符的运算法则:



2.4.2 关系运算

关系运算符的运算法则:

- 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



2.4.2 关系运算

关系运算符的运算法则:

- 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
- 当参与比较的量是两个尺寸相同的矩阵时，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尺寸与原矩阵相同的矩阵，它的元素由 0 或 1 组成



2.4.2 关系运算

关系运算符的运算法则:

- 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
- 当参与比较的量是两个尺寸相同的矩阵时, 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尺寸与原矩阵相同的矩阵, 它的元素由 0 或 1 组成
- 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尺寸与原矩阵相同的矩阵, 它的元素由 0 或 1 组成



2.4.2 关系运算

例 2-7：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。



2.4.2 关系运算

例 2-7: 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=fix((90-10+1)*rand(5)+10)
```



2.4.2 关系运算

例 2-7：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=fix((90-10+1)*rand(5)+10)
```

(2) 判断 A 的元素是否可以被 3 整除

```
>> P=rem(A,3)==0
```

其中， $\text{rem}(A,3)$ 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 同型的零矩阵，P 是进行等于（==）比较的结果矩阵。



2.4.3 逻辑运算

MATLAB 提供了 3 种逻辑运算符：&（与）、|（或）和 ~（非）。



2.4.3 逻辑运算

逻辑运算的运算法则:



2.4.3 逻辑运算

逻辑运算的运算法则:

- 在逻辑运算中，非零元素为真，用 1 表示，零元素为假，用 0 表示



2.4.3 逻辑运算

逻辑运算的运算法则:

- 在逻辑运算中，非零元素为真，用 1 表示，零元素为假，用 0 表示
- 设参与逻辑运算的是两个标量 a 和 b ，那么，当 a 、 b 全为非零时， $a \& b$ 运算结果为 1，否则为 0； a 、 b 中只要有一个非零， $a | b$ 运算结果为 1；当 a 是零时， $\sim a$ 运算结果为 1，当 a 非零时， $\sim a$ 运算结果为 0



2.4.3 逻辑运算

逻辑运算的运算法则:

- 在逻辑运算中，非零元素为真，用 1 表示，零元素为假，用 0 表示
- 设参与逻辑运算的是两个标量 a 和 b ，那么，当 a 、 b 全为非零时， $a \& b$ 运算结果为 1，否则为 0； a 、 b 中只要有一个非零， $a | b$ 运算结果为 1；当 a 是零时， $\sim a$ 运算结果为 1，当 a 非零时， $\sim a$ 运算结果为 0
- 若参与逻辑运算的是两个同型矩阵，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型的矩阵，其元素由 1 或 0 组成



2.4.3 逻辑运算

- 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型的矩阵，其元素由 1 或 0 组成



2.4.3 逻辑运算

- 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型的矩阵，其元素由 1 或 0 组成
- 逻辑非是单目运算符，也服从矩阵运算规则



2.4.3 逻辑运算

- 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型的矩阵，其元素由 1 或 0 组成
- 逻辑非是单目运算符，也服从矩阵运算规则
- 在算术运算、关系运算、逻辑运算中，算术运算优先级最高，逻辑运算优先级最低（逻辑非除外）



2.4.3 逻辑运算

例 2-8: 建立矩阵 A, 然后找出大于 4 的元素的位置, 并输出相应位置的元素。



2.4.3 逻辑运算

例 2-8: 建立矩阵 A, 然后找出大于 4 的元素的位置, 并输出相应位置的元素。

(1) 建立矩阵 A

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



2.4.3 逻辑运算

例 2-8: 建立矩阵 A, 然后找出大于 4 的元素的位置, 并输出相应位置的元素。

(1) 建立矩阵 A

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

(2) 找出大于 4 的元素的位置

```
>> k=find(A > 4)
```



2.4.3 逻辑运算

例 2-8: 建立矩阵 A, 然后找出大于 4 的元素的位置, 并输出相应位置的元素。

(1) 建立矩阵 A

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

(2) 找出大于 4 的元素的位置

```
>> k=find(A > 4)
```

(3) 输出相应位置的元素

```
>> A(k)
```

注意: find 函数得到的矩阵元素位置是以元素序号来表示的。

