

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

2022.9.30



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 2.5 矩阵分析
- ② 2.6 矩阵的超越函数
- ③ 2.7 字符串
- ④ 2.8 结构数据和单元数据
- ⑤ 2.9 稀疏矩阵



2.5.1 对角阵与三角阵

1. 对角阵:



2.5.1 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。



2.5.1 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。



2.5.1 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。



2.5.1 对角阵与三角阵

(1) 提取矩阵的对角线元素:



2.5.1 对角阵与三角阵

(1) 提取矩阵的对角线元素:

- 设 A 为 $m \times n$ 矩阵, `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。



2.5.1 对角阵与三角阵

(1) 提取矩阵的对角线元素:

- 设 A 为 $m \times n$ 矩阵, `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。
- `diag(A)` 函数还有一种形式 `diag(A,k)`, 其功能是提取第 k 条对角线的元素。



2.5.1 对角阵与三角阵

(2) 构造对角矩阵:



2.5.1 对角阵与三角阵

(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。



2.5.1 对角阵与三角阵

(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 $n \times n$ ($n=m+|k|$) 对角阵, 其第 k 条对角线的元素即为向量 V 的元素。



2.5.1 对角阵与三角阵

例 2-9: 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。



2.5.1 对角阵与三角阵

例 2-9: 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
```



2.5.1 对角阵与三角阵

例 2-9: 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
```

如果要对 A 的每列元素乘以同一个数, 可以用一个对角阵右乘矩阵 A。



2.5.1 对角阵与三角阵

2. 三角阵:

- 上三角阵是矩阵的对角线以下的元素全为 0 的一种矩阵。
- 下三角阵是矩阵的对角线以上的元素全为 0 的一种矩阵。



2.5.1 对角阵与三角阵

(1) 上三角矩阵:

- 求矩阵 A 的上三角阵的 MATLAB 函数是 `triu(A)`。



2.5.1 对角阵与三角阵

(1) 上三角矩阵:

- 求矩阵 A 的上三角阵的 MATLAB 函数是 $\text{triu}(A)$ 。
- $\text{triu}(A)$ 函数也有另一种形式 $\text{triu}(A,k)$ ，其功能是求矩阵 A 的第 k 条对角线以上的元素。



2.5.1 对角阵与三角阵

(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，提取矩阵 A 的第 2 条对角线以上的元素，形成新的矩阵 B:

```
>> A=[7,13,-28;2,-9,8;0,34,5];
```

```
>> B=triu(A,2)
```



2.5.1 对角阵与三角阵

(2) 下三角矩阵:

在 MATLAB 中, 提取 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。



2.5.2 矩阵的转置与旋转

- 矩阵的转置：转置运算符是 (`.'`)，共轭转置运算符是 (`'`)。



2.5.2 矩阵的转置与旋转

- 矩阵的转置：转置运算符是 (`.'`)，共轭转置运算符是 (`'`)。
- 矩阵的旋转：利用函数 `rot90(A,k)` 将矩阵 `A` 逆时针旋转 90° 的 `k` 倍，当 `k` 为 1 时可省略。



2.5.2 矩阵的转置与旋转

- 矩阵的转置：转置运算符是 (`.'`)，共轭转置运算符是 (`'`)。
- 矩阵的旋转：利用函数 `rot90(A,k)` 将矩阵 `A` 逆时针旋转 90° 的 `k` 倍，当 `k` 为 1 时可省略。
- 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)`。



2.5.2 矩阵的转置与旋转

- 矩阵的转置：转置运算符是 (`.'`)，共轭转置运算符是 (`'`)。
- 矩阵的旋转：利用函数 `rot90(A,k)` 将矩阵 `A` 逆时针旋转 90° 的 `k` 倍，当 `k` 为 1 时可省略。
- 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)`。
- 矩阵的上下翻转：对矩阵实施上下翻转是将原矩阵的第一行和最后一行调换，第二行和倒数第二行调换，……，依次类推。
MATLAB 对矩阵 `A` 实施上下翻转的函数是 `flipud(A)`。



2.5.3 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：

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

则称 B 为 A 的逆矩阵，当然， A 也是 B 的逆矩阵。



2.5.3 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：

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

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 $\text{inv}(A)$ 。



2.5.3 矩阵的逆与伪逆

例 2-10: 用求逆矩阵的方法解线性方程组。

$$x + 2y + 3z = 5$$

$$x + 4y + 9z = -2$$

$$x + 8y + 27z = 6$$



2.5.3 矩阵的逆与伪逆

例 2-10: 用求逆矩阵的方法解线性方程组。

$$x + 2y + 3z = 5$$

$$x + 4y + 9z = -2$$

$$x + 8y + 27z = 6$$

$$Ax=b \Rightarrow x=A^{-1}b$$



2.5.3 矩阵的逆与伪逆

例 2-10: 用求逆矩阵的方法解线性方程组。

$$x + 2y + 3z = 5$$

$$x + 4y + 9z = -2$$

$$x + 8y + 27z = 6$$

$$Ax=b \Rightarrow x=A^{-1}b$$

```
>> A=[1,2,3;1,4,9;1,8,27];
```

```
>> b=[5,-2,6]';
```

```
>> x=inv(A)*b
```



2.5.3 矩阵的逆与伪逆

2. 矩阵的伪逆:

若矩阵 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：

$$A \cdot B \cdot A = A$$

$$B \cdot A \cdot B = B$$

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)`。



2.5.4 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 $\det(A)$ 。



2.5.5 矩阵的秩与迹

- 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)`。



2.5.5 矩阵的秩与迹

- 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)`。
- 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)`。



2.5.6 向量和矩阵的范数

- 矩阵或向量的范数用来度量矩阵或向量在某种意义下的长度。
- 范数有多种方法定义，其定义不同，范数值也就不同。



2.5.6 向量和矩阵的范数

1. 向量的 3 种常用范数及其计算函数:



2.5.6 向量和矩阵的范数

1. 向量的 3 种常用范数及其计算函数:

- `norm(V,1)`: 向量 V 的 1-范数, 计算向量元素的绝对值之和。



2.5.6 向量和矩阵的范数

1. 向量的 3 种常用范数及其计算函数:

- $\text{norm}(V,1)$: 向量 V 的 1-范数, 计算向量元素的绝对值之和。
- $\text{norm}(V)$ 或 $\text{norm}(V,2)$: 向量 V 的 2-范数, 计算向量元素平方和的平方根。



2.5.6 向量和矩阵的范数

1. 向量的 3 种常用范数及其计算函数:

- $\text{norm}(V,1)$: 向量 V 的 1-范数, 计算向量元素的绝对值之和。
- $\text{norm}(V)$ 或 $\text{norm}(V,2)$: 向量 V 的 2-范数, 计算向量元素平方和的平方根。
- $\text{norm}(V,\text{inf})$: 向量 V 的 ∞ -范数, 计算向量元素绝对值中的最大值。



2.5.6 向量和矩阵的范数

2. 矩阵的范数及其计算函数:



2.5.6 向量和矩阵的范数

2. 矩阵的范数及其计算函数:

- 矩阵 A 的 1-范数: 所有矩阵列元素绝对值之和的最大值。



2.5.6 向量和矩阵的范数

2. 矩阵的范数及其计算函数:

- 矩阵 A 的 1-范数: 所有矩阵列元素绝对值之和的最大值。
- 矩阵 A 的 2-范数: $A'A$ 矩阵的最大特征值的平方根。



2.5.6 向量和矩阵的范数

2. 矩阵的范数及其计算函数:

- 矩阵 A 的 1-范数: 所有矩阵列元素绝对值之和的最大值。
- 矩阵 A 的 2-范数: $A'A$ 矩阵的最大特征值的平方根。
- 矩阵 A 的 ∞ -范数: 所有矩阵行元素绝对值之和的最大值。



2.5.6 向量和矩阵的范数

2. 矩阵的范数及其计算函数:

- 矩阵 A 的 1-范数: 所有矩阵列元素绝对值之和的最大值。
- 矩阵 A 的 2-范数: $A'A$ 矩阵的最大特征值的平方根。
- 矩阵 A 的 ∞ -范数: 所有矩阵行元素绝对值之和的最大值。
- 矩阵 A 的 Frobenius-范数: 所有矩阵元素平方和的平方根。



2.5.7 矩阵的条件数

计算矩阵 A 的 3 种条件数 (condition number) 的函数是:



2.5.7 矩阵的条件数

计算矩阵 A 的 3 种条件数 (condition number) 的函数是:

- `cond(A,1)`: 计算 A 的 1-范数下的条件数。



2.5.7 矩阵的条件数

计算矩阵 A 的 3 种条件数 (condition number) 的函数是:

- $\text{cond}(A,1)$: 计算 A 的 1-范数下的条件数。
- $\text{cond}(A)$ 或 $\text{cond}(A,2)$: 计算 A 的 2-范数下的条件数。



2.5.7 矩阵的条件数

计算矩阵 A 的 3 种条件数 (condition number) 的函数是:

- `cond(A,1)`: 计算 A 的 1-范数下的条件数。
- `cond(A)` 或 `cond(A,2)`: 计算 A 的 2-范数下的条件数。
- `cond(A,inf)`: 计算 A 的 ∞ -范数下的条件数。



2.5.8 矩阵的特征值与特征向量

计算矩阵 A 的特征值和特征向量的函数是 $\text{eig}(A)$ ，格式有 3 种：



2.5.8 矩阵的特征值与特征向量

计算矩阵 A 的特征值和特征向量的函数是 $\text{eig}(A)$ ，格式有 3 种：

- $E=\text{eig}(A)$: 求矩阵 A 的全部特征值，构成向量 E 。



2.5.8 矩阵的特征值与特征向量

计算矩阵 A 的特征值和特征向量的函数是 $\text{eig}(A)$ ，格式有 3 种：

- $E=\text{eig}(A)$ ：求矩阵 A 的全部特征值，构成向量 E 。
- $[V,D]=\text{eig}(A)$ ：求矩阵 A 的全部特征值，构成对角阵 D ，并求 A 的特征向量构成 V 的列向量。



2.5.8 矩阵的特征值与特征向量

计算矩阵 A 的特征值和特征向量的函数是 $\text{eig}(A)$ ，格式有 3 种：

- $E=\text{eig}(A)$ ：求矩阵 A 的全部特征值，构成向量 E 。
- $[V,D]=\text{eig}(A)$ ：求矩阵 A 的全部特征值，构成对角阵 D ，并求 A 的特征向量构成 V 的列向量。
- $[V,D]=\text{eig}(A, 'nobalance')$ ：与第 2 种格式类似，但第 2 种格式中先对 A 作相似变换后求矩阵 A 的特征值和特征向量，而格式 3 直接求矩阵 A 的特征值和特征向量。



2.5.8 矩阵的特征值与特征向量

例 2-11: 用求特征值的方法解方程。

$$3x^5 - 7x^4 + 5x^2 + 2x - 18 = 0$$



2.5.8 矩阵的特征值与特征向量

例 2-11: 用求特征值的方法解方程。

$$3x^5 - 7x^4 + 5x^2 + 2x - 18 = 0$$

```
>> p=[3,-7,0,5,2,-18];  
  
>> A=compan(p);      % A 的伴随矩阵  
  
>> x1=eig(A)         % 求 A 的特征值  
  
>> x2=roots(p)       % 直接求多项式 p 的零点
```



矩阵的超越函数

- 矩阵平方根 `sqrtm`: `sqrtm(A)` 计算矩阵 A 的平方根。



矩阵的超越函数

- 矩阵平方根 `sqrtm`: `sqrtm(A)` 计算矩阵 A 的平方根。
- 矩阵对数 `logm`: `logm(A)` 计算矩阵 A 的自然对数。此函数输入参数的条件与输出结果间的关系和函数 `sqrtm(A)` 完全一样。



矩阵的超越函数

- 矩阵平方根 `sqrtm`: `sqrtm(A)` 计算矩阵 A 的平方根。
- 矩阵对数 `logm`: `logm(A)` 计算矩阵 A 的自然对数。此函数输入参数的条件与输出结果间的关系和函数 `sqrtm(A)` 完全一样。
- 矩阵指数 `expm`: `expm(A)` 的功能是求矩阵指数 e^A 。例如，对上面计算所得到的 A 的自然对数 L ，求其矩阵指数 $B=e^L$ ，命令如下：
 $B=\text{expm}(L)$



矩阵的超越函数

- 矩阵平方根 `sqrtm`: `sqrtm(A)` 计算矩阵 A 的平方根。
- 矩阵对数 `logm`: `logm(A)` 计算矩阵 A 的自然对数。此函数输入参数的条件与输出结果间的关系和函数 `sqrtm(A)` 完全一样。
- 矩阵指数 `expm`: `expm(A)` 的功能是求矩阵指数 e^A 。例如，对上面计算所得到的 A 的自然对数 L ，求其矩阵指数 $B=e^L$ ，命令如下：
 $B=\text{expm}(L)$
- 普通矩阵函数 `funm`: `funm(A,@fun)` 对方阵 A 计算由 `fun` 定义的函数的矩阵函数值。求矩阵的平方根只能用 `sqrtm` 函数。



2.7.1 字符串的表示

- 在 MATLAB 中，字符串是用单撇号括起来的字符序列。



2.7.1 字符串的表示

- 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。



2.7.1 字符串的表示

- 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。



2.7.1 字符串的表示

例 2-12: 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, 其余不变。
- (4) 统计字符串中小写字母的个数。



2.7.1 字符串的表示

例 2-12: 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, 其余不变。
- (4) 统计字符串中小写字母的个数。

```
>> 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
```



2.7.2 字符串的操作

1. 字符串的执行:

与字符串有关的一个重要函数是 `eval`，它的作用是把字符串的内容作为对应的 MATLAB 命令来执行，其调用格式为：

`eval(s)`

其中 `s` 为字符串。



2.7.2 字符串的操作

1. 字符串的执行:

与字符串有关的一个重要函数是 `eval`，它的作用是把字符串的内容作为对应的 MATLAB 命令来执行，其调用格式为：

`eval(s)`

其中 `s` 为字符串。

2. 字符串与数值之间的转换:

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。



2.7.2 字符串的操作

3. 字符串的连接:

在 MATLAB 中, 要将两个字符串连接在一起, 有两种常见方法: 一是用字符串向量, 即用中括号将若干个字符串括起来, 从而得到一个更长的字符串; 二是用 `strcat` 函数。



2.7.2 字符串的操作

3. 字符串的连接:

在 MATLAB 中, 要将两个字符串连接在一起, 有两种常见方法: 一是用字符串向量, 即用中括号将若干个字符串括起来, 从而得到一个更长的字符串; 二是用 `strcat` 函数。

4. 字符串的比较:

字符串的比较有两种方法: 当两个字符串拥有相同的长度时, 可以利用关系运算符对字符串进行比较, 比较的规则是按 ASCII 值大小逐个字符进行比较, 比较的结果是一个数值向量, 其元素为对应字符比较的结果; 字符串比较函数用于判断字符串是否相等, 有 4 种比较方式。



2.7.2 字符串的操作

5. 字符串的查找与替换:

MATLAB 提供了许多函数，用来对字符串中的字符进行查找与替换。常用的有以下两个。

- `findstr(s1,s2)`: 返回短字符串在长字符串中的开始位置。
- `strrep(s1,s2,s3)`: 将字符串 `s1` 中的所有子字符串 `s2` 替换为 `s3`。



2.8.1 结构数据

1. 结构矩阵的建立与引用:



2.8.1 结构数据

1. 结构矩阵的建立与引用:

- 结构矩阵的元素可以是不同的数据类型，它可将一组具有不同属性的数据纳入到一个统一的变量名下进行管理。



2.8.1 结构数据

1. 结构矩阵的建立与引用:

- 结构矩阵的元素可以是不同的数据类型，它可将一组具有不同属性的数据纳入到一个统一的变量名下进行管理。
- 建立一个结构矩阵可采用给结构成员赋值的办法。具体格式为：
结构矩阵名.成员名=表达式
其中，表达式应理解为矩阵表达式。



2.8.1 结构数据

2. 结构成员的修改:



2.8.1 结构数据

2. 结构成员的修改:

- 要给结构矩阵 **a** 增加一个成员 **x4**, 可给 **a** 中任一元素增加成员 **x4**:

```
>> a(1).x4='410075';
```

但其他成员均为空矩阵, 可以使用赋值语句给它赋确定的值。



2.8.1 结构数据

2. 结构成员的修改:

- 要给结构矩阵 **a** 增加一个成员 **x4**, 可给 **a** 中任一元素增加成员 **x4**:

```
>> a(1).x4='410075';
```

但其他成员均为空矩阵, 可以使用赋值语句给它赋确定的值。

- 要删除结构的成员, 则可以使用 **rmfield** 函数来完成。例如, 删除成员 **x4**:

```
>> a=rmfield(a,'x4');
```



2.8.2 单元数据

- 建立单元矩阵和一般矩阵相似，只是矩阵元素用大括号括起来。



2.8.2 单元数据

- 建立单元矩阵和一般矩阵相似，只是矩阵元素用大括号括起来。
- 可以用带有大括号下标的形式引用单元矩阵元素。例如 `b{3,3}`。单元矩阵的元素可以是结构或单元数据。



2.8.2 单元数据

- 建立单元矩阵和一般矩阵相似，只是矩阵元素用大括号括起来。
- 可以用带有大括号下标的形式引用单元矩阵元素。例如 `b{3,3}`。单元矩阵的元素可以是结构或单元数据。
- 可以使用 `celldisp` 函数来显示整个单元矩阵，如 `celldisp(b)`。还可以删除单元矩阵中的某个元素。



2.9.1 矩阵存储方式



2.9.1 矩阵存储方式

- 完全存储方式：将矩阵的全部元素按列存储。以前讲到的矩阵的存储方式都是按这个方式存储的，此存储方式对稀疏矩阵也适用。



2.9.1 矩阵存储方式

- 完全存储方式：将矩阵的全部元素按列存储。以前讲到的矩阵的存储方式都是按这个方式存储的，此存储方式对稀疏矩阵也适用。
- 稀疏存储方式：稀疏存储方式仅存储矩阵所有的非零元素的值及其位置，即行号和列号。在 MATLAB 中，稀疏存储方式也是按列存储的。



2.9.1 矩阵存储方式

- 完全存储方式：将矩阵的全部元素按列存储。以前讲到的矩阵的存储方式都是按这个方式存储的，此存储方式对稀疏矩阵也适用。
- 稀疏存储方式：稀疏存储方式仅存储矩阵所有的非零元素的值及其位置，即行号和列号。在 MATLAB 中，稀疏存储方式也是按列存储的。

注意，在讲稀疏矩阵时，有两个不同的概念，一是指矩阵的 0 元素较多，该矩阵是一个具有稀疏特征的矩阵，二是指采用稀疏方式存储的矩阵。



2.9.2 稀疏存储方式的产生

1. 将完全存储方式转化为稀疏存储方式:



2.9.2 稀疏存储方式的产生

1. 将完全存储方式转化为稀疏存储方式:

- 函数 $A = \text{sparse}(S)$ 将矩阵 S 转化为稀疏存储方式的矩阵 A 。



2.9.2 稀疏存储方式的产生

1. 将完全存储方式转化为稀疏存储方式:

- 函数 $A = \text{sparse}(S)$ 将矩阵 S 转化为稀疏存储方式的矩阵 A 。
- $\text{sparse}(m,n)$: 生成一个 $m \times n$ 的所有元素都是 0 的稀疏矩阵。



2.9.2 稀疏存储方式的产生

1. 将完全存储方式转化为稀疏存储方式:

- 函数 $A = \text{sparse}(S)$ 将矩阵 S 转化为稀疏存储方式的矩阵 A 。
- $\text{sparse}(m,n)$: 生成一个 $m \times n$ 的所有元素都是 0 的稀疏矩阵。
- $\text{sparse}(u,v,S)$: u 、 v 、 S 是 3 个等长的向量。 S 是要建立的稀疏矩阵的非 0 元素, $u(i)$ 、 $v(i)$ 分别是 $S(i)$ 的行和列下标, 该函数建立一个 $\max(u)$ 行、 $\max(v)$ 列并以 S 为稀疏元素的稀疏矩阵。



2.9.2 稀疏存储方式的产生

1. 将完全存储方式转化为稀疏存储方式:

- 函数 $A = \text{sparse}(S)$ 将矩阵 S 转化为稀疏存储方式的矩阵 A 。
- $\text{sparse}(m,n)$: 生成一个 $m \times n$ 的所有元素都是 0 的稀疏矩阵。
- $\text{sparse}(u,v,S)$: u 、 v 、 S 是 3 个等长的向量。 S 是要建立的稀疏矩阵的非 0 元素， $u(i)$ 、 $v(i)$ 分别是 $S(i)$ 的行和列下标，该函数建立一个 $\max(u)$ 行、 $\max(v)$ 列并以 S 为稀疏元素的稀疏矩阵。
- $[u,v,S] = \text{find}(A)$: 返回矩阵 A 中非 0 元素的下标和元素。这里产生的 u 、 v 、 S 可作为 $\text{sparse}(u,v,S)$ 的参数。



2.9.2 稀疏存储方式的产生

1. 将完全存储方式转化为稀疏存储方式:

- 函数 $A = \text{sparse}(S)$ 将矩阵 S 转化为稀疏存储方式的矩阵 A 。
- $\text{sparse}(m,n)$: 生成一个 $m \times n$ 的所有元素都是 0 的稀疏矩阵。
- $\text{sparse}(u,v,S)$: u 、 v 、 S 是 3 个等长的向量。 S 是要建立的稀疏矩阵的非 0 元素, $u(i)$ 、 $v(i)$ 分别是 $S(i)$ 的行和列下标, 该函数建立一个 $\max(u)$ 行、 $\max(v)$ 列并以 S 为稀疏元素的稀疏矩阵。
- $[u,v,S] = \text{find}(A)$: 返回矩阵 A 中非 0 元素的下标和元素。这里产生的 u 、 v 、 S 可作为 $\text{sparse}(u,v,S)$ 的参数。
- $\text{full}(A)$: 返回和稀疏存储矩阵 A 对应的完全存储方式矩阵。



2.9.2 稀疏存储方式的产生

2. 产生稀疏存储矩阵:

只把要建立的稀疏矩阵的非 0 元素及其所在行和列的位置表示出来
后由 MATLAB 自己产生其稀疏存储, 这需要使用 `spconvert` 函数, 该函数将 A 所描述的一个稀疏矩阵转化为一个稀疏存储矩阵, 其调用格式为:

$$B = \text{spconvert}(A)$$


2.9.2 稀疏存储方式的产生

3. 带状稀疏存储矩阵:

用 `spdiags` 函数产生带状稀疏矩阵的稀疏存储, 调用格式是:

`A=spdiags(B,d,m,n)`

其中, 参数 m 、 n 为原带状矩阵的行数与列数。 B 为 $r \times p$ 阶矩阵, 这里 $r = \min(m, n)$, p 为原带状矩阵所有非零对角线的条数, 矩阵 B 的第 i 列即为原带状矩阵的第 i 条非零对角线。



2.9.2 稀疏存储方式的产生

4. 单位矩阵的稀疏存储:

单位矩阵只有对角线元素为 1，其他元素都为 0，是一种具有稀疏特征的矩阵。函数 `eye` 产生一个完全存储方式的单位矩阵。MATLAB 还有一个产生稀疏存储方式的单位矩阵的函数，这就是 `speye`。函数 `speye(m,n)` 返回一个 $m \times n$ 的稀疏存储单位矩阵。



2.9.3 稀疏矩阵应用举例

稀疏存储矩阵只是矩阵的存储方式不同，它的运算规则与普通矩阵是一样的。所以，在运算过程中，稀疏存储矩阵可以直接参与运算。当参与运算的对象不全是稀疏存储矩阵时，所得结果一般是完全存储形式。

