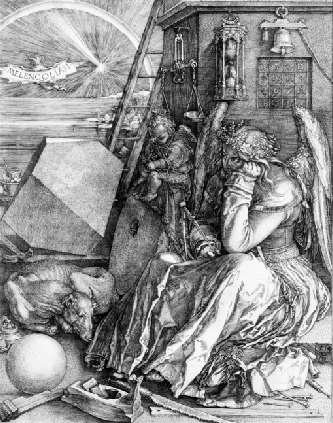
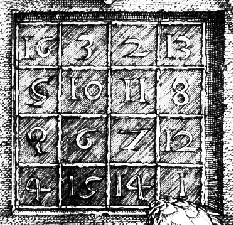
## 矩阵和幻方矩阵

### 关于矩阵

在 MATLAB® 环境中，矩阵是由数字组成的矩形数组。有时，1×1 矩阵（即标量）和只包含一行或一列的矩阵（即向量）会附加特殊含义。MATLAB 采用其他方法来存储数值数据和非数值数据，但刚开始时，通常最好将一切内容都视为矩阵。MATLAB 旨在尽可能简化运算。其他编程语言一次只能处理一个数字，而 MATLAB 允许您轻松快捷地处理整个矩阵。本手册中使用的有效示例矩阵摘自德国艺术家和业余数学家 Albrecht Dürer 在文艺复兴时期的雕刻 Melencolia I。



这幅图布满了数学符号，通过仔细观察，您会发现右上角有一个矩阵。此矩阵称为幻方矩阵，在 Dürer 所处的时代，此幻方矩阵被视为富有真正的神秘性质。它具有某些值得让人深究的迷人特征。



### 输入矩阵

开始学习 MATLAB 的最佳方法是了解如何处理矩阵。启动 MATLAB 并按照每个示例操作。

您可以采用多种不同方法在 MATLAB 中输入矩阵：

* 输入元素的明确列表。
* 从外部数据文件加载矩阵。
* 使用内置函数生成矩阵。
* 使用您自己的函数创建矩阵，并将其保存在文件中。

首先，以元素列表的形式输入丢勒的矩阵。您只需遵循一些基本约定：

* 使用空格或逗号分隔行的元素。
* 使用分号 ; 表示每行末尾。
* 使用方括号 [ ] 将整个元素列表括起来。

要输入丢勒矩阵，只需在命令行窗口中键入即可

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

MATLAB 显示刚才您输入的矩阵：

A =

16 3 2 13

5 10 11 8

9 6 7 12

4 15 14 1

此矩阵与雕刻中的数字一致。输入矩阵之后，MATLAB 工作区会自动记住此矩阵。您可以将其简称为 A。现在，您已经在工作区中输入 A，让我们看看它为什么如此有趣吧。它有什么神奇的地方呢？

### 矩阵求和、转置和对角矩阵

您可能已经注意到，幻方矩阵的特殊属性与元素的不同求和方法相关。如果沿任何行或列求和，或者沿两条主对角线中的任意一条求和，您将始终得到相同数字。让我们使用 MATLAB 来验证这一点。尝试的第一个语句是

sum(A)

MATLAB 返回的结果为

ans =

34 34 34 34

如果未指定输出变量，MATLAB 将使用变量 ans（answer 的缩略形式）来存储计算结果。您已经计算包含 A 的列总和的行向量。每个列的总和都相同，即幻数和 34。

行总和如何处理？MATLAB 会优先处理矩阵的列，因此获取行总和的一种方法是转置矩阵，计算转置的列总和，然后转置结果。

MATLAB 具有两个转置运算符。撇号运算符（例如，A'）执行复共轭转置。它会围绕主对角线翻转矩阵，并且还会更改矩阵的任何复数元素的虚部符号。点撇号运算符 (A.') 转置矩阵，但不会影响复数元素的符号。对于包含所有实数元素的矩阵，这两个运算符返回相同结果。

因此

A'

生成

ans =

16 5 9 4

3 10 6 15

2 11 7 14

13 8 12 1

而

sum(A')'

生成包含行总和的列向量

ans =

34

34

34

34

有关避免双重转置的其他方法，请在 [sum](http://ww2.mathworks.cn/help/matlab/ref/sum.html) 函数中使用维度参数：

sum(A,2)

生成

ans =

34

34

34

34

使用 [sum](http://ww2.mathworks.cn/help/matlab/ref/sum.html) 和 [diag](http://ww2.mathworks.cn/help/matlab/ref/diag.html) 函数可以获取主对角线上的元素的总和：

diag(A)

生成

ans =

16

10

7

1

而

sum(diag(A))

生成

ans =

34

从数学上讲，另一条对角线（即所谓的反对角线）并不是十分重要，因此 MATLAB 没有对此提供现成的函数。但原本用于图形的函数 [fliplr](http://ww2.mathworks.cn/help/matlab/ref/fliplr.html) 可以从左往右地翻转矩阵：

sum(diag(fliplr(A)))

ans =

34

您已经验证丢勒雕刻中的矩阵确实是一个幻方矩阵，同时在验证过程中，您已经尝试了几个 MATLAB 矩阵运算。下面各部分继续使用此矩阵来演示 MATLAB 的其他功能。

### magic 函数

MATLAB 实际包含一个内置函数，该函数可创建几乎任意大小的幻方矩阵。此函数称为 [magic](http://ww2.mathworks.cn/help/matlab/ref/magic.html) 也就不足为奇了：

B = magic(4)

B =

16 2 3 13

5 11 10 8

9 7 6 12

4 14 15 1

此矩阵几乎与丢勒雕刻中的矩阵相同，并且具有所有相同的“神奇”性质；唯一区别在于交换了中间两列。

您可以交换 B 的中间两列，使其看起来像丢勒 A。针对 B 中的每一行，按照指定顺序（1、3、2、4）对列进行重新排列：

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

A =

16 3 2 13

5 10 11 8

9 6 7 12

4 15 14 1

### 生成矩阵

MATLAB 软件提供了四个用于生成基本矩阵的函数。

|  |  |
| --- | --- |
| [zeros](http://ww2.mathworks.cn/help/matlab/ref/zeros.html) | 全部为零 |
| [ones](http://ww2.mathworks.cn/help/matlab/ref/ones.html) | 全部为 1 |
| [rand](http://ww2.mathworks.cn/help/matlab/ref/rand.html) | 均匀分布的随机元素 |
| [randn](http://ww2.mathworks.cn/help/matlab/ref/randn.html) | 正态分布的随机元素 |

下面给出了一些示例：

Z = zeros(2,4)

Z =

0 0 0 0

0 0 0 0

F = 5\*ones(3,3)

F =

5 5 5

5 5 5

5 5 5

N = fix(10\*rand(1,10))

N =

9 2 6 4 8 7 4 0 8 4

R = randn(4,4)

R =

0.6353 0.0860 -0.3210 -1.2316

-0.6014 -2.0046 1.2366 1.0556

0.5512 -0.4931 -0.6313 -0.1132

-1.0998 0.4620 -2.3252 0.3792

**MATLAB矩阵的创建**

**MATLAB矩阵的创建**

MATLAB的矩阵创建有很多种方法，但总的原则是不变的。原则如下：

a、矩阵元素必须在”[ ]”内（用步长创建的单行向量除外）；

b、矩阵的同行元素之间用空格或”,”隔开；

c、矩阵的行与行之间用”;”（或回车符）隔开；

d、矩阵的元素可以是数值、变量、表达式或函数，当然也可以是矩阵本身。如果是矩阵本身，则可看做是矩阵的粘合；

e、矩阵的尺寸不必预先定义。

举例如下：

a = [1 2 3; 4 5 6]

a = 1:10

a = [1:10; 11:20]

a = [1:10; 11:20]’

a = [1:0.5:5.5; 5.5:0.5:10]

a = [1 2 3

4 5 6

7 8 9]

b = [10 11 12; 13 14 15; 16 17 18]

A = [a b]

A = [a; b]

因为转置’是一种运算，所以a = 1:20’是合法的，但是等价于a = 1:20，相当于对20这个只有一个元素的矩阵转置，然后带入到1:20中，所以a = (1:20)’与a = [1:20]’都是可以的，但是显然a = (1:20; 21:40)’是非法的。

对于矩阵元素的选取，总的语法是(行,列)，所以，取A的3行2列元素可以是A(3,2)。

同时可以使用单行矩阵（即向量）的形式来选取子矩阵。例如，选取矩阵A的1,3,5行和2,3列组成子矩阵，可以使用A([1 3 5],[2 3])。这样，显然地，可以使用步长的方法，如选取A的3到6行和1到3列就可以写成A(3:6,1:3)。同时，可以使用单独的:来表示全部。如取A的3、5、7行和所有列，可以写成A(3:2:7,:)。

最后，可以使用MATLAB提供的函数创建特殊矩阵，常用特殊函数如下：

**magic(n)**创建n维魔方矩阵。

**vander(v)**生成以向量V为基础向量的范得蒙矩阵。

**rand(n)** n阶随机矩阵。可以试试randn(m,n)。

**hilb(n)** n阶希尔伯特矩阵。

**pascal(n)** n阶帕斯卡矩阵。

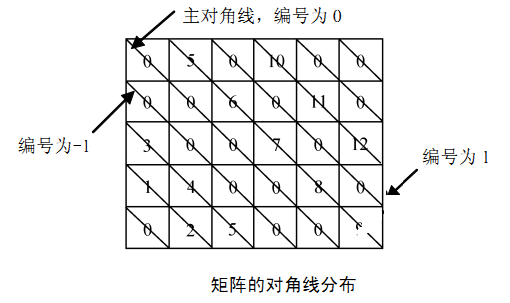
* 概念： 当一个矩阵中只含一部分非零元素，而其余均为“0”元素时，我们称这一类矩阵为稀疏矩阵（sparse matrix）。
* 在实际问题中，相当一部分的线性方程组的系数矩阵是大型稀疏矩阵，而且非零元素在矩阵中的位置表现得很有规律。
* 建立稀疏矩阵目的:
* 节省存储空间和计算时间，提高工作效率 。
* 以 sparse 创建稀疏矩阵

其语法为

* (1) S = sparse(A)：将一个满矩阵A转化为一个稀疏矩阵S。若S本身就是一个稀疏矩阵，则sparse(S)返回S。
* (2) S = sparse(i,j,s,m,n)：在第i 行、第j 列输入数值s，矩阵共 m行n 列，输出S为一个稀疏矩阵，给出(i,j)及s。

(3) S = sparse(i,j,s)：比较简单的格式，只输入非零元的数据s 以及各非零元的行下标i和列下标j

* 例3.28将满矩阵A转化为一个稀疏矩阵。
* >> A=[1 2 0;0 2 3;1 0 2];
* >> S=sparse(A)
* S =
* (1,1) 1
* (3,1) 1
* (1,2) 2
* (2,2) 2
* (2,3) 3
* (3,3) 2
* 这是特殊的稀疏矩阵存储方式，它的特点是所占内存少，运算速度快。如果想得到矩阵的全元素存储方式，可用下面的语句：
* >>B = full(S)
* B =
* 1 2 0
* 0 2 3
* 1 0 2



diag(v,k)  
以向量v的元素作为矩阵X的第k条对角线元素，当k=0时，v为X的[主对角线](https://www.baidu.com/s?wd=%E4%B8%BB%E5%AF%B9%E8%A7%92%E7%BA%BF&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1Y3rAP9PAPWrjT3uAFhPWFb0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnWb3njn4n1f3" \t "_blank)；当k>0时，v为上方第k条对角线；当k<0时，v为下方第k条对角线。  
例：  
>> v=[1 2 3];  
>> x=diag(v,-1)  
x =  
0 0 0 0  
1 0 0 0  
0 2 0 0  
0 0 3 0  
下面不知是否是你想要的：  
>> clear  
>> v(1)=1;  
>> n=5;%可以安自己需求  
>> v(2:n)=2;  
>> x=diag(v,-1)  
  
x =  
  
0 0 0 0 0 0  
1 0 0 0 0 0  
0 2 0 0 0 0  
0 0 2 0 0 0  
0 0 0 2 0 0  
0 0 0 0 2 0

创建[对角矩阵](https://www.baidu.com/s?wd=%E5%AF%B9%E8%A7%92%E7%9F%A9%E9%98%B5&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1Y3PAfYmHN-n1TvPvfkPvfz0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EPjD3njn3n1fd" \t "_blank)对角元素是1 2 3 4 5 执行如下  
>> C=diag(1:5)  
  
C =  
1 0 0 0 0  
0 2 0 0 0  
0 0 3 0 0  
0 0 0 4 0  
0 0 0 0 5

取出a阵的对角元，然后构建一个以a对角元为对角的[对角矩阵](https://www.baidu.com/s?wd=%E5%AF%B9%E8%A7%92%E7%9F%A9%E9%98%B5&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1Y4uHwbuWIBnW63nyF9uyc40ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3En1RLrjm1rjfd" \t "_blank)。  
A =  
  
1 2   
3 4   
  
>> diag(diag(A))  
  
ans =  
  
1 0

## 索引

### 下标

A 的行 i 和列 j 中的元素通过 A(i,j) 表示。例如，A(4,2) 表示第四行和第二列中的数字。在幻方矩阵中，A(4,2) 为 15。因此，要计算 A 第四列中的元素的总和，请键入

A(1,4) + A(2,4) + A(3,4) + A(4,4)

此下标生成

ans =

34

但这不是计算某列总和的最佳方法。

此外，还可以使用单一下标 A(k) 引用矩阵的元素。单一下标是引用行和列向量的常见方法。但是，也可以对满二维矩阵应用单一下标。在这种情况下，数组被视为一个由原始矩阵的列构成的长列向量。因此，在幻方矩阵中，A(8) 是另一种引用存储在 A(4,2) 中的值 15 的方法。

如果尝试使用矩阵外部元素的值，则会生成错误：

t = A(4,5)

索引超出矩阵维度。

相反，如果将值存储在矩阵外部元素中，则会增大大小以便容纳新元素：

X = A;

X(4,5) = 17

X =

16 3 2 13 0

5 10 11 8 0

9 6 7 12 0

4 15 14 1 17

### 冒号运算符

冒号 : 是最重要的 MATLAB® 运算符之一。它以多种不同形式出现。表达式

1:10

是包含从 1 到 10 之间的整数的行向量：

1 2 3 4 5 6 7 8 9 10

要获取非单位间距，请指定增量。例如，

100:-7:50

为

100 93 86 79 72 65 58 51

而

0:pi/4:pi

为

0 0.7854 1.5708 2.3562 3.1416

包含冒号的下标表达式引用部分矩阵：

A(1:k,j)

表示 A 第 j 列中的前 k 个元素。因此，

sum(A(1:4,4))

计算第四列的总和。但是，执行此计算有一种更好的方法。冒号本身引用矩阵行或列中的所有元素，而关键字 [end](http://ww2.mathworks.cn/help/matlab/ref/end.html) 引用最后一个行或列。因此，

sum(A(:,end))

计算 A 最后一列中的元素的总和：

ans =

34

为什么 4×4 幻方矩阵的幻数和等于 34？如果将介于 1 到 16 之间的整数分为四个总和相等的组，该总和必须为

sum(1:16)/4

当然，也即

ans =

34

### 串联

串联是连接小矩阵以便形成更大矩阵的过程。实际上，第一个矩阵是通过将其各个元素串联起来而构成的。成对的方括号 [] 即为串联运算符。例如，从 4×4 幻方矩阵 A 开始，组成

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

结果会生成一个 8×8 矩阵，这是通过连接四个子矩阵获得的：

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

此矩阵是一个接近于幻方矩阵的矩阵。此矩阵的元素是经过重新排列的整数 1:64。此矩阵的列总和符合 8×8 幻方矩阵的要求：

sum(B)

ans =

260 260 260 260 260 260 260 260

但是其行总和 sum(B')' 并不完全相同。要使其成为有效的 8×8 幻方矩阵，需要进行进一步操作。

### 删除行和列

只需使用一对方括号即可从矩阵中删除行和列。首先

X = A;

然后，要删除 X 的第二列，请使用

X(:,2) = []

这会将 X 更改为

X =

16 2 13

5 11 8

9 7 12

4 14 1

如果您删除矩阵中的单个元素，结果将不再是矩阵。因此，以下类似表达式

X(1,2) = []

将会导致错误。但是，使用单一下标可以删除一个元素或元素序列，并将其余元素重构为一个行向量。因此

X(2:2:10) = []

生成

X =

16 9 2 7 13 12 1

### 标量扩展

可以采用多种不同方法将矩阵和标量合并在一起。例如，通过从每个元素中减去标量而将其从矩阵中减去。幻方矩阵的元素平均值为 8.5，因此

B = A - 8.5

形成一个列总和为零的矩阵：

B =

7.5 -5.5 -6.5 4.5

-3.5 1.5 2.5 -0.5

0.5 -2.5 -1.5 3.5

-4.5 6.5 5.5 -7.5

sum(B)

ans =

0 0 0 0

通过标量扩展，MATLAB 会为范围中的所有索引分配一个指定标量。例如，

B(1:2,2:3) = 0

将 B 的某个部分清零：

B =

7.5 0 0 4.5

-3.5 0 0 -0.5

0.5 -2.5 -1.5 3.5

-4.5 6.5 5.5 -7.5

### 逻辑下标

根据逻辑和关系运算创建的逻辑向量可用于引用子数组。假定 X 是一个普通矩阵，L 是一个由某个逻辑运算生成的同等大小的矩阵。那么，X(L) 指定 X 的元素，其中 L 的元素为非零。

通过将逻辑运算指定为下标表达式，可以在一个步骤中完成这种下标。假定您具有以下数据集：

x = [2.1 1.7 1.6 1.5 NaN 1.9 1.8 1.5 5.1 1.8 1.4 2.2 1.6 1.8];

[NaN](http://ww2.mathworks.cn/help/matlab/ref/nan.html) 是用于缺少的观测值的标记，例如，无法响应问卷中的某个项。要使用逻辑索引删除缺少的数据，请使用 isfinite(x)，对于所有有限数值，该函数为 true；对于 NaN 和 Inf，该函数为 false：

x = x(isfinite(x))

x =

2.1 1.7 1.6 1.5 1.9 1.8 1.5 5.1 1.8 1.4 2.2 1.6 1.8

现在，存在一个似乎与其他项很不一样的观测值，即 5.1。这是一个离群值。下面的语句可删除离群值，在本示例中，即比均值大三个标准差的元素：

x = x(abs(x-mean(x)) <= 3\*std(x))

x =

2.1 1.7 1.6 1.5 1.9 1.8 1.5 1.8 1.4 2.2 1.6 1.8

标量扩展对于另一示例，请使用逻辑索引和标量扩展将非质数设置为 0，以便高亮显示丢勒幻方矩阵中的质数的位置。（请参阅 [magic 函数](http://ww2.mathworks.cn/help/matlab/learn_matlab/matrices-and-magic-squares.html#f2-484)。）

A(~isprime(A)) = 0

A =

0 3 2 13

5 0 11 0

0 0 7 0

0 0 0 0

### find 函数

[find](http://ww2.mathworks.cn/help/matlab/ref/find.html) 函数可用于确定与指定逻辑条件相符的数组元素的索引。find 以最简单的形式返回索引的列向量。转置该向量以便获取索引的行向量。例如，再次从丢勒的幻方矩阵开始。（请参阅 [magic 函数](http://ww2.mathworks.cn/help/matlab/learn_matlab/matrices-and-magic-squares.html#f2-484)。）

k = find(isprime(A))'

使用一维索引选取幻方矩阵中的质数的位置：

k =

2 5 9 10 11 13

使用以下命令按 k 确定的顺序将这些质数显示为行向量

A(k)

ans =

5 3 2 11 7 13

将 k 用作赋值语句的左侧索引时，会保留矩阵结构：

A(k) = NaN

A =

16 NaN NaN NaN

NaN 10 NaN 8

9 6 NaN 12

4 15 14 1

## 函数简介

语法格式:

n = numel(A)

返回[数组](https://baike.so.com/doc/5545345-5760453.html" \t "_blank)A中元素个数。

n = numel(A, index1, index2, ... indexn)

返回A(index1, index2, ... indexn)中元素的个数，其中indexi可以是切片运算、[算术表达式](https://baike.so.com/doc/3711933-3900615.html" \t "_blank)、逻辑表达式等。

当一个[表达式](https://baike.so.com/doc/5777526-5990305.html" \t "_blank)产生一个由逗号隔开的列表(包括形如A{index1, index2, ..., indexn}这种大括号括起来的索引列表，或者使用成员操作符进行[结构体](https://baike.so.com/doc/6031580-6244581.html)成员访问)，MATLAB软件就会[隐式调用](https://baike.so.com/doc/1956956-2070998.html" \t "_blank)numel内建函数。

相关函数:size,[length](https://baike.so.com/doc/6084374-6297476.html)

## [折叠](https://baike.so.com/doc/3377139-3555338.html)[编辑本段](https://baike.so.com/create/edit/?eid=3377139&sid=3555338&secid=2)程序示例

>> a = rand(4);

>> a(:, :, 2) = a'

a(:,:,1) =

0.1190 0.5853 0.5060 0.5472

0.4984 0.2238 0.6991 0.1386

0.9597 0.7513 0.8909 0.1493

0.3404 0.2551 0.9593 0.2575

a(:,:,2) =

0.1190 0.4984 0.9597 0.3404

0.5853 0.2238 0.7513 0.2551

0.5060 0.6991 0.8909 0.9593

0.5472 0.1386 0.1493 0.2575

>> numel(a)

ans = 32

>> numel(a, a > 0.15)

ans = 26

>> numel(a, a > 0.8)

ans = 6

>> numel(a, 1:3, 2:4, :)

ans = 18

折叠size简介  
函数功能：返回数组的尺寸  
语法格式：  
d = size(X)  
返回数组X每一维的大小，其中d是一个向量，元素个数为ndims(X)。如果X是一个标量（在Matlab也就是一个1行乘以1列的数组），size(X)将返回[1 1]。  
[m,n] = size(X)  
返回矩阵X的尺寸信息， 并存储在m、n中。其中m中存储的是行数，n中存储的是列数。  
m = size(X,dim)  
返回X的第dim维的尺寸。  
[d1,d2,d3,...,dn] = size(X),  
当n大于1时， 返回X各维的尺寸， 并存储在变量d1、d2、……、dn中。 其中n（输出参数的个数）必须等于ndims(X)。如果不等于，则：  
n < ndims(X)时， 对于大于等于1且小于n的i， di等于X第i维的尺寸； dn等于X的其他剩余维数的尺寸的乘积。  
n > ndims(X)时， 多出来的输出参数等于1。  
[相关函数](https://www.baidu.com/s?wd=%E7%9B%B8%E5%85%B3%E5%87%BD%E6%95%B0&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1Y4PvczrHwWuj6vmHKbrj7B0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EPHD3PHfLPjmsrHcYPj61nW6d)：ndims、length

1、size

获取数组的行数和列数

2、length

数组长度，即行数和列数中的较大值，相当于max(size(a))

3、numel

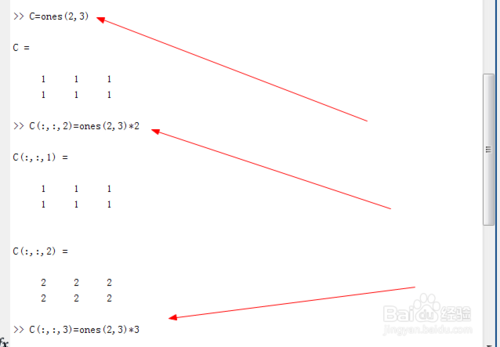
返回元素总数

**高维数组创建**

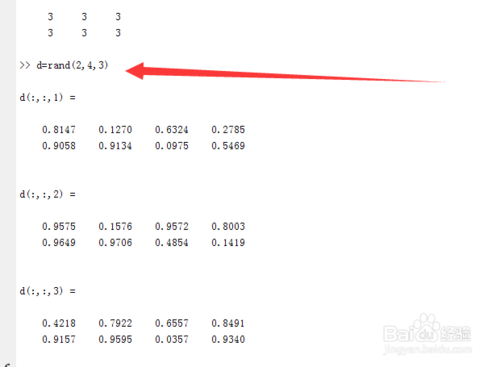
[全下标元素赋值方法](http://jingyan.baidu.com/album/922554464f6529851648f4cb.html?picindex=6" \t "_self)

[](http://jingyan.baidu.com/album/922554464f6529851648f4cb.html?picindex=2)

1. 由低维数组合成高维数组

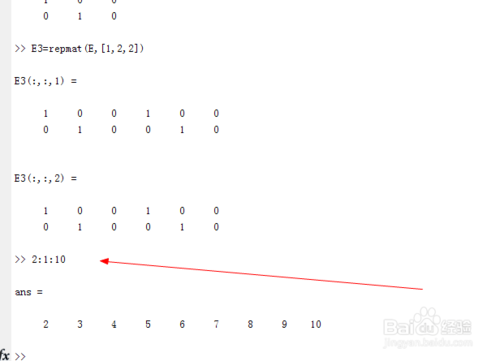
[](http://jingyan.baidu.com/album/922554464f6529851648f4cb.html?picindex=3)

1. 利用rand，randn等指令创建随机数矩阵

[](http://jingyan.baidu.com/album/922554464f6529851648f4cb.html?picindex=4)

1. 借助cat，repmat等创建

[](http://jingyan.baidu.com/album/922554464f6529851648f4cb.html?picindex=5)

[](http://jingyan.baidu.com/album/922554464f6529851648f4cb.html?picindex=6)

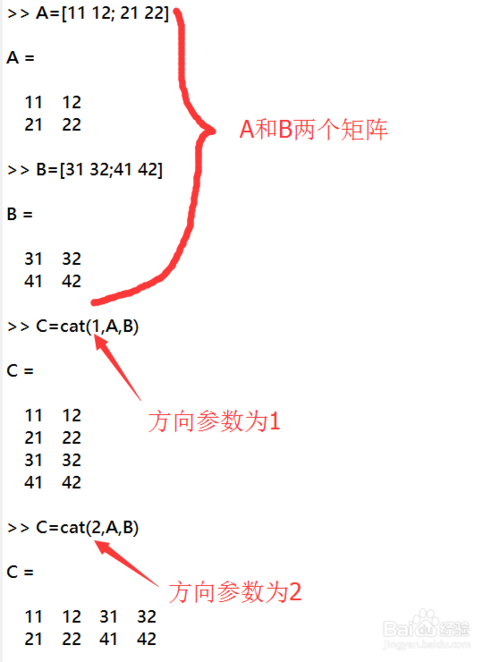
**矩阵拼接**

[A,B]相当于水平拼接A和B，即horzcat(A,B)

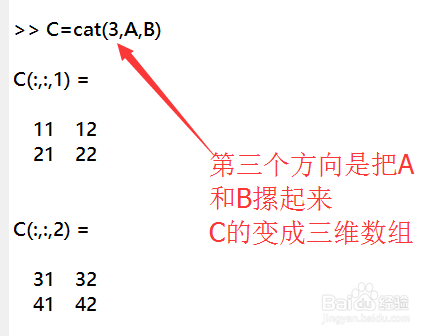
[A;B]相当于垂直拼接A和B，即vertcat(A,B)

1. 首先，我们定义A和B两个2\*2的矩阵。

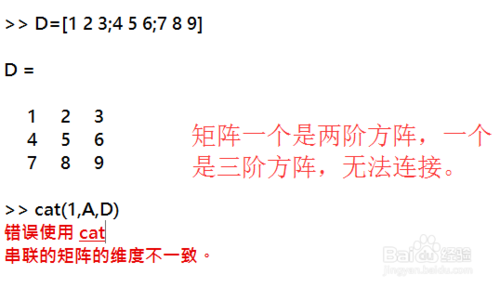
**cat(维度,A,B)**可以按照给定方向拼接矩阵。如图使用cat(1,A,B)把这两个矩阵竖着连起来。使用cat(2,A,B)把这两个矩阵横着连起来。

[](http://jingyan.baidu.com/album/9f7e7ec0e38e926f281554ff.html?picindex=1)

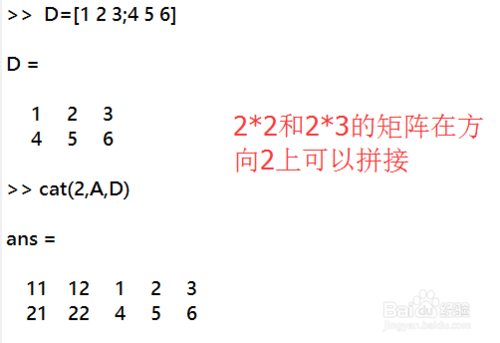
1. C=cat(3,A,B)把两个矩阵摞成一个三维数组，如图。

[](http://jingyan.baidu.com/album/9f7e7ec0e38e926f281554ff.html?picindex=2)

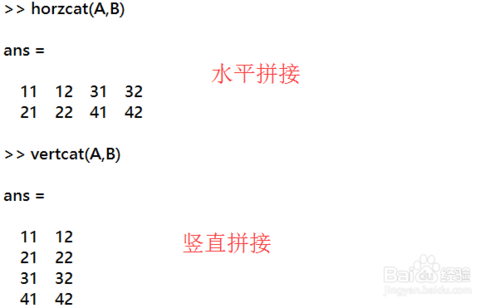
1. 使用cat要注意，在拼接方向上的宽度要一致才可拼接。

[](http://jingyan.baidu.com/album/9f7e7ec0e38e926f281554ff.html?picindex=3)

1. 如图，2\*2的矩阵和2\*3的矩阵在方向2上（水平方向）可以用cat拼接。

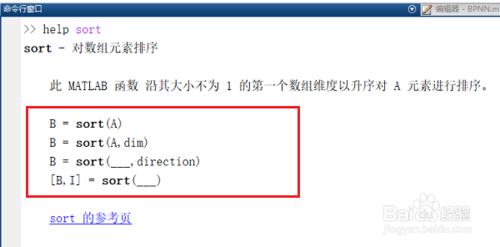
[](http://jingyan.baidu.com/album/9f7e7ec0e38e926f281554ff.html?picindex=4)

1. 使用Horzcat可以水平连接矩阵，使用vertcat可以垂直连接矩阵，相当于cat的专用版本。

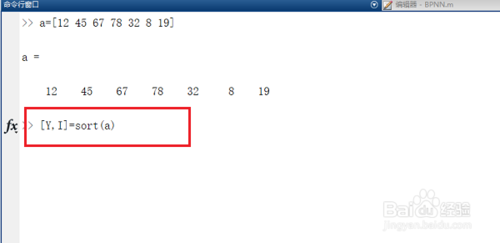
[](http://jingyan.baidu.com/album/9f7e7ec0e38e926f281554ff.html?picindex=5)

**矩阵排序**

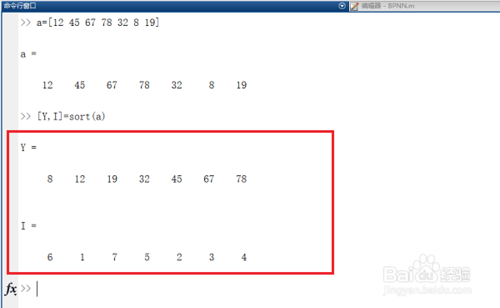
[一步我们首先需要知道matlab中对矩阵元素排序需要用到sort函数，在命令行窗口help sort可以看到函数用法，如下图所示：](http://jingyan.baidu.com/album/4e5b3e19f4ee3c91901e240b.html?picindex=6" \t "_self)

[](http://jingyan.baidu.com/album/4e5b3e19f4ee3c91901e240b.html?picindex=1)

1. 第二步我们输入a=[12 45 67 78 32 8 19];[Y,I]=sort(a),对a矩阵元素进行排序，如下图所示：

[](http://jingyan.baidu.com/album/4e5b3e19f4ee3c91901e240b.html?picindex=2)

1. 第三步按回车键得到结果之后，可以看到Y是排序好的矩阵，I是矩阵元素原先的下标位置，如下图所示：

[](http://jingyan.baidu.com/album/4e5b3e19f4ee3c91901e240b.html?picindex=3)

1. 第四步sort(a)默认是升序排序，我们也可以设置为降序排序，输入[Y,I]=sort(a,'descend')，如下图所示：

[](http://jingyan.baidu.com/album/4e5b3e19f4ee3c91901e240b.html?picindex=4)

1. 第五步按回车键可以看到排序后的矩阵Y按降序排序，从大到小，如下图所示：

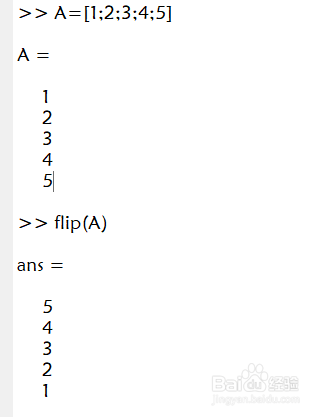
[](http://jingyan.baidu.com/album/4e5b3e19f4ee3c91901e240b.html?picindex=5)

1. 第六步我们也可以通过[Y,I]=sort(a,'ascend')来升序排序，如下图所示：

[](http://jingyan.baidu.com/album/4e5b3e19f4ee3c91901e240b.html?picindex=6)

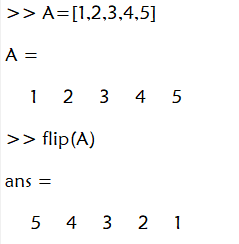
**矩阵反转**

1. 对于一个列向量，使用flip(A)可以将向量上下颠倒。如图。

[](http://jingyan.baidu.com/album/2f9b480de32ebc41cb6cc229.html?picindex=1)

1. 2

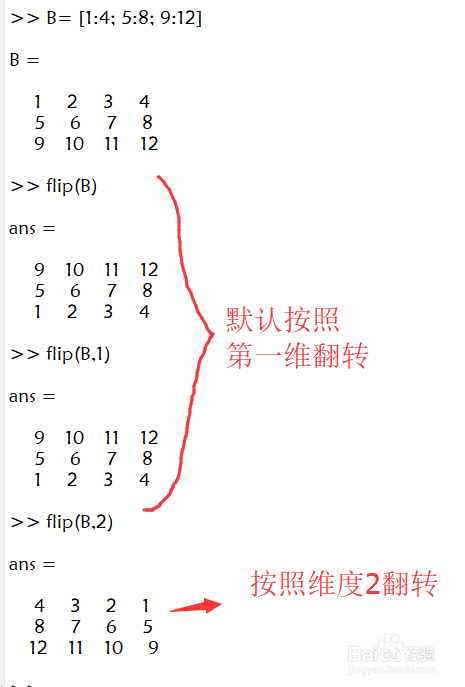
对于一个行向量，使用flip(A)默认可以将列表左右翻转。

[](http://jingyan.baidu.com/album/2f9b480de32ebc41cb6cc229.html?picindex=2)

1. 3

对于一个矩阵，使用flip函数默认沿着第一维（上下）翻转。

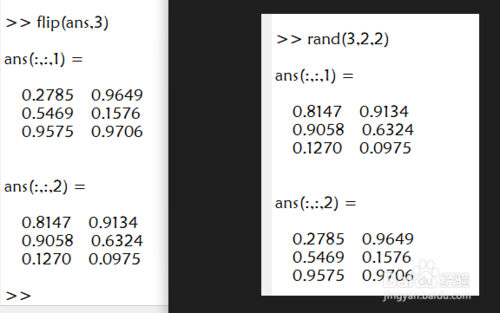
flip可以有第二个参数，翻转维度。flip(B,1)将沿着第一维（上下）翻转，flip(B,2)将沿着第二维（左右）翻转。

[](http://jingyan.baidu.com/album/2f9b480de32ebc41cb6cc229.html?picindex=3)

1. 4

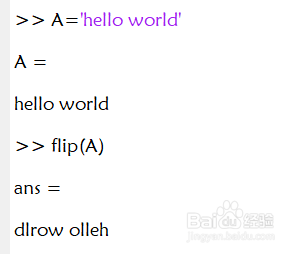
对于一个三维或者更高维数组，同样可以使用flip.

如图，把一个3\*2\*2的高维数组沿着第三维（两层）翻转。

[](http://jingyan.baidu.com/album/2f9b480de32ebc41cb6cc229.html?picindex=4)

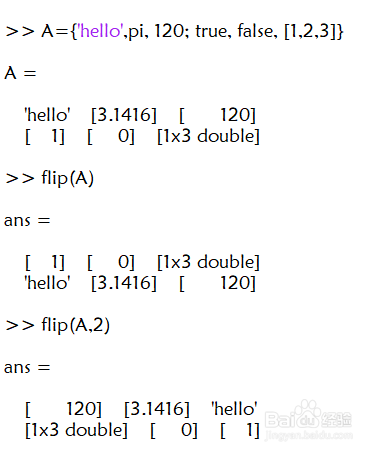
1. 5

flip还可以翻转字符串。此时flip(A)不需要带维度参数。

[](http://jingyan.baidu.com/album/2f9b480de32ebc41cb6cc229.html?picindex=5)

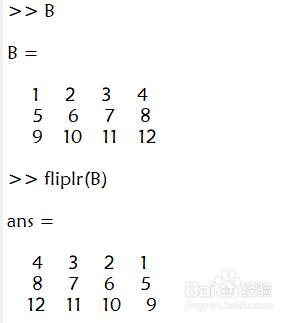
1. 6

flip还可以用于翻转元胞数组，如图，一个元胞数组里面有数字，字符串，数组等。使用flip可以将其沿着不同维度翻转。

[](http://jingyan.baidu.com/album/2f9b480de32ebc41cb6cc229.html?picindex=6)

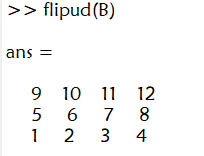
1. 7

fliplr函数可以将矩阵左右翻转(left right).

[](http://jingyan.baidu.com/album/2f9b480de32ebc41cb6cc229.html?picindex=7)

1. 8

flipud函数可以将矩阵上下翻转(up down).

[](http://jingyan.baidu.com/album/2f9b480de32ebc41cb6cc229.html?picindex=8)

**矩阵旋转**

【功能简介】矩阵旋转90°或90°的倍数。

【语法格式】

1．B=rot90(A)

将矩阵A逆时针旋转90°以后返回B。

2．B=rot90(A,k)

将矩阵A逆时针旋转（90×k）°以后返回B，k取负数时表示顺时针旋转。

对矩阵进行逆时针和顺时针旋转。

>> a=[1,2,3;4,5,6;7,8,9]

a =

     1     2     3

     4     5     6

     7     8     9

>> b=rot90(a)           %逆时针旋转90°

b =

     3     6     9

     2     5     8

     1     4     7

>> c=rot90(a,-1)        %顺时针旋转90°

c =

     7     4     1

     8     5     2

     9     6     3

---------------------

**矩阵重排**

reshape函数重新调整矩阵的行数、列数、维数。

## 1函数用法

[编辑](https://baike.sogou.com/Create.e?sp=2&sp=l71772363&sp=1)

B = reshape(A,m,n)

B = reshape(A,m,n,p,...)

B = reshape(A,[m n p ...])

B = reshape(A,...,[ ],...)

B = reshape(A,siz)

## 2函数简介

[编辑](https://baike.sogou.com/Create.e?sp=2&sp=l71772363&sp=2)

调用格式：

B = reshape(A,m,n)

返回一个m\*n的 矩阵B， B中元素是按列从A中得到的。如果A中元素个数没有m\*n个， 则会引发错误。

B = reshape(A,m,n,p,...)和B = reshape(A,[m n p ...])

返回一个和A具有相同元素的n维 数组。但B的尺寸是m\*n\*p\*...，m\*n\*p\*...必须和 prod(size(A))相等。即A和B元素个数相等。

B = reshape(A,...,[ ],...）

计算由[占位符](https://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=7812561&ss_c=ssc.citiao.link" \t "_blank)[ ]所表示的尺寸的长度，这样维度的乘积等价于prod(size(A))。而prod(size(A))的值必须能被的指定维数的乘积所整除。在这里面占位符[ ] 只能使用一次。

B = reshape(A,siz 返回一个和[A元素](https://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=10675960&ss_c=ssc.citiao.link" \t "_blank)相同的n维数组，但是由向量siz来决定重构数组维数的大小，prod(siz)的数量值必须和prod(size(A))的保持一致。

相关函数： shiftdim, squeeze, circshift, permute, repmat

## 3程序示例

[编辑](https://baike.sogou.com/Create.e?sp=2&sp=l71772363&sp=3)

close all; clear; clc;

A = [1 2 3; 4 5 6; 7 8 9; 10 11 12] % 4 by 3

B = reshape(A, 2, 6) % 2 by 6

% C = reshape(A, 2, 4) % error

% D = reshape(A, 2, 10) % error

E = reshape(A, 2, 3, 2) % 2 by 3 by 2

注意：reshape函数对原数组的抽取是按照列抽取的（对原数组按列抽取，抽取的元素填充为新数组的列）

运行结果：

A =

1 2 3

4 5 6

7 8 9

10 11 12

B =

1 7 2 8 3 9

4 10 5 11 6 12

E(:,:,1) =

1 7 2

4 10 5

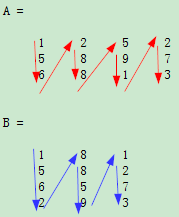
E(:,:,2) =

8 3 9

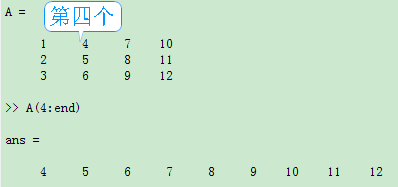
11 6 12

1）reshape函数：B=reshape（A，m,n）,将数组A中的元素重新排列进m\*n的数组中去，例子：

[MATLAB编程语句或函数介绍（1）](http://jingyan.baidu.com/album/0964eca2585f628285f536de.html?picindex=1)

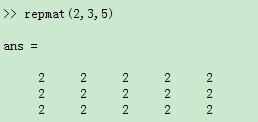
[](http://jingyan.baidu.com/album/0964eca2585f628285f536de.html?picindex=2)

1. 2）end函数返回下标的最大值 。例：A=[1 4 7 10;2 5 8 11;3 6 9 12]那么A(4:end)指的是从第4个到最后一个,举例如下：

[](http://jingyan.baidu.com/album/0964eca2585f628285f536de.html?picindex=3)

1. 3

3）repmat函数：repmat全称是Replicate Matrix ，意思是复制和平铺矩阵。D=repmat（2，m,n），将“2”复制平铺成m\*n的数组，举例如下：

[](http://jingyan.baidu.com/album/0964eca2585f628285f536de.html?picindex=4)