

MATLAB 实验报告

学号		姓名	
实验名称	MATLAB 科学计算实验		
实验目的	1、掌握数组的创建和操作方法 2、掌握常用的数值计算方法 3、掌握常用的符号计算方法		
实验方案	<p>1、一维数组创建实验</p> <p>1) 直接输入法: 测试数据: <code>b1=[1 2 3 4 5]</code></p> <p>2) 步长增长法 测试数据: <code>b2=1:2:13</code></p> <p>3) 定数线性采样的方法 (<code>linspace(a,b,n)</code>): <code>a</code> 为数组的第一个元素, <code>b</code> 为数组的最后一个元素, <code>n</code> 表示采样的点数。 测试数据: <code>b3= linspace(0,10,5)</code></p> <p>4) 定数对数采样的方法(<code>logspace(a,b,n)</code>): <code>n</code> 为数组的总个数, <code>a</code> 为第一个元素, <code>b</code> 为数组的最后一个元素。经过常用对数采样生成一维数组。 测试数据: <code>b4=logspace(-1,1,4)</code></p> <p>2、创建高维数组:</p>		

1) 直接输入的方法:

测试数据:

```
a1=[1,2,3;4,5,6]
```

2) 使用下标生成数组的方法:

测试数据:

```
a2(2,2,2)=1
```

3) 低维数生成高维数组:

```
a3=ones(2,3);a3(:,:,2)=ones(2,3)*2;a3(:,:,3)=ones(2,3)*3
```

4) 使用函数创建高维数组

a. cat 函数

cat 函数用于连接数组,标准形式为 $c=cat(dim,A1,A2,A3, \dots)$ 其中, dim 表示传见数组的维数, A1, A2, A3 表示各个维度上的数组。

测试数据:

```
a4=cat(3,ones(2,3),ones(2,3)*2,ones(2,3)*3)
```

b. reshape 函数

resharpe 用于修改数组的大小,根据用户的需求,将二维数组修改成三维数组,调用的格式为 $reshape(A[m\ n\ p])$ A 表示待重组的矩阵,后面的输入参数表示各个维度的度数。

测试数据:

```
a5=reshape(1:12,2,2,3)
```

c. repmat 函数

repmat 表示复制数组的块数,该命令的表现形式 $repmat[A,[m\ n\ p]]$ 其中 A 表示要复制的矩阵块, mnp 分别表示在各个维度上该数组复制的块数。

测试数据:

```
a6=repmat(ones(2,3),[1,1,3])
```

3、标准数组创建实验:

1) magic (n) 随机产生 n 行 n 列的幻方。

测试数据:

```
c1=magic(3)
```

2) ones(n) 创建 n 行 n 列矩阵所有元素为 1 的矩阵。

测试数据:

```
c2=ones(2,3)
```

3) eye(n) 创建的 n 行矩阵中主对角线为 1 的单位阵。

测试数据:

```
c3=eye(3,3)
```

4) randn(n,n) 随机生成一个 n 行 n 列的矩阵

测试数据:

```
c4=randn(2,3)
```

5) zeros(n,n) 创建 n 行 n 列矩阵元素全为 0 的矩阵, 即零矩阵。

测试数据:

```
c5=zeros(2,4)
```

4. 数组变换实验部分

1) diag 用于选取矩阵对角线的数据, 和创建对角阵。

```
a=[1,2,3,4;5,6,7,8;9,10,11,12]
```

```
a1=diag(a,0)
```

```
a11=diag([1,2,-1,4])
```

2) ' 转置函数, 将数组转置

```
a12=a'
```

3) fliplr 函数根据矩阵的中线左右翻转矩阵中的对称元素。

```
a2=fliplr(a)
```

4) flipud 函数根据矩阵的水平中线上下翻转矩阵中的对称位置的元素。

```
a3=flipud(a)
```

5) rot90 函数 90 度逆时针旋转二维数组中的元素

```
a4=rot90(a)
```

6) tril (a, k)

```
a5=tril(a)
```

```
a6=triu(a,1)
```

5. 字符串数组的创建和操作实验

1) 字符串的创建

a. 直接输入的方法创建字符串数组

在直接创建数组的时候，可以直接用单引号将字符串括起来作为一个字符串数组，字符窗中如果有单引号，可以利用转义的方法，在单引号前面在加一个单引号。一个字符串的数组还可以由其他的字符串数组拼接而成。

测试数据：

```
a='I'
```

```
b="'m'
```

```
c='techer'
```

```
d=[a,b,',c,',!]
```

b. 通过函数创建 `str2mat` , `strvcat`

`str2mat` (`a`, `b...`) 用于创建矩阵，其中 `a`, `b` 被连接在一起，构成一个字符串的矩阵。

`Strvcat` 用于创建字符串矩阵，创建的时候，根据每个字符串的最大长度作为生成矩阵的列数目，如果某一个字符串的字符长度小于该值，则补充空格。

测试数据：

```
con=str2mat('frist','second','a','b')
```

```
con=strvcat('frist','second','a','b')
```

2) 处理字符串中的空格

a. `deblank` 清楚字符串尾部的空格

测试数据：

```
a=['cutString ','cut String ']
```

```
deblank(a)
```

b. `strtrim` 用于清除字符串前后的空格

测试数据：

```
a=[' mat ']
```

```
strtrim(a)
```

c. `strrep` 用于替代数组中的某一个字符串

测试数据：

```
s1='a string';
```

```
s2=strep(s1,'a','two')
```

d. **strread** 用于从原始数据中读取有效的数值信息

测试数据:

```
str='<table border=5 width=100% cellspacing=0>';
```

```
[border width space]=strread(str,'%*s%*s %c %*s
```

e. **strtok** 用于将字符串分割成不同的部分

测试数据:

strtok(str1,str2) 返回第一个 **str1** 中含有 **str2** 之前的所有字符串。

```
str='abcdef';
```

```
str1='d';
```

```
str=strtok(str,str1);
```

6、架构数组的创建与操作实验:

构建数组中每一个元素具有名称和数值两个部分组成

1) 直接法构建

测试数据:

```
= 'heu';
```

```
= 'hrb';
```

```
= 'good';
```

2) 使用 **struct** 命令创建

在使用 **struct** 创建的过程中, 格式为 **struct** (字段 1, 字段 1 的值, 字段 2, 字段 2 的值...)

测试数据:

```
u=struct('name',{'hrb'},'location',{'hrb'},'quality',{'good'})
```

3) 访问数组中的数据

a. **fieldname** 用于获取建构数组的域名

fieldname 用于获取建构数组的名称, 返回数组中所有名集合的内容。

测试数据:

```
= 'heu';
```

```
= 'hrb';
```

```
= 'good';
```

```
name = fieldnames(university)
```

b. `getfield` 获取具体架构数组中的内容

`getfield` (构建名, 域名) 查找指定构建名中域名的具体值

测试数据:

```
= 'heu';
```

```
= 'hrb';
```

```
= 'good';
```

```
value = getfield(university, 'name')
```

c. `setfield` 设置具体架构数组中的内容

和上一个函数内容相同, 不过将指定域名中的具体值, 按照 `setfield` 中第三个参数的内容进行修改。

测试数据:

```
= 'heu';
```

```
= 'hrb';
```

```
= 'good';
```

```
s = setfield(university, 'name', 'hrbeu')
```

7、基本运算符号实验:

1) 矩阵加法+

两个矩阵要求同行同列, 操作完成对应元素的加法。

```
data = [1,2,3,4;5,6,7,8;9,10,11,12];
```

```
b = [1,1;2,2;3,3];
```

```
a+b
```

2) 矩阵减法-

两个矩阵要求同行同列, 操作完成对应元素的减法

```
a = [1,2;3,4;5,6];
```

```
b = [1,1;2,2;3,3];
```

```
a-b
```

3) 矩阵乘法*

两个矩阵要求第一个矩阵的列数等于第二个矩阵的行数。对应矩阵的乘法运算。

`c=[1,2,3,4;5,6];`

`d=[1,1,1;1,1,1]`

`c*d`

4) 数组点乘.*

`e=[1,2,3];`

`f=[6,3,2];`

`e.*f`

5) 矩阵乘方^(乘方的矩阵必须是正方形)

`c=[1,2;3,4;5,6];`

`d=[1,1,1;1,1,1]`

`(c*d)^2`

6) 数组的乘方.^

计算规则和内积相同，为内容相同的向量之间的内积运算。

`e=[1,2,3];`

`e.^2`

7) 矩阵的左除

`e=[1,2,3];`

`f=[6,3,2];`

`e\`

8) 矩阵的右除/

`e=[1,2,3];`

`f=[6,3,2];`

`e/f`

9) 数组的左除

同内积的运算法则。

`e=[1,2,3];`

`f=[6,3,2];`

e.□

10) 数组的右除./

同内积的运算法则

```
e=[1,2,3];
```

```
f=[6,3,2];
```

```
e./f
```

11) 克罗内克张量积

kron(a,b) 表示利用 a 的每一个元素和 b 矩阵相乘,乘后分别作为结果矩阵的一块.

```
a=[1,2,3;4,5,6];
```

```
b=[1,0,0;0,1,0;0,0,1];
```

```
kron(a,b)
```

```
kron(b,a)
```

12) 逻辑与&

当数组中的元素数字不为 0, 均表示逻辑 1。

```
a=[1,2,3];
```

```
b=[3,4,5];
```

```
a&b
```

```
a=[-1,1,1]
```

```
a&b
```

```
a=[0,1,0];
```

```
a&b
```

13) 逻辑或|

两个矩阵之间对应元素作或运算。

```
a=[1,2,3];
```

```
b=[3,4,5];
```

```
a|b
```

```
a=[-1,1,1]
```

```
a|b
```

```
a=[0,1,0];
```


`a\b`

14) 逻辑非 `~`

一个矩阵中的所有元素取非，非零元素的非表示零。

`a=[1,2,3];`

`~a`

15) 逻辑异或 `xor`

两个矩阵之间异或运算。

`a=[1,2,3];`

`b=[3,4,5];`

`xor(a,b)`

8、矩阵分析实验

a. 矩阵的范数

求取矩阵的范数。

`a=[1,2;3,4];`

`norm(a)`

b. 条件数

求取一个矩阵的条件数

`a=[1,2;3,4];`

`cond(a)`

c. 行列式

将一个矩阵转换为对应的行列式，并计算行列式的值。

`a=[1,2;3,4];`

`det(a)`

d. 矩阵的秩

求一个具体矩阵的秩

`a=[1,2;3,4];`

`rank(a);`

e. 特征值

求一个矩阵的特征值

`eig(a)`

f. 化零矩阵 `null`

将矩阵转化为零矩阵

```
a=[1,2;3,4];
```

```
null(a)
```

g. `chol` 分解

将矩阵按照 `chol` 方法分解

```
n=5;
```

```
X=pascal(n);
```

```
R=chol(X);
```

```
c=transpose(R)*R
```

i. `Lu` 分解

采用 `LU`函数的方式分解矩阵

```
a=[-1,8,-5;9,-1,2;2,-5,7];
```

```
[L1,U1]=lu(A);
```

j. 正交分解 (`qr`)

求一个幻方的正交分解后的值

```
a=magic(3);
```

```
[q,r]=qr(a);
```

k. 奇异值分解 (`svd`)

求一个矩阵的奇异值分解后的值

```
a=[1,2;3,4;5,6;7,8];
```

```
[u,s,v]=svd(a);
```

9、数值计算实验：

1) 零点 (`fzero` 、 `fsolve`)

```
x=[-3::4];
```

```
y=sin(x).*x.^2-x+1;
```

```
plot(x,y,'r','LineWidth',
```

```

x=[-3::4];
y=sin(x).*x.^2-x+1;
plot(x,y,'r','LineWidth',
hold on
h=line([-3,4],[0,0]);
set(h,'LineWidth',
set(h,'color','k')
set(gca,'Xtick',[-3::4])
title('the zero of function')
grid
xlabel('x')
[x1,f1,exitflag1]=fzero(f;
x=[-5::5];
y=x;
[X,Y]=meshgrid(x,y);
Z=2*X-Y-exp(-1*X);
surf(X,Y,Z);
shading interp
x0=[-5;-5];
options=optimset('Display','iter');
[x,fval]=fsolve(@fsolvefun,x0,options)
2) 导数 (diff )
syms x;
f=x^3+2*x+3;
dfdx=diff(f,x);
3) 梯度 (gradient )
v = -2::2;
[x,y] = meshgrid(v);

```

```
[px,py] = gradient(z,.2,.2);  
contour(v,v,z), hold on, quiver(v,v,px,py), hold off
```

4) fminbnd、fminsearch、fminunc 极值

```
x1 = 0;  
x2 = 2*pi;  
[x,y] = fminbnd(@func,x1,x2)  
x=fminsearch(@func,x1)  
y=fminunc(@func,x1)
```

5) 积分 (quadl)

```
function f=hcurve(t)  
f=sqrt(4*cos(2*t).^2+sin(t).^2+1);  
len2=quadl(@hcurve,0,3*pi);
```

6) roots 求一个多项式的根

```
p = [1 -6 -72 -27];  
r = roots(p)
```

10、符号计算实验：

1) 化简：

```
syms x;  
f=cos(x)+sqrt(-1*sin(x).^2);  
simple(f)
```

2) 求 的解

```
x=solve('(x+2)^x=2')
```

实 1、一维数组创建实验

验 1) 直接输入法：

记 b1 =

录 1 2 3 4 5

2) 步长增长法：

b2 =

3) 定数线性采样的方法 (`linspace(a,b,n)`):

`b3 =`

0

4) 定数对数采样的方法(`logspace(a,b,n)`):

`b4 =`

2、高维数组创建实验:

1) 直接输入的方法:

`a1 =`

1 2 3

4 5 6

2) 使用下标生成数组的方法:

`a2(:,:,1) =`

0 0

0 0

`a2(:,:,2) =`

0 0

0 1

3) 低维数生成高维数组:

`a3(:,:,1) =`

1 1 1

1 1 1

`a3(:,:,2) =`

2 2 2

2 2 2

`a3(:,:,3) =`

3 3 3

3 3 3

) 使用函数创建高维数组

a. cat 函数

a4(:,:,1) =

```
1 1 1
1 1 1
```

a4(:,:,2) =

```
2 2 2
2 2 2
```

a4(:,:,3) =

```
3 3 3
3 3 3
```

b. reshape 函数

a5(:,:,1) =

```
1 3
2 4
```

a5(:,:,2) =

```
5 7
6 8
```

a5(:,:,3) =

```
9 11
10 12
```

c. repmat 函数

a6(:,:,1) =

```
1 1 1
1 1 1
```

a6(:,:,2) =

```
1 1 1
1 1 1
```

a6(:,:,3) =

```
1 1 1
```

3、标准数组创建实验：

1) magic (n)

c1 =

```
8 1 6
```

```
3 5 7
```

```
4 9 2
```

2) ones(n)

c2 =

```
1 1 1
```

```
1 1 1
```

3) eye(n)

c3 =

```
1 0 0
```

```
0 1 0
```

```
0 0 1
```

4) randn(n,n)

c4 =

5) zeros(n,n)

c5 =

```
0 0 0 0
```

```
0 0 0 0
```

4、数组变换实验部分

1) 使用 diag

a1 =

```
1
```

6

11

构造对角 `diag([1,2,-1,4])`

`a11=`

1 0 0 0

0 2 0 0

0 0 -1 0

0 0 0 4

2) 使用 `fliplr`

`a2 =`

4 3 2 1

8 7 6 5

12 11 10 9

3) 使用 `flipud`

`a3 =`

9 10 11 12

5 6 7 8

1 2 3 4

4) 使用 `rot90`

`a4 =`

12 11 10 9

8 7 6 5

4 3 2 1

5) 使用 `tril`

`a5 =`

1 0 0 0

5 6 0 0

9 10 11 0

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/845104324200011332>