

# matlab文件操作

山东理工大学理学院 周世祥

20151127

## Contents

---

- 保存数据到文件
- 从文件中读数据
- 读入txt文件
- matlab界面中导入数据
- 高级文件输入/输出
- 文件的状态和定位
- 输入文件函数的比较
- 写入文件
- 写入和读取电子表格类文件
- 使用MAT文件变量

## 保存数据到文件

---

### Description of first code block

```
mymat = rand(2,3)
save testfile.dat mymat -ascii %save总是从文件开头写起
% 显示文件内容
type testfile.dat
```

```
mymat =

    0.9593    0.1386    0.2575
    0.5472    0.1493    0.8407

    9.5929143e-01    1.3862444e-01    2.5750825e-01
    5.4721553e-01    1.4929401e-01    8.4071726e-01
```

## 追加数据

```
mymat=rand(3,3);
save testfile.dat mymat -ascii -append
type testfile.dat
```

```
    9.5929143e-01    1.3862444e-01    2.5750825e-01
    5.4721553e-01    1.4929401e-01    8.4071726e-01
    2.5428218e-01    9.2926362e-01    2.5108386e-01
    8.1428483e-01    3.4998377e-01    6.1604468e-01
    2.4352497e-01    1.9659525e-01    4.7328885e-01
```

保存的默认格式

```
savefile = 'pqfile.mat';
p = rand(1, 10);
q = ones(10);
save(savefile, 'p', 'q')
clear
load pqfile.mat
p
```

p =

Columns 1 through 7

0.3517    0.8308    0.5853    0.5497    0.9172    0.2858    0.7572

Columns 8 through 10

0.7537    0.3804    0.5678

小练习：提示用户输入矩阵的行数和列数，创建一个随机矩阵，然后将矩阵写到一个文件中

从文件中读数据

---

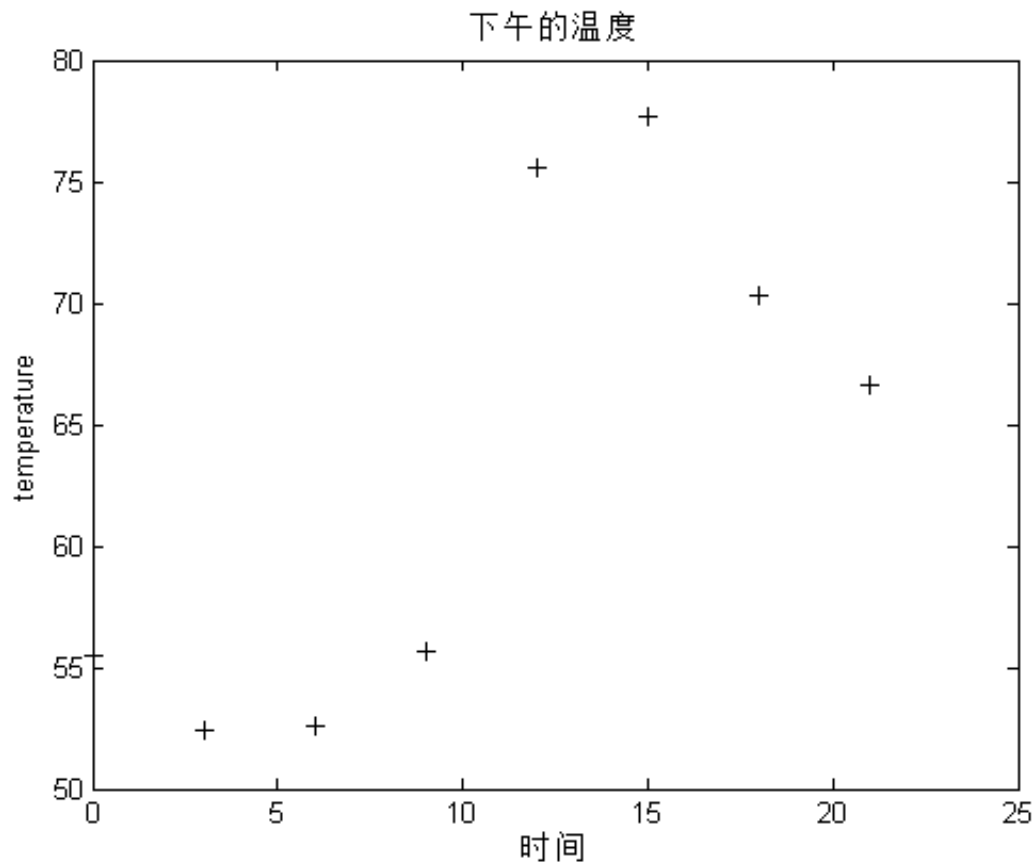
```
clear
load testfile.dat
who
```

Your variables are:

testfile

**load**命令仅仅能读出与列数相同的文件，所以读出的数据可以存到矩阵中，**save**命令仅仅能将矩阵中的数据写到文件中。如果不是这样就要用到低级的文件I/O函数。**timetemp.m**

```
load timetemp.dat %列数必须相同
%0 3 6 9 12 15 18 21
% 55.5 52.4 52.6 55.7 75.6 77.7 70.3 66.6
time = timetemp(1,:);
temp = timetemp(2,:);
plot(time,temp,'k+')
xlabel('时间'),ylabel('temperature'),title('下午的温度')
```



现在有一个数据文件按逆序排序，如何把其纠正过来

```
clear
load expresults.dat
% 7 55.2
% 6 51.9
% 5 49.5
% 4 53.4
% 3 44.3
% 2 50.0
% 1 55.5
expresults
correctorder=flipud(expresults)
save neworder.dat correctorder -ascii
```

expresults =

7.0000	55.2000
6.0000	51.9000
5.0000	49.5000
4.0000	53.4000
3.0000	44.3000
2.0000	50.0000
1.0000	55.5000

correctorder =

1.0000	55.5000
2.0000	50.0000
3.0000	44.3000

```
4.0000    53.4000
5.0000    49.5000
6.0000    51.9000
7.0000    55.2000
```

## 读入txt文件

```
clear
a=linspace(1,30,8);
save d:\exper.txt a -ascii;
b=load('d:\exper.txt');
```

如果文本文件中存储了不同类型的字符或数据，分类读取数据用`textread`函数。不需要使用`fopen`命令打开文件就可以直接读取txt内容。`[A,B,C,...]=textread('文件名','格式',N,'headerline',M),N`指读取的次数，每次读一行 `headerlines` 表示从第M+1行开始读入

```
clear
[name, type1, x, y, answer]=textread('d:\t.txt','%s Type%d %f %n %s',2,'headerlines',1);%从第二行开始读，读2次
```

## matlab界面中导入数据

Home面板上，"import Data"按钮

## 高级文件输入/输出

前面讲过`input()`读取输入值，`disp()`和`fprints`在屏幕上显示信息的输出函数。通过`load`和`save`函数进行文件的输入和输出。

用`load`和`save`函数时，一直使用`Ascii`格式文件，典型的文件类型为`.dat`或`.txt`。仅当每行值的个数相同且值的类型相同时，才用`load`命令，便于把这些数据存储在一个矩阵中，`save`命令仅仅把数据从矩阵写入到文件中。

如果要写入的数据或读取的文件的格式或读取的文件的格式不同，必须使用低级文件输入/输出函数。`matlab`软件一些函数可以从不同的文件类型读取和写入数据，如电子表，`excel`。`matlab`也有自己的二进制文件类型，使用扩展名`.mat`，可以用来存储`matlab`中创建的变量。

低级的文件I/O函数 当从文件中读取数据时，只有在文件中的数据是规则时(每行数据类型相同且每行格式相同)，`load`函数才能工作，以便能读入到一个矩阵中。当不能用`load`函数时，`matlab`有低级文件输入函数可以调用。打开文件，查找和建立文件，及在文件起始处放置标志，读取完成时，关闭文件。

打开和关闭文件

`fid=fopen('filename','permission string');` 打开文件用来读取 权限字符串包括：`r`读取(默认)，`w`写入，`a`追加； 通过`help fopen`查阅。若打开文件失败，返回-1；若成功，返回一个整数做文件标识符。

```
fid=fopen('samp.dat');
if fid == -1
    disp('File open not successful')
else
    % 执行和使用该文件
end
% 测试返回值来决定打印错误信息

% 当程序完成读取或写入后，应关闭文件，fclose()，关闭成功返回0，也可以通过指定文件标识符来关闭单个文件，或
```

% 如果打开了多个文件，通过传入字符串'all'到fclose函数来关闭所有的打开状态的文件。

```
closeresult=fopen(fid);
closeresult=fopen('all');
```

File open not successful

Error using fclose

Invalid file identifier. Use fopen to generate a valid file identifier.

Error in fileReadWrite (line 113)

```
closeresult=fopen(fid);
```

用if-else语句来检测fclose函数是否成功关闭文件。

```
fid=fopen('filename','permission string');
if fid == -1
    disp('File open not successful')
else
    closeresult = fclose(fid);
    if closeresult == 0
        disp('File close successful')
    else
        disp('File close not successful')
    end
end
```

2. 读取文件 fscan函数读取格式化数据到一个矩阵，转换格式符%d用于整数，%s用于字符串，%f用于实型。textscan函数从一个文件中读取文本数据并将其存储在元胞数组中。fgets函数和fgetl函数每次从一个文件中读取一行字符串，若行末有换行符，fgets函数会保留它，而fgetl函数不会。

因为fgetl和fgets每次读一行所以这些函数通常是以循环方式执行，fscan和textscan函数可以把一个完整的数据文件读入到一个数据结构里。这两个函数按级别，介于load函数和其他如fgetl的低级函数之间。feof函数，当读取到文件结尾，feof函数返回逻辑真，如果文件标识符fid到了文件的结尾，feof(fid)函数调用将返回逻辑真，否则为假。

```
fid = fopen('filename');
if fid == -1
    disp('这个文件未成功打开!');
else
    while feof(fid) == 0%或 while ~feof(fid)
        aline=fgetl(fid);
    end
    closeresult=fopen(fid);
    if closeresult == 0
        disp('文件关闭成功!')
    else
        disp('文件未关闭成功!')
    end
end
```

例如

```
fid=fopen('kl.txt','r');
```

```

while ~feof(fid) %在文件结束前按行读取数据
    s=fgets(fid);
    fprintf(1, '%s', s); %读换行符
end
fclose(fid);
%fgetl不读换行符

```

权限字符串可以包含在fopen函数的调用中，如，fid=fopen('文件名','r');

例题，假如有一数据文件subjexp.dat

5.3 a

2.2 b

3.3 a

4.4 a

1.1 b

有数字和文本，load函数不能把它读到一个矩阵中，可以用fgetl函数把每行读成一个字符串，然后用字符串函数来分割数字和字符

```

% fileex.m
fid =fopen(' subjexp.dat');
if fid ==-1
    disp(' 文件打开失败');
else
    while feof(fid) ==0
        aline =fgetl(fid);
        [num ,charcode]=strtok(aline);
        fprintfd(' %.2f %s\n', str2num(num), charcode)
    end
    closeresult =fclose(fid);
    if closeresult ==0
        disp(' 文件关闭成功');
    else
        disp(' 文件关闭失败');
    end
end
% 运行
% fileex

% 小练习： 修改文件， 对上述文件中的数字求和。

```

总结： 打开和读写文件： fopen,fclose 格式化读写： fprintf,fscanf,fgetl,fgets 非格式化读写： fread,fwrite

```

clear
t=0:0.001:1;
fp=fopen(' k.txt', 'w');
fprintf(fp, '%d', t); %将数据写到fp指定的文件中
length(t)

clear

```

```

x=0:10;
y=[x;exp(x)];
fid=fopen('k1.txt','w');
fprintf(fid,'%6.2f %12.8f\n',y);
fclose(fid);
% 计算f(x)=e^x值并把结果写入k1.txt
fid=fopen('k1.txt','r');
[a,count]=fscanf(fid,'%f %f',[2 inf]);
%[2 inf]表示每列两个元素，读出fid所指的全部元素
fprintf(1,'%f %f\n',a);% 1表示输出到屏幕
fclose(fid);
% mat = fscanf(fileID, format, [dimensions])
% fscanf直接读取文件到一个矩阵中

```

**fscanf**按列的方式读取数据到矩阵变量**mat**中

```

fid=fopen('subjexp.dat')
mat=fscanf(fid,'%f %c',[2 inf])% 若不知值的个数，第二维用inf
% 因为矩阵存储相同的类型的值，字符以其字符编码中对应的ASCII码形式存储，有效的方法是把行分成向量变量，
% 并把第二行转换成字符。

nums=mat(1,:);
charcodes=char(mat(2,:));
fclose(fid);

```

## 文件的状态和定位

```

clc;
close all;
fid =fopen('k.txt','r');
fseek(fid,0,'eof');% 设置文件位置指针
x=ftell(fid);%查询当前文件指针的位置
fprintf(1,'File size=%d\n',x);
frewind(fid);%使文件指针重返文件开头
x=ftell(fid);
fprintf(1,'File positon =%d\n',x);
fclose(fid);
% status=feof(fid);%检查文件是否已结束

```

**textscan**函数从文件中读取文本数据并将其存储在一个元胞数组中. **cellarray=textscan(fid,'format');**

```

clear
fid=fopen('subjexp.dat','r');
subjdata=textscan(fid,'%f %c');%1*2的元胞数组变量
fclose(fid);
subjdata
subjdata{1}
subjdata{2}
subjdata{3}(3) %大括弧索引元胞数组，再用小括弧在向量中索引

```

小练习：先不要看解答。读入上面的数据并打印它的脚本如下： **textscanex.m** 用**textscan**读取数据

```

fid =fopen('subjexp.dat');
if fid == -1

```

```

disp(' 文件未成功打开! ');
else
    subjdata=textscan(fid,'%f %c');
    len=length(subjdata{1});
    for i=1:len
        fprintf('%1f %c\n',subjdata{1}(i),subjdata{2}(i));
    end
    closeresult=fclose(fid);
    if closeresult ==0
        disp(' 文件关闭成功! ');
    else
        disp(' 文件关闭未成功! ');
    end
end
% 运行 textscanex, 回车

```

## 输入文件函数的比较

```

clear
type xypoints.dat
% x2.3y4.56
% x7.7y11.11
% x12.5y5.5
% 文件行中存储字符和数字组合，因此不能用load函数，要将字符从数字中 分离出来
% fileInpCompare.m
fid =fopen(' xypoints.dat');
if fid ==-1
    disp(' 文件打开不成功! ');
else
    plot(x,y,'k*') %此处用四种不同的方法获取点的数据
    xlabel(' x')
    ylabel(' y')
    closeresult=fclose(fid);
    if closeresult==0
        disp(' 文件关闭成功! ');
    else
        disp(' 文件关闭不成功! ');
    end
end
end

```

### ■ 第一种方法：用fgetl

```

x=[],y=[];
while feof(fid) == 0
    aline=fgetl(fid);
    aline=aline(2:end);
    [xstr rest]=strtok(aline,' y');%函数用于自左向右提取字符串中某字符或字符串，提取的依据是给定的delimiter
    x=[x str2num(xstr)];
    ystr=rest(2:end);
    y=[y str2num(ystr)];
end

```

### ■ 第二种方法 用fscanf 需要指定文件的每行的格式：字符，数字，字符，数字，换行符 创建的矩阵会将文件的每行存储为单独的列，维数是4\*n，n是文件行数(若不确定inf) x字符在矩阵的第一行('x'为Ascii码)，x坐标值在第二行，'y'等价的ascii值在第三列，第四行存储y坐标值。



```
mat=fscanf(fid,'%c%f%c%f\n',[4 inf]);%换行符必须有
x=mat(2,:);y=mat(4,:);
```

- 第三种方法：还是用**fscanf** 因为每行中都包含'**x**','**y**'，可以将其创建为格式化字符串

```
mat = fscanf(fid,'x%fy%f\n',[2,inf]);
x=mat(1,:);
y=mat(2,:);
%这种情况，字符'x','y'不会被读入矩阵
```

- 第四种方法：用**textscan**

```
xydat=textscan(fid,'x%fy%f');
%不需要换行符的原因是，使用textscan时，格式化字符串指定了文件中的列的格式，而对fscanf，指定了文件每行的格式
x=xydat{1};
y=xydat{2};
```

- 总结：四种读取文件的方法，**load**函数只适用于文件中类型相同且每行值的个数相同，这样可以把它们读入一个矩阵中。**fscanf**函数读入矩阵时，把字符转换为对应的**ascii**码。**textscan**函数代之以将数据读入到一个元胞数组中，且把文件的每一行分别存储到元胞数组的列向量中。在循环中使用**fgetl**函数把文件的每行作为一个单独的字符串来读取，再用字符串操作函数把该串分割并转换成数值。

## 写入文件

有几种低级函数可以用来写入文件，**fprintf**函数 **fprintf(fid,'format',variable(s));**%屏幕是默认的输出设备，默认文件，标识符中**1**是指屏幕 如果不想看到返回写入的文件字节数，可以将其存入变量

```
clear
fid = fopen('tryit.txt','w');
for i=1:3
    fprintf(fid,'循环变量为%d\n',i);
end
fclose(fid);
```

- 像读取文件一样，要检查**fopen**和**fclose**的结果以确认其操作是否成功。**fopen**函数试图打开文件用来写入，如果文件存在，但内容被删除，如果文件不存在，就要创建一个新文件；如果没有空间创建一个新文件，则**fopen**有可能失败。为查看写入内容，可用**fgetl**循环读取每一行

```
fid =fopen('tryit.txt');
while ~feof(fid)
    aline=fgetl(fid)
end
```

例题：把矩阵写入文件

```
clear
mat =randint(2,3,[5,20])
fid =fopen('randmat.dat','w');
fprintf(fid,'%d %d\n',mat);
fclose(fid);
```

```
%因为是一个矩阵，可以用load函数读取
load randmat.dat
randmat
randmat' %原形式
% 若追加文件内容 fid=fopen('filename','a');
```

## 写入和读取电子表格类文件

写入到ranexcel.xls文件中

```
ranmat=randi(100,[5,3]);
xlswrite('ranexcle',ranmat);
% xlsread函数可以读取电子表格类文件
ssnums=xlsread('ranexcle');
```

若电子表格中还有文本：

```
% a 315 shuxue
% b 333 suzanne
% c 432 bangongshi
% d 312 Xinke
[nums,txt]=xlsread('texttest.xls');%会把数值读到一个double型向量中，文本读到一个元胞数组txt中
txt{1,2}
txt{1,3}
%通过循环再次打印表格中的值
for i=1:length(nums)
    fprintf('%c %d %s \n',txt{i,1},nums(i),txt{i,3});
end
```

## 使用MAT文件变量

除数据文件类型，Matlab还允许从文件中读取和保存变量的函数。这些文件称为mat文件，可以存储变量的名称和内容。

### 1. 将变量写入到文件

```
mymat=rand(3,5)
x=1:6;
y=x.^2;
who
save sess1
who -file sess1
% 只保存一个变量到文件中
save sess2 mymat
who -file sess2
% 将变量追加到mat文件中
save -append sess2 x
who -file sess2
% 不指定具体变量
save -append
%会把当前命令窗口中所有的变量添加到文件中，若有重名的变量被覆盖
```

### 1. 从Mat文件中读取

```
who
```

```
load sess2
who
```

.....

*Published with MATLAB® R2012b*