



数学软件 Matlab

—— 文件操作

主要内容

- 文件的打开与关闭
- 文本文件的写与读
- 二进制文件的写与读

文件操作介绍

- 文件操作是一种重要的输入输出方式，Matlab 提供了一系列输入输出函数，专门用于文件操作。
- Matlab中的输入输出函数是以 C 语言标准库函数中的输入输出函数为基础开发的，所以这些函数与 C 语言的输入输出函数相类似。
- Matlab文件操作主要有三个步骤：首先**打开**文件，然后对文件进行**读写**操作，最后要**关闭**文件。

文件的打开

`fid=fopen(文件名, 打开方式)`

- 其中文件名用字符串形式表示（可以带路径名）

- 打开方式有

'r'	只读，文件必须存在（缺省的打开方式）
'w'	写文件，若文件已存在则原内容将被覆盖； 若文件不存在则新建一个
'a'	在文件末尾添加，文件若不存在则新建一个
'r+'	可读可写，文件必须存在
'w+'	可读可写，若文件已存在则原内容将被覆盖； 若文件不存在则新建一个
'a+'	可读可写可添加，文件若不存在则新建一个

- 打开方式中也可以指定文件类型：

't'	文本文件
'b'	二进制文件

文件的打开

- `fid` 为文件句柄，其它函数通过它才能对该文件进行操作。如果句柄值大于 0，则表示文件打开成功；若打开失败，`fid` 的返回值为 -1。

例：

```
fid=fopen('output.txt', 'wt+');  
fprintf(fid, 'Hello world!\n');  
fclose(fid);
```

- 有两个标准代码文件，不需打开就可以直接使用，分别为：`fid=1` 标准输出文件，`fid=2` 标准错误文件。

例：

```
fprintf(1, 'Hello world!\n');  
fprintf(2, 'Hello world!\n');
```

文件的关闭

```
status=fopen(fid);
```

- 其中 `fid` 为所要关闭的文件的句柄，`status` 为关闭文件的返回代码，若关闭成功则为 `0`，否则为 `-1`
- 文件操作结束后一定要关闭

主要内容

- 文件的打开与关闭
- 文本文件的写与读
- 二进制文件的写与读

文本文件的写入

```
count=fprintf(fid, format, 输出变量列表)
```

- 将指定的变量按指定的格式写入文件中
- 若省略 `fid`，则表示在屏幕上输出
- `count` 返回所写入的数据元素个数（可省略）
- `format` 用来指定数据输出时采用的格式，包含：
普通字符串、格式字符串、转义字符

格式字符串：以 % 开头，包括：

- flags（可选）
- Width and precision fields（可选）
- Conversion character（格式说明符，必须）

fprintf 举例

```
fprintf('a= %-12.5f \n', pi)
```

格式说明符

以 % 开头

flag

field width

precision

- :左对齐
+ :输出符号
0 :空白处添 0
空格:前面加一空格

字段宽度

小数点后的位数

格式说明符和转义字符

● 常见的格式说明符

c	字符型	g	浮点数（自动）
d	十进制整数	o	八进制
e	浮点数（科学计数法）	s	字符串
f	浮点数（小数形式）	x/X	十六进制

● 常见的转义字符

\b	退后一格	\t	水平制表符
\f	换页	\\	反斜杠
\n	换行	''	单引号
\r	回车	%%	百分号

文本文件的写入举例

例:

```
x=0:0.1:1;
y=exp(x);
fid=fopen('output.txt','wt');
fprintf(fid,'    x        exp(x)\n');
for k=1:length(x)
    fprintf(fid,'%6.2f  %12.8f\n', x(k), y(k));
end
status=fopen(fid);
```

- **注: 格式的重复使用**

```
x=0:0.1:1; y=exp(x); z=[x; y];
fid=fopen('output.txt','wt');
fprintf(fid,'    x        exp(x)\n');
fprintf(fid,'%6.2f  %12.8f\n', z);
status=fopen(fid);
```

文本文件的读取

```
[A, count]=fscanf(fid, format, size)
```

- **A** 用来存放读取的数据
- **count** 返回读取数据的个数，为可选项
- **fid** 为文件句柄
- **size** 为可选项，缺省为读取整个文件，其取值可以是：

N	读取 N 个数据到一个列向量
Inf	读取整个文件
[m,n]	读取 m × n 个数据到一个 m × n 矩阵中，按列存放

主要内容

- 文件的打开与关闭
- 文本文件的写与读
- 二进制文件的写与读

二进制文件的写入

```
count=fwrite(fid,A,precision)
```

- 按指定的数据类型将矩阵 **A** 中的元素写入到文件中。
其中 **count** 返回所写入的数据元素个数（可省略）

例：

```
A=magic(5);  
fid=fopen('magic5.dat','w');  
fwrite(fid,A,'int8');  
fclose(fid);  
  
fid=fopen('magic5.dat','r');  
B=fread(fid,[5,inf],'int8');  
fclose(fid);
```

二进制文件

- **precision** 代表写入的数据的类型，缺省为 **uchar**

'uchar'	无符号字符	'uint16'	16位无符号整数
'schar'	带符号字符	'uint32'	32位无符号整数
'int8'	8位带符号整数	'uint64'	64位无符号整数
'int16'	16位带符号整数	'float32'	32位浮点数
'int32'	32位带符号整数	'float64'	64位浮点数
'int64'	64位带符号整数	'double'	64位双精度数
'uint8'	8位无符号整数		

二进制文件

- 以下数据类型与使用的平台有关：

'char'	带符号字符
'short'	16位带符号整数
'int'	32位带符号整数
'long'	32或64位带符号整数
'ushort'	16位无符号整数
'uint'	32位无符号整数
'ulong'	32或64位无符号整数
'float'	32位浮点数

二进制文件的读取

```
A=fread(fid,size,precision)
```

- **A** 用来存放读取的数据
- **fid** 为文件句柄
- **size** 为可选项，缺省为读取整个文件。取值可以是：

N	读取 N 个数据到一个列向量
Inf	读取整个文件
[m,n]	读取 m × n 个数据到一个 m × n 矩阵中，按列存放

二进制文件读写举例

例:

```
A=rand(100);  
fid=fopen('output.dat','wb');  
fwrite(fid,A,'double');  
status=fclose(fid);
```

```
fid=fopen('output.dat','rb');  
B=fread(fid,[100,100],'double');  
status=fclose(fid);
```