

シミュレーション基礎(8)

第6章 ファイル入出力

7.2 テキストファイルの読み書き

◆ファイルに書き込む:

```
EX70201:  
X=1:10;Y=[X;X.^2];  
Fid=fopen('data1.txt','wt');  
fprintf(Fid,'%2d%5d¥n',Y);  
fclose(Fid);
```

C言語と同じ手順:
ファイルをオープンする
ファイルに変数の値を書き込む(整数2桁, 整数5桁, 改行)
ファイルをクローズする

実行したら type data1.txt とコマンドラインに入力してみよう

◆ファイルから読み込む:

```
EX70202:  
Fid=fopen('data1.txt','rt');  
Z=fscanf(Fid,'%d%d',[2,inf]);  
fclose(Fid);  
Z'
```

C言語と同じ手順:
ファイルをオープンする
ファイルから変数の値を読み出す(整数, 整数)
ファイルをクローズする

読み込んだデータを2行の行列の形にして変数Zに格納
infはデータのある限りファイルの終わりまで読み込むことを意味

fopenのまとめ

[fid,message] = fopen(filename,permission)

filename ファイル名

permission オプション

fid ファイル識別子（数値）とエラーメッセージmessage（文字列）を返す.

オプションpermissionにはつぎのものがある(tはtextをあらわす):

'rt' ファイルの読み込み

'rt+' ファイルと読み込みと書き込み

'wt' ファイルの書き込み(もし同名のファイルがあればそれを削除する)

'wt+' ファイルと読み込みと書き込み(もし同名のファイルがあればそれを削除する)

'at' ファイルへの追加書き込み

'at+' ファイルへの追加書き込みと読み込み

fprintf のまとめ

count=fprintf(fid,format,A,····)

A 変数(複数個)
count 書き込んだデータ数 (数値)

フォーマットの例

%-12.5f 数字を符合部分と幅12桁の中に小数点以下5桁をとり固定小数点の形で書き込む

変換文字の種類

c 文字
d 10進整数表示
e 指数表示 3.1415e+00
E 指数表示 3.1415E+00
f 固定小数点表示 3.1415
g 指数表示または固定小数点表示のいずれか簡潔な方(指数が小文字)
G 指数表示または固定小数点表示のいずれか簡潔な方(指数が大文字)
o 8進数表示(符合なし)
s 文字列
u 符合なしの10進整数
x 16進数表示(小文字)
X 16進数表示(大文字)

改行などのいわゆる制御文字(エスケープ文字)

¥n 改行(ラインフィード)
¥t タブ
¥b バックスペース
¥r キャリッジリターン
¥f フォームフィード

fscanfのまとめ

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

- fid ファイル識別子
- format フォーマットと読み込むデータの個数size
- A データを格納する変数名
- count 読み込んだデータの個数

sizeはつぎの形で指定:

- n n個のデータを列ベクトルに格納する
- inf ファイルの終わりまでのデータを列ベクトルに格納する
- [m, n] m行n列の行列に列方向にデータを格納する. nはinfでもよい.

◆タイトルを付けて書き込む
データの説明文もいっしょに書いておく

```
EX70203:  
S=' x  x^2 '  
X=1:10; Y=[X;X.^2]  
Fid=fopen('data3.txt','wt');  
fprintf(Fid,'%s¥n',S);  
fprintf(Fid,'%2d%5d¥n',Y);  
fclose(Fid);
```

実行したのちtypeで見てみよう:
type data3.txt

◆タイトルも読み込む

```
EX70204:  
clear all  
Fid=fopen('data3.txt','rt');  
T=fscanf(Fid,'%7c',1);  
Z=fscanf(Fid,'%d%d',[2,inf]);  
fclose(Fid);  
T, Z'
```

文字数が7文字とわかっていなければいけない

◆タイトル行のある文字数が不明のとき

```
EX70205:  
  
Fid=fopen('data3.txt','rt');  
T=fgetl(Fid);  
Z=fscanf(Fid,'%d%d',[2,inf]);  
fclose(Fid);  
T, Z'
```

fgetl 1行分のデータを文字列として読み込む(改行文字などは読まない)
改行文字まで読み込みたいときは fgets を使う

◆ファイルから文字列を探す:

fgetlの簡単な応用

EX70206:

```
Fid=fopen('data3.txt','rt');  
while feof(Fid)==0  
Line=fgetl(Fid);  
Index=findstr(Line,'6');  
N=length(Index);  
if N>0  
fprintf('%d:%s¥n',N,Line);  
end  
end  
fclose(Fid);
```

eof(Fid): ファイルの最後を検知すると1となる

Fidから1行を変数Lineに読み込む

findstr 文字列から指定された文字列を探し出し、始まる位置を配列indexに返す

length 配列の大きさを返す

fprintfにファイル識別子がない場合は標準出力:ディスプレイ

◆数値を1行に書く

EX70207:

```
X=1:10; Y=[X;X.^2]  
Fid=fopen('data7.txt','wt');  
fprintf (Fid, 'x=%2d x^2 = %5d¥n', Y);  
fclose(Fid);
```

行列Yは1列, 2列と列順に書き込まれていく

type data7.txt で、出来上がったファイルを出力してみよう

◆1行に書かれている文字列と数値から数値を読み込む

EX70208:

```
Fid=fopen('data7.txt','rt');  
Z = fscanf(Fid, '%2s%d%4s%d' , [8,inf]);  
fclose(Fid) ;  
Z'  
Y=[Z(3, :) ;Z(8, :)]
```

8行の行列としてデータの終わりまで読む

◆ファイルに追加して書き込む
オープンするとき、オプションとしてatを指定

EX70209:

```
X=1:10; Y=[X;X.^2]
```

```
Fid=fopen('data9.txt1','wt');
```

```
fprintf (Fid, '%2d%5d¥n', Y) ;
```

```
fclose(Fid);
```

```
X=11:20; Y=[X;X.^2]
```

```
Fid=fopen('data9.txt','at');
```

```
fprintf(Fid,'%2d%5d¥n',Y);
```

```
fclose(Fid);
```

fopenにオプションとしてatを指定

7.3 バイナリファイルの読み書き

◆ファイルに書き込む:fwrite

データをバイナリで保存. 無駄なく高速に読み書き可能

EX70301:

1. `X=1:10; Y=[X;X.^2]`

2. `Fid=fopen('data1.bin','w');`

3. `fwrite(Fid,Y,'int16');`

4. `fclose(Fid);`

Data1.binというファイルをバイナリ書き込みモードでオープンする

2バイト整数で書き出す

`[fid,message]=fopen(filename,permission)`

permission:

'r' ファイルの読み込み

'r+' ファイルの読み込みと書き込み

'w' ファイルの書き込み(同名のファイルがあればそれを削除する)

'w+' ファイルの読み込みと書き込み(同名のファイルがあればそれを削除する)

'a' ファイルへの追加書き込み

'a+' ファイルへの追加書き込みと読み込み

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

または

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

A; 変数

precision精度を指定するオプション

, 書き込んだデータ数 count (数値)を返します.

変数Aが行列のときは, 第1列がまず書き込

まれ, つぎに第2列が書き込まれ, というふうに

列方向にデータが格納さわ

ていきます. また, 精度には

| | | |
|-----------|-----------------|------|
| 'char' | 文字 | 1バイト |
| 'schar' | 'uch符号付き文字 | 1バイト |
| 'char' | 'int8'bbb符号なし文字 | 1バイト |
| 'int16' | 整数 | 2バイト |
| 'int32' | 整数 | 4バイト |
| 'int64' | 整数 | 8バイト |
| 'uint8' | 符号なし整数 | 1バイト |
| 'uint16' | 符号なし整数 | 2バイト |
| 'uint32' | 符号なし整数 | 4バイト |
| 'uint64' | 符号なし整数 | 8バイト |
| 'float32' | 実数 | 4バイト |
| 'float64' | 実数 | 8バイト |



◆ファイルから読み込む :fread p226

EX70302:

```
Fid=fopen('data1.bin','r');
```

```
Z=fread(Fid, [2, 10], 'int16');
```

```
fclose(Fid);
```

```
Z'
```

[2, 10]: データを2行10列の行列として読み出す
(データは1列目, 2列目, と列順に代入される.
行順ではないことに注意)

freadのまとめ

[A,count]= fread(fid,size, precision)

または

[A,count]=fread(fid,size,precision,skip)

引数:

Fid ファイル識別子

size 読み込むデータサイズ. 以下の種類がある

n : n個のデータを列ベクトルに格納する

inf : ファイルの終わりまでのデータを列ベクトルに格納する

[m,n] : m行n列の行列に格納する(1列, 2列, ... と)

precision 精度

前頁を参照のこと

戻り値

A : 変数名

count 読み込んだデータの個数

◆ファイルの中を調べる (ファイルダンプ)

指定されたファイルを1バイトずつ読み、それを16進数として表示することをファイルをダンプするという。プログラム作成前にasciiコードを復習しよう。

2バイト(16進数2桁)asciiコード表

| | 00 | 10 | 20 | 30 | 40 | 50 | 60 | 70 |
|----|----|----|----|----|----|----|----|----|
| 00 | NL | DE | SP | 0 | @ | P | ' | p |
| 01 | SH | D1 | ! | 1 | A | Q | a | q |
| 02 | SX | D2 | " | 2 | B | R | b | r |
| 03 | EX | D3 | # | 3 | C | S | c | s |
| 04 | ET | D4 | \$ | 4 | D | T | d | t |
| 05 | EQ | NK | % | 5 | E | U | e | u |
| 06 | AK | SN | & | 6 | F | V | f | v |
| 07 | BL | EB | ' | 7 | G | W | g | w |
| 08 | BS | CN | (| 8 | H | X | h | x |
| 09 | HT | EM |) | 9 | I | Y | i | y |
| 0A | LF | SB | * | : | J | Z | j | z |
| 0B | HM | EC | + | ; | K | [| k | { |
| 0C | CL | → | , | < | L | ¥ | l | |
| 0D | CR | ← | - | = | M |] | m | } |
| 0E | SO | ↑ | . | > | N | ^ | n | ~ |
| 0F | SI | ↓ | / | ? | O | _ | o | DL |

50は'P',51は'Q', 20はSP(スペース)を表すことが分かる

練習

あるテキストファイルを16進数で読み出したところ次のようになった。この結果をasciiコード表を参考に解読しなさい。

746865206C6573736F6E732070726F63....

(答) 74=t, 68=h,65=e,20=SP,6C=l...と読み進むと the lessons proc...となる。

◆ファイルの中を調べる (ファイルダンプ)関数



EX70303:

```
function ex70303(FileName)
```

```
% Dump any file
```

```
Fid=fopen(FileName,'r');
```

読み出した行数

```
NLine=0;
```

```
NChar=0;
```

読み出した文字数

```
while feof(Fid)==0
```

```
Buf=fread(Fid,1,'uchar');
```

1文字(1バイト)読み出す

```
if length(Buf)>0
```

```
    NChar=NChar+1;
```

16進数に直し変数Hbufに代入

```
    HBuf=dec2hex(Buf);
```

```
    if length(HBuf)<2, HBuf=['0' HBuf]; end
```

Hbufが1桁の場合は0を付け加え2桁とする

```
if mod(NChar, 16)==1
```

```
    NLine=NLine+1 ;
```

```
    fprintf('%5d %s',NLine,HBuf);
```

行の初めなら(1行16文字ずつとする)行番号とHbuf
を出力

```
else
```

```
    fprintf('%s',HBuf);
```

行の初めでなければHbufのみを出力

```
end
```

```
if mod(NChar,16)==0, fprintf('¥n'); end
```

行の最後16文字目であれば改行コードを出力

```
end
```

```
end
```

```
fprintf('¥n');
```

```
fclose(Fid);
```

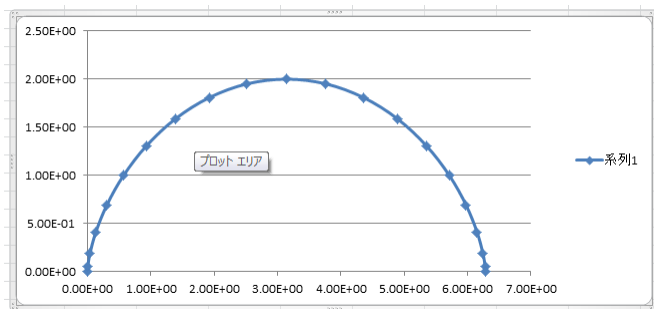
練習

1. サイクロイド曲線

$$\begin{cases} x = \theta - \sin \theta \\ y = 1 - \cos \theta \end{cases}$$

について区間 $0 \leq \theta \leq 4\pi$ まで $\pi/10$ ずつ θ を変化させるときの、点(X,Y)を計算し、それらを(X,Y)の組み合わせとしてテキストファイルrensyu.txtに書き出しなさい。

また、このテキストファイルをエクセルで読み出し、エクセルのグラフ機能を用いて曲線を描きなさい（折れ線または散布図）



2. 1で作成したファイルからデータを読み出し、曲線をプロットするプログラムを作成しなさい。