ファイル入出力

作ったanimationをgifファイルとして保存したり，測定データなどを読み込んで表示する手軽な方法がある．そのためにはファイルとのやりとりをする必要があります．

データの入出力

ファイル名の取得

ファイル名の取得は，Javaの標準関数を使ったMapletパッケージのGetFile関数を使う．GetFile関数を呼びだして開いたファイル選択ウィンドウでファイルを指定するとファイルのパスがfile1に入る．

```
> restart;
  with(Maplets[Examples]):
  file1:=GetFile();
```

```
file1 := "/Users/bob/MapleTest/data1.txt"                    (1.1.1.1)
```

Windowsでは"¥"を"/"に変換する必要がある．日本語のファイル名は文字化けして使えない．

```
> with(StringTools):
  file2:=SubstituteAll(file1,"¥¥¥¥","/");
```

```
file2 := "/Users/bob/MapleTest/data1.txt"                    (1.1.1.2)
```

ファイル名の変更は手でやるか，あるいはSubstitute関数を使う．

```
> with(StringTools):
  file2:=Substitute(file1,"data1","data2");
```

```
file2 := "/Users/bob/MapleTest/data2.txt"                    (1.1.1.3)
```

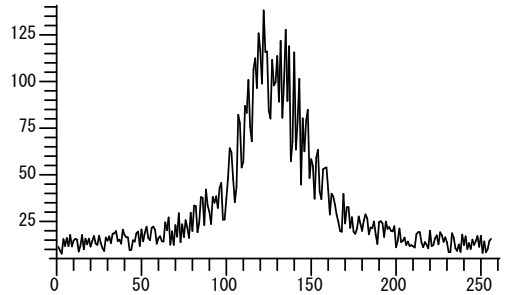
簡単なデータのやりとり

ファイルとの単純なデータのやりとりはwritedata,readdata関数を使う．例えば，以下のようなデータを作ったとする．これをファイルへ書きだすには，以下のようする．

```
> f1:=t->subs({a=10,b=40000,c=380,d=128},a+b/(c+(t-d)^2) ):
  T:=[seq(f1(i)*(0.6+0.8*evalf(rand()/10^12)),i=1..256)]:
  writedata(file1,T);
```

同じようにして読み込んで表示させてみる．

```
> T:=readdata(file1,1):
  with(plots):
  listplot(T);
```



少し高度なデータのやりとり

writeto関数で出力を外部ファイルへ切り替えることも可能．

```
> interface(quiet=true);
  writeto(file2);
   for i from 1 to 10 do
    s1:=data||i;
    printf("%10.5f %s\n",evalf(f1(i)),s1);
   end do:
  writeto(terminal):
  interface(quiet=false);
```

```
                              false
                               true                          (1.1.3.1)
```

C言語の標準的なデータ読み込みに似せた動きもできる．以下はfopen, readline, sscanf, fcloseを使ったデータの入力．

```
> fd:=fopen(file2,READ);
  for i from 1 to 2 do
  l1:=readline(fd);
  d:=sscanf(l1,"%f %s");
  end do;
  fclose(fd):
```

```
                              fd := 1
                    l1 := "  12.42292 data1"
                    d := [12.42292, "data1"]
                    l1 := "  12.46063 data2"
                    d := [12.46063, "data2"]                  (1.1.3.2)
```

fdにファイル識別子(file descripter)を持っていき，readlineで1行ずつ読ませる．これをsscanfでformatにしたがってl1に格納していく．l1には自動的に適

切な形式で変数を入れてくれる.

```
> d[1];
  whattype(d[1]);
```

$$12.46063$$

$$\mathrm{float}$$ (1.1.3.3)

animationの出力

animationなどのgif形式のplotを外部ファイルへ出力して表示させるには，以下の一連のコマンドのようにする.

```
> plotsetup(gif,plotoutput=file2):
  display(tmp,insequence=true);
  plotsetup(default):
```

こいつをquicktimeなどに食わせれば，Maple以外のソフトで動画表示が可能となる．3次元図形の標準規格であるvrmlも同じようにして作成することが可能です(?vrml;参照).

linuxでのフィルターとしての利用法

linux版では文字ベースのmapleを使って，filterとして高度な作業をさせることが出来ます．スクリプトの中に外部ファイルとの入出力を組み込めば，いままで紹介してきた複雑な動作をブラックボックスの内部処理としてそのまま使えます.

```
[bob@asura0 ~/test]$ cat test.txt
T:=readdata("./data101");
interface(quiet=true);
writeto("./result");
print(T[1]);
writeto(terminal);
interface(quiet=false);
```

とすれば，data101から読み込んだデータに何らかの処理を施した結果をresultに打ち出すことが可能．interface(quiet=true)で余計な出力を抑止しています．これをmapleに食わせると

```
[bob@asura0 ~/test]$ /usr/local/maple9.5/bin/maple < test.txt
     |¥^/|     Maple 9.5 (IBM INTEL LINUX)
._|¥|   |/|_. Copyright (c) Maplesoft, a division of Waterloo
Maple Inc. 2004
 ¥  MAPLE  /  All rights reserved. Maple is a trademark of
 <____ ____>  Waterloo Maple Inc.
      |       Type ? for help.
> T:=readdata("./data101");
                        T := [1.23, 2.35]
> interface(quiet=true);
                           false
                            true
> quit
bytes used=211000, alloc=262096, time=0.00
```

めでたく出力されているはず.

```
[bob@asura0 ~/test]$ cat result
                            1.23
```

演習

>