情報処理演習 A8　（第 14 回）　　　　　　　　　　　　　　　　　　2003/7/09　藤野　毅

【グラフを書く】

(1)linux 上でグラフを作成したい場合,GNUPLOT というグラフ作成ソフトを使用すると簡単にできます．
%gnuplot
と入力すると次のプロンプトが出てグラフ描画ツール GNUPLOT が立ち上がります．
以下のように入力してみてください．

```
Terminal type set to 'x11'
gnuplot> plot sin(x)
gnuplot> replot (sin(x))**2            ←グラフを重ねて描画
gnuplot> set xrange [3:2*pi]           ←グラフ作成範囲の変更
gnuplot> replot                        ←再描画
gnuplot> set xrange [-3:2*pi]
gnuplot> replot
gnuplot> quit
```

(2)上記のように簡単に関数のグラフを書くことができますが，Fortran プログラムでファイルに出力した結果をプロットさせることもできます．
前回作成した，testfunc.f というプログラムを本日のディレクトリ 020709 へコピーして，下記のように変更してください．

```
      PROGRAM TESTFUNC
      PARAMETER(M=50)
      REAL A,B,X(M),Y(M),FUNC1
      INTEGER I
      WRITE(*,*)'#ENTER A,B'
      READ(*,*)A,B
      WRITE(*,*)'#A:',A,' ,B:',B              ←GNUPLOT のために#をつけておく
      WRITE(*,*)'#   X       Y'
      DO 10 I=1,M
         X(I)=0.1*I
         Y(I)=FUNC1(A,B,X(I))
 10   CONTINUE
      DO 20 I=1,M
         WRITE(*,602)X(I),Y(I)
 602     FORMAT(2F7.3)
 20   CONTINUE
      STOP
      END
C
      REAL FUNCTION FUNC1(A,B,X)
      REAL A,B,X
      FUNC1=A*SQRT(X)+B
      RETURN
      END
```

(3)入力データファイルを in1.dat を作成し，上記プログラムを下記のように実行して，out1.dat というファイルを出力してください．

```
%./testfunc < in1.dat > out1.dat
======= in1.dat の内容
2 -1
[EOF]

======= out1.dat1 の内容(GNUPLOT の読み込みファイル形式)
#ENTER A,B
 #A:   2. ,B: -1.                    ←#をつけると GNUPLOT ではコメント
 #    X       Y
  0.100  -0.368
  0.200  -0.106
  0.300   0.095
  0.400   0.265
  0.500   0.414
[以下省略]
```

(4)上記で出力したファイルを使って，GNUPLOT でグラフを作成します．上記のプログラムとは別な入力ファイル A,B を in2.dat に指定して out2.dat を作成し,out1.dat グラフに重ねて描いてみてください．

```
gnuplot> plot "out1.dat"
gnuplot> plot "out1.dat" with line
gnuplot> replot "out2.dat" w l
gnuplot> quit
```

(5)Advance testfunc.f を変更して 3 年目までは金利 0.5%，4 年目以降の金利が 1%の預金の利息が得られる場合の関数を作成し，１万円預けたときの，20 年目までの預金量をグラフにしなさい．

【数値計算で積分を実行する】

(1)Fortran を使って数値計算で簡単に定積分ができます．
定積分したい範囲を細かい短冊に分割して台形とみなして，その台形の和として積分値を求めます．
短冊への分割数を多くするほど，精度のよい積分ができます．
★台形近似したときの積分公式は以下のとおりです．
積分区間[a,b]を n 等分し，h=(b-a)/n，x_k=a+k*h とおき，f_k=f(x_k)とすると

$$\int_a^b f(x)dx \approx \sum_{k=0}^{n-1} \frac{h}{2}[f_k + f_{k+1}]$$

として求めることができます．

(2)実際に f(x)=x^2 の定積分をやってみます．[0,1]の積分範囲で積分値を求めてください．
分割数 NUM を大きくして，積分値の精度がどのように上がっていくか確かめなさい．

```
      PROGRAM INTEG
      REAL A,B,NUM,H,SUM
      WRITE(*,*)'INPUT SEKIBUN HANI (a,b)'
      READ(*,*)A,B
      WRITE(*,*)'INPUT BUNKATU SUU'
      READ(*,*)NUM
      H=(B-A)/NUM
    〔                    〕←和をとっていくときの変数 SUM を初期化する．
      DO 10 I=1,NUM
         SUM=SUM+(FUNC1((I-1)*H+A)+FUNC1(I*H+A))*H/2
 10   CONTINUE
      WRITE(*,*)'SEKIBUN:',SUM
      STOP
      END
      REAL FUNCTION FUNC1(X)
      REAL X
    〔                    〕←関数式を定義する．
      RETURN
      END
```

(3)f(x)=sin(x)に変更して[0,π]，[0,2π]で積分をしてみてください．
(4)上記プログラムを変更して，分割数 NUM を 2,4,8,16,32・・・,1024 としたときの積分値を表形式で出力しなさい．
(5) <u>Advance</u> 台形公式よりも，精度が高い（分割数が少なくても）公式として，以下のシンプソンの公式があります．[a,b]を 2n 等分すると h=(b-a)/2n となり，積分値は以下のようになります．

$$\int_a^b f(x)dx \approx \sum_{k=0}^{n-1} \frac{h}{3}[f_{2k} + 4f_{2k+1} + f_{2k+2}]$$

シンプソンの公式を使用して，積分値を求めるプログラムを作成し，上記(4)と同様に分割数 NUM を 2,4,8,16,32・・・,1024 として積分値を計算した表を作成し，台形公式より，精度が高いことを確認しなさい