

# Digital Fortran 解説

2009/April

## 1. プログラム形態とデータ構成

### ●最小自乗法プログラム (testlsm.for)

m組の実験データ  $(x_i, y_i)$  に最も近似する直線式  $(y = a x + b)$  を最小自乗法で決定する。  
入力データは組数mと、m組の  $(x_i, y_i)$  値、出力データは直線式の係数 a, bとなる。

#### ・入力データ

m=4	$(x_i, y_i)$	X=1.50	Y=1.00
		X=2.30	Y=1.30
		X=4.05	Y=2.05
		X=5.60	Y=2.60

#### ・最小自乗法 (直線近似) の計算法

$[x] = \sum x_i$ ,  $[x y] = \sum x_i y_i$  等として、正規方程式は次式で与えられる。

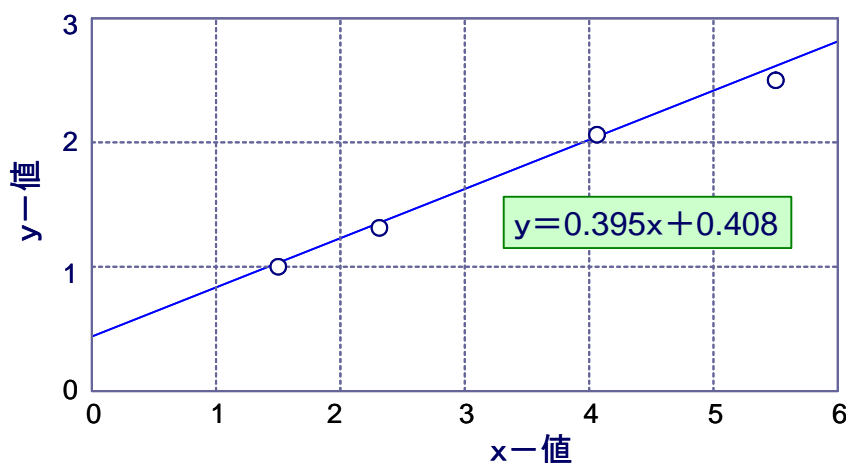
$$[x] a + m \cdot b = [y]$$

$$[x^2] a + [x] b = [x y]$$

#### ・計算結果 (出力データ)

a=0.3953    b=0.4084

#### ・結果の表示



● *testlsm.for* (例題プログラム)

```

01  C    ### testlsm.for ###
02  C    (Least Square Method)
03  C
04      DIMENSION X(100),Y(100)
05  C
06      character*48 fname0, fname50, fname6, fname51
07      write(*, ' (A)') '*** Input & Output File Name ***'
08      read(*, ' (A)') fname0
09      open(1, file=fname0)
10      read(1, *) fname50, fname6, fname51
11      close(1)
12  C
13      open(50, file=fname50)
14      open( 6, file=fname6)
15      open(51, file=fname51)
16  C
17      READ(50, *) NDATA
18      READ(50, *) (X(I), Y(I), I=1, NDATA)
19      WRITE(6, 1000) NDATA
20      WRITE(6, 1001) (I, X(I), Y(I), I=1, NDATA)
21  1000 FORMAT(1H , 'Test Program (Least Square Method)' //1H ,
22      *          ' NDATA=', I3/)
23  1001 FORMAT(1H , ' No. ', I3, 6X, ' X=', E11. 4, 4X, ' Y=', E11. 4)
24  C
25      SUMX=0.
26      SUMY=0.
27      SUMX2=0.
28      SUMXY=0.
29      DO 10 I=1, NDATA
30          SUMX=SUMX +X(I)
31          SUMY=SUMY +Y(I)
32          SUMX2=SUMX2+X(I)*X(I)
33      10 SUMXY=SUMXY+X(I)*Y(I)
34      AB=NDATA*SUMX2-SUMX*SUMX
35      AA=(NDATA*SUMXY-SUMX*SUMY)/AB
36      BB=(SUMX2*SUMY-SUMX*SUMXY)/AB
37      WRITE(6, 2000) SUMX, SUMY, AA, BB
38  2000 FORMAT(/1H , '*** SUMX=', E12. 4, 4X, ' SUMY=', E12. 4, ' ***'
39      *          //1H , ' ===>> A=', E12. 4, ' B=', E12. 4)
40      WRITE(51, *) SUMX, SUMY, AA, BB
41  C
42      close(50)
43      close(6)
44      close(51)
45  C
46      STOP
47      END

```

●用意するデータファイル

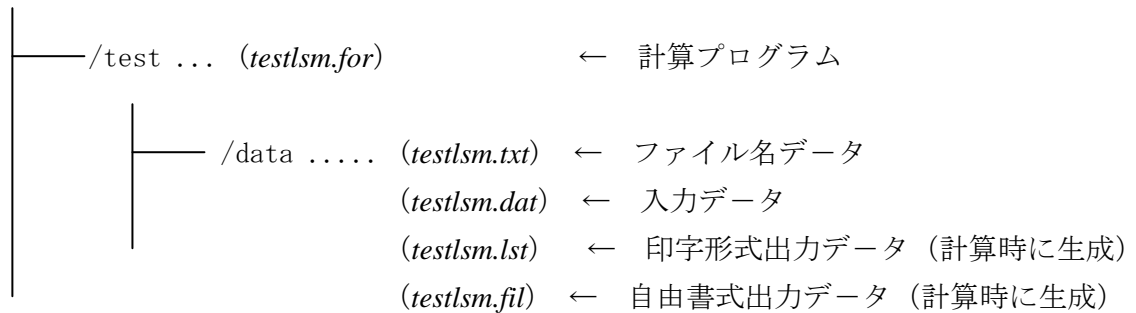
- testlsm.dat (=fname50/入力データファイル/READ(50,\*)に対応)

```
4
1.5  1.0
2.3  1.3          ※4組のデータ群 (X,Y)
4.05 2.05
5.6  2.6
```

- testlsm.txt (=fname0/入出力データファイルの名前を格納したファイル)

```
'./data/testlsm.dat'
'./data/testlsm.lst'      ※/data フォルダに3つの入出力ファイルを格納
'./data/testlsm.fil'
```

●パソコン内のフォルダ構成とファイルの格納



●計算結果の出力

- testlsm.lst (=fname6/印字出力用ファイル/WRITE(6,\*)に対応)

```
Test Program (Least Square Method)

NDATA= 4

No.  1      X= 0.1500E+01      Y= 0.1000E+01
No.  2      X= 0.2300E+01      Y= 0.1300E+01
No.  3      X= 0.4050E+01      Y= 0.2050E+01
No.  4      X= 0.5600E+01      Y= 0.2600E+01

*** SUMX= 0.1345E+02      SUMY= 0.6950E+01 ***

====>>  A= 0.3953E+00      B= 0.4084E+00
```

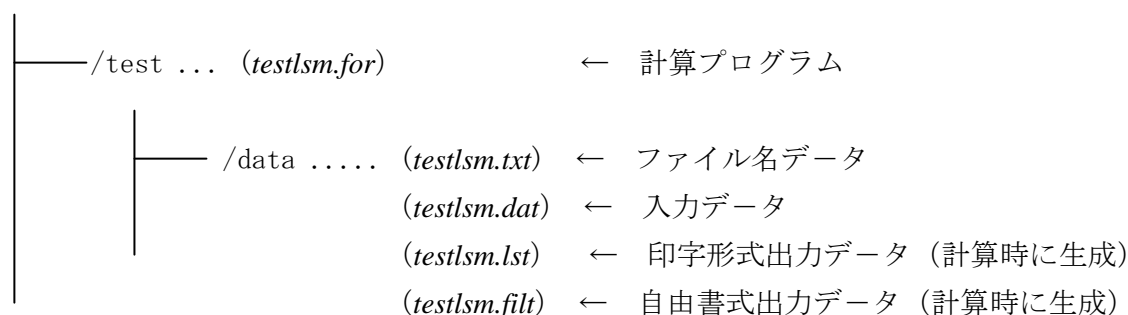
- testlsm.fil (=fname51/自由書式データ出力/WRITE(51,\*)に対応)

```
13.45000      6.950000      0.3952738      0.4083917
```

## 2. Fortran Soft “salford” の利用手順

### ①計算前の準備：

- 1) 使用する Fortran プログラム用のフォルダ（例：/test）を作成する。
- 2) /test フォルダにプログラム（例：testlsm.for）を格納する。
- 3) /test フォルダの下に、/data フォルダを作成する。
- 4) 入力データファイル（testlsm.dat）とファイル名データファイル（testlsm.txt）を作成し、/data フォルダに格納する。（データファイルの作成にはメモ帳か Word Pad を用いる）



### ②計算操作手順：

- 1) fcpad.exe をダブルクリックで起動する。
- 2) (ファイルを開く) 操作で、/test フォルダに格納したプログラム testlsm.for を読み込む。  
※プログラムの修正は、この状態で行う。
- 3) 翻訳 (make) & 実行で “\*\*\* Input & Output File Name \*\*\*” 表示の下に  
(プログラム内でキー入力を促す Input 文が置いてある)  
**./data/testlsm.txt**  
をキー入力して実行 (Return キー) を押す。
- 4) 正常終了なら “Press any key to exit (---)” の message が表示される。  
エラーが生じたら、エラーメッセージが表示される。
- 5) 計算結果 (testlsm.lst , testlsm.filt) は /data フォルダ内に出力される。

### 3. Fortran の文法 (JISBasic との対応)

#### # 書式:

- 1 行の 1~5 桁に行番号を、7~72 桁に命令文を書く。
- 1 行で書けない場合は、次行の 6 桁目に継続記号を書いて続ける。
- 注釈行 (1 桁目に C と書く、演算とは関係ない行)
- I, J, K, L, M, N で始まる変数は整数扱い。
- STOP プログラムの実行の終了
- END プログラムの最後

#### # 入出力: ファイルからデータを読み取り (入力)、ファイルに計算結果を書出す (出力)。

```
READ (50, *) NDATA
READ (50, *) (X(I), Y(I), I=1, NDATA)
WRITE (6, 1000) NDATA
WRITE (6, 1001) (I, X(I), Y(I), I=1, NDATA)
1000 FORMAT(1H, ' Test Program (Least Square Method)' //1H, ' NDATA=', I3/)
1001 FORMAT(1H, '  No. ', I3, 6X, ' X=', E11. 4, 4X, ' Y=', E11. 4)
WRITE (51, *) SUMX, SUMY, AA, BB
```

- >READ(50,\*) . . . . . 50 番で OPEN した (書式自由の) ファイルから数値を読み込む。
- >WRITE(6, 1000) . . . . . 6 番で OPEN したファイルに 1000 行の仕様 (FORMAT) で書き出す。
- >1000 FORMAT( . . . . . データ出力の書式を書く (印刷形式の出力の際に必要)

1H, 改行  
' NDATA=' 見出し語  
/ 空白行を入れる  
6X 6 桁の空白文字を入れる  
I3 整数を 3 桁で書く  
E11. 4 実数 (1.5) を指数の形 (0.1500E+01) で書く (全体 11 桁、小数部 4 桁)  
F7. 3 実数 (1.5) を通常形 (1.500) で書く (全体 7 桁、小数部 3 桁)

- >WRITE(51,\*) . . . . . 51 番で OPEN したファイルに (自由書式で) 書き出す。

#### # 演算・制御関係

- basic の LET は不要
- 四則演算 (+, -, \*, /) は同じ。べき乗 ( $x^3$  は  $x**3$  と書く)
- 関数 平方根 SQRT(X)、逆正接 ATAN(X) など違うものもあるが、大体は basic と同じ

- ・ 繰り返し計算 (basic の for I=n1 to n2 ~ next I)

```
DO 150 I=n1, n2 ~ 150 CONTINUE
```

行番号(150)までの演算を、変数 I を n1 から n2 まで変えて繰り返す

- ・ GO TO n 分岐 (行番号 n まで飛べ)

- ・ 条件分岐 (basic の if a>b then \*\*\*)

```
IF (a.GT.b) *** a>b なら *** を行え、否なら下行の演算へ
```

.EQ. : 等しい、.NE. : 等しくない、.GE. : 以上、.LE. : 以下

- ・ DIMENSION A(100), B(20, 5) 配列宣言 (プログラムの先頭に置く)

C ### ex2jis.BAS ### Example-2

<Fortran>

<JisBasic>

	40 PRINT
READ(50,*) NA	INPUT PROMPT"ファンカツ スウ (0=シュウリョウ) ":NA
	IF NA=<0 THEN STOP
DA=180/NA	LET DA=180/NA
CK=3.14159/180	LET CK=3.14159/180
S=0.0	LET S=0
DO 100 I=1,NA-1	FOR I=1 TO NA-1
A=DA*I*CK	LET A=DA*I*CK
S=S+SIN(A)	LET S=S+SIN(A)
100 CONTINUE	NEXT I
S=S*DA*CK	LET S=S*DA*CK
SE=(2.-S)/2.*100	LET SE=(2.-S)/2.*100
WRITE(6,1000) S,SE	PRINT
	PRINT USING"メンセキ=##.#### / コ`サ=##.## percent":S,SE
1000 FORMAT(/1H , "メンセキ=", E9.2, " / ", "コ`サ=", E10.3, " percent")	
STOP	GOTO 40
END	END

※Fortran で数値 (NA) を変えて計算を繰り返す場合は、NA 自体を変数にし、READ から WRITE までを DO ループとし、繰り返し計算を行う。NA に関する数値データは予め複数個用意する。