

## 初心者のためのFORTRANプログラミング II

— ステップ学習方式による —

宮 地 功\*

(昭和 53 年 4 月 28 日 受 理)

## FORTRAN Programming for Beginners II

— Stepwise Learning Method —

Isao MIYAJI

(Received April 28, 1978)

A method which make beginners learn the FORTRAN programming has been described in the previous paper. The practice results showed that the method had a marked effect. This paper deals with a method capable of being certainly learned about the FORTRAN programming in addition to the previous method.

### 1 序 論

電子計算機の急激な普及により、かなり広範囲の分野に電子計算機についての正しい理解が要請されている。特に、工学の分野を志向する技術者にとって、電子計算機を活用し、プログラムを作成することは現在必須の基礎的技術となっているようである。その為、プログラミングに関する授業は、ほとんどの工学系の学校において行なわれているようである。演習や実習を行ないながら授業を進めても、消化不良を起す学習者が多い。そこで、このような状態を少しでも改善したいと考えて、科学技術計算によく利用され、ほとんどの電子計算機に用意されているFORTRAN語を短期間に確実に修得する方法について先に報告した<sup>2),3)</sup>。この方法で、学生ごとに第2節に示した例題のような異なった問題を与えて、1年間実習してみた。その結果、実習した全員が先に報告したようにプログラム1~6の全部をコーディングし、TSSで処理し、完成した<sup>3)</sup>。このような良好な結果が得られたのは、1つにはTSSコマンドの使い方を標準化したことがあげられる<sup>4)</sup>。これらの改善の結果、初歩のFORTRANプログラミングを更に確実に修得させるプログラム作成法の開発が企画された。その際、前に報告した条件と同じもので<sup>2)</sup>、問題はプログラム1~6を作成したものと同一問題を用いることを前提として開発すること

にした。

次に開発されたプログラムの内容を概略述べる。まず、第3節において、プログラム2を基にしてプログラム6を参照して作成する「引数のあるサブルーチン副プログラム」を説明する。第4節において、プログラム7を基にしてプログラム5を参照して作成する「サブルーチン副プログラムの中に用いた関数副プログラム」を説明する。第5節において、プログラム3を基にした表形式に処理結果が印刷されるプログラム9を説明する。このプログラム9は、プログラム10~13の主プログラムの基になるプログラムである。第6節において、プログラム9に「最小値および最大値」を求める部分を付加したプログラム10を説明する。第7節において、プログラム9の大部分を主プログラムとして、「並び換え」をするサブルーチン副プログラムを付加したプログラム11を説明する。第8節において、プログラム9の大部分を主プログラムとして、「ヒストグラム」を印刷するのに必要なサブルーチン副プログラムを付加したプログラム12を説明する。最後に、第9節においては、プログラム9の大部分を主プログラムとして、「グラフ」を印刷するのに必要なサブルーチン副プログラムを付加したプログラム13を説明する。これらは、FORTRAN入門のたいていの教科書に散見される内容である。これらの節で示したプログラムは、参考文献(1)の主張を取入れ、前報よりも更にプログラムの論理構造のわかりやすい書法

\*機械工学科

に努めたものである。従って、これらのプログラムは、初心者が真似して利用するのに十分耐えるものと思う。

この一連のFORTRANプログラミング法を一步一步階段を踏みしめて昇るようにプログラミングして行くというような意味で、「ステップ学習方式によるFORTRANプログラミング」と名付けようと思う。この方式によって、従来FORTRANプログラミングの授業において消化不良を生じ、1年間に1つのプログラムさえも作成しないで授業を終えていた学習者に福音になると思うし、そのように願っている。また、従来の授業でも充分こなしていた学習者に対して、更により確実にプログラミング法が修得できると確信している。これらの事実は、実習した結果やアンケートなどからも裏付けされていることを第10節で述べる。

## 2 問 題

以下に示すプログラミングに用いた問題は、前に報告した問題と同じである<sup>2)</sup>が、説明の都合上次に再度示す。

水平と角度  $\theta$  (rad) をなす斜面上に重力  $F$  (N) の物体がある。  $F$  と  $\theta$  の値を読み込んで、式  $R = F \cos \theta$  に従って、この物体に働く垂直抗力  $R$  (N) を計算し、印刷せよ。

以下の節に述べたプログラム1~6は、参考文献<sup>3)</sup>を参照している。また、第3節以下に述べるプログラムにおいては、前に報告したプログラム作成法および文法は理解されているものと考えている。また、この続報では、新しく出現する術語およびステートメントなどは余りないが、新出のものは太字で示した。

また、ここで示したプログラムは通常の電子計算機のオペレーション・システムで処理することを予想しているため、入力装置としてカード読取装置、出力装置としてラインプリンタとし、装置番号をそれぞれ5、6とした。

## 3 引数のあるサブルーチン副プログラム

### 3.1 プログラム7

プログラム2を、読み込み、計算および印刷の3つの部分に分けて、それぞれを引数のあるサブルーチン副プログラム YOMU, KEISAN および INSATU にする。プログラム6のように、主プログラムでこれらを次々に呼び出して実行させるプログラムを作成せよ。

プログラム6では、引数のない副プログラムを用い、プログラムごとのデータの受け渡しはCOMMON文を用いて行なった。ここでは、データの受け渡しは、引数を用いて行なう。プログラム6とよく比較してみると使い方がよくわかると思う。副プログラム名は、プログラム6と同じに

する。このプログラムのフローチャートをFig.1に、プログラムおよび結果の例をFig.2に示す。

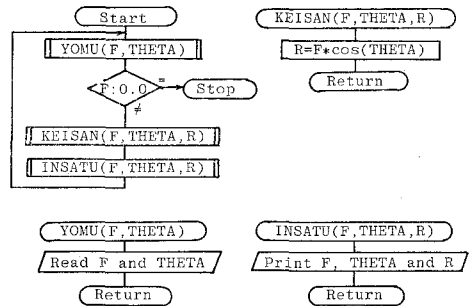


Fig. 1 Flow chart of program 7.

```

1 C *** SUICHOKU-KOPYOKU NO KEISAN (7)
2 C ... MAIN PROGRAM
3 100 CALL YOMU(F,THETA)
4 IF (F.EQ.0.0) STOP
5 CALL KEISAN(F,THETA,R)
6 CALL INSATU(F,THETA,R)
7 GO TO 100
8 END
9 C ... DATA NO YOMU
10 SURROUTINE YOMU(F,THETA)
11 READ(5,5000) F,THETA
12 5000 FORMAT(2F10.4)
13 RETURN
14 END
15 C ... R NO KEISAN
16 SURROUTINE KEISAN(F,THETA,R)
17 R=F*COS(THETA)
18 RETURN
19 END
20 C ... DATA TO KEKKA NO INSATU
21 SURROUTINE INSATU(F,THETA,R)
22 WRITE(6,6000) F,THETA,R
23 6000 FORMAT(1H,' F =',1PE15.6,' (N)'/
24 * 1H,' THETA =',E15.6,' (RAD)'/
25 * 1H,' R =',E15.6,' (N)'/)
26 RETURN
27 END
  
```

```

F = 6.789000E 00 (N)
THETA = 1.047200E 00 (RAD)
R = 3.394487E 00 (N)
  
```

```

F = 1.234500E 01 (N)
THETA = 5.236000E-02 (RAD)
R = 1.232809E 01 (N)
  
```

```

F = 1.028600E 02 (N)
THETA = 7.853999E-01 (RAD)
R = 7.273286E 01 (N)
  
```

Fig. 2 Program 7 and the results.

### 3.2 プログラム7の説明

第3行: 実引数FおよびTHETAと第10行のサブルーチン副プログラム YOMU の仮引数FおよびTHETA との対応を取ってデータを受け渡し、サブルーチン副プログラム YOMU の実行を開始させるためのCALL文である。この仮引数は実引数と同じ変数名である必要はなく、変数の型が一致しさえすればよい。詳しくは、プログラム11, 12, 13を参照するとよい。

第5, 6行: 第3行と同様に実引数F, THETA およびRとサブルーチン副プログラム KEISAN および INSATU の仮引数F, THETA およびRとの対応を取ってデータを受け渡し、サブルーチン KEISAN および INSATU の実行を開始させるためのCALL文である。

第10行：仮引数がFおよびTHETAのサブルーチン副プログラムYOMUであることを定義するSUBROUTINE文である。サブルーチン副プログラムの名前は、関数副プログラムや文関数の場合と異なり、計算結果とは何の関係もない。データの受け渡しは全部引数を通して行なわれる。

第16, 21行：それぞれ仮引数がF, THETA, およびRのサブルーチン副プログラムKEISANとINSATUであることを定義するSUBROUTINE文である。

#### 4 サブルーチン副プログラムの中での関数副プログラムの引用

##### 4.1 プログラム 8

プログラム7の、サブルーチン副プログラムKEISANの中の算術式をプログラム5と同じ関数副プログラムで定義する。その関数を用いて、計算するようにプログラム7を修正せよ。

プログラム5と7を参考にすればこのプログラム8は容易に作成できる。ここでは、サブルーチン副プログラムの中で関数副プログラムを引用している。関数副プログラムは、プログラム5とまったく同じものを利用すればよい。ここでは関数名をSUIKOとする。サブルーチン副プログラムの中で他のサブルーチン副プログラムを呼び出す例は、

```

1  C *** SUICHOKU-KORYOKU NO KEISAN (R)
2  C ... MAIN PROGRAM
3  100  CALL YOMU(F,THETA)
4      IF (F.EQ.0.0) STOP
5      CALL KEISAN(F,THETA,R)
6      CALL INSATU(F,THETA,R)
7      GO TO 100
8  END
9  C ... DATA NO YOMIKOMI
10  SUBROUTINE YOMU(F,THETA)
11  READ(5,5000) F,THETA
12  5000  FORMAT(2F10.0)
13  RETURN
14  END
15  C ... R NO KEISAN
16  SUBROUTINE KEISAN(F,THETA,R)
17  R=SUIKO(F,THETA)
18  RETURN
19  END
20  C ... DATA TO KEKKA NO INSATSU
21  SUBROUTINE INSATU(F,THETA,R)
22  WRITE(6,6000) F,THETA,R
23  6000  FORMAT(1H, ' F =',1PE15.6,' (N)'/
24        * 1H, ' THETA =',E15.6,' (RAD)'/
25        * 1H, ' R =',E15.6,' (N)'/)
26  RETURN
27  END
28  C * KANSU FUKU-PROGRAM
29  FUNCTION SUIKO(X,Y)
30  SUIKO=X*cos(Y)
31  RETURN
32  END
    
```

```

F = 6.799000E 00 (N)
THETA = 1.047200E 00 (RAD)
R = 3.394487E 00 (N)
    
```

```

F = 1.234500E 01 (N)
THETA = 5.236000E-02 (RAD)
R = 1.232800E 01 (N)
    
```

```

F = 1.028600E 02 (N)
THETA = 7.853999E-01 (RAD)
R = 7.273286E 01 (N)
    
```

Fig. 3 Program 8 and the results.

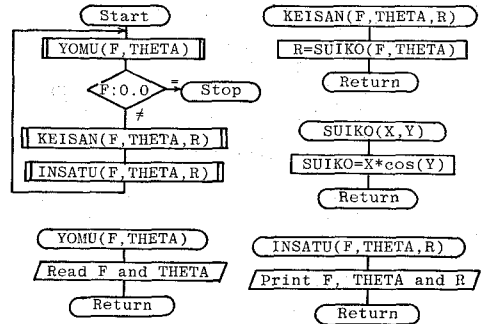


Fig. 4 Flow chart of program 8.

プログラム11で作成する。このプログラムのフローチャートを Fig.4 に、プログラムおよび結果の例を Fig.3 に示す。

##### 4.2 プログラム 8 の説明

第17行：関数副プログラムSUIKOを用いて抗力Rを計算する算術代入文である。

#### 5 処理結果の表形式印刷

##### 5.1 プログラム 9

データの読み込み部分は、データの組の数Nを読むで、N組のデータをDO型並びREAD文を用いて読み込む。計算部分は、プログラム3と同様DOループにする。印刷部分は、表題として第1行に印刷するデータの意味を、第2行にその単位を印刷する。第3行以下にデータと計算結果をDO型並びWRITE文を用いて印刷する。このようなプログラムを作成せよ。

今までのプログラムは印刷形式が全て同じである。ここでは Fig. 6 のように表の形式で結果を印刷する。すなわち、第1, 2行に表題を、第3行以降に読み込んだデータおよび計算結果を印刷する。このような形式の方が結果が見やすいが、表題の位置がずれないようにいままでより少し出力設計に注意する必要がある。このプログラム9は、この後のプログラム10~13の主プログラムの基礎となるものである。このプログラムのフローチャートを Fig.5 に、プログラムおよび結果の例を Fig.6 に示す。

##### 5.2 プログラム 9 の説明

第6行：DO型並びREAD

文である。これは、次のように書くのとほぼ

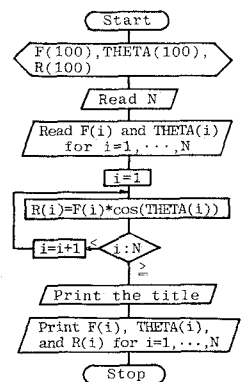


Fig. 5 Flow chart of program 9.

```

1 C *** SUICHOKU-KORYOKU NO KEISAN (9)
2 DIMENSION F(100),THETA(100),R(100)
3 C ... DATA NO YOMIKOMI
4 READ(5,5000) N
5 5000 FORMAT(I5)
6 READ(5,5010) (F(I),THETA(I),I=1,N)
7 5010 FORMAT(2F10.0)
8 C ... R NO KEISAN
9 DO 100 I=1,N
10 R(I)=F(I)*COS(THETA(I))
11 100 CONTINUE
12 C ... DATA TO KEKKA NO INSATSU
13 WRITE(6,6000)
14 6000 FORMAT(/1H,9X,'F',12X,'THETA',
15 * 12X,'R'/
16 * 1H,8X,'(N)',11X,'(RAD)',
17 * 11X,'(N)')
18 WRITE(6,6010) (F(I),THETA(I),
19 * R(I),I=1,N)
20 6010 FORMAT(1H,1P3E15.6)
21 STOP
22 END

```

| F<br>(N)     | THETA<br>(RAD) | R<br>(N)     |
|--------------|----------------|--------------|
| 6.789000E 00 | 1.047200E 00   | 3.394487E 00 |
| 1.234500E 01 | 5.236000E-02   | 1.232808E 01 |
| 1.028600E 02 | 7.853999E-01   | 7.273286E 01 |

Fig. 6 Program 9 and the results.

同じ意味である。

DO 200 I=1, N

READ(5, 5010) F(I), THETA(I)

200 CONTINUE

第13行：並びのないWRITE文である。この文は、文番号6000のFORMAT文の中にかかれていた文字を表題として印刷することを命令する。

第14~17行：表題としての文字をFig.6に示すように印刷する形式を示すFORMAT文である。ここで、9Xは文字欄記述子X変換と呼ばれ、Xの前の数値だけ空白を取ることを意味する。

第20行：F(I), THETA(I), およびR(I)の値を印刷する形式を示すFORMAT文である。3E15.6は、E15.6, E15.6, E15.6と並べて書くのと同じ意味であって、F(I), THETA(I) およびR(I)を1行に並べて表題の下に表の形式になるように、それぞれE15.6で印刷することを示す。

## 6 最小値および最大値

### 6.1 プログラム10

プログラム9の印刷部分に続いて、計算した結果の中から最小値および最大値を求め、それらの値を印刷する部分を付加したプログラムを作成せよ。

最小値および最大値は、DOループを用いて計算した結果（ここではR）全てを論理IF文を用いて比較すれば求められる。このプログラム10は、プログラム9に、この最小値AMINおよび最大値AMAXを求める部分を付加すればよい訳である。プログラム12で、この部分をサブルーチン化して用いる。このプログラムのフローチャートをFig.7に、プログラムおよび結果の例をFig.8に示す。

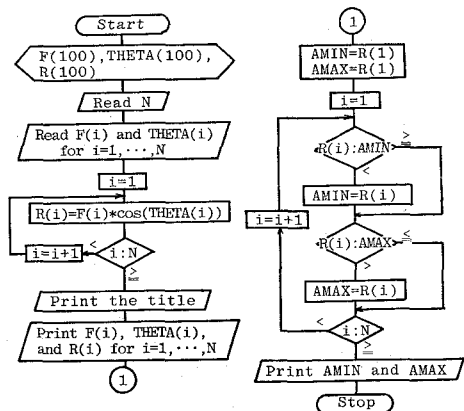


Fig. 7 Flow chart of program 10.

```

1 C *** SUICHOKU-KORYOKU NO KEISAN (10)
2 DIMENSION F(100),THETA(100),R(100)
3 C ... DATA NO YOMIKOMI
4 READ(5,5000) N
5 5000 FORMAT(I5)
6 READ(5,5010) (F(I),THETA(I),I=1,N)
7 5010 FORMAT(2F10.0)
8 C ... R NO KEISAN
9 DO 100 I=1,N
10 R(I)=F(I)*COS(THETA(I))
11 100 CONTINUE
12 C ... DATA TO KEKKA NO INSATSU
13 WRITE(6,6000)
14 6000 FORMAT(1H,9X,'F',12X,'THETA',
15 * 12X,'R'/
16 * 1H,8X,'(N)',11X,'(RAD)',
17 * 11X,'(N)')
18 WRITE(6,6010) (F(I),THETA(I),
19 * R(I),I=1,N)
20 6010 FORMAT(1H,1P3E15.6)
21 C ... SAISHO-CHI TO SAIDAI-CHI NO KEISAN
22 AMIN=R(1)
23 AMAX=R(1)
24 DO 200 I=1,N
25 IF(R(I).LT.AMIN) AMIN=R(I)
26 IF(R(I).GT.AMAX) AMAX=R(I)
27 200 CONTINUE
28 WRITE(6,6100) AMIN,AMAX
29 6100 FORMAT(/1H,9X,'R NO SAISHO-CHI =',
30 * 1P3E15.6/
31 * 1H,9X,'R NO SAIDAI-CHI =',
32 * 1P3E15.6)
33 STOP
34 END

```

| F<br>(N)     | THETA<br>(RAD) | R<br>(N)     |
|--------------|----------------|--------------|
| 6.789000E 00 | 1.047200E 00   | 3.394487E 00 |
| 1.234500E 01 | 5.236000E-02   | 1.232808E 01 |
| 1.028600E 02 | 7.853999E-01   | 7.273286E 01 |
| 3.254900E 00 | 4.719499E-01   | 2.902573E 00 |
| 5.874499E 00 | 5.470999E-01   | 8.436493E 00 |

R NO SAISHO-CHI = 2.902573E 00  
R NO SAIDAI-CHI = 8.436493E 00

Fig. 8 Program 10 and the results.

### 6.2 プログラム10の説明

第22, 23行：最小値および最大値を入れておくAMINおよびAMAXにそれぞれ初期値として、まず計算結果の最初の値R(1)を入れておく算術代入文である。

第25行：R(I)とAMINを比較して、R(I)の方が小さければ、AMINにR(I)を入れ、そうでなければ、何もしないで次のステートメントを実行する論理IF文である。

第26行: R(I) と AMAX を比較して, R(I) の方が大きければ, AMAX に R(I) を入れ, そうでなければ, 何もしないで次のステートメントを実行する論理 IF 文である。

第28行: 最小値 AMIN および最大値 AMAX を印刷させるための WRITE 文である。

第29~32行: AMIN および AMAX を印刷するときの形式を示す FORMAT 文である。

## 7 並び換え

### 7.1 プログラム11

プログラム 9 と同じようにデータを読んで, 計算し, データを読んだ順番に付番して印刷する。その後, 計算結果を選択交換法によって大きい順に並び換え, 順位と読み込んだ順番を付けて大きい順に印刷するプログラムを作成せよ。ただし, 並び換えの部分は, サブルーチン副プログラム名 NARABI として, これを主プログラムで呼び出すようにせよ。

データを大きい順に並び換える方法はいくつかあるが, ここでは, 最大値を選び出してはデータを交換して行く方法 (選択交換法または入替え法) を用いる。この並び換えの部分は, サブルーチン副プログラム NARABI とする。選択交換法を一般的に説明すると,  $a_i$  と  $a_j$  ( $i < j$ ) と比較して,  $a_i \geq a_j$  ならば何もしないで,  $a_i < a_j$  ならば  $a_i$  と  $a_j$  を入れ換える。このような比較を任意の  $i$  ( $1 \leq i \leq n-1$ ) に対して,  $j = i+1, \dots, n$  について行ない, 同様の操作を繰り返す。ここで,  $n$  はデータの個数である。

また, データに Fig. 10 のように番号 NO を付けて, 何番のデータが何番目に大きいかわかるようにする。従っ

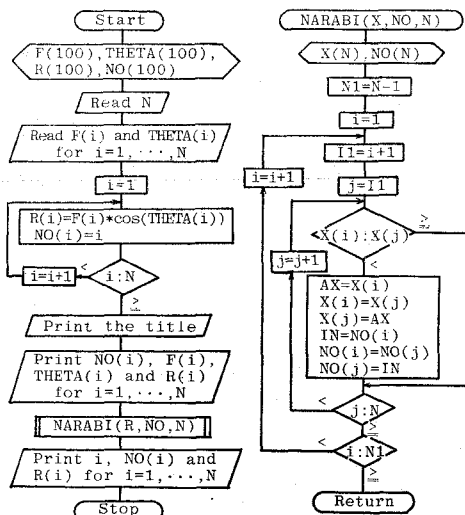


Fig. 9 Flow chart of program 11.

て, 並び換えの際, NO も同時に入れ換える必要がある。

このプログラム11以降のプログラムでは, 主プログラムのみそれぞれの問題に応じて作成すればよい。従って, サブルーチン副プログラムは例題のプログラムのままを使用すればよい。また, サブルーチン副プログラム NARABI は, プログラム13において, 判定の部分のみ修正して用いる。このプログラムのフローチャートを Fig. 9 に, プログラムおよび結果の例を Fig. 10 に示す。

```

1  C *** SUICHOKU-KORYOKU NO KEISAN (11)
2  DIMENSION F(100),THETA(100),
3  R(100),NO(100)
4  C ... DATA NO YOMIKOMI
5  READ(5,5000) N
6  5000 FORMAT(I5)
7  READ(5,5010) (F(I),THETA(I),I=1,N)
8  5010 FORMAT(2F10.0)
9  C ... KEISAN
10 DO 200 I=1,N
11 R(I)=F(I)*COS(THETA(I))
12 NO(I)=I
13 200 CONTINUE
14 C ... DATA TO KEKKA NO INSATSU
15 WRITE(6,6000)
16 6000 FORMAT(/1H,3X,'NO.',6X,'F',7X,
17 'THETA',7X,'R',/
18 '1H,11X,'(N)',6X,'(RAD)',6X,'(N)')
19 WRITE(6,6010) (NO(I),F(I),THETA(I),
20 R(I),I=1,N)
21 6010 FORMAT(1H,15,2X,1P3E10.3)
22 C ... NARABI-KAE
23 CALL NARABI(R,NO,N)
24 C ... OHKII-JUN NI KEKKA WO INSATSU-SURU
25 WRITE(6,6100)
26 6100 FORMAT(/1H,'JUN-I',4X,'NO.',8X,
27 'R(N)')
28 WRITE(6,6110) (I,NO(I),R(I),I=1,N)
29 6110 FORMAT(1H,15,1X,15,2X,1PE15.6)
30 STOP
31 END
32 C *** OHKII-JUN NI NARABI KAERU
33 SURROUTINE NARABI(X,NO,N)
34 DIMENSION X(N),NO(N)
35 N1=N-1
36 DO 110 I=1,N1
37 I1=I+1
38 DO 100 J=I1,N
39 IF(X(I1).GE.X(J)) GO TO 100
40 AX=X(I)
41 X(I)=X(J)
42 X(J)=AX
43 IN=NO(I)
44 NO(I)=NO(J)
45 NO(J)=IN
46 100 CONTINUE
47 110 CONTINUE
48 RETURN
49 END
50

```

| NO. | F<br>(N)  | THETA<br>(RAD) | R<br>(N)  |
|-----|-----------|----------------|-----------|
| 1   | 6.789E 00 | 1.047E 00      | 3.394E 00 |
| 2   | 1.234E 01 | 5.236E -02     | 1.233E 01 |
| 3   | 1.029E 02 | 7.854E -01     | 7.273E 01 |
| 4   | 3.259E 00 | 4.720E -01     | 2.908E 00 |
| 5   | 9.876E 00 | 5.471E -01     | 8.435E 00 |

| JUN-I | NO. | R(N)         |
|-------|-----|--------------|
| 1     | 3   | 7.273286E 01 |
| 2     | 2   | 1.232808E 01 |
| 3     | 5   | 8.434893E 00 |
| 4     | 1   | 3.394487E 00 |
| 5     | 4   | 2.902573E 00 |

Fig. 10 Program 11 and the results.

### 7.2 プログラム11の説明

第12行: データに番号を付けて印刷するために, 配列名 NO に順番に番号 1~N を入れる算術代入文である。

第20, 21行: NO(I), F(I), THETA(I), および R(I) を印刷

させるDO型並びWRITE文である。

第22行：1行にNO(I)をI変換I5で、他のF(I), THETA(I) および R(I) をE変換E10.3で印刷することを示すFORMAT文である。

第24行：データを並び換えるサブルーチン副プログラムNARABIの実行を開始させるためのCALL文である。実引数R, NOおよびNは、それぞれ計算して求められた抗力の記憶されている配列名、そのデータの番号の配列名およびデータの個数を意味する。

第26行：'JUN-I', 'NO', および 'R(N)' という文字を印刷させるための並びのないWRITE文である。

第27, 28行：第26行で説明した文字を印刷する形式を示したFORMAT文である。

第29行：順位I, データ番号NO(I), および結果R(I)を印刷させるためのDO型並びWRITE文である。

第30行：IおよびNO(I)をI変換I5で、R(I)をE変換E15.6で印刷することを示したFORMAT文である。

第34行：以下に続くプログラムがデータを並び換えるためのサブルーチン副プログラムNARABIであることを定義するSUBROUTINE文である。仮引数は、データの配列X, データ番号NO, およびデータの個数Nである。

第35行：確保すべき記憶場所の数(配列の次元の寸法)を変数名Nで与えるDIMENSION文である。これを整合配列という。このようにして用いる配列名(この例ではXおよびNO) および配列の各次元の寸法を指示するのに用いる変数名(この例ではN)は全て、仮引数として定義しておかなければならない。

第37行：制御変数I, 初期値パラメータ1, 終値パラメータN1のDO文である。このDO文は、2重DOループの外側のループの最初の行となっている。

第39行：制御変数J, 初期値パラメータI1, 終値パラメータNのDO文である。このDO文は、内側のDOループの最初の行になっている。

第40行： $X(I) \geq X(J)$  の場合は何もしないで内側のDOループの終端に行き、 $X(I) < X(J)$  の場合は、第41~46行の置き換えを行なうことを判定する論理IF文である。

第41~43行： $X(I)$  が  $X(J)$  よりも小さい時にこれらを置き換えるための算術代入文である。

第44~46行： $X(I)$  と  $X(J)$  を置き換える時に、それらと対応しているデータ番号NOも一緒に置き換える必要がある。それで、番号NO(I)とNO(J)を置き換えるための算術代入文である。

## 8 ヒストグラムの印刷

### 8.1 プログラム12

プログラム9と同じように、データを読んで計算し、結果を印刷する。結果の値の全範囲をKUKAN個の区間に分け、各区間に入る結果の値のヒストグラム(度数分布グラフ)を印刷するプログラムを作成せよ。このヒストグラムを印刷する部分および、結果の値の範囲を求める部分は、それぞれサブルーチン副プログラムHISTO および MINMAXとする。

プログラム9に、サブルーチン副プログラムMINMAXおよびHISTOを呼び出すCALL文を付加して主プログラムとする。ヒストグラムを印刷する部分をサブルーチン副プログラムHISTOとする。ヒストグラムの下限値と上限値を計算した結果の値の最小値AMINと最大値AMAXと

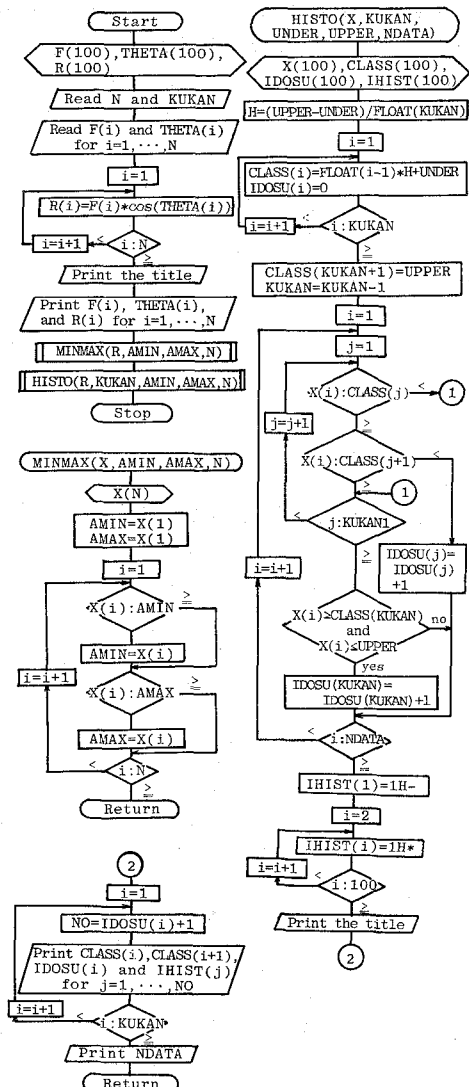


Fig. 11 Flow chart of program 12.

する。ここではプログラム10の最小値および最大値を求める部分をサブルーチン副プログラムにして主プログラムで呼び出すようにする。下限値と上限値、区間の数からきざみ $h$  (区間の幅) を求める。区間の数 KUKAN は始めに主プログラムで読み込む。サブルーチン副プログラム HISTO の

```

1 C *** SUICHOKU-KORYOKU NO KEISAN (12)
2 DIMENSION F(100),THETA(100),R(100)
3 C ... DATA NO YOMIKOMI
4 READ(5,5000) N,KUKAN
5 5000 FORMAT(2I5)
6 READ(5,5010) (F(I),THETA(I),I=1,N)
7 5010 FORMAT(6F10.0)
8 C ... R NO KEISAN
9 DO 100 I=1,N
10 R(I)=F(I)*COS(THETA(I))
11 100 CONTINUE
12 C ... DATA TO KEKKA NO INSATSU
13 WRITE(6,6000)
14 6000 FORMAT(/1H ,9X,'F',12X,'THETA',
15 * 12X,'R',
16 * 1H ,8X,'(N)',11X,'(RAD)',
17 * 11X,'(N)')
18 WRITE(6,6010) (F(I),THETA(I),
19 * R(I),I=1,N)
20 6010 FORMAT(1H ,1P3E15.6)
21 C ... SAIDAI-CHI TO SAISHO-CHI NO KEISAN
22 CALL MINMAX(R,AMIN,AMAX,N)
23 C ... HISTOGRAM NO INSATSU
24 CALL HISTO(F,KUKAN,AMIN,AMAX,N)
25 STOP
26 END
27 C *** HISTOGRAM NO INSATSU
28 SUBROUTINE HISTO(X,KUKAN,UNDER,
29 * UPPER,NDATA)
30 DIMENSION X(100),CLASS(100),
31 * IDOSU(100),IHIST(100)
32 C ... KAKU-KUKAN NO DOSU NO KEISAN
33 H=(UPPER-UNDER)/FLOAT(KUKAN)
34 DO 100 I=1,KUKAN
35 CLASS(I)=FLOAT(I-1)*H+UNDER
36 IDOSU(I)=0
37 100 CONTINUE
38 CLASS(KUKAN+1)=UPPER
39 KUKAN1=KUKAN+1
40 DO 130 J=1,NDATA
41 DO 110 I=1,KUKAN1
42 IF(X(I).LT.CLASS(J)) GO TO 110
43 IF(X(I).LT.CLASS(J+1)) GO TO 120
44 110 CONTINUE
45 IF(X(I).GE.CLASS(KUKAN) .AND.
46 * X(I).LE.UPPER) IDOSU(KUKAN)=
47 * IDOSU(KUKAN)+1
48 GO TO 130
49 IDOSU(J)=IDOSU(J)+1
50 130 CONTINUE
51 C ... HISTOGRAM NO INSATSU
52 IHIST(1)=1H-
53 DO 200 I=2,100
54 IHIST(I)=1H*
55 200 CONTINUE
56 WRITE(6,6000) (I,I=1,4)
57 6000 FORMAT(/1H ,29X,'**** HISTGRAM',
58 * '****',
59 * 1H ,3X,'KAGEN',8X,'JOGEN',
60 * 1H ,3X,'DOSU',2X,'(5X,I5)/',
61 * 1H ,29X,'(0-----5-----)',0')
62 DO 220 I=1,KUKAN
63 NO=IDOSU(I)+1
64 WRITE(6,6010) CLASS(I),CLASS(I+1),
65 * IDOSU(I),
66 * (IHIST(J),J=1,10)
67 6010 FORMAT(1H ,1PE9.2,' ...',E9.2,
68 * 15,1X,41A1/,
69 * (1H ,29X,40A1))
70 220 CONTINUE
71 WRITE(6,6020) NDATA
72 6020 FORMAT(1H ,9X,'GOKETI',8X,I5)
73 RETURN
74 END
75 C *** SAISHO-CHI TO SAIDAI-CHI
76 SUBROUTINE MINMAX(X,AMIN,AMAX,N)
77 DIMENSION X(N)
78 AMIN=X(1)
79 AMAX=X(1)
80 DO 100 J=1,N
81 IF(X(J).LT.AMIN) AMIN=X(J)
82 IF(X(J).GT.AMAX) AMAX=X(J)
83 100 CONTINUE
84 RETURN
85 END

```

Fig. 12 Program 12.

中で各区間に入る結果の値の度数を求め、それらをヒストグラムにして印刷する。このプログラムのフローチャートを Fig.11 にプログラムを Fig.12 に、結果の例を Fig.13 に示す。

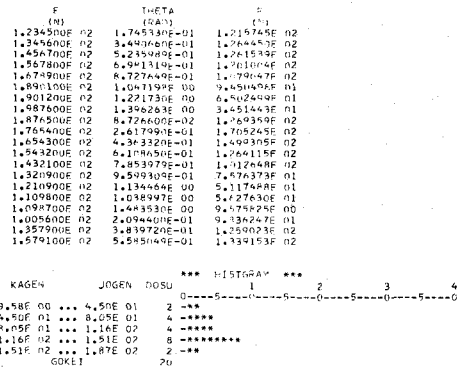


Fig. 13 The results of program 12.

## 8.2 プログラム12の説明

第4行：データの組の個数Nおよびヒストグラムを描く時に必要な区間の数 KUKAN を読み込むための READ 文である。

第7行：F(I)と THETA(I) をそれぞれ F 変換 F10.0 で、1 行に 3 組読み込むことを示した FORMAT 文である。

第22行：結果の値Rの範囲を求めるためのサブルーチン副プログラム MINMAX の実行を開始させるための CALL 文である。計算した結果の値R、データの組の個数N を実引数としてサブルーチン副プログラム MINMAX に引き渡し、Rの最小値および最大値をそれぞれ AMIN および AMAX に持ち帰る。

第24行：ヒストグラムを印刷するためのサブルーチン副プログラム HISTO の実行を開始させるための CALL 文である。計算した結果の値R、区間の数 KUKAN、R の最小値 AMIN、R の最大値 AMAX、およびデータの組の個数N を実引数としてサブルーチン副プログラム HISTO に引き渡す。

第28行：以下に続くプログラムがヒストグラムを印刷するためのサブルーチン副プログラム HISTO であることを定義する SUBROUTINE 文である。仮引数は、データの配列名X、区間の数 KUKAN、ヒストグラムの下限值 UNDER、上限値 UPPER、およびデータの組の数 NDATA である。

第33行：ヒストグラムの各階級の幅Hを求めるための算術代入文である。FLOAT は組込み関数であって、整数型の引数 (この例では KUKAN) を実数化する。

第35行：各階級の下限と上限を配列 CLASS に入れておくための算術代入文である。

第36行：各階級の度数を入れておくための配列 IDOSU をクリアするため、0を入れておく算術代入文である。

第38行：最後の階級の上限にデータの最大値 UPPER を入れておくための算術代入文である。

第42, 43行：各階級の度数を求めるための論理 IF 文である。ここで、CLASS (J+1) のように添字が添字式になっている。

第45～47行：最後の階級の度数を求めるための論理IF文である。最後の階級のみ上限値に等しいデータも度数として数えるために、別の論理IF文で求めるようにしてある。この論理IF文で用いている .AND. は、論理演算子と呼ばれる。

第52行：配列の IHIST (1) に文字 'ー' を入れておくための文字代入文である。ヒストグラムでは、Fig. 13 に示すように、この 'ー' に続いて度数だけ '\*' を印刷する。

第54行：度数だけ '\*' を印刷するために、配列 IHIST に '\*' を入れておくための文字代入文である。文字定数を代入するためには、このようなH変換を用いる。Hの前に文字数 ( $\leq 2$ ) を書き、Hの後に代入したい文字を書く。

第56行：Fig. 13のように目盛を印刷するための WRITE 文である。

第61行：4 ('0-----5-----') はかっこの中の内容を4回書くのと同じ意味である。

第63行：'ー' を印刷して、度数だけ '\*' を印刷するので、度数に1を加えた個数を NO とするための算術代入文である。

第64行：階級の下限 CLASS (I), 上限 CLASS (I+1), 度数 IDOSU (I), および '\*' の入っている IHIST (J) を度数だけ印刷するための WRITE 文である。Fig. 13 からわかるように度数は1行に40個分印刷できる。40を越える場合は、次の行に続けて印刷できるように、第69行のように繰り返す欄記述子をかっこでくくってある。

第67～69行：CLASS (I) と CLASS (I+1) を E 変換 E9.2 で、IDOSU (I) を I 変換 I5 で、IHIST を文字欄記述子 A 変換 A1 で印刷し、度数が40を越えたならば次の行に続けて印刷できることを示した FORMAT 文である。文字の入出力には A 変換を用いる。通常 8 bit で1文字記憶することができる。

第71行：データの数 NDATA を合計として印刷するための WRITE 文である。

第76行：以下に続くプログラムが最小値および最大値を求めるサブルーチン副プログラム MINMAX であることを定義する SUBROUTINE 文である。仮引数は、データ X, その最小値 AMIN, その最大値 AMAX, およ

びデータの個数 N である。

第77行：X を整合配列として N 個の記憶場所を確保するための DIMENSION 文である。

第78～83行：プログラム12の第22～27行と同じことを行なう。

## 9 グラフの印刷

### 9.1 プログラム13

与えられた問題の算術式をある1つの変数 (X とする) についてのみ変化する関数とする (1 変数の関数にする)。他の変数には適当な一定値を与え、これをプログラム 4 と同じ文関数名で定義する。変数 X の値を N 個読んで、それぞれの X に対応する関数値 Y を文関数によって計算する。それらの点 (X, Y) のグラフを、31行56列のほぼ正方形の中にプロットするプログラムを作成せよ。ただし、縦軸に変数 X をとり、横軸に関数値 Y をとる。X 軸は5間隔に、Y 軸は最小、最大およびその中間に目盛を印刷する。また、データを並び換える部分、データを印刷する部分、グラフを印刷する部分および X, Y 軸の最小、最大を求める部分をそれぞれサブルーチン副プログラム NARABI, DATA, GRAPH, および MINMAX とする。

計算された結果をグラフに表示すると単に印刷された数値を見るのと違って、結果に対する理解度は飛躍的に増大する。ここではグラフをラインプリンター用紙に表示する方法を練習する。このプログラムは、いままでのプログラミングの最後の仕上げとして総合問題となっている。主プログラムと4つのサブルーチン副プログラム DATA, NARABI, GRAPH および MINMAX から構成されている。ここで描くグラフは、例題の算術式で  $F=10.0$  として R が THETA のみについて変化するかと考える。従って、文関数を、 $SUIKO(X)=10.0 * \cos(X)$  として、X を変化させて、この関数関係をプロットさせることにする。主プログラムで、サブルーチン副プログラム DATA と GRAPH を呼び出している。DATA は、データを印刷するためのサブルーチン副プログラムである。また、DATA では、読み込んだデータ X と計算した結果 Y を対応づけて、X の値の小さい順に印刷する。1行にこれらを2組印刷するために、入力したデータ数 N が奇数か偶数かによって最後の行の印刷が異なる。また、データ X の値の小さい順に印刷するために、サブルーチン副プログラム DATA の中でサブルーチン副プログラム NARABI を呼び出している。この NARABI は、プログラム13の NARABI と少し異なっている。グラフを印刷するためのサブルーチン副プログラム GRAPH では、X, Y の範囲を知り、両端に目盛を印刷するために、

最小値および最大値を求めるサブルーチン 副プログラム MINMAX を呼び出している。点(X, Y)をプロットするための配列 IPLOT は、31×56の2次元に取り、X, Y 両軸方

向にそれぞれ31行、56桁のほぼ正方形の中にグラフを印刷するようにしてある。これらの範囲に入るように両軸にどの程度縮尺(倍尺)すればよいかを計算する必要がある。配列の添字は整数型でなければならない。そのため、実数値(X, Y)を整数値(IX, IY)に変換する必要がある。この時、四捨五入が行なわれるように考慮しなければならない。このプログラムのフローチャートを Fig.14 に、プログラムを Fig.15 に、その結果の例を Fig.16 に示す。

## 9.2 プログラム13の説明

第3行: Fを定数10.0とし、THETA について変わる関数を考え、THETA に対応する仮引数をXとした文関数定義文である。文関数定義文の作り方はプログラム4を参考にするとい。

第8行: Xに対応するTHETAのデータを1行に7個F変換F10.0の形式で読み込むことを知らせるFORMAT文である。

第9行: データの表題を印刷するのに必要な文字を配列MOJIに読み込むためのREAD文である。このMOJIの内容は、サブルーチンDATAに引数を介して引き渡す。

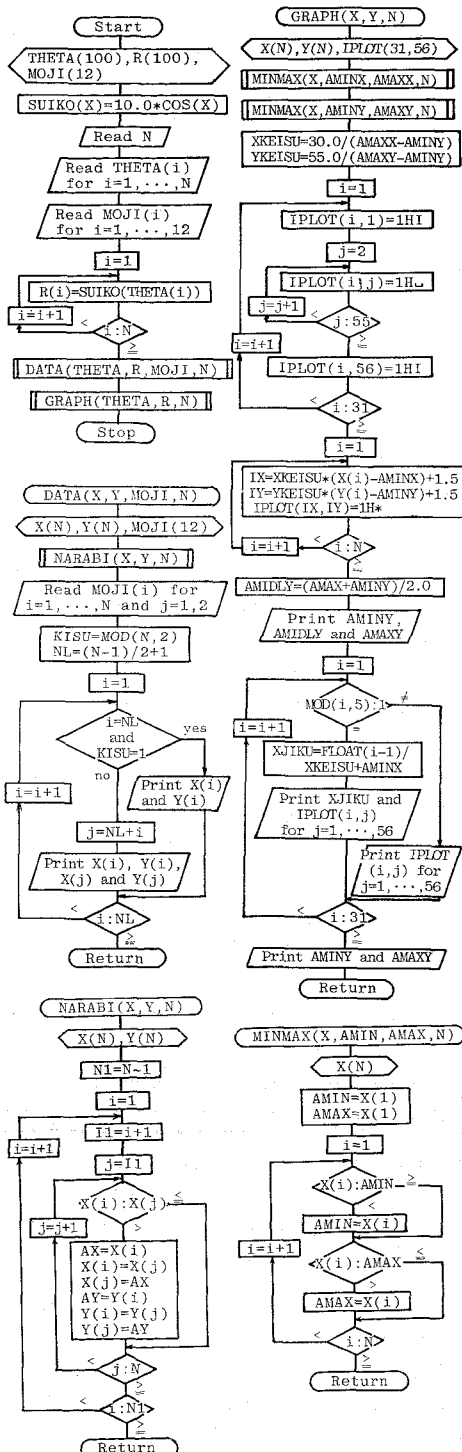


Fig. 14 Flow chart of program 13.

```

1  C *** SUICHOKU-KORYOKU NO KEISAN  (13)
2  DIMENSION THETA(100),R(100),MOJI(12)
3  SUIKO(X)=10.0*COS(X)
4  C ... DATA NO YOMIKOMI
5  READ(5,5000) N
6  5000 FORMAT(15)
7  READ(5,5010) (THETA(I),I=1,N)
8  5010 FORMAT(7F10.6)
9  READ(5,5020) (MOJI(I),I=1,12)
10 5020 FORMAT(12A2)
11 C ... R NO KEISAN
12 DO 100 I=1,N
13   R(I)=SUIKO(THETA(I))
14 100 CONTINUE
15 C ... DATA NO INSATSU
16 CALL DATA(THETA,R,MOJI,N)
17 C ... GRAPH NO INSATSU
18 CALL GRAPH(THETA,R,N)
19 STOP
20 FND
21 C *** DATA NO INSATSU
22 SUBROUTINE DATA(X,Y,MOJI,N)
23 DIMENSION X(N),Y(N),MOJI(12)
24 CALL NARABI(X,Y,N)
25 WRITE(6,6000) ((MOJI(I),I=1,12),J=1,2)
26 6000 FORMAT(/,1H,2(3X,6A2),10X,2(3X,6A2))
27 KISU=MOD(N,2)
28 NL=(N-1)/2+1
29 DO 110 I=1,NL
30   IF(I.EQ.NL.AND.KISU.EQ.1) GO TO 100
31   J=NL+I
32   WRITE(6,6010) X(I),Y(I),X(J),Y(J)
33   6010 FORMAT(1H,1P2F15.6,10X,2E15.6)
34   GO TO 110
35 100 WRITE(6,6020) X(I),Y(I)
36 6020 FORMAT(1H,1P2F15.6)
37 110 CONTINUE
38 RETURN
39 END
40 C *** CHIISAI JUN NI NARABI KAERU
41 SUBROUTINE NARABI(X,Y,N)
42 DIMENSION X(N),Y(N)
43 NI=N-1
44 DO 110 I=1,NI
45   I1=I+1
46   DO 100 J=I1,N
47     IF(X(I).LE.X(J)) GO TO 100
48     AX=X(I)
49     X(I)=X(J)
50     X(J)=AX
51     AY=Y(I)
52     Y(I)=Y(J)
53     Y(J)=AY
54 100 CONTINUE
55 110 CONTINUE
56 RETURN
57 END

```

Fig.15 (a) Program 13 (part).

```

58 C *** GRAPH NO INSATSU
59 SUBROUTINE GRAPH(X,Y,N)
60 DIMENSION X(N),Y(N),IPLT(31,56)
61 C ... SHOKI-SETTEI
62 CALL MINMAX(X,AMINX,AMAXX,N)
63 CALL MINMAX(Y,AMINY,AMAXY,N)
64 XKEISU=30.0/(AMAXX-AMINX)
65 YKEISU=55.0/(AMAXY-AMINY)
66 DO 110 I=1,31
67     IPLT(I,1)=1H
68     DO 100 J=2,55
69         IPLT(I,J)=1H
70     100 CONTINUE
71     IPLT(I,56)=1H
72 110 CONTINUE
73 DO 120 I=1,N
74     IX=XKEISU*(X(I)-AMINX)+1.5
75     IY=YKEISU*(Y(I)-AMINY)+1.5
76     IPLT(IX,IY)=1H*
77 120 CONTINUE
78 C ... GRAPH NO INSATSU
79 AMIDLY=(AMAXY+AMINY)/2.0
80 WRITE(6,6000) AMINY,AMIDLY,AMAXY
81 6000 FORMAT(/,1H ,1PE15.2,11X,E15.2,
82 *      13X,E15.2/
83 *      1H ,9Y,11('I---'),*I'/
84 DO 210 I=1,31
85     IF(MOD(I,5).EQ.1) GO TO 200
86     WRITE(6,6010) (IPLT(I,J),J=1,56)
87 6010 FORMAT(1H ,9X,56A1)
88     GO TO 210
89 200 XJIKU=FLOAT(I-1)/XKEISU+AMINX
90     WRITE(6,6020) XJIKU,(IPLT(I,J),
91 *      J=1,56)
92 6020 FORMAT(1H ,1PE9.2,56A1)
93 210 CONTINUE
94 WRITE(6,6030) AMINY,AMIDLY,AMAXY
95 6030 FORMAT(1H ,9X,11('I---'),*I'/
96 *      1H ,1PE15.2,11X,E15.2,
97 *      13X,E15.2)
98 RETURN
99 END
100 C *** SAISHO-CHI TO SAIDAI-CHI
101 SUBROUTINE MINMAX(X,AMIN,AMAX,N)
102 DIMENSION X(N)
103 AMIN=X(1)
104 AMAX=X(1)
105 DO 100 I=1,N
106     IF(X(I).LT.AMIN) AMIN=X(I)
107     IF(X(I).GT.AMAX) AMAX=X(I)
108 100 CONTINUE
109 RETURN
110 END

```

Fig.15 (b) Program 13 (continuation).

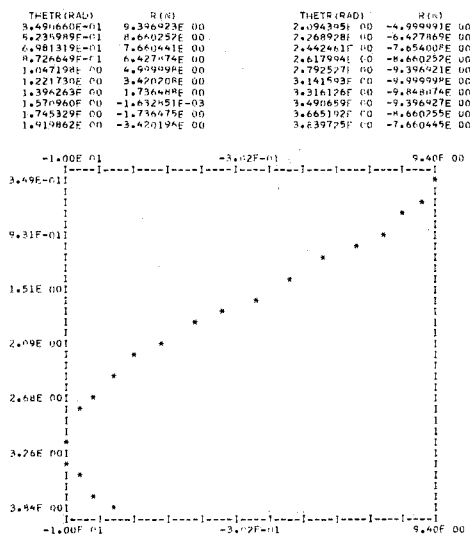


Fig.16 The results of program 13.

第10行: MOJI(1)から MOJI(6)までにXに対応するデータの表題を12文字以内で, MOJI(7)から MOJI(12)までにYに対応するデータの表題を同じく12文字以内でA変換A2で入力することを示す FORMAT 文である。12文字より少ない表題の場合は, 12文字分の両端に同じだけの空白を取り, 表題が中央にくるように文字データを入力する。A2は, 文字2文字を MOJI(I)に読み込むことを意味する。

第16行: データを印刷するサブルーチン副プログラム DATAを実行させるための CALL 文である。Xに対応する THETA, Yに対応する R, これらの表題を印刷するための MOJI, およびデータの個数Nを実引数としている。

第18行: グラフを印刷させるサブルーチン副プログラム GRAPHの実行を開始させるための CALL 文である。Xに対応する THETA, Yに対応する R, およびデータの個数Nを実引数としている。

第22行: 以下に続くプログラムがデータを印刷するためのサブルーチン副プログラム DATAであることを定義する SUBROUTINE 文である。仮引数は, x, y 座標のデータ X, Y, 表題を印刷するための MOJI, およびデータの個数Nである。

第23行: 配列 X, Y がN個の記憶場所を確保する整合配列であり, 配列 MOJI が12個の記憶場所を確保することを宣言する DIMENSION 文である。

第24行: データを小さい順に並び替えるためにサブルーチン副プログラム NARABI の実行を開始させるための CALL 文である。

第25行: MOJI に記憶されている表題を Fig.16のように1行に2組印刷させるための WRITE 文である。これに合わせて, 32行の WRITE 文でデータも1行に Fig.16のように2組印刷する。

第26行: 表題を印刷する形式を示す FORMAT 文である。ここで, 2(3X, 6A2)は, 3X, 6A2, 3X, 6A2を省略した書き方である。

第27行: データの個数Nが奇数の場合 KISU に1を入れ, 偶数の場合 KISU に0を入れるための算術代入文である。MOD は, 剰余を求める組込み関数である。

第28行: 印刷する行数 NL を求めるための算術代入文である。

第30行: 印刷の最後の行 (I=NL) で, データの個数が奇数 (KISU=1) であるかどうかを判定するための論理IF文である。この関係式が真の場合は, 最後の行は第35行の WRITE 文を用いて1組のデータのみ印刷することになる。

第32行: データXとYを1行に2組印刷するためのWRITE

文である。最初に  $X(I)$  と  $Y(I)$  を、その隣りに  $X(J)$  と  $Y(J)$  を印刷させる。ただし、 $J$  は  $J=NL+1$  とその前の行で与えられている。

第41行：以下に続くプログラムがデータの並び換えをするサブルーチン副プログラム **NARABI** であることを定義する **SUBROUTINE** 文である。仮引数は、 $x, y$  座標のデータ  $X, Y$ 、およびデータの個数  $N$  である。

第42行： $X$  と  $Y$  を整合配列として宣言するための **DIMENSION** 文である。

第47行： $X(I) \leq X(J)$  の場合は、何もしないで内側の **DO** ループの終端に行き、 $X(I) > X(J)$  の場合は、第48～53行の置き換えを行なうための判定をする論理 **IF** 文である。

第48～50行： $X(I) > X(J)$  の場合、これらを置き換えるための算術代入文である。

第51～53行： $Y(I)$  と  $X(J)$  を置き換えるときに、それらと対応している  $Y(I)$  と  $Y(J)$  も一緒に置き換える必要があるため、これらは  $Y(I)$  と  $Y(J)$  を置き換えるための算術代入文である。

第59行：以下に続くプログラムがグラフを印刷するサブルーチン副プログラム **GRAPH** であることを定義する **SUBROUTINE** 文である。仮引数は、 $x, y$  座標のデータ  $X, Y$ 、およびデータの個数  $N$  である。

第60行：この **DIMENSION** 文により配列  $X$  と  $Y$  は整合配列で、それぞれ  $N$  個の記憶場所が確保される。また、配列 **IPLOT** はグラフを描くために縦を  $x$  軸、横を  $y$  軸として、それぞれ  $x$  軸方向に31個、 $y$  軸方向に56個の記憶場所が確保される。

第62, 63行：最小値および最大値を求めるサブルーチン副プログラム **MINMAX** の実行を開始させるための **CALL** 文である。最初の行の **CALL** 文の実引数は、データの配列  $X$ 、その最小値 **AMINX**、その最大値 **AMAXX**、およびデータの個数  $N$  である。次の行の **CALL** 文の実引数は、データの配列  $Y$ 、その最小値 **AMINY**、その最大値 **AMAXY**、およびデータの個数  $N$  である。

第64, 65行： $x, y$  軸のそれぞれの1区間の長さの逆数 **XKEISU** と **YKEISU** を求めるための算術代入文である。今の場合、プロットする範囲は、 $X, Y$  軸についてそれぞれ31, 56であるから区間の数は、それぞれ30, 55となる。

第66～72行：グラフを印刷させるための配列 **IPLOT** の点  $(X, Y)$  に相当する場所に '\*' を入れる前に、クリアして空白にすると共に、**Fig.16** のように両端に棒として 'I' を入れておくための **DO** ループである。

第73～77行：点  $(X, Y)$  に対応する  $N$  個の場所に '\*' を入れるための **DO** ループである。配列の添字は整数でな

ければならないので、'\*' を配列 **IPLOT** のどこに入れるかを知るためには、点  $(X, Y)$  を整数化しなければならない。そこで、点  $(X, Y)$  を縮尺 (倍尺) すると共に整数化した点を、 $(IX, IY)$  とする。従って、**IPLOT** ( $IX, IY$ ) に '\*' を入れればよい。ところで、 $IX$  と  $IY$  を求める算術式において、最後に1.5を加えている。そのうち、1.0は配列の添字が1以上の整数でなければならないので第1項が0になる時のために加え、残りの0.5は実数を整数化する際に小数点以下が切り捨てとなるので四捨五入するために加えてある。

第80, 94行： $y$  軸方向の最小値 **AMINY**、中間値 **AMIDLY**、最大値 **AMAXY**、および棒を印刷するための **WRITE** 文である。

第85行：5行ごとに  $x$  軸の座標を印刷する **WRITE** 文 (第90行) を実行するため、制御変数  $I$  を5で割った剰余が1かどうかを判定するための論理 **IF** 文である。

第86行：グラフの1行分を印刷するための **DO** 型並び **WRITE** 文である。**IPLOT** の内容を  $I$  行について、1～56桁印刷する。

第89行：次の第90行の印刷に必要な  $x$  軸の座標 (**XJIKU**) の値を計算するための算術代入文である。

第90行： $x$  軸の座標 (**XJIKU**) の値および **IPLOT** の内容の1行分印刷するための **WRITE** 文である。

第100～110行：プログラム12のサブルーチン副プログラム **MINMAX** とまったく同じである。

## 10 実習結果および検討

プログラム1～13の概略を **Table 1** にまとめた。最後の欄の平均処理件数を除いて、学生に与えた問題に余り依存しないので例題についてのものを示す。機械工学科5年情報コースの8人がプログラム1～6の実習を終えて、更にここに報告したプログラム7～13の実習を行なった。これらの実習では、全員が与えた問題についてのプログラム全てを **TSS** で処理し、完成した。この一連の実習での平均処理件数を **Table 1** の最後の欄に示す。平均処理件数が最小であるのは、プログラム6であり、最大であるのは、プログラム11であった。1つのプログラム当りの平均処理件数は、2.4件であった。また、この一連の実習において1人当りの平均処理件数は、30.7件であった。

各プログラムの特に参考にすべきものを **Table 2** に示した。一般には、前に作成したプログラムを常に参考に、比較しながら作成すれば、理解が深まるものと思う。

プログラミング入門の実習とか演習において、最も頭を痛めることに学習者に与える問題の作成がある。学習者ごとに異なる問題を与えるのが教育的であると考えられるが、膨大な数の問題を作成しなければならないこと、指導

Table 1. The summary of practice results.

| Program number | Compile time (s) | Execute time (s) | Data lineage | No. of steps | No. of programs | Average No. of jobs |
|----------------|------------------|------------------|--------------|--------------|-----------------|---------------------|
| 1              | 20               | 4                | 1            | 14           | 1               | 2.6                 |
| 2              | 20               | 10               | 4            | 15           | 1               | 3.0                 |
| 3              | 22               | 10               | 4            | 22           | 1               | 2.6                 |
| 4              | 22               | 10               | 4            | 23           | 1               | 1.9                 |
| 5              | 24               | 10               | 4            | 27           | 2               | 2.0                 |
| 6              | 48               | 10               | 4            | 36           | 4               | 1.5                 |
| 7              | 26               | 10               | 4            | 28           | 4               | 2.8                 |
| 8              | 28               | 10               | 4            | 33           | 5               | 2.0                 |
| 9              | 22               | 7                | 4            | 23           | 1               | 1.9                 |
| 10             | 23               | 11               | 6            | 35           | 1               | 2.6                 |
| 11             | 27               | 13               | 6            | 50           | 2               | 3.4                 |
| 12             | 30               | 23               | 8            | 85           | 3               | 2.3                 |
| 13             | 39               | 29               | 5            | 111          | 5               | 2.3                 |

Table 2. Reference program.

| Program number    | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |    |
|-------------------|---|---|---|---|---|---|---|---|----|----|----|----|----|
| Reference program | 1 | 1 | 3 | 3 | 3 | 2 | 5 | 3 | 9  | 9  | 9  | 4  | 9  |
|                   |   | 2 |   | 4 |   | 6 | 7 |   |    |    | 10 | 10 | 11 |

に時間がかかり過ぎること、できない学習者が多数出ること、などの問題点がある。全ての学習者に同じ問題を与えることにすると作成する問題は少なくてよいが、他人のプログラムを丸写しする学習者が多く出て、演習とか実習とかの効果が余り期待できないなどの問題点がある。これらの点に、ステップ学習方式によるプログラミングを開発し、提案する理由がある。ここに提案したステップ学習方式によれば、これらの点がほぼ解消されるように思う。すなわち、例題のような簡単な問題を学習者の数だけ作成しておけば、入門の最後まで使用できる。問題ごとに異なる部分は決っているので、学習者の作成したプログラムの指導は、短時間に適確にできる。また、学習者に与える問題は全て異なるので、同じ問題を与えた時に生じていたプログラムの丸写しなどによる弊害を除くことができる。例題についてのプログラムを参考にすれば、与えられた問題についてのプログラムは容易に短時間に作成できる。自己が前に作成したプログラムと比較しながら作成すること、他の学習者と相談しながらプログラムを作成することなどによって、理解が深まりやすい。これらの事実は、指導した経験、学生の提出した報告書およびアンケートなどから裏付けされている。詳しい報告は次の機会に譲る。

最後に、FORTRAN プログラミング入門の教育をする際、時間に制限がある場合に、ここに提案したプログラムをどのように実施したらよいかを述べてみたい。5時間程度では、プログラム1~3の指導ができよう。10時間程度では、プログラム1~6の指導ができよう。これによって、一通り文法の知識が得られるはずである。15時間程度では、プログラム1~6、7,9,10および11が指導できると思う。20時間程度あれば、前に報告したプログラム1~6およびここ

に示したプログラム7~13の指導が可能であると思う。

## 11 結 言

ここに提案したステップ学習方式によるプログラミングが、プログラミング入門の教育法改善に貢献できることを実習結果が示している。更に、学習者が授業

を欠席した場合、復習したい場合などに便利のように個別学習機を用いて、ここに開発した方法でFORTRANプログラミングを学習させることを計画している。現在、そのための個別学習機の検討を行なっている段階である。

このステップ学習方式によるFORTRANプログラミングの特徴をまとめると、次のような点にあると考えられる。

- (1)この方式でプログラミングする学習者全員が、自らプログラムを作成し、完成できるので、学習者に相当の自信を与えることができる。
- (2)プログラミングの指導は、短時間で容易に適確に行なうことができる。
- (3)学習者に与える問題は簡単なものでよいので、容易に多数作成できる。
- (4)プログラムの作成は、比較的短時間に行なうことができる。
- (5)ステートメントの機能が比較しやすいので、理解されやすい。

終りに、ここに開発した方法にはOR的な考え方が根底に流れているが、そのORの分野で常に御指導戴いている京都大学工学部教授三根久先生、助教授大野勝久先生に深く感謝致します。また、常日頃からいろいろと御指導戴いている本校中原寿喜太先生、教育方法改善に深く御理解戴き、常に暖い御支援を戴いている本校堀悦男先生、並びに、実習の雑事をうまく処理して戴いた赤堀登美子氏の方々に深く感謝致します。

## 参 考 文 献

- 1) 木村泉訳(B. W. Kernighan & P. J. Plauger著); プログラム書法, (1976) 共立出版。
- 2) 宮地功; 津山高専紀要, No.14(1976) 65-71.
- 3) 宮地功; NEAC-Journal, No.46(1977) 85-104.
- 4) 宮地功; 津山高専紀要, No.15(1977) 35-41.
- 5) 宮地功; NEAC-Journal, No.47(1977) 89-111.