

Title	FORTTRANとの比較によるPL/I 入門 (6)
Author(s)	塩野, 充
Citation	大阪大学大型計算機センターニュース. 1981, 42, p. 95-118
Version Type	VoR
URL	https://hdl.handle.net/11094/65492
rights	
Note	

Osaka University Knowledge Archive : OUKA

<https://ir.library.osaka-u.ac.jp/>

Osaka University

FORTRANとの比較によるPL/I入門(6)

大阪大学工学部 塩 野 充

省エネルギーのあおりで、以前なら6月中頃から入っていた研究室の冷房が、今年は7月に入っても仲々入れて頂けない有様で、扇風機の生ぬるい風にあたりつつ、この原稿を書くハメになった。しかし計算機センターだけはコンピュータ様を冷やさなければならない関係上、大変涼しい。特に磁気テープ装置のある部屋など、長時間いると寒けがしてくる程である。だから今の時期はセンターへ夕涼みならぬ昼涼みに出かける人もいるのではないかと思う。筆者もその一人かも知れない。

1月から3月初めにかけてのセンターの大混雑期、いわゆるメチャ混みの時期が過ぎて4月から夏休み中にかけては閑散期、すなわちセンターに閑古鳥の鳴く季節である。新たに導入されたシステムⅡも6月末からは省エネルギーのために、ジョブの少ないときは運転休止となっている。もちろん、ジョブが多くなってきたときは運転しているが、この切り換えが今のところあまりスマートではないように筆者には思える。システムⅠでAジョブやBジョブが十数件WAITINGになっているのにシステムⅡが休止している日があると思えば、その逆にシステムⅠにWAITINGが殆んどないのにシステムⅡも運転している日がある。これは多分、人間がジョブの混雑具合をディスプレイで見てシステムⅡの電源スイッチON-OFFの判断をしているからではないかと思う。(もし違っていたら失礼)。それよりも筆者が思うに、システムⅠにシステムⅡの電源スイッチON-OFFをまかせたらどうだろうか。ジョブが多くなってきてシステムⅠの負荷がある程度以上大きくなると、システムⅠ自身がシステムⅡの電源スイッチをONにする仕組みにしておく。又、逆に負荷が少なくなるとシステムⅠがシステムⅡの電源スイッチをOFFにするわけである。勿論、電源OFFにするといっても、カード読取中やジョブ実行中、印刷出力中にいきなり全部が動かなくなるとは困るので、システムⅠに制御される音声出力装置によって、「只今よりシステムⅡへの入力を中止致します。」とかなんとかのアナウンス(勿論、ウグイス嬢の美声が必要で、暑くるしい男声ではこれから入力しようとしていた利用者がよけいにカッとくるであろう)をさせる。電源スイッチをONにするときも勿論自動アナウンスを行う。このようなソフトやハードは簡単に作る事ができるのではないかと思う。

しかし乍ら、このようなことはそもそもシステムⅠとシステムⅡが有機的に結合していないから必要になるのであって、システムⅠとシステムⅡとが有機的に結合していて、完全なるデュプレックスあるいはデュアルシステムを構成していれば、少なくともユーザーから見てシステムⅠとシステムⅡの区別はなくなるはずである。カードリーダーやラインプリンタがシステムⅠ用とシステムⅡ用に別れているのもおかしなことなのである。どのカードリーダーから入力してもユーザーに

は無関係に、負荷に応じてジョブがシステムⅠとシステムⅡに振り分けられるのでなければなるまい。どちらのシステムで処理されたかをユーザーは意識しなくてよいのである。こうなっておればシステムⅡの電源がONになっていようがOFFになっていようが、ユーザーには全く気がつかれないし、ユーザーにシステムⅠとシステムⅡのカードリーダーの間を右往左往させる迷惑をかけることもない。世界に名だたるNECの通信技術をもってすればこのようなことは朝飯前だろうと思うのだが……。それとも、もうしばらくすれば最大のACOSシステム1000が入ってどっちみち1台のシステムになるのだから、と行ってサボっているのかな(……?)。とにかくにも、NEC技術陣の奮起を切に願うものである。今のままではあまりにもブサイクで、かつ田舎くさい。

さて、余談はこれくらいにして、拙稿(というのは半分謙遜で本音は玉稿とでも書きたい……これは冗談)もいよいよ今回が最終回となった。前回までにPL/Iの文法に関する事柄は大体全部説明が終った。但し、あくまで基本的なことばかりで、むずかしいことは書いていないし、書いても読んでもらえないだろうし、第一、書く力がないのである。書く力といっても筆圧のことではない(!)。筆圧は強すぎると鉛筆が折れてしょうがないし、万年筆はオシャカになる。というような関係のない話はさておき、今回の内容としては、実際にACOSでPL/Iのプログラムを使用する際のJCL(ジョブ制御言語)の説明や、PL/IプログラムとFORTRANプログラムの結合の方法と実際例についての説明を行う。

第18章 PL/IプログラムのためのACOS-6 JCL(ジョブ制御言語)

ACOSシステムによってPL/Iプログラムを使用するときのデッキ構成を以下に示す。なお、PL/Iは最近TSSでも使えるようになったが、後述するFORTRANとの結合ができるのはバッチジョブの場合だけなので、バッチの場合について解説することにする。まず、最も一般的なデッキ構成を以下に示す。

	1カラム	8カラム	16カラム
(A)	\$	SNUMB	
(B)	\$	JOB	……………
(C)	\$	BREAK	
(D)	\$	OPTION	PL1
(E)	\$	PL1	LSTIN

} PL/Iプログラム

```

(F)  $      GO
(G)  $      LIMITS .....
      }
      } データカード
      }
(H)  $      ENDJOB
(I)  ***EOF

```

まず、(A)の\$SNUMB文と(B)の\$JOB文はFORTRANの場合と全く同様である。その次の(C)の\$BREAK文は何のために入れるのかを説明しよう。まず、(B)の\$JOB文というのは後述する(F)の\$GO文と同様に、ACOS-6のJCLではないのである。ウンだと思ふ人はACOSマニュアルのジョブ制御言語説明書をとくと読んで\$JOB文と\$GO文を捜してみればよい。このようなJCLは載っていない。\$JOB文と\$GO文はこのセンターでだけ用いられているJCLで、JCLの数をなるべく減らして初心者にも使い易いようにするために、いくつかのJCLを1まとめにした、いわゆるカタログド・プロシージャである。そしてこれはあくまでFORTRANユーザーを対象としている。FORTRANジョブのLP出力リストの第1ページを見ると、JCLが印刷されているが、その中に\$JOBや\$GOというのはない。その代わりに#印の付いたいくつかのJCLが印刷されているであろう。その#印の付いたJCLが、カタログド・プロシージャ\$JOB、\$GOを構成しているJCLである。すなわち、

```

$ JOB = {
          $ IDENT      6092.....
          $ JOBDEF      .....
          $ USERID     6092.....
          $ LIMITS     1, , , 1500
          $ LOWLOAD
          $ OPTION     FORTRAN, RELMEM

```

```

$ GO = {
          $ OPTION     NOMAP
          $ EXECUTE
          $ SELECT     .....
          $ PRMFL      MH, .....
          $ PRMFL      CH, .....
          $ PRMFL      DH, .....

```

となる。又、\$のあとにAの付いているのがアクティビティである。\$FORTRAN(コンパイラのアクティビティ)や、\$EXECUTE(実行のアクティビティ)である。

ところで、FORTRANのジョブを行うときはこのままでよいのだが、PL/Iのジョブを行うときは、\$JOB文を構成するJCLの最後の、\$OPTION FORTRAN, ……がジャマになる。これを打ち消すのが\$BREAK文である。\$BREAK文によって、\$OPTION FORTRAN, ……を打ち消しておいて、次の(D)の\$OPTION PL1で改めて定義し直すわけである。ここで、

```
$          OPTION  PL1
```

とすれば、TYPE(AB)のコンパイラ(ACOS-6 PL/I)を表わし、

```
$          OPTION  PLONE
```

とすれば、TYPE(A)のコンパイラ(ACOS-6 標準PL/I)を表わす。\$BREAK文を入れなくてもプログラムは一応動作するが、F0フォルトが発生するときがある。

次の(E)は言語指定である。すなわち、\$FORTRAN文に相当する。LSTINのところはコンパイラオプションであり、\$FORTRAN文と同様にいろいろ指定できる。LSTINを省略すれば、プログラムはノーリストとなる。オプション指定はOPTZとする。これも\$FORTRAN文と同様である。他にPL/I独特のものとして大事なものに次のものがある。

(i)TYPE(A), TYPE(AB)……………TYPE(A)とすればACOS-6 標準PL/I, TYPE(AB)とすればACOS-6 PL/Iコンパイラとなる。省略すれば、TYPE(AB)と見なされる。

(ii)SNUMBER……………このオプションを入れておけば、実行中にエラーが発生したとき、プログラムの発生箇所の行ナンバーを印刷する。虫取りの段階では入れておく方がよい。

(iii)MULTI……………PL/Iで書かれたサブルーチン(外部手続き)があるときは必ず入れる。入れなければメインプログラム(メインブロック)とサブルーチン(サブブロック)の区切りがなく連続したものと解釈され、エラーだらけとなる。

この他にも沢山のオプションがあるが、初心者にとって重要なのはこれくらいであろう。(D)と(E)の例を示す。TYPE(AB)ならば、

```
$          OPTION  PL1
```

```
$          PL1          LSTIN, SNUMBER, MULTI
```

となり、TYPE(A)ならば、

```
$          OPTION  PLONE
```

```
$          PL1          LSTIN, SNUMBER, MULTI, TYPE(A)
```

となる。\$PL1文の前には必ず\$OPTION文を付けなければプログラムは動かない。

(F)の\$GO文以降はFORTRANの場合と全く同じなので説明は省略する。磁気テープやPRMFLなどの各種入出力ファイルを定義する場合もFORTRANの場合と同様にすればよい。ただ、第15章で述べたようにファイルコードがFORTRANのときのような2ケタの数字ではなくて、2文字の英数字になるので注意が必要である。

第19章 PL/IプログラムとFORTRANプログラムの結合

PL/I言語がFORTRAN言語よりはるかに機能が大きくて便利なのにもかかわらず、あまり普及しない原因の1つに次のようなことが考えられる。つまり、今までずっとFORTRANでプログラムを作ってきたのに、今さらPL/Iでプログラムを作ると、今までのFORTRANプログラムの蓄積が全部パーになってしまう。これまでコツコツと苦勞して作ってきた沢山のFORTRANサブルーチン群を捨てるわけにはいかない。だから仲々PL/Iを使う気にはなれない……、と考える人が多いのではないかと思う。ところが心配御無用、PL/IプログラムでもFORTRANサブルーチンが使えるのです。又、自分で作ったFORTRANサブルーチンだけではなく、メーカーのサブルーチンライブラリMATHLIB-6や、センターのサブルーチンライブラリもPL/Iで使えるのである。この章ではPL/IプログラムとFORTRANプログラムの結合方法についてお話しすることにしよう。

〈19-1〉結合する際の制約条件

PL/IプログラムとFORTRANプログラムを結合する場合、ACOSシステムでは次のような制約条件がある。

- (i) 必ずPL/IプログラムがCALLする方で、FORTRANプログラムがCALLされる方(サブルーチン)でなければならない。FORTRANプログラムからPL/IプログラムをCALLすることはできない。
- (ii) 入出力文はPL/IプログラムかFORTRANプログラムのどちらか一方にしか使えない。両方に入出力文があるとFRCアボートが起こる。メインプログラムで入出力文を使わないわけにはいかないだろうから通常は入出力文をPL/Iプログラムの方にまとめて、FORTRANサブルーチンでは入出力文を使わないようにすればよい。幸い、MATHLIB-6やセンターライブラリでは入出力文は殆んど使われていないようである。全部について確かめたわけではないので全く使われていないとは言いきれないが、殆んどどのサブルーチンにIERRというエラーインディケータが引数として入っており、このIERRの値によってサブルーチンがうまく動作したかどうか分かるようになっているので、サブルーチンから直接メッセージを印刷したりすることは必要ないからであろう。

以上の2つの制約条件はPL/I言語に固有のものではなく、あくまでACOSシステムの場合

だけの話であり、いずれは解除されるべき性質のものであるし、解除されるであろう。

<19-2>FORTRANプログラムをCALLするPL/Iプログラムの形式

FORTRANプログラムをCALLするPL/Iプログラムは次のような形となる。呼ばれるFORTRANサブルーチンの名前を例えば、FSUB1としよう。プログラムは次のようになる。

(PL/Iプログラム)

```
EX1:PROC OPTIONS(MAIN);
      DCL A(10,20) FLOAT BIN(27);
      DCL X FLOAT BIN(27);
      DCL K FIXED BIN(35);
      DCL FSUB1 ENTRY((10,20)FLOAT BIN(27),
                     FLOAT BIN(27),FIXED BIN(35))
                     OPTIONS(FORTRAN);
      .....
      CALL FSUB1(A,X,K);
      .....
END EX1;
```

(FORTRANプログラム)

```
SUBROUTINE FSUB1(F,Y,L)
REAL F(10,20),Y
INTEGER L
.....
RETURN
END
```

第7章でも述べたようにFORTRANと結合する際は、PL/Iの変数の型宣言は精度まできっちり宣言しなくてはならない。すなわち、次の表19-1のようになる(但し、これはACOS-6の場合であり、対応は処理系によって異なる)。この例では、A(10,20)はF(10,20)に、XはYに、KはLに対応づけられる。FSUB1ではYやLは実際は宣言する必要はないが、対応を明確にするために記してある。ここで重要なのは4番目のDCL文である。ENTRYというのは、FSUB1がサブルーチン名であることを示している。ENTRYの後のカッコの中には、そのサブルーチンの引数の型を順番通りに列挙しておく。配列の場合は例に示すように型

表19-1 等価なデータの型

型	FORTRAN	PL/I
整数	INTEGER	FIXED BIN(35)
実数	REAL	FLOAT BIN(27)
複素数	COMPLEX	COMPLEX FLOAT BIN(27)

の前に配列の大きさを書く。そしてカッコを閉じた後に、OPTIONS(FORTRAN);と書けばよいのである。こう書けばそのサブルーチンがFORTRANで書かれていることを意味する。このDCL文の一般的な形式は次のようになる。

```
DCL サブルーチン名 ENTRY(第1引数の型、第2引数の型、第3引数の型、……
                        ……………) OPTIONS(FORTRAN);
```

ところで、FORTRANのサブルーチンにはよく整合配列が使われる。この場合には次のようにする。同じ例で、F(10, 20)が整合配列F(I, J)になったときを考える。

(PL/Iプログラム)

```
EX2:PROC OPTIONS(MAIN);
      DCL A(10, 20) FLOAT BIN(27);
      DCL X FLOAT BIN(27);
      DCL (I, J, K) FIXED BIN(35);
      DCL FSUB2 ENTRY((*, *)FLOAT BIN(27),
                      FIXED BIN(35), FIXED BIN(35),
                      FLOAT BIN(27), FIXED BIN(35))
                      OPTIONS(FORTRAN);
```

```
.....
      I=10;
      J=20;
      CALL FSUB2(A, I, J, X, K);
      .....
```

```
END EX2;
```

(FORTRANプログラム)

```
SUBROUTINE FSUB2(F, I, J, Y, L)
      REAL F(I, J), Y
      INTEGER L
```


.....
RETURN

END

すなわち、整合配列の大きさは第 10 章で述べたように星印 (*) で表わせばよい。

なお、呼ぶ方の PL / I プログラムはいつも PROC OPTIONS (MAIN) ; である必要はなく、サブルーチンの PROC であってもよい。

< 19-3 > JCL の構成

PL / I プログラムと FORTRAN プログラムを結合する際の JCL は次のようになる。

```
1カラム      8カラム      16カラム
$           SNUMB
$           JOB           .....
$           BREAK
$           OPTION      PL1
$           USE         .PSETU, .FSETU
$           PL1         LSTIN, .....

        }
        } PL / I プログラム

$       FORTRAN BIN, LSTIN, .....

        }
        } FORTRAN プログラム

$       GO           BIN
$       LIMITS      .....

        }
        } データカード

$       ENDJOB

***EOF
```

前述したPL/Iプログラムだけの場合と大体似ているが、異なるのは\$OPTION文の次の\$USE文である。この\$USE文によって、.PSETU(PL/Iセットアップルーチン)と.FSETU(FORTRANセットアップルーチン)を呼び出している。又、\$FORTRANと\$GO文のオプションにBIN(2進モード)が入っている。ここは通常のFORTRANは省略されてHEX(16進モード)となっているところだが、PL/Iと結合する場合はBINの指定が必要である。

実際例を1つ示そう。問題は誰でもよく御存知の2次方程式の根の公式である。これを近頃の高校の教科書では根の公式と書かずに、解の公式としている。筆者のように「根の公式」の方がしみついた世代には「解の公式」というのはなんとなく言いにくく、ゴロも悪いように見えるのだが、こんなことを言っていると高校生諸君からオジンよばわりされそうである。近頃は30どころか、20歳を過ぎるとオジンの定義に入ってしまうそうだから……。余談はさておき、まず最初にFORTRANだけでプログラムを作った。図19-1にメインプログラム、図19-2にサブルーチンKON、図19-3に実行結果を示す。各々の根(解)の前半は実部、後半は虚部である。なお、入力データはカードに、

```
1 カラム
1, 1, 1
3, 2, 1
2, 4, 2
3, 5, 2
11, 28, 17
-4, 2, -3
-5, -8, 0
61, -55, 29
0, 0, 0
```

と打ってある。同じFORTRANサブルーチンKONを今度はPL/IプログラムでCALLした例を示そう。図19-4はJCLリスト、図19-5はメインプログラム、図19-6は実行結果である。入力データは全く同じである。GET文にCOPYオプションを付けているので入力データがそのまま印字されている。出力は同じE変換(E書式)でもFORTRANでは有効数字が小数第1位から始まっているのに対し、PL/Iでは有効数字が1の位から始まっているのが分る。

<19-4>ライブラリとの結合

自分で作ったFORTRANサブルーチンだけではなく、メーカーライブラリMATHLIB-

6やセンターライブラリをCALLすることもできる。この場合のJCL構成は次のようになる。

```
1カラム      8カラム      16カラム
$           SNUMB
$           JOB           .....
$           BREAK
$           LIBRARY MB, CB
$           OPTION  PL1
$           USE           .PSETU, .FSETU
$           PL1          LSTIN, .....

          }
          } PL/Iプログラム
          }

$           GO           BIN
$           LIMITS      .....

          }
          } データカード

$           ENDJOB

***EOF
```

すなわち、\$LIBRARY文が必要となる。この場合、BINモードゆえライブラリのファイルコードはMH、CHではなくてMB、CBとなる。以下に4つの例を示そう。なお、いずれもメーカーライブラリMATHLIB-6を使った例である。センターライブラリを使う場合も全く同様である。

```

1      C      *** 2-JI HOUTEISHIKI NO KON ***
2      INTEGER A,B,C,D
3      COMPLEX X1,X2
4      WRITE(6,500)
5      500  FORMAT(1H1)
6      1  READ(5,*) A,B,C
7      IF(A.EQ.0) GO TO 2
8      WRITE(6,100) A,B,C
9      100  FORMAT(// '      HOUTEISHIKI : ',I4,'*X**2+',I4,'*X+',I4,' = 0 ***'
10     & )
11     CALL KON(A,B,C,D,X1,X2)
12     WRITE(6,200) D
13     200  FORMAT(1H, '      HANBETSU SHIKI D=',I8)
14     WRITE(6,300) X1,X2
15     300  FORMAT(1H, '      KOTAE : X1=',2E13.3,' , X2=',2E13.3//)
16     GO TO 1
17     2  WRITE(6,400)
18     400  FORMAT(// '      *** KEISAN OWARI (FORTRAN) ***')
19     STOP
20     END

```

☒ 19-1

```

1      C      *** KON (KON NO KOU SHIKI) ***
2      SUBROUTINE KON(A,B,C,D,X1,X2)
3      INTEGER A,B,C,D
4      COMPLEX X1,X2,ROOTD
5      IF(A.EQ.0) RETURN
6      D=B*B-4*A*C
7      ROOTD=CSQRT(D)
8      X1=(-B+ROOTD)/(2*A)
9      X2=(-B-ROOTD)/(2*A)
10     RETURN
11     END

```

☒ 19-2

HOUTEISHIKI : $1 \times X^{**2} + 1 \times X + 1 = 0$ ***
 HANBETSU SHIKI D= -3
 KOTAE : X1= -0.500E+00 0.866E+00 / X2= -0.500E+00 -0.866E+00

HOUTEISHIKI : $3 \times X^{**2} + 2 \times X + 1 = 0$ ***
 HANBETSU SHIKI D= -8
 KOTAE : X1= -0.333E+00 0.471E+00 / X2= -0.333E+00 -0.471E+00

HOUTEISHIKI : $2 \times X^{**2} + 4 \times X + 2 = 0$ ***
 HANBETSU SHIKI D= 0
 KOTAE : X1= -0.100E+01 0. / X2= -0.100E+01 0.

HOUTEISHIKI : $3 \times X^{**2} + 5 \times X + 2 = 0$ ***
 HANBETSU SHIKI D= 1
 KOTAE : X1= -0.667E+00 0. / X2= -0.100E+01 0.

HOUTEISHIKI : $11 \times X^{**2} + 28 \times X + 17 = 0$ ***
 HANBETSU SHIKI D= 36
 KOTAE : X1= -0.100E+01 0. / X2= -0.155E+01 0.

HOUTEISHIKI : $-4 \times X^{**2} + 2 \times X - 3 = 0$ ***
 HANBETSU SHIKI D= -44
 KOTAE : X1= 0.250E+00 -0.829E+00 / X2= 0.250E+00 0.829E+00

HOUTEISHIKI : $-5 \times X^{**2} + -8 \times X + 0 = 0$ ***
 HANBETSU SHIKI D= 64
 KOTAE : X1= -0.160E+01 0. / X2= 0. 0.

HOUTEISHIKI : $61 \times X^{**2} + -55 \times X + 29 = 0$ ***
 HANBETSU SHIKI D= -4051
 KOTAE : X1= 0.451E+00 0.522E+00 / X2= 0.451E+00 -0.522E+00

*** KEISAN OWARI (FORTRAN) ***

☒ 19-3

```

0001 ¥ SNUMB M770C
0002 ¥* CPROC A/E, ██████████, N, E, C/1, HOLD
0003# ¥ IDENT ██████████, N, E 0630811639
0004# ¥ JOBDEF DEST=, CLASS=C/1, OPTION=HOLD
0005# ¥* USERID ██████████
0006# ¥ LIMITS 1, 1500
0007# ¥ LOWLOAD
0008# ¥ OPTION FORTRAN, RELMEM
0009 ¥ OPTION PL1
0010 ¥ USE PSETU, FSETU
0011 A¥ PL1 LSTIN
0012 A¥ FORTRAN LSTIN, BIN
0013 ¥* CPROC CB/HGAAA, E
0014# ¥ OPTION NOMAP
0015# A¥ EXECUTE
0016# ¥¥ SELECT OPNSUTIL/G0/E
0017* ¥ LIMITS 1, 16K, -4K, 1500
0018# ¥¥ PRMFL MB, R, R, LIB/MLIBB
0019# ¥¥ PRMFL CB, R, R, LIB/CLIBB
0020# ¥¥ PRMFL DB, R, R, LIB/DLIBB
0021# ¥¥ PRMFL GB, R, R, LIB/GLIBB
0022# ¥¥ PRMFL XB, R, R, LIB/XLIBB
0023 ¥ LIMITS 1, 100K, -2K, 1500
0024 ¥ ENDJOB
TOTAL CARD COUNT THIS JOB = 000064

```

```

* ACTY-01 ¥CARD #0011 PL1 06/30/81 REAL MODE SW=210200000000
* NORMAL TERMINATION AT 023554 BA=000000200000 I=4020 SW=210200000000

* ACTY-02 ¥CARD #0012 FORTY 06/30/81 REAL MODE SW=210210000000
* NORMAL TERMINATION AT 005440 BA=000000200000 I=4060 SW=210210000000

* ACTY-03 ¥CARD #0015 GELOAD 06/30/81 REAL MODE SW=000000000000
* NORMAL TERMINATION AT 040336 BA=000000200000 I=5000 SW=000000000000

```

図 19-4

```

LINE +-----+-----+-----+-----+-----+-----+-----+-----+
1+ /* 2-JI HOUTEISHIKI NO KON */
2  EQA:PROC OPTIONS(MAIN);
3  DCL (A,B,C,D) FIXED BIN(35);
4  DCL (X1,X2) COMPLEX FLOAT BIN(27);
5  DCL KON ENTRY(FIXED BIN(35),FIXED BIN(35),FIXED BIN(35),FIXED BIN(35),
6+ COMPLEX FLOAT BIN(27),COMPLEX FLOAT BIN(27)) OPTIONS(FORTRAN);
7  PUT PAGE;
8  L1: GET LIST(A,B,C) COPY;
9  IF A=0 THEN GO TO L2;
10 PUT SKIP(3);
11 PUT EDIT(' *** HOUTEISHIKI : ',A,'*X**2+',B,'*X+',C,
12+ ' = 0 ***' )(A,F(4),A,F(4),A,F(4),A);
13 CALL KON(A,B,C,D,X1,X2);
14 PUT SKIP EDIT(' HANBETSU SHIKI D=',D)(A,F(8));
15 PUT SKIP EDIT(' KOTAE : X1=',X1,' , X2=',X2)(A,C(E(13,3)),
16+ A,C(E(13,3)));
17 PUT SKIP(3);
18 GO TO L1;
19 L2:PUT SKIP EDIT(' *** KEISAN OWARI (PL/I) ***')(A);
20 END EQA;

```

図 19-5

1,1,1

*** HOUTEISHIKI : 1*X**2+ 1*X+ 1 = 0 ***
HANBETSU SHIKI D= -3
KOTAE : X1= -5.000E-001 8.660E-001 / X2= -5.000E-001 -8.660E-001

3,2,1

*** HOUTEISHIKI : 3*X**2+ 2*X+ 1 = 0 ***
HANBETSU SHIKI D= -8
KOTAE : X1= -3.333E-001 4.714E-001 / X2= -3.333E-001 -4.714E-001

2,4,2

*** HOUTEISHIKI : 2*X**2+ 4*X+ 2 = 0 ***
HANBETSU SHIKI D= 0
KOTAE : X1= -1.000E+000 0.000E+000 / X2= -1.000E+000 0.000E+000

3,5,2

*** HOUTEISHIKI : 3*X**2+ 5*X+ 2 = 0 ***
HANBETSU SHIKI D= 1
KOTAE : X1= -6.667E-001 0.000E+000 / X2= -1.000E+000 0.000E+000

11,28,17

*** HOUTEISHIKI : 11*X**2+ 28*X+ 17 = 0 ***
HANBETSU SHIKI D= 36
KOTAE : X1= -1.000E+000 0.000E+000 / X2= -1.545E+000 0.000E+000

-4,2,-3

*** HOUTEISHIKI : -4*X**2+ 2*X+ -3 = 0 ***
HANBETSU SHIKI D= -44
KOTAE : X1= 2.500E-001 -8.292E-001 / X2= 2.500E-001 8.292E-001

-5,-8,0

*** HOUTEISHIKI : -5*X**2+ -3*X+ 0 = 0 ***
HANBETSU SHIKI D= 64
KOTAE : X1= -1.600E+000 0.000E+000 / X2= 0.000E+000 0.000E+000

61,-55,29

*** HOUTEISHIKI : 61*X**2+ -55*X+ 29 = 0 ***
HANBETSU SHIKI D= -4051
KOTAE : X1= 4.508E-001 5.217E-001 X2= 4.508E-001 -5.217E-001

0,0,0

*** KEISAN OWARI (PL/I) ***

図 19-6

(例1) 一様乱数の発生 (S9UNI2)

乗算合同法による一様乱数の発生である。このS9UNI2がS9UNI1と異なる点は、乱数の計算初期値(いわゆる乱数のタネ)が指定できることである。ここではタネはサブルーチンにまかせてある。(IND=0とする)。図19-7にFORTRANのメインプログラム、図19-8にその実行結果、図19-9にPL/Iで行う場合のJCLリスト、図19-10にPL/Iメインプログラム、図19-11にその実行結果を示す。図19-8と全く等しいことが分る。(改行数が違っているのは筆者のちょっとしたミスである)。JCLリストの\$LIBRARY文ではMATHLIB-6を使う場合はMBだけでよく、CBは実際は不要である。

```
1      C      *** TEST OF S9UNI2 (ICHIYOU RANSUU) ***
2          REAL RR(100)
3          IND=0
4          WRITE(6,100)
5          100  FORMAT(1H1///6X,'*** S9UNI2 NI YORU RANSUU HASSEI (CALLED BY FORTR
6              &AN)')//)
7          DO 1 I=1,100
8              CALL S9UNI2(IND,IR,R)
9              1  RR(I)=R
10         WRITE(6,200) RR
11         200  FORMAT(/6X,10F10.5)
12         STOP
13         END
```

図19-7

*** S9UNI2 NI YORU RANSUU HASSEI (CALLED BY FORTRAN)

0.15036	0.46722	0.71502	0.24261	0.82001	0.78171	0.18305	0.22280	0.30162	0.23202
0.33619	0.60803	0.04258	0.00945	0.11064	0.40024	0.68021	0.78872	0.48973	0.11993
0.45633	0.13301	0.76116	0.42210	0.87244	0.03221	0.59742	0.54973	0.86240	0.72807
0.53631	0.44622	0.80698	0.49662	0.38733	0.43273	0.36820	0.13829	0.49600	0.18339
0.33413	0.93037	0.09210	0.30156	0.21248	0.90668	0.29749	0.37313	0.25974	0.45364
0.24862	0.79637	0.00488	0.60396	0.70409	0.64719	0.92448	0.15546	0.14633	0.14244
0.86424	0.33497	0.20523	0.52204	0.75266	0.62357	0.15502	0.00097	0.32049	0.44080
0.02461	0.09559	0.45054	0.22793	0.99057	0.89910	0.80369	0.41393	0.18461	0.73572
0.05258	0.29936	0.48924	0.95941	0.64555	0.38528	0.75820	0.44926	0.80597	0.16392
0.92862	0.51709	0.12227	0.22664	0.56524	0.96405	0.17362	0.12034	0.59333	0.20456

図19-8


```

0001 # SNJMB 9916B
0002 ¥* CPROC A/F, ,N,E,C/1,HOLD
0003# ¥ IDENT ,N,E 0709811359
0004# ¥ JOBDEF UEST=,CLASS=C/1,OPTION=HOLD
0005# ¥¥ USERID
0006# ¥ LIMITS 1,,1500
0007# ¥ LOWLOAD
0008# ¥ OPTION FORTRAN,RELMEM
0009 ¥ BREAK
0010 ¥ LIBRARY AB,CB
0011 ¥ OPTION PL1
0012 ¥ USE .PSETU,.FSETU
0013 A ¥ PL1 LISTIN,NUMBER
0014 ¥* CPROC CB/HGAAA,,E
0015# ¥ OPTION NOMAP
0016#A ¥ EXECUTE
0017# ¥¥ SELECT OPNSUTIL/GD/E
0018* ¥ LIMITS 1,16K,-4K,1500
0019# ¥¥ PRMFL AB,R,R,LIB/MLIB
0020# ¥¥ PRMFL CB,R,R,LIB/CLIB
0021# ¥¥ PRMFL DB,R,R,LIB/DLIB
0022# ¥¥ PRMFL GB,R,R,LIB/GLIB
0023# ¥¥ PRMFL XB,R,R,LIB/XLIB
0024 ¥ LIMITS 1,100K,-2K,1500
0025 ¥ ENDJOB
TOTAL CARD COUNT THIS JOB = 000641

```

```

* ACTY-01 ¥CARD #0013 PL1 07/09/81 REAL MODE SW=210200000000
* NORMAL TERMINATION AT 023554 BA=000000200000 I=4020 SW=210200000000
* ACTY-02 ¥CARD #0016 GELOAD 07/09/81 REAL MODE SW=000000000000
* NORMAL TERMINATION AT 035336 BA=000000200000 I=5000 SW=000000000000

```

☒ 19-9

```

LINE +-----+-----+-----+-----+-----+-----+-----+-----+
1+ /* TEST OF S9UN12 (ICHIYOU RANSUU) */
2 S9:PROC OPTIONS(MAIN);
3 DCL (RR(100),R) FLOAT BIN(27);
4 DCL (I,II,IND,IR) FIXED BIN(35);
5 DCL S9UN12 ENTRY(FIXED BIN(35),FIXED BIN(35),FLOAT BIN(27))
6+ OPTIONS(FORTRAN);
7 IND=0;
8 PUT PAGE EDIT('*** S9UN12 NI YORU RANSUU HASSEI (CALLED BY PL/1)',
9 ' ***')(SKIP(3),X(5),A,A); PUT SKIP(2);
10 DO I=1 TO 100;
11 1 II=I;
12 1 CALL S9UN12(IND,IR,R);
13 1 RR(II)=R;
14 1 END;
15 PUT EDIT(RR)(SKIP, X(5),(10)F(10,5));
16 END S9;

```

☒ 19-10

*** S9UN12 NI YORU RANSUU HASSEI (CALLED BY PL/1) ***

0.15036	0.46722	0.71502	0.24261	0.82001	0.76171	0.14305	0.22280	0.30162	0.23202
0.33619	0.60813	0.94258	0.00945	0.11064	0.40024	0.68021	0.78872	0.48973	0.11993
0.45633	0.13301	0.76116	0.42210	0.87244	0.03221	0.59742	0.54973	0.86240	0.72807
0.53631	0.44622	0.80994	0.49662	0.58733	0.43273	0.36820	0.13829	0.49600	0.18339
0.33413	0.92037	0.07210	0.30156	0.21248	0.90668	0.29749	0.87313	0.25974	0.45364
0.24862	0.79637	0.00498	0.60396	0.70409	0.64719	0.92448	0.15546	0.14633	0.14244
0.86424	0.33497	0.27523	0.52204	0.75264	0.62357	0.15502	0.00097	0.32049	0.44080
0.02461	0.09559	0.45054	0.22793	0.99057	0.89910	0.80369	0.41393	0.18461	0.73572
0.05258	0.29936	0.44924	0.95941	0.64555	0.38528	0.75820	0.44926	0.80597	0.16392
0.92862	0.51709	0.12227	0.27634	0.56524	0.96405	0.17362	0.12034	0.59333	0.20456

☒ 19-11

(例2) 高次方程式の解 (SDBAIR)

ベアストウ法による高次方程式の数値解法である。ここでは5次方程式に適用している。係数を入力データとしてカード上に、次のような3種類の5次方程式を与える。

```
1 カラム
2, 3, 4, 5, 6, 7
2 2, 4, 3 0, 2 9, 1, 2 6
-8, -3, 1, 1, 3, 9
0, 0, 0, 0, 0, 0
```

FORTRANのメインプログラムを図19-12に、その実行結果を図19-13に示す。PL/Iのメインプログラムを図19-14に、その実行結果を図19-15に示す。GET文にCOPYオプションがあるので入力データをそのまま印字している。解の前半は実部、後半は虚部である。

(例3) 数値積分 (SKTRPZ)

台形公式による定積分の数値解法である。ここで少し注意が必要なことがある。このサブルーチンSKTRPZは引数の1つが関数名となっている。すなわち被積分関数を与えるためである。これをFORTRANで使う場合はEXTERNAL宣言をして用いる。ところがPL/Iではこのようなことはできないので、関数名を直接SKTRPZに渡すことはできない。そこで仲介役のFORTRANサブルーチンFLIB1を作り、PL/IメインプログラムからはこのFLIB1をCALLして、関数名以外の引数を渡す。FLIB1ではEXTERNAL宣言して、関数名をつけ加えてからSKTRPZをCALLする。関数は勿論FORTRANで作っておかなければならない。このように、関数名を引数としているサブルーチンをPL/IからCALLする場合は仲介役のサブルーチン(ここではFLIB1)を作ってやらなければならないので少し面倒ではある。図19-16にFORTRANのメインプログラム、図19-17にFORTRANのFUNCTION、図19-18にその実行結果を示す。図19-19にPL/Iのメインプログラム、図19-20にFLIB1、図19-21にその実行結果を示す。被積分関数は手で計算すると、

$$\begin{aligned} & \int_1^2 (15x^4 + 8x^3 - 9x^2 + 4x + 1) dx \\ &= \left[\frac{15}{5}x^5 + \frac{8}{4}x^4 - \frac{9}{3}x^3 + \frac{4}{2}x^2 + x \right]_1^2 \\ &= \left[3x^5 + 2x^4 - 3x^3 + 2x^2 + x \right]_1^2 \\ &= 3 \times 32 + 2 \times 16 - 3 \times 8 + 2 \times 4 + 2 - (3 + 2 - 3 + 2 + 1) \\ &= 96 + 32 - 24 + 8 + 2 - 5 \\ &= 109 \end{aligned}$$

となる。

```

1      C      *** TEST OF SDBAIR (KOUJI HOUTEISHIKI NO KAI) ***
2      REAL A(6),XR(5),XI(5),W1(5),W2(5)
3      N=5
4      L=100
5      LI=7
6      WRITE(6,90)
7      90  FORMAT(1H1)
8      1  READ(5,*) A
9      IF(A(1).EQ.0.) STOP
10     WRITE(6,100) A
11     100 FORMAT( //6X, '*** 5-JI HOUTEISHIKI : ',F5.2,'*X**5+',F5.2,
12     &'*X**4+',F5.2,'*X**3+',F5.2,'*X**2+',F5.2,'*X+',F5.2,' =0 ***')
13     CALL SDBAIR(N,A,L,LI,NR,XR,XI,W1,W2,IERR)
14     WRITE(6,200) IERR
15     200 FORMAT(//6X,'*** IERR=',I2, ' (CALLED BY FORTRAN)')
16     WRITE(6,300) ((XR(I),XI(I)),I=1,5)
17     300 FORMAT(/6X,'X=',2E13.3)
18     GO TO 1
19     END
EQUALITY OR NON-EQUALITY COMPARISON MAY NOT BE MEANINGFUL IN LOGICAL IF EXPRESSIONS

```

☒ 19-12

*** 5-JI HOUTEISHIKI : 2.00*X**5+ 3.00*X**4+ 4.00*X**3+ 5.00*X**2+ 6.00*X+ 7.00 =0 ***

*** IERR= 0 (CALLED BY FORTRAN)

X= -0.130E+01 0.
X= -0.673E+00 0.111E+01
X= -0.673E+00 -0.111E+01
X= 0.573E+00 0.113E+01
X= 0.573E+00 -0.113E+01

*** 5-JI HOUTEISHIKI : 22.00*X**5+ 4.00*X**4+30.00*X**3+29.00*X**2+ 1.00*X+26.00 =0 ***

*** IERR= 0 (CALLED BY FORTRAN)

X= -0.104E+01 0.
X= 0.361E+00 0.872E+00
X= 0.361E+00 -0.872E+00
X= 0.693E-01 0.113E+01
X= 0.693E-01 -0.113E+01

*** 5-JI HOUTEISHIKI : -8.00*X**5+-3.00*X**4+ 1.00*X**3+ 1.00*X**2+ 3.00*X+ 9.00 =0 ***

*** IERR= 0 (CALLED BY FORTRAN)

X= 0.106E+01 0.
X= -0.902E+00 0.525E+00
X= -0.902E+00 -0.525E+00
X= 0.185E+00 0.969E+00
X= 0.185E+00 -0.969E+00

☒ 19-13

```

LINE  +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1+   /* TEST OF SDBAIR (KOUJI HOUTEISHIKI NO KAI) */
2     SD:PROC OPTIONS(MAIN);
3     DCL (A(6),XR(5),XI(5),W1(5),W2(5)) FLOAT BIN(27);
4     DCL (N,L,LI,NR,IERR) FIXED BIN(35);
5     DCL SDBAIR ENTRY(FIXED BIN(35),(6)FLOAT BIN(27),FIXED BIN(35),
6+    FIXED BIN(35),FIXED BIN(35),(5)FLOAT BIN(27),(5)FLOAT BIN(27),
7+    (5)FLOAT BIN(27),(5)FLOAT BIN(27),FIXED BIN(35))OPTIONS(FORTRAN);
8     N=5;
9     L=100;
10    LI=7;
11    PUT PAGE;
12    L1: GET LIST(A) COPY;
13    IF A(1)=0 THEN STOP;
14    PUT      EDIT('*** 5-JI HOUTEISHIKI : ',A(1),'*x**5+',A(2),
15+    '*x**4+',A(3),'*x**3+',A(4),'*x**2+',A(5),'*x+',A(6),' =0 ***'
16+    )(SKIP(3), X(5),A,F(3),A,F(3),A,F(3),A,F(3),A,F(3),A);
17    CALL SDBAIR(N,A,L,LI,NR,XR,XI,W1,W2,IERR);
18    PUT SKIP(2);
19    PUT EDIT('*** IERR=',IERR,' (CALLED BY PL/I)')(X(5),A,F(2),A);
20    PUT EDIT('(X=',XR(I),XI(I) DO I=1 TO 5) )(SKIP(2),X(5),A,
21+    (2)E(13,3));
22    PUT SKIP(2);
23    GO TO L1;
24    END SD;

```

☒ 19-14

2,3,4,5,6,7

```

*** 5-JI HOUTEISHIKI : 2*x**5+ 3*x**4+ 4*x**3+ 5*x**2+ 6*x+ 7 =0 ***
*** IERR= 0 (CALLED BY PL/I)
X= -1.300E+000 0.000E+000
X= -6.728E-001 1.107E+000
X= -6.728E-001 -1.107E+000
X= 5.729E-001 1.130E+000
X= 5.729E-001 -1.130E+000

```

22,4,30,29,1,26

```

*** 5-JI HOUTEISHIKI : 22*x**5+ 4*x**4+ 30*x**3+ 29*x**2+ 1*x+ 26 =0 ***
*** IERR= 0 (CALLED BY PL/I)
X= -1.043E+000 0.000E+000
X= 3.614E-001 8.722E-001
X= 3.614E-001 -8.722E-001
X= 6.932E-002 1.125E+000
X= 6.932E-002 -1.125E+000

```

-8,-3,1,1,3,9

```

*** 5-JI HOUTEISHIKI : -8*x**5+ -3*x**4+ 1*x**3+ 1*x**2+ 3*x+ 9 =0 ***
*** IERR= 0 (CALLED BY PL/I)
X= 1.060E+000 0.000E+000
X= -9.024E-001 5.246E-001
X= -9.024E-001 -5.246E-001
X= 1.849E-001 9.694E-001
X= 1.849E-001 -9.694E-001

```

0,0,0,0,0,0

☒ 19-15

```

1      C      *** TEST OF SKTRPZ (DAIKEI KUOSHIKI NI YORU SEKIBUN) ***
2      EXTERNAL FUNC1
3      XA=1.0
4      XB=2.0
5      L=100
6      WRITE(6,100) XA,XB,L
7      100 FORMAT(1H1//6X,'*** DAIKEI KOUISHIKI NI YORU SEKIBUN ***'//6X,
8      &'F(X)=15*X**4+8*X**3-9*X**2+4*X+1'//6X,'XA=',F5.2,'XB=',F5.2,
9      &25X,'L=',I4)
10     CALL SKTRPZ(XA,XB,FUNC1,L,XIG,IERR)
11     WRITE(6,200) IERR,XIG
12     200 FORMAT(/6X,'IERR=',I2,10X,'SEKIBUNCHI=',F8.3,' (CALLED BY FORTRAN
13     &)' )
14     STOP
15     END

```

☒ 19-16

```

1      C      ***FUNCTION FUNC1 ***
2      FUNCTION FUNC1(X)
3      FUNC1=15.0*X**4+8.0*X**3-9.0*X**2+4.0*X+1.0
4      RETURN
5      END

```

☒ 19-17

```

*** DAIKEI KOUISHIKI NI YORU SEKIBUN ***

F(X)=15*X**4+8*X**3-9*X**2+4*X+1

XA= 1.00      XB= 2.00      L= 100

IERR= 0      SEKIBUNCHI= 109.004 (CALLED BY FORTRAN)

```

☒ 19-18

```

LINE  +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1+    /* TEST OF SKTRPZ (DAIKEI KOUISHIKI NI YORU SEKIBUN) */
2      SKT:PROC OPTIONS(MAIN);
3      DCL FLIB1 ENTRY (FLOAT BIN(27),FLOAT BIN(27),FIXED BIN(35),
4+      FLOAT BIN(27),FIXED BIN(35)) OPTIONS(FORTRAN);
5      DCL (XA,XB,XIG) FLOAT BIN(27);
6      DCL (L,IERR) FIXED BIN(35);
7      XA=1.0;
8      XB=2.0;
9      L=100;
10     PUT PAGE EDIT('*** DAIKEI KOUISHIKI NI YORU SEKIBUN ***',
11+     'F(X)=15*X**4+8*X**3-9*X**2+4*X+1', 'XA=',XA, 'XB=',XB, 'L=',L)
12+     (SKIP(5),X(5),A,SKIP(2),X(5),A,SKIP(2),X(5),A,F(5,2),X(5),A,F(5,2)
13+     ,X(5),A,F(4));
14     CALL FLIB1(XA,XB,L,XIG,IERR);
15     PUT EDIT('IERR=',IERR,'SEKIBUNCHI=',XIG,' (CALLED BY PL/I)')
16+     (SKIP(2),X(5),A,F(2),X(10),A,F(8,3),A);
17     END SKT;

```

☒ 19-19

```

1      C      ***FLIB1 ***
2          SUBROUTINE FLIB1(XA,XB,L,XIG,IERR)
3          EXTERNAL FUNC1
4          CALL SKTRPZ(XA,XB,FUNC1,L,XIG,IERR)
5          RETURN
6          END

```

図 19-20

```

*** DAIKEI KOUSHIKI NI YORU SEKIBUN ***
F(X)=15*X**4+8*X**3-9*X**2+4*X+1
XA= 1.00      XB= 2.00      L= 100
IERR= 0      SEKIBUNCHI= 109.004 (CALLED BY PL/I)

```

図 19-21

(例4) 行列と行列の積 (SAMMLT)

行列どおしのかけ算である。ここでは6行5列の行列Aと、5行7列の行列Bをかけて6行7列の行列Cを求めてみる。FORTRANのメインプログラムを図19-22に、その実行結果を図19-23に示す。PL/Iのメインプログラムを図19-24に、その実行結果を図19-25に示す。第7章で述べたように、2次元以上の配列の場合、PL/IとFORTRANではその要素の並び方が異なり、PL/Iでは横走査、FORTRANでは縦走査の順に並んでおり、そのままでは対応しないのだが、この例を見ると分るように全く同じ実行結果となっており、結合の際にコンパイラがうまく変換してくれているらしい。変換プログラムを作って、要素の並び方を変換してから受け渡しを行うようにしてみたが、かえってオカシな結果になってしまった。従って2次元以上の配列でもそのまま受け渡しをすればよいようである。

以上、5つの実験例によってPL/IとFORTRANの結合方法を示した。

全6回にわたってPL/Iの入門をFORTRANと対応づけながら解説してきた。説明が十分でなく分りにくい箇所もあったかも知れないが、とにかくPL/Iとはどのような言語かという大筋は大体分っていただけたのではないかと思う。この拙稿を読まれてPL/Iに興味を持たれた方は是非一度PL/Iプログラムを作ってみることをお勧めする。ACOS-6 PL/I (又は標準PL/I)は優秀なコンパイラだから、あなたの初めて作ったPL/Iプログラムを適確に採点してくれるであろう。長い間、読んで頂き有難う。 /* SEE YOU AGAIN */

(プログラム相談員)

```

1      C      *** TEST OF SAMMLT (GYURETSU NO KAKEZAN) ***
2      REAL   A(6,5),B(5,7),C(6,7)
3      MA=6
4      MB=5
5      MC=6
6      N=6
7      M=5
8      L=7
9      READ(5,*) ((A(I,J),J=1,5),I=1,6), ((B(I,J),J=1,7),I=1,5)
10     WRITE(6,100)((A(I,J),J=1,5),I=1,6), ((B(I,J),J=1,7),I=1,5)
11     100    FORMAT(1H1//6X,'*** TEST OF SAMMLT (BY FORTRAN) ***' //6X,'MATRIX
12     & A : '//6X,6(/6X,5F7.2)//6X, 'MATRIX B : '//6X,5(/6X,7F7.2))
13     CALL SAMMLT(MA,MB,MC,N,M,L,A,B,C,IERR)
14     WRITE(6,200) IERR, ((C(I,J),J=1,7),I=1,6)
15     200    FORMAT('//6X, 'IERR=',I3//6X,'MATRIX C=A*B : '//6X,6(/6X,7F9.2))
16     STOP
17     END

```

図 19-22

*** TEST OF SAMMLT (BY FORTRAN) ***

MATRIX A :

1.00	2.00	3.00	4.00	5.00
6.00	7.00	8.00	9.00	10.00
11.00	12.00	13.00	14.00	15.00
16.00	17.00	18.00	19.00	20.00
21.00	22.00	23.00	24.00	25.00
26.00	27.00	28.00	29.00	30.00

MATRIX B :

1.00	2.00	3.00	4.00	5.00	6.00	7.00
8.00	9.00	10.00	11.00	12.00	13.00	14.00
15.00	16.00	17.00	18.00	19.00	20.00	21.00
22.00	23.00	24.00	25.00	26.00	27.00	28.00
29.00	30.00	31.00	32.00	33.00	34.00	35.00

IERR= 0

MATRIX C=A*B :

295.00	310.00	325.00	340.00	355.00	370.00	385.00
670.00	710.00	750.00	790.00	830.00	870.00	910.00
1045.00	1110.00	1175.00	1240.00	1305.00	1370.00	1435.00
1420.00	1510.00	1600.00	1690.00	1780.00	1870.00	1960.00
1795.00	1910.00	2025.00	2140.00	2255.00	2370.00	2485.00
2170.00	2310.00	2450.00	2590.00	2730.00	2870.00	3010.00

図 19-23

```

LINE  +-----+-----+-----+-----+-----+-----+-----+-----+-----+
1+    /* TEST OF SAMMLT (GYOURETSU NO KAKEZAN) */
2      SAM:PROC OPTIONS(MAIN);
3      DCL (A(6,5),B(5,7),C(6,7),N(50)) FLOAT BIN(27);
4      DCL (MA,MB,MC,N,M,L,IERR,I,J,K) FIXED BIN(35);
5      DCL SAMMLT ENTRY(FIXED BIN(35),FIXED BIN(35),FIXED BIN(35),
6+          FIXED BIN(35),FIXED BIN(35),FIXED BIN(35),
7+          (**)FLOAT BIN(27),(**)FLOAT BIN(27),(**)FLOAT BIN(27),
8+          FIXED BIN(35)) OPTIONS(FORTRAN);
9      MA=6;
10     MB=5;
11     MC=6;
12     N=6;
13     M=5;
14     L=7;
15     GET LIST(A,B);
16     PUT PAGE EDIT('*** TEST OF SAMMLT (BY PL/I) ***','MATRIX A :',
17+         A,'MATRIX B :',B)(SKIP(3),X(5),A,SKIP(2),X(5),A,
18+         SKIP(2),X(5),C)(SKIP(1),X(5),C),F(7,2)),SKIP(2),
19+         X(5),A,SKIP(2),X(5),C)(SKIP(1),X(5),C),F(7,2));
20     CALL SAMMLT(MA,MB,MC,N,M,L,A,B,C,IERR);
21     PUT EDIT('IERR=',IERR,'MATRIX C=A*B :',C)(SKIP(2),X(5),A,F(3),
22+         SKIP(2),X(5),A,SKIP(2),X(5),C)(SKIP(1),X(5),C),F(9,2));
23     END SAM;

```

☒ 19-24

*** TEST OF SAMMLT (BY PL/I) ***

MATRIX A :

1.00	2.00	3.00	4.00	5.00
6.00	7.00	8.00	9.00	10.00
11.00	12.00	13.00	14.00	15.00
16.00	17.00	18.00	19.00	20.00
21.00	22.00	23.00	24.00	25.00
26.00	27.00	28.00	29.00	30.00

MATRIX B :

1.00	2.00	3.00	4.00	5.00	6.00	7.00
8.00	9.00	10.00	11.00	12.00	13.00	14.00
15.00	16.00	17.00	18.00	19.00	20.00	21.00
22.00	23.00	24.00	25.00	26.00	27.00	28.00
29.00	30.00	31.00	32.00	33.00	34.00	35.00

IERR= 0

MATRIX C=A*B :

295.00	310.00	325.00	340.00	355.00	370.00	385.00
670.00	710.00	750.00	790.00	830.00	870.00	910.00
1045.00	1110.00	1175.00	1240.00	1305.00	1370.00	1435.00
1420.00	1510.00	1600.00	1690.00	1780.00	1870.00	1960.00
1795.00	1910.00	2025.00	2140.00	2255.00	2370.00	2485.00
2170.00	2310.00	2450.00	2590.00	2730.00	2870.00	3010.00

☒ 19-25

参 考 文 献

- (1) 日本電気：“ACOS-6プログラム管理PL/I文法説明書（FGD04-1）”
- (2) “ ”：“ ” “ ” PL/Iプログラミング説明書（FGD05-1）”
- (3) “ ”：“ ” “ ” 標準PL/I文法説明書（FGD02-3）”
- (4) “ ”：“ ” “ ” 標準PL/Iプログラミング説明書（FGD03-4）”
- (5) “ ”：“ ” “ ” PL/I概説書（FGD01-2）”
- (6) “ ”：“ ” “ ” FORTRAN文法説明書（FGB02-3）”
- (7) “ ”：“ ” “ ” FORTRANプログラミング説明書（FGB03-4）”
- (8) “ ”：“ ” “ ” FORTRANサブルーチンライブラリ説明書
（FGB04-4）
- (9) “ ”：“ACOS-2/4/6 数値計算ライブラリ説明書（MATHLIB-2/4/6
概念/機能編）（FXF01-5）
- (10) “ ”：“ ” “ ” （MATHLIB-2/4/6
例題編）（FXF03-3）
- (11) 富士通：“FACOM OSW PL/I文法書（64SP-3061-1）”
- (12) “ ”：“ ” “ ” FORTRAN文法書（64SP-3030-2）”
- (13) “ ”：“FACOM PL/I基礎コース（45ET-0221-3）”
- (14) “ ”：“ ” “ ” PL/I上級コース（45ET-0231-3）”
- (15) 日本ソフトウェア：“PL/Iの学び方（初級編）”，東京電機大学出版局（1971）。
- (16) 大日方，小川：“PL/Iの学び方（中級編）”，東京電機大学出版局（1972）。
- (17) 竹下 亨：“入門PL/I”，オーム社（1972）。
- (18) 二宮，谷山，高梨：“PL/Iプログラミング入門”，森北出版（1977）。
- (19) C. T. フェイク著，赤木，加賀美，鈴木共訳：“科学者のためのPL/I”，共立出版
（1973）
- (20) 大泉充郎監修：“JISに準拠したFORTRAN基本コース”，オーム社（1968）。
- (21) “ ”：“ ” “ ” “ ” 拡充コース”，オーム社（1969）。

以 上