



Title	FORTTRAN 77概説 (1)
Author(s)	大中, 幸三郎; 後藤, 米子
Citation	大阪大学大型計算機センターニュース. 1981, 43, p. 59-78
Version Type	VoR
URL	https://hdl.handle.net/11094/65501
rights	
Note	

The University of Osaka Institutional Knowledge Archive : OUKA

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

The University of Osaka

FORTRAN 77 概説 (1)

大阪大学大型計算機センター研究開発部

大 中 幸三郎 ・ 後 藤 米 子

1. はじめに

当センターで従来から用いられているFORTRANの仕様はJIS FORTRANの水準7000に機能を追加したものでした。このJIS規格は元をたどればISO規格、さらには1966年に制定されたASA規格までさかのぼることができます。当センターの場合のみならず、計算機メーカーから提供されるFORTRANは、各々のメーカーがこれらの規格に機能の追加を行ったもので、各メーカー間にはかなりの差異があり、プログラムの互換性の面で種々の問題となっています。

本来、FORTRANは数値計算用の言語ですが、計算機の利用分野の拡大、問題の大型化ならびに複雑化などにより、上記の規格に基づくFORTRAN(以下FORTRAN66と言う)では対応が困難となってきました。各種の追加機能はFORTRANを少しでも使いやすくしようとしたものですが、なにぶんにも規格外であり、標準言語としてのFORTRANの互換性をそこなうものでしかありません。抜本的な対策となる規格の改正作業は1970年からアメリカのANSI(旧名ASA)で始められ、1977年にはFORTRAN77と言う名前で原案が発表されました。このFORTRAN77はANSI規格として1978年に制定され、ISOでも採用となりました。これに伴い、JIS規格の改正も今年度中には発表される予定です。

FORTRAN77では、文字処理とファイル操作の機能が追加され、一部分では構造化プログラミング機能もとり入れられています。他にも追加、変更になった機能が数多くありますが、その中にはすでにACOS FORTRAN66にとり入れられている機能も多くみられます。本稿ではACOSのFORTRAN66と77の相異を中心にしてACOS FORTRAN77の仕様について述べます。

2. 動作モードと運用

ACOS FORTRAN66は、コンパイラオプションによって分類することができました。その主なものは、

動作モード	RモードとVモード
浮動小数点数の内部表現	HEXモードとBINモード
内部コード	JISコードとBCDコード

の3点ですが、これらは選択の範囲が広がるという長所よりも、オブジェクトプログラムの互換性がないという短所の方が目立ちました。ACOS FORTRAN77ではHEXモードとJISコードに統一され、BINモードとBCDコードは廃止されます。RとVの両モードは残りますが、現在使用できるのはRモードのみで、Vモードが当センターで使用可能となるのは、早くても来年の5～6月頃となる予定です。

当センターでの運用は、当面はFORTRAN66と77の2つが併存する形となりますが、FORTRAN77のVモードが使用可能となった後、適当な時期にFORTRAN77に統一する予定です。今後新しくFORTRANでプログラムを作られる場合は、極力、FORTRAN77をお使い下さい。なお、ACOS FORTRAN77にはACOS FORTRAN66と互換性を保つためのコンパイラオプションとしてCOMPATオプションが用意されており、機能としてはACOS FORTRAN66のHEXモード、JISコードの場合を含んでいます。しかしながら、COMPATオプション指定時の仕様は、本来のFORTRAN77とは言えず、移行時に使われる以外は使用しない方が良くと思います。COMPATオプション指定時の仕様については、以下で述べませんから、マニュアルを御覧下さい。

3. 文字と文

3.1 FORTRAN文字集合

英 字 A～Z および \$

数 字 0～9

特殊文字 空白 = + - * / () , . & ' : ; "

JIS FORTRAN77では\$は特殊文字となっている。ACOS FORTRAN77でもコンパイラオプションにDLRを指定すると、\$は特殊文字として扱われる。

3.2 文と行の順序

下図は、プログラム単位を構成する文と注釈行の順序を規定したものである。縦線は混在できる文の種類を示し、横線は順序の定まっている文を示す。

注釈行	PROGRAM文, FUNCTION文, SUBROUTINE文 またはBLOCK DATA文			
	FORMAT文 および ENTRY文	PARAMETER文	IMPLICIT文	
			他の宣言文	DATA文
	文関数定義文			
	END文以外の実行文			
E N D 文				

ただし、上図以外に次の4つの順序関係がある。

- 初期値を与えている文は、その中で初期値を与えられている要素に対する宣言文よりも後になければならない。
- ENTRY文は、ブロックIF文とそれに対応するEND IF文の間およびDO文とそのDOループの端末文の間に現れてはならない。
- 定数名の型を指定する宣言文は、その定数名を定義するPARAMETER文より前になければならない。また、PARAMETER文は、そのPARAMETER文によって定義されている定数名を引用する他の文の前になければならない。
- 配列の寸法宣言子に現れる定数名および変数名が、暗黙の型宣言によって整数型に指定されていないならば、その配列宣言を行っている文よりも前に、IMPLICIT文または型宣言文で整数型として宣言しなければならない。

なお、JIS FORTRAN 77では、DATA文はすべての宣言文の後となっており、ACOS FORTRAN 77とは異っている。

4. データの型

4.1 浮動小数点数の内部表現

実数型、倍精度実数型、4倍精度実数型、複素数型、倍精度複素数型および4倍精度複素数型の内部表現はすべて16進指数部浮動小数点形式であり、ACOS FORTRAN 66にあった2進指数部浮動小数点形式は廃止された。

4.2 論理型の内部表現

論理型は1語(36ビット)を占める。値が真のときは右端のビットが1、他のビットは0となり、偽のときは全てのビットが0となる。ACOS FORTRAN 66では、36ビット中、いずれかのビットが1であれば真、全てのビットが0であれば偽であった。

4.3 文字型

文字型のデータは、JISコードで表現できる文字によって構成された列である。この列は1～

5 1 1文字から成り、各々の文字は1バイト(9ビット)を占める。文字数が4の倍数でない場合、語の境界とずれることに注意を要する。ACOS FORTRAN 66では語の境界と合せるために使用できないバイトがそう入された。

文字定数の形は ' $h_1 h_2 \cdots h_n$ ' または " $h_1 h_2 \cdots h_n$ " の形である。ただし、 h_j は JISコードで表現できる文字とする。JIS FORTRAN 77では ' $h_1 h_2 \cdots h_n$ ' のみであり、" $h_1 h_2 \cdots h_n$ " は許されていない。

5. 配列と部分列

5.1 配列の寸法

配列の寸法宣言子にコロンで区切って寸法の下限と上限(下限 \leq 上限)を指定することができる。この下限と上限は寸法の限界式と呼ばれる算術式であり、その式の中のすべての定数、定数名および変数は整数型でなければならない。ただし、式の中に関数あるいは配列要素の引用を含んではならない。FORTRAN 66と同じ形式の寸法宣言子は、上限のみが指定されたものとして扱われ、下限は1とみなされる。FORTRAN 66では、下限を指定できないばかりでなく、寸法の限界も整数型の定数、定数名および変数であり、式を書くことはできなかった。

例 DIMENSION A(10), B(I, 5), C(5*10, 2*K+1), D(-10:10, 10)
FORTRAN 66では不可

限界式の中に現れる定数名および変数名が、暗黙の型宣言によって整数型に指定されていないならば、限界式の中に現れるより前に、IMPLICIT文または型宣言文で整数型として宣言しておかなければならない。

5.2 添字

添字式は算術式である。添字式の値が整数型以外の型であれば、算術変換の規則によって整数型に変換される。JIS FORTRAN 77では、整数型の算術式のみが許されている。

プログラム単位内で、各々の添字式の値は、その配列宣言子の対応する寸法の下限より大きいかまたは同じでなければならない。ACOS FORTRAN 77では、添字式の値が対応する寸法の上限を超えることも許されているが、JIS FORTRAN 77では禁止されている。

添字の値は、配列要素を定めるものであり、添字の値が p のとき、配列の p 番目の要素が識別される。添字の値と添字式の値の意味が異なることに注意を要する。

例 配列宣言子 A(-1:1, 2)によって規定された配列 A の要素は、下記に示す添字の順序に従って並んでいる。

添字	(-1, 1)	(0, 1)	(1, 1)	(-1, 2)	(0, 2)	(1, 2)
添字の値	1	2	3	4	5	6

5.3 整合配列と整合寸法

整合寸法に含まれる変数名は、仮引数またはその副プログラム内の共通ブロックのいずれかに現れなければならない。

整合配列を含む副プログラムの実行中に、整合寸法に含まれる変数が再定義されたり不定となったりしても、配列の性質（次元の数、各寸法の大きさと限界、要素の数と順序）は変わらない。

FORTRAN 66では、整合寸法に含まれる変数を再定義したり不定とすることは禁止されていた。

5.4 擬寸法配列

擬寸法配列は、配列の宣言において最後の次元の寸法の上限が星印であるものを言う。他の寸法の限界式は5.1で述べたものと同じである。

擬寸法配列は仮配列であり、その大きさ（要素の数）は対応する実引数の添字の値が最大の要素までとなる。すなわち、擬寸法配列の大きさは次の規則によってきまる。

- 実引数が文字型以外の配列名るとき、仮配列の大きさは実引数の配列の大きさとなる。
- 実引数が大きさ n の配列中の添字の値 p をもった文字型以外の配列要素名るとき、仮配列の大きさは $n + 1 - p$ となる。
- 実引数が文字型の配列名、配列要素名または配列要素の部分列名（5.5参照）るとき、実引数が s バイトの大きさをもつ配列の t バイト目から始まり、仮配列の一要素が l バイトであるとすれば、仮配列の大きさは $\text{INT}((s + 1 - t) / l)$ となる。

例 下記のプログラムでは、サブルーチンSUBの中で仮配列BとYをB(3, 2), Y(2, 3)と宣言することに等しい。

```
DIMENSION A(3, 2), X(3, 2)      SUBROUTINE SUB(B, Y)
      )                          DIMENSION B(3, *), Y(2, *)
CALL SUB(A, X)                  )
      )                          )
END                              END
```

FORTRAN 66に擬寸法配列と対応するものはなかった。しいて対応づけるならば、最後の次元の寸法に1を書いたものと対応するが、この表現は文法違反であった。

5.5 文字部分列

文字部分列は、文字型のデータの隣接している一部分であり、その型は文字型である。文字部分列は、部分列名で識別され、値を割当てることおよび引用することができる。FORTRAN 66には、文字部分列に対応する機能はなかった。

部分列名は $v(e_1:e_2)$ の形である。ここで、 v は文字型の変数名または配列要素名である。 e_1 および e_2 は、算術式であり、文字位置式（部分列式）と言う。文字位置式の値が整数型以外の型であれば、算術変換の規則に従って整数型に変換される。JIS FORTRAN 77では、文字位置

式は整数型の算術式となっており、ACOS FORTRAN77 より制限されている。

e_1 の値は部分列の左端の文字位置を指定し、 e_2 の値は右端を指定する。ただし、 $1 \leq e_1 \leq e_2 \leq l$ でなければならない。ここで、 l は文字型の変数または配列要素の長さ(文字数)である。 e_1 を省略すると1とみなされ、 e_2 を省略すると l とみなされる。

6. 式

6.1 算術式

算術定数式は、算術定数、算術定数名、算術演算子(+, -, *, /, **)および括弧のみから構成される算術式である。ただし、べき乗演算子は、そのべき指数が整数型のときにのみ算術定数式に現れることができる。

例 2.0 ** 2 算術定数式
2.0 ** 2.0 算術式ではあるが、算術定数式ではない。

ACOS FORTRAN77 では算術定数式の値はすべてコンパイル中に決定され、実行時にはその値が引用されるのみである。

6.2 文字式

文字列の連結に文字演算子//を用いる。 x_1 と x_2 が演算子の左側と右側のオペランドを表すものとする、 $x_1//x_2$ の演算結果は、 x_1 の右側に x_2 を連結した値をもつ文字列であり、その長さは x_1 の長さ+ x_2 の長さの和となる。

例 'ABC'//'XYZ'の値は、文字列ABCXYZである。

6.3 関係式

関係式は、二つの算術式の値あるいは二つの文字式の値を比較するために使用される。関係式によって、算術式の値と文字式の値を比較することはできない。

文字式の値を比較する場合の文字の大小順序は、JISコード体系によって定まる。二つの文字式の値の長さが異なる場合には、短い方のオペランドの右側に空白を補い、長い方のオペランドの長さまで拡張したものと考える。

6.4 論理式

論理演算子に論理等価(.EQV.)と論理非等価(.NEQV.)を追加する。これらの真理値表と従来からの論理演算子との優先順位を次に示す。

x_1	x_2	$x_1.EQV.x_2$	$x_1.NEQV.x_2$
真	真	真	偽
真	偽	偽	真
偽	真	偽	真
偽	偽	真	偽

演算子	優先順位
.NOT.	高 位
.AND.	}
.OR.	
.EQV., .NEQV.	低 位

6.5 演算子の優先順位

算術演算子の優先順位は従来通りであり、論理演算子については6.4に記した。文字演算子は一つしかなく、関係演算子の間に優先順位はない。これらの演算子の間の優先順位を右に示す。

演 算 子	優先順位
算術演算子	高 位
文字演算子	}
関係演算子	
論理演算子	低 位

括弧が用いられている場合には、括弧が優先し、次に演算子の優先順位に従う。同じ優先順位をもつ演算子の解釈

は、べき乗が右から左になっている以外、すべては左から右に解釈される。

6.6 式の評価

文の中の関数引用の実行によって、その文の中の他の要素の値を変えてはならない。共通ブロック中の要素の値が、文の中の関数引用の値に影響を与える場合、その文の中の他の関数引用の実行は、その要素の値を変えてはならない。ただし、論理 I F 文中の分枝の条件を与える論理式内の関数引用の場合は、その実行によって、論理式が真のときに実行される論理 I F 文中の文の要素を変更してもよい。

例 関数 $G(X)$ が X を定義する場合、 $Y=G(X)$ は許されるが、 $Y=G(X)+X$ は禁止される。

式のすべてのオペランドを評価しないで式の評価を決定できる場合には、式の一部のオペランドを評価しないことがある。

例 論理式 $X.GT.Y.OR.L(Z)$ において、 X が Y より大きければ論理型の関数 $L(Z)$ を評価するとはかぎらない。したがって、 L がその引数を定義するものであれば、論理式の実行によって Z は不定となることがある。

文が関数引用を二つ以上含む場合、おのおの関数引用によって得られる値は、コンパイラがその関数引用を評価する順序と無関係でなければならない。

7. 宣 言 文

7.1 DIMENSION文

DIMENSION文は、配列の英字名と寸法を宣言するのに用いられるが、COMMON文あるいは型宣言文によって配列を宣言することもできる。

7.2 EQUIVALENCE文

EQUIVALENCE文による結合は、並びに現れたそれぞれの要素の最初のバイトが同じ記憶場所を共有することを指定する。その並びは、変数名、配列要素名、配列名および部分列名から成るが、いずれの場合も仮引数であってはならない。並びの中の添字式と文字位置式は整定数式でなければならない。

ACOS FORTRAN 66, 77のいずれの場合においても、倍精度実数型、4倍精度実数型、複素数型、倍精度複素数型および4倍精度複素数型のデータは、最初の語が偶数番地となるように記憶される(以下、偶数境界と言う)。この制限はJIS FORTRAN 77に違反する。したがって、ACOS FORTRAN 77では、コンパイラオプションにNALCを指定することにより、偶数境界の制限は解除される^{注)}。通常は、偶数境界となるALCオプションであり、NALCオプションの場合よりも、実行速度が早い。偶数境界はCOMMON文による結合(7.3参照)においても制限となる。

文字型のデータは、他の文字型のデータとだけ結合することができる。結合されるデータの長さは、同じである必要はない。FORTRAN 66では文字型と他の型を結合することは可能であったことに注意を要する。

配列要素名がEQUIVALENCE文中に現れた場合、その添字式の数は配列宣言子の次元の数以下であればよい。添字式の数が次元の数より小さい場合には、不足する添字式の値として、対応する次元の寸法の下限が指定されたものとして扱われる。JIS FORTRAN 77では、添字式の数と配列宣言子の次元の数は一致しなければならない。

7.3 COMMON文

名前付き共通ブロック、無名共通ブロックにかかわらず、共通ブロックの最初のバイトは必ず偶数番地の語の最初のバイトに割り当てられる。

一つの共通ブロック内に、文字型と他の型を混在させることは禁止されている。ACOS FORTRAN 66では、文字型と他の型を混在させることができた。

7.2で述べた偶数境界は、COMMON文の場合にも制限事項となり、JIS FORTRAN 77に違反する。すなわち、ALCオプションの場合は、偶数境界となるようにダミーの語がそう入される。

注) NALCオプションは現在使用できない。使用可能となるのは早くとも昭和57年5~6月頃の予定である。

例 下記の結合を行ったプログラムは、ALCオプションでは違反であり、NALCオプションを指定してコンパイルしなければならない。

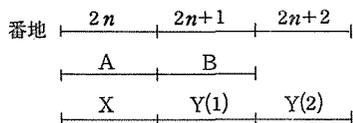
```
DOUBLE PRECISION A,B
DIMENSION C(3)
EQUIVALENCE(A,C), (B,C(2))
```

例 DIMENSION C(2)
CHARACTER A*4, B*4, C*3
EQUIVALENCE(A, C(1)), (B, C(2))

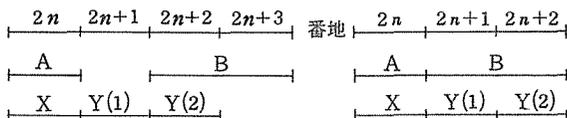
例 右のプログラムの中A, B, X, Y の対応は、データの型と、

ALC, NALCのオプションに依存する。

○ A, B, X, Y がすべて実数型の場合の対応は、オプションに依存しない。



○ A, X, Y が実数型、Bが倍精度実数型の場合の対応は、オプションに依存する。(左がALC, 右がNALC)



```
SUBROUTINE SUB1
```

```
  型宣言文
```

```
COMMON A, B
```

```
END
```

```
SUBROUTINE SUB2
```

```
  型宣言文
```

```
COMMON X, Y(2)
```

```
END
```

7.4 型宣言文(1)

型宣言文は次の9つである。

INTEGER, REAL, DOUBLE PRECISION, QUADRUPLE PRECISION,
COMPLEX, COMPLEX DOUBLE PRECISION,
COMPLEX QUADRUPLE PRECISION, LOGICAL, CHARACTER.

型宣言文によって配列宣言と初期値設定を行うことができる。JIS FORTRAN 77では、型宣言文によって初期値を与えることはできない。

総称関数名 (ACOS FORTRAN 66 では一般名) が型宣言文に現れた場合、総称性を失う。JIS FORTRAN 77では総称性は失われず、まったく逆となっている。組込み関数 (ACOS FORTRAN 66 では組込み関数と基本外部関数に対応) を異なる型として宣言した場合、その名前は組込み関数を識別しなくなる。

7.5 型宣言文(2)

CHARACTER文は型宣言文であり、7.4に述べたことはすべて適用される。しかし、他の型宣言文と異なる部分も多く、本節ではCHARACTER文にのみ適用されることを述べる。

CHARACTER文によって、配列宣言と配列の一要素の長さ (文字数) を指定する場合、配列の寸法の指定と一要素の長さ指定の順序が、ACOS FORTRAN 66 とは逆になっている。また、長さ指定を省略すると、1とみなされ、ACOS FORTRAN 66がJISコードで8、BCDコードで6となっていたのと異なる。

例 Aを一要素の長さが4文字で寸法5の配列、Bを長さが1文字の変数、Cを長さが8文字の変数とすれば、次の宣言を行えばよい。

```
FORTRAN 77 CHARACTER A(5)*4, B, C*8
ACOS FORTRAN 66 (JISコード) CHARACTER A*4(5), B*1, C
```

長さ指定として書くことができるものは、

- (1) 符号なし整数。
 - (2) 括弧で囲まれた整数名、整数式。
 - (3) 括弧で囲まれた*印、この場合を擬文字長指定と言う。
- } ACOS FORTRAN 66
} では不可。

の3種類である。ただし、配列の場合には、長さ指定は各配列要素の長さを指定することになる。ACOS FORTRAN 66では整数名に対して括弧は不要であり、(2)の形とは異なる。

擬文字長指定の使用できる範囲は、(1)と(2)の場合よりも制限されており、次の3つの場合しかない。

- 外部関数名で、同じ副プログラム中のFUNCTION文またはENTRY文に現れているもの。このような関数が実行される場合、その長さは、その関数を引用しているプログラム単位内で指定された長さとなる。
- 仮手続きを除く仮引数。この場合の長さは、実引数の長さとなる。
- 定数名。この場合の長さは、PARAMETER文に現れた対応する定数式の長さとなる。

7.6 IMPLICIT文

一つのプログラム単位内のすべてのIMPLICIT文において、同じ英字が、一つの英字としてまたは英字の範囲に二度以上現れると、後に現れた宣言が優先する。JIS FORTRAN 77では、二度以上現れることを禁止している。

7.7 PARAMETER文

PARAMETER文は、次の形である。

```
PARAMETER (  $p_1 = e_1, p_2 = e_2, \dots, p_n = e_n$  )
```

ただし、 p_j は定数名、 e_j は定数式とする。

定数名 p_j の型は、暗黙の型宣言に従わない場合には、その定数名がPARAMETER文の最初に現れるより前に、その型を型宣言文またはIMPLICIT文で指定しなければならない。ACOS FORTRAN 66では、PARAMETER文の並びを括弧で囲むことはせず、また、 p_j の型は e_j の型によって決定された。

定数名 p_j が整数型、実数型、倍精度実数型、4倍精度実数型、複素数型、倍精度複素数型あるいは4倍精度複素数型である場合には、 e_j は算術定数式(6.1参照)でなければならない。定数名 p_j が文字型あるいは論理型である場合には、 e_j の型は p_j の型と一致していなければならない。定

数名 p_j の値は、 e_j から代入文の規則によって定められる値となる。

定数式 e_j に現れる定数名は、同じ PARAMETER 文または同じプログラム単位内の他の PARAMETER 文によって、前もって定義されていなければならない。

定数名を他の定数の一部として用いてはならない。ただし、複素定数の実部または虚部に現われることは許される。JIS FORTRAN 77 ではこの場合にも禁止となっている。

例 COMPLEX C

```
PARAMETER ( I=2, PAI=3.14, PAI2=2*PAI, C=(PAI, 1.0) )
```

JISでは禁止

ACOS FORTRAN 66 では PARAMETER 文の並びの括弧と COMPLEX 文は不要であった。

7.8 EXTERNAL 文

EXTERNAL 文に初期設定副プログラム名を書くことができる。ACOS FORTRAN 66 では、名前付の初期値設定副プログラムは存在しなかった。

EXTERNAL 文に総称関数名が現れた場合、その総称名の下にある組込み関数名の中で総称名と異なる名前は、依然として組込み関数を識別する。

7.9 INTRINSIC 文

INTRINSIC 文に次の 10 種類の組込み関数名を書くことはできない。すなわち、これらの組込み関数は実引数として使用できないことになる。

型変換、最大値、最小値、文字列の大小比較、論理積、論理和、論理否定、排他的論理和、ビットフィールドの抽出、型の無視。

総称関数名が INTRINSIC 文に現れても、その英字名は総称性を失うことはない。ただし、実引数として渡される名前は、総称名ではなく、個々の組込み関数の名前である。

7.10 SAVE 文

SAVE 文は、副プログラム内の RETURN 文あるいは END 文の実行後、要素の定義状態を保つために用いられる。SAVE 文で指定された要素は、その副プログラム内の RETURN 文あるいは END 文の実行の結果として不定とはならない。SAVE 文に対応する文は、FORTRAN 66 にはなかった。

SAVE 文は、次の形である。

```
SAVE  $a_1, a_2, \dots, a_n$ 
```

ただし、 a_j は名前付共通ブロック名の前後に斜線を付けたものか、仮引数と共通ブロック内の要素を除いた、変数名あるいは配列名とする。

SAVE 文に並びがない場合には、その副プログラムで SAVE 文に現れることが許されるすべ

てのものを指定したものとみなされる。また、前後を斜線で囲まれた名前付き共通ブロック名が現れた場合には、その共通ブロック内のすべての要素に対して効果がある。

実行可能プログラム中の一つの副プログラムにおいて、ある共通ブロック名をSAVE文に指定した場合、その共通ブロックを宣言しているすべての副プログラムにおいて、その共通ブロック名をSAVE文に指定しなければならない。

共通ブロック中になく局所的な要素がSAVE文に指定されている場合、その要素が副プログラム内のRETURN文あるいはEND文を実行したときに確定であれば、その副プログラムを次に引用したときに、同じ値で確定となっている。

副プログラム内でRETURN文あるいはEND文が実行された場合には、その副プログラム内では以下に示す要素を除いて不定となる。

- (1) SAVE文で指定された要素。
- (2) 無名共通ブロック内の要素。
- (3) 初期確定となっている要素で、再定義されておらず、かつ不定となっていない要素。
- (4) 副プログラム内でCOMMON宣言されており、かつその副プログラムを直接あるいは間接に引用しているプログラム単位のうち、少なくとも一つでCOMMON宣言されている名前付き共通ブロック内の要素。

上記の(4)から、主プログラムでCOMMON宣言されている名前付き共通ブロックは、その共通ブロックを用いている副プログラムにおいて、SAVE文に指定する必要はないことになる。たとえSAVE文に指定しても影響はない。

主プログラム中にSAVE文を書いてもよいが、何の効果もない。

SAVE文の仕様は、JISとACOSの間に差異はない。しかし、ACOS FORTRAN 77では、オーバーレイ構造を用いている場合を除いて、SAVE文を使用しなくても不定とはならない(マニュアルに記述なし)。したがって、ACOSではSAVE文を使用する必要はほとんどない。

8. DATA文

8.1 DATA文の制限

DATA文は、変数、配列、配列要素あるいは部分列に対して初期値を与える文であるが、型宣言文あるいはDIMENSION文で初期値を与えることもできる。JIS FORTRAN 77では型宣言文およびDIMENSION文で初期値を与える機能はない。

名前付き共通ブロック内の要素に対しては、初期値設定副プログラム内でのみ初期値を与えることができる。ACOS FORTRAN 66では、初期値設定副プログラム以外でも可能であった。

初期値が設定される要素と初期値を表わす定数あるいは定数名の間には、8.3で述べる場合を除

いて、1対1の対応がなければならない。ACOS FORTRAN 66 では型も一致しなければならないが、ACOS FORTRAN 77 では、算術変換の規則が適用される場合には、型は一致しなくてもよい。

8.2 で述べる DO 形並びの場合を除いて、添字式および文字位置式は整定数式でなければならない。ACOS FORTRAN 66 では、添字式は整定数あるいは整定数名に制限されていた。

8.2 DATA 文中の DO 形並び

DO 形並びは配列要素あるいは部分列に初期値を与えるのに用いられる。

DO 形並びの動作は、次の3つの制限以外は、DO ループの場合 (10.9 参照) と全く同様である。

- DO 形制御変数は整数型。
- 初期値パラメータ、終値パラメータおよび増分パラメータは整定数式。
- 繰り返し数は正。

DO 形並びに現れる添字式および文字位置式は、整定数式でなければならないが、その添字式および文字位置式を範囲に含む DO 形制御変数を含んでもよい。

JIS FORTRAN 77 では DO 形並びで部分列に初期値を与えることはできない。また、ACOS FORTRAN 66 では、各パラメータは整定数あるいは整定数名であり、添字式の形も制限されていた。

8.3 配列名に対応している文字定数

配列名に対して文字定数あるいは文字定数名が対応していて、かつそのいずれもが並びの最後の要素である場合、すなわち、配列全体に一つの文字定数あるいは文字定数名が対応していることが明らかなる場合には、配列要素の数だけの定数が与えられなくてもよい。

この場合には、配列の一要素に入る文字数ごとに定数が区切られて、順に配列要素に対応づけられる。すべての配列要素に対応しない場合には、対応しない配列要素に初期値は与えられない。

JIS FORTRAN 77 では配列名に一つの定数を対応させる機能はなく、配列要素の数に等しい数の定数を与えなければならない。

9. 代入文

9.1 文番号代入文 (ASSIGN 文)

文番号代入文、すなわち ASSIGN 文の実行により、整数型の変数を文番号の値によってそのプログラム単位内でのみ確定とすることができる。この変数は、JIS FORTRAN 77 で規定されている割当て形 GO TO 文および入出力文の書式識別子として引用される以外に、計算形 GO TO 文、算術 IF 文、CALL 文の選択もどり指定子、入出力文の誤り指定子あるいは入出力文の

ファイル終了指定子でも用いることができる。ACOS FORTRAN 66ではスイッチ変数と呼ばれていたものである。FORTRAN 77にスイッチ変数という名称はないが、本稿では、ASSIGN文によって文番号が代入された整数型の変数をスイッチ変数とすることにする。

9.2 文字代入文

ACOS FORTRAN 66では、左辺の型は任意でよいが、FORTRAN 77では文字型でなければならない。左辺は変数名、配列要素名または部分列名である。

左辺の定義されるいずれの文字位置も、右辺で引用されてはならない。左辺の長さが右辺の長さより長い場合は右側が空白によって拡張され、短い場合は右側が切り捨てられる。

右辺の値は、左辺を定義するのに必要な部分だけが確定であればよい。左辺が部分列の場合には、右辺の値はその部分列にのみ割り当てられ、左辺で指定されていない部分列の定義状態は変わらない。

10. 制御文

10.1 計算形 GO TO文

計算形 GO TO文

$GO\ TO\ (s_1, s_2, \dots, s_n), e$

において、 s_j は文番号またはスイッチ変数、 e は算術式である。ただし、 e の直前のコンマは省略できる。ACOS FORTRAN 66では、 e の直前のコンマは省略できなかった。また、JIS FORTRAN 77では、 s_j は文番号、 e は整数型の算術式となっており、スイッチ変数と整数型以外の算術式は許されていない。

式 e が整数型以外の算術式の場合には、 e の値を評価後、整数化される。式 e の値が1より小さいか、 n より大きい場合、ACOS FORTRAN 66ではエラーとなったが、FORTRAN 77ではCONTINUE文の実行と同様に扱われる。

10.2 割当て形 GO TO文

割当て形 GO TO文

$GO\ TO\ i, (s_1, s_2, \dots, s_n)$ または $GO\ TO\ i$

において、 i はスイッチ変数、 s_j は文番号である。ただし、 i の直後のコンマは省略できる。ACOS FORTRAN 66では、 i の直後のコンマは省略できなかった。計算形GO TO文の場合と異なり、 s_j にスイッチ変数を用いることはできない。

スイッチ変数 i の値に等しい文番号が s_j の中になければ、エラーとなる。

10.3 算術 IF文

算術 IF文

$IF(e) s_1, s_2, s_3$

次のEND IF文が実行される。式 e の値が偽であれば、そのブロックIF文と同じIFの深さをもつ次のELSE IF文、ELSE文またはEND IF文が実行される。

IFブロックの外からIFブロックの内へ制御を移してはならない。

IFブロック内の最後の文の実行が制御の移行を伴わないならば、制御は、そのIFブロックの直前のブロックIF文と同じIFの深さをもつ次のEND IF文に移される。

10.6 ELSE IF文

ELSE IF文は次の形である。

ELSE IF(e) THEN ただし、 e は論理式とする。

ELSE IFブロックは、ELSE IF文の直後から、それと同じIFの深さをもつ次のELSE IF文、ELSE文またはEND IF文の直前までのすべての実行文からなる。ELSE IFブロックは空でもよい。

ELSE IF文の実行によって、 e が評価される。式 e の値が真であれば、そのELSE IFブロックの最初の文が実行される。ELSE IFブロックが空であれば、そのELSE IF文と同じIFの深さをもつ次のEND IF文が実行される。式 e の値が偽であれば、そのELSE IF文と同じIFの深さをもつ次のELSE IF文、ELSE文またはEND IF文が実行される。

ELSE IFブロックの外からELSE IFブロックの内へ制御を移してはならない。ELSE IF文に文番号をつけることはできるが、その文番号はどの文からも参照することはできない。

ELSE IFブロック内の最後の文の実行が制御の移行を伴わないならば、制御は、そのELSE IFブロックの直前のELSE IF文と同じIFの深さをもつ次のEND IF文に移される。

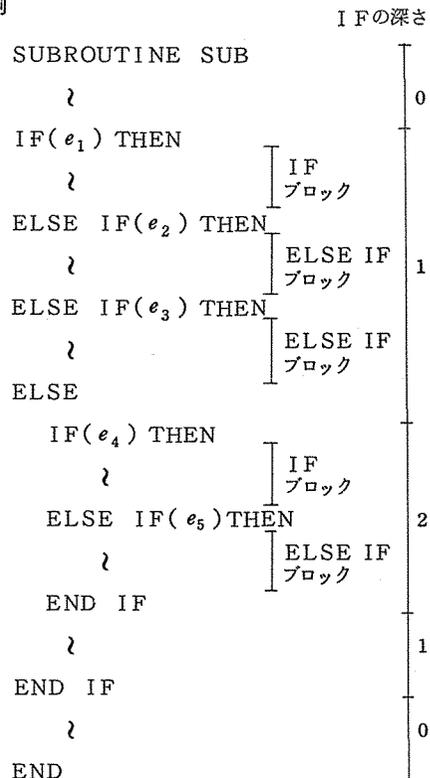
10.7 ELSE文

ELSE文は次の形である。

ELSE

ELSEブロックは、ELSE文の直後から、それと同じIFの深さをもつ次のEND IF文の直前までのすべての実行文からなる。

例



ELSEブロックは空でもよい。

ELSE文と同じIFの深さをもつELSE IF文および他のELSE文は、そのELSEブロックの中に現れてはならない。

ELSE文の実行は、何の効果ももたない。すなわち、CONTINUE文と同様に扱われる。

ELSEブロックの外からELSEブロックの内へ制御を移してはならない。ELSE文に文番号をつけることはできるが、その文番号はどの文からも参照することはできない。

10.8 END IF文

END IF文は次の形である。

```
END IF
```

END IF文の実行は、何の効果ももたない。すなわち、CONTINUE文と同様に扱われる。

各々のブロックIF文に対して、そのブロックIF文と同じIFの深さをもつ次のEND IF文が必ず対応していなければならない。

10.9 DO文とDOループ

10.9.1 DO文とDOループの端末文

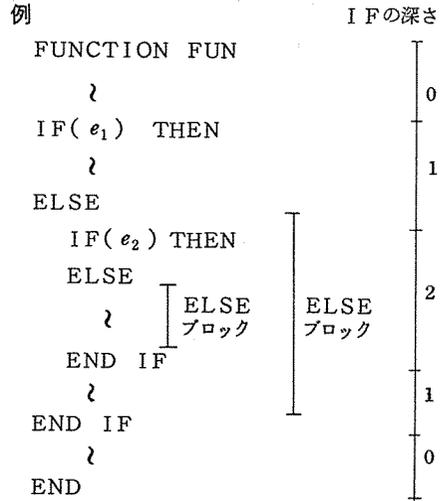
DO文は次の形である。

```
DO s, i = e1, e2, e3
```

ここで、 s は文番号である。 i は整数型、実数型、倍精度実数型または4倍精度実数型の変数名であり、DO変数という。 e_1 、 e_2 および e_3 は算術式である。また、 s と i の間のコンマ、および e_2 の後は省略できる。

ACOS FORTRAN 66では、 s と i の間にコンマはなく、 i の型も整数型に限定されていた。また、 i の名称もDO変数ではなく、制御変数となっていた。JIS FORTRAN 77では、 e_1 、 e_2 および e_3 に複素数関係の型を書くことはできない。

DOループの端末文は、単純GO TO文、割当て形GO TO文、算術IF文、ブロックIF文、ELSE IF文、ELSE文、END IF文、RETURN文、STOP文、END文 およびDO文であってはならない。ただし、ACOS FORTRAN 77では、分岐先を一つでも省略した算術IF文は許している。



10.9.2 DOループの範囲

DOループの範囲は、DO文の直後から、対応する端末文までに現れるすべての実行文からなる。ただし、その端末文は含まれる。

IFブロック、ELSE IFブロックまたはELSEブロック中にDO文がある場合、そのDOループの範囲は、そのブロック中に含まれていなければならない。DOループの範囲内にブロックIF文がある場合、対応するEND IF文もそのDOループの範囲内に含まれていなければならない。

10.9.3 DOループの状態

DOループは、活動状態または休止状態のいずれかとなっている。最初は休止状態であり、DO文が実行されたときにのみ活動状態となる。活動状態のDOループは次のいずれかの場合にのみ休止状態となる。

- (1) 繰返し数(10.9.4参照)がゼロの時。
- (2) DOループの範囲内のRETURN文が実行された時。
- (3) 同一プログラム単位内にあり、かつそのDOループの範囲外にある文に制御が移った時。
- (4) 実行可能プログラム内のSTOP文が実行された時または何らかの理由によって実行が終了した時。

上記の(3)の場合、もう一度DOループを活動状態にするには、DO文を実行しなければならない。すなわち、FORTRAN 66にあったDOの拡張範囲は禁止されている。

DOループが休止状態になった時、そのDOループのDO変数は、最後に定義された値となっている。FORTRAN 66では、DOが満足された時には不定となっていた。

10.9.4 DO文の実行

DO文の実行によって、次の処理が順に実行される。

- (1) 初期値パラメータ m_1 、終値パラメータ m_2 および増分パラメータ m_3 が、それぞれ e_1 、 e_2 および e_3 を評価することによって定められる。そのとき、必要ならば算術変換の規則に従って、DO変数の型に変換される。式 e_3 が省略されている場合には、 m_3 の値は1とする。増分パラメータ m_3 の値はゼロであってはならない。

- (2) DO変数 i に m_1 の値を代入する。

例 DO文 繰返し数

- (3) 繰返し数 l が次式によって定められる。

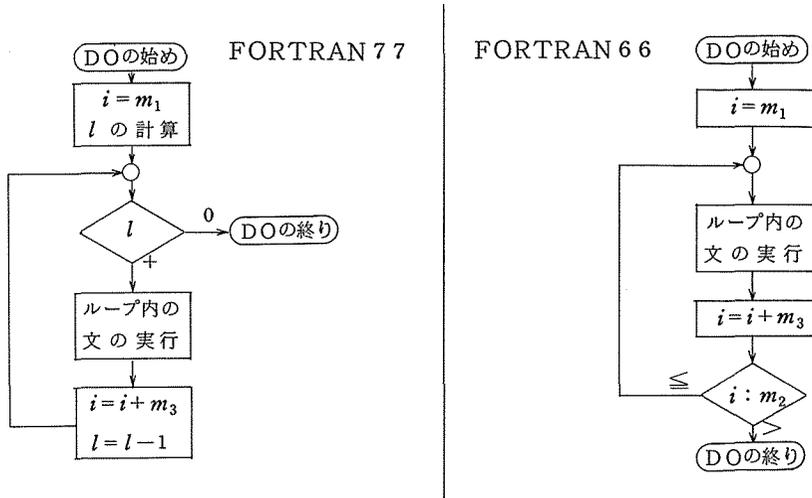
$$l = \text{MAX}(\text{INT}((m_2 - m_1 + m_3) / m_3), 0)$$

各パラメータの値は、 m_3 がゼロであってはならない以外に制限はない。FORTRAN 66では、すべて正でなければならなかった。

DO 10 I=1, 10	10
DO 20 I=10, 1, 2	0
DO 30 I=10, 1, -2	5
DO 40 I=2.1, 5.6, 1.5	4
DO 50 A=2.1, 5.6, 1.5	3

10.9.5 DOループの制御

DOループの動作は、FORTRAN 66と77では異なる。その違いを次の流れ図によって示す。



上図からわかるように、FORTRAN 77ではループ内の文の実行に先立って判定が行われる。したがって、DOループが一度も実行されない場合もある。一方、FORTRAN 66では判定は後で行われるから、DOループは少なくとも一度は実行された。

10.10 STOP文とPAUSE文

STOP n とPAUSE n において、 n は1~5桁の符号なし整数または文字定数、またはそれらの定数名あるいは変数名であり、ACOS FORTRAN 66と77での差異はない。しかし、JIS FORTRAN 77では、定数名と変数名は許されていない。

ACOSではバッチ処理でPAUSE文を使用すると、システムコンソールに表示される。したがって、当センターでは、バッチ処理でのPAUSE文の使用を禁止している。

10.11 END文

END文は制御文であり、当然ながら、実行文である。関数副プログラムまたはサブルーチン副プログラム中では、RETURN文と同じ効果をもつ。主プログラム中では、STOP文と同じ効果をもつ。

END文は、開始行のみに書くことができ、継続行をもつことはできない。END文以外の文の開始行は、END文と同じ形であってはならない。

プログラム単位の最後の行は、END文でなければならない。また、END文は最後の行にのみ現れることができる。

FORTRAN 66では、ENDは行であり、実行文ではなかった。しかし、FORTRAN 77でもEND文は必ずしも実行される必要はなく、実質的な差はない。

10.12 CALL文とRETURN文

CALL文とRETURN文も制御文であるが、関数およびサブルーチンの章で述べる。

(以下次号)