

FORTRAN 言語の構文

現在のプログラム使用している構文について説明する

数値

整数は「1」、「2」など、実数は「1.0」、「2.0」など、指数は「1.0E2」、「2.0E-3」など

演算・代数式 (左辺は変数でなければならない)

A=1.0 : 変数 A に 1 を代入

A=B+C : B と C の和を A に代入

A=B-C : B と C の差を A に代入

A=B*C : B と C の積を A に代入

A=B/C : B と C の商を A に代入

A=B**C : B の C 乗を A に代入

演算の優先順位は累乗、(積・商)、(和・差)であり、記述の左から行われる。また、括弧「()」「」で囲まれた部分が先に演算される

組み込み関数 (「引数」とは関数に渡される値で、数値あるいは変数、または演算式でもよい)

INT(引数) 引数を超えない最大の整数を与える

IFIX(引数) 引数の小数を切り捨てた整数を与える

SQRT(引数) 平方根

SIN(引数) 正弦 (引数はラジアンで与える)

COS(引数) 余弦 (引数はラジアンで与える)

ABS(引数) 絶対値

MOD(L1,L2) 剰余 (L1 を L2 で除した時の余りを与える、引数は共に整数型)

構文

DO 文 繰り返し制御文

DO ステートメント番号 整数型変数=開始値, 最終値 [, ステップ値]

DO 文からステートメント番号の行まで、変数を開始値から最終値までステップ値ごと変えながら繰り返し実行する。ステップ値を省略した場合は1ずつ加算される。ステップ値は負値でもよい (この時は、開始値 < 最終値で指定する)。ステートメント番号の行は実行文でもよいが、通常は「CONTINUE 文」とする。

```

DO 100 I=1,10
  .
  実行文
  .
100 CONTINUE

```

CONTINUE 文 制御文

この文は何も行わないで次の行へ移る。通常は「DO 文」と共に用いる。あるいは、「GOTO 文」の飛び先として使用してもよい。

READ 文 入力文

READ (装置番号 , ステートメント番号) 変数 1 , 変数 2 , . . .

ステートメント番号の行で定義された「FORMAT 文」の形式に従って、装置番号で割り当てられた装置からの入力データを変数に代入する。デフォルトではキーボードからの入力は装置番号「5」に割り当てられている。

```
READ(5,100) A,B
100 FORMAT(書式)
```

ファイルからデータの入力を行う場合には、装置番号「5」以外を指定する。

ステートメント番号の代わりに、FORMAT の書式を直接記述しても良い。また、書式を指定せずに変数への代入もできる（この時は、「*」を指定する。下参照）。

```
READ(5,*) A,B
```

WRITE 文 出力文

WRITE (装置番号 , ステートメント番号) 変数 1 , 変数 2 , . . .

ステートメント番号の行で定義された「FORMAT 文」の形式に従って、装置番号で割り当てられた装置へ変数データを出力する。デフォルトではディスプレイへの出力は装置番号「6」に割り当てられている。

```
READ(5,100) A,B
100 FORMAT(書式)
```

ファイルへデータの出力を行う場合には、装置番号「6」以外を指定する。

ステートメント番号の代わりに、FORMAT の書式を直接記述しても良い。また、書式を指定せずに変数の出力もできる（この時は、「*」を指定する。下参照）。

```
WRITE(6,*) A,B
```

FORMAT 文 入出力の書式指定文（出力形式のみを説明）

ステートメント番号 FORMAT (書式 1 , 書式 2 , . . .)

書式には、変数の書式を定義ものと、制御書式がある。変数書式の数は、入出力を行う変数の数と一致しなければならない。

変数書式

F 形式：実数形式 (Fn.m : 全体 n 桁で小数部 m 桁)

全体桁数には制御桁、符号桁、小数点桁を含むので、 $n > m + 3$

例：F8.3：全体 8 桁で小数部 3 桁 (5.000, -10.000 等と表示)

E 形式：指数形式 (En.m：全体 n 桁で小数部 m 桁)

全体桁数には制御桁，符号桁，小数点桁，指数桁 (2 桁) を含むので， $n > m + 5$

例：E12.5：全体 12 桁で小数部 5 桁 (0.55190E+01, -0.55190E-01 等と表示)

I 形式：整数形式 (In：n 桁)

例：I5：全体 5 桁 (12550, -12345 等と表示)

A 形式：文字列形式 (An：n 桁)

例：A30：全体 30 文字

”(ダブルクォーテーション) 指定：文字列出力

例：”PWM CONVERTER”：ダブルクォーテーション間の文字列を出力
制御書式

空白 (スペース)：nX：n 桁の空白を出力)

例：10X：10 桁の空白

改行：/：スラッシュは改行を出力

```
WRITE(5,100) I,A,B
100 FORMAT("DATA1=",I5,5X,"DATA2=",F6.3/"DATA3=",E15.7/)
```

単純 IF 文 条件 (判別) が真の場合に 1 つの実行を行う

IF (条件) 実行文

条件の書式は以下のとおり (A, B は共に変数, あるいは一方が変数)

A.EQ.B または $A = B$ A と B が等しい

A.GT.B または $A > B$ A が B より大きい

A.GE.B または $A \geq B$ A が B より大きいか等しい

A.LT.B または $A < B$ A が B より小さい

A.LE.B または $A \leq B$ A が B より小さいか等しい

分岐 IF 文 条件 (判別) に応じて複数の実行を行う

IF (条件) THEN ~ ELSE ~ ENDIF

条件が真の時に「THEN」以下，偽の時に「ELSE」以下の実行を行う。

```
IF (条件) THEN
  .
  実行文
  .
ELSE
  .
  実行文
  .
ENDIF
```

```
IF (条件) THEN
  .
  実行文
  .
ENDIF
```

IF 文はネスト (IF 文の中に IF 文を入れること) できる。

GOTO 文 実行を強制的に分岐する
 GOTO ステートメント番号
 (他に, 算術 GOTO 文がある)

CALL 文 サブルーチン呼び出す
 CALL サブルーチン名 (引数 1 , 引数 2 , . . .)
 引数は数値, 変数で指定し, それらの型は定義するサブルーチンの引数の型と一致しなければならない。また, 配列の場合には配列数も一致しなければならない。

END 文 プログラムの終りを定義する

サブルーチン定義

サブルーチンは以下のように記述する。

SUBROUTINE 名前(引数 1 , 引数 2 , . . .) 変数型宣言など 実行文 . RETURN END

サブルーチンプログラムはメインプログラムに続けて記述するか, あるいはサブルーチンプログラムだけを別ファイルで作成して, 「INCLUDE」文で結合する。

その他の構文

OPEN 文 データ入出力用のファイルをオープンする
 OPEN (装置番号, file='ファイル名')
 装置番号は「5」「6」以外を指定し, このファイルに対する入出力は「READ」文「WRITE」文の装置番号で使用する。

CLOSE 文 データ入出力用のファイルをクローズする
 CLOSE (装置番号)

INCLUDE 文 サブルーチンファイルをメインプログラムに結合する
 INCLUDE “ファイル名”

数値計算用行列演算サブルーチン (ユーザ作成)

MMUL1S(HA, HB, HH, N1, N2, N3, ILL)

配列 HA(N1 行 N2 列)と配列 HB(N2 行 N3 列)の積を演算し, 結果を配列 HH(N1 行 N3 列)に返す
ILL はダミー変数

MMUL2S(FA, FB, N, ILL)

配列 FA(N 行 N 列)と配列 FB(N 行 N 列)の積を演算し, 結果を配列 FB(N 行 N 列)に返す
ILL はダミー変数

MADDS(F1, F2, F3, N1, N2, ILL)

配列 F1(N1 行 N2 列)と配列 F2(N1 行 N2 列)の和を演算し, 結果を配列 F3(N1 行 N2 列)に返す
ILL はダミー変数

MSUBS(F1, F2, F3, N1, N2, ILL)

配列 F1(N1 行 N2 列)と配列 F2(N1 行 N2 列)の差 (F1-F2) を演算し, 結果を配列 F3(N1 行 N2 列)に返す
ILL はダミー変数

MINV(H, N)

配列 H(N 行 N 列)の逆行列を演算し, 結果を配列 H に返す

TRANSn(A, T, F)

定係数行列 A(n 行 n 列)に対して, 時間 T における遷移行列演算し, 結果を配列 F(n 行 n 列)に返す