Fortran 90 の紹介

渡部 善隆 *

1997 年 1 月からサービスを開始する新スーパーコンピュータシステム FUJITSU VPP700/56(cf.[9]) の単一 PE(Processing Element) で動作する Fortran は，国際規格 ISO/IEC 1539:1991(JIS 規格 JIS X 3001-1994) に準拠した Fortran 90/VP システムと呼ばれるものです[1] ．

Fortran 90/VP は，旧スーパーコンピュータ VP2600/10 の FORTRAN77 EX/VP の言語仕様を (ごく一部の非互換を除いて) 完全に包含しています．従って，現在 VP2600/10 で使用中の Fortran プログラムはそのまま翻訳可能です．また，配列演算や利用者のデータ型定義，ポインタなど，強力な新機能が多数追加されました．

本稿では，新 JIS 規格となった Fortran 90 の概要を，FORTRAN 77 から追加された機能を中心に紹介します．なお，Fortran 90 のプログラミングについての詳しい解説は [4], [5], [6] などを御覧下さい[2] ．

1 Fortran 90 規格化の背景

FORTRAN 77 規格の改正作業は，FORTRAN 言語を近年の科学技術の諸問題を解決するに足りるプログラミング言語にするべく着手されました．そのために，実際に利用者に提供されている FORTRAN コンパイラはもちろん，Pascal, Ada など他のプログラミング言語の機能も注意深く調査されました[3] ．新しい Fortran の国際標準化規格 (ISO) は 1991 年 8 月に制定され，アメリカの規格 (ANSI) は 1992 年 9 月に，日本工業規格 (JIS) は 1994 年 1 月に制定されました．また，この規格から，従来の大文字の “FORTRAN” から，“Fortran” と記述するようになりました．

Fortran 90 には 2 つの大きな目標があります．

> **目標 1** 従来の FORTRAN 77 プログラムはすべて動くようにする
>
> **目標 2** Fortran 言語を時代とともに進化させる

目標 1 は FORTRAN 77 の言語規格に乗っとって書かれたプログラムの上位互換性を保持するもので，各コンパイラが持つ拡張仕様 (例えば富士通独自の仕様) を全て包含しているわけではありません．しかし，FORTRAN 77 の解説書に記述してある文法を用いる限り，FORTRAN 77 からの移行は心配いりません．

Fortran 90 では，3 章で紹介する多くの新しい機能が追加されました．それとともに，規格では冗長であり不要な機能をリストアップし，「廃止予定事項」として指定しています．また，新しい規格の導入により，新機能に置き換え可能な部分もかなりの数にのぼっています．これら置き換え可能な部分も

*九州大学大型計算機センター・研究開発部　E-mail : watanabe@cc.kyushu-u.ac.jp

[1] 正式名は “FUJITSU UXP/V Fortran90/VP V10L10” です．“V10L10” は “Version 10 Level 10” を意味し，今後バージョンアップ，レベルアップしていく予定です．

[2] この原稿を書いている時点 (1996 年 10 月) で Fortran 90 に関する日本語の解説書は極めて少数です．これは，Fortran 90 の言語仕様をフルサポートしたコンパイラがまだまだ普及していないのが原因だと思います．しかし，FORTRAN 77 が JIS 規格から外れてしまった以上，Fortran 90 の解説本が今後 (数はともかく) 出版されることは間違いないでしょう．

[3] 幾つか文献を読みましたが，C, C++ を「注意深く調査」したとは書いてありませんでした．

将来の規格改正で削除される《かも》知れません[4]．目標 2 は，このように近代的な言語仕様[5]へのゆるやかな移行をねらったものです．

2　センターの Fortran システム

1997 年 1 月からのセンターの Fortran システムは以下のラインナップです[6]．

計算機	VPP700/56	M-1800/20U		S-4/1000E
OS	UXP/V V10	UXP/M V12	MSP EX	SunOS 5.4
ホスト名	kyu-vpp	kyu-cc	kyu-msp	wisdom
IP アドレス	《当面非公開》	133.5.9.1	133.5.9.2	133.5.9.9
浮動小数点形式	IEEE 形式	IBM(M) 形式	IBM(M) 形式	IEEE 形式
Fortran 言語	Fortran 90/VP Fortran 90/VPP	Fortran 90/VP Fortran 90 FORTRAN77 EX/VP FORTRAN77 EX	FORTRAN77 EX	Fortran 90

VPP700/56 での TSS の運用開始は 1997 年 4 月頃を予定しており，それまで IP アドレスは非公開です．VPP700/56 の利用は当面従来の VP2600/10 と同様バッチ処理のみです．表中 “/VP” がつくものは単一 PE でのベクトル化コンパイラを意味します．“/VPP” がつくと複数 PE を用いた並列化 Fortran を，何もつかないものはスカラー版を意味します．

2.1　VPP700/56 の Fortran

新スーパーコンピュータ VPP700/56 システムの Fortran は，単一 PE 上で動作する Fortran 90/VP と，複数 PE 上で並列に動作する Fortran 90/VPP の 2 種類が用意されています．

Fortran 90/VP は FORTRAN77 EX/VP の言語仕様を包含しています．Fortran 90/VPP は，Fortran 90/VP 仕様のソースプログラムに並列化に関する制御行を埋め込み，翻訳・実行します．並列化に関する指示行は，富士通独自の仕様であり Fortran 90 の規格外です．また，他のメーカーの並列化コンパイラとの互換性はありません ([2])[7]．

以降，VPP700/56 の Fortran 90/VP システムについて，JIS 規格からさらに拡張された仕様を FUJITSU で表します．拡張仕様の使用はプログラムの移植性の点からできるだけ避けるべきなのは当然ですが，なかなか便利な機能もあります．拡張仕様をプログラムに組み込む際は，他のコンパイラとの非互換が生じる可能性があることに十分注意願います[8]．

なお，Fortran 90/VPP では，富士通拡張仕様の FORTRAN IV 互換モードは使えません．

2.2　M-1800/20U の Fortran

汎用計算機の Fortran は従来と変わりません．MSP システムの Fortran 90 は未サポートです．また，UXP システムの Fortran 90 は，ソースプログラムを解釈して FORTRAN 77 EX システムに渡

[4] [6] の 11 章には「使わない方がよい機能」として　EQUIVALENCE 文，COMMON 文，BLOCK DATA 文，ENTRY 文，INCLUDE 行，固定プログラム形式などをあげています．ただし，現在これらを全部廃止すると暴動が起きるのは必至です．

[5]「近代的なプログラミング言語とは何か？」については意見が分かれるところです．少なくとも筆者には時代を見据えた大言荘語を言い放つ知識も意志もありません．

[6] VP2600/10 は 1997 年 2 月まで暫定運用となりますが，表からは除外しています．

[7] 複数 PE を用いた並列処理を行う別の手段として，PVM, MPI, PARMACS という，PE のデータ授受を管理するサブルーチン群 (メッセージパッシングライブラリと呼びます) がサポートされます．これらも Fortran 90 の規格外です．

[8] 拡張仕様の中には，すべてのコンパイラで非互換というわけでもなく「業界標準」の仕様もあります ([1] 11.9 節参照)．

す「プリプロセッサ方式」をとっています．このため，変換ができない (つまりサポートできない)Fortran 90 の仕様が一部あります (cf.[3])[9]．また，汎用機の Fortran 90 では，富士通拡張仕様の FORTRAN IV 互換モードは使えません．

2.3 ライブラリ・サーバーの Fortran

ライブラリ・サーバー S-4/1000E(ホスト名 wisdom) には，ワークステーション版の Fortran 90 システムが搭載されています．言語仕様は VPP700/56 の Fortran 90/VP と同じです．ベクトル化に向かないプログラムの実行，言語レベルのデバッグの際に利用下さい．

3 Fortran 90 の新機能

この章では，Fortran 90 で追加された新しい機能について紹介します．

3.1 ソースプログラムの書き方

3.1.1 自由形式

Fortran 90 のプログラム形式は，どこからでも自由に記述できる自由形式です[10]．

FUJITSU ファイルのサフィックスが .f90 であるか，翻訳時に -Free オプションを指定した場合，ソースプログラムは自由形式だとみなされます．これに対し，サフィックスが .f であるか，-Fixed オプションを指定した場合，固定形式だとみなされます． wisdom でも同様です．

自由形式のプログラムは，1 行の長さが 132 文字以内で，1 桁目から 132 桁目のどこからでも書き始めることができます．固定形式の説明は省略します．

3.1.2 継続行

プログラムを継続する場合は，継続する行の最後に & を付加します．行を継続する場合，追加の継続行は最大で 39 行まで可能です．& は継続する行の最初にも付けることができます．特に，文字列を途中で継続する場合は，空白が途中に入らないようにするために，継続行の最初の文字に & を記述する必要があります．

FUJITSU Fortran 90/VP では，継続行は無制限です．アプリケーションの相互乗り入れの時代にともない，*Mathematica* や Maple などの数式処理ソフトの出力 (得てして長いものです) をソースプログラムに貼り付けることも多くなりました．富士通仕様の継続行無制限は，長大な出力リストを抱える利用者には朗報です．ただし，Fortran 90 の仕様でないことにご注意下さい．

[9] 正直なところ，汎用機の Fortran 90 は，新機能の解釈やエラーメッセージの出力など，安定しているとはとても言えません．特にモジュールを使ったプログラムを書く場合は，ライブラリ・サーバー wisdom の Fortran コンパイラとのクロスチェックを強くお勧めします．

[10] 富士通の『FORTRAN77 文法書』には，固定形式を「標準形式」と書いてあります．新しい規格では「標準」が逆転しました．

3.1.3 注釈

行に！を記述することで，それ以下に注釈 (コメント) を書くことができます[11]．また，1 桁目に！または C または ＊があるときは，その行自体が注釈 (注釈行) となります．コンパイラは ！以下を無視しますので，行に対応した注釈を記述しておくと，後で自分すら解読できなくなったり，プログラムを引き継いだ後輩が頭を抱える悲劇が確実に減ります．

なお，ISO 規格では，コメントとして英語以外の各国文字の使用を許しています．従って，漢字を含む日本語のコメントを記述しても文法違反とはなりませんので，安心して日本語のコメントを書いて下さい．ただし，記述している日本語コード (EUC, JIS, SJIS) はしっかり認識しておきましょう．

3.1.4 文の分離

短い文を続けて書く場合，; を文の区切りとして複数の文を 1 行に書くことができます．ただし，調子に乗ってどんどん書くと，読みにくいプログラムになるのでご注意を．

以上をまとめた簡単なプログラム例をあげます．

```
!        1         2         3         4         5         6         7         8
PROGRAM EXAMPLE
 INTEGER I,J,K
 REAL    A(9),B(9),C(2)
 READ(5,*) A,B
  I=1;  J=10;  K=5                                        ! 1 行に複数の命令
  C(1) =  (I+K+J)*39*A(1)*B(2)
  C(2) = 160*A(1)*B(2) + 80*A(3)*B(2) - 120*A(4)*B(2) + & ! 継続行
          20*A(7)*B(2) + 16*A(8)*B(2) - 8*A(9)*B(2)
 WRITE(6,*) 'NEXT ITERATION PARA&
            &METER:',C                                    ! 文字列の継続は空白が
END PROGRAM EXAMPLE                                       ! 入らないように
```

3.1.5 固定形式との互換性

自由形式では，従来の固定形式に比べて格段に自由度が増しました．しかし，複数のコンパイラを使いこなす人にとっては，自由形式と固定形式との互換性を保ちたいところです．そのような方のため，[6] では，次の規則を守ってプログラミングすることを推奨しています．

1. 固定形式でも通用するように，文番号は第 1 ~ 5 文字，文は第 7 ~ 72 文字に記述する．
2. 自由形式でも通用するように，空白を意味のある文字として扱う．
3. 注釈は ！ のみを用いる．また ！ を第 6 文字の位置に置かない[12]．
4. 文を継続するときは，前の行の第 73 文字の位置と継続行の第 6 文字位置に & を書く．

先ほどのプログラムを上の規則で書くと，次のようになります[13]．

[11] TEX の % と同じ機能です．
[12] 固定形式では継続のつもりでも，コメントアウトされてしまうからです．
[13] FORTRAN77 EX では，! を Fortran 90 の先取り機能としてサポートしています．

```
!       1         2         3         4         5         6         7         8
      PROGRAM EXAMPLE
       INTEGER I,J,K
       REAL    A(9),B(9),C(2)
       READ(5,*) A,B
        I=1;  J=10;  K=5
        C(1) =  (I+K+J)*39*A(1)*B(2)
        C(2) = 160*A(1)*B(2) + 80*A(3)*B(2) - 120*A(4)*B(2) +           &
     &          20*A(7)*B(2) + 16*A(8)*B(2) - 8*A(9)*B(2)
         WRITE(6,*) 'NEXT ITERATION PARAMETER:',C         ! 72 文字に収まる
      END PROGRAM EXAMPLE
```

3.2 文字の拡張

名前は 31 文字まで許されます．英小文字も使用可能ですが，文字定数以外は大文字と等価です．名前は，英数字，下線 (_)，数字で構成し，先頭の文字は英字でなければいけません．下線を上手に使えば，そのものズバリの変数名も定義できます．

```
      INTEGER Number_of_Partitions    ! NUMBER_OF_PARTITIONS と等価
      REAL    Width_of_Finite_Element ! WIDTH_OF_FINITE_ELEMENT と等価
```

FUJITSU 英字として $ の使用も許されます．また，水平タブは，固定形式では第 1 桁 ~ 6 桁に指定した場合，水平タブ直後の文字位置を第 7 桁とみなします．固定形式の第 7 桁以降に指定した場合と，プログラムが自由形式の場合は，1 個の空白とみなします．

大文字と等価とはいえ英小文字がサポートされたことで，プログラム中で大文字，小文字をうまく使いこなし，見やすいプログラムを書くことが推奨されるようになりました．

3.3 ファイルからのソースの取り込み

C でよく使われる include 文が Fortran 90 でも使用できます．INCLUDE に続けて，アポストロフィまたは引用符で括りファイル名を指定することでソースの組み込みができます．

例えば，ファイル test.f90 の中身が

```
     REAL    A
     INTEGER I,J
```

だとします．このとき，INCLUDE 行を

```
     INCLUDE 'test.f90'   ! ファイル test.f90 の取り込み
     A = 1.0
     I = 10
     J = 20
```

と書くと，呼び出された箇所で test.f90 の中身が展開され，プログラムとしては

```
REAL    A
INTEGER I,J
A = 1.0
I = 10
J = 20
```

と同じになります．INCLUDE 行は，多数の手続きごとに全く同じ (しかも何行にもわたる) COMMON 文や EQUIVALENCE 文などを記述する羽目に陥った場合に威力を発揮します[14]．

3.4 型宣言文の拡張

従来の型宣言文は，例えば

```
INTEGER   NMAX
PARAMETER(NMAX=1001)
DIMENSION A(NMAX)
INTEGER   L(9),IP(9)
```

などでした．Fortran 90 では，型の宣言と同時に様々な属性をあわせて指定することができます．

```
INTEGER,PARAMETER    :: NMAX=1001   ! 名前つきの定数の属性
REAL,DIMENSION(NMAX) :: A           ! 実数配列の属性
INTEGER,DIMENSION(9) :: L,IP        ! 整数配列の属性
```

変数の前の :: は “具体的な対応関係” の意味です．属性を指定しない場合は省略できます．

また，KIND パラメータを用いて，利用者がデータのタイプを自由に宣言できるようにもなりました．

```
REAL(KIND=16)             :: A       ! 4 倍精度実数型
INTEGER(KIND=8)           :: B       ! 8 バイト整数型
CHARACTER(LEN=10,KIND=1) :: C        ! 文字長 10 の文字型
REAL(KIND=SELECTED_REAL_KIND(6,70)) :: D
                                     ! 実数型で 10 進精度で有効桁 6 桁以上，
                                     ! 指数部が 70 以上
```

[14]ただし，[6] では INCLUDE 行を「構造が不明確」だとして「使わないほうがよい機能」に分類しています．

FUJITSU KIND パラメータに渡す値は処理系に依存します．センターの Fortran 90 では，以下の型が対応します．

宣言	データの型
INTEGER(KIND=1)	1 バイト整数型
INTEGER(KIND=2)	2 バイト整数型
INTEGER(KIND=4)	4 バイト整数型
INTEGER(KIND=8)	8 バイト整数型
REAL(KIND=4)	実数型
REAL(KIND=8)	倍精度実数型
REAL(KIND=16)	4 倍精度実数型
CMPLX(KIND=4)	複素数型
CMPLX(KIND=8)	倍精度複素数型
CMPLX(KIND=16)	4 倍精度複素数型
LOGICAL(KIND=1)	1 バイト論理型
LOGICAL(KIND=4)	4 バイト論理型

3.5 構造型データの定義

利用者が自由に型を定義することもできるようになりました．いくつかのまとまった変数を構造型として宣言することで，構造型をひとつの変数として扱うことができます．

構造型の宣言は TYPE 文で行います．

```
TYPE INTERVAL                              ! 構造型 INTERVAL の定義
   REAL(KIND=8) LOWER,UPPER                ! 区間の上限と下限を設定
END TYPE INTERVAL                          ! 定義の終了

TYPE(INTERVAL) :: A,B(2)                   ! INTERVAL の変数 A,B の宣言

A%LOWER = -1.0D0                           ! 下限の代入
A%UPPER =  4.0D0                           ! 上限の代入
B(1)=A                                     ! 構造型の代入
```

% は「成分演算子」と呼ばれるもので，構造型変数と成分を具体的に結びつける役割を果たします．

FUJITSU 拡張された仕様である STRUCTURE 文による構造型の定義も可能です (cf.[1]) ．

3.6 暗黙の型宣言の抑止

IMPLICIT NONE により，暗黙の型宣言を無効にし，言語のあいまいさをなくすこともできます．暗黙の型宣言は，便利さの反面バグの潜む可能性も増大しますので，新しく Fortran プログラムを書く場合は IMPLICIT NONE を宣言することをお勧めします．

```
      PROGRAM TEST
       IMPLICIT NONE          ! 暗黙の型宣言の抑止
       INTEGER I,J,K          ! 明示的な宣言が必要
       REAL    A(5),B
```

3.7 関係演算子記号

やっと次の演算子記号が使えるようになりました．従来の記号も引続き使えます．

演算子	新機能
.EQ.	==
.NE.	/=
.LT.	<
.LE.	<=
.GT.	>
.GE.	>=

```
      IF (n==i) THEN            ! n=i の場合
       k=k+1
       x(k)=i
      ELSE IF (n>i+j) THEN      ! n>i+j の場合
       k=k+3
       x(k)=i+j
      END IF
```

3.8 配列の要素に対する演算

Fortran 90 では，配列に対する機能も大きく拡張されています．とりわけ，配列要素ごとの演算と代入が定義されたことは大きな拡張点です．たとえば，FORTRAN 77 で

```
      PROGRAM ARRAY
      DOUBLE PRECISION A(100),B(100),C(100)
      INTEGER I
      READ(5,*) B,C

      DO 10 I=1,100
       A(I) = B(I)*2 + SIN(C(I))
10    CONTINUE

      WRITE(6,*) A
      END
```

となるプログラムを，Fortran 90 では次のように書くことができます．

```
      PROGRAM ARRAY
      REAL(KIND=8),DIMENSION(100) :: A,B,C
      READ(5,*) B,C

      A = B*2 + SIN(C)

      WRITE(6,*) A
      END PROGRAM ARRAY
```

ただし，演算は「配列要素ごと」であることにご注意下さい．例えば，2 次元配列 A, B に対し，A*B は要素ごとの積を計算するだけで，行列の意味での積とは異なります．

配列に関しては，配列の値に応じて式の評価および代入を行う WHERE 文もサポートされています．

3.8.1 配列に関する組み込み手続き

新しく規格に盛り込まれた配列に関する組み込み手続で主なものを以下にあげます．汎用性のある処理が手続き化されたことで，プログラミングが (少し) 容易になりました[15]．

手続き名	機能	使用例
DOT_PRODUCT	1 次元配列の内積を計算	s=DOT_PRODUCT(X,Y)
MATMUL	行列積を計算	C=MATMUL(A,B)
MAXVAL	配列要素の最大値を求める	s=MAXVAL(A)
MAXLOC	配列要素の最大値の位置を求める	k=MAXLOC(A)
MINVAL	配列要素の最小値を求める	s=MINVAL(B)
MINLOC	配列要素の最小値の位置を求める	k=MINLOC(B)
PRODUCT	配列要素の積を求める	s=PRODUCT(A)
SUM	配列要素の和を求める	s=SUM(B)
TRANSPOSE	行列を転置する	B=TRANSPOSE(A)

3.9 実行制御の拡張

Fortran 90 と FORTRAN 77 プログラムの見た目の最大の違いは，文番号を指定した CONTINUE 文の代わりに END DO 文で終ることができる点です．例えば，

```
      DO 100 J=N1+1,N
        DO 200 I=1,N
          A(I,J)=A(I,J)-W(J,1)*W(I,2)
200     CONTINUE
100   CONTINUE
```

は

```
      DO J=N1+1,N
        DO I=1,N
          A(I,J)=A(I,J)-W(J,1)*W(I,2)
        END DO
      END DO
```

と書くことができます．

また，IF 文に似た形で，複数の中から一つを選択する CASE 構文もサポートされました．

FUJITSU 拡張仕様では，DO UNTIL 文が使用できます．

[15] 処理系依存のブラックボックスに放り込むことを心配する声もあります．Fortran で同機能のプログラムを組んで，手続きとの性能を比較すると面白いでしょう．

3.10　ポインタ機能

Fortran 90 では，C でおなじみのポインタを使用することもできます．ポインタとは，データを格納している領域の記憶場所に付けられた番号 (アドレス) を代入することができる変数です．

```
REAL,POINTER :: P1,P2   ! ポインタ属性を宣言
REAL,TARGET  :: T       ! 結合する可能性のある実体の宣言

P1 => T                 ! ポインタ代入文による結合
P1 = 1.0                ! 指示先 T に値を代入
```

3.11　入出力文の拡張

ファイルの入出力文も，幾つかの点が拡張されています．例えば，OPEN 文に ACCESS='SEQUENTIAL' および POSITION='APPEND' を指定し，かつファイルが存在している場合，そのファイルにデータを追加することができるようになりました．

```
      OPEN(10,ACCESS='SEQUENTIAL',FILE='out.data')
      WRITE(10,*) A
      CLOSE(10,STATUS='KEEP')

      OPEN(10,ACCESS='SEQUENTIAL',FILE='out.data',POSITION='APPEND')
      WRITE(10,*) B
      CLOSE(10,STATUS='KEEP')
```

3.12　モジュール

データの宣言，構造体・手続きの定義および手続き引用の仕様を持つ新しい単位としてモジュールという副プログラムを定義することができます．利用者はモジュール内で定義した型に対し，*, + などの演算や代入などを再定義することが可能になります[16]．例として，区間演算のモジュールの一部をあげます．もちろん完成のためには和以外の演算も定義する必要があります．

[16]つまり，モジュールをうまく使いこなすことで，Fortran 90 によるオブジェクト指向プログラミングが可能ということになります．しかし，C++ から乗り換える人がいるかどうか...

```
module INTERVAL_ARITHMETIC
 implicit none

type INTERVAL                          ! 型 INTERVAL の宣言
   real(kind=8) LOWER,UPPER
end type INTERVAL

interface assignment(=)                ! 異なる型の代入 (=) を再定義します
 module procedure INTERVAL_FROM_REAL
end interface

interface operator(+)                  ! 異なる型の和 (+) を再定義します
 module procedure ADD_INTERVALS
end interface

contains

 function ADD_INTERVALS(A,B)           ! 和に関する演算の定義です
  type(INTERVAL) ADD_INTERVALS,A,B
      intent(in) A,B
  ADD_INTERVALS%LOWER = A%LOWER + B%LOWER
  ADD_INTERVALS%UPPER = A%UPPER + B%UPPER
 end function ADD_INTERVALS

 subroutine INTERVAL_FROM_REAL(A,B)    ! 倍精度型実数から INTERVAL への代入
   type(INTERVAL) A                    ! を定義します .
   real(kind=8)   B                    ! 引用の仕様は interface assignmen
      intent(out) A                    ! で行います .
       intent(in) B
   A%LOWER = B
   A%UPPER = B
 end subroutine
end module INTERVAL_ARITHMETIC
```

定義したモジュールはメインプログラムで次のように指定します .

```
program interval_test
 use INTERVAL_ARITHMETIC               ! モジュール INTERVAL_ARITHMETIC の呼出
 implicit none
 type(INTERVAL)  A,B,C                 ! 区間の宣言

 A = INTERVAL(cos(2.0D0),cos(1.0D0))
 B%LOWER = 3.14D0
 B%UPPER = 3.15D0

    C = A + B                          ! 区間 + 区間
end program interval_test
```

3.13 その他の新機能

その他，一群の変数をまとめて全体として宣言する NAMELIST 文，手続きの再帰呼び出し，ビット操作，浮動小数点数操作関数，日付，時刻を返すサブルーチンなどが Fortran 90 の新機能としてサポートされました .

3.14　4 倍精度型

FUJITSU Fortran 90 の規格では，『処理系は基本種別よりも高い精度をもつ表現方法を少なくとも一つは用意しなければならない』とあります．従って，Fortran 90/VP の 4 倍精度は (倍精度がありますので) 必ずしも Fortran の規格に沿って存在するものではありません．しかし，VP2600/10 の FORTRAN77 EX/VP と同様，4 倍精度型実数，4 倍精度型複素数，およびそれらの型を引数とする組み込み関数は引続きサポートされますので，御安心を．

3.15　数学関数

FUJITSU 以下の組み込み関数は，Fortran 90 規格から拡張された組み込み関数です．名前は総称名のみをあげています．すべて FORTRAN77 EX からの継続拡張サポートです．

関数名	機能
CBRT(X)	立方根 $x^{1/3}$
EXP2(X)	2^x
EXP10(X)	10^x
LOG2(X)	対数 $\log_2 x$
COTAN(X)	余接 $\cot x$
ERF(X)	誤差関数 $\frac{2}{\sqrt{\pi}}\int_0^x e^{-u^2}\,du$
ERFC(X)	誤差関数 $1-\frac{2}{\sqrt{\pi}}\int_0^x e^{-u^2}\,du$
GAMMA(X)	ガンマ関数 $\int_0^\infty u^{x-1}e^{-u}\,du$
LGAMMA(X)	ガンマ関数 $\log_e\left(\int_0^\infty u^{x-1}e^{-u}\,du\right)$
SIND(X), COSD(X), TAND(X), COTAND(X) ASIND(X), ACOSD(X), ATAND(X), ATAN2D(Y,X)	$\sin x$, $\cos x$, $\tan x$, $\cot x$ の引数 x を度で渡す $\arcsin x$, $\arccos x$, $\arctan x$, $\arctan(y/x)$ の結果を度で返す
SINQ(X), COSQ(X), TANQ(X), COTANQ(X) ASINQ(X), ACOSQ(X), ATANQ(X), ATAN2Q(Y,X)	$\sin x$, $\cos x$, $\tan x$, $\cot x$ の引数 x を象限で渡す $\arcsin x$, $\arccos x$, $\arctan x$, $\arctan(y/x)$ の結果を象限で返す

4　廃止予定事項

最後に，Fortran 90 の廃止予定事項に指定されているものを紹介します．以下の事項に該当するプログラムは，将来仕様から外れる可能性があります．少なくとも，新しく Fortran 90 プログラムを勉強する方は使用しないことをお勧めします．

4.1　算術 IF 文

算術 IF 文は，算術式が負，ゼロ，正のどの値を持っているかに応じて 3 方向に分岐する機能でした．FORTRAN77 規格が制定される以前の本にしばしば登場します．

```
      IF (P-Q) 1,2,3          ! P-Q の値が負，零，正に応じて文番号 1,2,3 に
1     P=0.0                   ! それぞれ分岐します．IF 文および IF 構文で
      GO TO 4                 ! 代替可能です．
2     P=1.0
      GO TO 4
3     Q=-1.0
4     WRITE(6,*) Q
```

4.2　実数型，倍精度実数型の DO 変数

DO ループの制御式に実数型を用いても現在はエラーとなりませんが，予想外に精度が落ちる例が多数報告されており，使用するのは危険です．

```
     REAL A,B
     DO 1 A=1,15.7,2.1       ! DO 変数，反復の限界値，増分値は実数型，倍精度
      B = B + A              ! 型でも可能です．丸め誤差の影響で精度が著しく
1    CONTINUE                ! 損失する可能性があります．
```

4.3　DO 制御式, 共有 DO 端末文

DO ループの終りは，END DO または CONTINUE でなくても文番号の付いた実行文で代替できました．

```
      DO 110 I=1,N           ! 文番号のついた実行文で区切っています．
  110 W(I,1)=A(I,K)          ! END DO, CONTINUE 文に取り替えます．
```

この代替は，カードリーダーでプログラムを読み込む時代には，CONTINUE 文を節約できるため重宝がられました．が，現在ではデバッグや「読みやすさ」の観点からお勧めできません．

同じく，入れ子になった DO ループを同じ文番号の付いた端末文で共有することもできました．

```
      DO 100 K=1,N
       DO 100 J=1,N
        DO 100 I=1,N
         A(I,J)=A(I,J)+B(I,K)*C(K,J)
100   CONTINUE                          ! DO ループを共有
```

これも，文番号と行の節約になるためよく使われています．しかし，Fortran 90 では文番号を指定する必要がなくなりましたので，次のように書き換えた方がいいでしょう[17]．

```
      DO K=1,N
       DO J=1,N
        DO I=1,N
         A(I,J)=A(I,J)+B(I,K)*C(K,J)
        END DO
       END DO
      END DO
```

4.4 その他の廃止予定事項

その他，Fortran 90 の廃止予定事項に指定された機能です．

- ASSIGN 文，割り当て形 GO TO 文，割り当て形 FORMAT 仕様
- IF 構文のブロックの外からその END IF 文への飛び越し
- 選択戻り，選択 RETURN 文
- PAUSE 文
- H 形編集記述子

翻訳オプションとして `-v d9` と指定すると，プログラムが Fortran 90 規格に準拠しているか，また，Fortran 90 規格の廃止事項，廃止予定事項に引っかかっていないかどうか調べてくれます．ライブラリ・サーバー wisdom でも同様です．

```
wisdom% frt -v d9 test.f90  ↵
Fortran 90 diagnostic messages: program name(test)
  jwd3900i-w  ‘‘test.f90’’, line 7: この文は廃止予定事項です.
```

参考文献

[1] UXP/V Fortran 90/VP 使用手引書 V10 用, J2U5-0050, 富士通株式会社 (1996).

[2] UXP/V Fortran 90/VPP 使用手引書 V10 用, J2U5-0080, 富士通株式会社 (1996).

[3] UXP/M Fortran 90 拡張機能使用手引書 V10 用 (暫定版), 富士通株式会社 (1994).

[4] JIS プログラミング言語 Fortran, JIS X3001-1994, 日本規格協会 (1994).

[5] 東田 幸樹, 山本 芳人, 熊沢 友信 : 入門 Fortran 90 実践プログラミング, ソフトバンク (1994).

[6] M. Metcalf, J. Reid (西村 恕彦, 和田 英穂, 西村 和夫, 高田 正之 訳) : 詳解 Fortran 90, bit 別冊, 共立出版 (1993).

[17] DO 制御式と共有 DO 端末文は (ひと昔前の) 教科書や公開されている Fortran プログラムに頻繁に出てきますので，将来仕様から外れてしまっても業界標準としては生き残ると思われます．

[7] UXP/M FORTRAN77 EX 使用手引書 V12 用, 94SP–5010, 富士通株式会社 (1991).

[8] UXP/M FORTRAN77 EX/VP 使用手引書 V12 用, 94SP–5030, 富士通株式会社 (1991).

[9] 島崎 眞昭, 次期スーパーコンピュータシステムについて, 九州大学大型計算機センター広報, Vol.29, No.3, pp.221–222 (1996).