

標準 Fortran コーディングルール

*豊田英司 (気象庁予報部数値予報課)

1. はじめに

数値予報モデルの開発は単に動くものが書ければよいというようなものではなく、高速かつ安定に良い結果を出すための知見を集積する作業が大きな比重を占めます。数値予報モデルの高度化に伴い、情報集積能力はこれまで以上に重要となっていくでしょう。知見の収集という点では、たとえば Linux のようなフリーソフトウェアの成功をみてもわかるように、組織や地理的所在を越えたオープンなユーザコミュニティを形成し、そこから有用な知見を引き出す能力に負うところが大きくなると思われます。

個人や顔の見える範囲のグループだけで利用するソフトウェアを保守管理するときと比べて大きなコミュニティが維持開発に関与するときには、自ずと書法 style に関する要請が異なってきます。他人の作ったソフトウェアを扱う人にとっては、ソフトウェアが一貫した書法や習慣で作られていることが判れば、解読も容易になりますし、変更してできるプログラムとの比較も容易になるでしょう。

このことは本来計算機プログラムに限った話ではなく、自然言語 (日本語や英語) の文書との類推で考えてみれば納得いただけることでしょう。同じ類推から、テクニカルライティングに書法規範や入門書があるように、プログラム言語にも文法書だけでなく書法規範や書法の指針を示す文献があるべきだと結論されます。

プログラム言語の書法規範の一般的な原理を議論する書物はソフトウェア工学ないしは計算機科学の分野が作ってくれています。古典的な地位をもつのは FORTRAN 66 時代の Kernighan and Plauger [1] でしょう。その後の計算機科学の発展の成果はたとえば McConnell の包括的な文献 [2] にまとめられています。あとは、私たちの使う環境に合わせた具体的な規則を作るという仕事が残っているだけなのです。

プログラム言語 C については、さまざまな公開開発プロジェクトが書法ガイドを発表しています。その中でもっとも古典的とされているのは、AT&T Bell Laboratory で使われていた Indian Hill C Style Manual [3] です。

残念ながら、現代の計算機科学者は Fortran を

あまり使わないので、Fortran を使うつもりならば計算機科学者がいいものを作ってくれるのをただ待つというわけにはいきません。気象モデリングに携わる者が自身の Fortran 利用の経験を蓄積して望ましい利用法規範を作っていくしかないのです。実際、英国気象局など欧州の気象機関ではコーディングルールを策定する動き [4] があります。また、オクラホマ大学の ARPS でも Fortran 90 の採用に伴い独自の規則を定めています [5]。

2. 標準 Fortran コーディングルール

日本においても、気象庁予報部数値予報課と気象研究所気候研究部が中心となって標準コーディングルールを作成する活動が始まっています。標準 Fortran コーディングルールは現在 <http://www.mri-jma.go.jp/Dep/fo/mrinpd/coderule.html> で公開されています。以下、2001 年 7 月末現在の内容についていくつかのトピックを紹介しますが、最新状況については随時 Web をご参照ください。

2.1 自由形式

FORTRAN 77 では文番号を 1~5 桁目、継続行の印を 6 桁目、残りの文を 7~72 桁目に入れなければなりません (固定形式) が、Fortran 90 では任意の桁位置を使うことができる自由形式を選択できるようになりましたので、自由形式を推奨しています。継続行の指定法が変わるので移行には少々手間がかかってしまうのですが、簡単なフィルタで書きかえることができます。

2.2 宣言文

FORTRAN 77 では型宣言文と属性宣言文 (PARAMETER 文, SAVE 文など) は別々に指定しましたが、新しく追加された属性をコンマで区切って列挙する書式を推奨しています。

2.3 動的メモリ管理

実行時にならないと大きさが決められない配列を動的に確保する方法がいくつか導入されました。大きな配列をとる場合、スタックの制限が問題となりにくいので ALLOCATABLE 配列を使うことを推奨しています。

2.4 制御構造

FORTRAN 77 にはループ構造として繰り返し数をループに入る前に指定する DO ループしか

く、収束判定などの条件で終了するループを記述するためには GOTO 文を用いる必要がありました。Fortran 90 では EXIT 文 (C の break; に相当) などが追加されたため、GOTO を使わないことを推奨しています。

2.5 モジュール

FORTRAN 77 では分割翻訳の単位は主プログラム、関数、サブルーチン、ブロックデータだけでしたが、Fortran 90 ではモジュールというものが追加されました。モジュールはプログラムの部品同士が共有すべき情報をモジュールの引用という形で指示するための簡便かつポータブルな方法です。

標準コーディングルールではモジュールの使い方を (いまのところ) 定数参照型、変数参照型、パッケージ型 (サブルーチンを提供する) の 3 つの類型に分けて説明しています。現時点ではこの部分の規則が予報モデルだけを想定しているためやや限定的過ぎるくらいがありますが、これから問題を切り分けていこうと考えています。

2.6 時代遅れの機能

新しい機能や書法を提案するということは、時代遅れで避けるべき書法を指定することと表裏一体です。いままでに挙げたほかにも FORMAT 文、ENTRY 文、INCLUDE 行、文関数、選択戻り、H 型編集記述子などが挙げられましよう。

また、処理系固有の拡張もなかなかプログラマの気がつきにくいトラブルの種です。これらについてはなるべく具体例を挙げて説明していくべきだと考えています。

3. 今後の展望

Fortran 言語の現行規格 Fortran 95 [6] で追加された FORALL 文 (HPF にあったものと同じ) や要素別処理関数は自動並列化・自動ベクトル化がやりやすいように設計された構文です。このようなものが一般的になれば、効率向上のための推奨事項は変わってくることでしょう。

現在審議が大詰めの段階にある Fortran 200X (おそらく年内に採決され ISO/IEC 1539-1:2001 となるものと思われる) ではオブジェクト指向構文の追加だけでなく、IEEE 浮動小数点例外のサポート、非同期入出力のサポート、バイト単位のファイルアクセス、(コマンドライン引数の取得などの) 新しい組込手順といった比較的地道な改良も多数導入される模様です。言語の進化の動向を見据えた上で、望ましい書法の指針をとりまとめる努力は当面継続

されなければならないでしょう。

2001 年秋には気象庁非静力学モデルの MPI 並列対応版が公開される予定ですが、ここで紹介するコーディングルールになるべく従った Fortran 90 コードを提供するべく準備をしています。この作業を通じて現実的指針としての側面は充実していくものと思われまます。

標準 Fortran コーディングルールはまだまだ発展途上の段階にありますが、今後はさまざまなユーザの意見を反映させていきたいと考えています。

引用文献

- [1] B. W. Kernighan and P. J. Plauger, 1978: *The Elements of Programming Style, Second Ed.* Bell Labs. 日本語訳は木村泉訳「プログラム書法」共立出版。
- [2] S. McConnell, 1993: *Code Complete.* Microsoft Press. 日本語訳は石川勝訳「コードコンプリート - 完全なプログラミングを目指して」アスキー出版局
- [3] L. W. Cannon et al., *Recommended C Style and Coding Standards.*
<ftp://ftp.cs.washington.edu/pub/cstyle.tar.Z>
日本語訳は豊田他訳「プログラム言語 C の推奨されるスタイルとコーディング規範」
<http://www.gfd-dennou.org/arch/comptech/cstyle/>
- [4] P. Andrews et al., 1994: *European Standards for Writing and Documenting Exchangeable Fortran 90 Code.*
http://www.meto.gov.uk/sec5/NWP/NWP_F90Standards.html
- [5] New ARPS Coding Standards with Fortran 90.
<http://caps.ou.edu/ARPS/coding/>
- [6] ISO/IEC 1539-1:1997. 日本工業規格としては JIS X 0301-1997.