1. What is Fortran?

Fortran is a general purpose programming language, mainly intended for mathematical computations in science applications (e.g. physics). Fortran is an acronym for FORmula TRANslation, and was originally capitalized as FORTRAN. However, following the current trend to only capitalize the first letter in acronyms, we will call it Fortran. Fortran was the first high-level programming language. The work on Fortran started in the 1950's at IBM and there have been many versions since. By convention, a Fortran version is denoted by the last two digits of the year the standard was proposed. Thus we have Fortran 66, Fortran 77 and Fortran 90 (95).

The most common Fortran version today is still Fortran 77, although Fortran 90 is growing in popularity. Fortran 95 is a revised version of Fortran 90 which is expected to be approved by ANSI soon (1996). There are also several versions of Fortran aimed at parallel computers. The most important one is High Performance Fortran (HPF), which is a de-facto standard.

Users should be aware that most Fortran 77 compilers allow a superset of Fortran 77, i.e. they allow non-standard extensions. In this tutorial we will emphasize standard ANSI Fortran 77.

Why learn Fortran?

Fortran is the dominant programming language used in scientific applications. It is therefore important for physics (or engineering) students to be able to read and modify Fortran code. From time to time, so-called experts predict that Fortran will rapidly fade in popularity and soon become extinct. This may actually happen as C (or C++) is rapidly growing in popularity. However, previous predictions of the downfall of Fortran have always been wrong. Fortran is the most enduring computer programming language in history. One of the main reasons Fortran has survived and will survive is *software inertia*. Once a company has spent many people-years and perhaps millions of dollars on a software product, it is unlikely to try to translate the software to a different language. Reliable software translation is a very difficult task and there's 40 years of Fortran code to replace!

Portability

A major advantage Fortran has is that it is standardized by ANSI (American National Standards Institute) and ISO (International Standards Organization). Consequently, if your program is written in ANSI Fortran 77 then it will run on any computer that has a Fortran 77 compiler. Thus, Fortran programs are portable across computer platforms

2. Fortran 77 Basics

A Fortran program is just a sequence of lines of text. The text has to follow a certain *syntax* to be a valid Fortran program. We start by looking at a simple example where we calculate the area of a circle:

```
      program circle
      real r, area
c This program reads a real number r and prints
c the area of a circle with radius r.
      write (*,*) 'Give radius r:'
      read  (*,*) r
      area = 3.14159*r*r
      write (*,*) 'Area = ', area
      stop
      end
```

The lines that begin with a "c" are *comments* and have no purpose other than to make the program more readable for humans. Originally, all Fortran programs had to be written in all upper-case letters. Most people now write lower-case since this is more legible.

Program organization

A Fortran program generally consists of a main program (or driver) and possibly several subprograms (or procedures or subroutines). For now we will assume all the statements are in the main program; subprograms will be treated later. The structure of a main program is:

```
      program name
      declarations
      statements
      stop
      end
```

In this tutorial, words that are in *italics* should not be taken as literal text, but rather as a generic description. The `stop` statement is optional and may seem superfluous since the program will stop when it reaches the end anyway but it is recommended to always terminate a program with the `stop` statement to emphasize that the execution flow stops there.

Column position rules

Fortran 77 is *not* a free-format language, but has a very strict set of rules for how the source code should be formatted. The most important rules are the column position rules:

```
Col. 1    : Blank, or a "c" or "*" for comments
Col. 2-5  : Statement label (optional)
Col. 6    : Continuation of previous line (optional)
Col. 7-72 : Statements
Col. 73-80: Sequence number (optional, rarely used today)
```

Most lines in a Fortran 77 program starts with 6 blanks and ends before column 72, i.e. only the statement field is used. Note that Fortran 90 allows free format.

Comments

A line that begins with the letter "c" or an asterisk in the first column is a comment. Comments may appear anywhere in the program. Well-written comments are crucial to program readability. Commercial Fortran codes often contain about 50% comments. You may also encounter Fortran programs that use the exclamation mark (!) for comments. This is highly non-standard in Fortran 77, but is allowed in Fortran 90. The exclamation mark may appear anywhere on a line (except in positions 2-6).

Continuation

Occasionally, a statement does not fit into one single line. One can then break the statement into two or more lines, and use the continuation mark in position 6. Example:

```
c23456789 (This demonstrates column position!)

c The next statement goes over two physical lines
      area = 3.14159265358979
```