V Rajaraman

This article describes the features of the latest International standard for Fortran known as Fortran 95 which was published in October 1996.

Introduction

The death of Fortran, the oldest high level programming language, has been predicted by many computer scientists but it refuses to die. It is alive and kicking and is steadily improving. The latest version of Fortran, Fortran 95, has been published as an International Standard in October 1996. In fact, Fortran has come a long way since its compiler was written in 1954 by John Backus and his group (Box 1). Fortran 90 and its successor Fortran 95 have introduced revolutionary changes in Fortran (Box 2). It has now absorbed all the good features of programming languages such as Pascal and C and has gone further in providing facilities to compute with arrays and matrices. Fortran 90 is not an evolution but a revolution in the development of Fortran. It has introduced many new features not available in Fortran 77. The most useful of these for numerical computing are:

- Operations on whole arrays which enable Fortran compilers to optimize code generated for pipelined/vector and parallel computers.
- Allocatable arrays which allow good storage management.
- The length of mantissa and the range of exponents of reals may be parametrised by a KIND declaration which allows Fortran 90 code to be machine independent.
- User defined data structures and operators ease programming a variety of applications.
- Pointer datatype allows efficient manipulation of sparse
matrices, lists and graphs.

- Recursive procedures simplify manipulation of recursively defined data structures and writing recursive functions.

We give some code fragments to illustrate the simplicity of dynamic allocation of storage and array assignment in Fortran 90 programs. Comments follow exclamation mark ! in Fortran 90.

```fortran
REAL, DIMENSION (:, :, :), ALLOCATABLE :: D
READ *, N
ALLOCATE D (N, N + 1, 10) ! Allocate storage
A = D (1 : 5, 1 : 6, 1 : 4) ! Assign part of D to A
B = D (6 : 10, 7 : 12, 1 : 4) ! Assign part of D to B
```

Operations on arrays may be performed without using explicit DO loops needed in Fortran 77. For example, we can write:

```
C = A + B
```

where A and B are conformable arrays in Fortran 90. In Fortran 77 this would have been written as:

---

**Box 1**

I don't know what the characteristics of the standard language for scientific and engineering computation in the year 2000 will be ...... but I know it will be called Fortran.

*John Backus*

Designer of the first FORTRAN Compiler

---

**Box 2 History of Fortran**

<table>
<thead>
<tr>
<th>Year</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>1957</td>
<td>FORTRAN for IBM 704 developed.</td>
</tr>
<tr>
<td>1958</td>
<td>FORTRAN II for IBM 704.</td>
</tr>
<tr>
<td>1962</td>
<td>FORTRAN IV for IBM 7030 STRETCH Computer.</td>
</tr>
<tr>
<td>1978</td>
<td>American National Standards Institute (ANSI) FORTRAN 77.</td>
</tr>
<tr>
<td>1996</td>
<td>ISO publishes Fortran 95 standard.</td>
</tr>
<tr>
<td>2001</td>
<td>The next Fortran standards document scheduled to be released.</td>
</tr>
</tbody>
</table>