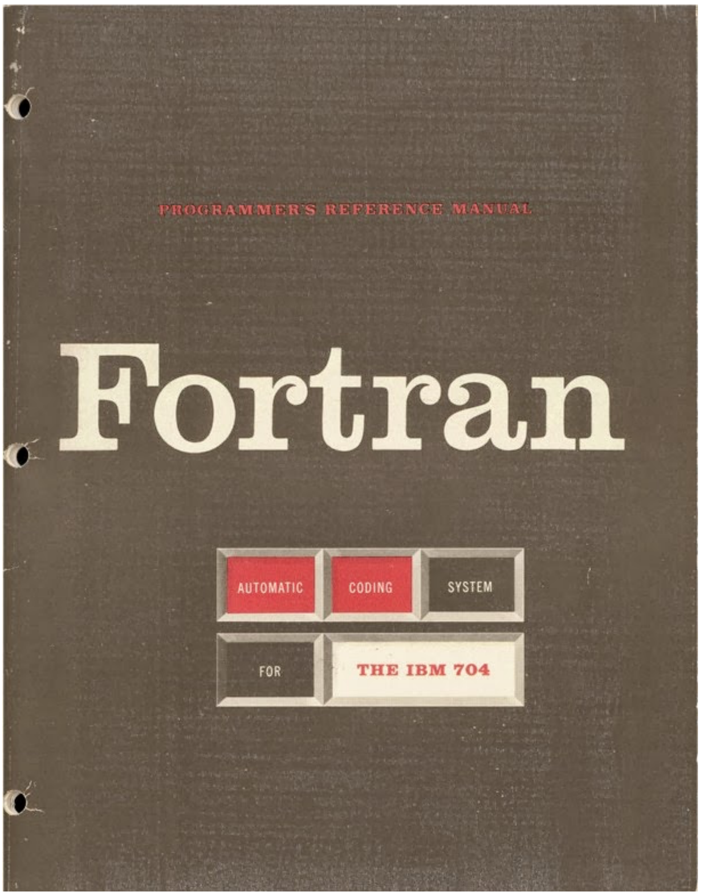
SciHi Blog
daily blog on science, tech & art in history

FORTRAN – The First Programming Language for Numeric Calculations



PROGRAMMER'S REFERENCE MANUAL

Fortran

AUTOMATIC CODING SYSTEM

FOR THE IBM 704

Cover of The Fortran Automatic Coding System, the first book about FORTRAN

On October 15, 1956, the Reference Manual for the Programming Language **FORTRAN** – *The IBM Mathematical Formula Translating System* – was published. It is considered the very first commercially available high-level programming language. FORTRAN was developed at IBM under the guidance of John W. Backus to develop a more practical alternative to assembly language for programming their IBM 704 mainframe computer. FORTRAN became to dominate the area of numerical programming early on and has been in continual use for over half a century in computationally intensive areas such as numerical weather prediction, finite element analysis, computational fluid dynamics, computational physics and computational chemistry.

The Story of FORTRAN

The story of FORTRAN begins in late 1953, when computer scientist John W. Backus submitted a proposal to his superiors at IBM to develop a more practical alternative to assembly language for programming their IBM 704 mainframe computer. If you are not a programmer, you have to know that programming in assembly language means to break an abstract programming problem down into machine instructions. Thus, blowing up simple algorithmic tasks to complex problems, which are rather hard to manage and to maintain. Moreover, assembly language depends always on the computer's architecture and thus, programs written in assembly language won't be able to be shared among several different computers. Therefore, soon the idea of a high-level programming language came up, raising the tedious task of programming to a more abstract level, and enabling programs to be generally portable across multiple architectures. To achieve this, higher-level programming languages must be (automatically) translated into machine instructions with the help of a special program called interpreter or compiler.

"God is Real, unless declared Integer." (J. Allan Toogood, FORTRAN programmer)

The IBM Mathematical Formula Translating System

John W. Backus' historic FORTRAN team consisted of programmers Richard Goldberg, Sheldon F. Best, Harlan Herrick, Peter Sheridan, Roy Nutt, Robert Nelson, Irving Ziller, Lois Haibt, and David Sayre. One of the most important concepts to be realized included easier entry of equations into a computer, an idea developed by J. Halcombe Laning and demonstrated in his GEORGE compiler of 1952. A draft specification for *The IBM Mathematical Formula Translating System* was completed by mid-1954. The first manual for FORTRAN then appeared in October 15, 1956, with the first FORTRAN compiler delivered in April 1957. The main problem of early compilers was efficiency, i.e. the translation process from higher-level programming language into machine instructions usually ended up in sub-optimal machine programs, which compared to hand-coded machine programs were much more complex and also slower in the execution. Therefore, the FORTRAN compiler was realized as an optimizing compiler, because customers were reluctant to use a high-level programming language unless its compiler could generate code whose performance was comparable to that of hand-coded assembly language.

"The determined Real Programmer can write FORTRAN programs in any language." (Ed Post, Real Programmers Don't Use Pascal, 1982)

The Determined Real Programmer

While the community was skeptical that this new method could possibly outperform hand-coding, it reduced the number of programming statements necessary to operate a machine by the incredible factor of 20, and therefore quickly gained acceptance. John Backus said during a 1979 interview with '*Think, the IBM employee magazine*', *“Much of my work has come from being lazy. I didn’t like writing programs, and so, when I was working on the IBM 701, writing programs for computing missile trajectories, I started work on a programming system to make it easier to write programs.*“

In FORTRAN 77 (and most earlier versions), variable names beginning with the letters I–N had a default type of integer, while variables starting with any other letters defaulted to real, although programmers could override the defaults with an explicit declaration. This led to the joke: “*In Fortran, GOD is REAL (unless declared INTEGER).*”

FORTRAN Adoption

FORTRAN was widely adopted by scientists for writing numerically intensive programs, which encouraged compiler writers to produce compilers that could generate faster and more efficient code. The inclusion of a complex number data type in the language made Fortran especially suited to technical applications such as electrical engineering. Significantly, the increasing popularity of FORTRAN spurred competing computer manufacturers to provide FORTRAN compilers for their machines, so that by 1963 more than 40 FORTRAN compilers for various computer architectures existed. For these reasons, FORTRAN is considered to be the first widely used programming language supported across a variety of computer architectures.

As a short example of FORTRAN programming code, here is FORTRAN version of the ‘hello-world’-program:

```
program helloworld
  print *, "Hello, world!"
end program helloworld
```

Also by today, FORTRAN is still one of the most popular languages in the area of high-performance computing and is the language used for programs that benchmark and rank the world’s fastest supercomputers. Also, software for NASA probes Voyager 1 and Voyager 2 was originally written in FORTRAN 5, and later ported to FORTRAN 77. On the other hand, high-level languages such as Matlab, Python, or R have become popular in particular areas of computational science.

Thomas Slawig, *Computer Science in Ocean and Climate Research: Lecture 3: The Fortran Programming Language* [3]

References and Further Reading:

- [1] The History of FORTRAN, in the Computer Museum
- [2] Comprehensive Fortran Standards Documents from the gfortran project
- [3] Thomas Slawig, *Computer Science in Ocean and Climate Research: Lecture 3: The Fortran Programming Language, CAU Kiel, Thomas Slawig @ youtube*
- [4] Backus, John (October–December 1998). “The History of Fortran I, II, and III” (PDF). *IEEE Annals of the History of Computing*. **20** (4): 68–78.
- [5] Ansi x3.9-1978. *American National Standard – Programming Language FORTRAN*. American National Standards Institute.
- [6] Timeline of Programming Languages, via Wikidata