Who is Fortran?

Introduction

We find ourselves in a highly technological environment, where we indirectly depend on coding languages. As of 2025, the top 5 coding languages were JavaScript, TypeScript, HTML, SQL and Python (Universidad ORT Uruguay, 2026). Further development in technology will also lead to development in coding languages and their usage, but even so, there remains certain languages that have been irreplaceable throughout the years, though they've tended to be deemed "unpopular".

Known mostly as a programming language that is extensively used in high-performance computing, Fortran maintains a strong base for scientific programming, forming part in organizations such as weather forecasters, financial trading and in engineering simulations.

Background

Fortran was born in 1954 and became the first computer language standard. Before its creation, computer programming was exclusive to small groups of specialists who wrote code in machine language, making its use almost inconceivable to any outsider. Fortran introduced discrepancy of programming usage, allowing this exclusive realm to open to everybody else, by providing individuals with the ability to input their problems directly into the computer without relying on a programmer to translate their issues into machine code. It became a "translator" that captured "the human intent of a program and recast it in a way that a computer could process" (IBM, n.d.).

Performance

The world's most powerful supercomputer ranking is based on performance in the LINPACK benchmark, a computationally intensive test that measures a computer's floating-point performance by solving dense systems of linear equations. The interesting part is that the LINPACK itself was written in Fortran, meaning that any supercomputer evaluation is essentially tested through highly optimized Fortran routines.

These libraries have become a trusted foundation of numerical linear algebra, meaning that Fortran is not just a language, but also the way we define and measure computational speed at the highest level. This is a deep reminder of how the language remains connected to modern scientific computing.

Decades of scientific software has been written in Fortran, such as global climate modeling, molecular dynamics, nuclear physics and aerospace engineering. For such reasons, hardware manufacturers such as

NVIDIA and AMD (seen in Table 1), specifically design and optimize their HPC compilers to ensure Fortran runs as fast as possible on their newest CPUs and GPUs.

Company	**Country**	**Revenue**
CNRS (1)	France	39.3M
NVIDIA (2)	United States	215.93B
AMD (3)	United States	10.7B
ANSYS (4)	United States	882.2M
ONERA (5)	France	207M

Table 1. Who uses Fortran?

The "*Centre National de la recherche scientifique*" (CNRS), founded in 1939, is a public organization under responsibility of the French Ministry of Education and Research, that includes research in disciplines like biology, engineering, physics, nuclei particles, etc. Fortran is employed in HPC, numerical simulations and scientific modeling in fields like oceanography, particle physics and astrophysics.

Ansys is an analyzing software that provides an understanding of how products will work or not work in real conditions. This is also called engineering simulation software or finite element analysis. Users can use Fortran to allow "user programmable features" in which they can customize the solver behavior, creating tailored subroutines for material models, elements, or boundary conditions. Even so, to fully understand Fortran's importance, we can compare it to one of its most known competitors, with a considerable 35% growth in popularity since 2020 (TIOBE Software, 2026), C++ has been the language closest to Fortran's high performance.

C++

Even though languages like Python or Java often steal the spotlight in modern data science, Fortran must be compared to a language that performs within the scientific computing area. C++, born in the 1980s, has had a constantly growing popularity (currently standing in 3rd place according to TIOBE's March 2026 ranking), most famously for being a "high performance, general purpose" programming language.

Furthermore, some universities still teach Fortran, such as Texas A&M University (TAMU), University of California LA (UCLA), and University of California Berkley (seen in Table 2). TAMU provides this language for HPC and numerical computation, allowing the community to use Fortran for engineering simulations and modeling. The same case is seen in the other two universities, where UCLA dominates HPC for chemistry fields, physics (such as quantum mechanics) and engineering, and University of CA Berkley utilizes Fortran for HPC to maintain and modernize scientific codebases (seen in Table 2).

Country	**University**	**Languages Taught**
CHILE	Univ Chile	Fortran, C, Pascal
USA	Texas A&M (6)	Python, C, C++, Fortran
USA	Purdue University	Fortran, C, C++
USA	Univ. CA Los Angeles (7)	Fortran, C, Pascal
USA	Univ. CA Berkley (8)	Fortran, C, Pascal, LISP
USA	Univ IL Urbana-Champaing	Fortran, C, C++

Table 2. What universities teach Fortran?

Highly known universities keep Fortran within their syllabus, mostly for their STEM programs, even when C++ is the dominant language found in a wide variety of industries such as game development, IoT devices, databases, web browsers, VR applications, etc.

The primary feature to be compared is "complex data type", the ability to handle Composite Data Type. In Fortran the data is intrinsic, meaning the data is built directly into the core of the language which allows the compiler to perform math without you having to write any extra code or include external files. C++ can implement complex data using *typedef* structure, but problems arise when operations such as addition, subtraction, multiplication and division must be performed (the readability of the code is diminished).

A second feature is "data precision". Fortran90 (its most popular version) has a selector that allows the user to specify the desired precision for each variable and function. The Fortran standard also provides double precision (8 bytes) for floating data points, giving long decimal results and providing twice the memory of a "Real" baseline (4 bytes); all of this to ensure accurate calculations. On the other hand, in C++ the data types for whole numbers are short, and for decimals the double precision is applied. This creates a confusing process of conversion between numbers, making the programmer lose control of the conversion, increasing the time of the CPU and causing differentiating levels of precision due to tiny "rounding artifacts" which accumulate over millions of calculations leading to future incorrect final answers.

Conclusion

To answer "who is Fortran" we look into the "why" for the elaboration of this language, the "what" it can be applied on in the modern day, and the "how" it differs from the popularly known C++ language. Fortran continues to excel in precision and efficiency of human computation. The brief comparison between two HPC languages reveals that general-purpose languages are versatile, often sacrificing precise and intrinsic math handling, that Fortran can run by default. Numerical precision and hardware-level optimization is what measures our computational progress, and technological advancement in many areas such as engineering, physics, biology, among others.

Ultimately, Fortran is a tool that has consistently evolved to meet demands of supercomputers. As long as our scientific community continues to push boundaries of simulations and HPC, a place for Fortran will remain, ensuring that calculations are fast and close to exact.

References

- Ansys. (2025, February 20). *Ansys announces Q4 and FY 2024 financial results.* *https://investors.ansys.com/news-releases/news-release-details/ansys-announces-q4-and-fy-2024-financial-results*
- CNRS. (n.d.) *Disciplines.* https://www.cnrs.fr/en/our-research/disciplines
- Codecademy. (2021, February 2). *What is C++ used for?* https://www.codecademy.com/resources/blog/what-is-c-plus-plus-used-for/
- Enlyft. (n.d.). *Companies using Fortran and its marketshare.* *https://enlyft.com/tech/products/fortran*
- IBM. (n.d.) *Fortran.* https://www.ibm.com/history/fortran
- Jansen, P. (2026, March). *TIOBE index for March 2026.* TIOBE. https://www.tiobe.com/tiobe-index/
- Lawrence Berkeley National Laboratory. (2025). *Computer languages & systems software group.* https://go.lbl.gov/class
- Le Sommer, J. (2023, February 15). *Research engineer in machine learning and climate modeling at CNRS.* https://lesommer.github.io/2023/02/15/research-engineer-ml/
- Macrotrends (2024). *AMD revenue 2010-2024/ AMD.* Retrieved from https://www.macrotrends.net/stocks/charts/AMD/amd/revenue
- Macrotrends. (2024) *NVIDIA revenue 2010-2024/ NVDA.* Retrieved from https://www.macrotrends.net/stocks/charts/NVDA/nvidia/revenue
- Prospeo. (n.d.). *CNRS revenue, funding & valuation.* Retrieved from https://prospeo.io/c/cnrs-revenue
- ONERA. (2024) *annual report 2023.* https://onera.fr/sites/default/files/ressources_documentaires/RA-2023-VA.pdf
- Staccone, S. (2025, September 4). Fortran in modern scientific computing: An unexpected comeback. *Medium.* https://medium.com/@stack1/fortran-in-modern-scientific-computing-an-unexpected-comeback-55be2564d22d
- Texas A&M University High Performance Research Computing. (n.d.). *Introduction to Fortran.* https://hprc.tamu.edu/training/intro_fortran.html
- TheirStack. (n.d.). *Companies using Fortran: Tech stack and job listings.* Retrieved from https://theirstack.com/es/technology/fortran
- Universidad ORT Uruguay. (2026, February 17). *Los 10 lenguajes de programación más usados actualmente.* https://fi.ort.edu.uy/blog/los-10-lenguajes-de-programacion-mas-usados-actualmente
- University of California, Los Angeles Institute for Digital Research and Education. (n.d.). *IDRE: Institute for Digital Research and Education.* https://idre.ucla.edu/
- Wharton, L. (1995). *Should C replace FORTRAN as the language of scientific programming?* [Paper presentation]. San Diego State University. https://fweber.sdsu.edu/p317/comparison.pdf

Authors

Maria Isabel Zamudio Reyes

Contact: maria.zamudiors@udlap.mx

Christian Lagarza Cortes

Ph.D. in Engineering from the National Autonomous University of Mexico. Full-time professor in the Department of Industrial and Mechanical Engineering at Universidad de las Américas Puebla.

Contacto: christian.lagarza@udlap.mx