Introducing the Fortran-lang community

Ivan Pribec Vincent Magnin Asdrubal Lozada-Blanco Ondřej Čertik
Jeremie Vandenplas Milan Curcic Rohit Goswami Sebastian Ehlert
Ioannis Nikiteas Gabriele Bellomia Carl Burkert Emanuele Pagone
Brad Richardson Amir Shahmoradi

deRSE23 - 20-22 Feb 2023, Paderborn
Conference for Research Software Engineering in Germany

Overview

Why Fortran?

- 60+ years of high-level, high-performance computing

Fortran-lang community

- State of the Fortran ecosystem
- A new online community for Fortran users
- Fortran-lang open source software

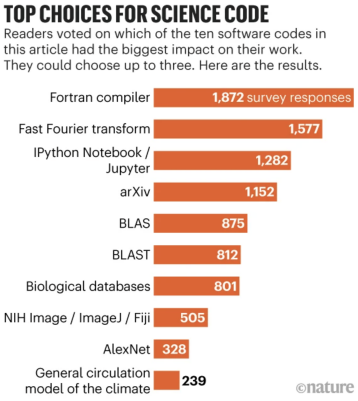
The future is bright

- Current projects

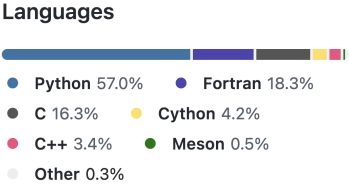
Fortran Summary

- Developed in 1954-1957 at IBM (over 60 years old).
- Name stands for *FOR*mula *TRAN*slation.
- The purpose was (and remains) translation of mathematical formulae to machine code for scientists and engineers.
- First high-level, cross-platform programming language; highly influential.
- Many scientific apps and libraries developed in Fortran.
- Base language has evolved, current standard ISO/IEC 1539-1:2018.
- Supports multiple programming paradigms: procedural, object-oriented, functional, parallel

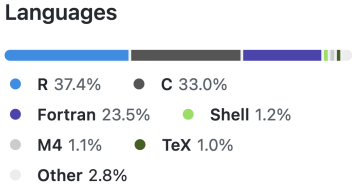
Relevance of Fortran for Science



- Ten computer codes that transformed science, *Nature* 589, 344-348 (2021)



- SciPy (`https://github.com/scipy/scipy`)



- R (`https://github.com/wch/r-source`)

Some wisdoms from the past

> “FORTRAN has for most of [its] life been the blue-collar worker of the programming language set. What it lacked in [*savoir faire*] and style, it returned in cost effectiveness.”
>
> — Martin N. Greenfield

Source: Greenfield, Martin N. “History of FORTRAN standardization.” Proceedings of the June 7-10, 1982, National Computer Conference. 1982.

> “FORTRAN is a language to avoid—unless you want some answers.”
>
> — Anonymous

Source: Mark Jones Lorenzo, The History of the Fortran Programming Language, 2019, SE BOOKS.

Why we use Fortran?

- Performance
- Native *N*-d array syntax including slicing and whole array expressions
- Simplicity (strong, static typing, syntax is easy to understand)
- Backward compatibility longevity
- Parallel

The State of Fortran

Read more about Fortran in our CSE paper:

Kedward, L. J., et al. (2022). The State of Fortran. *CSE*, 24(2), 63-72. `https://doi.org/10.1109/MCSE.2022.3159862`

Fortran Ecosystem

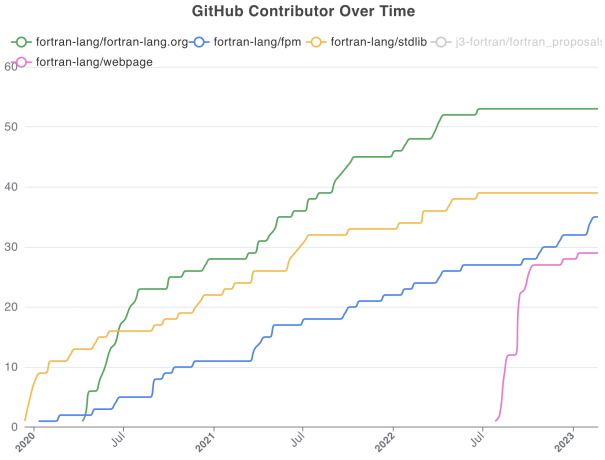
Problem

Stagnation of the Fortran ecosystem across multiple fronts due to lack of organization and communication between isolated communities.

- Lacking general-purpose programming tools (projects tend to re-invent the wheel)
- Building and distributing Fortran software is difficult (no single recommended method)
- No prominent website for new users to discover and learn to use Fortran
- Poor user experience in terms of tooling compared with other languages

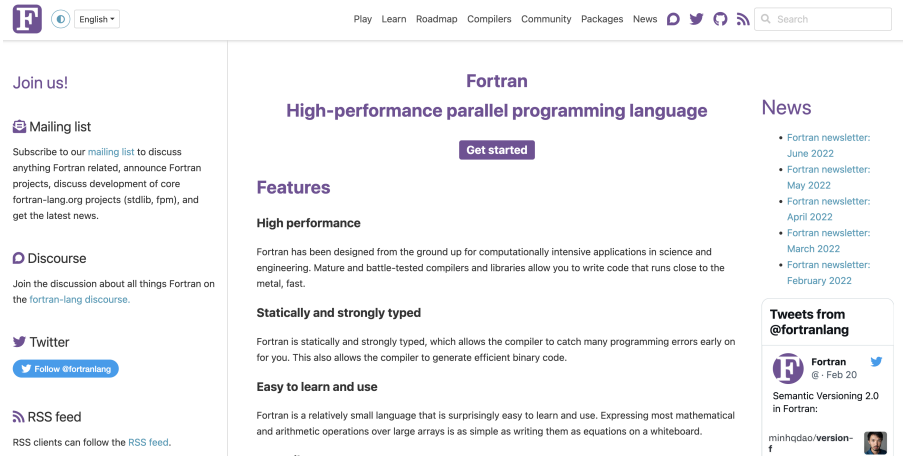
Formation of Fortran-lang

- In 2019 a small group of developers including Ondřej Čertik and Milan Curcic forms in recognition of the shortcomings
- After a few months of discussions the Fortran **`stdlib`** and **Fortran package manager** repositories are born.
- April 2020 the **Fortran-lang website** is launched.
- May 2020 the **Fortran Discourse** is open.
- From here a snowball starts to form...



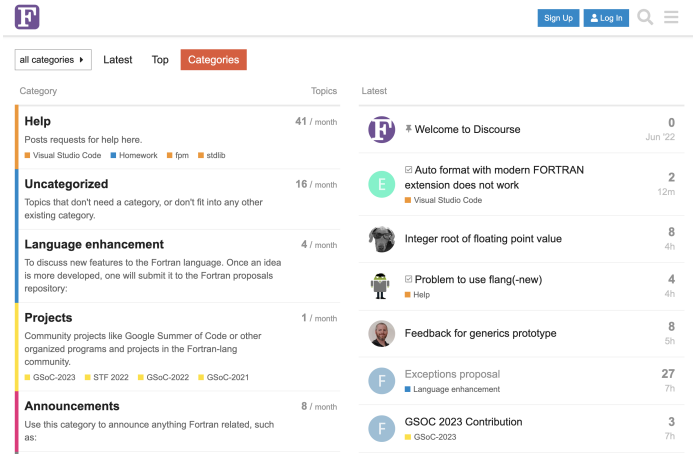
Contributor graph

Fortran-lang – Website



https://fortran-lang.org/en/

Fortran-lang – Discourse



https://fortran-lang.discourse.group/

What about tooling?

Source: https://comic.browserling.com/49

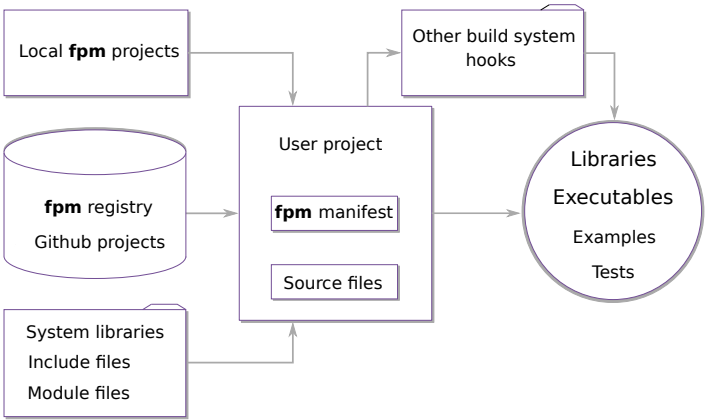
Fortran Package Manager - fpm

Aim

Fortran-specific build system and package manager to ease the learning curve for starting new Fortran projects and composing Fortran software

- GitHub: `https://github.com/fortran-lang/fpm`
- Automatically scans sources for module and submodule dependencies
- Incremental and parallel builds
- Tree-shaking
- fpm is written in Fortran! (Lower the barrier for Fortran users to contribute)

Fortran Package Manager - fpm



Source: Kedward, L. J., et al. (2022). The State of Fortran. *CSE*, 24(2), 63-72. https://doi.org/10.1109/MCSE.2022.3159862

Fortran standard library (stdlib)

Aim

Develop a community-driven *de facto* standard library for Fortran

- GitHub: `https://github.com/fortran-lang/stdlib`
- Specification: `https://stdlib.fortran-lang.org`

Modules

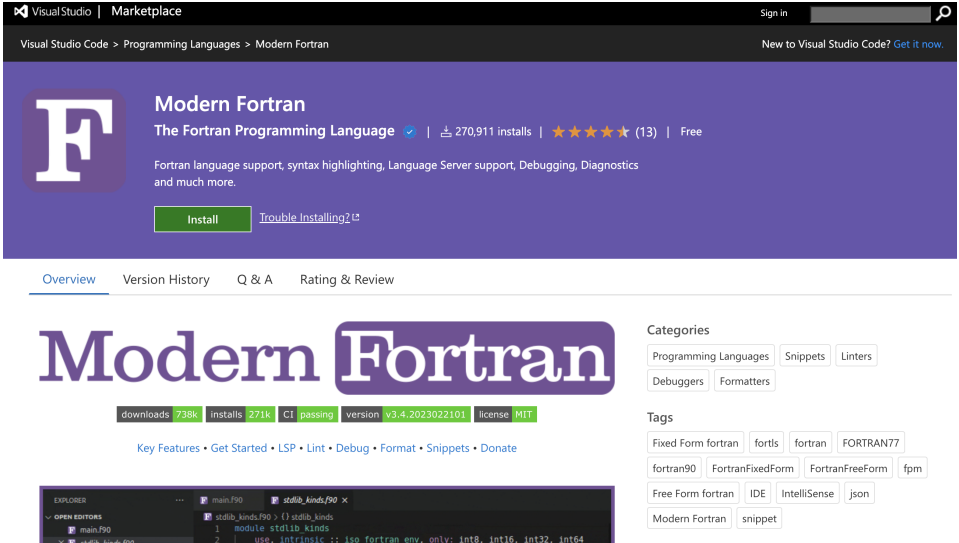
ansi	hashmaps	math	specialfunctions
array	io	optval	stats
ascii	io_npy	quadrature	string_type
bitsets	kinds	random	stringlist_type
error	lingalg	selection	strings
hash_32,64 _bit)	logger	sorting	version

LFortran Compiler

- New LLVM-based Community Compiler
- Also aims to support interactive usage (Jupyter notebooks)
- Multiple backends (LLVM, x86, C, Julia, WASM, ...)
- Modular design centred around two modules—AST and ASR
- Can be used as a library for developing standalone tools
- Used for prototyping newly suggested language features (generics!)

- Website: `https://lfortran.org/`
- Live playground: `https://dev.lfortran.org/`

Fortran support for Visual Studio Code



The Future is Bright

- Google Summer of Code 2023 (hopefully will be accepted for the third year in a row—announcements today!)
- Fortran-lang received funding from the Sovereign Tech Fund (`https://sovereigntechfund.de/fortran.html`)
  - Fpm package registry
  - LFortran compiler
- Growing number of fpm packages
- First “experiments” using fpm in Fortran training courses (Universität Bonn, Forschungszentrum Jülich)

The End... or a New Beginning?

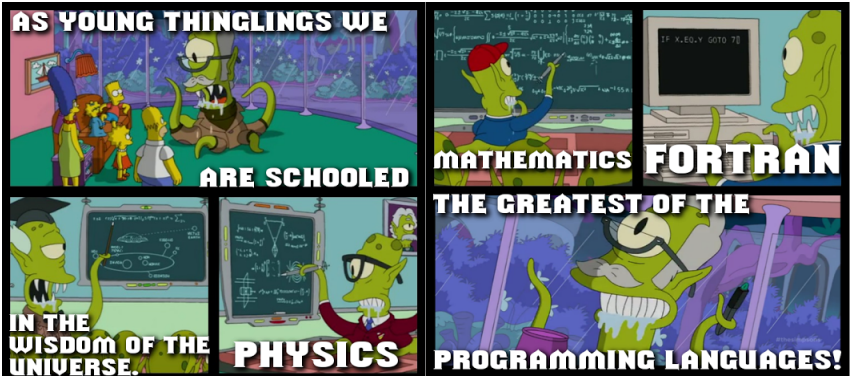


Image from u/crazycold15 (Reddit)
Originally in Simpsons episode “The Man Who Came to Be Dinner” (2015)
https://youtu.be/kMQvC03Wx50?t=80