

# Fortran : retour vers le futur !

Vincent Magnin (MdC)



*Café Calcul CNRS du 7 juin 2024*



# Plan

- 1 La communauté Fortran-lang
  - fpm, le gestionnaire de paquets Fortran
  - stdlib, la bibliothèque standard
  - Le Fortran Discourse
  - Un site central
- 2 Compilateurs et autres outils
- 3 Fortran Moderne
  - Les normes Fortran
  - Normes du 21<sup>e</sup> siècle
- 4 Conclusion

# Diagnostic en 2019 : créons Fortran-lang !

A la traîne par rapport à d'autres langages plus récents :

- une galaxie éclatée :
  - utilisateurs loin du comité de normalisation.
  - Difficile de distribuer des projets Fortran (systèmes de construction multiples, pas de dépôt central).
  - Pas de librairie standard comme en C ou Python.
  - Pas de site web centralisé.
- Pas de compilateur pour expérimenter de nouvelles fonctionnalités.

**Un diagnostic efficient :**

- création de la communauté Fortran-lang fin 2019, en particulier par Ondřej Čertík et Milan Curcic.
- Inclusive : de l'étudiant au membre du comité de normalisation.
- **Un projet pour remédier à chaque point.**

# Contribuer à l'élaboration des normes Fortran

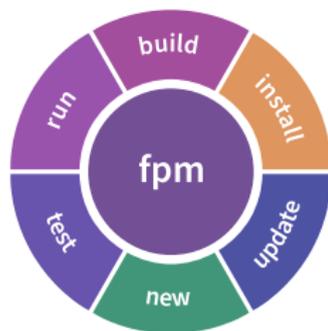
Les normes sont :

- gérées par le groupe de travail ISO/IEC JTC1/SC22/WG5 (Steve Lionel),
- élaborées par le J3 (*US Fortran Programming Language Standards Technical Committee*) : <https://j3-fortran.org/>

En octobre 2019, O. Čertík crée <https://github.com/j3-fortran>

- où chacun peut proposer ou discuter de nouvelles fonctionnalités,
- en particulier pour les génériques prévus pour Fortran 202Y (2028 ?).

# Le gestionnaire de paquets Fortran fpm



- A la fois système de construction et gestionnaire de paquets, inspiré du cargo de Rust.
- Essentiellement écrit en Fortran.
- Licence MIT.
- Version 0.10.1 sortie le 24 mars 2024.
- <https://fpm.fortran-lang.org/fr/>

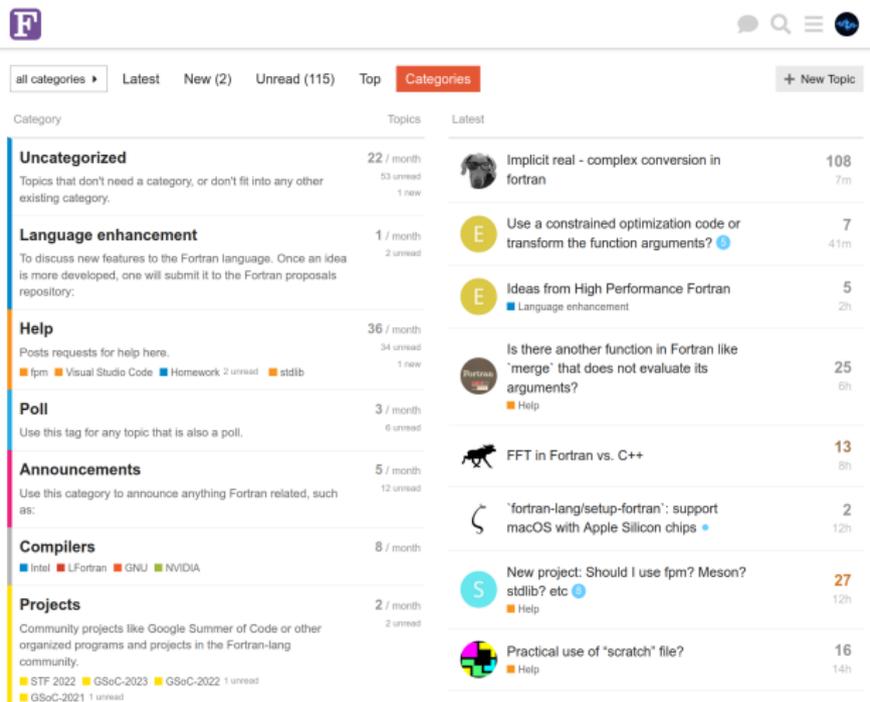
```
-$ ./demo
```

# Une bibliothèque Fortran standard (de facto) : stdlib

- <https://github.com/fortran-lang/stdlib>
- Lancée en déc. 2019 sous licence MIT.
- Version 0.6.1 sortie le 13 mai 2024.
- Documentation : <https://stdlib.fortran-lang.org/>
  - Des centaines de fonctions utilitaires (chaînes de caractères, fichiers, intégration avec le système d'exploitation, tests unitaires, journalisation. . .),
  - des algorithmes divers (recherche, tri, fusion. . .)
  - et des fonctions mathématiques (algèbre linéaire, fonctions spéciales, transformée de Fourier rapide, nombres aléatoires, statistiques, intégration numérique, optimisation. . .).
- Peut être utilisée très simplement avec fpm.

```
~$ ./demo
```

# On discute sur le Fortran Discourse



The screenshot shows the Fortran Discourse website interface. At the top, there is a navigation bar with a search icon, a menu icon, and a profile icon. Below the navigation bar, there is a header with "all categories" and a "Categories" button. The main content area is divided into two columns. The left column lists categories with their respective counts and descriptions. The right column lists topics with their titles, counts, and timestamps.

Category	Topics	Latest
<b>Uncategorized</b> Topics that don't need a category, or don't fit into any other existing category.	22 / month 53 unread 1 new	 Implicit real - complex conversion in fortran 108 7m
<b>Language enhancement</b> To discuss new features to the Fortran language. Once an idea is more developed, one will submit it to the Fortran proposals repository:	1 / month 2 unread	 Use a constrained optimization code or transform the function arguments? 7 41m
<b>Help</b> Posts requests for help here.  fpm  Visual Studio Code  Homework 2 unread  stdlib	36 / month 34 unread 1 new	 Ideas from High Performance Fortran  Language enhancement 5 2h
<b>Poll</b> Use this tag for any topic that is also a poll.	3 / month 6 unread	 Is there another function in Fortran like "merge" that does not evaluate its arguments? 25 6h  Help
<b>Announcements</b> Use this category to announce anything Fortran related, such as:	5 / month 12 unread	 FFT in Fortran vs. C++ 13 8h
<b>Compilers</b>  Intel  LFortran  GNU  NVIDIA	8 / month	 "fortran-lang/setup-fortran": support macOS with Apple Silicon chips 2 12h
<b>Projects</b> Community projects like Google Summer of Code or other organized programs and projects in the Fortran-lang community.  STF 2022  GSoC-2023  GSoC-2022 1 unread  GSoC-2021 1 unread	2 / month 2 unread	 New project: Should I use fpm? Meson? stdlib? etc 27 12h  Help
		 Practical use of "scratch" file? 16 14h  Help

- <https://fortran-lang.discourse.group/>
- Lancé le 5 mai 2020.
- 1543 utilisateurs inscrits (le 13 mai 2024).

## Join us!

### 📧 Mailing list

Subscribe to our [mailing list](#) to discuss anything Fortran related, announce Fortran projects, discuss development of core fortran-lang.org projects (stdlib, fpm), and get the latest news.

### 🗨️ Discourse

Join the discussion about all things Fortran on the [fortran-lang discourse](#).

### 🐦 Twitter

[@fortranlang](#)

### 📡 RSS feed

RSS clients can follow the [RSS feed](#).

### 📄 Open source

Contribute code, report bugs and request features at [GitHub](#).

## Fortran

### Un langage de programmation parallèle haute performance

Débuter en Fortran

## Caractéristiques

### Calcul haute performance

Le langage Fortran a été conçu dès le départ pour des applications de calcul intensif en sciences et ingénierie. Des compilateurs et des bibliothèques matures et éprouvés vous permettent d'écrire rapidement un code exploitant au mieux la puissance de votre matériel.

### Typage statique fort

Le Fortran utilise un typage statique fort, ce qui permet au compilateur de détecter immédiatement de nombreuses erreurs de programmation. Cela lui permet également de générer un code binaire performant.

### Facile à apprendre et à utiliser

Le Fortran a un vocabulaire relativement restreint et est étonnamment facile à apprendre et à utiliser. L'écriture de la plupart des opérations mathématiques et arithmétiques sur des tableaux de grande taille se fait naturellement, comme sur le papier.

### Multiparadigme

Le Fortran vous permet d'écrire votre code dans le style qui correspond le mieux à votre problème : impératif, procédural, matriciel, orienté objet ou fonctionnel.

### Parallèle

Le Fortran est d'office un langage de programmation parallèle utilisant une syntaxe intuitive de type tableau pour communiquer les données entre processeurs. Vous pouvez exécuter quasiment le même code que ce soit sur un seul processeur, sur un système multicœur à mémoire partagée, sur un système HPC à mémoire distribuée, ou un système de type cloud. Les concepts de co-tableaux, équipes, événements et sous-routines collectives vous permettent d'utiliser différents modèles de programmation parallèle adaptés à vos problèmes.

## News

- Fortran newsletter: June 2022
- Fortran newsletter: May 2022
- Fortran newsletter: April 2022
- Fortran newsletter: March 2022
- Fortran newsletter: February 2022

- <https://fortran-lang.org/>
- Partiellement traduit en français.

# Organisation Fortran-lang sur GitHub

 **fortran-lang**

[Overview](#) [Repositories 28](#) [Projects 6](#) [Packages](#) [Teams 21](#) [People 86](#)



## The Fortran Programming Language

577 followers <https://fortran-lang.org> [@fortranlang](#)

[Follow](#)

### Pinned

 **stdlib** Public  
Fortran Standard Library  
Fortran 931 152

 **fpm** Public  
Fortran Package Manager (fpm)  
Fortran 778 89

 **fftpack** Public  
Double precision version of fftpack  
Fortran 62 19

 **vscode-fortran-support** Public  
Fortran language support for Visual Studio Code  
TypeScript 174 27

 **webpage** Public  
New Fortran webpage  
Python 43 36

 **fortls** Public  
fortls - Fortran Language Server  
Python 204 32

### Repositories

Type ▾ Language ▾ Sort ▾

[View as: Public ▾](#)

You are viewing the README and pinned repositories as a public user.

### People



[View all](#)

### Top languages

- Fortran
- JavaScript
- Python
- Shell
- HTML

• <https://github.com/fortran-lang>

# Financement des projets de la communauté

- Fortran-lang et LFortran participent au *Google Summer Of Code* (GSoC) depuis 2021. Des étudiants sont financés pour des projets durant 10 semaines.
- Fin 2022, l'organisme allemand *Sovereign Tech Fund*, dépendant du ministère fédéral de l'Économie et du Climat, a accordé l'équivalent de trois temps pleins pour travailler six mois sur fpm et LFortran.
- Avril 2024 : l'organisation NumFOCUS sponsorise désormais Fortran-lang.

# A propos de quelques compilateurs

- GFortran (GCC) :
  - Logiciel libre (développé par des volontaires), *fork* de G95 (2003).
  - Nécessite la bibliothèque OpenCoarrays pour le calcul parallèle (<http://www.opencoarrays.org/>).
- Intel ifort / ifx (Intel OneAPI) :
  - ont été les premiers à implémenter intégralement Fortran 2018.
  - ifx, basé sur l'infrastructure LLVM, remplace ifort.
  - Gratuits depuis fin 2020.
- Défunts (fin 2022) : Lahey et Absoft.
- En cours de développement :
  - nouveau Flang basé sur LLVM.
  - LFortran : console interactive, plusieurs *backends* (LLVM, C, C++, Julia, WASM et x86-64). En version alpha.

```
~$ ./demo
```

- Autres compilateurs : <https://fortran-lang.org/compilers/>

# Quelques outils populaires dans la communauté

## IDE

- FortLS, "A Language Server for Fortran providing code completion, diagnostics, hovering and more",  
<https://fortls.fortran-lang.org/>
- Pour Visual Studio Code (extension *Modern Fortran*), Atom, Sublime Text, neovim, Vim, EMACS, Kakoune

## Documentation automatique

- FORD : <https://forddocs.readthedocs.io/>

## Préprocesseur Fortran

- fypp : <https://github.com/aradi/fypp>

## Tests unitaires

- test-drive : <https://github.com/fortran-lang/test-drive>

## Installeur Fortran pour Windows

- <https://github.com/LKedward/quickstart-fortran>

## Quatre normes au 20<sup>e</sup> siècle et quatre au 21<sup>e</sup>



Figure – Photo ©Jacob Williams (source : Mastodon).

Norme ISO/IEC 1539-1

<https://www.iso.org/fr/standard/82170.html>

Les « documents d'interprétation » (brouillons des normes) sont sur le site du J3 : chercher des noms contenant 007, comme 24-007.pdf

## Fortran 2003 : norme majeure

- 585 pages : +55 % par rapport à Fortran 95 !
- **Orienté Objet** : se base sur les modules et les types dérivés introduits par Fortran 90.
  - Les méthodes sont des *type-bound procedures*.
  - Attributs `extends` et `abstract`.
- Normalisation de l'**interopérabilité avec le C** : correspondance entre les types de données (*kinds* définis dans le module `iso_c_binding`) et entre les pointeurs C et Fortran.
- Modules intrinsèques `ieee_exceptions`, `ieee_arithmetic` et `ieee_features` (IEEE 754 = ISO/IEC/IEEE 60559 :2011).
- Améliorations des entrées-sorties (transferts asynchrones...).
- Support optionnel des caractères Unicode.
- Accès aux arguments de la ligne de commande et aux variables d'environnement.
- ...

# Orienté objet

Exemple d'objets déclarés dans un module :

```
1  type, abstract :: Polygon
2      complex(dp), allocatable :: vertex(:)
3  contains
4      procedure :: draw => polygon_draw
5      procedure :: print => polygon_print
6  end type Polygon
7
8  type, extends(Polygon) :: Hat_polykite
9  contains
10     procedure :: set => Hat_polykite_set
11     final :: Hat_polykite_clear
12 end type
```

Utilisation de l'objet :

```
1  type(Hat_polykite) :: hat
2  call hat%set(start=cmplx(x+10._dp, y, dp),
   ↪ hx_side=HX_SIDE)
```

# Exemple d'interopérabilité Fortran / C

gtk-fortran [4] offre des interfaces Fortran vers la librairie GTK :  
<https://github.com/vmagnin/gtk-fortran/wiki>

```
void gtk_window_set_title (GtkWindow *window,  
                           const char *title);
```

```
1  subroutine gtk_window_set_title(window, title) bind(c)  
2    use, intrinsic :: iso_c_binding, only: c_ptr, c_char  
3    type(c_ptr), value :: window  
4    character(kind=c_char), dimension(*) :: title  
5  end subroutine
```

```
1  type(c_ptr) :: window  
2  ...  
3  call gtk_window_set_title(window, "Hello GLib & GTK  
   ↪ world!"//c_null_char)
```

```
~$ ./demo
```

- **Calcul parallèle :**

- Les co-arrays, dont la notation est inspirée de la notation tensorielle :
  - `num_images()` et `this_image()` renvoient le nombre d'*images* (copies du programme) et le numéro (co-indice) de l'image courante.
  - On peut synchroniser les images, avec par exemple `sync all`.
- `do concurrent` : indique qu'une boucle est parallélisable.

- **Le module intrinsèque `iso_fortran_env` normalise des *kinds* tels que `real64` ou `int64`.**

- Sous-modules.

- `block...end block` permet de déclarer localement des variables (ou d'importer des modules).

- Nombre maxi de dimensions (*rank*) d'un tableau passe de 7 à 15.

- `execute_command_line()`.

- L'attribut `contiguous` indique au compilateur qu'un tableau reçu en argument d'une routine occupera une zone contiguë de la mémoire.

- Ajout de nombreuses fonctions intrinsèques.

# Fortran 2018

- Nombreuses améliorations :
- **Nouveaux concepts pour le calcul parallèle :**
  - les images peuvent être regroupées en équipes (*teams*) travaillant en parallèle sur des tâches différentes.
  - Les événements (*events*) permettent aux images de communiquer simplement entre elles.
  - Les procédures collectives (*collective subroutines*) permettent d'effectuer d'une façon simple des opérations de réduction.
- Améliorations pour l'interopérabilité avec le C.
- ...

```
↳ $ ./demo
```

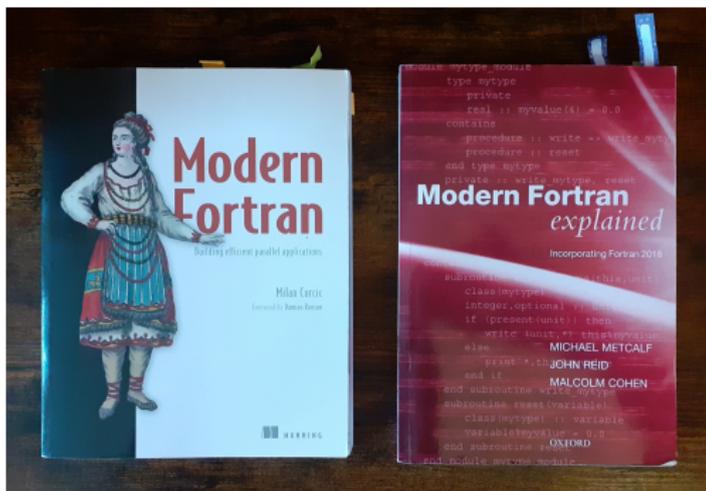
- **Nombreuses améliorations** : Reid, John. « The New Features of Fortran 2023 », 13 mars 2023,  
<https://wg5-fortran.org/N2201-N2250/N2212.pdf>
- **des lignes de code pouvant faire 10000 caractères** et plus de limitation pour leur continuation & (pour la génération automatique de code),
- **boucles do concurrent avec réduction,**
- **les expressions conditionnelles** (même syntaxe qu'en C),
- de nouvelles fonctions pour les chaînes de caractères, en particulier pour interopérabilité C / Fortran,
- l'attribut `simple` permet de définir des fonctions pure qui n'accèdent à aucune donnée extérieure si ce n'est par l'intermédiaire de ses arguments,
- des fonctions trigonométriques en degrés,
- etc.
- **Les éditeurs de compilateurs sont au travail. . .**

# Fortran 202Y (norme majeure pour fin 2028 ?)

- **Au conditionnel** : le travail du J3 peut être suivi sur <https://j3-fortran.org/>
- **programmation générique avec en particulier des modèles (*templates*)** comme en C++. Expérimentations avec LFortran.
- **Préprocesseur Fortran.**
- Définition dans le code des types utilisés par défaut pour chaque type de données.
- Gestion des tâches asynchrones.
- Typage implicite par défaut déclaré obsolète.
- ...

# En conclusion : Fortran... et puis quoi encore ?

- Rejoignez la communauté Fortran-lang.org.
- Essayez ses outils : fpm, stdlib, etc.
- Si vous avez besoin de vous mettre à niveau en Fortran (moderne!) :



Magnin V., Alves J., Arnoud A., Markus A., Marzino M.E., « Fortran... et puis quoi encore ? », Bulletin 1024, Société Informatique de France, n° 22, p. 143-161, novembre 2023, <https://doi.org/10.48556/SIF.1024.22.143>, licence CC BY-NC-ND 4.0. Traduction officielle en anglais "Fortran... ok, and what's next?" : <https://hal.science/hal-04448657> ou <https://arxiv.org/abs/2402.07520>

# ANNEXES

## Quelques liens supplémentaires

- <https://github.com/Beliavsky/Fortran-packages-list>
- <https://github.com/Beliavsky/Fortran-code-on-GitHub>
- <https://github.com/Beliavsky/Fortran-Tools>
- <https://fr.wikipedia.org/wiki/Fortran>
- <https://groups.google.com/g/comp.lang.fortran>
- <https://stevlionel.com/drfortran/>
- <https://fortranwiki.org/>
- Fortran Programmers  
<https://www.linkedin.com/groups/145550/>
- <https://doc.ubuntu-fr.org/fortran>
- Vidéo "Fortran : Array Language" :  
<https://youtu.be/vdaps6Z0kJY>



Mark Jones Lorenzo.

Abstracting Away the Machine : The History of the FORTRAN Programming Language (FORmula TRANslation).

SE Books, August 2019.



John Backus.

A.M. Turing Award 1977 : [https:](https://amturing.acm.org/award_winners/backus_0703524.cfm)

[//amturing.acm.org/award\\_winners/backus\\_0703524.cfm.](https://amturing.acm.org/award_winners/backus_0703524.cfm)



W. P. Heising.

The Emergence of FORTRAN IV from FORTRAN II.

In Annals of the History of Computing, volume 1, pages 28–32,  
January 1984.



V. MAGNIN, J. TAPPIN, J. HUNGER, and J. DELISLE.

gtk-fortran : a GTK+ binding to build Graphical User Interfaces in Fortran.

[The Journal of Open Source Software](#), 5(34), February 2019.

# 70 ans résumés en quelques dates !

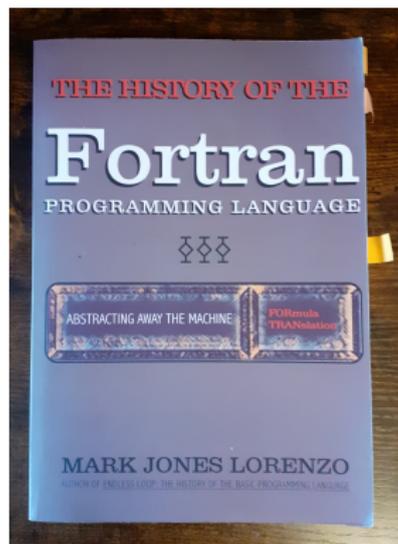


Figure – *The History of the FORTRAN Programming Language - Abstracting Away the Machine*, 2019, ISBN 9781082395949

- Déc. 1953 : John Backus (IBM) propose le projet FORTRAN [1].
- Avril 1957 : déploiement du compilateur chez les clients.
- 1966 : premier langage normalisé.
- 1977 : Prix Turing de Backus [2].
- 1990 : norme Fortran 90.
- 2019 : communauté Fortran-lang.
- Nov. 2023 : 4<sup>e</sup> norme du XXI<sup>e</sup> siècle.

- 1955 : le groupe d'utilisateurs SHARE des machines IBM 704. *Néanmoins, la base technique de FORTRAN était suffisamment solide pour que son utilisation fasse boule de neige. Il y eut bientôt des centaines de clients qui firent des centaines de suggestions d'améliorations. Ils trouvaient des bogues et les envoyaient – pas seulement des rapports d'erreur, mais dans de nombreux cas, les corrections étaient envoyées en même temps. De nombreuses suggestions portaient sur des sujets tels que l'amélioration des diagnostics – de petites choses pratiques – et c'était comme si des centaines de personnes travaillaient à l'amélioration de FORTRAN. Les suggestions affluaient et nous les mettions en œuvre aussi vite que possible. [3]*
- 1970 : le *Fortran Specialist Group* de la *British Computer Society* (<https://fortran.bcs.org/>).
- 1983 : groupe de discussion Usenet `net.lang.f77`, renommé `comp.lang.fortran` en 1986.
- Fin 2019 : la communauté `Fortran-lang`.

# TIOBE Index : mais que se passe-t-il ?

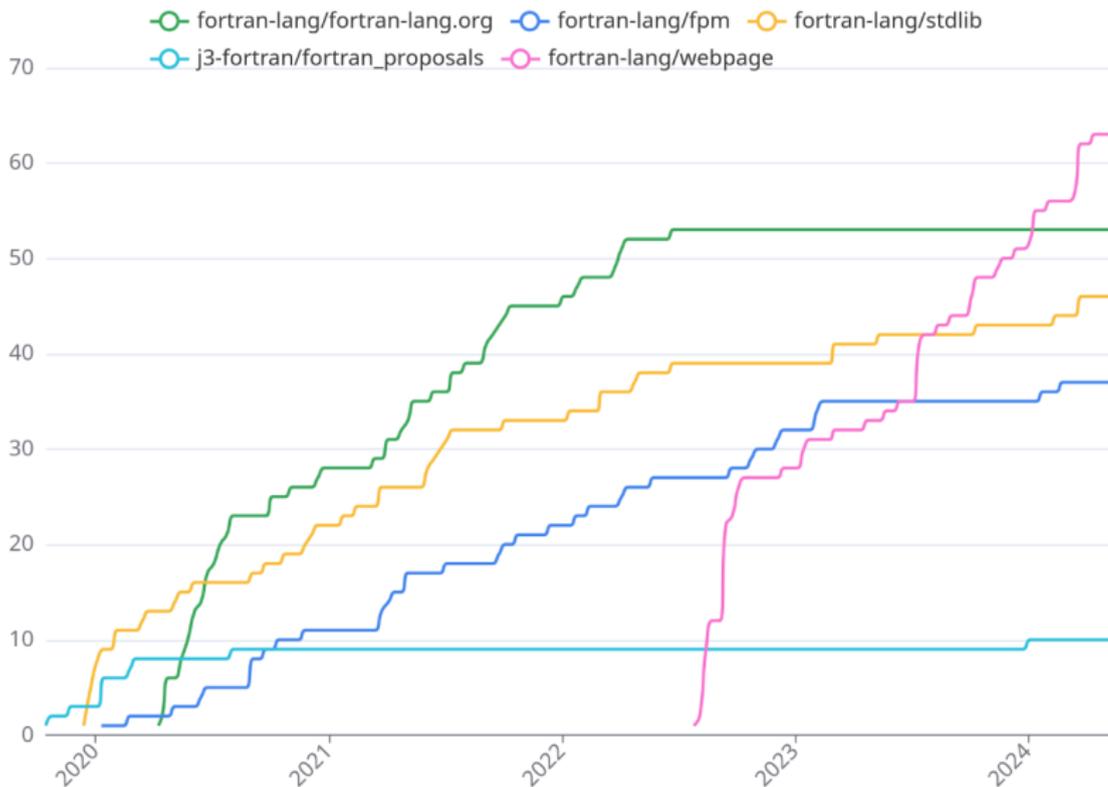
Avril 2021 : Fortran entre dans le top 20. En avril et mai 2024, il est 10<sup>e</sup> :  
"May Headline : Fortran in the top 10, what is going on?"

May 2024	May 2023	Change	Programming Language	Ratings	Change
1	1		 Python	16.33%	+2.88%
2	2		 C	9.98%	-3.37%
3	4	▲	 C++	9.53%	-2.43%
4	3	▼	 Java	8.69%	-3.53%
5	5		 C#	6.49%	-0.94%
6	7	▲	 JavaScript	3.01%	+0.57%
7	6	▼	 Visual Basic	2.01%	-1.83%
8	12	▲▲	 Go	1.60%	+0.61%
9	9		 SQL	1.44%	-0.03%
10	19	▲▲	 Fortran	1.24%	+0.46%
11	11		 Delphi/Object Pascal	1.24%	+0.23%
12	10	▼	 Assembly language	1.07%	-0.13%

Figure – Mai 2024. <https://www.tiobe.com/tiobe-index/>

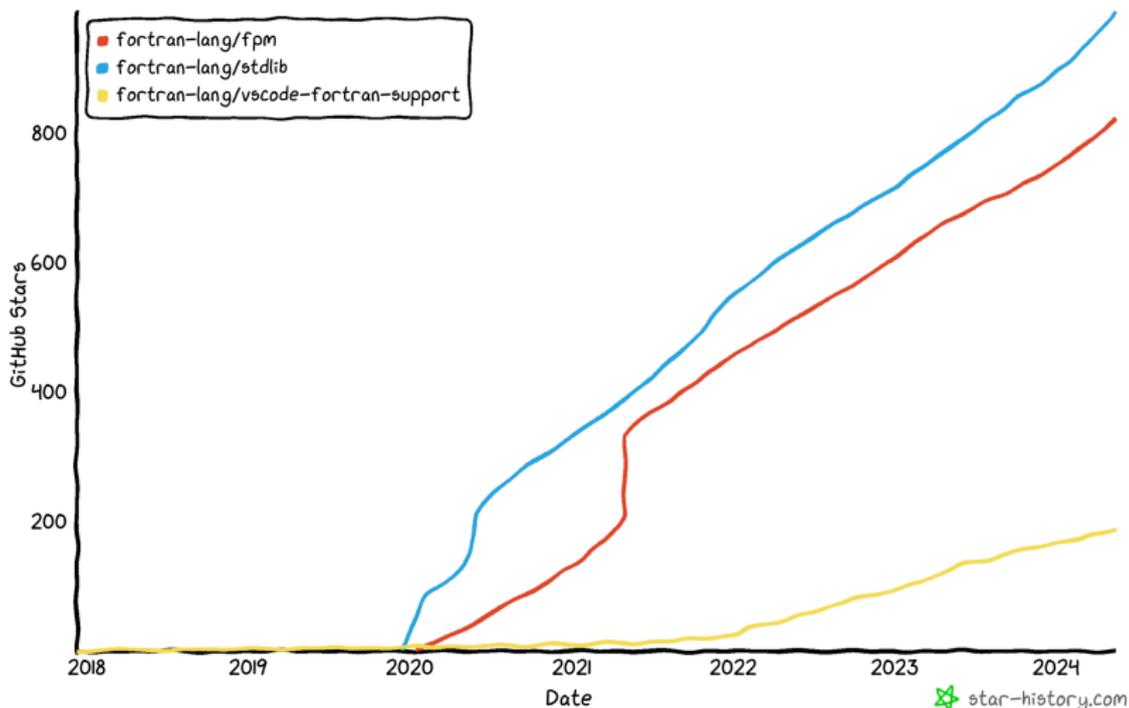
# Contributions aux principaux projets

## GitHub Contributor Over Time



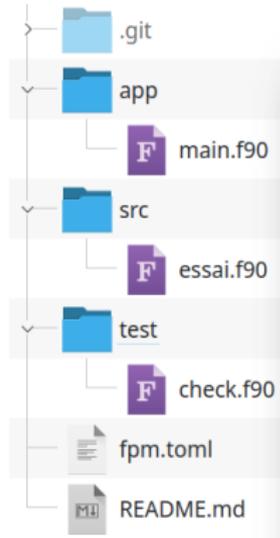
# Popularité

## Star History



# Créer un projet avec fpm

```
$ fpm new essai
$ cd essai
$ fpm build
+ mkdir -p build/dependencies
essai.f90           done.
libessai.a         done.
main.f90           done.
essai              done.
[100%] Project compiled successfully.
$ fpm test
check.f90          done.
check              done.
[100%] Project compiled successfully.
Put some tests in here!
$ fpm run
Project is up to date
Hello, essai
```



# Sous-commandes de fpm

```
$ fpm
```

```
Fortran Package Manager:
```

```
USAGE: fpm [ SUBCOMMAND [SUBCOMMAND_OPTIONS]
```

```
↪ ]|[--list|--help|--version]
```

```
where SUBCOMMAND is commonly new|build|run|test
```

subcommand may be one of

build      Compile the package placing results in the "*build*"

↪ directory

help      Display help

list      Display this list of subcommand descriptions

new      Create a new Fortran package directory with sample files

run      Run the local package application programs

test      Run the test programs

update    Update and manage project dependencies

install   Install project

clean     Delete the build

publish   Publish package to the registry

Enter "*fpm --list*" for a brief list of subcommand options. Enter "*fpm --help*" or "*fpm SUBCOMMAND --help*" for detailed descriptions.

# Le manifeste fpm.toml d'un projet

```
name = "mon_projet_fortran"  
version = "1.0.0"  
license = "GNU GPL"  
author = "vous"  
maintainer = "votre adresse de courriel"  
copyright = "Copyright 2024, vous"
```

## [build]

```
auto-executables = true  
auto-tests = true  
auto-examples = true  
module-naming = false
```

## [install]

```
library = false
```

## [fortran]

```
implicit-typing = false  
implicit-external = false  
source-form = "free"
```

# Fortran 90, norme majeure : Fortran moderne !

- **Langage orienté tableaux.**
  - Allouables : `allocatable`, `allocate()`, `deallocate()`.
  - Procédures `elemental`.
- **Format libre pour le code source** (`.f90` est désormais l'extension à utiliser, au lieu de `.f` pour le format fixe).
- **Concept de kind** pour les types de données.
- `selected_real_kind()`, `selected_int_kind()`
- **Types dérivés.**
- **Modules** : un début de programmation objet...
- **Pointeurs Fortran** : `pointer`, `target`.

# Fortran 95, norme mineure

- **Première approche de la parallélisation :**
  - boucles parallélisables `forall` (obsolètes dans Fortran 2018 !  
Remplacées par `do concurrent`).
  - Attribut `pure` : pas d'effet de bord.
  - A charge au compilateur de paralléliser ou non.
- On peut écrire ses propres fonctions `elemental`.
- ...