RÉPUBLIQUE FRANÇAISE
Liberté
Égalité
Fraternité

METEO FRANCE

Hands-on, fypp and Field API

Judicaël Grasset, MétéoFrance
ACCORD Working Week Oct. 2025, Toulouse

Fypp

Fypp is a Python powered preprocessor for Fortran.

https://github.com/aradi/fypp

Can be a mean to counter the lack of generic in Fortran.

```
interface myfunc
#:for dtype in ['real', 'dreal', 'complex', 'dcomplex']
  module procedure myfunc_${dtype}$
#:endfor
end interface myfunc

character(*), parameter :: comp_date = "${time.strftime('%Y-%m-%d')}$"
```

What is field API ?

Field API is a library developped jointly by Météo-France and ECMWF.

The aim of the library is to ease the porting of the code on GPU by providing a light abstraction over the directives used to transfer the data to and from the GPU and abstract the set of directives used (OpenACC, OpenMP, CUDA).

Made in object-oriented fortran, with a high dose of fypp.

Field API Introduction

There are two fundamentals derived type in field api:

- The field owner
- The field wrapper

The field owner is designed to handle data entirely with field api

The field wrapper is designed to handle data that needs to be encapsulated with field api but that are declared and initialised outside of field api reach.

Field owner creation example

```
USE FIELD_MODULE
USE FIELD_FACTORY_MODULE
CLASS(FIELD_2RB), POINTER :: FO => NULL()
CALL FIELD_NEW(FO, LBOUNDS=[1,1],UBOUNDS=[N,M])
! Use it here
CALL FIELD_DELETE(FO)
```

Field wrapper creation example

```
INTEGER :: MY_DATA(:,:)
ALLOCATE(MY_DATA(N,M)

! Lots of compute code here
!
! Now in another subroutine, that received MY_DATA

USE FIELD_MODULE
USE FIELD_FACTORY_MODULE
CLASS(FIELD_2RB), POINTER :: FW => NULL()
CALL FIELD_NEW(FW, DATA=MY_DATA)
! Use it here
CALL FIELD_DELETE(FO)
```

Field API, usage example

```
CLASS(FIELD_3RB) :: F => NULL()
REAL(KIND=JPRB) :: PTR(:,:,:)
CALL FIELD_NEW(F, UBOUNDS=[NPROMA,NFLEVG,NGPBLKS])
! Get a pointer on host's data (cpu) and do some work with it
CALL F%GET_HOST_DATA_RDWR(PTR)
PTR=A+B
! Get a pointer on device's data (gpu) and work with it on the gpu
! Since we have asked for a read-write pointer on the cpu data before, field api will copy the
data to the gpu before handling a pointer to the user.
CALL F%GET_DEVICE_DATA_RDWR(PTR)
!$ACC KERNELS
DO 1,10
!COMPUTE STUFF WITH PTR
ENDDO
!$ACC END KERNELS
```

Refactoring with field API, YOMRADF

- TRADF type contains a list of allocatable arrays
- We replace them by pointers, and add field API types to hold the data
- Replace the previous init/destroy code by new methods in TRADF type
- Add method to update the view on the blocks