

Multiple data representation with Field API ?

- YDVARs
- Dynamics
- Physics
- A possible implementation ?

YDVARs

```
TYPE FIELD_VARIABLES
```

```
TYPE (VARIABLE_3D) :: U
```

```
TYPE (VARIABLE_3D) :: V
```

```
...
```

```
TYPE (VARIABLE_3D), ALLOCATABLE :: GHG(:)
```

```
...
```

```
TYPE (VARIABLE_3D), ALLOCATABLE :: CHEM(:)
```

Dynamics : loop over 3D fields

```
DO JFLD = 1, SIZE (YDVAR%GFL_PTR)
  IF (YDVAR%GFL_PTR (JFLD)%YCOMP%LCDERS) THEN
    DO JLEV=1,KFLEV
      DO JROF=KST,KEN
        YDVAR%GFL_PTR (JFLD)%YV%DM (JROF,JLEV) = &
          & YDVAR%GFL_PTR (JFLD)%YV%DM (JROF,JLEV) * PGM (JROF)
        YDVAR%GFL_PTR (JFLD)%YV%DL (JROF,JLEV) = &
          & YDVAR%GFL_PTR (JFLD)%YV%DL (JROF,JLEV) * PGM (JROF)
      ENDDO
    ENDDO
  ENDDO
ENDDO
```

Physics : use 4D arrays

```
TYPE (VARIABLE_3D) , ALLOCATABLE :: GHG (:)
```

```
YDVAR%GHG (1) %T0 (: , :)
```

```
YDVAR%GHG (2) %T0 (: , :)
```

```
SUBROUTINE CHEM_MAIN (...)
```

```
...
```

```
REAL (KIND=JPRB) , INTENT (OUT) :: PCHEM2AER (KLON , KLEV , NCHEM2AER)
```

```
REAL (KIND=JPRB) , INTENT (OUT) :: PCHEM2GHG (KLON , KLEV , NCHEM2GHG)
```

A possible implementation ?

- A new class : FIELD_LIST, FIELD_SET, FIELD_GROUP or FIELD_BUFFER ?
- **Inherit** from existing FIELD_RANKSUFF
- Parent object (n dimensions) manages a list of children FIELD objects (with dimensions = n – 1)
- Keep coherency between parent and children

Creation

```
CLASS (FIELD_4RB), POINTER :: YLF4
```

```
CLASS (FIELD_3RB_PTR), ALLOCATABLE :: YLF3 (:)
```

```
REAL(KIND=JPRB) :: RDATA4(NPROMA, NFLEVG, NDIM, NGPBLKS)
```

```
! Create YLF4 & YLF3
```

```
CALL FIELD_NEW (YLF4, CHILDREN=YLF3, DATA=RDATA4)
```

Usage

```
REAL (KIND=JPRB) , POINTER :: ZDATA4 (:, :, :, :)  
REAL (KIND=JPRB) , POINTER :: ZDATA3_2 (:, :, :), ZDATA3 (:, :, :)  
...  
ZDATA4 => GET_HOST_DATA_RDWR (YLF4)  
...  
ZDATA3_2 => GET_DEVICE_DATA_RDONLY (YLF3 (2) %PTR)  
  
!$OMP PARALLEL DO PRIVATE (JBLK, ZDATA3)  
DO JBLK = 1, NGPBLKS  
    ZDATA3 => YLF4%GET_VIEW (JBLK)
```

Hints on implementation

- PARENT member for each field (possibly NULL)
- CHILDREN list in field buffer
- DEVPTR of child = non contiguous slice of DEVPTR of parent (only for the block dimension)
- GET_*_DATA & GET_VIEW of parent update status of children
- GET_*_DATA & GET_VIEW of child update status of parent