

# New fields storage in IFS

Tomas Wilhelmsson

ECMWF

February 2016

## Fields are now passed by argument

- All field related data structures are soon passed by argument instead of through USE statements (Thanks Olivier!)
- But sometimes the same component are accessible from two arguments!
- Here is an example from GP\_MODEL:

```
CALL EC_PHYS_DRV (YDGMV , YDSURF , CLCONF , PSLBUF2 , ... , SD_PF , &
& GMV , GMVS , YDGMV%GMVT1 , YDGMV%GMVT1S , &
& YDGFL%GFLT1 , GFLPT , GFL , GFLSLP , &
& SP_SB , SP_SG , SP_SL , SP_RR , &
& SP_EP , SP_X2 , &
& SD_VF , SD_VN , SD_VD , &
& SD_WS , SD_WW , &
& SD_XA , SD_X2 , &
& SP_OM , SD_V2 , SD_V3 , & !KPP
& ...)
```

- But SP\_SB => YDSURF%SP\_SB etc
- This **violates** the Fortran standard!

## Explicitly passing the surface fields

- In the call from GP\_MODEL to EC\_PHYS\_DRV there are 17 surface arrays.
- SP\_SB has YSP\_SBD for dimensioning and YSP\_SB for indexing *etc*
- So 51 arguments if the surface fields were passed explicitly.
- Here it make more sense to pass just YDSURF!
- But this surface grouping is hardly used anywhere, right?

- Yannick would like a cleaner and more flexible fields structure
- Currently the model state is spread over many arrays and structures:
  1. GMV 3D dynamic fields
  2. GMVS 2D dynamic fields
  3. 9 prognostic SP\_\* surface structures
  4. (And also 19 diagnostic SD\_\* structures)
- All use arrays with an extra *number of fields* dimension.

While keeping the look and feel of YGFL intact:

- Remove the GMV data arrays and make each field an allocatable component of `TYPE_GFL_COMP`.
- Instead of “field pointers” `MP`, `MP9`, *etc* there could be arrays `YP`, `YP9`
- The individual fields can now be allocated as required, instead of in one go.
- The could even be allocated by the ATLAS library!

- Include also surface and other 2D fields into the new GFL.
- With individually allocated arrays, different number of dimensions can now be accommodated.

- Fold GMV into new GFL.
- Fold GMVS into new GFL.
  
- All fields state would then be in one structure.

1. Replace GFL data array with new component of YGFL:
  - Initially, let the proposed new ALLOCATABLE array component of YGFL instead be a POINTER into the old GFL array.
  - Incrementally replace uses of GFL with YGFL pointer component.
  - Restore ALLOCATABLE attributes and remove GFL array.
2. Similarly, incrementally replace surface fields.
3. Add GMV field components into new GFL. As above, incrementally replace use of GMV.
4. Finally also replace use of GMVS.



## Problems to be solved

- How to neatly strip last NBLOCKS array dimension in calls from OpenMP loops.
- Need an iterator over all fields in YGFL.
- What is the best way to handle time levels. There are both permanent levels  $t_{-1}$ ,  $t_0$  and transient  $t_{+1}$ .
- It will be a lot of work, even though many edits should be scriptable.
- What else?
- Please come forward with your ideas and concerns!