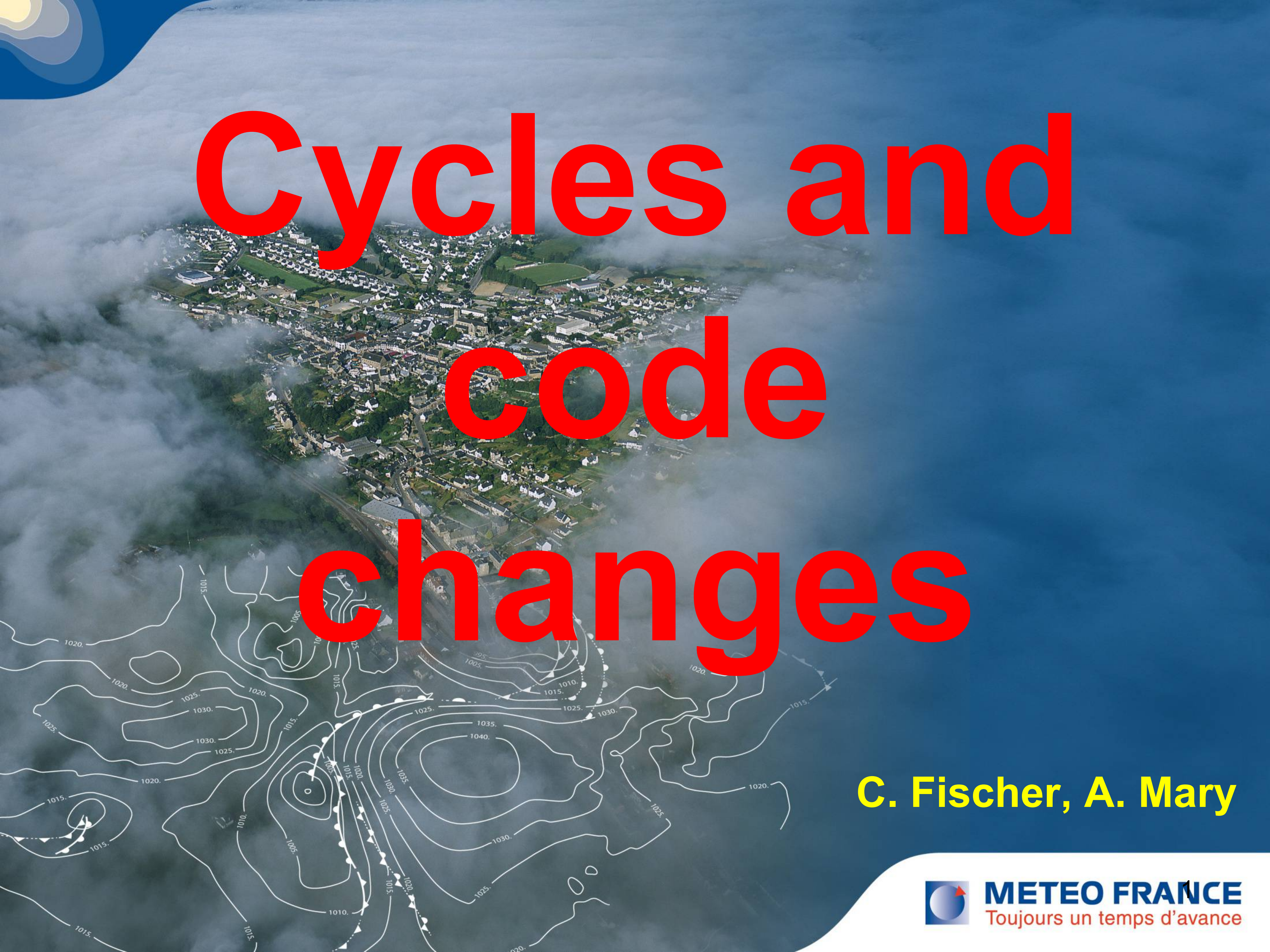


Cycles and code changes

C. Fischer, A. Mary

Content

- Overview of IFS/Arpège/LAM R&D cycles
- IFS re-factoring for OOPS
- OOPS, other stuff and outlook into the future

IFS/Arpège/LAM cycles

CY43 : released on 25 February 2016

- Observation operator re-factoring ; SPAMing of Geometry variables (for OOPS)
- Scientific changes from IFS CY42R1, from Arpège CY41T1_op1

CY43T1 : mid-April – June 2016 ; provisional list tbc

- A number of system contributions by GMAP and some Hirlam
- New convection and Surfex in Arpège ; observation changes for assimilation
- ORORAD (GMAP+Hirlam) ; cloud overlap calculation (Hirlam)
- Assimilation methods : global wavelet Jb, ensemble Jb aspects (GMAP) ; B-normalized perturbations, LAM 4D-VAR fixes (Hirlam)
- Dynamics options (Hirlam)
- RACMO scheme for Harmonie ; radiation changes (in IFS) ; CA updates (Hirlam)

CY44 : Sept – Oct 2016 (dates tbc)

- More code re-factoring for OOPS (new GMV/GFL layout and data structure) ; SPAMing Model variables
- Pruning of dynamics options (GMAP)
- Scientific changes from CY43T1 and CY43R1/R2

Code changing aspects

* **Encapsulation and SPAMing of variables** :

Goal : enable multiple instantiation of objects

=> remove global variables, remove USE MODULE, use derived types and passing by arguments instead.

=> GEOMETRY in CY43 (not totally finished), MODEL in CY44

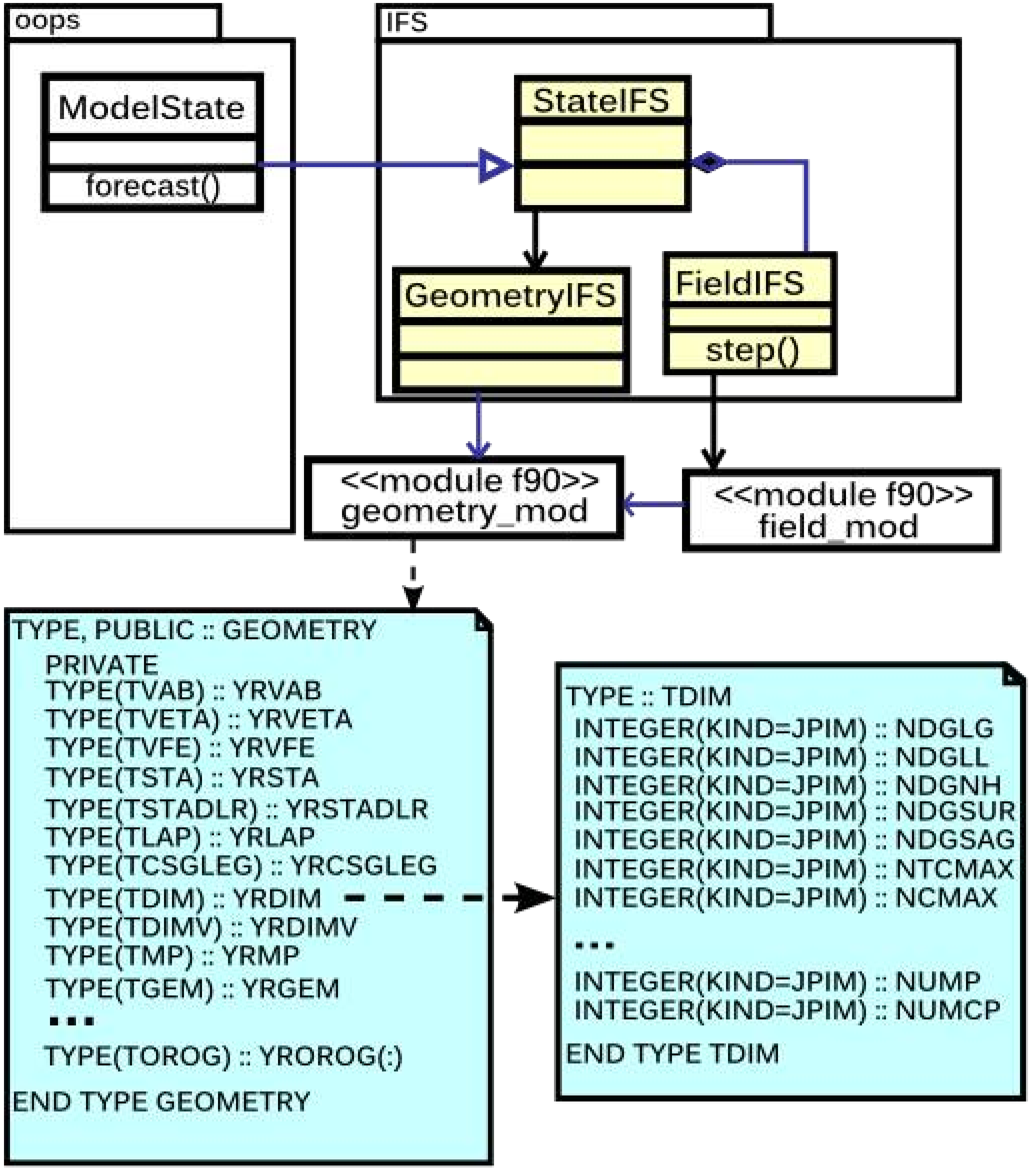
* **Observation operator code** : new interface from IFS to ODB, new GOM_PLUS, new HOP, etc.

Encapsulation of variables in MODULEs

- Group all variables in a MODULE into one derived type
- Pass this derived type as argument in calling seq:

  STEPO_OOPS(FIELDS FLD) won’t use the global variable NPROMA, but instead:

  FLD%GEOM%YRDIM%NPROMA



```
TYPE, PUBLIC :: GEOMETRY
   PRIVATE
   TYPE(TVAB) :: YRVAB
   TYPE(TVETA) :: YRVETA
   TYPE(TVFE) :: YRVFE
   TYPE(TSTA) :: YRSTA
   TYPE(TSTADLR) :: YRSTADLR
   TYPE(TLAP) :: YRLAP
   TYPE(TCSGLEG) :: YRCSGLEG
   TYPE(TDIM) :: YRDIM
   TYPE(TDIMV) :: YRDIMV
   TYPE(TMP) :: YRMP
   TYPE(TGEM) :: YRGEM
   ...
   TYPE(TOROG) :: YROROG(:)
END TYPE GEOMETRY
```

```
TYPE :: TDIM
 INTEGER(KIND=JPIM) :: NDGLG
 INTEGER(KIND=JPIM) :: NDGLL
 INTEGER(KIND=JPIM) :: NDGNH
 INTEGER(KIND=JPIM) :: NDGSUR
 INTEGER(KIND=JPIM) :: NDGSAG
 INTEGER(KIND=JPIM) :: NTCMAX
 INTEGER(KIND=JPIM) :: NCMAX
 ...
 INTEGER(KIND=JPIM) :: NUMP
 INTEGER(KIND=JPIM) :: NUMCP
END TYPE TDIM
```

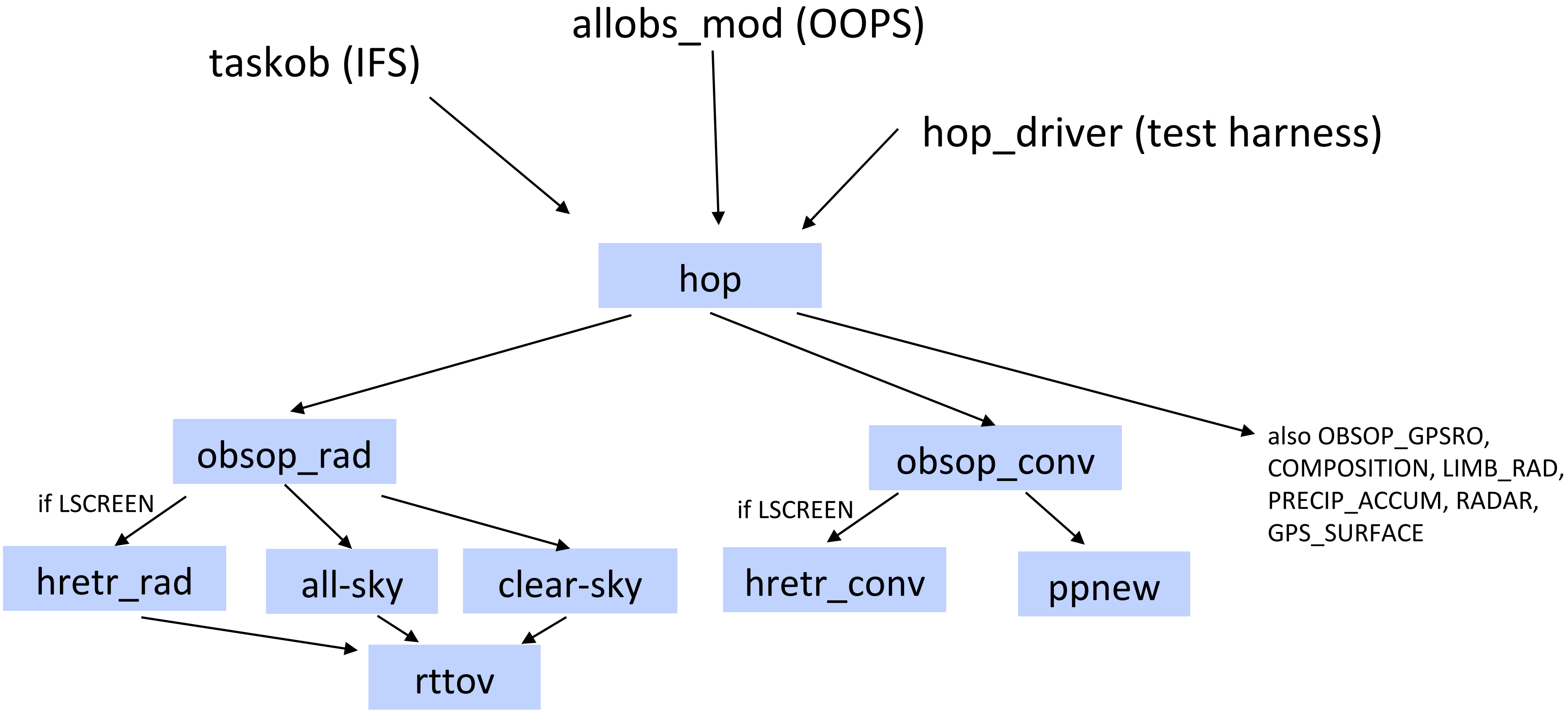
Observation operator cleaning

- GOM_PLUS is a new module/derived type for model quantities in observation space which:
  - encapsulates all model-related code currently called from the observation operator (e.g. gphpre)
  - makes it easy to access model fields in the code and pass from one subroutine to another.
  - will be the input to the OOPS observation operator (not the GOM directly)
  - one per observation set
- Hop/tl/ad, hretr – major cleaning, with most observation operator code being moved down into subroutines.
- Test harness to call observation operator and generate FG departures and TL/AD testing without calling any model code or initialising any model modules : full FORTRAN harness test; unit test of H/H'/H* in OOPS/C++

A few of the practical rules inside the new HOP

- No accessing model variables from observation operator: All necessary information should come through gom_plus
- No accessing the ODB from the model or data assimilation
  - ODB is internal to the observation code
- No accessing DA state information (e.g. lifstraj in yomct0)
  - We still need a few switches like LSCREEN, to be passed in by argument
- Preint* routines, long argument lists for model fields (e.g. PTF, PQF, PO3F, PXPP…), and gems_profs have been removed and replaced by gom_plus
- NVNUMB(X) loop and ZXPP removed; Operators are now selected by varno (e.g. varno%bend_angle triggers GPSRO operator)
- VarBC code is encapsulated in fortran "object"
- Observation sets are encapsulated in fortran object

New structure



Single routine covers direct, TL and AD cases

Direct/TL/AD polymorphism with F95

Optional gom_plus arguments.
If PRESENT they indicate TL or
AD behaviour

```
SUBROUTINE HOP(YDGP5, YDVARBC, YDSET, PHOFX, PBIAS, YDGP_TL, YDGP_AD)
```

```
CALL OBSOP_CONV(ROBHDR, ROBODY, SATHDR, SATBODY, ROBSU, YDGP5,
        VARNOS_TO_PROCESS, LSCREEN, LLFINAL_TRAJ, YDSET, PHOFX, YDVARBC,
YLVARBC_EXTRA_PRED, YDGP_TL=YDGP_TL,YDGP_AD=YDGP_AD)
```

```
CALL PPNEW(YDGP5, ZVPOBS, ICOUNT, ZXPP, CDPAR=CLV, YDGP_TL=YDGP_TL,
      YDGP_AD=YDGP_AD)
```

Direct (usually always)

IF (LLTL)

IF(LLAD)

Separate TL and AD routines are still sometimes
necessary at the very lowest levels

OOPS outlook

* IFS/CY44: SPAMing of MODEL variables, finish obs op, re-factor FIELDS object (=> changes in GMV/GFL), INCREMENT object, TRAJECTORY object (new code in IFS)

=> apply SPAMing to Arpège and LAM variables (for CY44 or soon after)

=> phase especially LAM code to changes of GMV/GFL structures

=> learn about new structures and codes (new CONTROL variable code for OOPS, ...)

* generalize obs operator test programs (H/H'/H*); how to share these test programs with the other partners ?

* VAR prototypes for Arpège or Arome: update them to CY42R3 or CY43 ?

OOPS, other stuff and outlook into the future

CY44 could be the last significant re-factoring cycle of IFS; CY45 could be the first stable OOPS-compliant IFS cycle (2017); OOPS in e-suite at EC at present expected for 2018

Other re-factoring going around at GMAP:

* Full-POS towards more flexible input possibilities and OOPS interfacing => so-called configuration 903 (Ryad)

* oopsification of LBC code (Bogdan)

* studying the DFI code in the OOPS context (Aladin partners)

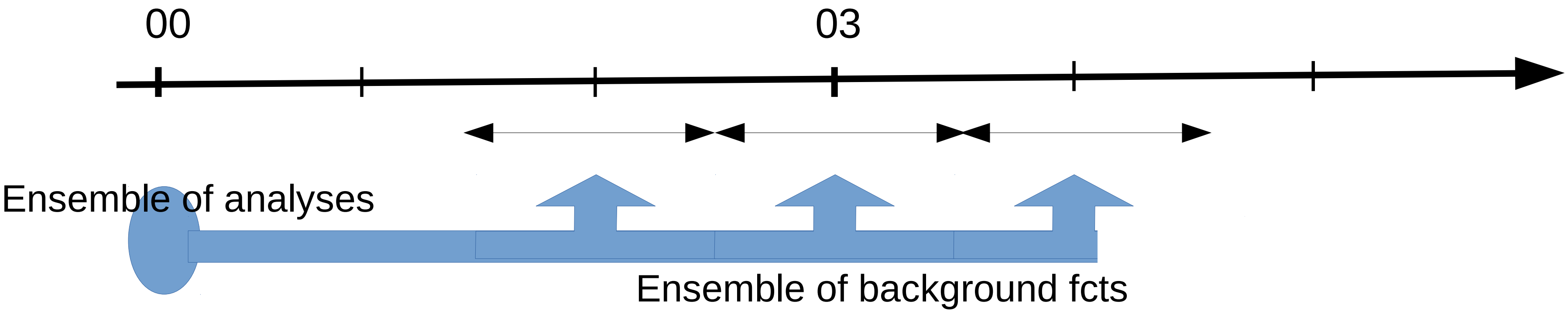
There is more, on the longer run, than OOPS, to come: ATLAS/MIR, COPE, ...

OOPS prototype of LAM VAR

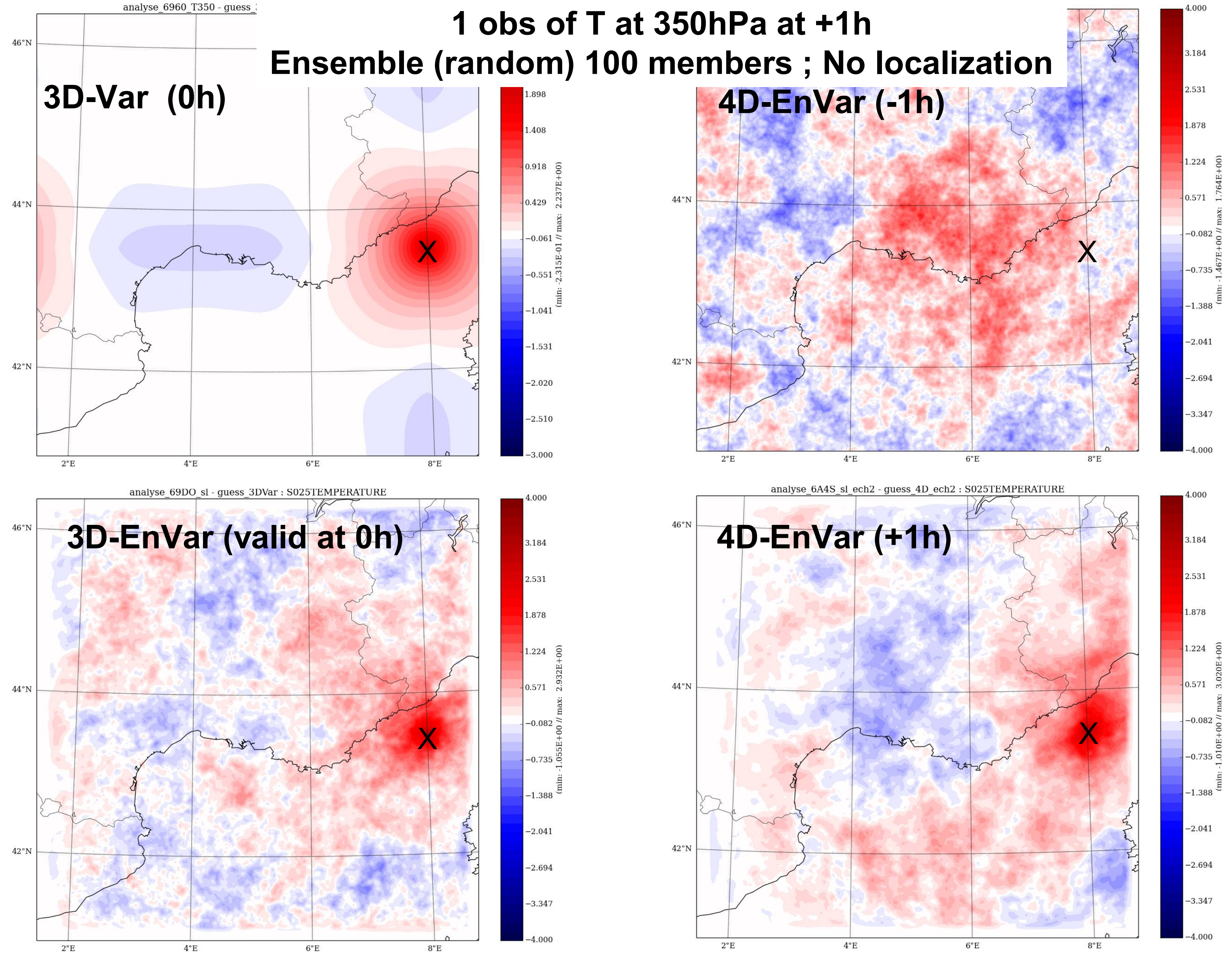
- **AROME 4D Screening at model resolution (2.5 or 1.3km) :** AROME microphysics used for the Reflectivity obs op ; Reflectivity observations => relative humidity pseudo-profiles

- **4D-EnVar minimisation at model resolution**
  - B is constructed from an ensemble of 100 members
  - all trajectories are computed over 4h of fct ; initial states are taken from an initial AEARO experiment

- **Assimilation window of 3h with 1h timeslots** centered on -1h, 0 and +1h.



1 obs of T at 350hPa at +1h
Ensemble (random) 100 members ; No localization


analyse_6960_T350 - guess_
3D-Var (0h)
4D-EnVar (-1h)
3D-EnVar (valid at 0h)
4D-EnVar (+1h)
analyse_69DO_sl - guess_3DVar : S025TEMPERATURE
analyse_6A4S_sl_ech2 - guess_4D_ech2 : S025TEMPERATURE
X
46°N
44°N
42°N
2°E
4°E
6°E
8°E
1.898
1.408
0.918
0.429
−0.061
−0.551
−1.041
−1.531
−2.020
−2.510
−3.000
(min: -2.315E-01 // max: 2.237E+00)
4.000
3.184
2.531
1.878
1.224
0.571
−0.082
−0.735
−1.388
−2.041
−2.694
−3.347
−4.000
(min: -1.467E+00 // max: 1.764E+00)
(min: -1.055E+00 // max: 2.932E+00)
(min: -1.010E+00 // max: 3.020E+00)

End of the boring talk

- For questions , please check the codes ...

```
algoritmo OlaMundo;
início
    imprima("Olá, Mundo!");
fim
```

End of the boring talk

- For questions , please check the codes ...

G-Portugol - A programming language with Portuguese keywords

Obrigado pela sua atenção

Code management aspects

IFS/Arpège coordination meetings (S1/2016): 21 January, 13 April, 6 June (main physical meeting before CY44)

MF/Aladin/Hirlam coordination meetings : technical videoconference on 17 March, HMG/CSSI meeting on 8 April

For the **obs operator re-factoring work** :

* kick-off 3 day meeting in Toulouse (July 2015)
* about 3 technical videoconferences in autumn 2015
* a not countable number of e-mail exchanges between EC and MF

OOPS Board meeting on 19 January 2016

* acknowledged significant progress in re-factoring
* EC want to run a simple 4D-VAR by March or April 2016
* partners : contributions to OOPS code by MF and Hirlam ; participation to the obs operator work (TL/AD) ; build of unitary tests for the H/H'/H* objects (MF)

Encapsulation of variables in MODULEs: ASSOCIATE

Use of an alias in order to keep source code readable:

Definition of the alias

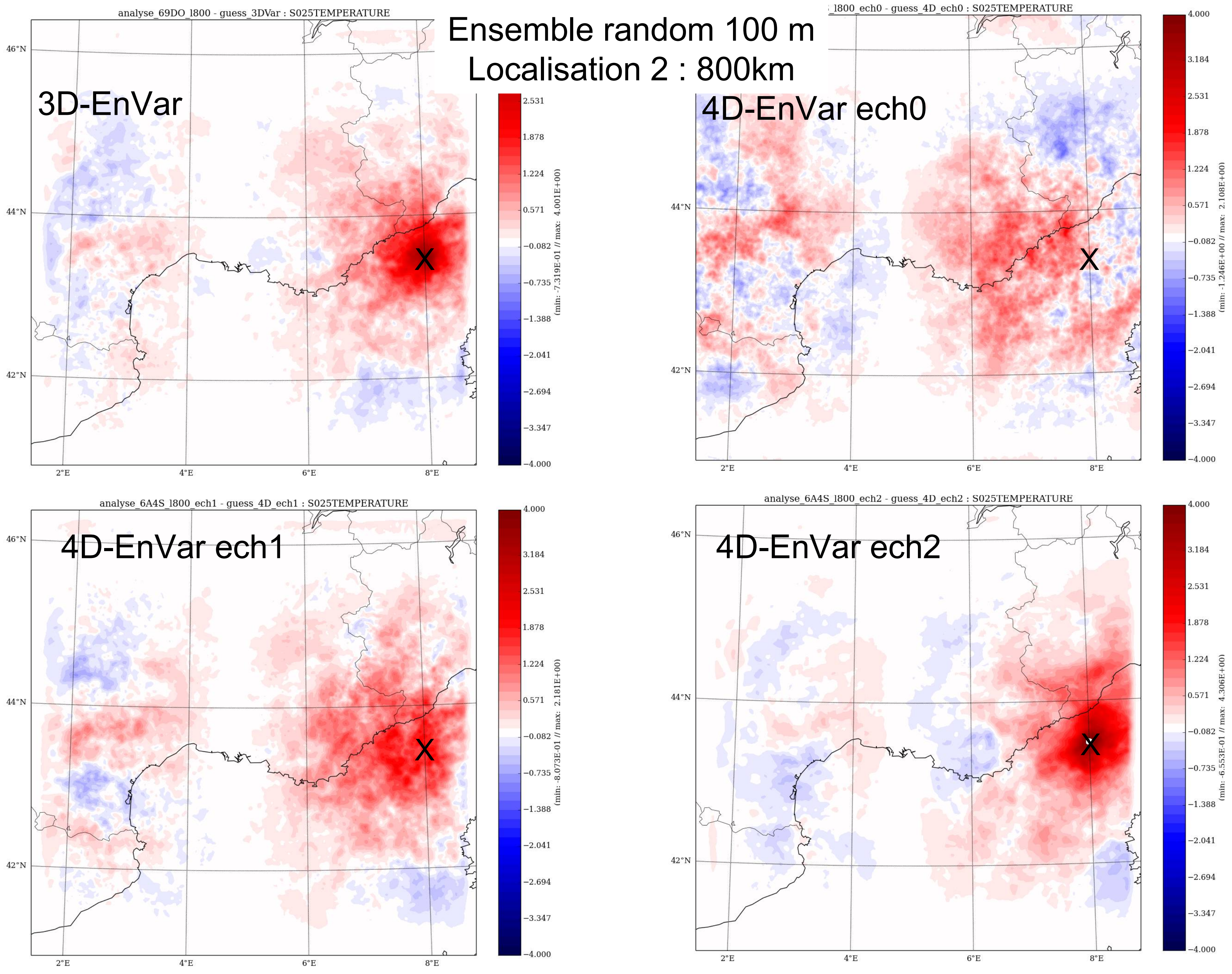
```
SUBROUTINE MF_PHYS (..., YRDIM, ...)
!--------------------------------------------------------
ASSOCIATE(NPROMA=>YRDIM%NPROMA, NPROMM=>YRDIM%NPROMM, &
 & YT0=>YRGMV%YT0, YT9=>YRGMV%YT9, YT1=>YRGMV%YT1, &
...
 & NFLEVG=>YRDIMV%NFLEVG, NFLSA=>YRDIMV%NFLSA, &
 & NFLSUL=>YRDIMV%NFLSUL)
IF (LHOOK) CALL DR_HOOK('MF_PHYS',0,ZHOOK_HANDLE)

...
IF (LLDIAB.AND.(NDPSFI == 1)) THEN
   CALL CPQSOL(NPROMA,KST,KEND,PRE0,PSP_RR(1,YSP_RR%YT
   %MP0),PQS,ZQSATS,PQSOL)
ENDIF
...


IF (LHOOK) CALL DR_HOOK('MF_PHYS',1,ZHOOK_HANDLE)
END ASSOCIATE
END SUBROUTINE MF_PHYS
```

Use of the aliased variables inside compute code

Ensemble random 100 m
Localisation 2 : 800km


analyse_69DO_l800 - guess_3DVar : S025TEMPERATURE
3D-EnVar
46°N
44°N
42°N
2°E
4°E
6°E
8°E
X
2.531
1.878
1.224
0.571
−0.082
−0.735
−1.388
−2.041
−2.694
−3.347
−4.000
(min: -7.319E-01 // max: 4.001E+00)
_l800_ech0 - guess_4D_ech0 : S025TEMPERATURE
4D-EnVar ech0
44°N
42°N
2°E
4°E
6°E
8°E
X
4.000
3.184
2.531
1.878
1.224
0.571
−0.082
−0.735
−1.388
−2.041
−2.694
−3.347
−4.000
(min: -1.246E+00 // max: 2.108E+00)
analyse_6A4S_l800_ech1 - guess_4D_ech1 : S025TEMPERATURE
4D-EnVar ech1
46°N
44°N
42°N
2°E
4°E
6°E
8°E
X
4.000
3.184
2.531
1.878
1.224
0.571
−0.082
−0.735
−1.388
−2.041
−2.694
−3.347
−4.000
(min: -8.073E-01 // max: 2.181E+00)
analyse_6A4S_l800_ech2 - guess_4D_ech2 : S025TEMPERATURE
4D-EnVar ech2
46°N
44°N
42°N
2°E
4°E
6°E
8°E
X
4.000
3.184
2.531
1.878
1.224
0.571
−0.082
−0.735
−1.388
−2.041
−2.694
−3.347
−4.000
(min: -6.553E-01 // max: 4.306E+00)