OOPS technical meeting Number 3

about content in CY38 and progress report on 3D-VAR prototype & OOPS toy model

held on Monday April 18 ;
participants (MF) : Stéphane Martinez, Guillaume Beffrey, Florence Rabier, Patrick Moll, Karim Yessad, Claude Fischer, Pascal Lamboley, Yves Bouteloup, Ludovic Auger
participants (EC) : Deborah Salmond, Tomas Wilhelmsson, John Hague, George Mozdzynski

1. Deborah's talk from Norrköping – a replay

Deborah gave per visio-conference her talk from the Aladin/Hirlam workshop to MF attendance. See her PDF file at :
http://www.cnrm.meteo.fr/aladin/spip.php?article222

2. various questions addressed during and after her presentation :

- Karim will prepare a new version of his cleaning document (V7) probably for June. The new version will be pruned by all items which have been done by EC and MF over the past year
- MPI setup would be done below the C++ layer, so any "fieldset" carries on its own grid and grid data distribution (only the very highest level of MPI initialization remains global : MPI_world, number of processors, …)
- all CNT. Routines will disappear in the prototype, including CNT4 (they're replaced by C++ codes)
- STEPO disappears in the prototype
- MF asks whether the "fieldset" structures with pointers totally is F95-ANSI compliant => EC answers 'yes'. The extensions of F95 will only be needed to compile the extra thin Fortran layer containing the interfaces to the methods. Those will be called directly from C++ using the C_PTR facility (which is a F2003 extension). MF indicates that the C++/F95 interoperability still is not working properly on NEC/SX9 (eg. The toy OOPS does not compile on MF's NEC for the time being – this problem has been submitted to the NEC support).
- EC mentions that Anne, during her visit in May, can bring the so-called « mini-oops » code with her, for showing it and helping installation at MF. The mini-oops would contain the C++/F95 interfaces and the presently existing code for calling the IFS obs operators
- Jb/CVAR3IN : to be discussed in detail on Wednesday morning, with a decision about whether the encapsulated code by Mike can enter CY38 already.
- Norm violations : a significant progress has been achieved over the last 3 cycles. MF and EC must make sure that this situation can be maintained and even further improved. EC mentions that more could be done in MF's physics (phys_dmn/ code), especially about useless declarations of variables and the systematic usage of USE, ONLY for modules.
- MF asks whether EC could make a version of pre-CY37R3 available as soon as possible for visual checks, as this pre-cycle already contains much of the IFS-cleaning and re-factoring for the mini-oops. EC will make the pre-cycle available asap, even before a decision is taken about Mike's Jb code. A second version could be issued later on, including the encapsulated Jb, if the latter also is chosen to enter CY38. MF would then test compilation and execution of a global Arpège forecast on NEC and maybe workstation.
- MF asks about the handling of spectral buffers (SPA3, SPA2)for OOPS: EC says that this

has not yet been investigated for the 3d-VAR prototype (nor CY38) since they are not needed. They will however be considered for after CY38. For the time being, the idea is to have the spectral fields as part of the "fieldset" super-structure.

Items to enter CY38 on the side of ECMWF :

- encapsulated GOM/GLOBS
- COBS+COBSLAG => one single COBSALL
- "fieldset" structure using GMV/GFL/surface_fields derived-type structures containing POINTERS (instead of ALLOCATABLE arrays)

Items to enter CY38 on the side of MF :

- various encapsulated variables : arrays for GPXYB, CPG_GP ; structures created for SL parameters at the level of CALL_SL, LARCINA/B, (E)LARMES, LAPINEA/B (but for instance not (E)LASCAW which is a routine that could be externalized later on, so we want to pass only dummy arguments to it)
- additional attribute for surface_fields (SU_SURF_FIELDS) to tell whether a field actually is activated or not => this info will be used later in the obs operator to check for consistency between the activated model surface fields and the fields requested in the GOMS arrays for computing the obs equivalents
- cleaning in FMR15 (Arpège-Climate physics code)
- several in-linings of code in the spectral computation, in I/O, …
- miscellaneous : removal of DFI1 routine (DFI2 is kept), more diffusive filters in Full-Pos, …
- encapsulated and more robust Davies relaxation code for LAM (*this contribution will come after CY37T1, but will enter CY38*)

Leftover for after CY38 :

- cleaning and re-organization of HOP (make HOP_obstype)
- link between surface_fields activation attribute & consistency checks when allocating GOMS for surface field model equivalents in obs operators

to be further checked : MF will run the norm checker on CY37T1, and start some cleaning in MF's physics code (possibly before CY38)

3. <u>About 3D-VAR prototype & OOPS toy model</u>

- Yannick has modified the QG model code so that it looks more like what the OOPS/IFS would do (push all computational code into Fortran)
- Mats Hamrud will do the technical coordination on the Fortran OOPS/IFS side for the 3D-Var demo
- EC will arrange a full scientific and technical review of the C++ toy model code in May. Baudoin Raoult & IBM contact persons will take part as reviewer. It was left open whether MF should participate.
- A discussion took place on the usage of "templates" in the latest versions of the toy : while templates avoid some potentially heavy code/link analysis for the compiler at execution time, they are felt by some OO-experts as a step away from polymorphic derivation and the use of the full potential of C++ for OO. EC confirms that one reason to use templates was the fear of a breakdown of performances at execution if too many features of dynamic

polymorphism are to be solved by the compiler. Conversely, MF asks why this should be so important if the C++/OO layer remains shallow in the OOPS system, since it should then represent a marginal part of CPU (compared with the computational code in Fortran). Pascal points out that using templates forces to pass much more code into the ".h" header files, which may be an eventually inelegant way of handling the C++ code. Thus, templates were felt to make the code less readable in the last toy/OOPS versions.

About the 3D-VAR prototype: EC expects to have this code ready by September/October; MF won't be able to work on it before completion of CY38, so not before November.
Technical ingredients in the prototype: a real ODB database + an IFS GRIB file (model state input to feed the background field) + XML/namelist configuration files & Fortran main program + thin C++ algorithmic layer + the Fortran method layer (C_PTR's) + the IFS CY38 with possibly some additional stuff to make the prototype run.

MF's potential involvement could be: read and understand the prototype, install it and make it work, evaluate its pros & cons in terms of IFS software evolution; adapt to a global Arpège file and run the prototype in Arpège-mode; plug-in a LAM ODB and a LAM Jb and try out an Aladin version.
Note: these actions will be further discussed at MF (they are not yet clearly planned in MF's workplans & resources).

Post-meeting question (by Claude, and answer by EC):

```
    2. a question arose post-meeting: will the IFS fieldset contain all
    time step arrays (T0, T9, T1) or will there be one fieldset per time
    step (sort of "fieldset_T0, fieldset_T9, fieldset_T1 ...)  ?
    I remember in January Yannick saying that the fieldset should contain
    everything to re-start a model integration.
    Have you eventually kept this logic ?

The *state* should contain everything to re-start integration. It has
not decided yet whether the fieldset should be only one time step, or if
it should have all time levels of the full model state.  For the 3D-Var
demonstrator only one time level is needed.
```

4. AOB:

MF should initiate a test with 3 point visio-conf when ready: EC (contact is Rémy Giraud), MF, Aladin/Hirlam partner (SMHI/UIf Andrae is interested)

Remainder : next meeting will be devoted to the new Jb code, on Wednesday, April 20 (9.30am Reading ; 10h30 Toulouse) + Fortran coding standards (agree on final version ?)

next physical coordination meeting will be held in Reading, on Monday, June 27 (whole day). MF delegation will be Claude, Ryad and Guillaume.