OOPS technical meeting - 20th October 2011
<u>Coding norms, IFS cleaning and OOPS aspects</u>

Participants (MF) : Claude Fischer, Karim Yessad, Thibaut Montmerle,
Stéphane Martinez
Participants (EC) : Deborah Salmond, John Hague, George Mozdzynski
Mike Fisher, Tomas Wilhelmsson

1. New Coding Norms Document for IFS

The latest version of the coding norm document was discussed. A few adaptations will be done:

- add a template to Section 1
- add a pointer to the old coding norm document, stating that the new one supersedes it
- add some more specific comments about when cosmetic cleaning should be avoided
- be more specific, though flexible, about splitting up big modules
- new title to Section 4: “detailed Fortran coding standards”

A new norm checker (or a revised version of the old one) needs to be prepared. This would be done by MF.

Action 1: Claude to produce updated version of coding norm document, and send it back to Mike

Action 2: Claude to check if MF can provide a new norm checker version. Deborah to check who is contact at ECMWF

2. Cleaning action, based on Karim's document V7c

Karim prepared a 2-page note listing his own expected contribution to CY38T1.

The follow-on discussion was based on the “cleaning” action document, version 7c by Karim. 3 action have been emphasized:

1. remove the linear algebra routines “LUBKSB” and “LUDCMP” and replace them by standard calls to BLAS or LAPACK libraries. Alternatively, move those two routines to the XLA project => action ECMWF
2. replace the RIPI* set of arrays used in ECMWF's physics by the derived type now declared in CY38 => action ECMWF
3. replace the setup calls to SUGAWA (initialization of Gaussian weights) by the equivalent calls to TFL setup. According to Karim, two occurrences are trivial but a third one (the first call in the Setup) requires some re-ordering in the setup. George mentions that ECMWF will include the Fast Legendre Transform code into a CY38R1, and this may create interference. Also, Tomas' work on the Geometry object can have an impact (as the grid definitions and setups would be re-arranged in an OOPS manner). Deborah will check the issue with Mats Hamrud. The cleaning will only be done if feasible with respect to the other changes expected for CY39 (ECMWF to decide). George mentioned that the core code for FLT had been re-written in IFS-style at ECMWF, due to licensing considerations.

Actions: ECMWF

- **move or replace LU* routines from ECMWF's physics[1]**
- **extend the use of the encapsulated RIPI* arrays**
- **decision about SUGAWA**

Furthermore, some of the appendixes were addressed: E, I, (L, M). It is decided to come back to this discussion after Karim has updated his cleaning document. For Appendix I, we will need to set priorities (packing some small modules together). I1 probably is to be done by MF after CY39. For appendix E, we would start with some simple cleaning (renaming), since we will need more brainstorming and overview about how we need to handle namelists with respect to encapsulated Fortran code along with the code re-arrangements for OOPS.

Action: Karim to deliver an updated version of his cleaning document, after CY38 is declared

3. <u>OOPS/C++ aspects & evolution of B-matrix and control vector code:</u>

Claude gave a summary about the porting issues of the toy model code on NEC/SX9: MF has been working on porting an old version from April 2010 (probably enough to test C-binding). Some good progress has been achieved:

- NEC has provided a new compiler version that allows compilation of C-binding
- several small bugs have been fixed for NEC; one was subject to a compiler correction by NEC
- the present limitation to the exercise is that C-binding and double precision options are not compatible as compiler options. Therefore, all C++/Fortran Fortran interface routines had to be compiled after explicitly declaring double precision (KIND) for all REAL variables. This is probably acceptable within the IFS, as explicit KIND declarations are the rule. This would be problematic, and presently blocking, in case C-binded Fortran code is imported from outside the IFS collaboration[2]
- MF will consolidate this information: further control and validate the OOPS toy QG and L95 runs & further discuss with NEC on the C-binding/-Adouble incompatibility

Action: Claude will summarize the code patches in the old toy model, and send the information to Mike. Mike will check and adapt the OOPS code w/r to NEC porting issues.

Claude stresses the excellent collaboration with Louis-François Meunier, from MF's IT Dept.

The status and foreseen evolution of the B-matrix code (and control vector) was discussed both from the side of the Fortran IFS bottom-up (will we need further significant changes w/r to the already performed re-coding for CY38 ?) and from the side of C++ top-down design.
Mike explained that the Fortran code would not be changed significantly any more. All what is needed to progress with the 3D-VAR demonstrator is in. The control vector handling also would remain in Fortran, with only pointer management done at C++ level. The latter aspect however is not yet totally analysed, and several issues remain open (additional variables in the CV, for instance

[1] Note from after the meeting: LUBKSUB and LUDCMP are from Numerical Recipes (and contain comments to that effect). They are also included as module routines in MODULE ARRAYTOOLS (aeolus/support). They can be replaced with DGETRS and DGETRF from LAPACK. (In “cloudsc”, they are called one after the other, so we could use the simpler LAPACK routine DGESV, which does the decomposition and solves in a single call)

[2] After the meeting, Mike pushed changes to our internal OOPS repository to make the real kind explicit in the OOPS toy models. We can now compile with automatic promotion of REALs turned off and get bit-identical results.

).
Thibaut and Claude stress that MF and the LAM partners will need a good analysis and ensure that/how developments of extended CV and B-matrices can be implemented, and whether this can be done rather soon or whether one should wait for more progress with OOPS. Typically, we have hybrid B-matrices (Thibaut wants to move from adding an extra dimension to the B matrix arrays towards multiple declarations of a baseline B) and hybrid VAR (Hirlam will wish to add the α-coefficients that weight ensemble member statistics in addition to a baseline B). Mike agrees that the optimal solution is to have multiple elements rather than a single matrix with an extra dimension in Fortran. The decision is to keep the inclusion of these developments into the IFS code ***on hold***, and first study and understand the VAR demonstrator.

Mike further explained that the demonstrator would have a different pre-conditioning strategy than the IFS VAR. So far, it has only multiplications by B/B-1, not their square roots, and this is done in the minimizer. The CV object presently contains δx, rather than χ. Also, CONGRAD has not been ported into OOPS[3].
MF is somewhat sceptical about these choices, as they do not reflect the existing VAR code in the IFS. Claude asks whether ECMWF keeps the goal to have equivalent algorithms in OOPS and IFS. ECMWF conforms that yes. This issue could be addressed during the scientific review.

C++ tutorials: MF will organize a first training for beginners in mid-November. Another session dedicated to the OOPS code should follow (advanced tutorial). ECMWF will also arrange a C++ tutorial, as well as a specific OOPS code training. Claude points out that MF and ECMWF should exchange on their respective experience with the tutorials, and possibly assess whether some common actions can be envisaged later on. This can be of interest to the wider Hirlam/Aladin community as well.

4. Next meetings:

There will be a coordination teleconf between MF and ECMWF on Thursday, November 24th. We could then address as a specific issue the progress and timetable of COPE developments, and what would enter CY39.

An OOPS Steering Committee meeting is scheduled for November 30th. We plan to present the new coding norm document at this SC meeting.

The next OOPS technical visio-conf could take place in December, with at least the following items on the agenda:

[3] Post meeting note on CONGRAD from Mike:
As Karim states in his document, CONGRAD has become a bit too multi-purpose and tied to the IFS. For this reason, I do not intend to simply port it to OOPS. We will have to decide what functionality we want to port, and how we can implement it more cleanly than in the current code.
There are two equivalent ways to precondition the conjugate gradient algorithm. In one, we define a pre-conditioner M that approximates the inverse of the system matrix (the Hessian in our case). In the other, we perform a change-of-variables using the square-root of M. The two produce identical iterates in exact arithmetic. (For 3D/4D-Var, M=B.)
The square-root approach has a disadvantage when it comes to wavelet Jb in that the control space is much larger than the dimension of the model. Also, preconditioning with M means that the 3d-Var demonstrator will not be delayed by the work on the control vector.
Many algorithms can easily be transformed from one approach to the other. However, we are aware that there are some cases where we need the square-root approach. (In particular, this is the case whenever we make use of the fact that the background error covariance matrix for the control vector is the identity matrix if we use the square-root preconditioning.) For this reason, we fully intend to implement the square-root preconditioning in OOPS. It is already there for the toy models, but not yet for IFS.

- wrap-up on the actions decided in the October visio-confs (6 and 20)
- status on 3D-VAR demonstrator
- Karim's updated cleaning document: consolidation of cleaning actions for CY39
- if relevant: a wrap-up on Tomas' work with the Geometry object