OOPS technical meeting Number 1

about the preparations for a 3D-VAR demonstrator & updates on the toy model

held on Thursday 20 January 2011 ;
participants (MF) : Ryad El Khatib, Patrick Moll, Karim Yessad, Claude Fischer, Pascal Lamboley, Thibaut Montmerle
participants (EC) : Yannick Trémolet, Deborah Salmond, Anne Fouilloux, Mike Fisher, Tomas Wilhelmsson

1. Definition of Model « State » and module classification

Tomas has started to analyze and prepare modules extracted from the IFS, in order to plug them in the toy system. He is identifying the fields (GMV,GFL,Surface fields etc) which are part of the « state » and will wrap them into an OOPS "fieldset". One issue which will be analyzed is the exact definition of a « state ». Tomas will start by pulling together what is needed for passing a state from the forecast model to the obs operator for the 3D-Var demo.

Claude asked whether the state would contain only a single timestep (this is his question also for the further adaptation of the SW code) ; Yannick explained that the state should contain anything needed for a model re-start, as this is what is needed to perform one single model timestep. So one should keep a « re-start file » type of idea in mind (for instance : for a two-time level scheme). Mike explains that the toy QG has simple timestepping so far (only one timestep), but he could add a two-time level scheme.

The work will continue, always with the idea to change as little as necessary the IFS code and eventually plug in the new IFS-derived code directly into the OOPS toy.

Action: Tomas will write a report about his work on the externalization of the modules for the obs operators, and send it to MF.

2. Claude's Shallow water work

Claude briefly explained the exercise with the SW code : clean it, re-organize it at Fortran level in the OOPS spirit and plug it in the toy. The SW code originally has been coded in an IFS-style, with a certain number of options (Eulerian versus SL, LAM versus cyclic, some I/O and LBC coupling). It does not have a TL/AD or MPI/Open-MP however. So the exercise would be pedagogical and addressing development and analysis aspects from a different view than in QG and L95 (which were coded from scratch in the toy environment).

Yannick raised a few issues on the present version sent by Claude :

- « fieldsets » should be defined and included, for defining and handling either spectral or gridpoint data. Also, any useful low-level tool (not bearing any meteorological content, only data handling) should be appended at this level, to the « fieldset » (I/O, data distribution, change of resolution). « fieldsets » should be associated with any information related to data structure on that grid. This is an important aspect before addressing the definition of a « state ». So, MF will need to study the toy model code more in depth with that respect …
- Claude explained that he has not yet a good view on what to put in the state : this aspect will need further investigation (see discussion above on « re-start » definition)

- Yannick has spotted useless structures in the modules, like the « look-up » table. This type of feature indeed relates with global management of states, which is not OO-compliant.
- A discussion took place on how to best continue the code adaptation. One possibility could be to duplicate existing routines from the L95 or QG level, empty them and fill them with SW code. This strategy eventually is not retained so far ; the bottom-up recoding may be more efficient for the learning process. Guidelines for how to implement a new model were considered (not decided however ; let's see if we can proceed without them)

Claude mentioned that his intention rather is to implement the model forecast code, and then possibly test some ensemble system wrapper. Yannick said that it will be useful to also code and implement a variational assimilation part for the SW, since otherwise the analysis and implementation learning process would not be complete. A 3D-VAR could be useful already (addresses B, obs, some elements of a trajectory and a minimization). This idea is retained for … some future ...

3. Encapsulation of OBSHOR from IFS

John Hague has started to analyze the horizontal interpolations for the 3D-VAR demonstrator. This work will be reported in a future meeting.

4. Encapsulation of HOP from IFS

Deborah presents the work with Anne on the side of the Observation Operator :

- Horizontal interpolation of model variables on model levels are computed by a preliminary screening / Traj0 step. This step fills an ODB and GOM arrays.
- Code for the radiances has been extracted from HOP into a new HOP_RAD, along with the full RTTOV code. A C++ wrapper has been coded to call HOP_RAD (NB : no TASKOB level seems needed). For now we assume vertical level interpolations are done inside RTTOV, but the option of doing those in HOP_RAD will be investigated later.
- The new application so far calls the setup for ODB and RTTOV then reads the ODB and the prepared GOM data, calls HOP_RAD which computes the RT part. Then, the H(X) data are stored in the ODB.
- This 'mini-OOPS' which for now just runs the Obs_operator uses the 37r1 IFS libraries called from the C++ OOPS framework.
- As yet VarBC has not been included in this test.
- Deborah mentioned that only little recoding in HOP was necessary. ECMWF expects that HOP will be split by obs types, as required by the proposed class derivation in OOPS (refer to Anne's presentations in 2010)

action: Deborah will send the code to MF, along with some description. Patrick will be MF's main contact (*NB by Claude : at MF, we may have to spread a little the information to a few other obs developers, for instance involve our radar staff*)

5. Encapsulation of JB from IFS

Mike gave an overview of the work he has started in December on externalizing the Jb code :

- isolate any Jb code, and strip out whatever is not strictly speaking B-handling like VarBC and other extra control vector elements (*NB by Claude : MF will have to pay attention to the mean wind handling for Aladin*)

- this work has started at the level of CVAR3IN, rather than CHAVARIN (which contains too many and various codes)
- Mike now has an isolated code which compiles and links
- LAM switches have been kept untouched (dummy routines are handled for linking)
- next steps : AD code and setup
- further next steps : brainstorm on control « vector » / « variable » and « state » definitions

action: Mike will send code and description to Thibaut – done after the meeting.
(NB from Deborah: We must be careful in future to check that our emails are not 'spammed' away. Claude: MF staff has difficulties for tracing emails with zipped attachments, as those are placed in "quarantine")

6. Update on OOPS toy

Yannick explained that more work has been done in the toy C++ layer over the past months, but that the OO layer little by little becomes stable. For instance, the interface layer between C++ and F95 should not change significantly now. OO code related to model definitions also should be stable ; code specifically prepared for the obs operators still is at work. MF mentions that the C++ code layer remains difficult to read and understand (Pascal and Claude).

On forthcoming changes / developments :

- control vector / control variable
- minimization algorithms and their interface / Yannick mentions that CERFACS will import the OOPS code ; collaboration is started with Serge Gratton and his team / *post-meeting info : the minimizers will be coded in C++ (CONGRAD from the IFS, as well as CGMOD and RPCG by Cerfacs ; NB : not M1QN3 but Yannick thinks that this would not be too difficult to do)*
- outer loop

Remainder : next meeting will be devoted to IFS cleaning actions, on Tuesday 25 January (1pm Reading ; 14h Toulouse)

next meeting on OO aspects and 3D-VAR demonstrator : to be decided after exchange of doc and code ; MF would propose a date in due time