PROPOSAL OF CLEANINGS IN ARPEGE/IFS IN 2009-2010.

YESSAD Karim.

June 3, 2009

Version 5d. Basis of study: CY35T2R3

1 Introduction and purpose.

This paper is an updated (and shortened) version (basis CY36) of a paper written in 2007, and we don't recall all the introduction of the previous versions. To sum-up we briefly re-call the purpose of this paper and of the actions listed inside.

- The annual increase of code volume is around 6% (i.e 50% each period of 7 years and 100% each period of 12 years). That means that there is a necessity to reduce this annual increase.
- There is a necessity to do regularly cleaning actions in the code, in particuliar for the following topics: proper naming of routines, prune obsolete code and obsolete variables, better use of F90 features to improve the code concision, improving comments, improving the library architecture, externalisations, reduce useless code duplication, respect to the coding standards.
- It would be desirable from now on to have an automatic cleaning cycle every 2 years (3 full cycles). An automatic cleaning cycle would be desirable in 2010.

This version focusses on actions which should be done during the second half of 2009 and in 2010 (targets: cycles 37 and 38).

Three levels of priority to do these cleanings have been defined:

- (P0): the high level priority tasks which should be done for CY37 or in an automatic cleaning cycle done just after CY37 if decided.
- (P1): the medium level priority tasks which should be completed before the end of 2010.
- (P2): the low level priority tasks which can be completed after 2010.

This "V5" version of "cleaning memorandum" is the basis of work and is the version of the document which will be available at the ARPEGE/IFS coordination meeting (25th June 2009 at ECMWF). It includes some proposals of Ryad.

2 Cleaning of obsolete options.

The following options can be good candidates for pruning, and the target is CY37 for the high priority prunings (P0).

Target for CY37:

- (P0) Option LPC_OLD in the non-hydrostatic model: scories in the TL and AD codes.
- (P0) Option LSITRIC=T in the semi-implicit scheme (currently useless, can work only for unstretched global models with LIMPF=F).
- (P0) Old content of routines LAIDLIAD and LAIDDIAD.
- (P0) Options NTRSLTYPE=0 and 1 in the TL code of the semi-Lagrangian code (only NTRSLTYPE=2 is used operationally; the code of NTRSLTYPE=1 is not complete and probably does not work).
- (P0) Unused options for DFI: optimal filter (OPTFIL), recursive filters (RECFIL), and maybe also ideal low-pass filter (SMPFIL).
- (P0) TRAGEO, TRAGEOAD and normal mode initialisation with variable mesh in ARPEGE/IFS (code calling TRAGEO). Replace calls to TRAGEO and TRAGEOAD with ABOR1 (these applications will remain for unstretched untilted geometry only).
- (P0) Configuration 912 (rotation matrices for TRAGEO, unused since 1997).
- (P0) In the calculation of the displacement in the SL scheme, remove the useless option (LSETTLST,LPC_NESCT)=(F,F). That will lead to a significant simplification of routines (E)LARMES, LAVENT (and also their adjoint and TL codes).
- (P0) Remove key LPSLHDN and the obsolete option LPSLHDN=T (NAMDYN,SUDYN).
- (P2) Option LPC_OLD in the non-hydrostatic model: direct code. About this option a decision must be taken in December 2009 with the ALADIN partners to know if its removal can be kept as a target for CY37 or if it must remain a couple of years more in the code.

Keeping or pruning the following options is also questionable in the future.

- (P2): Some values of NLANTYPE seem to be not used in configuration 601 and the corresponding code is not regularly validated. What is used in the MF PEARP (ensemble prevision at METEO-FRANCE) is NLANTYPE=1 (the NLANTYPE=6 code may be called in a configuration 601 with NLANTYPE=1). NLANTYPE=5 is maybe useful also. Options which seem useless are NLANTYPE=2,3,4,7,8,9,10,11.

  Requires information from more people to have a comprehensive list of useful or potentially useful options.
- (P2): remove the minimizer N1CG1 (no longer used for conjugate gradient) and all the pieces of code using it (will wait a couple of years, some people still want to keep it for tests for the time being).

3 Cleaning of obsolete (unused) variables.

Some variables have been used in the past; they are now not used any longer, but their declaration and set-up has not been completely removed. The list of useless variables is partly provided in Appendix G. In project ARPEGE/IFS, modules to be examinated are ...mix.F90, par...F90, per...F90, ptr...F90, qa...F90, yem...F90, yhl...F90, yoe...F90, yom...F90, yop...F90.

Actions and priority:

- (P0): remove all the namelist and module useless variables in project ARP/IFS and ALD, referenced in appendix G.
- (P2): remove useless variables in the other projects.

4 Improvement of commenting.

A lot of routines may be insufficiently commented, or may contain false comments. Concerning the last points, there have been code evolutions where the update of the corresponding comments (which were originally correct) has not been done properly, so the comments have become partially wrong during the time. It is not straightforward to search for all the ill-commented routines, and such work must be regularly done by the responsible people. In particular we can focus in two sets of comments:

- Comments in modules: remaining uncommented modules are listed in Appendix A.
- Comments in routine header to say briefly what the routine does: list routines where they are missing and add missing comments in English. A partial list is provided in Appendix C.
- Comments for dummy arguments: list routines where they are missing, and add missing comments in English. The comprehensive list of relevant routines is not provided here and is actually difficult to obtain (spread in all directories: probably a lot of routines; some of them have been found in ECMWF radiation routines or CANARI routines).
- Comments in French: translate them in English where no English comment is currently provided. French comments without any English translation are found mainly in the following routines:
  - arp/canari/ca...F90 routines (CANARI).
  - most of arp/dia DDH routines.
  - most of arp/function/qa...h routines (CANARI).
  - most of arp/phys_dmn routines.
  - most of xrd/ddh routines.
  - most of xrd/fa routines.
  - most of xrd/lfi routines.
  - some of xla/internal/minim routines.

  Some other routines (not listed) may contain untranslated French comments.

  To sum-up, French comments are mainly localised in routines which are not used at ECMWF at all (DDH, LFI and LFA software, MF physics, CANARI OI assimilation) with some isolated exceptions for DDH, but they may be used by some of our HIRLAM or ALADIN partners: some actions to do English translation may be required if expressed by some of our HIRLAM or ALADIN partners.
- Isolated other language comments may appear (German for example).

Actions and priority:

- (P0): add missing comments (or correct false comments) in declaration modules.
- (P0): add missing comments for routine meaning (header).
- (P1) or (P2): add missing comments for dummy arguments.
- (P2): add missing comments or update/correct comments in other routines.
- (P2): translate French comments into English.

5 Improvement of the library architecture (projects and directories inside each project).

5.1 ∗ Right place of routines.

Cleaning has been done for CY35. For the following cycles, checking must been done for new routines to ensure they are properly named and placed in projects/directories.

∗ **Dynamical-physical interface routines:** The current status is that the dynamical-physical interface routines are not easy to recognise in the code (no specific prefix in their names; they are spread among several directories like adiab, phys_dmn, phys_ec or phys_radi). People working on physics or dynamics may need to work on these routines and to know what they are allowed or not allowed to do in these routines (for example is it allowed to call GPPRE or not?). A provisional list of such routines has been provided in Appendix B.

Several proposals may be formulated to improve that:

- Proposal 1: isolate the dynamical-physical interface routines (like MF_PHYS, APLPAR) in new directories "phydyn_interface1" and "phydyn_interface2". Directory "phydyn_interface1" is for the first interface level (level of MF_PHYS, RADDRV, EC_PHYS_DRV). Directory "phydyn_interface2" is for the second interface level (level of APLPAR, CALLPAR). Calling adiab/gp... routine (and particuliarly gppre.F90, gppref.F90, gppreh.F90 and gpxyb.F90) will not be allowed in routines stored in "phydyn_interface2" and in routines remaining in phys_dmn, phys_ec or phys_radi.
- Proposal 2 (minimal): move routines which are not in one of the "phys_.." directory in the appropriate "phys_.." directory (for example move CPSPE and POSTPHY in phys_ec). Maintain a documentation where the list of dynamical-physical interface routines is provided.
- Proposal 3: like proposal 2 but with an additional renaming of dynamical-physical interface routines in order to be able to recognise them (example: name containing root "pdi1" and "pdi2", not necessary at the beginning of the name).

  Example: mf_phys.F90 renamed into mf_phys_pdi1.F90; aplpar.F90 renamed into aplpar_pdi2.F90, aplparsad.F90 renamed into aplpars_pdi2_ad.F90.

Remarks about these proposals:

- Proposal 1 does not seem to have the agreement of all partners; the best solution still needs to be discussed.
- Proposal 3: finding a "unique" prefix for level 2 dynamical-physical interface routines will not be easy. There are currently existing prefixes such as "apl", "callpar", "radpar", "hl_apl". The same issue may also occur for the first level ones which do very different things (some of them compute diagnostic input quantities for physics, like moisture convergence, and these quantities may be used elsewhere in the code in the future). This issue disappears if roots "pdi1" and "pdi2" are introduced in the middle or the end of routines names.

∗ **Project UTI:** It would be desirable, for each directory of this project, to have the subdirectories "include", "module", "programs", "src", as it is already done for directories "ctpini" and "pinuts". More details about this proposal are given in Appendix H (part H4).

∗ **Projects ODB, SAT, AEO:** It would be desirable to dispatch routines in directories "external", "include", "module", "programs", "internal" according to their content and the way how they are called. This topic still need discussions. This action will be detailed in a further version of this document.

∗ **Other topics (a proposal of movings has been given in Appendix H):**

- Routine xrd/eclite/syminv.F (which appeared in the pre-cy34) is a linear algebra one (like MINV, but inverts a symmetric definite positive matrix): its right place is in xla/external/linalg.
- Place of routine xrd/eclite/uv2sd.F (which gives the argument and modulus of a vector) is questionable: project XRD or XLA?
- Move arp/utility/chien.F90 and ald/utility/echien.F90 in XRD (proposal of Ryad, to be done at the same time of the externalisation of EGGX).
- Move routines arp/dia/ini1wrfp.F90, arp/dia/ini2wrfp.F90, arp/dia/ini3wrfp.F90 from arp/dia to arp/fullpos. Moving in arp/fullpos is questionable also for suppdate.F90 and fpspnorm.F90.
- Move routine arp/op_obs/expbesu.F90 into arp/pp_obs (this was probably its original place in CY34 and moving it in arp/op_obs was an error).
- Routine arp/var/suvifce.F90 looks like a hat routine for pp_obs routines (vertical interpolator) and is used in applications which are not only variational ones: moving in arp/pp_obs is questionable.

∗ **Priorities for these actions:**

- (P0): move syminv.F in xla/external/linalg and all movings other than the ones referenced below.
- (P0): externalisation of EGGX + move (e)chien.F90.
- (P0 or P1) Appropriate storage of dynamical-physical interface routines according to the solution which will be retained (must be done in an automatic cleaning cycle).
- (P0 or P1): Reorganisation of project UTI according to the solution which will be retained (must be done in an automatic cleaning cycle).
- (P1 or P2): Reorganisation of projects AEO, ODB and SAT according to the solution which will be retained.

5.2 ∗ Externalisation.

∗ Linear algebra: Externalisation in the new project XLA has been done in CY35. Since TRICSI is a good candidate for pruning, the remaining issue is the externalisation of CONGRAD (or a deliverance of a portable version of CONGRAD which can be stored in project XLA/ALGOR).

CONGRAD:

- contrary to N1CG1, CONGRAD is not easily externalizable because its design makes it very linked to its purpose (use in the 4DVAR and in some other applications), the Lanczos algorithm (pure algorithmics) is imbricated with features linked to the application which is done with this minimizer (use of variables such as NCONF, LMPCGL, XMIN_RITZ, call of routines like SIM4D for example).
- a solution must be found about the possibility to have a portable version of CONGRAD which can be stored in project XLA/ALGOR.

Priorities for these actions:

- (P2) Portable version of CONGRAD (in XLA/ALGOR).

∗ Possible externalizations of tasks currently present in projects ARP/IFS and ALD: What I call externalization is to put a set of routines in a specific project, so that this project can work as an independent one (that means that it can be in theory used by a caller software which is not necessary ARPEGE/IFS). One calls a header and this header calls internal routines of this project which can mostly be seen as a blackbox.

Here is the sum-up of some proposals coming from different people:

- Externalize configuration 911 (put them in TFL).
- Externalize group EGGX (in XRD/IFSAUX).
- Externalize the semi-Lagrangian interpolators.
- Externalize the semi-Lagrangian halo calculation and more generally all routines dedicated to distributed memory communications.
- Externalize the observation operator.
- Externalize the vertical interpolator (used in FULL-POS for example).
- Externalize FULL-POS, or some parts of FULL-POS.
- Externalize configurations 9xx like 923, 931, 932 (but some of them may become obsolete in some years).
- Externalize the lateral boundary coupling for LAMs (externalisation of bi-periodicisation has already been done in CY34).
- Externalize the OASIS coupler.
- Externalize physics.
- Externalize some SURFEX applications which neeed to be modified at GMAP, separate them from the other MSE routines.

Some remarks about these proposals:

- Lateral boundary coupling for LAMs: there is a code reorganisation (not difficult to do) to do prior to externalization (all the Davies relaxation must be externalized, but all the semi-implicit terms calculations and addings must remain in project "arp").
- Observation operator, vertical interpolator of FULL-POS (also used in the observation operator, horizontal interpolator of FULL-POS: these tasks are not completely independent. About routines like POS, ENDPOS, PHYMFPOS, a better separation of dynamics from interpolations is desired.
- Semi-Lagrangian interpolator: routines doing interpolations (LAI... routines) or weights calculations (LASCAW+ELASCAW) are candidates for externalisation.

Priorities for externalizations:

- (P0): Externalize configuration 911 in project "TFL".
- (P0): Externalize group EGGX (move it in XRD/IFSAUX).
- (P1): Split MSE into two projects: SURFEX and MSE.
- (P1)+(P2): Other externalisation tasks.

6 Misnamed routines and decks.

Some work has been done in CY35. About the following cycles, naming of new routines entering these cycles must be checked.
New proposals of decks renaming have been listed in Appendix H.

∗ Priorities for renaming:

- (P0) for all renamings (must be done in an automatic cleaning cycle).

7 Misplaced variables.

7.1 Misplaced variables in project ARP/IFS:

∗ **Dynamical quantities of YOMCT0/NAMCT0:** Some "dynamical" variables currently in YOMCT0/NAMCT0 (for historical reasons, because they have been introduced at a period where YOMDYNA/NAMDYNA did not exist) could be moved into YOMDYNA/NAMDYNA.
It has been noticed that ECMWF is not in favour of such variable moving, especially for namelist variables.

∗ DM environment variables of NAMPAR0:

- It is desirable to create a new module YOMPAR0.
- The following variables are currently in YOMCT0 and it is desirable to move them in YOMPAR0: NPROC, NOUTPUT, NPRGPNS, NPRGPEW, NPRTRW, NPRTRV, NSPECRESMIN, LMPOFF, LMPDIAG, LCOUPLO4,
- The following variables are currently in YOMMP and it is desirable to move them in YOMPAR0: MP_TYPE, MBX_SIZE,
- It is desirable to move the following (YOMGSTATS/NAMPAR0)-variables in a new namelist NAMGSTATS (read in SUMPINI): NTRACE_STATS, LSTATS, LSTATSCPU, LSYNCSTATS, LDETAILED_STATS, LXML_STATS, LSTATS_MEM, NSTATS_MEM, LSTATS_ALLOC, LBARRIER_STATS, NPRINT_STATS. In SUMPINI, calculation of default values must be put in a well separated paragraph (no need to code a specific routine SUGSTATS).

∗ DM environment variables of NAMPAR1:

- It is desirable to create a new module YOMPAR1.
- The following variables are currently in YOMMP and it is desirable to move them in YOMPAR1: LSPLIT, LEQ_REGIONS, LOCKIO, NSLPAD, NAPSETS, NSTRIN, NSTROUT, NFLDIN, NFLDOUT, NCOMBFLEN, NINTYPE, NOUTTYPE, LSPLITOUT, NGATHOUT, NWRTOUT, NBLKOUT, LSYNC_SLCOM, LSYNC_TRANS.
- The following variables are currently in YOMOMPDIST and it is desirable to move them in YOMPAR1: LOMPDIST.
- (YOMMASK/NAMPAR1) variables: two possible actions: move them in a new namelist NAMMASK, or move them in YOMPAR1 (the first solution is easier to implement). Relevant variable: LSLONDEM.
- (YOMCPG/NAMPAR1) variable LCPG_SPLIT: move it in YOMPAR1 (and remove module YOMCPG).
- It is desirable to move the following (YOMIO/NAMPAR1)-variables in a new namelist NAMIO (read in SUMP0): LPPTSF, NPPBUFLEN, NPPFILES.

∗ Priorities for moving variables:

- (P1) Actions about NAMPAR0 variables.
- (P1) Actions about NAMPAR1 variables.

8 Consistency between module and namelist naming.

It would be desirable that a namelist variable computed in namelist NAM[XXX] be in module YOM[XXX] (or YEM[XXX], QA[XXX]) and computed in the set-up routine SU[XXX] (to be compliant with a recommendation given in document (IDCRT)). In practical this is not always the case. The constraint on the name of the set-up routine which reads namelist NAM[XXX] is not always easy to match. But we have to try to match at least consistency between namelist name and module name when they contain the same variables. Additionally to the YOMCT0 variables listed in the previous section, some inconsistencies between namelist name and module name are listed in Appendix E.

It has been noticed that ECMWF is not in favour of moving a too long list of variables from a namelist to another, but we can study at least a minimal set of easy changes of modules names (listed in paragraph "A first set of actions to be done" of Appendix E).

∗ **Priorities for ensuring name consistency between namelist and module:**

- (P0) Renaming modules or namelists, listed in paragraph "A first set of actions to be done" of Appendix E.
- (P2) Some other actions, if an agreement is found between the different partners.

9 Duplicata.

∗ **Duplicata of names.** In this topic we can find:

- D1/ two variables in two different modules of the same project, having the same name (but not necessary the same meaning).
- D2/ two variables in two different projects, having the same name (but not necessary the same meaning).
- D3/ a variable has the same name as a subroutine, in the same project.
- D4/ a variable has the same name as a subroutine, but in different projects.
- D5/ a variable has the same name as a subroutine of the scientific library or an intrinsic routine (example: local variable named ISMAX).
- D6/ two routines have the same name, but in different projects (example: MINIMA).
- D7/ two different variables of the same project (in different modules) have exactly the same meaning (ex: ROMEGA in yomcst.F90 and OMEGA in yhlconst.F90).

D1 should be avoided. It does not generate any problem of compilation if both versions of the variable are not used in the same routine, but if a code development leads to use both version in a same routine, that will oblige to change the name of one of them.

D2 is allowed.

D3 should be avoided.

D4 is not recommended, but it is difficult to completely avoid it: such occurrences are present in the code.

D5 should be avoided, but there probably remains such occurrences in the code (for example local variables named ISMAX).

D6 should be avoided if possible (there are occurrences of such duplicata in the code currently): it is desirable to ensure convergence between the two versions and to use only one of these versions in the future. If not possible the "duplicata" must be renamed (names keeping a common root).

D7 should be avoided because that causes useless complexification of the code.

These types of duplicata have not been checked in a comprehensive way (this task is very difficult to do without an adequate automatic procedure which remains to be written). We can notice some existing duplicata:

- D1/ A lot of occurrences have been found (see appendix G), in particuliar in the ECMWF radiation.
- D3/ For example: dummy variable named POS (in PNTERP for ex.) and routine POS.
- D5/ For example: local variables named ISMAX, ISMIN.
- About D6, we must notice that there are no duplicata left (contrary to some years ago) between ARP/IFS and ALD. Some renamings have been done in CY35T2, but all occurrences have not been completely removed (two versions of ISMIN for example, in XRD/IFSAUX and mse/internals; two versions of MINIMA, in ARP/IFS and SCT).
- About D7, this case is encountered for the constants used in the HIRLAM physics: some variables of module yhlconst.F90 (generally not DOCTOR compliant) have their counterpart in yomcst.F90, according to the following table of correspondance:

yhlconst.F90	yomcst.F90
PI	RPI
LATVAP	RLVTT
RAIR	RD
CPAIR	RCPD
CCPQ	equiv. to RCPV/RCPD - 1
EPSILO	???
GRAVIT	RG
TMELT	RTT
LATICE	RLMLT
RHOS	???
RHOWATER	???
SOLAR	RIO
STEBOL	RKBOL
CARMAN	no equivalent
REARTH	RA
OMEGA	ROMEGA
CPV	RCPV

EPSILO, RHOS, RHOWATER do not seem to have equivalents in yomcst.F90, but maybe they can be recomputed from yomcst.F90 constants or other modules reference variables (for example yomsta.F90).

The equivalent of CARMAN (Von Karman constant) is not defined in yomcst.F90, but in yomphy0.F90 (VKARMN), and also in yoevdf.F90 (RKAP) for ECMWF vertical diffusion.

∗ **Priorities to reduce duplicata of names:**

- (P0) Rename local variable ISMAX into INSMAX.
- (P0) Rename local variable ISMIN into INSMIN.
- (P0) Rename dummy variable POS into POSI (in routine PNTERP).
- (P0) Rename dummy variable POS into POSS (in routine QASTAT).
- (P0) Rename the XRD version of ISMIN and ISMAX (since it is used it has to be moved into another directory than "not_used"). Cf. renamings in Appendix H (must be done in an automatic cleaning cycle).
- (P0) Rename D1/-type duplicata (listed in Appendix G) if not already done in CY35T2.
- (P2) Use YOMCST constants instead of YHLCONST constants when possible in the HIRLAM physics (remove the YHLCONST versions of these constants for constants which are already in YOMCST).

10 ∗ Unsufficient use of the F90 shortness possibilies for code writing.

No comprehensive list of pieces of code which can be significantly shortened will be given in this paper, because this is a huge task impossible to do. But we can explore some solutions and give example of pieces of code which can be shortened.

10.1 ∗ Features which do not use enough the F90 possibilities, and what to do?

∗ **Loops involving arrays: array-syntax or use of DO loops.** About this topic, a compromise must be found between the shortness of the code and its optimisation properties. The conclusions about the optimisation properties lead to use array syntax only for simple operations (like initializing or copying whole arrays) and to use DO loops otherwise (cf. recommandation CCPT(10) of documentation (IDCRT)).

∗ **Reducing the number of dummy arguments, and better use of the derivated type variables.** There are still routines with an outstanding huge number of dummy arguments (in CY35T1, 168 routines have more than 50 dummy arguments), and an effort has to be done to match the CTRL(10) recommandation of documentation (IDCRT): avoid if possible to have more than 9 dummy arguments.

The use of derivated type variables can be a way to reduce the number of dummy arguments, and the data flow rewritings done these last years (implementation of the GMV and GFL data flow) has allowed to reduce the

number of dummy arguments in routines called in the dynamics. This way must go further, and some other sets of variables might be transformed into new derivated type variables or global buffers.

Some routines have useless dummy arguments or the obsolescent CDLOCK checking dummy argument (mainly in arp/phys_dmn for CDLOCK): desirable to be removed. Caution about the statement "Argument NOT used": argument is sometimes actually used. List of routines containing CDLOCK is provided in Appendix D.

∗ **Code modularisation.** Modularisation can reduce the size of the code in some cases (repetition of the same piece of code). There is currently a rather good level of modularisation in the code (at least for ARP/IFS and ALD) but it remains some very (too) long routines. Routines having more than 2000 lines (comments included) should be avoided.

There are also pieces of code where an excessive level of modularisation leads to lengthen the amount of code: for example routines, the content of which is just to call another one, should be avoided (examples: VTRANM which just calls VTRAN, EIGSOL which just calls RG).

∗ **Reducing code duplication.** Additionally to the "D6-type" code duplication we spoke in a previous part, there are routines doing nearly the same thing. We should recall that one rule of the project ARPEGE is that, for a given application, there is only one routine doing it.

10.2 ∗ Examples of pieces of code which can be shortened or where duplication could be reduced.

∗ **Reduction of the number of dummy arguments in the grid-point calculations under GP_MODEL.** This is the main part of the code where there remains a lot of routines with a very long list of dummy arguments (and this part of the code is frequently modified), even if slight improvements have been brought when implementing the GMV, GMVS and GFL dataflow. Some cleaning work has been done in CY36, but uncomplete. The other parts of the code (excepted some FULL-POS routines) are less critical (at least in ARP/IFS and ALD) and will not be examinated in detail.

- Pass globally GFL, GMV, GMVS, GFLT1, GMVT1, GMVT1S, PB1 and PB2 in grid-point calculations: work remaining to do in MF_PHYS and CPG_DIA.
- Pass globally POBB1 in OBSHOR, SLINT (+ TL and AD codes); pass globally the interpolated quantities of the observation horizontal interpolator in SLINT, MPOBSEQ_PACK (+ TL and AD codes).
- Some other groups of variables can be good candidates to be replaced by new derivated types variables or global buffers:
  - RIPI0, RIPI1, RIPI2 (YOMLEG), which can be merged into a new array RIPI.
  - RSLD1, RSLD2, RSLD3 (YOMLEG), which can be merged into a new array RSLD.
  - The output of semi-Lagrangian interpolations (buffer SLBUFF to be created, with new pointers in a new PTRSLBF, can be a way to do that).

  and also (some of these proposals still need discussions among the different partners).

  - geometry arrays of module YOMGC $->$ new variable YGEOGC.
  - computational sphere geometry arrays of module YOMLEG (at least RW, RMU, R1MU2, R1MUI, R1MUA, RSQM2, R1QM2, RACTHE, RLATIG and maybe also RLATI) $->$ new variable YGEOLEG.
  - computational sphere or plane projection geometry arrays of module YOMGEM (RCOLON, RSILON, RINDX, RINDY, RATATX, RATATH) $->$ YGEOGEM.
  - Interpolation weights computed in the semi-Lagrangian scheme: arrays KL0, KLH0, PDLO, PCLO, PCLOSLD, PDLAT, PCLA, PCLASLD, KLEV, PDVER, PVINTW, PVINTWSLD, PVINTWS, PVDERW, PHVW must be put in a derivated type structure. This derivated type structure can also be used for half level counterparts or trajectory counterparts of these arrays.
  - ZCOSCO, ZSINCO, ZSINLA, ZCOPHI (semi-Lagrangian scheme), which can be merged into a new array or derivated type structure. Same kind of work to do on ZCOSCO5, ZSINCO5, ZSINLA5, ZCOPHI5.
  - ZLON, ZLAT, ZQX, ZQY (semi-Lagrangian scheme), which can be merged into a new array or derivated type structure. Same kind of work to do on ZLON5, ZLAT5, ZQX5, ZQY5. Use naming ZQX, ZQY (or PQX, PQY) everywhere (abandon namings ZR, ZS, PR, PS, PO, PQ).
  - The intermediate t and $t - \delta t$ quantities computed by CPG_GP. Several derivated type structures could be defined, for example:
    - Hydrostatic pressure variables and output of GPXYB, GPGRXYB at time t (including PRE0, PRE0L, PRE0M, PRE0F, PDELP0, PRDELP0, PLNPR0, PALPH0), and also at time $t - \Delta t$.

    - Specific LVERCOR quantities: PVC.. arrays.
    - Specific LRWSDLR or LRWSDLW quantities: PWDL.. arrays.
    - Lagrangian adiabatic tendencies: PTND.. arrays.
    - Upper-air quantities: for example PLHU0, PLHV0, PHI0, PHIF0, P3DIVG.
    - Surface quantities: for example PQS, PDBBC.
  - All the output fluxes of the physics, which can become attributes of a derivated type new array GDF (for grid-point diabatic fluxes) or of several derivated type new arrays for each kind of parameterization. For example, instead of having a separate array PFHPCL (ZFHPCL) for the liquid water convective condensation enthalpy flux, we can have the attribute GDF%FHPCL of the array GDF. GDF should be passed as dummy argument to MF_PHYS, APLPAR, CPG_DIA. The individual physical parameterisations will keep the individual fluxes.
  - arrays of YOMFOUTC containing transmission coefficients in Fourier space.
  - arrays containing transmission coefficients in grid-point space (under MF_PHYS + TL + AD).

∗ **Sets of routines doing roughly the same thing.** Some work has been done in CY36 to merge routines doing roughly the same thing. But some work remains to do.

- First I have noticed several routines doing the inversion of a set of tridiagonal linear systems, without pre-computing a LU decomposition in the set-up (tridia.F90, sgtsl.F, which are now in the new project XLA, and hltridiag.F90, mse/internals/tridiag_ground_1d.mnh). Additionally to that, there are also inversions by (time-dependent) tridiagonal linear systems in some other parts of the code, for example in arp/phys_ec/cubidiag.F90 (which has TL and AD codes).

  Duplication of code must be reduced. For example, moprj.F90 and moprjad.F90 must call TRIDIA instead of SGTSL.
- Directory "arp/parallel" contains useless duplication; for example we can mention:
  - ISNDGPF+IRCVGPF does the same thing as DISGRID.
  - OSNDGPF+ORCVGPF does the same thing as DIWRGRID.
  - RDCSET, PHCSET are nearly identical and can be easily merged.
  - SLRSET, RDRSET, PHRSET are nearly identical and can be easily merged.
  - GATHERBDY can be probably significantly shortened by calling twice DIWRGRID (or can simply be replaced by two calls to DIWRGRID).
- In directory "arp/pp_obs":
  - BOB does roughly the same thing as PPQ; PPQ can be used instead with minor modifications.
  - PPRH does the same thing as GPRH+PPQ: same remark valid for TL and AD codes.
  - A partial reunification between PPGEOP and PPGEOP_OLD and PPUV and PPUV_OLD, and their TL and AD counterparts, can be done to avoid redundant codes (this is more difficult to do for PPT and I suggest to keep PPT_OLD which is too different from PPT). It would be better to in-line the LOLDPP code in the new PP.. routines and to have LOLDPP vs .NOT. LOLDPP code only in parts where code is different (and to remove the PP..._OLD.. decks). For example most parts of PPGEOP and PPGEOP_OLD are identical.

    Additionally it is not recommended to use key LECMWF inside these routines (use another key set-up according to LECMWF instead).

∗ **Rationalisation in the physics packages.** Even if some convergence has been tried in order to reduce the number of physical parameterizations (radiation for example), we notice that there are currently six sets of physical parameterizations:

- METEO-FRANCE physical parameterizations.
- AROME physical parameterizations.
- ALARO physical parameterizations.
- ARPEGE/CLIMAT physical parameterizations.
- ECMWF physical parameterizations.
- HIRLAM physical parameterizations.

Some packages of simplified parameterizations must be added to this list.

The convergence between physical parameterizations should go further and the number of routines doing physical parameterizations should more decrease, and this question must remain steered by people working on physical parameterizations. More generally, the considerations on physics interfaces developed in this note do not take into account the discussions held elsewhere on the convergence of physics (for example convergence meeting of 24-25 Sept 2008).

∗ **Rationalisation in the horizontal interpolators.** There are several horizontal interpolators we can sum-up as follows:

- The interpolator used in the semi-Lagrangian scheme (LAI... routines), also used in the observation horizontal interpolator under SLINT.
- The FULL-POS horizontal interpolator (FPINT... routines).
- The horizontal interpolator used in some C9xx configurations (arp/c9xx/INTER... and ald/c9xx/EINTER... routines).
- The horizontal interpolator used in the ECMWF physics to pass from refined to coarse grid (or the reverse operation) when using coarse resolution physics, in particuliar in the radiation scheme (routine arp/phys_ec/LCUBINT under RADINT).
- The horizontal interpolator used to go from the high resolution trajectory towards the low resolution increments in the ECMWF 4D-VAR (routines arp/utility/GRID_...).
- The horizontal interpolator (bilinear interpolations) in SUHIFCE (under INTERP_OBS) used in the observation pre-processing at ECMWF.

There is no action planned about that for the time being, but we could maybe benefit from an externalised "multi-purpose" horizontal interpolator to use less different pieces of code for these different applications.

∗ **Calculation of the RHS of temperature equation.** The same calculation is done at several locations of the code (LATTEX, CPEULDYN, CPDYDDH). It is desirable to do this calculation once (the best place for that is in CPG_GP). TL and AD codes must be updated too.

This kind of work is more difficult to extend to momentum equation for different reasons (curvature terms, different treatments of Coriolis term): unification of the codes present under LATTEX, CPEULDYN, CPDYDDH and LAVENT is not convenient to do and for the time being no action is planned.

∗ **Priorities to reduce duplicata and to shorten the size of the code:**

- (P0) Reduce the number of dummy arguments in the SL interpolator: interpolated quantities (LAPINEA, LAPINEB); same work also for TL and AD codes.
- (P0): Merge the arrays RIPI.. and RSLD..
- (P0) Replace the calls of ISNDGPF+IRCVGPF by calls to DISGRID.
- (P0) Replace the calls of OSNDGPF+ORCVGPF by calls to DIWRGRID.
- (P0) Merge RDRSET, PHRSET with SLRSET.
- (P0) Merge RDCSET with PHCSET.
- (P0) Put the RHS of T equation in one location in the code (CPG_GP) instead of 3 currently; same work also for TL and AD codes.
- (P0) Eliminate the occurrences of CDLOCK in the MF physics.
- (P0) Use PPQ instead of BOB; remove routine BOB.
- (P0) Use GPRH+PPQ instead of PPRH; remove routine PPRH; same work also for TL and AD codes.
- (P0) Partial reunification between PPGEOP and PPGEOP_OLD, PPUV and PPUV_OLD, and their TL and AD counterparts.
- (P0) Replace the multiple calls to CPUTQY, CPUTQYS, CPUTQY_AROME by one in MF_PHYS. The same work must be done in the TL and AD codes (CPUTQYSTL, CPUTQYSAD).
- (P0) Reduce the number of dummy arguments in VPOS+POS, and also in ENDPOS, PHYMFPOS.
- (P0) Reduce the number of dummy arguments in the COBS+COBSLAG observation horizontal interpolator; this action must be combined with another one allowing to make easier inclusion of new surface variables in this interpolator.
- (P1): MOPRJ and MOPRJAD (NMI): replace the calls to SGTSL by calls to TRIDIA. Remove xla/external/linalg/sgtsl.F after that.
- (P0+P1+P2) Eliminate useless dummy arguments.
- (P1) Rewrite GATHERBDY by using two calls of DIWRGRID (if possible).
- (P1 + P2) Reduce the number of dummy arguments in the physics (at least MF_PHYS and the APLPAR.. routines): part one (no creation of new derivated types, but pass at one time GFL, GMV, GMVS, GFLT1, GMVT1, GMVT1S, and also PB1 and PB2 if possible).
- (P1) or (P2) create new derivated types for geometry arrays of YOMGC, YOMLEG and YOMGEM.
- (P2) create new derivated types or global array for transmission coefficients used in the simplified physics.
- (P2) Reduce the number of dummy arguments in the dynamics: part two (creation of new derivated types).
- (P2) Reduce the number of dummy arguments in the physics: part two (creation of new derivated types, for example for the output fluxes of the physics).
- (P1 + P2) Rationalisation in the physics packages.

11 Ensure consistency of variables names.

In this topic I see at least two points, but it is possible that other problematic items exist for some other groups of variables.

∗ **Post-processing of surface variables.** The roots used in variables names of the post-processed surface variables and those used in variables names of the surface buffers are not always the same ones: it is desirable to change the name of the variables used in the post-processing (FULL-POS) to make it compliant with the one of the surface buffers (example for the percentage of land: root LAN for post-processing buffers but CONT for surface buffers). Parts 2.6 and 2.7 of the documentation about FULL-POS (IDFPOS) give an extensive list of discrepancies between the FULL-POS names and the surface names, and a specific separate technical memorandum has to be writen to describe all the replacements to do to make the names consistent.

∗ **GFL variables.**

- There are scories of using root SV for extra-GFL (in FULL-POS, nudging, DDH, AROME physics), root EXT should be used instead.
- TFP_GFL (YOMAFN) is mis-named because it contains only extra-GFL variables: it must be renamed into TFP_EXT.
- The root R is used for rain, but that can make a confusion with the air constant, it would be recommended to use RR instead. About this topic there is a lot of work to replace all the R by RR (a lot of global and local grid-point arrays).

A specific separate technical memorandum has been writen to describe all the replacements to do (ASCII file).

∗ **More generally.** A "standard" list of name roots has to be done for all meteorological variables in the future (for example "LAN" for fraction of land, "2T" for 2 meter temperature): this is a huge task, and for the time being this work is not yet completed.

∗ **Priorities to ensure consistency of variables names.**

- (P0) FULL-POS: change SV into EXT in the names of variables containing extra-GFL; update comments in order to mention "extra GFL" and not "passive scalars" where relevant.
- (P0) Change CONT into LAN in the name of surface variables containing percentage of land (for example in class VARSF).
- (P0) Rename TFP_GFL into TFP_EXT.
- (P1) DDH, nudging, AROME physics: change SV into EXT in the names of variables containing extra-GFL; update comments in order to mention "extra GFL" and not "passive scalars" where relevant.
- (P2) Make consistency between the post-processing names and the variable names for surface, using preferently a root easily understandable in English.
- (P2) Change R into RR in the names of variables containing rain.

12 Ensure consistency between the AD and TL codes and their direct code counterpart.

For some pieces of code, the AD and TL code still matches with old versions of the direct code. This is the case for some pieces of the semi-Lagrangian scheme. The comprehensive list of mis-matchings is not provided, but in the semi-Lagrangian scheme we find at least:

- LATTEX_DNT_AD and LATTEX_DNT5 match with old versions of LATTEX_DNT (but LATTEX_DNT5 will be removed in a pruning action).
- LATTESTL and LATTESAD match with old versions of LATTES.

Updating the TL and AD codes will be required.

∗ **Priorities to update TL and AD code:**

- (P0) update LATTEX_DNT_AD.
- (P0) update LATTESTL and LATTESAD.

13 Make the code norm-compliant.

See (IDCRT) for details about these norms. Most of non-compliancies can be restaured by automatic cycle cleanings, and it would be desirable to plan such an automatic cleaning cycle before the end of 2010.

Most of the norm violations (in projects ARP/IFS and ALD only) are listed in a compilation by "gmkpack". According to a mail of Ryad El Khatib, norm violations for the last cycles were:

```
cycle 28    : 4712 violations
cycle 29    : 4423 violations
cycle 30    : 4811 violations
cycle 31    : 6928 violations
cycle 32    : 7479 violations
cycle 33    : 6861 violations
cycle 35    : 5551 violations
cycle 35t1  : 5810 violations, 5375 violations after cleaning of PPFIDH
cycle 35t2  : 5262 violations
```

In CY35T2, a more comprehensive status of these violations shows:

```
* 1290 CCPT(04) violations [useless declarations of local variables or module variables]
* 1881 CTRL violations:
  - 408 CTRL(35) violations ["Macro DOC is old-fashioned"]
  - 1223 CTRL(10) violations [routines having more than 9 dummy arguments,
    including 153 routines having more than 50 dummy arguments]
  - 187 CTRL(03) violations [Alternate returns to be avoided]
  - 37 CTRL(27) violations [Calls to MPL_ routines should have the argument
    CDSTRING='caller...' where caller is the name of the calling routine]
  - 26 CTRL(20) violations [improper arguments passed to DR_HOOK]
* 1783 NORM violations:
  - 711 NORM(13) violations [statement GOTO]
  - 916 NORM(09) violations [USE without ONLY]
  - 113 NORM(10) violations [non-DOCTOR variable prefixes for local variables or dummy arguments].
    Non-DOCTOR variable prefixes for module variables do not appear in this list.
  -   6 NORM(03) violations [IMPLICIT NONE missing]
  -  29 NORM(16) violations [statement EQUIVALENCE]
  -   2 NORM(07) violations [Integers and reals should be declared with explicit kind]
  -   3 NORM(21) violations [Continuation lines should start with &]
  -   1 NORM(02) violation [TAB characters not allowed]
* 205 PRES violations:
  - 204 PRES(07) violations [Empty lines should be left empty, remove !]
  -   1 PRES(22) violation [Break lines with comma at end of line]
* 103 occurrences of "Unusual kind value used" (JPRH)
```

And, additionally:

```
*  15 occurrences of END DO, END IF, END WHERE (where ENDDO, ENDIF, ENDWHERE is expected)
```

We can see that some violations are easy to remove (useless declarations, "Macro DOC", non-DOCTOR variables, empty lines starting by "!"), some other ones need a significant work to be removed.

In particuliar, we can see that there is still an outstanding surprising number of GOTO or "GO TO" left (principally in the screening and the observation treatment, but not only). Such an antiquated F66 feature may make the code very difficult to read. Some of them can be easily removed by using EXIT or "DO WHILE(condition)" statements; some other ones need more thinking and work to be removed. A high priority of cleaning must be put about GOTO statements which match the three following conditions:

- not obsolescent code.
- GOTO rather easy to replace, the replacement of which will make code easier to read, without increase of its cost.
- routines frequently modified, used by several persons.

Additionally, no GOTO statement must be introduced in new routines.
The list of routines containing GOTO statements is provided in the appendix F (projects ARP/IFS, ALD, SUR, TFL and TAL only).

Additional remarks can be done:

- INTENT statement with mention "UNDETERMINED INTENT": it is generally possible to determine the right intent and these "UNDETERMINED INTENT" must be progressively fixed (roughly 500 occurrences).

- There are missing some INTENT statements in TFL and TAL, and probably in some F90 compliant routines of project XLA/ALGOR.
- Cosmetic cleanings to improve the header and the declarations/includes: cleaning has been done in arp/adiab and ald/adiab and must be progressively extended to other directories. To sum-up: readable presentation of declarations and includes with separators between each topic, right place of the first CALL DR_HOOK statement, removal of obsolete and often mis-placed comments like "INPUT", "OUTPUT", "LOCAL" or "DUMMY", removal of (obsolete) "ifdef DOC" statements. See for example arp/adiab/laidli.F90 to see a proper readable presentation.

∗ **Priorities to reduce norms violations.**

- (P0) Reduce to under 5000 violations in CY36.
- (P0) Reduce to under 4000 violations in CY37.
- (P0+P1) Remove the GOTO statements with "high priority" (see above).
- (P1) Reduce to under 3000 violations in CY38.
- (P1) Complete the removal of "ifdef DOC" statements.

∗ **Priorities for other cleanings.**

- (P0) Restore missing INTENT statements in projects TFL, TAL and F90 routines of XLA/ALGOR: directory "external".
- (P0) Finish cleaning in the header+declarations for ALADIN.
- (P0) Replace the statement "Argument NOT used" by proper INTENT statement (at least in projects ARP/IFS and ALD) for dummy arguments which are actually used.
- (P0+P1) Replace "UNDETERMINED INTENT" by proper INTENT statement in projects ARP/IFS and ALD.
- (P0+P1+P2) Continue cleaning in the header+declarations for ARP/IFS (at least 2 directories per cycle).
- (P1) Restore missing INTENT statements in projects TFL, TAL and F90 routines of XLA/ALGOR: directories "module" and "internal".

14 Make available new tools for code management.

∗ **Proposal of new tools for code management.** The new tools which can very helpful are the following ones (for some of them that can be new macros implemented in CLEARCASE).

- T01: Procedure to count the total number of routines (project by project). Available in summer 2008 (command cc_count).
- T02: Procedure to count the total number of lines (project by project). Available in summer 2008 (command cc_count).
- T03: Procedure to count the total number of active lines (project by project). Not available.
- T04: When a subroutine is in a module, procedure giving the module-deck where is the routine. Not available.
- T05: When a subroutine is in a module, procedure editing the module-deck where is the routine. Not available.
- T06: For a given subroutine, procedure giving all the callers of this subroutine. Partially available in the source navigator but incomplete.
- T07: For a given module, procedure giving all the routines using this module. Partially available in the source navigator but incomplete.
- T08: For a given variable, procedure giving all the routines using this variable. Partially available in the source navigator but incomplete.
- T09: For a given project, procedure giving all the projects containing the callers and the users (upstream dependencies). Not available.
- T10: For a given project, procedure giving all the projects containing the callees and the used modules (downstream dependencies). Not available.
- T11: Procedure giving all the occurrences of forbidden dependencies between projects (for example ARP/IFS routine calling a TAL routine, TFL routine calling an ARP/IFS routine). Not available.
- T12: List of routines which have several versions (two routines having the same name, but in different directories or different projects). Not available.

- T13: List of module names (variables) or routines names used for at least two purposes (for example variable which has the same name of a routine or of an intrinsic routine of the scientific library, variable declared in two different modules). Not available.
- T14: Procedure giving the list of routines having more than [nnn] dummy arguments, when [nnn] is an argument of this procedure. Not available.

  The norm checker used in the compilator at least gives all the routines with more than 9 dummy arguments (projects ARP/IFS and ALD).
- T15: Procedure giving the list of routines which are not called anywhere. Not available.
- T16: Procedure giving the list of module variables which are not used anywhere. Not available.

∗ **Additional remarks.** The question might be raised whether only source code repository administrators should have the authorization to create/remove directories or to move or delete decks under the CLEARCASE/PERFORCE arborescence, or whether this facility shall be opened to more developers. For example, under CLEARCASE, the following macros can be interesting to move or delete decks or to do externalization, without waiting that one administrator makes that; I can list for example the following possible commands:

- cc_move: moves an existing deck from a directory to another one, with keeping the history (for example when doing cc_diff).
- cc_rmfile: removes an obsolete deck.
- cc_mkdir: makes a new directory.
- cc_rmdir: removes an obsolete (empty) directory.

∗ **Priorities to make available new tools for code management.**

- (P0) Make available tools T03, T04, T05, T06, T15, T16.
- (P0) Restore the missing dependencies in the source navigator.
- (P1+P2) Make available the other tools.

15 Conclusion and timing of cleaning.

The present document will be presented at the ARPEGE/IFS coordination meeting of the 25/06/2009 for discussions. It must also be further disseminated among the ECMWF and MF staff, as well as the Aladin and Hirlam consortia.

The proposed calendar of actions does not take into account other constraints which may interfere, such as computer migration or the SRNWP Interoperability project (all being actions which may divert manpower from some code maintenance actions).

Completing the (P0) actions for CY37 and (P1) actions for CY38 does not mean that the cleanings will stop after this date. Some remaining (P2) work will be done, maybe other actions will be identified, and more generally each cycle must have a minimal amount of cleaning dispatched among a significant number of persons. Some other documents like the present one will be written at requested time.

16 References:

- (IDCRT) El Khatib, R., 2003: Coding standards for ARPEGE/IFS/ALADIN. Internal note, 42pp.
- (IDARC) Yessad, K., 2008: Library architecture and history of the technical aspects in ARPEGE/IFS, ALADIN and AROME in the cycle 35 of ARPEGE/IFS. Internal note.
- (IDFPOS) Yessad, K., 2008: FULL-POS in the cycle 35 of ARPEGE/IFS (internal note).
- Yessad, K., 2009: New presentation for routine POS, version 2 (internal note = technical documentation to rewrite POS and some other vertical interpolator routines, 11pp).

Appendix A: list of insufficiently commented declaration modules.

Here is the list of insufficiently commented declaration modules in project "ARP/IFS" (status CY35T1).

- gems_profiles.F90: missing comments for attributes.
- grg_photolysis.F90: missing comments for attributes.
- grib_header_def.F90: missing comments for attributes.
- gridpoint_buffers.F90: missing comments for attributes.
- iostream.F90: missing comments for attributes and for some variables.
- parrrtm.F90: missing comments (and not-DOCTOR "NG.." variables).
- parrtm1d.F90: missing comments.
- parsrtm.F90: missing comments (and not-DOCTOR "NG.." variables).
- parssmi.F90: missing comments.
- partrmm.F90: missing comments.
- random_streams.F90: missing comments.
- spectral_columns.F90: missing comments for attributes.
- yoeaermap.F90: missing comments.
- yoeaersrc.F90: missing some comments.
- yoeneur.F90: missing comments.
- yoerrt....F90 modules: missing comments.
- yoesrta16.F90 to yoesrta29.F90: missing some comments.
- yoe_mcica.F90: missing comments.
- yoe_uvrad.F90: missing comments.
- yomangm.F90: missing comments.
- yomfger.F90: missing comments.
- yomgetmini.F90: missing comments.
- yomglobs.F90: missing comments.
- yomjg.F90: missing some comments (for example in type definition).
- yomjq.F90: missing some comments (for example in type definition).
- yommts.F90: missing some comments (variables NSAT, NTYPE, LMTSCL).
- yompaddh.F90: missing comments for NGPUMASK and NGLALIST (used in DDH).
- yomrain_lb.F90: missing comments.
- yomstack.F90: missing comments.
- yom_ssmi.F90: missing comments.

Here is the list of insufficiently commented declaration modules in project "XRD/IFSAUX" (status CY35T1).

- yomlcz.F90: missing some comments.

Here is the list of insufficiently commented declaration modules in project "TFL".

- tpm_dim.F90: missing some comments.
- tpm_fft.F90: missing comments.

Here is the list of insufficiently commented declaration modules in project "TAL".

- tpmald_dim.F90: missing comments.
- tpmald_distr.F90: missing comments.
- tpmald_fft.F90: missing comments.
- tpmald_fields.F90: missing comments.
- tpmald_geo.F90: missing comments.
- tpmald_tcdis.F90: missing comments.

Appendix B: list of physical-dynamical interface routines (status CY35T2).

∗ **How has this list been done.** We have tried to list all the routines which are not purely dynamical ones nor purely physical ones. Two level of physical-dynamical interface routines have been listed.

In the first level we find routines which do:

- calculation of input quantities which are used only in the physics (for example moisture convergence, solar angle).
- interface routine to call physics.
- conversion of the physics outputs (fluxes or tendencies of physics prognostic variables) into tendencies of dynamics prognostic variables.
- optionally additional diagnostics.
- temporal advance of surface and soil variables.

This is the case of routines like mf_phys.F90, ec_phys_drv.F90, raddrv.F90. Most of these routines are allowed to call adiab/gp... routines (in particuliar GPPRE, GPPREF, GPPREH, GPXYB).

In the second level we find routines which call individual parameterisations, but these routines must remain portable and do not contain model-oriented structure like GFL, GMV. This is the case of routines like APLPAR, APL_AROME, CALLPAR, RADPAR. These routines are not allowed to call adiab/gp... routines, and in particuliar they are not allowed to call GPPRE, GPPREF, GPPREH, GPXYB. These routines are not allowed to call level 1 routines.

Routines remaining in phys_dmn, phys_ec and phys_radi after this moving are not allowed to call adiab/gp... routines and to contain model-oriented structure like GFL, GMV (they must remain portable into other models).

∗ **List:** Surely:

- arp/adiab/cpg_pt.F90 (level 1)
- arp/adiab/cpmvvs.F90 (level 1)
- arp/phys_dmn/cpozo.F90 (level 1, like for cputqy.F90)
- arp/adiab/cpqtuv.F90 (level 1, like for cputqy.F90)
- arp/phys_ec/cpspe.F90 (level 1, like for cptend.F90)
- arp/adiab/cptend.F90 (level 1)
- arp/adiab/cptend_new.F90 (level 1)
- arp/adiab/cptends.F90 (level 1)
- arp/adiab/cptendsm.F90 (+TL,AD) (level 1)
- arp/adiab/cptendsmat.F90 (level 1)
- arp/adiab/cputqy_arome.F90 (level 1)
- arp/adiab/cputqy.F90 (level 1)
- arp/adiab/cputqys.F90 (+TL,AD) (level 1)
- arp/adiab/cpwts.F90 (level 1, like for cptends.F90)
- arp/adiab/gpaddslphy.F90 (level 1)
- arp/adiab/gpmktend.F90 (+AD) (level 1)
- arp/adiab/postphy.F90 (level 1)
- arp/phys_dmn/apl2phy.F90 (level 2)
- arp/phys_dmn/apl_arome.F90 (level 2)
- arp/phys_dmn/aplpar.F90 (level 2)
- arp/phys_dmn/aplpars.F90 (+TL,AD) (level 2)
- arp/phys_dmn/aplparsadt.F90 (level 2)
- arp/phys_dmn/hl_aplpar.F90 (level 2)

- arp/phys_dmn/initaplpar.F90 (level 2, like for aplpar.F90)
- arp/phys_dmn/mf_phys.F90 (+TL,AD) (level 1)
- arp/phys_dmn/mf_phys_prep.F90 (level 1)
- arp/phys_ec/callpar.F90 (+TL,AD) (level 2)
- arp/phys_ec/ec_phys.F90 (+TL,AD) (level 1)
- arp/phys_ec/ec_physg.F90 (level 1)
- arp/phys_ec/ec_phys_drv.F90 (level 1)
- arp/phys_ec/ec_phys_lslphy.F90 (level 1)
- arp/phys_ec/gpmktend.F90 (level 1)
- arp/phys_ec/radcfg.F90 (level 1, because calls GP.. routines: like for mf_phys.F90)
- arp/phys_ec/raddiag.F90 (level 2)
- arp/phys_ec/raddrv.F90 (level 1)
- arp/phys_ec/radpar.F90 (level 2, like for APLPAR, CALLPAR)
- arp/phys_ec/sltend.F90 (level 2, because called by CALLPAR)
- arp/phys_ec/sltend1.F90 (level 2, because called by CALLPAR)
- arp/phys_ec/sltend2.F90 (level 1, because calls GPPRE)

And maybe also (to be confirmed):

- arp/phys_dmn/aplpassh.F90 (level 1): must be in the same directory as cpqsol.F90 (complementarity between these two routines to be studied).
- arp/adiab/cpedia.F90 (level 1, like postphy.F90); place may also be in arp/dia.
- arp/adiab/cpfhpfs.F90: place may be in "level 2" of physics-dynamics interface, or in phys_dmn. Current place in "adiab" is improper.
- arp/adiab/cpphinp.F90 (+TL,AD) (level 1)
- arp/adiab/cppsolan.F90 (level 1)
- arp/adiab/cpqsol.F90 (level 1)
- arp/adiab/gpino3ch.F90 (level 1)
- arp/adiab/gpinozst.F90 (level 1)
- arp/phys_dmn/cpchet.F90 (level 1)
- arp/phys_dmn/ecr1d.F90 (level 1, or arp/dia)
- arp/phys_dmn/ecr2df.F90 (level 1, or arp/dia)
- arp/phys_dmn/ecr2dv.F90 (level 1, or arp/dia)
- arp/phys_dmn/ecradfr15.F90 (level 1)
- arp/phys_dmn/ecrpnebh.F90 (level 1, or arp/dia)
- arp/phys_dmn/mts_phys.F90 (place in phys_dmn is questionable, maybe this deck must go in pp_obs).
- arp/phys_dmn/nomfi.F90 (level 1, or arp/dia)
- arp/phys_dmn/profilechet.F90 (level 1, or arp/dia)
- arp/phys_dmn/qngcor.F90 (level 2, because called under APLPAR)
- arp/phys_ec/radint.F90 (level 1, or separate directory for radiation interpolator)
- arp/phys_ec/radintg.F90 (level 1, or separate directory for radiation interpolator)
- arp/phys_dmn/writechet.F90 (level 1, or arp/dia)
- arp/phys_dmn/writephysio.F90 (level 1, or arp/dia)
- arp/phys_dmn/writeprofile.F90 (level 1, or arp/dia)
- arp/phys_ec/phys_nl.F90 (level 1, or arp/onedvar)
- arp/phys_ec/phys_tl.F90 (level 1, or arp/onedvar)
- arp/phys_ec/phys_ad.F90 (level 1, or arp/onedvar)

Appendix C: list of routines without any general comment describing action (status CY35T2).

The following projects have been examinated: ARP/IFS, ALD, BIP, XRD/IFSAUX, XLA/ALGOR, TFL, TAL.

∗ Project ARP/IFS: ECMWF physics (principally radiation).

- arp/phys_ec/aer_phy1.F90 (insufficient commenting)
- arp/phys_ec/cuadjtq.F90 (insufficient commenting)
- arp/phys_ec/gwprofil.F90 (insufficient commenting: Stress profile?)
- arp/phys_ec/gwsetup.F90 (insufficient commenting)
- arp/phys_ec/gwsetupad.F90 (insufficient commenting)
- arp/phys_ec/gwsetuptl.F90 (insufficient commenting)
- arp/phys_radi/rrtm_cmbgb[2 to 16].F90 (insufficient commenting => missing comments saying that extensive explanations are in rrtm_cmbgb1.F90)
- arp/phys_radi/rrtm_gasabs1a_140gp.F90
- arp/phys_radi/rrtm_init_140gp.F90
- arp/phys_radi/rrtm_kgb[1 to 16].F90
- arp/phys_radi/rrtm_taumol[2 to 16].F90 (insufficient commenting => missing comments saying that extensive explanations are in rrtm_taumol1.F90)
- arp/phys_radi/srtm_cldprop.F90
- arp/phys_radi/srtm_cmbgb[17 to 29].F90 (insufficient commenting => missing comments saying that extensive explanations are in srtm_cmbgb16.F90)
- arp/phys_radi/srtm_kgb[16 to 29].F90
- arp/phys_radi/srtm_srtm_224gp.F90 (insufficient commenting)
- arp/phys_radi/srtm_srtm_224gp_mcica.F90 (insufficient commenting)
- arp/phys_radi/srtm_taumol[16 to 29].F90
- arp/phys_radi/su_c[11,12,22]clim.F90
- arp/phys_radi/su_ccl4clim.F90
- arp/phys_radi/su_ch4clim.F90
- arp/phys_radi/su_co2clim.F90
- arp/phys_radi/su_ozoclim.F90
- arp/phys_radi/surrtab.F90
- arp/phys_radi/surrtftr.F90
- arp/phys_radi/surrtpk.F90
- arp/phys_radi/surrtrf.F90

∗ Project ARP/IFS: other topics.

- arp/obs_preproc/extrap.F90
- arp/obs_preproc/extrapad.F90
- arp/obs_preproc/extraptl.F90
- arp/op_obs/tropopause.F90
- arp/op_obs/writenn.F90 (insufficient commenting)
- arp/phys_dmn/mts_phys.F90
- arp/pp_obs/heapsort.F90
- arp/var/add_modbias_ad.F90
- arp/var/add_modbias_tl.F90
- arp/var/chavar....F90 (insufficient commenting)
- arp/var/convddr.F90
- arp/var/evjcdfi.F90
- arp/var/get_jbvcoord_coeffs.F90
- arp/var/supavarc.F90
- arp/var/suprepjcdfi.F90

* Project XRD/IFSAUX.

- cma/oldcma_get_address.F (maybe useless?)
- cma/oldcma_set_address.F (maybe useless?)
- most of "eclite" files.
- grib_io/grib_set.F
- grib_mf/gsbyte_mf.F
- most of "misc" files.
- parallel/cmpl_binding.F90
- parallel/coml_binding.F90
- parallel/jfh_bind.F90
- programs/datefa.F
- programs/gribdiff.F
- programs/gribtrace.F
- most of "support" files.
- most of "utilities" files.

* Project XLA/ALGOR.

- most of "internal/lanczos" routines.
- internal/linalg/cdiv.F
- internal/linalg/mxva.F
- internal/linalg/sgemmx.F (insufficient commenting)
- internal/minim/ctcab_1dv.F
- internal/minim/ctonb_1dv.F
- internal/minim/dpseuclid.F90
- internal/minim/ecube_1dv.F
- internal/minim/euclid_1dv.F
- internal/minim/euclid.F
- internal/minim/m1qn3a....F routines (insufficient commenting)

* Project TFL.

- module/abort_trans_mod.F90
- module/fspgl_int_mod.F90
- module/set_resol_mod.F90
- module/setup_dims_mod.F90
- module/setup_geom_mod.F90
- module/spnormc_mod.F90
- module/spnorm_ctl_mod.F90
- module/spnormd_mod.F90
- module/sufft_mod.F90
- programs/aatestprog.F90
- programs/test_adjoint.F90

* Project TAL.

- module/cpl_int_mod.F90
- module/eset_resol_mod.F90
- module/esetup_dims_mod.F90
- module/esetup_geom_mod.F90
- module/espnormc_mod.F90
- module/espnorm_ctl_mod.F90
- module/espnormd_mod.F90
- module/suefft_mod.F90
- programs/aatestprog.F90
- programs/test_adjoint.F90

Appendix D: list of routines with dummy argument CDLOCK.

* **arp/phys_dmn.**
  - acacon.F90
  - acbl89.F90
  - accdev.F90
  - ac_cloud_model.F90
  - acclph.F90
  - accoefk.F90
  - accoll.F90
  - acconv.F90, acconvtl.F90, acconvad.F90
  - accvimpd.F90
  - accvimpdgy.F90
  - accvimp.F90
  - accvimpgy.F90
  - accvimp_v3.F90
  - accvud.F90
  - acdifoz.F90
  - acdifsp.F90, acdifsptl.F90, acdifspad.F90
  - acdifspadt.F90
  - acdifus.F90
  - acdifv1.F90
  - acdifv2.F90
  - acdnshf.F90
  - acdrac.F90
  - acdrag.F90
  - acdragl.F90, acdragltl.F90, acdraglad.F90
  - acdrme.F90, acdrmetl.F90, acdrmead.F90
  - acdro.F90
  - acdrov.F90
  - acevmel.F90
  - acevolet.F90
  - acfluso.F90
  - achmt.F90, achmttl.F90, achmtad.F90
  - achmtls.F90
  - acmixlentm.F90
  - acmixlenz.F90
  - acmodo.F90
  - acnebc.F90
  - acnebcond.F90
  - acnebn.F90
  - acnebr.F90
  - acnpart.F90
  - acntcls.F90, acntclstl.F90, acntclsad.F90
  - acozone.F90
  - acpblh.F90
  - acpblhtm.F90
  - acpluie.F90

- acpluis.F90
- acptke.F90
- acqwlsr.F90, acqwlsrtl.F90, acqwlsrad.F90
- acradcoef.F90
- acradin.F90
- acrads.F90, acradstl.F90, acradsad.F90
- acralu.F90
- acraneb.F90
- acsol.F90
- acsolw.F90
- actcpnf.F90
- actke.F90
- actqsat.F90
- actqsats.F90
- actsec.F90, actsectl.F90, actsecad.F90
- acturb.F90
- acupd.F90
- acupm.F90
- acupu.F90
- acveg.F90
- acvppkf.F90
- apl2phy.F90
- aplmini.F90
- aplmphys.F90
- aplpar.F90
- aplpars.F90, aplparstl.F90, aplparsad.F90
- aplparsadt.F90
- arp_ground_param.F90
- hl_aplpar.F90
- hlturb.F90

All the other occurrences have been removed in CY35T2.

Appendix E: inconsistencies between namelist naming and module naming (status CY35T2).

* NAC.., NAD.., NAI.. and NAL.. namelists (CANARI).

- NACOBS: some variables are in QAPABO, not in QACOBS.

* NAE.. namelists (ECMWF physics).

- NAEAER: variables are spread among YOEAERSRC, YOEAERATM, YOEDBUG, YOEAERMAP.
- NAEPHY: some variables are spread among YOEWCOU, YOMCOAPHY, not in YOEPHY.
- NAERAD: some variables are spread in YOMPRAD, YOERDI, YOE_UVRAD.

* NAM.. namelists.

- NAM_CANAPE: variables are in QAREF.
- Variables of namclddet.h are in YOMIASI.
- NAMCT0 variables are spread among several modules.
- NAMDDH: logical variables are in YOMLDDH; integer variables are in YOMMDDH.
- NAMDYN: some variables are not in YOMDYN (they may be for example in YOMCVER if finite elements, YOMMASS if mass corrector, YOMTRAJ for high resolution trajectory, YOMSEP, etc...)
- NAMFPC: RENTR is in YOMFPC and also in YOEVDF.
- NAMFPSC2 and NAMFPSC2_DEP variables are in YOMFPSC2.
- NAMGFL: variables are in YOM_YGFL and not in YOMGFL.
- NAMGRIB: variables are in YOMGRB.
- NAMHLOPT: variables are in YHLOPTION.
- NAMJBCODES: variables are in YOMJG.
- NAMJG: some variables are not in YOMJG (they are in YOMWAVELET or YOMCOSJB).
- NAMJO: variables are in YOMCOSJO.
- NAMMODERR: some variables are in YOMMODEL_ERROR, some others in YOMJQ.
- NAMNMI: variables are in YOMNMIA.
- NAMNN: variables are in YOMNNE.
- NAMNPROF: variables are in YOMECTAB.
- NAMOPH: variables are in YOMOP.
- NAMPAR0: variables are spread among YOMGSTATS, YOMCT0, YOMMP.
- NAMPAR1: variables are spread among YOMIO, YOMMP, YOMOMPDIST, YOMMASK, YOMCPG.
- NAMPARAR: variables are in YOMARAR.
- NAMRCF: variables are spread among YOMRES and YOMMASS.
- NAMSATS: variables are in YOMTVRAD.
- NAMSCEN: RI0 is in YOMCST, not in YOMSCEN.
- NAMTLEVOL: some variables are in YOEPHLI.
- NAMTRAJP: variables are spread among YOPHNC, YOMIOP, YOEPHLI, YOMNCL.
- NAMVAR: some variables are spread among YOMTRAJ, YOMJQ, YOMVCGL.
- NAMVWRK: variables are in YOMTRSL.

* NEM.. namelists (LAM models).

- NEMVAR: variables are in YEMVARGP.

* Other inconsistencies.

- nammatchup.h (ODB) contains a namelist called NAM_MATCHUP.
- namsort.h (ODB) contains a namelist called NAM_SORT.
- namstdin.h (ODB) contains a namelist called NAM_STDIN.
- namwt.h (ODB) contains a namelist called NAM_WT.
- namclddet.h (ARP) contains a namelist called CLOUD_DETECT_COEFFS.
- nammwave.h: some variables are also in namonedvar.h
- name NAMTESTVAR is used in deck namtestvar.h but is also redefined in testvar_mix.F90 with a different content.

∗ A first set of actions to be done.

- rename YOMIASI into YOMCLDDET.
- rename YOMGRB into YOMGRIB (or rename NAMGRIB into NAMGRB).
- rename YHLOPTION into YHLOPT.
- rename YOMCOSJO into YOMJO (or rename NAMJO into NAMCOSJO).
- rename YOMMODEL_ERROR into YOMMODERR.
- rename YOMNMIA into YOMNMI.
- rename YOMNNE into YOMNN and SUNNE into SUNN (or rename NAMNN into NAMNNE).
- rename YOMOP into YOMOPH.
- rename YOMARAR into YOMPARAR.
- rename YOMTVRAD into YOMSATS.
- rename YOMTRSL into YOMVWRK.
- rename YEMVARGP into YEMVAR and SUEVARGP into SUEVAR (or rename NEMVAR into NEMVARGP).
- rename CLOUD_DETECT_COEFFS into NAMCLDDET in namclddet.h.
- rename NAM_CANAPE into NACREF, QAREF into QACREF (CANARI).
- rename NAMGFL into NAM_YGFL.

and also:

- ODB namelist: rename deck nammatchup.h into nam_matchup.h.
- ODB namelist: rename deck namsort.h into nam_sort.h.
- ODB namelist: rename deck namstdin.h into nam_stdin.h.
- ODB namelist: rename deck namwt.h into nam_wt.h.

and later:

- YOMTRAJ variables of NAMVAR must be moved in a new namelist element NAMTRAJ.
- Some other actions not detailed for the time being.

Appendix F: routines containing GOTO statements (status CY35T2).

Projects other than ARP/IFS, ALD, SUR, TFL, TAL have not been examinated. No GOTO or GO TO statement has been reported in projects SUR, TFL, TAL.

∗ Project ARP/IFS.

- arp/c9xx: cseaice.F90, incli2.F90, incli5.F90, incli6.F90, incli7.F90, inclir.F90, inter10.F90, intrv2.F90.
- arp/canari: caapar.F90, cabine.F90, cainsu.F90, calico.F90, calina.F90, calver.F90, canada.F90, canali.F90, cancer.F90, caneva.F90, capdgu.F90, carnak.F90, caspia.F90, caviar.F90.
- arp/control: cprep1.F90, cprep5.F90.
- arp/dfi: optfilb.F90, remez.F90 (both candidates for pruning).
- arp/dia: sumddh.F90.
- arp/fullpos: sufpdom.F90, sufpdyn.F90.
- arp/nmi: moprjm.F90, moprjmad.F90.
- arp/obs_error: supererr.F90.
- arp/obs_preproc: addoer.F90, airepin.F90, awprfin.F90, biascor.F90, biascor_odb.F90, chdeve2.F90, chdeve.F90, chdsta.F90, dribuin.F90, dupli.F90, dupli_no_sq.F90, dwlin.F90, ers1if.F90, ewprfin.F90, fgchk.F90, fgwnd.F90, flgdco.F90, flgdse.F90, geosrin.F90, iniersca.F90, lndsyin.F90, mertsin.F90, metarin.F90, minima.F90, movpl.F90, movpl_no_sq.F90, new_rs_trh_bias.F90, new_thinner.F90, new_thinner_no_sq.F90, new_thinn.F90, ngedeve2.F90, ngedeve.F90, ngereve.F90, ngersta.F90, nscatin.F90, obadat.F90, obinssp.F90, p_4_sort.F90, paobin.F90, pgpsin.F90, pilotin.F90, pnterp.F90, post_prsta.F90, post_thinner.F90, pre_thinner.F90, pre_thinn_radar.F90, prlmchk.F90, prsta.F90, ptendcor.F90, rad1cin.F90, redgl.F90, redgl_no_sq.F90, redml.F90, redml_no_sq.F90, redprof.F90, redrp1.F90, redrp1_no_sq.F90, redrp.F90, redrp_no_sq.F90, redsm.F90, redsm_no_sq.F90, redts.F90, reini.F90, rejmv.F90, reo3sin.F90, repra.F90, repsel.F90, s0towind.F90, satamin.F90, satemis.F90, satobin.F90, scaqc.F90, selec.F90, settc.F90, shipin.F90, sonde_country_db_match.F90, ssmimas.F90, thinner.F90, thinner_no_sq.F90, thinn.F90, thinn_radar.F90, tovshris.F90, tovslris.F90, upecma.F90, verco.F90.
- arp/op_obs: exheiz2p.F90, grg_ak_ad.F90, grg_ak_op.F90, grg_ak_tl.F90, hradpad.F90, hradp_ml_ad.F90, hradp_ml_tl.F90, hradptl.F90, nox2no2ad.F90, nox2no2.F90, nox2no2tl.F90, preint2d.F90, preint2dtl.F90, preint.F90, preintsad.F90, preinttl.F90, tropopause.F90.
- arp/phys_dmn: nomfi.F90, raddiag15.F90, radint15.F90, suecradi15.F90.
- arp/phys_ec: cuadjtqsad.F90, cuadjtqstl.F90, cudlfsn.F90, cuinin2.F90, cuinin.F90, legtriv.F90, radintg.F90.
- arp/phys_radi: rrtm_setcoef_140gp.F90, rrtm_taumol2.F90, srtm_setcoef.F90, suecrad.F90, suecradl.F90.
- arp/pp_obs: poaero.F90, ppobsaad.F90, ppobsa.F90, ppobsasad.F90, ppobsas.F90, ppobsastl.F90, ppobsatl.F90, ppobsaza.F90, ppobsaz.F90, ppobsaztl.F90, ppreq.F90.
- arp/programs: merge_varbc.F90.
- arp/sekf: sekf_costf.F90, sekf_gain.F90, sekf_matinv.F90, sm_ekf_main.F90.
- arp/setup: suarg.F90, suspecb.F90, suspecg1.F90.
- arp/sinvect: balanced_reduction.F90.
- arp/var: ecset.F90, sureo3.F90.

∗ Project ALD.

- ald/c9xx: eincli2.F90, eincli5.F90, eincli6.F90, eincli7.F90, einter10.F90.
- ald/programs: blend.F90.
- ald/setup: suebicu.F90.
- ald/utility: eggx.F90.

Appendix G: Obsolete variables in arp/module.

At least the following variables have been found to be useless (basis: CY35T2). Variables in italics are only declared in a declaration module and not referenced elsewhere.

* ...mix.F90:

- GRG_PHOTOLYSIS_MIX: *TAUA1*, *TAUB1*, all *A1..* and *B1..* variables, *C7_TAU*, *A2_O1D*, *B2_O1D*, *A3_O1D*, *B3_O1D*, *C4_O1D*, *C5_O1D*, *TJ_O1D*, *A2_H2O2*, *B2_H2O2*, *A3_H2O2*, *B3_H2O2*, *C4_H2O2*, *C5_H2O2*, *C6_H2O2*, *TJ_H2O2*, *A2_HNO3*, *B2_HNO3*, *A3_HNO3*, *B3_HNO3*, *C4_HNO3*, *C5_HNO3*, *C6_HNO3*, *TJ_HNO3*, *A2_HNO4*, *B2_HNO4*, *A3_HNO4*, *B3_HNO4*, *C4_HNO4*, *C5_HNO4*, *A2_N2O5*, *B2_N2O5*, *A3_N2O5*, *B3_N2O5*, *C4_N2O5*, *C5_N2O5*, *C6_N2O5*, *TJ_N2O5*, *A3_COH2*, *B3_COH2*, *C4_COH2*, *C5_COH2*, *C6_COH2*, *TJ_COH2*, *A3_CHOH*, *B3_CHOH*, *C4_CHOH*, *C5_CHOH*, *C6_CHOH*, *TJ_CHOH*, *A2_CH3OOH*, *B2_CH3OOH*, *A3_CH3OOH*, *B3_CH3OOH*, *C4_CH3OOH*, *C5_CH3OOH*, *C6_CH3OOH*, *C6_NO2O*, *C7_NO2O*, *TJ_NO2O*, *C7_NOO2*, *TJ_NOO2*, *A2_PAN*, *B2_PAN*, *A3_PAN*, *B3_PAN*, *C4_PAN*, *C5_PAN*, *C6_PAN*, *TJ_PAN*, *A2_CH3ONO2*, *B2_CH3ONO2*, *A3_CH3ONO2*, *B3_CH3ONO2*, *C4_CH3ONO2*, *C5_CH3ONO2*, *A2_CH3CHO*, *B2_CH3CHO*, *A3_CH3CHO*, *B3_CH3CHO*, *C4_CH3CHO*, *C5_CH3CHO*, *TP_CH3CHO*.

* par...F90:

- PARCMA: JPMXDEPL (only allocates useless variables).
- PARRRTM:
  Variables NG.. have not DOCTOR norm compliant names: to be renamed with prefix JP (JPG1, JPG2, etc).

Inventory not yet finished.

* per...F90:

- PERDIM: JPN1SP (only allocates useless variables).

* ptr...F90: no useless variable found.

* qa...F90:

- QABOIT: *MBOXK*; USE PARDIMO and USE QAPABO are useless.
- QACOSS: *XPOHV1*.
- QACOST: *QPOHUR*.
- QADORE: *MPNPV*, *MIPNVP*, *MOPT*.
- QAKEKI: NDBCVS, NDBCVPR, NDBCVCQ, NDBCVCQM, NDBCVACQ, NDBCVAU, NDBPAS, NDBPAPR, NDBPACQM, NCQVCO, NCQVPC, NCQVPI, NCQVIN.
- QAPAVU: *JPPYQ*.

* yem...F90:

- YEMCT0: RLSDX, NLAM, NECRILSG.
- YEMDIM: NS3DFD, NS2DFD, NS3DFI, NS2DFI, NS3DYX, NS2DYX, NFTM, NDGLSUR, NSECPL (directly setup NSECPLG to KSEFRE/4 in SUEDIM), NAS1B, NAS1E.
- YEMDYN: HDRJBC.
- YEMGEO: *LEGGNEWC*.
- YEMGT3B: *NLENGT3B*, *GT3GRD*, NLATGT3, NLONGT3, NSP3UBC, SP3UBC (allocated but not used).
- YEMJK: FJKNORM1 (allocated but not used).
- YEMLAP: *NCPL2MG*, *NCPL4MG*, NESRN (allocated but not used).
- YEMSPBC: *SPUBBC*, *SPVBBC*.

* yhl...F90:

- YHLCOND: STPEVP, U00MAX, HU00, AECON, CONAE, CFREEZ, HCUNRM, HTAUCU, HP0, HVTERM, HVSNOW, HKMELT, SQVSNO, TANVIL, BFEFF, HKEVAP, HKAP, HECDR, HDLDCP, HLDCP, COALCU, COALST, ELOTCI, ASNOW, BSNOW, SNOREF.
- YHLCONST: RHOWATER, SOLAR, CARMAN.
- YHLTURB: CTVAR, CQVAR, CHT1, CHT2.

∗ **yoe...F90:**

- YOEAERATM: *NINIDAY*, NDD1, NSS1, RMFMIN, LAERGBUD, LAERPRNT.
- YOEAERC: *RSINCV*, RSINCT, RSINCSO4.
- YOEAERMAP: NAERMAP, RAERMAP.
- YOEAEROP: ALF_FA, ASY_BC, ASY_DD, ASY_FA, ASY_OM, ASY_SS, ASY_SU, OMG_FA.
  ASY_SU also appears in YOEAERSU (this version is also useless).
  ALF_SU also appears in YOEAERSU.
  OMG_SU also appears in YOEAERSU (this version is useless).
- YOEAERSNK: RFRGAS, NBRH, RRHMAX, RRHO_SS, RSSGROW, RMMD_SS, RRHO_DD, RVSEDSIC, RVSEDLND, RVSEDLIC.
  Name RRHTAB also appears in YOEAERSU (must be renamed).
- YOEAERSRC: NBINAER, LAEREXTR, RSSLIM, RGELAV, RGEMUV, RDGLAV, RDGMUV, RCLONV, RSLONV, RDCLONV, RDSLONV, RLATVOL, RLONVOL.
  Some JK... integers have not DOCTOR compliant names.
- YOEAERSU: ASY_SU, OMG_SU.
- YOECLD: RRHM.
- YOECLDP: RENTRTU, RENTRRA, RSATQ, RASMICE, RBSMICE.
- YOECLOP: RYFWCD, RYFWCE, RASWCA to RASWCF, REBCUG to REBCUJ, REFFIB, RTIW, RRIW.
  Names RYFWCA to RYFWCF, REBCUA to REBCUF, RASWCA to RASWCF, REBCUG to REBCUJ, REFFIA, REFFIB, RTIW, RRIW also appear in YOESW (useful YOECLOP versions must be renamed).
- YOECLOP550: *RSC55*, *RSD55*, *RSE55*, *RSF55*, *RFB55*, *RFC55*, *RFD55*.
  RSA55, RSB55, RFA55 are used but not set-up anywhere.
- YOECND: CEVAPCU, REPFLS.
- YOECUMF: RHCDD.
- YOECUMF2: LMFPEN2.
- YOEGWD: GVCRIT, GHMAX, GRAHILO, GSIGCR.
- YOELW: NIPD2, NTR, RNTNU,
  Name TREF also appears in YOERRTRF and YOESRTWN (must be renamed).
  Name NG1 also appears in YOERRTA1 (parameter, must be renamed with JP prefix) and in PARRRTM (must be renamed JPG1 to be DOCTOR compliant).
- YOEPHY: *NPHPROMA*, LESHCV.
- YOERAD: NLW, LERADHS.
- YOERDI: *RHVAR*, *RFVAR*, *RINCSOL*.
- YOERIP: RIP0M.
- YOERRTA1 to YOERRTA16:
  Local parameter variables NG1 to NG16 must be renamed into JPNG1 to JPNG16.
  Some K... integers have not DOCTOR compliant names.
  Some identical variable names are re-used in all these modules (example ABSA).
- YOERRTFTR:
  Name NGM also appears in YOESRTWN (to be renamed).
  Name WT also appears in YOESRTWN and in YOMOPF (to be renamed).
- YOERRTO1 to YOERRTO16:
  Local parameter variables NO1 to NO16 must be renamed into JPNO1 to JPNO16.
  Some K... integers have not DOCTOR compliant names.
  Some identical variable names are re-used in all these modules (example FRACREFBO).
- YOERRTRWT:
  Name RWGT also appears in YOESRTWN (to be renamed).
- YOERRTWN: TOTPLK16.
  Names NG, NSPA, NSPB also appear in YOESRTWN (to be renamed, for those which are not useless).

- YOESRTA16 to YOESRTA29:
  Local parameter variables NG16 to NG29 must be renamed into JPNG16 to JPNG29.
  Some K... integers and some other parameter N.. integers have not DOCTOR compliant names.
  Some identical variable names are re-used in all these modules (example ABSA).
- YOESRTCOP: all RSSSI.. arrays.
- YOESRTOP: *ABSCOICE*, *ABSCOLIQ*, EXTICE2, SSAICE2, ASYICE2, ABSCLD1.
- YOESW: *RTWEIGHT*, *RWEIGS*, *RWEIGV*, RSUSHE, RSUSHF, RSUSHH, RSUSHK, RSUSHA, RSUSHG, RSUSHFA, RSUSHC, RSUSHD, REFFIB, RTIW, RRIW, RROMA, RROMB, RRASY, RHSRA, RHSRB, RHSRC, RHSRD, RHSRE, RHSRF, RHSRTA, RHSRTB, RWEIGHT, NMPSRTM, NTYPS.
- YOETHF:
  Some identical variable names are re-used in other modules (example YOMANCS, YOMNUMC): renaming must be planned.
- YOEVDF: RPAR, RPARSRF.
- YOEVDFS: RCHETC.
- YOEWCOU: NRESUM.
- YOE_UVRAD: RFD0, RFD1, RFD2, RFD3, RFCOZO.

∗ **yom...F90:**

- YOMAERD15: *RAESC15*, *RAESS15*, *RAELC15*, *RAELS15*, *RAEUC15*, *RAEUS15*, *RAEDC15*, *RAEDS15*.
- YOMARAR: *MSFU*, *MSFV*, *MSFRV*, *MSFTH*, *MUM*, *MVM*, *MTM*, *MPABSM*, *MPSURF*, *MZZ*, *MRHODREF*, *MSFSV*, *NBUPROC*, NJBUDG1, NJBUDG2, JPAROBUD.
- YOMARG: *UBETA*.
- YOMCHCOD: CIDLEV, CPRESCD (module to be deleted after elimination of useless code in sucodes.F90).
- YOMCHEV: CR2EVENT, CRBLEVEN, CH2EVENT, CHDBLEVE.
- YOMCLOP15: REBCUG15, REBCUH15, RTIW15, RRIW15.
- YOMCMA: *NMXSATS*, *NDEPLISL*, *NDEPLIST*, *NOSIMUL*, *LICMAMER*, *LOCMAMER*, NMXGICML, NMXGRCML, NUSDUPD.
- YOMCMBDY: *NCMFIBL*, *NCMFRBL*, *NCMRDFL*, *NCMDBLE*, *NCMPRL*, *NCMOMN*, *NCMOMF*, *NCMFOE*, *NCMOER*, *NCMRER*, *NCMPER*, *NCMFGE*, *NCMFGC1*, *NCMFGC2*,
  NCMRBLN, *NCMRBIN*, *NCMRBRN*, *NCMIOM0*, *NCMIFC1*, *NCMIFC2*, *NCMIOMN*, *NCMIOMSN*, *NCMRBVC*, *NCMRBPIO*, *NCMRBOE*,
  NCMTBLN, *NCMTBLNI*, *NCMTELID*, *NCMPILID*,
  NCMPITII, *NCMPILTI*, *NCMPILNI*, *NCM1DVC*, *NCMTORDE*, *NCMTORBA*, *NCM1DVCA*, *NCMTORDEA*, *NCMSSRB*, *NCMS1DVC*, *NCMSSRDE*, *NCMSSCCF*, *NCMSSCBW*, *NCMSSANP*, *NCMSCBAA*, *NCMSCBIA*, *NCMSCKP*, *NCMSCRES*, *NCMSCDIS*, *NCMRO3HEI*,
  NCMTETII, NCMTELTI, NCMTELNI,
  NCMBPFOC, NCMBVFOC, NCMBHSBP, NCMBHSOC, NCMBQSBP, NCMBQSOC, NCMBSHBP, NCMBSHOC, NCMBPABP, NCMBPAOC, NCMBUABP, NCMBUAOC,
  NTLIDOC, NTMWLBP, NTMWLOC, NTTROBP, NTTROOC, NTSURBP, NTSUROC, NTSWLBP, NTSWLOC, NTSTLBP, NTSTLOC, NPLIDOC, NPMWLBP, NPMWLOC, NPTROBP, NPTROOC, NPSWLOC, NPSTLBP, NPSTLOC.
  The following variables must be checked (names also appear in ODB routines but do not seem to refer to the YOMCMBDY version), they are maybe useless in YOMCMBDY: NCMFBL, NCMVNM, NCMVCO, NCMPPP, NCMVAR, NCMSYPC, NCMTORB.
- YOMCMDDR: NCD1NRAD, NCD1PV1 to NCD1PV7, NCD1EN2 to NCD1EN15, NCD1VAT, NCMD1HO, NCMD1MIO, NCMD1SO, NCMD1YO, NCMD1MOO, NCMD1DO
- YOMCMHDR: *NCMFHL*, *NCMFIHL*, *NCMFCHL*, *NCMFRHL*, *NCMRFL*, *NCMRBLE*, *NCMBOX*, *NCMSTD*, *NCMSUBTYPE*, *NCMSID2* to *NCMSID8*, *NCMMOR*, *NCMTLA*, *NCMTLO*,
  all the *NCM..* variables from *NCMOHL* to *NCMVS4*,
  all the *NCM1..* variables from *NCM1DIT* to *NCM11DFI*,
  all the *NCM1DA..* and *NCM1DB..* variables from *NCM1DBP* to *NCM1DACA*,
  all the *NCM1DV..* variables from *NCM1DVP* to *NCM1DVO2*,
  all the *NCM1BCP..* variables from *NCM1BCP1* to *NCM1BCP16*,

NCMSSIA,

all the *NCM..* variables from *NCMSCLA* to *NCMZAN*,

all the *NCME1..* variables from *NCME1LA* to *NCME1BT2*,

all the *NCME2..* variables from *NCME2LA* to *NCME2BT2*,

all the *NCME3..* variables from *NCME3LA* to *NCME3BT2*,

all the *NCM..* variables from *NCM11DSS* to *NCMRO3NL*,

NYYOFF, NMMOFF, NDDOFF, NHHOFF, NMIOFF, NSSOFF, NLTFLOC, NLNFLOC, NDTFLOC, NTMFLOC, NALFLOC, NHMSBP, NHMSOC, NQCSBP, NQCSOC, NORFBP, NORFOC, NHFSBP, NHFSOC,

NPIA4BP, NPIA4OC.

NCMRST is used (in chrsta.F90) but not set-up anywhere.

NCMREV1 is used (in chreve.F90) but not set-up anywhere.

- YOMCOCTP: *NMBPLTSQ*.
- YOMCOM: ALATN1, ALATN2, ALATN3, ALATN4, ALATS1, ALATS2, ALATS3, ALATS4, GAMAT, GAMAD, HMEL_MIN, TRAFLX.

  Some other variables are also present in YOMCPL and only the YOMCPL version is used: FCSTSU, FCSTSV, FCCHAS, FCRSOS, FCCHLL, FCCHLN, FCCHSS: remove them in YOMCOM.
- YOMCOU: *NPICP*, *NPIOC*.
- YOMCPL: FCEVAP.
- YOMCST: RMRA.
- YOMCT0: NPRTRM.

  Remark: variables LSFCFLX, REXTSHF, REXTLHF are used in VDFMAIN but not set-up anywhere.
- YOMCT1: N1MASS.
- YOMCT3: LGPQINSP.
- YOMCVA: *NVA3GP*.
- YOMDB: *L_DEGREES*, *L_ACTIVE*, NPOOLS_CCMA, NPOOLS_ECMASCR.

  Additionally, some of YOMDB variables appear only in ODB routines.
- YOMDIM: *NPROME*, *NPROMP*, *NPROMV*, NFPPYX, NFPPYE, NMTCMAX.

  Remark: some of YOMDIM variables also appear as attribute names in YOMPRAD and YOM_PHYS_GRID; it would be probably more convenient to rename these duplicata with a specific prefix or a specific appendix.
- YOMDIMO: *NOB2DO*, *NLNPF*, NOBTOTM, NOBTOVMX, NDZLAY.
- YOMDPHY: *NTSL*, *NCHAC*, *NCHIN*, *NSIRA*, NTVG, NLOA, NLOE, NVTEND.

  Remark: name NVTEND appear in YOMDPHY and YOMSLPHY. Only the YOMSLPHY version is currently used.
- YOMDYN: SIRUB, S2ETA.

  Name LDRY_ECMWF is not DOCTOR compliant, to be renamed into LRDRY_ECMWF.
- YOMDYNCORE: *DRAGX*, *DRAGY*.

  Name LDYNCORE is not DOCTOR compliant, to be renamed into LRDYNCORE.
- YOMECTAB: *NTSL2DO*, *NTSL2DOOFF*, MPROMOBS, NOOPOT.
- YOMERR: *LFINERCO*, *ERR_PAOB*, LERRTHI, LERRPWC.

  Remark: LPERERCO is used (in supererr.F90) but not set-up anywhere.
- YOMERSCA: *MBITPAD*, XPHI0, DTETA.
- YOMFPC: C1FP3DFS, C1FP3DFP, C1FP3DFT, C1FP3DFV, C1FPDOM, MFP3DYN, MFP2DYN, NFPLNPR, LASQ.
- YOMGEM: NESTAGP, NBEEGP, NBNEGP.
- YOMGLOBS: *MTOVS_TYPE*, *MSCATT_TYPE*, *MNONTOVS_TYPE*, *NTYPRECV*, *NTYPSENDAD*, *NGLOBKEY*, MAXSENDOBS, MAXRECVOBS, MTOTRECVOBS, MAXSENDOBSAD, MAXRECVOBSAD, NLOCSEND, NLOCRECVAD.
- YOMGPSK: *GPSUABUF*, *GPSSUBUF*.
- YOMGRB: *NCALVAL*, *MLOCGRB*, *MTOTENS*, *MENSFNB*, *MBITSSH*, *MBITSGG*, *MJDIAG*, *MJDOMAI*, *MJITER*, MGRBS3, MGRBS2, NGRBGP2, NGRBGP3, NBITSGG.

  Remark: MSEC0 is used (in pregrbenc.F90) but not set-up anywhere.

- YOMIOS: NPCKFGP5.
- YOMJG: *MAPMOCV*, LEVSREDNMC, NLEVREDNMC, EPSBALM, EPSBALW.
- YOMLAP: *NATCM0*, NTCINDX.
- YOMLDDH: LHDEF.
- YOMLIM: NLAT90, NLAT20, NCENYEAL, NSYCCHA, RLAT90, RLON360, RLON0, RCONLIM, RERRPP, RPPADJ, RPPLMT, RPPDIF.
- YOMLIMB: *MSENSOR_MLSSQ*.
  Some PARAMETER integers have not DOCTOR compliant names.
- YOMLVLY: LOBTHLAY, LOBPWLAY, NOPWLAY, BPWLAY, TPWLAY.
- YOMLW15: NIPD15, NIPD215, NTR15, RNTNU15.
- YOMMKODB: LNOEDGES, LSURPR, LZTCONS, LSINOSOL, LSCATTHI, LTOTSCTH, LNOPROC, LMKCMARPLRUN, NOSORTSL.
- YOMMP: LOCKIO, NFLDOUT, NINTYPE, NGATHOUT, NBLKOUT.
- YOMMPG: NPOSCP, NPOSTP.
- YOMMWAVE: RLIMIT_TS,RLIMIT_TP,
  Some variable names are also present in YOMONEDVAR: RLIMIT_SD, RLIMIT_LAT, N_SATSENS. Must be renamed in YOMMWAVE.
  Some variable names are also present in YOM_SSMI: OB_FROM,OB_JOBS,OB_JROF_GP,GP_COUNT,OB_COUNT. Must be renamed in YOM_SSMI.
  Some LD... logicals have not DOCTOR compliant names.
- YOMNMCOD: *NMXPPCD*, *NMXLID*, NMXSBI1, NMXSBI2, NSMI2I2, NMXSMVV, NSMVV, NMXSMWW, NSMWW, NMXSMXX, NSMXX, NMXSMYY, NSMYY, NMXSMAA, NSMAA, NMXSMBB, NSMBB, NMXSMCC, NSMCC, NINTESC, NREALSC, NCHARSC.
- YOMNNE: *NOUTPUTS*, *NPATS*, *OUTPRED*, *GINP*, LNNLEARN, LNNJAC, LNNOUT, LNNM1QN3, NO_OF_EPOCHS, SELECTINP, NU, NU2, LISS, OUT_MOY, OUT_STD, EIG_VEC, EIG_VAL; and probably also INP_MOY, INP_STD.
  Some INP... integers have not DOCTOR compliant names.
  Some JAC... integers have not DOCTOR compliant names.
- YOMOBS: *LSCASUR*, LOBCLN, LSLRT2, LSLRRH2, LHDREJ, LHDRW10, LHDRT2, LHDRRH2.
  Some LDRIBUS... logicals have not DOCTOR compliant names.
- YOMOERR: RHERRY.
- YOMONEDVAR: RCOR_R, RLIMIT_RR, RLIMIT_LS.
  Some IFAIL... integers have not DOCTOR compliant names.
- YOMOP: *CNAMDB*, *CIDDB*, *CNAMFS*, *CNAMFG*, *CNAMFD*, *NADCIO*, *NAROIO*, *NBSOUT*, CFNTLSH, CFNFUN, CFNSNC, CFNIT0, CFNPT0, CFNOPA, CFNDEP, CFNTC.
- YOMOZO: TOZ1D, TOZ3DBG,
- YOMPHY: *LFGELS*, LHUNEG, LECTBR, LECTLIM, LAJUCV, LBCCOND, NPRAG, NLEND, LEDMF.
  Some LD... logicals have not DOCTOR compliant names.
- YOMPHY0: *ACG*, *UNTIER*, HUCOE2, AGREKE, GFRIC, GRRINTE, UETEPS, UPRECLP, ARSC2, ARSCT, AJ1MEPS, AJ1PEPS, NAJITER, HCMIN, GWBFAUT, RAUITN, RAUITX, RAUIUSTE, RLMLH2, RLMLH3.
- YOMPHY1: RZHGLA, RZHMER.
- YOMPHY2: RIPBLC.
- YOMPHY3: QLIMI, QLIP0.
- YOMPPC: *LMSLP*, MPLEV.
  LMOVIE can be also removed (just switches a printing on, no other action done).
- YOMRAD15:
  Remark: LRADLB15 is used (in radint15.F90) but not set-up anywhere.
- YOMRDU15: NIMP15, NOUT15, REPSCW15.
- YOMRES: *LRSTEP*, *NUMRFS*.
- YOMRPLIM: RP500.

- YOMSCREE: *ND1DQS*, *CA1DQS*, NGLSHI, NGLAIR, NGLDRI, NGLTOV, NGLGEO, NGLSSM, NGLSAT, NGLSAM, NCDYTL, NCDSTL, NCDATL, NCDDTL, NCDTTL, NCDGTL, NCDITL, NCDWTL, NCDMTL, NA1DQS.
- YOMSEKF: FKF_TENT_1, FKF_TENT_2, FKF_TENT_3, FKF_TENT_4, VSM_PERT_RES,
- YOMSPNRM: *AVNRMSP*, *AVNRMVOR*, *AVNRMT*, *AVNRMQ*, *AVNRMKE*, *AVNRMSPD*, *AVNRMSVD*.
- YOMSSG: TSSGB.
  This module can be removed.
- YOMSSMI: LSSMITHI.
- YOMSTRE: RPCLOU,RCLOUD.
- YOMSW15: RSWCE15,RSWCP15.
- YOMTAG: MTAGTIDE, MTAGGLOBSR, MTAGBDY, MTAGOZON, MTAGREADVEC.
- YOMTHLIM: NSCIND1, NSCIND2.
- YOMTIT: *NSTAIT15B*.
- YOMTNH: *NSTANH95B*.
- YOMTOPH: *NTCOEFE*, *ETCOEFE*.
- YOMTPHY: NSTAGT95B, NSTAGTT5B, NSTAGTS5B.
- YOMTRAJ: *MTYPE_SLAG_TRAJ*, *MTYPE_PHYS_TRAJ*, MTRAJSFC, MTRAJCST.
- YOMTRSL: NSTAGTSLB.
- YOMTVRAD: *NHULAY*, *MMETHOD_WVCL*, *MMETHOD_IR*, *MMETHOD_VIS*, *MMETHOD_WVMIX*, *MMETHOD_COM_SPEC*, *MGEOSATN_METEOSAT*, *MGEOSATN_GMS*, *MGEOSATN_GOES*, *MGEOSATN_MSG*, *MGEOSATN_INDSAT*, *MGEOSATN_FY*, *MGEOSATN_XXXXXX*, *MGEOSATN_MODIS*.
- YOMVAR: *NPCKFANE*, *NEXPBANE*, LAEROD.
- YOMVAREPS: NRES_MA.
- YOM_OAS: *NBID*, *LPMODEL*, *CPOASIS*.

∗ **yop...F90:**

- YOPHLC: LCZDEB.

Appendix H: Decks to be moved or renamed.

This appendix does not take account of the namelist or module renamings listed in Appendix E to ensure name consistency between modules and namelists.

H1: Routines to be moved without renaming into existing directories.

Routine	Current dir	New proposal of dir
expbesu.F90	arp/op_obs	arp/pp_obs
expbesutl.F90	arp/op_obs	arp/pp_obs
expbesuad.F90	arp/op_obs	arp/pp_obs
ini1wrfp.F90	arp/dia	arp/fullpos
ini2wrfp.F90	arp/dia	arp/fullpos
ini3wrfp.F90	arp/dia	arp/fullpos
mxmaop.h	xrd/include	xla/interface
yomlcz.F90	xla/module	arp/module
syminv.F	xrd/eclite	xla/external/linalg

Remarks:

- expbesu.F90 (and its TL and AD code) has been moved by error in arp/op_obs in CY35: it must return into arp/pp_obs.
- moving of mxmaop.h into XLA has been forgotten in CY35.
- yomlcz.F90 is no longer used in XLA so it can return into ARP/IFS.

H2: Routines to be renamed without moving.

Current name	New proposal of name
arp/adiab/laidditlad.F90	arp/adiab/laiddiad.F90
arp/adiab/laidlitlad.F90	arp/adiab/laidliad.F90
arp/adiab/laitritlad.F90	arp/adiab/laitriad.F90
arp/adiab/laitlitlad.F90	arp/adiab/laitliad.F90
ald/adiab/gpspng.F90	ald/adiab/gp_spng.F90 (g.p. calc., not GPS)
arp/obs_preproc/flspeedbad.F90	arp/obs_preproc/flspeedwrong.F90 (not AD)
arp/pp_obs/ppinitza.F90	arp/pp_obs/ppinitzad.F90 (adjoint code)
arp/pp_obs/ppobsaza.F90	arp/pp_obs/ppobsazad.F90 (adjoint code)
arp/var/surad.F90	arp/var/susats.F90 (not AD)
xrd/module/eggangles.F90	xrd/module/eggangles_mod.F90
xrd/module/eggmrt.F90	xrd/module/eggmrt_mod.F90
xrd/module/eggpack.F90	xrd/module/eggpack_mix.F90
xrd/module/local_trafos.F90	xrd/module/local_trafos_mod.F90
xrd/module/mpl_groups.F90	xrd/module/mpl_groups_mod.F90
sct/qretrieve/minima.F	sct/qretrieve/minima_sct.F

Remarks:

- About LAI..TLAD routines, remove the obsolete content of routines LAIDDIAD and LAIDLIAD before.

H3: Routines to be renamed and moved.

Current directory/name	New proposal of directory/name
xrd/not_used/ismin.F	xrd/utilities/ismin_ifsaux.F
xrd/not_used/ismax.F	xrd/utilities/ismax_ifsaux.F

Remarks:

- ismin.F and ismax.F are still used (for example in odb/extras/gribex/mxmncr.F), we suggest to rename them with an appendix "ifsaux" in order to avoid potential confusions with other versions spread in other projects, and also with local variables named ISMIN or ISMAX.

H4: Reorganisation of project UTI (move decks in new directories).

Current directory/name	New proposal of directory/name
uti/add_cloud_fields/add_clouds_fields.F90	uti/add_cloud_fields/programs/add_clouds_fields.F90
uti/addozoaer/addozoaer.F90	uti/addozoaer/programs/addozoaer.F90
uti/addsurf/proajout.F	uti/addsurf/programs/proajout.F
uti/addsurf/proajoutec.F	uti/addsurf/src/proajoutec.F
uti/addsurf/prolecfa.F	uti/addsurf/src/prolecfa.F
uti/combi/combi.F90	uti/combi/programs/combi.F90
uti/combi/combi_opti.F90	uti/combi/src/combi_opti.F90
uti/combi/combi_pert.F90	uti/combi/src/combi_pert.F90
uti/combi/combi_stat.F90	uti/combi/src/combi_stat.F90
uti/combi/masque.F90	uti/combi/src/masque.F90
uti/combi/proba.F90	uti/combi/src/proba.F90
uti/dategrib/dategrib.F90	uti/dategrib/programs/dategrib.F90
uti/gobptout/gobptout.F	uti/gobptout/programs/gobptout.F
uti/gobptout/prochien.F	uti/gobptout/src/prochien.F
uti/gobptout/procor1.F	uti/gobptout/src/procor1.F
uti/gobptout/proecrn.F	uti/gobptout/src/proecrn.F
uti/gobptout/proensuite.F	uti/gobptout/src/proensuite.F
uti/gobptout/proentete.F	uti/gobptout/src/proentete.F
uti/gobptout/progeom.F	uti/gobptout/src/progeom.F
uti/gobptout/proindex.F	uti/gobptout/src/proindex.F
uti/gobptout/prolecn.F	uti/gobptout/src/prolecn.F
uti/gobptout/proordspe.F	uti/gobptout/src/proordspe.F
uti/pregpssol/pregpssol.F90	uti/pregpssol/programs/pregpssol.F90
uti/pregpssol/filter_spssol.F90	uti/pregpssol/src/filter_spssol.F90
uti/pregpssol/get_model_gpssol.F90	uti/pregpssol/src/get_model_gpssol.F90
uti/pregpssol/get_tslot_gpssol.F90	uti/pregpssol/src/get_tslot_gpssol.F90
uti/pregpssol/read_list_gpssol.F90	uti/pregpssol/src/read_list_gpssol.F90
uti/pregpssol/read_obsoul_gpssol.F90	uti/pregpssol/src/read_obsoul_gpssol.F90
uti/pregpssol/write_obsoul_gpssol.F90	uti/pregpssol/src/write_obsoul_gpssol.F90
uti/progrid/progrid.F	uti/progrid/programs/progrid.F
uti/progrid/prochec.F	uti/progrid/src/prochec.F
uti/progrid/procor2.F	uti/progrid/src/procor2.F
uti/progrid/prodom.F	uti/progrid/src/prodom.F
uti/progrid/proecr.F	uti/progrid/src/proecr.F
uti/progrid/profac.F	uti/progrid/src/profac.F
uti/progrid/prolec.F	uti/progrid/src/prolec.F
uti/progrid_cadre/procadre.F90	uti/progrid_cadre/programs/procadre.F90
uti/progrid_cadre/prolec2.F	uti/progrid_cadre/src/prolec2.F
uti/progrid_cadre/proselect.F90	uti/progrid_cadre/src/proselect.F90
uti/progrid_cadre/unshrink.F90	uti/progrid_cadre/src/unshrink.F90
uti/sst_nesdis/lect_bdap.F90	uti/sst_nesdis/programs/lect_bdap.F90