GFL DATAFLOW AND SETUP: SOME PROPOSALS.

YESSAD Karim.

June 10, 2013

Version 4 (basis: CY40).

1 Introduction and purpose.

The current GFL dataflow and setup is not easy to understand, and difficulties are encountered by people wanting to add attributes or new GFL variables. In this technical paper we give some ideas for simplifications. In particuliar the following points must be studied:

- gather all the GFL setup at one location.
- rationalise the number of namelist GFL attributes.
- rationalise the number of GFL attributes which are not in namelists.

Basis of study is CY40.

2 Differences between GFL and GMV.

First we must recall when a variable can be considered as a GFL one, and what are the differences between a GFL and a GMV variable. In a purely historic point of view, GFL variables have replaced old passive scalars, but GFL notion has been extended to some quantities which must be kept in the code from one timestep to the following one, in particuliar in diabatic models.

There are some necessary and sufficient conditions to know if a variable can be considered as a GFL or a GMV one.

- GFL:
  - A necessary condition is $[dGFL/dt]_{adiab} = 0$, but this condition is not a sufficient one.
  - GFL variables are not deeply linked to the system of equations used; in other words we can say that the adiabatic model can run without any GFL variable. For example we can run a completely dry hydrostatic primitive equation model or a completely dry NH model, without any GFL variable.
  - For this reason specific humidity is a GFL variable.
- GMV:
  - $[dGMV/dt]_{adiab}$ may be different from 0, but this is a sufficient condition, not a necessary one.
  - GMV variables are deeply linked to the system of equations chosen. For example in a hydrostatic primitive equation model we cannot remove any of them.
  - For example if a system of equations uses θ (potential temperature) as thermodynamic variable, θ cannot be removed from the system of equations (even if $[d\theta/dt]_{adiab} = 0$). For this reason θ can be a GMV variable but not a GFL one.

3 GFL setup.

In CY40, GFL setup is done under SUGFL, called by SU0YOMA. We can notice that LSPLTHOIGFL is still in YOMDYN and its calculation is done at the end of SUGFL3.

Further actions will be done according to the way we will reorganise the GFL attributes. Encapsulation of set-up routines in a mixed module containing all what is currently in YOM_YGFL can be studied.

4 GFL attributes.

I try to sum-up some informal discussions I had with Sylvie Malardel during the first term of 2011.

4.1 Current status.

There are currently several modules containing type definitions, declarations, and set-up for GFL: type_gfls.F90, type_gflflds.F90, yom_ygfl.F90 and gfl_subs_mod.F90. The following study will focus on what is currently in type_gfls.F90 and yom_ygfl.F90.

The main drawbacks I currently see are:

- there are too many attributes, with some redundancies.
- namelist attributes appear in structure TYPE_GFL_NAML, but also in TYPE_GFL_COMP. In some pieces of code it is not easy to know if we must use Y[X] or Y[X]_NL; sometimes both have not the same value!
- some namelist attributes should not be in a namelist.
- some namelist attributes have wrong defaults and it is not always straightforward to put the right default.
- insufficient separation between true GFL variables and pseudo-GFL variables.

- routine SUDEFO_GFLATTR is currently very long (around 2400 lines) and is difficult to maintain, because we have one default for each package of physics, and around 25 attributes and 40 GFL variables.
- there are missing attributes saying for example if a GFL variable is diffused (horizontal diffusion), nudged, or coupled by spectral nudging.

4.2 Rationalise the number of namelist attributes.

4.2.1 Active spectral or grid-point GFL?

We have currently 4 attributes to say if a GFL variable is active, has a spectral representation (with or without derivatives) or a grid-point one (without derivatives in this case): LGP, LSP, LCDERS and LACTIVE. But only 4 combinations are actually meaningful and we could replace these attributes by NACTIVE, with the following meaning:

- NACTIVE=-1: grid-point active GFL (without derivatives).
- NACTIVE=0: not active GFL.
- NACTIVE=1: spectral active GFL (without derivatives).
- NACTIVE=2: spectral active GFL (with derivatives).

4.2.2 GFL identifier.

GFL variables have two identifiers: a name (attribute CNAME) and a GRIB code (attribute IGRBCODE). We can keep these attributes, and they can remain namelist ones.

4.2.3 Is GFL a true GFL one or a pseudo-GFL one?

There is currently no attribute saying if a GFL variable is a true one or a pseudo-GFL one. Only attribute LT1 allows to distinguish between the two categories, but this is not a clean way to do that. A specific namelist attribute can be useful for the following reasons:

- a limited amount of GFL variables may to have to be considered as true GFL or pseudo-GFL according to the model or the physics used (that can be the case of the cloud fraction).
- a true GFL can be advected or not, but a pseudo-GFL cannot be advected. Checkings about attribute LADV (see below) must consider the fact that we have a true GFL or a pseudo-GFL.

We can recall the difference between a true GFL and a pseudo-GFL:

- A true GFL is a prognostic variable where the different timesteps are well identified in the code. Additionally:
  - this GFL variable may be present in initial, coupling or historical files.
  - it can be advected or not.
  - it can be diffused or not.
  - a non-zero diabatic contribution may exist.
- A pseudo-GFL rather looks a diagnostic variable, which can be used for example:
  - to bring information from one timestep to the following one (this is the case for some output quantities of the physics which are used as input quantities during the following timestep).
  - to do some diagnostics.

  Additionally:

  - these quantities are generally not present in input or in coupling files.
  - they are not advected.
  - only one instant is stored.
  - they are generally grid-point ones.

4.2.4 Is GFL variable read in a file?

There are currently three namelist attributes: NREQIN, LGPINGP and REFVALI (REFVALI is used only if NREQIN=-1). LGPINGP could be removed, provided we add one value for NREQIN. We could keep NREQIN and REFVALI with the following values:

- NREQIN=-1: field not read in a file, but filled everywhere with REFVALI.
- NREQIN=0: field not read in a file, but filled everywhere with 0.
- NREQIN=1: field read in a file, with the two following possibilities:
  - if the GFL variable is a spectral one, file contains a spectral field.
  - if the GFL variable is a grid-point one, file contains a grid-point field.
- NREQIN=2: field read in a file, with the two following possibilities:
  - if the GFL variable is a spectral one, file contains a grid-point field.
  - if the GFL variable is a grid-point one, file contains a spectral field.

Implementing NREQIN=2 would be very convenient for example to be able to run forecasts with grid-point q from files containing spectral q.

NREQIN can be different from 0 only for active GFL.

4.2.5 Is GFL variable written in a file?

There is currently one namelist attribute: LREQOUT. This attribute can remain unchanged.

LREQOUT can be T only for active GFL.

4.2.6 Is GFL coupled (for LAM models)?

There are currently two namelist attributes: NCOUPLING and REFVALC (REFVALC is used only if NCOUPLING=-1). These attributes can remain unchanged, but we can add one option for NCOUPLING.

- NCOUPLING=-1: LBC field not read in a file, but filled everywhere with REFVALC.
- NCOUPLING=0: LBC field not read in a file, but filled everywhere with 0.
- NCOUPLING=1: LBC field read in a file, with the two following possibilities:
  - if the GFL variable is a spectral one, coupling file contains a spectral field.
  - if the GFL variable is a grid-point one, coupling file contains a grid-point field.
- NCOUPLING=2: LBC field read in a file, with the two following possibilities:
  - if the GFL variable is a spectral one, coupling file contains a grid-point field.
  - if the GFL variable is a grid-point one, coupling file contains a spectral field.

Implementing NCOUPLING=2 would be very convenient for example to be able to run forecasts with grid-point q, using coupling files containing spectral q.

NCOUPLING can be different from 0 only for active GFL.

4.2.7 Is GFL advected?

There are currently two namelist attributes: LADV and LADV5 (this one for trajectory). These attributes can remain unchanged.

LADV can be T only for active GFL. For Eulerian advection, LADV cannot be T for grid-point GFL.

4.2.8 Split physics for GFL?

There is currently one namelist attribute: LPHY. This attribute can remain unchanged.

LPHY can be T only if all the following conditions are matched:

- for active advected GFL.
- semi-Lagrangian scheme.
- if LAGPHY=T and LSLPHY=T.

4.2.9 Type of interpolation in the semi-Lagrangian scheme for GFL?

There are several namelist attributes:

- LTDIABLIN: linear interpolation for diabatic terms.
- LINTLIN: linear interpolation for all terms.
- LHORTURB: 3D turbulence used in the high-order interpolation.
- LSLHD: SLHD high-order interpolation.
- LHV: high-order interpolation with conventional Hermite cubic vertical interpolations.
- LVSPLIP: high-order interpolation with spline Hermite cubic vertical interpolations.

and that starts to look tricky. Some of them (LHV and LVSPLIP) are currently restricted to ozone. LHV and LVSPLIP are redundant and LHV would probably disappear in the future.

Something better must be found because all these keys cannot be T at the same time. For example we can replace them by two character variables:

- CSLINTGFL: type of interpolation applied to "GFL" evaluated at the origin point.
- CSLINTDIAB: type of interpolation applied to diabatic terms evaluated at the origin point.

replacing attribute CSLINT which is currently in TYPE_GFL_COMP but not in TYPE_GFL_NAML.

Additionally we will probably need an additional (logical) attribute saying if we must do two separate interpolations or one interpolation for "GFL" and diabatic terms (its meaning will be close to LSPLTHOIGFL), but this attribute can be retrieved from CSLINTGFL and CSLINTDIAB (and maybe other variables) and must not be a namelist one.

This will be a good occasion to find a good way to deal with the current variable LSPLTHOIGFL which means: there is at least one active advected GFL variable which requires a separate interpolation for GFL and diabatic terms.

Defining the type of interpolation must be done when the following conditions are matched:

- for active advected GFL.
- semi-Lagrangian scheme.

4.2.10 Monotonic interpolation in the semi-Lagrangian scheme for GFL?

There are two namelist attributes:

- LQM: monotonic interpolations.
- LQMH: monotonic interpolations on the horizontal only.

They could be replaced by one integer namelist attribute NQM, with the following values:

- NQM=0: not monotonic interpolation (current LQM=F, LQMH=F).
- NQM=1: horizontally monotonic interpolation (current LQM=F, LQMH=T).
- NQM=2: monotonic interpolation (current LQM=T, LQMH=F).

Additionally, having two versions NQMGFL and NQMDIAB of this variable (for "GFL" and diabatic terms) can be relevant, because if we want to do monotonic 32 points interpolations for "GFL" and trilinear interpolations for diabatic terms, we will have to use NQMGFL=2 and NQMDIAB=0.

Non-zero values for NQMGFL and NQMDIAB will be possible only for:

- active advected GFL.
- semi-Lagrangian scheme.

4.2.11 Is GFL spectrally nudged (for LAM models)?

There is currently no attribute saying if a GFL variable is spectrally nudged: only q is assumed to be spectrally nudged (if LESPCPL=T, SPNUDQ>0 and if q is active with a spectral representation).

A way must be found to deal with that (for more flexibility about spectral nudging of GFL variables) if that becomes necessary. For example the following attributes could be added in the future: NESPCPL, REFVALSPC and SPNUD.

- NESPCPL=-1: LBC file field not used for spectral nudging; spectral nudging is done using a pseudo-LBC equal to REFVALSPC everywhere.
- NESPCPL=0: no spectral nudging for this GFL variable.
- NESPCPL=1: LBC file field used for spectral nudging.

4.2.12 Is GFL nudged (for climatic applications)?

There is currently no attribute saying if a GFL variable is nudged: in CPNUDG (nudging done in grid-point space), only q is assumed to be nudged (if LNUDQG); in SPCHOR (nudging done in spectral space), only q is assumed to be nudged (if LNUDSH).

A way must be found to deal with that if that becomes necessary. For example the following attributes could be added in the future:

- LNUDG (activates nudging in grid-point space).
- XNUDG (nudging coefficient in grid-point space).
- LNUDS (activates nudging in spectral space).
- XNUDS (nudging coefficient in spectral space).

For example, LNUDSH will be replaced by YQ_NL%LNUDS.

This nudging done in spectral space (for global models) must not be confused with "spectral nudging" for LAM models (these are two completely different pieces of code which use different variables).

4.2.13 Is GFL horizontally diffused?

There is currently no attribute saying if a GFL variable is horizontally diffused: only q and ozone are assumed to be horizontally diffused (if these GFL variables are active ones with a spectral representation).

A way must be found to deal with that if that becomes necessary. For example the following namelist attributes could be added in the future:

- LHDF (activates horizontal diffusion in spectral space).
- RDAMP (equivalent of current RDAMQ, RDAMPO3).

For example, RDAMPQ will be replaced by YQ_NL%RDAMP.

Other attributes can be added (HDIR, HRDIR) but they must not be in NAMGFL.

Only active spectral true GFL can be diffused.

Use of such attributes can work only if LNEWHD=T, and in practical implementation can be done only after pruning option LNEWHD=F.

4.2.14 Configurations using TL or AD codes.

Additional requirements may be necessary, for example when the TL or AD codes use only a subset of the GFL variables used in the direct code. The current way to deal with that is not satisfactory, messy, and is regularly source of bugs; changing the list of GFL variables taken into account in the TL and AD codes requires deep (and difficult) developments. Features are currently often hard coded, explicitly using individual GFL variables, not easy to understand, and potentially bugged. No extensive details will be given, but the following cases are encountered:

- The TL or the AD model uses less GFL variables than the direct one.
- In the TL or AD code, calculation of increments of moist air constant may use a subset of GFL increments, or no GFL increment (current use of key LDRY_ECMWF).
- In the TL or AD code, there are pieces of code (currently under LPROCLDTL) which conditionally take account of liquid water and ice in some diabatic calculations.

It is maybe time to think about a set of namelist attributes to deal with that. An idea would be to have three TYPE_GFL_NAML-typed variables for each GFL variable: one for the direct code, one for the trajectory, and one for the TL and AD codes. Example for specific humidity: YQ_NL, YQ5_NL and YQ_TLAD_NL (note that YQ_NL%LADV5 will be replaced by YQ5_NL%LADV in this case); in the TL and AD codes we will find references to YQ_TLAD_NL (increments) and YQ5_NL (trajectory code) but no reference to YQ_NL.

A properly designed way to deal with that will probably allow to remove routines DEACT_CLOUD_GFL and REACT_CLOUD_GFL which are a potential source of bugs.

4.2.15 Other namelist GFL attributes.

There are other GFL attributes which are used in specific ECMWF applications (they are used only for a subset of GFL variables: for example TRAC and GRG) and for which I will not give any detail. Namelist attributes are for example NCOUPLO4, LASSIM, IGRIBTC, IGRIBSFC, LDIFF, LCONV, LNEGFIX, RMOLMASS. Indeed, having attributes which have a sense only for a subset of GFL only is questionable: is it the right method to deal with that?

4.3 Rationalise the number of attributes which are not namelist ones.

4.3.1 Timestep levels.

There are three attributes: LT9, LT1 and LT5. There are currently in NAMGFL but they must not be in a namelist. These attributes must be computed according to the configuration and the advection scheme used. For example:

- LT1: must be T for an active true GFL, F otherwise.
- LT9: must be T for an active true GFL if a leap-frog advection scheme is used. Its value may depend on some other variables like LPC_FULL.
- LT5: will be T only for configurations calling TL and AD code (requiring to store trajectory GFL).

Future management of attribute LT5 may interfere with what we said in paragraph "Configurations using TL or AD codes".

4.3.2 GFL in ICI schemes (predictor-corrector).

There are two attributes: LPT and LPC. There are currently in NAMGFL but they must not be in a namelist. Such attributes must be computed according to the configuration and the advection scheme used, and the value of LPC_FULL. For example:

- LPT can be T only if LPC_FULL=T, for true GFL with non-zero diabatic tendencies.
- LPC can be T only if LPC_FULL=T, for true GFL.

For more details:

- for LPT, all the following conditions must be matched to set it to T:
  - true GFL, with a representation at t and $t + dt$.
  - diabatic run (LEPHYS.OR.LMPHYS.OR.LSIMPH).
  - calculation of a diabatic tendency for this GFL.
  - LPC_FULL=T.
  - semi-Lagrangian scheme.
- for LPC, all the following conditions must be matched to set it to T:
  - true GFL, with a representation at t and $t + dt$.
  - LPC_FULL=T.
  - (.NOT.LPC_CHEAP and .NOT.LTWOTL) or (LPC_CHEAP and LTWOTL).

4.3.3 GFL trajectory for TL and AD codes read in a file?

There is one attribute: LTRAJIO. It is currently in NAMGFL but it must not be in a namelist. This attribute must be computed according to the configuration and the value of keys like LTRAJHR and LTRAJHR_ALTI. All the following conditions must be matched to set it to T:

- configuration using a TL or an AD code.
- active true GFL, with a representation at t and $t + dt$.
- this GFL is required in the trajectory.
- LTRAJHR (and maybe LTRAJHR_ALTI too) is T.

Future management of attribute LTRAJIO may interfere with what we said in paragraph "Configurations using TL or AD codes".

4.3.4 MP pointers GFL attributes.

No significant change is required for these attributes, at least for the direct model.

About TL and AD codes, the ideal way to deal with that may interfere with what we said in paragraph "Configurations using TL or AD codes":

- Trajectory: we have currently attributes like MP5, but I wonder if it would not be better to have a variable Y[X]5 and to use attribute MP: for example replace YQ%MP5 by YQ5%MP. About dimensions YGFL%NDIM5 could become YGFL5%NDIM.
- Increments: we use the same structures as for the direct code, but I wonder if it would not be better to have a variable Y[X]_TLAD and to use attribute MP: for example replace YQ%MP by YQ_TLAD%MP. About dimensions YGFL%NDIM could become YGFL_TLAD%NDIM.

4.3.5 Other GFL attributes.

Other attributes are LTHERMACT, R, RCP, LADJUST0, LADJUST1, LBIPER, CSLINT.

- LTHERMACT, R and RCP go together and allow to compute moist air constant and moist c_p.
- LADJUST0 and LADJUST1 are not used, and can probably be removed.
- LBIPER is currently not used (but its use is maybe planned in future developments?). LBIPER has a sense for LAM models only.
- CSLINT: see above paragraph “Type of interpolation in the semi-Lagrangian scheme for GFL”.

Apart from useless ones, no significant change is required for these attributes.

4.4 Timing.

Such a work still needs thinking and coordination. It can be split in several bits so work can be spread among several development cycles. High priority must be given on attributes allowing to simplify routines SUDEFO_GFLATTR and SUCTRL_GFLATTR (this is the case for example of attributes NACTIVE, CSLINTGFL and CSLINTDIAB). Items about horizontal diffusion, nudging and spectral nudging are the less stringent to deal with.

4.5 Additional consequences.

Rewriting dataflow (attributes) will have large consequences of the set-up under the new SUGFL.

- Code under SUGFL1 will be deeply modified. Once done SUGFL1, all the namelist attributes must be computed, and they must not be modified later.
- Routines DEFINE_GFL_COMP and SET_GFL_ATTR must be reorganised; The modified version of SET_GFL_ATTR must compute all the TYPE_GFL_COMP GFL attributes which are not in NAMGFL.
- It is possible that we have to merge SUGFL2 and SUGFL3 into one routine.

5 Encapsulation of all GFL applications (type definition, declarations, set-up) in one module: gfl_mod.F90.

This new module will contain:

- type definition: content of type_gfls.F90 and type_gflflds.F90.
- variable declaration: content of yomgfl.F90 and yom_ygfl.F90.
- namelist: NAMGFL.
- set-up, allocations, deallocations:
  - routines encapsulated in gfl_subs_mod.F90: currently DEFINE_GFL_COMP, SET_GFL_ATTR, PRINT_GFL, DEACT_CLOUD_GFL, REACT_CLOUD_GFL, FALSIFY_GFLC, NOADVECT_GFLC, COPY_GFLC_GFLC and maybe some other ones recently introduced.
  - the new SUGFL, SUGFL1, SUGFL2, SUGFL3.
  - SUDEFO_GFLATTR and SUCTRL_GFLATTR.
  - maybe SUCSLINT too.

6 Conclusion.

We have identified several tasks. They can be split among several development cycles. In particuliar, item about TL and AD codes will be a difficult one where the right solution is not easy to find (and probably requires deep recoding).