CODING OF SETUP MODIFICATIONS

GMAP stay report (9/10/05 - 18/11/05)

Jure Cedilnik
(Environmental Agency of the Republic of Slovenia)

GMAP supervisor: Yann Seity

Toulouse, 18/11/2005

Introduction

Before reading this report, one should read the document titled «Proposal on additional GFL attributes and consistency of their use», by the same author which is in the appendix of this report.

The coding was done in four hierarchical steps, each with their own view and pack. These are summed in the following table:

	VIEW name	PACK name	BINARY name	TEST directory
0			AROME_30t1_ref	reference
1	gfl_mod.v03	gfl_setup_test	AROME_no_ladvamv	no_ladvamv
2	su_no_lcldpin	su_no_lcldpin	AROME_no_lcldpin	no_lcldpin
3	ncoupling	ncoupling	AROME_ncoupling	ncoupling
4	cslint	cslint	AROME_cslint	cslint

The view name is the name of the view in Clearcase on **andante**.
The top directory for all the packs is: **tora:~mrpa669/mypack**.
The directory for all the binaries is: **cougar:~mrpa669/bin/arome_setup_mods_tests**.
The top directory for all test directories is **tora:~mrpa669/arome_setup_mods_tests**.

Each view sees the one below and one has to be careful with the exports: to obtain a full set of modifications for **cslint**, the cc_export command should be executed in each of the previous views.

What follows is an extensive description of modifications in all of the four steps (views).

1. gfl_mod.v03 : removal of LADVAMV

Modified routines and their modifications:

yomdyn

- removal of LADVAMV;

namdyn.h

- removal of LADVAMV;

sudyn

- move of definitions of default values for LADVAMV when LECMWF or when LAROME to routine **sudim1**, just before call to NAMGFL;
- modifications of consistency checks (Y[X]_NL%LADV instead of LADVAMV);
- prints out of Y[X]_NL%LADV instead of LADVAMV;
- use of Y[X]_NL%LADV when settiing attributes to GFLs;
- removal of LAROME switch when calling SET_GFL_ATTR for TKE, Q and EXT;
- using LARPHY switch to call SET_GFL_ATTR for TKE with correct interpolator (this is only temporary - to be able to test the pack: TKE can only work with CSLINT='LAITQM');

sudim1

- in the code coming from **sudyn**: replacement of LADVAMV with Y[X]_NL% LADV;
- setting defaults for advection for GFLs (before calling NAMGFL) for cases of LECMWF or LARPHY;
- replacement of switch LAROME with LARPHY;

mf_phys

- additional call to CPUTQY_AROME when there is advection of hydrometeors and no advection of TKE (this modification is for testing purposes); this (YTKE_NL%LADV=.F.) only works without the PC sheme or with YTKE_NL% LPT=.F.)

2. su_no_lcldpin : removal of LCLDPIN and replacement of LREQIN with NREQIN

Modified routines and their modifications:

namdim.h
- removal of LCLDPIN;

yomdim
- removal of LCLDPIN;

gflsubs
- NREQIN instead of LREQIN;

sugfl
- NREQIN instead of LCLDPIN in call to DEFINE_GFL_COMP;

sudim2
- LCLDPIN is replaced by local LLCLDPIN, which is set to ((YA_NL%NREQIN==1).AND.(YL_NL%NREQIN==1).AND.(YI_NL%NREQIN==1))*;
- additional USE YOM_YGFL for YA_NL, YL_NL, YI_NL;

sugridug

same as **sudim2**;

rdfpinc
- JNREQIN instead of LLREQIN;
- NREQIN instead of LREQIN;

sugridua
- NREQIN instead of LREQIN;

type_gfls
- NREQIN instead of LREQIN;

wrgridua
- NREQIN instead of LREQIN;

sudim1
- setup of default value for LCLDPIN depending on which model is used is replaced by setup of YI_NL%NREQIN, YA_NL%NREQIN and YL_NL%NREQIN, this is connected to **sudim2** and the corresponding ECMWF issues;

---

* I believe this is only used by ECMWF in such way. The present solution is not very nice and a different one should be sought with the help of ECMWF.

3. ncoupling : replacement of LCOUPLING by NCOUPLING, introduction of LREQOUT, REFVALI and REFVALC

gfl_subs

- replacement of LCOUPLING with NCOUPLING;
- additional LREQIN, REFVALI, REFVALC;

type_gfls

same as gfl_subs;

sudyn

- additional REFVALC in calls to SET_GFL_ATTR;
- use of NCOUPLING instead of LCOUPLING;

suedyn

- modification of one test: (NCOUPLING==1) instead of (LCOUPLING);

sugfl

- additional arguments Y[X]_NL%REFVALI and Y[X]_NL%LREQOUT in call to DEFINE_GFL_COMP;

sudim1

- replacement of definitons of default LCOUPLINGs with NCOUPLINGs;
- additional defaults for TKE in case of LARPHY (NREQIN=-1, REFVALI=0.000001);
- additional defaults for LREQOUT (.T. if LARPHY);

scan2mtl

- instead of test (.NOT.LCOUPLING) use (NCOUPLING==1);

scan2mdm

same as for **scan2mtl**;

scan2mad

same as for **scan2mtl**;

elswa3

- instead of test (.NOT.LCOUPLING) use (NCOUPLING==1);
- additonal option if (NCOUPLING==-1)

ecoupl1

- replacement of 1st occurrence of if(LCOUPLING) with if(NCOUPLING==1) - this fills buffer for coupling, additonal option else(NCOUPLING==-1), to fill the buffer with REFVALC;
- replacement of 2nd occurrence of if(LCOUPLING) with if (.NOT. (NCOUPLING==0));
- replacement of 3rd occurrence of if(LCOUPLING) (for time-stepping) with if (NCOUPLING==1)

ecoupl1ad

- same as **ecoupl1**, except that there are only two occurrences of LCOUPLING (first two);

4. cslint : additional GFL attributes: LQM, LQMH, LSLHD, LRSPLINE...

namdyn.h
- removal of LQMQ, LQMV, LQMHQ, LQMHV, LRSPINE_Q, LRSPLINE_O3, LRSPLINE_V, LHV03, *note:* LVSPLIP is not removed;

yomdyn
- same as **namdyn.h**;

namdyna.h
- removal of LSLHD_TKE, LSLHD_Q, LSLHD_O3, LSLHD_CIW, LSLHD_PREC, LSLHD_V;

yomdyna
- same as **namdyna.h**

namct0.h
- removal of LVSPLIP;

suct0
- LVSPLIP is set to .FALSE. by default, it is re-set later in **sudim1** if any of the GFLs need it;

suhdf
- removal of USE YOMDYNA : LSLHD_O3, LSLHD_Q;
- additional USE YOM_YGFL: YQ_NL, YO3_NL;
- replacement of LSLHD_O3 with YO3_NL%LSLHD and LSLHD_Q with YQ_NL% LSLHD;

suehdf
- same as **suhdf**

type_gfls
- removal of CSLINT attribute from _NL GFL structure;
- additional logical attributes in _NL GFL structure:LRSPLINE, LQMH, LQM, LHV, LVSPLIP, LSLHD;

sudyna
- removal of LSLHD_Q, LSLHD_O3, LSLHD_V, LSLHD_TKE, LSLHD_CIW, LSLHD_PREC, their new equivalents are not used in this routine - setup of their default values will be in **sudim1**;
- LSLHD model switch now computed only from LSLHD_* swithces which remain (these are GMV variables only);

sudyn
- removal of default setup for interpolation desriptors for if(LECMWF) or similar - this goes to **sudim1**;
- modification of call to SET_GFL_ATTR - now with new Y[X]_NL%* - this is for all GFLs, including AERO, EXT, CPF, SPF and CVGQ
- additional USE YOM_YGFL: CVGQ_NL
- replacement of QMV, QMHV, LHVO3, LSLHD_Q with Y[X]_NL%*
- removal of if(LARPHY) call to SET_GFL_ATTR for TKE introduced in step 1;
- call to SET_GFL_ATTR is always with Y[X]%CSLINT, which is computed with routine SUCSLINT using values of Y[X]_NL%*;

sudim1
- before reading of namelist NAMGFL, set-up of default values for all possible interpolation descriptors, including AERO, EXT, SPF, CPF, CVGQ;
- after reading namelist NAMGFL, re-set the values of LVSPLIP and LSLHD (if any GFL requiers those types of interpolations),
- consistency checks for ozone and LVSPLIP;
- consistency checks for other variables - not allowed to use LVSPLIP or LHV

(yet);

5. Validation and remarks

All the four packs were validated with AROME binary on Gard case and default settings. Norms from 20 time steps were compared.
There are still some issues to validate: it is not completely sure if NREQIN==0 or -1 and LREQOUT=F work as supposed to, the NCOUPLING=-1 has also not been thouroghly tested. I plan to continue working on this validations.
The REFVALI attribute was tested with TKE. It changed the norms if a different initial value of TKE was defined in the namelist. I believe this is sufficient.
The YTKE_NL%LADV=.F. was tested (with hydrometeors advected) and it works but only with YTKE_LPT%LPT=.F. or without the PC scheme. Other combinations of (non) advection of hydrometeors cannot be tested, because at the moment the necessary code is not present in MF_PHYS.

Further tests with other models are needed (ARPEGE and ALADIN).

There is a need to discuss the two subjects with ECMWF:
1. In case they are not in favour of replacing the LREQIN with NREQIN with the next cycle, the LREQIN will have to be put back in the model, to be used by ECMWF's i/o routines.
2. The connection between LCLDPIN and YA_NL%LREQIN, YL_NL%LREQIN and YI_NL%LREQIN is not very clear to me. Is LCLDPIN equivalent to ((YA_NL% LREQIN=T).AND.(YL_NL%LREQIN=T).AND.(YI_NL%LREQIN=T)) or not? (*see also section 2 and the footnote there*). Again, if ECMWF would like to keep LCLDPIN in their i/o routines it can be put back in the code, but it is not needed anymore on the ARPEGE/ALADIN/AROME side.

APPENDIX: Proposal on additional GFL attributes and consistency of their use

Toulouse, 25/10/2005

by Jure Cedilnik

(after discussion with Filip Vana, Karim Yessad, Gwenaëlle Hello, Claude Fischer and Yann Seity)

Introduction

The GFL structure offers a very nice way to introduce new variables without any stress whatsoever. However, some features are missing and what is more, its rules are not always respected. For instance: use of LADVAMV switch, which determines (non) advection of GFLs (except for ozone and humidity, which are treated seperately) is obsolete. There is a GFL advection attribute called LADV, which is in many cases simply disregarded.

2. Input/output and coupling issues

Currently there are GFL attributes for LREQIN (input of a GFL field from the file) and LCOUPLING (whether to apply coupling on it or not) which determine the associated properties of a GFL. But the combination of the two do not cover all the possibilities. Fixed value coupling or fixed value initialisation is not taken into account. What is more, there is no control of the output of a GFL: in the present state, LREQIN is used for input and output of a GFL variable.

*EXAMPLE: TKE in AROME is currently initialised with value 10**-6, is not coupled and we are very much interested in its output. This means we need to inject it in the initial and coupling files with this value so that we will get an output.*

The idea is to replace LCOUPLING by NCOUPLING and LREQIN with NREQIN. This is a consequence of nine options possible for initialisation and coupling (see the table).

option	NREQIN	NCOUPLING
Initialisation from file, no coupling	1	0
Initialisation from file, LBC coupling	1	1
Initialisation from file, coupling with reference value	1	-1
Initialisation with reference value, no coupling	-1	0
Initialisation with reference value, LBC coupling	-1	1

Initialisation with reference value, coupling with reference value	-1	-1
No initialisation, no coupling	0	0
No initialisation, LBC coupling	0	1
No initialisation, coupling with reference value	0	-1

In general, NREQIN and NCOUPLING would be 1 if the field is to be read from file, -1 if it is not read but set to a reference value or 0 if it is neither read nor set to a certain reference value. For a global model, one does not need to care about NCOUPLING value, in this case it should be by default 0.

It is true, that the last two options in the table seem very unusual, but we should preserve them as possibilities. To make the matter even more complicated, there should be two reference values prescribed: one for coupling and one for initialisation. Again, these options would be present to make things as general and consistent as possible and without having any particular use of them in mind. So there should be two reference values as GFL attributes: REFVALI and REFVALC, for initialisation and coupling respectively.

Another proposed attribute for the GFLs is the LREQOUT (field requiered in output) (similar to now already former LREQIN), which would be by default TRUE for all variables, but would enable that a field may not be in the initial file, but can be in the output. As it is done now, LREQIN and LCLDPIN control GFL 's output as well.

With replacement of LREQIN with NREQIN we are dealing with a subject that is strongly connected with the IFS. Therefore any changes should be taken in close agreement with the ECMWF. In case ECMWF would not want to switch to NREQIN with the next cycle, is possible to keep the LREQIN attribute as it is and have a transition period during which both of them would exist.

With all these modifications there is no further need for LCLDPIN switch. In fact, LCLDPIN is a duplication of the GFL structure and should be removed by default. All input/output and coupling would be controled by NCOUPLING, NREQIN, LREQOUT, REFVALI and REFVALC.

There would be no consistency check necessary at this point since all options are in principle allowed.

2. Advection and choice of interpolators for GFLs

As mentioned in the introduction: not necessarily all GFL fields need to be advected, some do and others do not. This is already taken care of in the GFL structure itself (with LADV attribute), but this choice is more or less everywhere overwritten with value of LADVAMV from NAMDYN. This means that the setup routines would need some cleaning: there is no need for LADVAMV anymore and the routines DEFINE_GFL_COMP and SET_GFL_ATTR should be called with the proper GFL attributes' values. In the present state, these routines are called with some particular variables' switches, some general switches (like LADVAMV) and what is the worst, with some hardcoded values.

Another issue is the choice of interpolators. There are now many logical switches used in

the setup governing the choice of interpolators for one or more GFLs (for instance LSLHD_O3 for ozone, LQMHV for other GFLs...). These need to be replaced by GFL attributes. The GFL attribute CSLINT (describing interpolation type) should remain, but should not be anymore an attribute of the namelist counterpart of GFL. CSLINT should always be computed in routine SUCSLINT (as it is already done for some GFLs) which would use the values of GFL namelist attributes. New *_NL attributes would be: LSLHD (for diffusive interpolation), LQM (for quasi-monotonous interpolation), LQMH (for horizontal quasi-monotonous interpolation), LRSPLINE (for 4 points spline interpolation), LHV (for vertical Hermite interpolation) and LVSPLIP (for full spline interpolation along vertical). Along with the introduction of the new attributes, some new consistency checks should be introduced as well:

- we only allow LVSPLIP=T attribute to be used with ozone, and in that case, any other newly defined attributes must be set to F (for other GFLs or with another combination of switches it is not yet coded)
- LHV=T can also only be used for ozone (again because the code does not exist) and in this case LSLHD must be F and LRSPLINE must be F

If these requirements are not met, the model should complain and abort.

These newly defined attributes are a compromise solution. The most consistent solution would be to use fully GMV-like set of interpolator switches. This would mean LSLHD_TKE for use of semilagrangian interpolators for TKE, like we have LSLHD_T for use of semilagrangian interpolators for temperature. But such a set of switches would grow very fast with the number of GFLs and there is no guarantee that for a newly introduced GFL all corresponding switches would be defined. What is more, there is a chance that with an introduction of a new GFL it might again come to a common switch for more than one GFL (like now LADVAMV). And this is the problem we are currently trying to solve with this proposal.

EXAMPLE: Instead of logical switch LSLHD_TKE one would use a GFL structure switch YTKE%LSLHD.

*EXAMPLE: We would like to have a TKE field, with SLHD and quasimonotonous interpolation, without advection, coupled and initialised with reference value 10**-6, without being present in the initial file. This would be defined in the namelist as*
YTKE_NL%NREQIN=-1
YTKE_NL%LREQOUT=.T.
YTKE_NL%NCOUPLING=-1
YTKE_NL%REFVALI=0.000001_JPRB
YTKE_NL%REFVALC=0.000001_JPRB
YTKE_NL%LADV=.F.
YTKE_NL%LSLHD=.T.
YTKE_NL%LQM=.T.
YTKE_NL%LQMH=.F.
YTKE_NL%LRSPLINE=.F.
YTKE_NL%LHV=.F.
YTKE_NL%LVSPLIP=.F.

Of course many of these options can be set to default values according to some other logical switches in the setup and there would be no need to define them all in the namelist if one just uses defaults.

NOTE: The particular physics packages' switches would still remain (for instance LARPHY), but they would only define the default values of Y[X]%* attributes, that will be used if they aren't mentioned in the namelist. It is very important that such switches should never overwrite something one asked for in the namelist. If something is inconsistent, the model should complain and abort.