CONFIGURATION 801 IN THE CYCLE 40 OF ARPEGE/IFS.

YESSAD K. (METEO-FRANCE/CNRM/GMAP/ALGO)

July 3, 2013

Abstract:

This documentation describes the algorithmic and technical aspects of the configuration 801 of ARPEGE/IFS, which allows in particular to do sensitivity studies. An example of namelist is provided.

Résumé:

Cette documentation décrit les algorithmes et les aspects techniques de la configuration 801 d'ARPEGE/IFS, qui permet en particulier de faire des études de sensibilité. On fournit un exemple de namelist.

Contents

1 Introduction and purpose.

This configuration is used to do sensitivity studies and can help to answer to the following question: defining a scalar cost function (preferably a quadratic one), what is the sensitivity of this cost function to some basic meteorological variables (for example the state variables of the model)? What are the basic variables which have the most impacts on this cost function?

At least the adjoint code is used in this configuration. Such a configuration has been briefly described in some papers or internal documentations, for example (NTA28), (Rabier et al., 1996) or (Soci et al, 2003). A more detailed documentation (ID801O) has been provided in the past but it has not been updated since 1995; most aspects of this documentation remain valid for cycle 40 and cycle AL40. One currently tries to give a more detailed updated description of this configuration.

∗ Modifications since cycle 39:

- Removal of options LAVARC, LMINI, L801TL.

2 Algorithm of this configuration.

The direct model is defined by the primitive equations model, but it would be possible to do sensitivity experiments on a 2D shallow-water model or on a 3D non-hydrostatic model (currently not yet coded).

2.1 Notations.

- U (resp. V): components of the horizontal wind.
- D is the horizontal wind divergence.
- χ is the velocity potential ($D = \nabla^2 \chi$).
- ζ is the horizontal wind vorticity.
- ψ is the stream function ($\zeta = \nabla^2 \psi$).
- T is the temperature.
- $T_{\rm ST}$ is the standard atmosphere temperature.
- q is the humidity.
- Π is the hydrostatic pressure.
- $\Pi_{\rm ST}$ is the standard atmosphere hydrostatic pressure.
- $\Pi_{\rm s}$ is the hydrostatic surface pressure.
- $c_{\rm p}$ is the specific heat at constant pressure for air.
- $L_{\rm v}$ is the latent heat vaporisation constant for water.
- A_Π is a constant to make homogeneous the formula defining f.
- K_ψ, K_χ, $K_{\rm U}$, $K_{\rm V}$, $K_{\rm T}$, $K_{\rm q}$, K_Π are coefficients (generally 0 or 1) allowing to select only a subset of variables in function f.

2.2 Case without any minimization.

The simplest case to study is the case where one wants to answer the following question: once defined a scalar cost function (preferably a quadratic one), what is the sensitivity of this cost function to some basic meteorological variables (for example the state variables of the model)? No minimization is involved in this case; the adjoint code is used; the tangent linear code may be used in the limited area model in some cases.
It follows the following steps, using the direct and the adjoint codes. The forecast is assumed to have $N_{\rm stop}$ timesteps. One provides an initial situation (vector $\mathbf{X}_0$) and a reference analysis or forecast valid for the instant $t = t_{N_{\rm stop}}$ (vector $\mathbf{Y}_{N_{\rm stop}}$).

- The initial step is denoted by the vector of state variables $\mathbf{X}_0$. The forecast direct integration provides the steps $\mathbf{X}_1$, $\mathbf{X}_2$, ..., $\mathbf{X}_{N_{\rm stop}}$. These quantities are used as a trajectory for the adjoint integrations. The components of vector $\mathbf{X}$ are x_i. The model integration is symbolically denoted by

$$\mathbf{X}_{N_{\rm stop}} = \mathcal{M}_{0->N_{\rm stop}} \mathbf{X}_0$$

- One has also another final state $\mathbf{Y}_{N_{\rm stop}}$ which is a priori different from $\mathbf{X}_{N_{\rm stop}}$. That can be another forecast, or an analysis. The difference between $\mathbf{X}_{N_{\rm stop}}$ and $\mathbf{Y}_{N_{\rm stop}}$ is assumed to be not too big and is denoted by $[\delta\mathbf{Y}]_{N_{\rm stop}}$.
- Let us define a scalar function of $\mathbf{X}$: $f(\mathbf{X})$. See paragraph 2.3 for examples of available functions in ARPEGE/IFS and LAM versions. This is the "cost function". One computes $f([\delta\mathbf{Y}]_{N_{\rm stop}})$.
- Applying the adjoint model to the perturbation $(\delta\mathbf{Y})_{N_{\rm stop}}$ provides the gradient of the cost function with respect to the initial state.

$$\nabla f = \mathcal{M}^*_{0->N_{\rm stop}} [\delta\mathbf{Y}]_{N_{\rm stop}}$$

  where $\mathcal{M}^*_{0->N_{\rm stop}}$ is the adjoint operator of $\mathcal{M}_{0->N_{\rm stop}}$.

Remark: use of configuration with minimizations does not exist any longer in the code.

2.3 Aspect of the function f.

This is a scalar function of the model state variables, a quadratic one being desirable. Three choices are currently available (variable **NJROPT**):

- **NJROPT**=1: the following quadratic one:

$$f(\psi,\chi,T,q,\log\Pi) = 0.5\int_{\mathcal{D}}\left(K_\psi\psi^2 + K_\chi\chi^2 + K_{\rm T}c_{\rm p}T^2 + K_{\rm q}L_{\rm v}q^2\right)dsd(\log\Pi) + 0.5\int_{\mathcal{D}_{\rm s}} K_\Pi A_\Pi(\log\Pi)^2 ds$$

- **NJROPT**=2: a linear one:

$$f(U,V,T,q,\Pi) = 0.5\int_{\mathcal{D}}\left(K_{\rm U}\frac{U}{U_{\rm r}} + K_{\rm V}\frac{V}{V_{\rm r}} + K_{\rm T}\frac{(T-T_{\rm ST})}{T_{\rm r}} + K_{\rm q}\frac{q}{q_{\rm r}}\right)ds\frac{d\Pi}{\Pi_{\rm sr}} + 0.5\int_{\mathcal{D}_{\rm s}} K_\Pi\frac{(\Pi-\Pi_{\rm ST})}{\Pi_{\rm sr}}ds$$

  In practical: $U_{\rm r} = V_{\rm r}$=1 m/s; $T_{\rm r}$=1 K; $q_{\rm r}$=1 g/m^3; $\Pi_{\rm sr}$=101325 Pa.

- **NJROPT**=3: an enstrophy cost-function.

$$f(\psi) = 0.5\int_{\mathcal{D}}\psi^2 dsd(\log\Pi)$$

  This is a sub-option of the quadratic function defined for **NJROPT**=1.

- Cases **NJROPT**=2 and **NJROPT**=3 are obsolescent and candidates for pruning (not regularly used nor validated).

This function can be zeroed out of a pre-defined domain. $\mathcal{D}$ is the domain where f is non-zero, $\mathcal{D}_{\rm s}$ is the surface projection of $\mathcal{D}$. This possibility is interesting for example to do sensitivity studies in a limited region.

3 The sequences of STEPO called.

3.1 General architecture under STEPO.

```
control/STEPO ->
* Management of read/write: utility/IOPACK
* Inverse spectral transforms: transform/TRANSINVH -> TRANSINV_MDL
* Grid point computations: control/SCAN2M
* Direct spectral transforms: transform/TRANSDIRH -> TRANSDIR_MDL
* Spectral computations: control/(E)SPCM (see below).
```

The sequences of call to **STEPO** are controlled by a variable (often called **CDCONF** or **CLCONF**) containing 9 letters or zeros [L1][L2][L3][L4][L5][L6][L7][L8][L9]

- L1 controls the file write/read.
- L2+L3 controls the inverse transforms.
- L4 controls the grid-point computations for dynamics and physics.
- L5 controls the grid-point computations for some diagnostics.
- L6 controls the grid-point computations for assimilation.
- L7+L8 controls the direct transforms.
- L9 controls the spectral computations.

For example a model integration time-step is defined by the sequence [L1]AAA00AAA.
The tangent linear and adjoint codes of **STEPO** are respectively **STEPOTL** and **STEPOAD**.

3.2 Example of sequences of STEPO, STEPOTL and STEPOAD called in a configuration 801 ARPEGE/IFS run with N_{stop} timesteps.

```
SIMUL0 (SIM4D called by CGR1):
 -----------------
 TAAA00AAA STEPO    (jstep=1)
 ..................
 TAAA00AAA STEPO    (jstep=Nstop)
 -----------------
 B00000000 STEPO
 -----------------
 0K0000000 STEPO
 0GB0L0AA0 STEPOTL
 0GB0L0AA0 STEPOAD
 -----------------
 A00000000 STEPO
 -----------------
 0E0000000 STEPO    (jstep=Nstop)
 0AAA00AAA STEPOAD
 ..................
 0E0000000 STEPO    (jstep=1)
 0AAA00AAA STEPOAD
 -----------------
 A00000000 STEPO
 000000000 CNT0
 -----------------
```

- First the model runs N_{stop} sequences of type 'TAAA00AAA STEPO': this is the direct model which computes the trajectory; the trajectory is written on a buffer ([L1]='T').
- The model then reads $\mathbf{Y}_{N_{\text{stop}}}$ on a file called 'ICMRF[cojo]0000' and it computes $[\delta\mathbf{Y}]_{N_{\text{stop}}} = \mathbf{X}_{N_{\text{stop}}} - \mathbf{Y}_{N_{\text{stop}}}$ (sequence 'B00000000 STEPO').
- The sequence '0K0000000 STEPO' + '0GB0L0AA0 STEPOTL' + '0GB0L0AA0 STEPOAD' is called only if the cost-function is non-zero in a limited subdomain (**LOCNORM**=.T.).

- The vector of components $[\partial f/\partial y_i]_{N_{\rm stop}}$ is written on file 'MG[cojo]000+[Nstop]' (sequence 'A00000000 STEPO'). [Nstop] is coded on 4 digits.
- The model then runs $N_{\rm stop}$ sequences of type '0E0000000 STEPO + 0AAA00AAA STEPOAD' ('0E0000000 STEPO' converts the spectral trajectory into grid-point space, '0AAA00AAA STEPOAD' calls the adjoint model).
- The vector of components $[\partial f/\partial y_i]_0$ is written on file 'MG[cojo]000+0000' (sequence 'A00000000 STEPO').

3.3 Example of sequences of STEPO, STEPOTL and STEPOAD called in a configuration E801 run (LAM model) with $N_{\rm stop}$ timesteps.

Sequences are generally the same as for ARPEGE. Setting **LWREINI**=.T. (currently maintained in LAM models only) additionally writes $\mathbf{X}_{j=0} - R_{\rm dx}\mathcal{M}^*_{0->N_{\rm stop}}(\delta\mathbf{Y})_{N_{\rm stop}}$ on file 'ICMSH[cojo]+0000'. $R_{\rm dx}$ is a constant set-up in variable **RDX**.

4 Organigramme.

4.1 Features other than SIM4D.

Only the main features of the organigramme are described; the "deepest" routines are often omitted.

* CNT0:

```
CNT0 ->
 * SU0YOMA (0-level setup, part A) -> (comprehensive organigramme not detailed)
   - SUCT0 (reads NAMCT0)
   - SUVAR (reads NAMSENS, NAMVAR and NAMVRTL)
   - SUOPHY (reads NAMTRAJP)
 * SU0YOMB (0-level setup, part B) -> (comprehensive organigramme not detailed)
   - SULCZ or SUELCZ (reads NAMLCZ)
 * CGR1    (main control routine for conf 801) -> (see below)
```

* CGR1:

```
CGR1 ->
 * part 1:
   - ALLOCATE_CTLVEC
   - SU1YOM ->
     * SUINIF ->
       - SUGRIDF + SUGRIDU + SUGRIDO
       - SPECRT (global model) or ESPECRT (LAM model)
     * SUVAZX -> SUINIF + CHAVAR
     * SUSPEC + SUGRIDF + SUGRIDU (reading files)
     * SUCOS -> SUJR
 * part 2:
   - SUSLB if (if LETRAJP)
   - SUALLT (LAM models) -> ALLOCATE_TRAJECTORY
   - SIM4D -> (see below)
   - SCAAS
 * part 3:
   - GRTEST if LTEST -> (organigramme not detailed)
   - EWREINI if (LELAM and LWREINI) ->
     * ALLOCATE_CTLVEC
     * CAIN
     * STEP0('A00000000') -> (see section about STEP0)
     * DEALLOCATE_CTLVEC
 * final deallocation
   - DEALLOCATE_CTLVEC
```

* CNT3:

```
CNT3 ->
 * CSTA (if LNF, in global geometry only) -> does nothing in conf 801 !
 * ELSAC (in LAM model) -> (organigramme not detailed)
 * UPSPEC (if LOBSC1)
   - RDFPINC -> (organigramme not detailed)
   - STEP0('T00000000') -> (see section about STEP0)
 * DFI (if DFI initialisation, in ARPEGE only) -> (organigramme not detailed)
 * CNT4 ->
   - Prepare occurences of I/O events:
     * MONIO + MONVAR + (MOEVAR if LAM model)
   - Direct integration (features about diagnostics are omitted):
     * UPDTIM -> (organigramme not detailed)
     * SUHEG (for SI scheme)
     * SUHDU (for horizontal diffusion)
     * SURAND2
     * READ_TRAJECTORY + GET_TRAJ_SPEC
     * STEP0('TAAAOOAAA') -> (see section about STEP0)
```

* CNT3TL:

```
CNT3TL ->
 * SU3YOM
 * CSTA -> does nothing in conf 801 !
 * CNT4TL ->
   - Prepare occurences of I/O events:
     * MONIO + MONVAR + (MOEVAR if LAM model)
```

```
  - Direct + TL integration (features about diagnostics are omitted):
    * UPDTIM -> (organigramme not detailed)
    * SUHEG (for SI scheme)
    * SUHDU (for horizontal diffusion)
    * ELSINOTA -> (organigramme not detailed)
    * GET_TRAJ_SPEC
    * STEPO('B00000000') -> (see section about STEPO)
    * STEPO('0K0000000') -> (see section about STEPO)
    * STEPOTL('0GB0L0AA0') -> (see section about STEPO)
```

* CNT3AD:

```
CNT3AD ->
 * SU3YOM
 * CDSTA -> SUSPE0
 * CNT4AD ->
   - Prepare occurences of I/O events:
     * MONIO + MONVAR + (MOEVAR if LAM model)
     * STEPO('B00000000') -> (see section about STEPO)
     * COSTRA ->
       - COSJR (global model) or ECOSJR (LAM model) if NJROPT=1 ->
         (STEPO('0K0000000') + STEPOTL('0GB0L0AA0') + STEPOAD('0GB0L0AA0') if LOCNORM=T)
       - COSJL if NJROPT=2 -> (organigramme not detailed).
       - COSENS if NJROPT=3 -> (organigramme not detailed).
   - Adjoint integration (features about diagnostics are omitted):
     * GET_TRAJ_SPEC
     * UPDTIM -> (organigramme not detailed)
     * SUHEG (for SI scheme)
     * SUHDU (for horizontal diffusion)
     * ELSINOTA -> (organigramme not detailed)
     * STEPO('0E0000000') -> (see section about STEPO)
     * STEPOAD('0AAA00AAA') -> (see section about STEPO)
```

4.2 The simulator SIM4D.

This routine is the simulator needed for example in the 3D-VAR or in the 4D-VAR algorithm. It computes the cost-function and its gradient by calling the model and its adjoint. Some deepest callees are omitted (for example the organigramme of CHAVAR and CHAVARIN are not detailed). A detailed description of this simulator is given in documentation (IDVAR).

```
SIM4D ->
 * 1/ Initialize:
   - MONVAR
   - SU2YOM
   - SUCOS -> SUJR
   - FJVARBC
   - SUALLT
 * 2/ Fill SPA3 by the control variable and come back to physical space:
   - SUSPEC
   - ALLOCATE_CTLVEC + CHAVAR + CAIN + DEALLOCATE_CTLVEC
   - CHAVARIN
 * 3/ Direct integration:
   - CNT3 -> (see organigramme under CNT3)
 * 4/ Adjoint integration:
   - CNT3AD -> (see organigramme under CNT3AD)
 * 5/ Come back to control variable space:
   - CHAVARINAD
 * 6/ Sum up cost function, release space for trajectory:
   - DEALLT
   - EVCOST
   - SCAAS
```

4.3 Action of each routine.

- **ALLOCATE_CTLVEC**: allocation of **CONTROL_VECTOR** type variables.
- **ALLOCATE_TRAJECTORY**: allocation of some arrays used to store the trajectory needed for TL and AD model.
- **CAIN**: canonical injection.
- **CGR1**: controls the sensitivity job at level 1.
- **CHAVAR**: change of variable used for example in the variational assimilation.

- **CHAVARIN**: inverse action of **CHAVAR**.
- **CNT2**: control routine at level 2.
- **CNT3**: control routine at level 3.
- **COSENS**: computes the cost-function if **NJROPT**=3.
- **COSJL**: computes the cost-function if **NJROPT**=2.
- **COSJR** (**ECOSJR** for LAM model): computes the cost-function if **NJROPT**=1.
- **COSTRA**: computes the cost-function and its gradient.
- **CSTA**: control routine for reading the file of initial conditions.
- **DEALLOCATE_CTLVEC**: deallocation of **CONTROL_VECTOR** type variables.
- **DFI**: control routine for digital filter initialization.
- **EVCOST**: provides some diagnostics about the cost function.
- **EWREINI**: write perturbed initial file (LAM model).
- **FJVARBC**: cost associated with variational bias parameters.
- **GET_TRAJ_SPEC**: reads the trajectory needed for TL and AD model (spectral variables).
- **GRTEST**: test of the cost-function and its gradient.
- **LCNORTL**: operator to localize a perturbation in grid-point space.
- **MONIO**: management of the IO events.
- **MONVAR**: management of the 3-D/4-D VAR events.
- **RDFPINC**: reads in the TL trajectory of the model from ARPEGE/GRIB files.
- **READ_TRAJECTORY**: reads the trajectory needed for TL and AD model.
- **SCAAS**: computes the scalar product of two control vectors.
- **SIM4D**: simulator used for example in the 3DVAR and 4DVAR.
- **SPECRT**: computes spectral coefficients of virtual temperature from the spectral coefficients of temperature and moisture (can also do the reverse operation).
- **STEPO**: controls the different phases in a time step.
- **SU1YOM**: initialize level 1 setup.
- **SU2YOM**: initialize level 2 setup.
- **SUALLT**: allocate space for the grid-point and spectral trajectory.
- **SUCOS**: routine to initialize cost functions.
- **SUGRIDF**: reads the grid-point surface fields on a file.
- **SUGRIDU**: reads the grid-point upper air fields on a file.
- **SUGRIDO**: reads the ocean mixed layer model grid-point fields on a file.
- **SUHDU**: setup for the "unified scheme of horizontal diffusion".
- **SUHEG**: initialize the solver of the Helmholtz equation in the semi-implicit, in case of stretching and use of geographical divergence.
- **SUINIF**: routine to initialize the fields of the model.
- **SUJR**: routine to initialize cost function of module **YOMCOSJR**.
- **SURAND2**: initialize stochastic physic parameters: part 2.
- **SUSLB**: initialize pointers of **PTRSLB1** and **PTRSLB2**.
- **SUSPEC**: reads the spectral upper air fields on a file.
- **DEALLT**: release space for the grid-point and spectral trajectory.
- **UPDTIM**: update of the timestep.
- **UPSPEC**: update spectral arrays.

5 Modules and namelists.

These modules are auto-documented so description of each variable is provided in the code source. We can recall here the most important variables to know for each module:

- **YOMCT0** (0-level control): in particular NCONF, NFRHIS, NHISTS, LSIMOB. Some of these variables are in namelist **NAMCT0**.
- **YOMIOP** (trajectory for adjoint of physics): in particular NPCKFT95, NEXPBT95. These variables are in namelist **NAMTRAJP**.
- **YOMLCZ** (variables for Lanczos algorithm).
  - YVAZX0, YVAZG0.
  - LOCNORM, ALAT1, ALON1, ALAT3, ALON3, NLEVMIN, NLEVMAX, COEQTERM.
  - LSELU, LSELV, LSELT, LSELQ, LSELSP.

  Some of them are in the namelist **NAMLCZ**.
- **YOMSENS** (variables for sensitivity). These variables are in the namelist **NAMSENS**.
- **YOMVAR**: in particular LGRASCAL, RDX, NFRREF, NREFTS, LWREINI. Some of them are in namelist **NAMVAR**.
- **YOMVRTL**: Some of them are in namelist **NAMVRTL**.

6 An example of namelist.

Example taken assumes that complete physics is activated in the direct model, no physics is called in the adjoint and tangent linear model. The cost-function is quadratic (**NJROPT**=1) and is computed in all the atmosphere (**LOCNORM**=.F.). 5 timesteps are performed.

We can start from a configuration 1 namelist (Eulerian or SL2TL advection scheme). Such namelists are provided in documentations (IDEUL) and (IDSL). Some namelist elements must be modified:

- NAMCT0: add LGUESS=.FALSE., LOBS=.FALSE., LOBSC1=.FALSE., LSIMOB=.TRUE., NCNTVAR=1; one historic file at the final range must be asked for.
- No DDH, ISP, CFU, XFU diagnostics asked for; no post-processing asked for.
- NAMDYN: add LRDRY_ECMWF=.TRUE.,
- NAMLCZ, NAMSENS, NAMTRAJP, NAMVAR, NAPHLC must be filled (see below).

```
&NAMLCZ
  LOCNORM=.FALSE.,
  LSELU=.TRUE.,
  LSELV=.TRUE.,
  LSELT=.TRUE.,
  LSELQ=.TRUE.,
  LSELSP=.TRUE.,
  ALON1=355.0,
  ALAT1=50.0,
  ALON3=4.,
  ALAT3=43.0,
  NLEVMIN=1,
  NLEVMAX=31,
/

&NAMSENS
  LGRVOL=.TRUE.,
  NJROPT=1,
/

&NAMTRAJP
  NPCKFT95=1,
/

&NAMVAR
  LREFINC=.FALSE.,
  LTEST=.FALSE.,
  LGRASCAL=.TRUE.,
  RDX=0.1,
  NFRREF=1,
  NREFTS(0)=1,
  NREFTS(1)=5,
  LWREINI=.FALSE.,
/

&NAPHLC
  LSPHLC=.FALSE.,
  LVDFLC=.FALSE.,
  LVDFDS=.FALSE.,
  LSDRDS=.FALSE.,
  LZMCON=.FALSE.,
  LKEXP=.FALSE.,
/
```

In the command line NCONF must be set to 801.

The following files must be provided: ICMSH[cojo]INIT (initial file), ICMSH[cojo]IMIN (copy of ICMSH[cojo]INIT), ICMRF[cojo]0000 (forecast or analysis at timestep N_{stop}).

7 References and internal notes.

7.1 Publications.

- Courtier, Ph., C. Freydier, J.F. Geleyn, F. Rabier and M. Rochas, 1991: The ARPEGE project at METEO-FRANCE. ECMWF Seminar Proceedings 9-13 September 1991, Volume II, 193-231.
- Rabier, F., E. Klinker, Ph. Courtier and A. Hollingsworth, 1996: Sensitivity of forecast errors to initial conditions. *Quart. J. Roy. Meteor. Soc.*, **122**, 121-150.
- Soci, C., A. Horányi and C. Fischer, 2003: Preliminary results of high resolution sensitivity studies using the adjoint of the ALADIN mesoscale numerical weather prediction model. *Quarterly Journal of the Hungarian Meteorological Service*, **107 nr 1**, 49-65.

7.2 Some internal notes and other ARPEGE notes.

- (TDECDAS) 2012: IFS technical documentation (CY38R1). Part II: data assimilation.
- (TDECTEC) 2012: IFS technical documentation (CY38R1). Part VI: technical and computational procedures. Available at "http://www.ecmwf.int/research/ifsdocs/".
- (ID801O) Bouttier, F., 1995: Sensitivity jobs with configuration 801. Internal documentation valid for cycle 13r3, 13pp, available on the intranet server " http://gco.meteo.fr/ifsdoc/html/".
- (IDVAR) Fischer, C., and L. Berre, 2007: The variational computations inside ARPEGE/ALADIN: cycle CY32. Internal note, 77pp. Available on the intranet server "http://www.cnrm.meteo.fr/gmapdoc/".
- (NTA28) Rabier, F., Ph. Courtier, O. Talagrand et J.F. Geleyn, 1992: Two applications of adjoint models to sensitivity analysis. Note de travail ARPEGE numéro **28**, 44pp.
- (IDBAS) Yessad, K., 2013: Basics about ARPEGE/IFS, ALADIN and AROME in the cycle 40 of ARPEGE/IFS (internal note).
- (IDEUL) Yessad, K., 2013: Integration of the model equations, and Eulerian dynamics, in the cycle 40 of ARPEGE/IFS (internal note).
- (IDDM) Yessad, K., 2013: Distributed memory features in the cycle 40 of ARPEGE/IFS (internal note).
- (IDMINI) Yessad, K., 2013: Minimizations in the cycle 40 of ARPEGE/IFS (internal note).
- (IDSL) Yessad, K., 2013: Semi-Lagrangian computations in the cycle 40 of ARPEGE/IFS. Internal note, available on "http://www.cnrm.meteo.fr/gmapdoc/" (topics "Dynamics").