CONFIGURATION 801 IN THE CYCLE 39 OF ARPEGE/IFS.

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

November 28, 2012

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. **3**

2 Algorithm of this configuration. **4**
2.1 Notations. . . . 4
2.2 Case without any minimization. . . . 4
2.3 Case with minimization. . . . 5
2.4 Aspect of the function f. . . . 5

3 The sequences of STEPO called. **6**
3.1 General architecture under STEPO. . . . 6
3.2 Example of sequences of STEPO, STEPOTL and STEPOAD called in a configuration 801 ARPEGE/IFS run with N_{stop} timesteps. . . . 7
3.3 Example of sequences of STEPO, STEPOTL and STEPOAD called in a configuration 801 ALADIN run with N_{stop} timesteps. . . . 9

4 Organigramme. **10**
4.1 Features other than SIM4D. . . . 10
4.2 The simulator SIM4D. . . . 12
4.3 Action of each routine. . . . 12

5 Modules and namelists. **14**

6 Example of namelist. **15**

7 References and internal notes. **23**
7.1 Publications. . . . 23
7.2 Some internal notes and other ARPEGE notes. . . . 23

1 Introduction and purpose.

This configuration is used to do sensitivity studies and can help to answer to the two following questions:

- 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?
- One does a forecast ("experiment forecast"), and uses a "reference forecast". The question is: what is the perturbation to bring to the initial state, in order the model forecast to get the closest as possible as the "reference forecast"?

At least the adjoint code is used in this configuration, and some particular options of this configuration also use the tangent linear code. 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 39 and cycle AL39. One currently tries to give a more detailed updated description of this configuration.

∗ **Modifications since cycle 38:** none

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 first 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, and only the direct and adjoint codes are used, excepted for particular applications where the cost function is defined on a limited area (in this case the tangent linear code may be sometimes called).
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 2.4 for examples of available functions in ARPEGE/IFS and ALADIN. 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 $\mathcal{M}_{0->N_{\rm stop}}$.

2.3 Case with minimization.

We now want to answer the second question: what is the perturbation to bring to an initial state, in order the model forecast to get the closest as possible as the "reference forecast"? One follows the previous algorithm, but additionally an iterative scheme is performed as follows:

- define a perturbation $[\delta\mathbf{X}]_0(iter=0)$.
- run the direct model, taking now as initial state $\mathbf{X}_0+[\delta\mathbf{X}]_0(iter=0)$, then the adjoint model.
- computation of f and its gradient also to determine the next value $[\delta\mathbf{X}]_0(iter=1)$ to test.
- run the direct model, taking now as initial state $\mathbf{X}_0+[\delta\mathbf{X}]_0(iter=1)$, then the adjoint model.
- etc...

The algorithm stops when $[\delta\mathbf{X}]_0(iter)$ is close enough to the searched solution or when the maximum number of iterations we have previously pre-defined is reached. Note that one can replace at this stage the direct integration by a tangent linear integration from the initial perturbation $[\delta\mathbf{X}]_0(iter)$. This algorithm looks like a 4D-VAR problem but it is simpler: no observation, the cost-function has only one component and the guess is replaced by a forecast (or an analysis) at instant $t_{N_{\rm stop}}$ (that means in particular that contrary to the 4D-VAR assimilation there is no interpolation at some observation points which are not points of the Gaussian grid).

2.4 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:

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

- **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.

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

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

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 ->
* Gestion 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 controls the inverse Legendre transforms.
- L3 controls the inverse Fourier 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 controls the direct Fourier transforms.
- L8 controls the direct Legendre 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.

∗ **Case L801TL=.F.; LMINI=.F.:**

```
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 $[\partial f/\partial y_i]_{N_{\text{stop}}}$ is written on file 'MG[cojo]000+[Nstop]' (sequence 'A00000000 STEPO'). [Nstop] is coded on 4 digits.
- The model then runs N_{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').

∗ **Case L801TL=.T.; LMINI=.F.:** This case seems have no significant interest compared to the previous one; the only differences noticed are that the sequence 'B00000000 STEPO' is replaced by a sequence 'V00000000 STEPO' and that the sequence 'A00000000 STEPO' writing $[\partial f/\partial y_i]_{N_{\text{stop}}}$ on file 'MG[cojo]000+[Nstop]' disappears.

∗ Case L801TL=.F.; LMINI=.T.:

```
SIMUL_0 (SIM4D called by CGR1):
 -----------------
 TAAA00AAA STEP0    (jstep=1)
 ..................
 TAAA00AAA STEP0    (jstep=Nstop)
 -----------------
 B00000000 STEP0
 -----------------
 0K0000000 STEP0
 0GB0L0AA0 STEP0TL
 0GB0L0AA0 STEP0AD
 -----------------
 A00000000 STEP0
 -----------------
 0E0000000 STEP0    (jstep=Nstop)
 0AAA00AAA STEP0AD
 ..................
 0E0000000 STEP0    (jstep=1)
 0AAA00AAA STEP0AD
 -----------------
 A00000000 STEP0
 000000000 CNT0
 -----------------
SIMUL_1 to SIMUL_Nsimul (SIM4D called by the minimizer):
do jsimul=1 to Nsimul
 -----------------
 TAAA00AAA STEP0    (jstep=1)
 ..................
 TAAA00AAA STEP0    (jstep=Nstop)
 -----------------
 B00000000 STEP0
 -----------------
 0K0000000 STEP0
 0GB0L0AA0 STEP0TL
 0GB0L0AA0 STEP0AD
 -----------------
 A00000000 STEP0
 -----------------
 0E0000000 STEP0    (jstep=Nstop)
 0AAA00AAA STEP0AD
 ..................
 0E0000000 STEP0    (jstep=1)
 0AAA00AAA STEP0AD
 -----------------
 A00000000 STEP0
 -----------------
enddo
```

The sequences under 'SIMUL_0' are identical to those done in the case **LMINI**=.T., but there is an additional minimization (minimizing the function f) which launches N_{simul} times the same list of sequences. If the function f has a quadratic expression of $[\delta\mathbf{X}]_{N_{\text{stop}}}$, it has no longer a quadratic expression of the unknown $[\delta\mathbf{X}]_0$ because the model is non-linear. So the minimizer which is used must be valid to minimize non-quadratic functions (the code currently sticks to the minimizer **M1QN3** which uses a quasi-Newton method and can be applied to non-quadratic functions). For more details about the algorithms of minimization, see documentation (IDMINI). Note that the number of simulations N_{simul} can be above the number of iterations of the minimization (this is generally the case for **M1QN3**).

∗ Case L801TL=.T.; LMINI=.T.:

```
SIMUL_0 (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
 -----------------
SIMUL_1 to SIMUL_Nsimul (SIM4D called by the minimizer):
do jsimul=1 to Nsimul
 -----------------
 0E0000000 STEPO    (jstep=1)
 0AAA00AAA STEPOTL
 ..................
 0E0000000 STEPO    (jstep=Nstop)
 0AAA00AAA STEPOTL
 -----------------
 V00000000 STEPO
 -----------------
 0K0000000 STEPO
 0GB0L0AA0 STEPOTL
 0GB0L0AA0 STEPOAD
 -----------------
 A00000000 STEPO
 -----------------
 0E0000000 STEPO    (jstep=Nstop)
 0AAA00AAA STEPOAD
 ..................
 0E0000000 STEPO    (jstep=1)
 0AAA00AAA STEPOAD
 -----------------
 A00000000 STEPO
 -----------------
enddo
```

The main differences compared to the case (**L801TL**=.T.; **LMINI**=.T.) are that the sequences 'TAAA00AAA STEPO' calling the direct model have been replaced by sequences '0E0000000 STEPO' (transformation of the trajectory into grid-point space) + '0AAA00AAA STEPOTL' (tangent-linear model), and that the sequence 'B00000000 STEPO' has been replaced by 'V00000000 STEPO'. If the function f has a quadratic expression of $[\delta\mathbf{X}]_{N_{\rm stop}}$, it has also a quadratic expression of $[\delta\mathbf{X}]_0$: that would be enable to use efficient minimizers valid to quadratic functions. The code currently sticks to the minimizer **M1QN3** which uses a quasi-Newton method and can be applied to non-quadratic functions. The solution found for $[\delta\mathbf{X}]_0$ assumes that the model is quasi-linear, and is less exact that the one found with **L801TL**=.F. . The only interest to use (**L801TL**=.T.; **LMINI**=.T.) seems to find $[\delta\mathbf{X}]_0$ more efficiently.

3.3 Example of sequences of STEPO, STEPOTL and STEPOAD called in a configuration 801 ALADIN run with $N_{\rm stop}$ timesteps.

Sequences are generally the same as for ARPEGE. Setting **LWREINI**=.T. (currently maintained in ALADIN 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' (this works at least for case **LMINI**=.F.). $R_{\rm dx}$ is a constant set-up in variable **RDX**. Cases **L801TL**=.T. and **LMINI**=.T. have not be recently validated.

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)
   - SUOPHY (reads NAMTRAJP)
   - SUVAR (reads NAMSENS, NAMVAR and NAMVRTL)
 * 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 (ARPEGE) or ESPECRT (ALADIN)
     * SUVAZX -> SUINIF + CHAVAR
     * SUSPEC + SUGRIDF + SUGRIDU if .NOT.L801TL (reading files)
     * SUCOS -> SUJR
 * part 2:
   - SUALLT7 if L801TL
   - CNT2 if L801TL ->
     * SU2YOM
     * SUALLT -> ALLOCATE_TRAJECTORY
     * CNT3 ->
       - CSTA (if LNF)
       - UPSPEC (if LOBSC1)
       - DFI (if DFI initialisation)
       - CNT4 -> (see below)
   - SUSLB if (L801TL or LETRAJP)
   - SUALLT -> ALLOCATE_TRAJECTORY
   - SIM4D -> (see below)
   - SCAAS
 * part 3:
   - GRTEST if LTEST -> (organigramme not detailed)
   - EWREINI if (LELAM and LWREINI) ->
     * ALLOCATE_CTLVEC
     * CAIN
     * STEPO('A00000000') -> (see section about STEPO)
     * DEALLOCATE_CTLVEC
 * part 4 if (LMINI and .NOT.L801TL)
   - ALLOCATE_CTLVEC
   - SUHESS -> (organigramme not detailed)
   - M1QN3 (Quasi Newton minimisation)
   - CAIN
   - STEPO('A00000000') -> (see section about STEPO)
   - SIM4D -> (see below)
   - SCAAS
   - DEALLOCATE_CTLVEC
 * part 5 if (LMINI and L801TL)
   - ALLOCATE_CTLVEC
   - SUHESS -> (organigramme not detailed)
   - M1QN3 (Quasi Newton minimisation)
   - SCAAS
   - GRTEST -> (organigramme not detailed)
   - DEALLOCATE_CTLVEC
 * final deallocation
   - DEALLOCATE_CTLVEC
```

* CNT3:

```
CNT3 ->
 * CSTA (if LNF, in ARPEGE only) -> SUINIF ->
   - SUGRIDF + SUGRIDU + SUGRIDO
   - SPECRT (ARPEGE) or ESPECRT (ALADIN)
 * ELSAC (in ALADIN) -> (organigramme not detailed)
 * UPSPEC (if LOBSC1)
   - RDFPINC -> (organigramme not detailed)
   - STEPO('T00000000') -> (see section about STEPO)
 * DFI (if DFI initialisation, in ARPEGE only) -> (organigramme not detailed)
 * CNT4 ->
   - Prepare occurences of I/O events:
     * MONIO + MONVAR + (MOEVAR if ALADIN)
   - Direct integration (features about diagnostics are omitted):
     * SUALLAVT (if LAVARC and LLWRTRA)
     * UPDTIM -> (organigramme not detailed)
     * SUNHSI, SUHEG (for SI scheme)
     * SUHDU (for horizontal diffusion)
     * SURAND2
     * AVARCT -> SUSPEC
     * READ_TRAJECTORY + GET_TRAJ_SPEC
     * STEPO('TAAAOOAAA') -> (see section about STEPO)
```

* CNT3TL:

```
CNT3TL ->
 * SU3YOM
 * CSTA (if LNF) -> SUINIF ->
   - SUGRIDF + SUGRIDU + SUGRIDO
   - SPECRT (ARPEGE) or ESPECRT (ALADIN)
 * CNT4TL ->
   - Prepare occurences of I/O events:
     * MONIO + MONVAR + (MOEVAR if ALADIN)
   - Direct + TL integration (features about diagnostics are omitted):
     * UPDTIM -> (organigramme not detailed)
     * SUNHSI, SUHEG (for SI scheme)
     * ELSINOTA -> (organigramme not detailed)
     * SUHDU (for horizontal diffusion)
     * 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 ALADIN)
     * STEPO('B00000000') -> (see section about STEPO)
     * COSTRA ->
       - COSJR (ARPEGE) or ECOSJR (ALADIN) 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)
     * SUNHSI, 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
   - SBSFGS (if L801TL)
   - ALLOCATE_CTLVEC + CAININ + CHAVARIN + DEALLOCATE_CTLVEC (if L801TL and LBACKG)
   - STEPO('A00000000') -> (see section about STEPO)
 * 3/ Tangent linear integration if L801TL:
   - ALLOCATE_CTLVEC + LCNORTL + CAIN + DEALLOCATE_CTLVEC
   - CNT3TL -> (see organigramme under CNT3TL)
 * 3b/ Direct integration if .NOT.L801TL:
   - 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.
- **AVARCT**: used in variational assimilation of corrections.
- **CAIN**: canonical injection.
- **CAININ**: inverse of 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 ALADIN): 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 inital file (ALADIN).
- **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.
- **M1QN3**: minimizer using a quasi-Newton algorithm (can be used to minimize non-quadratic functions).
- **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.
- **SUALLT7**: allocate space for the spectral trajectory of **YOMSPT7**.
- **SUALLAVT**: allocate tables for **AVARC** trajectory minus reference to store them for the minimization.
- **SBSFGS**: substracts first guess in spectral space to **SPA3**, **SPA2** and **SPA1** arrays.
- **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 Helmholz equation in the semi-implicit, in case of stretching and use of geographical divergence.
- **SUHESS**: reads the pre-conditioner used in the minimizers **M1QN3** or **N1CG1**.
- **SUINIF**: routine to initialize the fields of the model.
- **SUJR**: routine to initialize cost function of module **YOMCOSJR**.
- **SUNHSI**: initialize the solver of the Helmholz equation in the NH model.
- **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 particuliar NCONF, NFRHIS, NHISTS, LSIMOB. Some of these variables are in namelist **NAMCT0**.
- **YOMIOP** (trajectory for adjoint of physics): in particuliar 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 particuliar LGRASCAL, RDX, NFRREF, NREFTS, LWREINI. Some of them are in namelist **NAMVAR**.
- **YOMVRTL**: in particuliar L801TL. Some of them are in namelist **NAMVRTL**.

6 Example of namelist.

The following namelist is valid for an ARPEGE job with 5 timesteps, resolution TL31L70c1 (linear grid), two-time level semi-Lagrangian advection scheme, no minimization. Complete physics is activated in the direct model (it matches with the physics package which was in the operational suite on 01/10/2012), no physics in the adjoint and tangent linear model. The cost-function is quadratic (**NJROPT**=1) and is computed in all the atmosphere (**LOCNORM**=.F.).

```
&NACIETEO
/
&NACOBS
/
&NACTAN
/
&NACTEX
/
&NACVEG
/
&NADOCK
/
&NAEAEM7
/
&NAEAER
/
&NAECOAPHY
/
&NAEPHLI
/
&NAEPHY
/
&NAERAD
  LRRTM=.TRUE.,
  LSRTM=.FALSE.,
  NRADFR=-3,
  NSW=6,
  RLWINHF=0.9,
/
&NAEVOL
/
&NAIMPO
/
&NALORI
/
&NAMAFN
  TFP_I%LLGP=.TRUE.,
  TFP_L%LLGP=.TRUE.,
/
&NAMARPHY
/
&NAMCA
/
&NAMCAPE
/
&NAMCFU
  LCUMFU=.TRUE.,
  LFDUTP=.TRUE.,
  LFPLC=.TRUE.,
  LFPLS=.TRUE.,
  LFR=.TRUE.,
  LFSF=.TRUE.,
  LFSOIL=.TRUE.,
  LMOON=.TRUE.,
  LNEBPAR=.TRUE.,
  LSTRD=.TRUE.,
  LSTRT=.TRUE.,
  LFRRC=.TRUE.,
  LRAYD=.TRUE.,
  NFRCFU=1,
  NCFUTS(0)=-1,
  NCFUTS(1)=-97,
/
&NAMCHEM
/
&NAMCHET
/
&NAMCHK
/
&NAMCLA
/
&NAMCLDP
/
&NAMCLI
/
&NAMCLOP15
/
```

```
&NAMCLTC
/
&NAMCOK
/
&NAMCOM
/
&NAMCOSJO
/
&NAMCOUPLO4
/
&NAMCTO
  LARPEGEF=.TRUE.,
  LFBDAP=.FALSE.,
  LFDBOP=.FALSE.,
  LGUESS=.FALSE.,
  LOBS=.FALSE.,
  LOBSC1=.FALSE.,
  LSIMOB=.TRUE.,
  LTWOTL=.TRUE.,
  LSPRT=.TRUE.,
  LRETCFOU=.FALSE.,
  LWRTCFOU=.FALSE.,
  LRFOUTCNORM(1)=.FALSE.,
  LRGPTCNORM(1)=.FALSE.,
  NCNTVAR=1,
  NDHPTS(0)=-1,-999,
  NFRDHP=1,
  NFRHIS=1,
  NFRPOS=10000,
  NFRSDI=1,
  NHISTS(0)=2,
  NHISTS(1)=0,
  NHISTS(2)=5,
  NPOSTS(0)=1,
  NPOSTS(1)=1,
  NSDITS(0)=2,
  NSDITS(1)=0,
  NSDITS(2)=5,
  NSPPR=1,
  NFRISP=1,
  NPISPS(0)=-1,
  NPISPS(1)=-999,
  LALLOPR=.FALSE.,
  LOPT_SCALAR=.FALSE.,
  LREGETA=.FALSE.,
  LVFE_REGETA=.FALSE.,
  NPRINTLEV=0,
/
&NAMCT1
  LRFILAF=.FALSE.,
  N1POS=0,
  N1RES=0,
/
&NAMCUMF
/
&NAMCUMFS
/
&NAMCVER
  LVERTFE=.TRUE.,
  NVSCH=3,
/
&NAMCVMNH
  OTADJS=10800.,
  XCDEPTH=1.,
  XCDEPTH_D=4000.,
  XDTPERT=0.3,
  XENTR=0.013,
/
&NAMDDH
/
&NAMDFI
/
&NAMDIM
  NPROMA=-469,
  NTCMAX=6,
/
&NAMDIMO
/
&NAMDPHY
/
&NAMDYN
  BETADT=1.,
  RDAMPDIV=2.,
  LDRY_ECMWF=.TRUE.,
/
&NAMDYNA
```

```
  LAPRXPK=.TRUE.,
/
&NAMDYNCORE
/
&NAMEMIS_CONF
/
&NAMENKF
/
&NAMFA
  NBITCS=30,
  NBITPG=30,
  NSTRON=-1,
  YFAI%NBITS=16,
  YFAL%NBITS=16,
  YFAR%NBITS=16,
  YFAS%NBITS=16,
  YFALRAD%NBITS=16,
  YFAIRAD%NBITS=16,
  YFACLF%NBITS=6,
  YFATKE%NBITS=16,
/
&NAMFPC
/
&NAMFPD
/
&NAMFPDY2
/
&NAMFPDYH
/
&NAMFPDYI
/
&NAMFPDYP
/
&NAMFPDYS
/
&NAMFPDYT
/
&NAMFPDYV
/
&NAMFPF
/
&NAMFPG
/
&NAMFPIOS
/
&NAMFPMOVE
/
&NAMFPPHY
/
&NAMFPSC2
/
&NAMFPSC2_DEP
/
&NAMGEM
  REFLKUO=10000.,
/
&NAMGFL
  YI_NL%LGPINGP=.TRUE.,
  YI_NL%LGP=.TRUE.,
  YI_NL%LT1=.TRUE.,
  YI_NL%LPHY=.FALSE.,
  YI_NL%NREQIN=1,
  YI_NL%LREQOUT=.FALSE.,
  YI_NL%LADV=.FALSE.,
  YI_NL%LQM=.FALSE.,
  YL_NL%LGPINGP=.TRUE.,
  YL_NL%LGP=.TRUE.,
  YL_NL%LT1=.TRUE.,
  YL_NL%LPHY=.FALSE.,
  YL_NL%NREQIN=1,
  YL_NL%LREQOUT=.FALSE.,
  YL_NL%LADV=.FALSE.,
  YL_NL%LQM=.FALSE.,
  YR_NL%LGPINGP=.TRUE.,
  YR_NL%LGP=.TRUE.,
  YR_NL%LT1=.TRUE.,
  YR_NL%LPHY=.FALSE.,
  YR_NL%NREQIN=1,
  YR_NL%LREQOUT=.FALSE.,
  YR_NL%LADV=.FALSE.,
  YR_NL%LQM=.FALSE.,
  YS_NL%LGPINGP=.TRUE.,
  YS_NL%LGP=.TRUE.,
```

```
  YS_NL%LT1=.TRUE.,
  YS_NL%LPHY=.FALSE.,
  YS_NL%NREQIN=0,
  YS_NL%LREQOUT=.FALSE.,
  YS_NL%LADV=.FALSE.,
  YS_NL%LQM=.FALSE.,
  YTKE_NL%LGPINGP=.TRUE.,
  YTKE_NL%LGP=.TRUE.,
  YTKE_NL%LT1=.TRUE.,
  YTKE_NL%NREQIN=0,
  YTKE_NL%LREQOUT=.FALSE.,
  YTKE_NL%NCOUPLING=0,
  YTKE_NL%LADV=.FALSE.,
  YTKE_NL%LQM=.TRUE.,
  YIRAD_NL%LGP=.TRUE.,
  YIRAD_NL%NREQIN=0,
  YIRAD_NL%LREQOUT=.TRUE.,
  YLRAD_NL%LGP=.TRUE.,
  YLRAD_NL%NREQIN=0,
  YLRAD_NL%LREQOUT=.TRUE.,
  YA_NL%LGP=.TRUE.,
  YA_NL%NREQIN=0,
  YA_NL%LREQOUT=.TRUE.,
/
&NAMGRIB
/
&NAMGWD
/
&NAMGWWMS
/
&NAMHCP
/
&NAMICE
/
&NAMINI
  LDFI=.FALSE.,
/
&NAMIOMI
/
&NAMIOS
/
&NAMIO_SERV
/
&NAMJBCODES
/
&NAMJFH
/
&NAMJG
/
&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,
/
&NAMLSFORC
/
&NAMMARS
/
&NAMMCC
/
&NAMMCUF
/
&NAMMKODB
/
&NAMMODERR
/
&NAMMTS
/
&NAMMWAVE
/
&NAMNPROF
/
&NAMNUD
/
&NAMOBS
/
```

```
&NAMONEDVAR
/
&NAMOPH
  LINC=.TRUE.,
  CFNHWF='ECHIS',
/
&NAMOPTCMEM
/
&NAMPAR0
  NPROC=1,
  NPRGPNS=1,
  NPRTRW=1,
  NPRGPEW=1,
  NPRTRV=1,
  NOUTPUT=1,
  MP_TYPE=2,
  MBX_SIZE=128000000,
/
&NAMPAR1
  LSPLIT=.FALSE.,
  NSTRIN=1,
  NSTROUT=1,
  NCOMBFLEN=1638400,
  LEQ_REGIONS=.FALSE.,
/
&NAMPARAR
/
&NAMPHMSE
/
&NAMPHY
  CGMIXLEN='AY',
  LAERODES=.TRUE.,
  LAEROLAN=.TRUE.,
  LAEROSEA=.TRUE.,
  LAEROSOO=.TRUE.,
  LCONDWT=.TRUE.,
  LDIFCONS=.TRUE.,
  LFPCOR=.TRUE.,
  LNEWD=.TRUE.,
  LNOIAS=.TRUE.,
  LO3ABC=.TRUE.,
  LPROCLD=.TRUE.,
  LRAY=.FALSE.,
  LRAYFM=.TRUE.,
  LRAYLU=.TRUE.,
  LRNUMX=.TRUE.,
  LSSD=.TRUE.,
  LSTRA=.FALSE.,
  LVGSN=.TRUE.,
  LCVPPKF=.TRUE.,
  LECDEEP=.TRUE.,
  LECSHAL=.TRUE.,
  LECT=.TRUE.,
  LFLUSO=.TRUE.,
  LNEBECT=.FALSE.,
  LO3FL=.TRUE.,
  LECTFL=.TRUE.,
  LZOHSREL=.TRUE.,
  LADJCLD=.TRUE.,
  LSMITH_CDEV=.TRUE.,
  NCALLRAD=2,
  NDPSFI=0,
/
&NAMPHY0
  EDD=1.,
  EDK=1.,
  GCVNU=5.E-05,
  GCVPSI=1.,
  GCVPSIE=1.,
  GWDCD=5.4,
  GWDSE=0.005,
  GWDVALI=0.5,
  QSNEBC=-1.,
  QSSUSC=5.,
  RCVEVAP=0.25,
  REVASX=2.E-07,
  RICRLM=0.5,
  TDDGP=0.6,
  TUDGP=0.6,
  USURIC=0.175,
  USURICL=1.,
  USURID=0.1,
  VZOCM=1.0E-4,
  XBLM=8.5,
  XMAXLM=5000.,
  XMINLM=10.,
  ALMAVX=1000.,
  GCVHMIN=30000.,
```

```
  RFACNSM=1.2,
  RFLCHCE=0.25,
  RKFBTAU=3600.,
  RPRTH=1.,
  RQICRT2=0.,
  RQICVMIN=1.E-5,
  SXNBCO=1.,
  ALMAVE=0.,
  RQCRNS=0.,
  RQICRSN=1.,
  TFVI=0.08,
  TFVL=0.02,
  TFVS=1.5,
  GCVOMGE=1.0,
  GCVOMGQ=-1.0,
  ECTMAX=35.,
/
&NAMPHY1
  ALBMIN=0.65,
  ALCRIN=0.75,
  EMMGLA=0.98,
  EMMMER=0.99,
/
&NAMPHY2
  FACRAF=4.5,
  XMULAF=0.,
  LRAFTKE=.TRUE.,
  HTKERAF=20.0,
/
&NAMPHY3
/
&NAMPHYDS
/
&NAMPONG
/
&NAMPPC
/
&NAMPRE
/
&NAMRAD15
/
&NAMRADCMEM
/
&NAMRCF
/
&NAMRCOEF
/
&NAMRES
/
&NAMRGRI
/
&NAMRINC
/
&NAMRIP
/
&NAMSATS
/
&NAMSCC
/
&NAMSCEN
/
&NAMSCM
/
&NAMSEKF
/
&NAMSENS
  LGRVOL=.TRUE.,
  NJROPT=1,
/
&NAMSIMPHL
/
&NAMSPNG
/
&NAMSPSDT
/
&NAMSTA
/
&NAMSTOPH
/
&NAMSWE
/
&NAMTESTVAR
/
&NAMTOPH
  ETCVIM=5000.,
  ETNEBU=5000.,
  ETPLUI=5000.,
  XDRMTK=2.E-08,
```

```
  XDRMTP=800.,
  XDRMTX=4.E-07,
  XDRMUK=1.E-07,
  XDRMUP=800.,
  XDRMUX=2.E-06,
/
&NAMTRAJP
  NPCKFT95=1,
/
&NAMTRANS
/
&NAMTS
/
&NAMVAR
  LREFINC=.FALSE.,
  LTEST=.FALSE.,
  LGRASCAL=.TRUE.,
  RDX=0.1,
  NFRREF=1,
  NREFTS(0)=1,
  NREFTS(1)=5,
  LWREINI=.FALSE.,
/
&NAMVARBC
/
&NAMVARBC_AIREP
/
&NAMVARBC_ALLSKY
/
&NAMVARBC_GBRAD
/
&NAMVARBC_RAD
/
&NAMVARBC_SFCOBS
/
&NAMVARBC_TCWV
/
&NAMVARBC_TO3
/
&NAMVAREPS
/
&NAMVDOZ
/
&NAMVRTL
  L801TL=.FALSE.,
/
&NAMVV0
/
&NAMVV1
/
&NAMVWRK
/
&NAMWAVELETJB
/
&NAMXFU
  LXCLP=.TRUE.,
  LXCLS=.TRUE.,
  LXFU=.TRUE.,
  LXHHCLS=.TRUE.,
  LXICV=.TRUE.,
  LXNEBPA=.TRUE.,
  LXNEBTT=.TRUE.,
  LXQCLS=.TRUE.,
  LXSOIL=.TRUE.,
  LXTGST=.TRUE.,
  LXTTCLS=.TRUE.,
  LXXGST=.TRUE.,
  NFRXFU=1,
  NXFUTS(0)=-1,
  NXFUTS(1)=-97,
/
&NAM_CANAPE
/
&NAM_DISTRIBUTED_VECTORS
/
&NAPHLC
  LSPHLC=.FALSE.,
  LVDFLC=.FALSE.,
  LVDFDS=.FALSE.,
  LSDRDS=.FALSE.,
  LZMCON=.FALSE.,
  LKEXP=.FALSE.,
/
&NEMCT0
/
&NEMDIM
/
```

```
 &NEMDYN
 /
 &NEMELBC0A
 /
 &NEMELBC0B
 /
 &NEMFPEZ0
 /
 &NEMGEO
 /
 &NEMJK
 /
 &NEMVAR
 /
 &NEMWAVELET
 /
```

In the command line one finds:

```
NCONF=801
VERSION=meteo
CNMEXP=[cojo]
TSTEP=900.
NSTOP=t5
ADVEC=sli
```

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 (CY37R2). Part II: data assimilation.
- (TDECTEC) 2012: IFS technical documentation (CY37R2). 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., 2012: Basics about ARPEGE/IFS, ALADIN and AROME in the cycle 39 of ARPEGE/IFS (internal note).
- (IDEUL) Yessad, K., 2012: Integration of the model equations, and Eulerian dynamics, in the cycle 39 of ARPEGE/IFS (internal note).
- (IDDM) Yessad, K., 2012: Distributed memory features in the cycle 39 of ARPEGE/IFS (internal note).
- (IDMINI) Yessad, K., 2012: Minimizations in the cycle 39 of ARPEGE/IFS (internal note).