

# BASICS ABOUT ARPEGE/IFS, ALADIN AND AROME IN THE CYCLE 45 OF ARPEGE/IFS.

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

June 28, 2017

ARPEGE/IFS, ALADIN AND AROME FOR DUMMIES  
ARPEGE/IFS, ALADIN ET AROME POUR LES NULS

## *Abstract:*

*This documentation has the aim to give a general overview of the ARPEGE/IFS, ALADIN and AROME models, for beginners or for someone which knows some specific points but has no general overview of the whole models. It can be seen as a synthetic sum-up of some other documentations, and it contains also a list of references as complete as possible which form the set of documentations.*

## *Résumé:*

*Cette documentation a pour but de donner un aperçu général de ce que sont les modèles ARPEGE/IFS, ALADIN et AROME et de leur contenu, sans trop entrer dans les détails, et du même coup va servir de documentation d'introduction à diverses documentations beaucoup plus détaillées et beaucoup plus ciblées aussi qui existent déjà. Cela sera en quelque sorte un résumé le plus synthétique possible de la documentation existante. Cette documentation généraliste constitue une sorte d'“ARPEGE pour les nuls” et est destinée en priorité aux débutants et aussi à ceux déjà pratiquants qui ont une connaissance très parcellaire de ces modèles et une vision d'ensemble restant à acquérir.*

# Contents

<b>1</b>	<b>Introduction and general purpose of this documentation.</b>	<b>4</b>
<b>2</b>	<b>A brief description of ARPEGE, ALADIN, AROME.</b>	<b>5</b>
<b>3</b>	<b>The list of available configurations and some basics about the architecture of the code.</b>	<b>6</b>
3.1	The list of available configurations. . . . .	6
3.2	Basics about the architecture of the code. . . . .	7
<b>4</b>	<b>The different projects.</b>	<b>8</b>
<b>5</b>	<b>The sequences in a forecast suite.</b>	<b>9</b>
5.1	ARPEGE forecast suite. . . . .	9
5.2	AROME forecast suite. . . . .	10
<b>6</b>	<b>Basics about geometry.</b>	<b>11</b>
6.1	Global model ARPEGE/IFS. . . . .	11
6.2	Limited area model ALADIN/AROME. . . . .	11
<b>7</b>	<b>The forecast: configurations 1, 201, 202.</b>	<b>12</b>
<b>8</b>	<b>Basics about the physics.</b>	<b>16</b>
<b>9</b>	<b>Basics about dataflow.</b>	<b>19</b>
<b>10</b>	<b>The tangent linear (TL) and adjoint (AD) codes.</b>	<b>20</b>
10.1	The tangent linear (TL) code. . . . .	20
10.2	The adjoint (AD) code. . . . .	21
<b>11</b>	<b>The diagnostics.</b>	<b>22</b>
11.1	Introduction. . . . .	22
11.2	FULL-POS. . . . .	22
11.3	DDH. . . . .	24
11.4	CFU and XFU. . . . .	25
11.5	CHKEVO. . . . .	25
11.6	Physical tendencies diagnostics. . . . .	25
11.7	MCUF diagnostics (monitoring the coupling update frequency). . . . .	26
11.8	SURFEX diagnostics. . . . .	26
11.9	SLDIA diagnostics. . . . .	26
11.10	Spectral and grid-point norms. . . . .	26
<b>12</b>	<b>Screening and observation interpolator.</b>	<b>27</b>
12.1	What does the screening do. . . . .	27
12.2	Basics about the architecture of the screening code. . . . .	27
12.3	More documentation about the screening. . . . .	27
<b>13</b>	<b>The initialisation.</b>	<b>28</b>
13.1	The different types of initialisation schemes. . . . .	28
13.2	Normal mode initialisation (NMI). . . . .	28
13.3	Digital filter initialisation (DFI). . . . .	28
<b>14</b>	<b>Algorithms of assimilation.</b>	<b>29</b>
14.1	Introduction: the different algorithms. . . . .	29
14.2	Optimal interpolation. . . . .	30
14.3	3D-VAR (three-dimensional variational) assimilation algorithm. . . . .	31
14.4	4D-VAR (quadri-dimensional variational) assimilation algorithm. . . . .	31
14.5	3D-VAR FGAT assimilation algorithm. . . . .	32
14.6	Sequences involved in a 4D-VAR assimilation. . . . .	33
14.7	ENVAR assimilation. . . . .	33
14.8	Documentation. . . . .	34

<b>15 Configuration 801.</b>	<b>35</b>
<b>16 Ensemble prediction and ensemble assimilation.</b>	<b>35</b>
16.1 Definition and purpose of ensemble prediction. . . . .	35
16.2 Ensemble assimilation. . . . .	36
16.3 Example: the operational ensemble prediction and assimilation at METEO-FRANCE (PEARP, AEARP, PEARO). . . . .	36
16.4 Documentation about ensemble prediction and assimilation. . . . .	36
<b>17 Singular vectors calculation and configuration 601.</b>	<b>37</b>
17.1 Introduction. . . . .	37
17.2 Configuration 601. . . . .	37
17.3 Documentation. . . . .	37
<b>18 Input surface climatology for models.</b>	<b>38</b>
18.1 Configuration 923. . . . .	38
18.2 Surface climatology for SURFEX. . . . .	39
<b>19 The other C9xx configurations.</b>	<b>39</b>
<b>20 Minimizations.</b>	<b>40</b>
<b>21 Observation treatment before assimilation.</b>	<b>42</b>
21.1 List of observations. . . . .	42
21.2 Observation treatment. . . . .	42
21.3 Documentation. . . . .	42
<b>22 IO and the different types of files used.</b>	<b>43</b>
<b>23 Distributed memory and OpenMP aspects.</b>	<b>44</b>
<b>24 Launching experiments.</b>	<b>45</b>
<b>25 References and documentation.</b>	<b>47</b>
25.1 Introduction. . . . .	47
25.2 Bibliography. . . . .	47
25.2.1 Publications relative to the physical parameterizations. . . . .	47
25.2.2 Other publications. . . . .	48
25.2.3 Theses (PHD). . . . .	50
25.2.4 ARPEGE notes “Notes de travail ARPEGE” 1 to 33. . . . .	50
25.2.5 Internal documentations describing operational configurations. . . . .	51
25.2.6 Other internal documentations. . . . .	52

# 1 Introduction and general purpose of this documentation.

The ARPEGE/IFS project has started in 1987 at METEO-FRANCE (for ARPEGE) and ECMWF (for IFS), and has lead to an operational use in 1992 at METEO-FRANCE and in 1994 at ECMWF. For the METEO-FRANCE side, the aim was to merge the old couple EMERAUDE (global spectral model) and PERIDOT (limited area grid-point model) into one global spectral model, introducing a variable-mesh geometry allowing to have the same resolution on France as for the old model PERIDOT (in the beginning of the 1990s this resolution was roughly 30 km). At METEO-FRANCE, the first operational use of ARPEGE with a variable-mesh geometry has been performed during summer 1993. During the 1990s, climate versions of ARPEGE/IFS have been developed on both sides.

In the beginning of the 1990s, a limited area spectral version of ARPEGE (called ALADIN) has been developed, and a first operational use has been done in 1994; this model is developed in collaboration with some other countries, principally some Eastern Europe and Mediterranean countries. An important part of the code is common to the ARPEGE/IFS and ALADIN software. The current operational versions of ALADIN have a mesh-size between 3 and 10 km.

In the early 2000's, it was decided to launch a new project in order to build a non-hydrostatic limited-area model. The name of this model is AROME, and the scientific choices have lead to use the ALADIN code for the adiabatic part of the model. This model contains a specific set of physical parameterizations coming from MESO-NH. This model has been put into operations in December 2008.

ARPEGE/ALADIN code has also included some HIRLAM features.

The aim of this documentation:

- tells about all topics one can encounter in these models, as brief as possible, and minimizing the number of formulae.
- for a topic, gives references to all the detailed documentations written about this topic.
- for a topic, gives references to all the configurations involved in this topic.
- for a configuration, gives some very basic information about the code architecture.

## 2 A brief description of ARPEGE, ALADIN, AROME.

\* **ARPEGE/IFS:** ARPEGE is the acronym of “Action de Recherche Petite Echelle Grande Echelle”. This model is used both at METEO-FRANCE and ECMWF; at ECMWF it is named IFS (Integrated Forecasting system). ARPEGE is a global spectral model, with a Gaussian grid for the grid-point calculations. The vertical discretisation is done according a following-terrain pressure hybrid coordinate. There is a possibility to have an horizontal variable-mesh: the change of horizontal representation is defined by a change of pole (one defines a high resolution pole which can be different from the northern pole), and a conformal transformation done according to (Schmidt, 1977). If one denotes by  $c$  the stretching factor of the conformal transformation, the mapping factor  $M$  writes  $M = 0.5(c + 1/c)\mu + 0.5(c - 1/c)$  where  $\mu$  is the sine of the apparent latitude on the transformed sphere.  $M$  varies between  $c$  at the high resolution pole and  $1/c$  at the low resolution pole. ARPEGE contains different models (a 3D primitive-equation model, a 3D non-hydrostatic model and a 2D shallow-water model), different assimilation schemes (optimal interpolation, 3D variational, 4D variational), an internal post-processing scheme (FULL-POS) and some other diagnostics, an initialisation scheme (digital filter initialisation). A tangent linear code and an adjoint code have been coded for most applications. ARPEGE/IFS can work with different physics packages; there is for example one physics package used at METEO-FRANCE and one other at ECMWF. ARPEGE can be used for climate applications and in this case it uses a slightly different physics package. ARPEGE is used operationally at METEO-FRANCE; IFS is used operationally at ECMWF.

\* **ALADIN:** ALADIN is the acronym of “Aire Limitée Adaptation Dynamique INitialisation”. ALADIN is a limited-area version of ARPEGE. It is still a spectral model; its horizontal domain covers only a limited area, so the fields are “bi-periodicised” to be made able to match with a spectral representation. The vertical coordinate is the same as the one of ARPEGE. ALADIN needs to be forced by a global model which has to provide lateral boundary conditions; the lateral forcing is done according to a Davies relaxation (Davies, 1976). ALADIN is generally used with a plane projection (one can currently use a conformal Lambert projection of a Mercator projection), but it was also originally planned to have a “toric” version of ALADIN (never completed nor validated). When using ALADIN with a domain which is not too big, the horizontal grid is a quasi-regular grid (the mapping factor  $M$  has only weak variations and remains close to 1). Some applications can require a big domain where  $M$  can become significantly greater than 1. ALADIN can work with most of the options of ARPEGE (but the shallow-water model and the 4D-VAR assimilation scheme are not coded in ALADIN). Most of the code is common to ALADIN and ARPEGE, but there are specific applications which require different parts of code. ALADIN is used operationally by Eastern Europe countries and Mediterranean countries.

\* **AROME:** AROME is the acronym of “Applications of Research to Operations at MESoscale”. AROME is a limited-area model designed for fine modeling (horizontal mesh-size around 1 to 2.5 km). AROME takes most of the ALADIN code concerning the adiabatic part of the code (in particular the non-hydrostatic code), but uses a specific physics package well adapted for small mesh-sizes around 1 km: this physics package is mainly an adaptation of the one which is currently used in the research non-hydrostatic model MESO-NH (used by the team CNRM/GMME for research applications). AROME is used operationally at METEO-FRANCE.

\* **ALARO:** ALARO is the acronym of a version of ALADIN used in some central and eastern European countries. This version of ALADIN uses an AROME-oriented meso-scale physics which is adapted to perform also on horizontal mesh-sizes of 3-10 km; for specific points the physics is different from the ARPEGE one and also from the AROME one.

\* **HARMONIE:** ALARO is the acronym of a version of LAM model used by HIRLAM partners.

\* **Coding norms:** The code now reaches roughly 1000000 lines (project ARP/IFS only) and more than 3000000 lines (all projects) and such a code needs severe rules of coding in order to maintain it understandable and manageable. Some code norms have been specified at the origin of the ARPEGE project (and naturally extended to ALADIN and AROME). The main norm is the called DOCTOR norm (in particular this norm provides constraints on the names of variables and also on a minimal amount of comments to provide in the code). More details about this DOCTOR norm can be found in (NTA04). Some standards have also emerged about prefixing the names of a significant subset of routines: for example “LA” for semi-Lagrangian scheme routines, “GP” or “CP” for grid-point calculations routines, “SP” for spectral calculations routines, “CA” for CANARI (optimal interpolation assimilation scheme) routines, “SU” for setup routines. The current code mainly follows the FORTRAN-90 norms, with some remaining FORTRAN-77 features; it is optimised to be run on various types of machines (vectorial or scalar distributed memory machines, clusters of workstations). The current coding rules can be found in documentation (IDCS11).

The code is currently managed on UNIX station under the software GIT.

\* **History of the code:** The history of the main implementations and the main technical changes can be found in documentation (IDARC).

### 3 The list of available configurations and some basics about the architecture of the code.

One can find more details in documentations (IDARC) and (TDECTEC), here is a brief sum-up.

#### 3.1 The list of available configurations.

The code has different configurations which can do for example hydrostatic forecasts, data assimilation, sensitivity studies; each configuration has a code between 1 and 999 (variable **NCONF** of **NAMARG**). The following ranges of configurations are:

- 0- 99 : 3-D integration job.
- 100-199 : variational job.
- 200-299 : 2-D integration job.
- 300-349 : KALMAN filter.
- 350-399 : predictability model (currently not used).
- 400-499 : test of the adjoint.
- 500-599 : test of the tangent linear model.
- 600-699 : eigenvalue/vector solvers.
- 700-799 : optimal interpolation.
- 800-899 : sensitivity experiments.
- 900-999 : miscellaneous other configurations.

Only a small subset of these numbers are used, the existing configurations are currently:

- 1: 3-D integration job (hydrostatic and non-hydrostatic).
- 131: incremental 4-D VAR/3-D VAR (for LAM models only the 3-D VAR is currently available).
- 201: 2D shallow-water model (unstretched untilted IFS only).
- 202: 2D vorticity equation model (unstretched untilted IFS only).
- 302: simplified extended Kalman filter (SEKF).
- 401: test of adjoint with 3-D hydrostatic primitive equation model.
- 421: test of adjoint with 2D shallow-water model (unstretched untilted IFS only).
- 422: test of adjoint with 2D vorticity equation model (unstretched untilted IFS only).
- 501: test of tangent linear with 3-D hydrostatic primitive equation model.
- 521: test of tangent linear with 2D shallow-water model (unstretched untilted IFS only).
- 522: test of tangent linear with 2D vorticity equation model (unstretched untilted IFS only).
- 601: eigenvalue/vector solver for 3-D hydrostatic primitive equation model.
- 701: optimal interpolation with CANARI. This optimal interpolation can be limited to surface variables.
- 801: sensitivity studies with 3-D hydrostatic primitive equation model.
- 901: sets up initial conditions (routine **CPREP1**), reads a MARS file and writes data on an ARPEGE file (ARPEGE only).
- 903: alternate way to do off-line FULL-POS.
- 911: computes dilatation matrices for spectral space (ARPEGE only); now externalised in UTILITIES.
- 923: initialisation of climatological files.
- 931: creation of an ARPEGE file containing the NESDIS SST (routine **CSSTBLD**) (ARPEGE only).
- 932: interpolates the sea-ice concentration field from satellite data to the ARPEGE grid (routine **CSEAICE**) (ARPEGE only).

Additional “apparent” configurations are used (the “apparent” number of the configuration is not the one put in variable **NCONF**):

- 2: 3-D integration job with screening (**NCONF**=1).

## 3.2 Basics about the architecture of the code.

The code is very complex, but most of the configurations (essentially those between 1 and 899) match the following scheme:

```
MASTER -> CNT0 ->
* SUOYOMA (setup of level 0, part A) ->
  - set-up before SUGOMETRY
  - SUGOMETRY
  - set-up after SUGOMETRY
* SUOYOMB (setup of level 0, part B).
* CNT1 for conf 1-99 or 200-299
* CVA1 for conf 100-199
* CSEKF1 for conf 301-349
* CAD1 for conf 401-499
* CTL1 for conf 501-599
* CUN1 for conf 601-699
* CAN1 for conf 701-799
* CGR1 for conf 801-899
* CPREP1 for conf 901
* CPREP3 for conf 903
* INCLIO for conf 923
* CSSTBLD for conf 931
* CSEAICE for conf 932
```

For example under **CNT1** one will find the following architecture:

```
CNT1 ->
* SU1YOM (setup of level 1)
* CNT2 ->
  - SU2YOM (setup of level 2)
  - CNT3 ->
    * SU3YOM (setup of level 3)
    * CSTA -> (reads the files containing the initial situation)
    * CNT4 ->
      - some setup routines of level 4.
      - STEPO
      * some other setup routines for particular applications.
  - some other setup routines for particular applications.
* some other setup routines for particular applications.
```

We can see that there are 5 levels of setup routines numbered from 0 to 4 and 5 levels of control routines **CNT0**, **CNT1**, **CNT2**, **CNT3**, **CNT4**.

Routine **STEPO** manages a timestep integration, the following steps are:

- Writing some results on files.
- Inverse transforms from spectral space to grid-point space.
- Grid-point calculations (for example all the explicit adiabatic part of the dynamics, and the physics).
- Direct transforms from grid-point space to spectral space.
- Spectral calculations (for example the inversion of Helmholtz equation in the semi-implicit scheme and the horizontal diffusion scheme).

Concerning the configurations which do not call **CNT1**:

- The organigramme under **CAD1**, **CTL1**, **CGR1**, **CSEKF1** is more complex because some adjoint or tangent linear code may be called.
- The organigramme under **CAN1** calls **SU1YOM**, **SU2YOM**, **SU3YOM**, some other setup routines to read the observations and the guess, then calls some specific CANARI routines to perform the optimal interpolation assimilation.
- The organigramme under **CPREP1**, **CPREP3**, **CSSTBLD**, **CSEAICE** is generally simple.
- The organigramme under **INCLIO** has a moderate complexity and generally uses routines specific to configuration 923.

## 4 The different projects.

The code has been progressively split in several “projects”, a significant subset of these projects being used for the pre-processing of observations which is required before the data-assimilation scheme. These projects and their dependencies have been described for example in documentations (IDFSTB) and (IDARC). These projects are managed under the library manager GIT which are implemented on the workstation “merou” at METEO-FRANCE. The users’ guide of GIT is provided in documentation (IGIT). For people interested in the initialisation+model part itself the main projects to know are (names are those given in library GIT):

- ALADIN: specific LAM routines (limited area models, in particular ALADIN and AROME).
- ARPIFS (called IFS at ECMWF): global model routines (ARPEGE, IFS), and routines common to global and LAM models. This is the core of the ARPEGE/IFS software.
- TRANS: spectral transforms for global models.
- ETRANS: spectral transforms for LAM models.
- ALGOR: linear algebra (including minimizers).
- IFSAux: some application routines, for example reading or writing on “LFI” or ARPEGE files, interface routines for distributed memory environment.
- SURF: ECMWF surface scheme.
- RADIATION: ECMWF new radiation scheme.
- BIPER: bi-periodicisation package (limited area models).
- COUPLING: lateral coupling and spectral nudging in LAM models.
- MPA: upper air MESO-NH/AROME type physics.
- MSE and SURFEX: SURFEX surface scheme; MSE is the interface for SURFEX.

The following ones contains some utilities for specific applications:

- UTILITIES: utilities package, containing various tools like the operational FA-to-GRIB converter (PROGRID), or programs to operate on ODB and radiances bias corrections.
- SCRIPTS: scripts used at ECMWF to launch different configurations of IFS; some of them are used in the ECMWF procedures “prep\_an” and “prep\_ifs”.
- MITRAILLE: MITRAILLETTE environnement (environnement to do simple code validations).

The other projects (see following list) are often devoted to observation pre-processing or climate simulations.

- AEOLUS: package for satellite lidar wind data pre-processing (aeolus).
- BLACKLIST: package for blacklisting.
- COPE: package for “cope” (new way to pre-process observations).
- OBSTAT: statistics of observation feedback data (only used at ECMWF).
- ODB: ODB (Observational DataBase software), needed by ARPEGE/ALADIN for their analysis or their assimilation cycle.
- SATRAD: satellite data handling package, needed by assimilation cycle.
- SCAT: scatterometers handling.

## 5 The sequences in a forecast suite.

### 5.1 ARPEGE forecast suite.

One describes the succession of jobs which are necessary in an operational suite: the example of ARPEGE-METROPOLE TL1198L105c2.2 will be taken (operational configuration at 01/07/2016). (denotation TL1198L105c2.2 means: TL1198 for linear Gaussian grid and triangular truncation 1198; L105 for 105 layers; c2.2 for stretching coefficient=2.2). The reader can for example read documentations (NTA30) and (IDTS) to have more information about the spectral representation on a field.

\* **Resolutions and denotations:** “ARP-METROPOLE HR” is TL1198L105c2.2 (high resolution for ARPEGE-METROPOLE); “ARP-METROPOLE LR1” is TL149L105c1 (low resolution for first minimisation in ARPEGE-METROPOLE); “ARP-METROPOLE LR2” is TL399L105c1 (low resolution for second minimisation in ARPEGE-METROPOLE); AROME-FRANCE resolution is 1.3 km, 90 levels.

#### \* Assimilation cycle:

```
-- PROVIDE AN ANALYSIS --
01/ Conf 1: 6 hour forecast (ARP-METROPOLE HR) [guess or forecast]
02/ Observation pre-processing [bator]
03/ Conf 1 with screening (ARP-METROPOLE HR) [screening]
04/ FULL-POS (ARP-METROPOLE HR) -> (ARP-METROPOLE LR1) on previ3 [update_1/fph2l]
05/ Conf 131 (adiabatic) at (ARP-METROPOLE LR1) [update_1/minim]
06/ FULL-POS (ARP-METROPOLE LR1) -> (ARP-METROPOLE HR) on the increments [update_1/fpl2hg and fpl2ha]
07/ Conf 1 with updating the trajectory by adding the increments
and doing a quality control (ARP-METROPOLE HR) [update_1/traj]
08/ FULL-POS (ARP-METROPOLE HR) -> (ARP-METROPOLE LR2) on previ3 [update_2/fph2l]
09/ Conf 131 (with simplified physics) at (ARP-METROPOLE LR2) [update_2/minim]
10/ FULL-POS (ARP-METROPOLE LR2) -> (ARP-METROPOLE HR) on the increments [update_2/fpl2hg and fpl2ha]
11/ Conf 1 with updating the trajectory by adding the increments
and doing a quality control (ARP-METROPOLE HR) [update_2/traj]
12/ Conf 931 and 932 doing ice and SST files daily [surfan/c931 and surfan/c932]
13/ Observation pre-processing for surface [batorsurf]
14/ Conf 701 (surface analysis) at resolution (ARP-METROPOLE HR) [surfan/canari]
Return to step 01/ for the following 6 hours.
```

#### \* Forecast cycle:

```
-- PROVIDE AN ANALYSIS --
01b/ Conf 1: 6 hour forecast (ARP-METROPOLE HR) [guess or forecast]
02b/ Observation pre-processing [bator]
03b/ Conf 1 with screening (ARP-METROPOLE HR) [screening]
04b/ FULL-POS (ARP-METROPOLE HR) -> (ARP-METROPOLE LR1) on previ3 [update_1/fph2l]
05b/ Conf 131 (adiabatic) at (ARP-METROPOLE LR1) [update_1/minim]
06b/ FULL-POS (ARP-METROPOLE LR1) -> (ARP-METROPOLE HR) on the increments [update_1/fpl2hg and fpl2ha]
07b/ Conf 1 with updating the trajectory by adding the increments
and doing a quality control (ARP-METROPOLE HR) [update_1/traj]
08b/ FULL-POS (ARP-METROPOLE HR) -> (ARP-METROPOLE LR2) on previ3 [update_2/fph2l]
09b/ Conf 131 (with simplified physics) at (ARP-METROPOLE LR2) [update_2/minim]
10b/ FULL-POS (ARP-METROPOLE LR2) -> (ARP-METROPOLE HR) on the increments [update_2/fpl2hg and fpl2ha]
11b/ Conf 1 with updating the trajectory by adding the increments
and doing a quality control (ARP-METROPOLE HR) [update_2/traj]
12b/ Conf 931 and 932 doing ice and SST files daily [surfan/c931 and surfan/c932]
13b/ Observation pre-processing for surface [batorsurf]
14b/ Conf 701 (surface analysis) at resolution (ARP-METROPOLE HR) [surfan/canari]
-- PROVIDE A FORECAST --
15b/ Conf 1: 102 hour forecast (ARP-METROPOLE HR) [forecast]
16b/ FULL-POS on LALON domains from ARPEGE outputs [fullpos]
Return to step 01b/ for the following 6 hours.
When the steps (15b/ to 16b/) are not done, the sequence (01b/ to 14b/) is replaced by (01/ to 14/).
```

01b/ can differ from 01/ because the analysis used in 01b/ may have been done with less observations (short cut-off) than the analysis used in 01a/. (02b/ to 14b/) differ from (02/ to 14/) because the cut-off is shorter (use of less observations). The reason is that the analysis provided after step 14b/ must be available before the one provided after step 14/. The frequency of steps (01/ to 14/) is 6 hours, the one of steps (01b/ to 14b/) is also 6 hours, but the steps (15b/ to 16b/) can be done only every 12h.

\* **Comments:** One can see that at least the following configurations of ARPEGE are currently used in the operational suite:

- **NCONF=1:** 3-D integration job.

- **NCONF=131**: incremental 4-D VAR.
- **NCONF=701**: optimal interpolation with CANARI for surface fields.
- **NCONF=1**: 3-D integration job with screening.
- **NCONF=1**: 3-D integration job with trajectory updating and screening.
- **NCONF=1**: FULL-POS used for a change of horizontal and vertical geometry.
- **NCONF=1**: FULL-POS used to provide outputs on “LALON” grids.

## 5.2 AROME forecast suite.

One describes the succession of jobs which are necessary in an operational suite: the example of AROME-FRANCE will be taken (operational configuration at 01/07/2016).

### \* Forecast cycle:

```
-- PROVIDE AN ANALYSIS --
01/ FULL-POS (ARP-METROPOLE HR) -> AROME-FRANCE resolution [cpinit + coupling + couplingsurf]
02/ Conf E001: 1 hour forecast using LBC from 01/ (AROME-FRANCE resolution) [forecast + fullpos]
03/ Observation pre-processing [obsextract, batodb, merge_varbc]
04/ FULL-POS (ARP-METROPOLE HR) -> AROME-FRANCE resolution [coupling_fc];
    add some surface fields needed in var setup in the background.
05/ Conf E001 with screening (AROME-FRANCE resolution) [screening]
06/ Conf E131 (3DVAR minimization, AROME-FRANCE resolution) [minim]
-- PROVIDE A FORECAST --
07/ Conf E001: forecast using LBC from 04/ (AROME-FRANCE resolution) [guess or forecast]
    + in-line FULL-POS on LALON domains from AROME outputs.
```

\* **Comments:** One can see that at least the following configurations are currently used in the operational suite:

- **NCONF=E001**: 3-D integration job.
- **NCONF=E131**: 3-D VAR.
- **NCONF=E001**: 3-D integration job with screening.
- **NCONF=E001**: FULL-POS used for a change of horizontal and vertical geometry.
- **NCONF=E001**: FULL-POS used to provide outputs on “LALON” grids.

## 6 Basics about geometry.

### 6.1 Global model ARPEGE/IFS.

\* **Spectral data:** They are organised according to a triangular truncation: there are spectral coefficients for each couple of wavenumbers  $(m, n)$  ( $m$  is the zonal wavenumber,  $n$  is the total wavenumber), where  $n$  varies between 0 and the model truncation  $N_s$  (in variable **NSMAX**), and  $m$  varies between 0 and  $n$ .

\* **Grid-point data:** Grid-point data are organised on a Gaussian grid. There are **NDLON** longitudes and **NDGLG** latitudes. A 2D field has a total amount of **NGPTOTG** grid-points.

For a constant mesh geometry, with no pole tilting:

- latitudes are nearly equidistant.
- the total number of latitudes is assumed to be an even number, so there is no latitude at the equator.
- there is no point at the pole.
- on each latitude, points are equidistant.
- the Gaussian grid can be unreduced, and in this case each latitude has the same number of points.
- the Gaussian grid can be reduced (option used by default in the operational models), and in this case the number of points on each latitude roughly varies with the cosine of the latitude; use of reduced Gaussian grid allows to save 30 per cent of the grid points.
- alternate (cheaper) reduced Gaussian grids are currently investigated: for example the octahedric grid.

Geometry can be tilted by defining an apparent pole different from the true Northern pole. The apparent pole has a longitude  $\lambda_{cen}$  (variable **RLOCEN** in **YOMGEM**) and a latitude  $\theta_{cen}$  (variable **RMUCEN** in **YOMGEM** contains  $\sin(\theta_{cen})$ ).

Geometry can be stretched, a conformal transformation is done according to (Schmidt, 1977). If one denotes by  $c$  the stretching factor of the conformal transformation (variable **RSTRET** in **YOMGEM**), the mapping factor  $M$  writes  $M = 0.5(c + 1/c)\mu + 0.5(c - 1/c)$  where  $\mu$  is the sine of the apparent latitude on the transformed sphere.  $M$  varies between  $c$  at the high resolution pole and  $1/c$  at the low resolution pole. Stretching can be combined with pole tilting.

Properties of the Gaussian grid listed above remain valid when there is stretching or pole tilting.

### 6.2 Limited area model ALADIN/AROME.

\* **Spectral data:** They are organised according to an elliptic truncation: there are spectral coefficients for each couple of wavenumbers  $(m, n)$  ( $m$  is the zonal wavenumber,  $n$  is the meridian wavenumber), where  $n$  varies between 0 and the model meridian truncation  $N_s$  (in variable **NSMAX**), and  $m$  varies between 0 and the model zonal truncation  $N_{ms}$  (in variable **NMSMAX**). Additionally,  $(n/N_s)^2 + (m/N_{ms})^2$  remains bounded by 1.

\* **Grid-point data:** Grid-point data are organised on a rectangular grid (containing **NDLON** longitudes and **NDGLG** latitudes) which is on a limited area plane projection. A bi-periodicity constraint is maintained on this grid. This rectangular grid is subdivided in three sub-domains C, I and E.

- C is the central or inner zone: this is the sub-domain where dynamics and physics can evolve freely, without be touched directly by the lateral forcing.
- E is the extension zone, where data completely come from the lateral forcing.
- I is the intermediate zone, where data is a combination of inner data and lateral forcing data.

This rectangular grid is close to a constant mesh grid: the mapping factor actually slightly varies, and generally remains close to 1 (and always  $\geq 1$ ) if the limited area domain is not too large. The maximum value of the mapping factor is in variable **RSTRET** of **YOMGEM**.

There are currently 4 types of plane projections:

- polar stereographic projection.
- conformal conical Lambert projection.
- Mercator projection.
- Mercator rotated tilted projection.

The three first ones are defined as projections from an untilted unstretched global geometry. The Mercator rotated tilted projection is defined as a projection from a tilted unstretched global geometry. The main variables to know in order to define the kind of plane projection are:

- **LMRT** (in **YEMGEO**): T for Mercator rotated tilted projection, F otherwise.
- **ERPK** (in **YEMGEO**): projection parameter; 1 for polar stereographic projection, between 0 and 1 (excluded) for conformal conical Lambert projection, 0 for Mercator projection.

More information about limited area projections can be found in documentations (**IDMOAL**), (**IDPROJ**) and (**IDPRLAM**).

## 7 The forecast: configurations 1, 201, 202.

\* **The different types of equations:** The equations governing the atmosphere can be represented with more or less approximations.

- The non-hydrostatic system of equations can represent the non-hydrostatic effects; there are several ways to write a non-hydrostatic system of equations, some ways filter the acoustic waves, some other ways keep them. The non-hydrostatic system of equations which is currently coded is the fully elastic set of equations, and this system does not filter acoustic waves. It is interesting to use a non-hydrostatic system of equations for a model, when the horizontal mesh-size is under 5 km.
- For coarser resolutions (above 5 km), it is useless to represent non-hydrostatic effects and one uses a primitive equations (hydrostatic) system of equations.
- Deep vs shallow atmosphere formulations:
  - The deep atmosphere system of equations retains the  $w$ -terms in the Coriolis term and takes account of the true distance from the current point to the Earth centre; additionally, the vertical variations of the gravitation acceleration  $\mathbf{g}$  can be taken into account.
  - The shallow (or thin layer) atmosphere system of equations neglects the  $w$ -terms in the Coriolis term and replaces the true distance from the current point to the Earth centre by a constant. The classical system of primitive equations is a shallow atmosphere system where the gravitation acceleration is assumed to be constant.
- Another way of refinement is to represent the latitudinal variations of the Earth radius, or the latitudinal variations of the gravitation acceleration  $\mathbf{g}$ .
- A deeper stage of approximations is to consider models with only one vertical layer: for example the shallow water system of equations or the “vorticity” system of equations. These systems of equations are used for simulations on an idealised environment.

A detailed review of the different systems of equations and of the approximations with their range of validity can be found in (White, 2000; White et al., 2005).

Configuration 1 runs either the non-hydrostatic system of equations or the primitive equations system. Configuration 201 runs the shallow water system of equations. Configuration 202 runs the vorticity system of equations.

\* **Representation of equations:** Each equation of a system of equations can generally write:

$$\frac{dX}{dt} = \mathcal{A} + \mathcal{F}$$

where  $X$  is a prognostic variable, the evolution of which one wants to know.  $\mathcal{A}$  represents all the effects which can be explicitly represented for the current resolution (often called “adiabatic effects”). That can be for example:

- The Coriolis force (momentum equation).
- The pressure-gradient force term (momentum equation).
- The conversion term (temperature equation).
- The divergence term (continuity equation).

$\mathcal{F}$  represents all the sub-scale effects (often called “diabatic effects”). That can be for example:

- radiation.
- stratiform precipitations.
- convection, and convective precipitations.
- vertical diffusion.
- microphysics.
- orographic gravity wave drag.
- exchanges with the surface, interaction with the surface vegetation.

When writing

$$\frac{dX}{dt} = \mathcal{A} + \mathcal{F}$$

we consider the total temporal derivative of  $X$  or Lagrangian derivative of  $X$ . Schemes discretizing such an equation are called semi-Lagrangian schemes. The semi-Lagrangian scheme is activated by setting variable **LSLAG=.T.** in namelist **NAMARG.**

This equation can also be rewritten (for example for a vertical coordinate which is denoted by  $\eta$ ):

$$\frac{\partial X}{\partial t} = -\mathbf{V}\nabla X - \dot{\eta}\frac{\partial X}{\partial \eta} + \mathcal{A} + \mathcal{F}$$

where  $\mathbf{V}$  is the horizontal wind,  $\nabla$  is the horizontal gradient operator (along  $\eta$  surfaces).  $-\mathbf{V}\nabla X$  is the horizontal advection term,  $-\dot{\eta}\frac{\partial X}{\partial \eta}$  is the vertical advection term. Schemes discretizing such an equation are called Eulerian schemes. The Eulerian scheme is activated by setting variable **LSLAG=.F.** in namelist **NAMARG.**

In the Eulerian schemes and the “three-time level” semi-Lagrangian schemes, one has to compute  $X(t + \Delta t)$  knowing  $X(t - \Delta t)$ : such schemes are called leap-frog schemes. In the “two-time level” semi-Lagrangian scheme, one has to compute  $X(t + \Delta t)$  knowing  $X(t)$ . The two-time level semi-Lagrangian scheme is activated by setting variable **LTWOTL=.T.** in namelist **NAMCT0.**

The semi-Lagrangian schemes are now currently used in operational models and they allow timesteps significantly larger (several times) than those allowed by Eulerian schemes. For more details about equations and their discretization the reader can refer to documentation (IDEUL), and for details about semi-Lagrangian schemes the reader can refer to documentation (IDSL); in the following one will only list some systems of equations in Lagrangian form.

The discretization of equations needs to separate the linear terms (generating fast waves like gravity waves) and the non-linear terms. The linear terms are discretized in a semi-implicit manner with one part of the computations done in grid-point space, and one other part in spectral space. See documentation (IDSI) for more details about semi-implicit schemes.

Some models (especially some options of the non-hydrostatic model) can require to use “iterative-centred implicit” schemes (sometimes called “predictor-corrector” schemes): the same equation is integrated several times using different iterations of  $X(t + \Delta t)$ ; for more details about the “iterative-centred implicit” schemes, see documentations (IDSI), (IDPC), (IDPCEU) and (IDPCSL).

#### \* Some notations:

- $\mathbf{V}$  is the horizontal wind. Its zonal component (on the Gaussian grid) is denoted by  $U$ . Its meridional component (on the Gaussian grid) is denoted by  $V$ .
- $D$  is the horizontal wind divergence.
- $\zeta$  is the horizontal wind vorticity.
- $T$  is the temperature.
- $q$  is the humidity.
- $\Pi$  is the hydrostatic pressure.
- $\Pi_s$  is the hydrostatic surface pressure.
- $\Omega$  is the Earth rotation angular velocity.
- $(\lambda, \theta)$  are the geographical longitude-latitude coordinates.
- $\omega = \frac{d\Pi}{dt}$  is the total temporal derivative of the hydrostatic pressure.
- $p$  is the pressure,  $p_s$  is the surface pressure.
- $gz$  is the geopotential height.
- $\Phi$  is the total geopotential (in some cases it can be different from  $gz$ ).
- $\Phi_s = gz_s$  is the surface geopotential (i.e. the orography).
- $g$  is the gravity acceleration constant.
- $R$  is the gas constant for air,  $R_d$  the gas constant for dry air and  $R_v$  the gas constant for water vapour.
- $c_p$  is the specific heat at constant pressure for air and  $c_{p,d}$  is the specific heat at constant pressure for dry air.
- $c_v$  is the specific heat at constant volume for air and  $c_{v,d}$  is the specific heat at constant volume for dry air.
- $\nabla$  is the first order horizontal gradient on  $\eta$ -surfaces.
- $\alpha_T$  is a vertical-dependent coefficient used to define a thermodynamic variable  $T + \delta_{TR} \frac{\alpha_T \Phi_s}{R_d T_{st}}$  less sensitive to orography than temperature  $T$ . Subscript “st” stands for “standard atmosphere”.
- $\rho$  is the mass per volume unit of air.

\* **The 3D primitive equation system:** This system is written here for the classical approximation (no  $w$  Coriolis term, constant gravitation acceleration  $g$ ). Prognostic variables are the wind components of the horizontal wind  $\mathbf{V}$ , temperature  $T$ , specific humidity  $q$  and the logarithm of hydrostatic surface pressure  $\log \Pi_s$ . Momentum equation: Coriolis force can be treated explicitly ( $\delta_{\mathbf{v}}=0$ ) or implicitly ( $\delta_{\mathbf{v}}=1$ ) in the Lagrangian equation.

$$\frac{d(\mathbf{V} + \delta_{\mathbf{v}}(2\boldsymbol{\Omega} \wedge \mathbf{r}))}{dt} = -2(1 - \delta_{\mathbf{v}})(\boldsymbol{\Omega} \wedge \mathbf{V}) - \nabla\Phi - RT\nabla(\log \Pi) + \mathbf{F}_{\mathbf{v}} \quad (1)$$

$\mathbf{F}_{\mathbf{v}}$  is the physical contribution on horizontal wind.

Temperature equation:

$$\frac{d\left(T + \delta_{\text{TR}} \frac{\alpha_{\text{T}} \Phi_s}{R_d T_{\text{st}}}\right)}{dt} = \frac{d\left(\delta_{\text{TR}} \frac{\alpha_{\text{T}} \Phi_s}{R_d T_{\text{st}}}\right)}{dt} + \frac{RT}{c_p \Pi} \omega + F_{\text{T}} \quad (2)$$

$F_{\text{T}}$  is the physical contribution on temperature.  $\delta_{\text{TR}} = 1$  is generally used in semi-Lagrangian models.

Specific humidity equation:

$$\frac{dq}{dt} = F_{\text{q}} \quad (3)$$

$F_{\text{q}}$  is the physical contribution on moisture.

Continuity equation:

$$\frac{d\left[\log \Pi_s + \delta_{\text{TR}} \frac{\Phi_s}{R_d T_{\text{st}}}\right]}{dt} = -\frac{1}{\Pi_s} \int_{\eta=0}^{\eta=1} \nabla \cdot \left(\mathbf{v} \frac{\partial \Pi}{\partial \eta}\right) d\eta + \mathbf{V} \nabla \cdot \left(\log \Pi_s + \delta_{\text{TR}} \frac{\Phi_s}{R_d T_{\text{st}}}\right) - \frac{1}{\Pi_s} \left[\dot{\eta} \frac{\partial \Pi}{\partial \eta}\right]_{\eta=1} + \frac{1}{\Pi_s} \left[\dot{\eta} \frac{\partial \Pi}{\partial \eta}\right]_{\eta=0} \quad (4)$$

Diabatic terms which can be non-zero in some options have been omitted. This shape of continuity equation is allowed by the use of a vertical terrain-following hybrid coordinate.

Additional prognostic variables like ozone, liquid water, ice, cloud fraction, can be added to this list, their equation looks like the one of specific humidity.

\* **The fully elastic non-hydrostatic system of equations (NHEE):** The right-hand sides of momentum equation and temperature equation are slightly modified. There are two additional equations, one for a variable which represents the difference between the total pressure  $p$  and the hydrostatic pressure  $\Pi$ , and the other one for a variable linked with the vertical divergence  $d$  or the vertical velocity  $w = \frac{dz}{dt}$ . The NH model currently coded (denoted NH-PDVD or NHEE model) uses  $d$  or  $d_4 = d + \mathbf{X}$ , and  $\log(p/\Pi)$  as NH prognostic variables. For the vertical divergence equation it is desirable to run with  $d_4 = d + \mathbf{X}$  (see IDEUL) for expression of  $\mathbf{X}$ ) to ensure stability. This is the one used in AROME.

The reader will find more information about the formulation of the non-hydrostatic equations in ALADIN in some documentations like (IDNHPB), (IDSL), (IDVNH1), (IDVNH2) and (IDVNH3). Variable controlling the choice of a non-hydrostatic model is **LNHDYN** in **NAMCT0**.

\* **The quasi elastic non-hydrostatic system of equations (NHQE):** This alternate version filters acoustic waves and uses a diagnostic relationship for pressure departure variable. Evaluation of such a system (not yet coded) is planned soon.

\* **The shallow-water system of equations:** The model is assumed to have only one vertical layer of depth  $\Phi$  (equivalent height). Prognostic variables are the wind components of the horizontal wind  $\mathbf{V}$ , and the equivalent height  $\Phi$ . The ‘‘diabatic’’ terms are replaced by zero.

Momentum equation: Coriolis force can be treated explicitly ( $\delta_{\mathbf{v}}=0$ ) or implicitly ( $\delta_{\mathbf{v}}=1$ ) in the Lagrangian equation.

$$\frac{d(\mathbf{V} + \delta_{\mathbf{v}}(2\boldsymbol{\Omega} \wedge \mathbf{r}))}{dt} = -2(1 - \delta_{\mathbf{v}})(\boldsymbol{\Omega} \wedge \mathbf{V}) - \nabla\Phi \quad (5)$$

Continuity equation:

$$\frac{d(\Phi - (1 - \delta_{\text{TR}})\Phi_s)}{dt} = -(\Phi - \Phi_s)D + \delta_{\text{TR}} \mathbf{V} \nabla \cdot (\Phi_s) \quad (6)$$

\* **The 2D ‘‘vorticity’’ system of equations:** The system of equations starts from the shallow-water one, but  $\Phi$  and  $\Phi_s$  have to be replaced by zero. So continuity equation becomes  $\Phi = 0$ .

\* **Architecture of the code for configurations 1, 201 and 202.** A basic description has been given in part 3.2. Let us detail what is done under routine **STEPO**. The discretisation of equations leads to integrate the model with a timestep  $\Delta t$ ; each call of **STEPO** matches with one timestep integration.

Routine **STEPO** manages a timestep integration, the following steps are:

- Writing some results on files if required, for example writing the forecast on historical files (routine **IOPACK**). Written variables are generally spectral ones (this is the case for all variables with a spectral representation); some variables remaining in grid-point space are written in grid-point space (that includes in particular all surface variables used in the physics).
- Inverse transforms from spectral space to grid-point space (sequence of routines **TRANSINVH** – > **TRANSINV\_MDL**): applied to all prognostic variables; at this moment are also computed some horizontal derivatives which are used in the right-hand side of the equations.
- Grid-point calculations (code under **SCAN2M**); they include:
  - the explicit adiabatic part of the dynamics (for example in a primitive equations system: the Coriolis term, the pressure-gradient term, the conversion term in  $T$  temperature, and the divergence term in continuity equation).
  - the physics; the physical contributions are computed in a physical package of parameterizations.
- Lateral coupling (code under **ECOUPL1**; for limited area models only).
- Direct transforms from grid-point space to spectral space applied to the provisional ' $t + \Delta t$ ' variables (sequence of routines **TRANSDIRH** – > **TRANSDIR\_MDL**).
- Spectral calculations (routine **(E)SPCM**); they include:
  - the inversion of Helmholtz equation of the implicit linear problem (semi-implicit scheme).
  - spectral nudging for LAM models.
  - the horizontal diffusion scheme.

The reader would find more details about the algorithms involved in these different steps in more specialized documentations. For example:

- Spectral transforms are documented in documentations (IDTS) and (IDTSN) for ARPEGE, (IDTSAL) for ALADIN. In ARPEGE/IFS spectral transforms are a combination of a Fourier transform and a Legendre transform. In ALADIN and AROME spectral transforms are done by a double Fourier transform.
- The semi-Lagrangian scheme is documented in documentation (IDSL).
- The semi-implicit scheme and the iterative-centred implicit scheme are documented in documentation (IDSI).
- The horizontal diffusion scheme is documented in documentation (IDDH).

In the code 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 controls the grid-point coupling in LAM models.
- L8 controls the direct transforms.
- L9 controls the spectral computations.

For example a model integration time-step is defined by the sequence [L1]AAA0AAA, with [L1]=0 if there is no writing of any historical or diagnostic file and [L1]='A' if an historical file is written.

\* **Restart.** There is a possibility (at least in the hydrostatic primitive equation model) to save the current state of the model (spectral and grid-point data) on files, and to re-launch the model directly from these files. This option, which can be useful for the operational suite, is called “restart”, and can be activated by variables **NFRRES** and **NRESTS** in namelist **NAMRES**. It has been briefly documented in the past (in 1992) in the small documentation (IDREST), but no recent update has been provided.

## 8 Basics about the physics.

\* **Introduction:** As we have seen before, it is necessary to treat some sub-scale meteorological phenomena (like convection, radiation) in a special package called “physical parameterizations”. There are currently several physical parameterizations packages in the environment ARPEGE/IFS/ALADIN. That concerns the primitive equation model and the non-hydrostatic model.

- one for complete physics used at ECMWF.
- one for complete physics used at METEO-FRANCE for operational applications (ARPEGE and ALADIN).
- one for complete physics used at METEO-FRANCE for climatic applications (ARPEGE-CLIMAT).
- one for simplified physics used at ECMWF.
- one for simplified physics used at METEO-FRANCE (currently used in the last minimization of the 4D-VAR algorithm).
- one for a very simplified physics (Buizza physics).
- one for the MESO-NH/AROME physics used at METEO-FRANCE for meso-scale applications.
- one for the “ALARO” physics.
- one for the HIRLAM physics.

The physics package does not necessary give the tendencies for exactly the same prognostic variables as for the adiabatic part of the model. For example the adiabatic part of the code uses the temperature  $T$  as prognostic variable, but the physics uses the total energy  $h = c_p T + \Phi + Lq$ , so a change of variable is necessary to retrieve the physical tendency of temperature.

There are two possibilities in the discretization to evaluate the physical terms; either at the origin instant ( $t - \Delta t$  for a leap-frog scheme,  $t$  for a two-time level semi-Lagrangian scheme) or at the final instant ( $t + \Delta t$ ); a combination of both is possible; it is not possible to evaluate the physics at the mid-instant of the integration timestep ( $t$  for for a leap-frog scheme for example) because physics contains highly non-linear terms which make this option numerically unstable. At METEO-FRANCE the choice which has been done is to compute physics at the origin instant (unlagged physics). At ECMWF physics is computed at the final instant (lagged physics) and there is an option to split the physics between the origin and the final instants. Variable controlling the choice of lagged/unlagged physics is **LAGPHY** in **NAMPHY**. **LAGPHY=.T.** means “lagged physics”.

Each physical contribution can be called in a sequential or a parallel scheme. The purely parallel approach means that all the physical parameterizations use the same entry (the same origin instant or the same final instant variables). All physical tendencies can be computed separately, and at the end the total tendency is simply the sum of all tendencies. The purely sequential approach means that each physical parameterization updates the input dynamical variables; the following parameterization called uses an updated state of input variables; the result depends on the order of calling the parameterizations; this approach has a sense for a lagged package of physics. Combining the two approaches into a “mixed” approach is possible; in this case the physical parameterizations package can be divided into several subsets; the treatment is purely parallel in each subset, but the total contribution of each subset updates the input variable. The purely parallel approach is used at METEO-FRANCE. The “mixed” approach is used at ECMWF.

\* **Physical package used at ECMWF (and its simplified version for variational applications):** This physical package is well described in the ECMWF documentation (TDECPHY) and in particular a sum-up is given in part 1 of this documentation, so it is not necessary to give more details here. The entry piece of code is **EC\_PHYS\_DRV** – > **EC\_PHYS**, which calls **CALLPAR** for the complete physical package. This physics package is significantly different from the one used at METEO-FRANCE and significantly more expensive also. The activation of ECMWF physics is done by setting **LEPHYS=.T.** in **NAMPHY**.

\* **Complete physical package used at METEO-FRANCE for operational applications.** The entry routine in the code is **MF\_PHYS**, this routine calls **APLPAR** for the complete physical package. The name of routines performing the physics generally begin by “**AC**”. The activation of METEO-FRANCE physics is done by setting **LMPHYS=.T.** in **NAMPHY**. The six following phenomena are described (the brief description of each parameterization is written after a sum-up provided by Jean-Marcel Piriou, and updated by Y. Bouteloup and F. Bouyssel):

- Radiation: The ECMWF operational radiation scheme is used. The RRTM scheme (Rapid Radiative Transfer Model, Mlawer et al., 1997) represents the longwave radiative exchanges. The objective in the development of RRTM has been to obtain an accuracy in the calculation of fluxes and heating rates

consistent with the best line-by-line models. It uses the correlated-k method and shows its filiation to the Atmospheric and Environmental Research, Inc. (AER) line-by-line model (LBLRTM) through its use of absorption coefficients for the relevant k-distributions derived from LBLRTM. An improved version of the Fouquart's scheme represents the shortwave radiative exchanges. This shortwave scheme uses 6 spectral bands. The cloud optical properties are the same as in operational IFS model.

- Grid-scale (resolved) clouds and large-scale (stratiform) precipitations: the large-scale prognostic cloud and precipitation scheme developed by Philippe Lopez (2002) is implemented. The operational version uses four new prognostic variables for hydrometeors (cloud liquid water, cloud solid water, rain and snow). This scheme is based on a statistical cloud scheme (Smith, 1990) which represents cloud condensation and evaporation.

Separation of cloud liquid water and cloud solid water is function of the temperature. Large-scale cloudiness and cloud water contents are combined with the convective ones to provide total cloudiness and total cloud water contents to the radiation scheme. Microphysical processes explicitly represented in the scheme are autoconversion of cloud condensate into precipitation (Kessler type), collection of cloud condensate by precipitation, precipitation evaporation and melting.

- Sub-grid scale deep convection and convective precipitations: the original scheme is the one described in (Bougeault, 1985): mass-flux scheme, Kuo-type closure and triggering, sensible heat and water vapour budgets taking into account the contribution of turbulent fluxes in order to avoid an automatic CISK (convective instability of second kind) behaviour of the "original" Kuo approach in nearly balanced conditions. A certain number of modifications and/or options have been operationally introduced along the years (not extensively described).
- Turbulence scheme (boundary layer) and surface fluxes: A prognostic TKE (turbulent kinetic energy) scheme is used (Cuxart et al., 2000), associated with a mass flux scheme for shallow convection (Bechtold et al., 2001).

ECUME is used for surface fluxes above oceans.

- Gravity wave drag: four levels of increasing sophistication:
  - basic linear model (Boer and al., 1984).
  - non linearities (Mlynarz, 1990) like trapping, resonant damping (Peltier and Clark, 1987).
  - surface anisotropy effects (Baines and Palmer, ECMWF, following Phillips, 1984).
  - blocking.
- Soil processes, exchanges with the surface (fluxes at the surface, vegetation, run-off): this is based on the ISBA scheme described by (Noilhan and Planton, 1989) and by (Giard and Bazile, 2000). Some modifications have been added to the scheme for taking into account the freezing-melting effects of the soil water at different levels.
- There are two layers for the surface: a surface reservoir and a deep reservoir.

The reader can find a more detailed description of these parameterizations in (IDPHYE).

**\* Complete physical package used at METEO-FRANCE for climatic applications.**

The entry routine in the code is `MF_PHYS`, this routine calls `APLPAR` for the complete physical package. The name of routines performing the physics generally begins by "AC". There are noticeable differences and some completely different schemes can be used, for example for turbulence. More details can be found in documentation (IDARPC6).

**\* Simplified physical package used at METEO-FRANCE for operational applications.**

The entry routine in the code is `MF_PHYS`, this routine calls `APLPARS` for the simplified physical package. The name of routines performing the physics generally begin by "AC". Since the complete physical parameterizations provides a code which is often difficult to linearize, some simplifications have been brought to provide a simplified version of the physical parameterizations (the adjoint and tangent linear codes are easier to provide on this type of simplified parameterizations). The simplified physical package is described in part 2 of (PHDPHYSSIM), and in particular a summary of the simplifications brought are given in part 2.8 of (Janisková, 1998). The simplified physical package is used in all applications where a tangent linear or an adjoint code is required; this is the case in the 4D-VAR assimilation. The activation of METEO-FRANCE simplified physics is done by setting `LMPHYS=.T.` in `NAMPHY` and `LSIMPH=.T.` in `NAMSIMPHL`.

The simplified physical package has not been completely updated relatively to the complete physical package (radiation, stratiform precipitations), and currently the operational METEO-FRANCE suite uses only a subset of these parameterizations (vertical diffusion, gravity wave drag).

\* **Very simplified “Buizza” physical package.** The entry routine in the code is `CPG_DYN`, this routine calls `VDIFLCZ` for the simplified physical package (the call to `VDIFLCZ` could be moved in `MF_PHYS` and `EC_PHYS` in the future). This simplified physics only represents the vertical diffusion and the surface drag; it is applied to the wind components and optionally to the dry static energy. The activation of the very simplified “Buizza” physical package is done by setting `LSPHLC=.T.` in `NAPHLC`.

\* **Physical package for AROME:** it is an adaptation of the physical package currently used in MESO-NH (research non-hydrostatic model used jointly by the “Laboratoire d’Aérodynamique” (Université Paul Sabatier, Toulouse) and the team CNRM/GMME). The entry routine in the code is `MF_PHYS`, this routine calls `APL_AROME` for the complete physical package. The activation of AROME physics is done by setting `LMSE=.T.` in `NAMARPHY` for surface processes and `LMPA=.T.` in `NAMARPHY` for upper air processes. The seven following phenomena are described:

- Radiation (same as ARPEGE).
- Turbulence and Shallow convection. As in ARPEGE a prognostic TKE scheme is used (Cuxart et al., 2000) associated with a mass flux scheme with a CAPE type closure for shallow convection but the mass flux scheme comes from the work of (Pergaud et al., 2009) and contains a parametrisation of dry thermals.
- Microphysics (warm clouds, atmospheric ice, sub-grid condensation). The scheme is described in (Pinty and Jabouille, 1998) and (Caniaux et al., 1994). Five classes of hydrometeors are considered (cloud liquid water and cloud ice, rain, snow and graupels). Sedimentation of cloud ice and cloud liquid water is also activated to obtain a better simulation of cirrus and fog. A statistical cloud scheme based on a gaussian PDF (Bechtold et al. 1992) is used.
- Surface processes (SURFEX package, see below).
- Additional remarks:
  - Compared to the complete large-scale METEO-FRANCE physics, there is no distinction between stratiform and convective precipitations and there is no separate parameterization of gravity wave drag.

And additionally the atmospheric chemistry can be activated. Documentation about this physical package can be found in chapters 15 to 23 of the MESO-NH documentation (IDMNHS). A brief description of each parameterization can be found in (Seity et. al., 2011).

\* **Physical package for ALARO:** This physical package will be used for mesh-sizes between 5 and 10 km, by the ALADIN partners. It has significant differences from the ARPEGE/ALADIN one. For example it uses a prognostic convection scheme.

\* **Physical package used by HIRLAM:** the entry routine in the code is `MF_PHYS`, this routine calls `APLPAR` for the complete physical package. The old HIRLAM physical package has been abandoned and only a small subset of “HL”-named routines is still used.

\* **SURFEX:** This is the surface scheme now used in AROME. The different surface processes are represented:

- surface exchanges for nature.
- surface exchanges for urban areas.
- surface exchanges for seas and oceans.
- surface exchanges for lakes.

The main entry in the code is `ARO_GROUND_PARAM`. Detailed documentation can be found in (IDSFEX).

## 9 Basics about dataflow.

\* **Objects:** Recent rewritings of the code have lead to a more object-oriented code. Variables are gathered into some sorts of objects, including for example:

- **INIT:** variables like **NCONF**, **LNHDTN**. They are set-up under **SU0YOMA**, before calling **SUGOMETRY**.
- **GEOMETRY:** variables describing horizontal and vertical geometry. They are set-up under **SUGOMETRY**.
- **MODEL:** model variables (for example horizontal diffusion coefficients). They are set-up after calling **SUGOMETRY**.
- **FIELDS:** fields, like **GMV**, **GFL** (see below).
- **MTRAJ:** trajectory variables.

Each of these main objects has subdivisions.

Variables **YRGEOMETRY**, **YRMODEL**, **YRFIELDS**, **YRMTRAJ** (declared in **CNT0**) respectively contain **GEOMETRY**, **FIELDS**, **MODEL**, **TRAJ** objects variables.

\* **Groups of prognostic variables in “FIELDS” object:** this object is divided into **GMV**, **GMVS**, **GFL**, surface fields.

- Upper-air quantities:
  - **GMV** (3D) variables ( $\mathcal{A}$  is non-zero). Example: wind components (**VOR/DIV** in spectral calculations), temperature, additional NH variables. The **GMV** variables other than the wind components or divergence/vorticity are the “thermodynamical variables” (there are **NFTHER** thermodynamical variables in the model).
  - **GMVS** (2D) variables ( $\mathcal{A}$  is non-zero). Example: logarithm of surface pressure.
  - **GFL** (3D) variables ( $\mathcal{A}$  is zero). Example: humidity, liquid water, ice, **TKE**, ozone, etc...  
This list also contains some pseudo-historic variables (ex **CPF** = convective precipitation flux).
- Surface prognostic quantities: buffers **SP...** of the surface dataflow. Examples: temperature and water content of the soil reservoirs.

\* **Spectral variables in “FIELDS” object:**

- **YRFIELDS%YRSPEC%[X]:** spectral variable for [X]. Example [X]=**VOR,DIV,T,Q,SP**.
- **YRFIELDS%YRSPEC%GFL:** all **GFL** spectral variables.
- **YRFIELDS%YRSPEC%SP3D:** all 3D variables.
- **YRFIELDS%YRSPEC%SP2D:** all 2D variables (+ the spectral orography).
- **YRFIELDS%YRSPEC%SP1D:** mean wind, in **LAM** models only.

\* **Grid-point variables in “FIELDS” object and in some additional buffers:**

- **YRFIELDS%YRGMV:** gathers the  $t - \Delta t$  and  $t$  **GMV** variables (including horizontal derivatives).
- **YRFIELDS%YRGMVT1:** gathers the  $t + \Delta t$  **GMV** variables.
- **YRFIELDS%YRGMVS:** gathers the  $t - \Delta t$  and  $t$  **GMVS** variables (including horizontal derivatives).
- **YRFIELDS%YRGMVT1S:** gathers the  $t + \Delta t$  **GMVS** variables.
- **YRFIELDS%YRGFL:** gathers the  $t - \Delta t$  and  $t$  **GFL** variables (including horizontal derivatives).
- **YRFIELDS%YRGFLT1:** gathers the  $t + \Delta t$  **GFL** variables.
- **YRFIELDS%YRSURF%SP\_[group]:** prognostic surface dataflow. In particular contains 2D surface variables used in the physics.
- **YRFIELDS%YRSURF%SD\_[group]:** diagnostic surface dataflow.
- Individual variables (for example under **CPG**):  
**P[X]T0:**  $X$  at  $t$ ; (**P[X]T0L,P[X]T0M**):  $\text{grad}(X)$  at  $t$ .  
**P[X]T9:**  $X$  at  $t$ ; (**P[X]T9L,P[X]T9M**):  $\text{grad}(X)$  at  $t - \Delta t$ .  
**P[X]T1:**  $X$  at  $t + \Delta t$ .  
Sometimes appendix F for full level, H for half level.
- Semi-Lagrangian buffer for interpolations: **SLBUF1** or **PB1**. Stores the quantities to be interpolated.
- **PB2** stores some quantities to be communicated between the different grid-point calculations, which cannot be stored in the **GMV+GFL** arrays.

## 10 The tangent linear (TL) and adjoint (AD) codes.

### 10.1 The tangent linear (TL) code.

\* **Basics about TL code.** The TL code computes the evolution of a small perturbation  $\delta X$  assuming that the evolution of this perturbation is linear. Assuming that the direct model computes the evolution of  $X$ :

$$\frac{dX}{dt} = f(X)$$

the TL code computes the evolution of  $\delta X$  as:

$$\frac{d[\delta X]}{dt} = f'(X)[\delta X]$$

(if  $X$  is a vector,  $f'(X)$  is a matrix). If the model has  $N_{\text{stop}}$  timesteps, it is first necessary to integrate the direct model in order to compute the ‘‘trajectory’’  $f'(X)$  (or some intermediate variables allowing to recover  $f'(X)$ ) at each timestep. The TL model is then integrated on  $N_{\text{stop}}$  timesteps.

Let us give an example: The direct model discretizes the scalar 2D equation:

$$\frac{dX}{dt} = KX^2$$

where  $K$  is a constant. So the tangent linear model discretizes the equation:

$$\frac{d[\delta X]}{dt} = 2KX[\delta X]$$

At each timestep of the TL code, the code will look like (for a leap-frog scheme and omitting the advection terms)

$$[\delta X](t + \Delta t) = [\delta X](t - \Delta t) + 2KX_5(t)[\delta X](t)$$

This discretisation uses the trajectory  $X_5(t)$  previously computed by the direct model. In the code, variables containing the trajectory have a name generally ending by ‘5’.

The TL code has generally the same structure as the direct code, but routines containing TL code have a name with an additional suffix ‘TL’. For example, the tangent linear routine of **STEPO** is **STEPOTL**. Some direct routines representing purely linear processes have no separate TL code (the TL code is identical to the direct code). This is the case for example for **SPCHOR**.

#### \* Configurations using TL code.

- Configuration 501 allows to test the TL code in the 3D primitive equations hydrostatic model. The way to do that is to check that the following quantity computed in spectral space for a couple of wavenumbers  $(m, n)$  converges towards 1 when the coefficient  $C$  converges towards zero:

$$R_{N_{\text{stop}}}(m, n) = \frac{Y_{N_{\text{stop}}}(m, n) - X_{N_{\text{stop}}}(m, n)}{C[\delta X]_{N_{\text{stop}}}(m, n)}$$

where  $Y_{N_{\text{stop}}}$  (resp.  $X_{N_{\text{stop}}}$ ) is the direct forecast done with an initial state of  $X_0 + C[\delta X]_0$  (resp.  $X_0$ );  $[\delta X]_{N_{\text{stop}}}$  is the TL forecast done with an initial increment of  $[\delta X]_0$ . This configuration runs the direct model and the TL model for several values of  $C$ ,  $C$  becoming smaller and smaller. Configuration 501 could do the same thing for the non-hydrostatic model but the TL code is currently incomplete for the non-hydrostatic model.

- Configuration 521 does the same thing as configuration 501 but in the 2D shallow-water model.
- Configuration 522 does the same thing as configuration 501 but in the 2D vorticity model.
- Some other configurations use the TL code: configuration 131 (minimization in the incremental 4D variational algorithm), configuration 601 (eigenvalue/vector solver for 3-D hydrostatic primitive equation model) and some options of the configuration 801.

\* **Existing documentations about TL code.** The reader will find more details about the TL code in some other documentations. For basics about TL code and use in the ARPEGE/ALADIN/AROME environment, see (IDADJINTRO), (IDADJAPPLI), or the part 26 of documentation (IDEUL). For details about configuration 501, 521 and 522 see (IDTESTLI) (this documentation is valid for an old cycle 13r3 but the most part of this documentation is still valid) or the part 26 of documentation (IDEUL).

## 10.2 The adjoint (AD) code.

\* **Basics about AD code.** The linear model coded in the TL code can be written as a matricial product:

$$[\Delta X]_{N_{\text{stop}}} = \mathbf{M}[\Delta X]_0$$

Taking the scalar product between  $[\Delta X]_{N_{\text{stop}}}$  and another vector denoted by  $[\Delta Y]$  writes:

$$\langle [\Delta X]_{N_{\text{stop}}}, [\Delta Y] \rangle = \langle \mathbf{M}[\Delta X]_0, [\Delta Y] \rangle$$

It can be rewritten:

$$\langle [\Delta X]_{N_{\text{stop}}}, [\Delta Y] \rangle = \langle [\Delta X]_0, \mathbf{M}^T[\Delta Y] \rangle$$

$\mathbf{M}^T$  is the adjoint operator of  $\mathbf{M}$ . In a finite dimension vectorial space,  $\mathbf{M}$  is represented by a square matrix and  $\mathbf{M}^T$  is its transpose.

For coding the adjoint code, one needs to get first the TL code. One starts from the TL code and “adjoints” the code: for example a matricial set of expressions of the following type

$$Y = \mathbf{M}X$$

$$Z = \mathbf{N}X$$

leads to the expression

$$X = X + \mathbf{M}^T Y + \mathbf{N}^T Z$$

in the adjoint code ( $X$  is set to zero in the beginning of the adjoint code).

The AD code has generally the same structure as the TL code, but in a reverse order for the part of the code which is not a purely “trajectory code”. Routines containing AD code have a name with an additional suffix “AD”. For example, the adjoint routine of **STEPO** is **STEPOAD**.

\* **Configurations using AD code.**

- Configuration 401 allows to test the AD code in the 3D primitive equations hydrostatic model. The way to do that is to compare the two following scalar products:

$$S1 = \langle \mathbf{M}[\delta X]_0, [\delta Y]_{N_{\text{stop}}} \rangle$$

and:

$$S2 = \langle [\delta X]_0, \mathbf{M}^T[\delta Y]_{N_{\text{stop}}} \rangle$$

the vector  $\mathbf{M}[\delta X]_0$  being obtained after a TL integration of  $N_{\text{stop}}$  timesteps from the initial perturbation  $[\delta X]_0$ , and the vector  $\mathbf{M}^T[\delta Y]_{N_{\text{stop}}}$  being obtained after an AD integration of  $N_{\text{stop}}$  timesteps from the perturbation  $[\delta Y]_{N_{\text{stop}}}$ . Both scalar products  $S1$  and  $S2$  must be equal. Configuration 401 could do the same thing for the non-hydrostatic model but the AD code is currently incomplete for the non-hydrostatic model.

- Configuration 421 does the same thing as configuration 401 but in the 2D shallow-water model.
- Configuration 422 does the same thing as configuration 401 but in the 2D vorticity model.
- Some other configurations use the AD code: configuration 131 (minimization in the incremental 4D variational algorithm), configuration 601 (eigenvalue/vector solver for 3-D hydrostatic primitive equation model) and configuration 801 for sensitivity studies.

\* **Existing documentations about AD code.** The reader will find more details about the AD code in some other documentations. For basics about AD code and use in the ARPEGE/ALADIN/AROME environment, see (IDADJINTRO), (IDADJAPPLI), or the part 27 of documentation (IDEUL). For details about configuration 401, 421 and 422 see (IDTESTAD) (this documentation is valid for an old cycle 13r3 but the most part of this documentation is still valid) or the part 27 of documentation (IDEUL).

# 11 The diagnostics.

## 11.1 Introduction.

To study the outputs of the model, one needs to provide some diagnostics, often on surfaces which are not necessarily the horizontal surface of model calculation (the hybrid coordinate surfaces in ARPEGE/IFS, ALADIN and AROME). These diagnostic surfaces can be for example the hydrostatic pressure surfaces or the potential vorticity surfaces. For some applications, diagnostics can represent a significant part of the code. There are currently several type of diagnostics coded in the ARPEGE/IFS, ALADIN and AROME software:

- FULL-POS: the post-processing on different type of surfaces (for example temperature on 700 hPa surface or geopotential height on 500 hPa).
- DDH (abbreviation of the French expression “Diagnostics sur Domaines Horizontaux”, i.e. diagnostics on horizontal domains): to perform diagnostics based on the flux-form of an equation. Such diagnostics can be used to study for example what are the different contributions to the evolution of the total amount of moisture in a box bounded by two model layers and by some latitudes and longitudes.
- CFU (cumulated fluxes): to diagnose some temporally cumulated fluxes computed from the physical parameterizations, on model surfaces. They can be used for example to diagnose the total amount of rainfall which will fall on the ground between the forecast ranges 12h and 24h.
- XFU (instantaneous fluxes): to diagnose some instantaneous fluxes computed from the physical parameterizations, on model surfaces. They can be used for example to diagnose the surface rainfall intensity at a precise instant.
- CHKEVO: this non-standard name is actually the name of a routine which is used for this diagnostics software. These diagnostics allow to study the temporal evolution of some quantities.
- Physical tendencies diagnostics.
- MCFU (monitoring the coupling update frequency) diagnostics.
- SURFEX diagnostics.
- SLDIA (semi-Lagrangian) diagnostics.
- We can add to the previous list some norms printings, the main one being the spectral norms printed by routine **SPNORM**.

We now will give more details about these diagnostics.

## 11.2 FULL-POS.

\* **What does FULL-POS do:** FULL-POS is the post-processing and allows to do the post-processing of different types of variables on different types of surfaces and on different types of horizontal domains.

The different post-processable types of variables are:

- 3D dynamical variables (like temperature, specific humidity), abbreviated into DYN3D.
- 2D dynamical variables (like logarithm of surface hydrostatic pressure), abbreviated into DYN2D.
- Surface fields used in the physics (for example the surface albedo or the land-sea mask), or produced by the physics (for example the surface reservoir content of water), abbreviated into PHYSOL.
- Fields computed by the CFU (cumulated fluxes) software.
- Fields computed by the XFU (instantaneous fluxes) software.

The different types of surfaces of post-processing are:

- Hydrostatic pressure surfaces.
- Geopotential height surfaces.
- Hybrid coordinate model surfaces.
- Potential temperature surfaces.
- Potential vorticity surfaces.
- Temperature surfaces.
- Flight level surfaces.

The different types of horizontal domains of post-processing are:

- The whole Earth in spectral representation.
- The whole Earth in grid-point representation on a reduced or not reduced Gaussian grid.
- The whole Earth in grid-point representation on a “lat-lon” grid.
- A LAM sub-domain in spectral representation.

- A LAM sub-domain in grid-point representation.
- A “lat-lon” sub-domain in grid-point representation (in this case several domains can be treated in the same job).

The main usage of FULL-POS is to obtain fields on different types of surfaces and different types of horizontal domains in order to plot them, for example the geopotential height at 500 hPa, the mean sea level pressure, the total amount of forecast rainfall, but another important application of FULL-POS is to make a file in a different geometry (with a different horizontal geometry and also a different spacing of vertical levels). For example, what one can do:

- make a global file of truncation  $N_{s2}$ , stretching  $c_2$ , high resolution pole of coordinates  $(\theta_{pe2}, \lambda_{pe2})$ , vertical hybrid coordinate defined by a pair of coefficients  $(A_2, B_2)$ , from a global file of truncation  $N_{s1}$ , stretching  $c_1$ , high resolution pole of coordinates  $(\theta_{pe1}, \lambda_{pe1})$ , vertical hybrid coordinate defined by a pair of coefficients  $(A_1, B_1)$ .
- make an ALADIN or AROME file from a global ARPEGE file.
- make an ALADIN or AROME file from an ALADIN file covering a bigger domain.

It is also possible to do the following applications:

- change a spectral field into its grid-point representation (or the reverse thing) in a file.
- add fields containing zeros (or some climatological data for PHYSOL surface fields) in ARPEGE or ALADIN files.
- add non-hydrostatic variables in files initially containing only hydrostatic variables, in order to be used by a non-hydrostatic model.
- remove the poles in grid-point fields in old ARPEGE files still containing poles.

FULL-POS is available only in configuration 1. It is not available for the 2D models (configurations 201 and 202).

\* **Basics about the architecture of FULL-POS:** First there is an important number of set-up routines concerning FULL-POS. FULL-POS can be used in an “off-line” context: one launches a job with **NSTOP=0** (zero timestep) and doing only post-processing on the initial situation (an initial file is read and some fields are diagnosed). But FULL-POS can be also used in an “in-line” context: one launches a job with **NSTOP>0** doing a forecast, and for a subset of forecast ranges (for example 6h, 12h, ...) some fields can be post-processed. When post-processing is required for a certain time-step, there are calls to routine **STEPO\_FPOS** governing calls to horizontal or vertical interpolations, and additional file writing to store the post-processed fields. **STEPO\_FPOS** has now its own definition of **CLCONF** which is different from the model one.

When using FULL-POS through configuration 1, the main architecture under **CNT4** is generally:

```
CNT4 ->
- SU4FPOS
- GRIDFPOS ->
  * SUFPSUW -> CPCLIMI -> HPOS, FPOSHOR and FPOSHORLAG (for input climatological
    surface fields if post-processing of such fields is required).
  * SCAN2M_HPOS -> HPOS for PHYSOL, CFU, XFU
  * FPOSHOR and FPOSHORLAG for PHYSOL, CFU, XFU
  * WRHFP (writing PHYSOL, CFU, XFU)
- PREDYNFPOS
- DYNFPOS ->
  * STEPO_FPOS ->
    - SCAN2M_VPOS -> VPOS -> POS (unlagged vertical interpolations)
    - TRANSDIR_FP (model direct transforms)
    - SPOS and SPOSGF (spectral calculations in model space)
    - TRANSINV_FP (model inverse transforms)
    - [SCAN2M_HPOS -> HPOS] and [FPOSHOR+FPOSHORLAG] (horizontal interpolations)
    - in some cases [SCAN2M_MPOS -> HPOS and VPOS] and [FPOSHOR+FPOSHORLAG] (hor+vert interpolations)
    - ENDVPOS -> ENDPOS (lagged vertical interpolations)
    - TRANSDIR_FP (output direct transforms)
    - SPOS and SPOSGF (spectral calculations in output space)
    - TRANSINV_FP (output inverse transforms)
  * IOFPOS -> WRHFP and WRSFP (writing grid-point and spectral DYN3D, DYN2D)
```

Most of vertical interpolations are done before the horizontal interpolations.

An alternate way to do off-line FULL-POS is to use configuration 903. The code does not pass through **CNT0** -> **CNT1** -> **CNT2** -> **CNT3** -> **CNT4** but through **CNT0** -> **CPREP3**. The main architecture under **CPREP3** is generally:

CPREP3 ->  
- SU4FPOS  
- SUFFINIF (reads input files)  
- SUFFRFPBUF\_CLIM (reads output geometry climatology)  
- GRIDFPOS -> (see above)  
- DYNFPOS -> STEPO\_FPOS -> (see above)  
- and some other routines

About set-up, the main features to be known for beginners are:

CNT0 ->  
- SUOYOMA -> SUGOMETRY, SUFFC  
- SUOYOMB -> SUPPVI, SUBFPOS

The code used for vertical interpolations is the same as the one used to interpolate model data at the observation locations in the assimilation. One can find it under the sequence **VPOS** -> **POS**. For horizontal interpolations there are specific routines, different from the ones used for interpolations in the assimilation or in the semi-Lagrangian scheme. The code for horizontal interpolations can be found under the routines **HPOS** and **FPOSHOR**. For some fields it is necessary to do some computations after horizontal interpolations and these calculations are done under routine **ENDVPOS**.

Writing post-processed data on files is generally done by routines **WRHFP** (grid-point fields) and **WRSFP** (spectral fields). Some applications of FULL-POS can require the reading of climatological files (input geometry climatological file and output geometry climatological file).

The activation of FULL-POS is done by setting variable **NFPOS** to an allowed non-zero integer in namelist **NAMCT0**.

\* **More documentation about FULL-POS:** For more details about the architecture of the code and the algorithms involved, see (IDFPOS).

### 11.3 DDH.

\* **Purpose of these diagnostics.** The aim is to use the flux form of the evolution equation of an intensive quantity  $X$ , in order to diagnose the different contributions which act on the total content of  $Q_X$  in a box,  $Q_X$  being the additive quantity which is associated with  $X$ . These different contributions are:

- each contribution coming from the physical parameterizations.
- each contribution coming from the adiabatic part.
- the residual contribution coming from horizontal diffusion and the semi-implicit correction.

If we denote by  $Q_1$  the total mass in a box (the corresponding intensive quantity is  $X = 1$ ), the relationship between  $X$  and  $Q_X$  writes:

$$Q_X = Q_1 X$$

( $Q_X$  is the content of  $X$  in the same box). For example  $X$  can be  $q$  (specific humidity),  $\mathbf{V}$  (momentum).

The flux form of the Eulerian equation of  $X$  writes:

$$\frac{\partial \left[ \frac{\partial \Pi}{\partial \eta} X \right]}{\partial t} = -\nabla \left[ \mathbf{V} \frac{\partial \Pi}{\partial \eta} X \right] - \frac{\partial \left[ X \dot{\eta} \frac{\partial \Pi}{\partial \eta} \right]}{\partial \eta} + \left[ \frac{\partial \Pi}{\partial \eta} \right] \left[ \frac{dX}{dt} \right]_{\text{ad}} + \left[ \frac{\partial \Pi}{\partial \eta} \right] F_X - gX \frac{\partial F_p}{\partial \eta} \quad (7)$$

The box where such diagnostics are done can be the whole atmosphere or a box bounded by two vertical layers, two longitudes and two latitudes. For example, one can use these diagnostics to diagnose the evolution of the whole mass of the atmosphere, or the whole content of humidity of the atmosphere; in particular this is a tool to check the temporal evolution of some invariants of the atmosphere (and thus the conservation properties of the model).

The DDH diagnostics are currently available in configuration 1.

\* **Basics about the architecture of the DDH code:** Some specific setup routines define the domains where the diagnostics are done, and also the list of variables which are diagnosed. The main body of the diagnostics is done in the grid-point part under the sequence **STEPO** -> **SCAN2M** -> **GP\_MODEL** -> **CPG\_DRV** -> **CPG** -> **CPG\_DIA** in routines **CPDYDDH**, **CPPHDDH** and **CPCUDDH**. For ECMWF physics an additional routine **CPPHDDHE** is called under the sequence **EC\_PHYS** -> **POSTPHY**. Printings and writing on files are done in routine **POSDDH** called by **STEPO**.

\* **More documentation about the DDH code:** Description of the DDH diagnostics and user's guide is provided in the documentations (IDDDH), (IDDDHU), (IDDDHN). The formulation and the discretisation of the flux form of the equation of a variable  $X$  is described in section 20 of documentation (IDEUL).

## 11.4 CFU and XFU.

\* **Purpose of these diagnostics.** CFU (cumulated fluxes) are used to diagnose some temporally cumulated fluxes computed from the physical parameterizations, on model surfaces. They can be used for example to diagnose the total amount of rainfall which will fall on the ground between the forecast ranges 12h and 24h. XFU (instantaneous fluxes) are used to diagnose some instantaneous fluxes computed from the physical parameterizations, on model surfaces. They can be used for example to diagnose the surface rainfall intensity at a precise instant. The cumulated flux  $\bar{F}$  is computed from the instantaneous flux  $F(t)$  by the formula:

$$\bar{F} = \int_{t'=0}^{t'=t} F(t')dt \quad (8)$$

The CFU and XFU diagnostics are currently available in configuration 1 (they can be used for both hydrostatic and non-hydrostatic models).

\* **Basics about the architecture of the CFU and XFU code:** Some specific setup routines define the list of variables which are diagnosed. The main setup routines are **SUCFU** for CFU and **SUXFU** for XFU, called by **SU0YOMB**. The main body of the diagnostics is done in the grid-point part under the sequence **STEPO** -> **SCAN2M** -> **GP\_MODEL** -> **CPG\_DRV** -> **CPG** -> **CPG\_DIA** in routines **CPCFU** for CFU and **CPXFU** for XFU (valid for the METEO-FRANCE unlagged physics). For ECMWF physics the specific routine **CPEDIA** is called under the sequence **EC\_PHYS** -> **POSTPHY**. Writing on files is done in routines **WRFU** (CFU) and **WRXFU** (XFU) (for METEO-FRANCE configurations). Activation of these diagnostics is controlled by the variables **LCUMFU** in **NAMCFU** (CFU) and **LXFU** in **NAMXFU** (XFU).

\* **More documentation about the CFU and XFU code:** Description of the CFU and XFU diagnostics is provided in the documentation (IDFU), at least for configurations used at METEO-FRANCE.

## 11.5 CHKEVO.

\* **Purpose of these diagnostics.** These diagnostics allow to study the temporal evolution of some quantities (the prognostic variables of the model) on the whole domain, and also to print some values (of the prognostic variables of the model) for one or several grid-points.

The CHKEVO diagnostics are currently available in configuration 1 (they can be used for both hydrostatic and non-hydrostatic models).

\* **Basics about the architecture of the CHKEVO code:** The main setup routine is **SUECHK**, called by **SU0YOMA**. Diagnostics and file writing are done by routine **CHKEVO** called by **CNT4**. Activation of these diagnostics is controlled by the variable **LECHKEVO** in namelist **NAMCHK**.

\* **More documentation about the CHKEVO code:** There is currently no specific documentation about these diagnostics.

## 11.6 Physical tendencies diagnostics.

\* **Purpose of these diagnostics.** These diagnostics allow to study the physical tendencies, for one or several grid-points, for example to understand the formation of some instabilities.

The physical tendencies diagnostics are currently available in configuration 1 for physics packages called under **MF\_PHYS** (they can be used for both hydrostatic and non-hydrostatic models).

\* **Basics about the architecture of the code:** The main setup routine is **SUCHET**, called by **SU0YOMA**. Diagnostics are done by routines **CPCHET** and **PROFILECHET** called by **MF\_PHYS**. The use of these diagnostics is controlled by the variable **GCHETN** in namelist **NAMCHET**.

\* **More documentation about the physical tendencies diagnostics code:** There is currently no specific documentation about these diagnostics.

## 11.7 MCUF diagnostics (monitoring the coupling update frequency).

\* **Purpose of these diagnostics.** They diagnose a variable based on  $\log \Pi_s$  in order to know if, in a coupling file for LAM models, there is a risk to miss some important features due to a too large time interval of coupling.

The MCUF diagnostics are currently available in configuration 1 (they can be used for both hydrostatic and non-hydrostatic models).

\* **Basics about the architecture of the code:** The main setup routine is **SUMCUF**, called by **SU0YOMB**. Diagnostics are done by routine **SPMCUF** called by **CNT4**. The use of these diagnostics is controlled by the variable **LMCUF** in namelist **NAMMCUF**.

\* **More documentation about the MCUF diagnostics code:** Description of the MCUF diagnostics is provided in the internal note (IDMCUF).

## 11.8 SURFEX diagnostics.

The SURFEX package (used for AROME and MESO-NH surface processes) has its own diagnostics, called under the sequence **STEPO** → **IOPACK** → **ARO\_SURF\_DIAGH** → **ARO\_SURF\_DIAG** (MSE project).

## 11.9 SLDIA diagnostics.

Semi-Lagrangian diagnostics, which are done in the semi-Lagrangian scheme. Calculations are done in routines **LAPINEA**, **LAPINEB** and **CPDYSLDIA**. The use of these diagnostics is controlled by the variable **LSLDIA** in namelist **NAMDYNA**.

\* **More documentation about SLDIA diagnostics:** There is currently no specific documentation about these diagnostics.

## 11.10 Spectral and grid-point norms.

Some routines do printings on the listing of spectral or grid-point statistics. The list of main routines is:

- **FPGPNORM**: average, minimum and maximum values of a grid-point field (in FULL-POS).
- **GRNORMA**: average of a grid-point field.
- **SPNORM**: spectral norms in spectral space for prognostic variables.
- **SPECNORM** (in TRANS): spectral norms in spectral space (global model).
- **ESPECNORM** (in ETRANS): spectral norms in spectral space (LAM model).

## 12 Screening and observation interpolator.

### 12.1 What does the screening do.

The observations which are needed in the assimilation are not always exact, and not always conveniently distributed, so it is necessary to eliminate some wrong observations, or some observations in some places where there are too many observations. Such a work can be partly done in a separate task for the part of the work which does not require a comparison with a model forecast (this is the case for example if an observation provides a 2m temperature of 92 degrees instead of 33 degrees because the observer has used the Fahrenheit scale instead of the Celsius one). But there remains a part of this work which requires a comparison with the model forecasts. Screening is this last step of the pre-treatment of the data. The observations are controlled against the background, to verify their vertical consistency, and then thinned when their spatial density is too high compared to the resolution of the analysis.

Screening consists in running the direct forecast model, plus one or several calls to the observation operators, followed by the first guess check, the quality control and the thinning of the observations. One usually separates 3D-screening (where only the initial model state is read and observation operators are called once), from 4D-screening where the full forecast is actually computed and observation operators are called in the middle of each timeslot. In 3D-screening all the observations are considered as co-located in time, while in 4D-screening, the observations are distributed into timeslots (for example 1 hour timeslots in the operational model). A comparison can be done between the screening and FULL-POS for the code architecture: the 3D-screening can be compared to an “off-line” version of FULL-POS (**NSTOP**=0) and the 4D-screening can be compared to an “in-line” version of FULL-POS (**NSTOP**>0) where FULL-POS diagnostics would be called for example once per hour. The code architecture of the screening looks like some options of the code architecture of FULL-POS. Algorithms require to do interpolations at the locations of observations: there are horizontal and vertical interpolations, and some vertical interpolations routines are used both for FULL-POS and the observation interpolator. The screening is implemented only in configuration 1.

### 12.2 Basics about the architecture of the screening code.

There are specific set-up routines, in particular under the sequence **CNT1** – > **SUOBS**.

Under **CNT4** one finds routines doing horizontal and vertical interpolations (model data must be interpolated at the observations locations, and comparison between observations and model data are done at the observations locations). The main architecture under **CNT4** is:

```
CNT4 ->
- STEPO('0AA00V000') ->
  * inverse spectral transforms
  * SCAN2M -> COBSALL
- OBSV
```

**COBSALL** manages the horizontal interpolations, **OBSV** manages the vertical interpolations.

For the time being the horizontal interpolations are performed before the vertical interpolations.

After the interpolations and at the end of the time-integration step one returns in **CNT1**: selection of observations and writing of the selected observations on files is done according the following organigramme:

```
CNT1 ->
- SCREEN
- SACMAC1
```

**SCREEN** does the comparison between observations and model at the observations locations and uses some selection criteria to eliminate undesired observations; **SACMAC1** writes the selected observations on a file. Variable **LSCREEN** of **NAMCT0** must be set to .T. to activate the screening.

### 12.3 More documentation about the screening.

The main aspects of the screening are described in the part 4.1 of documentation (IDVAR) and more detailed organigrammes are provided.

For ECMWF model, screening and observation interpolator description is done in several parts of the ECMWF documentation:

- Observation operator is described in part 1 of (TDECOBS). In particular, part 1.3 gives information about observation vertical interpolator (some routines are also used in FULL-POS).
- Screening is described in part 3 of (TDECOBS).

For more details about vertical interpolations (and extrapolations) used in FULL-POS and in the observation interpolator, see part 3 of documentation (IDFPOS). For more details about horizontal interpolations used in the semi-Lagrangian scheme and also in the observation interpolator, see part 13 of documentation (IDSL).

## 13 The initialisation.

### 13.1 The different types of initialisation schemes.

In the atmosphere there are “slow” evolving waves (Rossby waves) and “fast” evolving waves (for example gravity waves). The aim of the initialisation is to control the amount of gravity waves by some filtering in the analysis. There are mainly two classes of algorithms to do that: the normal mode initialisation (NMI) and the digital filter initialisation (DFI). The normal mode initialisation can be easily implemented in a non-transformed spherical geometry but the presence of tilting and stretching brings some difficulties to implement such a scheme (it is necessary in this case to do some calculations in the geographical space). The digital filter initialisation can be implemented on all types of geometry. It performs better in a stretched tilted spherical geometry than the NMI initialisation scheme. The NMI scheme is no longer used. METEO-FRANCE currently uses only the DFI scheme.

### 13.2 Normal mode initialisation (NMI).

\* **Algorithm:** A variable  $X$  can be written  $X_R + X_G$  where  $X_R$  is the slow component (Rossby waves) and  $X_G$  is the fast component (gravity waves). The linear NMI initialisation replaces  $X_G$  by zero. The non-linear NMI initialisation replaces  $\frac{dX_G}{dt}$  by zero. The separation between slow and fast modes is first done vertically using a linear operator (which is also used in the semi-implicit scheme). Each vertical mode has its own list of horizontal modes. The fastest vertical modes are eliminated. For the slowest vertical modes only a subset of horizontal modes is kept. The NMI scheme is no longer implemented in ARPEGE/IFS (pruned in 2012).

\* **Documentation:** A rather short description of the NMI algorithm and its implementation in IFS can be found in the chapter 7 of ECMWF documentation (TDECDCAS), this is the most recent available documentation. Some documentations have been written in the times on METEO-FRANCE side but they are now partly obsolete. One can find in part 2.6 of (NTA22) a brief description of the NMI scheme as it has been implemented in ARPEGE in the beginning of the 1990s. More details about the algorithms and the code as it was in the beginning of the 1990s is described in (NTA23). Documentation (IDNMI) provides some namelists allowing to run the NMI scheme, but this documentation has also a significant couple of years. Some other documentations such (NTA31) or (NTA33), written in the beginning of the 1990s, give more theoretical aspects, and some examples of alternative proposals which have probably never been implemented in ARPEGE/IFS.

### 13.3 Digital filter initialisation (DFI).

\* **Algorithm and basics about the organigramme:** Contrary to the NMI initialisation which is a purely spatial filtering, the DFI initialisation is a temporal filtering and requires a forward forecast and also a backward forecast in order to display a model state in a time span centred on  $t_0$ . If  $M_{\text{span}}$  is the number of integration timesteps of the time span, high frequency oscillations of  $X$  are removed by replacing  $X$  by a “good” temporal average of  $X$ :

$$\bar{X}(t_0) = \sum_{k=-M_{\text{span}}}^{k=M_{\text{span}}} \alpha_k X(t_k)$$

where  $t_k = t_0 + k\Delta t$ . This temporal average is computed in spectral space. Such a scheme is used at METEO-FRANCE, and is well adapted to a variable-mesh version of ARPEGE (there is no need to perform calculations in the geographical spectral space). The organigramme includes specific setup routines (in particular **SUDFI**) and some code under the sequence **CNT3** – > **DFI**. DFI scheme can be done by setting **LDFI=T** in namelist **NAMINI**. DFI scheme is implemented in configuration 1, and the linear tangent and adjoint codes are also provided.

\* **Documentation:** A rather short description of the DFI algorithm and its implementation in IFS can be found in the chapter 7 of ECMWF documentation (TDECDCAS), this is the most recent available documentation. More details about the theoretical aspects of the DFI can be found in (Lynch et al., 1997), (Lynch et al., 1999), (IDDFI01). Some documentations have been written in the times on METEO-FRANCE side, for example (IDDFI94), but they are based on old cycles (IDDFI94 is valid for cycle 11r2) and some parts probably need to be updated. (IDDFI94) is more oriented towards the description of the code and also of the namelists. A more synthetic update of (IDDFI94) has been written (IDDFI).

## 14 Algorithms of assimilation.

### 14.1 Introduction: the different algorithms.

\* **Main description:** The different parameters of the atmosphere are regularly measured, and the assimilation techniques have to use these observations to make an initial analysis from that the model forecast can start. The technique which has commonly be used during the last years is to modify a forecast (generally a 6 hours forecast) by observations. Of course, observations are not horizontally and vertically regularly spaced, and their locations are not the ones of the model grid-points. Doing interpolations at the model grid-points is not satisfactory, and clearly difficult for some observations with a very irregular spatial repartition. The facts that must be taken into account are the following ones:

- Model variables are not independent each other.
- What occurs at a location is correlated with what occurs in the neighbourhood of this location, this correlation generally decreases when the distance between two points increases.
- An observation will influence not only the point where it is located but also its neighbourhood.
- An observation is not completely perfectly just, and is representative of a region around this observation; so the statistics of observation errors must be taken into account. Less accurate observations must have a smaller weight in the assimilation algorithm than more accurate observations.
- The provided analysis must be well balanced, the forecast starting from it must not generate spurious gravity waves and the minimum spin-up as it is possible to do.

The assimilation algorithm has the following sequences:

- One starts from a guess: it is generally a 6 hours forecast (1 hour in the hourly cycle for AROME).
- One modifies the guess using information given by observations. The final result must be a good compromise between the observations and the guess.

The guess provides  $\mathbf{X}_g$ , the observations provide  $\mathbf{Y}$ . The solution  $\mathbf{X}$  (the increment is denoted by  $\delta\mathbf{X} = \mathbf{X} - \mathbf{X}_g$ ) must minimize the cost function:

$$J(\mathbf{X}) = J_b(\mathbf{X}) + J_o(\mathbf{X}) + J_c(\mathbf{X})$$

where:

- $J_b$  measures the distance from the guess.

$$J_b = 0.5(\mathbf{X} - \mathbf{X}_g)^T \mathbf{B}^{-1}(\mathbf{X} - \mathbf{X}_g) = 0.5(\delta\mathbf{X})^T \mathbf{B}^{-1}(\delta\mathbf{X})$$

$\mathbf{B}$  is the background error covariance matrix.  $\mathbf{B}$  takes into account the dependence between each variable, the horizontal auto-correlation of each variable and the vertical auto-correlation of each variable. Using the denotations of documentation (IDVAR) and omitting the spectral transforms, matrix  $\mathbf{B}$  can be written as  $\mathbf{B} = [\mathbf{K}(\mathbf{SD})(\mathbf{VW})][\mathbf{K}(\mathbf{SD})(\mathbf{VW})]^T$  where  $\mathbf{K}$  is the cross-covariance matrix between each variable,  $(\mathbf{SD})$  is the horizontal auto-correlation matrix of each variable and  $(\mathbf{VW})$  is the vertical auto-correlation matrix of each variable.

- $J_o$  measures the distance from the observations.

$$J_o = 0.5(\mathbf{H}\mathbf{X} - \mathbf{Y})^T \mathbf{R}^{-1}(\mathbf{H}\mathbf{X} - \mathbf{Y}) = 0.5(\mathbf{H}\delta\mathbf{X} - (\mathbf{Y} - \mathbf{H}\mathbf{X}_g))^T \mathbf{R}^{-1}(\mathbf{H}\delta\mathbf{X} - (\mathbf{Y} - \mathbf{H}\mathbf{X}_g))$$

$\mathbf{R}$  is the observation error matrix.  $\mathbf{H}$  is the operator doing interpolations from the model grid-points towards the observations locations. This operator also takes account of the fact that the observations do not give always information in the same unit than the model variables. Satellites measure radiances but the corresponding model variable is temperature. Radiosoundings measure relative humidity but the corresponding model variable is the specific humidity.

- $J_c$  is a filtering constraint which is optional. For example this filtering constraint can be a digital filtering and in this case  $J_c$  is proportional to  $(\delta\mathbf{X} - \overline{(\delta\mathbf{X})})^T (\delta\mathbf{X} - \overline{(\delta\mathbf{X})})$  where  $\overline{(\delta\mathbf{X})}$  is a temporal average of  $\delta\mathbf{X}$  done with the same coefficients as in the digital filtering initialisation.

It is more usual and convenient to use the incremental formulation with  $\delta\mathbf{X}$ . The assimilation algorithm finds the unknown  $\delta\mathbf{X}$ , then another jobs adds  $\delta\mathbf{X}$  to  $\mathbf{X}_g$ . There are different ways to find  $\delta\mathbf{X}$ :

- The optimal interpolation algorithm: it assumes that all observations are done at the same instant which is the instant of the assimilation. At METEO-FRANCE it is still used to assimilate some surface and soil data in in the CANARI analysis (conf E701), the surface analysis used in ARPEGE, ALADIN-OM and AROME.
- The 3D-VAR (three-dimensional variational) assimilation algorithm: it assumes that all observations are done at the same instant which is the instant of the assimilation.

- The 4D-VAR (quadri-dimensional variational) assimilation algorithm: it takes account of the fact that the observations are done at different instants; the maximum time interval of the observations taken in account is generally 6h (this is the case in the current 4D-VAR assimilation done at METEO-FRANCE). At METEO-FRANCE it is used to assimilate upper air data and surface pressure. The 4D-VAR assimilation algorithm is also used in the operational ECMWF model.
- There is an hybrid algorithm between 3D-VAR and 4D-VAR called 3D-VAR FGAT (FGAT meaning “First Guess at Appropriate Time”).
- The 4D-EnVar (quadri-dimensional ensemble variational) assimilation algorithm: alternative algorithm using products of ensemble predictions (see section 16).

The optimal interpolation technique has been largely used in the past, variational techniques have been implemented more recently (in the second part of the 1990’s). Assimilation can be done in several steps in increasing smaller resolutions  $N_{s1}, N_{s2}, \dots$  (all being lower than the forecast resolution): this strategy is called multi-incremental assimilation. Multi-incremental assimilation is used when a 4D-VAR algorithm is involved to reduce the cost of the assimilation.

\* **Case of limited area models:** A first guess of the initial state can be obtained by interpolating the initial state of a global model (or of a limited area model covering a larger area). The same interpolations can be applied to the forecasts coming from the forcing model in order to make the lateral boundary conditions files for the limited area model. Applying a data assimilation algorithm to the limited area model guess is not compulsory (it is currently done in the operational AROME with a 3DVAR algorithm in a hourly data assimilation cycle), but it can be done in order to take account of small scale information coming from observations and from the guess to improve the analysis.

## 14.2 Optimal interpolation.

\* **Algorithm:** The basic idea of the optimal interpolation is to take each grid-point individually and to assume that only a limited set of observations situated in the neighbourhood of this grid-point will influence this grid-point. For the horizontal auto-covariances, they are assumed to be zero when the distance between two points is greater than a pre-defined threshold. Term  $J_c$  is zero in this case (the consequence is that an initialisation must be done in a separate job). Zeroing the derivative  $\frac{dJ}{d[\delta\mathbf{X}]}$  leads to solve the linear system:

$$\delta\mathbf{X} = (\mathbf{B}^{-1} + \mathbf{H}^T \mathbf{R}^{-1} \mathbf{H})^{-1} \mathbf{H}^T \mathbf{R}^{-1} (\mathbf{Y} - \mathbf{H}\mathbf{X}_g)$$

This linear system is solved in the grid-point space for each grid-point; the size of each linear system is at maximum several hundreds, so direct algorithms can be used to invert matrices (the routine **MINV** is current used in the code). The characteristics of the code are:

- no minimization routine is involved.
- no time integration is involved, all calculations are done at the same timestep  $t_0$ .
- the tangent linear and adjoint codes are not involved.
- most calculations are done in grid-point space.

This algorithm has been commonly used in the past, before the implementation of the variational techniques. There are some limitations of such algorithms, the main one being the difficulty to treat some satellite data which give information in some units far from the ones of the model variables (for example the radiances given by satellites); so some intermediate data conversions must be done by external intermediate tasks (for example retrieving temperature profiles from radiances) and these conversions can deteriorate the quality of data. In the operational version of ARPEGE used at METEO-FRANCE, the optimal interpolation is still used to assimilate some soil and surface data (soil temperature, soil content of humidity, temperature and humidity at 2 m).

\* **Main features of the code:** The optimal interpolation algorithm is activated by configuration 701 of ARPEGE/IFS/ALADIN and uses the “CANARI” code. The main features of the organigramme are:

```
CNTO ->
- SUOYOMA
- SUOYOMB
- CAN1 ->
* some setup routines.
* CANARI ->
- some setup routines.
- reading the observation files.
- some CA.. routines
- CADAVER -> STEPO('OM100C000') -> interpolations at the locations of observations, i.e.
* inverse spectral transforms.
```

```

* SCAN2M -> COBSALL for horizontal interpolations.
* OBSV for vertical interpolations.
- some CA.. routines.
- CAVODK -> CARNAK -> CANEVA -> preparation and inversion of the linear system.
- some CA.. routines.

```

The observation interpolator uses the direct code and the same part of the code as in the screening. Most routines specific to “CANARI” can be recognized in the code since they have a name starting by “CA”.

### 14.3 3D-VAR (three-dimensional variational) assimilation algorithm.

\* **Algorithm:** The basic idea of the variational techniques is to solve globally the problem of minimization of  $J$  by an iterative algorithm of minimization; in the ARPEGE/IFS and ALADIN frame the minimization is done in spectral space. Since  $J$  is a quadratic function, one can use efficient minimizers valid for (symmetric definite) quadratic functions: currently it is possible to use minimizers based on a quasi-Newton algorithm (**M1QN3**) or a gradient conjugate algorithm (**N1CG1** or **CONGRAD**). Matrix B has some parts computed in grid-point space and some other parts computed in spectral space. For a purely 3D-VAR assimilation algorithm, term  $J_c$  is generally set to zero (a separate initialisation is done), it is not convenient to put a DFI initialisation constraint in  $J_c$  (because it needs a temporal integration of the model which is not present in the 3D-VAR assimilation). In a 3D-VAR assimilation algorithm, all calculations are done at  $t_0$ , and all the observations which are used are assumed to be at the instant  $t_0$ . The tangent linear and the adjoint codes are not involved. This type of algorithm has been used in the operational version of ARPEGE at METEO-FRANCE between May 1997 and June 2000. It is now used in AROME. It is commonly used in some other weather forecast centres.

\* **Main features of the code:** The 3D-VAR algorithm is activated by configuration 131 of ARPEGE/IFS/ALADIN with **NSTOP=0**. The main features of the organigramme are:

```

CNT0 ->
- SUOYOMA
- SUOYOMB ->
  * reading the guess on a file.
  * SUJBBAL + 'SUJBBCOV or SUJBWAVELET': computation of matrix B.
- CVA1 ->
  * SUOBS: reading observation files.
  * SU1YOM
  * SUCOS: set up computing intermediate quantities used in the cost-function J.
  * CNT2 -> CNT3 -> CNT4 -> STEPO ->
    - inverse spectral transforms.
    - SCAN2M -> COBSALL for horizontal interpolations.
    - OBSV for vertical interpolations.
  * CVA2 ->
    - SIM4D (simulator).
    - minimization of J (call to M1QN3, N1CG1 or CONGRAD).

```

The observation interpolator uses the direct code and the same part of the code as in the screening.

### 14.4 4D-VAR (quadri-dimensional variational) assimilation algorithm.

\* **Algorithm:** The basic idea of the variational techniques is to solve globally the problem of minimization of  $J$  by an iterative algorithm of minimization; in the ARPEGE/IFS and ALADIN frame the minimization is done in spectral space. The main particularity of the 4D-VAR algorithm compared to the 3D-VAR algorithm is that the real time of observations is taken into account.

The  $J_o$  part of the cost function now writes:

$$J_o = \sum_{n_{\text{step}}=0}^{n_{\text{step}}=N_{\text{stop}}} 0.5(\mathbb{H}_L[\delta\mathbf{X}]_{n_{\text{step}}} - (\mathbf{Y}_{n_{\text{step}}} - \mathbb{H}[\mathbf{X}_g]_{n_{\text{step}}}))^T [\mathbb{R}^{-1}]_{n_{\text{step}}} (\mathbb{H}_L[\delta\mathbf{X}]_{n_{\text{step}}} - (\mathbf{Y}_{n_{\text{step}}} - \mathbb{H}[\mathbf{X}_g]_{n_{\text{step}}}))$$

where  $\mathbb{M}$  is the matricial operator of the tangent linear model ( $[\delta\mathbf{X}]_{n_{\text{step}}} = \mathbb{M}_{n_{\text{step}}}[\delta\mathbf{X}]_0$ ).  $[\mathbf{X}_g]_{n_{\text{step}}}$  is the  $n_{\text{step}}$  timesteps direct forecast from the guess  $[\mathbf{X}_g]_0$ .  $J_o$  can be rewritten:

$$J_o = \sum_{n_{\text{step}}=0}^{n_{\text{step}}=N_{\text{stop}}} 0.5(\mathbb{H}_L\mathbb{M}_{n_{\text{step}}}[\delta\mathbf{X}]_0 - (\mathbf{Y}_{n_{\text{step}}} - \mathbb{H}[\mathbf{X}_g]_{n_{\text{step}}}))^T [\mathbb{R}^{-1}]_{n_{\text{step}}} (\mathbb{H}_L\mathbb{M}_{n_{\text{step}}}[\delta\mathbf{X}]_0 - (\mathbf{Y}_{n_{\text{step}}} - \mathbb{H}[\mathbf{X}_g]_{n_{\text{step}}}))$$

The  $J_b$  part of the cost function writes:

$$J_b = 0.5([\delta\mathbf{X}]_0)^T \mathbf{B}^{-1}([\delta\mathbf{X}]_0)$$

Note that if  $t_0$  is the instant of the analysis and if  $Dt$  is the timespan, thus  $n_{\text{step}} = 0$  matches with instant  $t_0 - 0.5Dt$  and  $n_{\text{step}} = N_{\text{stop}}$  matches with instant  $t_0 + 0.5Dt$ .

The unknown is now  $[\delta\mathbf{X}]_0$ . In fact the cost function  $J$  remains quadratic because one replaces in this case the operator  $\mathbf{H}$  by its tangent linear version  $\mathbf{H}_L$ . Since  $J$  is a quadratic function, one can still use efficient minimizers valid for (symmetric definite) quadratic functions (the same ones as in the 3D-VAR assimilation): currently it is possible to use minimizers based on a quasi-Newton algorithm (**M1QN3**) or a gradient conjugate algorithm (**N1CG1** or **CONGRAD**). Matrix  $\mathbf{B}$  has some parts computed in grid-point space and some other parts computed in spectral space; it is specified at one instant (currently  $t_0 - 0.5Dt$ ) and its computation is the same as in the 3D-VAR algorithm. Term  $J_c$  can be non-zero with a DFI initialisation constraint. The tangent linear and the adjoint codes are involved to advect the increment  $[\delta\mathbf{X}]$  and also to do the observation interpolator.

This type of algorithm is currently used in the operational version of ARPEGE at METEO-FRANCE (implementation in June 2000) and also in the ECMWF IFS operational model: the timespan is 6h and  $J_c$  uses a DFI initialisation constraint. The main differences between the operational use at METEO-FRANCE and ECMWF are: list of observations assimilated, formulation of matrices  $\mathbf{B}$  and  $\mathbf{R}$ . In practical there is one additional approximation in these operational versions: a small window (timeslot) of 1 hour is defined and each observation is considered to be done at a full hour (if an observation is done at 1h47UTC it is considered to be done at 2hUTC).

\* **Main features of the code:** The 4D-VAR algorithm is activated by configuration 131 of ARPEGE/IFS/ALADIN with **NSTOP**>0. The main features of the organigramme are:

```
CNT0 ->
- SUOYOMA
- SUOYOMB ->
  * reading the guess on a file.
  * SUJBBAL + SUJBCOV: computation of matrix B.
- CVA1 ->
  * SU1YOM
  * SUOBS: reading observation files.
  * SUCOS: set up computing intermediate quantities used in the cost-function J.
  * CNT2 -> CNT3 -> CNT4 -> STEP0 -> trajectory
  * CVA2 ->
    - SIM4D (simulator).
    - minimization of J (call to M1QN3, N1CG1 or CONGRAD).
```

There is one call to the direct model to make and store the trajectory. Each call to the simulator launches the tangent linear model and the adjoint model with  $N_{\text{stop}}$  timesteps, to run the TL model and also to do the TL of the observation interpolator as following:

```
SIM4D -> CNT3TL -> CNT4TL -> STEP0TL ->
launches the TL model, and also the following for TL of observation interpolator:
- inverse spectral transforms.
- SCAN2MTL -> COBSALLTL for horizontal interpolations.
- OBSVTL for vertical interpolations.
```

## 14.5 3D-VAR FGAT assimilation algorithm.

\* **Algorithm:** The 3D-VAR FGAT assimilation algorithm is a simplified version of the 4D-VAR algorithm, a sort of hybrid algorithm between the 3D-VAR and the 4D-VAR algorithms. The main features are the same as for the 4D-VAR assimilation (in particular the fact that there is a model integration of  $N_{\text{stop}} > 0$  timesteps), but the tangent linear model which makes evolving the increment  $\delta\mathbf{X}$  is replaced by identity.

The  $J_o$  part of the cost function now writes:

$$J_o = \sum_{n_{\text{step}}=0}^{n_{\text{step}}=N_{\text{stop}}} 0.5(\mathbf{H}_L[\delta\mathbf{X}]_0 - (\mathbf{Y}_{n_{\text{step}}} - \mathbf{H}[\mathbf{X}_g]_{n_{\text{step}}}))^T [\mathbf{R}^{-1}]_{n_{\text{step}}} (\mathbf{H}_L[\delta\mathbf{X}]_0 - (\mathbf{Y}_{n_{\text{step}}} - \mathbf{H}[\mathbf{X}_g]_{n_{\text{step}}}))$$

$J_b$  has the same expression as in the 4D-VAR algorithm. Note that this algorithm is not identical to a 3D-VAR algorithm. The direct model runs with  $N_{\text{stop}}$  timesteps and computes a trajectory, and all the tangent linear calculations which remain in this algorithm (observation interpolator) are linearized around this trajectory. The direct model also keeps on to give a temporal evolution from the guess.

\* **Main features of the code:** They are the same as for the 4D-VAR assimilation scheme, but some calculations are replaced by identity in the TL and AD models. The 3D-VAR FGAT algorithm is activated by configuration 131 of ARPEGE/IFS/ALADIN with **NSTOP**>0, and with **L3DFGAT**=.T. in **NAMVAR**.

## 14.6 Sequences involved in a 4D-VAR assimilation.

The step finding the solution for the increment  $\delta\mathbf{X}$  is not isolated and some other jobs have to be performed. If  $t_0$  is the instant of the analysis and for a 4DVAR assimilation  $Dt$  is the timespan, the different steps are generally (according to the current operational 4D-VAR assimilation):

- Step 0a) Provides a first guess  $\mathbf{X}_g$ : this is done by a 6 hours forecast (configuration 1) in resolution  $N_{\text{sfc}}$ ; saves  $\mathbf{X}_g(t_0 - 0.5Dt)$  [compulsory].
- Step 0b) Observation pre-processing.
- Step 0c) Performs a preliminary screening of observations (configuration 1 =  $Dt$ -range forecast from the guess  $\mathbf{X}_g(t_0 - 0.5Dt)$ ) [compulsory].
- Step 1a) Converts the first guess  $\mathbf{X}_g(t_0 - 0.5Dt)$  into a lower resolution  $N_{\text{sm1}}$  (use FULL-POS) [generally done for a 4D-VAR assimilation].  $N_{\text{sm1}} \leq N_{\text{sfc}}$ .
- Step 1b) Do the assimilation for upper air data and surface pressure at resolution  $N_{\text{sm1}}$ ; that provides an increment  $[\delta\mathbf{X}]_{\text{m1}}$  valid for instant  $t_0 - 0.5Dt$ .
- Step 1c) Converts the increment  $[\delta\mathbf{X}]_{\text{m1}}$  at the resolution  $N_{\text{sfc}}$  (use two jobs FULL-POS, one for  $\mathbf{X}$ , one for  $\mathbf{X}_g(t_0 - 0.5Dt)$ ) [generally done for a 4D-VAR assimilation].
- Step 1d) Adds the increment  $[\delta\mathbf{X}]_{\text{m1}}$  converted at resolution  $N_{\text{sfc}}$  to  $\mathbf{X}_g(t_0 - 0.5Dt)$  and performs a quality control [compulsory].
- Step 2a) Converts the first guess  $\mathbf{X}_g(t_0 - 0.5Dt)$  into a lower resolution  $N_{\text{sm2}}$  (use FULL-POS) [generally done for a 4D-VAR assimilation].  $N_{\text{sm1}} \leq N_{\text{sm2}} \leq N_{\text{sfc}}$
- Step 2b) Do the assimilation for upper air data and surface pressure at resolution  $N_{\text{sm2}}$ ; that provides an increment  $[\delta\mathbf{X}]_{\text{m2}}$  valid for instant  $t_0 - 0.5Dt$ .
- Step 2c) Converts the increment  $[\delta\mathbf{X}]_{\text{m2}}$  at the resolution  $N_{\text{sfc}}$  (use two jobs FULL-POS, one for  $\mathbf{X}$ , one for  $\mathbf{X}_g(t_0 - 0.5Dt)$ ) [generally done for a 4D-VAR assimilation].
- Step 2d) Adds the increment  $[\delta\mathbf{X}]_{\text{m2}}$  converted at resolution  $N_{\text{sfc}}$  to  $\mathbf{X}_g(t_0 - 0.5Dt)$  and performs a quality control [compulsory].
- Step 3) Do the assimilation for surface data at resolution  $N_{\text{sfc}}$  [compulsory].

Additional remarks:

- The increment  $[\delta\mathbf{X}]_{\text{m1}}$  found at the end of “Step 1b” is used as a “first guess” to start the minimization of “Step 2b”. The conversion of this increment from the resolution  $N_{\text{sm1}}$  to the resolution  $N_{\text{sm2}}$  is done by simply adding zeros in the spectrum (case where both minimizations are done with an unstretched and untilted geometry).
- There are two minimizations in our example but of course one can do experiments with more (or less) minimizations.
- The time-span remains constant in our example but it can be different for steps 1..) and 2..). In case of different time-spans (one speaks about “quasi-continuous” assimilation in this case) it has been shown (see for ex. (Veersé, 1997)) that it is desirable to have a larger timespan in the second minimization than in the first one.
- Of course the steps 1a, 1c, 1d, 2a, 2c, 2d disappear if the minimizations are done in the same geometry and the same resolution as the forecast.
- An initialisation can be done after step 3) if not already done via a  $J_c$  term in steps 1b and 2b.

Screening is described in section 12.

## 14.7 ENVAR assimilation.

\* **Algorithm:** The 4DEnVar formulation is an alternative to the 4D-Var counterpart, although the two formulations share many similarities. The 4DEnVar cost function is given by:

$$J(\underline{\mathbf{X}}) = J_b(\underline{\mathbf{X}}) + J_o(\underline{\mathbf{X}}) + J_c(\underline{\mathbf{X}})$$

$$J_b(\underline{\mathbf{X}}) = 0.5(\delta\underline{\mathbf{X}})^T \underline{\mathbf{B}}^{-1}(\delta\underline{\mathbf{X}})$$

$$J_o(\underline{\mathbf{X}}) = \sum_{n_{\text{step}}=0}^{n_{\text{step}}=N_{\text{stop}}} 0.5(\mathbf{H}_L[\delta\underline{\mathbf{X}}]_{n_{\text{step}}} - (\mathbf{Y}_{n_{\text{step}}} - \mathbf{H}[\underline{\mathbf{X}}_g]_{n_{\text{step}}}))^T [\mathbf{R}^{-1}]_{n_{\text{step}}} (\mathbf{H}_L[\delta\underline{\mathbf{X}}]_{n_{\text{step}}} - (\mathbf{Y}_{n_{\text{step}}} - \mathbf{H}[\underline{\mathbf{X}}_g]_{n_{\text{step}}}))$$

where the underline notation denotes the 4D nature of variables or matrices.

The main differences compared to the 4DVAR are:

- a different way to compute the background error covariance matrix: this matrix is now a 4D-one (this is why it is denoted by  $\underline{\mathbf{B}}$ ) and has a temporal evolution.
- the use of an ensemble of 4D perturbations, to compute  $\underline{\mathbf{B}}$ .
- the unknown is now a 4D-one:  $\delta\underline{\mathbf{X}}$  (which is the vector containing the  $\delta\mathbf{X}$  at all instants) and no longer  $[\delta\mathbf{X}]_0$ .
- no tangent linear nor adjoint model integrations are required to compute  $J_o$ .

An ensemble prediction is done, with  $L_e$  members. At timestep  $n_{\text{step}}$ , the ensemble prediction provides members  $([\mathbf{X}_{l=1}]_{n_{\text{step}}}, \dots, [\mathbf{X}_{l=L_e}]_{n_{\text{step}}})$ . We denote by  $\underline{\mathcal{X}}$  the following matrix:

$$\left( \begin{array}{cccc} \frac{1}{\sqrt{L_e-1}}([\mathbf{X}_{l=1}]_0 - [\bar{\mathbf{X}}]_0) & \dots & \frac{1}{\sqrt{L_e-1}}([\mathbf{X}_l]_0 - [\bar{\mathbf{X}}]_0) & \dots & \frac{1}{\sqrt{L_e-1}}([\mathbf{X}_{l=L_e}]_0 - [\bar{\mathbf{X}}]_0) \\ \dots & \dots & \dots & \dots & \dots \\ \frac{1}{\sqrt{L_e-1}}([\mathbf{X}_{l=1}]_{n_{\text{step}}} - [\bar{\mathbf{X}}]_{n_{\text{step}}}) & \dots & \frac{1}{\sqrt{L_e-1}}([\mathbf{X}_l]_{n_{\text{step}}} - [\bar{\mathbf{X}}]_{n_{\text{step}}}) & \dots & \frac{1}{\sqrt{L_e-1}}([\mathbf{X}_{l=L_e}]_{n_{\text{step}}} - [\bar{\mathbf{X}}]_{n_{\text{step}}}) \\ \dots & \dots & \dots & \dots & \dots \\ \frac{1}{\sqrt{L_e-1}}([\mathbf{X}_{l=1}]_{N_{\text{stop}}} - [\bar{\mathbf{X}}]_{N_{\text{stop}}}) & \dots & \frac{1}{\sqrt{L_e-1}}([\mathbf{X}_l]_{N_{\text{stop}}} - [\bar{\mathbf{X}}]_{N_{\text{stop}}}) & \dots & \frac{1}{\sqrt{L_e-1}}([\mathbf{X}_{l=L_e}]_{N_{\text{stop}}} - [\bar{\mathbf{X}}]_{N_{\text{stop}}}) \end{array} \right)$$

Due to the limited size of the ensemble, a filtered version of  $\underline{\mathbf{B}}$  is defined by:

$$\underline{\mathbf{B}} = \underline{\mathcal{X}} \underline{\mathcal{X}}^T \circ \underline{\mathbf{L}}$$

where:

- $\underline{\mathbf{L}}$  is a valid correlation matrix with the same size as  $\underline{\mathcal{X}} \underline{\mathcal{X}}^T$ .
- product “o” denotes the element-by-element product of the two matrices, also called the Hadamard-Schur product.

Different implementations of 4D-EnVar are feasible, and a 3D version of EnVar is also possible by omitting the time dimension in the formulation of the cost function.

\* **Main features of the code:** they differ from the 4D-VAR assimilation scheme ones about the following points:

- different pieces of code to compute matrix  $\underline{\mathbf{B}}$ .
- no tangent linear nor adjoint model is called.

Other features are the same as for the 4D-VAR assimilation scheme.

## 14.8 Documentation.

A lot of publications and other papers have been written about optimal interpolation and variational techniques, and variational techniques have been a major topics in the 1990's. The reader will find some references to these publications in the following documentations which will be listed now:

- For optimal interpolation (CANARI), the last available documentation is (IDCA), it is mainly a technical documentation with details about the organigramme of the code and a users'guide (examples of namelists are provided). (Giard and Bazile, 2000) give information about the assimilation of soil variables in ARPEGE (METEO-FRANCE operational version at the period where the paper were written).
- For variational techniques, (TDECAS) gives a good overview of 3D-VAR and 4D-VAR algorithms; see in particular chapters 1 to 3 for 3D-VAR and 4D-VAR formulations, chapter 4 for  $J_b$  calculation, chapter 7 for  $J_c$  calculation, chapters 9 and 10 for the algorithms of surface and soil variables analysis used at ECMWF.

On METEO-FRANCE side, a more oriented ALADIN documentation (IDVAR) has been written (but the major parts of this documentation are also valid for ARPEGE), covering different topics such as screening, 3D-VAR and 4D-VAR algorithms; some code organigrammes are provided.

Documentation (IDS4DV) is a users guide to launch 4D-VAR experiments; most parts of this documentation will become obsolete because the new software OLIVE can be now used to launch 4D-VAR experiments; chapter 3 of this documentation remains interesting and focuses in a synthetic way on some algorithmic aspects (4D-VAR and 3D-VAR FGAT algorithms, trajectory updating,  $J_c$  constraint, surface analysis).

The thesis manuscript (Veersé, 1997) gives theoretical aspects about the 4D-VAR algorithm, and some results with different type of incremental, multi-incremental and quasi-continuous strategies.

Part 4 of (Seity et. al., 2011) gives information about assimilation in AROME.

- Desroziers et al. (2014) gives a description of the EnVar algorithm.

## 15 Configuration 801.

\* **Purpose of this configuration and basics about the architecture of the code:** This configuration is used to do sensitivity studies and can help to answer to 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)? What are the basic variables which have the most impacts on this cost function?

At least the adjoint code is used in this configuration, and some particular options of this configuration also use the tangent linear code. The user has to provide an initial situation, and a reference “final” state which is generally an analysis or another forecast.

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). If the forecast is assumed to have  $N_{\text{stop}}$  timesteps, a direct integration (to store the trajectory) then an adjoint integration of  $N_{\text{stop}}$  timesteps are performed.

There are three options to define the cost function according to variable **NJROPT** (namelist **NAMSENS**), the first one being the most commonly used: in this case the cost function is a quadratic function of different prognostic variables defining an energy, and some components (for example the content of humidity) can be zeroed. The cost function can be defined globally or locally (in the latter case it is zeroed out of a domain defined by two longitudes, two latitudes and two layers).

The organigramme looks like:

```
CNT0 ->
* SUOYOMA
* SUOYOMB
* CGR1 ->
  - file reading
  - CNT2 -> CNT3 -> CNT4
```

Configuration 801 can be used in ARPEGE/IFS and LAM models for the hydrostatic primitive equations model. It is not yet implemented in the non-hydrostatic model.

\* **Documentation:** A 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 45 and cycle AL45. An additional documentation (ID801) has been provided recently. A brief description of some aspects of this configuration (sensitivity studies) has been provided in some papers or internal documentations like (NTA28), (Rabier et al., 1996) or (Soci et al, 2003).

## 16 Ensemble prediction and ensemble assimilation.

### 16.1 Definition and purpose of ensemble prediction.

Probabilistic prediction, in the form of ensemble forecasting, has now become an important component of numerical weather prediction.

When performing only one forecast, no information is obtained about the forecast uncertainty. It is important to know, especially for dangerous phenomena (for example heavy rain, thunderstorms, storms) what is the uncertainty, and which what percentage of risk a dangerous phenomenon may occur.

For that purpose we do not perform one forecast, but a set of several “slightly different” forecasts called “ensemble prediction”. Ensemble prediction consists of performing a number of numerical forecasts in parallel, the dispersion of the forecasts being taken as an estimate of the uncertainty of the future state of the atmosphere. An ensemble prediction performs a set of  $2N_{\text{fc}} + 1$  forecasts (we say that the ensemble prediction has  $2N_{\text{fc}} + 1$  members), the first one being the reference one and the following ones being forecasts with a perturbed initial state, and sometimes a perturbed model. For example, if only the initial state is perturbed, there are  $N_{\text{fc}}$  couples of forecasts with an initial perturbation and the perturbation of the opposite sign.

The choice of initial perturbations must maximize the final perturbations in order the set of forecasts to be able to show all the possible alternative scenarios. For example initial perturbations can be computed via singular vectors (see section 17)

The different scenarios given by the ensemble prediction may give very similar forecasts at some locations (small dispersion) and very different forecasts at other locations (large dispersion). Locations with large (resp. small) dispersion indicate a poor (resp. good) predictability at these locations.

More and more weather forecast services now use ensemble predictions. In Europe, this is the case at METEO-FRANCE, at ECMWF, and in several ALADIN or HIRLAM partners weather forecast services.

Several major meteorological centres, such as the European Centre for Medium-Range Weather Forecasts (ECMWF), National Centers for Environmental Prediction and Canadian Meteorological Center, have been running operational ensemble forecasting systems for a number of years. These systems focus on forecast lead times between 3 and 10 days and are mainly designed to represent uncertainties related to synoptic and large scales.

More recently, ensemble forecasting systems have been designed to estimate uncertainty for shorter lead times and length-scales. These systems use limited-area models with higher resolution than typically used for medium-range but covering only a limited part of the globe. The Meteorological Office, Spanish Met Service, Norwegian Meteorological Institute, Consortium for Small-scale Modelling Limited-Area Ensemble Prediction System and the Limited Area Model for Central Europe consortium have developed such systems.

## 16.2 Ensemble assimilation.

Uncertainties in the initial state (and in the associated data assimilation cycling) can be simulated by an ensemble data assimilation system. The analysis of each member uses perturbed inputs (namely, perturbed observations, and a perturbed background) in order to simulate associated errors.

Observation perturbations are produced from random draws of specified observation error covariances. Background perturbations result partly from the forecast evolution of previous analysis perturbations, and this is complemented by a multiplicative inflation of perturbations in order to represent model error contributions.

Such an ensemble assimilation system can be used to estimate flow-dependent background error covariances for the analysis, but also to provide initial perturbations to an ensemble prediction system (in addition to possible perturbations derived from singular vectors).

## 16.3 Example: the operational ensemble prediction and assimilation at METEO-FRANCE (PEARP, AEARP, PEARO).

PEARP (acronym of Prévision d'Ensemble ARPEGE) is the global short-range ensemble prediction system of METEO-FRANCE, performing forecasts up to 4.5 days. It uses the operational model ARPEGE and benefits from the variable horizontal resolution to reach mesh-sizes around 10 km over France and so pick up mesoscale systems. The vertical resolution is essentially the same as for the deterministic version, in particular in the boundary layer. The ensemble size is 35 members, including a control member, which is a lower resolution version of the operational ARPEGE forecast, and 34 perturbed members obtained from 17 perturbations symmetrized and centred around the control member.

AEARP (acronym of Assimilation d'Ensemble ARPEGE) is the global ensemble assimilation system at METEO-FRANCE. The current AEARP ensemble size corresponds to 25 members, and the 4D-Var analysis of each member uses perturbed inputs (namely, perturbed observations, and a perturbed background) in order to simulate associated errors.

Perturbations to the initial conditions are computed by combining the ensemble data assimilation system AEARP with singular vectors, computed by the configuration 601 over 10 geographic areas on which resolution, norm and optimization times are consistent with the local synoptic scale dynamics.

Model uncertainties are represented through a “multiphysics” approach with ten different physical parametrization sets, including the ARPEGE deterministic physical one. The others packages rely on alternative formulations for the parametrization of vertical diffusion, shallow and deep convection or oceanic fluxes.

PEARO (acronym of Prévision d'Ensemble AROME) is the AROME short-range ensemble prediction system of METEO-FRANCE, currently in test (expected to be put in operations in 2017).

## 16.4 Documentation about ensemble prediction and assimilation.

The ensemble prediction done at METEO-FRANCE is described in (Descamps et al., 2015). The ensemble prediction done at ECMWF is described in (TDECEPS). Ensemble assimilation concept is described in (Berre et al., 2007).

## 17 Singular vectors calculation and configuration 601.

### 17.1 Introduction.

A problem which has some applications in meteorology is to find an initial perturbation (or a set of initial perturbations) which give the biggest impacts on the forecast. If one assumes a model with  $N_{\text{stop}}$  timesteps, and if one does the assumption of a linear perturbation, one searches for the initial perturbation (or a set of orthogonal initial perturbations) which maximizes the ratio:

$$\frac{\langle [\Delta x]_{N_{\text{stop}}}; [\Delta x]_{N_{\text{stop}}} \rangle}{\langle [\Delta x]_0; [\Delta x]_0 \rangle}$$

where  $[\Delta x]_{N_{\text{stop}}}$  is the evolution by the tangent linear model from the initial perturbation  $[\Delta x]_0$ . The orthogonal initial perturbations which are computed by such a method are called “singular vectors”. The main known application is the ensemble prediction (see section 16). The norm  $\langle ., . \rangle$  is defined with a quadratic function.

The way to do such a computation in the ARPEGE and ALADIN code uses configuration 601. The tangent linear code and the adjoint code are used. The singular vectors used in the ECMWF ensemble prediction are currently provided by configuration 601.

### 17.2 Configuration 601.

**Method:** A quadratic positive definite function is used to define the norm  $\langle ., . \rangle$ , it measures a total energy and it is identical to the function of the option **NJROPT=1** of configuration 801. Like in the configuration 801, this function can be zeroed out of a domain defined by two latitudes, two longitudes and two layers. Some contributions of this functions (like the contribution of humidity) can be unplugged. Maximizing the above ratio is equivalent to find the biggest eigenvalues of the matrix **A** associated with the quadratic function and also the associated eigenvectors. A Lanczos method is used: a small symmetric tridiagonal matrix **T** is computed by an iterative algorithm, the main property of **T** being the convergence of the eigenvalues and the eigenvectors of **T** towards those of **A** when the number of iterations increases. If the size of **A** is in the range of “hundreds of thousands”, the maximum size of **T** is equal to the number of iterations (i.e. no more than several tens). The diagonalization of **T** can be done using the Lanczos method coded in the minimizer **CONGRAD**.

**Basics about the code architecture:** This algorithm is performed by configuration 601, setting the key **LANCZOS** to **.T.** . This configuration can be currently used for the primitive equations model (ARPEGE and ALADIN). The code for non-hydrostatic equations is not completely provided. The architecture of the code starts by:

```
CNT0 ->
- SUOYOMA
- SUOYOMB
- CUN1 -> CUN2
```

### 17.3 Documentation.

The most recent documentation has been provided by ECMWF but there is also a documentation provided at METEO-FRANCE describing the configuration 601 and valid for cycle 22T2. ECMWF documentation (TDECEPS) describes how the ensemble prevision is organized and gives a review of two algorithms which can be used to compute the singular vectors.

- The one described in 17.2.
- Another one mixing configurations 601 and 131 (Jacobi-Davidson), which has been removed from the code.

In particular one finds documentation about the singular vectors computation in chapter 1.3, a description of the Lanczos algorithm in appendix A1, and a description of the Jacobi-Davidson algorithm in appendix A2. ECMWF documentation does not give any detail about the code organigramme. On METEO-FRANCE side, a documentation about configuration 601 has been provided in the past (ID601), but it has not been recently updated, the last available version is valid for cycle 22T2: this documentation provides more details about the code organigramme and about the LAM model aspects.

## 18 Input surface climatology for models.

It is necessary to provide orography and a significant number of surface fields as model input. The present section deals with surface fields which are only month-depending (ex: vegetation) or constant around the year (ex: orography).

### 18.1 Configuration 923.

\* **Purpose of this configuration, and main features of the organigramme:** This configuration allows to make surface climatological fields in an ARPEGE or ALADIN file, from a database, when the SURFEX surface scheme is not used. These surface fields are divided into several classes, and there are several jobs treating each class. Each job generally reads surface climatological fields in a database, interpolates them on the ARPEGE/ALADIN grid, then writes them on an ARPEGE/ALADIN file. Since the surface orography is the only field to become a spectral one in the ARPEGE/ALADIN resolution, a special treatment is applied to this field in order to reduce the amount of Gibbs waves.

Configuration 923 can be used for ARPEGE in a METEO-FRANCE context (writing on ARPEGE FA-files) but not in a ECMWF context (no version allowing writing data on GRIB files). A LAM version of the code (configuration E923) is also available. Only surface fields can be treated, and also some constants allowing to retrieve the ozone vertical profile, but no other upper-air climatological field can be performed. The aim of this configuration is to provide lower boundary conditions for the model integration and in particular for the physics, not to provide monthly statistics about surface pressure or upper-air temperature for example.

Configuration 923 is made of a succession of eight jobs which treat the following class of fields:

- Job 1: surface orography, and some other fields having a relationship with the surface orography (do not depend on the month).
- Job 2: constant fields without any relationship with the surface orography (do not depend on the month).
- Job 3: monthly fields having a relationship with the surface temperature (depends on the month).
- Job 4: monthly fields having a relationship with the vegetation scheme (depends on the month).
- Job 5: modification of some fields (previously computed in jobs 2 and 4) on land, with more refined data (these refined data are available on Europe). Depends on the month.
- Job 6: monthly climatological value of surface temperature and surface moisture (depends on the month).
- Job 7: modification of some fields on water surfaces.
- Job 8: three coefficients allowing to retrieve the ozone vertical profile.
- Job 9: aerosols.
- Job 10: climatological fields for an aquaplanet.

The main features of the organigramme are the following ones:

```
CNT0 -> INCLIO ->
- INTERO
- SUALCLIA
- INCLI1
- INCLI2
- INCLI3
- INCLI4
- INCLI5
- INCLI6
- INCLI7
- INCLI8
- INCLI9
- INCLI10
```

INCLI1 to INCLI10 are replaced by EINCLI1 to EINCLI10 in the configuration E923 of ALADIN.

- **CNT0:** controls integration job at level 0.
- **INCLIO:** main routine for configuration 923.
- **INTERO:** determination of the size of interpolation boxes.
- **SUALCLIA:** allocates arrays used in the computation of spectral orography.
- **INCLI1 to INCLI10:** main routines for the jobs number 1 to 10.

Jobs 2 to 10 have the same structure which is generally:

- reading of a database file.

- horizontal interpolations to provide data on the final ARPEGE or ALADIN grid; horizontal interpolations are done by routines specific to configuration 923/E923, but these routines actually look like some interpolation routines used in FULL-POS.
- writing of interpolated fields on an ARPEGE or ALADIN file.

These jobs treat only grid-point fields.

The structure of job 1 (code under **INCL11**) is more complicated because surface orography has to be provided in spectral space, with filtering Gibbs waves. The horizontal interpolation provides a first guess. This guess will be later modified by minimizing a non-quadratic function (which measures the presence of undesired Gibbs waves), the solution having to be Gibbs waves free, spectrally fitted, and at the same time close to the good orography. The minimization follows a quasi-Newton iterative algorithm (use of the minimizer **M1QN3R**).

\* **Documentation:** Configuration 923 is described in chapter 7.2 of the ARPEGE-CLIMAT documentation (IDARPC6). A documentation (ID923E) is available for the ALADIN version E923. The way of computing climatological fields at ECMWF is completely different (configuration 923 is not used, a software called PREPCLIM is used instead). The way of making climatological fields at ECMWF is described in chapter 11 of ECMWF documentation (TDECPHY).

## 18.2 Surface climatology for SURFEX.

\* **Sequence of tasks:** The sequence of tasks is more complicated, and can be summarized as follows:

- Compute a grid-point orography by using the SURFEX code.
- Use the first part of configuration 923 (**INCL11** or **EINCL11**):
  - Input orography is the one provided by the SURFEX code.
  - **(E)INCL11** performs the anti-Gibbs treatment and the spectral fit.
  - The spectrally fitted orography is stored in the output climatologic file.
- The other surface climatologic fields are provided by using the SURFEX code (read on a database, do horizontal interpolations internal to SURFEX).

## 19 The other C9xx configurations.

\* **Configuration 901:** This configuration does a transformation from a GRIB file (format used at ECMWF) towards an ARPEGE file (format used at METEO-FRANCE). It allows the utilisation of analyses performed at ECMWF. The code of this configuration can be found under the sequence **CNT0** – > **CPREP1** and is rather simple. (ID901A) provides a short documentation to use this configuration.

\* **Configuration 903:** This is an alternate way to do off-line FULL-POS. The code of this configuration can be found under the sequence **CNT0** – > **CPREP3**. More details can be found in documentation (IDFPOS). In the future it will replace the old off-line FULL-POS and conf 901 too.

\* **Configuration 911:** This configuration computes a set of matrices allowing a spectral transformation between a stretched geometry of stretching  $c$  and truncation  $N_s$  and a non-stretched geometry of truncation  $N_c$ ,  $N_c$  being slightly greater than  $c * N_s$ . Matrices are written on a LFI file. These matrices are required in some applications where the variable-mesh spherical geometry is used and where some spectral calculations have to be done in a constant-mesh space (this is the case when filtering some fields in FULL-POS). The code of this configuration can be found in library UTILITIES and requires a specific main program **MASTER911** (calling **SUDIL**). A more detailed documentation about this configuration can be found in (IDRD).

\* **Configuration 931:** This configuration is used for climatic applications and reads a SST (sea surface temperature) field on a specific file, and after some horizontal interpolations, writes the interpolated SST on an ARPEGE file (this configuration works like some parts of the configuration 923). The code of this configuration can be found under the sequence **CNT0** – > **CSSTBLD**.

\* **Configuration 932:** This configuration reads the sea-ice concentration field measured by a satellite on a satellite data file, and after some horizontal interpolations, writes the interpolated field on an ARPEGE file (this configuration works like some parts of the configuration 923). The code of this configuration can be found under the sequence **CNT0** – > **CSEAICE**.

## 20 Minimizations.

\* **Introduction:** In different parts of the model there are some big linear systems to solve; the dimension of such systems can reach several hundreds of thousands, and it is out of question to solve them by direct methods. If the matrix  $\mathbf{A}$  is symmetric and positive definite, solving the linear system  $\mathbf{AX} = \mathbf{B}$  is equivalent to find the solution which minimizes the value of the quadratic function  $J(\mathbf{X}) = 0.5\mathbf{X}^T\mathbf{AX} + \mathbf{BX} + J(0)$ . We have also to find the solution minimizing non-quadratic functions, for example in configuration 923. In ARPEGE/IFS, the main applications requiring function minimizations or solving of big linear systems are the following ones:

- Configuration 131 for the 3D-VAR and the 4D-VAR assimilation algorithms (the function is a quadratic one).
- The part of configuration 923 (doing climatological surface fields) computing the spectral orography (the function is not quadratic).

Such problems are solved by iterative algorithms: one computes a guess for  $\mathbf{X}$ , then computes  $J(\mathbf{X})$  and the gradient of  $J(\mathbf{X})$ . A direction and an increment are computed to modify  $\mathbf{X}$ . A new value of  $J(\mathbf{X})$  and of its gradient are computed (this new value of  $J(\mathbf{X})$  is assumed to be lower than the previous one and the gradient is also assumed to decrease). There are several criteria to stop the algorithm, the main one being that the norm of the final gradient must be below a pre-defined ratio of the initial norm. Each minimizer requires a simulator to compute  $J(\mathbf{X})$  and its gradient. The simulator involved in configuration 923 is **SIMREL**; the simulator involved in configurations 131 is **SIM4D**. There are several classes of iterative algorithms, the two ones used in the ARPEGE/ALADIN framework are:

- The quasi-Newton algorithms (can be used for non-quadratic functions).
- The conjugate gradient algorithms (generally used for quadratic functions).

More exactly, there are three minimizers used in ARPEGE/IFS.

### \* The minimizers available in ARPEGE and ALADIN:

- **M1QN3** is the minimizer provided by INRIA (Institut de Recherche en Informatique et en Automatique). It follows a quasi-Newton algorithm and can be used to minimize non-quadratic functions. In ARPEGE/IFS it can be used in all applications requiring a minimization (see above). This is the only minimizer which can be currently used in configuration 923.
- **M1QN3R**: Version of **M1QN3** which does not use the structure "CONTROL VECTOR".
- **N1CG1** follows a Fletcher-Reeves conjugate gradient method. It can be used to minimize symmetric positive definite quadratic functions. It is available in the 3D-VAR and the 4D-VAR assimilation algorithms.
- **CONGRAD** follows a conjugate gradient method combined with a Lanczos algorithm (the Lanczos algorithm allowing to estimate the biggest eigenvalues and the corresponding eigenvectors). It can be used to minimize symmetric positive definite quadratic functions. It is available in the 3D-VAR and the 4D-VAR assimilation algorithms.

For the 3D-VAR and the 4D-VAR assimilation:

- **CONGRAD** is activated setting (**LCONGRAD;LN1CG1**)=(.T.;.F.) in **NAMVAR**.
- **N1CG1** is activated setting (**LCONGRAD;LN1CG1**)=(.F.;.T.) in **NAMVAR**.
- **M1QN3** is activated setting (**LCONGRAD;LN1CG1**)=(.F.;.F.) in **NAMVAR**.

For the minimization of symmetric positive definite quadratic functions, **CONGRAD** is generally the most efficient and **M1QN3** the less efficient. One iteration of each minimizer has a comparable CPU cost. **M1QN3** and **N1CG1** are available in the ALGOR project; **CONGRAD** is available in the ARPIFS project.

\* **Preconditioning:** It is desirable to have a well conditioned matrix  $\mathbf{A}$  (that means that the ratio between the absolute values of the biggest and the lowest eigenvalues must be not too far from 1). In most cases one prefers to solve  $\mathbf{H}^{-1}\mathbf{AX} = \mathbf{H}^{-1}\mathbf{B}$  (left preconditioning) or  $\mathbf{AH}^{-1}[\mathbf{HX}] = \mathbf{B}$  (right preconditioning), where  $\mathbf{H}$  is a matrix easily computable and invertible and where  $\mathbf{H}^{-1}\mathbf{A}$  has a better conditioning than  $\mathbf{A}$ . Left preconditioning can be used in **M1QN3** and **N1CG1** with a quasi-Newton preconditioner. Right preconditioning can be used in **CONGRAD**, the calculation of the preconditioner using an approximation of the first eigenvalues (and the corresponding eigenvectors) of  $\mathbf{A}$  computed by a Lanczos algorithm. With a right preconditioning the unknown becomes  $[\mathbf{HX}]$  instead of  $\mathbf{X}$ . For **M1QN3** and **N1CG1**, preconditioners are computed inside these routines, there are written on a file by routine **SAVHESS** and read in this file by routine **SUHESS**. For **CONGRAD**, the eigenvalues and eigenvectors allowing to compute the preconditioner is done under **CONGRAD**, the remaining part of the calculation of the preconditioner is done by routine **PREPPCM**; the change of variable is done inside routine **PRECOND**.

\* **Documentation and references about minimizers:** Literature provides a large number of books or papers explaining these algorithms, with often different denotations. For example (Bonnans et al., 1997) gives (in French) a good overview of these algorithms, (Shewchuk, 1994) is more oriented towards conjugate gradient methods with good practical examples, (Barrett et al., 1994) provides a list of algorithms and the way to code them. The internal paper (IDNEWT: Veersé, 1996) gives (in French) a synthetic overview of Newton and quasi-Newton algorithms. (IDQN3) gives a description of the minimizer **M1QN3** and of its preconditioner. (IDCGRA) gives a synthetic overview of different existing minimizers and ways of preconditioning. The algorithm used in **CONGRAD** is described in (IDCGRA) and in the part 6 of ECMWF documentation (TDECDAS). Calculation of the preconditioner of **CONGRAD** is described in the part 6.5 of ECMWF documentation (TDECDAS). Documentation (IDMINI) gives an overview of minimization algorithms and the way they are implemented in the ARPEGE/IFS framework; it provides the main features of the organigramme of **M1QN3**, **N1CG1** and **CONGRAD**, and a list of variables involved in the minimizations.

## 21 Observation treatment before assimilation.

### 21.1 List of observations.

The list of available observations can be put into nine classes. In each class, there are several types of messages, for example in class SYNOP for surface observations one can find for example SYNOP for land observations and SHIP for ship observations.

- Conventional observations:
  - SYNOP [SYNOP,SHIP,SYNOR,RADOME,GPS-SOL]: surface observations.
  - AIREP: aircrafts observations.
  - DRIBU [BUOY,DRIFTER,BATHY,TESAC]: drifting buoys observations.
  - TEMP [TEMP,TEMP-SHIP,TEMP-MOBIL,TEMP-DROP]: vertical soundings.
  - PILOT [PILOT,PILOT-SHIP,EUROPROFIL,PROFILER]: vertical wind profile.
  - PAOB: forced pressure observations, for example in hurricanes.
- Satellite observations:
  - SATOB: satellites observations (atmospheric motion vectors).
  - SATEM: satellites observations (radiances).
  - SCATT: scatterometers observations.
  - LIMB: GNSS radio-occultation observation.
- Radar observations (for reflectivities and radial velocities).

### 21.2 Observation treatment.

The organigramme including observation treatment before assimilation and screening inside a model configuration 1 can be written as follows (scheme used at METEO-FRANCE; things can be significantly different at ECMWF):

- a) One starts from BDM (abbreviation from “Banque de Données Météorologiques”) files; they contain observations written in a “BUFR” code.
- b) Application of the procedures **OULAN** and **BDMO\_LECTURE** which read data in the BDM, samples data in space and time, and writes the sampled data in an ASCII file called “OBSOUL” (for **OULAN**). **BDMO\_LECTURE** gets BUFR files from BDM.
- c) Application of the procedure **BATOR** which:
  - reads data in “OBSOUL” and “BUFR” files.
  - may read a complementary blacklist (the main one is read during the screening).
  - writes all these data with some additional information (saying if these observations have to be used or not) in an ODB file.
  - computes some intermediate matrices (observation errors statistics, horizontal and vertical autocorrelations) allowing to compute the matrices **B** and **R** used in the linear system which has to be solved in the assimilation, and writes them in an ODB file.
- d) Screening (containing tasks **MERGE+SCREENING+OBSCCMA**). After the screening, the final ODB files still contain all the “entrance” observations, but with some additional information saying if these observations have to be used or not.
- e) Final selection of surface data by procedure **OBSCCMACAN**.

Remarks:

- Procedure **BATOR** can be found in project ODB.
- Procedures **OBSCCMA** and **OBSCCMACAN** can be found in project ODB.
- At ECMWF there is an important part **OBSPROC** which do tests on observations before the screening, at the place of step c). At METEO-FRANCE the most part of tests are done in the screening.
- The projects involved are ARPIFS, ALADIN for LAM models only, ODB, SATRAD, UTILITIES. Additionally at ECMWF some other projects are involved: OBSTAT.

### 21.3 Documentation.

More details about the list of observations used in the assimilation can be found in the chapter 3 of (IDCA) or in (IDUOBS). All the observation treatment before assimilation done at ECMWF (**OBSPROC**) is described in (TDECOBS25) and parts 2 and 3 (TDECOBS).

## 22 IO and the different types of files used.

\* **List:** When launching the ARPEGE/IFS project in year 1987 it was wished to have only one type of file (contrary to the old system EMERAUDE/PERIDOT which counted until 7 types of files). This target has not been completely fulfilled, and there are still several types of files, especially if one considers all the observation processing. The different types of files are:

- LFI files (LFI stands for “Logiciel de Fichiers Indexés”, i.e. indexed files software): these indexed files are used to store some constants like rotation and dilatation matrices, or the transmission coefficients for the simplified radiation scheme. This files software is used at METEO-FRANCE but not at ECMWF. Routines allowing operations on LFI files can be found in the project IFS AUX.
- FA files (FA stands for “Fichier ARPEGE”): this is an overlayer of the LFI package, but these files have a frame containing some information for example about the resolution of the model, the Gaussian Grid or the list of vertical levels. This type of files is used for initial conditions, lateral boundary conditions (for LAM models), historical files, post-processing files, surface climatology. This files software is used at METEO-FRANCE but not at ECMWF. Routines allowing operations on FA files can be found in the project IFS AUX.
- GRIB files (GRIB stands for “GRIdded Binary”): they are used at ECMWF where the FA software is used at METEO-FRANCE. At METEO-FRANCE the GRIB software is used for some particular applications (for example to store the pre-conditioner used in a minimization, the increment produced by a minimization, or to store some post-processed data). Routines allowing operations on GRIB files can be found in the project IFS AUX. There is an external procedure **PROGRID** which allows to convert FA files into GRIB files. GRIB is an international format defined by WMO.
- ODB files (ODB stands for “Observational Data Base”): these files are used to store observations and pre-processed statistics. Routines allowing operations on ODB files can be found in the project ODB.

Some other types of files can be used in a limited set of applications.

- Input data for configuration 923 are available in ASCII or in IEEE binary format.
- Observations are available first in BDM files (BDM stands for “Banque de Données Météorologiques”). Storage uses the BUFR format. The BUFR format is an international WMO format designed for the meteorological data. Abbreviation BUFR stands for “Binary Universal Form for the Representation of meteorological data”.
- OBSOUL files: between application of the procedures **OULAN** and **BATOR** observations are stored in ASCII files called OBSOUL.
- GSA format is used for some input files of configuration 131 (Stabal).

### \* **Documentation:**

- The LFI software is described in (IDLFI) (recent update of an old documentation (NTA12)).
- The FA software is described in (IDFA) (recent update of an old documentation (NTA17)).
- The ODB software is described in (IDODBO), (IDODBT), (IDODBU), (IDODBZ).
- The BUFR format is described in chapter 7 of ECMWF documentation (TDECOBS25), and also in some notes available on INTERNET, see for example (IDBUFR).
- There is no ARPEGE/IFS specific documentation about the GRIB format but some documentations can be found on INTERNET, see for example (IDGRIB), (IDGRIG) or (IDGRIC).
- The GSA format is described in (IDGSA).
- The list of files read and written by a subset of commonly used ARPEGE/ALADIN configurations is described in documentation (IDIO).
- Some technical aspects of the IO are described in documentation (IDTECIO).

## 23 Distributed memory and OpenMP aspects.

\* **Evolution of computers during the last years:** Since the first computers in the 1950's, the highest power of the available computers has been multiplied by a factor 100 each decade, and this growing rate does not seem to slow yet. The most powerful computer now reaches more than  $10^{15}$  flops, i.e. operations/second. In the beginning of the 1980's vectorial architecture machines appeared, and the possibility of using multitasking shared memory architectures. METEO-FRANCE has used operationally such machines until 1998. During the mid-1990's distributed memory machines appeared (they have been used for operational applications at METEO-FRANCE since 1998); currently new generations of highly massively parallel machines are progressively used in foreign weather forecast centres.

There are two ways to do distribution:

- Message passing (MPI): call to MPI. routines.
- OpenMp: use of directives.

We now try to give a brief description of the "MPI" distributed memory architecture.

\* **Distributed memory features:** In a distributed memory architecture, each processor has its own memory: that means that a quantity which is declared in several different processors corresponds to different parts of memory. The consequence is that some communications between processors will be necessary when:

- different parts of the code have not the same way of sharing data between processors.
- there is a need to collect data or to dispatch data.

In ARPEGE/IFS, there are **NPROC** processors (and additionally **NPROC\_IO** processors of IO server). These **NPROC** processors are organised into two levels of distribution defined by a couple of variables giving the total amount of processors in both levels. Each part of the model has its couple of variables.

	A-level parallelisation	B-level parallelisation
Spectral calculations where all zonal wavenumbers are independent		
- semi-implicit scheme:	<b>NPRTW</b>	<b>NPRTN</b>
- horizontal diffusion:	<b>NPRTW</b>	<b>NPRTV</b>
Legendre transforms:	<b>NPRTW</b>	<b>NPRTV</b>
Fourier space calculations:	<b>NPRTW</b>	<b>NPRTV</b>
Fast Fourier transforms:	<b>NPRTNS</b>	<b>NPRTV</b>
Grid-point calculations:	<b>NPRGPS</b>	<b>NPRGPEW</b>
Reading ARPEGE file:	<b>NSTRIN</b>	1
Writing on ARPEGE file:	<b>NSTROUT</b>	1
IO server:	<b>NPROC_IO</b>	1

For example, for grid-point calculations, **NPROC=NPRGPS\*NPRGPEW**.

Remark: there is only one level of distribution for reading and writing files.

The first level of distribution is done by a horizontal spatial distribution for grid-point calculations, Fourier transforms and Fourier space calculations, and is done by dispatching the zonal wavenumbers for Legendre transforms and spectral calculations.

The second level of distribution is done by a horizontal spatial distribution for grid-point calculations, a vertical distribution for the spectral transforms and the horizontal diffusion.

For reading and writing, the total amount of fields is shared into a set of 2D horizontal fields and the distribution on processors is done on these 2D fields (the same processor completely reads or writes a 2D field).

There is an optimized version of the second level of distribution well suited for spherical geometry with a reduced Gaussian grid (option **LEQ\_REGIONS=T**) which improves the load balance in the grid-point calculations in this case.

Communication between processors (also called "message passing") is done via the software **MPL** (available in the library IFSAUX): the routines have a name starting by **MPL**. An additional set of "overlayer" routines have been provided in the projects ARPIFS and ALADIN (generally in the directory "parallel") which call **MPL** routines: these routines do transpositions or some other communications between processors for particular applications.

The consequence of the distributed memory architecture is that for a same quantity there can be two different variables in the code: one which is global to all processors (its name generally ends by "G"), and one which is local to each processor (it can take a different value in each processor, its name often ends by "L"). For example the "global" total number of grid-points on a layer is **NGPTOTG**, but each processor acts on **NGPTOT** grid-points in the grid-point calculations.

Most variables having a relationship with the distributed memory architecture are in modules **YOMMP0** and **YOMMP**, some of them are in namelists **NAMPAR0** and **NAMPAR1**.

\* **Grid-point space:** For a 2D field calculations are done on **NGPTOTG** grid-points. Each processor treats its own version of **NGPTOT** grid-points. The **NGPTOT** grid-points are subdivided into **NGPBLKS** packets of **NPROMA** points. **NPROMA** is a tunable variable (in **NAMDIM**) and its optimal value depends on machine architecture. A processor treats a whole column of **NFLEVG** model levels.

\* **Splitting distributed memory features between message passing and OpenMp:** On recent machines there are **NNODE** nodes, each node has several processors. Assuming a job using **NPROC+NPROC\_IO** processors:

- If possible, **NPROC+NPROC\_IO** must be a multiple of **NNODE**.
- Distribution per node is  $(\text{NPROC+NPROC\_IO})/\text{NNODE}$  processors for MPI distribution and **NTHREADS** for OpenMp distribution.
- Product  $(\text{NPROC+NPROC\_IO})/\text{NNODE} \times \text{NTHREADS}$  must be consistent with machine architecture to optimise performances.

\* **Documentation about distributed memory:** The distributed memory architecture is described in the ECMWF documentation: part 3 of (TDECTEC25), part 2 of (TDECTEC). On METEO-FRANCE side a documentation (IDDM) has been written as a complement to the ECMWF documentation and gives an as comprehensive as possible list of **MPL** and other processors communication routines which are available in the code; some distributed memory aspects are dispatched in other documentations. A brief documentation (IDIDM) gives basics about distributed memory architecture, it is presented like slides. Documentation (IDEQR) provides a description of option **LEQ\_REGIONS=T**.

About OpenMp, some documentation can be found on INTERNET, see for example (IDOMP1) and (IDOMP2).

## 24 Launching experiments.

\* **Introduction:** At METEO-FRANCE, experiments may be launched via tools like OLIVE, VORTEX or MITRAILLETTE. ECMWF on its side used in the past softwares such **PREP\_AN** for assimilations and **PREP\_IFS** for forecasts (and a lot of scripts used by ECMWF are stored on the project “SCRIPTS”). Changes are in progress and ECMWF currently migrates towards **ecFlow**.

\* **Structure of a script:** It may change from a configuration to another one, but one generally retrieves at least a subset of the following structure:

- a header.
- a part defining some environment variables.
- reading of a namelist or sometimes several namelists.
- reading of an initial conditions file (name **ICHSH[code]INIT**); some configurations can require climatologies (**FULL-POS**), observation files (screening, configuration 131), rotation or dilatation matrices, constants for radiation scheme, etc.
- copy of an executable on the local file **MASTER**.
- launching of the local executable **MASTER**.
- saving of the output listings (“lola” and some files, the name of which starts by “NODE”; for example, the file containing the output coming from the main processor has the name **NODE\_001\_01**).
- writing of the output files (historic files, **FULL-POS** files for example).
- a trailer.

For a subset of usual configurations, documentation (IDIO) describes, for each configuration, the list of files required as input (with the local name) and the list of files produced as output (with the local name).

The following example is given for a 6-hour forecast producing a 6-hour range historic file (no other files are assumed to be produced); if the experiment name is **CNMEXP=ARPE**, the model reads a file, the local name of it being **ICMSHARPEINIT**, and produces an historic file at range 6h, the local name of it being **ICMSHARPE+0006**.

\* **Namelist.** In order to avoid to have to recompile all the code when changing the value of a variable (for example a dimension), as it was the case in the old **PERIDOT/EMERAUDE** system, some variables are provided in different namelists, so it is possible to change the value in the namelist without recompiling the code. All variables which have to be modified compared to the default value can be specified in the namelist, but there is also an option where the model retrieves some data (date and model geometry) by reading the frame of some input files. An example of namelist for the **ARPEGE** operational forecast is provided in part 19of documentation (IDSL).

\* **List of machines:** ARPEGE/IFS, AROME and ALADIN have been designed to run on very powerful machines, but for applications which do not require too many resources, the use of one or a set of workstations (with a good compiler) is possible.

\* **OLIVE:** This software has been developed to reproduce easily the operational (or parallel) suite and can be used for some applications close to the operational ones. For example, the following applications can be launched: AROME 3dvar analysis, AROME forecast, ARPEGE 4dvar analysis, ARPEGE forecast. Connect to OLIVE to see the extensive list of possible configurations.

The use of OLIVE can be summarized as follows:

- connect on server  
[http://sxalgo1.cnrm.meteo.fr:8181/swapp\\_entry/groucho/Swapp/Browse/etc/apps/Swapp/](http://sxalgo1.cnrm.meteo.fr:8181/swapp_entry/groucho/Swapp/Browse/etc/apps/Swapp/)  
from a local workstation; provide a users name and a password.
- create an experiment, define its features, and send the scripts on a front-desk machine (that can be for example “sxalgo1”).
- connect on its own LINUX machine and launch “xcdp”: a window opens which shows the sequence of tasks of the job (after filling a request of user and password); the right button of the mouse allows to launch this job or a subset of this job on the computer, via a menu.

An in-line printable documentation (IDOLIV) is available when connecting on server:

[http://sxalgo1.cnrm.meteo.fr:8181/swapp\\_entry/groucho/Swapp/Browse/etc/apps/Swapp/](http://sxalgo1.cnrm.meteo.fr:8181/swapp_entry/groucho/Swapp/Browse/etc/apps/Swapp/)

\* **VORTEX:** A successor (VORTEX) is currently under development and is already used for a subset of operational applications. Some documentation about VORTEX (IDVORTEX) is available at:

<http://sicompas/vortex/doku.php/start>

One difference with OLIVE is that scripts use PYTHON language.

\* **MITRAILLETTE:** For code validation, there is a script called MITRAILLETTE which uses standard scripts and namelists, and provides a set of scripts for validation. This software is used to validate simple configurations which use a limited amount of input files. The environnement files are stored on GIT library (project MITRAILLE). For more details see documentations (IDMITR) and (IDMITS).

The current limitations are: use of scripts written in shell; no PYTHON scripts; no configuration using observations.

## 25 References and documentation.

### 25.1 Introduction.

This chapter presents a list of internal notes and publications describing the code of ARPEGE/IFS (cycle 45) and ALADIN (cycle AL45). A special section will provide, for each topic, the list of documentations where some explanation will be found when existing. Some rules have been used when listing references:

- for internal notes describing one particular cycle of ARPEGE/IFS, only the most recent release available is cited. Some too old documentations which have become obsolete have not been cited.
- for ARPEGE notes, all the 33 notes have been referenced; some of them are now obsolete or do not deal directly with ARPEGE.
- for bibliography only the main papers describing one part of ARPEGE/IFS have been referenced; a list of references has been added for the physical parameterizations.

Remarks:

- Most of these documentations are available on the internet server:  
<http://www.umr-cnrm.fr/gmapdoc/>
- A list of documentations written between 1992 and 1995 (and not updated since this date) can be found on the internet server:  
<http://gco.meteo.fr/ifsdoc/html/>  
topics “VPP5000 Online Manual”, then “Parent Directory”, then “IFS/ARPEGE Documentation”; some parts of these documentations (mainly the algorithmic aspects) can still be used. Only a small subset of these documentations has been referenced below.

### 25.2 Bibliography.

#### 25.2.1 Publications relative to the physical parameterizations.

- Arakawa, A., and W.H. Schubert, 1974: Interaction of a cumulus cloud ensemble with the large-scale environment. *Journal of Atmospheric Sciences*, **31**, 674-701.
- Baines, P.G., and T.N. Palmer, 1990: Rationale for a new physically based parameterization of subgrid-scale orographic effects. *Technical Memorandum 2, ECMWF*.
- Bechtold, P., C. Fravallo, and J.P. Pinty, 1992: A model of marine boundary-layer cloudiness for mesoscale application. *Journal of Atmospheric Sciences*, **49**, 1723-1744.
- Bechtold, P., E. Bazile, F. Guichard, P. Mascart, and E. Richard, 2001: A mass-flux convection scheme for regional and global models. *Quart. J. Roy. Meteor. Soc.*, **127**, 869-886.
- Boer, G.J., N.A. Mc Farlane, R. Laprise, J.D. Henderson, and J.-P. Blanchet, 1984: The Canadian Climate Centre spectral atmospheric general circulation model. *Atmosphere-Ocean*, vol **22/4**, 397-429.
- Bougeault, Ph., 1985: A simple parameterization of the large-scale effects of deep cumulus convection. *Mon. Wea. Rev.*, **113**, 2108-2121.
- Caniaux, G., J.-L. Redelsperger, and J.P. Lafore, 1994: A numerical study of the stratiform region of a fast-moving squall line. Part I. General description, and water and heat budgets. *Journal of Atmospheric Sciences*, **51**, 2046-2074.
- Clough, S.A., and R.A.A. Franks, 1991: The evaporation of frontal and other stratiform precipitation. *Quart. J. Roy. Meteor. Soc.*, **111**, 1057-1080.
- Cuxart, J., P. Bougeault, and J.L. Redelsperger, 2000: A turbulence scheme allowing for mesoscale and large-eddy simulations. *Quart. J. Roy. Meteor. Soc.*, **126**, 1-30.
- Dandin, Ph., and J.J. Morcrette, 1996: The ECMWF FMR scheme in the METEO-FRANCE climate model ARPEGE-CLIMAT. *Note de Centre CNRM no 50*, 74pp.
- Ducrocq, V., and Ph. Bougeault, 1995: Simulations of an observed squall line with a meso-beta-scale hydrostatic model. *Weather and Forecasting*, vol **10/2**, 380-399.
- Fouquart, Y., and B. Bonnel, 1980: Computations of solar heating of the earth’s atmosphere: a new parameterization. *Beitr. Phys. Atmos.*, **53**, 35-62.
- Geleyn, J.-F., and A. Hollingsworth, 1979: An economical analytical method for the computation of the interaction between scattering and line absorption of radiation. *Beitr. Phys. Atmos.*, **52**, 1-16.
- Geleyn, J.-F., C. Girard and J.-F. Louis, 1982: A simple parameterization of moist convection for large-scale atmospheric models. *Beitr. Phys. Atmos.*, **55**, 325-334.
- Geleyn, J.-F., 1986: Use of a modified Richardson number for parameterizing the effect of shallow convection. *Journal of the Meteorological Society of Japan*, WMO/IUGG NWP Symposium special issue (August 1986), 141-149.

- Gérard, L., 2001: Physical parameterisations for a high resolution operational Numerical Weather Prediction Model. *Université Libre de Bruxelles, Faculté des Sciences Appliquées, Aug 2001*.
- Girard, C., and Y. Delage, 1990: Stable schemes for nonlinear vertical diffusion in atmospheric circulation models. *Mon. Wea. Rev.*, **118**, 137-146.
- Gregory, D., R. Kershaw and P. Inness, 1997: Parameterization of momentum transport by convection. II: Tests in single-column and general circulation models. *Quart. J. Roy. Meteor. Soc.*, **123**, 1153-1183.
- Kessler, E., 1969: On the distribution and continuity of water substance in atmospheric circulations. *Meteorological Monographs (published by the American Meteorological Society)*, vol **10** nr **32**, 84pp.
- Lopez, Ph., 2002: Implementation and validation of a new prognostic large-scale cloud and precipitation scheme for climate and data assimilation purposes. *Quart. J. Roy. Meteor. Soc.*, **128**, 229-257.
- Louis, J.-F., 1979: A parameteric model of vertical eddy fluxes in the atmosphere. *Bound.-Lay. Meteorol.*, **17**, 187-202.
- Louis, J.-F., M. Tiedtke, and J.-F. Geleyn, 1981: A short history of the operational PBL-parameterization of ECMWF. *Workshop on Planetary Boundary Layer Parameterization*, ECMWF, 25-27 November 1981, 59-79.
- Mlawer, E.J. et al., 1997: Radiative transfer for inhomogeneous atmospheres: RRTM, a validated correlated-k model for the longwave. *J.G.R.*, **102D**, 16663-16682.
- Morcrette, J.-J., L. Smith, and Y. Fouquart, 1986: Pressure and temperature dependence of the absorption in longwave radiation parameterization. *Beitr. Phys. Atmos.*, **59**, 455-469.
- Morcrette, J.-J., 1989: Description of the radiation scheme in the ECMWF model. *ECMWF Technical Memorandum*, **165**.
- Noilhan, J., and S. Planton, 1989: A simple parameterization of land surface processes for meteorological models. *Mon. Wea. Rev.*, **117**, 536-549.
- Peltier, W.R., and T.L. Clark, 1987: Non linear mountain waves and wave-mean flow interaction: elements of a drag parameterization. *Proceedings of ECMWF Seminar/Workshop 15-20 Sept 1986*, vol **1**, 223-249.
- Pergaud, J., V. Masson, and S. Malardel, 2009: A parameterization of dry thermals and shallow cumuli for mesoscale numerical weather prediction. *Boundary-Layer Meteorol.*, **132**, 83-106.
- Phillips, D.S., 1984: Analytical surface pressure and drag for linear hydrostatic flow over three-dimensional elliptical mountains. *J. Atmos. Sci.*, vol **41**/6, 1073-1084.
- Pinty, J.-P., and P. Jabouille, 1998: A mixed-phased cloud parameterization for use in a mesoscale non-hydrostatic model: simulations of a squall line and of orographic precipitation. *Proceeding, Conference on Cloud Physics, Everett, WA, Amer. Meteor. Soc.*, 217-220.
- Ritter, B., and J.-F. Geleyn, 1992: A comprehensive radiation scheme for numerical weather prediction models with potential applications in climate simulations. *Mon. Wea. Rev.*, **120**, 303-325.
- Schneider, E.K., and R.S. Lindzen, 1976: A discussion of the parameterisation of momentum exchange of cumulus convection. *J.G.R.*, **81**, 3158-3160.
- Smith, R. N. B., 1990: A scheme for predicting layer clouds and their water content in a general circulation model. *Quart. J. Roy. Meteor. Soc.*, **116**, 435-460.

## 25.2.2 Other publications.

This list is not complete; some other references can be found in the bibliography of the following publications or internal notes.

- Arakawa, A., and C. S. Konor, 2009: Unification of the anelastic and quasi-hydrostatic systems of equations. *Mon. Wea. Rev.*, **137**, 710-726.
- Barrett R., M. Berry, T. F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine and H. Van der Vorst, 1994: Templates for the solution of linear systems: building blocks for iterative methods, 2nd edition, Pub. SIAM, Philadelphia. Available on <http://www.netlib.org/templates/Templates.html>
- Berre L., O. Pannekoucke, G. Desroziers, S. E. Ștefănescu, B. Chapnik, and L. Raynaud, 2007: A variational assimilation ensemble and the spatial filtering of its error covariances: increase of sample size by local spatial averaging. *ECMWF Seminar Proceedings 11-13 June 2007 (Workshop on Flow-dependent aspects of data assimilation)*, 151-168.
- Bénard, P., J. Vivoda, J. Mašek, P. Smolíková, K. Yessad, C. Smith, R. Brozkova, and J.-F. Geleyn, 2010: Dynamical kernel of the ALADIN-NH spectral limited-area model: revised formulation and sensitivity experiments. *Quart. J. Roy. Meteor. Soc.*, **136**, 155-169.
- Bonnans, J.F., J.C. Gilbert, C. Lemaréchal and C. Sagastizabal, 1997: Optimisation numérique, aspect théorique et pratique. (Collection Mathématiques et Applications, édité chez Springer).

- Brousseau, P., Y. Seity, D. Ricard and J. Léger, 2016: Improvement of the forecast of convective activity from the AROME-France system. *Quart. J. Roy. Meteor. Soc.*, **142**, 2231-2243.
- Bubnová, R., G. Hello, P. Bénard and J.F. Geleyn, 1995: Integration of the fully elastic equations cast in the hydrostatic-pressure terrain-following coordinate in the framework of the ARPEGE/ALADIN NWP system. *Mon. Wea. Rev.*, **123**, 515-535.
- Courtier, Ph., C. Freyrier, 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.
- Courtier, Ph., and J.F. Geleyn, 1988: A global numerical weather prediction model with variable resolution: Application to the shallow-water equations. *Quart. J. Roy. Meteor. Soc.*, **114**, 1321-1346.
- Davies, H.C., 1976: A lateral boundary formulation for multi-level prediction models. *Quart. J. Roy. Meteor. Soc.*, **102**, 405-418.
- Descamps, L., C. Labadie, A. Joly, E. Bazile, P. Arbogast and P. Cébron, 2015: PEARP, the Météo-France short-range ensemble prediction system. *Quart. J. Roy. Meteor. Soc.*, **141**, 1671-1685.
- Desroziers, G., J.T. Camino, and L. Berre, 2014: 4DnVar: link with 4D state formulation of variational assimilation and different possible implementations. *Quart. J. Roy. Meteor. Soc.*, **140**, 2097-2110.
- Fischer, C., T. Montmerle, L. Berre, L. Auger, S. E. Ștefănescu, 2005: An overview of the variational assimilation in the ALADIN/France numerical weather prediction system. *Q. J. R. Meteorol. Soc.*, **131** (issue 613), 3477-3492.
- Giard, D., and E. Bazile, 2000: Implementation of a new assimilation scheme for soil and surface variables in a global NWP model. *Mon. Wea. Rev.*, **128**, 997-1015.
- Laprise, R., 1992: The Euler equations of motion with hydrostatic pressure as an independent variable. *Mon. Wea. Rev.*, **120**, 197-207.
- Laprise, R., 1992: The resolution of global spectral models. *Bulletin American Meteor. Society*, **73**, 1453-1454.
- Lauritzen, P.H., C. Jablonowski, M.A. Taylor, and R.D. Nair, 2010: Numerical techniques for global atmospheric models. *Lecture notes in computational science and engineering (Springer)*, **80**, 556pp.
- Lynch, P., D. Giard and V. Ivanovici, 1997: Improving the efficiency of a digital filtering scheme for diabatic initialisation. *Mon. Wea. Rev.*, **125**, 1976-1982.
- Lynch, P., R. McGrath and A. McDonald, 1999: Digital filter initialisation for HIRLAM. *HIRLAM Technical Report Nr42*, 22pp.
- Pailleux, J., J.-F. Geleyn, M. Hamrud, Ph. Courtier, J.-N. Thépaut, F. Rabier, E. Andersson, D. Burridge, A. Simmons, D. Salmond, R. El Khatib, C. Fischer, 2014: Twenty-five years of IFS/ARPEGE. *ECMWF Newsletter nr 141*, 22-30.
- 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.
- Radnoti, G., 1995: Comments on "A spectral limited-area formulation with time-dependent boundary conditions applied to the shallow-water equations". *Mon. Wea. Rev.*, **123**, 3122-3124.
- Ritchie, H., C. Temperton, A. Simmons, M. Hortal, T. Davies, D. Dent, and M. Hamrud, 1995: Implementation of the semi-Lagrangian method in a high resolution version of the ECMWF forecast model. *Mon. Wea. Rev.*, **123**, 489-514.
- Rochas, M.J., P. Courtier, K. Yessad, Y. Bouteloup, 1991: Rotations in spectral space. *Beiträge zur Physik der Atmosphäre*, vol **64/3**, 201-206.
- Schmidt, F., 1977: Variable fine-mesh in spectral global model. *Beitr. Phys. Atmos.*, **50**, 211-227.
- Seity, Y., P. Brousseau, S. Malardel, G. Hello, P. Bénard, F. Bouttier, C. Lac, and V. Masson, 2011: The AROME-France Convective-Scale Operational Model. *Mon. Wea. Rev.*, **139**, 976-991.
- Shewchuk, J.R., 1994: An introduction to the conjugate gradient method without the agonizing pain. Edition 1 $\frac{1}{4}$ . Note, 64pp, available on <http://www.cs.cmu.edu/~jrs/>
- 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.
- Temperton, C., 1991: On scalar and vector transform methods for global spectral models. *Mon. Wea. Rev.*, **119**, 1303-1307.
- Untch, A., and M. Hortal, 2004: A finite-element scheme for the vertical discretization of the semi-Lagrangian version of the ECMWF forecast model. *Q. J. R. Meteorol. Soc.*, **130**, 1505-1530.
- Wedi, N.P., K. Yessad and A. Untch, 2009: The nonhydrostatic global IFS/ARPEGE: model formulation and testing. *ECMWF technical memorandum*, **594**, 37pp.

- White, A. A., 2000: A view of the equations of meteorological dynamics and various approximations. *Forecasting Research Scientific Paper nr*, **58**, 106pp.
- White, A. A., B. J. Hoskins, I. Roulstone and A. Staniforth, 2005: Consistent approximate models of the global atmosphere: shallow, deep, hydrostatic, quasi-hydrostatic and non-hydrostatic. *Q. J. R. Meteorol. Soc.*, **131**, 2081-2107.
- Wood, N., and A. Staniforth, 2003: The deep-atmosphere Euler equations with a mass-based vertical coordinate. *Q. J. R. Meteorol. Soc.*, **129**, 1289-1300.
- Yessad, K., and P. Bénard, 1996: Introduction of a local mapping factor in the spectral part of the METEO-FRANCE global variable mesh numerical forecast model. *Quart. J. Roy. Meteor. Soc.*, **122**, 1701-1719.
- Yessad, K., and N.P. Wedi, 2011: The hydrostatic and non-hydrostatic global model IFS/ARPEGE: deep-layer model formulation and testing. *ECMWF technical memorandum*, **657**, 28pp.

### 25.2.3 Theses (PHD).

- (PHDPHYSSIM) Janisková, M., 1998: Realization of a simplified, differentiable and realistic physical parameterization for incremental four-dimensional variational data assimilation. (PHD report), 220pp.
- Veersé, F., 1997: Stratégies de minimisation pour le calcul de l'état initial en météorologie. (PHD report - Manuscrit de thèse), 121pp.

### 25.2.4 ARPEGE notes “Notes de travail ARPEGE” 1 to 33.

- (NTA01) Courtier, Ph., 1988: Constantes. Note de travail ARPEGE numéro **1**, 10pp.
- (NTA02) Rochas, M., 1988: Un système cohérent d'équations pour décrire le cycle atmosphérique de l'eau. Note de travail ARPEGE numéro **2**, 10pp.
- (NTA03) Geleyn, J.F., 1988: Transformations de coordonnées pour aller de la sphère terrestre réelle à la sphère fictive de la grille de collocation et vice-versa. Note de travail ARPEGE numéro **3**, 4pp.
- (NTA04) Clochard, J., 1988: Norme de codage DOCTOR pour le projet ARPEGE. Note de travail ARPEGE numéro **4**, 21pp.
- (NTA05) Rochas, M., 1988: Une autre forme de l'équation de continuité: application à la méthode semi-Lagrangienne. Note de travail ARPEGE numéro **5**, 11pp.
- (NTA06) Durand, Y., Ph. Bougeault et M.C. Pierrard, 1989: Utilisation des mesures satellitaires dans le système d'assimilation à méso-échelle PERIDOT. Première partie: le modèle d'analyse. Note de travail ARPEGE numéro **6**, 38pp.
- (NTA07) Craplet, A., 1989: Interpolations sur la sphère. Note de travail ARPEGE numéro **7**, 17pp.
- (NTA08) Talagrand, O., 1989: Sur l'estimation de la qualité d'une procédure d'estimation. Note de travail ARPEGE numéro **8**, 4pp.
- (NTA09) Duhaut, P., 1989: Recherches de trajectoires; schémas d'advection. Application à un modèle spectral semi-Lagrangien. Note de travail ARPEGE numéro **9**, 39pp.
- (NTA10) Rochas, G., 1989: Traitement de la température et de l'humidité au sol dans les modèles EMERAUDE et PERIDOT. Note de travail ARPEGE numéro **10**, 11pp.
- (NTA11) Bouteloup, Y., 1989: Le relief dans ARPEGE. Note de travail ARPEGE numéro **11**, 21pp.
- (NTA12) Clochard, J., 1989: Logiciel de fichiers indexés. Note de travail ARPEGE numéro **12**, 28pp.
- (NTA13) Bouteloup, Y., 1989: Rotations dans l'espace spectral. Note de travail ARPEGE numéro **13**, 13pp.
- (NTA14) Clochard, J., 1990: Changement de repère local et transformation du vent dans la transformation ARPEGE. Note de travail ARPEGE numéro **14**, 5pp.
- (NTA15) Duhaut, P., et Ph. Courtier, 1990: Comment créer un nouveau cycle de la bibliothèque “maître” ou d'une autre. Note de travail ARPEGE numéro **15**, 5pp.
- (NTA16) Duhaut, P., et Ph. Courtier, 1990: Procédures diverses sous NOS/VE. Note de travail ARPEGE numéro **16**, 34pp.
- (NTA17) Clochard, J., 1990: Interfaces de manipulation des fichiers ARPEGE. Note de travail ARPEGE numéro **17**, 16pp.
- (NTA18) Thepaut, J.N., et Ph. Courtier, 1991: Four-dimensional variational data assimilation using the adjoint of a multilevel primitive equation model. Note de travail ARPEGE numéro **18**, 43pp.
- (NTA19) Rochas, M., Ph. Courtier, K. Yessad, Y. Bouteloup et J. Clochard, 1991: Rotations et dilatations dans l'espace spectral. Note de travail ARPEGE numéro **19**.

- (NTA20) Caille, Ph, 1991: De l'utilisation de procédures élaborées dans le cadre du développement du modèle commun ARPEGE/IFS. Note de travail ARPEGE numéro **20**, 27pp.
- (NTA21) Déqué, M., E. Cordoneanu, J.F. Geleyn, 1991: Initialisation des constantes climatologiques du modèle ARPEGE. Note de travail ARPEGE numéro **21**, 14pp.
- (NTA22) Courtier, Ph., C. Freydier, J.F. Geleyn, F. Rabier and M. Rochas, 1991: The ARPEGE project at METEO-FRANCE. Note de travail ARPEGE numéro **22**, 38pp.
- (NTA23) Marquet, P., et D. Giard, 1992: L'initialisation dans ARPEGE. Note de travail ARPEGE numéro **23**, 41pp.
- (NTA24) Courtier, Ph., 1991: Two difficulties encountered in the validation of cycle 7. Note de travail ARPEGE numéro **24**, 3pp.
- (NTA25) Courtier, Ph., and F. Rabier, 1991: Four-dimensional assimilation in presence of baroclinic instability. Note de travail ARPEGE numéro **25**, 52pp.
- (NTA26) Déqué, M., 1992: La transformation du modèle ARPEGE en modèle de circulation générale atmosphérique. Note de travail ARPEGE numéro **26**, 16pp.
- (NTA27) Joly, A., 1992: Diagnostics sur domaines horizontaux. Guide d'utilisation et de maintenance. Note de travail ARPEGE numéro **27**, 47pp.
- (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.
- (NTA29) Rabier, F., Ph. Courtier, J. Pailleux, O. Talagrand and D. Vasiljevic, 1992: A comparison between four-dimensional variational assimilation and simplified sequential assimilation relying on three-dimensional variational analysis. Note de travail ARPEGE numéro **29**, 28pp.
- (NTA30) Rochas, M., et Ph. Courtier, 1992: La méthode spectrale en météorologie. Note de travail ARPEGE numéro **30**, 58pp.
- (NTA31) Juvanon du Vachat, R., 1993: Initialisation intrinsèque sur la sphère transformée ARPEGE: étude analytique préliminaire. Note de travail ARPEGE numéro **31**, 22pp.
- (NTA32) Bouttier, F., 1993: The Kalman filter in ARPEGE/IFS. Note de travail ARPEGE numéro **32**, 18pp.
- (NTA33) Juvanon du Vachat, R., 1993: Schémas itératifs d'initialisation non linéaire et d'initialisation dynamique par modes normaux. Note de travail ARPEGE numéro **33**, 24pp.

Remarks:

- Notes (NTA06), (NTA18), (NTA22), (NTA24), (NTA25), (NTA28), (NTA29), (NTA32) are in English; the other notes are in French.
- Notes (NTA06) and (NTA10) concern rather the old models EMERAUDE and PERIDOT.
- Notes (NTA15), (NTA16) and (NTA20) are now obsolete (valid for NOS/VE system on CDC machines).
- Notes (NTA12) and (NTA17) have been recently updated and the most recent versions (IDLFI) and (IDFA) are available on "<http://www.umr-cnrm.fr/gmapdoc/>".
- Note (NTA18) has been published in *Quart. J. Roy. Meteor. Soc.*.
- Note (NTA19) is a compilation of 4 papers. The first one is published in *Beiträge zur Physik der Atmosphäre* as (Rochas et al., 1991). The last one has been updated as (IDRD) (see below).
- Note (NTA22) has been published in an ECMWF seminar proceeding as (Courtier et al., 1991).
- Note (NTA25) has been published in *Quart. J. Roy. Meteor. Soc.*, July 1992 issue.
- Note (NTA27) has been updated by J.M. Piriou as internal note (IDDDH), see below.
- Note (NTA28) has been published in *Beitr. Phys. Atmos.*, August 1992 issue (Vol. **65** nr 3, 177-192).
- Note (NTA29) has been published in *Quart. J. Roy. Meteor. Soc.*, July 1993 issue (Vol. **119**, 845-880).
- Some aspects of (NTA30) have been translated in English and introduced in documentation (IDTS).

### 25.2.5 Internal documentations describing operational configurations.

- (ITOP1) 2017. Modèles: configurations et contrôle. Internal note, available on "<http://controle.meteo.fr/>".

## 25.2.6 Other internal documentations.

Each documentation has a code and the alphabetical order is the one of the code. The name of the code starts by "ID" for internal documentations, "IT" for some internal papers, "TDEC" for technical documentations written at ECMWF.

- (ID601) Fischer, C., 2001: The computation of singular vectors inside ARPEGE/ALADIN. Internal documentation valid for CY22T2/AL13, 12pp.
- (ID801O) Bouttier, F., 1995: Sensitivity jobs with configuration 801. Internal documentation valid for cycle 13r3, 13pp, available on "<http://gco.meteo.fr/ifsdoc/html/>" (file "sensit").
- (ID801) Yessad, K., 2013: Configuration 801 in the cycle 40 of ARPEGE/IFS (internal note). Internal note available on "<http://www.umr-cnrm.fr/gmapdoc/>".
- (ID901A) Saez, P., 2008: Notice d'utilisation de la configuration 901 ARPEGE ISBA. Internal note available on "<http://www.umr-cnrm.fr/gmapdoc/>".
- (ID923OROG) Giard, D., 2005: Computation of spectral orography in configuration 923. Internal note, 3pp, available on "<http://www.umr-cnrm.fr/gmapdoc/>".
- (ID923E) Pristov, N., 2002: Configuration E923 user's guide (AL12). Internal note, 11pp, available on "<http://www.umr-cnrm.fr/gmapdoc/>".
- (IDAB) Bénard, P., 2008: On the design of the vertical coordinate  $\eta$ . Internal note (7pp), available on "<http://www.umr-cnrm.fr/gmapdoc/>".
- (IDADJAPPLI) Bouttier, F., 1992: Applications of TL and AD codes in ARPEGE. Internal note, valid for cycle 13r3, available on "<http://gco.meteo.fr/ifsdoc/html/>" (file "adjappli").
- (IDADJINTRO) Bouttier, F., 1992: Introduction to TL and AD codes in ARPEGE. Internal note, valid for cycle 13r3, available on "<http://gco.meteo.fr/ifsdoc/html/>" (file "adjintro").
- (IDARC) Yessad, K., 2017: Library architecture and history of the technical aspects in ARPEGE/IFS, ALADIN and AROME in the cycle cycle 45 of ARPEGE/IFS. Internal note, available on "<http://www.umr-cnrm.fr/gmapdoc/>".
- (IDARPC6) 2017: ARPEGE-CLIMAT V6.2. Algorithmic documentation (CY37T1). Internal note (351pp), available on "<http://www.umr-cnrm.fr/gmapdoc/>".
- (IDBDAP) Piriou, J.-M., 2005: Documentation succincte des articles physiques de fichiers ARPEGE-ALADIN (Documentation of physical rows). Internal note, 8pp, available on "<http://www.umr-cnrm.fr/gmapdoc/>".
- (IDBDMO) Guillaume, F., 2016: Getting observational data from BDMO for atmospheric models (two parts). Internal note, available on "<http://www.umr-cnrm.fr/gmapdoc/>".
- (IDBLEN) (author=???), 2001: Blending of initial fields in ALADIN. Internal note, 29pp.
- (IDBUFR) Thorpe, W., 1998: A guide to the WMO code form FM 94 BUFR. Internal note, 89pp, available on "<http://rda.ucar.edu/docs/formats/bufr/>" (file "bufr.pdf"). (Remark: there are other "pdf" files dated from 2002 giving additional documentation).
- (IDCA) Taillefer, F., 2002: CANARI technical documentation in CY25T1/AL25T1. Internal note, 55pp, available on "<http://www.umr-cnrm.fr/gmapdoc/>".
- (IDCGRA) Fisher M., 1998: Minimization algorithms for variational data assimilation. Proceedings of ECMWF seminar on recent developments in numerical methods for atmospheric modelling, 7-11 Sept 1998, pp 364-385.
- (IDCPN) Coiffier, J., 2009: Les bases de la prévision numérique du temps. Cours et manuels no 18, 256pp.
- (IDCRT) El Khatib, R., 2003: Coding standards for ARPEGE/IFS/ALADIN. Internal note, 42pp, available on "<http://www.umr-cnrm.fr/gmapdoc/>".
- (IDCS11) Rivière, O., C. Fischer and M. Fisher, 2011: Coding standards for ARPEGE, IFS and ALADIN systems (V6). Internal note, 10 pp, available on "<http://www.umr-cnrm.fr/gmapdoc/>".
- (IDCS13A) ???, 2013: C++ coding guidelines. Internal note, 4 pp (based on a draft document from Baudouin Raoult).
- (IDCS13B) ???, 2013: Règles de codage V0R6 avec annexes. Internal note, 26 pp.
- (IDDDH) Piriou, J.-M., 2014: Diagnostics in Horizontal Domains (DDH). Variables and budget equations, in horizontal mean. ARPEGE, ALADIN and AROME models. Guide for users and developpers. Internal note, 75pp, available on "<http://www.umr-cnrm.fr/gmapdoc/>".
- (IDDDHU) Piriou, J.M., 2002: The use of DDH in ARPEGE - ALADIN. Internal note, 13pp, available on "<http://www.umr-cnrm.fr/gmapdoc/>".

- (IDDDHN) Riviere, O., 2009: New dataflow for DDH in Aladin/Arpege/Alaro/Arome (CY35T2): documentation of a new diagnostic dataflow for DDH. Internal note, 10pp, available on "<http://www.umr-cnrm.fr/gmapdoc/>".
- (IDDFI94) Giard, D., 1994: L'initialisation par filtre digital dans ARPEGE. Internal documentation in French valid for cycle 11r2, 22pp.
- (IDDFI01) Giard, D., 2001: Digital filtering and data assimilation. Internal documentation, 22pp, available on "<http://www.umr-cnrm.fr/alatnet/gourdon/cours/DG/Plan.html>".
- (IDDFI) Yessad, K., 2017: Digital filtering initialisation in the cycle 45 of ARPEGE/IFS. Internal note, available on "<http://www.umr-cnrm.fr/gmapdoc/>".
- (IDDH) Yessad, K., 2017: Horizontal diffusion in the cycle 45 of ARPEGE/IFS. Internal note, available on "<http://www.umr-cnrm.fr/gmapdoc/>".
- (IDDM) Yessad, K., 2017: Distributed memory features in the cycle 45 of ARPEGE/IFS. Internal note, available on "<http://www.umr-cnrm.fr/gmapdoc/>".
- (IDEQR) Mozdzynski, G., 2006: A new partitioning approach for IFS. Internal note, 6pp.
- (IDEUL) Yessad, K., 2017: Integration of the model equations, and Eulerian dynamics, in the cycle 45 of ARPEGE/IFS. Internal note, available on "<http://www.umr-cnrm.fr/gmapdoc/>".
- (IDFA) Clochard, J., R. El Khatib and D. Paradis, 2002: FA subroutines of the ARPEGE/ALADIN files package (Interfaces de manipulation des fichiers ARPEGE/ALADIN) Internal note, 34 pp; is an updated version of (NTA17) valid for cycle 24, available in English and French on "<http://www.umr-cnrm.fr/gmapdoc/>".
- (IDFPMA) El Khatib R., 2013: Fabrication des matrices de filtrage pour le post-traitement d'ARPEGE. Internal note, 2pp.
- (IDFPOS) Yessad, K., 2017: FULL-POS in the cycle 45 of ARPEGE/IFS. Internal note, available on "<http://www.umr-cnrm.fr/gmapdoc/>".
- (IDFPOS2) El Khatib R., 2012: FULLPOS-2: design, specifications study and development (version 6). Internal note, 15pp.
- (IDFSTB) El Khatib, R., 2003: From source code management to binaries production for ARPEGE/ALADIN. Internal note, 14pp, available on "<http://www.umr-cnrm.fr/gmapdoc/>".
- (IDFU) Yessad, K., 2017: CFU (cumulated fluxes) and XFU (instantaneous fluxes) in the cycle 45 of ARPEGE/IFS. Internal note, available on "<http://www.umr-cnrm.fr/gmapdoc/>".
- (IDGFL) Tudor, M., 2004: A short description of physics/dynamics interface in the new data flow. Internal note, 21pp.
- (IDGMKPACK) El Khatib, R., 2014: Getting started with GMKPACK environment to compile and make binaries. Internal note, available on "<http://www.umr-cnrm.fr/gmapdoc/>".
- (IDGOMU) Geer, A., 2013: GOM user guide: interpolation to observation space. Internal note, 3pp.
- (IDGPS) Poli, P., 2007: Pre-processing of ZTD observations from GPS stations. Internal note, available on "<http://www.umr-cnrm.fr/gmapdoc/>".
- (IDGRIB) (updated version of (IDGRIG) in 2000, in HTML form, see below).
- (IDGRIC) GRIB Format, Edition 1. Internal note, available on "<http://weather.unisys.com/wxp/Appendices/Formats/GRIB.html>".
- (IDGRIG) Dey, C.H., 1998: The WMO format for the storage of weather product information and the exchange of weather product messages in gridded binary form as used by NCEP central operations. Internal note, 106pp, available on "<http://rda.ucar.edu/docs/formats/>" (topics "WMO GRIB Edition 1, Adobe PDF").
- (IDGSA) Bouttier, F., 1996: Specifications of the GSA file format (version 1.1). Internal note, available on "<http://intra.cnrm.meteo.fr/assim/doc/gsaspec.html>".
- (IDHIST) El Khatib, R., and J.M. Piriou, 2002: Fichiers historiques ARPEGE/ALADIN. Internal documentation in French, 6pp).
- (IDHL) Sass, B. H., 2006: Implementation of HIRLAM physics in ALADIN. Internal note, 17pp.
- (IDIDM) El Khatib, R., 2002: An introduction to the memory-distributed aspect of the code ARPEGE/IFS/ALADIN. Internal note, 11pp, available on "<http://www.umr-cnrm.fr/gmapdoc/>".
- (IDIO) Yessad, K., 2017: IO in the cycle 45 of ARPEGE/IFS. Internal note, available on "<http://www.umr-cnrm.fr/gmapdoc/>".

- (IDTECIO) Marguinaud, Ph., 2013: ARPEGE/ALADIN/AROME IO in 39t1. Internal note, 18pp. Available on the intranet server "<http://www.umr-cnrm.fr/gmapdoc/>".
- (IDIPRO) (author=???), 2002: Procédures diverses pour ARPEGE/ALADIN (ASCS, CHAGAL, ECTO, EDF, FRODO, LAMFLAG\_ODB, MITRAILLETTE, SUBGRID, SURGERY). Compilation of internal notes, available on "<http://www.umr-cnrm.fr/gmapdoc/>".
- (IDLAM) Zagar, M., and C. Fischer, 2007: The ARPEGE/ALADIN Tech'Book: Implications of LAM aspects on the global model code for CY33/AL33. Internal note, 31pp, available on "<http://www.umr-cnrm.fr/gmapdoc/>".
- (IDLFA) Piriou, J.M., 2001: LFA autodocumented files software. Internal note, 27pp available on "<http://www.umr-cnrm.fr/gmapdoc/>".
- (IDLFI) Bouttier, F., and J. Clochard, 2002: The LFI file access library (valid for cycle 24T1). (internal note, 8 pp; is an updated version of (NTA12)). Available on "<http://www.umr-cnrm.fr/gmapdoc/>".
- (IDMCUF) Termonia, P., 2004: Monitoring the coupling update frequency (MCUF): Implementation in cy28t2, ARPEGE/ALADIN. Internal note, 6 pp.
- (IDMINI) Yessad, K., 2017: Minimizations in the cycle 45 of ARPEGE/IFS (internal note). Internal note, available on "<http://www.umr-cnrm.fr/gmapdoc/>".
- (IDMITR) Vanda Sousa da Costa, A. Deckmyn, A. Dziedzic, N. Bouzouita and G. Bölöni, 2006: MITRAILLETTE: procedure to validate a release of the code Aladin, version 4. Internal note, available on "<http://www.umr-cnrm.fr/gmapdoc/>".
- (IDMITS) Yessad, K., 2017: MITRAILLETTE: environnement files and user's guide (version v022017 valid for cycle 45). Internal note, 8pp, available on "<http://www.umr-cnrm.fr/gmapdoc/>".
- (IDMNHS) 2001: The Meso-NH atmospheric simulation system: scientific documentation. Internal note, 337pp.
- (IDMOAL) Joly, A., 1992: ARPEGE/ALADIN: adiabatic model equations and algorithm (internal note, 56 pp).
- (IDNEWT) Veersé, F., 1996: Quelques algorithmes de minimisation pour l'assimilation variationnelle de données météorologiques. Internal note (slides-sized), 26pp.
- (IDNGFL) Yessad, K., 2017: User's guide to add new GFL variables or new GFL attributes in ARPEGE/IFS, ALADIN, AROME: cycle 45. Internal note, 10pp, available on the internet server "<http://www.umr-cnrm.fr/gmapdoc/>".
- (IDNHPB) Bénard, P., 2013: Scientific documentation for ALADIN NH model (version 3.1). Internal note (94pp), available on "<http://www.umr-cnrm.fr/gmapdoc/>".
- (IDNMI) ???, 1993: Namelists pour l'initialisation par NMI. Internal note in French valid for cycle 11 (dated June 1993), 9pp.
- (IDNSUR) Yessad, K., 2017: User's guide to add new surface variables or new surface attributes in ARPEGE/IFS, ALADIN, AROME: cycle 45. Internal note, 8pp, available on the internet server "<http://www.umr-cnrm.fr/gmapdoc/>".
- (IDOBSTAT) Poli, P., 2008: OBSTAT in OLIVE. Internal note (14pp), available on "<http://www.umr-cnrm.fr/gmapdoc/>".
- (IDODOBO) Kertész, S., 2001: A short overview of ODB. Internal note, 14 pp, available on "<http://www.umr-cnrm.fr/gmapdoc/>".
- (IDODBT) Puech, D., 2009: Documentation technique sur ODB. compilation of internal notes (generally in French) available on "<http://www.umr-cnrm.fr/gmapdoc/>".
- (IDODBU) Kertész, S., 2002: Using ODB in ALADIN. Internal note, 21pp, available on "<http://www.umr-cnrm.fr/gmapdoc/>".
- (IDODBZ) Fouilloux, A., 2013: ODB documentation (several .tar files, restricted access on ECMWF intranet).
- (IDOLIV) Mathiot, V., E. Sevault, and Ph. Marguinaud, 2006: OLIVE User's Guide (SWAPP 2.0). Internal note, 143pp, available on "[http://sxalgo1.cnrm.meteo.fr:8181/swapp\\_entry/groucho/Swapp/Browse/etc/apps/Swapp/](http://sxalgo1.cnrm.meteo.fr:8181/swapp_entry/groucho/Swapp/Browse/etc/apps/Swapp/)".
- Bénard, P., 2002: Harmonization and rationalization of iterative time-schemes in IFS/ARPEGE/ALADIN. Internal note (11pp).
- (IDOMP1) Gondet E. et P.F. Lavallée: Cours OpenMP, CNRS-IDRIS, 2000. 157pp.
- (IDOMP2) OpenMP Application program interface, version 3.0, 2008. 326pp.
- (IDOUL) Guillaume, F., 2007: Retrieval of observational data for ARPEGE/ALADIN assimilation. User's guide (50 pp), available on "<http://www.umr-cnrm.fr/gmapdoc/>".

- (IDPC) Bénard, P., 2002: Incremental versus non-incremental predictor-corrector schemes. Appendix A of the previous internal note.
- (IDPCEU) Vivoda, J., 2004: Three-time level iterative centred implicit scheme with Eulerian advection treatment. Internal note, 16pp.
- (IDPCSL) Vivoda, J., 2005: Two-time level iterative centred implicit semi-Lagrangian scheme with grid-point prognostic variable  $gw$  and semi-implicit variable  $d$  or  $d_4$ . Internal note, 15pp.
- (IDPHYE) Gérard, L., 2001: Physical parameterizations in ARPEGE/ALADIN (CY23T1/AL14 ?). Internal note, available on "<http://www.umr-cnrm.fr/gmapdoc/>". (this documentation has been written in 2000 for CY23T1/AL14 and some parts seem have been updated for more recent cycles).
- (IDPHYS) Piriou, J.M., 1999: La physique d'ARPEGE. Version cycle 18. Internal note in French valid for cycle 18 describing not only MF physics but also some other aspects of the code.
- (IDPRAL) Janoušek, M., 2001: New port of the new geographical routines to ALADIN/ARPEGE (internal note).
- (IDPRLAM) Bénard, 2011: "Rotated/Tilted Mercator" geometry in ALADIN. Internal note, 26pp, available on "<http://www.umr-cnrm.fr/gmapdoc/>".
- (IDPROJ) Gril, J-D., 2009: Projections used in ALADIN and AROME limited area models (LAM). Generality and description of the code. Internal note, 41pp, available on "<http://www.umr-cnrm.fr/gmapdoc/>".
- (IDQN3) Gilbert, J.C., and C. Lemaréchal, 1993: The modules M1QN3 and N1QN3. Version 2.0b. Internal note, 12pp.
- (IDRD) Yessad, K., 2017: Sphere to sphere transforms in spectral space in the cycle 45 of ARPEGE/IFS: configuration 911 (internal note). Internal note, available on "<http://www.umr-cnrm.fr/gmapdoc/>".
- (IDREST) El Khatib, R., 1992: Le redémarrage (restart) dans ARPEGE. Internal note in French valid for cycle 10 (dated 16 June 1992), 2pp.
- (IDRGRI) El Khatib, R., 2013: Calcul des grilles de Gauss réduites pour ARPEGE. Internal note in French valid for cycle 40 (dated 27 May 2013), 2pp.
- (IDRVN) Yessad, K., 2017: Recommended variable naming in ARPEGE/IFS. Internal note.
- (IDS4DV) Desroziers, G., et P. Riber, 2002: Documentation du script 4D-VAR. Internal note in French valid for cycle 25t1, 33pp, available on "<http://www.umr-cnrm.fr/gmapdoc/>".
- (IDSCM) Piriou, J.M., F. Bouyssel and E. Bazile, 1999: ARPEGE-ALADIN single column model: scientific and algorithmical aspects. Internal note, 23pp, available on "<http://www.umr-cnrm.fr/gmapdoc/>".
- (IDSFEX) Le Moigne, P., 2009: SURFEX scientific documentation (version 5). Internal note, 211pp, available on "<http://www.umr-cnrm.fr/gmapdoc/>".
- (IDSI) Yessad, K., 2017: Semi-implicit spectral computations in the cycle 45 of ARPEGE/IFS. Internal note, available on "<http://www.umr-cnrm.fr/gmapdoc/>".
- (IDSL) Yessad, K., 2017: Semi-Lagrangian computations in the cycle 45 of ARPEGE/IFS. Internal note, available on "<http://www.umr-cnrm.fr/gmapdoc/>".
- (IDSLHD) Vaña, F., 2006: Semi-Lagrangian horizontal diffusion in ALADIN/ARPEGE (valid for CY30T1). Internal note, 13pp, available on "<http://www.umr-cnrm.fr/gmapdoc/>".
- (IDSLIF) Mašek, J., and F. Vaña, 2006: Study of semi-Lagrangian interpolators in idealized framework. Report from RC LACE scientific stay, 27pp.
- (IDSPL) Vaña, F., 2005: Spline interpolation in semi-Lagrangian advection scheme of ALADIN/ARPEGE/IFS. Internal note, 8pp, available on "<http://www.umr-cnrm.fr/gmapdoc/>".
- (IDSULC) El Khatib, R., 1993: Le setup par ligne de commande: description et mode d'emploi. Internal note in French valid for cycle 11 (dated 3 June 1993), 7pp. Update internal note from 15 Feb 1994, 1p.
- (IDSVTSM) Hortal, M., and A. Untch, 2003: A new interpolation for the vertical computation of the semi-Lagrangian trajectory. Internal note, 7pp.
- (IDTESTAD) Bouttier, F., 1995: Test of the adjoint model with configuration nr 401. Internal note, valid for cycle 13r3, available on "<http://gco.meteo.fr/ifsdoc/html/>" (file "testad").
- (IDTESTLI) Bouttier, F., 1995: Test of the tangent linear model with configuration nr 501. Internal note, valid for cycle 13r3, available on "<http://gco.meteo.fr/ifsdoc/html/>" (file "testli").
- (IDTS) Yessad, K., 2017: Spectral transforms in the cycle 45 of ARPEGE/IFS. Internal note, available on "<http://www.umr-cnrm.fr/gmapdoc/>".

- (IDTSAL) Radnoti, G., 2001: The transformation package for ALADIN. Internal note, 6pp, available on “<http://www.umr-cnrm.fr/gmapdoc/>”.
- (IDTSN) Hamrud, M., 2002: New transform library. Internal note (9pp) available on “<http://www.umr-cnrm.fr/gmapdoc/>”.
- (IDUOBS) Bénichou, H., 2008: Utilisation des différents types d’observation dans l’assimilation opérationnelle d’ARPEGE. Internal note available on “<http://controle.meteo.fr/>”.
- (IDVAR) Fischer, C., and L. Berre, 2007: The variational computations inside ARPEGE/ALADIN: cycle CY32. Internal note, 77pp, available on “<http://www.umr-cnrm.fr/gmapdoc/>”.
- (IDVARALD) Fischer, C., T. Montmerle and V. Mathiot, 2005: Description technique du cycle d’assimilation 3DVAR ALADIN-FRANCE. Internal note, 14pp, available on “<http://www.umr-cnrm.fr/gmapdoc/>”.
- (IDVNH1) Smolíková, P., 2001: New strategies for non-hydrostatic temporal scheme. Internal note, 24pp.
- (IDVNH2) Smolíková, P., 2002: New NH variables:  $d_4$  in three dimensions (in the cycle CY25T1). Internal note, 16pp.
- (IDVNH3) Smolíková, P., 2003: The use of diagnostic BBC (boundary bottom condition) in semi-Lagrangian schemes. Internal note, 8pp.
- (IDVORTEX) 2017: VORTEX: versatile objects rounded-up in a toolbox for environmental experiments. Available at <http://sicompas/vortex/doku.php/start>.
- (IGIT) Martinez, S., and C. Fischer, 2013: Météo-France’s GIT toolbox: GIT tools (version 1). Internal note, 22pp, available on “<http://www.umr-cnrm.fr/gmapdoc/>”.
- (IGIT2) Martinez, S., 2016: GIT toolbox version 2.0: main new features. Internal note, available on “<http://www.umr-cnrm.fr/gmapdoc/>”.
- (ITAS) 1989: ARPEGE. Partie I: aspects scientifiques.
- (ITAA) 1989: ARPEGE. Partie II: aspects algorithmiques.
- (TDECOBS25) 2002: IFS technical documentation (CY25R1). Part I: observation processing.
- (TDECOBS) 2016: IFS technical documentation (CY41R2). Part I: observation.
- (TDECOB) Saarinen, S., 2004: IFS technical documentation (CY28R1). Part I: ODB.
- (TDECDAS) 2016: IFS technical documentation (CY41R2). Part II: data assimilation.
- (TDECDYN) 2016: IFS technical documentation (CY41R2). Part III: dynamics and numerical procedures.
- (TDECPHY) 2016: IFS technical documentation (CY41R2). Part IV: physical processes.
- (TDECEPS) 2016: IFS technical documentation (CY41R2). Part V: ensemble prediction system.
- (TDECTEC25) 2002: IFS technical documentation (CY25R1). Part VI: technical and computational procedures.
- (TDECTEC) 2016: IFS technical documentation (CY41R2). Part VI: technical and computational procedures.
- (TDECWAV) 2016: IFS technical documentation (CY41R2). Part VII: ECMWF wave model.

Remarks:

- The IFS technical documentation can be found on:  
<https://software.ecmwf.int/wiki/display/IFS/Official+IFS+Documentation>
- Documentation (ID901A) actually describes the particular case when a FULL-POS configuration is launched on a file produced by a configuration 901; this is not a detailed description of configuration 901.

## Appendix 1: topics and description location.

### a) Codes of topics.

Code	Meaning
C001FU	Configuration 1, CFU and XFU diagnostics
C001CHET	Configuration 1, Physical tendencies diagnostics
C001DDH	Configuration 1, DDH diagnostics
C001DFI	Configuration 1, digital filters initialisation (direct code)
C001DHEC	Configuration 1, ECMWF horizontal diffusion scheme (direct code)
C001DHMF	Configuration 1, MF horizontal diffusion scheme (direct code)
C001ECR	Configuration 1, writing on historical files
C001EU	Configuration 1, Eulerian grid-point calculations (direct code)
C001FPOS	Configuration 1, FULLPOS
C001LEC	Configuration 1, reading files
C001MCUF	Configuration 1, MCF (monitoring the coupling update freq) diagnostics
C001NH	Configuration 1, non hydrostatic aspects
C001NUD	Configuration 1, nudging (in spectral space)
C001PC	Configuration 1, predictor-corrector scheme (direct code)
C001PHYEC	Configuration 1, ECMWF physics
C001PHYMF	Configuration 1, METEO-FRANCE physics
C001PHYMFC	Configuration 1, METEO-FRANCE physics in the climate model
C001PHYSMF	Configuration 1, METEO-FRANCE simplified physics (direct code)
C001PHYHL	Configuration 1, HIRLAM physics
C001RES	Configuration 1, restart
C001SCREEN	Configuration 1, screening of observations
C001SURFEX	Configuration 1, SURFEX surface scheme
C001SI	Configuration 1, semi-implicit scheme (direct code)
C001SL	Configuration 1, semi-Lag. grid-point calculations (direct code)
C001TS	Configuration 1, spectral transforms (direct code)

Topics for configuration 1.

Code	Meaning
C131	Configuration 131: incremental 4-D VAR/3-D VAR

Topics for configurations 101 to 199 (variational jobs).

Code	Meaning
C20xDHEC	Configurations 201 and 202, ECMWF horizontal diffusion scheme (direct code)
C20xDHMF	Configurations 201 and 202, MF horizontal diffusion scheme (direct code)
C20xECR	Configurations 201 and 202, writing on historical files
C20xEU	Configurations 201 and 202, Eulerian grid-point calculations (direct code)
C20xLEC	Configurations 201 and 202, reading files
C20xSI	Configurations 201 and 202, semi-implicit scheme (direct code)
C20xSL	Configurations 201 and 202, semi-Lag. grid-point calculations (direct code)
C20xTS	Configurations 201 and 202, spectral transforms (direct code)

Topics for configurations 201 to 202 (2D models).

Code	Meaning
C401	Configuration 401: test of adjoint with 3-D primitive equation model
C421	Configuration 421: test of adjoint with shallow-water model
C422	Configuration 422: test of adjoint with vorticity equation model
C501	Configuration 501: test of tangent linear with 3-D primitive equation model
C521	Configuration 521: test of tangent linear with shallow-water model
C522	Configuration 522: test of tangent linear with vorticity equation model

Topics for configurations 401 to 599 (test of adjoint and tangent linear).

Code	Meaning
C601	Configuration 601: eigenvalue/vector solver for 3-D primitive equation model

Topics for configurations 601 to 699 (eigenvalue/vector solver).

Code	Meaning
C701	Configuration 701: optimal interpolation with CANARI for 3-D primitive equation model

Topics for configurations 701 to 799 (optimal interpolation with CANARI).

Code	Meaning
C801	Configuration 801: sensitivity with 3-D primitive equation model

Topics for configurations 801 to 899 (tests of sensitivity).

Code	Meaning
C901	Configuration 901: set up initial conditions (CPREP1)
C903	Configuration 903: off-line FULL-POS (CPREP3)
C911	Configuration 911: computes dilatation matrices for spectral space
C923	Configuration 923: initialisation of climatological files at METEO-FRANCE
C931	Configuration 931: creation of an ARPEGE file containing the NESDIS SST (CSSTBLD)
C932	Configuration 932: interpolates the sea-ice concentration field from satellite data to the ARPEGE grid (CSEAICE)

Topics for configurations 901 to 999.

Code	Meaning
BLEND	Blending
BUFR	BUFR format files
CLIMATEC	Climatic simulations at ECMWF
CLIMATMF	Climatic simulations at METEO-FRANCE
CODRUL	Norms of codage
DM	Distributed memory features
FA	ARPEGE "FA" files
FICLIMEC	Initialisation of climatological files at ECMWF
FSTB	Actions from source code management to binaries production
GIT	GIT software
GRIB	GRIB format files
IO	IO: file reading and writing
IPRO	Procedures (for ex: EDF, MITRAILLETTE)
LAM	Limited area aspects
LFA	LFA files
LFI	LFI files
MECA	Mechanics of the model, technical overview
MINIM	Minimisation of functions, linear algebra
MITRAILLETTE	Procedure MITRAILLETTE
OLIVE	Software OLIVE, SWAPP
OBSINTERP	Observation interpolator
OBSPROC	Processing of observations
ODB	Observational Database
OPEREC	Operational model at ECMWF
OPERENSEC	Ensemble prediction operational at ECMWF
OPERMF	Operational model at METEO-FRANCE
SCM	Single column model
SETUP	Setup of the model
TRAGEDI	Transformations from stretched geometry to geographical geometry
UOBS	Way of use of observations in the assimilation
VORTEX	Software VORTEX

General topics.

## b) Where to find documentation for each topic?

Remark: when several references are cited the most recent has to be taken first, some of them can contain obsolete features.

Topic code	Documentation code
C001FU	(IDFU)
C001CHET	No detailed documentation available
C001DDH	(IDDDH), (IDDDHU), (IDDDHN), (NTA27)
C001DFI	(Lynch et al., 1997), (Lynch et al, 1999), (IDDFI01), (IDDFI)
C001DHEC	(TDECDYN)
C001DHMF	(IDDH), (Yessad and Bénard, 1996)
C001ECR	No detailed documentation available
C001EU	(NTA22), (TDECDYN), (ITAS), (ITAA), (IDEUL)
C001FPOS	(IDFPOS), (IDFPOS2) (NTA07) and (NTA14) for horiz. interp., (TDECTEC)
C001LEC	No detailed documentation available
C001MCFU	(IDMCFU)
C001NH	(IDNHPB), (IDVNH1), (IDVNH2), (IDEUL)
C001NUD	No detailed documentation available
C001PC	(IDSI), (IDPC), (IDPCEU), (IDPCSL)
C001PHYEC	(TDECPHY)
C001PHYMF	(IDPHYS), (IDPHYE), (NTA02), (NTA22)
C001PHYMFC	(IDARPC6)
C001PHYSMF	(PHDPHYSSIM)
C001PHYHL	(IDHL)
C001RES	(IDREST)
C001SCREEN	(IDVAR)
C001SURFEX	(IDSFEX)
C001SI	(IDSI), (Yessad and Bénard, 1996), (TDECDYN)
C001SL	(IDSL), (NTA05), (NTA07), (NTA09), (NTA14), (TDECDYN), (ITAA)
C001TS	(IDTS), (IDTSAL), (IDTSN), (NTA30), (ITAS), (Temperton, 1991)

Documentations for configuration 1.

Topic code	Documentation code
C131	(IDVAR), (IDS4DV), (TDECDAS)

Documentations for configurations 101 to 199 (variational jobs).

Topic code	Documentation code
C20xDHEC	No detailed documentation available
C20xDHMF	(IDDH), (Yessad and Bénard, 1996)
C20xECR	No detailed documentation available
C20xEU	(NTA22), (ITAA), (IDEUL)
C20xLEC	No detailed documentation available
C20xSI	(IDSI), (Yessad and Bénard, 1996)
C20xSL	(IDSL), (NTA05), (NTA07), (NTA09), (NTA14), (ITAA)
C20xTS	(IDTS), (IDTSN), (NTA30), (ITAS), (Temperton, 1991)

Documentations for configurations 201 to 202 (2D models).

Topic code	Documentation code
C401	(IDEUL)
C421	(IDEUL)
C422	(IDEUL)
C501	(IDEUL)
C521	(IDEUL)
C522	(IDEUL)

Documentations for configurations 401 to 599 (test of adjoint and tangent linear).

Topic code	Documentation code
C601	(ID601), (TDECEPS)

Documentations for configurations 601 to 699 (eigenvalue/vector solver).

Topic code	Documentation code
C701	(Giard and Bazile, 2000), (IDCA), (ITAS)

Documentations for configurations 701 to 799 (optimal interpolation with CANARI).

Topic code	Documentation code
C801	(NTA28), (ID801O), (ID801)

Documentations for configurations 801 to 899 (tests of sensitivity).

Topic code	Documentation code
C901	(ID901A), not yet completely documented
C903	(IDFPOS)
C911	(IDRD), (NTA19)
C923	(IDARPC6), see also (NTA11) for orography, (ID923E)
C931	No detailed documentation available
C932	No detailed documentation available

Documentations for configurations 901 to 999.

Topic code	Documentation code
BLEND	(IDBLEN)
BUFR	(IDBUFR)
CLIMATEC	No detailed documentation available
CLIMATMF	(IDARPC6), (NTA26)
CODRUL	(NTA04), (IDCRT), (IDCS11), (IDCS13A), (IDCS13B)
DM	(IDIDM), (IDDM), (TDECTEC25), (TDECTEC), (IDEQR)
FA	(NTA17), (IDFA), (IDBDAP)
FICLIMEC	(TDECPHY)
FSTB	(IDFSTB)
GIT	(IGIT)
GRIB	(IDGRIB), (IDGRIC), (IDGRIG)
IO	(IDIO), (IDTECIO)
IPRO	(IDIPRO)
LAM	(IDLAM)
LFA	(IDLFA)
LFI	(NTA12), (IDLFI)
MECA	(TDECDYN), (TDECTEC25), (TDECTEC)
MINIM	(IDQN3), (IDCGRA), (IDMINI)
MITRAILLETTE	(IDMITR), (IDMITS)
OLIVE	(IDOLIV)
OBSINTERP	(NTA07) and (NTA14) for horizontal interpolations, (TDECDAS)
OBSPROC	(TDECOBS25), (TDECDAS)
ODB	(IDODBO), (IDODBT), (IDODBU), (IDODBZ), (TDECOBDB)
OPEREC	(Ritchie et al., 1995), documentations (TDEC...), (ITOPI)
OPERENSEC	(TDECEPS)
OPERMF	(ITOPI)
SCM	(IDSCM)
SETUP	(NTA01) and (NTA21) for constants, (TDECTEC25), (ITAS), (IDSULC)
TRAGEDI	(IDRD) and (NTA19) for conf 911, (NTA03) and (ITAA) for TRACARE/TRARECA
UOBS	(IDUOBS)
VORTEX	(IDVORTEX)

General topics.