November 2002

CANARI

- TECHNICAL DOCUMENTATION -

Based on ARPEGE cycle **CY25T1** (AL25T1 for ALADIN)

Françoise TAILLEFER - CNRM/GMAP

CONTENTS

1. General presentation

CANARI ("Code d'Analyse Nécessaire à ARPEGE pour ses Rejets et son Initialisation" in French) is a software which allows to produce an ARPEGE/ALADIN output file containing all the analyzed fields needed by the forecast model. It is included in the IFS/ARPEGE/ALADIN source library, so it is coded according to the same well known "doctor" rules.

CANARI works in distributed memory configuration and the results are of course independent of the number of used processors; but it is not so efficient as the configuration 1 for example, because there are some parts which are not so well vectorized (distance calculating, linear systems solving,...) and it is not possible to balance the tasks on each processor because the size of the linear systems is very variable according to the observations distribution.

The method used to produce the analysis is based on optimum interpolation (OI analysis), which is quite easy to perform numerically. Basically, it realizes a combination of informations coming from a first-guess (6H forecast in most of the cases) and from observations.

The configuration which allows to run CANARI is the 701 one (in the ARPEGE world, numbers 700 to 799 are reserved for non-variational analysis configurations, but only one is known !).

The information read or written by CANARI is available in ARPEGE/ALADIN format files, except for the observations which are stored in a database called ODB.

CANARI is able to run in ARPEGE or in ALADIN configuration; in the CANARI code itself, there is no specific routine for ALADIN, all the differences versus ARPEGE are controlled by the logical key LELAM. But of course ALADIN routines are used to define the geometry and other specifities in the top level initialization and to deal with any part of the code which involves geometry.

The fact that CANARI is included in the IFS/ARPEGE/ALADIN library means it uses common parts of the code as much as possible (setups, input files reading, output files writing, various transformations on the different fields, departures calculations,...). But a large amount of routines is only used by CANARI (observations selection, linear system definition and inversion, ...); all these routines begin by CA (QA for the commons and NA for the namelists).

There are two major parts in CANARI, executed after the general initializations and before some final diagnostics and statistics; the study of the observations quality in order to select only the pertinent ones (flags update) and the proper analysis of the different fields.

As most of the configurations, it is possible (and recommended) to run CANARI with a command line, which avoids to describe a lot of variables in the different namelists read by the job. This command line is :

EXE -c$NCONF -v$VERS -m$MODEL -e$CNMEXP -t$TSTEP -f$NSTOP -a$ADVEC

where EXE is the name of the executable library and the various options correspond to :

c : number of the wanted configuration (**701** for CANARI),
v : version of the code (**meteo** for ARPEGE and ALADIN),
m : global or limited area model (**aladin** for ALADIN or nothing for ARPEGE),
e : name of the experiment in 4 characters (**ANAL** for example),
t : timestep (no matter for CANARI, **1.** usually, avoid 0.),
f : integration duration, in hours or in timesteps (**t0** for CANARI),
a : wanted dynamical scheme, eulerian or semi-lagrangian, no matter for CANARI (**sli** as usual).

2. What does CANARI perform ?

2.1. Some non-technical reminders

It is well known (see OI scientific documentation for more details) that the formulation of the optimum interpolation is :

$$X_a = X_g + PH^t(O+HPH^t)^{-1}(HX_g\text{-}Y)$$

where X is the state of the atmosphere and Y the observations, O is the variance-covariance matrix of the observations errors, P is the variance-covariance matrix of the model errors (first-guess errors), H the observations operator which is supposed to be linear.

So the role of the code consists in estimating the analysis increment by calculating on one hand HX_g-Y which corresponds to the observed departures (difference between each observation and the model equivalent at the same location), and on the other hand by building the two matrixes and solving the linear system involved.

Pratically, the analysis is performed for each gridpoint at each vertical level and this point by point, sequentially. And as it is impossible to solve a linear system including all the observations for all the gridpoints, the global system is divided in n smaller systems (where n is the number of gridpoints), each built with the best observations around the gridpoint; so the size of the systems to solve is reduced to a realistic value.

As H is a linear operator, an OI analysis can not make use of satellite radiances; so only conventional data are integrated by CANARI.

CANARI performs a multi-variate analysis of upperair variables (temperature, wind and relative humidity) and of surface pressure. It realizes a mono-variate analysis of surface fields (snow and SST (sea surface temperature)) and of boudary layer fields too (2 meters temperature and relative humidity, 10 meters wind) in order to initialize the surface fields over land for the model. Indeed, as we do not have any observation of these surface fields, surface temperature and water content can not be analyzed; they are derivated from two meters temperature and relative humidity analysis increments.

2.2. CANARI organization

The first step in CANARI is to define the quality of the observations (as any analysis system) in order to avoid the use of any incorrect parameter. A crude test is performed over each message (latitude, longitude, altitude and so on coherence basic check). It is possible to use a blacklist to define some observations which have systematically to be rejected. The observed departures for all the parameters of all the observations are then calculated; they will be usefull in the second part and they allow here to reject "wrong" observations, those for which the difference with the first-guess is too big (previous definition of a confidence threshold). Another control is performed, a spatial one, which permits to make up for some observations which are far from the first-guess if they are several in this case and close enough. Then the observations (and the various parameters that they include) are flagged according to an advanced mixing of all the previous checks results; and the analysis will use only the parameters with a positive quality flag.

The second step in CANARI is to perform the analysis itself. This is done sequentially gridpoint by gridpoint. So there is a loop over the gridpoints, and for each gridpoint all the variables are analyzed at each vertical level (if necessary only of course) one by one. This means that for any point *i*, are analyzed the surface pressure, then the wind and the temperature (level by level), then the relative humidity (level by level too), then the 2 meters temperature, then the 2 meters relative humidity, then the 10 meters wind, then the snow height. Finally, the surface fields are derivated from boundary layer or surface parameters and some climatological relaxation is applied to simulate their seasonal cycle in a realistic way.

3. The observations

3.1. Data description

In the ARPEGE/IFS code, the observations are classified in 10 types, according to the original messages read in the meteorological databank. So is coded CANARI, but mainly 6 of them are usefull, due to the intrinsic content of these classes.

3.1.1. *SYNOP*

In this first class are available the SYNOP, SHIP, SYNOR and RADOME messages, including automatic observations. The measured parameters are : surface pressure, 2 meters temperature and relative humidity, 10 meters wind, precipitations, snow height, and sea surface temperature when it is possible.

3.1.2. *AIREP*

This class corresponds to the aircraft messages, named AIREP, AMDAR or ACAR. The available measurements are pressure (or altitude flight), wind and temperature.

3.1.3. *SATOB*

As indicated by its name, this class contains all the SATOB messages. They provide pressure, wind and temperature data, derived from the geostationnary satellite imagery.

3.1.4. *DRIBU*

This observation type includes all the BUOY, DRIFTER, BATHY and TESAC messages, send by drifting buoys or some ships. The measured parameters are : surface pressure, 2 meters temperature, 10 meters wind and sea surface temperature.

3.1.5. *TEMP*

This class includes all the radiosondes messages, this means TEMP, TEMPSHIP, TEMPMOBIL and TEMPDROP ones. They provide on a lot of levels pressure, wind, temperature and humidity, and for some of them "surface" measurements as 10 meters wind, 2 meters temperature and humidity, even sea surface temperature if it is possible.

3.1.6. *PILOT*

This observation type includes all the PILOT messages, PILOT, PILOTSHIP, PILOTMOBIL, PROFILER and EUROPROFIL. The only measured parameter is the wind at several levels (with the corresponding height of course), and sometimes the 10 meters wind.

3.1.7. *SATEM*

In this class are available the SATEM, the SSMI, the TOVS and ATOVS messages. SATEM data contain humidity and temperature vertical profiles (14 layers), retrieved from radiances measurements. It is the only message that CANARI can use. The other messages contain pre-processed radiances (clarified data) or even raw radiances (they were expected to be in the tenth class, but the code is like that).

3.1.8. *PAOB*

This class corresponds to the PAOB messages, which contain some forced observations of pressure (by a forecaster in case of typhoon for example, derived from satellite imagery).

3.1.9. *SCATT*

This observation type corresponds to all the messages sent by the different existing scatterometers. They provide surface wind measurements.
There is no observation of this type currently.

3.1.10. *RAWRA*

This class has been introduced at the beginning of ARPEGE/IFS to contain the raw radiances, but it is empty because they are really stored in the seventh class. So CANARI knows this type; but it is sure now that the use of raw radiances in the OI will never be investigated. The structure is still maintained for compatibilty with the variational analysis code.

3.2. Data organization

To be used by ARPEGE/IFS, and of course by CANARI, the observations have to be stored in a database called ODB. This concept has been introduced in cycle 23. This means that the data are not all available in the central memory during the whole program execution, but only the needed parameters are taken from the base and kept into memory according to the part of the code which is running. This is done by special sql requests which have been implemented in strategic parts of CANARI.
For each conventional data, the observation is divided in two parts : the first one which contains all the information about the observation itself (date, time, type, latitude, longitude and so on), and the second which describes the measured parameters themselves.
For more details, see the ODB documentation written by D. Puech [1].

3.3. Database filling

Before the execution of the analysis, the ODB contains the values of the observed parameters, their spatial and temporal location, the adequate observation errors statistics and some quality flags put by some preprocessing tests done by the meteorological databank software itself.
After the analysis, a lot of additional information is introduced, the observed and analyzed departures, the first-guess errors statistics, the various flags updated by the control and the analysis.
If someone wants to change the list of the parameters needed in the database (reading or writing access), he has to change the sql request called by the appropriate subroutine, then he has to compile

the ODB library, and of course load the ARPEGE library again because it is linked with ODB library.

4. Input and output files

4.1. The first-guess file

This input file has to be an ARPEGE/ALADIN format file, this means with all the fields read by the model and as expected by the code (spectral or gridpoint field). For more details about this format and the associated nomenclature, see the documentation written by J. Clochard (ARPEGE Notes n° 12 and 17, updated later [2]).

Before the execution of the analysis, this file has to be linked with **ICMSHxxxxINIT** for an ARPEGE configuration and **ELSCFxxxxALBC000** for an ALADIN configuration, where xxxx is the name of the experiment (as defined by $CNMEXP in the command line). It corresponds to the initial state of the model and it is indispensable for the job (ABORT if not present).

4.2. The observations database

To allow CANARI to use the ODB, a special environment has to be defined by setting some variables in the script :

```
export ODB_STATIC_LINKING=1
export TO_ODB_ECMWF=0        (for ARPEGE/ALADIN case)
export ODB_CMA=yCMA    with y=E (usual case) or y=C (for compressed obs database)

export ODB_SRCPATH_yCMA= ...       }  path where to find the different tables
export ODB_DATAPATH_yCMA= ...   }    of the database

export IOASSIGN= ...     path of a file containing different information about the tables structure
```

In a debugging phase, it is possible to add : export ODB_TRACE_FILE=listing_odb
and after the analysis, the "listing_odb" file will contain some information about the different accesses to the database, the volume of extracted data, the calling subroutines and so on.

Concerning the observations use, another file is necessary, called **rszcoef_fmt**, but it is without any interest for CANARI (just a part of the variational analysis code which is not controlled by a logical key !). This file is not linked with the ODB concept.

4.3. The climatology files

The climatology file corresponding to the month of the run date is necessary if a climatologic relaxation has to be applied (according to the associated namelist switch). Anyway, this file has to be present the first day of every month in order to update the monthly surface constants of the model (vegetation, albedo and so on). This file is used too in case of snow analysis, in order to estimate the observed departures.

It has to be linked with the name **ICMSHxxxxCLIM** (xxxx is still the experiment name).

The climatological constraint is applied on surface fields only (temperature and water content). It is possible to use a second file (which has to be linked with **ICMSHxxxxCLI2**) in order to improve the quality of the climatological fields by a temporal interpolation (previous month or later month according to the run date).

It is obvious that these two files have to be on the same grid than the model one (that is the first-guess grid too).

4.4. The ISBA files

ISBA is the surface scheme used both in the analysis and in the model integration to describe the surface variables evolution. In the analysis, it concerns the surface and soil temperatures, and the surface and soil water contents (see [3] for more details).

Technically, it is necessary to have a file containing the polynomial terms used in the formulation which evaluates the soil moisture; this file has to be linked with **fort.61**.

The use of an assimilated increment file (for both surface temperature and humidity and for the soil water content) in order to smooth the fields and to take into account the diurnal cycle gives better results. This file has to be linked with the name **ICMSHxxxxLISSE**.

This last file is updated after the analysis with the new increments, it is called **ICMSHxxxxLISSEF**. It is necessary to save it at the end of the job to be able to use it in the next analysis in an assimilation cycle.

4.5. The SST file

This file can be used in the ALADIN case if the configuration to produce it is developped, but it is of a very limited interest because it contains a SST field which will be used as a relaxation field for the analyzed sea surface temperature. It has been implemented for the global analysis in order to have an up-to-date "climatological" field because there is a lack of conventional observations in some regions and consequently the SST analysis is not so performant comparatively to the other variables analysis.

The initial data consist in the NCEP SST analysis, performed on a 1°x1° latitude/longitude grid, estimated with both conventional and satellite data. This field is interpolated on the ARPEGE grid and is stored in an ARPEGE format file by the configuration number 941. This last file has to be linked later with the name **ICMSHxxxxSST** to be used by CANARI.

4.6. The errors statistics files

As the optimum interpolation is a method which allows to know the variance of the analysis error, it is possible to use this result in order to improve the quality of the variance of the first-guess error (in an assimilation cycle). This has been coded for the upperair variables (and used when CANARI was operational in Meteo-France for the analysis of these variables). So, the input file, linked with **ICMSHxxxxSTAT**, contains error standard deviation both for the first-guess and the analysis of the previous run (6 hours before) for the geopotential and the humidity fields, in spectral coordinates, on each model level. So CANARI can use "dynamic" statistics instead of fixed ones (but necessary if the input file is missing). And CANARI produces an output field, linked with **ICMSHxxxxSTA2**, which contains the same fields, but for the current run.

4.7. The incremental mode files

It is possible to read three input files to build a non-classical starting state for CANARI; a combination is done on the spectral fields only : $G_1=G_0+A-G$. This is usefull in the incremental mode operational at Meteo-France in order to define a new guess (G_1), based on the initial first-guess G_0 updated by the increments of the upperair analysis calculated at a lower resolution (A-G). G_0 is still linked with ICMSHxxxxINIT, A has to be linked with **ICMSHxxxxANIN** and G with **ICMSHxxxxFGIN**.

This option is not valid for ALADIN, it has been coded for the global case only, for the needs of the variational analysis operational at Meteo-France.

4.8. The analysis file

This output file is called **ICMShxxxx+0000**. It is equivalent to the first-guess file (initial conditions) which has been updated without any time integration (general ARPEGE coding rules). If the program would run without any observation and without any climatological relaxation, the analyzed fields would be the same than those of the first-guess (quite the same in fact because the go and back spectral / gridpoint transformations are applied in any case).

5. Distributed memory features

Only the specific CANARI aspects of the distributed memory architecture will be described in this chapter. To get more details about the general structure of the code (high level code organization regarding distributed memory and message passing), refer to some internal documentation [4]).

It is important to precise here that the upperair fields of the model (input first-guess file) are stored under their spectral form, instead of the surface fields which are gridpoint representated. The analysis is performed over gridpoint fields representation; so spectral arrays are useless (once they have been transformed) except a short part at the beginning of CANARI which is performed only in incremental mode, in order to update the upperair initial fields with an accurate analysis. But it is without any interest in this chapter.

So CANARI uses the ARPEGE subroutine STEPO which manages the gridpoint calculations in the distributed memory context.

All the control prints produced by CANARI are global, this means that all the processors communicate their own results to the first one in all the concerned subroutines before the prints.

5.1. The observations distribution

It is possible to divide the ODB in several pools before the analysis by a special procedure; each pool corresponds to a set of observations that only one processor can access in a basic calling sequence. The observations distribution is done according to geographical criteria, in order to balance the number of observations by type on each pool.

But each processor can access to the whole observations package, and this possibility will be used in CANARI because it is easier in this case to perform any isotropic selection among the observations.

5.2. The departures calculation

In this part, each processor has got in its own memory a list of gridpoints, and a set of observations, which of course do not correspond geographically speaking. A subroutine is executed quite at the beginning of the program in order to fill some global arrays which allow any processor to know the list of the observations which are located near its own gridpoints. So later in the program each processor calculates the model equivalent (involving its own gridpoints) for the list of observations geographically assigned to it. After that some communications between the processors are executed in order to allow each of them to save in its memory the model equivalent of each of its own observations. This model equivalent corresponds to the horizontal interpolation of the model fields. In a second hand, each processor performs the vertical interpolation of the model equivalent for its own observations and fills the corresponding pool of the database with the observed departures and the associated quality flag.

The same subroutines are used to calculate the analyzed departures once the analysis has been performed.

5.3. The spatial quality control

For each observation, this control consists in an analysis performed at the observation point without using this observation. If the difference between the observation value and the obtained analyzed value is smaller than a tunable threshold, the observation will be considered as a good one whatever may be the result of the first-guess check.

So in this part each processor works with a loop over its own observations. But all the observations are read by each processor in order to perform the analysis calculations (see below chapter 5.5 for more details).

This means that each processor opens the ODB and reads the information it needs regarding its own observations. It saves these data in some local arrays and then closes the database. After that, the ODB is opened a second time; each processor reads some listed parameters for all the observations, execute the loop over its own observations; then the ODB is closed again. Once the quality control has been performed, the ODB is opened a third time and each processor updates the database with the results concerning its own observations. Then the ODB is closed for the last time in this part.

Note that the model fields are not used in these calculations.

5.4. Model variables analysis

The analysis itself (that is the analyzed increment calculation for each variable) is performed point by point (as described before in chapter 2.2). So each processor loops over all its own gridpoints. According to the number of gridpoints which has to be considered, in order to increase the efficiency of the code by a better vectorization, the gridpoints are processed by packet (sized by the famous NPROMA). But the final estimation is done point after point, with a loop on the model levels according to the variable which is currently analyzed.

In this part, as the calculation loop is over the gridpoints, the ODB is accessed only one time.

Each processor updates the model fields for each of its own gridpoints, and the database for the observations it has used.

5.5. The proper analysis

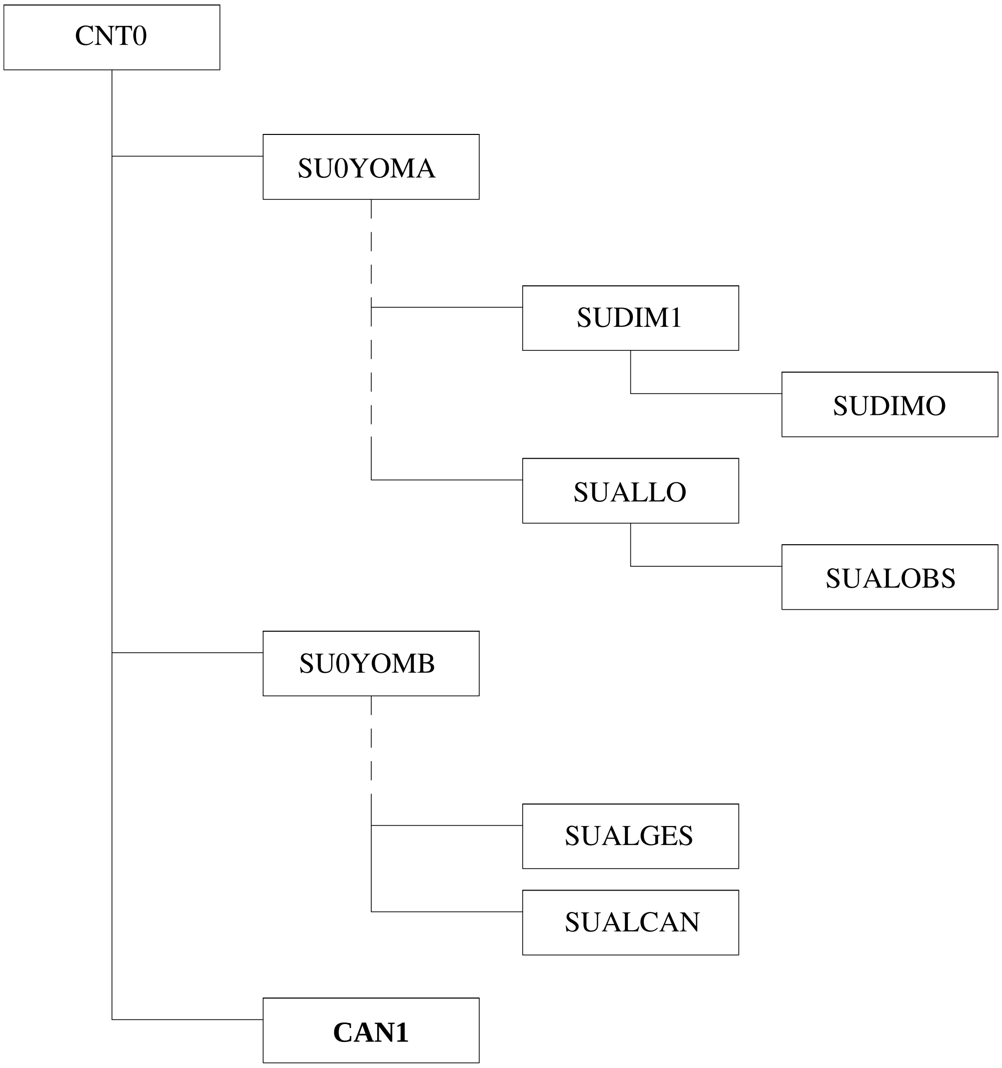
Whether the proper analysis is performed for the spatial quality control or for the analyzed increment estimation, it obeys to the same rules. For each point where the analysis has to be performed, it is necessary to select the more informative observations in order to build the numerical linear system with a reasonable size. So an observations selection has to be done, and the first criterion is of course a geographical one (due to the correlation function which is a gaussian curve). To ensure the reproductibility of the analysis whatever may be the number of used processors, it has been decided to allow each processor to access to all the observations. Once each processor has in its own memory all the information it needs regarding all the observations, there is no specific treatment due to the distributed memory architecture.

6. Code description

6.1. Initialization and level 0 control

The initialization and the top level control (level 0) of CANARI are done through some subroutines which might be ARPEGE/ALADIN general or CANARI specific.
The 701 configuration starts with the subroutine CAN1, but before its execution the program performs a lot of initializations which are the same for most of the configurations. Only the subroutines where there are specific CANARI instructions will be described.

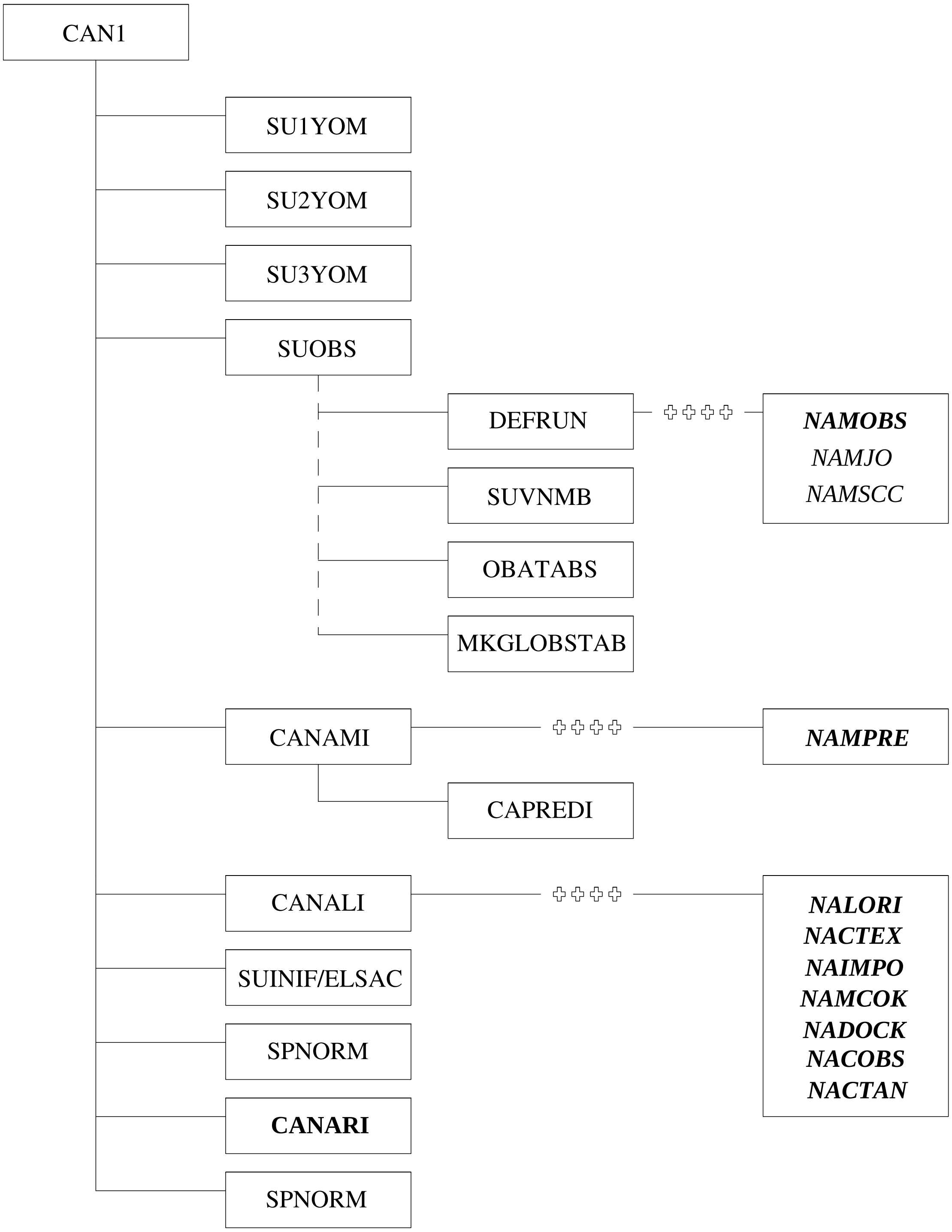
See below the calling tree for CAN1 (the main program is MASTER, which calls CNT0) :



SU0YOMA calls SUDIM1 which sets the dimensions for most of the model arrays and then calls SUDIMO which do the same thing but for the observations related arrays. SUALLO is called later; it performs the allocation of the global arrays for the model and calls SUALOBS which allocates the observations related arrays (both for variational and OI analyses).

SU0YOMB calls another set of subroutines in order to complete the definition of the general environment, including SUALGES which allocates the first-guess errors statistics arrays and SUALCAN which defines some CANARI specific arrays.

6.2. CANARI initialization and level 1 control



Before starting the OI with the subroutine CANARI, the level 1 control routine CAN1 calls some other subroutines to initialize some additional model common parameters, specific ones and to read the fist-guess file.

SU1YOM : YOMCT1 setup, for the trajectory and diagnostics management;

SU2YOM and SU3YOM : YOMCT2 and YOMCT3 setup respectively, both for the integration management;
SUOBS : setup for the observations processing (cf. §6.2.1);
CANAMI : setup of the predictors/predictands list for OI (cf. §6.2.2);
CANALI : setup of the main OI parameters (cf. §6.2.3);
SUINIF : input file reading (SUINIF for ARPEGE and ELSAC for ALADIN) (cf. §6.2.4);
SPNORM : computation of spectral fields norms and control prints;
CANARI : main OI subroutine (cf. §6.3).

6.2.1. Setup for the observations processing

The control routine SUOBS calls a set of subroutines which are usefull only for the analysis configurations. Among them, only those which are of interest for CANARI are described.

DEFRUN : assigns default values to the run control parameters and reads the associated namelists ***NAMOBS***, *NAMJO* and *NAMSCC*; only ***NAMOBS*** is usefull for CANARI. The variables to set in NAMOBS are LCAPACH, LCACHMT, LSLREJ and NOBSHOR. LCAPACH and LCACHMT are the logical keys for the use of the ARPEGE (instead of IFS) observation operators (default is .TRUE. so OK). This means that when calculating the model equivalent (vertical part) for an upperair parameter, on one hand (first key) APACH will be called (use of the model orography in order to modify the pressure of the observed parameter and to realize an accurate vertical interpolation) and for a surface parameter, on the other hand (second key) ACHMT will be called (use of the model surface characteristics to define the boundary layer before vertical interpolation). LSLREJ is the logical key which controls the use of the land/sea mask of the model for the horizontal interpolation of the surface fields when calculating the model equivalent for a given observation in CANARI. The default is .FALSE., it is better to set .TRUE. NOBSHOR corresponds to the method for the guess horizontal interpolation, 201 for a bilinear one (forced for the surface fields), 203 for a bicubic one (default value).

SUVNMB : sets the observation variable code number.

OBATABS : first chek of the ODB and calculation of the latitude and longitude coordinates of the observations on the model grid (according to its geometry).

MKGLOBSTAB : creates global observations tables needed in the message passing of model equivalent computation.

6.2.2. Predictors and predictands definition

The subroutine CANAMI realizes the predictors and predictands definition both for the quality control and the analysis, then reads the namelist ***NAMPRE*** and calls CAPREDI which checks the NAMPRE inputs according to the defined statistical models and updates the local predictors/predictands tables.

List of the predictands classes, with the list of the allowed associated predictors, and the corresponding code numbers which have to be set in ***NAMPRE*** (saved in the module QAPREX) :

a) spatial quality control case :
- Z, u and v (geopotential and wind) : u, v, geopotential, u_{10}, v_{10} [3, 4, 1, 41, 42]
- T (temperature) : T, T_2 [2, 39]
- Hu (relative humidity) : Hu, PWC, H_2 [29, 9, 58]
- Dz (layer thickness) : geopotential, Dz, T [1, 57, 2]
- SST (sea surface temperature) : SST [11]
- Sn (snow) : snow height [92]
- QQ (specific humidity) : QQ [7]
- T_2 (2 meters temperature) : T_2 [39]

H_2 (2 meters relative humidity) : H_2 [58]
V_{10} (10 meters wind) : u_{10}, v_{10} [41, 42]

b) analysis case :
Ps (surface pressure) : u, v, geopotential, Dz, T, u_{10}, v_{10} [3, 4, 1, 57, 2, 41, 42]
T : u, v, gepotential, Dz, T, T_2, u_{10}, v_{10} [3, 4, 1, 57, 2, 39, 41, 42]
u and v : mixed analysis with T analysis
Hu : Hu, PWC, H_2 [29, 9, 58]
T_2 : T_2 [39]
H_2 : H_2 [58]
V_{10} : u_{10}, v_{10} [41, 42]
Sn : snow height   [92]
SST : SST   [11]
QQ : QQ   [7]

6.2.3. Setup of the main OI parameters

The main parameters which define and organize the CANARI run are distributed among a few common modules according to the part that they control. All these parameters are initialized in CANALI, which later reads the corresponding namelists and controls them.

NACTEX : controls the different steps of the OI analysis. All the variables of the associated module QACTEX are tunable by the namelist. They are :

LAEOMF : calculation of the observed departures	}	
LAEOMN : calculation of the analyzed departures	}	
LAECHK : execution of the spatial quality control	}	
LAEPDS : execution of the surface pressure analysis	}	logical variables
LAEUVT : execution of the upperair wind and temperature analysis	}	
LAEHUM : execution of the upperair relative humidity analysis	}	default value
LAET2M : execution of the 2 meters temperature analysis	}	equal to
LAEH2M : execution of the 2 meters relative humidity analysis	}	.FALSE.
LAEV1M : execution of the 10 meters wind analysis	}	
LAESNM : execution of the snow analysis	}	
LAESST : execution of the sea surface temperature analysis	}	
LAEICS : calculation of the surface fields	}	
LAECDS : calculation of the surface diagnostic fields	}	
LAEINC : incremental mode (updating of the guess)	}	
LAESTU : use of the assimilated errors statistics (previous run)	}	
LAESTA : saving of the analysis error statistics	}	
LAEWIO : writing of the analysis output file	}	
LAERFO : updating of the ODB	}	
LVERAL : no initialization of the quality flags	}	

NAEINC : updated word in ODB by the observed departures (integer)
0 ---> OMF   1 ---> OMF + FC1   2 ---> OMN   default = 0
RCLIMCA : relaxation coefficient for the land surface fields (real)   default = 0.0
RCLISST : relaxation coefficient for the sea surface temperature field (real)   default = 0.0
NSSTLIS : use of the NCEP SST in the relaxation field (integer)   default = 0 --> no use
(corresponds to the number of authorized days late)

NACTAN : defines the analysis area (it can be smaller than the guess domain thanks to a special mask field). It allows to modify some variables of QACTAN. They are :
LANMASK : .TRUE. analysis performed on a reduced domain (default=.FALSE.)

ALATNB : northern latitude of the analysis domain }
ALATSB : southern latitude of the analysis domain } real values
ALONWB : western longitude of the analysis domain } default corresponding to the
ALONEB : eastern longitude of the analysis domain } first-guess domain

NACOBS : sets up some observations related variables, which belong to the modules QACOBS (OROLIM and ORODIF) or QAPABO (NBODLA and NBODLO).
OROLIM : maximum obs altitude for a SYNOP use (default=10000 m)
ORODIF : maximum difference allowed between a SYNOP altitude and the corresponding model orography (default=10000 m)
NBODLA : box size in latitude for the observations arrangement in memory
NBODLO : box size in longitude for the observations arrangement in memory
(default=10 for ARPEGE, 5 for ALADIN).

NADOCK : defines the observations selection criteria, included in the module QADOCK. The variables are :
NMXGQA : maximum number of observations by quadrant (integer array, size JPNOTP)
QDSTRA : maximum distance for the horizontal selection (en m) (real array, size JPNOTP)
QDSTVA : maximum distance for the vertical selection (en hPa) (real array, size JPNOTP)
MINMA : predictors number by predictand (integer array, size JPANAL)
QCORMIN : minimum correlation for the selection by predictand (real array, size JPANAL)
QDELPI : minimum distance between 2 selected levels of one observation
(real array, size JPANAL)
where JPNOTP is the number of observations types and JPANAL the number of variables to be analyzed (parameters defined in the modules PARDIMO and QAPREX respectively).

NAMCOK : list of the rejection thresholds for the quality control various steps, which allows to fill the array RCOEFK of the module QAMCOK. Thirteen thresholds are defined, under the form RCxxyy where xx is the measured parameter and yy the observation type.

NALORI : contains the parameter RCALPH only, included in the module QALORI which describes some smoothing functions. The default for RCALPH is zero. It corresponds to the coefficient of the function used to take into account the stretching of the grid in the estimation of the correlations. This function is $\exp [a.(x-1/x)]$ where a=RCALPH and x is the local stretching factor.

NAIMPO : controls some observations related prints during the execution of CANARI. The variables are included in the module QAIMPO. They are :
LAVISO : prints at the end of the CANARI run;
LAVOLO : prints after all the initialisations concerning the observations;
LAVORO : prints after the observed departures calculations;
LAVOCQ : prints after the spatial quality control and the update of the quality flags;
LAVOAN : prints after the analysis of the various variables;
NBSYVO : number of SYNOP observations to visualize (default=5);
NBAIVO : number of AIREP observations to visualize (default=5);
NBSBVO : number of SATOB observations to visualize (default=5);
NBDRVO : number of DRIBU observations to visualize (default=5);
NBTPVO : number of TEMP observations to visualize (default=3);
NBPLVO : number of PILOT observations to visualize (default=3);
NBSTVO : number of SATEM observations to visualize (default=5);
NINDVO : 0 ---> prints without any observation array modification,
1 ---> prints with the usual units (instead of SI) for the parameters;
NBCLVO : number of columns for the prints (80 or 132 (default)).

6.2.4. Initial conditions file reading

The initial conditions file is read in almost all the ARPEGE configurations, by a package of common subroutines, led by SUINIF in the ARPEGE case and by ELSAC in the ALADIN case. This file is read in two steps, the first for the spectral fields and the second for the gridpoint fields. So all the arrays of the model are filled, SPA3 by the spectral 3D fields (vorticity, divergence, temperature and specific humidity) and SPA2 by the spectral 2D fields (surface pressure and orography) which belong to the module YOMSP, GPPBUF which is the gridpoint buffer of the physical fields, included in the module YOMGPPB.

This routines package is used to read the climatology file(s) just after the guess file; these new fields are stored in GPPBUF too.

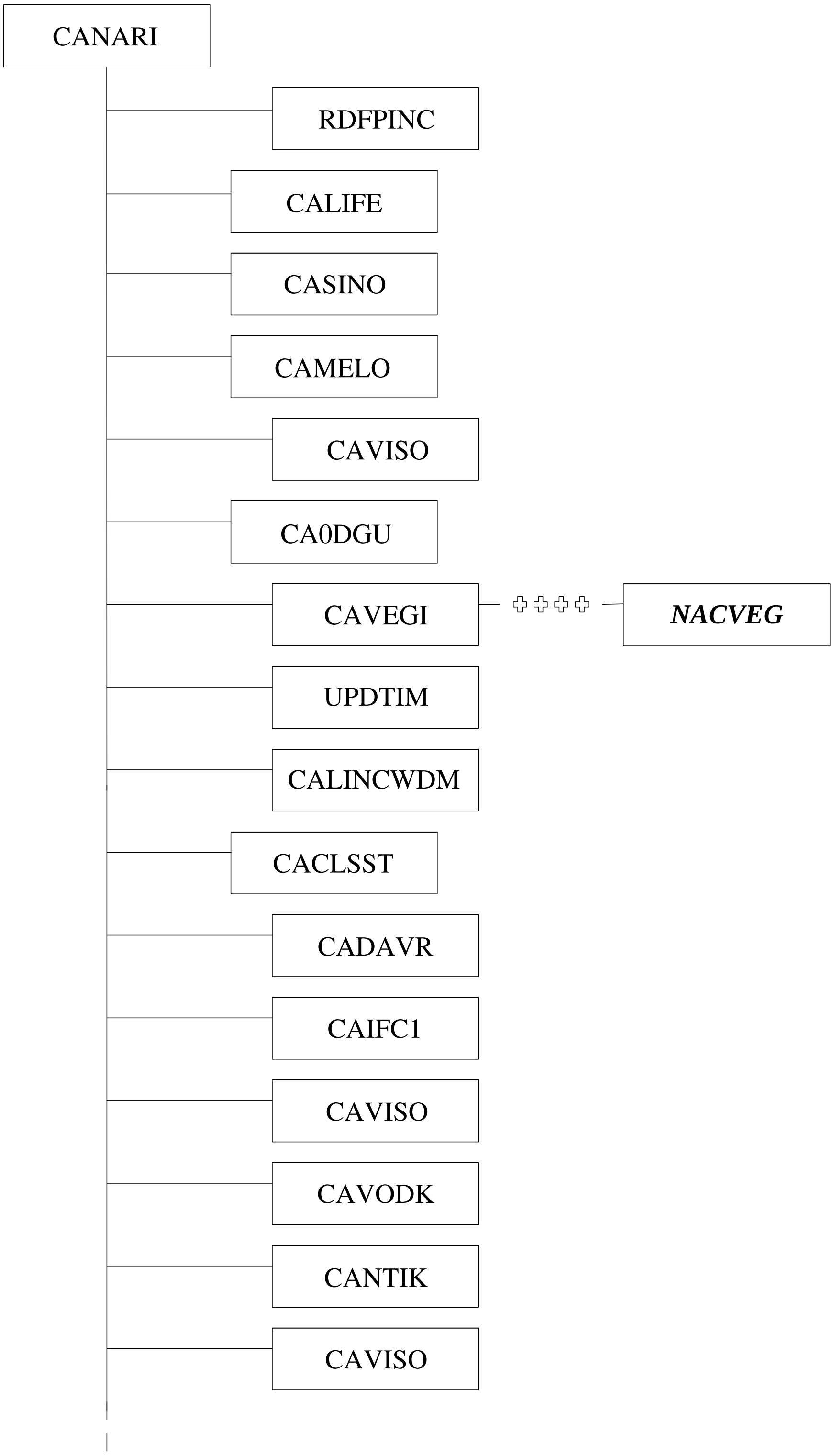
It is important to mention here that there is a block in GPPBUF which is dedicated for some CANARI specific fields. This block contains NVCAN fields (NVCAN=13 today), the last five being for the ISBA increments; it is pointed by MCANRI0. These fields are :

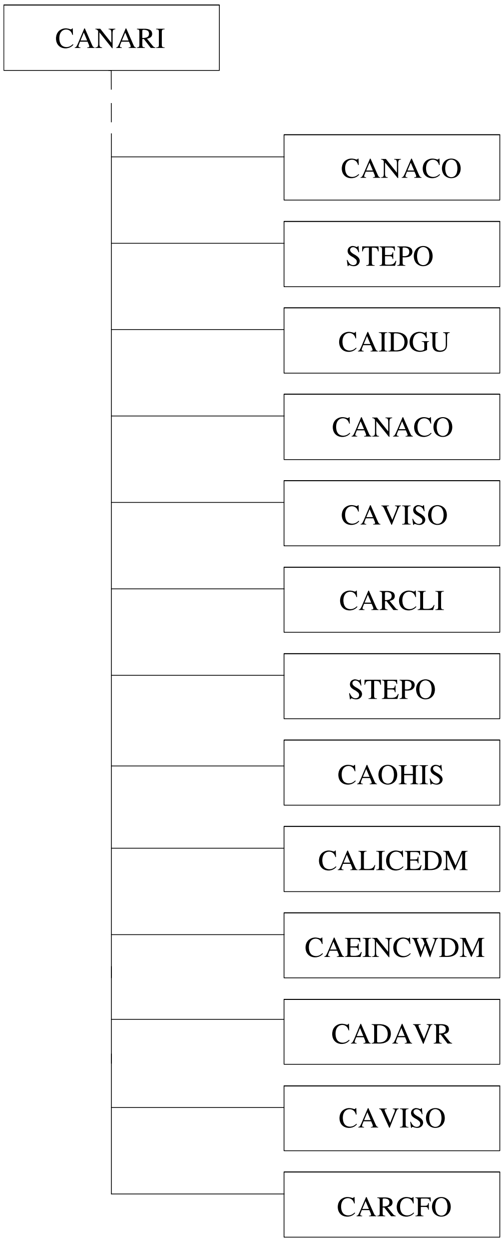
- first-guess error standard deviation for the geopotential field	SURFETP.GEOPOTEN
- analysis error standard deviation for the geopotential field	SURFETA.GEOPOTEN
- analysis mask (1 --> perform analysis at this gridpoint, 0 --> do not)	SURFANALYSISMASK
- 2 meters temperature field	CLSTEMPERATURE
- 2 meters relative humidity field	CLSHUMI.RELATIVE
- zonal component of the 10 meters wind field	CLSVENT.ZONAL
- meridian component of the 10 meters wind field	CLSVENT.MERIDIEN
- final relaxation field for the SST field	SURFSST.CLIM.
- soil water content increment of the previous run	PROFINC.RESERV.1
- idem but 2 runs old	PROFINC.RESERV.2
- idem but 3 runs old	PROFINC.RESERV.3
- 2 meters temperature mean increment	SURFINC.TEMPERAT
- 2 meters relative humidity mean increment	SURFINC.HUMIDITE

These fields are stored in GPPBUF after the execution of SUINIF/ELSAC; if they are missing from the initial file, they are set to 0 (1 for the analysis mask). In fact, at this stage, only the four ’CLS’ fields are filled.

6.3. CANARI management main subroutine

The CANARI subroutine is the starting point of the analysis code itself. It organizes the different steps of the work in logical consecutive tasks, as follow (the routines which are in a box slightly shifted to the left are unconditionally called, while the others are controlled mainly by the variables of the namelist NACTEX) :


CANARI
RDFPINC
CALIFE
CASINO
CAMELO
CAVISO
CA0DGU
CAVEGI
NACVEG
UPDTIM
CALINCWDM
CACLSST
CADAVR
CAIFC1
CAVISO
CAVODK
CANTIK
CAVISO



The first thing done, according to the value of LAEINC, is the updating of the first-guess fields in the incremental mode by RDFPINC (see §4.7 for information, but this subroutine will not be described in detail in this documentation because it is not directly linked to CANARI).

Then some indispensable initializations for the analysis are performed, on one hand about the model errors statistics through the subroutine CALIFE (cf. chapter 7) and on the other hand about the observations by the subroutines CASINO and CAMELO (cf. chapter 8).

CAVISO is a subroutine which reads the ODB and prints the observations arrays in order to control the good behaviour of CANARI as the run progresses; it is called several times according to the logical keys of the namelist NAIMPO (described in §6.2.3).

CA0DGU is the subroutine which allocates and initializes some arrays (module QAPDGU) dedicated to the print of statistics about the analyzed fields (mean, standard deviation and extrema values of the fields themselves, analysis increments, mean size of the linear systems, mean analysis error). These arrays will be filled during the execution of CANARI through the subroutine CACDGU and printed at the end by the subroutine CAIDGU. This is done in order to have a first control about the quality of the analyzed fields.

Then some initializations concerning the surface fields are performed, only if they have of course to be updated (according to LAEICS), in order to prepare the ISBA action. This is done by the subroutine CAVEGI, which reads the polynomial terms file (described in §4.4), initializes the variables of the module QACVEG, reads and controls the associated namelist ***NACVEG***, checks that the first-guess file is coherent with the assimilated increments file and finally prints the QACVEG variables. The subroutine UPDTIM is called after CAVEGI in order to initialize the time of the run and to update the solar constants because they are used later by ISBA. Then, if the smoothing of the analyzed increments is wanted (LISSEW=.TRUE. in *NACVEG*), the subroutine CALINCWDM is called in order to read the increments fields in the assimilated file (described in §4.4). These fields are stored in GPPBUF (last five fields in the CANARI block). The variables of *NACVEG* are not detailed because they are linked to ISBA (and not to CANARI) ; only a physics specialist can tune them.

After that, the relaxation field for the calculation of the definitive SST field is prepared by the subroutine CACLSST; it controls the corresponding keys in *NACTEX* (NSSTLIS and RCLISST), reads the NCEP SST file (described in §4.5), modifies the climatological surface temperature field with the NCEP field over the free-ice sea and stores this new field in GPPBUF (SURFSST.CLIM. field) if everything is correct.

Then, in CANARI, the analysis mask is updated according to the variables of QACTAN.

Now, the effective work of the analysis can begin.

Under the key LAEOMF, the observed departures calculation is performed (cf. chapter 9) by the subroutine CADAVR. CAIFC1 is called according to the value of NAEINC in order to duplicate the observed departures in the ODB.

The next step is the spatial quality control, performed by the subroutine CAVODK (cf. chapter 10) under the control of LAECHK.

The quality flags of the observations are then updated by CANTIK (cf. chapter 11) according to the key LVERAL. CANTIK performs too a lot of control prints about the validity of the observations, parameter by parameter.

At this stage, all is ready to perform the analysis itself, gridpoint by gridpoint. So the subroutine CANACO is called in order to read all the needed observations parameters. After that, the subroutine STEPO is called in order to manage the analysis steps (cf. chapter 12). Then CANACO is called again in order to update some observation flags in the ODB and to close it after.

CARCLI is executed if the run date is the first day of the month in order to update the monthly fields of the model (surface "constant" fields).

Then the output file is written, by another call to STEPO (configuration 'A00000000').

According to the value of LAESTA, the analysis errors standard deviation fields are computed and saved in the appropriate file ICMSHxxxxSTA2 (see §4.6) by CALICEDM (previous call to CAOHIS in order to open the output file)..

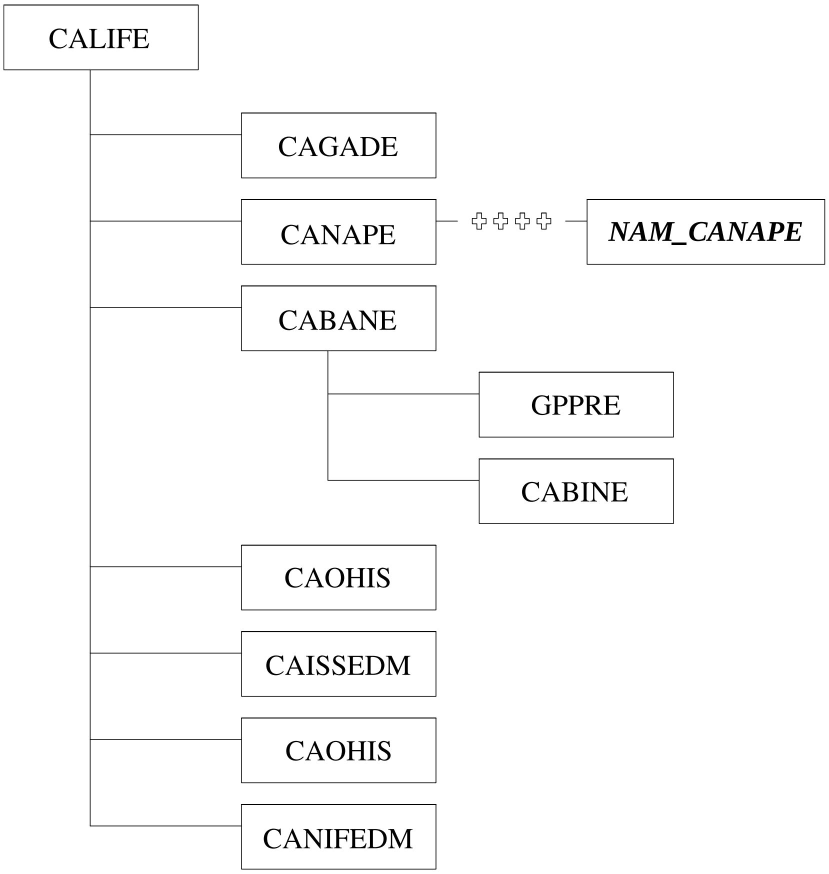
Then the file ICMSHxxxxLISSEF (see §4.4) is filled with the new surface analysis increments by CAEINCWDM.

CADAVR is called again in order to calculate the analyzed departures under the control of LAEOMN.

Finally the ODB is updated by all the new information about the observations by the subroutine CARCFO.

7. First-guess check and model errors statistics initialization

The subroutine CALIFE checks the validity of the first-guess file and initializes the model errors statistics as follow :



The subroutine CAGADE checks the agreement between the date of the first-guess file and the date of the run (ABORT if not) and the validity of the guess type.

The subroutine CANAPE initializes various coefficients which will allow to define the statistical model of the analysis. These coefficients are included in the module QAREF and are the reference values for this statistical model : pressure levels, model error standard devation for some basic fields (both surface and upperair levels), horizontal and vertical correlation lengthscales, boundary layer representation (ageostrophism and vertical extent coefficients, nu and Kp). Then the namelist ***NAM_CANAPE*** is read in order to modify QAREF if necessary (description at the end of this chapter).

After that, CABANE is called to initialize the statistical model itself (variables of the module QACOSS) from QAREF values. GPPRE is called to calculate the pressure of the model levels and CABINE performs the vertical interpolation of all the statistical model parameters from the reference levels to these pressure levels. At this stage, a basic statistical model is defined for the present analysis (included in QACOSS for the correlation functions and in QASSET for the model errors standard deviations).

Then, if it is set in namelist that the use of assimilated statistics is needed (key LAESTU), the corresponding file is opened (ICMSHxxxxSTAT, cf. §4.6) by CAOHIS. The subroutine CAISSEDM reads the geopotential and humidity error standard deviation fields, transforms them from their spectral form to a gridpoint one, derives guess statistics from these analyzed ones and applies an horizontal function and a vertical one in order to smooth these fields. After that, the array ESIG (module QASSET) is definitively filled with the accurate model errors statistics.

According to the key LAESTA, CAOHIS is called again to open the output statistical file (ICMSHxxxxSTA2, cf. §4.6). CANIFEDM transforms the model errors fields in spectral coordinates and stores them in the output file ICMSHxxxxSTA2.

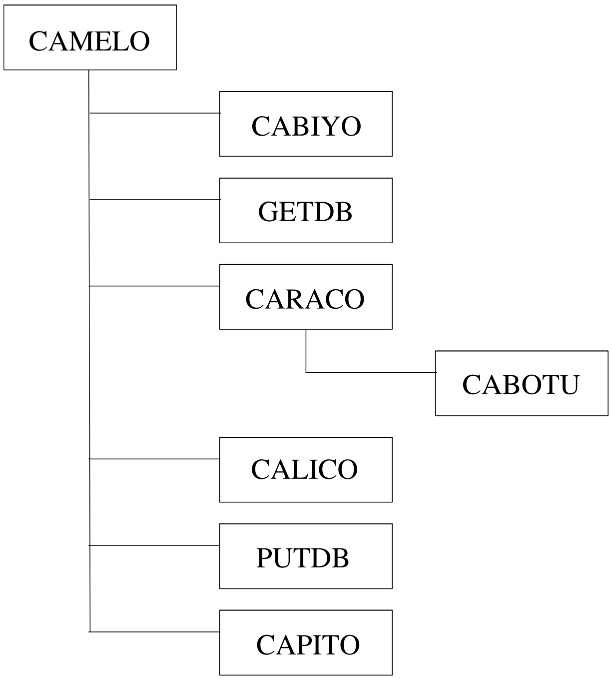
Detailed description of ***NAM_CANAPE*** (cf. annex 2) :

REF_STAT (JPNVS,7) : array containing reference values for various parameters on the JPNVS levels of the basic statistical model (JPNVS=19, module QAPASS) :
REF_STAT (.,1) : pressure of the JPNVS levels,
REF_STAT (.,2) : geopotential error standard deviation,
REF_STAT (.,3) : temperature error standard deviation,
REF_STAT (.,4) : wind error standard deviation,
REF_STAT (.,5) : relative humidity error standard deviation,
REF_STAT (.,6) : vertical lengthscale,
REF_STAT (.,7) : horizontal lengthscale,
REF_S_SST : model error standard deviation for the sea surface temperature,
REF_S_SN : model error standard deviation for the snow,
REF_S_T2 : model eror standard deviation for the 2 meters temperature,
REF_S_H2 : model eror standard deviation for the 2 meters relative humidity,
REF_S_V1 : model eror standard deviation for the 10 meters wind,
REF_A_SST : reference horizontal lengthscale for the sea surface temperature,
REF_A_SN : reference horizontal lengthscale for the snow,
REF_A_T2 : reference horizontal lengthscale for the 2 meters temperature,
REF_A_H2 : reference horizontal lengthscale for the 2 meters relative humidity,
REF_A_VOR1 : reference horizontal lengthscale for the 10 meters wind vorticity,
REF_A_DIV1 : reference horizontal lengthscale for the 10 meters wind divergence,
REF_AP_SN : reference vertical lengthscale for the snow (due to orography),
REF_NU_BL : ageostrophism coefficient in the boundary layer,
REF_KP_BL : vertical extent coefficient for the boundary layer,
REF_PHUD : ratio of the reference horizontal lengthscales for divergence and geopotential,
REF_PHHU : ratio of the reference horizontal lengthscales for relative humidity and geopotential,
REF_PVH : ratio of the reference vertical lengthscales for relative humidity and geopotential,
REF_COEFN : coefficient for the geopotential standard deviation in the latitudes [+25,+90],
REF_COEFT : coefficient for the geopotential standard deviation in the latitudes [-15,+15],
REF_COEFS : coefficient for the geopotential standard deviation in the latitudes [-90,-25],
REF_MU[1,2,3] : used in (wind divergence standard deviation / wind vorticity standard deviation)
μ(p) = (REF_MU1 - max(0,REF_MU2*(p-REF_MU3)/(100000-REF_MU3)))*sin(lat)
REF_NU[1,2,3] : used in the formulation of the ageostrophism coefficient :
ν(p)=(REF_NU1+max(0,REF_NU2*(p-REF_NU3)/(100000-REF_NU3)))*max(cos(lat),.5*sqrt(3))

8. Observations processing

When the subroutine CANARI is running, the ODB validity has been checked and the general environment for the observations use has been defined. But it is necessary to set some other parameters to allow CANARI to use these observations.

CASINO performs some basic initializations and CAMELO is called later and we have :



CABIYO performs the initialization of the module QAETEO, i. e. the observation error standard deviation for each measured parameter of each observation type.

Then the ODB is accessed by the specific subroutine GETDB (with the specific sql request for the needs of CAMELO); this allows to fill the observations arrays with the adequate values.

CARACO fills the array QCAOBS (from the module QALORI) with the coefficient to apply to the horizontal lengthscale in order to take into account the local stretching factor of the grid. Then it calls CABOTU which assigns to each observation the number of the geographic box where it is located (these boxes have been defined by CASINO).

CALICO updates the adequate observations array with the observation error standard deviation and initializes the quality flags for each parameter of each observation.

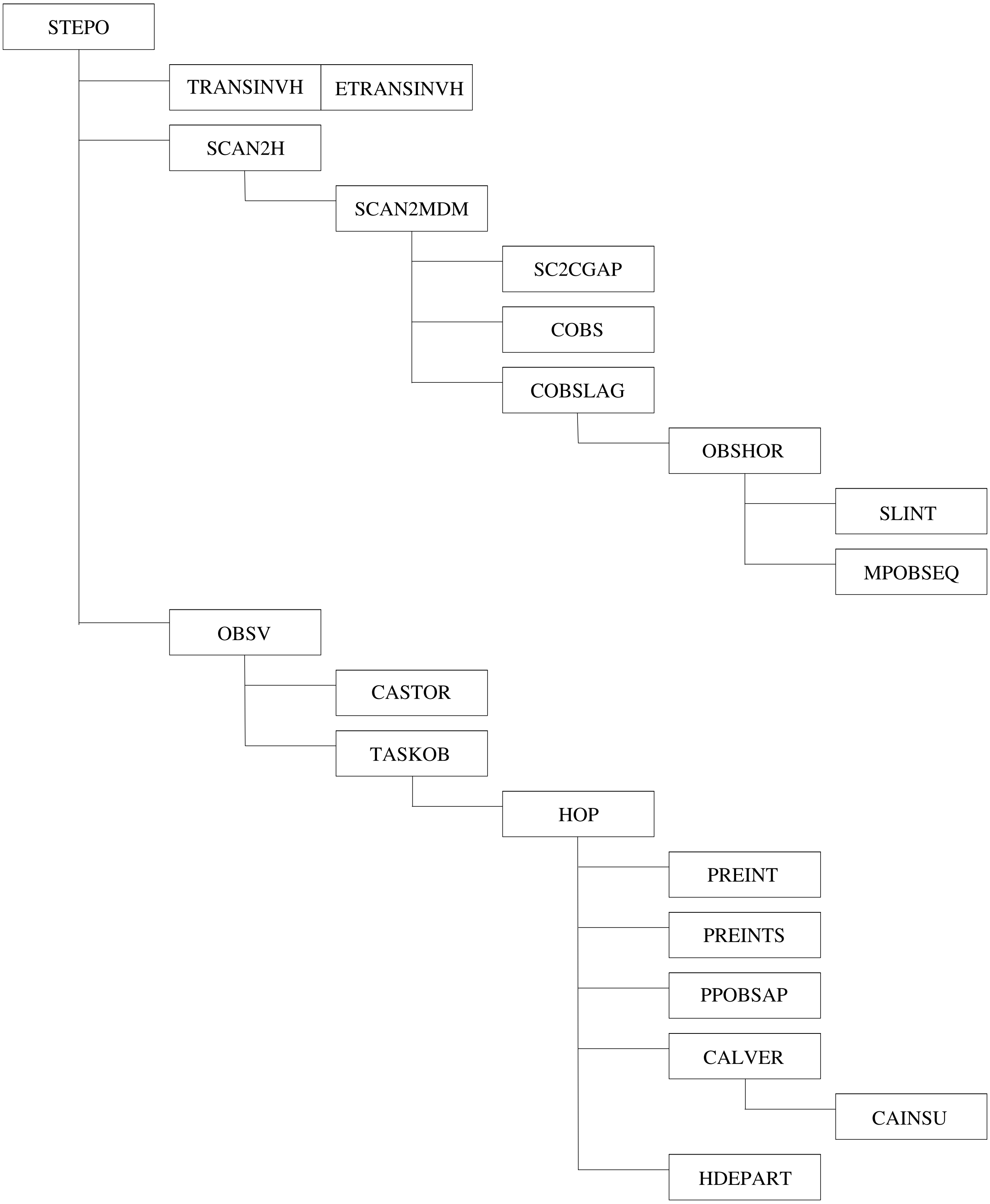
Then the specific subroutine PUTDB is called in order to really update the database.

CAPITO is executed in order to print some information on the execution listing about the observations distribution and their error standard deviation.

9. <u>Departures calculation</u>

The departures calculation is not a specific CANARI code, the same subroutines are used to compute the departures for the different analysis methods (3D-var, 4D-var, screening). Only the code which concerns CANARI is described here.

This part is controlled by the subroutine CADAVR, which is called at the beginning of CANARI to compute the observed departures and at the end to compute the analyzed departures (under control of the key LRESCPL (module QALORI)). Main of the work is done by STEPO, called with the configuration ’0M100C000’ by CADAVR. The corresponding tree is :



The subroutine TRANSINVH (ARPEGE case, or ETRANSINVH in the ALADIN case) is called (due to the 1 in the third position in the STEPO configuration) in order to transform the spectral arrays of the model fields in their gridpoint representation. Then SCAN2H is called, it is the subroutine which manages the gridpoint computations from STEPO. It calls SCAN2MDM which checks the size of the model fields buffer (filled after TRANSINVH), calls SC2CGAP in order to initialize the pointer address for each field, manages a loop over points packets (NPROMA size) which calls COBS in order to fill some corresponding local arrays, each with a small part of a model field, and finally calls COBSLAG which prepare the horizontal interpolations. COBSLAG organizes the work with a loop over observations packets (known thanks to observations related global arrays (cf. MKGLOBSTAB)) and calls the subroutine OBSHOR which marks out the latitude and the longitude positions, calls SLINT which really performs the interpolations for all the model fields and then calls MPOBSEQ which fills the global arrays with the model equivalent of each field for each geographical location listed by COBSLAG. These global arrays are included in the module YOMMVO for the model fields (upperair and surface), YOMCTOBS for the constant fields (albedo, vegetation and so on) and QARBUR for the model error fields (CANARI specific). At this stage, all the quantities known by the model are interpolated at each observation point (horizontal part only). And note that during all this work, the ODB is closed because all is done in connection with the gridpoints, there is no local observation array.

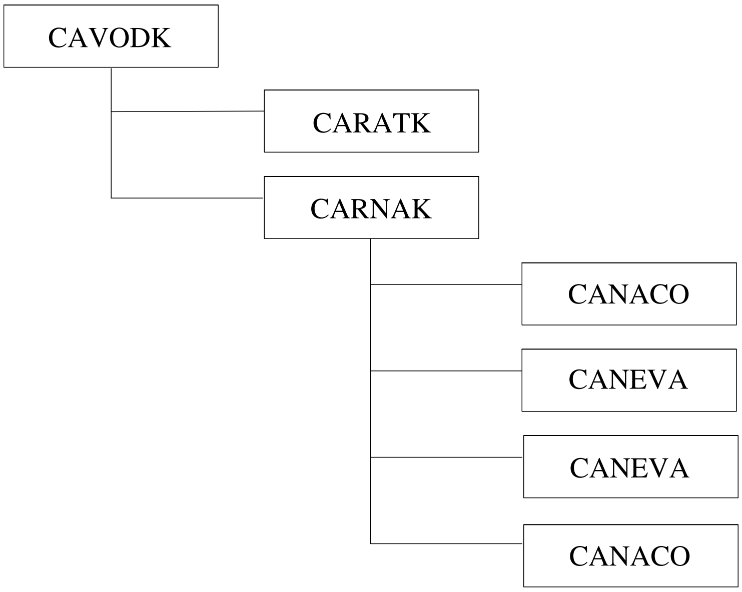
Then the work involving the obervations begins. STEPO calls OBSV (due to the 'C' in the sixth position of the call configuration) in order to manage the vertical part of the interpolation. OBSV first calls CASTOR which accesses to the ODB and divides the observations in several packets of same type (information saved in the module YOMOBSET). Then OBSV calls TASKOB which calls the observation operator subroutine HOP in a loop over the observations packets. HOP is a very important subroutine which initializes some local arrays according to the observations it has to deal with, accesses to the ODB, calls PREINT in order to prepare the interpolation by filling some local arrays with all the necessary information about the observations (orography, horizontal first-guess equivalent computed just before) and about the model (pressure levels, temperature profile), calls PREINTS which adds local arrays filled with some information necessary for the surface observations. Then HOP prepares a loop on the model variables so the same observed parameters of the whole packet are processed together. And in this loop, according to the parameter type, the corresponding observation operator is called : PPOBSAP for the upperair variables (LCAPACH supposed to be .TRUE.), PPOBSAC for the surface variables (LCACHMT supposed to be .TRUE.) except the snow and the surface pressure which are processed by a specific operator, PPOBSN and PPOBSAZ respectively. Then HOP calls CALVER which performs the vertical interpolation of the model error standard deviation (stored in QARBUR by MPOBSEQ), calls CAINSU in order to define the parameters of the statistical model on the current levels, then updates the ODB with the appropriate model error standard deviation for each parameter. Finally, HOP calls HDEPART which computes the difference observation minus model equivalent and updates the ODB with the observed departure for all the parameters of the current observations packet.

Once the departures have been calculated, CADAVR calls the subroutine CANCER in order to check the observations validity versus the first-guess. CANCER computes the normalized departures, then prints some statistics (for each parameter of each observation type) in order to have an idea about the distribution of the observations (by classes or globally). CANCER updates some quality flags of the observations, the "first-guess" one and the "final" one (cf. chapter 11 for a detailed description of these flags) and saves them in the ODB as well as the normalized departures.

10. Spatial quality control

As already mentioned in this documentation (§2.2 and §5.3), the spatial quality control consists in an analysis performed at each observation point without this observation itself. The calling tree of the analysis subroutines is shown here, but not these subroutines themselves, this is done in a specific chapter about the heart of the OI itself (cf. chapter 13).

This part of the code is controlled by the subroutine CAVODK, called by CANARI (cf. §6.3) according to the value of the key LAECHK (cf. §6.2.3). The calling tree is :



CARATK performs some initializations for the specific OI of the spatial quality control, these variables are stored in the module QADORE (please note that these initializations are tunable for the analysis step itself (via NADOCK for QADOCK) as they are done by DATA instructions here). Then CAVODK accesses to the ODB in order to save in some local arrays all the necessary information about the observations of the involved pool (cf. §5.3 to remind how this part is built regarding distributed memory) and calls CARNAK which performs the control itself. To do this, CARNAK calls CANACO which allows to access to the ODB and to have a global vision of it (i.e. for all the observations of all the pools). After this CARNAK works by a loop over the observations of its own pool (thanks to the local arrays filled by CARNAK), it calls CANEVA a first time in order to perform the geographical selection around the observation which is being processed, and then in a loop over the parameters calls CANEVA again to do the statistical selection and build the linear system and solve it. Then CARNAK updates the spatial quality flag of the parameter according to the result of the local analysis (cf. chapter 11). At the end, CANACO is called again to close the global access to the ODB. After all these steps, CARNAK accesses to the ODB again in order to save the spatial quality flag.

11. Quality flag

11.1. Description of the flags in the ODB

The main flag which is used in CANARI is called "anflag" in the ODB nomenclature, as analysis flag. This flags exists for each parameter (BODY table). Some other flags appear in the ODB, as "status" in the HDR table, but it is only used in the screening configuration. Another one is "rdbflag", as database flag (for each parameter, BODY table), used in CANARI too. These flags consist in a list of bits where some information is coded. The 1-bit information is accessible thanks to a set of in-line functions coded in the module QABITU which allow to read and to write some bits in the flags. Each bit has a name which is assigned in the module QAKEKI.

The database flag rdbflag is coded over 30 bits, the first half concerns the quality of the vertical coordinate and the second half the quality of the parameter itself. Each 15 bits packet has the same structure (the name of each bit is given too (first packet/second packet)) :

bit 1	0 no human control	NDBCVS/NDBPAS
	1 human control	
bit 2	0 no correction by the meteorological databank preprocessing	NDBCVPR/NDBPAPR
	1 correction by the meteorological databank preprocessing	
bit 3	0 no constraint on the parameter	NDBCVF/NDBPAF
	1 "forced" parameter	
bits 4 and 5	0 correct parameter	NDBCVCQ/NDBPACQ
	1 probably correct parameter	
	2 probably incorrect parameter	
	3 incorrect parameter	
bit 6	0 no controlled or software initialized parameter	NDBCVCQM/NDBPACQM
	1 human control initialized parameter	
bits 7 and 8	0 correct parameter versus previous analysis	NDBCVACQ/NDBPAACQ
	1 probably correct parameter versus previous analysis	
	2 probably incorrect parameter versus previous analysis	
	3 incorret parameter versus previous analysis	
bit 9	0 parameter no used by the previous analysis	NDBCVAU/NDBPAAU
	1 parameter used by the previous analysis	
bits 10 to 15	0 not used	

The analysis flag is coded over 29 bits, the first 20 correspond to the quality of the parameter and the next 9 to its use in any variable analysis. So its structure is :

bits 1 to 4 : final quality code	NCQFxx
bits 5 to 8 : first-guess quality code	NCQExx
bits 9 to 12 : spatial quality control code	NCQKxx
bits 13 to 16 : variational quality code (not used in CANARI)	NCQVxx
bits 17 to 20 : blacklist code	NCLNxx
bit 21 : if set to 1, parameter used in the surface pressure analysis	NCUPS
bit 22 : if set to 1, parameter used in the wind and temperature analysis	NCUUVT
bit 23 : if set to 1, parameter used in the relative humidity analysis	NCUHU
bit 24 : if set to 1, parameter used in the 2 meters temperature analysis	NCUT2M
bit 25 : if set to 1, parameter used in the 2 meters relative humidity analysis	NCUH2M
bit 26 : if set to 1, parameter used in the 10 meters wind analysis	NCUV10
bit 27 : if set to 1, parameter used in the precipitations analysis (not coded yet)	NCURR
bit 28 : if set to 1, parameter used in the snow analysis	NCUSN
bit 29 : if set to 1, parameter used in the SST analysis	NCUSST

and for each of the five first codes :

bit 1	0 no information	
	1 correct parameter	(xx=CO)
bit 2	0 no information	
	1 probably correct parameter	(xx=PC)
bit 3	0 no information	
	1 probably incorrect parameter	(xx=PI)
bit 4	0 no information	
	1 incorrect parameter	(xx=IN)

The bits 21 to 29 are updated by the different variables analysis steps themselves.

11.2. Update of the quality flags

The blacklist flag is updated by the software which builds the ODB (called batodb). Only the bits 1 or 4 (NCLNCO or NCLNIN) can be changed.

The first-guess flag is updated by the subroutine CANCER (cf. end of chapter 9) for each parameter once the normalized observed departure has been calculated. First, the parameter is declared incorrect (NCQEIN=1 and even NCQFIN=1) if it is obviously bad (erroneous altitude, too big difference between the observation altitude and the model orography, observed departure not calculated,...). Then, the normalized observed departure is compared to a threshold (RCOEFK, module QAMCOK, set by NAMCOK (cf. §6.2.3)). If it is bigger than the corresponding RCOEFK (according to the parameter and the observation type), the parameter is declared probably incorrect (NCQEPI=1). If it is between RCOEFK and 70% of RCOEFK, the parameter is declared probably correct (NCQEPC=1). If it is smaller than 70% of RCOEFK, the parameter is declared correct (NCQECO=1).

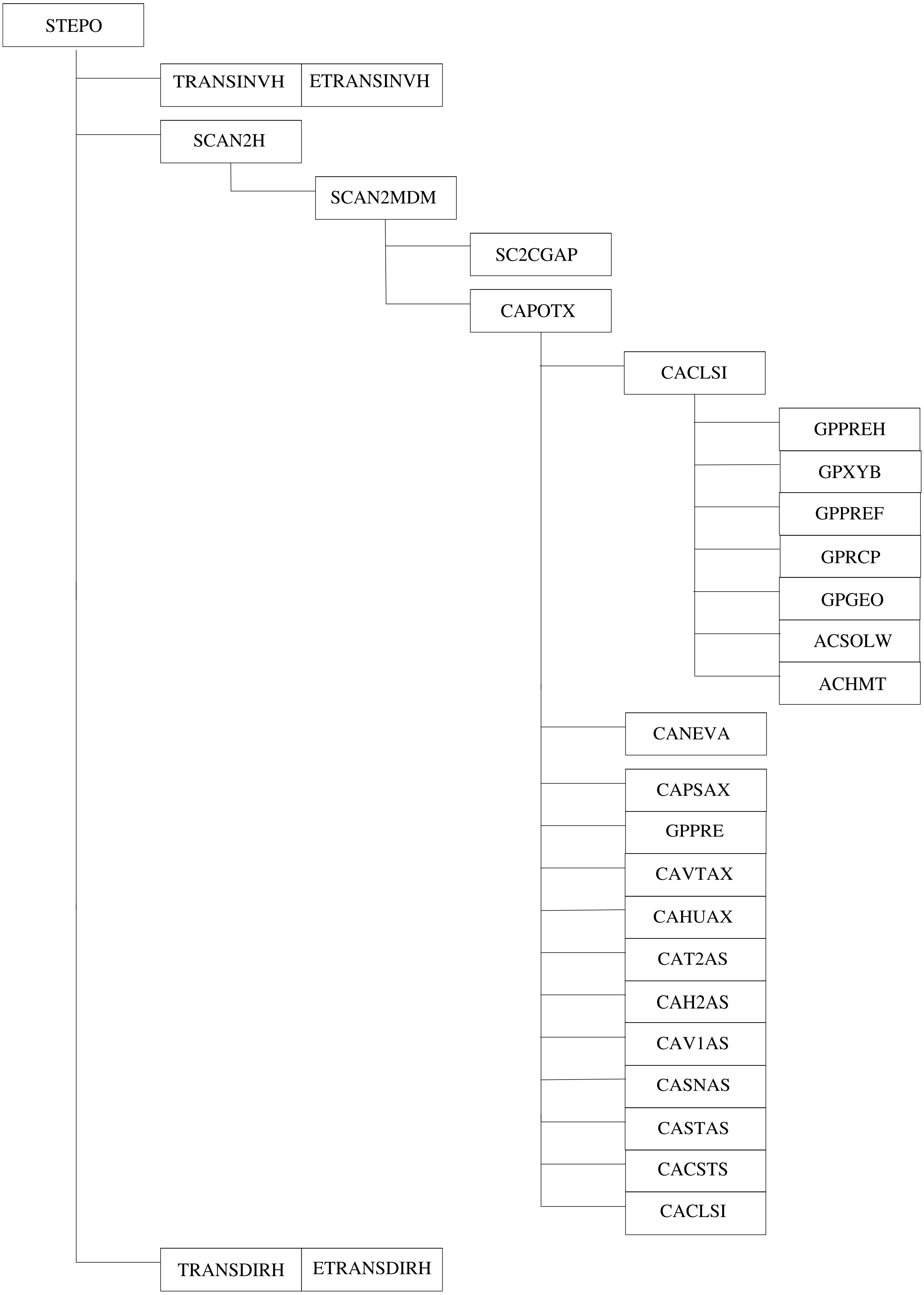
The spatial quality code is updated by the subroutine CARNAK (cf. chapter 10) for each parameter once the analysis without the observation to control has been performed. The same test against the same threshold than in CANCER has been performed for the normalized difference (obs-ana) and the various bits of the new flag are initialized (NCQKCO, NCQKPC, NCQKPI). The parameter is declared incorrect (NCQKIN=1) if the analysis error variance is bigger than 90% of the first-guess error variance.

The final quality flag is updated by the subroutine CANTIK before the model variables analysis itself. It is this code which defines if a parameter can be used or no in the analysis. This flag is a synthesis of the previously updated codes (NCQExx and NCQKxx) and of the database flag (rdbflg, updated by the software batodb). In any case, if the parameter is blacklisted (NCLNIN=1), it is incorrect and the final quality flag is updated (NCQFIN=1). If it is forced (NDBPAF=1 in rdbflag), it is declared correct and NCQFCO is set to 1. If it has been declared bad by the first-guess check (NCQEIN=1), it is still incorrect and NCQFIN is set to 1. Otherwise, according to the spatial quality control, if the parameter is correct (NCQKCO=1) it is still correct and NCQFCO is set to 1, if it is probably incorrect NCQFPI is set to 1. In the other case (NCQKPC=1 or NCQKIN=1 or no control performed (LAECHK=.FALSE.), this means that the spatial quality control only do not allow to conclude), the first-guess check result is taken into account and if the parameter is correct (NCQECO=1) it is still correct and NCQFCO is set to 1, if it is probably incorrect (NCQEPI=1) NCQFPI is set to 1; in the other case (NCQEPC=1), as we can not conclude at this stage, the final flag is updated according to the result of the meteorological databank preprocessing, if NDBPACQ<=1 then NCQFCO is set to 1 and if NDBPACQ>1 then NCQFPI is set to 1.

12. Meteorological variables analysis organization

12.1. General organization

Once the system has defined which observations are of a good quality, the analysis itself can begin. It is managed by the subroutine STEPO, called under the configuration '0M1001AA0', by the subroutine CANARI (cf. §6.3). It is reminded here that a call to CANACO has been performed just before in order to access to the ODB and to produce a global view of the observations whatever the number of processors may be. The calling tree is :


STEPO
TRANSINVH
ETRANSINVH
SCAN2H
SCAN2MDM
SC2CGAP
CAPOTX
CACLSI
GPPREH
GPXYB
GPPREF
GPRCP
GPGEO
ACSOLW
ACHMT
CANEVA
CAPSAX
GPPRE
CAVTAX
CAHUAX
CAT2AS
CAH2AS
CAV1AS
CASNAS
CASTAS
CACSTS
CACLSI
TRANSDIRH
ETRANSDIRH

The subroutine TRANSINVH (ARPEGE case, or ETRANSINVH in the ALADIN case) is called (due to the 1 in the third position in the STEPO configuration) in order to transform the spectral arrays of the model fields in their gridpoint representation. Then SCAN2H is called, it is the subroutine which manages the gridpoint computations from STEPO. It calls SCAN2MDM which checks the size of the model fields buffer (filled after TRANSINVH), calls SC2CGAP in order to initialize the pointer address for each field, manages a loop over points packets (NPROMA size) inside which CAPOTX is called. CAPOTX is the deeper level subroutine which organizes the analysis. It is a very modular subroutine, all the necessary information concerning the model fields is passed as explicit arguments. The analysis mask is used just before the call to CAPOTX to pack the model data in order to send to the analysis modules only the information about the points which have to be analyzed. In SCAN2MDM, after CAPOTX, the analyzed gridpoint values for the various variables are put at the right place in the source arrays. Finally, TRANSDIRH (ARPEGE case, or ETRANSDIRH in the ALADIN case) is called (due to the A in the eighth position in the STEPO configuration) in order to transform back the gridpoint arrays of the model fields in their spectral representation.

12.2. Local organization : the subroutine CAPOTX

The model variables are passed to CAPOTX as explicit arguments in the form of to be analyzed gridpoints arrays, one array for each model upperair field and another array for the surface fields. The first thing to do now is the computation of the boundary layer fields (they are not in memory yet because they are not explicit model variables). This is done by the subroutine CACLSI. It calls some common routines in order to define the vertical structure of the model and to initialize some meteorological quantities which are necessary before calling the subroutine ACHMT which really performs the calculation of these fields (vertical complex interpolation between surface and lower level of the model for each field). The second step is the selection of the observations which will be usefull for the current set of gridpoints. This is done by a call to CANEVA, which performs only a geographical selection (according to the calling options). The involved observations are stored in arrays which will be accessed by all the analysis modules. After that, according to the options of *NACTEX*, the various meteorological variables analysis subroutines are called :
- CAPSAX for the surface pressure analysis,
- CAVTAX for the upperair wind and temperature analysis,
- CAHUAX for the upperair relative humidity analysis,
- CAT2AS for the 2 meters temperature analysis,
- CAH2AS for the 2 meters relative humidity analysis,
- CAV1AS for the 10 meters wind analysis,
- CASNAS for the snow analysis,
- CASTAS for the sea surface temperature analysis.

After CAPSAX, if any upperair analysis has to be performed, GPPRE is called in order to compute the new half and full model levels pressure derived from the new surface pressure. The eight variable analysis subroutines are built in the same way : the model errors statistics related local arrays are initialized according to the predictors list defined for the analysis of the current predictand (module QAPREX), then CANEVA is called in order to build and solve the linear system for each gridpoint (loop on the horizontal dimension first, and on the vertical dimension later if upperair analysis)(cf. chapter 13). To finish, the local model field array is incremented with the analysis result, the analysis error is saved and the diagnostics related arrays (module QADIAG) are filled. It is not mentioned on the tree (to save place), but the subroutine CACDGU is called several times in order to prepare these analysis diagnostic arrays. The purpose is to control the work done by the analysis; so for each analyzed variable, some statistics are calculated (mean, standard deviation, minimum, maximum) for the next five fields : guess, analysis, increment, size of the linear systems, analysis error standard deviation divided by the guess error standard deviation (marker of the analysis efficiency). CACDGU updates the appropriate arrays for the current set of points, variable by variable.

Once the OI analyses have been performed, CAPOTX calls CACSTS in order to calculate the surface fields over land on one hand, where the surface and soil temperatures and the surface and soil water content are derivated from the 2 meters analyzed increment (temperature and relative humidity) according to ISBA and smoothed by the use of the historical increments (ICMSHxxxxLISSE) and the climatological relaxation, the snow height is updated too according to the surface temperature and the climatological relaxation, over sea on the other hand where the surface temperature is updated by the use of the SST analysis and smoothed by the climatological relaxation too (and the soil temperature is set equal to the surface temperature). Finally, according to LAECDS, CACLSI is called in order to compute again the boundary layer fields according to the new surface conditions.

13. The heart of the numerical OI : linear systems management and solving

The numerical optimum interpolation is managed by the subroutine CANEVA, which is called several times in CANARI. This routine allows to realize the geographical selection among the observations on one hand, the statistical selection among the parameters (predictors) on the other hand and then to build the linear system and solve it (this can be done for a vertical or point by point). The work performed by CANEVA is controlled by some calling options, and the different steps can be executed separately (but there is a logical link between them) or in one call. All the involved subroutines are part of what is called the heart of the OI analysis package. Their name is always CA...A. An effort has been made to document the code, so there are a lot of commentary lines that everybody can refer to in order to understand how each subroutine is coded.

CANEVA is called with a large number of options in order to increase its modularity. Here is the description of the input arguments :

- KQUOI : integer variing from -3 to 3 which specifies what CANEVA has to do :
  - KQUOI = 0 : only geographical selection (geo),
  - KQUOI = 1 : geo + point by point both statistical selection (stat) and analysis (ana),
  - KQUOI = 2 : geo + stat on a vertical + point by point ana,
  - KQUOI = 3 : geo + both stat and ana on a vertical,
  - KQUOI < 0 : idem KQUOI > 0 but without the geographical selection;
- KANTY : integer variing from 0 to 10 which specifies how CANEVA has to be used :
  - KANTY = 0 : analysis,
  - KANTY = 1, 2,...,10 : spatial quality control of the observations of type KANTY;
- KTASK : number of the computer task;
- PDISTR : maximum distance of each observation type research around a point (in meters);
- PDELTA : maximum vertical distance of each observation type selection in pressure (in Pa);
- PDELPI : minimum distance between 2 selected levels from the same observation (in Pa);
- KMXGSL : maximum number of observations of each type allowed for the geogaphical research;
- KMXGQ : maximum number of observations of each type allowed in each quadrant after the geographical selection;
- PCORMIN : minimum correlation for a predictor to be selected;
- PSTRET : stretching function for the horizontal correlation estimation;
- PLATA : latitude of the points to be analyzed;
- PLONA : longitude of the points to be analyzed;
- PRESSS : pressure of the points to be analyzed;
- KNBPH : number of points to be analyzed on the horizontal grid;
- KNBPV : number of points to be analyzed on the vertical;
- KNXI : number of the statistical model for each predictand;

- KNLXI : number of predictand types (length of KNXI);
- KNYJ : number of the statistical model for each predictor;
- KNLYJ : number of predictor types (length of KNYJ);
- KNM : number of predictors to be selected by analysis point for each predictand of KNXI;
- PSIGS : first-guess error basic standard deviation.

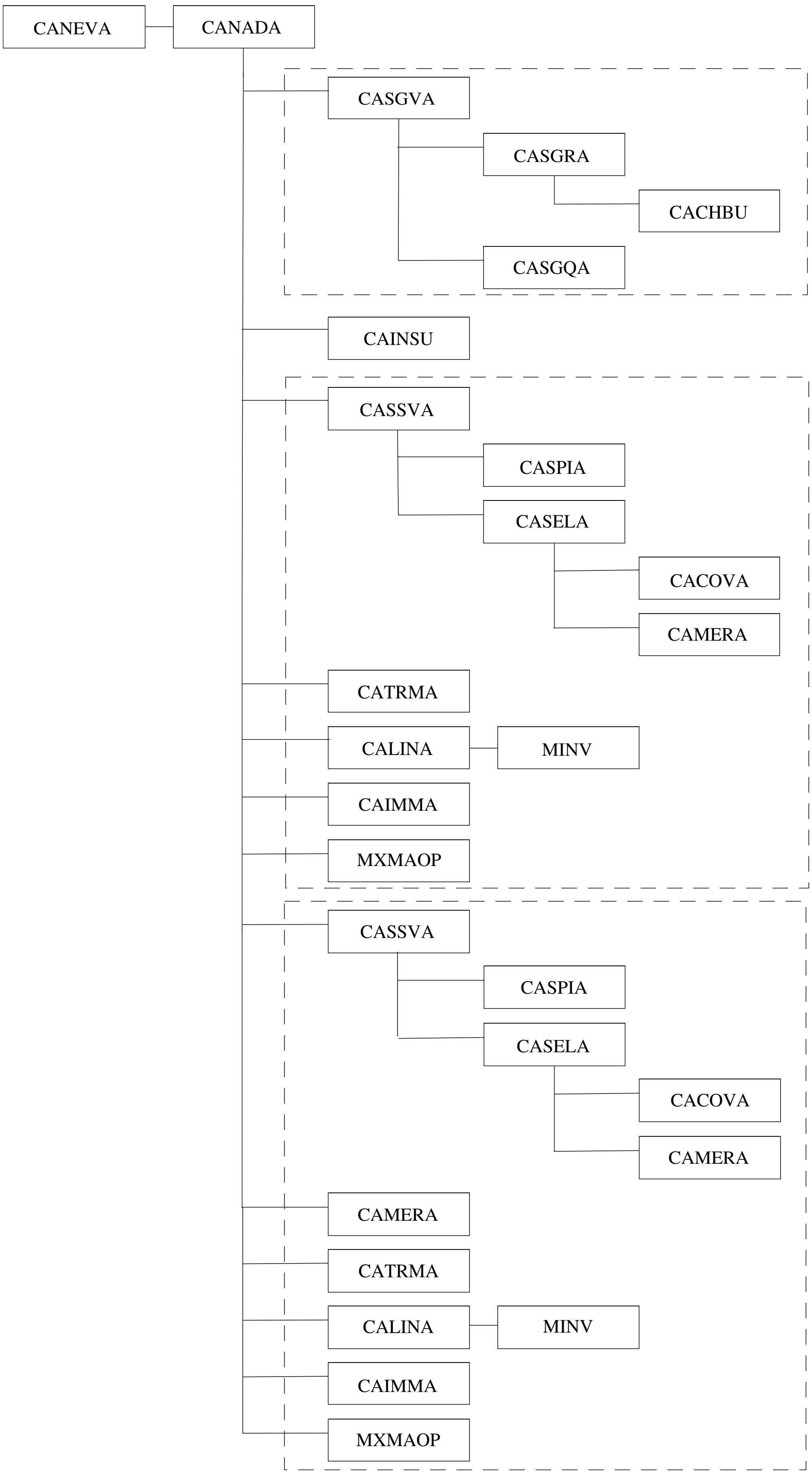
Output arguments :
- PSIGPS : updated first-guess error standard deviation (really used - cf. module QAVARI);
- PRESIA : analysis increments;
- PSIGA : analysis error standard deviation;
- KNPR : number of used predictors by gridpoint.

It is important here to understand the difference between an observation and its parameters, i.e. between a spatial-and-time position and the observed values. The geographical selection is performed over a list of latitudes/longitudes, even when the statistical selection is performed among the observed parameters of the spatially selected observations. That is the reason why CANEVA is called a first time in CAPOTX in order to perform the geographical selection for the concerned gridpoints (packet treated by this CAPOTX call), and CANEVA is called later with a negative KQUOI by each analysis subroutine (CA...X or CA...S, cf. chapter 12) according to the predictand (variable to be analyzed) (on the same way, for the spatial quality control, in CARNAK, there is a loop on the observations, with a first call to CANEVA, and inside another loop on the parameters with a new call to CANEVA for each of them).

The purpose of the OI subroutines package is to be able to analyze every parameter (predictand) with every parameter (as predictor this time); but in fact this is reduced according to the frame involved by *NAMPRE* (saved in QAPREX, cf. §6.2.2) and the fact that each parameter has to have an associated statistical model (saved in QAVARA by CANAMI, cf. §6.2.2 again).

CANEVA is the interface subroutine between the OI heart and its users. It checks the validity of some input arguments (some analysis arrays have a size which is limited by a parameter (cf. list in the module QAPABL)), then organizes the predictors and the predictands and finally calls CANADA with the same arguments which really manages the work to do (in a loop over the horizontal dimension KNBPH).

The calling tree is as shown below (the three dash boxes correspond to the three main actions : geogaphical selection, statistical selection and analysis on a vertical, statistical selection and analysis point by point) :


CANEVA
CANADA
CASGVA
CASGRA
CACHBU
CASGQA
CAINSU
CASSVA
CASPIA
CASELA
CACOVA
CAMERA
CATRMA
CALINA
MINV
CAIMMA
MXMAOP
CASSVA
CASPIA
CASELA
CACOVA
CAMERA
CAMERA
CATRMA
CALINA
MINV
CAIMMA
MXMAOP

13.1. The geographical selection

CANADA calls CASGVA which manages this geographical selection. CASGVA organizes a loop over all the observation types and calls CASGRA and then CASGQA for each type.

CASGRA first calls CACHBU in order to determine the number of the observation boxes which have at least a part which is included in a circle of radius the maximum research length PDISTR and centered on the position (latitude PLAT and longitude PLON) of the processed point. The use of these boxes allows to save a lot of time and computations (their number has been initialized by CABOTU (cf. chapter 8)). Then CASGRA computes the distance for all the observations of all the selected boxes and extracts only those which are located around the vertical of the cylinder with the previously referred radius until the maximum number of observations allowed KMXGSL (for each type). And the sum of all the selected observations must be inferior to the fixed parameter JPK3SE (module QAPABL) (be carefull, these last constraint can bring about the non-reproductibility of CANARI according to the number of processors involved in the job).

CASGQA realizes the selection by geographical sector (i.e. a quadrant in most of the cases) : the selected observations are sorted by sector and if the maximum number KMXGQ allowed is reached then the nearest observations of the vertical (PLAT,PLON) by sector are saved in the final observations arrays (this is done in order to have a maximum isotropy in the distribution).

It is important to note that this selection is done by vertical, this means that the result is the same for all the points of a model vertical; so it is not necessary to perform this selection successively for several points of the same vertical ...

13.2. The statistical selection

This statistical selection can be performed for a point or for a vertical. CANADA calls CASSVA in order to manage this very important part of the numerical OI; a call to CAINSU performed just before initializes some parameters of the structure functions depending on the location.

The first step is performed by CASPIA which selects the interesting predictors among the previous selected observations (a predictor is a measured parameter at a given level). In fact these interesting predictors are those which are listed as right predictors for the involved predictands (defined by KNYJ), which have a quality flag set to "good" and which are in a given pressure layer. If the selection is done for a vertical, all the levels above a given pressure (pressure of the lowest level plus PDELTA) are kept; if the selection is done for a point, only the levels which are included in a layer around the point pressure are kept. More, in order to avoid some problems with the observations which have a lot of levels, an interdistance constraint (PDELPI) is applied on the vertical (between the parameters of each observation itself), in such a way that the nearest level of the work level is kept.

After CASPIA, CASSVA calls CASELA which performs the statistical selection itself. First CACOVA is called in order to calculate the correlations between the first-guess errors at a given point for the variables to be analyzed (predictands) and the first-guess errors for a list of selected predictors (horizontal part first, and then vertical part for the upperair variables). From these correlations, we obtain the covariances and the normalized correlations (by the observation error and the first-guess error at the observation point (predictor)). CAMERA is called later by CASELA in order to select the best predictors according to the next criteria : the predictor has a minimum correlation (PCORMIN) with at least one predictand, and are kept those which present the maximal correlation with the predictands. This last step is done in the order of the predictands as defined in the table NVXINV (module QAVARA), and as a maximum KNM predictors are kept by predictand in order to select different predictors for each predictand but without exceeding the limit KNM*KNBPV; and once a predictor has been selected by a predictand, the other predictands can not take it and they have to select less correlated predictors. Then CASELA saves the accurate covariances because they provide the rightenside of the linear system.

It is important to note that it is not necessary to do the statistical selection for some points which are very close on the same vertical, because it is quite sure that the selected predictors will be the same ... It is better to perform this selection on a vertical !

13.3. The linear system

If the predictors selection has been performed on a vertical and if the solving of the linear system has to be done point by point, a new call to CAMERA is performed in order to select the predictors for the given point, but only between those which have been selected for the vertical.

The flag which describes if a parameter has been used by one analysis is updated (cf. §11.1).

CANADA calls CATRMA in order to build the matrix of the linear system, i.e. CATRMA calculates all the interdistances between the predictors, then the covariances between the first-guess errors and between the observations errors (hypothesis of separability done for the estimation of the correlations) for the involved predictors are computed, and the variances too.

Then CANADA calls CALINA which organizes the array containing the leftenside (matrix (n,n)) and the rightenside (matrix (m,o)) of the linear system and then calls the mathematic subroutine MINV with this array in order to solve the linear system.

If there is a problem, the program crashes but just before a call to CAIMMA is done in order to print the linear system (to allow the user to understand where the error occured).

BIBLIOGRAPHY

[1] - ODB documentation -

ODB : guide pratique

written by Dominique Puech - CNRM/GMAP - 09/10/2002 -

[2] - ARPEGE files format -

Note de travail ARPEGE n°12 : Logiciel de Fichiers Indexes
- Jean Clochard - METEO-FRANCE - 11/1989 -

Note de travail ARPEGE n°17 : Interfaces de manipulation des fichiers ARPEGE
- Jean Clochard - METEO-FRANCE - 04/1990 -

Internal documentation : ARPEGE/ALADIN files package
- J. Clochard, R. El Khatib, D. Paradis - METEO-FRANCE - 11/2000 -

Internal documentation : The LFI file access library
- F. Bouttier and J. Clochard - METEO-FRANCE - 06/2002 -

[3] - ISBA description -

A simple parameterization of land surface processes for meteorological models.
J. Noilhan and S. Planton. Monthly Weather Review. 1989. N°117 : 536-549.

Implementation of a new assimilation scheme for soil and surface variables in a global NWP model.
D. Giard and E. Bazile. Monthly Weather Review. 2000. N°128 : 997-1015.

[4] - Distributed memory -

Internal documentation :

An introduction to the memory-distributed aspect of the ARPEGE/IFS/ALADIN
- Ryad El Khatib - CNRM/GMAP - 12/2002 -

Distributed memory features in the cycle 25 of ARPEGE/IFS
- Karim Yessad - CNRM/GMAP - 05/2002 -

Annex 1

Example of a script to submit in order to run CANARI for ALADIN France on the Meteo-France VPP computer (1 processor).

This job is quasi-operational at Meteo-France, it runs with surface observations only, extracted on a domain a little bigger than the France. All the variables are analyzed by the OI (but the update of the surface fields is not performed because there is no interest in doing this). The purpose is to examine the boundary layer fields only and to have the report of the "surface" observations on the vertical (to compute later convection diagnostic fields).

```
# @$-eo -r e701
# @$-lT 1200 -lM 2500mb -lV 0mb
# @$-lP 1

set -x
export VPP_MBX_SIZE=32000000
cd $tmpdir

#---------------------------------------------------------------------
dat=2002091309
#---------------------------------------------------------------------

AA=$( echo $dat | cut -c1-4 )
MM=$( echo $dat | cut -c5-6 )
JJ=$( echo $dat | cut -c7-8 )
HH=$( echo $dat | cut -c9-10 )

#
#  Input files recovery (obs, guess)
#  ------------------------
ftget<<XX
/chaine/mxpt/mxpt001/ald_horaire/oper/$AA/$MM/$JJ/guess_$HH    ICMSHANALINIT
/chaine/mxpt/mxpt001/ald_horaire/oper/$AA/$MM/$JJ/OBSODBF.tar  fichiers_obs
XX

tar xvf fichiers_obs
cp obsodbf.$HH  ficobs

mkdir obs_init
cd obs_init
tar xvf $tmpdir/ficobs
cd $tmpdir

#
#  Specific CANARI files recovery
#  ----------------------------------------
cp ~mxpt001/arpege/france/oper/const/autres/rszcoef_fmt  rszcoef_fmt

cp ICMSHANALINIT ELSCFANALALBC000

export IOASSIGN=~mxpt001/arpege/france/oper/const/autres/ioassign
```

```
#
#  Configuration definition
#  ------------------------------
NCONF=701
VERSION=meteo
TSTEP=1.
NSTOP=t0
ADVEC=sli
MODEL=aladin

export TO_ODB_ECMWF=0
export ODB_STATIC_LINKING=1
export ODB_SRCPATH_ECMA=$tmpdir/obs_init
export ODB_DATAPATH_ECMA=$tmpdir/obs_init

export ODB_CMA=ECMA
export ODB_TRACE_FILE=listing_odb

#
#  Namelist
#  -----------
cat << FIN > fort.4
 &NACOBS
   OROLIM=1500.,
   ORODIF=800.,
 /
 &NACTAN
   LANMASK=.TRUE.,
   ALATNB=53.,
   ALATSB=41.,
   ALONWB=353.,
   ALONEB=10.,
 /
 &NACTEX
   LAEOMF=.TRUE.,
   LAECHK=.TRUE.,
   LAEPDS=.TRUE.,
   LAEUVT=.TRUE.,
   LAEHUM=.TRUE.,
   LAET2M=.TRUE.,
   LAEH2M=.TRUE.,
   LAEV1M=.TRUE.,
   LAESNM=.FALSE.,
   LAESST=.FALSE.,
   LAEICS=.FALSE.,
   LAECDS=.FALSE.,
   LAESTU=.FALSE.,
   LAESTA=.FALSE.,
   LAEWIO=.TRUE.,
   LAEOMN=.TRUE.,
   LAERFO=.TRUE.,
   RCLIMCA=0.0,
```

```
  RCLISST=0.0,
  LAEINC=.FALSE.,
  NAEINC=0,
  NSSTLIS=0,
/
&NACVEG
/
&NADOCK
  QDSTRA(1)=1000000.,0.,0.,1000000.,1000000.,1000000.,0.,0.,0.,0.,
  QDSTVA(1)=10000.,10000.,10000.,10000.,10000.,10000.,10000.,0.,0.,0.,
  MINMA(1)=15,5,5,15,15,15,8,0,0,0,
  QCORMIN(1)=0.05,0.05,0.05,0.05,0.05,0.05,0.05,0.05,0.05,0.,
  QDELPI(1)=3000.,3000.,3000.,1000.,5000.,5000.,5000.,1500.,5000.,0.,
/
&NAEPHY
/
&NAERAD
/
&NAIMPO
/
&NALORI
  RCALPH=0.0,
/
&NAMAFN
/
&NAMCAPE
/
&NAMCFU
/
&NAMCHK
/
&NAMCOK
  RCT2TP=3.0,
  RCT2SY=3.9,
  RCH2SY=2.5,
  RCV1SY=4.1,
/
&NAMCT0
  LFDBOP=.FALSE.,
  NSPPR=1,
/
&NAMCT1
  LRFILAF=.FALSE.,
/
&NAMCVA
  NVATRU=-1,
/
&NAMDDH
/
&NAMDIF
/
&NAMDIM
```

```
  NPROMA=8190,
  NFPP3M=0,
/
&NAMDMSP
/
&NAMDPHY
  NVRS=4,
/
&NAMDYN
  NLOSP=0,
/
&NAMFA
  NBITCS=-1,
  NSTRON=-1,
  NBITPG=-1,
/
&NAMFFT
  NFFTW=1020,
/
&NAMFPC
/
&NAMFPD
/
&NAMGEM
/
&NAMGMS
/
&NAMGOES
/
&NAMGRIB
/
&NAMINI
  NEINI=0,
/
&NAMIOS
/
&NAMJG
/
&NAMJO
/
&NAMLCZ
/
&NAMLEG
/
&NAMMCC
/
&NAMMETEOSAT
/
&NAMNMI
/
&NAMNUD
/
```

```
&NAMOBS
  LSLREJ=.TRUE.,
/
&NAMOPH
/
&NAMPAR0
  NPROC=1,
  NPROCA=1,
  NPROCB=1,
  NOUTPUT=1,
/
&NAMPAR1
  LOCKIO=.FALSE.,
  LSPLIT=.FALSE.,
  NSTRIN=1,
  NSTROUT=1,
/
&NAMPHY
/
&NAMPHY0
  USURICL=4.,
/
&NAMPHY1
/
&NAMPHY2
/
&NAMPHY3
/
&NAMPHYDS
/
&NAMPONG
/
&NAMPPC
/
&NAMPRE
  IPRSK(1,1) =  1,  3,  4,
  IPRSK(1,2) =  2,
  IPRSK(1,3) = 29,
  IPRSK(1,4) =  1,  2, 57,
  IPRSK(1,5) = 11,
  IPRSK(1,6) = 80, 92,
  IPRSK(1,8) = 39,
  IPRSK(1,9) = 58,
  IPRSK(1,10)= 41, 42,
  IPRSA(1,1) =  1,  2,  3,  4,
  IPRSA(1,2) =      2,  3,  4, 41, 42, 39,
  IPRSA(1,4) = 29, 58,
  IPRSA(1,5) = 39,
  IPRSA(1,6) = 58,
  IPRSA(1,7) = 41, 42,
  IPRSA(1,8) = 80, 92,
  IPRSA(1,9) = 11,
```

```
/
&NAMRES
/
&NAMRIP
/
&NAMSCAT
/
&NAMSCC
/
&NAMSENS
/
&NAMSIMPHL
/
&NAMSTA
/
&NAMSTOPH
/
&NAMTOPH
/
&NAMTOVS
/
&NAMTRAJP
/
&NAMTRANS
/
&NAMVAR
/
&NAMVDOZ
/
&NAMVFP
/
&NAMVRTL
/
&NAMVV1
/
&NAMVWRK
/
&NAMXFU
/
&NAMZDI
/
&NAM_CANAPE
  REF_STAT( 1,2) =  7.00,
  REF_STAT( 2,2) =  7.20,
  REF_STAT( 3,2) =  7.60,
  REF_STAT( 4,2) =  7.90,
  REF_STAT( 5,2) =  8.30,
  REF_STAT( 6,2) =  8.70,
  REF_STAT( 7,2) = 10.10,
  REF_STAT( 8,2) = 11.30,
  REF_STAT( 9,2) = 12.70,
  REF_STAT(10,2) = 14.40,
```

```
REF_STAT(11,2) = 15.60,
REF_STAT(12,2) = 16.90,
REF_STAT(13,2) = 19.90,
REF_STAT(14,2) = 22.00,
REF_STAT(15,2) = 24.10,
REF_STAT(16,2) = 26.40,
REF_STAT(17,2) = 29.00,
REF_STAT(18,2) = 31.20,
REF_STAT(19,2) = 35.20,
REF_STAT( 1,3) =  2.50,
REF_STAT( 2,3) =  2.50,
REF_STAT( 3,3) =  2.30,
REF_STAT( 4,3) =  1.80,
REF_STAT( 5,3) =  1.30,
REF_STAT( 6,3) =  1.20,
REF_STAT( 7,3) =  1.20,
REF_STAT( 8,3) =  1.20,
REF_STAT( 9,3) =  1.20,
REF_STAT(10,3) =  1.30,
REF_STAT(11,3) =  1.40,
REF_STAT(12,3) =  1.50,
REF_STAT(13,3) =  1.60,
REF_STAT(14,3) =  1.70,
REF_STAT(15,3) =  1.80,
REF_STAT(16,3) =  1.90,
REF_STAT(17,3) =  1.90,
REF_STAT(18,3) =  2.00,
REF_STAT(19,3) =  2.10,
REF_STAT( 1,4) =  3.70,
REF_STAT( 2,4) =  3.70,
REF_STAT( 3,4) =  3.70,
REF_STAT( 4,4) =  3.10,
REF_STAT( 5,4) =  2.70,
REF_STAT( 6,4) =  2.80,
REF_STAT( 7,4) =  3.10,
REF_STAT( 8,4) =  3.40,
REF_STAT( 9,4) =  3.80,
REF_STAT(10,4) =  4.10,
REF_STAT(11,4) =  4.00,
REF_STAT(12,4) =  4.00,
REF_STAT(13,4) =  3.90,
REF_STAT(14,4) =  3.80,
REF_STAT(15,4) =  3.60,
REF_STAT(16,4) =  4.20,
REF_STAT(17,4) =  5.00,
REF_STAT(18,4) =  5.30,
REF_STAT(19,4) =  6.10,
REF_STAT( 1,5) =  0.20,
REF_STAT( 2,5) =  0.20,
REF_STAT( 3,5) =  0.20,
REF_STAT( 4,5) =  0.20,
REF_STAT( 5,5) =  0.20,
```

```
  REF_STAT( 6,5) =  0.21,
  REF_STAT( 7,5) =  0.21,
  REF_STAT( 8,5) =  0.22,
  REF_STAT( 9,5) =  0.22,
  REF_STAT(10,5) =  0.22,
  REF_STAT(11,5) =  0.22,
  REF_STAT(12,5) =  0.22,
  REF_STAT(13,5) =  0.22,
  REF_STAT(14,5) =  0.22,
  REF_STAT(15,5) =  0.22,
  REF_STAT(16,5) =  0.22,
  REF_STAT(17,5) =  0.22,
  REF_STAT(18,5) =  0.22,
  REF_STAT(19,5) =  0.22,
  REF_S_SST = 0.8,
  REF_S_SN  = 1.0,
  REF_S_T2  = 3.0,
  REF_S_H2  = 0.20,
  REF_S_V1  = 5.,
  REF_A_SST = 300000.,
  REF_A_SN  = 100000.,
  REF_A_T2  =  40000.,
  REF_A_H2  =  40000.,
  REF_A_VOR1=  60000.,
  REF_A_DIV1=  50000.,
  REF_NU_BL = 1.25,
  REF_KP_BL = 400.,
 /
 &NAM_DISTRIBUTED_VECTORS
 /
 &NAPHLC
 /
 &NEMBICU
 /
 &NEMCT0
  NBICOP=2,
  NBICOQ=2,
  NBICOT=2,
  NBICOU=2,
  NECRIPL=0,
 /
 &NEMDIM
 /
 &NEMDYN
 /
 &NEMGEO
 /
FIN

#
# Looking for the executable
# -------------------------------
```

```
cp /u/ch/mxpt/mxpt001/aladin/horaire/dbl/exec/ALADIN  MASTER

#
# Execution
# ------------
MASTER -v$VERSION -eANAL -c$NCONF -t$TSTEP -f$NSTOP -a$ADVEC -m$MODEL

cat NODE.001_01
#cat listing_odb

#
# Output analysis file saving
# --------------------------------
ftput -q ICMSHANAL+0000 J98/analyse/alad_redo_al15_d${dat}

#
# Observations database saving
# -----------------------------------
cd ${ODB_SRCPATH_ECMA}
tar cvf toto *
ftput -q toto J98/obs/odb_redo_d${dat}

ja
```

Annex 2

Description of the statistical model which is coded in CANARI :

1. General definitions :

Guess errors covariances for variable A at point M_1 and variable B at point M_2 are written :

$$\langle A_1,B_2\rangle = \sigma_A \, \sigma_B \, cor(A_1,B_2)$$

with : σ_A standard deviation for the variable A,
σ_B standard deviation for the variable B,
$cor(A_1,B_2)$ correlation between guess error on variable A at point 1 and guess error on variable B at point 2.

Homogeneity, isotropy and separability hypotheses allow us to write :

$$cor(A_1,B_2) = corh(A_1,B_2) * corv(A_1,B_2)$$

$$corh(A_1,B_2) = f_d(r)$$

$$corv(A_1,B_2) = g_p(z)$$

with : $corh(A_1,B_2)$ horizontal correlation,
$corv(A_1,B_2)$ vertical correlation,
$f_d(r)$ function of the horizontal distance r between the point M_1 and the point M_2 with a characteristic length d,
$g_p(z)$ function of the departure z between L_1 (pressure logarithm at point 1) and L_2 (pressure logarithm at point 2) with a charateristic parameter p.

NB : Even if in practice we allow "slow" variations of the variances and the parameters which define correlations with altitude and latitude, these parameters are considered constant in the formal computations.

2. To define the statistical model we use the variables ϕ (geopotential), ψ (streamfunction), χ (potential velocity) and q (specific humidity), with the hydrostatic relation for T (temperature) and Helmotz relation for V (wind vector) :

$$T = -1/R_d * d\phi/dL$$

$$V = V_\psi + V_\chi = k \wedge \nabla\psi + \nabla\chi$$

with : $dL=d(\ln P)=dP/P$,
R_d the perfect gaz constant for dry air
($R_d=6.0221367\ 10^{+23} * 1.380658\ 10^{-23} / 28.9644\ 10^{-3} = 287.0596$ J/K/Kg),

V_ψ the rotational part of the wind,
V_χ the divergent part of the wind.

We define u_i^l as the longitudinal component (component of wind tangent to the arc from the point M_1 to the point M_2) and u_i^t as the transverse component (direct orthogonal component to u_i^l) :

$$u_i^l = u_i \cos \theta_i + v_i \sin \theta_i$$

$$u_i^t = -u_i \sin \theta_i + v_i \cos \theta_i$$

with : u_i eastward component,
v_i northward component,
θ_i angle between East at point i and arc from point M_1 to point M_2.

Then it will be possible to express easily $\nabla\psi$ and $\nabla\chi$ correlations from ψ and χ correlations using on the sphere latitude and longitude for which arc from point M_1 to point M_2 is on the equator. Then for the point M_i latitude is zero and longitude is "λ_i" instead of geographical latitude "lat_i" and longitude "lon_i".

With : R_t the radius of the earth (R_t =6371229m),

$$\cos \alpha = \langle OM_1 \mid OM_2 \rangle / R_t{**}2$$
$$= \cos(lat_1) \cos(lat_2) \cos(lon_2-lon_1) + \sin(lat_1) \sin(lat_2)$$
$$= 0.5 \, [\cos(lat_2-lat_1) (\cos(lon_2-lon_1)+1) + \cos(lat_2+lat_1) (\cos(lon_2-lon_1)-1)],$$

$$A = sqrt(1-(\cos \alpha){**}2) = |\sin\alpha|$$

$$r = R_t \, \alpha \sim 2R_t \sin (\alpha/2) \sim R_t \, A \qquad r=R_t{*}|\lambda_2 - \lambda_1| = R_t * sign(\lambda_2 - \lambda_1) * (\lambda_2 - \lambda_1)$$

$$dr/d\lambda_1 =-R_t * sign(\lambda_2 - \lambda_1)$$

$$dr/d\lambda_2 = R_t * sign(\lambda_2 - \lambda_1)$$

$$dr/d\phi_i = 0$$

$$d^2r/d\phi_1 \, d\phi_2 \,|_{\phi_1=0 \text{ and } \phi_2=0} = -R_t{**}2 / R_t \, A \sim -R_t{**}2 / r$$

We have :

$$\cos(\theta 1) = \cos(lat_2) \sin(lon_2\text{-}lon_1) / A$$
$$\cos(\theta 2) = \cos(lat_1) \sin(lon_2\text{-}lon_1) / A$$
$$\sin(\theta 1) = [\sin(lat_2) \cos(lat_1) - \sin(lat_1) \cos(lat_2) \cos(lon_2 \text{-}lon_1)] / A$$
$$= 0.5 [\sin(lat_2\text{-}lat_1) (\cos(lon_2\text{-}lon_1)+1) - \sin(lat_2+lat_1) (\cos(lon_2\text{-}lon_1)\text{-}1)] / A$$
$$\sin(\theta 2) = [-\sin(lat_1) \cos(lat_2) + \sin(lat_2) \cos(lat_1) \cos(lon_2\text{-}lon_1)] / A$$
$$= 0.5 [\sin(lat_2\text{-}lat_1) (\cos(lon_2\text{-}lon_1)+1) + \sin(lat_2+lat_1) (\cos(lon_2\text{-}lon_1)\text{-}1)] / A$$

and

$$<u_1,u_2> = \cos(\theta_1) \cos(\theta_2) <u_1^l,u_2^l> + \sin(\theta_1) \sin(\theta_2) <u_1^t,u_2^{\,t}>$$
$$<v_1,v_2> = \sin(\theta_1) \sin(\theta_2) <u_1^l,u_2^l> + \cos(\theta_1) \cos(\theta_2) <u_1^t,u_2^{\,t}>$$
$$<u_1,v_2> = \cos(\theta_1) \sin(\theta_2) <u_1^l,u_2^l> - \sin(\theta_1) \cos(\theta_2) <u_1^t,u_2^{\,t}>$$
$$<v_1,u_2> = \sin(\theta_1) \cos(\theta_2) <u_1^l,u_2^l> - \cos(\theta_1) \sin(\theta_2) <u_1^t,u_2^{\,t}>$$

NB : $<v_2,u_1>=<u_1,v_2>$ different from $<v_1,u_2>=<u_2,v_1>$
except if $\theta_2\text{-}\theta_1=0$ or π, i.e. $lon_2=lon_1$ or $lon_2=lon_1+\pi$.

3. We set :

$$<\phi_1,\phi_2> = \sigma_{\phi_1} * \sigma_{\phi_2} * f_a(r) * g_k(z)$$

$$<\psi_1,\psi_2> = \sigma_{\psi_1} * \sigma_{\psi_2} * f_a(r) * g_k(z)$$

$$<\psi_1,\phi_2> = \mu * \sigma_{\psi_1} * \sigma_{\phi_2} * f_a(r) * g_k(z)$$

$$<\phi 1,\psi 2> = \mu * \sigma_{\phi_1} * \sigma_{\psi_2} * f_a(r) * g_k(z)$$

$$<\chi_1,\chi_2> = \sigma_{\chi_1} * \sigma_{\chi_2} * f_b(r) * g_k(z)$$

$$<\chi_1,\phi_2> = <\phi_1,\chi_2> = 0$$

$$<\chi_1,\psi_2> = <\psi_1,\chi_2> = 0$$

$$<\theta_1,\theta_2> = \sigma_{\theta_1} * \sigma_{\theta_2} * f_c(r) * g_l(z)$$

$$<\theta_1,\phi_2> = <\phi_1,\theta_2> = 0$$

$$<\theta_1,\psi_2> = <\psi_1,\theta_2> = 0$$

$$<\theta_1,\chi_2> = <\chi_1,\theta_2> = 0$$

μ is a coefficient related to the geostrophism. The characteristic length d used by the function f_d is a for the geopotential and the streamfunction, b for the potential velocity and c for the specific humidity. The characteristic length p used by the function g_p is k for the geopotential, the streamfunction and the potential velocity, and l for the specific humidity.

Then using hydrostatism and Helmotz relation :

$$<T_1,T_2> = \sigma_{\phi_1}/R_d * \sigma_{\phi_2}/R_d * f_a(r) * d^2g_k/dL_1dL_2(z)$$

$$<u_1^l,u_2^l> = - [\sigma_{\psi_1} * \sigma_{\psi_2}/r * df_a/dr(r) + \sigma_{\chi_1} * \sigma_{\chi_2} * d^2f_b /dr^2(r)] * g_k(z)$$

$$<u_1^t,u_2^t> = - [\sigma_{\psi_1} * \sigma_{\psi_2} * d^2f_a/dr^2(r) + \sigma_{\chi_1} * \sigma_{\chi_2}/r * df_b/dr(r)] * g_k(z)$$

$$<u_1^l,u_2^t> = <u_1^t,u_2^l> = 0$$

$$<u_1^l,\phi_2> =<\phi_1,u_2^l> =0$$

$$<u_1^t,\phi_2> = -\mu * \sigma_{\psi_1} * \sigma_{\phi_2} * df_a/dr(r) * g_k(z)$$

$$<\phi_1,u_2^t> = \mu * \sigma_{\phi_1} * \sigma_{\psi_2} * df_a/dr(r) * g_k(z)$$

We also lay down :

$$\sigma_\psi = \sigma_\phi / f = \sigma_\phi / \max(2\Omega \sin(lat),\Omega)$$

$$\sigma_{u\chi} = \nu * \sigma_{u\psi}$$

with : Ω the rotation of the earth with respect to stars
($\Omega = 2\pi/86400 * [1. +1./ (365.25*2\pi/6.283076)] \sim 2\pi/86164 =7.2921151\ 10^{-5}$ rd/s).

4. We fix the horizontal and the vertical correlation functions :

$$f_d(r) = exp(-1/2*(r/d)**2)$$

$$g_p(z) = 1. / (1.+ p z**2)$$

then we have :

$$lim_{2->1}<T_1,T_2> = lim_{1->2} [\sigma_{\phi_1}/R*\sigma_{\phi_2}/R*exp(-1/2*(r/a)**2)*2*k*(1-3k*z**2)/(1+k*z**2)**3]$$

$$\sigma_{T_1} **2 = (\sigma_{\phi_1}/R)**2 * 2 * k$$

in a similar way we obtain : $\sigma_u = \sigma_v =$ sqrt $1/2*[(\sigma_{u\psi})^2+(\sigma_{u\chi})^2]$ = sqrt$[(\sigma_\psi/a)^2+(\sigma_\chi/b)^2]$

$$\sigma_T = \sigma_\phi / R_d * sqrt(2*k)$$

$$\sigma_\chi = \nu * b/a * \sigma_\psi$$

$$\sigma_u = \sigma_v = sqrt(1+\nu**2) * (\sigma_\psi/a)$$

and then

$$<T_1,T_2> = \sigma_{T_1} * \sigma_{T_2} * exp(-1/2*(r/a)^2) * (1-3*k*z^2)/(1+k*z^2)^3$$

$$<T_1,\phi_2> = \sigma_{T_1} * \sigma_{\phi_2} * sqrt(2*k) * exp(-1/2*(r/a)^2 * z/(1+k*z^2)^2$$

$$<\phi_1,T_2> = -\sigma_{\phi_1} * \sigma_{T_2} * sqrt(2*k) * exp(-1/2*(r/a)^2 * z/(1+k*z^2)^2$$

$$<T_1,u_2>= \mu * \sigma_{T_1} * \sigma_{u_2} * sqrt(2*k) * sin(\theta_2) * r/a * exp(-1/2*(r/a)^2) * [z/(1+k*z^2)^2]$$

$$<T_1,v_2>=-\mu * \sigma_{T_1} *\sigma_{v_2} * sqrt(2*k) * cos(\theta_2) * r/a * exp(-1/2*(r/a)^2) * [z/(1+k*z^2)^2]$$

$$<u_1,T_2>= \mu * \sigma_{u_1} * \sigma_{T_2} * sqrt(2*k) * sin(\theta_1) * r/a * exp(-1/2*(r/a)^2) * [z/(1+k*z^2)^2]$$

$$<v_1,T_2>=-\mu * \sigma_{v_1} * \sigma_{T_2} *sqrt(2*k) * cos(\theta_1) * r/a * exp(-1/2*(r/a)^2) * [z/(1+k*z^2)^2]$$

$$<u_1{}^l,u_2{}^l> = \sigma_{u_1}*\sigma_{u_2}/(1+\nu^2)*[exp(-1/2*(r/a)^2)+\nu^2*(1-(r/b)^2)*exp(-1/2*(r/b)^2)]*[1/(1+k*z^2)]$$

$$<u_1{}^t,u_2{}^t> = \sigma_{u_1}*\sigma_{u_2}/(1+\nu^2)*[(1-(r/a)^2)*exp(-1/2*(r/a)^2)+\nu^2*exp(-1/2*(r/b)^2)]*[1/(1+k*z^2)]$$

$$\langle u_1,u_2\rangle = \sigma_{u_1}*\sigma_{u_2}/(1+\nu^2)*\{[\cos(\theta_1)\cos(\theta_2)+\sin(\theta_1)\sin(\theta_2)(1-(r/a)^2)]*\exp(-1/2*(r/a)^2) + \nu^2*[\cos(\theta_1)\cos(\theta_2)(1-(r/b)^2)+\sin(\theta_1)\sin(\theta_2)]*\exp(-1/2*(r/b)^2)]\} * [1/(1+k*z^2)]$$

$$\langle v_1,v_2\rangle = \sigma_{u_1}*\sigma_{u_2}./(1+\nu^2)*\{[\sin(\theta_1)\sin(\theta_2)+\cos(\theta_1)\cos(\theta_2)(1-(r/a)^2)]*\exp(-1/2*(r/a)^2) + \nu 2*[\sin(\theta_1)\sin(\theta_2)(1-(r/b)^2) + \cos(\theta_1)\cos(\theta_2)]*\exp(-1/2*(r/b)^2)]\} * [1/(1+k*z^2)]$$

$$\langle u_1,v_2\rangle = \sigma_{u_1}*\sigma_{u_2}./(1+\nu^2)*\{[\cos(\theta_1)\sin(\theta_2) - \sin(\theta_1)\cos(\theta_2)(1-(r/a)^2)]*\exp(-1/2*(r/a)^2) + \nu^2 *[\cos(\theta_1)\sin(\theta_2)(1-(r/b)^2) - \sin(\theta_1)\cos(\theta_2)]*\exp(-1/2*(r/b)^2)]\} * [1/(1+k*z^2)]$$

$$\langle v_1,u_2\rangle = \sigma_{u_1}*\sigma_{u_2}/(1+\nu^2)*\{[\sin(\theta_1)\cos(\theta_2) - \cos(\theta_1)\sin(\theta_2)(1-(r/a)^2)] *\exp(-1/2*(r/a)^2) + \nu^2 *[\sin(\theta_1)\cos(\theta_2)(1-(r/b)^2) - \cos(\theta_1)\sin(\theta_2)]*\exp(-1/2*(r/b)^2)]\} * [1/(1+k*z^2)]$$

$$\langle u_1,\phi_2\rangle = -\mu*\sigma_{u_1} *\sigma_{\phi_2} * [1./\mathrm{sqrt}(1+\nu^2)] * \sin(\theta_1) * r/a * \exp(-1/2*(r/a)^2 *[1/(1+k*z^2)]$$

$$\langle v_1,\phi_2\rangle = \mu*\sigma_{v_1} *\sigma_{\phi_2} * [1./\mathrm{sqrt}(1+ \nu^2)] * \cos(\theta_1) * r/a * \exp(-1/2*(r/a)^2 *[1/(1+k*z^2)]$$

$$\langle \phi_1,u_2\rangle = \mu*\sigma_{\phi_1} *\sigma_{u_2} * [1/\mathrm{sqrt}(1+\nu^2)] * \sin(\theta_2) * r/a * \exp(-1/2*(r/a)^2 *[1/(1+k*z^2)]$$

$$\langle \phi_1,v_2\rangle = -\mu*\sigma_{\phi_1} *\sigma_{v_2} * [1/\mathrm{sqrt}(1+\nu^2)] * \cos(\theta_2) * r/a * \exp(-1/2*(r/a)^2 *[1/(1+k*z^2)]$$

at last for thickness : $\Delta\phi = \phi s - \phi i$

$$\sigma_{\Delta\phi}{}^2 = \sigma_{\phi s}{}^2 + \sigma_{\phi i}{}^2 - 2\,\sigma_{\phi s}*\sigma_{\phi i}*g_k(z) = \sigma_{\phi s}{}^2 + \sigma_{\phi i}{}^2 - 2\,\sigma_{\phi s}*\sigma_{\phi i}*[1/(1+k*z^2)]$$

$$\mathrm{cor}(\Delta\phi_1, \Delta\phi_2) = [\sigma_{\phi s_1}*\sigma_{\phi s_2} *\mathrm{cor}(\phi s_1,\phi s_2) + \sigma_{\phi i_1}*\sigma_{\phi i_2} *\mathrm{cor}(\phi i_1,\phi i_2) - \sigma_{\phi s_1}*\sigma_{\phi i_2}*\mathrm{cor}(\phi s_1, \phi i_2) - \sigma_{\phi i_1}*\sigma_{\phi s_2} *\mathrm{cor}(\phi i_1,\phi s_2)] / [\sigma_{\Delta\phi_1}*\sigma_{\Delta\phi_2}]$$

$$\mathrm{cor}(\Delta\phi_1, B_2) = [\sigma_{\phi s_1} * \mathrm{cor}(\phi s_1,B_2) -\sigma_{\phi i_1} * \mathrm{cor}(\phi i_1,B_2)] / \sigma_{\Delta\phi_1}$$

$$\mathrm{cor}(B_1,\Delta\phi_2) = [\sigma_{\phi s_2} * \mathrm{cor}(B_1,\phi s_2) - \sigma\phi i_2 * \mathrm{cor}(B_1,\phi i_2)] / \sigma_{\Delta\phi_2}$$

5. CANARI statistical model is then determined by the standard errors σ_ϕ and σ_q, the horizontal length scales a, b and c, the inverse of the vertical characteristic sizes k and l, the geostrophic coefficient μ and the divergent coefficient ν.

We allow slow variations with latitude and altitude of the statistical model parameters. In practice only σ_ϕ, σ_{Hu}, a, k, μ and ν are prescribed; the other parameters are linked to formers. σ_q is computed from σ_{Hu} using $q_{sat}(T_{standard})$.

pressure (hPa)	σ_ϕ (mgp)	σ_T (K)	σ_u/σ_v (m/s)	σ_{Hu} (%)	a (km)	k (hPa)	dp
1000.	8.0	1.90	2.30	17.	470.	24.17	205.
950.	8.2	1.80	2.30	18.	480.	20.64	211.
900.	8.6	1.70	2.30	19.	500.	16.74	222.
850.	9.0	1.60	2.30	20.	530.	13.54	234.
800.	9.4	1.50	2.40	20.	530.	10.91	246.
700.	9.9	1.40	2.50	21.	530.	8.57	244.
600.	11.5	1.40	2.75	21.	560.	6.35	244.
500.	12.8	1.40	3.00	22.	570.	5.13	228.
400.	14.4	1.40	3.30	22.	590.	4.05	207.
300.	16.4	1.50	3.60	22.	610.	3.58	166.
250.	17.7	1.60	3.50	22.	680.	3.50	140.
200.	19.2	1.70	3.50	22.	740.	3.36	115.
150.	22.6	1.80	3.40	22.	890.	2.72	97.
100.	25.0	1.90	3.30	22.	1020.	2.47	68.
70.	27.4	2.00	3.20	22.	1150.	2.28	50.
50.	30.0	2.10	3.70	22.	1090.	2.10	37.
30.	33.0	2.20	4.40	22.	1010.	1.90	24.
20.	35.5	2.30	4.70	22.	1020.	1.80	16.
10.	40.0	2.40	5.40	22.	1000.	1.54	9.

μ = [1.0 - max(0., 0.15 * (P(hPa)-900./100.)] * sin(lat)
ν = [0.3 + max(0., 0.6 * (P(hPa)-900./100.)] * max(cos(lat), sqrt(3)/2)

b = 0.7 * a
c = 0.8 * a
l = 6 * k

σ_ϕ (lat) = coef * σ_ϕ with coef = 1.4 for lat $\in$ [-90, -25]
coef = 0.6 for lat $\in$ [-15, +15]
coef = 1.2 for lat $\in$ [+25, +90]
and linear variation of coef between [-25, -15] and [+15, +25].

6. The stretching is taken into account with the functions $s_1(m)$ and $s_2(m)$ decreasing as the local map factor m increases. We assume with respect to the stretching :

$$\sigma_\psi(m) / \sigma_\phi(m) = cte$$

$$\sigma_\chi(m) = cte$$

$$b(m) / a(m) = cte$$

$$\sigma_q(m) / \sigma_\phi(m) = cte$$

then

$\sigma_\phi(m) = \sigma_\phi * s_1(m)$
$\sigma_\psi(m) = \sigma_\psi * s_1(m)$
$\sigma_T(m) = \sigma_T * s_1(m)$
$\nu(m) = \nu / s_1(m)$
$a(m) = a * s_2(m)$
$\sigma_u(m) = \sigma_v(m) = \sigma_u * s_1(m) / s_2(m) * \mathrm{sqrt}[(1+\nu/s_1(m))^2/(1+\nu^2)]$

As the local resolution increases, horizontal characteristic length scale of errors, mass variables and humidity standard deviations decrease, but error on the divergent part of the flow remains the same. The size of the errors decreases slower than the intensity of the errors.

We have chosen $s_2(m) = \exp(-\alpha(m-1/m))$ and $\alpha=0.08$. That leads to 30% of variation for the characteristic size from one pole to equator with a stretching coefficient of 3.5.

7. The analysis gives at each grid point an estimate of $\sigma_{Hu_{an}}$ and an estimate of $\sigma_{\phi_{an}}$ from

$$\sigma_{\phi_{an}} = \sigma_{\psi_{an}} * \max(2\Omega\sin(lat),\Omega), \text{ with } \sigma_{\psi_{an}}{}^2 = 0.5 * (\sigma_{u_{an}}{}^2+\sigma_{v_{an}}{}^2) * a^2 / (1+\nu^2).$$

This estimate $\sigma_{X_{an}}$ is used in a linear combination with $\sigma_{X_{clim}}$ to simulate the increase of error during the forecast, and we impose the result of the linear combination to be smaller than $\sigma_{X_{clim}}$ and greater than $\lambda_X{*}\sigma_{X_{clim}}$. At last σ_ϕ is horizontally filtered in the spectral space (for NSMAX the amplitude of the standard deviation is divided by 10^{+4}) and vertically using the correlation function (not used).

$\alpha_{cl} = 0.05$ * (length of the forecast in hours) / 6.

$\sigma_{\phi_{clim}} = 3. * \sigma_\phi\ (m)$

$\lambda_\phi = 0.3$

$\sigma_{\phi_{guess}}{**}2 = \max[(1-\alpha_{cl}) * \min(\sigma_{\phi_{an}}{}^2,\sigma_{\phi_{clim}}{}^2) + \alpha_{cl} *\sigma_{\phi_{clim}}{}^2, (\lambda_\phi{*}\sigma_{\phi_{clim}})^2]$

$\sigma_{\phi_{guess}}(n) = \sigma_{\phi_{guess}}(n) * \exp[-0.5 * cx * (n/nsmax)^2]$ with $cx = \ln(10^{+8})$

$\sigma_{\phi_{guess}}(level\_i) = \Sigma_j\, \sigma_{\phi_{guess}}(level\_j) * w_{ij} / \Sigma_j\, w_{ij}$ with $w_{ij}=\beta{*}corv[\phi(level\_i),\phi(level\_j)]$

$j \in [i-j_{inf}, i\ [\ U\ ]\ i, i+j_{sup}]$, $\beta=?$ and $w_{ii}=1$

$\sigma_{Hu_{clim}} = 1.2 \; * \sigma_{Hu}(m)$

$\lambda_{Hu} = 0.2$

$\sigma_{Hu_{guess}} = \max[(1-\alpha_{cl})*\sigma_{Hu_{an}} + \alpha_{cl}*\sigma_{Hu_{clim}} , \lambda_{Hu}*\sigma_{Hu_{clim}}]$

8. For the boundary layer parameters (T_{2m}, Hu_{2m}, U_{10m}, V_{10m}), snow and SST, specific statistical models are defined. There ara gaussian on the horizontal with characteristics fixed in namelist.

σ_{SST} = 0.8 K	a_{SST} = 300 km		
σ_{snow} = 1.0 cm	a_{snow} = 50 km		
σ_{T_2} = 3.0 K	a_{T_2} = 50 km		
σ_{Hu_2} = 30 %	a_{Hu_2} = 50 km		
$\sigma_{u_{10}}/\sigma_{v_{10}}$ = 6 m/s	a_{vor} = 50 km	a_{div} = 50 km	ν_bl = 1

There are no cross-correlation between different parameters. We choose the ratio between the error on the divergent part of the flow and the error on the rotational part of the flow to be one, with the same characteristic lengths for the vorticity and the divergence. That gives a local impact for a ten meters wind observation.

On the vertical the auto-correlation is always one (the analysis is done on height surface), but to allow the use of boundary layer parameters in upperair analysi, we define a vertical correlation between U/V/T and U_{10}/V_{10} and T_2 with a characteristic parameter height enough to limit in the boundary layer the impact of a surface observation.

kp_bl = 100.

9. The following standard error are prescribed for the observations.

NB : σ_q is computed from σ_{Hu} using $q_{sat}(T_{obs})$.

	SYNOP	SHIP	DRIBU
σ_ϕ (mgp)	8	8	8
σ_T (K)	1.4	1.4	1.4
$\sigma_u = \sigma_v$ (m/s)	2.0	3.0	3.0
σ_{Hu} (%)	10	10	
σ_{SST} (K)	1.5	1.5	1.5

For the SATOB wind the value prescribed for the guess is used with a factor f_{sat} depending of the satellite :

$f_{meteosat} = 1.1$
$f_{goes} = 1.2$
$f_{himawari} = 1.3$
$f_{indsat} = 1.4$

For SATEM thickness standard deviations are :

Layer (hPa)	Standard deviation (mgp)
1000 - 700	18
700 - 500	16
500 - 300	23
300 - 100	49
100 - 50	40
50 - 30	32
30 - 10	69

and for precipitable water content :

Layer (hPa)	Standard deviation (%)
1000 - 700	14
700 - 500	17
500 - 300	20

For TEMP and AIREP standard deviation are prescribed by levels :

pressure (hPa)	TP σ_ϕ (mgp)	TP σ_T (K)	TP σ_u/σ_v (m/s)	TP σ_{Hu} (%)	AI σ_u/σ_v (m/s)	AI σ_T (K)
1000.	8.0	1.70	2.30	12.	2.50	1.90
950.	8.2	1.60	2.30	12.	2.50	1.80
900.	8.6	1.50	2.30	12.	2.50	1.70
850.	9.0	1.40	2.40	12.	2.60	1.60
800.	9.4	1.40	2.50	12.	2.70	1.50
700.	9.9	1.30	2.50	12.	2.80	1.40
600.	11.4	1.30	2.80	12.	3.20	1.40
500.	12.7	1.30	3.00	12.	3.40	1.40
400.	14.0	1.30	3.30	12.	3.80	1.40
300.	16.0	1.40	3.60	12.	4.00	1.50
250.	15.7	1.50	3.70	12.	4.10	1.60
200.	17.2	1.50	3.80	12.	4.20	1.70
150.	20.1	1.60	3.80	12.	4.20	1.80
100.	22.0	1.60	3.80	12.	4.20	1.90
70.	24.4	1.60	3.80	12.	4.20	2.00
50.	27.0	1.70	3.90	12.	4.30	2.10
30.	30.0	1.80	4.10	12.	4.60	2.20
20.	31.5	1.90	4.30	12.	4.80	2.30
10.	36.0	2.00	4.50	12.	5.00	2.40

For PILOT winds the same standard deviations as for TEMP winds are used.