

# **CANARI**

## **Some technical features**

Françoise Taillefer  
METEO-France

NetFAM working days – Oslo – March 2009

# Plan

- Introduction
- General overview
- Code organisation
- Conclusion and perspectives

# Introduction

# CANARI genesis

- Analysis code developed for the ARPEGE model
- To have an initial state for all the model variables
- For both surface and altitude fields
- Optimal interpolation (start coding in 1988)
- Integrated in the ARPEGE/IFS software
- Used operationally since 1992

# Code evolution

- 20 years of evolution

 **structure** and scientific improvements

1993 : adaptation to LAM (CANARI ALADIN birth)

1995 : from F77 to F90

1997 : from shared memory to distributed memory

1998 : ISBA for soil moisture and soil temperature analysis

1999 : diagpack introduction

2001 : ODB implementation (instead of CMAFOC)

2006 : new surface fields data flow

# Operational use

- CANARI is used in operations in several countries, mainly in its ALADIN version
  
- At METEO-France :
  1. Global surface analysis for ARPEGE
  2. Limited area surface analysis for ALADIN France
  3. Hourly diagnostic analysis over a smaller France domain
- 1. Expected next year in the AROME system (T2m, Hu2m and SST analyses)

# General overview

- some reminders -

# Optimum interpolation - basis theory -

$$\underline{X}_a = \underline{X}_b + PH^t(O+HPH^t)^{-1}(Y-H\underline{X}_b)$$

where  $\underline{X}$  is the atmosphere state vector (analysed or background),  
 $Y$  the observations vector,  
 $H$  the observation operator (supposed to be linear),  
 $P$  the background error variance-covariance matrix,  
 $O$  the observation error variance-covariance matrix.



# Optimum interpolation

## - general settings -

- Background : 6h model forecast
- Observations : only conventional data
- The state vector is represented by a limited number of variables (the model ones), each variable is computed separately (increment).
- The equation is solved point by point (linear system), on the model grid, at each model level.
- Statistical model defined for each predictand/predictor type.
- Selection of the most informative predictors.

# Analysed variables (predictands)

- surface pressure } multivariate
- upperair wind and temperature } analyses
- upperair relative humidity
- 2 meters temperature }
- 2 meters relative humidity } monovariate
- 10 meters wind } analyses
- sea surface temperature }
- snow

# Diagnostic variables

- Over land, the following model variables are estimated from the BL increments, according to the ISBA scheme :
  1. surface temperature,
  2. soil temperature,
  3. surface water content,
  4. deep soil water content.
- In addition, relaxation towards a climatology (also for SST and snow fields).

# Predictors

- Upperair observed parameters :
  - U, V
  - Z, DZ
  - T
  - RH
- BL parameters :
  - T at 2m
  - RH at 2m
  - U, V at 10m
- Surface parameters :
  - snow
  - RR
  - SST

# Code organisation

# Main CANARI features

- Inputs : various data
- Checkings and internal initialisations
- *Observation departures computation*
- *Spatial quality control of the observations*
- Synthesis of the previous tests and obs rejection flag settings
- *Analysis of the specified variables*
- ISBA variables and observations informations updating
- Outputs

# Inputs / Outputs

- As CANARI is included in the ARPEGE/ALADIN software, the IO routines are common, for both FA use -file reading (background, climatology) and file writing (analysis)-, and ODB acces (observations), with all the facilities for distribution (or gathering) of the data between several processors/nodes.
- In the same way, all the model checking processes are run (configuration and namelists coherence, geometry, date, processors, ...)
- CANARI works only in grid-point space.

# Data distribution

- Two types of data : the model (represented by the grid points) and the observations.
- ODB : the database is not in memory during the whole execution time, it is opened, partially read (and updated if required), then closed by each main action of the program. More often, the observations are distributed among the processors (NOBTOT).
- GP : at first, the grid points to deal with are distributed among the processors, by equivalent sets (NGPTOT) of successive points (according to the grid definition).



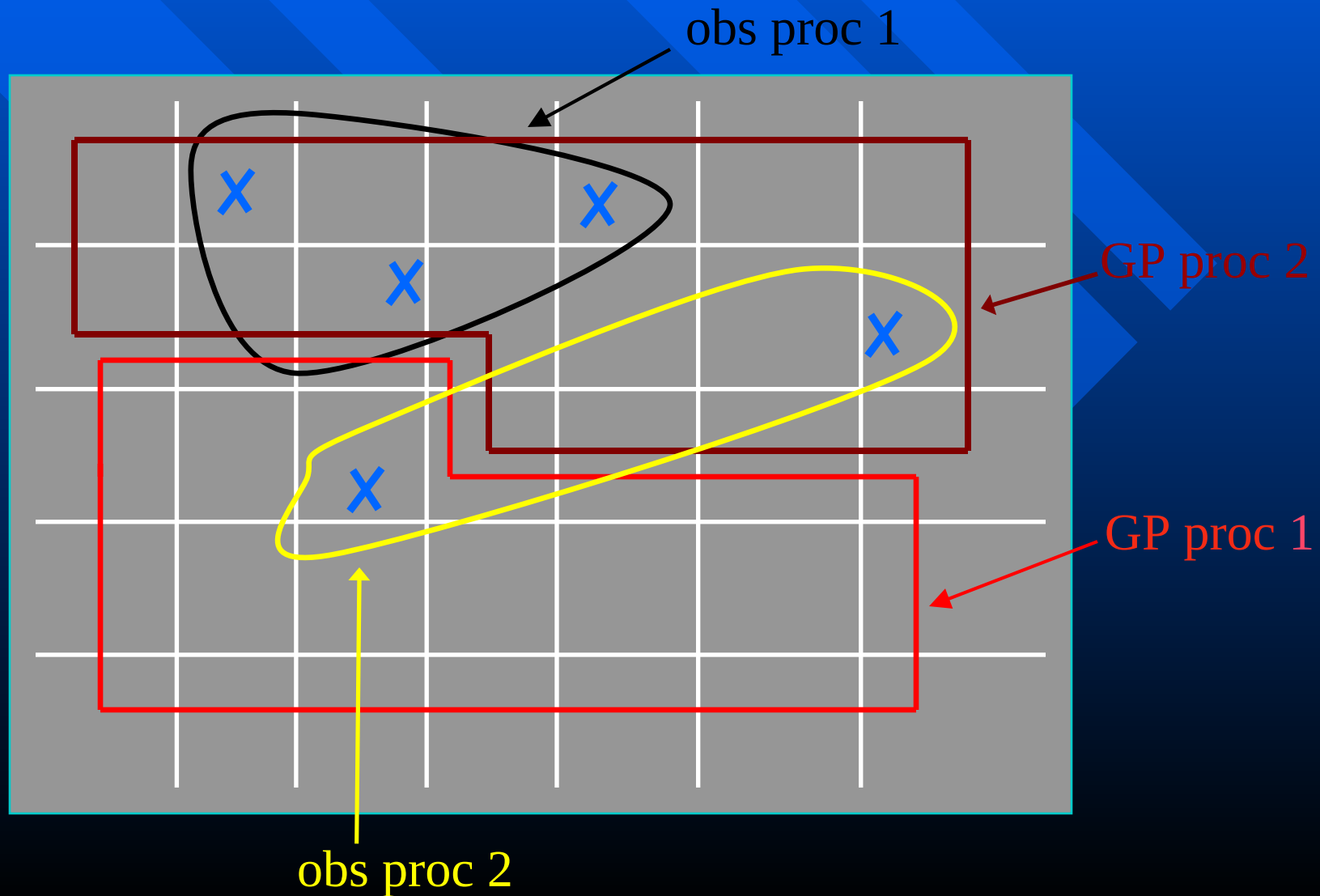
# Observation departures

- Shared code with 3Dvar and 4Dvar (to some extent)
- Observation departures computed in 2 steps :
  - horizontal interpolation of the background fields to each obs position,
  - model equivalent estimation for each obs parameter (vertical interpolation with an obs operator if necessary).
- Very hard to understand from a code organisation point of view due to the distribution of the data among the processors :
  - input grid-point fields distributed according to the geometrical order of the model grid,
  - observations distributed according to the geographical order,



need for a sophisticated data exchange package

# Data distribution example



# Observation spatial quality control

- Innovative idea developed in CANARI.
- The goal is to use as many observations as possible.
- It allows more flexibility to accept observations, despite the result of the first-guess test, if there is at least one another observation in the surroundings which has an obs departure of a similar « big » value.
- It consists in performing an analysis at the obs point but without using the obs which is being controlled.
- The code used to do this is the same than the one used to analyse the model variables (the OI core), inside a loop over all the NOBTOT observations -no need of information from the grid points-

# Model variable analyses

- One « master call » routine for each analysed variable sequentially called.

CANARI —  CAPOTX —

  
CAPSAX  
CAVTAX  
CAHUAX  
CAT2AS  
CAH2AS  
CAV1AS  
(CASNAS)  
CASTAS  
CACSTS

 internal loop  
over the  
NPROMA  
points




loop over the packets of  
NPROMA size from the  
own set of points of each  
processor (NGPTOT)

# Analysis routines

- All these « master call » routines are built on the same principle : fix the considered predictand, select the information for the authorized predictors, initialize the first-guess errors and then call the OI core package!
- To avoid problems due to the distribution of observations, at this stage each processor has in memory the needed information about **all** the available observations -option offered by ODB, used in CANARI because of the relative small number of observations-

# OI core

- The master routine of the OI core code is CANEVA.
- Independant part of code, modular, all the information is passed through arguments, the only common is the observation one (due to ODB use).
- Inputs : predictand/predictors characteristics
- Main actions :
  - \* geographical selection among the observations,
  - \* statistical selection among the predictors,
  - \* correlations computation and matrices building,
  - \* system solving  MINV
- Outputs : analysis increment, analysis error standard deviation
- Older part of the code, few modifications, only when a new predictor or predictand is added.

# Conclusion

- CANARI is a very efficient and robust software.
- It is a sophisticated tool but easy to install and to use.
- It can perform the analysis of any variable
  - » add the appropriate observation type if necessary,
  - » develop the corresponding obs operator involved,
  - » define the associated statistical model.

# Perspectives

- Scientific improvements :
  - \_ SST analysis with use of satellite retrievals,
  - \_ snow analysis development,
  - \_ statistical model refinement.
  
- Code evolution :
  - \_ add an option which allows to run only in 2D,
  - \_ SURFEX.