



# Assimilation of Lagrangian Data Documentation of the observation operator. Demonstrator for the twin experiments

Claire Chauvin, Arthur Vidard, Maëlle Nodet

## ► To cite this version:

Claire Chauvin, Arthur Vidard, Maëlle Nodet. Assimilation of Lagrangian Data Documentation of the observation operator. Demonstrator for the twin experiments. [Contract] D2.3.1, 2010, pp.16. hal-00941640

**HAL Id: hal-00941640**

**<https://inria.hal.science/hal-00941640>**

Submitted on 4 Feb 2014

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

*Deliverable D2.3.1:*

## **Assimilation of Lagrangian Data**

### **Documentation of the observation operator. Demonstrator for the twin experiments**

*Claire Chauvin, Maëlle Nodet and Arthur Vidard*

*INRIA/LJK, Grenoble*

## Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Documentation of the observation operator</b>	<b>3</b>
2.1	Assimilation of drifter float positions in the context of NEMOVAR . . . . .	3
2.2	Construction of the observation operator $\mathcal{H}$ . . . . .	3
2.3	Construction of an interpolation operator adapted to differentiation . . . . .	4
2.4	Tangent and adjoint observation operators . . . . .	5
2.5	An adapted implementation of grid_search for Lagrangian data. . . . .	7
<b>3</b>	<b>Reference Manual of the Lagrangian Observation Operator in NEMO / NEMOVAR</b>	<b>7</b>
3.1	Lagrangian observation operator modules in NEMO . . . . .	7
3.2	The Lagrangian data structure . . . . .	8
3.3	The Lagrangian observation operator . . . . .	9
3.4	Tangent and adjoint models of the Lagrangian Observation Operator . . . . .	10
3.5	Lagrangian Observation Error . . . . .	12
<b>4</b>	<b>Demonstrator for the twin experiments</b>	<b>12</b>
<b>A</b>	<b>APPENDIX : Subtasks, milestones and deliverables for Task 2.3</b>	<b>15</b>

# 1 Introduction

In the framework of VODA, the task 2.3 has been set up in order to develop specific Observation Operators: **Assimilation of Lagrangian Data**. As an introduction to this deliverable, we first recall the main objectives of this task.

In the framework of the Argo program, profiling drifting floats are now routinely launched in the world's oceans. These floats provide (among other information) data about their position, sampled every ten days, representative of their Lagrangian drift. Previous works by M. Nodet and J. Blum (Nodet 2005, Nodet 2006, Blum and Nodet 2006) have shown the interest of assimilating this new type of data.

The assimilation of Lagrangian-type data such as position of drifting floats is not straightforward; it involves the careful implementation of a complex, non-linear observation operator but it is of importance for an application in operational oceanography. For instance this work is useful for localized experiments with surface-drifting floats: the information provided by the Lagrangian data could help to improve any operational system aiming to forecast currents for various applications with significant societal impact, such as coastal rescue or pollutant spreading. The sequence of images (pictures from satellite) provides another important source of Lagrangian data: indeed, it describes the displacement of Lagrangian structures (vortex, fronts ...). Assimilation of images is beyond the scope of this task and is targeted in the ANR project ADDISA and its follow-up GEO-Fluids, but experience gained within the VODA project will open the way to assimilation of any kind of Lagrangian data in NEMOVAR. Finally, this work could provide guidance for the design of the observational network (distribution of large-scale Argo floats / number of localized surface floats / drifting depth, and so on).

The final objective is of course to work with real data, but several issues have to be addressed before real data assimilation is feasible, the different steps are detailed in Appendix A.

## 2 Documentation of the observation operator

### 2.1 Assimilation of drifter float positions in the context of NEMOVAR

In the context of NEMOVAR, a structure of observation operator has already been set up, and the idea here is to use this formalism to implement the Lagrangian data structure. The variational assimilation needs several ingredients: a direct model, an observation operator that translates the observation variables into state variables, in order to construct a cost function. The iterative method used to minimize this cost function is based on a gradient method. In the incremental formulation of variational assimilation, the gradient of the cost function is evaluated by means of the tangent and adjoint operators associated to the model, and the observation operator. In the next sections, we present the several ingredients to assimilate Lagrangian observations: the observation operator, that needs a particular interpolation scheme, as well as the construction of its tangent and adjoint models.

### 2.2 Construction of the observation operator $\mathcal{H}$

We assume that the floats are drifting at a fixed depth  $z_0$  in the fluid domain  $\mathcal{D} \subset \mathbb{R}^3$ . A float position  $\mathcal{X}(t) \in \mathbb{R}^2$  at time  $t$  is submitted to the fluid velocity  $U$  in the following way

$$\begin{cases} \frac{d\mathcal{X}}{dt} = U(t, \mathcal{X}(t), z_0) \\ \mathcal{X}(0) = \mathcal{X}_0, \end{cases} \quad (1)$$

where  $\mathcal{X}_0$  the initial position of the float.

The continuous variable  $\mathcal{X}(t)$  is discretized in time with the same time step  $\Delta t$  as the direct model NEMO. Let  $[0, T]$  be the time interval, divided into  $N$  time steps, then we define the successive positions of the float with the vector  $(\mathcal{X}_0, \dots, \mathcal{X}_N) \in \mathbb{R}^{2(N+1)}$ . Equation (1) is approximated by a leap-frog scheme. At each time step  $t_k = k\Delta t$ , one needs to evaluate the velocity field at the position  $\mathcal{X}_k$ , which is done by an interpolation of the velocity fluid  $U_k$  discretized at  $t_k$ . This velocity, called  $\mathcal{U}_k$ , defines the float velocity at time  $t_k$ . Let  $\mathbf{I}(f, \mathcal{X})$  be the operator that interpolates a function  $f$  defined in an appropriate space at a point  $(\mathcal{X}, z_0) \in \mathcal{D}$ . This operator will be detailed in the next subsection 2.3. The discrete time scheme then writes:

- Initialization

$$\begin{cases} \mathcal{X}_0 & \text{given,} \\ \mathcal{U}_0 = \mathbf{I}(U_0, \mathcal{X}_0). \end{cases}$$

- First time step

$$\begin{cases} \mathcal{X}_1 = \mathcal{X}_0 + \Delta t \mathcal{U}_0, \\ \mathcal{U}_1 = \mathbf{I}(U_1, \mathcal{X}_1). \end{cases}$$

- From  $t_k$ ,  $k = 2, \dots, N$ :

$$\begin{cases} \mathcal{X}_k = \mathcal{X}_{k-2} + 2\Delta t \mathcal{U}_{k-1}, \\ \mathcal{U}_k = \mathbf{I}(U_k, \mathcal{X}_k). \end{cases}$$

This scheme is very simple, and thus has the advantage to be differentiable (up to the differentiation of  $\mathbf{I}$  w.r.t.  $\mathcal{X}_k$ ). The tangent code associated to the  $k$ -th iteration writes:

$$\begin{cases} \delta \mathcal{X}_k = \delta \mathcal{X}_{k-2} + 2\Delta t \delta \mathcal{U}_{k-1}, \\ \delta \mathcal{U}_k = \mathbf{I}(\delta U_k, \mathcal{X}_k) + \delta \mathcal{X}_k \cdot \frac{\partial \mathbf{I}}{\partial \mathcal{X}_k}(U_k, \mathcal{X}_k). \end{cases} \quad (2)$$

Computation of  $\delta \mathcal{U}_k$  in (2) raises the problem of the differentiation of  $\mathbf{I}$  with respect to  $\mathcal{X}_k$ . This is discussed in the next section.

### 2.3 Construction of an interpolation operator adapted to differentiation

The operator  $\mathbf{I}$  is not linear in  $\mathcal{X}$  in general, implying complicated tangent and adjoint operators. If the grid is regular, a cheap solution is to define  $\mathbf{I}$  by mean of the eight corners  $Q_{ij}$  of the mesh that contains  $\mathcal{X}$ . We can express  $\mathbf{I}(f, \mathcal{X})$  by:

$$\mathbf{I}(f, \mathcal{X}) \approx \sum_{i,j=1}^2 a_{ij} f(Q_{ij}), \quad (3)$$

where  $a_{ij}$  are weighting coefficients. The tangent code of (3) writes:

$$\delta \mathbf{I}(f, \mathcal{X}) = \mathbf{I}(\delta f, \mathcal{X}) + \delta \mathcal{X} \cdot \partial_{\mathcal{X}} \mathbf{I}(f, \mathcal{X}). \quad (4)$$

$$= \sum_{i,j=1}^2 a_{ij} \delta f(Q_{ij}) + \sum_{i,j=1}^2 \delta a_{ij} f(Q_{ij}) \quad (5)$$

The first term  $\mathbf{I}(\delta f, \mathcal{X})$  of (4) is already implemented in NEMOVAR, in the `obs_inter_h2dtam` module. The second term, which comes from the Lagrangian formulation, requires the computation of the weight's partial derivatives with respect to  $\mathcal{X}$ . There are five methods to evaluate the weights  $a_{ij}$  in NEMO, but among these methods, only one is totally differentiable: the method proposed by Daget, and called de General bilinear remapping interpolation (see [the technical report CMGC 06 18](#)). We recall the main idea of this algorithm here, in order to write explicitly its tangent and adjoint models. The method proposed by Daget is based on a mapping from the four points  $\{Q_{ij}\}$  on the square cell  $\{0, 1\}^2$ . Computation of weights  $a_{ij}$  is equivalent to the computation of the local coordinates  $\alpha, \beta$  of  $\mathcal{X}$  inside the unit cell. The relation between  $a_{ij}$  and  $\alpha, \beta$  writes:

$$\begin{aligned} a_{11} &= (1 - \alpha)(1 - \beta) \\ a_{12} &= (1 - \alpha)\beta \\ a_{21} &= \alpha(1 - \beta) \\ a_{22} &= \alpha\beta. \end{aligned}$$

The associated operator  $A : \mathbb{R}^2 \rightarrow \mathbb{R}^4$  defines the transformation:

$$a = A(\alpha, \beta).$$

By then, the relation between float position  $\mathcal{X}$  and  $\{Q_{ij}\}$  writes:

$$\mathcal{X} = F(\alpha, \beta),$$

where  $F : \mathbb{R}^2 \rightarrow \mathbb{R}^2$  is a non linear function depending on  $\{Q_{ij}\}$ :

$$\begin{aligned} \mathcal{X}^1 &= (1 - \alpha)(1 - \beta)Q_{11}^1 + \alpha(1 - \beta)Q_{21}^1 + \alpha\beta Q_{22}^1 + (1 - \alpha)\beta Q_{12}^1, \\ \mathcal{X}^2 &= (1 - \alpha)(1 - \beta)Q_{11}^2 + \alpha(1 - \beta)Q_{21}^2 + \alpha\beta Q_{22}^2 + (1 - \alpha)\beta Q_{12}^2. \end{aligned}$$

The algorithm proposed by Daget consists in finding the couple  $\alpha = (\alpha_1, \alpha_2)$  as the solution of an iterative process given by:

Start with a given  $(\alpha_1, \alpha_1)$ . While  $\delta\alpha$  is great enough, inverse the linear system:

$$\delta\mathcal{X} = \tilde{F}_\alpha(\delta\alpha),$$

where  $\tilde{F}$  is the linearization of  $F$  around  $(\alpha)$ , and update the quantities:

$$\alpha \leftarrow \alpha + \delta\alpha.$$

## 2.4 Tangent and adjoint observation operators

The tangent model of  $\mathcal{H}$  writes at time  $k$ :

$$\begin{cases} \delta\mathcal{X}_k = \delta\mathcal{X}_{k-2} + 2\Delta t \delta\mathcal{U}_{k-1}, \\ \delta\mathcal{X}_k = \tilde{F}_{\alpha_k}(\delta\alpha_k), \\ \delta a_k = \tilde{A}_{\alpha_k}(\delta\alpha_k), \\ \delta\mathcal{U}_k = \sum_{i,j=1}^2 a_{ij} \delta U_k(Q_{ij}) + \sum_{i,j=1}^2 \delta a_{k,ij} U_k(Q_{ij}), \end{cases} \quad (6)$$

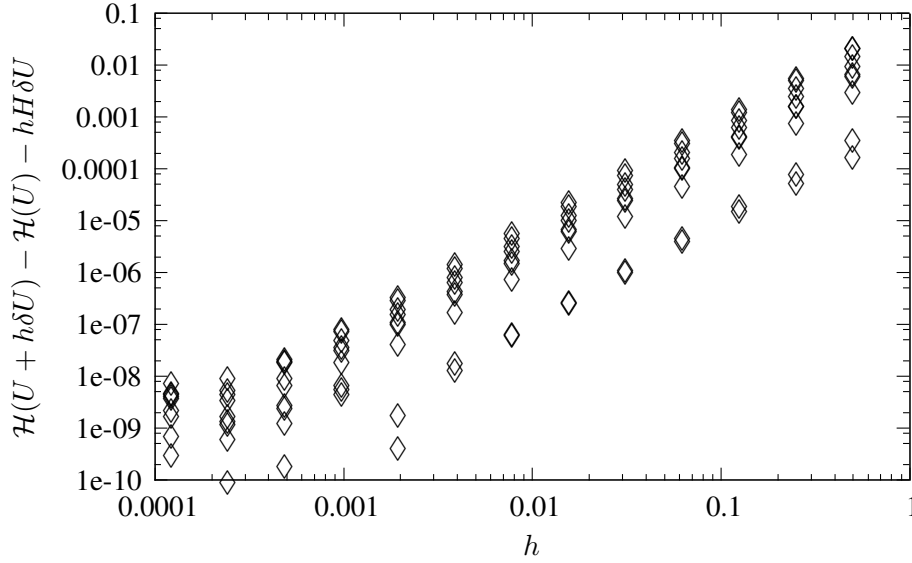


Figure 1: Validation of the tangent construction. For a given  $h$ , evaluation of  $\mathcal{H}(U + h\delta U) - \mathcal{H}(U) - hH\delta U$  for 18 float trajectories, after 6 days.

where  $\tilde{A}$  is the tangent model of operator  $A$ , and  $U_k, \delta U_k$  are the direct and tangent model of the velocity field. The adjoint model is easily deduced from the tangent one. Figure 1 illustrates the linear tangent hypothesis in the GYRE configuration, for 18 floats, after 6 days: for an increment of the velocity model  $h\delta U$ , with varying  $h$ , one observes a behaviour of the quantity  $\mathcal{H}(U + h\delta U) - \mathcal{H}(U) - hH\delta U$  in  $O(h^2)$ .

Listing 1: Adjoint test output for the Lagrangian Observation Operator - with ORCA2 - for 18 float trajectories - after 6 days.

Routine (L)	( L * dx ) <sup>T</sup> W dy	dx <sup>T</sup> L <sup>T</sup> W dy	Rel. Err.	Mach. Eps.	Status
-----	-----	-----	----	-----	-----
dia_obs_adj T1	No T profile observations				
dia_obs_adj T2	No S profile observations				
dia_obs_adj T3	No SLA observations				
dia_obs_adj T4	No SST observations				
dia_obs_adj T5	.680160300861511E+02	.680160300861511E+02	.4E-15	.2E-15	warning
dia_obs_adj T6	.449357099728177E+02	.449357099728177E+02	.9E-15	.2E-15	warning

For a perturbation  $\delta X$  on the state variable, it consists in the comparison of the two scalar products: The classical test to validate and adjoint  $H^T$  of a linear operator  $H$  is:

$$\langle H\delta X, Y \rangle \text{ and } \langle \delta X, H^T Y \rangle .$$

In NEMOVAR, the test is made with a particular choice of  $Y$ :  $Y = H\delta X$ , and we have chosen to implement the same test for the adjoint model related to (6). Listing 1 shows the validation of our adjoint model in the ORCA configuration. For line T1 – T5, the component of the state variable linked to the observed variable is perturbed; second and third columns show the scalar product values, forth and fifth columns are the relative error between these two scalar products, and the expected machine error. Line T5 stands for the Lagrangian Observation Operator only, for which an analytical velocity model is given, depending on space and time coordinates. In Line T6, all the components of the state variable are perturbed, and the sum of all the

scalar products with respect to each observation operator is done. In this case, the velocity field is constant, implying a different result from the Lagrangian tangent and adjoint model viewpoint: the two scalar products are different from line T5.

## 2.5 An adapted implementation of grid\_search for Lagrangian data.

The specific nature of Lagrangian particles is that they move according to the model velocity field. At each time step of the algorithm, one needs to find the cell  $\mathcal{C}_j$  that contains the particle. The actual *obs\_grid\_search* algorithm consists in looking among all the grid cells, and is aimed to initialize the observation positions in the geographical grid. Such an algorithm is not appropriate when the positions have to be evaluated at each time step.

A way to tackle this issue is to make a quick search for particle at time  $t_n$  knowing its grid cell at time  $t_{n-1}$ : indeed, the time step is small enough to guarantee that the particle either stays in the same cell, or goes to a neighbor cell. In this viewpoint, the Lagrangian nature of particles allows to construct an efficient grid search.

## 3 Reference Manual of the Lagrangian Observation Operator in NEMO / NEMOVAR

The code structure for Lagrangian Observation Operator follows the philosophy of the other observation operators. Specific directories have been created in NEMO and NEMOVAR, and the modules contained therein are detailed in the next subsection. Then, a specific attention is given to the Fortran structure type *obs\_lag* defining the Lagrangian observation object, and to the construction of the Lagrangian observation operator (module *obs\_lag\_adv*).

### 3.1 Lagrangian observation operator modules in NEMO

The following files are located in NEMO/OPA\_SRC/LAG directory:

```
obs_lag_adv.F90    ! Update the Lagrangian observation data structure at each time step
obs_lag_def.F90    ! Definition of the Lagrangian observation type
obs_lag_read.F90   ! Read the float coordinates
obs_lagrangian.F90 ! Storage space for Lagrangian observations
obs_lag_write.F90  ! Write observation and model equivalent to a file
```

File *obs\_lagrangian* is the equivalent of *obs\_profiles.F90*. File *obs\_lag\_def* contains also allocation and deallocation routines of *obs\_lag* data type objects.

As for the other observations, the Lagrangian data structure is loaded from the *dia\_obs\_init* routine of module *diaobs*. Three namelist parameters have been added to the namelist *namobs*:

*ln\_lag* : logical flag that enables Lagrangian observation operator

*lagfiles* : list of observation files

*ln\_ena\_lag* : Load the float observations from a feedback file, or a profile file.

For the moment, last flag *ln\_ena\_lag* has been implemented but not tested. Some problems are still present during the read of Netcdf files, written in Feedback format or not. The input file format is the



following: first line contains the number of floats, and then each line contains the following fields: float number, time step, position  $(\lambda, \phi, z)$  of the observation, and position of its model equivalent.

### 3.2 The Lagrangian data structure

The data structure `obs_lag` defined below contains the general information on the Lagrangian observation: the number of floats (`nlag`), the number of time steps of the direct model `nstp`, as well as other parameters such as the structure `isin` (tells if there is an observation of float  $j$  at time  $t_k$  for each float and each time step), counters of enabled / disabled floats (`nout`, `kout`), depth of each float (`mk`: constant and read from input file), beginning and end of observation for one float (fields `ista` and `iend`).

Object `obs_lag` contains fields called `var`: they correspond to the information needed for the zonal and meridional component: mesh grid index (`mi` and `mj`), observed position and its counterpart (`vobs` and `vmod`), as well as additional informations located in table `vext` (for instance float velocity at each time step...) and `rext` (for instance component of the matrix  $R$ ). The coordinates of a float at time  $t_k$  is contained in `(var(1)%vmod(kt),var(2)%vmod(kt))`.

Listing 2: `obs_lag` derived type

---

```

TYPE obs_lag
  INTEGER :: &
    & nlag, &           !: Number of floats
    & nstp, &          !: Time step of each trajectory
    & nstpup, &        !: Time step of each trajectory
    & next              !: number of extra variables

  LOGICAL, POINTER, DIMENSION(:, :) :: &
    & isin              !: true while the particle stays inside the domain

  INTEGER, POINTER, DIMENSION(: ) :: &
    & nout, &          !: Number of outgoing float
    & kout, &          !: Index when outgoing float
    & ista, &          !: Time index for the first observation time
    & iend, &          !: Time index for the last observation time
    & mk               !: k-th grid coordinate for interpolating X data

  TYPE(obs_lag_var), POINTER, DIMENSION(: ) :: &
    & var              !: coordinates in real space

  REAL(wp), POINTER, DIMENSION(: , : , :) :: & !:
    & lmask            !: land/ocean mask at T-, U-, V- and F-points

END TYPE obs_lag

```

---

Listing 3: `obs_lag_var` derived type

---

```

TYPE obs_lag_var

  INTEGER, POINTER, DIMENSION(: , :) :: &
    & mi, &            !: i-th grid coordinate for interpolating X data
    & mj, &            !: j-th grid coordinate for interpolation X data

  REAL(KIND=wp), POINTER, DIMENSION(: , :) :: & !: coordinates in real space

```

---

```

& vobs, &
& vmod, &
& rext          !: extra variable

REAL(KIND=wp), POINTER, DIMENSION(:, :, :) :: &
& vext          !: extra variable

END TYPE obs_lag_var

```

In NEMO, only the float velocities are used and updated during the observation diagnostics. They correspond to the component `var(i)%vext(:, 1, :)` of object `obs_lag_var`. In NEMOVAR, other quantities as to be saved during the computation, namely the tangent variables, the weighted misfit, etc. The definition of these different components is done in module `lag_obspoint` of `NEMOASSIM/VAR_SRC/LAGTAM`, keeping the code structure of other observation operators.

Listing 4: Lagrangian observation related global variables

```

INTEGER, PARAMETER :: mlagcoast = 11 ! Distance to coast
!!! mlagvelo must be equal to 1 (used in nemo) !!!
INTEGER, PARAMETER :: mlagvelo = 1 ! Velocity associated to component var
INTEGER, PARAMETER :: mlagvtan = 2 ! Tangent linear particle velocity
INTEGER, PARAMETER :: mlagberr = 3 ! Background error
INTEGER, PARAMETER :: mlagctan = 4 ! Tangent linear coordinate
INTEGER, PARAMETER :: mlagcwdt = 5 ! Weighted misfit
INTEGER, PARAMETER :: mlagvwdt = 6 ! Weighted misfit
INTEGER, PARAMETER :: mlaginn = 8 ! Innovation

!!! mlagdep must be equal to 1 (used in nemo) !!!
INTEGER, PARAMETER :: mlagdep = 1 ! Depth
INTEGER, PARAMETER :: mlagoerr = 2 ! Observation error

```

### 3.3 The Lagrangian observation operator

The direct lagrangian model is contained in file `obs_lag_adv`, which is composed of three routines:

Listing 5: `obs_lag_dev` module's header

```

MODULE obs_lag_adv

PUBLIC &
& obs_lag_adv_stp, &
!!-----
!!
!! ** Purpose : Advance Float positions at each time step: compute the
!!               model counterpart of float positions
!!
!! ** Method  : Update float velocity with the interpolation of U, V
!!               at each float position and then
!!               Explicit Euler scheme for updating positions.
!!-----

& obs_lag_intp, & !:
!!-----
!!

```

```

!! ** Purpose : Interpolate the model velocity at each float position
!!               located inside the domain
!!
!! ** Method  : Compute float velocity by an interpolation of U, V
!!               at each float position.
!!               Interpolation scheme k2dint = 3 available only
!!               (single interpolation scheme that is differentiable
!!               w.r.t. velocity model)
!!-----
& obs_lag_dom_chk  !:
!!-----
!!
!! ** Purpose : Checks that the updated lagrangian positions belongs
!!               to the domain
!!
!! ** Method  : Loop on the floats positions, and check that they
!!               belong to the U grid and the V grid.
!!
!! ** Action   : Compute the logical field obs_lag%isin
!!               as well as cell indices (mi,mj) in grids U and V.
!!-----

```

For a given time step, routine `obs_lag_adv_stp` computes the new float positions following a leap-frog scheme. In `obs_lag_intp`, float velocities are updated by mean of an interpolation of the model velocity at float positions. Finally, `obs_lag_dom_chk` checks and counts the floats that are still inside the domain, and computes the indices of the mesh grid that contains each float. During the initialization of the Lagrangian data structure (in routine `obs_lag_init`, of module `obs_lag_read`), the fields `mi` and `mj` are initialized for each time step and each float: this is essential since in NEMOVAR the adjoint model is applied from `nitend` to `nit000` with no direct model computation: the grid coordinates have to be computed once from the model equivalent of the float positions.

### 3.4 Tangent and adjoint models of the Lagrangian Observation Operator

Tangent and adjoint models for the assimilation of float position are implemented in the directory `LAGTAM` of `NEMOASSIM/VAR_SRC`. This directory contains the following modules:

```

lag_obs_inter_tam.F90 ! TL and AD model counterpart of lagrangian obs., and TL test
lag_obs_oper_tam.F90  ! Tangent and adjoint routines of the Lagrangian obs. operator
lag_obspoint.F90      ! Pointers for variable numbers in obs_lag structure
lag_obsstore.F90      ! Storage space for Lagrangian observations

```

Module `lag_obs_inter_tam` contains a set of tangent and adjoint routines associated to the bilinear remapping interpolation scheme presented in section 2.3. The other routines of this module are called by the two routines presented. By then they are not mentionned for the sake of simplicity. It corresponds respectively to the third and second lines of System (6).

Listing 6: `lag_obs_inter_tam` module's header

```

MODULE lag_obs_inter_tam

```

```

PUBLIC &
& obs_int_h2d_init_tan,      & ! TL Hor. interpolation to the observation point
!!-----

```

```

!! ** Purpose : Computes TL weights for horizontal interpolation to the
!!               observation point, for lagrangian data.
!!-----

& obs_int_h2d_tan ,          & !
!!-----
!! ** Purpose : Impact of TL weights
!!-----

& obs_int_h2d_init_adj,      &
!!-----
!! ** Purpose : Propagation of AD weights for lagrangian data.
!!-----

& lag_obs_int_h2d_adj ,      & !
!!-----
!! ** Purpose : Impact of the model on AD weights
!!-----

& obs_int_h2d_adj_lag ,      & !
!!-----
!! ** Purpose : Impact of AD weights on the state variable
!!-----

```

Module `lag_obs_oper_tam` is composed by the tangent and adjoint lagrangian observation operators, and calls the previous routines.

Listing 7: `lag_obs_oper_tam` module's header

```

MODULE lag_obs_oper_tam

PUBLIC obs_lag_opt_tan,      &
!!-----
!! ** Purpose : TL position and velocity for lagrangian data
!!-----

& obs_lag_opt_adj,          & ! Compute the AD model counterpart of lagrangian obs.
!!-----
!! ** Purpose : Adjoint model for Lagrangian Observation Operator
!!-----

& obs_lag_opt_tan_tst       ! TL test
!!-----
!! ** Purpose : Test the tangent routine for lagrangian data.
!!-----

```

The Lagrangian Observation Operator is the only observation operator that is non linear with respect to the state variable, since the float position depends on the model velocity. The tangent operator is a linear approximation of the direct operator, and is validated in the routine that tests the adjoint model of observation operators `dia_obs_adj_tst`. In this routine, a fifth part has been added to test the Adjoint model of Lagrangian observation operator. It has to be underlined that here, we have implemented the test for an analytical model velocity, in order to validate the code not only for a single time step, but also for a full float trajectory.

### 3.5 Lagrangian Observation Error

```
taille des structures nlag * nstp
nstp =nitend-nit000+2
```

dans diaass

```
= calcul nombre d'obs ilagt (autres obs version mpp)
en fonction kindic (incremental ou non: calcul delta x
appel a la fonction cout obs\_cost
```

```
obscost: calcul wdt = tan - inn
appel a oer\_rpy\_dset\_lag pour le vecteur de toutes les obs a tous les temps
```

nemovar\_inner load innovation in array to apply sim\_var before each of the three

obshand define size of observation vector initialise mlagctan

obs\\_tan init ctan

simvar loading tangent and cwdt and inn vectors

obsinc maj inno where is\\_obs and isin

diaobs\_tan

Initialisation of Lagrangian observation is done in `dia_obs_init`, with the call of `obs_lag_init`. The read of feedback format for lagrangian observation is written in routine `obs_rea_lag_dri`, but still not tested. Initialisation of TL observation operator (routine `obs_tan_init` of module `obs_tan`) appears in a lot of routines, and thus is suppressed in this diagram (it appears in `simvar`, `obshand`, `obscost`, `diaass`, as well as in the test of Lagrangian observation operator. Initialisation of the innovation vector, made in routine `obs_inc_init` of module `obs_inc.F90`, is called in `obshand` and `vartst`. The initialisation of the Lagrangian adjoint variable is done in `diaobs_tan_init`.

## 4 Demonstrator for the twin experiments

The information of this part are available on the VODA project website <http://voda.gforge.inria.fr>.

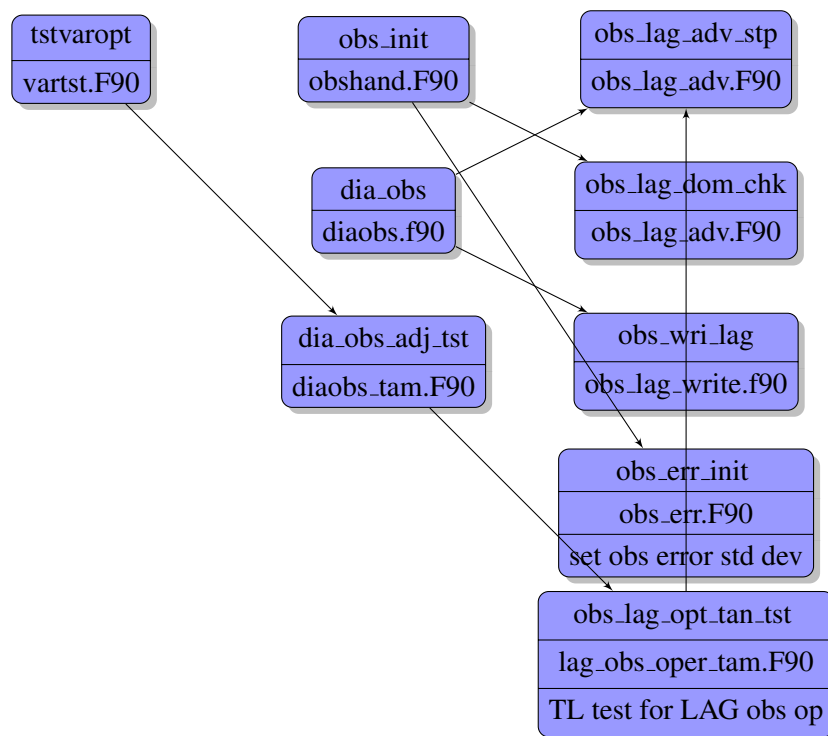


Figure 2: From the main assimilation routine, diagram of the calls ('namtst' for adjoint, gradient, tangent tests, 'namobs' for single obs.experiemment and 'namhlt' for linear tangent hypothesis)

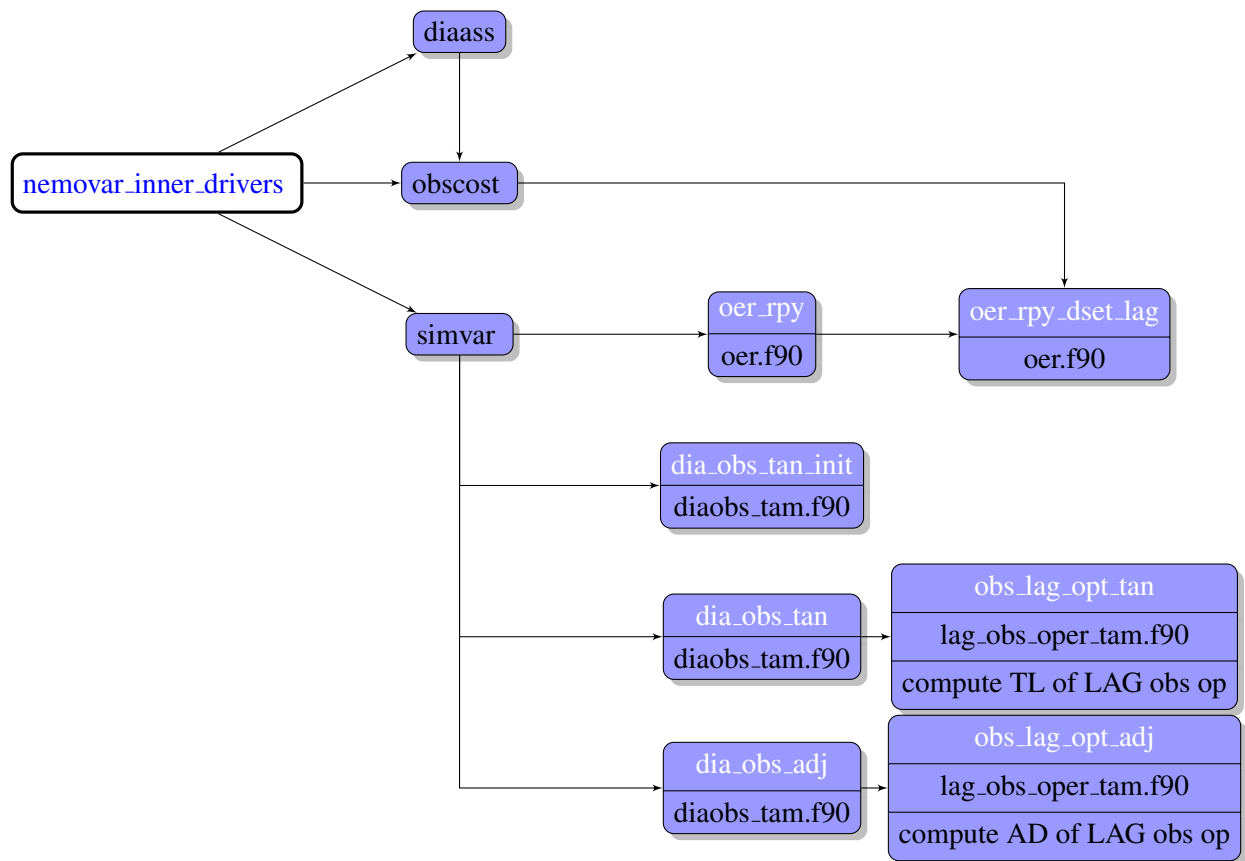


Figure 3: From the main assimilation routine, diagram of the calls ('namtst' for adjoint, gradient, tangent tests, 'namobs' for single obs.experiemment and 'namhlt' for linear tangent hypothesis)

## A APPENDIX : Subtasks, milestones and deliverables for Task 2.3

### Subtask 2.3.1: Identical twin experiments.

Identical twin experiments will be performed in the low-resolution model, in order to validate the assimilation of Lagrangian data. The main work of this subtask will be the correct formulation of the Lagrangian observation operator.

### Subtask 2.3.2: Non-identical twin experiments: the representativeness error issue.

Then, the problem of the representativeness error must be addressed. Indeed, in the real ocean, Lagrangian drifting floats are subject to the full range of scales of the fluid motion, while the resolution of the ocean model filters the smaller scales. We will generate high-resolution observations using the high-resolution component of WP4 T4.2. Developments of subtask 2.3.1 will be then used to assimilate these pseudo data into the low-resolution data assimilation system. The problem we are likely to face is the modelling of the Lagrangian trajectories into the low-resolution model. This issue could be addressed by adding new terms in the floats advection equation, e.g., stochastic noise, smaller scale parameterization. Should this approach fail, we would try instead to filter the high-resolution data before assimilation, and propose a satisfying pre-processing of the Lagrangian data. Another point of this subtask is to assess the complementarities between the Lagrangian position data, and two other types of data: temperature and salinity profiles, altimetry. At this point we will be able to optimize the observing system design in the given domain.

### Subtask 2.3.3: Real data assimilation.

This will be done in the framework of WP4 T4.3, using developments and expertise from the previous subtasks. The issue specifying the measurement error will have to be specifically addressed. For Argo floats, errors come mainly from the wind-induced drift of the float at the surface of the ocean. Indeed, the floats are located by GPS, but there are some gaps in time between the arrival of the float at the surface and the moment it is first seen by the satellite. These drifts depend on the surface wind and surface currents, so that they are inhomogeneous in time and in space.

### Milestone:

M 2.3 implementation of the observation operator for the general case (subT2.3.2, T0+18).

### Deliverables:

- D 2.3.1: Documentation of the observation operator. Demonstrator for the twin experiments. (T0+12)
- D 2.3.2: Documentation of the corrected observation operator and/or the adequate pre-processing of the Lagrangian positions. Demonstrator for the twin experiments. (T0+24)
- D 2.3.3: Documentation of the modelling of real errors and the pre-processing of real data. Demonstrator for the real data assimilation experiments. (T0+36) Demonstrators will consist of a web page accessible from the project web site. It will describe the method and show some results.



### **APPENDIX : Acronyms**

- NEMO : Nucleus for European Modelling of the Ocean.
- NEMOTAM : TAM pour NEMO
- NEMOVAR : Variational Data Assimilation System for NEMO
- OPA : Ocean Parallélisé. Direct ocean model, component of NEMO.
- OPATAM : TAM for OPA 8.x
- OPAVAR : Variational Data Assimilation System for OPA 8.x
- ORCA : Global grid for OPA
- TAM : Tangent and Adjoint Model