Simplification operator and its inverse for multi-incremental assimilation

Arthur Vidard

Delivrable D3.1.1:

Simplification operator and its inverse for multi-incremental assimilation

Arthur Vidard
INRIA/LJK, Grenoble

VODA: ANR-08-COSI-016

Contents

1 Introduction

The variational assimilation problem solved by NEMOVAR is defined very generally as the minimization of a cost function of the form

$$J[\mathbf{v}] \;=\; \frac{1}{2}\,[\mathbf{v}-\mathbf{v}^b]^T\,[\mathbf{v}-\mathbf{v}^b] \;+\; \frac{1}{2}\,[\,G\,(U(\mathbf{v}))-\mathbf{y}^o\,]^T\,\mathbf{R}^{-1}\,[\,G\,(U(\mathbf{v}))-\mathbf{y}^o\,] \tag{1}$$

where $\mathbf{v}$ is the control (analysis) vector, $\mathbf{v}^b$ is the background estimate of the control vector, $\mathbf{y}^o$ is the vector of observations, $\mathbf{R}$ is an estimate of the observation error covariance matrix, U is a (possibly) nonlinear operator that maps the control vector onto the model state (initial condition) space ($\mathbf{x} = U(\mathbf{v})$)[1] , and G is a nonlinear operator that maps from model state space onto the space of the observation vector (this includes the integration of the model from initial time to the observation times, as well as the interpolation onto the observation points). The background-error covariance matrix of the control vector is assumed to be the identity matrix ($\mathbf{B}_{(\mathbf{v})} = \mathbf{I}$) as evident by the use of the canonical inner product for the background term in (1). In other words, background errors for $\mathbf{v}^b$ are assumed to be uncorrelated and to have unit variance. There are two advantages that result from this formulation where the background term takes on a very simple form. First, it generally improves the convergence properties of the minimization when the problem is solved with a conjugate gradient algorithm. For quadratic cost functions, this is often explained by a reduction in the condition number of the Hessian (Golub & Van Loan 1996). Second, all constraints in the assimilation problem are now imposed through the nonlinear observation operators G and U, including multivariate and smoothness constraints that are used in conventional model-space (matrix) formulations of the background-error covariance model. In particular, this opens the way for incorporating potentially more realistic (nonlinear) multivariate balance relationships in the analysis problem. Details on techniques for constructing the transformation U (and its inverse U^{-1}) can be found in Weaver et al. (2005).

Currently NEMOVAR employs a variant of the incremental algorithm (Courtier et al. 1994) for approximately minimizing the nonquadratic cost function (1). The algorithm is defined by the iterative minimization of a sequence, $k = 1, ..., K_o$, of quadratic cost functions

$$\begin{aligned} J^k[\delta\mathbf{v}^k] \;&=\; \frac{1}{2}\,\left[\delta\mathbf{v}^k-\mathbf{d}^{b,k-1}\right]^T\,\left[\delta\mathbf{v}^k-\mathbf{d}^{b,k-1}\right] \\ &+\; \frac{1}{2}\,\left[\,\mathbf{G}^{k-1}\mathbf{U}^{k-1}\delta\mathbf{v}^k-\mathbf{d}^{o,k-1}\,\right]^T\,\mathbf{R}^{-1}\,\left[\,\mathbf{G}^{k-1}\mathbf{U}^{k-1}\delta\mathbf{v}^k-\mathbf{d}^{o,k-1}\,\right] \end{aligned} \tag{2}$$

where

$$\mathbf{d}^{b,k-1} \;=\; \mathbf{v}^b-\mathbf{v}^{k-1}, \tag{3}$$

$$\mathbf{d}^{o,k-1} \;=\; \mathbf{y}^o-G(U(\mathbf{v}^{k-1})) \;=\; \mathbf{y}^o-G(\mathbf{x}^{k-1}), \tag{4}$$

[1]By interpreting $\mathbf{x}$ to be the initial conditions, the model and external forcing fields are tacitly assumed to be perfect. This assumption can be relaxed in the above formulation by considering $\mathbf{x}$ to contain model-error or external forcing terms in addition to the initial conditions.

$\mathbf{v}^{k-1}$ is a reference state, $\delta\mathbf{v}^k$ is an increment defined by $\mathbf{v}^k = \mathbf{v}^{k-1} + \delta\mathbf{v}^k$, and $\mathbf{G}^{k-1}$ and $\mathbf{U}^{k-1}$ are linearized operators defined such that $G(U(\mathbf{v}^{k-1} + \delta\mathbf{v}^k)) \approx G(U(\mathbf{v}^{k-1})) + \mathbf{G}^{k-1}\mathbf{U}^{k-1}\delta\mathbf{v}^k$ (when this equation is satisfied exactly, (3) is identical to (1)). The superscript $k-1$ indicates that $\mathbf{G}^{k-1}$ is the result of linearizing G about $\mathbf{v}^{k-1}$. The sequence $k = 1, ..., K_o$ are called outer iterations while the minimization iterations performed within each outer loop are called inner iterations. Equations (3) and (4) are the effective "background" and "observation" vectors for the inner loop minimization. In practice, it is customary to set $\mathbf{v}^0 = \mathbf{v}^b$ and to choose $\mathbf{v}^{k-1}$, for $k = 2, ..., K_o$, to be the solution obtained at the end of the previous outer loop. The minimum of (3) after the K_o-th outer iteration defines the analysis increment, $\delta\mathbf{v}^a = \delta\mathbf{v}^{K_o}$.

In order to deal with the increase of computing cost and non linearities when going toward higher resolution applications, Numercal Weather Prediction center (ECMWF, Météo-France, UK-MetOffice, ...) usually use the so-called Multi-incremental approach: the models used in the successive minimizations of the inner loop are approximation of the tangent model (lower resolution and simplified physics). The first outer loop is performed using a very coarse grid for the inner loops models, and the resolution (and the physics) is improve for each subsequent outer loops. The hypothesis becomes: $G(U(\mathbf{v}^{k-1} + \mathbf{S}^{-I}\delta\mathbf{v}^k)) \approx G(U(\mathbf{v}^{k-1})) + \mathbf{S}^{-I}\mathbf{G}_L^{k-1}\mathbf{U}_L^{k-1}\delta\mathbf{v}^k$ (where the indice $_L$ means lower resolution and $\mathbf{S}$ is the simplification operator). The incremental algorithm used for minimizing (1) is summarized in Fig. 1.

This document describes the implementation of the Simplification Operator and its generalized inverse within the NEMOVAR framework. The next section will present the context and highlight the difficulties associated to such operators. section 3 will quickly describe the interpolation operator (*i.e.* the generalized inverse of the simplification operator) with illustration an a practical test case. Section 4 will describe more extensively the actual simplification operator and use the same test case as previously as illustration. Finally section 5 will present the technical aspects of the current implementation in NEMOVAR.

2 Formalisme

Multi-incremental versions of variational assimilation require an operator to transform ("simplify") the state vector from high resolution (HR) to low resolution (LR) and a generalized inverse of that operator to transform ("interpolate") the increment from low resolution to high resolution. In the algorithm proposed in Fig. 1, the simplification operator, $\mathbf{S}$, is needed to define the basic state at low resolution of the linearized operators. It is easier to define the interpolation operator first ($\mathbf{S}^{-I}$) and to derive the simplification operator from the solution that minimizes the objective function

$$F(\mathbf{x}_L) = \frac{1}{2}[\mathbf{S}^{-I}\mathbf{x}_L - \mathbf{x}]^T \ \mathbf{W} \ [\mathbf{S}^{-I}\mathbf{x}_L - \mathbf{x}] \tag{6}$$

where $\mathbf{x}$ is the (known) $N \times 1$ state vector at high resolution, $\mathbf{x}_L$ is the (unknown) $L \times 1$ state vector at low resolution ($L < N$), and $\mathbf{W}$ is a $N \times N$ symmetric, positive definite

INITIALIZATION

- Given the model (background) initial state : $\mathbf{x}^0 = \mathbf{x}^0(t_0) = \mathbf{x}^b$
  - Compute the model (background) state trajectory : $\mathbf{x}^0(t_i),\ 0 \leq t_i \leq T$
  - Store the trajectory at (possibly) lower spatial and temporal resolution : $\mathbf{x}_L^0(t_i) = \mathbf{S}\,\mathbf{x}^0(t_i)$
  - Compute the observation-minus-background misfit vector : $\mathbf{d}^{o,0} = \mathbf{y}^o - G[\mathbf{x}^0]$
  - Initialize the control misfit vector : $\mathbf{d}^{b,0} = \mathbf{v}^b - \mathbf{v}^0 = 0$
  - Evaluate the initial value of the unapproximated (*nonquadratic*) cost function (Eq. 1) :

$$J^0[\mathbf{v}^0] = \frac{1}{2}\left[\mathbf{d}^{b,0}\right]^T\left[\mathbf{d}^{b,0}\right] + \frac{1}{2}\left[\mathbf{d}^{o,0}\right]^T\mathbf{R}^{-1}\left[\mathbf{d}^{o,0}\right] = \frac{1}{2}\left[\mathbf{d}^{o,0}\right]^T\mathbf{R}^{-1}\left[\mathbf{d}^{o,0}\right]$$

BEGIN OUTER LOOP : $k = 1$ to K

- Given the background state $\mathbf{x}^b$, most recent estimate $\mathbf{x}^{k-1}$, and misfit vectors $\mathbf{d}^{b,k-1}$ and $\mathbf{d}^{o,k-1}$:
  - Retrieve $\mathbf{x}_L^{k-1}(t_i)$ to define the basic state of the linearized operators $\mathbf{G}^{k-1}$ and $\mathbf{U}^{k-1}$
  - Initialize the optimization first-guess of the model state increment : $\delta\mathbf{v}^{k,0} = 0$

  BEGIN INNER LOOP : $m = 1$ to M
  - Update the increment $\delta\mathbf{v}^{k,m}$ in the approximated (*quadratic*) cost function (Eq. 3) :

$$\begin{aligned} J^k[\delta\mathbf{v}^k] &= \frac{1}{2}\left[\delta\mathbf{v}^k - \mathbf{d}^{b,k-1}\right]^T\left[\delta\mathbf{v}^k - \mathbf{d}^{b,k-1}\right] \\ &+ \frac{1}{2}\left[\mathbf{G}^{k-1}\,\mathbf{U}^{k-1}\delta\mathbf{v}^k - \mathbf{d}^{o,k-1}\right]^T\mathbf{R}^{-1}\left[\mathbf{G}^{k-1}\,\mathbf{U}^{k-1}\delta\mathbf{v}^k - \mathbf{d}^{o,k-1}\right] \end{aligned} \tag{5}$$

  END INNER LOOP

- Given the optimized control increment : $\delta\mathbf{v}^k = \delta\mathbf{v}^{k,M}$
  - Transform $\delta\mathbf{v}^k$ from control space to state space : $\delta\mathbf{x}_L^k \approx \mathbf{U}^{k-1}\,\delta\mathbf{v}^k$
  - Interpolate $\delta\mathbf{x}_L^k$ to (possibly) higher resolution : $\delta\mathbf{x}^k \approx \mathbf{S}^{-I}\,\delta\mathbf{x}_L^k$
  - Update the model initial state : $\mathbf{x}^k \approx \mathbf{x}^{k-1} + \delta\mathbf{x}^k = \mathbf{x}^0 + \sum_{l=1}^k \delta\mathbf{x}^l$
  - Update the model state trajectory (direct initialization or IAU) : $\mathbf{x}^k(t_i),\ 0 \leq t_i \leq T$
  - Store the trajectory at (possibly) lower spatial and temporal resolution : $\mathbf{x}_L^k(t_i) = \mathbf{S}\,\mathbf{x}^k(t_i)]$
  - Update the observation-minus-model misfit vector : $\mathbf{d}^{o,k} = \mathbf{y}^o - G[\mathbf{x}^k]$
  - Update the background-minus-model control misfit vector : $\mathbf{d}^{b,k} = \mathbf{v}^b - \mathbf{v}^k = -\sum_{l=1}^k \delta\mathbf{v}^l$
  - Evaluate the value of the unapproximated (*nonquadratic*) cost function (Eq. 1) :

$$J^k[\mathbf{v}^k] = \frac{1}{2}\left[\mathbf{d}^{b,k}\right]^T\left[\mathbf{d}^{b,k}\right] + \frac{1}{2}\left[\mathbf{d}^{o,k}\right]^T\mathbf{R}^{-1}\left[\mathbf{d}^{o,k}\right]$$

END OUTER LOOP

Figure 1: Flow diagram of the multi-incremental variational assimilation algorithm proposed for NEMOVAR. The role of the Simplification and interpolation operators are highlighted in red.

weighting matrix. The minimizing solution of (6) is

$$\mathbf{x}_L \;=\; \left[(\mathbf{S}^{-I})^T\,\mathbf{W}\,\mathbf{S}^{-I}\right]^{-1}\,(\mathbf{S}^{-I})^T\,\mathbf{W}\,\mathbf{x} \;\equiv\; \mathbf{S}\,\mathbf{x}. \tag{7}$$

The operators $\mathbf{S}$ and $\mathbf{S}^{-I}$ satisfy the mathematical properties $\mathbf{S}\mathbf{S}^{-I} = \mathbf{I}$ and $\mathbf{S}^{-I}\mathbf{S} = \mathbf{P} = \mathbf{P}^2$; i.e., transforming from LR $\rightarrow$ HR $\rightarrow$ LR does not change the solution, whereas transforming from HR $\rightarrow$ LR $\rightarrow$ HR is a projection. The filtering properties of $\mathbf{S}$ are controlled by the weighting matrix $\mathbf{W}$. Equation (7) requires the inversion of a large matrix. This could be achieved, for example, by iteratively minimizing (6) using CONGRAD. Since the simplified state is required only for defining the basic state of linearized operators (see Fig. 1), it may be acceptable to replace (7) by an approximate solution,

$$\mathbf{x}_L \;\approx\; \mathbf{W}_L^{-1}\,(\mathbf{S}^{-I})^T\,\mathbf{W}\,\mathbf{x} \tag{8}$$

where $\mathbf{W}_L$ is a $L \times L$ symmetric, positive definite weighting matrix with simpler structure than the matrix $\left[(\mathbf{S}^{-I})^T\,\mathbf{W}\,\mathbf{S}^{-I}\right]^{-1}$ in (7). Equation (8) may be interpreted as the adjoint of the interpolation operator with respect to the inner products $\mathbf{x}^T\mathbf{W}\mathbf{x}$ on HR-space and $\mathbf{x}_L^T\mathbf{W}_L\mathbf{x}_L$ on LR-space. The choice of $\mathbf{W}_L$ is a key point and will be discussed later on.

3 Interpolation operator

The general interpolation operators (and their adjoints) used in the profile and altimeter observation operators have been exploited here in defining $\mathbf{S}^{-I}$ (and $(\mathbf{S}^{-I})^T$) considering that the location of each High-Resolution gridpoint as an observation location. The grid search and the horizontal and vertical interpolation routine are exactly the same, the interface that calls the interpolation routines has been adapted to nemo-gridded data. As for the observation operator, several interpolation scheme are available (see the observation operator documentation for more information). For the simplification operator, it is recommended to use the general bilinear remapping interpolation (Daget 2006).

The fields that requires to be interpolated from low resolution to high resolution are restricted to assimilation increments. The Interpolation operator behave reasonably well with no obvious problems along the coast and at singular points as shown in figure 2. A special treatment has been done for sea point at higher resolution that are in the interior of land areas in the low resolution, such as closed, sea, channels or fjord (coastal point are generally OK, see Fig 3). In these areas, there is no obvious way to estimate the increment. It has therefore been set to 0, which is fine since we are only treating increments. If one needs to use the interpolator for the full field, this particular problem will need to be addressed.

Regarding the CPU cost, the interpolation of an entire assimilation increment (four 3D fields and one 2D field) from Orca2° to Orca1° (resp Orca$\frac{1}{4}$°) takes 35s (resp 8mn) on a common workstation

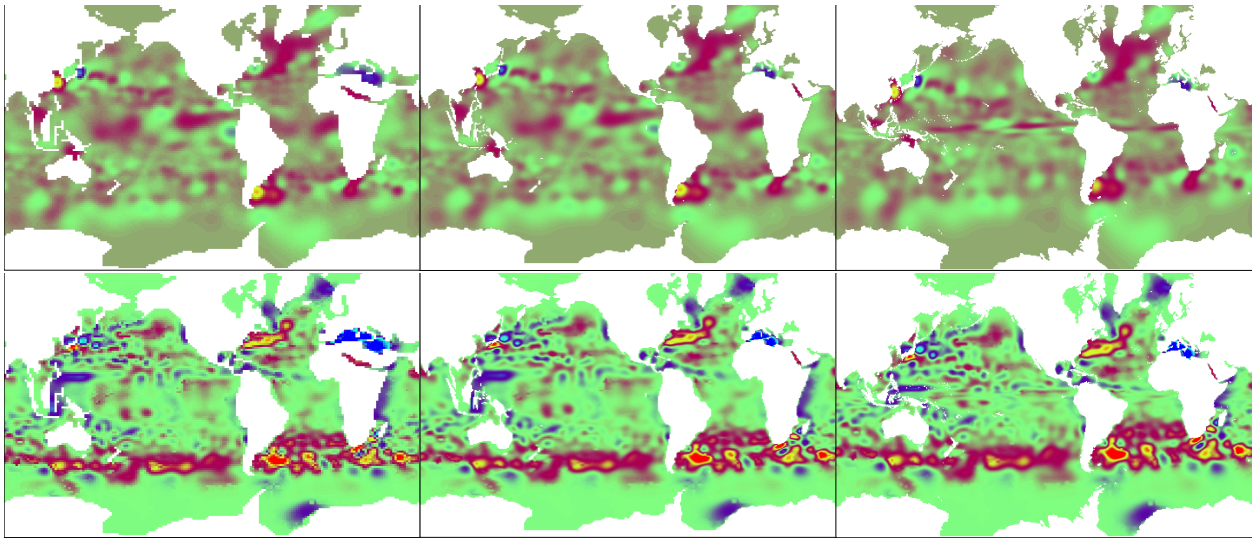

Figure 2: Interpolation of a temperature (top) and SSH (bottom) increments from Orca2° (left) to Orca1°(middle) and Orca$\frac{1}{4}^{\circ}$ (right)

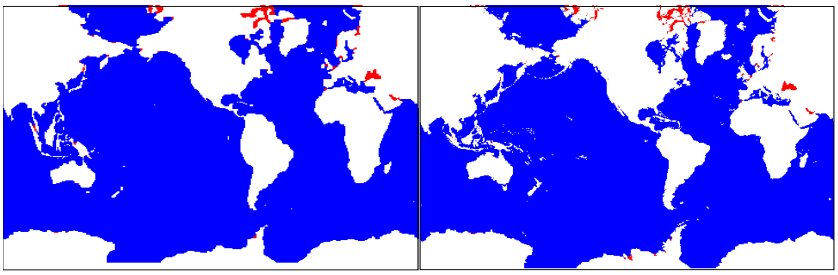

Figure 3: In red: High-resolution Sea Points where the interpolation failed to find a suitable value

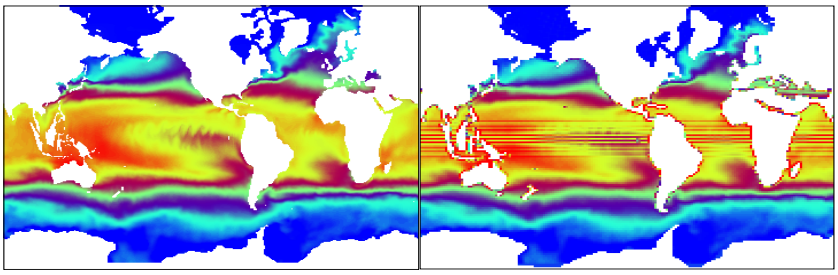

Figure 4: Original Orca1° background temperature field and the simplified Orca2° field obtained with the approximate simplification operator and using the LR volume elements as weighting matrix

4 Simplification operator

4.1 Approximate Simplification operator - the choice of $\mathbf{W}_L$

A simple and natural choice for $\mathbf{W}$ is the diagonal matrix of volume elements (scale factors) that define the HR model grid. Therefore, the first idea for $\mathbf{W}_L$, as presented in the VODA proposal was to approximate it by a diagonal matrix of volume elements for the LR model grid. However this leads to an unsatisfactory result (to say the least) as shown in Fig. 4.

However surprising at first, this bad behavior has a clear explanation. The bad results are located mainly along the coasts and in the equatorial belt. These are actually coming from 2 separate problems:

- Along the coasts and around the islands: This kind of weighting does not account for the land point of the HR grid that are sea point of the LR-grid. Indeed the weight is the same whatever the number of sea-point present in the HR cells used to compute the value of the corresponding LR cell

- Equatorial belt: this is a bit more complicate and is case dependent. This is actually due to the increase of meridional resolution in both ORCA1° and ORCA2°. Looking, for instance, at what happens along the 180°th meridian (see Fig. 5) In both extra equatorial cases, everything is fine, with the adjoint of the interpolation operator, the contributions to the lower resolution grid come, in the south, from the two HR points before and the two after and, in the north, from the one before, the one after and the one right spot on. The weighting by the volume elements do the trick. In the third case (Fig. 5 bottom), and that is what is happening in the equatorial belt, some problems arise. For instance, the LR point at the equator (0,180) gets contributions from the HR point at the equator (with weight 1) and the two surrounding HR gridpoints (both with weight 1/3) (total weight = 5/3) ; while the point at (0.5,180) gets contributions from the two surrounding HR gridpoint only, both with weight 2/3 (total weight = 4/3). and there is a cycling of this 5/3 - 4/3. The volume elements

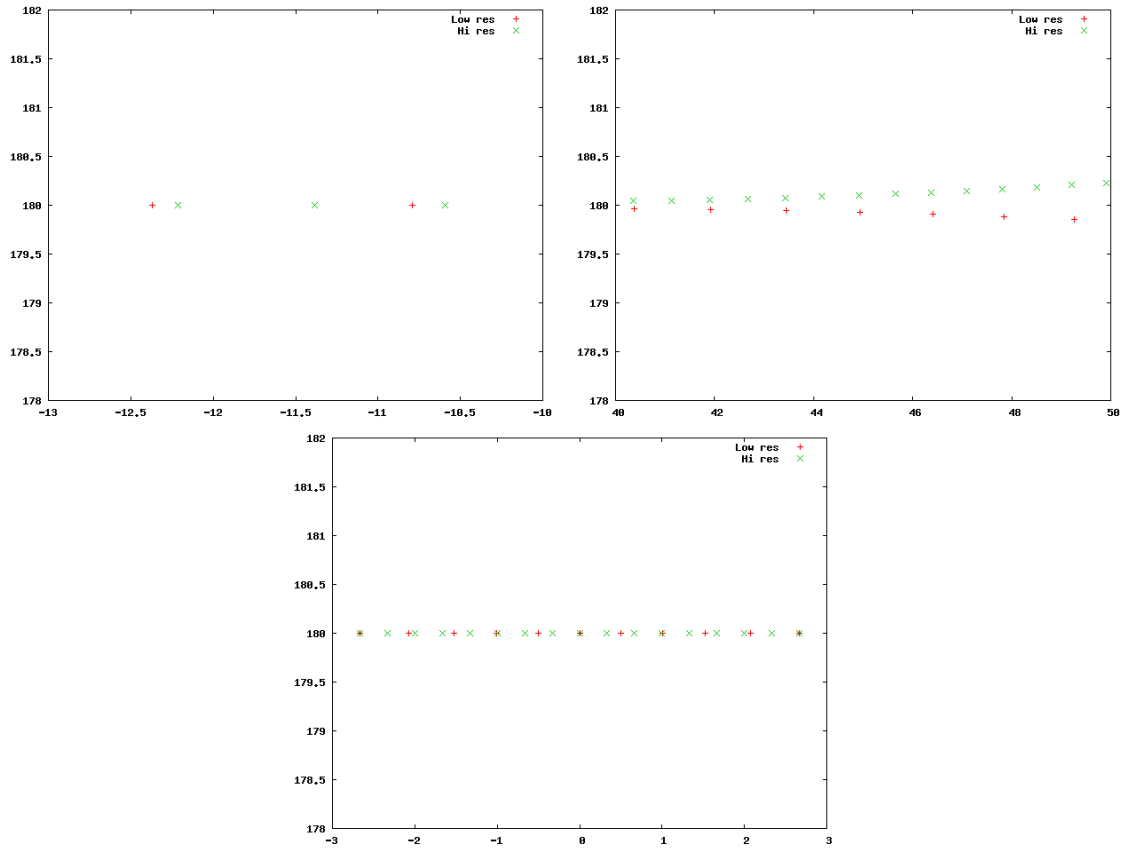

Figure 5: Grid points locations along the 180°th meridian for some part of south (top left) and north hemisphere and for the equatorial belt (bottom). Red pluses are the locations of the ORCA2° T-grid points and the green crosses are the ORCA1° T-grid points.

being roughly constant for each grid in that area, its use as a weight cannot correct this problem and we get the oscillation seen on Fig. 4.

If we assume that the weighting matrix $\mathbf{W}_L$ is diagonal, which is a reasonable assumption, the computation of $\mathbf{W}_L$ is actually straightforward. Indeed, remind that we want to find a weighting matrix such as $\mathbf{S}\mathbf{S}^{-I} = \mathbf{I}$, so combining this with equations (7) and (8) we get

$$\begin{aligned} \mathbf{x} &\approx \mathbf{W}_L^{-1} \left(\mathbf{S}^{-I}\right)^T \mathbf{W}\, \mathbf{S}^{-I}\mathbf{x} \\ \mathbf{W}_L\mathbf{x} &\approx \left(\mathbf{S}^{-I}\right)^T \mathbf{W}\, \mathbf{S}^{-I}\mathbf{x} \end{aligned} \tag{9}$$

For all $\mathbf{x}$ from the high resolution grid. This is in particular true for $\mathbf{x} = 1_{HR}$ the vector with 1 everywhere. Thus we get, since we assumed $\mathbf{W}_L$ diagonal,

$$\mathbf{W}_L \approx \left(\mathbf{S}^{-I}\right)^T \mathbf{W}\, \mathbf{S}^{-I} 1_{HR} \tag{10}$$

that can be computed thanks to the interpolation operator and its adjoint.

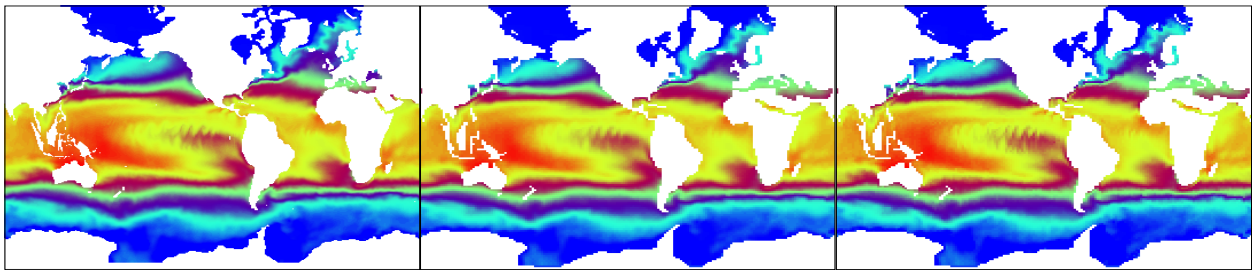

Figure 6: Original Orca1° background temperature field (left) and the simplified Orca2° field obtained with the Approximate Simplification Operator (middle) and the Full Simplification Operator (right)

4.2 Full Simplification operator (FSO) versus Approximate Simplification Operator (ASO)

Alternatively the FSO can be applied through the minimization of the function described in equation 6. This leads to a different result with slightly sharper details as shown in Fig. 6. The fact that the ASO tends to smooth out details may not be detrimental, indeed small scales features may not be well represented by the coarse grid model. For that reason, it is not obvious to assess whether it is better to use the output from the ASO or from the FSO as background state of the inner loop, and a more in depth study would be required.

Obviously, the result of the FSO is closer to the actual $\mathbf{S}$ than the one from ASO, but it comes at a cost. Figure 7 shows the norm $\| \mathbf{S}\mathbf{S}^{-I} - I \|^2$ (that should be equal to zero for a perfect $\mathbf{S}$) along with the CPU cost on a standard workstation respect to the number of minimization iteration for the FSO using the result of the ASO as a first guess (therefore 0 iterations means ASO). A good compromise may be to do only a few iterations of FSO (less than 5).

5 Technical specifications and User's Guide

5.1 Internals

The simplification operator makes use of the NEMOVAR observation operator[2] in order to perform $\mathbf{S}^{-I}$ and $(\mathbf{S}^{-I})^T$. It is linked to the NEMO library which has been compiled for the Low-Resolution grid. The High resolution grid points are considered as independent observation and the observation operator is used to compute their low resolution grid counterpart (this is equivalent to applying $\mathbf{S}^{-I}$). All the information about the two grids are loaded in the derived type described in listing 1. Apart from `nemo_grid%vmod` that is loaded from an auxiliary input file, the grid information such as coordinate and mask come from the nemo global variable (for the low resolution grid) or for a meshmask type file (for the High resolution grid) that is therefore required. The other informations

[2]See the relevant documentation for a detailed description

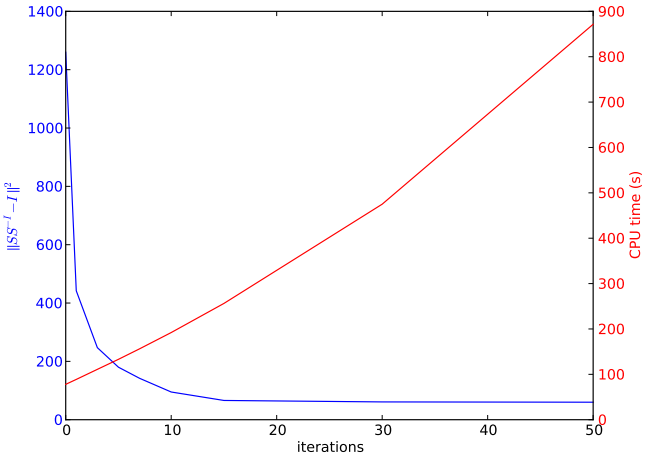

Figure 7: Error on the simplification operator (blue) and CPU time (red) regarding the number of FSO iterations (0=ASO)

included in this derived type are computed during the grid search of the observation operator.

Listing 1: The nemo-grid derived type

```
TYPE nemo_grid
   INTEGER ::  i, j, k !: size of the grid
   INTEGER      , ALLOCATABLE, DIMENSION(:,:)    :: &
      & npvsta, &       !: Start of each variable profile in full arrays
      & npvend, &       !: End of each variable profile in full arrays
      & mi,     &       !: i-th grid coord. for interpolating to profile T data
      & mj,     &       !: j-th grid coord. for interpolating to profile T data
      & i_proc          !: processor number whre the profile belong
   REAL(KIND=wp), ALLOCATABLE, DIMENSION(:,:)    :: &
      & rphi,   &       !: coordinate of the grid points
      & rlam            !:
   REAL(KIND=wp), ALLOCATABLE, DIMENSION(:,:,:) :: &
      & rdepth          !: depth of the grid levels (partial steps)
   REAL(KIND=wp), ALLOCATABLE, DIMENSION(:)      :: &
      & rdepth_0        !: depth of the grid levels
   INTEGER      , ALLOCATABLE, DIMENSION(:)      :: &
      & mvk             !: k-th grid coord. for interpolating to profile data
   REAL(KIND=wp), ALLOCATABLE, DIMENSION(:,:,:) :: &
      & vmod,   &       !: Field values
      & rmsk,   &       !: mask
      & wght            !: Wheight to the data
   LOGICAL ::                                        &
      & lalloc = .FALSE.

END TYPE nemo_grid
```

5.2 Compiling and running

The simplification operator makes use of the NEMOVAR compiling environment, a simple call to fcmvmake.ksh with the usual arguments and the `nemosim` keyword should do the trick.

```
> fcmvmake.ksh -c $COMPILE -t $WORKDIR -B nemosim
```

However it is not yet integrated into VODA's PIANO (Chauvin 2010) running environment yet and has to be run manually. Doing so requires to add the namelist block descried in listing 2 in the standard NEMO namelist. In addition to this namelist, and the usual input files describing the nemo configuration (*e.g.* `coordinate.nc`, `bathy_meter.nc`, ...) one needs to provide the meshmask for the high resolution grid (called `meshmask_HR.nc` in our example) and the input file corresponding to `ctype` (`assim_background_state_Jb.nc`, `tam_trajectory.nc` or `restart.anainc.nc` for `bck`, `trj` and `inc` respectively)

Listing 2: Namelist bloc

```
!-----------------------------------------------------------------------
!       namsim       simplification operator parameters
!-----------------------------------------------------------------------
! nsimex      type of operation
!                = 1 interpolator
!                = 2 approx. simplificator
!                = 3 full simplificator
!                = 4 test the approx simplificator
!                = 5 test the full   simplificator
! nitmax       maximum number of iterations for the simplificator
! lntest       flag to test the adjoint of the simpl.operator
! lnextrap     flag for performing extrapolation to one land point
!               prior to simplification
! n1dsin       Type of vertical interpolation method
!                0 = Linear intepolation.
!                1 = Cubic spline interpolation.
! n2dsin       Type of horizontal interpolation method
!               0 = Distance-weighted interpolation
!               1 = Distance-weighted interpolation (small angle)
!               2 = Bilinear interpolation (geographical grid)
!               3 = Bilinear remapping interpolation (general grid)
!               4 = Polynomial interpolation
! cfiout       name of the output file
! cmeshfile    name of the meshmask file (for the higher resolution)
! ctype        type of input file ('bck', 'trj' or 'inc')
!
&namsim
   nsimex    = 3
   nitmax    = 3
   cmeshfile = 'meshmask_HR.nc'
   cfiout    = 'result.nc'
   ctype     = 'bck'
   lntest    = false
   lnextrap  = true
```

```
   n1dsin     = 0
   n2dsin     = 2
/
```

5.3 adding a type of file to be treated

By default the simplification operator is only able to treat three kinds of file: increments, background and trajectory. The choice between these three being driven by the namelist parameter `ctype` ('inc', 'bck', 'trj' respectively). However it is relatively simple to add another case. All you need is to initialize a `field_list` type (see Listing 3) by adding a **`CASE`** in the subroutine `fill_list` of module `sim_opt_def`, following the example given in Listing 4. Note that the simplification or the interpolation are done one field at a time (*i.e.* only one is loaded in memory at once), and each time one variable is not on the same grid as the previous one, it has to redo the grid search. It is therefore better, when filling in the `filed_list`, to group the variables that are on the same grid.

Listing 3: Field and field-list derived type

```
   TYPE field                        !: Field description
      CHARACTER(LEN=8) :: cvname !: variable name
      CHARACTER(LEN=1) :: cgrid  !: gridpoint (T, U, V, W, T)
      INTEGER          :: ndim   !: number of dimensions (2 or 3)

   END TYPE field
   TYPE field_list ! list of field to be treated
      INTEGER :: nfield
      TYPE(field), ALLOCATABLE, DIMENSION(:) :: fields
   END TYPE field_list
```

Listing 4: Example of field-list

```
      CASE('inc')
         plist%nfield = 5
         ALLOCATE ( plist%fields(plist%nfield) )
         plist%fields(1)%cvname = 'bckint'
         plist%fields(1)%cgrid  = 'T'
         plist%fields(2)%cvname = 'bckins'
         plist%fields(2)%cgrid  = 'T'
         plist%fields(3)%cvname = 'bckineta'
         plist%fields(3)%cgrid  = 'T'
         plist%fields(4)%cvname = 'bckinu'
         plist%fields(4)%cgrid  = 'U'
         plist%fields(5)%cvname = 'bckinv'
         plist%fields(5)%cgrid  = 'V'

         plist%fields(:)%ndim   = 3
         plist%fields(3)%ndim   = 2
```

6 Current limitations and future improvments

As it is the Simplification and Interpolation operators work fine, but there are several improvements that will be required in order to fulfill all the needs of an efficient multi incremental scheme (the following list may not be exhaustive)

- Some configurations of NEMO make use of the so-called 'partial steps', where the depth of the last cell on the vertical may be modified compared to the standard depth of the corresponding level, in order to better suit the local bathymetry. Since the interpolation used here is based on the observation operator that decompose the 3D interpolation into a 2D horizontal and a 1D vertical interpolation steps, it implicitly assume that all the point of a given level are at the same depth, therefore it cannot cope with the partial steps option that would require a real 3D interpolation scheme. Note that this is also a problem for the observation operator, but it is less crucial since the observations do not go very deep in general.

- The current implementation is not efficient enough memory-wise in mpp, indeed all the high resolution points are loaded on each processor even though it is using only the values that are located on the local domain. This is mainly an IO issue but it needs to be addressed in order to be used with very large configurations. Another possible path would be to make use of AGRIF's multiresolution capabilities to handle both low- and high-resolution grids as current NEMO grids at the same time.

- The grid search is one expensive part of the interpolation or the simplification, however it is unique between two given configurations. Therefore it could be computed once and for all and stored in a file.

References

Chauvin, C. (2010), Python interface for assimilation in nemo - reference manual, Technical report, VODA.
URL: *http://ljk.imag.fr/membres/Claire.Chauvin/VODA/piano-reference-manual.pdf*

Courtier, P., Thépaut, J.-N. & Hollingsworth, A. (1994), 'A strategy for operational implementation of 4d-var, using an incremental approach', *Q. J. R. Meteorol. Soc.* **120**, 1367–1387.

Daget, N. (2006), Interpolation d'une grille orca2 vers une grille régulière, Technical Report TR/CMGC/06/18, CERFACS.

Golub, G. & Van Loan, C. F. (1996), *Matrix Computations*, The Johns Hopkins University Presss, London.

Weaver, A. T., Deltel, C., Machu, E. & Ricci, S. (2005), 'A multivariate balance operator for variational ocean data assimilation. appeared in 2006 in a special issue.', *Q. J. R. Meteorol. Soc.* **131**, 3605–3625.