On the guess consistency in multi-incremental multi-resolution variational data assimilation

Nicolas Baillot d'Étiaux[1], Benjamin Ménétrier[2], Selime Gürol[3], and Yann Michel[4]

[1]CNRM, Université de Toulouse, Météo-France, CNRS, Toulouse, France
[2]IRIT, ...
[2]CERFACS, ...
[4]Météo-France, ...

Correspondence: Benjamin Ménétrier (benjamin.menetrier@irit.fr)

Abstract. Variational Data Assimilation (DA) schemes are often used to adress high dimensional non-linear problems in operational applications in the Numerical Wheather Prediction (NWP) domain. Because of the high computational cost of such minimization problems, various methods can be applied to improve the convergence at a reasonable numerical cost. One of these methods currently applied in operational DA schemes is the multi-incremental approach that consists in solving a succession of linearized versions of the original non-linear problem in several outer loops, by using well known algorithms to ensure the convergence of the linear problem at the inner loop level, and using the solution of the inner loops to update the problem at each outer loop. In order to save computational cost, the multi-incremental multi-resolution method consists in starting the minimization at a lower resolution than the original one, and increasing it at the outer loop level until the full resolution of the problem. In such a scheme, the way to compute the new guess at each outer loop from the previous iterations is crucial. We adress the question of the guess consistency in the standard method currently used in operational systems, and also present a new method which ensures the guess consistency and need simpler calculations. On the other hand, the conditionning of NWP problems is often poor, and one can use preconditionning techniques in order to improve the convergence. We have applied the multi-incremental multi-resolution scheme to a simplified problem in order to study the guess consistency as well as the equivalence between two well known preconditionnings ("full" $\mathbf{B}$ or "square root" $\mathbf{B}$) in such a scheme. Some equivalence conditions between the updating methods and the two preconditionnings are drawn according to the way the resolution change is realised at the outer loop level.

1 Introduction

intro...

2 Data Assimilation Problem

In Data Assimilation (DA), one wants to minimize the following non linear cost function representing the ability of a model state to be compatible with observations:

$$\mathcal{J}(\mathbf{x}) = \frac{1}{2}\left(\mathbf{x}-\mathbf{x}^b\right)^{\mathrm{T}}\mathbf{B}^{-1}\left(\mathbf{x}-\mathbf{x}^b\right) + \frac{1}{2}\left(\mathbf{y}^o-\mathcal{H}(\mathbf{x})\right)^{\mathrm{T}}\mathbf{R}^{-1}\left(\mathbf{y}^o-\mathcal{H}(\mathbf{x})\right), \tag{1}$$

where $\mathbf{x}\in\mathbb{R}^n$ is the state in model space of size n, $\mathbf{x}^b\in\mathbb{R}^n$ is the background state, $\mathbf{B}\in\mathbb{R}^{n\times n}$ is the background error covariance matrix, $\mathbf{y}^o\in\mathbb{R}^p$ is the observation vector in observation space of size p (note that in general $p<n$), $\mathbf{R}\in\mathbb{R}^{p\times p}$ is the observation error covariance matrix, and $\mathcal{H}:\mathbb{R}^n\to\mathbb{R}^p$ is the observation operator which maps the model space to the observation space.

2.1 Problem linearization

In general the observation operator is nonlinear and can be linearized around a guess state $\mathbf{x}_k^g\in\mathbb{R}^n$ so that:

$$\mathcal{H}(\mathbf{x}) \approx \mathcal{H}(\mathbf{x}_k^g) + \mathbf{H}_k\delta\mathbf{x}_k \quad \text{for } \mathbf{x}\approx\mathbf{x}_k^g, \tag{2}$$

defining the increment $\delta\mathbf{x}_k = \mathbf{x}-\mathbf{x}_k^g$, and where $\mathbf{H}_k\in\mathbb{R}^{p\times m}$ is the observation operator linearized around the guess state:

$$\mathbf{H}_{k,ij} = \left.\frac{\partial\mathcal{H}_i}{\partial x_j}\right|_{\mathbf{x}=\mathbf{x}_k^g}. \tag{3}$$

Instead of minimizing the full cost function $\mathcal{J}(\mathbf{x})$, it is now possible to minimize successive quadratic approximations around successive guess states:

$$J\left(\delta\mathbf{x}_k\right) = \frac{1}{2}\left(\delta\mathbf{x}_k-\delta\mathbf{x}_k^b\right)^{\mathrm{T}}\mathbf{B}^{-1}\left(\delta\mathbf{x}_k-\delta\mathbf{x}_k^b\right) + \frac{1}{2}\left(\mathbf{d}_k-\mathbf{H}_k\delta\mathbf{x}_k\right)^{\mathrm{T}}\mathbf{R}^{-1}\left(\mathbf{d}_k-\mathbf{H}_k\delta\mathbf{x}_k\right) \tag{4}$$

where k indicates the k^{th} iteration (hereafter these iterations are called "outer loops" since the minimization of the successive approximations are realized using well known iterative solvers such as Lanczos alorithms. We call "inner loops" the iterations of these algorithms), $\delta\mathbf{x}_k^b = \mathbf{x}^b-\mathbf{x}_k^g$ is the background increment and $\mathbf{d}_k = \mathbf{y}^o-\mathcal{H}(\mathbf{x}_k^g)$ is the innovation vector.

Setting the gradient of $J(\delta\mathbf{x}_k)$ to zero gives the analysis increment $\delta\mathbf{x}_k^a$:

$$\begin{aligned}
&\mathbf{B}^{-1}\left(\delta\mathbf{x}_k^a-\delta\mathbf{x}_k^b\right) - \mathbf{H}_k^{\mathrm{T}}\mathbf{R}^{-1}\left(\mathbf{d}_k-\mathbf{H}_k\delta\mathbf{x}_k^a\right) = 0\\
\Leftrightarrow\ &\left(\mathbf{B}^{-1}+\mathbf{H}_k^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{H}_k\right)\delta\mathbf{x}_k^a = \mathbf{B}^{-1}\delta\mathbf{x}_k^b + \mathbf{H}_k^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{d}_k\\
\Leftrightarrow\ &\boxed{\mathbf{A}_k^{\mathbf{x}}\delta\mathbf{x}_k^a = \mathbf{b}_k^{\mathbf{x}}}
\end{aligned} \tag{5}$$

where $\mathbf{A}_k^{\mathbf{x}}\in\mathbb{R}^{n\times n}$ and the right hand side $\mathbf{b}_k^{\mathbf{x}}\in\mathbb{R}^n$ are defined as:

$$\mathbf{A}_k^{\mathbf{x}} = \mathbf{B}^{-1} + \mathbf{H}_k^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{H}_k, \tag{6}$$

$$\mathbf{b}_k^{\mathbf{x}} = \mathbf{B}^{-1}\delta\mathbf{x}_k^b + \mathbf{H}_k^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{d}_k. \tag{7}$$

The problem can now be solved in this form using Gauss-Newton algorithm. It is very common to use the background state as a guess for the first iteration $k = 1$: $\mathbf{x}_1^g = \mathbf{x}^b$. Then for $k > 1$, the analysis of the previous iteration is used to define the guess: $\mathbf{x}_k^g = \mathbf{x}_{k-1}^a$. Thus, the first background increment is $\delta\mathbf{x}_1^b = \mathbf{x}_1^g - \mathbf{x}^b = 0$, and the following ones can be computed as:

$$\begin{aligned}\delta\mathbf{x}_k^b &= \mathbf{x}^b - \mathbf{x}_k^g,\\ &= \mathbf{x}^b - \mathbf{x}_{k-1}^a,\\ &= \mathbf{x}^b - \left(\mathbf{x}_{k-1}^g + \delta\mathbf{x}_{k-1}^a\right),\\ &= \delta\mathbf{x}_{k-1}^b - \delta\mathbf{x}_{k-1}^a,\end{aligned} \tag{8}$$

which can be combined recursively to yield:

$$\delta\mathbf{x}_k^b = -\sum_{i=1}^{k-1} \delta\mathbf{x}_i^a. \tag{9}$$

In general, the condition number of this problem is poor, and one has to use preconditionning techniques to improve it.

2.2 Preconditionning

In this section we describe the square root $\mathbf{B}$ preconditionning, which is widely used in DA. The $\mathbf{B}$ matrix have the important property of being positive definite, so that there is an infinity of square-roots $\mathbf{U}$ verifying $\mathbf{B} = \mathbf{U}\mathbf{U}^{\mathrm{T}}$. The square root $\mathbf{B}$ preconditionning consists in defining a new control variable $\delta\mathbf{v}_k = \mathbf{U}^{\mathrm{T}}\mathbf{B}^{-1}\delta\mathbf{x}_k$, so that the linear system (5) can now be written in control space as $\mathbf{A}_k^{\mathbf{v}}\delta\mathbf{v}_k^a = \mathbf{b}_k^{\mathbf{v}}$ with:

$$\mathbf{A}_k^{\mathbf{v}} = \mathbf{I}_m + \mathbf{U}^{\mathrm{T}}\mathbf{H}_k^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{H}_k\mathbf{U}, \tag{10}$$

$$\mathbf{b}_k^{\mathbf{v}} = \delta\mathbf{v}_k^b + \mathbf{U}^{\mathrm{T}}\mathbf{H}_k^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{d}_k. \tag{11}$$

Using this technique allows to recursively solve the system without using $\mathbf{B}^{-1}$ which is, in general, not available due to its high dimension, even if it is needed in general to compute the right-hand side $\mathbf{b}_k^{\mathbf{v}}$. With this preconditionning, $\mathbf{U}^{\mathrm{T}}\mathbf{B}^{-1}$ can be applied on both side of equation (9), leading to:

$$\delta\mathbf{v}_k^b = -\sum_{i=1}^{k-1} \delta\mathbf{v}_i^a, \tag{12}$$

which can be used to compute the right hand side $\mathbf{b}_k^{\mathbf{v}}$ without requiring $\mathbf{B}^{-1}$.

2.3 Changing the resolution between the outer loops

In special cases, the background error covariance matrix can be updated between outer iterations defining $\mathbf{B}_k$ matrix at iteration k, and its square-root $\mathbf{U}_k$. In this case, it is not systematically possible to compute the background increment without using $\mathbf{B}^{-1}$. One example of such a scheme is the multi-incremental multi-resolution approach, in which the resolution increases at each outer loop for computational efficiency, and therefore, the $\mathbf{B}$ matrix depends on k. In this case, Equation (12) is valid and one can obtain the background increment as follows:

$$\begin{aligned}\delta\mathbf{v}_k^b &= -\mathbf{U}_k^{\mathrm{T}}\mathbf{B}_k^{-1}\sum_{i=1}^{k-1}\delta\mathbf{x}_i^a,\\ &= -\sum_{i=1}^{k-1}\mathbf{U}_k^{\mathrm{T}}\mathbf{B}_k^{-1}\mathbf{U}_i\delta\mathbf{v}_i^a.\end{aligned} \tag{13}$$

It should be emphasized that if $\mathbf{U}_k^{\mathrm{T}}\mathbf{B}_k^{-1}\mathbf{U}_i\delta\mathbf{v}_i^a \neq \delta\mathbf{v}_i^a$, equation (12) cannot be used consistently.

To realize the change of resolution, one needs to use interpolators in model space $\mathbf{T}^{\mathbf{x}}_{i\to k} \in \mathbb{R}^{n_k \times n_i}$ and in control space $\mathbf{T}^{\mathbf{v}}_{i\to k} \in \mathbb{R}^{m_k \times m_i}$ from the resolution $\mathcal{R}_i$ to $\mathcal{R}_k$, where n_k and m_k respectively denotes the size of the model space and the control space. For clarity, we now assume that the resolutions are stricly increasing: for $i < k$, $n_i < n_k$ and $m_i < m_k$, and that the full resolution is obtained at the last iteration K. We define two interpolators from resolution $\mathcal{R}_i$ to resolution $\mathcal{R}_k$:

- $\mathbf{T}^{\mathbf{x}}_{i\to k} \in \mathbb{R}^{n_k \times n_i}$ in model space,
- $\mathbf{T}^{\mathbf{v}}_{i\to k} \in \mathbb{R}^{m_k \times m_i}$ in control space,

A special class of interpolators called "transitive interpolators" have three extra properties:

- Upscaling transitivity: for $n_i < n_j$ and $n_i < n_k$:

$$\mathbf{T}^{\mathbf{x}}_{j\to k}\mathbf{T}^{\mathbf{x}}_{i\to j} = \mathbf{T}^{\mathbf{x}}_{i\to k} \tag{14}$$

- Downscaling transitivity: for $n_i < n_j < n_k$:

$$\mathbf{T}^{\mathbf{x}}_{j\to i}\mathbf{T}^{\mathbf{x}}_{k\to j} = \mathbf{T}^{\mathbf{x}}_{k\to i} \tag{15}$$

- Right-inverse: for $n_i < n_k$

$$\mathbf{T}^{\mathbf{x}}_{k\to i}\mathbf{T}^{\mathbf{x}}_{i\to k} = \mathbf{I}_{n_i} \tag{16}$$

and similarly for $\mathbf{T}^{\mathbf{v}}_{i\to k}$ in control space, replacing n with m.

A generic family of transitive interpolators is based on a zero-padding operator surrounded by an orthogonal transform. For instance in model space:

$$\mathbf{T}^{\mathbf{x}}_{i\to k} = \mathbf{S}_k^{\mathrm{T}}\boldsymbol{\Delta}_{i\to k}\mathbf{S}_i, \tag{17}$$

where

- $\mathbf{\Delta}_{i\rightarrow k} \in \mathbb{R}^{n_k \times n_i}$ is a zero-padding operator:

$$\Delta_{i\rightarrow k,\alpha\beta} = \begin{cases} 1 & \text{if} \quad \alpha = \beta \\ 0 & \text{if} \quad \alpha \neq \beta \end{cases} \tag{18}$$

- $\mathbf{S}_k \in \mathbb{R}^{n_k \times n_k}$ is any orthogonal transform:

$$\mathbf{S}_k\mathbf{S}_k^{\mathrm{T}} = \mathbf{S}_k^{\mathrm{T}}\mathbf{S}_k = \mathbf{I}_{n_k}. \tag{19}$$

It is easy to check the three required properties:

- Upscaling transitivity: for $n_i \leq n_j$ and $n_i \leq n_k$:

$$\begin{aligned} \mathbf{T}^{\mathbf{x}}_{j\rightarrow k}\mathbf{T}^{\mathbf{x}}_{i\rightarrow j} &= \mathbf{S}_k^{\mathrm{T}}\mathbf{\Delta}_{j\rightarrow k}\mathbf{S}_j\mathbf{S}_j^{\mathrm{T}}\mathbf{\Delta}_{i\rightarrow j}\mathbf{S}_i \\ &= \mathbf{S}_k^{\mathrm{T}}\mathbf{\Delta}_{j\rightarrow k}\mathbf{\Delta}_{i\rightarrow j}\mathbf{S}_i \\ &= \mathbf{S}_k^{\mathrm{T}}\mathbf{\Delta}_{i\rightarrow k}\mathbf{S}_i \\ &= \mathbf{T}^{\mathbf{x}}_{i\rightarrow k}. \end{aligned} \tag{20}$$

- Downscaling transitivity: for $n_i \leq n_j \leq n_k$:

$$\begin{aligned} \mathbf{T}^{\mathbf{x}}_{j\rightarrow i}\mathbf{T}^{\mathbf{x}}_{k\rightarrow j} &= \mathbf{S}_i^{\mathrm{T}}\mathbf{\Delta}_{j\rightarrow i}\mathbf{S}_j\mathbf{S}_j^{\mathrm{T}}\mathbf{\Delta}_{k\rightarrow j}\mathbf{S}_k \\ &= \mathbf{S}_i^{\mathrm{T}}\mathbf{\Delta}_{j\rightarrow i}\mathbf{\Delta}_{k\rightarrow j}\mathbf{S}_k \\ &= \mathbf{S}_i^{\mathrm{T}}\mathbf{\Delta}_{k\rightarrow i}\mathbf{S}_k \\ &= \mathbf{T}^{\mathbf{x}}_{k\rightarrow i}. \end{aligned} \tag{21}$$

- Right-inverse: for $n_i \leq n_k$

$$\begin{aligned} \mathbf{T}^{\mathbf{x}}_{k\rightarrow i}\mathbf{T}^{\mathbf{x}}_{i\rightarrow k} &= \mathbf{S}_i^{\mathrm{T}}\mathbf{\Delta}_{k\rightarrow i}\mathbf{S}_k\mathbf{S}_k^{\mathrm{T}}\mathbf{\Delta}_{i\rightarrow k}\mathbf{S}_i \\ &= \mathbf{S}_i^{\mathrm{T}}\mathbf{\Delta}_{k\rightarrow i}\mathbf{\Delta}_{i\rightarrow k}\mathbf{S}_i \\ &= \mathbf{S}_i^{\mathrm{T}}\mathbf{S}_i \\ &= \mathbf{I}_{n_i}. \end{aligned} \tag{22}$$

Associated to the various resolutions, we qualify a $\mathbf{B}$ matrix family as "projective" if the low-resolution members can be defined as a projection of the high-resolution ones, using transitive interpolators. For $n_i < n_k$, that would mean:

$$\mathbf{U}_k\mathbf{T}^{\mathbf{v}}_{i\rightarrow k} = \mathbf{T}^{\mathbf{x}}_{i\rightarrow k}\mathbf{U}_i. \tag{23}$$

Finally, the multi-resolution problem should be solved with the following requirements in mind:

- The background $\mathbf{x}^b$ is provided at full resolution, but it can be simplified at resolution $\mathcal{R}_k$:

  $$\mathbf{x}_k^b = \mathbf{T}_{K\to k}^{\mathbf{x}} \mathbf{x}^b \tag{24}$$

- A full resolution guess denoted $\mathbf{x}_k^{g+}$ has to be computed at each outer iteration to run model trajectories used in the operators linearization. This full resolution guess can be simplified at resolution $\mathcal{R}_k$ to give the actual guess of the outer iteration k:

  $$\mathbf{x}_k^g = \mathbf{T}_{K\to k}^{\mathbf{x}} \mathbf{x}_k^{g+} \tag{25}$$

- Only δ-quantities should be interpolated to higher resolution, and then possibly added to full quantities at full resolution.

3 Guess consistency

In the linear systems solved at each outer iteration, the the full resolution guess $\mathbf{x}_k^{g+}$ appears for two distinct purposes:

- First, it is explicitely defined to be used as the linearization state for the observation operator at each outer loops and to compute the innovation $\mathbf{d}_k$. The first guess for $k=1$ is taken as the background state, also provided at full resolution $\mathbf{x}_1^{g+} = \mathbf{x}^b$. At the end of iteration k, without any loss of generality, we can define the analysis increment at full resolution $\delta\mathbf{x}_{k-1}^{a+}$ that updates the full resolution guess:

  $$\mathbf{x}_k^{g+} = \mathbf{x}_{k-1}^{g+} + \delta\mathbf{x}_{k-1}^{a+} \tag{26}$$

  The way the analysis increment at full resolution $\delta\mathbf{x}_{k-1}^{a+}$ is obtained from the products of previous outer iterations does not matter at this point.

- Second, it is implicitly present in the first term of the right-hand side:

  $$\begin{aligned}\delta\mathbf{v}_k^b &= \mathbf{U}_k^{\mathrm{T}}\mathbf{B}_k^{-1}\delta\mathbf{x}_k^b \\ &= \mathbf{U}_k^{\mathrm{T}}\mathbf{B}_k^{-1}\left(\mathbf{x}_k^b - \mathbf{x}_k^g\right) \\ &= \mathbf{U}_k^{\mathrm{T}}\mathbf{B}_k^{-1}\mathbf{T}_{K\to k}^{\mathbf{x}}\left(\mathbf{x}^b - \mathbf{x}_k^{g+}\right)\end{aligned} \tag{27}$$

The question of the consistency between these two occurences of the guess state in the minimization is crucial and is discussed in the following sections. We compare different methods to compute the guess and present a new method ensuring its consistency.

3.1 Theoretical method

In the "theoretical" method, one assumes that $\mathbf{B}_k^{-1}$ is available and can be used to compute the increment $\delta\mathbf{v}_k^b$ explicitly from $\mathbf{x}_k^{g+}$ using equations (27). In this method, there is no requirement on the expression of the analysis increment at full resolution $\delta\mathbf{x}_{k-1}^{a+}$.

3.2 Standard method

In the "standard" method, which is currently used in operational multi-incremental multi-resolution schemes, $\mathbf{B}_k^{-1}$ is not available and the first term of the right-hand side is computed separately, using transformed versions of equation (12) with appropriate interpolations:

$$\delta\mathbf{v}_k^b = -\sum_{i=1}^{k-1}\mathbf{T}^{\mathbf{v}}_{i\rightarrow k}\delta\mathbf{v}_i^a. \tag{28}$$

This additional constraint imposes a unique expression for the analysis increment at full resolution $\delta\mathbf{x}_{k-1}^{a+}$, if the guess consistency has to be maintained.

3.2.1 General case

Using equation (26) and the fact that the first full resolution guess is taken as the background state, the full resolution guess at iteration k can be expressed as:

$$\mathbf{x}_k^{g+} = \mathbf{x}^b + \sum_{i=1}^{k-1}\delta\mathbf{x}_i^{a+} \tag{29}$$

which is equivalent to:

$$\mathbf{x}^b - \mathbf{x}_k^{g+} = -\sum_{i=1}^{k-1}\delta\mathbf{x}_i^{a+} \tag{30}$$

This result can be introduced into equation (27) to get the background increment as a function of the analysis increment at full resolution $\delta\mathbf{x}_i^{a+}$:

$$\delta\mathbf{v}_k^b = -\mathbf{U}_k^{\mathrm{T}}\mathbf{B}_k^{-1}\mathbf{T}^{\mathbf{x}}_{K\rightarrow k}\sum_{i=1}^{k-1}\delta\mathbf{x}_i^{a+} \tag{31}$$

Comparing equations (28) and (31), the guess consistency is maintained if:

$$\begin{aligned}
&\mathbf{U}_k^{\mathrm{T}}\mathbf{B}_k^{-1}\mathbf{T}^{\mathbf{x}}_{K\rightarrow k}\sum_{i=1}^{k-1}\delta\mathbf{x}_i^{a+} = \sum_{i=1}^{k-1}\mathbf{T}^{\mathbf{v}}_{i\rightarrow k}\delta\mathbf{v}_i^a,\\
\Leftrightarrow\ &\mathbf{T}^{\mathbf{x}}_{K\rightarrow k}\sum_{i=1}^{k-1}\delta\mathbf{x}_i^{a+} = \mathbf{U}_k\sum_{i=1}^{k-1}\mathbf{T}^{\mathbf{v}}_{i\rightarrow k}\delta\mathbf{v}_i^a,\\
\Leftrightarrow\ &\boxed{\mathbf{T}^{\mathbf{x}}_{K\rightarrow k}\delta\mathbf{x}_{k-1}^{a+} = \mathbf{U}_k\sum_{i=1}^{k-1}\mathbf{T}^{\mathbf{v}}_{i\rightarrow k}\delta\mathbf{v}_i^a - \mathbf{T}^{\mathbf{x}}_{K\rightarrow k}\sum_{i=1}^{k-2}\delta\mathbf{x}_i^{a+}}.
\end{aligned} \tag{32}$$

To understand this result, one needs to distinguish between two cases according to the number of outer loops to be realized: if $K = 2$, equation (32) corresponding to the last iteration $k = K = 2$ becomes $\delta\mathbf{x}_1^{a+} = \mathbf{U}_2\mathbf{T}^{\mathbf{v}}_{1\rightarrow 2}\delta\mathbf{v}_1^a$ leading to an explicit expression of $\delta\mathbf{x}_1^{a+}$. In the case where $K > 2$, equation (32) can be solved explicitly if $\mathbf{T}^{\mathbf{x}}_{K\to k}$ has a known right-inverse denoted $(\mathbf{T}^{\mathbf{x}}_{K\to k})^{-1}_{\text{right}}$, such that $\mathbf{T}^{\mathbf{x}}_{K\to k}(\mathbf{T}^{\mathbf{x}}_{K\to k})^{-1}_{\text{right}} = \mathbf{I}_{n_k}$. In this case, equation (32) becomes:

$$\delta\mathbf{x}_{k-1}^{a+} = (\mathbf{T}^{\mathbf{x}}_{K\to k})^{-1}_{\text{right}} \left(\mathbf{U}_k \sum_{i=1}^{k-1} \mathbf{T}^{\mathbf{v}}_{i\to k} \delta\mathbf{v}_i^a - \mathbf{T}^{\mathbf{x}}_{K\to k} \sum_{i=1}^{k-2} \delta\mathbf{x}_i^{a+} \right) \tag{33}$$

Note that if this right-inverse exists, its unicity depends on the rank of the kernel of $\mathbf{T}^{\mathbf{x}}_{K\to k}$: if $\delta\mathbf{x}_{k-1}^{a+}$ is a solution of equation (32) and $\mathbf{u} \in \text{Ker}\left(\mathbf{T}^{\mathbf{x}}_{K\to k}\right)$, then $\left(\delta\mathbf{x}_{k-1}^{a+} + \mathbf{u}\right)$ is also a solution.

3.2.2 Complete standard method

If one uses transitive interpolators, the right inverse of $\mathbf{T}^{\mathbf{x}}_{K\to k}$ exists and is defined as $\mathbf{T}^{\mathbf{x}}_{k\to K}$, so that:

$$\boxed{\delta\mathbf{x}_{k-1}^{a+} = \mathbf{T}^{\mathbf{x}}_{k\to K} \left(\mathbf{U}_k \sum_{i=1}^{k-1} \mathbf{T}^{\mathbf{v}}_{i\to k} \delta\mathbf{v}_i^a - \mathbf{T}^{\mathbf{x}}_{K\to k} \sum_{i=1}^{k-2} \delta\mathbf{x}_i^{a+} \right)} \tag{34}$$

Equation (34) estimates $\delta\mathbf{x}_{k-1}^{a+}$ using results from outer iterations 1 to $k-1$, not outer iteration $k-1$ only. Hereafter, this expression is refered to as "complete expression".

3.2.3 Simplified standard method

The right-hand side of equation (34) can be split in order to extract the term coming from outer iteration $k-1$:

$$\delta\mathbf{x}_{k-1}^{a+} = \mathbf{T}^{\mathbf{x}}_{k\to K}\mathbf{U}_k\mathbf{T}^{\mathbf{v}}_{k-1\to k}\delta\mathbf{v}_{k-1}^a + \mathbf{T}^{\mathbf{x}}_{k\to K} \sum_{i=1}^{k-2} \left(\mathbf{U}_k\mathbf{T}^{\mathbf{v}}_{i\to k}\delta\mathbf{v}_i^a - \mathbf{T}^{\mathbf{x}}_{K\to k}\delta\mathbf{x}_i^{a+} \right) \tag{35}$$

If the $\mathbf{B}$ family is projective, then:

$$\begin{aligned} \mathbf{T}^{\mathbf{x}}_{k\to K}\mathbf{U}_k\mathbf{T}^{\mathbf{v}}_{k-1\to k} &= \mathbf{T}^{\mathbf{x}}_{k\to K}\mathbf{T}^{\mathbf{x}}_{k-1\to k}\mathbf{U}_{k-1} \\ &= \mathbf{T}^{\mathbf{x}}_{k-1\to K}\mathbf{U}_{k-1} \end{aligned} \tag{36}$$

so a simplified expression of $\delta\mathbf{x}_{k-1}^{a+}$ is:

$$\boxed{\delta\mathbf{x}_{k-1}^{a+} = \mathbf{T}^{\mathbf{x}}_{k-1\to K}\mathbf{U}_{k-1}\delta\mathbf{v}_{k-1}^a} \tag{37}$$

that verifies (34) since both terms inside the summation cancel each other:

$$\begin{aligned} \mathbf{U}_k\mathbf{T}^{\mathbf{v}}_{i\to k}\delta\mathbf{v}_i^a - \mathbf{T}^{\mathbf{x}}_{K\to k}\delta\mathbf{x}_i^{a+} &= \mathbf{U}_k\mathbf{T}^{\mathbf{v}}_{i\to k}\delta\mathbf{v}_i^a - \mathbf{T}^{\mathbf{x}}_{K\to k}\mathbf{T}^{\mathbf{x}}_{i\to K}\mathbf{U}_i\delta\mathbf{v}_i^a \\ &= \left(\mathbf{U}_k\mathbf{T}^{\mathbf{v}}_{i\to k} - \mathbf{T}^{\mathbf{x}}_{i\to k}\mathbf{U}_i\right)\delta\mathbf{v}_i^a \\ &= 0 \end{aligned} \tag{38}$$

Equation (37) now estimates $\delta\mathbf{x}_{k-1}^{a+}$ using results from outer iterations $k-1$ only and this expression is refered to as "simplified expression" in the following.

3.3 Consitent method

An new alternative method in which the guess consistency is guaranteed can be defined. This method also have the advantage of requiring much more simpler calculations than the "standard" method and gives the same results. The basic idea is to reverse the order of computations: the first term of the right-hand side is computed first from equation (28), and then the background increment is given by:

$$\delta \mathbf{x}_k^b = \mathbf{U}_k \delta \mathbf{v}_k^b \tag{39}$$

Finally, the full resolution guess is deduced as:

$$\mathbf{x}_k^{g+} = \mathbf{x}^b - \mathbf{T}_{k \to K}^{\mathbf{x}} \delta \mathbf{x}_k^b, \tag{40}$$

ensuring its consistency.

3.4 Methods comparision

Guess consistency is always ensured for the "theoretical" and "consistent" methods. For the "standard" method, it is only ensured if one uses transitive interpolators, and the correct formula for the analysis increment at full resolution $\delta \mathbf{x}_{k-1}^{a+}$. In the general case, it depends whether equation (32) can be solved or not. In the general case, if a solution exists for the "standard" method, the same $\delta \mathbf{x}_{k-1}^{a+}$ is used in the "theoretical" method and all methods give the same results. In the case where no solution can be found for the "standard" method, another expression is used for the "theoretical" one, which leads to different results from the "consistent" method. In the case of transitive interpolators, "standard" and "consistent" methods always give the same results and are in agreement with the "theoretical" method if the same expression is used for $\delta \mathbf{x}_{k-1}^{a+}$. The case of a projective $\mathbf{B}$ family is just a subcase of transitive interpolators one, where the expression of $\delta \mathbf{x}_{k-1}^{a+}$ can be simplified for the "theoretical" and "standard" methods.

The figure (1), (2) and (3) respectively illustrate the workflow of the three methods.

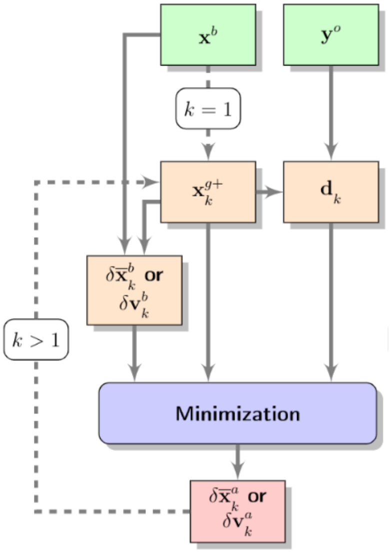
$\mathbf{x}^b$
$\mathbf{y}^o$
$k=1$
$\mathbf{x}_k^{g+}$
$\mathbf{d}_k$
$\delta\overline{\mathbf{x}}_k^b$ or $\delta\mathbf{v}_k^b$
$k>1$
Minimization
$\delta\overline{\mathbf{x}}_k^a$ or $\delta\mathbf{v}_k^a$

Figure 1. Illustrative workflow of the theoretical method. The full resolution background and guess are used at each outer iteration to compute the $\delta\overline{\mathbf{x}}_k^b$ or $\delta\mathbf{v}_k^b$, using $\mathbf{B}_k^{-1}$:

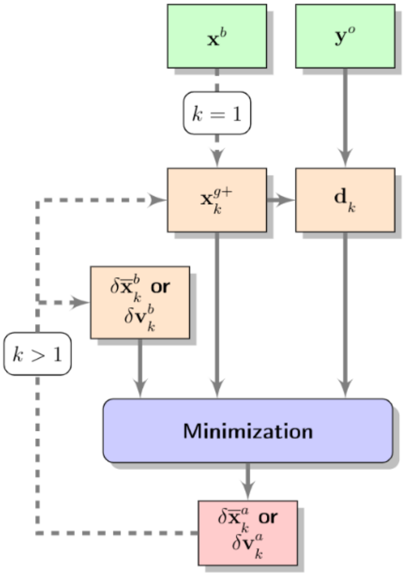
$\mathbf{x}^b$
$\mathbf{y}^o$
$k=1$
$\mathbf{x}_k^{g+}$
$\mathbf{d}_k$
$\delta\overline{\mathbf{x}}_k^b$ or $\delta\mathbf{v}_k^b$
$k>1$
Minimization
$\delta\overline{\mathbf{x}}_k^a$ or $\delta\mathbf{v}_k^a$

Figure 2. Illustrative workflow of the standard method. The potential guess inconsistency of the "standard" method comes from the fork in the use of the minimization output, to compute the full resolution guess and the first term of the right-hand side independently. If the correct formula is not used, then a guess inconsistency can arise.

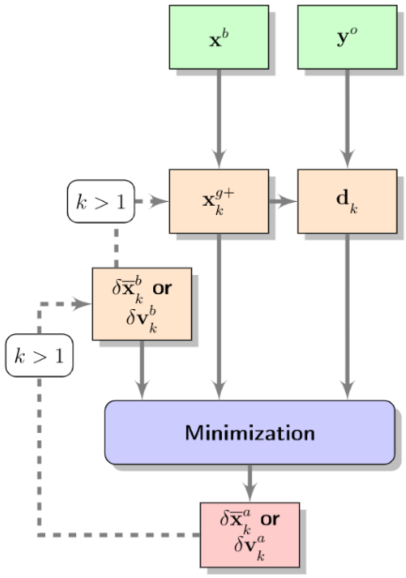
$\mathbf{x}^b$
$\mathbf{y}^o$
$k > 1$
$\mathbf{x}_k^{g+}$
$\mathbf{d}_k$
$\delta\overline{\mathbf{x}}_k^b$ or $\delta\mathbf{v}_k^b$
$k > 1$
Minimization
$\delta\overline{\mathbf{x}}_k^a$ or $\delta\mathbf{v}_k^a$

Figure 3. Illustrative workflow of the consistent method. Because of it closed cycle form (no fork), the "consistent" method ensures the guess consistency, without requiering $\mathbf{B}_k^{-1}$.

The fork in the "standard" method using the analysis of iteration k to compute separately the background increment and the guess of iteration $k+1$ is responsible of the potential inconsistency of the guess.

The differences between the three methods coming from this guess inconsistency are illustraed using a toy problem which is describe in section 5.

Another important problem is the equivalence between the square root $\mathbf{B}$ preconditionning and the full $\mathbf{B}$ preconditionning which may also be used for computationnal efficiency. This equivalence in the case of a multi-incremental multi-resolution framework is discussed in the following section.

4 Equivalence between preconditionners

(NB: or "full B precontionning" - is it necessary to put the equations ? Maybe we can just say that the same steps can be realized with B.)

Another preconding technique consists in defining a new variable $\delta\overline{\mathbf{x}}_k = \mathbf{B}_k^{-1}\delta\mathbf{x}_k$ so that the linear system to solve can be written as:

$$\mathbf{A}_k^{\overline{\mathbf{x}}}\delta\overline{\mathbf{x}}_k^a = \mathbf{b}_k^{\overline{\mathbf{x}}}, \tag{41}$$

$\delta\overline{\mathbf{x}}_k^a$ being the preconditionned analysis increment, and with:

$$\mathbf{A}_k^{\overline{\mathbf{x}}} = \mathbf{I}_n + \mathbf{H}_k^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{H}_k\mathbf{B}_k, \tag{42}$$

and the right hand side now written as:

$$\mathbf{b}_k^{\overline{\mathbf{x}}} = \delta\overline{\mathbf{x}}_k^b + \mathbf{H}_k^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{d}_k. \tag{43}$$

An equivalent form of equation (44) can be found by applying $\mathbf{B}^{-1}$ on both side of equation (9) to express the so preconditionned background increment as:

$$\delta\overline{\mathbf{x}}_k^b = -\sum_{i=1}^{k-1}\delta\overline{\mathbf{x}}_i^a. \tag{44}$$

In the same way as for the square root $\mathbf{B}$ preconditionning, if the $\mathbf{B}$ matrix changes between the outer loops and so depends on k, one can write:

$$\begin{aligned}\delta\overline{\mathbf{x}}_k^b &= -\mathbf{B}_k^{-1}\sum_{i=1}^{k-1}\delta\mathbf{x}_i^a \\ &= -\sum_{i=1}^{k-1}\mathbf{B}_k^{-1}\mathbf{B}_i\delta\overline{\mathbf{x}}_i^a\end{aligned} \tag{45}$$

and similarly if $\mathbf{B}_k^{-1}\mathbf{B}_i\delta\overline{\mathbf{x}}_i^a \neq \delta\overline{\mathbf{x}}_i^a$, equation (44) cannot be used consistently.

One can also write equivalent forms of equations (28) and (31) using this preconditionner in the "standard" method:

$$\delta\overline{\mathbf{x}}_k^b = -\sum_{i=1}^{k-1}\mathbf{T}_{i\to k}^{\mathbf{x}}\delta\overline{\mathbf{x}}_i^a, \tag{46}$$

$$\delta\overline{\mathbf{x}}_k^b = -\mathbf{B}_k^{-1}\mathbf{T}_{K\to k}^{\mathbf{x}}\sum_{i=1}^{k-1}\delta\mathbf{x}_i^{a+}. \tag{47}$$

Now comparing equations (46) and (47), the guess consistency is maintained if:

$$\begin{aligned}&\mathbf{B}_k^{-1}\mathbf{T}_{K\to k}^{\mathbf{x}}\sum_{i=1}^{k-1}\delta\mathbf{x}_i^{a+} = \sum_{i=1}^{k-1}\mathbf{T}_{i\to k}^{\mathbf{x}}\delta\overline{\mathbf{x}}_i^a, \\ \Leftrightarrow\ &\mathbf{T}_{K\to k}^{\mathbf{x}}\sum_{i=1}^{k-1}\delta\mathbf{x}_i^{a+} = \mathbf{B}_k\sum_{i=1}^{k-1}\mathbf{T}_{i\to k}^{\mathbf{x}}\delta\overline{\mathbf{x}}_i^a, \\ \Leftrightarrow\ &\boxed{\mathbf{T}_{K\to k}^{\mathbf{x}}\delta\mathbf{x}_{k-1}^{a+} = \mathbf{B}_k\sum_{i=1}^{k-1}\mathbf{T}_{i\to k}^{\mathbf{x}}\delta\overline{\mathbf{x}}_i^a - \mathbf{T}_{K\to k}^{\mathbf{x}}\sum_{i=1}^{k-2}\delta\mathbf{x}_i^{a+}}.\end{aligned} \tag{48}$$

Once again, one can write a complete "expression" of the full resolution analysis increment if a right inverse exists for the interpolation operator:

$$\delta\mathbf{x}_{k-1}^{a+} = (\mathbf{T}_{K\to k}^{\mathbf{x}})_{\text{right}}^{-1}\left(\mathbf{B}_k\sum_{i=1}^{k-1}\mathbf{T}_{i\to k}^{\mathbf{x}}\delta\overline{\mathbf{x}}_i^a - \mathbf{T}_{K\to k}^{\mathbf{x}}\sum_{i=1}^{k-2}\delta\mathbf{x}_i^{a+}\right), \tag{49}$$

leading to:

$$\delta\mathbf{x}^{a+}_{k-1} = \mathbf{T}^{\mathbf{x}}_{k\to K}\left(\mathbf{B}_k\sum_{i=1}^{k-1}\mathbf{T}^{\mathbf{x}}_{i\to k}\delta\overline{\mathbf{x}}^a_i - \mathbf{T}^{\mathbf{x}}_{K\to k}\sum_{i=1}^{k-2}\delta\mathbf{x}^{a+}_i\right), \tag{50}$$

in the case of transitive interpolators. The right-hand side of equation (50) can be split to extract the term coming from outer iteration $k-1$:

$$\delta\mathbf{x}^{a+}_{k-1} = \mathbf{T}^{\mathbf{x}}_{k\to K}\mathbf{B}_k\mathbf{T}^{\mathbf{x}}_{k-1\to k}\delta\overline{\mathbf{x}}^a_{k-1} + \mathbf{T}^{\mathbf{x}}_{k\to K}\sum_{i=1}^{k-2}\left(\mathbf{B}_k\mathbf{T}^{\mathbf{x}}_{i\to k}\delta\overline{\mathbf{x}}^a_i - \mathbf{T}^{\mathbf{x}}_{K\to k}\delta\mathbf{x}^{a+}_i\right), \tag{51}$$

and if the $\mathbf{B}$ family is projective, then:

$$\begin{aligned}\mathbf{T}^{\mathbf{x}}_{k\to K}\mathbf{B}_k\mathbf{T}^{\mathbf{x}}_{k-1\to k} &= \mathbf{T}^{\mathbf{x}}_{k\to K}\mathbf{T}^{\mathbf{x}}_{k-1\to k}\mathbf{B}_{k-1}\\ &= \mathbf{T}^{\mathbf{x}}_{k-1\to K}\mathbf{B}_{k-1}\end{aligned} \tag{52}$$

so a simplified expression of $\delta\mathbf{x}^{a+}_{k-1}$ can be written as:

$$\boxed{\delta\mathbf{x}^{a+}_{k-1} = \mathbf{T}^{\mathbf{x}}_{k-1\to K}\mathbf{B}_{k-1}\delta\overline{\mathbf{x}}^a_{k-1}} \tag{53}$$

that verifies (50) since both terms inside the summation cancel each other:

$$\begin{aligned}\mathbf{B}_k\mathbf{T}^{\mathbf{x}}_{i\to k}\delta\overline{\mathbf{x}}^a_i - \mathbf{T}^{\mathbf{x}}_{K\to k}\delta\mathbf{x}^{a+}_i &= \mathbf{B}_k\mathbf{T}^{\mathbf{x}}_{i\to k}\delta\overline{\mathbf{x}}^a_i - \mathbf{T}^{\mathbf{x}}_{K\to k}\mathbf{T}^{\mathbf{x}}_{i\to K}\mathbf{B}_i\delta\overline{\mathbf{x}}^a_i\\ &= \left(\mathbf{B}_k\mathbf{T}^{\mathbf{x}}_{i\to k} - \mathbf{T}^{\mathbf{x}}_{i\to k}\mathbf{B}_i\right)\delta\overline{\mathbf{x}}^a_i\\ &= 0\end{aligned} \tag{54}$$

Finally, in the consistent methos, the background increment is simply obtained as:

$$\delta\mathbf{x}^b_k = \mathbf{B}_k\delta\overline{\mathbf{x}}^b_k. \tag{55}$$

The full $\mathbf{B}$ preconditionning is mathematically equivalent to the square root $\mathbf{B}$ one, but in the case of a multi-incremental multi-resolution scheme where the $\mathbf{B}$ matrix depends on k, this equivalence is not guaranteed anymore. We discuss this point in more details in section 5.

5 Methodology and results

(NB: talk about the tests we have done with nearest neighbor interpolation ?)

In order to illustrate the differences arising between the three methods due to the guess consistency, we have compared the results obtained with all the methods on a toy problem detailed in the present section.

5.1 Methodology

In this section we illustrate the differences arising between the three above-mentionned methods related to the guess consistency. To do so, we have built a bidimensional toy problem as follows:

- We consider a periodic domain composed of bidimensional square grids with $m_x = 101$ x-coordinates and $m_y = 101$ y-coordinates (leading to a total of $m = m_x \times m_y$ points) and assume that this resolution corresponds to the full resolution of our prolem (e.g. this resolution corresponds to the last iteration $k = K$ of our multi-incremental multi-resolution scheme).

- A $\mathbf{B}$ matrix is introduced with gaussian spectral variances defined as:

  $$\mathbf{B}_{\mathbf{k},\mathbf{l}} = e^{-2((k^2+l^2)\pi L_b)^2}, \tag{56}$$

  where $k \in [0 : m_x/2]$, $l \in [-m_y/2 : m_y/2]$ are the accessible wave numbers and L_b is a correlation lenghtscale. We also define its grid point variance as: (NB: quand on prend sigmabvar = 0, quest ce qu'on fait ?)

  $$\mathbf{B}_{\mathbf{i},\mathbf{j}} = 1 + \sigma^b \sin(2\pi x_i) \sin(2\pi y_j), \tag{57}$$

  where σ^b can be modified, x_i is the i^{th} x-coordinate of the grid, and y_j is the j^{th} y-coordinate of the grid.

- A full resolution truth state $\mathbf{x^t}$ is defined on this grid by applying the square root of $\mathbf{B}$ (denoted $\mathbf{U}$) on a random field ν^t with normal distribution: (NB: I do not remember why is there $x^t = 1 + x^t$ in the code? - Find a way to write the identity operator):

  $$\mathbf{x}^t = 1_m + \mathbf{U}\nu^t. \tag{58}$$

- A full resolution background state is defined in same manner with a different random field ν^b (NB: idem for this):

  $$\mathbf{x}_b = \mathbf{x}^t + \mathbf{U}\nu^b. \tag{59}$$

- In order to have the simpler problem, the observation operator is taken to be the identity. We also introduce $p = 2000$ observation points that are not necessarily on grids points, and a linear observation operator: (NB: we do not discuss about the cubic version ?)

  $$\mathcal{H}\mathbf{x} = \mathbf{x}. \tag{60}$$

- Finally the observation error covariance matrix $\mathbf{R} \in \mathbb{R}^{p \times p}$ is simply defined as:

  $$\mathbf{R}_{i,j} = \sigma^o. \tag{61}$$

In order to illustrate the differences between the different cases we have mentioned, we applied a multi-incremental multi-resolution scheme using the three methods as well as different interpolators. The spectral decomposition is an orthogonal operator, which can be associated with a zero padding operator to build a transitive interpolator. It should be noted that usual grid-points interpolators like (bi-)linear and nearest-neighbor interpolators are not transitive in general. An important exception is the (bi-)linear interpolator which is transitive when coarse grid points are colocated with fine grid points.

5.2 Results

NB:Show the quadratic cost function (obs and background) + xgstd-xgth (+ plots for transitivity ?).

To illustrate the differences between the different cases that we have mentionned, we have choosen to use a spectral interpolation as a transitive interpolator, and we also have tested bilinear interpolation which is transitive if the grid points of different iterations are colocated. We arbitrarily have taken the values of $L_b = 0.1$, $\sigma^o = 0.1$. According to the convergence of our algorithm, we decided to realize 6 outer loops of 3 inner loops. The change of resolution has been taken as follows: $\mathcal{R}_1 = 21 \times 21$, $\mathcal{R}_2 = 31 \times 31$, $\mathcal{R}_3 = 41 \times 41$, $\mathcal{R}_4 = 51 \times 51$, $\mathcal{R}_5 = 81 \times 81$, $\mathcal{R}_6 = 101 \times 101$.

As expected, in the first case where transitive a interpolator is used, with a projective $\mathbf{B}$ matrix family, all the methods give the same results (not shown), and the two preconditionners are equivalent at machine precision.

Figures (**??**) and (**??**) respectively show the observational part of the quadratic cost function obtained with the complete (resp. simplified) versions of the "standard" and "theoretical" methods with both preconditionners in the case of an interpolator with right inverse and a "non projective $\mathbf{B}$ matrix family (corresponding to the results summarized in Table (2)). As expected, all the "complete" methods give the same results but the two preconditionners are not equivalent.

idem with simplified. except in the case of the simplified theoretical method.

The fact that the cost function shows small peaks at each beginning of outer loop is due to the relinearization of the problem around the new guess as well as the change of resolution, making the problem different from the one that have been solved at the previous outer loop.

The Tables (1) and (2) summerize the differences or equivalence between the three methods and the two preconditionners in the case of an interpolator with right inverse and a "non projective" $\mathbf{B}$ matrix family (Table (1)), and in the case of an interpolator without right inverse and a "non projective" $\mathbf{B}$ matrix family (Table (2)).

			Full $\mathbf{B}$					Square-root $\mathbf{B}$				
			Theoretical		Standard		Consistent	Theoretical		Standard		Consistent
			Comp.	Simp.	Comp.	Simp.		Comp.	Simp.	Comp.	Simp.	
Full $\mathbf{B}$	Theoretical	Comp.	✓	✗	✓	✗	✓	✗	✗	✗	✗	✗
		Simp.	✗	✓	✗	✗	✗	✗	✓	✗	✗	✗
	Standard	Comp.	✓	✗	✓	✗	✓	✗	✗	✗	✗	✗
		Simp.	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗
	Aternative		✓	✗	✓	✗	✓	✗	✗	✗	✗	✗
Square-root $\mathbf{B}$	Theoretical	Comp.	✗	✗	✗	✗	✗	✓	✗	✓	✗	✓
		Simp.	✗	✓	✗	✗	✗	✗	✓	✗	✗	✗
	Standard	Comp.	✗	✗	✗	✗	✗	✓	✗	✓	✗	✓
		Simp.	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗
	Aternative		✗	✗	✗	✗	✗	✓	✗	✓	✗	✓

Table 1. summary of equivalence between the three methods and the two preconditionners in the case of an interpolator with right inverse and a "non projective" $\mathbf{B}$ matrix family.

			Full $\mathbf{B}$					Square-root $\mathbf{B}$				
			Theoretical		Standard		Consistent	Theoretical		Standard		Consistent
			Comp.	Simp.	Comp.	Simp.		Comp.	Simp.	Comp.	Simp.	
Full $\mathbf{B}$	Theoretical	Comp.	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗
		Simp.	✗	✓	✗	✗	✗	✗	✓	✗	✗	✗
	Standard	Comp.	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗
		Simp.	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗
	Aternative		✗	✗	✗	✗	✓	✗	✗	✗	✗	✗
Square-root $\mathbf{B}$	Theoretical	Comp.	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗
		Simp.	✗	✓	✗	✗	✗	✗	✓	✗	✗	✗
	Standard	Comp.	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗
		Simp.	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗
	Aternative		✗	✗	✗	✗	✗	✗	✗	✗	✗	✓

Table 2. summary of equivalence between the three methods and the two preconditionners in the case of an interpolator without right inverse and a "non projective" $\mathbf{B}$ matrix family where $K > 2$ (the case where $K = 2$ gives the same results as in Table (2)).

6 Conclusions

Multi-incremental multi-resolution scheme is a powerfull way to improve computational efficiency of DA problems solving and might be more and more used in operational DA systems in a near future. In such an approach, we have shown that the question of the guess consistency is crucial since the results can be very different from a method to another according to the way the guess is calculated. We have discussed in detail this question and have shown that the choice of the interpolation operator used to realize the change of resolution at the outer loop level was very important. We have defined a class of interpolators called "transitive" interpolators that satisfy certain conditions ensuring the guess consistency. A subcase of transitive interpolators have been discussed concerning the $\mathbf{B}$ matrix family, introducing the "projective" $\mathbf{B}$ matrices for which low resolution members can be obtained from the high resolution ones using transitive interpolators. In this case, a simplified expression can be derived for the analysis increment and only the previous iteration is needed instead of all the previous ones.

Through a toy DA problem, we have illustrated that in the "standard" method currently used in operational DA systems, the way the guess is used in the minimization is inconsistent and could give results that are very different from the theoretical ones. We introduced a new "consistent" method to compute the guess based on the idea of reversing the order of calculations. This method have two strong advantages: first, it ensures the guess consistency, and second, it requires much more simpler calculations. We have shown the ability of this method to reproduce the results of the "standard" one in the case where ().

We have also discussed the equivalence of two well known preconditionners in a multi-incremental multi-resolution context and have shown that this equivalence does not necessarily hold. It is maintained if transitive interpolators are used and the $\mathbf{B}$ matrix family is "projective". If these conditions are not satisfied, the differences in the cost functions obtained with the two preconditionners can be huge and grow with the outer iterations.

References

REFERENCE 1

REFERENCE 2