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.

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: $\mathcal{H}(\mathbf{x}) \approx \mathcal{H}(\mathbf{x}_k^g) + \mathbf{H}_k \delta\mathbf{x}_k$ for $\mathbf{x} \approx \mathbf{x}_k^g$, 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: $H_{k,ij} = \frac{\partial \mathcal{H}_i}{\partial x_j}\Big|_{\mathbf{x}=\mathbf{x}_k^g}$. Instead of minimizing the full cost function $\mathcal{J}(\mathbf{x})$, it is now possible to minimize successive quadratic approximations around successive guess states:

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

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 $\delta\mathbf{x}_k^a$:

$$\begin{aligned}
&\mathbf{B}^{-1}\left(\delta\mathbf{x}_k^a - \delta\mathbf{x}_k^b\right) - \mathbf{H}_k^{\mathrm{T}}\mathbf{R}^{-1}\left(\mathbf{d}_k - \mathbf{H}_k\delta\mathbf{x}_k^a\right) = 0\\
\Leftrightarrow\ &\left(\mathbf{B}^{-1} + \mathbf{H}_k^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{H}_k\right)\delta\mathbf{x}_k^a = \mathbf{B}^{-1}\delta\mathbf{x}_k^b + \mathbf{H}_k^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{d}_k\\
\Leftrightarrow\ &\boxed{\mathbf{A}_k^{\mathbf{x}}\delta\mathbf{x}_k^a = \mathbf{b}_k^{\mathbf{x}}}
\end{aligned} \tag{3}$$

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

$$\mathbf{A}_k^{\mathbf{x}} = \mathbf{B}^{-1} + \mathbf{H}_k^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{H}_k, \tag{4}$$

$$\mathbf{b}_k^{\mathbf{x}} = \mathbf{B}^{-1}\delta\mathbf{x}_k^b + \mathbf{H}_k^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{d}_k. \tag{5}$$

The problem can now be solved using Gauss-Newton algorithm. It is very common to use the background state as a guess for the first iteration $k = 1$ ($\mathbf{x}_1^g = \mathbf{x}^b$), and 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:

$$\begin{aligned}
\delta\mathbf{x}_k^b &= \mathbf{x}^b - \mathbf{x}_k^g,\\
&= \mathbf{x}^b - \mathbf{x}_{k-1}^a,\\
&= \mathbf{x}^b - \left(\mathbf{x}_{k-1}^g + \delta\mathbf{x}_{k-1}^a\right),\\
&= \delta\mathbf{x}_{k-1}^b - \delta\mathbf{x}_{k-1}^a,
\end{aligned} \tag{6}$$

which can be combined recursively to yield:

$$\delta\mathbf{x}_k^b = -\sum_{i=1}^{k-1}\delta\mathbf{x}_i^a. \tag{7}$$

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 variable $\delta\mathbf{v}_k = \mathbf{U}^{\mathrm{T}}\mathbf{B}^{-1}\delta\mathbf{x}_k$, so that the linear system (3) can now be written as $\mathbf{A}_k^{\mathbf{v}}\delta\mathbf{v}_k^a = \mathbf{b}_k^{\mathbf{v}}$ with:

$$\mathbf{A}_k^{\mathbf{v}} = \mathbf{I}_m + \mathbf{U}^{\mathrm{T}}\mathbf{H}_k^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{H}_k\mathbf{U}, \tag{8}$$

$$\mathbf{b}_k^{\mathbf{v}} = \delta\mathbf{v}_k^b + \mathbf{U}^{\mathrm{T}}\mathbf{H}_k^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{d}_k. \tag{9}$$

Using these techniques 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 $\mathbf{b}_k^{\mathbf{v}}$: $\mathbf{U}^{\mathrm{T}}\mathbf{B}^{-1}$ can be applied on both side of equation (7), leading to:

$$\delta\mathbf{v}_k^b = -\sum_{i=1}^{k-1}\delta\mathbf{v}_i^a, \tag{10}$$

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

2.3 Changing B at outer loop level

In special cases, the background error covariance matrix can be updated between outer iterations defining $\mathbf{B}_k$, 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 (10) is valid and one can obtain the background increment as follows:

$$\begin{aligned}
\delta\mathbf{v}_k^b &= -\mathbf{U}_k^{\mathrm{T}}\mathbf{B}_k^{-1}\sum_{i=1}^{k-1}\delta\mathbf{x}_i^a, \\
&= -\sum_{i=1}^{k-1}\mathbf{U}_k^{\mathrm{T}}\mathbf{B}_k^{-1}\mathbf{U}_i\delta\mathbf{v}_i^a.
\end{aligned} \tag{11}$$

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 (10) cannot be used consistently.

3 Guess consistency

3.1 Theoretical method

3.2 Standard method

3.2.1 Simplified standard method

3.2.2 Corrected standard method

3.3 Consitent method

4 Methodology and results

Show the quadratc cost function (obs and background) + tables

5 Conclusions

TEXT

Code availability. TEXT

Data availability. TEXT

Code and data availability. TEXT

Sample availability. TEXT

Video supplement. TEXT

Appendix A: Equivalence between preconditionners

Another precondi technique consists in defining a new variable $\delta\overline{\mathbf{x}}_k = \mathbf{B}_k^{-1}\delta\mathbf{x}_k$, where $\delta\mathbf{x}_k$ is the state model increment of iteration k of a Gauss-Newton algorithm, and $\mathbf{B_k}$ is the model error covariance matrix, so that the linear system to solve can be written as $\mathbf{A}_k^{\overline{\mathbf{x}}} \delta\overline{\mathbf{x}}_k^a = \mathbf{b}_k^{\overline{\mathbf{x}}}$, $\delta\overline{\mathbf{x}}_k^a$ being the preconditionned analysis increment, and with $\mathbf{A}_k^{\overline{\mathbf{x}}} = \mathbf{I}_n + \mathbf{H}_k^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{H}_k\mathbf{B}_k$ and the right hand side $\mathbf{b}_k^{\overline{\mathbf{x}}} = \delta\overline{\mathbf{x}}_k^b + \mathbf{H}_k^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{d}_k$, where $\mathbf{R}_k$ is the observation error covariance matrix, $\mathbf{H}_k$ is the observation operator linearized around a guess state, and $\mathbf{d_k}$ is the innovation vector **??**.

$$\text{and} \quad \delta\mathbf{v}_k^b = -\sum_{i=1}^{k-1} \delta\mathbf{v}_i^a. \tag{A1}$$

With the full $\mathbf{B}$ preconditioning, $\mathbf{B}^{-1}$ can be applied on both side of equation (7):

$$\boxed{\delta\overline{\mathbf{x}}_k^b = -\sum_{i=1}^{k-1} \delta\overline{\mathbf{x}}_i^a} \tag{A2}$$

With the full $\mathbf{B}$ preconditioning:

$$\begin{aligned} \delta\overline{\mathbf{x}}_k^b &= -\mathbf{B}_k^{-1}\sum_{i=1}^{k-1} \delta\mathbf{x}_i^a \\ &= -\sum_{i=1}^{k-1} \mathbf{B}_k^{-1}\mathbf{B}_i \delta\overline{\mathbf{x}}_i^a \end{aligned} \tag{A3}$$

If $\mathbf{B}_k^{-1}\mathbf{B}_i\delta\overline{\mathbf{x}}_i^a \neq \delta\overline{\mathbf{x}}_i^a$, equation (A2) cannot be used consistently.

$$\delta\overline{\mathbf{x}}_k^b = -\mathbf{B}_k^{-1}\sum_{i=1}^{k-1} \delta\mathbf{x}_i^a = -\sum_{i=1}^{k-1} \mathbf{B}_k^{-1}\mathbf{B}_i \delta\overline{\mathbf{x}}_i^a, \tag{A4}$$

Author contributions. TEXT

Competing interests. TEXT

Disclaimer. TEXT

Acknowledgements. TEXT

References

REFERENCE 1

REFERENCE 2