Multi-incremental multi-resolution variational method: practical constraints

Benjamin Ménétrier

Last update: November 9, 2020

Contents

1 Problem linearization
1.1 Full cost function
1.2 Operators linearization
1.3 Quadratic cost function
1.4 Linear system

2 Iterative solvers and preconditionings
2.1 Full $\mathbf{B}$ preconditioning
2.2 Square-root $\mathbf{B}$ preconditioning
2.3 Equivalence condition for preconditioners

3 Practical computations
3.1 Getting rid of $\mathbf{B}^{-1}$
3.2 Changing the resolution
3.3 General requirements

4 Guess consistency
4.1 Theoretical method
4.2 Standard method
4.3 Alternative method
4.4 Graphical summary

1 Problem linearization

1.1 Full cost function

The full cost function, nonquadratic, is defined as:

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

where:

- $\mathbf{x} \in \mathbb{R}^n$ is the state in model space,
- $\mathbf{x}^b \in \mathbb{R}^n$ is the background state,
- $\mathbf{R} \in \mathbb{B}^{n\times n}$ is the background error covariance matrix,
- $\mathbf{y}^o \in \mathbb{R}^p$ is the observation vector,
- $\mathbf{R} \in \mathbb{R}^{p\times p}$ is the observation error covariance matrix,
- $\mathcal{H} : \mathbb{R}^n \rightarrow \mathbb{R}^p$ is the observation operator, nonlinear.

1.2 Operators linearization

The guess state $\mathbf{x}^g_k \in \mathbb{R}^n$ is introduced to define the increment:

$$\delta\mathbf{x}_k = \mathbf{x} - \mathbf{x}^g_k \tag{2}$$

and to linearize the observation operator for $\mathbf{x} \approx \mathbf{x}^g_k$:

$$\mathcal{H}(\mathbf{x}) \approx \mathcal{H}(\mathbf{x}^g_k) + \mathbf{H}_k\delta\mathbf{x}_k \tag{3}$$

where $\mathbf{H}_k \in \mathbb{R}^{p\times m}$ is the observation operator linearized around $\mathbf{x}^g_k$:

$$H_{k,ij} = \left.\frac{\partial\mathcal{H}_i}{\partial x_j}\right|_{\mathbf{x}=\mathbf{x}^g_k} \tag{4}$$

1.3 Quadratic cost function

Instead of minimizing the full cost function $\mathcal{J}(\mathbf{x})$, we minimize successive quadratic approximations around the guess $\mathbf{x}^g_k$, which can be written as:

$$J\left(\delta\mathbf{x}_k\right) = \frac{1}{2}\left(\delta\mathbf{x}_k-\delta\mathbf{x}^b_k\right)^{\mathrm{T}}\mathbf{B}^{-1}\left(\delta\mathbf{x}_k-\delta\mathbf{x}^b_k\right) + \frac{1}{2}\left(\mathbf{d}_k-\mathbf{H}_k\delta\mathbf{x}_k\right)^{\mathrm{T}}\mathbf{R}^{-1}\left(\mathbf{d}_k-\mathbf{H}_k\delta\mathbf{x}_k\right) \tag{5}$$

where:

- $\delta\mathbf{x}^b_k \in \mathbb{R}^n$ is the background increment:

$$\delta\mathbf{x}^b_k = \mathbf{x}^b - \mathbf{x}^g_k \tag{6}$$

- $\mathbf{d}_k \in \mathbb{R}^p$ is the innovation vector:

$$\mathbf{d}_k = \mathbf{y}^o - \mathcal{H}(\mathbf{x}^g_k) \tag{7}$$

1.4 Linear system

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

$$\begin{aligned}
& \mathbf{B}^{-1}\left(\delta \mathbf{x}_k^a - \delta \mathbf{x}_k^b\right) - \mathbf{H}_k^{\mathrm{T}}\mathbf{R}^{-1}\left(\mathbf{d}_k - \mathbf{H}_k \delta \mathbf{x}_k^a\right) = 0 \\
\Leftrightarrow\ & \left(\mathbf{B}^{-1} + \mathbf{H}_k^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{H}_k\right)\delta \mathbf{x}_k^a = \mathbf{B}^{-1}\delta \mathbf{x}_k^b + \mathbf{H}_k^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{d}_k \\
\Leftrightarrow\ & \boxed{\mathbf{A}_k^{\mathbf{x}}\delta \mathbf{x}_k^a = \mathbf{b}_k^{\mathbf{x}}} \qquad (8)
\end{aligned}$$

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

$$\mathbf{A}_k^{\mathbf{x}} = \mathbf{B}^{-1} + \mathbf{H}_k^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{H}_k \qquad (9)$$

$$\mathbf{b}_k^{\mathbf{x}} = \mathbf{B}^{-1}\delta \mathbf{x}_k^b + \mathbf{H}_k^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{d}_k \qquad (10)$$

2 Iterative solvers and preconditionings

For high-dimensional problems, linear system (8) can be solved with iterative methods, for instance the Lanczos method. Preconditioners are useful to accelerate the convergence. This section is based on **?**.

2.1 Full B preconditioning

A new variable $\delta \overline{\mathbf{x}}_k \in \mathbb{R}^n$ is defined as:

$$\delta \overline{\mathbf{x}}_k = \mathbf{B}_k^{-1}\delta \mathbf{x}_k \ \Leftrightarrow\ \delta \mathbf{x}_k = \mathbf{B}_k \delta \overline{\mathbf{x}}_k \qquad (11)$$

Linear system (8) is transformed into:

$$\boxed{\mathbf{A}_k^{\overline{\mathbf{x}}}\delta \overline{\mathbf{x}}_k^a = \mathbf{b}_k^{\overline{\mathbf{x}}}} \qquad (12)$$

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

$$\mathbf{A}_k^{\overline{\mathbf{x}}} = \mathbf{I}_n + \mathbf{H}_k^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{H}_k\mathbf{B}_k \qquad (13)$$

$$\mathbf{b}_k^{\overline{\mathbf{x}}} = \delta \overline{\mathbf{x}}_k^b + \mathbf{H}_k^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{d}_k \qquad (14)$$

The PLanczosIF method with a preconditioner $\mathbf{P}_k = \mathbf{B}\mathbf{C}_k$, where $\mathbf{C}_k \in \mathbb{R}^{n\times n}$ and $\mathbf{P}_k \in \mathbb{R}^{n\times n}$ is detailed in algorithm 1.

A common (but not unique) way to define the preconditioner $\mathbf{P}_k = \mathbf{B}\mathbf{C}_k$ consists in using the spectral Limited Memory Preconditioner (LMP) approximated from the Ritz pairs:

- For the first outer iteration ($k = 1$): $\mathbf{C}_1 = \mathbf{I}_n$.

- For subsequent outer iteration ($k > 1$):

$$\mathbf{C}_{k+1} = \mathbf{C}_k + \overline{\mathbf{V}}_k \left(\mathbf{\Lambda}_k^{-1} - \mathbf{I}_{I_k}\right) \mathbf{V}_k^{\mathrm{T}} \tag{15}$$

  where

$$\overline{\mathbf{V}}_k = \mathbf{C}_k \underline{\overline{\mathbf{V}}}_k \tag{16}$$

$$\mathbf{V}_k = \mathbf{B}\overline{\mathbf{V}}_k \tag{17}$$

It should be noted that the Ritz vectors are orthogonal with respect to the $\mathbf{P}_k$-inner product:

$$\underline{\overline{\mathbf{V}}}_k^{\mathrm{T}} \mathbf{P}_k \underline{\overline{\mathbf{V}}}_k = \mathbf{I}_{I_k} \tag{18}$$

As a consequence, the inverse of $\mathbf{C}_{k+1}$ can be easily computed from the Woodbury matrix identity:

$$\begin{aligned}
\mathbf{C}_{k+1}^{-1} &= \mathbf{C}_k^{-1} - \mathbf{C}_k^{-1}\overline{\mathbf{V}}_k \left(\left(\mathbf{\Lambda}_k^{-1} - \mathbf{I}_{I_k}\right)^{-1} + \mathbf{V}_k^{\mathrm{T}}\mathbf{C}_k^{-1}\overline{\mathbf{V}}_k\right)^{-1} \mathbf{V}_k^{\mathrm{T}}\mathbf{C}_k^{-1} \\
&= \left(\mathbf{I}_n - \underline{\overline{\mathbf{V}}}_k \left(\left(\mathbf{\Lambda}_k^{-1} - \mathbf{I}_{I_k}\right)^{-1} + \mathbf{I}_{I_k}\right)^{-1} \mathbf{V}_k^{\mathrm{T}}\right) \mathbf{C}_k^{-1} \\
&= \left(\mathbf{I}_n + \underline{\overline{\mathbf{V}}}_k \left(\mathbf{\Lambda}_k - \mathbf{I}_{I_k}\right) \mathbf{V}_k^{\mathrm{T}}\right) \mathbf{C}_k^{-1}
\end{aligned} \tag{19}$$

Algorithm 1 PLanczosIF algorithm with a preconditioner $\mathbf{P}_k = \mathbf{B}\mathbf{C}_k$

Set the number of iterations: I_k

Objects sizes:
α_i, $\beta_i \in \mathbb{R}$ for $0 \leq i \leq I_k$
$\mathbf{q}_i$, $\mathbf{r}_i$, $\bar{\mathbf{t}}_i$, $\mathbf{t}_i$, $\mathbf{v}_i$, $\mathbf{w}_i$, $\bar{\mathbf{z}}_i$, $\mathbf{z}_i \in \mathbb{R}^n$ for $0 \leq i \leq I_k$
$\mathbf{e_1} = [1, 0, \ldots, 0]^{\mathrm{T}} \in \mathbb{R}^{I_k}$
$\mathbf{T}$, $\mathbf{\Theta}$, $\mathbf{Y}$, $\mathbf{\Lambda}_k \in \mathbb{R}^{I_k \times I_k}$
$\underline{\mathbf{V}}$, $\overline{\underline{\mathbf{V}}}_k \in \mathbb{R}^{n \times I_k}$

Initialization:
$\mathbf{v}_0 = 0$
$\mathbf{r}_0 = \delta\overline{\mathbf{x}}^b_k + \mathbf{K}^{\mathrm{T}}_k\mathbf{H}^{\mathrm{T}}_k\mathbf{R}^{-1}\mathbf{d}_k$
$\bar{\mathbf{t}}_0 = \mathbf{C}_k\mathbf{r}_0$
$\mathbf{t}_0 = \mathbf{B}\bar{\mathbf{t}}_0$
$\beta_0 = \sqrt{\mathbf{r}^{\mathrm{T}}_0\mathbf{t}_0}$
$\mathbf{v}_1 = \mathbf{r}_0/\beta_0$
$\bar{\mathbf{z}}_1 = \bar{\mathbf{t}}_0/\beta_0$
$\mathbf{z}_1 = \mathbf{t}_0/\beta_0$
$\beta_1 = 0$

for $1 \leq i \leq I_k$ **do**
    Store the Lanczos vector $\mathbf{v}_i$ as the i^{th} column of $\underline{\mathbf{V}}$
    Update scalars and vectors:
    $\mathbf{q}_i = \bar{\mathbf{z}}_i + \mathbf{K}^{\mathrm{T}}_k\mathbf{H}^{\mathrm{T}}_k\mathbf{R}^{-1}\mathbf{H}_k\mathbf{K}_k\mathbf{z}_i - \beta_i\mathbf{v}_{i-1}$
    $\alpha_i = \mathbf{q}^{\mathrm{T}}_i\mathbf{z}_i$
    $\mathbf{w}_i = \mathbf{q}_i - \alpha_i\mathbf{v}_i$
    $\bar{\mathbf{t}}_i = \mathbf{C}_k\mathbf{w}_i$
    $\mathbf{t}_i = \mathbf{B}\bar{\mathbf{t}}_i$
    $\beta_{i+1} = \sqrt{\mathbf{w}_i\mathbf{z}_i}$
    $\mathbf{v}_{i+1} = \mathbf{w}_i/\beta_{i+1}$
    $\bar{\mathbf{z}}_{i+1} = \bar{\mathbf{t}}_i/\beta_{i+1}$
    $\mathbf{z}_{i+1} = \mathbf{t}_i/\beta_{i+1}$
    Fill the tridiagonal matrix $\mathbf{T}$:
    $T_{ii} = \alpha_i$
    if $i > 1$ **then**
        $T_{(i-1)i} = T_{i(i-1)} = \beta_i$
    end if
end for
Compute $(\mathbf{\Theta}, \mathbf{Y})$, the eigendecomposition of $\mathbf{T} = \mathbf{Y}\mathbf{\Theta}\mathbf{Y}^{\mathrm{T}}$
Compute the analysis increment: $\delta\mathbf{x}^a_k = \mathbf{P}_k\underline{\mathbf{V}}\mathbf{Y}\mathbf{\Theta}^{-1}\mathbf{Y}^{\mathrm{T}}\left(\beta_0\mathbf{e}_1\right)$
Store the Ritz pairs $\left(\mathbf{\Lambda}_k, \overline{\underline{\mathbf{V}}}_k\right) = (\mathbf{\Theta}, \underline{\mathbf{V}}\mathbf{Y})$

2.2 Square-root $\mathbf{B}$ preconditioning

Since $\mathbf{B}$ is positive definite, there is an infinity of square-roots $\mathbf{U} \in \mathbb{R}^{n \times m}$ with $m \geq n$ verifying:

$$\mathbf{B} = \mathbf{U}\mathbf{U}^{\mathrm{T}} \tag{20}$$

A new variable $\delta\mathbf{v}_k \in \mathbb{R}^m$ is defined as:

$$\delta\mathbf{v}_k = \mathbf{U}^{\mathrm{T}}\mathbf{B}^{-1}\delta\mathbf{x}_k \;\Rightarrow\; \delta\mathbf{x}_k = \mathbf{U}\delta\mathbf{v}_k \tag{21}$$

Linear system (8) is transformed into:

$$\boxed{\mathbf{A}^{\mathbf{v}}_k\delta\mathbf{v}^a_k = \mathbf{b}^{\mathbf{v}}_k} \tag{22}$$

with $\mathbf{A}^{\mathbf{v}}_k \in \mathbb{R}^{m \times m}$ and $\mathbf{b}^{\mathbf{v}}_k \in \mathbb{R}^m$ defined as:

$$\mathbf{A}^{\mathbf{v}}_k = \mathbf{I}_m + \mathbf{U}^{\mathrm{T}}\mathbf{H}^{\mathrm{T}}_k\mathbf{R}^{-1}\mathbf{H}_k\mathbf{U} \tag{23}$$

$$\mathbf{b}^{\mathbf{v}}_k = \delta\mathbf{v}^b_k + \mathbf{U}^{\mathrm{T}}\mathbf{H}^{\mathrm{T}}_k\mathbf{R}^{-1}\mathbf{d}_k \tag{24}$$

The Lanczos method with a preconditioner $\mathbf{Q}_k = \mathbf{Q}^{1/2}_k\mathbf{Q}^{\mathrm{T}/2}_k$, where $\mathbf{Q}^{1/2}_k \in \mathbb{R}^{m \times m}$, is detailed in algorithm 2.

A common (but not unique) way to define the preconditioner $\mathbf{Q}_k$ consists in using the spectral Limited Memory Preconditioner (LMP) approximated from the Ritz pairs:

- For the first outer iteration ($k = 1$): $\mathbf{Q}^{1/2}_1 = \mathbf{I}_m$.
- For subsequent outer iteration ($k > 1$):

$$\mathbf{Q}^{1/2}_k = \mathbf{Q}^{1/2}_{k-1}\mathbf{F}_k \tag{25}$$

where $\mathbf{F}_k \in \mathbb{R}^{m \times m}$ is defined from the Ritz pairs of the previous outer iteration:

$$\mathbf{F}_{k+1} = \mathbf{I}_m + \underline{\widetilde{\mathbf{V}}}_k\left(\mathbf{\Lambda}^{-1/2}_k - \mathbf{I}_{I_k}\right)\underline{\widetilde{\mathbf{V}}}^{\mathrm{T}}_k \tag{26}$$

It should be noted that the Ritz vectors are orthogonal with respect to the canonical inner product:

$$\underline{\widetilde{\mathbf{V}}}^{\mathrm{T}}_k\underline{\widetilde{\mathbf{V}}}_k = \mathbf{I}_{I_k} \tag{27}$$

As a consequence, the inverse of $\mathbf{Q}^{1/2}_k$ can be easily computed using the Woodbury matrix identity:

$$\left(\mathbf{Q}^{1/2}_k\right)^{-1} = \mathbf{F}^{-1}_k\left(\mathbf{Q}^{1/2}_{k-1}\right)^{-1} \tag{28}$$

where

$$\begin{aligned}
\mathbf{F}^{-1}_{k+1} &= \mathbf{I}_m - \underline{\widetilde{\mathbf{V}}}_k\left(\left(\mathbf{\Lambda}^{-1/2}_k - \mathbf{I}_{I_k}\right)^{-1} + \underline{\widetilde{\mathbf{V}}}^{\mathrm{T}}_k\underline{\widetilde{\mathbf{V}}}_k\right)^{-1}\underline{\widetilde{\mathbf{V}}}^{\mathrm{T}}_k \\
&= \mathbf{I}_m - \underline{\widetilde{\mathbf{V}}}_k\left(\left(\mathbf{\Lambda}^{-1/2}_k - \mathbf{I}_{I_k}\right)^{-1} + \mathbf{I}_{I_k}\right)^{-1}\underline{\widetilde{\mathbf{V}}}^{\mathrm{T}}_k \\
&= \mathbf{I}_m + \underline{\widetilde{\mathbf{V}}}_k\left(\mathbf{\Lambda}^{1/2}_k - \mathbf{I}_{I_k}\right)\underline{\widetilde{\mathbf{V}}}^{\mathrm{T}}_k
\end{aligned} \tag{29}$$

Algorithm 2 Lanczos algorithm with a preconditioner $\mathbf{Q}_k$

Set the number of iterations: I_k

Objects sizes:
α_i, $\beta_i \in \mathbb{R}$ for $0 \leq i \leq I_k$
$\mathbf{q}_i$, $\mathbf{r}_i$, $\mathbf{v}_i$, $\mathbf{w}_i \in \mathbb{R}^m$ for $0 \leq i \leq I_k$
$\mathbf{e_1} = [1, 0, \dots, 0]^{\mathrm{T}} \in \mathbb{R}^{I_k}$
$\mathbf{T}$, $\boldsymbol{\Theta}$, $\mathbf{Y}$, $\boldsymbol{\Lambda}_k \in \mathbb{R}^{I_k \times I_k}$
$\underline{\mathbf{V}}$, $\widetilde{\underline{\mathbf{V}}}_k \in \mathbb{R}^{m \times I_k}$

Initialization:
$\mathbf{v}_0 = 0$
$\mathbf{r}_0 = \mathbf{Q}_k^{\mathrm{T}/2} \left(\delta \mathbf{v}_k^b + \mathbf{U}^{\mathrm{T}} \mathbf{H}_k^{\mathrm{T}} \mathbf{R}^{-1} \mathbf{d}_k \right)$
$\beta_0 = \|\mathbf{r}_0\|_2$
$\mathbf{v}_1 = \mathbf{r}_0 / \beta_0$
$\beta_1 = 0$

for $1 \leq i \leq I_k$ **do**
  Store the Lanczos vector $\mathbf{v}_i$ as the i^{th} column of $\underline{\mathbf{V}}$
  Update scalars and vectors:
  $\mathbf{q}_i = \mathbf{Q}_k^{\mathrm{T}/2} \left(\mathbf{I}_m + \mathbf{U}^{\mathrm{T}} \mathbf{H}_k^{\mathrm{T}} \mathbf{R}^{-1} \mathbf{H}_k \mathbf{U} \right) \mathbf{Q}_k^{1/2} \mathbf{v}_i - \beta_i \mathbf{v}_{i-1}$
  $\alpha_i = \mathbf{q}_i^{\mathrm{T}} \mathbf{v}_i$
  $\mathbf{w}_i = \mathbf{q}_i - \alpha_i \mathbf{v}_i$
  $\beta_{i+1} = \|\mathbf{w}_i\|_2$
  $\mathbf{v}_{i+1} = \mathbf{w}_i / \beta_{i+1}$
  Fill the tridiagonal matrix $\mathbf{T}$:
  $T_{ii} = \alpha_i$
  if $i > 1$ **then**
    $T_{(i-1)i} = T_{i(i-1)} = \beta_i$
  end if
end for

Compute $(\boldsymbol{\Theta}, \mathbf{Y})$, the eigendecomposition of $\mathbf{T} = \mathbf{Y} \boldsymbol{\Theta} \mathbf{Y}^{\mathrm{T}}$
Compute the analysis increment: $\delta \mathbf{v}_k^a = \mathbf{Q}_k^{1/2} \underline{\mathbf{V}} \mathbf{Y} \boldsymbol{\Theta}^{-1} \mathbf{Y}^{\mathrm{T}} \left(\beta_0 \mathbf{e}_1 \right)$
Store the Ritz pairs $\left(\boldsymbol{\Lambda}_k, \widetilde{\underline{\mathbf{V}}}_k \right) = (\boldsymbol{\Theta}, \underline{\mathbf{V}} \mathbf{Y})$

2.3 Equivalence condition for preconditioners

Both approaches are equivalent if the preconditioners are linked via:

$$\mathbf{P}_k^{1/2} = \mathbf{Q}_k^{1/2}\mathbf{U} \tag{30}$$

which is verified for the spectral preconditioner approximated with the Ritz pairs. Figure 1 summarize the relationships between the different quantities.

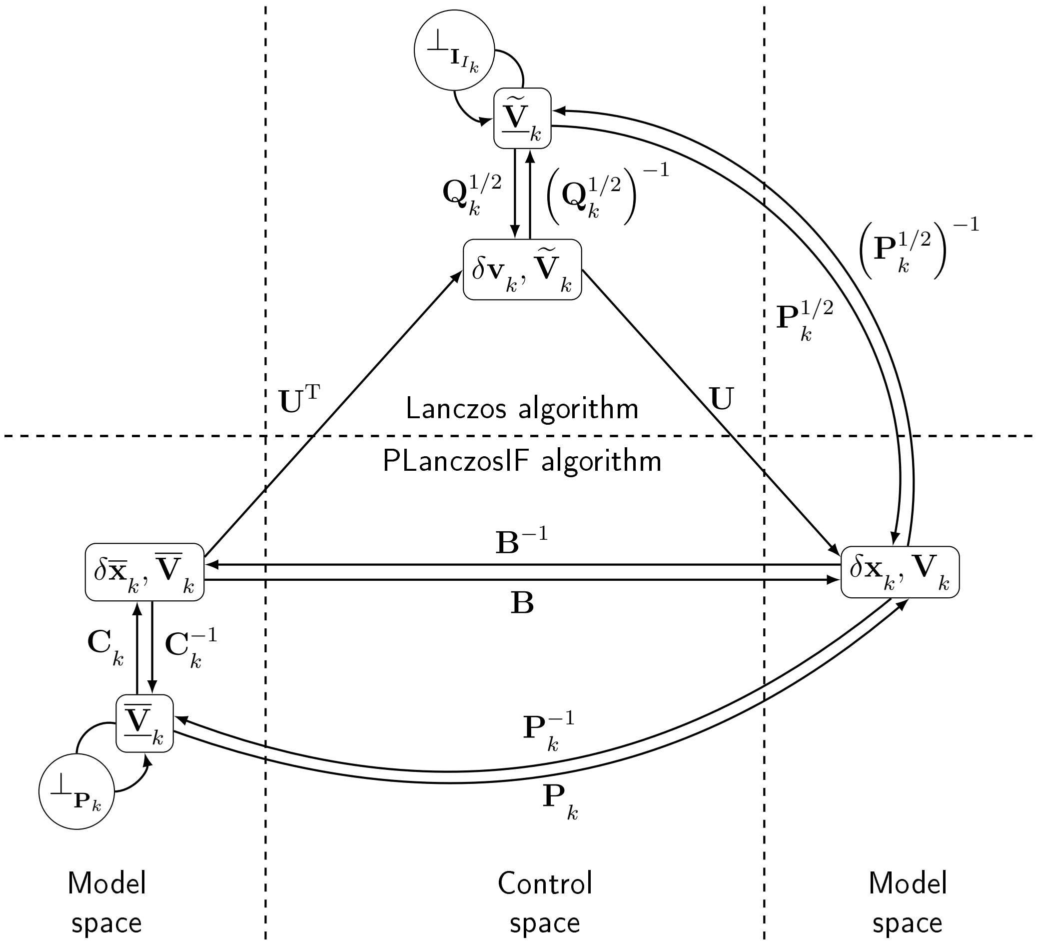

Figure 1: Map of spaces and links between them. Circles indicate orthogonality properties.

3 Practical computations

3.1 Getting rid of $\mathbf{B}^{-1}$

In practice, the inverse of the background error covariance matrix $\mathbf{B}^{-1}$ is not available, even if it is needed in general to compute the right-hand sides $\mathbf{b}^{\overline{\mathbf{x}}}_k$ and $\mathbf{b}^{\mathbf{v}}_k$. However, it is very usual to define the guess as follows:

- for $k = 1$: $\mathbf{x}^g_1 = \mathbf{x}^b$,
- for $k > 1$: $\mathbf{x}^g_k = \mathbf{x}^a_{k-1}$.

Thus:

- for $k = 1$:

$$\delta\mathbf{x}^b_1 = \mathbf{x}^g_1 - \mathbf{x}^b = 0 \tag{31}$$

- for $k > 1$:

$$\begin{aligned}
\delta\mathbf{x}^b_k &= \mathbf{x}^b - \mathbf{x}^g_k \\
&= \mathbf{x}^b - \mathbf{x}^a_{k-1} \\
&= \mathbf{x}^b - \left(\mathbf{x}^g_{k-1} + \delta\mathbf{x}^a_{k-1}\right) \\
&= \delta\mathbf{x}^b_{k-1} - \delta\mathbf{x}^a_{k-1}
\end{aligned} \tag{32}$$

which can be combined recursively to yield:

$$\delta\mathbf{x}^b_k = -\sum_{i=1}^{k-1} \delta\mathbf{x}^a_i \tag{33}$$

With the full $\mathbf{B}$ preconditioning, $\mathbf{B}^{-1}$ can be applied on both side of equation (33):

$$\boxed{\delta\overline{\mathbf{x}}^b_k = -\sum_{i=1}^{k-1} \delta\overline{\mathbf{x}}^a_i} \tag{34}$$

and with the square-root $\mathbf{B}$ preconditioning, $\mathbf{U}^\mathrm{T}\mathbf{B}^{-1}$ can be applied on both side of equation (33):

$$\boxed{\delta\mathbf{v}^b_k = -\sum_{i=1}^{k-1} \delta\mathbf{v}^a_i} \tag{35}$$

Equations (34) and (35) can be used to compute $\mathbf{b}^{\overline{\mathbf{x}}}_k$ and $\mathbf{b}^{\mathbf{v}}_k$ respectively, without requiring $\mathbf{B}^{-1}$.

3.2 Changing the resolution

For computational efficiency, it is common to start the optimization at a lower resolution, and to increase it at each outer iteration k. At resolution $\mathcal{R}_k$, the model space size is denoted n_k and the control space size m_k. It is assumed that the full resolution is obtained at the last iteration K.

Obviously, $\mathbf{B}$ now depends on iteration k. Hereafter, it is denoted $\mathbf{B}_k$, and its square-root is denoted $\mathbf{U}_k$.

We define two interpolators from resolution $\mathcal{R}_i$ to resolution $\mathcal{R}_k$:

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

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

- Transitivity:
  - For $n_i \leq n_j$ and $n_i \leq n_k$:
$$\mathbf{T}^{\mathbf{x}}_{j\rightarrow k}\mathbf{T}^{\mathbf{x}}_{i\rightarrow j} = \mathbf{T}^{\mathbf{x}}_{i\rightarrow k} \tag{36}$$
  - For $n_i \leq n_j \leq n_k$:
$$\mathbf{T}^{\mathbf{x}}_{j\rightarrow i}\mathbf{T}^{\mathbf{x}}_{k\rightarrow j} = \mathbf{T}^{\mathbf{x}}_{k\rightarrow i} \tag{37}$$
- Right-inverse: for $n_i \leq n_k$
$$\mathbf{T}^{\mathbf{x}}_{k\rightarrow i}\mathbf{T}^{\mathbf{x}}_{i\rightarrow k} = \mathbf{I}_{n_i} \tag{38}$$
$$\mathbf{T}^{\mathbf{x}}_{i\rightarrow k}\mathbf{T}^{\mathbf{x}}_{k\rightarrow i} \neq \mathbf{I}_{n_k} \tag{39}$$

and similarly for $\mathbf{T}^{\mathbf{v}}_{i\rightarrow k}$ in control space.

3.3 General requirements

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 $\mathcal{R}_k$:
$$\boxed{\mathbf{x}^b_k = \mathbf{T}^{\mathbf{x}}_{K\rightarrow k}\mathbf{x}^b} \tag{40}$$
- A full resolution guess denoted $\mathbf{x}^{g+}_k$ 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:
$$\boxed{\mathbf{x}^g_k = \mathbf{T}^{\mathbf{x}}_{K\rightarrow k}\mathbf{x}^{g+}_k} \tag{41}$$
- Only δ-quantities should be interpolated to higher resolution, and then possibly added to full quantities at high resolution.

4 Guess consistency

In the linear systems solved at each outer iteration, the guess appears in two main places:

1. explicitely as the linearization state for nonlinear operators and to compute the innovation,
2. implicitely in the first term of the right-hand side.

This section details how to define all occurences consistently.

4.1 Theoretical method

For the first iteration, the full resolution guess is the background state, also provided at full resolution:

$$\mathbf{x}_1^{g+} = \mathbf{x}^b \tag{42}$$

At the end of iteration k, an analysis increment $\delta \mathbf{x}_k^a$ is produced at resolution $\mathcal{R}_k$. In the previous section, the guess of iteration k was defined as the analysis of iteration $k-1$, which is not possible anymore since the resolution increases. Some interpolation of the output of the previous outer iteration is required to generate the analysis increment at full resolution $\delta \mathbf{x}_{k-1}^{a+}$, which updates the full resolution guess:

$$\mathbf{x}_k^{g+} = \mathbf{x}_{k-1}^{g+} + \delta \mathbf{x}_{k-1}^{a+} \tag{43}$$

How the analysis increment at full resolution $\delta \mathbf{x}_{k-1}^{a+}$ is obtained does not matter at this point. The first term of the right-hand side as:

$$\begin{aligned} \delta \overline{\mathbf{x}}_k^b &= \mathbf{B}_k^{-1} \delta \mathbf{x}_k^b \\ &= \mathbf{B}_k^{-1} \left(\mathbf{x}_k^b - \mathbf{x}_k^g \right) \end{aligned} \tag{44}$$

or

$$\begin{aligned} \delta \mathbf{v}_k^b &= \mathbf{U}_k^{\mathrm{T}} \mathbf{B}_k^{-1} \delta \mathbf{x}_k^b \\ &= \mathbf{U}_k^{\mathrm{T}} \mathbf{B}_k^{-1} \left(\mathbf{x}_k^b - \mathbf{x}_k^g \right) \end{aligned} \tag{45}$$

Here, both explicit and implicit guesses are consistent, thanks to the use of $\mathbf{B}^{-1}$.

4.2 Standard method

If $\mathbf{B}^{-1}$ is not available, the first term of the right-hand side is computed separately, using transformed versions of equations (34) and (35) with appropriate interpolations:

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

and

$$\boxed{\delta\underline{\mathbf{v}}^b_k = -\sum_{i=1}^{k-1} \mathbf{T}^{\mathbf{v}}_{i\to k}\delta\mathbf{v}^a_i} \tag{47}$$

Underlines emphasize the fact that these quantities are not necessarily equal to their exact counterparts of equations (44) and (45).

Merging equations (42) and (43), the guess at full resolution can be expressed as:

$$\mathbf{x}^{g+}_k = \mathbf{x}^b + \sum_{i=1}^{k-1} \delta\mathbf{x}^{a+}_i \tag{48}$$

which can be introduced into equations (44) and (45) to get background increments as functions of the analysis increment at full resolution $\delta\mathbf{x}^{a+}_i$:

$$\begin{aligned}\delta\overline{\mathbf{x}}^b_k &= \mathbf{B}^{-1}_k\left(\mathbf{x}^b_k - \mathbf{x}^g_k\right)\\ &= \mathbf{B}^{-1}_k\mathbf{T}^{\mathbf{x}}_{K\to k}\left(\mathbf{x}^b - \mathbf{x}^{g+}_k\right)\\ &= -\mathbf{B}^{-1}_k\mathbf{T}^{\mathbf{x}}_{K\to k}\sum_{i=1}^{k-1}\delta\mathbf{x}^{a+}_i\end{aligned} \tag{49}$$

and

$$\begin{aligned}\delta\mathbf{v}^b_k &= \mathbf{U}^{\mathrm{T}}_k\mathbf{B}^{-1}_k\left(\mathbf{x}^b_k - \mathbf{x}^g_k\right)\\ &= \mathbf{U}^{\mathrm{T}}_k\mathbf{B}^{-1}_k\mathbf{T}^{\mathbf{x}}_{K\to k}\left(\mathbf{x}^b - \mathbf{x}^{g+}_k\right)\\ &= -\mathbf{U}^{\mathrm{T}}_k\mathbf{B}^{-1}_k\mathbf{T}^{\mathbf{x}}_{K\to k}\sum_{i=1}^{k-1}\delta\mathbf{x}^{a+}_i\end{aligned} \tag{50}$$

Both explicit and implicit guesses in the standard method are consistent if $\delta\underline{\overline{\mathbf{x}}}^b_k = \delta\overline{\mathbf{x}}^b_k$ or $\delta\underline{\mathbf{v}}^b_k = \delta\mathbf{v}^b_k$:

- Comparing equations (46) and (49), $\delta\underline{\overline{\mathbf{x}}}^b_k = \delta\overline{\mathbf{x}}^b_k$ if:

  $$\mathbf{B}^{-1}_k\mathbf{T}^{\mathbf{x}}_{K\to k}\delta\mathbf{x}^{a+}_i = \mathbf{T}^{\mathbf{x}}_{i\to k}\delta\overline{\mathbf{x}}^a_i \tag{51}$$

  which can be simplified for transitive interpolators into:

  $$\boxed{\delta\mathbf{x}^{a+}_i = \mathbf{T}^{\mathbf{x}}_{k\to K}\mathbf{B}_k\mathbf{T}^{\mathbf{x}}_{i\to k}\delta\overline{\mathbf{x}}^a_i} \tag{52}$$

- Similarly in control space, comparing equations (47) and (50), $\delta\underline{\mathbf{v}}^b_k = \delta\mathbf{v}^b_k$ if and only if:

  $$\mathbf{U}^{\mathrm{T}}_k\mathbf{B}^{-1}_k\mathbf{T}^{\mathbf{x}}_{K\to k}\delta\mathbf{x}^{a+}_i = \mathbf{T}^{\mathbf{v}}_{i\to k}\delta\mathbf{v}^a_i \tag{53}$$

  which can be simplified for transitive interpolators into:

  $$\boxed{\delta\mathbf{x}^{a+}_i = \mathbf{T}^{\mathbf{x}}_{k\to K}\mathbf{U}_k\mathbf{T}^{\mathbf{v}}_{i\to k}\delta\mathbf{v}^a_i} \tag{54}$$

Other ways to obtain the analysis increment at full resolution $\delta\mathbf{x}^{a+}_i$ lead to an inconsistency between the explicit guess and implicit guesses.

4.3 Alternative method

Reversing the order of computations yields the same results as the consistent standard method, but with fewer and simpler computations. The first term of the right-hand side is computed first from equations (46) or (47). Then, the background increment is given by:

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

or

$$\delta \mathbf{x}^b_k = \mathbf{U}_k \delta \underline{\mathbf{v}}^b_k \tag{56}$$

Finally, the explicit guess is deduced at resolution $\mathcal{R}_k$ as:

$$\mathbf{x}^g_k = \mathbf{x}^b_k - \delta \mathbf{x}^b_k \tag{57}$$

and at full resolution as:

$$\mathbf{x}^{g+}_k = \mathbf{x}^b - \mathbf{T}^{\mathbf{x}}_{k \rightarrow K} \delta \mathbf{x}^b_k \tag{58}$$

4.4 Graphical summary

From left to right: theoretical method, standard method and alternative method. The potential inconsistency of the standard method (center) 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.