Multi-incremental method: nonlinearities and resolution change

Benjamin Ménétrier

Last update: July 3, 2019

Contents

1 Problem linearization **1**
1.1 Full cost function 1
1.2 Operators linearization 2
1.3 Quadratic cost function 2
1.4 Linear system 2

2 Iterative solvers and preconditionings **3**
2.1 Full $\mathbf{B}$ preconditioning 3
2.2 Square-root $\mathbf{B}$ preconditioning 4
2.3 Equivalence condition for preconditioners 7

3 Practical computations **8**
3.1 Getting rid of $\mathbf{B}^{-1}$ 8
3.2 Changing the resolution 9
3.3 Usual but inconsistent strategy 9
3.4 Consistent strategy 10
3.5 Equivalence condition on the background increment 11
3.6 Equivalence condition on Ritz vectors 11
3.7 Example of equivalence 12

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,

- $\mathbf{R} \in \mathbb{R}^{p\times p}$ is the observation error covariance matrix,
- $\mathcal{H} : \mathbb{R}^n \to \mathbb{R}^p$ is the observation operator, nonlinear,
- $\mathbf{y}^o \in \mathbb{R}^p$ is the observation vector.

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 full 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

For $\mathbf{x} \approx \mathbf{x}^g_k$, the quadratic approximation of $\mathcal{J}(\mathbf{x})$ 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 $\delta\mathbf{x}^a_k$:

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

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

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

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

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 Gurol (2013).

2.1 Full $\mathbf{B}$ preconditioning

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

$$\delta\overline{\mathbf{x}}_k = \mathbf{B}^{-1}\delta\mathbf{x}_k \;\Leftrightarrow\; \delta\mathbf{x}_k = \mathbf{B}\delta\overline{\mathbf{x}}_k \tag{11}$$

Linear system (8) is transformed into:

$$\mathbf{A}^{\overline{\mathbf{x}}}_k \delta\overline{\mathbf{x}}^a_k = \mathbf{b}^{\overline{\mathbf{x}}}_k \tag{12}$$

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

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

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

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 $\mathbf{P}_k = \mathbf{B}\mathbf{C}_k$ is to use 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 ($k > 1$):

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

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:

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

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

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

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 $m \geq n$, such as:

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

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

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

Linear system (8) is transformed into:

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

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 $\mathbf{Q}_k$ is to use 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 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 $0 \leq i \leq I_k$
$\mathbf{q}_i$, $\mathbf{r}_i$, $\overline{\mathbf{t}}_i$, $\mathbf{t}_i$, $\mathbf{v}_i$, $\mathbf{w}_i$, $\overline{\mathbf{z}}_i$, $\mathbf{z}_i \in \mathbb{R}^n$ for $0 \leq i \leq I_k$
$\mathbf{e_1} = [1, 0, \ldots, 0]^{\mathrm{T}} \in \mathbb{R}^{I_k}$
$\mathbf{T}$, $\mathbf{\Theta}$, $\mathbf{Y}$, $\mathbf{\Lambda}_k \in \mathbb{R}^{I_k \times I_k}$
$\underline{\mathbf{V}}$, $\underline{\overline{\mathbf{V}}}_k \in \mathbb{R}^{n \times I_k}$

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

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:
  $\mathbf{q}_i = \overline{\mathbf{z}}_i + \mathbf{K}_k^{\mathrm{T}}\mathbf{H}_k^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{H}_k\mathbf{K}_k\mathbf{z}_i - \beta_i\mathbf{v}_{i-1}$
  $\alpha_i = \mathbf{q}_i^{\mathrm{T}}\mathbf{z}_i$
  $\mathbf{w}_i = \mathbf{q}_i - \alpha_i\mathbf{v}_i$
  $\overline{\mathbf{t}}_i = \mathbf{C}_k\mathbf{w}_i$
  $\mathbf{t}_i = \mathbf{B}\overline{\mathbf{t}}_i$
  $\beta_{i+1} = \sqrt{\mathbf{w}_i\mathbf{z}_i}$
  $\mathbf{v}_{i+1} = \mathbf{w}_i/\beta_{i+1}$
  $\overline{\mathbf{z}}_{i+1} = \overline{\mathbf{t}}_i/\beta_{i+1}$
  $\mathbf{z}_{i+1} = \mathbf{t}_i/\beta_{i+1}$
  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: $\delta\mathbf{x}_k^a = \mathbf{P}_k\underline{\mathbf{V}}\mathbf{Y}\mathbf{\Theta}^{-1}\mathbf{Y}^{\mathrm{T}}\left(\beta_0\mathbf{e}_1\right)$
Store the Ritz pairs $\left(\mathbf{\Lambda}_k, \underline{\overline{\mathbf{V}}}_k\right) = (\mathbf{\Theta}, \underline{\mathbf{V}}\mathbf{Y})$

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 $0 \leq i \leq I_k$
$\mathbf{e_1} = [1, 0, \dots, 0]^{\mathrm{T}} \in \mathbb{R}^{I_k}$
$\mathbf{T}$, $\mathbf{\Theta}$, $\mathbf{Y}$, $\mathbf{\Lambda}_k \in \mathbb{R}^{I_k \times I_k}$
$\underline{\mathbf{V}}$, $\widetilde{\underline{\mathbf{V}}}_k \in \mathbb{R}^{m \times I_k}$

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

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.

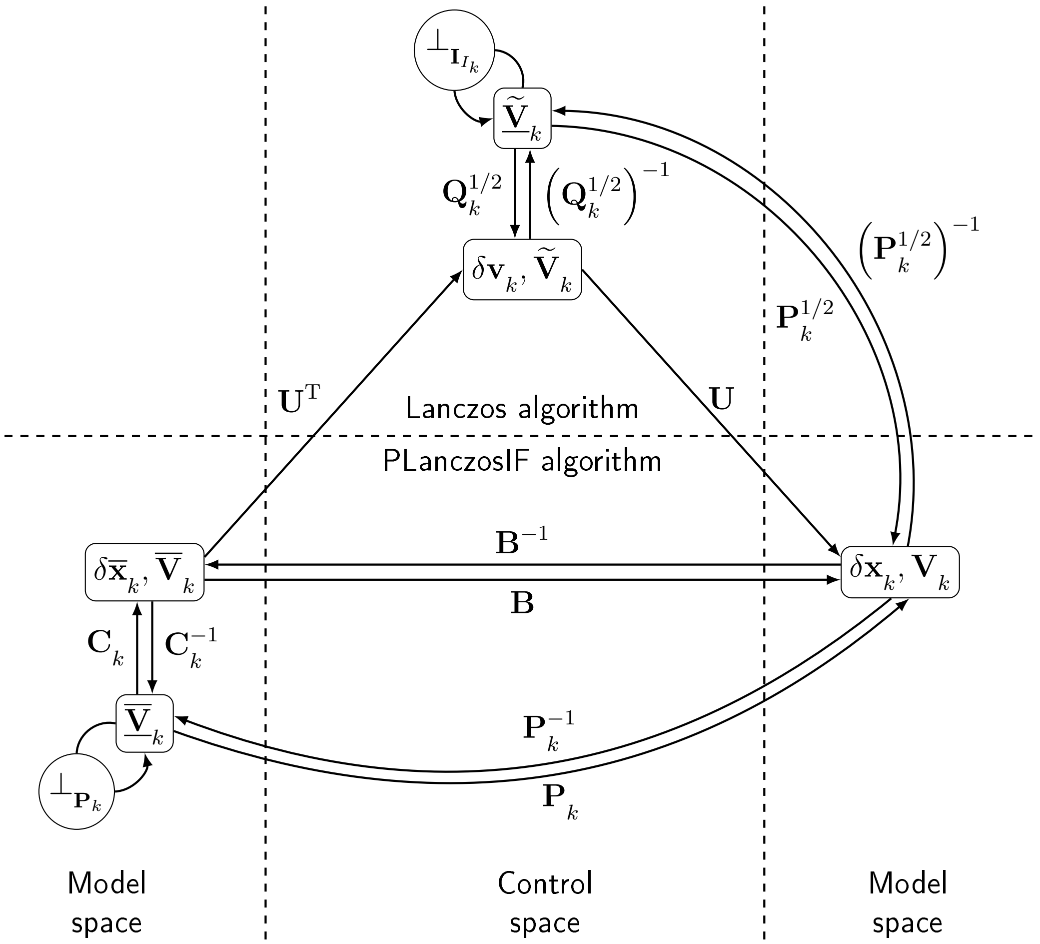

Figure 1: Map of spaces and links between them. Circles indicate orthogonality properties.

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. It is needed to compute the right-hand sides $\mathbf{b}_k^{\overline{\mathbf{x}}}$ and $\mathbf{b}_k^{\mathbf{v}}$. However, it is very usual to define:

- for $k = 1$: $\mathbf{x}_1^g = \mathbf{x}^b$,
- for $k > 1$: $\mathbf{x}_k^g = \mathbf{x}_{k-1}^a$.

Thus:

- for $k = 1$:

$$\delta\mathbf{x}_1^b = \mathbf{x}_1^g - \mathbf{x}^b = 0 \tag{31}$$

- for $k > 1$:

$$\begin{aligned}\delta\mathbf{x}_k^b &= \mathbf{x}^b - \mathbf{x}_k^g \\ &= \mathbf{x}^b - \mathbf{x}_{k-1}^a \\ &= \mathbf{x}^b - \left(\mathbf{x}_{k-1}^g + \delta\mathbf{x}_{k-1}^a\right) \\ &= \delta\mathbf{x}_{k-1}^b - \delta\mathbf{x}_{k-1}^a\end{aligned} \tag{32}$$

As a conclusion:

$$\delta\mathbf{x}_k^b = -\sum_{i=1}^{k-1} \delta\mathbf{x}_i^a \tag{33}$$

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

$$\delta\overline{\mathbf{x}}_k^b = -\sum_{i=1}^{k-1} \delta\overline{\mathbf{x}}_i^a \tag{34}$$

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

$$\delta\mathbf{v}_k^b = -\sum_{i=1}^{k-1} \delta\mathbf{v}_i^a \tag{35}$$

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

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 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 $\mathbf{B}_k$ at iteration k, and its square-root is denoted $\mathbf{U}_k$.

For $i < k$, we define three operators:

- $\mathbf{S}^{\mathbf{x}}_{k \rightarrow i} \in \mathbb{R}^{n_i \times n_k}$: simplification in model space from resolution $\mathcal{R}_k$ to resolution $\mathcal{R}_i$.
- $\mathbf{T}^{\mathbf{x}}_{i \rightarrow k} \in \mathbb{R}^{n_k \times n_i}$: interpolation in model space from resolution $\mathcal{R}_i$ to resolution $\mathcal{R}_k$.
- $\mathbf{T}^{\mathbf{v}}_{i \rightarrow k} \in \mathbb{R}^{m_k \times m_i}$: interpolation in control space from resolution $\mathcal{R}_i$ to resolution $\mathcal{R}_k$.

It should be noted that the interpolation operator $\mathbf{T}^{\mathbf{x}}_{i \rightarrow k}$ is only the right-inverse of the simplification operator $\mathbf{S}^{\mathbf{x}}_{k \rightarrow i}$. For $i < k$, $n_i < n_k$ and:

$$\mathbf{S}^{\mathbf{x}}_{k \rightarrow i}\mathbf{T}^{\mathbf{x}}_{i \rightarrow k} = \mathbf{I}_{n_i} \tag{36}$$

$$\mathbf{T}^{\mathbf{x}}_{i \rightarrow k}\mathbf{S}^{\mathbf{x}}_{k \rightarrow i} \neq \mathbf{I}_{n_k} \tag{37}$$

3.3 Usual but inconsistent strategy

It is a usual strategy to define a full resolution guess at each iteration k, denoted $\mathbf{x}^{g+}_k$. This full resolution guess can be simplified at resolution $\mathcal{R}_k$ if required:

$$\mathbf{x}^{g}_k = \mathbf{S}^{\mathbf{x}}_{K \rightarrow k}\mathbf{x}^{g+}_k \tag{38}$$

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

$$\mathbf{x}^{g+}_1 = \mathbf{x}^b \tag{39}$$

At the end of iteration k, an analysis increment $\delta\mathbf{x}^a_k$ is produced at resolution $\mathcal{R}_k$. In the previous section, the guess of iteration k has been defined as the analysis of iteration $k-1$, which is not possible anymore since the resolution increases. A common practice is to interpolate the analysis increment $\delta\mathbf{x}^a_{k-1}$ at full resolution, in order to compute the full resolution guess $\mathbf{x}^{g+}_k$:

$$\mathbf{x}^{g+}_k = \mathbf{x}^{g+}_{k-1} + \mathbf{T}^{\mathbf{x}}_{k-1 \rightarrow K}\delta\mathbf{x}^a_{k-1} \tag{40}$$

However, $\mathbf{x}^{g+}_{k-1} \neq \mathbf{T}^{\mathbf{x}}_{k-1 \rightarrow K}\mathbf{x}^{g}_{k-1}$, so $\mathbf{x}^{g+}_k$ is different from the analysis of iteration $k-1$ interpolated at full resolution:

$$\mathbf{T}^{\mathbf{x}}_{k-1 \rightarrow K}\mathbf{x}^{a}_{k-1} = \mathbf{T}^{\mathbf{x}}_{k-1 \rightarrow K}\left(\mathbf{x}^{g}_{k-1} + \delta\mathbf{x}^a_{k-1}\right) \neq \mathbf{x}^{g+}_k \tag{41}$$

It is possible to define the background increment at full resolution for each iteration k:

$$\delta \mathbf{x}_k^{b+} = \mathbf{x}^b - \mathbf{x}_k^{g+} \tag{42}$$

which can be simplified at resolution $\mathcal{R}_k$:

$$\begin{aligned}\delta \mathbf{x}_k^b &= \mathbf{S}_{K\rightarrow k}^{\mathbf{x}} \delta \mathbf{x}_k^{b+}\\ &= \mathbf{S}_{K\rightarrow k}^{\mathbf{x}} \mathbf{x}^b - \mathbf{x}_k^g\end{aligned} \tag{43}$$

This strategy would normally require $\mathbf{B}_k^{-1}$ to compute the right-hand sides $\mathbf{b}_k^{\overline{\mathbf{x}}}$ and $\mathbf{b}_k^{\mathbf{v}}$. In practice, transformed versions of equations (34) and (35) are used:

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

and

$$\delta \underline{\mathbf{v}}_k^b = - \sum_{i=1}^{k-1} \mathbf{T}_{i\rightarrow k}^{\mathbf{v}} \delta \mathbf{v}_i^a \tag{45}$$

It should be emphasized that this strategy is inconsistent:

$$\delta \underline{\overline{\mathbf{x}}}_k^b \neq \mathbf{B}_k^{-1} \delta \mathbf{x}_k^b \tag{46}$$

and

$$\delta \underline{\mathbf{v}}_k^b \neq \mathbf{U}_k^{\mathrm{T}} \mathbf{B}_k^{-1} \delta \mathbf{x}_k^b \tag{47}$$

3.4 Consistent strategy

A consistent strategy would start from equations (44) or (45) to define the background increment $\delta \mathbf{x}_k^b$:

$$\delta \mathbf{x}_k^b = \mathbf{B}_k \delta \underline{\overline{\mathbf{x}}}_k^b \tag{48}$$

or

$$\delta \mathbf{x}_k^b = \mathbf{U}_k \delta \underline{\mathbf{v}}_k^b \tag{49}$$

The guess would be defined as:

$$\mathbf{x}_k^g = \mathbf{S}_{K\rightarrow k}^{\mathbf{x}} \mathbf{x}^b - \delta \mathbf{x}_k^b \tag{50}$$

and the full resolution guess as:

$$\mathbf{x}_k^{g+} = \mathbf{x}^b - \mathbf{T}_{k\rightarrow K}^{\mathbf{x}} \delta \mathbf{x}_k^b \tag{51}$$

3.5 Equivalence condition on the background increment

The consistent strategy detailed in the previous section could yield different results depending on the interpolation operators. The condition required to ensure that both full $\mathbf{B}$ and square-root $\mathbf{B}$ preconditionings give the same result can be derived from the background increment implied by equations (44) and (45):

$$\begin{aligned}\delta\mathbf{x}_k^b &= \mathbf{B}_k \delta\underline{\overline{\mathbf{x}}}_k^b \\ &= -\mathbf{B}_k \sum_{i=1}^{k-1} \mathbf{T}_{i\to k}^{\mathbf{x}} \delta\overline{\mathbf{x}}_i^a \end{aligned} \tag{52}$$

whereas:

$$\begin{aligned}\delta\mathbf{x}_k^b &= \mathbf{U}_k \delta\underline{\mathbf{v}}_k^b \\ &= -\mathbf{U}_k \sum_{i=1}^{k-1} \mathbf{T}_{i\to k}^{\mathbf{v}} \delta\mathbf{v}_i^a \\ &= -\mathbf{U}_k \sum_{i=1}^{k-1} \mathbf{T}_{i\to k}^{\mathbf{v}} \mathbf{U}_i^{\mathrm{T}} \delta\overline{\mathbf{x}}_i^a \end{aligned} \tag{53}$$

Both implied background increments of equations (52) and (53) are equal if and only if:

$$\mathbf{U}_k^{\mathrm{T}} \mathbf{T}_{i\to k}^{\mathbf{x}} = \mathbf{T}_{i\to k}^{\mathbf{v}} \mathbf{U}_i^{\mathrm{T}} \tag{54}$$

3.6 Equivalence condition on Ritz vectors

If the resolution changes, the equivalence condition for preconditioners (30) must be adapted since $\mathbf{U}$ depends on k:

$$\mathbf{P}_k^{1/2} = \mathbf{Q}_k^{1/2} \mathbf{U}_k \tag{55}$$

In the case where the preconditioners are built using Ritz vectors, the map of Figure 1 shows that this condition is equivalent to a relation between the Ritz vectors:

$$\mathbf{W}_k \underline{\overline{\mathbf{V}}}_k = \underline{\widetilde{\mathbf{V}}}_k \tag{56}$$

with $\mathbf{W}_k = \left(\mathbf{Q}_k^{1/2}\right)^{-1} \mathbf{U}_k^{\mathrm{T}} \mathbf{C}_k$ If the resolution changes, the Ritz vectors also require an interpolation from their original resolution to the resolution $\mathcal{R}_k$ in order to build the preconditioner for iteration k, so the condition becomes:

$$\mathbf{W}_k \mathbf{T}_{i\to k}^{\mathbf{x}} \underline{\overline{\mathbf{V}}}_i = \mathbf{T}_{i\to k}^{\mathbf{v}} \underline{\widetilde{\mathbf{V}}}_i \tag{57}$$

which is verified if and only if:

$$\mathbf{W}_k \mathbf{T}_{i\to k}^{\mathbf{x}} = \mathbf{T}_{i\to k}^{\mathbf{v}} \mathbf{W}_i \tag{58}$$

The last condition concerns the orthogonality or Ritz vectors, that should not be lost during the interpolation process:

(59)

$$\left(\mathbf{T}^{\mathbf{x}}_{i\to k}\underline{\overline{\mathbf{V}}}_i\right)^{\mathrm{T}}\mathbf{P}_k\left(\mathbf{T}^{\mathbf{x}}_{i\to k}\underline{\overline{\mathbf{V}}}_i\right)=\mathbf{I}_{I_k} \tag{60}$$

and

$$\left(\mathbf{T}^{\mathbf{v}}_{i\to k}\underline{\widetilde{\mathbf{V}}}_i\right)^{\mathrm{T}}\left(\mathbf{T}^{\mathbf{v}}_{i\to k}\underline{\widetilde{\mathbf{V}}}_i\right)=\mathbf{I}_{I_k} \tag{61}$$

3.7 Example of equivalence

An example of equivalence is the following:

- The vector $\boldsymbol{\gamma}\in\mathbb{R}^{n_K}$ of positive coefficients is used to define the diagonal matrices $\boldsymbol{\Gamma}_k\in\mathbb{R}^{n_k\times n_k}$:

  $$\Gamma_{k,\alpha\beta}=\left\{\begin{array}{lll}\gamma_i & \text{if} & \alpha=\beta\\ 0 & \text{if} & \alpha\neq\beta\end{array}\right. \tag{62}$$

  At resolution $\mathcal{R}_k$, the background error covariance matrix $\mathbf{B}_k$ is defined as:

  $$\mathbf{B}_k=\mathbf{F}_k\boldsymbol{\Gamma}_k\mathbf{F}_k^{-1} \tag{63}$$

  where $\mathbf{F}_k\in\mathbb{R}^{n_k\times n_k}$ is orthogonal:

  $$\mathbf{F}_k\mathbf{F}_k^{\mathrm{T}}=\mathbf{F}_k^{\mathrm{T}}\mathbf{F}_k=\mathbf{I}_{n_k} \tag{64}$$

  Its square-root $\mathbf{U}_k$ is simply:

  $$\mathbf{U}_k=\mathbf{F}_k\boldsymbol{\Gamma}_k^{1/2}\mathbf{G}_k \tag{65}$$

  where $\mathbf{G}\in\mathbb{R}^{n_k\times m_k}$ is any matrix verifying:

  $$\mathbf{G}_k\mathbf{G}_k^{\mathrm{T}}=\mathbf{I}_{n_k} \tag{66}$$

- The interpolators are defined as:

  $$\mathbf{T}^{\mathbf{x}}_{i\to k}=\mathbf{F}_k\boldsymbol{\Delta}_{i\to k}\mathbf{F}_i^{\mathrm{T}} \tag{67}$$

  and

  $$\mathbf{T}^{\mathbf{v}}_{i\to k}=\mathbf{G}_k^{\mathrm{T}}\boldsymbol{\Delta}_{i\to k}\mathbf{G}_i \tag{68}$$

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

  $$\Delta_{i\to k,\alpha\beta}=\left\{\begin{array}{lll}1 & \text{if} & \alpha=\beta\\ 0 & \text{if} & \alpha\neq\beta\end{array}\right. \tag{69}$$

Thus:

$$
\begin{aligned}
\mathbf{U}_k^{\mathrm{T}}\mathbf{T}^{\mathbf{x}}_{i\rightarrow k} &= \mathbf{G}_k^{\mathrm{T}}\mathbf{\Gamma}_k^{1/2}\mathbf{F}_k^{\mathrm{T}}\mathbf{F}_k\mathbf{\Delta}_{i\rightarrow k}\mathbf{F}_i^{\mathrm{T}} \\
&= \mathbf{G}_k^{\mathrm{T}}\mathbf{\Gamma}_k^{1/2}\mathbf{\Delta}_{i\rightarrow k}\mathbf{F}_i^{\mathrm{T}}
\end{aligned}
\tag{70}
$$

and

$$
\begin{aligned}
\mathbf{T}^{\mathbf{v}}_{i\rightarrow k}\mathbf{U}_i^{\mathrm{T}} &= \mathbf{G}_k^{\mathrm{T}}\mathbf{\Delta}_{i\rightarrow k}\mathbf{G}_i\mathbf{G}_i^{\mathrm{T}}\mathbf{\Gamma}_i^{1/2}\mathbf{F}_i^{\mathrm{T}} \\
&= \mathbf{G}_k^{\mathrm{T}}\mathbf{\Delta}_{i\rightarrow k}\mathbf{\Gamma}_i^{1/2}\mathbf{F}_i^{\mathrm{T}}
\end{aligned}
\tag{71}
$$

Since $\mathbf{\Gamma}_k^{1/2}\mathbf{\Delta}_{i\rightarrow k} = \mathbf{\Delta}_{i\rightarrow k}\mathbf{\Gamma}_i^{1/2}$, the equivalence condition on the background increment (54) is verified.

Equivalence conditions on preconditioners (58), (59), (61) are a lot harder to proove, but the code multi shows that it seems to work (`https://github.com/benjaminmenetrier/multi`).

References

Gurol S. 2013. Résolution de problèmes des moindres carrés non-linéaires régularisés dans l'espace dual avec applications à l'assimilation de données. PhD thesis, Université de Toulouse.