A modern implementation of the spectral background error covariance matrix

Benjamin Menetrier

September 24th, 2025 - 47th EWGLAM meeting - Norrköping

Norwegian Meteorological Institute

Outline

The spectral background error covariance model

Old implementation

A quick tour of SABER

New implementation

Interoperability of old and new implementations

Taking advantage of SABER

Conclusions

General formulation

- The spectral **B** is a robust and well-conditioned covariance model, based on a sequence of parametrized operators.
- The parameters can be estimated using ensemble data over a long period, giving a climatological estimate.
- The general structure of **B** is given by: $\mathbf{B} = \mathbf{WKCK}^{\mathrm{T}}\mathbf{W}^{\mathrm{T}}$ where:
  - The input variables are: v_x, v_y, T, p_s and q.
  - **W** is an inverse spectral transform including a wind transform from $\left(\widehat{\zeta}, \widehat{\eta}, \widehat{T}, \widehat{p}_s, \widehat{q}\right)$ to (v_x, v_y, T, p_s, q).
  - **K** is a balance operator in spectral space.
  - **C** is a univariate covariance operator in spectral space.
- The resulting covariances are non-separable, homogeneous and isotropic.
- Reference papers: Derber and Bouttier (1999) and Berre (2000)

Calibration process

Each component of $\mathbf{B} = \mathbf{W}\mathbf{K}\mathbf{C}\mathbf{K}^{\mathrm{T}}\mathbf{W}^{\mathrm{T}}$ is calibrated **sequentially** and **from outer to inner components**:

1. We start from an ensemble of N members in grid-point space, with variables v_x, v_y, T, p_s and q.
2. The inverse $\mathbf{W}^{-1}$ is applied to each ensemble member.
3. The regression matrices are estimated from the updated ensemble to define the balance operator $\mathbf{K}$ (block-lower triangular).
4. The inverse $\mathbf{K}^{-1}$ is applied to each ensemble member.
5. The covariance matrices are estimated from the updated ensemble to define the univariate covariance operator $\mathbf{C}$ (block-diagonal).

Outline

The spectral background error covariance model

Old implementation

A quick tour of SABER

New implementation

Interoperability of old and new implementations

Taking advantage of SABER

Conclusions

Old implementation

In the IAL code (IFS-Arpege-LAM), the common code of the ACCORD consortium, the implementation of the spectral **B** is obsolete:

- Non-object-oriented Fortran 90, some parts coded in 1996.
- LAM and global options all mixed together.
- Hard-coded variables and fixed operators structures.
- Unreadable data format (binary), even if NetCDF is available.

and with problematic development practices:

- Almost impossible to compile on local machines or laptops.
- No proper testing system.
- Duplicated work because of the IAL versioning system (cycles 46, 49, 50, etc.) → very time-consuming.
- Externalized calibration tool FESTAT, inconsistent with the application in minimization, and scaling poorly.

Outline

The spectral background error covariance model

Old implementation

A quick tour of SABER

New implementation

Interoperability of old and new implementations

Taking advantage of SABER

Conclusions

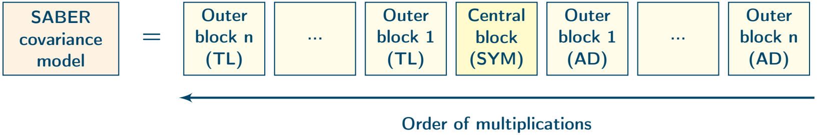
SABER overview

SABER, for System-Agnostic Background Error Representation, is the repository handling generic background error covariance operators in the JEDI project lead by the JCSDA and the UKMO.

SABER is interfaced with OOPS, the Object-Oriented Prediction System, a C++ layer that drives the data assimilation.

There are two different flavors of OOPS: ECMWF and JEDI. SABER can run with both versions of OOPS.

The SABER covariance model is built as a succession of blocks:

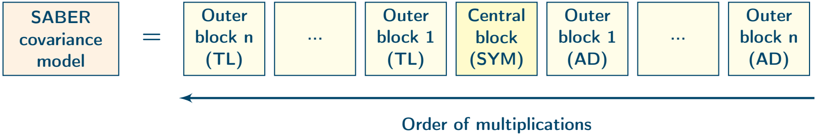


SABER covariance model = Outer block n (TL) | ... | Outer block 1 (TL) | Central block (SYM) | Outer block 1 (AD) | ... | Outer block n (AD)

Order of multiplications

SABER design

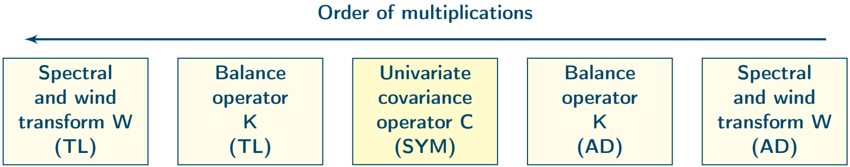
The SABER blocks architecture:



- Two classes of blocks with their own constructors/methods:
  - Central block: auto-adjoint (symmetric).
  - Outer blocks: with forward (TL), adjoint (AD) and left-inverse multiplications.
- Each block includes its own calibration method (if needed).
- Blocks are interoperable and model-agnostic: ATLAS-based geometry and fields.
- Each block is properly tested with the `ctest` utility.

SABER calibration

Calibration example with the spectral **B**:



Order of multiplications

Spectral and wind transform W (TL)

Balance operator K (TL)

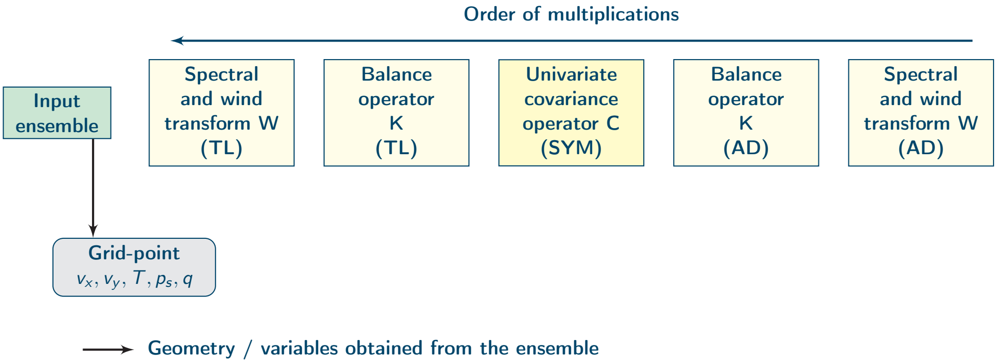
Univariate covariance operator C (SYM)

Balance operator K (AD)

Spectral and wind transform W (AD)

SABER calibration

Calibration example with the spectral **B**:



Order of multiplications

Input ensemble

Spectral and wind transform W (TL)

Balance operator K (TL)

Univariate covariance operator C (SYM)

Balance operator K (AD)

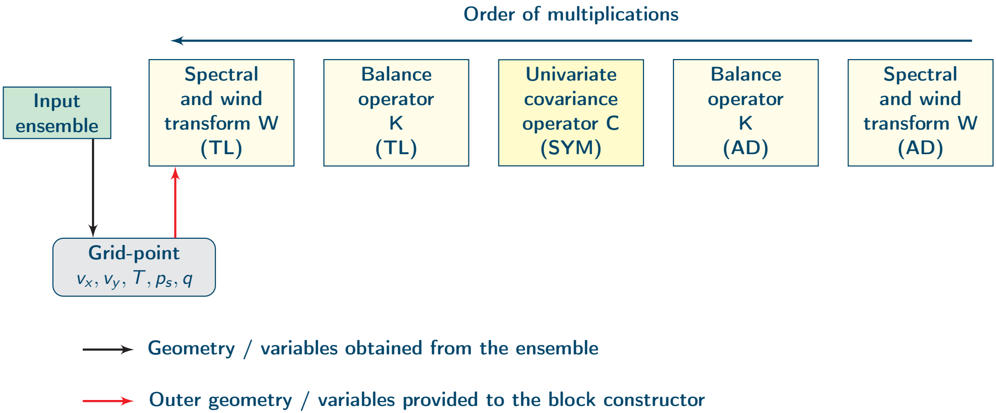
Spectral and wind transform W (AD)

Grid-point
v_x, v_y, T, p_s, q

→ **Geometry / variables obtained from the ensemble**

SABER calibration

Calibration example with the spectral **B**:



Order of multiplications

Input ensemble

Spectral and wind transform W (TL)

Balance operator K (TL)

Univariate covariance operator C (SYM)

Balance operator K (AD)

Spectral and wind transform W (AD)

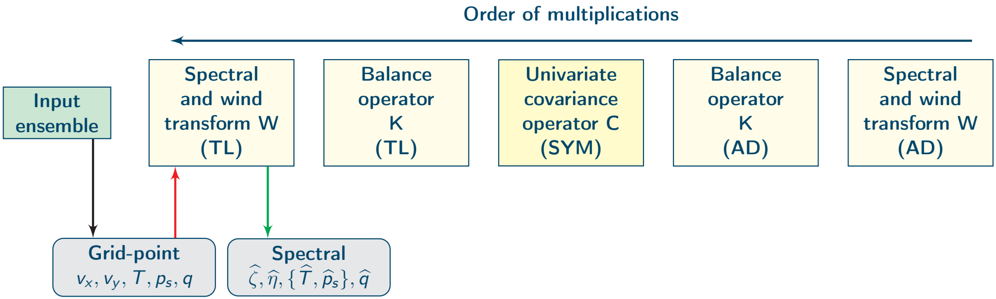
Grid-point
v_x, v_y, T, p_s, q

Geometry / variables obtained from the ensemble

Outer geometry / variables provided to the block constructor

SABER calibration

Calibration example with the spectral **B**:



Order of multiplications

Input ensemble

Spectral and wind transform W (TL)

Balance operator K (TL)

Univariate covariance operator C (SYM)

Balance operator K (AD)

Spectral and wind transform W (AD)

Grid-point
v_x, v_y, T, p_s, q

Spectral
$\widehat{\zeta}, \widehat{\eta}, \{\widehat{T}, \widehat{p_s}\}, \widehat{q}$

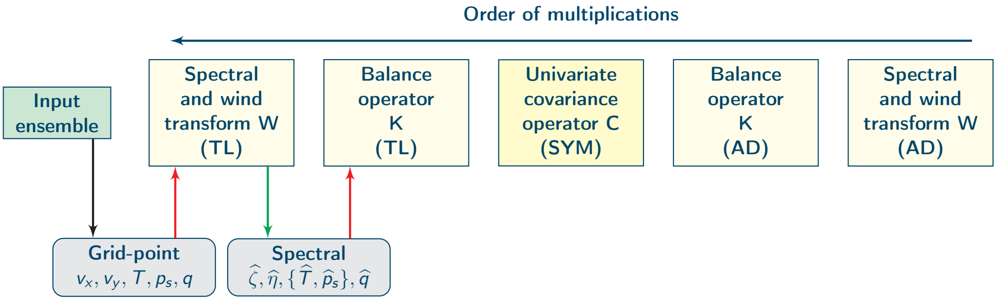
→ **Geometry / variables obtained from the ensemble**

→ **Outer geometry / variables provided to the block constructor**

→ **Inner geometry / variables returned by the block**

SABER calibration

Calibration example with the spectral **B**:



Order of multiplications

Input ensemble

Spectral and wind transform W (TL)

Balance operator K (TL)

Univariate covariance operator C (SYM)

Balance operator K (AD)

Spectral and wind transform W (AD)

Grid-point
v_x, v_y, T, p_s, q

Spectral
$\widehat{\zeta}, \widehat{\eta}, \{\widehat{T}, \widehat{p_s}\}, \widehat{q}$

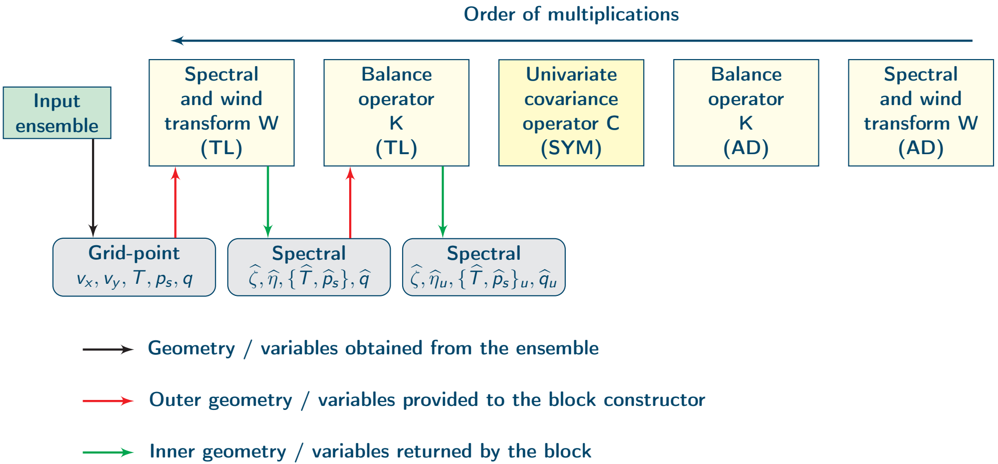
→ **Geometry / variables obtained from the ensemble**

→ **Outer geometry / variables provided to the block constructor**

→ **Inner geometry / variables returned by the block**

SABER calibration

Calibration example with the spectral **B**:



Order of multiplications

Input ensemble

Spectral and wind transform W (TL)

Balance operator K (TL)

Univariate covariance operator C (SYM)

Balance operator K (AD)

Spectral and wind transform W (AD)

Grid-point
v_x, v_y, T, p_s, q

Spectral
$\widehat{\zeta}, \widehat{\eta}, \{\widehat{T}, \widehat{p}_s\}, \widehat{q}$

Spectral
$\widehat{\zeta}, \widehat{\eta}_u, \{\widehat{T}, \widehat{p}_s\}_u, \widehat{q}_u$

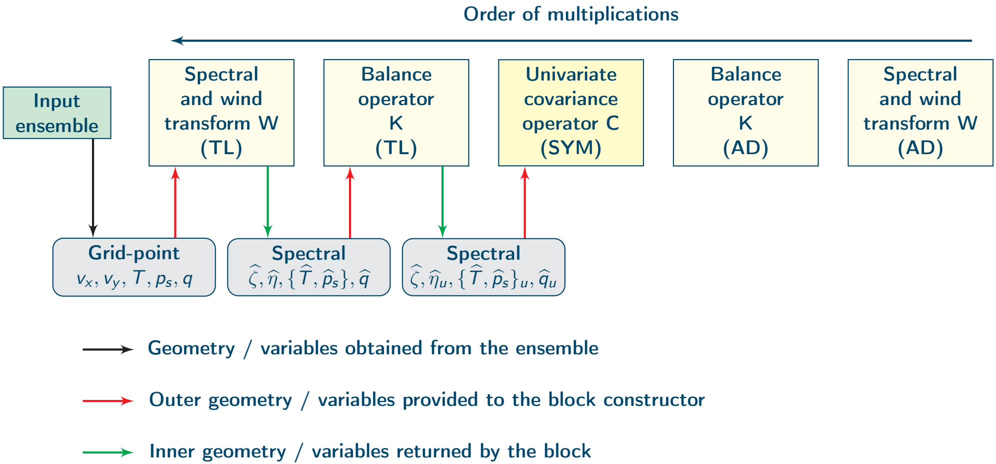
→ **Geometry / variables obtained from the ensemble**

→ **Outer geometry / variables provided to the block constructor**

→ **Inner geometry / variables returned by the block**

SABER calibration

Calibration example with the spectral **B**:



Order of multiplications

Input ensemble

Spectral and wind transform W (TL)

Balance operator K (TL)

Univariate covariance operator C (SYM)

Balance operator K (AD)

Spectral and wind transform W (AD)

Grid-point v_x, v_y, T, p_s, q

Spectral $\widehat{\zeta}, \widehat{\eta}, \{\widehat{T}, \widehat{p}_s\}, \widehat{q}$

Spectral $\widehat{\zeta}, \widehat{\eta}_u, \{\widehat{T}, \widehat{p}_s\}_u, \widehat{q}_u$

Geometry / variables obtained from the ensemble

Outer geometry / variables provided to the block constructor

Inner geometry / variables returned by the block

Outline

The spectral background error covariance model

Old implementation

A quick tour of SABER

New implementation

Interoperability of old and new implementations

Taking advantage of SABER

Conclusions

New spectral **B** in SABER

Cleaner code to make development and maintenance easier:

- Clean object-oriented C++.
- Options provided in parameters files with human-readable formats (e.g JSON).
- Configurable variables and operators structures.
- NetCDF data format (default) + legacy Fortran reader/writer for backward compatibility with the old implementation.

Modern coding practices:

- Open source code on GitHub reviewed by contributors (JCSDA and UKMO).
- Easy installation on any Linux machine.
- Thorough unit testing on mutliple compilers/plateforms.
- Independent from the IAL versioning policy.

Code improvements

New features compared to the old implementation:

- Choice of the FFT backend (FFTW or ECTRANS).
- Better scalability and consistency of the calibration procedure.
- Iterative calibration option: more I/O but **only one** ensemble member in memory at any time.
- Covariance filtering option (to damp sampling noise).
- Sliding time-average option to increase seasonal flow-dependency.
- Inter-operability with **all** the other SABER blocks (increase of the size of the extension zone, interpolation, change of variables, hybridization, etc.).

Outline

The spectral background error covariance model

Old implementation

A quick tour of SABER

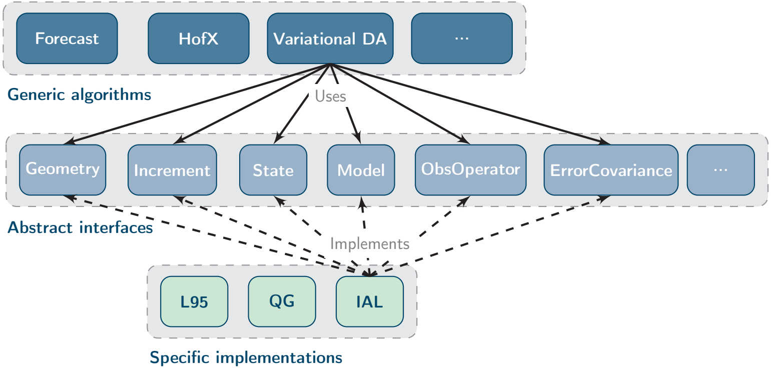
New implementation

Interoperability of old and new implementations

Taking advantage of SABER

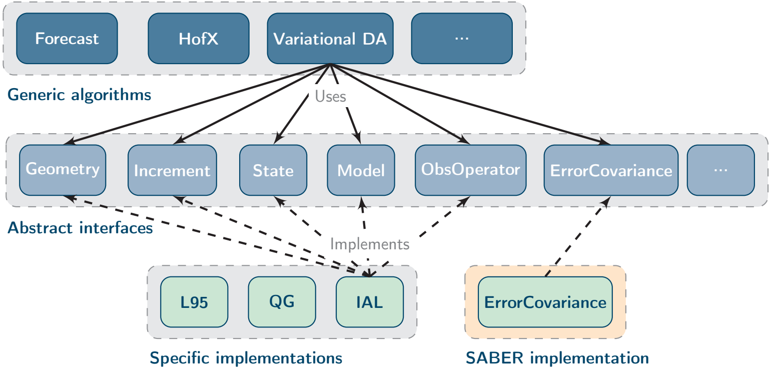
Conclusions

OOPS structure and SABER addition



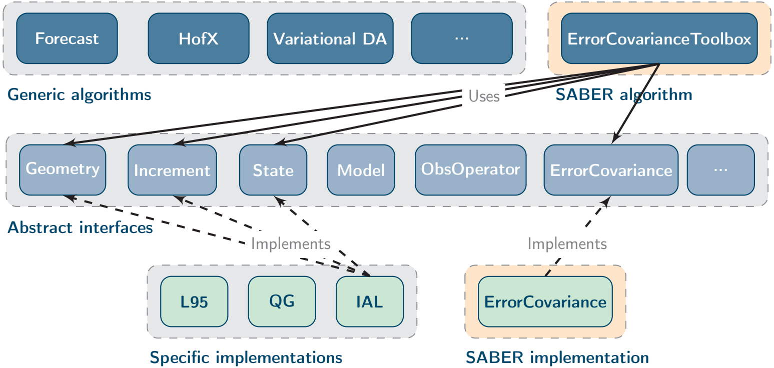
- IAL implements all the required interfaces for variational DA

OOPS structure and SABER addition



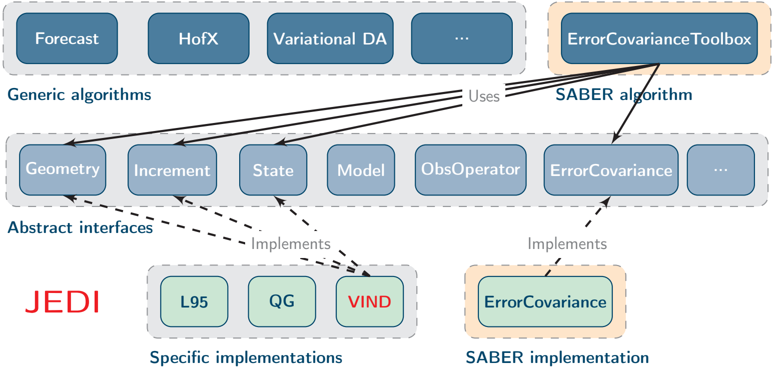
- IAL implements all the required interfaces for variational DA
- SABER generic implementation of ErrorCovariance

OOPS structure and SABER addition



- IAL implements all the required interfaces for variational DA
- SABER generic implementation of ErrorCovariance
- SABER generic algorithm ErrorCovarianceToolbox for calibrating/testing

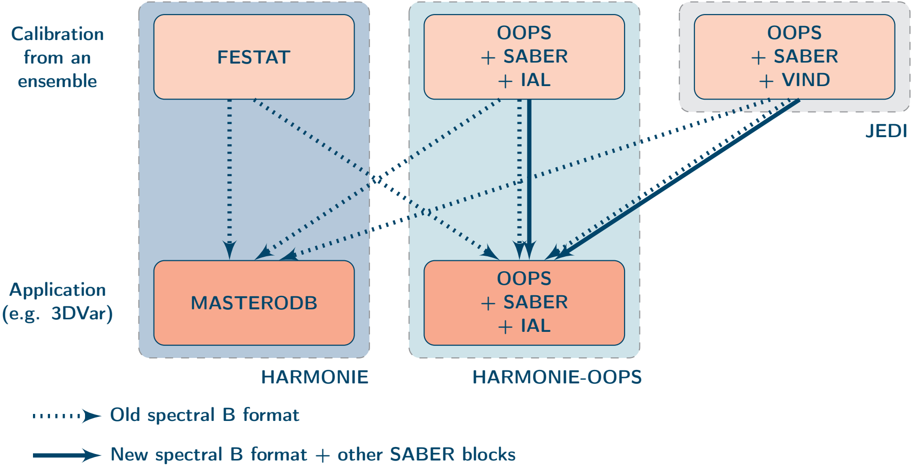
OOPS structure and SABER addition



- IAL implements all the required interfaces for variational DA
- SABER generic implementation of ErrorCovariance
- SABER generic algorithm ErrorCovarianceToolbox for calibrating/testing
- In JEDI, the VIND implementation can read FA files

Combining calibration and application steps

Interoperability between environments:



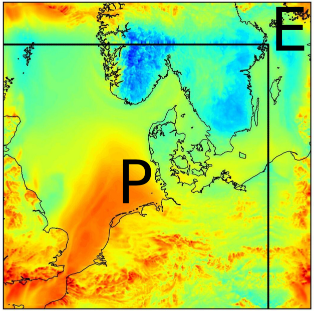
Outline

- The spectral background error covariance model
- Old implementation
- A quick tour of SABER
- New implementation
- Interoperability of old and new implementations
- Taking advantage of SABER
- Conclusions

Expanding the extension zone

A non-physical "extension" zone can make regional fields periodic, to use spectral transforms. The size of this zone should be:

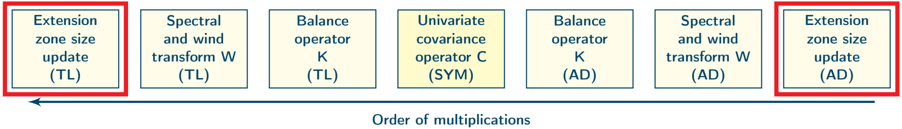
- rather small in the dynamics, to save CPU,
- larger in data assimilation, to avoid increments wrapping.

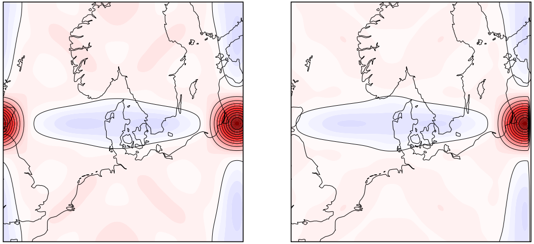


Physical (P) and extension (E) zone of a limited-area domain

Expanding the extension zone

An extra block **E** can be added to change change the size of the extension zone: $\mathbf{B} = \mathbf{E}\mathbf{W}\mathbf{K}\mathbf{C}\mathbf{K}^\mathrm{T}\mathbf{W}^\mathrm{T}\mathbf{E}^\mathrm{T}$ or in SABER block representation:

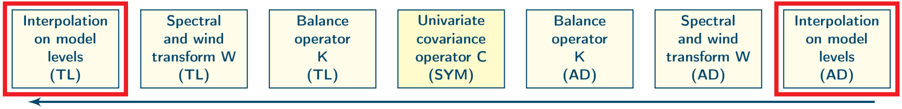

Extension zone size update (TL)
Spectral and wind transform W (TL)
Balance operator K (TL)
Univariate covariance operator C (SYM)
Balance operator K (AD)
Spectral and wind transform W (AD)
Extension zone size update (AD)
Order of multiplications



Extension size update: 11 points (left) and 81 points (right)

Changing the vertical coordinate

An extra vertical interpolation block **V** can define B on pressure levels instead of model levels: $\mathbf{B} = \mathbf{V}\mathbf{W}\mathbf{K}\mathbf{C}\mathbf{K}^{\mathrm{T}}\mathbf{W}^{\mathrm{T}}\mathbf{V}^{\mathrm{T}}$ or in SABER block representation:



Interpolation on model levels (TL)

Spectral and wind transform W (TL)

Balance operator K (TL)

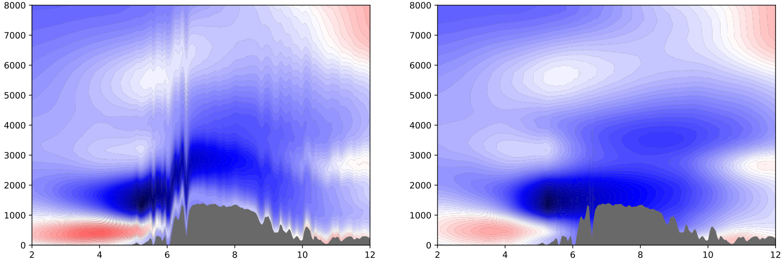
Univariate covariance operator C (SYM)

Balance operator K (AD)

Spectral and wind transform W (AD)

Interpolation on model levels (AD)

Order of multiplications



Increment section in altitude coordinate, without (left) and with **V** (right)

Conclusions

- The current 3D-Var relies on a spectral **B** with many issues:
  - Obsolete coding and development standards, no testing.
  - Impossible to compile/run on local machines.
  - Painful development with multiple IAL cycles.
- The transition to a pure EnVar requires a large ensemble (at least 50 members), which will not be available in a near future.
- Porting the spectral **B** in the SABER framework opens new R&D paths:
  - Easier maintenance and development, complete testing.
  - Improved calibration process, with backward compatibility.
  - Combination with other SABER blocks.
  - Enhanced collaboration with other centers.