The Normalized Interpolated Convolution on an Adaptive Subgrid (NICAS) method, a new implementation of localization for EnVar applications

Benjamin Ménétrier	Météo-France/CNRS, CNRM
Etienne Arbogast	Météo-France/CNRS, CNRM
Loïk Berre	Météo-France/CNRS, CNRM
Yannick Trémolet	JCSDA

Workshop on Sensitivity Analysis and Data Assimilation
July 2nd, 2018 - Aveiro, Portugal

METEO FRANCE

Explicit convolution

Main goal:

Designing a generic method to apply a localization matrix for EnVar (normalized convolution operator) **on any grid type**

Standard methods:

- Spectral/wavelet transforms → regular grid required
- Recursive filters → regular grid required
- Explicit/implicit diffusion → normalization issues

Advantages of an explicit convolution **C** :

- Work on any grid type
- Exact normalization ($C_{ii} = 1$)

Drawback: the computational cost scales as $O(n^2)$, where n is the size of the model grid...

METEO FRANCE

Explicit convolution

Main goal:

Designing a generic method to apply a localization matrix for EnVar (normalized convolution operator) **on any grid type**

Standard methods:

- Spectral/wavelet transforms → regular grid required
- Recursive filters → regular grid required
- Explicit/implicit diffusion → normalization issues

Advantages of an explicit convolution **C** :

- Work on any grid type
- Exact normalization ($C_{ii} = 1$)

Drawback: the computational cost scales as $O(n^2)$, where n is the size of the model grid...

Explicit convolution

Main goal:

Designing a generic method to apply a localization matrix for EnVar (normalized convolution operator) **on any grid type**

Standard methods:

- Spectral/wavelet transforms $\rightarrow$ regular grid required
- Recursive filters $\rightarrow$ regular grid required
- Explicit/implicit diffusion $\rightarrow$ normalization issues

Advantages of an explicit convolution **C** :

- Work on any grid type
- Exact normalization ($C_{ii} = 1$)

Drawback: the computational cost scales as $O(n^2)$, where n is the size of the model grid...

Explicit convolution

To limit the computational cost, we approximate $\mathbf{C}$ on a subgrid (subset of n^s points of the model grid):

$$\mathbf{C} \approx \mathbf{S}\mathbf{C}^s\mathbf{S}^{\mathrm{T}}$$

where

- $\mathbf{S}$ is an interpolation from the subgrid to the model grid
- $\mathbf{C}^s$ is a convolution matrix on the subgrid

If $n^s \ll n$, then the total cost scales as $O(n)$ (interpolation cost).

Issues with this approach:

- If the subgrid density is too coarse compared to the convolution length-scale, the convolution is distorded.
- Normalization breaks down because of the interpolation: even if $\mathbf{C}^s$ is normalized, $\mathbf{S}\mathbf{C}^s\mathbf{S}^{\mathrm{T}}$ is not.

METEO FRANCE

Explicit convolution

To limit the computational cost, we approximate $\mathbf{C}$ on a subgrid (subset of n^s points of the model grid):

$$\mathbf{C} \approx \mathbf{S}\mathbf{C}^s\mathbf{S}^\mathrm{T}$$

where

- $\mathbf{S}$ is an interpolation from the subgrid to the model grid
- $\mathbf{C}^s$ is a convolution matrix on the subgrid

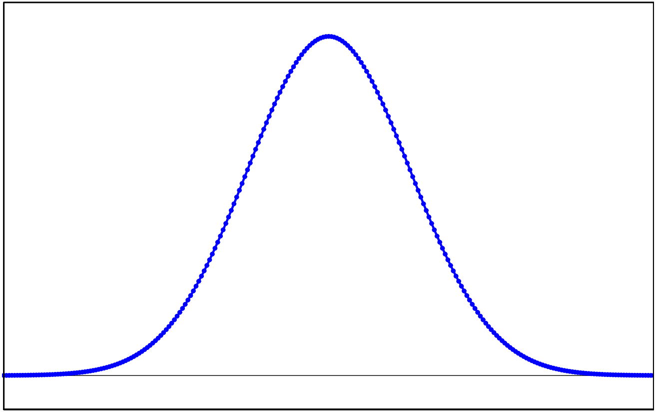
If $n^s \ll n$, then the total cost scales as $O(n)$ (interpolation cost).

Issues with this approach:

- If the subgrid density is too coarse compared to the convolution length-scale, the convolution is distorded.
- Normalization breaks down because of the interpolation: even if $\mathbf{C}^s$ is normalized, $\mathbf{S}\mathbf{C}^s\mathbf{S}^\mathrm{T}$ is not.

Convolution on a subgrid

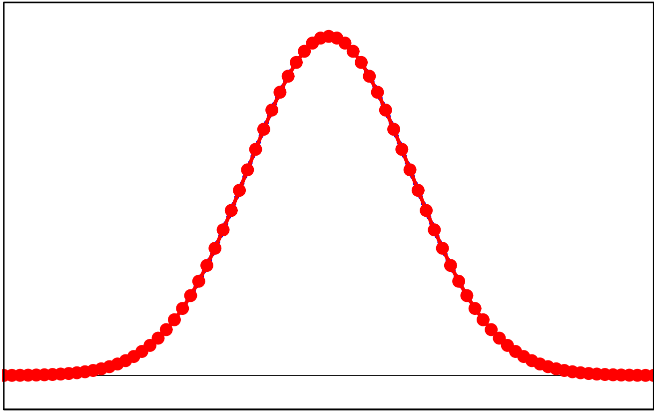
Convolution function on model grid



Model grid (blue)
Large convolution length-scale

Convolution on a subgrid

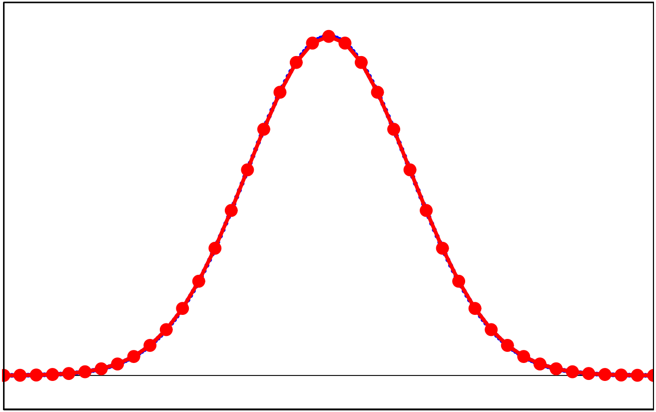
Subsampling: 1 point over 3



Model grid (blue) and subgrid (red)
Large convolution length-scale

Convolution on a subgrid

Subsampling: 1 point over 6

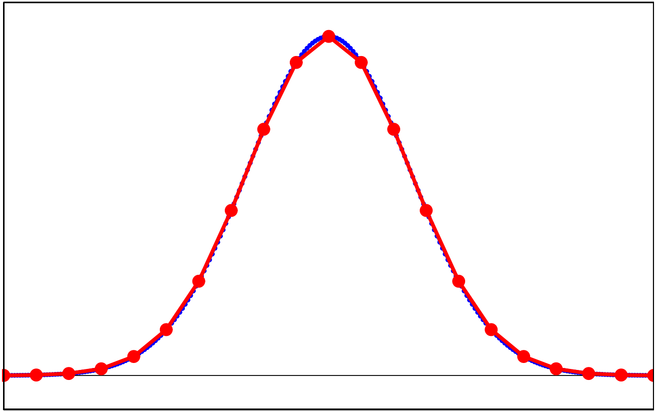


Model grid (blue) and subgrid (red)
Large convolution length-scale

METEO FRANCE

Convolution on a subgrid

Subsampling: 1 point over 12

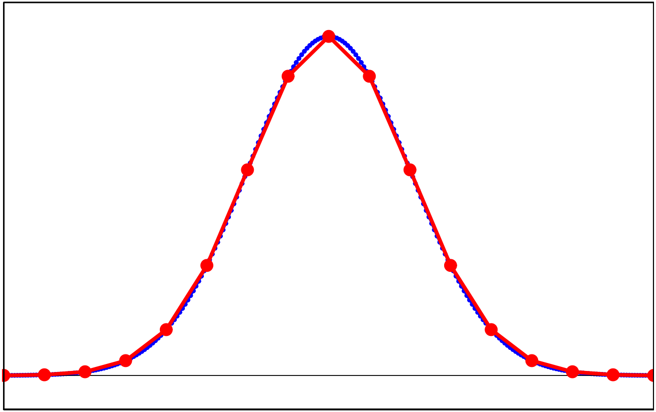


Model grid (blue) and subgrid (red)

Large convolution length-scale

Convolution on a subgrid

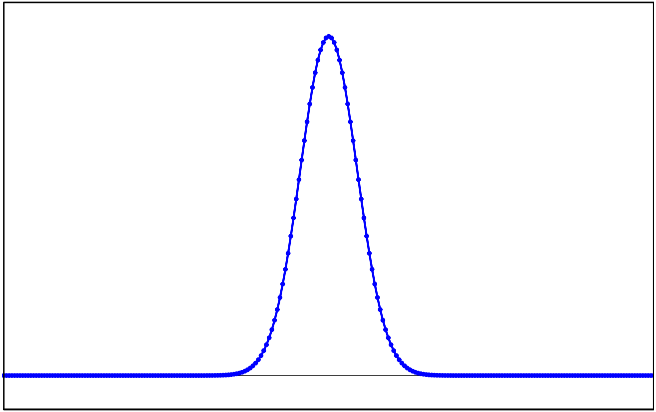
Subsampling: 1 point over 15



Model grid (blue) and subgrid (red)
Large convolution length-scale

Convolution on a subgrid

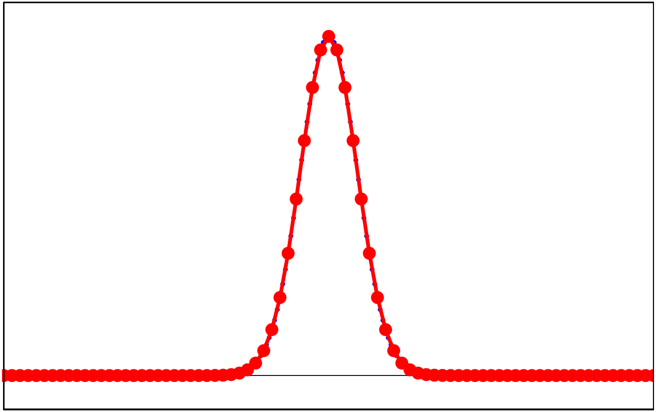
Convolution function on model grid



Model grid (blue)
Small convolution length-scale

Convolution on a subgrid

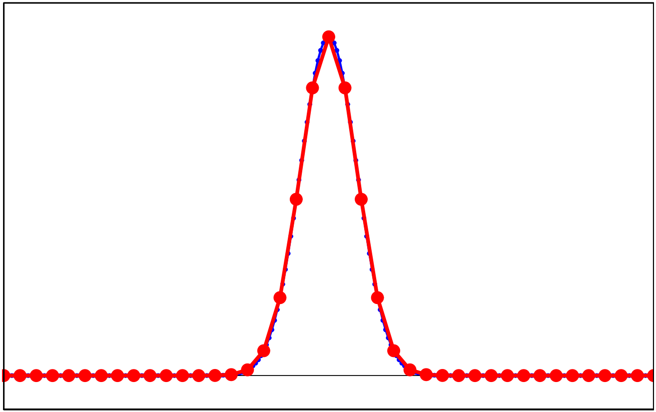
Subsampling: 1 point over 3



Model grid (blue) and subgrid (red)
Small convolution length-scale

Convolution on a subgrid

Subsampling: 1 point over 6



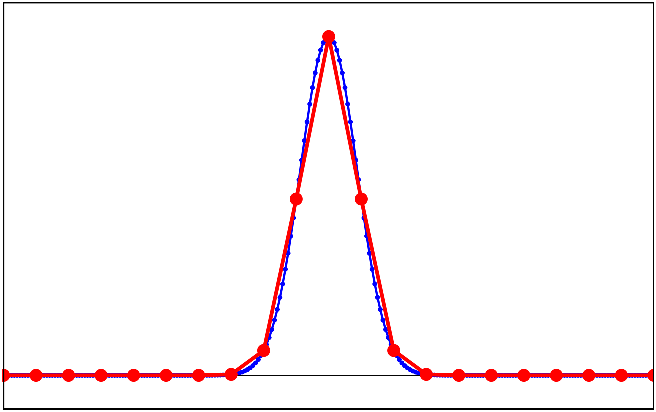
Model grid (blue) and subgrid (red)

Small convolution length-scale

METEO FRANCE

Convolution on a subgrid

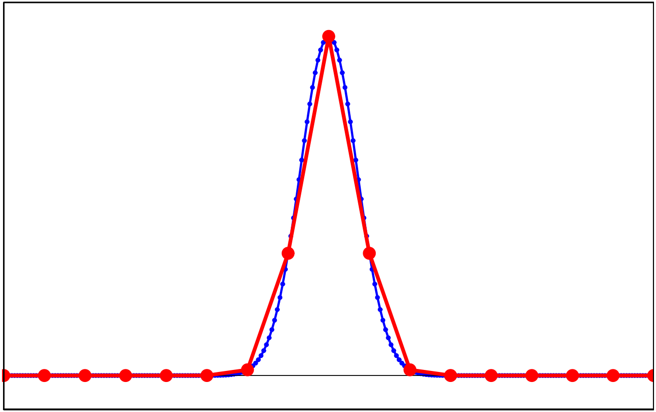
Subsampling: 1 point over 12



Model grid (blue) and subgrid (red)
Small convolution length-scale

Convolution on a subgrid

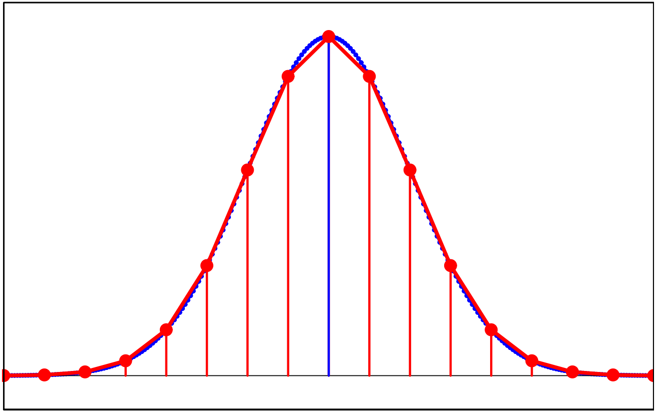
Subsampling: 1 point over 15



Model grid (blue) and subgrid (red)
Small convolution length-scale

Convolution on a subgrid

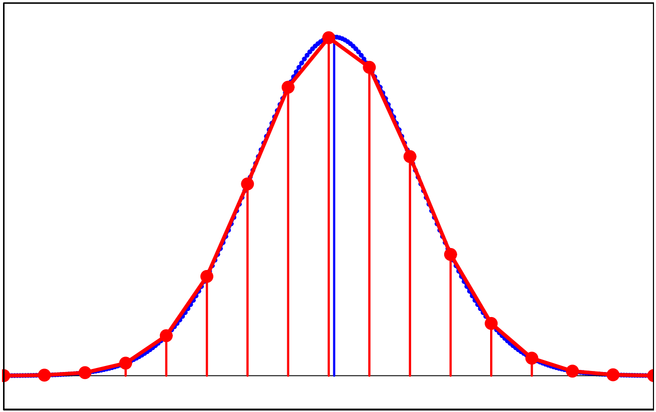
Normalization issue:



Model grid (blue) and subgrid (red)
Large convolution length-scale

Convolution on a subgrid

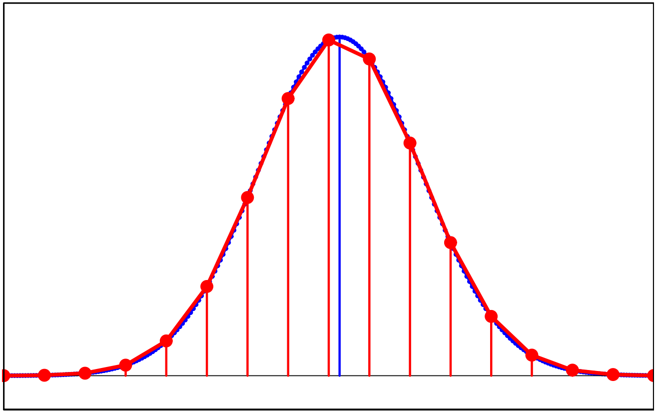
Normalization issue:



Model grid (blue) and subgrid (red)
Large convolution length-scale

Convolution on a subgrid

Normalization issue:

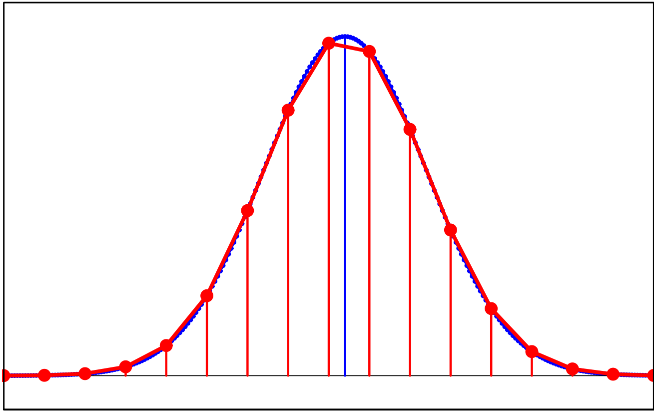


Model grid (blue) and subgrid (red)

Large convolution length-scale

Convolution on a subgrid

Normalization issue:



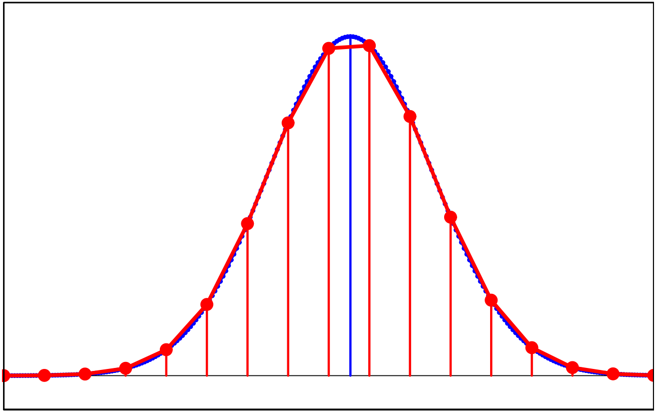
Model grid (blue) and subgrid (red)

Large convolution length-scale

METEO FRANCE

Convolution on a subgrid

Normalization issue:



Model grid (blue) and subgrid (red)
Large convolution length-scale

Explicit convolution

The **NICAS** method (Normalized Interpolated Convolution from an Adaptive Subgrid) is given by:

$$\widetilde{\mathbf{C}} = \mathbf{N}\mathbf{S}\mathbf{C}^s\mathbf{S}^\mathrm{T}\mathbf{N}^\mathrm{T}$$

where

- $\mathbf{N}$ is a diagonal normalization matrix.
- The subgrid is locally adapted to the convolution length-scale.

Several questions:

- What subgrid?
- What convolution function?
- What parallelization method?
- What software infrastructure?

METEO FRANCE

Explicit convolution

The **NICAS** method (Normalized Interpolated Convolution from an Adaptive Subgrid) is given by:

$$\widetilde{\mathbf{C}} = \mathbf{N}\mathbf{S}\mathbf{C}^s\mathbf{S}^\mathrm{T}\mathbf{N}^\mathrm{T}$$

where

- **N** is a diagonal normalization matrix.
- The subgrid is locally adapted to the convolution length-scale.

Several questions:

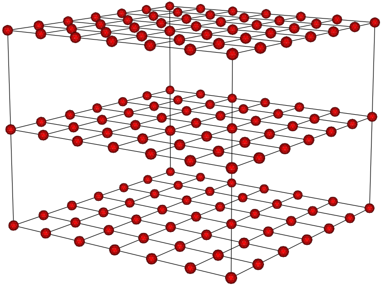
- What subgrid?
- What convolution function?
- What parallelization method?
- What software infrastructure?

METEO FRANCE

Outline

Subgrid definition

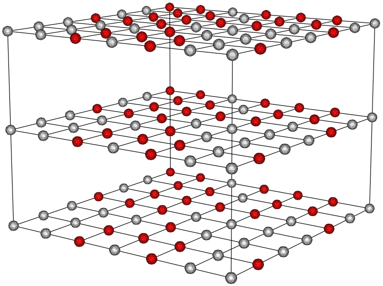
- The model grid is subsampled to define the convolution subgrid following three steps:
  1. horizontal subsampling, level-independent,
  2. vertical subsampling, similar for all columns,
  3. horizontal subsampling, level-dependent.



Full model grid

Subgrid definition

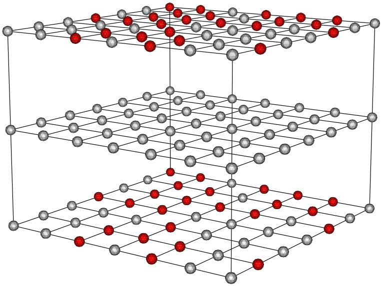
- The model grid is subsampled to define the convolution subgrid following three steps:
  1. horizontal subsampling, level-independent,
  2. vertical subsampling, similar for all columns,
  3. horizontal subsampling, level-dependent.



Step 1: horizontal subsampling, level-independent

Subgrid definition

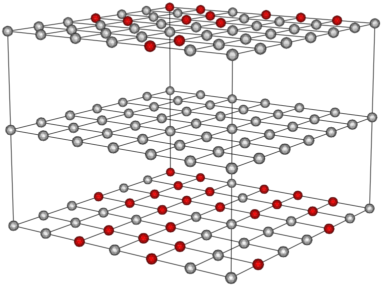
- The model grid is subsampled to define the convolution subgrid following three steps:
    1. horizontal subsampling, level-independent,
    2. vertical subsampling, similar for all columns,
    3. horizontal subsampling, level-dependent.



Step 2: vertical subsampling, similar for all columns

Subgrid definition

- The model grid is subsampled to define the convolution subgrid following three steps:
  1. horizontal subsampling, level-independent,
  2. vertical subsampling, similar for all columns,
  3. horizontal subsampling, level-dependent.



Step 3: horizontal subsampling, level-dependent

METEO FRANCE

Subgrid definition

- The model grid is subsampled to define the convolution subgrid following three steps:
  1. horizontal subsampling, level-independent,
  2. vertical subsampling, similar for all columns,
  3. horizontal subsampling, level-dependent.
- Each step takes the local convolution length-scales (horizontal or vertical) into account.
- The interpolation from the subgrid to the model grid is built backward from these three steps:

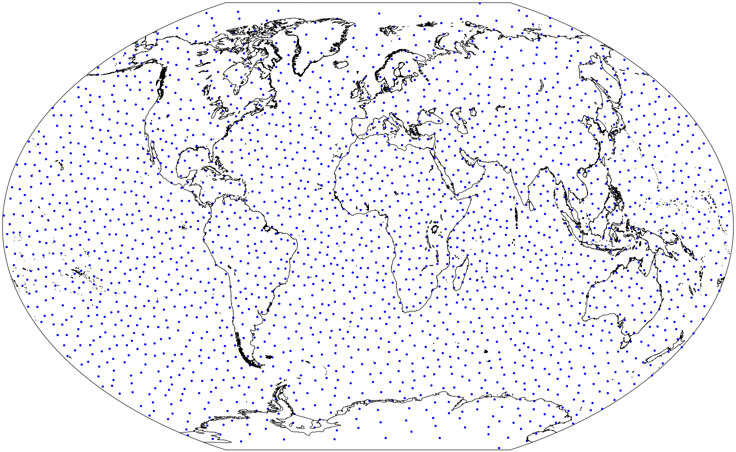
$$\underbrace{\mathbf{S}}_{\text{Total interpolation}} = [\text{model grid}] \underbrace{\mathbf{S}^h}_{\text{Horizontal level-independent}} \underbrace{\mathbf{S}^v}_{\text{Vertical}} \underbrace{\mathbf{S}^s}_{\text{Horizontal level-dependent}} [\text{subgrid}]$$

Subgrid definition

- The model grid is subsampled to define the convolution subgrid following three steps:
  1. horizontal subsampling, level-independent,
  2. vertical subsampling, similar for all columns,
  3. horizontal subsampling, level-dependent.
- Each step takes the local convolution length-scales (horizontal or vertical) into account.
- The interpolation from the subgrid to the model grid is built backward from these three steps:

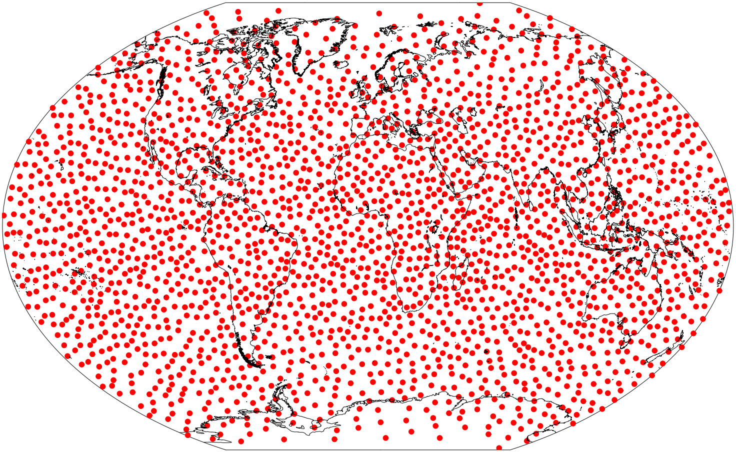
$$\underbrace{\mathbf{S}}_{\text{Total interpolation}} = [\text{model grid}] \underbrace{\mathbf{S}^h}_{\text{Horizontal level-independent}} \underbrace{\mathbf{S}^v}_{\text{Vertical}} \underbrace{\mathbf{S}^s}_{\text{Horizontal level-dependent}} [\text{subgrid}]$$

Horizontal grid definition



Blue dots: basic subset

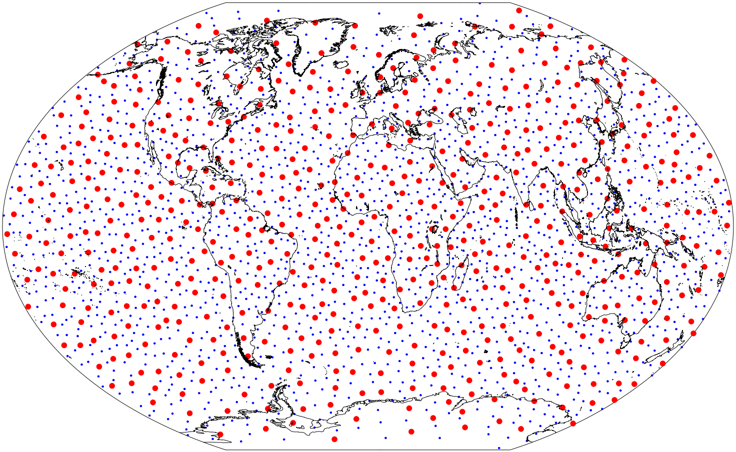
Horizontal grid definition



Blue dots: basic subset

Red dots: final subset with a short convolution length-scale

Horizontal grid definition

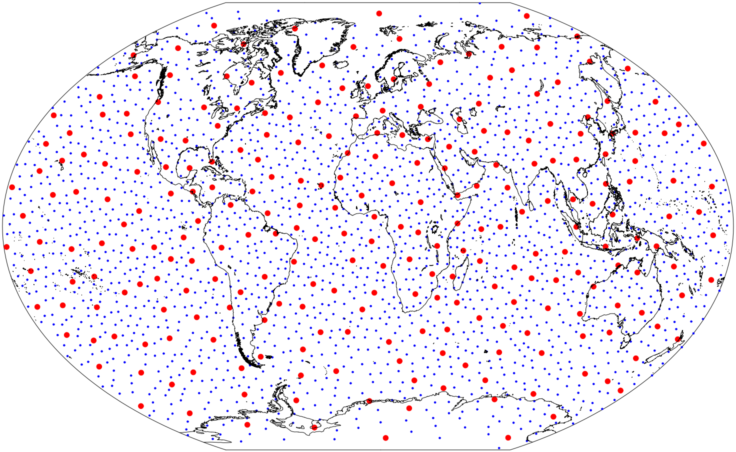


Blue dots: basic subset

Red dots: final subset with a medium convolution length-scale

METEO FRANCE

Horizontal grid definition



Blue dots: basic subset

Red dots: final subset with a large convolution length-scale

Outline

Principles

Subgrid definition

Convolution function

Parallelization

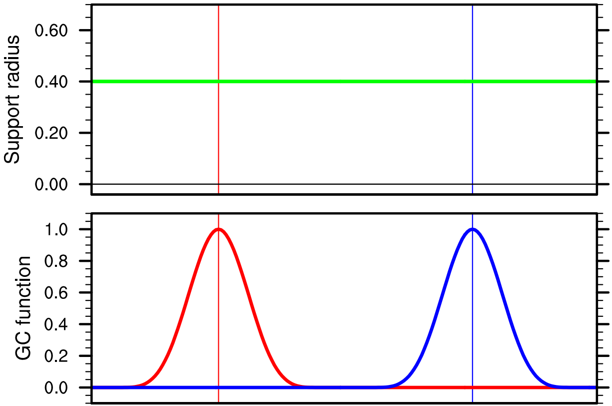
The BUMP software

Conclusions

Convolution function

Gaspari and Cohn (1999) function, global support radius r

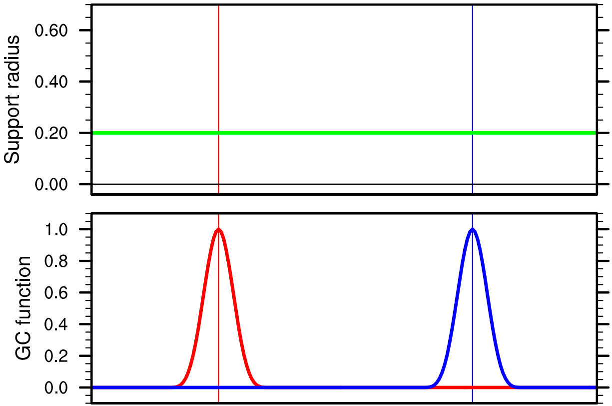
$\rightarrow$ homogeneous normalized distance $d'_{ij} = \frac{d_{ij}}{r}$



Convolution function

Gaspari and Cohn (1999) function, global support radius r

$\rightarrow$ homogeneous normalized distance $d'_{ij} = \frac{d_{ij}}{r}$

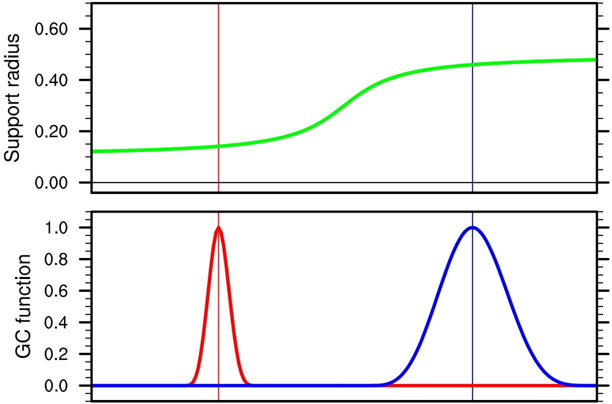


METEO FRANCE

Convolution function

Gaspari and Cohn (1999) function, local support radius **r**

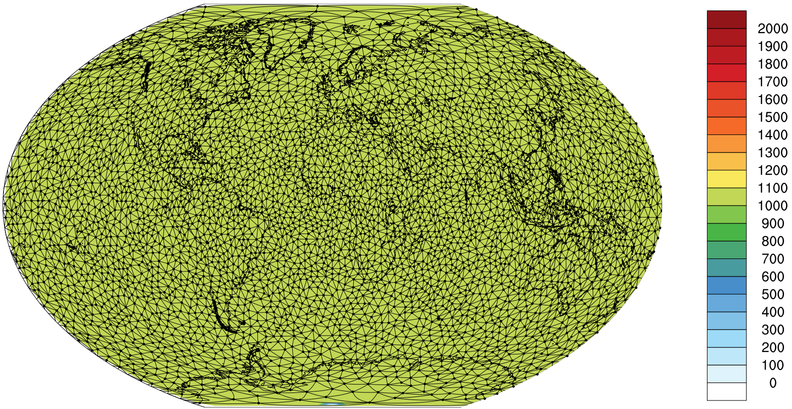
$\rightarrow$ heterogeneous normalized distance $d'_{ij} = \frac{d_{ij}}{\sqrt{(r_i^2 + r_j^2)/2}}$



METEO FRANCE

Length-scale and mesh density

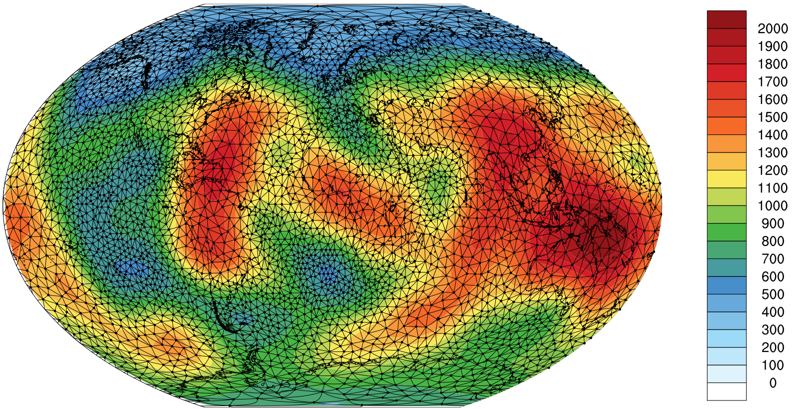
Homogeneous convolution length-scale → homogenous subgrid:



A fast trial-and-error algorithm using a K-D tree ensures that the horizontal subsampling is well distributed.

Length-scale and mesh density

Heterogenous convolution length-scale → heterogenous subgrid:

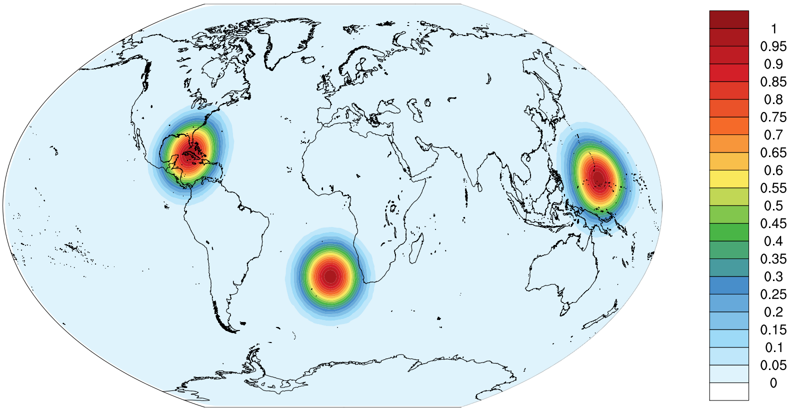


A fast trial-and-error algorithm using a K-D tree ensures that the horizontal subsampling is well distributed.

METEO FRANCE

Length-scale and mesh density

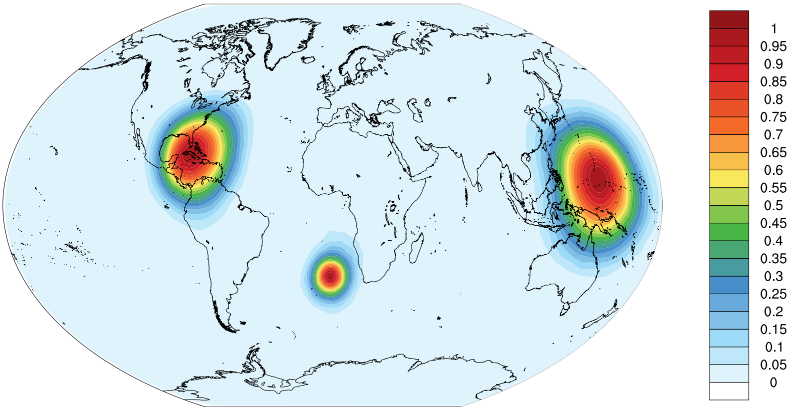
Convolution with a homogenous length-scale



METEO FRANCE

Length-scale and mesh density

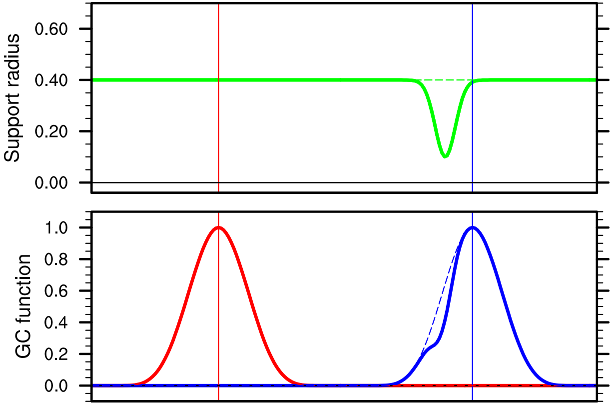
Convolution with a heterogeneous length-scale



Sharp convolution length-scale gradients

Gaspari and Cohn (1999) function, local support radius **r**

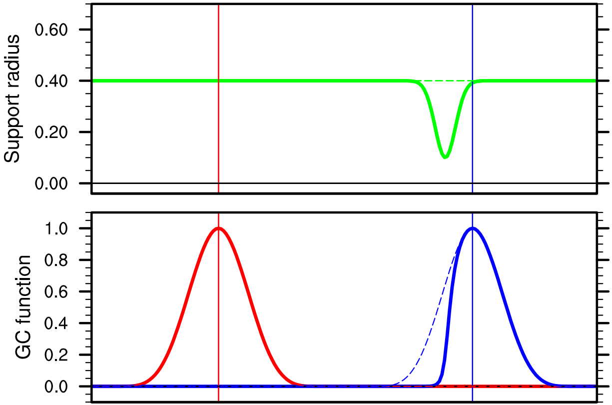
$\rightarrow$ heterogeneous normalized distance $d'_{ij} = \frac{d_{ij}}{\sqrt{(r_i^2 + r_j^2)/2}}$



Sharp convolution length-scale gradients

Gaspari and Cohn (1999) function, local support radius **r**

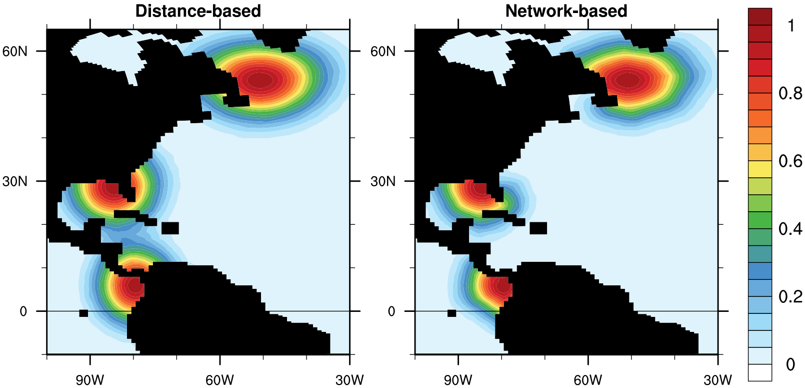
$\rightarrow$ heterogeneous normalized distance $\widetilde{d}'_{ij} = \sum_{k=i}^{j-1} d'_{k,k+1}$ (network)



Sharp convolution length-scale gradients

Convolution functions with complex boundaries:

- distance-based approach (left)
- network-based approach (right)

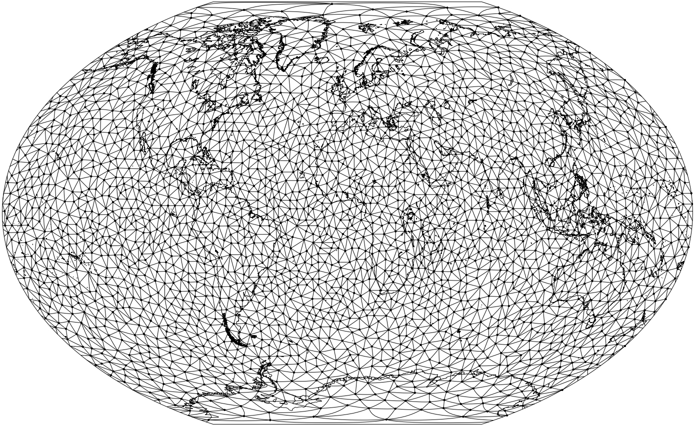


NICAS is exactly normalized for both approaches.

METEO FRANCE

Subgrid resolution

The subgrid resolution ρ is defined as the number of points required to describe half the Gaspari and Cohn (1999) function.

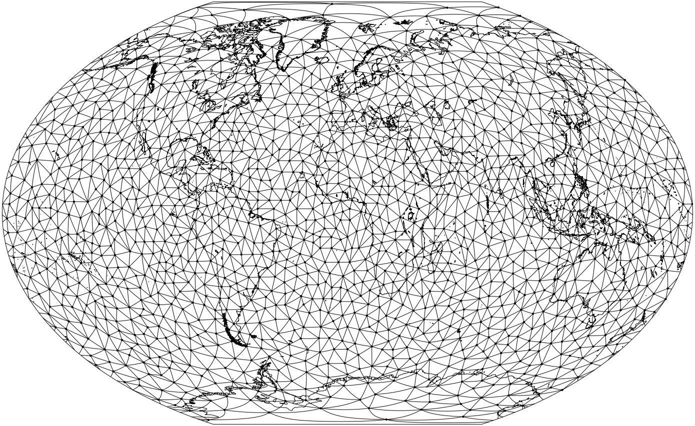


$\rho = 8$ (2827 points)

METEO FRANCE

Subgrid resolution

The subgrid resolution ρ is defined as the number of points required to describe half the Gaspari and Cohn (1999) function.

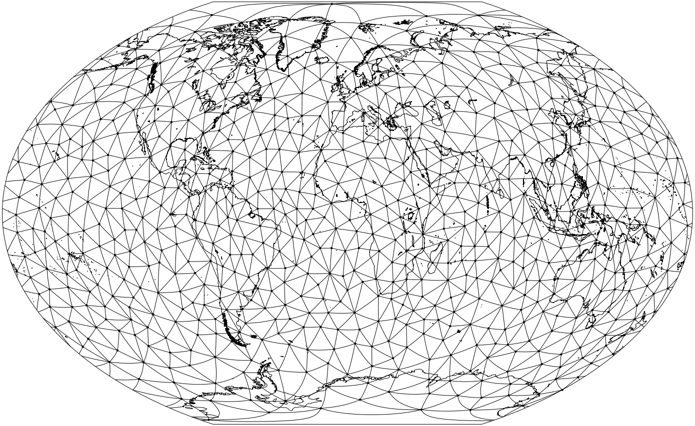


$\rho = 6$ (1590 points)

METEO FRANCE

Subgrid resolution

The subgrid resolution ρ is defined as the number of points required to describe half the Gaspari and Cohn (1999) function.

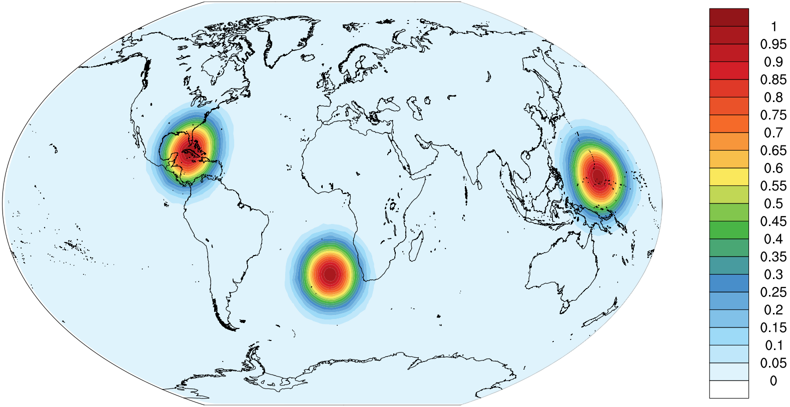


$\rho = 4$ (706 points)

METEO FRANCE

Subgrid resolution

The subgrid resolution ρ is defined as the number of points required to describe half the Gaspari and Cohn (1999) function.

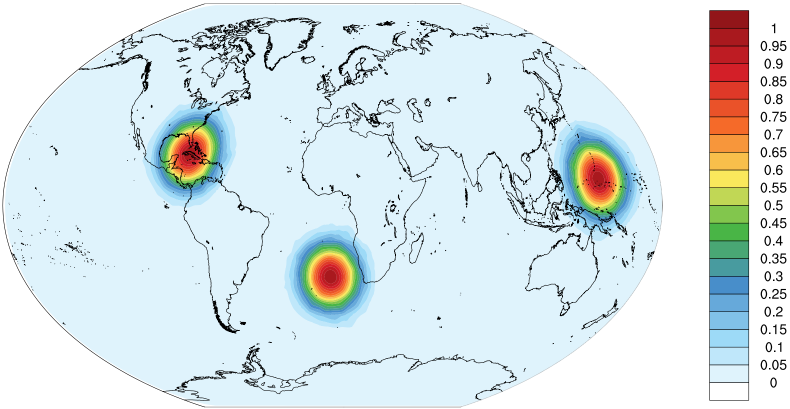


$\rho = 8$ (2827 points)

METEO FRANCE

Subgrid resolution

The subgrid resolution ρ is defined as the number of points required to describe half the Gaspari and Cohn (1999) function.

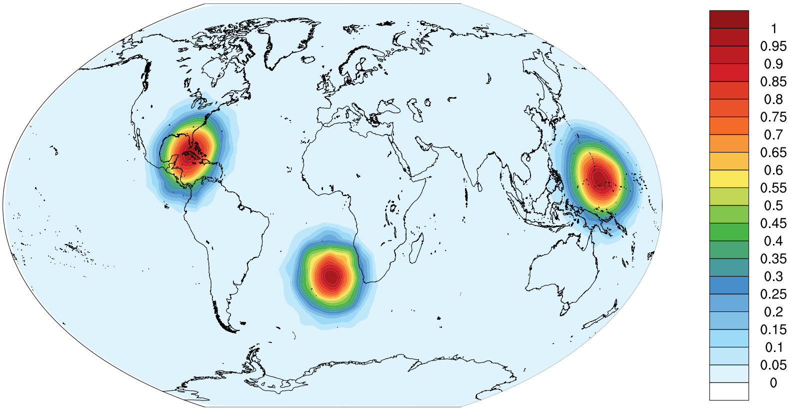


$\rho = 6$ (1590 points)

METEO FRANCE

Subgrid resolution

The subgrid resolution ρ is defined as the number of points required to describe half the Gaspari and Cohn (1999) function.



$\rho = 4$ (706 points)

METEO FRANCE

Square-root formulation

- Basic **NICAS** method:

$$\widetilde{\mathbf{C}} = \mathbf{N}\mathbf{S}\mathbf{C}^s\mathbf{S}^\mathrm{T}\mathbf{N}^\mathrm{T}$$

- If $\mathbf{C}^s$ is built as $\mathbf{U}^s\mathbf{U}^{s\mathrm{T}}$, then the square-root of $\widetilde{\mathbf{C}}$ is given by:

$$\widetilde{\mathbf{U}} = \mathbf{N}\mathbf{S}\mathbf{U}^s$$

which can be useful for square-root preconditioning in EnVar minimizations.

- Using the formulation:

$$\widetilde{\mathbf{C}} = \mathbf{N}\mathbf{S}\mathbf{U}^s\mathbf{U}^{s\mathrm{T}}\mathbf{S}^\mathrm{T}\mathbf{N}^\mathrm{T}$$

also ensures that $\widetilde{\mathbf{C}}$ is positive-semidefinite.

- A good approximation of the Gaspari and Cohn (1999) function square-root can be obtained by multiplying the function length-scale by 0.721 (empirical value).

METEO FRANCE

Square-root formulation

- Basic **NICAS** method:

$$\widetilde{\mathbf{C}} = \mathbf{NSC}^s\mathbf{S}^\mathrm{T}\mathbf{N}^\mathrm{T}$$

- If $\mathbf{C}^s$ is built as $\mathbf{U}^s\mathbf{U}^{s\mathrm{T}}$, then the square-root of $\widetilde{\mathbf{C}}$ is given by:

$$\widetilde{\mathbf{U}} = \mathbf{NSU}^s$$

which can be useful for square-root preconditioning in EnVar minimizations.

- Using the formulation:

$$\widetilde{\mathbf{C}} = \mathbf{NSU}^s\mathbf{U}^{s\mathrm{T}}\mathbf{S}^\mathrm{T}\mathbf{N}^\mathrm{T}$$

also ensures that $\widetilde{\mathbf{C}}$ is positive-semidefinite.

- A good approximation of the Gaspari and Cohn (1999) function square-root can be obtained by multiplying the function length-scale by 0.721 (empirical value).

METEO FRANCE

Square-root formulation

- Basic **NICAS** method:

$$\widetilde{\mathbf{C}} = \mathbf{N}\mathbf{S}\mathbf{C}^s\mathbf{S}^\mathrm{T}\mathbf{N}^\mathrm{T}$$

- If $\mathbf{C}^s$ is built as $\mathbf{U}^s\mathbf{U}^{s\mathrm{T}}$, then the square-root of $\widetilde{\mathbf{C}}$ is given by:

$$\widetilde{\mathbf{U}} = \mathbf{N}\mathbf{S}\mathbf{U}^s$$

which can be useful for square-root preconditioning in EnVar minimizations.

- Using the formulation:

$$\widetilde{\mathbf{C}} = \mathbf{N}\mathbf{S}\mathbf{U}^s\mathbf{U}^{s\mathrm{T}}\mathbf{S}^\mathrm{T}\mathbf{N}^\mathrm{T}$$

also ensures that $\widetilde{\mathbf{C}}$ is positive-semidefinite.

- A good approximation of the Gaspari and Cohn (1999) function square-root can be obtained by multiplying the function length-scale by 0.721 (empirical value).

Square-root formulation

- Basic **NICAS** method:

$$\widetilde{\mathbf{C}} = \mathbf{N}\mathbf{S}\mathbf{C}^s\mathbf{S}^\mathrm{T}\mathbf{N}^\mathrm{T}$$

- If $\mathbf{C}^s$ is built as $\mathbf{U}^s\mathbf{U}^{s\mathrm{T}}$, then the square-root of $\widetilde{\mathbf{C}}$ is given by:

$$\widetilde{\mathbf{U}} = \mathbf{N}\mathbf{S}\mathbf{U}^s$$

  which can be useful for square-root preconditioning in EnVar minimizations.

- Using the formulation:

$$\widetilde{\mathbf{C}} = \mathbf{N}\mathbf{S}\mathbf{U}^s\mathbf{U}^{s\mathrm{T}}\mathbf{S}^\mathrm{T}\mathbf{N}^\mathrm{T}$$

  also ensures that $\widetilde{\mathbf{C}}$ is positive-semidefinite.

- A good approximation of the Gaspari and Cohn (1999) function square-root can be obtained by multiplying the function length-scale by 0.721 (empirical value).

METEO FRANCE

Outline

Principles

Subgrid definition

Convolution function

Parallelization

The BUMP software

Conclusions

METEO FRANCE

MPI communications

Running **NICAS** with several MPI tasks:

- Communications are always performed **on the subgrid**, never on the model grid.
- Only **local** communications between halos are required, no global communications.
- **NICAS** can be applied with 1, 2 or 3 communication steps:

$$\widetilde{\mathbf{C}} = \mathbf{N}\mathbf{S} \boxtimes \mathbf{U}^s \boxtimes \mathbf{U}^{s\mathrm{T}} \boxtimes \mathbf{S}^\mathrm{T}\mathbf{N}^\mathrm{T}$$

  More communication steps $\Rightarrow$ smaller halos.
- Hybrid parallization with OpenMP is used to improve efficiency.

METEO FRANCE

MPI communications

Running **NICAS** with several MPI tasks:

- Communications are always performed **on the subgrid**, never on the model grid.
- Only **local** communications between halos are required, no global communications.
- **NICAS** can be applied with 1, 2 or 3 communication steps:

$$\widetilde{C} = \mathbf{N}\mathbf{S} \boxtimes \mathbf{U}^s \boxtimes \mathbf{U}^{s\mathrm{T}} \boxtimes \mathbf{S}^{\mathrm{T}}\mathbf{N}^{\mathrm{T}}$$

More communication steps $\Rightarrow$ smaller halos.

- Hybrid parallization with OpenMP is used to improve efficiency.

METEO FRANCE

MPI communications

Running **NICAS** with several MPI tasks:

- Communications are always performed **on the subgrid**, never on the model grid.
- Only **local** communications between halos are required, no global communications.
- **NICAS** can be applied with 1, 2 or 3 communication steps:

$$\widetilde{\mathbf{C}} = \mathbf{NS} \boxtimes \mathbf{U}^s \boxtimes \mathbf{U}^{s\mathrm{T}} \boxtimes \mathbf{S}^{\mathrm{T}}\mathbf{N}^{\mathrm{T}}$$

  More communication steps ⇒ smaller halos.

- Hybrid parallization with OpenMP is used to improve efficiency.

MPI communications

Running **NICAS** with several MPI tasks:

- Communications are always performed **on the subgrid**, never on the model grid.
- Only **local** communications between halos are required, no global communications.
- **NICAS** can be applied with 1, 2 or 3 communication steps:

$$\widetilde{\mathbf{C}} = \mathbf{NS} \boxtimes \mathbf{U}^s \boxtimes \mathbf{U}^{s\mathrm{T}} \boxtimes \mathbf{S}^{\mathrm{T}}\mathbf{N}^{\mathrm{T}}$$

  More communication steps ⇒ smaller halos.
- Hybrid parallization with OpenMP is used to improve efficiency.

MPI communications

Running **NICAS** with several MPI tasks:

- Communications are always performed **on the subgrid**, never on the model grid.
- Only **local** communications between halos are required, no global communications.
- **NICAS** can be applied with 1, 2 or 3 communication steps:

$$\widetilde{\mathbf{C}} = \mathbf{N}\mathbf{S} \boxtimes \mathbf{U}^s \boxtimes \mathbf{U}^{s\mathrm{T}} \boxtimes \mathbf{S}^\mathrm{T}\mathbf{N}^\mathrm{T}$$

  More communication steps $\Rightarrow$ smaller halos.
- Hybrid parallization with OpenMP is used to improve efficiency.

METEO FRANCE

MPI communications

Running **NICAS** with several MPI tasks:

- Communications are always performed **on the subgrid**, never on the model grid.
- Only **local** communications between halos are required, no global communications.
- **NICAS** can be applied with 1, 2 or 3 communication steps:

$$\widetilde{\mathbf{C}} = \mathbf{N}\mathbf{S} \boxtimes \mathbf{U}^s \boxtimes \mathbf{U}^{s\mathrm{T}} \boxtimes \mathbf{S}^{\mathrm{T}}\mathbf{N}^{\mathrm{T}}$$

  More communication steps $\Rightarrow$ smaller halos.

- Hybrid parallization with OpenMP is used to improve efficiency.

MPI communications

Running **NICAS** with several MPI tasks:

- Communications are always performed **on the subgrid**, never on the model grid.
- Only **local** communications between halos are required, no global communications.
- **NICAS** can be applied with 1, 2 or 3 communication steps:

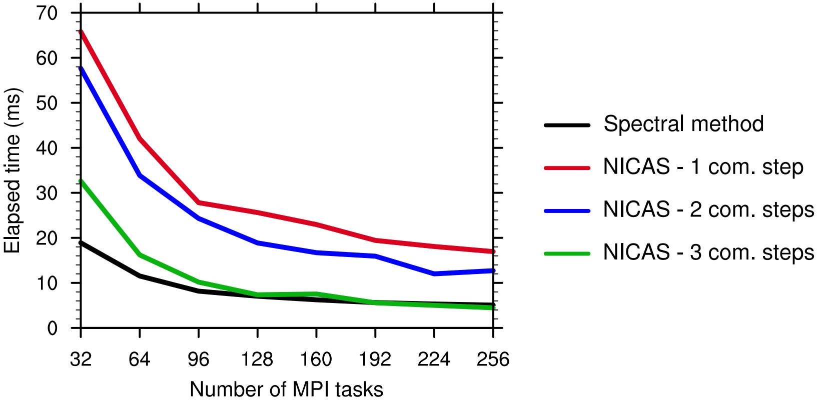
$$\widetilde{\mathbf{C}} = \mathbf{N}\mathbf{S} \boxtimes \mathbf{U}^s \boxtimes \mathbf{U}^{s\mathrm{T}} \boxtimes \mathbf{S}^{\mathrm{T}}\mathbf{N}^{\mathrm{T}}$$

  More communication steps $\Rightarrow$ smaller halos.
- Hybrid parallization with OpenMP is used to improve efficiency.

METEO FRANCE

Scaling

Comparison of the standard spectral method with **NICAS**:



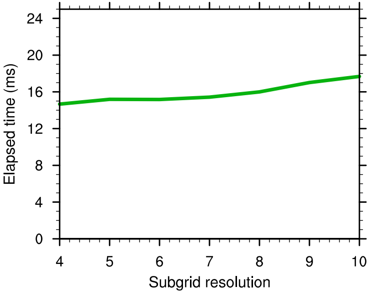
Elapsed time for one application of **NICAS** - ARPEGE (T399, L105)

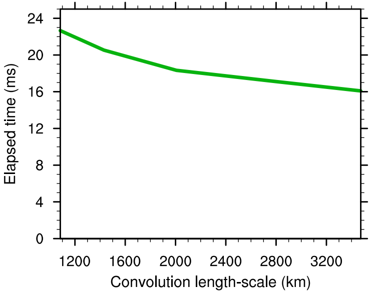
Elapsed time decreases for more communication steps.

METEO FRANCE

Subgrid resolution and length-scale impact

Preliminary tests show a slight sensitivity to the subgrid resolution and to the convolution length-scale:





Elapsed time for one application of **NICAS** - ARPEGE (T399, L105) - 64 MPI tasks

The computational cost increases for:

- a more precise description of the convolution function,
- a smaller convolution lenght-scale.

METEO FRANCE

Outline

METEO FRANCE

The **BUMP** software

BUMP : **B** matrix on an **U**nstructured **M**esh **P**ackage

- Capabilities:
  1. Covariance / correlation diagnostics
  2. Localization functions diagnostics [Ménétrier et al., 2015]
  3. Hybridization diagnostics [Ménétrier and Auligné, 2015]
  4. Local correlation tensors diagnostics
  5. Preparation and application of the **NICAS** method
- Object-oriented Fortran code $\sim$ 16.700 lines
- Two execution modes:
  - Offline : execution using a namelist and NetCDF input data
  - Inline: called from another code, via a generic interface
- Used as a research tool by scientists at: CERFACS, ECMWF, Météo-France, MetOffice, NASA, NCAR, NOAA (JCSDA)
- Open-source CeCILL-C license, code available at:

  `https://github.com/benjaminmenetrier/bump`

METEO FRANCE

The **BUMP** software

BUMP : **B** matrix on an **U**nstructured **M**esh **P**ackage

- Capabilities:
  1. Covariance / correlation diagnostics
  2. Localization functions diagnostics [Ménétrier et al., 2015]
  3. Hybridization diagnostics [Ménétrier and Auligné, 2015]
  4. Local correlation tensors diagnostics
  5. Preparation and application of the **NICAS** method
- Object-oriented Fortran code ∼ 16.700 lines
- Two execution modes:
  - Offline : execution using a namelist and NetCDF input data
  - Inline: called from another code, via a generic interface
- Used as a research tool by scientists at: CERFACS, ECMWF, Météo-France, MetOffice, NASA, NCAR, NOAA (JCSDA)
- Open-source CeCILL-C license, code available at:

  `https://github.com/benjaminmenetrier/bump`

The **BUMP** software

BUMP : **B** matrix on an **U**nstructured **M**esh **P**ackage

- Capabilities:
  1. Covariance / correlation diagnostics
  2. Localization functions diagnostics [Ménétrier et al., 2015]
  3. Hybridization diagnostics [Ménétrier and Auligné, 2015]
  4. Local correlation tensors diagnostics
  5. Preparation and application of the **NICAS** method
- Object-oriented Fortran code $\sim$ 16.700 lines
- Two execution modes:
  - Offline : execution using a namelist and NetCDF input data
  - Inline: called from another code, via a generic interface
- Used as a research tool by scientists at: CERFACS, ECMWF, Météo-France, MetOffice, NASA, NCAR, NOAA (JCSDA)
- Open-source CeCILL-C license, code available at:

  `https://github.com/benjaminmenetrier/bump`

The **BUMP** software

BUMP : **B** matrix on an **U**nstructured **M**esh **P**ackage

- Capabilities:
  1. Covariance / correlation diagnostics
  2. Localization functions diagnostics [Ménétrier et al., 2015]
  3. Hybridization diagnostics [Ménétrier and Auligné, 2015]
  4. Local correlation tensors diagnostics
  5. Preparation and application of the **NICAS** method
- Object-oriented Fortran code $\sim$ 16.700 lines
- Two execution modes:
  - Offline : execution using a namelist and NetCDF input data
  - Inline: called from another code, via a generic interface
- Used as a research tool by scientists at: CERFACS, ECMWF, Météo-France, MetOffice, NASA, NCAR, NOAA (JCSDA)
- Open-source CeCILL-C license, code available at:

  https://github.com/benjaminmenetrier/bump

METEO FRANCE

The **BUMP** software

BUMP : **B** matrix on an **U**nstructured **M**esh **P**ackage

- Capabilities:
  1. Covariance / correlation diagnostics
  2. Localization functions diagnostics [Ménétrier et al., 2015]
  3. Hybridization diagnostics [Ménétrier and Auligné, 2015]
  4. Local correlation tensors diagnostics
  5. Preparation and application of the **NICAS** method
- Object-oriented Fortran code $\sim$ 16.700 lines
- Two execution modes:
  - Offline : execution using a namelist and NetCDF input data
  - Inline: called from another code, via a generic interface
- Used as a research tool by scientists at: CERFACS, ECMWF, Météo-France, MetOffice, NASA, NCAR, NOAA (JCSDA)
- Open-source CeCILL-C license, code available at:

  `https://github.com/benjaminmenetrier/bump`

METEO FRANCE

Outline

Principles

Subgrid definition

Convolution function

Parallelization

The BUMP software

Conclusions

METEO FRANCE

Conclusions

- A new implementation of localization for EnVar applications has been developed: **NICAS**
- **NICAS** is heterogeneous and can deal with complex boundaries, yet it is exactly normalized
- **NICAS** speed is slightly sensitive to the subgrid resolution and convolution length-scale.
- **NICAS** is as fast as the standard spectral method for ARPEGE (T399, L105) if the number of MPI tasks is sufficient (> 100).
- The open-source software **BUMP** implementing **NICAS** is available online. It can be easily interfaced with other codes.

METEO FRANCE

Conclusions

- A new implementation of localization for EnVar applications has been developed: **NICAS**
- **NICAS** is heterogeneous and can deal with complex boundaries, yet it is exactly normalized
- **NICAS** speed is slightly sensitive to the subgrid resolution and convolution length-scale.
- **NICAS** is as fast as the standard spectral method for ARPEGE (T399, L105) if the number of MPI tasks is sufficient (> 100).
- The open-source software **BUMP** implementing **NICAS** is available online. It can be easily interfaced with other codes.

Conclusions

- A new implementation of localization for EnVar applications has been developed: **NICAS**
- **NICAS** is heterogeneous and can deal with complex boundaries, yet it is exactly normalized
- **NICAS** speed is slightly sensitive to the subgrid resolution and convolution length-scale.
- **NICAS** is as fast as the standard spectral method for ARPEGE (T399, L105) if the number of MPI tasks is sufficient (> 100).
- The open-source software **BUMP** implementing **NICAS** is available online. It can be easily interfaced with other codes.

Conclusions

- A new implementation of localization for EnVar applications has been developed: **NICAS**
- **NICAS** is heterogeneous and can deal with complex boundaries, yet it is exactly normalized
- **NICAS** speed is slightly sensitive to the subgrid resolution and convolution length-scale.
- **NICAS** is as fast as the standard spectral method for ARPEGE (T399, L105) if the number of MPI tasks is sufficient (> 100).
- The open-source software **BUMP** implementing **NICAS** is available online. It can be easily interfaced with other codes.

Conclusions

- A new implementation of localization for EnVar applications has been developed: **NICAS**
- **NICAS** is heterogeneous and can deal with complex boundaries, yet it is exactly normalized
- **NICAS** speed is slightly sensitive to the subgrid resolution and convolution length-scale.
- **NICAS** is as fast as the standard spectral method for ARPEGE (T399, L105) if the number of MPI tasks is sufficient (> 100).
- The open-source software **BUMP** implementing **NICAS** is available online. It can be easily interfaced with other codes.

Conclusions

- A new implementation of localization for EnVar applications has been developed: **NICAS**
- **NICAS** is heterogeneous and can deal with complex boundaries, yet it is exactly normalized
- **NICAS** speed is slightly sensitive to the subgrid resolution and convolution length-scale.
- **NICAS** is as fast as the standard spectral method for ARPEGE (T399, L105) if the number of MPI tasks is sufficient (>100).
- The open-source software **BUMP** implementing **NICAS** is available online. It can be easily interfaced with other codes.

Thank you for your attention

METEO FRANCE

Conclusions

- A new implementation of localization for EnVar applications has been developed: **NICAS**
- **NICAS** is heterogeneous and can deal with complex boundaries, yet it is exactly normalized
- **NICAS** speed is slightly sensitive to the subgrid resolution and convolution length-scale.
- **NICAS** is as fast as the standard spectral method for ARPEGE (T399, L105) if the number of MPI tasks is sufficient (> 100).
- The open-source software **BUMP** implementing **NICAS** is available online. It can be easily interfaced with other codes.

Thank you for your attention

METEO FRANCE

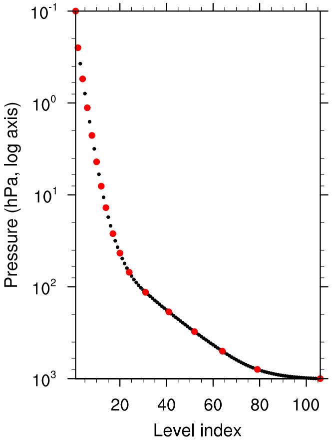
Conclusions

- A new implementation of localization for EnVar applications has been developed: **NICAS**
- **NICAS** is heterogeneous and can deal with complex boundaries, yet it is exactly normalized
- **NICAS** speed is slightly sensitive to the subgrid resolution and convolution length-scale.
- **NICAS** is as fast as the standard spectral method for ARPEGE (T399, L105) if the number of MPI tasks is sufficient (> 100).
- The open-source software **BUMP** implementing **NICAS** is available online. It can be easily interfaced with other codes.

Thank you for your attention

Vertical grid definition

Levels are subsampled depending on the vertical convolution length-scale:



Black dots: model levels - red dots: subgrid levels

Normalization computation

Normalization coefficient:

$$N_{ii} = \left(\delta_i^{\mathrm{T}} \mathbf{S} \mathbf{U}^s \mathbf{U}^{s\mathrm{T}} \mathbf{S}^{\mathrm{T}} \delta_i\right)^{-1/2}$$

$$= \|\mathbf{U}^{s\mathrm{T}} \mathbf{S}^{\mathrm{T}} \delta_i\|^{-1}$$

where δ_i is a Dirac vector (1 at point i, 0 elsewhere).

- Brute force computation: full computation of $\mathbf{U}^{s\mathrm{T}} \mathbf{S}^{\mathrm{T}} \delta_i$ for every model grid point i.
  $\rightarrow$ prohibitive cost $\sim O(n^2)$
- Efficient computation: exact determination of the subgrid nodes involved in the computation of $\mathbf{U}^{s\mathrm{T}} \mathbf{S}^{\mathrm{T}} \delta_i$, allowing for a fast computation (number of involved nodes $\ll n^s$).
  $\rightarrow$ affordable cost $\sim O(n)$

Normalization computation

Normalization coefficient:

$$N_{ii} = \left(\delta_i^\mathrm{T} \mathbf{S} \mathbf{U}^s \mathbf{U}^{s\mathrm{T}} \mathbf{S}^\mathrm{T} \delta_i\right)^{-1/2}$$
$$= \|\mathbf{U}^{s\mathrm{T}} \mathbf{S}^\mathrm{T} \delta_i\|^{-1}$$

where δ_i is a Dirac vector (1 at point i, 0 elsewhere).

- Brute force computation: full computation of $\mathbf{U}^{s\mathrm{T}} \mathbf{S}^\mathrm{T} \delta_i$ for every model grid point i.
  $\rightarrow$ prohibitive cost $\sim O(n^2)$
- Efficient computation: exact determination of the subgrid nodes involved in the computation of $\mathbf{U}^{s\mathrm{T}} \mathbf{S}^\mathrm{T} \delta_i$, allowing for a fast computation (number of involved nodes $\ll n^s$).
  $\rightarrow$ affordable cost $\sim O(n)$

Normalization computation

Normalization coefficient:

$$N_{ii} = \left(\boldsymbol{\delta}_i^{\mathrm{T}} \mathbf{S} \mathbf{U}^s \mathbf{U}^{s\mathrm{T}} \mathbf{S}^{\mathrm{T}} \boldsymbol{\delta}_i\right)^{-1/2}$$
$$= \|\mathbf{U}^{s\mathrm{T}} \mathbf{S}^{\mathrm{T}} \boldsymbol{\delta}_i\|^{-1}$$

where $\boldsymbol{\delta}_i$ is a Dirac vector (1 at point i, 0 elsewhere).

- Brute force computation: full computation of $\mathbf{U}^{s\mathrm{T}} \mathbf{S}^{\mathrm{T}} \boldsymbol{\delta}_i$ for every model grid point i.
  $\rightarrow$ prohibitive cost $\sim O(n^2)$
- Efficient computation: exact determination of the subgrid nodes involved in the computation of $\mathbf{U}^{s\mathrm{T}} \mathbf{S}^{\mathrm{T}} \boldsymbol{\delta}_i$, allowing for a fast computation (number of involved nodes $\ll n^s$).
  $\rightarrow$ affordable cost $\sim O(n)$