BUMP, a generic tool for background error covariance modeling

Benjamin Menetrier, IRIT, Toulouse, France
benjamin.menetrier@irit.fr
Funded by the JCSDA

6th JEDI Academy - June 2021

Outline

Outline

What is B?

In the variational framework, the background error covariance matrix B describes the uncertainty of the background x^b:

$$\mathcal{J}(\mathrm{x}) = \underbrace{\left(\mathrm{x}-\mathrm{x}^b\right)^{\mathrm{T}} \mathrm{B}^{-1} \left(\mathrm{x}-\mathrm{x}^b\right)}_{\mathcal{J}_b(\mathrm{x})} + \underbrace{\left(\mathrm{y}^o - \mathcal{H}(\mathrm{x})\right)^{\mathrm{T}} \mathrm{R}^{-1} \left(\mathrm{y}^o - \mathcal{H}(\mathrm{x})\right)}_{\mathcal{J}_o(\mathrm{x})}$$

In the equivalent BLUE framework, B is the last operator applied to define the analysis increment:

$$\mathrm{x}^a = \mathrm{x}^b + \mathrm{B}\mathrm{H}^{\mathrm{T}} \left(\mathrm{H}\mathrm{B}\mathrm{H}^{\mathrm{T}} + \mathrm{R}\right)^{-1} \left(\mathrm{y}^o - \mathcal{H}\left(\mathrm{x}^b\right)\right)$$

B contains complex multivariate 3D structures (potentially 4D for the 4DEnVar algorithm).

Basic facts

- Background error covariance (aka B matrix) is a key aspect of variational DA systems.
- In the practical implementation, the B matrix itself is not required, only its effect on an increment vector.
- B must be symmetric and positive, so we generally build its square-root U: $\mathrm{B} = \mathrm{U}\mathrm{U}^{\mathrm{T}}$.
- Usual covariance models are:
  - static B: a sequence of parametrized operators
  - ensemble-based B: a localized sample covariance matrix
  - hybrid B: a linear combination of previous models
- All covariance models require a smoother to spread the innovation information. In most implementations, smoothers are grid-specific (e.g. spectral transform, recursive filters).

Hybrid background error modeling

Hybrid background error covariance matrix:

$$\mathbf{B} = \underbrace{\beta^{e2}\,\mathbf{L}\circ\widetilde{\mathbf{B}}}_{\text{“ensemble” term}} + \underbrace{\beta^{s2}\,\mathbf{B}^{s}}_{\text{“static” term}}$$

where:

- $\widetilde{\mathbf{B}}$ is sampled from an ensemble of forecasts
- $\mathbf{L}$ is the localization matrix
- $\circ$ denotes a Schur product (element-by-element)
- β^{e} and β^{s} are scalar coefficients
- $\mathbf{B}^{s}$ is a “static” covariance matrix, usually modeled as:

$$\mathbf{B}^{s} = \mathbf{K}\boldsymbol{\Sigma}\mathbf{C}\boldsymbol{\Sigma}\mathbf{K}^{\mathrm{T}}$$

where:

- $\mathbf{K}$ is a balance operator
- $\boldsymbol{\Sigma}$ is the diagonal matrix of standard-deviations
- $\mathbf{C}$ is a correlation operator

BUMP overview

- BUMP stands for "Background error on an Unstructured Mesh Package".
- BUMP is one of the background error covariance libraries of SABER (System Agnostic Background Error Representation), a component of the JEDI project.
- BUMP works with any model grid and is able to take complex boundaries into account (important for ocean or land models).
- Written in modern Fortran 90 ($\sim$ 25.000 lines), the code can be easily called through Fortran and C++ interfaces.
- Interfaces with BUMP are implemented for most of JEDI models, via OOPS.

BUMP functionalities

BUMP provides tools based on ensembles of forecasts to estimate:

- Horizontal and vertical localization length-scales of $\mathbf{L}$ and hybrid weights β^e and β^s, estimated locally using the theory of Ménétrier and Auligne (2015) [HDIAG].
- Vertical statistical regressions of the balance operator $\mathbf{K}$ estimated locally [VBAL].
- Standard-deviations $\boldsymbol{\Sigma}$ objectively filtered with the method of Ménétrier *et al.* (2015a,b) [VAR].
- Horizontal and vertical correlation length-scales of $\mathbf{C}$ estimated locally and simultaneously [HDIAG], or alternatively Local Correlation Tensors [LCT] for anisotropic functions.

A grid smoother is required to apply $\mathbf{L}$ and $\mathbf{C}$ in a variational framework: BUMP implements the NICAS method (Normalized Interpolated Convolution from an Adaptive Subgrid) [NICAS].

BUMP credo: subgrids

High-resolution grids are great for models dynamics, but very costly and not really useful for background error covariance modeling.

In general, the size of the background error structures that we can represent is significantly larger than the model grid cell size.

The leading ideas of BUMP are:

1 - Use of cleverly defined subgrids.

2 - All costly operations performed on the subgrids.

3 - Final results interpolated on the model grid.

Outline

Static B basics

- The static B is a robust and well-conditioned model, based on a sequence of parametrized operators.
- The parameters can be defined using ensemble data over a long period, giving a climatological estimate.
- They can also be estimated over a shorter sliding window, and updated at every cycle.
- The most common static B model is:

$$\mathbf{B}^s = \mathbf{K}\boldsymbol{\Sigma}\mathbf{C}\boldsymbol{\Sigma}\mathbf{K}^{\mathrm{T}}$$

where:

- $\mathbf{K}$ is a balance operator
- $\boldsymbol{\Sigma}$ is the diagonal matrix of standard-deviations
- $\mathbf{C}$ is a correlation operator

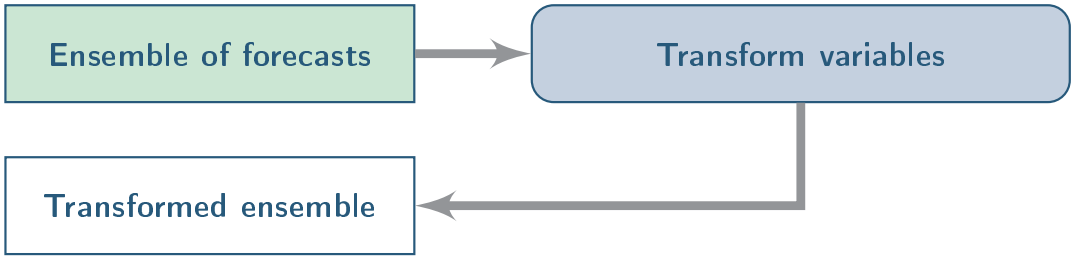
Static B with BUMP

BUMP can be used for all these operators:

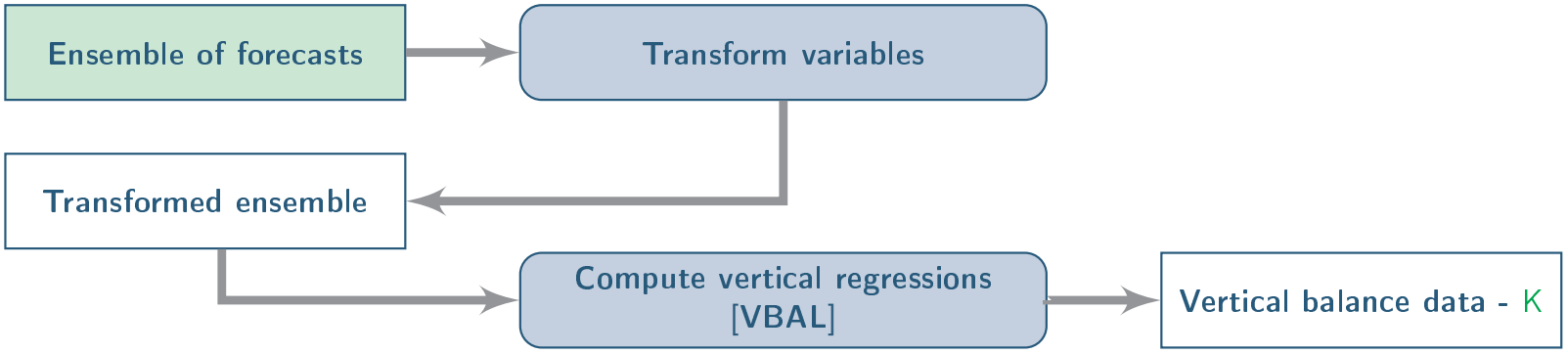
- As part of a more complex $\mathbf{K}$, vertical regressions can be estimated locally and applied.
- The standard-deviations in $\mathbf{\Sigma}$ are estimated locally and potentially filtered, to remove the sampling noise (objective filtering of Ménétrier *et al.*, 2015a,b).
- The correlation length-scales of $\mathbf{C}$ are estimated locally and used to set up the NICAS smoother, which is exactly normalized ($C_{ii} = 1$).

In the OOPS framework, the static B components can come from BUMP or from your own model: operators can be combined as you wish.

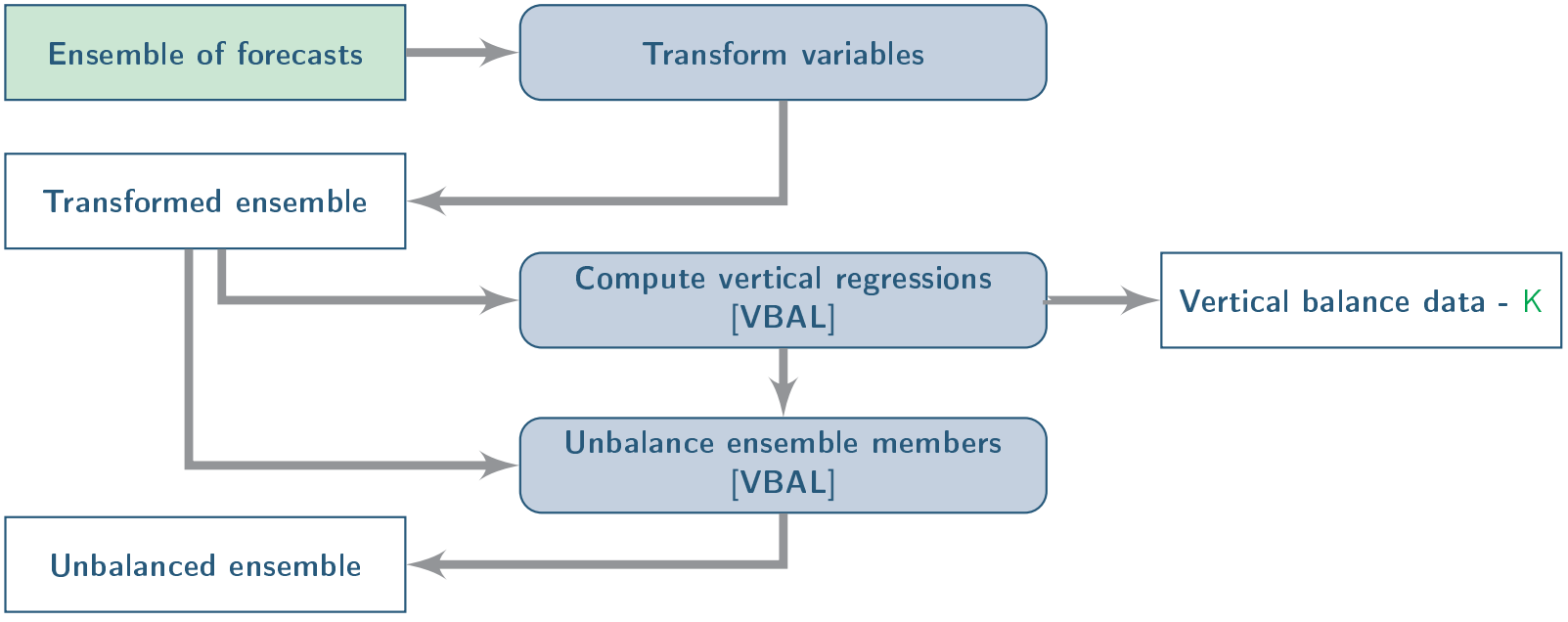
Static covariance training



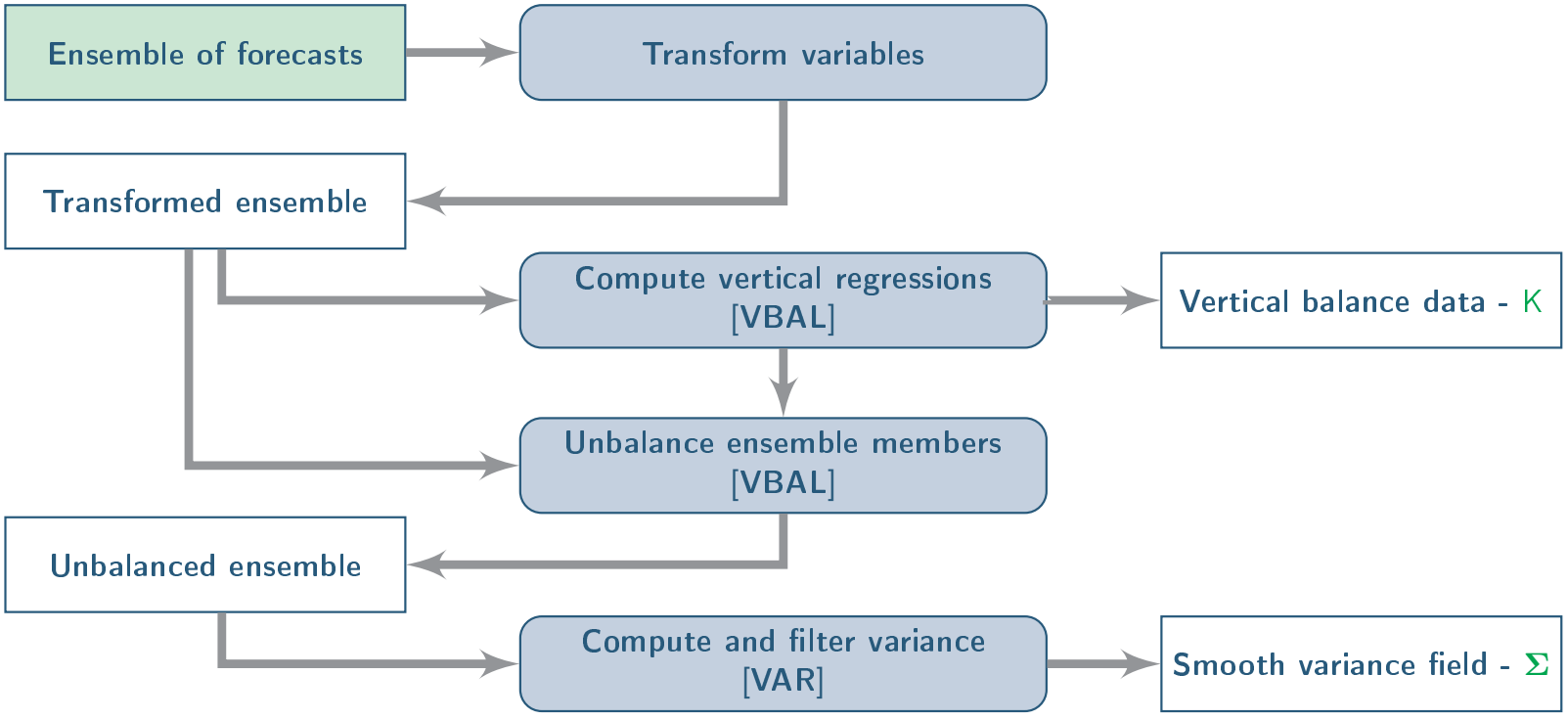
Static covariance training



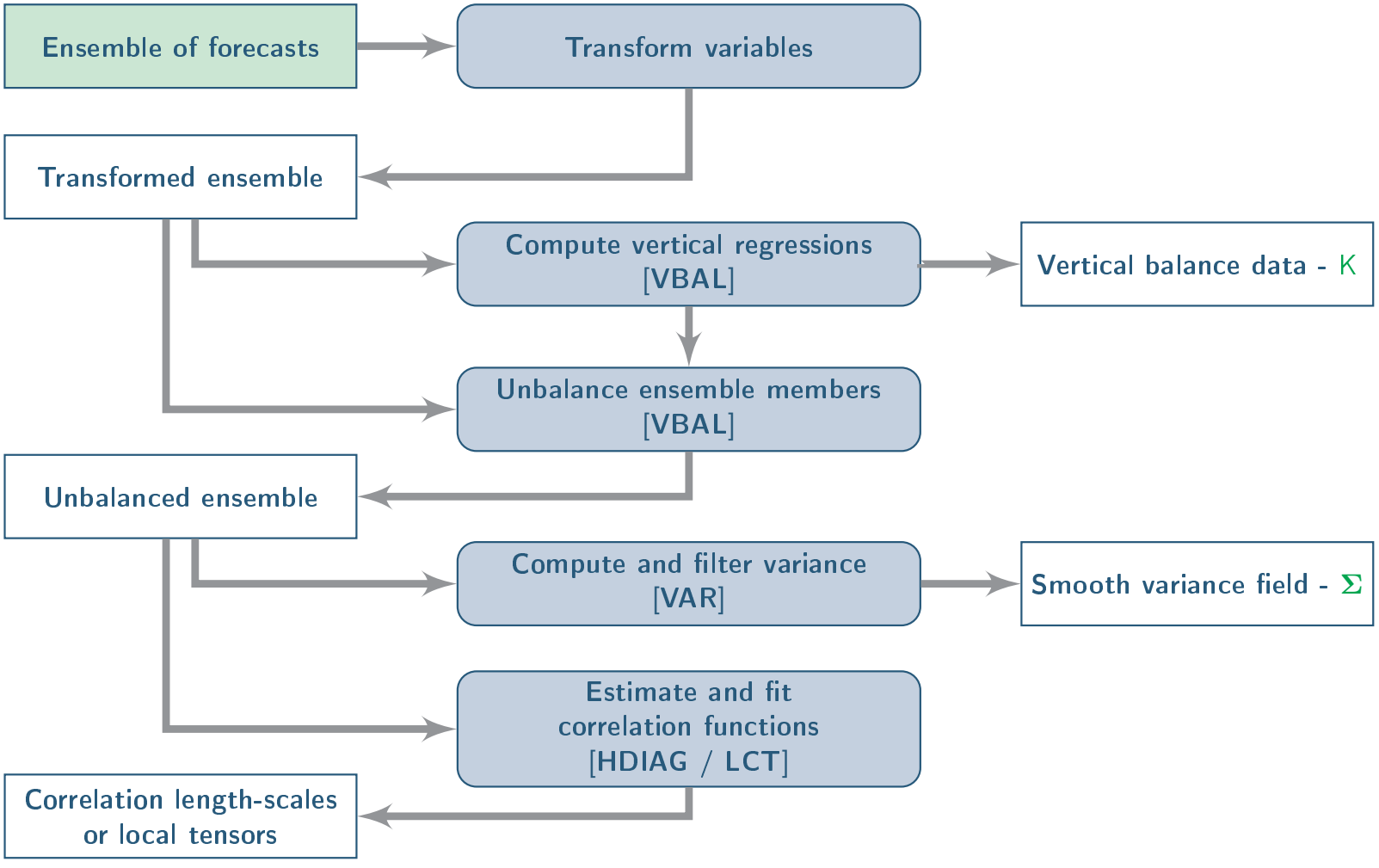
Static covariance training



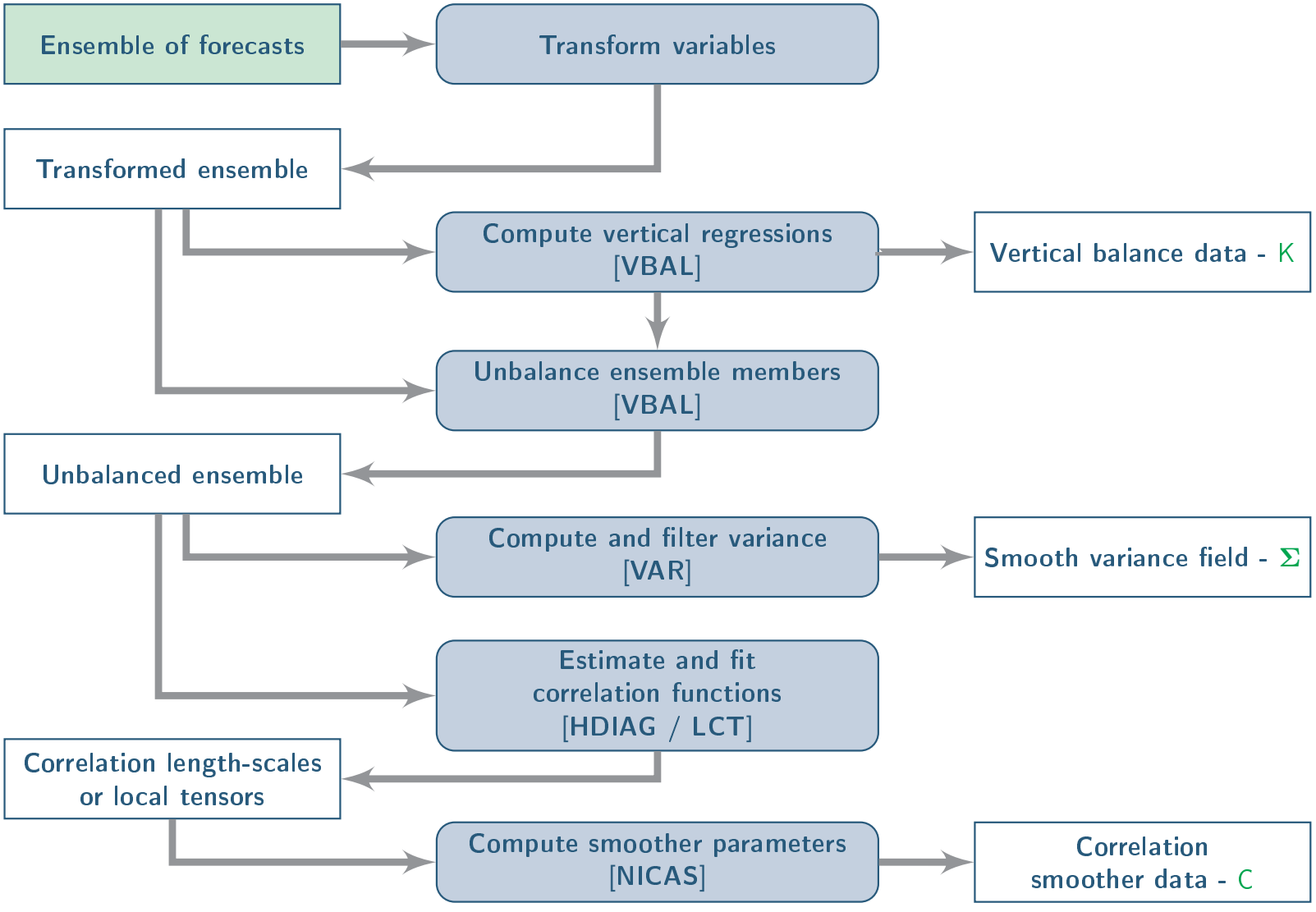
Static covariance training



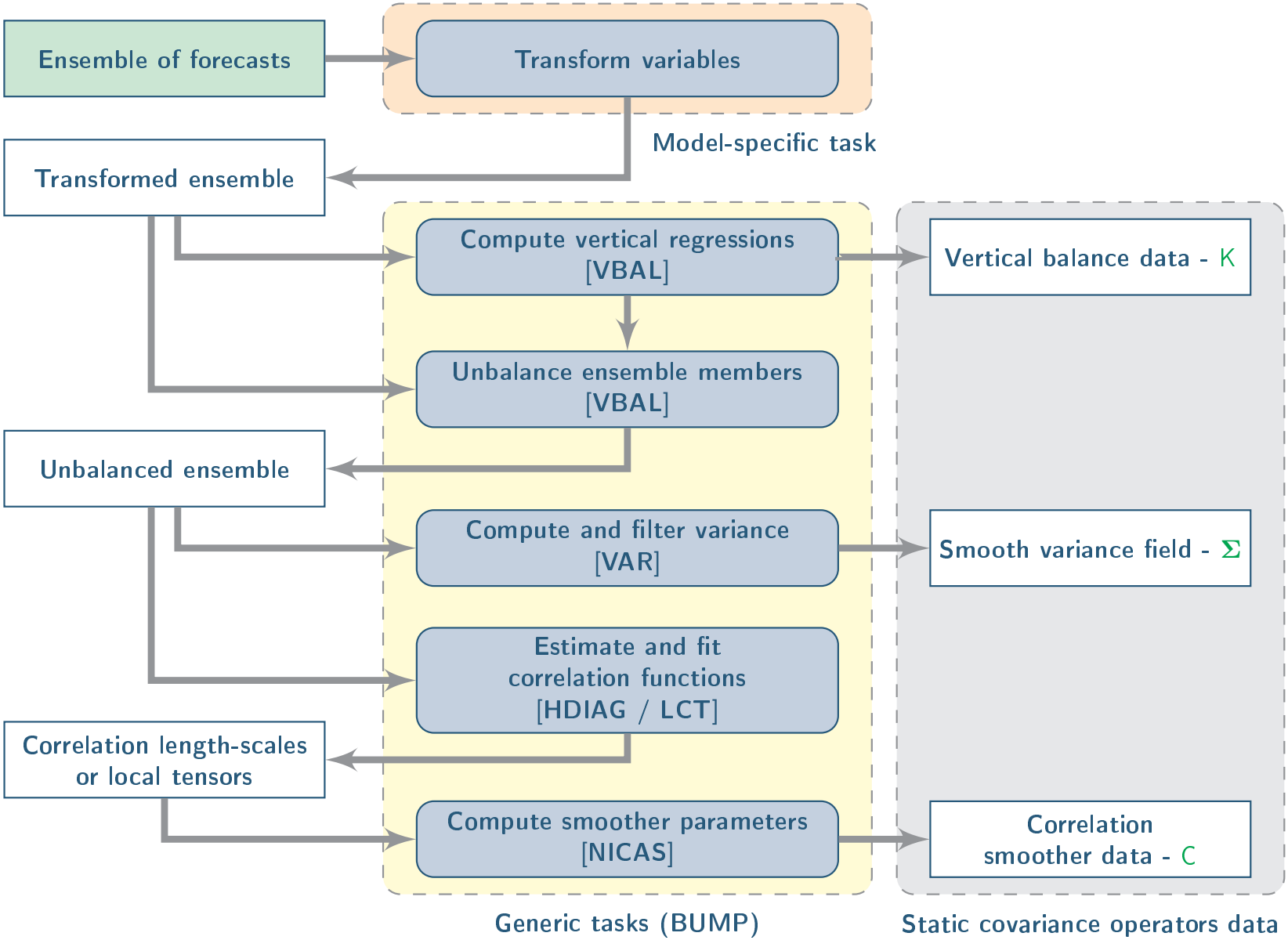
Static covariance training



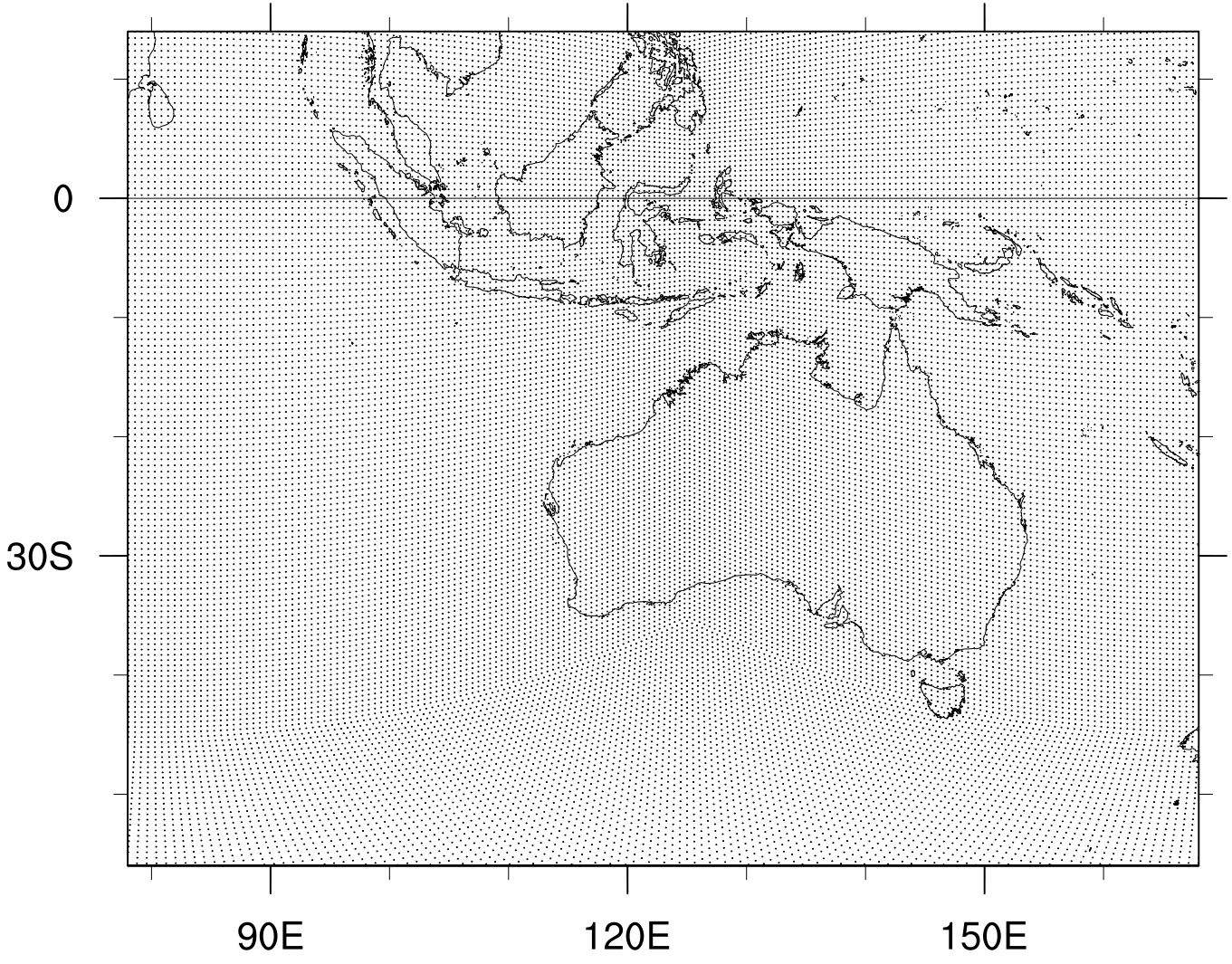
Static covariance training



Static covariance training

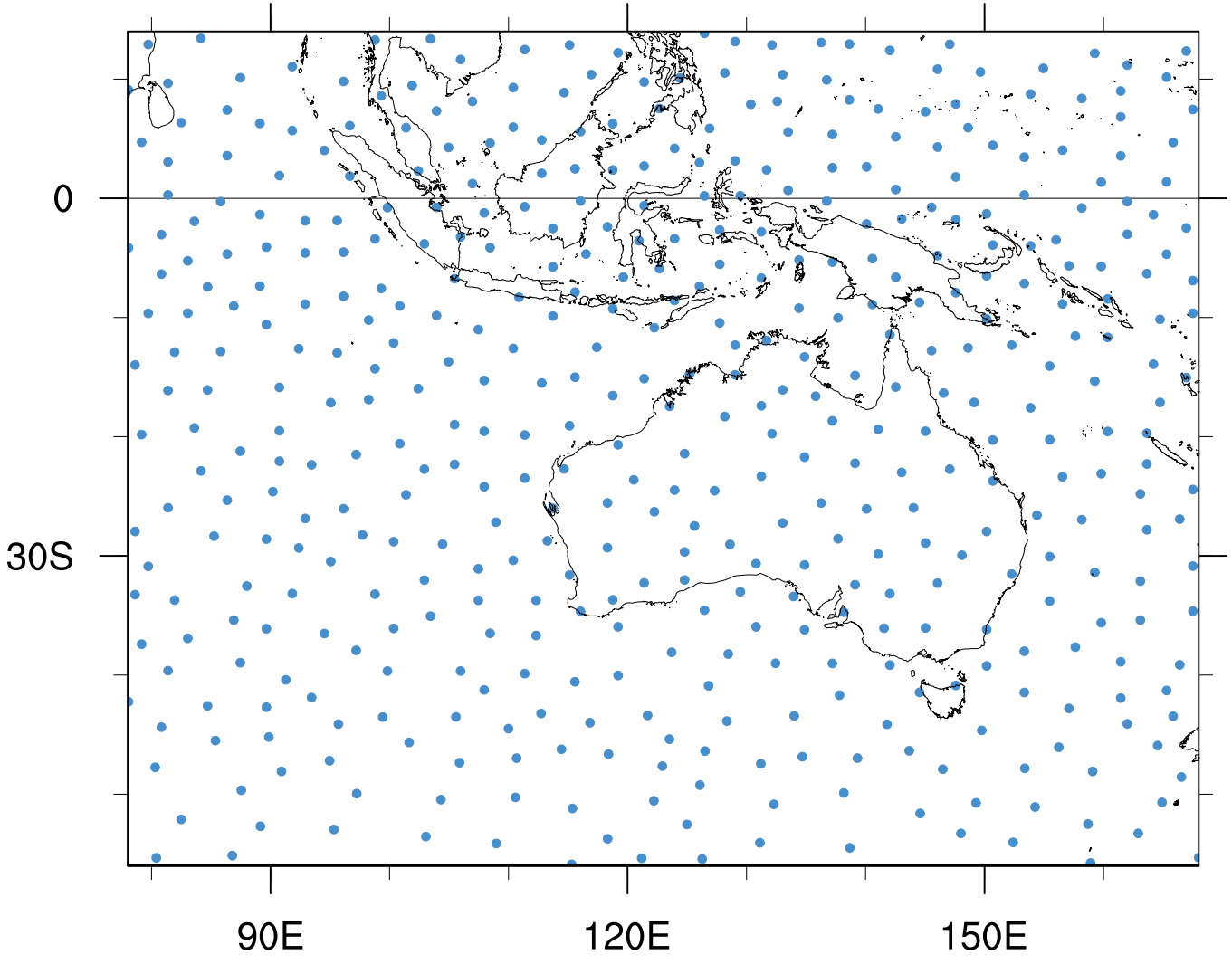


Full model grid



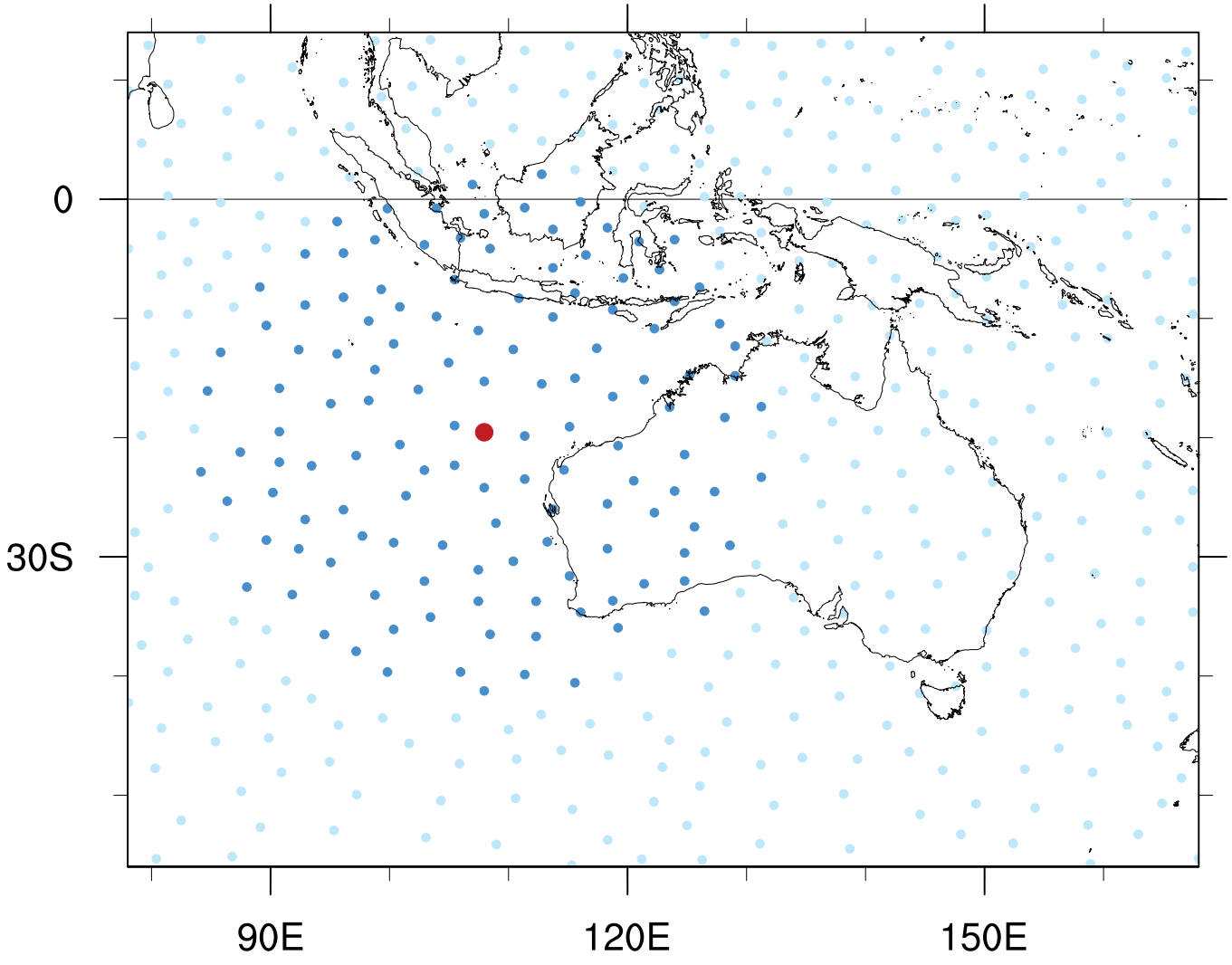
Full model grid (FV3 cubed-sphere grid - C180)

Local estimation of K with VBAL



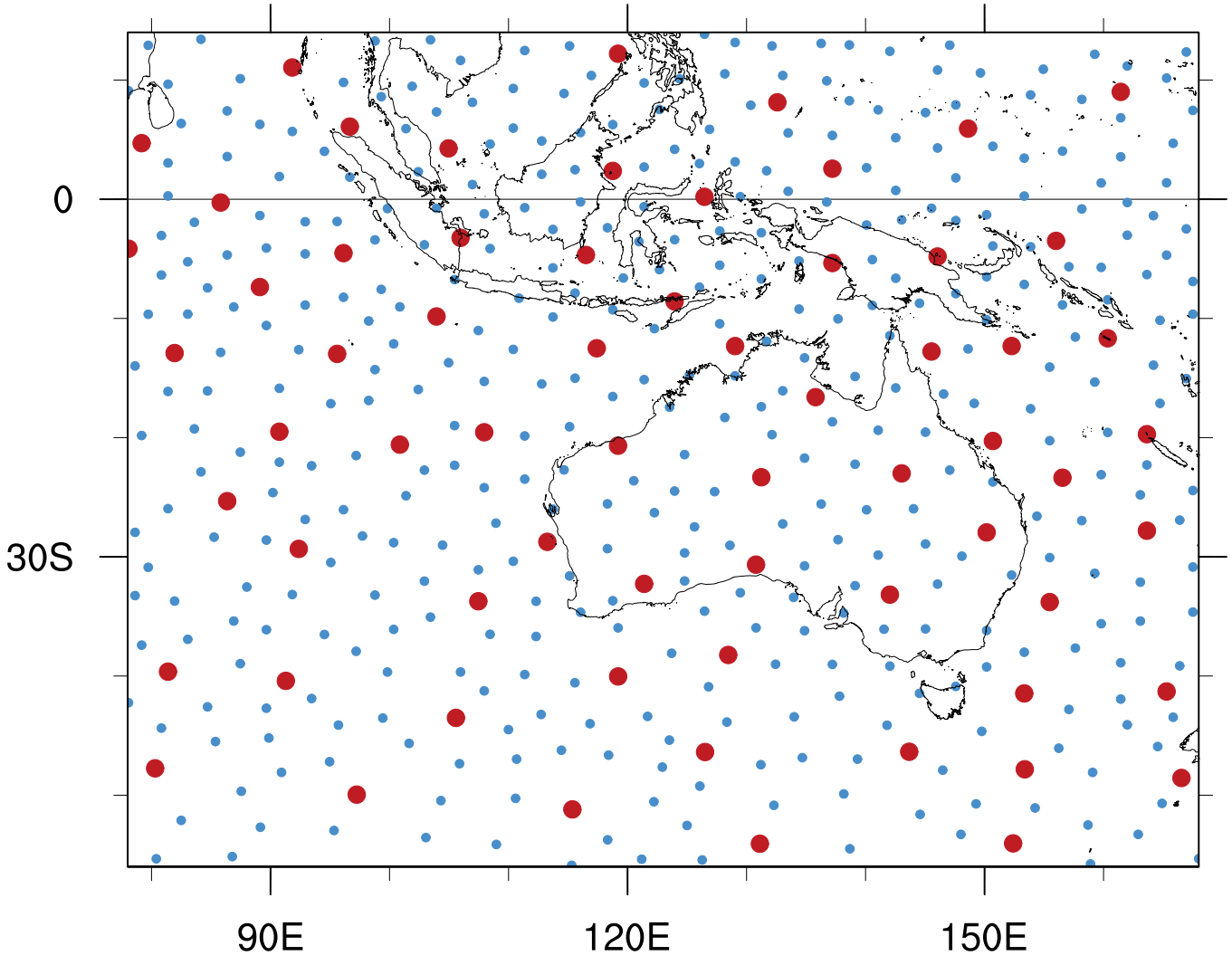
Vertical covariances are estimated for each subgrid point

Local estimation of K with VBAL



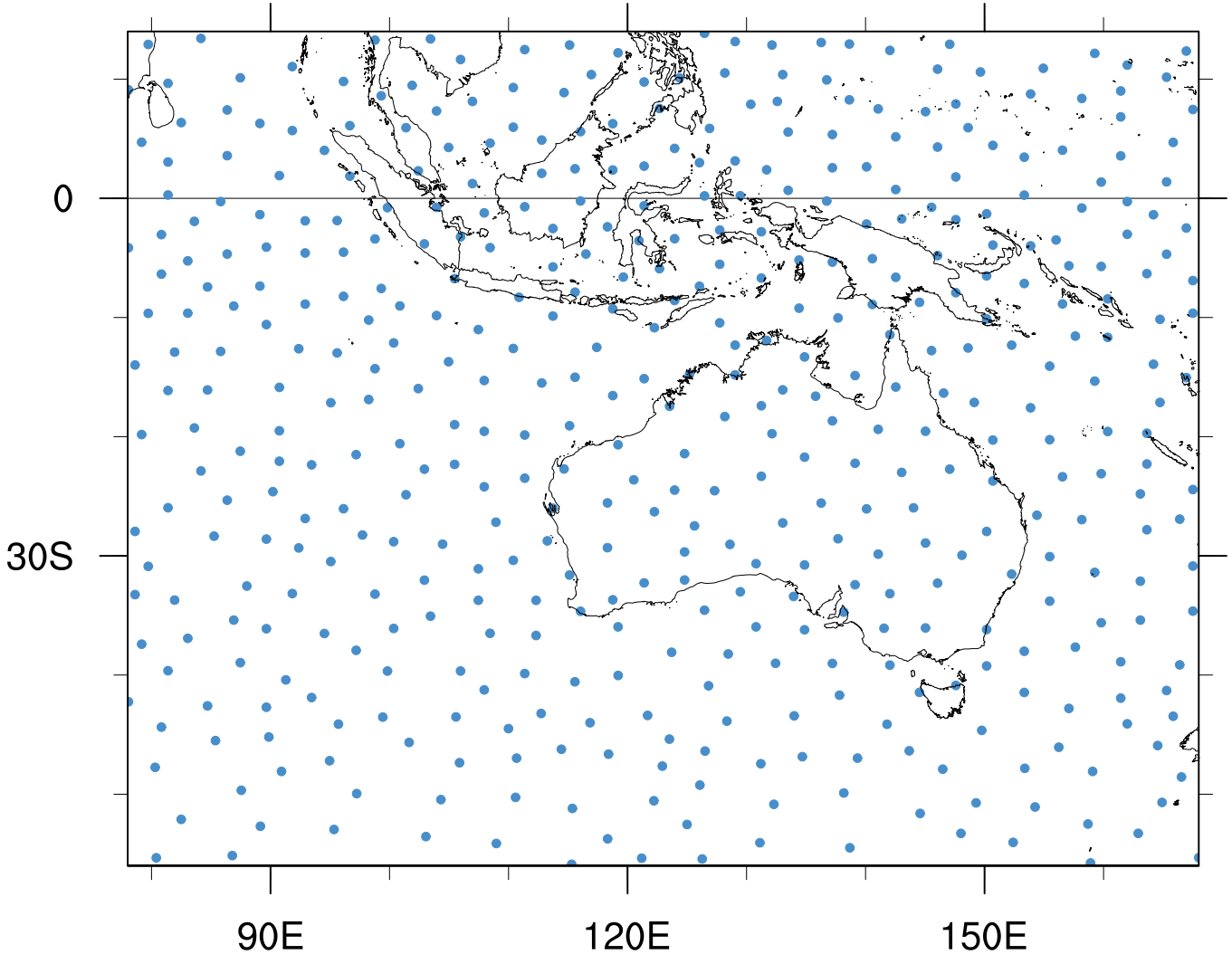
Vertical covariances are averaged over a given radius (dark blue points) at a local averaging center (red point)

Local estimation of K with VBAL



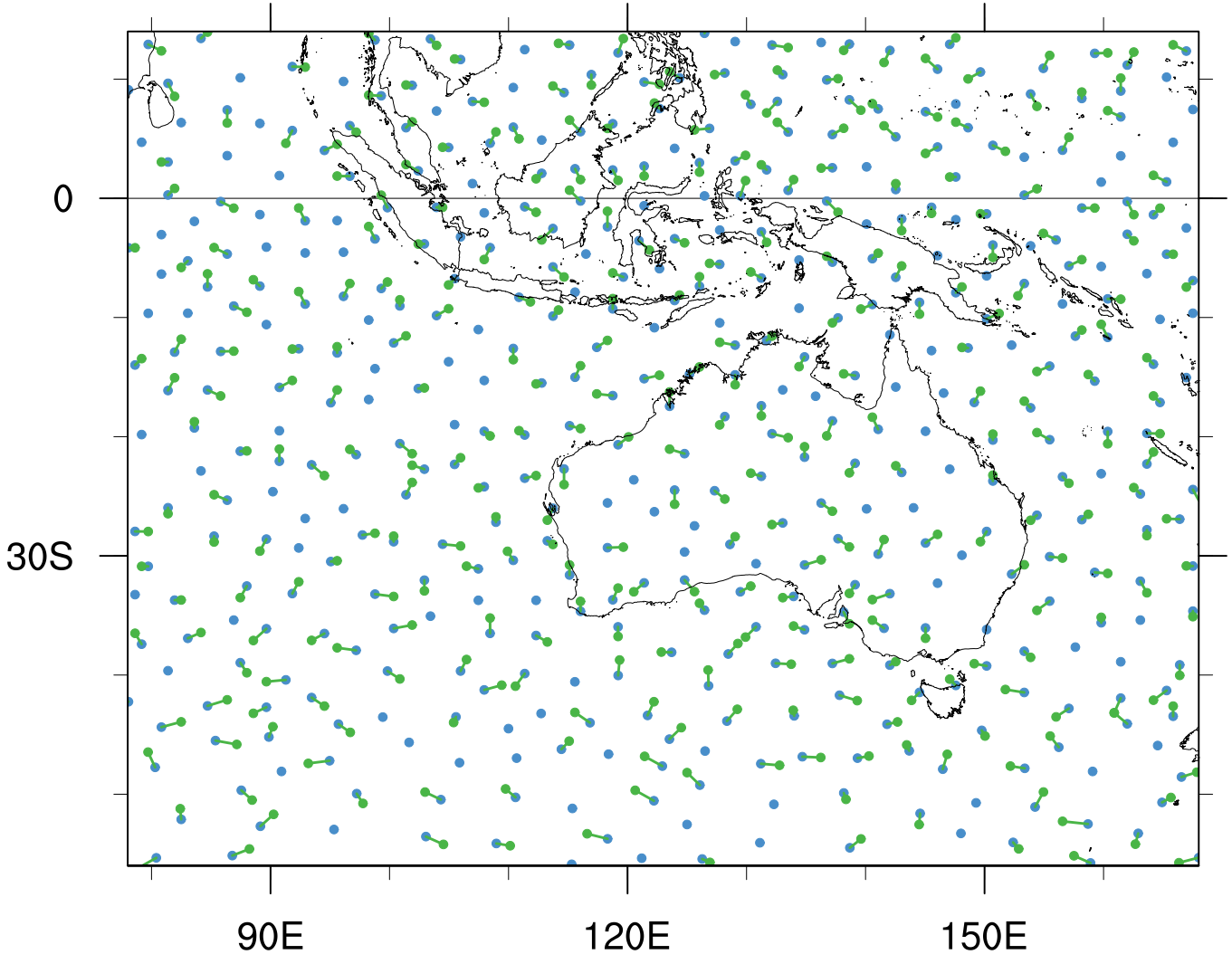
Same process repeated for many local averaging center (red points).
The balance operator is then interpolated to the model grid.

Local estimation of C with HDIAG



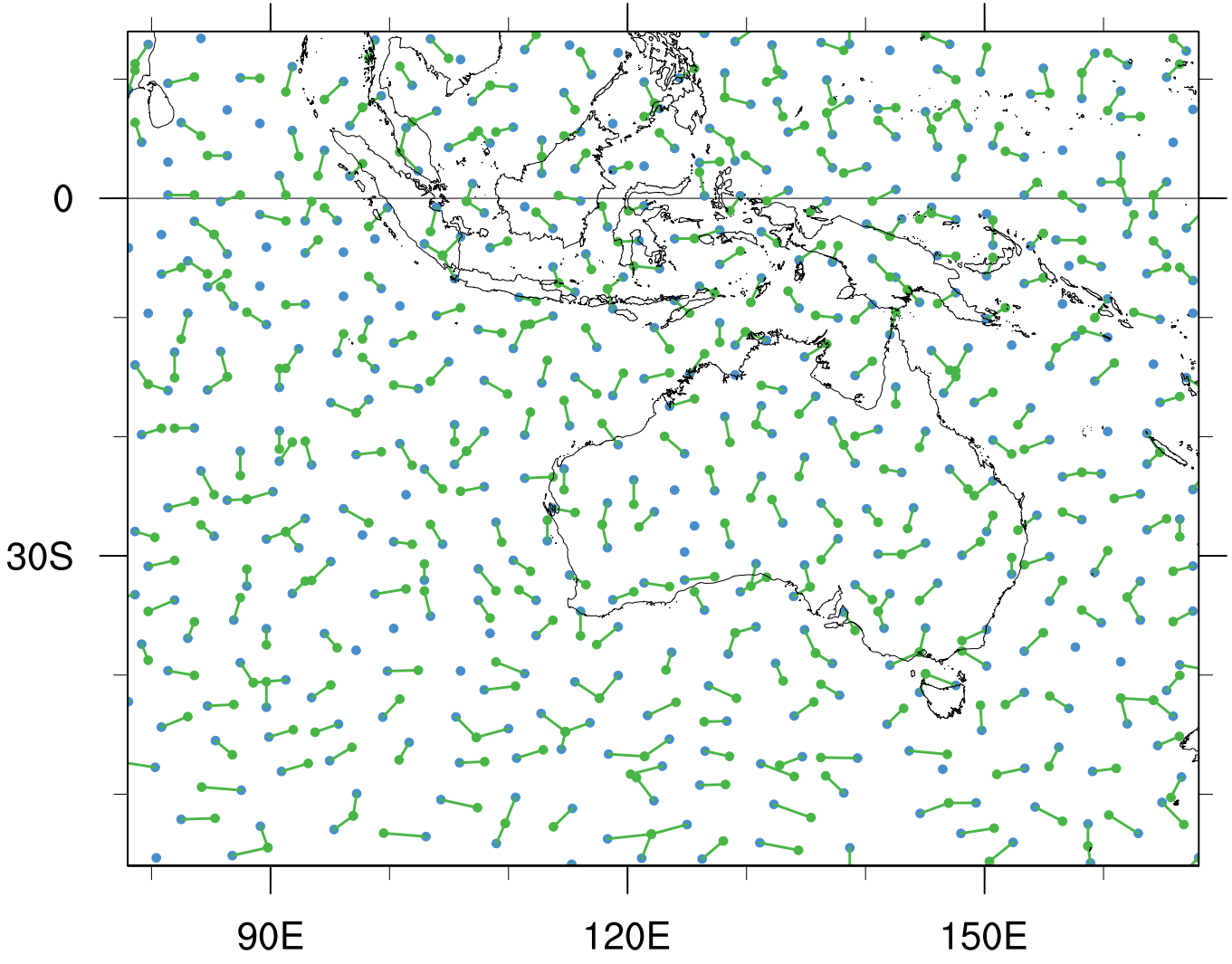
A first set of origin points is defined

Local estimation of C with HDIAG



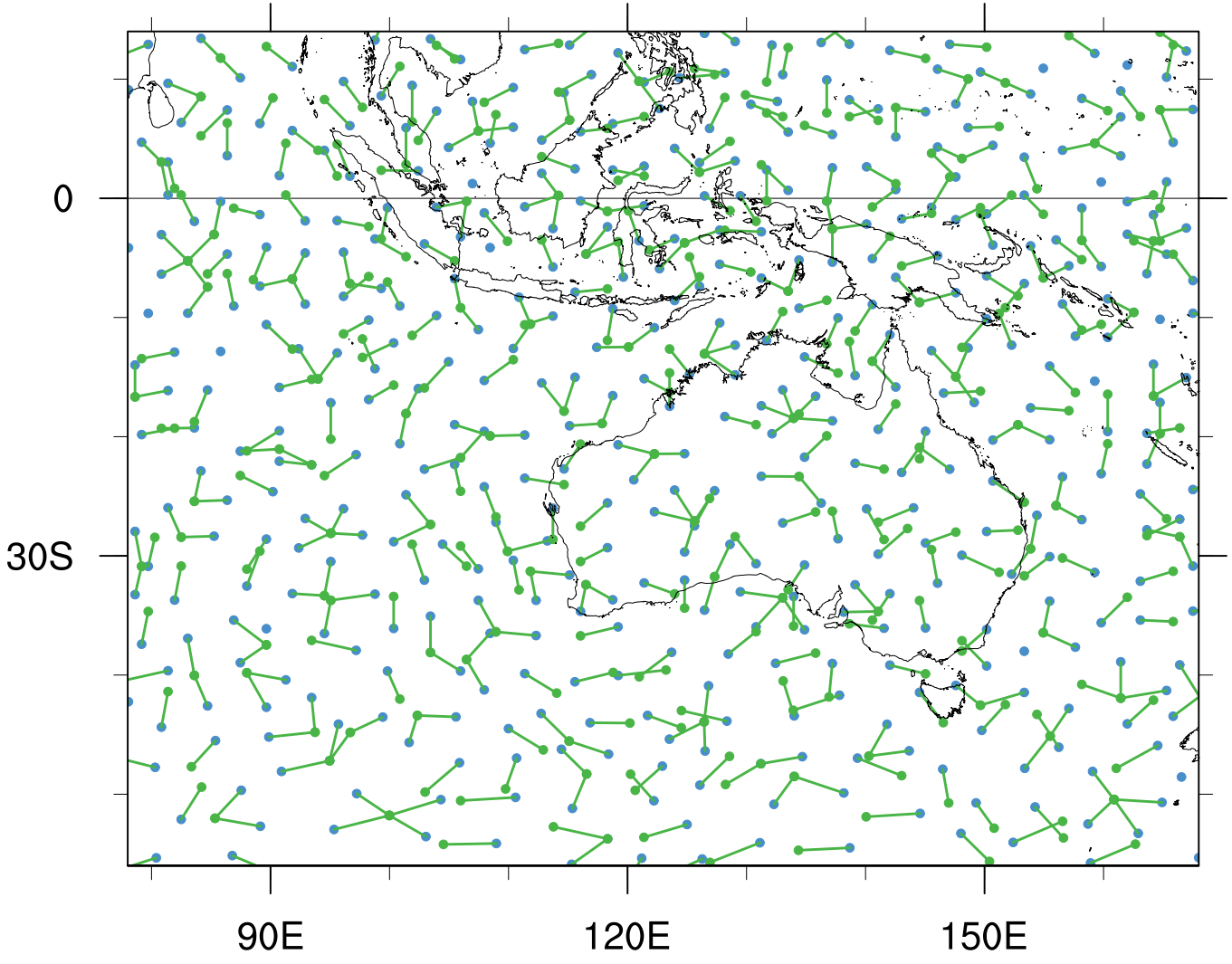
Isotropic sampling for distant points (increasing distance class)

Local estimation of C with HDIAG



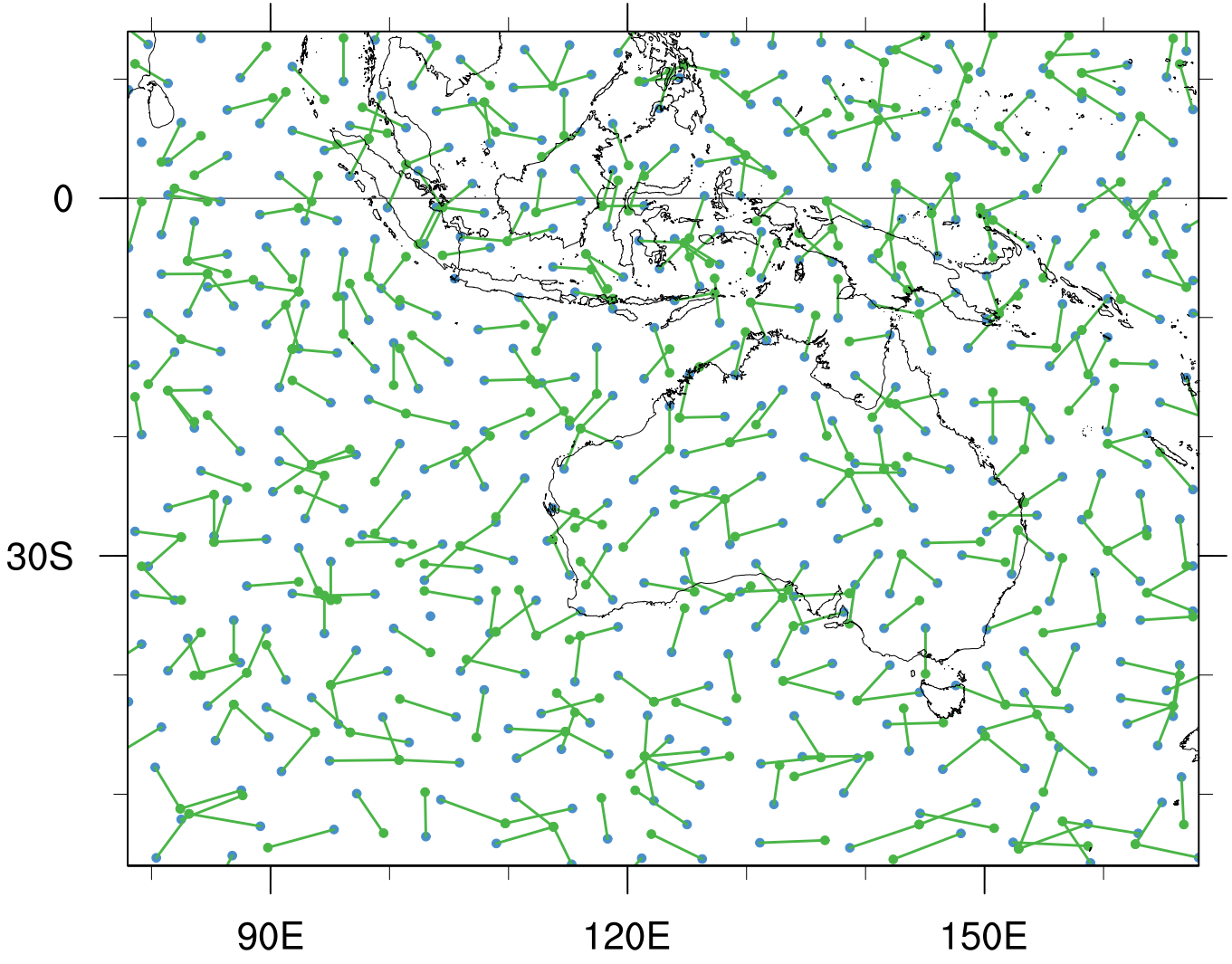
Isotropic sampling for distant points (increasing distance class)

Local estimation of C with HDIAG



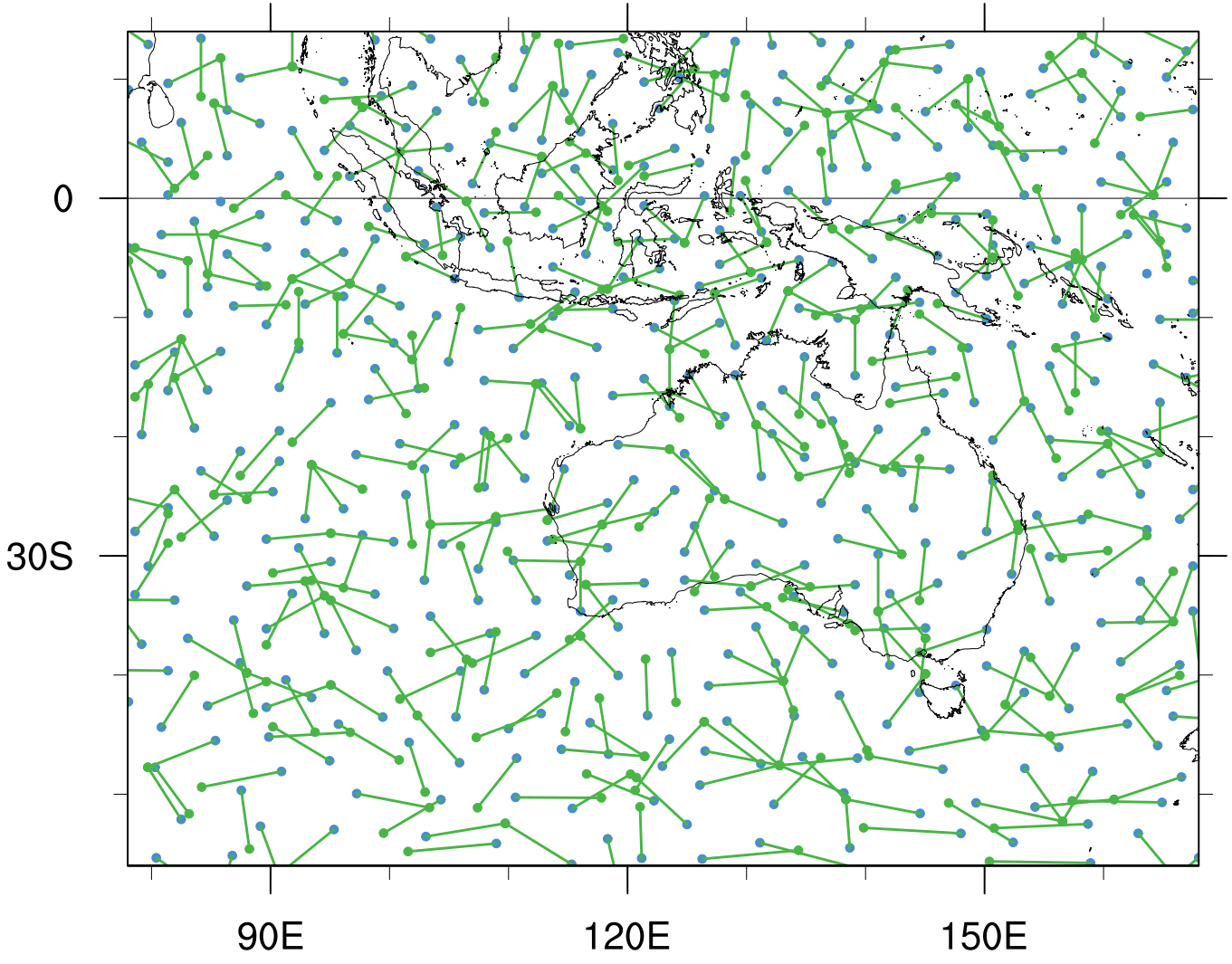
Isotropic sampling for distant points (increasing distance class)

Local estimation of C with HDIAG



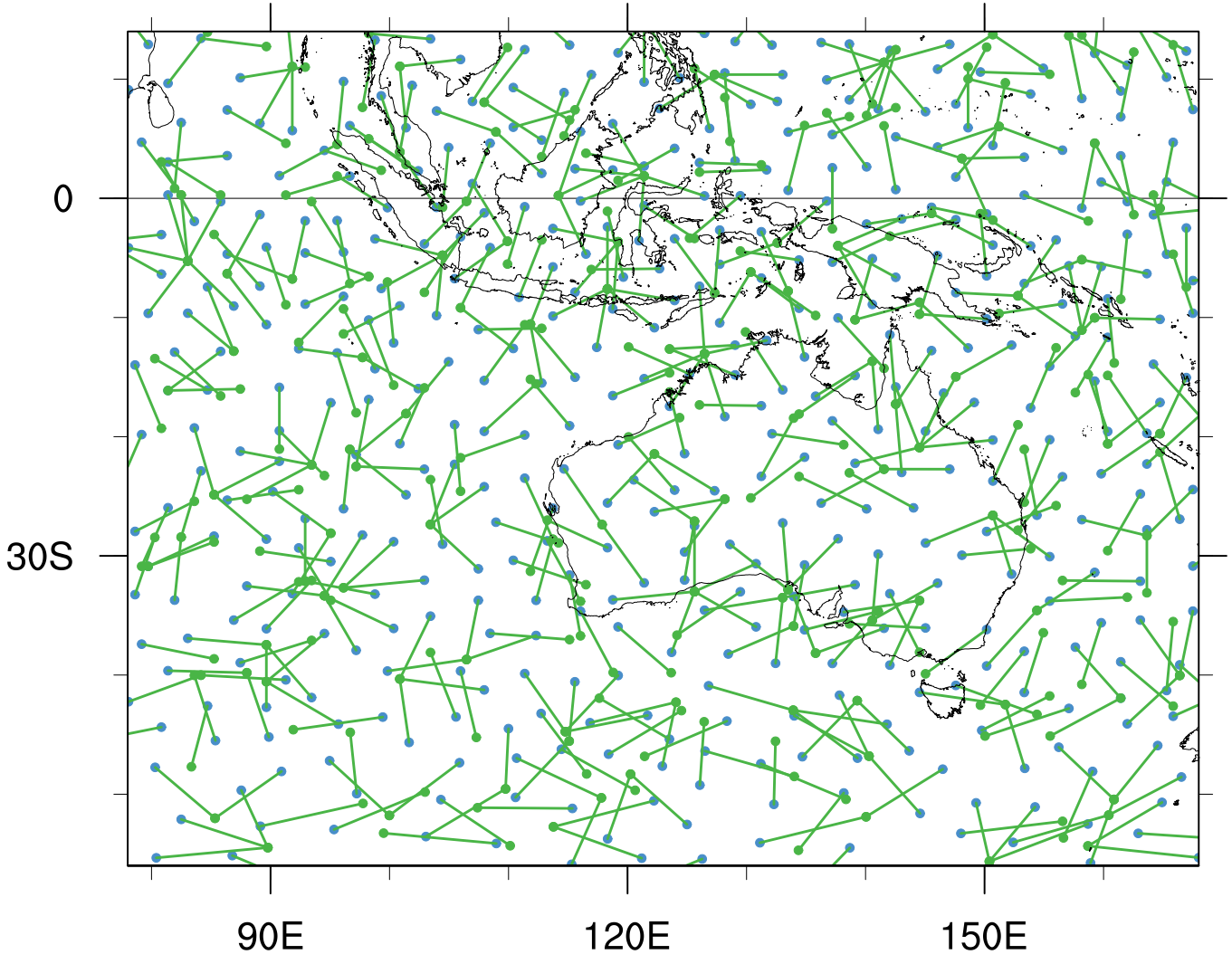
Isotropic sampling for distant points (increasing distance class)

Local estimation of C with HDIAG



Isotropic sampling for distant points (increasing distance class)

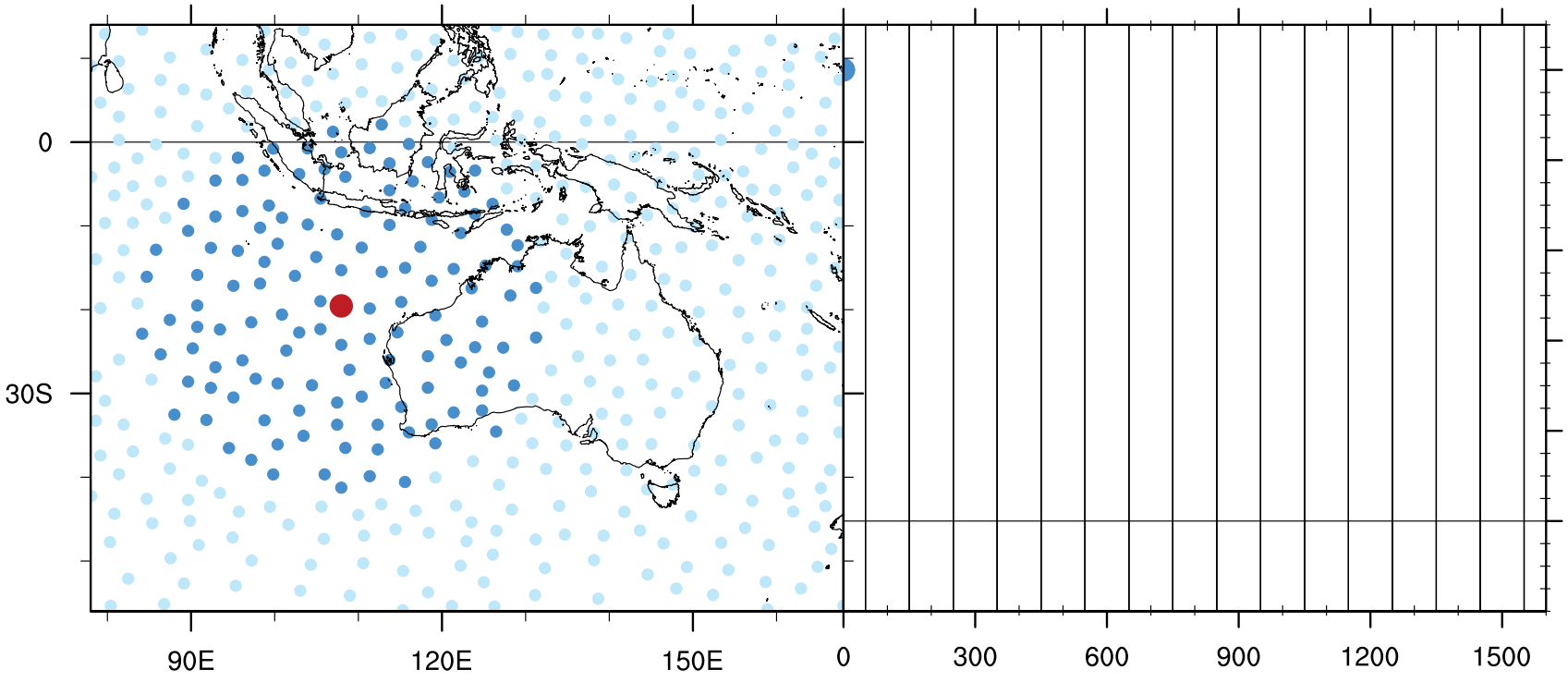
Local estimation of C with HDIAG



Isotropic sampling for distant points (increasing distance class)

Local estimation of C with HDIAG

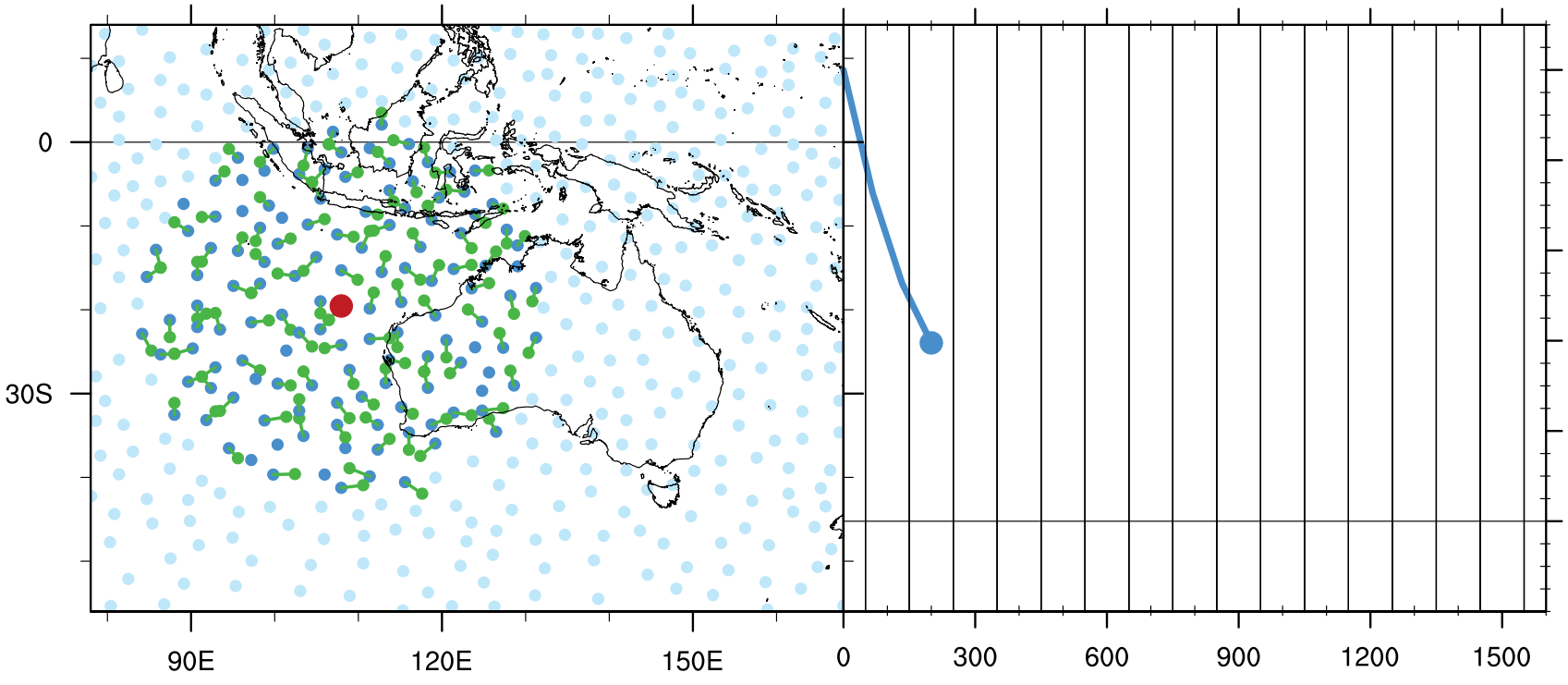
For a given local averaging center:



Averaged couples for a given local averaging center (left)
Diagnosed correlation as a function of distance in km (right)

Local estimation of C with HDIAG

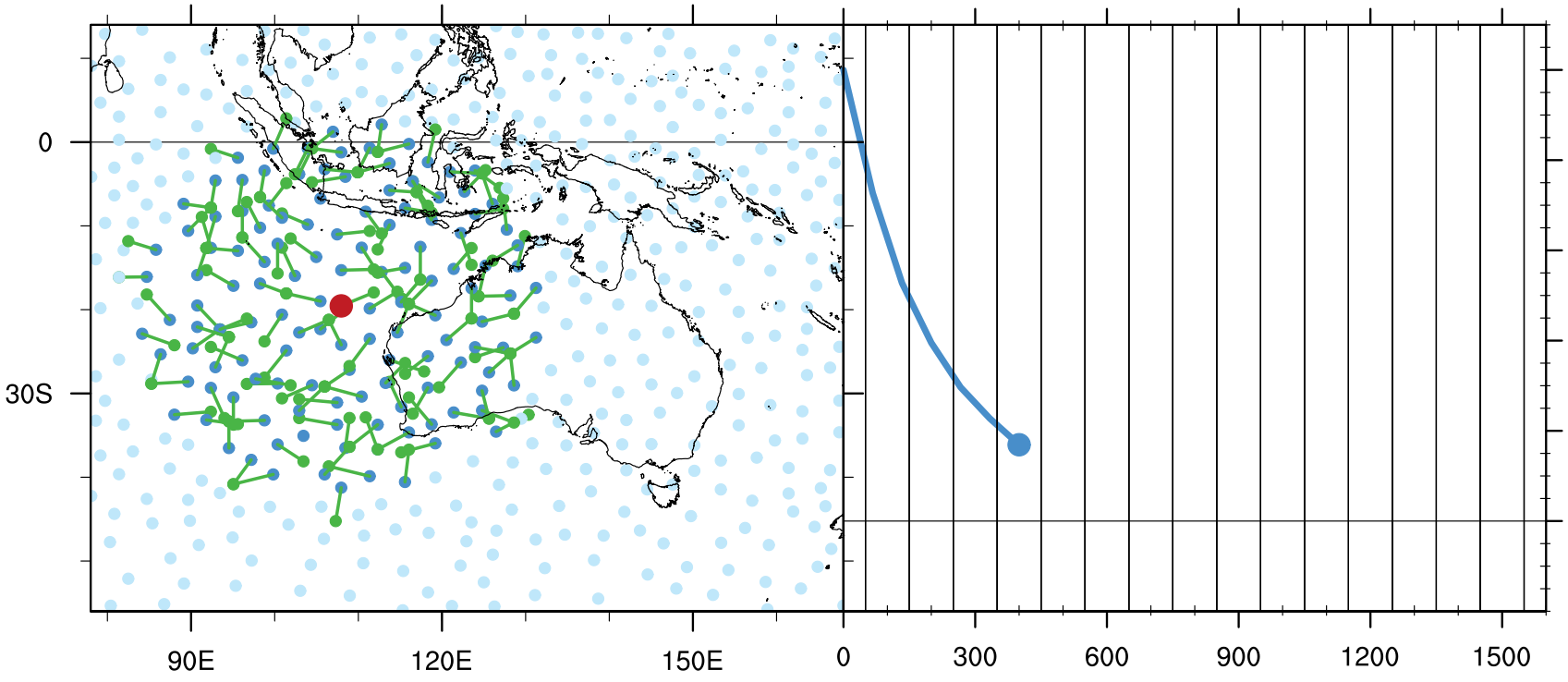
For a given local averaging center:



Averaged couples for a given local averaging center (left)
Diagnosed correlation as a function of distance in km (right)

Local estimation of C with HDIAG

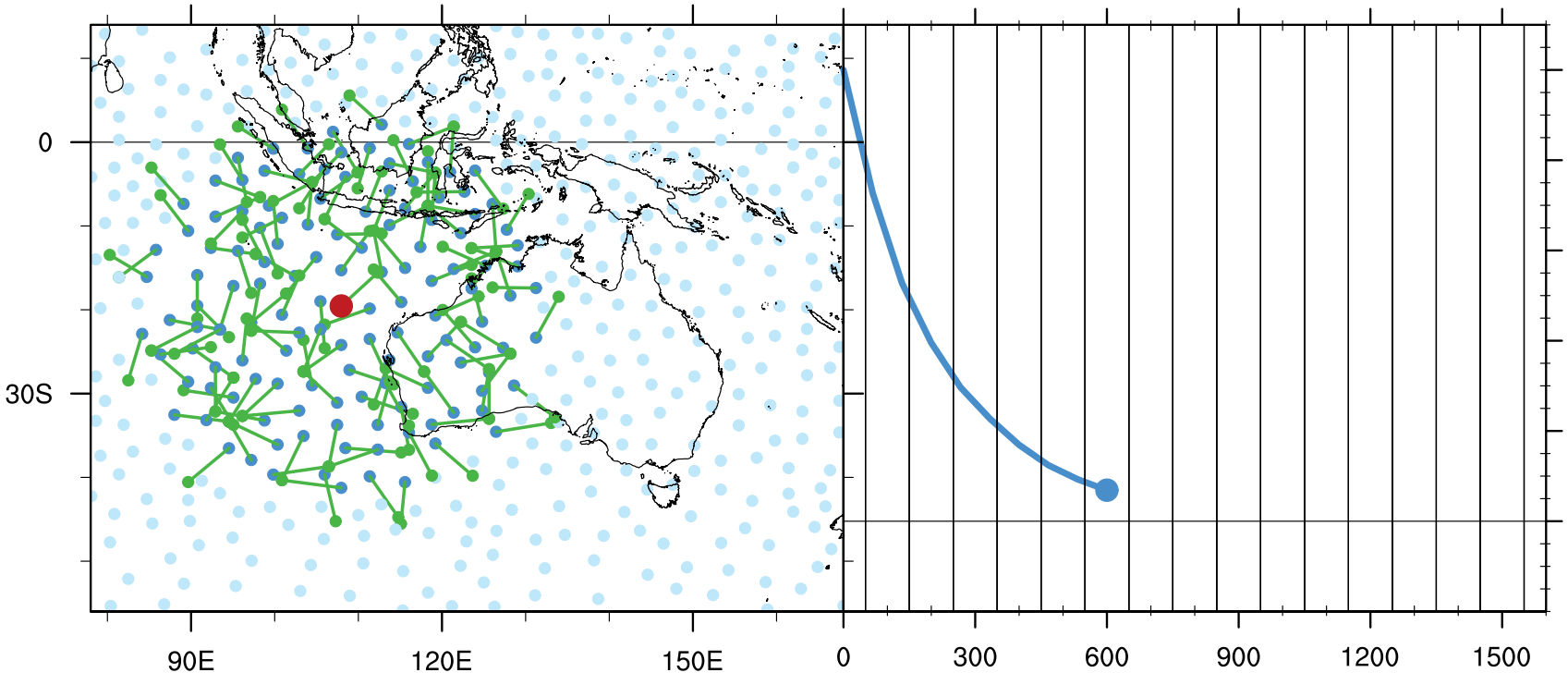
For a given local averaging center:



Averaged couples for a given local averaging center (left)
Diagnosed correlation as a function of distance in km (right)

Local estimation of C with HDIAG

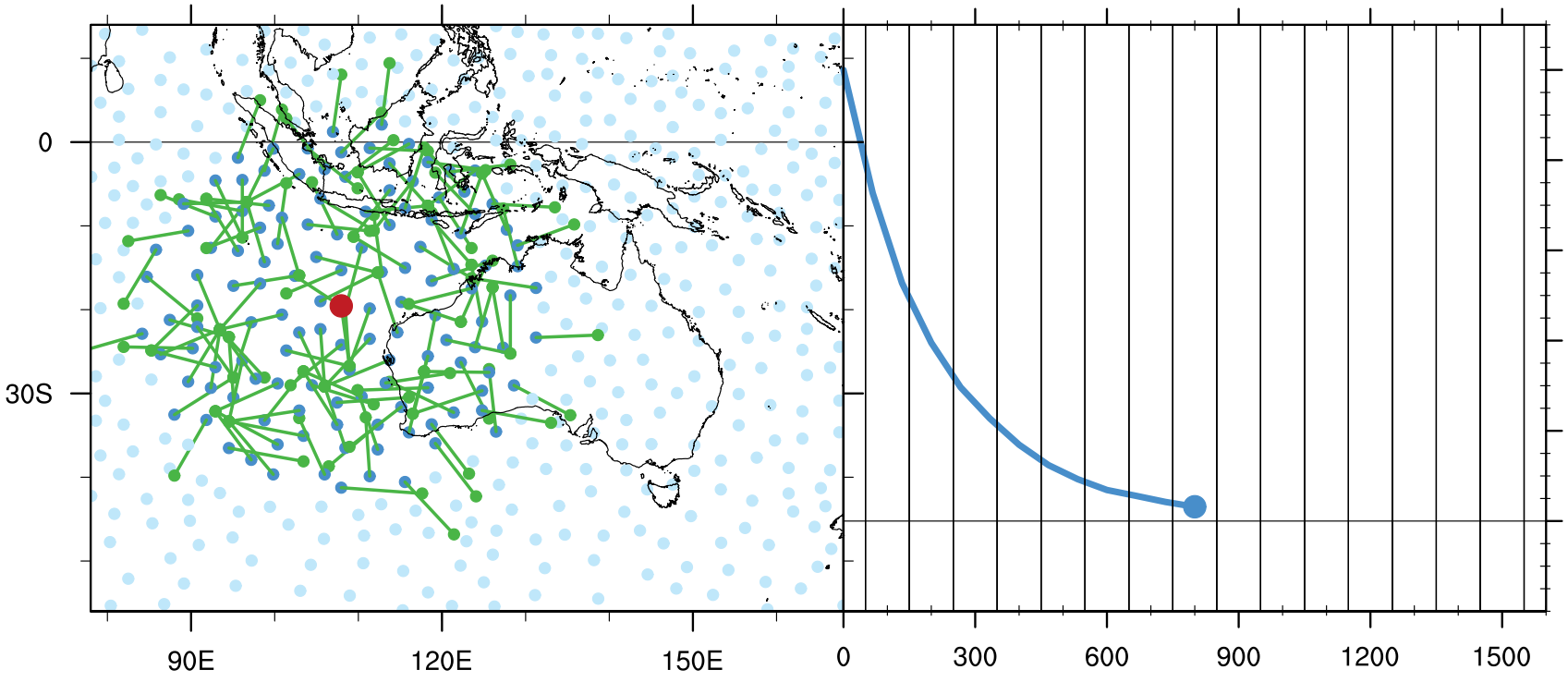
For a given local averaging center:



Averaged couples for a given local averaging center (left)
Diagnosed correlation as a function of distance in km (right)

Local estimation of C with HDIAG

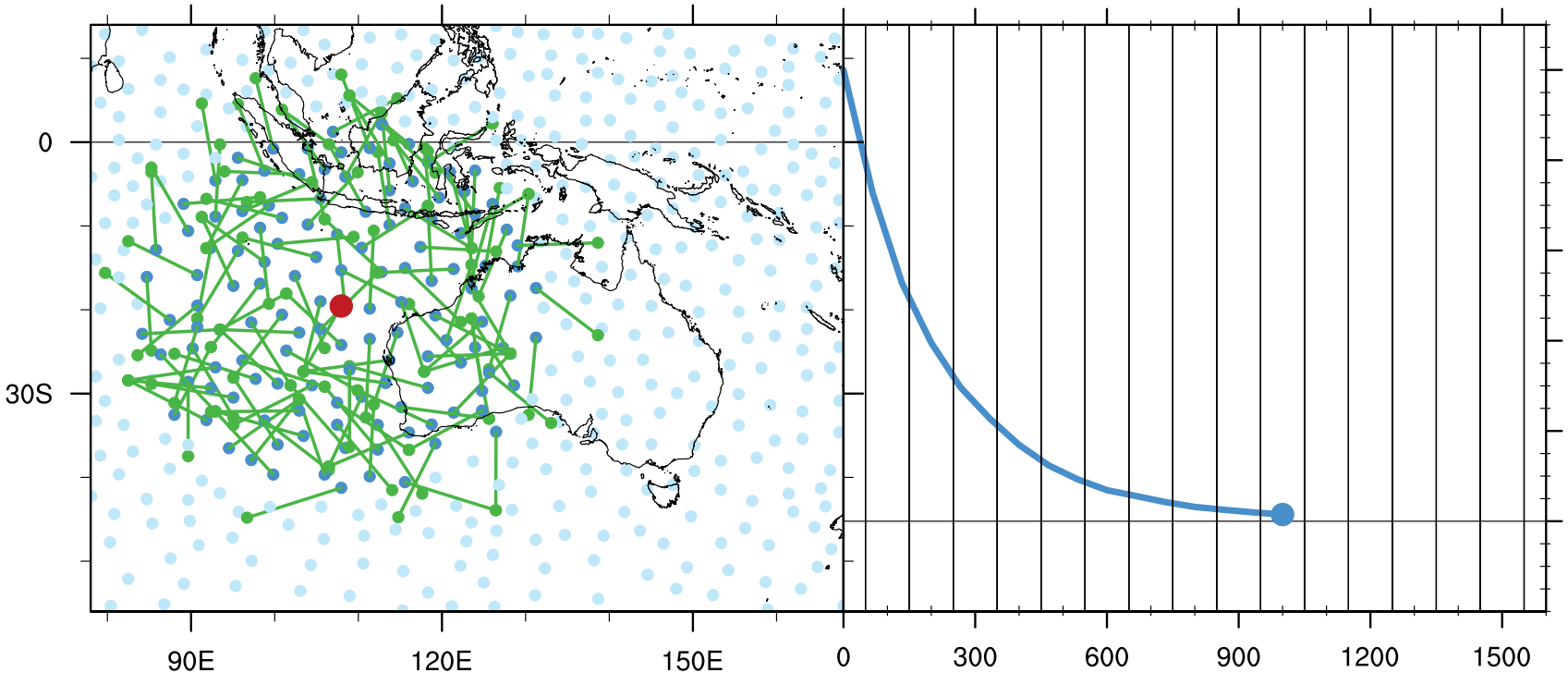
For a given local averaging center:



Averaged couples for a given local averaging center (left)
Diagnosed correlation as a function of distance in km (right)

Local estimation of C with HDIAG

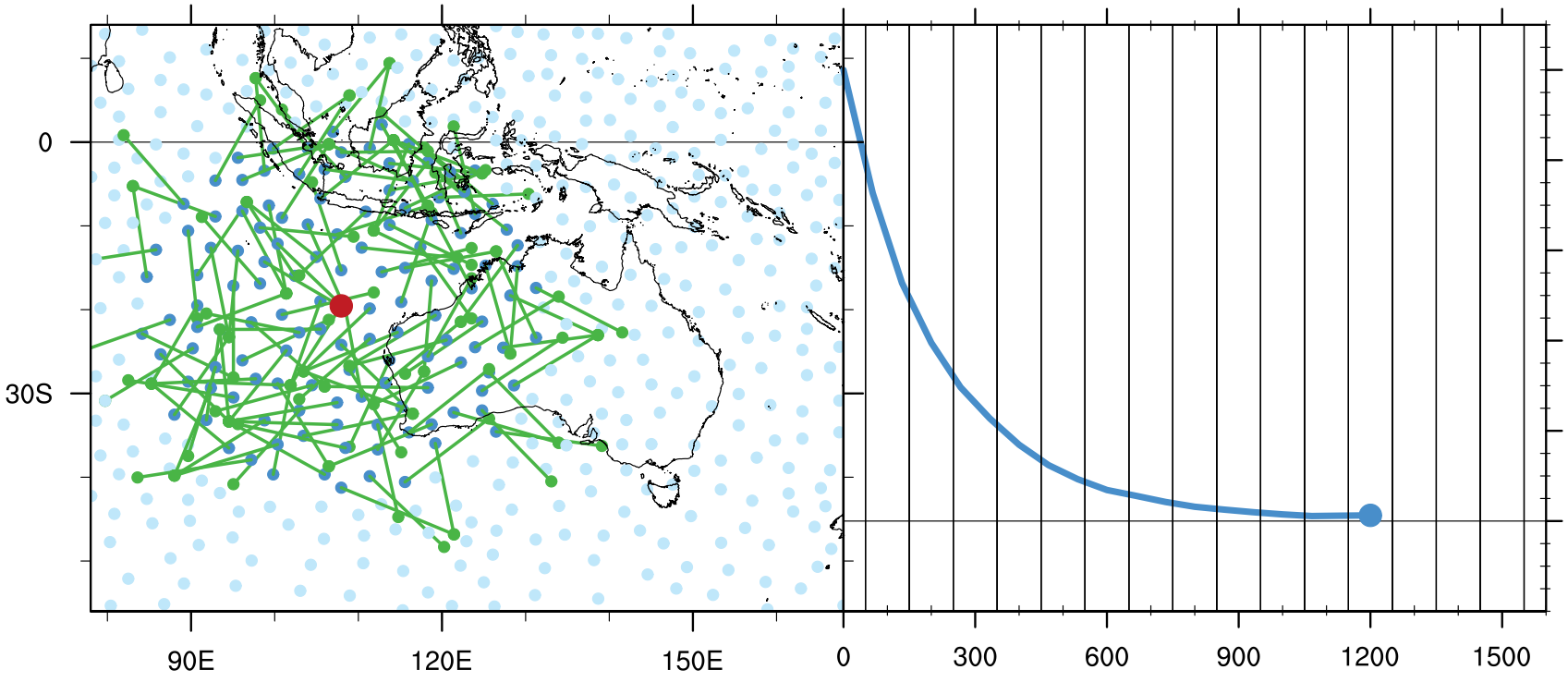
For a given local averaging center:



Averaged couples for a given local averaging center (left)
Diagnosed correlation as a function of distance in km (right)

Local estimation of C with HDIAG

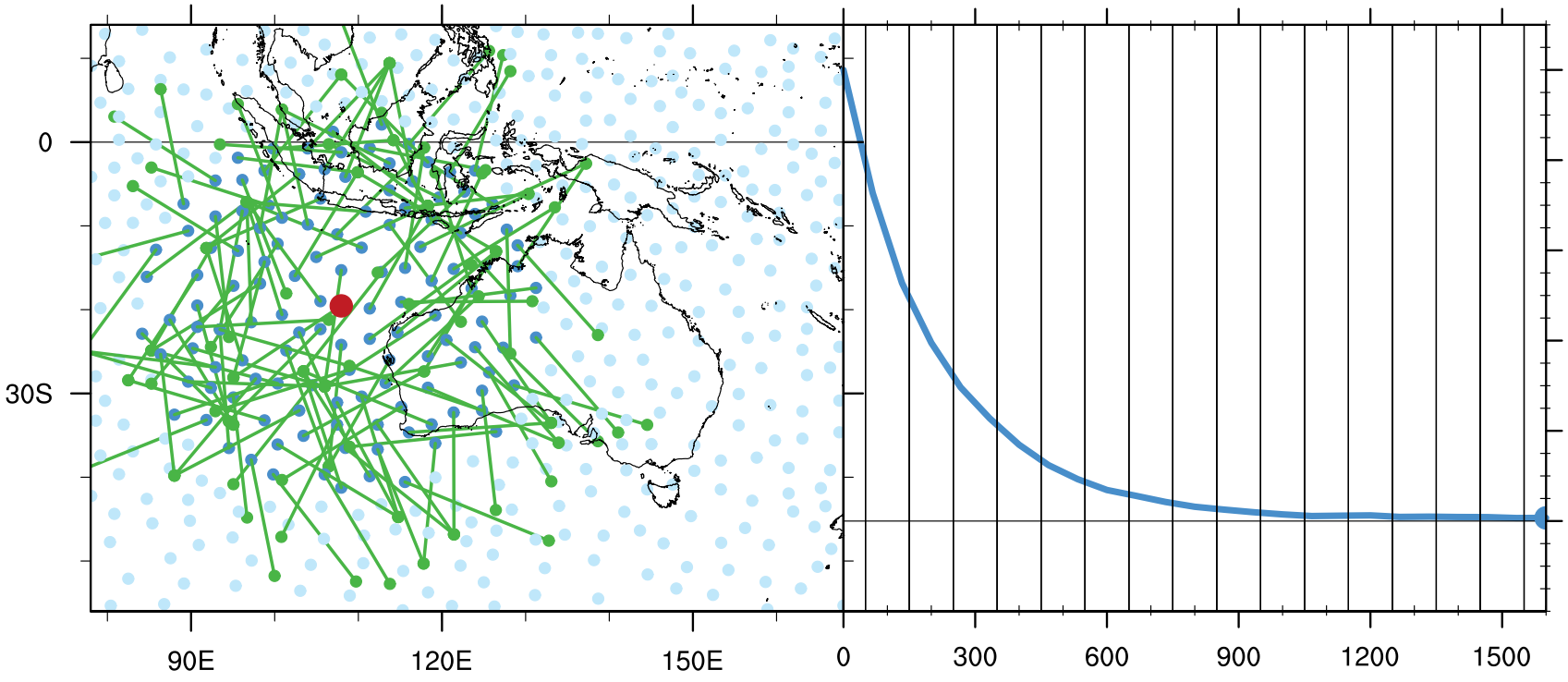
For a given local averaging center:



Averaged couples for a given local averaging center (left)
Diagnosed correlation as a function of distance in km (right)

Local estimation of C with HDIAG

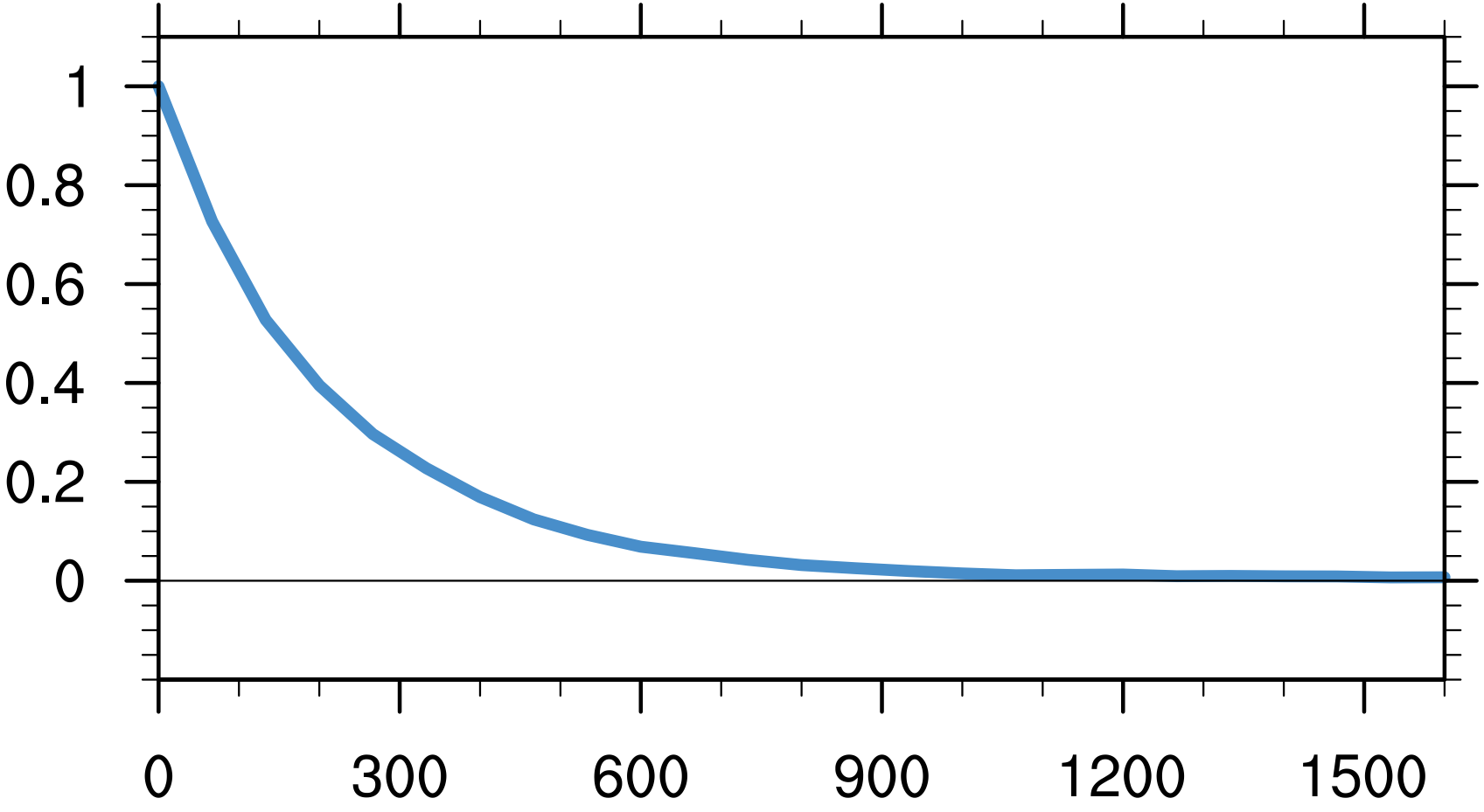
For a given local averaging center:



Averaged couples for a given local averaging center (left)
Diagnosed correlation as a function of distance in km (right)

Local estimation of C with HDIAG

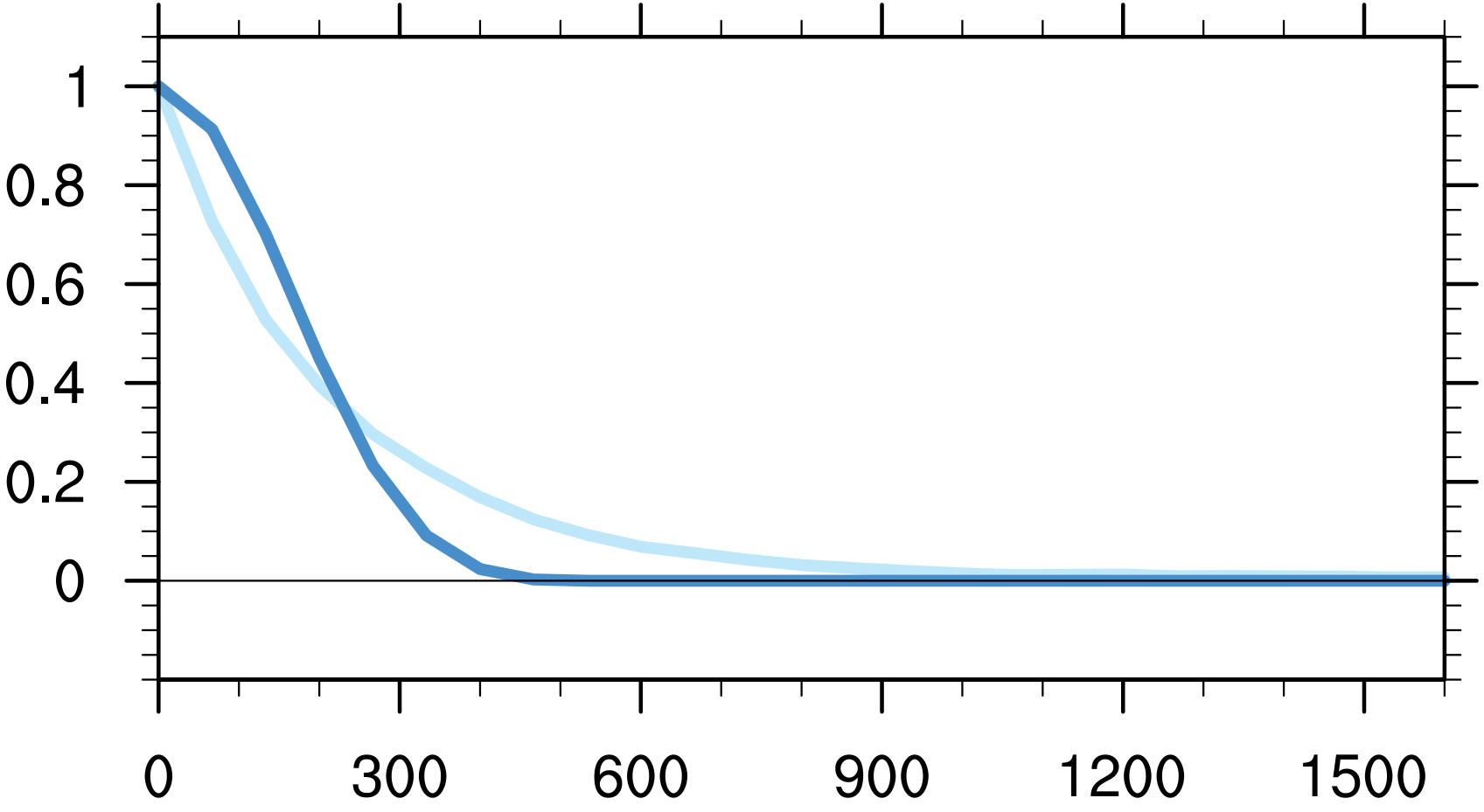
For a given local averaging center:



Raw correlation as a function of distance (km)

Local estimation of C with HDIAG

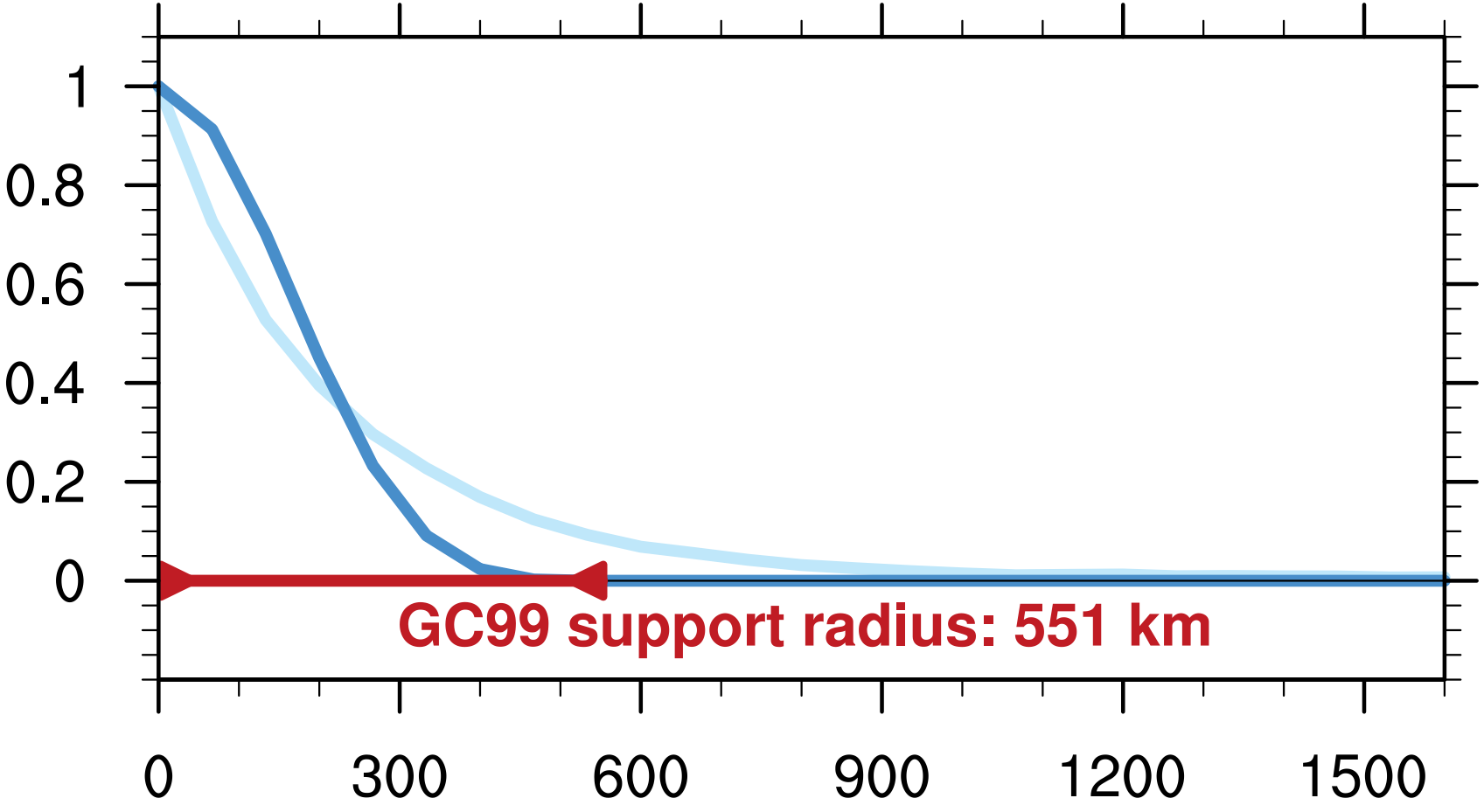
For a given local averaging center:



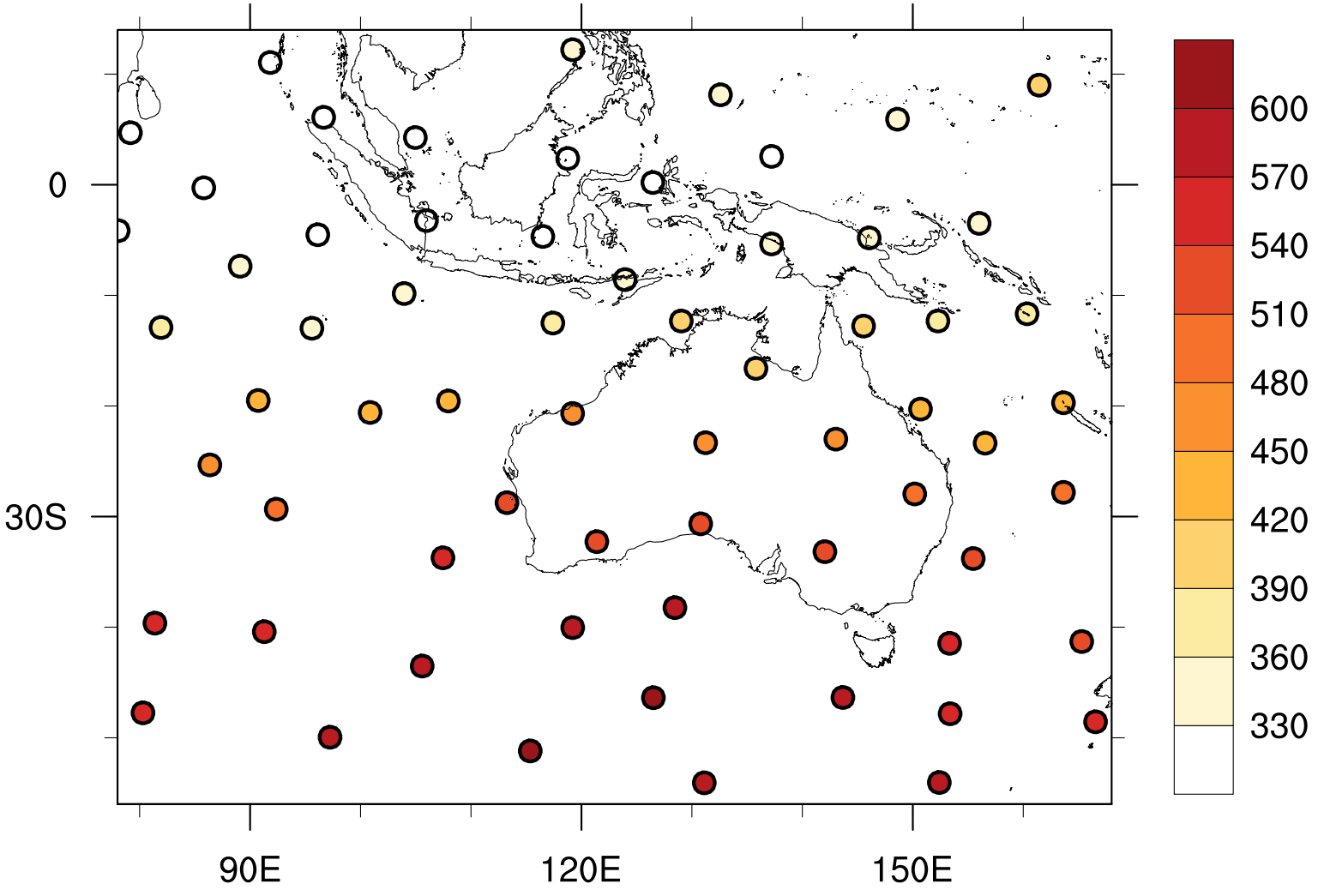
Fit with a Gaspari-Cohn (1999) function... not the best choice

Local estimation of C with HDIAG

For a given local averaging center:

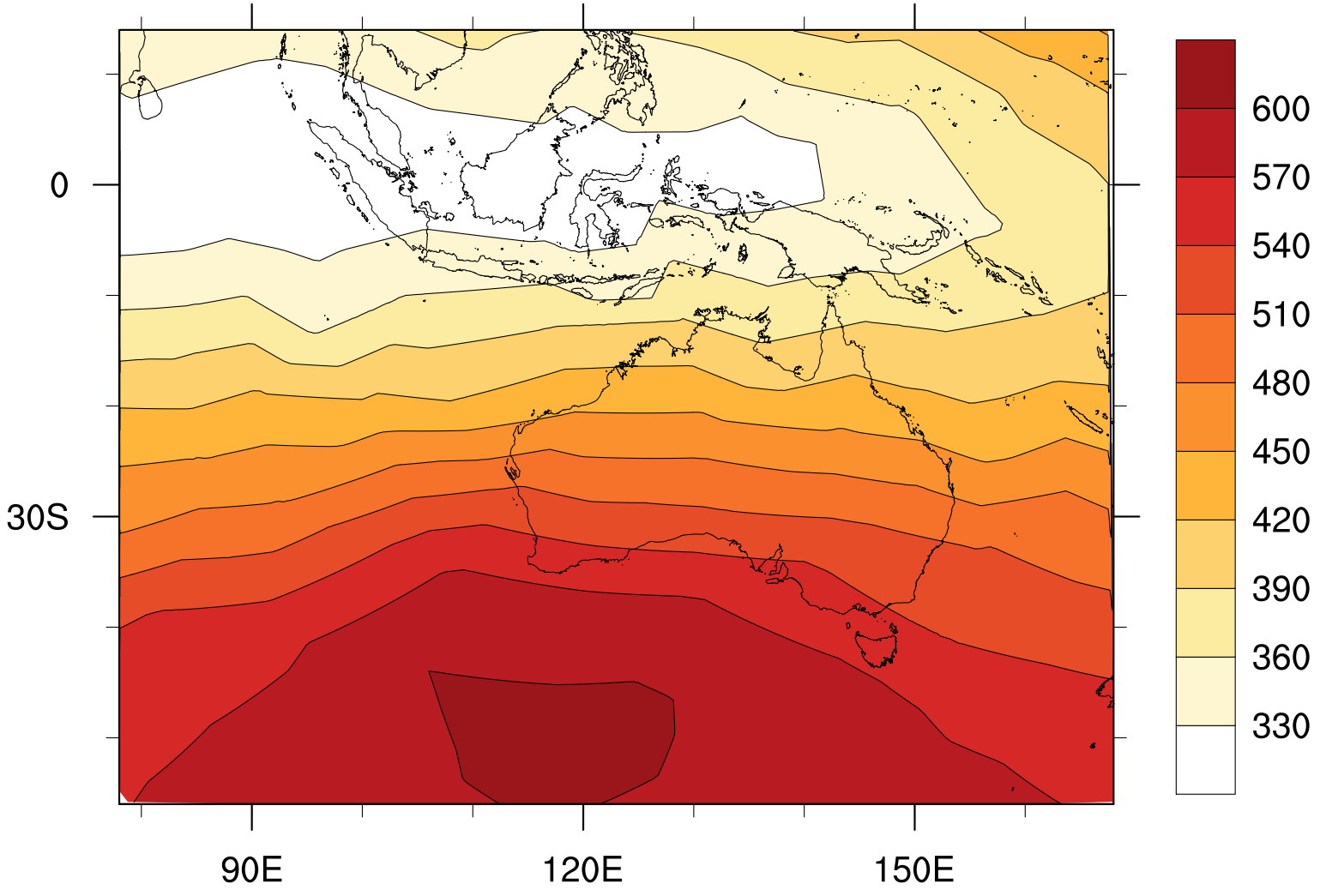


Local estimation of C with HDIAG



Correlation support radius (km) at local averaging centers

Local estimation of C with HDIAG



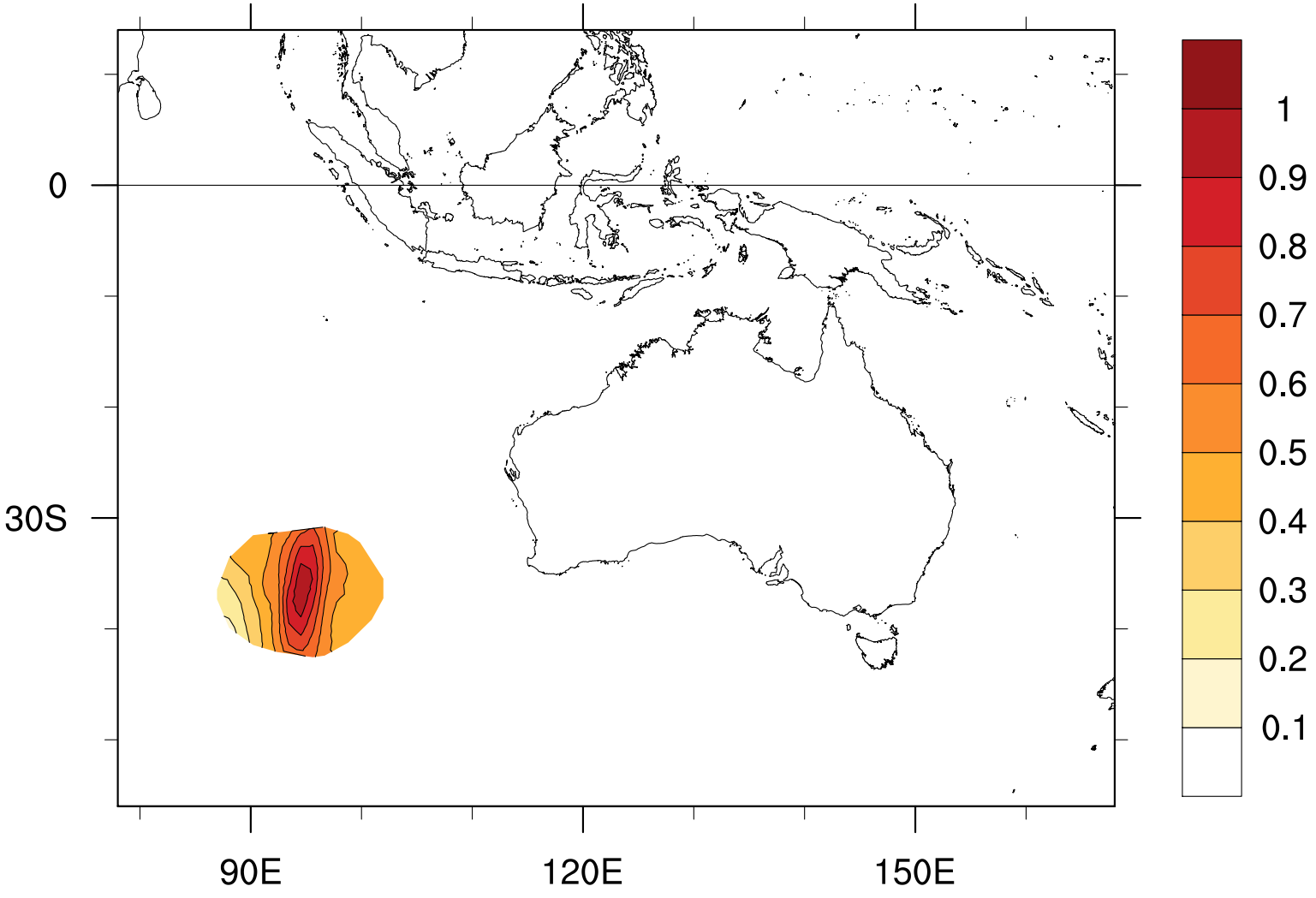
Correlation support radius (km) interpolated on the model grid

Local estimation of C with HDIAG

HDIAG summary (correlation):

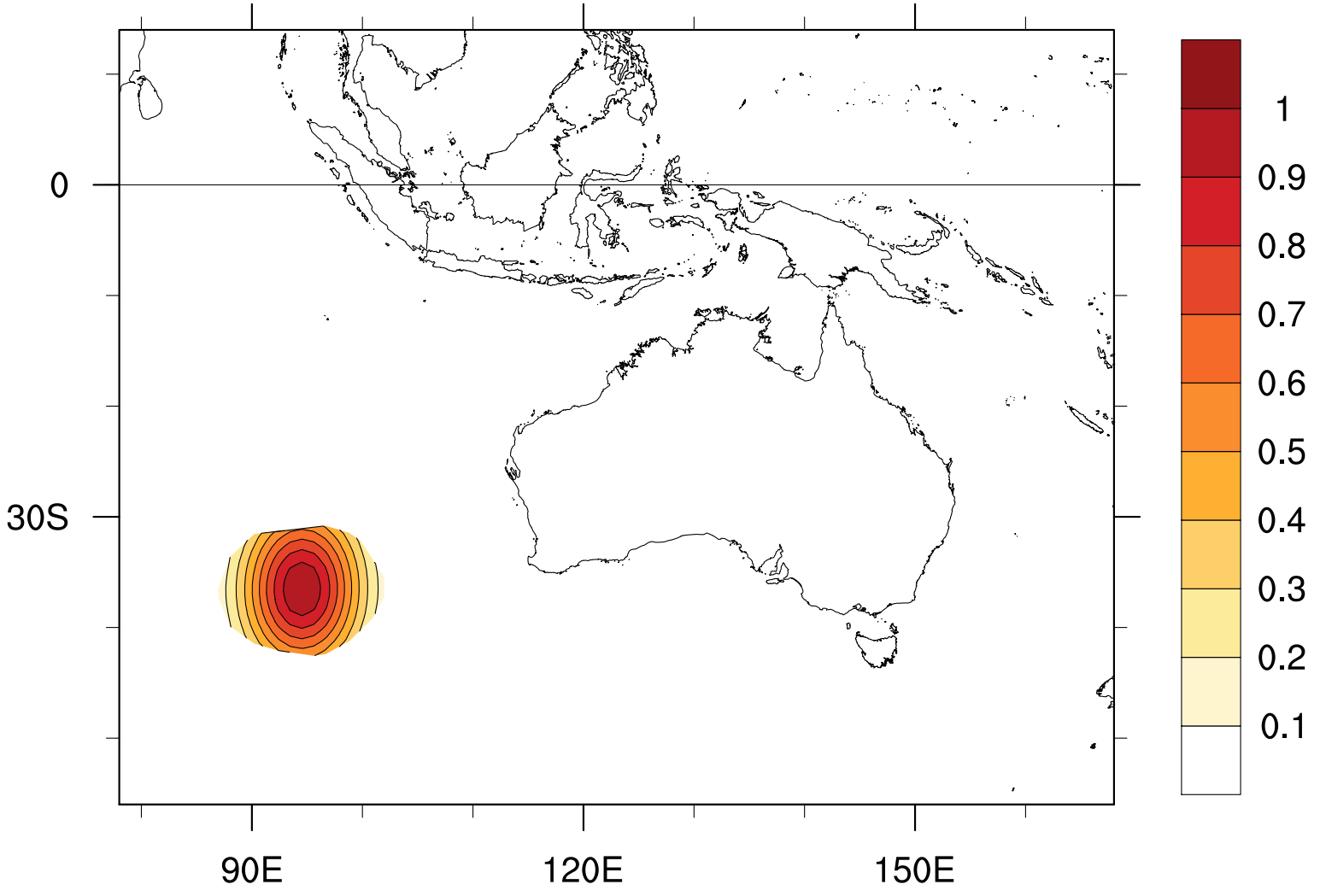
1. Define a homogeneous subgrid (origin points).
2. For each origin point, find distant points for a series of distance classes.
3. For each local averaging center:
   - Average diagnostic values over a given radius.
   - Compute the correlation.
   - Fit the raw correlation function with a parametrized function (e.g. Gaspari-Cohn, 1999) to get the correlation support radius.
4. Optionally filter the correlation support radius on the subgrid.
5. Interpolate the correlation support radius over the model grid.

Local estimation of C with LCT



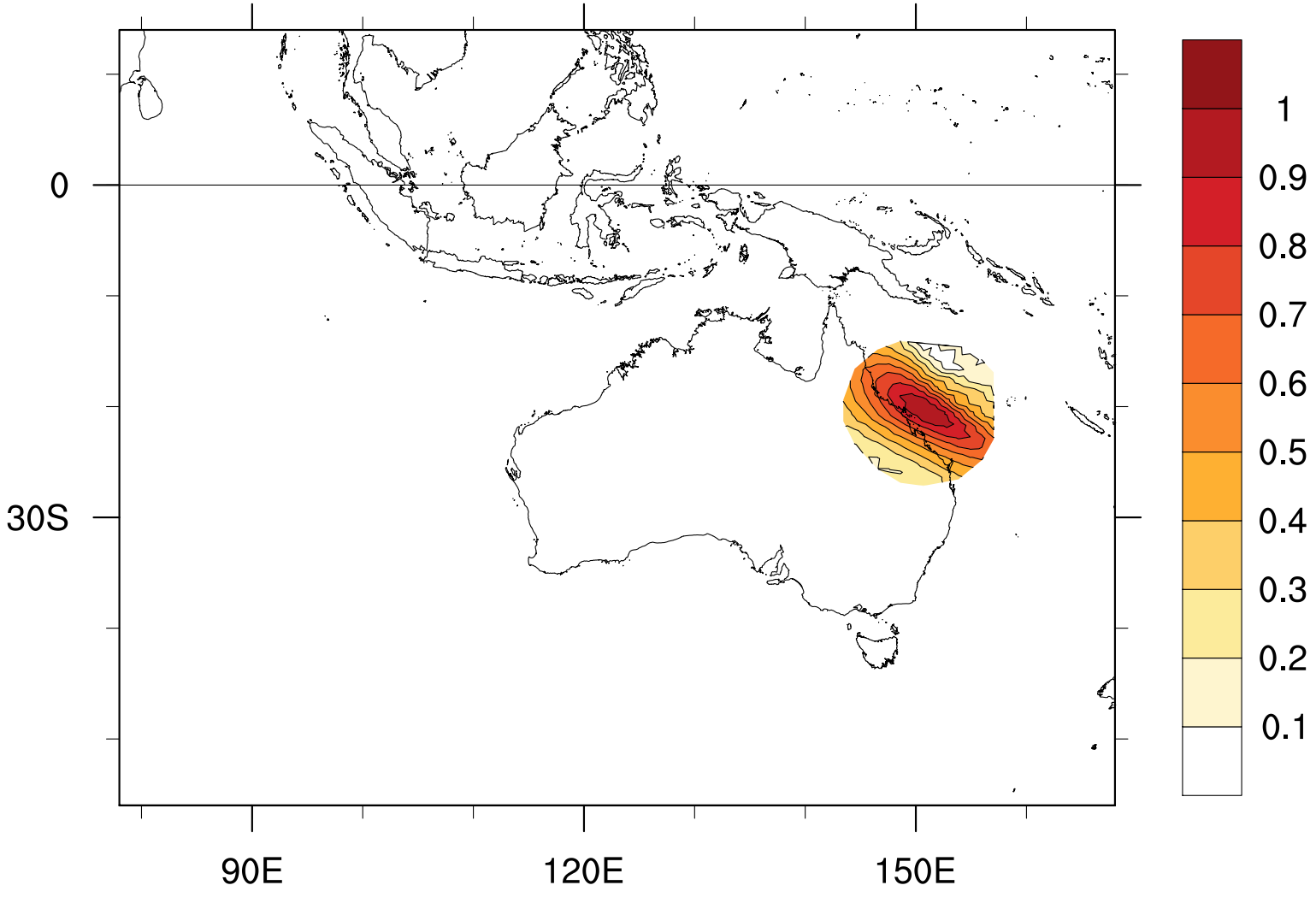
Estimation of a local correlation function on the model grid

Local estimation of C with LCT



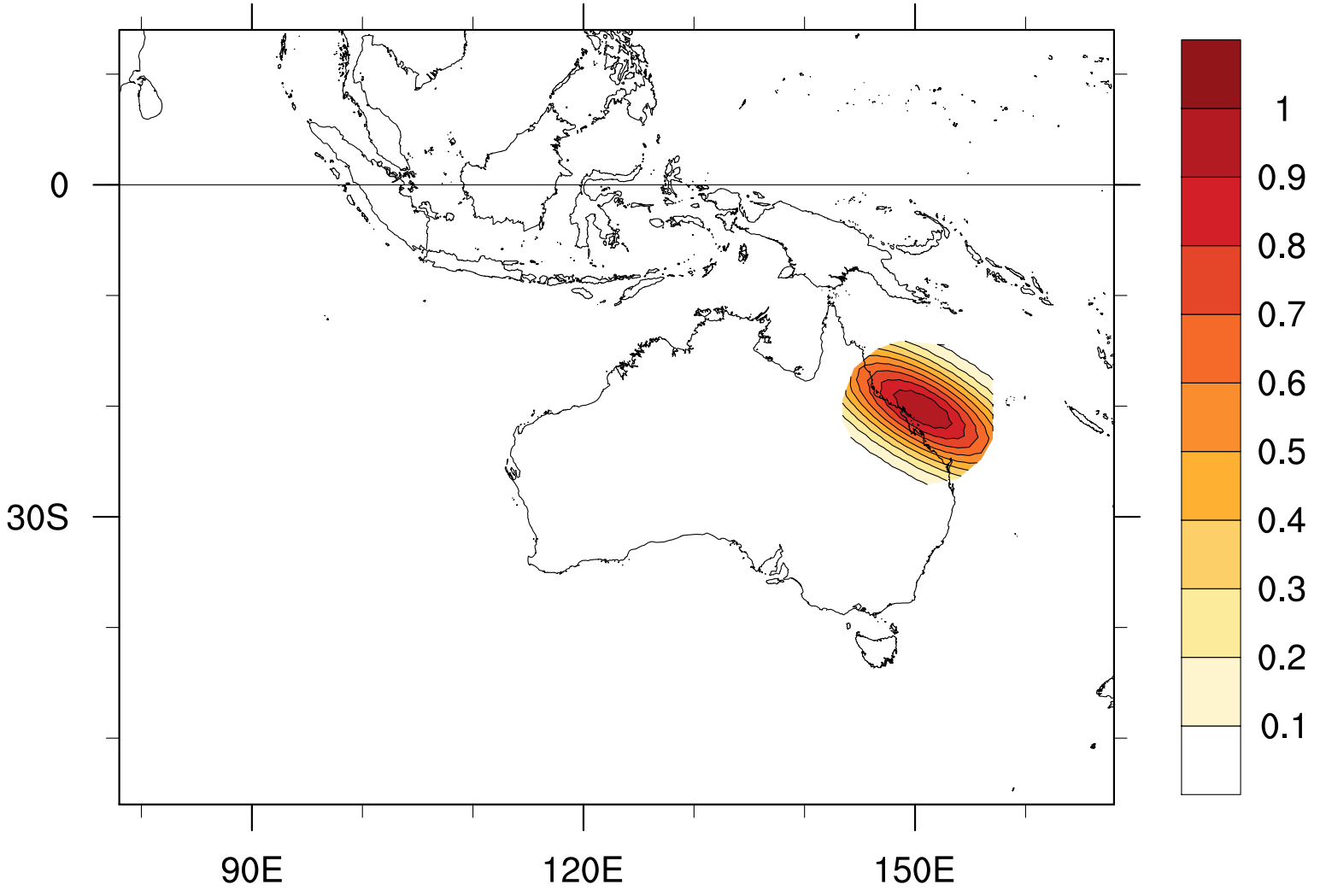
Fit with a Local Correlation Tensor (LCT)

Local estimation of C with LCT



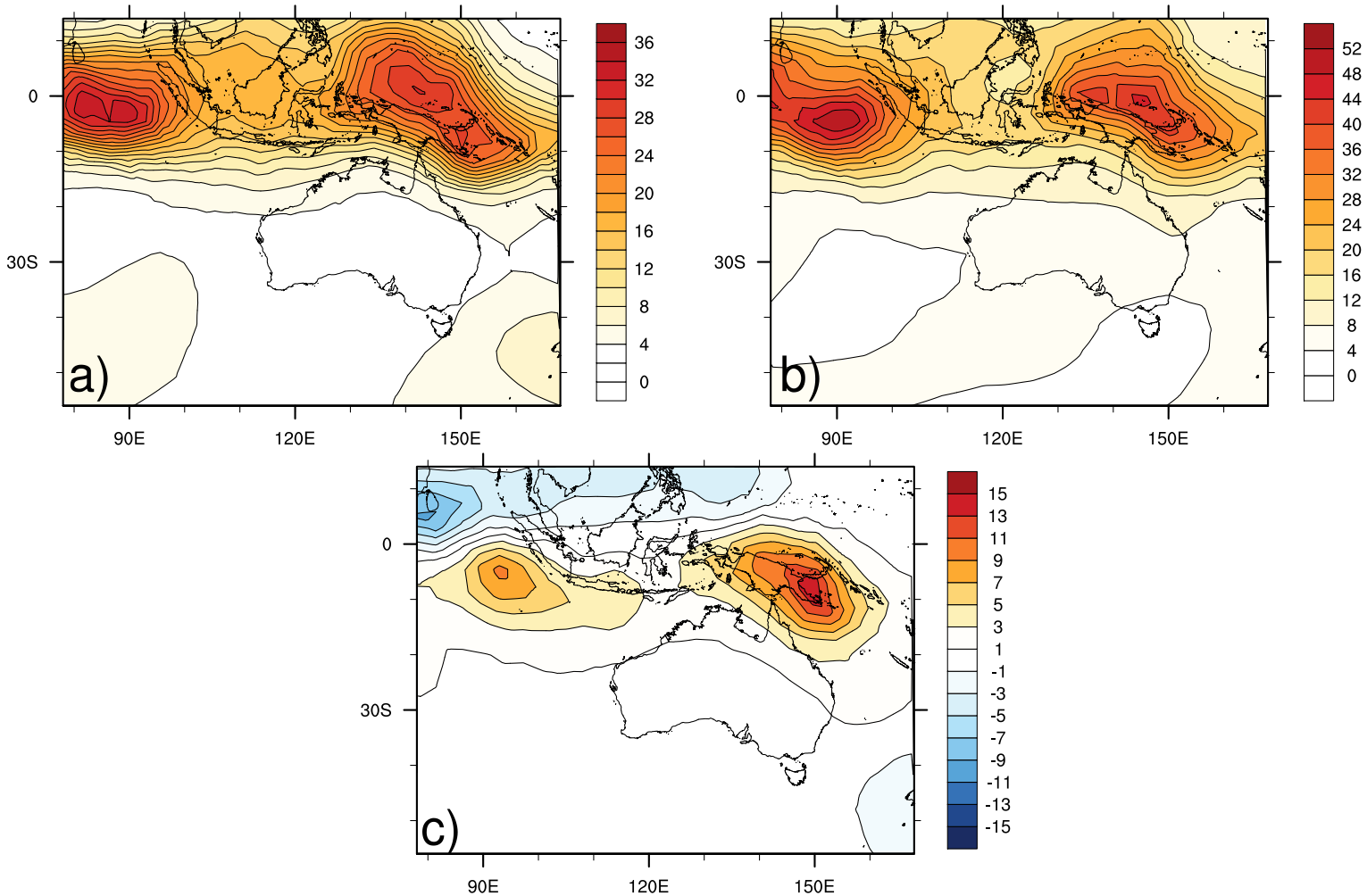
Estimation of a local correlation function on the model grid

Local estimation of C with LCT



Fit with a Local Correlation Tensor (LCT)

Local estimation of C with LCT



Local Correlation Tensor horizontal components
interpolated over the model grid
(a: zonal direction, b: meridional direction, c: off-diagonal)

Local estimation of C with LCT

LCT summary:

1. Define a homogeneous subgrid.
2. For each subgrid point, compute the local correlation function on the model grid.
3. Fit the raw correlation function with a Local Correlation Tensor (LCT)-based parametrized function (e.g. Gaspari-Cohn, 1999 or Matern function) to get the LCT components.
4. Optionally filter the LCT components on the subgrid.
5. Interpolate the LCT components over the model grid.

Outline

Sample covariance

An ensemble of N forecasts $\{\mathrm{x}_p^b\}$ is used to estimate the sample covariance matrix $\widetilde{\mathrm{B}}$:

$$\widetilde{\mathrm{B}} = \frac{1}{N-1} \sum_{p=1}^{N} \delta\mathrm{x}_p^b \delta\mathrm{x}_p^{b\mathrm{T}}$$

where $\delta\mathrm{x}_p^b$ is the p^{th} ensemble perturbation:

$$\delta\mathrm{x}_p^b = \mathrm{x}_p^b - \langle \mathrm{x}^b \rangle \quad \text{and} \quad \langle \mathrm{x}^b \rangle = \frac{1}{N} \sum_{p=1}^{N} \mathrm{x}_p^b$$

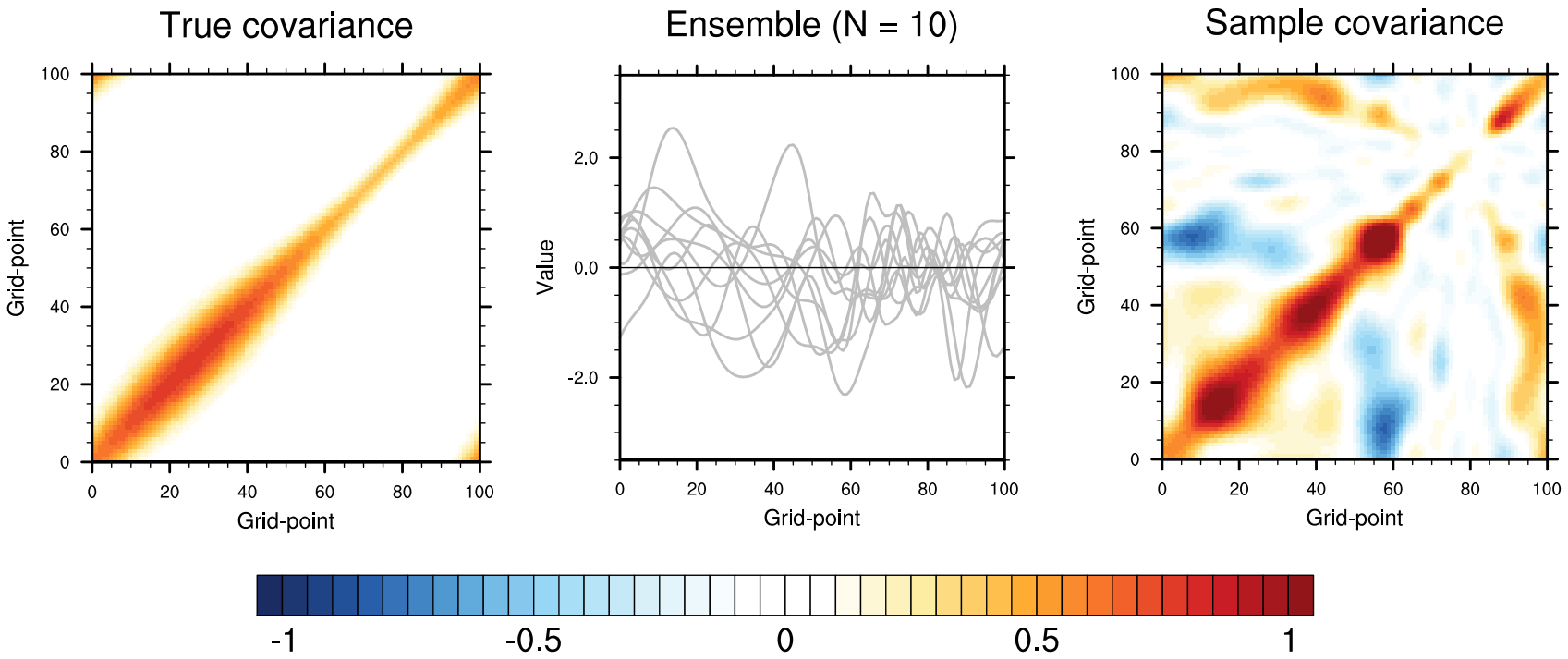
Asymptotic sample covariance: $\widetilde{\mathrm{B}}^\star = \lim_{N \to \infty} \widetilde{\mathrm{B}}$

Since the ensemble size $N < \infty$, $\widetilde{\mathrm{B}}$ is affected by sampling noise:

$$\widetilde{\mathrm{B}}^e = \widetilde{\mathrm{B}} - \widetilde{\mathrm{B}}^\star$$

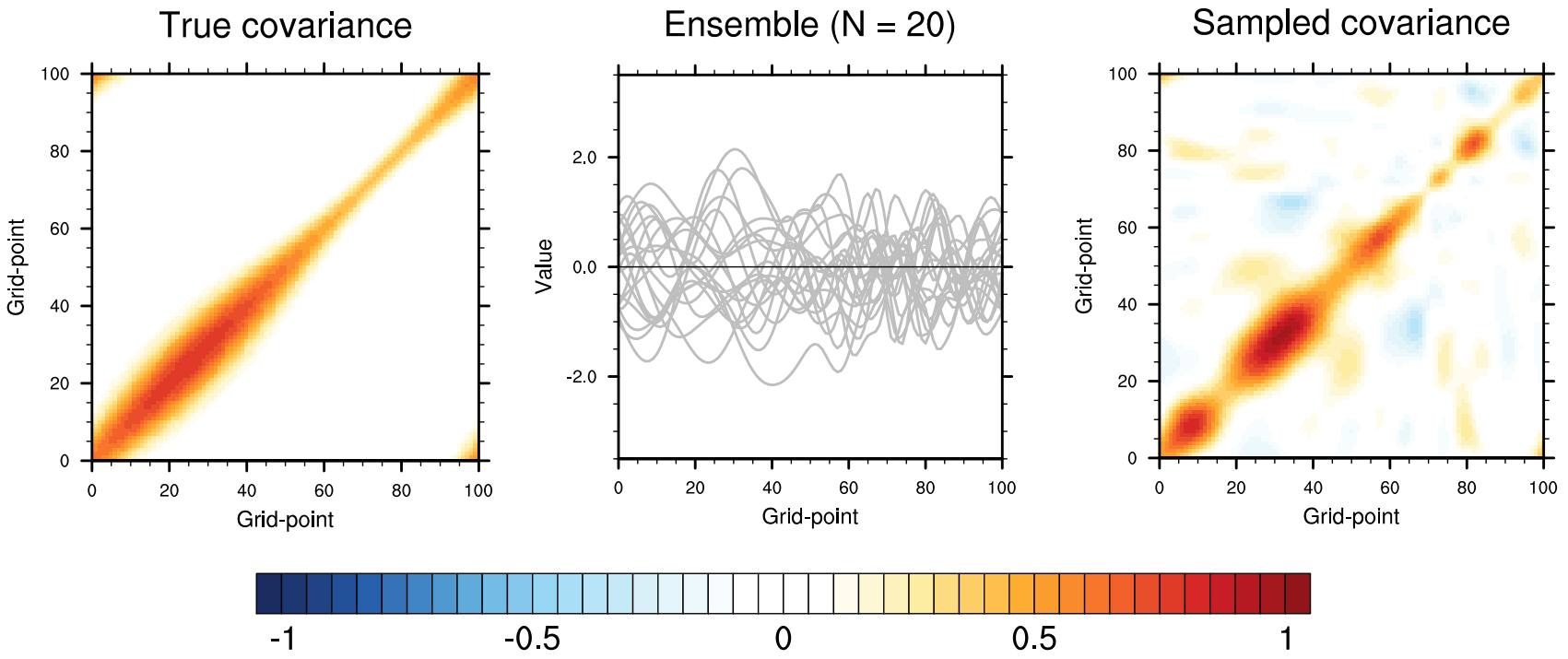
Sample covariance

Sampling noise strongly depends on the ensemble size:



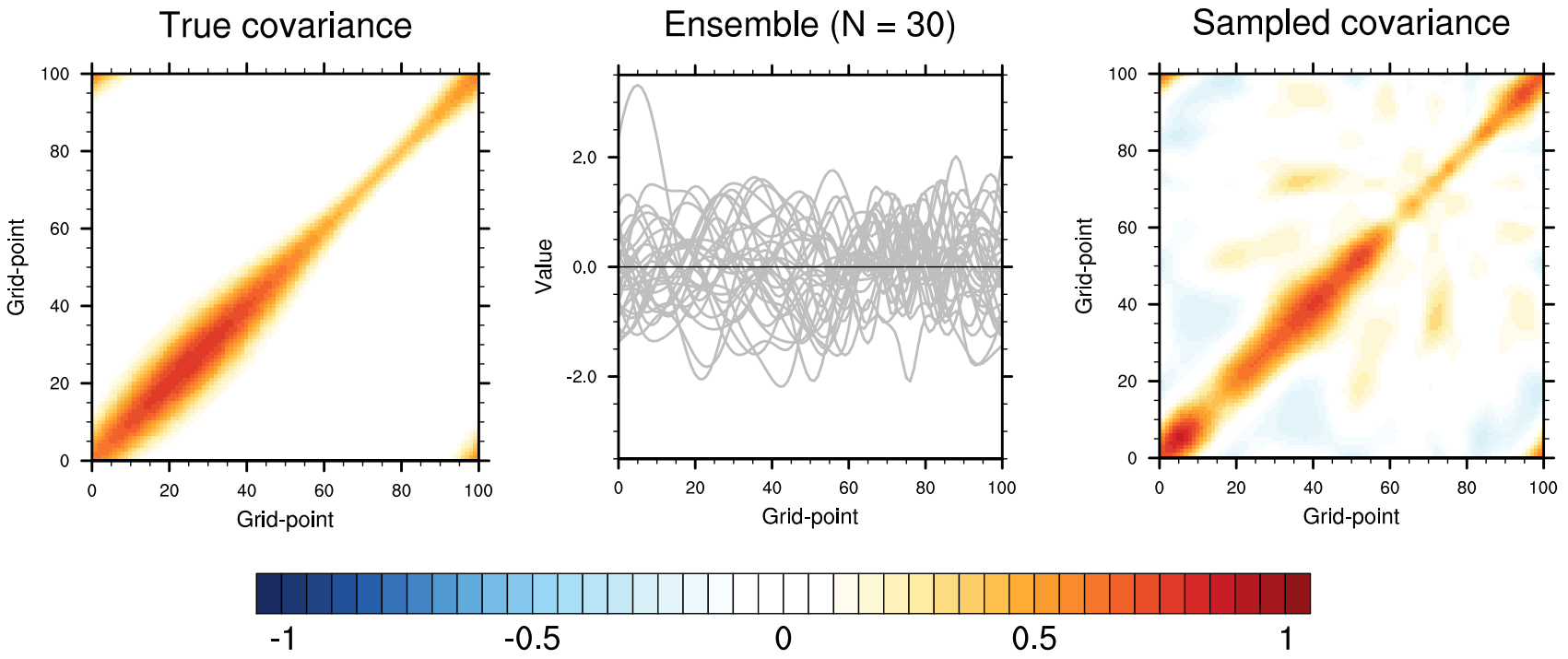
Sample covariance

Sampling noise strongly depends on the ensemble size:



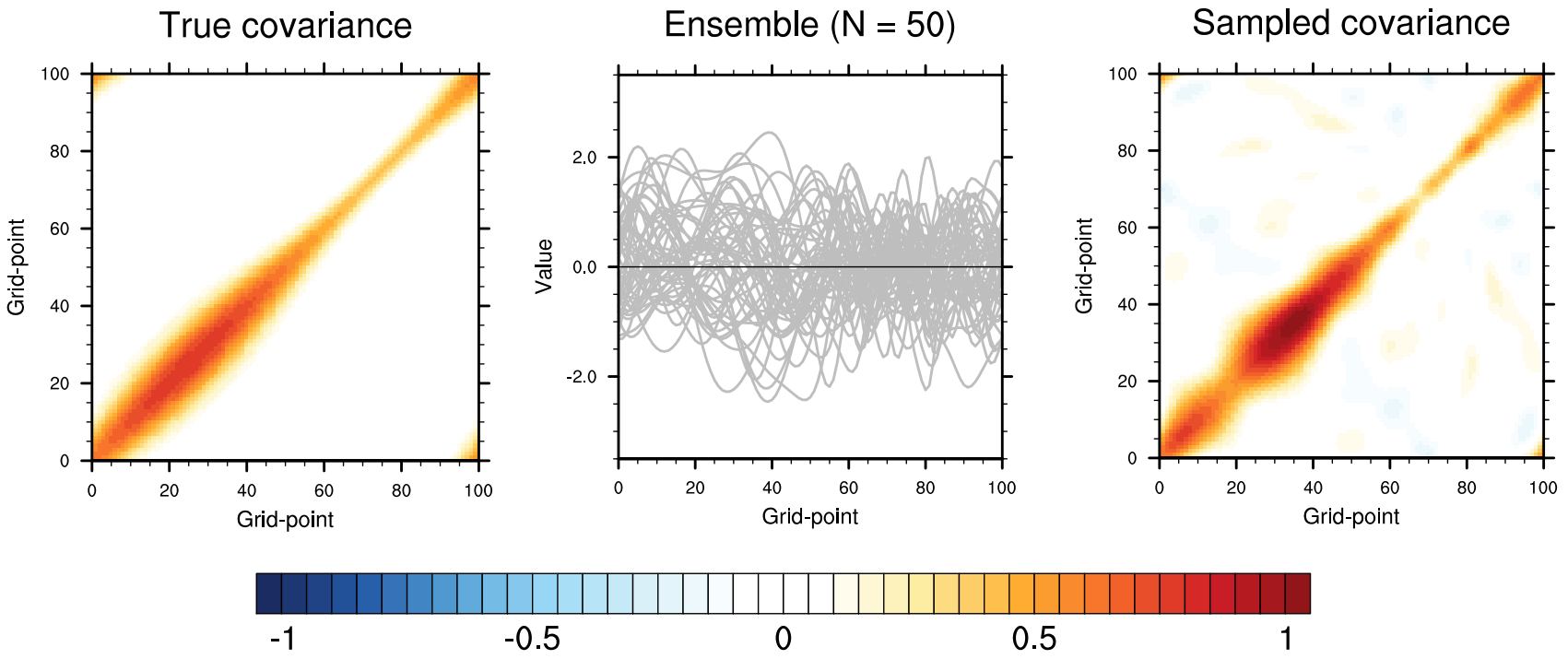
Sample covariance

Sampling noise strongly depends on the ensemble size:



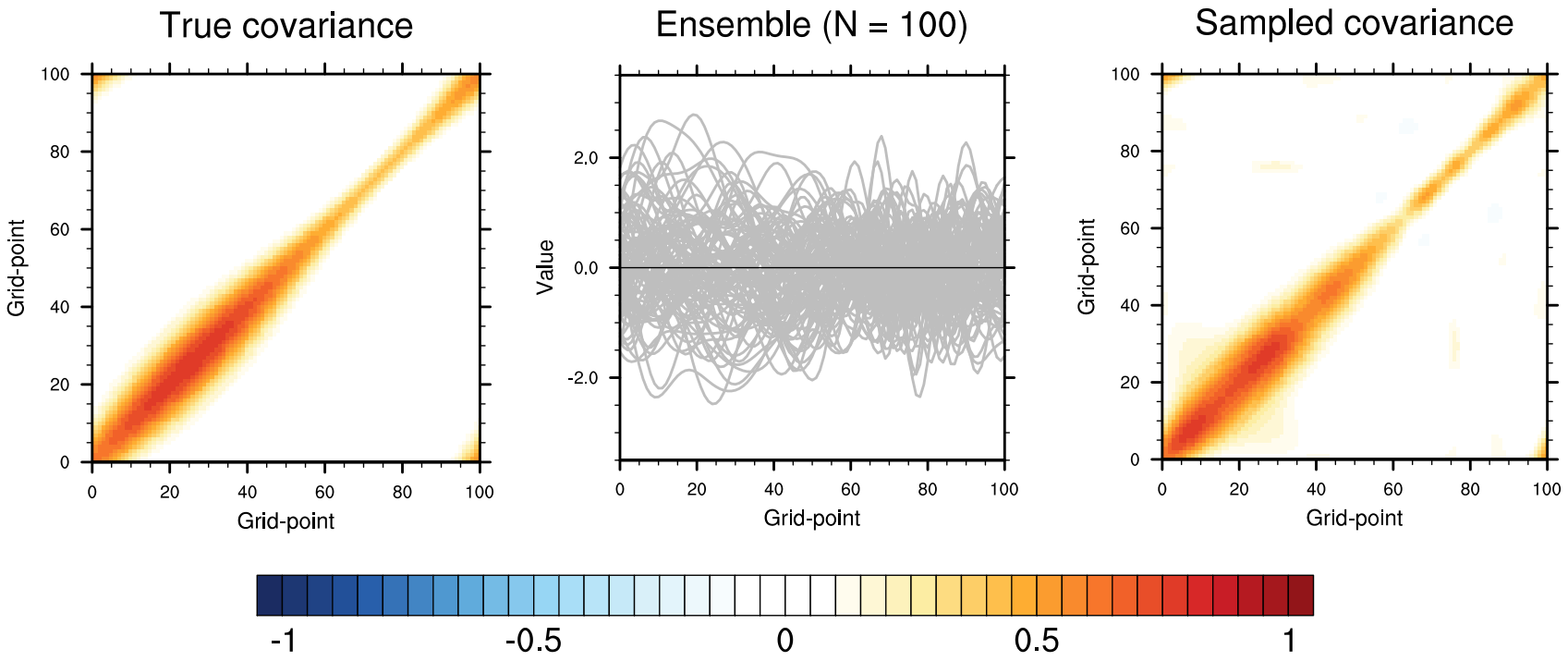
Sample covariance

Sampling noise strongly depends on the ensemble size:



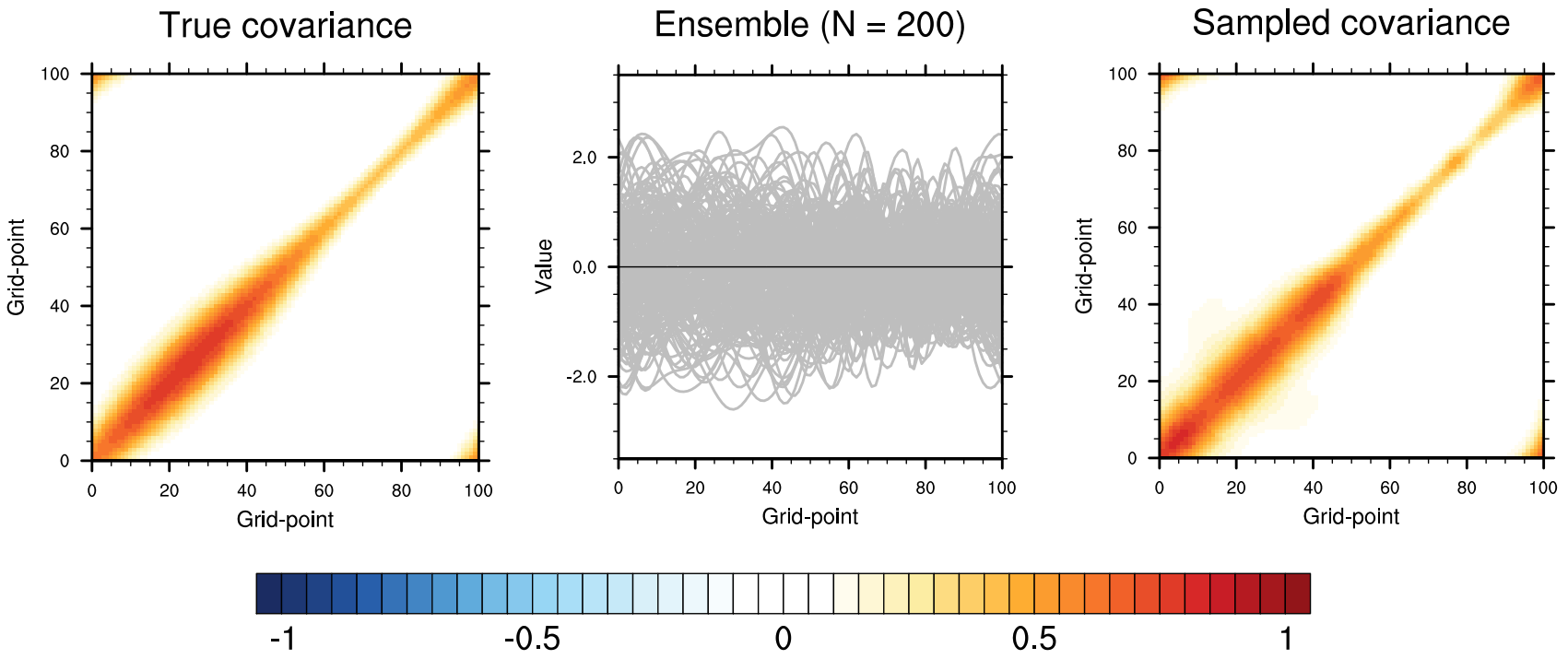
Sample covariance

Sampling noise strongly depends on the ensemble size:



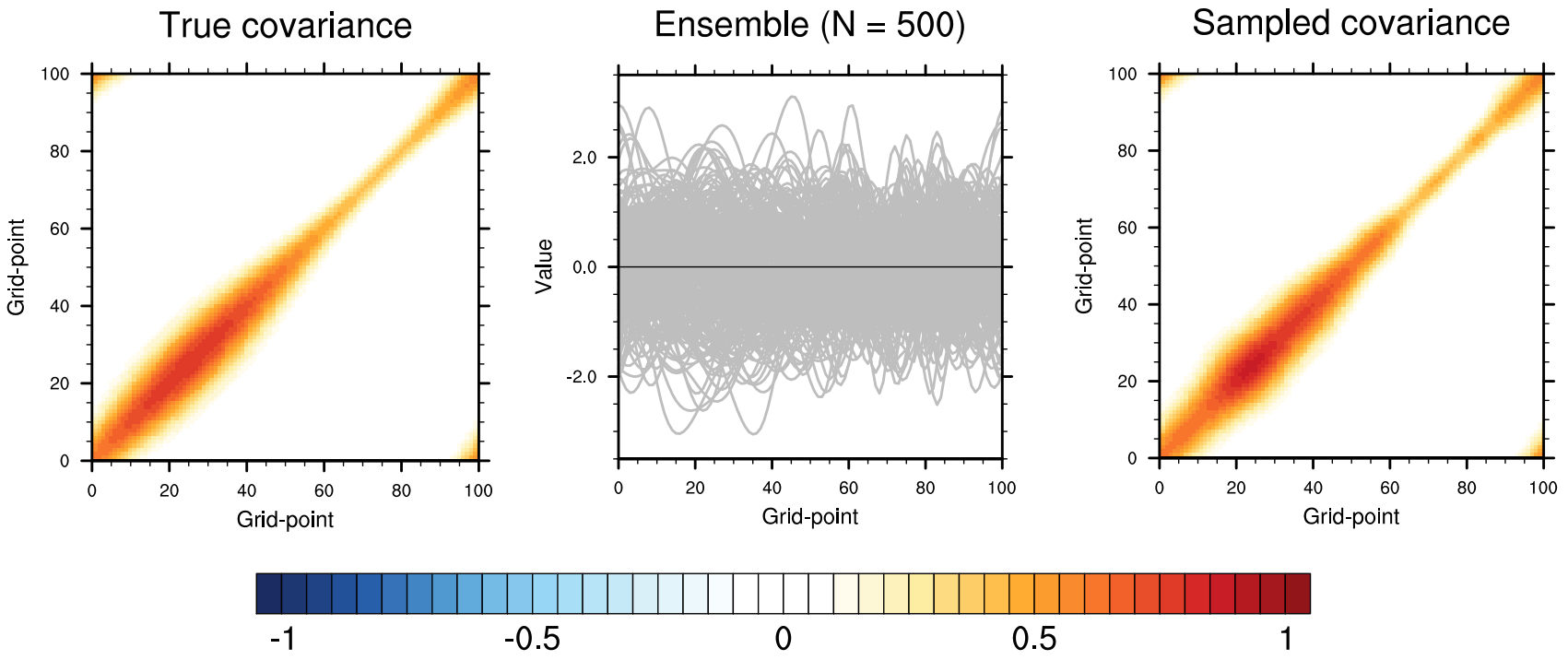
Sample covariance

Sampling noise strongly depends on the ensemble size:



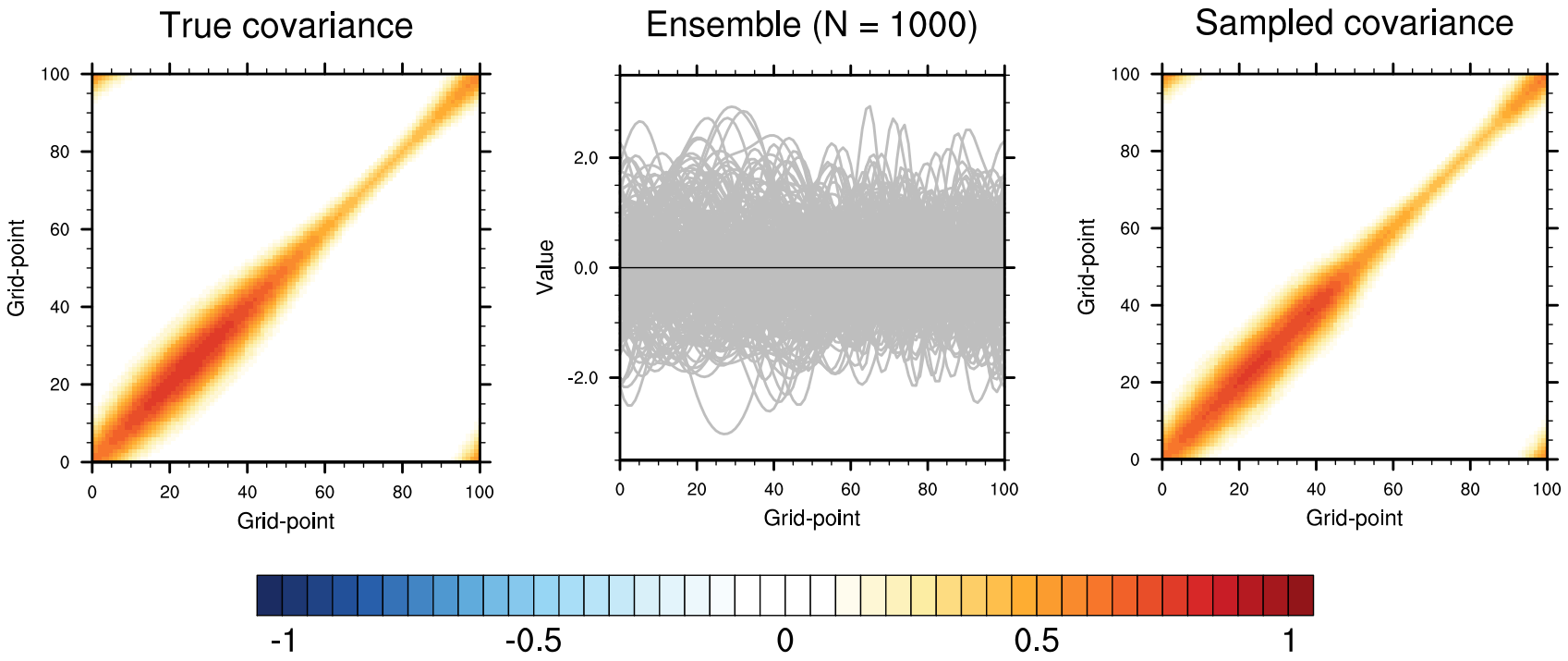
Sample covariance

Sampling noise strongly depends on the ensemble size:



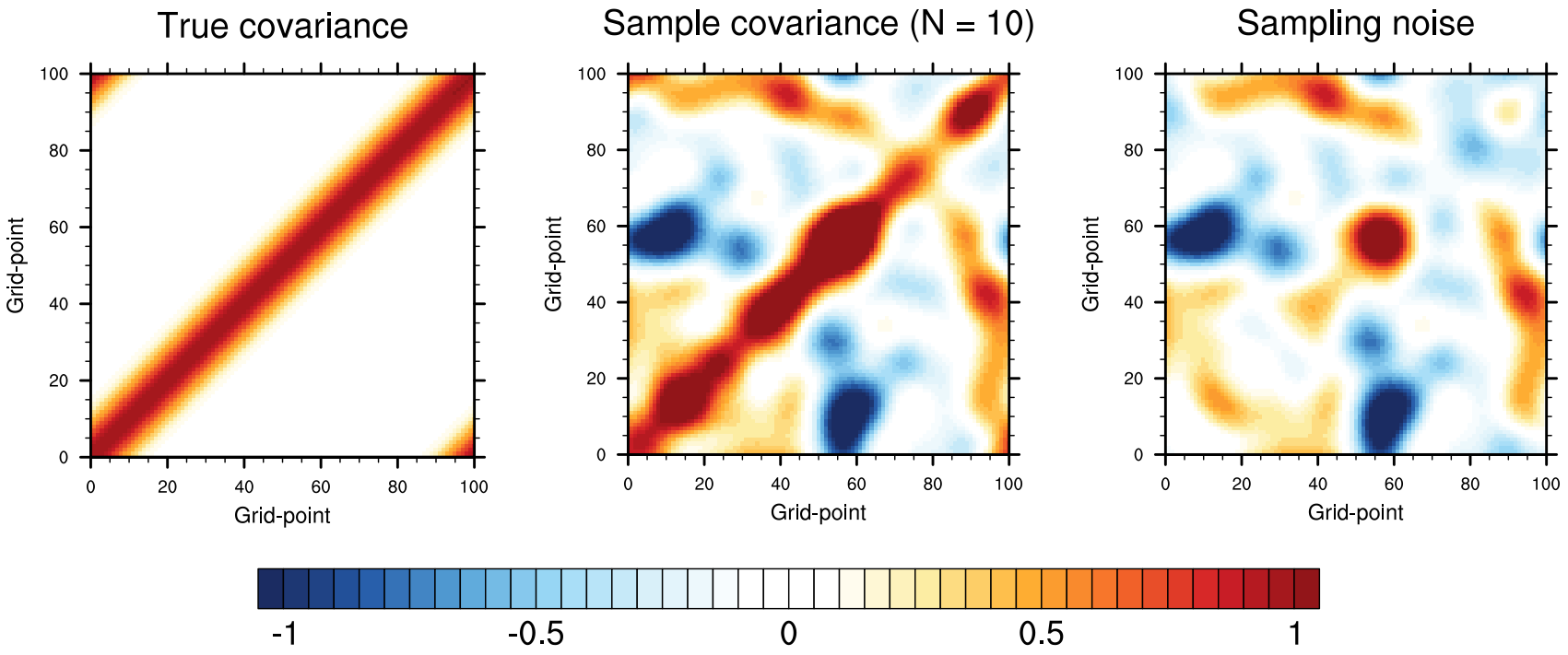
Sample covariance

Sampling noise strongly depends on the ensemble size:



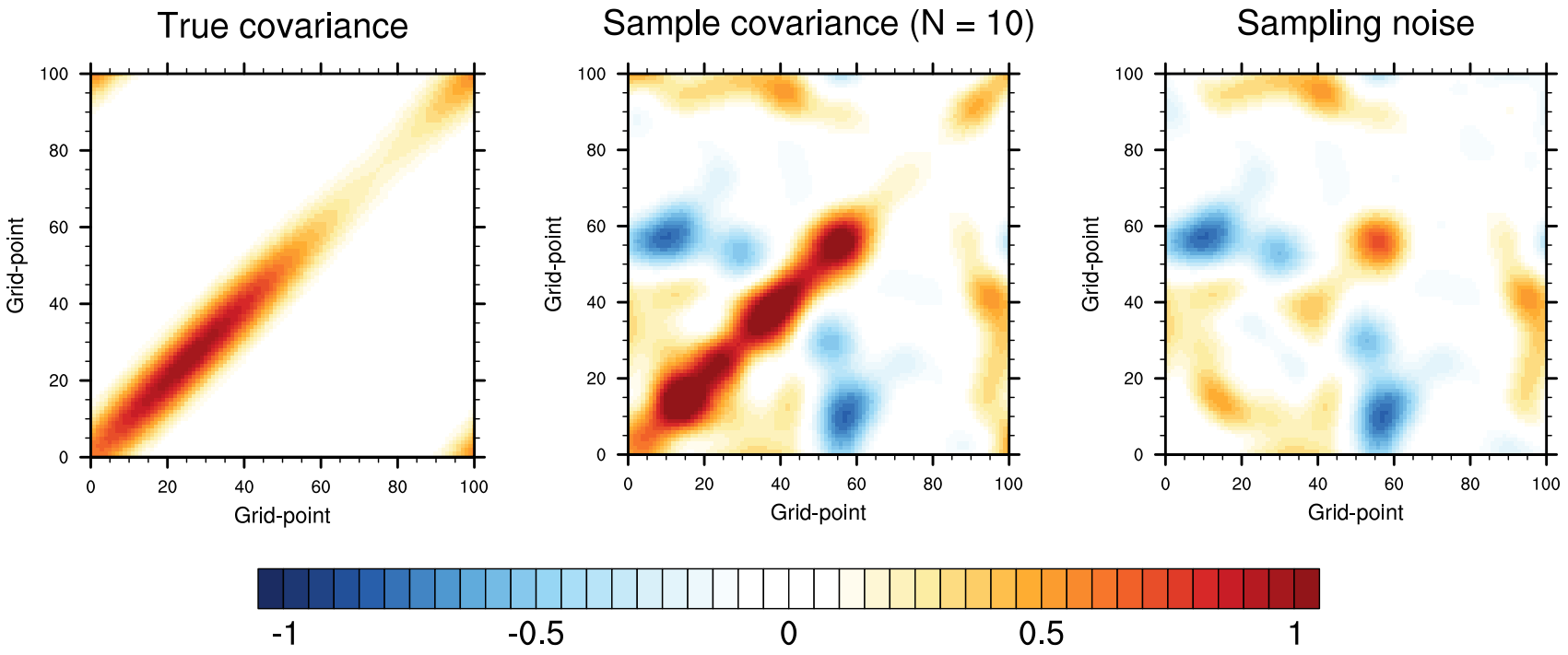
Sampling noise properties

Homogeneous variance / length-scale



Sampling noise properties

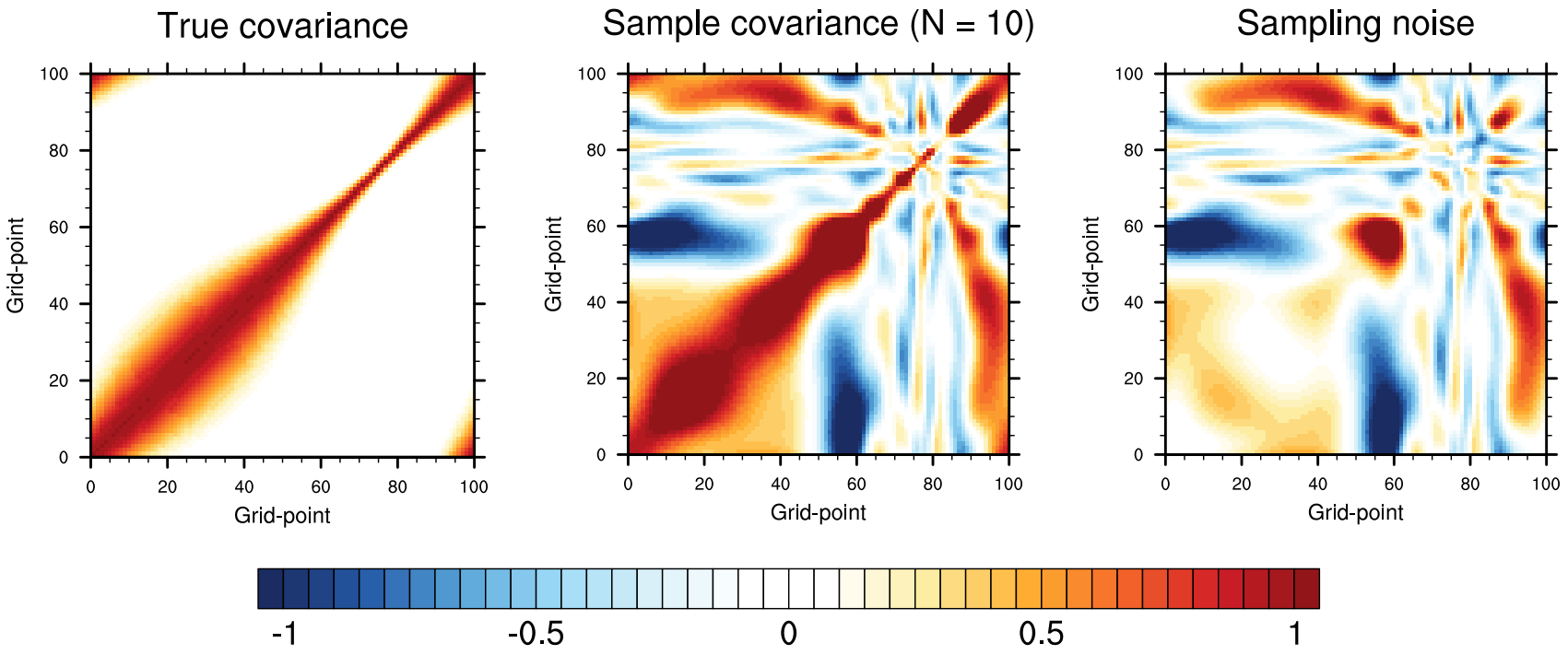
Heterogeneous variance / homogeneous length-scale



Sampling noise amplitude related to the asymptotic variance

Sampling noise properties

Homogeneous variance / heterogeneous length-scale



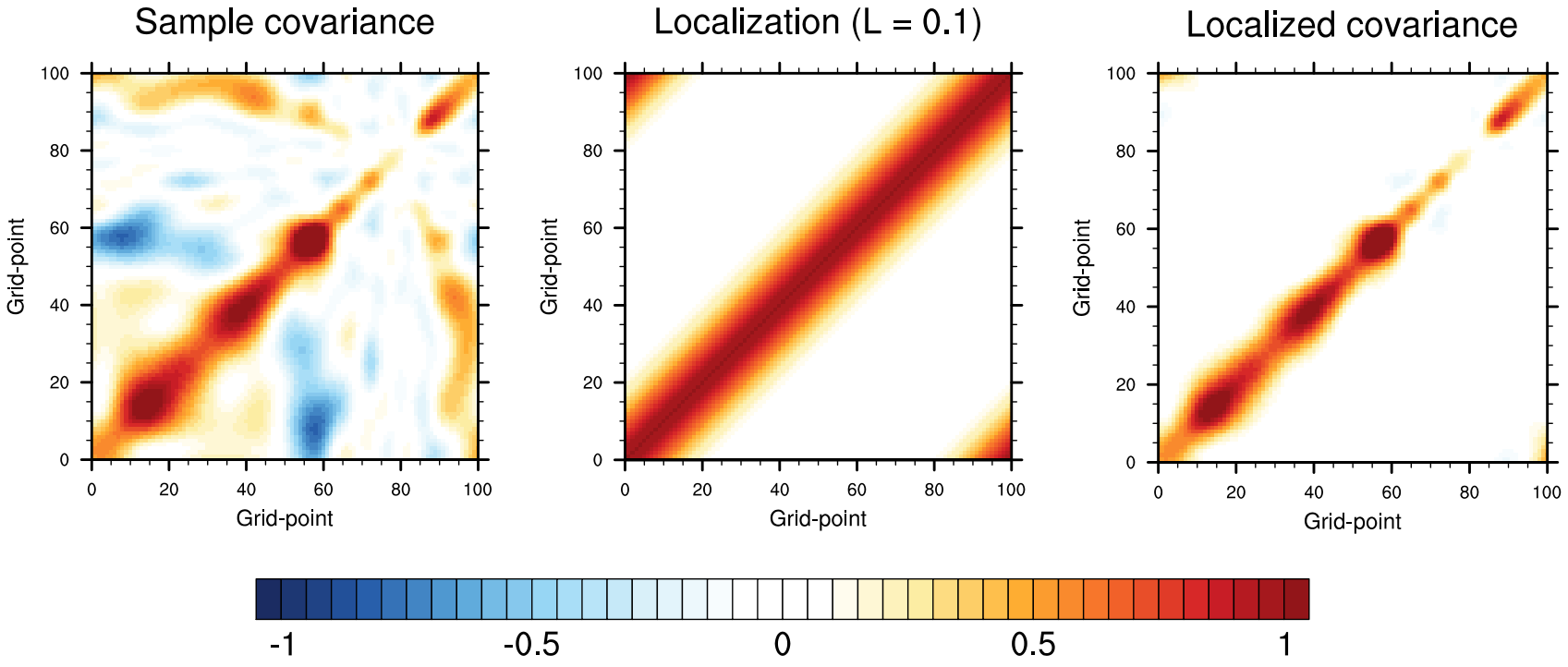
Sampling noise length-scale related to the asymptotic length-scale

Localized covariance

Sampling noise on $\widetilde{\mathbf{B}}$ can be damped via a Schur product (element-by-element) with a localization matrix L:

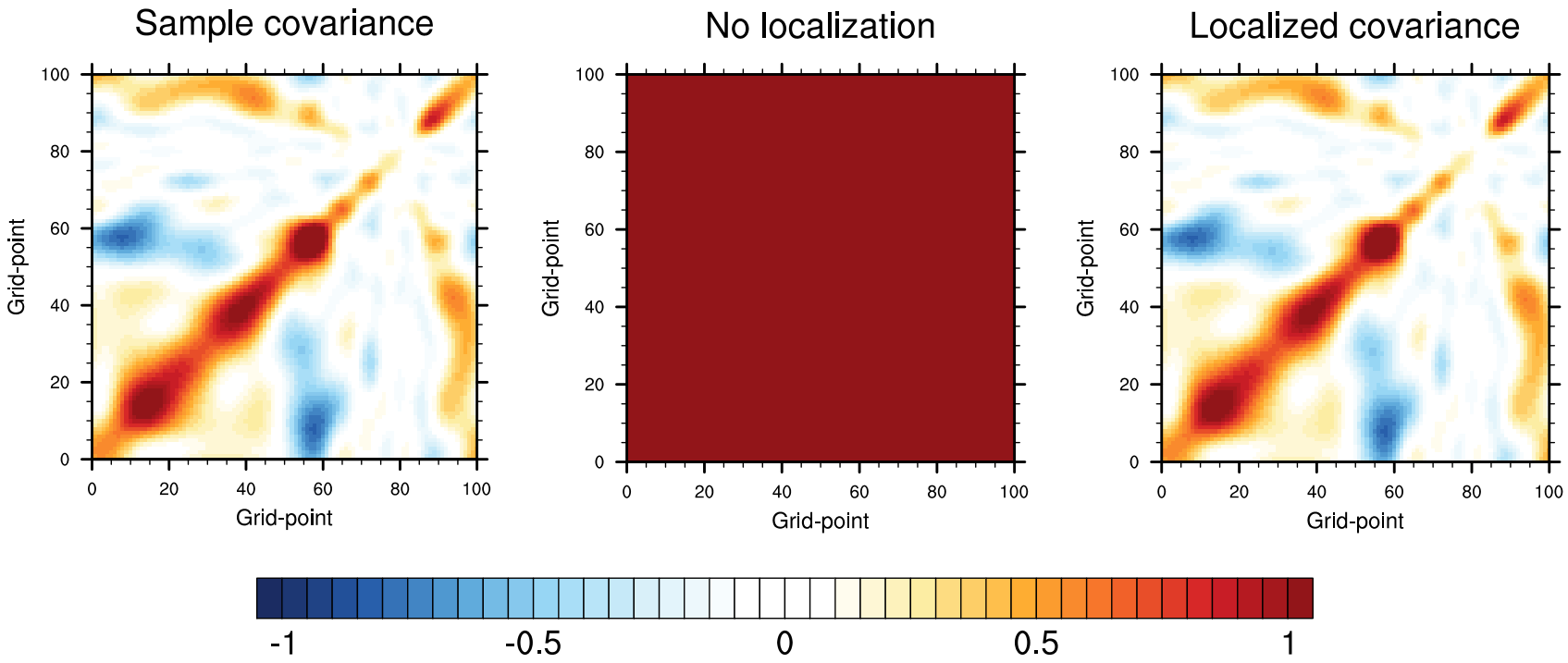
$$\mathbf{B} = \mathbf{L} \circ \widetilde{\mathbf{B}} \quad \Leftrightarrow \quad B_{ij} = L_{ij} \widetilde{B}_{ij}$$

In practice, L damps the long-distance correlations that are small and more affected by sampling noise (hence the "localization").



Localization: what is the optimal length-scale?

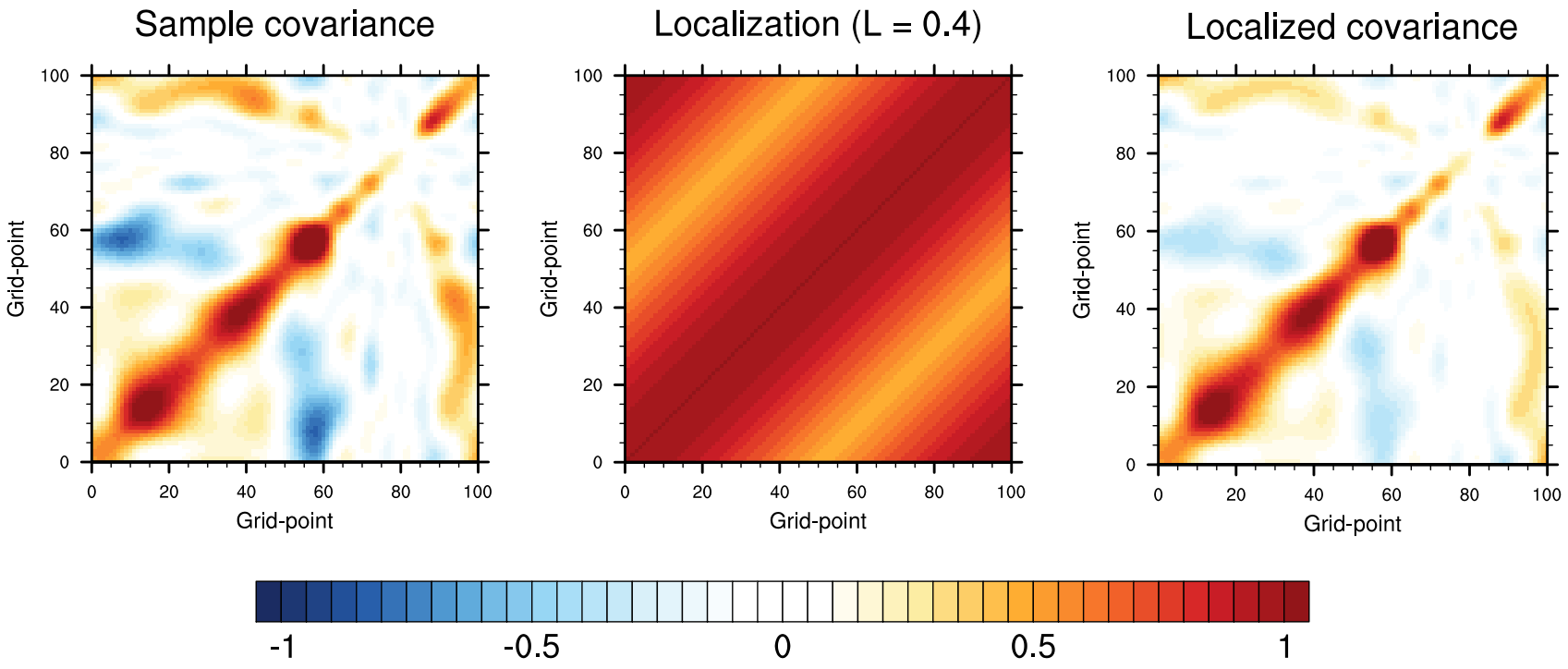
The localization length-scale is critical to remove the sampling noise while keeping the relevant covariance signal:



No impact

Localization: what is the optimal length-scale?

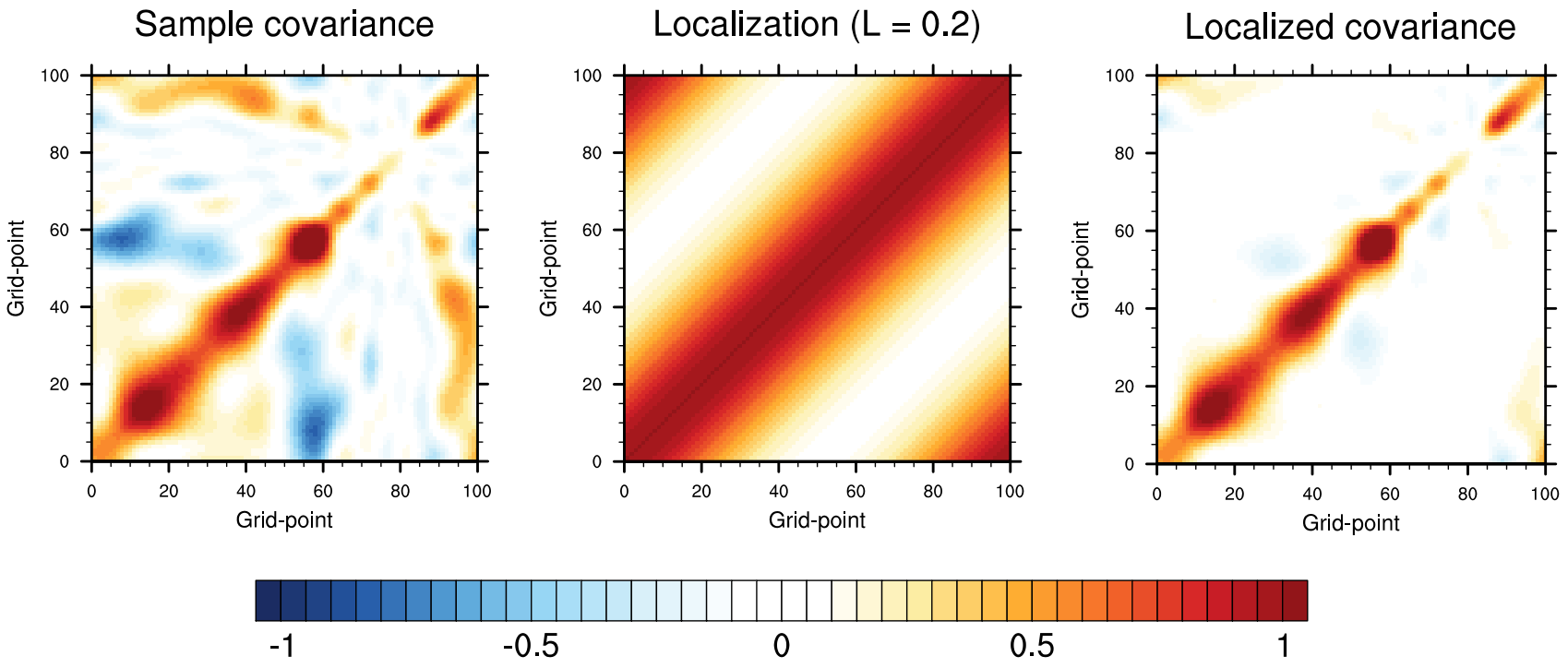
The localization length-scale is critical to remove the sampling noise while keeping the relevant covariance signal:



Start reducing the sampling noise...

Localization: what is the optimal length-scale?

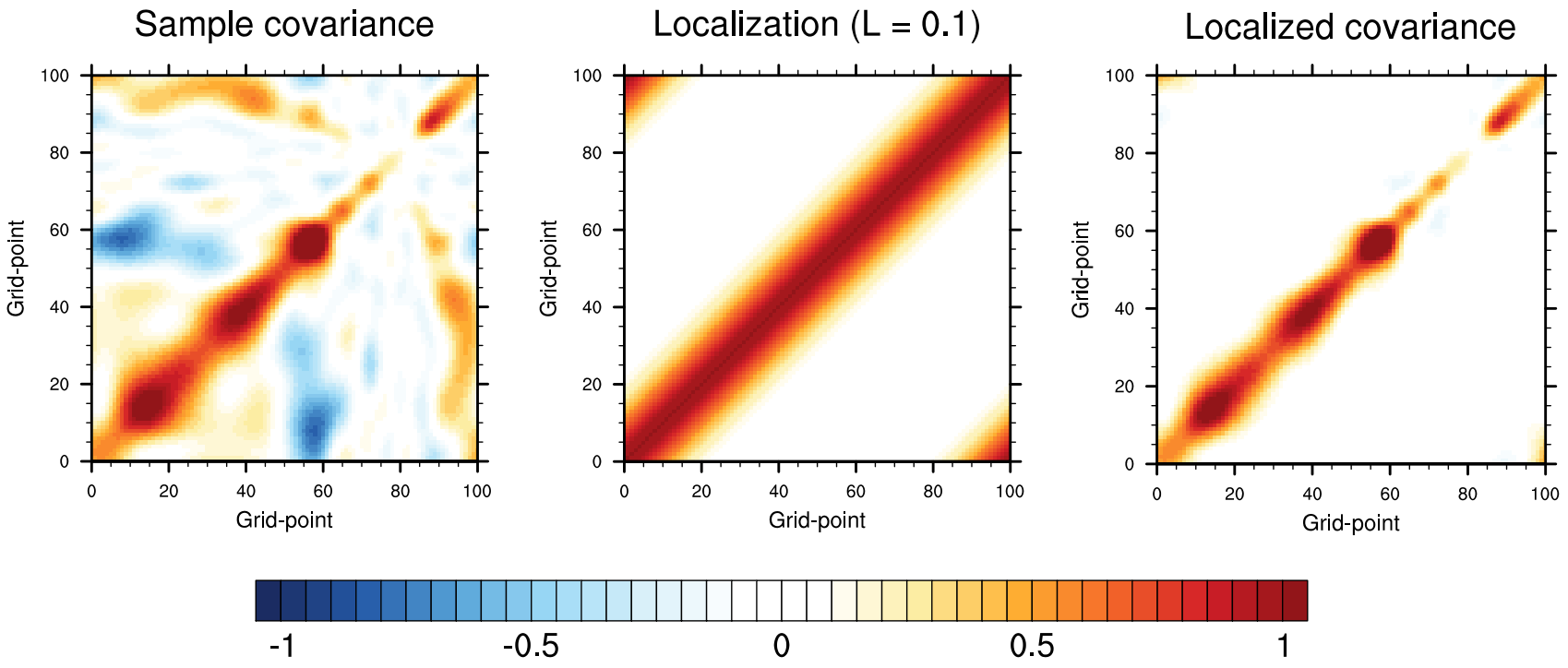
The localization length-scale is critical to remove the sampling noise while keeping the relevant covariance signal:



Less and less sampling noise...

Localization: what is the optimal length-scale?

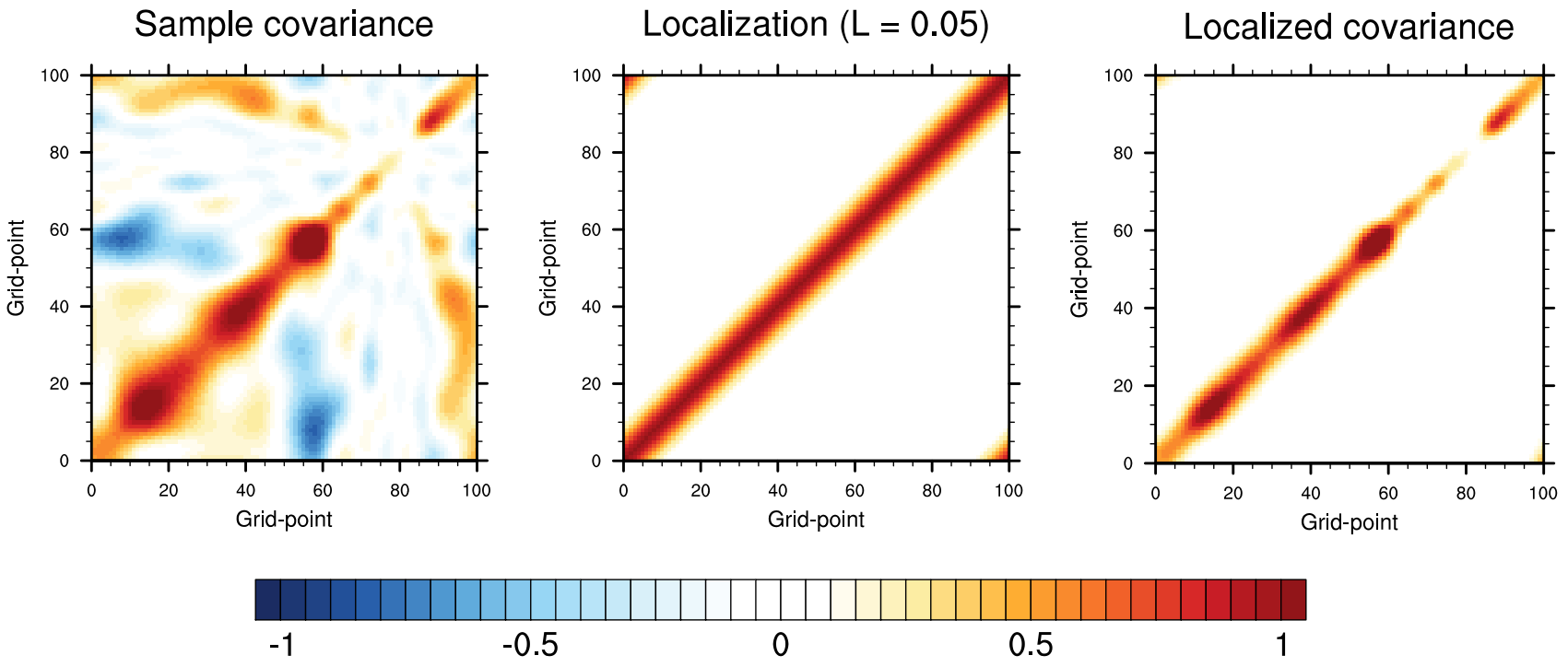
The localization length-scale is critical to remove the sampling noise while keeping the relevant covariance signal:



Good ! Almost no sampling noise anymore...

Localization: what is the optimal length-scale?

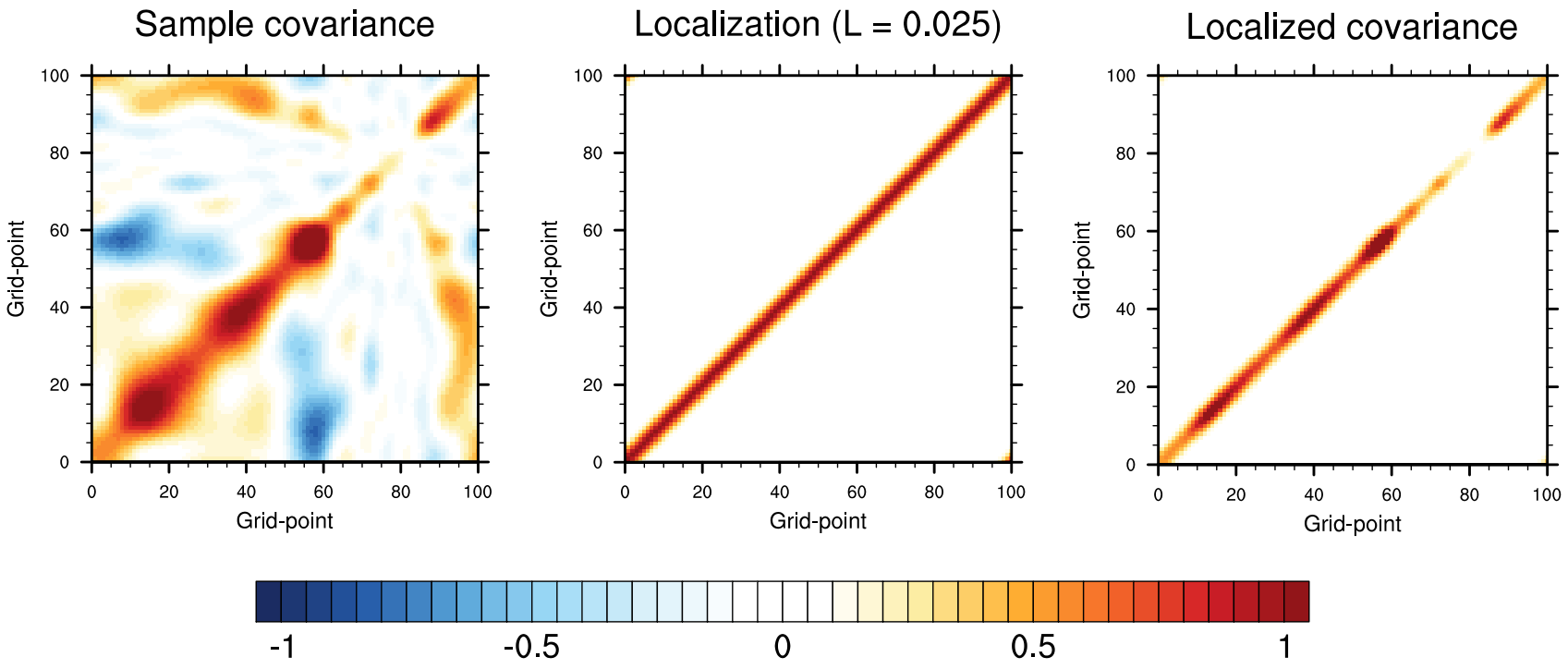
The localization length-scale is critical to remove the sampling noise while keeping the relevant covariance signal:



Well, we are loosing some signal now...

Localization: what is the optimal length-scale?

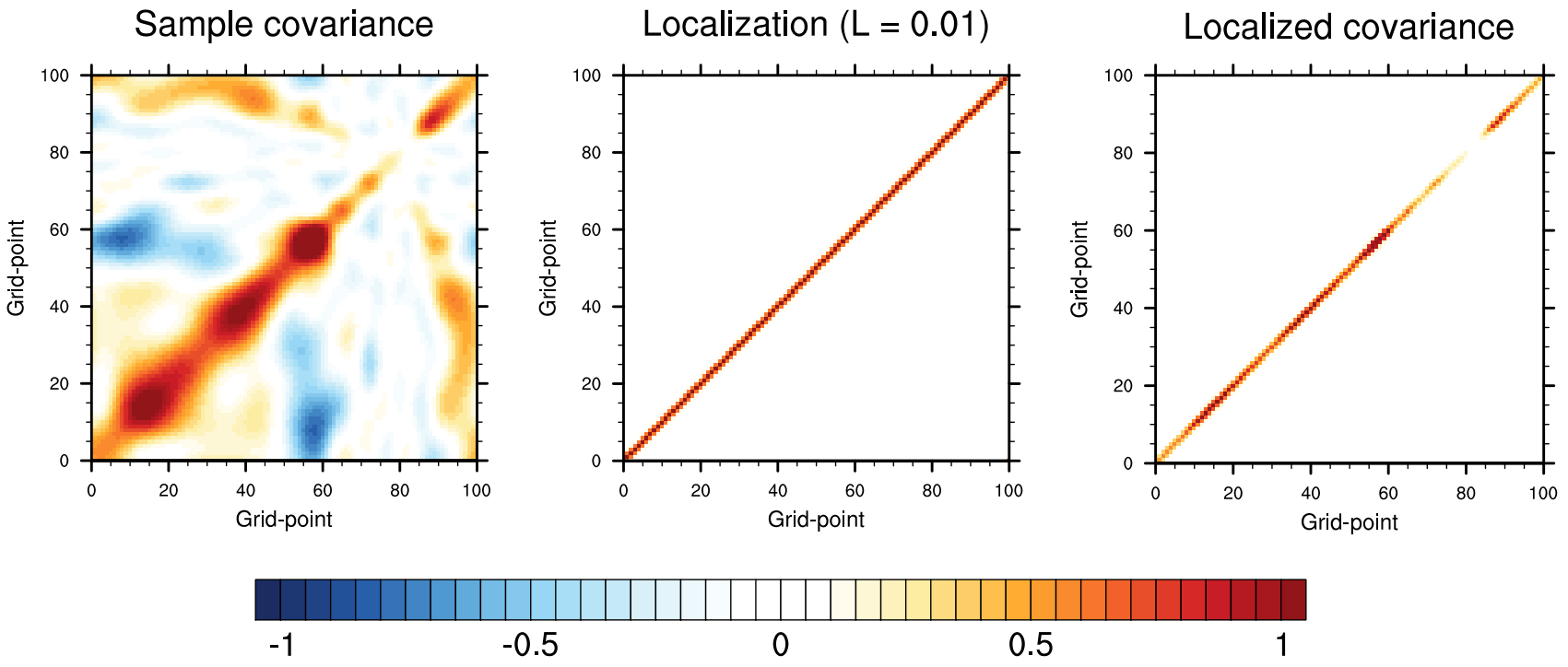
The localization length-scale is critical to remove the sampling noise while keeping the relevant covariance signal:



Hey, stop loosing signal !

Localization: what is the optimal length-scale?

The localization length-scale is critical to remove the sampling noise while keeping the relevant covariance signal:



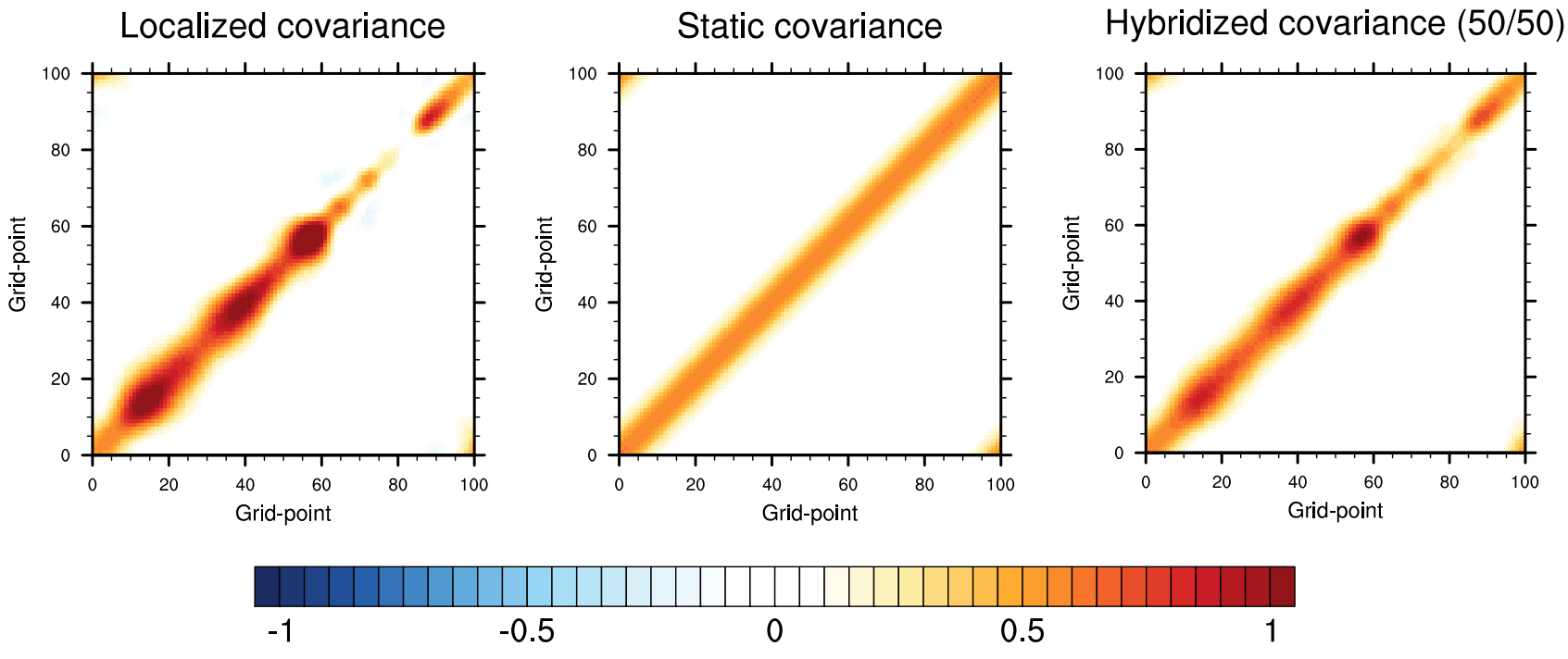
No more signal !

Localized and hybridized covariance

Deficiencies of the localized covariance matrix can be corrected via a hybridization with a static covariance matrix:

$$\mathbf{B} = \underbrace{\beta^{e2}\, \mathbf{L} \circ \widetilde{\mathbf{B}}}_{\text{“ensemble” term}} + \underbrace{\beta^{s2}\, \mathbf{B}^{s}}_{\text{“static” term}}$$

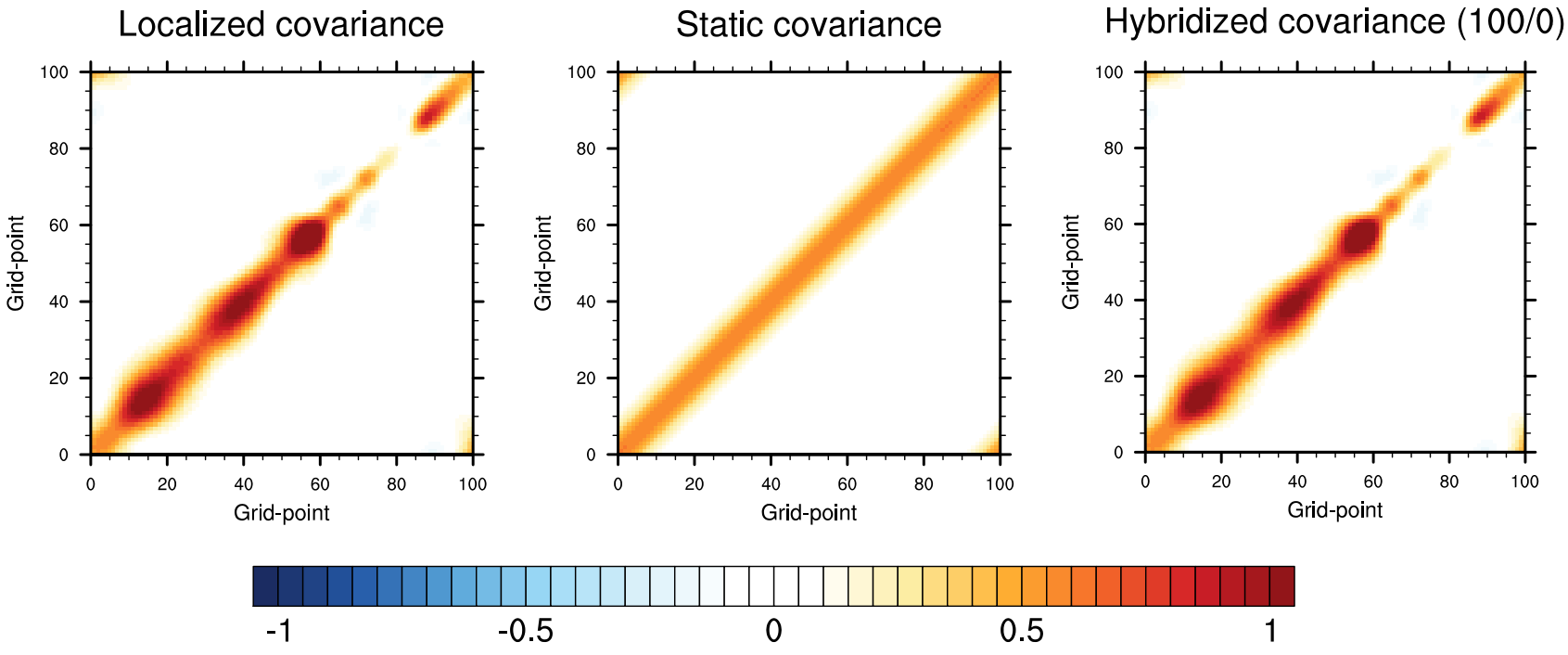
For a homogeneous $\mathbf{B}^{s}$:



Hybridization: what are the optimal weights?

Localization + hybridation:

$$\mathbf{B} = \underbrace{\beta^{e2}\, \mathbf{L} \circ \widetilde{\mathbf{B}}}_{\text{“ensemble” term}} + \underbrace{\beta^{s2}\, \mathbf{B}^{s}}_{\text{“static” term}}$$

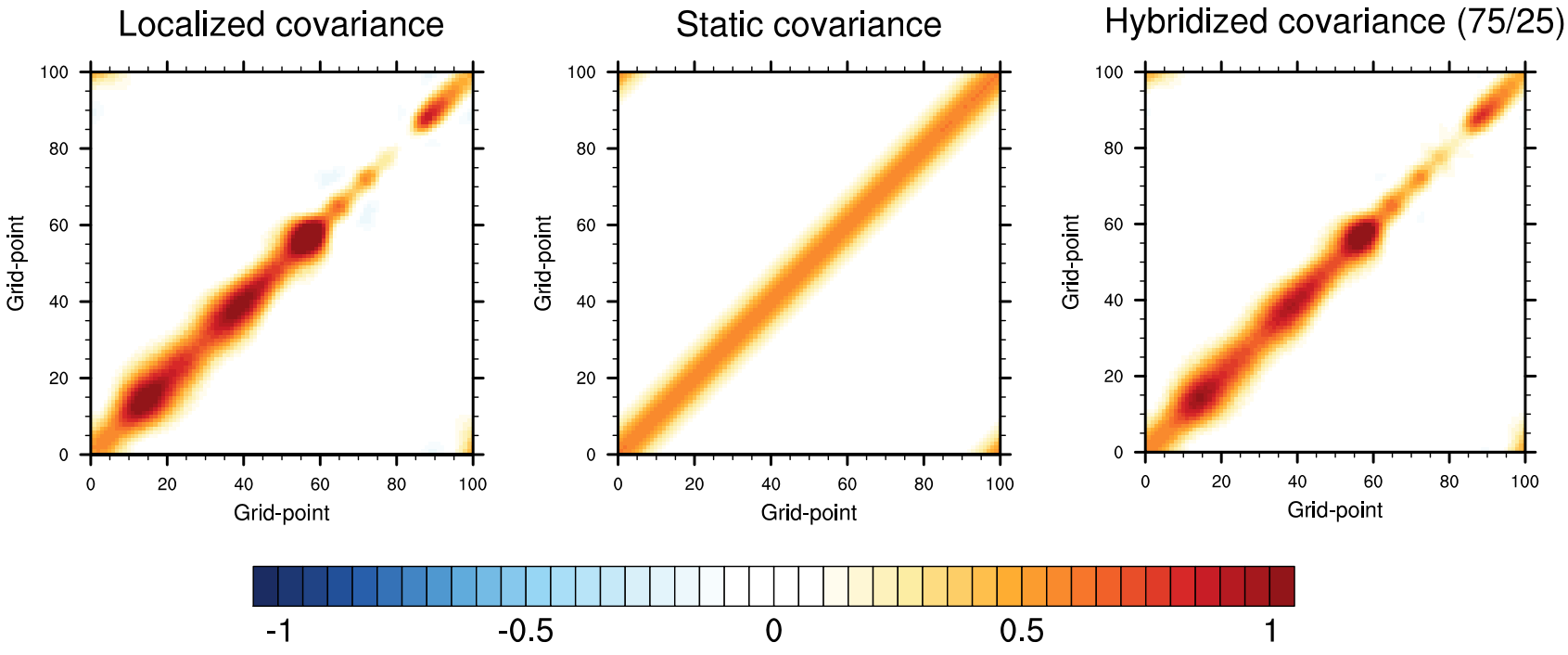


100% on the localized covariance matrix

Hybridization: what are the optimal weights?

Localization + hybridation:

$$\mathbf{B} = \underbrace{\beta^{e2}\, \mathbf{L} \circ \widetilde{\mathbf{B}}}_{\text{“ensemble” term}} + \underbrace{\beta^{s2}\, \mathbf{B}^{s}}_{\text{“static” term}}$$

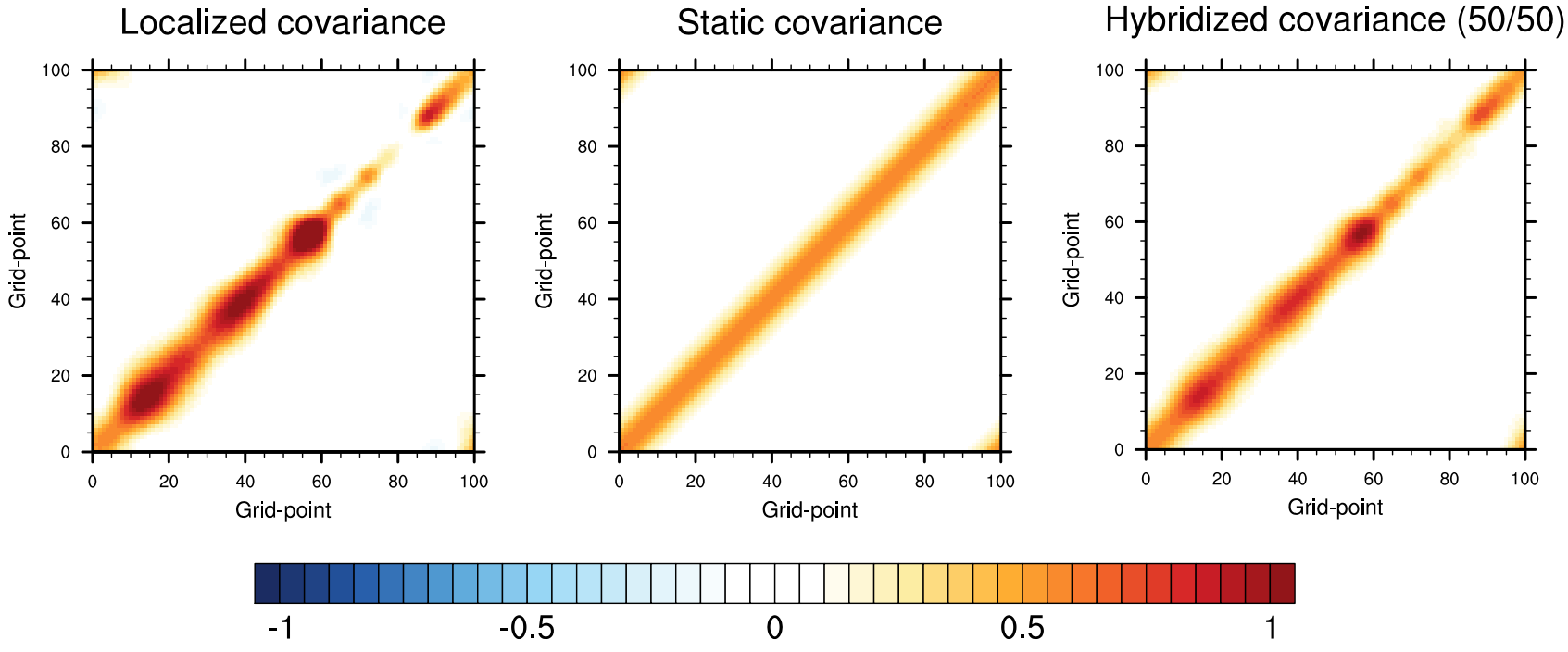


75% on the localized covariance matrix

Hybridization: what are the optimal weights?

Localization + hybridation:

$$\mathbf{B} = \underbrace{\beta^{e2}\, \mathbf{L} \circ \widetilde{\mathbf{B}}}_{\text{“ensemble” term}} + \underbrace{\beta^{s2}\, \mathbf{B}^{s}}_{\text{“static” term}}$$

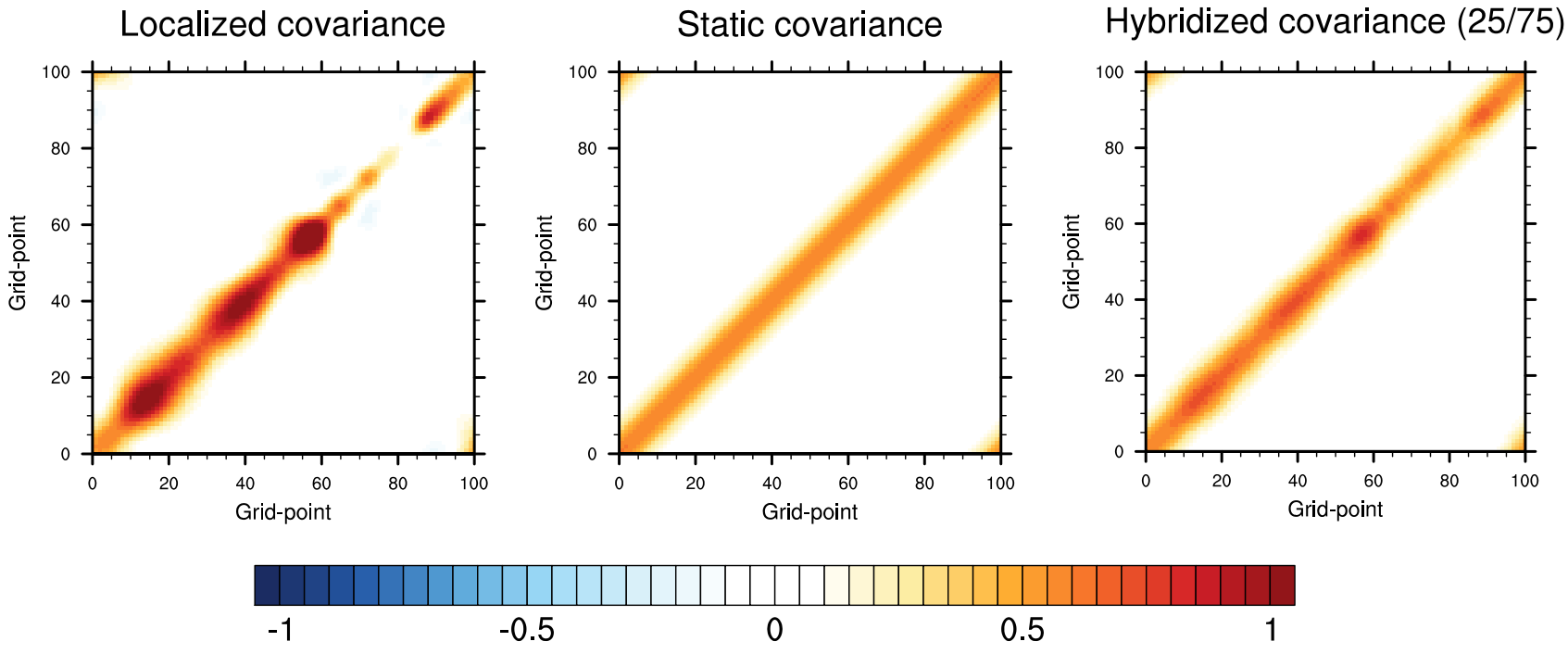


Equal weights on both covariance matrices

Hybridization: what are the optimal weights?

Localization + hybridation:

$$\mathbf{B} = \underbrace{\beta^{e2}\, \mathbf{L} \circ \widetilde{\mathbf{B}}}_{\text{“ensemble” term}} + \underbrace{\beta^{s2}\, \mathbf{B}^{s}}_{\text{“static” term}}$$

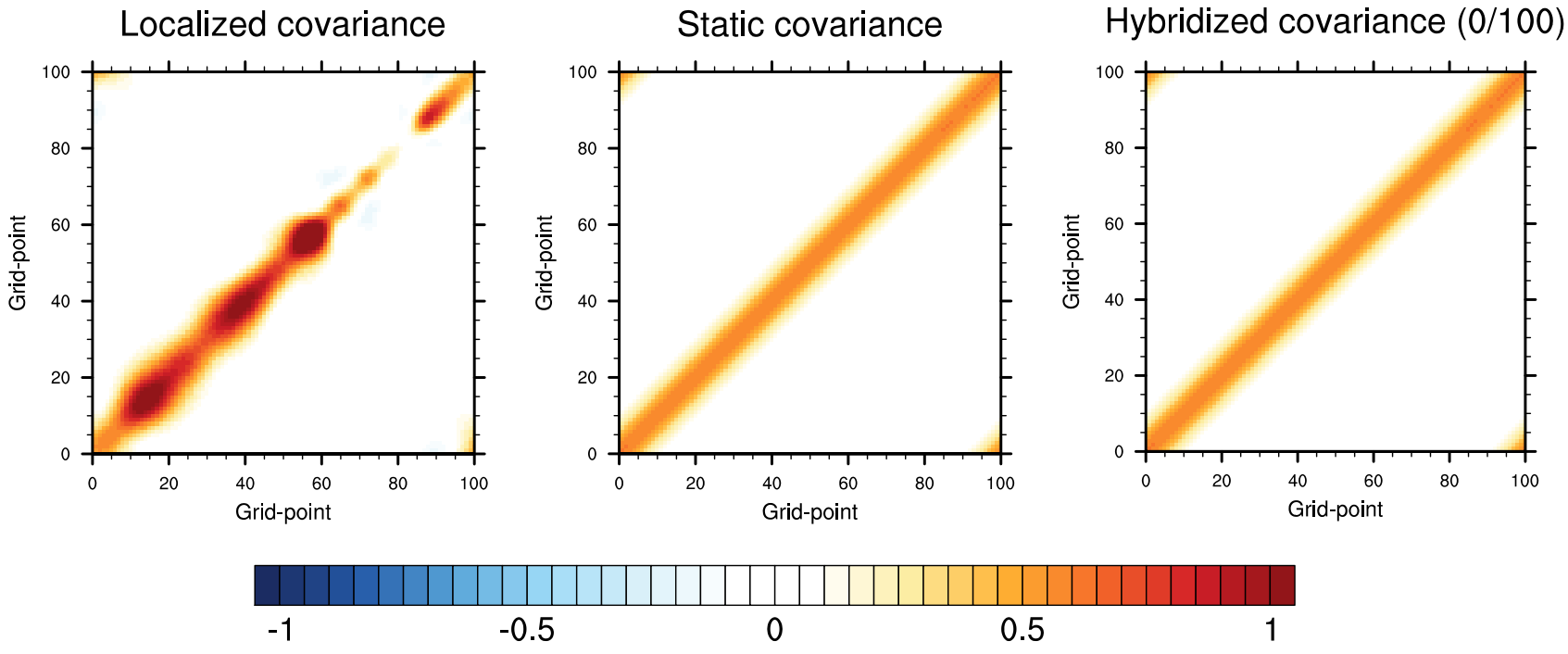


75% on the static covariance matrix

Hybridization: what are the optimal weights?

Localization + hybridation:

$$\mathbf{B} = \underbrace{\beta^{e2}\, \mathbf{L} \circ \widetilde{\mathbf{B}}}_{\text{“ensemble” term}} + \underbrace{\beta^{s2}\, \mathbf{B}^{s}}_{\text{“static” term}}$$



100% on the static covariance matrix

How to optimize localization and hybridization?

Existing methods are empirical and costly (e.g. OSSE, brute-force optimization). We need a new method that:

- uses only ensemble members,
- is affordable for high-dimensional systems,
- can be generic enough to be run with all kinds of grids.

Principle :

$$\underbrace{\mathbf{B}}_{\substack{\text{Localized/hybridized}\\ \text{covariance}}} = \underbrace{\beta^{e2}}_{\substack{\text{Ensemble}\\ \text{weight}}} \underbrace{\mathbf{L}}_{\substack{\text{Localization}\\ \text{matrix}}} \circ \underbrace{\widetilde{\mathbf{B}}}_{\substack{\text{Sampled}\\ \text{covariance}}} + \underbrace{\beta^{s2}}_{\substack{\text{Static}\\ \text{weight}}} \underbrace{\mathbf{B}^{s}}_{\substack{\text{Static}\\ \text{covariance}}}$$

Localization + hybridization = linear filtering of $\widetilde{\mathbf{B}}$

How to optimize localization and hybridization?

Asymptotic sample covariance: $\widetilde{\mathbf{B}}^{\star} = \lim_{N \to \infty} \widetilde{\mathbf{B}}$

Residual noise (after localization/hybridization): $\mathbf{B} - \widetilde{\mathbf{B}}^{\star}$

Objectives:

- Express β^{e2}, β^{s2} and $\mathbf{L}$ minimizing the error $\mathbb{E}\left[\|\mathbf{B} - \widetilde{\mathbf{B}}^{\star}\|^2\right]$.

  $\rightarrow$ Linear filtering theory.

  Some statistics involve the asymptotic sample covariance.

- Express statistics on asymptotic quantities (unknown) with expected sample quantities (knowable).

  $\rightarrow$ Centered moments sampling theory (non-Gaussian case).

How to optimize localization and hybridization?

Optimal localization alone (without hybridization):

$$L'_{ij} = \frac{(N-1)^2}{N(N-3)} + \frac{N-1}{N(N-2)(N-3)} \frac{\mathbb{E}\left[\widetilde{B}_{ii}\widetilde{B}_{jj}\right]}{\mathbb{E}\left[\widetilde{B}_{ij}^2\right]} - \frac{N}{(N-2)(N-3)} \frac{\mathbb{E}\left[\widetilde{\Xi}_{ijij}\right]}{\mathbb{E}\left[\widetilde{B}_{ij}^2\right]}$$

where $\widetilde{\Xi}$ is the sample fourth-order centered moment.

How to optimize localization and hybridization?

Optimal localization alone (without hybridization):

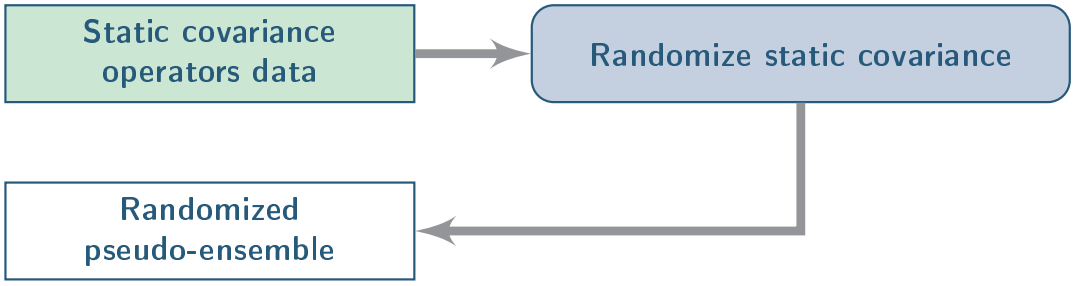
$$L'_{ij} = \frac{(N-1)^2}{N(N-3)} + \frac{N-1}{N(N-2)(N-3)} \frac{\mathbb{E}\left[\widetilde{B}_{ii}\widetilde{B}_{jj}\right]}{\mathbb{E}\left[\widetilde{B}_{ij}^2\right]} - \frac{N}{(N-2)(N-3)} \frac{\mathbb{E}\left[\widetilde{\Xi}_{ijij}\right]}{\mathbb{E}\left[\widetilde{B}_{ij}^2\right]}$$

where $\widetilde{\Xi}$ is the sample fourth-order centered moment.

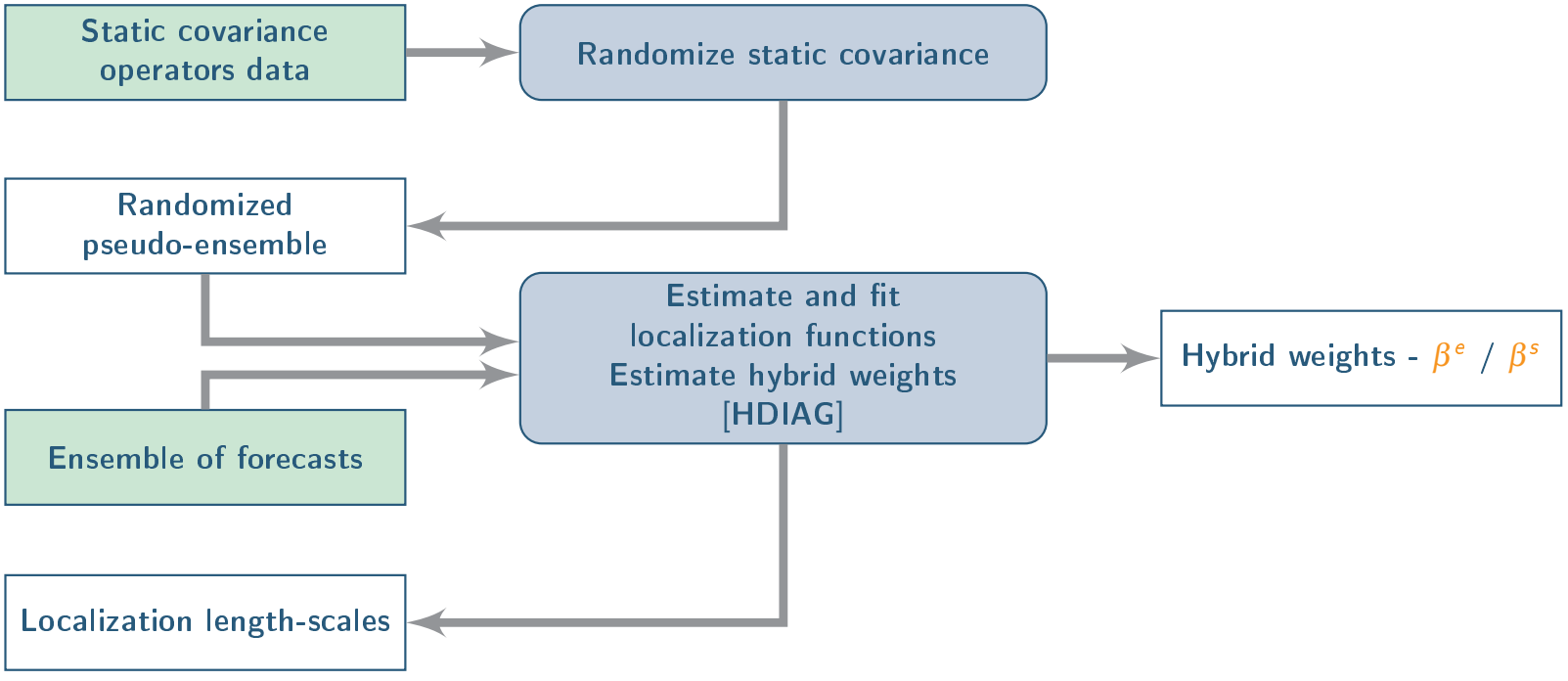
Hybrid coefficients and adated localization:

$$\beta^{s2} = \frac{\sum_{ij}\left(1-L'_{ij}\right)\mathbb{E}\left[\widetilde{B}_{ij}\right]B^s_{ij}}{\sum_{ij}\frac{\mathrm{Var}\left[\widetilde{B}_{ij}\right]}{\mathbb{E}\left[\widetilde{B}_{ij}^2\right]}B^{s2}_{ij}} \quad \text{and} \quad \beta^{e2}L_{ij} = L'_{ij} - \frac{\mathbb{E}\left[\widetilde{B}_{ij}\right]}{\mathbb{E}\left[\widetilde{B}_{ij}^2\right]}\beta^{s2}B^s_{ij}$$

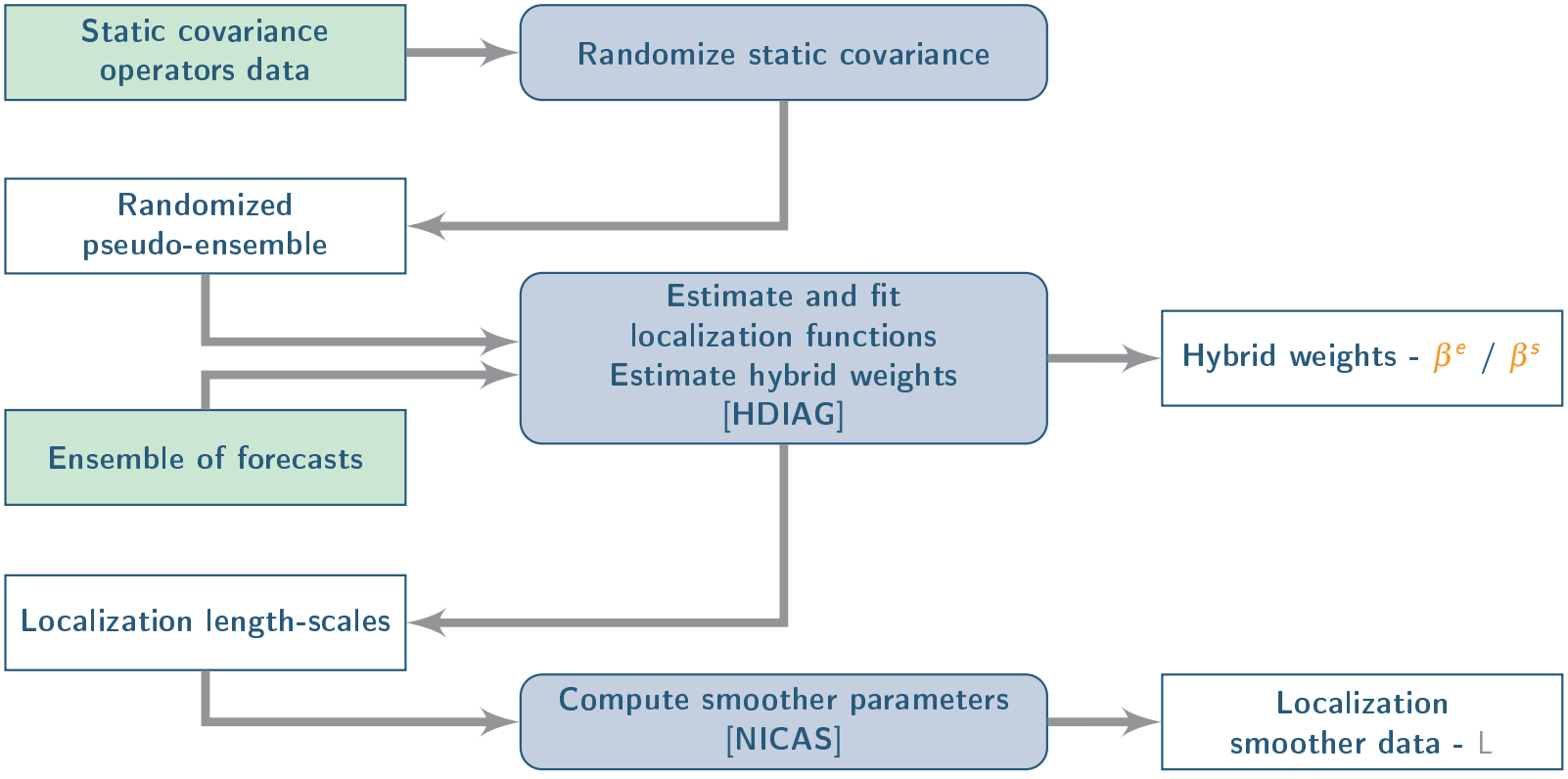
Localization and hybrid weights training



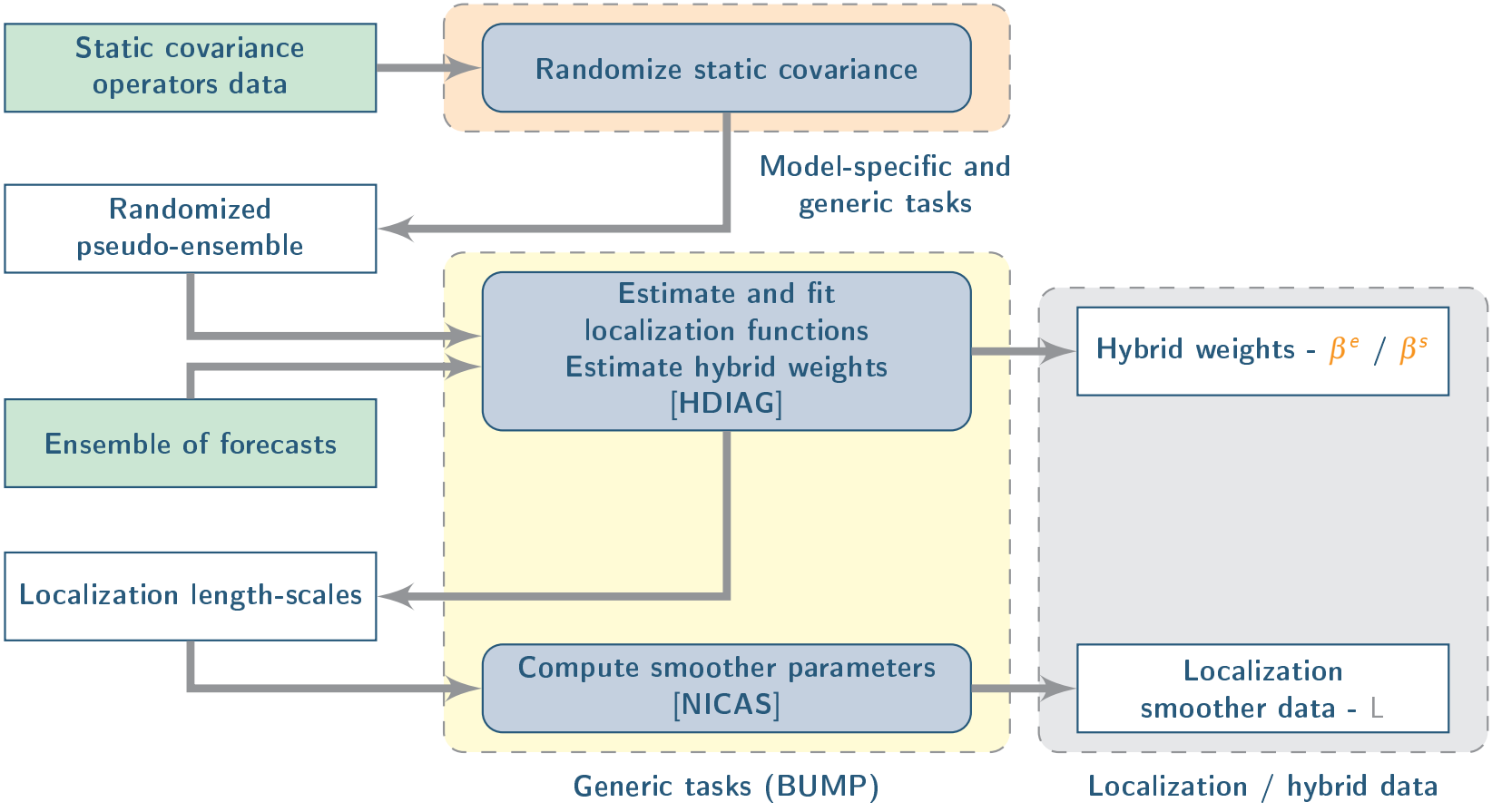
Localization and hybrid weights training



Localization and hybrid weights training



Localization and hybrid weights training



Local estimation of L with HDIAG

For each couple of points, BUMP uses the ensemble to estimate:

- the sample variances $\widetilde{B}_{ii}$ and $\widetilde{B}_{jj}$
- the sample covariance $\widetilde{B}_{ij}$
- the sample fourth-order centered moment $\widetilde{\Xi}_{ijij}$

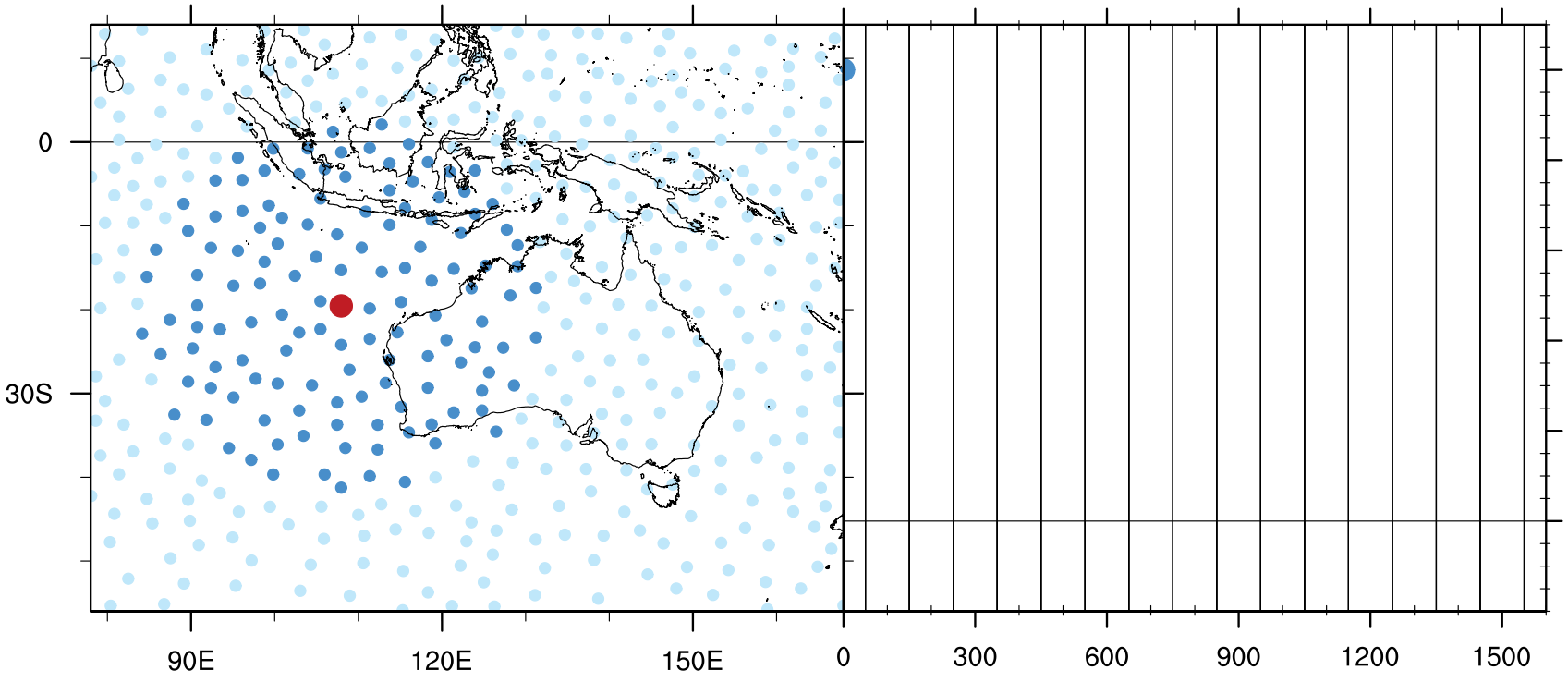
With the spatial and angular ergodicity assumption, these values are averaged locally for each distance class, to get estimations of the following expectations:

$$\mathbb{E}\left[\widetilde{B}_{ii}\widetilde{B}_{jj}\right] \quad , \quad \mathbb{E}\left[\widetilde{B}_{ij}^2\right] \quad \text{and} \quad \mathbb{E}\left[\widetilde{\Xi}_{ijij}\right]$$

These quantities are useful to compute the localization function and hybrid weights with the previous formulas.

Local estimation of L with HDIAG

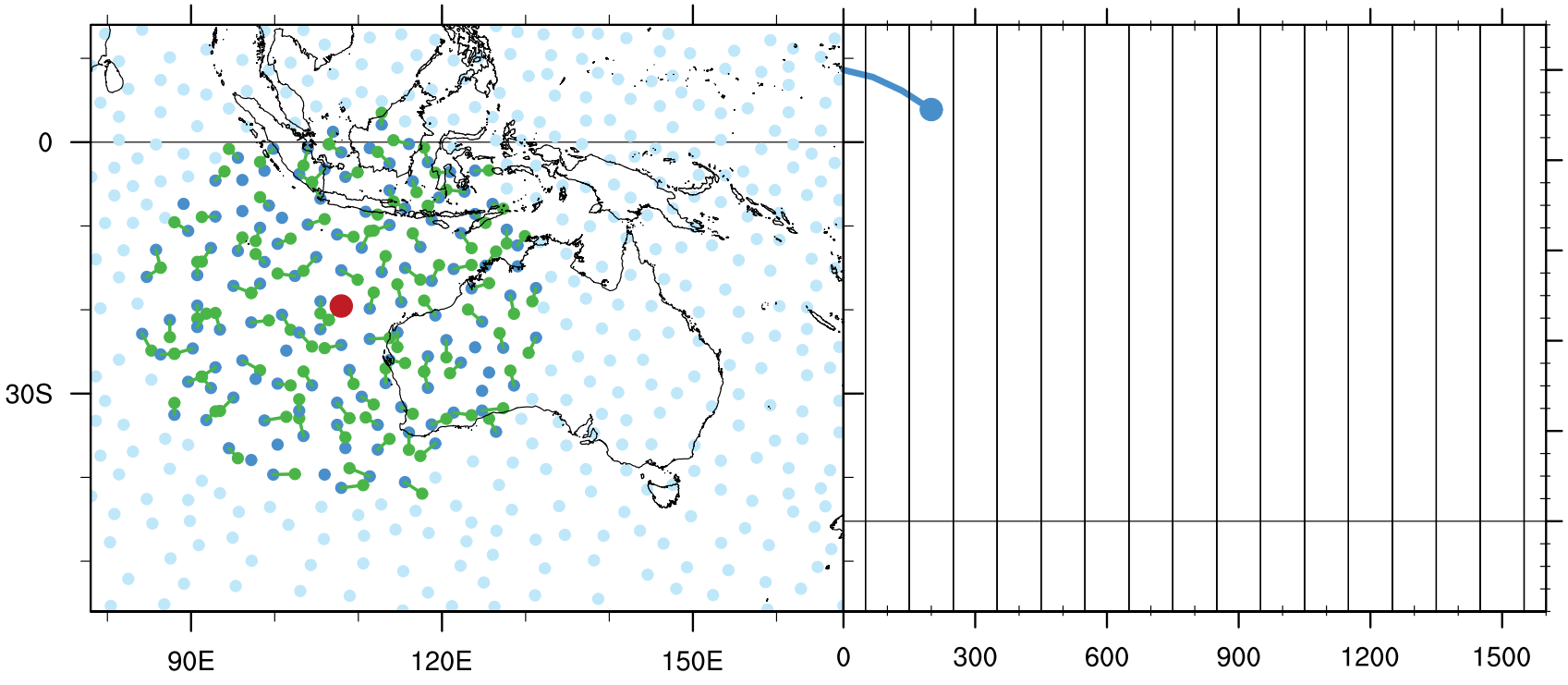
For a given local averaging center:



Averaged couples for a given local averaging center (left)
Diagnosed localization as a function of distance in km (right)

Local estimation of L with HDIAG

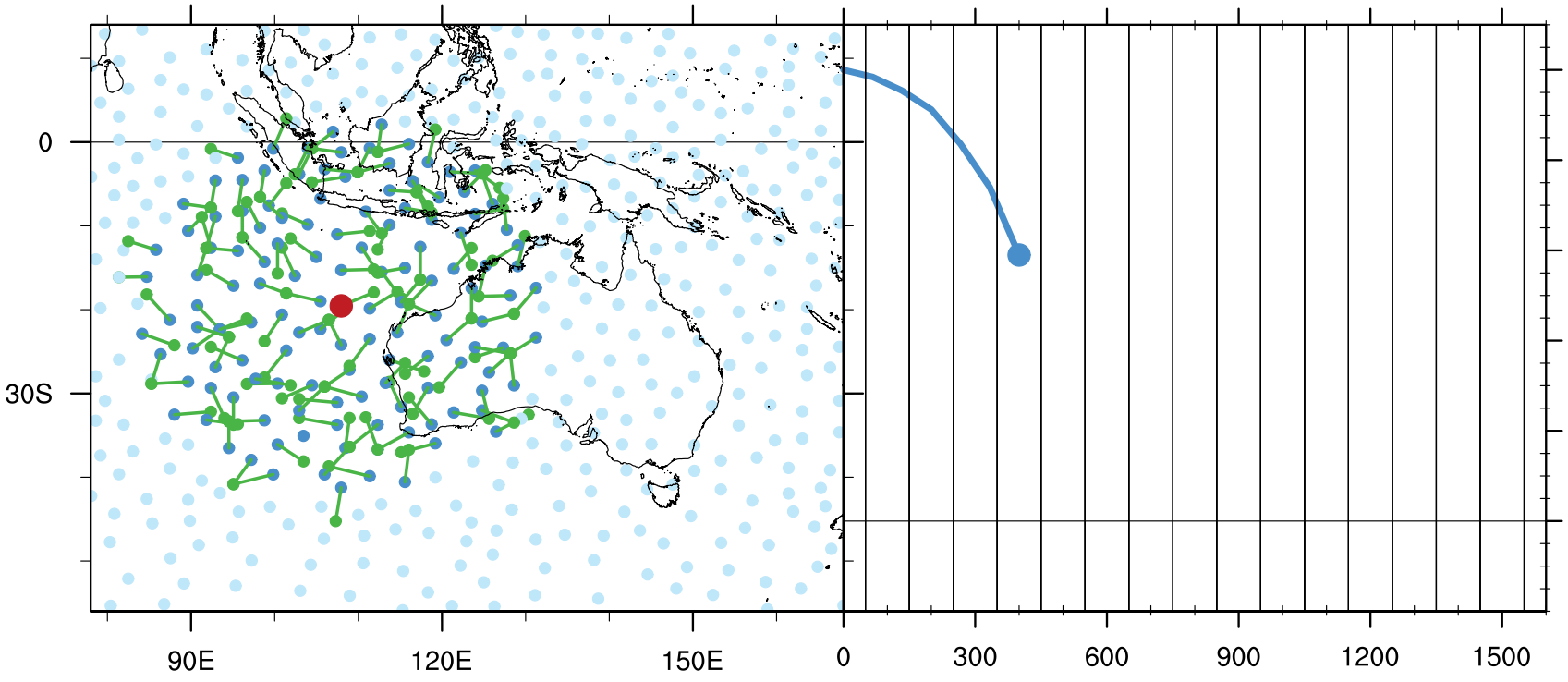
For a given local averaging center:



Averaged couples for a given local averaging center (left)
Diagnosed localization as a function of distance in km (right)

Local estimation of L with HDIAG

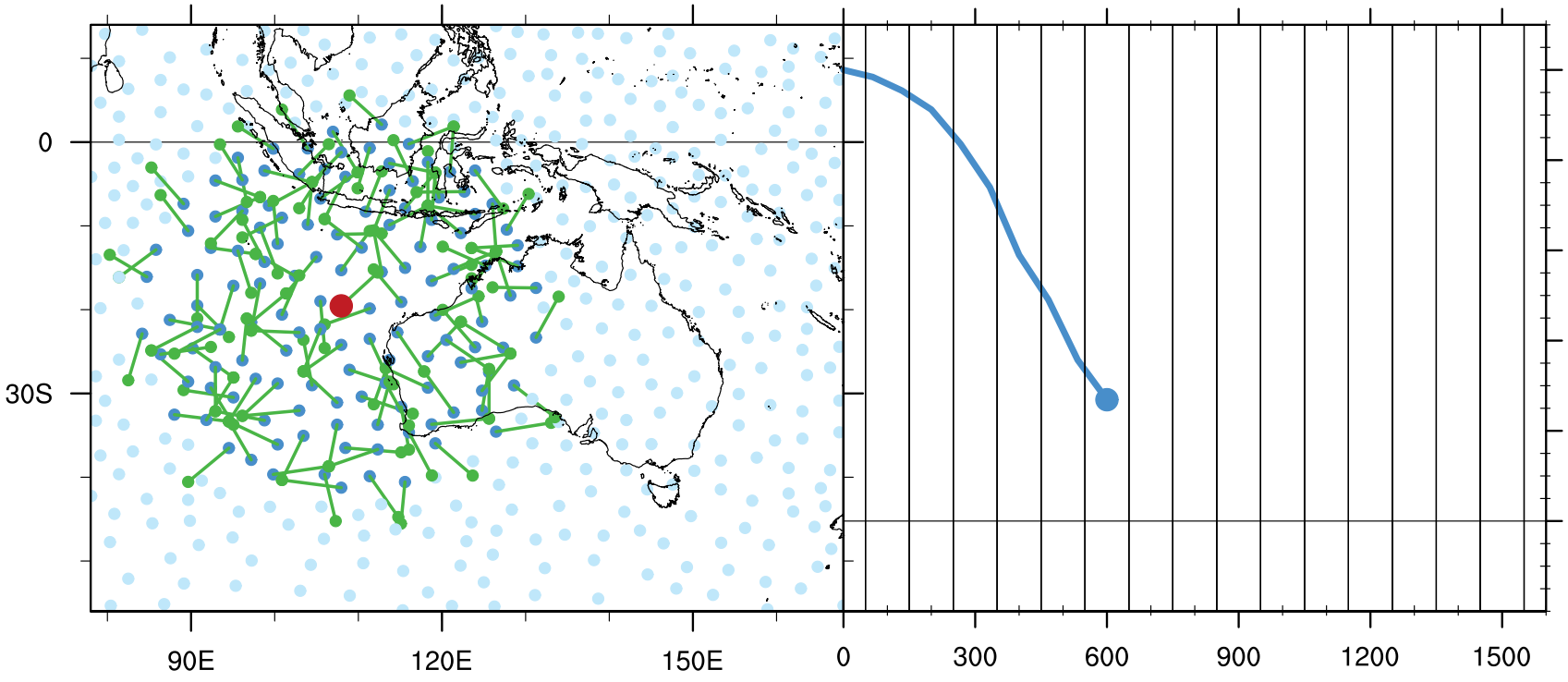
For a given local averaging center:



Averaged couples for a given local averaging center (left)
Diagnosed localization as a function of distance in km (right)

Local estimation of L with HDIAG

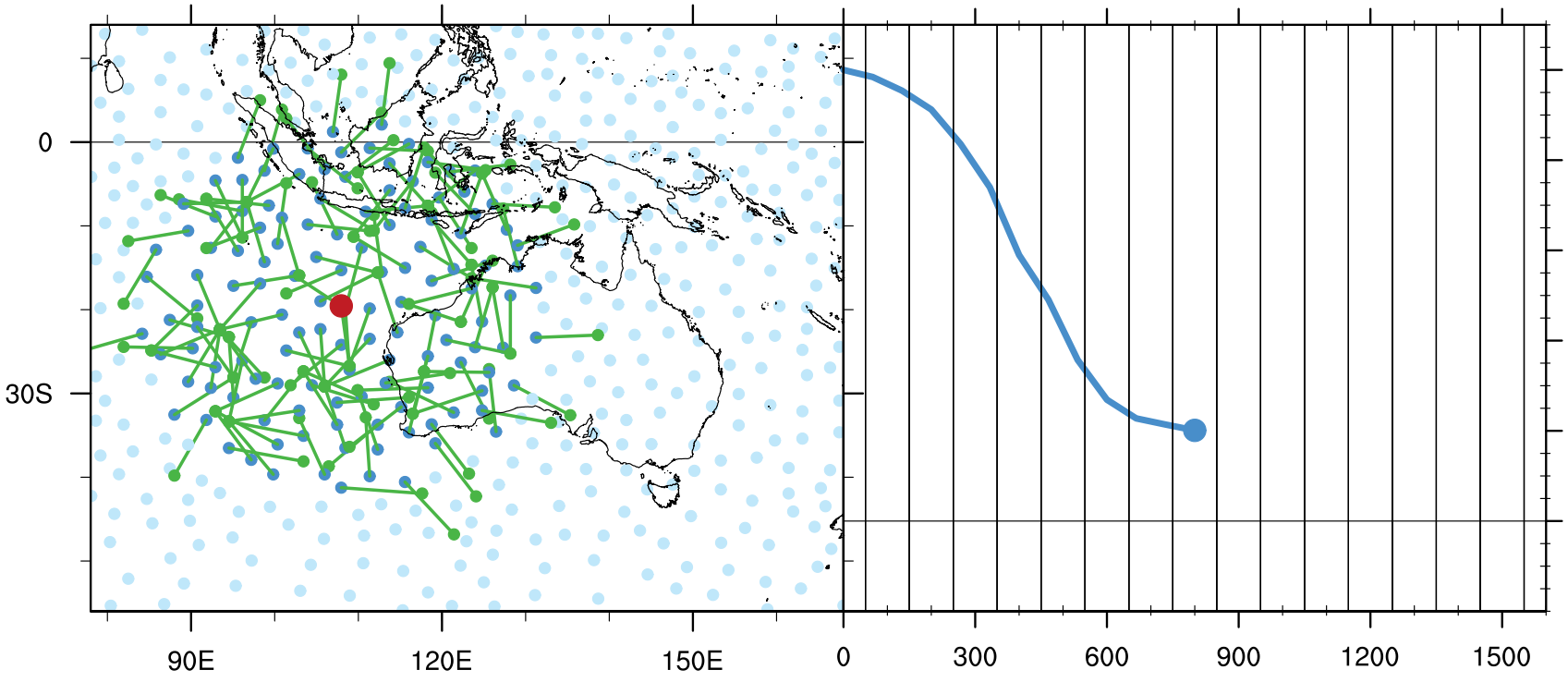
For a given local averaging center:



Averaged couples for a given local averaging center (left)
Diagnosed localization as a function of distance in km (right)

Local estimation of L with HDIAG

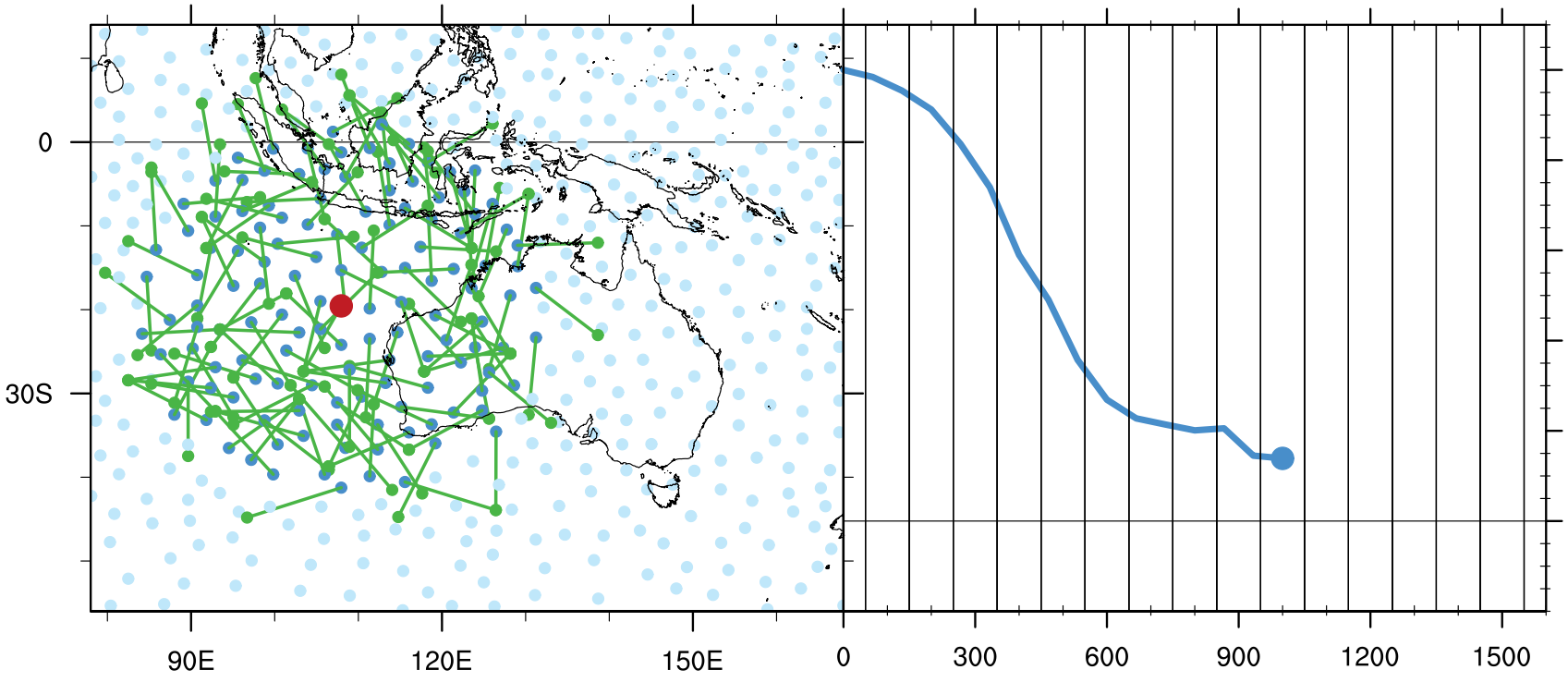
For a given local averaging center:



Averaged couples for a given local averaging center (left)
Diagnosed localization as a function of distance in km (right)

Local estimation of L with HDIAG

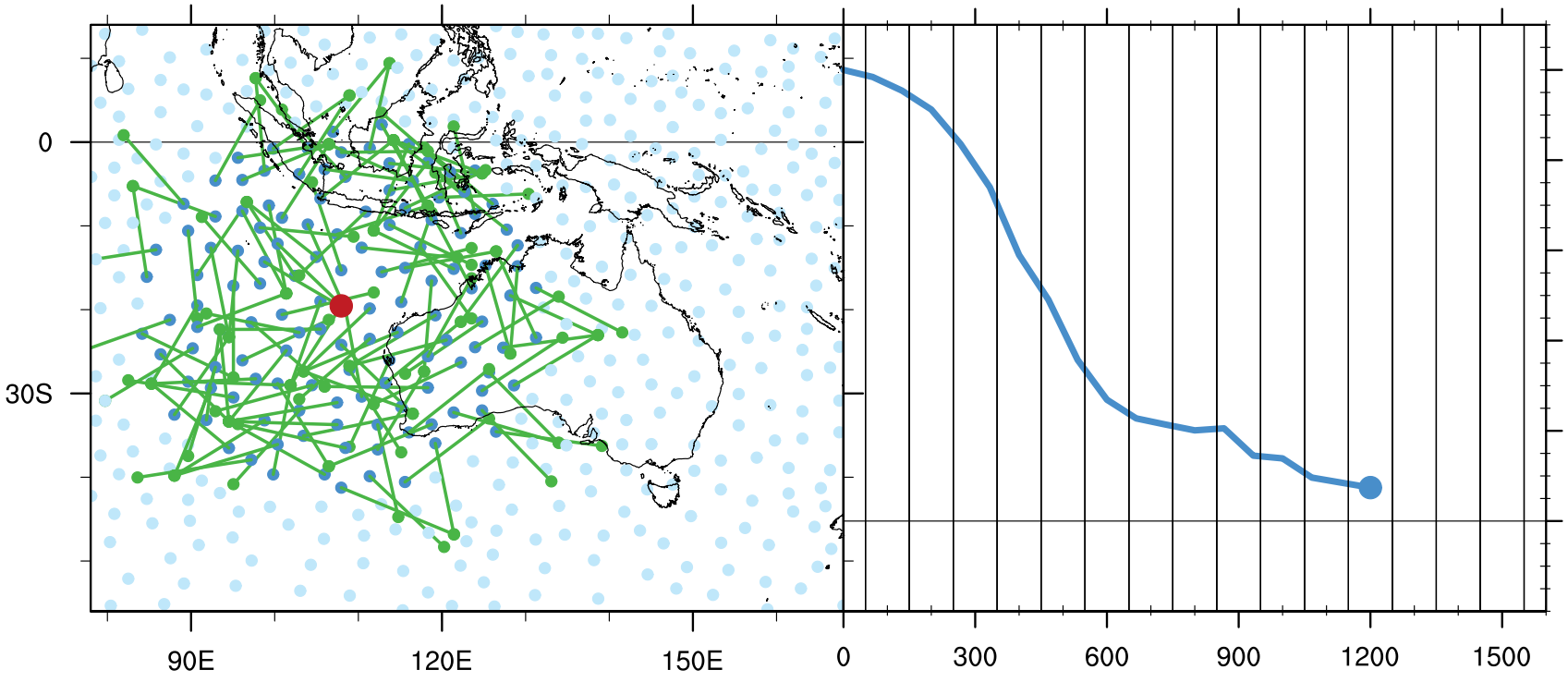
For a given local averaging center:



Averaged couples for a given local averaging center (left)
Diagnosed localization as a function of distance in km (right)

Local estimation of L with HDIAG

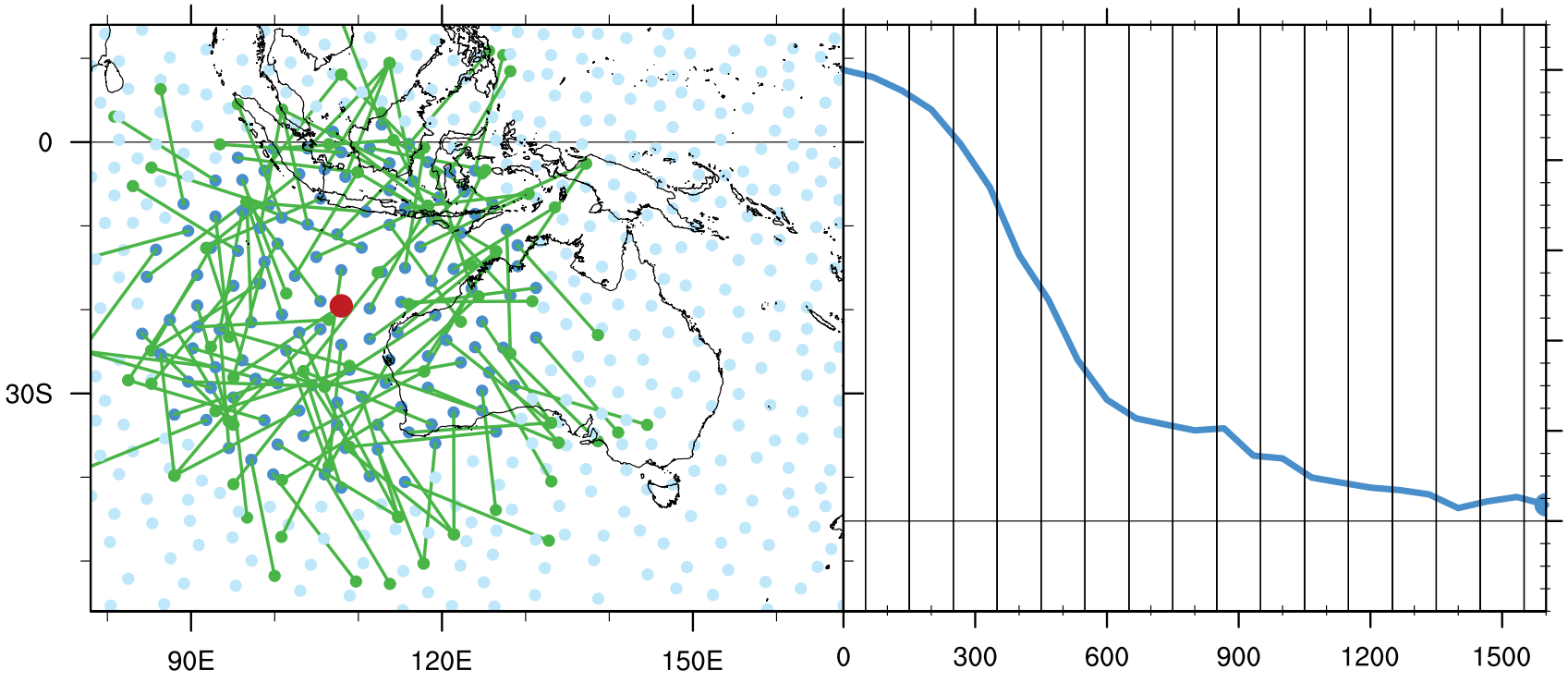
For a given local averaging center:



Averaged couples for a given local averaging center (left)
Diagnosed localization as a function of distance in km (right)

Local estimation of L with HDIAG

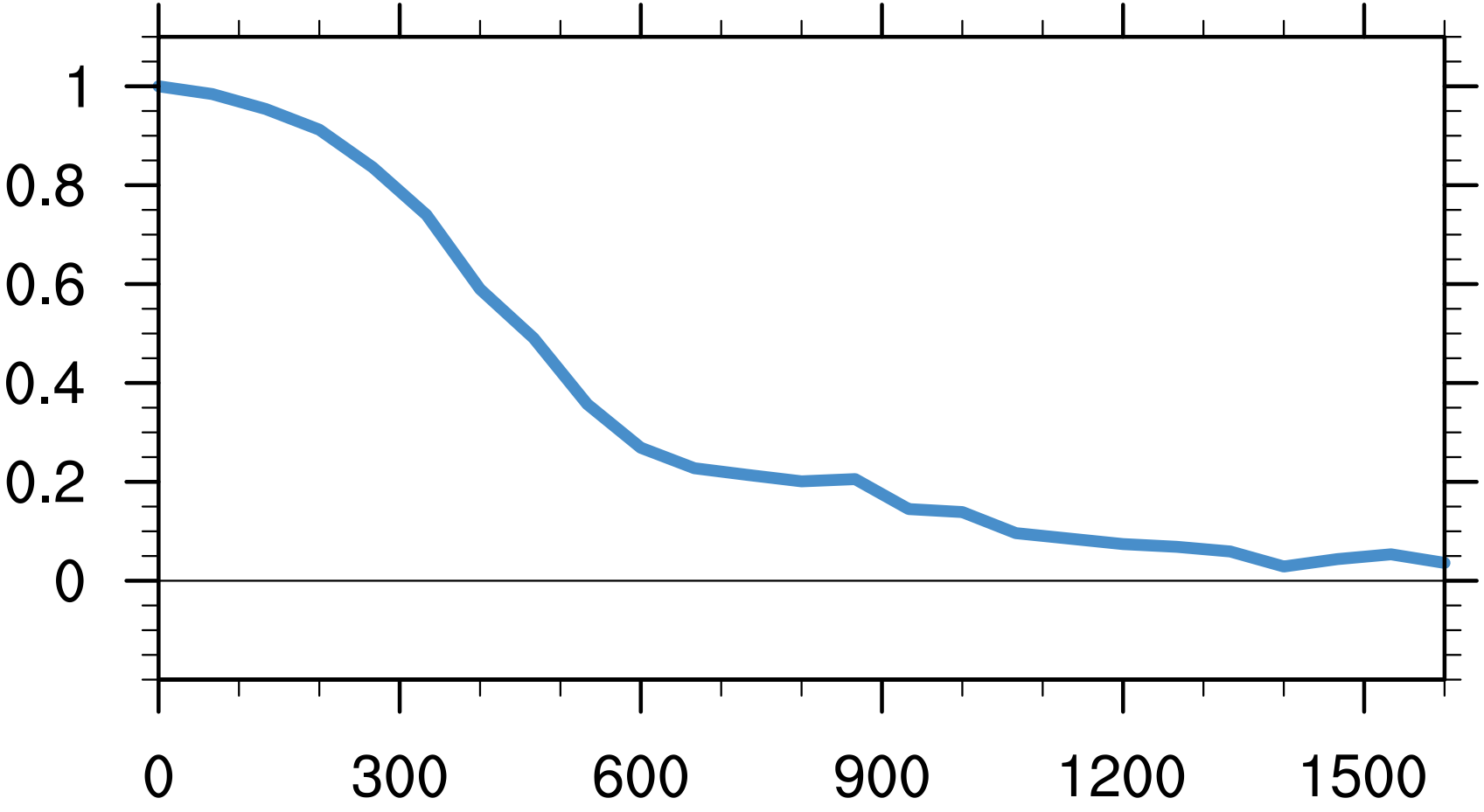
For a given local averaging center:



Averaged couples for a given local averaging center (left)
Diagnosed localization as a function of distance in km (right)

Local estimation of L with HDIAG

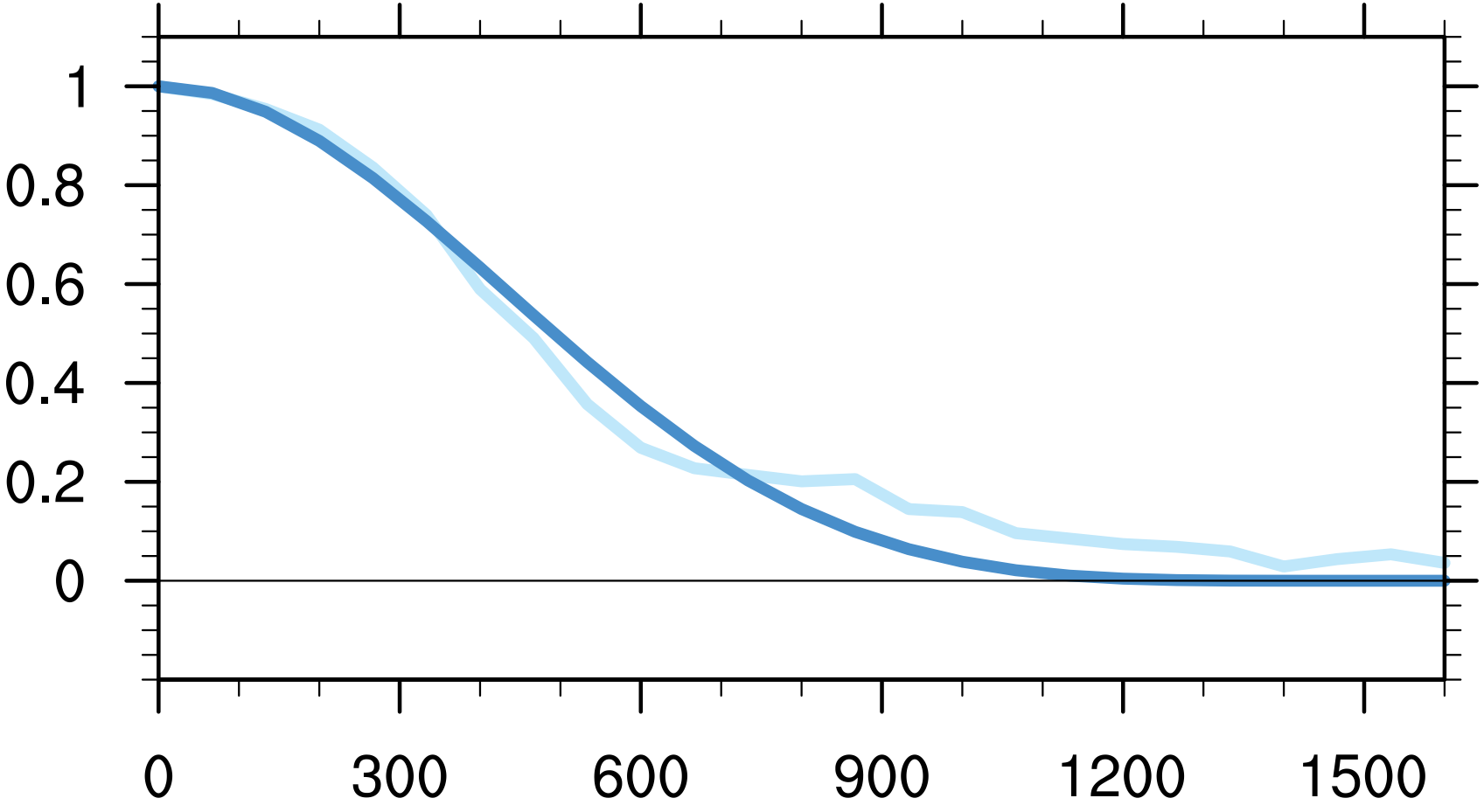
For a given local averaging center:



Raw localization as a function of distance (km)

Local estimation of L with HDIAG

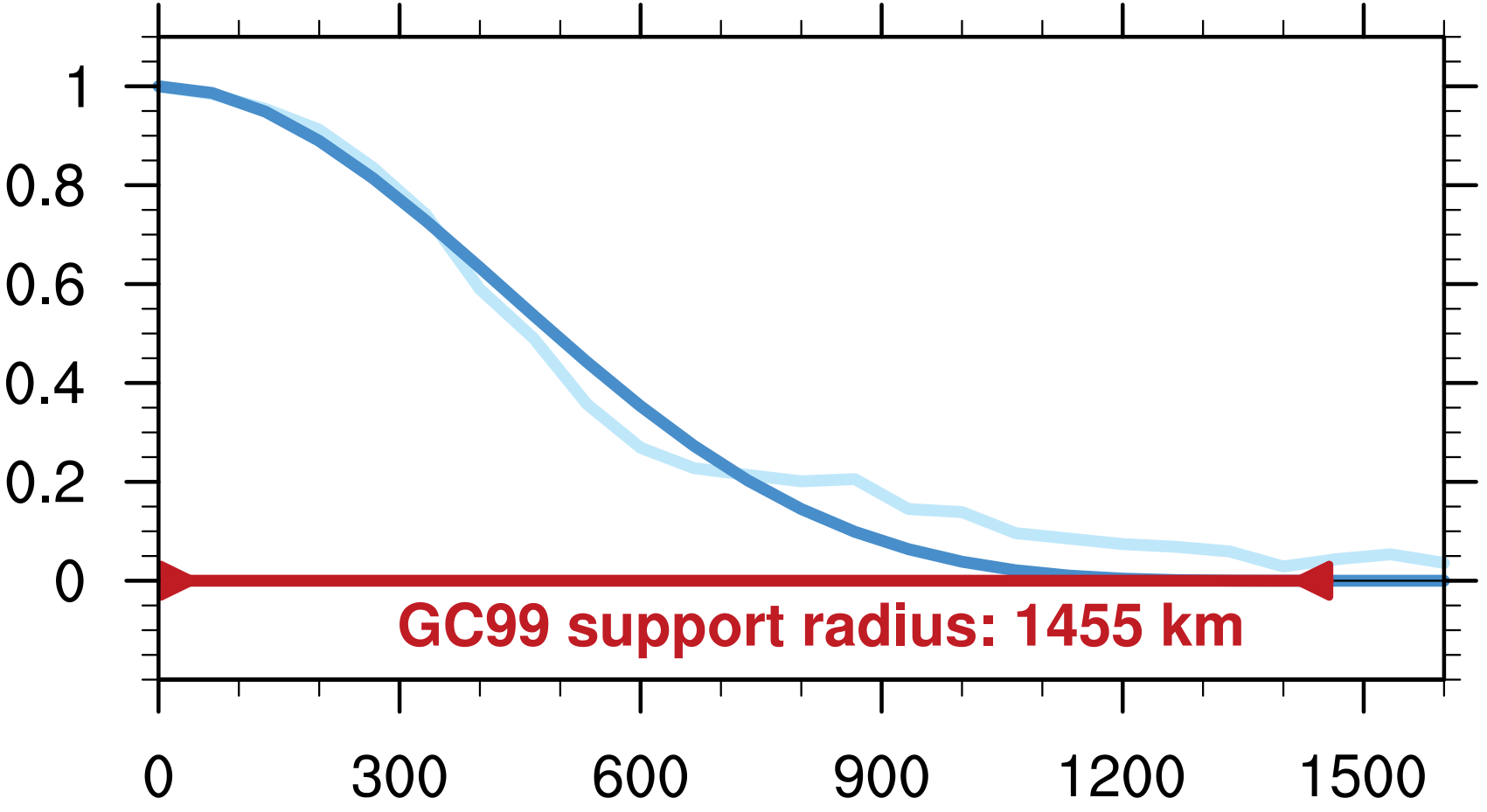
For a given local averaging center:



Fit with a Gaspari-Cohn (1999) function

Local estimation of L with HDIAG

For a given local averaging center:



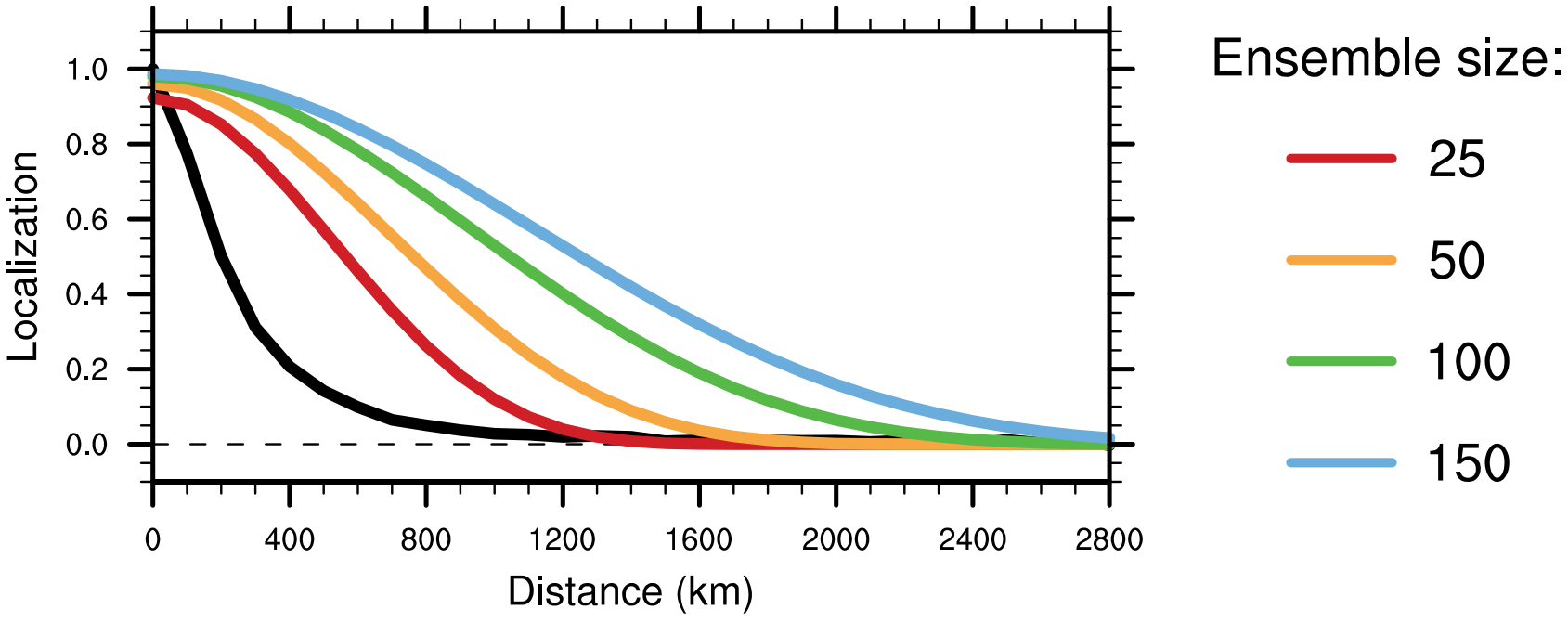
Localization length-scale (support radius actually, km)

Local estimation of L with HDIAG

HDIAG summary (localization):

1. Define a homogeneous subgrid (origin points).
2. For each origin point, find distant points for a series of distance classes.
3. For each local averaging center:
   - Average diagnostic values over a given radius.
   - Compute the localization and hybrid weights using the Ménétrier et al. (2015) formula.
   - Fit the raw localization function with a parametrized function (e.g. Gaspari-Cohn, 1999) to get the localization support radius.
4. Optionally filter the localization support radius and the hybrid weights on the subgrid.
5. Interpolate the localization support radius and hybrid weights over the model grid.

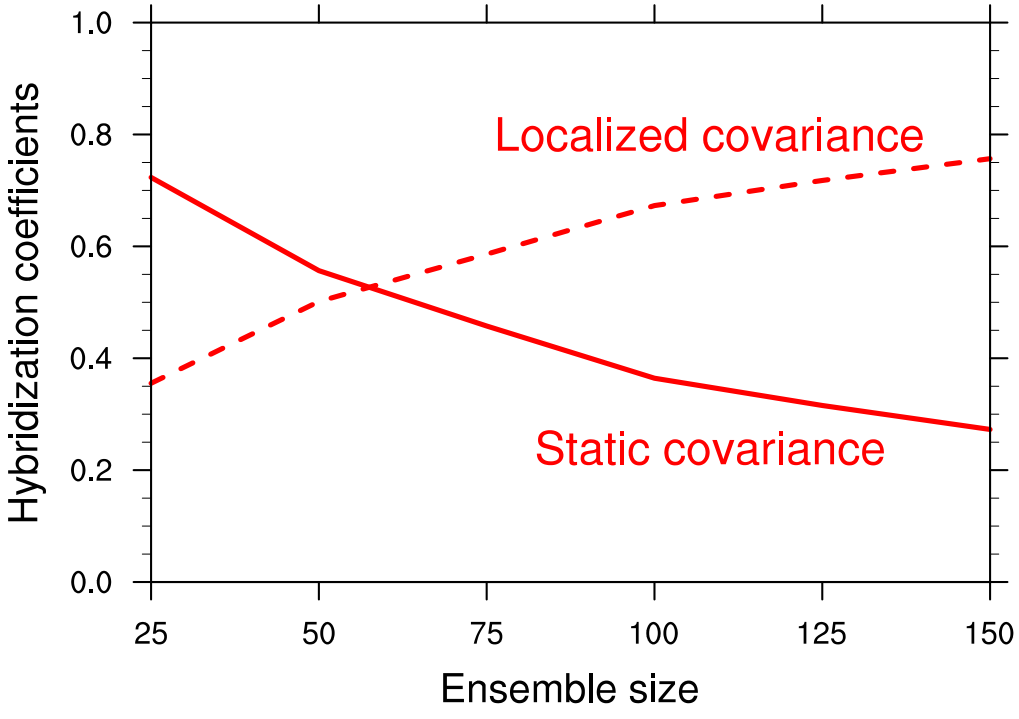
Ensemble size sensitivity



Correlation (black) et localization (colors) for various ensemble sizes

Localization length-scale increases as the ensemble size increases (less sampling noise to remove)

Ensemble size sensitivity



Ensemble and static weights as a function of the ensemble size

Less weight on the static covariance as the ensemble size increases (less deficiencies to correct).

Summary

- The sample covariance is affected by sampling noise.
- This sampling noise decreases if the ensemble size increases.
- Using a huge ensemble is too costly for operational applications.
- Localization and hybridization can be used to filter out the sampling noise.
- The asymptotic sample covariance is the filtering target.
- We combine the linear filtering theory and the centered moments sampling theory.
- This leads to practicable formulas for optimal localization and hybrid weights.
- We compute robust estimates using ergodicity assumptions.
- Thus, we get local estimates of the optimal localization and hybrid weights.

Outline

Explicit convolution

Main goal: designing a generic method to apply a normalized convolution operator **on any grid type**.

Standard methods:

- Spectral/wavelet transforms $\rightarrow$ regular grid required
- Recursive filters $\rightarrow$ regular grid required + normalization issue
- Explicit/implicit diffusion $\rightarrow$ potentially high cost + normalization issue

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 C on a subgrid (subset of n^s points of the model grid):

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

where

- S is an interpolation from the subgrid to the model grid
- 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 C^s is normalized, $\mathrm{S}\mathrm{C}^s\mathrm{S}^{\mathrm{T}}$ is not.

Explicit convolution

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

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

where

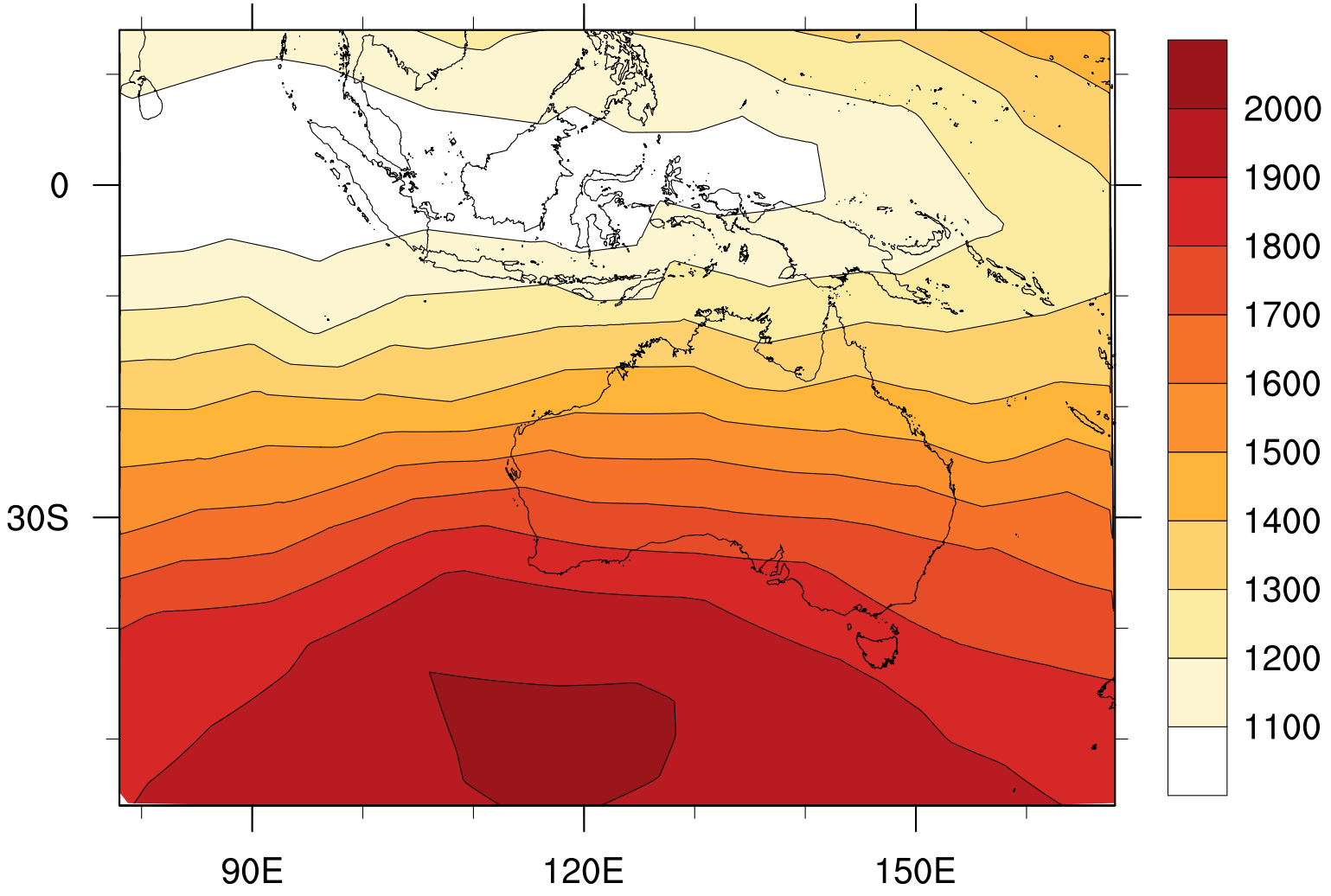
- N is a diagonal normalization matrix.
- The subgrid is locally adapted to the convolution length-scale.

To illustrate how NICAS works:

- Example of adaptive subgrid.
- Steps of a Dirac test: apply $\widetilde{\mathrm{C}}$ to a vector δ^k where

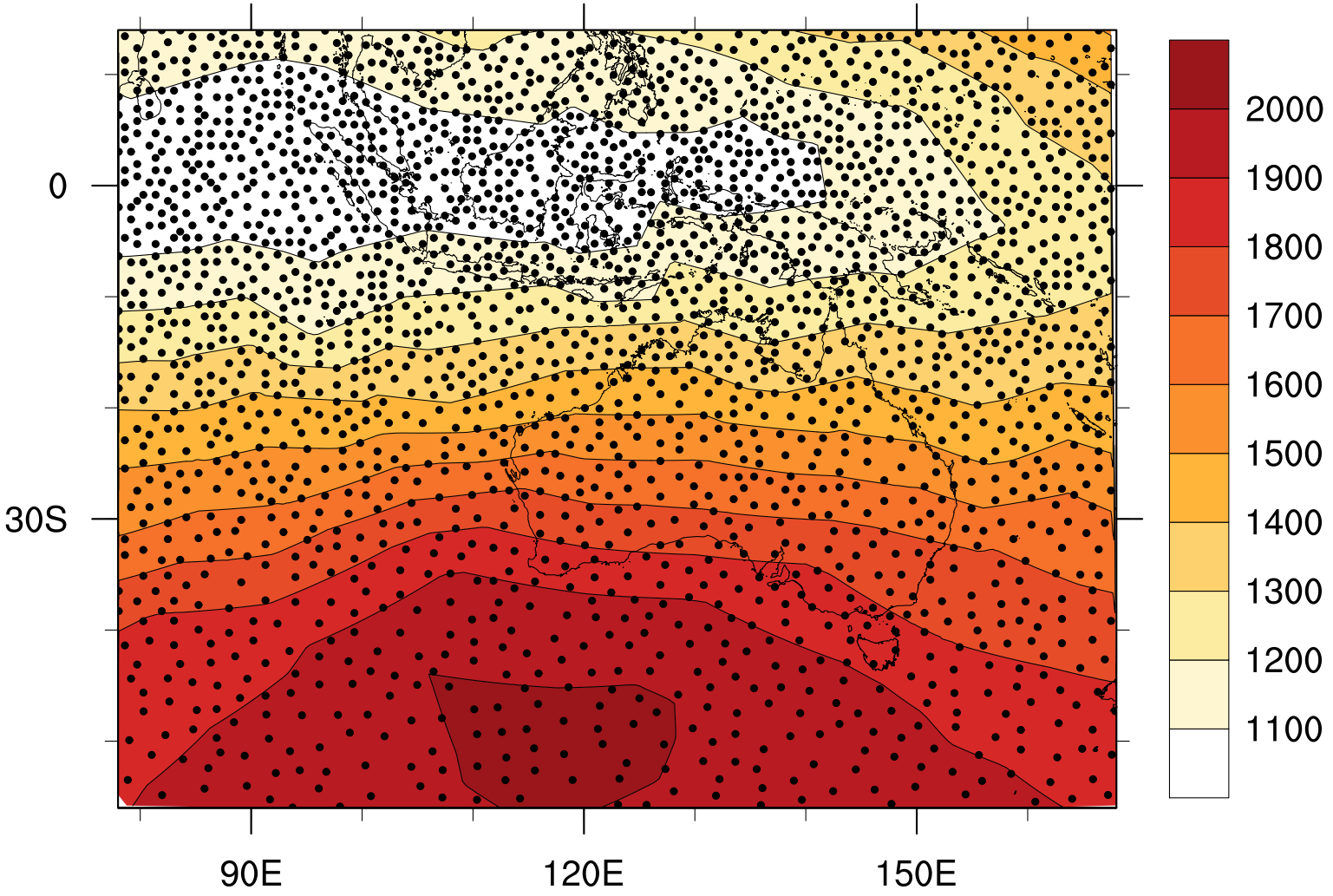
$$\delta_i^k = \begin{cases} 1 & \text{if} \quad i = k \\ 0 & \text{if} \quad i \neq k \end{cases}$$

Adaptive subgrid



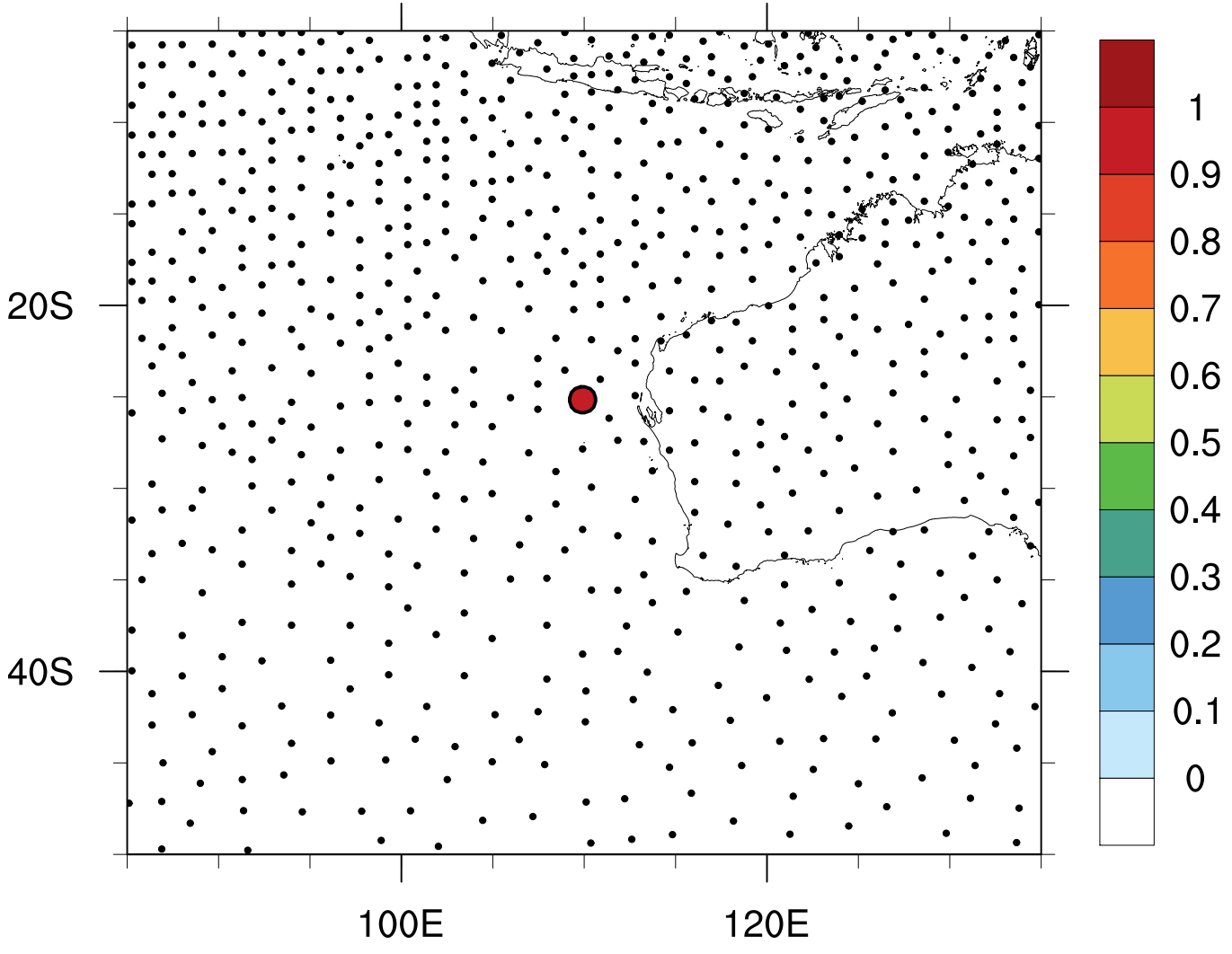
Localization support radius (km) interpolated on the model grid

Adaptive subgrid



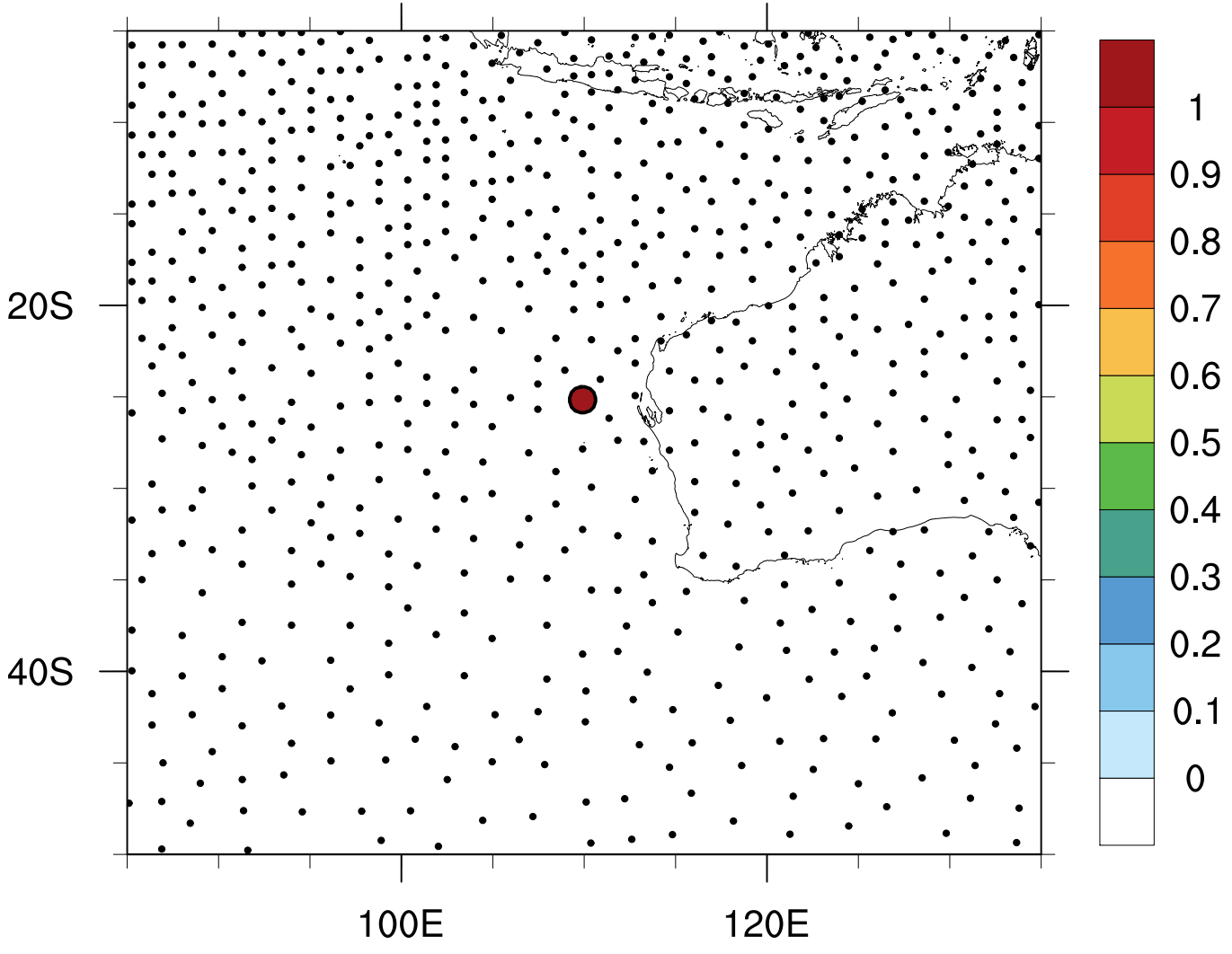
Adaptive support radius-based subgrid

NICAS steps



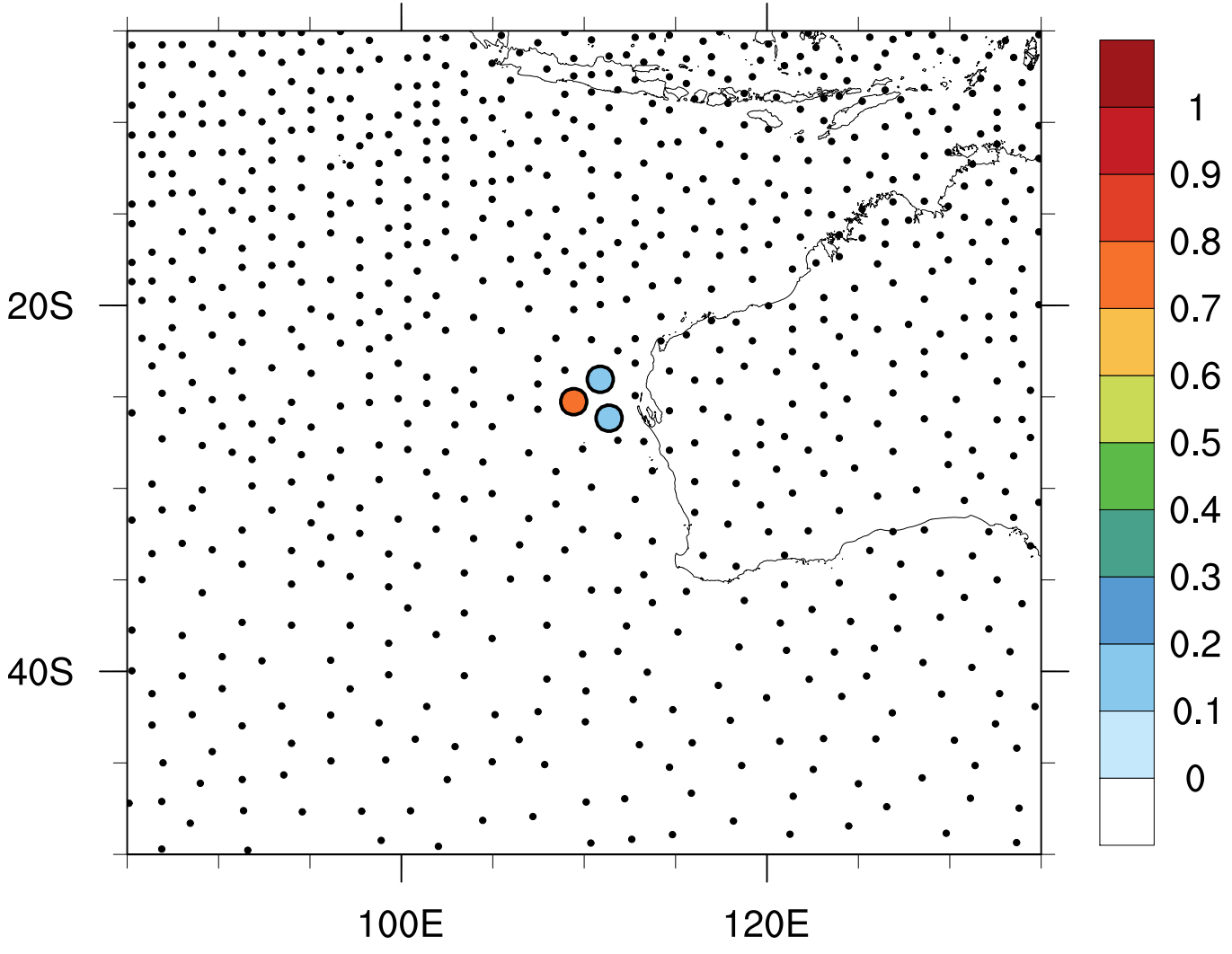
Initial vector: δ^k

NICAS steps



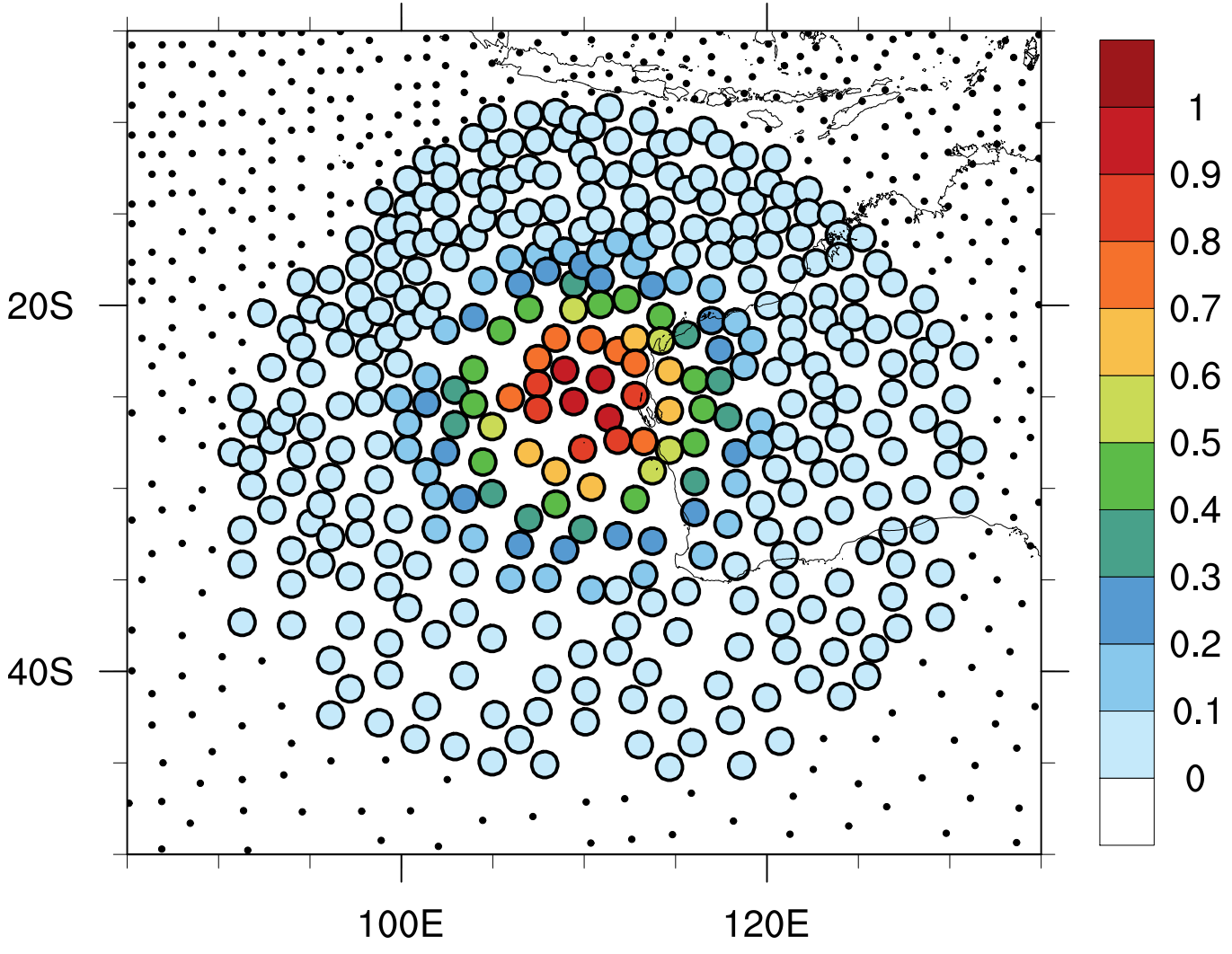
Adjoint normalization: $\mathrm{N}^{\mathrm{T}}\boldsymbol{\delta}^{k}$

NICAS steps



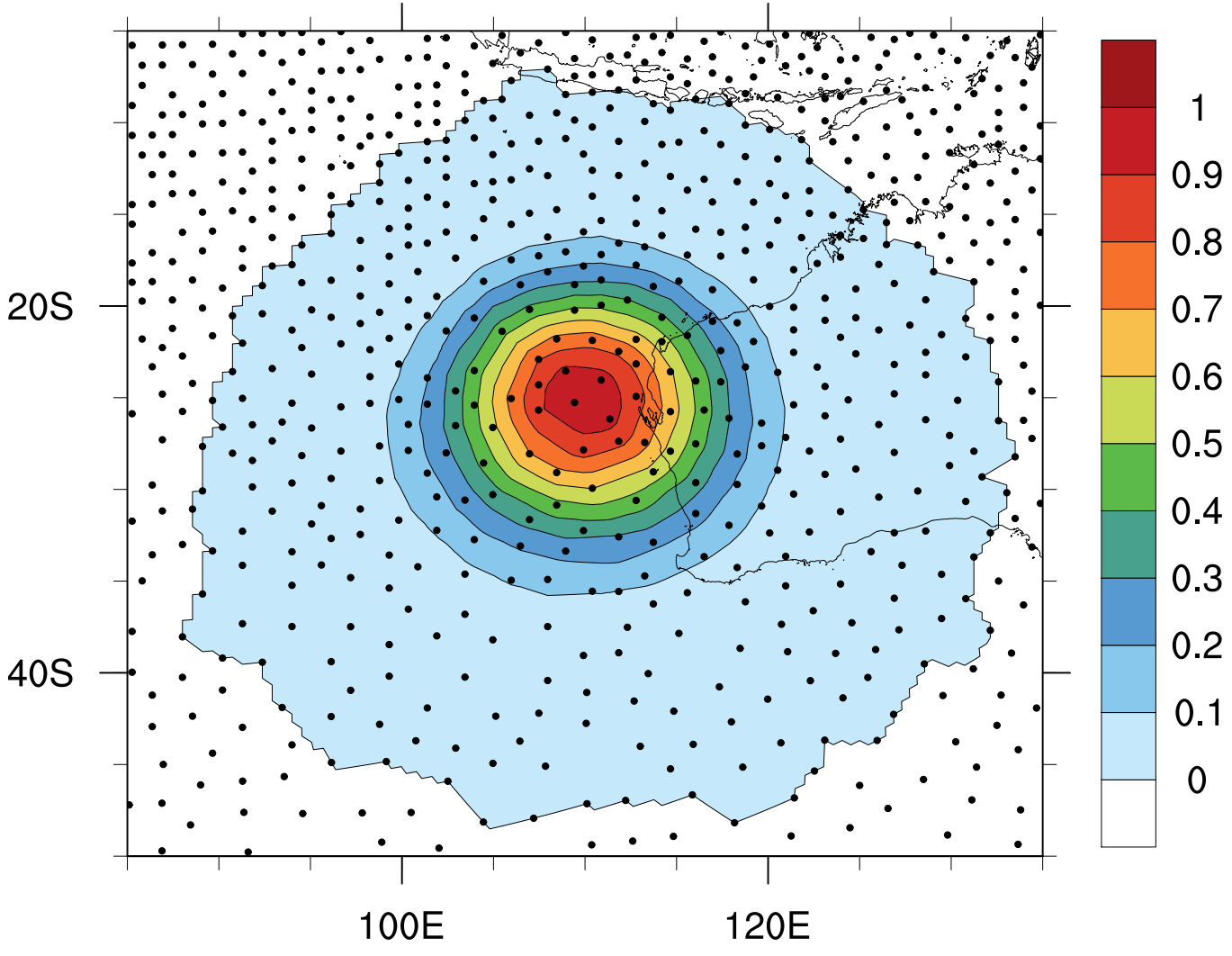
Adjoint interpolation: $S^T N^T \delta^k$

NICAS steps



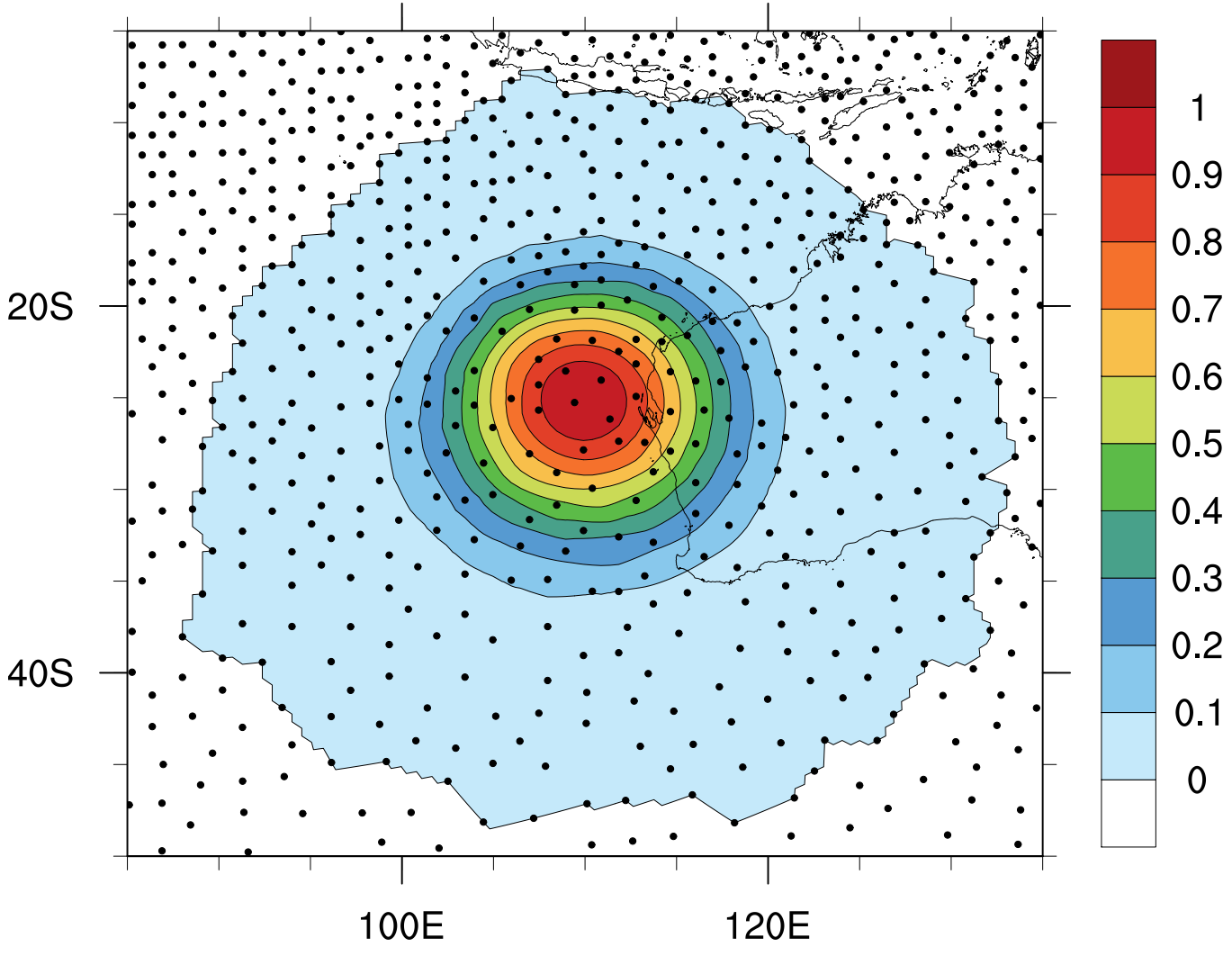
Convolution: $C^s S^T N^T \delta^k$

NICAS steps



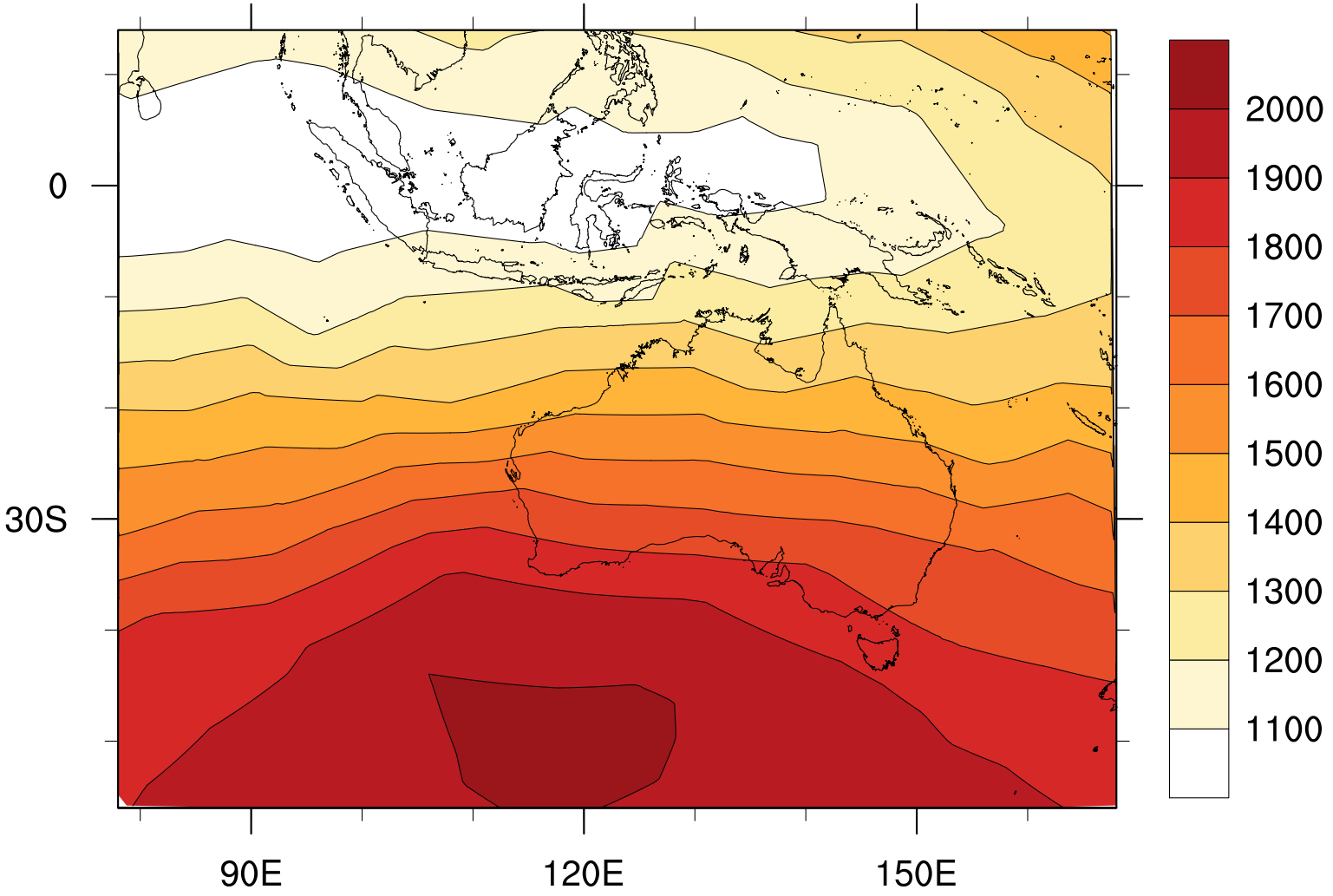
Interpolation: $\mathbf{S}\mathbf{C}^{s}\mathbf{S}^{\mathrm{T}}\mathbf{N}^{\mathrm{T}}\boldsymbol{\delta}^{k}$

NICAS steps



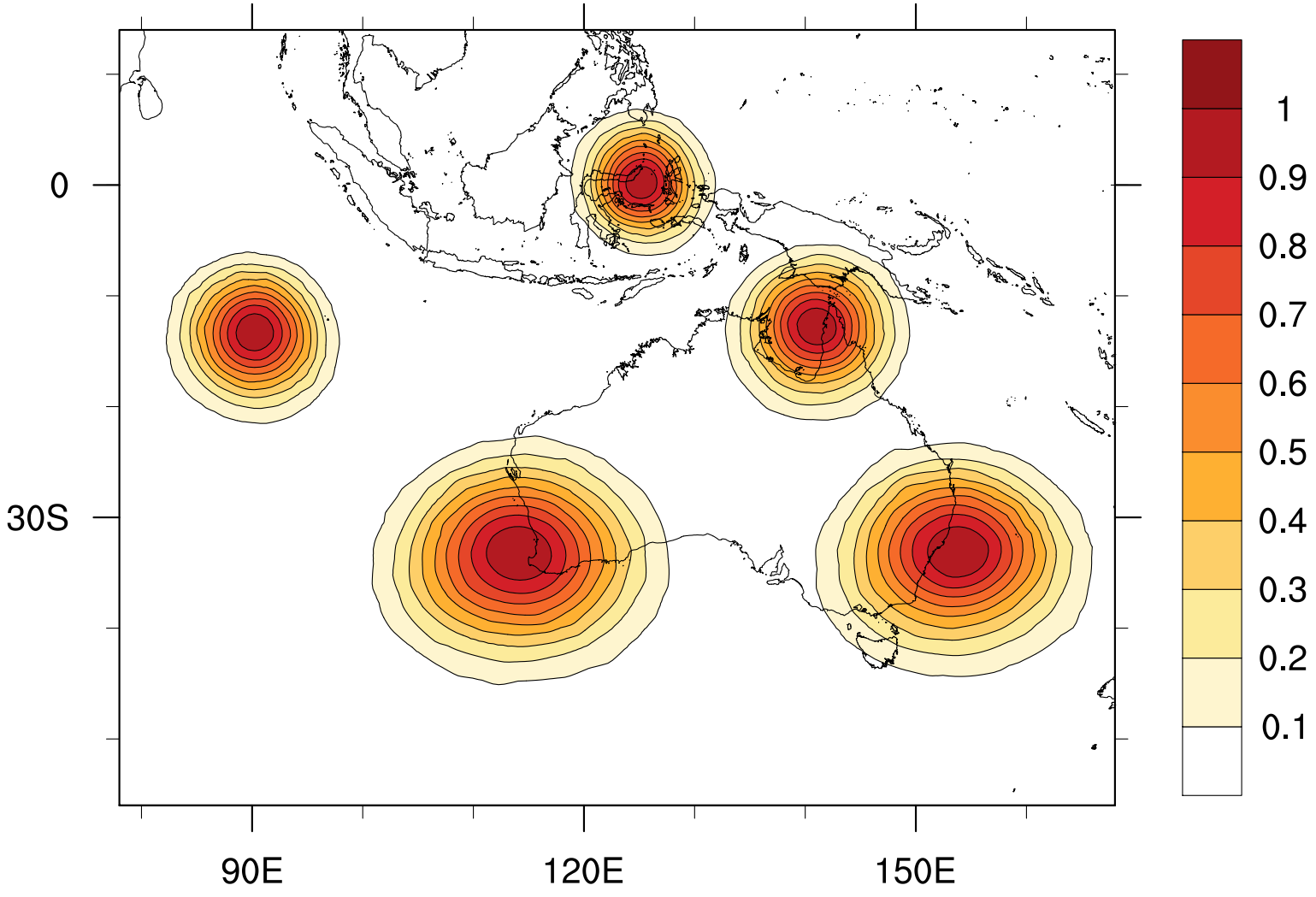
Normalization: $\mathrm{NSC}^{s}\mathrm{S}^{\mathrm{T}}\mathrm{N}^{\mathrm{T}}\delta^{k}$

Result of the Dirac test



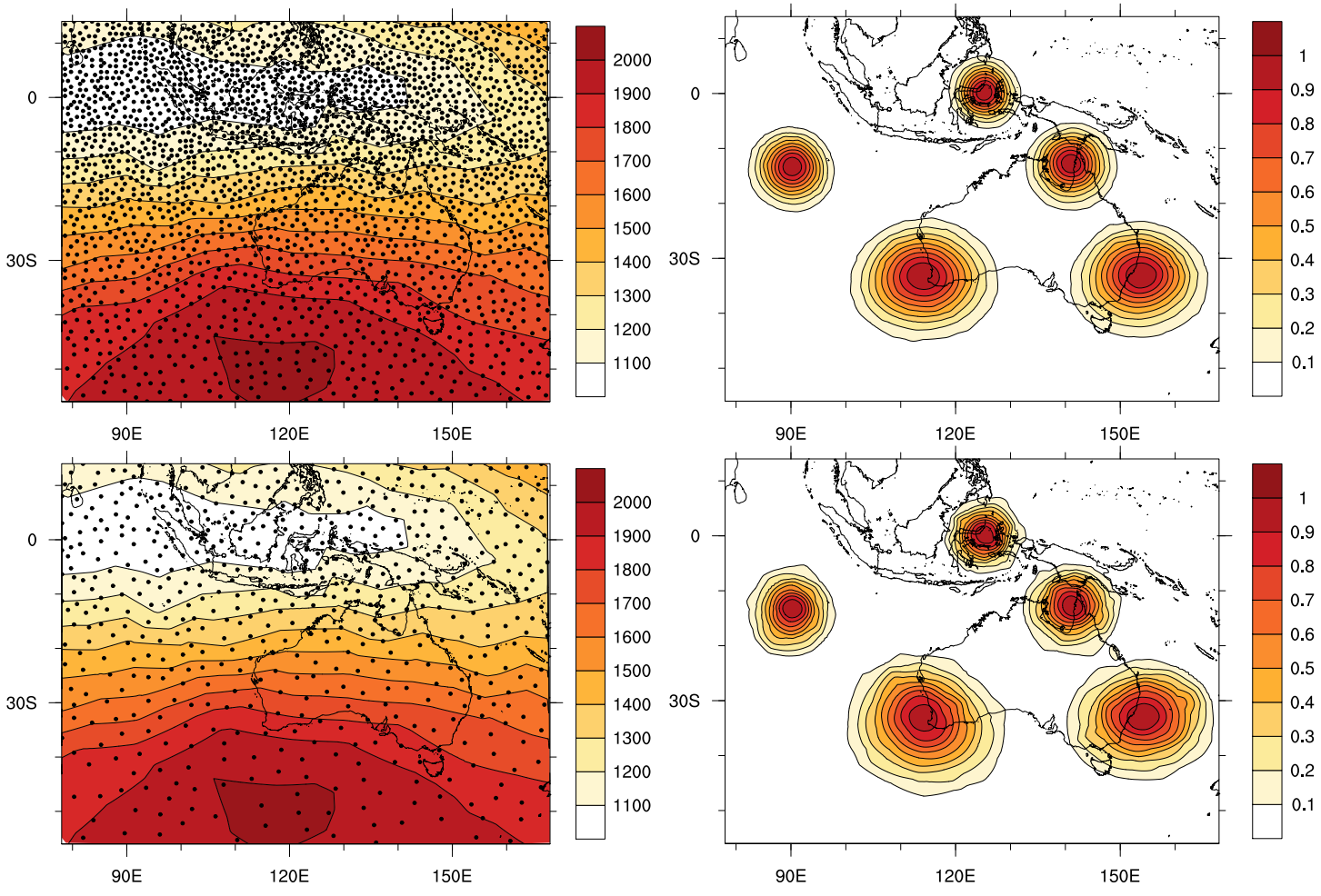
Localization support radius (km)

Result of the Dirac test



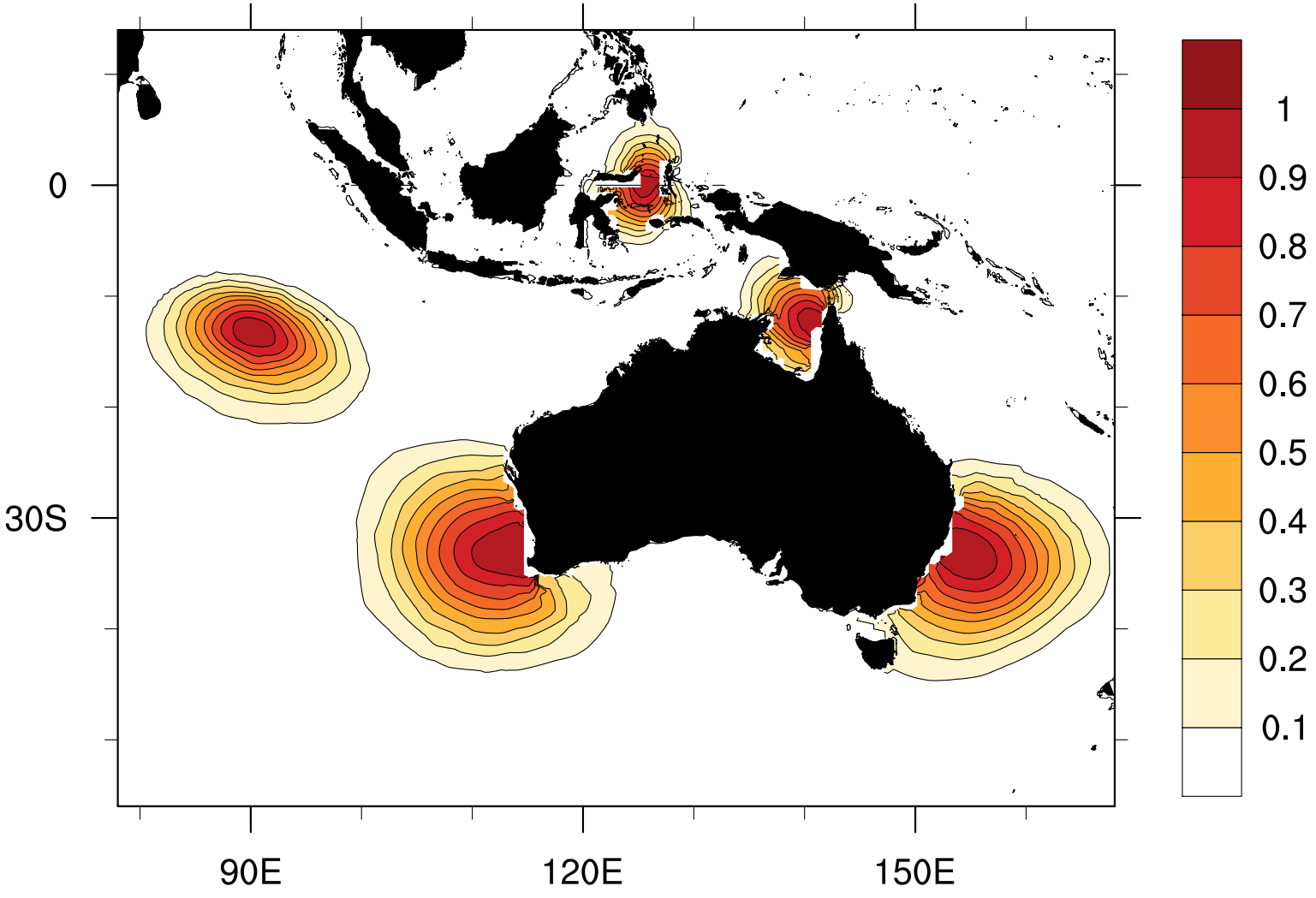
Application of NICAS on Dirac functions

Impact of the subgrid resolution



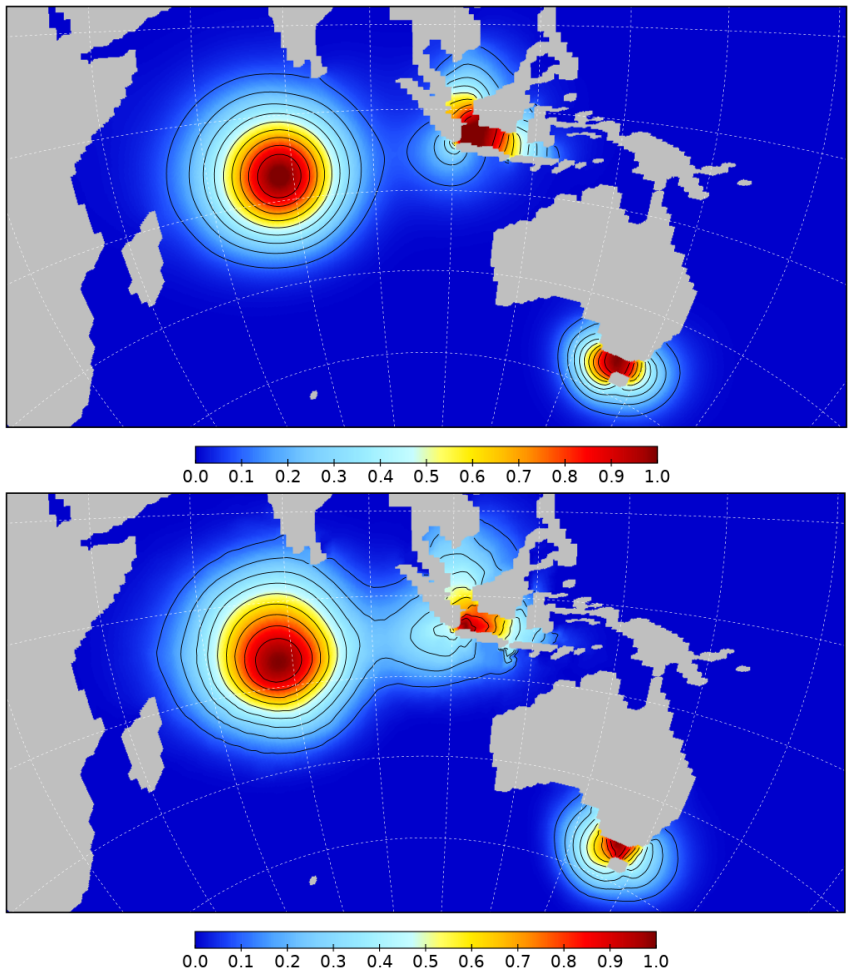
A scalar parameter controls the subgrid resolution.
Simple trade-off between cost and accuracy.

Complex boundaries



Application of masked anisotropic NICAS on Dirac functions

Complex boundaries



On the ORCA1 grid of NEMOVAR: implicit diffusion (top) and NICAS (bottom)

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 variational methods.

- Using the formulation:

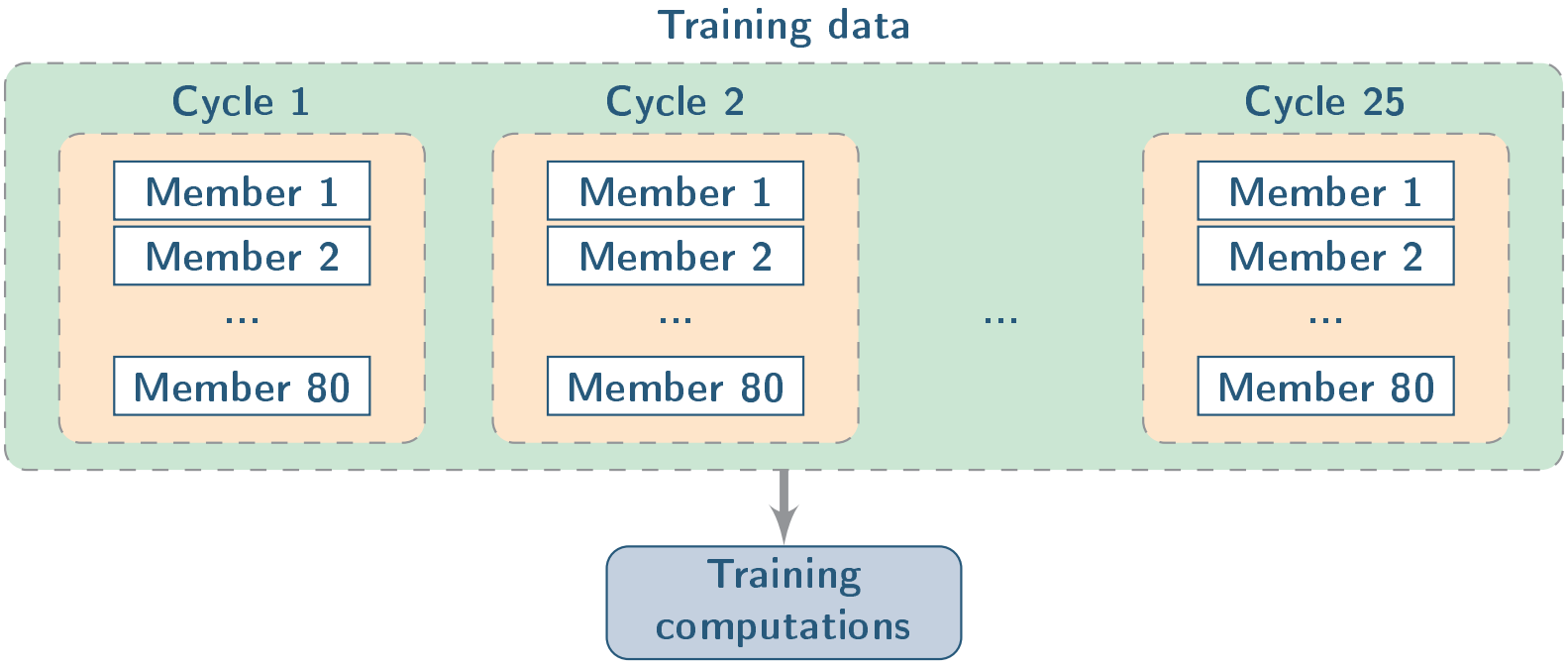
$$\widetilde{\mathbf{C}} = \mathbf{N}\mathbf{S}\mathbf{U}^{s}\mathbf{U}^{s\mathrm{T}}\mathbf{S}^{\mathrm{T}}\mathbf{N}^{\mathrm{T}}$$

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

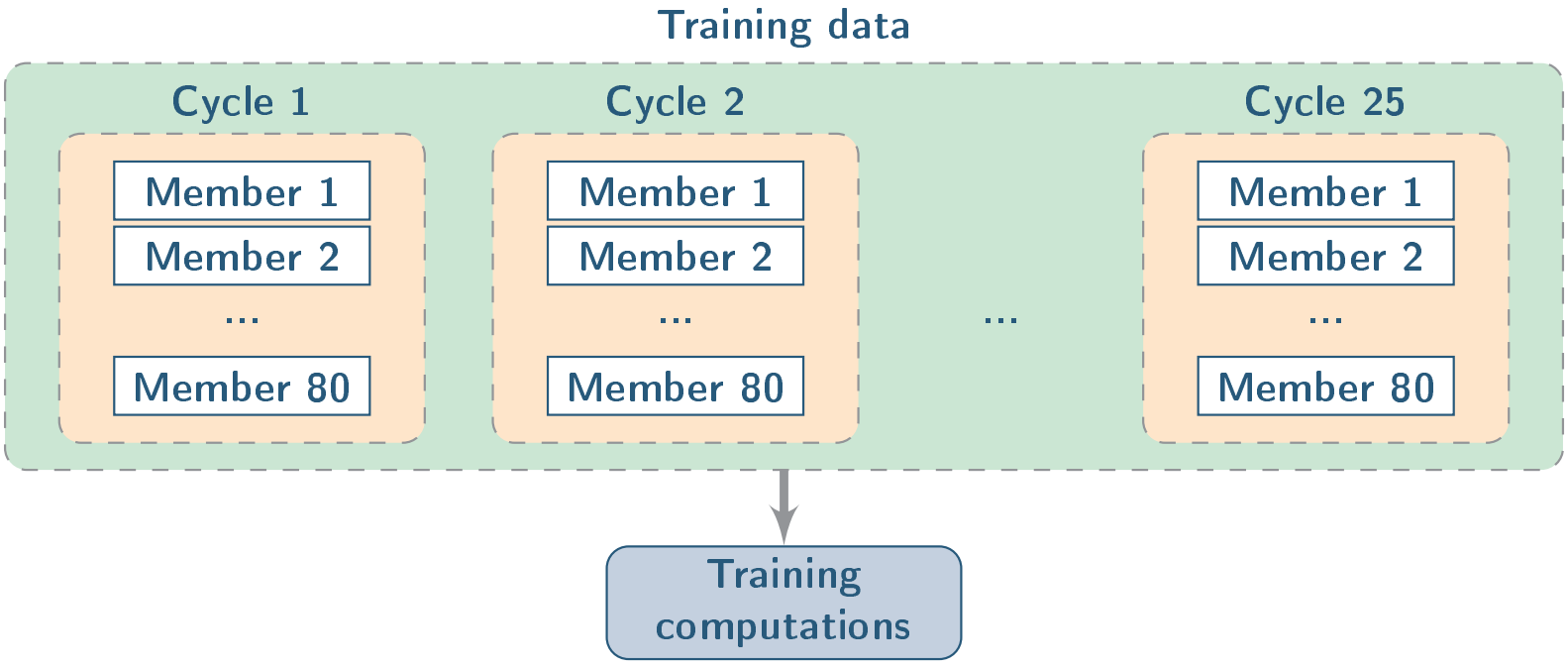
- A good approximation of the Gaspari and Cohn (1999) function square-root can be obtained by multiplying the function length-scale by an empirical scalar factor.

Outline

Critical memory issue



Critical memory issue

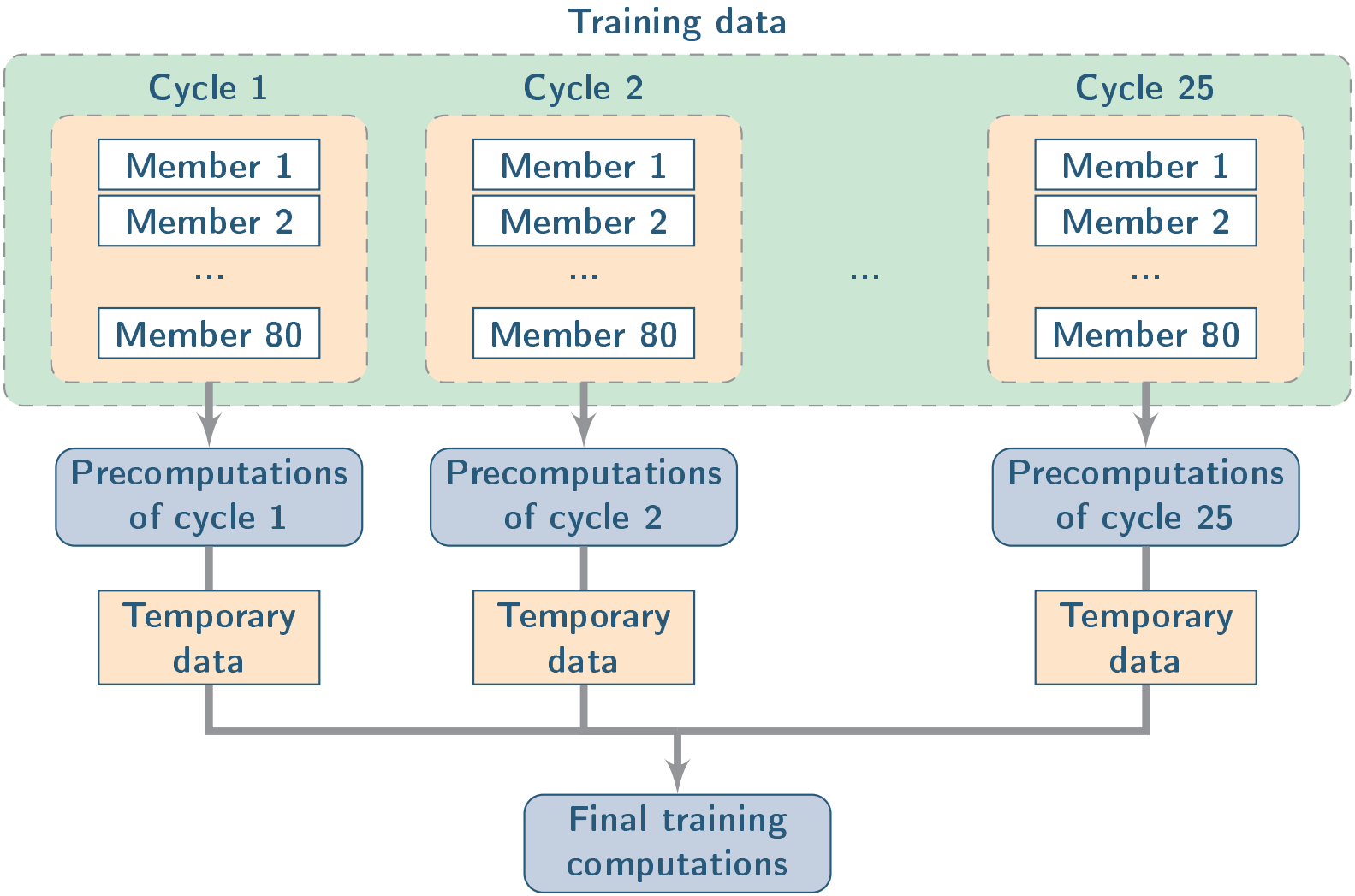


For FV3 at resolution C384:

- 1 member: $\sim$ 11 Gb
- 1 cycle of 80 members: $\sim$ 880 Gb
- 25 cycles of 80 members: $\sim$ 22 Tb

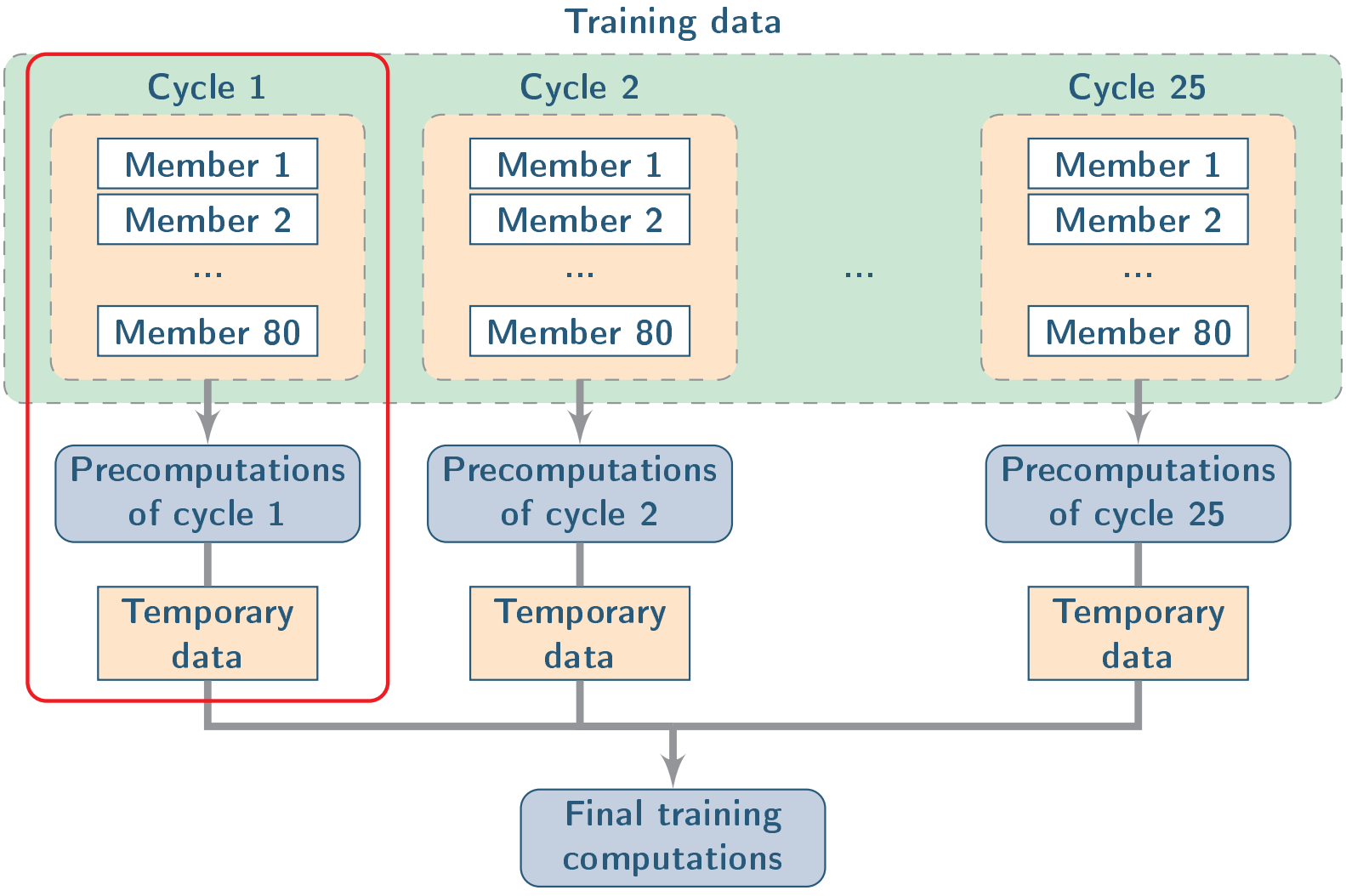
Difficult to load simultaneously on disk, not to say in memory...

BUMP solution: split the work

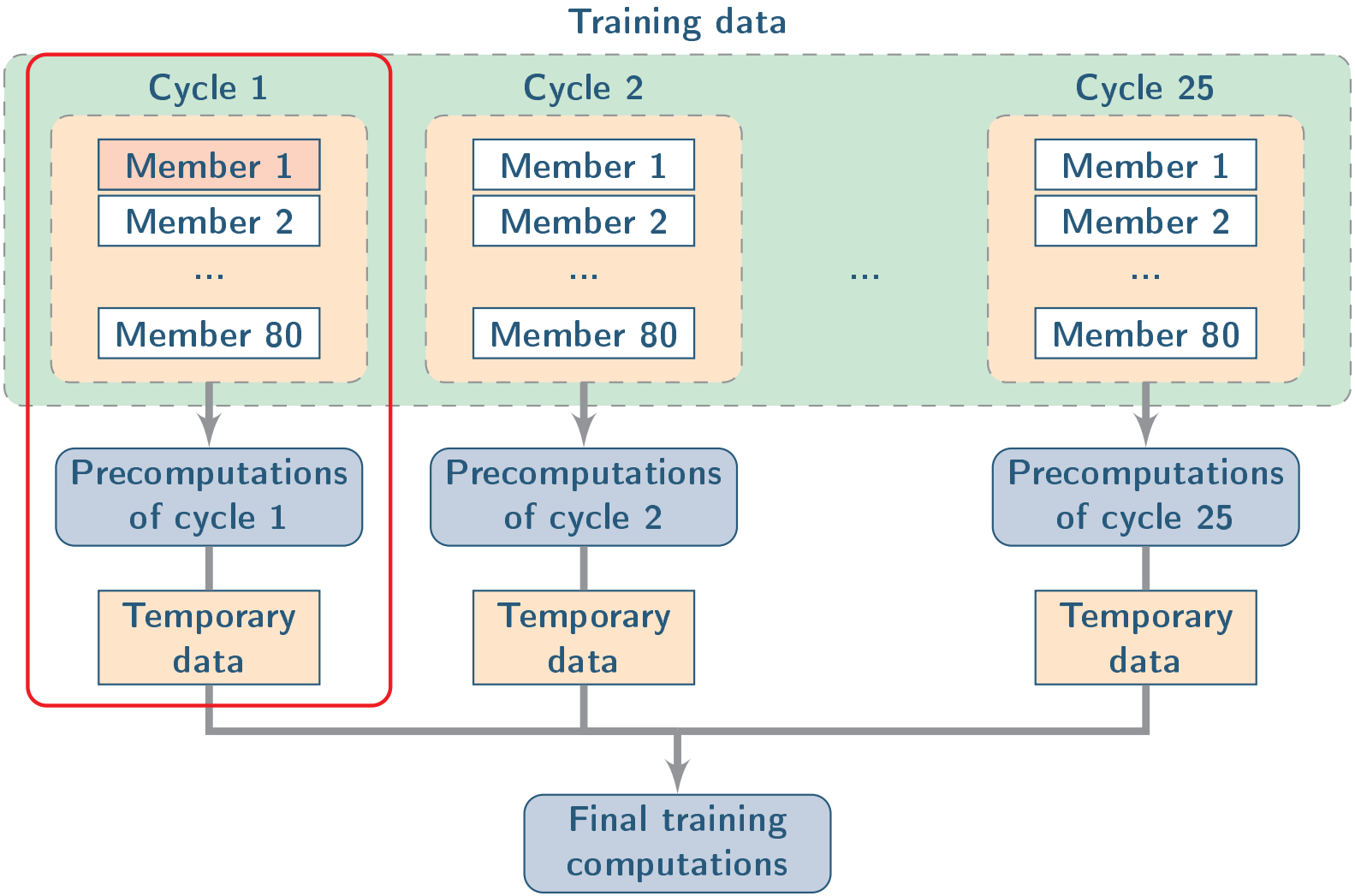


Easy to do for variance [VAR] and correlation [HDIAG / LCT], required theoretical developments for vertical regressions [VBAL].

BUMP solution: load members sequentially

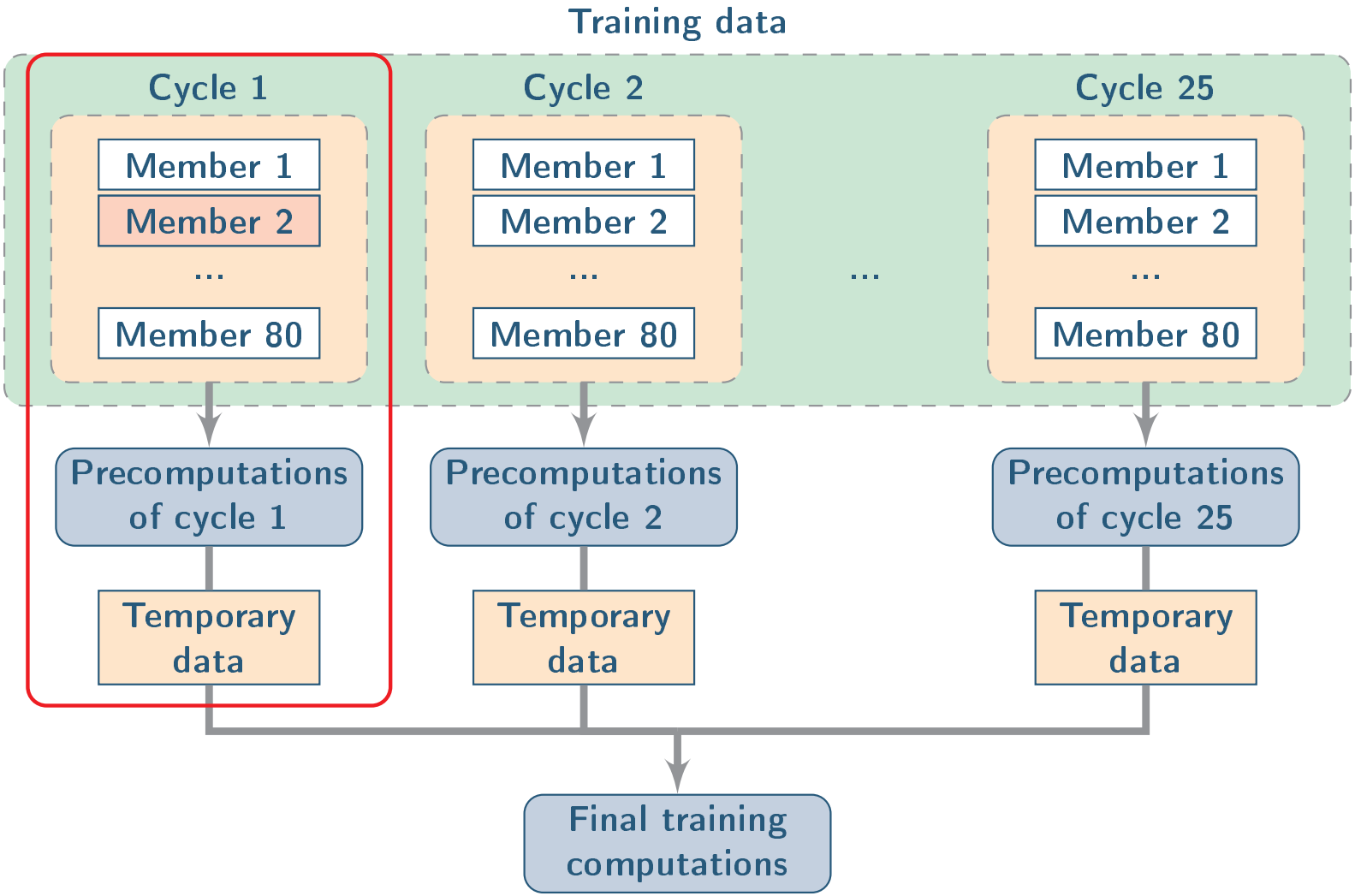


BUMP solution: load members sequentially



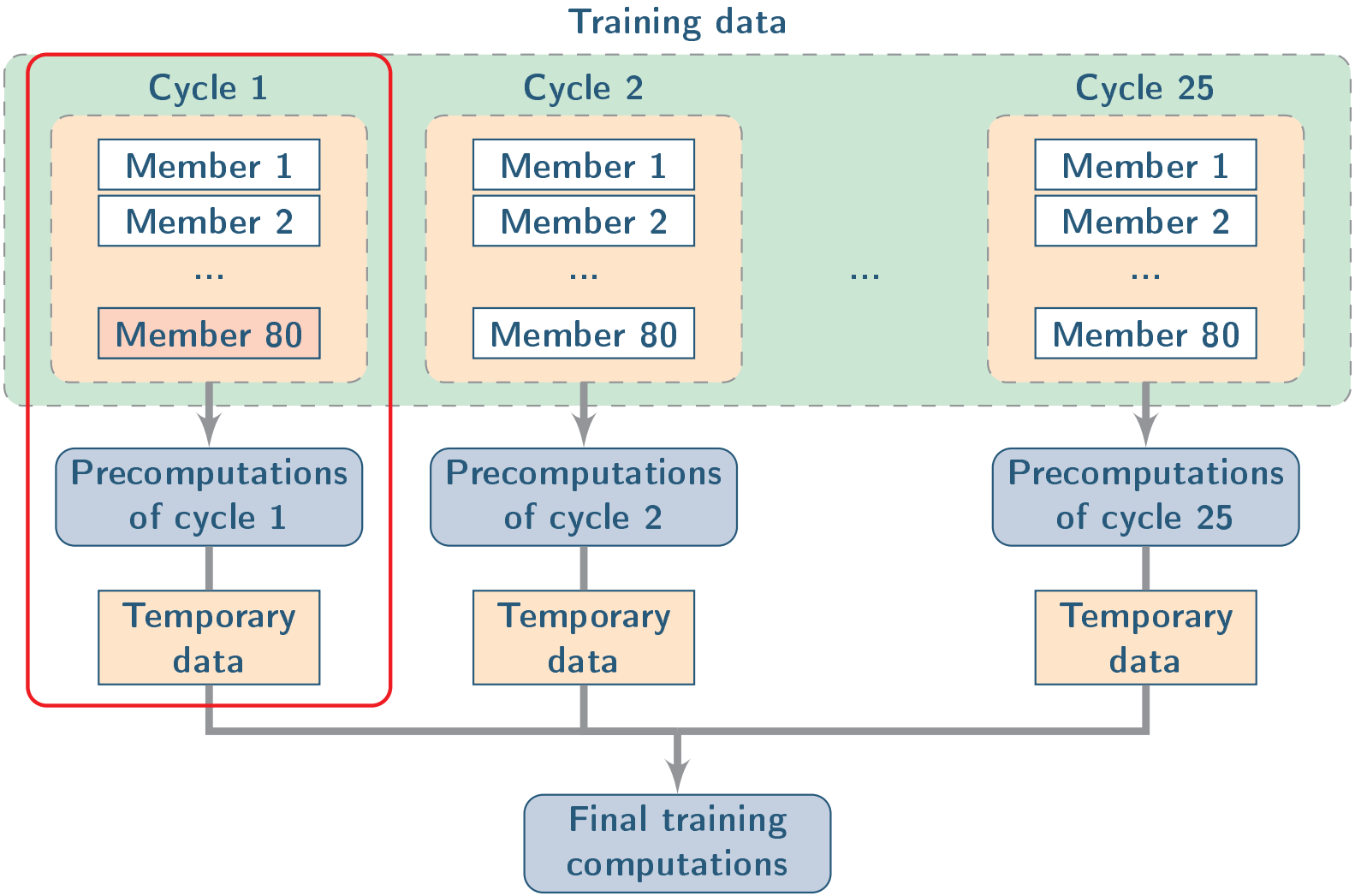
Members can be loaded sequentially to reduce the memory footprint. No slow down, no precision loss.

BUMP solution: load members sequentially



Members can be loaded sequentially to reduce the memory footprint. No slow down, no precision loss.

BUMP solution: load members sequentially



Members can be loaded sequentially to reduce the memory footprint. No slow down, no precision loss.

BUMP usage in OOPS

- BUMP is fully interfaced with OOPS.
- The BUMP parameters are set from the YAML file.
- BUMP works in two main steps:
  1. Prepare BUMP operators with `EstimateParams.h`.
     Several executions can be required for complex operators.
  2. Apply BUMP operators in an application, for instance with `Dirac.h` or `Variational.h`.
- Between steps 1 and 2, BUMP operators data can be stored in a BUMP-specific NetCDF format (subgrid data) or in the model file format (full grid data).

Prepare BUMP operators

Typical YAML structure to use `EstimateParams.h`:

```
geometry:
  ...                       # Model geometry
input variables: [...]      # Input variables for BUMP
date: ...                   # Date
background:
  ...                       # Background file
bump:
  ...                       # BUMP parameters
ensemble:
- ...                       # Ensemble members
- ...
input:
- ...                       # Input fields for BUMP
- ...                       # stored in model files
output:
- ...                       # Output fields of BUMP
- ...                       # stored in model files
```

Prepare BUMP operators

To read members sequentially, the ensemble is an item of bump:

```
geometry:
  ...                       # Model geometry
input variables: [...]      # Input variables for BUMP
date: ...                   # Date
background:
  ...                       # Background file
bump:
  ...                       # BUMP parameters
  ensemble:
  - ...                     # Ensemble members
  - ...
input:
- ...                       # Input fields for BUMP
- ...                       # stored in model files
output:
- ...                       # Output fields of BUMP
- ...                       # stored in model files
```

Applying BUMP operators

How to apply $\mathbf{B}^s = \mathbf{K}\boldsymbol{\Sigma}\mathbf{C}\boldsymbol{\Sigma}\mathbf{K}^T$?

$\rightarrow$ A central correlation operator + variable changes:

```
background error:
  covariance model: BUMP
  bump:
    ...                          # BUMP-based correlation operator
  variable changes:
  - variable change: StdDev
    ...                          # BUMP-based standard-deviation
  - variable change: StatsVariableChange
    ...                          # BUMP-based vertical balance
  - variable change: Control2Analysis
    ...                          # Psi/chi to u/v
```

Applying BUMP operators

It is possible to mix BUMP-based and model-specific operators (and other SABER-based operators in the future):

```
background error:
  covariance model: ModelSpecificKey
  ...                          # Any model-specific
  ...                          # correlation operator
  variable changes:
  - variable change: StdDev
    ...                        # BUMP-based standard-deviation
  - variable change: StatsVariableChange
    ...                        # BUMP-based vertical balance
  - variable change: Control2Analysis
    ...                        # Psi/chi to u/v
```

How to start?

- Two sources for examples:
  - the QG model also used in SABER
  - the FV3-JEDI system, with a full static B training
- BUMP parameters default values and short descriptions can be found in `saber/src/saber/bump/type_nam.fypp`
- Documentation can be found on GitHub page: `https://github.com/JCSDA/saber`
- Support through the JCSDA forum, the SABER GitHub repo for issues, or my email: `benjamin.menetrier@irit.fr`

BUMP, a generic tool for background error covariance modeling

Benjamin Menetrier, IRIT, Toulouse, France
benjamin.menetrier@irit.fr
Funded by the JCSDA

6th JEDI Academy - June 2021