BUMP, a generic tool for background error covariance modeling

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

5^{th} JEDI Academy - November 2020

Outline

Introduction

Static B

Ensemble/hybrid B

The NICAS smoother

BUMP usage

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

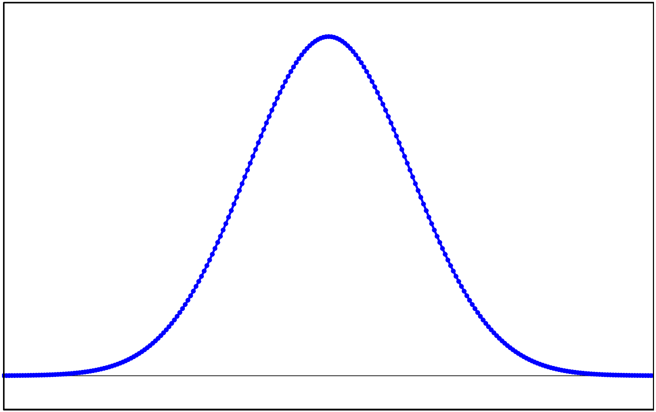
If $n^s \ll n$, then the total cost scales as $O(n)$ (interpolation cost).

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.

Convolution on a subgrid

Convolution function on model grid

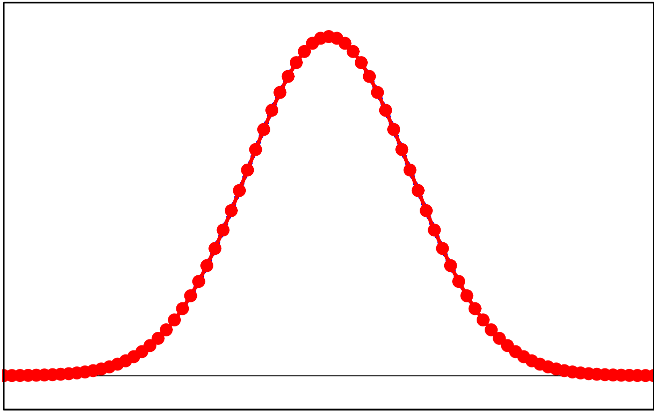


Model grid (blue)

Large convolution length-scale

Convolution on a subgrid

Subsampling: 1 point over 3

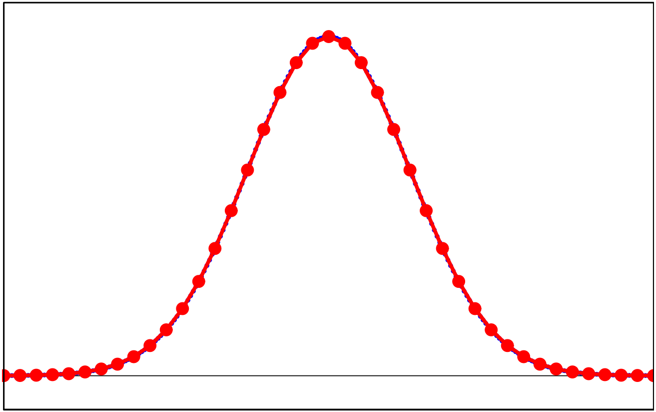


Model grid (blue) and subgrid (red)

Large convolution length-scale

Convolution on a subgrid

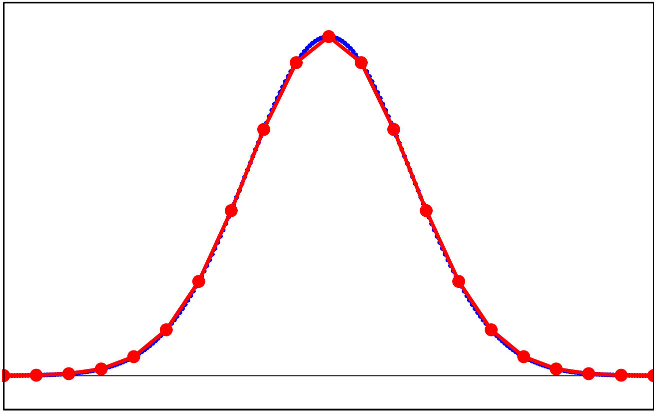
Subsampling: 1 point over 6



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

Convolution on a subgrid

Subsampling: 1 point over 12

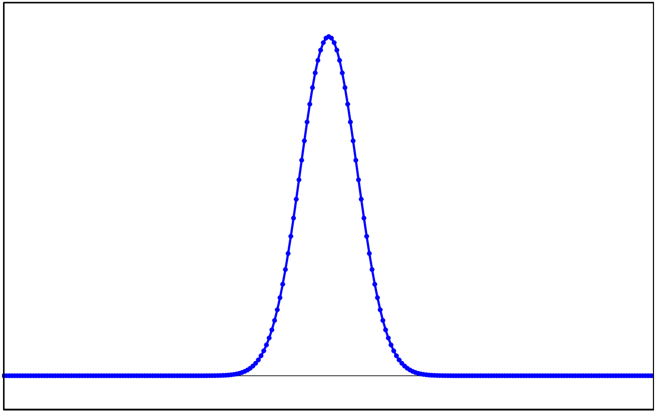


Model grid (blue) and subgrid (red)

Large convolution length-scale

Convolution on a subgrid

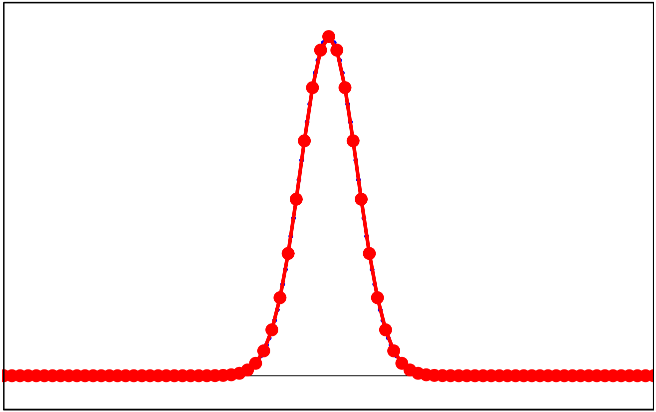
Convolution function on model grid



Model grid (blue)
Small convolution length-scale

Convolution on a subgrid

Subsampling: 1 point over 3

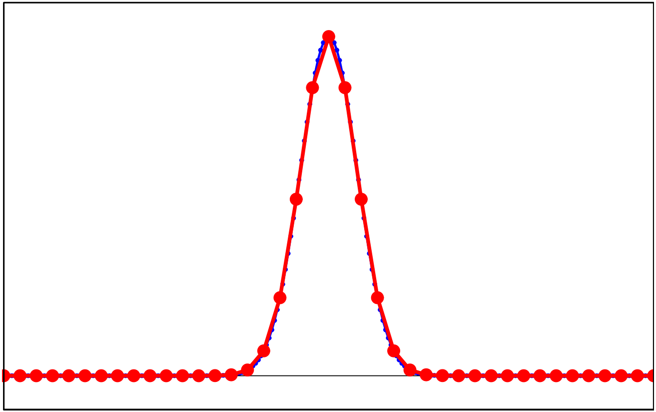


Model grid (blue) and subgrid (red)

Small convolution length-scale

Convolution on a subgrid

Subsampling: 1 point over 6

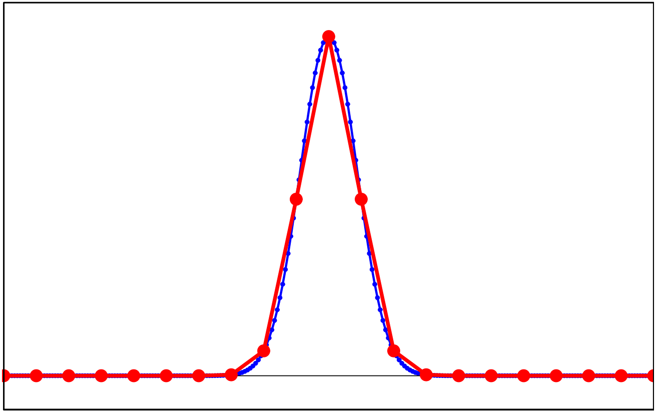


Model grid (blue) and subgrid (red)

Small convolution length-scale

Convolution on a subgrid

Subsampling: 1 point over 12



Model grid (blue) and subgrid (red)

Small convolution length-scale

Explicit convolution

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

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

where

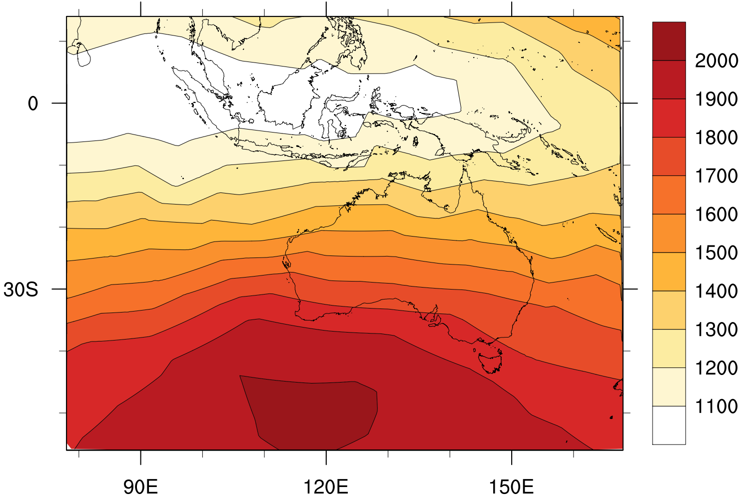
- N is a diagonal normalization matrix.
- The subgrid is locally adapted to the convolution length-scale.

To illustrate how NICAS works:

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

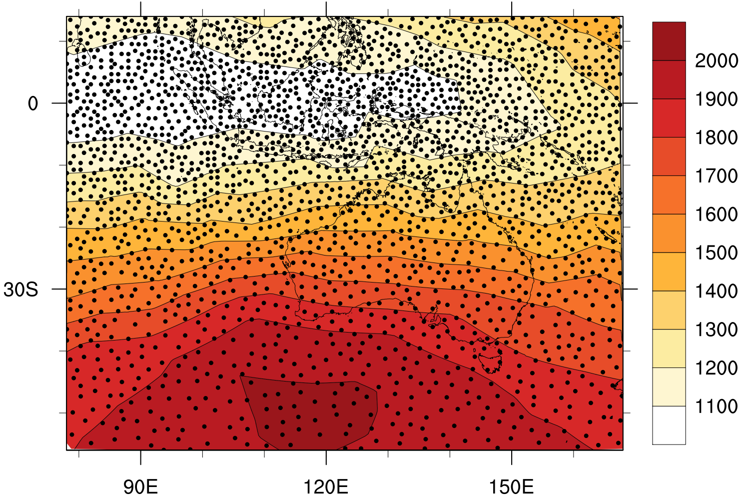
$$\delta_i^k = \begin{cases} 1 & \text{if} \quad i = k \\ 0 & \text{if} \quad i \neq k \end{cases}$$

Adaptive subgrid



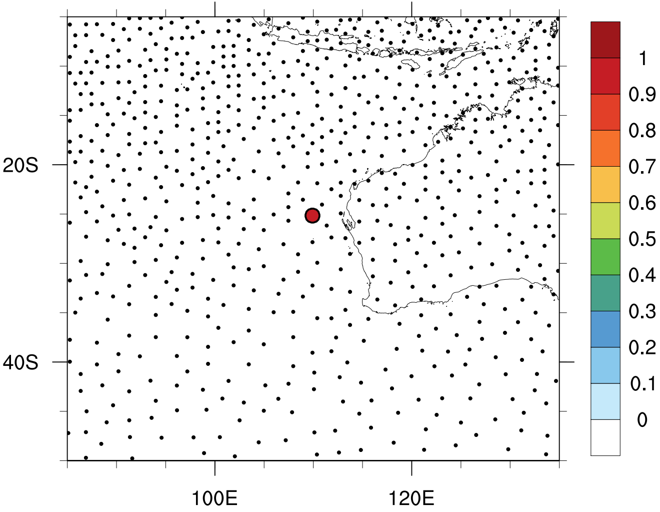
Localization support radius (km) interpolated on the model grid

Adaptive subgrid



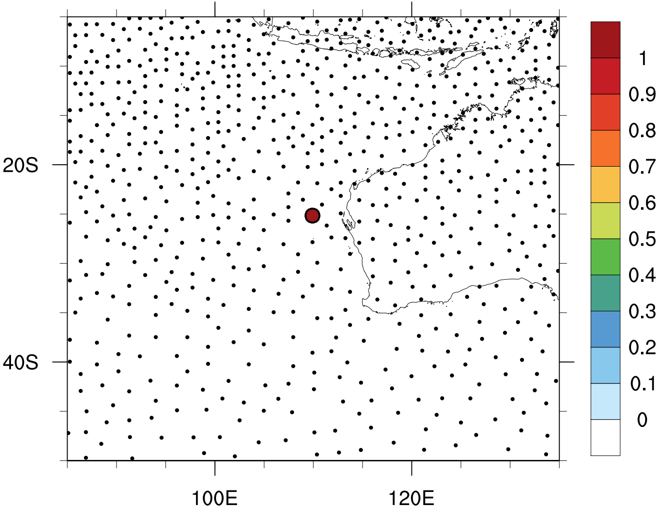
Adaptive support radius-based subgrid

NICAS steps



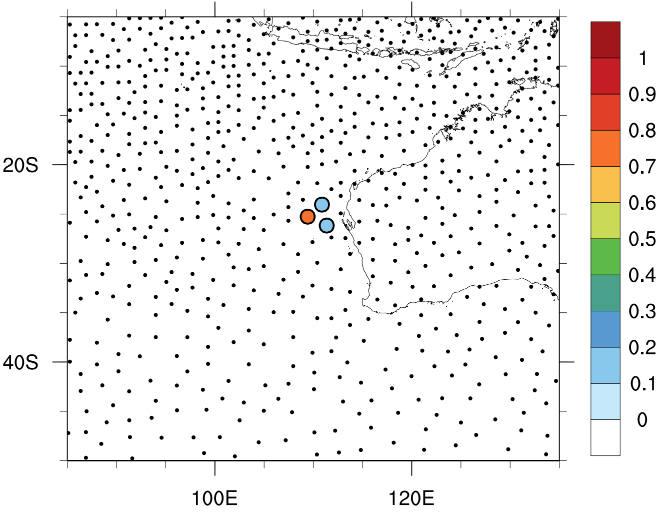
Initial vector: δ^k

NICAS steps



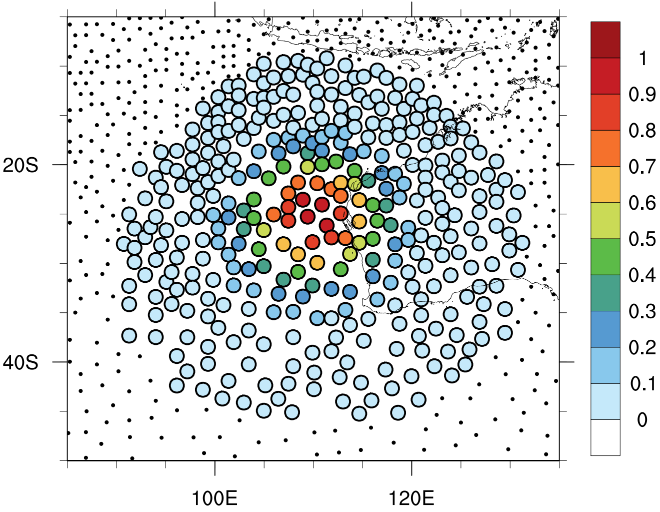
Adjoint normalization: $\mathrm{N}^{\mathrm{T}}\delta^k$

NICAS steps



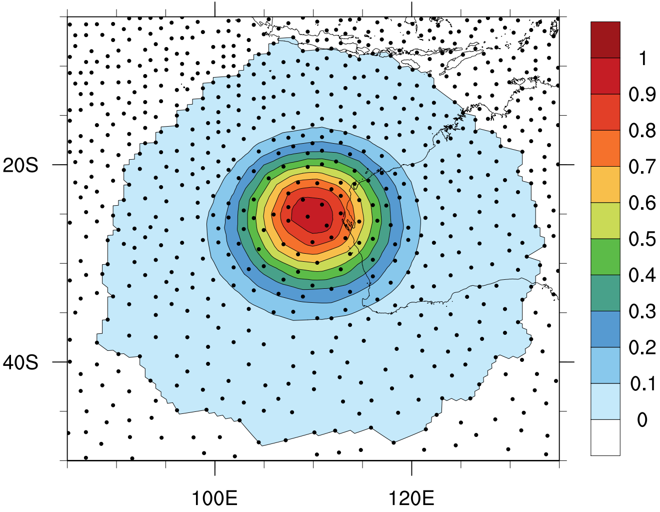
Adjoint interpolation: $\mathrm{S}^{\mathrm{T}}\mathrm{N}^{\mathrm{T}}\delta^k$

NICAS steps



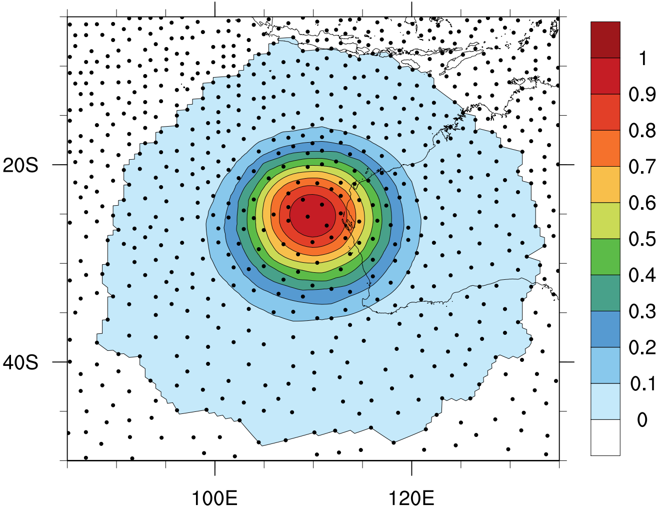
Convolution: $\mathrm{C}^s \mathrm{S}^\mathrm{T} \mathrm{N}^\mathrm{T} \delta^k$

NICAS steps



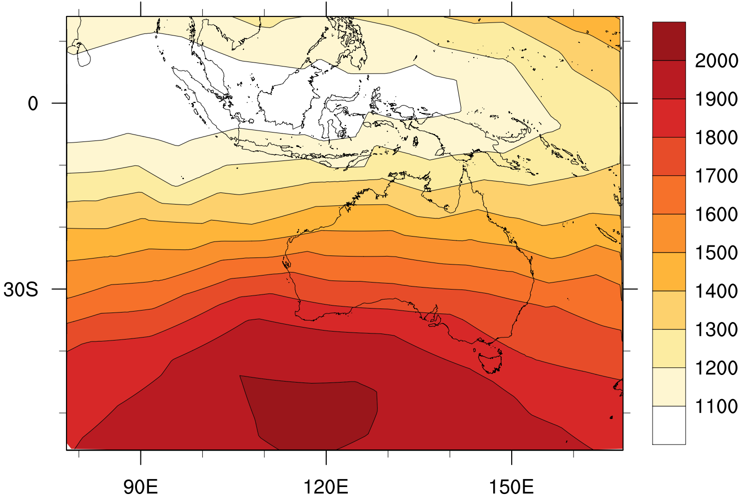
Interpolation: $\mathrm{S}\mathrm{C}^s\mathrm{S}^\mathrm{T}\mathrm{N}^\mathrm{T}\delta^k$

NICAS steps



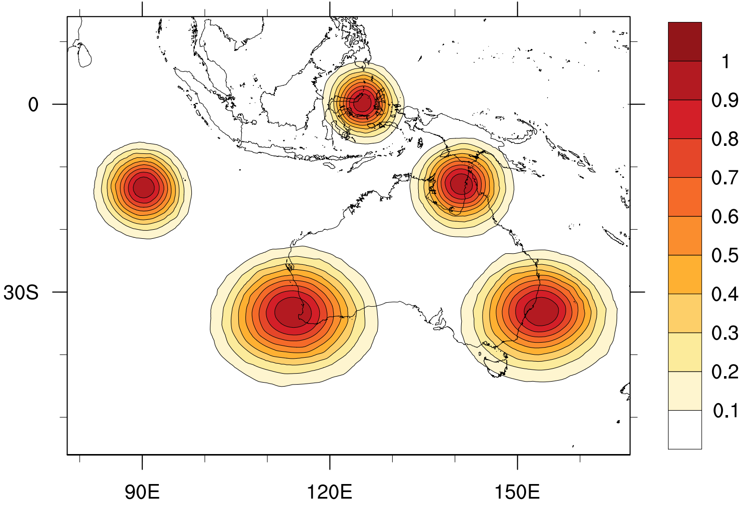
Normalization: $\mathrm{NSC}^s\mathrm{S}^\mathrm{T}\mathrm{N}^\mathrm{T}\delta^k$

Result of the Dirac test



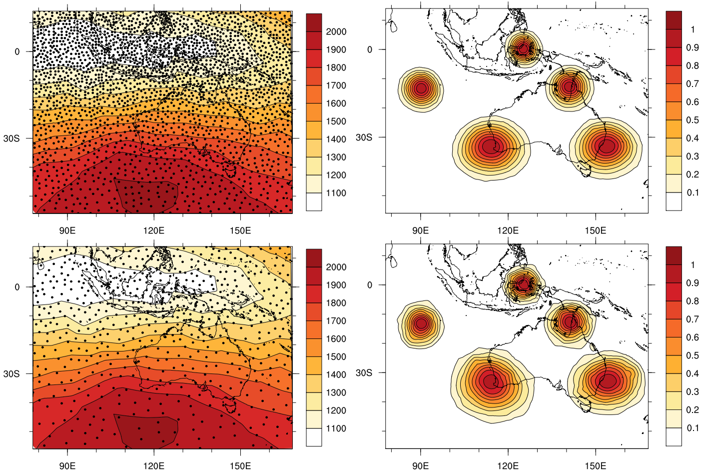
Localization support radius (km)

Result of the Dirac test



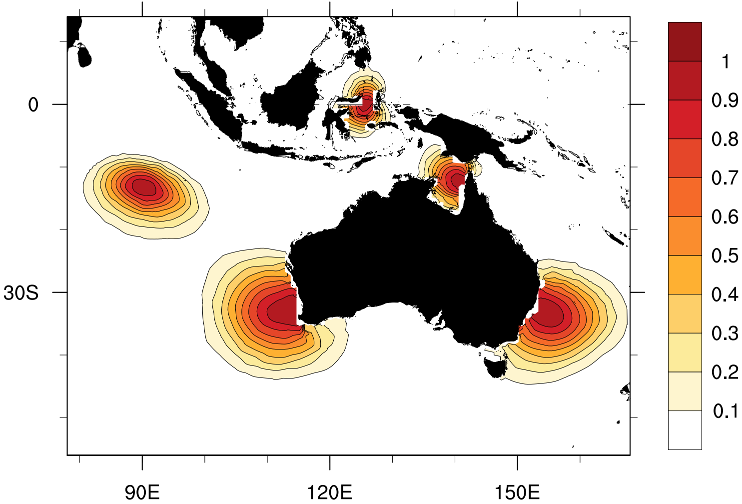
Application of NICAS on Dirac functions

Impact of the subgrid resolution



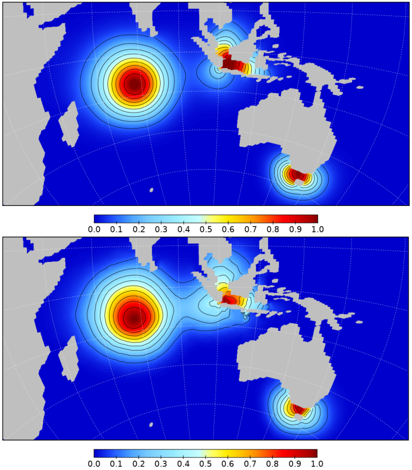
A scalar parameter controls the subgrid resolution.
Simple trade-off between cost and accuracy.

Complex boundaries



Application of masked anisotropic NICAS on Dirac functions

Complex boundaries



On the ORCA1 grid of NEMOVAR: implicit diffusion (top) and NICAS (bottom, 50x faster)

Square-root formulation

- Basic NICAS method:

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

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

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

  which can be useful for square-root preconditioning in variational methods.

- Using the formulation:

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

  also ensures that $\widetilde{\mathrm{C}}$ 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.

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:

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

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

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

  More communication steps $\Rightarrow$ smaller halos.

- Hybrid parallelization with OpenMP is used to improve efficiency.

Outline

Introduction

Static B

Ensemble/hybrid B

The NICAS smoother

BUMP usage

BUMP usage in OOPS

- BUMP is fully interfaced with OOPS.
- The BUMP parameters are set from the YAML file.
- Default parameters and short descriptions can be found in bump/type_nam.F90
- Documentation can be found on GitHub page: `https://github.com/JCSDA/saber`
- Support using the GitHub page or my email: `benjamin.menetrier@irit.fr`
- To get started, some examples of BUMP usage for a static B and for an ensemble/hybrid B.

Prepare static B with BUMP: example 1

NICAS smoother (for correlation) with a prescribed support radius

```
bump:
  datadir: bump                         # Bump-specific directory
  forced_radii: 1                       # Forced length-scale
  method: cor                           # Static correlation
  mpicom: 2                             # NICAS communication steps
  new_nicas: 1                          # New NICAS smoother
  ntry: 10                              # Subsampling quality
  prefix: your_experiment               # BUMP files base
  resol: 8.0                            # NICAS Subgrid resolution
  rh: 1000.0e3                          # Horizontal support radius
  rv: 1000.0                            # Vertical support radius
  strategy: specific_univariate         # Multivariate strategy
```

Important keys you might change:

- The horizontal support radius (in m): `rh`
- The vertical support radius (in your vertical unit): `rv`
- The NICAS subgrid resolution: `resol`

Prepare static B with BUMP: example 2

Covariance estimation with HDIAG and VAR (using an ensemble)

```
bump:
  datadir:bump                  # Bump-specific directory
  dc: 100.0e3                   # Diagnostic bins size
  method: cor                   # Static correlation
  nc1: 500                      # Diagnostic subsampling size
  nc3: 20                       # Number of diagnostic bins
  ne: 50                        # Ensemble size
  new_hdiag: 1                  # New HDIAG diagnostic
  new_var: 1                    # Compute variance
  nl0r: 11                      # Number of diagnostic levels
  ntry: 10                      # Subsampling quality
  prefix: your_experiment       # BUMP files base
  strategy: specific_univariate # Multivariate strategy
```

Important keys you might change:

- The diagnostic horizontal bin size (in m): `dc`
- The diagnostic subsampling size: `nc1`
- The number of diagnostic bins: `nc3`

Variance iterative filtering is possible with additional keys.

Prepare static B with BUMP: example 3

Use HDIAG diagnostic in NICAS

```
bump:
  datadir:bump                       # Bump-specific directory
  dc: 100.0e3                        # Diagnostic bins size
  method: cor                        # Static correlation
  mpicom: 2                          # NICAS communication steps
  nc1: 500                           # Diagnostic subsampling size
  nc3: 20                            # Number of diagnostic bins
  ne: 50                             # Ensemble size
  new_hdiag: 1                       # New HDIAG diagnostic
  new_nicas: 1                       # New NICAS smoother
  new_var: 1                         # Compute variance
  nl0r: 11                           # Number of diagnostic levels
  ntry: 10                           # Subsampling quality
  prefix: your_experiment            # BUMP files base
  resol: 8.0                         # NICAS subgrid resolution
  strategy: specific_univariate      # Multivariate strategy
```

In the same run:

- Run HDIAG to estimate correlation radius
- Use it to set up the NICAS smoother
- Run VAR to estimate the variance field

Use pre-computed static B with BUMP: example 4

Use pre-computed NICAS and variance files in another application

```
covariance model: BUMP
bump:
  datadir: bump                    # Bump-specific directory
  load_nicas: 1                    # Load pre-computed NICAS
  mpicom: 2                        # NICAS communication steps
  prefix: your_experiment          # BUMP files base
  strategy: specific_univariate    # Multivariate strategy
variable changes:
- variable change: StdDev          # Apply standard-deviation
                                   # as a variable change
  input variables: your_vars       # Input variables
  output variables: your_vars      # Output variables
  bump:
    datadir: bump                  # Bump-specific directory
    load_var: 1                    # Load pre-computed variance
    prefix: your_experiment        # BUMP files base
```

BUMP parameters appear in two sections:

- The correlation model (NICAS)
- The variable change (standard deviation)

Ensemble/hybrid B with BUMP

All the previous examples can be reused for the localization of an ensemble B, with only two keys to update:

- Method for localization: `method: loc`
- Multivariate strategy: `strategy: common`
  Other multivariate strategies are available in BUMP.

For the hybrid B the method is set at: `method: hyb-rnd` and a randomized pseudo-ensemble must be provided.

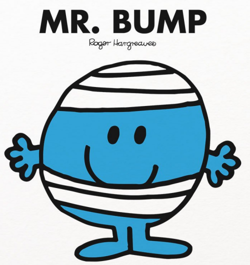
Full YAML files examples are provided with the QG model.

Other advanced BUMP features (anisotropic correlations, etc.) can also be activated from the YAML file.

BUMP, a generic tool for background error covariance modeling

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

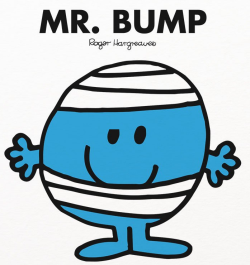
5th JEDI Academy - November 2020



BUMP, a generic tool for background error covariance modeling

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

5^{th} JEDI Academy - November 2020

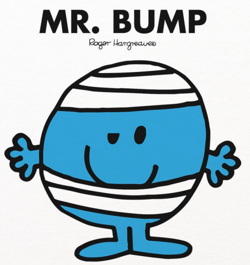


BUMP, a generic tool for background error covariance modeling

Benjamin Menetrier, IRIT, Toulouse, France
benjamin.menetrier@irit.fr

Funded by the JCSDA

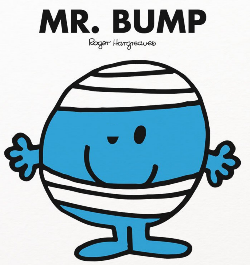
5th JEDI Academy - November 2020



BUMP, a generic tool for background error covariance modeling

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

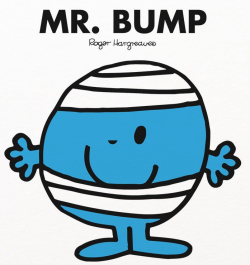
5th JEDI Academy - November 2020



BUMP, a generic tool for background error covariance modeling

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

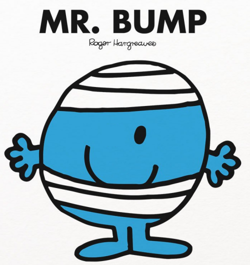
5th JEDI Academy - November 2020



BUMP, a generic tool for background error covariance modeling

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

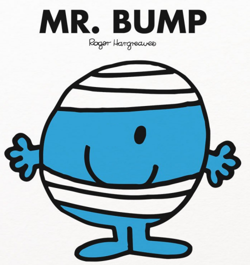
5^{th} JEDI Academy - November 2020



BUMP, a generic tool for background error covariance modeling

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

5th JEDI Academy - November 2020

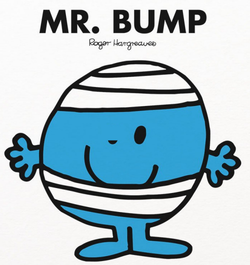


BUMP, a generic tool for background error covariance modeling

Benjamin Menetrier, IRIT, Toulouse, France
benjamin.menetrier@irit.fr

Funded by the JCSDA

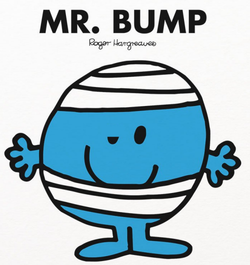
5th JEDI Academy - November 2020



BUMP, a generic tool for background error covariance modeling

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

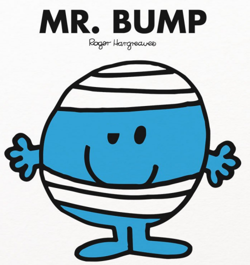
5^{th} JEDI Academy - November 2020



BUMP, a generic tool for background error covariance modeling

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

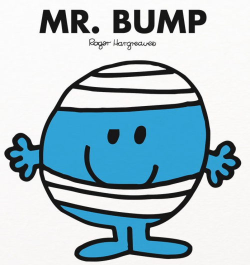
5^{th} JEDI Academy - November 2020



BUMP, a generic tool for background error covariance modeling

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

5^{th} JEDI Academy - November 2020



BUMP, a generic tool for background error covariance modeling

Benjamin Menetrier, IRIT, Toulouse, France
benjamin.menetrier@irit.fr

Funded by the JCSDA

5^{th} JEDI Academy - November 2020