Recent advances in diffusion-based correlation modelling for global ocean variational DA

A. Weaver[1], M. Chrust[2], S. Gürol[1], A. Piacentini[1], J. Tshimanga[1]

[1] CERFACS, Toulouse
[2] ECMWF, Reading

December 13, 2016

Outline

- Data assimilation (DA) algorithms for global ocean applications require manipulating huge covariance matrices ($\mathbf{C}$).
- In variational data assimilation (VDA), $\mathbf{C}$ are defined by means of a matrix-vector product (covariance operator) $\mathbf{C}\boldsymbol{\psi}$ for some vector $\boldsymbol{\psi}$.
- This is the key algorithmic feature of VDA that makes it possible to account for full-rank, non-diagonal formulations of $\mathbf{C}$.
- Rather than defining $\mathbf{C}\boldsymbol{\psi}$ through an explicit matrix-vector multiplication (which is only possible by invoking a reduced-rank approximation) or through an integral transform (e.g., spectral), we can define it as the **solution of a PDE**.
- This is the basic idea behind the **diffusion**-based approaches to covariance modelling.

We define $\mathbf{C}\boldsymbol{\psi}$ to be the matrix representation of the linear operator $C : \psi \rightarrow \psi_M$ for $\psi, \psi_M \in \mathbb{R}^d$, given by the solution of the d-dimensional elliptic equation

$$(1 - \nabla \cdot \boldsymbol{\kappa} \nabla)^M \psi_M = \psi \tag{1}$$

where M is a positive (preferably even) integer, and $\boldsymbol{\kappa}$ is a scale tensor.

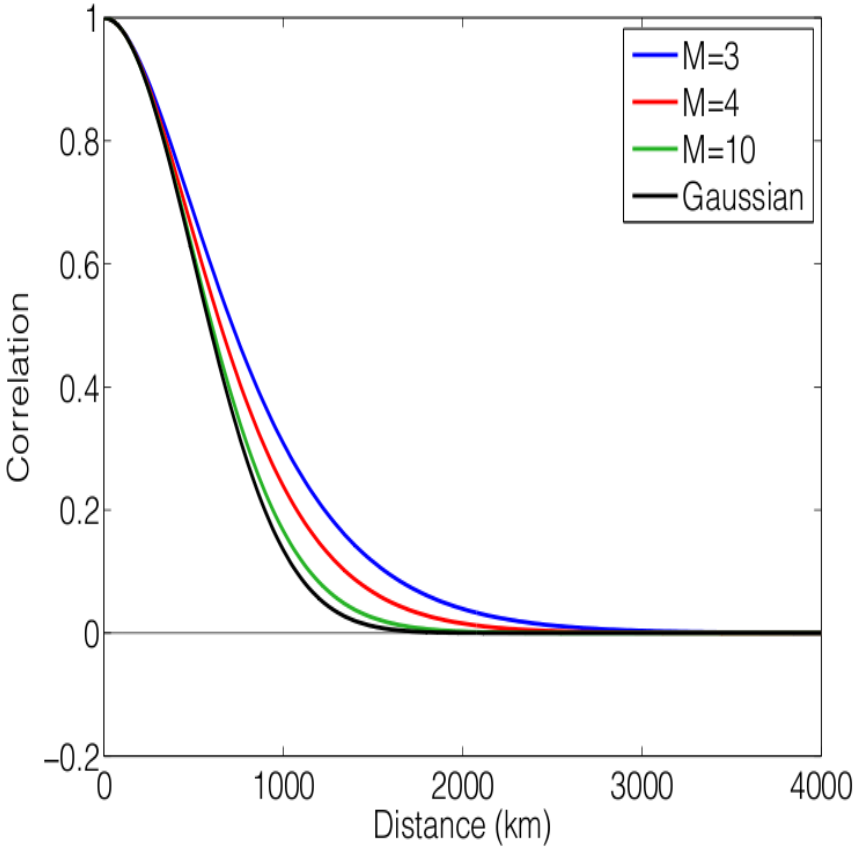
- Eq. (1) can be interpreted as an implicitly formulated diffusion operator acting over M pseudo-time steps of unit length, and with $\boldsymbol{\kappa}$ the diffusion tensor.
- For constant $\boldsymbol{\kappa}$, Eq. (1) admits covariance functions from the **Matérn class** (Guttorp and Gneiting 2006).
- The PDE can be generalized to represent a wider class of covariance functions (e.g., oscillatory).
- Eq. (1) has been studied quite extensively for ocean VDA (e.g., see Weaver and Mirouze 2013).

$$(1 - \nabla \cdot \boldsymbol{\kappa} \nabla)^M \psi_M = \psi \tag{1}$$

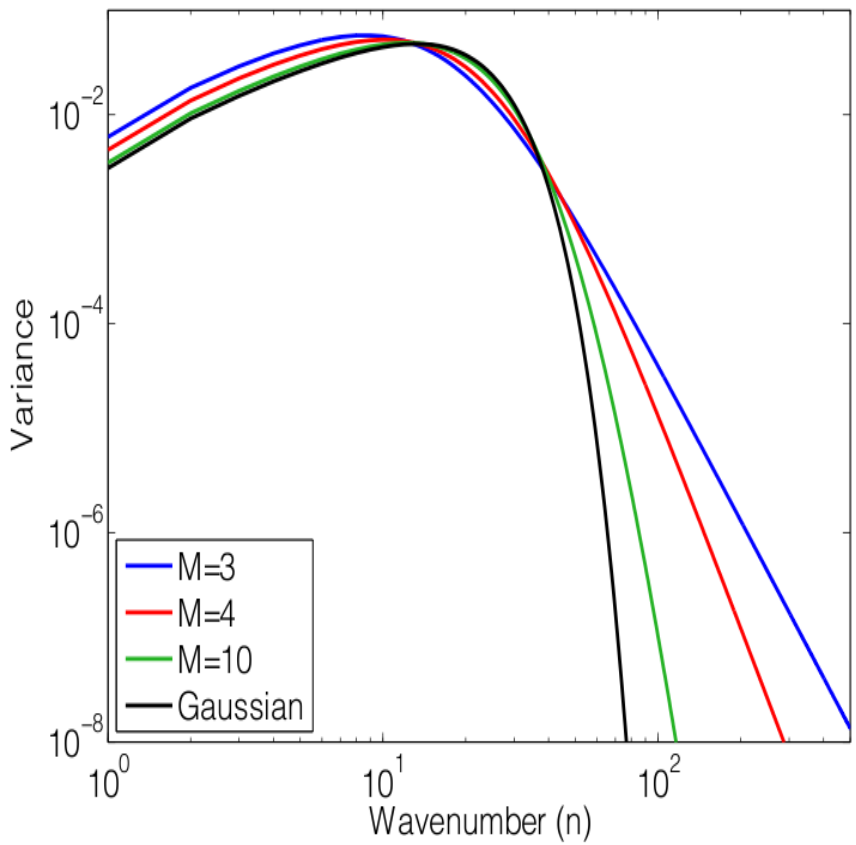
- Covariance operators based on Eq. (1) have been studied independently in fields other than DA.
  - *Spatial statistics* (Whittle 1963; Lindgren *et al.* 2010; Simpson *et al.* 2016)
  - *Seismic inversion* (Bui-Thanh *et al.* 2013)
  - *Uncertainty Quantification* (Gmeiner *et al.* 2016)
- Those studies have focused on solutions with $M = 2$, and discretizations using finite-element methods **(see O. Guillet poster)**.
- The original Whittle model is actually very general, and is based on a fractional PDE (non-integer values of M). This is not practical for numerical applications.

- Example of the shape and spectrum for different M and constant κ.

Correlation function



Variance spectrum



- Consider the NEMOVAR hybrid **B** formulation:

$$\mathbf{B} \;=\; \beta_{\mathrm{m}}^2 \underbrace{\left(\mathbf{B}_{\mathrm{m}_1} + \mathbf{B}_{\mathrm{m}_2} + \ldots\right)}_{\mathbf{B}_{\mathrm{m}}} + \beta_{\mathrm{e}}^2\, \mathbf{B}_{\mathrm{e}}$$

  wnere β_{m}^2 and $\beta_{\mathrm{e}}^2$are constant weights.

- Multiple covariance models for representing different scales:

$$\mathbf{B}_{\mathrm{m}_i} \;=\; \mathbf{K}_{\mathrm{b}}\, \mathbf{D}_i^{1/2}\, \mathbf{C}_{\mathrm{m}_i}\, \mathbf{D}_i^{1/2}\, \mathbf{K}_{\mathrm{b}}^{\mathrm{T}}$$

- A localized ensemble-based correlation matrix:

$$\mathbf{B}_{\mathrm{e}} \;=\; \mathbf{K}_{\mathrm{b}}\, \mathbf{D}_{\mathrm{e}}^{1/2} \left(\mathbf{L} \circ \widetilde{\mathbf{X}}\, \widetilde{\mathbf{X}}^{\mathrm{T}}\right) \mathbf{D}_{\mathrm{e}}^{1/2}\, \mathbf{K}_{\mathrm{b}}^{\mathrm{T}}$$

  where the columns of $\widetilde{\mathbf{X}} = \mathbf{D}_{\mathrm{e}}^{-1/2}\, \mathbf{K}_{\mathrm{b}}^{-1}\, \mathbf{X}^{\mathrm{b}}$ are transformed background ensemble perturbations.

- The localization matrix **L** is a correlation matrix. In operator form:

$$\left(\mathbf{L} \circ \widetilde{\mathbf{X}}\, \widetilde{\mathbf{X}}^{\mathrm{T}}\right)\mathbf{v} = \sum_{p=1}^{N_{\mathrm{e}}} \left(\widetilde{\mathbf{x}}_p \circ \mathbf{L}\left(\widetilde{\mathbf{x}}_p \circ \mathbf{v}\right)\right) \quad \text{where} \quad \widetilde{\mathbf{X}} = \left(\widetilde{\mathbf{x}}_1, \ldots, \widetilde{\mathbf{x}}_{N_{\mathrm{e}}}\right)$$

$$(1 - \nabla \cdot \boldsymbol{\kappa} \nabla)^M \psi_M = \psi$$

- Let **A** be the discretized, matrix representation of the self-adjoint operator $1 - \nabla \cdot \boldsymbol{\kappa} \nabla$.
- The identity $\mathbf{C} = \left(\mathbf{A}^M\right)^{-1} = \left(\mathbf{A}^{-1}\right)^M$ suggests two ways of applying **C**.
  1. Solve the single linear, self-adjoint, positive-definite (SAPD) system
  $$\mathbf{A}^M \boldsymbol{\psi}_M = \boldsymbol{\psi} \tag{2}$$
  2. Solve the sequence of M linear SAPD systems
  $$\left.\begin{array}{rcl} \mathbf{A}\boldsymbol{\psi}_1 & = & \boldsymbol{\psi} \\ \mathbf{A}\boldsymbol{\psi}_2 & = & \boldsymbol{\psi}_1 \\ & \vdots & \\ \mathbf{A}\boldsymbol{\psi}_M & = & \boldsymbol{\psi}_{M-1} \end{array}\right\} \tag{3}$$
- The linear systems in (3) are better conditioned than the linear system (2).
- Between 10^5 and 10^6 linear systems must be solved on a typical assimilation cycle of an ocean hybrid **B** VDA system $\Longrightarrow$ efficiency is crucial!

The answer to this question is problem dependent. Here, we are concerned with applications to **global ocean models**.

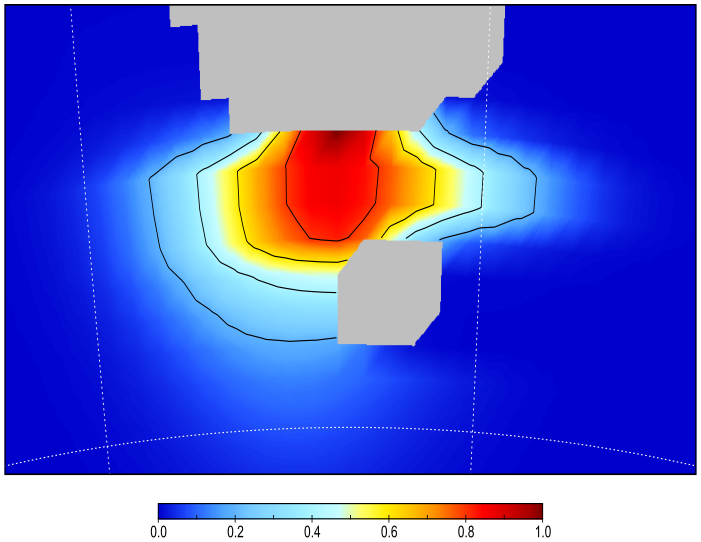
- *Direct solver* (Cholesky factorization,...).
  Accurate but memory requirements can be large, technical implementation and parallelization can be difficult.

- *Multigrid solver*.
  Well suited for elliptic problems but difficult to design a hierarchy of grids and associated mapping operators when geometry is complex and grids are nonstandard.

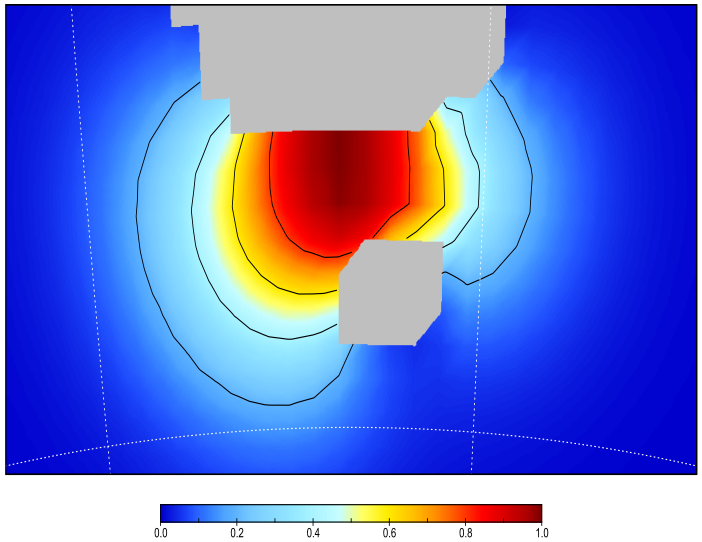
- *Approximate the 2D operator as a product of simpler 1D operators*.
  (cf. recursive filter). Resulting algorithm involves small, easily invertible matrices but is difficult to generalize and parallelize.

- *Polynomial-based iterative methods* (conjugate gradient, Chebyshev iteration,...). Straightforward to implement and parallelize but possibly slow convergence.

- The n×1D approach has some appealing properties, BUT is difficult to make anisotropic, produces numerical artefacts near complex boundaries, and scales poorly on massively parallel machines.



(a) 2 × 1D



(b) 2D

- These problems, especially lack of scalability, compelled us to develop a new approach.

- The **conjugate gradient** method and the **Chebyshev iteration** belong to this class of solvers.
- The linear system described previously can be transformed to standard form

$$\widehat{\mathbf{A}}\mathbf{x} = \mathbf{b}$$

where $\widehat{\mathbf{A}}$ is symmetric, positive definite (SPD).

- A **polynomial-based iterative method** produces, at iteration k, an estimate $\mathbf{x}_k$ of the true solution $\mathbf{x}^* = \widehat{\mathbf{A}}^{-1}\mathbf{b}$ such that the error is

$$\mathbf{x}^* - \mathbf{x}_k = \varphi_k(\widehat{\mathbf{A}})\,(\mathbf{x}^* - \mathbf{x}_0)$$

where $\varphi_k(\widehat{\mathbf{A}})$ is a polynomial in $\widehat{\mathbf{A}}$ of degree k with $\varphi_k(0) = 1$.

- This equation can also be expressed as

$$\mathbf{r}_k = \varphi_k(\widehat{\mathbf{A}})\,\mathbf{r}_0$$

where $\mathbf{r}_k = \widehat{\mathbf{A}}\mathbf{x}_k - \mathbf{b}$ is the residual vector.

- The resulting iterative algorithms involve recurrence relations. (The *orthogonal residual polynomials* $\varphi_k(\widehat{\mathbf{A}})$ are never constructed explicitly).

$$\mathbf{x}^* - \mathbf{x}_k \;=\; \varphi_k(\widehat{\mathbf{A}})(\mathbf{x}^* - \mathbf{x}_0),$$

Conjugate gradients (CG)

Choose the polynomial $\varphi_k = \varphi_k^{\mathrm{CG}}$ to be the unique solution of

$$\min_{\substack{\varphi(0)\,=\,1 \\ \deg(\varphi)\,\leq\,k}} \left\|\varphi(\widehat{\mathbf{A}})(\mathbf{x}^* - \mathbf{x}_0)\right\|_{\widehat{\mathbf{A}}}$$

Chebyshev iteration (CI)

Choose the polynomial $\varphi_k = \varphi_k^{\mathrm{CI}}$ to be the unique solution of

$$\min_{\substack{\varphi(0)\,=\,1 \\ \deg(\varphi)\,\leq\,k}} \max_{\lambda\in[\lambda_{\min},\lambda_{\max}]} \left|\varphi(\lambda)\right|$$

where $\lambda_{\min}$ and $\lambda_{\max}$ are the extreme eigenvalues of $\widehat{\mathbf{A}}$. The resulting φ_k^{CI} are *shifted and scaled Chebyshev polynomials*.

- Convergence in at most N iterations (in exact arithmetic) where N is the dimension of $\mathbf{x}$.
- Often affected by round-off error

  => may need re-orthogonalization of residual vectors.
- **Parameter-free method**.
- Computation of α_k and β_k in CG involve **inner products**.
  - **Global MPI communications** in a parallel domain decomposition.

    => a well-known performance bottleneck on massively parallel machines.
  - **Nonlinear solver** if stopped before full convergence

    => the linearity and symmetry of $\mathbf{C}$ is no longer guaranteed.

- Convergence is not guaranteed in a finite number of iterations.
- Not seriously affected by round-off error.
- **Parameter-dependent method**

  => requires estimates of the eigenvalue bounds of $\widehat{\mathbf{A}}$.

  => these can be set to the extreme Ritz values $\theta_{\min}$ and $\theta_{\max}$ pre-computed with a combined CG/Lanczos procedure.
- Computation of α_k and β_k depends on the input parameters.
  - **No inner products**

    => no global MPI communications are required.

    => local MPI communications are needed before each application of $\widehat{\mathbf{A}}$.
  - **Strictly linear solver**

    => linearity of $\mathbf{C}$ is guaranteed even for a finite number of iterations.

    => exact symmetry of $\mathbf{C}$ can be enforced numerically using a factored formulation of $\mathbf{C} = \mathbf{U}\mathbf{U}^{\mathrm{T}}$ and a fixed number of iterations (K).

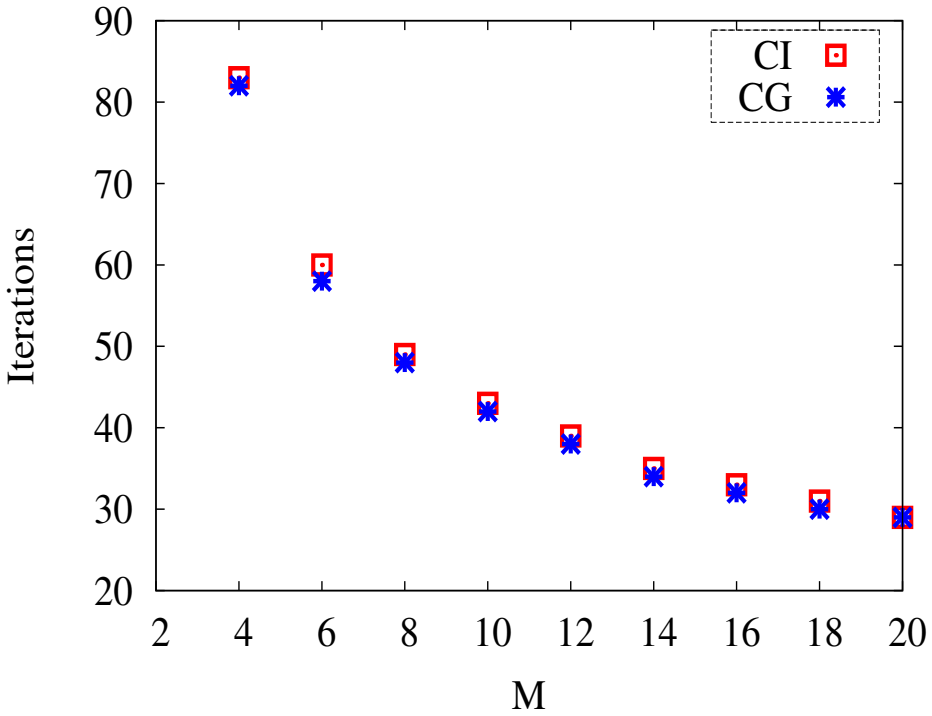
Chebyshev Iteration

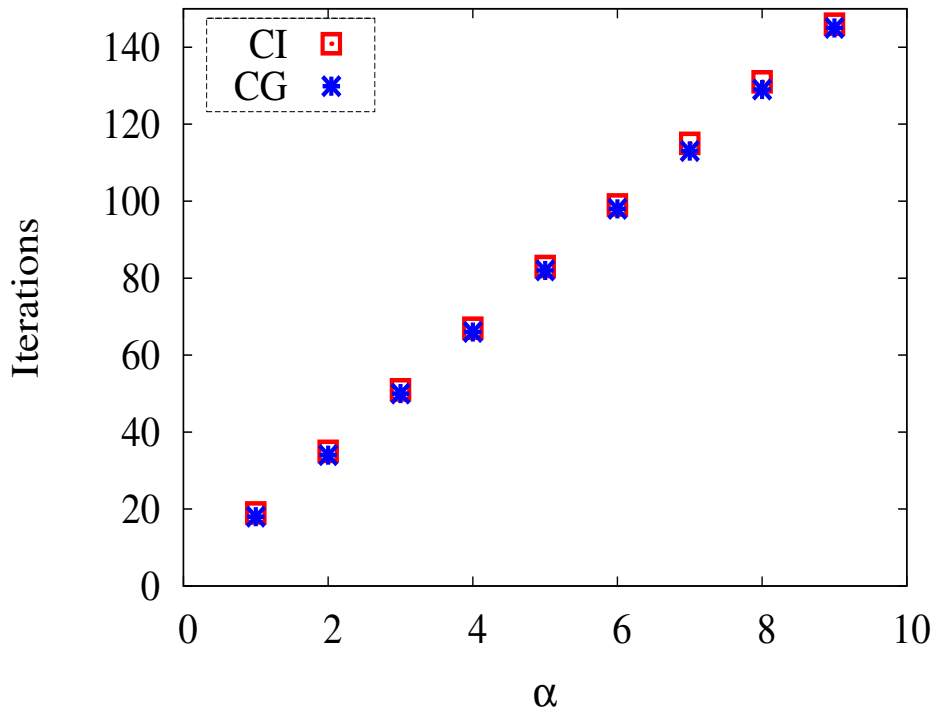
- $\sigma = (\theta_{\max} + \theta_{\min})/2$
- $\delta = (\theta_{\max} - \theta_{\min})/2$
- $\alpha_0 = 1/\sigma \quad ; \quad \beta_1 = (\delta\alpha_0)^2/2$
- **for** $k = 1, \ldots, K-1$ **do**
  - $\alpha_k = 1/(\sigma - \beta_k/\alpha_{k-1})$
  - $\beta_{k+1} = (\delta\,\alpha_k/2)^2$
- **end for**
- $\mathbf{x}_0$ = initial estimate
- $\mathbf{r}_0 = \widehat{\mathbf{A}}\mathbf{x}_0 - \mathbf{b}$
- $\mathbf{p}_0 = -\mathbf{r}_0$
- **for** $k = 0, \ldots, K-1$ **do**
  - $\mathbf{q}_k = \widehat{\mathbf{A}}\mathbf{p}_k$
  - $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k\mathbf{p}_k$
  - $\mathbf{r}_{k+1} = \mathbf{r}_k + \alpha_k\mathbf{q}_k$
  - $\mathbf{p}_{k+1} = -\mathbf{r}_{k+1} + \beta_{k+1}\mathbf{p}_k$
- **end for**

$$\boldsymbol{\kappa} = \mathrm{diag}\,(\kappa_1, \kappa_2) \quad \text{where} \quad \kappa_i = \frac{1}{2(M-2)} D_i^2 \quad \text{and} \quad D_i = \alpha e_i$$

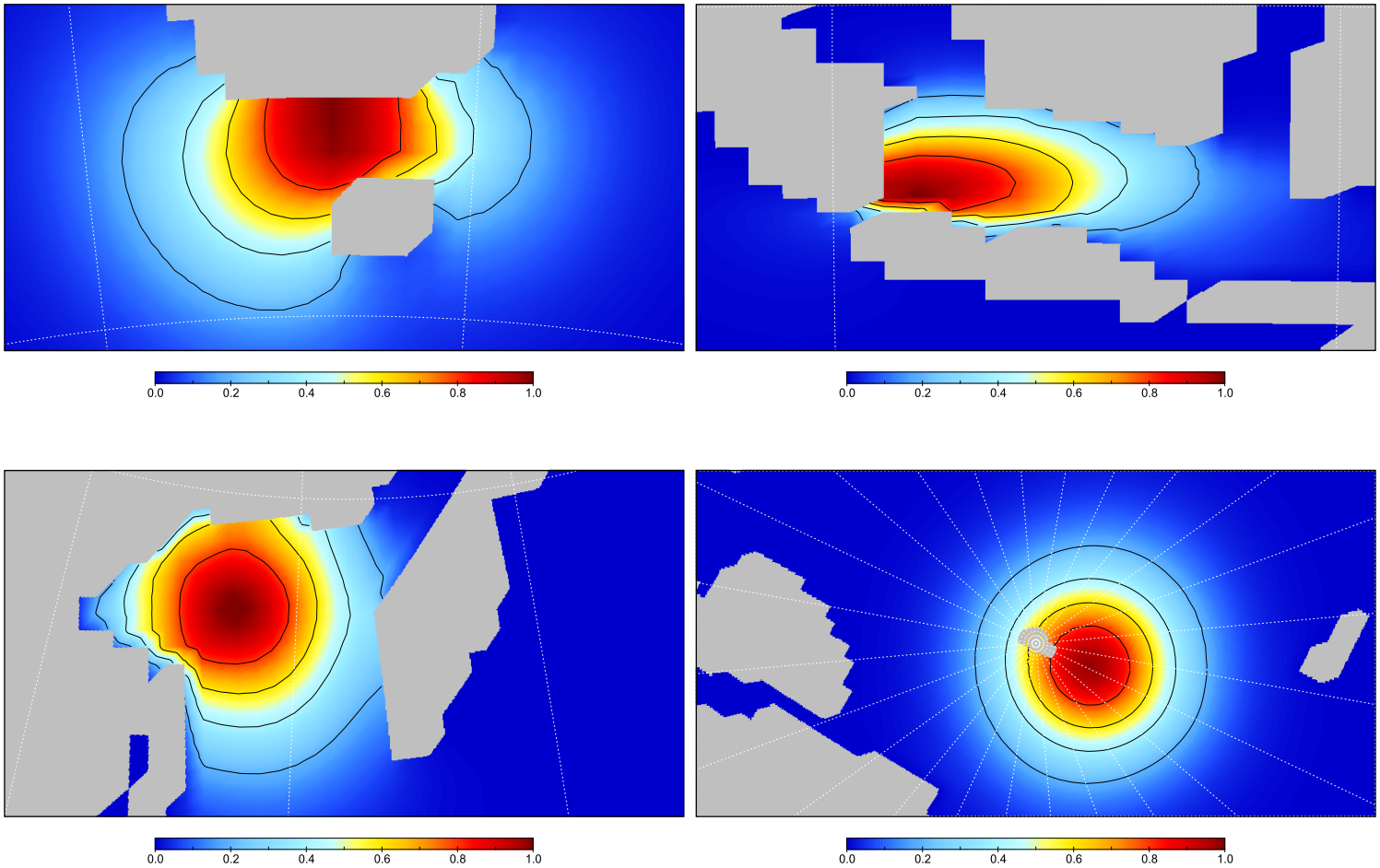
- CI and CG have very similar convergence properties **for the elliptic problem under consideration**, which is reasonably well conditioned.

$$\lambda_{\min} \approx 1 \quad \text{and} \quad \lambda_{\max} \approx 1 + \left(\frac{\kappa_1}{e_1^2} + \frac{\kappa_2}{e_2^2} \right) \approx 1 + \frac{4\,\alpha^2}{M-2}$$





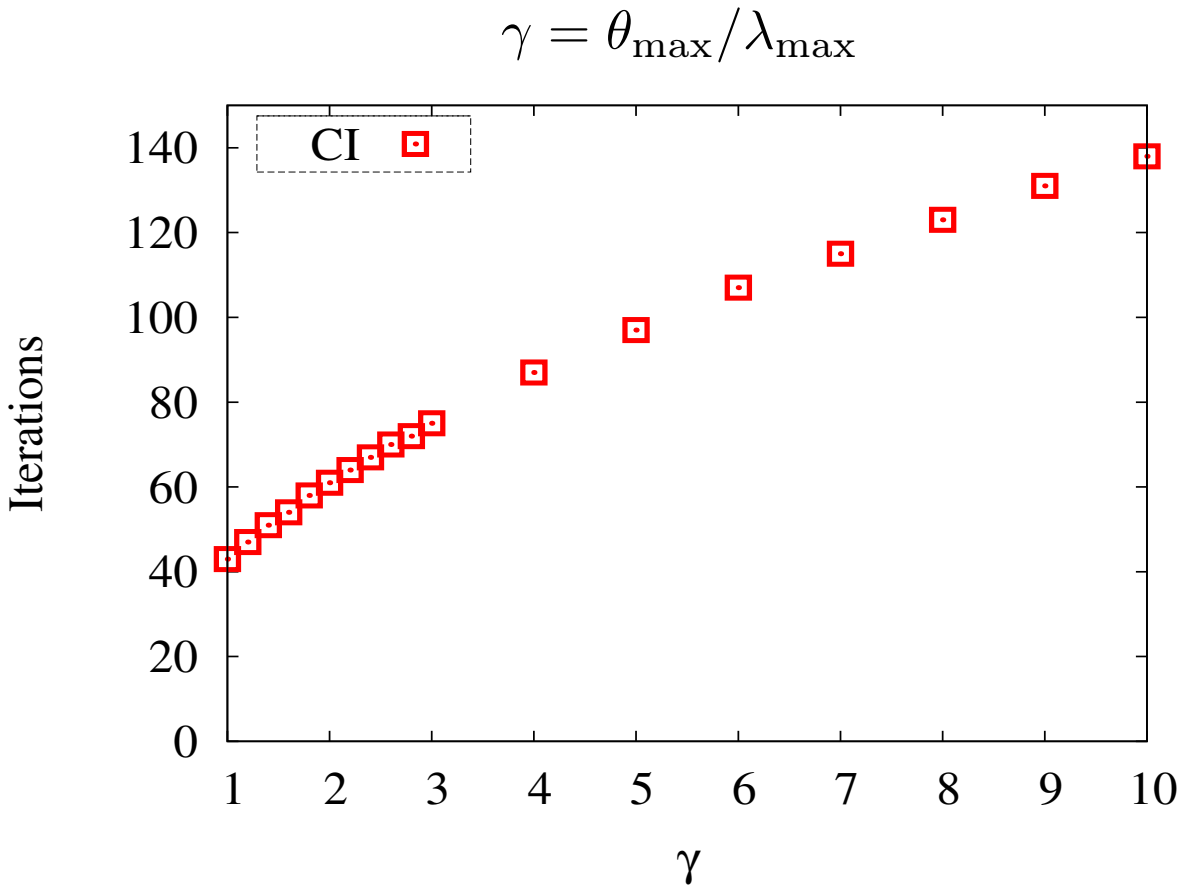
Correlations at selected points with $M = 10$ and $\alpha = 5$.



(From Weaver *et al.* 2016)

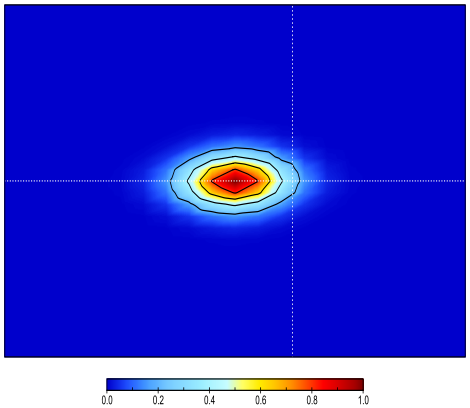
What happens if $\lambda_{\max}$ is misspecified?

- For this system, $\lambda_{\min} \approx 1$, so only $\lambda_{\max}$ needs to be estimated.
- Underestimating $\lambda_{\max}$ causes the algorithm to diverge.
- Overestimating $\lambda_{\max}$ slows convergence.

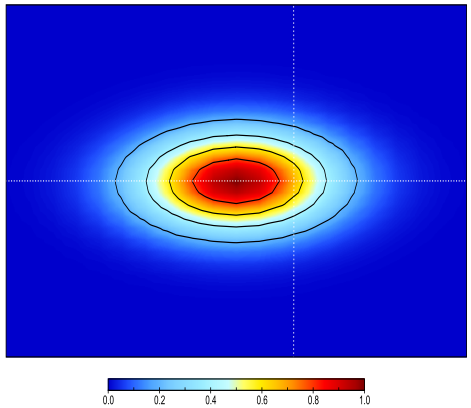


(From Weaver *et al.* 2016)

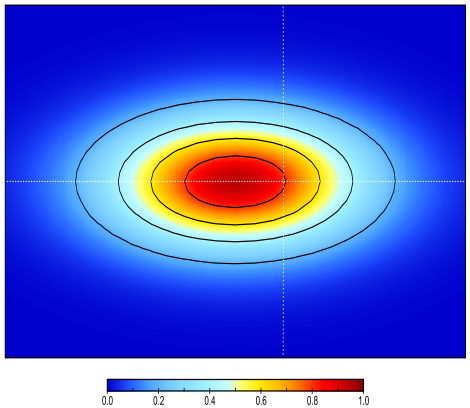
- We don't need a strict convergence criterion to get an adequate solution.



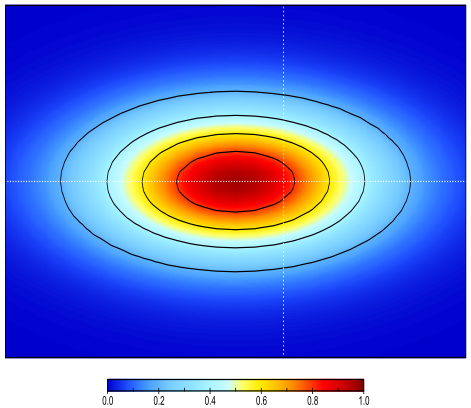
(c) $K = 2$ ($\epsilon_k = 0.3$)



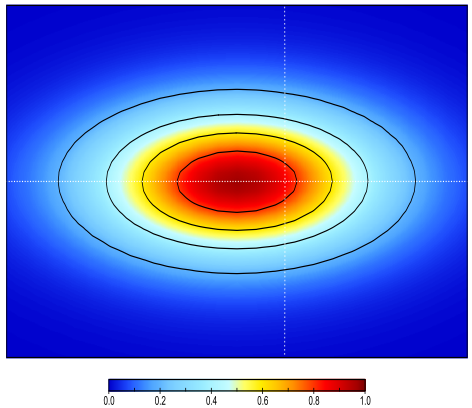
(d) $K = 4$ ($\epsilon_k = 10^{-1}$)



(e) $K = 9$ ($\epsilon_k = 10^{-2}$)



(f) $K = 13$ ($\epsilon_k = 10^{-3}$)



(g) $K = 43$ ($\epsilon_k = 10^{-10}$)

(From Weaver *et al.* 2016)

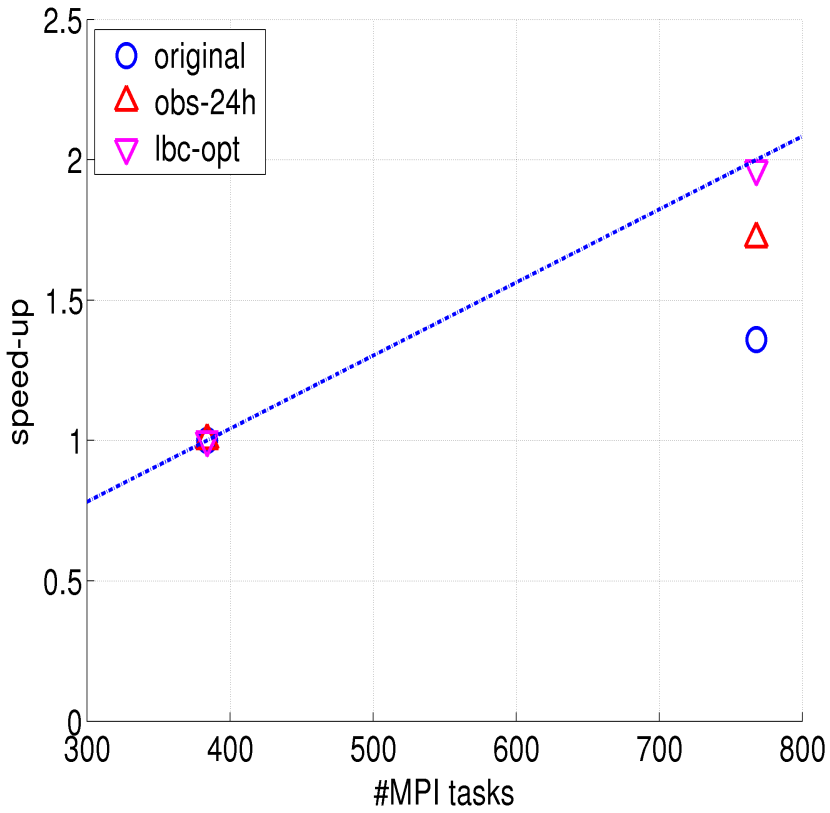
Initialization: Consider $\widehat{\mathbf{A}}\,\mathbf{x} = \mathbf{b}$ with $\mathbf{b}$ set to a random vector.

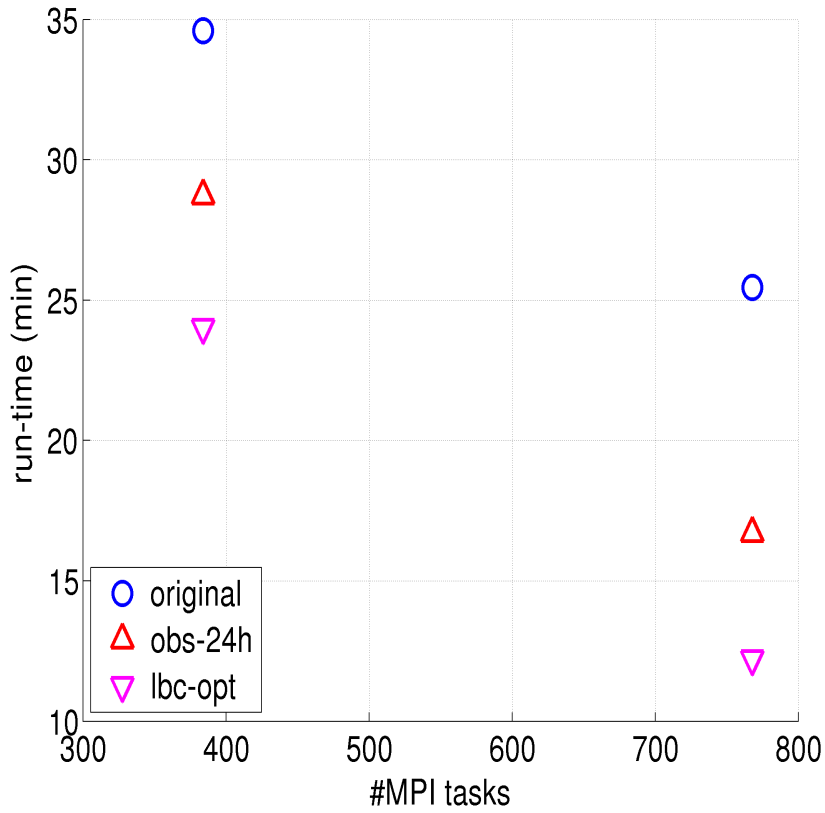
- **Step 1**: Use CG combined with a Lanczos procedure to estimate the eigenvalue bounds of $\widehat{\mathbf{A}}$.
- **Step 2**: Use CI with the eigenvalue bounds computed from Step 1 to diagnose the number of Chebyshev iterations needed to satisfy a desired solver precision.

Application within the correlation operator, $\mathbf{C} = \mathbf{U}\mathbf{U}^{\mathrm{T}}$:

- **Step 3**: Use CI (and its adjoint), with the eigenvalue bounds from Step 1 and the number of Chebyshev iterations fixed from Step 2.

NEMOVAR 3D-Var analysis, with full 3D diffusion-based correlation operator, for 1/4° global ocean model (NEMO ORCA025 configuration).





(Courtesy M. Chrust, ECMWF)

- Significant progress has been made in NEMOVAR:
  1. in making the diffusion-based correlation model more general and flexible;
  2. in improving computational aspects of the algorithm (Chebyshev iteration).
- Nevertheless, further improvements are needed, especially to reduce the computational cost for future applications with higher resolution global models ($1/12^\circ$) and a hybrid $\mathbf{B}$.
- Recall that the condition number of $\widehat{\mathbf{A}} \sim \kappa_i / e_i^2$:

  high-resolution => small e_i

  large localization scales => large κ_i

- Mixture of single and double precision. (M. Chrust)
- "Time"-parallel diffusion. (S. Gürol)
- Coarse-resolution grid and transfer operators for treating correlations with "large" length scales. (A. Vidard)
- Restricted Additive Schwarz preconditioner, with coarse grid solver. (M. Chrust)
- Improved algorithms for estimating normalization factors. (B. Ménétrier)

- Mixture of single and double precision. (M. Chrust)
- "Time"-parallel diffusion. (S. Gürol)
- Coarse-resolution grid and transfer operators for treating "large" length-scale correlations. (A. Vidard).
- Restricted Additive Schwarz preconditioner, with coarse grid solver. (M. Chrust)
- Improved algorithms for estimating normalization factors. (B. Ménétrier)

- Rewrite the sequence of M symmetric linear systems as a single nonsymmetric linear system:

$$\left.\begin{array}{rcl} \mathbf{A}\psi_1 &=& \psi_0 \\ \mathbf{A}\psi_2 &=& \psi_1 \\ &\vdots& \\ \mathbf{A}\psi_M &=& \psi_{M-1} \end{array}\right\} \Longrightarrow \left.\begin{array}{rcccl} & & & \mathbf{A}\psi_1 &= \psi_0 \\ -\psi_1 & + & & \mathbf{A}\psi_2 &= \mathbf{0} \\ & & & \vdots & \\ & & -\psi_{M-1} + & \mathbf{A}\psi_M &= \mathbf{0} \end{array}\right\}$$

- This has the form $\mathcal{A}\psi = \zeta$ where

$$\mathcal{A} = \begin{bmatrix} \mathbf{A} & & & \\ -\mathbf{I} & \mathbf{A} & & \\ & \ddots & \ddots & \\ & & -\mathbf{I} & \mathbf{A} \end{bmatrix}, \quad \psi = \begin{bmatrix} \psi_1 \\ \psi_2 \\ \vdots \\ \psi_M \end{bmatrix}, \quad \zeta = \begin{bmatrix} \psi_0 \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix}.$$

- $\mathcal{A}$ and $\mathbf{A}$ have the same eigenspectrum.
- We can solve the nonsymmetric system using the CI.
- The advantage of the $\mathcal{A}$ system is that the $\mathbf{A}$-matrix operators can be applied in parallel on each iteration of CI.

- Define "run-time cost" as the number of sequential **A**-matrix products.
- The red crosses and blue circles show the total number of **A**-matrix products required to achieve a residual reduction of 10^{-4} for the symmetric and nonsymmetric systems, for experiments with different values of M.
- The violet circles represent the potential gain from "time" parallelism.