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

A. Weaver\textsuperscript{1}, M. Chrust\textsuperscript{2}, S. Gürol\textsuperscript{1}, A. Piacentini\textsuperscript{1}, J. Tshimanga\textsuperscript{1}

\textsuperscript{1} CERFACS, Toulouse
\textsuperscript{2} ECMWF, Reading

December 13, 2016
1. Motivation and background

2. Implicit diffusion using polynomial-based iterative solution methods

3. Ongoing and future developments
1 Motivation and background

2 Implicit diffusion using polynomial-based iterative solution methods

3 Ongoing and future developments
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) $\mathbf{C}\psi$ for some vector $\psi$.

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 $\mathbf{C}\psi$ 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 $C\psi$ 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

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

(1)

where $M$ is a positive (preferably even) integer, and $\kappa$ 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 $\kappa$ the diffusion tensor.

- For constant $\kappa$, 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).
\[(1 - \nabla \cdot \kappa \nabla)^M \psi_M = \psi \] (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.
Implicit diffusion correlation functions on $S^2$

- Example of the shape and spectrum for different $M$ and constant $\kappa$.

**Correlation function**

**Variance spectrum**
Where do we need to apply a C-like operator?

Consider the NEMOVAR hybrid $B$ formulation:

$$B = \beta_m^2 \left( B_{m1} + B_{m2} + \ldots \right) + \beta_e^2 B_e$$

where $\beta_m^2$ and $\beta_e^2$ are constant weights.

Multiple covariance models for representing different scales:

$$B_{mi} = K_b D_i^{1/2} C_{mi} D_i^{1/2} K_T$$

A localized ensemble-based correlation matrix:

$$B_e = K_b D_e^{1/2} \left( L \circ \tilde{X} \tilde{X}^T \right) D_e^{1/2} K_b^T$$

where the columns of $\tilde{X} = D_e^{-1/2} K_b^{-1} X^b$ are transformed background ensemble perturbations.

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

$$\left( L \circ \tilde{X} \tilde{X}^T \right) v = \sum_{p=1}^{N_e} \left( \tilde{x}_p \circ L \left( \tilde{x}_p \circ v \right) \right) \text{ where } \tilde{X} = (\tilde{x}_1, \ldots, \tilde{x}_{N_e})$$
How to apply $C$?

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

- Let $A$ be the discretized, matrix representation of the self-adjoint operator $1 - \nabla \cdot \kappa \nabla$.

- The identity $C = (A^M)^{-1} = (A^{-1})^M$ suggests two ways of applying $C$.

  1. Solve the single linear, self-adjoint, positive-definite (SAPD) system
     $$A^M \psi_M = \psi$$
     (2)

  2. Solve the sequence of $M$ linear SAPD systems
     $$\begin{align*}
     A\psi_1 &= \psi \\
     A\psi_2 &= \psi_1 \\
     &\vdots \\
     A\psi_M &= \psi_{M-1}
     \end{align*}$$
     (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!
How to solve this large, sparse, linear SAPD system?

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

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

1 Motivation and background

2 Implicit diffusion using polynomial-based iterative solution methods

3 Ongoing and future developments
Polynomial-based iterative methods

- 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
  \[ \hat{A}x = b \]
  where \( \hat{A} \) is symmetric, positive definite (SPD).

- A **polynomial-based iterative method** produces, at iteration \( k \), an estimate \( x_k \) of the true solution \( x^* = \hat{A}^{-1}b \) such that the error is
  \[ x^* - x_k = \varphi_k(\hat{A})(x^* - x_0) \]
  where \( \varphi_k(\hat{A}) \) is a polynomial in \( \hat{A} \) of degree \( k \) with \( \varphi_k(0) = 1 \).

- This equation can also be expressed as
  \[ r_k = \varphi_k(\hat{A})r_0 \]
  where \( r_k = \hat{A}x_k - b \) is the residual vector.

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

\[ \mathbf{x}^* - \mathbf{x}_k = \phi_k(\hat{\mathbf{A}})(\mathbf{x}^* - \mathbf{x}_0), \]

**Conjugate gradients (CG)**

Choose the polynomial \( \phi_k = \phi_k^{CG} \) to be the unique solution of

\[
\min_{\phi(0) = 1, \deg(\phi) \leq k} \| \phi(\hat{\mathbf{A}})(\mathbf{x}^* - \mathbf{x}_0) \|_{\hat{\mathbf{A}}} \]

**Chebyshev iteration (CI)**

Choose the polynomial \( \phi_k = \phi_k^{CI} \) to be the unique solution of

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

where \( \lambda_{\min} \) and \( \lambda_{\max} \) are the extreme eigenvalues of \( \hat{\mathbf{A}} \). The resulting \( \phi_k^{CI} \) 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

$\Rightarrow$ may need re-orthogonalization of residual vectors.

Parameter-free method.

Computation of $\alpha_k$ and $\beta_k$ in CG involve inner products.

- Global MPI communications in a parallel domain decomposition.
  $\Rightarrow$ a well-known performance bottleneck on massively parallel machines.

- Nonlinear solver if stopped before full convergence
  $\Rightarrow$ the linearity and symmetry of $\mathbf{C}$ is no longer guaranteed.
Properties of CI

- 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 \( \hat{A} \).

=> these can be set to the extreme Ritz values \( \theta_{\text{min}} \) and \( \theta_{\text{max}} \) pre-computed with a combined CG/Lanczos procedure.

- Computation of \( \alpha_k \) and \( \beta_k \) depends on the input parameters.

  - **No inner products**
    => no global MPI communications are required.
    => local MPI communications are needed before each application of \( \hat{A} \).

  - **Strictly linear solver**
    => linearity of \( C \) is guaranteed even for a finite number of iterations.
    => exact symmetry of \( C \) can be enforced numerically using a factored formulation of \( C = UU^T \) and a fixed number of iterations (\( K \)).
Chebyshev Iteration

- $\sigma = (\theta_{\text{max}} + \theta_{\text{min}})/2$
- $\delta = (\theta_{\text{max}} - \theta_{\text{min}})/2$
- $\alpha_0 = 1/\sigma$ ; $\beta_1 = (\delta \alpha_0)^2/2$
- \textbf{for} $k = 1, \ldots, K - 1$ \textbf{do}
  - $\alpha_k = 1/(\sigma - \beta_k/\alpha_{k-1})$
  - $\beta_{k+1} = (\delta \alpha_k/2)^2$
- \textbf{end for}

- $x_0 = \text{initial estimate}$
- $r_0 = \hat{A}x_0 - b$
- $p_0 = -r_0$
- \textbf{for} $k = 0, \ldots, K - 1$ \textbf{do}
  - $q_k = \hat{A}p_k$
  - $x_{k+1} = x_k + \alpha_k p_k$
  - $r_{k+1} = r_k + \alpha_k q_k$
  - $p_{k+1} = -r_{k+1} + \beta_{k+1} p_k$
- \textbf{end for}
Convergence properties of CI as a function of $M$ and $\alpha$.

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

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

$$\lambda_{\text{min}} \approx 1 \quad \text{and} \quad \lambda_{\text{max}} \approx 1 + \left( \frac{\kappa_1}{e_1^2} + \frac{\kappa_2}{e_2^2} \right) \approx 1 + \frac{4 \alpha^2}{M - 2}$$
Examples from a global-ocean VDA system

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

(From Weaver et al. 2016)
What happens if $\lambda_{\text{max}}$ is misspecified?

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

$$\gamma = \theta_{\text{max}} / \lambda_{\text{max}}$$

(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)
A strategy for applying CI in (Ensemble) VDA

*Initialization*: Consider $\hat{A}x = b$ with $b$ set to a random vector.

- **Step 1**: Use CG combined with a Lanczos procedure to estimate the eigenvalue bounds of $\hat{A}$.

- **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, $C = UU^T$*:

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

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

(Courtesy M. Chrust, ECMWF)
Outline

1 Motivation and background

2 Implicit diffusion using polynomial-based iterative solution methods

3 Ongoing and future developments
Ongoing and future developments

• 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°) and a hybrid $B$.

• Recall that the condition number of $\hat{A} \sim \kappa_i / e_i^2$:
  high-resolution $\Rightarrow$ small $e_i$
  large localization scales $\Rightarrow$ large $\kappa_i$
Ongoing and future developments

- 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)
Ongoing and future developments

- 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)
A “time”-parallel variant of implicit diffusion

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

$$
\begin{align*}
A\psi_1 &= \psi_0 \\
A\psi_2 &= \psi_1 \\
& \quad \vdots \\
A\psi_M &= \psi_{M-1}
\end{align*}
\quad \implies 
\begin{align*}
-\psi_1 + A\psi_2 &= 0 \\
& \quad \vdots \\
-\psi_{M-1} + A\psi_M &= 0
\end{align*}
$$

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

$$
\mathcal{A} = \begin{bmatrix} A & \cdots & A \\
-1 & A & \cdots \\
& \ddots & \ddots & \ddots \\
& & -1 & A
\end{bmatrix}, \quad \psi = \begin{bmatrix} \psi_1 \\
\psi_2 \\
\vdots \\
\psi_M
\end{bmatrix}, \quad \zeta = \begin{bmatrix} \psi_0 \\
0 \\
\vdots \\
0
\end{bmatrix}.
$$

- $\mathcal{A}$ and $A$ have the same eigenspectrum.

- We can solve the nonsymmetric system using the CI.

- The advantage of the $\mathcal{A}$ system is that the $A$-matrix operators can be applied in parallel on each iteration of CI.
A “time”-parallel variant of implicit diffusion

- 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.