Seminar of the PhD in Mathematical Engineering
Universidad EAFIT

Background Error Estimation In Sequential Data Assimilation

Elias D. Niño-Ruiz

Applied Math and Computer Science Laboratory (AML-CS)
Department of Computer Science
Universidad del Norte
BAQ 080001, Colombia

Outline I

Motivation I

- Weather forecasts and warnings are the most important services provided by the meteorological profession.
- Forecasts are used by
  - Government and industry to protect life and property.
  - To improve the efficiency of operations.
  - Individuals to plan a wide range of daily activities.
- Weather forecasting today is a highly developed skill:
  - It is grounded in scientific principles and methods.
  - Makes use of advanced technological tools.
- **How do we forecast the state of (highly non-linear) dynamical system?**
  - An imperfect numerical forecast.
  - Observations of the actual state.
  - Observation operator.

Components in DA [BS12] I

- We want to estimate $\mathbf{x}^* \in \mathbb{R}^{n \times 1}$. $n \sim \mathcal{O}\left(10^8\right)$.
- Imperfect numerical model:

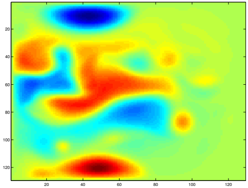
$$\mathbf{x}_{\text{next}} = \mathcal{M}_{t_{\text{current}} \rightarrow t_{\text{next}}}\left(\mathbf{x}_{\text{current}}\right),$$

  where $\mathbf{x} \in \mathbb{R}^{n \times 1}$.
- Noisy observations:

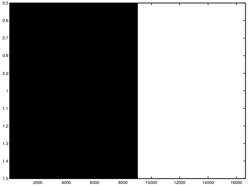
$$\mathbf{y} = \mathcal{H}\left(\mathbf{x}\right) + \boldsymbol{\epsilon} \in \mathbb{R}^{m \times 1},$$

  where $\mathcal{H} : \mathbb{R}^n \rightarrow \mathbb{R}^m$ and $\boldsymbol{\epsilon} \sim \mathcal{N}\left(\mathbf{0}_m, \mathbf{R}\right)$.$m \sim \mathcal{O}\left(10^6\right)$.
- Prior estimate $\mathbf{x}^b \in \mathbb{R}^{n \times 1}$ with errors following $\mathcal{N}\left(\mathbf{0}, \mathbf{B}\right)$.

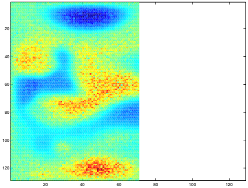
Components in DA [BS12] II



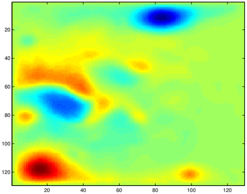
(a) $\mathbf{x}^*$



(b) $\mathbf{H}$



(c) $\mathbf{y} = \mathbf{H} \cdot \mathbf{x}^* + \epsilon$



(d) $\mathbf{x}^b$

Components in DA [BS12] III

- By Bayes' Theorem we know that:

$$\mathcal{P}(\mathbf{x}|\mathbf{y}) \propto \mathcal{P}(\mathbf{x}) \cdot \mathcal{L}(\mathbf{x}|\mathbf{y})$$

where

$$\begin{aligned} \mathcal{P}(\mathbf{x}) &\propto \exp\left(-\frac{1}{2} \cdot \left\| \mathbf{x} - \mathbf{x}^b \right\|_{\mathbf{B}^{-1}}^2\right) \\ \mathcal{L}(\mathbf{x}|\mathbf{y}) &\propto \exp\left(-\frac{1}{2} \cdot \| \mathbf{y} - \mathbf{H} \cdot \mathbf{x} \|_{\mathbf{R}^{-1}}^2\right) \end{aligned}$$

and therefore,

$$\mathbf{x}^a = \arg \max_{\mathbf{x}} \mathcal{P}(\mathbf{x}|\mathbf{y}) \, ,$$

Components in DA [BS12] IV

- It can be easily shown that:

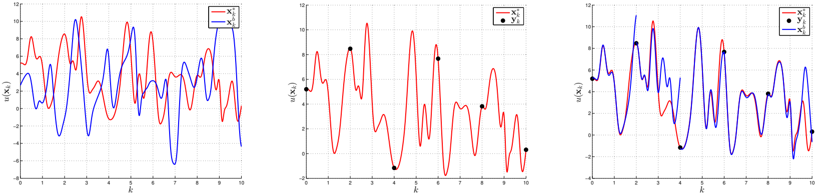
$$\begin{aligned}\mathbf{x}^a &= \mathbf{x}^b + \mathbf{A} \cdot \mathbf{H}^T \cdot \mathbf{R}^{-1} \cdot \mathbf{d} = \mathbf{A} \cdot \left[\mathbf{B}^{-1} \cdot \mathbf{x}^b + \mathbf{H}^T \cdot \mathbf{R}^{-1} \cdot \mathbf{y}\right] \\ &= \mathbf{x}^b + \mathbf{B} \cdot \mathbf{H}^T \cdot \left[\mathbf{R} + \mathbf{H} \cdot \mathbf{B} \cdot \mathbf{H}^T\right]^{-1} \cdot \mathbf{d}\end{aligned}$$

  where $\mathbf{A} = \left[\mathbf{B}^{-1} + \mathbf{H}^T \cdot \mathbf{R}^{-1} \cdot \mathbf{H}\right]^{-1} \in \mathbb{R}^{n \times n}$, and $\mathbf{d} = \mathbf{y} - \mathbf{H} \cdot \mathbf{x}^b \in \mathbb{R}^{m \times 1}$.

- Posterior distribution:

$$\mathbf{x} \sim \mathcal{N}\left(\mathbf{x}^a,\, \mathbf{A}\right).$$

Sequential Data Assimilation Problem



(a) Free run (b) Observations (c) Sequential DA process

Figure: Sequential Data Assimilation process.

At assimilation steps, we do need to estimate $\mathbf{x}^b$ and B (moments of the prior error distribution).

Ensemble Based Methods

- We can make use of an ensemble of model realizations:

$$\mathbf{X}^b = \left[\mathbf{x}^{b[1]}, \mathbf{x}^{b[2]}, \ldots, \mathbf{x}^{b[N]}\right] \in \mathbb{R}^{n \times N}$$

- Empirical moments of the ensemble:

$$\begin{aligned} \mathbf{x}^b \approx \overline{\mathbf{x}}^b &= \frac{1}{N} \cdot \mathbf{X}^b \cdot \mathbf{1}_N \in \mathbb{R}^{n \times n}\,, \\ \mathbf{B} \approx \mathbf{P}^b &= \frac{1}{N-1} \cdot \delta\mathbf{X} \cdot \delta\mathbf{X}^T\,, \end{aligned}$$

and $\delta\mathbf{X} = \mathbf{X}^b - \overline{\mathbf{x}}^b \cdot \mathbf{1}_N^T \in \mathbb{R}^{n \times N}$.

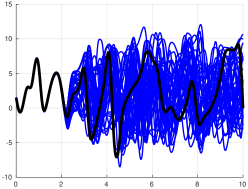
The Lorenz 96 Model - Toy Model I

- The Lorenz 96 model:

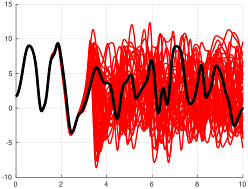
$$\frac{dx_j}{dt} = \begin{cases} (x_2 - x_{n-1}) \cdot x_n - x_1 + F & \text{for } i = 1, \\ (x_{i+1} - x_{i-2}) \cdot x_{i-1} - x_i + F & \text{for } 2 \leq i \leq n-1, \\ (x_1 - x_{n-2}) \cdot x_{n-1} - x_n + F & \text{for } i = n, \end{cases} \quad (1)$$

  where x_i stands for the i-th model component, for $1 \leq i \leq n$.
- Each model component stands for a particle which fluctuates in the atmosphere.
- Exhibits chaotic behaviour when the external force F is set to 8.

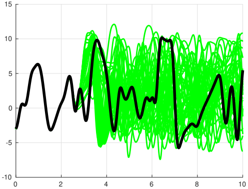
The Lorenz 96 Model - Toy Model II



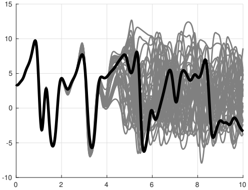
(a) x_5



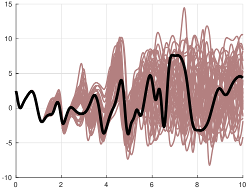
(b) x_{10}



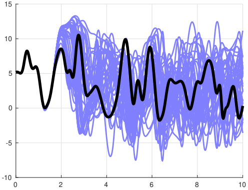
(c) x_{20}



(d) x_{30}

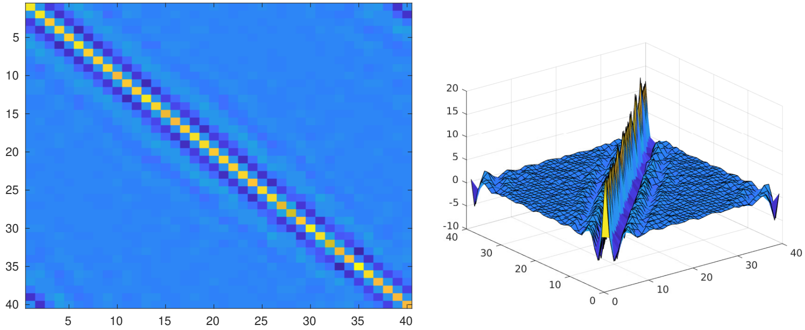


(e) x_{35}



(f) x_{40}

Estimation of **B** via $N = 10^5$.



(a) Structure (b) Surf

Figure: Estimation of **B** via $N = 10^5$.

The Stochastic Ensemble Kalman Filter [Eve06] I

- Sequential Monte Carlo method for parameter and state estimation.
- Analysis ensemble (posterior ensemble):

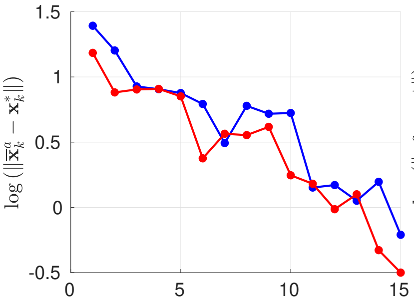
$$
\begin{aligned}
\mathbf{X}^a &= \mathbf{X}^b + \mathbf{P}^b \cdot \mathbf{H}^T \cdot \left[\mathbf{R} + \mathbf{H} \cdot \mathbf{P}^b \cdot \mathbf{H}\right] \cdot \mathbf{D} \\
\mathbf{X}^a &= \mathbf{X}^b + \mathbf{P}^a \cdot \mathbf{H}^T \cdot \mathbf{R}^{-1}\mathbf{D} \in \mathbb{R}^{n \times N}, \\
\mathbf{X}^a &= \mathbf{P}^a \cdot \left[\mathbf{H}^T \cdot \mathbf{R}^{-1} \cdot \mathbf{Y}^s + \left[\mathbf{P}^b\right]^{-1} \cdot \mathbf{X}^b\right] \in \mathbb{R}^{n \times N},
\end{aligned}
$$

where $\mathbf{P}^a = \left[\mathbf{H}^T \cdot \mathbf{R}^{-1} \cdot \mathbf{H} + \left[\mathbf{P}^b\right]^{-1}\right] \in \mathbb{R}^{n \times n}$, and the e-th column of $\mathbf{D} \in \mathbb{R}^{m \times N}$ and $\mathbf{Y}^s \in \mathbb{R}^{n \times N}$ are:

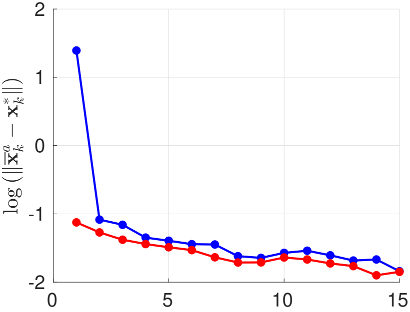
$$
\mathbf{d}^{[e]} = \mathbf{y} + \boldsymbol{\epsilon}^{[e]} - \mathcal{H}\left(\mathbf{x}^{b[e]}\right) \in \mathbb{R}^{m \times 1}, \text{ and } \mathbf{y}^{s[e]} = \mathbf{y} + \boldsymbol{\epsilon}^{[e]},
$$

respectively, for $1 \le e \le N$, and $\boldsymbol{\epsilon}^{[e]} \sim \mathcal{N}\left(\mathbf{0}_m, \mathbf{R}\right)$.

$L-2$ Error Norms in Time, $N = 10^5$



(a) $p = 50\%$

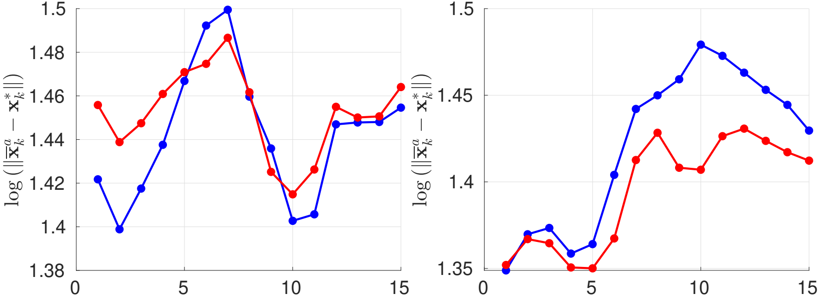


(b) $p = 100\%$

Figure: $L-2$ error norms in time, $N = 10^5$.

But too many samples!!! In practice, model realizations are constrained by the hundreds...

$L-2$ error norms in time, $N=10$

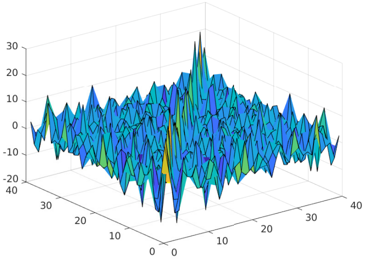


(a) $p = 50\%$ (b) $p = 100\%$

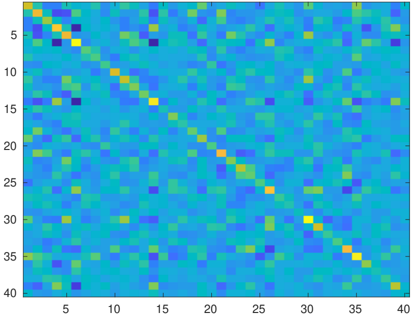
Figure: $L-2$ error norms in time, $N=10$.

What is going on here?...

Estimation of $\mathbf{B}$ via $N = 10$



(a) Structure



(b) Surf

Figure: Estimation of $\mathbf{B}$ via $N = 10$.

What can we do? Localization methods...

Localization Methods

- Avoid the impact of spurious correlations.
- If

$$\frac{\log(n)}{N}$$

  is bounded (and small)... the resulting estimator is well-conditioned.
- Three different flavors:
  1. Covariance Matrix Localization. (Precision Localization) [NRSD15, NRSD17, NR17, NRSD18].
  2. Spatial Domain Localization [OHS$^+$04].
  3. Observation Localization [AND07, AND09].

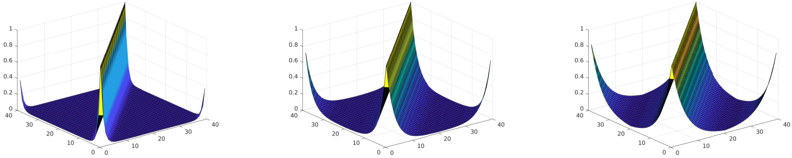
Covariance Matrix Localization

- Impose the desired structure on $\mathbf{P}^b$ via a decorrelation matrix.

$$\widehat{\mathbf{P}} = \mathbf{L} \otimes \mathbf{P}^b \,, \tag{2}$$

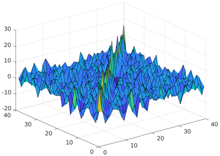
where, for instance,

$$\{\mathbf{L}\}_{i,j} = \exp\left(-\frac{\phi(i,j)^2}{r^2}\right) \,.$$

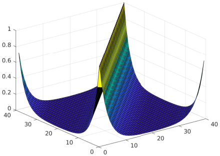


(a) $r = 1$ (b) $r = 3$ (c) $r = 5$

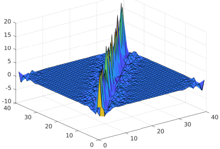
Effects of Covariance Matrix Localization



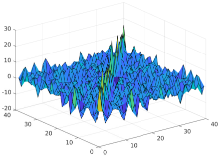
(a) $\mathbf{P}^b$, $N = 30$



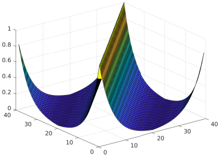
(b) $\mathbf{L}$ for $r = 3$



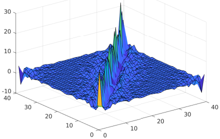
(c) $\widehat{\mathbf{P}} = \mathbf{L} \cdot \mathbf{P}^b$



(d) $\mathbf{P}^b$, $N = 30$

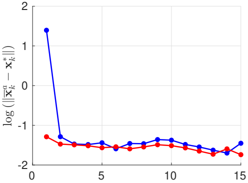


(e) $\mathbf{L}$ for $r = 5$

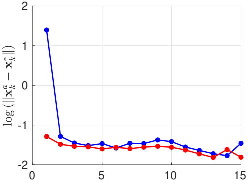


(f) $\widehat{\mathbf{P}} = \mathbf{L} \cdot \mathbf{P}^b$

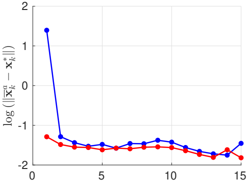
$L - 2$ error norms in time.



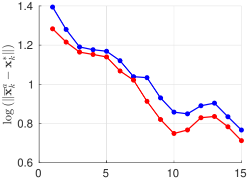
(a) $N = 30,\ r = 1,$
$p = 100\%$



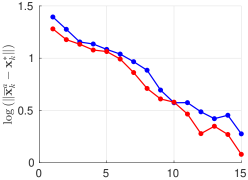
(b) $N = 30,\ r = 3,$
$p = 100\%$



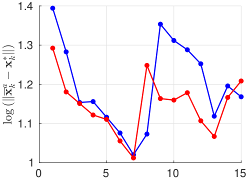
(c) $N = 30,\ r = 5,$
$p = 100\%$



(d) $N = 30,\ r = 1,$
$p = 50\%$



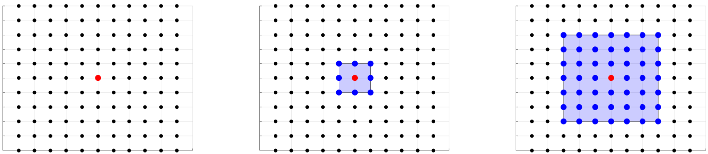
(e) $N = 30,\ r = 3,$
$p = 50\%$



(f) $N = 30,\ r = 5,$
$p = 50\%$

Precision Matrix Localization I

- Component-wise products are prohibitive in high-dimensional spaces.
- *When two model components are conditional independent, their corresponding entry in the precision covariance matrix is zero*.



(a) $r = 0$ (b) $r = 1$ (c) $r = 3$

Precision Matrix Localization II

- Modified Cholesky Decomposition:

$$\widehat{\mathbf{B}}^{-1} = \mathbf{T}^{T} \cdot \mathbf{D}^{-1} \cdot \mathbf{T}$$

where the non-zero elements from $\mathbf{T} \in \mathbb{R}^{n \times n}$ are given by fitting models of the form:

$$\mathbf{x}^{[i]} = \sum_{q \in P(i,r)} \mathbf{x}^{[q]} \cdot \{-\mathbf{T}\}_{i,q} + \boldsymbol{\epsilon}^{[i]} \in \mathbb{R}^{N \times 1}, \text{ for } 1 \leq i \leq n\,,$$

and $\{\mathbf{D}\}_{i,i} = \mathbf{var}\left(\boldsymbol{\epsilon}^{[i]}\right)$.

1	5	9	13
2	6	10	14
3	7	11	15
4	8	12	16

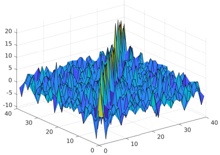
(a) $N(6,1)$

1	5	9	13
2	6	10	14
3	7	11	15
4	8	12	16

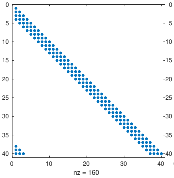
(b) $P(6,1)$

Precision Matrix Localization III

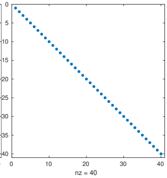
- An estimate:



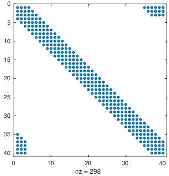
(a) $\mathbf{P}^b$



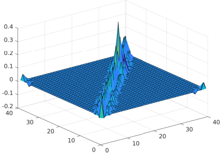
(b) $\mathbf{T}$



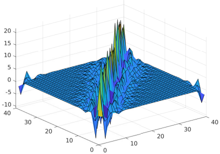
(c) $\mathbf{D}$



(d) $\widehat{\mathbf{B}}^{-1}$ Str



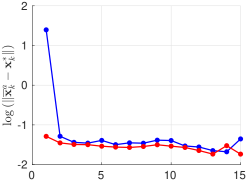
(e) $\widehat{\mathbf{B}}^{-1}$



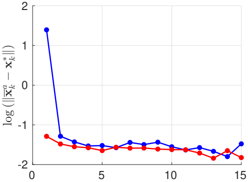
(f) $\widehat{\mathbf{B}}$

Results:

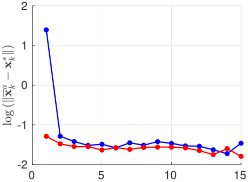
Precision Matrix Localization IV



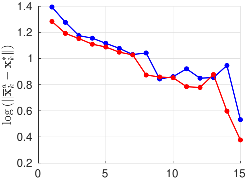
(a) $N = 30,\ r = 1,\ p = 100\%$



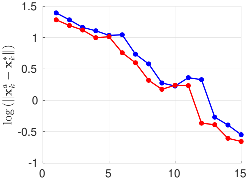
(b) $N = 30,\ r = 3,\ p = 100\%$



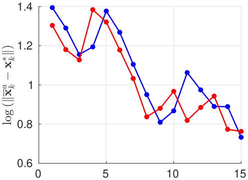
(c) $N = 30,\ r = 5,\ p = 100\%$



(d) $N = 30,\ r = 1,\ p = 50\%$



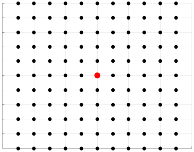
(e) $N = 30,\ r = 3,\ p = 50\%$



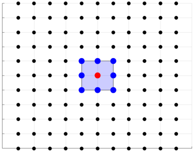
(f) $N = 30,\ r = 5,\ p = 50\%$

Spatial Domain Localization [Bue11] I

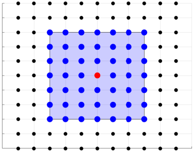
- Very simple idea:



(a) $r = 0$



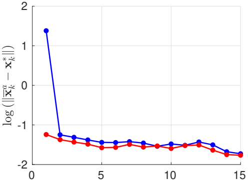
(b) $r = 1$



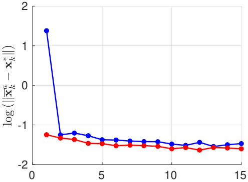
(c) $r = 3$

- Then...
  1. Use local observations.
  2. Use local estimators of covariance matrices.
  3. Hybrid methods work very well.
  4. Evidently, we mitigate the impact of sampling errors...

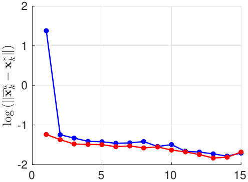
Spatial Domain Localization [Bue11] II



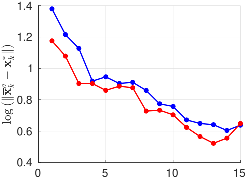
(a) $N = 30,\ r = 1,$ $p = 100\%$



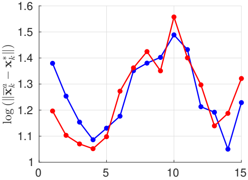
(b) $N = 30,\ r = 3,$ $p = 100\%$



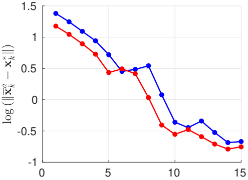
(c) $N = 30,\ r = 5,$ $p = 100\%$



(d) $N = 30,\ r = 1,$ $p = 50\%$



(e) $N = 30,\ r = 3,$ $p = 50\%$



(f) $N = 30,\ r = 5,$ $p = 50\%$

Shrinkage Covariance Matrix Estimation I

- Samples $\{s_i\}_{i=1}^{N}$, where $s_i \sim \mathcal{N}(\mathbf{0}_n, \mathbf{C})$
- Structure of matrices:

$$\widehat{\mathbf{C}} = \gamma \cdot \mathbf{T} + (1-\gamma) \cdot \mathbf{C}_s \in \mathbb{R}^{n \times n},$$

  optimal value of γ in squared loss sense $\mathbb{E}\left[\left\|\widehat{\mathbf{C}} - \mathbf{C}\right\|_F^2\right]$ where $\mathbf{C} \in \mathbb{R}^{n \times n}$ is the true covariance matrix. $\mathbf{T} = \frac{\mathrm{tr}(\mathbf{C}_s)}{n} \cdot \mathbf{I}$.
- Properties:
  - Have been proven more accurate than the sample covariance matrix [CM14].
  - Better conditioned than the true covariance matrix [CWEH10].
  - They are strong under the condition $n \gg N$ [CWH11].

Shrinkage Covariance Matrix Estimation II

- **Ledoit and Wolf estimator [LW04, CWEH10]**:

$$\gamma_{LW} = \min \left(\frac{\sum_{i=1}^{N} \left\| \mathbf{C}_s - s_i \otimes s_i^T \right\|_F^2}{N^2 \cdot \left[\mathrm{tr}\left(\mathbf{C}_s^2\right) - \frac{\mathrm{tr}^2(\mathbf{C}_s)}{n} \right]}, \, 1 \right)$$

- **Rao-Blackwell Ledoit and Wolf estimator [CWEH10]**:

$$\gamma_{RBLW} = \min \left(\frac{\frac{N-2}{n} \cdot \mathrm{tr}\left(\mathbf{C}_s^2\right) + \mathrm{tr}^2\left(\mathbf{C}_s\right)}{(N+2) \cdot \left[\mathrm{tr}\left(\mathbf{C}_s^2\right) - \frac{\mathrm{tr}^2(\mathbf{C}_s)}{n} \right]}, \, 1 \right)$$

- It is proven that [CWH11]:

$$\mathbb{E}\left[\left\| \widehat{\mathbf{C}}_{RBLW} - \mathbf{C} \right\|_F^2 \right] \leq \mathbb{E}\left[\left\| \widehat{\mathbf{C}}_{LW} - \mathbf{C} \right\|_F^2 \right] .$$

RBLW in the EnKF context

- **Replace $\mathbf{P}^b$ by a better estimator of B.**
- RBLW estimator in the EnKF context:

$$\widehat{\mathbf{B}} = \gamma_{\widehat{\mathbf{B}}} \cdot \left[\mu_{\widehat{\mathbf{B}}} \cdot \mathbf{I}_{n \times n}\right] + \left(1 - \gamma_{\widehat{\mathbf{B}}}\right) \cdot \widehat{\boldsymbol{\delta} \mathbf{X}} \cdot \widehat{\boldsymbol{\delta} \mathbf{X}}^T \in \mathbb{R}^{n \times n}.$$

where $\widehat{\boldsymbol{\delta} \mathbf{X}} = \frac{1}{\sqrt{N-1}} \cdot \boldsymbol{\delta} \mathbf{X} \in \mathbb{R}^{n \times N}$.

- Parameters:

$$\mu_{\widehat{\mathbf{B}}} = \frac{\mathrm{tr}\left(\mathbf{P}^b\right)}{n}$$

$$\gamma_{\widehat{\mathbf{B}}} = \min\left(\frac{\frac{N-2}{n} \cdot \mathrm{tr}\left(\mathbf{P}^{b^2}\right) + \mathrm{tr}^2\left(\mathbf{P}^b\right)}{(N+2) \cdot \left[\mathrm{tr}\left(\mathbf{P}^{b^2}\right) - \frac{\mathrm{tr}^2\left(\mathbf{P}^b\right)}{n}\right]}, 1\right)$$

- **The direct implementation is prohibitive, recall $n \sim \mathcal{O}\left(10^8\right)$.**

Efficient Implementation of the RBLW I

- Recall:

$$\begin{aligned} \mathrm{tr}\left(\mathbf{P}^{b}\right) &= \sum_{i=1}^{n} \sigma_i = \sum_{i=1}^{N-1} \sigma_i \\ \mathrm{tr}\left(\mathbf{P}^{b^2}\right) &= \sum_{i=1}^{n} \sigma_i^2 = \sum_{i=1}^{N-1} \sigma_i^2 \,. \end{aligned}$$

- Note

$$\begin{aligned} \mathbf{P}^{b} &= \widehat{\delta\mathbf{X}} \cdot \widehat{\delta\mathbf{X}}^{T} = \left[\mathbf{U}_{\widehat{\delta\mathbf{X}}} \cdot \widehat{\mathbf{\Sigma}}_{\widehat{\delta\mathbf{X}}} \cdot \mathbf{V}_{\widehat{\delta\mathbf{X}}}^{T}\right] \cdot \left[\mathbf{U}_{\widehat{\delta\mathbf{X}}} \cdot \widehat{\mathbf{\Sigma}}_{\widehat{\delta\mathbf{X}}} \cdot \mathbf{V}_{\widehat{\delta\mathbf{X}}}^{T}\right]^{T} \\ &= \mathbf{U}_{\widehat{\delta\mathbf{X}}} \cdot \widehat{\mathbf{\Sigma}}_{\widehat{\delta\mathbf{X}}}^{2} \cdot \mathbf{U}_{\widehat{\delta\mathbf{X}}}^{T} \end{aligned}$$

Efficient Implementation of the RBLW II

this implies

$$\sigma_i\left(\mathbf{P}^b\right) = \widehat{\sigma}_i^{\,2}\left(\widehat{\boldsymbol{\delta}\mathbf{X}}\right),$$

for $1 \leq i \leq N-1$.

- The estimator reads:

$$\widehat{\mathbf{B}} = \gamma_{\widehat{\mathbf{B}}} \cdot \left[\mu_{\widehat{\mathbf{B}}} \cdot \mathbf{I}_{n\times n}\right] + \left(1-\gamma_{\widehat{\mathbf{B}}}\right) \cdot \widehat{\boldsymbol{\delta}\mathbf{X}} \cdot \widehat{\boldsymbol{\delta}\mathbf{X}}^T \in \mathbb{R}^{n\times n}.$$

- Efficient computation of the parameters:

$$\mu_{\widehat{\mathbf{B}}} = \frac{\sum_{i=1}^{N-1} \widehat{\sigma}_i^{\,2}}{n},$$

$$\gamma_{\widehat{\mathbf{B}}} = \min\left(\frac{\frac{N-2}{n} \cdot \sum_{i=1}^{N-1} \widehat{\sigma}_i^{\,4} + \left[\sum_{i=1}^{N-1} \widehat{\sigma}_i^{\,2}\right]^2}{(N+2) \cdot \left[\sum_{i=1}^{N-1} \widehat{\sigma}_i^{\,4} - \frac{\left[\sum_{i=1}^{N-1} \widehat{\sigma}_i^{\,2}\right]^2}{n}\right]}, 1\right).$$

Efficient Implementation of the RBLW III

- $\widehat{\sigma}_i$ is the i-th singular value of $\widehat{\boldsymbol{\delta}\mathbf{X}} \in \mathbb{R}^{n \times N}$, for $1 \leq i \leq N-1$.

k

- EnKF model space, with $\varphi = \mu_{\widehat{\mathbf{B}}} \cdot \gamma_{\widehat{\mathbf{B}}}$ and $\delta = 1 - \gamma_{\widehat{\mathbf{B}}}$:

$$\mathbf{X}^a = \mathbf{X}^b + \mathbf{E} \cdot \mathbf{\Pi} \cdot \mathbf{Z}_{\widehat{\mathbf{B}}} + \varphi \cdot \mathbf{H}^T \cdot \mathbf{Z}_{\widehat{\mathbf{B}}},$$

  where $\mathbf{E} = \sqrt{\delta} \cdot \widehat{\boldsymbol{\delta}\mathbf{X}} \in \mathbb{R}^{n \times N}$, $\mathbf{\Pi} = \mathbf{H} \cdot \mathbf{E} \in \mathbb{R}^{m \times N}$, and $\mathbf{Z}_{\widehat{\mathbf{B}}} \in \mathbb{R}^{m \times N}$:

$$\left(\mathbf{\Gamma} + \mathbf{\Pi} \cdot \mathbf{\Pi}^T\right) \cdot \mathbf{Z}_{\widehat{\mathbf{B}}} = \left[\mathbf{Y} - \mathcal{H}\left(\mathbf{X}^b\right)\right],$$

$$\mathbf{\Gamma} = \mathbf{R} + \varphi \cdot \mathbf{H} \cdot \mathbf{H}^T \in \mathbb{R}^{m \times m}.$$

- EnKF ensemble space:

$$\mathbf{X}^a = \mathbf{X}^b + \mathbf{U} \cdot \boldsymbol{\lambda}^* \in \mathbb{R}^{n \times N}.$$

  where $\mathbf{U} = \sqrt{N-1} \cdot \widehat{\boldsymbol{\delta}\mathbf{X}} \in \mathbb{R}^{n \times N}$ and $\boldsymbol{\lambda}^* \in \mathbb{R}^{N \times N}$ minimizes

$$\mathcal{J}_{\text{ens}}(\boldsymbol{\lambda}) = \frac{1}{2} \cdot \|\mathbf{U} \cdot \boldsymbol{\lambda}\|^2_{\widehat{\mathbf{B}}^{-1}} + \frac{1}{2} \cdot \left\|\mathbf{Y} - \mathcal{H}\left(\mathbf{X}^b\right) - \mathbf{Q} \cdot \boldsymbol{\lambda}\right\|^2_{\mathbf{R}^{-1}}$$

  with $\mathbf{Q} = \mathbf{H} \cdot \mathbf{U} \in \mathbb{R}^{m \times N}$.

Synthetic Members

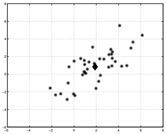
- The size of the ensemble can be increased by synthetic members:

$$\mathbf{x}_i^s \sim \mathcal{N}\left(\overline{\mathbf{x}}^b,\, \widehat{\mathbf{B}}\right),\ \text{for } 1 \leq i \leq K.$$

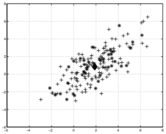
- Sampling from the above distribution does not require to build $\widehat{\mathbf{B}}$, instead:

$$\widehat{\mathbf{B}} \equiv \left[\widehat{\delta\mathbf{X}}, \mu_{\widehat{\mathbf{B}}},\, \gamma_{\widehat{\mathbf{B}}}\right]$$

- Prior distributions:



(a) $K = 0$



(b) $K = 120$

Sampling in High Dimensions I

- Taking the samples

$$\mathbf{x}_i^b = \overline{\mathbf{x}}^b + \widehat{\mathbf{B}}^{1/2} \cdot \boldsymbol{\xi}_i = \overline{\mathbf{x}}^b + \left(\varphi \cdot \mathbf{I}_{n \times n} + \delta \cdot \widehat{\boldsymbol{\delta} \mathbf{X}} \cdot \widehat{\boldsymbol{\delta} \mathbf{X}}^T \right)^{1/2} \cdot \boldsymbol{\xi}_i$$

  where $\boldsymbol{\xi}_i \sim \mathcal{N}\left(\mathbf{0}_n,\, \mathbf{I}_{n \times n}\right)$, $\varphi = \mu_{\widehat{\mathbf{B}}} \cdot \gamma_{\widehat{\mathbf{B}}}$ and $\delta = 1 - \gamma_{\widehat{\mathbf{B}}}$.

- Consider the random vectors

$$\begin{aligned} \boldsymbol{\xi}_i^1 &\sim \mathcal{N}\left(\mathbf{0}_n,\, \mathbf{I}_{n \times n}\right) \in \mathbb{R}^{n \times 1}\,, \\ \boldsymbol{\xi}_i^2 &\sim \mathcal{N}\left(\mathbf{0}_N,\, \mathbf{I}_{N \times N}\right) \in \mathbb{R}^{N \times 1}\,, \end{aligned}$$

  and let

$$\begin{aligned} \mathrm{Cov}\left(\boldsymbol{\xi}_i^1,\, \boldsymbol{\xi}_i^2\right) &= \boldsymbol{\xi}_i^1 \otimes {\boldsymbol{\xi}_i^2}^T = \mathbf{0}_{n \times N}\,, \\ \mathrm{Cov}\left(\boldsymbol{\xi}_2,\, \boldsymbol{\xi}_1\right) &= \boldsymbol{\xi}_i^2 \otimes {\boldsymbol{\xi}_i^1}^T = \mathbf{0}_{N \times n}\,. \end{aligned}$$

Sampling in High Dimensions II

We make the following substitution:

$$\widehat{\mathbf{B}}^{1/2} \cdot \boldsymbol{\xi}_i \sim \sqrt{\varphi} \cdot \boldsymbol{\xi}_i^1 + \sqrt{\delta} \cdot \widehat{\boldsymbol{\delta}\mathbf{X}} \cdot \boldsymbol{\xi}_i^2 .$$

Sampling in High Dimensions III

- The statistics are not changed:

$$\mathbb{E}\left[\left(\sqrt{\varphi}\cdot\boldsymbol{\xi}_i^1+\sqrt{\delta}\cdot\widehat{\boldsymbol{\delta}\mathbf{X}}\cdot\boldsymbol{\xi}_i^2\right)\left(\sqrt{\varphi}\cdot\boldsymbol{\xi}_i^1+\sqrt{\delta}\cdot\widehat{\boldsymbol{\delta}\mathbf{X}}\cdot\boldsymbol{\xi}_i^2\right)^T\right]$$

$$=\varphi\cdot\underbrace{\boldsymbol{\xi}_i^1\otimes\boldsymbol{\xi}_i^{1\,T}}_{\text{Cov}\left(\boldsymbol{\xi}_i^1,\boldsymbol{\xi}_i^1\right)=\mathbf{I}_{n\times n}}+\sqrt{\varphi\cdot\delta}\cdot\underbrace{\boldsymbol{\xi}_i^1\otimes\boldsymbol{\xi}_i^{2\,T}}_{\text{Cov}\left(\boldsymbol{\xi}_i^1,\boldsymbol{\xi}_i^2\right)=\mathbf{0}_{n\times N}}$$

$$+\sqrt{\varphi\cdot\delta}\cdot\underbrace{\boldsymbol{\xi}_i^2\otimes\boldsymbol{\xi}_i^{1\,T}}_{\text{Cov}\left(\boldsymbol{\xi}_i^2,\boldsymbol{\xi}_i^1\right)=\mathbf{0}_{N\times n}}$$

$$+\delta\cdot\widehat{\boldsymbol{\delta}\mathbf{X}}\cdot\underbrace{\boldsymbol{\xi}_i^2\otimes\boldsymbol{\xi}_i^{2\,T}}_{\text{Cov}\left(\boldsymbol{\xi}_i^2,\boldsymbol{\xi}_i^2\right)=\mathbf{I}_{N\times N}}\cdot\widehat{\boldsymbol{\delta}\mathbf{X}}^T=\varphi\cdot\mathbf{I}_{n\times n}+\delta\cdot\widehat{\boldsymbol{\delta}\mathbf{X}}\cdot\widehat{\boldsymbol{\delta}\mathbf{X}}^T$$

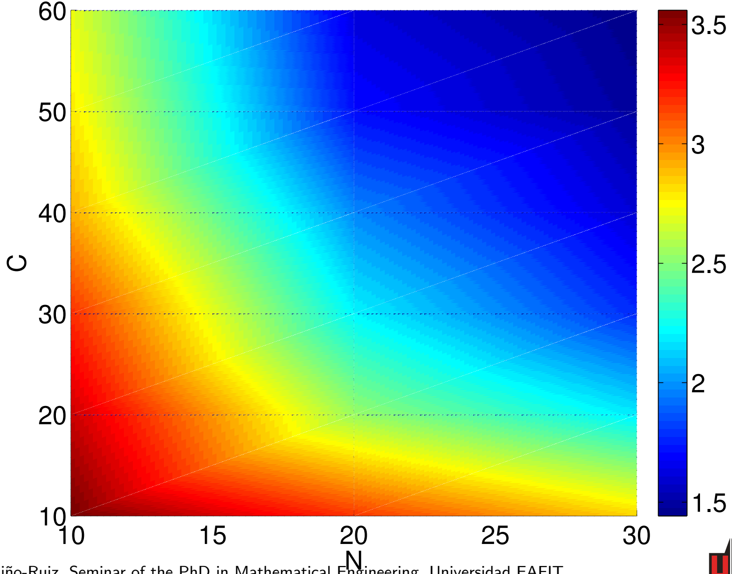
$$=\widehat{\mathbf{B}}\,.$$

Sampling in High Dimensions IV

- The synthetic members are obtained as follows:

$$\mathbf{x}_i^s = \overline{\mathbf{x}}^b + \sqrt{\varphi} \cdot \boldsymbol{\xi}_i^1 + \sqrt{\delta} \cdot \widehat{\boldsymbol{\delta}\mathbf{X}} \cdot \boldsymbol{\xi}_i^2, \quad i = 1, \ldots, K.$$

Importance of Synthetic Members

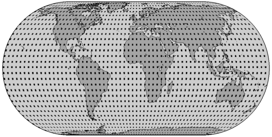


(c)

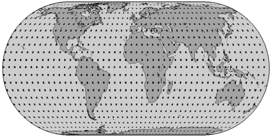
EnKF-MC and EnKF-SC with the SPEEDY Model I

- We make use of FORTRAN 90 in order to code the EnKF-MC and the EnKF-RBLW (from now on EnKF-SC).
- 96 ensemble members were used for the experiments.
- The initial perturbation of the background state is 5% the true state of the system.
- The model is propagated for a period of 24 days, observations are taken every 2 days.
- The SPEEDY model is used with T-63 resolution (96×192) with 4 variables. 8 layers per variable. $n \approx 590,000$.
- Three sparse observational networks were used for the tests.
- We compare the results with the LETKF [OHS+04, BT99].

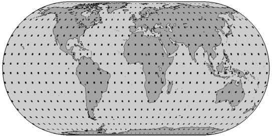
EnKF-MC and EnKF-SC with the SPEEDY Model II



(d) $p = 12\%$



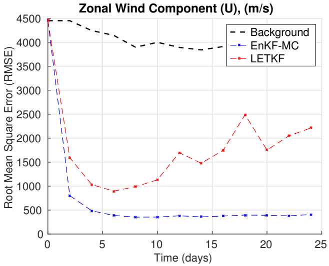
(e) $p = 6\%$



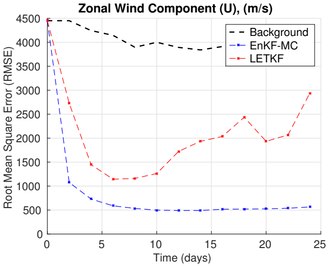
(f) $p = 4\%$

Figure: Observational networks for different values of p.

Accuracy of the EnKF-MC I



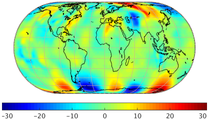
(a) $r = 3$ and $p = 12\%$



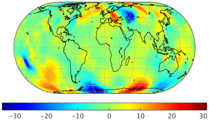
(b) $r = 5$ and $p = 6\%$

Figure: RMSE of the LETKF and EnKF-MC implementations for different model variables, radii of influence and observational networks.

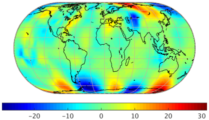
Accuracy of the EnKF-MC II



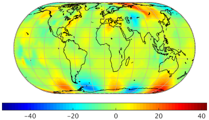
(a) Reference



(b) Background



(c) EnKF-MC

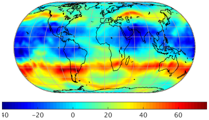


(d) LETKF

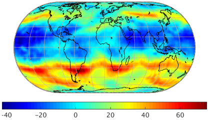
Figure: 5-th layer of the meridional wind component (v).

UNIVERSIDAD DEL NORTE

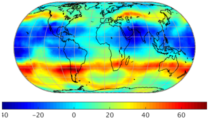
Accuracy of the EnKF-MC III



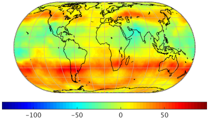
(a) Reference



(b) Background



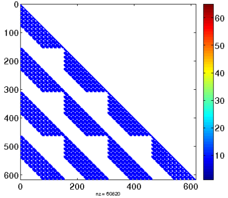
(c) EnKF-MC



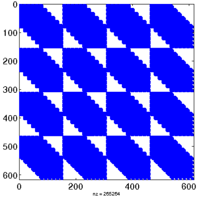
(d) LETKF

Figure: 2-th layer of the zonal wind component (u).

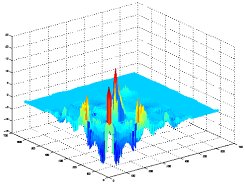
Local Estimation of $\mathbf{B}^{-1}$



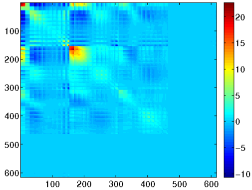
(a) $\mathbf{T}$



(b) $\widehat{\mathbf{B}}^{-1}$

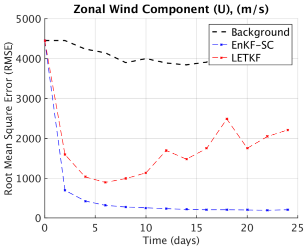


(c) $\widehat{\mathbf{B}}$

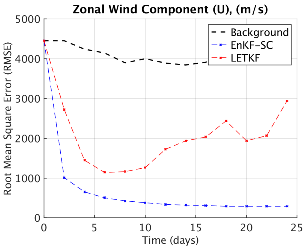


(d) $\widehat{\mathbf{B}}$

Accuracy of the EnKF-RBLW I



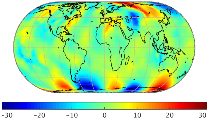
(e) $r = 3$ and $p = 12\%$



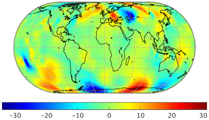
(f) $r = 5$ and $p = 6\%$

Figure: RMSE of the LETKF and EnKF-RBLW implementations for different model variables, radii of influence and observational networks.

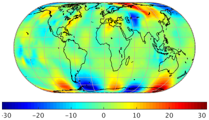
Accuracy of the EnKF-RBLW II



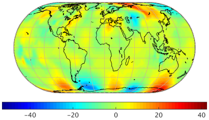
(a) Reference



(b) Background



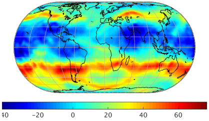
(c) EnKF-RBLW



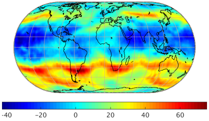
(d) LETKF

Figure: 5-th layer of the meridional wind component (v).

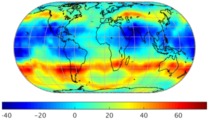
Accuracy of the EnKF-RBLW III



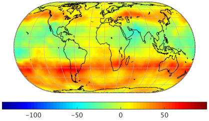
(a) Reference



(b) Background



(c) EnKF-RBLW



(d) LETKF

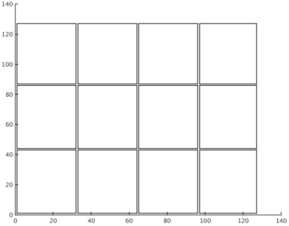
Figure: 2-th layer of the zonal wind component (u).

Parallel implementations of ensemble based methods

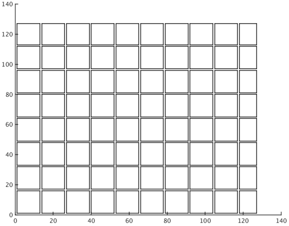
- Blueridge Super Computer @ VT
  - BlueRidge is a 408-node Cray CS-300 cluster.
  - Each node is outfitted with two octa-core Intel Sandy Bridge CPUs and 64 GB of memory.
  - Total of 6,528 cores and 27.3 TB of memory systemwide.
  - Eighteen nodes have 128 GB of memory.
  - In addition, 130 nodes are outfitted with two Intel MIC (Xeon Phi) coprocessors.
- The methods are coded in FORTRAN using MPI.
- LAPACK [ABD+90] and BLAS [BDD+01] are used in order to efficiently perform matrix computations.
- We vary the number of processors from 96 (16 computing nodes) to 2,048 (128 computing nodes)

Parallel implementations of ensemble based methods I

- The approximations are based on **domain decomposition**



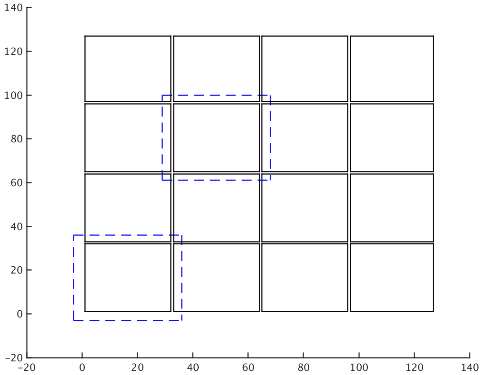
(a) 12



(b) 80

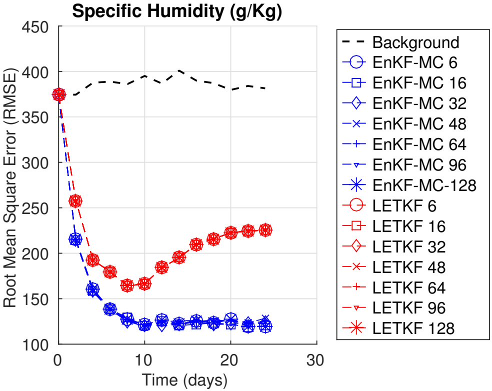
Parallel implementations of ensemble based methods II

- Boundary information



Parallel implementations of ensemble based methods III

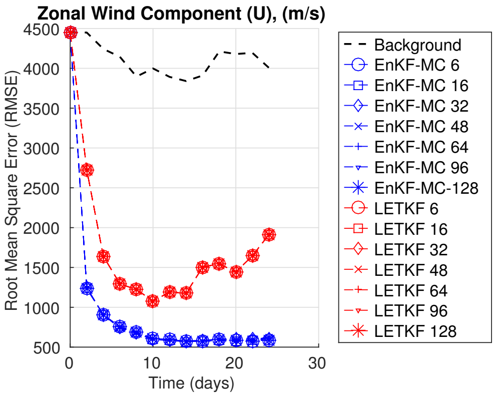
- Accuracy (EnKF-MC): number of processors ranges from 96 (16 computing nodes) to 2,048 (128 computing nodes)



UNIVERSIDAD DEL NORTE

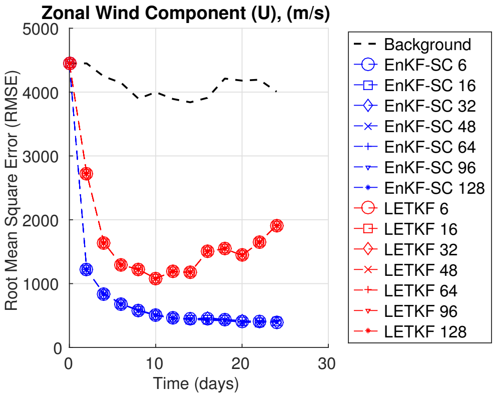
Parallel implementations of ensemble based methods IV

- Accuracy (EnKF-MC): number of processors ranges from 96 (16 computing nodes) to 2,048 (128 computing nodes)



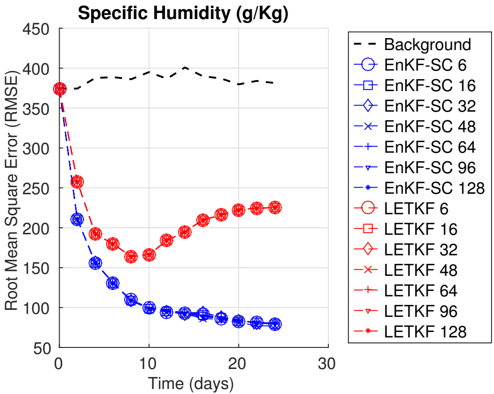
Parallel implementations of ensemble based methods V

- Accuracy (EnKF-RBLW): number of processors ranges from 96 (16 computing nodes) to 2,048 (128 computing nodes)



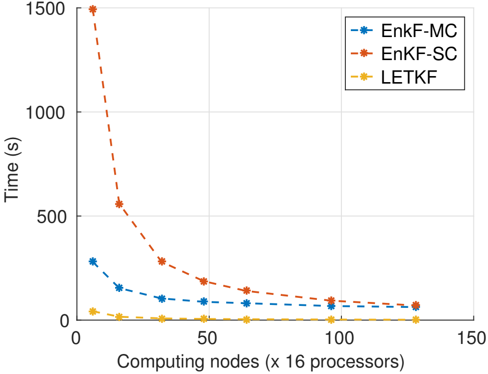
Parallel implementations of ensemble based methods VI

- Accuracy (EnKF-RBLW): number of processors ranges from 96 (16 computing nodes) to 2,048 (128 computing nodes)



Parallel implementations of ensemble based methods VII

- Computational time: number of processors ranges from 96 (16 computing nodes) to 2,048 (128 computing nodes)



EnKF-MC Publications

1. Elias D. Nino-Ruiz, Adrian Sandu, and Xinwei Deng. "An Ensemble Kalman Filter Implementation Based on Modified Cholesky Decomposition for Inverse Covariance Matrix Estimation", SIAM Journal on Scientific Computing 40:2, A867-A886 (2018).
2. Elias D. Nino-Ruiz, "A Matrix-Free Posterior Ensemble Kalman Filter Implementation Based on a Modified Cholesky Decomposition", Atmosphere Journal, MDPI Publisher, 8:125, (2017).
3. Elias D. Nino-Ruiz, Adrian Sandu, and Xinwei Deng. "A parallel implementation of the ensemble Kalman filter based on modified Cholesky decomposition", Journal of Computational Science, Elsevier, (2017).

EnKF-SC Publications

1. Elias D. Nino-Ruiz, and Adrian Sandu. "Efficient Parallel Implementation of DDDAS Inference using an Ensemble Kalman Filter with Shrinkage Covariance Matrix Estimation". Cluster Computing, Springer. (2017).
2. Cosmin G. Petraa, Victor M. Zavalab, Elias D. Nino-Ruiz, and Mihai Anitescud. "A high-performance computing framework for analyzing the economic impacts of wind correlation." Electric Power Systems Research, Elsevier, 141 (2016): 372-380.
3. Nino-Ruiz, Elias D., and Adrian Sandu. "Ensemble Kalman filter implementations based on shrinkage covariance matrix estimation." Ocean Dynamics, Springer, 65.11 (2015): 1423-1439.

Bibliography I

[ABD+90] E. Anderson, Z. Bai, J. Dongarra, A. Greenbaum, A. McKenney, J. Du Croz, S. Hammerling, J. Demmel, C. Bischof, and D. Sorensen. LAPACK: A Portable Linear Algebra Library for High-performance Computers. In *Proceedings of the 1990 ACM/IEEE Conference on Supercomputing*, Supercomputing '90, pages 2–11, Los Alamitos, CA, USA, 1990. IEEE Computer Society Press.

[AND07] JEFFREY L. ANDERSON. An adaptive covariance inflation error correction algorithm for ensemble filters. *Tellus A*, 59(2):210–224, 2007.

[AND09] JEFFREY L. ANDERSON. Spatially and temporally varying adaptive covariance inflation for ensemble filters. *Tellus A*, 61(1):72–83, 2009.

[BDD+01] L. S. Blackford, J. Demmel, J. Dongarra, I. Duff, S. Hammarling, G. Henry, M. Heroux, L. Kaufman, A. Lumsdaine, A. Petitet, R. Pozo, K. Remington, and R. C. Whaley. An Updated Set of Basic Linear Algebra Subprograms (BLAS). *ACM Transactions on Mathematical Software*, 28:135–151, 2001.

[BS12] M. Bocquet and P. Sakov. Combining Inflation-free and Iterative Ensemble Kalman Filters for Strongly Nonlinear Systems. *Nonlinear Processes in Geophysics*, 19(3):383–399, 2012.

[BT99] Craig H. Bishop and Zoltan Toth. Ensemble Transformation and Adaptive Observations. *Journal of the Atmospheric Sciences*, 56(11):1748–1765, 1999.

[Bue11] Mark Buehner. Evaluation of a Spatial/Spectral Covariance Localization Approach for Atmospheric Data Assimilation. *Monthly Weather Review*, 140(2):617–636, 2011.

[CM14] Romain Couillet and Matthew McKay. Large Dimensional Analysis and Optimization of Robust Shrinkage Covariance Matrix Estimators. *Journal of Multivariate Analysis*, 131(0):99–120, 2014.

[CWEH10] Yilun Chen, A Wiesel, Y.C. Eldar, and AO. Hero. Shrinkage Algorithms for MMSE Covariance Estimation. *Signal Processing, IEEE Transactions on*, 58(10):5016–5029, Oct 2010.

[CWH11] Yilun Chen, A Wiesel, and AO. Hero. Robust Shrinkage Estimation of High-Dimensional Covariance Matrices. *Signal Processing, IEEE Transactions on*, 59(9):4097–4107, Sept 2011.

[Eve06] Geir Evensen. *Data Assimilation: The Ensemble Kalman Filter*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

Bibliography II

[LW04] Olivier Ledoit and Michael Wolf. A Well-conditioned Estimator for Large-dimensional Covariance Matrices. *Journal of Multivariate Analysis*, 88(2):365 – 411, 2004.

[NR17] Elias D Nino-Ruiz. A matrix-free posterior ensemble kalman filter implementation based on a modified cholesky decomposition. *Atmosphere*, 8(7):125, 2017.

[NRSD15] Elias D. Nino-Ruiz, Adrian Sandu, and Xinwei Deng. A parallel ensemble kalman filter implementation based on modified cholesky decomposition. In *Proceedings of the 6th Workshop on Latest Advances in Scalable Algorithms for Large-Scale Systems*, ScalA '15, pages 4:1–4:8, New York, NY, USA, 2015. ACM.

[NRSD17] Elias D Nino-Ruiz, Adrian Sandu, and Xinwei Deng. A parallel implementation of the ensemble kalman filter based on modified cholesky decomposition. *Journal of Computational Science*, 2017.

[NRSD18] Elias D Nino-Ruiz, Adrian Sandu, and Xinwei Deng. An ensemble kalman filter implementation based on modified cholesky decomposition for inverse covariance matrix estimation. *SIAM Journal on Scientific Computing*, 40(2):A867–A886, 2018.

[OHS+04] Edward Ott, Brian R. Hunt, Istvan Szunyogh, Aleksey V. Zimin, Eric J. Kostelich, Matteo Corazza, Eugenia Kalnay, D. J. Patil, and James A. Yorke. A local ensemble kalman filter for atmospheric data assimilation. *Tellus A*, 56(5):415–428, 2004.