Some ideas to improve LETKF

We can adapt ideas developed within 4D-Var:

- No-cost smoother
- Accelerating the spin-up: Running in place
- "Outer loop" and nonlinearities
- Forecast sensitivity to observations
- Coarse analysis resolution interpolating weights
- Low-dimensional model bias correction

Local Ensemble Transform Kalman Filter (LETKF)

Globally:

Forecast step: $\mathbf{x}^b_{n,k} = M_n\left(\mathbf{x}^a_{n-1,k}\right)$

Analysis step: construct $\mathbf{X}^b = \left[\mathbf{x}^b_1 - \bar{\mathbf{x}}^b \mid \ldots \mid \mathbf{x}^b_K - \bar{\mathbf{x}}^b\right];$

$$\mathbf{y}^b_i = H(\mathbf{x}^b_i);\ \mathbf{Y}^b_n = \left[\mathbf{y}^b_1 - \bar{\mathbf{y}}^b \mid \ldots \mid \mathbf{y}^b_K - \bar{\mathbf{y}}^b\right]$$

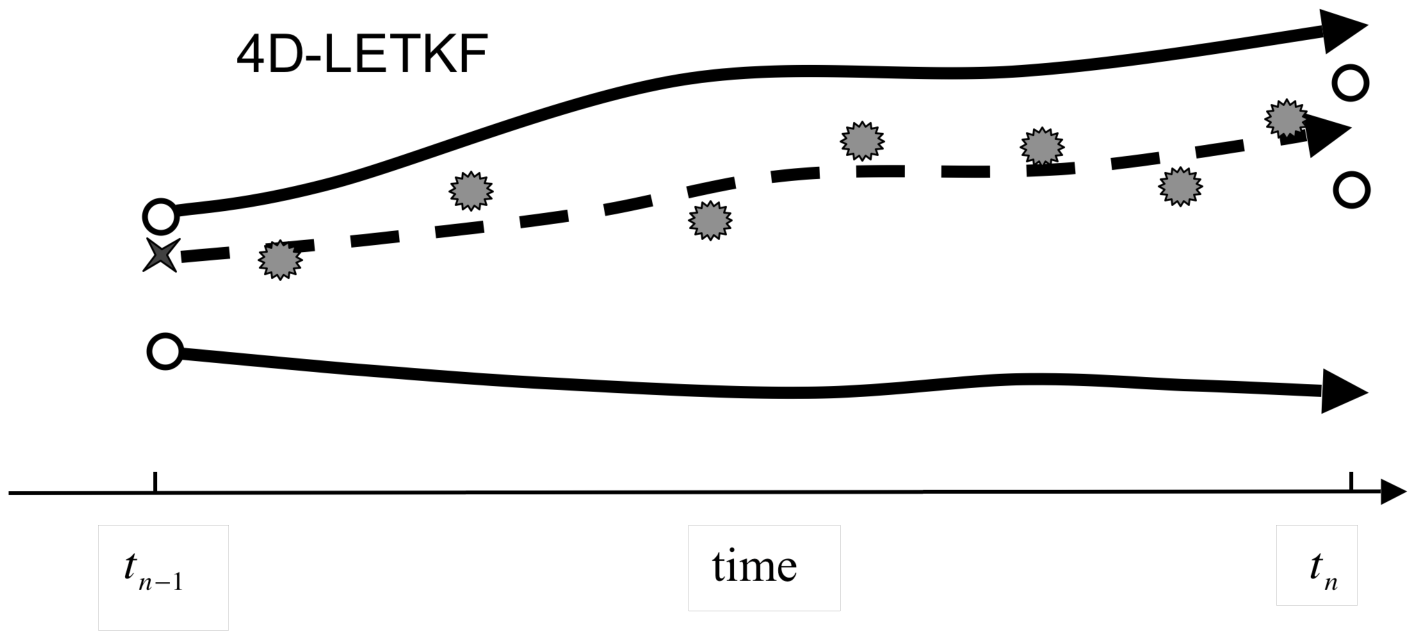
Locally: Choose for each grid point the observations to be used, and compute the local analysis error covariance and perturbations in ensemble space:

$$\tilde{\mathbf{P}}^a = \left[\left(K-1\right)\mathbf{I} + \mathbf{Y}^{bT}\mathbf{R}^{-1}\mathbf{Y}^b\right]^{-1};\ \mathbf{W}^a = [(K-1)\tilde{\mathbf{P}}^a]^{1/2}$$

Analysis mean in ensemble space: $\bar{\mathbf{w}}^a = \tilde{\mathbf{P}}^a\mathbf{Y}^{bT}\mathbf{R}^{-1}(\mathbf{y}^o - \bar{\mathbf{y}}^b)$

and add to $\mathbf{W}^a$ to get the analysis ensemble in ensemble space

The new ensemble analyses in model space are the columns of $\mathbf{X}^a_n = \mathbf{X}^b_n\mathbf{W}^a + \bar{\mathbf{x}}^b$. Gathering the grid point analyses forms the new global analyses. Note that the LETKF produces analysis weights $\bar{\mathbf{w}}^a$ and perturbation analysis matrices of weights $\mathbf{W}^a$



The LETKF produces an analysis in terms of weights of the ensemble forecast members at the analysis time t_n, giving the trajectory that best fits all the observations in the assimilation window, in 3D- or in 4-D

No-cost LETKF smoother (cross): apply at t_{n-1} the same weights found optimal at t_n, works for 3D- or 4D-LETKF

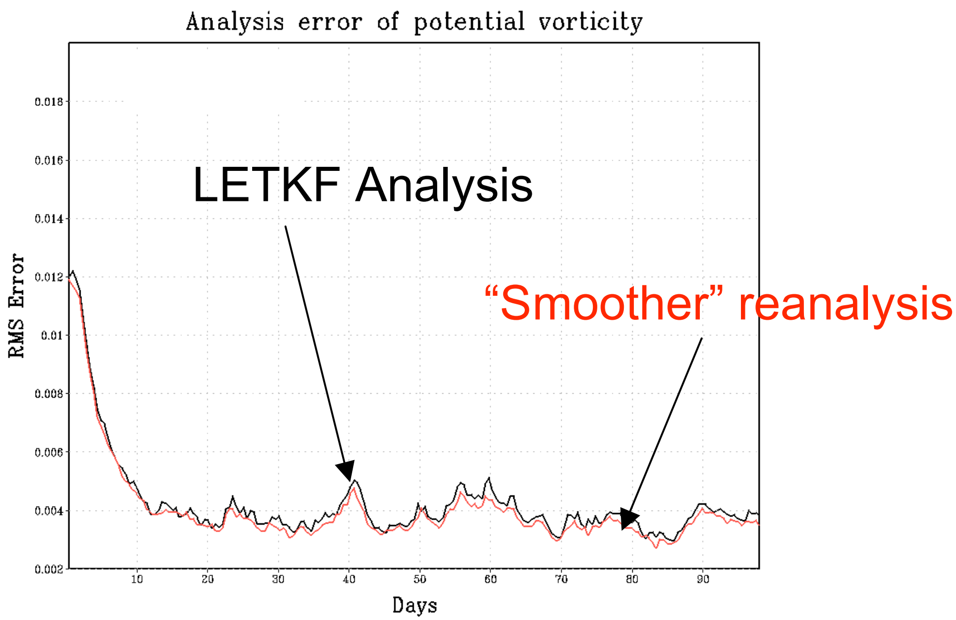
No-cost LETKF smoother

LETKF analysis
at time n

$$\bar{\mathbf{x}}_n^a = \bar{\mathbf{x}}_n^f + \mathbf{X}_n^f \bar{\mathbf{w}}_n$$

Smoother analysis
at time n-1

$$\tilde{\mathbf{x}}_{n-1}^a = \bar{\mathbf{x}}_{n-1}^f + \mathbf{X}_{n-1}^f \bar{\mathbf{w}}_n$$



“Running in place” to spin-up faster

Kalnay and Yang (2008)

- 4D-Var spins-up faster than EnKF because it is a smoother: it keeps iterating until it fits the observations within the assimilation window as well as possible
- EnKF spins-up fast if starting from a “good” initial state, e.g., 3D-Var
- In a severe storm where radar observations start with the storm, there is little real time to spin-up
- Caya et al. (2005): “EnKF is eventually better than 4D-Var” (but it is too late to be useful, it misses the storm). Also Jidong Gao, (pers. comm.), spin-up ruins the use of EnKF for storm prediction.

Can we use the data more than once?

- **Hunt et al., 2007**: The background term represents the evolution of the maximum likelihood trajectory given all the observations in the past

$$\sum_{j=1}^{n-1}\left[\mathbf{y}_j^o-\mathbf{H}_j\mathbf{M}_{t_n,t_j}\mathbf{x}\right]^T\mathbf{R}_j^{-1}\left[\mathbf{y}_j^o-\mathbf{H}_j\mathbf{M}_{t_n,t_j}\mathbf{x}\right]=\left[\mathbf{x}-\overline{\mathbf{x}}_n^b\right]^T\left(\mathbf{P}_n^b\right)^{-1}\left[\mathbf{x}-\overline{\mathbf{x}}_n^b\right]+c$$

- After the analysis a similar relationship is valid:

$$\left[\mathbf{x}-\overline{\mathbf{x}}_n^b\right]^T\left(\mathbf{P}_n^b\right)^{-1}\left[\mathbf{x}-\overline{\mathbf{x}}_n^b\right]+\left[\mathbf{y}_n^o-\mathbf{H}_n\mathbf{x}\right]^T\left(\mathbf{R}_n^{-1}\right)\left[\mathbf{y}_n^o-\mathbf{H}_n\mathbf{x}\right]=\left[\mathbf{x}-\overline{\mathbf{x}}_n^a\right]^T\left(\mathbf{P}_n^a\right)^{-1}\left[\mathbf{x}-\overline{\mathbf{x}}_n^a\right]+c'$$

- From here one can derive the linear KF equations
- Also the rule: “Use the data once and then discard it”

Can we use the data more than once?

- The rule: "Use the data once and then discard it" makes sense when the analysis/forecasts are the most likely given all the past data, not when we start from scratch. Hence we propose "Running in place" until we extract the maximum information form the observations.
- We need
  - 4D-LETKF (Hunt et al, 2004) to use all the observations within an assimilation window at their right time
  - A No-Cost Smoother (Kalnay et al., 2007b)
  - An appropriate iterative scheme

“Running in Place”

- EnKF is a sequential data assimilation system where, after the new data is used at the analysis time, it should be discarded…
- **only** if the previous analysis and the new background are the most likely states given the past observations.
- **If the system has converged after the initial spin-up all the information from past observations is already included in the background.**
- **During the spin-up we should use the observations repeatedly if we can extract extra information**
- But we should avoid overfitting the observations

Running in Place algorithm (1)

- Cold-start the EnKF from any initial ensemble mean and random perturbations at t_0, and integrate the initial ensemble to t_1. The “running in place” loop with $n=1$, is:

Running in Place algorithm (2)

a) Perform a standard EnKF analysis and obtain the analysis weights at t_n, saving the mean square observations minus forecast (OMF) computed by the EnKF.

b) Apply the no-cost smoother to obtain the smoothed analysis ensemble at t_{n-1} by using the same weights obtained at t_n.

c) Perturb the smoothed analysis ensemble with a small amount of random Gaussian perturbations, a method similar to additive inflation.

d) Integrate the perturbed smoothed ensemble to t_n. If the forecast fit to the observations is smaller than in the previous iteration according to some criterion, go to a) and perform another iteration. If not, let $t_{n-1} \leftarrow t_n$ and proceed to the next assimilation window.

Running in Place algorithm (notes)

Notes:

c) *Perturb the smoothed analysis ensemble with a small amount of random Gaussian perturbations, a method similar to additive inflation.*

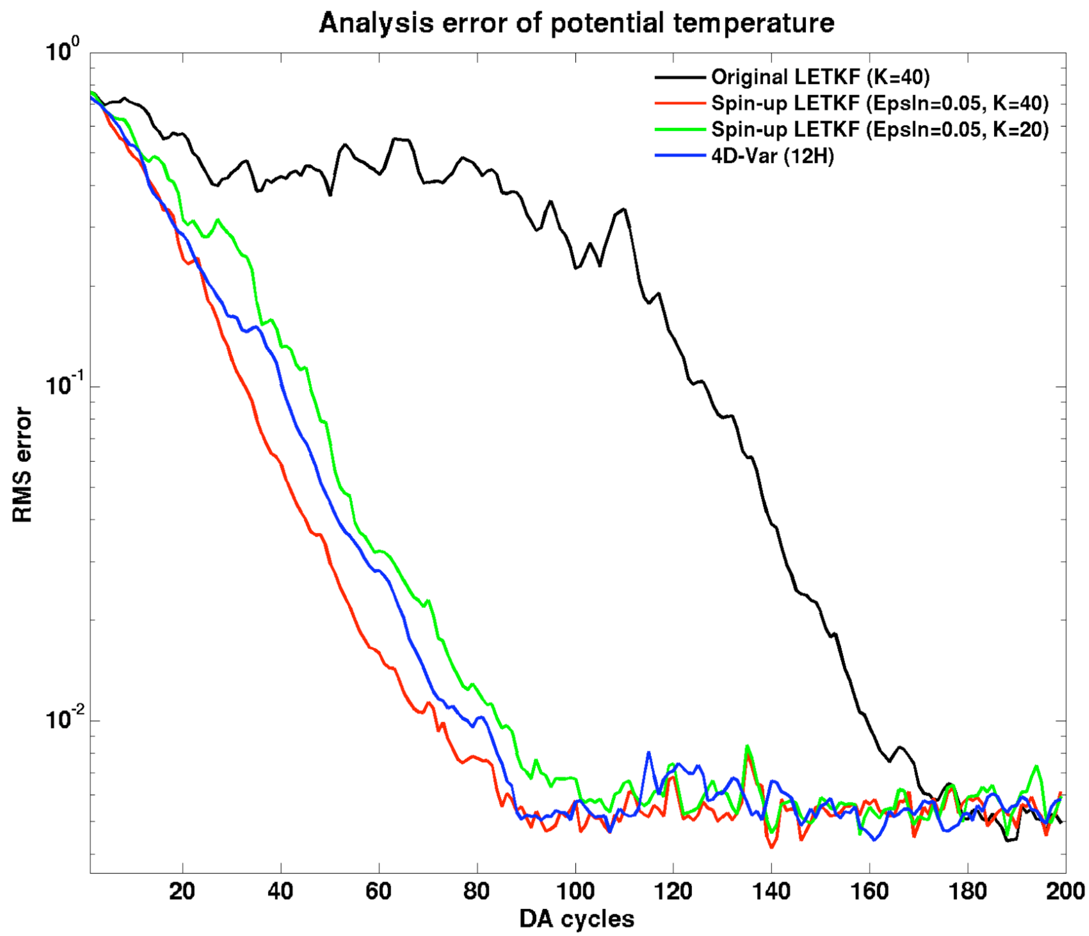
This perturbation has two purposes:

1) Avoid reaching the same analysis as before, and
2) Encourage the ensemble to explore new unstable directions

d) *Convergence criterion*: if $$\frac{OMF^2(iter) - OMF^2(iter+1)}{OMF^2(iter)} > \varepsilon$$

with $\varepsilon \sim 5\%$ do another iteration. Otherwise go to the next assimilation window.

Results with a QG model



It works well (red)… spin-up becomes as fast as 4D-Var (blue). With 5% criterion for reduction of OMF errors (red), it takes only 2-4 iterations. With 20 members (green) it still works.

Discussion of spin-up acceleration

- Number of iterations during spin-up: 2-4, computationally acceptable
- We could use the weights interpolation of Yang et al. (2008b) and run in place only where “the action is”.
- There are many applications where a fast spin-up is important.

Nonlinearities and "outer loop"

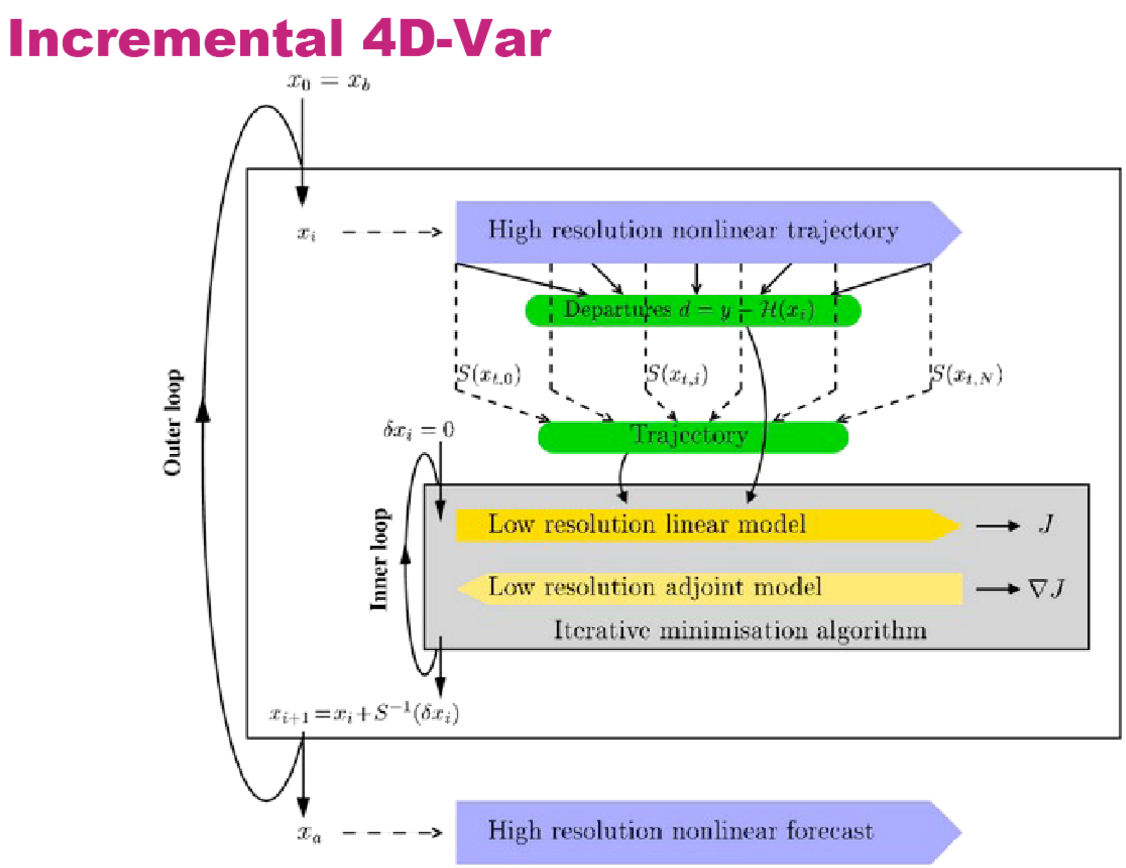
- The main disadvantage of EnKF is that it cannot handle nonlinear (non-Gaussian) perturbations and therefore needs short assimilation windows.
- It doesn't have the important outer loop so important in 3D-Var and 4D-Var (DaSilva, pers. comm. 2006)

Lorenz -3 variable model (Kalnay et al. 2007a Tellus), RMS analysis error

	4D-Var	LETKF	
Window=8 steps	0.31	**0.30**	(linear window)
Window=25 steps	**0.53**	0.66	(nonlinear window)

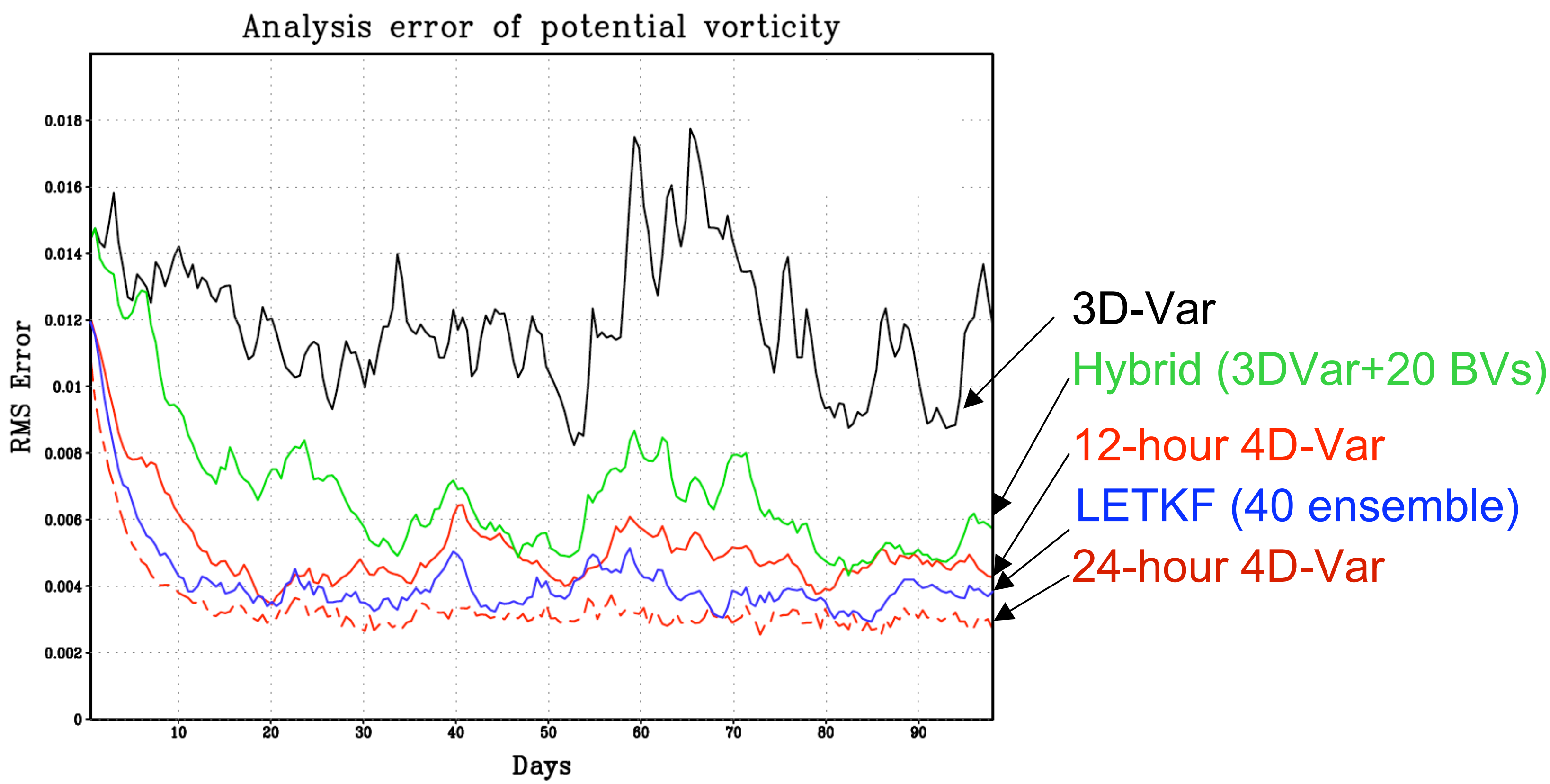
Long windows + Pires et al. => 4D-Var wins!

“Outer loop” in 4D-Var



ECMWF

Comparison of ensemble-based and variational-based data assimilation schemes in a Quasi-Geostrophic model.



EnKF does not handle well long windows because ensemble perturbations become non-Gaussian. 4D-Var simply iterates and produces a more accurate control. We can imitate this with the “outer loop” idea for LETKF.

Nonlinearities and "outer loop"

Outer loop: do the same as 4D-Var, and use the final weights to correct only the <u>mean</u> initial analysis, keeping the initial perturbations. Repeat the analysis once or twice. It centers the ensemble on a more accurate nonlinear solution.

Lorenz -3 variable model RMS analysis error

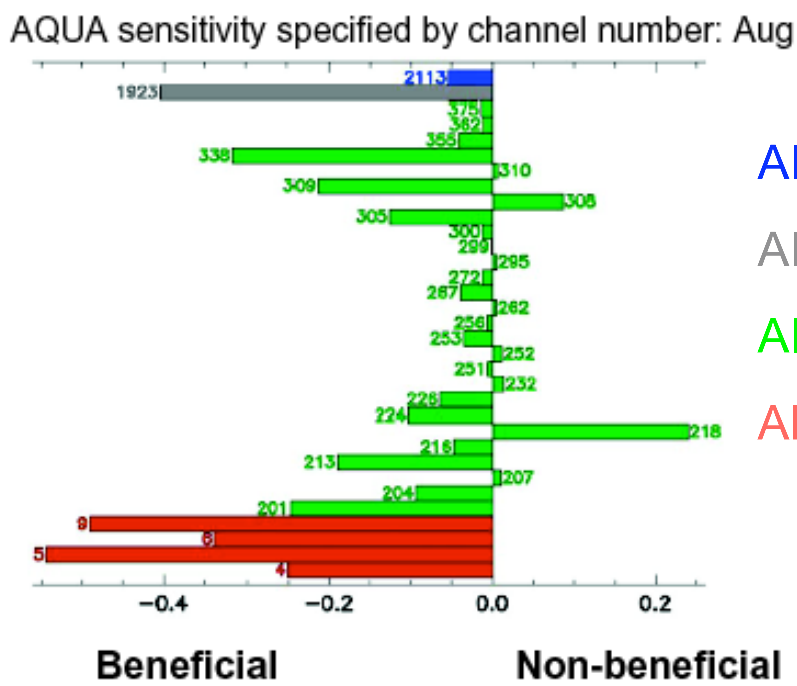
	4D-Var	LETKF	LETKF +outer loop
Window=8 steps	0.31	0.30	**0.27**
Window=25 steps	0.53	0.66	**0.48**

Running in place further reduces from 0.48 to 0.39!

Estimation of observation impact without adjoint in an ensemble Kalman filter

Junjie Liu and Eugenia Kalnay

Background



AIRS shortwave 4.180 μm

AIRS shortwave 4.474 μm

AIRS longwave 14-13 μm

AMSU/A

- The adjoint method proposed by Langland and Baker (2004) and Zhu and Gelaro (2007) quantifies the reduction in forecast error for each individual observation source
- The adjoint method detects the observations which make the forecast worse.
- The adjoint method requires adjoint model which is difficult to get.

Objective and outline

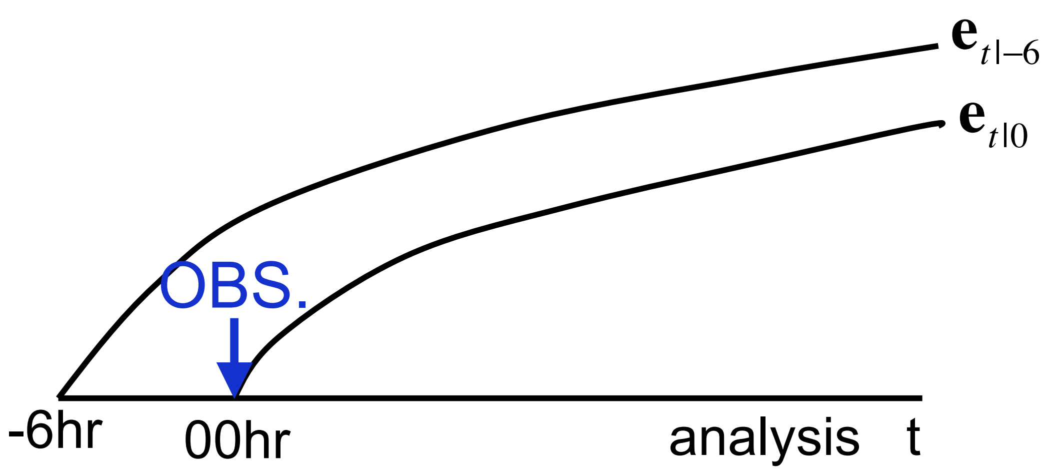
Objective

- Propose an ensemble sensitivity method to calculate observation impact without using adjoint model.

Outline

- Illustrate and derive the ensemble sensitivity method;
- With Lorenz-40 variable model, compare the ensemble sensitivity method with adjoint method in
  - the ability to represent the actual error reduction;
  - the ability to detect the poor quality observations.
- Summary

Schematic of the observation impact on the reduction of forecast error



$$\mathbf{e}_{t|-6} = \overline{\mathbf{x}}^f_{t|-6} - \overline{\mathbf{x}}^a_t$$

$$\mathbf{e}_{t|0} = \overline{\mathbf{x}}^f_{t|0} - \overline{\mathbf{x}}^a_t$$

(Adapted from Langland and Baker, 2004)

The only difference between $\mathbf{e}_{t|0}$ and $\mathbf{e}_{t|-6}$ is the assimilation of observations at 00hr.

- Observation impact on the reduction of forecast error: $J = \frac{1}{2}(\mathbf{e}^T_{t|0}\mathbf{e}_{t|0} - \mathbf{e}^T_{t|-6}\mathbf{e}_{t|-6})$

The ensemble sensitivity method

Euclidian cost function: $$J = \frac{1}{2}(\mathbf{e}_{t|0}^T \mathbf{e}_{t|0} - \mathbf{e}_{t|-6}^T \mathbf{e}_{t|-6}) \qquad \mathbf{v}_0 = \mathbf{y}_0^o - h(\overline{\mathbf{x}}_{0|-6}^b)$$

Cost function as function of obs. increments: $$J = \left\langle \mathbf{v}_0, \frac{\partial J}{\partial \mathbf{v}_0} \right\rangle$$

The ensemble sensitivity method

Euclidian cost function: $J = \frac{1}{2}(\mathbf{e}_{t|0}^T \mathbf{e}_{t|0} - \mathbf{e}_{t|-6}^T \mathbf{e}_{t|-6})$ $\quad \mathbf{v}_0 = \mathbf{y}_0^o - h(\overline{\mathbf{x}}_{0|-6}^b)$

Cost function as function of obs. Increments: $J = \left\langle \mathbf{v}_0, \frac{\partial J}{\partial \mathbf{v}_0} \right\rangle$

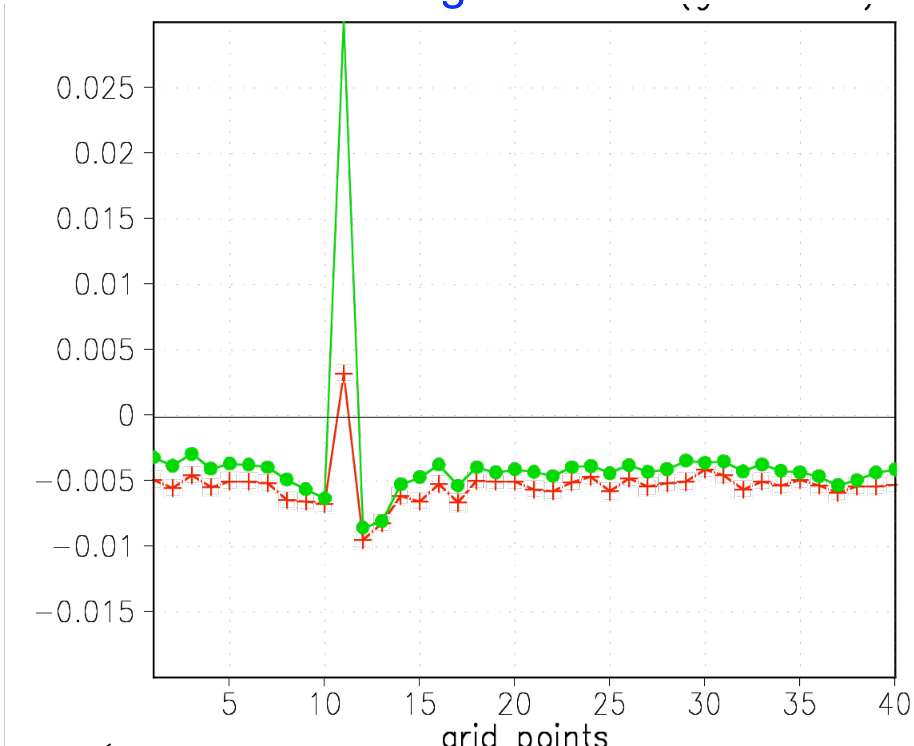
The sensitivity of cost function with respect to the assimilated observations:

$$\frac{\partial J}{\partial \mathbf{v}_0} = \left[\tilde{\mathbf{K}}_0^T \mathbf{X}_{t|-6}^{fT} \right] \left[\mathbf{e}_{t|-6} + \mathbf{X}_{t|-6}^f \tilde{\mathbf{K}}_0 \mathbf{v}_0 \right]$$

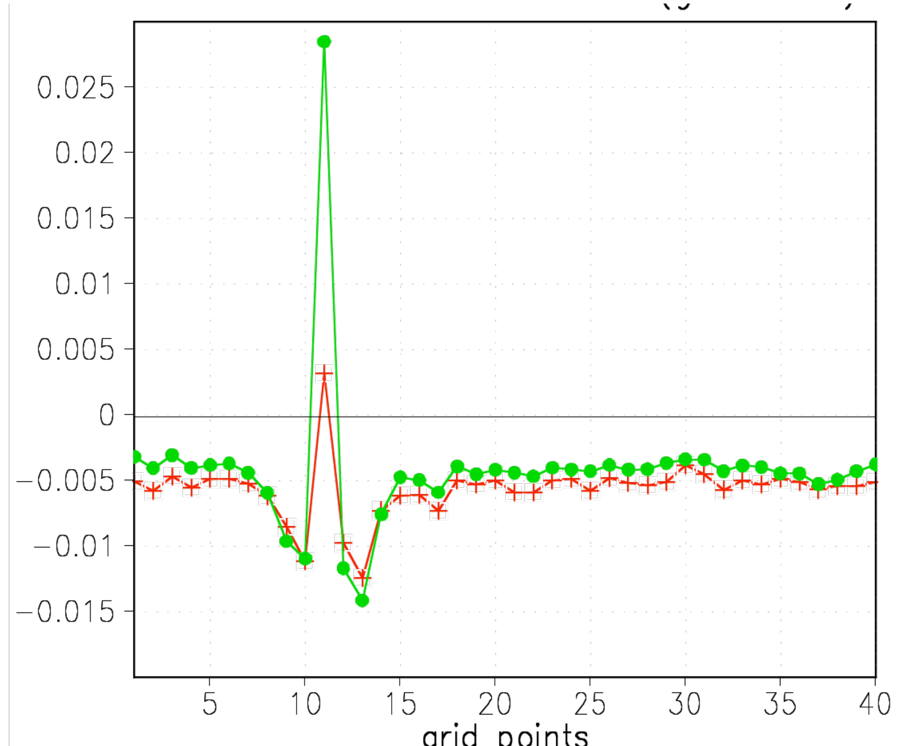
Ability to detect the poor quality observation

Observation impact from LB (red) and from ensemble sensitivity method (green)

Larger random error



Biased observation case



✓ Like adjoint method, ensemble sensitivity method can detect the observation poor quality (11th observation location)

✓ The ensemble sensitivity method has a stronger signal when the observation has negative impact on the forecast.

Summary for forecast sensitivity to obs.

- Derived a formula to calculate the observation impact based on the ensemble without using the adjoint model which usually is not available.

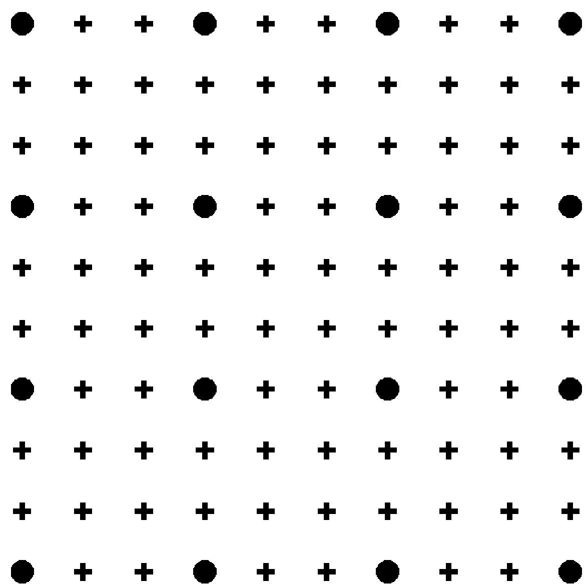
- The results based on Lorenz-40 variable model show that ensemble sensitivity method without using adjoint model gives results similar to adjoint method .

- Like adjoint method, ensemble sensitivity method can detect the observation which either has larger random error or has bias. Under such conditions, the ensemble sensitivity method has stronger signal.

- It provides a powerful tool to check the quality of the observations.

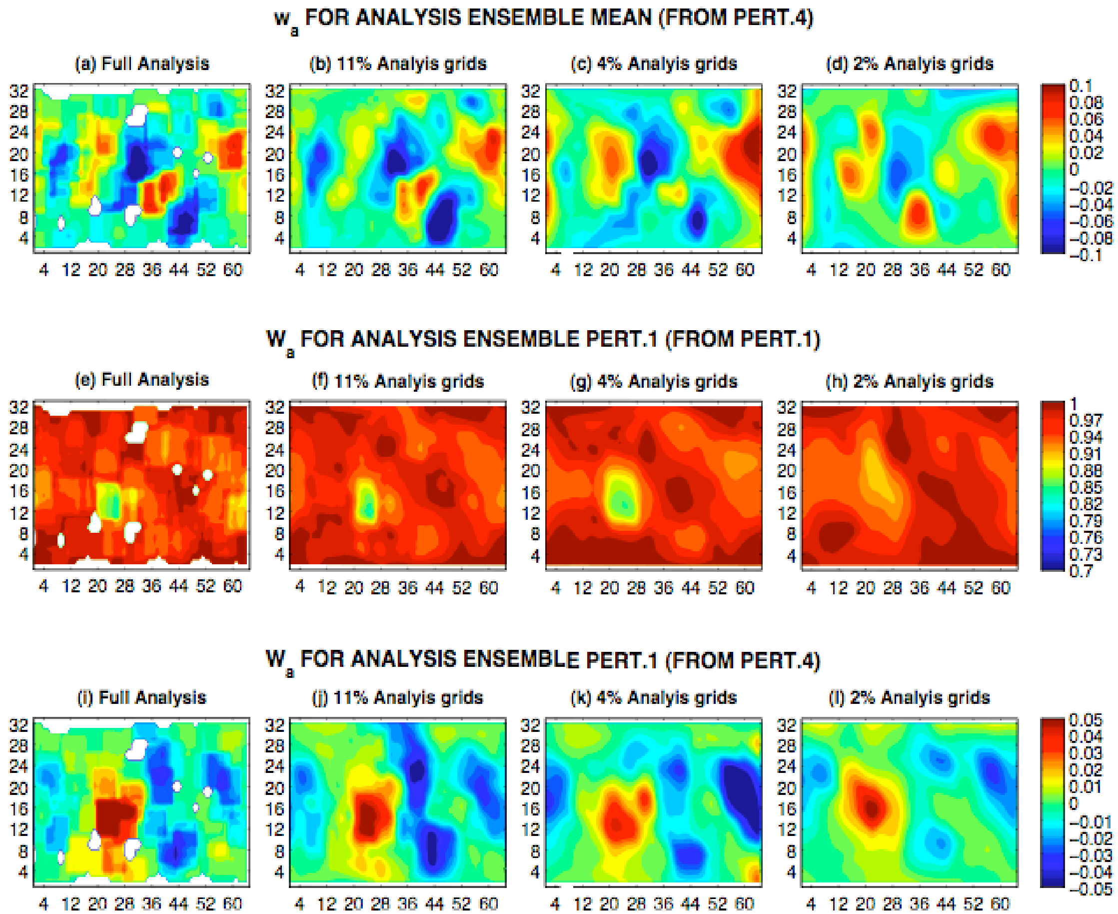
Coarse analysis with interpolated weights
Yang et al (2008)

- In EnKF the analysis is a weighted average of the forecast ensemble
- We performed experiments with a QG model **interpolating weights** compared to **analysis increments**.
- Coarse grids of 11%, 4% and 2% interpolated analysis points.



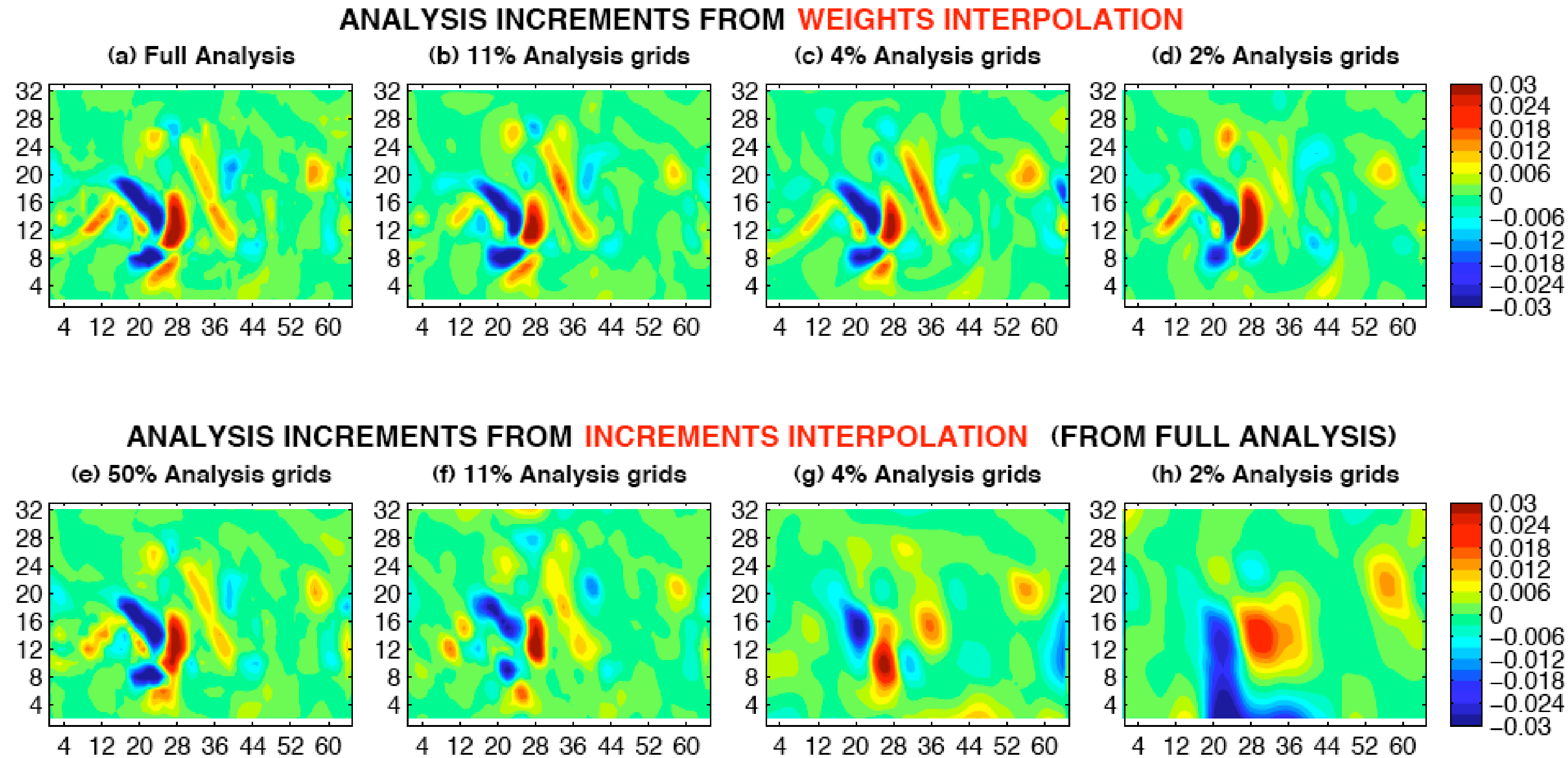
1/(3x3)=11% analysis grid

Coarse analysis with interpolated weights



- Weights vary on very scales: they interpolate well.
- Interpolated weights are obtained even for data void areas.

Analysis increments

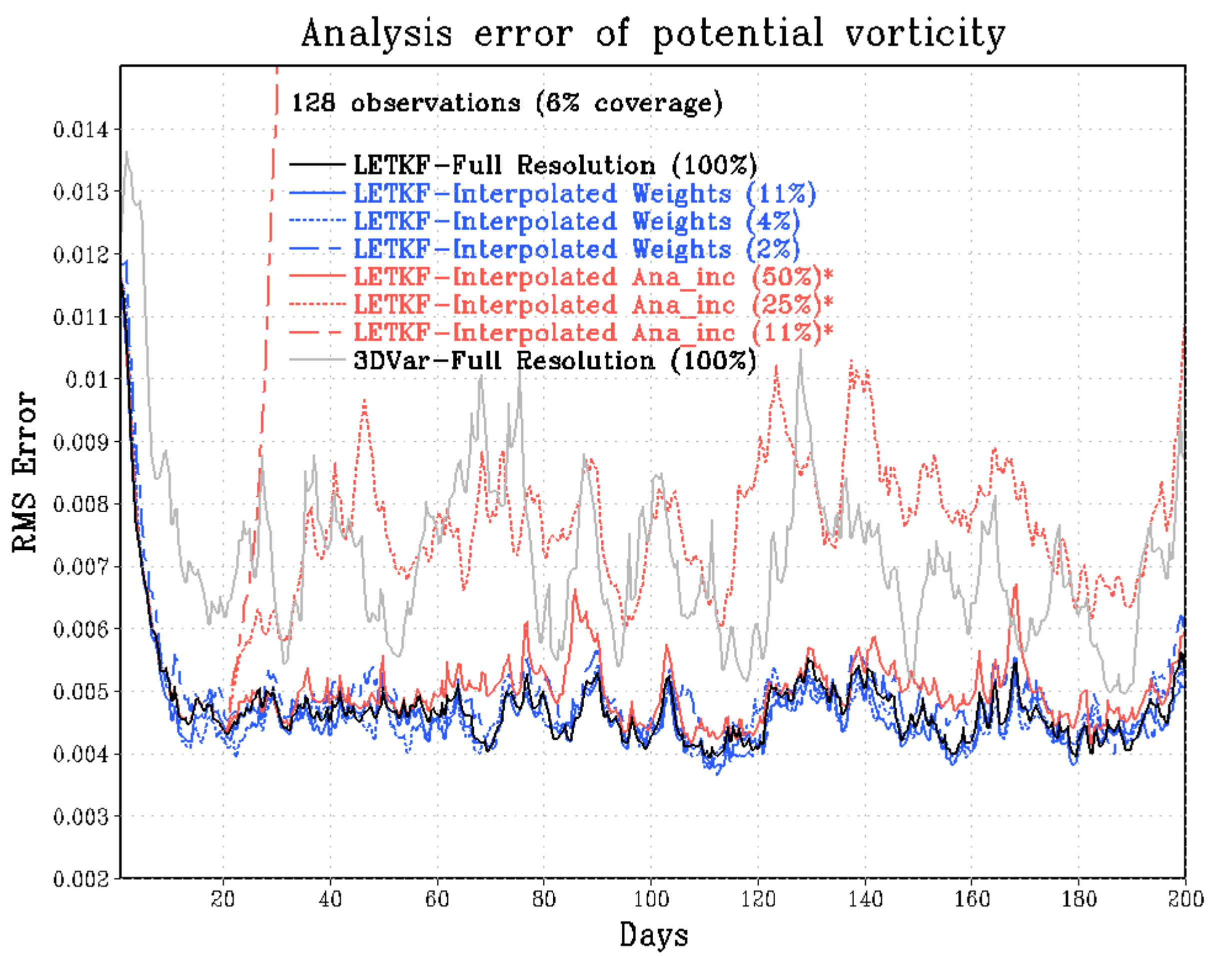


With increment interpolation, the analysis is OK only with 50% analysis coverage

With weight interpolation, there is almost no degradation!

EnKF maintains balance and conservation properties

Impact of coarse analysis on accuracy



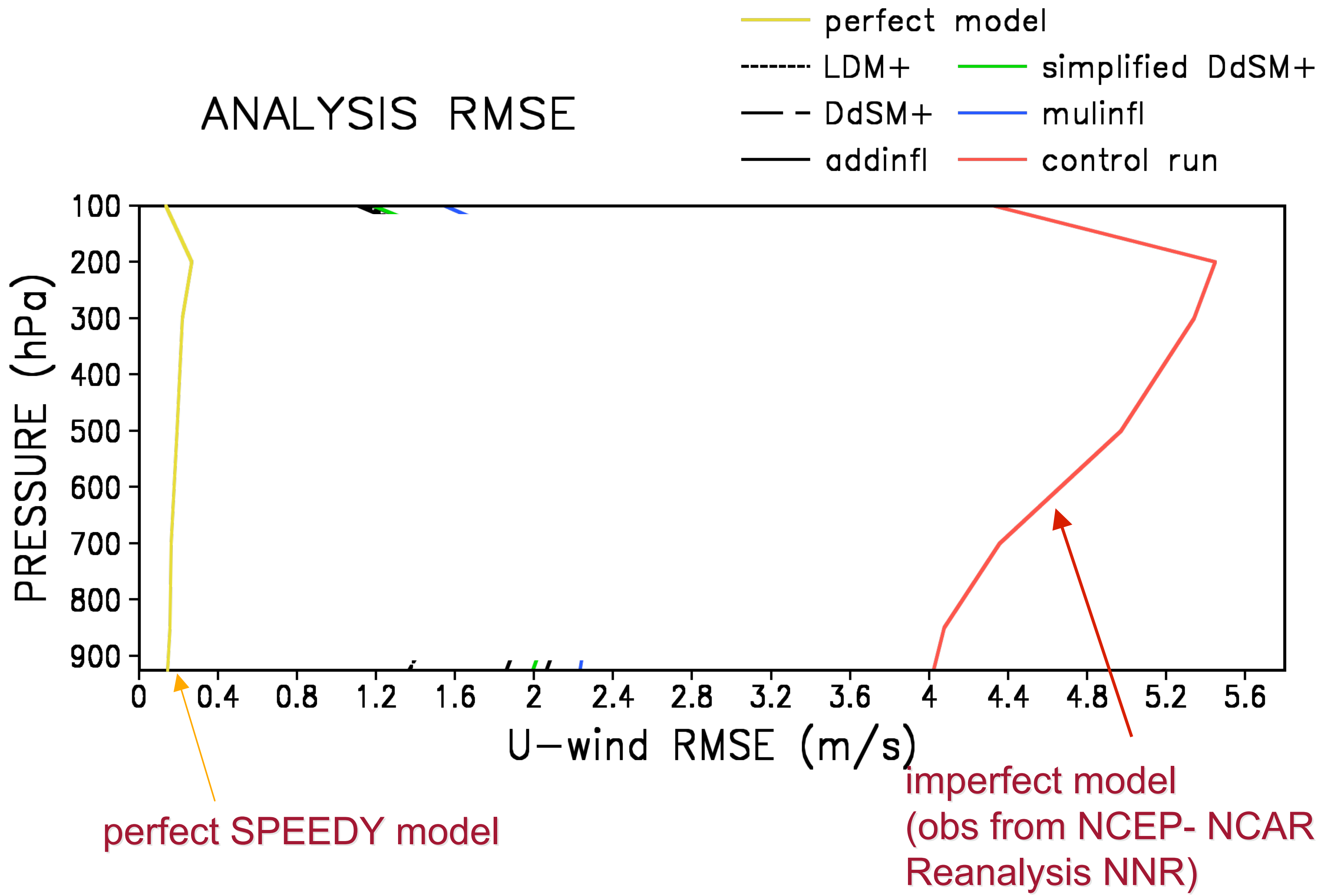
With increment interpolation, the analysis degrades

With weight interpolation, there is no degradation!

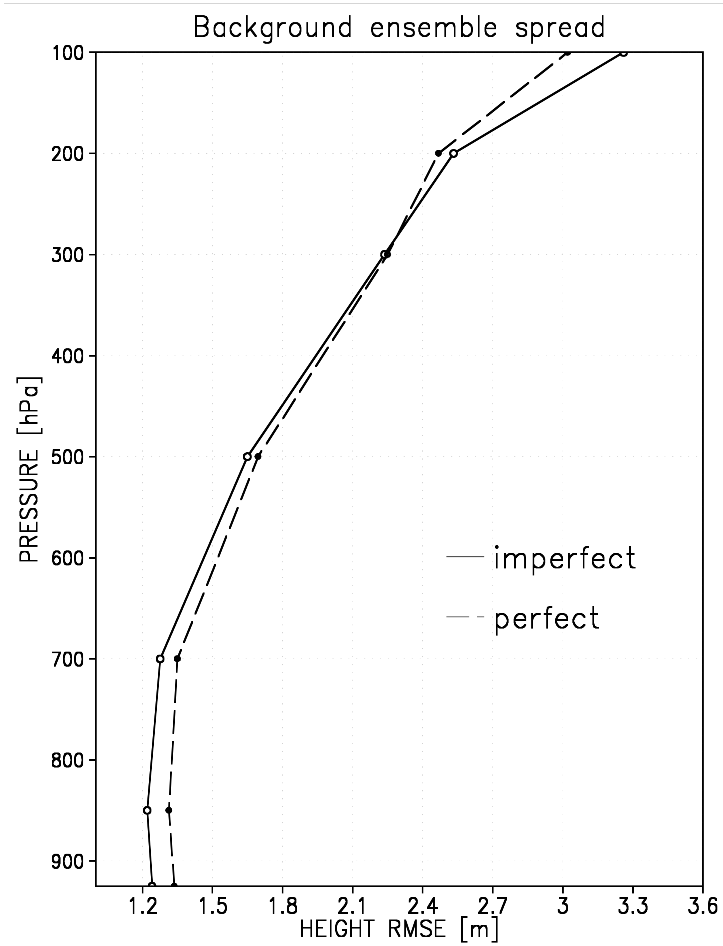
Model error: comparison of methods to correct model bias and inflation

Hong Li, Chris Danforth, Takemasa Miyoshi,
and Eugenia Kalnay

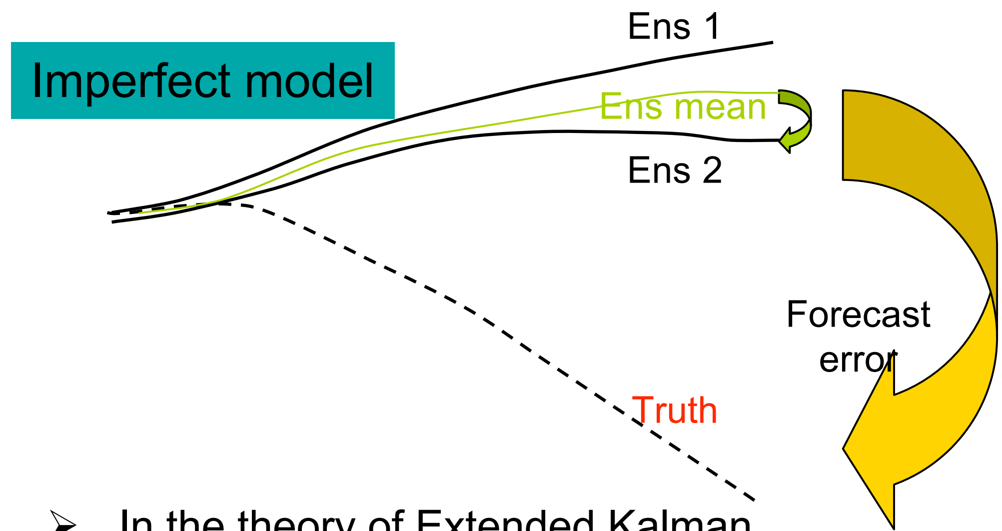
Model error: If we assume a perfect model in EnKF, we underestimate the analysis errors (Li, 2007)



— Why is EnKF vulnerable to model errors ?



The ensemble spread is 'blind' to model errors



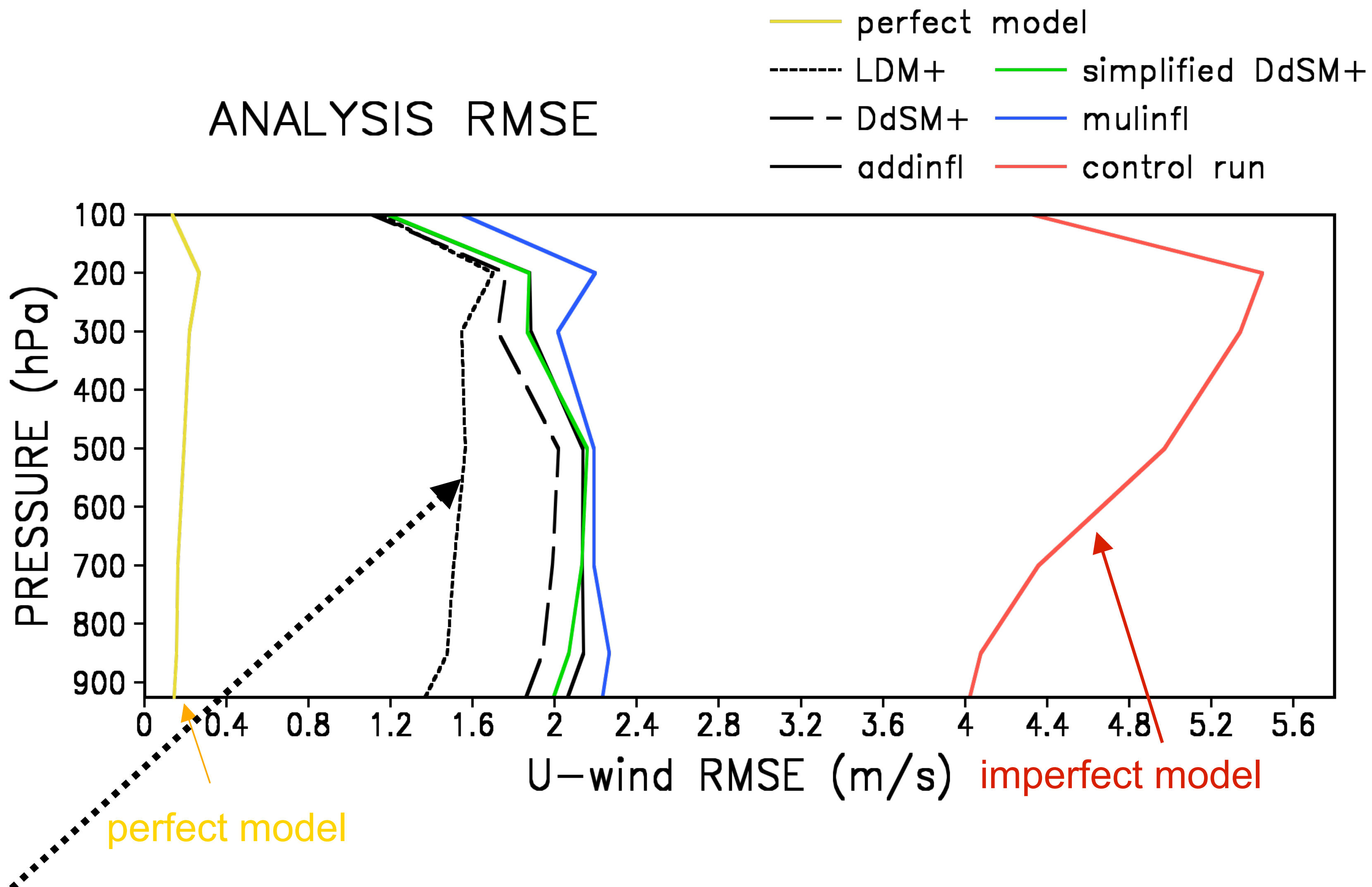
- In the theory of Extended Kalman filter, forecast error is represented by the growth of errors in IC and the model errors.

$$\mathbf{P}_i^f = \mathbf{M}_{\mathbf{x}_{i-1}^a} \mathbf{P}_{i-1}^a \mathbf{M}_{\mathbf{x}_{i-1}^a}^T + \boxed{\mathbf{Q}}$$

- However, in ensemble Kalman filter, error estimated by the ensemble spread can only represent the first type of errors.

$$\mathbf{P}_i^f \approx \frac{1}{k-1}\sum_{i=1}^{K}(x_i^f - \overline{x^f})(x_i^f - \overline{x^f})^T$$

We compared several methods to handle bias and random model errors

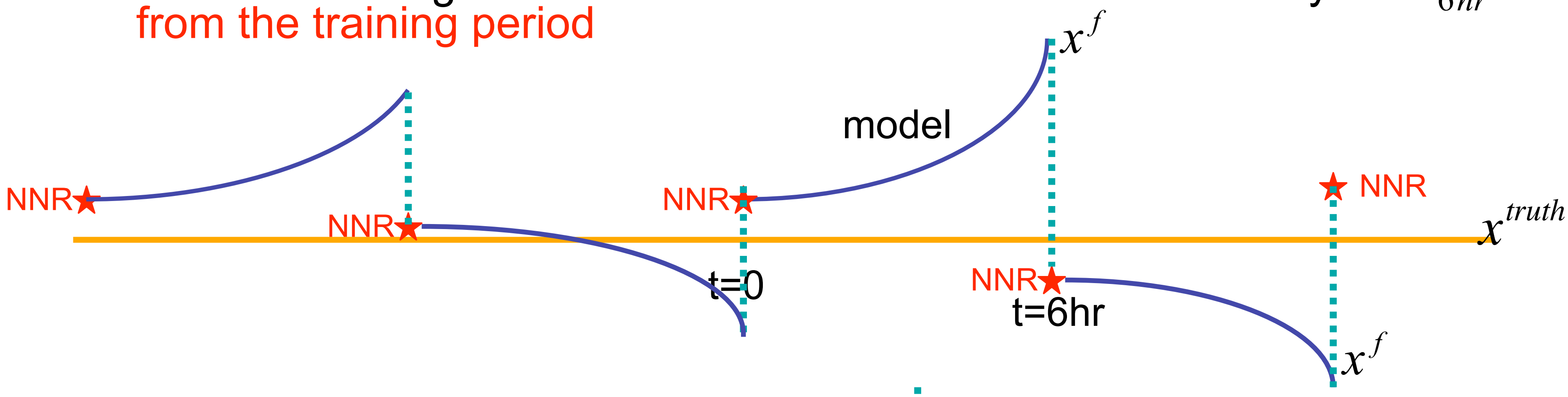


Low Dimensional Method to correct the bias (Danforth et al, 2007) combined with additive inflation

Bias removal schemes (Low Dimensional Method)

2.3 Low-dim method (Danforth et al, 2007: Estimating and correcting global weather model error. *Mon. Wea. Rev, J. Atmos. Sci., 2007*)

- Generate a long time series of model forecast minus reanalysis x^{e}_{6hr} from the training period



We collect a large number of estimated errors and estimate from them bias, etc.

$$\varepsilon^{f}_{n+1} = \mathbf{x}^{f}_{n+1} - \mathbf{x}^{t}_{n+1} = \boxed{M(\mathbf{x}^{a}_{n}) - M(\mathbf{x}^{t}_{n})} + \boxed{\mathbf{b} + \sum_{l=1}^{L} \beta_{n,l}\mathbf{e}_{l} + \sum_{m=1}^{M} \gamma_{n,m}\mathbf{f}_{m}}$$

Forecast error due to error in IC

Time-mean model bias

Diurnal model error

State dependent model error

Low-dimensional method

Include Bias, Diurnal and State-Dependent model errors:

$$\text{model error} = \mathbf{b} + \boxed{\sum_{l=1}^{2} \beta_{n,l} \mathbf{e}_l} + \sum_{m=1}^{10} \gamma_{n,m} \mathbf{f}_m$$

Having a large number of estimated errors allows to estimate the global model error beyond the bias

SPEEDY 6 hr model errors against NNR (diurnal cycle)

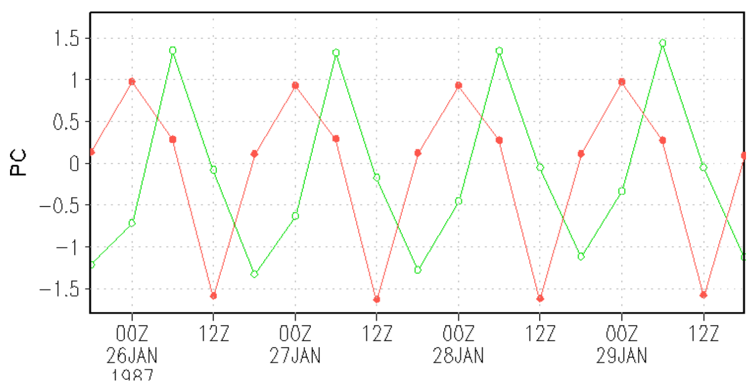
1987 Jan 1~ Feb 15

Error anomalies

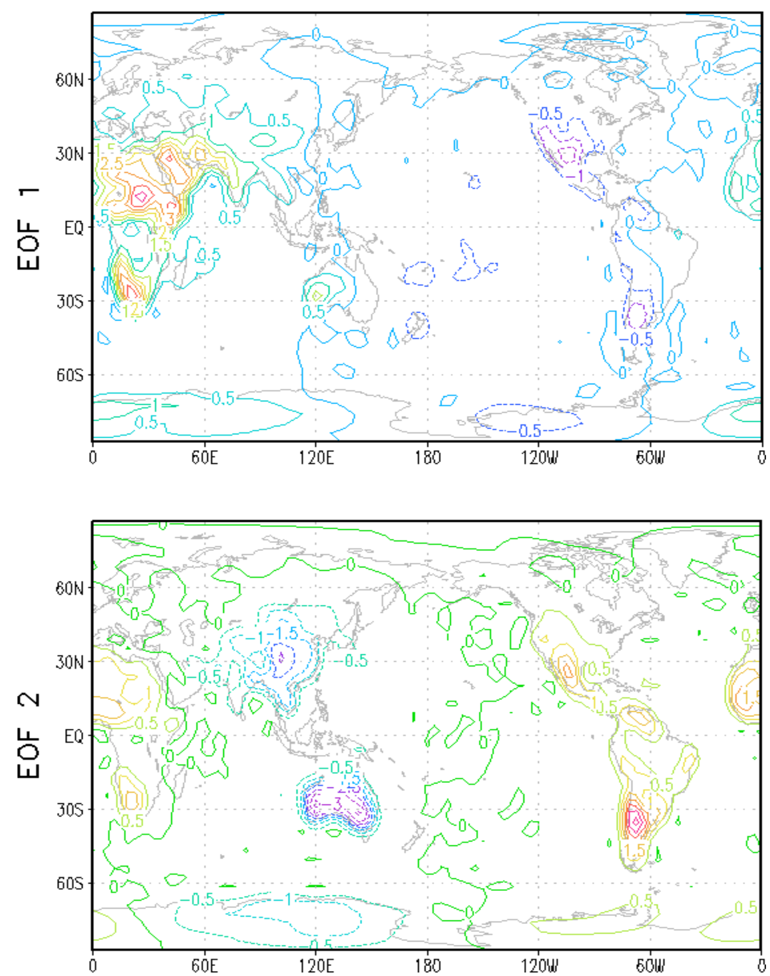
$$x^{e\,'}_{6hr(i)} = x^{e}_{6hr} - \overline{x^{e}_{6hr}}$$

pc1

pc2



Leading EOFs for 925 mb TEMP



- For temperature at lower-levels, in addition to the time-independent bias, SPEEDY has diurnal cycle errors because it lacks diurnal radiation forcing

Discussion: 4D-Var vs. EnKF war

- Correcting the bias with a simple low-dim method (Danforth et al. 2007) and combining it with additive inflation should effectively deal with both systematic and random model errors. But it needs an “unbiased” reanalysis.
- We should be able to adopt some simple strategies to capture the advantages of 4D-Var:
  - Smoothing and running in place
  - A simple outer loop to deal with nonlinearities
  - Adjoint sensitivity without adjoint model
  - Coarse resolution analysis without degradation

- It seems like there is nothing that 4D-Var can do that EnKF cannot do as well, usually simpler, cheaper and better.