

Introduction to Ensemble Kalman Filter

E. Kalnay with Hong Li and T. Miyoshi

University of Maryland

Chaos-Weather Group

B. Hunt, J. Yorke, E.Ott

K. Ide, T. Miyoshi, E. Kalnay,

I. Szunyogh (TAMU), E. Kostelich (ASU) and 16 current
students and Ph.D.s

JCSDA Summer Colloquium 2009

Recall the basic formulation of OI

OI for a scalar (analysis):

$$T_a = T_b + w(T_o - T_b) \quad 0 \leq w \leq 1$$

Optimal weight to minimize the analysis errors: $w = \frac{\sigma_b^2}{\sigma_b^2 + \sigma_o^2}$

Recall the basic formulation of OI

OI for a scalar (analysis):

$$T_a = T_b + w(T_o - T_b) \quad 0 \leq w \leq 1$$

Optimal weight to minimize the analysis errors: $w = \frac{\sigma_b^2}{\sigma_b^2 + \sigma_o^2}$

OI for a vector: $\mathbf{x}_i^b = M\mathbf{x}_{i-1}^a$

$$\mathbf{x}_i^a = \mathbf{x}_i^b + \mathbf{K}(\mathbf{y}_i^o - H\mathbf{x}_i^b)$$

$$\mathbf{K} = \mathbf{P}^b \mathbf{H}^T \left[\mathbf{H} \mathbf{P}^b \mathbf{H}^T + \mathbf{R} \right]^{-1}$$

$$\mathbf{P}^a = [\mathbf{I} - \mathbf{K}\mathbf{H}] \mathbf{P}^b$$

$$\mathbf{P}^b = \delta \mathbf{x}_b \bullet \delta \mathbf{x}_b^T \quad \delta \mathbf{x}_b = \mathbf{x}_b - \mathbf{x}_t$$

In OI \mathbf{P}^b is statistically pre-estimated and constant in time.
Is this a good approximation?

OI and Kalman Filter for a scalar

OI for a scalar: $T_b(t_i) = MT_a(t_{i-1}); \quad \sigma_b^2 = (1+a)\sigma_a^2 = \frac{1}{1-w}\sigma_a^2 = const$

$$w = \frac{\sigma_b^2}{\sigma_b^2 + \sigma_o^2}; \quad \sigma_a^2 = (1-w)\sigma_b^2$$

$$T_a(t_i) = T_b(t_i) + w(T_o(t_i) - T_b(t_i)) \quad 0 \leq w \leq 1$$

OI and Kalman Filter for a scalar

OI for a scalar: $T_b(t_i) = MT_a(t_{i-1}); \quad \sigma_b^2 = (1+a)\sigma_a^2 = \frac{1}{1-w}\sigma_a^2 = const$

$$w = \frac{\sigma_b^2}{\sigma_b^2 + \sigma_o^2}; \quad \sigma_a^2 = (1-w)\sigma_b^2$$

$$T_a(t_i) = T_b(t_i) + w(T_o(t_i) - T_b(t_i)) \quad 0 \leq w \leq 1$$

Kalman Filter

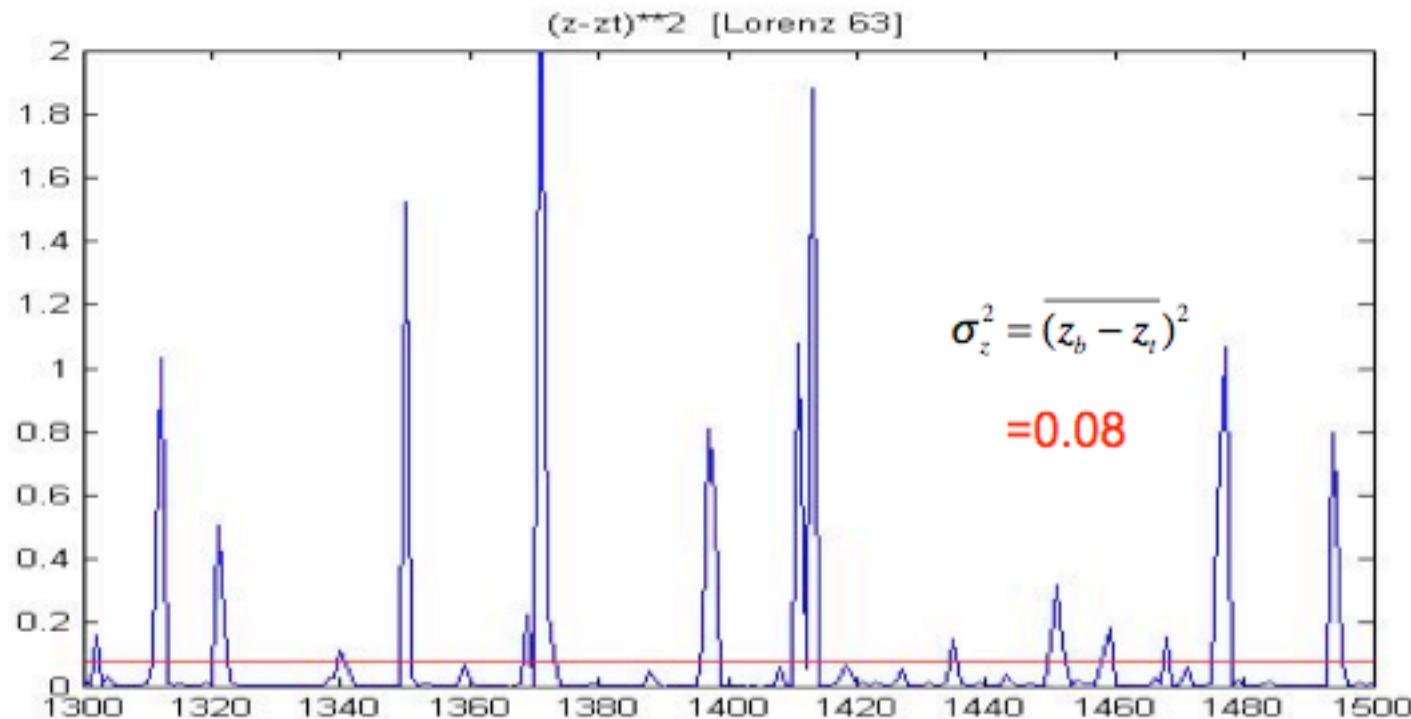
for a scalar: $T_b(t_i) = MT_a(t_{i-1}); \quad \sigma_b^2 = (\mathbf{L}\sigma_a)(\mathbf{L}\sigma_a)^T; \quad \mathbf{L} = dM / dT$

$$w = \frac{\sigma_b^2}{\sigma_b^2 + \sigma_o^2}; \quad \sigma_a^2 = (1-w)\sigma_b^2$$

$$T_a(t_i) = T_b(t_i) + w(T_o(t_i) - T_b(t_i)) \quad 0 \leq w \leq 1$$

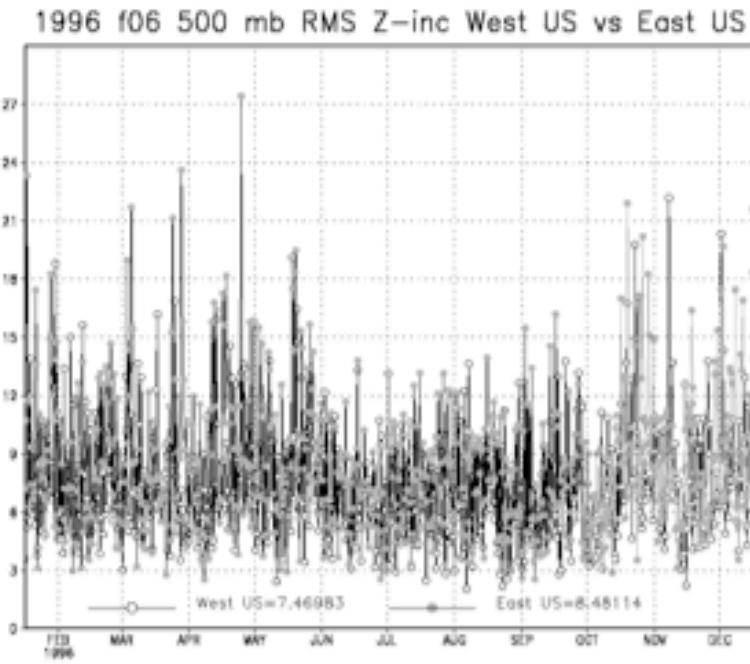
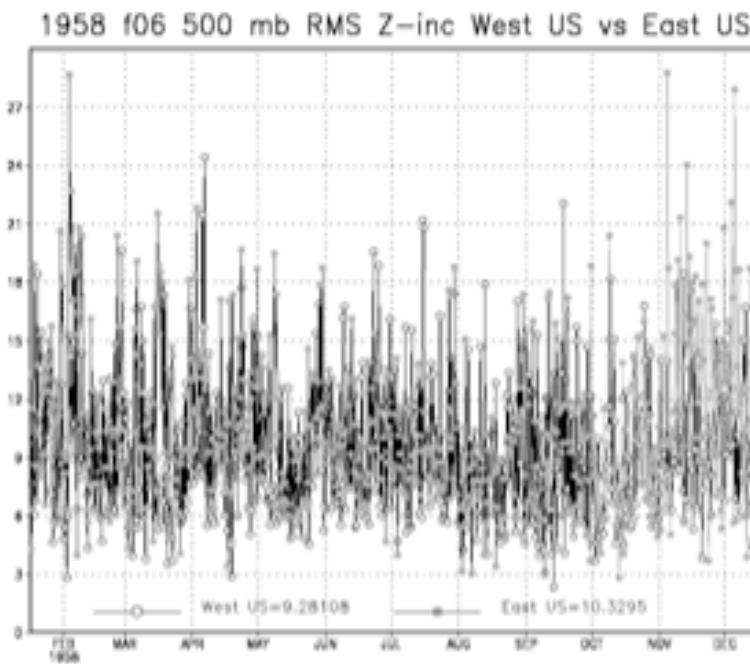
Now the background error variance is **forecasted** using the linear tangent model and its “adjoint”.

“Errors of the day” computed with the Lorenz 3-variable model: compare with rms (constant) error



- Not only the amplitude, but also the **structure of B** is constant in 3D-Var/OI
- This is important because analysis increments occur only within the subspace spanned by B

“Errors of the day” obtained in the reanalysis (figs 5.6.1 and 5.6.2 in the book)



- Note that the mean error went down from 10m to 8m because of improved observing system, but the “errors of the day” (on a synoptic time scale) are still large.
- In 3D-Var/OI not only the amplitude, but also the structure of B is fixed with time

Flow independent error covariance

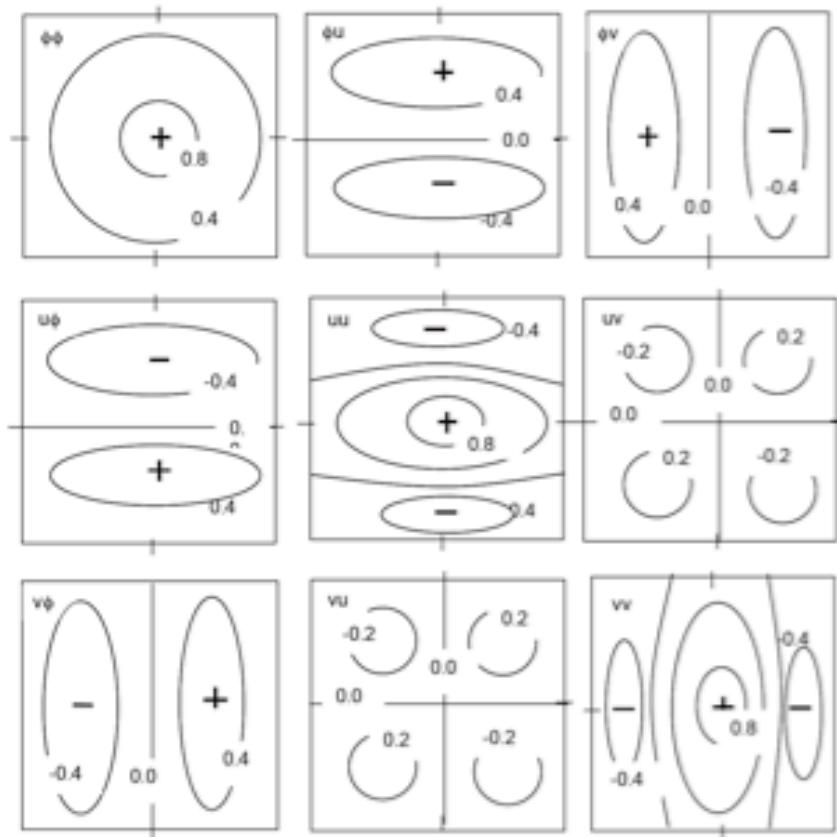


➤ If we observe only Washington, D.C., we can get estimate for Richmond and Philadelphia corrections through the error correlation (off-diagonal term in B).

$$B = \begin{bmatrix} \sigma_1^2 & \text{cov}_{1,2} & \dots & \text{cov}_{1,n} \\ \text{cov}_{1,2} & \sigma_2^2 & & \text{cov}_{2,n} \\ \dots & \dots & \dots & \dots \\ \text{cov}_{1,n} & \text{cov}_{2,n} & \dots & \sigma_n^2 \end{bmatrix}$$

➤ In OI(or 3D-Var), the scalar error correlation between two points in the same horizontal surface is assumed homogeneous and isotropic. (p162 in the book)

Typical 3D-Var error covariance

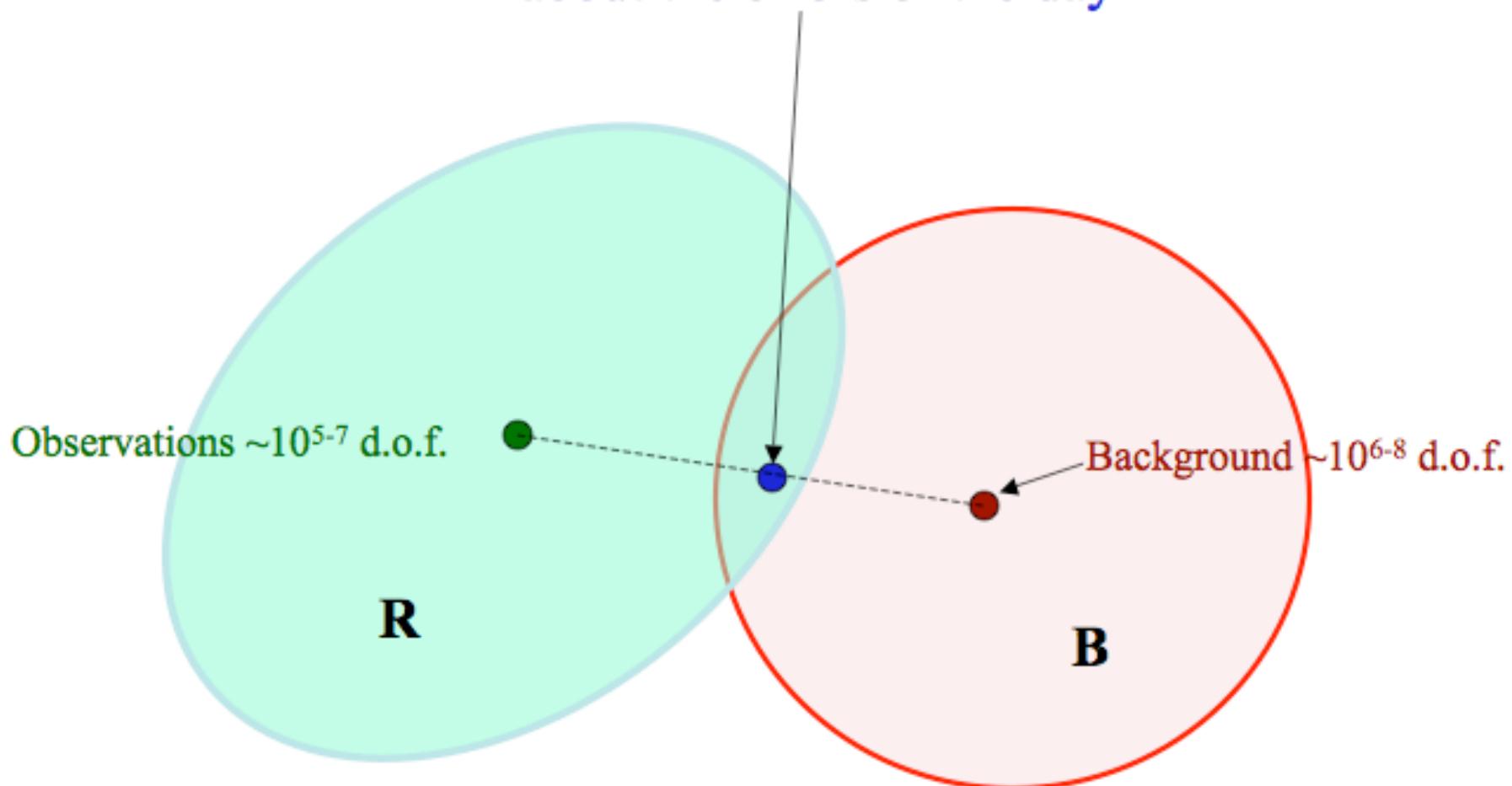


$$B = \begin{bmatrix} \sigma_1^2 & \text{cov}_{1,2} & \dots & \text{cov}_{1,n} \\ \text{cov}_{1,2} & \sigma_2^2 & & \text{cov}_{2,n} \\ \dots & \dots & \dots & \dots \\ \text{cov}_{1,n} & \text{cov}_{2,n} & \dots & \sigma_n^2 \end{bmatrix}$$

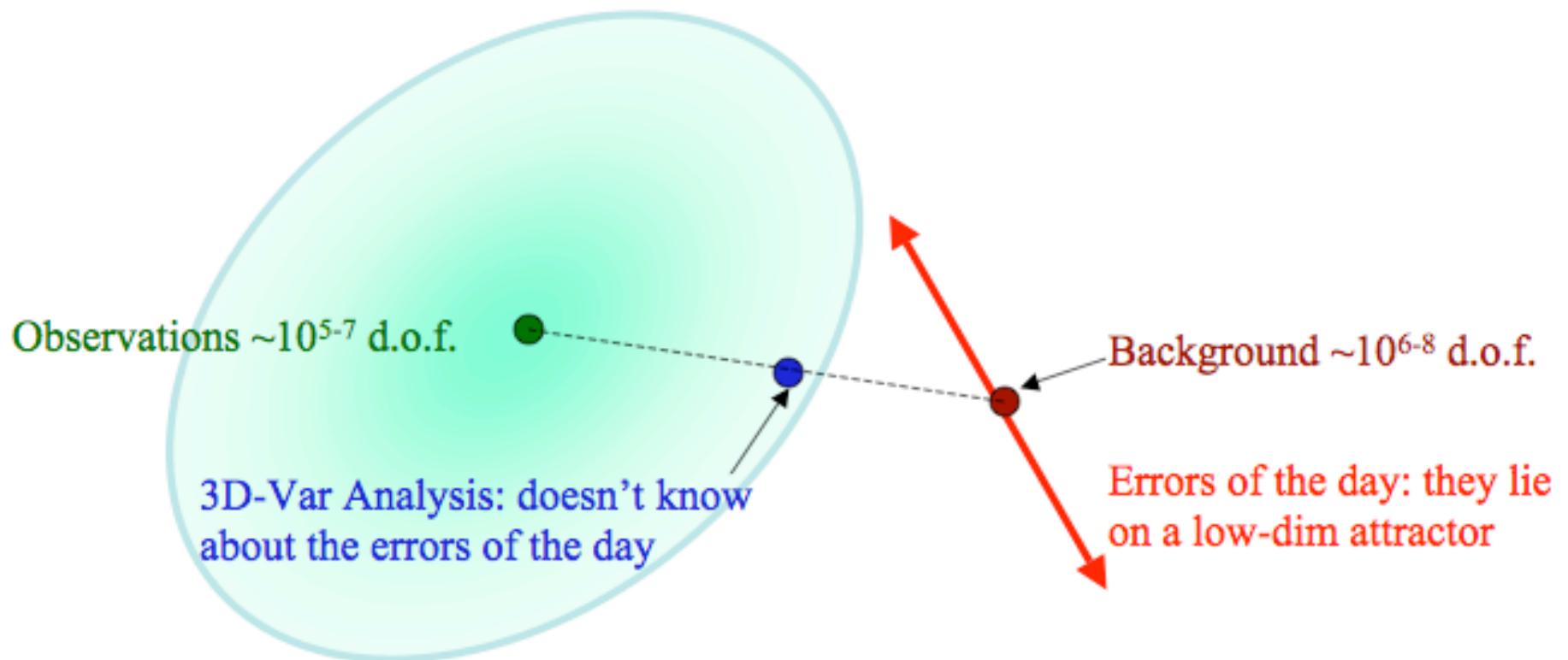
➤ In OI(or 3D-Var), the error correlation between two mass points in the same horizontal surface is assumed **homogeneous and isotropic**. (p162 in the book)

Suppose we have a 6hr forecast (background) and new observations

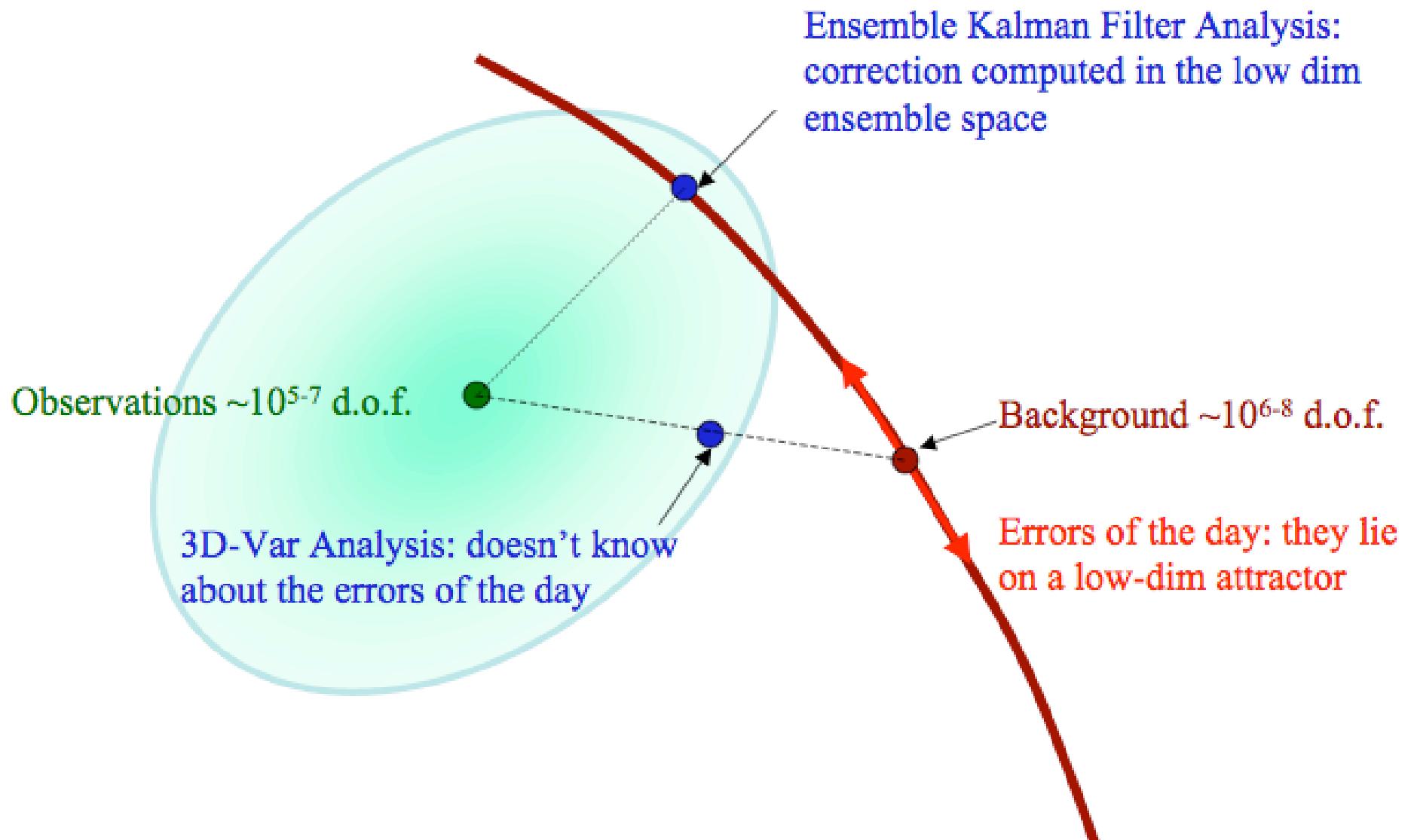
The 3D-Var Analysis doesn't know
about the errors of the day



With Ensemble Kalman Filtering we get perturbations pointing to the directions of the “errors of the day”



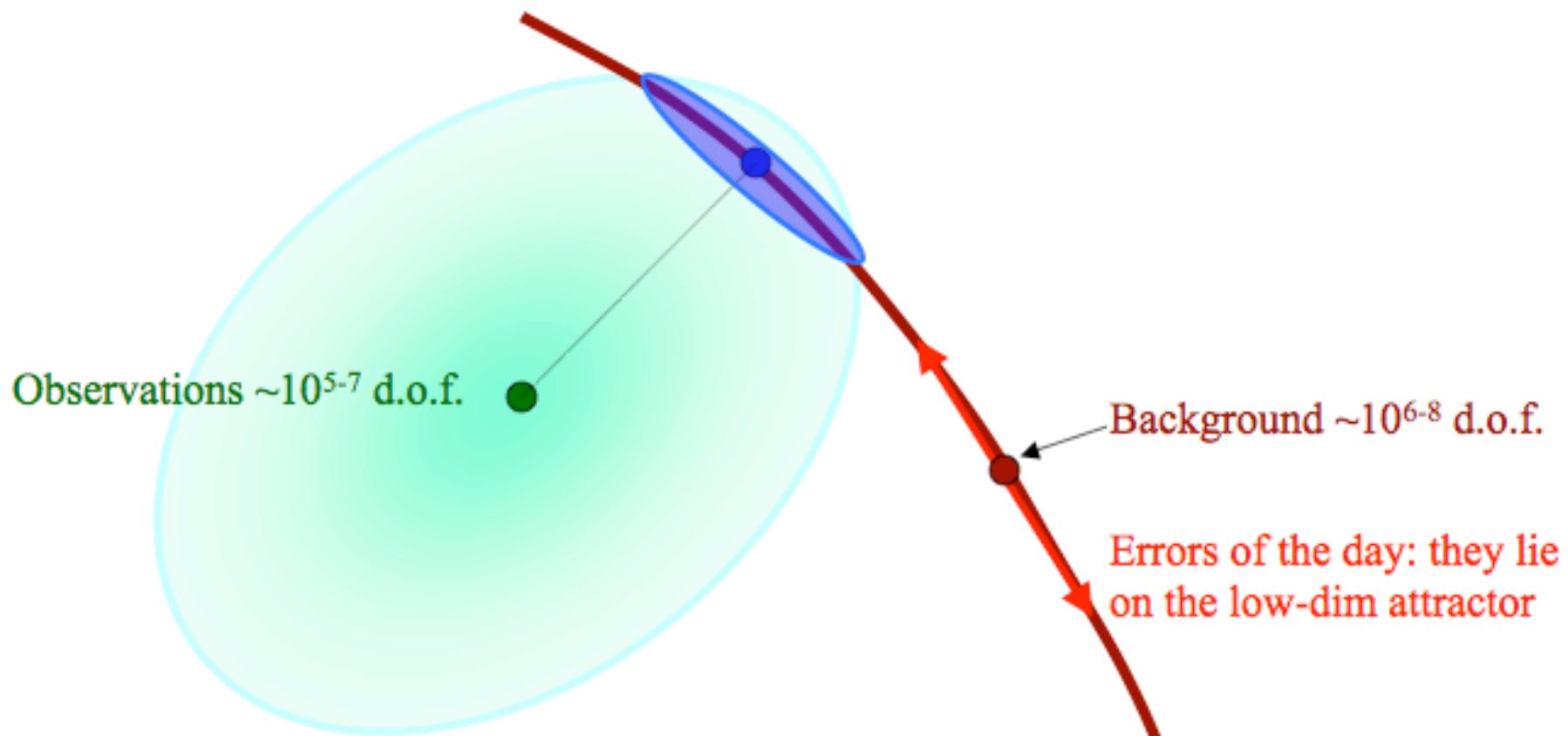
Ensemble Kalman Filtering is efficient because matrix operations are performed in the low-dimensional space of the ensemble perturbations



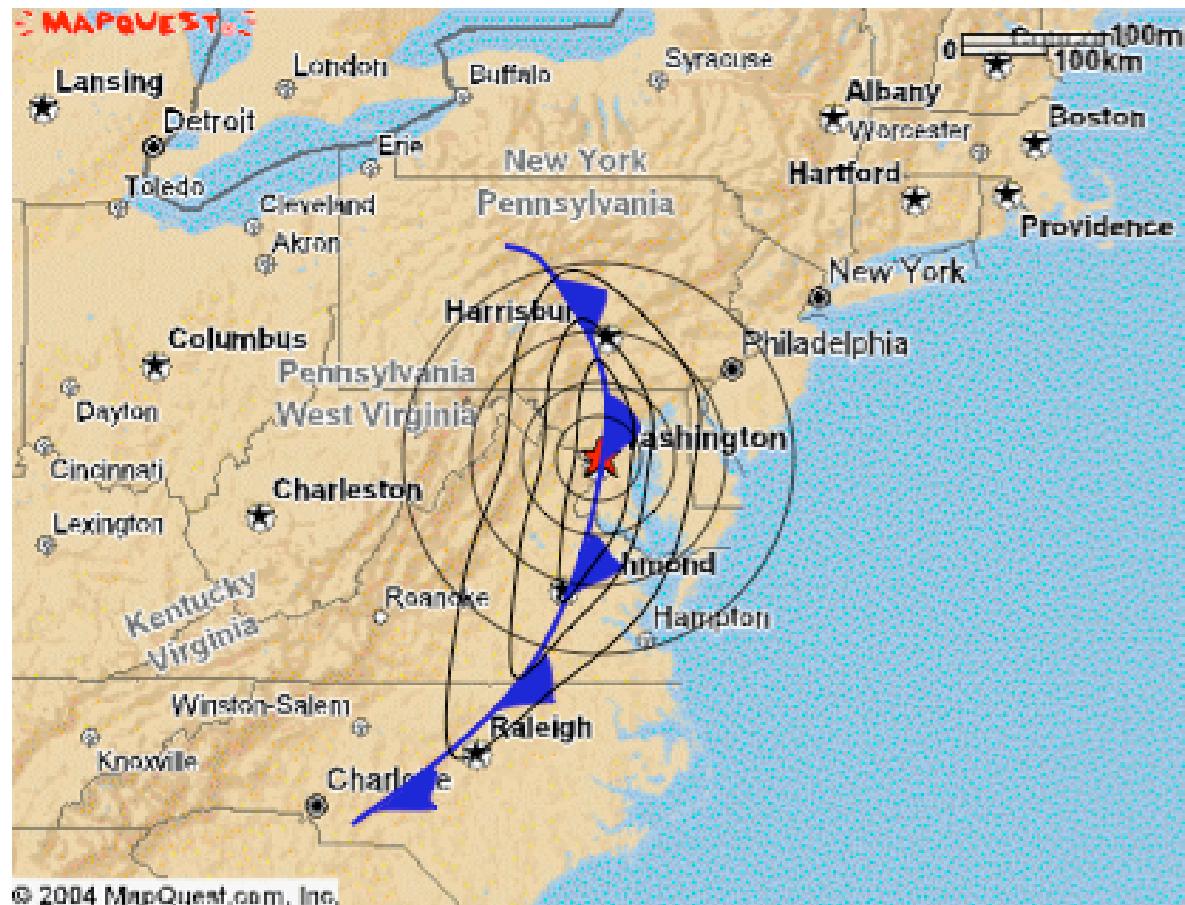
After the EnKF computes the analysis and the analysis error covariance matrix \mathbf{A} , the new ensemble initial perturbations $\delta\mathbf{a}_i$ are computed:

$$\sum_{i=1}^{k+1} \delta\mathbf{a}_i \delta\mathbf{a}_i^T = \mathbf{A}$$

These perturbations represent the analysis error covariance and are used as **initial perturbations** for the next ensemble forecast



Flow-dependence – a simple example (Miyoshi, 20



There is a cold front in our area...

What happens in this case?

This is not appropriate

This does reflects the flow-dependence.

Extended Kalman Filter (EKF)

- Forecast step

$$\mathbf{x}_i^b = M\mathbf{x}_{i-1}^a$$

$$\mathbf{P}_i^b = \mathbf{L}_{i-1}\mathbf{P}_{i-1}^a\mathbf{L}_{i-1}^T + \mathbf{Q} \quad * \text{ (derive it)}$$

- Analysis step

$$\mathbf{x}_i^a = \mathbf{x}_i^b + \mathbf{K}_i(\mathbf{y}_i^o - \mathbf{H}\mathbf{x}_i^f)$$

$$\mathbf{K}_i = \mathbf{P}_i^b \mathbf{H}^T [\mathbf{H} \mathbf{P}_i^b \mathbf{H}^T + \mathbf{R}]^{-1} \quad **$$

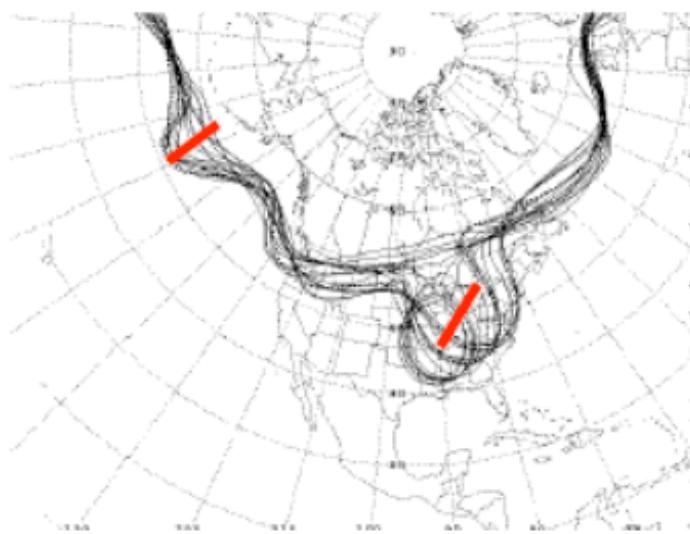
$$\mathbf{P}_{i \times n}^a = [\mathbf{I} - \mathbf{K}_i \mathbf{H}] \mathbf{P}_i^b = [(\mathbf{P}_i^b)^{-1} + \mathbf{H}^T \mathbf{R}^{-1} \mathbf{H}]^{-1}$$

- Using the flow-dependent \mathbf{P}_i^b , analysis is expected to be improved significantly

However, it is computational hugely expensive. \mathbf{P}_i^b , \mathbf{L}_i $n \times n$ matrix, $n \sim 10^7$
computing equation * directly is impossible

Ensemble Kalman Filter (EnKF)

$$\mathbf{P}_i^b = \mathbf{L}_{i-1} \mathbf{P}_{i-1}^a \mathbf{L}_{i-1}^T + \mathbf{Q} \quad *$$



Physically,

- “errors of day” are the instabilities of the background flow. Strong instabilities have **a few dominant shapes** (perturbations lie in a low-dimensional subspace).
- It makes sense to assume that large errors are in similarly low-dimensional spaces that can be represented by a low order EnKF.

$$\mathbf{P}_i^b = \frac{1}{m-1} \sum_{k=1}^m (\mathbf{x}_k^f - \mathbf{x}^t)_i (\mathbf{x}_k^f - \mathbf{x}^t)_i^T, \quad m \rightarrow \infty$$

We use the ensemble to estimate *

$$\begin{aligned} \mathbf{P}_i^b &\approx \frac{1}{K-1} \sum_{k=1}^m (\mathbf{x}_k^f - \bar{\mathbf{x}}^f)_i (\mathbf{x}_k^f - \bar{\mathbf{x}}^f)_i^T \\ &= \frac{1}{K-1} \mathbf{X}_i^b \mathbf{X}_i^{bT} \end{aligned}$$

Problem left: how to estimate the analysis error covariance and the analysis perturbations?

$$\mathbf{P}_i^a = \frac{1}{K-1} \mathbf{X}_i^a \mathbf{X}_i^{aT}$$

Ensemble Update: two approaches

1. Perturbed Observations method:

An “ensemble of data assimilations”

- It has been proven that an **observational ensemble** is required (e.g., Burgers et al. 1998). Otherwise $\mathbf{P}_i^a \in \mathbb{R}^{n \times n} = [\mathbf{I} - \mathbf{K}_i \mathbf{H}] \mathbf{P}_i^b$ is not satisfied.

- Random perturbations are added to the observations to obtain observations for each independent cycle

$$\mathbf{y}_{i(k)}^o = \mathbf{y}_i^o + noise$$

$$\mathbf{x}_{i(k)}^b = M\mathbf{x}_{i-1(k)}^a$$

$$\mathbf{P}_i^b \approx \frac{1}{K-1} \sum_{k=1}^K (\mathbf{x}_k^b - \bar{\mathbf{x}}^b)(\mathbf{x}_k^b - \bar{\mathbf{x}}^b)^T$$

$$\mathbf{K}_i = \mathbf{P}_i^b \mathbf{H}^T [\mathbf{H} \mathbf{P}_i^b \mathbf{H}^T + \mathbf{R}]^{-1}$$

$$\mathbf{x}_{i(k)}^a = \mathbf{x}_{i(k)}^b + \mathbf{K}_i (\mathbf{y}_{i(k)}^o - H\mathbf{x}_{i(k)}^b)$$

- However, it introduces a source of sampling errors when perturbing observations.

Ensemble Update: two approaches

2. Ensemble square root filter (EnSRF)

- Observations are assimilated to update only the ensemble mean.

$$\overline{\mathbf{x}}^a = \overline{\mathbf{x}}^b + \mathbf{K}_i(\mathbf{y}_i^o - H\overline{\mathbf{x}}^b)$$

- Assume analysis ensemble perturbations can be formed by transforming the forecast ensemble perturbations through a transform matrix

$$\begin{aligned}\mathbf{x}_i^b &= M\mathbf{x}_{i-1}^a \\ \mathbf{P}_i^b &\approx \frac{1}{K-1} \sum_{k=1}^K (\mathbf{x}_k^b - \overline{\mathbf{x}}^b)(\mathbf{x}_k^b - \overline{\mathbf{x}}^b)^T \\ \mathbf{K}_i &= \mathbf{P}_i^b \mathbf{H}^T [\mathbf{H} \mathbf{P}_i^b \mathbf{H}^T + \mathbf{R}]^{-1}\end{aligned}$$

$$\overline{\mathbf{x}}^a = \overline{\mathbf{x}}^b + \mathbf{K}_i(\mathbf{y}_i^o - H\overline{\mathbf{x}}^b)$$

$$\mathbf{X}_i^a = \mathbf{T}_i \mathbf{X}_i^b$$

$$\mathbf{x}_i^a = \overline{\mathbf{x}}_i^a + \mathbf{X}^a$$

$$\frac{1}{k-1} \mathbf{X}^a \mathbf{X}^{a^T} = \mathbf{P}_i^a \underset{n \times n}{=} [\mathbf{I} - \mathbf{K}_i \mathbf{H}] \mathbf{P}_i^b = [\mathbf{I} - \mathbf{K}_i \mathbf{H}] \frac{1}{k-1} \mathbf{X}^b \mathbf{X}^{b^T} \Rightarrow \mathbf{X}_i^a = \mathbf{T}_i \mathbf{X}_i^b$$

Several choices of the transform matrix

- EnSRF, Andrews 1968, Whitaker and Hamill, 2002)

$$\mathbf{X}^a = (\mathbf{I} - \tilde{\mathbf{K}}\mathbf{H})\mathbf{X}^b, \tilde{\mathbf{K}} = \alpha\mathbf{K}$$

- EAKF (Anderson 2001)

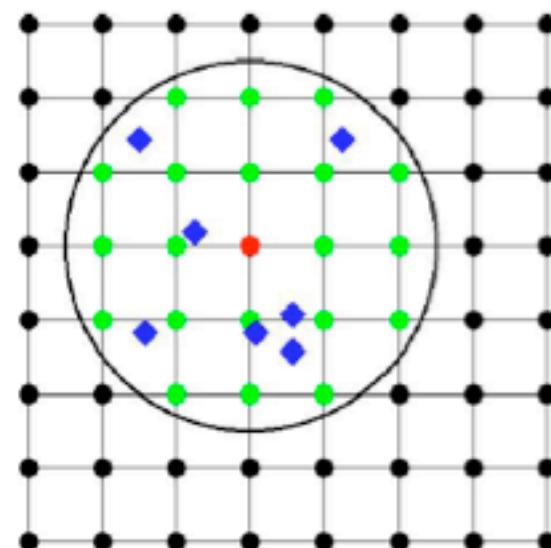
$$\mathbf{X}^a = \mathbf{A}\mathbf{X}^b$$

- ETKF (Bishop et al. 2001)

$$\mathbf{X}^a = \mathbf{X}^b \mathbf{T}$$

- LETKF (Hunt, 2005)

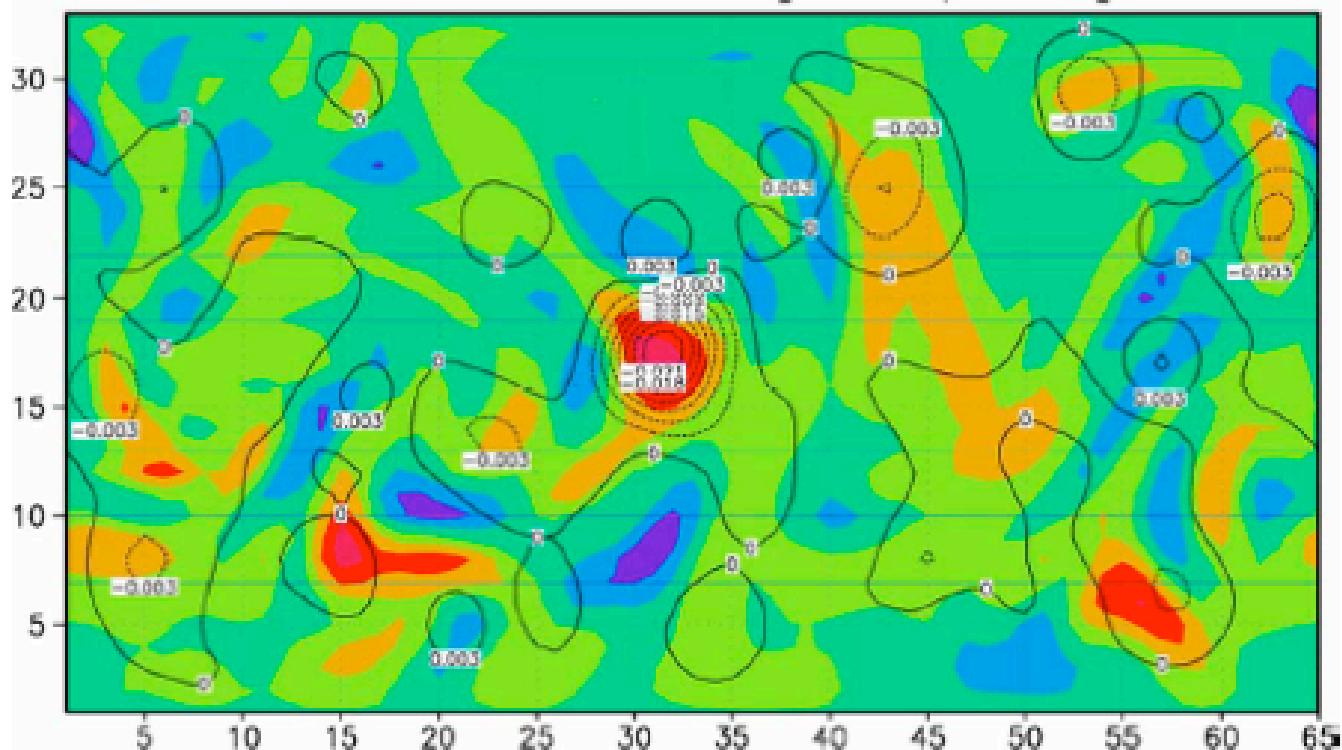
Based on ETKF but perform analysis simultaneously in a local volume surrounding each grid point



An Example of the analysis corrections 3D-Var (Kalnay, 2004)

An example with the QG system (Corazza et al, 2003)

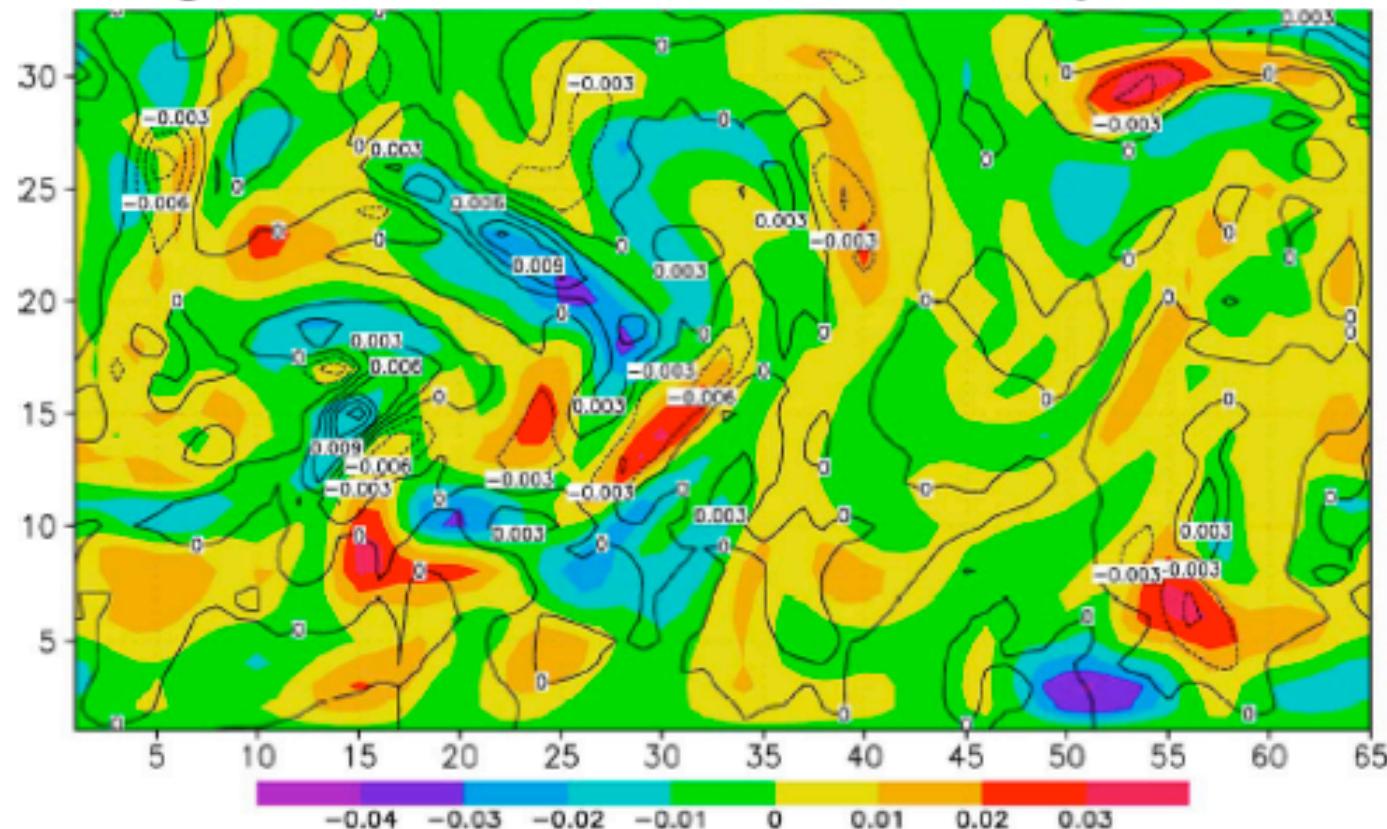
Background error (color) and 3D-Var analysis correction (conto



An Example of the analysis corrections from EnKF (Kalnay, 2004)

QG model example of Local Ensemble KF (Corazza et al)

Background error (color) and LEKF analysis correction

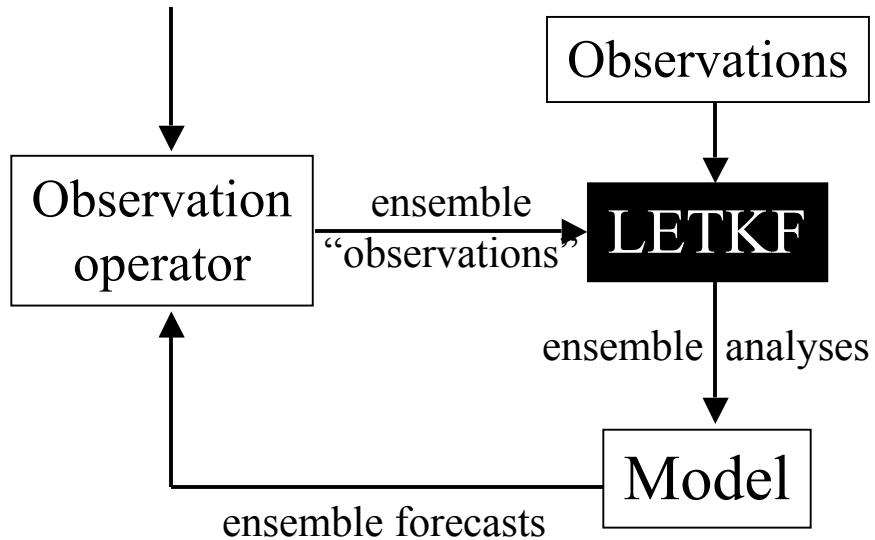


The LEKF does better because it captures the errors of the day

Local Ensemble Transform Kalman Filter

(Ott et al, 2004, Hunt et al, 2004, 2007)

(Start with initial ensemble)

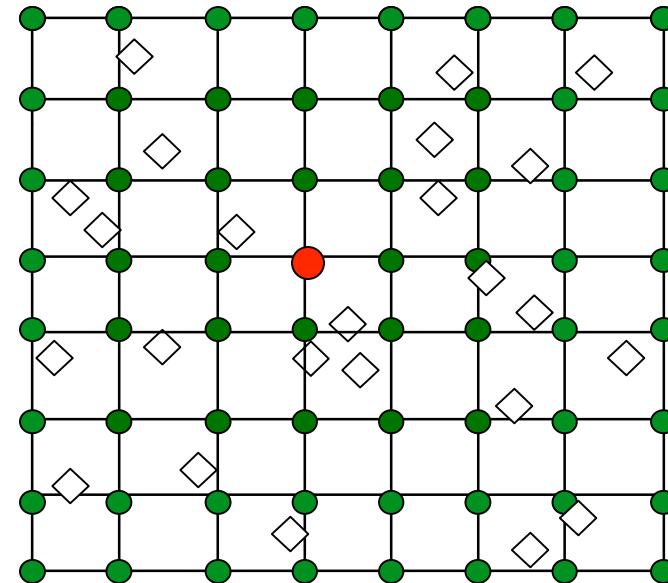


- Model independent (black box)
- Obs. assimilated **simultaneously** at each grid point
- 100% parallel: fast
- No **adjoint** needed
- **4D LETKF extension**

Localization based on observations

Perform data assimilation in a local volume, choosing observations

The state estimate is updated at the central grid **red** dot

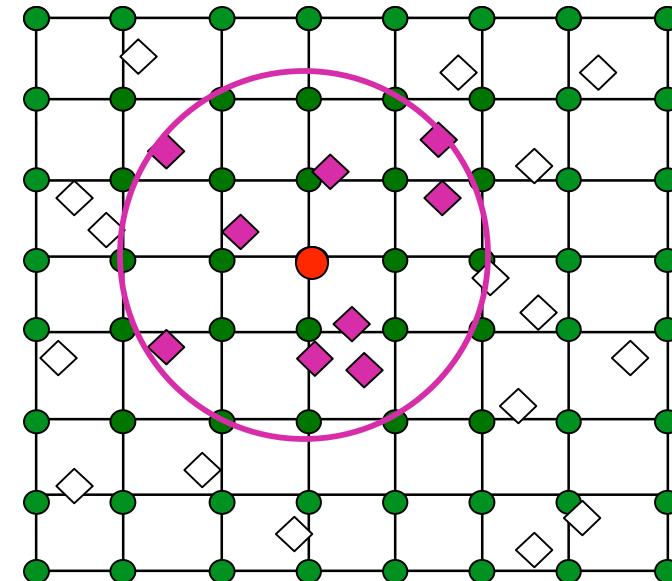


Localization based on observations

Perform data assimilation in a local volume, choosing observations

The state estimate is updated at the central grid **red** dot

All observations (**purple** diamonds) within the local region are assimilated



The LETKF algorithm can be described **in a single slide**

Local Ensemble Transform Kalman Filter (LETKF)

Globally:

Forecast step: $\mathbf{x}_{n,k}^b = M_n(\mathbf{x}_{n-1,k}^a)$

Analysis step: construct $\mathbf{X}^b = [\mathbf{x}_1^b - \bar{\mathbf{x}}^b | \dots | \mathbf{x}_K^b - \bar{\mathbf{x}}^b];$

$$\mathbf{y}_i^b = H(\mathbf{x}_i^b); \mathbf{Y}_n^b = [\mathbf{y}_1^b - \bar{\mathbf{y}}^b | \dots | \mathbf{y}_K^b - \bar{\mathbf{y}}^b]$$

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

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

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 $\mathbf{X}_n^a = \mathbf{X}_n^b \mathbf{W}^a + \bar{\mathbf{x}}^b$. Gathering the grid point analyses forms the new global analyses. Note that the output of the LETKF are analysis weights $\bar{\mathbf{w}}^a$ and perturbation analysis matrices of weights \mathbf{W}^a . These weights multiply the ensemble forecasts.

Repeat: LETKF, globally, with notes (4D-LETKF)

Globally:

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

Analysis step: construct $\mathbf{X}^b = \left[\mathbf{x}_1^b - \bar{\mathbf{x}}^b \mid \dots \mid \mathbf{x}_K^b - \bar{\mathbf{x}}^b \right]$;

$$\mathbf{y}_i^b = H(\mathbf{x}_i^b); \mathbf{Y}_n^b = \left[\mathbf{y}_1^b - \bar{\mathbf{y}}^b \mid \dots \mid \mathbf{y}_K^b - \bar{\mathbf{y}}^b \right]$$

Notes:

$$\mathbf{P}^b = \mathbf{X}^b \mathbf{X}^{bT}$$

$$\mathbf{Y}^b \approx \mathbf{H} \mathbf{X}^b, \text{ but it is computed nonlinearly}$$

4D-LETKF is done by simply concatenating in the vertical the vectors \mathbf{y}^o , $\bar{\mathbf{y}}^b$ and matrix \mathbf{Y}^b valid at different times in the assimilation window. The rest is the same.

Repeat: LETKF, locally, with notes

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

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

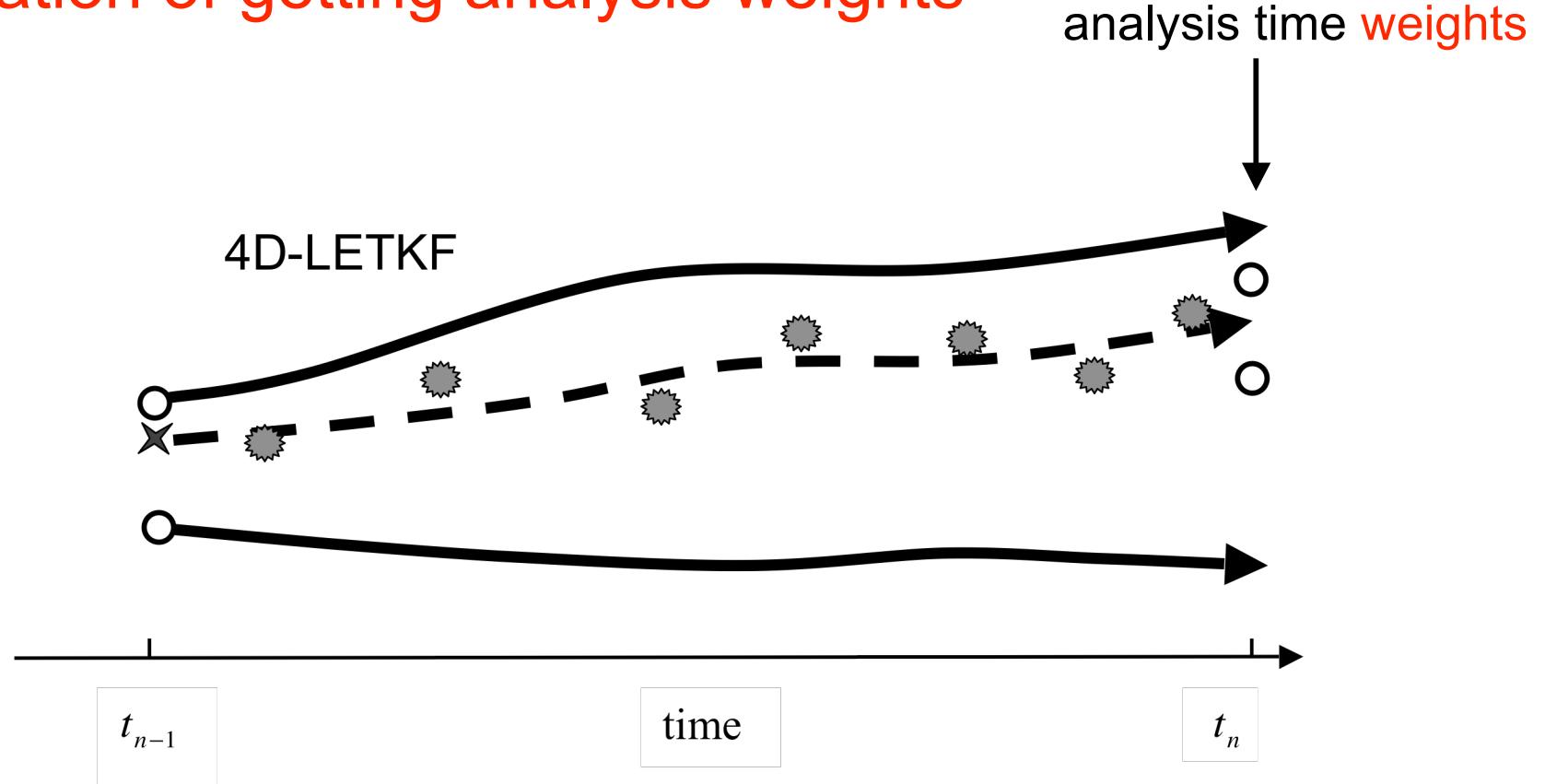
Analysis mean in ensemble space: $\bar{\mathbf{w}}^a = \tilde{\mathbf{P}}^a \mathbf{Y}^{bT} \mathbf{R}^{-1} (\mathbf{y}^o - \bar{\mathbf{y}}^b)$
and add to \mathbf{W}^a to get the analysis ensemble in ensemble space

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

Notes: $\tilde{\mathbf{P}}^a = \left[(K - 1) \mathbf{I} + \mathbf{Y}^{bT} \mathbf{R}^{-1} \mathbf{Y}^b \right]^{-1}$ is $\mathbf{P}^{a-1} = \left[\mathbf{P}^{b-1} + \mathbf{H}^T \mathbf{R}^{-1} \mathbf{H} \right]$
in ensemble space, where $\tilde{\mathbf{P}}^{b-1} = \mathbf{I} / (K - 1)$ is the unit matrix/(K-1)!

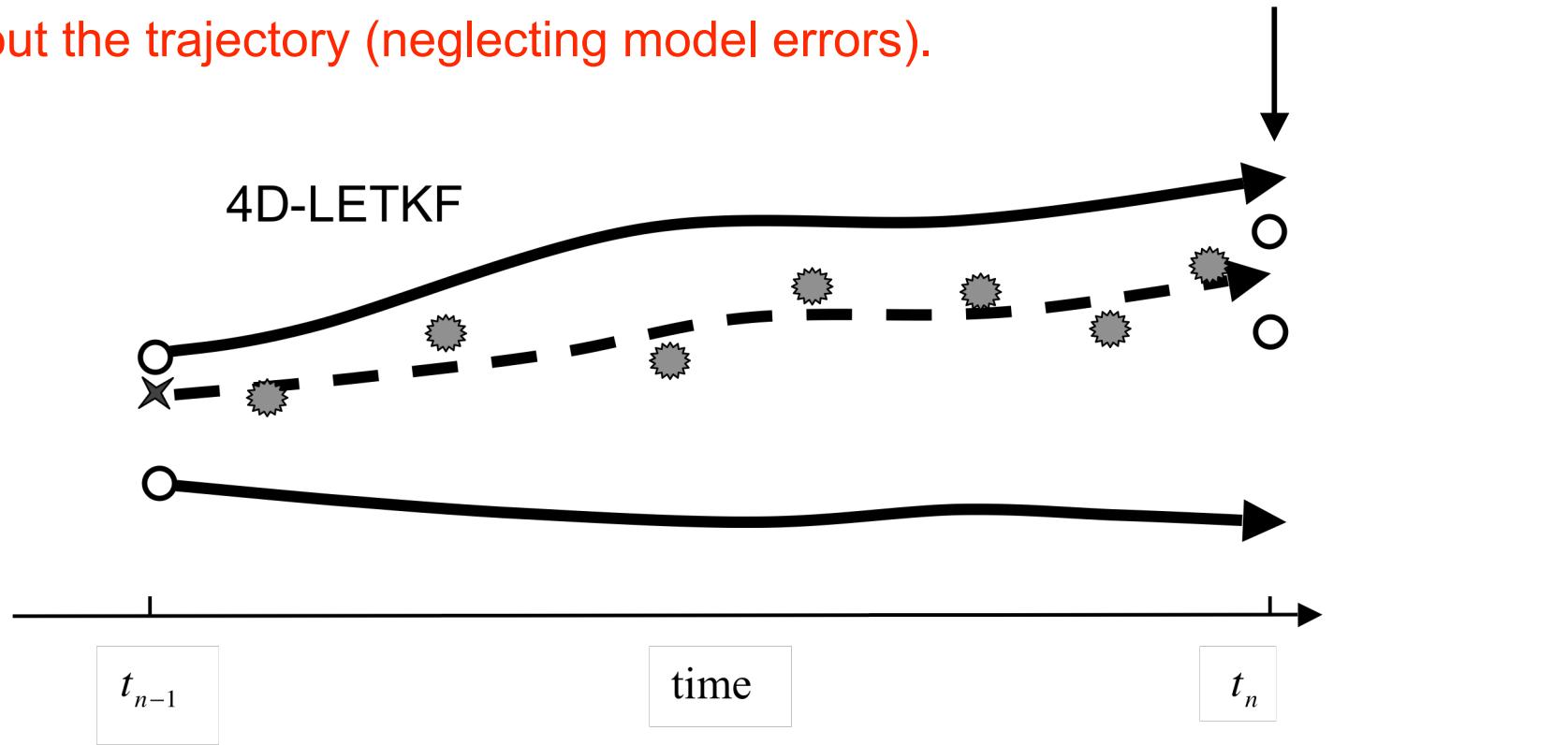
Note that here, the columns of \mathbf{X}^a are the full analysis ensemble members (not the perturbations).

Implication of getting analysis weights



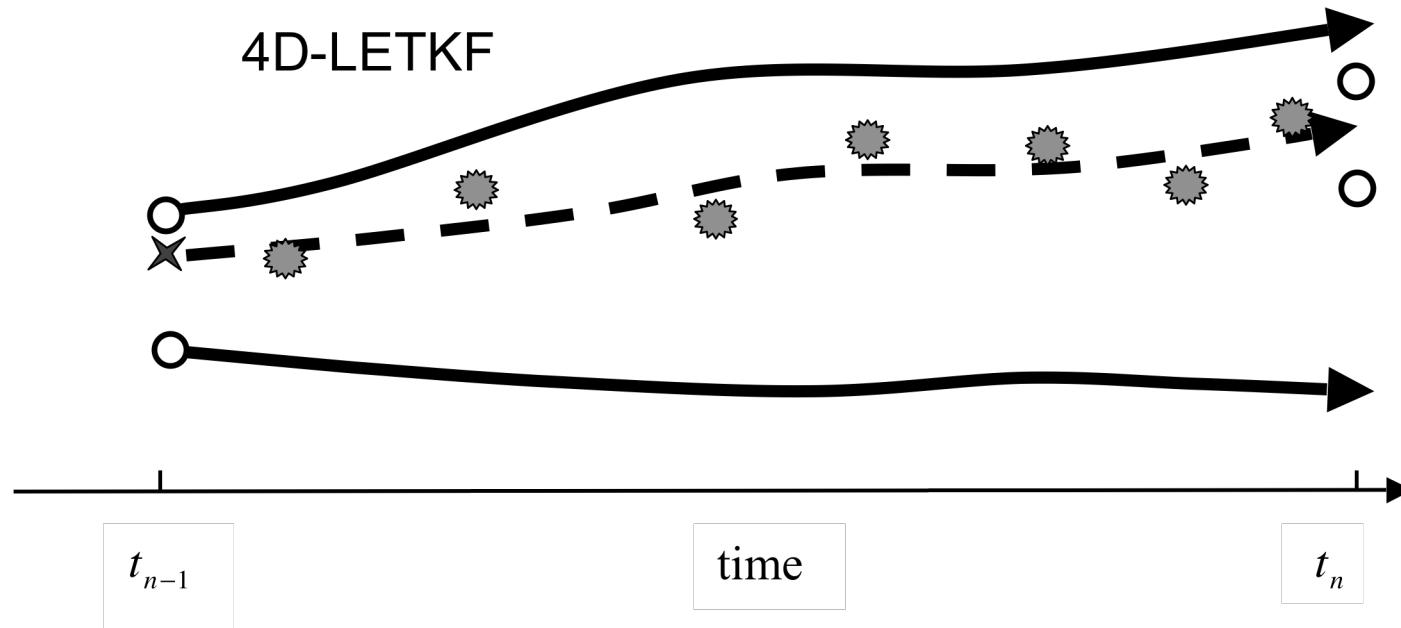
The 4D-LETKF produces an analysis in terms of **weights** of the ensemble forecast members at the analysis time t_n , giving the **trajectory** that best fits **all the observations** in the assimilation window.

A linear comb. of trajectories is ~a trajectory. If it is close to the truth at the end it should be close to the truth throughout the trajectory (neglecting model errors).



The 4D-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.

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



The no-cost smoother makes possible:

- ✓ Outer loop (like in 4D-Var)
- ✓ “Running in place” (faster spin-up)
- ✓ Use of future data in reanalysis
- ✓ Ability to use longer windows

No-cost LETKF smoother tested on a QG model: It works!

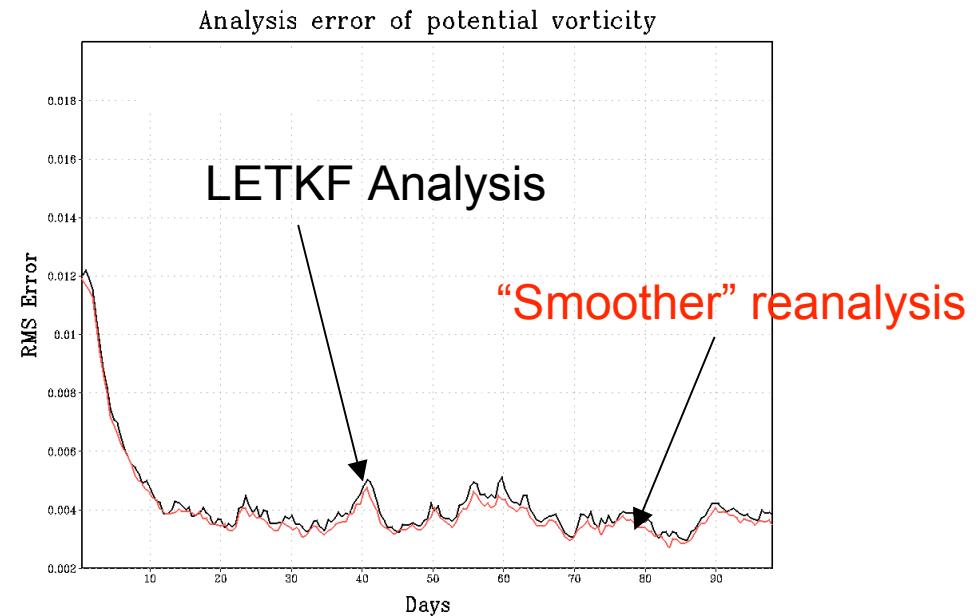
LETKF analysis
at time n

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



Smoother analysis
at time $n-1$

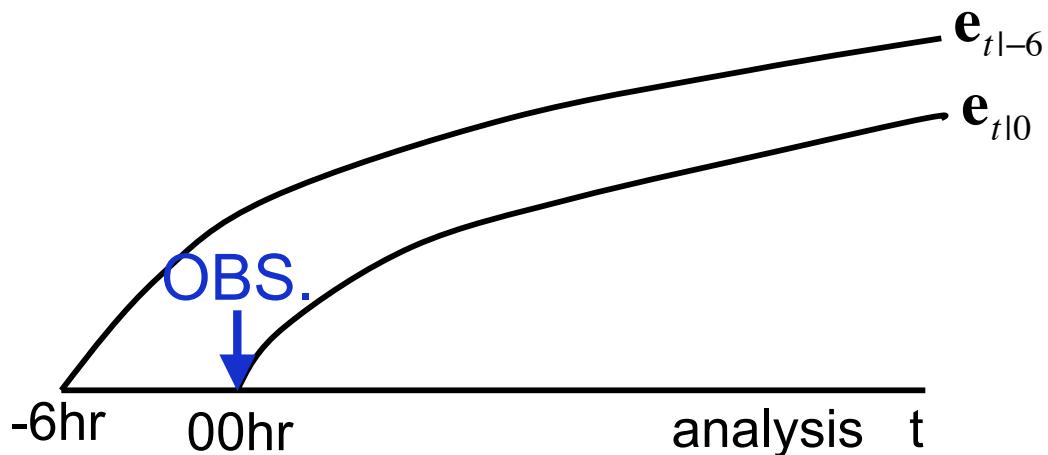
$$\tilde{\mathbf{x}}_{n-1}^a = \bar{\mathbf{x}}_{n-1}^f + \mathbf{X}_{n-1}^f \bar{\mathbf{w}}_n^a$$



This very simple smoother allows us to go back and forth in time within an assimilation window:
it allows assimilation of future data in reanalysis

Example: forecast sensitivity to observations

Liu, Hong and Kalnay, in preparation



$$\mathbf{e}_{t|0} = \bar{\mathbf{x}}_{t|0}^f - \bar{\mathbf{x}}_t^a$$

(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:

$$\Delta \mathbf{e}^2 = (\mathbf{e}_{t|0}^T \mathbf{e}_{t|0} - \mathbf{e}_{t|-6}^T \mathbf{e}_{t|-6})$$

Example: forecast sensitivity to observations

$$\begin{aligned}\Delta \mathbf{e}^2 &= (\mathbf{e}_{t|0}^T \mathbf{e}_{t|0} - \mathbf{e}_{t|-6}^T \mathbf{e}_{t|-6}) = (\mathbf{e}_{t|0}^T - \mathbf{e}_{t|-6}^T)(\mathbf{e}_{t|0} + \mathbf{e}_{t|-6}) \\ &= (\bar{\mathbf{x}}_{t|0}^f - \bar{\mathbf{x}}_{t|-6}^f)^T (\mathbf{e}_{t|0} + \mathbf{e}_{t|-6}) \\ &= [\mathbf{M}(\bar{\mathbf{x}}_0^a - \bar{\mathbf{x}}_{0|-6}^b)]^T (\mathbf{e}_{t|0} + \mathbf{e}_{t|-6}), \text{ so that}\end{aligned}$$

$$\Delta \mathbf{e}^2 = [\mathbf{MK}(\mathbf{y} - H(\mathbf{x}_{0|-6}^b))]^T (\mathbf{e}_{t|0} + \mathbf{e}_{t|-6})$$

Langland and Baker (2004) solve this with the adjoint:

$$\begin{aligned}\Delta \mathbf{e}^2 &= [\mathbf{MK}(\mathbf{y} - H(\mathbf{x}_{0|-6}^b))]^T (\mathbf{e}_{t|0} + \mathbf{e}_{t|-6}) \\ &= [(\mathbf{y} - H(\mathbf{x}_{0|-6}^b))]^T \mathbf{K}^T \mathbf{M}^T (\mathbf{e}_{t|0} + \mathbf{e}_{t|-6})\end{aligned}$$

This requires the adjoint of the model and of the data assimilation system (Langland and Baker, 2004)

Example: forecast sensitivity to observations

Langland and Baker (2004):

$$\begin{aligned}\Delta \mathbf{e}^2 &= \left[\mathbf{M} \mathbf{K} (\mathbf{y} - H(\mathbf{x}_{0|-6}^b)) \right]^T (\mathbf{e}_{t|0} + \mathbf{e}_{t|-6}) \\ &= \left[(\mathbf{y} - H(\mathbf{x}_{0|-6}^b)) \right]^T \mathbf{K}^T \mathbf{M}^T (\mathbf{e}_{t|0} + \mathbf{e}_{t|-6})\end{aligned}$$

With EnKF we can use the original equation without “adjoining”:
Recall that $\mathbf{K} = \mathbf{P}^a \mathbf{H}^T \mathbf{R}^{-1} = \mathbf{X}^a \mathbf{X}^{aT} \mathbf{H}^T \mathbf{R}^{-1}$ so that we can write
 $\mathbf{M} \mathbf{K} = \mathbf{M} \mathbf{X}^a (\mathbf{X}^{aT} \mathbf{H}^T) \mathbf{R}^{-1} = \mathbf{X}_{t|0}^f \mathbf{Y}^{aT} \mathbf{R}^{-1}$ so that

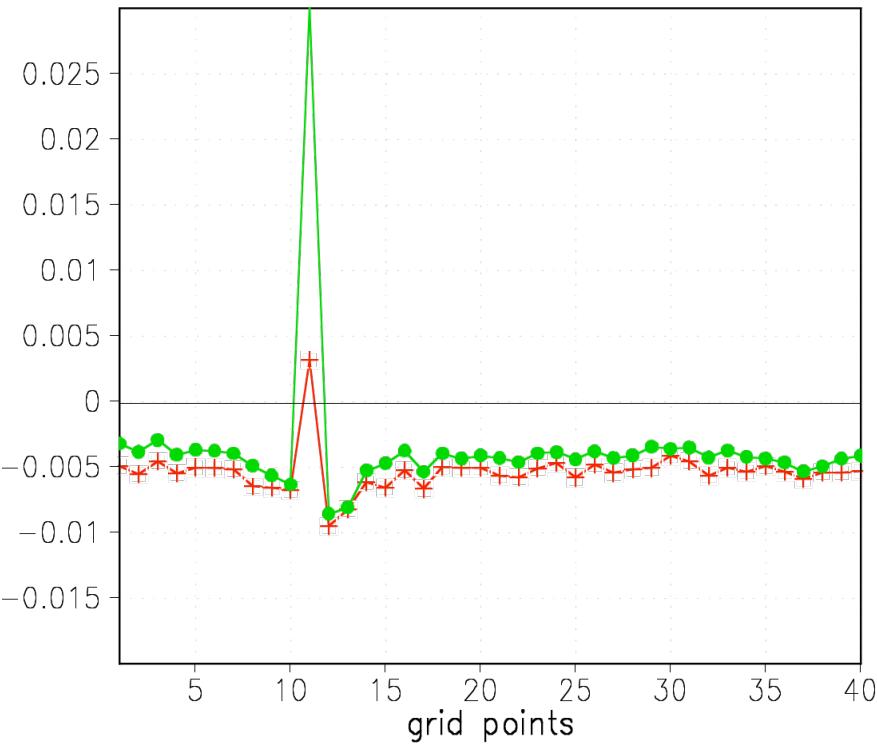
$$\begin{aligned}\Delta \mathbf{e}^2 &= \left[\mathbf{M} \mathbf{K} (\mathbf{y} - H(\mathbf{x}_{0|-6}^b)) \right]^T (\mathbf{e}_{t|0} + \mathbf{e}_{t|-6}) \\ &= \left[\mathbf{X}_{t|0}^f \mathbf{Y}_0^{aT} \mathbf{R}^{-1} (\mathbf{y} - H(\mathbf{x}_{0|-6}^b)) \right]^T (\mathbf{e}_{t|0} + \mathbf{e}_{t|-6})\end{aligned}$$

This is a simple product using the available nonlinear forecast ensemble $\mathbf{X}_{t|0}^f$ and $\mathbf{Y}_0^{aT} = (\mathbf{H} \mathbf{P}^a)^T$

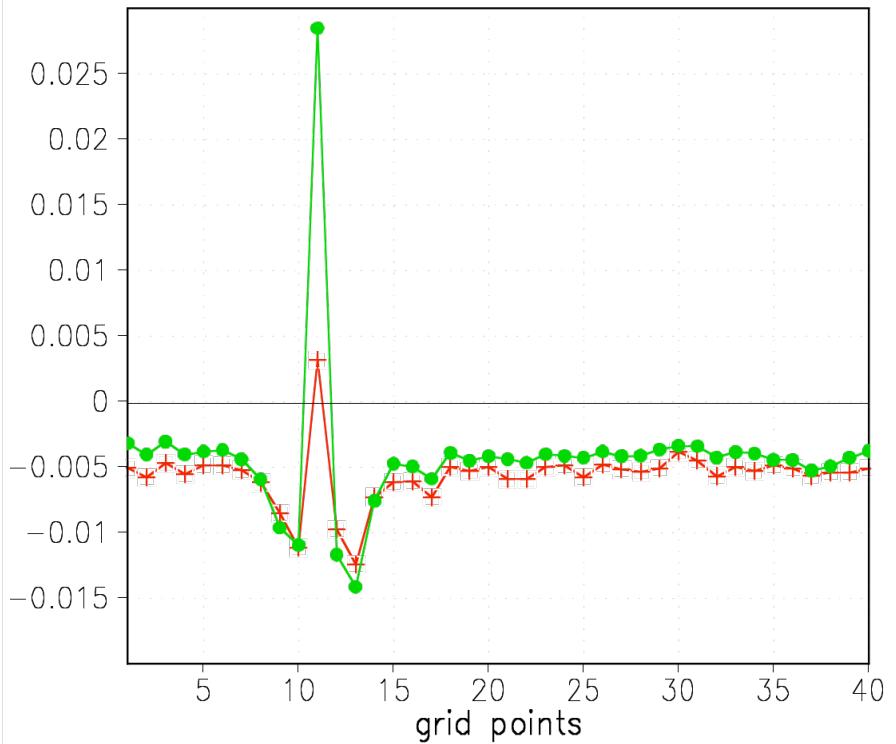
Test ability to detect the poor quality observation on the Lorenz 40 variable model

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

Larger random error



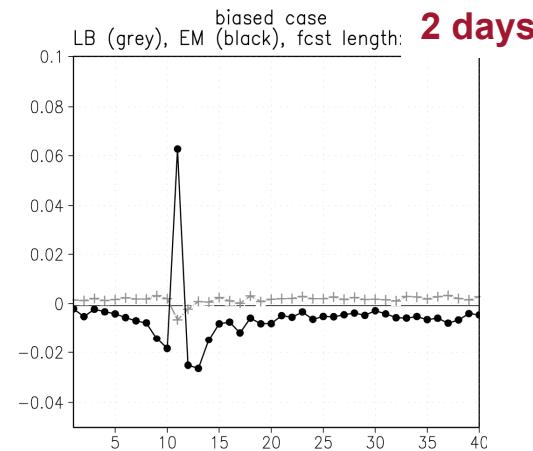
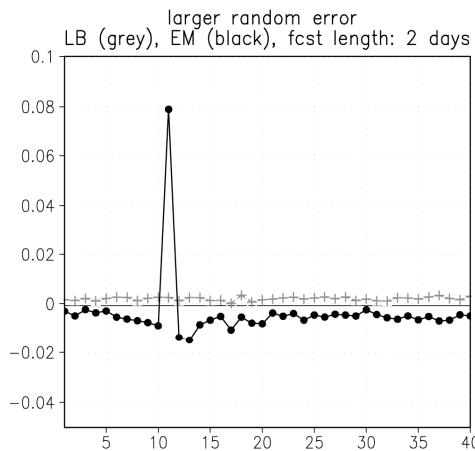
Biased observation case



- ✓ 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.

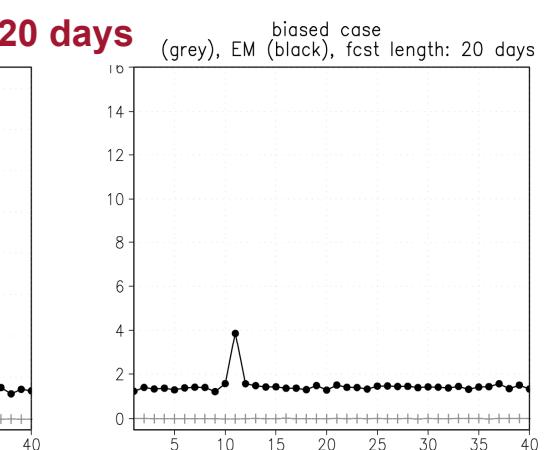
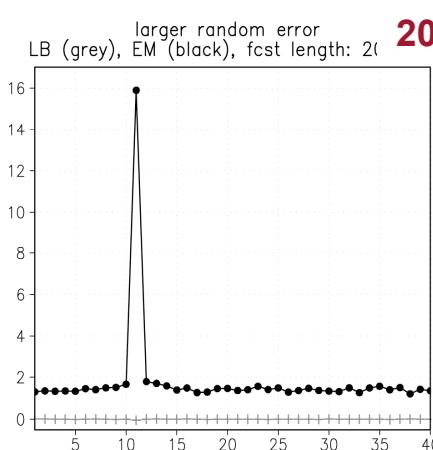
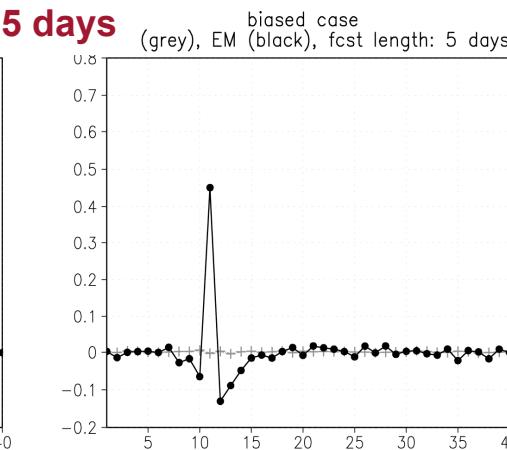
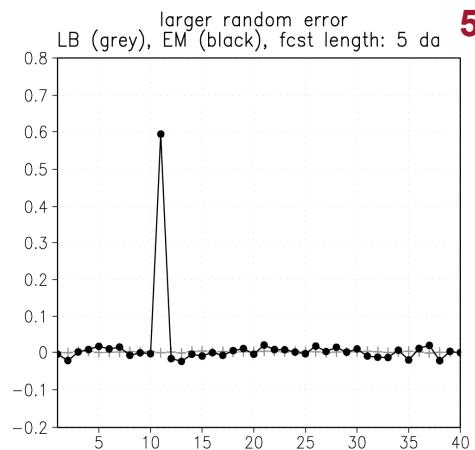
Test ability to detect poor quality observation for different forecast lengths

Larger random error Biased observation case

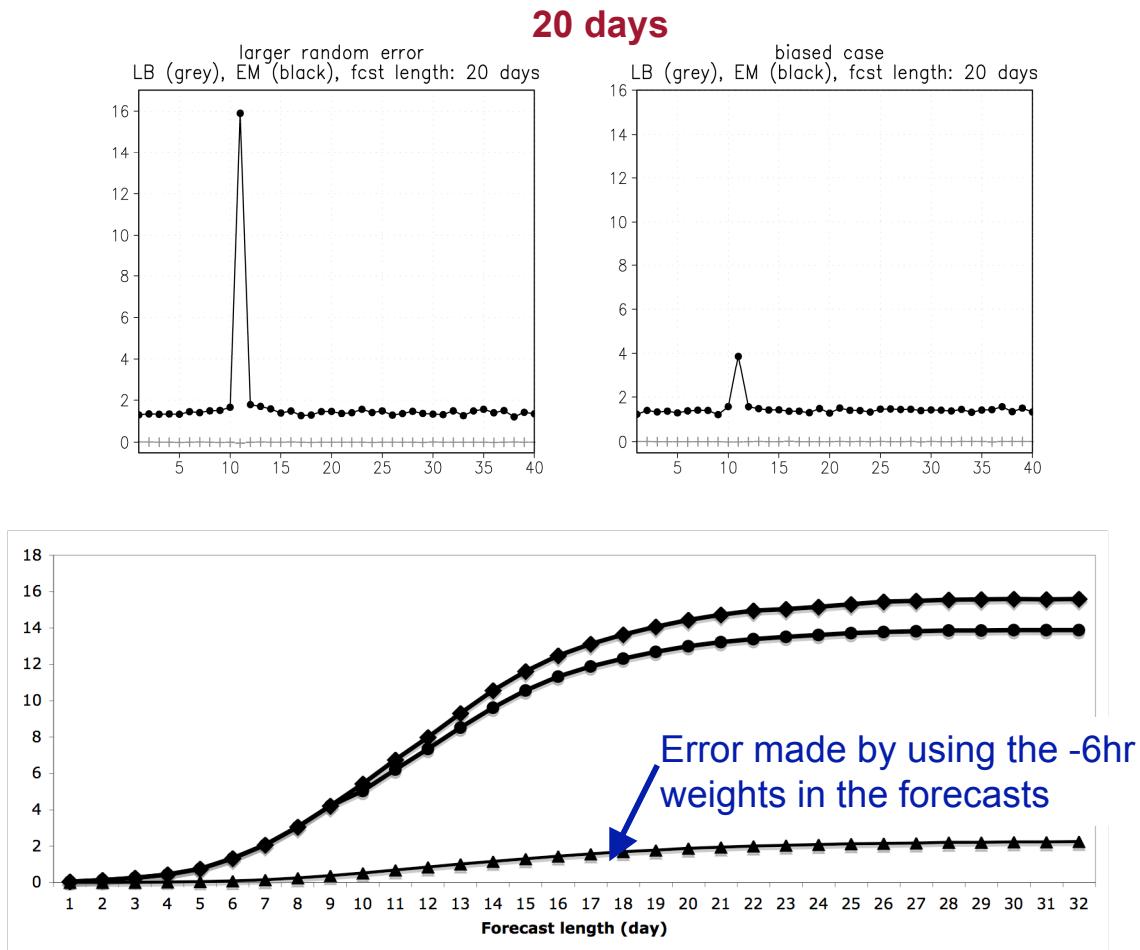


✓ After 2-days the adjoint has the wrong sensitivity sign!

✓ The ensemble sensitivity method has a strong signal even after forecast error has saturated!



How can we possibly detect bad observations even after all skill is lost???



Mean Square Error of the -6hr weighted forecasts (diamonds),
MSE of the 0hr ensemble mean (circles) and MS Difference
between ensemble mean and weighted forecasts (triangles).

- ✓ After 20-days there is no forecast skill but the ensemble sensitivity still detects the wrong observation.
- ✓ The ensemble sensitivity is based on the assumption that the analysis weights can be used in the forecasts. This is accurate even after forecast error has saturated (triangles).
- ✓ As a result we can identify a bad observation even after forecast skill is lost.

Summary

- EnKF is simple, does not require linear tangent model or adjoint of either the model M or the forward operator H
- There are two types of EnKF: perturbed observations (e.g., Evensen, Houtekamer), and Square Root Filter (e.g., Whitaker and Hamill, Anderson, Bishop). These assimilate one obs at a time.
- The LETKF (Hunt et al.) is a SQRF done locally in space: the analysis and analysis perturbations are weighted averages of the ensemble forecasts.
- This makes possible a “no cost smoother”.
- EnKF calculates exactly \mathbf{P}^a (the Hessian!), and the Kalman gain matrix \mathbf{K} . This makes possible to compute longer forecasts sensitivity to observations.