Relationship between Singular Vectors, Bred Vectors, 4D-Var and EnKF

Eugenia Kalnay and Shu-Chih Yang
with Alberto Carrasi, Matteo Corazza
and Takemasa Miyoshi

ECODYC10, Dresden

28 January 2010

Relationship between Singular Vectors, Bred Vectors, 4D-Var and EnKF

Eugenia Kalnay and Shu-Chih Yang
with Alberto Carrasi, Matteo Corazza
and Takemasa Miyoshi

ECODYC10, Dresden

28 January 2010

Outline

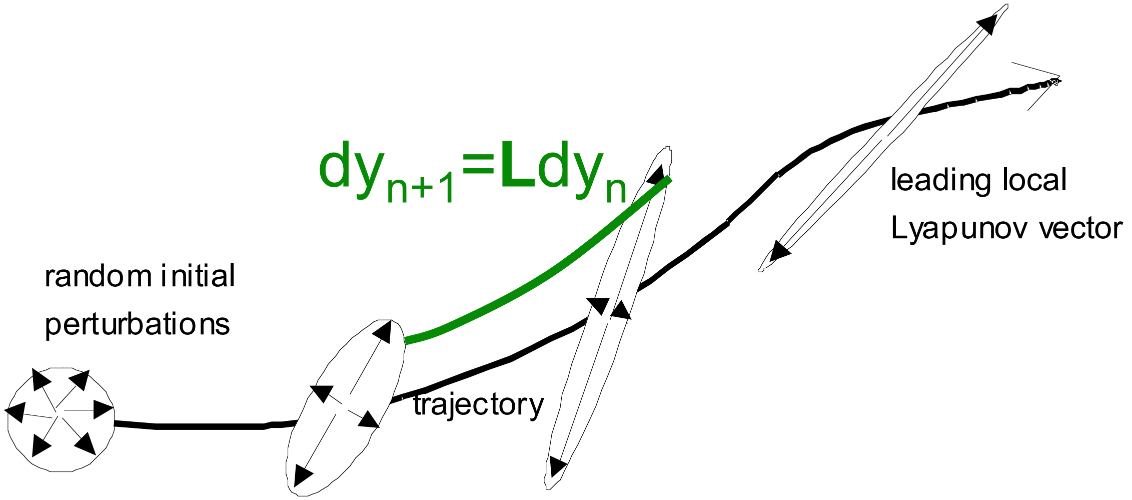
- Bred Vectors and Singular Vectors
- Dependence of initial SVs on the norm
- Introduction to 4D-Var
- Introduction to LETKF
- No-cost smoother
- Applications: Outer Loop and “Running in Place”
- Analysis corrections in 3D-Var, 4D-Var and LETKF
- Analysis corrections at the beginning of the assimilation window
- Summary

Lorenz (1965) introduced (without using their current names) all the concepts of: Tangent linear model, Adjoint model, Singular vectors, and Lyapunov vectors for a low order atmospheric model, and their consequences for ensemble forecasting.

He also introduced the concept of “errors of the day”: predictability is not constant: It depends on the stability of the evolving atmospheric flow (the basic trajectory or reference state).

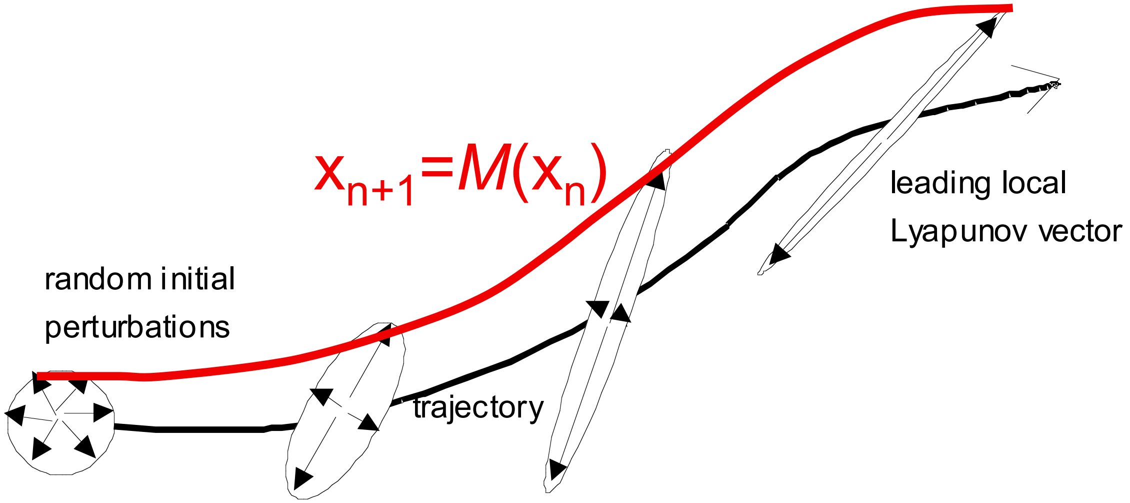
When there is an instability, all perturbations converge towards the fastest growing perturbation (leading Lyapunov Vector). The LLV is computed applying the linear tangent model **L** on each perturbation of the nonlinear trajectory

Fig. 6.7: Schematic of how all perturbations will converge towards the leading Local Lyapunov Vector



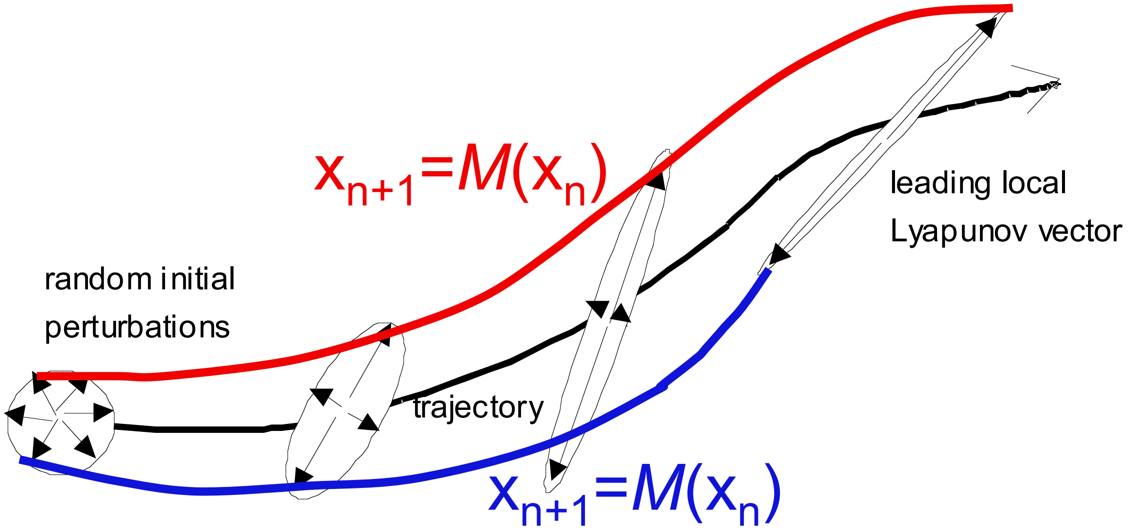
Bred Vectors: nonlinear generalizations of Lyapunov vectors, finite amplitude, finite time

Fig. 6.7: Schematic of how all perturbations will converge towards the leading Local Lyapunov Vector



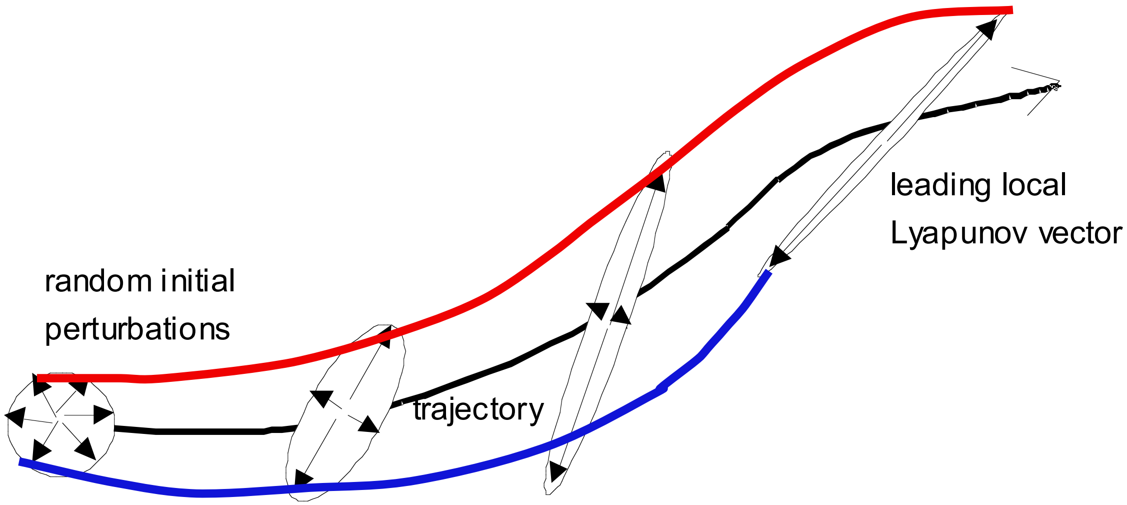
Bred Vectors: nonlinear generalizations of Lyapunov vectors, finite amplitude, finite time

Fig. 6.7: Schematic of how all perturbations will converge towards the leading Local Lyapunov Vector



Bred Vectors: nonlinear generalizations of Lyapunov vectors, finite amplitude, finite time

Fig. 6.7: Schematic of how all perturbations will converge towards the leading Local Lyapunov Vector



Breeding: integrate the model twice, rescale the differences periodically and add them to the control.

SV: Apply the linear tangent model forward in time to a ball of size 1

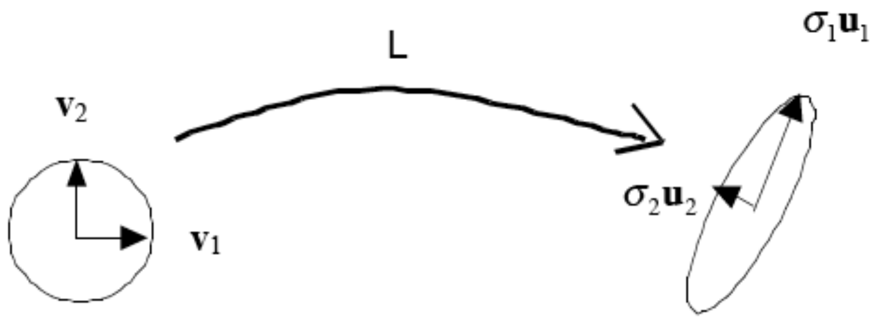
$$\mathbf{L}\mathbf{v}_i = \sigma_i \mathbf{u}_i$$

$\mathbf{v}_i$ are the initial singular vectors

$\mathbf{u}_i$ are the final singular vectors

σ_i are the singular values

Fig. 6.3: Schematic of the application of the TLM to a sphere of perturbations of size 1 for a given interval (t_0,t_1).



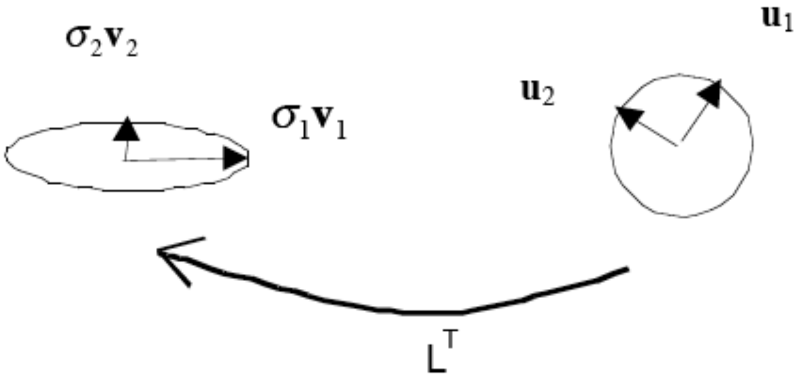
- The ball becomes an ellipsoid, with each final SV $\boldsymbol{u_i}$ multiplied by the corresponding singular value σ_i .
- Both the initial and final SVs are orthogonal.

If we apply the adjoint model backwards in time

$$\mathbf{L}^T \mathbf{u}_i = \sigma_i \mathbf{v}_i$$

v$_i$ are the initial singular vectors
u$_i$ are the final singular vectors
σ_i are the singular values

Fig. 6.4: Schematic of the application of the adjoint of the TLM to a sphere of perturbations of size 1 at the final time.



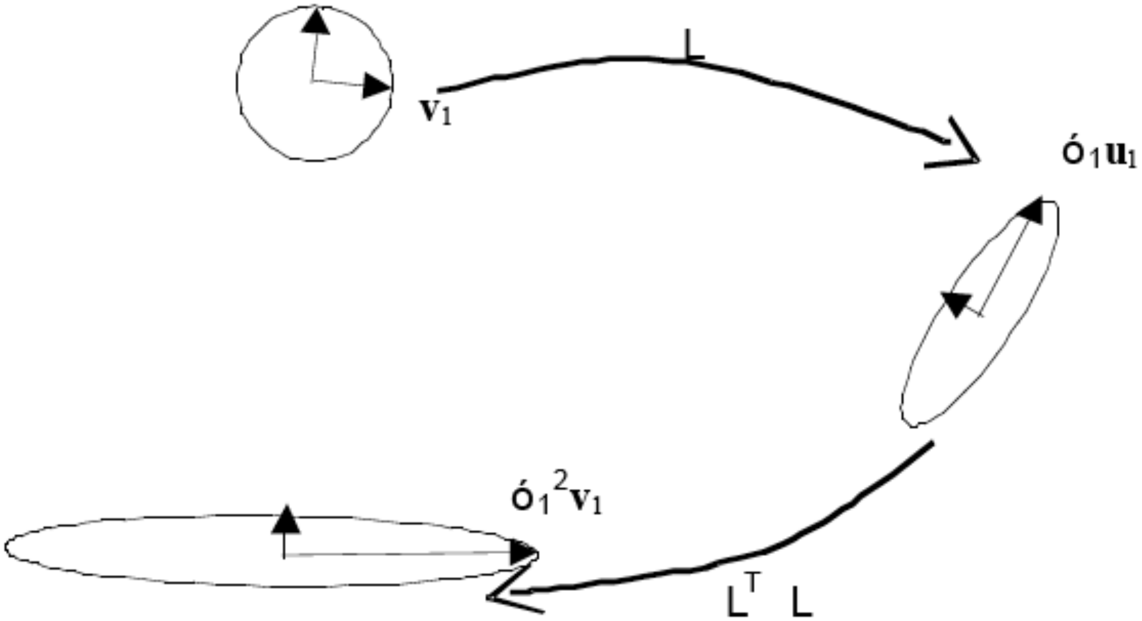
- The final SVs get transformed into initial SVs, and are also multiplied by the corresponding singular value σ_i .

Apply both the linear and the adjoint models

$$\mathbf{L}^T\mathbf{L}\mathbf{v}_i = \sigma_i\mathbf{L}^T\mathbf{u}_i = \sigma_i^2\mathbf{v}_i$$

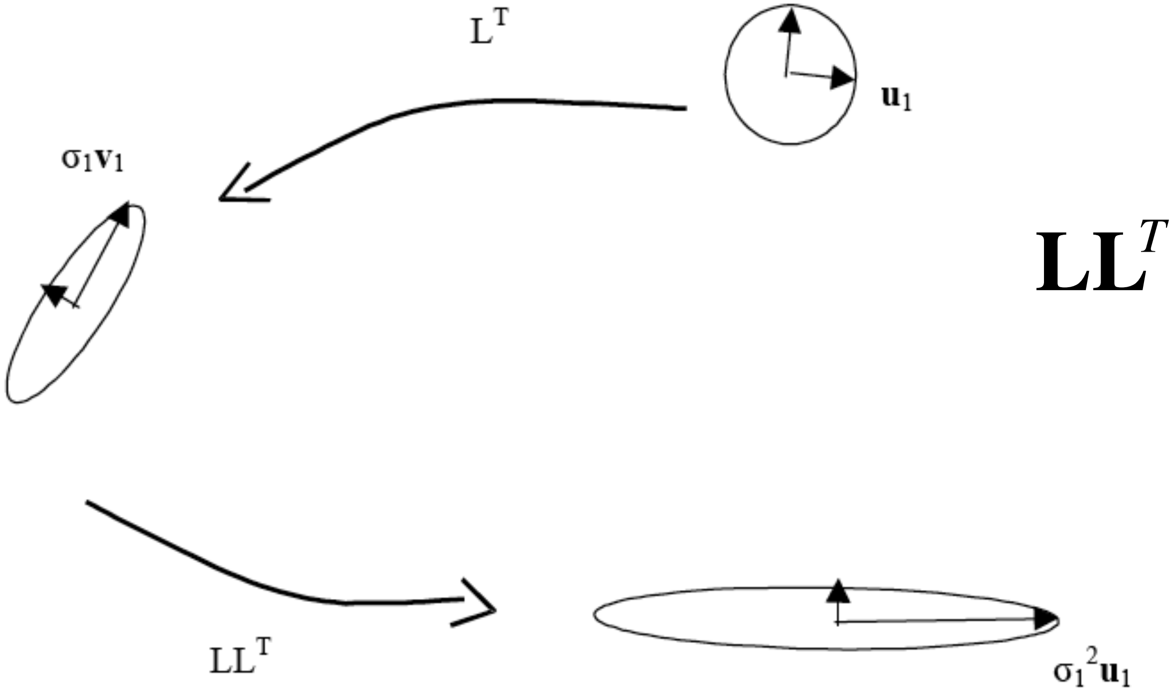
So that $\mathbf{v}_i$ are the eigenvectors of $\mathbf{L}^T\mathbf{L}$ and σ_i^2 are the eigenvalues

Fig. 6.5: Schematic of the application of the TLM forward in time followed by the adjoint of the TLM to a sphere of perturbations of size 1 at the initial time.



Conversely, apply the adjoint model first and then the TLM

Fig. 6.6: Schematic of the application of the adjoint of the TLM backward in time followed by the TLM forward to a sphere of perturbations of size 1 at the final time.



$$\mathbf{L}\mathbf{L}^T\mathbf{u}_i = \sigma_i^2\mathbf{u}_i$$

More generally,

A perturbation is advanced from t_n to t_{n+1} $\mathbf{y}_{n+1} = \mathbf{L}\mathbf{y}_n$

Find the final size with a final norm **P**:

$$\left\|\mathbf{y}_{n+1}\right\|^2 = (\mathbf{P}\mathbf{y}_{n+1})^T(\mathbf{P}\mathbf{y}_{n+1}) = \mathbf{y}_n^T\mathbf{L}^T\mathbf{P}^T\mathbf{P}\mathbf{L}\mathbf{y}_n$$

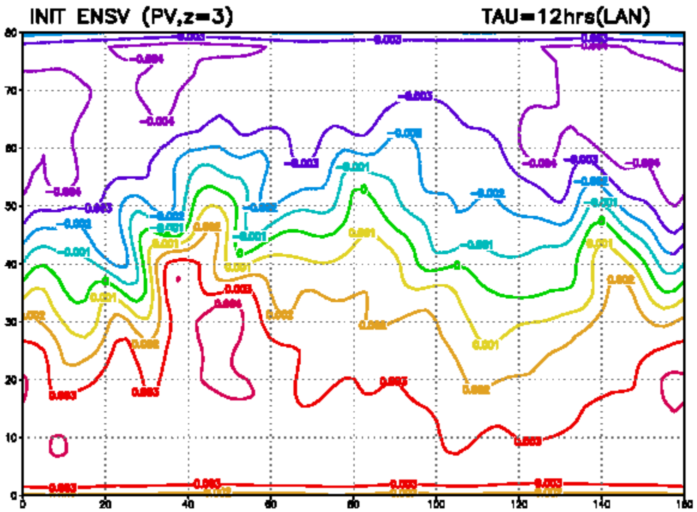
This is subject to the constraint that all the initial perturbations being of size 1 (with some norm **W** that measures the *initial* size):

$$\mathbf{y}_n^T\mathbf{W}^T\mathbf{W}\mathbf{y}_n = 1$$

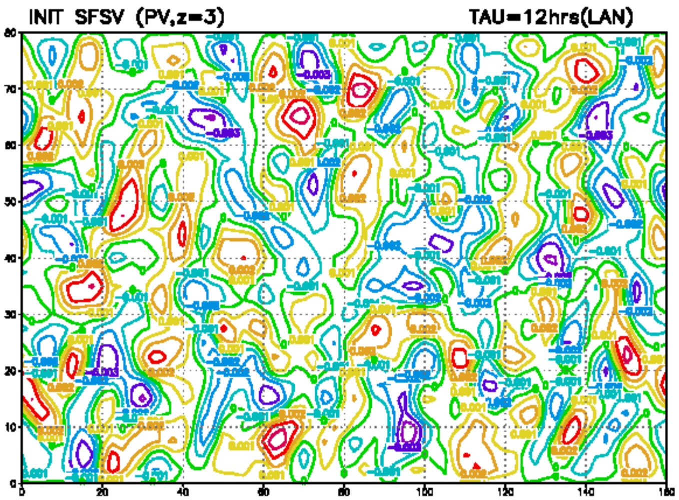
The **initial** leading SVs depend strongly on the initial norm **W** and on the optimization period $T = t_{n+1}\text{-}t_n$

QG model: Singular vectors using either enstrophy/streamfunction initial norms (12hr)

Initial SV with enstrophy norm

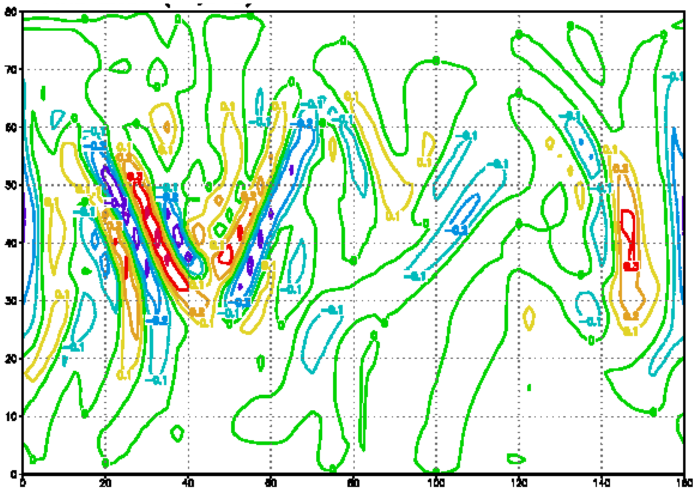


Initial SV with streamfunction norm

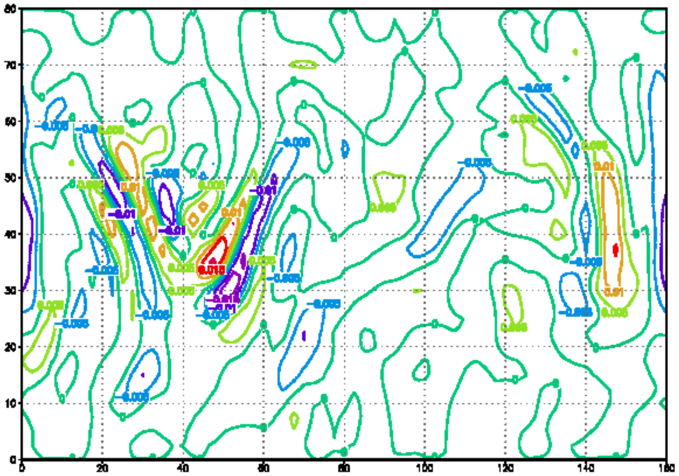


Initial SVs are very sensitive to the norm

Final SV with enstrophy norm



Final SV with streamfunction norm

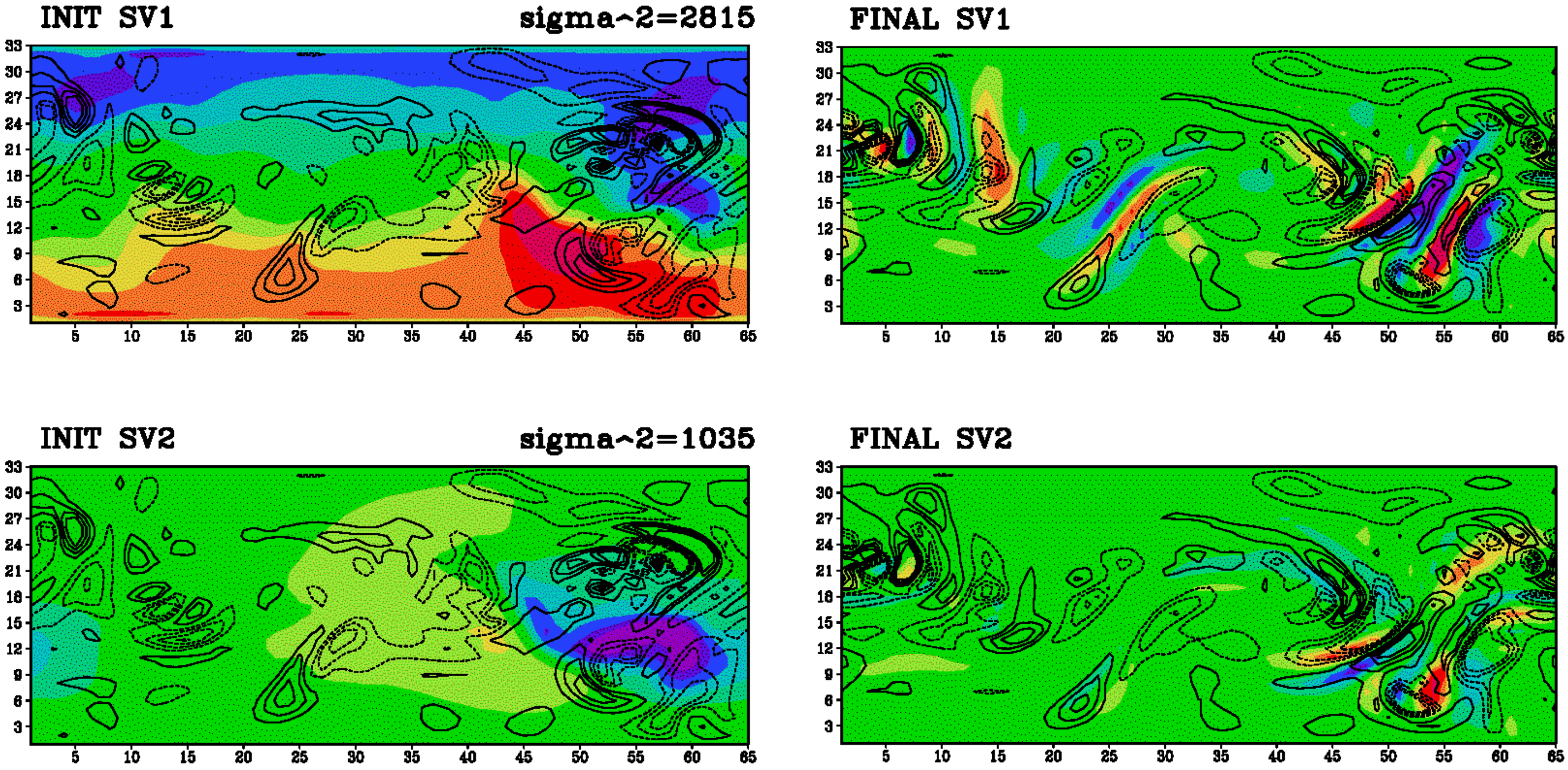


Final SVs look like bred vectors (or Lyapunov vectors)

(Shu-Chih Yang)

Two initial and final SV (24hr, vorticity^2 norm)

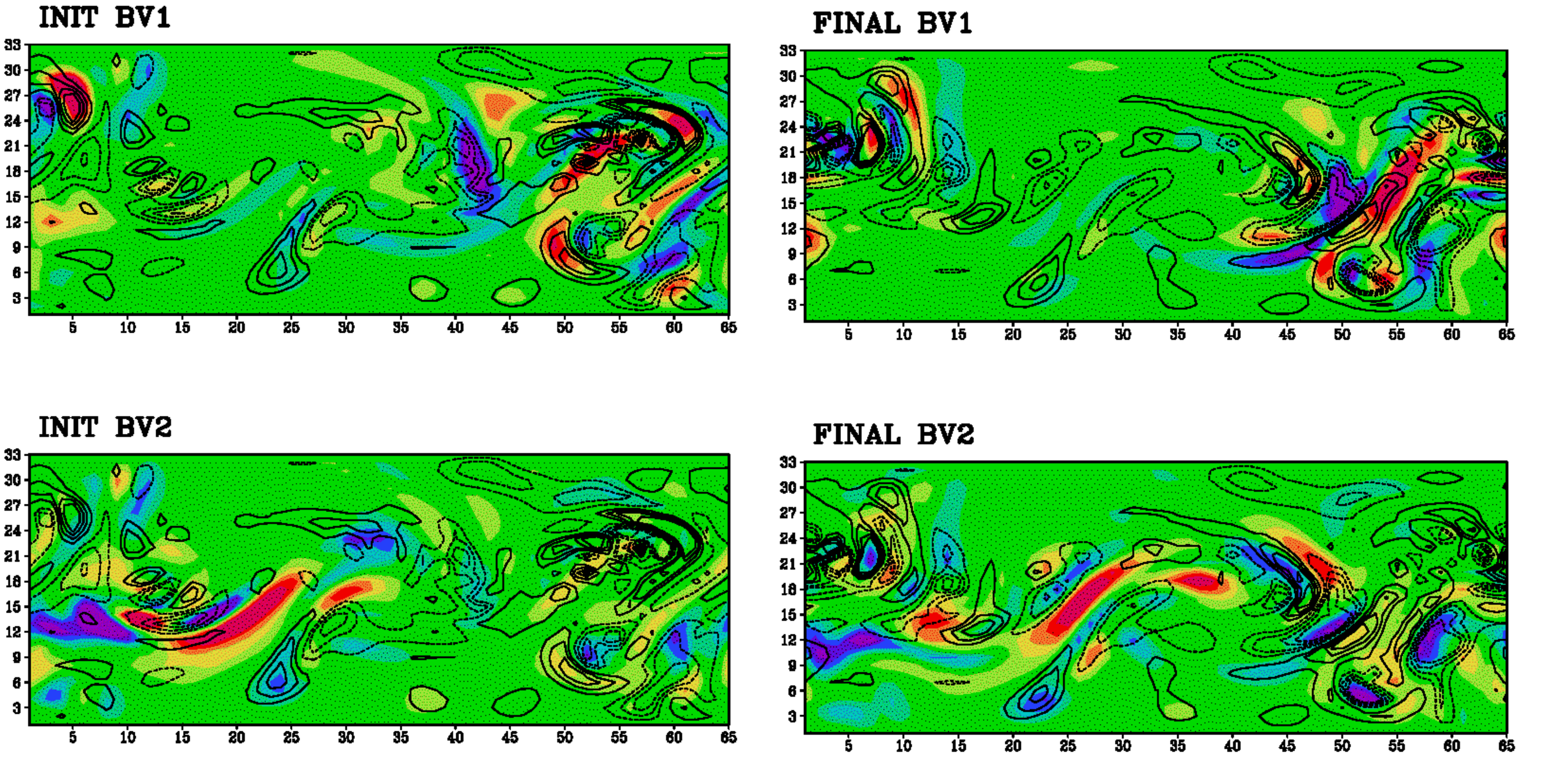
contours: 3D-Var forecast errors, colors: SVs



With an enstrophy norm, the initial SVs have large scales, by the end of the”optimization” interval, the final SVs look like BVs (and LVs)

Two initial and final BV (24hr)

contours: 3D-Var forecast errors, colors: BVs



INIT BV1

FINAL BV1

INIT BV2

FINAL BV2

The BV (colors) have shapes similar to the forecast errors (contours)

Example of nonlinear, tangent linear and adjoint codes:

Lorenz (1963) third equation: $\boxed{\dot{x}_3 = x_1 x_2 - b x_3}$

Nonlinear model, forward in time

$$x_3(t+\Delta t) = x_3(t) + [x_1(t)x_2(t) - bx_3(t)]\Delta t$$

M

Example of nonlinear, tangent linear and adjoint codes:

Lorenz (1963) third equation: $\boxed{\dot{x}_3 = x_1 x_2 - b x_3}$

Nonlinear model, forward in time

$$x_3(t+\Delta t) = x_3(t) + [x_1(t)x_2(t) - bx_3(t)]\Delta t$$

M

Tangent linear model, forward in time

$$\delta x_3(t+\Delta t) = \delta x_3(t) + [x_2(t)\delta x_1(t) + x_1(t)\delta x_2(t) - b\delta x_3(t)]\Delta t$$

L

Example of nonlinear, tangent linear and adjoint codes:

Lorenz (1963) third equation: $\boxed{\dot{x}_3 = x_1 x_2 - b x_3}$

Nonlinear model, forward in time

$$x_3(t+\Delta t) = x_3(t) + [x_1(t)x_2(t) - bx_3(t)]\Delta t$$

M

Tangent linear model, forward in time

$$\delta x_3(t+\Delta t) = \delta x_3(t) + [x_2(t)\delta x_1(t) + x_1(t)\delta x_2(t) - b\delta x_3(t)]\Delta t$$

L

In the adjoint model the above line becomes

$$\delta x_3^*(t) = \delta x_3^*(t) + (1 - b\Delta t)\delta x_3^*(t+\Delta t)$$

$$\delta x_2^*(t) = \delta x_2^*(t) + (x_1(t)\Delta t)\delta x_3^*(t+\Delta t)$$

$$\delta x_1^*(t) = \delta x_1^*(t) + (x_2(t)\Delta t)\delta x_3^*(t+\Delta t)$$

$$\delta x_3^*(t+\Delta t) = 0$$

backward in time

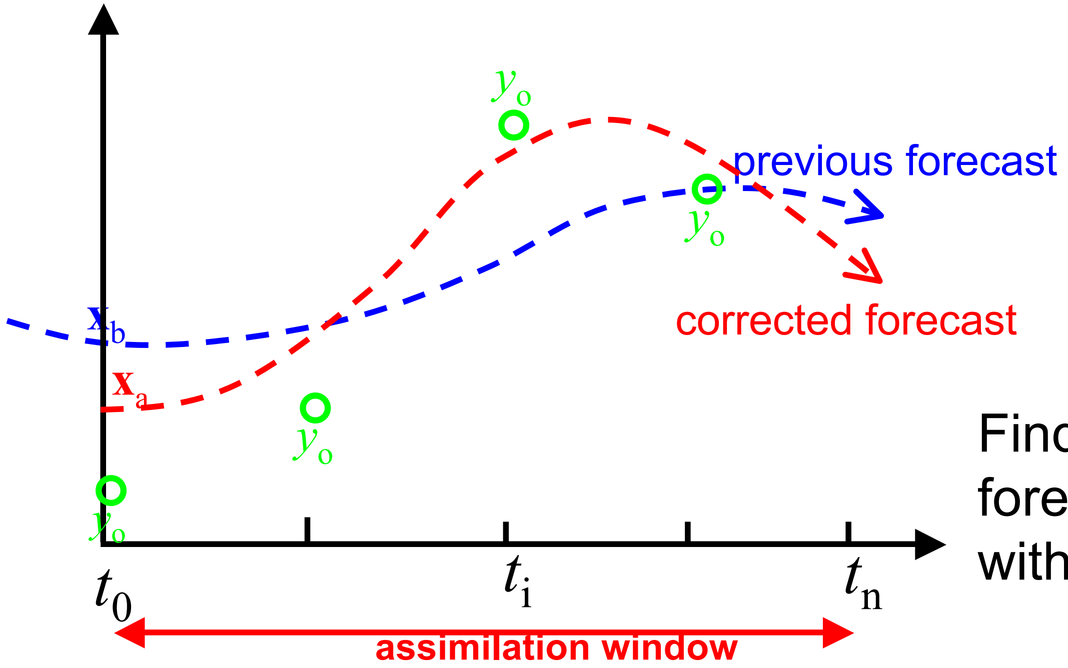
L^T

SVs summary and extra properties

- To obtain the SVs we need the TLM and the ADJ models.
- The leading SVs are obtained by the Lanczos algorithm.
- One can define an initial and a final norm (size), this gives flexibility (and arbitrariness, Ahlquist, 2000).
- The leading initial SV is the vector that will grow fastest (starting with a very small initial norm and ending with the largest final norm).
- The leading SVs grow initially faster than the Lyapunov vectors, but at the end of the period, they look like LVs (and bred vectors~LVs).
- The initial SVs are *very* sensitive to the norm used. The final SVs look like LVs~BVs

4D-Var

$J(\mathbf{x})$ is generalized to include observations at different times.



Find the initial condition such that its forecast best fits the observations within the assimilation interval

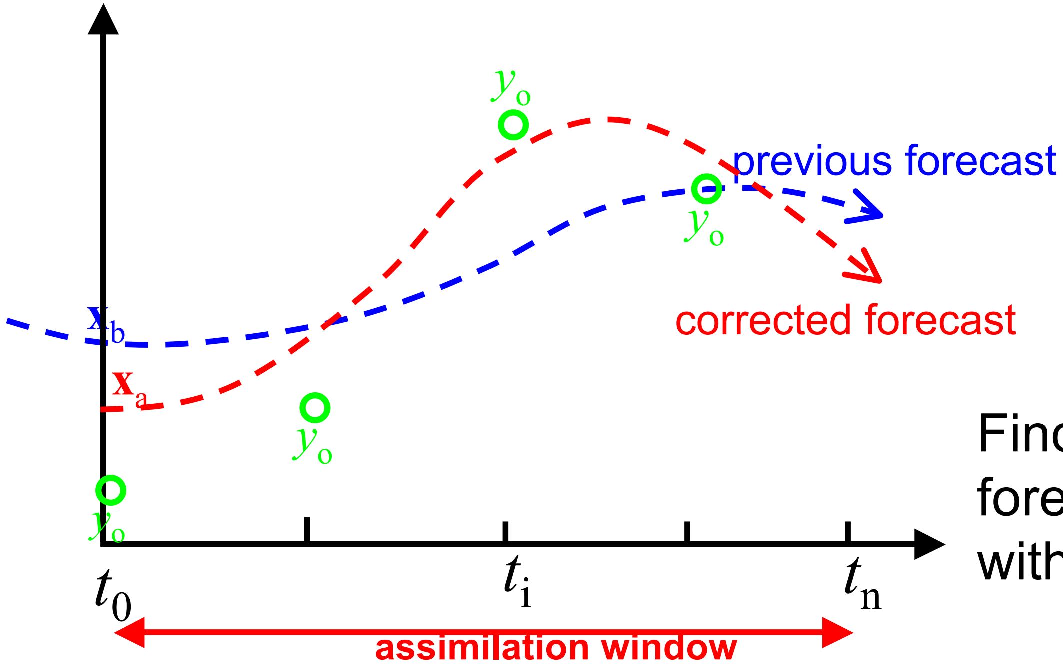
Minimize the 4D-Var cost function:

$$J(\mathrm{x}(t_0)) = \frac{1}{2}[\mathbf{x}(t_0)-\mathbf{x}^b(t_0)]^T \mathbf{B}_0^{-1}[\mathbf{x}(t_0)-\mathbf{x}^b(t_0)] + \ \frac{1}{2}\sum_{i=0}^{i=N}\mathbf{y}^o_i - H(\mathbf{x}_i)]^T \mathbf{R}_i^{-1}[\mathbf{y}^o_i - H(\mathbf{x}_i)]$$

Distance to the background at t_0 Distance to the observations, t_0 - t_n

4D-Var

$J(\mathbf{x})$ is generalized to include observations at different times.

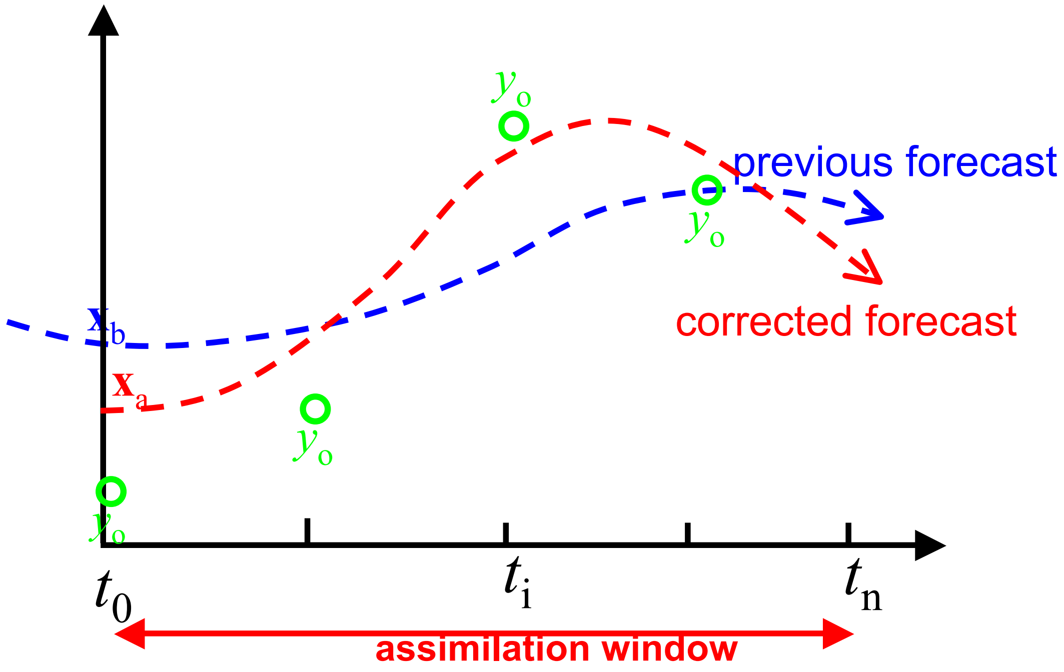


Find the initial condition such that its forecast best fits the observations within the assimilation interval

$$J(\mathrm{x}(t_0)) = \frac{1}{2}[\mathbf{x}(t_0)\text{-}\mathbf{x}^{\mathrm{b}}(t_0)]^T \mathbf{B}_0^{-1}[\mathbf{x}(t_0)\text{-}\mathbf{x}^{\mathrm{b}}(t_0)] + \ \frac{1}{2}\sum_{i=0}^{i=N}\mathbf{y}^{\mathrm{o}}_{\mathrm{i}}\text{-}H(\mathbf{x}_{\mathrm{i}})]^T \mathbf{R}_{\mathrm{i}}^{-1}[\mathbf{y}^{\mathbf{o}}_{\mathrm{i}}\text{-}H(\mathbf{x}_{\mathrm{i}})]$$

The form of the cost function suggests that the analysis increments in 4D-Var will be dominated by leading SVs.

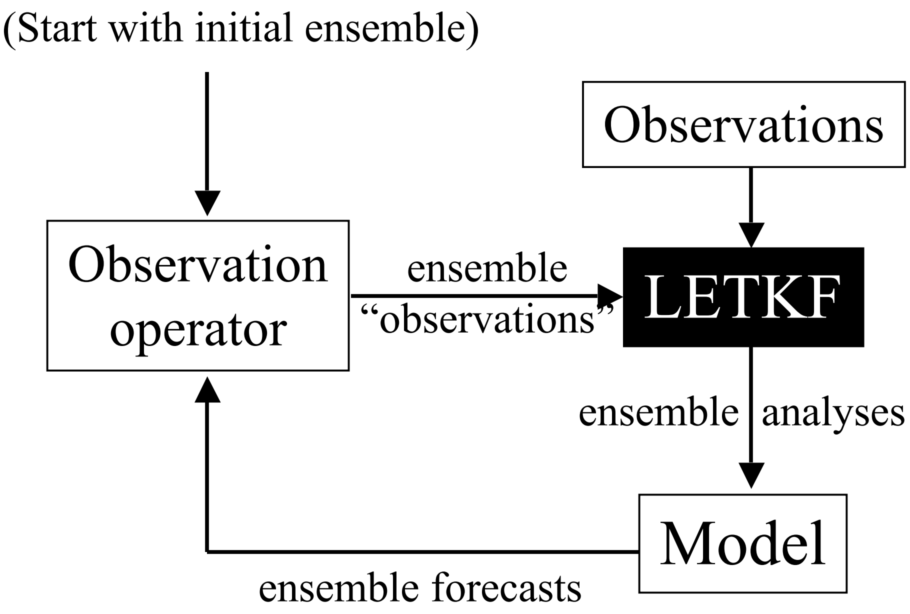
4D-Var is a smoother



The corrected forecast is the 4D-Var analysis throughout the assimilation window

What about LETKF, a sequential method?

Local Ensemble Transform Kalman Filter (Ott et al, 2004, Hunt et al, 2004, 2007) (a square root filter)

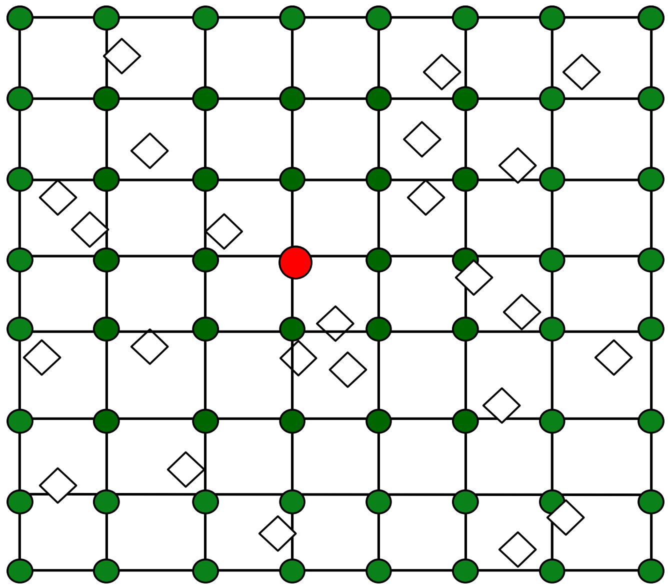


- Model independent (black box)
- Obs. assimilated simultaneously at each grid point
- Parallel analysis: each grid point is independent
- **4D LETKF extension**

Localization based on observations

Perform data assimilation in a local volume, choosing observations

The state estimate is updated at the central grid red dot

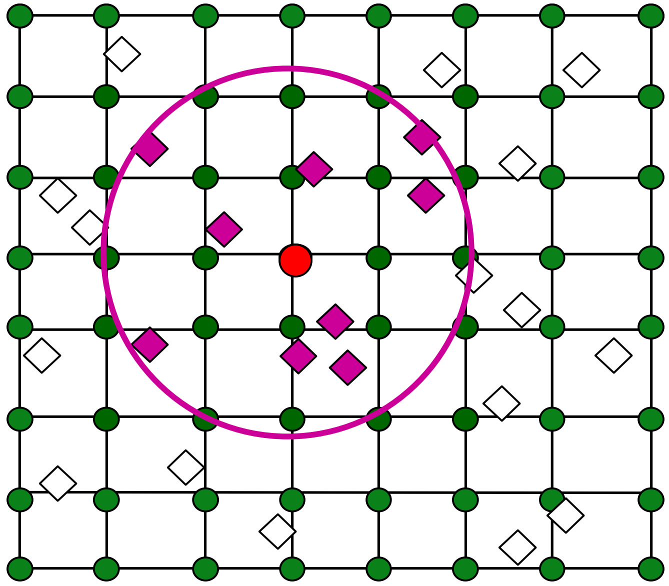


Localization based on observations

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

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

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

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

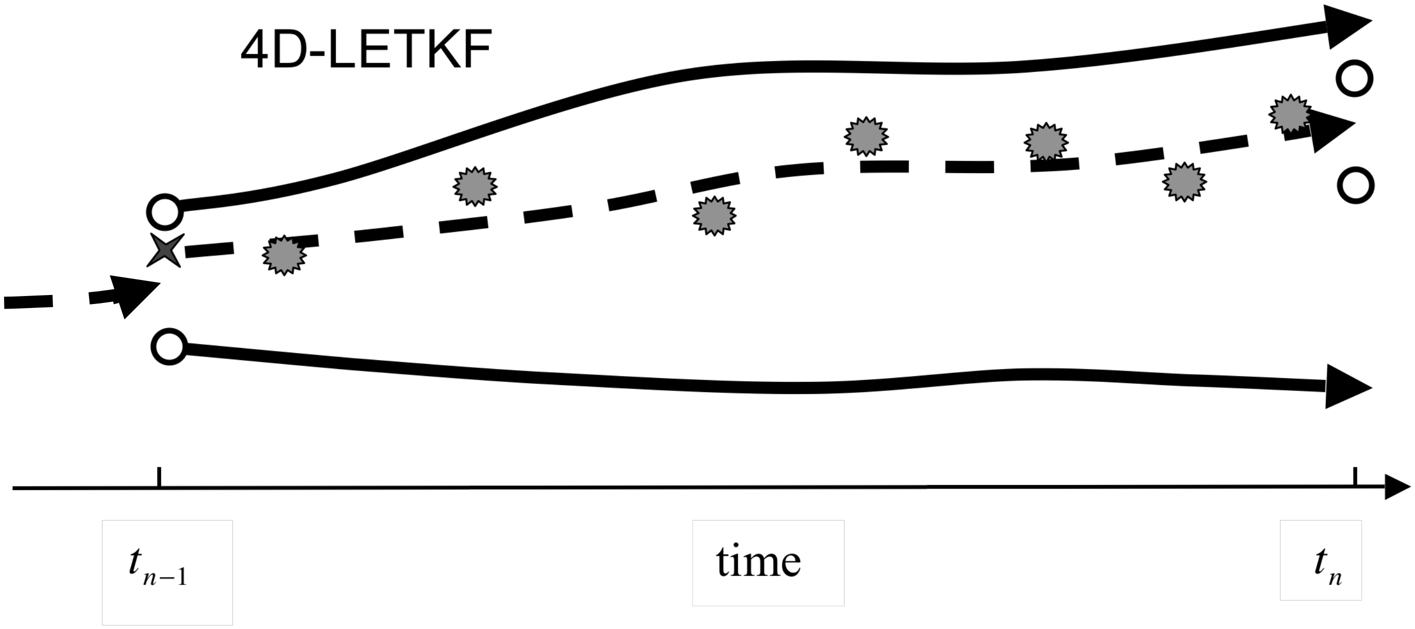
$$\tilde{\mathbf{P}}^a = \left[(K-1)\mathbf{I} + \mathbf{Y}^{bT}\mathbf{R}^{-1}\mathbf{Y}^b\right]^{-1}; \mathbf{W}^a = [(K-1)\tilde{\mathbf{P}}^a]^{1/2}$$

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

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

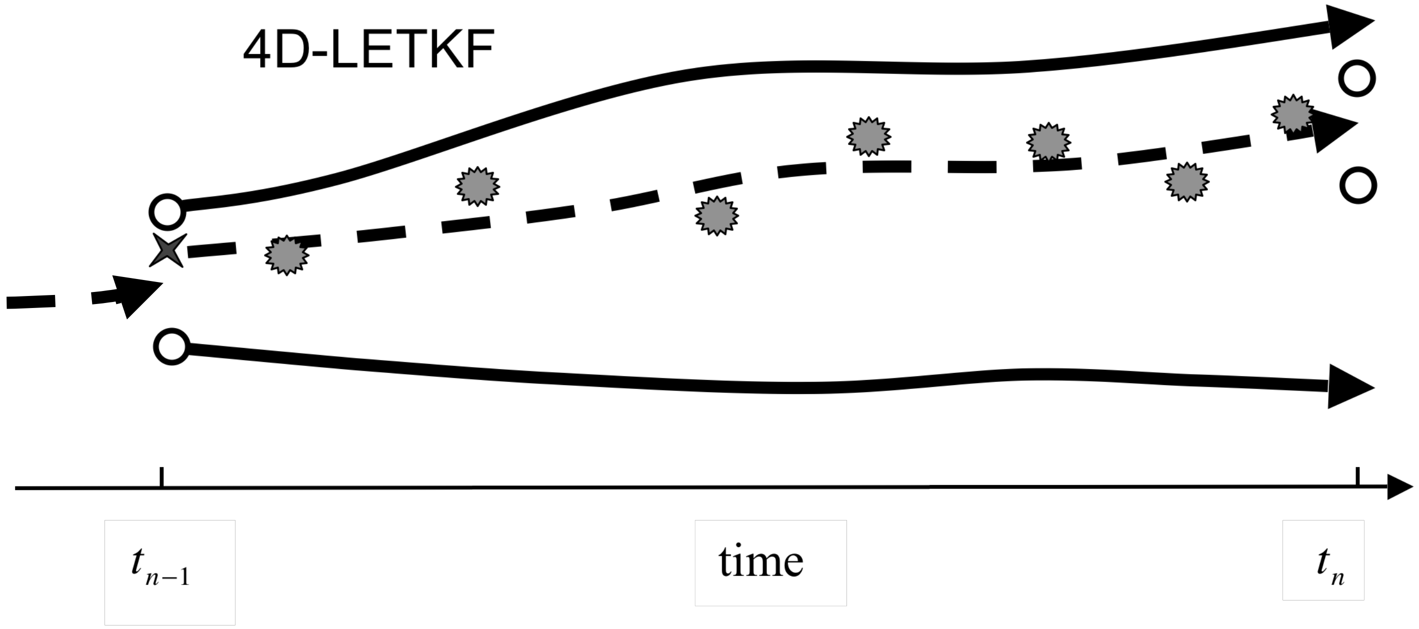
The new ensemble analyses in model space are the columns of $\mathbf{X}^a_n = \mathbf{X}^b_n\mathbf{W}^a + \overline{\mathbf{x}}^b$. Gathering the grid point analyses forms the new global analyses. Note that the the output of the LETKF are analysis weights $\overline{\mathbf{w}}^a$ and perturbation analysis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

The weights are determined at the end of the window…



But the weights are valid throughout the assimilation window!

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



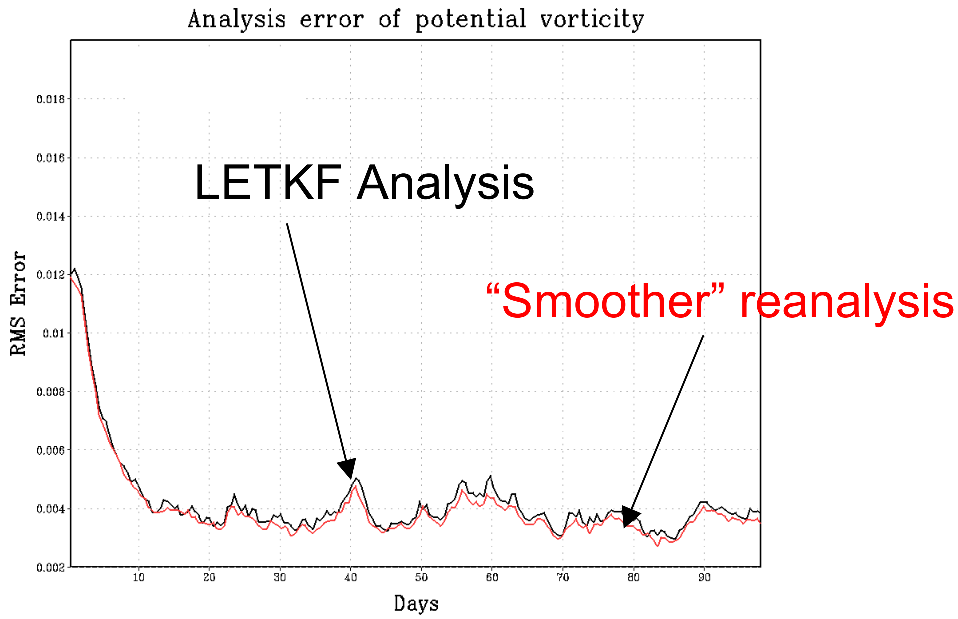
No-cost LETKF smoother tested on a QG model: it works…

LETKF analysis
at time *n*

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

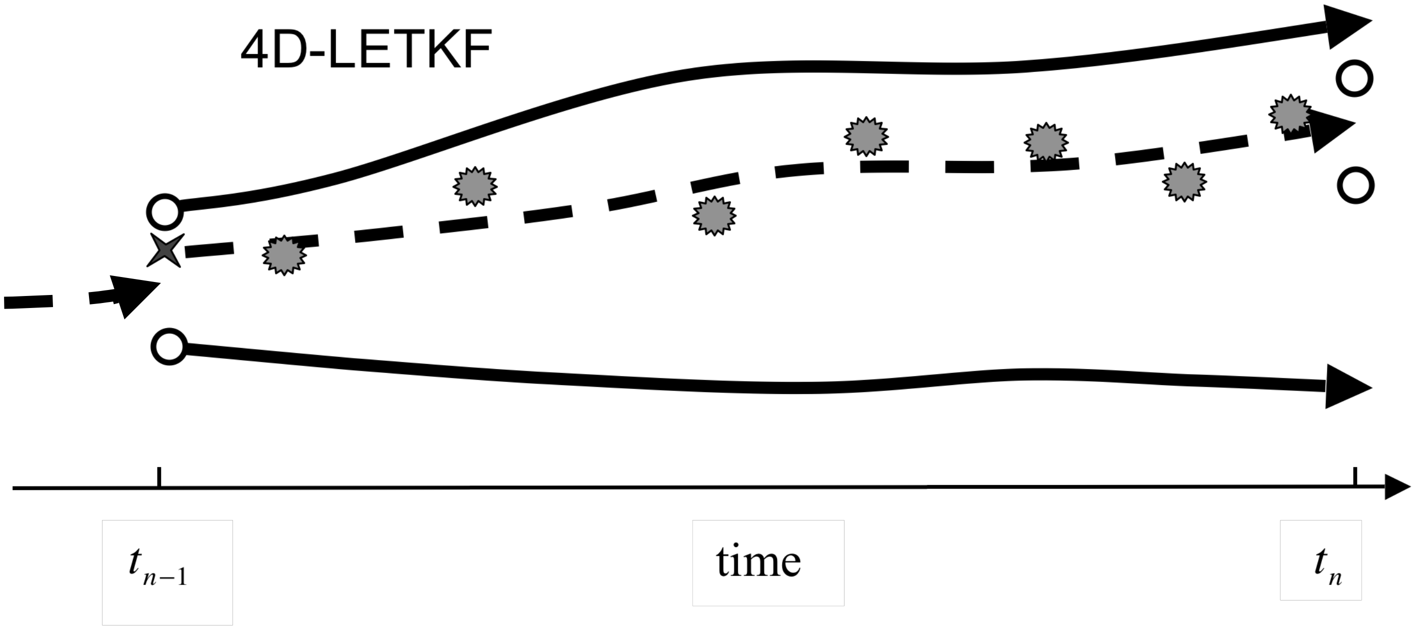
Smoother analysis
at time *n*-1

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



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

No-cost LETKF smoother (✕): 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: dealing with nonlinearity/non-Gaussianity

Nonlinearities and the "outer loop"

- A disadvantage of EnKF is that it cannot handle well nonlinear (non-Gaussian) perturbations and therefore needs short assimilation windows.
- It doesn't have the **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)

With long windows + Pires et al. => 4D-Var clearly wins!

Nonlinearities: “Outer Loop”

Outer loop: similar to 4D-Var: use the final weights to correct only the mean initial analysis, keeping the initial perturbations. Repeat the analysis once or twice.
It re-centers the ensemble on a more accurate nonlinear solution.

Lorenz -3 variable model RMS analysis error

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

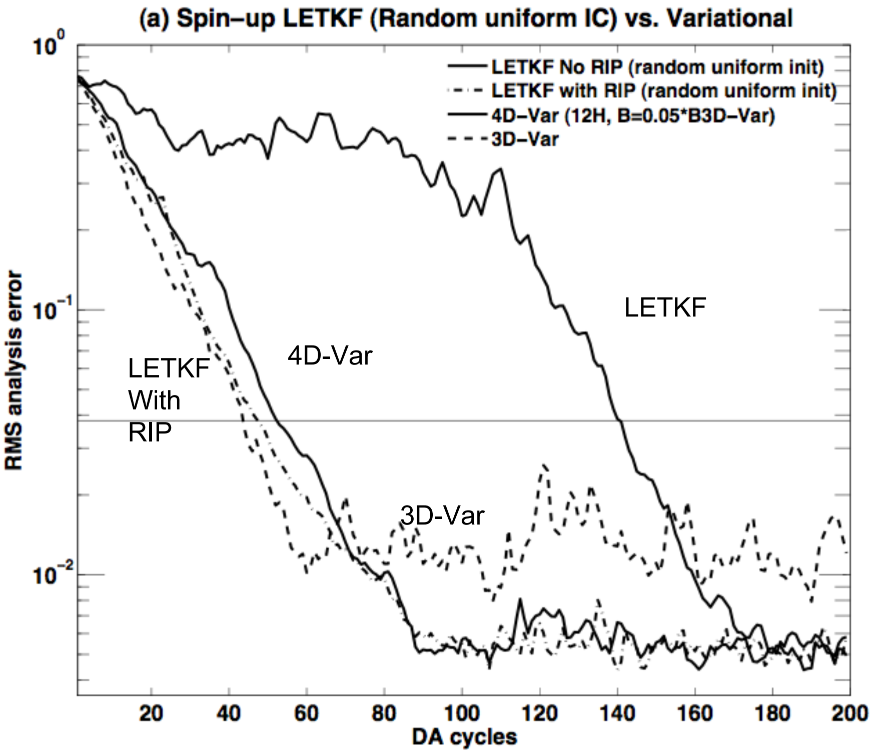
Nonlinearities, “Outer Loop” and “Running in Place”

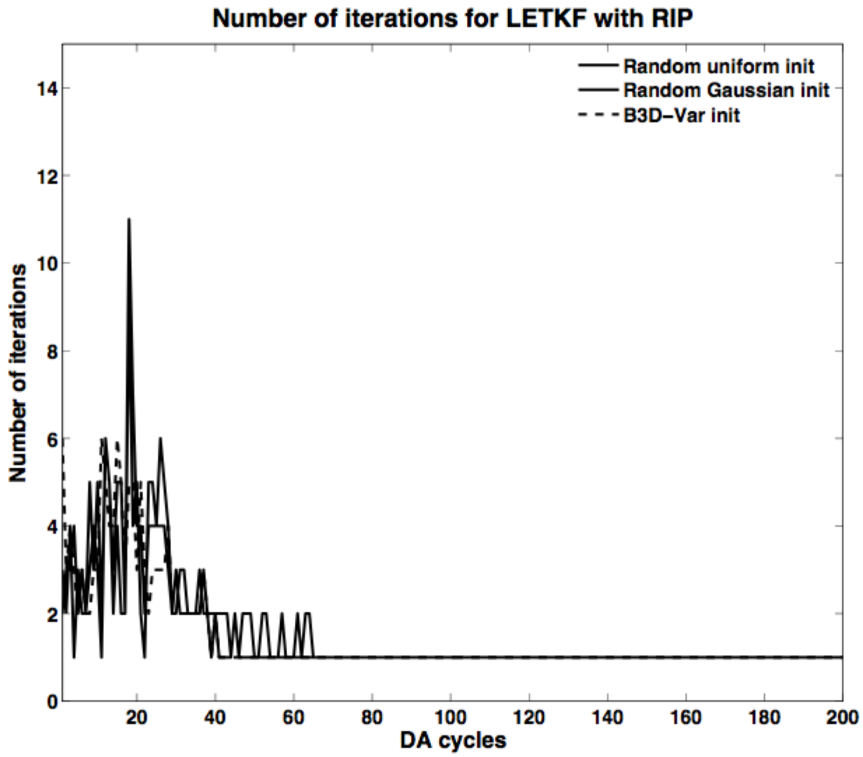
“Running in place”: like the outer loop but smoothing both the **analysis** and the **analysis error covariance** and iterating a few times…

Lorenz -3 variable model RMS analysis error

	4D-Var	LETKF	LETKF	LETKF
			+outer loop	+RIP
Window=8 steps	0.31	0.30	**0.27**	0.27
Window=25 steps	0.53	0.66	**0.48**	**0.39**

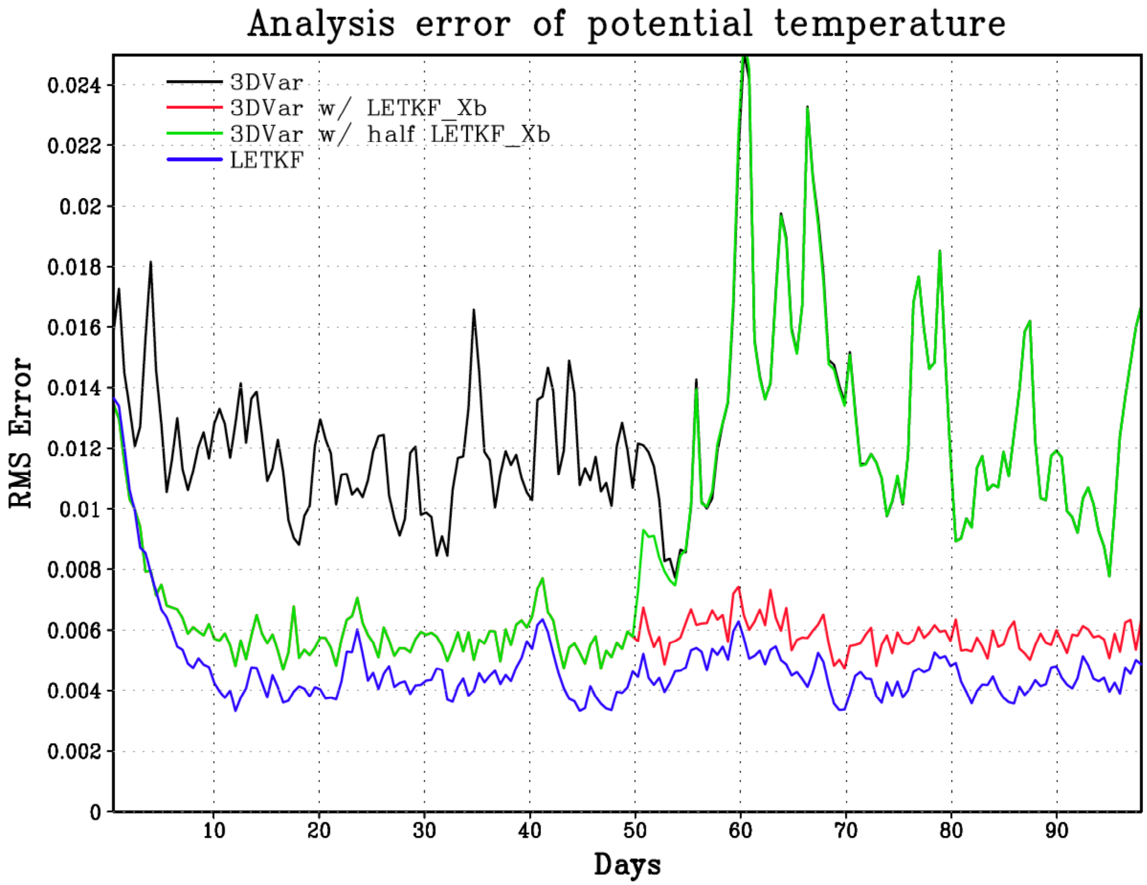
Running in Place: Accelerates spin-up





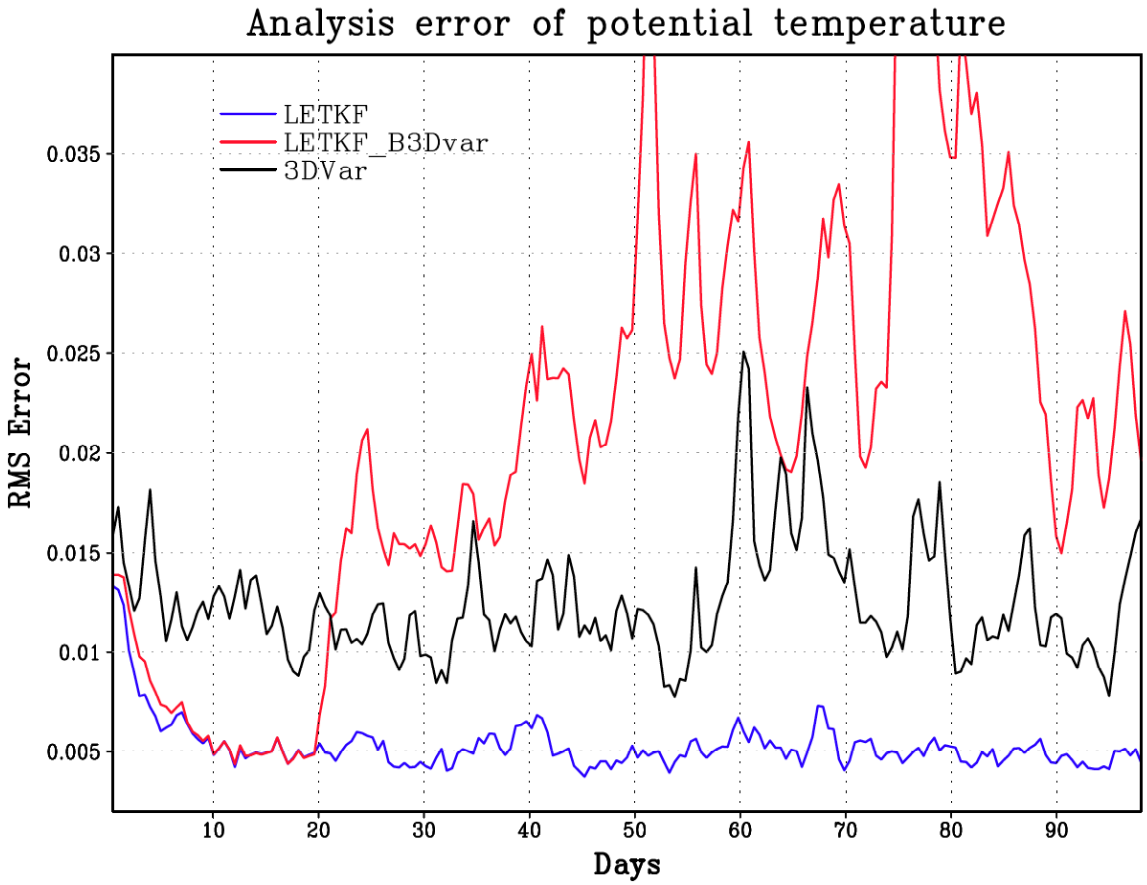
Spin-up depends on initial perturbations, but RIP works well even with random perturbations. It becomes as fast as 4D-Var (blue). RIP takes only 2-6 iterations, and it turns off after spin-up.

Comparison of LETKF with 3D-Var: use P_{LETKF} instead of $P_{3D\text{-}Var}$

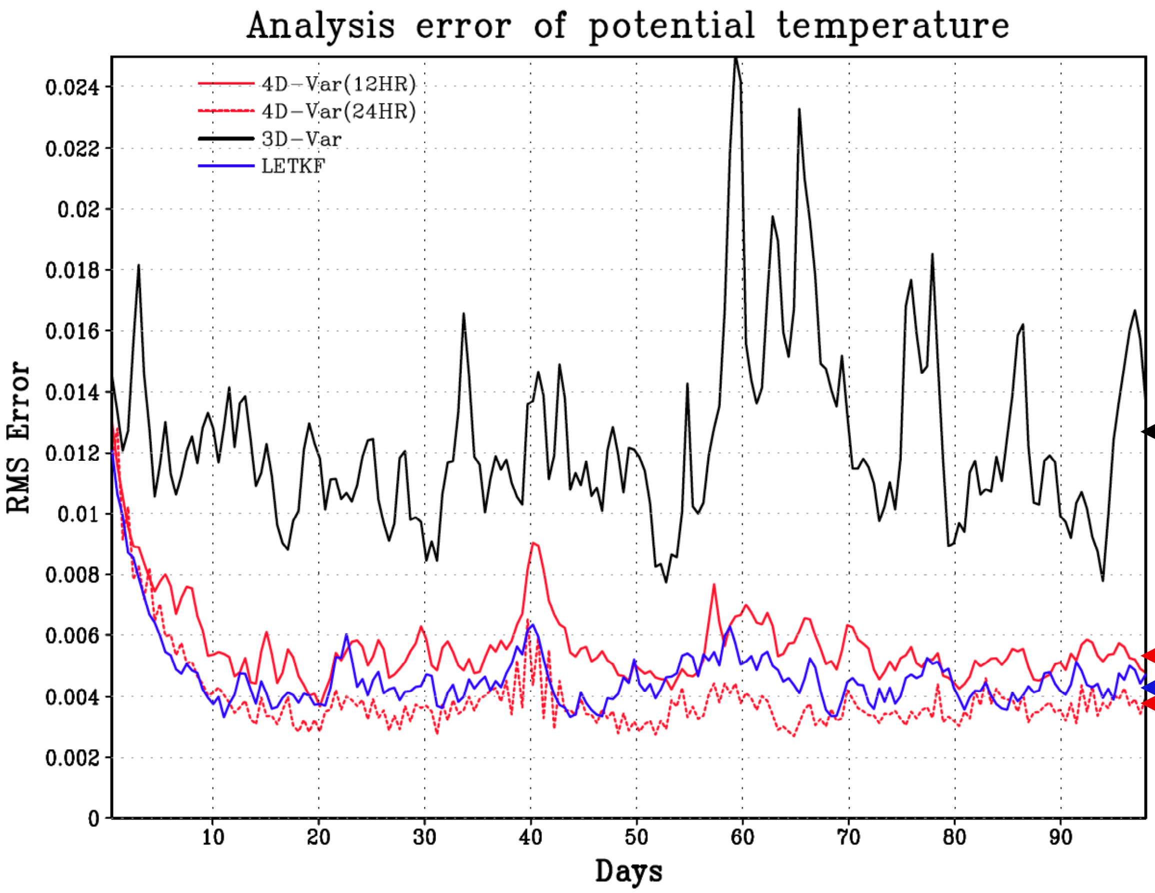


LETKF is clearly better than 3D-Var

LETKF with $P_{3D\text{-}Var}$ diverges



Comparison of 3D-Var, 4D-Var and LETKF



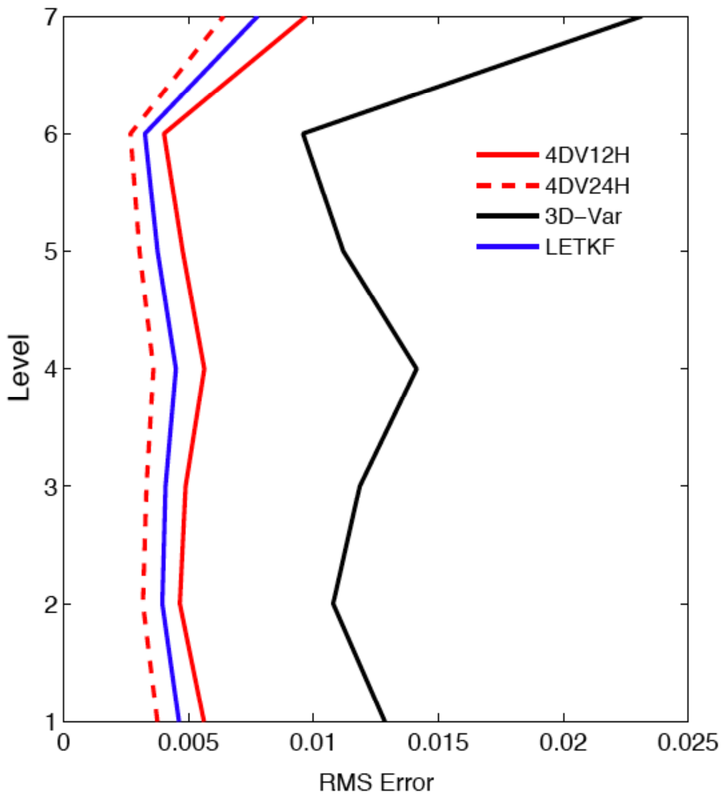
3D-Var, 12hr window

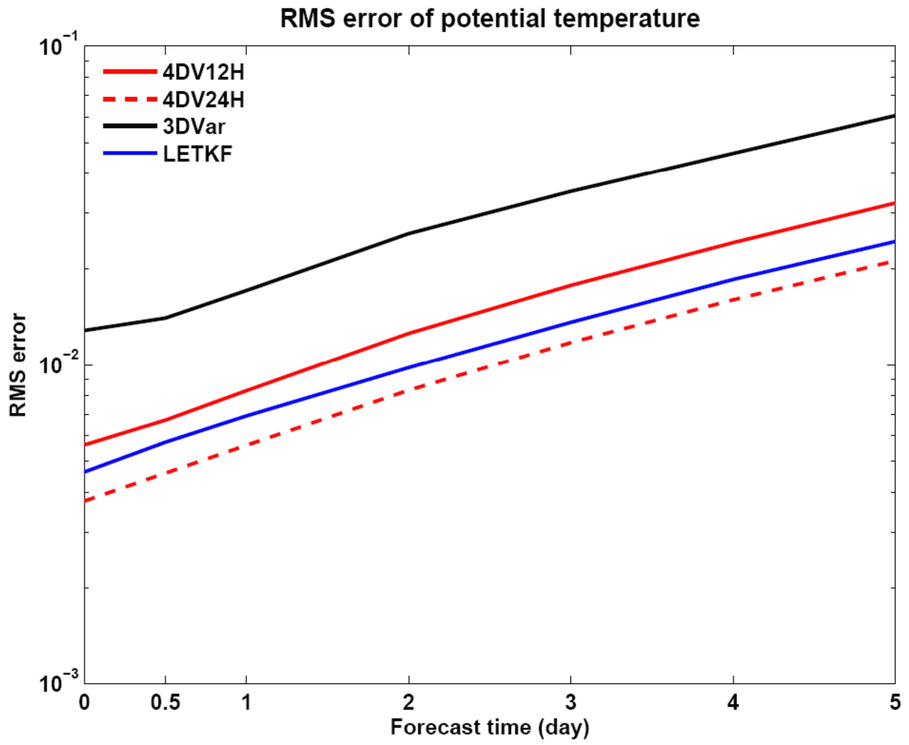
4D-Var, 12hr window

LETKF, 12hr window

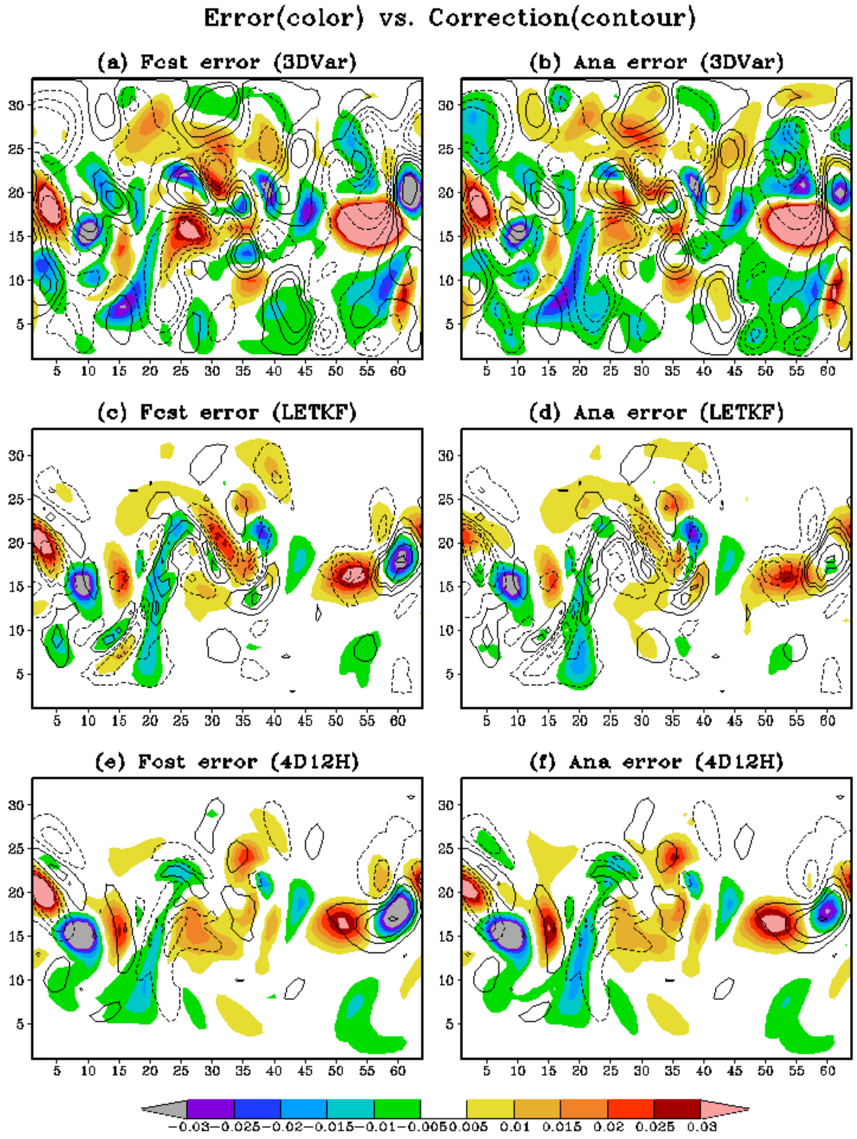
4D-Var, 24hr window

Consistent results: 3D-Var much worse, LETKF slightly better than 4D-Var-12hr and slightly worse than 4D-Var-24hr





Forecast/analysis errors (colors) and analysis correction (contours) at the end of the assimilation window



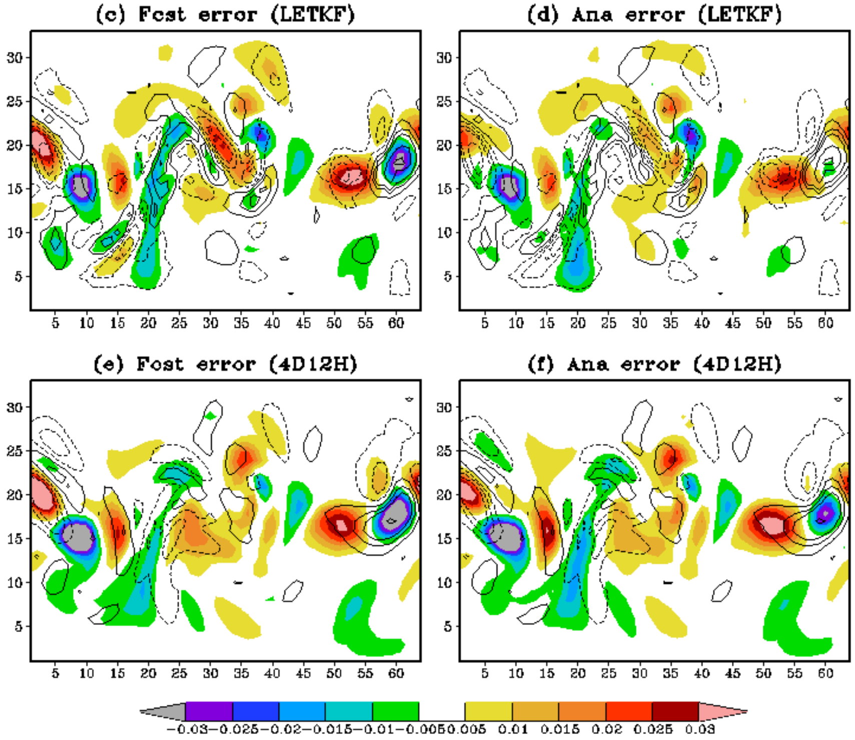
3D-Var
misses correcting the
"errors of the day"

LETKF

very similar corrections of
the "errors of the day"

4D-Var-12hr

At the end of the assimilation window, the 4D-Var and LETKF corrections are clearly very similar.



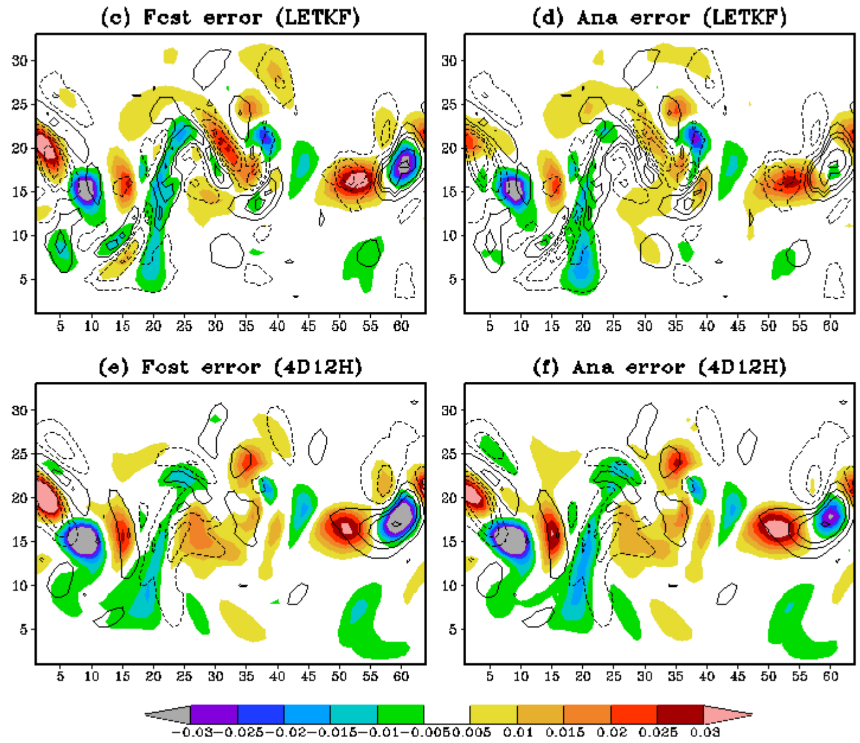
LETKF

4D-Var-12hr

At the end of the assimilation window, the 4D-Var and LETKF corrections are clearly very similar.

What about at the beginning of the assimilation window?

4D-Var is already a smoother, we know the initial corrections. We can use the “no-cost” LETKF smoother to also obtain the “initial” EnKF corrections.



(c) Fcst error (LETKF)

(d) Ana error (LETKF)

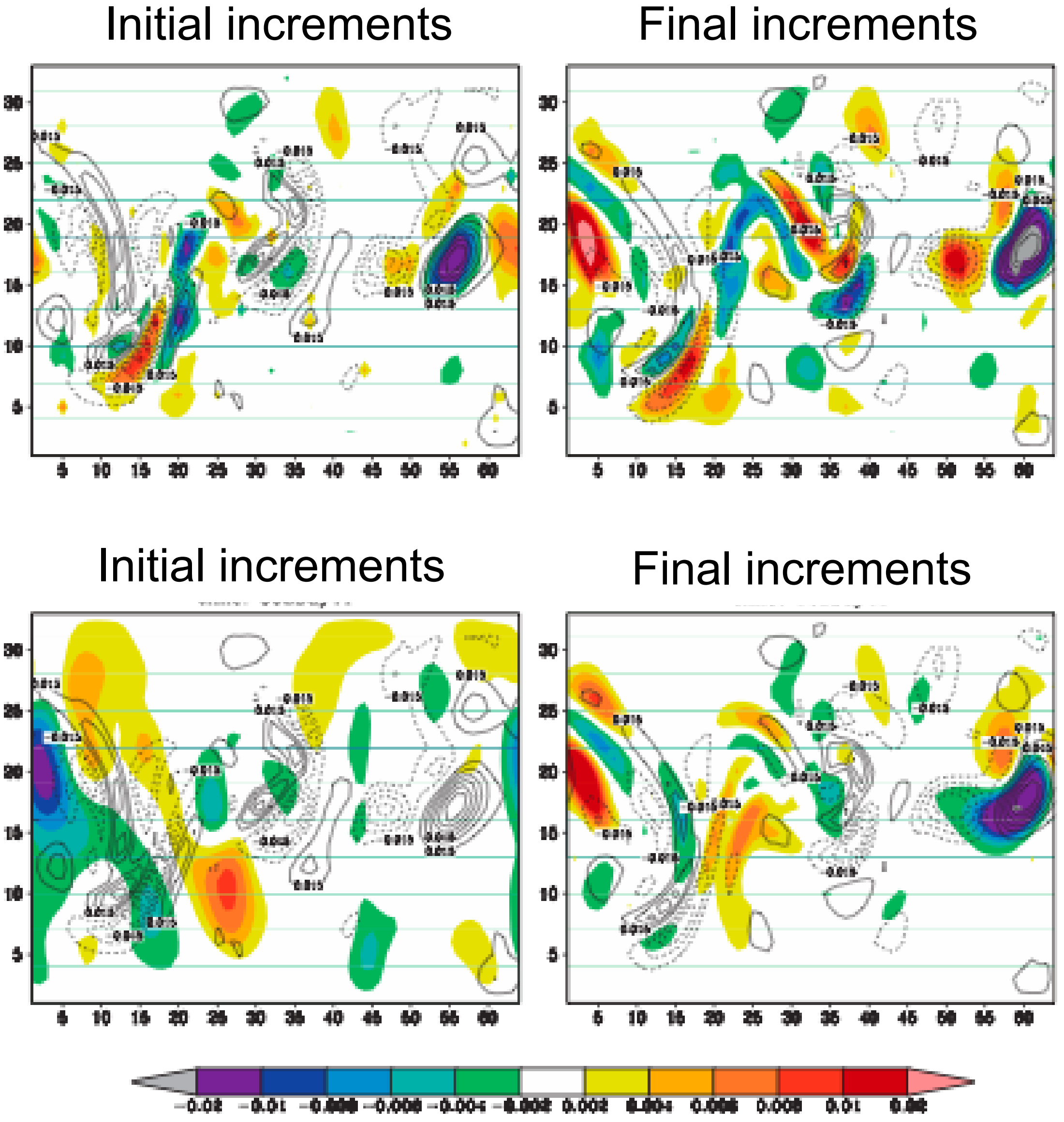
(e) Fcst error (4D12H)

(f) Ana error (4D12H)

LETKF

4D-Var-12hr

Initial and final analysis corrections (colors), with one BV (contours)



Initial increments

Final increments

LETKF

LETKF

Initial increments

Final increments

4D-Var-12hr

4D-Var-12hr

Summary

- Bred Vectors, like leading Lyapunov vectors are norm-independent.
- Initial Singular Vectors depend on the norm.
- 4D-Var is a smoother: it provides an analysis throughout the assimilation window.
- We can define a “No-cost” smoother for the LETKF.
- Applications: Outer Loop and “Running in Place”.
- Comparisons:4D-Var and LETKF better than 3D-Var.
- Analysis corrections in 3D-Var: missing errors of the day
- Analysis corrections in 4D-Var and LETKF are very similar at the end of the assimilation window.
- Analysis corrections at the beginning of the assimilation window look like bred vectors for the LETKF and like norm-dependent leading singular vectors for 4D-Var.

References

Kalnay et al., Tellus, 2007a (review)

Kalnay et al., Tellus, 2007b (no cost smoother)

Yang, Carrassi, Corazza, Miyoshi, Kalnay, MWR (2009) (comparison of 3D-Var, 4D-Var and EnKF, no cost smoother)

Yang and Kalnay, QJRMS, under revision, 2010. (Application of outer loop and RIP to handle nonlinearities and non-Gaussianities)

Kalnay and Yang, QJRMS, under revision, 2010. (Acceleration of EnKF spin-up, RIP)

Please see the “UMD Weather Chaos” web page for publications.