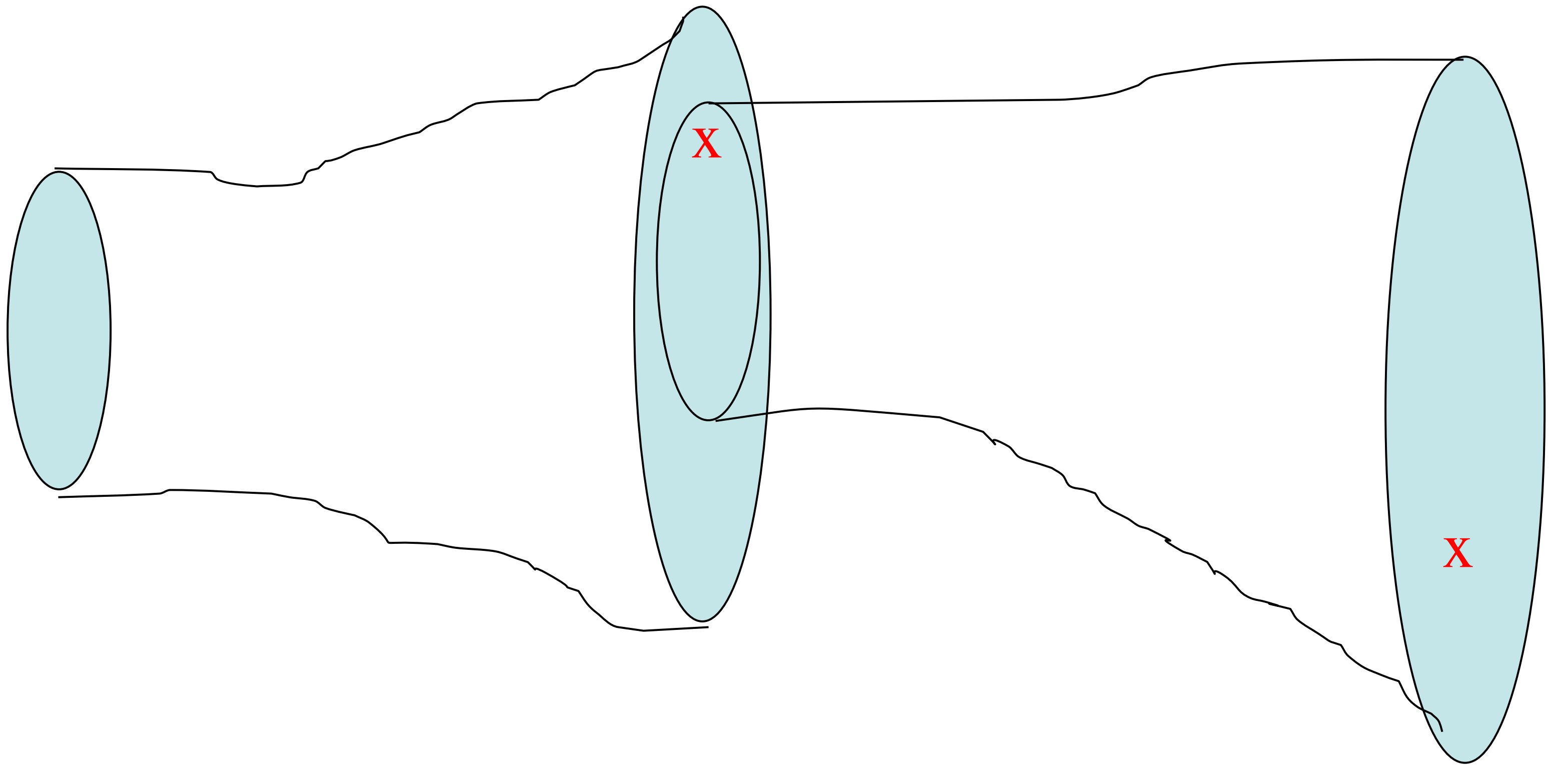
Hybrid Data Assimilation

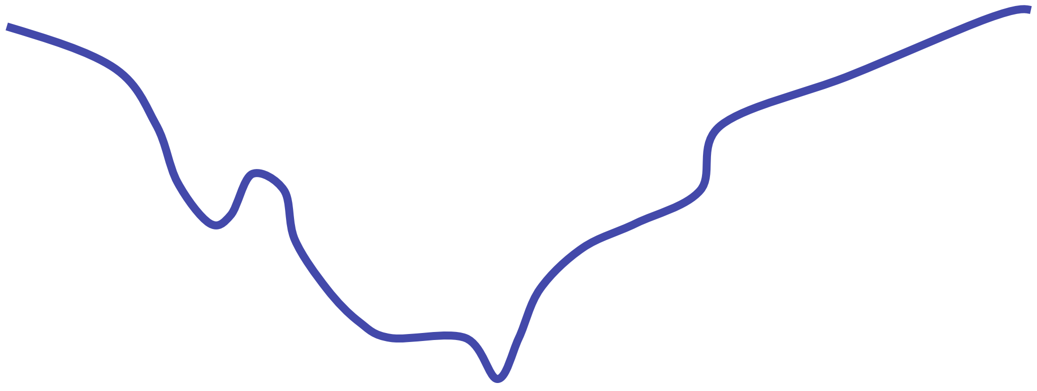
Peter Jan van Leeuwen
Data Assimilation Research Centre (DARC)
National Centre for Earth Observation
http://www.met.reading.ac.uk/~darc/
University of Reading
p.j.vanleeuwen@reading.ac.uk

Present-day data-assimilation methods for NWP:

- EnKF:



- 3/4DVar:



Issues with both methods:

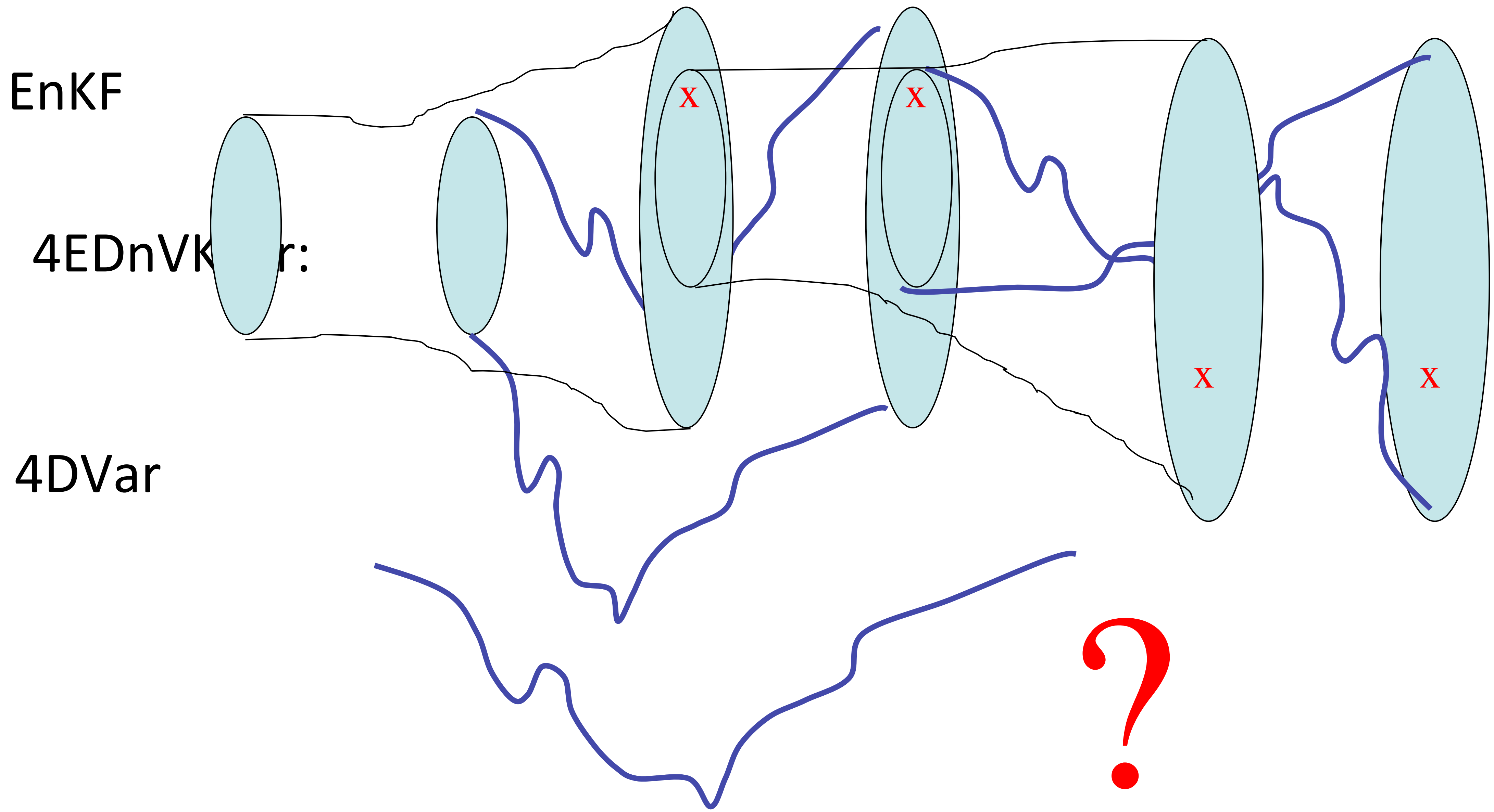
4DVar

1) Prior is assumed Gaussian
2) Previous observations inform first guess, not B matrix, so B matrix is static
3) No error estimate,
4) Separate ensemble prediction system
5) Possibility of getting stuck in local minima

EnKF

1) Prior is assumed Gaussian
2) Low-rank approximation to B, so needs localisation
3) Needs inflation
4) Less efficient when number of observations is large
5) 'Doesn't minimise anything'

Combinations of the two: Hybrid Methods



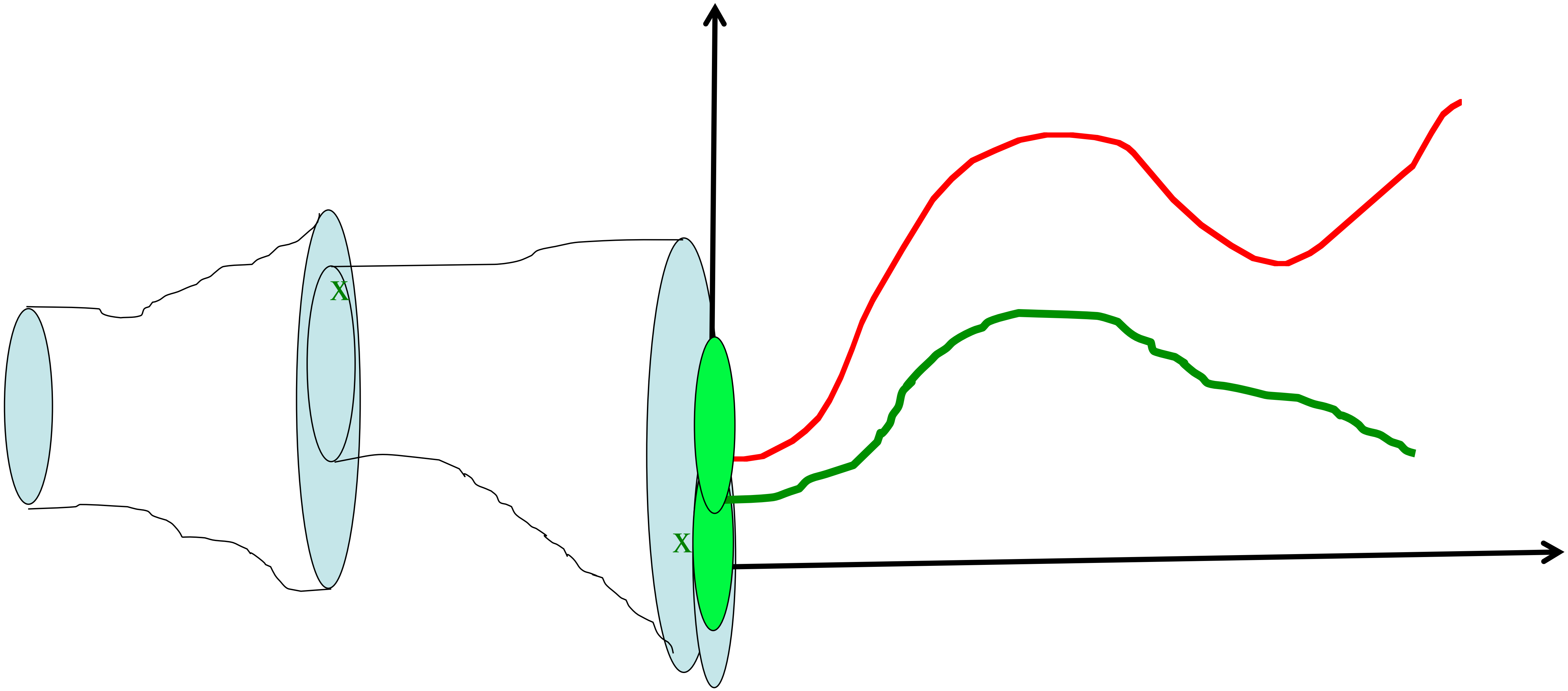
Operational Hybrid methods

1. ETKF-3/4DVar with control variable transform

2. Ens4DVar ‘ensemble of data assimilations’ EDA

3. 4DEnsVar

1. ETKF-3DVar



$$B = B_c$$

$$B = \beta_c^2 B_c + \beta_e^2 P^b \circ C_{loc}$$

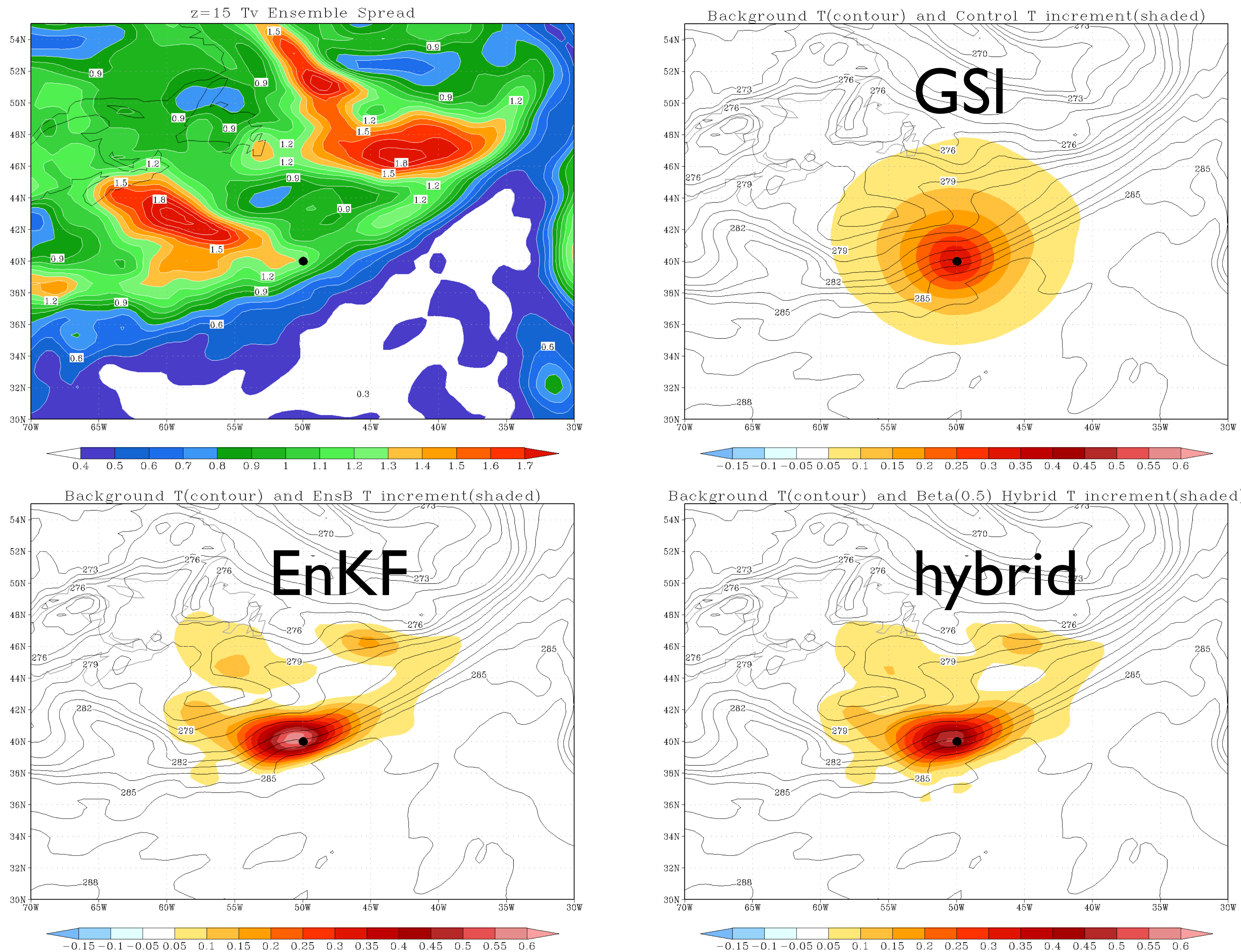
ETKF-3/4DVar algorithm

- Run ETKF to observation time (low resolution)
- Form new hybrid B matrix

$$B = \beta_c^2 B_c + \beta_e^2 P^b \circ C_{loc}$$

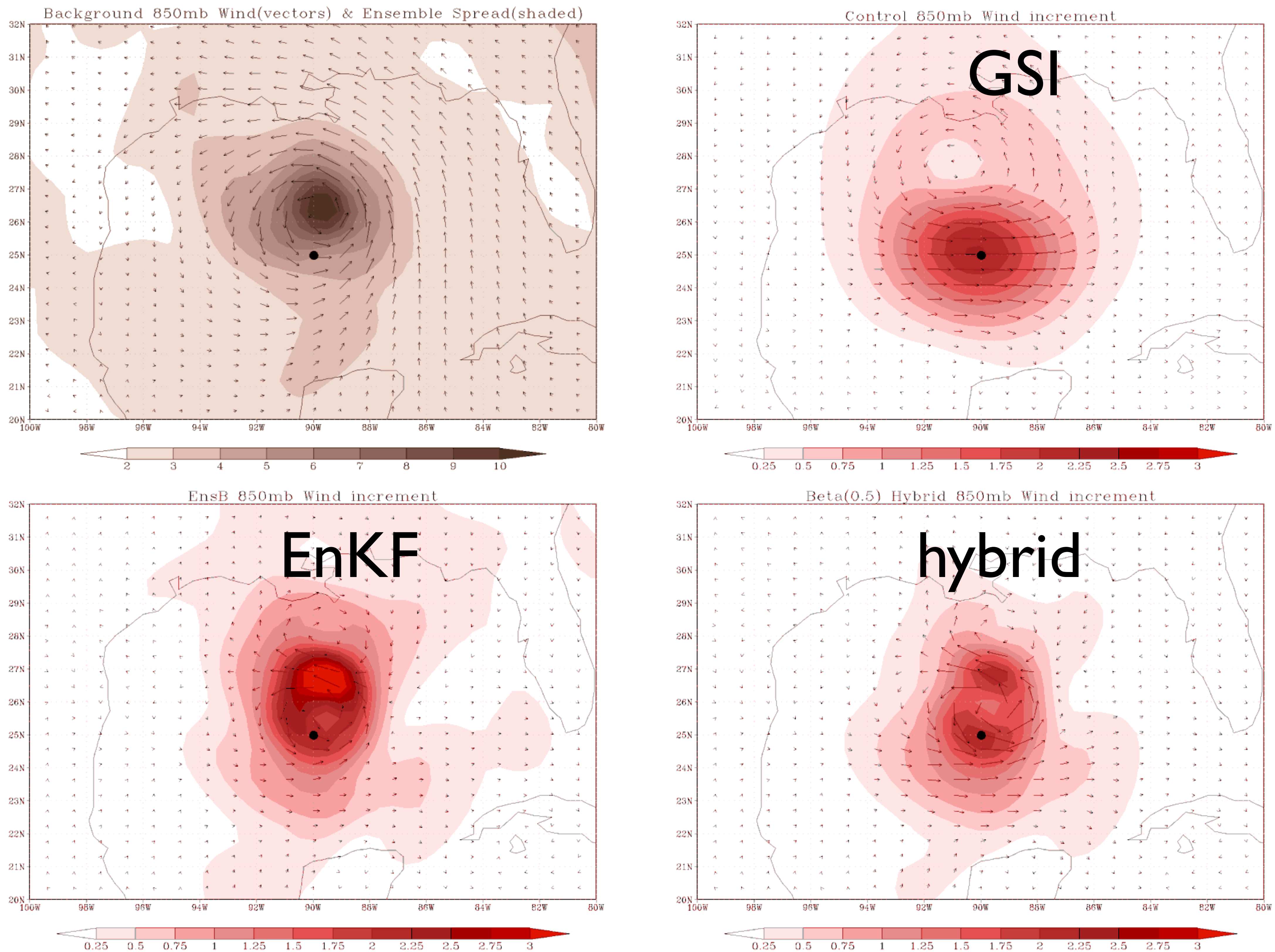
- Centre around 3/4Dvar forecast
- Precondition
- Calculate 3/4DVar solution(high resolution)
- Run ETKF to next observation time (low resolution)
- Etc.

Single-observation increments



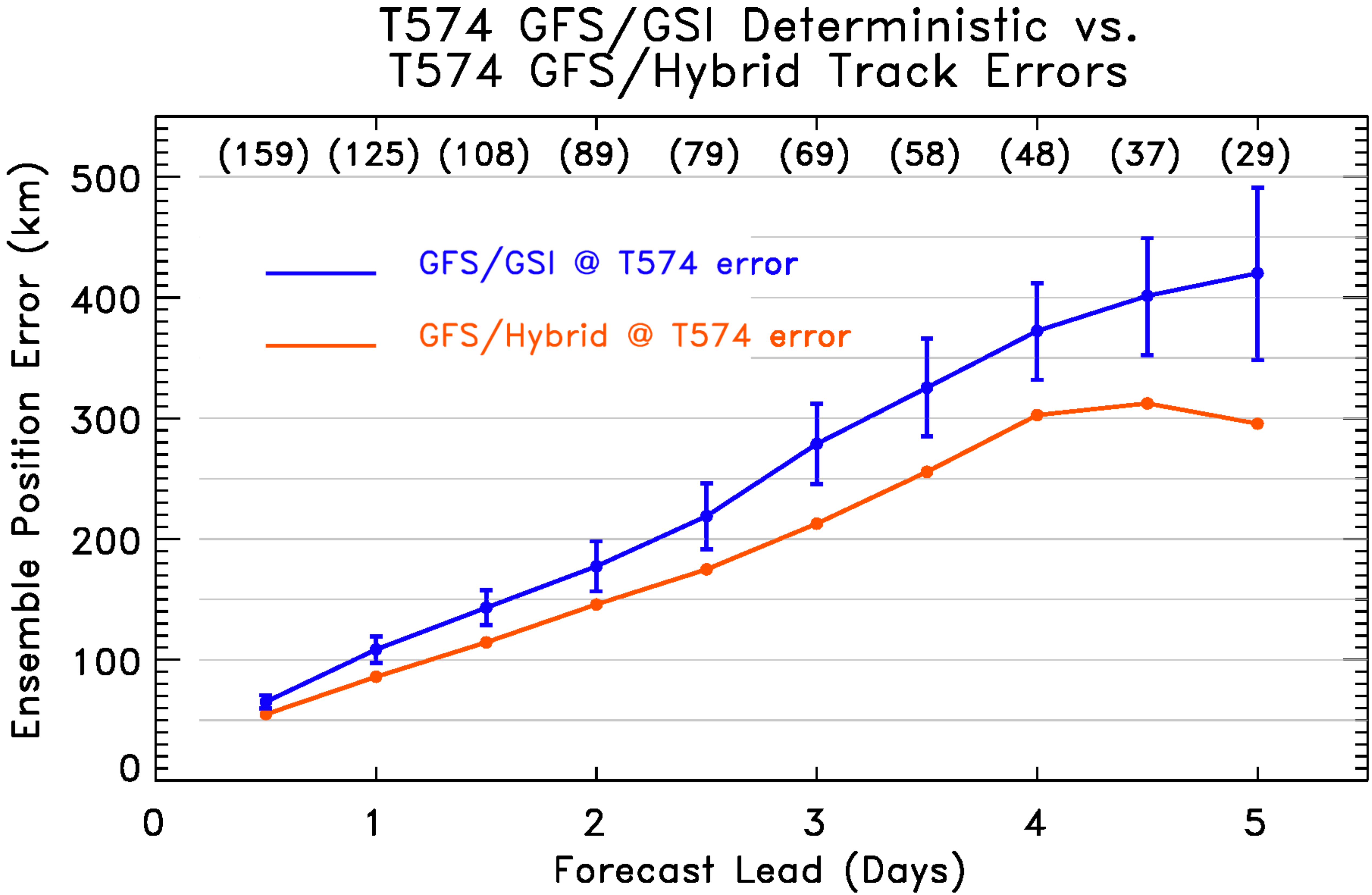
Single 850 hPa T_v observation (1K O-F, 1K error)

Single-observation increments



Single 850 hPa zonal wind observation (3 m/s O-F, 1m/s error) Hurricane Ike

The excellent news



Practical implementation

The 3DVar costfunction reads:

$$J(x) = \frac{1}{2}(x - x_b)^T B^{-1}(x - x_b) + \frac{1}{2}(y - H(x))^T R^{-1}(y - H(x))$$

with now: $$B = \beta_c^2 B_c + \beta_e^2 P^b \circ C_{loc}$$

Use alpha control variable transform:

$$\delta x = \beta_c B_c^{1/2} v + \beta_e X' \circ \alpha$$

Practical implementation

With this control variable transform

$$\delta x = \beta_c B_c^{1/2} v + \beta_e X' \circ \alpha$$

we find for the background term:

$$\frac{1}{2}\delta x^T B^{-1} \delta x =$$

$$\frac{1}{2}\left(\beta_c B_c^{1/2} v + \beta_e X' \circ \alpha\right)^T \left(\beta_c^2 B_c + \beta_e^2 P^b \circ C_{loc}\right)^{-1} \left(\beta_c B_c^{1/2} v + \beta_e X' \circ \alpha\right) =$$

$$= \frac{1}{2} v^T v + \frac{1}{2}\alpha^T C_{loc}^{-1} \alpha$$

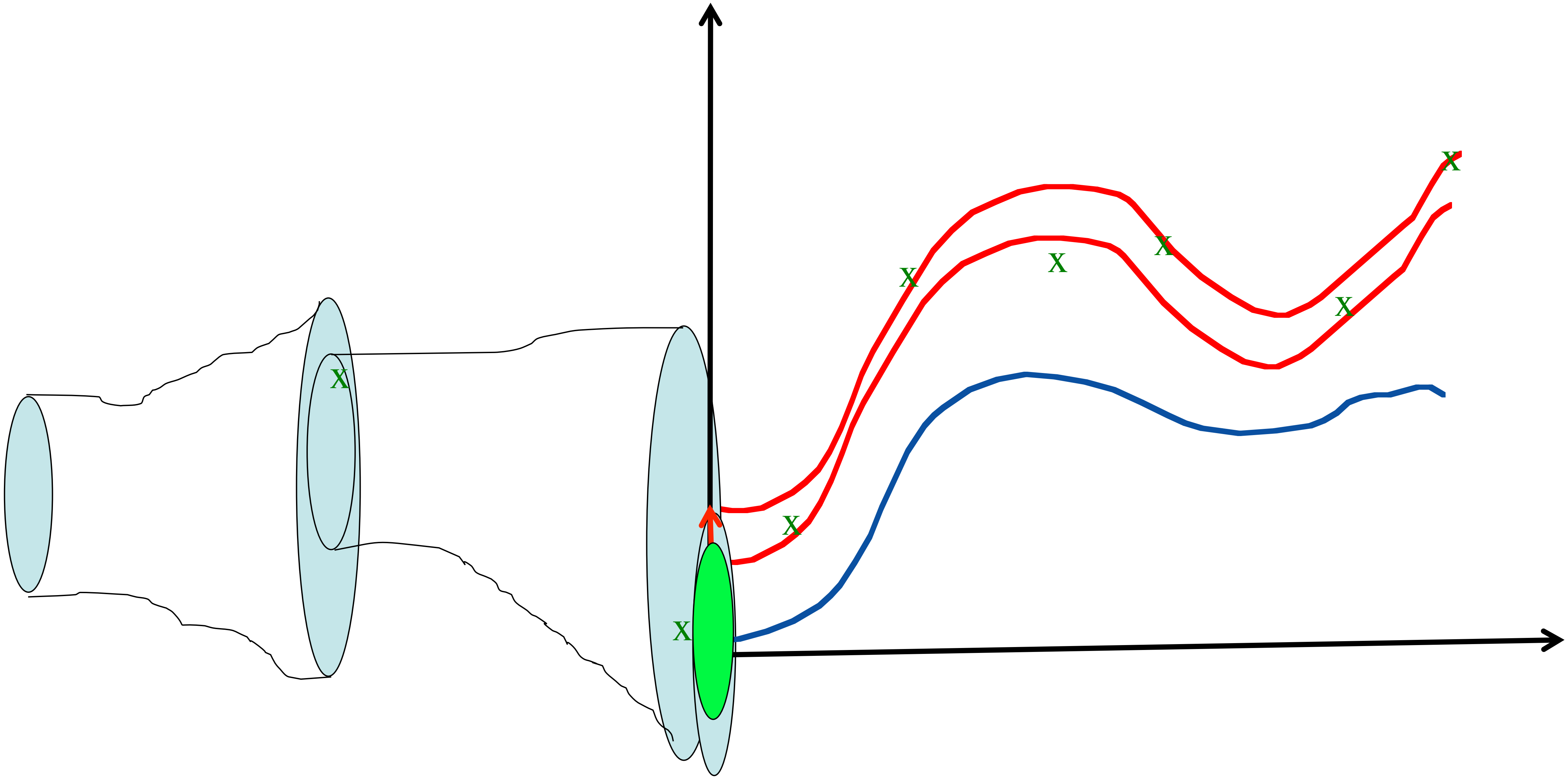
Meaning of control-variable transform

The control-variable transform can be written as:

$$\delta x = \beta_c B_c^{1/2} v + \beta_e X' \circ \alpha = \delta x_c + \delta x_e$$

- The increment is now a weighted sum of the static B_c component and the flow-dependent, ensemble based B_e
- The flow-dependent increment is a linear combination of ensemble perturbations X', modulated by the α fields
- If the α fields were homogeneous δx_e only spans Nens-1 degrees of freedom;
- We allow for varying α fields, which effectively increases the degrees of freedom
- C_{loc} is the localisation matrix for the flow-flow-dependent increments: it controls the spatial variation of α

ETKF-4DVar, e.g. Met Office



$$B = \beta_c^2 B_c + \beta_e^2 P^b \circ C_{loc}$$

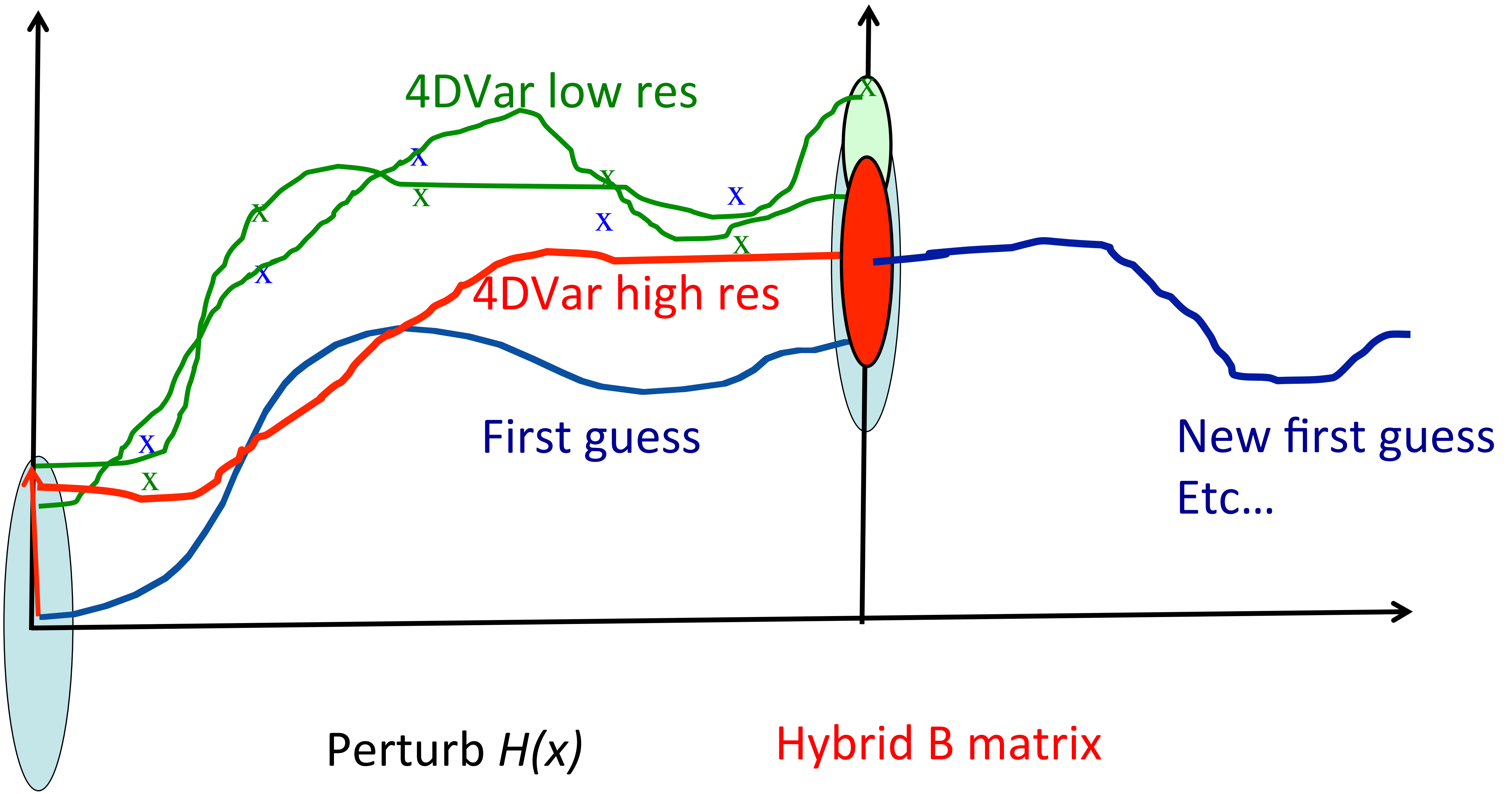
Practical implementation ETKF-4DVar

- Up to now this has only been implemented in strong-constraint 4Dvar
- Implementation similar to EnKF-3DVar.
- All usual tricks like preconditioning and incremental 4DVar are fully explored.

ETKF-4DVar

1) Prior is assumed Gaussian
2) B matrix is informed by previous observations, flow dependent B matrix
3) Error estimate from the EnKF ensemble
4) Natural ensemble prediction system
5) Possibility of getting stuck in local minima
6) Extra linearity by replacing ensemble mean by 4Dvar solution

2. EnsVars
'ensemble of data-assimilations' EDA



EnsVars algorithm

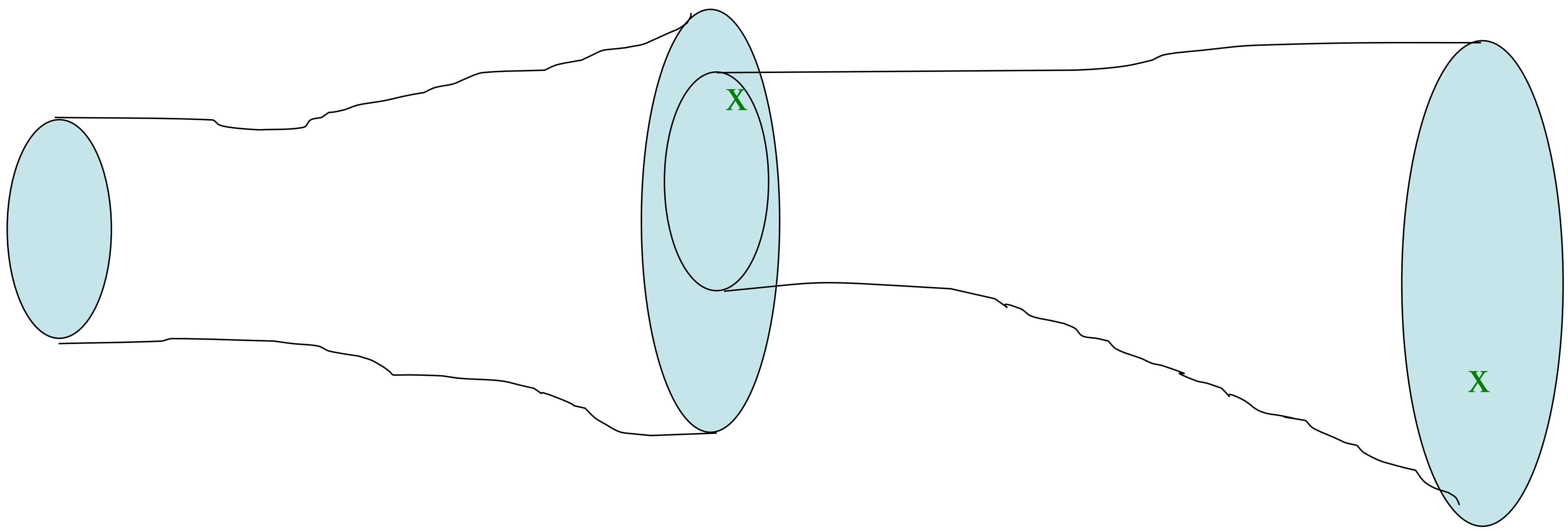
- Perturb *H(x)* in 4Dvar costfunction N_e times (‘perturb observations’)
- Solve 4DVar for each of these costfunctions (low resolution)
- Solve high-res 4Dvar over same window
- Use ensemble of 4Dvar ensemble forecasts to form hybrid B matrix for 4DVars of next time window.
- All usual tricks like preconditioning and incremental 4DVar are fully explored.

Characteristics of EnsVars

1) Prior is assumed Gaussian
2) B matrix is informed by previous observations, flow dependent B matrix
3) Error estimate from the low-res 4DVar ensemble
4) Natural ensemble prediction system
5) Possibility of getting stuck in local minima
6) Extra linearity by replacing ensemble mean by high-res 4Dvar solution

3. 4DEnsVar

Perform 4DVar without an adjoint.



Use ensemble to build space-time covariance as in an Ensemble Kalman smoother. Use this covariance in an incremental 4DVar.

Strong-constraint 4DEnsVar

The costfunction reads:

$$J(x_0) = \frac{1}{2}(x_0 - x_0^b)^T {P^b}^{-1}(x_0 - x_0^b) + \frac{1}{2}\sum_i \left(y_i - \mathcal{H}_i\left(\mathcal{M}_i(x_0)\right)\right)^T R^{-1}\left(y_i - \mathcal{H}_i\left(\mathcal{M}_i(x_0)\right)\right)$$

in which the background covariance comes from the ensemble:

$$P^b = X_0' {X_0'}^T$$

with:

$$X_0' = \frac{1}{\sqrt{N_e - 1}}\left[x_0^{b,(1)} - \overline{x_0^b}, \ldots, x_0^{b,(N_e)} - \overline{x_0^b}\right]$$

The gradient of the costfunction

$$\nabla J(x_0) = {P^b}^{-1}(x_0 - x_0^b) - \sum_i {P^b}^{-1} P^b M_i^T H_i^T R^{-1} \left(y_i - \mathcal{H}_i\left(\mathcal{M}_i(x_0)\right)\right)$$

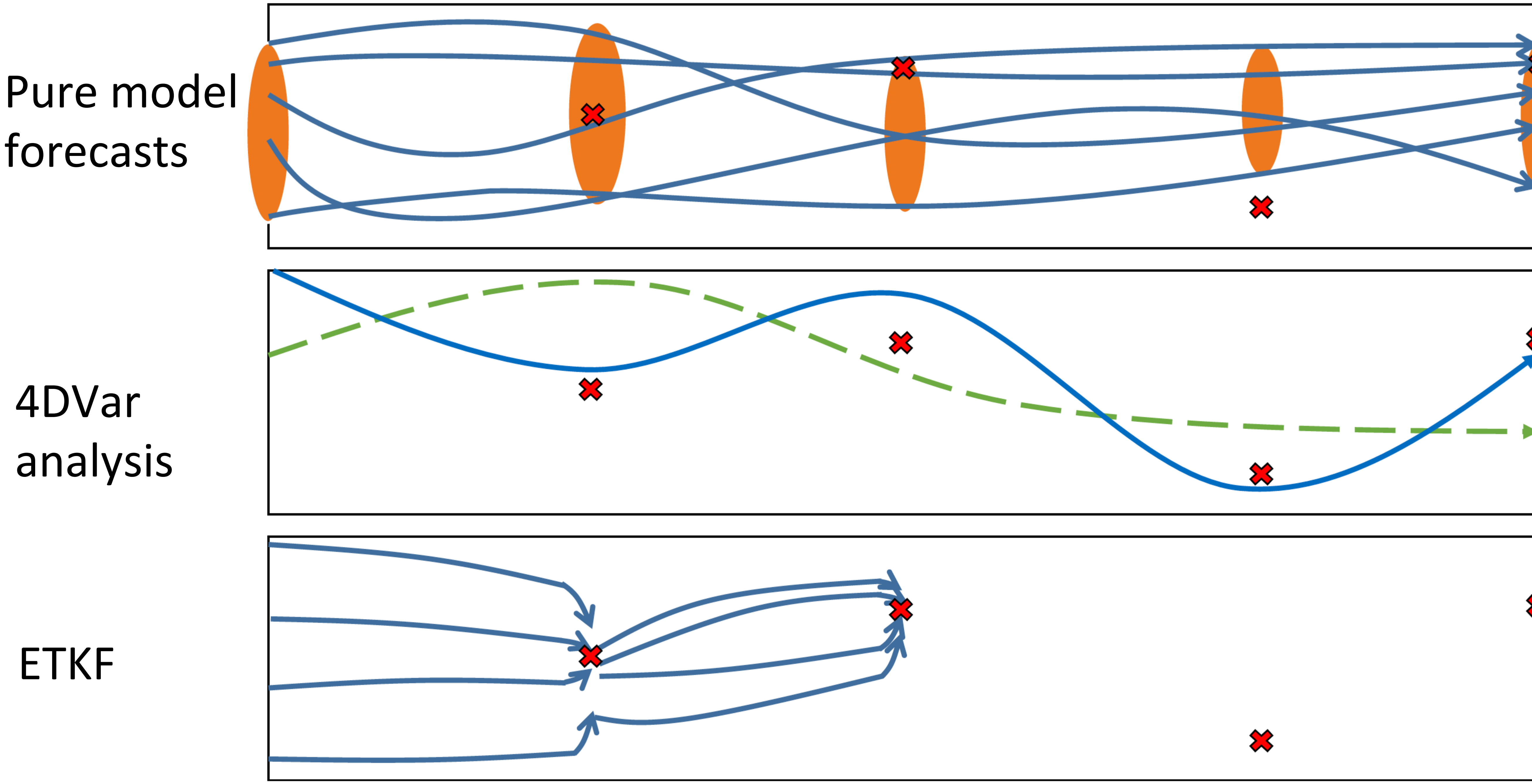
where we multiplied the observation term by ${P^b}^{-1} P^b = I$

Now write:

$$\begin{aligned} P^b M_i^T H_i^T &= X_0' {X_0'}^T M_i^T H_i^T \\ &= X_0' (H_i M_i X_0')^T \\ &= X_0' (H_i X_i')^T \\ &= X_0' (Y_i')^T \end{aligned}$$

Hence instead of the adjoint operator M_i^T we now have the measured ensemble at time *i*. No adjoint needed!

A schematic of 4DEnsVar

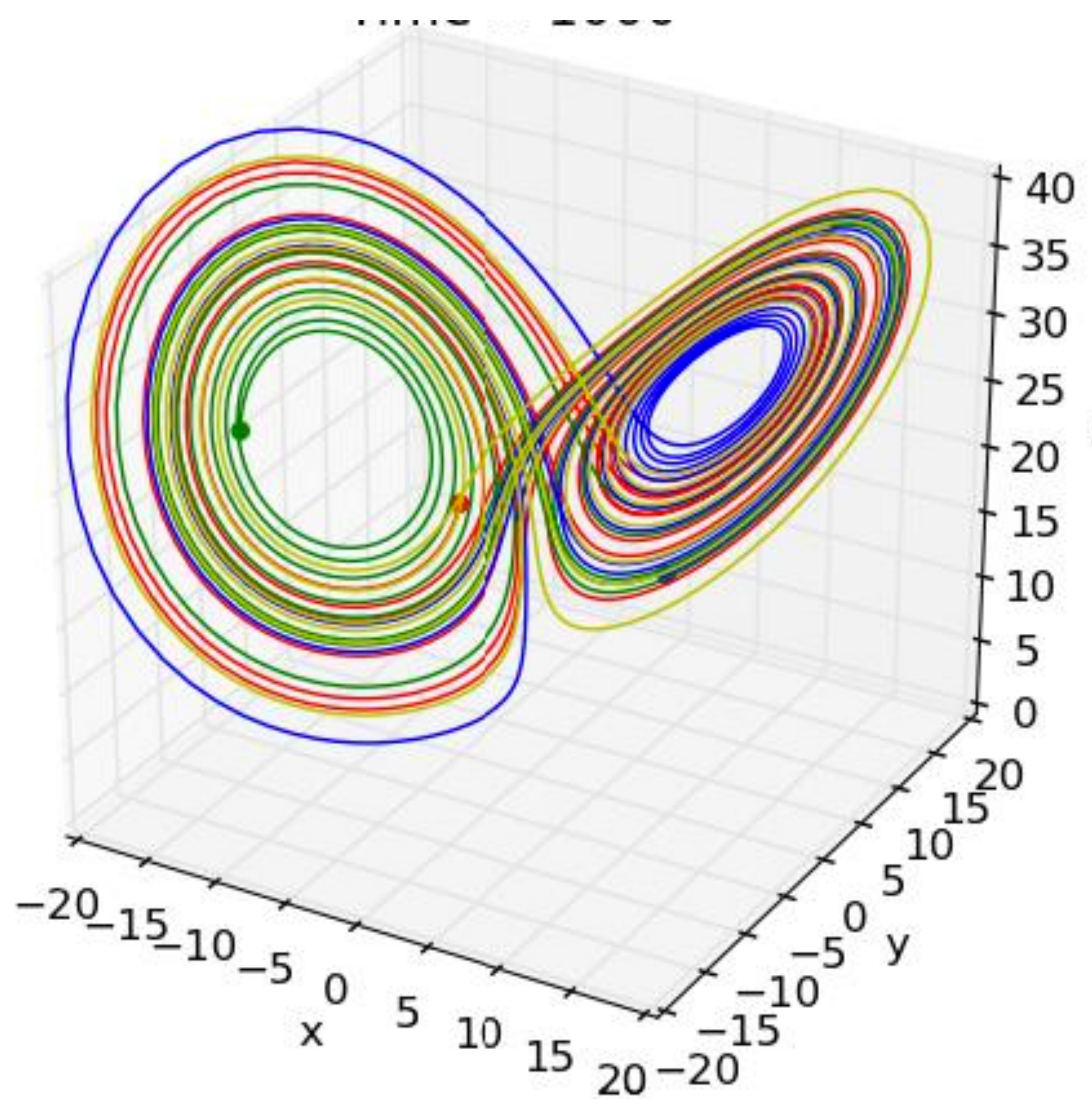


ETKF ensemble will provide new starting points for ensemble members in upper figure. Can be done in parallel with 4DVar.

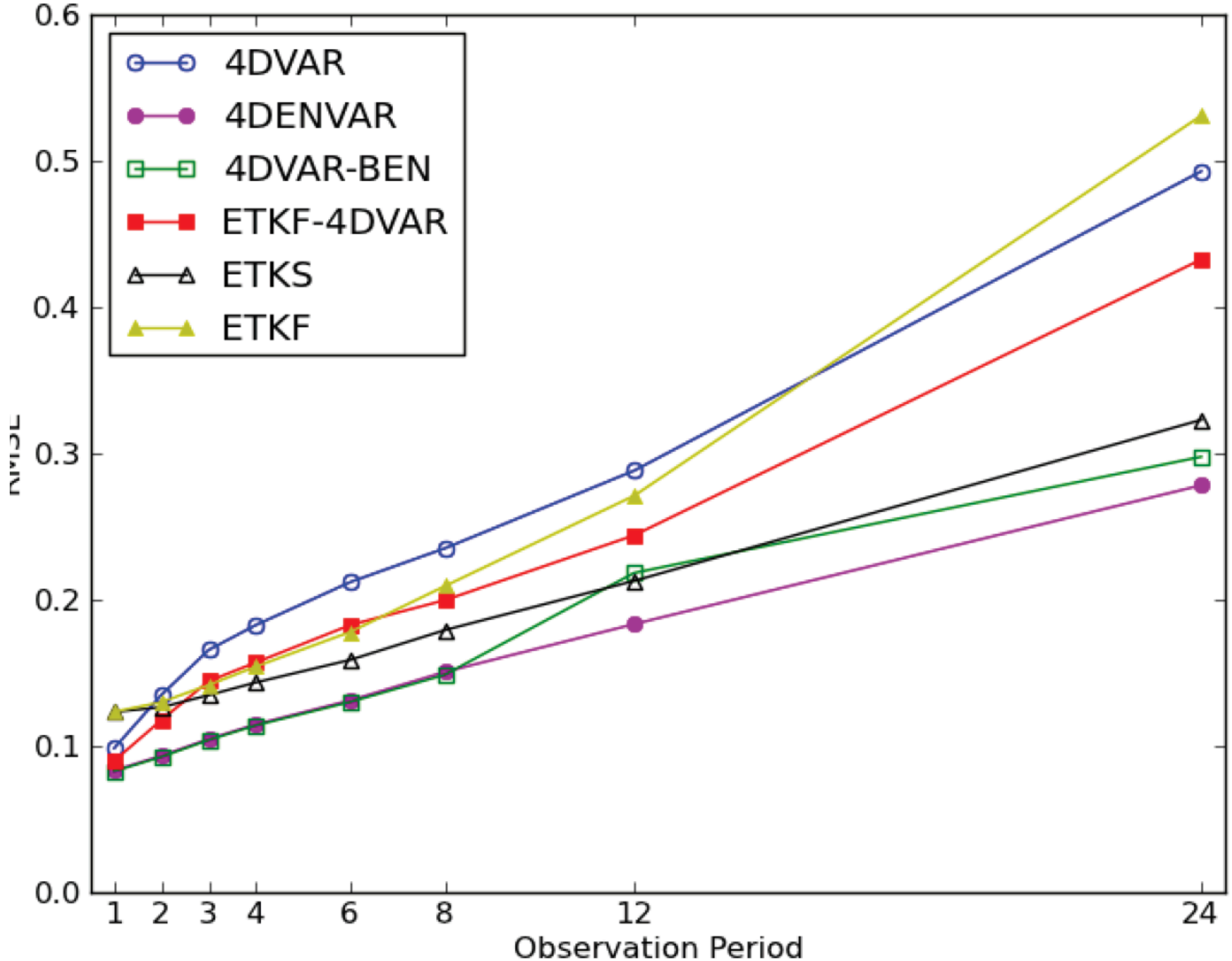
Example of Hybrid methods

Lorenz 1963 model
3 variables

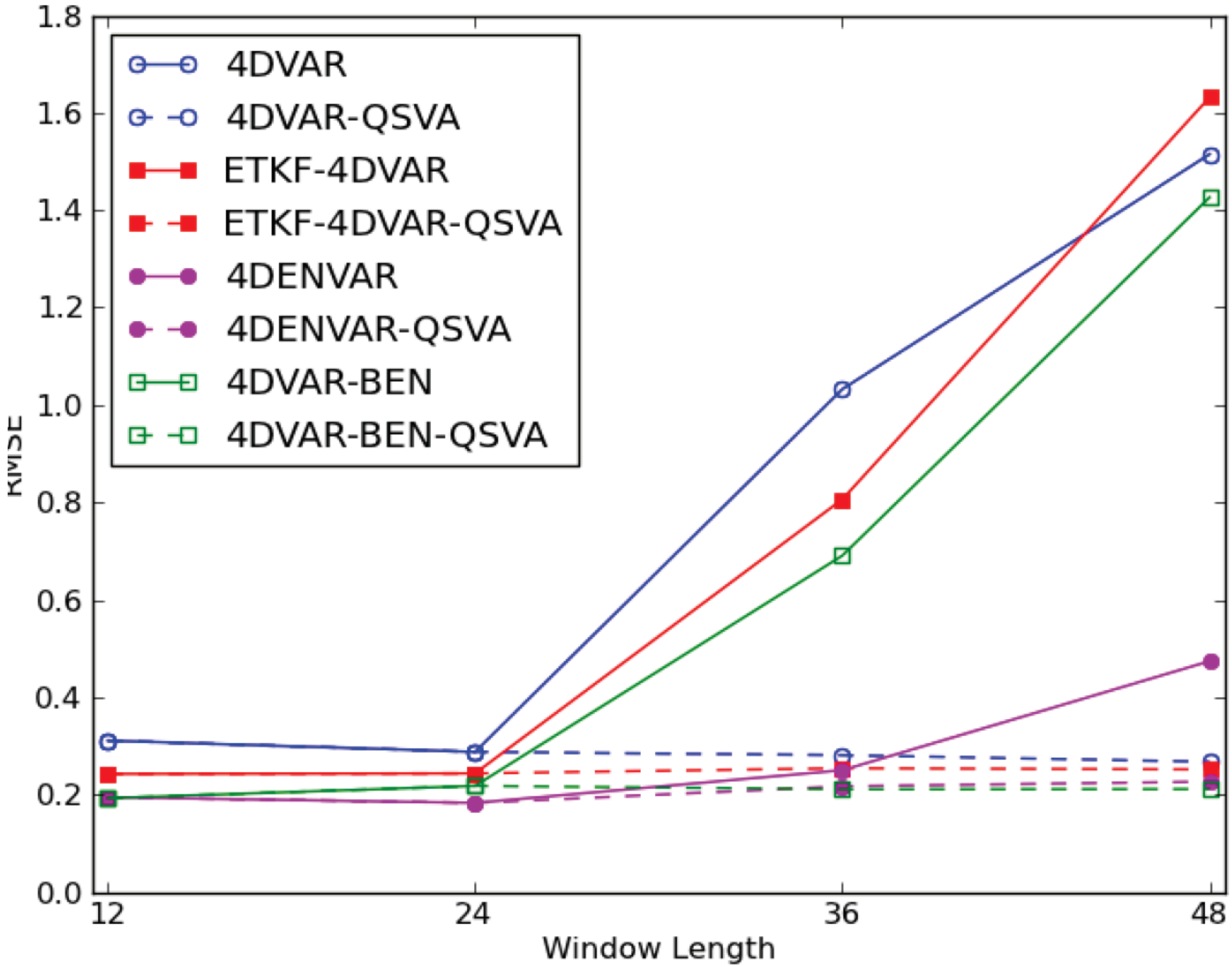
One revolution is
about 100 time steps



Comparison for a window length of 24 time steps



Comparison different window lengths with observation period 12 time steps



4DEnsVar on large systems

- When the system is high dimensional and the number of ensemble members is much smaller we need localisation.

- There is, however, an issue with localisation on ensemble perturbation matrices that are far apart in time, e.g.

$$X_0'(X_i')^T$$

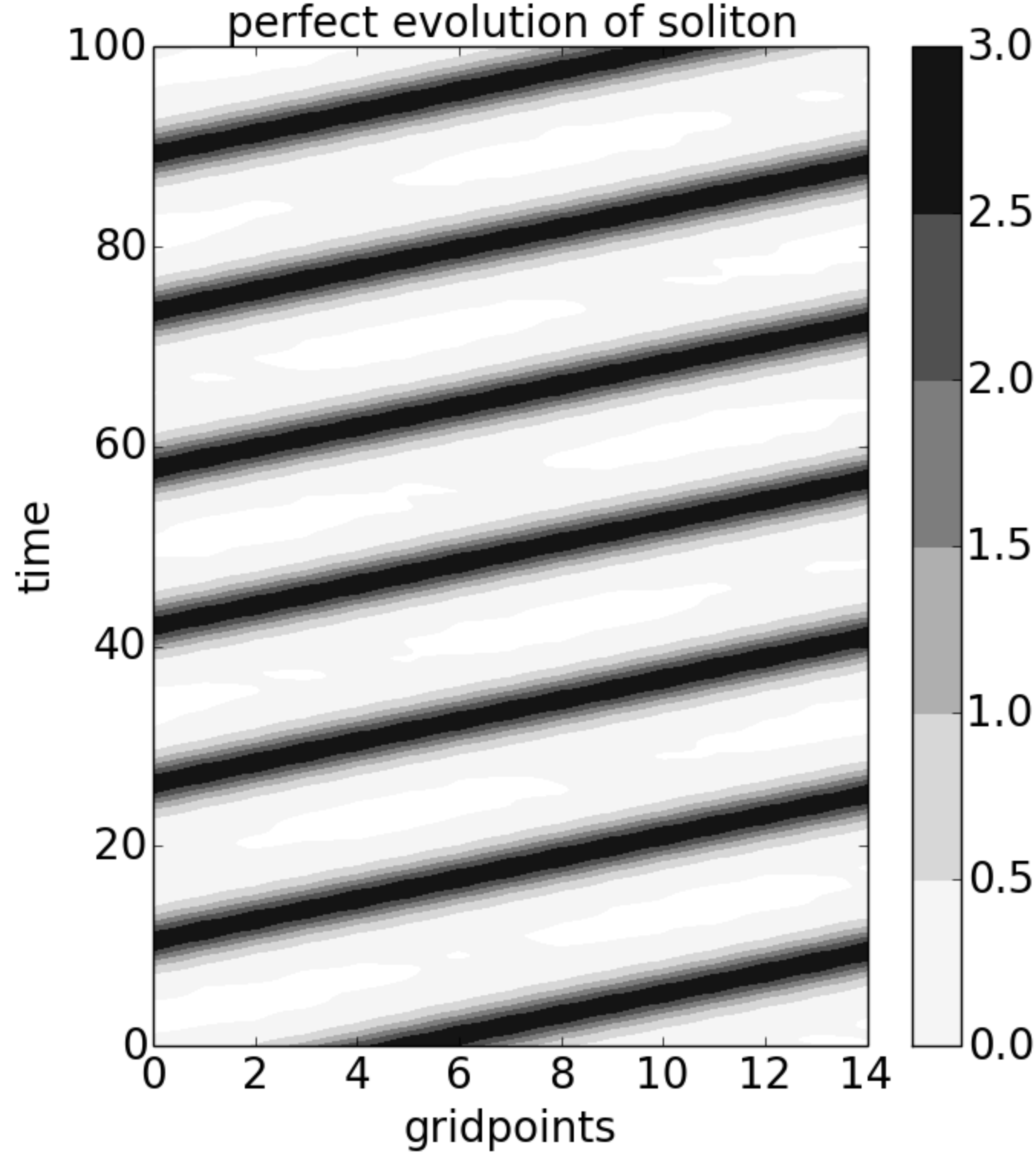
- At what time do we localise, at *t=0*, or *t=i* ?

Example: Korteweg-DeVries model

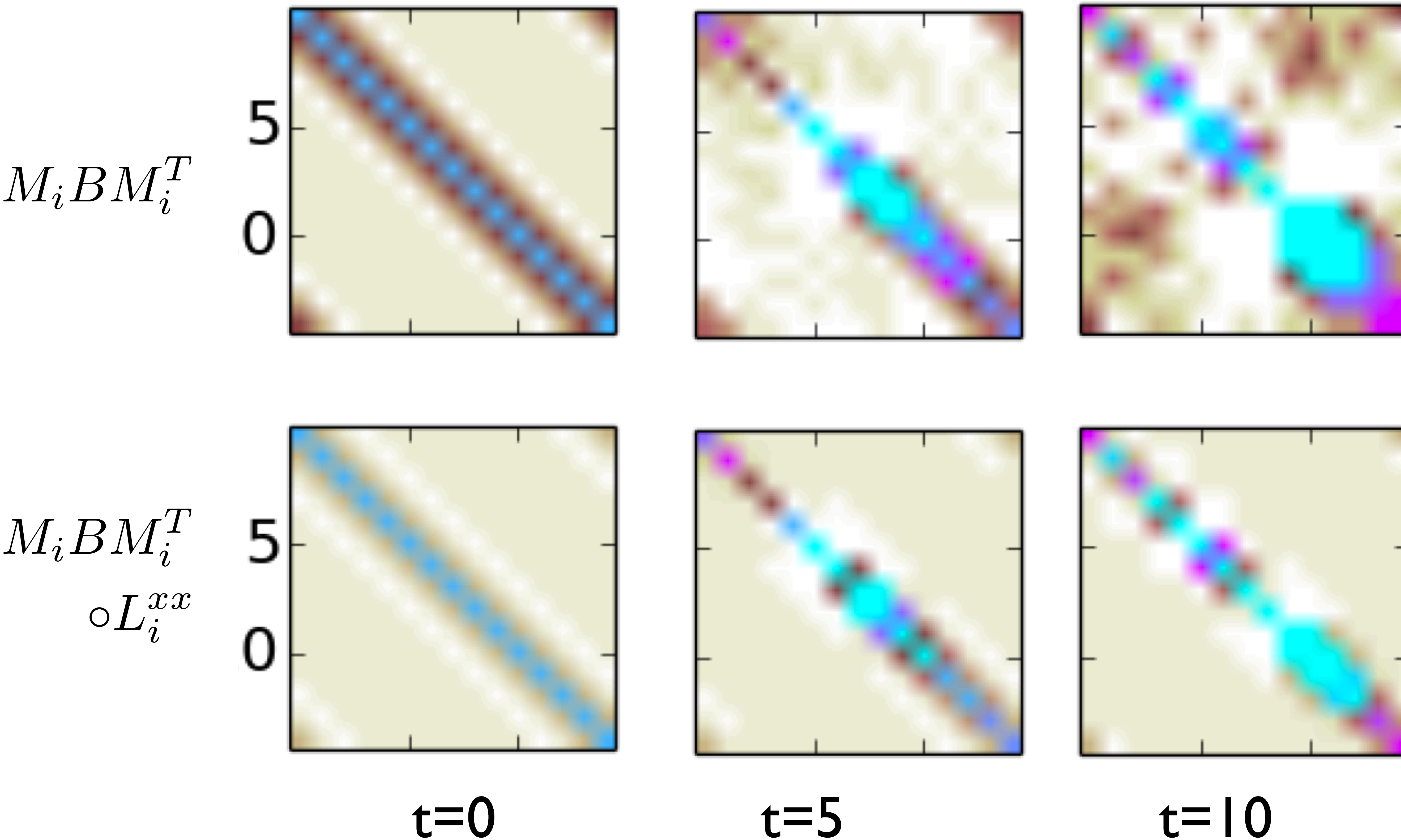
$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial s} + \frac{\partial^3 u}{\partial s^3} = 0$$

- Has soliton solutions, so solutions that don't change shape over time.
- We run it on a periodic domain and study how a covariance matrix is evolved by this system.

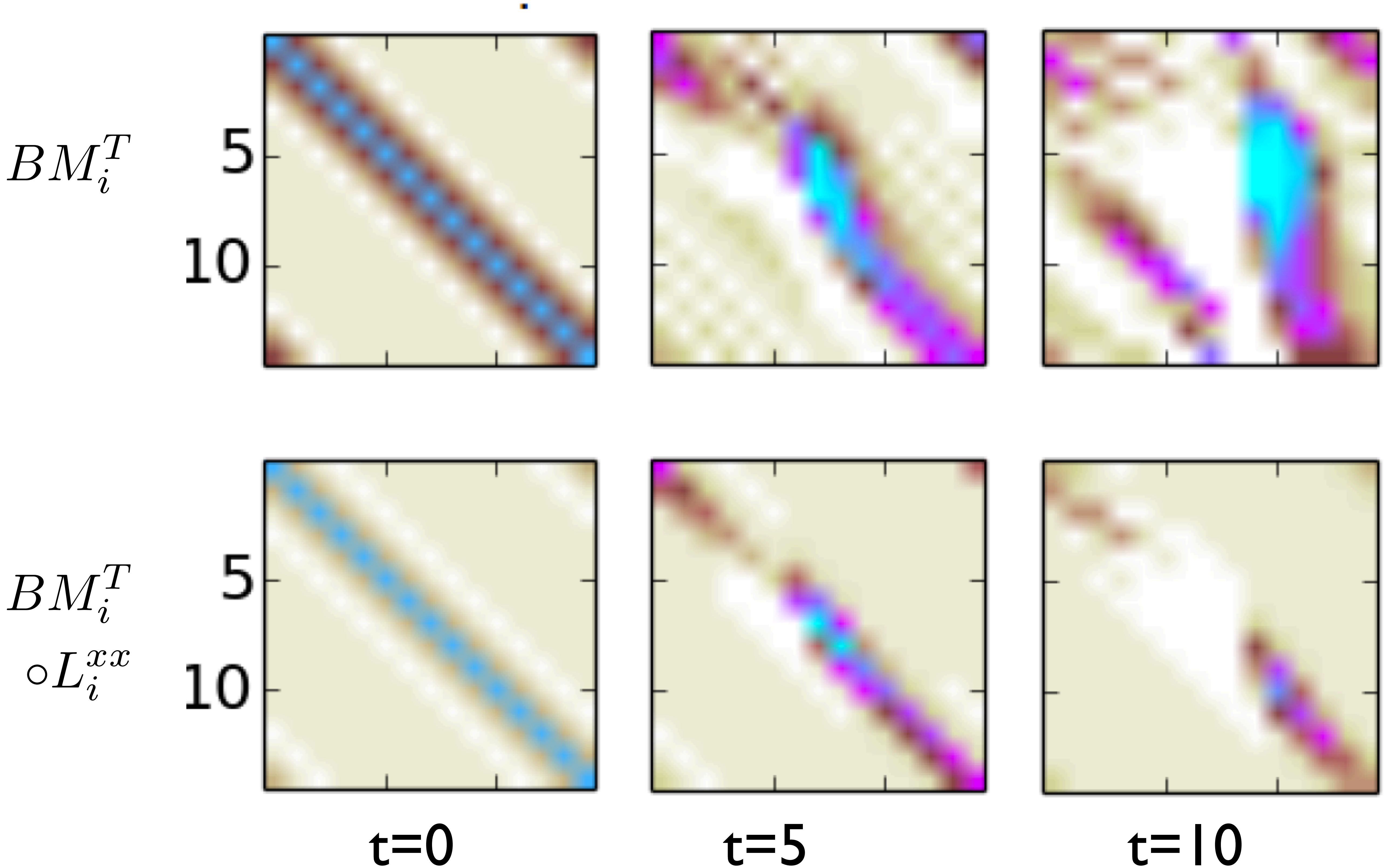
Example propagation of soliton with KdV equation.



Propagation of B with KvD model

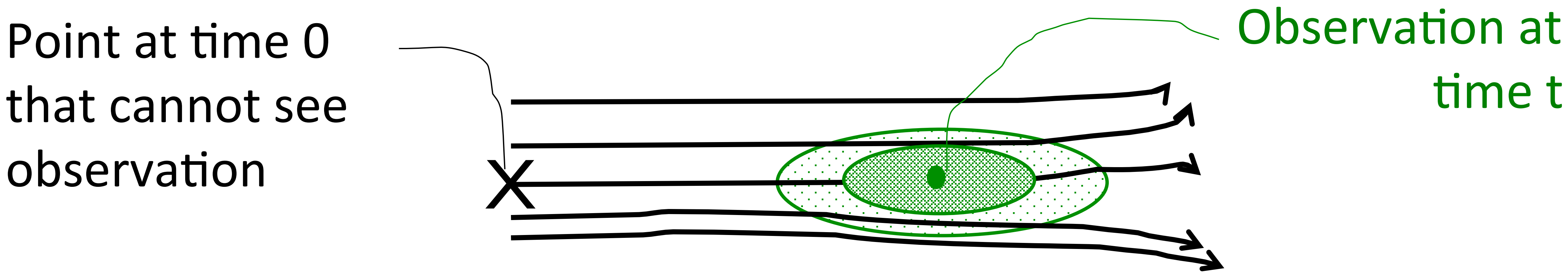


Propagation of B with KdV model



Localisation issues

When the assimilation window is long and winds are strong observations at the end of this window cannot influence the State at the beginning of the time window:

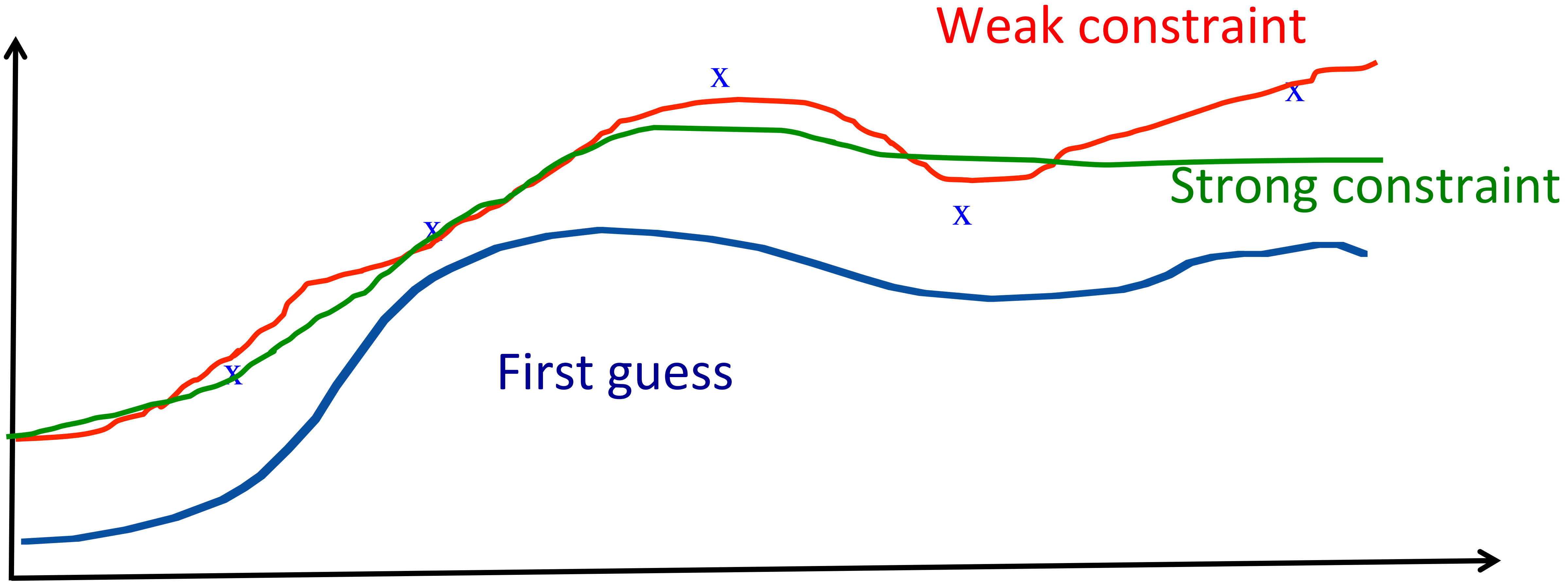


Two solutions have been proposed:

1. Advecting the localisation area with the flow (e.g. Meteo France)
2. Using a weak-constraint formulation

Weak-constraint variational methods

A weak-constraint variational method might partially solve the localisation issue as the observations influence the state local in time, and not only at the beginning of the time window:



Weak-constraint 4DEnsVar

The costfunction reads:

$$J(x_{0:T}) = \frac{1}{2}(x_0 - x_0^b)^T {P^b}^{-1}(x_0 - x_0^b) + \frac{1}{2}\sum_j (x_j - \mathcal{M}_j(x_{j-1}))^T Q^{-1}(x_j - \mathcal{M}_j(x_{j-1})) + \frac{1}{2}\sum_i (y_i - \mathcal{H}_i(x_i))^T R^{-1} (y_i - \mathcal{H}_i(x_i))$$

where we introduced the model error term.

The gradient of the costfunction

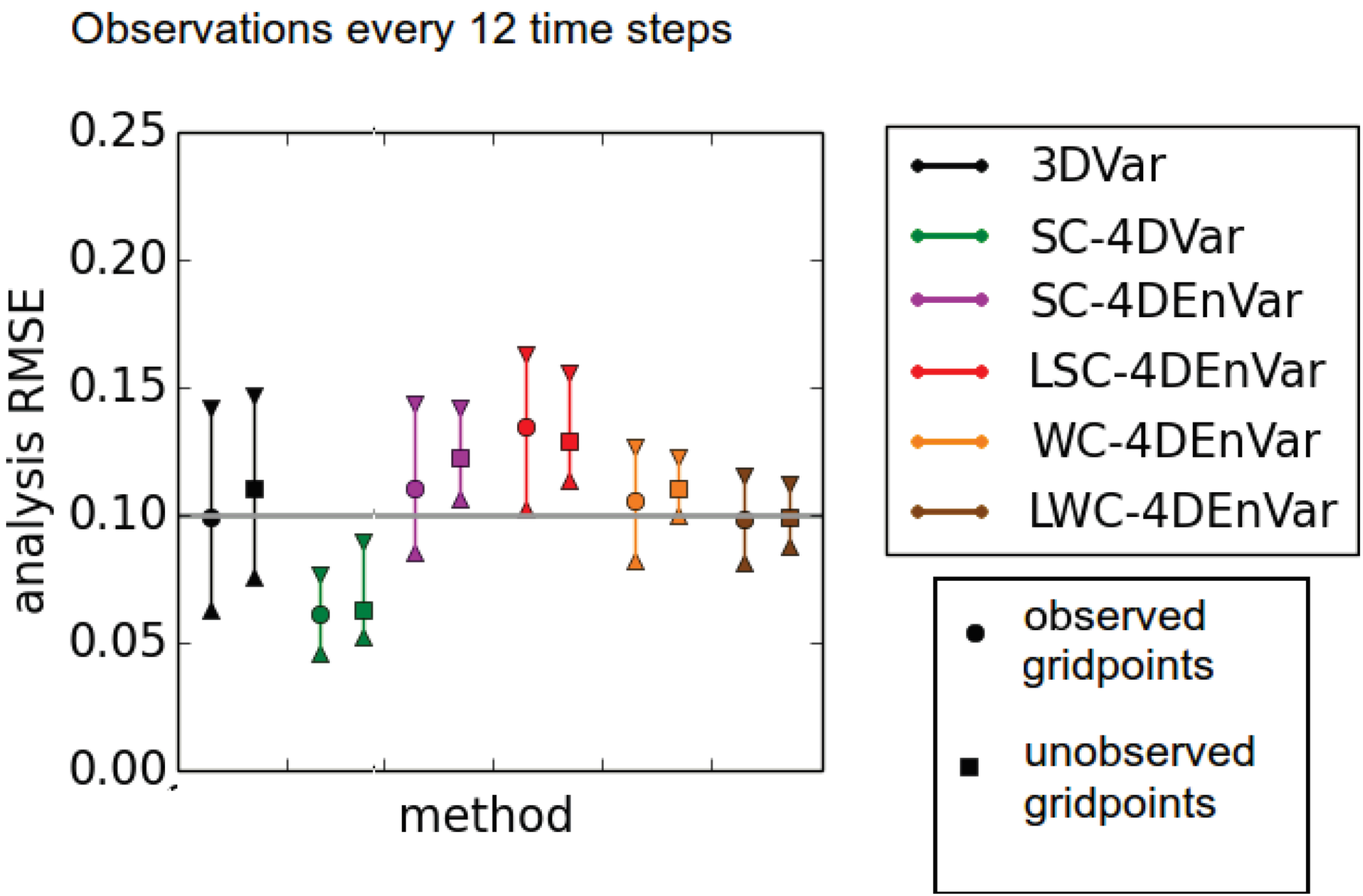
$$\nabla J(x_i) = -{P^b}^{-1} P^b M_{i+1}^T Q^{-1}(x_{i+1} - \mathcal{M}_{i+1}(x_{i-1}))$$
$$+ Q^{-1}(x_i - \mathcal{M}_i(x_{i-1}))$$
$$- {P^b}^{-1} P^b M_i^T H_i^T R^{-1}\left(y_i - \mathcal{H}_i(x_i)\right)$$

We can again use: $$P^b M_i^T H_i^T = X_0'(Y_i')^T$$

and also $$P^b M_{i+1}^T Q^{-1} = X_0'(X_{i+1}')^T Q^{-1}$$

to avoid the adjoint operator M_i^T. So again no adjoint needed!

Comparison of methods on KdV model

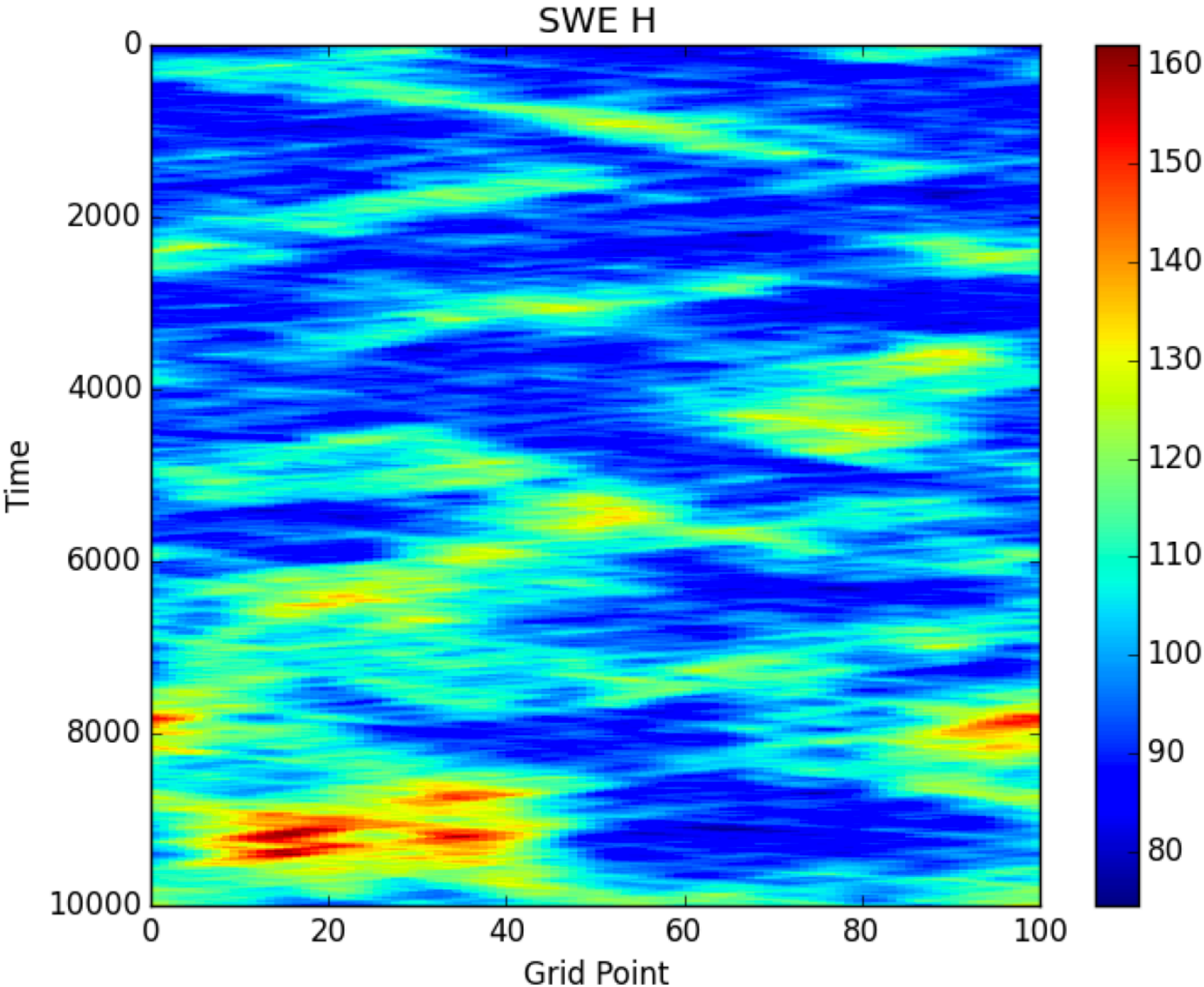


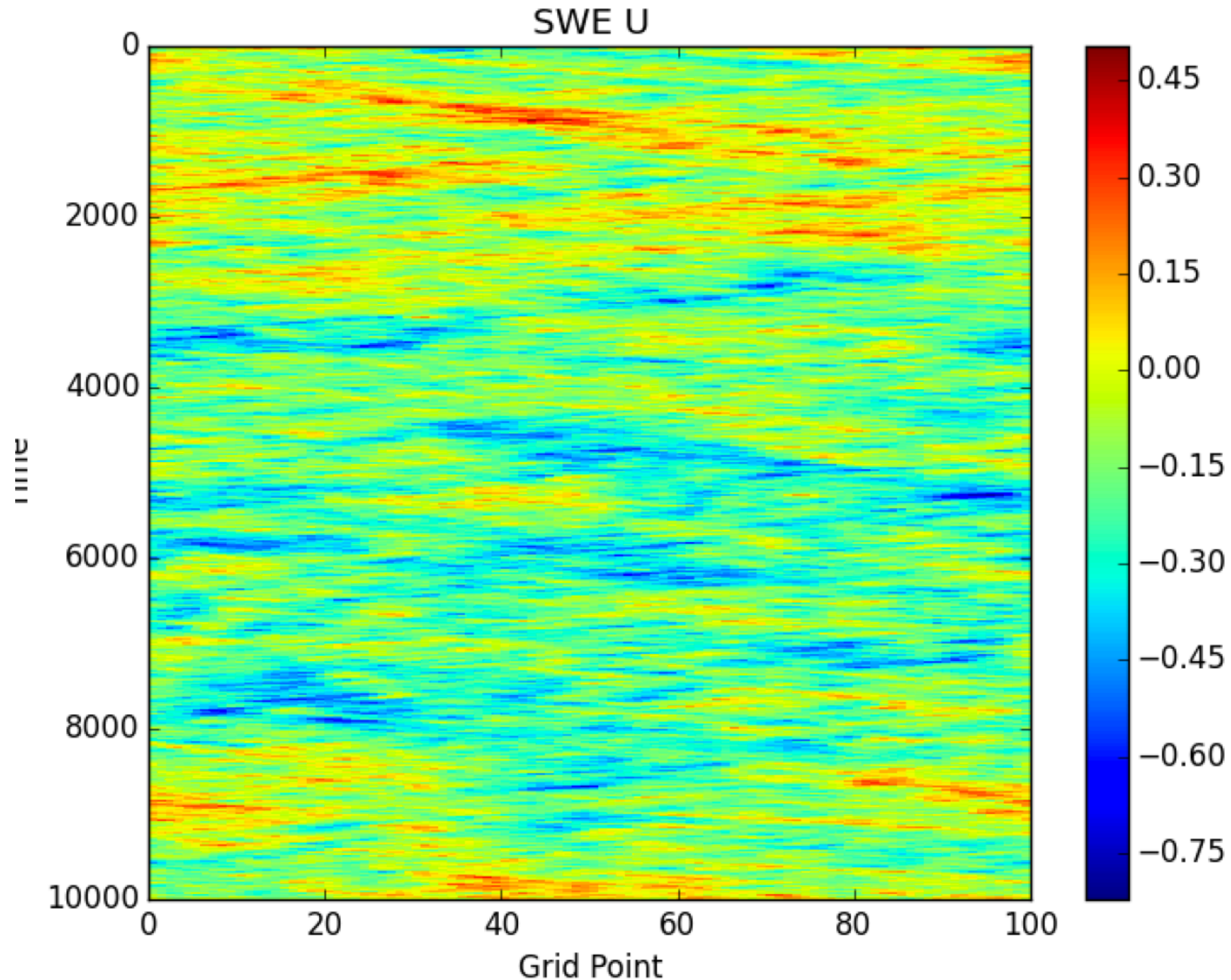
A simple model with convection

momentum
$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial s} + \frac{\partial \phi + r}{\partial s} = K\frac{\partial^2 u}{\partial s^2} + F$$

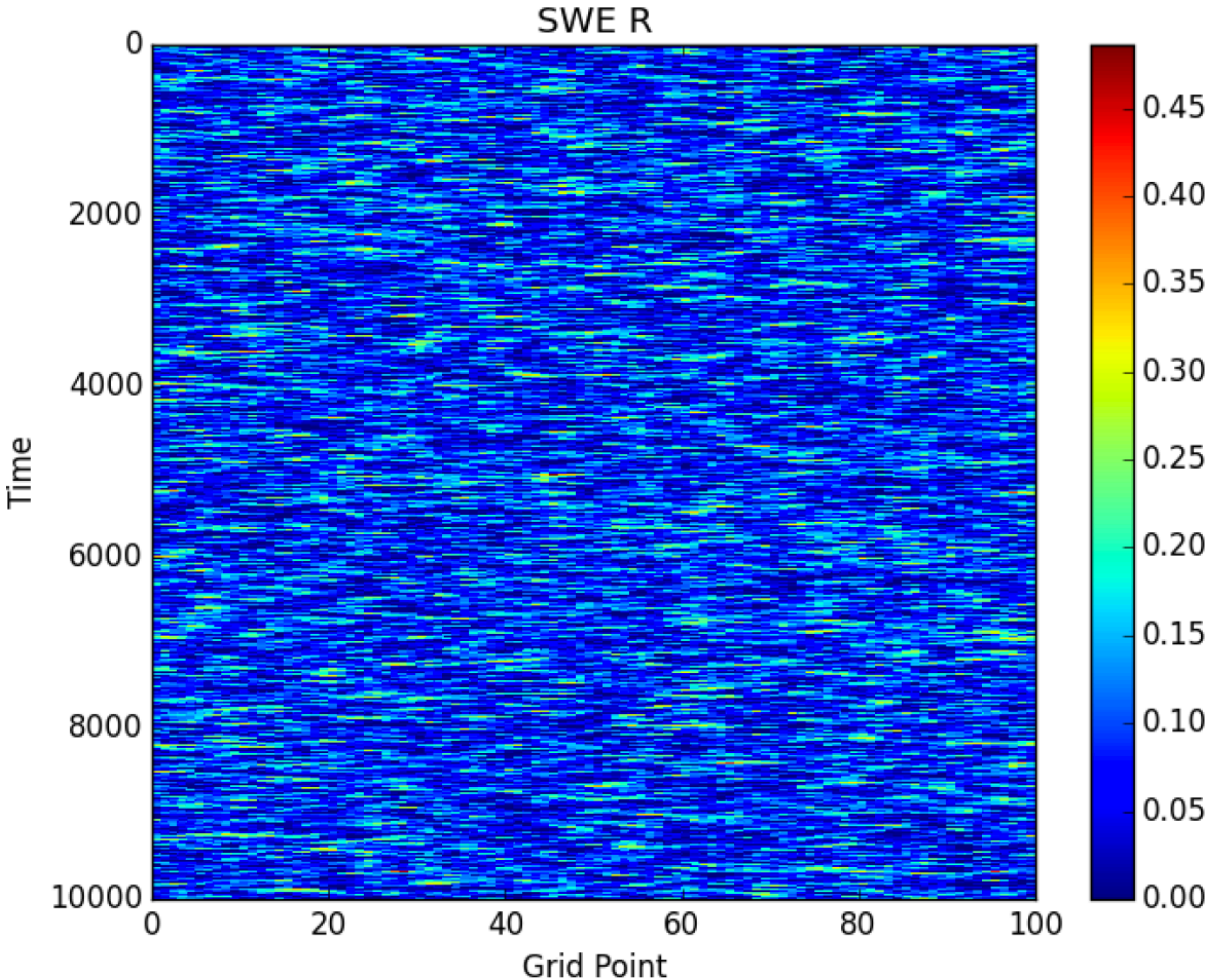
mass
$$\frac{\partial h}{\partial t} + \frac{\partial u(H+h)}{\partial s} = K\frac{\partial^2 h}{\partial s^2}$$

rain rate
$$\frac{\partial r}{\partial t} + u\frac{\partial r}{\partial s} = K_r\frac{\partial^2 r}{\partial s^2} - \alpha r - \beta\frac{\partial u}{\partial s}$$


SWE H
Time
Grid Point
0
2000
4000
6000
8000
10000
0
20
40
60
80
100
160
150
140
130
120
110
100
90
80

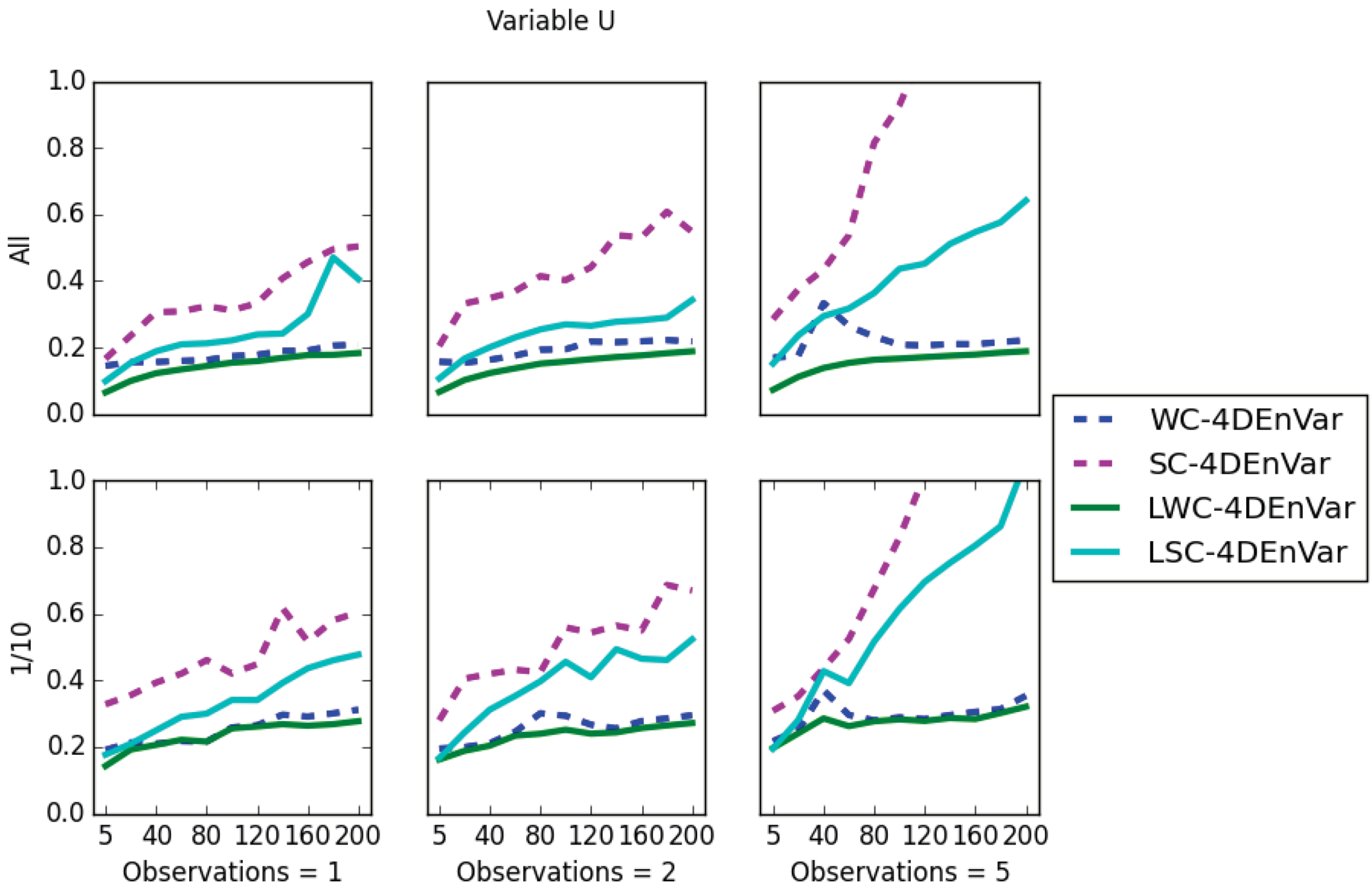


SWE U
0
2000
4000
6000
8000
10000
Time
0
20
40
60
80
100
Grid Point
0.45
0.30
0.15
0.00
−0.15
−0.30
−0.45
−0.60
−0.75



SWE R
Time
Grid Point
0
2000
4000
6000
8000
10000
0
20
40
60
80
100
0.45
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00

Results for different 4DEnsVar variants



Conclusions hybrid methods

Variational methods are the most popular data-assimilation scheme, but use static B matrix and need adjoint.

Ensemble-Kalman Filters are used to make B flow dependent: ETKF-4DVar, EnsVars (EDA), 4DEnsVar. The latter doesn't need an adjoint.

Strong-constraint 4DEnsVar, in which the space-time covariances are generated by an ensemble run has issues with localisation.

Weak-constraint 4DEnsVar partially solves localisation issue.

MORE WORK IS NEEDED !