PDAF Parallel Data Assimilation Framework

ALFRED-WEGENER-INSTITUT HELMHOLTZ-ZENTRUM FÜR POLAR- UND MEERESFORSCHUNG

Ensemble-Based Data Assimilation: Concepts, Methods, and Hands-On Tutorials

Lecture 2: Ensemble Data Assimilation

Lars Nerger

Alfred Wegener Institute
Helmholtz Center for Polar and Marine Research
Bremerhaven, Germany

HELMHOLTZ
SPITZENFORSCHUNG FÜR GROSSE HERAUSFORDERUNGEN

MarDATA Workshop on Data Assimilation, Bremerhaven, July 24-25, 2024

Overview – Lecture 2

Introduce to sequential data assimilation and Kalman filters from the traditional linear Kalman filter to basic ensemble-based Kalman filters

- (Extended) Kalman filter
- Low-rank Kalman filters
- Ensemble Kalman filters
- Square-root Kalman filters

BLUE

Influence of observation error

BLUE as Statistical Estimate with Model

Best linear unbiased estimate for

Model prediction: $x^b = x^t + e_b$ with: $\mathrm{var}(e_b) = \sigma_b^2$

Observation: $y = x^t + e_o$ with: $\mathrm{var}(e_o) = \sigma_o^2$

Solution:

$$\hat{x} = \frac{1}{\sigma_b^2 + \sigma_o^2}\left(\sigma_o^2 x_b + \sigma_b^2 y\right)$$

Equivalent to

$$\hat{x} = \underbrace{x_b}_{\text{background}} + \underbrace{\frac{\sigma_b^2}{\sigma_b^2 + \sigma_o^2}}_{\text{gain}}\underbrace{(y - x_b)}_{\text{innovation}}$$

background + gain x innovation

Error in estimate

$$var(\hat{x}) = \left(\frac{1}{\sigma_o^2} + \frac{1}{\sigma_b^2}\right)^{-1}$$

$$= \left(1 - \frac{\sigma_b^2}{\sigma_b^2 + \sigma_o^2}\right)\sigma_b^2$$

Equivalent to minimizing:

$$J(x) = \frac{1}{2}\frac{(x - x_b)^2}{\sigma_b^2} + \frac{1}{2}\frac{(x - y)^2}{\sigma_o^2}$$

Example: Dependence on observation error

Assume that

$$x_b = 1 \qquad y = 0$$

$$\sigma_b = 1$$

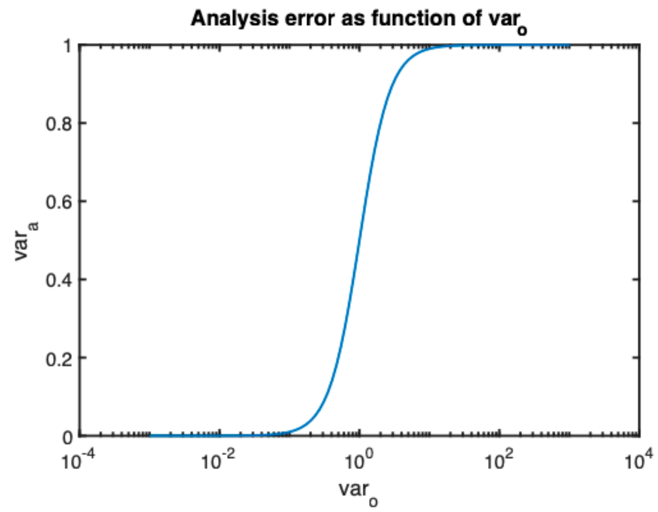
Now vary σ_o

Analysis error

Error reduced to

- 50% if $\text{var}_b = \text{var}_o$
- 91% if $\text{var}_o = 10\ \text{var}_b$
- 9% if $\text{var}_o = 0.1\ \text{var}_b$

Only depends on variances, not on state and observation



Varying σ_o has only an effect on DA result if within 1-2 orders of magnitude from σ_b

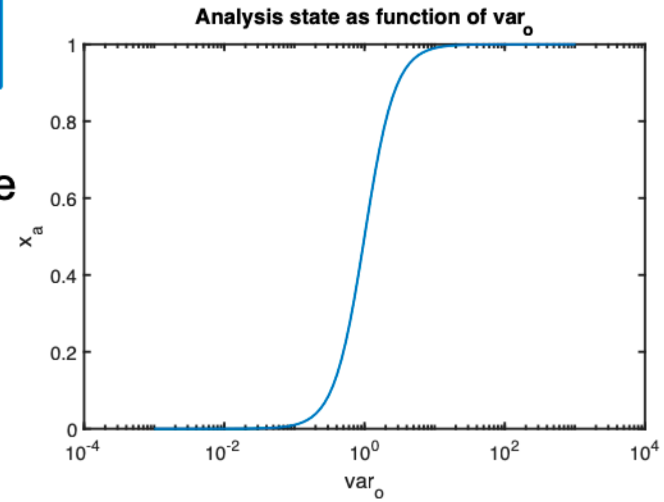
Notation in plots:

$$\text{var}_b := \sigma_b^2$$

$$\text{var}_o := \sigma_o^2$$

$$\text{var}_a := \text{var}(\hat{x})$$

Analysis state shows the same dependence as analysis error (note, that there are no errors on x_b and y)



Optimal Interpolation

Error Estimates

BLUE in Vector Form – Error of State Estimate

Analysis state

$$\mathbf{x}^a = \mathbf{x}^b + \underbrace{(\mathbf{B}^{-1} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})^{-1}\mathbf{H}^T\mathbf{R}^{-1}}_{\text{gain matrix}}\underbrace{(\mathbf{y} - \mathbf{H}\mathbf{x}^b)}_{\text{innovation vector}}$$

Equivalent (using Sherman-Morrison-Woodbury identity)

$$\mathbf{x}^a = \mathbf{x}^b + \mathbf{B}\mathbf{H}^T(\mathbf{H}\mathbf{B}\mathbf{H}^T + \mathbf{R})^{-1}(\mathbf{y} - \mathbf{H}\mathbf{x}^b)$$

Now define the *gain matrix* as

$$\mathbf{K} = \mathbf{B}\mathbf{H}^T\left(\mathbf{H}\mathbf{B}\mathbf{H}^T + \mathbf{R}\right)^{-1} = \mathbf{P}^a\mathbf{H}^T\mathbf{R}^{-1}$$

thus

$$\mathbf{x}^a = \mathbf{x}^b + \mathbf{K}(\mathbf{y} - \mathbf{H}\mathbf{x}^b) = \mathbf{x}^b + \mathbf{P}^a\mathbf{H}^T\mathbf{R}^{-1}(\mathbf{y} - \mathbf{H}\mathbf{x}^b)$$

Analysis error

$$\mathbf{P}^a = (\mathbf{I} - \mathbf{K}\mathbf{H})\,\mathbf{B} = (\mathbf{B}^{-1} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})^{-1}$$

This alternative form can be useful when we factorize **B = LL**^{T}

Error Estimate in OI

OI needs estimate of $\mathbf{B}$

- usually fixed over time
- but error is reduced in analysis: $\mathbf{P}^a = (\mathbf{I} - \mathbf{K}\mathbf{H})\,\mathbf{B}$

  → OI does not track development of error over time

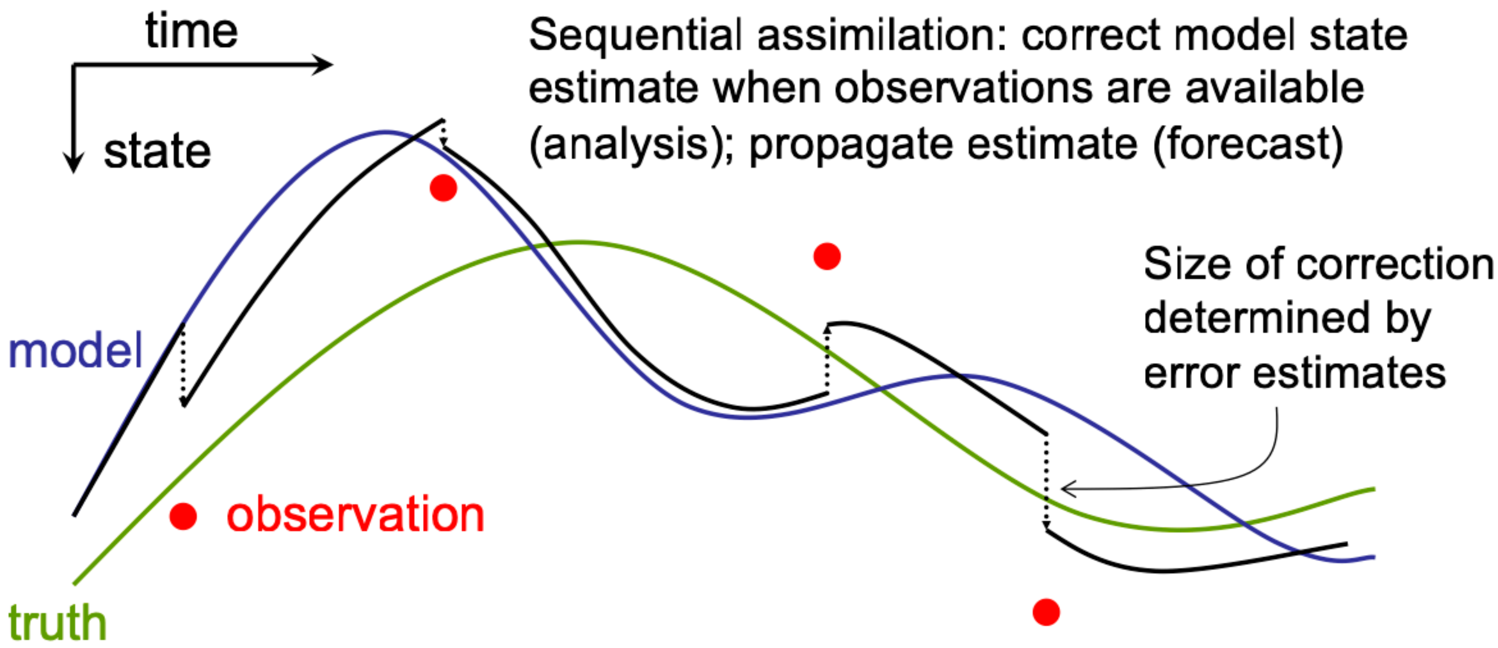
  → OI only estimates state, but not error over time

Sequential Data Assimilation

Kalman filters

Sequential Data Assimilation

Consider some physical system (ocean, atmosphere,…)



3D-Var is “sequential” but usually not called like it

Error propagation

Linear stochastic dynamical model

$$\mathbf{x}_i = \mathbf{M}_{i-1,i}\mathbf{x}_{i-1} + \boldsymbol{\eta}_i$$

Assume that $p(\mathbf{x}_{i-1}) = \mathcal{N}\left(\mathbf{x}_{i-1}, \mathbf{P}^a_{i-1}\right)$

Also assume uncorrelated state errors and model errors $\boldsymbol{\eta}_i$

Then

$$\mathbf{P}^f_i = \mathbf{M}_{i-1,i}\mathbf{P}^a_{i-1}(\mathbf{M}_{i-1,i})^T + \mathbf{Q}_{i-1}$$

With model error covariance matrix $\mathbf{Q}_{i-1}$

Error propagation builds the foundation of the Kalman filter

More later…

Propagation of covariance matrix

Forecast of error estimate – use definition of **P**:

$$\mathbf{P}^a_{i-1} = \mathrm{cov}(\mathbf{x}_i) = \frac{1}{N-1}\sum_{k=1}^{n} \mathbf{e}^k_{i-1}\ (\mathbf{e}^k_{i-1})^T$$

with errors

$$\mathbf{e}^k_{i-1} = (\mathbf{x}^k_{i-1} - \overline{\mathbf{x}}_{i-1})$$

➜ error propagation

$$\mathbf{e}^k_i = \mathbf{M}_{i-1,i}\mathbf{e}^k_{i-1} + \eta_{i-1}$$

Propagated covariance matrix ($\mathrm{cov}(\eta_{i-1}) = \mathbf{Q}_{i-1}$)

$$\mathbf{P}^f_i = \frac{1}{N-1}\sum_{k=1}^{n} \mathbf{e}^k_i(\mathbf{e}^k_i)^T = \frac{1}{N-1}\sum_{k=1}^{n} \mathbf{M}_{i-1,i}\mathbf{e}^k_{i-1}\ (\mathbf{e}^k_{i-1})^T\mathbf{M}^T_{i-1,i} + \mathbf{Q}_{i-1}$$

(We used $\mathrm{cov}(\mathbf{e}_{i-1}, \eta_{i-1}) = 0$)

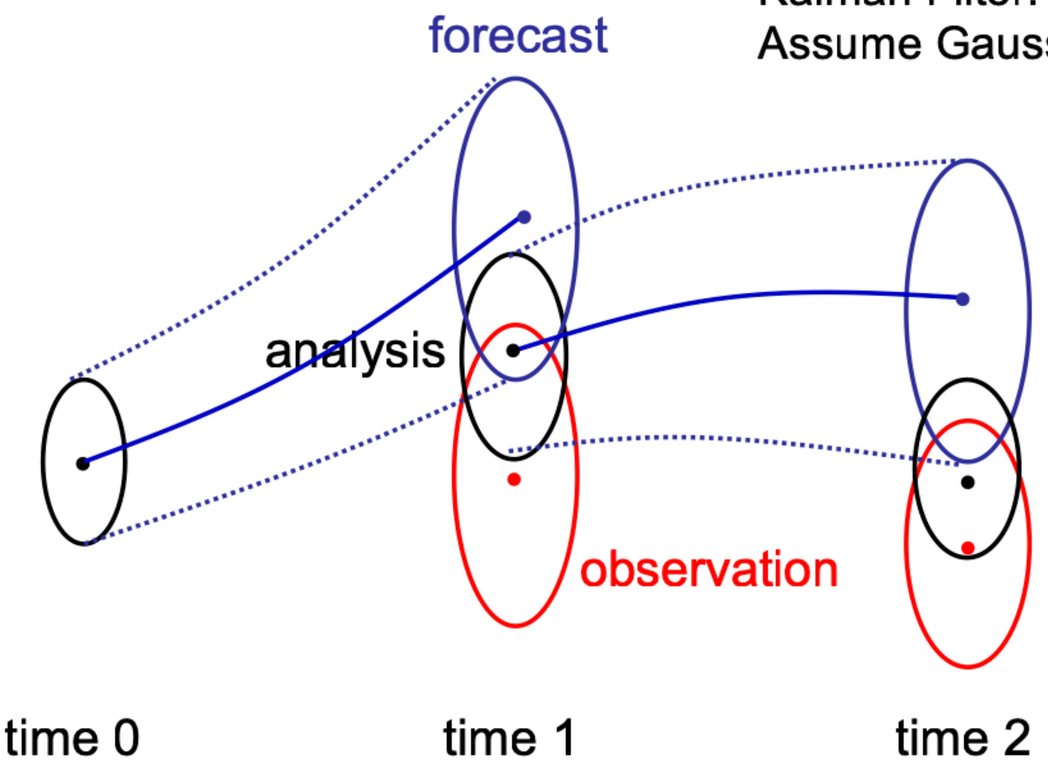
We don't have the factorization, so we directly evolve **P**:

$$\mathbf{P}^f_i = \mathbf{M}_{i-1,i}\mathbf{P}^a_{i-1}(\mathbf{M}_{i-1,i})^T + \mathbf{Q}_{i-1}$$

Probabilistic view: Optimal estimation

Consider probability distribution of model and observations

Kalman Filter:
Assume Gaussian distributions



The Kalman Filter

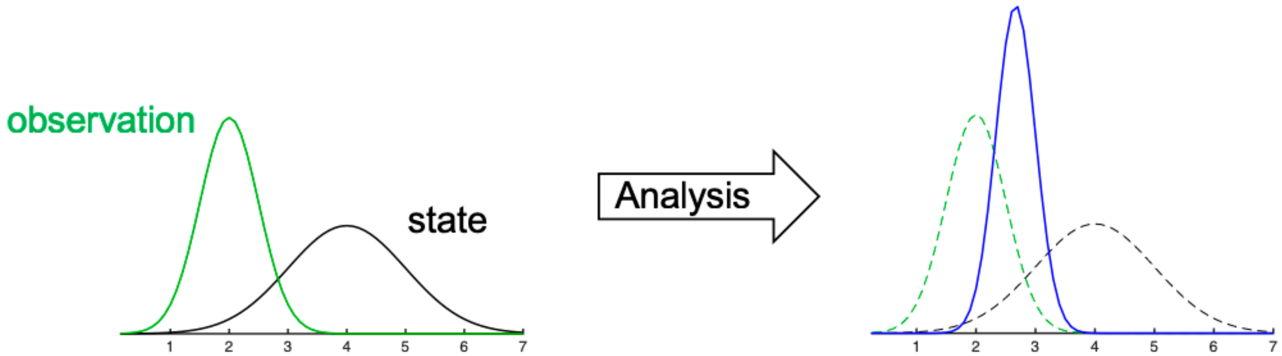
Assume Gaussian distributions
fully described by

- mean state estimate
- covariance matrix

➜ Strong simplification of estimation problem

Analysis is combination auf two Gaussian distributions
computed as

- Correction of state estimate
- Update of covariance matrix



Error assumptions

Errors: state $\mathbf{e}^x$

model η

observations $\mathbf{e}^o$

Assumptions

Errors are unbiased

$$E(\mathbf{e}^x) = E(\mathbf{e}^o) = E(\eta) = 0$$

Errors are Gaussian with known covariances

$$E\left(\mathbf{e}^o_k(\mathbf{e}^o_{k'})^T\right) = \mathbf{R}_k\delta_{kk'}$$

$$E\left(\mathbf{e}^x_0(\mathbf{e}^x_0)^T\right) = \mathbf{P}_0$$

$$E\left(\eta_k(\eta_{k'})^T\right) = \mathbf{Q}_k\delta_{kk'}$$

Some errors are independent

$$E\left(\mathbf{e}^x_0(\mathbf{e}^o_k)^T\right) = 0$$

$$E\left(\eta_k(\mathbf{e}^o_{k'})^T\right) = 0$$

Filter optimality depends on these assumptions

Kalman Filter (Kalman, 1960)

Forecast:

State propagation

$$\mathbf{x}_i = \mathbf{M}_{i-1,i}\mathbf{x}_{i-1} + \boldsymbol{\eta}_i$$

Propagation of error estimate

$$\mathbf{P}_i^f = \mathbf{M}_{i-1,i}\mathbf{P}_{i-1}^a(\mathbf{M}_{i-1,i})^T + \mathbf{Q}_{i-1}$$

Analysis at time t_k:

State update

$$\mathbf{x}_k^a = \mathbf{x}_k^f + \mathbf{K}_k\left(\mathbf{y}_k - \mathbf{H}_k\mathbf{x}_k^f\right)$$

Update of error estimate

$$\mathbf{P}_k^a = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\,\mathbf{P}_k^f$$

with “Kalman gain”

$$\mathbf{K}_k = \mathbf{P}_k^f\mathbf{H}_k^T\left(\mathbf{H}_k\mathbf{P}_k^f\mathbf{H}_k^T + \mathbf{R}_k\right)^{-1}$$

Derivation of Analysis Error

Start with analysis state

$$\mathbf{x}_k^a = \mathbf{x}_k^f + \mathbf{K}_k\left(\mathbf{y}_k - \mathbf{H}_k\mathbf{x}_k^f\right)$$
$$= (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\,\mathbf{x}_k^f + \mathbf{K}_k\mathbf{y}_k$$

We have the errors

$$\mathbf{x}_k^f = \mathbf{x}_k^t + \mathbf{e}_k^x \qquad \mathbf{y}_k = \mathbf{x}_k^t + \mathbf{e}_k^o$$

with

$$\mathrm{cov}(\mathbf{e}_k^x) = \mathbf{P}_k^f \qquad \mathrm{cov}(\mathbf{e}_k^x, \mathbf{e}_k^o) = 0 \qquad \mathrm{cov}(\mathbf{y}_k) = \mathbf{R}_k$$

Hence

$$\boxed{\mathbf{P}_k^a = \mathrm{cov}(\mathbf{x}_k^a) = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_k^f(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)^T + \mathbf{K}_k\mathbf{R}_k\mathbf{K}_k^T}$$

Using the Kalman gain $\mathbf{K}_k = \mathbf{P}_k^f\mathbf{H}_k^T\left(\mathbf{H}_k\mathbf{P}_k^f\mathbf{H}_k^T + \mathbf{R}_k\right)^{-1}$

We get $\boxed{\mathbf{P}_k^a = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_k^f}$

Important: This **only** holds for the optimal Kalman gain!

Extended Kalman Filter (E.g. Jazwinsky 1970)

Forecast:

Nonlinear state propagation

$$\mathbf{x}_i = M_{i-1,i}(\mathbf{x}_{i-1}) + \epsilon_i$$

Propagation of error estimate

$$\mathbf{P}_i^f = \mathbf{M}_{i-1,i}\mathbf{P}_{i-1}^a(\mathbf{M}_{i-1,i})^T + \mathbf{Q}_{i-1}$$

Analysis at time t_k:

State update

$$\mathbf{x}_k^a = \mathbf{x}_k^f + \mathbf{K}_k\left(\mathbf{y}_k - \mathbf{H}_k\mathbf{x}_k^f\right)$$

Update of error estimate

$$\mathbf{P}_k^a = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\,\mathbf{P}_k^f$$

with “Kalman gain”

$$\mathbf{K}_k = \mathbf{P}_k^f\mathbf{H}_k^T\left(\mathbf{H}_k\mathbf{P}_k^f\mathbf{H}_k^T + \mathbf{R}_k\right)^{-1}$$

The KF (Kalman, 1960)

With nonlinear model: Extended Kalman filter

Initialization: Choose initial state estimate **x** and corresponding covariance matrix **P**

Forecast: Evolve state estimate with non-linear model. Evolve columns/rows of covariance matrix with linearized model.

Analysis: Combine state estimate with observations based on weights computed from error estimates of state estimate and observations. Update matrix **P** according to relative error estimates.

Issues of the Kalman Filter

- Storage of covariance matrix unfeasible for high-dimensional models (n^2 with n of $\mathcal{O}(10^6\text{-}10^9)$)
- Evolving covariance matrix is extremely costly
- Extended Kalman filter:
  - error propagation only valid to first order
  - state propagation valid to higher order

    ➜ can lead to biased estimates
- Linearized evolution (like in Extended KF) can be unstable (e.g. discussed by Evensen 1992, 1993)

➜ Need to

1. reduce the cost
2. improve propagation for nonlinear systems

Low-rank Kalman Filters

"Suboptimal" Filters – development of the 1990ies

Approaches to reduce the cost of the Kalman filter

- Simplified error evolution
  (constant, variance only)
- Represent $\mathbf{P}$ by low-rank matrix
- Reduce resolution of model
  (at least for the error propagation)
- Reduce model complexity

Examples:

- „suboptimal schemes", Todling & Cohn 1994
- Approximate KF, Fukumori & Malanotte, 1995
- RRSQRT, Verlaan & Heemink, 1995/97
- SEEK, Pham et al., 1998

Low-rank approximation of P

Example: **SEEK filter** (Pham et al., 1998)

Approximate $\mathbf{P}_i^a \approx \mathbf{V}_i \mathbf{U}_i \mathbf{V}_i^T$

(truncated eigendecomposition)

Mode matrix $\mathbf{V}_i$ has size $n \times r$ $\mathbf{U}_i$ has size $r \times r$

Forecast of r „modes":

$$\mathbf{V}_{i+1} = \mathbf{M}_{i,i+1} \mathbf{V}_i$$

for nonlinear model

$$\mathbf{V}_{i+1} \approx M_{i,i+1} \left(\mathbf{V}_i + [\mathbf{x}_i^a, \ldots, \mathbf{x}_i^a] \right) - M_{i,i+1} [\mathbf{x}_i^a, \ldots, \mathbf{x}_i^a]$$

Now use in analysis step:

$$\tilde{\mathbf{P}}_k^f \approx \mathbf{V}_k \mathbf{U}_{k-1} \mathbf{V}_k^T$$

The SEEK filter (Pham et al., 1998)

Initialization: Approximate covariance matrix by low-rank matrix in the form $\mathbf{P}=\mathbf{V}\mathbf{U}\mathbf{V}^T$. Choose state $\mathbf{x}$.

Forecast: Evolve state estimate with non-linear model. Evolve modes **V** of covariance matrix with linearized model.

Analysis: Apply EKF update step to ensemble mean and the „eigenvalue matrix“ **U**. Covariance matrix represented by modes and **U**.

Re-Initialization: Occasionally perform re-orthogonalization of modes of covariance matrix

SEEK's Representation of Covariance Matrix

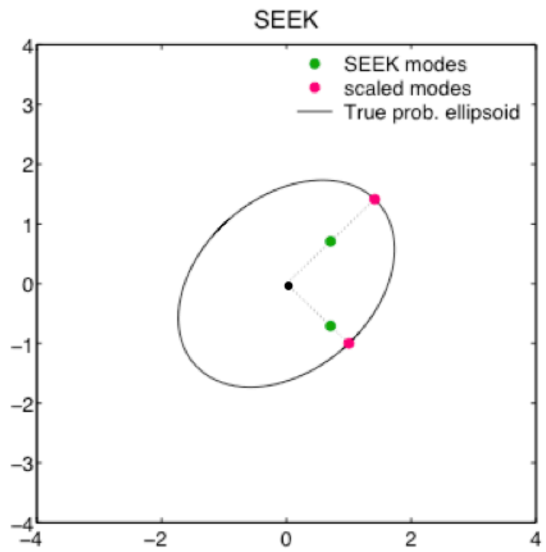
Example matrix and state

$$\mathbf{P}_t = \begin{pmatrix} 3.0 & 1.0 & 0.0 \\ 1.0 & 3.0 & 0.0 \\ 0.0 & 0.0 & 0.01 \end{pmatrix}; \; \mathbf{x}_t = \begin{pmatrix} 0.0 \\ 0.0 \end{pmatrix}$$

Eigenvalues:

4, 2, and 0.01

→ Approximate by matrix of rank 2 dropping the direction of smallest eigenvalue



Using eigenvectors and eigenvalues directly is particular way to sample the covariance matrix

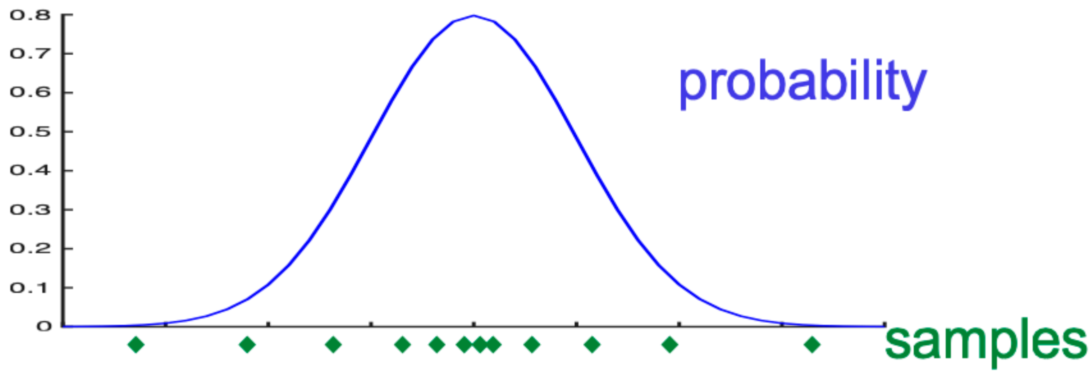
General sampling of probability distribution

Kalman filter assumes Gaussian distribution
using mean and covariance matrix

General representation:

- Sample $p(\mathbf{x})$ by N random state realizations $\mathbf{x}^{(j)}$:

$$p(\mathbf{x}) = \frac{1}{N}\sum_{j=1}^{N}\delta(\mathbf{x}-\mathbf{x}^{(j)})$$



General sampling of probability distribution

Kalman filter assumes Gaussian distribution
using mean and covariance matrix

General representation:

- Sample $p(\mathbf{x})$ by N random state realizations $\mathbf{x}^{(j)}$:

$$p(\mathbf{x}) = \frac{1}{N}\sum_{j=1}^{N}\delta(\mathbf{x}-\mathbf{x}^{(j)})$$

- State **ensemble**

$$\mathbf{X} = \left[\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}\right]$$

- Ensemble mean $\bar{\mathbf{x}} = \frac{1}{N}\sum_{j=1}^{N}\mathbf{x}^{(j)}$

Ensemble representation (approximation) of P

Approximate

$$\mathbf{P}_i^a \approx \frac{1}{N-1}\left(\mathbf{X}_i - \bar{\mathbf{X}}_i\right)\left(\mathbf{X}_i - \bar{\mathbf{X}}_i\right)^T$$

($\bar{\mathbf{X}}_i$ holds ensemble mean in each column)

Forecast of N ensemble states:

$$\mathbf{X}_{i+1}^f = \mathbf{M}_{i,i+1}\mathbf{X}_{i+1}^a$$

for nonlinear model

$$\mathbf{X}_{i+1}^f = M_{i,i+1}\mathbf{X}_{i+1}^a$$

Now use in analysis step:

$$\hat{\mathbf{P}}_i^f \approx \frac{1}{N-1}\left(\mathbf{X}_i^f - \bar{\mathbf{X}}_i^f\right)\left(\mathbf{X}_i^f - \bar{\mathbf{X}}_i^f\right)^T$$

Sampling Example – Monte Carlo sampling

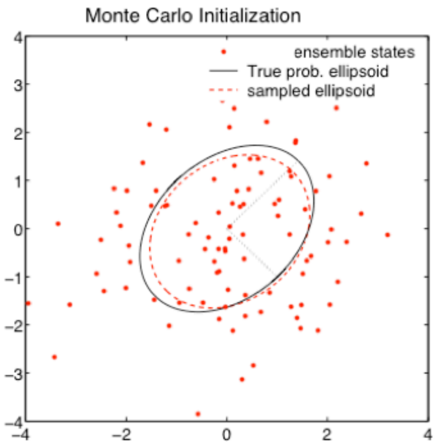
Example matrix and state

$$\mathbf{P}_t = \begin{pmatrix} 3.0 & 1.0 & 0.0 \\ 1.0 & 3.0 & 0.0 \\ 0.0 & 0.0 & 0.01 \end{pmatrix}; \quad \mathbf{x}_t = \begin{pmatrix} 0.0 \\ 0.0 \end{pmatrix}$$

Monte Carlo sampling of $\mathbf{P}$

- 100 random samples transformed by square root of $\mathbf{P}$
- Estimated covariance matrix has sampling errors

- Center of ellipse represents state estimate $\boldsymbol{x}$



More on sampling

- Ensemble is not unique
- Gaussian assumption simplifies sampling (covariance matrix & mean state)

Example: 2nd-order exact sampling (Pham et al. 1998)

Use $\mathbf{P}_i^a \approx \mathbf{V}_i \mathbf{S}_i \mathbf{V}_i^T$

(truncated eigendecomposition)

Create ensemble states as

$$\mathbf{X} = \bar{\mathbf{X}} + \sqrt{N-1}\mathbf{V}\mathbf{S}^{1/2}\boldsymbol{\Omega}^T$$

$\boldsymbol{\Omega}$ is random matrix with columns orthonormal and orthogonal to vector $(1, \ldots, 1)^T$. Size $N \times (N-1)$

Ensemble size $N = r + 1$

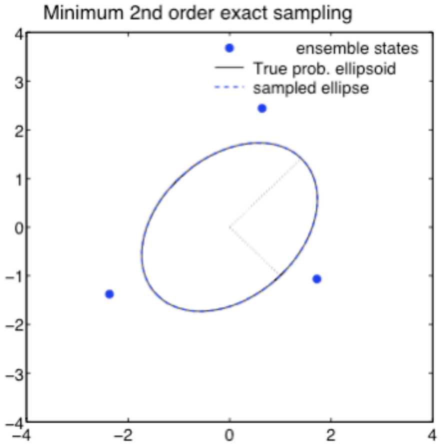
Sampling Example – 2nd-order exact

Example matrix and state

$$\mathbf{P}_t = \begin{pmatrix} 3.0 & 1.0 & 0.0 \\ 1.0 & 3.0 & 0.0 \\ 0.0 & 0.0 & 0.01 \end{pmatrix}; \quad \mathbf{x}_t = \begin{pmatrix} 0.0 \\ 0.0 \end{pmatrix}$$

2nd order exact sampling

- rank 2 matrix is exactly sampled using 3 state realizations



Same as spherical simplex sampling (Wang et al., 2004)

Error Subspace Algorithms

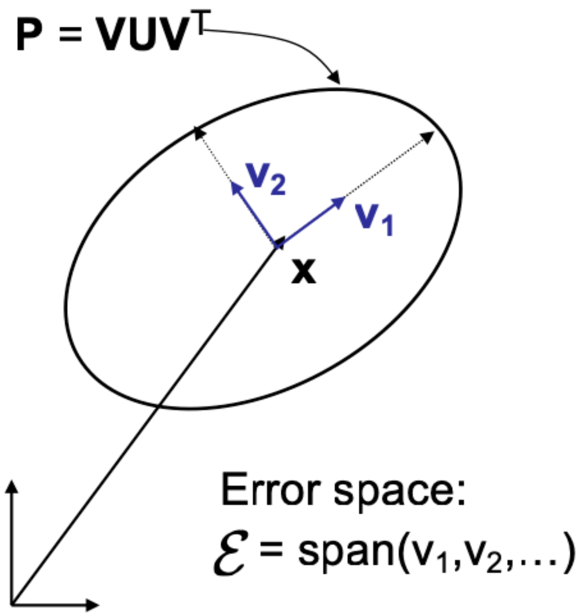
⇨ Approximate state covariance matrix by low-rank matrix

⇨ Keep matrix in decomposed form ($\mathbf{XX}^T$, $\mathbf{VUV}^T$)

Mathematical motivation:

- state error covariance matrix represents error space at location of state estimate
- directions of different uncertainty
- consider only directions with largest errors (error subspace)

⇒ degrees of freedom for state correction in analysis: rank($\mathbf{P}$)



Lars Nerger et al., Tellus 57A (2005) 715-735

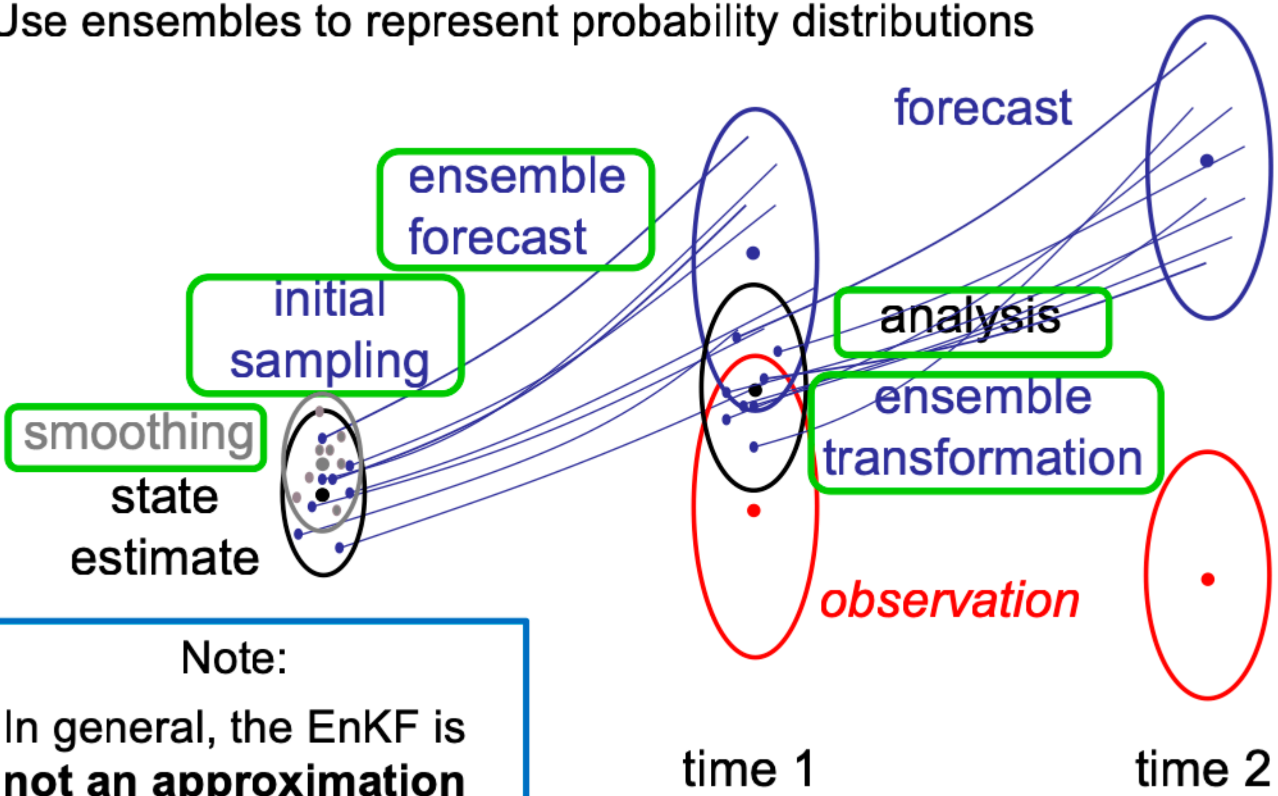
Ensemble-based Kalman filters

Ensemble Kalman Filters

First formulated by G. Evensen (EnKF, J. Geophys. Res. 1994)

Kalman filter: express probability distributions by mean and covariance matrix

EnKF: Use ensembles to represent probability distributions



Note:

In general, the EnKF is **not an approximation** of the Kalman filter!

Much research into how to perform these operations

Most can be implemented in generic form

Available in our DA software PDAF

Use sampled matrix P in analysis

$\mathbf{P}_k^f$ can be approximated by ensemble: $\tilde{\mathbf{P}}_k^f$

$$\tilde{\mathbf{P}}_k^f = \frac{1}{N-1}\left(\mathbf{X}_k^f - \overline{\mathbf{X}}_k^f\right)\left(\mathbf{X}_k^f - \overline{\mathbf{X}}_k^f\right)^T$$

Analysis at time t_k:

$$\mathbf{x}_k^a = \mathbf{x}_k^f + \tilde{\mathbf{K}}_k\left(\mathbf{y}_k - \mathbf{H}_k\mathbf{x}_k^f\right)$$

Kalman gain

$$\tilde{\mathbf{K}}_k = \tilde{\mathbf{P}}_k^f\mathbf{H}_k^T\left(\mathbf{H}_k\tilde{\mathbf{P}}_k^f\mathbf{H}_k^T + \mathbf{R}_k\right)^{-1}$$

This only provides the analysis state estimate

➜ but not the analysis ensemble

The EnKF - Monte Carlo analysis update

Analysis step of **Ensemble Kalman Filter** (EnKF, Evensen 1994)

- Generate observation ensemble

$$\mathbf{y}^{(j)} = \mathbf{y} + \epsilon^{(j)} \quad \text{with} \quad [\epsilon^{(1)}, \ldots, \epsilon^{(N)}][\epsilon^{(1)}, \ldots, \epsilon^{(N)}]^T \approx \mathbf{R}$$

- Update each ensemble member

$$\mathbf{x}_k^{a(i)} = \mathbf{x}_k^{f(i)} + \tilde{\mathbf{K}}_k \left(\mathbf{y}_k^{(i)} - \mathbf{H}_k \mathbf{x}_k^{f(i)} \right)$$

Advantage:

- Simple implementation – combined analysis and resampling

Issues:

- Generation of observation ensemble
- Introduction of sampling noise through $\epsilon^{(j)}$
- Costly inversion of $m \times m$ matrix (~m^3 operations)

EnKF update – practical computing steps

1. Generate observation ensemble

$$\mathbf{y}^{(j)} = \mathbf{y} + \epsilon^{(j)} \quad \text{with} \quad [\epsilon^{(1)}, \ldots, \epsilon^{(N)}][\epsilon^{(1)}, \ldots, \epsilon^{(N)}]^T \approx \mathbf{R}$$

2. Update each ensemble member

$$\mathbf{x}_k^{a(i)} = \mathbf{x}_k^{f(i)} + \tilde{\mathbf{K}}_k \left(\mathbf{y}_k^{(i)} - \mathbf{H}_k \mathbf{x}_k^{f(i)}\right)$$

$$\tilde{\mathbf{K}}_k = \tilde{\mathbf{P}}_k^f \mathbf{H}_k^T \left(\mathbf{H}_k \tilde{\mathbf{P}}_k^f \mathbf{H}_k^T + \mathbf{R}_k\right)^{-1}$$

Note: Applying H directly to a state vector allows a nonlinear operation

By computing $\mathbf{H}_k \mathbf{x}_k^{f(i)}$ for all ensemble members i yielding matrix $\mathbf{H}_k \mathbf{X}_k^f$ and the mean of the observed ensemble, in matrix form $\overline{\mathbf{H}_k \mathbf{X}_k^f}$

Then compute

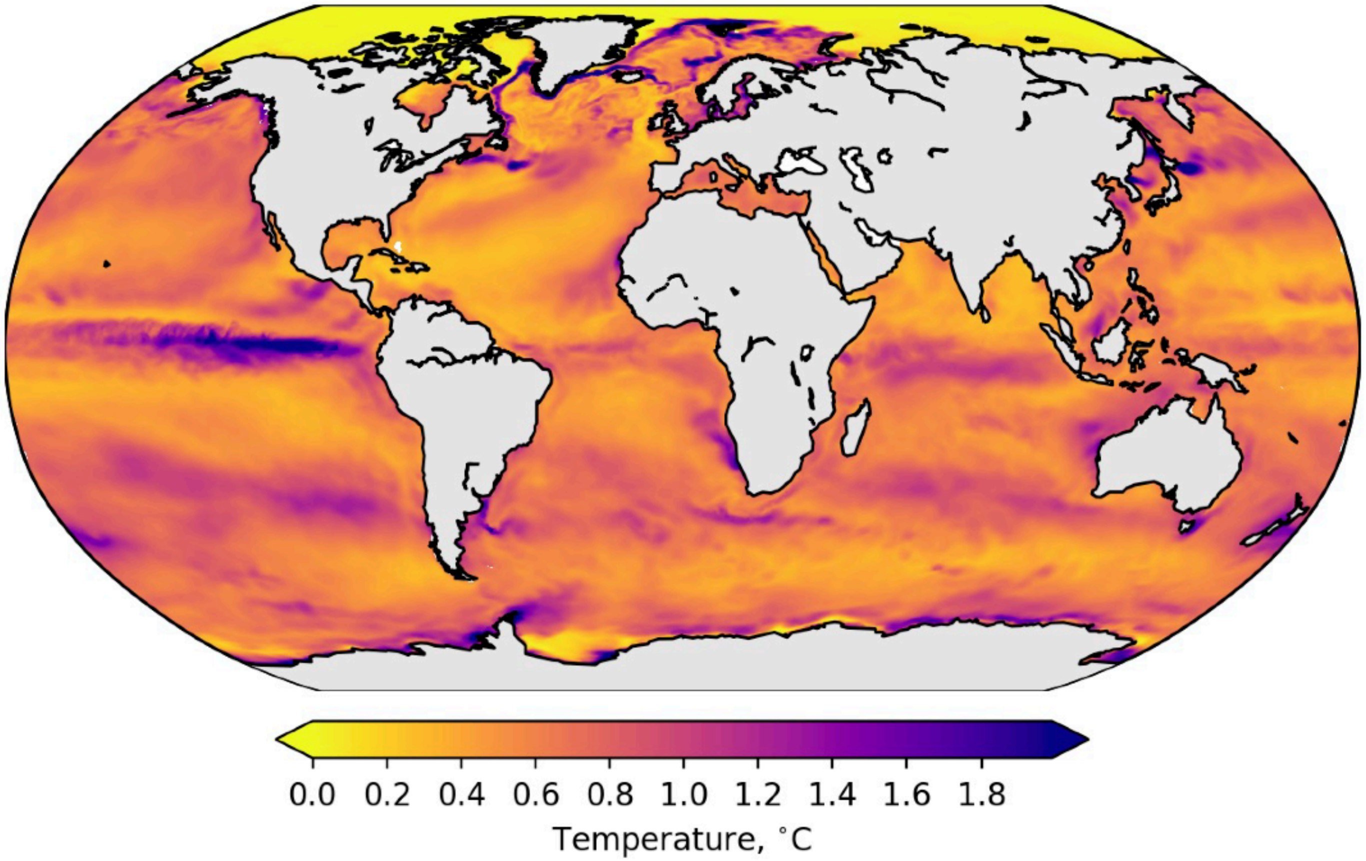
$$\mathbf{H}_k \tilde{\mathbf{P}}_k^f \mathbf{H}_k^T = \left(\mathbf{H}_k \mathbf{X}_k^f - \overline{\mathbf{H}_k \mathbf{X}_k^f}\right)\left(\mathbf{H}_k \mathbf{X}_k^f - \overline{\mathbf{H}_k \mathbf{X}_k^f}\right)^T$$

$$\tilde{\mathbf{P}}_k^f \mathbf{H}_k^T = \left(\mathbf{X}_k^f - \overline{\mathbf{X}_k^f}\right)\left(\mathbf{H}_k \mathbf{X}_k^f - \overline{\mathbf{H}_k \mathbf{X}_k^f}\right)^T$$

Efficient Computing is essential because we work with very large matrices! Avoid large temporary matrices!

Ensemble: Model Error Estimate – free run

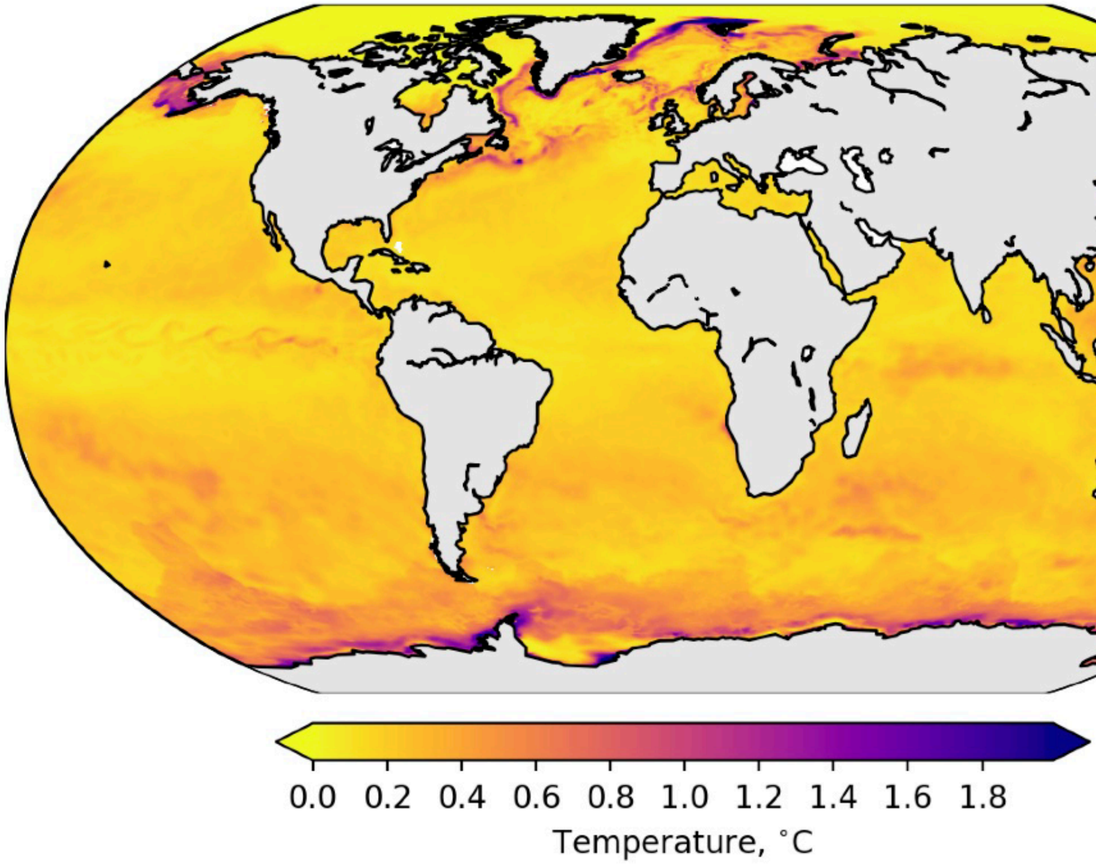
Ensemble-estimate of SST standard deviation on December 31



- Provide uncertainty information
  - Diagonal: variances
  - Off-diagonal: covariances
- Generated dynamically by propagating ensemble of model states

Ensemble: Model Error Estimate – with assimilation

Ensemble-estimate of SST standard deviation on December 31



- With assimilation: Reduced uncertainty
  - ➜ Error estimate accounts for incorporated observational information

Square-root Kalman filters

(the current efficient work horses)

Ensemble-based/error-subspace Kalman filters

A little "zoo" (not complete):

Which filter should one use?

The original 'Ensemble Kalman Filter of 1994:
- EnKF(94/98)

'suboptimal schemes' 1990s:
- RRSQRT
- ROEK
- SEEK
- SEIK

advanced EnKFs 2000/2010s:
- EnKF(2003)
- EnKF(2004)
- EAKF
- EnSRF
- DEnKF
- ETKF
- ESTKF

other KF based methods:
- MLEF
- SPKF
- ESSE
- RHF

- Different researchers developed 'their' filter
- Properties and differences were not well understood (no full mathematical analysis)
- We learned from studying relations and differences (all is very suboptimal)

Ensemble-based/error-subspace Kalman filters

A little "zoo" (not complete):

Which filter should one use?

EnKF(94/98)

Today's commonly used filters

(what is used often depends on the software you use)

RRSQRT

ROEK

SEEK

SEIK

'suboptimal schemes'
1990s

EnKF(2003)

EnKF(2004)

EAKF

EnSRF

DEnKF

ETKF

ESTKF

advanced EnKFs
2000/2010s

MLEF

SPKF

ESSE

RHF

other KF based methods

Ensemble transformations

$\mathbf{P}_k^f$ can be approximated by ensemble or modes: $\tilde{\mathbf{P}}_k^f$

Analysis at time t_k:

Update of state (ensemble mean)

$$\mathbf{x}_k^a = \mathbf{x}_k^f + \tilde{\mathbf{K}}_k \left(\mathbf{y}_k - \mathbf{H}_k \mathbf{x}_k^f \right)$$

Update of error estimate

$$\tilde{\mathbf{P}}_k^a = \left(\mathbf{I} - \tilde{\mathbf{K}}_k \mathbf{H}_k \right) \tilde{\mathbf{P}}_k^f$$

This is incomplete!

We are missing the analysis ensemble $\mathbf{X}_k^a$

Square-roots

Forecast ensemble covariance matrix:

$$\mathbf{P}_k^f = \frac{1}{N-1}\mathbf{X}_k^{f\prime}(\mathbf{X}_k^{f\prime})^T$$

Analysis ensemble covariance matrix:

$$\mathbf{P}_k^a = \frac{1}{N-1}\mathbf{X}_k^{f\prime}\mathbf{A}(\mathbf{X}_k^{f\prime})^T$$

with a symmetric matrix $\mathbf{A}$

Analysis ensemble perturbations then given by

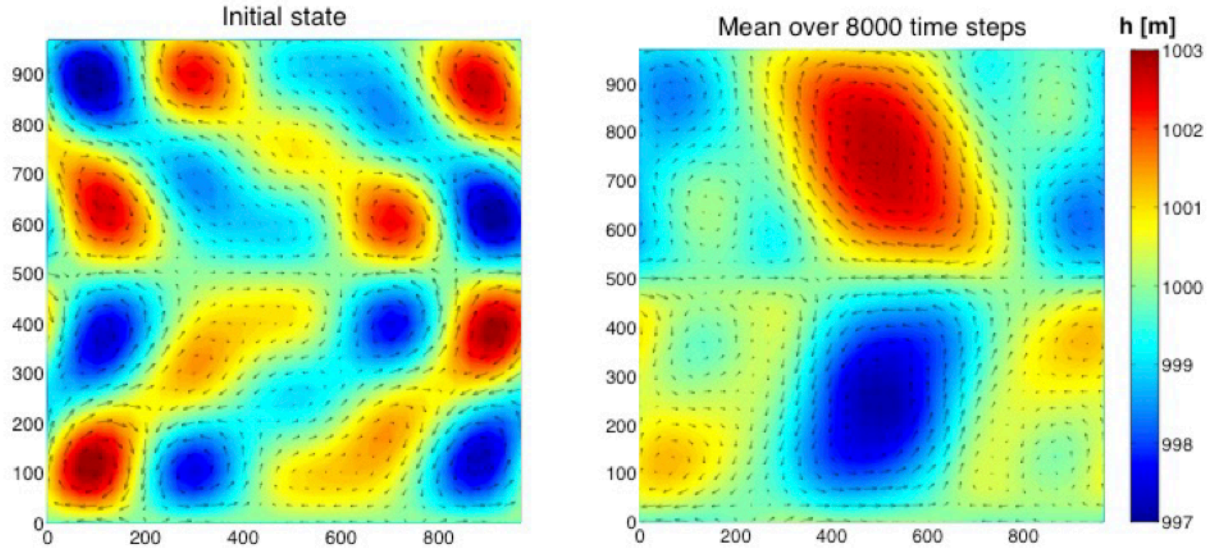
$$\mathbf{X}_k^{a\prime} = \mathbf{X}_k^{f\prime}\mathbf{A}^{1/2}$$

What is $\mathbf{A}$?

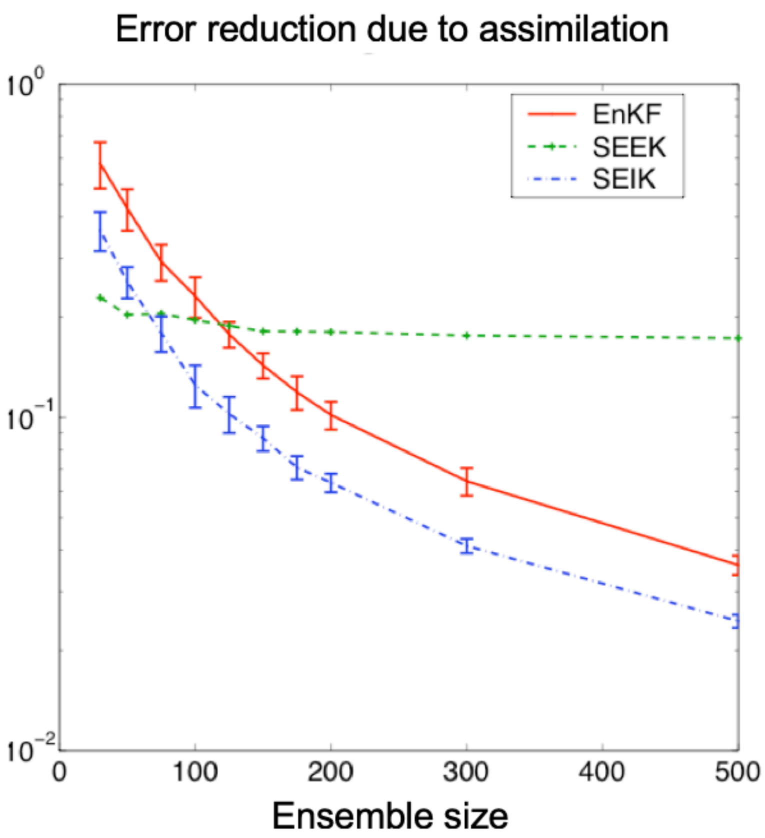
$\mathbf{A}$ is not unique!
Different filters use different definitions or distinct ways to compute it

A simple test problem

- Twin experiment with nonlinear shallow water equations
- Initial state estimate: temporal mean state
- Initial cov. matrix: variability around mean state
- Compare EnKF, SEEK, and ensemble square-root filter SEIK



Shallow water model: filter performances



- SEEK uses linearized forecast: here it stagnates
- same convergence behavior for EnKF and SEIK
- higher errors for EnKF than for SEIK
- EnKF ensemble 1.5-2 times larger than SEIK ensemble for same filter performance (caused by sampling errors)
- EnKF analysis step is also more costly to compute

Hands-On Tutorial 2

Kalman and Ensemble Kalman Filters

Hands-on Tutorial 2: Kalman and Ensemble Filters

Use the online tutorial in the browser

http://pdaf.awi.de/DA_demo/

1 (Extended) Kalman Filter vs. Optimal Interpolation

a) Use the model “Identity matrix” with default settings.
Compare the behaviour of Optimal Interpolation and Kalman Filter for both variables x1 and x2. How do they perform differently?

b) Use the model “Oscillation” with default settings.
How do Optimal Interpolation and Kalman Filter behave differently for the cases “Observed every x grid points” =2 and =1? (Check both state vector values and error covariance)

Hands-on 2: (Extended) Kalman Filter

2 (Extended) Kalman Filter

- Select ‘Method’ = ‘(Extended) Kalman Filter’

2.1 Use the model “1D advection in periodic domain” with default settings.

a) Using the default values, except reducing the model time step between observations from 5 to 4. Now we assimilate more frequently, yet no variable converges anymore to the truth. Why?

Hands-on 2: Ensemble Kalman Filter

3 Ensemble Kalman Filter vs. (Extended) Kalman Filter

3.1 Use the linear model "oscillation" – set 'Number of time steps' =200

a) Compare the results using the Extended Kalman Filter and the Ensemble Kalman Filters for the cases "Model time steps between observations" 5 and 4.

3.2 Use the model "Lorenz (1963)"

b) Set "Number of time steps to 80". Compare the results using the Extended Kalman Filter and the Ensemble Kalman Filters. How can be the results be explained taking into account the different forecast variants of the Extended KF and Ensemble KF?