Using Ensemble Kalman Filters to Assimilate Dynamic Ocean Topography Data into a Global Ocean Model

Lars Nerger

Alfred Wegener Institute for Polar and Marine Research
Bremerhaven, Germany

and
Bremen Supercomputing Competence Center BremHLR
Bremen, Germany

Lars.Nerger@awi.de

BremHLR
Kompetenzzentrum für Höchstleistungsrechnen Bremen

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

IGG, Uni. Bonn, 20.6.2013

Outline

- Ensemble-based Kalman filters
- Implementation aspects
- Application with global ocean model

Collaborations:

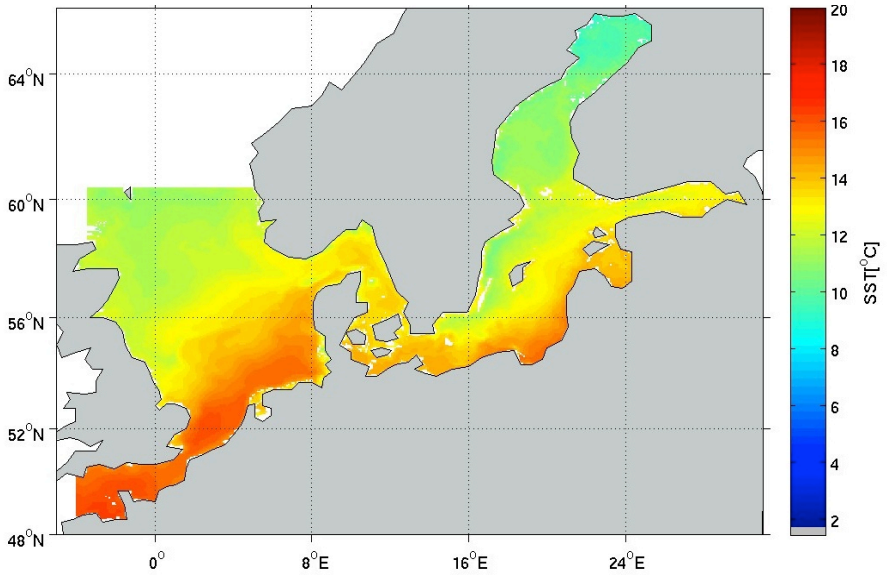
AWI: W. Hiller, J. Schröter, A. Alexandrov, P. Kirchgessner, S. Loza, T. Janjic (now DWD)

BSH: F. Janssen, S. Massmann

O.A.Sys GmbH: Reiner Schnur

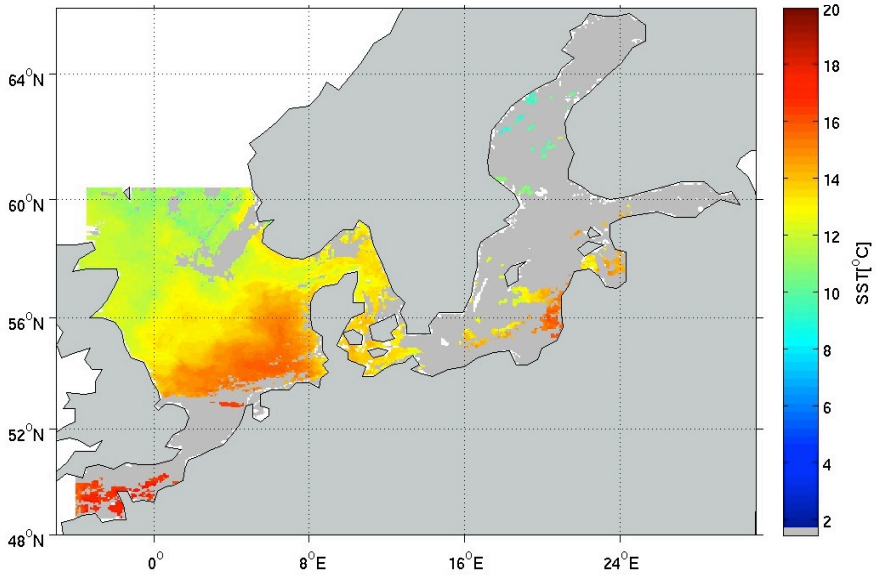
Application Example

Model surface temperature



Information: Model

Satellite surface temperature



Information: Observation

Combine both sources of information

quantitatively by computer algorithm

➜ data assimilation

Data Assimilation

- Optimal estimation of system state:
  - initial conditions (for weather/ocean forecasts, …)
  - state trajectory (temperature, concentrations, …)
  - parameters (growth of phytoplankton, …)
  - fluxes (heat, primary production, …)
  - boundary conditions and 'forcing' (wind stress, …)

- Characteristics of system:
  - high-dimensional numerical model - $\mathcal{O}(10^6\text{-}10^9)$
  - sparse observations
  - non-linear

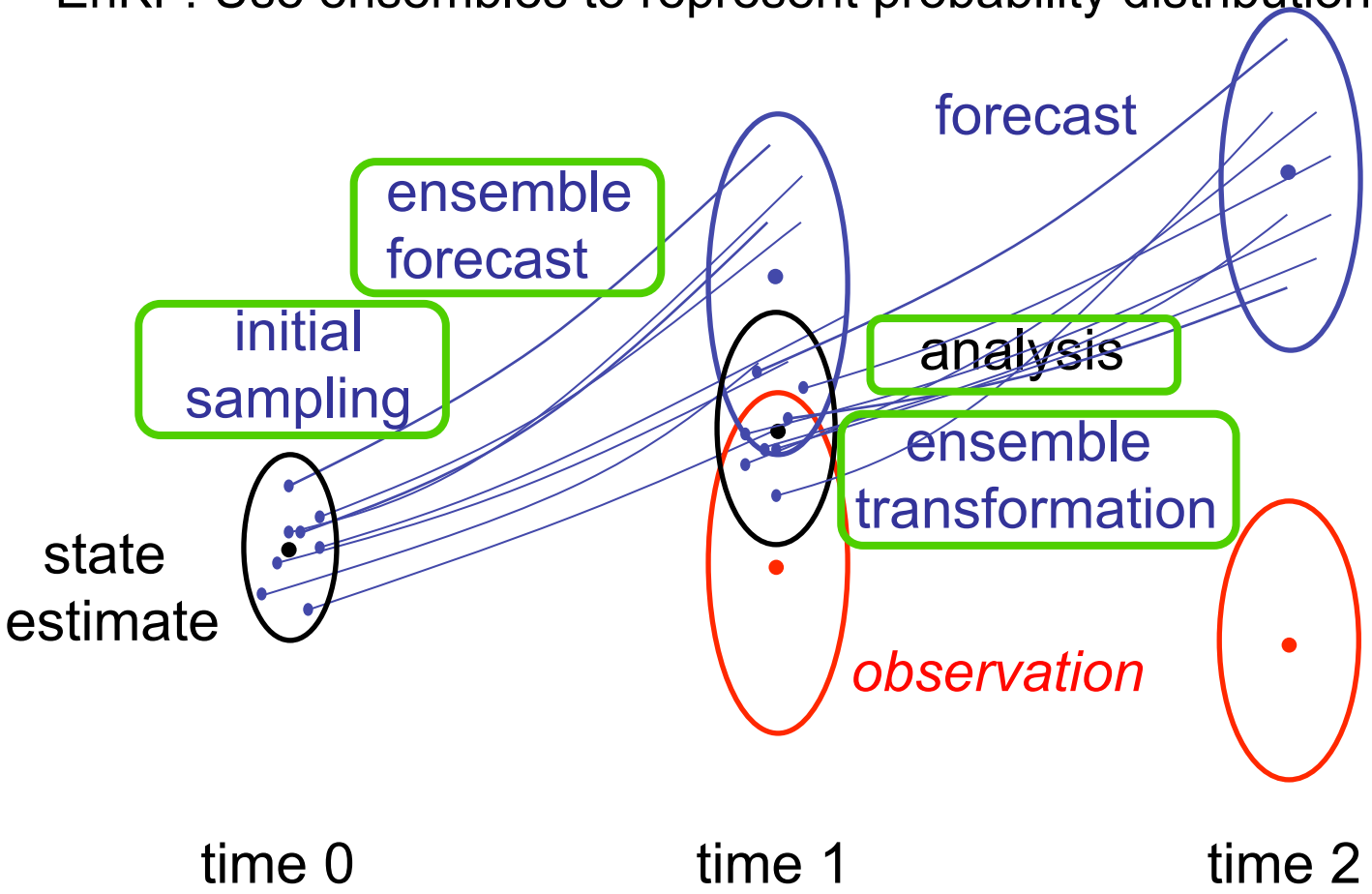
Ensemble-based Kalman Filters

Ensemble-based Kalman Filter

First formulated by G. Evensen (EnKF, 1994)

Kalman filter: express probability distributions by mean and covariance matrix

EnKF: Use ensembles to represent probability distributions



Looks trivial!

BUT:

There are many possible choices!

Which filter should one use?

Many choices - a little “zoo” (not complete):

EnKF(94/98)

Studied in Nerger et al. Tellus (2005)

RRSQRT

ROEK

SEEK

SEIK

EnKF(2003)

EnKF(2004)

EAKF

EnSRF

ETKF

MLEF

SPKF

ESSE

RHF

- Properties and differences are not fully understood

Data Assimilation – Model and Observations

Two components:

1. **State**: $\mathbf{x} \in \mathbb{R}^n$

   Dynamical model

   $$\mathbf{x}_i = M_{i-1,i}\left[\mathbf{x}_{i-1}\right]$$

2. **Obervations**: $\mathbf{y} \in \mathbb{R}^m$

   Observation equation (relation of observation to state $\mathbf{x}$):

   $$\mathbf{y} = H\left[\mathbf{x}\right]$$

   Observation error covariance matrix: $\mathbf{R}$

The Ensemble Kalman Filter (EnKF, Evensen 94)

Ensemble $\{\mathbf{x}_0^{a(l)}, l = 1, \ldots, N\}$

Analysis step:

Update each ensemble member

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

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

Kalman filter

Ensemble covariance matrix

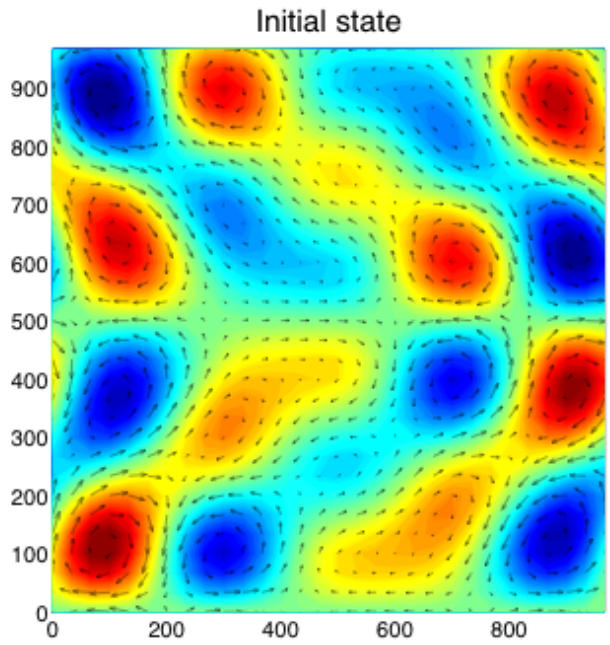
$$\mathbf{P}_k^f := \frac{1}{N-1}\sum_{l=1}^{N}\left(\mathbf{x}_k^{f(l)} - \overline{\mathbf{x}_k^f}\right)\left(\mathbf{x}_k^{f(l)} - \overline{\mathbf{x}_k^f}\right)^T$$

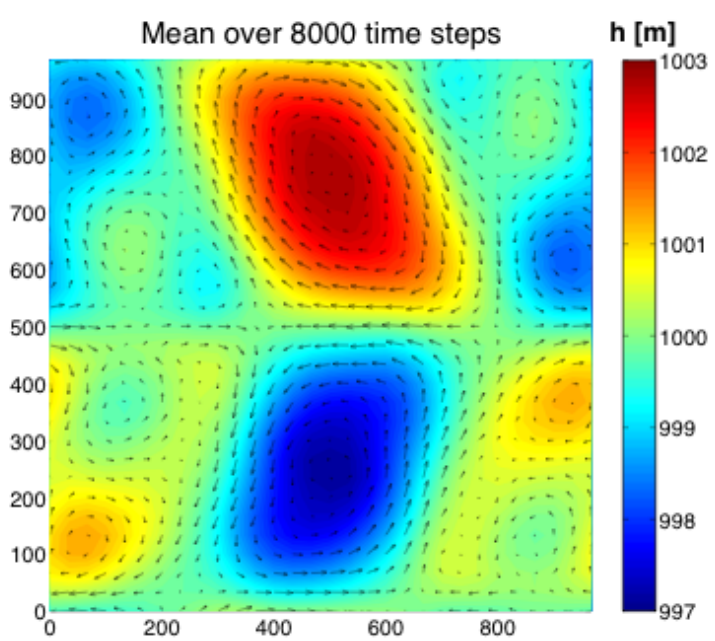
Ensemble mean (state estimate)

$$\mathbf{x}_k^a := \frac{1}{N}\sum_{l=1}^{N}\mathbf{x}_k^{a(l)}$$

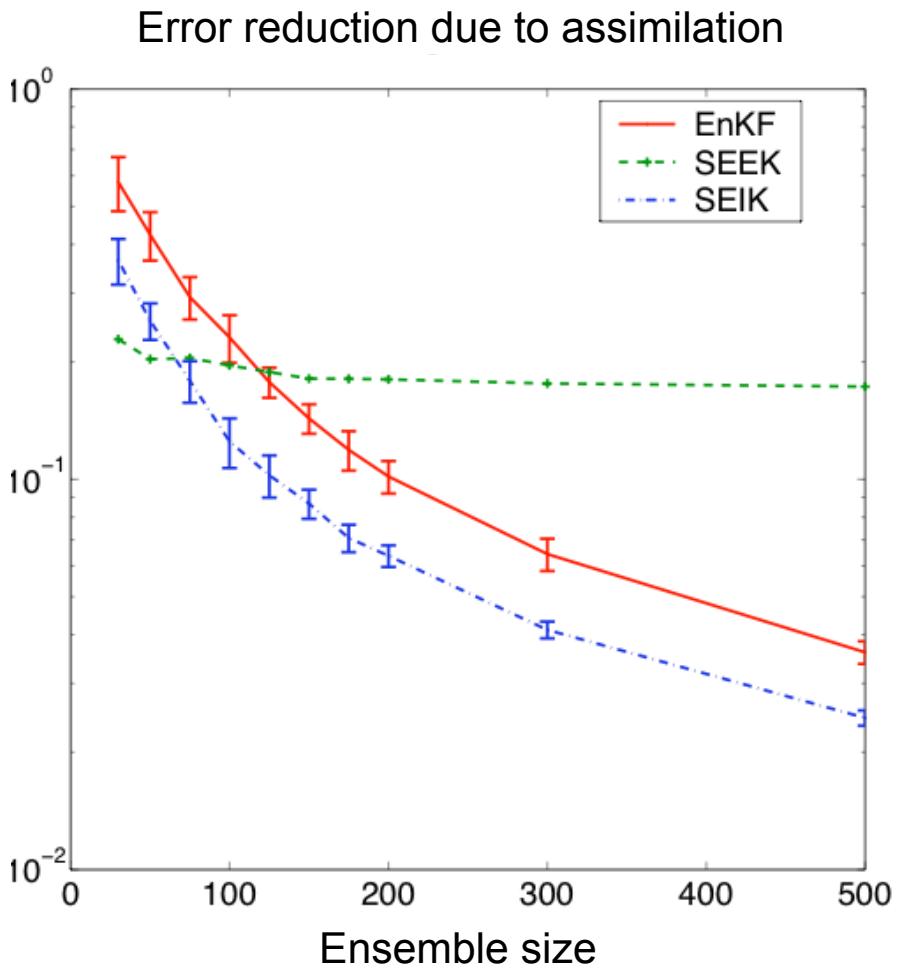
A simple test problem

- Twin experiment with nonlinear shallow water model
- Initial state estimate: temporal mean state
- Initial covariance matrix: variability around mean state





Shallow water model: filter performances



- SEEK stagnates
- same convergence behavior for EnKF and SEIK
- smaller performance for EnKF than for SEIK
- EnKF ensemble 1.5-2 times larger than SEIK ensemble for same filtering performance

Nerger, Hiller, Schröter. Tellus 57A (2005) 715-735

Some results: EnKF vs. SEIK

- EnKF94/98
  - very simple to implement
  - costly (compute analysis update in observation space)
  - observation ensemble introduces sampling errors
  - random ensemble initialization has slow convergence
- SEIK
  - more difficult to implement
  - much faster (analysis update in ensemble space)
  - faster convergence with initialization using singular value decomposition (empirical orthogonal functions)

What makes SEIK faster than EnKF?

Nerger, Hiller, Schröter. Tellus 57A (2005) 715-735

Two features of the SEIK filter

1. **Avoid perturbing observations**
   - Apply two step update:
     1. Update ensemble mean state
     2. Transform forecast ensemble to represent analysis **P**

2. **Ensemble transformation in ensemble space**
   - Degrees of freedom of analysis: *ensemble size – 1*
   - EnKF uses update in observation space (usually much larger than ensemble size)

Typical for *ensemble square-root Kalman filters*

Efficient use of ensembles

Kalman gain

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

Alternative form (Sherman-Morrison-Woodbury matrix identity)

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

Looks worse: $n \times n$ matrices need inversion

However: with ensemble $\tilde{\mathbf{P}}_k^f = (N-1)^{-1} \mathbf{X}' \mathbf{X}'^T$

$$\tilde{\mathbf{K}}_k = \mathbf{X}' \left[(N-1)\mathbf{I} + \mathbf{X}'^T \mathbf{H}^T \mathbf{R}^{-1} \mathbf{H} \mathbf{X}' \right]^{-1} \mathbf{X}'^T \mathbf{H}^T \mathbf{R}^{-1}$$

Inversion of $N \times N$ matrix

(Ensemble perturbation matrix $\mathbf{X}' = \mathbf{X} - \bar{\mathbf{X}}$)

Which filter should one use?

Many choices - a little “zoo” (not complete):

EnKF(94/98)

Studied in Nerger et al. Tellus (2005)

RRSQRT

ROEK

SEEK

SEIK

New study: Nerger et al., Mon. Wea. Rev. (2012)

EnKF(2003)

EnKF(2004)

EAKF

EnSRF

ETKF

ESTKF

MLEF

SPKF

ESSE

RHF

New filter formulation

Computations in ensemble-spanned space

Square root of covariance matrix (ensemble size *N*, state dim *n*)

$$\mathbf{Z} = \mathbf{X}^f \mathbf{T} \qquad \mathbf{P}^f = \mathbf{Z}\mathbf{Z}^T$$

Transformation matrix in ensemble space (small matrix)

$$\mathbf{A} = \left(\mathbf{G} + (\mathbf{H}\mathbf{Z})^T \mathbf{R}^{-1} \mathbf{H}\mathbf{Z}\right)^{-1}$$

Analysis state covariance matrix

$$\mathbf{P}^a = \mathbf{Z}\mathbf{A}\mathbf{Z}^T$$

Ensemble transformation based on square root of $\mathbf{A}$

$$\mathbf{X}^a \sim \mathbf{Z}\mathbf{L} \qquad \mathbf{L}\mathbf{L}^T = \mathbf{A}$$

Very efficient:
Transformation matrix computed in space of dim. *N or N-1*

The T matrix

SEIK and ETKF use different projections **T**

$$\mathbf{Z} = \mathbf{X}^f \mathbf{T}$$

➜ results in slightly different ensemble transformations

➜ SEIK is slightly faster than ETKF

ETKF uses minimal ensemble transformation – desirable feature!

Error Subspace Transform Kalman Filter (ESTKF)

Combine advantages of SEIK and ETKF

Redefine **T:**

1. Remove ensemble mean from all columns
2. Subtract fraction of last column from all others
3. Drop last column

Features of the ESTKF:

- Same ensemble transformation as ETKF
- Slightly cheaper computations
- Direct access to ensemble-spanned error space

L. Nerger et al., Monthly Weather Review 140 (2012) 2335-2345

Requirements for applying ensemble Kalman filters

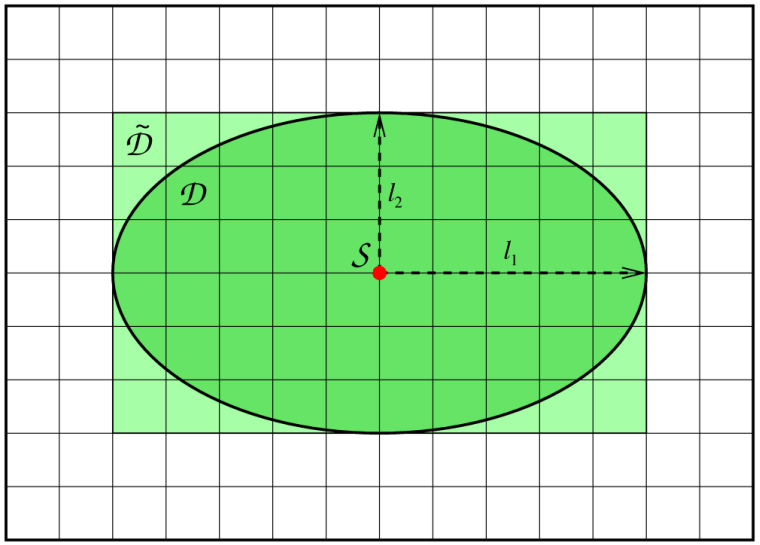
"Pure" ensemble-based Kalman filters have usually bad performance

- e.g. due to small ensemble size

Improvements through

- Covariance inflation
- Localization
- Model error simulation

Localization



S: Analysis region
D: Corresponding data region

Implementation Aspects

Computational and Practical Issues

Data assimilation with ensemble-based Kalman filters is costly!

Memory: Huge amount of memory required
(model fields and ensemble matrix)

Computing: Huge requirement of computing time
(ensemble integrations)

Parallelism: Natural parallelism of ensemble integration exists
(needs to be implemented)

„Fixes“: Filter algorithms do not work in their pure form
(„fixes“ and tuning are needed)
because Kalman filter optimal only in linear case

Implementing Ensemble Filters & Smoothers

Ensemble forecast

- can require model error simulation
- naturally parallel

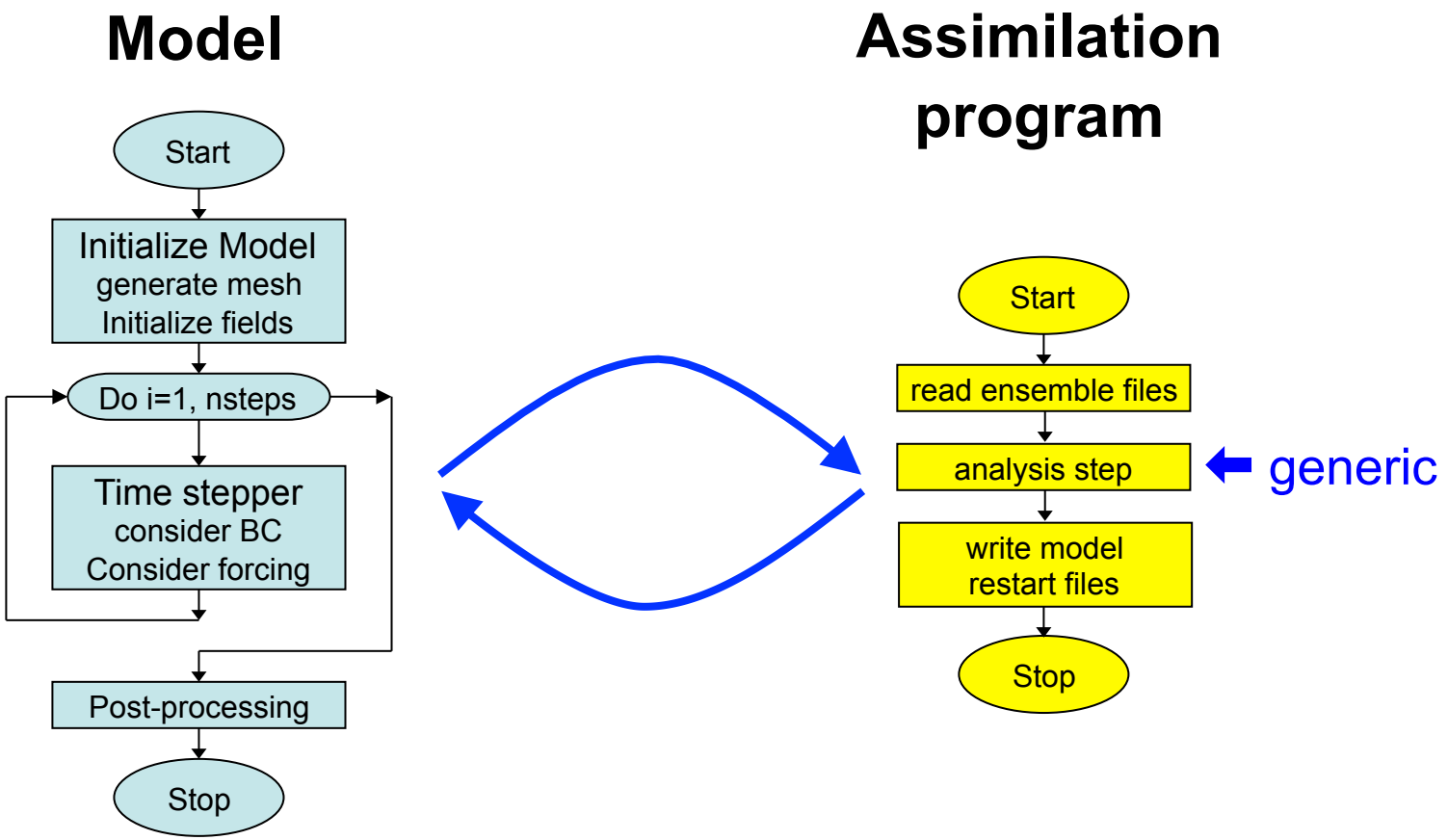
Analysis step of filter algorithms operates on abstract state vectors
(no specific model fields)

Analysis step requires information on observations

- which field?
- location of observations
- observation error covariance matrix
- relation of state vector to observation

➜ Analysis step can be implemented independently of model!

Offline mode – separate programs

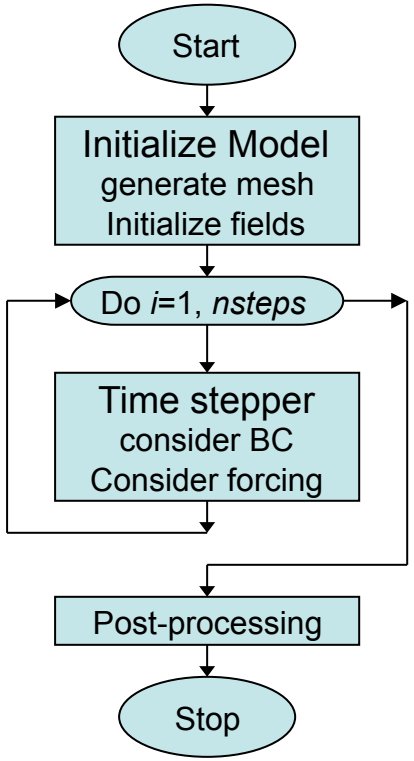


For each ensemble state

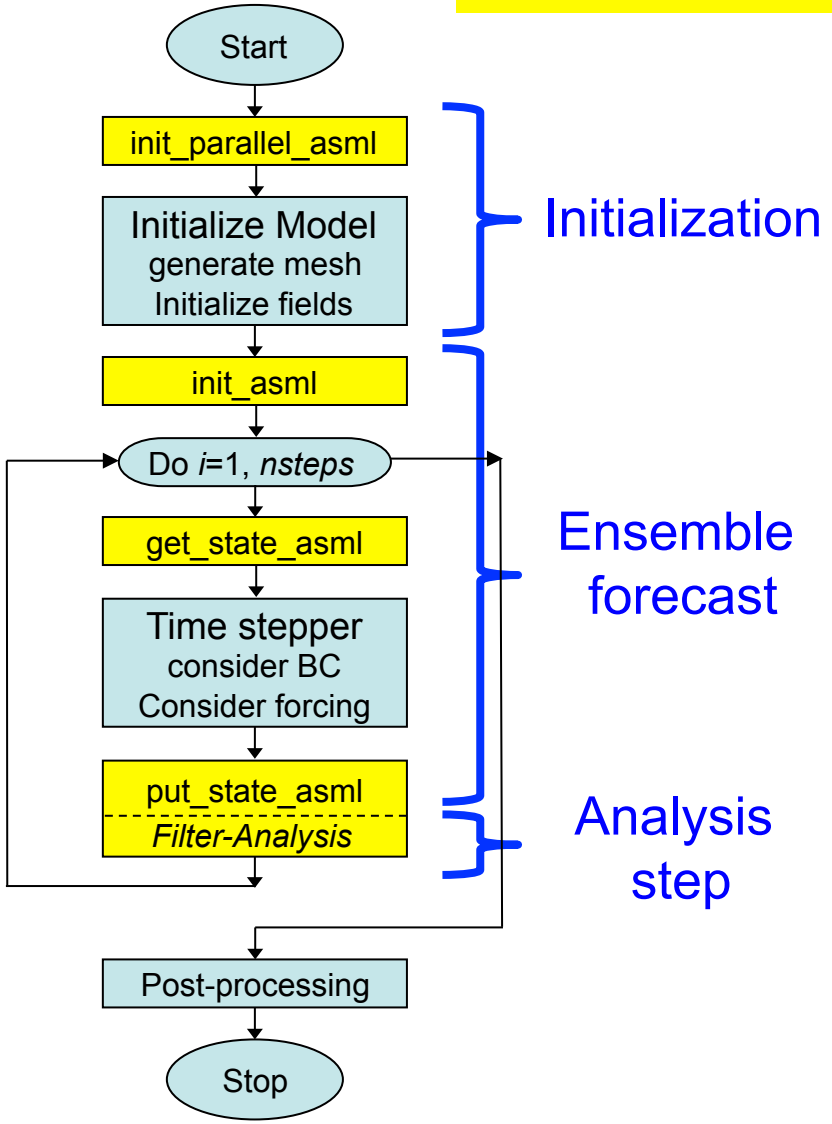
- Initialize from restart files
- Integrate
- Write restart files

- Read restart files (ensemble)
- Compute analysis step
- Write new restart files

Model



Extension for data assimilation



Online assimilation program:
➜ Avoid expensive writing and reading of ensemble files

Features of online program

- minimal changes to model code when combining model with filter algorithm (adding 4 routines)
- model not required to be a subroutine
- no change to model numerics
- control of assimilation program coming from model
- filter method encapsulated in subroutine
- simple switching between different filters and data sets
- complete parallelism in model, filter, and ensemble integrations

Implementation structure can be implemented in a generic framework (for online and offline modes)

PDAF: A tool for data assimilation

PDAF Parallel Data Assimilation Framework

PDAF - Parallel Data Assimilation Framework

- an environment for ensemble assimilation
- provide support for ensemble forecasts
- provide fully-implemented filter algorithms
- for testing algorithms and real applications
- useable with virtually any numerical model
- makes good use of supercomputers

Open source:
Code and documentation available at

http://pdaf.awi.de

L. Nerger, W. Hiller, Computers & Geosciences 55 (2013) 110-118

AWI

Current algorithms in PDAF

PDAF Parallel Data Assimilation Framework

PDAF originated from comparison studies of different filters

Filters

- EnKF (Evensen, 1994)
- ETKF (Bishop et al., 2001)
- SEIK filter (Pham et al., 1998)
- SEEK filter (Pham et al., 1998)
- ESTKF (Nerger et al., 2012)

Global filters

- LETKF (Hunt et al., 2007)
- LSEIK filter (Nerger et al., 2006)
- LESTKF (Nerger et al., 2012)

Localized filters

Smoothers for

- ETKF/LETKF
- ESTKF/LESTKF
- EnKF

AWI

Application examples

- Assimilation of satellite altimetry
  (Project GEOTOP, @ AWI)
- Ocean chlorophyll assimilation into global NASA Ocean Biogeochemical Model (with Watson Gregg, NASA GSFC)
  - ➢ Generation of daily re-analysis maps of chlorophyll at ocean surface
- Coastal assimilation of ocean surface temperature
  (within project “DeMarine”, AWI and BSH)
  - ➢ Improve operational forecast skill in North Sea and Baltic Sea

+ external users, e.g. with

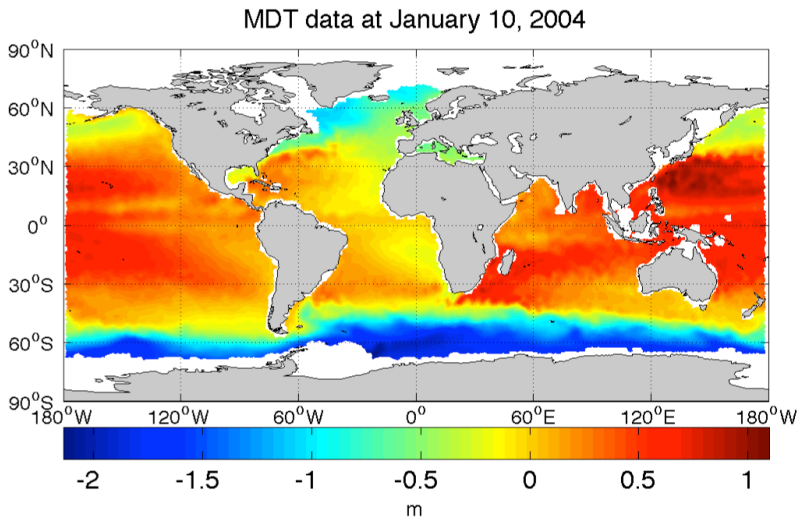
- MPI-OM (S. Brune/J. Baehr, MPI Hamburg)
- PARODY (A. Fournier, IPGP Paris)
- ADCIRC (I. Hoteit, KAUST, Saudi Arabia)

Application:

Assimilation of ocean topography data

Dynamic Topography Data

- Generated by IAPG Munich (A. Abertella within GEOTOP project)
- Difference of time-dependent altimetric sea surface height and geoid data
- SSH: altimeter data from
  ERS-1/2, ENVISAT,
  TOPEX/Poseidon, Jason-1/2
- Geoid: based on data from
  GRACE & GOCE
- SSH and geoid filtered to d/o 120 for consistency

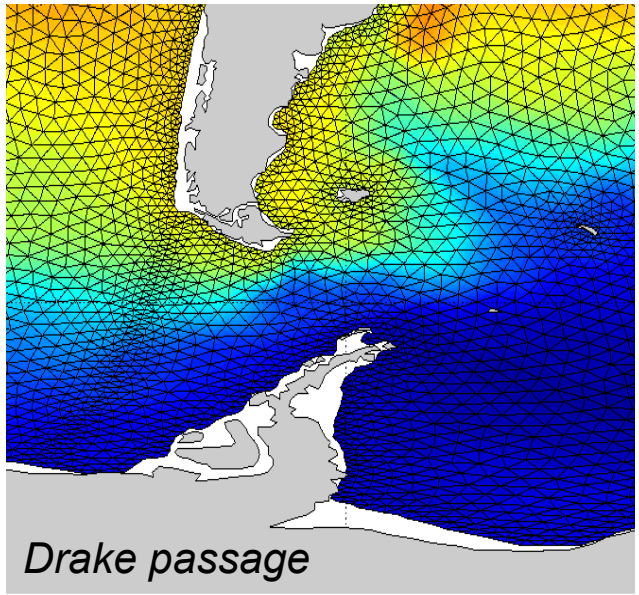


Global ocean model

FESOM (Finite Element Sea-ice Ocean model, Danilov et al. 2004)

Global configuration

- 1.3^{o} resolution, 40 levels
- horizontal refinement at equator
- state vector size 10^7


Drake passage

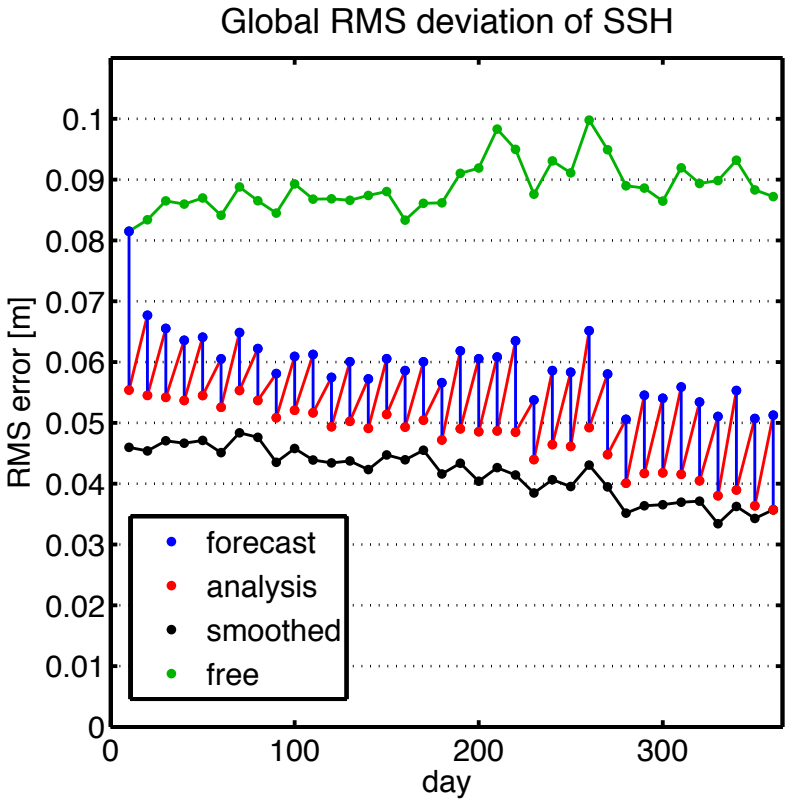
Experiments with DOT data

- ensemble size 50
- assimilate each 10th day over 1 year
- ESTKF with smoother extension and localization (using PDAF environment as single program)
- inflation tuned for optimal performance (ρ=0.9)
- run using 2000 processor cores (Timings: forecasts 8800s, filter+smoother 200s)

Assimilation impact

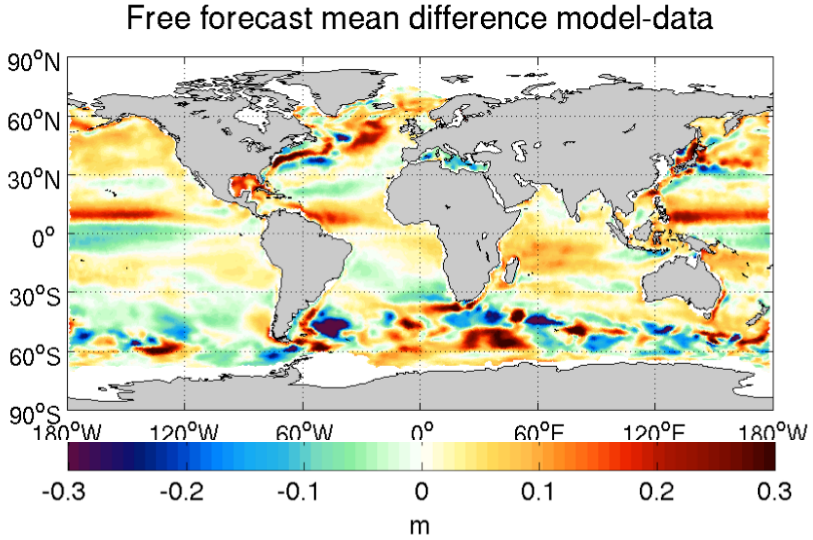
Sea surface height

- assimilation reduces deviation between data and model (as it has to...)
- growth of deviation during forecast phase
- further reduction of deviation by smoother

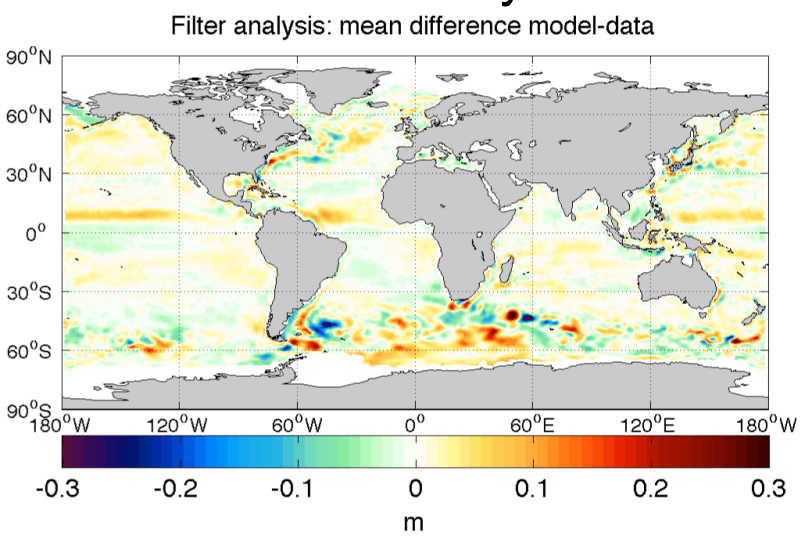


Sea surface height (~DOT)

Free forecast



Filter analysis



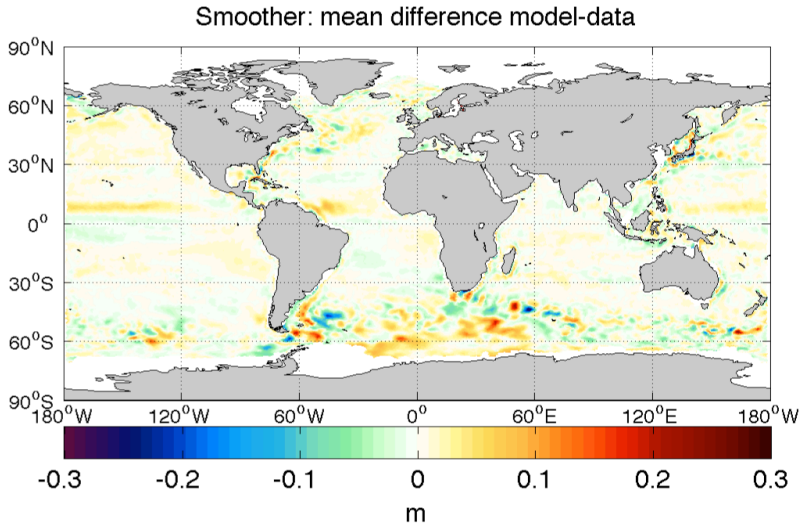
- Pattern with free forecast typical for forcing type (CORE-II)

With assimilation

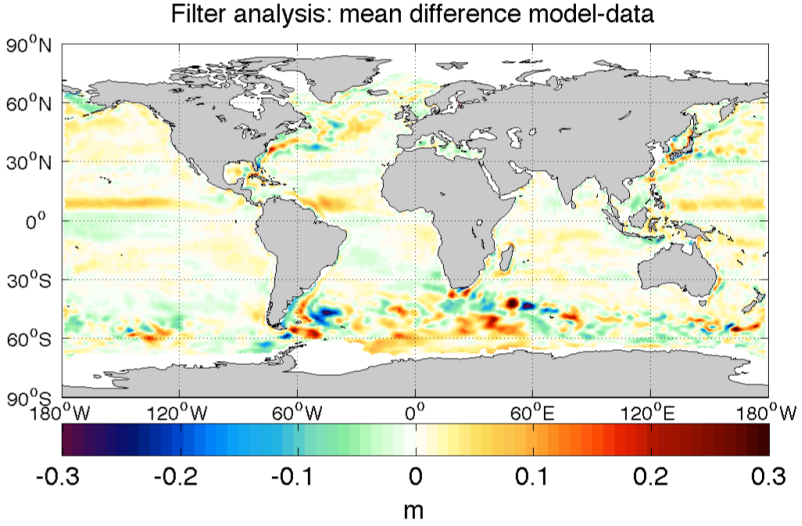
- Significant reduction of deviatons
- Largest deviations in southern ocean
- Deviations partly induced by forcing (windstress) as bias

Effect of Smoother

Smoother

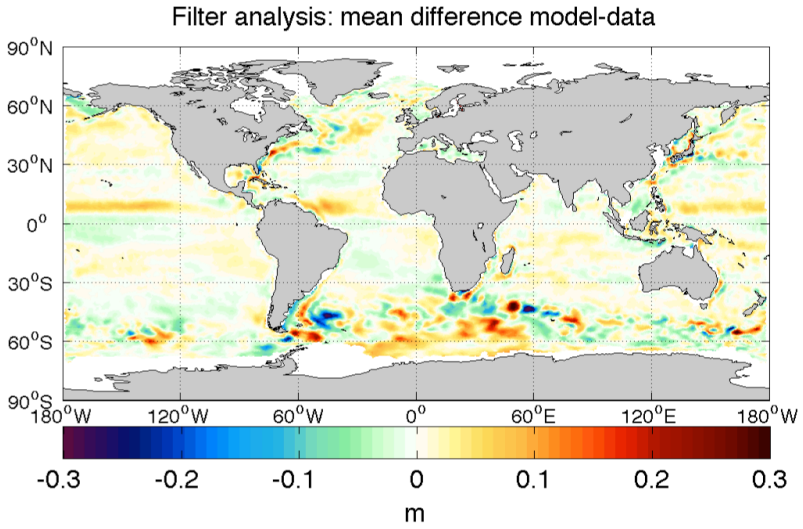


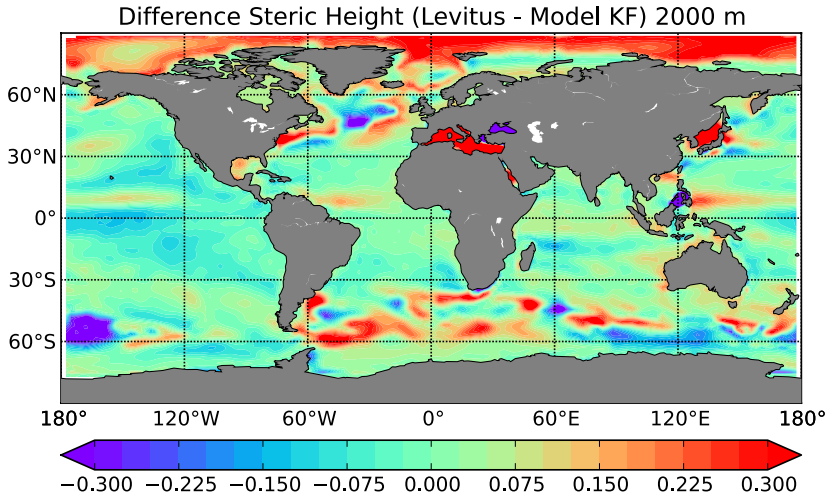
Filter



- Further reduction of deviations

Influence of assimilation on the ocean state



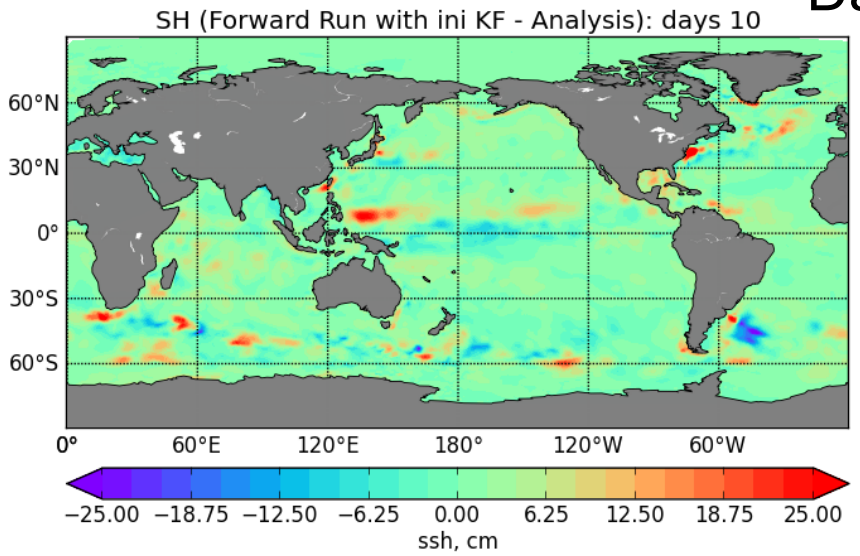


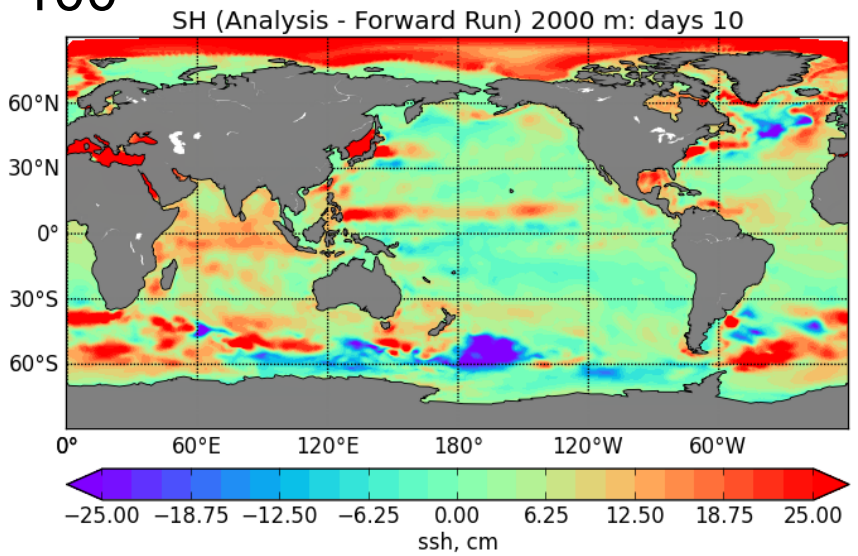
- steric height:
  - Vertically integrated height variation compared to reference density (function of temperature & salinity)
- SSH deviations widely correspond to differences in steric height
- assimilation improves steric height (hence the water column)

How does model preserve assimilation changes?

- Initialize free model run from assimilation at day 30 (after 3 analysis steps)

Day: 100



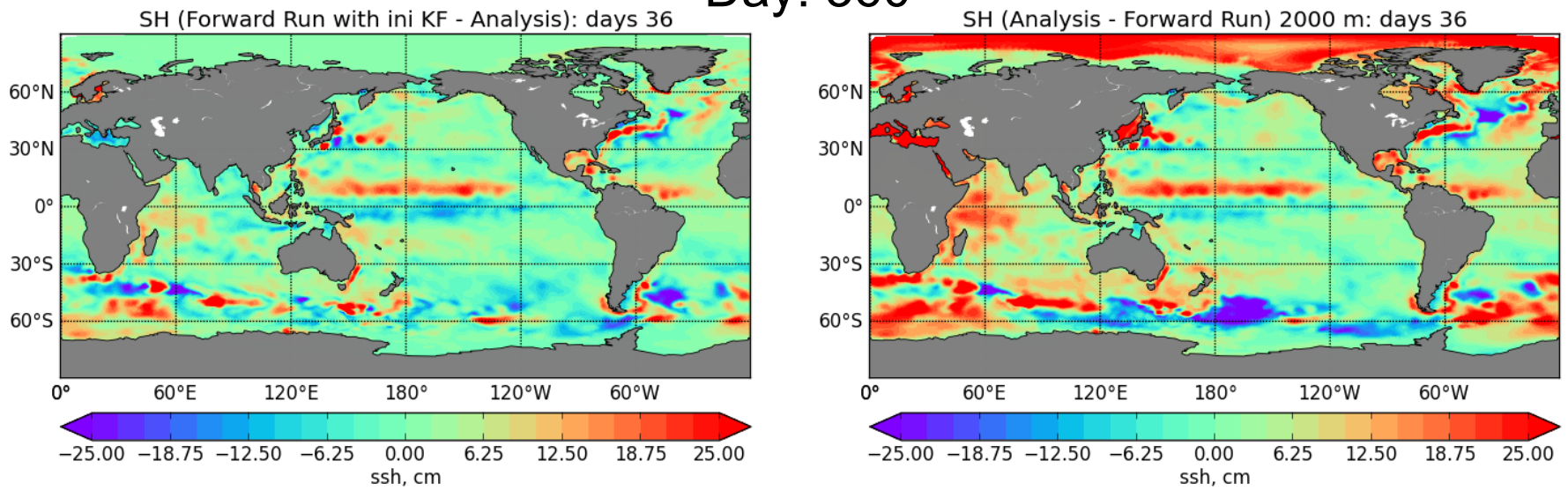


- Very small deviations after 70 days free forecast
- Forecast without any assimilation shows much larger deviations

How does model preserve assimilation changes?

- Initialize free model run from assimilation at day 30 (after 3 analysis steps)

Day: 360



- Notable deviations after 290 days free forecast
- Deviations still smaller than without any assimilation

Conclusion

Current ensemble Kalman filters

- efficient utilization of error space updates

Abstraction allows for generic filter implementation

- efficient assimilation framework possible

Assimilation of DOT data

- significant improvements of model state
- improvements preserved over longer time interval

Thank you!