Ensemble Data Assimilation

at the Alfred Wegener Institute

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

AWI

Outline

Research directions at AWI:

- Applications
- Algorithms
- Software development

Outline

- Parallel Data Assimilation Framework - PDAF
- Projects applying Data Assimilation
- Algorithmic developments

Parallel Data Assimilation Framework

PDAF

PDAF: A tool for data assimilation

PDAF - Parallel Data Assimilation Framework

- a software to provide assimilation methods
- for testing algorithms and real applications
- useable with virtually any numerical model
- also:
    - apply identical methods to different models
    - test influence of different observations
- makes good use of supercomputers
  (Fortran and MPI; tested on up to 4800 processors)

More information and source code available at

http://pdaf.awi.de

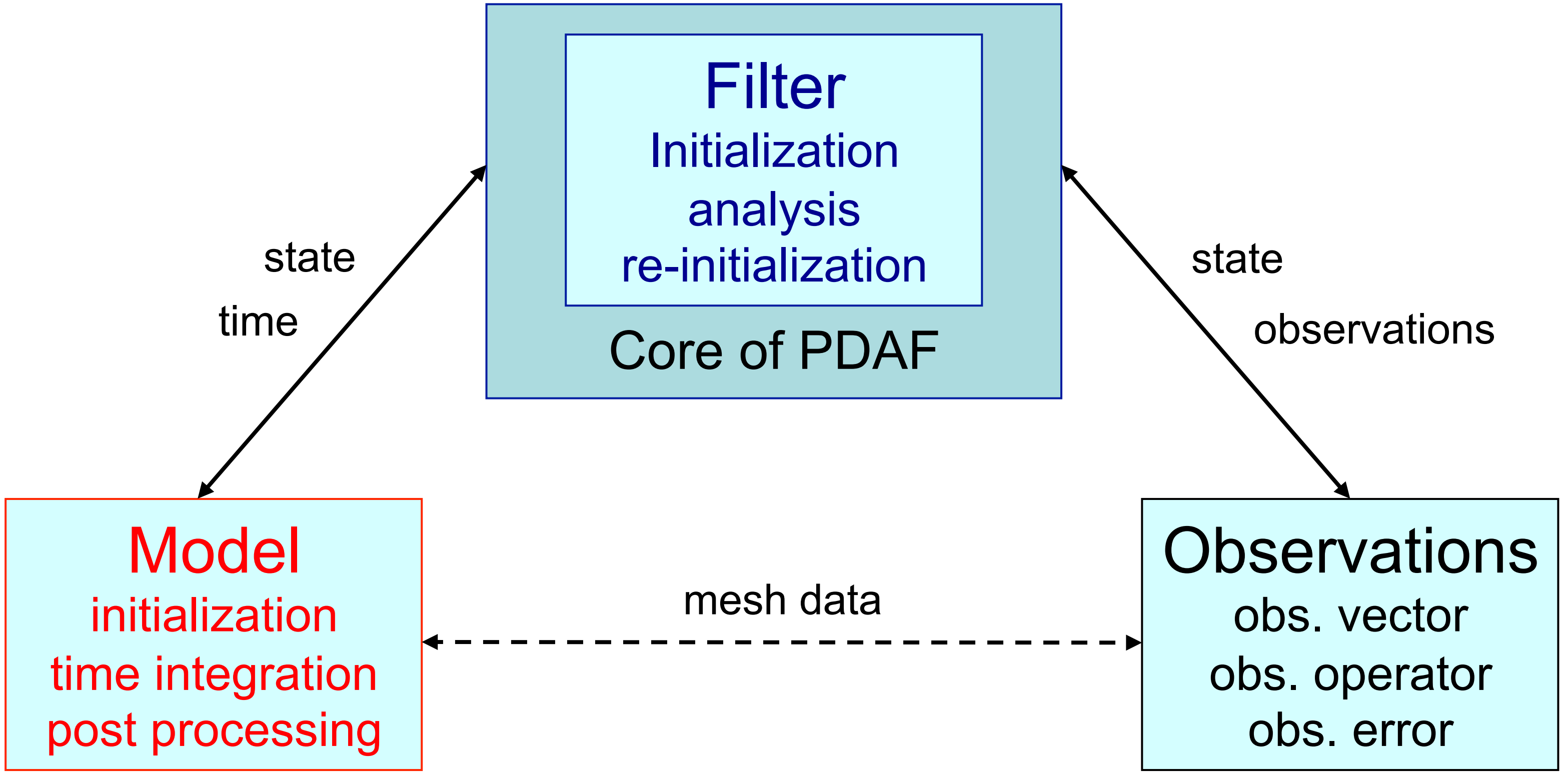
Models and Filter Algorithms

- Sequential assimilation algorithms require limited information
  - no physics needed!
  - relation of model fields to state vector
  - observations (time, type, location, error)

Because of this:

- Filter algorithms can be developed and implemented independently from model
- Model can be developed independently from the filter
- Parallelization of ensemble forecast can be implemented independently from model

Logical separation of assimilation system



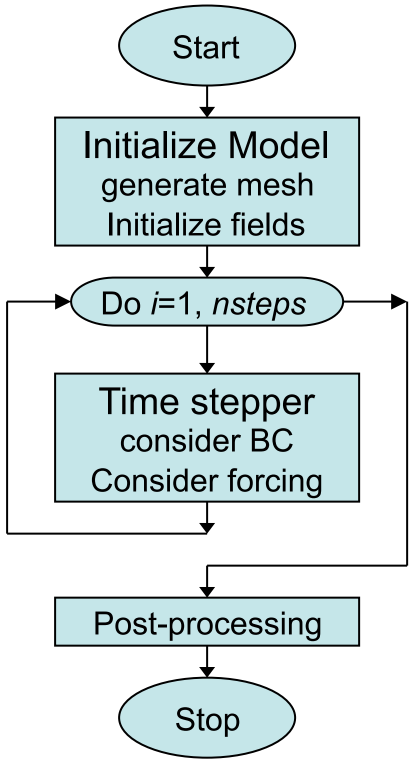
For online implementation:

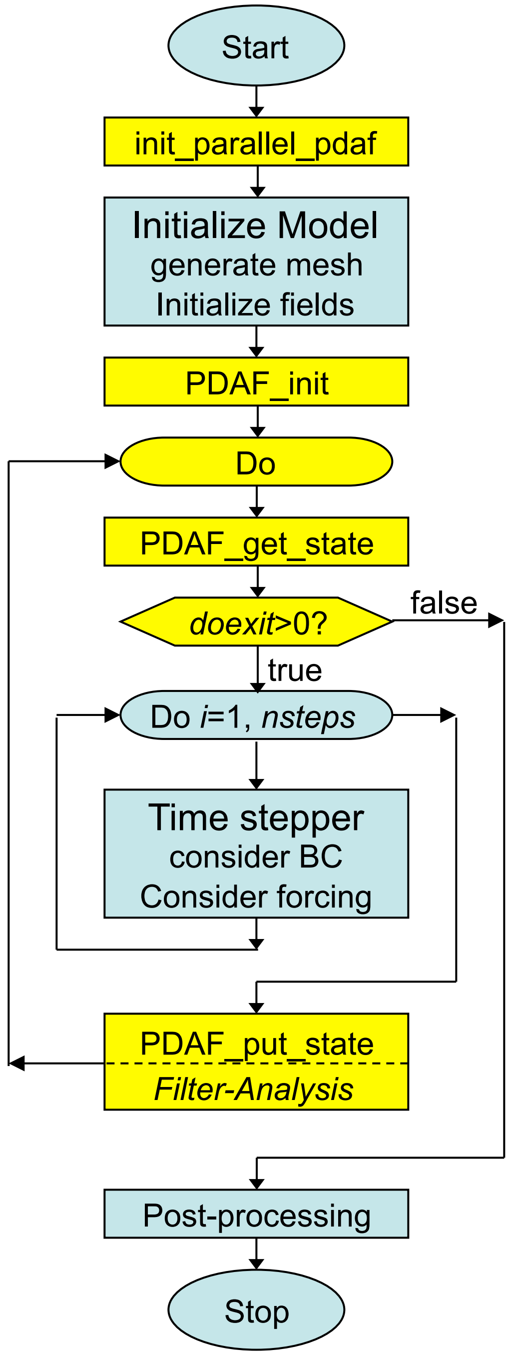
⟷ Explicit interface

⇠- - - -⇢ Indirect exchange (Fortran: module/common)

L. Nerger et al. (2005) in Use of High Performance Computing in Meteorology - Proceedings of the 11th ECMWF Workshop / Eds. W. Zwieflhofer, G. Mozdzynski. World Scientic, pp. 63-83

Model





Extension for data assimilation

External Do-loop can be avoided – lower flexibility!

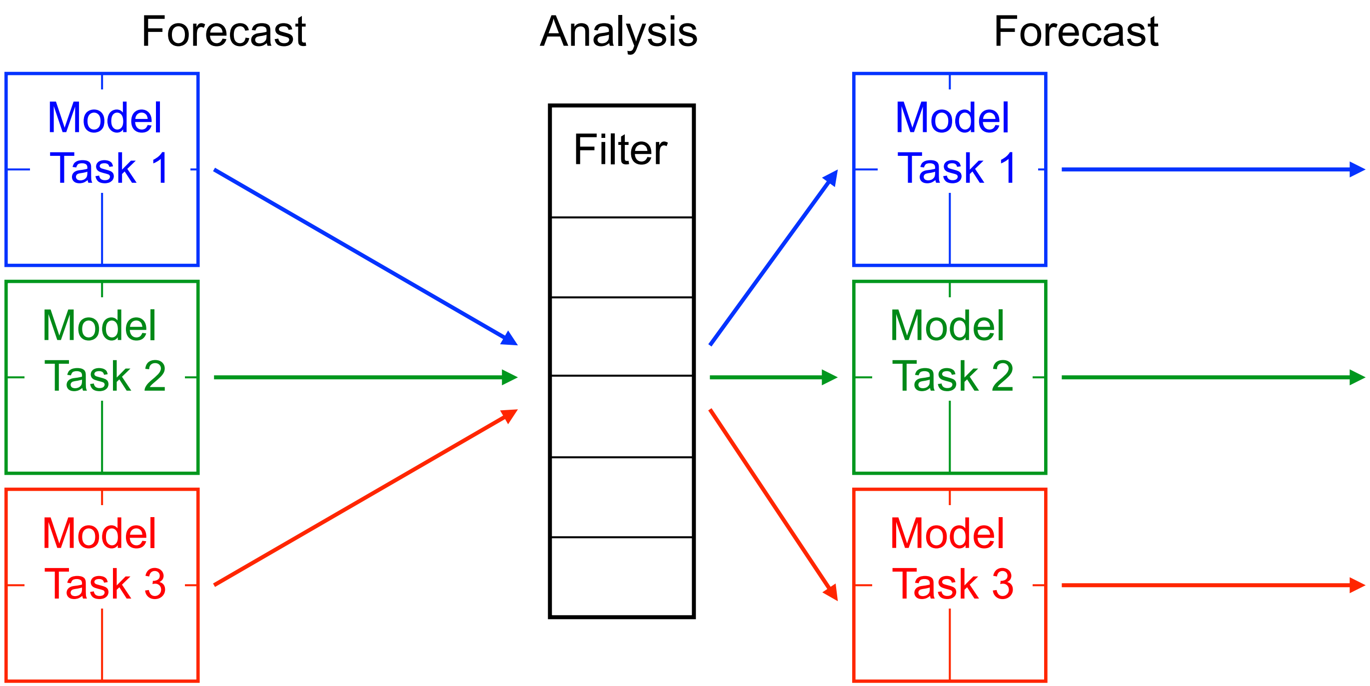
PDAF's Interface

Don't adapt the model to the assimilation system

➜ Attach DA functionality to model

- Model time stepper not required to be subroutine
- Model-sided configuration of assimilation system
- Low abstraction level for optimal performance
- Interface independent of filter
  (except for names of user-supplied subroutines)
- User-supplied routines for elementary operations in model context (e.g. using modules of model code):
  - field transformations between model and filter
  - observation-related operations

2-level Parallelism



1. Multiple concurrent model tasks
2. Each model task can be parallelized

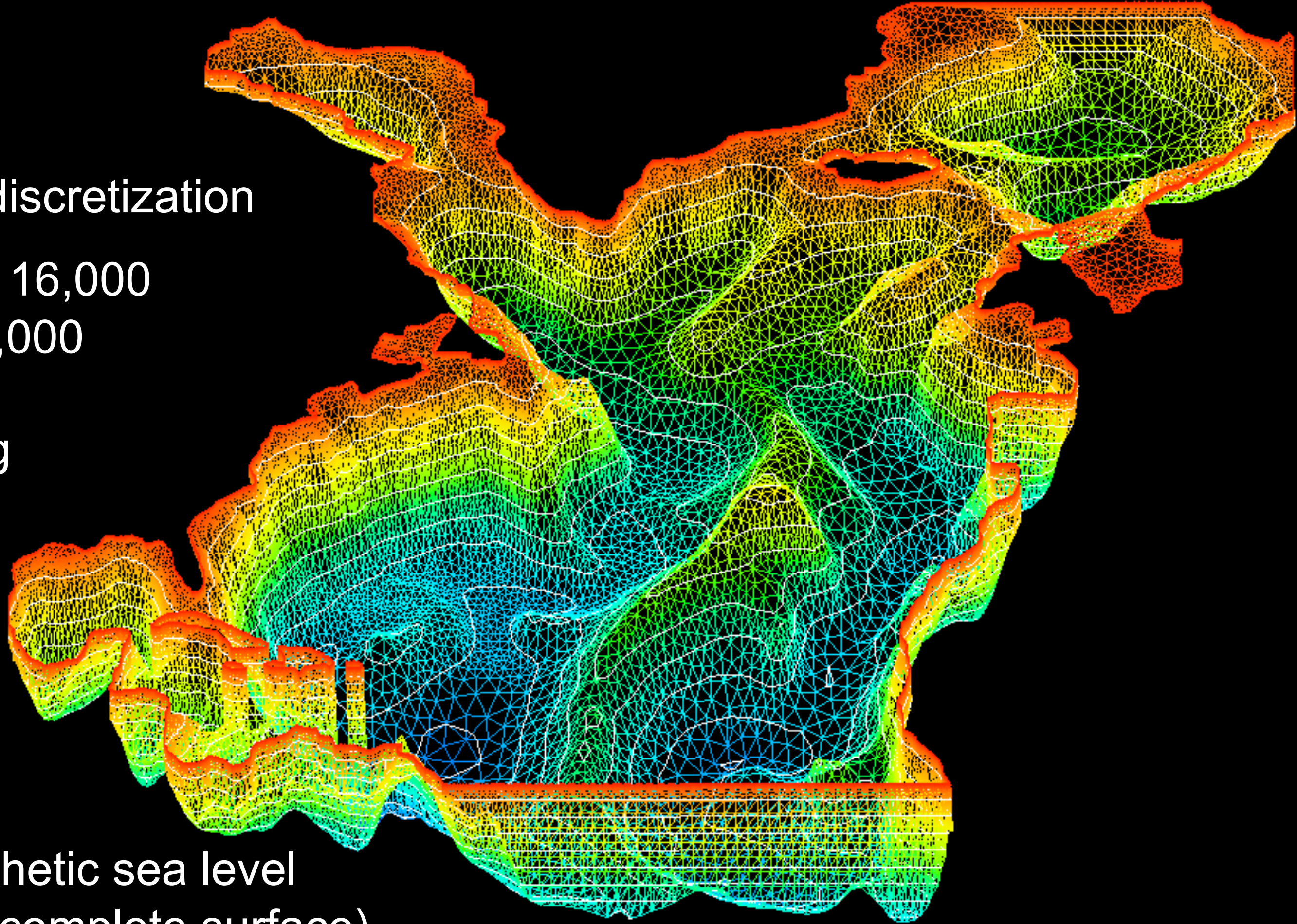
➢ Analysis step is also parallelized

Application case: FEOM – Coarse North Atlantic

Finite Element Ocean Model

finite-element discretization

surface nodes: 16,000
3D nodes: 220,000
z-levels: 23
eddy-permitting



Assimilate synthetic sea level data (10 days, complete surface)

Parallel Performance – DA system

Use between 64 and 4096 processors of SGI Altix ICE cluster (Intel processors)

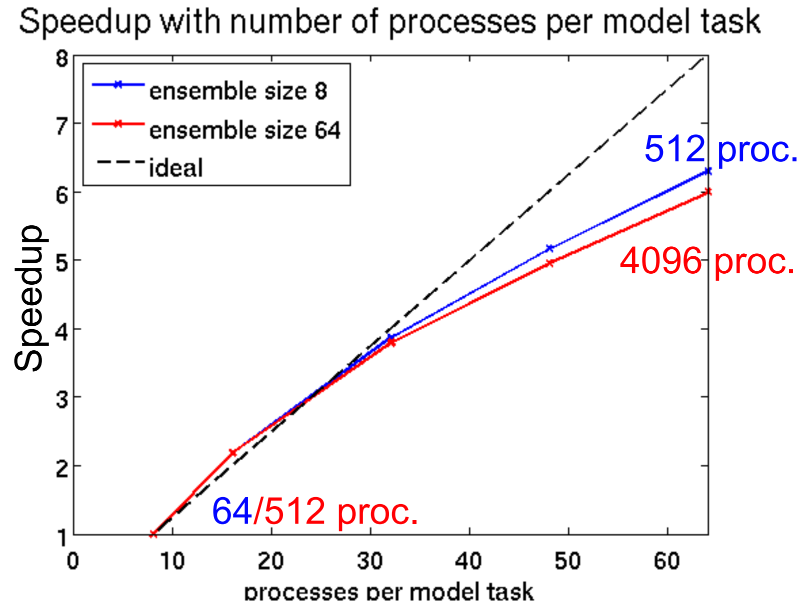
94-99% of computing time in model integrations

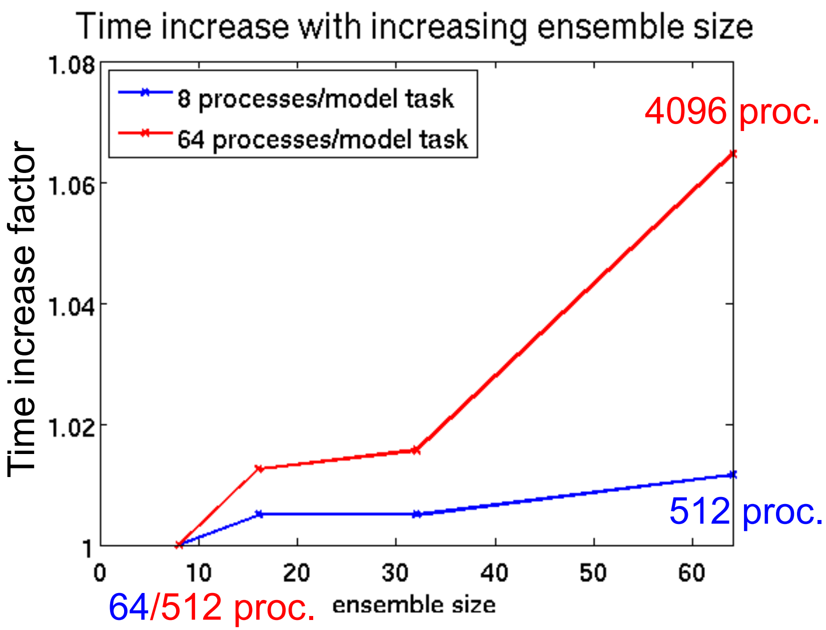
Speedup: Increase number of processes for each model task, fixed ensemble size

- factor 6 for 8x processes/model task
- one reason: time stepping solver needs more iterations

Scalability: Increase ensemble size, fixed number of processes per model task

- increase by ~7% from 512 to 4096 processes (8x ensemble size)
- one reason: more communication on the network





Parallel Performance – Filter only

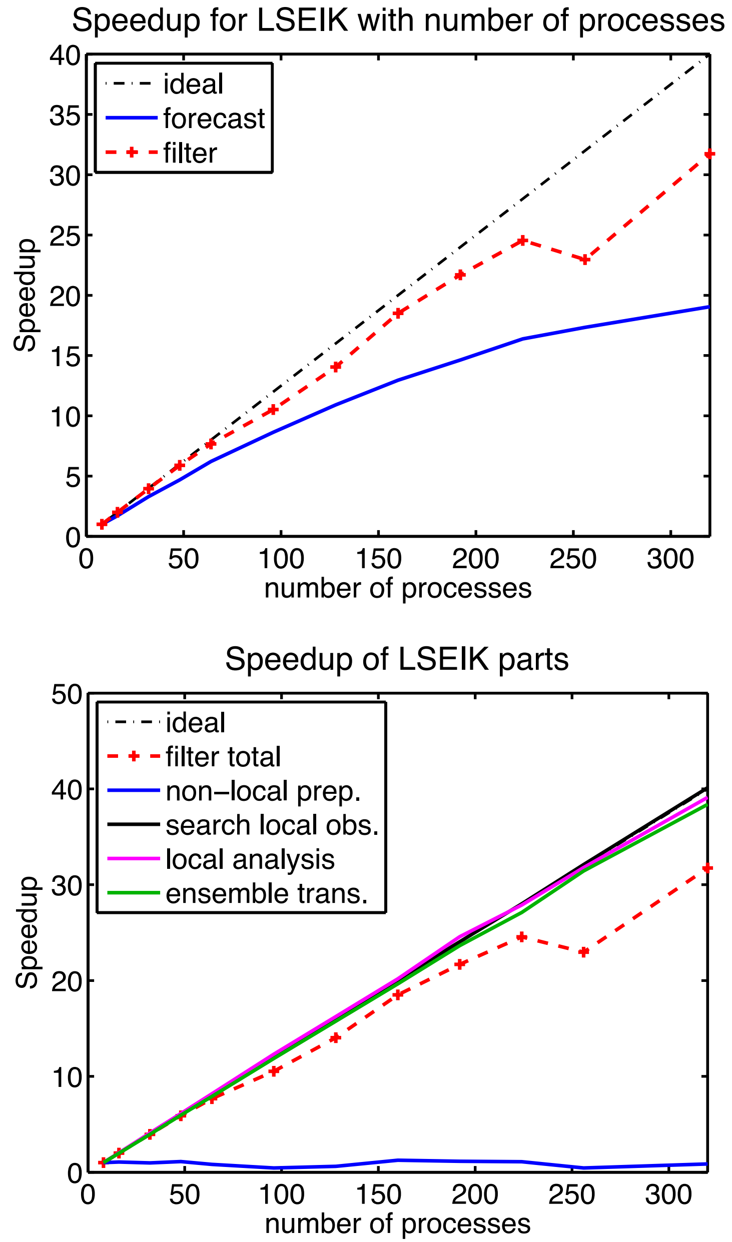
- Use between 8 and 320 processors; larger mesh (55.000 surface nodes)
- Assimilate each time step with LSEIK
- Up to 50% of computing time in filter analysis

Filter in total:

- Very good speedup up to 224 processes.
- 80% efficiency at 320 processes.
- Smaller speedup for forecasts

Filter parts:

- Most parts show ideal speedup
- Constant time for non-local preparation (Negligible cost for 8 processors)
  - read observations, initialize innovation



Existing Online Implementations

- FEOM (Finite-Element Ocean Model)
  - PDAF's "home" model; all features
- MIPOM (met.no, by I. Burud)
  - First implementation not done by myself
- NOBM (NASA Ocean-Biogeochemical Model)
  - For ocean-color assimilation
- BSHcmod (Project DeMarine Environment)
  - Toward operational use in North Sea and Baltic Sea
- OMCT (GFZ Potsdam, J. Saynisch)
  - Assimilating ocean angular momentum data
- ADCIRC (at KAUST, I. Hoteit, with U. Altaf)
  - 3 days for basic implementation

Assimilation for operational forecasting

In the North and Baltic Seas

(Project DeMarine Environment)

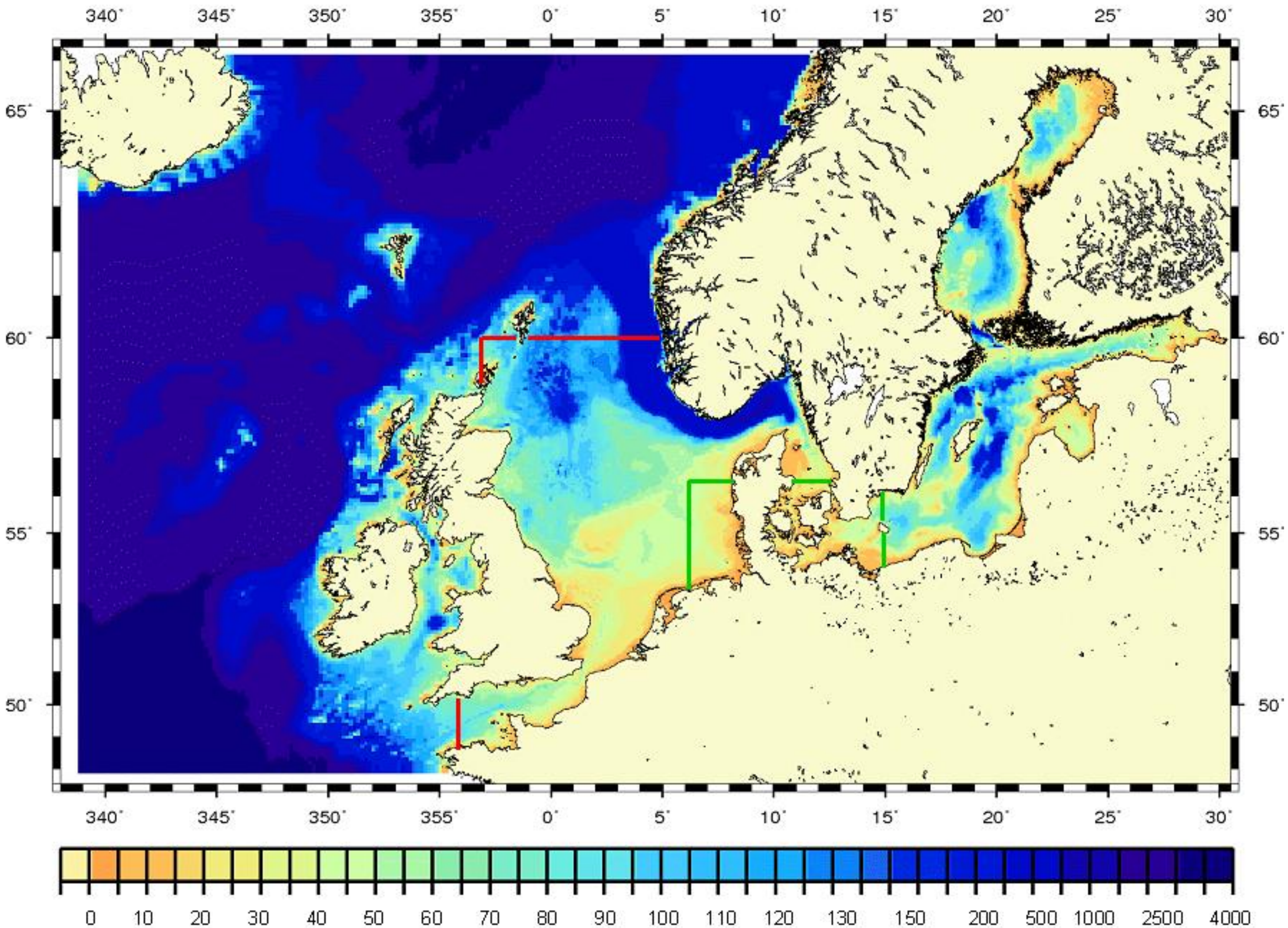
Cooperation of AWI and BSH
(German Maritime and Hygrographic Agency)

S. Loza, L. Nerger, J. Schröter (AWI)

F. Janssen, S. Massmann (BSH)

DeMarine Environment

Operational BSH Model (BSHcmod), Version 4



BSSC 2007, F. Janssen, S. Dick, E. Kleine

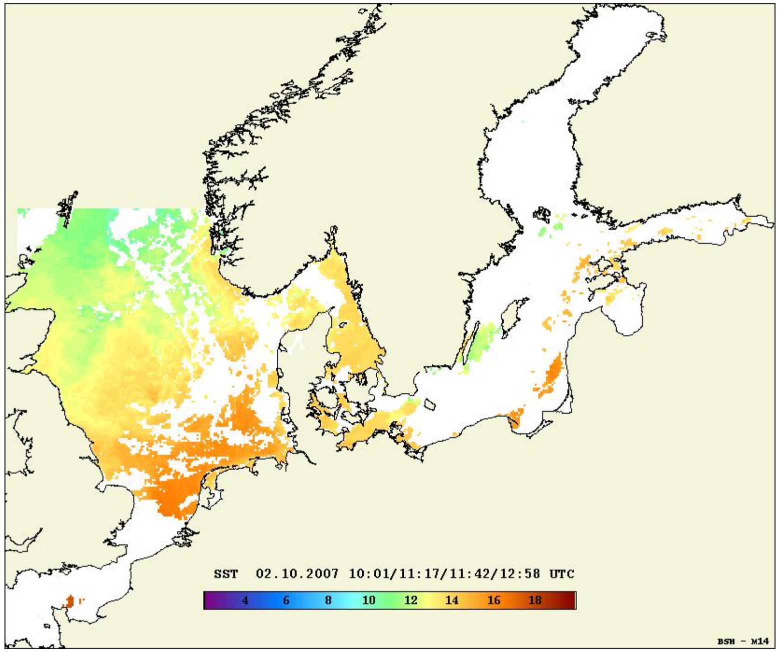
Grid nesting:

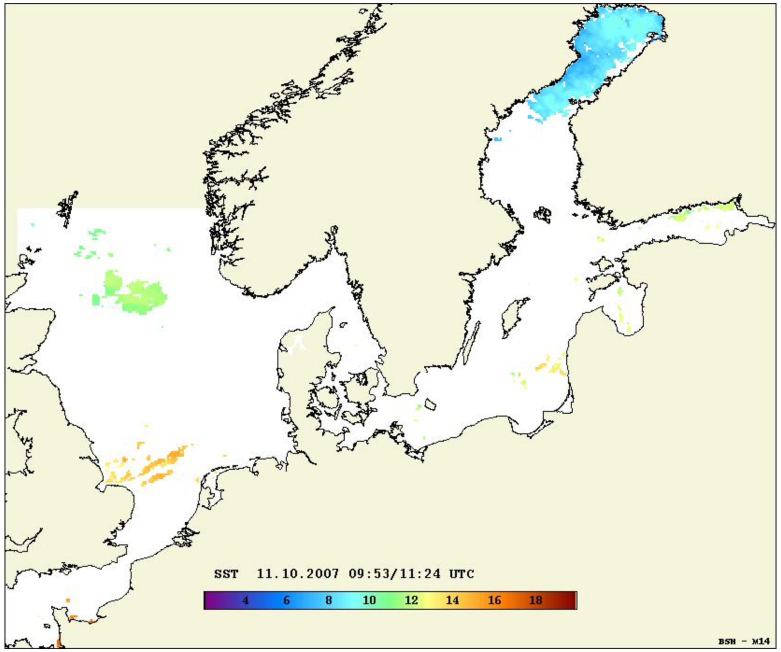
- 10 km grid
- 5 km grid
- 900 m grid

Data assimilation: 5 km grid

DeMarine Environment

Assimilated Data



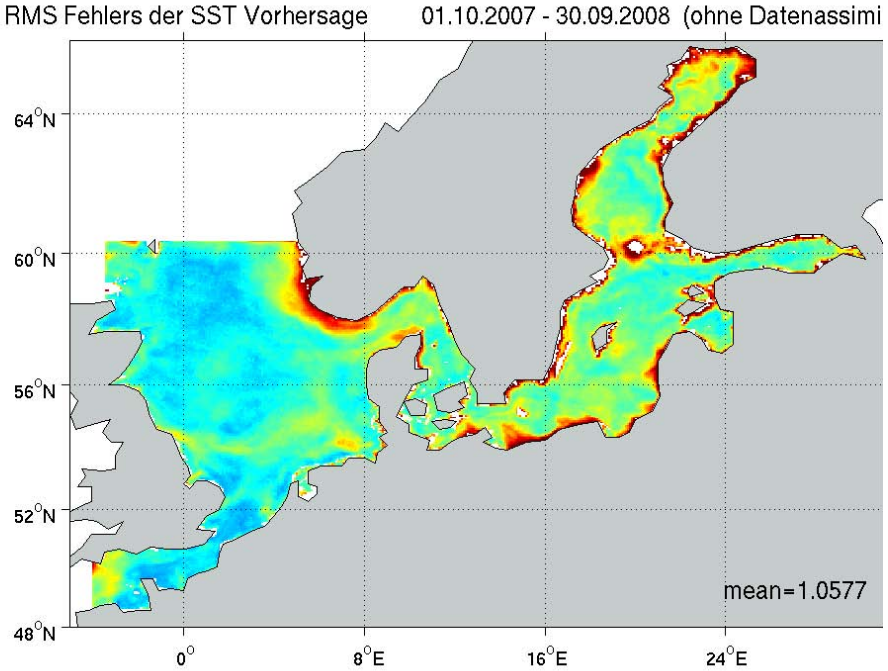


- Surface temperature (NOAA satellite data)
- 12-hour time window
- Strong variation of data coverage (clouds)
- Use observation error: 0.8 ^{o}C (empirical)

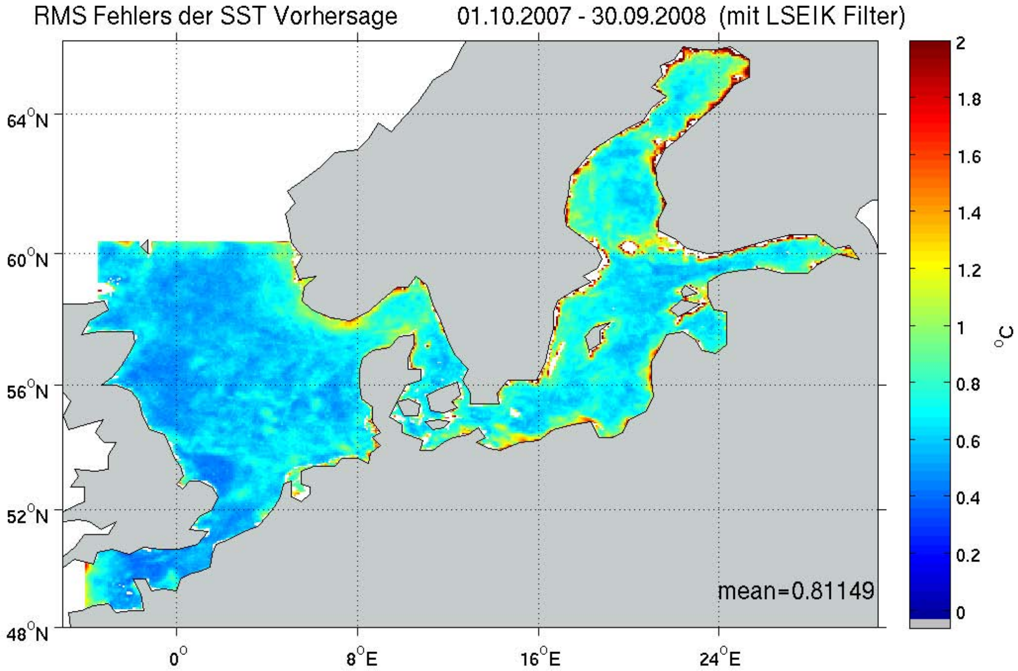
Deviation from NOAA Satellite Data

- Mean RMS over 1 year (10/2007 – 9/2008)
- Significant reduction of erros (spatial mean ~0.2 – 0.3°C)

No assimilation



Assimilation with LSEIK Filter



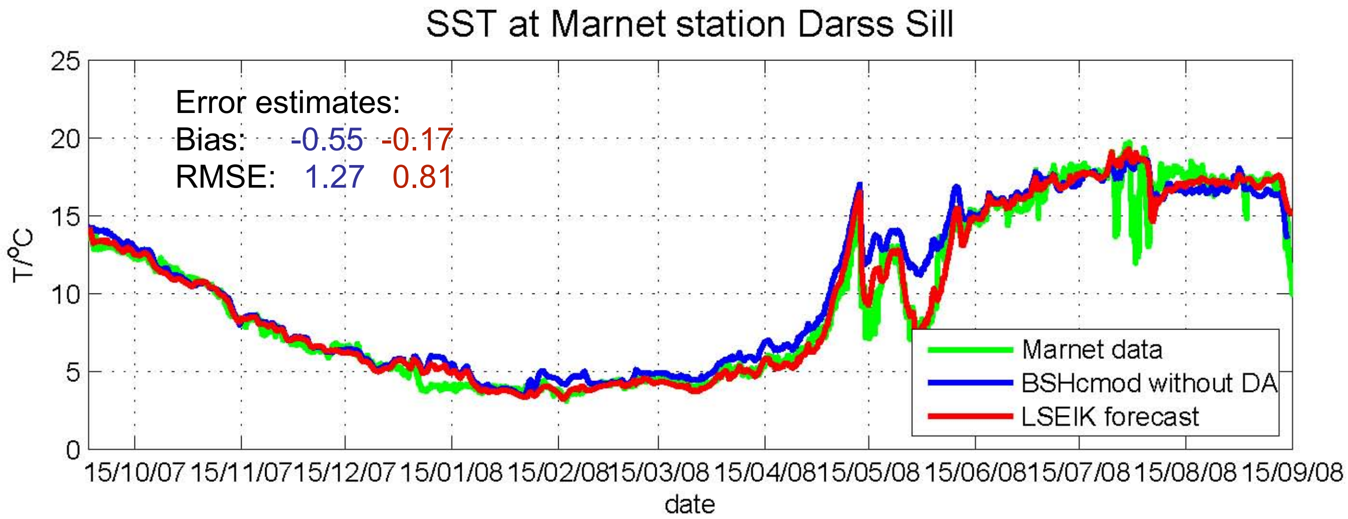
Validation with independent data

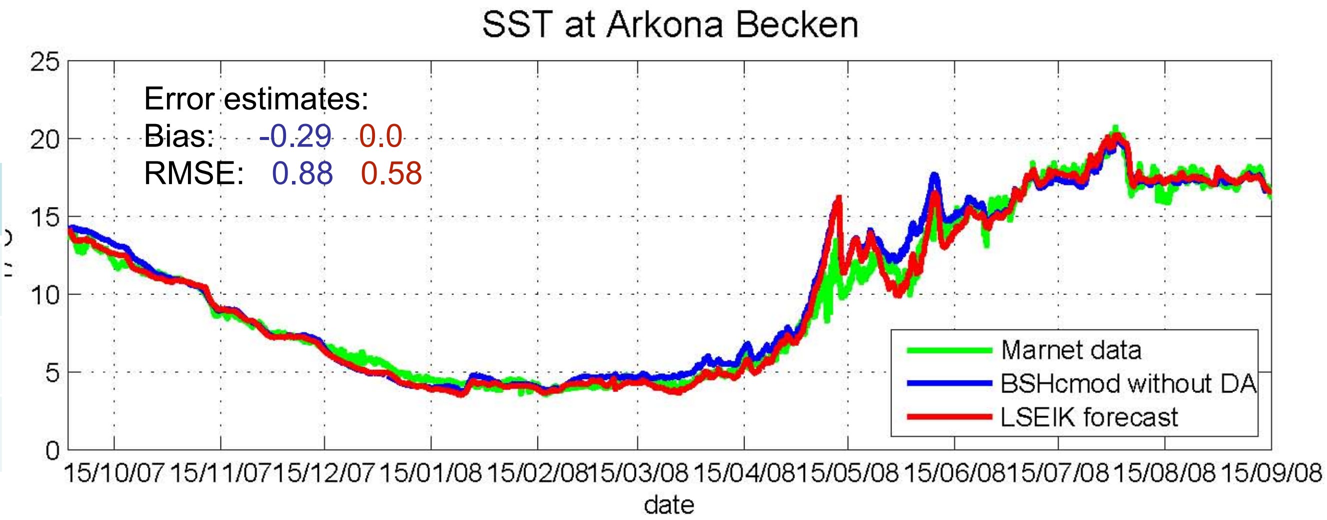
MARNET station data

- Reduction of
  - Bias
  - RMS error

1 year mean over 6 stations:

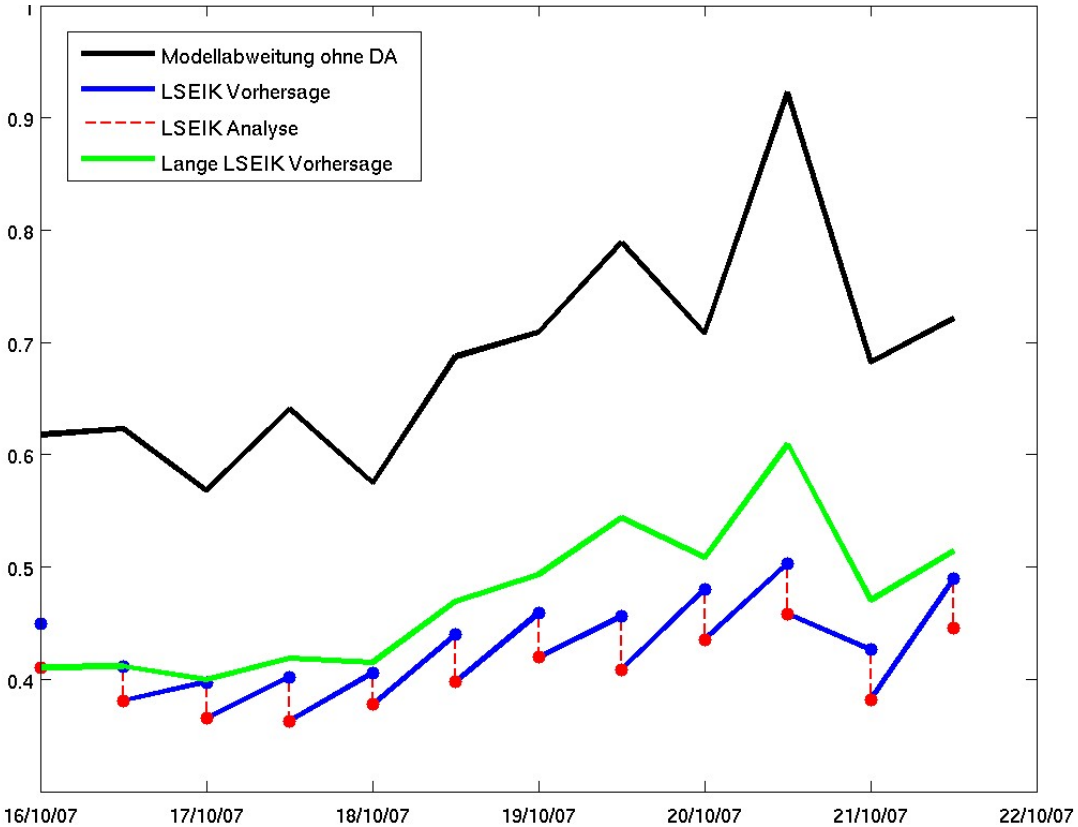
	RMSe	bias
free	0.87	0.3
data	0.59	0.11
asml	0.55	0.08





Improvement of long forecasts

RMS error over time



black: free model run

Blue/red: 12h assimilation/analysis cycles

green: 5 day forecast

➜ Very stable 5-day forecasts

Further work

- Pre-operational use during January – March 2011
- Assimilation of MARNET data
- Assimilation of CTD data for deep ocean
- Extension by biogeochemistry model (planned)

Assimilation of dynamic ocean topography from radar altimetry and GRACE/GOCE geoid (Project GEOTOP 3)

T. Janjic[1], J. Schröter[1], R. Savcenko[2], W.Bosch[2], A. Albertella[3], R. Rummel[3], O.Klatt[1]

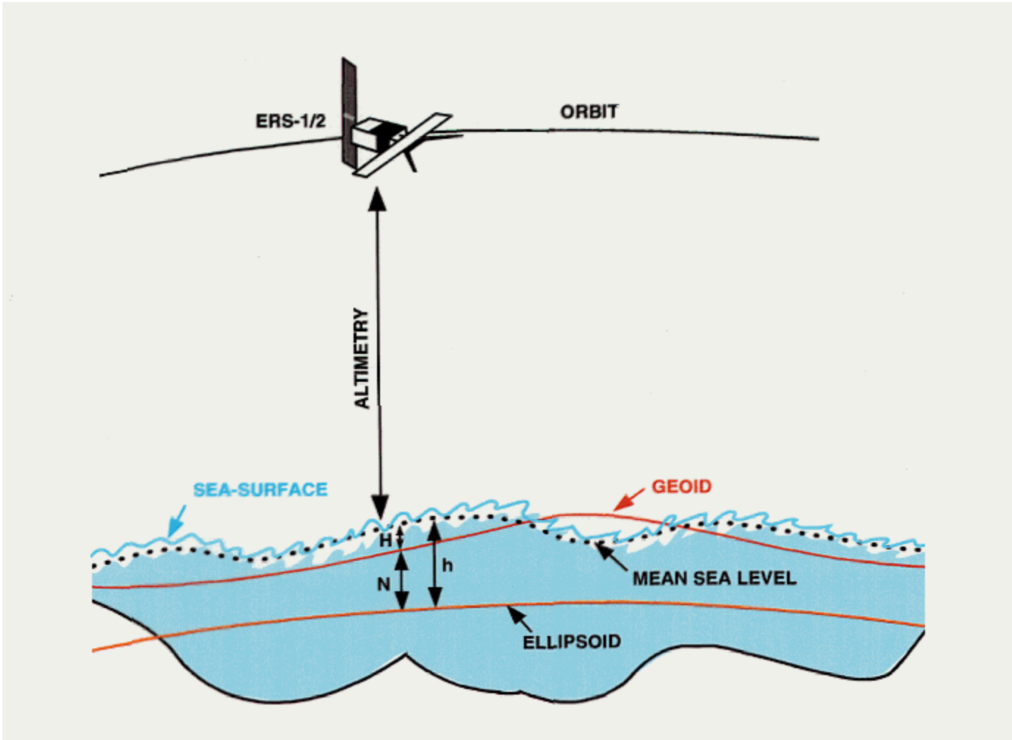
(1) Alfred Wegener Intitute
(2) German Geodetic Research Institute, Munich, Germany,
(3) Institute for Astronomical und Physical Geodesy, Munich, Germany

Geodetic DOT

DOT=H= h-N

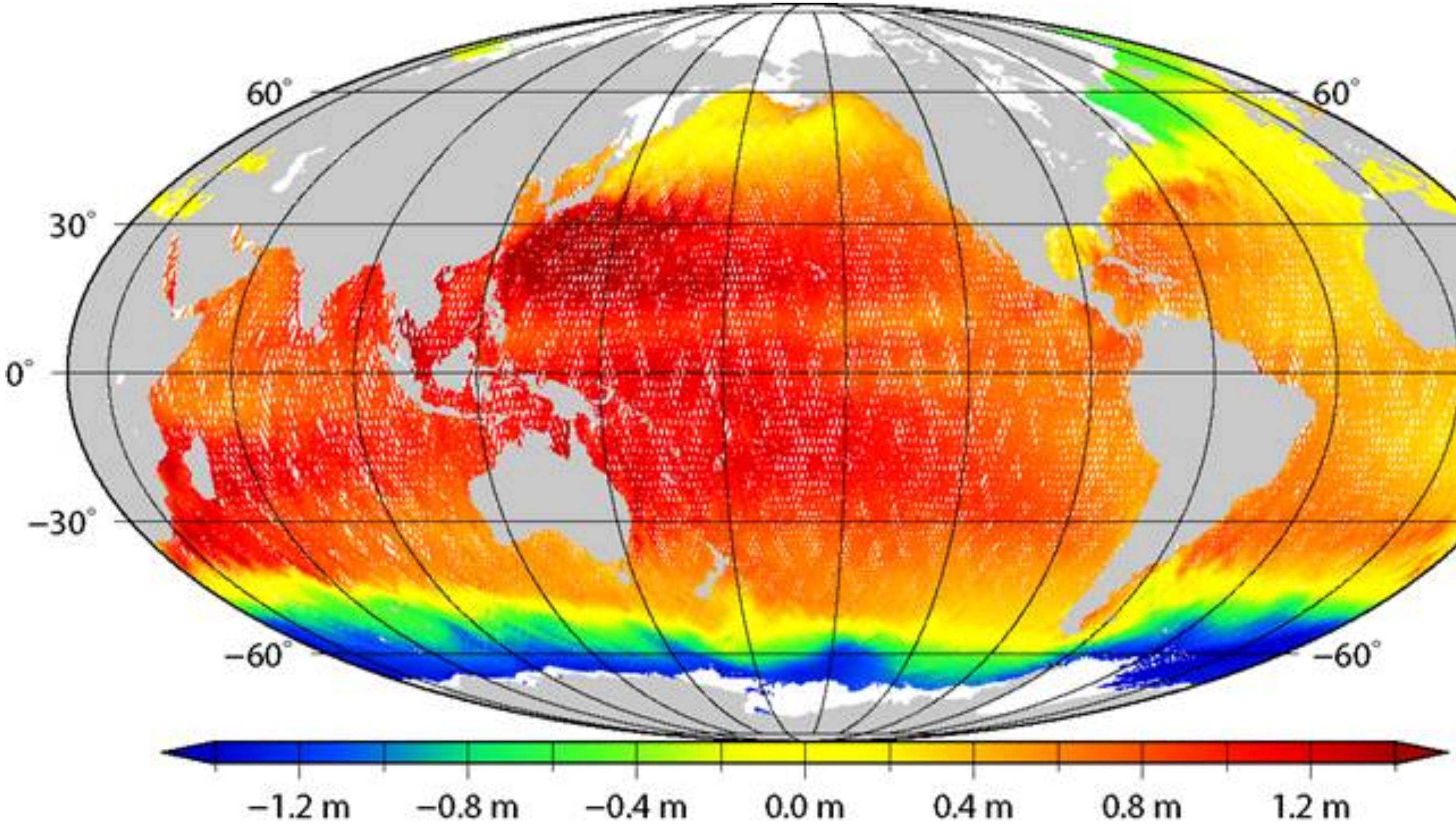
h = sea surface height (from altimetric measurements)

N = geoid height (from recent geoid models)



- The filter length is driven by the spectral resolution of the gravity
- Spectral consistency is achieved by applying a Gauss-type filter Field (Jekeli/Wahr) on sea surface and geoid.

DOT

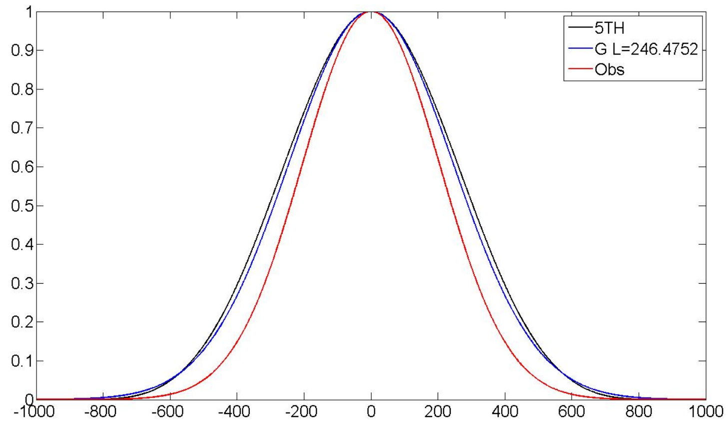


DOT from TOPEX, Jason-1, GFO, ENVISAT and GRACE/GOCE obtained from the data within a ten day interval.

Assimilation procedure

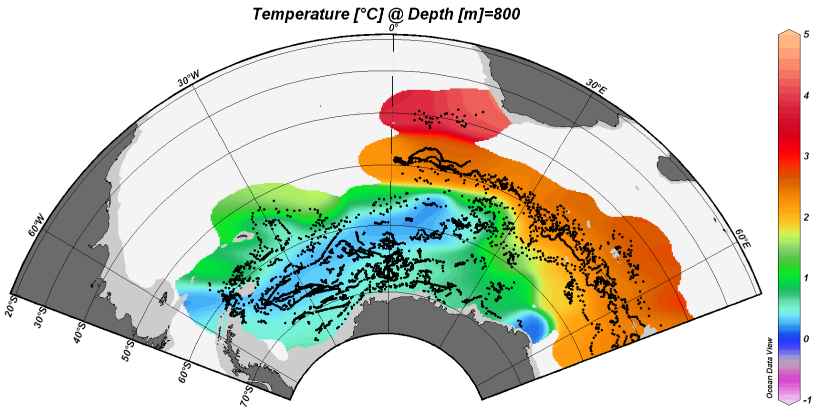
- DOTs filtered up to half width of 241 km (60), 121 km (120), and 97 km (150) are used for assimilation.
- Data are assimilated every 10 days into finite-element model FEOM.
- LSEIK with observation localization (weighting by Gauss-like correlation function)
- Observations within radius of 900 km, 450 km, 360 km are used depending on the filtering of DOT.
- The observational error standard deviation is 5 cm and 7cm for 150.

Observation weighting and filter width of observations

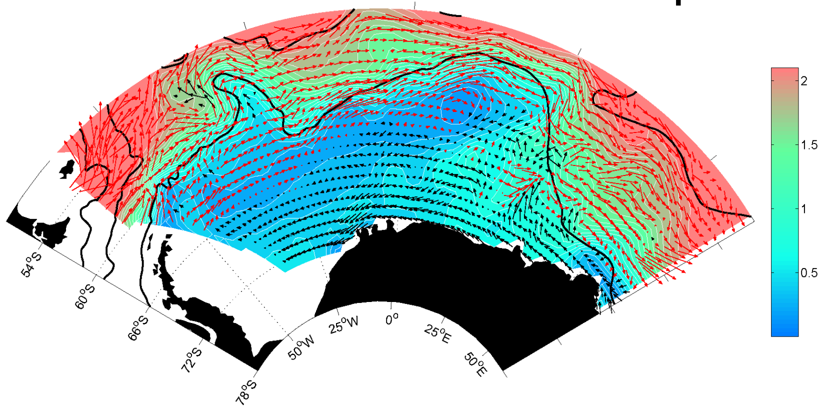


Comparisons to ARGO

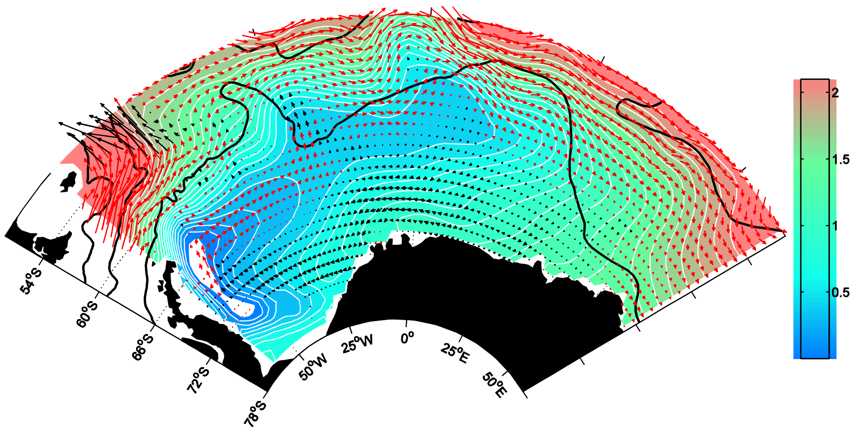
Locations of floats



ARGO composite



Model only 800 m



Temperature and Velocity at 800 m depth as result from:

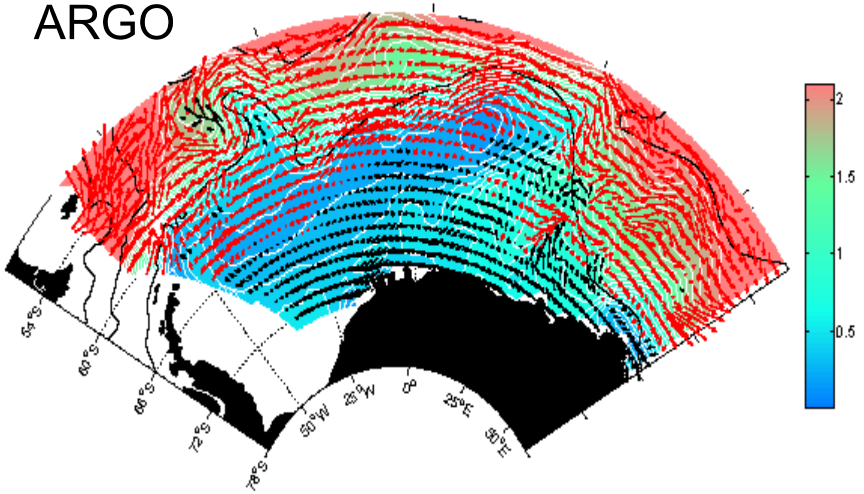
- Argo floats

and

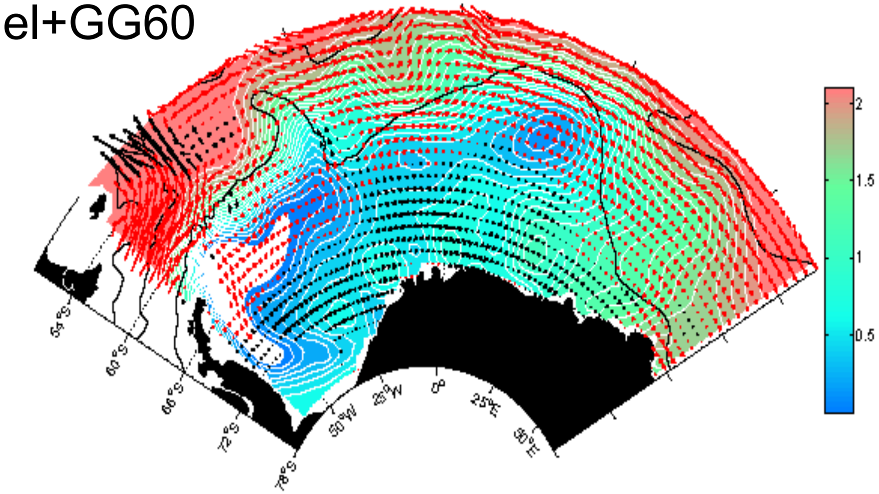
- Model without DOT data

Comparisons to ARGO

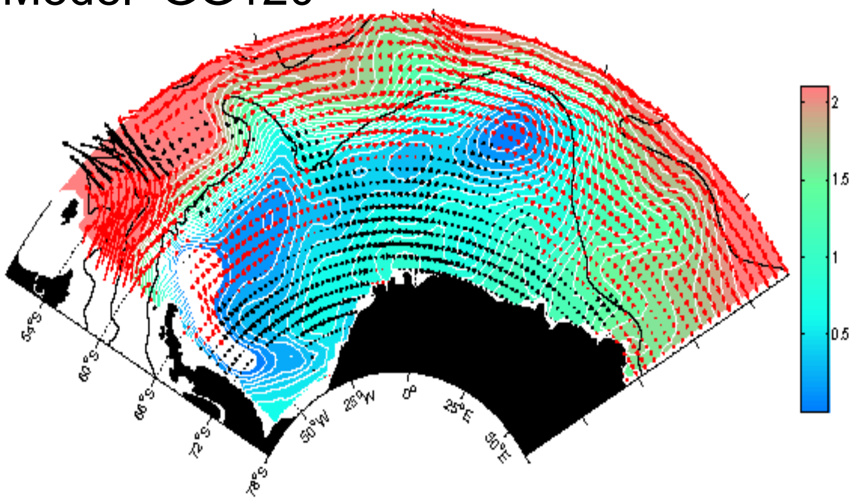
ARGO



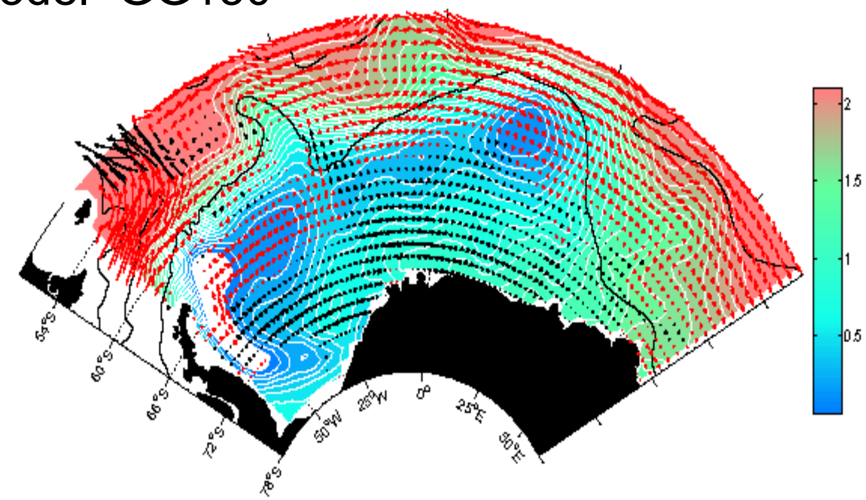
Model+GG60



Model+GG120



Model+GG150



Algorithms

some developments

Zoo of ensemble-based/error-subspace Kalman filters

- A little "zoo" (not complete):

Trying to order it …
(learn from studying relations)

EnKF(2003)

EnKF(2004)

RRSQRT

MLEF

EAKF

ROEK

EnKF(94/98)

SEEK

EnSRF

SPKF

Studied in Nerger et al. (2005)

ETKF

ESSE

SEIK

ESTKF

New filter formulation

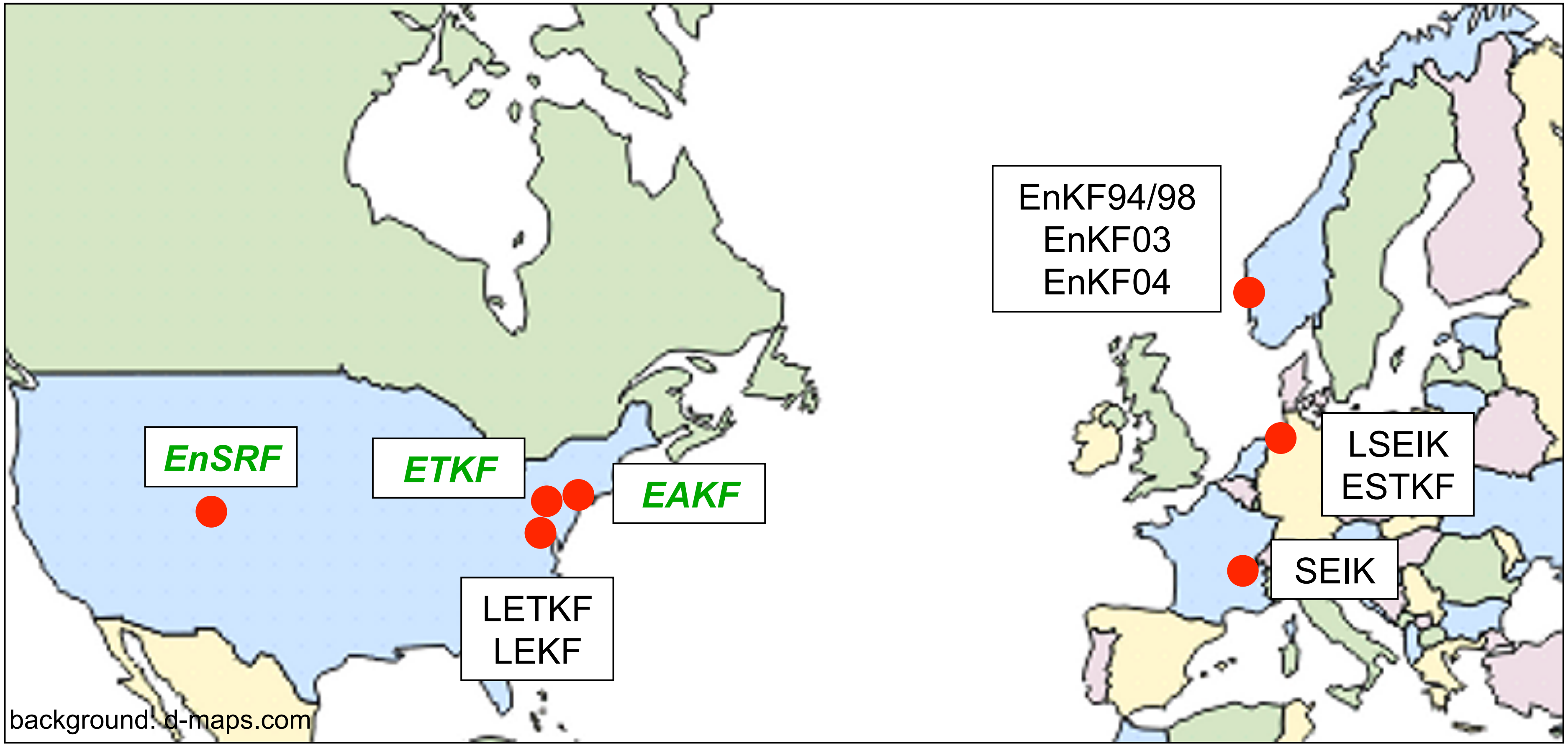
New study (submitted to MWR)

(Properties and differences are hardly understood)

Square-root filters and SEIK

Ensemble Square-root Filters

Categorization by Tippett et al. 2003



Analysis step and ensemble transformation

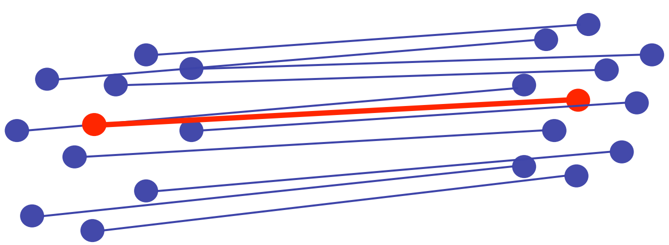
Analysis step of square-root filters:

1. correct state estimate
2. transform ensemble (forecast → analysis)

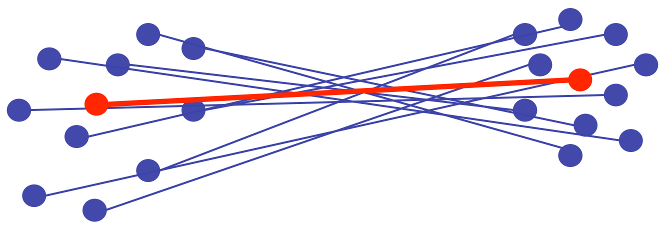
(both can be combined into a single operation)

Key element: Transformation matrix

- Computed in a space spanned by the ensemble members
- Not unique!



Minimum transformation



Random transformation with constraints

Ensemble transformations of ETKF and SEIK

ETKF

- Based on square-root of covariance matrix
- Compute transformation matrix in space of dimension **N** (ensemble size)
- Minimum transformation is standard

SEIK

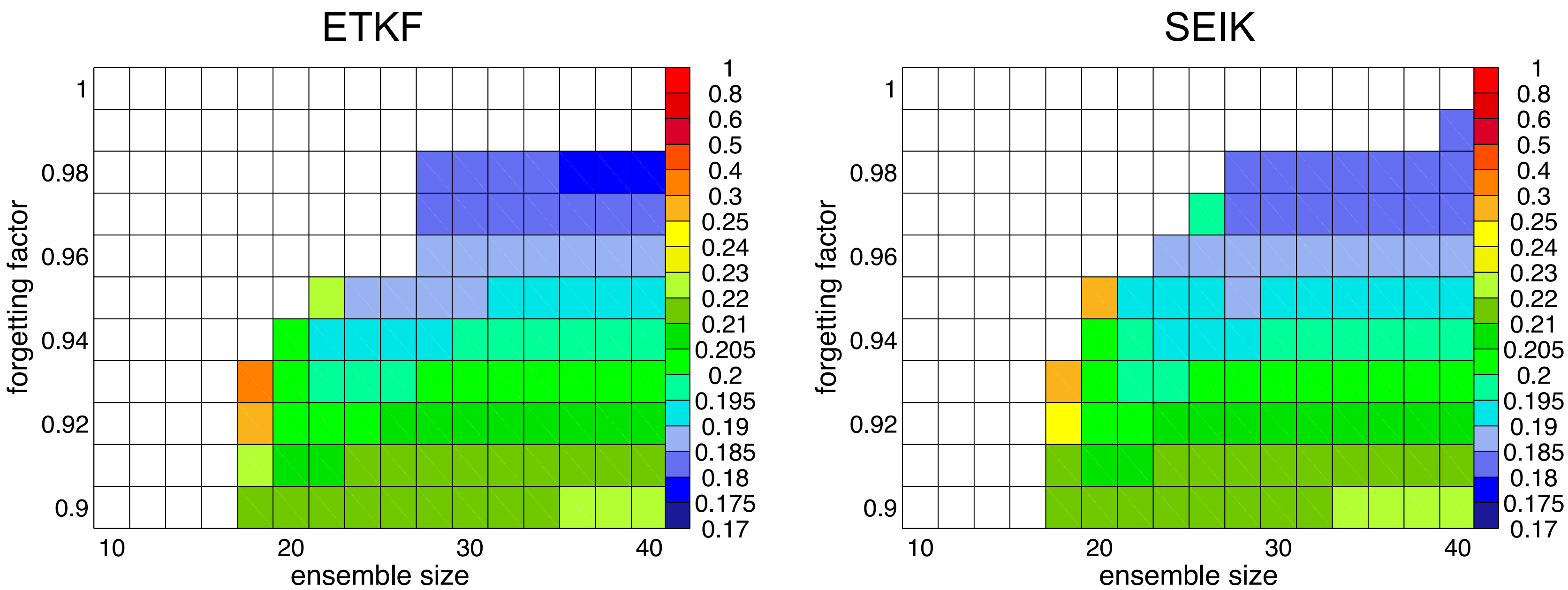
- Based on factorization $\mathbf{VUV}^T$ of covariance matrix
- Transformation matrix in space of dimension **N-1** uses square-root of matrix **U**
- Minimum transformation can be applied

➜ Ensembles after deterministic transformation nearly identical

➜ Then ETKF and SEIK are <u>almost</u> twins

L. Nerger et al., Unification of ensemble square-root filters, submitted to MWR, 2011

Lorenz96 experiment: ETKF & SEIK



- "forgetting factor" is inverse of covariance inflation (Introduced with SEIK)
- Averages over each 10 experiments (different initial ensembles)

➢ Small differences – but too large for numerical precision (relative initial difference in transformation matrices $O(10^{-4})$)

Analysis step and ensemble transformation

- ➜ But: ensemble transformation in SEIK depends on order of ensembles
- ➜ Something wrong with SEIK? ➜ Look into equations - sorry!

Forecast Covariance: $\check{\mathbf{P}}_k^f = \mathbf{L}_k \mathbf{G} \mathbf{L}_k^T$

with $\mathbf{L}_k := \mathbf{X}_k^f \mathbf{T}$ ($\mathbf{X}_k^f$: ensemble matrix)

$$\mathbf{G} := \frac{1}{N-1}\left(\mathbf{T}^T\mathbf{T}\right)^{-1}$$

$$\mathbf{T} := \begin{pmatrix} \mathbf{I}_{r\times r} \\ \mathbf{0}_{1\times r} \end{pmatrix} - \frac{1}{N}\left(\mathbf{1}_{N\times r}\right)$$

- ➜ Matrix $\mathbf{T}$ subtracts ensemble mean and removes last column
- ➜ Last column depends on ensemble ordering!

Ensemble order matters in SEIK

Distinct matrices **L** ➜ distinct matrices **U**:

$$\mathbf{U}_k^{-1} = \rho\mathbf{G}^{-1} + (\mathbf{H}_k\mathbf{L}_k)^T\mathbf{R}_k^{-1}\mathbf{H}_k\mathbf{L}_k$$

$$\check{\mathbf{P}}_k^a = \mathbf{L}_k\mathbf{U}_k\mathbf{L}_k^T$$ (this is always correct)

➜ Finally: slightly different eigenvalues and eigenvectors

Ensemble-transformation:

Square-root $$\mathbf{C}_k^{-1}(\mathbf{C}_k^{-1})^T = \mathbf{U}_k^{-1}$$ (SVD)

New ensemble: $$\mathbf{X}_k^a = \mathbf{X}_k^a + \sqrt{N-1}\,\mathbf{L}_k\mathbf{C}_k^T\mathbf{\Omega}_k^T$$

$\mathbf{\Omega}$ is projection from N-1 to N
(Random matrix from Householder reflections)

Solution:

Redefine **T:**

- Subtract ensemble mean
- Distribute last column over first N-1 columns
- Use correct scaling to preserve mean

$$\mathbf{T}_{i,j} = \begin{cases} 1 - \frac{1}{N}\frac{1}{\frac{1}{\sqrt{N}}+1} & \text{for } i = j, i < N \\ -\frac{1}{N}\frac{1}{\frac{1}{\sqrt{N}}+1} & \text{for } i \neq j, i < N \\ -\frac{1}{\sqrt{N}} & \text{for } i = N \end{cases}$$

➜ A deterministic form of Ω (Householder reflection)

With this:

$$\mathbf{G} := \frac{1}{N-1}\mathbf{I}$$

New filter - ESTKF

Use redefined **T** (= deterministic $\mathbf{\Omega}$)

Forecast Covariance: $\check{\mathbf{P}}_k^f = \mathbf{L}_k \mathbf{G} \mathbf{L}_k^T$

With $\mathbf{L}_k := \mathbf{X}_k^f \mathbf{\Omega}$

Matrix **U** simplifies to:

$$\mathbf{U}_k^{-1} = \rho(N-1)\mathbf{I} + (\mathbf{H}_k \mathbf{L}_k)^T \mathbf{R}_k^{-1} \mathbf{H}_k \mathbf{L}_k$$

(inverse of error covariance matrix in error space)

Ensemble transformation

$$\mathbf{X}_k^a = \overline{\mathbf{X}}_k^a + \sqrt{N-1}\, \mathbf{X}_k^f \boxed{\mathbf{\Omega} \mathbf{C}_k^T \tilde{\mathbf{\Omega}}^T}$$

- ➜ Consistent projections between state space and error space
- ➜ Transformation identical to ETKF (same eigenvalues/vectors)
- ➜ Cheaper than ETKF
- ➜ Not more expensive than SEIK

Regulated Localization

Localization Types

Covariance localization

- Applied to forecast covariance matrix
- Element-wise product with matrix of compact support
- Only possible if forecast covariance matrix is computed (not in ETKF or SEIK)

Domain localization

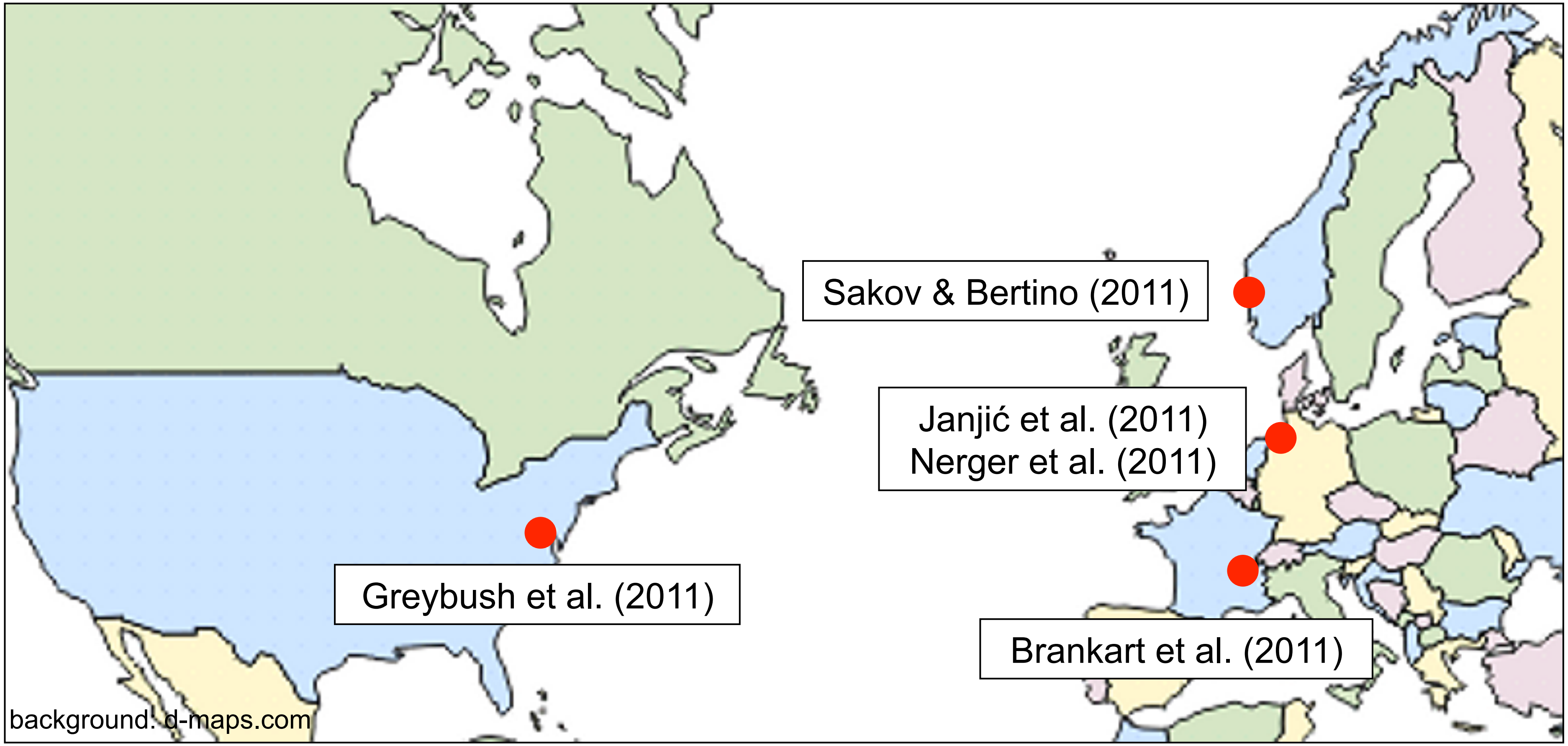
- Perform analysis in loop over domains in model grid
- Use only observations within specified influence distance
- Can be combined with weighting of observation errors („observation localization“)
- Possible in all filter formulations

E.g.: Houtekamer & Mitchell (1998, 2001), Whitaker & Hamill (2002)

E.g.: Evensen (2003), Ott et al. (2004), Nerger et al. (2006), Hunt et al. (2007)

Covariance vs. Observation Localization

Recently a *hot* topic …



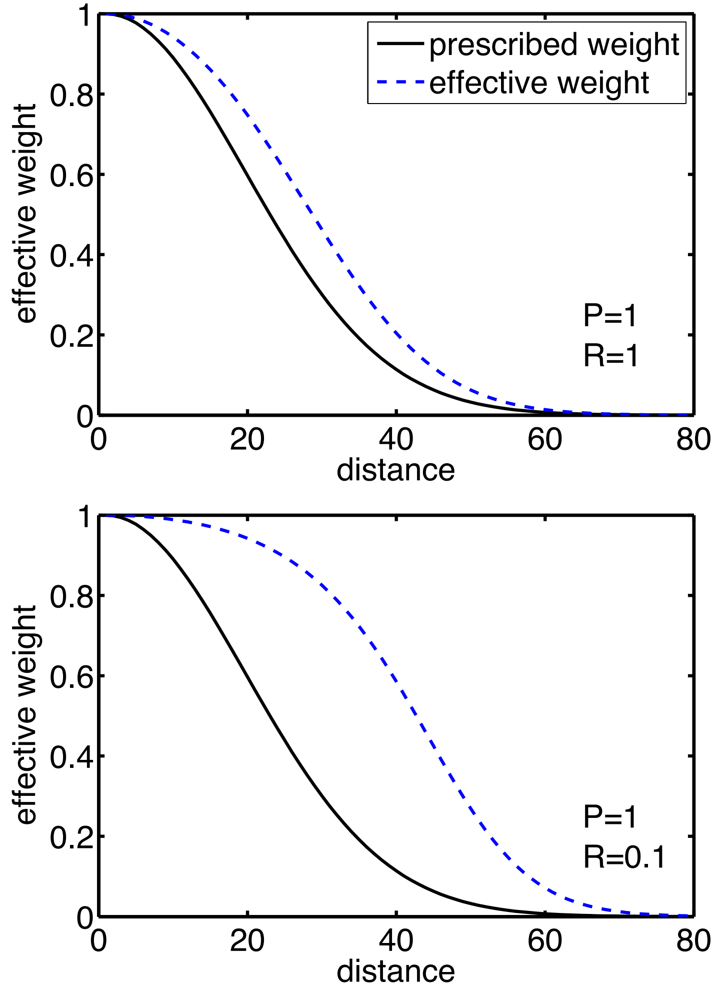
Covariance vs. Observation Localization

Some published findings:

- Both methods are "similar"
- Slightly smaller width required for observation localization

But note for observation localization:

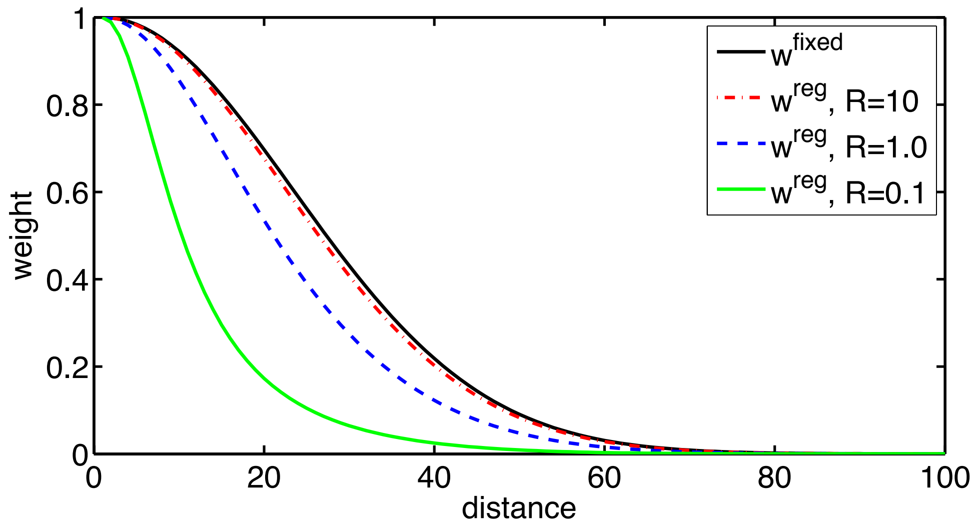
- Effective localization width depends on errors of state and observations
  - ➢ Small observation error ➜ wide localization
  - ➢ Possibly problematic:
    - in initial transient phase of assimilation
    - if large state errors are estimated locally



P: state error variance
R: observation error variance

Regulated Localization

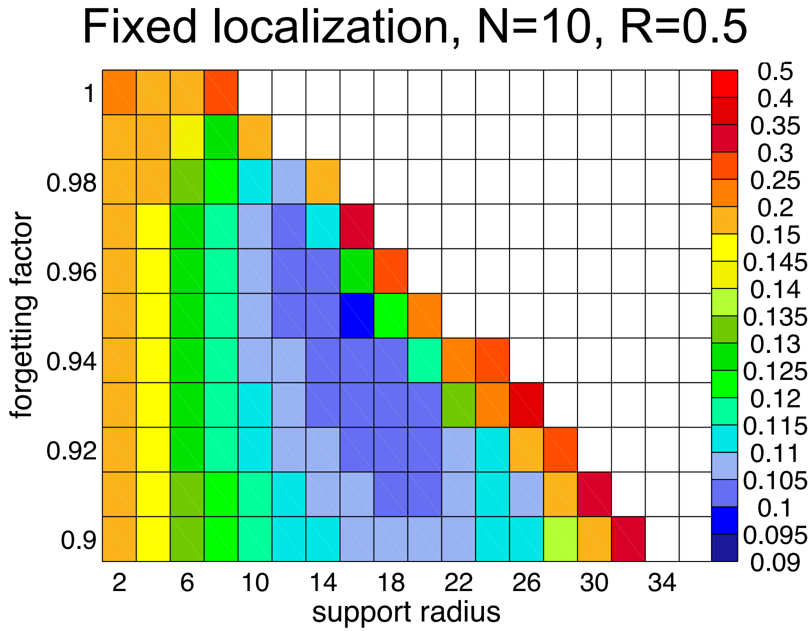
➜ New localization function
- formulated to keep effective width constant
- depending on state and observation errors
- depending on fixed localization function
- easy to compute to each observation

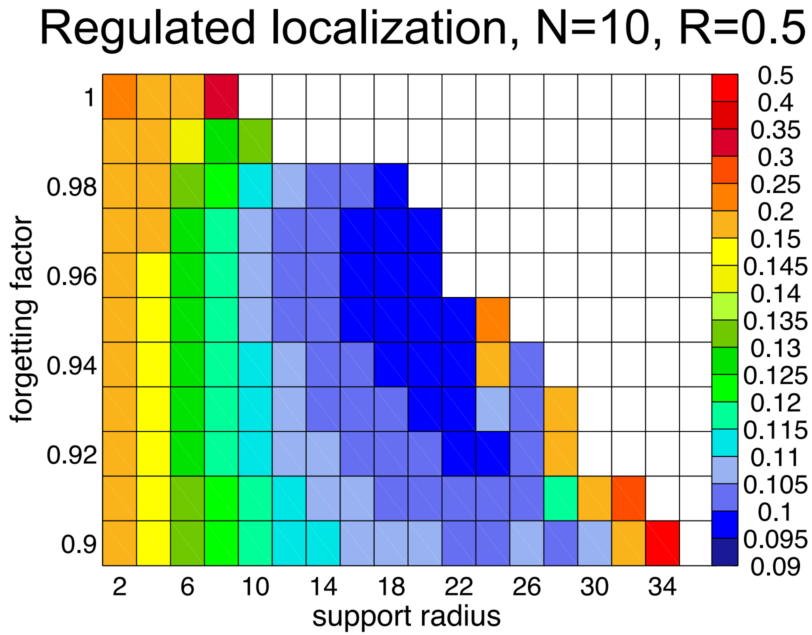


P: state error variance
R: observation error variance

L. Nerger et al. A regulated localization scheme for ensemble-based Kalman filters. QJ Roy. Met. Soc., early online access, 2011

Lorenz96 Experiment: Regulated Localization





- Reduced minimum rms errors
- Increased stability region
- Particularly pronounced for accurate observations

Thank you!