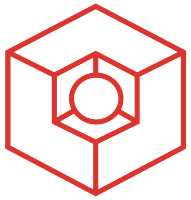
Data Assimilation –
Practical Aspects and Case Studies

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

KIAPS, May 30, 2013

Data Assimilation

Problem: Estimate model state (trajectory) from

- guess at initial time
- model dynamics
- observational data

Characteristics of system:

- approximated by discretized differential equations
- high-dimension - $\mathcal{O}(10^7\text{-}10^9)$
- sparse observations
- non-linear

Current “standard” methods:

- Optimization algorithms (“4DVar”)
- Ensemble-based estimation algorithms

This talk!

AWI

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

Overview

How do we apply the Ensemble Kalman filters?

- Assimilation software
- Application aspects
  - Localization
  - Covariance inflation
  - Observation errors
  - Model errors
  - Validation data
- Case studies

Assimilation Software

PDAF: A tool for data assimilation

PDAF Parallel Data Assimilation Framework

Discuss software aspects based on

PDAF - Parallel Data Assimilation Framework

- an environment for ensemble assimilation
- a software to provide assimilation methods
- 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

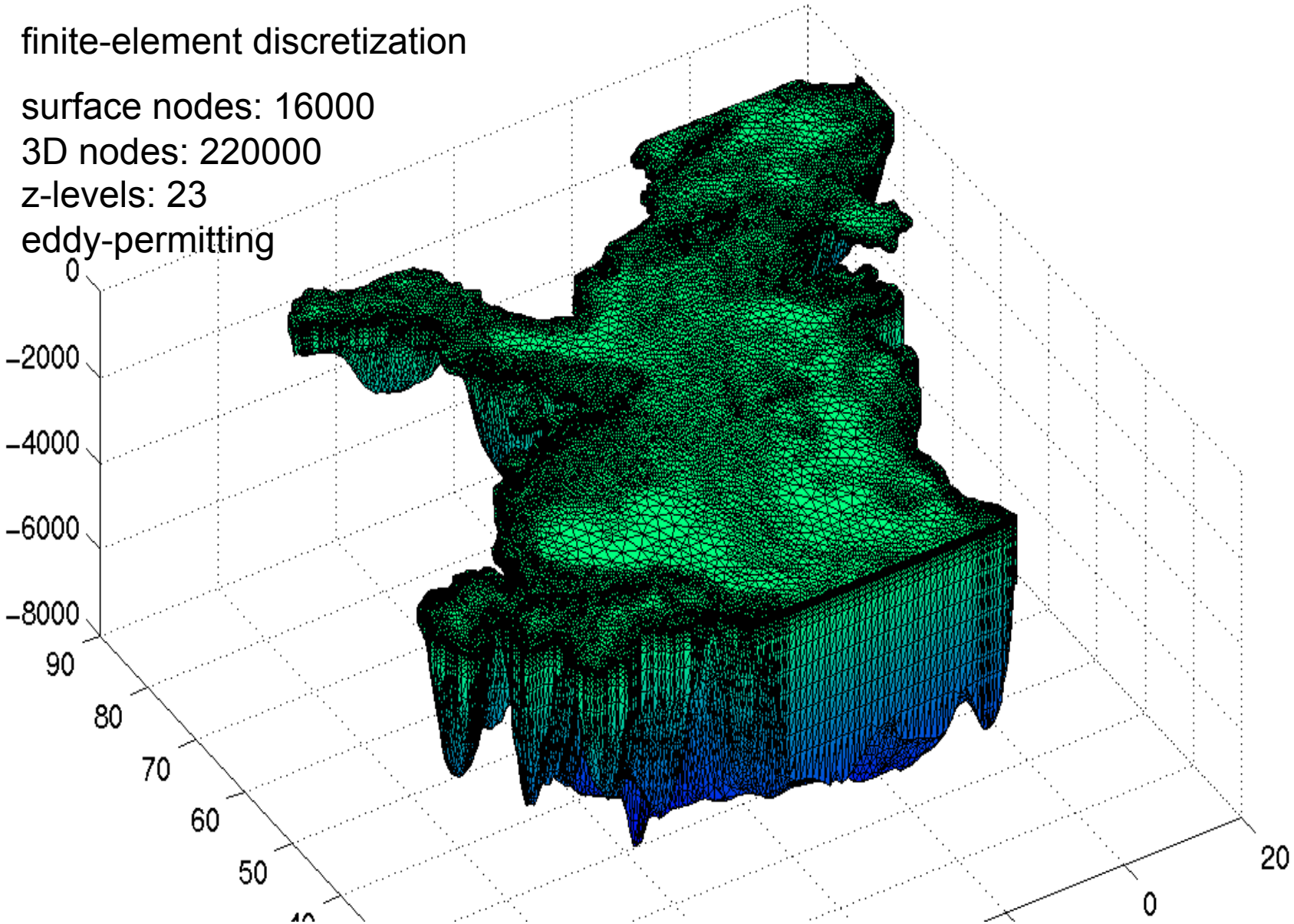
AWI

PDAF‘s “home model”

FEOM – Coarse mesh for North Atlantic

finite-element discretization

surface nodes: 16000
3D nodes: 220000
z-levels: 23
eddy-permitting



FEOM / FESOM

Finite Element Sea-ice Ocean circulation Model

- developed at AWI (Danilov et al. 2004)
- primitive equations
- horizontally unstructured meshes with varying resolution

Relevant for ensemble assimilation

- single grid point index (no direct location information)
- parallel grid decomposition through partitioning program (Metis) – irregular compact regions
- very different from regular grid models

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

Framework design

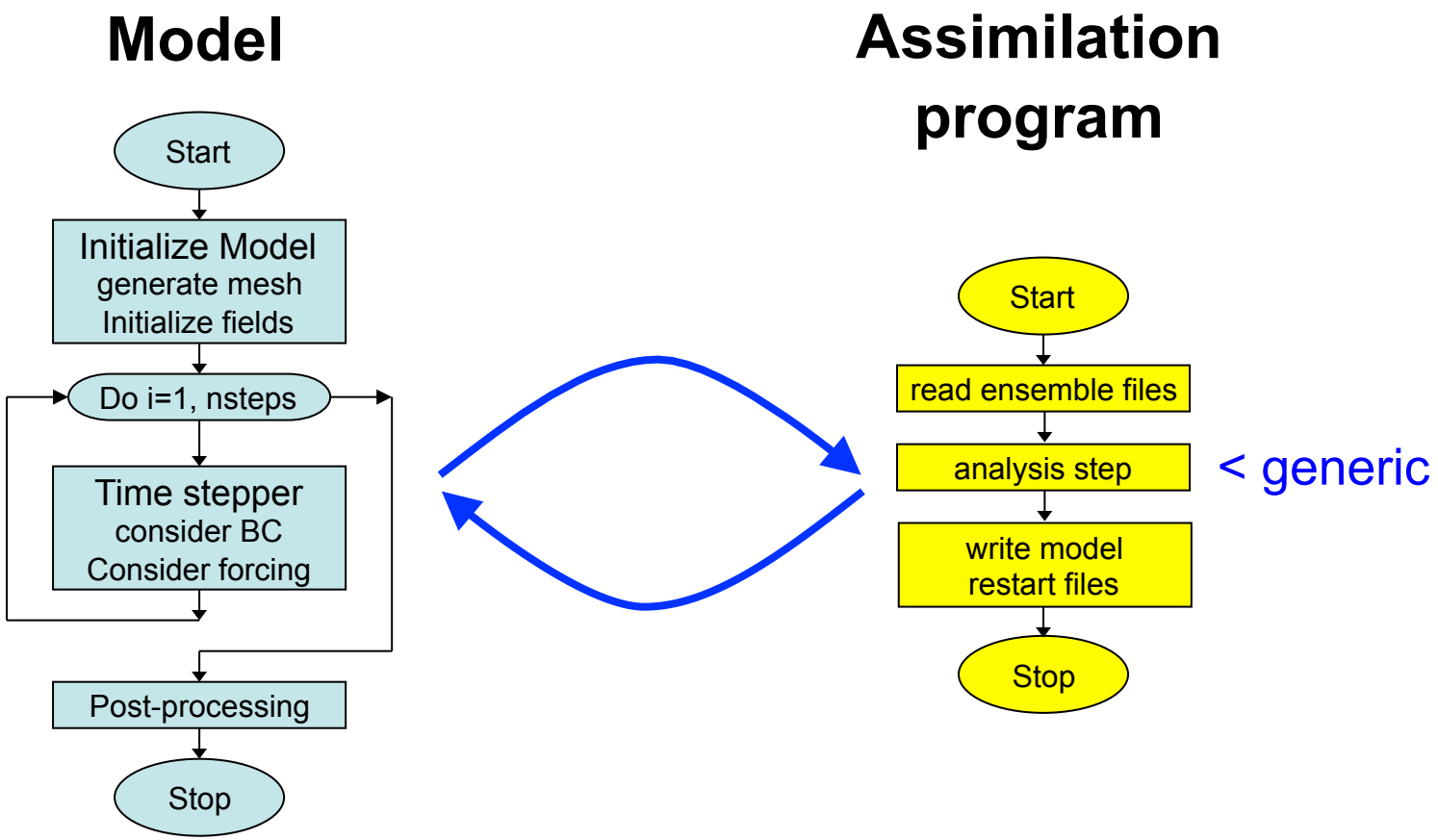
- Parallelization of ensemble forecast can be implemented independently from model
- Analysis step can be implemented independently from model (run it providing state vector and observational information)

Goals for a model-independent framework

- Simplify implementation of data assimilation systems based on existing models
- Provide parallelization support for ensemble forecasts
- Provide filter algorithms (fully implemented & parallelized)
- Provide collection of „fixes“ for filters, which showed good performance in studies

AWI

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

Online and Offline modes

Offline

- Separate executable programs for model and filter
- Ensemble forecast by running sequence of models
- Analysis by assimilation program
- Data exchange model-filter by files on disk

- *Advantage:*
  Rather easy implementation
  (file reading/writing routines, no change to model code)
- *Disadvantage:*
  Limited efficiency, cost of file reading & writing; restarting programs

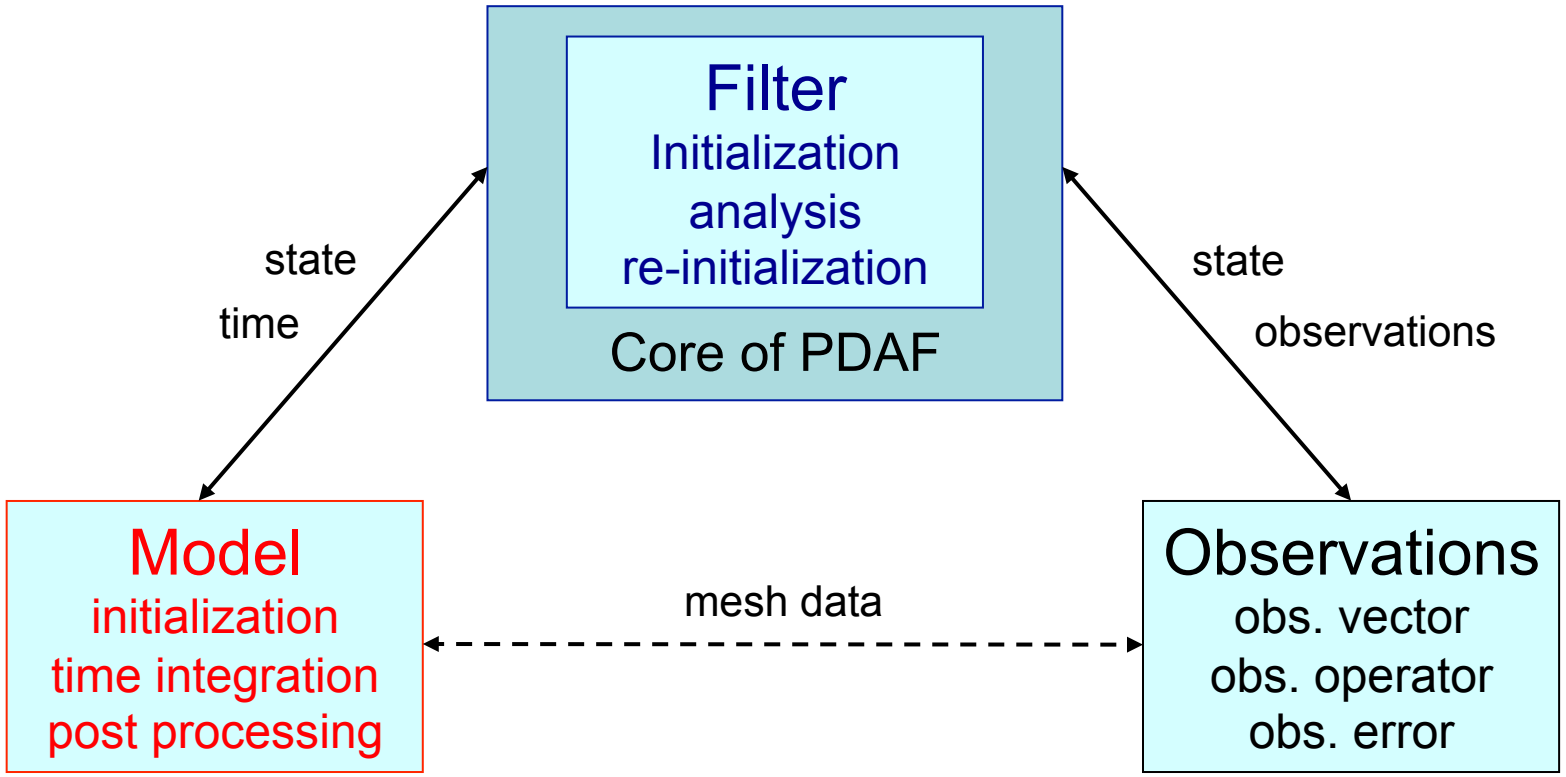
Online and Offline modes

Online

- Couple model and filter into single executable program
- Run single program for whole assimilation task (forecasts and analysis)

- *Advantage:*
  Computationally very efficient
  (less file outputs, no full program restarts)
- *Disadvantage:*
  More implementation work, incl. extension of model code.

Logical separation of assimilation system



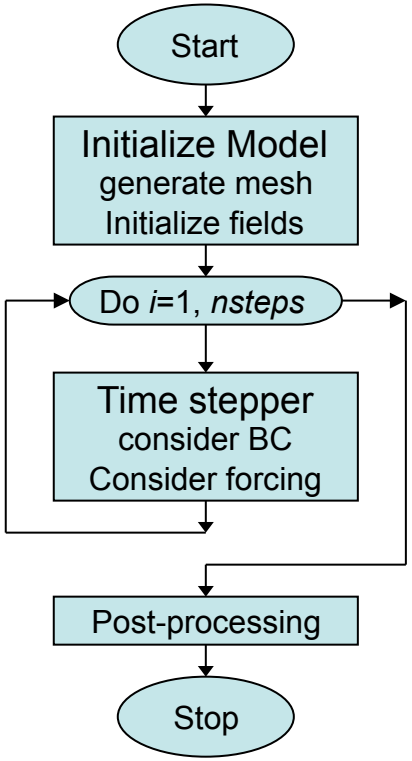
For online implementation:

⟷ Explicit interface

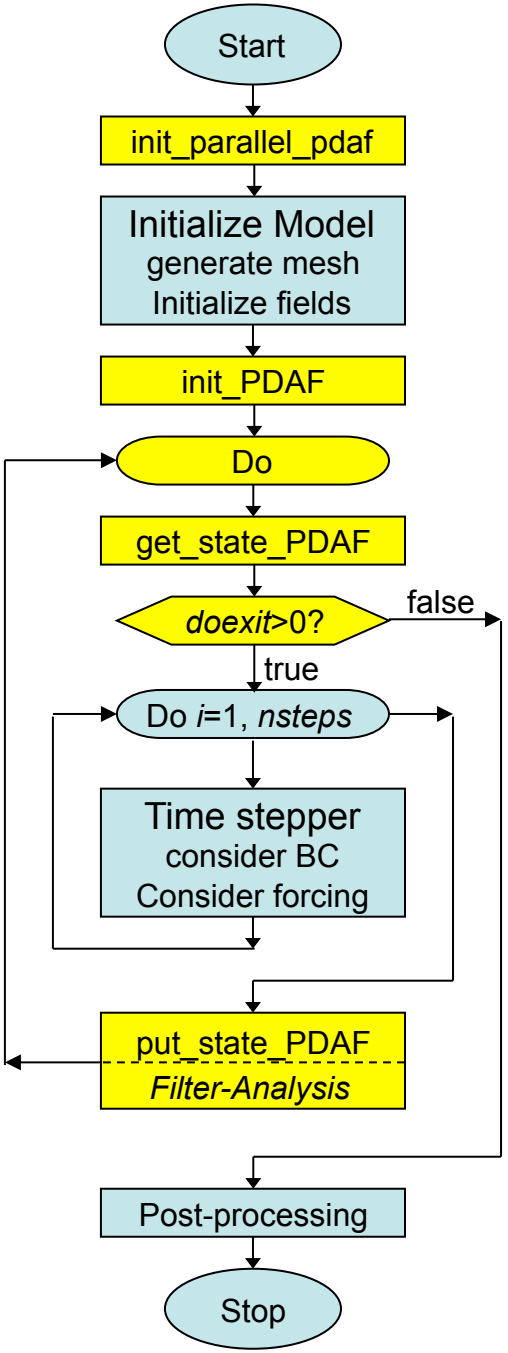
⟷ Indirect exchange (Fortran: module/common)

Nerger, L., Hiller, W. (2013). Software for Ensemble-based DA Systems – Implementation and Scalability. Computers and Geosciences. 55, 110-118

Model



External Do-loop can be avoided – lower flexibility!



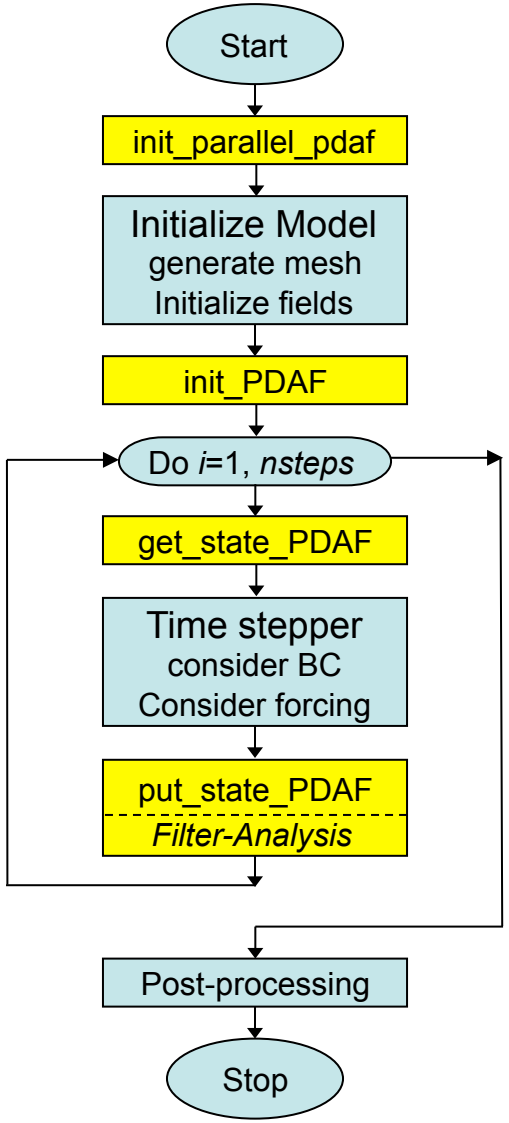
Extension for data assimilation

Initialization

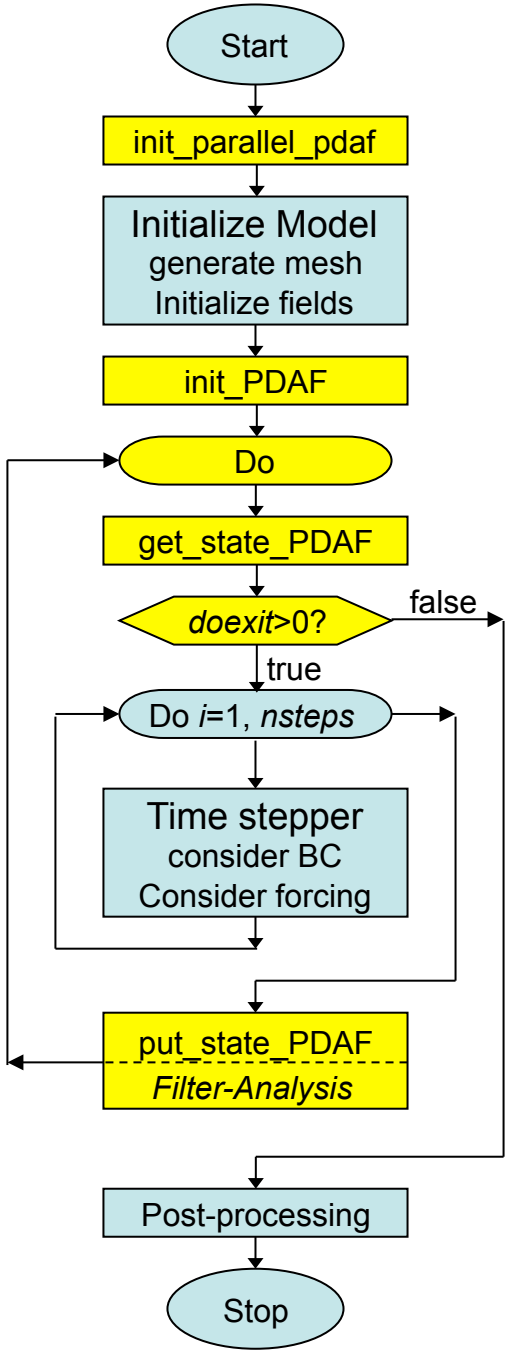
Ensemble forecast

Analysis step

Without external loop



Fully flexible



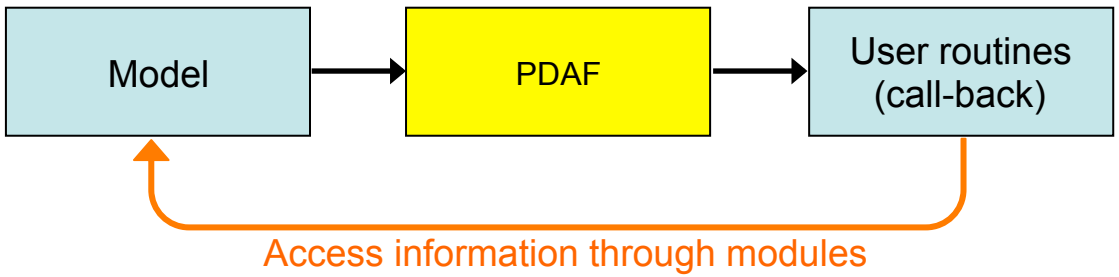
PDAF: Considerations for Implementation

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

PDAF interface structure

PDAF Parallel Data Assimilation Framework

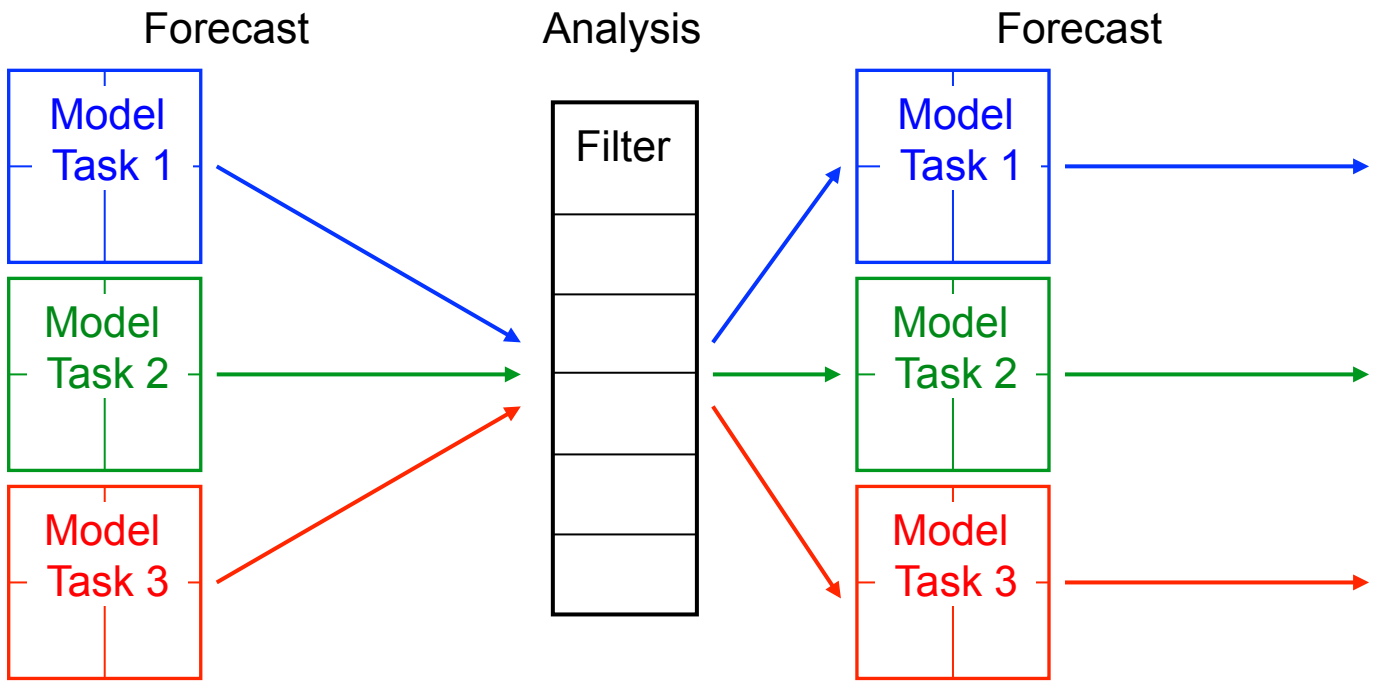
- Interface independent of filter
  (except for names of user-supplied subroutines)
- User-supplied call-back routines for elementary operations:
  - field transformations between model and filter
  - observation-related operations
  - filter pre/post-step
- User supplied routines can be implemented
  as routines of the model
  (e.g. share common blocks or modules)



AWI

2-level Parallelism

PDAF Parallel Data Assimilation Framework



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

➢ Analysis step is also parallelized

MPI communicators initialized in routine *init_parallel_pdaf*

AWI

Communicators

Communicators define a group of processes for data exchange

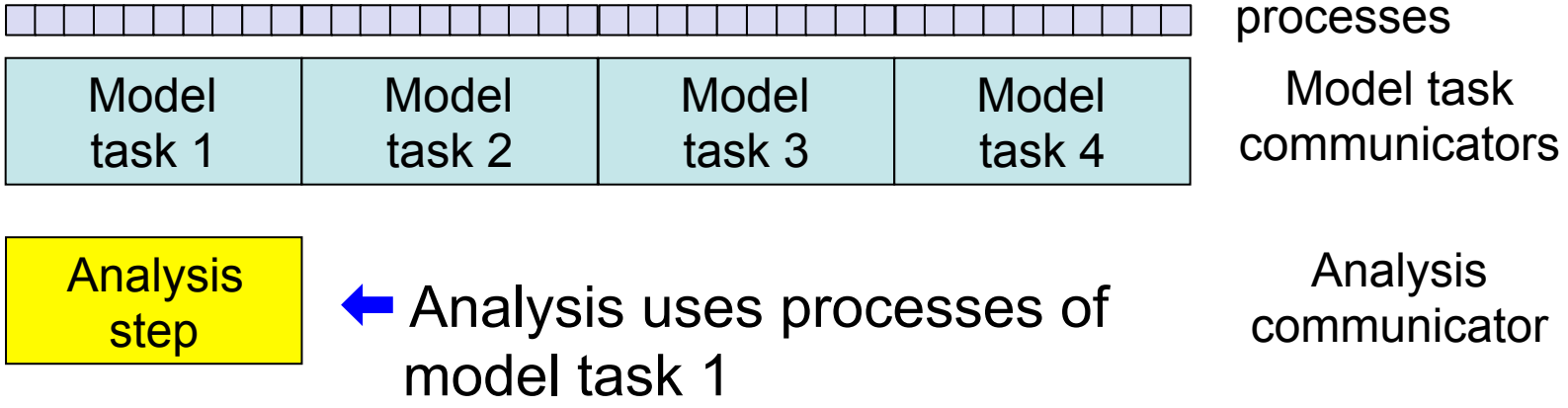
3 communicator sets are required:

1. Model communicators (one set for each model task)
2. Filter communicator (a single set of processes)
3. Coupling communicators
   – to send data between model and filter
   (one set for each filter process and connected model processes)

Configuring the parallelization

- Assume *4* ensemble members
- Model itself is parallelized (like domain decomposition)
- Configuration of “MPI communicators” (groups of processes)

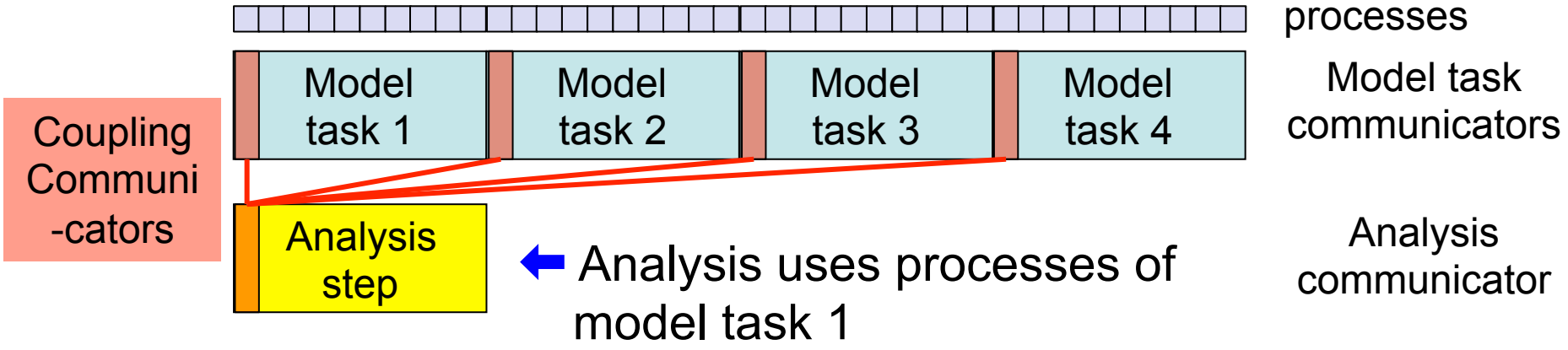
Variant 1:



Configuring the parallelization

- Assume *4* ensemble members
- Model itself is parallelized (like domain decomposition)
- Configuration of “MPI communicators” (groups of processes)

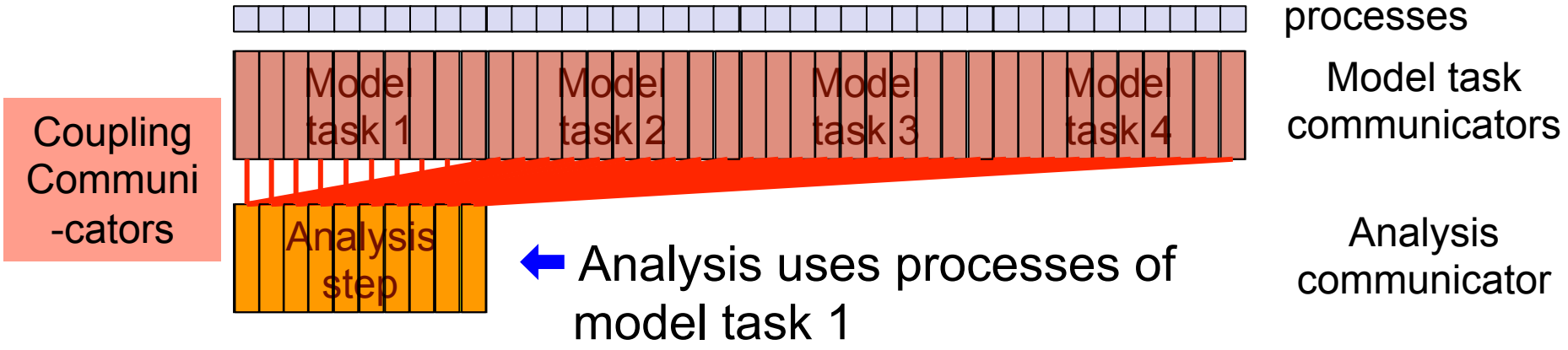
Variant 1:



Configuring the parallelization

- Assume *4* ensemble members
- Model itself is parallelized (like domain decomposition)
- Configuration of “MPI communicators” (groups of processes)

Variant 1:

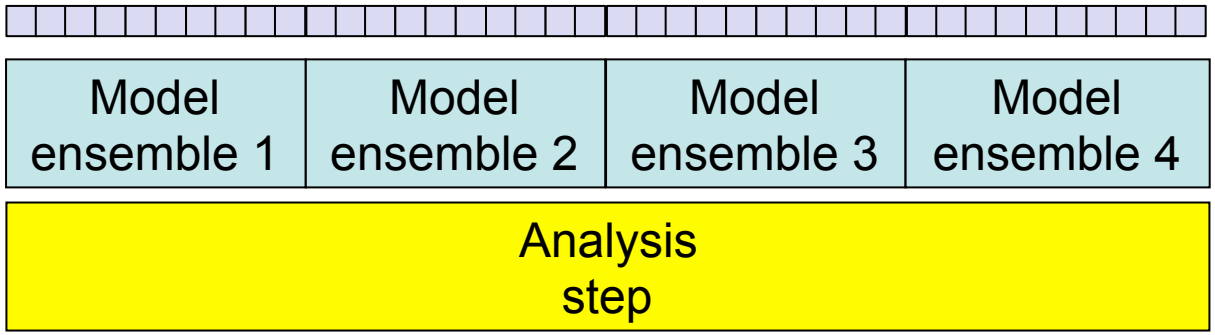


- Default communication variant of PDAF
- *init_parallel_pdaf* provides this configuration
- Reasoning: Convenience to use same domain decomposition for model and analysis (also efficient for ocean with satellite data)

Alternative Configurations

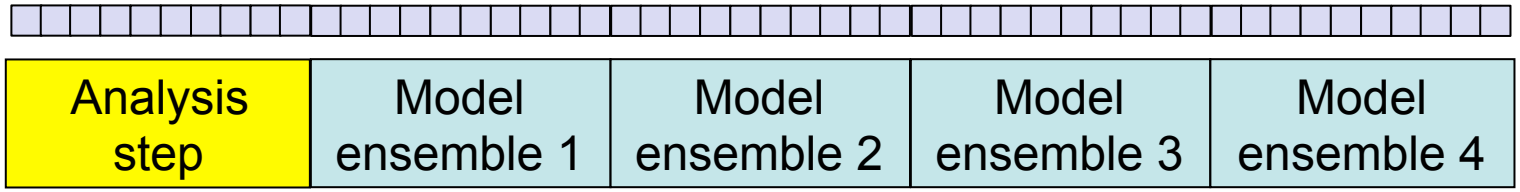
Issue: Configuration of coupling communicators is more complicated

Variant 2:



If you worry about idle processes

Variant 3:



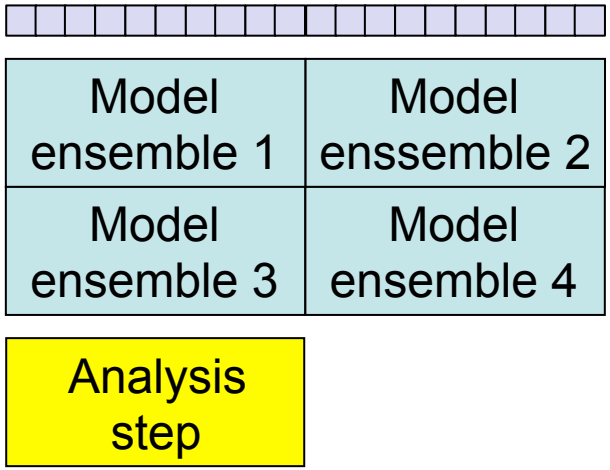
Separate processes

When memory is really limited
Analysis processes might idle during forecast

Alternative Configurations

Issue: Configuration of coupling communicators is more complicated

Variant 4:

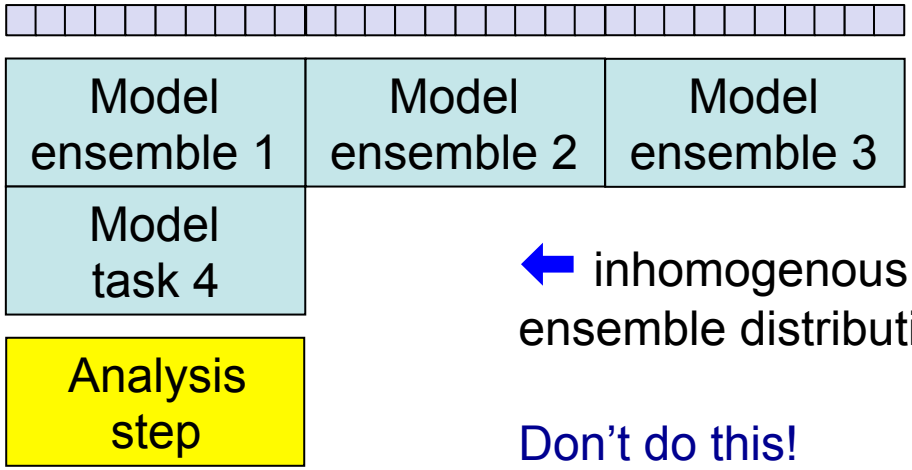


processes

⬅ less model tasks than ensemble members

Needs fully flexible implementation!

Variant 5:



⬅ inhomogenous ensemble distribution

Don’t do this!

Initialization of Assimilation

Init_PDAF

Set parameters, for example

- select filter
- set ensemble size

Call initialization routine of framework (PDAF_init)

- provide parameters according to interface
- provide MPI communicators
- provide name of routine for ensemble initialization

Ensemble initialization routine – called by PDAF_init

- a “call-back routine”
- defined interface: provides ensemble array for initialization
- user-defined initialization

Ensemble Forecast

get_state (PDAF_put_state)

get_state_PDAF

- the control routine for ensemble forecast
- set start time and number of time steps for forecasting an ensemble member (call-back routine)

put_state_PDAF

- initialize model fields from state vector (call-back routine)

Model integrates state

put_state (PDAF_put_state)

- write forecast fields into state vector (call-back routine)
- prepare to integrate next ensemble state

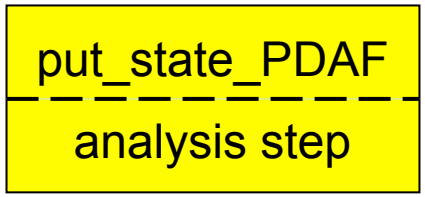
Jump back to *get_state* if more ensemble members need integration

AWI

Compute analysis step

put_state (PDAF_put_state)

- Checks if ensemble forecast is complete



If ensemble forecast is complete:

- Analysis step (filter) routine is called in *put_state*

Analysis step needs call-back routines

- Names are specified in call to PDAF_put_state
- Operations like
  - Apply observation operator to state vector
  - Initialize observation vector
  - Perform localization of state vector or observation

Current algorithms in PDAF

PDAF Parallel Data Assimilation Framework

PDAF originated from comparison studies of different filters

Filters

- Ensemble Kalman filter (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)

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

Smoothers for

- ETKF/LETKF
- ESTKF/LESTKF
- EnKF

AWI

Parallel Performance – DA system

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

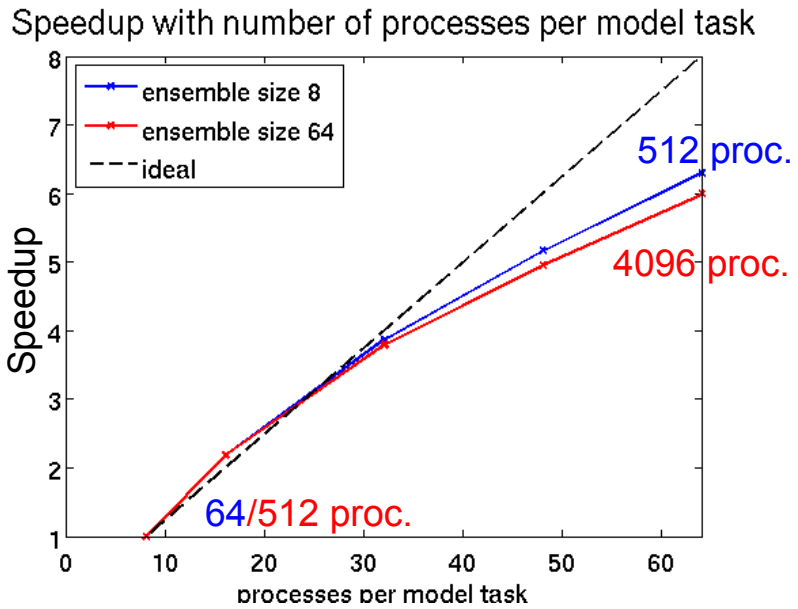
94-99% of computing time in model integrations

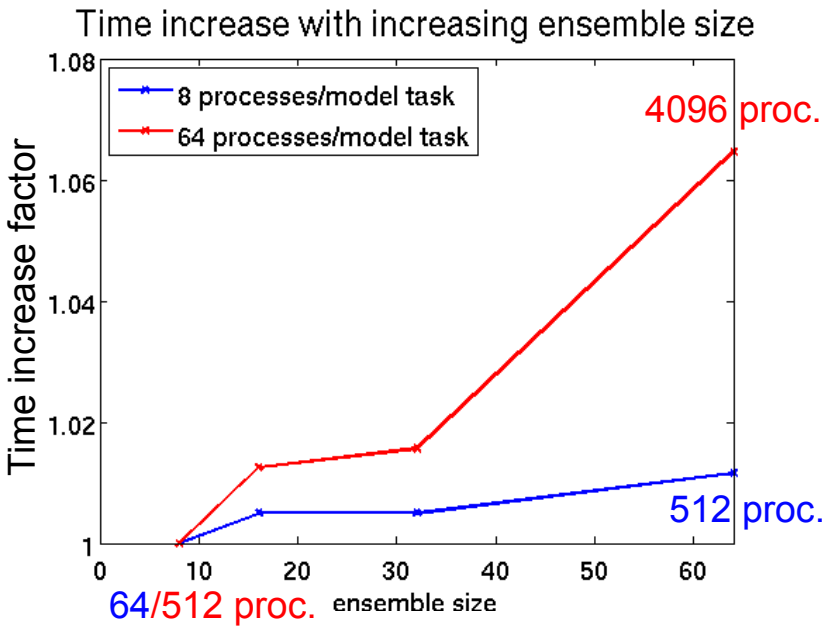
Speedup: Increase number of processes for each model task, fixed ensemble size

- 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

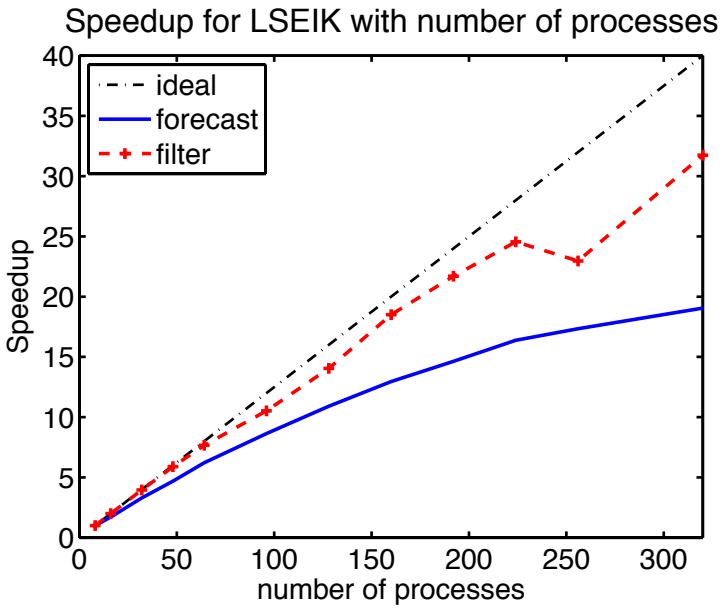
- Use between 8 and 320 processors; larger mesh (55.000 surface nodes)
- Assimilate each time step with LSEIK
- Up to 50% of computing time in filter analysis

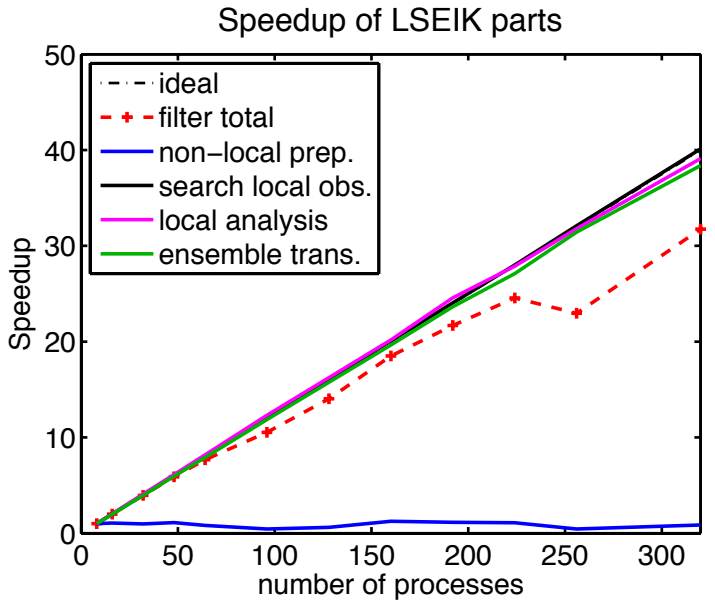
Filter in total:

- 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





Particularities of PDAF

PDAF Parallel Data Assimilation Framework

There are several frameworks or test beds for data assimilation

PDAF is particular in some ways:

- Typically create a single program: model + filter
- Extend model code to obtain assimilation system
- Minimal changes to the model code (>=4 subroutine calls)
- Model integration not needed to be subroutine
- Control of assimilation program by user-written routines
- Run assimilation like model with additional options

Open source: Code and documentation available at

http://pdaf.awi.de

AWI

Aspects of Framework-building

PDAF Parallel Data Assimilation Framework

PDAF is designed to work easily with existing models

- minimal changes code changes
- model time step not needed to be subroutine
- observation routines hidden from model (call-back functions)

These points should not interfere with compute performance!

If you are designing a new model for ensemble data assimilation

- model time step as a subroutine gives clean code
- Tighter integration of model and filter possible (perhaps even pointers to reduce memory)
- Initializing observations before calling analysis update is more direct

AWI

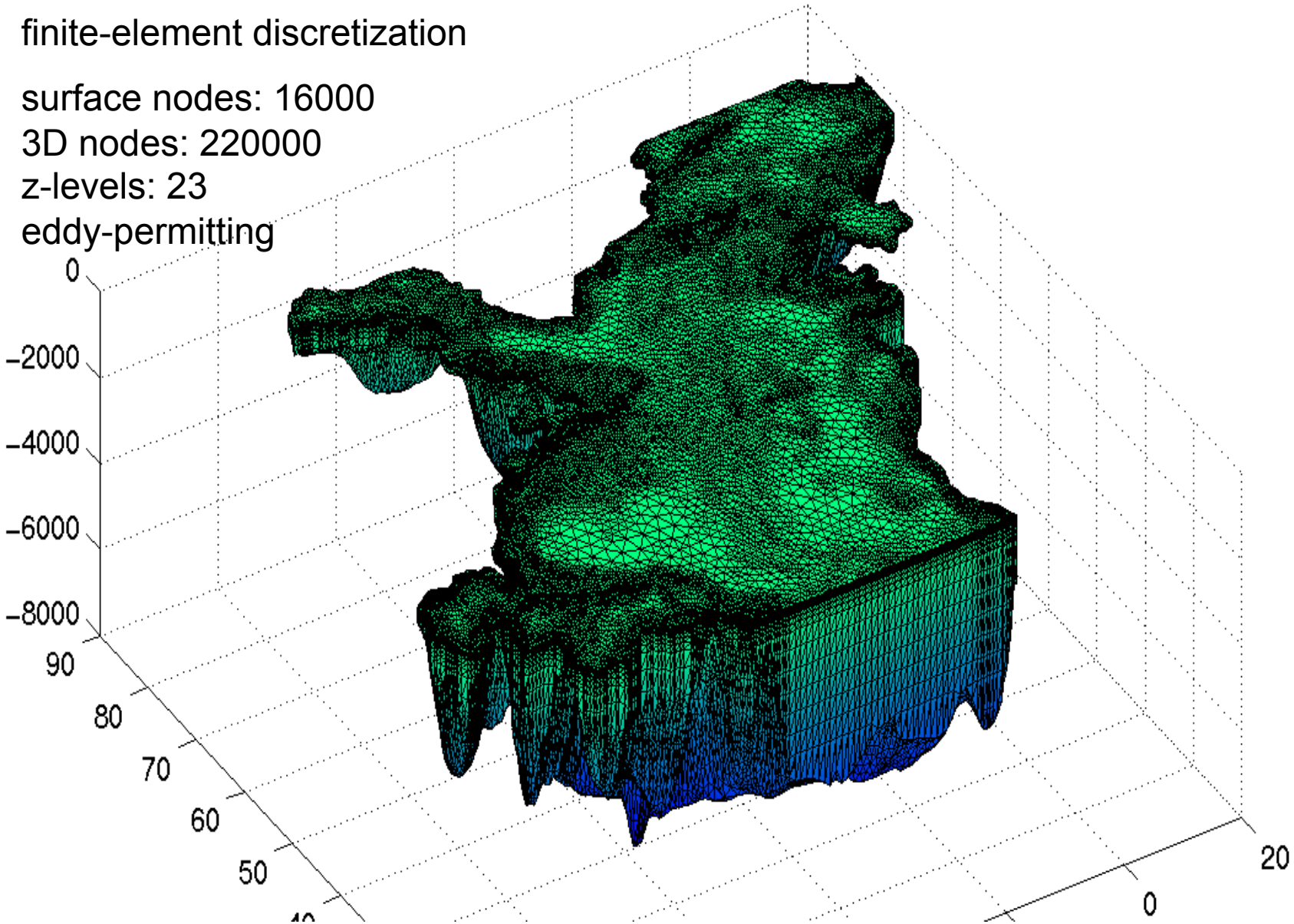
Case Study 1:

Assimilation of pseudo sea surface height observations in the North Atlantic (twin experiment)

FEOM – Coarse mesh for North Atlantic

finite-element discretization

surface nodes: 16000
3D nodes: 220000
z-levels: 23
eddy-permitting



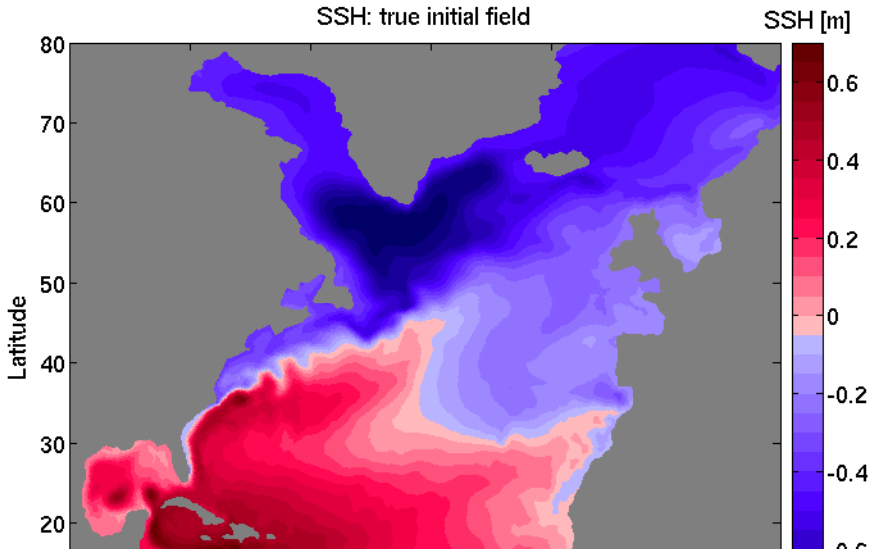
Configuration of twin experiments

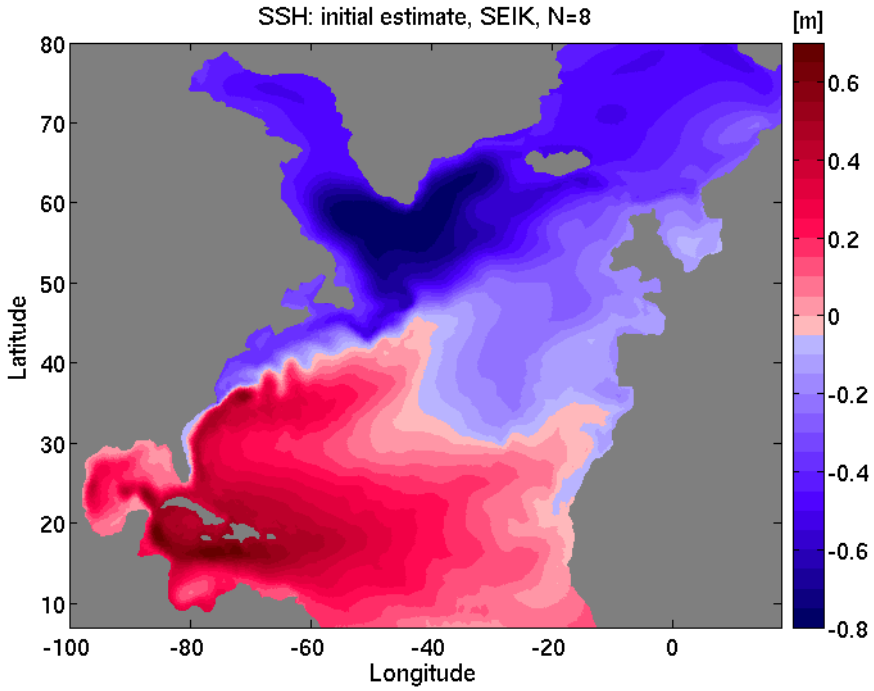
- Generate true state trajectory for 12/1992 - 3/1993
- Assimilate synthetic observations of sea surface height (generated by adding uncorrelated Gaussian noise with std. deviation 5cm to true state)
- Covariance matrix estimated from variability of 9-year model trajectory (1991-1999) initialized from climatology
- Initial state estimate from perpetual 1990 model spin-up
- Monthly analysis updates (at initial time and after each month of model integration)
- No model error; forgetting factor 0.8 for both filters

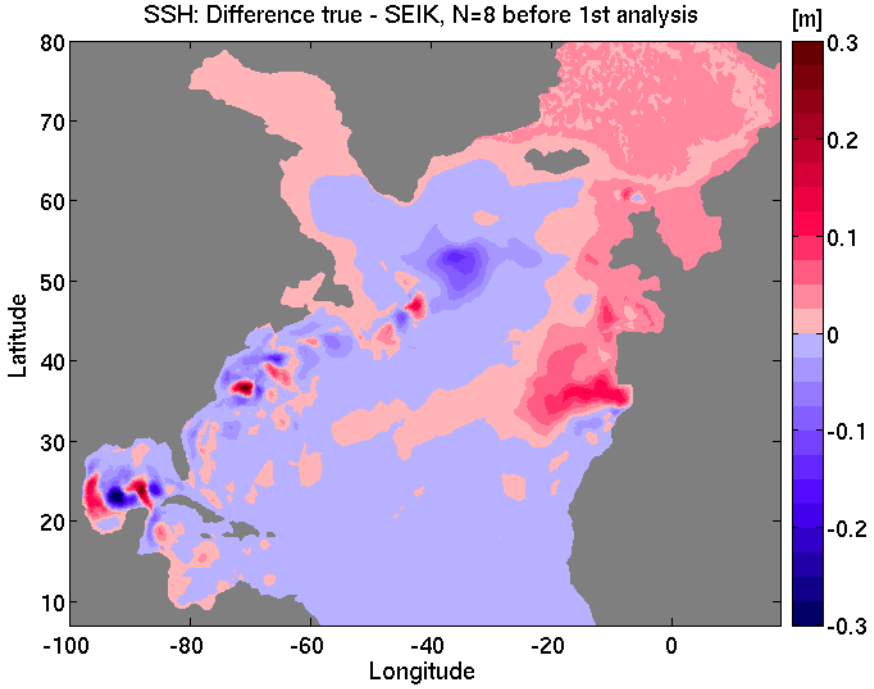
Nerger, L. et al., J. Mar. Syst. 65 (2007) 288-298

AWI

Modeled Sea Surface Height (Dec. 1992)



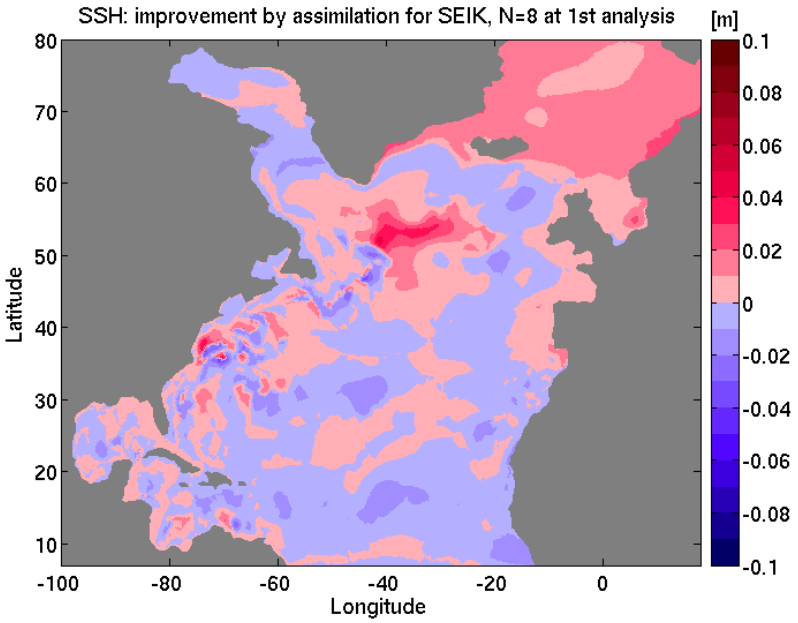




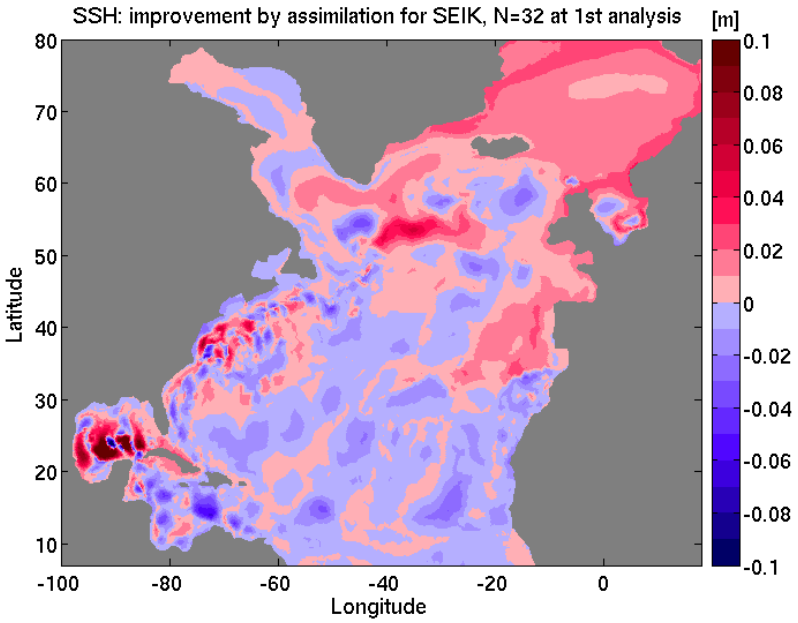
- large-scale deviations of small amplitude
- small-scale deviations up to 40 cm

Improvement of Sea Surface Height (Dec. 1992)

N=8



N=32



- Improvement: red - deterioration: blue

⇒ For N=8 rather coarse-scale corrections

⇒ Increased ensemble size adds finer scales (systematically)

Global SEIK filter - filtering behavior

- SEIK performs global optimization
- Degrees of freedom is small (ensemble size - 1)

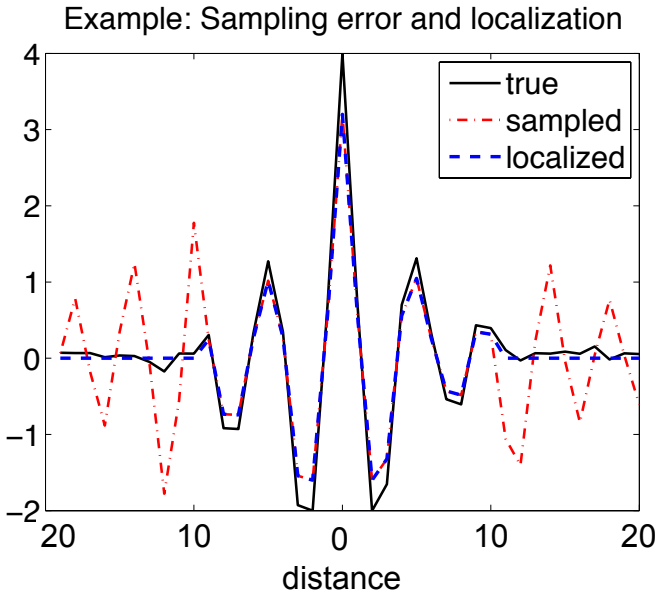
Implications:

- Global averaging in analysis can lead to local increase in estimation error
- Small-scale errors can be corrected, but error reduction is small
- True errors are underestimated
  (Due to inconsistency between true and estimated errors)

Localization

Localization: Why and how?

- Combination of observations and model state based on estimated error covariance matrices
- Finite ensemble size leads to significant sampling errors
  - particularly for small covariances!



- Remove estimated long-range correlations
  - ➜ Increases degrees of freedom for analysis (globally not locally!)
  - ➜ Increases size of analysis correction

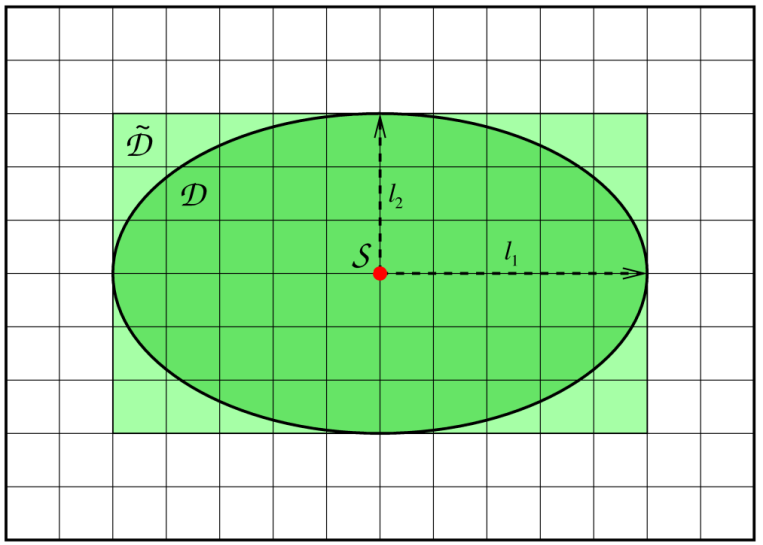
(introduced for EnKFs by Houtekamer & Mitchell 1998)

AWI

Local SEIK filter

Perform a loop over local analysis domains *S*

- Analysis:
  - Update small regions (e.g. single water columns)
  - Consider only observations within cut-off distance
    - ➢ neglects long-range correlations
- Ensemble Transformation:
  - Transform local ensemble
  - Use same transformation matrix in each local domain



S: Analysis region
D: Corresponding data region

Nerger, L. et al. Ocean Dynamics 56 (2006) 634

The SEIK Filter with local update

Analysis sub-domain: $\mathcal{S}_\sigma$

Observation sub-domain: $\mathcal{D}_\delta$

Local Analysis:

$$\mathbf{x}_\sigma^a = \overline{\mathbf{x}_\sigma^f} + \mathbf{L}_\sigma \mathbf{a}_\delta$$

$$\mathbf{a}_\delta = \mathbf{U}_\delta (\mathbf{H}_\delta \mathbf{L})^T \mathbf{R}_\delta{}^{-1} \left(\mathbf{y}_\delta^o - \mathbf{H}_\delta \overline{\mathbf{x}^f} \right)$$

$$\mathbf{U}_\delta^{-1} = \rho_\delta \mathbf{G}^{-1} + (\mathbf{H}_\delta \mathbf{L})^T \mathbf{R}_\delta^{-1} \mathbf{H}_\delta \mathbf{L}$$

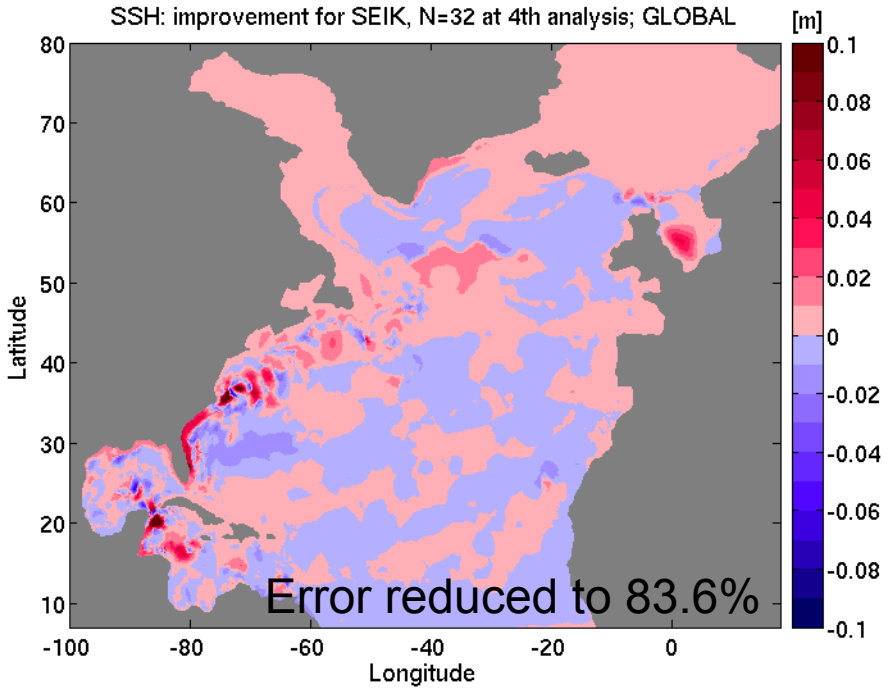
$$\mathbf{H}_\delta := \mathbf{D}_\delta \mathbf{H}$$

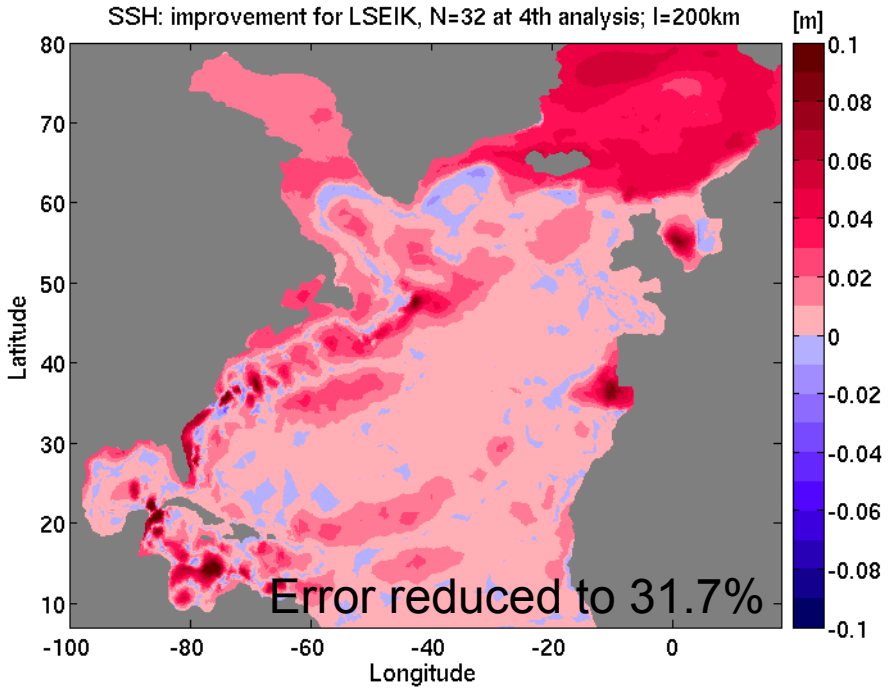
Local Re-Initialization:

$$\mathbf{X}_\sigma^a = \overline{\mathbf{X}_\sigma^a} + \sqrt{N-1}\, \mathbf{L}_\sigma \mathbf{C}_\delta^T \mathbf{\Omega}^T$$

with $\mathbf{C}_\delta^{-1} (\mathbf{C}_\delta^{-1})^T = \mathbf{U}_\delta^{-1}$ (Cholesky decomposition)

Global vs. local SEIK, N=32 (March 1993)

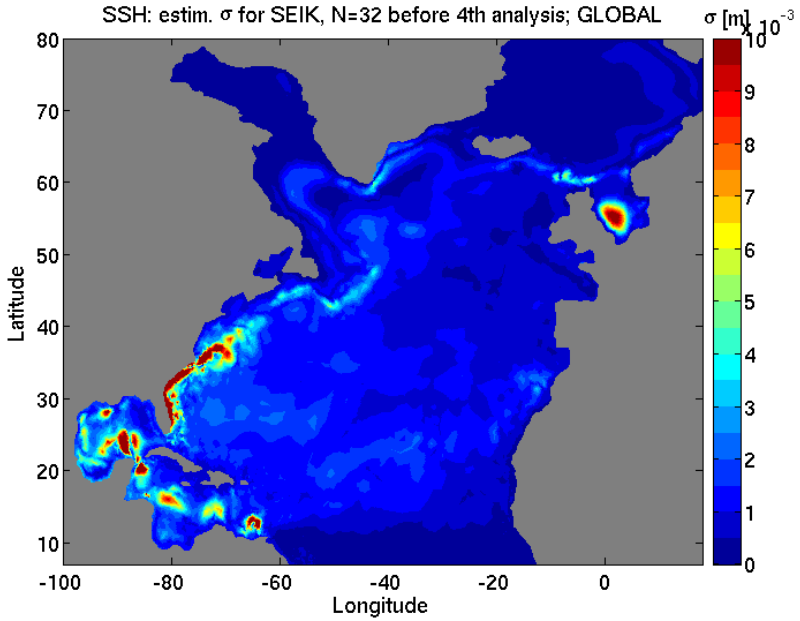


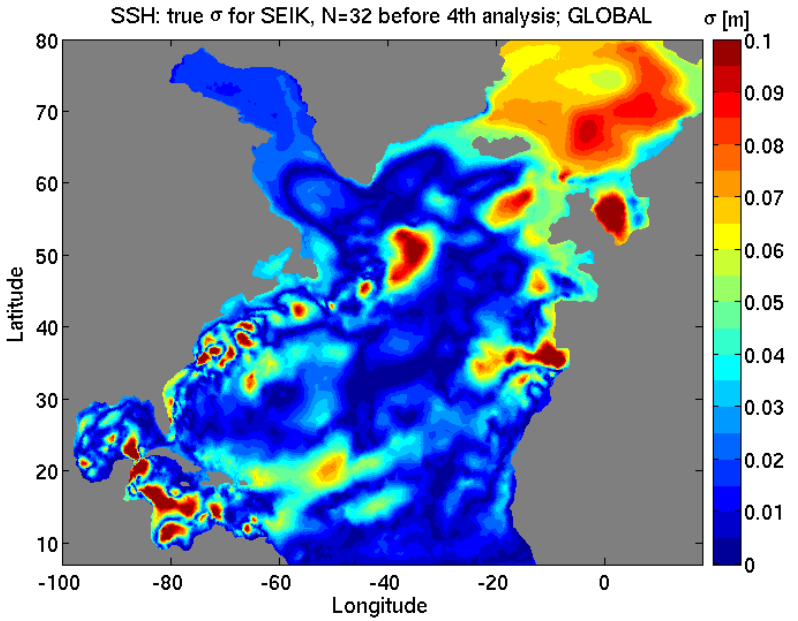


- Improvement is error reduction by assimilation
- Localization extents improvements into regions not improved by global SEIK
- Regions with error increase diminished for local SEIK
- Underestimation of errors reduced by localization

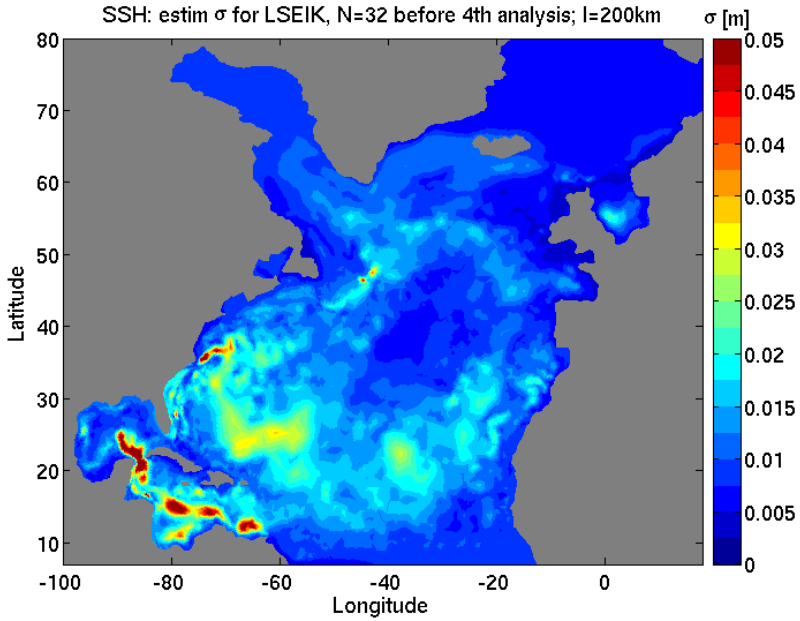
LSEIK: True and estimated errors - third forecast

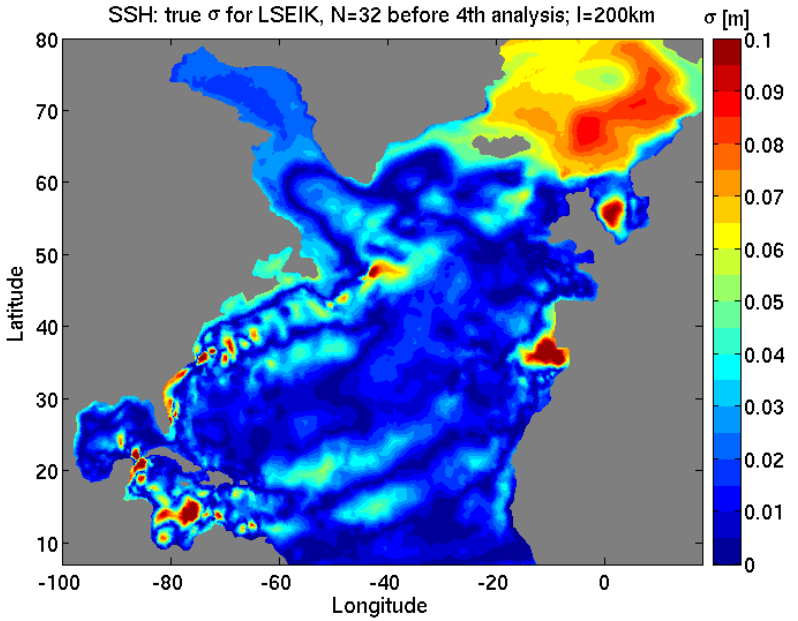
SEIK



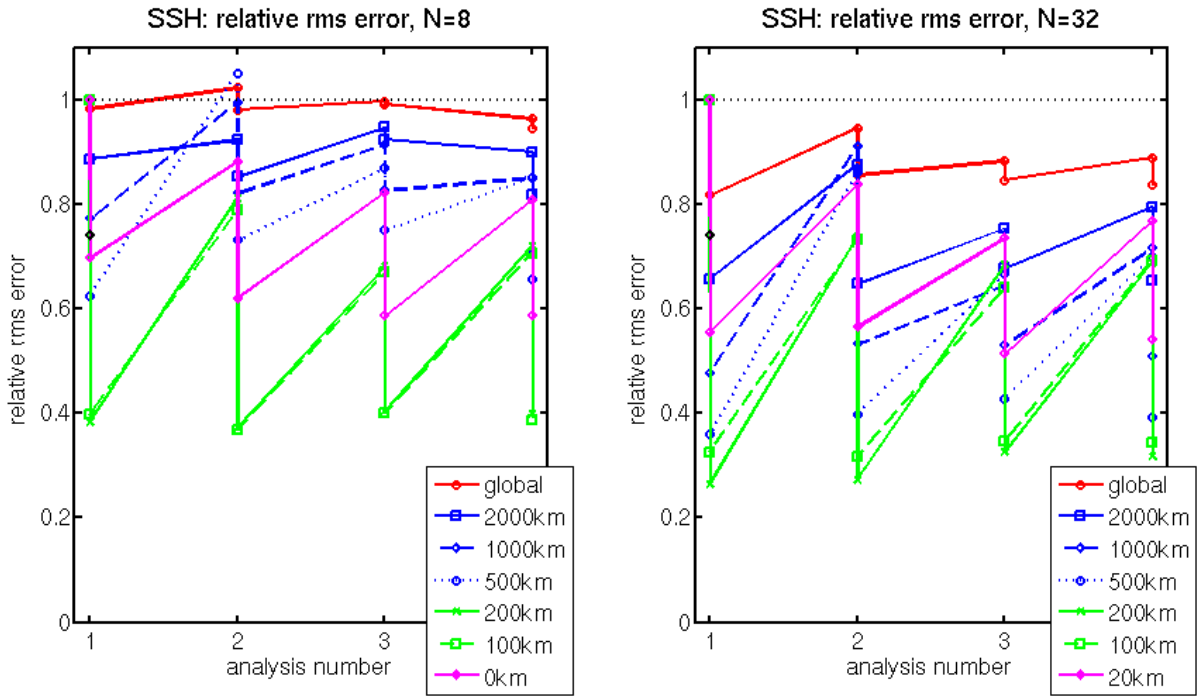


LSEIK





Relative rms errors for SSH



- global filter: significant improvement for larger ensemble
- global filter with N=100: relative rms error 0.74
- localization strongly improves estimate
  - larger error-reduction at each analysis update
  - but: stronger error increase during forecast
- very small radius results in over-fitting to noise

Local SEIK filter – filtering behavior

- LSEIK performs series of local optimizations
- Degrees of freedom given by ensemble size - 1 for each analysis domain

Implications:

- Localization can strongly improve filtering performance over the global SEIK
- Localization can lead to faster error-increase during forecast (imbalance problem)

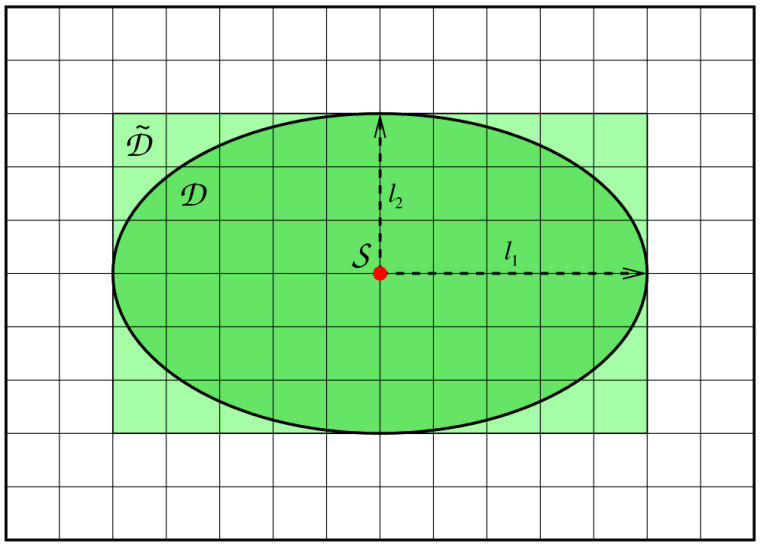
⇒ possible trade off between improved analysis update and forecast error-increase

- LSEIK is more costly than global SEIK, but computationally still efficient

Local SEIK filter – domain & observation localization

Local Analysis:

- Update small regions (like single vertical columns)
- Observation localizations: Observations weighted according to distance
- Consider only observations with weight >0
- State update and ensemble transformation fully local



S: Analysis region
D: Corresponding data region

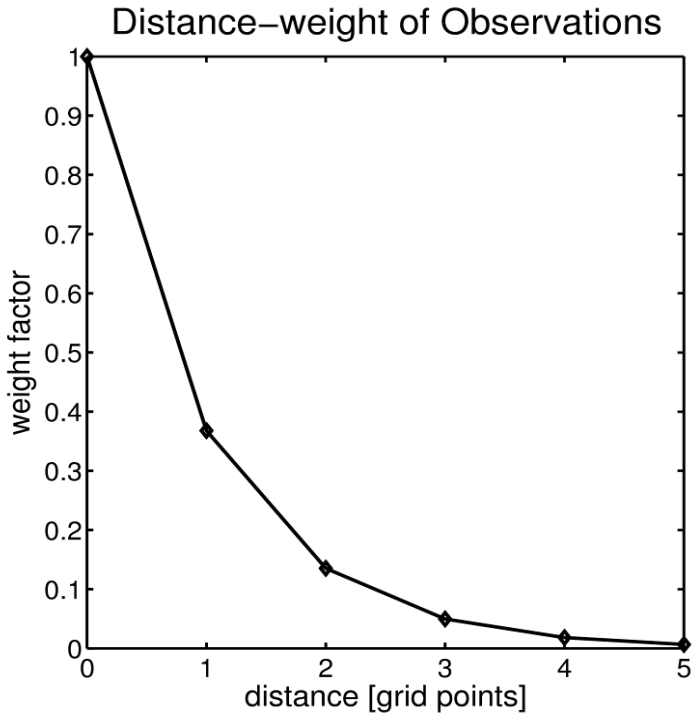
Similar to localization in LETKF (e.g. Hunt et al, 2007)

L. Nerger & W.W. Gregg, J. Mar. Syst. 68 (2007) 237

Observation localization

Localizing weight

- reduce weight for remote observations by increasing variance estimates
- use e.g. exponential decrease or polynomial representing correlation function of compact support
- similar, sometimes equivalent, to *covariance localization* used in other ensemble-based KFs



Localization Types

Simplified analysis equation:

$$\mathbf{x}^a = \mathbf{x}^f + \frac{\mathbf{P}^f}{\mathbf{P}^f + \mathbf{R}} \left(\mathbf{y} - \mathbf{x}^f \right)$$

Covariance localization

- Modify covariances in forecast covariance matrix $\mathbf{P}^f$
- Element-wise product with correlation matrix of compact support

Requires that $\mathbf{P}^f$ is computed (not in ETKF or SEIK)

E.g.: Houtekamer/Mitchell (1998, 2001), Whitaker/Hamill (2002), Keppenne/ Rienecker (2002)

Observation localization

- Modify observation error covariance matrix $\mathbf{R}$
- Needs distance of observation (achieved by local analysis or domain localization)

Possible in all filter formulations

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

Relation of Covariance and Observation Localization

Recently a *hot* topic …

- Sakov & Bertino, Comput. Geosci. (2011)
- Greybush et al., Mon. Wea. Rev. (2011)
- Brankart et al., Mon. Wea. Rev. (2011)

From AWI:

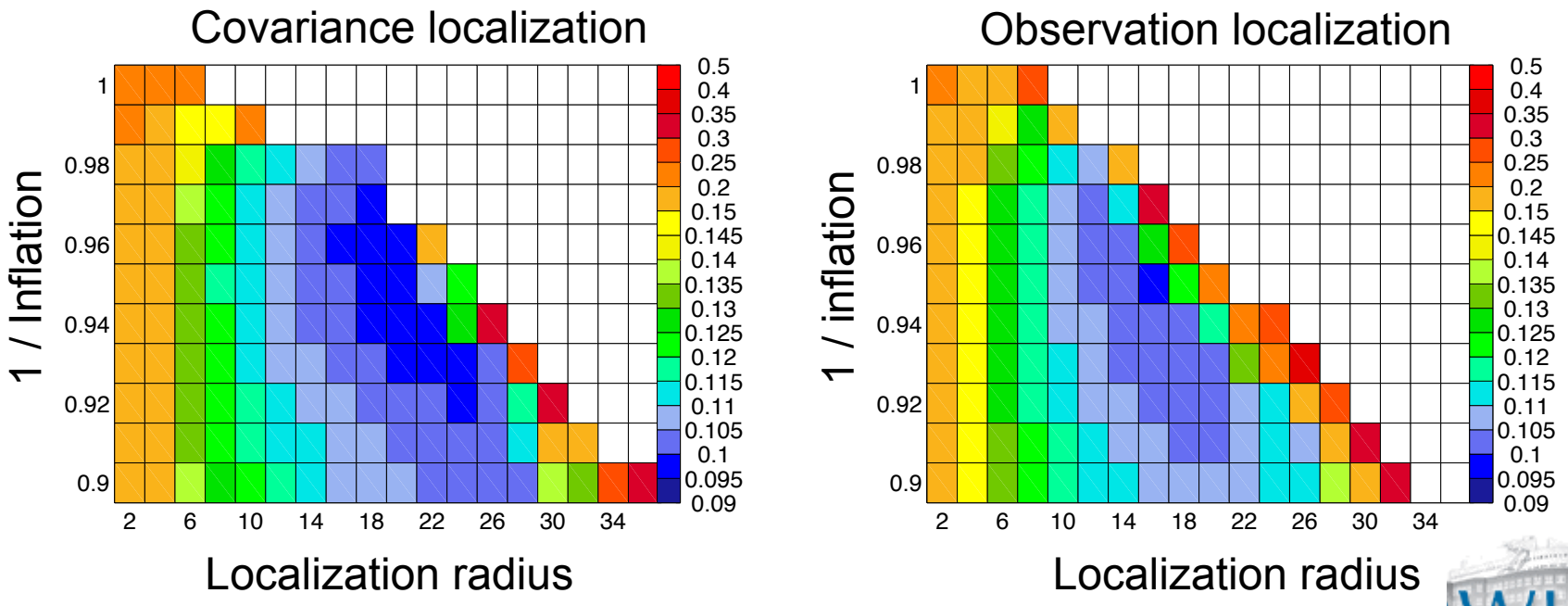
- Janjic et al., Mon. Wea. Rev. (2011)
- Nerger et al., QJ Roy. Meteorol. Soc. (2012)

Different effect of localization methods

Experimental result:

- Twin experiment with simple Lorenz96 model
- Covariance localization better than observation localization
  (Also reported by Greybush et al. (2011) with other model)

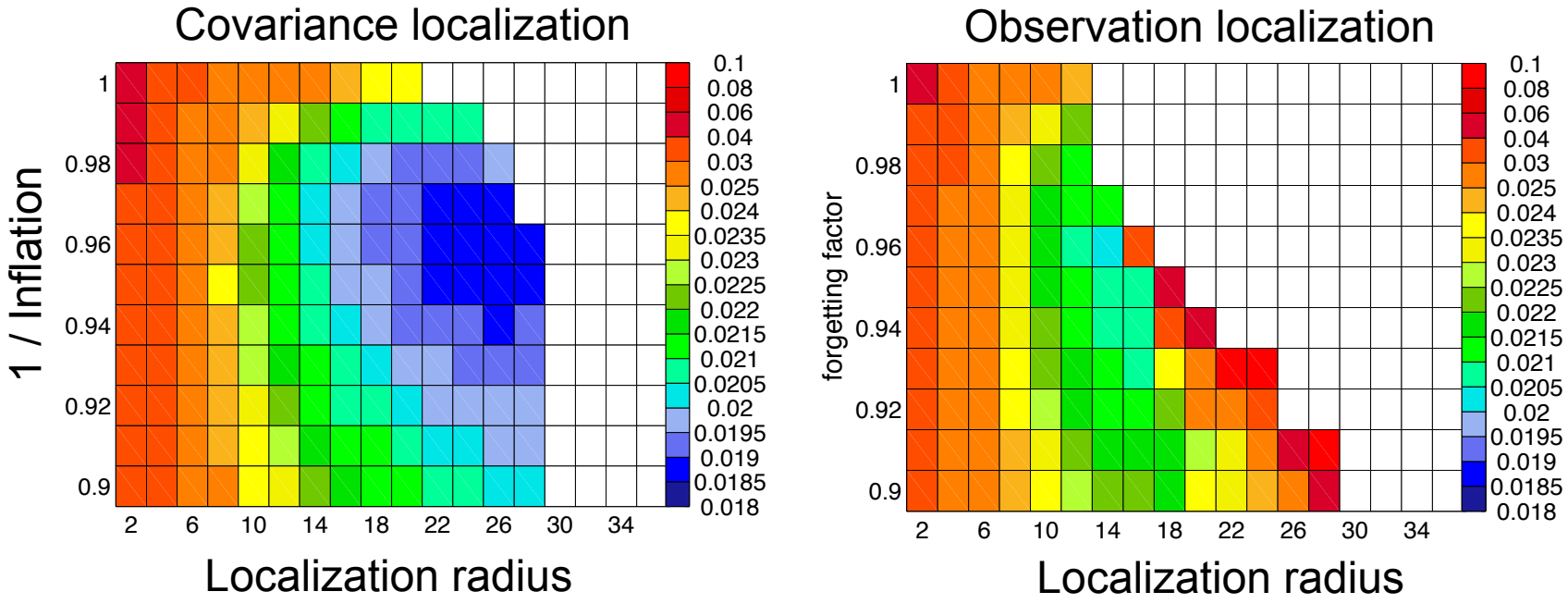
Time-mean RMS errors



T. Janjic et al., Mon. Wea. Rev. 139 (2011) 2046-2060

Different effect of localization methods (cont.)

Larger differences for smaller observation errors



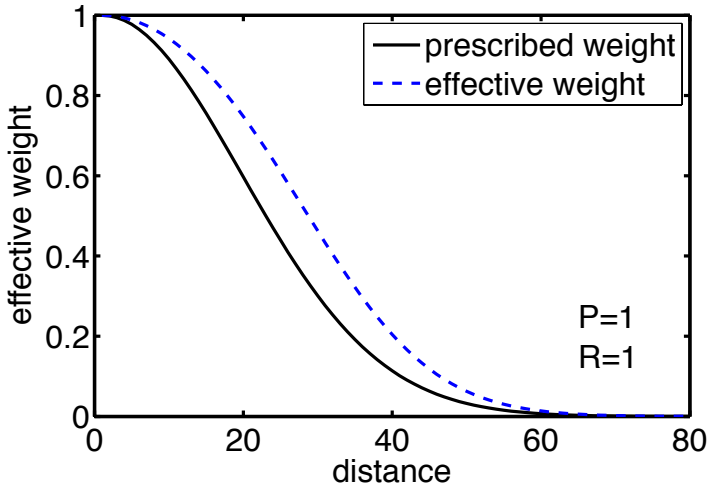
Covariance vs. Observation Localization

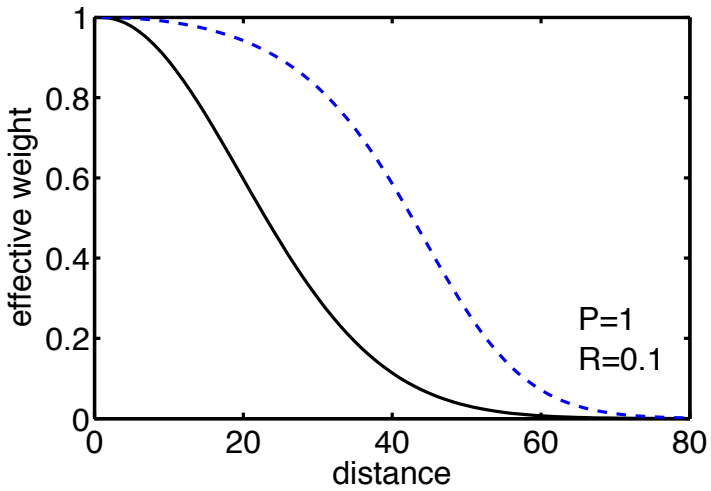
Some published findings:

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

But note for observation localization:

- Effective localization length 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

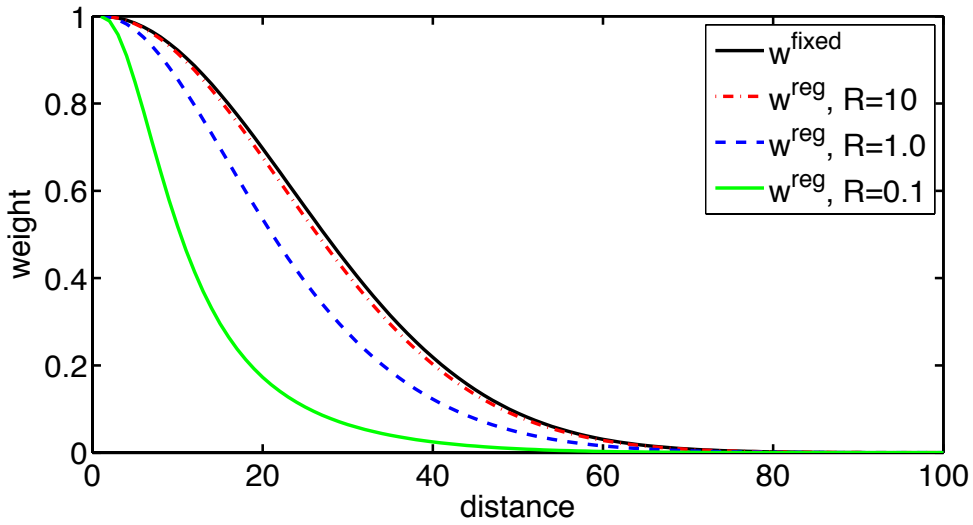




P: state error variance
R: observation error variance

Regulated Localization

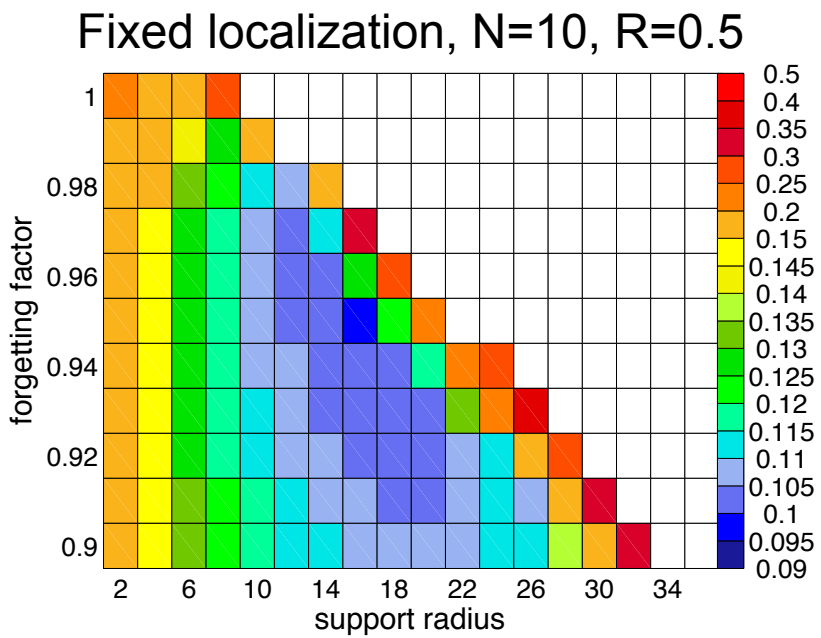
- New localization function for observation localization
  - formulated to keep effective length constant (exact for single observation)
  - depends on state and observation errors
  - depends on fixed localization function
  - cheap to compute for each observation
  - Only exact for single observation – works for multiple

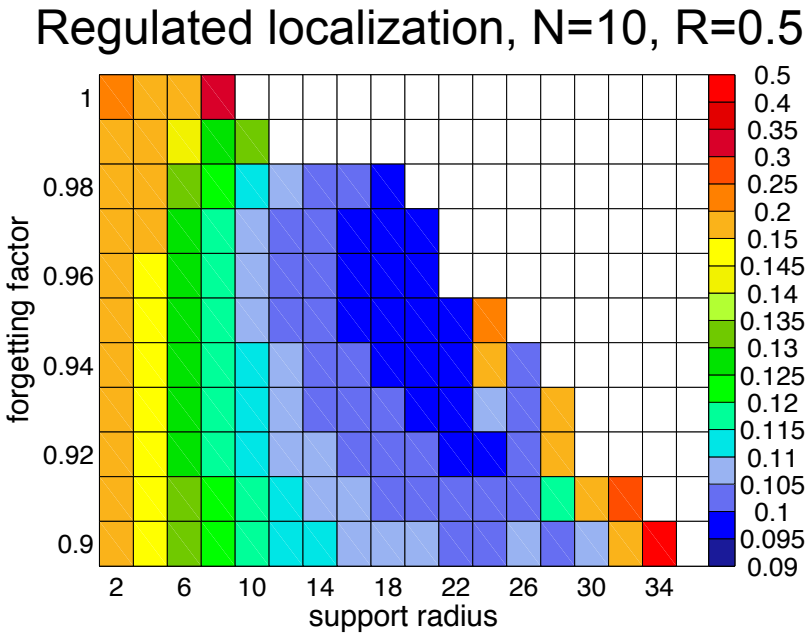


P=1

L. Nerger et al. QJ Royal. Meterol. Soc. 138 (2012) 802-812

Lorenz96 Experiment: Regulated Localization

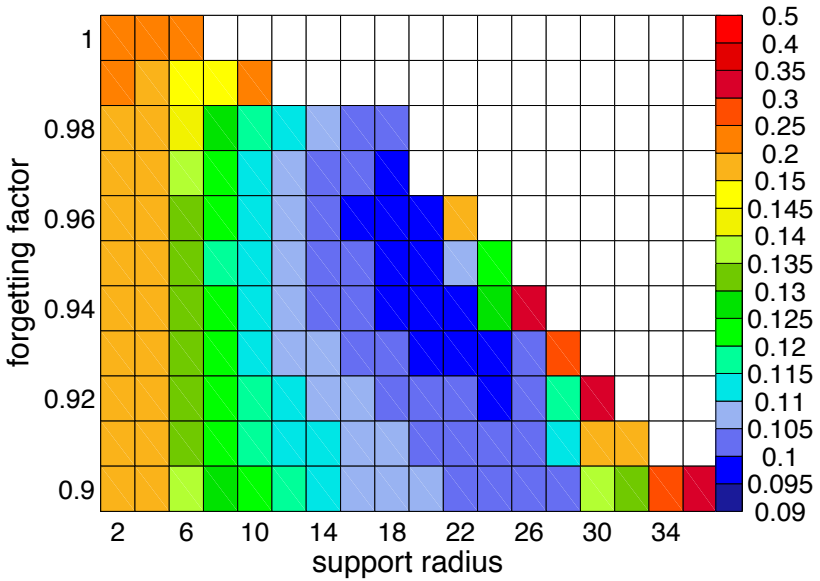




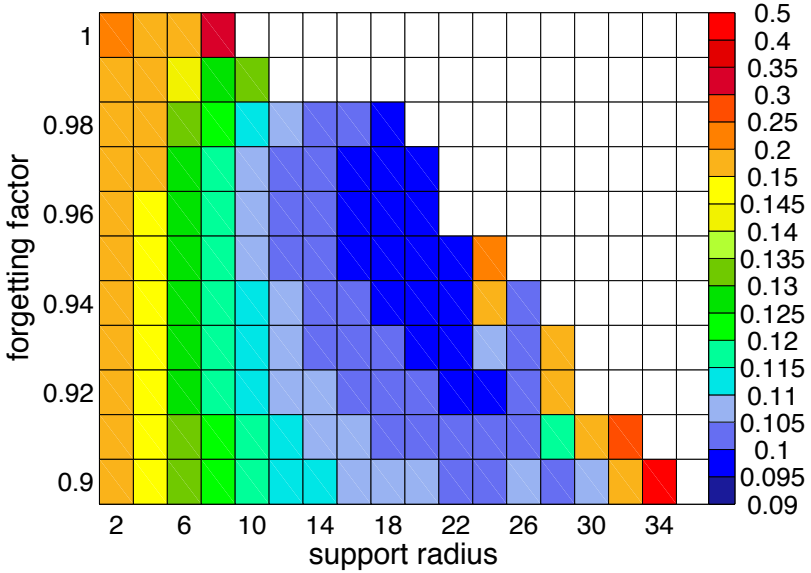
- Reduced minimum rms errors
- Increased stability region
- Description of effective localization length explains the findings of other studies!
- Impact also with FEOM ocean model (but smaller)

Lorenz96 Experiment: Regulated Localization

Covariance loc., N=10, R=0.5



Regulated localization, N=10, R=0.5



- Reduced minimum rms errors
- Increased stability region
- Description of effective localization length explains the findings of other studies!
- Impact also with FEOM ocean model (but smaller)

Covariance inflation

Covariance inflation

- True variance is always underestimated
  - finite ensemble size
  - sampling errors (unknown structure of P)
  - model errors

  ➜ can lead to filter divergence

- Simple remedy

  ➜ Increase error estimate before analysis

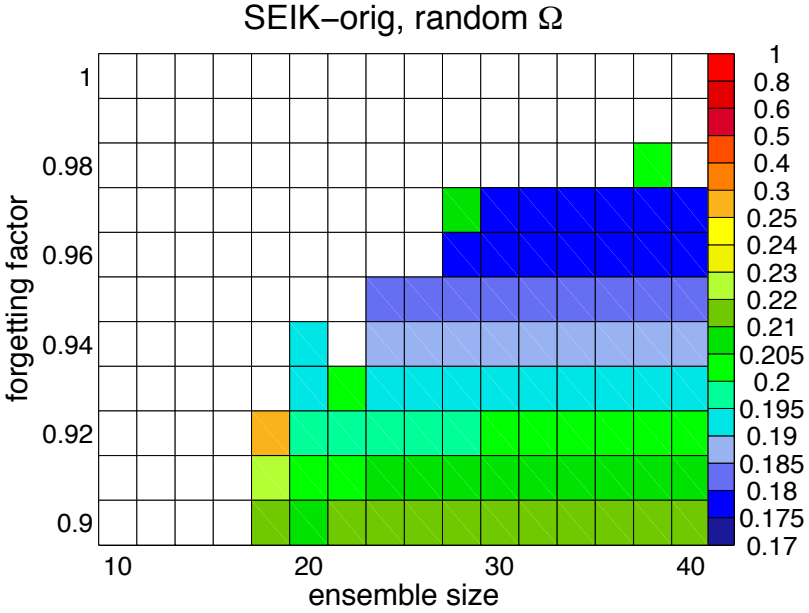
- Possibilities
  - Increase ensemble spread (“inflation”)
  - Multiply covariance matrix by a factor slightly above 1
  - Additive error term (e.g. on diagonal)

(Mathematically, this is a regularization)

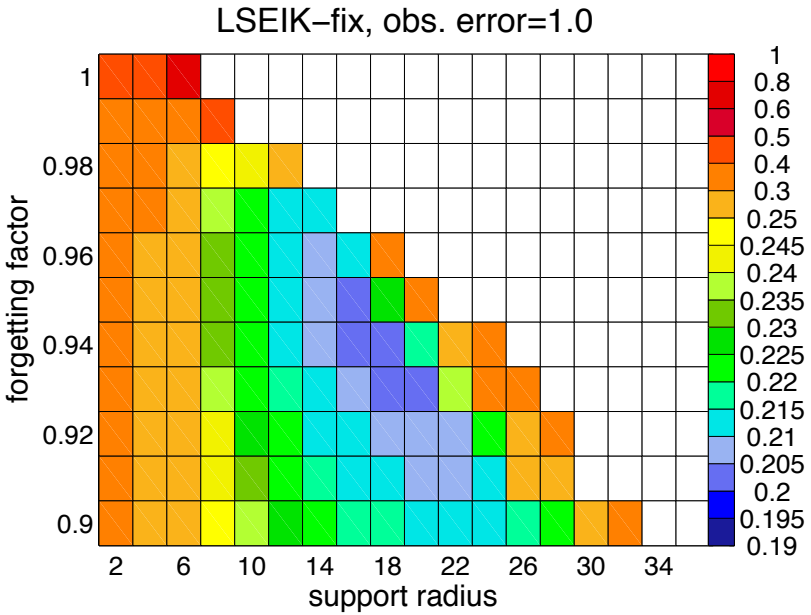
Impact of inflation on stability & performance

Experiments with Lorenz96 model

Global filter



Localized, ensemble size 10



- Increased stability with stronger inflation (smaller forgetting factor)
- Optimal choice for inflation factor

Observations and their errors

Real observations

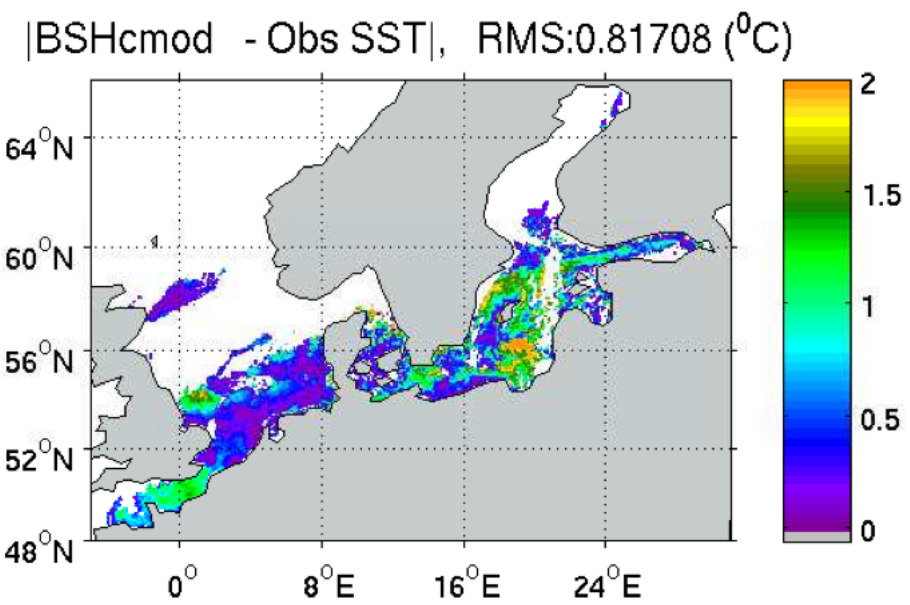
- Observation errors
  - measurement errors
  - representation errors
- Real observations are not ideal
  - Incomplete (space, time)
  - Errors only estimated
  - Errors can be correlated
  - Can be biased

➜ Usual way of handling: pragmatism

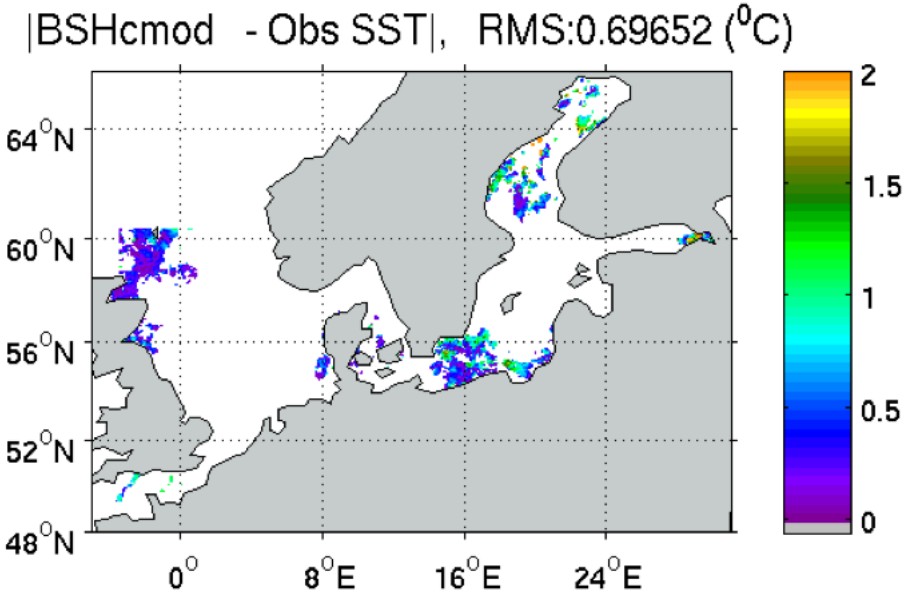
Observation availability

Surface temperature

14.10.2007 00:00±6h



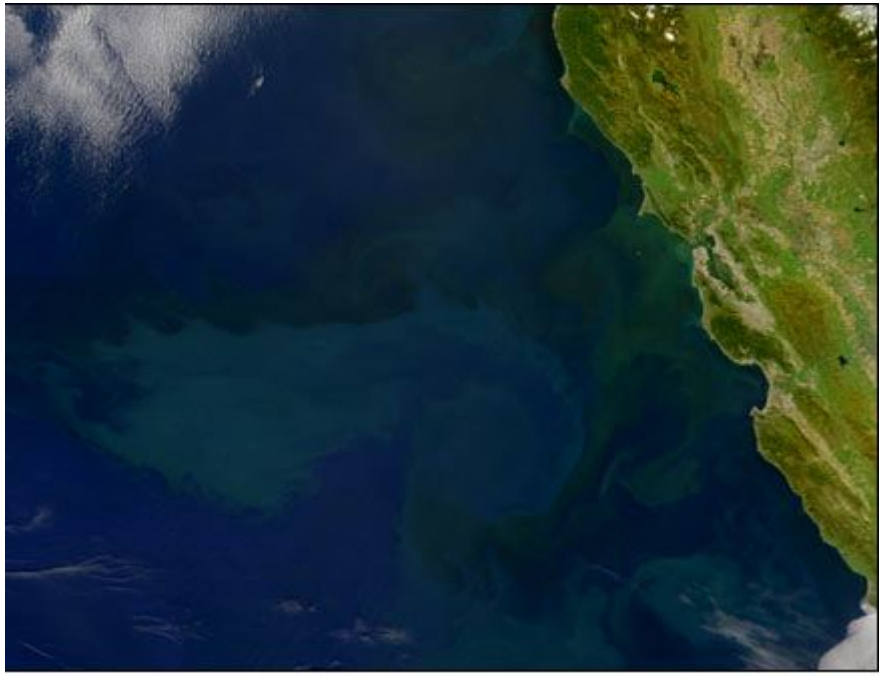
27.10.2007 00:00±6h



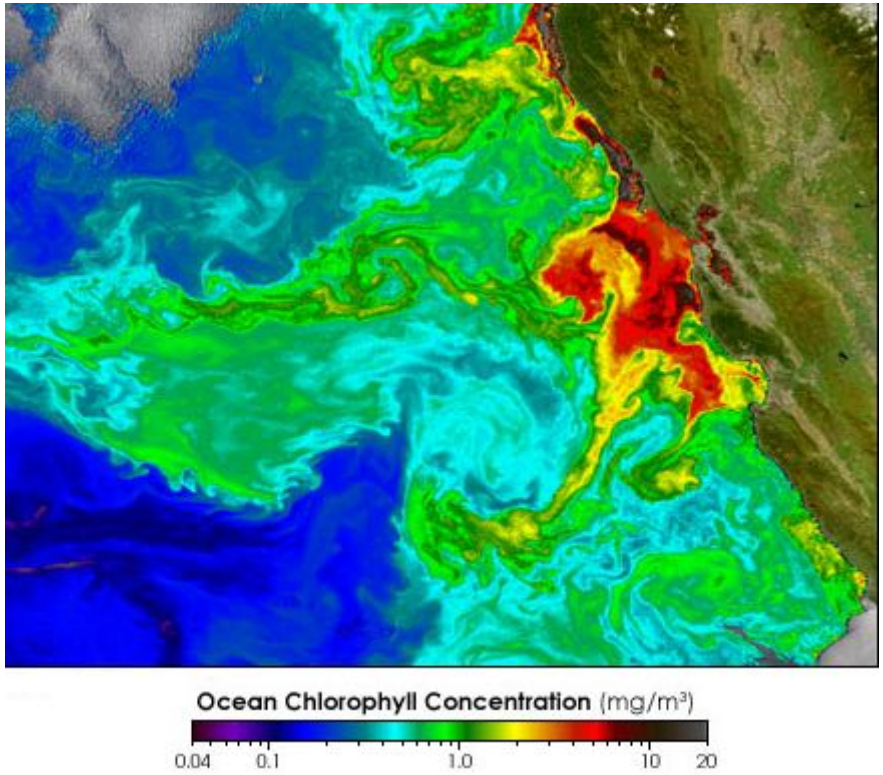
- Strongly irregular data availability
- Frequent data gaps
- Assume constant error and homogeneous spatial influence

Satellite Ocean Color (Chlorophyll) Observations

Natural Color 3/16/2004

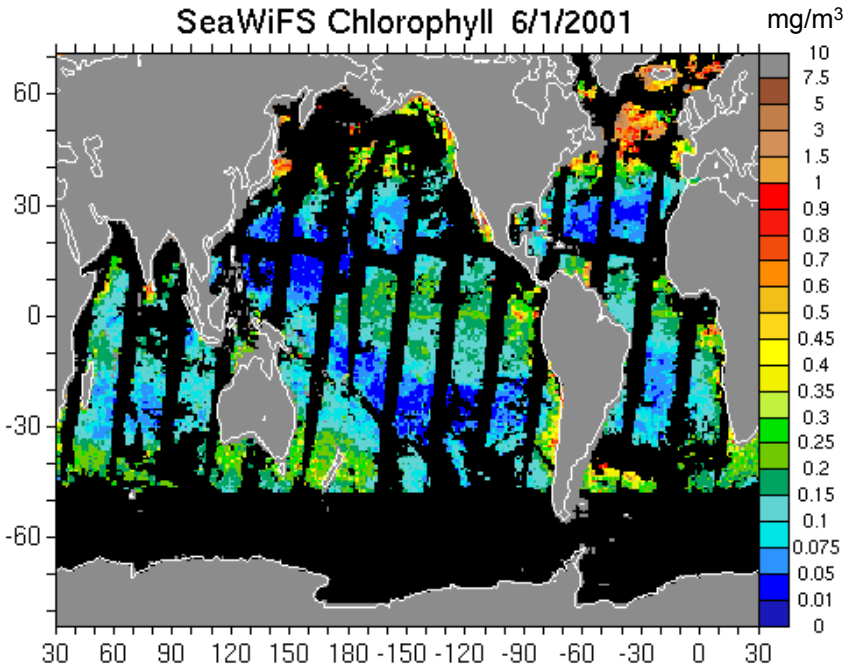


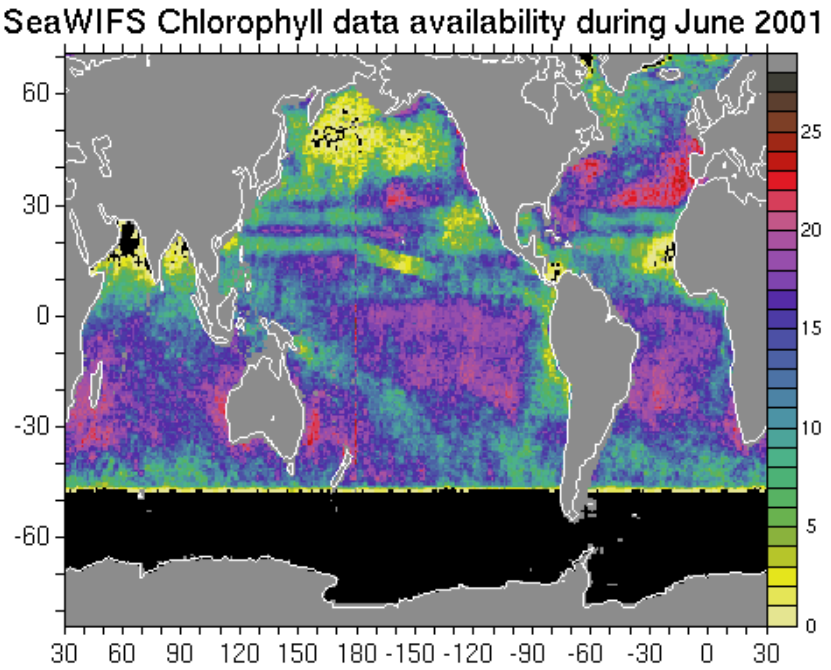
Chlorophyll Concentrations



Source: NASA "Visible Earth", Image courtesy the SeaWiFS Project, NASA/GSFC, and Orbimage

Assimilated Observations

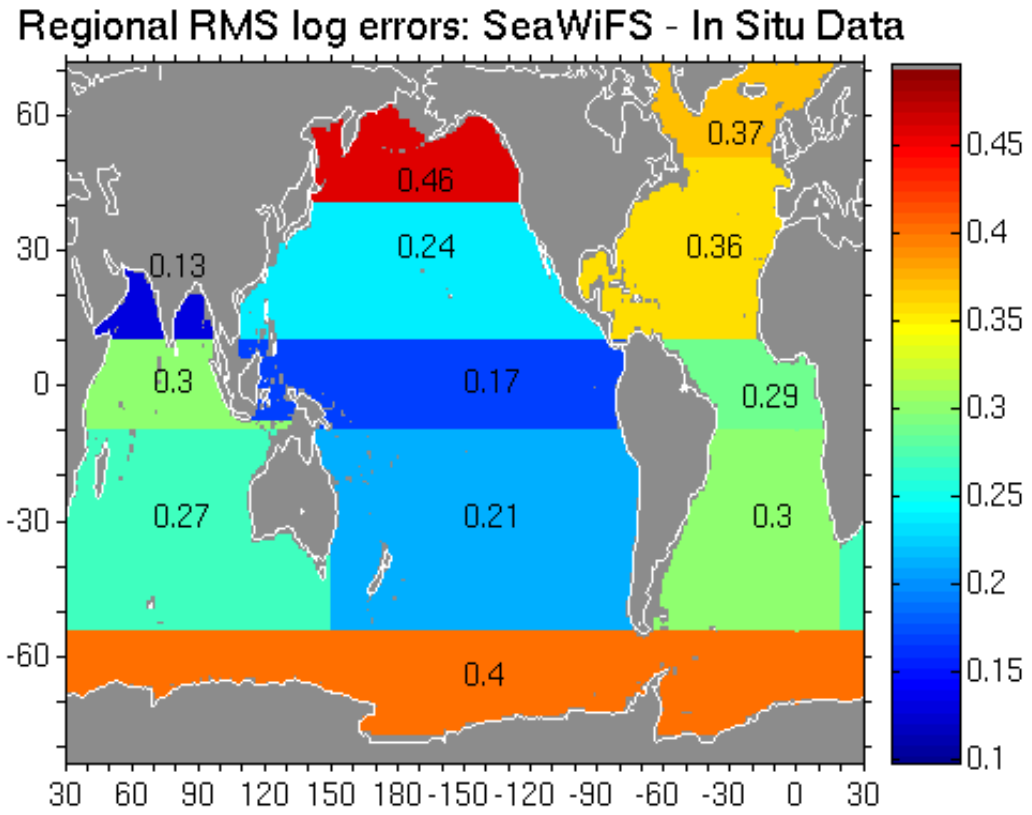




- Daily gridded SeaWiFS chlorophyll data
  - gaps: satellite track, clouds, polar nights
  - ~13,000-18,000 data points daily (of 41,000 wet grid points)
  - irregular data availability

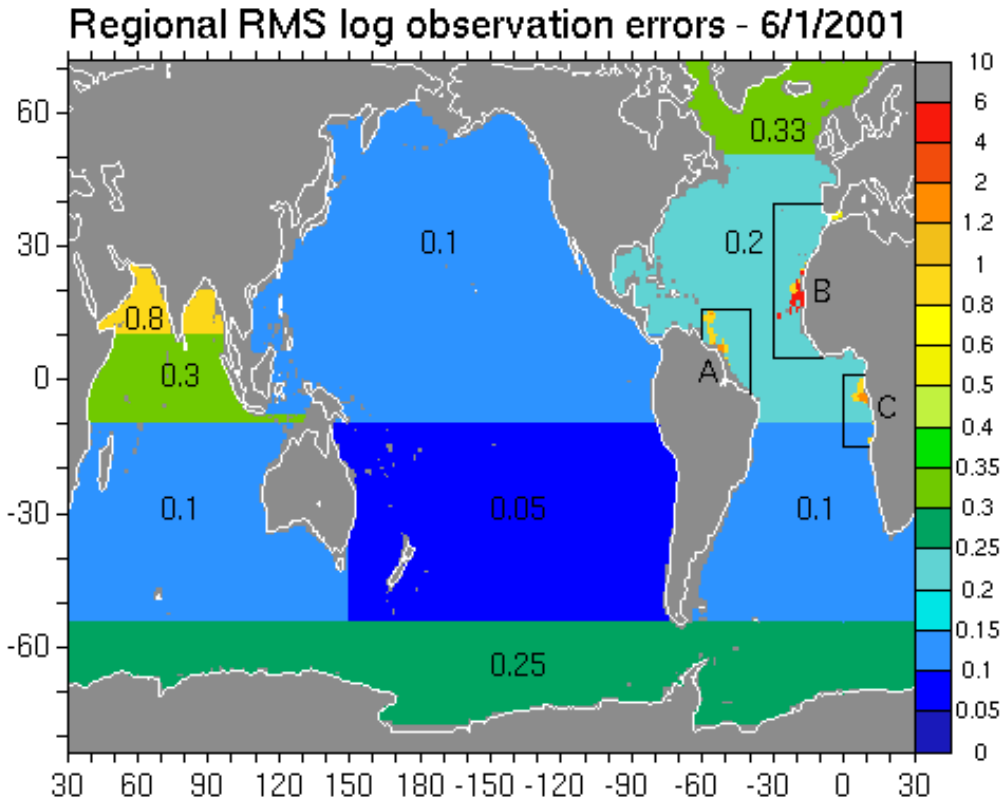
Nerger, L., and W.W. Gregg. J. Marine Systems **68** (2007) 237

Error Estimates



Regional data errors from comparison with 2186 collocation points of in situ data

Observation errors II



- Account regionally for larger errors caused by
  - aerosols (North Indian Ocean, tropical Atlantic)
  - CDOM (Congo and Amazon)
- Error estimates adjusted for filter performance and stability

Model Errors

Model errors

- Representation of reality is not exact
  - Insufficient resolution
  - Incomplete equations (e.g. missing processes)
  - Inexact forcing (e.g. wind stress on ocean surface)

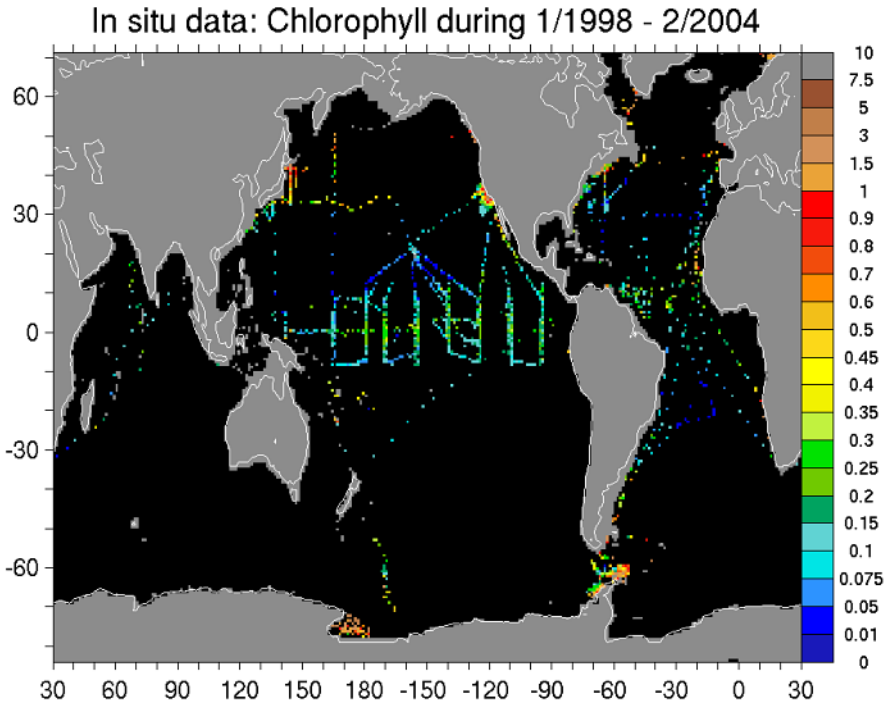
- Accounting for model error
  - Inflation (partly)
  - Simulate stochastic part
  - Bias estimation

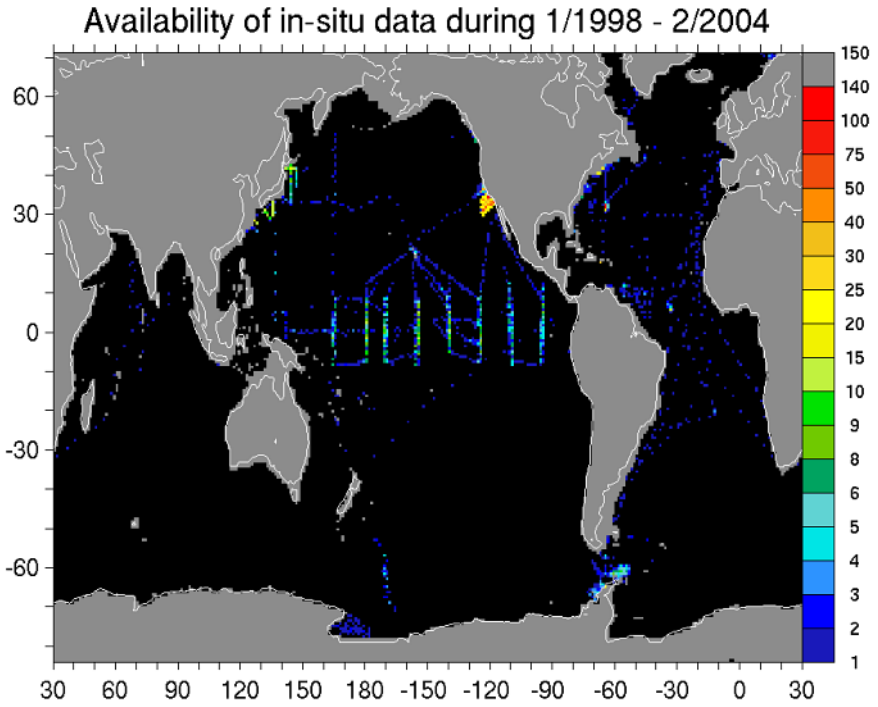
Validation data

Validating a data assimilation system

- Need independent data for validation
  - ➢ Necessary, but not sufficient:
    Reduction of deviation from assimilated data
  - Required:
    - Reduction of deviation from independent data
    - Reduction of errors for unobserved variables
- Want to assimilate all available data (in the ocean)
  - ➢ Data-withholding experiments
  - ➢ Twin experiments
  - ➢ Validate with data of small influence

In-Situ chlorophyll data





- In situ data from SeaBASS/NODC over 1/1998-2/2004
- Independent from SeaWiFS data
  (only used for verification of algorithms)
- North Central Pacific dominated by CalCOFI data
- North Central Atlantic dominated by BATS data

Case Study 2:

An ensemble-based forecasting system for the North and Baltic Seas

Joint work with

Svetlana Loza, Jens Schröter
Alfred Wegener Institute

Silvia Massmann, Frank Janssen
Federal Maritime and Hydrographic Agency (BSH)

DeMarine

AWI

Toward operational data assimilation in the North Sea and Baltic Sea

Joint project with German Federal Maritime and Hydrographic Agency (BSH)

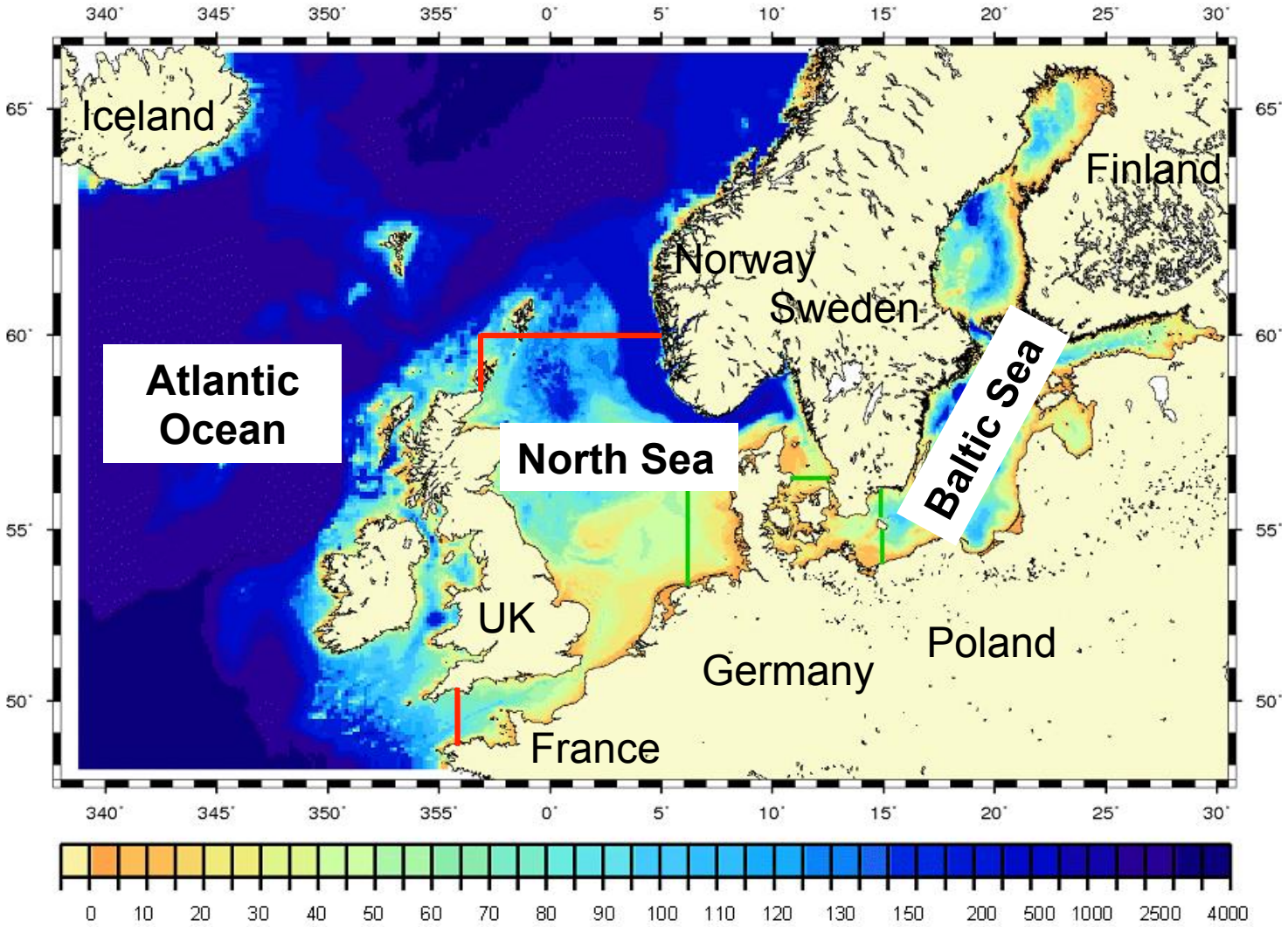
Aim:
Improve ocean forecasts by adding data assimilation

BUNDESAMT FÜR SEESCHIFFFAHRT UND HYDROGRAPHIE

DeMarine

AWI

Operational BSH Model (BSHcmod), Version 4



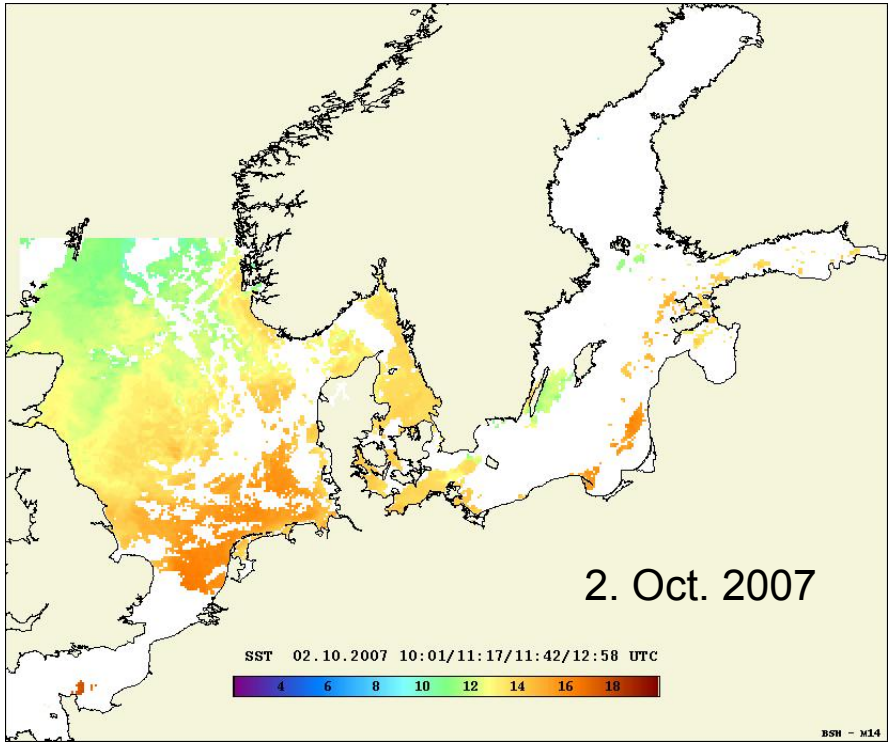
BSSC 2007, F. Janssen, S. Dick, E. Kleine

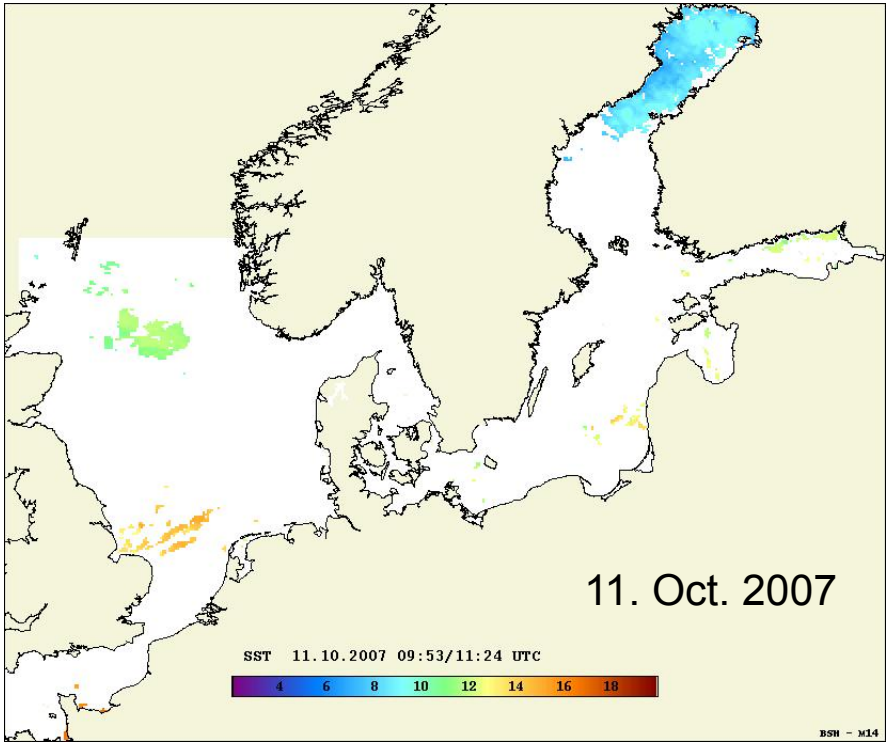
Grid nesting:

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

Data assimilation: 5 km grid

Assimilated Data - Satellite





- Surface temperature (from NOAA satellites)
- 12-hour composites
- Strong variation of data coverage (clouds)

Assimilation Methodology

- Ensemble Kalman filter (local SEIK)
- 12-hour forecast/analysis cycles
- Ensemble size 8 (sufficient for good results)
- Assumed data errors (SST):

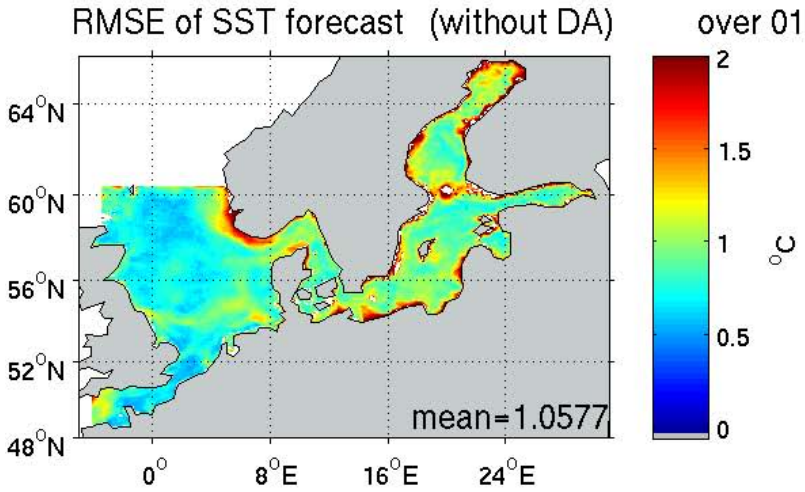
  uncorrelated, 0.8^{o}C (gave best results)
- Localization:
  - Weight on data errors
  - Exponential, e-folding at 100 km (tuned)
- Implementation:
  - Single program with PDAF

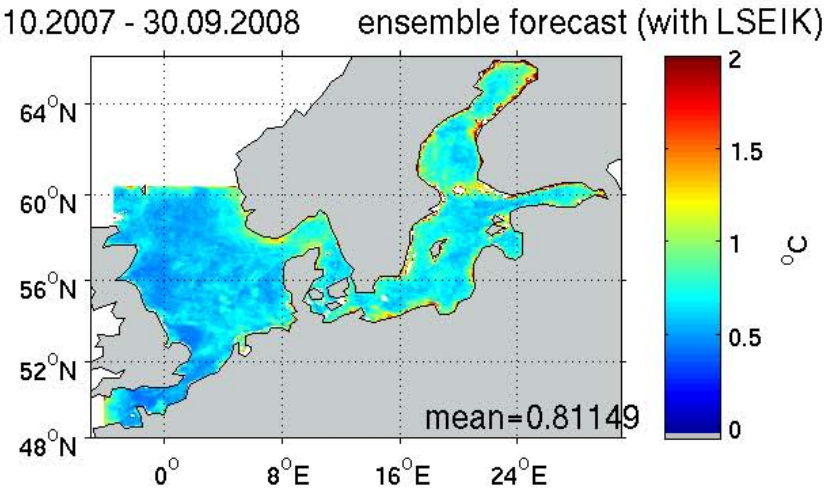
Deviation from NOAA Satellite Data

No assimilation

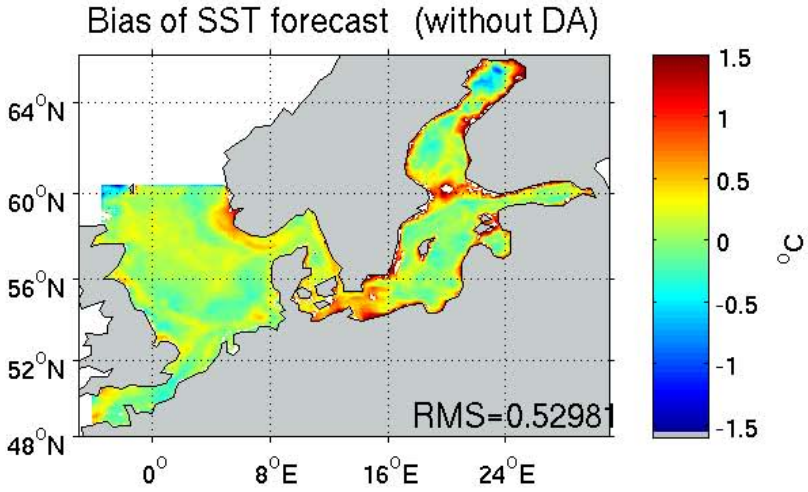
Assimilation

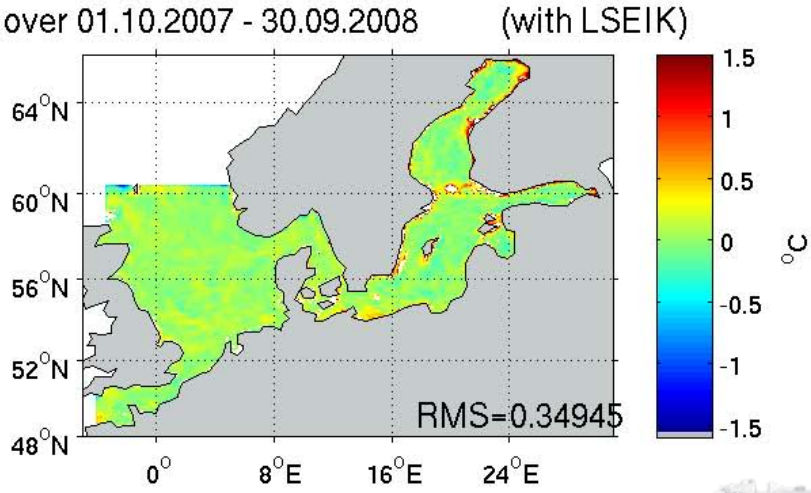
RMS



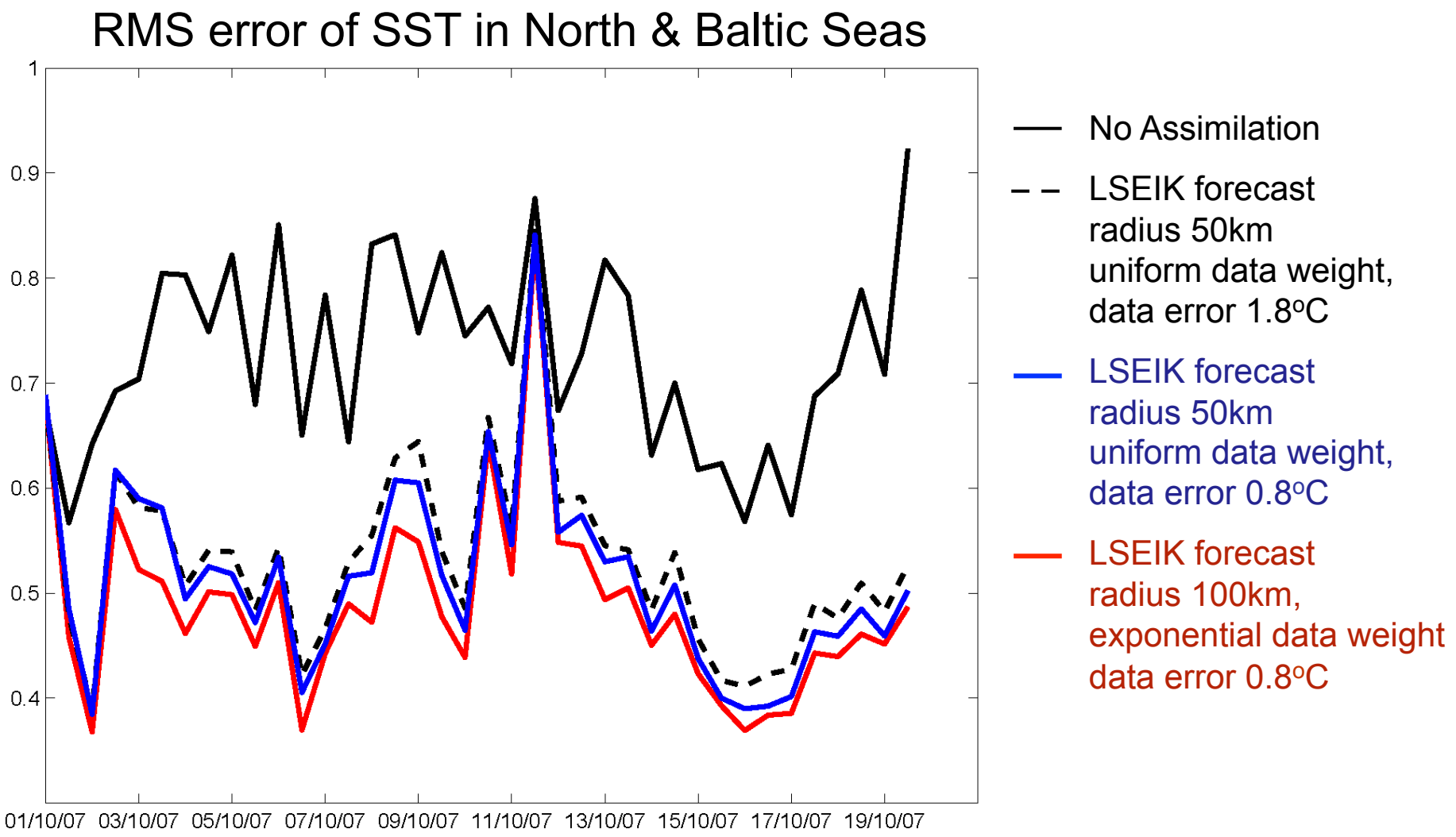


bias



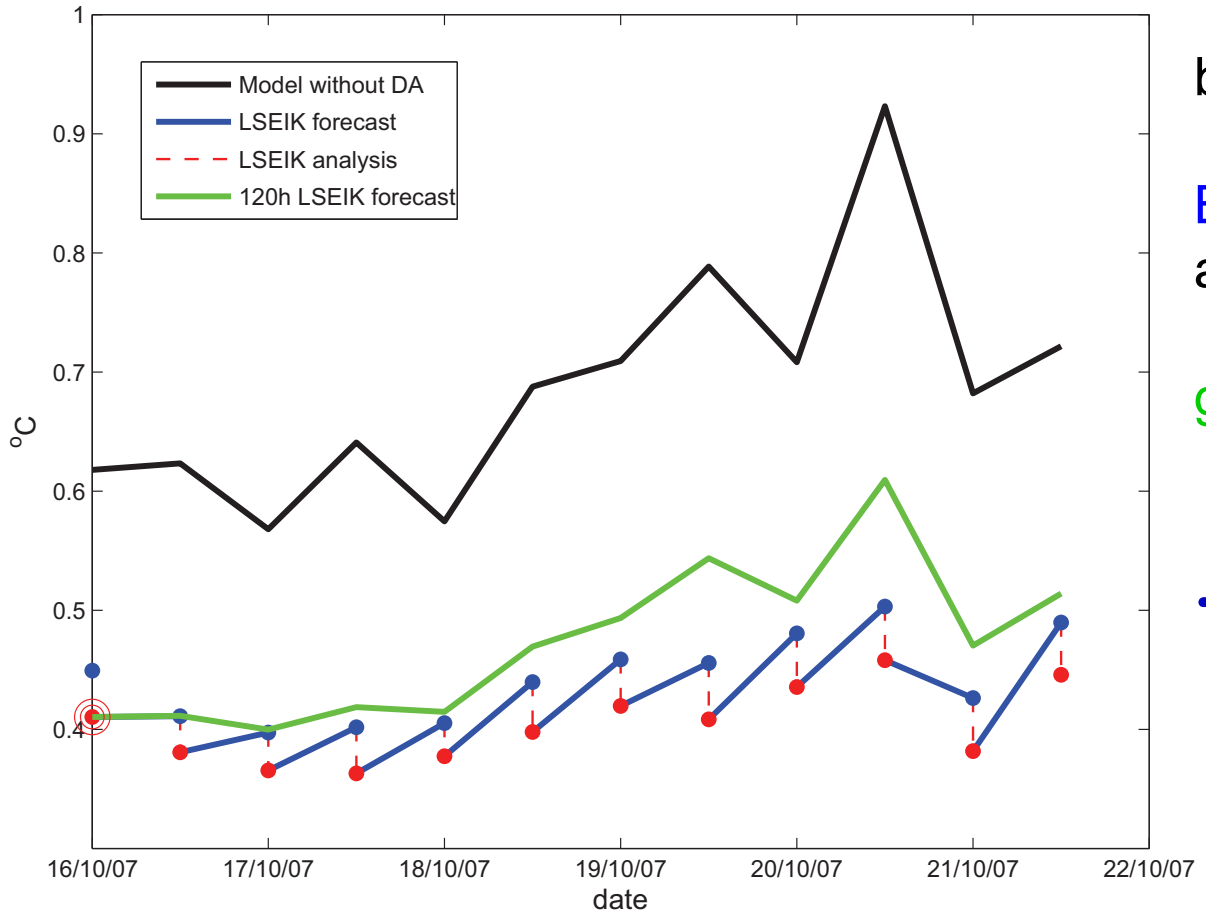


Influence of observation weighting



Improvement of long forecasts

SST RMS error over time



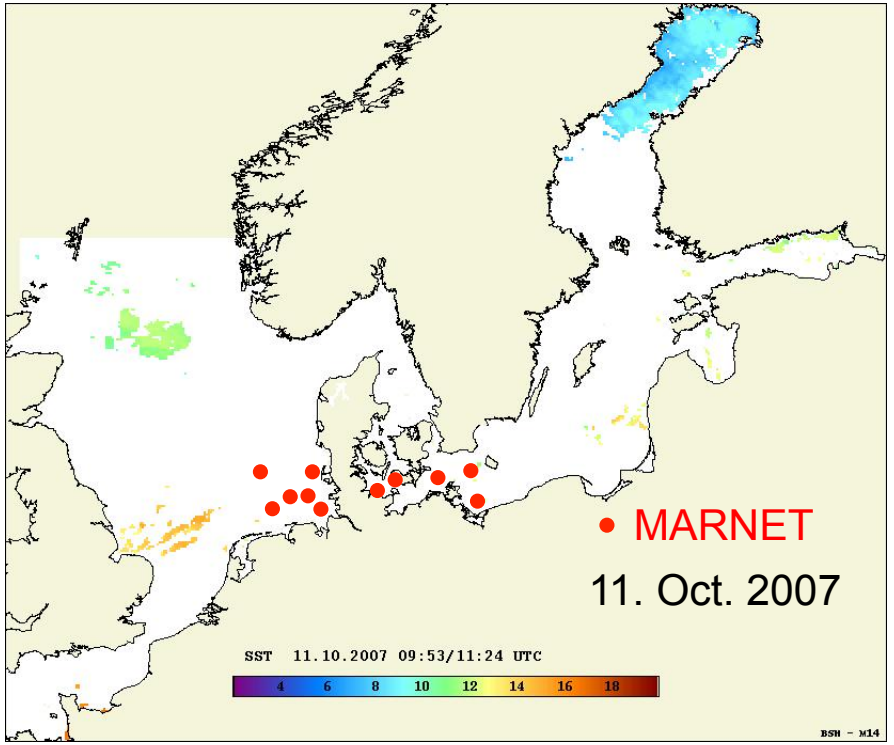
black: free model run

Blue/red: 12h assimilation/ analysis cycles

green: 5 day forecast

➜ Deviation grows very slowly

Validation data



- In situ data from MARNET network
- Fixed stations measuring atmosphere and various depths from surface to bottom
- Limited spatial coverage

Validation with independent data

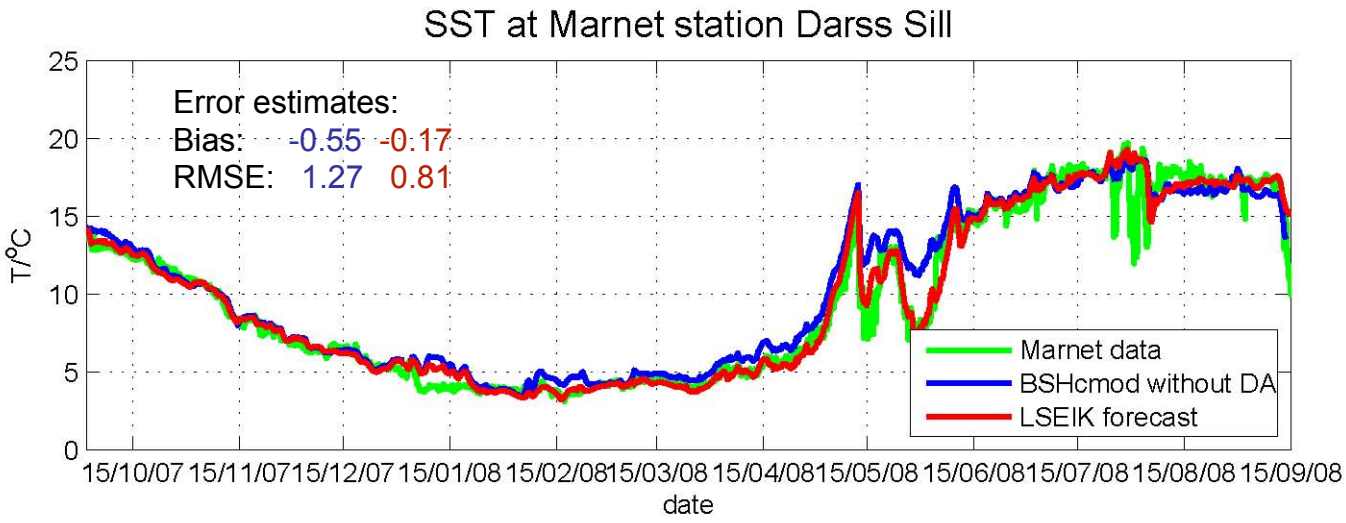
Assimilation of satellite SST data

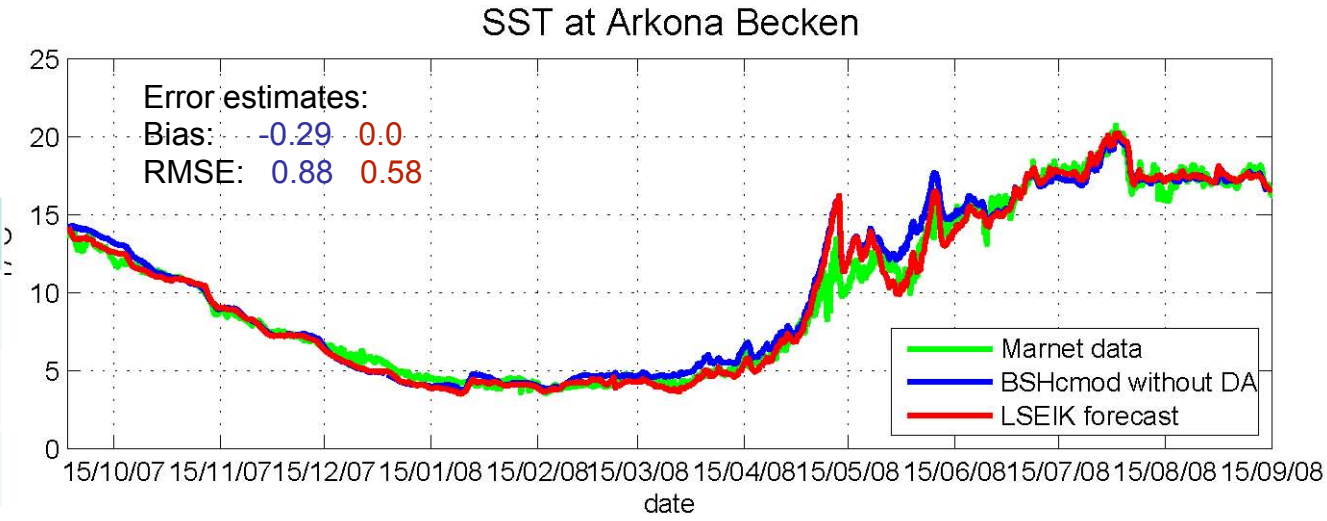
Reduction of
- Bias
- RMS error

In 12-hour forecasts

1 year mean over 6 stations:

	RMSe	bias
free	0.87	0.3
assim	0.55	0.08
data	0.59	0.11





Red: Assimilation 12h forecasts

Some conclusions from case study

- Significant improvement of surface temperature
- No deterioration of unobserved fields
- Very stable forecasts
- Tuning necessary
  (inflation, observation errors, localization radius, observation weights)

- The system was run pre-operationally by BSH
- Current work:
  - Addition of in situ data
  - Examining spatially variable localization
  - Addition of ecosystem model

Thank you!