An Overview on Data Assimilation

Lars Nerger

Alfred Wegener Institute for Polar and Marine Research
Bremerhaven, Germany

and

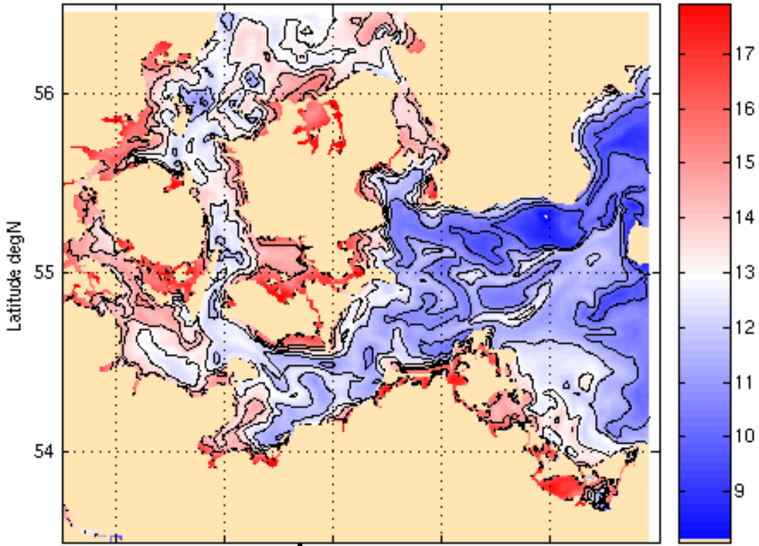
Bremen Supercomputing Competence Center BremHLR

Lars.Nerger@awi.de

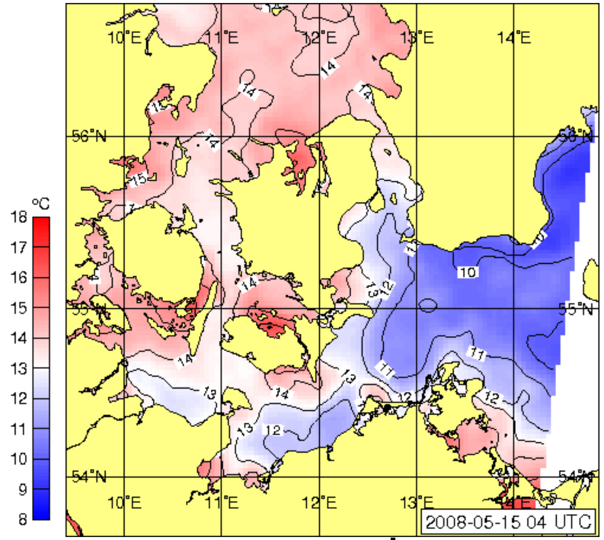
BremHLR
Kompetenzzentrum für Höchstleistungsrechnen Bremen

Concept of Data Assimilation

SST: Simulation (BSHcmod)



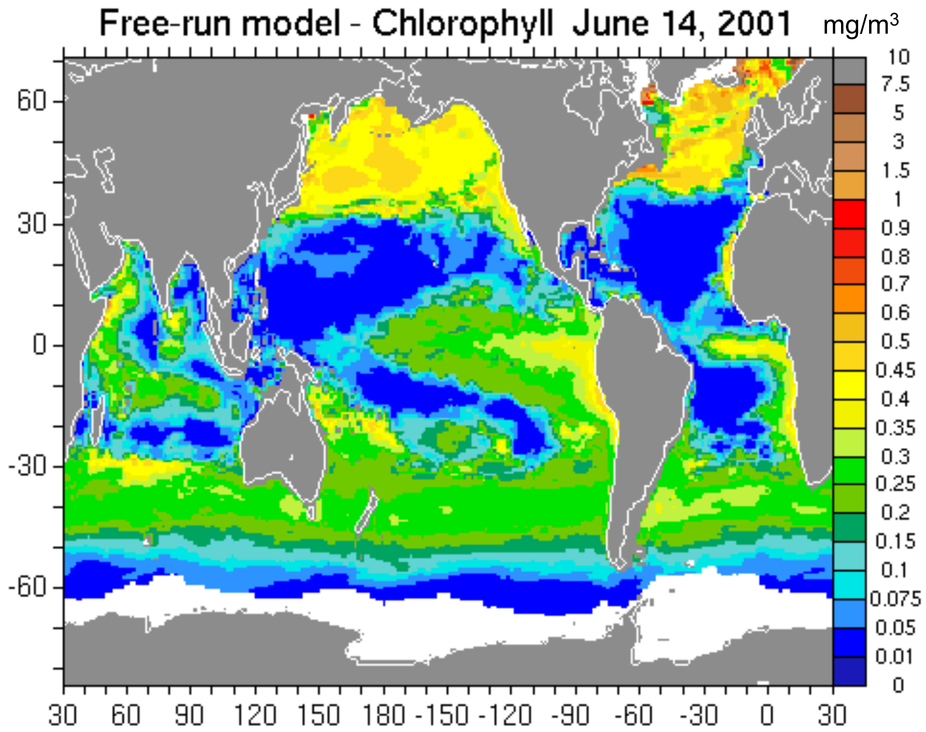
SST: Satellite (AVHRR)

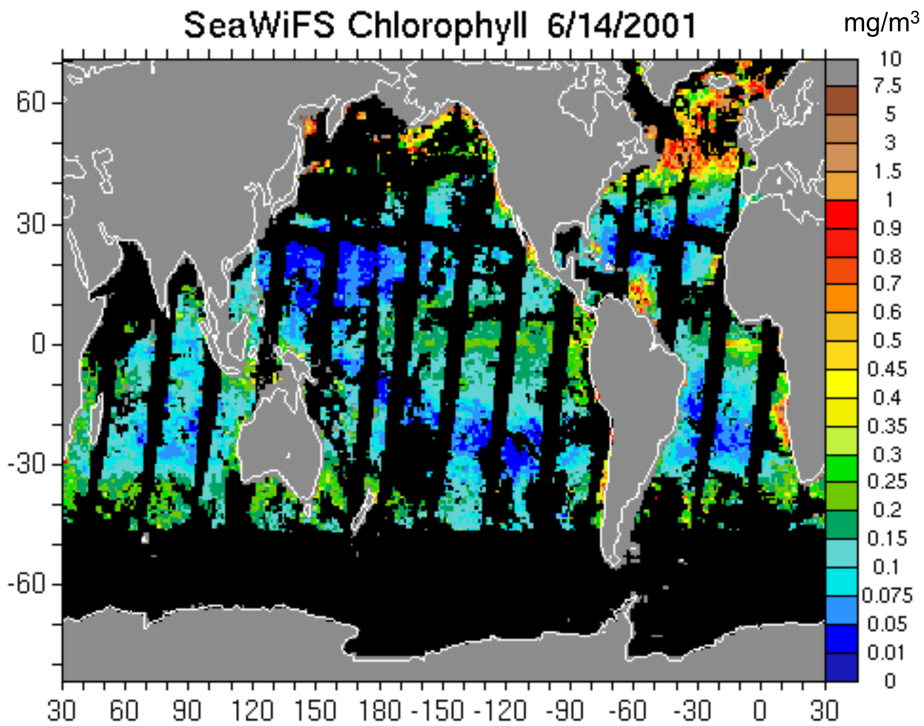


Combination of Information
through Data Assimilation

Improved analysis and forecast
of, for example,
- water temperature
- ice coverage

System Information: Chlorophyll in the ocean





Information: Model

- Generally correct, but has errors
- all fields, fluxes, …

Information: Observation

- Generally correct, but has errors
- sparse information
  (only surface, data gaps, one field)

Combine both sources of information by data assimilation

Overview

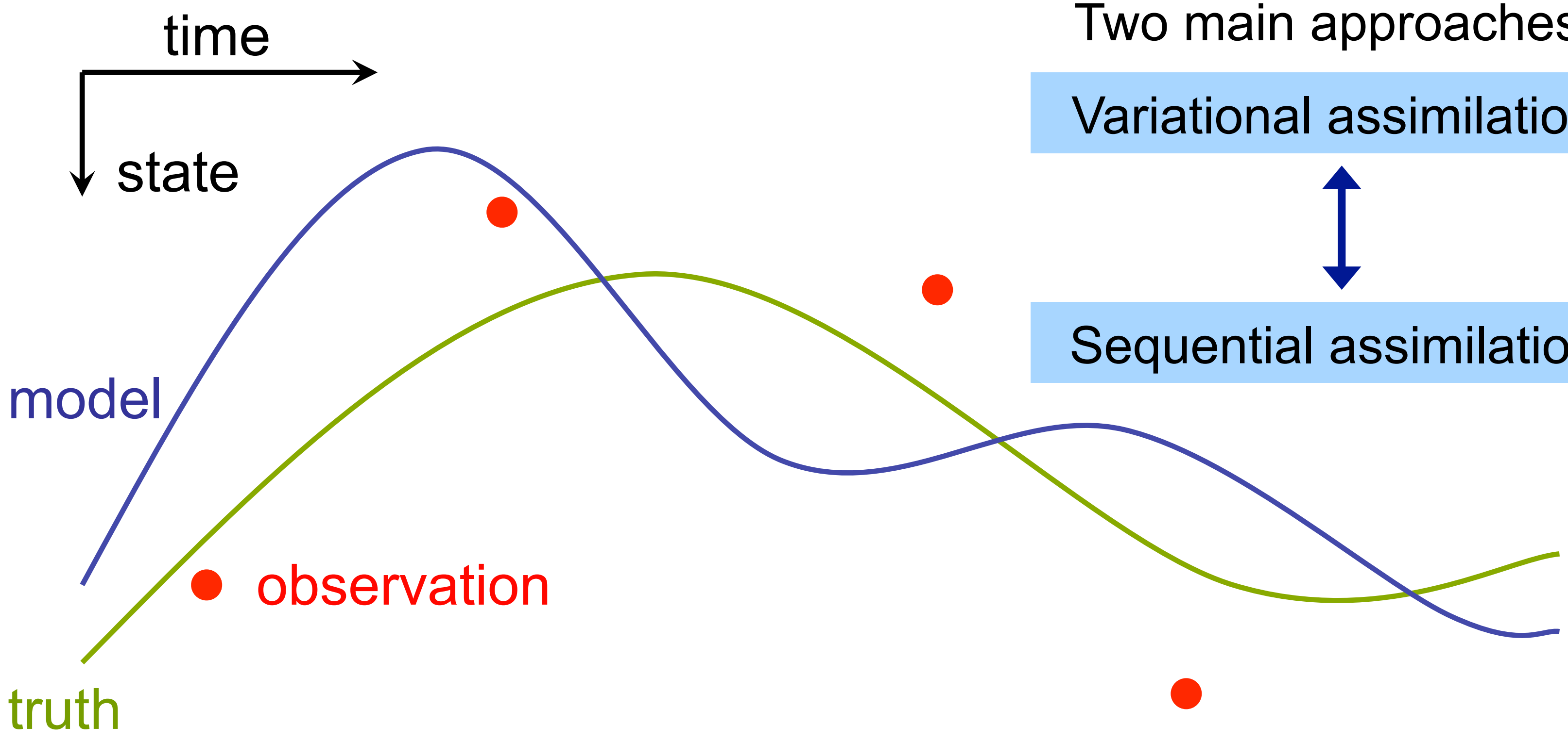
- Data assimilation
- Variational data assimilation
  - 3D-Var, 4D-Var, adjoint method
- Sequential data assimilation
  - Kalman filters
- Ensemble-based Kalman filters
  - SEIK and LSEIK filters

Data Assimilation

- Optimal estimation of system state:
  - initial conditions (for weather/ocean forecasts, …)
  - trajectory (temperature, concentrations, …)
  - parameters (growth of phytoplankton, …)
  - fluxes (heat, primary production, …)
  - boundary conditions and 'forcing' (wind stress, …)
- Characteristics of system:
  - high-dimensional numerical model - $\mathcal{O}(10^7)$
  - sparse observations
  - non-linear

Data Assimilation

Consider some physical system (ocean, atmosphere,…)



Two main approaches:

Variational assimilation

↕

Sequential assimilation

Optimal estimate basically by least-squares fitting

Variational Data Assimilation

3D-Var, 4D-Var, Adjoint method

Variational Data Assimilation - 4D-Var

- Formulate cost function J in terms of "control variable"
  Example: initial state x_0
- Problem:
  Find trajectory (defined by x_0) that minimizes cost J while fulfilling model dynamics
- Use gradient-based algorithm:
  - e.g. quasi-Newton
  - Gradient for $J[x_0]$ is computed using adjoint of tangent linear model operator
  - The art is to formulate the adjoint model and weights in J (No closed formulation of model operator)
  - Iterative procedure (local in control space)
- 3D-Var: optimize locally in time

Adjoint method - 4D-Var algorithm

1. Initialization: Choose initial estimate of x_0

2. Forward: Integrate model
start from x_0; store trajectory

3. Compute cost function;
exit, if cost is below limit

4. Backward: Integrate adjoint model backward in time
start from final residual (y-x); use trajectory from 2.

5. Optimizer: Update x_0
with optimization algorithm

Serial operation; difficult to parallelize

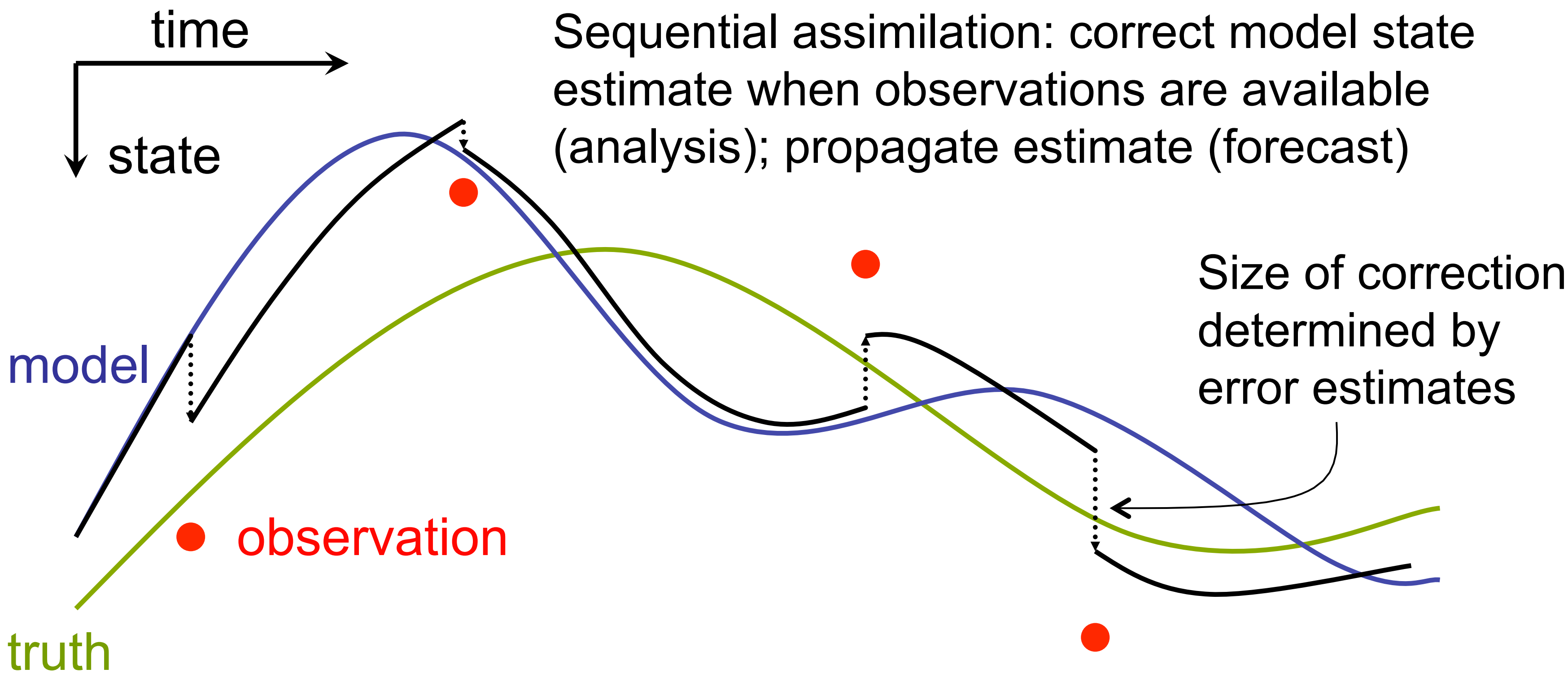
Sequential Data Assimilation

Kalman filters

Sequential Data Assimilation

Consider some physical system (ocean, atmosphere,…)

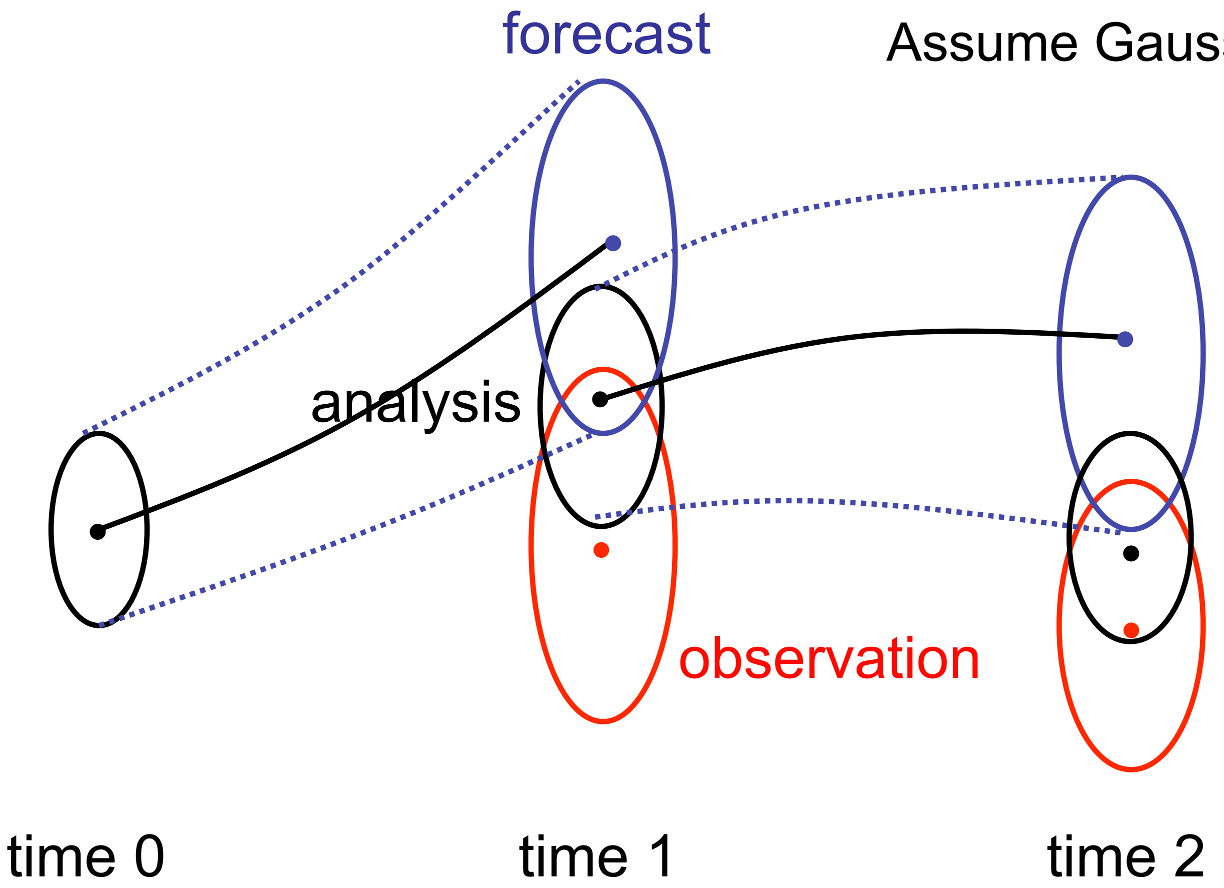
Sequential assimilation: correct model state estimate when observations are available (analysis); propagate estimate (forecast)



Probabilistic view: Optimal estimation

Consider probability distribution of model and observations

Kalman Filter:
Assume Gaussian distributions



Gaussianity

- Assumed by all KF-based filters (for optimal minimum-variance estimate)
  - ⇨ Gaussian forecast probability distribution
  - ⇨ Observation errors Gaussian distributed
- Analysis is combination of two Gaussian distributions
- Estimation problem can be formulated in terms of means and covariance matrices of probability distributions
- Cost function J is consistent with Gaussian assumptions

But: Nonlinearity will not conserve Gaussianity!

(Extended KF conserves Gaussianity by first-order approximation, but can be unstable)

More issues … application side

- Storage of covariance matrix can be unfeasible
- Evolution of covariance matrix extremely costly
- Linearized evolution (like in Extended KF) can be unstable

⇨ Reduce cost

  - simplify dynamics
  - approximate state covariance matrix

Ensemble-based Kalman filters

Ensemble-based Kalman Filters

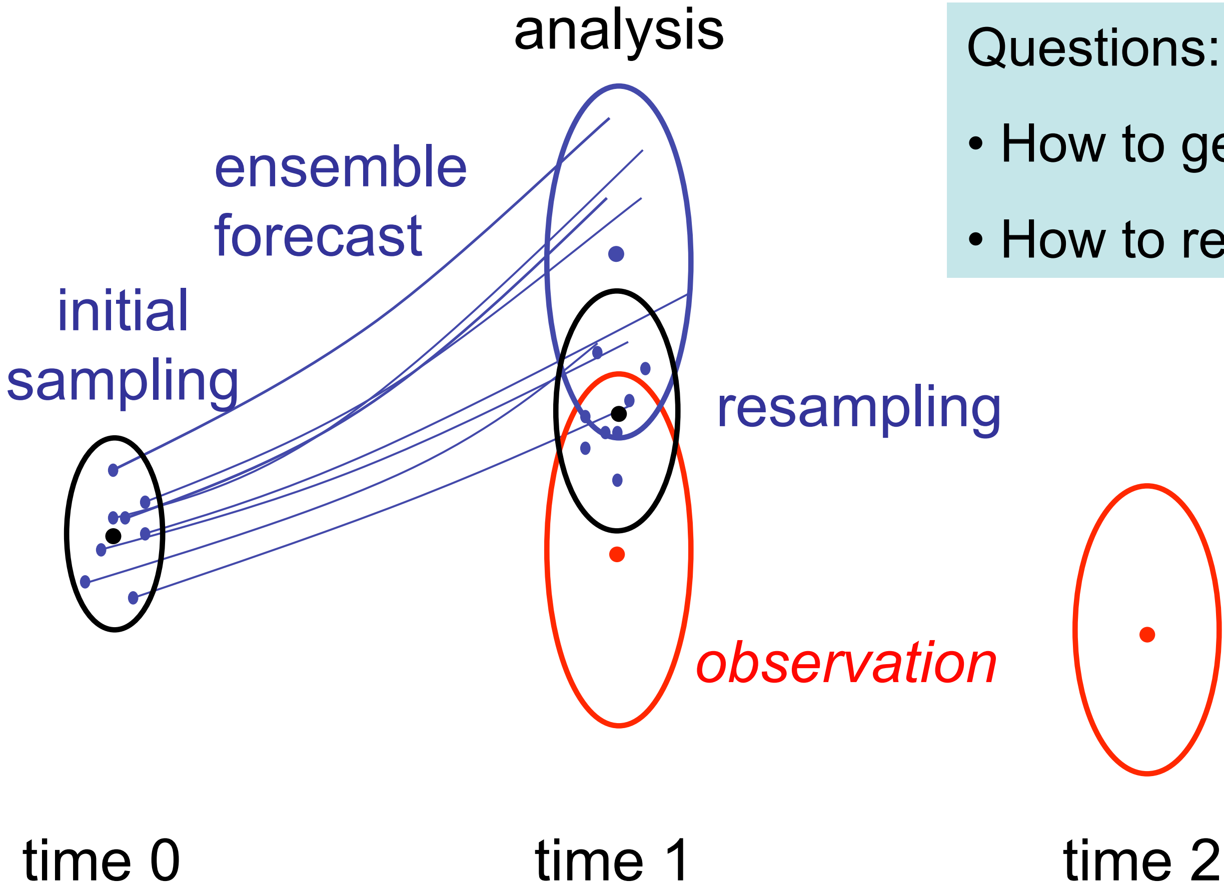
- Foundation: Kalman filter (Kalman, 1960)
  - optimal estimation problem
  - express problem in terms of state estimate **x** and error covariance matrix **P** (Gaussian distributions)
  - propagate matrix **P** by linear (linearized) model
  - variance-minimizing analysis

- Ensemble-based Kalman filter:
  - sample state **x** and covariance matrix **P** by ensemble of model states
  - propagate **x** and **P** by integration of ensemble states
  - Apply linear analysis of Kalman filter

First filter in oceanography: “Ensemble Kalman Filter” (Evensen, 1994), second: SEIK (Pham, 1998)

Ensemble-based Kalman Filter

Approximate probability distributions by ensembles



Questions:

- How to generate initial ensemble?
- How to resample after analysis?

Please note:

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

„The“ Ensemble Kalman Filter - EnKF (Evensen, 1994)

Initialization: Sample state **x** and covariance matrix **P** by Monte-Carlo ensemble of model states

↓

Forecast: Evolve each of the ensemble members with the full non-linear stochastic model

↓

Analysis: Apply EKF update step to each ensemble member with observation from an observation ensemble. Covariance matrix approx. by ensemble statistics, state estimate by ensemble mean.

Error Subspace Algorithms

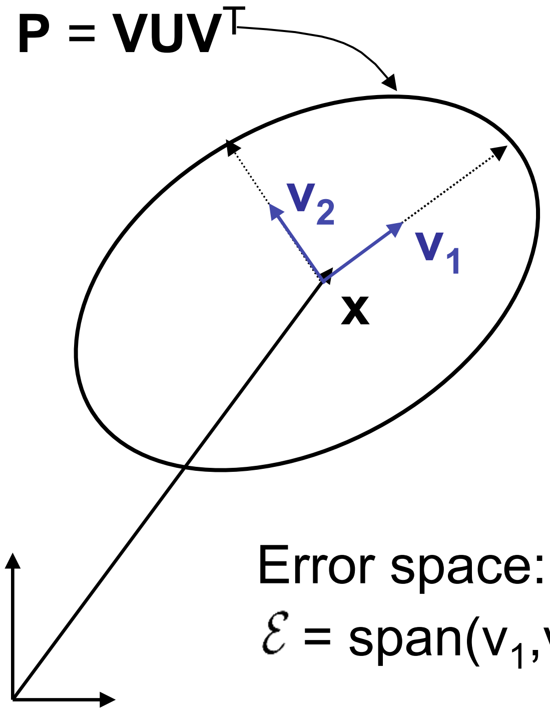
⇨ Approximate state covariance matrix by low-rank matrix

⇨ Keep matrix in decomposed form ($\mathbf{X}\mathbf{X}^T$, $\mathbf{V}\mathbf{U}\mathbf{V}^T$)

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($\mathbf{P}$)

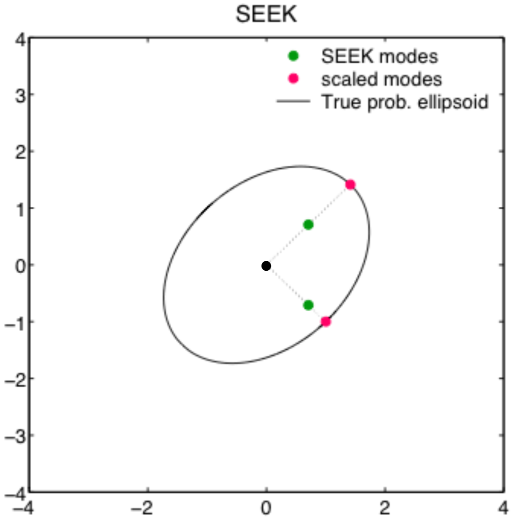


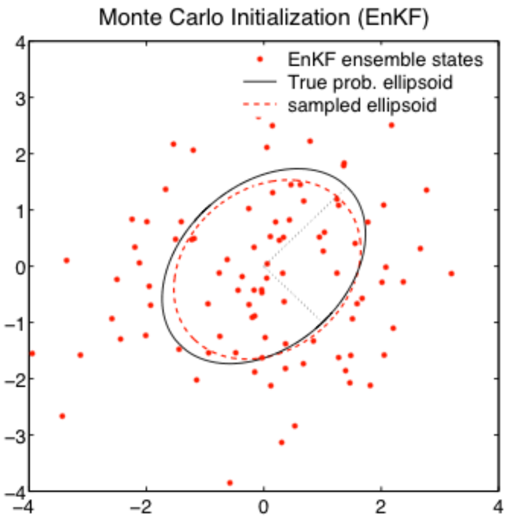
Error space:

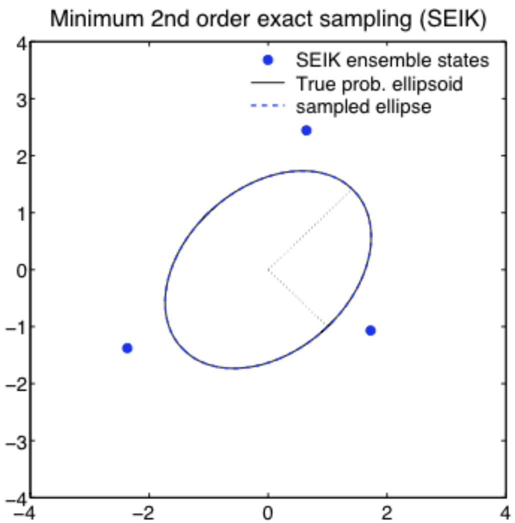
$\mathcal{E} = \mathrm{span}(v_1, v_2, \ldots)$

Sampling Example

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







More ensemble-based/error-subspace Kalman filters

- A little “zoo” (not complete):

EnKF(94/98)

SEIK

EAKF ETKF EnSQRTKF EnKF(2003) EnKF(2004)

SEEK RRSQRT ROEK

MLEF SPKF ESSE

(Properties and differences are hardly understood)

Computational Aspects

- Ensemble integration can be easily parallelized
- Filter algorithms can be implemented independently from model
- Observations need information about the fields and the location of data

- Motivation for PDAF (Parallel Data Assimilation Framework)
  - Software framework (Fortran) to simplify implementation of data assimilation systems based on existing models
  - Provide parallelization support for ensemble forecasts
  - Provide parallelized and optimized filter algorithms
  - Open source: **http://pdaf.awi.de**

The SEIK filter

The SEIK* filter (Pham, 1998)

- Use factorization of covariance matrix $\mathbf{P} = \mathbf{VUV}^T$ (singular value decomposition)
- Approximate **P** by truncation to leading singular values (low rank r « state dimension n)
- *Forecast*: Use ensemble of minimum size N = r+1
- *Analysis*:
  - Regular KF update of state estimate **x**
  - Update **P** by updating **U**
- *Re-initialization*: Transform ensemble states to represent new **x** and **P**

*****_Singular “Evolutive” Interpolated Kalman_

The SEIK filter (Pham, 1998) - differences from EnKF

Initialization: Approximate covariance matrix by low-rank matrix in the form **$P=VUV^T$**. Generate ensemble of minimum size exactly representing error statistics.

Forecast: Evolve each of the ensemble members with the full non-linear stochastic model.

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

Re-Initialization: Transform state ensemble to exactly represent updated error statistics.

Overall: A more efficient ensemble-based Kalman filter

The SEIK filter - Properties

- Computational complexity
  - linear in dimension of state vector
  - approx. linear in dimension of observation vector
  - cubic with ensemble size
- Low complexity due to explicit consideration of error subspace:

  ⇨ Degrees of freedom given by ensemble size -1

  ⇨ Analysis increment: combination of ensemble members with weight computed in error subspace
- Simple application to non-linear models due to ensemble forecasts (e.g. no linearized or adjoint models)
  ⇨but not "optimal"
- Equivalent of ETKF under particular conditions

Issues of ensemble-based/error-subspace KFs

- No filter works without tuning
  - ⇨ forgetting factor/covariance inflation
  - ⇨ localization
- Other issues
  - ⇨ Optimal initialization unknown (is it important?)
  - ⇨ ensemble integration still costly
  - ⇨ Simulating model error
  - ⇨ Nonlinearity
  - ⇨ Non-Gaussian fields or observations
  - ⇨ Bias (model and observations)
  - ⇨ …

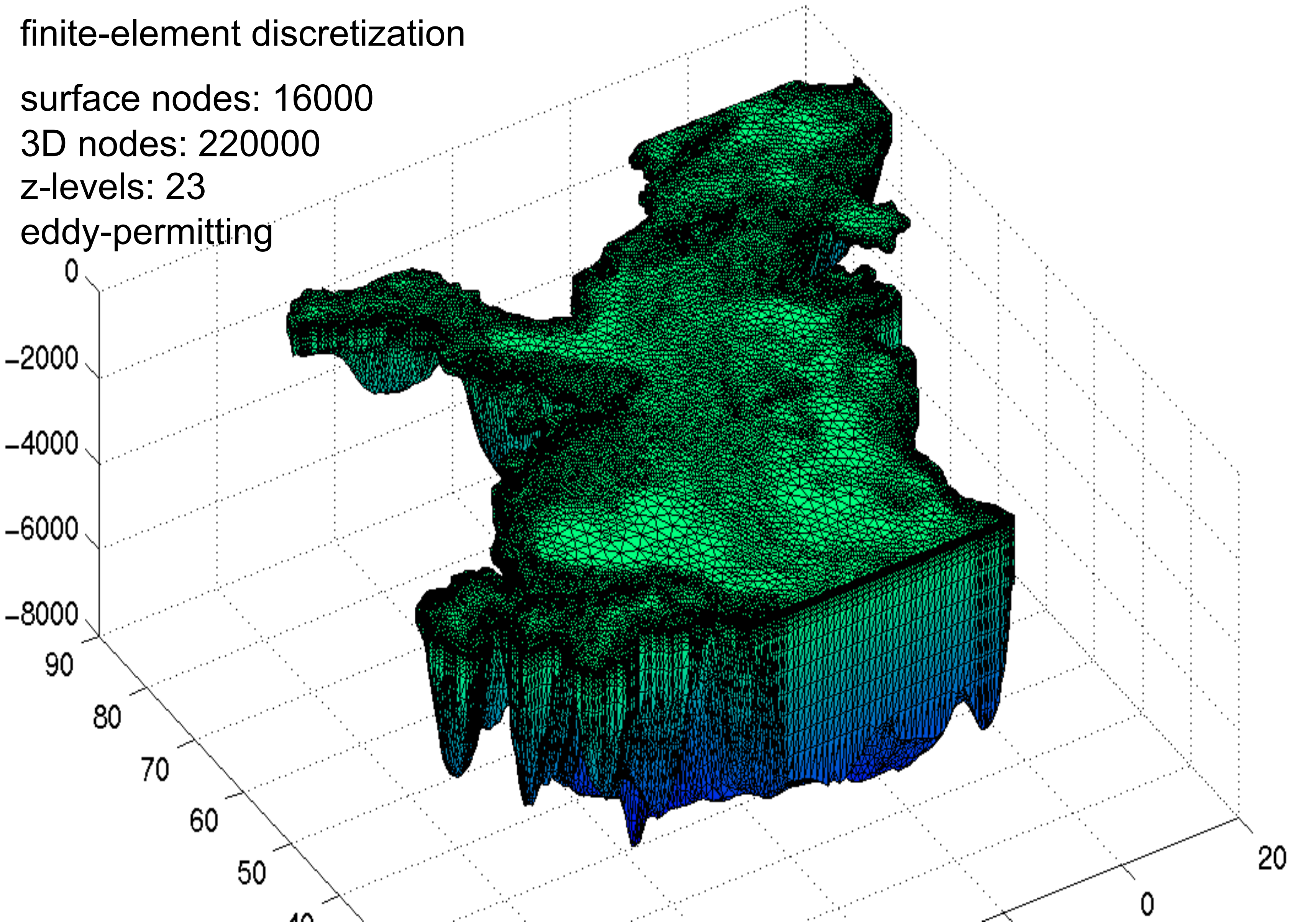
Example:

Assimilation of pseudo sea surface height observations in the North Atlantic

FEOM – Mesh for North Atlantic

finite-element discretization

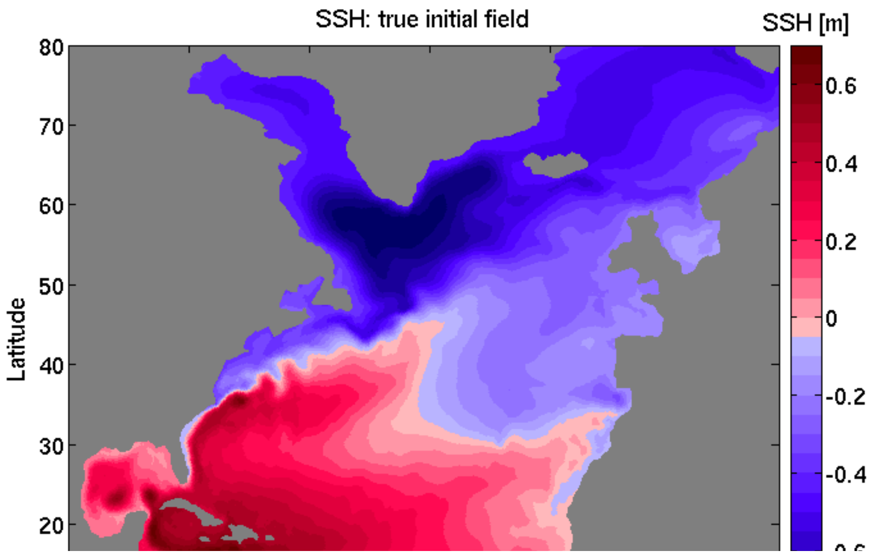
surface nodes: 16000
3D nodes: 220000
z-levels: 23
eddy-permitting

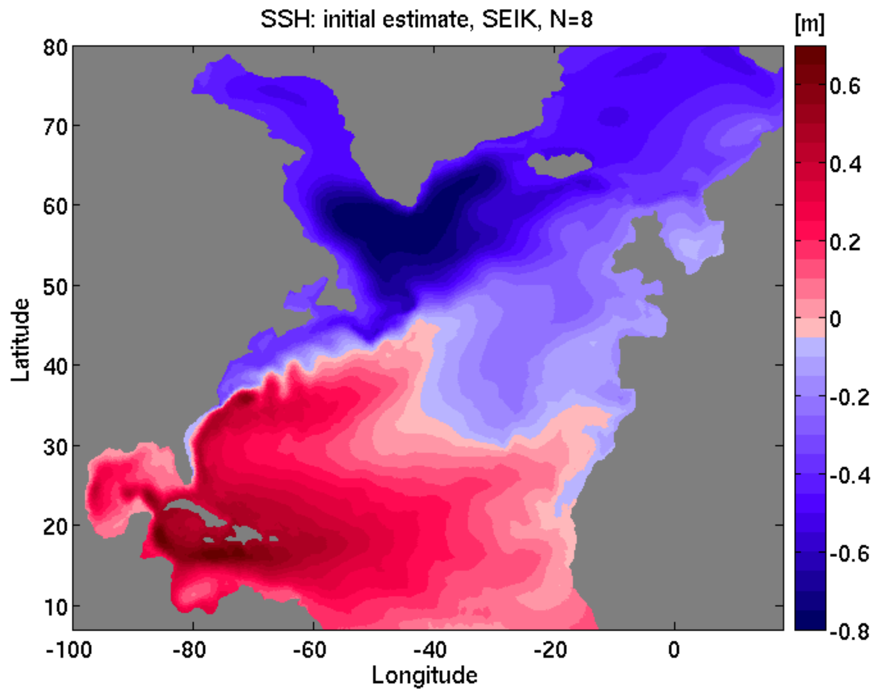


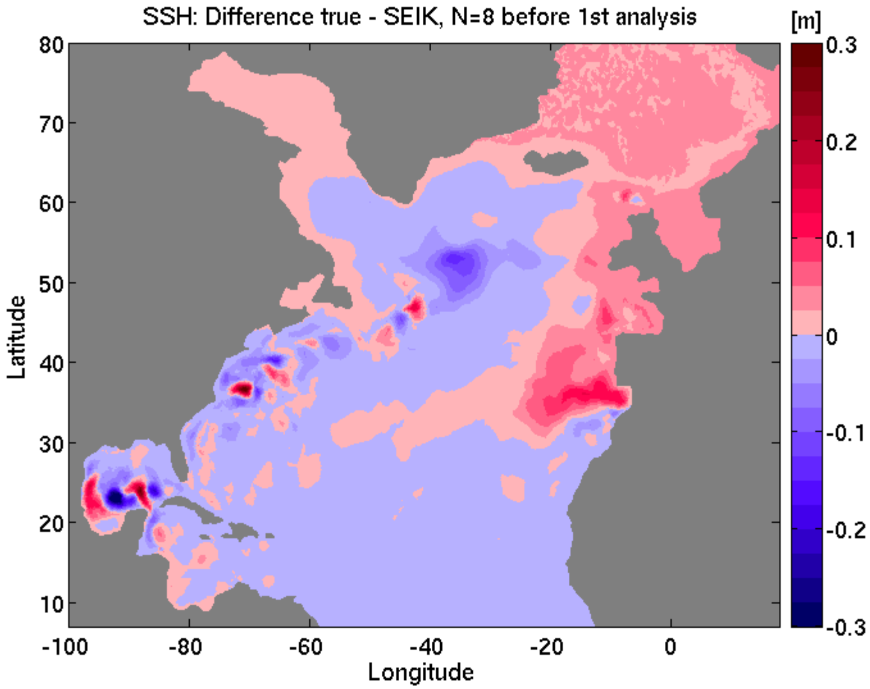
Configuration of twin experiments

- 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

Modeled Sea Surface Height (Dec. 1992)

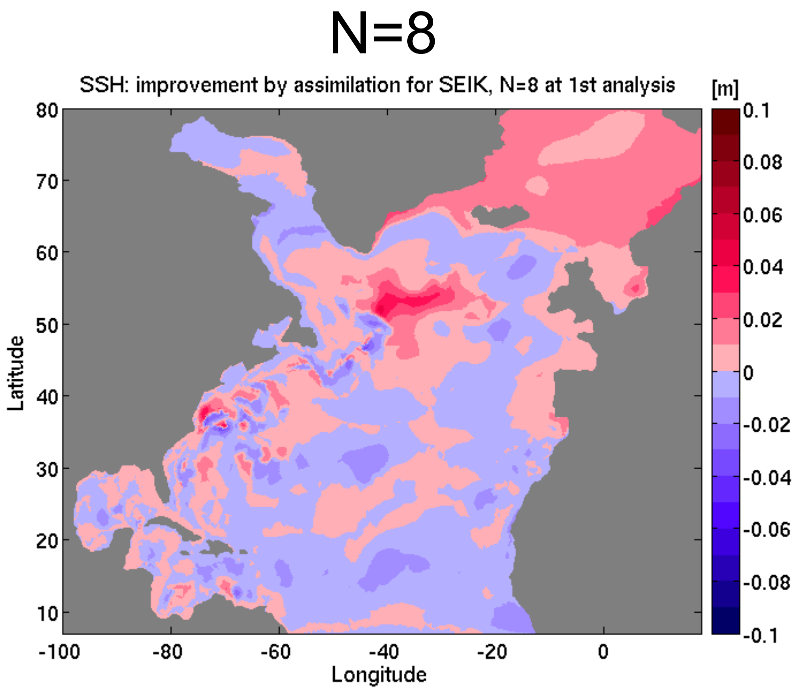


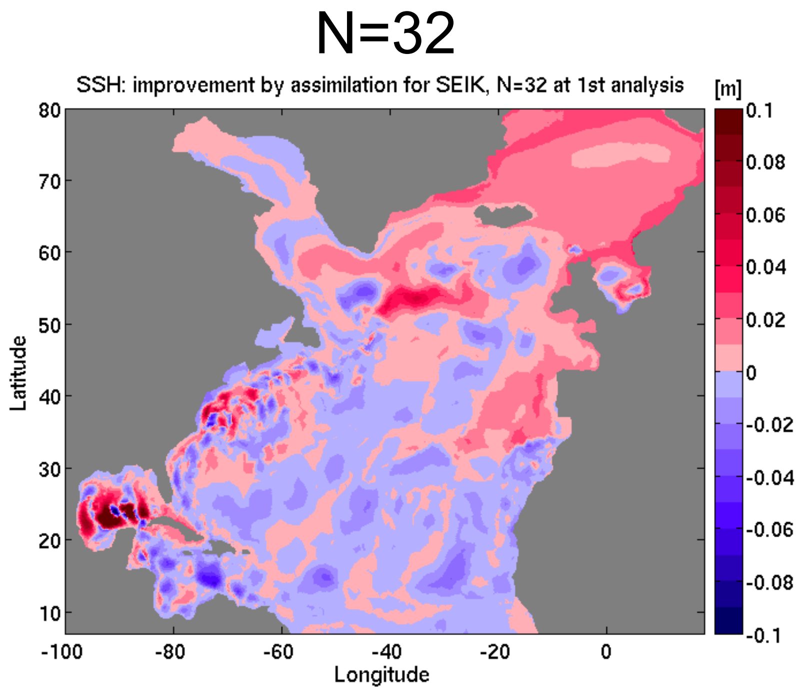




- large-scale deviations of small amplitude
- small-scale deviations up to 40 cm

Improvement of Sea Surface Height (Dec. 1992)





• Improvement: red - deterioration: blue

$\Rightarrow$ For N=8 rather coarse-scale corrections

$\Rightarrow$ Increased ensemble size adds finer scales (systematically)

Localization - LSEIK

Global SEIK filter - filtering behavior

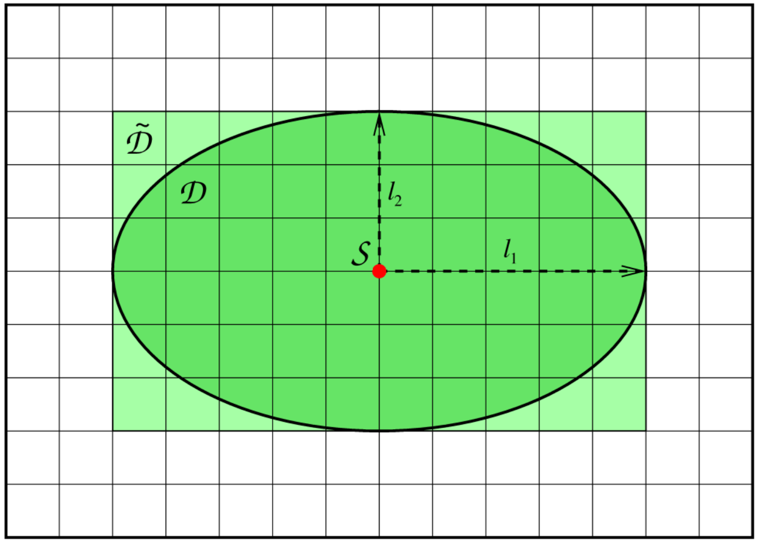
- SEIK performs global optimization
- Degrees of freedom is small (ensemble size - 1)

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)

Local SEIK filter

- Analysis:
  - Update small regions (e.g. single water columns)
  - Consider only observations within cut-off distance
    - ➢ neglects long-range correlations



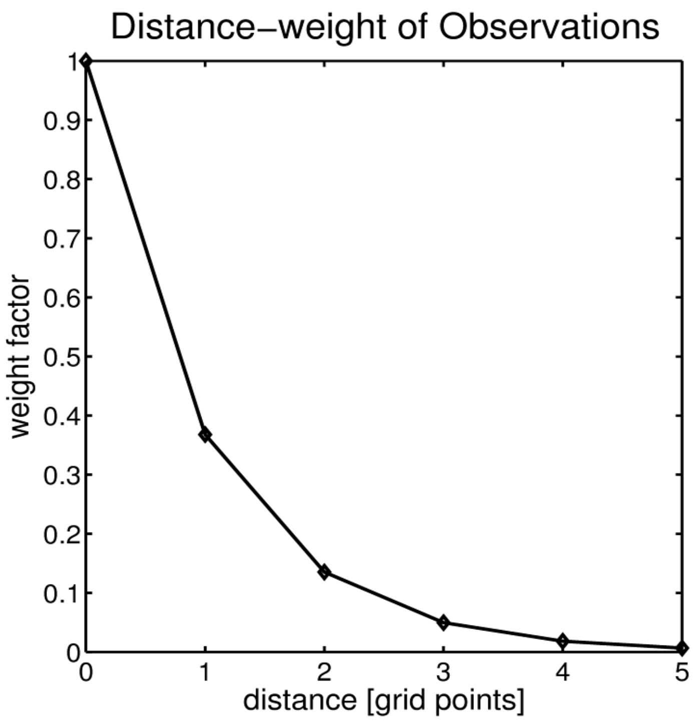
- Re-Initialization:
  - Transform local ensemble
  - Use same transformation matrix in each local domain

Nerger, L., S. Danilov, W. Hiller, and J. Schröter. Ocean Dynamics 56 (2006) 634

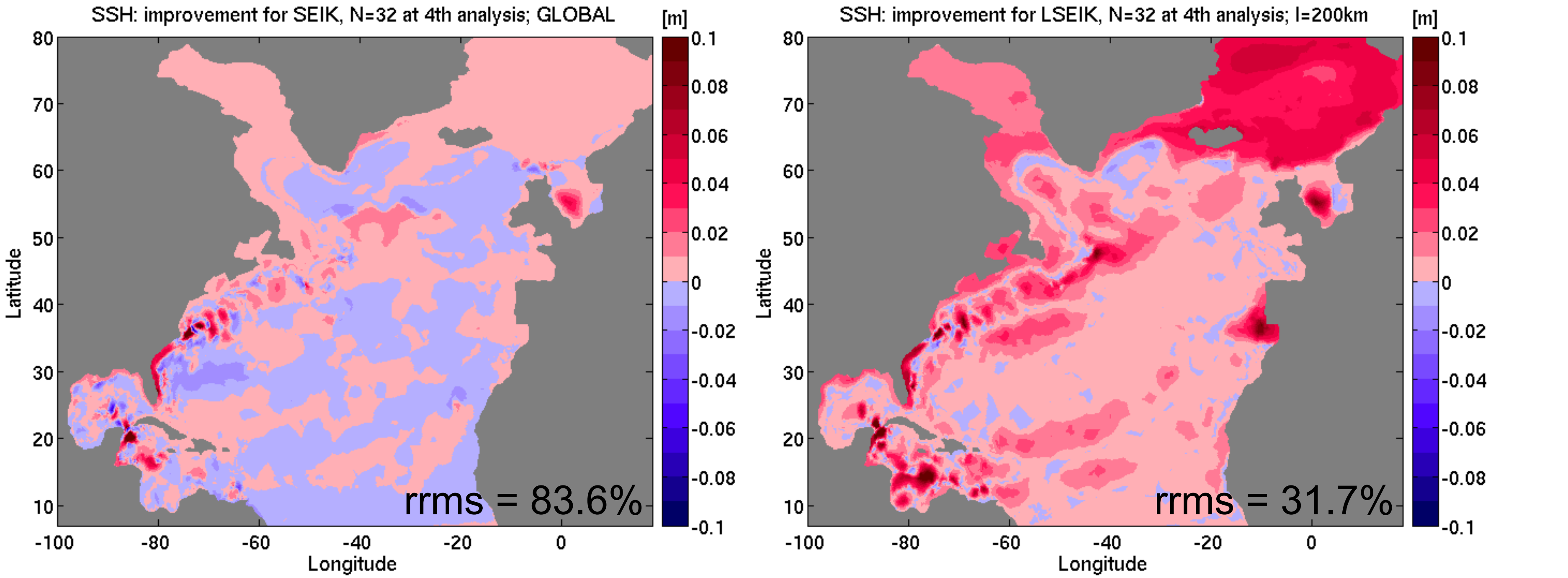
Local SEIK filter II

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



Global vs. Local SEIK, N=32 (Mar. 1993)

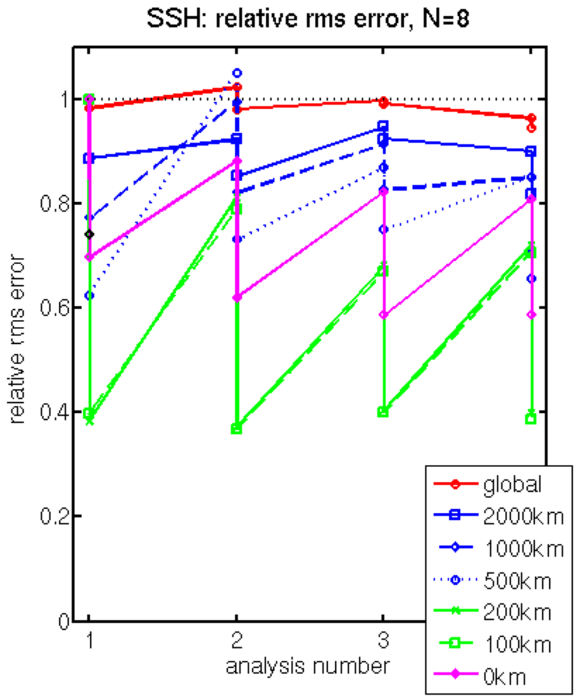


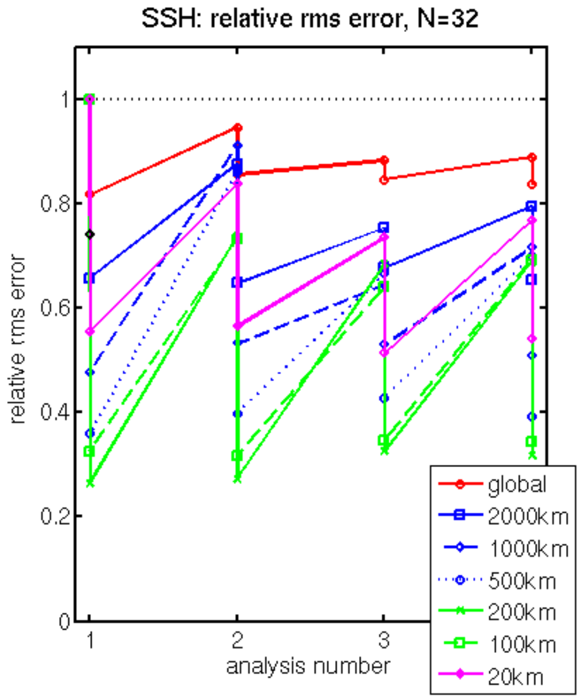
- Improvement regions of global SEIK also improved by local SEIK

- localization provides improvements in regions not improved by global SEIK

- regions with error increase diminished for local SEIK

Relative rms errors for SSH





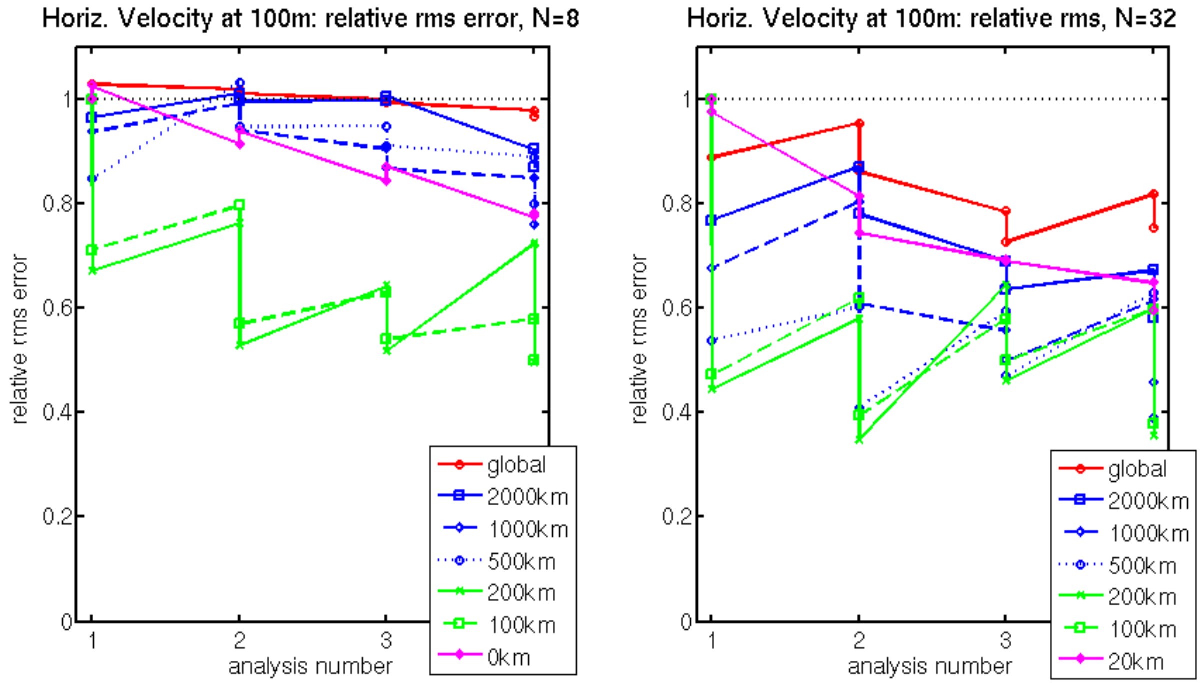
- global filter: significant improvement for larger ensemble

- 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

Effect of assimilation on non-observed fields



- velocity field updated via cross-correlations
- localization improves estimates
- minimum errors for 100km (N=8), 200km (N=32)
- special behavor for total localization (l=0km): overfitting

Local SEIK filter - findings

- 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)

$\Rightarrow$ possible trade off between improved analysis update and forecast error-increase

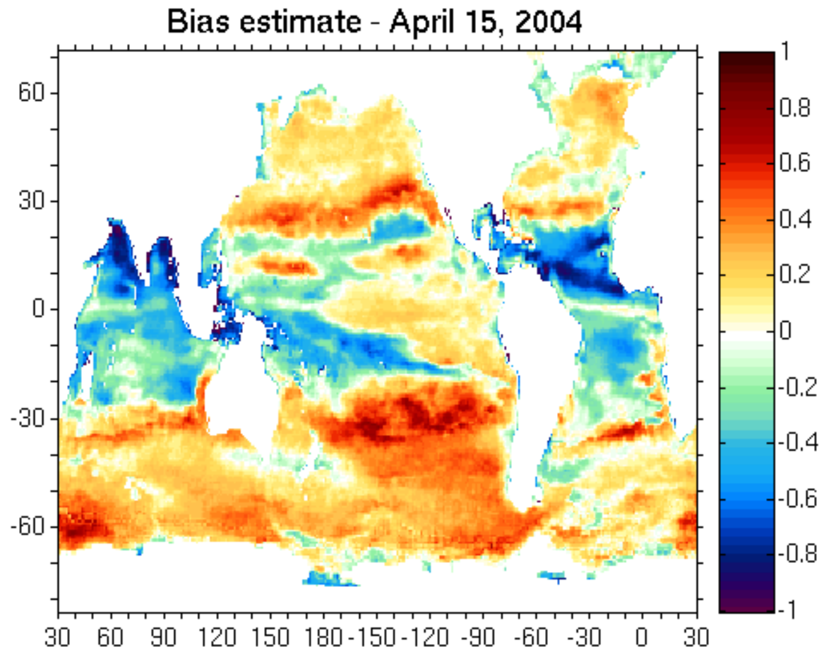
- LSEIK is more costly than global SEIK, but computationally still efficient

Bias Estimation

Bias Estimation

- un-biased system:
  fluctuation around true state

- biased system:
  systematic over- and underestimation
  (common situation with real data)



- 2-stage bias online bias correction

  1. Estimate bias
     (using fraction of covariance matrix used in 2.)

  2. Estimate de-biased state

- Forecast

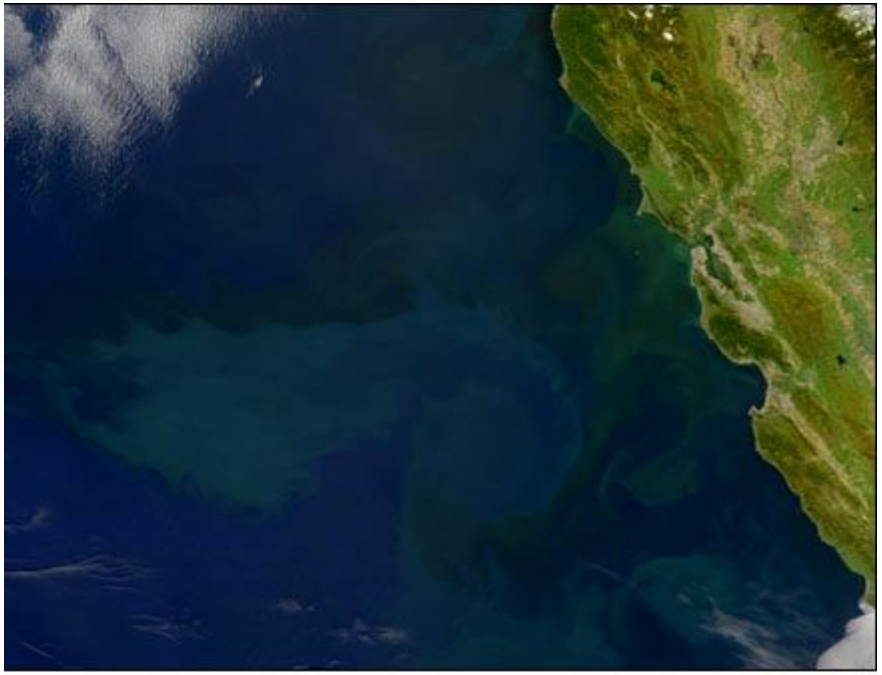
  1. forecast ensemble of biased states

  2. no propagation of bias vector

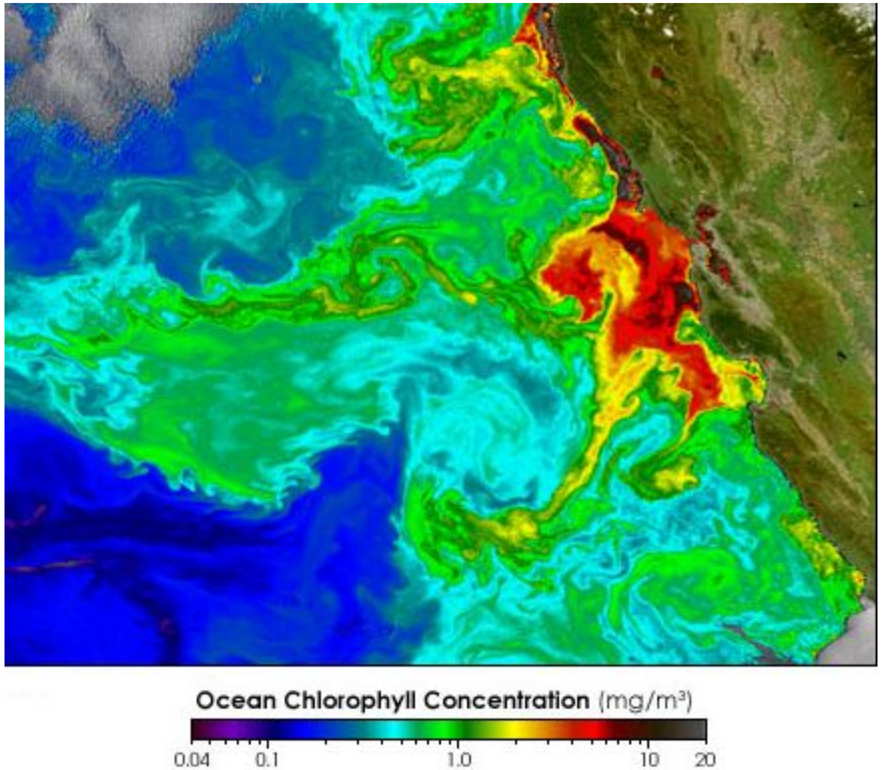
Nerger, L., and W.W. Gregg. J. Marine Systems, 73 (2008) 87-102

Satellite Ocean Color (Chlorophyll) Observations

Natural Color 3/16/2004

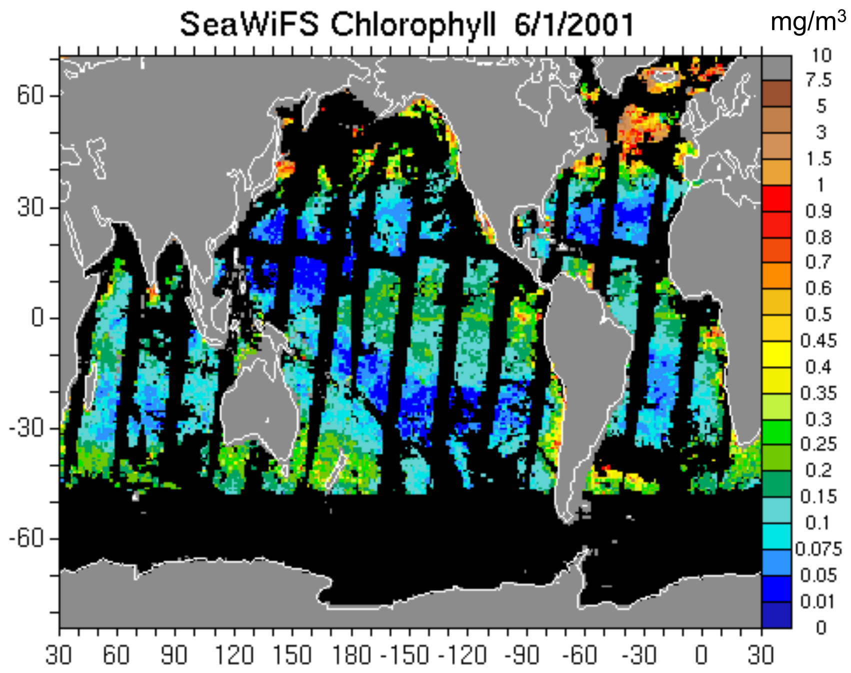


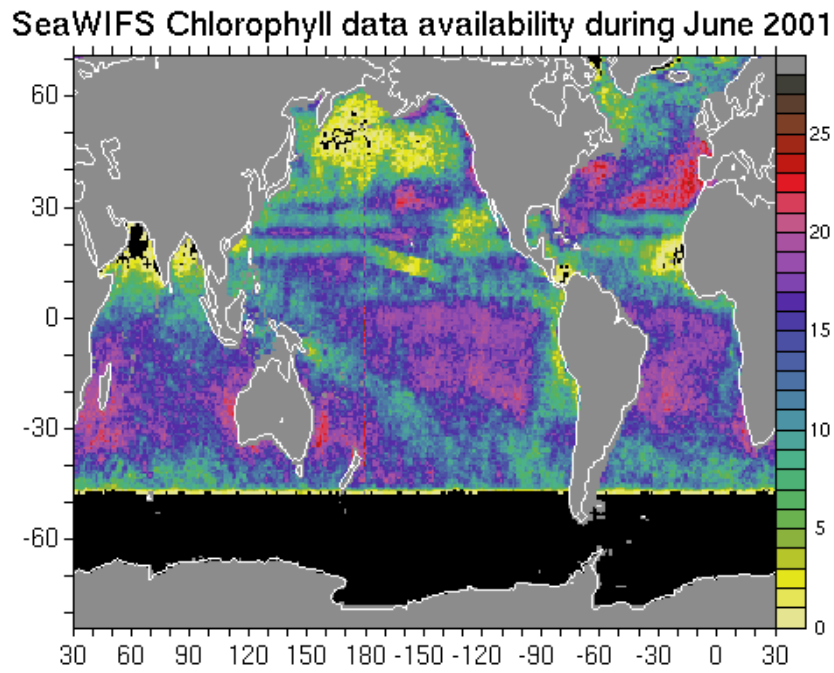
Chlorophyll Concentrations



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

Assimilated Observations

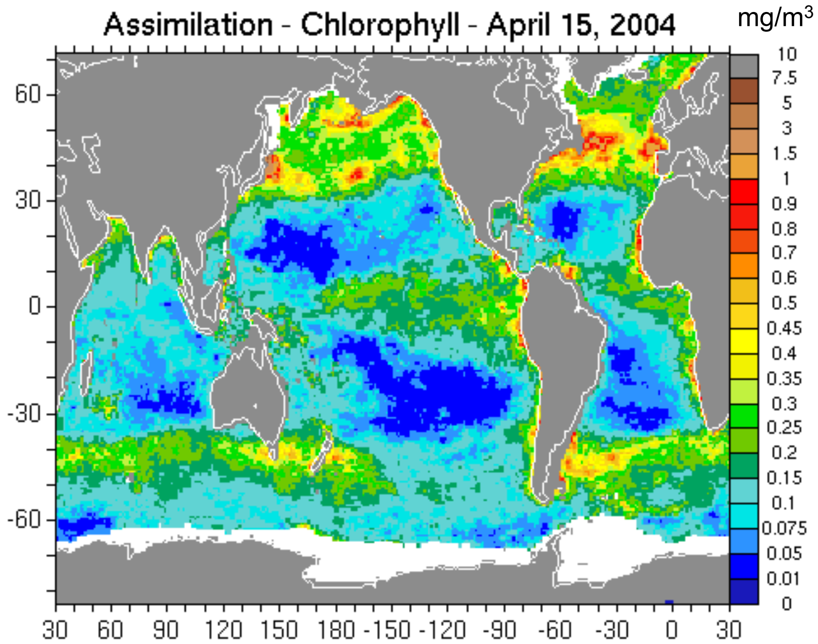


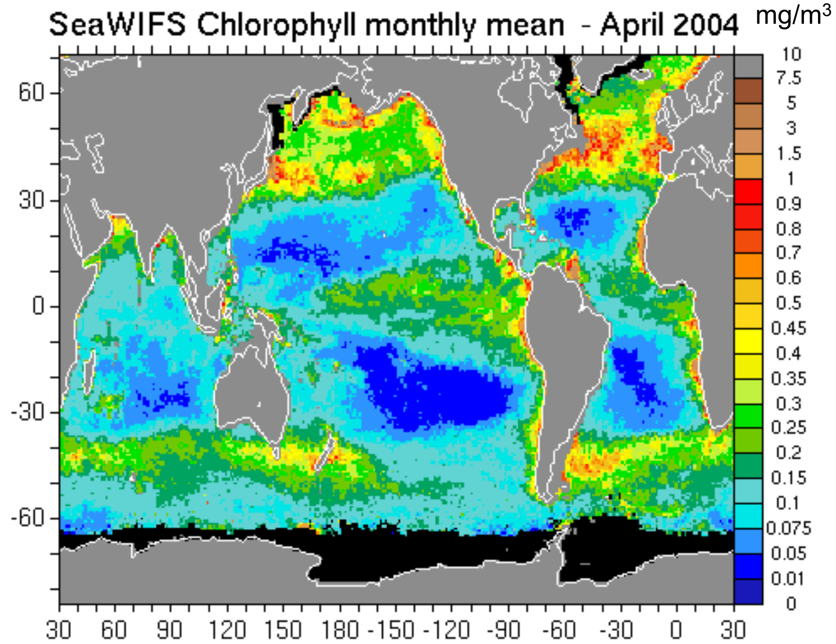


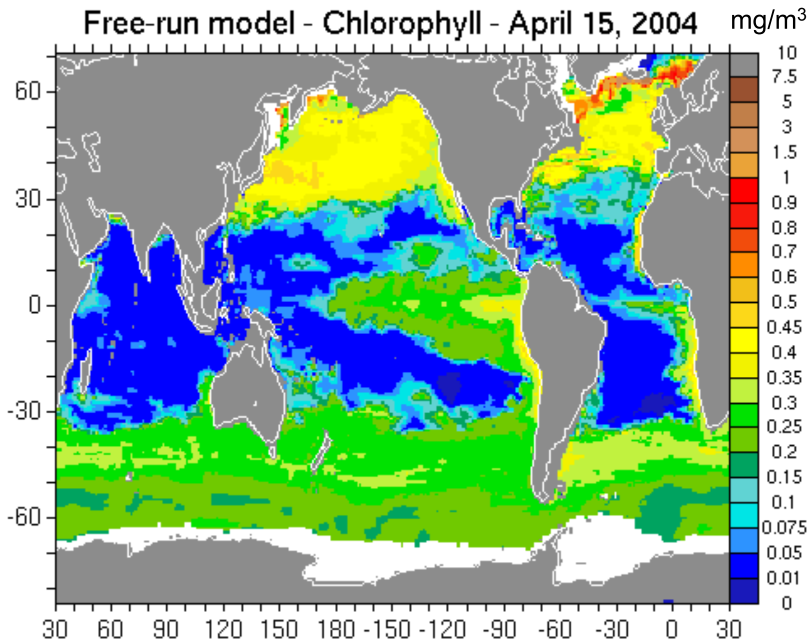
- Daily gridded SeaWiFS chlorophyll data
  - ➢ gaps: satellite track, clouds, polar nights
  - ➢ ~13,000-18,000 data points daily (of 41,000 wet grid points)
  - ➢ irregular data availability

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

Estimated Chlorophyll - April 15, 2004

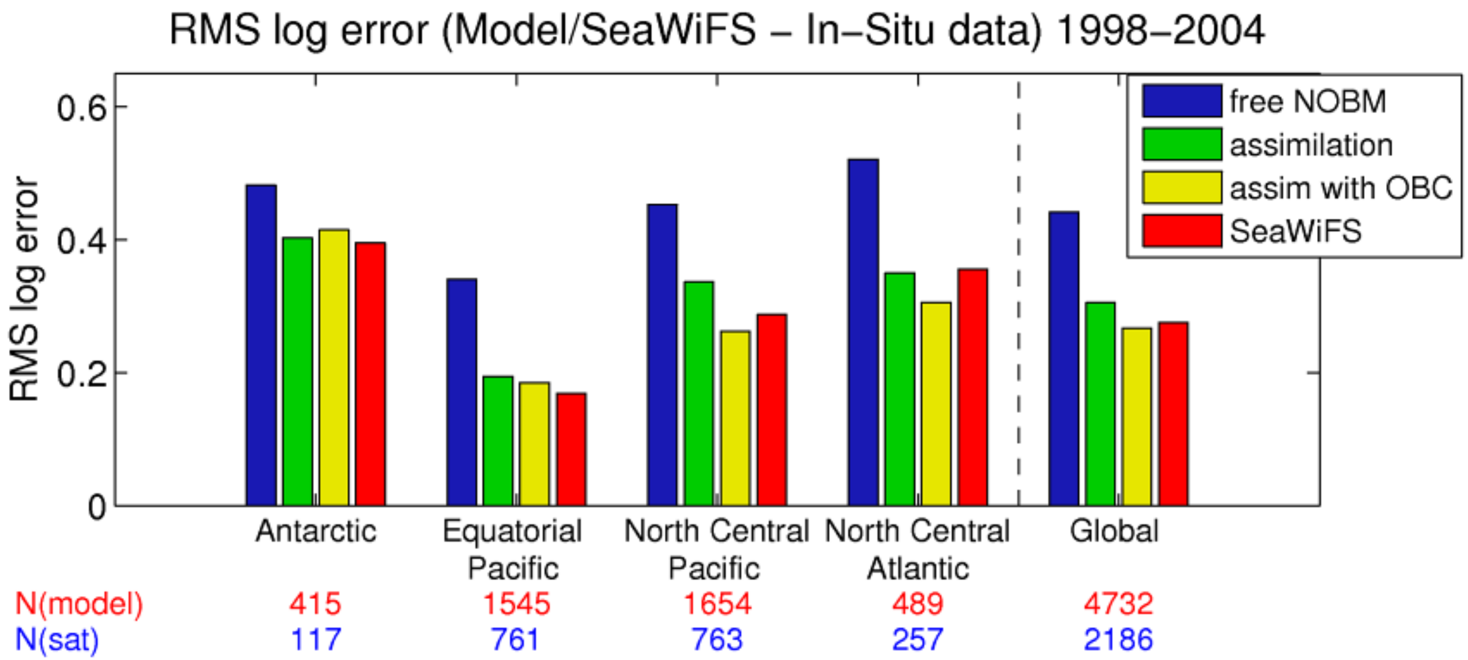






- strongly improved surface Chlorophyll estimate
- intended deviations (Arabian Sea, Congo, Amazon)
- other deviations in high-Chlorophyll regions

Comparison with independent data



- In situ data from SeaBASS/NODC over 1998-2004 (shown basins include about 87% of data)
- Independent from SeaWiFS data (only used for verification of algorithms)
- Compare daily co-located data points

⇒ Assimilation in most regions below SeaWiFS error

⇒ Bias correction improves almost all basins

Summary

- Data assimilation combines information from models and observations to generate improve estimates of the system.
- Ensemble-based Kalman filters are efficient assimilation methods. To some extent they can handle nonlinearity.
- Current assimilation algorithms require tuning
- There are various open issues regarding optimal application of assimilation algorithms.

Thank you!