First Institute of Oceanography, Qingdao, China, November 13, 2015

Ensemble Data Assimilation with the Parallel Data Assimilation Framework

Lars Nerger

Alfred Wegener Institute
Helmholtz Center 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

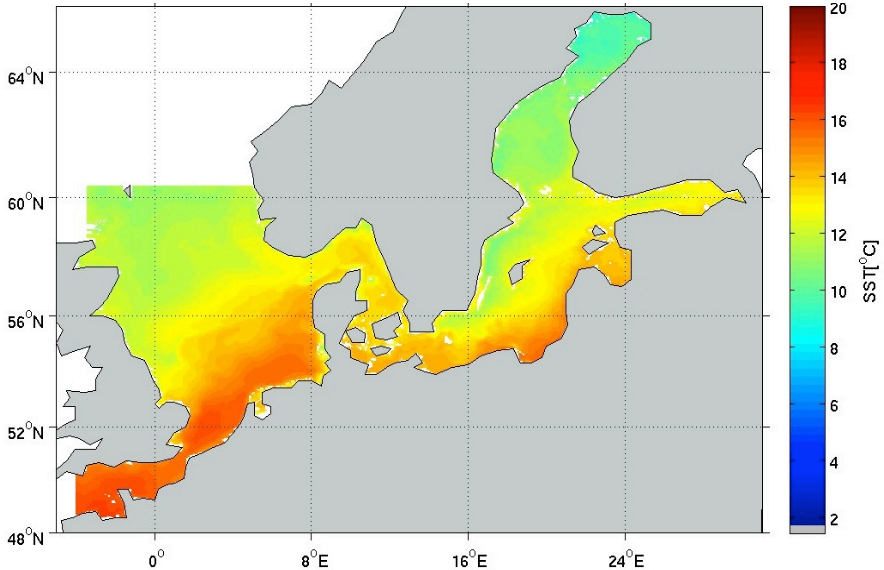
Outline

- Ensemble-based Kalman filters
- Implementation aspects
- PDAF – Parallel Data Assimilation Framework
- Application example

Motivation

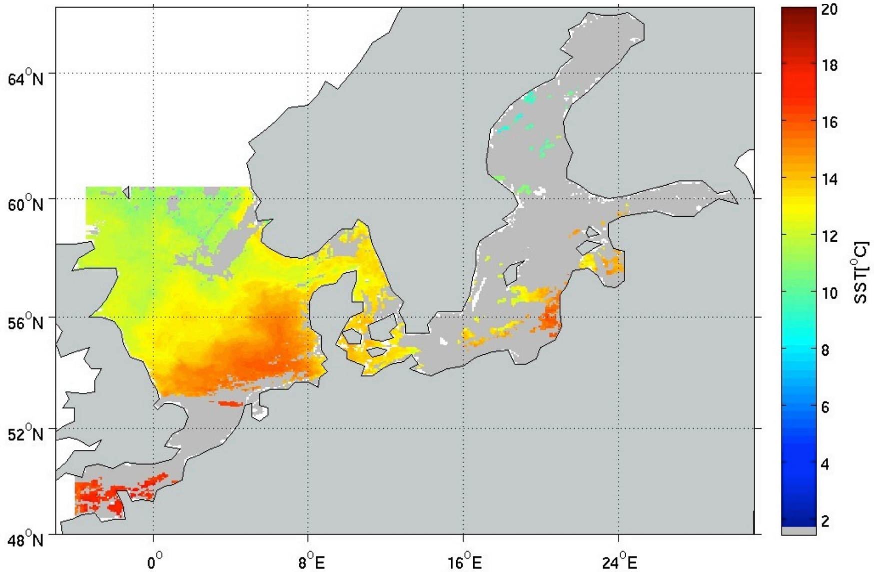
DeMarine

Model surface temperature



Information: Model

Satellite surface temperature



Information: Observations

Combine both sources of information

quantitatively by computer algorithm

➜ data assimilation

AWI

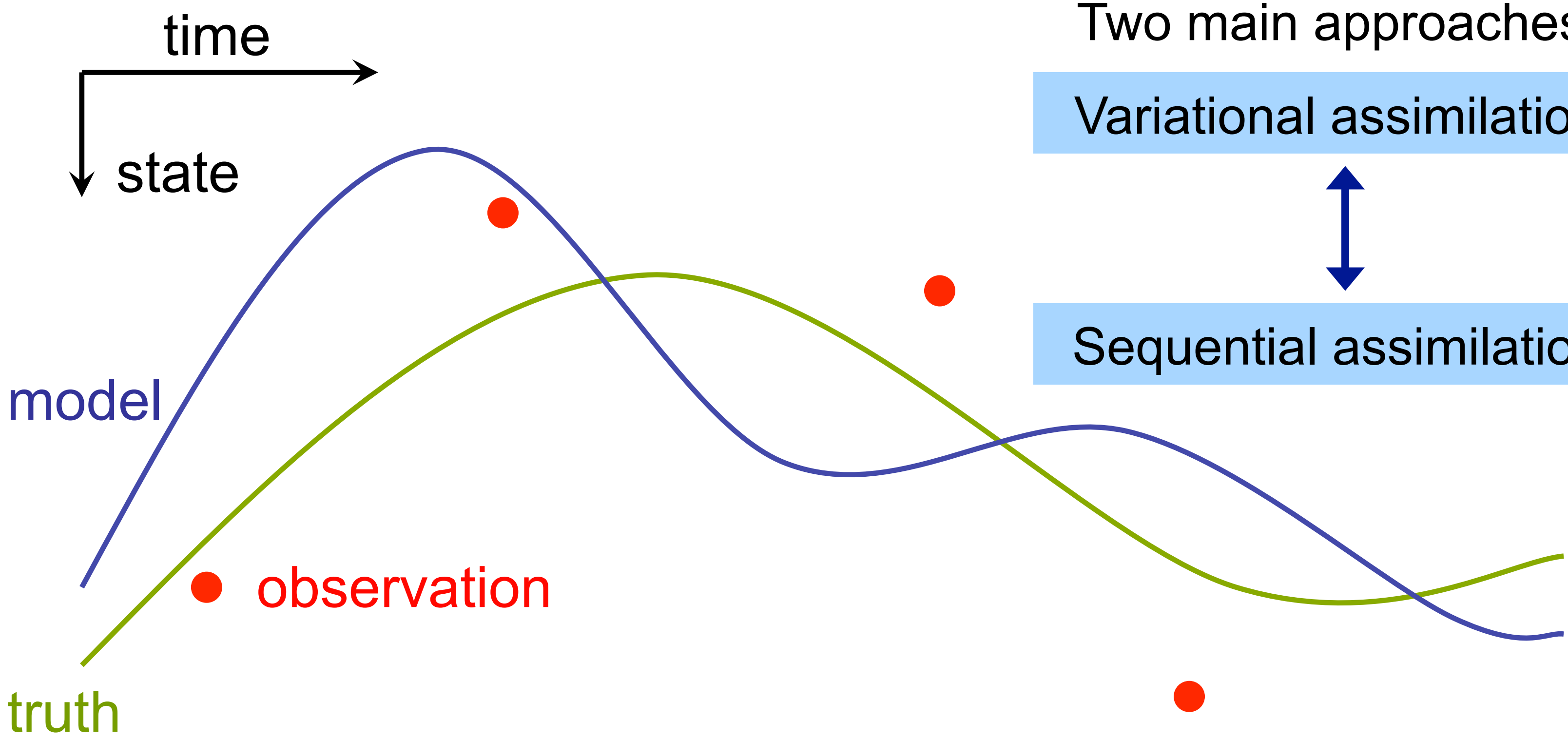
Losa, S.N. et al. *J. Marine Syst.* 105 (2012) 152-162

Data Assimilation

- Combine model with real data
- 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, …)
- Also: Improvement of model formulation
  - parameterizations (biogeochemistry, sea-ice, …)
- Characteristics of system:
  - high-dimensional numerical model – $\mathcal{O}(10^6\text{-}10^9)$
  - 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

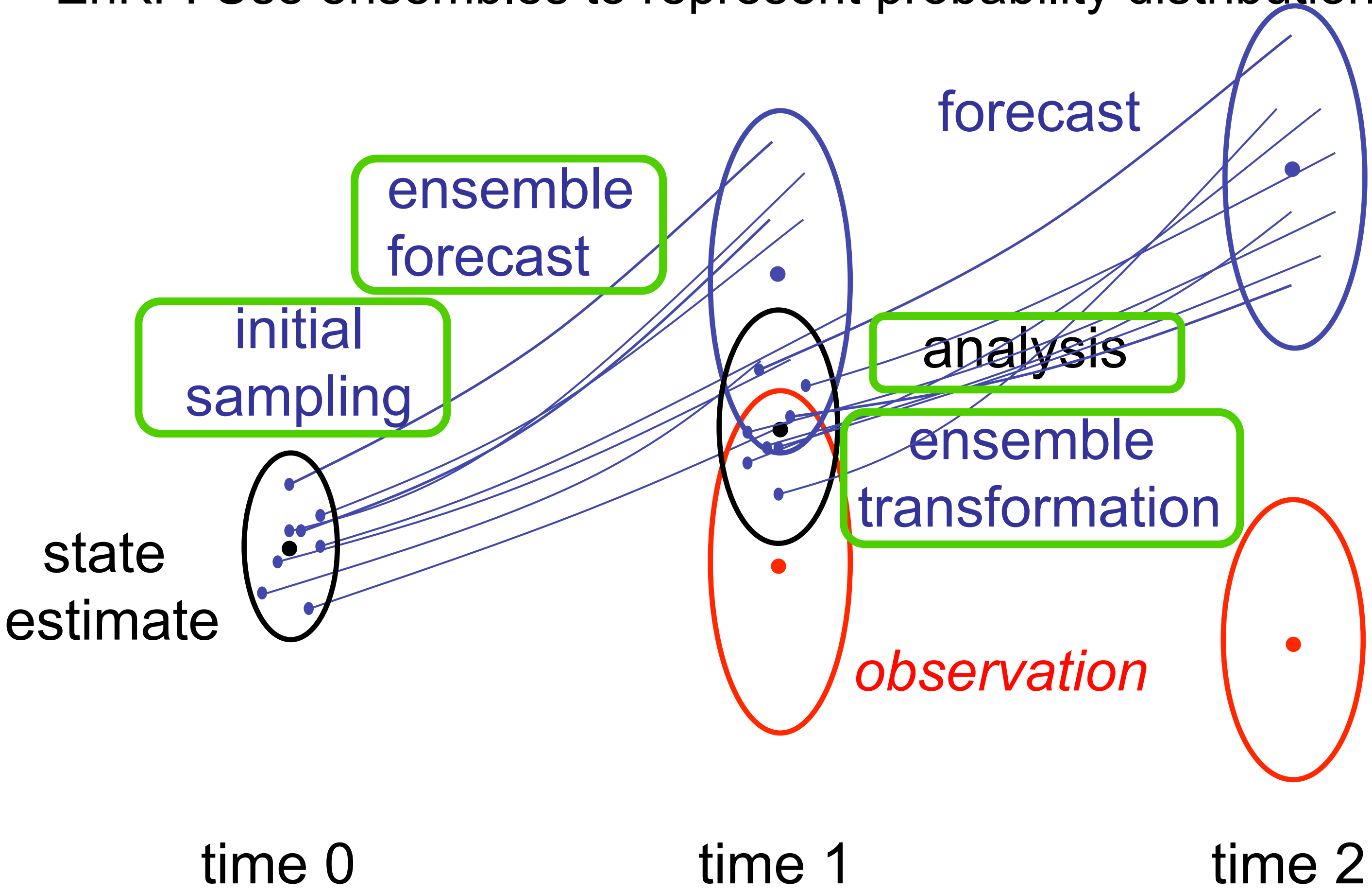
Ensemble-based Kalman Filters

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 simple!

BUT:

There are many possible choices!

What is optimal?

Ensemble-based/error-subspace Kalman filters

A little “zoo” (not complete):

Which filter should one use?

EnKF(94/98)

Studied in Nerger et al. (2005)

RRSQRT

ROEK

SEEK

SEIK

Studied in (Nerger et al. 2012)

EnKF(2003)

EnKF(2004)

EAKF

EnSRF

DEnKF

ETKF

ESTKF

MLEF

SPKF

ESSE

RHF

anamorphosis

New Study (Nerger 2015)

New filter formulation

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

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

L. Nerger, Monthly Weather Review 143 (2015) 1554-1567

Right sided ensemble transformation

$$\mathbf{X}'^{a} = \mathbf{X}'^{f}\mathbf{W}$$

With ensemble perturbation matrix $\mathbf{X}'$; ensemble size N

Very efficient: $\mathbf{W}$ is small ($N \times N$ or $(N-1) \times (N-1)$)

Used in:

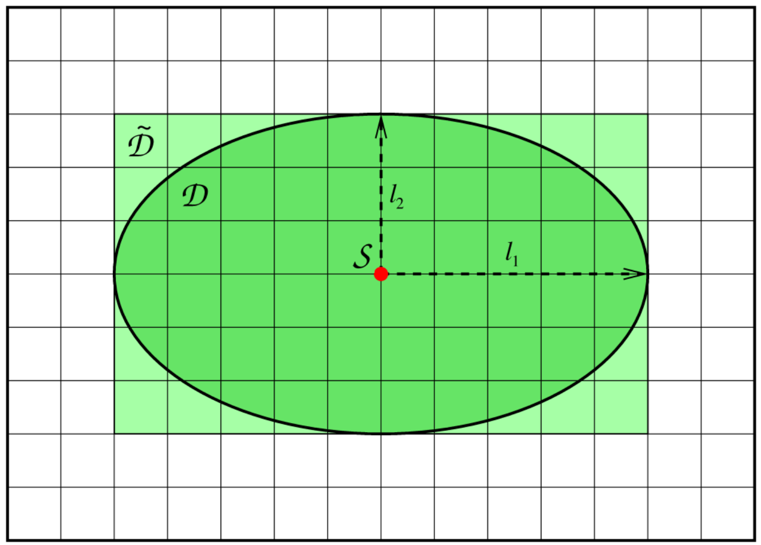
- **SEIK** (Singular Evolutive Interpolated KF, Pham et al. 1998)
- **ETKF** (Ensemble Transform KF, Bishop et al. 2001)
- **EnsRF** (Ensemble Square-root Filter, Whitaker/Hamill 2001)
- **ESTKF** (Error-Subspace Transform KF, Nerger et al. 2012)

Requirements for applying ensemble Kalman filters

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

- e.g. due to
  - small ensemble size
  - nonlinearity
  - bias in model or data

Localization



S: Analysis region
D: Corresponding data region

Improvements through

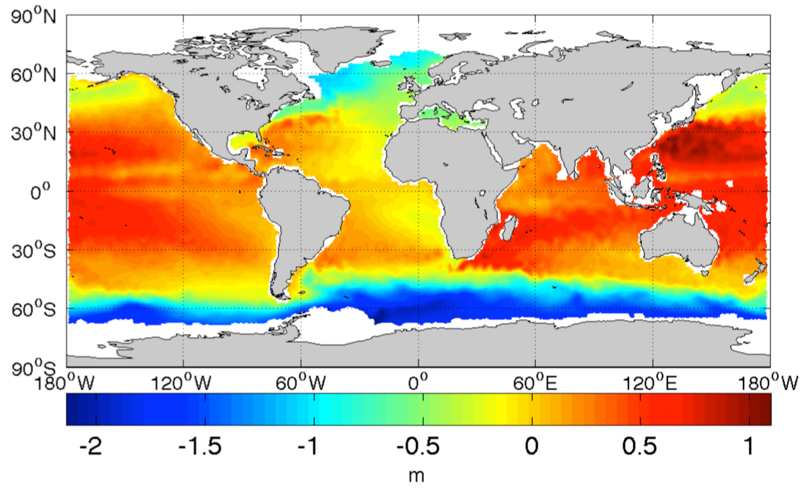
- Covariance inflation
- Localization
- Model error simulation

Implementation Aspects

Large scale data assimilation: Global ocean model

- Finite-element sea-ice ocean model (FESOM)
- Global configuration (~1.3 degree resolution with refinement at equator)
- State vector size: 10^7
- Scales well up to 256 processor cores

Sea surface elevation



- Ocean state estimation by assimilating satellite data („ocean topography“)
- Very costly due to large model size (Currently using up to 2048 processor cores)

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

➜ Abstraction of assimilation problem

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

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 for real applications
- easily useable with virtually any numerical model
- makes good use of supercomputers

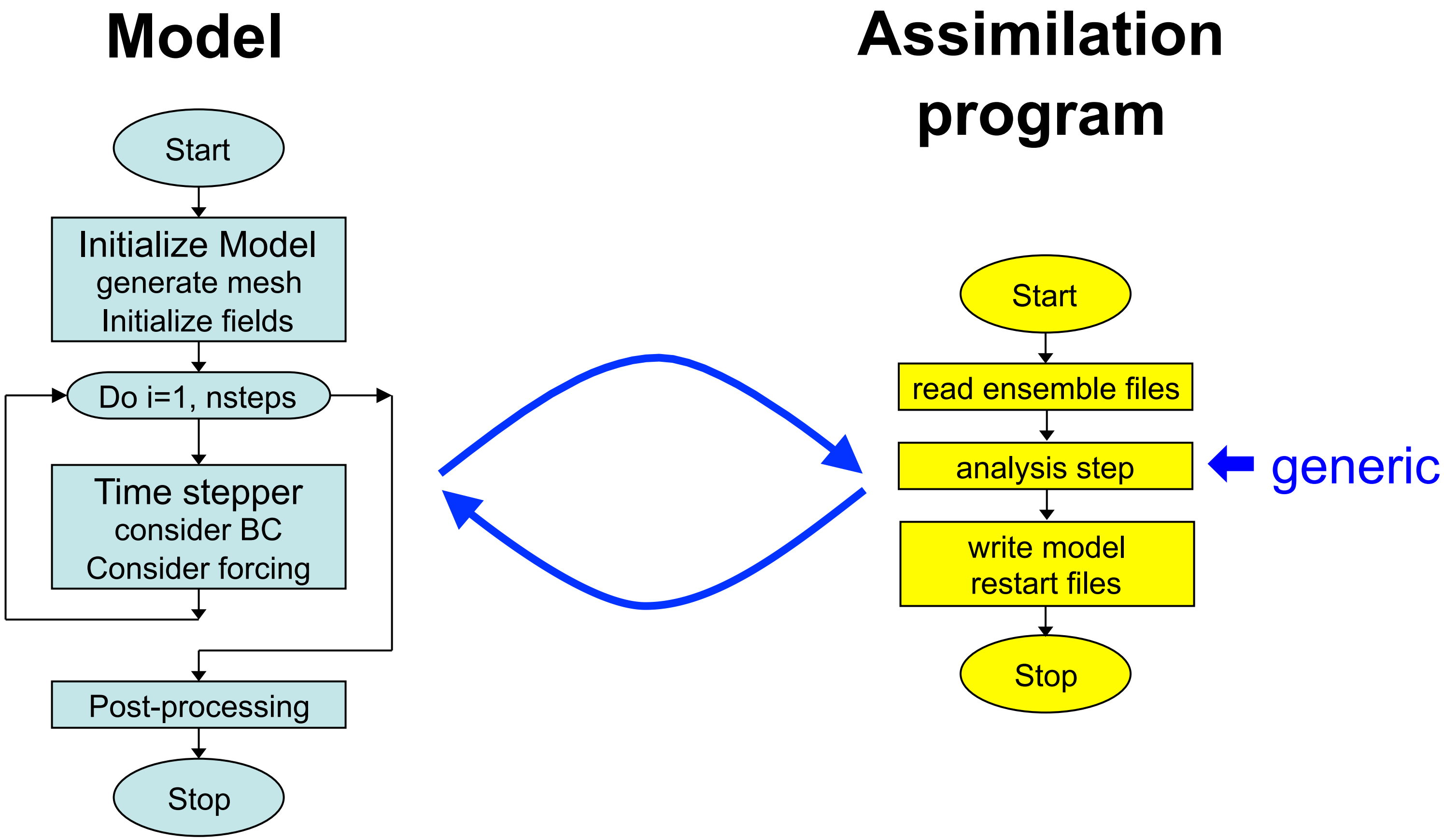
Open source:
Code and documentation available at

http://pdaf.awi.de

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

AWI

Offline mode – separate programs



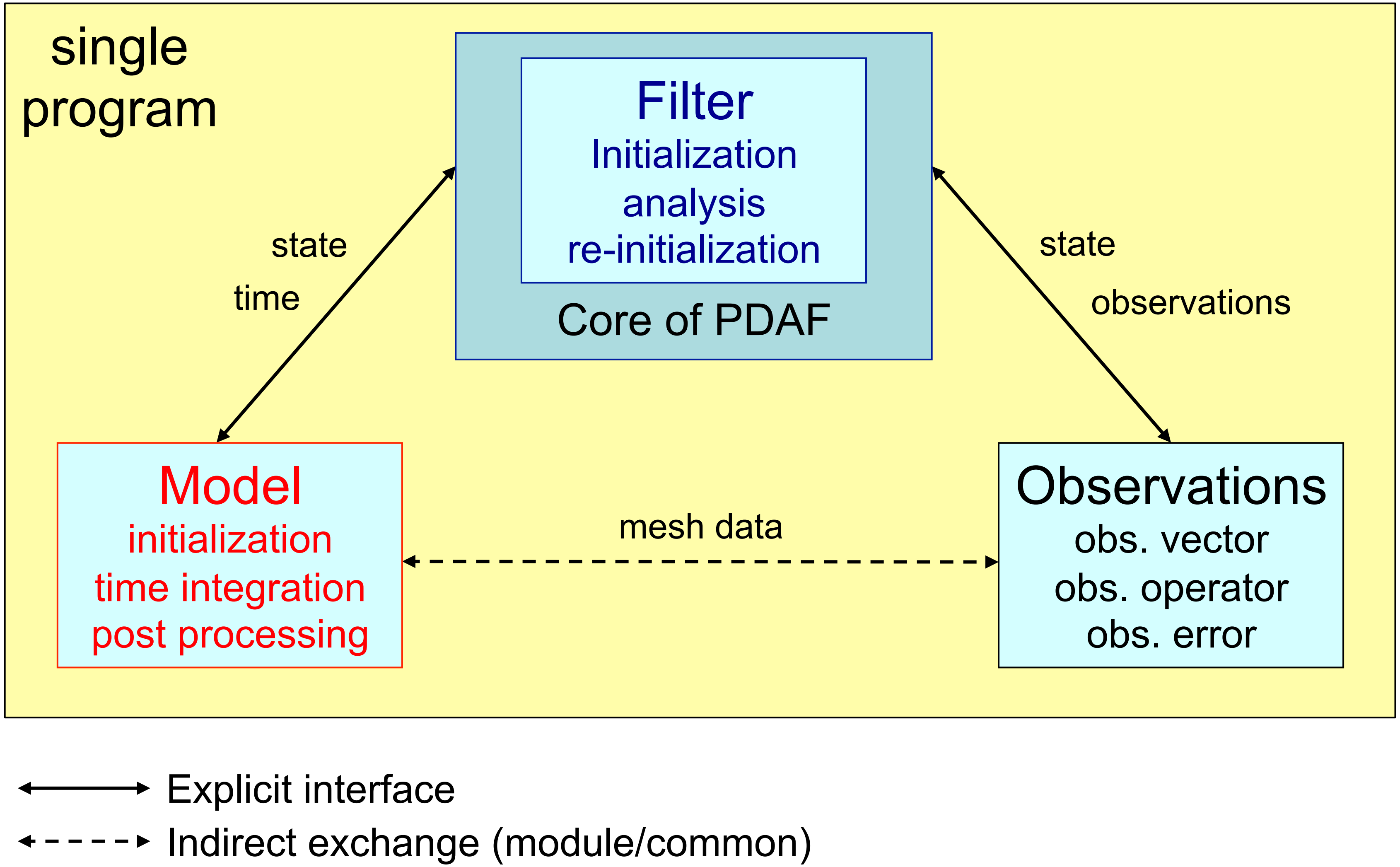
For each ensemble state

- Initialize from restart files
- Integrate
- Write restart files

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

Logical separation of assimilation system

PDAF Parallel Data Assimilation Framework

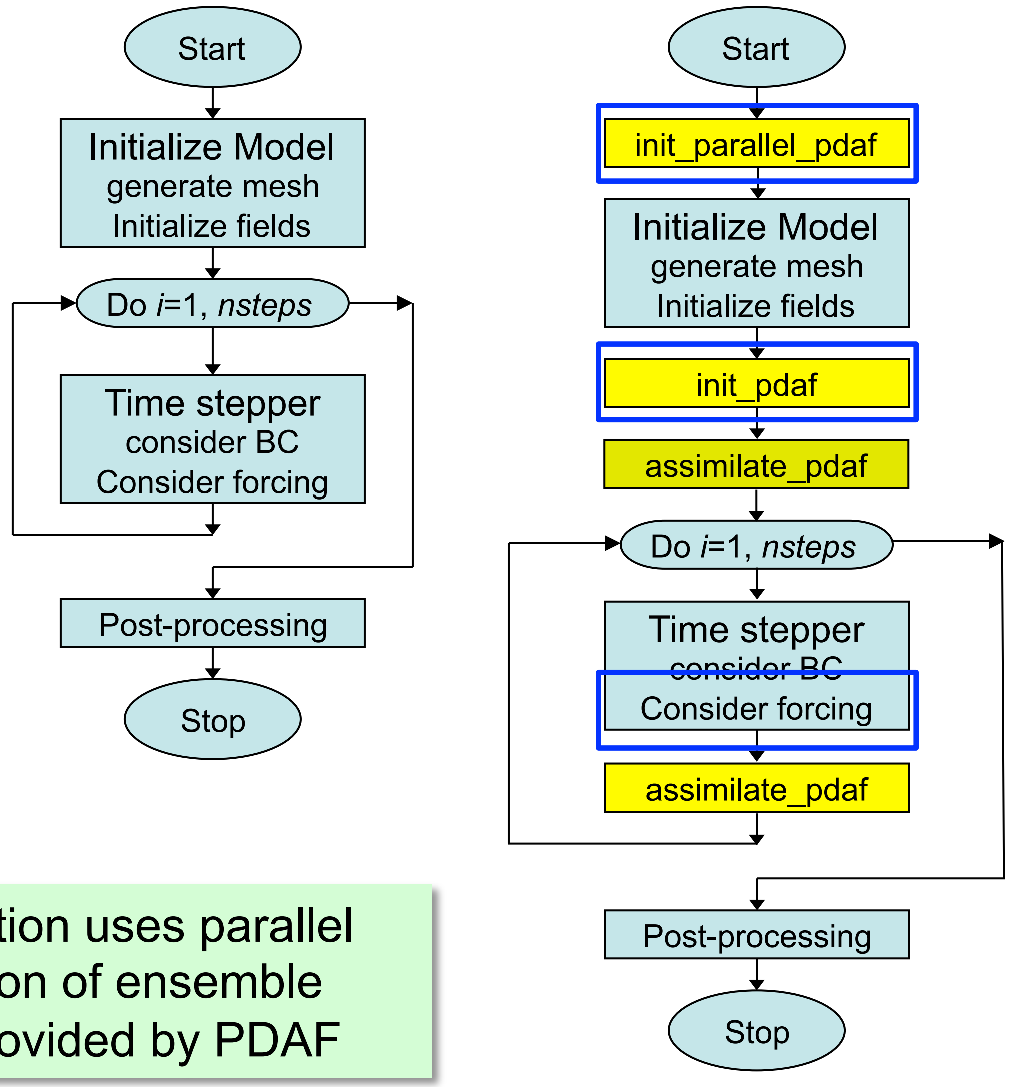


Nerger, L., Hiller, W. (2013). Software for Ensemble-based DA Systems – Implementation and Scalability. Computers and Geosciences. 55: 110-118

Extending a Model for Data Assimilation

PDAF Parallel Data Assimilation Framework

Model

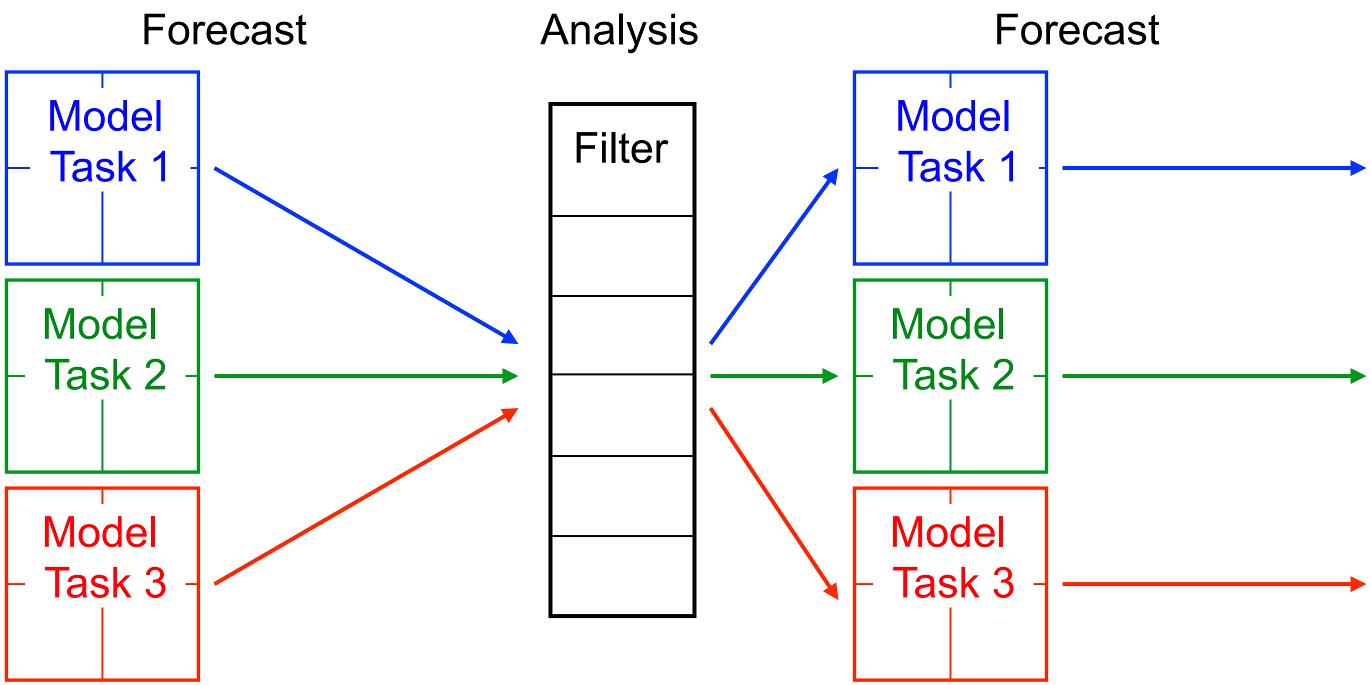


Extension for data assimilation

For operational forecasting use

Implementation uses parallel configuration of ensemble forecast provided by PDAF

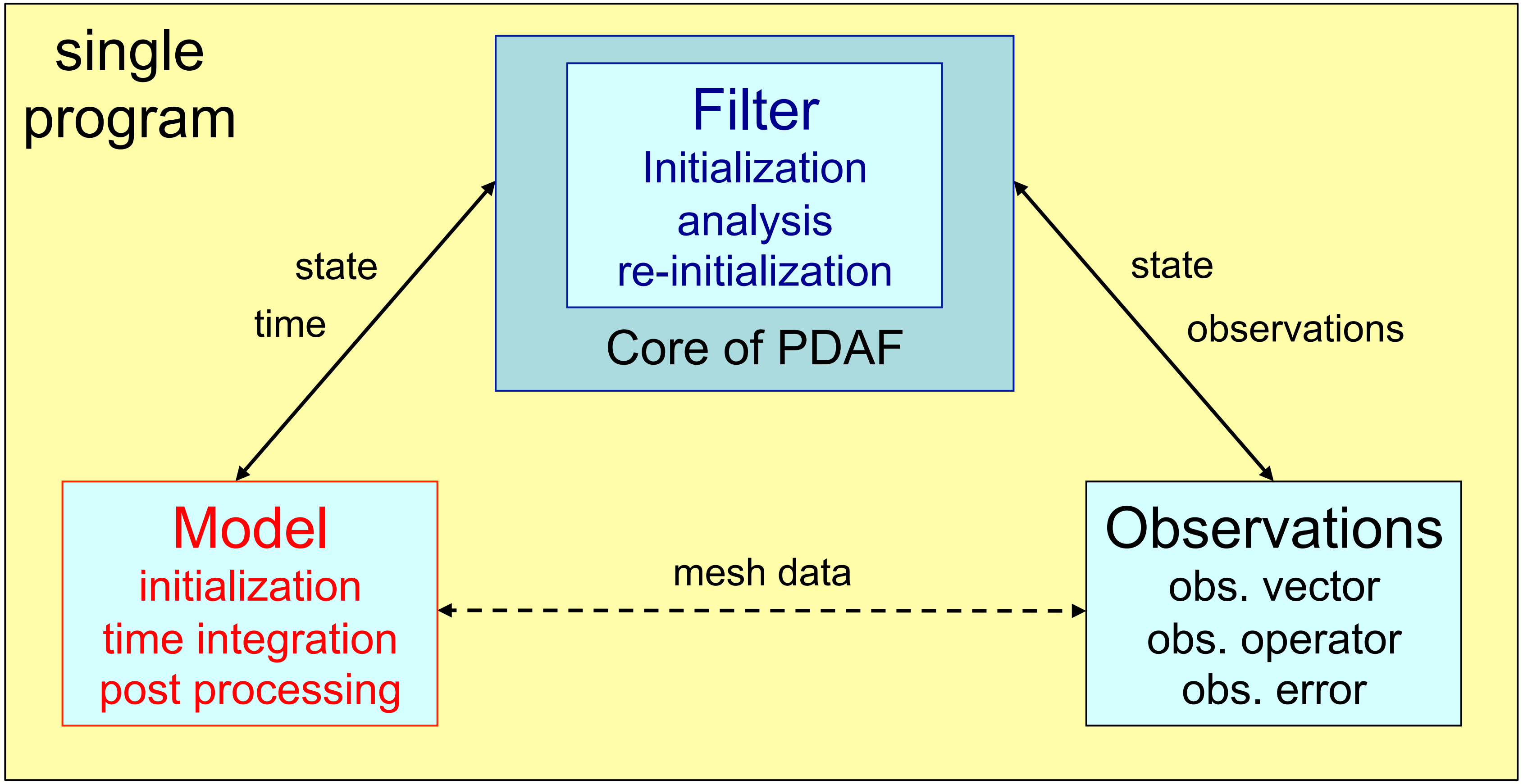
2-level Parallelism

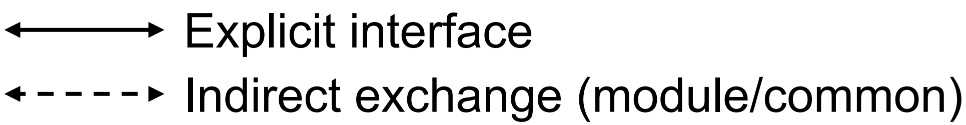


1. Multiple concurrent model tasks
2. Each model task can be parallelized

➢ Analysis step is also parallelized

User-supplied routines (call-back)

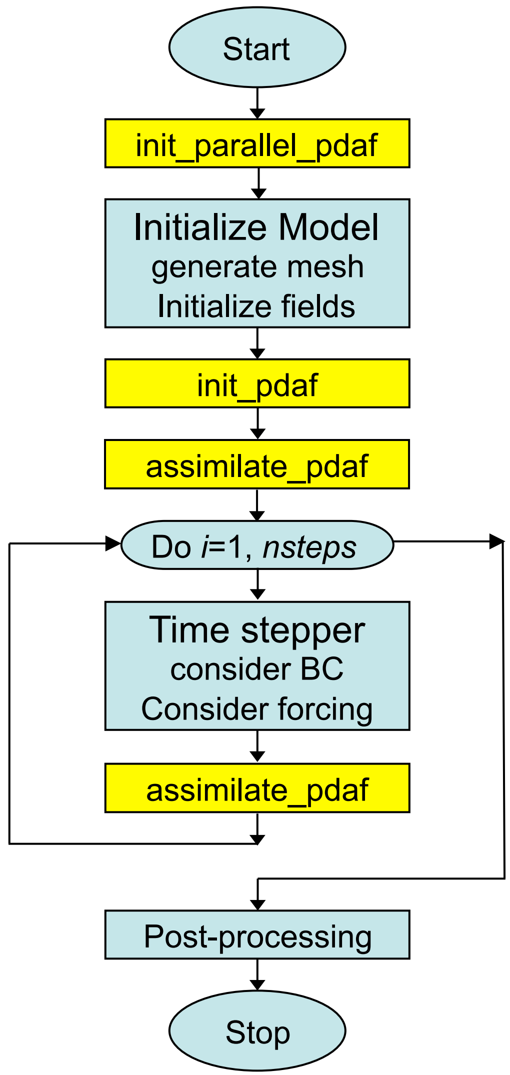




Features of online program

PDAF Parallel Data Assimilation Framework

- minimal changes to model code when combining model with filter algorithm
- model not required to be a subroutine
- no change to model numerics!
- model-sided control of assimilation program (user-supplied routines in model context)
- observation handling in model-context
- filter method encapsulated in subroutine
- complete parallelism in model, filter, and ensemble integrations



More Assimilation tools

- SANGOMA: Stochastic Assimilation for Next Generation Ocean Model Applications
- Project funded by European Union 2011-2015
- Different benchmark setups for data assimilation
- Development of set of data assimilation tools
  - Large set of different diagnostics (beyond RMS errors)
  - Tools for ensemble generation
  - Simplified filter analysis steps

www.data-assimilation.net

Parallel Performance of PDAF

Parallel performance of PDAF

- Performance tests on

  SGI Altix ICE at HRLN (German “High performance computer north”)

  nodes: 2 quad-core Intel Xeon Gainestown at 2.93GHz
  network: 4x DDR Infiniband
  compiler: Intel 10.1, MPI: MVAPICH2

- Ensemble forecasts

  - are naturally parallel

  - dominate computing time
    Example: parallel forecast over 10 days: 45s
    SEIK with 16 ensemble members: 0.1s
    LSEIK with 16 ensemble members: 0.7s

Parallel Performance

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

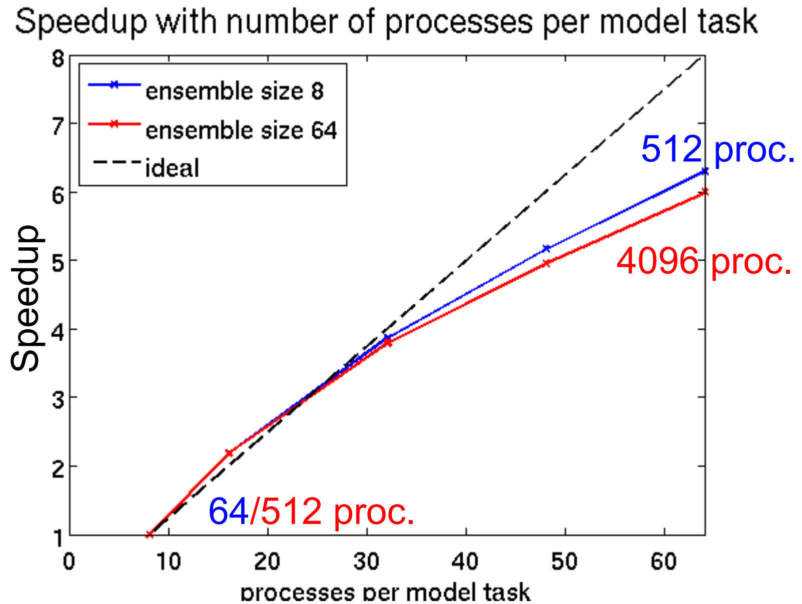
94-99% of computing time in model integrations

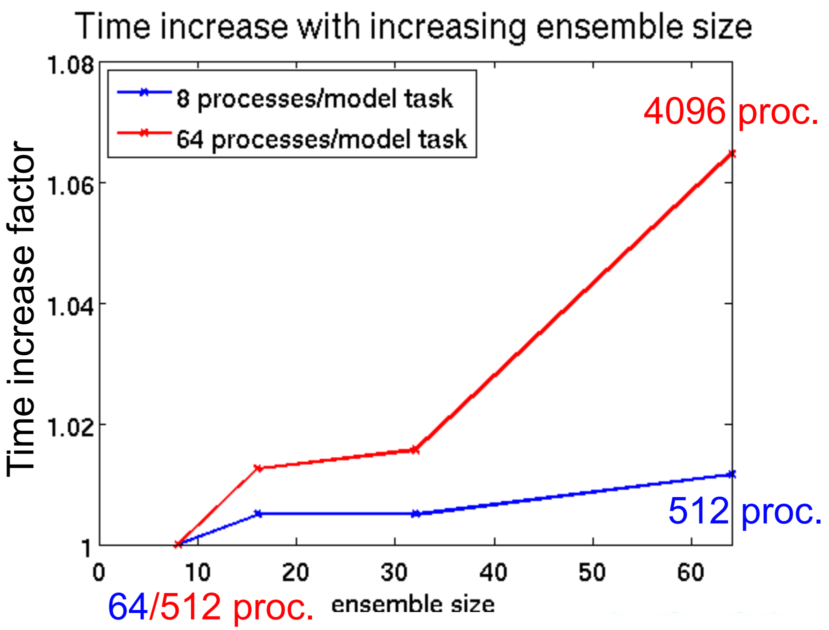
Speedup: Increase number of processes for each model task, fixed ensemble size

- 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

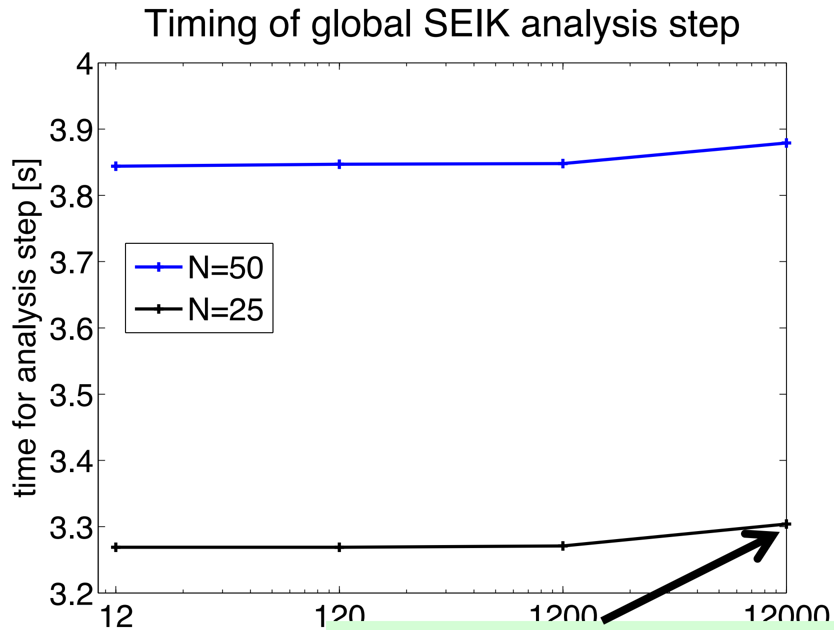




Very big test case

PDAF Parallel Data Assimilation Framework

- Simulate a “model”
- Choose an ensemble
  - state vector per processor: 10^7
  - observations per processor: $2 \cdot 10^5$
  - Ensemble size: 25
  - 2GB memory per processor
- Apply analysis step for different processor numbers
  - 12 – 120 – 1200 – 12000



State dimension: 1.2e11

Observation dimension: 2.4e9

- Close to ideal: Very small increase in analysis time (~1%)
- Didn’t try to run a real ensemble of largest state size (no model yet)

Application Example

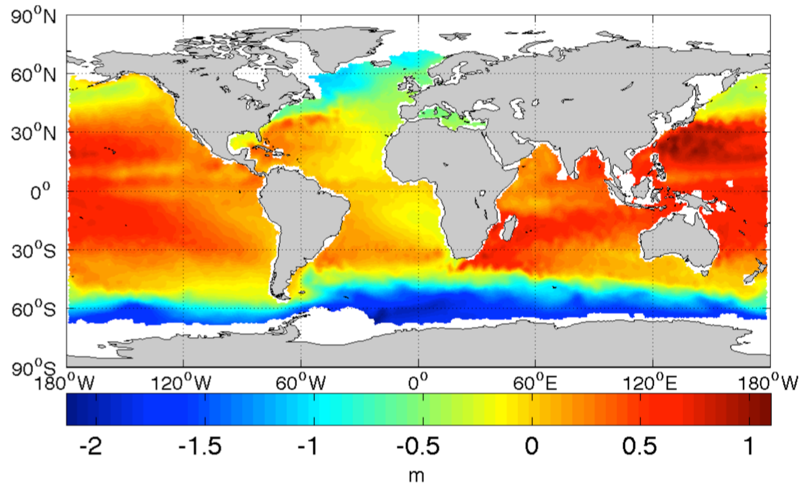
Ocean Topography Assimilation

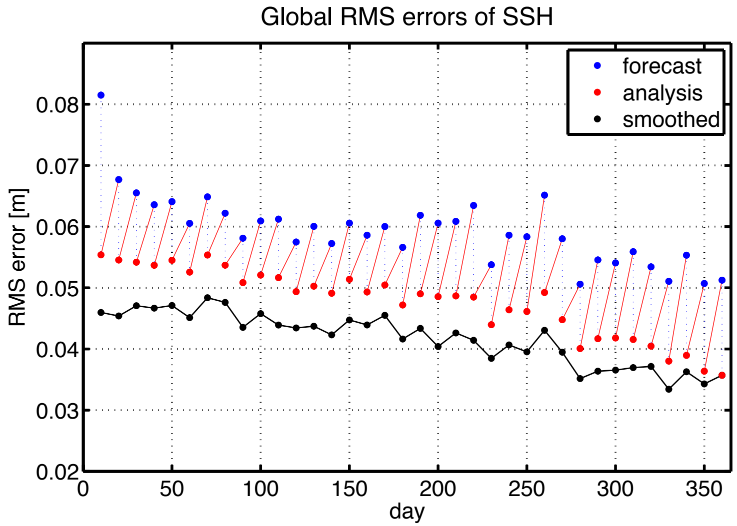
(Run by A. Androsov, R. Schnur)

- Assimilation of sea surface height data („ocean topography“)
- Full height generated from satellite altimetry and geoid data
- Apply ensemble-based filter and smoother methods

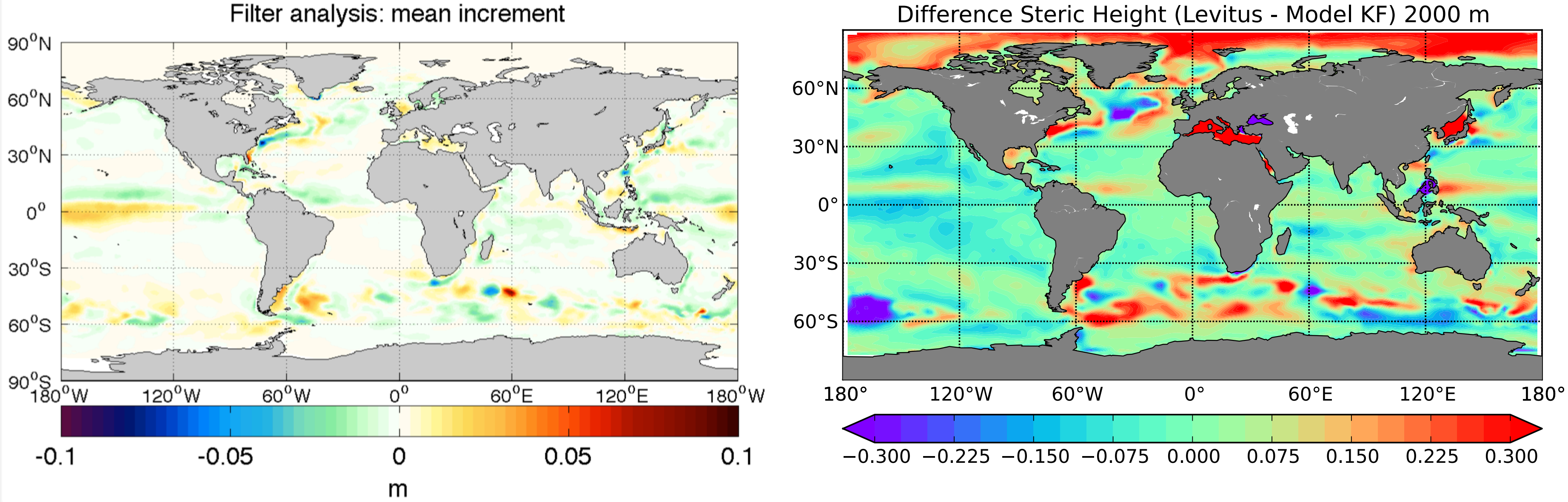
- Root-mean square errors significantly reduced
- Smoother results in smaller errors and smoother curve

Sea surface elevation



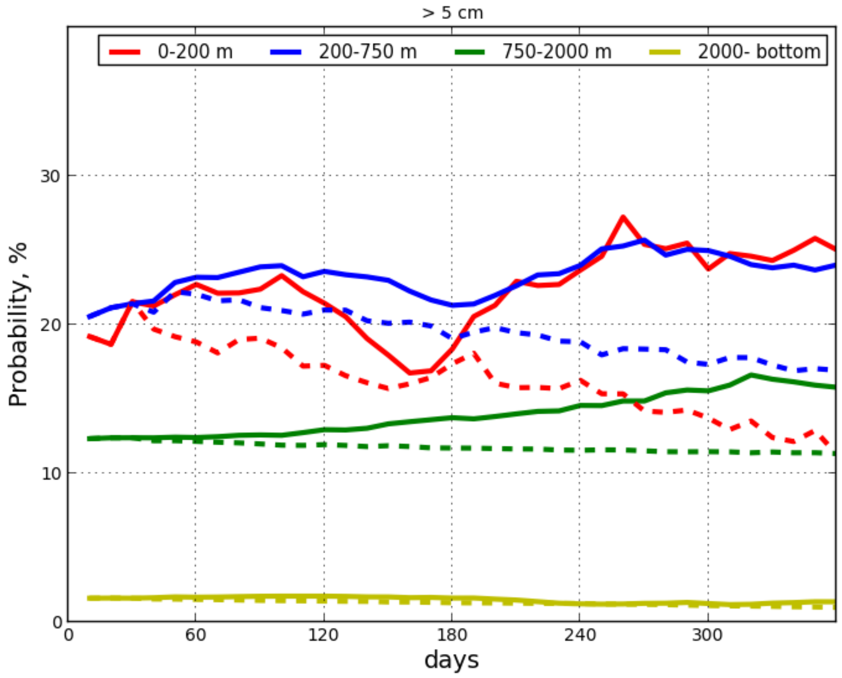


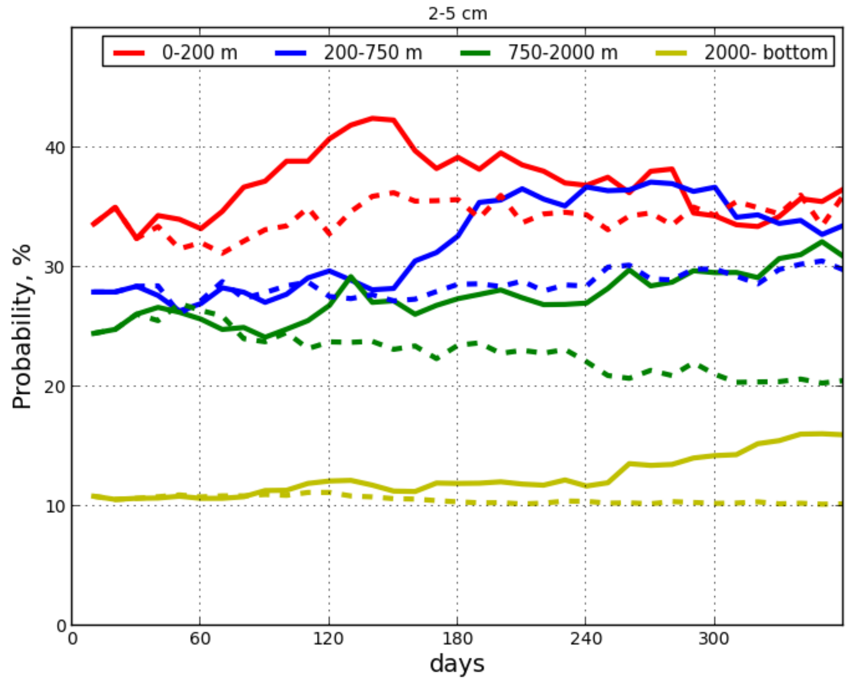
Correcting model biases



- Mean assimilation increments show that biases are corrected
- Consistently visible in steric height

Depth-dependent changes to steric height





- Significant influence of assimilation (>5cm) down to 2000m
- Influence of assimilation also below 2000m depth
- State changes quite stable if model is run freely (dashed lines)

Summary

- Ensemble-based Kalman filters:
  - Current efficient methods
    suited for large-scale problems
  - Tuning of filters required
- Simplification of technical implementation using PDAF
- Assimilation with high-dimensional global ocean model
  - Assimilating surface data improves mean ocean state
  - Significant influence on steric height down to 2000m

Thank you !