Seminar at NMEFC, Beijing, China, October 10, 2014

Ensemble Data Assimilation:

Algorithms and Software

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

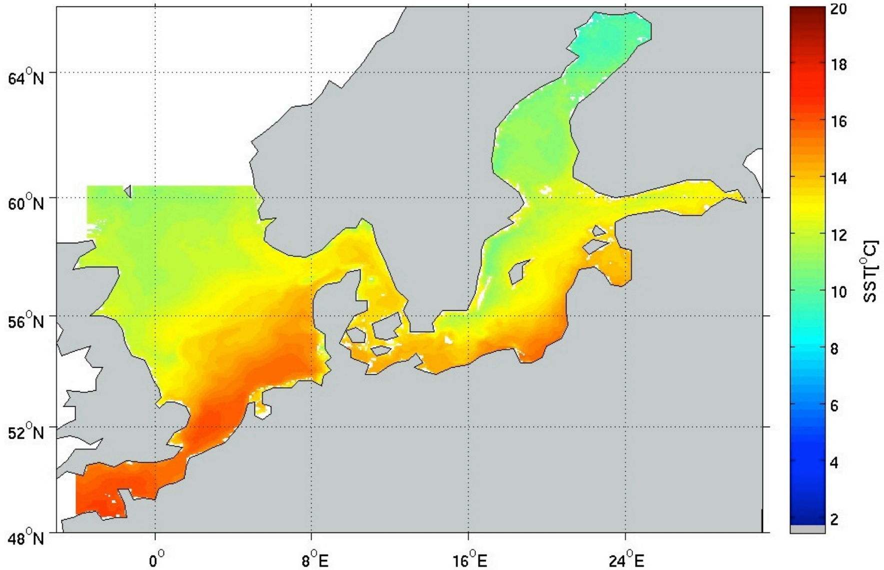
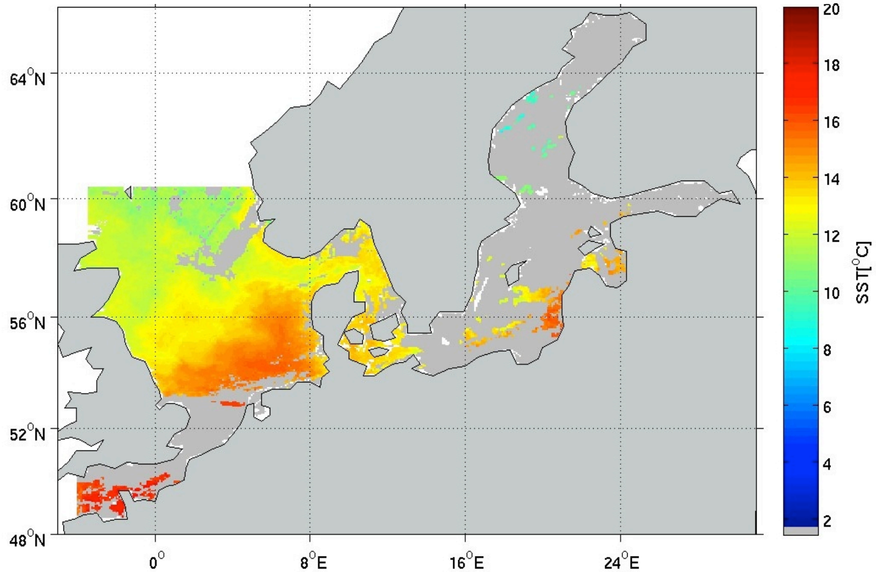
BremHLR
Kompetenzzentrum für Höchstleistungsrechnen Bremen

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

Outline

- Ensemble-based Kalman filters
- Implementation aspects
- Assimilation software PDAF

Motivation

Model surface temperature	*Satellite* surface temperature
	
Information: Model	Information: Observations

Combine both sources of information

quantitatively by computer algorithm

➜ data assimilation

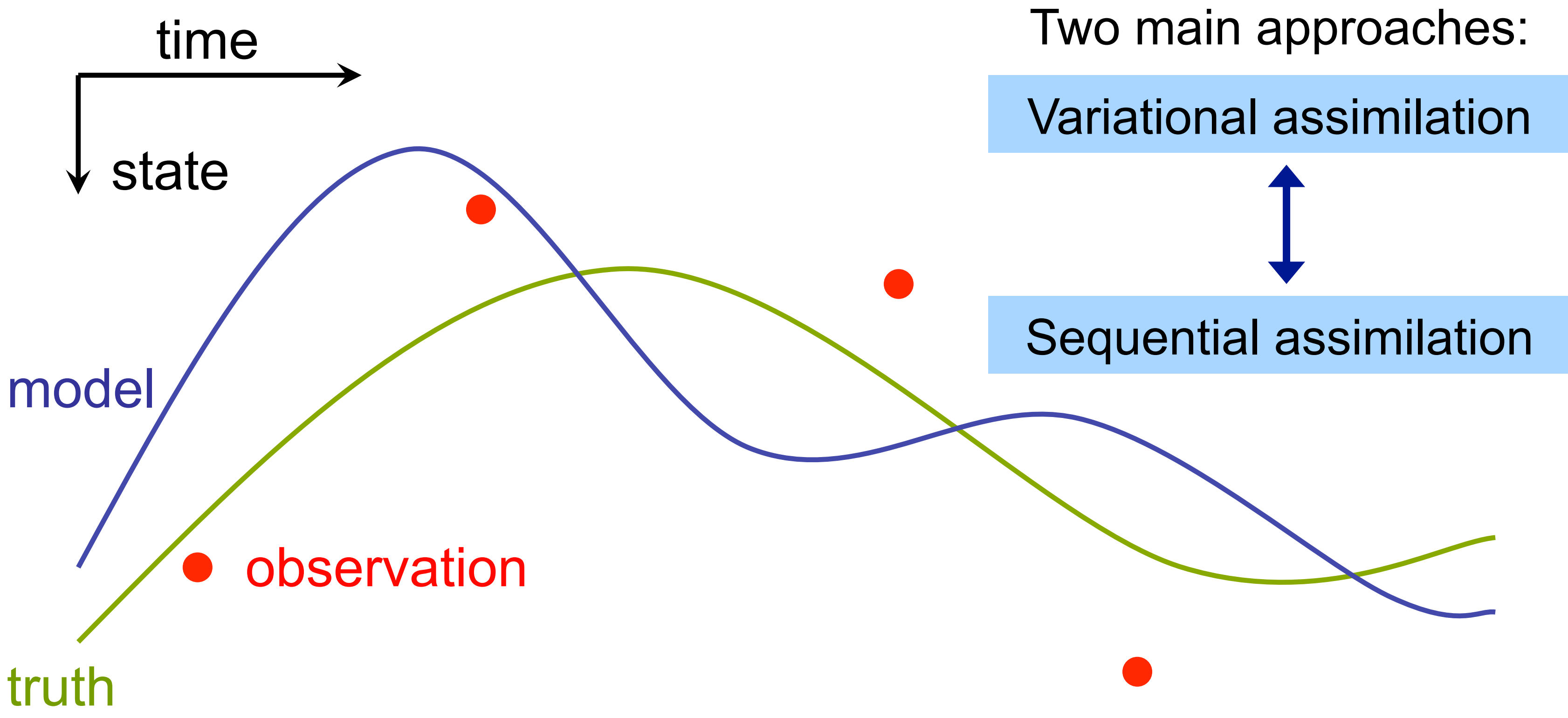
Losa, S.N. et al. *J. Marine Syst.* 105 (2012) 152-162

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,…)



Optimal estimate basically by least-squares fitting

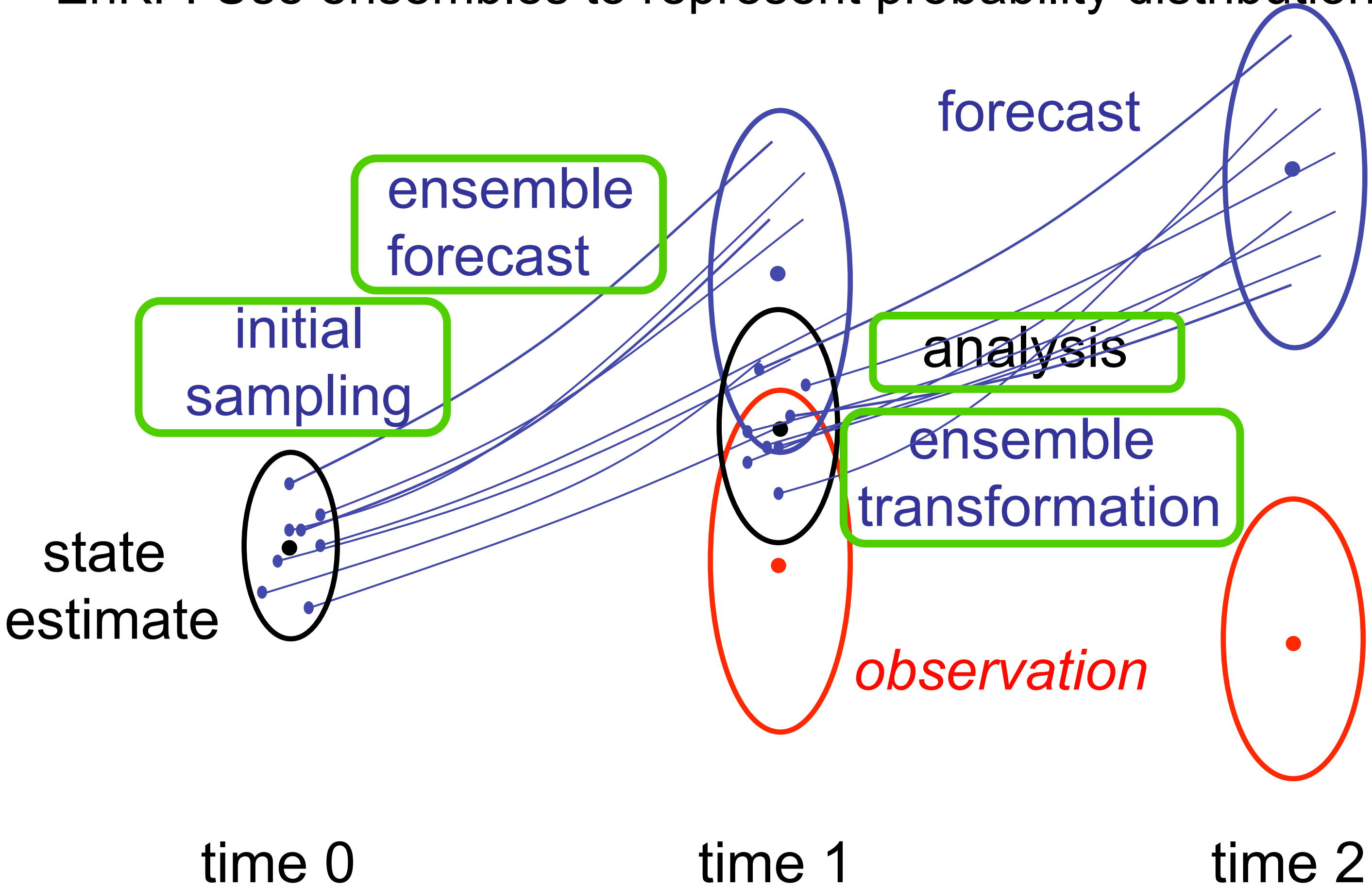
Ensemble-based Kalman Filters

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?

Data Assimilation – Model and Observations

Two components:

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

   Dynamical model

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

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

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

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

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

The Ensemble Kalman Filter (EnKF, Evensen 94)

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

Analysis step:

Update each ensemble member

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

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

Kalman filter

Ensemble covariance matrix

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

Ensemble mean (state estimate)

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

Efficient use of ensembles

Kalman gain

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

Alternative form (Sherman-Morrison-Woodbury matrix identity)

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

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

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

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

Inversion of $N \times N$ matrix

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

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

New study (Nerger et al. 2012)

EnKF(2003)

EnKF(2004)

EAKF

EnSRF

DEnKF

ETKF

ESTKF

MLEF

SPKF

ESSE

RHF

anamorphosis

New filter formulation

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

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

Right sided ensemble transformation

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

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)

ESTKF (Error-Subspace Transform KF)

Error-subspace basis matrix — size (n x N-1)

$$\mathbf{L} := \mathbf{X}^f \mathbf{T}$$

(T projects onto error space spanned by ensemble)

Analysis covariance matrix — (n x n)

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

"Transform matrix" in error subspace — (N-1 x N-1)

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

Transformation of ensemble perturbations — (n x N)

$$\mathbf{X}'^a = \mathbf{L}\mathbf{W}^{ESTKF}$$

Ensemble weight matrix — (N-1 x N)

$$\mathbf{W}^{ESTKF} = \sqrt{N-1}\,\mathbf{C}\mathbf{T}^T$$

- $\mathbf{C}$ is symmetric square root of $\mathbf{A}$

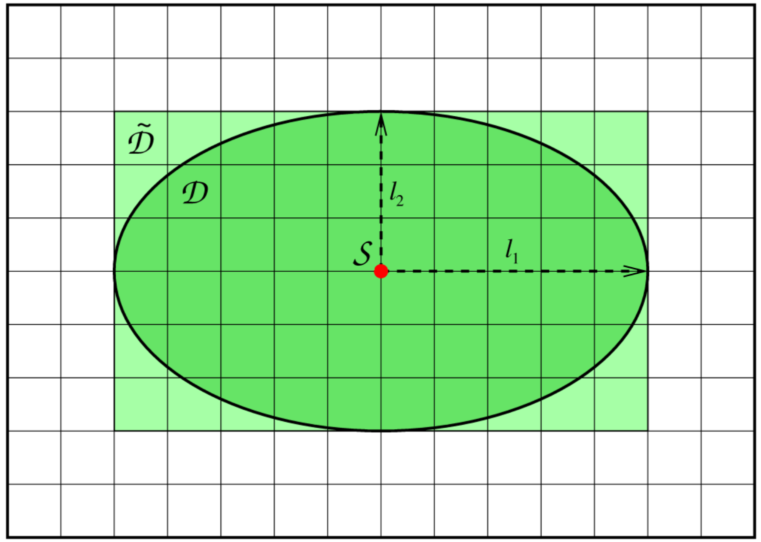
L. Nerger et al., Monthly Weather Review 140 (2012) 2335-2345

Requirements for applying ensemble Kalman filters

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

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

Localization



S: Analysis region
D: Corresponding data region

Improvements through

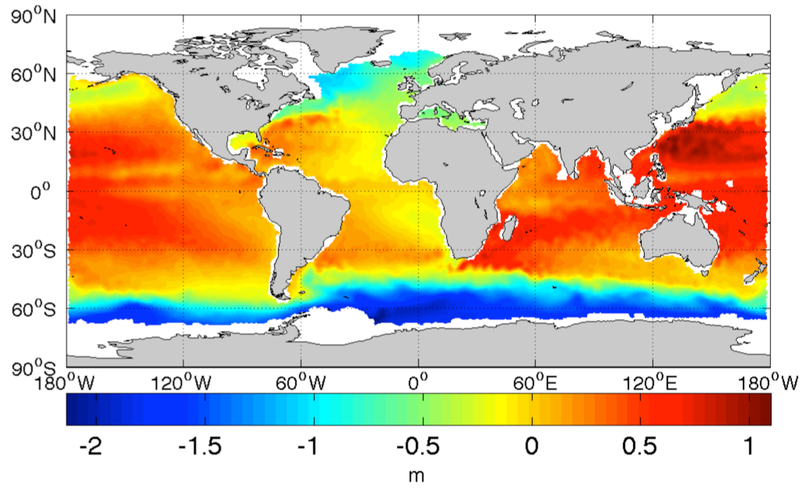
- Covariance inflation
- Localization
- Model error simulation

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

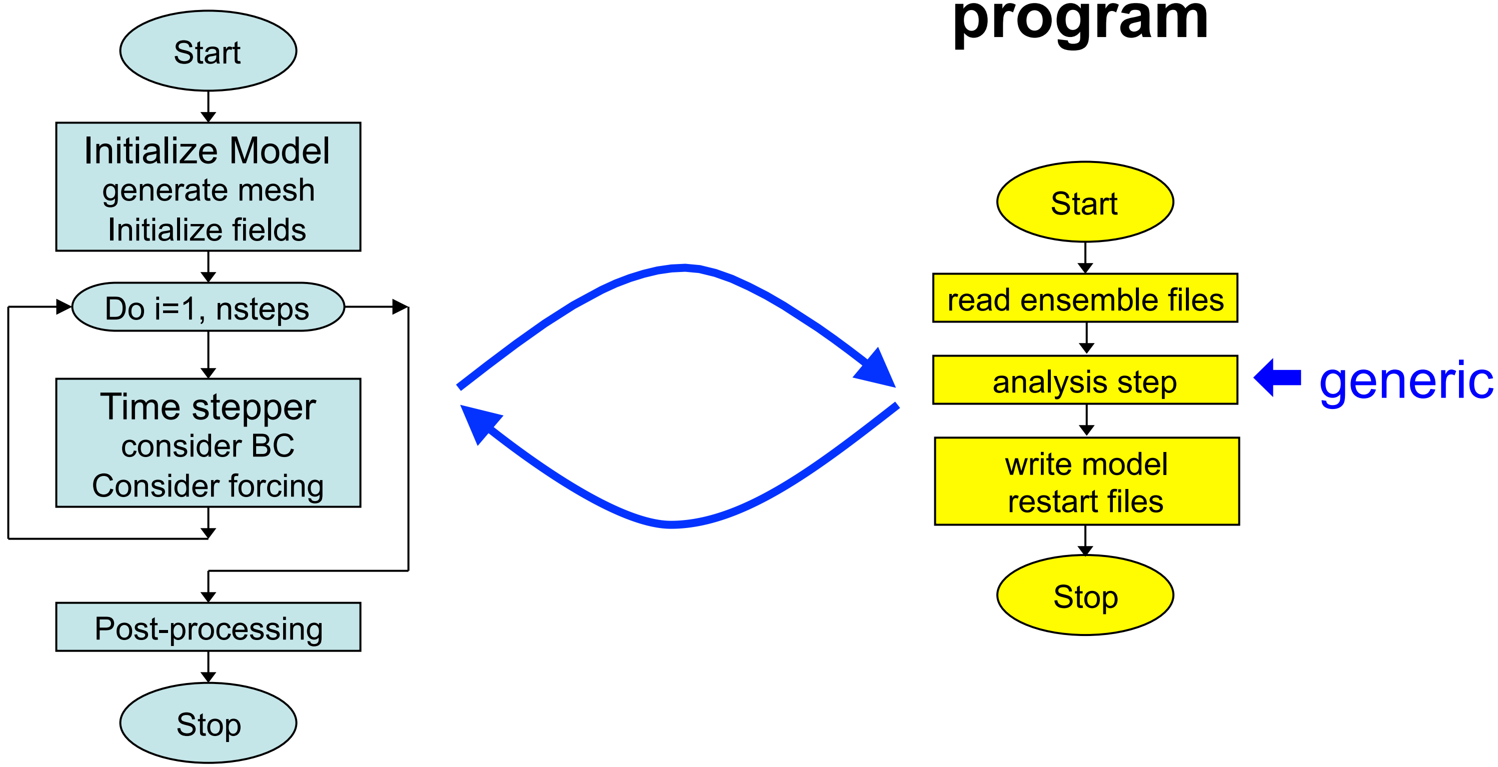
http://pdaf.awi.de

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

AWI

Offline mode – separate programs

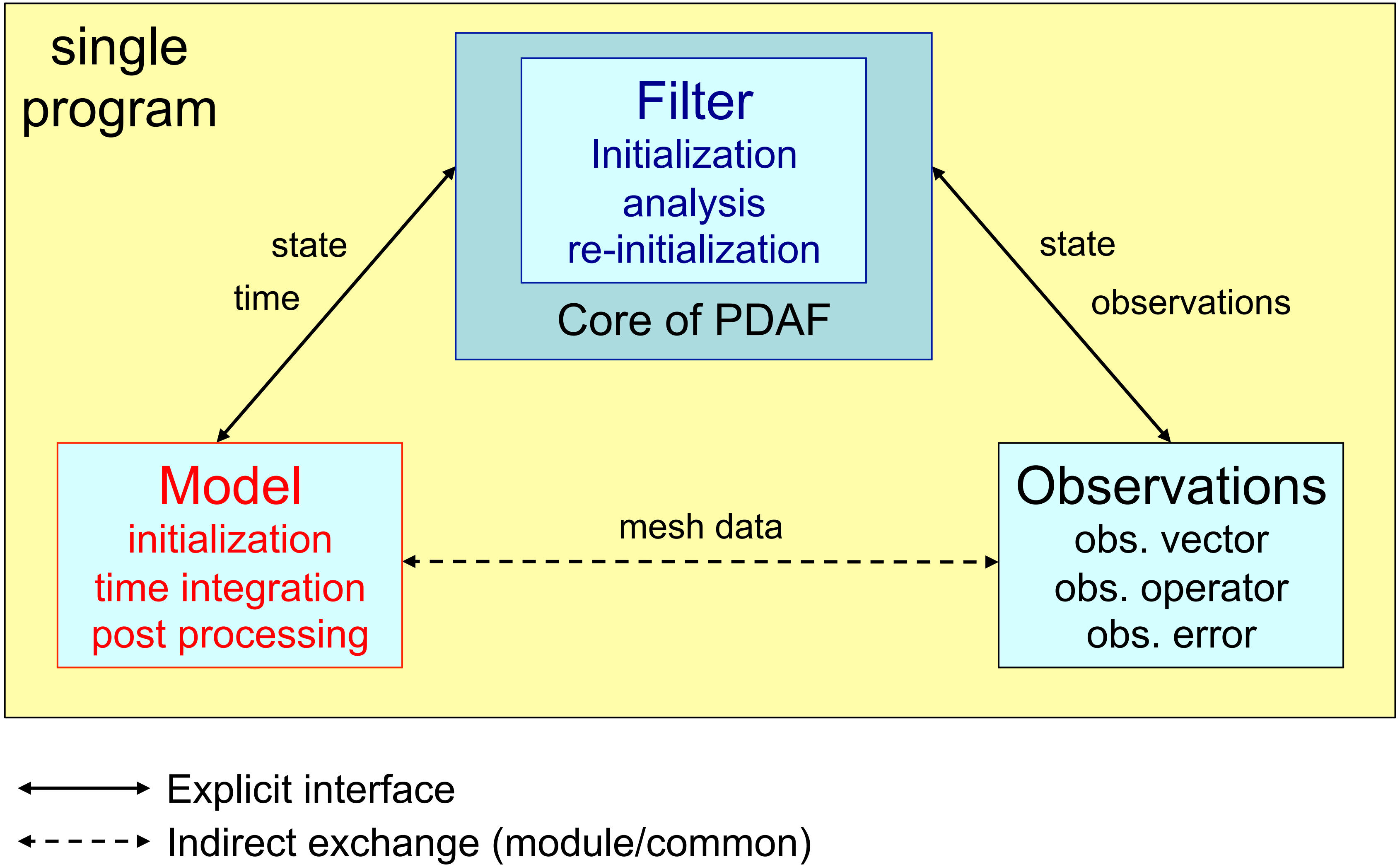
Model | **Assimilation program**



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

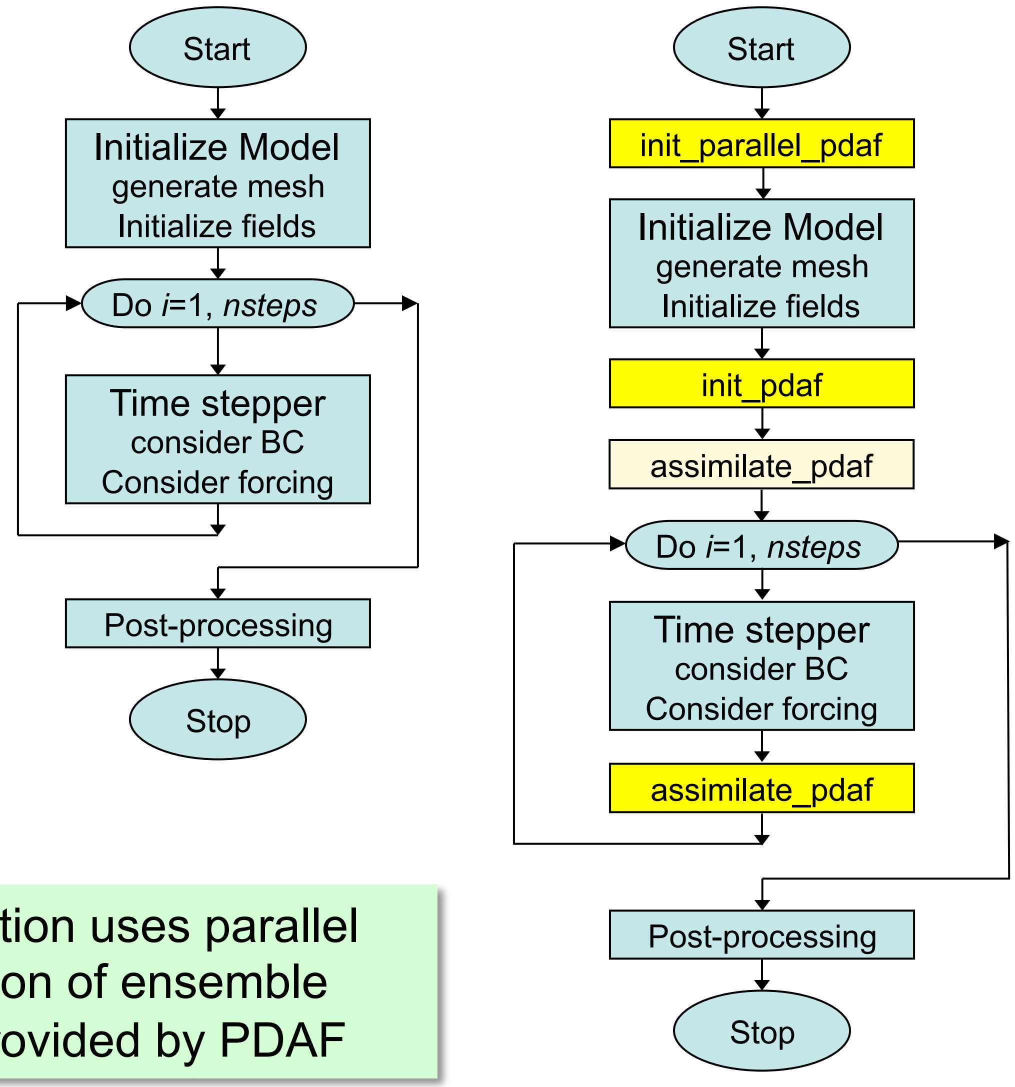


Nerger, L., Hiller, W. (2013). Software for Ensemble-based DA Systems – Implementation and Scalability. Computers and Geosciences. 55: 110-118

Extending a Model for Data Assimilation

PDAF Parallel Data Assimilation Framework

Model



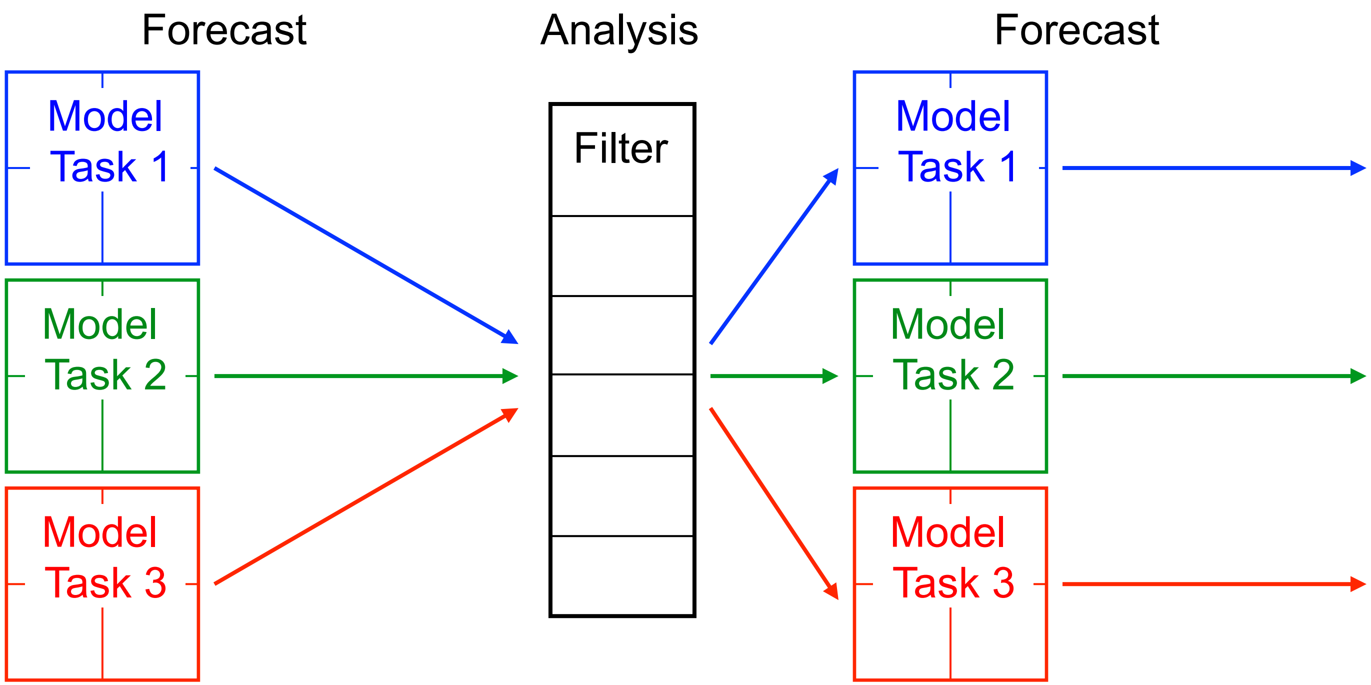
Extension for data assimilation

For operational forecasting use

Implementation uses parallel configuration of ensemble forecast provided by PDAF

2-level Parallelism

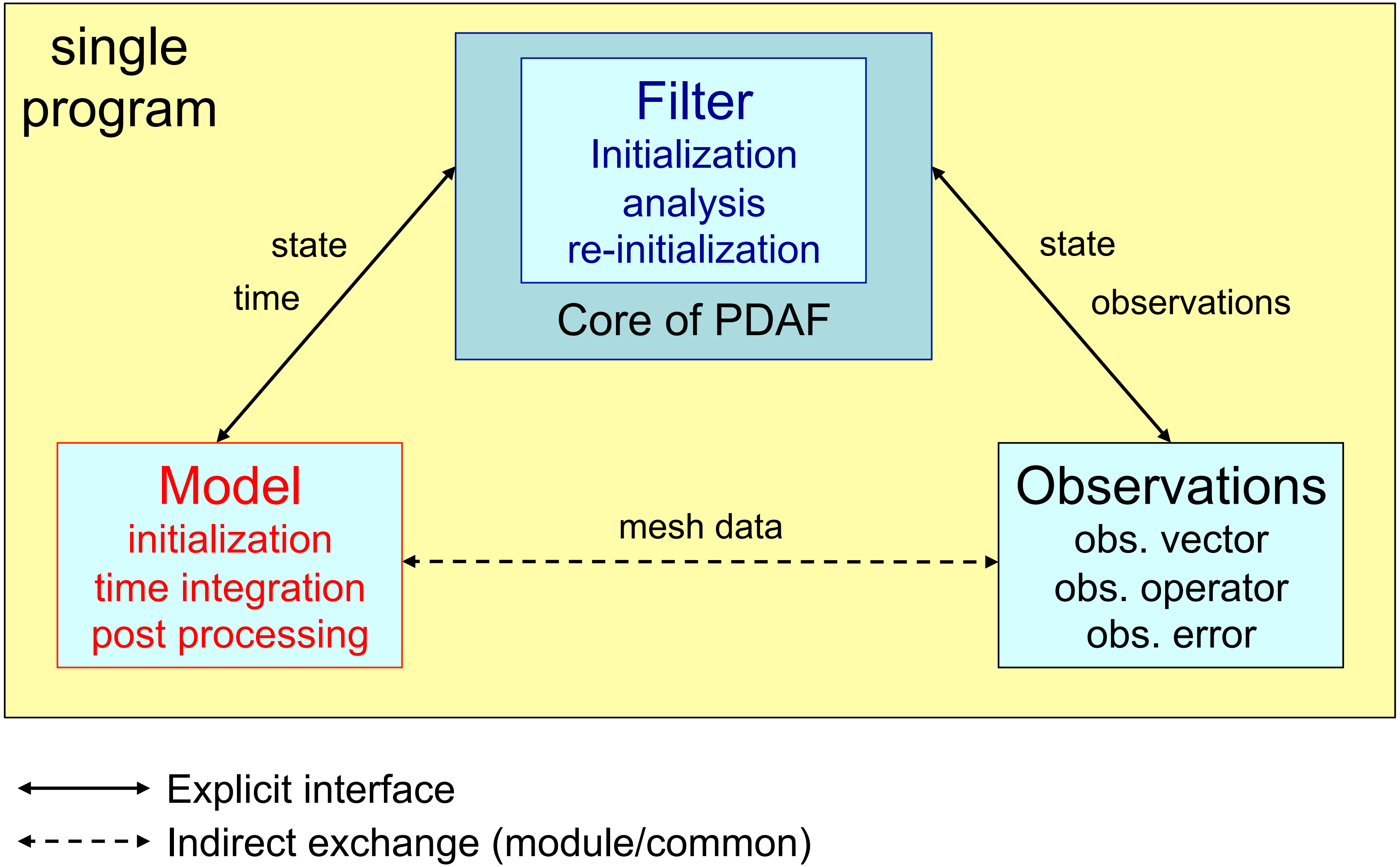
PDAF Parallel Data Assimilation Framework



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

➢ Analysis step is also parallelized

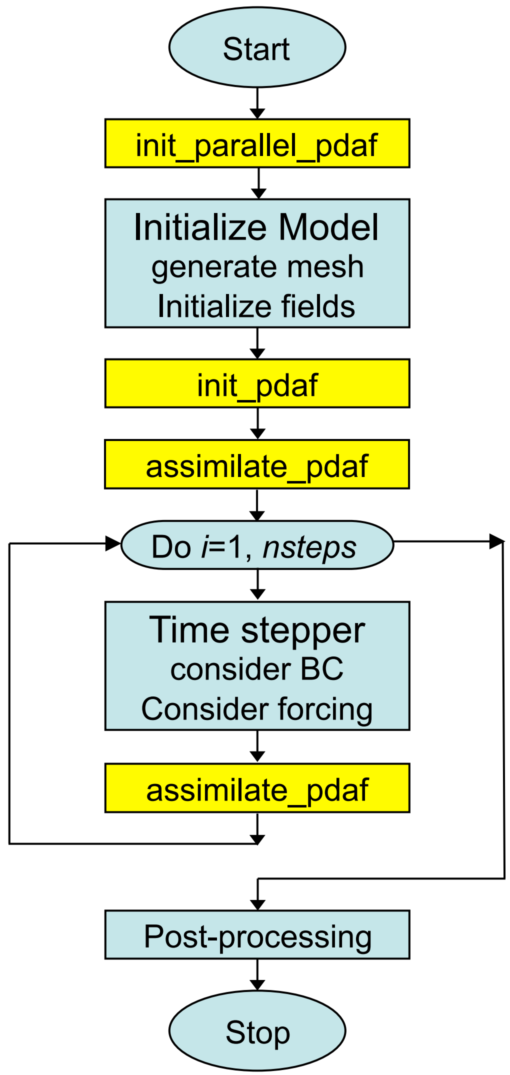
User-supplied routines (call-back)



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



Current algorithms in PDAF

PDAF originated from comparison studies of different filters

Filters

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

} Global filters

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

} Localized filters

Smoothers for

- ETKF/LETKF
- ESTKF/LESTKF
- EnKF

} Global and local smoothers

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)

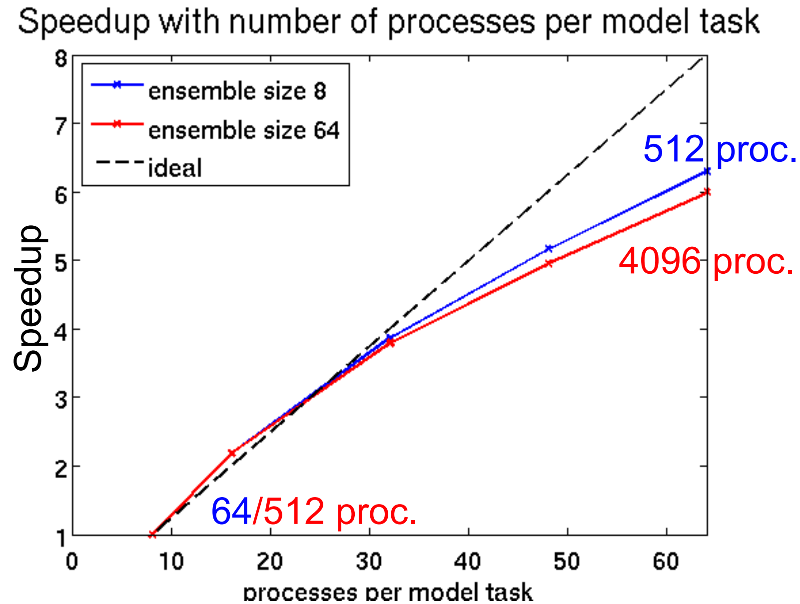
94-99% of computing time in model integrations

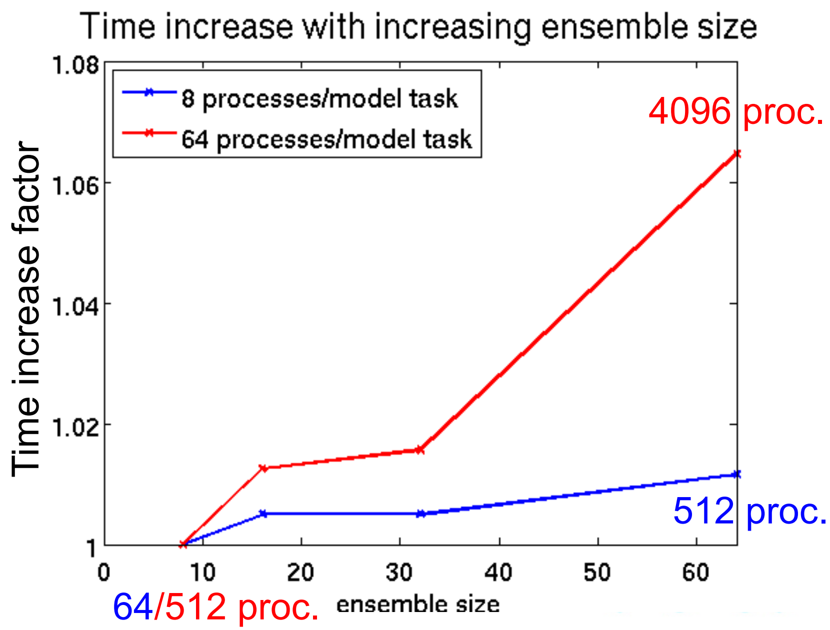
Speedup: Increase number of processes for each model task, fixed ensemble size

- 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



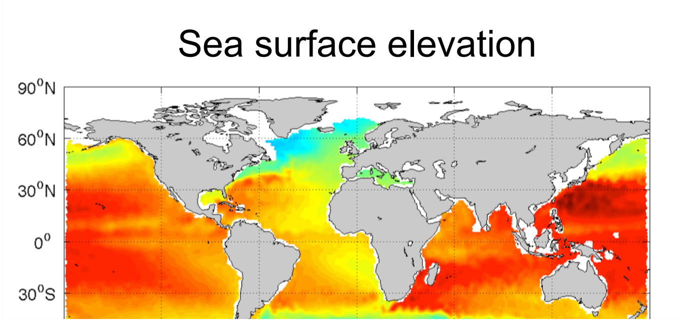


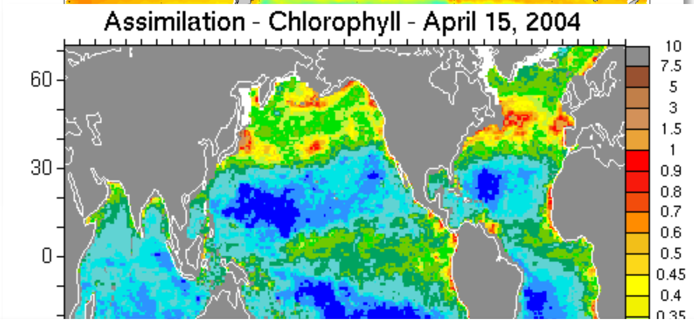
Application examples run with PDAF

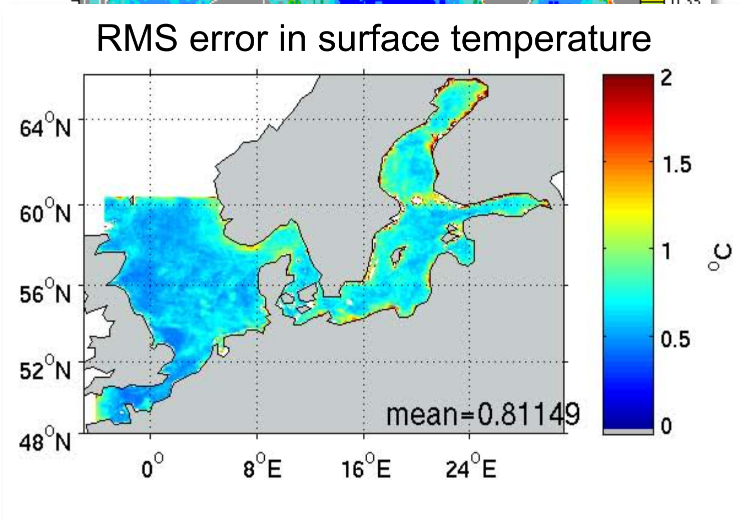
- Ocean state improvement by assimilation of satellite altimetry into global model
- Chlorophyll assimilation into global NASA Ocean Biogeochemical Model (with Watson Gregg, NASA GSFC)
- Coastal assimilation of ocean surface temperature
(S. Losa within project "DeMarine")

+ external users, e.g.

- NMEFC, China (Q. Yang)
- IPGP Paris (PARODY, A. Fournier)
- IFM HAMBURG, Germany
(MPI-OM, S. Brune/J. Baehr)
- U. Frankfurt (J. Tödter/B. Ahrens)







Summary

- Ensemble-based Kalman filters:
  - Current efficient methods suited for large-scale problems
  - Tuning of filters required
- Simplification of technical implementation using PDAF
- Application of the same assimilation software for test problems up to high-dimensional & operational systems

Thank you !