High-Dimensional Applications of Ensemble-Based Data Assimilation in Geosciences

Lars Nerger

Alfred Wegener Institute, Bremerhaven, Germany

Thanks to
Yuchen Sun, Sophie Vliegen

SIAM UQ24, Trieste, Italy, Feb. 27 – March 1, 2024

Overview

Application-centric viewpoint of ensemble-based data assimilation

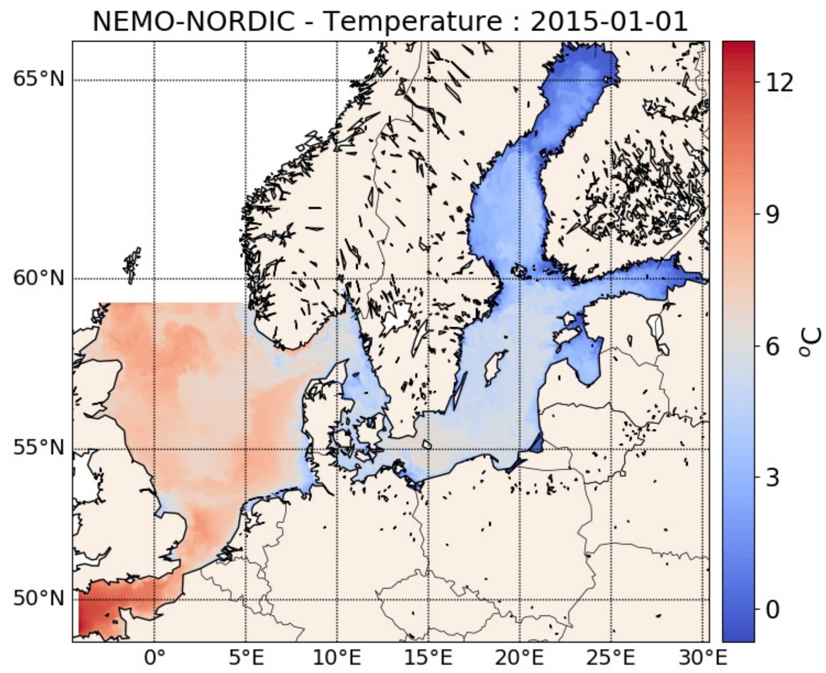
- Example of high-dimensional data assimilation application
- Methods
- Computational challenges
- Software
- Open points

High-dimensional Data Assimilation Application

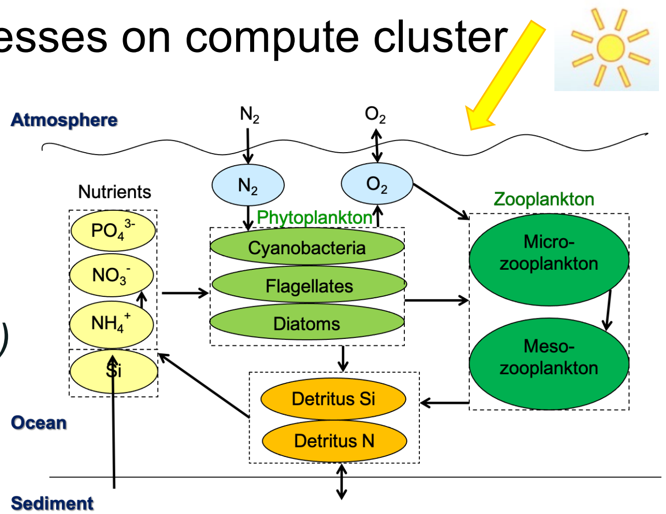
Model: NEMO-ERGOM

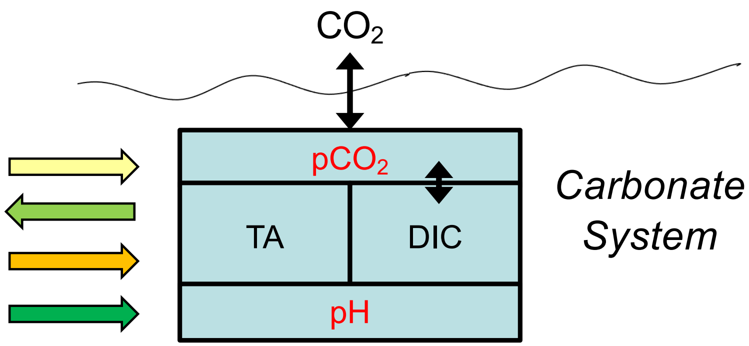
Operational configuration of Copernicus Marine Forecasting Center for Baltic Sea (BAL-MFC)

- Model setup
  - Ocean model NEMO
  - Coupled to ecosystem/carbon model ERGOM
  - 1.8 km resolution, 56 layers
  - Time step 90 sec
  - 192 processes on compute cluster



Ecosystem (biogeochemistry)





Carbonate System

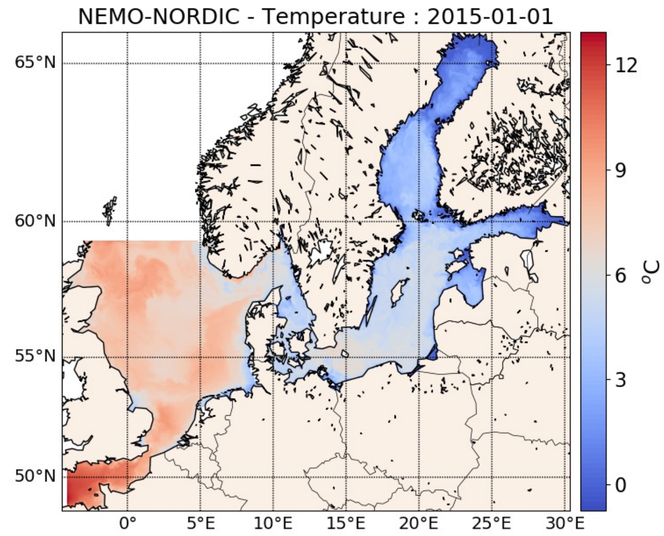
NEMO-ERGOM Coupled Data Assimilation

Ensemble States

- 5 physics variables
- 20 ecosystem variables
- State vector dimension: $704 \cdot 10^6$

Assimilation setup

- ensemble Kalman filter LESTKF
- ensemble size: 30
- Daily assimilation for 1 year
  - Observation data:
    surface temperature and chlorophyll concentration
- Localization: 20 km, Gaspari/Cohn function
- 186 x 30 = 5580 processors on compute cluster
- 12 days computing time
- Output approx. 900 GB (after compression)



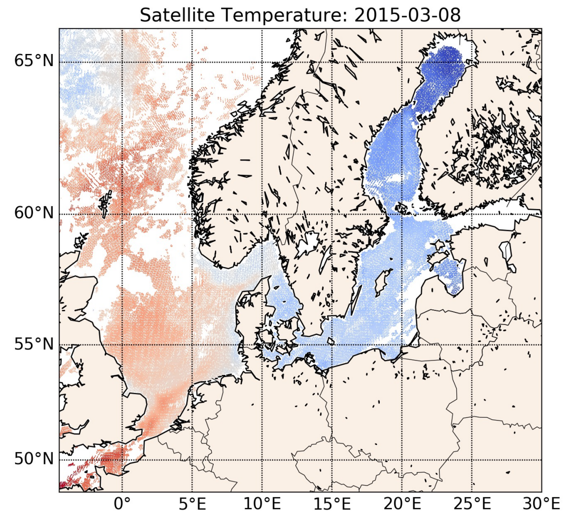
Costly to run, store and analyze outputs

This work has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101004032.

Observations

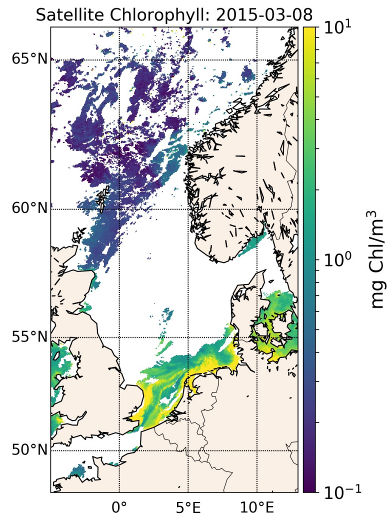
Sea Surface Temperature

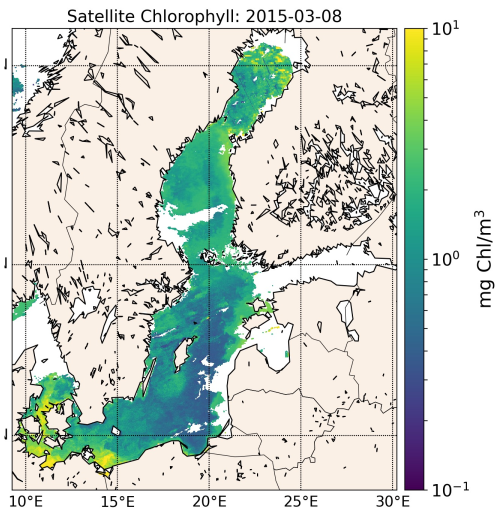
- Level 3 data from Copernicus Marine service (CMEMS)
- resolution 0.02°
- available daily
- No data in cloudy regions
- Number of observations: 17,000 – ~150,000
- observation error for DA: 0.8 °C
  (provided error fields not fully realistic)



Chlorophyll

- Level 3 data from CMEMS
- separate data products for North Sea and Baltic Sea
- resolution 1 km
- available daily
- No data in cloudy regions
- Number of observations: 0 – ~500,000
- observation error: relative error of 0.3





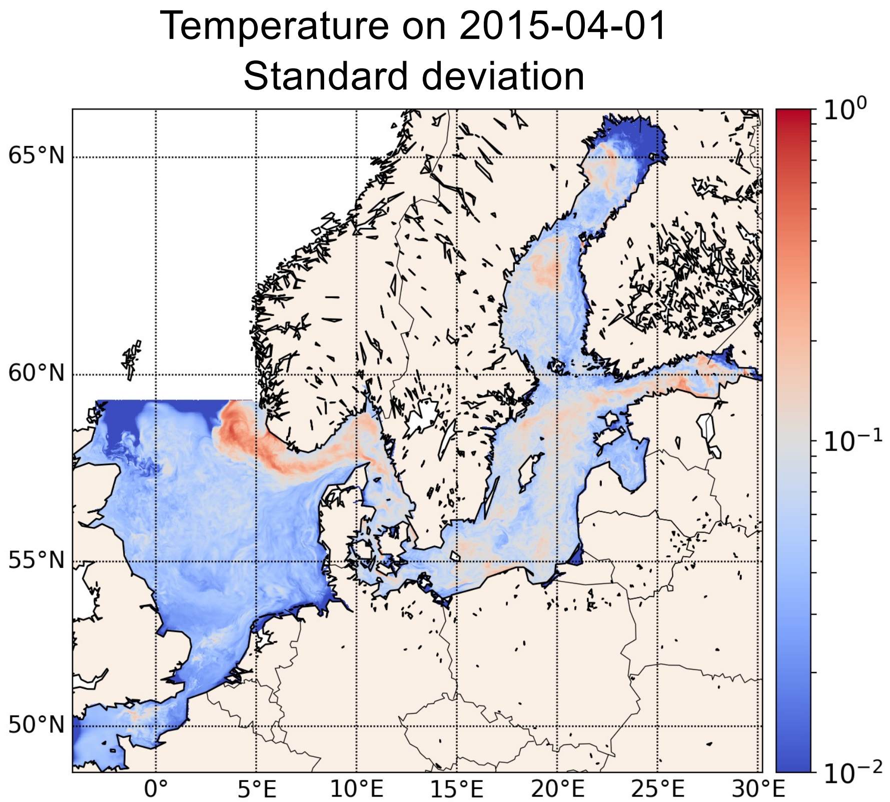
Methods

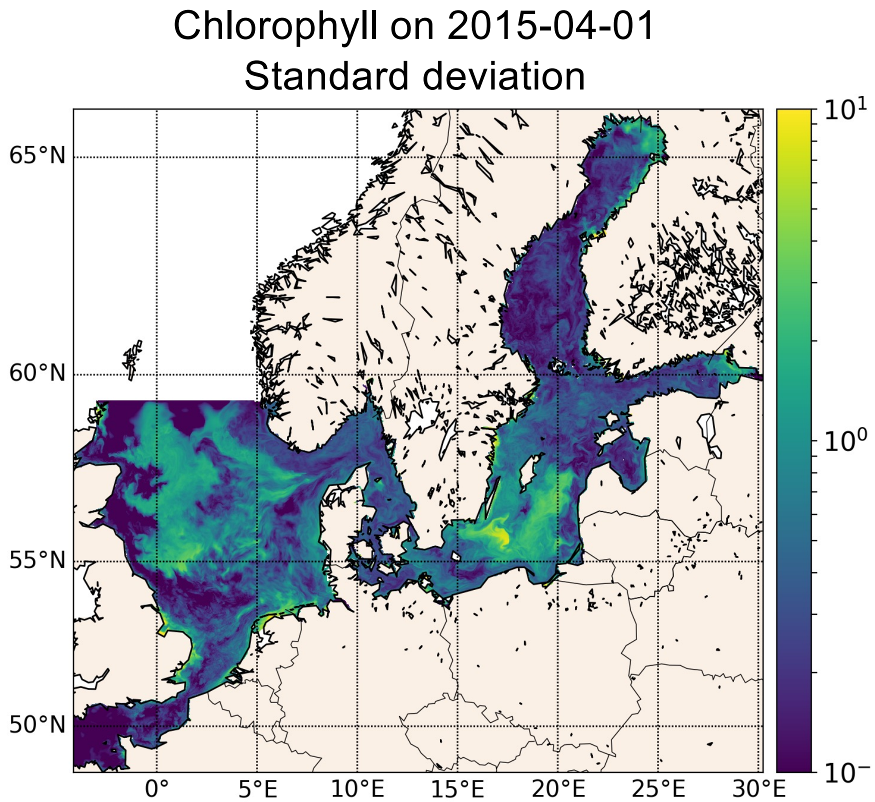
Ensemble-based data assimilation

Estimation by joining model and observational data

Uncertainty Estimates

Dynamic ensembles provide uncertainties (and covariances) for each day





Linear and Nonlinear Ensemble Filters

- Represent state and its error by ensemble $\mathbf{X}$ of N states (use ensemble perturbation matrix $\mathbf{X}^{'} = \mathbf{X} - \bar{\mathbf{X}}$)
- **Forecast**:
  - Integrate ensemble size N with numerical model
- **Analysis step:**
  - update ensemble mean $\overline{\mathbf{x}}^a = \overline{\mathbf{x}}^f + \mathbf{X}'^f \tilde{\mathbf{w}}$
  - update ensemble perturbations $\mathbf{X}'^a = \mathbf{X}'^f \mathbf{W}$

  (both can be combined in a single step)

Dimension of correction (error) space : N-1

- Ensemble Kalman & nonlinear filters: Different definitions of
  - weight vector $\tilde{\mathbf{w}}$ (dimension N)
  - Transform matrix $\mathbf{W}$ (dimension $N \times N$)

ETKF (Bishop et al., 2001)

- Ensemble Transform Kalman filter
  - Assume Gaussian distributions
  - Transform matrix

$$\mathbf{A}^{-1} = (N-1)\mathbf{I} + (\mathbf{H}\mathbf{X}'^f)^T\mathbf{R}^{-1}\mathbf{H}\mathbf{X}'^f$$

  - Mean update weight vector

$$\tilde{\mathbf{w}} = \mathbf{A}(\mathbf{H}\mathbf{X}'^f)^T\mathbf{R}^{-1}\left(\mathbf{y} - \mathbf{H}\overline{\mathbf{x}^f}\right)$$

(depends linearly on observation vector $\mathbf{y}$)

  - Transformation of ensemble perturbations

$$\mathbf{W} = \sqrt{N-1}\ \mathbf{A}^{1/2}\mathbf{\Lambda}$$

$\mathbf{\Lambda}$: mean-preserving random matrix or identity

Note: $\mathbf{W}$ depends only on $\mathbf{R}$, not observation $\mathbf{y}$

Algorithm designed for maximum computational efficiency

Excellent parallelization possibility when combined with localization

Linear filter:
- Gaussian distributions assumed
- Linear in effect of $\mathbf{y}$

NETF (Tödter & Ahrens, 2015)

- Nonlinear Ensemble Transform Filter
  - Mean update from Particle Filter weights:
    for Gaussian observation errors for all particles *i*

$$\tilde{w}^i \sim \exp\left(-0.5(\mathbf{y} - \mathbf{H}\mathbf{x}_i^f)^T \mathbf{R}^{-1} (\mathbf{y} - \mathbf{H}\mathbf{x}_i^f)\right)$$

(nonlinear function of observations $\mathbf{y}$)

  - Ensemble update
    - Transform ensemble to fulfill analysis covariance (like ETKF, but not assuming Gaussianity)
    - Derivation gives

$$\mathbf{W} = \sqrt{N}\left[\mathrm{diag}(\tilde{\mathbf{w}}) - \tilde{\mathbf{w}}\tilde{\mathbf{w}}^T\right]^{1/2} \Lambda$$

($\mathbf{\Lambda}$: mean-preserving random matrix; useful for stability)

Similar computational efficiency as ETKF

Excellent parallelization possibility when combined with localization

Nonlinear filter:
- No assumption of Gaussian distributions
- Nonlinear in $\mathbf{y}$

NETF is a second-order exact particle filter

 Tödter, J. and Ahrens, B. (2015) *Mon. Wea. Rev.* **143**,1347–1367

ETKF-NETF – Hybrid Filter Variants

Factorize the likelihood: ('tempering') $p(\mathbf{y}|\mathbf{x}) = p(\mathbf{y}|\mathbf{x})^{\gamma} p(\mathbf{y}|\mathbf{x})^{(1-\gamma)}$

- γ: hybrid weight (between 0 and 1; 1 for fully ETKF)

2-step updates

Variant 1 (*HNK*): NETF followed by ETKF

$$\tilde{\mathbf{X}}^a_{HNK} = \mathbf{X}^a_{NETF}[\mathbf{X}^f, (1-\gamma)\mathbf{R}^{-1}]$$
$$\mathbf{X}^a_{HNK} = \mathbf{X}^a_{ETKF}[\tilde{\mathbf{X}}^a_{HNK}, \gamma\mathbf{R}^{-1}]$$

- Both steps computed with increased $\mathbf{R}$ according to γ

Variant 2 (*HKN*): ETKF followed by NETF

Related methods:
Frei/Kuensch (2013)
Chustagulprom et al. (2016)
Robert et al. (2018)
Grooms/Robinson (2021)

Nerger, L. (2022) Q. J. R. Meteorol. Soc., **148**, 620-640 doi:10.1002/qj.4221

Choosing hybrid weight γ

- Hybrid weight shifts filter behavior

Some possibilities:

- Fixed value
- Adaptive - According to which condition?

Issue: Using N_{eff}
- only ensures non-collapsing ensemble
- does not ensure good analysis result
- Experimentally no obvious relation between N_{eff} and γ

  - Frei & Kuensch (2013) suggested using effective sample size $N_{eff} = \sum \frac{1}{(w^i)^2}$

(Usual choice for 'tempering')

    - γ_α : Choose γ so that N_{eff} is as small as possible but above minimum limit α (done iteratively)

  - Adaptive alternative

$$\gamma_{lin} = 1 - \frac{N_{eff}}{N_e}$$

(close to 1 if N_{eff} small; no iterations)

Effect of hybrid weight γ

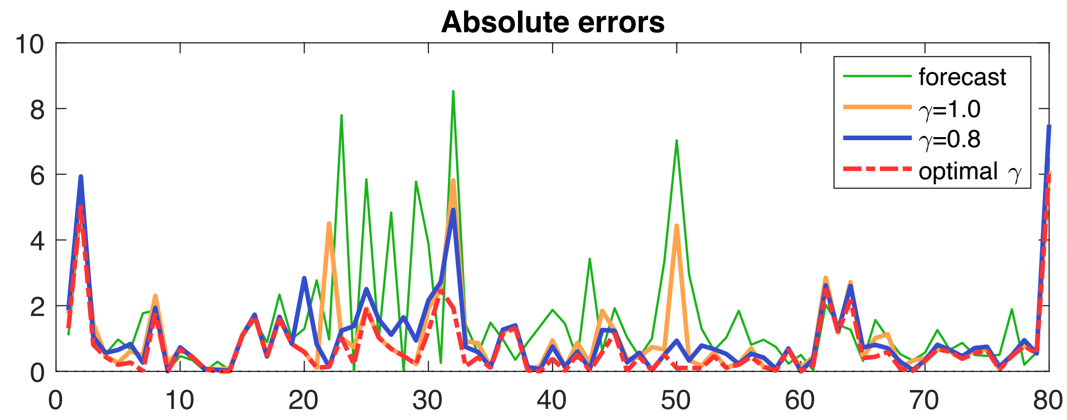
- Lorenz-96 model, size 80
- Examine single analysis step

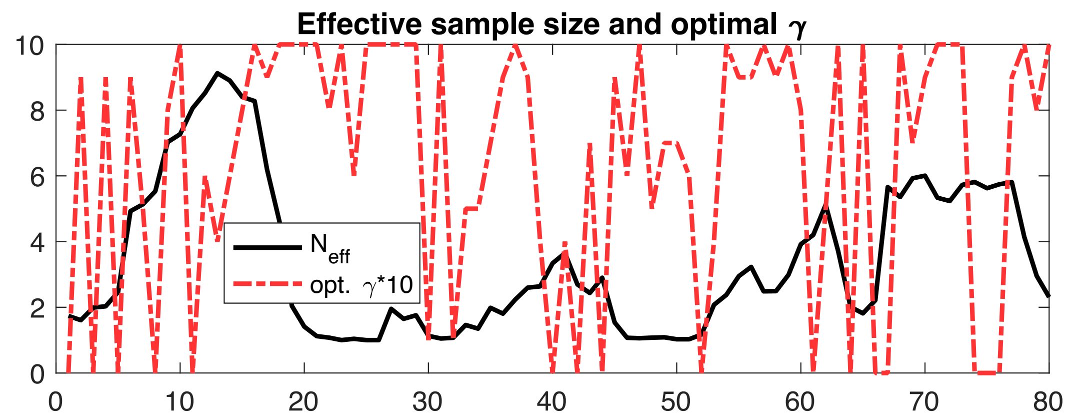
1. Run 33 analysis steps with γ=1 (LETKF)
2. Run analysis step 34 with one of
   a) γ=1
   b) γ=0.8
3. Examine N_{eff} and analysis errors

Additional experiment:

c) Adjust γ at each grid point to get minimum error

No obvious relation between N_{eff} and γ!





Account for non-Gaussianity: Skewness and Kurtosis

- **Mean** – 1st moment
- **Variance** – 2nd moment
- **Skewness** – 3rd moment

$$skew = \frac{\frac{1}{N_e}\sum_{i=1}^{N_e}\left(\mathbf{x}^i - \overline{\mathbf{x}}\right)^3}{\left[\frac{1}{(N_e-1)}\sum_{i=1}^{N_e}\left(\mathbf{x}^i - \overline{\mathbf{x}}\right)^2\right]^{3/2}}$$

- **Kurtosis** – 4th moment

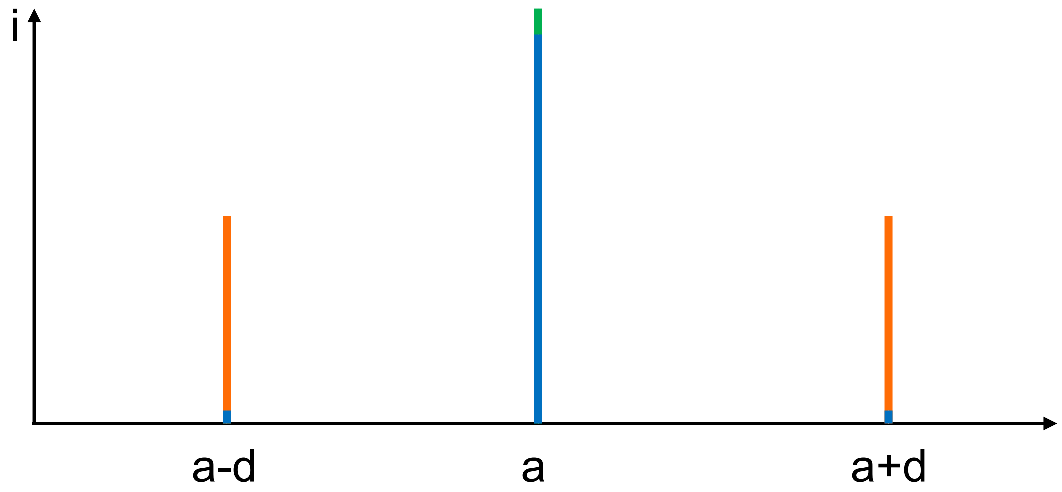
$$kurt = \frac{\frac{1}{N_e}\sum_{i=1}^{N_e}\left(\mathbf{x}^i - \overline{\mathbf{x}}\right)^4}{\left[\frac{1}{(N_e)}\sum_{i=1}^{N_e}\left(\mathbf{x}^i - \overline{\mathbf{x}}\right)^2\right]^{2}} - 3$$

- Skewness and kurtosis
  - generally not bounded
    - → but limits depend on ensemble size

Asymptotic properties of skewness and kurtosis

- Bounds of skewness and kurtosis depend on ensemble size
- Assess extreme cases

Case	Values	*skew* limit	*kurt* limit
max. skew	$x^{(1)} = a - d, \quad x^{(i)} = a,\ i = 2, \ldots, N_e$	$\sqrt{N_e}$	N_e
max. kurt	$x^{(1)} = a - d, \quad x^{(2)} = a + d, \quad x^{(i)} = a,\ i = 3, \ldots, N_e$	0	-2
min. kurt	$x^{(i)} = a - d,\ i = 1, \ldots N_e/2; \quad x^{(j)} = a + d,\ j = N_e/2 + 1, \ldots, N_e$	0	$N_e/2$



AWI

Using skewness and kurtosis to define hybrid weight γ

- Sampling errors are larger in NETF than ETKF
  → Always use ETKF for Gaussian (linear) cases
- Skewness and kurtosis describe deviation from Gaussianity
- mean absolute skewness (mas) and kurtosis (mak) of observed ensemble (with localization: use locally assimilated observations)
- Use normalized means:

$$nmas = \frac{1}{\sqrt{\kappa}} mas \qquad nmak = \frac{1}{\kappa} mak$$

standard value: $\kappa = N_e$

Now define

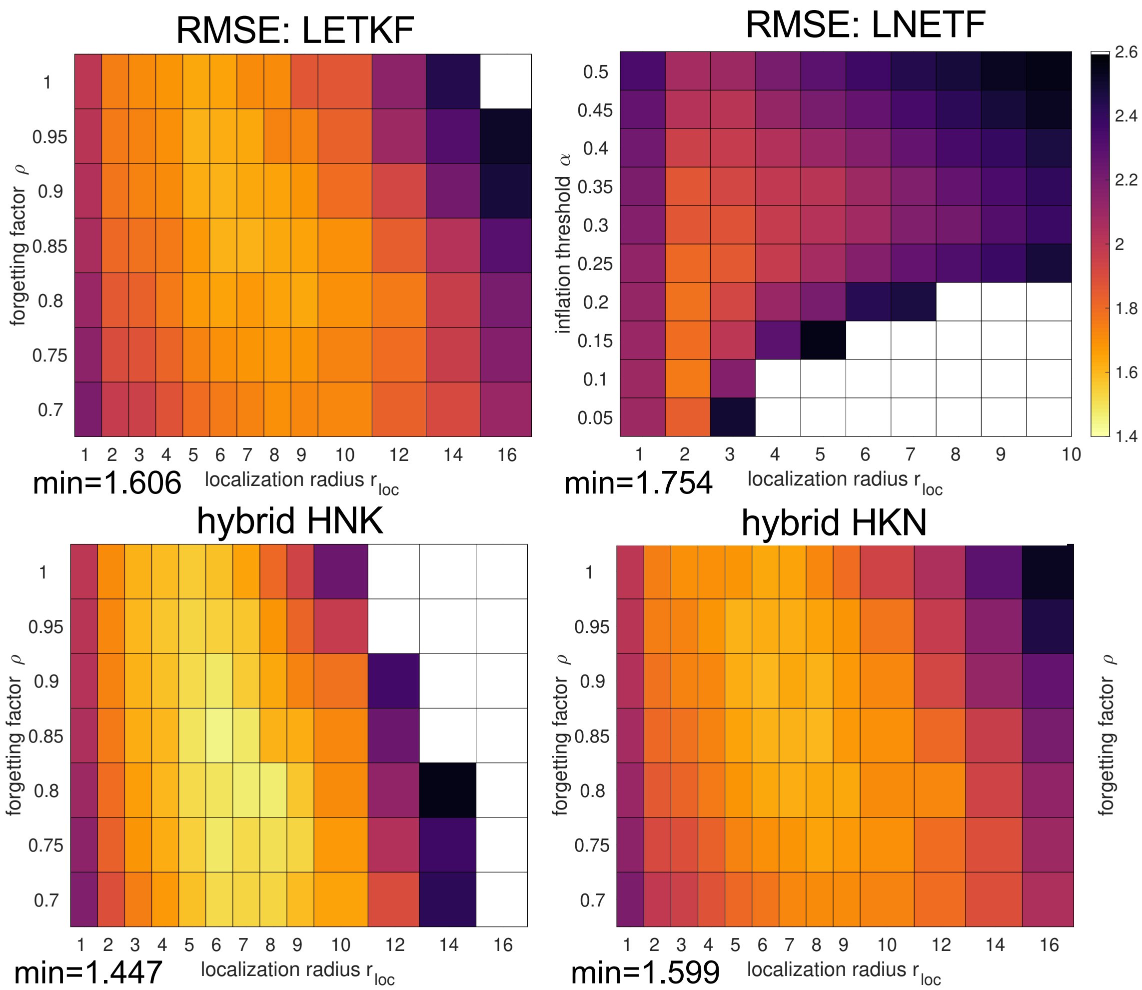
$$\gamma_{sk,\alpha} = \max\left[\min(1 - nmak,\ 1 - nmas),\ \gamma_\alpha\right]$$
$$\gamma_{sk,lin} = \max\left[\min(1 - nmak,\ 1 - nmas),\ \gamma_{lin}\right]$$

stronger influence of $nmas$ and $nmak$ limited by N_{eff}

Note: There are sampling errors, e.g. for skewness $\sigma_{skew} \sim \sqrt{6/N_e}$
→ For N_e=25: ~10% error in γ

Test with Lorenz-96 model

Ensemble size 15; Forecast length: 8 time steps; 20 observations



Strongly nonlinear DA setting

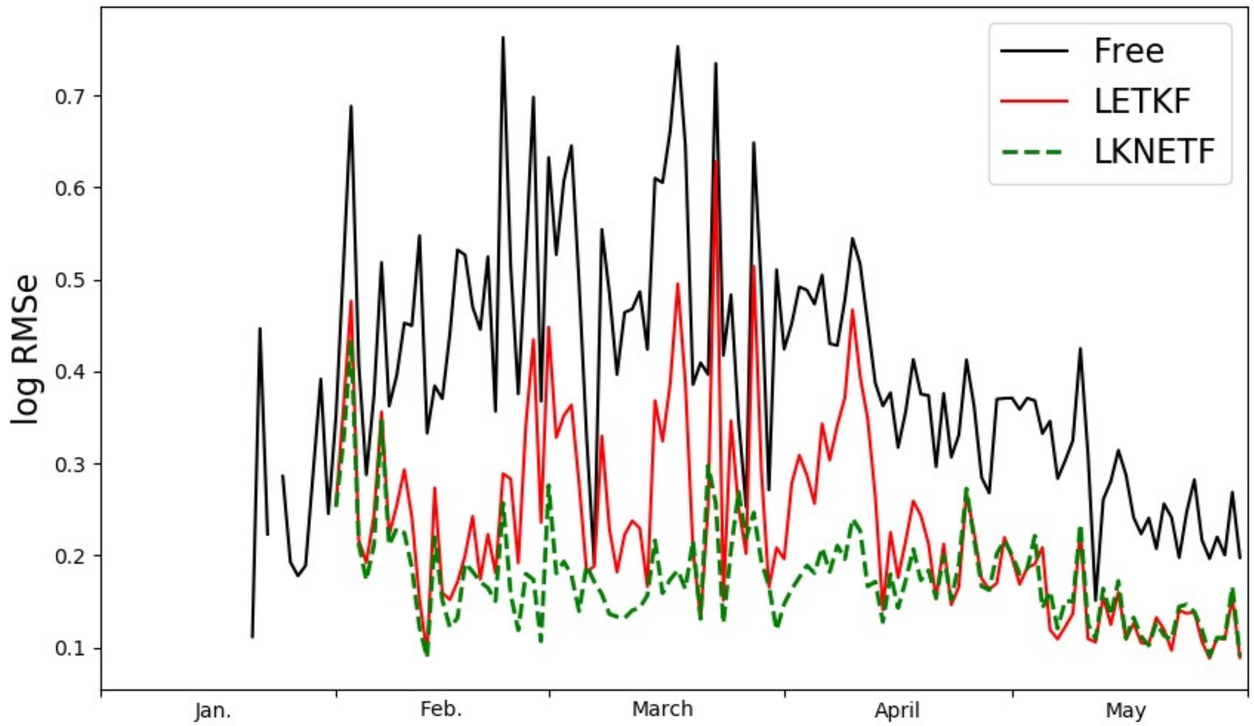
- Show RMS errors as function of inflation (forgetting factor or α) and localization radius
- *Smallest errors:* Hybrid HNK (10% error reduction)

  → hybrid filter able to utlize non-Gaussian information
- Other hybrid variants also improve the state estimate

Effect of hybrid filter in high-dimensional application

Assimilation using rule $\gamma_{sk,\alpha}$

RMS deviation for log-Chlorophyll



Only assimilate chlorophyll observations

Stronger assimilation effect of LKNETF

We still don't know optimal choice of rule for γ

Computational challenges

Computing challenges and features

High-dimensional models

- Costly to compute
- Large amount of output data
- Large size of state vectors
  - Containing all relevant model fields
  - Usually distributed due to parallelization

Ensemble-based data assimilation

- Multiply computing cost (parallel or sequential)
- Full ensemble output would multiply amount of output data
  - Usually only write ensemble mean and variance
- Computing time of model dominates over assimilation method

Software

PDAF: Parallel Data Assimilation Framework

PDAF Parallel Data Assimilation Framework

A unified tool for interdisciplinary data assimilation …

- a program library for data assimilation
- provide support for parallel ensemble forecasts
- provide assimilation methods – fully-implemented & parallelized
- provide tools for observation handling and for diagnostics
- easily useable with (probably) any numerical model
  (coupled to with range of models)
- run from laptops to supercomputers (Fortran, MPI & OpenMP)
- Usable for real assimilation applications and to study assimilation methods
- ensure separation of concerns (model – DA method – observations – covariances)

Open source:
Documentation and tutorial at

http://pdaf.awi.de

github.com/PDAF

Python interface:

https://github.com/yumengch/pyPDAF

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

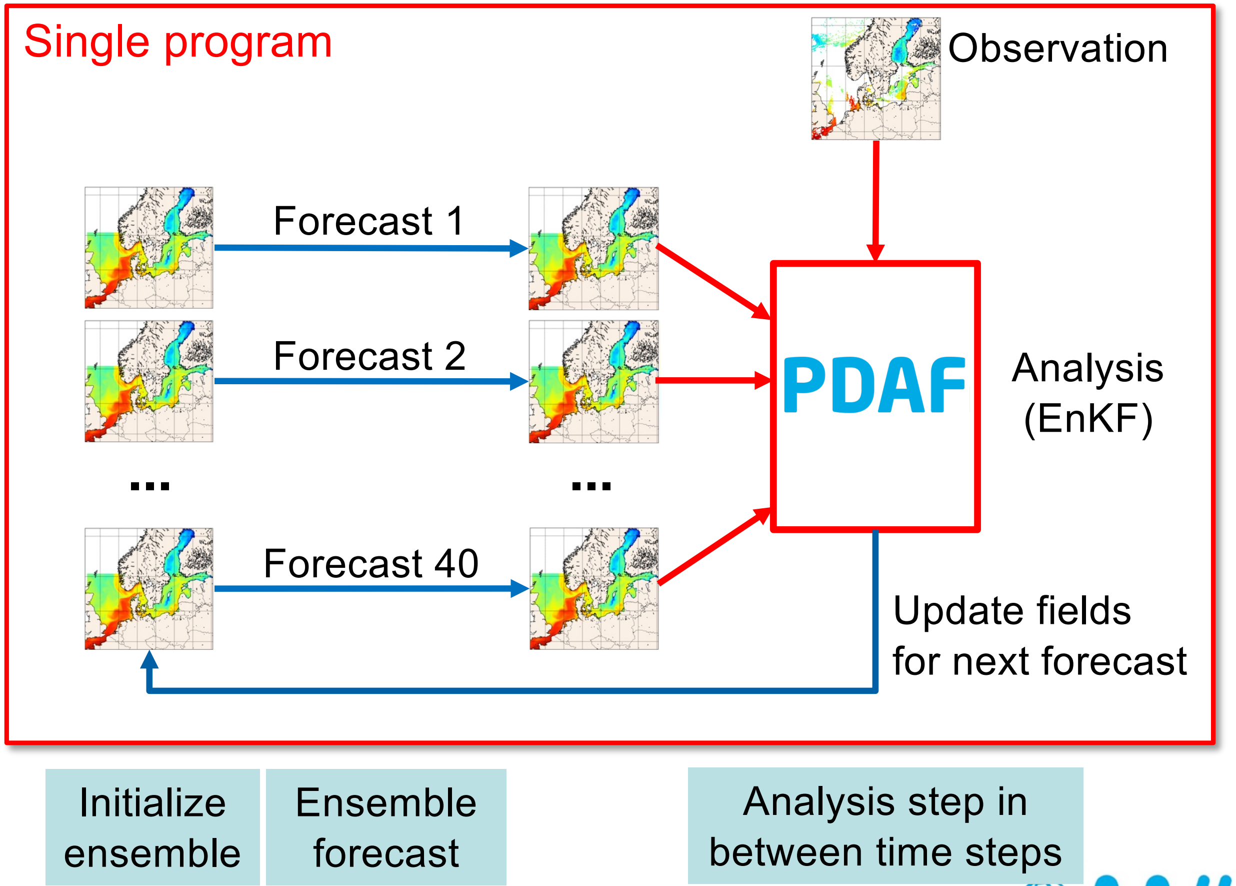
AWI

Online-Coupling – Assimilation-enabled Model

Couple a model with PDAF

- Modify model to simulate ensemble of model states
- Insert analysis step/solver to be executed at prescribed interval
- Run model as usual, but with more processors and additional options

- EnOI and 3D-Var also possible:
  - Evolve single model state
  - Prescribe ensemble perturbations or covariance



Open Points

Open points

Characteristics of problem
– merging observations and models in Geosciences

- Observations
  - High count – O(10^5 - 10^7)
  - Only a few of the model variables
  - Incomplete spatial and temporal coverage
  - Significant errors (measurement and representation)

- Models
  - Large state vectors – (10^6 - 10^9)
  - Costly to run
  - ‘balances’
  - Limited previous model runs available

Open points – linkage a 'modern' ML methods

Potential for improvements with novel ML methods

Reduce execution time

- Dominated by model!
- Faster DA method of little help

Surrogates might help, but

- Costly to build
- Unclear if sufficiently representative (need current representation of error)

Reduce time to tune method

- Tuning required and costly

Do pre-trained neural networks help?

Improve estimates

- Avoid assumptions on distributions (or avoid distributions)
- Avoid sampling errors
- Ensure small state changes to avoid disturbing 'balances'

Summary

PDAF Parallel Data Assimilation Framework

- High-dimensional application in geosciences
  - Costly to compute & large amount of outputs
  - Incompletely observed and with significant errors

- Current standard method basing on optimization or estimation
  - suffer from sampling errors and costly tuning

- Potential of new methods
  - reduce computing time
  - Reduce tuning effort
  - Improve estimates