GMAO, NASA/GSFC, Greenbelt, MD, February 8, 2018

A Hybrid Kalman-Nonlinear Ensemble Transform Filter

Lars Nerger

Alfred Wegener Institute, Bremerhaven, Germany

With inputs from

Paul Kirchgessner, Julian Tödter, Bodo Ahrens

on the NETF

SANGOMA

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

Overview

- Linear and nonlinear filters
- Nonlinear Ensemble Transform Filter – NETF (Tödter & Ahrens, MWR, 2015)
- Hybrid LETKF-NETF method
  for improved assimilation with small ensembles

Kalman and Nonlinear Filters

Ensemble filters – ensemble Kalman filters & NETF

- represent state and its error by ensemble $\mathbf{X}$ of N states
- Forecast:
  - Integrate ensemble with numerical model
- Analysis:
  - 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)

- Ensemble Kalman filters & NETF: Different definitions of
  - weight vector $\tilde{\mathbf{w}}$
  - Transform matrix $\mathbf{W}$

ETKF (Bishop et al., 2001)

- Ensemble Transform Kalman filter:
  - Transform matrix

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

  - Mean update weight vector

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

    (depends on $\mathbf{R}$ and $\mathbf{y}$)

  - Transformation of ensemble perturbations

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

    (depends only on $\mathbf{R}$, not $\mathbf{y}$)

Particle filters – fully nonlinear ensemble filters

- Avoid changing ensemble members ('particles')
- Instead: give particles a weight at change it at the analysis step
  - Initial weight: 1/N for all particles
- Weights are given by statistical likelihood of an observation
- Example: With Gaussian observation errors (for each particle *i*):

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

- Ensemble mean state computed with weights

$$\overline{\mathbf{x}}^a = \overline{\mathbf{x}}^f + \mathbf{X}'^f\tilde{\mathbf{w}} = \mathbf{X}^f\tilde{\mathbf{w}}$$

- This update does not assume any distribution of the state errors (and is not limited to Gaussian distributions)

Nonlinear Ensemble Transform Filter - NETF

- Ensemble Kalman:
  - Transformation according to KF equations

- NETF (Tödter & Ahrens, MWR, 2015)
  - Mean update from Particle Filter weights: for all particles *i*

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

  - Ensemble update
    - Transform ensemble to fulfill analysis covariance (like KF, 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)

Tödter, J. and Ahrens, B. 2015. A second-order exact ensemble square root filter for nonlinear data assimilation. *Mon. Wea. Rev.* **143**,1347–1367

Derivation of NETF

- Mean state update

$$\overline{\mathbf{x}}^a = \overline{\mathbf{x}}^f + \mathbf{X}'^f \tilde{\mathbf{w}} = \mathbf{X}^f \tilde{\mathbf{w}}$$

- Analysis covariance matrix

$$\mathbf{P}^a = \sum_{i=1,N} \tilde{w}_i \, (\mathbf{x}_i^f - \overline{\mathbf{x}}^a)(\mathbf{x}_i^f - \overline{\mathbf{x}}^a)^T$$

$$\mathbf{P}^a = \frac{1}{N}\mathbf{X}^f \mathbf{W}^2 (\mathbf{X}^f)^T$$

with

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

Difference of ETKF and NETF

- ETKF parameterizes ensemble distribution by a Gaussian distribution
- NETF uses particle filter weights to ensure correct update of ensemble mean and covariance

- Filter update:
  - in ETKF is linear in observations

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

  - in NETF is nonlinear in observations

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

Ensemble Smoothers – ETKS & NETS

- Smoother: Update past ensemble with future observations
- Rewrite ensemble update as
  - Filter:

$$\mathbf{X}^a_{k|k} = \mathbf{X}^f_{k|k-1}\hat{\mathbf{W}}_k$$

analysis time

Observations used up to time

- Smoother at time $i < k$

$$\mathbf{X}^a_{i|k} = \mathbf{X}^f_{i|k-1}\hat{\mathbf{W}}_k$$

➢ works likewise for ETKS and NETS

➢ also possible for localized filters

See, e.g., Nerger, Schulte & Bunse-Gerstner, QJRMS 140 (2014) 2249–2259

Filter performance of NETF

NETF

with small Lorenz-96 model

Configuration of Lorenz-96 model experiments

Lorenz-96:

- 1-dimensional period wave
- Chaotic dynamics

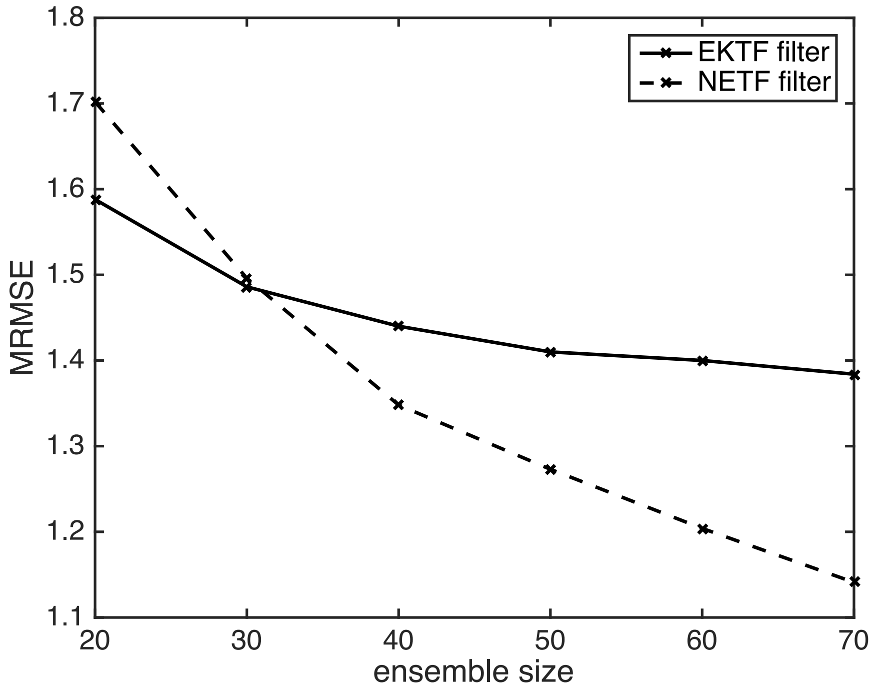
Configuration for assimilation experiments

- State dimension: 80
- Observed: 40 grid points
- Time steps between analysis steps: 8
- Double-exponential observation errors (stronger nonlinearity)
- Experiment length: 5000 time steps
- Observation error standard deviation: 1.0

➜ this is a difficult case for the assimilation
(and more realistic than typical 1-step forecast configuration)

Performance of NETF – Lorenz-96

- Double-exponential observation errors
- Run all experiments 10x with different initial ensemble

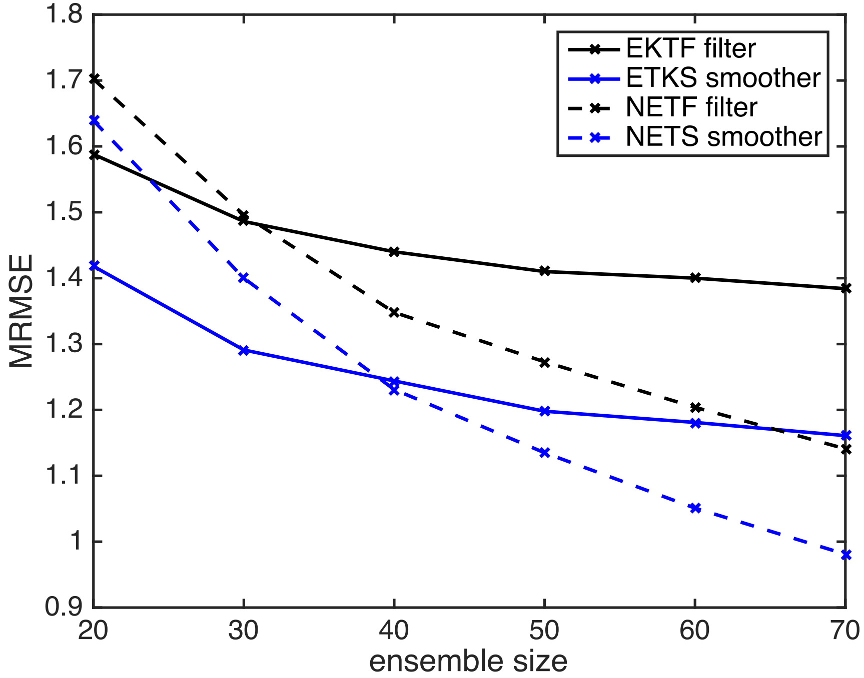


- NETF beats ETKF for ensemble size > 30

Kirchgessner, Tödter, Ahrens, Nerger. (2017) The smoother extension of the nonlinear ensemble transform filter. Tellus A, **69**, 1327766

Performance of NETF – Lorenz-96

- Performance for small model (Lorenz-96)



- Blue: Smoother
- NETS beats ETKS for ensemble size 40 and larger
  - Smoother slightly stronger for ETKS
  - NETF better than ETKF smoother for N=70

Kirchgessner, Toedter, Ahrens, Nerger. (2017) Tellus A 69:1, 1327766

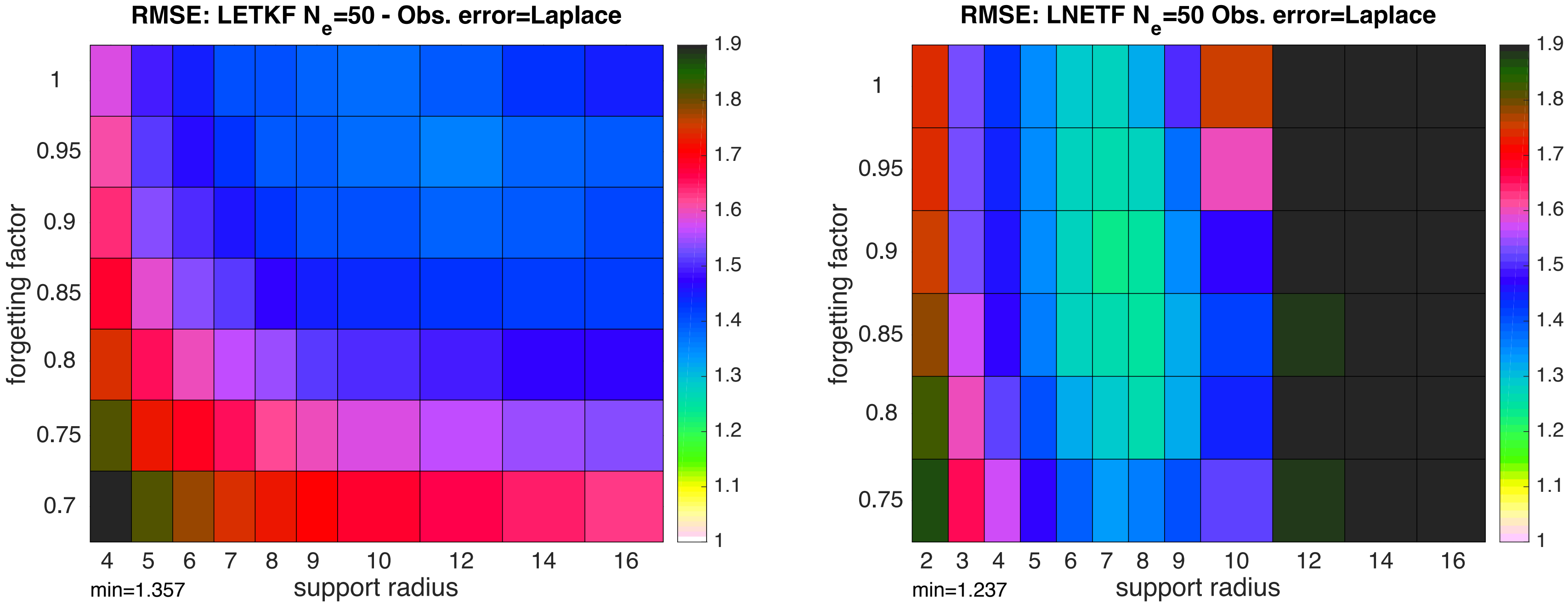
Parameter stability of NETF

RMS error varying

- inflation (forgetting factor)
- localization radius

For N=50 and Laplace observation errors

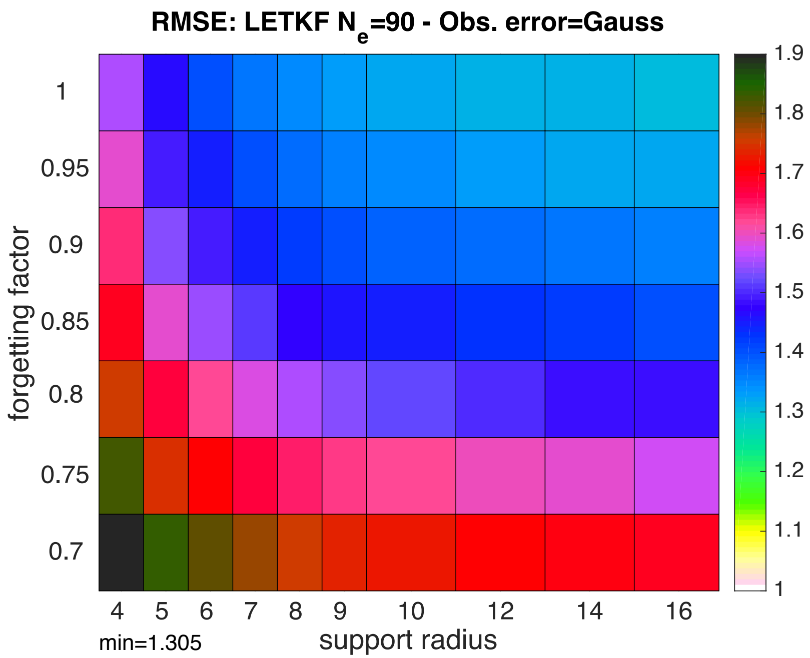
- Smaller error for NETF
- Smaller parameter region for low errors

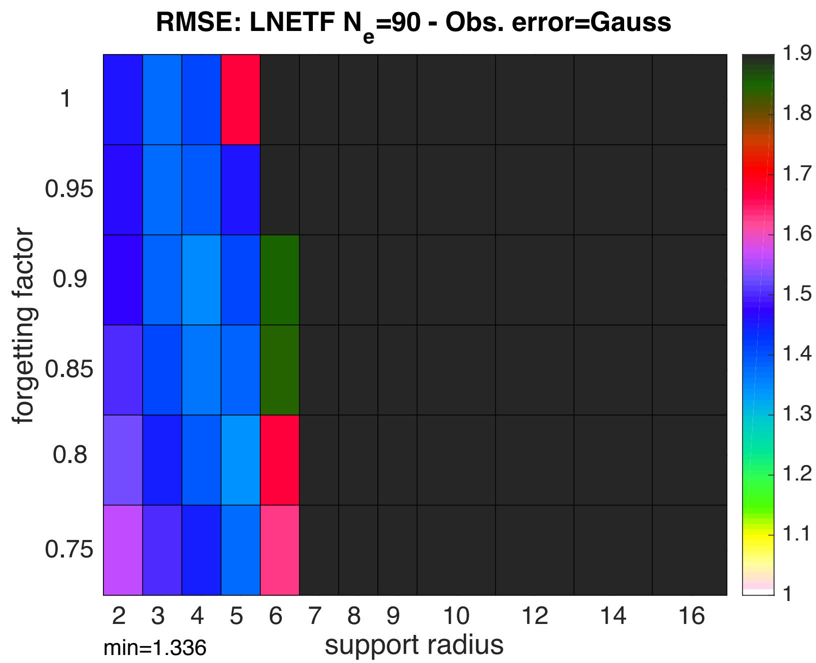


NETF with Gaussian observation errors

For Gaussian observation errors

- Need N=90 for comparable RMS errors
- NETF needs much smaller localization radius





NETF

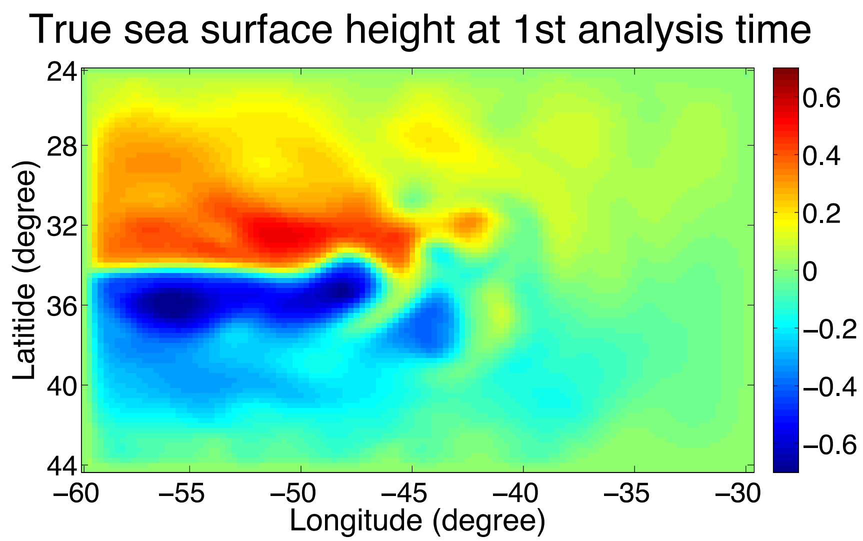
with
high-dimensional ocean model

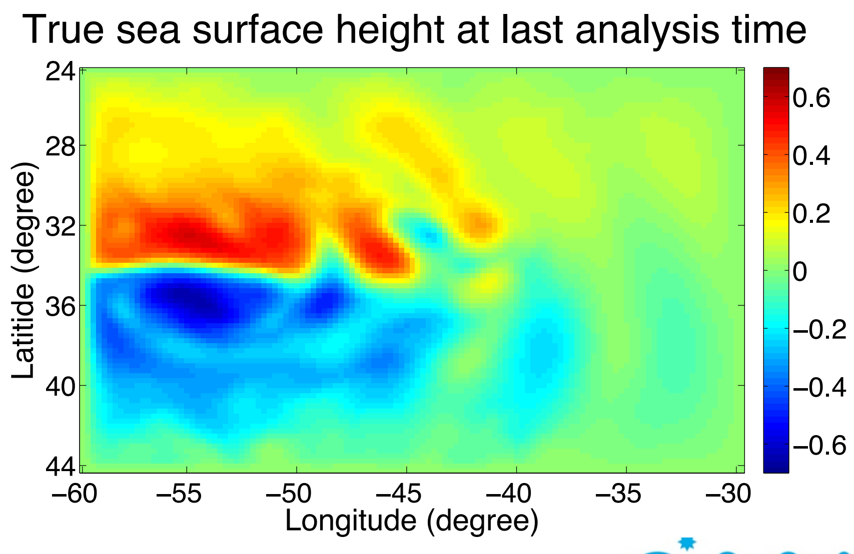
Assimilation into NEMO

European ocean circulation model

Model configuration

- box-configuration “SEABASS”
- ¼° resolution
- 121x81 grid points, 11 layers
  (state vector ~300,000)
- wind-driven double gyre
  (a nonlinear jet and eddies)
- medium size SANGOMA
  benchmark





SANGOMA

www.data-assimilation.net

PDAF: A tool for data assimilation

PDAF Parallel Data Assimilation Framework

PDAF - Parallel Data Assimilation Framework

- a program library for ensemble data assimilation
- provide support for parallel ensemble forecasts
- provide fully-implemented & parallelized filters and smoothers (EnKF, LETKF, NETF, EWPF … easy to add more)
- easily useable with (probably) any numerical model (applied with NEMO, MITgcm, FESOM, HBM, TerrSysMP, …)
- run from laptops to supercomputers (Fortran, MPI & OpenMP)
- first public release in 2004; continued development
- ~280 registered users; community contributions

Open source:
Code, documentation & tutorials at

http://pdaf.awi.de

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

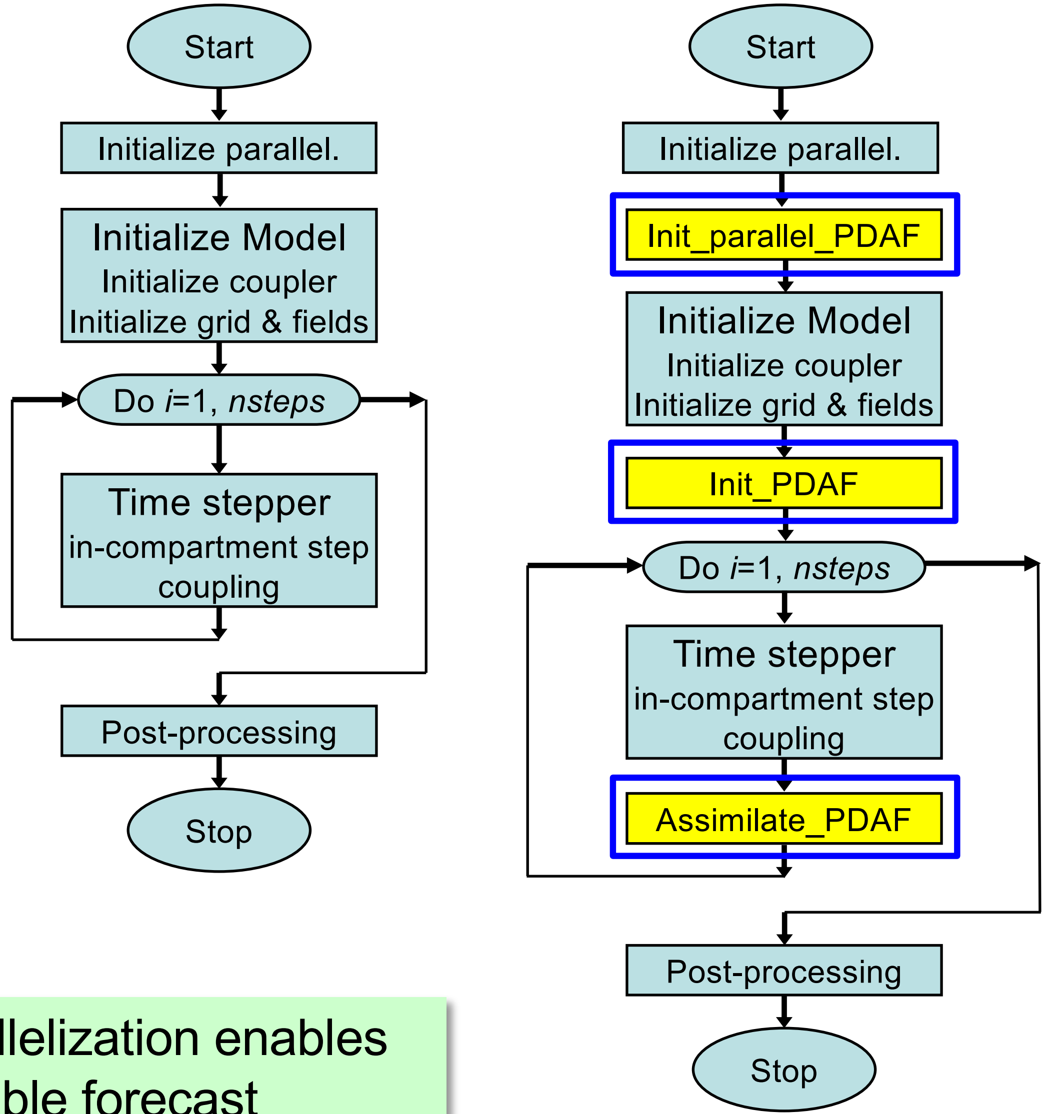
AWI

Extending a Model for Data Assimilation

Model

single or multiple executables

coupler might be separate program



Extension for data assimilation

plus:
Possible model-specific adaption:

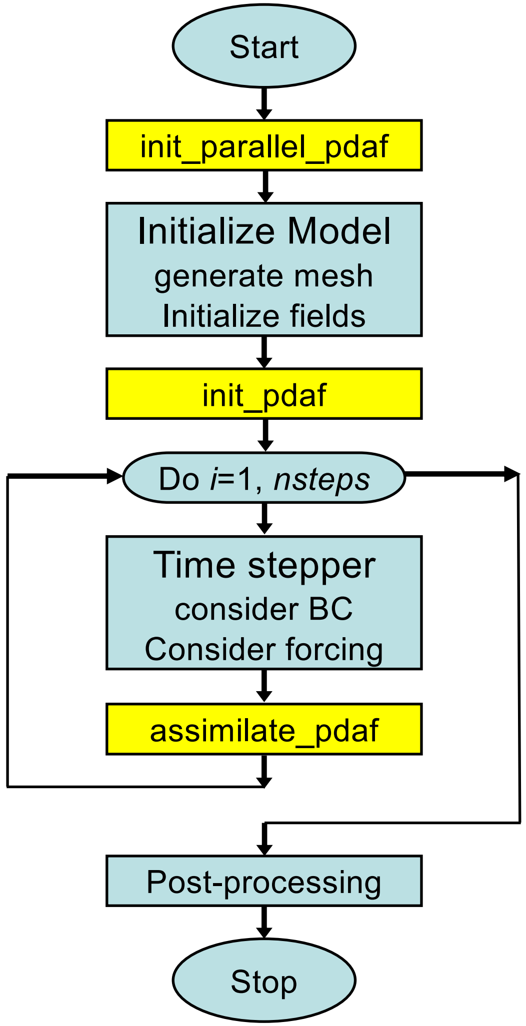
for NEMO: handle leapfrog time stepping

revised parallelization enables ensemble forecast

Features of online-coupled DA 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



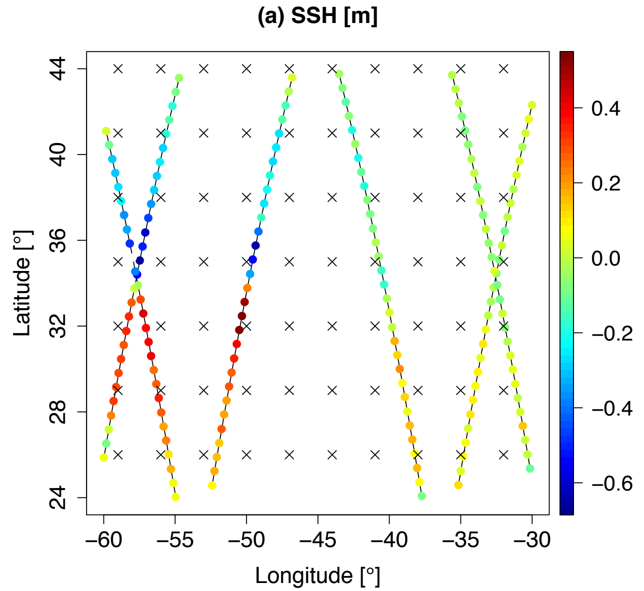
Observations and Assimilation Configuration

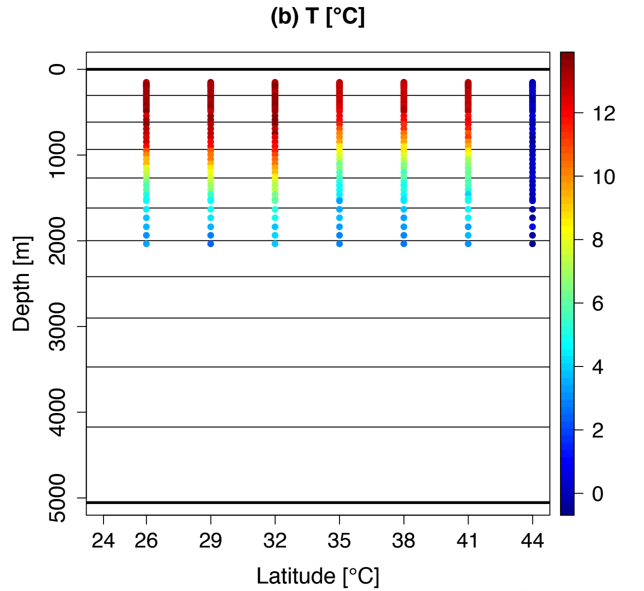
Observations

- Simulated satellite sea surface height SSH (Envisat & Jason-1 tracks), 5cm error
- Temperature profiles on 3ºx3º grid, surface to 2000m, 0.3ºC error

Data Assimilation

- Ensemble size: 120
- LETKF, LNETF
- Localization: weights on matrix $\mathbf{R}^{-1}$ (Gaspari/Cohn'99 function, 2.5º radius)
- Assimilate each 48h over 360 days



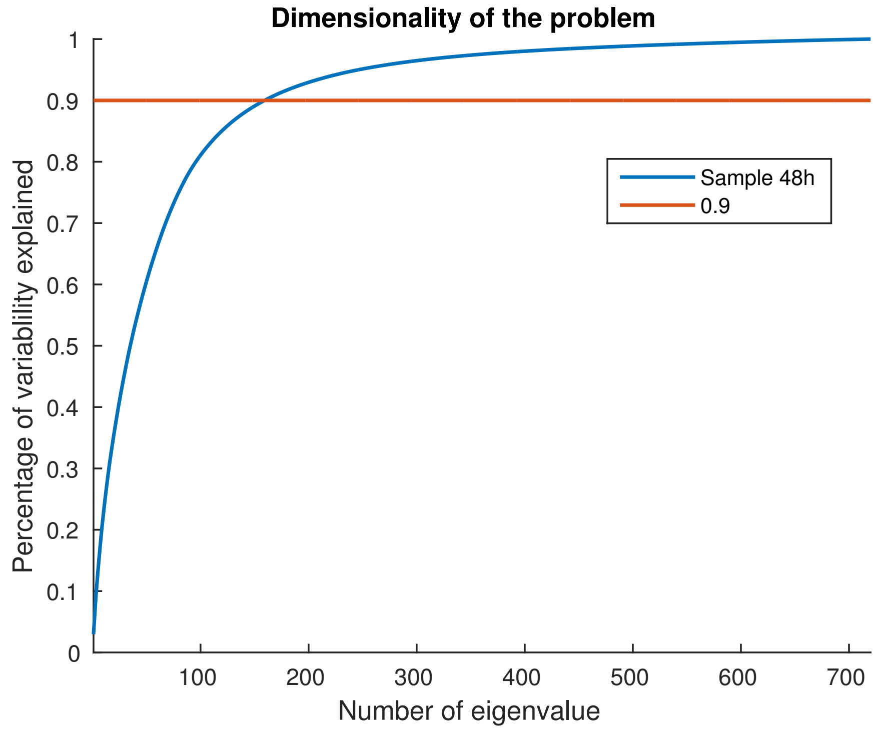


Dimensions of the problem

State vector dimension ~300,000

Dimension of dynamics (error space):

From eigenvalue decompositions (EOFs)

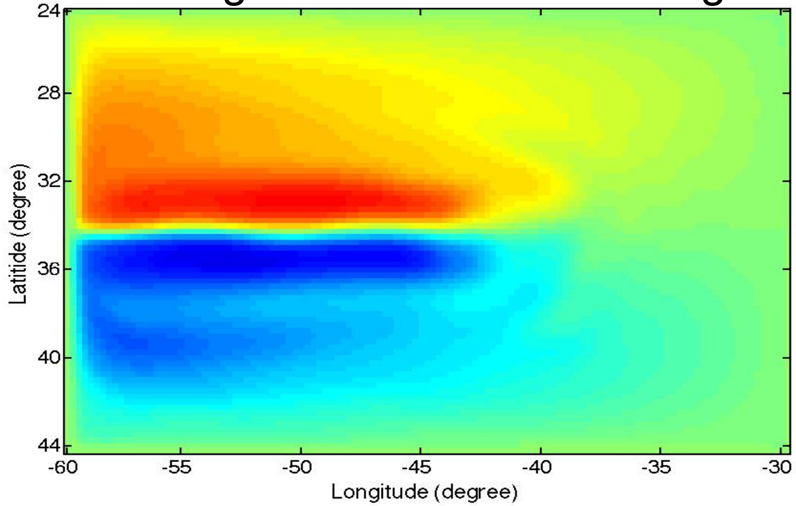


~180 modes for 90% of variability

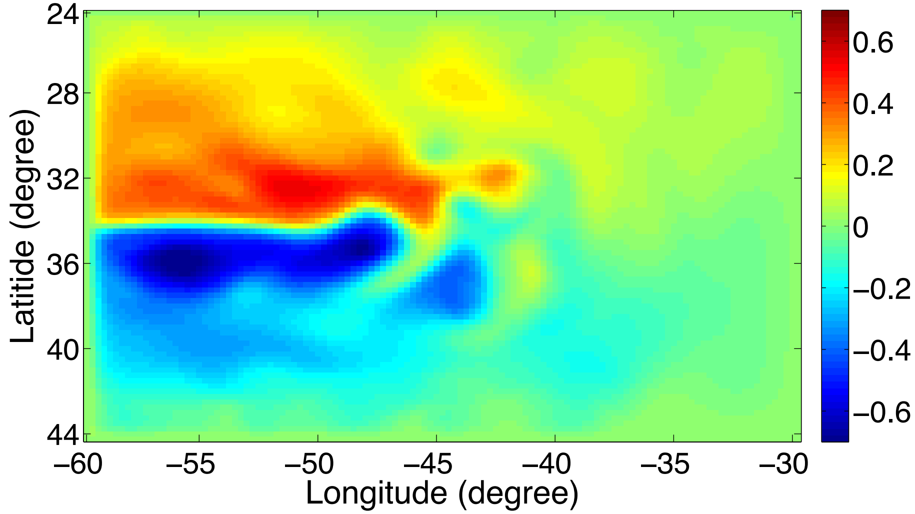
~400 modes for 99.9% of variability

Application of LETKF

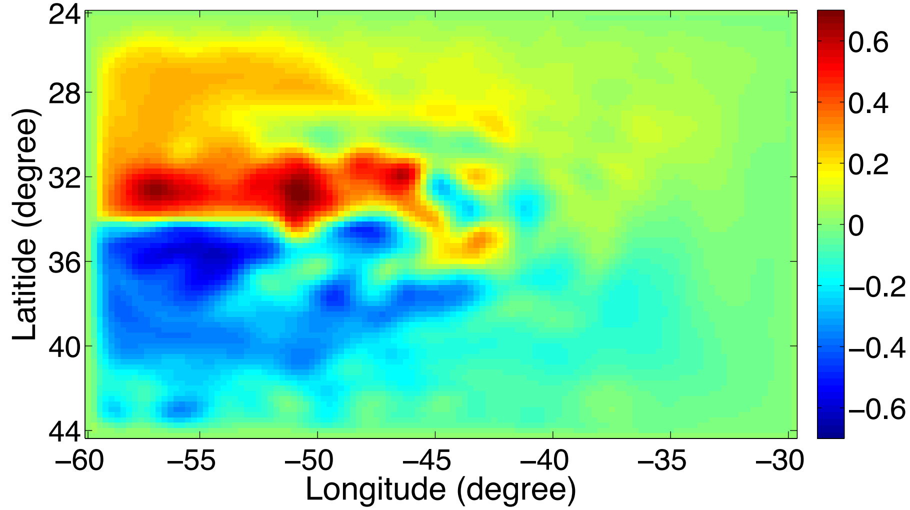
Initial guess of sea surface height



True sea surface height at 1st analysis time

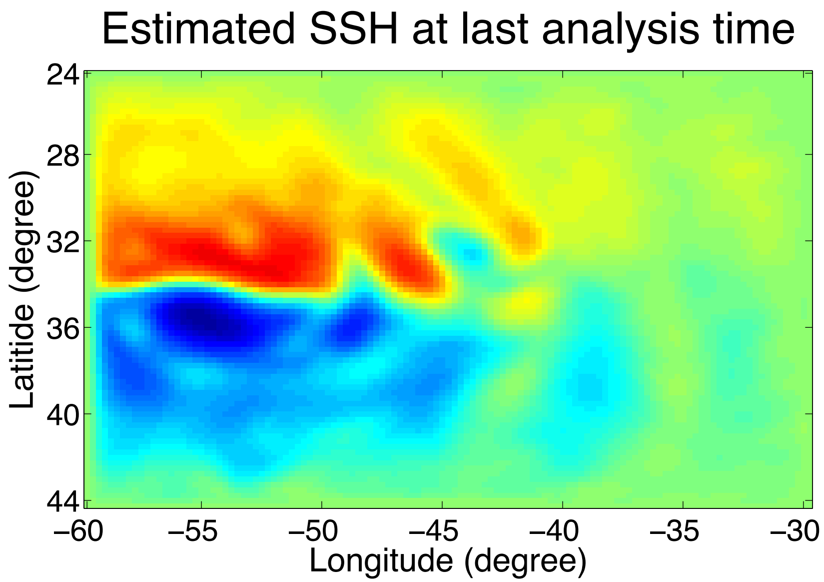


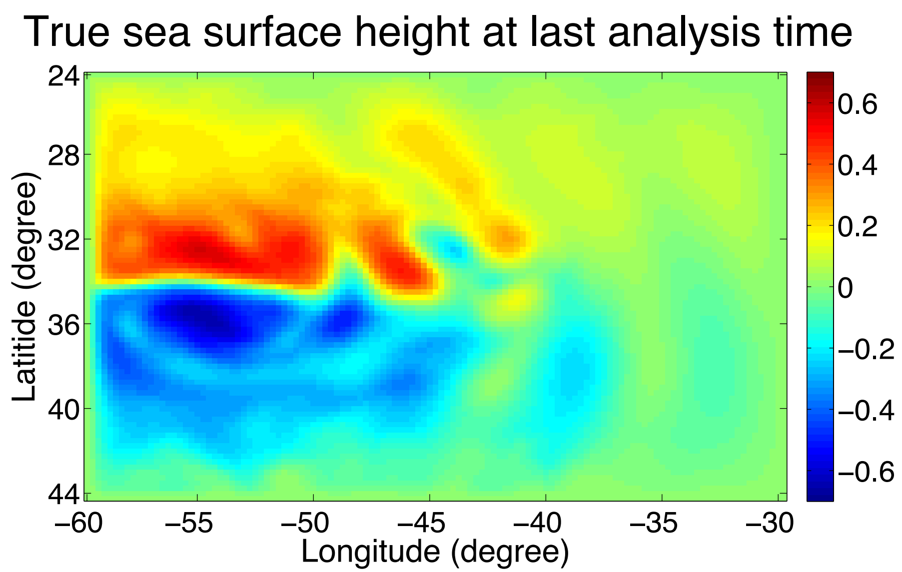
Estimated SSH at 1st analysis time

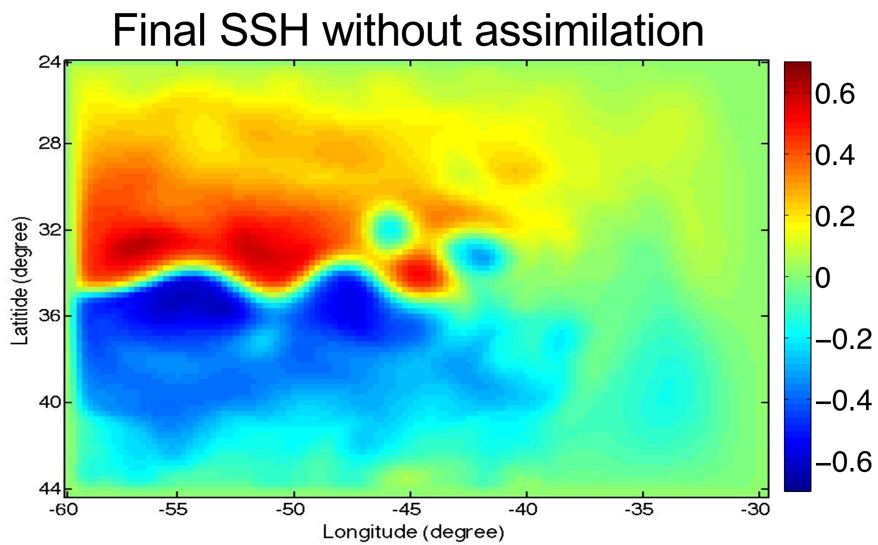


Hybrid Kalman-Nonlinear Ensemble Transform Filter

Application of LETKF (2)

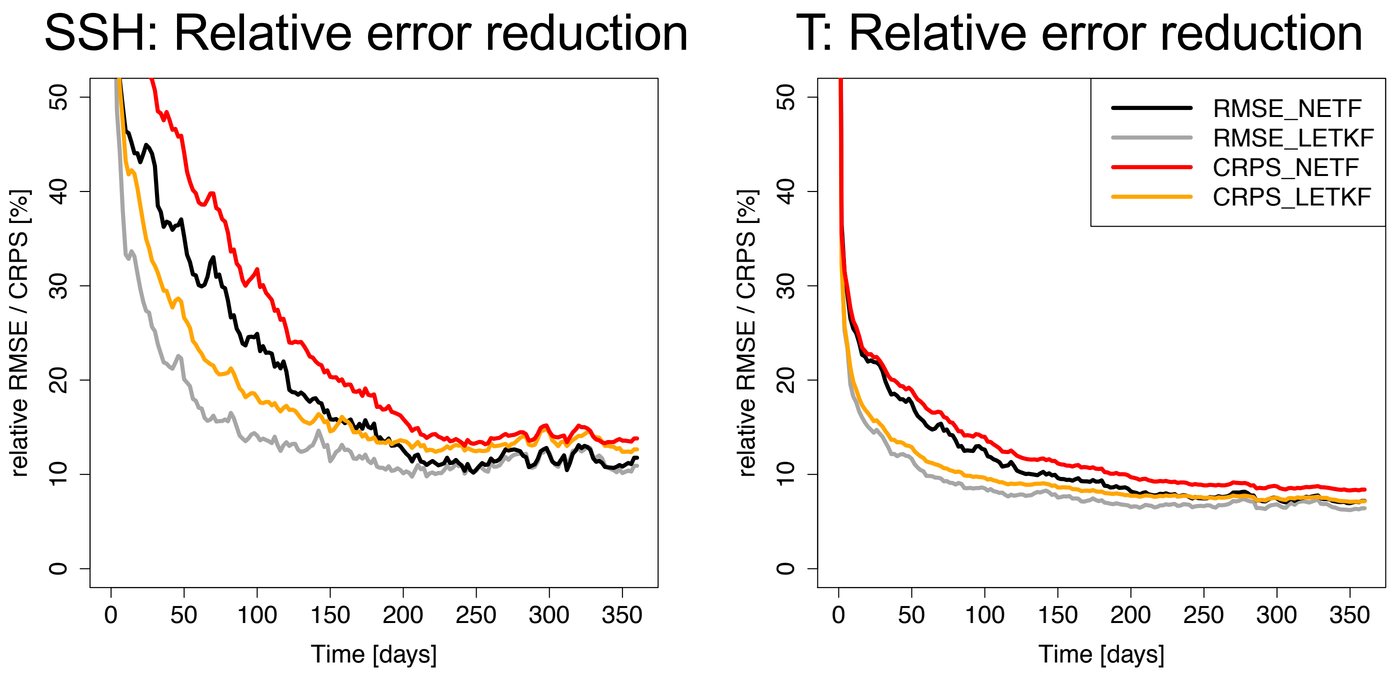






Filter performances in NEMO

- RMS errors reduced to 10% (velocities to 20%) of initial error
- Slower convergence for NETF, but to same error level as LETKF
- CRPS (Continuous Rank Probability Score) shows similar behavior



Tödter, Kirchgessner, Nerger & Ahrens, MWR 144 (2016) 409 – 427

Hybrid LETKF-NETF

Motivation

NETF

- can perform better than LETKF with nonlinear model
- needs rather large ensemble

LEKTF

- larger parameter region with convergence
- very stable

Hybrid filter

- Can we combine the strengths of LETKF and NETF?

Hybrid variants

1-step update (*HSync*)

$$\mathbf{X}^a_{HSync} = \overline{\mathbf{X}}^f + (1-\gamma)\Delta\mathbf{X}_{NETF} + \gamma\Delta\mathbf{X}_{ETKF}$$

- $\Delta\mathbf{X}$ is assimilation increment of a filter
- γ is hybrid weight (between 0 and 1; 1 for fully LETKF)

2-step updates

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

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

- Both steps computed with increased **R** according to γ

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

Choosing hybrid weight γ

- Hybrid weight shifts filter behavior
- How to choose it?

Some possibilities:

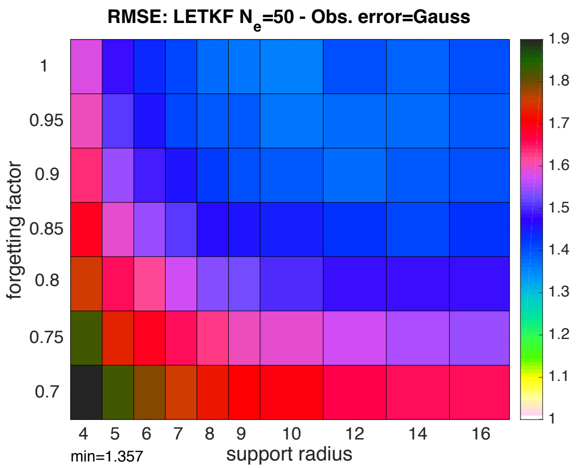
- Fixed value
- Adaptive
  - According to which condition?
  - For hybrid particle-EnKF, Frei & Kuensch (2013) suggested using effective sample size $N_{eff} = \sum_i 1/(w^i)^2$
    - Choose γ so that N_{eff} is as small as possible but above minimum limit
  - Alternative used here

$$\gamma_{adap} = 1 - N_{eff}/N_e$$

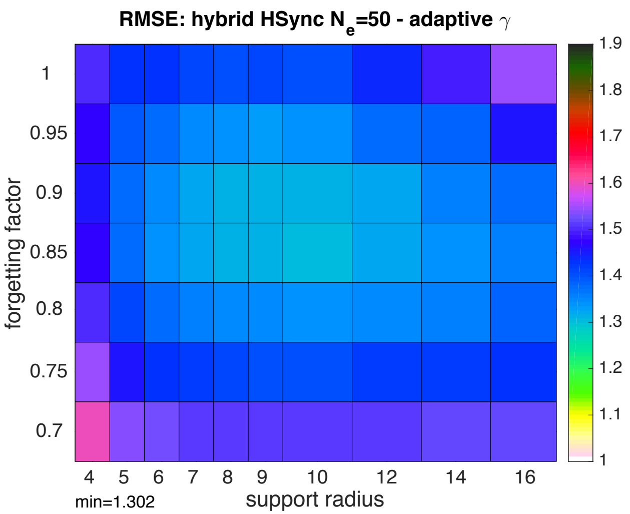
(close to 1 if N_{eff} small)

Test with Lorenz-96 model (n=80 as before)

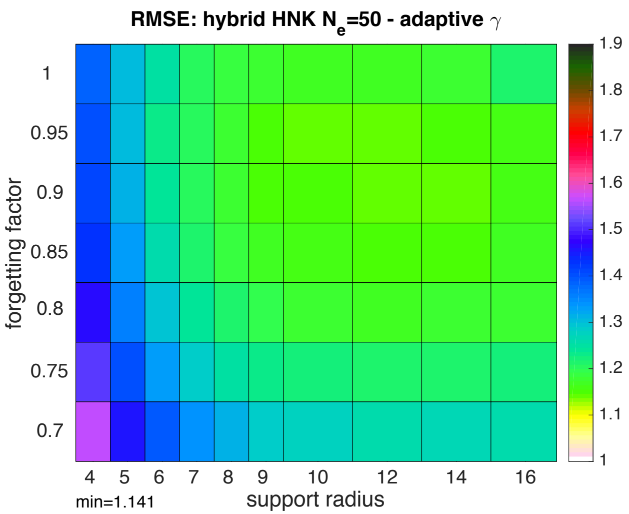
Ensemble size N=50



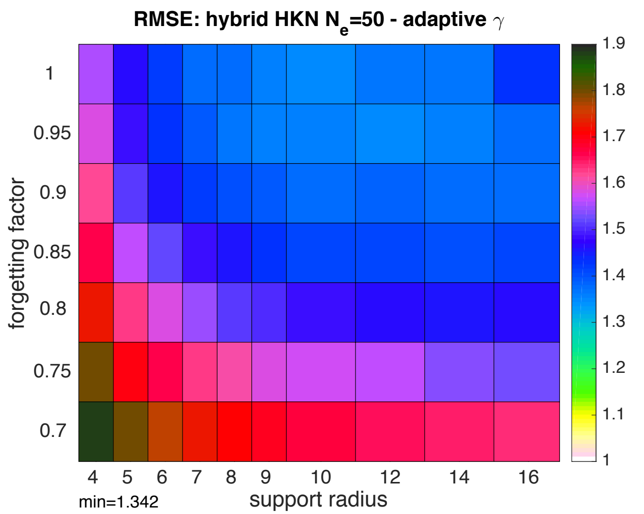
- All hybrid variants improve estimates compared to LETKF & NETF
- Similar stability as LETKF
- Dependence on forgetting factor & localization radius like LETKF
- Similar optimal localization radius
- Largest improvement for variant HNK (NETF before LETKF)



4% improvement



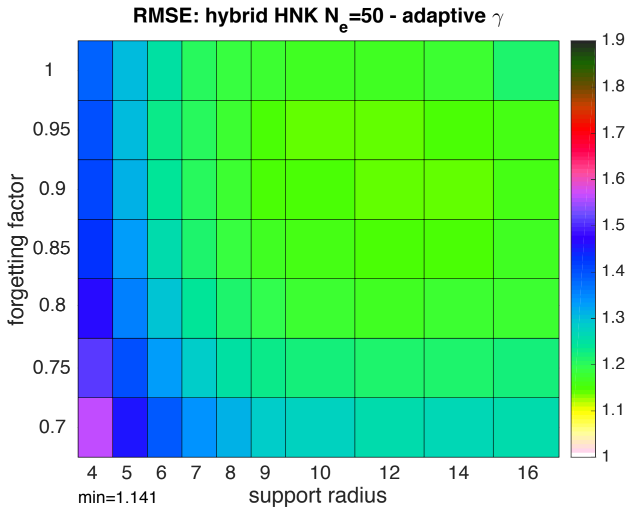
16% improvement



1% improvement

Hybrid Kalman-Nonlinear Ensemble Transform Filter

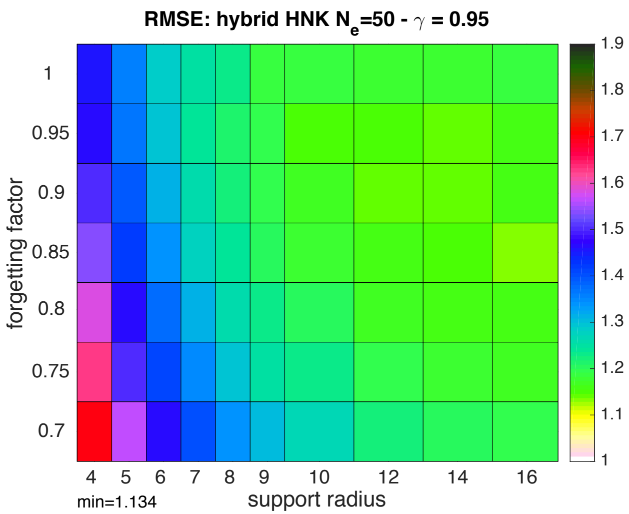
N=50 – adaptive and fixed hybrid weight γ

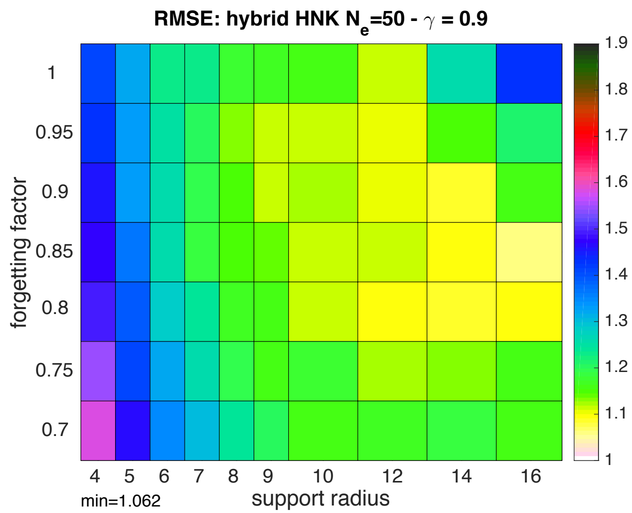


16% improvement

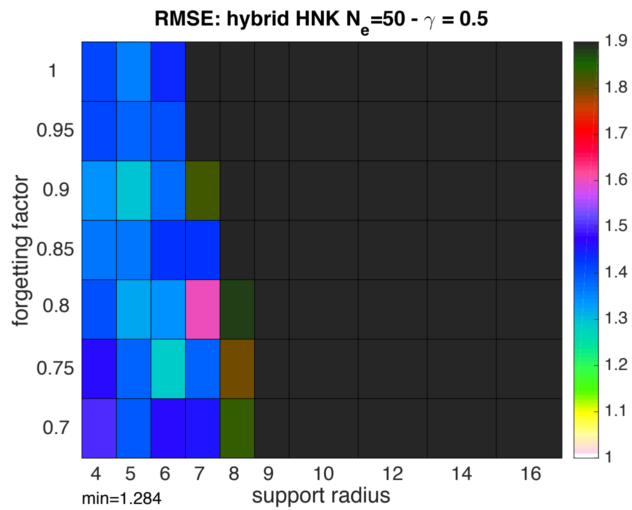
Consider only version HNK

- Fixed γ also successful, smaller errors than hybrid
- Has to be close to 1.0 (small NETF fraction)
- Smaller γ reduced stability



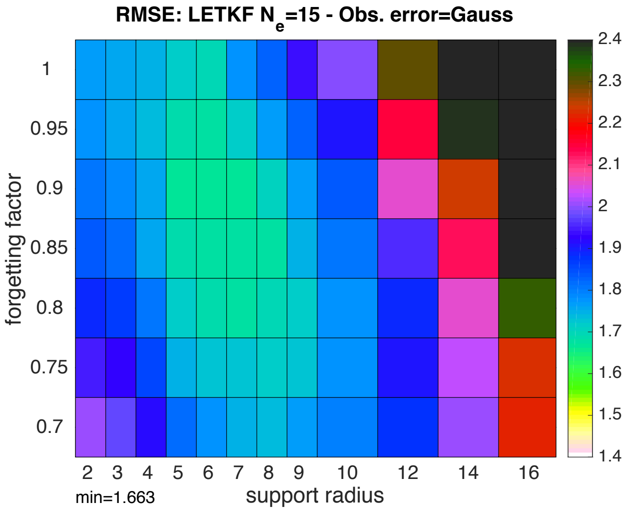


22% improvement

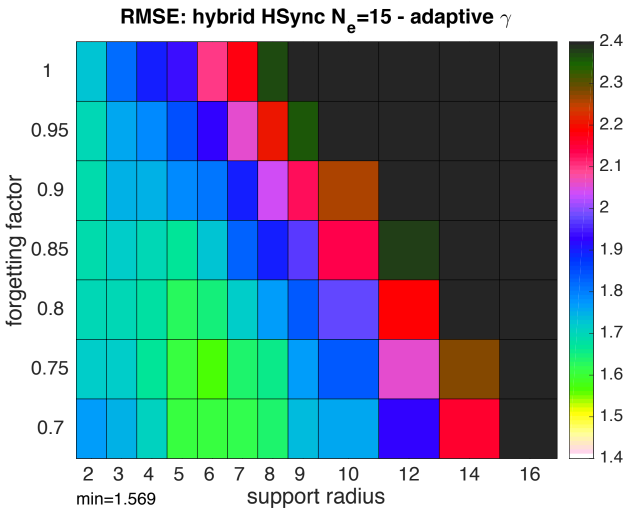


5% improvement

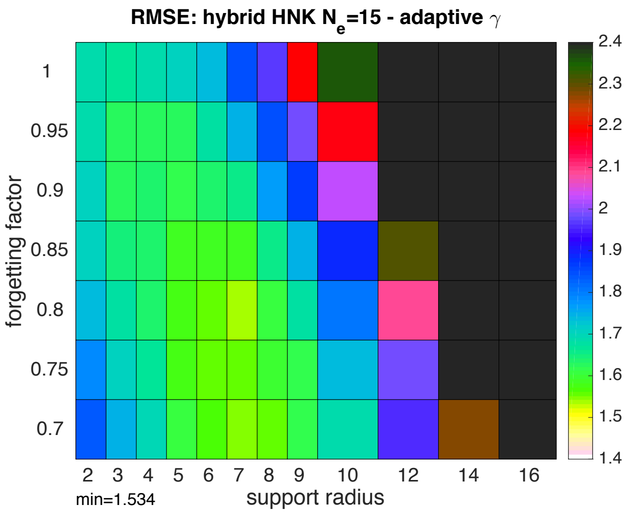
Small Ensemble N=15



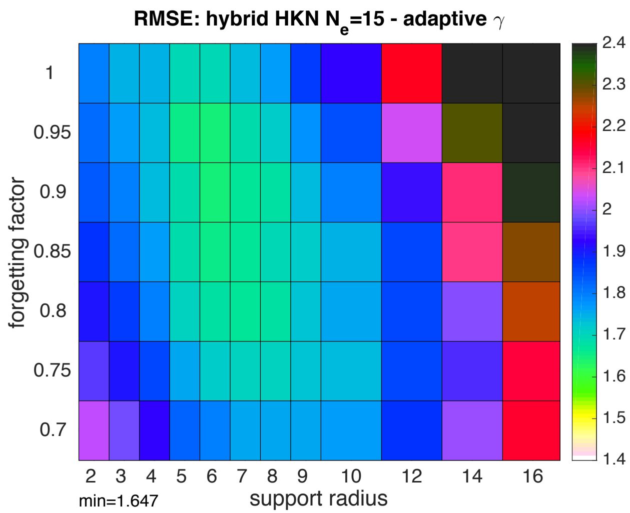
- Hybrid still positive influence
- Smaller improvement than for N=50
- Optimal parameters for HSync & HNK different from HKN
- HSync and HNK more similar



6% improvement

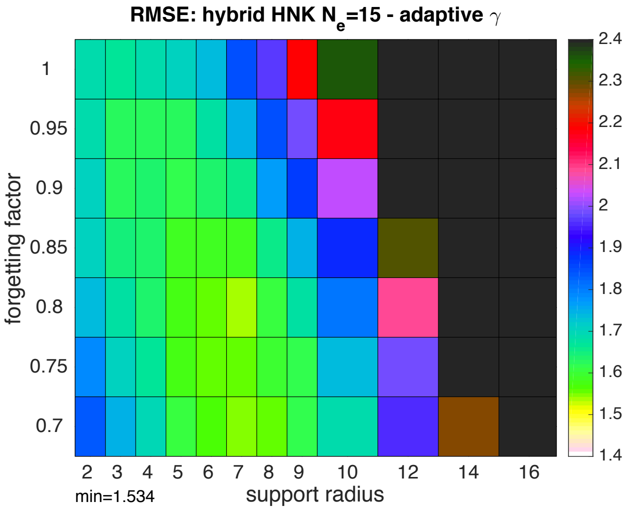


8% improvement



1% improvement

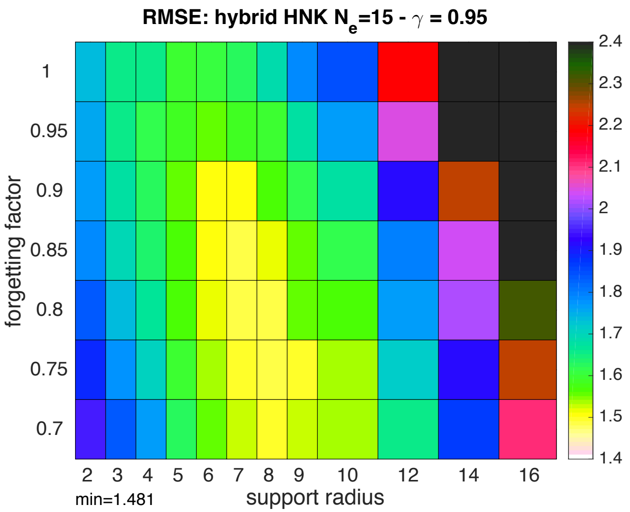
Small Ensemble N=15



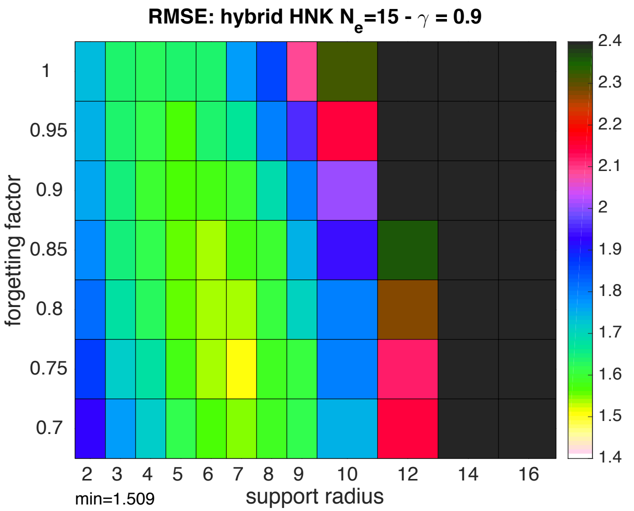
8% improvement

Fixed γ

- reduces error compared to adaptive γ
- Can increase stability region
- Needs to be even closer to 1 than for N=50



11% improvement



9% improvement

Summary

- Nonlinear ensemble transform filter (NETF)
  - Update state estimate as particle filter
    Transform ensemble using covariance matrix
- Hybrid LETKF-NETF
  - Combine analysis updates controlled by hybrid weight
  - Smaller errors than LETKF and NETF
  - Variant NETF-before-LETKF yield best results
  - Fixed hybrid height showed lower errors compared to simple adaptive weight
  - Next steps
    - reconsider adaptive weight
    - assess with more realistic model

Thank you!