Deutscher Wetterdienst
Wetter und Klima aus einer Hand

DWD Global Data Assimilation System (GME & ICON)

–

3D-Var & VarEnKF

Andreas Rhodin

DWD-HERZ Winterschool on Data Assimilation

2012-02-17

Models and Assimilation Systems at DWD

Present :

GME	global	30 km	3D-Var–PSAS
COSMO-EU	regional	7 km	Nudging
COSMO-DE	regional	2.8 km	Nudging, Latent Heat Nudging

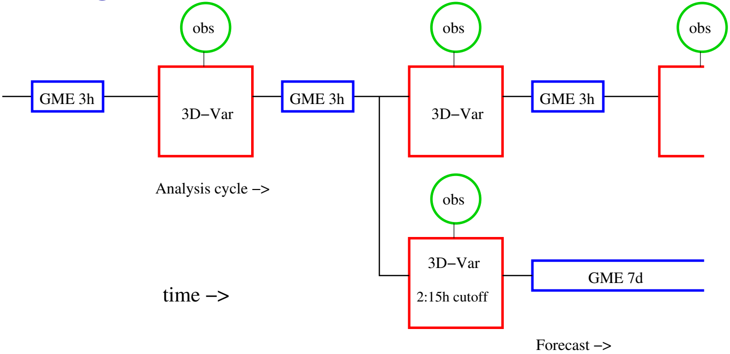
Future :

ICON	global, regional refinements		Hybrid 3D-Var/EnKF
COSMO-DE	regional	convective scale	LETKF

Course of this talk

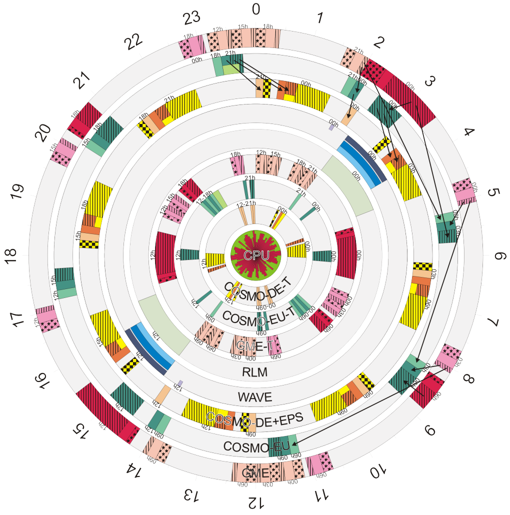
- Current operational system (3D-Var)
  - Operational setup
  - Algorithm (PSAS), inner & outer loop
  - Observation operators, tangent linear & adjoint
- Experimental LETKF
  - Preliminary results
- Localisation for remote sensing observations
- Plans for the hybrid VarENKF
  - Algorithm (additional control variables)
  - Advantages

Operational global data assimilation



- Cycled 3 hour GME forecast / 3D-Var analysis
- Long term (7 day) forecast at 00 and 12 UT
  - seperate (Hauptlauf) analysis with 2h15 cutoff
- In addition (not shown)
  - seperate (Hauptlauf) analysis with 2h15 cutoff
  - additional forecasts (6,18 UT) for COSMO EU boundary conditions
  - Sea Surface Analysis, Snow analysis
  - Soil moisture analysis

Operational timetable of the DWD model suite with dataflow



GME, COSMO: Analysis / Nudging
GME Analysis: serial part
GME, COSMO: Forecast
COSMO-DE-EPS: Interpolation
WAVE (GSM, LSM, MSM)
COSMO-EU: Surface moisture analysis
Main run
Pre-Assimilation
Assimilation

00..23	real time [UTC]
00h, 03h, ..	model time [UTC]
T	Testsuite

Operations / Pre-operations / Experiments

On sxw (vector computer) / lxw (Linux Cluster) – (Offenbach) :

- Operations
- Pre-operational suite (currently GME 20km)

On sxe / lxe – (currently in Ludwigshafen) :

- Experiments
  - NUMEX (numerical experimentation system)
    mimics dependencies in routine setup

3D-Var – PSAS Algorithm

Minimize cost function:

$$J = \frac{1}{2}(\mathbf{x} - \mathbf{x}_b)^T \mathbf{B}^{-1}(\mathbf{x} - \mathbf{x}_b) + \frac{1}{2}(\mathbf{y} - \mathcal{H}(\mathbf{x}_b))^T \mathbf{R}^{-1}(\mathbf{y} - \mathcal{H}(\mathbf{x}_b))$$

For the linear case ($\mathcal{H}(\mathbf{x}) \rightarrow \mathbf{Hx}$) we can derive the following linear equation for the analysis $\mathbf{x}_a$ which minimizes J:

$$\mathbf{x}_a - \mathbf{x}_b = \mathbf{BH}^T(\mathbf{HBH}^T + \mathbf{R})^{-1}(\mathbf{y} - \mathcal{H}(\mathbf{x}_b))$$

Size of $\mathbf{B}$, $\mathbf{x}$: $n_x \approx 10^8$ (number of model variables)
Size of $\mathbf{R}$, $\mathbf{y}$: $n_y \approx 10^6$ (number of observations)

How to solve this equation on the Computer ?

We use an iterative Conjugate Gradient algorithm.

Then we need not represent the matrices $\mathbf{B}, \mathbf{R}, \mathbf{H}, \mathbf{H}^T$
We merely need routines that calculate the respective matrix products

3D-Var PSAS - Observation Operator

$$\mathbf{x}_a - \mathbf{x}_b = \mathbf{B}\mathbf{H}^T(\mathbf{H}\mathbf{B}\mathbf{H}^T + \mathbf{R})^{-1}(\mathbf{y} - \mathcal{H}(\mathbf{x}_b))$$

Observation operator $\mathcal{H}(\mathbf{x})$

Calculates the model equivalent to the observations from the model state

$\mathcal{H}(\mathbf{x}) = \mathcal{H}_i(\mathcal{H}_o(\mathbf{x}))$
$\mathcal{H}_i$: Interpolation to the location to the observation
$\mathcal{H}_o$: Observation operator (may be complex: RTTOV, occultations)

Realised by respective subroutines

3D-Var PSAS - Linearised Observation Operator

$$\mathbf{x}_a - \mathbf{x}_b = \mathbf{B}\mathbf{H}^T(\mathbf{H}\mathbf{B}\mathbf{H}^T + \mathbf{R})^{-1}(\mathbf{y} - \mathcal{H}(\mathbf{x}_b))$$

H linearised observation operator

defined by the Jakobian of $\mathcal{H}$

$$\mathbf{H} = \frac{d}{d\mathbf{x}}\mathcal{H}(\mathbf{x})$$

i.e. we have to differentiate $\mathcal{H}$ with respect to $\mathbf{x}$

If the have a computer code that calculates $\mathcal{H}(\mathbf{x})$ we can obtain a code that calculates $\mathbf{H}\,\mathbf{x}$ by application of the chain rule (line by line) to that code (automatic differentiation).

$\mathcal{H}_3(\mathcal{H}_2(\mathcal{H}_1(\mathbf{x}))) = \mathbf{H}_3\mathbf{H}_2\mathbf{H}_1\mathbf{x}$

The linearised observation operator $\mathbf{H}_o$ for radiance assimilation is included in the RTTOV package.

3D-Var PSAS - Adjoint Observation Operator

$$\mathbf{x}_a - \mathbf{x}_b = \mathbf{B}\mathbf{H}^T(\mathbf{H}\mathbf{B}\mathbf{H}^T + \mathbf{R})^{-1}(\mathbf{y} - \mathcal{H}(\mathbf{x}_b))$$

For the L_2 norm used here the adjoint is given by the the transposed matrix.

We have:
$\mathbf{H}^T = (\mathbf{H}_3\mathbf{H}_2\mathbf{H}_1)^T = \mathbf{H}_1^T\mathbf{H}_2^T\mathbf{H}_3^T$

Thus we can apply the chain rule in reversed order line by line to the code that calculates $\mathcal{H}(\mathbf{x})$ (automatic differentiation).

The adjoint observation operator $\mathbf{H}_o^T$ for radiance assimilation is included in the RTTOV package as well.

3D-Var PSAS - B-Matrix

$$\mathbf{x}_a - \mathbf{x}_b = \mathbf{B}\mathbf{H}^T(\mathbf{H}\mathbf{B}\mathbf{H}^T + \mathbf{R})^{-1}(\mathbf{y} - \mathcal{H}(\mathbf{x}_b))$$

The background error covariance matrix is a large dense matrix in model space.

As shown in Harald Anlaufs talk we represent it by a sparse approximation in wavelet transformed space $\mathbf{B}_w$:

$$\mathbf{B} = \mathbf{W}\mathbf{B}_w\mathbf{W}^T$$

In fact we use a sparse square root representation: $\mathbf{B}_w = \mathbf{B}_w^{1/2}\,\mathbf{B}_w^{1/2\,T}$

$\mathbf{W}, \mathbf{W}^T$ are realised by the respective wavelet transformation routines.

$\mathbf{B}_w$ is not defined on the model grid but on a lat-lon grid.
Thus in $\mathbf{H}\mathbf{B}\mathbf{H}^T$ and $\mathbf{B}\mathbf{H}^T$ the operators $\mathbf{H}, \mathbf{H}^T$ include interpolation operators to the respective grid.

3D-Var PSAS - R-Matrix

$$\mathbf{x}_a - \mathbf{x}_b = \mathbf{B}\mathbf{H}^T(\mathbf{H}\mathbf{B}\mathbf{H}^T + \mathbf{R})^{-1}(\mathbf{y} - \mathcal{H}(\mathbf{x}_b))$$

In general **R** is diagonal or at least sparse.

Thus **R** can be represented explicitly.

3D-Var PSAS Conjugate Gradient Algorithm

$$\mathbf{x}_a - \mathbf{x}_b = \mathbf{B}\mathbf{H}^T(\mathbf{H}\mathbf{B}\mathbf{H}^T + \mathbf{R})^{-1}(\mathbf{y} - \mathcal{H}(\mathbf{x}_b))$$

We solve the linear set of equations for $\mathbf{z}$
by a preconditioned CG algorithm in observation space:

$$(\mathbf{H}\mathbf{B}\mathbf{H}^T + \mathbf{R})\,\mathbf{z} = \mathbf{y} - \mathcal{H}(\mathbf{x}_b)$$

That requires of order 15 to 25 iterations.

Finally the postmultiplication step to model space is required:

$$\mathbf{x}_a - \mathbf{x}_b = \mathbf{B}\mathbf{H}^T\,\mathbf{z}$$

3D-Var PSAS – Outer Loop

The CG algorithm solves the linear problem:

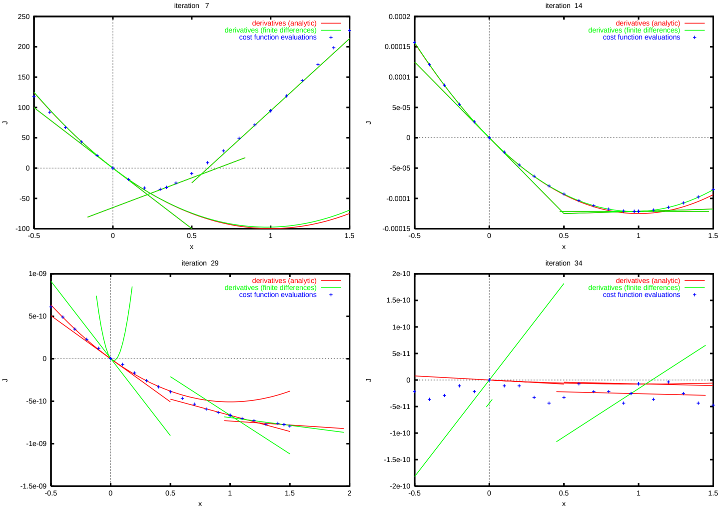
$$(\mathbf{HBH}^T + \mathbf{R})\,\mathbf{z} = \mathbf{y} - \mathcal{H}(\mathbf{x}_b)$$

In order to solve the full nonlinear problem we have to iterate in an outer loop:

1) Solve the linear system for an estimate $\mathbf{z}_i$
2) Re-linearise at the new estimate:
   a) recalculate the right hand side
   b) linearise $\mathcal{H}$ to obtain $\mathbf{H}$
   c) replace $\mathbf{R}$ by the inverse Hessian of J_o in case of VQC.
3) Proceed with 1).

After ≈ 10 outer loops our convergence criterium is met.
In order to ensure convergence we perform a line serch after each CG step.

3D-Var PSAS Algorithm – Line Search Monitoring



Remarks on the Convergence

For conventional data we can find the minimum of the cost function exactly.

For more complex operators minimisation does fail earlier if $\mathbf{H}$ is not sufficiently accurate.

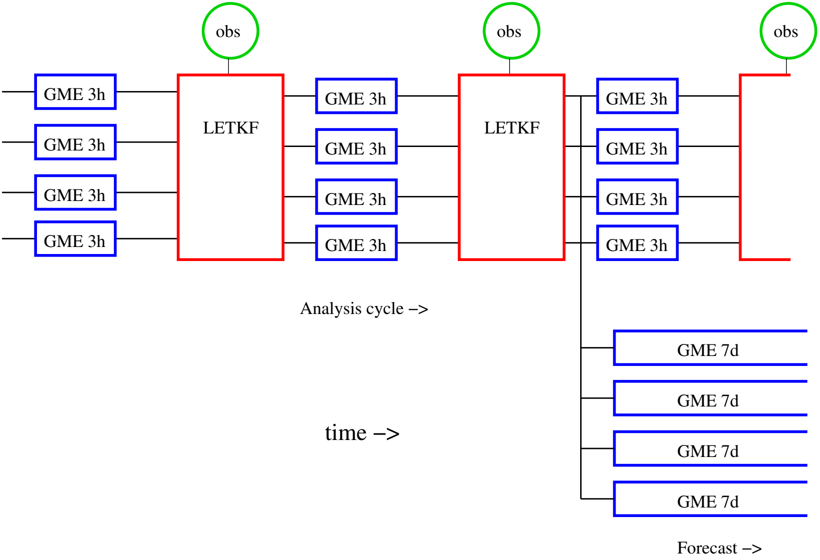
For practical porposes minimisation is stopped then the accuracy of the solution is small compared to the specified background and observational errors.

Global LETKF for GME - Formulation

LETKF following Hunt & al

- Apply the square root formulation of EnKF at every model gridpoint
- Use Observations in the vicinity of the gridpoint, with $\mathbf{R}^{-1}$ scaled in dependence on the distance using the Gaspari & Cohn function.
- Use this localisation in the vertical and horizontal.

Global LETKF for GME - Setup



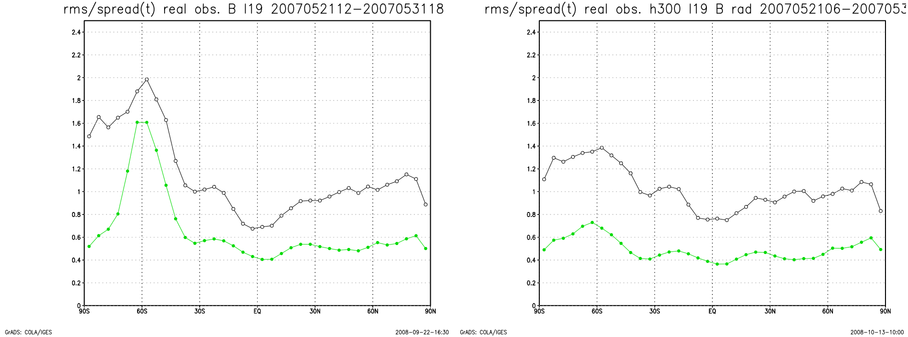
GME LETKF experiments

- Setup:
  - GME (ni=64), 6 hourly cycle, 32 ensemble members, 15 day period.
- Parameters changed:
  - Model error
    - additive model error (random noise generated by 3D-Var-B)
    - multiplicative inflation
  - Observations
    - conventional data only
    - conventional data + AMSU-A
    - artifitial data (nature run + gaussian noise on observations)
  - Localisation length scale
    - 200, 300, 500 km
- Results:
  - Additive model error with 300 km localisation length scale works best.
  - Strong positive impact of AMSU-A in SH

Global LETKF Experiments:

32 members, ni=64, 10 days of 6 h cycling

Spatial distribution of spread and rmse
Dependence on data density

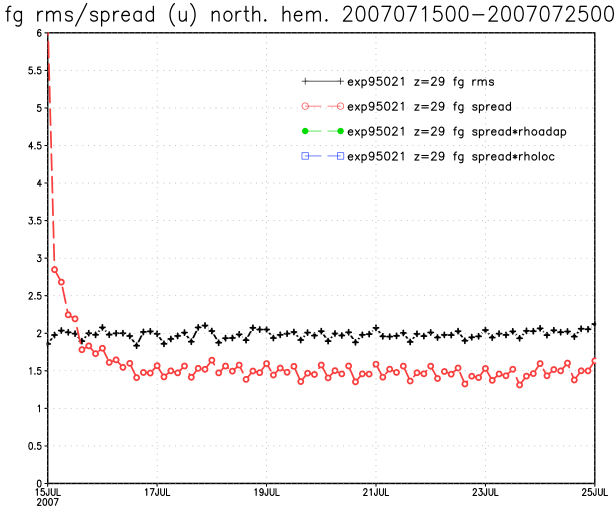


Meridional distribution of rmse (o) and spread (o).

500 hPa first guess temperature.

Without (left) and with (right) assimilation of AMSU-A.

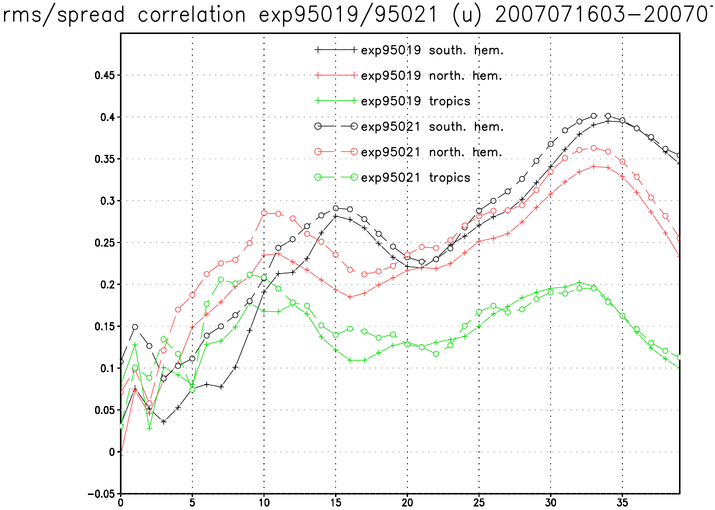
Global LETKF: Temporal evolution of spread and rmse



850 hPa zonal wind (u)

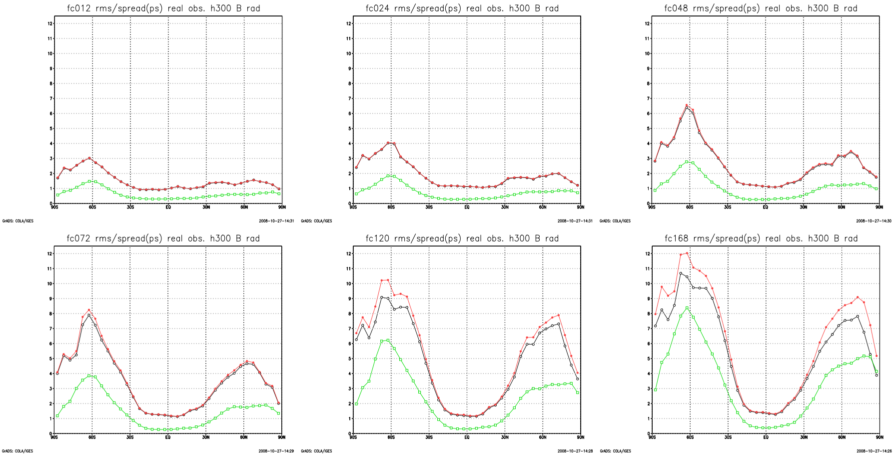
Temporal evolution of first guess ensemble spread (+) and rmse (o)

Status: Global LETKF



rmse/spread correlation as a function of height (SH, TR, NH)

Global LETKF: Forecast uncertainty



RMSE of deterministic forecast

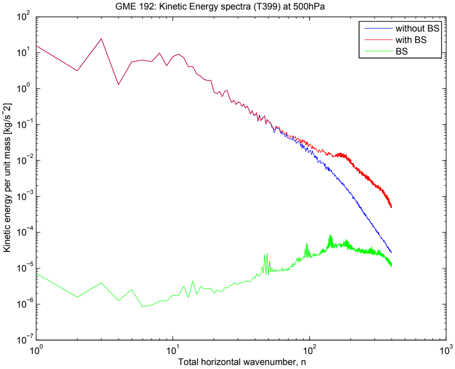
RMSE of ensemble mean

Ensemble spread

for 12, 24, 48, 72, 120, 168h forecasts of surface pressure (hPa).

Model error: Backscatter algorithm

As an alternative for the 3D-Var-B additive model error formulation the backscatter algorithm was explored by Jason Ambadan (cf. Poster)



Localisation for remote sensing observations

Localisation on B

- The ensebmble covariance matrix is rank deficient.
  Compare: number of degrees of freedom vs. ensemble size.
- Small entries of the empirical correlation matrix **S** are uncertain:
  $\sigma_{ij}^2 \equiv \mathrm{E}\left\{(S_{ij} - B_{ij})^2\right\} = \frac{1}{N-1}\left[B_{ii}B_{jj} + (B_{ij})^2\right]$
- Remedy: force small matrix elements to zero,
  i.e. Multiply ensemble covariance matrix **B** element by element
  (Schur product $\circ$) with a localisation matrix **C**.
  Choice of **C**:
  - **C** positive definite (so that $\mathbf{C} \circ \mathbf{B}$ is a valid covariance matrix).
  - $\mathbf{C}_{ii} = 1$, $\mathbf{C}_{ij} = 0$ for large distances.

  These requirements are fullfilled for the pieacewise rational functions proposed by Gaspary + Cohn.
- Choices for implementation:
  - Explicit Schur product:
    too expensive
  - Variational formulation:
    (uses operator implementation of $\mathbf{C}^{1/2} \cdot \mathbf{x}$)

Localisation in observation space

- *In situ observations*:
  - The Kalman gain equations use matrices $\mathbf{BH}^T$ and $\mathbf{HBH}^T$.
  - $\mathbf{H}(\mathbf{C} \circ \mathbf{B})\mathbf{H}^T$ is equivalent to $\mathbf{C} \circ (\mathbf{HBH}^T)$;
    $(\mathbf{C} \circ \mathbf{B})\mathbf{H}^T$ is equivalent to $\mathbf{C} \circ (\mathbf{BH}^T)$.
  - Fast (parallel) implementations exist.
    (size of observation space $\ll$ model space).
- *Remote sensing observations*:
  Localisation on $\mathbf{B}$ is different from localisation in observational space.
  Implementation:
  - assign a nominal position to the remote sensing observation.
    (in order to specify C_{ij})

  Choices:
  - Apply: $\mathbf{C} \circ (\mathbf{HBH}^T)$.
  - Apply: $\mathbf{C} \circ (\mathbf{R}^{-1})$ (Hunt et al.)

  Drawback
  - $\mathbf{H}(\mathbf{C} \circ \mathbf{B})\mathbf{H}^T$ is **not** equivalent to $\mathbf{C} \circ (\mathbf{HBH}^T)$.
  - $\mathbf{C} \circ (\mathbf{HBH}^T)$ is sub-optimal.
  - same applies to Hunt et al. algorithm

Localisation on R (Hunt et al.)

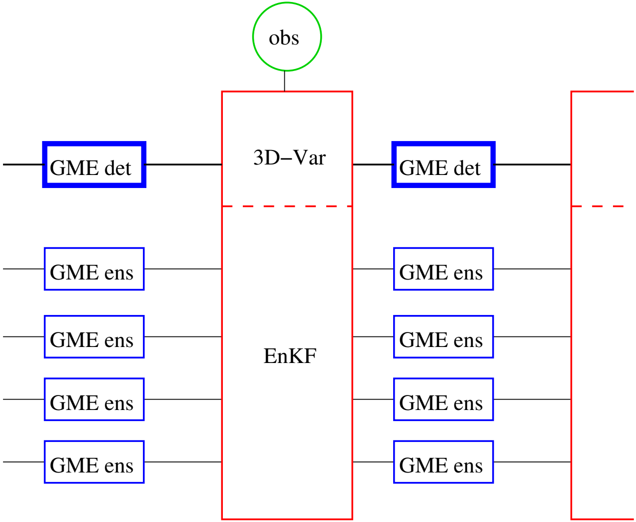
- LETKF algorithm proposed by Hunt et al.
  - Perform local analyses at each model gridpoint
  - Use observations only within a prescribed localisation distance
  - Weight of observations continuously approaches zero at bounds of localisation volume.
    (i.e. weight $\mathbf{R}^{-1}$)
- Advantage
  - fast
  - no constraints on weight function
    (varying localisation length scale)
- Disadvantage
  - inconsistent and suboptimal approach for remote sensing data

Plan for a Variational EnKF

Plan for a Variational EnKF

- Integrate the EnKF and the 3D-Var in a hybrid system
- Use the information from a lower dimensional EnKF for the update of the higher dimensional deterministic forecast.
- Method: use an operator implementation of the square root of the localisation matrix.
  Corresponds to the additional control variable approach (Mark Buehner)

Variational EnKF for GME/ICON



time →

← analysis cycle interval →

VarEnKF: Formulation

3D-Var update:

$$\mathbf{x}_a - \mathbf{x}_b = \mathbf{B}\mathbf{H}^T(\mathbf{H}\mathbf{B}\mathbf{H}^T + \mathbf{R})^{-1}(\mathbf{y} - \mathcal{H}(\mathbf{x}_b))$$

Replace **B** by an operator implementation of the localized ensemble B-matrix:
$\mathbf{B} \rightarrow \mathbf{C} \circ (\mathbf{W}\mathbf{W}^T)$

We need an operator implementation of the square root of the localisation matrix **C**:
$\mathbf{C} = \mathbf{L}\mathbf{L}^T$

Then (cf talk by Tijana):
$\mathbf{C} \circ \mathbf{W}\mathbf{W}^T = (\mathbf{W}\mathbf{L})(\mathbf{W}\mathbf{L})^T$

We can even replace the original **B** by a weighted sum:
$\mathbf{B} \rightarrow \alpha\mathbf{B}_{3DVar} + \beta\mathbf{C} \circ (\mathbf{W}\mathbf{W}^T)$

VarEnKF: Advantages

- Advantages
  - Smooth transition between 3D-Var and EnKF
  - Deterministic analysis using Ensemble B (and 3D-Var B)
  - Variational Quality Control applicable
  - Varational bias correction applicable
  - Consistent handling of remote sensing observations (localisation on **B**)