Stability of Ensemble Kalman Filters

Idrissa S. Amour, Zubeda Mussa, Alexander Bibov, Antti Solonen, John Bardsley†, Heikki Haario and Tuomo Kauranne

Lappeenranta University of Technology
† University of Montana

High-Performance Computing in Meteorology, ECMWF
October 27, 2014

Overview

Introduction

- Data assimilation calls for algorithms that often approximate the Extended Kalman Filter, or EKF, that itself is too heavy to run
- It is essential, but quite non-trivial, that the approximate Kalman filters used remain stable over the assimilation period.

Introduction

- Stability of a filter is related to the numerical stability of the corresponding algorithm, but numerical stability alone does not guarantee filter stability.
- It is also mandatory that the filter does not diverge from the true state of the system.
- Yet all filters applied to nonlinear models will diverge if there are no observations.

Introduction

- We study the general conditions for filter stability applicable to variational methods, and approximate Kalman filters many kinds ;
- Present several ways to stabilize filters;

Introduction

- Provide empirical results with a shallow-water model that illustrate a relation between ensemble spread and temporal and spatial density of observations that
- Generalizes the well-known Courant-Friedrichs-Lewy numerical stability condition to filter stability in a Hilbert space setting; and
- Explain the impact of model bias on filter stability in this context.

Overview

The Extended Kalman Filter (EKF)

Algorithm

Iterate in time

$$\begin{aligned}
\mathbf{x}^f(t_i) &= M(t_i, t_{i-1})(\mathbf{x}^a(t_{i-1})) \\
\mathbf{P}_i^f &= \mathbf{M}_i \mathbf{P}^a(t_{i-1}) \mathbf{M}_i^{\mathrm{T}} + \mathbf{Q} \\
\mathbf{K}_i &= \mathbf{P}^f(t_i) \mathbf{H}_i^{\mathrm{T}} (\mathbf{H}_i \mathbf{P}^f(t_i) \mathbf{H}_i^{\mathrm{T}} + \mathbf{R})^{-1} \\
\mathbf{x}^a(t_i) &= \mathbf{x}^f(t_i) + \mathbf{K}_i(\mathbf{y}_i^o - H(\mathbf{x}^f(t_i))) \\
\mathbf{P}^a(t_i) &= \mathbf{P}^f(t_i) - \mathbf{K}_i \mathbf{H}_i \mathbf{P}^f(t_i)
\end{aligned}$$

The Extended Kalman Filter (EKF)

Where

- $\mathbf{x}^f(t_i)$ *is the prediction at time* t_i
- $\mathbf{x}^a(t_i)$ *is the analysis at time* t_i
- $\mathbf{P}^f(t_i)$ *is the prediction error covariance matrix at time* t_i
- $\mathbf{P}^a(t_i)$ *is the analysis error covariance matrix at time* t_i
- $\mathbf{Q}$ *is the model error covariance matrix*
- $\mathbf{K}_i$ *is the Kalman gain matrix at time* t_i
- $\mathbf{R}$ *is the observation error covariance matrix*
- H *is the nonlinear observation operator*
- $\mathbf{H}_i$ *is the linearized observation operator at time* t_i
- $\mathbf{M}_i$ *is the linearized weather model at time* t_i

The Extended Kalman Filter (EKF)

Properties

- *The model is not assumed to be perfect*
- *Model integrations are carried out forward in time with the nonlinear model for the state estimate and*
- *Forward and backward in time with the tangent linear model and the adjoint model, respectively, for updating the prediction error covariance matrix*
- *There is no minimization, just matrix products and inversions*
- *Computational cost of EKF is prohibitive, because* $\mathbf{P}^f(t_i)$ *and* $\mathbf{P}^a(t_i)$ *are huge full matrices*

Ensemble Kalman Filters (EnKF)

Properties

- *Ensemble Kalman Filters are generally simpler to program than variational assimilation methods or EKF, because*
- *EnKF codes are based on just the non-linear model and do not require tangent linear or adjoint codes, but they*
- *Tend to suffer from slow convergence and therefore inaccurate analyses because ensemble size is small compared to model dimension*
- *Often underestimate analysis error covariance*

Ensemble Kalman Filters (EnKF)

Properties

- *Ensemble Kalman filters often base analysis error covariance on* **bred vectors**, *i.e. the difference between ensemble members and the background, or the ensemble mean*
- *One family of EnKF methods is based on perturbed observations, while*
- *Another family uses explicit linear transforms to build up the ensemble*

EnKF Cost functions

Algorithm

Minimize

$$(\mathbf{P}^f(t_i))^{-1} = (\beta \mathbf{B}_0 + (1-\beta)\frac{1}{N}\mathbf{X}^f(t_i)\mathbf{X}^f(t_i)^{\mathrm{T}})^{-1}$$

Algorithm

Minimize

$$\begin{aligned}
&\ell(\mathbf{x}^a(t_i)|\mathbf{y}_i^o) \\
&= (\mathbf{y}_i^o - H(\mathbf{x}^a(t_i)))^{\mathrm{T}}\mathbf{R}^{-1}(\mathbf{y}_i^o - H(\mathbf{x}^a(t_i))) \\
&+ \frac{1}{N}\sum^{N}(\mathbf{x}_i^f(t_i) - \mathbf{x}^a(t_i))^{\mathrm{T}}(\mathbf{P}^f(t_i))^{-1}(\mathbf{x}_i^f(t_i) - \mathbf{x}^a(t_i))
\end{aligned}$$

Overview

The Variational Ensemble Kalman Filter (VEnKF)

Algorithm

Iterate in time

Step 0: *Select a state* $\mathbf{x}^a(t_0)$ *and a covariance* $\mathbf{P}^a(t_0)$ *and set* $i = 1$

Step 1: *Evolve the state and the prior covariance estimate:*
(i) Compute $\mathbf{x}^f(t_i) = M(t_i, t_{i-1})(\mathbf{x}^a(t_{i-1}))$;
(ii) Compute the ensemble forecast $\mathbf{X}^f(t_i) = M(t_i, t_{i-1})(\mathbf{X}^a(t_{i-1}))$;
(iii) **Minimize** *from a random initial guess* $(\mathbf{P}^f(t_i))^{-1} = (\beta\mathbf{B}_0 + (1-\beta)\frac{1}{N}\mathbf{X}^f(t_i)\mathbf{X}^f(t_i)^{\mathrm{T}} + \mathbf{Q}_i)^{-1}$
by the LBFGS method;

The Variational Ensemble Kalman Filter (VEnKF)

Algorithm

Step 2: *Compute the Variational Ensemble Kalman Filter posterior state and covariance estimates:*

(i) **Minimize**

$$\ell(\mathbf{x}^a(t_i)|\mathbf{y}_i^o) = (\mathbf{y}_i^o - H(\mathbf{x}^a(t_i)))^{\mathrm{T}}\mathbf{R}^{-1}(\mathbf{y}_i^o - H(\mathbf{x}^a(t_i))) + (\mathbf{x}^f(t_i) - \mathbf{x}^a(t_i))^{\mathrm{T}}(\mathbf{P}^f(t_i))^{-1}(\mathbf{x}^f(t_i) - \mathbf{x}^a(t_i))$$

by the LBFGS method;

(ii) Store the result of the minimization as $\mathbf{x}^a(t_i)$*;*
(iii) Store the limited memory approximation to $\mathbf{P}^a(t_i)$*;*
(iv) Generate a new ensemble $\mathbf{X}^a(t_i) \sim N(\mathbf{x}^a(t_i), \mathbf{P}^a(t_i))$*;*

The Variational Ensemble Kalman Filter (VEnKF)

Properties

- *Follows the algorithmic structure of VKF, separating the time evolution from observation processing.*
- *A new ensemble is generated every observation step*
- *Bred vectors are centered on the* **mode**, *not the mean, of the ensemble, as in Bayesian estimation*
- *Like in VKF, a new ensemble and a new error covariance matrix is generated at every observation time*
- *No covariance leakage*
- *No tangent linear or adjoint code*
- *Asymptotically equivalent to VKF and therefore EKF when*

Overview

Kalman filtering with a biased model

- Having a biased model means that the model produces a forecast error with non-zero mean. In this case, our model equations:

$$\mathbf{x}(t_i) = M(t_i, t_{i-1})(\mathbf{x}(t_{i-1})) + \eta(t_i)$$

- entail that the expectation of model error is non-zero, but generally unknown, and can locally at t_i be approximated by a linear error.

Kalman filtering with a biased model

- This reads as

$$E(\eta(t_i)) = \mathbf{b}(t_i - t_{i-1}) \neq 0$$

- If the bias **b** is known, there are various ways to compensate for it, such as the ones presented by Dee (2005) and Trémolet (2005).

Kalman filters on combined state and observation space

- Let us denote the time interval between observations by Δt.
- To second order accuracy in Δt, we can derive a two term form for model evolution, when looking at it over short observation intervals Δt.
- This form separates the smoothly evolving model bias from stochastic Gaussian model noise.

Kalman filters on combined state and observation space

$$\mathbf{x}^0(t_i + \Delta t) = M(t_i, t_{i-1})(\mathbf{x}(t_i)) + \mathbf{b}\Delta t + \boldsymbol{\eta}(t_{i-1}) + \mathcal{O}(\Delta t^2)$$

- where $\mathbf{x}^0$ is the true future state and $\boldsymbol{\eta}(t_i)$ is Gaussian model noise with zero mean.

Kalman filters on combined state and observation space

- In this error decomposition, the smooth model bias term $\mathbf{b}\Delta t$ represents *drift*, see e.g. Orrell (2005), Orrell et al. (2001).
- It indicates a tendency of unknown direction.
- But the maximum speed $||\mathbf{b}||$ of the expected state of an imperfect model to drift away from the true evolution of the state of the system can be estimated from statistics of forecast systematic errors.

Kalman filters on combined state and observation space

- Let us recall the *innovation form* of the Kalman filter that we have used.

$$
\begin{aligned}
\mathbf{P}^f(t_i) &= \mathbf{M}_{i-1}\mathbf{P}^a(t_{i-1})\mathbf{M}_{i-1}^{\mathrm{T}} + \mathbf{Q}_i \\
\mathbf{K}_i &= \mathbf{P}^f(t_i)\mathbf{H}_i^{\mathrm{T}}(\mathbf{H}_t\mathbf{P}^f(t_i)\mathbf{H}_i^{\mathrm{T}} + \mathbf{R}_i)^{-1}. \\
\mathbf{x}^a(t_i) &= \mathbf{x}^f(t_i) + \mathbf{K}_i(\mathbf{y}_i^o - H_i(\mathbf{x}^f(t_i))).
\end{aligned}
\tag{1}
$$

- From the last equation, we see that the *state increment* $\delta\mathbf{x}(t_i)$ satisfies

$$\delta\mathbf{x}(t_i) = \mathbf{x}^a(t_i) - \mathbf{x}^f(t_i) \tag{2}$$

- and is therefore computed from the *innovation vector* $\mathbf{d}_i$

$$\mathbf{d}_i = \mathbf{y}_i^o - H_i(\mathbf{x}^f(t_i)) \tag{3}$$

- by solving a linear equation with the inverse of the Kalman gain matrix $\mathbf{K}_i$:

$$\mathbf{K}_i^{-1}\delta\mathbf{x}(t_i) = \mathbf{d}_i \tag{4}$$

Kalman filters on combined state and observation space

- Replacing the Kalman gain here with its definition in the second equation in the Kalman group above, we see that the two last equations are equivalent to the following system of two equations:

$$
\begin{aligned}
(\mathbf{H}_i\mathbf{P}^f(t_i)\mathbf{H}_i^{\mathrm{T}} + \mathbf{R}_i)\delta\mathbf{z}_i &= \mathbf{d}_i \\
\delta\mathbf{x}(t_i) &= \mathbf{P}^f(t_i)\mathbf{H}_i^{\mathrm{T}}\delta\mathbf{z}_i
\end{aligned}
\tag{5}
$$

- where $\delta\mathbf{z}_i$ is the *information vector*.

Kalman filters on combined state and observation space

- Inserting the first and third equations from our Kalman group of equations into the first equation above, we get a linear operator equation, local in time, whose operator we shall denote by $\mathbf{A}_i$.
- This operator can aptly be called the *symmetric Kalman filter operator* and it defines the *information form* of the Kalman filter.

$$\mathbf{A}_i \delta \mathbf{z}_i = (\mathbf{H}_i \mathbf{M}_{i-1} \mathbf{P}^a(t_{i-1}) \mathbf{M}_{i-1}^{\mathrm{T}} \mathbf{H}_i^{\mathrm{T}} + \mathbf{H}_i \mathbf{Q}_i \mathbf{H}_i^{\mathrm{T}} + \mathbf{R}_i) \delta \mathbf{z}_i = \mathbf{d}_i \tag{6}$$

Kalman filters on combined state and observation space

- The operator $\mathbf{A}_i$ above that defines the Kalman filter is applied to measurements sampled from a stochastic process, but it is itself a deterministic linear operator.
- $\mathbf{A}_i$ defines the metric in the quadratic form, in which the Kalman filter produces a least squares estimate that can also be interpreted as the maximum a posteriori estimate according to the Bayes theorem.

Kalman filters on combined state and observation space

- The equation (6) is an equation defined on the space of observations.
- Its various component operators are defined on different spaces as well:
- $\mathbf{P}^a(t_{i-1})$ is defined on the state space at time t_{i-1},
- $\mathbf{Q}_i$ is defined on the state space at time t_i and
- $\mathbf{R}_i$ is defined on the observation space at time t_i.

Stabilizing Kalman filters

- Because the symmetric Kalman filter operator (6) is defined on the observation space at time t_i, it imposes its implicit optimality condition as a final time observation space control.
- For small enough analysis increments $\delta\mathbf{x}(t_i)$, nonlinear Kalman filtering for smoothly evolving dynamical systems will be locally stable, if one of the following conditions is fulfilled:

Stabilizing Kalman filters

1. *An explicit or implicit static prior or background term with a positive weight is used, or*
2. *The state space is completely observable and the observation operator is a projection operator. These conditions imply that the spectrum of the error covariance operator* $\mathbf{P}^a(t_i)$ *is both bounded and positively bounded from below, or*

Stabilizing Kalman filters

1. *The model is* **not** *perfect on any observable subspace of the model state space, which implies that the spectrum of the error covariance operator* $\mathbf{Q}_i$ *is positively bounded from below, or*
2. *All observations are noisy and there are no rigid constraints between errors in different observations, which implies that the spectrum of the error covariance operator* $\mathbf{R}_i$ *is positively bounded from below*

Stabilizing Kalman filters

- The list above is formulated in terms of a small enough analysis increment $\delta\mathbf{x}(t_i)$, rather than a small time between observations Δt.
- It can be seen that the latter is a special case of the former. The condition of smallness of analysis increments covers small perturbations in any direction in the state space, and not just the ones parameterized by the time variable.

Stabilizing Kalman filters

- In both cases, the validity of the above statements depends on the smoothness assumption of the model evolution on the state space, so that the model evolution operator converges to identity as the magnitude of a perturbation decreases to zero.

Stabilizing Kalman filters

Theorem

Stability property of VKF under Gaussian model noise*. The Variational Kalman Filter algorithm for smoothly evolving dynamical systems is stable under Gaussian model noise, if any of the conditions in the list above is fulfilled, and the iterations employed in the VKF algorithm are carried out until convergence to a limit set by the lower bound* ϵ *on the spectrum of the symmetric Kalman Filter operator* $\mathbf{A}_i$*. This will take a finite number of steps independent of model resolution.*

A CFL like condition for stability

- Let us now look at the impact of model bias on the stability analysis above. We have denoted a local model bias vector in unit time by $\mathbf{b}$.
- It will thus represent the direction and speed of model drift $\mathbf{b}\Delta t$.
- We shall assume that the dynamics of both the true operator $\mathcal{M}$ and the biased discrete model M are smooth.

A CFL like condition for stability

- This assumption implies that the accumulation of bias in the model state will be proportional to the duration of model evolution to second order accuracy, in the form

$$\begin{aligned} &\mathbf{x}(t+\Delta t) - \mathbf{x}^0(t+\Delta t) = \\ &M(t+\Delta t, t)(\mathbf{x}(t)) - \mathcal{M}(t+\Delta t, t)(\mathbf{x}(t)) = \\ &\mathbf{b}\Delta t + \eta(t) + \mathcal{O}(\Delta t^2) \end{aligned} \tag{7}$$

A CFL like condition for stability

- where $\mathbf{x}^0(t)$ denotes the model state evolved with the unbiased true model $\mathcal{M}$ and
- where we have continued to assume that model noise $\boldsymbol{\eta}(t)$ is Gaussian and Markov.
- Because of the smoothness of model evolution, for small enough Δt, the drift $\mathbf{b}\Delta t$ will stay beneath ϵ, no matter what is the direction of the bias $\mathbf{b}$.

A CFL like condition for stability

- But as we have seen, any innovation direction that is modified by the local Kalman operator with a factor less than ϵ away from the identity will be suppressed by the static prior plus the noise term in the Variational Kalman Filter.

A CFL like condition for stability

- We can state this result as a conditional stability property of the Variational Kalman Filter against model bias

Theorem

Conditional stability under bias of VKF. *The Variational Kalman Filter is stable for smoothly evolving dynamical systems, if there is a sufficient temporal and spatial density of observations available, with an observation bias uncorrelated with the model bias, and such that the span of the temporally local model bias is observed.*

A CFL like condition for stability

- The conditional stability property above can be seen as a kind of Courant-Friedrichs-Lewy (CFL) stability condition, only in *state space* and not in the computational domain, for Kalman filtering algorithms.
- It means that if the model bias drives model evolution away from the true trajectory, the bias will not accumulate beyond a given threshold, if the corresponding drift can be countered fast enough with an observation with an error that is uncorrelated with model bias, before the drift has increased beyond the assumed model noise level.

A CFL like condition for stability

- If the drift has grown too large, VKF (and EKF) may choose to believe the biased forecast, rather than the contradicting observation.
- In the terminology or Orrell et al. (2001), the stability under bias property above gives a sufficient condition for the Kalman Filter to guarantee that the sequence of analyses produced by VKF will continue to *shadow* the truth with a bound that corresponds to the level of model error covariance $||\mathbf{P}^a(t_i)||$, or $||\mathbf{B}_0 + \mathbf{P}^f(t_i) + \mathbf{Q}_i||$ in VKF notation.

A CFL like condition for stability

- Shadowing cannot similarly be guaranteed for the strong constraint 4D-Var without a background term, because the absence of a model error term and the strictness of the model constraint prevent the strong constraint 4D-Var operator from being a Fredholm operator.
- There will always be directions in the state space that are not observable.

A CFL like condition for stability

- The background term does stabilize the filter, but if the background has been produced by the same biased model, the ensuing analysis will be *bias-blind*, in the terminology of Dee (2005), and continue to suffer from the same bias.
- This was also empirically observed by Orrell (2005). Strong-constraint 4D-Var is therefore not stable against model bias.

A CFL like condition for stability

- The CFL condition for a numerical model of the advection equation reads

Theorem

$$\Delta t \leq \Delta x / ||\mathbf{v}|| \tag{8}$$

- where Δx is the shortest spatial grid length and $||\mathbf{v}||$ is the fastest advection velocity in the system.

A CFL like condition for stability

- The corresponding expression for the *shadowing condition* above reads

Theorem

$$\Delta t \leq \epsilon / ||\mathbf{b}|| \tag{9}$$

- where ϵ is the amplitude of Gaussian noise used in the Kalman filter and $||\mathbf{b}||$ the speed of growth of forecast bias
- Verbally, the above formula says that we must have correcting observations in the direction of the bias $\mathbf{b}$ before the corresponding drift has become larger than the noise level of model error.

A CFL like condition for stability

- In practice, the assumptions of the stability under model bias property may not be fulfilled and models will exhibit bias, especially in poorly observed areas of their state space, such as the stratosphere.
- This can be countered with covariance inflation.

A CFL like condition for stability

- The shadowing condition with covariance inflation becomes

Theorem

$$\Delta t \leq ||\mathbf{B}_0 + \mathbf{Q}_i|| / ||\mathbf{b}|| \qquad (10)$$

- or, more generally,

Theorem

$$||\mathbf{B}_0 + \mathbf{Q}_i|| \geq ||\mathbf{b}|| ||\delta \mathbf{x}(t_i)|| \qquad (11)$$

- for any state increment $\delta \mathbf{x}(t_i)$.

Overview

The Shallow Water Model

- **MOD_FreeSurf2D** by Martin and Gorelick
- Finite-volume, semi-implicit, semi-Lagrangian MATLAB code
- Used to simulate a physical laboratory model of a Dam Break experiment along a 400 m river reach in Idaho
- The model consists of a system of coupled partial differential equations

The Shallow Water Model - 1

Shallow Water Equations

$$\frac{\partial U}{\partial t} + U\frac{\partial U}{\partial x} + V\frac{\partial U}{\partial y} = -g\frac{\partial \eta}{\partial x} + \epsilon\left(\frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2}\right) + \frac{\gamma_T(U_a - U)}{H}$$
$$-g\frac{\sqrt{U^2+V^2}}{Cz^2}U + fV,$$
$$\frac{\partial V}{\partial t} + U\frac{\partial V}{\partial x} + V\frac{\partial V}{\partial y} = -g\frac{\partial \eta}{\partial y} + \epsilon\left(\frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial y^2}\right) + \frac{\gamma_T(V_a - V)}{H}$$
$$-g\frac{\sqrt{U^2+V^2}}{Cz^2}V - fU,$$
$$\frac{\partial \eta}{\partial t} + \frac{\partial HU}{\partial x} + \frac{\partial HV}{\partial y} = 0$$

The Shallow Water Model - 2

Where

- *U is the depth-averaged x-direction velocity*
- *V is the depth-averaged y-direction velocity*
- *η is the free surface elevation*
- *g is the gravitational constant*
- *ϵ is the horizontal eddy viscosity coefficient*
- *γ_T is the wind stress coefficient*
- *U_a and V_a are the reference wind components for top boundary friction*
- *H is the total water depth*
- *C_z is the Chezy coefficient for bottom friction*

The Dam Break laboratory experiment

Where

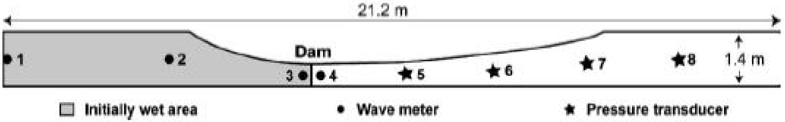
- *The 400 m long river stretch has been scaled down to 21.2 m*
- *Water depth is 0.20 m above the dam*
- *The dam is placed at the most narrow point of the river*
- *The riverbed downstream from the dam is initially dry*
- *In the experiment the dam is broken instantly and a flood wave sweeps downstream*
- *The total duration of the laboratory experiment is 130 seconds*

The observations

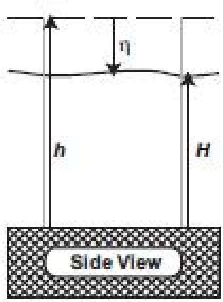
Where

- *The flow is measured with eight wave meters for water depth, placed irregularly at the approximate flume mid-line up and downstream from the dam*
- *Wave meters report the depth of water at 1 Hz, so with 1 s time intervals*
- *Computational time step is 0.103 s*

Flume geometry and wave meters



Vertical profile of flume

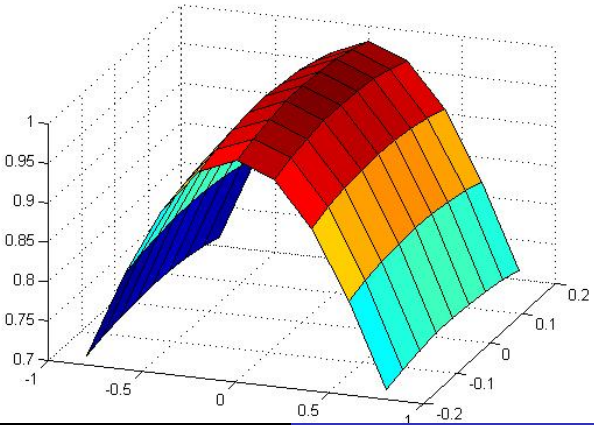


VEnKF applied to shallow-water equations

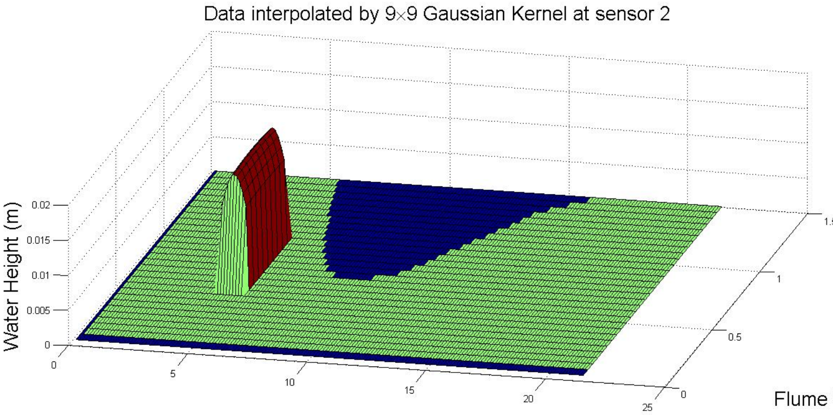
Where

- *Ensemble size 100*
- *Observations are interpolated in space and time*
- *A new ensemble is therefore generated every time step*

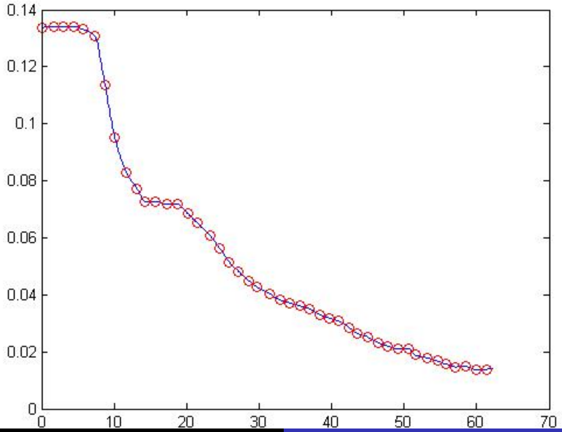
Interpolating kernel



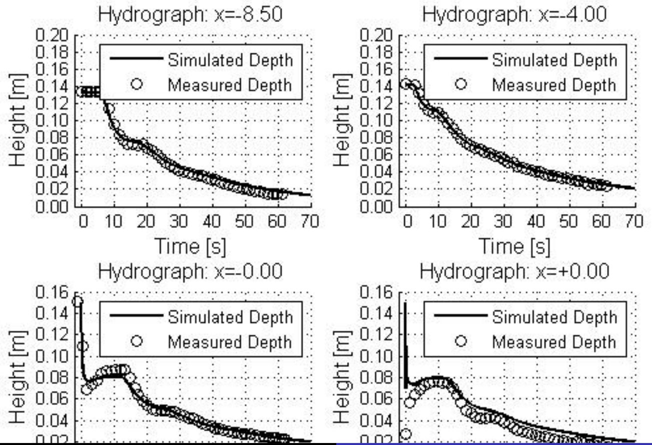
Observation interpolation in space



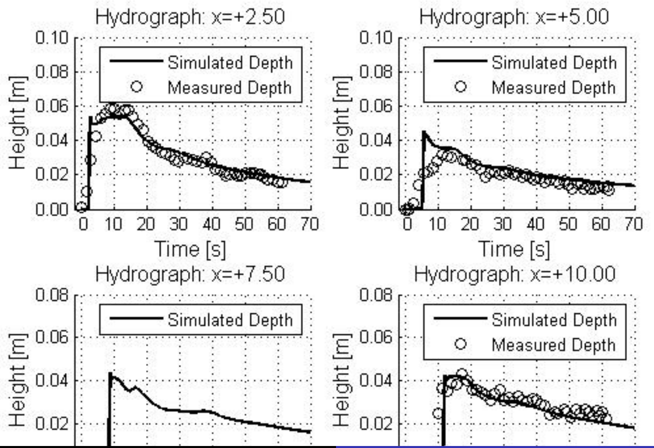
Observation interpolation in time



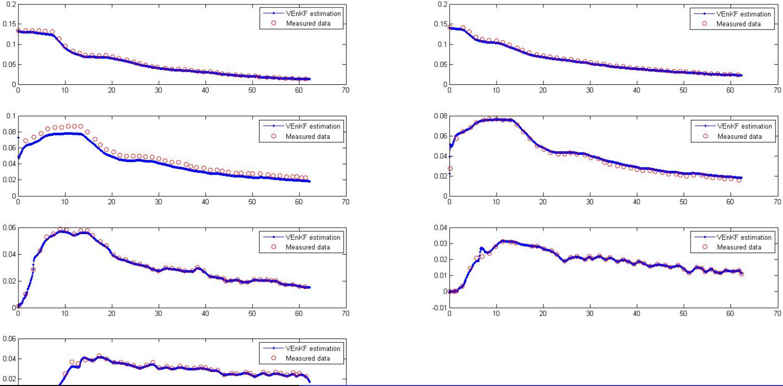
Model vs. hydrographs - 1



Model vs. hydrographs - 2



VEnKF vs. hydrographs

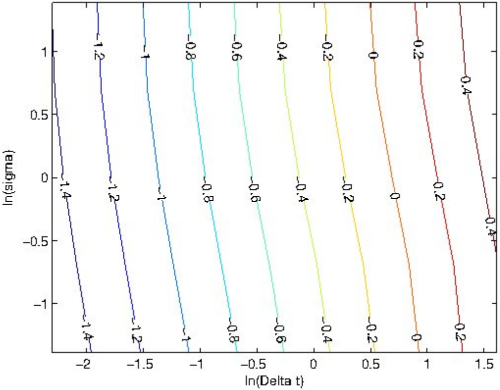


Overview

Evidence of the CFL condition in Hilbert space

- When observations are interpolated to appear on every time step, or less frequently
- The VEnKF algorithm always stays numerically stable, but
- With long time intervals between observations,
- Fails to capture waves present in the solution.
- Moreover, the empirical relationship between observation interval and filter divergence is linear

Ensemble spread vs. observation frequency



Overview

Conclusions - 1

- Kalman filters stabilize their estimates by a variety of means
- The most reliable way is to have abundant and frequent observations combined with
- Very short assimilation windows - even just one numerical time step

Conclusions - 2

- Generating a new ensemble every time step is optimal, because
- The more frequent the inter-linked updates of the ensemble and the error covariance estimate, the more accurate the analysis
- The local linearity assumption implicit in all Kalman filters remains more valid

Bibliography

- **Martin, M. and Gorelick, S. M.**: *MOD_FreeSurf2D: A MATLAB surface fluid flow model for rivers and streams*. Computers & Geosciences 31 (2005), pp. 929-946.
- **Auvinen, H., Bardsley J., Haario, H. and Kauranne, T.**: *The Variational Kalman Filter and an efficient implementation using limited memory BFGS*. Int. J. Numer. Meth. Fluids 64(3)2010, pp. 314-335(22).

Bibliography

- **Solonen, A., Haario, H., Hakkarainen, J., Auvinen, H., Amour, I. and Kauranne, T.**: *Variational ensemble Kalman filtering using limited memory BFGS*, Electronic Transactions on Numerical Analysis, 39(2012), pp-271-285(15).
- **Orrell, D.**: *The Future of Everything*, Thunder's Mouth Press, 2007.

Thank You

Thank You!