Variational Ensemble Kalman Filtering applied to shallow water equations

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

Lappeenranta University of Technology
[†] University of Montana

Ensemble Methods in Geophysical Sciences, Toulouse
November 13, 2012

Overview

3D Variational Assimilation (3D-Var)

Algorithm

Minimize

$$\begin{aligned} J(\mathbf{x}(t_i)) &= J_b + J_o \\ &= \frac{1}{2}(\mathbf{x}(t_i) - \mathbf{x}^b(t_i))^{\mathrm{T}}\mathbf{B}_0^{-1}(\mathbf{x}(t_i) - \mathbf{x}^b(t_i)) \\ &+ \frac{1}{2}(H(\mathbf{x}(t_i)) - \mathbf{y}_i^o)^{\mathrm{T}}\mathbf{R}^{-1}(H(\mathbf{x}(t_i)) - \mathbf{y}_i^o), \end{aligned}$$

3D Variational Assimilation (3D-Var)

Where

- $\mathbf{x}(t_i)$ *is the analysis at time* t_i
- $\mathbf{x}^{\mathbf{b}}(t_i)$ *is the background at time* t_i
- $\mathbf{y}_i^o$ *is the vector of observations at time* t_i
- $\mathbf{B}_0$ *is the background error covariance matrix*
- $\mathbf{R}$ *is the observation error covariance matrix*
- H *is the nonlinear observation operator*

3D Variational Assimilation (3D-Var)

Properties

- *3D-Var is computed at a snapshot in time where all observations are assumed contemporaneous*
- *3D-Var does not take into account atmospheric dynamics, by which*
- *It does not depend on the weather model*

4D Variational Assimilation (4D-Var)

Algorithm

Minimize

$$J(\mathbf{x}(t_0)) = J_b + J_o$$

$$= \frac{1}{2}(\mathbf{x}(t_0) - \mathbf{x}^b(t_0))^{\mathrm{T}}\mathbf{B}_0^{-1}(\mathbf{x}(t_0) - \mathbf{x}^b(t_0))$$

$$+ \frac{1}{2}\sum_{i=0}^{n}(H(M(t_i, t_0)(\mathbf{x}(t_0))) - \mathbf{y}_i^o)^{\mathrm{T}}\mathbf{R}^{-1}(H(M(t_i, t_0)(\mathbf{x}(t_0))) - \mathbf{y}_i^o)$$

4D Variational Assimilation (4D-Var)

Where

- $\mathbf{x}(t_0)$ *is the analysis at the beginning of the assimilation window*
- $\mathbf{x}^{\mathbf{b}}(t_0)$ *is the background at the beginning of the assimilation window*
- $\mathbf{B}_0$ *is the background error covariance matrix*
- $\mathbf{R}$ *is the observation error covariance matrix*
- H *is the nonlinear observation operator*
- M *is the nonlinear weather model*

4D Variational Assimilation (4D-Var)

Properties

- *The model is assumed to be perfect*
- *Model integrations are carried out forward in time with the nonlinear model and the tangent linear model, and backward in time with the corresponding adjoint model*
- *Minimization is sequential*
- *The weather model can run in parallel*

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*

The Variational Kalman Filter (VKF)

Algorithm

Iterate in time

Step 0: *Select an initial guess* $\mathbf{x}^a(t_0)$ *and a covariance* $\mathbf{P}^a(t_0)$, *and set* $i = 1$.

Step 1: *Compute the evolution model state estimate 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) **Minimize**

$$(\mathbf{P}^f(t_i))^{-1} = (\mathbf{M}_i\mathbf{P}^a(t_{i-1})\mathbf{M}_i^{\mathrm{T}} + \mathbf{Q})^{-1}$$

by the LBFGS method - or CG, as in incremental 4DVar;

The Variational Kalman Filter (VKF)

Algorithm

Step 2: *Compute the Variational Kalman filter state estimate and the posterior covariance estimate:*

(i) **Minimize**

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

by the LBFGS method - or CG, as in incremental 3DVar;

(ii) Store the result of the minimization as a VKF estimate $\mathbf{x}^a(t_i)$*;*

(iii) Store the limited memory approximation to $\mathbf{P}^a(t_i)$*;*

Step 3: *Update* $t := t + 1$ *and return to Step 1.*

The Variational Kalman Filter (VKF)

Where

- *Step 1(ii) is carried out with an auxiliary minimization that has a trivial solution but a random initial guess, and thereby generates a non-trivial minimization sequence*
- $\mathbf{P}^f(t_i)$ *and* $\mathbf{P}^a(t_i)$ *are kept in vector format, as a weighted sum of a diagonal or sparse background* $\mathbf{B}_0$*, a diagonal model error variance matrix* $\mathbf{Q}$ *and a low rank dynamical component* $\mathbf{P}^f(t_i)$ *that*
- *Is obtained from the Hessian update formula of the Limited Memory BFGS iteration*
- *The Kalman gain matrix is not needed*

The Variational Kalman Filter (VKF)

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 for updating the prediction error covariance matrix*
- *There are no matrix inversions, just matrix products and minimizations*
- *Computational cost of VKF is similar to 4D-Var*
- *Minimizations are sequantial*
- *Accuracy of analyses similar to EKF*

Overview

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

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

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 $\mathbf{X}^a(t_i) \sim N(\mathbf{x}^a(t_i), \mathbf{P}^a(t_i))$*;*

Step 3: *Update* $i := i + 1$ *and return to Step 1.*

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 ensemble size increases*

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*
- *f is the Coriolis parameter*

The Dam Break laboratory experiment

Where

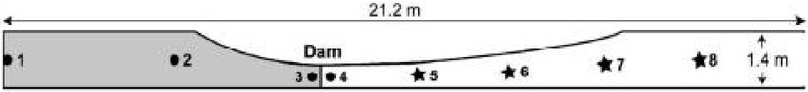
- *The 400 m long river stretch has been scaled down to 21.2 m*
- *Water depth is 0.20 m above the dam*
- *The dam is placed at the most narrow point of the river*
- *The riverbed downstream from the dam is initially dry*
- *In the experiment the dam is broken instantly and a flood wave sweeps downstream*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

The observations

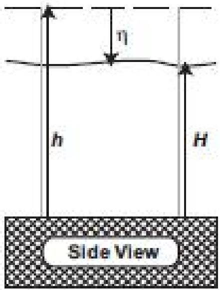
Where

- *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

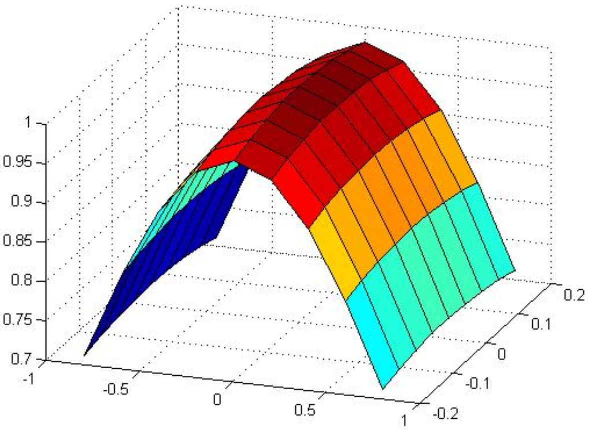


VEnKF applied to shallow-water equations

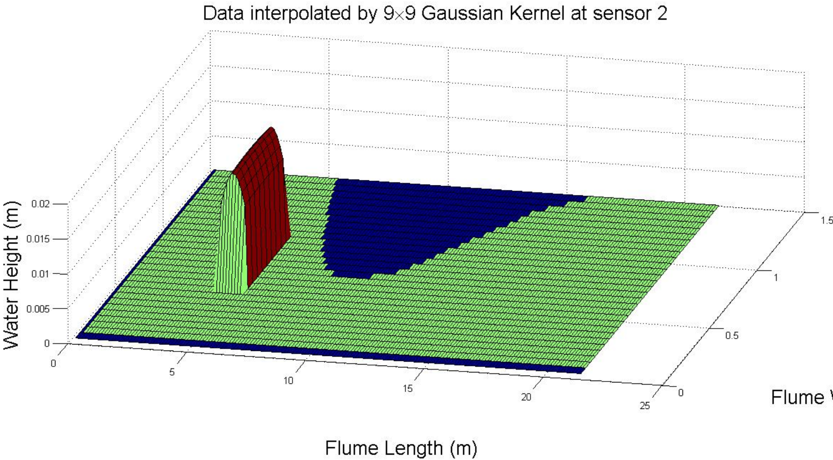
Where

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

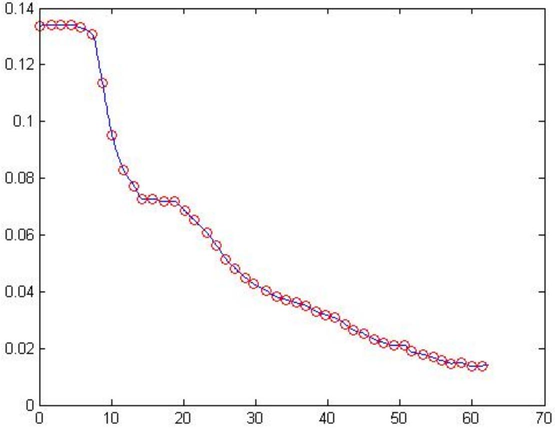
Interpolating kernel



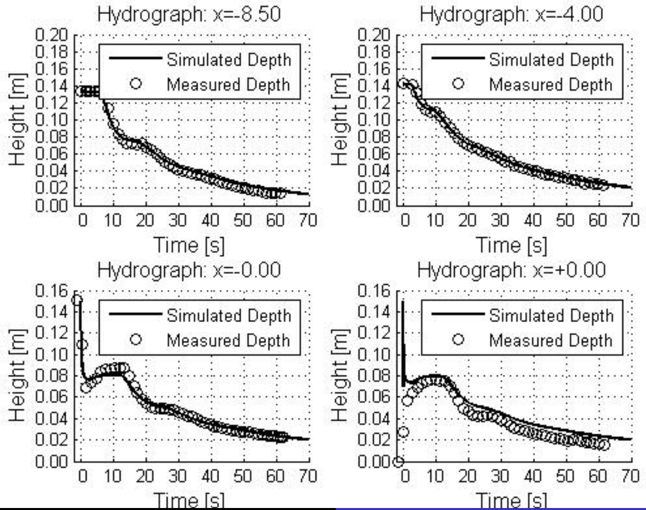
Observation interpolation in space



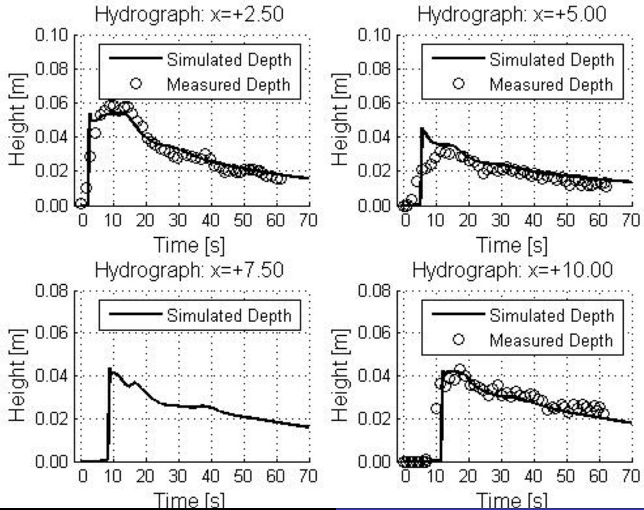
Observation interpolation in time



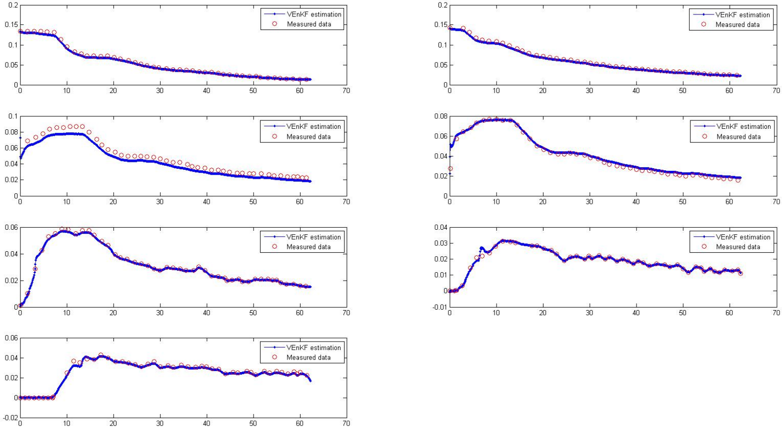
Model vs. hydrographs - 1



Model vs. hydrographs - 2



VEnKF vs. hydrographs



Overview

Conclusions - 1

- From earlier results with the Lorenz '95 model:
- VEnKF is asymptotically as good as EKF or VKF in forecast skill, but can be run without an adjoint code
- VEnKF attains equal quality to EKF only on large ensemble sizes, but
- VEnKF performs better than EnKF especially with small ensemble size
- VEnKF has proven to be able to compensate for model error in Shallow Water simulations

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
- There appears to be a trade-off between the accuracy of an assimilation method and its parallelism that needs to be decided by experiments

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).
- **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).

Thank You

Thank You!