Advanced data assimilation methods:
Extended Kalman Filter (EKF)
Ensemble Kalman Filter (EnKF)
And
Advanced Kalman Filter Techniques

Dave Kuhl, Hong Li and Eugenia Kalnay
University of Maryland

10:00-12:30 CSS 2428
November 12, 2008
Introduction

• Introduction
• Data Assimilation Review:
  – OI-Scaler
  – OI Multi-variate
• Kalman Filters
  – Extended Kalman Filter
  – Ensemble Kalman Filter
  – Advanced Ensemble Kalman Filters
• LETKF computer lab
Optimal weight to minimize the analysis error is:

\[ W = \frac{(\sigma^b)^2}{(\sigma^b)^2 + (\sigma^o)^2} \]
Sequential Assimilation of a Scalar (sect. 5.3.3)

Temperature

$T^a_i = T^b_i + W(T^o_i - T^b_i)$

$(\sigma^a_i)^2 = (1 - W)(\sigma^b_i)^2$

Include a model ($M$) to advance the time step:

$T^b_{i+1} = M[T^a_i]$
Multivariate Data Assimilation

\[ \mathbf{x}_i^a = \mathbf{x}_i^b + \mathbf{W}(\mathbf{y}_i^o - \mathbf{H}\mathbf{x}_i^b) \]

\[ \mathbf{x}_i^b = M[\mathbf{x}_i^a] \]

\( \mathbf{x}_i^a = \) analysis model state
\( \mathbf{x}_i^b = \) background model state
\( \mathbf{W} = \) weighting matrix
\( \mathbf{y}_i^o = \) observations
\( \mathbf{H} = \) observation operator (model->obs)
Multivariate OI Assimilation (sect. 5.4.1)

\[
\begin{align*}
\mathbf{x} &= \mathbf{x}^b + \mathbf{W} (\mathbf{y}^o - \mathbf{Hx}^b) \\
\mathbf{W} &= \mathbf{BH}^T [\mathbf{HBH}^T + \mathbf{R}]^{-1} \\
\mathbf{P}^a &= [\mathbf{I} - \mathbf{WH}] \mathbf{B} \\
\mathbf{x}_{i+1}^b &= \mathbf{M}[\mathbf{x}_i^a]
\end{align*}
\]

\(\mathbf{x}^a = \mathbf{x}^b + \mathbf{W}(\mathbf{y}^o - \mathbf{Hx}^b)\)
\(\mathbf{W} = \mathbf{BH}^T [\mathbf{HBH}^T + \mathbf{R}]^{-1}\)
\(\mathbf{P}^a = [\mathbf{I} - \mathbf{WH}] \mathbf{B}\)
\(\mathbf{x}_{i+1}^b = \mathbf{M}[\mathbf{x}_i^a]\)

\(\mathbf{P}^a = \mathbf{A} = \text{analysis error covariance Matrix}\)
\(\mathbf{P}^b = \mathbf{B} = \text{background error covariance Matrix}\)
\(\mathbf{P}^o = \mathbf{R} = \text{observation error covariance Matrix}\)
\(\mathbf{H} = \text{Linear observation operator}\)
Multivariate OI Assimilation (sect. 5.4.1)

• But where do you get B?

  • B is statistically pre-estimated, and constant in time in it’s practical implementations.

\[
\begin{align*}
\mathbf{x}_i^a &= \mathbf{x}_i^b + W(\mathbf{y}_i^o - H\mathbf{x}_i^b) \\
W &= BH^T [HBH^T + R]^{-1} \\
P^a &= [I - WH]B \\
\mathbf{x}_i^b &= M[\mathbf{x}_i^a]
\end{align*}
\]

\[
\delta\mathbf{x}^b = \mathbf{x}^b - \mathbf{x}^t
\]

True Atmos \[x^t\]
Background Error Covariance Issue

- The spatial variation of the errors is highly time dependent.
- With a constant (in time) Background Error Covariance (OI and 3D-Var) the assimilation cannot take advantage of the “errors of the day”

\[ \sigma_z^2 = (z_b - z_t)^2 = 0.08 \]

“Errors of the day” computed with the Lorenz 3 variable model: compare with rms (constant) error
“Errors of the day” obtained in the reanalysis (figs 5.6.1 and 5.6.2 in the book)

- Note that the mean error went down from 10m to 8m because of improved observing system, but the “errors of the day” (on a synoptic time scale) are still large.

- In 3D-Var/IOI not only the amplitude, but also the structure of B is fixed with time.
Flow independent error covariance

- In OI (or 3D-Var), the scalar error correlation between two points in the same horizontal surface is assumed homogeneous and isotropic. (p162 in the book)

- If we observe only Washington, D.C, we can get estimate for Richmond and Philadelphia corrections through the error correlation (off-diagonal term in B).

$$B = \begin{bmatrix}
\sigma_1^2 & \text{cov}_{1,2} & \cdots & \text{cov}_{1,n} \\
\text{cov}_{2,1} & \sigma_2^2 & \cdots & \text{cov}_{2,n} \\
\vdots & \vdots & \ddots & \vdots \\
\text{cov}_{n,1} & \text{cov}_{n,2} & \cdots & \sigma_n^2
\end{bmatrix}$$

- In OI (or 3D-Var), the scalar error correlation between two points in the same horizontal surface is assumed homogeneous and isotropic. (p162 in the book)
In OI (or 3D-Var), the error correlation between two mass points in the same horizontal surface is assumed homogeneous and isotropic. (p162 in the book)

\[
B = \begin{bmatrix}
\sigma_1^2 & \text{cov}_{1,2} & \cdots & \text{cov}_{1,n} \\
\text{cov}_{1,2} & \sigma_2^2 & \cdots & \text{cov}_{2,n} \\
\vdots & \vdots & \ddots & \vdots \\
\text{cov}_{1,n} & \text{cov}_{2,n} & \cdots & \sigma_n^2
\end{bmatrix}
\]
Suppose we have a 6hr forecast (background) and new observations. The 3D-Var Analysis doesn’t know about the errors of the day.
In Kalman Filter Data Assimilation the Background Error Covariance ($\mathbf{B}$) is advanced in time along with the analysis.

In Extended Kalman filtering a linear tangent model ($\mathbf{L}$) that transforms the initial perturbation at time $t_i$ to a final perturbation at time $t_{i+1}$.
Extended Kalman Filter Data (sect. 5.6.1)

\[ x_i^a = x_i^b + K_i (y_i^o - H x_i^b) \]
\[ P_i^a = [I - K_i H] B_i \]
\[ x_{i+1}^b = M [x_i^a] \]
\[ B_{i+1} = P_{i+1}^b = L_i P_i^a L_i^T + Q_i \]

**Symbols**
- \( x \): State
- \( x^a \): State in the autonomous frame
- \( x^b \): State in the body frame
- \( y^o \): Observation
- \( K_i \): Kalman Gain Matrix (weighting Matrix)
- \( Q_i \): Covariance Matrix of the Noise
- \( L \): Linear Tangent Model

**Equations**
- \( K_i = B_i H^T [H B_i H^T + R]^{-1} \)
With Ensemble Kalman Filtering we get perturbations pointing to the directions of the “errors of the day”

Observations $\sim 10^{5-7}$ d.o.f.

3D-Var Analysis: doesn’t know about the errors of the day

Background $\sim 10^{6-8}$ d.o.f.

Errors of the day: they lie on a low-dim attractor
Ensemble Kalman Filtering is efficient because matrix operations are performed in the low-dimensional space of the ensemble perturbations.

3D-Var Analysis: doesn’t know about the errors of the day.

Observations $\sim 10^{5-7}$ d.o.f.

Background $\sim 10^{6-8}$ d.o.f.

Errors of the day: they lie on a low-dim attractor.

Ensemble Kalman Filter Analysis: correction computed in the low dim ensemble space.
After the EnKF computes the analysis and the analysis error covariance $\mathbf{A}$, the new ensemble initial perturbations $\delta \mathbf{a}_i$ are computed:

$$
\sum_{i=1}^{k+1} \delta \mathbf{a}_i \delta \mathbf{a}_i^T = \mathbf{A}
$$

These perturbations represent the analysis error covariance and are used as initial perturbations for the next ensemble forecast.

Observations $\sim 10^{5-7}$ d.o.f.

Background $\sim 10^{6-8}$ d.o.f.

Errors of the day: they lie on the low-dim attractor.
Flow-dependence – a simple example (Miyoshi, 2004)

There is a cold front in our area…

What happens in this case?

This is not appropriate
This does reflect the flow-dependence.
An Example of the analysis corrections from 3D-Var (Kalnay, 2004)

An example with the QG system (Corazza et al, 2003)

Background error (color) and 3D-Var analysis correction (contours)

The analysis corrections due to the observations are isotropic because they don’t know about the errors of the day
An Example of the analysis corrections from EnKF (Kalnay, 2004)

QG model example of Local Ensemble KF (Corazza et al)

Background error (color) and LEKF analysis correction

The LEKF does better because it captures the errors of the day
• Updating the forecast error covariance matrix ($B$) ensures that the analysis takes into account the “errors of the day.”
• Unfortunately the EKF filter is exceedingly expensive since the linear tangent model ($L$) has size $n$, $n$=the number of degrees of freedom. For a modern model $\sim 10^6$.
• So updating $B$ is equivalent to performing $O(n)$ model integrations.
An ensemble of model forecasts are made from an ensemble of initial conditions created from the analysis and analysis error covariance.

How the ensemble of analysis initial conditions \( x^{a_i}_{i=1-K} \) is created (ensemble update) differentiates the different forms of the ensemble Kalman Filter.

The number of ensembles is usually of the order of 10-100.

How can such a small number of ensembles capture appropriately the error covariance?
• The Ensemble Kalman Filter only provides an estimate of the background error covariance.

Before with EKF:

$$B_{i+1} = P_{i+1}^b = L_i P_i^a L_i^T + Q_i$$

One form of EnKF:

$$B_{i+1} = P_{i+1}^b \approx \frac{1}{K-1} \sum_{k \neq 1}^K (x_{k}^f - \bar{x}^f) (x_{k}^f - \bar{x}^f)^T$$
Ensemble Kalman Filter (sect. 5.6.2)

- The ensemble Kalman filtering covariance is estimated from only a limited sample of ensemble members ($K \sim O(10-100)$), compared with a much larger number of degrees of freedom of the model ($n \sim O(10^7)$), so it is rank deficient ($K << n$).
- “errors of day” are the instabilities of the background flow. Strong instabilities have a few dominant shapes (perturbations lie in a low-dimensional subspace).
- It makes sense to assume that large errors are in similarly low-dimensional spaces that can be represented by a low order EnKF.
Ensemble Kalman Filter: Ensemble Update

- Again how do we update the ensemble: How do we get $x^a_i$?
- Two basic techniques
  1. Perturbed Observations Method
     - The Analysis Ensemble is formed by assimilating random perturbations added to the observations
  2. Ensemble Square Root Filter Method
     - The Analysis Ensemble is formed by transforming the forecast ensemble perturbations through a transform matrix.

**Perturbed Observation Method**

- Random perturbations added to obs
- $y^o$ to $x^a$, $x^b$

**EnSRF Method**

- Assimilate to mean Analysis $y^o$ to $x^a$, $x^b$
- Form ensemble through transformation matrix
Ensemble Update: two approaches

1. Perturbed Observations method:
   An “ensemble of data assimilations”
   - It has been proven that an observational ensemble is required (e.g., Burgers et al. 1998). Otherwise $P_{i,n\times n}^a = [I - K_iH]P_{i,b}$ is not satisfied.
   - Random perturbations are added to the observations to obtain observations for each independent cycle
     \[ y_i^{o}(k) = y_i^{o} + noise \]
   - However, it introduces a source of sampling errors when perturbing observations.
Ensemble Update: two approaches

2. Ensemble square root filter (EnSRF)
   - Observations are assimilated to update only the ensemble mean.
     \[ \bar{x}_i^a = \bar{x}_i^b + K_i(y_i^o - H\bar{x}_i^b) \]
   - Assume analysis ensemble perturbations can be formed by transforming the forecast ensemble perturbations through a transform matrix
     \[ X_i^{a(k)} = x_i^a + X_i^{a(k)} \]
     \[ X_i^a = T_i X_i^{b(k)} \]

EnSRF Method

Assimilate to mean Analysis

Form ensemble through transformation matrix

\[ x_i^b = Mx_{i-1} \]
\[ P_i^b \approx \frac{1}{K-1} \sum_{k=1}^{K} (x_k^b - \bar{x}_b)(x_k^b - \bar{x}_b)^T \]
\[ K_i = P_i^b H^T [HP_i^b H^T + R]^{-1} \]
\[ \bar{x}_i^a = \bar{x}_i^b + K_i(y_i^o - H\bar{x}_i^b) \]
Ensemble Update: two approaches

2. Ensemble square root filter (EnSRF)
   - Basically we know what $P^a$ is from the assimilation and we want to form an ensemble which satisfies that result.

   - Different Flavors of EnSRF:
     - EnSRF (Andrews 1968, Whitaker and Hamill 2002)
     - EAKF (Anderson 2001)
     - ETKF (Bishop et al 2001)
     - LETKF (Hunt 2005)

\[ X^a = (I - \tilde{K}H)X^b, \tilde{K} = \alpha K \]
\[ X^a = AX^b \]
\[ X^a = X^b T \]
\[ X^a = X^b (\tilde{P}^a)^{1/2} \]
LETKF is roughly based upon the ETKF assimilation however model space is broken into local regions where assimilation occurs simultaneously. Provides computational efficiency on parallel computer systems. Each local region can be computed separately and independently.
Summary steps of LETKF

1) Global 6 hr ensemble forecast starting from the analysis ensemble
2) Choose the observations used for each grid point
3) Compute the matrices of forecast perturbations in ensemble space $X^b$
4) Compute the matrices of forecast perturbations in observation space $Y^b$
5) Compute $P^b$ in ensemble space and its symmetric square root
6) Compute $w^a$, the k vector of perturbation weights
7) Compute the local grid point analysis and analysis perturbations.
8) Gather the new global analysis ensemble. Go to 1
LETKF algorithm summary, Hunt 2006

Forecast step (done globally): advance the ensemble 6 hours, global model size \( n \)
\[
x_n^{b(i)} = M(x_{n-1}^{a(i)}) \quad i = 1, \ldots, k
\]

Analysis step (done locally). Local model dimension \( m \), locally used obs \( s \)
\[
X^b = X - \bar{X}^b \quad (mxk) \quad Y^b = H(X) - H(\bar{X}^b) \approx HX^b \quad (sxk)
\]
\[
\tilde{P}^a = \left[ (k - 1)I + \left(HX^b\right)^T R^{-1}HX^b \right]^{-1} \quad \text{in ensemble space (kxk)}
\]
\[
P^a = X^aT X^a = X^{bT} \tilde{P}^a X^b \quad \text{in model space (mxm)}
\]
\[
X^a = X^b \left(\tilde{P}^a\right)^{1/2} \quad \text{Ensemble analysis perturbations in model space (mxk)}
\]
\[
w^a_n = \tilde{P}^a Y^{bT} R^{-1} (y^o_n - \bar{y}^b_n) \quad \text{Analysis increments in ensemble space (kx1)}
\]
\[
\bar{X}^a_n = \bar{X}^a_n + X^b w^a_n \quad \text{Analysis (mean) in model space (mx1)}
\]
\[
x^a_n = \bar{x}^a_n + x^a \quad \text{Analysis ensemble in model space (mxk)}
\]

We finally gather all the analysis and analysis perturbations from each grid point and construct the new global analysis ensemble (nxk) and go to next forecast step
As was discussed last time without any help the Kalman filter tends to calculate decreased error with each time step eventually leading to disregarding any new observations.

A tuned error covariance inflation factor must be used to balance out the decrease in error from each assimilation time step
- Too small of factor: Converges on model, Disregards Observations
- Too large of factor: Disregards model, Converges on Observations
• Set-up the Lorenz-96 model
  – Create Control forecast
  – Create Ensemble of forecasts
• Set-up fake observations:
  – Create Observations by perturbing the Control forecast
• Set-up LETKF (without localization) to assimilate observations
• Investigate the impact of different variables on results:
  – Number of observations
  – Observation Error
  – Size of the ensemble
  – Inflation factor
References and thanks:


