Data assimilation using ensemble methods for hurricane forecasting

Pushkar Kumar Jain

The University of Texas at Austin
Aerospace Engineering and Engineering Mechanics
Cockrell School of Engineering

INSTITUTE FOR COMPUTATIONAL ENGINEERING & SCIENCES

COMPUTATIONAL HYDRAULICS GROUP
THE UNIVERSITY OF TEXAS AT AUSTIN

Outline

- Introduction
- Kalman assimilation derivation
- Data assimilation techniques
- Forecasting
- Extended Kalman filter
- Ensemble Kalman filter
- Implementation
  - PDAF + GeoCLAW integration
  - Test cases
- Conclusion

Introduction

- Data assimilation addresses the problem of incorporating observations into a model of the system in an optimal way
- Given a noisy discrete model of dynamics of the system and noisy observations of the system, find estimates of the state of the system
- The objective is to investigate computational methods to advance the science of real time prediction of coastal and hydrological hazards and improve the designs of observational systems

"Data driven simulation"

Motivation

Problem position: Let's say we want to produce the best estimate of the temperature of the room, given:

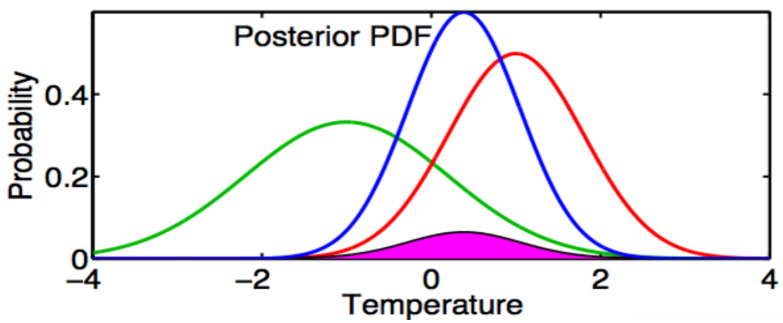
- A prior estimate: T_b (also called background)
  - Maybe previously measured some time ago, or a model that predicts temperature as a function of room size, #people etc.
- An observation: T_o using a thermometer

Let the true temperature be T^*

- The errors in T_b and T_o are:

$$\epsilon_b = T_b - T^*$$

$$\epsilon_o = T_o - T^*$$



- We will assume that –

$$\bar{\epsilon}_b = \bar{\epsilon}_o = 0$$

ϵ_b and ϵ_o are not correlated

Minimum variance approach

The estimated temperature is a linear combination of T_b and T_o -

$$T_a = \alpha T_o + \beta T_b + \gamma$$

$$\epsilon_a = T_a - T^*$$

$$T_a = \epsilon_a + T^* = \alpha(\epsilon_o + T^*) + \beta(\epsilon_b + T^*) + \gamma$$

Taking the mean and rearranging gives,

$$\overline{\epsilon_a} = (\alpha + \beta - 1)T^* + \gamma$$

$$\alpha + \beta = 1$$

$$\Longrightarrow T_a = \alpha T_o + (1-\alpha)T_b \text{ and } \epsilon_a = \alpha\epsilon_o + (1-\alpha)\epsilon_b$$

Optimality criteria - Minimization of variance of the estimate ($\overline{\epsilon_a^2}$) gives

$$\frac{d\overline{\epsilon_a^2}}{d\alpha} = \frac{d}{d\alpha}\left(\overline{\epsilon_o^2} + 2\alpha(1-\alpha)\overline{\epsilon_o\epsilon_b} + (1-\alpha^2)\epsilon_b^2\right) = 0$$

$$\Longrightarrow \alpha = \frac{\epsilon_b^2}{\epsilon_b^2 + \epsilon_o^2}$$

Extension to multiple dimensions

Nomenclature -

$\boldsymbol{x}_b$: State of a numerical model at some time. Also called background state.

$\boldsymbol{y}$: Collection of observations at different locations

- No longer a one-to-one correspondence between observation vector and background vector.
  - Observations might not be at grid points
  - Observed variables might not correspond with any variables of the model
- Assume an observation operator $\mathcal{H}$ (does not introduce any error), so that –

$$\mathcal{H}(\boldsymbol{x}^*) = \boldsymbol{y}^*,$$

Where x^* is the true state and y^* are observed quantities

Extension to multiple dimensions

Similar to the scalar case, the analysis case can be written as linear combination of available information.

$$\boldsymbol{x}_a = \boldsymbol{F}\boldsymbol{x}_b + \boldsymbol{G}\mathcal{H}(\boldsymbol{x}_b) + \boldsymbol{K}\boldsymbol{y} + \boldsymbol{c}$$

Error free inputs ($\boldsymbol{x}_b = \boldsymbol{x}^*$ and $\boldsymbol{y} = \boldsymbol{y}^*$) must result in error-free analysis –

$$\boldsymbol{x}^* = F\boldsymbol{x}^* + G\mathcal{H}(\boldsymbol{x}^*) + \boldsymbol{K}\boldsymbol{y}^* + \boldsymbol{c}$$

$$\boldsymbol{F} + \boldsymbol{G}\mathcal{H}(\cdot) = \boldsymbol{I} - \boldsymbol{K}\mathcal{H}(\cdot)$$

$$\boldsymbol{x}_a = \boldsymbol{x}_b + \boldsymbol{K}(\boldsymbol{y} - \mathcal{H}(\boldsymbol{x}_b))$$

$\boldsymbol{K}$ is called the gain matrix (equivalent to α in scalar case). It determines the weight given to the observations.

Error statistics

Define the analysis error statisitic as: $\boldsymbol{\epsilon_a} = \boldsymbol{x_a} - \boldsymbol{x}^*$

An important assumption that the errors are small, so that –

$$\mathcal{H}(\boldsymbol{x_b}) = \mathcal{H}(\boldsymbol{x}^*) + H\boldsymbol{\epsilon_b} + \mathcal{O}(\boldsymbol{\epsilon_b^2})$$

where, H is the Tangent linear operator (Jacobian) associated with $\mathcal{H}$

Previously obtained, $\boldsymbol{x_a} = \boldsymbol{x_b} + \boldsymbol{K}\big(\boldsymbol{y} - \mathcal{H}(\boldsymbol{x_b})\big)$

gives, $\boldsymbol{x_a} - \boldsymbol{x}^* = \boldsymbol{x_b} - \boldsymbol{x}^* + \boldsymbol{K}(\boldsymbol{y} - (\mathcal{H}(x^*) + \boldsymbol{H}\epsilon_b))$

$$\boldsymbol{\epsilon_a} = \boldsymbol{\epsilon_b} + \boldsymbol{K}(\boldsymbol{\epsilon_o} - \boldsymbol{H}\boldsymbol{\epsilon_b}))$$

Note that, $\overline{\boldsymbol{\epsilon_o}} = \overline{\boldsymbol{\epsilon_b}} = \boldsymbol{0} \implies \overline{\boldsymbol{\epsilon_a}} = \boldsymbol{0}$

Kalman gain

- In scalar example, the error variance was minimized.
- In multiple dimensions, analysis error covariance $\left(\overline{\boldsymbol{\epsilon}_a \boldsymbol{\epsilon}_a^T}\right)$ is formed and $\mathrm{tr}\left(\overline{\boldsymbol{\epsilon}_a \boldsymbol{\epsilon}_a^T}\right)$ is minimized.

$$\overline{\boldsymbol{\epsilon}_a \boldsymbol{\epsilon}_a^T} = (\boldsymbol{I} - \boldsymbol{K}\boldsymbol{H})\overline{\boldsymbol{\epsilon}_b \boldsymbol{\epsilon}_b^T}(\boldsymbol{I} - \boldsymbol{K}\boldsymbol{H})^T + \boldsymbol{K}\overline{\boldsymbol{\epsilon}_o \boldsymbol{\epsilon}_o^T}\boldsymbol{K}^T$$

$$\frac{\partial tr\left(\overline{\boldsymbol{\epsilon}_a \boldsymbol{\epsilon}_a^T}\right)}{\partial \boldsymbol{K}} = 0 \implies \boldsymbol{K} = \overline{\boldsymbol{\epsilon}_b \boldsymbol{\epsilon}_b^T}\boldsymbol{H}^T\left[\boldsymbol{H}\overline{\boldsymbol{\epsilon}_b \boldsymbol{\epsilon}_b^T}\boldsymbol{H}^T + \overline{\boldsymbol{\epsilon}_o \boldsymbol{\epsilon}_o^T}\right]^{-1}$$

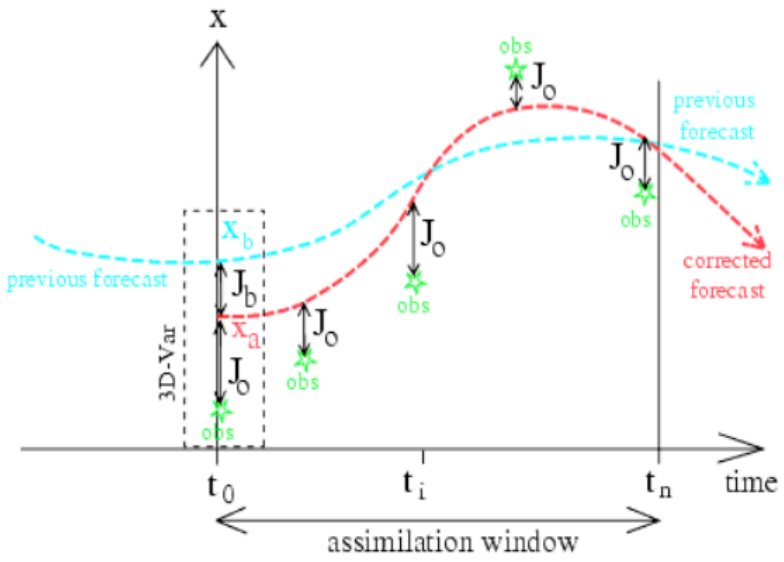
$$\boldsymbol{K} = \boldsymbol{P}_b\boldsymbol{H}^T[\boldsymbol{H}\boldsymbol{P}_b\boldsymbol{H}^T + \boldsymbol{R}]^{-1} \equiv [(\boldsymbol{P}_b)^{-1} + \boldsymbol{H}^T\boldsymbol{R}\boldsymbol{H}]^{-1}\boldsymbol{H}^T\boldsymbol{R}^{-1}$$

Data assimilation techniques

- **Optimal interpolation –**
  - Splits the global analysis into a number of boxes which can be analyzed independently.
  - Restricts the observations within boxes by selecting 'surrounding' observation points, thereby reducing the dimension of $[\boldsymbol{H}\boldsymbol{P}_{\boldsymbol{b}}\boldsymbol{B}^T + \boldsymbol{R}]$
  - Cheap and easy implementation
  - $P_b H^T$ becomes difficult for complex observation operators
- **3D var –**
  - Avoids calculating K by minimizing the cost function - $\frac{1}{2}(\boldsymbol{x_b} - \boldsymbol{x})^T(\boldsymbol{P_b})^{-1}(\boldsymbol{x_b} - \boldsymbol{x}) + \frac{1}{2}(\boldsymbol{y} - \mathcal{H}(\boldsymbol{x}))^T\boldsymbol{R}^{-1}(\boldsymbol{y} - \mathcal{H}(\boldsymbol{x}))$
  - Easy to use with complex observation operators
  - Can add regularization terms
  - Need to implement tangent linear operators (Jacobians) which is computationally expensive

Data assimilation techniques

- **4D var –**
  - Takes account of evolution of system between forecast time and observation time. Find a model trajectory minimizing the distance between the observations.
  - Propagates model fields to various times at which the observations were taken.
  - Avoids calculating K by minimizing the cost function - $\frac{1}{2}(\boldsymbol{x_b} - \boldsymbol{x})^T(\boldsymbol{P_b})^{-1}(\boldsymbol{x_b} - \boldsymbol{x}) + \frac{1}{2}\sum_{k=0}^{K}(\boldsymbol{y_k} - \mathcal{G}_k(\boldsymbol{x}))^T\boldsymbol{R}^{-1}(\boldsymbol{y_k} - \mathcal{G}_k(\boldsymbol{x}))$
  - Need to implement tangent linear operators (Jacobians) which is computationally exp
- **Extended Kalman filtering (EKF) –**
  - Provides an evolving forecast uncertainty in the form of covariance matrix
- **Ensemble Kalman filtering**
  - Uses an ensemble of state estimates.
  - Avoids the step to evolve the covariance .and does not use tangent linear operators.



Forecasting

- Given a prior and an observation state at a time, we have seen how to obtain the assimilated state
- We are interested in a sequence of analyses for various times t_0, t_1, t_2 …
- We have a forecasting model given by –

$$\boldsymbol{x_b}(t+1) = \mathcal{M}\big(\boldsymbol{x_b}(t)\big)$$

- For each analysis we require a prior estimate $\boldsymbol{x_b}$ of the state. Our best prior estimate is given by forecasting of the previous state

$$\boldsymbol{x_b}(t+1) = \mathcal{M}\big(\boldsymbol{x_a}(t)\big)$$

- Similar to derivation of analysis error covariance (P_a), background error covariance (P_b) can be derived as –

$$\boldsymbol{P_b} = \boldsymbol{M}\boldsymbol{P_a}\boldsymbol{M}^T + \boldsymbol{Q}$$

The Kalman filter is a sequential data assimilation technique for use with linear models of system dynamics and observations.

Extended Kalman filter (EKF)

- Unlike 3D Var and 4D var, EKF provides an evolving forecast uncerainty
- Combining the forecasting stage and the analysis stage, the following equations define the EKF –

$$\boldsymbol{x_b}(t+1) = \mathcal{M}\big(\boldsymbol{x_b}(t)\big) + \boldsymbol{\eta}(\boldsymbol{t})$$

$$\boldsymbol{P_b} = \boldsymbol{M}\boldsymbol{P_a}\boldsymbol{M}^T + \boldsymbol{Q}$$

$$\boldsymbol{K} = \boldsymbol{P_b}\boldsymbol{H}^T[\boldsymbol{H}\boldsymbol{P_b}\boldsymbol{H}^T + \boldsymbol{R}]^{-1} \equiv [(\boldsymbol{P_b})^{-1} + \boldsymbol{H}^T\boldsymbol{R}\boldsymbol{H}]^{-1}\boldsymbol{H}^T\boldsymbol{R}^{-1}$$

$$\boldsymbol{x}_a = \boldsymbol{x}_b + \boldsymbol{K}(\boldsymbol{y} - \mathcal{H}(\boldsymbol{x}_b))$$

$$\boldsymbol{P_a} = (\boldsymbol{I} - \boldsymbol{K}\boldsymbol{H})\boldsymbol{P_b}(\boldsymbol{I} - \boldsymbol{K}\boldsymbol{H})^T + \boldsymbol{K}\boldsymbol{P_o}\boldsymbol{K}^T$$

Note: The “extended” refers to the fact that non-linear observation operators are used and the state is propagated using non-linear model. The original Kalman filter was purely linear $\mathcal{H}$ and $\mathcal{M}$

- Bottleneck –
  - Need to implement tangent linear operators H and M.
  - Calculate inverse of large matrices – Impractical for large dimesnional systems
  - Covariance propagation step $P_a(t) \rightarrow P_b(t+1)$ is expensive

Algorithm

INPUTS

- Prior guess of initial state, $\boldsymbol{X}_b$
- Description of how state evolves in time, $\boldsymbol{x}_{new} = \mathcal{M}(\boldsymbol{x}_{old})$
- Observations through the assimilation window, $\boldsymbol{y}$
- Description of how observations map to the state, $\mathcal{H}(\cdot)$
- Prior guess error covariance matrix, $\boldsymbol{B}$
- Observation error covariance matrix, $\boldsymbol{R}$
- Model error covariance matrix, $\boldsymbol{Q}$

Algorithm - EKF

1. Set $\boldsymbol{x_b}(0) = \boldsymbol{x_a}(0) = \boldsymbol{X}_b$ and $\boldsymbol{P_b}(0) = \boldsymbol{B}$

2. <u>Prediction step</u>:

   a) Propagate prior state in time: $\boldsymbol{x}_b\ (t+1) = \mathcal{M}\big(\boldsymbol{x}_a\ (t)\big)$

   b) Propagate prior guess error covariance in time:

   $$\boldsymbol{P}_b\ (t+1) = \boldsymbol{M}\boldsymbol{P}_a\ (t)\boldsymbol{M}^T$$

   M is the tangent linear of $\mathcal{M}(x)$ evaluated at $\boldsymbol{x_b}\ (t)$.

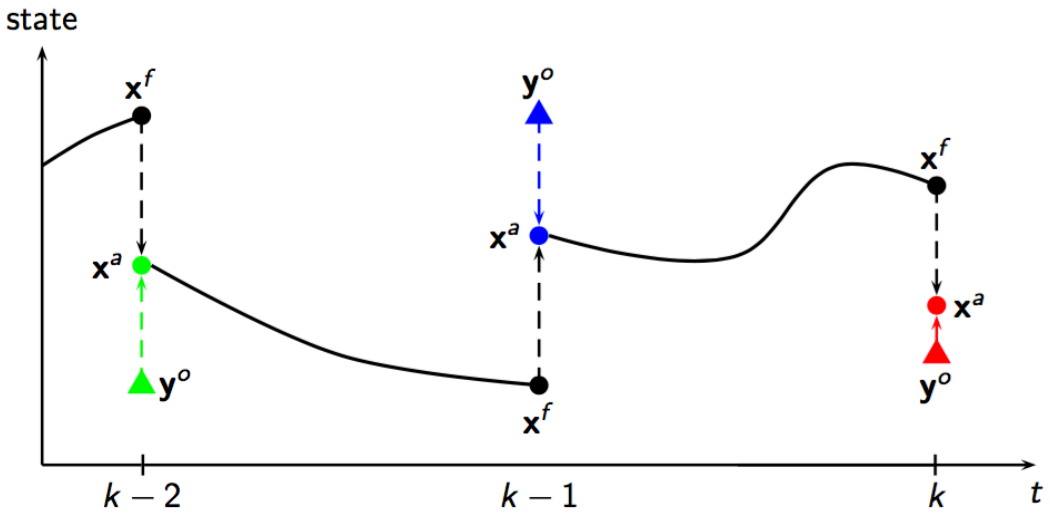
3. <u>Update step</u>:

   a) Calculate Kalman gain: $\boldsymbol{K} = \boldsymbol{P_b}\boldsymbol{H}^T(\boldsymbol{R} + \boldsymbol{H}\boldsymbol{P_b}\boldsymbol{H}^T)^{-1}$

   b) Calculate estimate of state: $\boldsymbol{x}_a = \boldsymbol{x}_b + \boldsymbol{K}(\boldsymbol{y} - \mathcal{H}(\boldsymbol{x}_b))$

   c) Calculate analysis error covariance: $\boldsymbol{P_a} = (\boldsymbol{I} - \boldsymbol{K}\boldsymbol{H})\boldsymbol{P_b}$

4. Repeat 2 and 3 till all observations are assimilated



Kalman filters for large dimensional systems

- Approximate Kalman filters have been developed that rely on low rank approximation of the covariance matrices of background and anaysis error
- Suppose $\boldsymbol{P_b}$ has rank $\mathrm{m} \ll N$, with $\boldsymbol{P_b} = \boldsymbol{X}\boldsymbol{X}^{\boldsymbol{T}}$, where $\boldsymbol{X}$ is $N \times m$.

$$\boldsymbol{K} = \boldsymbol{X}(\boldsymbol{HX})^T[(\boldsymbol{HX})(\boldsymbol{HX})^T + \boldsymbol{R}]^{-1}$$

$$\boldsymbol{P_a} = (\boldsymbol{I} - \boldsymbol{KH})\boldsymbol{P_b}(\boldsymbol{I} - \boldsymbol{KH})^T + \boldsymbol{KRK}^T \equiv \boldsymbol{XWX}^T$$

where W is of size $m \times m$

- Covariance propagation becomes: $\boldsymbol{P_b} = \boldsymbol{MPM}^T + \boldsymbol{Q} = (\boldsymbol{MX})\boldsymbol{W}(\boldsymbol{MX})^T + \boldsymbol{Q}$
- Advantages –
  - To calculate $\boldsymbol{K}$, apply $\boldsymbol{H}$ to m columns of $\boldsymbol{X}$, rather than N columns of $\boldsymbol{P_b}$
  - Covariance propagation requires only m integrations of the tangent linear model

Kalman filters for large dimensional systems

- The low rank approximation of covariance matrices suffer from spurious long-distance correlations causing increments in regions where there are no observations
- Remedies –
  - Local analysis
  - Shur product modification of covariances

Ensemble Kalman filtering (EnKF)

- Uses an ensemble of state estimates instead of single state estimate. This is a Monte Carlo approximation to pdfs
- Reduced rank Kalman filters that construct their covariance matrices as sample covariance matrices.

$$\mathbf{P_b} = \frac{1}{\mathrm{n}-1}\sum\left(\mathbf{x_b^{(i)}} - \bar{\mathbf{x}}\right)\left(\mathbf{x_b^{(i)}} - \bar{\mathbf{x}}\right)^{\mathrm{T}}$$

- Unlike Extended Kalman filtering, avoids the use of tangent linear operators by approximating –

$$P_b H^T \approx \frac{1}{\mathrm{n}-1}\sum\left(\mathbf{x_b^{(i)}} - \bar{\mathbf{x}}\right)\left(\mathcal{H}\left(\mathbf{x_b^{(i)}}\right) - \overline{\mathcal{H}(\boldsymbol{x})}\right)^{\mathrm{T}}$$

$$HP_b H^T \approx \frac{1}{\mathrm{n}-1}\sum\left(\mathcal{H}\left(\mathbf{x_b^{(i)}}\right) - \overline{\mathcal{H}(\boldsymbol{x})}\right)\left(\mathcal{H}\left(\mathbf{x_b^{(i)}}\right) - \overline{\mathcal{H}(\boldsymbol{x})}\right)^{\mathrm{T}}$$

where, $\overline{\mathcal{H}(\boldsymbol{x})} = \frac{1}{\mathrm{n}}\sum\mathcal{H}\left(\mathbf{x_b^{(i)}}\right)$

Note: To calculate $\boldsymbol{K} = \boldsymbol{P_b}\boldsymbol{H}^T[\boldsymbol{H}\boldsymbol{P_b}\boldsymbol{H}^T + \boldsymbol{R}]^{-1}$, huge matrix $\boldsymbol{P_b}$ is never explicitly formed.

Algorithm

INPUTS

- Prior guess of initial state, $\boldsymbol{X}_b$
- Description of how state evolves in time, $\boldsymbol{x}_{new} = \mathcal{M}(\boldsymbol{x}_{old})$
- Observations through the assimilation window, $\boldsymbol{y}$
- Description of how observations map to the state, $\mathcal{H}(\cdot)$
- Prior guess error covariance matrix, $\boldsymbol{B}$
- Observation error covariance matrix, $\boldsymbol{R}$
- Model error covariance matrix, $\boldsymbol{Q}$

Algorithm - EnKF

1. Generate ensemble of size N:

$$\boldsymbol{x}_{\boldsymbol{b}}^{(\boldsymbol{i})} = \boldsymbol{x}_{\boldsymbol{b}} + \boldsymbol{\eta}\ for\ i = 1, N \text{ and } \eta \sim \mathcal{N}(\mathbf{0}, \boldsymbol{B})$$

1. <u>Prediction step</u>:

   a) Propagate each ensemble member forward in time:

$$\boldsymbol{x}_{\boldsymbol{b}}^{(\boldsymbol{i})}(t+1) = \mathcal{M}\left(\boldsymbol{x}_{\boldsymbol{b}}^{(\boldsymbol{i})}(t)\right)$$

2. <u>Update step</u>:

   a) $\mathbf{P_b}$ is evolved B matrix: $\mathbf{P_b} = \frac{1}{\mathrm{N}-1}\sum\left(\mathbf{x}_{\mathbf{b}}^{(\mathrm{i})} - \bar{\mathbf{x}}\right)\left(\mathbf{x}_{\mathbf{b}}^{(\mathrm{i})} - \bar{\mathbf{x}}\right)^{\mathrm{T}}$

   b) Calculate Kalman gain: $\boldsymbol{K} = \boldsymbol{P_b}\boldsymbol{H}^T(\boldsymbol{R} + \boldsymbol{H}\boldsymbol{P_b}\boldsymbol{H}^{\boldsymbol{T}})^{-1}$

   c) Update each ensemble member:

$$\boldsymbol{x}_{\boldsymbol{b}}^{(i)} \leftarrow \boldsymbol{x}_{\boldsymbol{b}}^{(i)} + \boldsymbol{K}\left(\boldsymbol{y} + \boldsymbol{\epsilon} - \mathcal{H}\left(\boldsymbol{x}_{\boldsymbol{b}}^{(i)}\right)\right), \text{ where } \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{R})$$

   a) Calculate analysis error covariance: $\boldsymbol{P_a} = (\boldsymbol{I} - \boldsymbol{K}\boldsymbol{H})\boldsymbol{P_b}$ **(if needed)**

3. Repeat 2 and 3 till all observations are assimilated

Algorithm - EKF

1. Generate ensemble of size N:

$$\boldsymbol{x}_{\boldsymbol{b}}^{(\boldsymbol{i})} = \boldsymbol{x}_{\boldsymbol{b}} + \boldsymbol{\eta}\ for\ i = 1, N \text{ and } \eta \sim \mathcal{N}(\mathbf{0}, \boldsymbol{B})$$

2. <u>Prediction step</u>:

   a) Propagate each ensemble member forward in time:

$$\boldsymbol{x}_{\boldsymbol{b}}^{(\boldsymbol{i})}(t+1) = \mathcal{M}\left(\boldsymbol{x}_{\boldsymbol{b}}^{(\boldsymbol{i})}(t)\right)$$

3. <u>Update step</u>:

   a) $\mathbf{P_b}$ is evolved B matrix: $\mathbf{P_b} = \frac{1}{\mathrm{N}-1}\sum\left(\mathbf{x}_{\mathbf{b}}^{(\mathrm{i})} - \bar{\mathbf{x}}\right)\left(\mathbf{x}_{\mathbf{b}}^{(\mathrm{i})} - \bar{\mathbf{x}}\right)^{\mathrm{T}}$

   b) Calculate Kalman gain: $\boldsymbol{K} = \boldsymbol{P_b}\boldsymbol{H}^T(\boldsymbol{R} + \boldsymbol{H}\boldsymbol{P_b}\boldsymbol{H}^{\boldsymbol{T}})^{-1}$

   c) Update each ensemble member:

$$\boldsymbol{x}_{\boldsymbol{b}}^{(i)} \leftarrow \boldsymbol{x}_{\boldsymbol{b}}^{(i)} + \boldsymbol{K}\left(\boldsymbol{y} + \boldsymbol{\epsilon} - \mathcal{H}\left(\boldsymbol{x}_{\boldsymbol{b}}^{(i)}\right)\right), \text{ where } \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{R})$$

   d) Calculate analysis error covariance: $\boldsymbol{P_a} = (\boldsymbol{I} - \boldsymbol{K}\boldsymbol{H})\boldsymbol{P_b}$ **(if needed)**

4. Repeat 2 and 3 till all observations are assimilated

Algorithm

Extended Kalman filter (EKF)

1. Set $\boldsymbol{x_b}(0) = \boldsymbol{x_a}(0) = \boldsymbol{X}_b$ and $\boldsymbol{P_b}(0) = \boldsymbol{B}$

2. <u>Prediction step</u>:

   a) Propagate prior state in time: $\boldsymbol{x}_b\ (t+1) = \mathcal{M}\big(\boldsymbol{x}_a\ (t)\big)$

   b) Propagate prior guess error covariance in time:

   $$\boldsymbol{P}_b\ (t+1) = \boldsymbol{M}\boldsymbol{P}_a\ (t)\boldsymbol{M}^T$$

   M is the tangent linear of $\mathcal{M}(x)$ evaluated at $\boldsymbol{x_b}\ (t)$.

3. <u>Update step</u>:

   a) Calculate Kalman gain: $\boldsymbol{K} = \boldsymbol{P_b}\boldsymbol{H}^T(\boldsymbol{R} + \boldsymbol{H}\boldsymbol{P_b}\boldsymbol{H}^T)^{-1}$

   b) Calculate estimate of state: $\boldsymbol{x}_a = \boldsymbol{x}_b + \boldsymbol{K}(\boldsymbol{y} - \mathcal{H}(\boldsymbol{x}_b))$

   c) Calculate analysis error covariance: $\boldsymbol{P_a} = (\boldsymbol{I} - \boldsymbol{K}\boldsymbol{H})\boldsymbol{P_b}$

4. Repeat 2 and 3 till all observations are assimilated

Ensemble Kalman filter (EnKF)

1. Generate ensemble of size N:

   $\boldsymbol{x_b^{(i)}} = \boldsymbol{x_b} + \boldsymbol{\eta}\ for\ i = 1, N$ and $\eta \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{B})$

2. <u>Prediction step</u>:

   a) Propagate each ensemble member forward in time:

   $$\boldsymbol{x_b^{(i)}}(t+1) = \mathcal{M}\left(\boldsymbol{x_b^{(i)}}(t)\right)$$

3. <u>Update step</u>:

   a) $\mathbf{P_b}$ is evolved B matrix: $\mathbf{P_b} = \frac{1}{\mathrm{N}-1}\sum\left(\mathbf{x_b^{(i)}} - \bar{\mathbf{x}}\right)\left(\mathbf{x_b^{(i)}} - \bar{\mathbf{x}}\right)^{\mathrm{T}}$

   b) Calculate Kalman gain: $\boldsymbol{K} = \boldsymbol{P_b}\boldsymbol{H}^T(\boldsymbol{R} + \boldsymbol{H}\boldsymbol{P_b}\boldsymbol{H}^T)^{-1}$

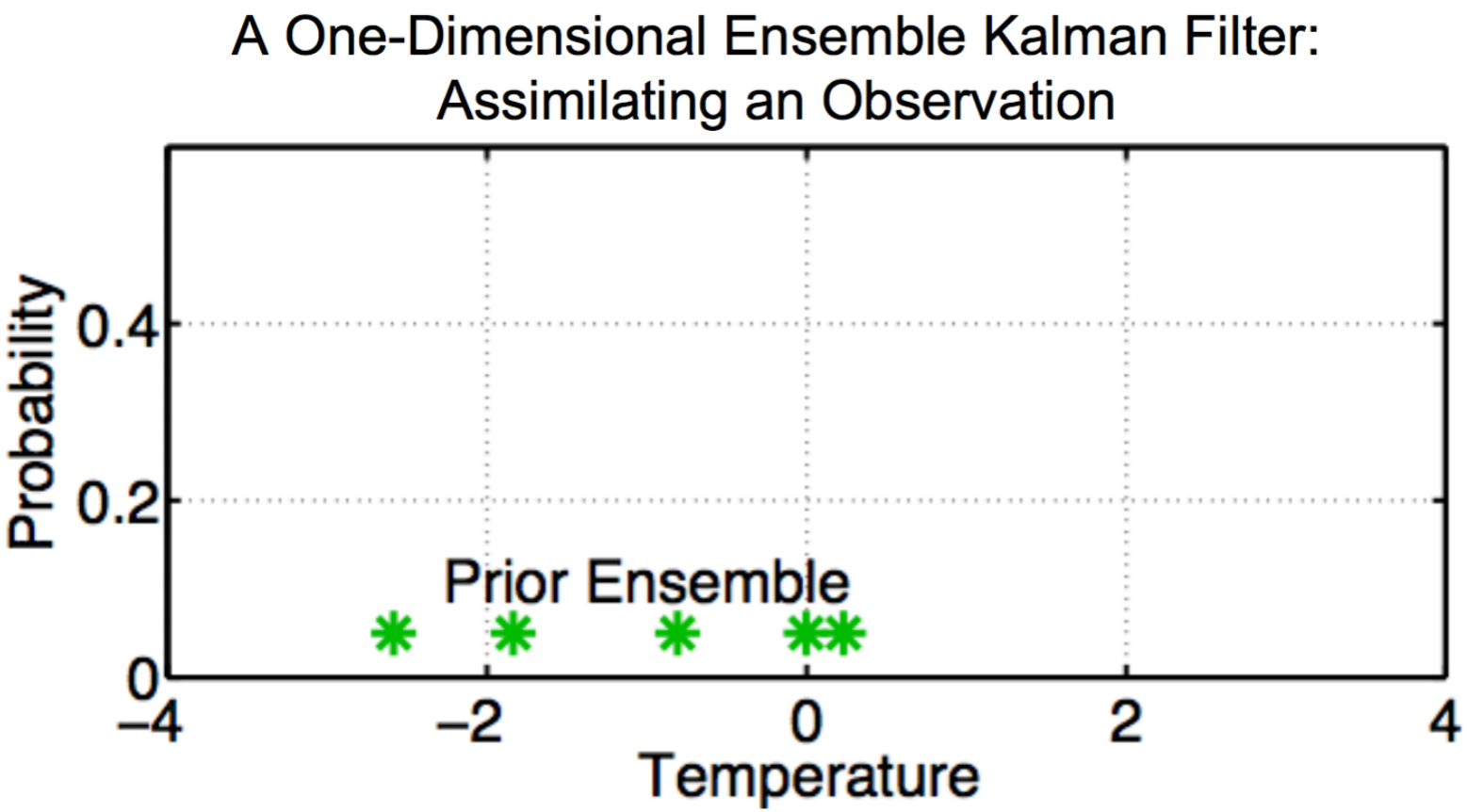
   c) Update each ensemble member:

   $$\boldsymbol{x_b^{(i)}} \leftarrow \boldsymbol{x_b^{(i)}} + \boldsymbol{K}\left(\boldsymbol{y} + \boldsymbol{\epsilon} - \mathcal{H}\left(\boldsymbol{x_b^{(i)}}\right)\right), \text{ where } \boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{R})$$

   d) Calculate analysis error covariance: $\boldsymbol{P_a} = (\boldsymbol{I} - \boldsymbol{K}\boldsymbol{H})\boldsymbol{P_b}$ **(if needed)**

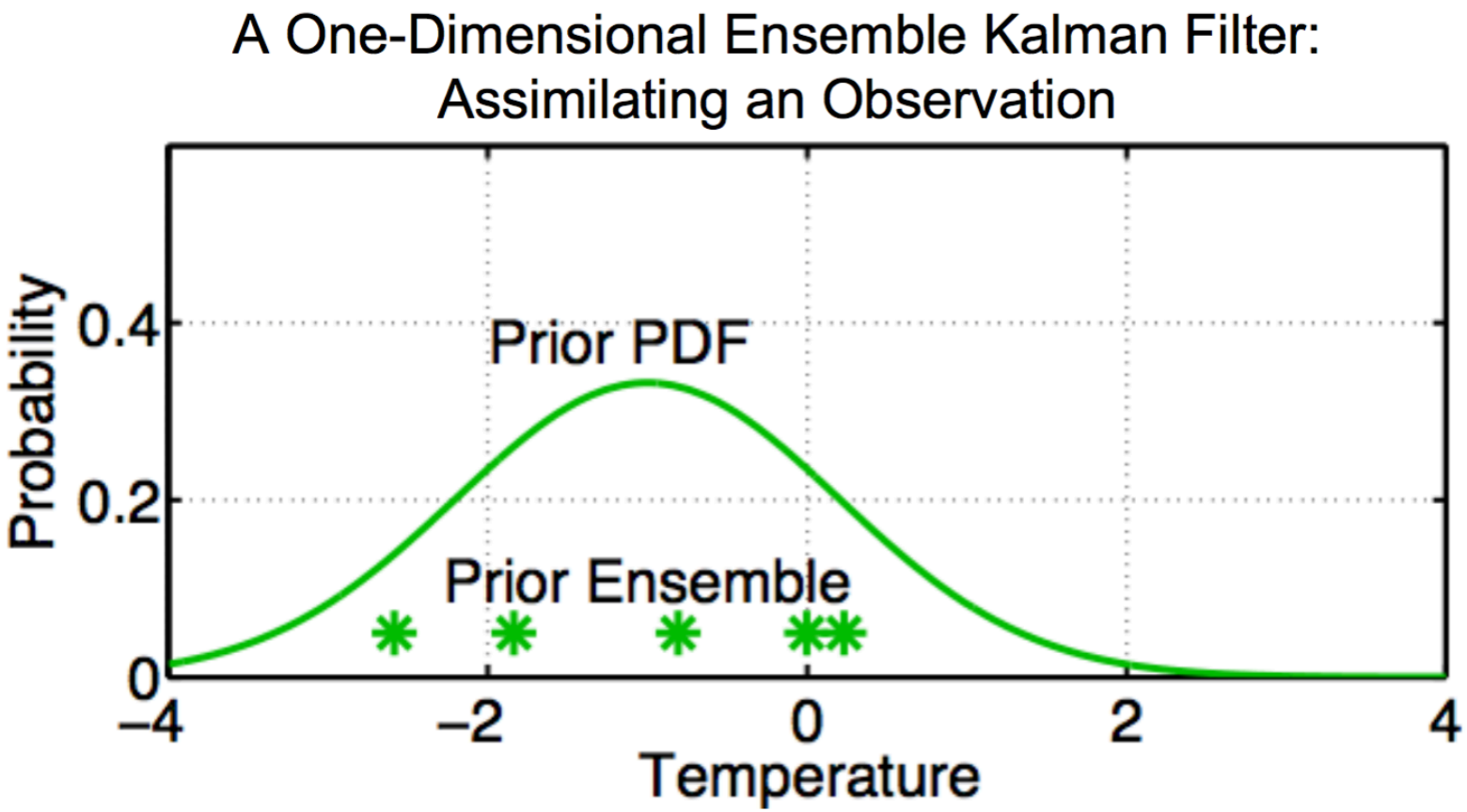
4. Repeat 2 and 3 till all observations are assimilated.

EnKF in action



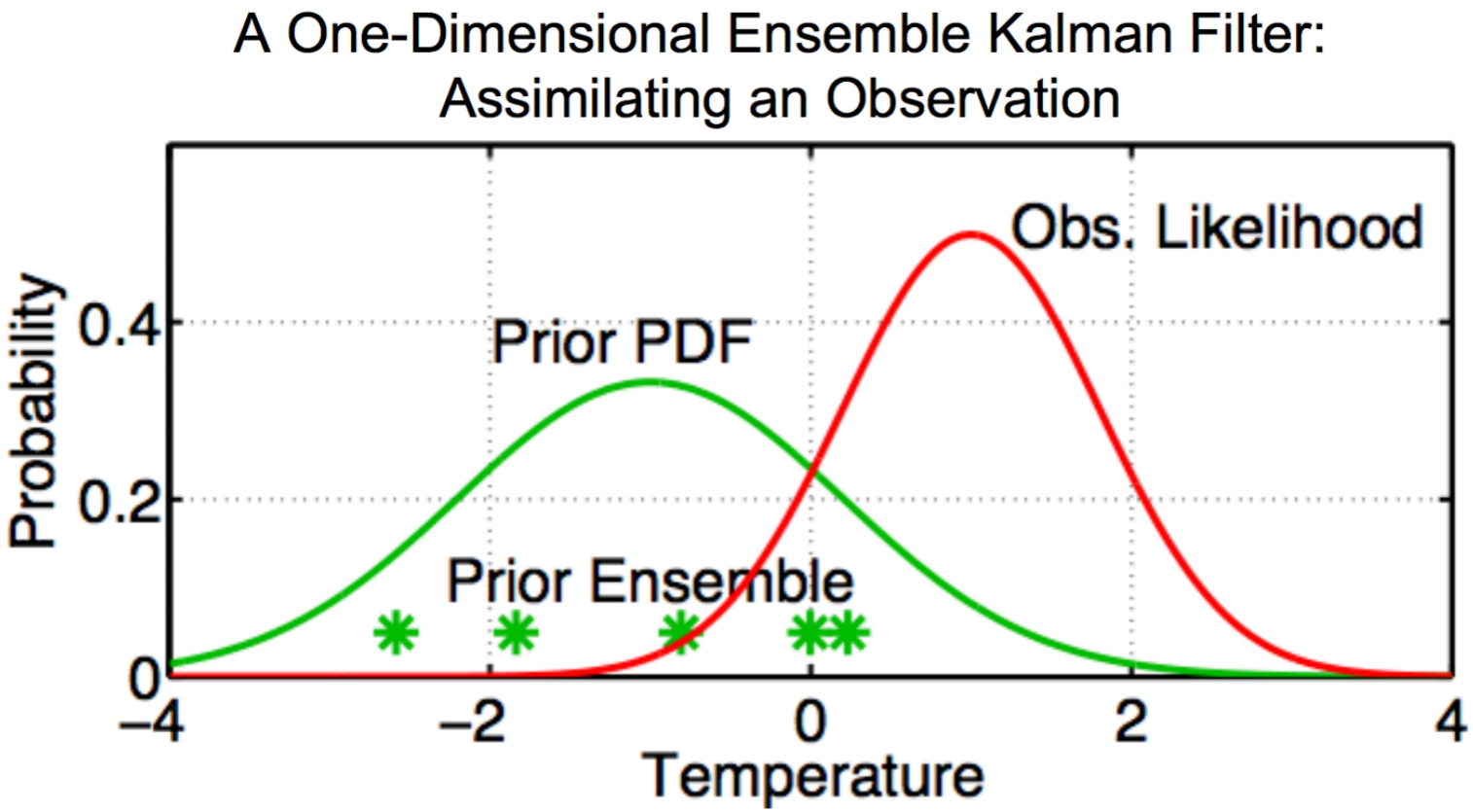
EnKF in action

Fit a Gaussian sample



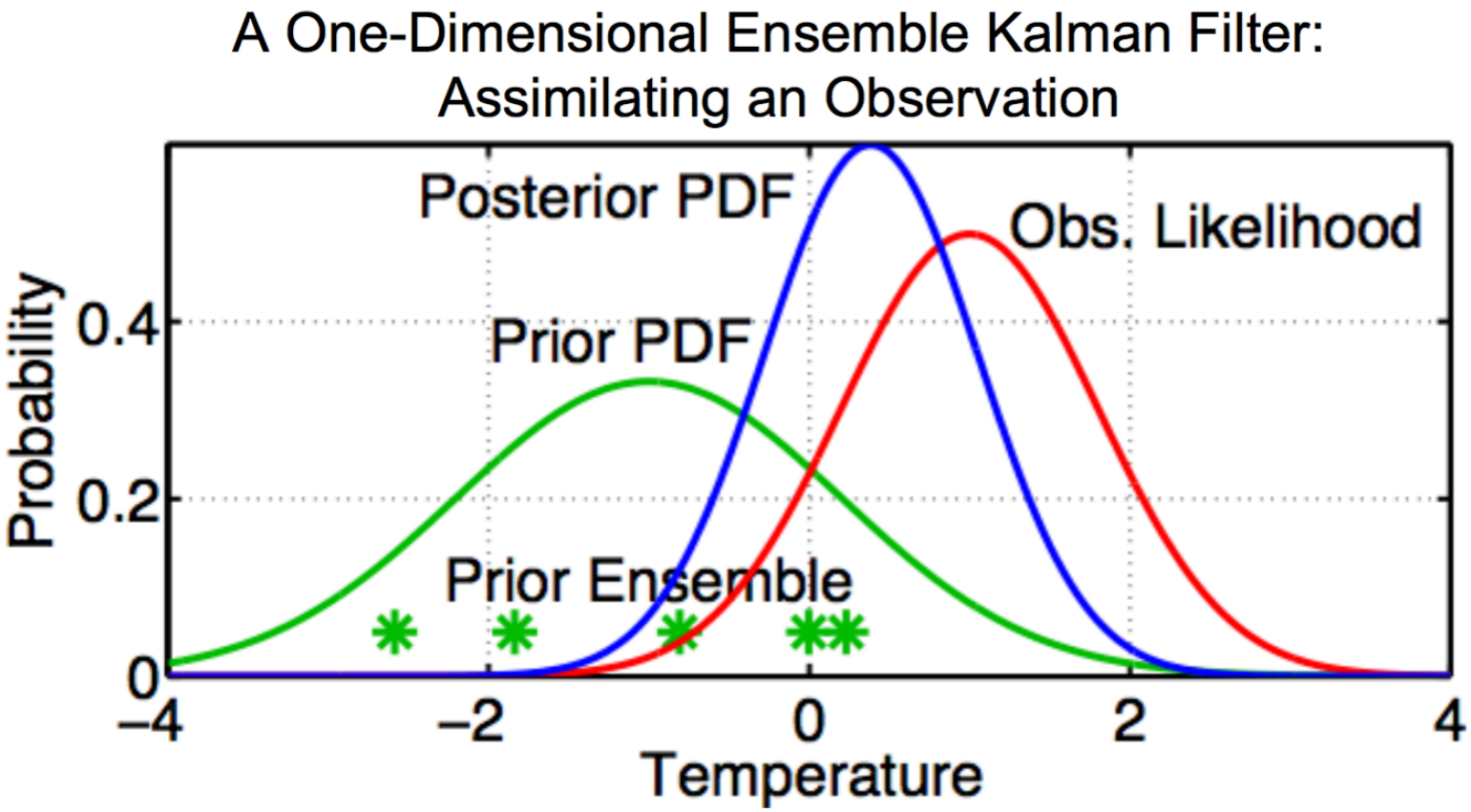
EnKF in action

Get observation likelihood



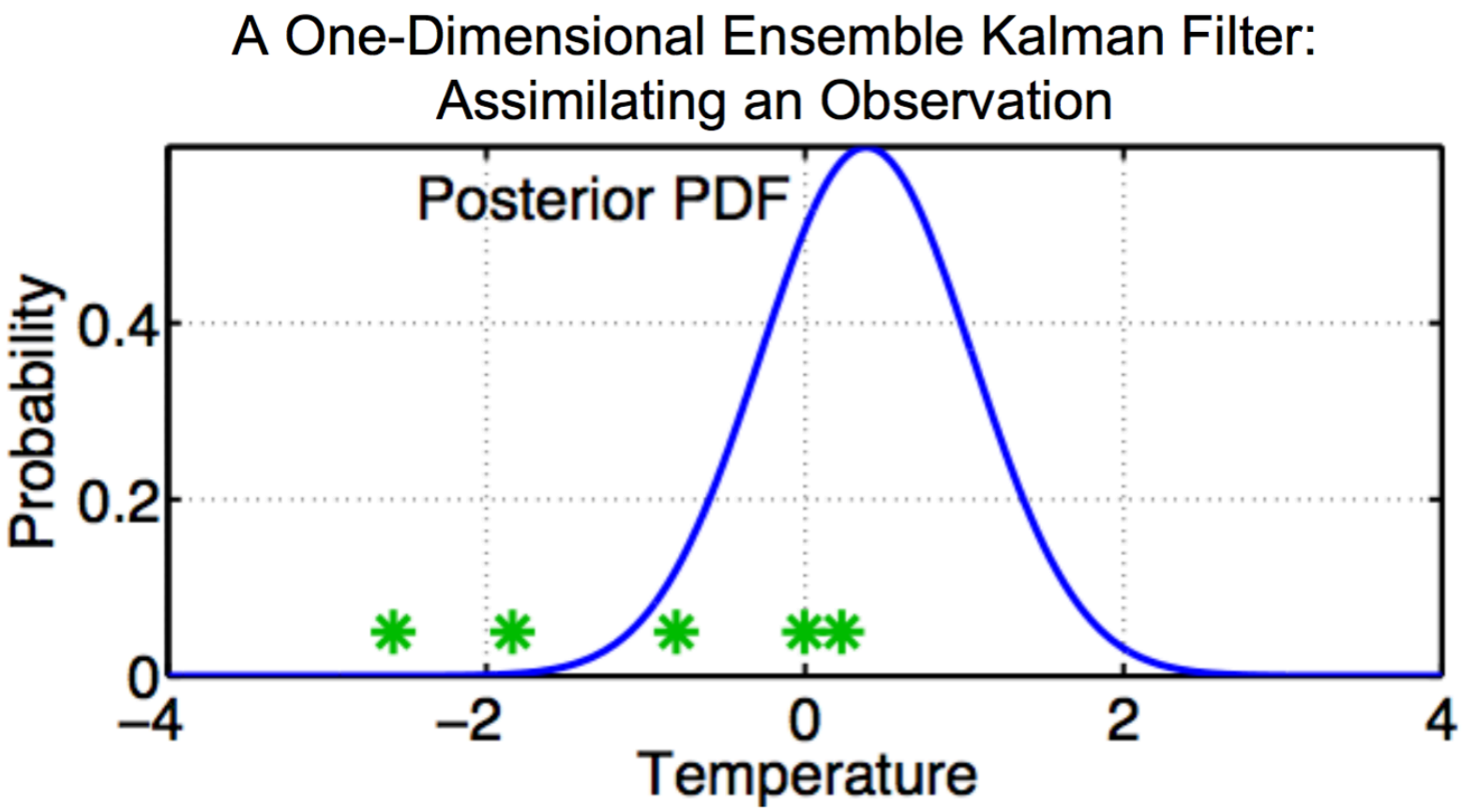
EnKF in action

Compute the continuous posterior pdf



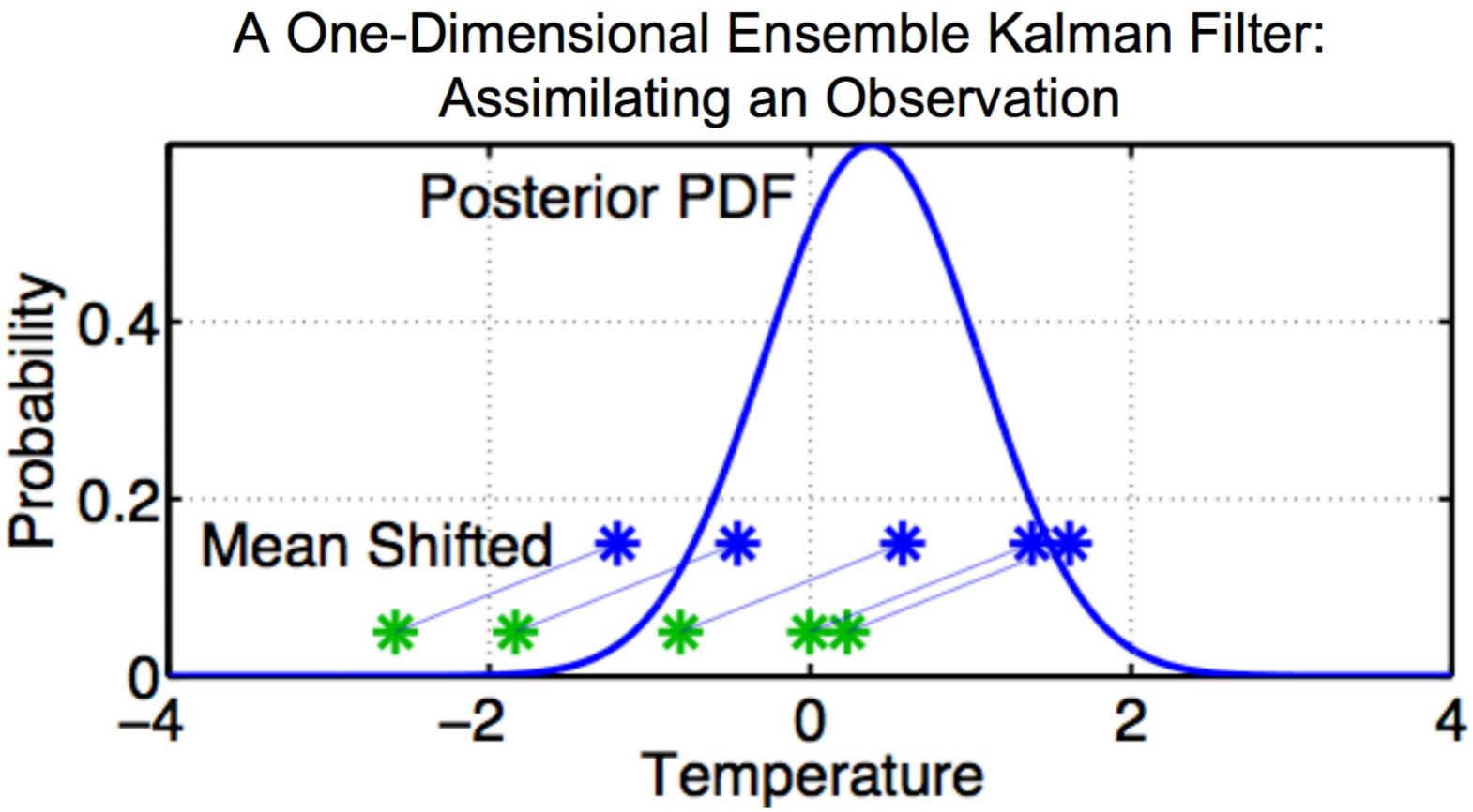
EnKF in action

Use a deterministic algorithm to adjust the ensemble



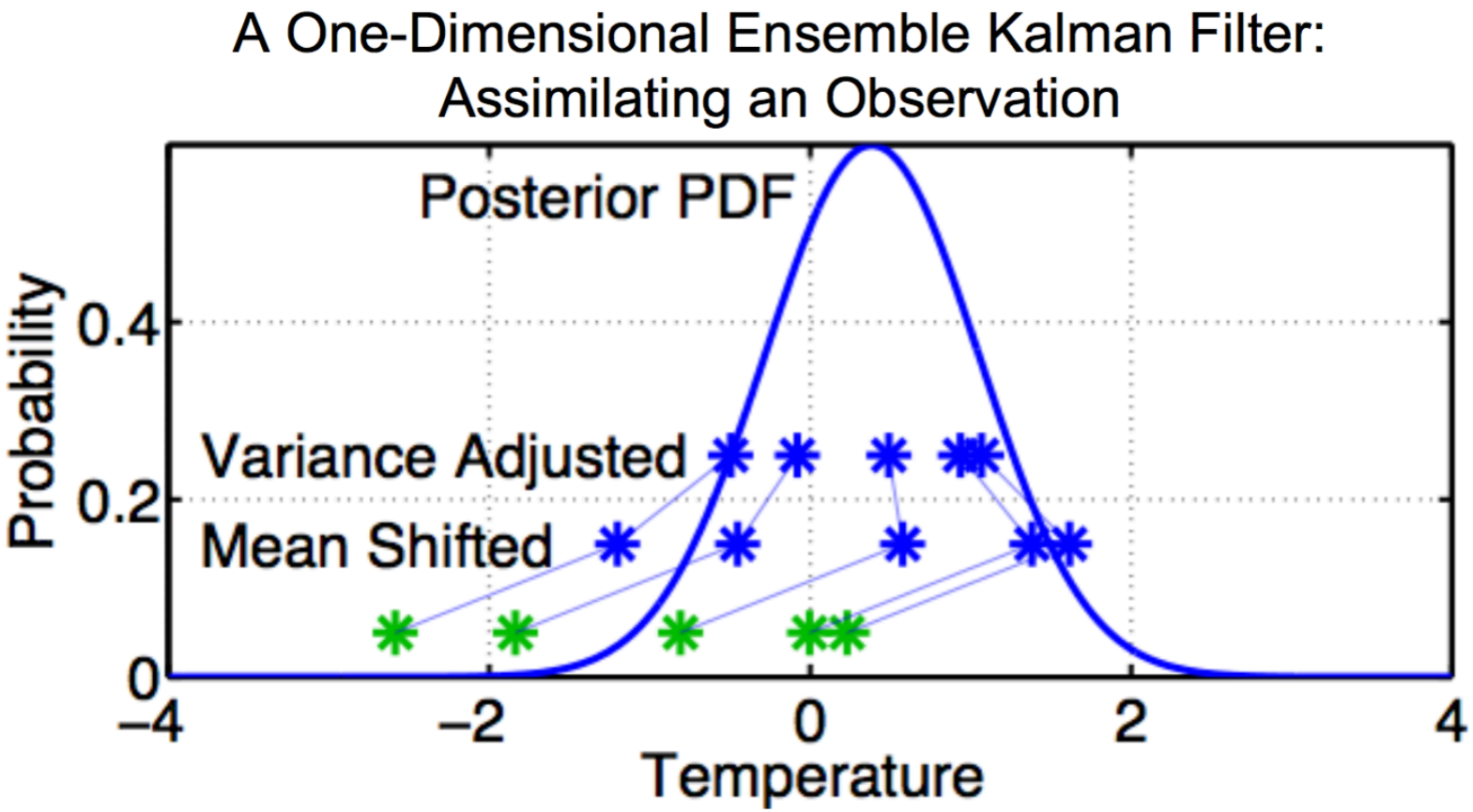
EnKF in action

First shift the ensemble to have the exact mean of posterior

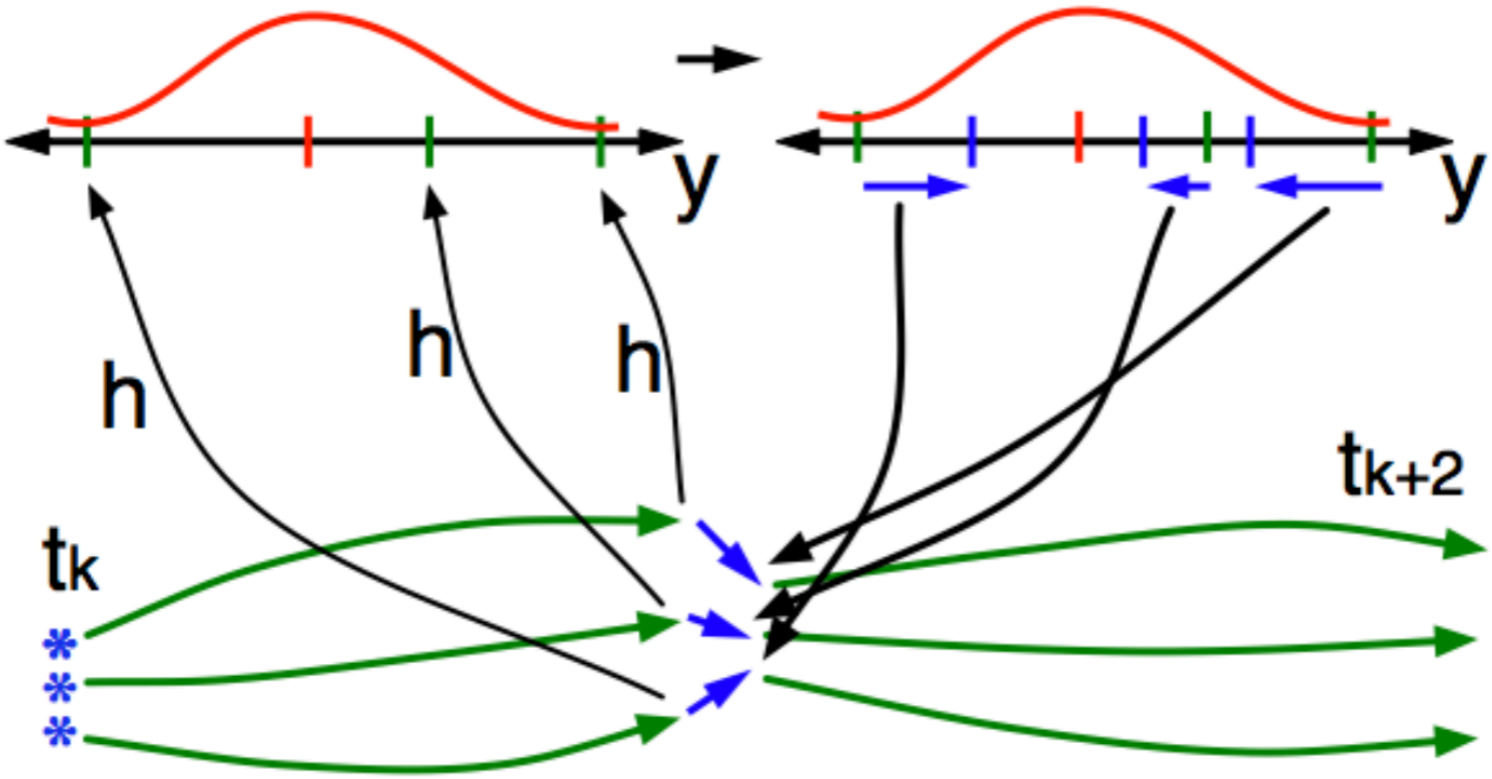


EnKF in action

Second linearly contract to have the exact variance of the posterior



EnKF in action

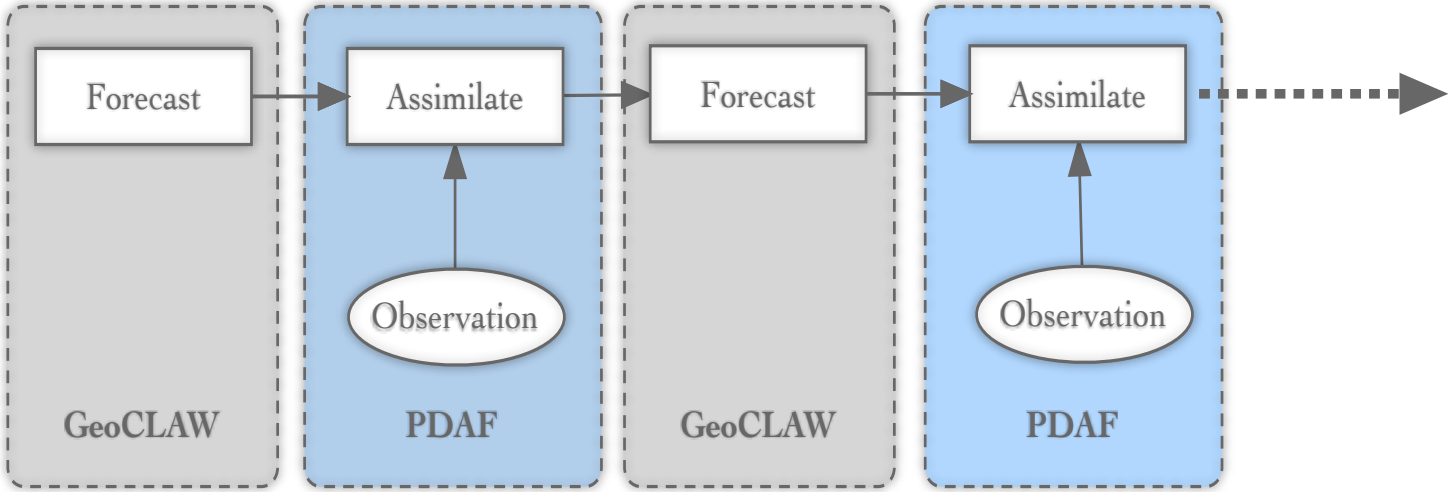


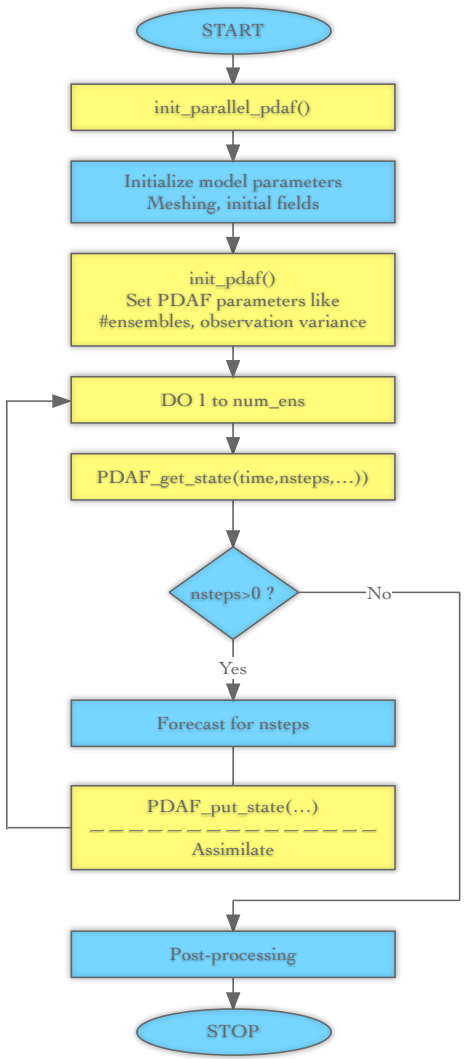
Difference between various ensemble filters is primarily in observation increment calculation

Implementation

- Objective is to implement sequential forecasting and analysis step with available tools and obtain the estimated analysis state
- Data assimilation is performed using the library **PDAF** - Parallel Data assimilation Framework developed at Computing Center of the Alfred Wegener Institute.
- The forecasting will be performed using **GeoCLAW** package -
  - Solves 2d depth averaged shallow water equations
  - Finite volume based
  - Adaptive mesh refinement in space and time
- The novelty in approach is the tight two way coupling between AMR and data assimilation

Implementation – PDAF + GeoCLAW





PDAF library is integrated with the GeoCLAW source code, known as '*flexible online implementation*'

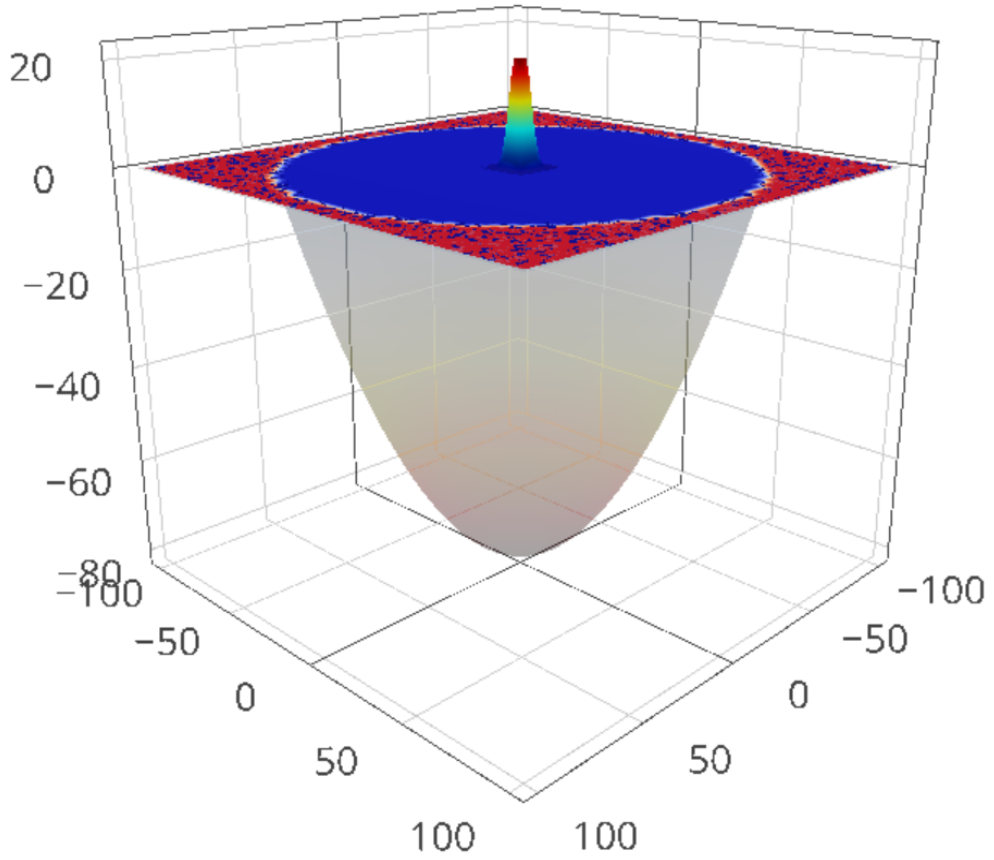
Test setup

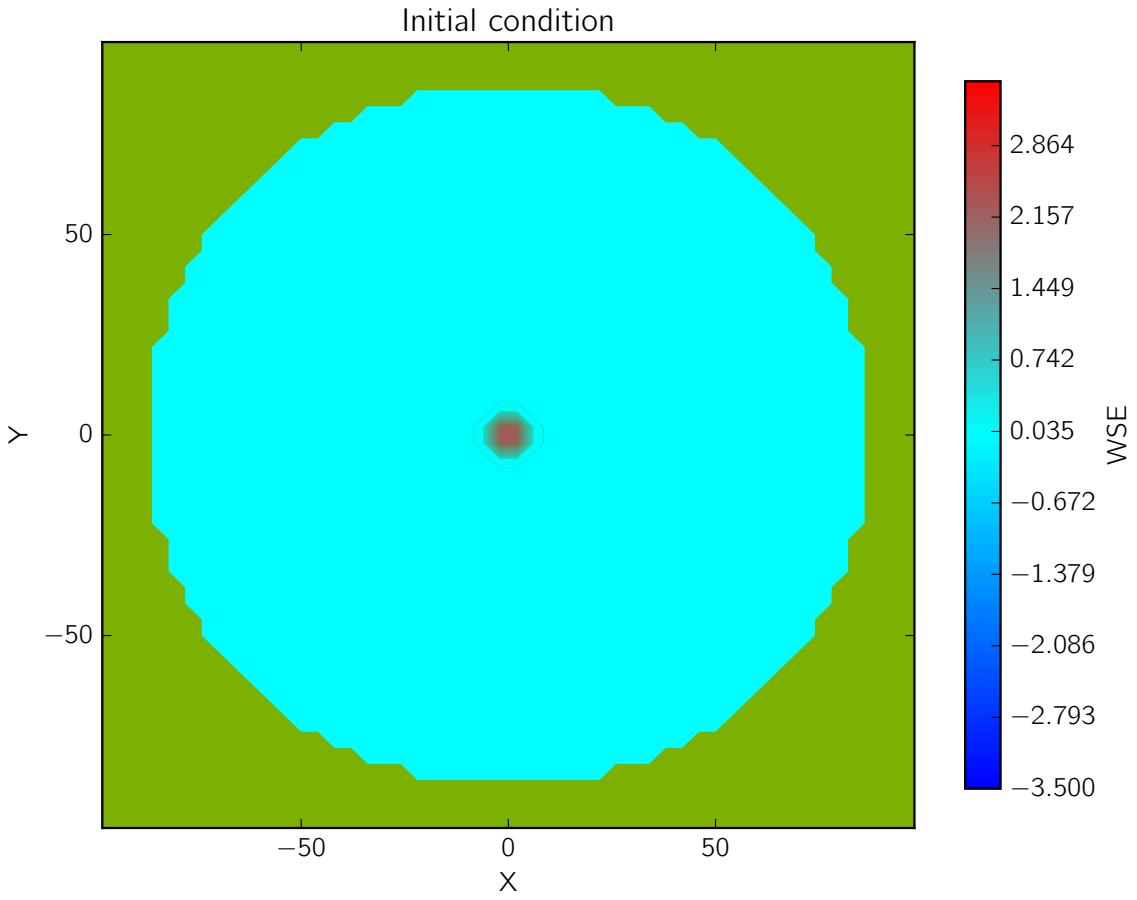
- The test is a verification case called radial bowl test case.
- Rectangular domain with $x \epsilon [-100.0, 100.0]$ and y $\epsilon [-100.0, 100.0]$ with 51×51 nodal points
- Time interval of $3.2s$ with a constant time step of $0.016s$.
- The initial condition is a Gaussian hump of water surface over a parabolic topography

Topography - $z(x, y) = 0.01(x^2 + y^2) - 80.0$

Initial condition - $\begin{cases} 5.0 \exp\left(-\frac{x^2+y^2}{10}\right) & , x^2 + y^2 < 100 \\ 0 & , otherwise \end{cases}$

Test setup – Bathymetry and Initial condition

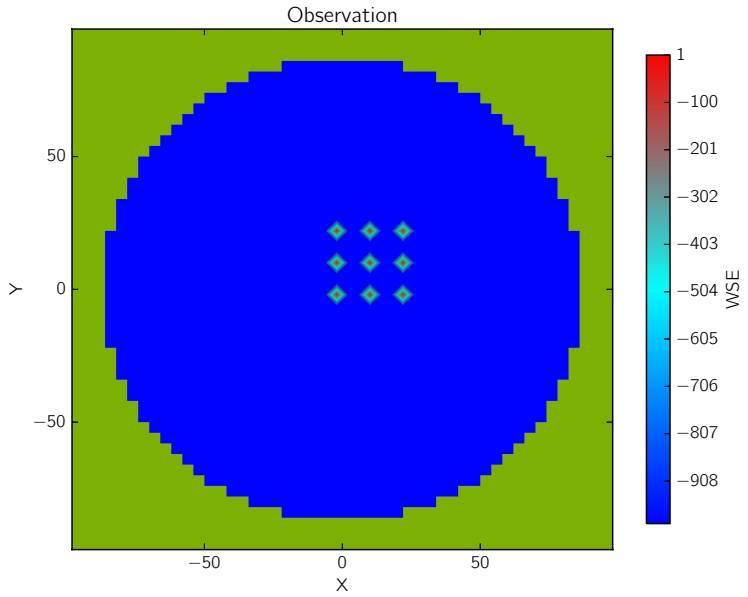




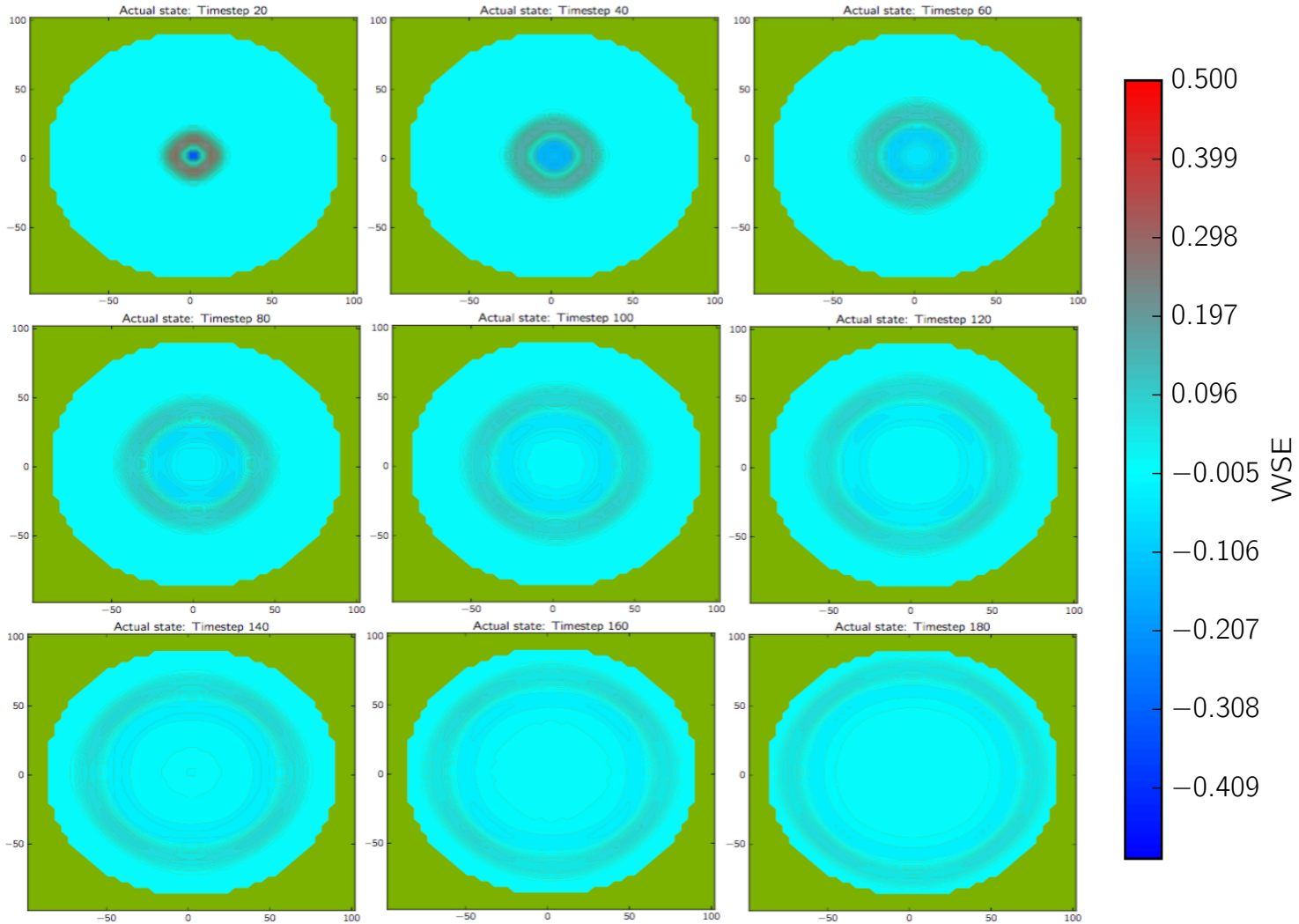
Test setup

- With the given configuration, we will perform two types of GeoCLAW executions –
  - Original GeoCLAW run (without assimilation). This will be the benchmark and the output of this run will be the source of observations as well.
  - GeoCLAW-PDAF run (with assimilation). An initial ensemble of initial condition will be provided and the observations recorded from the benchmark case will be assimilated.
- The observations are assimilated at every 20 timesteps.
- Comparison of GeoCLAW-PDAF run with Original GeoCLAW will give insight into the process of assimilation

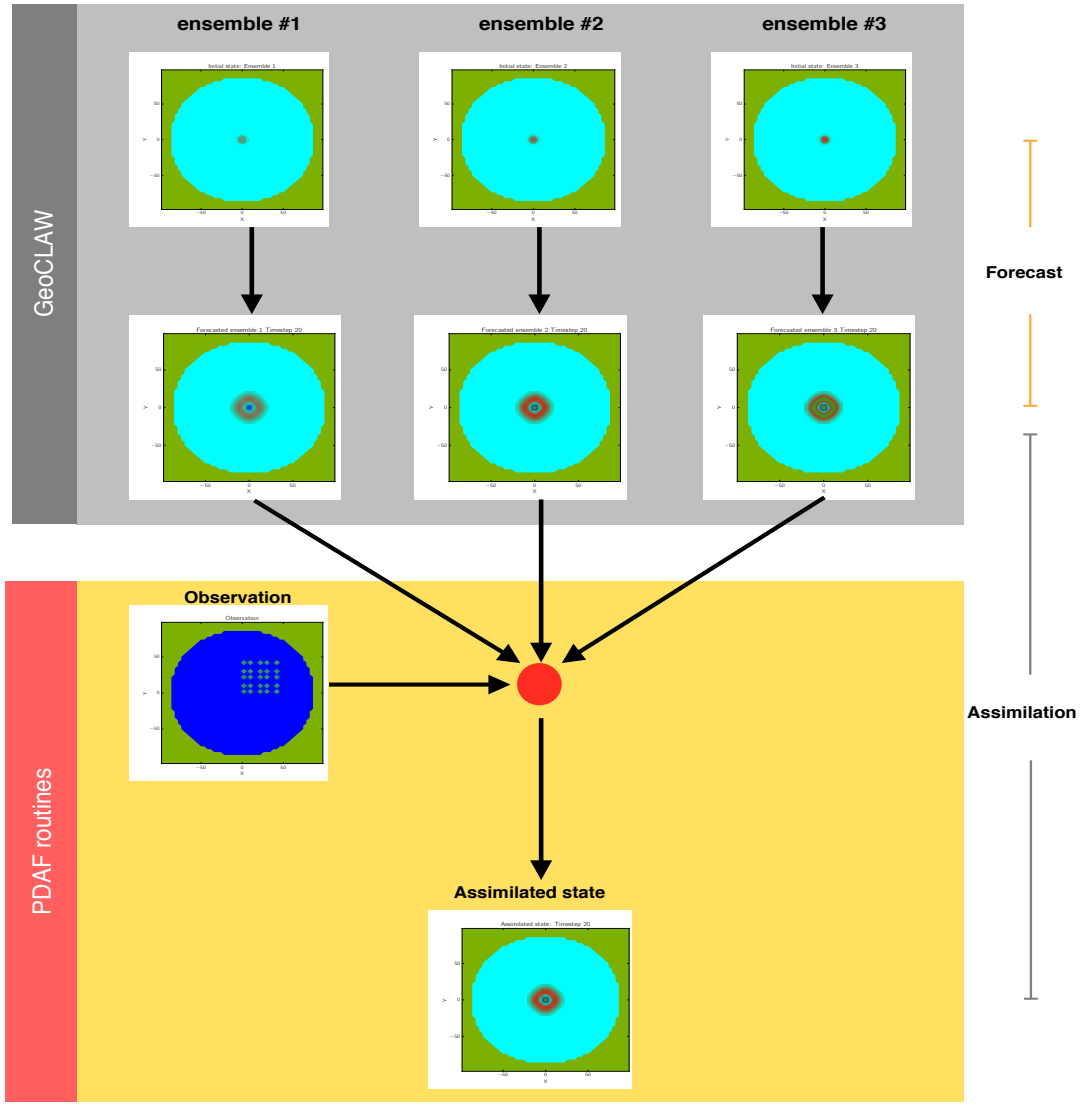
A typical observation data -



GeoCLAW without assimilation



Schematic representation



Testing methodology

- To compare the correct execution of assimilation, error analysis has been performed by calculating the L_2 error of water surface elevation between the benchmark case and the assimilated case.
- For preliminary testing, following tests are performed –
  1. Varying ensemble number with $rms\_obs = 0.01$
  2. Varying ensemble number with $rms\_obs = 1.0 \times 10^{-7}$ with no constraint on initial ensemble range
  3. Varying ensemble number with $rms\_obs = 1.0 \times 10^{-7}$ with fixed initial ensemble range
  4. Fixed ensemble number with varying observation points

Varying ensemble size with $rms\_obs = 0.01$

Table: Maximum amplitude for varying ensemble members using algorithm wherein amplitude increases with ensemble number

Ens	Maximum amplitude											
5	2.70	3.15	2.25	4.04	4.49							
9	2.70	3.15	3.59	4.04	2.25	4.94	5.39	5.84	6.29			
12	2.70	3.15	3.59	4.04	4.49	4.94	2.25	5.84	6.29	6.74	7.19	7.64

- The error norms for various ensemble members at different time steps, is $\mathcal{O}(10^{-2})$ which is not satisfactory as the water surface elevation in the domain are in $\mathcal{O}(10^{-5})$
- This is the effect of high $rms\_obs = 0.01$

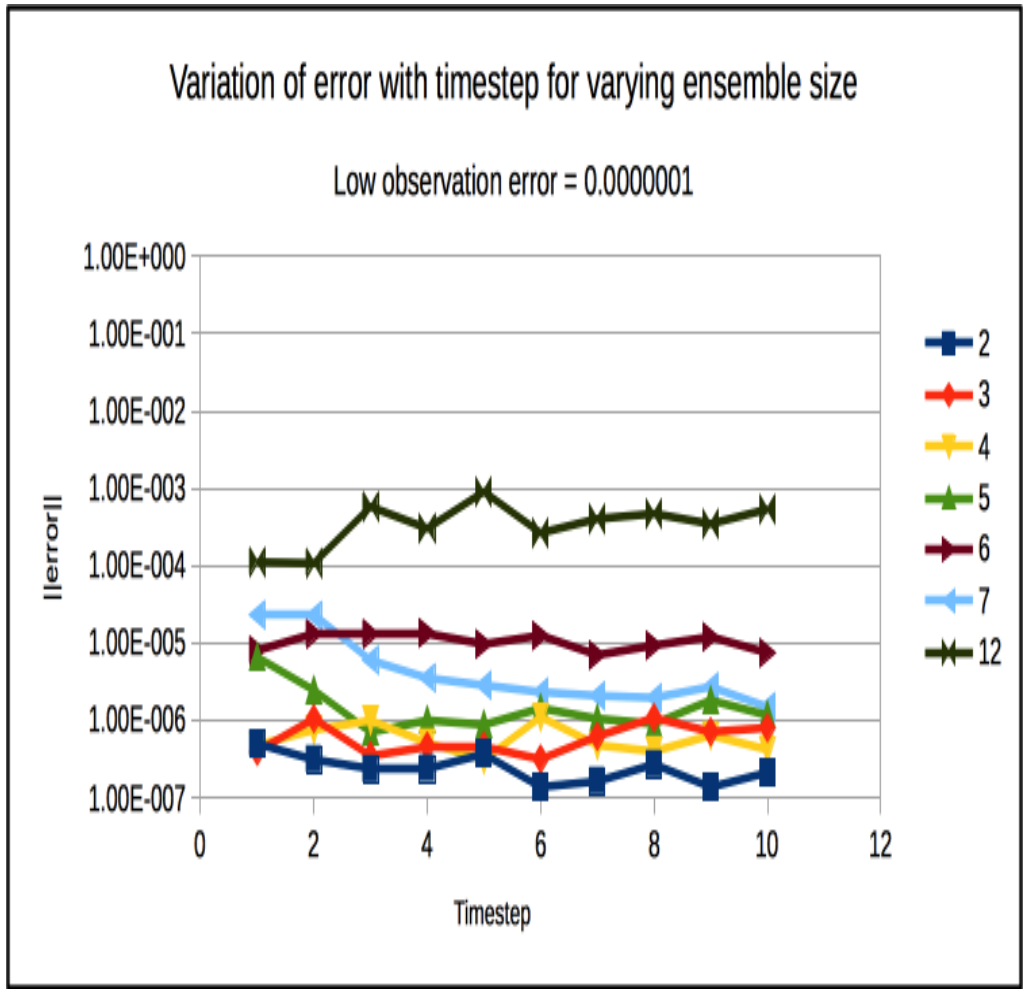
Table: L_2 error between assimilated state and benchmark for $rms\_obs = 0.01$

Timestep	Ensemble size			
	5	9	12	15
1	2.13E-02	2.31E-02	2.12E-02	2.70E-02
2	2.33E-02	2.66E-02	3.77E-02	3.89E-02
3	9.34E-03	2.58E-02	1.85E-02	4.12E-02
4	3.08E-02	2.68E-02	1.79E-02	1.99E-02
5	4.82E-02	2.62E-02	1.06E-02	2.83E-02
6	2.15E-02	3.39E-02	3.07E-02	1.31E-02
7	3.60E-02	2.13E-02	1.58E-02	2.09E-02
8	3.77E-02	3.39E-02	3.08E-02	4.71E-02
9	3.33E-02	2.61E-02	3.34E-02	3.33E-02
10	3.45E-02	1.73E-02	2.96E-02	3.18E-02

Varying ensemble size with $rms\_obs = 1.0\times10^{-7}$

Table: L_2 error between assimilated state and benchmark for $rms\_obs = 0.01$

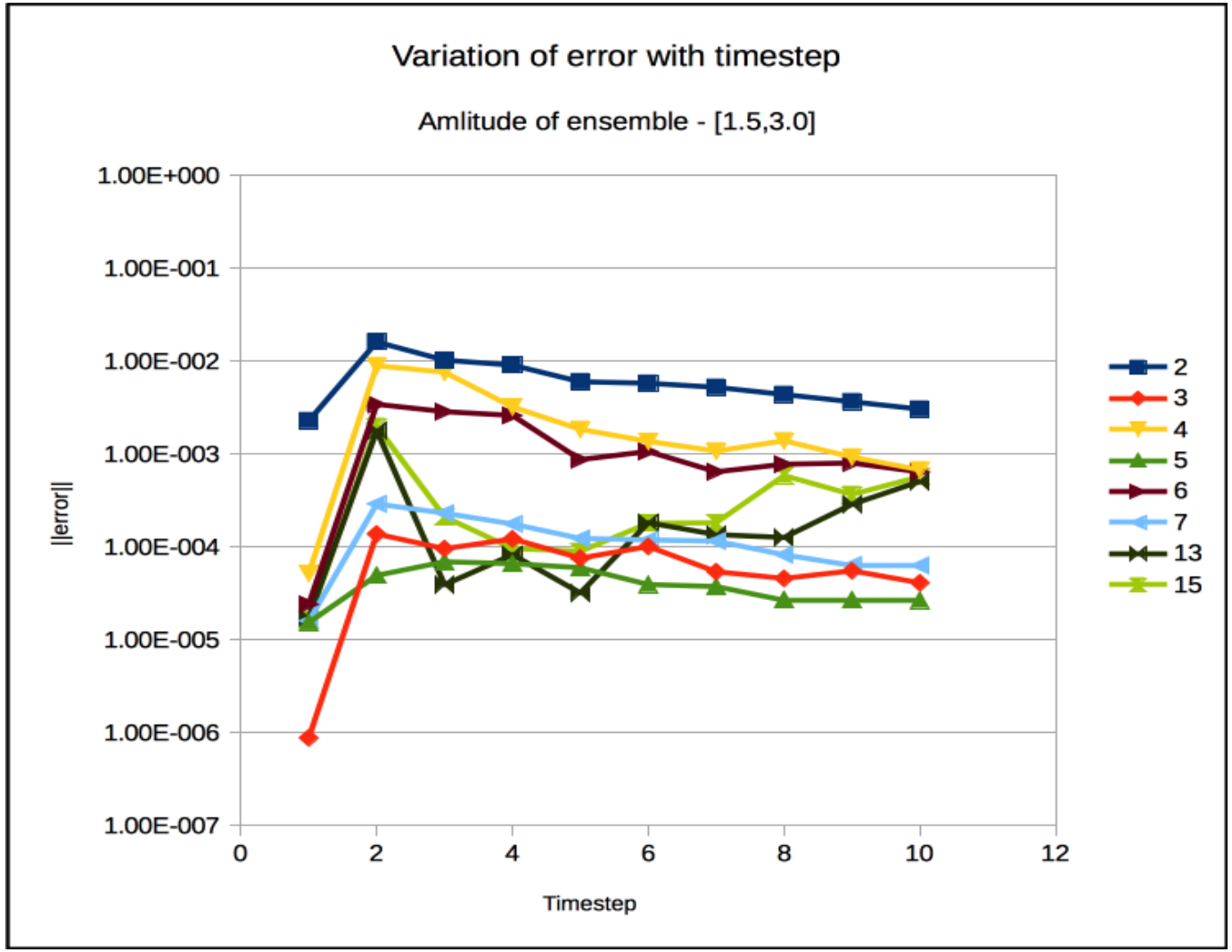
Timestep	Ensemble size						
	2	3	4	5	6	7	12
1	4.97E-07	3.97E-07	4.75E-07	6.55E-06	8.00E-06	2.31E-05	1.11E-04
2	3.10E-07	1.05E-06	7.52E-07	2.45E-06	1.30E-05	2.26E-05	1.07E-04
3	2.32E-07	3.48E-07	9.96E-07	7.04E-07	1.31E-05	5.98E-06	5.79E-04
4	2.36E-07	4.62E-07	5.13E-07	9.99E-07	1.35E-05	3.52E-06	3.04E-04
5	3.66E-07	4.64E-07	3.11E-07	8.78E-07	9.70E-06	2.85E-06	9.06E-04
6	1.37E-07	3.14E-07	1.11E-06	1.43E-06	1.25E-05	2.32E-06	2.65E-04
7	1.63E-07	6.20E-07	4.75E-07	1.05E-06	6.99E-06	2.08E-06	4.03E-04
8	2.69E-07	1.08E-06	3.97E-07	9.03E-07	9.33E-06	1.96E-06	4.71E-04
9	1.37E-07	7.10E-07	6.38E-07	1.81E-06	1.20E-05	2.71E-06	3.48E-04
10	2.11E-07	7.99E-07	4.19E-07	1.17E-06	7.46E-06	1.49E-06	5.36E-04



- On decreasing the value of rms_obs , the error norms are substantially reduced
- Note that the amplitude of the ensemble members increases with ensemble size

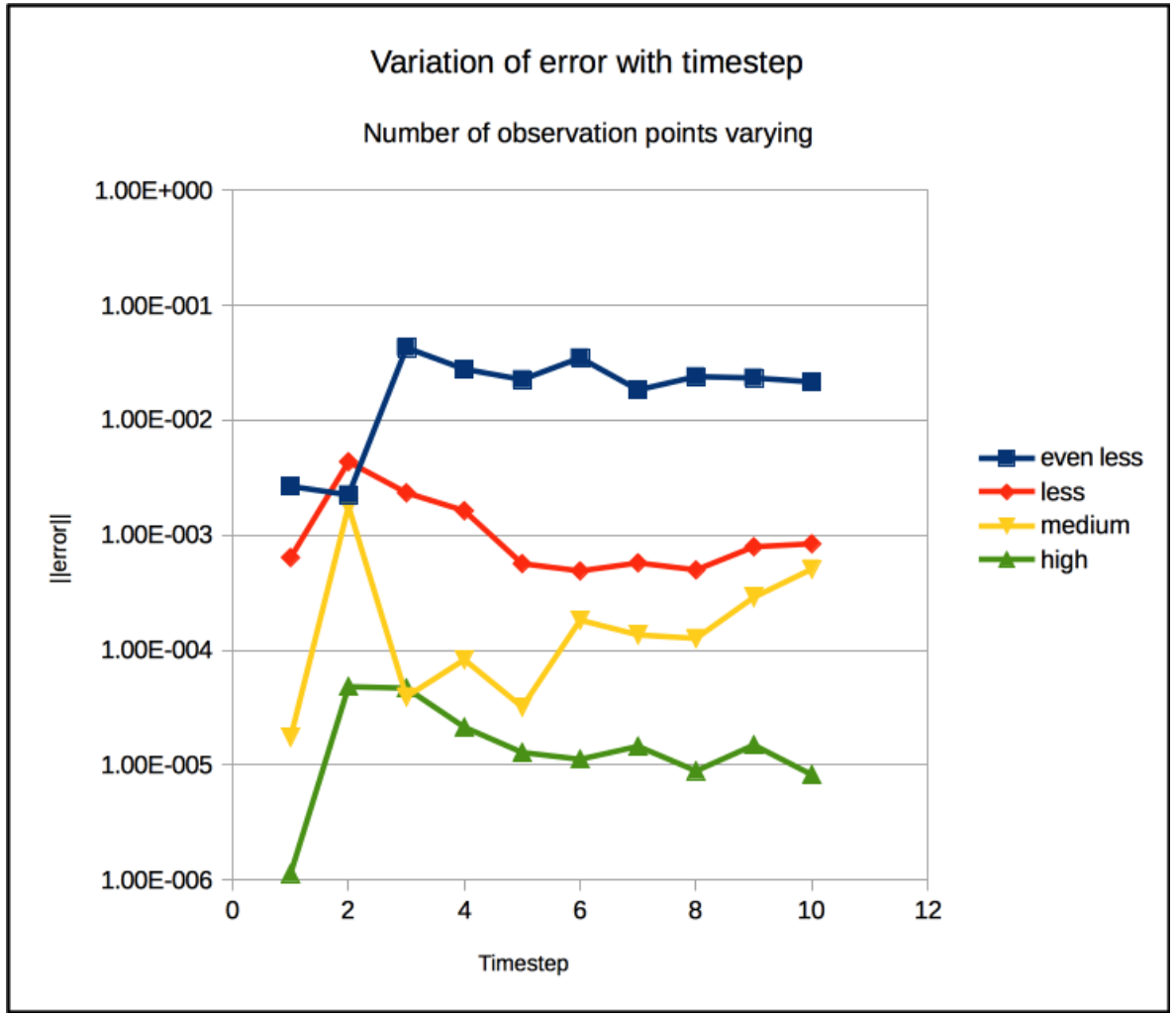
Varying ensemble size with $rms\_obs = 1.0\times10^{-7}$

- The amplitude for ensembles is in range $[1.5,3.0]$
- Assimilation using larger ensemble members tend to have lower error than lesser ensemble number. However, there is no definite pattern.



Fixed ensemble number with varying observations

- Assimilation is tested by running the test configuration with fixed ensemble members = 13, but with varying number of observation points, in the order of 1, 9, 25, 36 points in the domain.
- more observation points assimilated accounts for lesser errors in water surface elevation.



Topics to be explored

- Under-sampling
- Localization
- What is a good number of ensemble?
- Filter divergence
- Adaptive mesh affecting data assimilation?

Conclusion

- The technology will be tested using actual data from recent events, and implemented on high-performance computational platforms.
- These advances offer the promise of significantly transforming data-driven, real-time modeling of hydrological hazards, with potentially broader applications in other science domains.

Collaborators

- Prof. Clint Dawson, The University of Texas at Austin
- Kyle Mandli, Columbia University

- **Acknowledgement:** This work is supported by King Abdullah University of Science & Technology (KAUST)

Elementary statistics

Minimum variance approach

The estimated temperature is a linear combination of T_b and T_o -

$$T_a = \alpha T_o + \beta T_b + \gamma$$

With error of the estimate –

$$\epsilon_a = T_a - T^*$$

$$T_a = \epsilon_a + T^* = \alpha(\epsilon_o + T^*) + \beta(\epsilon_b + T^*) + \gamma$$

Taking the mean and rearranging gives,

$$\overline{\epsilon_a} = (\alpha + \beta - 1)T^* + \gamma$$

- $\alpha + \beta = 1$
- $\gamma = 0$

$$\Longrightarrow T_a = \alpha T_o + (1-\alpha)T_b \text{ and } \epsilon_a = \alpha\epsilon_o + (1-\alpha)\epsilon_b$$

Optimality criteria - Minimization of variance of the estimate $(\overline{\epsilon_a^2})$ gives

$$\frac{d\overline{\epsilon_a^2}}{d\alpha} = \frac{d}{d\alpha}\left(\overline{\epsilon_o^2} + 2\alpha(1-\alpha)\overline{\epsilon_o\epsilon_b} + (1-\alpha^2)\epsilon_b^2\right) = 0$$

$$\Longrightarrow \alpha = \frac{\epsilon_b^2}{\epsilon_b^2 + \epsilon_o^2} \Longrightarrow T_a = T_b + \alpha(T_a - T_b)$$

Bayesian approach

$$P(T/T_o \cap T_b) \propto P(T_o/T \cap T_b)P(T/T_b)$$

Prior $\propto e^{-\frac{1}{2}\left(\frac{T-T_b}{\epsilon_b}\right)^2}$

Likelihood $\propto e^{-\frac{1}{2}\left(\frac{T-T_o}{\epsilon_o}\right)^2}$

$$\Longrightarrow \text{Posterior} \propto e^{-\frac{1}{2}(\mathcal{L}(T))}$$

$$\mathcal{L}(T) = \left(\frac{T-T_b}{\epsilon_b}\right)^2 + \left(\frac{T-T_o}{\epsilon_o}\right)^2$$

$$\frac{d\mathcal{L}}{dT} = 0$$