Ensemble Kalman filter and related filters

Particle representation of model

$$\boldsymbol{x}^1, \ldots, \boldsymbol{x}^B \sim p(\boldsymbol{x})$$

Samples (independent) from prior model. Samples are equally weighted $w^b = 1/B$, $b = 1, \ldots, B$.
In many application the goal is to update this sample representation to an approximate posterior sample.

$$\boldsymbol{x}^1, \ldots, \boldsymbol{x}^B \sim p(\boldsymbol{x}|\boldsymbol{y})$$

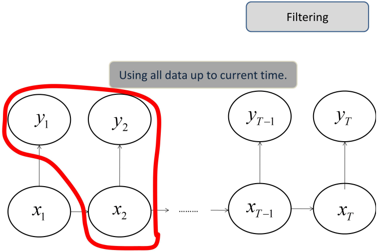
Samples can have non-equal weights w^b, $\sum_{b=1}^B w^b = 1$, or equal weights $1/B$.
For some methods the approximation converges to samples from the true posterior, under some regularity conditions.

Sequential Bayesian data assimilation

Methods like the particle filter or ensemble Kalman filtering methods have been very useful in **data assimilation** problems.

- Particle filter - introduced in 1990s for target tracking in real time and robotic applications.
- Ensemble Kalman filter - introduced in the 1990s for oceanography or meteorological applications.

Sequential Bayesian assimilation



$$p(\boldsymbol{x}_1), \quad p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}), \quad p(\boldsymbol{y}_t|\boldsymbol{x}_t), \; t = 2, 3, \ldots, T.$$

Dynamic model

Process model is described by:

$$p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \ldots, \boldsymbol{x}_1) = p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}),$$

This could be a differential equation, or it could be a simple linear process, or even a static process ($\boldsymbol{x}_t = \boldsymbol{x}_{t-1}$).

The data gathering process is described via the likelihood:

$$p(\boldsymbol{y}_t|\boldsymbol{x}_t, \ldots, \boldsymbol{x}_1, \boldsymbol{y}_{t-1}, \ldots, \boldsymbol{y}_1) = p(\boldsymbol{y}_t|\boldsymbol{x}_t)$$

This could also be nonlinear, or it could represent picking a subset of variables (with noise).

General formula

Filtering, solution:

$$
\begin{aligned}
p(\boldsymbol{x}_t|\boldsymbol{y}_1,\ldots,\boldsymbol{y}_{t-1}) &= \int p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})p(\boldsymbol{x}_{t-1}|\boldsymbol{y}_1,\ldots,\boldsymbol{y}_{t-1})d\boldsymbol{x}_{t-1}. \\
p(\boldsymbol{x}_t|\boldsymbol{y}_1,\ldots,\boldsymbol{y}_t) &= \frac{p(\boldsymbol{x}_t|\boldsymbol{y}_1,\ldots,,\boldsymbol{y}_{t-1})p(\boldsymbol{y}_t|\boldsymbol{x}_t)}{p(\boldsymbol{y}_t|\boldsymbol{y}_1,\ldots,\boldsymbol{y}_{t-1})}
\end{aligned}
$$

Note Markov assumption in process, and conditionally independent data.

Ensemble Kalman filter (EnKF)

- Monte Carlo based data assimilation
- Assimilation based on **linear** update

(Evensen, 1994, Evensen, 2009).

EnKF

- Initial: Independent prior samples $\boldsymbol{x}_0^{b,a} \sim p(\boldsymbol{x}_0)$, $b = 1, \ldots, B$.
- Iterate for samples $b = 1, \ldots, B$ and time steps $t = 1, \ldots, T$:
  Forecast variables (could be non-linear, black-box solver):
  $$\boldsymbol{x}_t^{b,f} = \boldsymbol{g}(\boldsymbol{x}_{t-1}^{b,a}; \boldsymbol{\epsilon}_t^b),$$
  Forecast data (could be non-linear, black-box solver):
  $$\boldsymbol{y}_t^b = \boldsymbol{h}(\boldsymbol{x}_t^{b,f}; \boldsymbol{\delta}_t^b),$$
  Assimilate:
  $$\boldsymbol{x}_t^{b,a} = \boldsymbol{x}_t^{b,f} + \hat{\boldsymbol{K}}_t(\boldsymbol{y}_t - \boldsymbol{y}_t^b).$$
  $$\hat{\boldsymbol{K}}_t = \hat{\boldsymbol{\Sigma}}_{xy,t}\hat{\boldsymbol{\Sigma}}_{y,t}^{-1}.$$

EnKF moves ensemble members

- No weights in the EnKF.
- All ensemble-members are shifted in the update.
- Ensemble-members are moved closer towards data.
- The move is linear and goes in along the same projection for all ensemble-members.

EnKF update

$$\boldsymbol{x}_t^{b,a} = \boldsymbol{x}_t^{b,f} + \hat{\boldsymbol{\Sigma}}_{xy,t}\hat{\boldsymbol{\Sigma}}_{y,t}^{-1}(\boldsymbol{y}_t - \boldsymbol{y}_t^b).$$

This is a regression problem.

$$\hat{\boldsymbol{\Sigma}}_{y,t} = \frac{1}{B}\sum_{b=1}^{B}(\boldsymbol{y}_t^b - \bar{\boldsymbol{y}}_t)(\boldsymbol{y}_t^b - \bar{\boldsymbol{y}}_t)', \quad \bar{\boldsymbol{y}}_t = \frac{1}{B}\sum_{b=1}^{B}\boldsymbol{y}_t^b$$

$$\hat{\boldsymbol{\Sigma}}_{xy,t} = \frac{1}{B}\sum_{b=1}^{B}(\boldsymbol{x}_t^{b,f} - \bar{\boldsymbol{x}}_t)(\boldsymbol{y}_t^b - \bar{\boldsymbol{y}}_t)', \quad \bar{\boldsymbol{x}}_t = \frac{1}{B}\sum_{b=1}^{B}\boldsymbol{x}_t^{b,f},$$

Matrix equivalent EnKF updates

- The linear update has several equivalent forms.
- From Sherman-Woodbury-Morrison formula it can be done in data domain.
- It can be regarded as a transform matrix from forecast to updated anomalies around mean.

There are also iterative EnKF versions solving a near-quadratic optimization problem. (We show only the standard solution to the quadratic problem.)

EnKF update in a univariate example

$$\boldsymbol{x}_t^{b,a} = \boldsymbol{x}_t^{b,f} + \hat{\boldsymbol{\Sigma}}_{xy,t}\hat{\boldsymbol{\Sigma}}_{y,t}^{-1}(\boldsymbol{y}_t - \boldsymbol{y}_t^b).$$

This is a regression problem, where the link between data mismatch and analysis-forecast in ensemble is given by $\beta = \hat{\boldsymbol{\Sigma}}_{xy,t}\hat{\boldsymbol{\Sigma}}_{y,t}^{-1}$

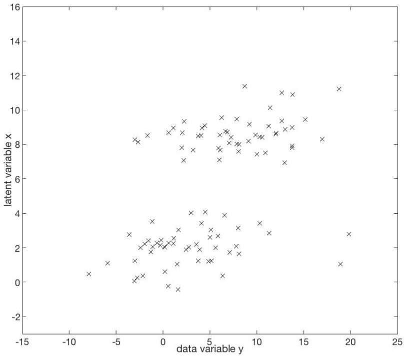
$$\boldsymbol{x}_t^{b,a} - \boldsymbol{x}_t^{b,f} = \beta(\boldsymbol{y}_t - \boldsymbol{y}_t^b).$$

$$\hat{\boldsymbol{\Sigma}}_{y,t} = \frac{1}{B}\sum_{b=1}^{B}(\boldsymbol{y}_t^b - \bar{\boldsymbol{y}}_t)(\boldsymbol{y}_t^b - \bar{\boldsymbol{y}}_t)', \quad \bar{\boldsymbol{y}}_t = \frac{1}{B}\sum_{b=1}^{B}\boldsymbol{y}_t^b$$

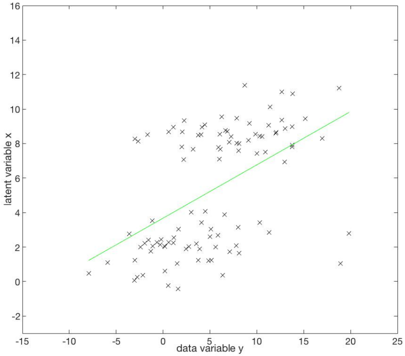
$$\hat{\boldsymbol{\Sigma}}_{xy,t} = \frac{1}{B}\sum_{b=1}^{B}(\boldsymbol{x}_t^{b,f} - \bar{\boldsymbol{x}}_t)(\boldsymbol{y}_t^b - \bar{\boldsymbol{y}}_t)', \quad \bar{\boldsymbol{x}}_t = \frac{1}{B}\sum_{b=1}^{B}\boldsymbol{x}_t^{b,f},$$

Univariate example - samples

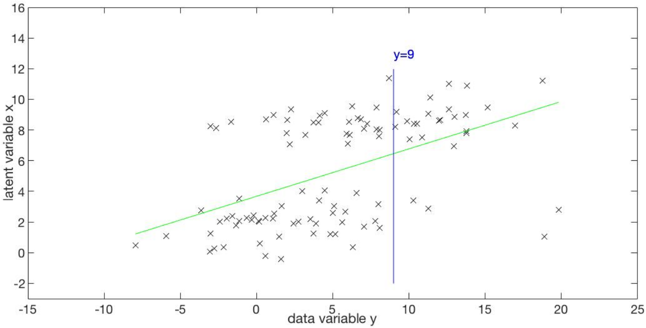
$$x^b \sim p(x), \qquad y^b = x^b + N(0, 5^2)$$



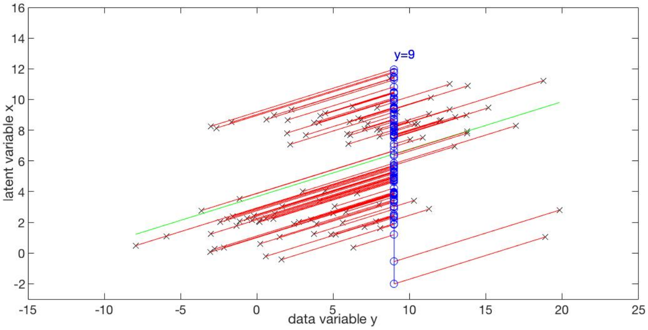
Univariate example - regression fit



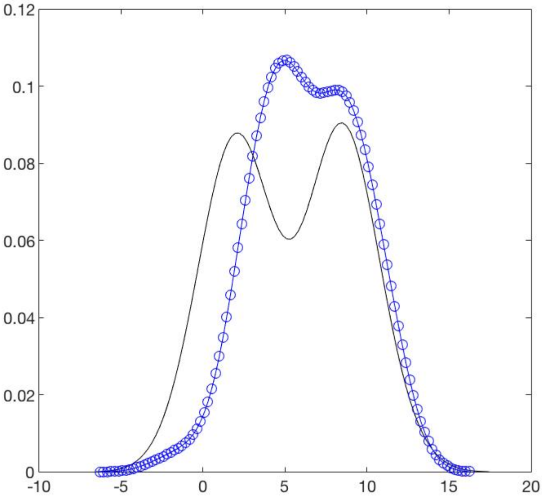
Univariate example - observation



Univariate example - analysis or update step



Univariate example - prior and posterior



EnKF, Gauss-linear likelihood

- Initial: Independent samples $\boldsymbol{x}_0^{b,a} \sim p(\boldsymbol{x}_0)$, $b = 1, \dots, B$.
- Iterate for samples $b = 1, \dots, B$ and time steps $t = 1, \dots, T$:
  Forecast variables (could be non-linear, black-box solver):

  $$\boldsymbol{x}_t^{b,f} = \boldsymbol{g}(\boldsymbol{x}_{t-1}^{b,a}; \boldsymbol{\epsilon}_t^b),$$

  Linear likelihood:

  $$\boldsymbol{y}_t^b = \boldsymbol{H}\boldsymbol{x}_t^{b,f} + N(0, \boldsymbol{T})$$

  Assimilate :

  $$\boldsymbol{x}_t^{b,a} = \boldsymbol{x}_t^{b,f} + \hat{\boldsymbol{\Sigma}}_t \boldsymbol{H}' (\boldsymbol{H}\hat{\boldsymbol{\Sigma}}_t \boldsymbol{H}' + \boldsymbol{T})^{-1} (\boldsymbol{y}_t - \boldsymbol{y}_t^b).$$

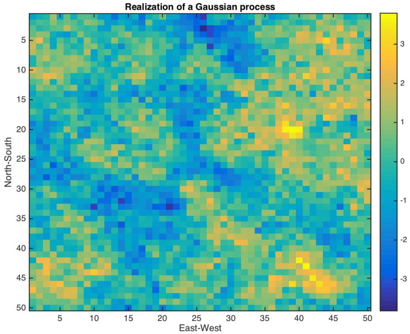
Estimation of $\boldsymbol{\Sigma}_t$

Standard approach:

$$\hat{\boldsymbol{\Sigma}}_t = \frac{1}{B}\sum_{b=1}^{B}(\boldsymbol{x}_t^{b,f} - \bar{\boldsymbol{x}}_t)(\boldsymbol{x}_t^{b,f} - \bar{\boldsymbol{x}}_t)', \qquad \bar{\boldsymbol{x}}_t = \frac{1}{B}\sum_{b=1}^{B}\boldsymbol{x}_t^{b,f}$$

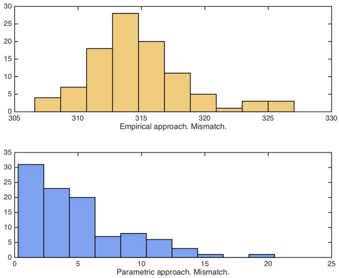
Gives less Monte Carlo error than straightforward estimator for Kalman gain.

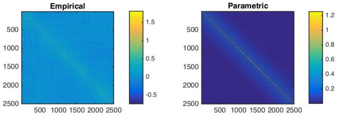
Gaussian random field



Use $B = 100$ ensembles (realizations) to estimate the covariance matrix.

Estimation of covariance matrix





There is lots of Monte Carlo error in the estimated covariance matrix and Kalman gain. Numerous tricks try to resolve this: inflation, localization, etc. Also, since the Kalman gain is estimated from data, there is coupling over many time steps which gives challenges. Resampling from a Gaussian approximation might solve this.

EnKF is an approximate filtering method

The EnKF is exact only under idealized conditions (linear, Gaussian assumptions and sample size B goes to infinity).
For non-linear systems or non-Gaussian systems there are no asymptotic results showing correctness. (It was hence criticized by statisticians earlier.)
But experience shows that it works well for complex systems, so it is used a lot.

Spatial AR(1) model

$$\boldsymbol{x}_0 \sim N(0, \boldsymbol{\Sigma}),$$

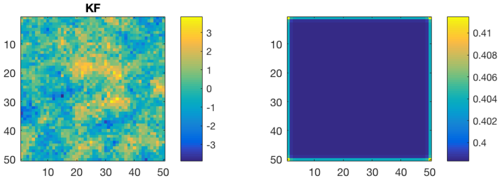
$$\boldsymbol{x}_t = \rho \boldsymbol{x}_{t-1} + N(0, (1-\rho^2)\boldsymbol{\Sigma}), \; t = 1, \ldots, T$$

$$\boldsymbol{y}_t = \boldsymbol{x}_t + N(0, \tau^2 \boldsymbol{I}), \; t = 1, \ldots, T$$

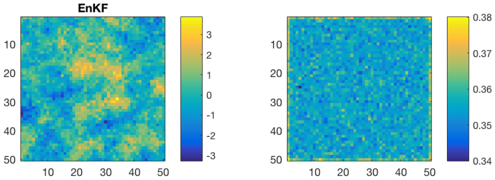
$T = 10$, 50×50 grid.

EnKF is run with $B = 5000$ ensemble members.

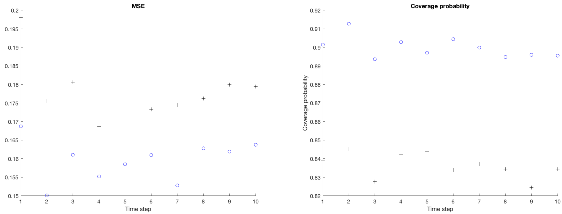
Spatial AR(1): KF pred



Spatial AR(1): EnKF pred

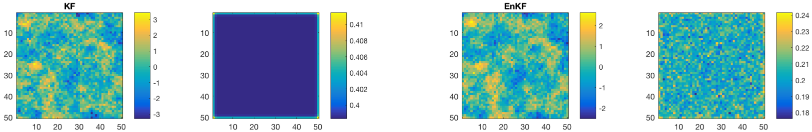


Spatial AR(1): MSE and coverage



KF (o), EnKF (+).

Spatial AR(1): EnKF pred ($B = 500$)



(Average coverage for EnKF is 0.38 for nominal 0.90. This is not uncommon for the EnKF to underestimate uncertainty.)

Problems with underestimation / ensemble collapse

- Localize the update. Only a subset of the variables are shifted. This is usually achieved by a tapering of the estimated Kalman gain or covariance matrices.
- Multiple data assimilation: Blows up variance of the update and assimilated data many times. $\tau \to \tau K$.
- Inflation of ensemble after update (scalar factor can enter in various ways (covariance or Kalman filter, or pull ensemble-members out from mean after update). It is often tuned to match some statistics.)
- Resampling from a Gaussian approximation at certain steps.

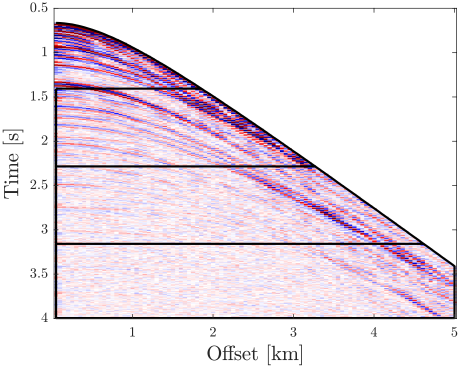
Seismic data example

The posterior model for the subsurface variables conditional on seismic data is computed by sequentially integrating data in blocks $k = 1, \ldots, K$. From Bayes' rule:

$$p(\boldsymbol{x}|\boldsymbol{y}_1, \ldots, \boldsymbol{y}_k) \propto p(\boldsymbol{y}_k|\boldsymbol{x})p(\boldsymbol{x}|\boldsymbol{y}_1, \ldots, \boldsymbol{y}_{k-1}), \tag{1}$$

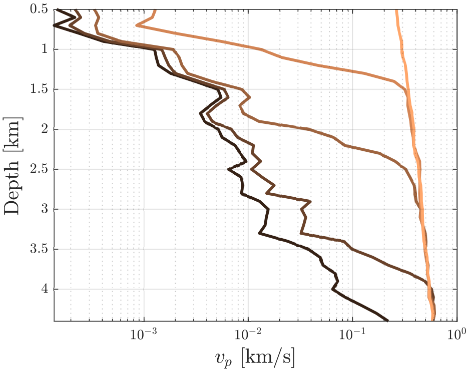
At step $k = K$, the ensemble represents the posterior distribution.

Seismic data and blocks



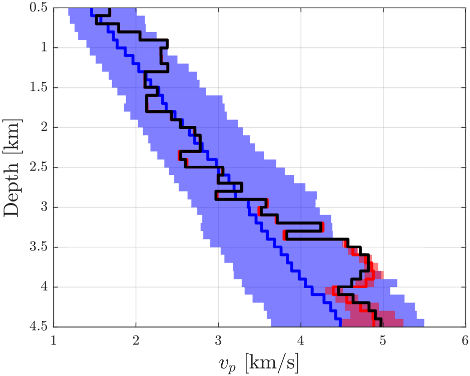
In this case there is a well log near the location of the seismic data. Ensemble-based solution is compared with the 'truth'.

Standard deviation over block assimilation



Standard deviation is naturally reduced top-down, as data in blocks are assimilated.

Data assimilation results

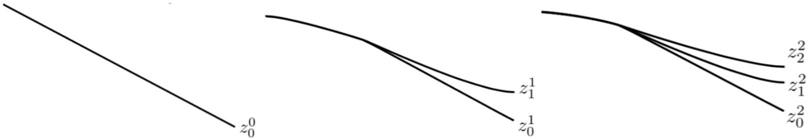


Marginal ensemble-based 80 % intervals. Prior ensemble blue. Posterior in red.

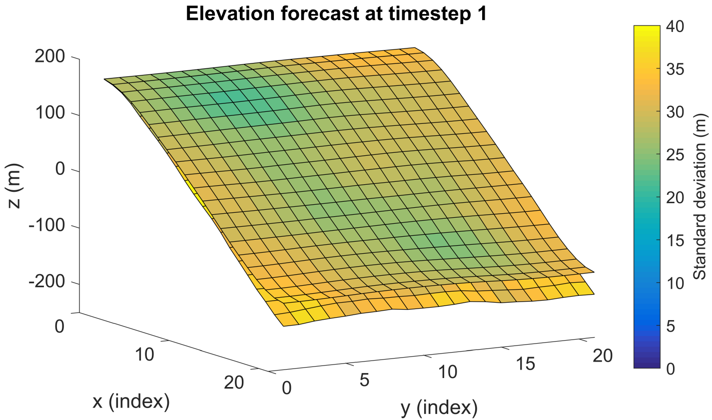
Geologic process models

Differential equation for sedimentation, corrected with data.

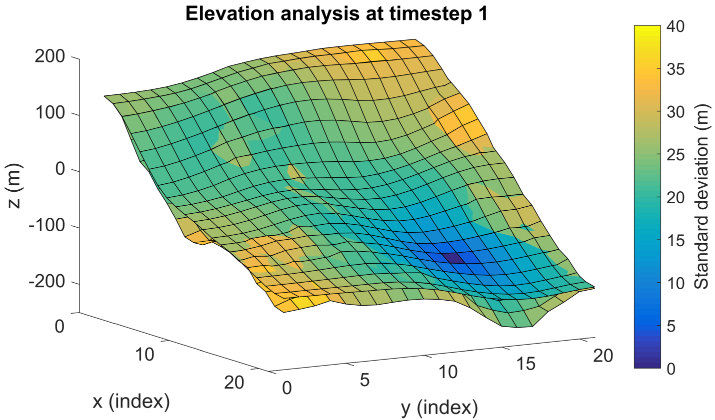
- Start with initial surface z_0^0 at time t_0
- Surface will ”diffuse” to yield new top surface z_1^1 at time t_1
- Elevation of surface j at time t_k is z_j^k for $j = 1, \ldots, k$



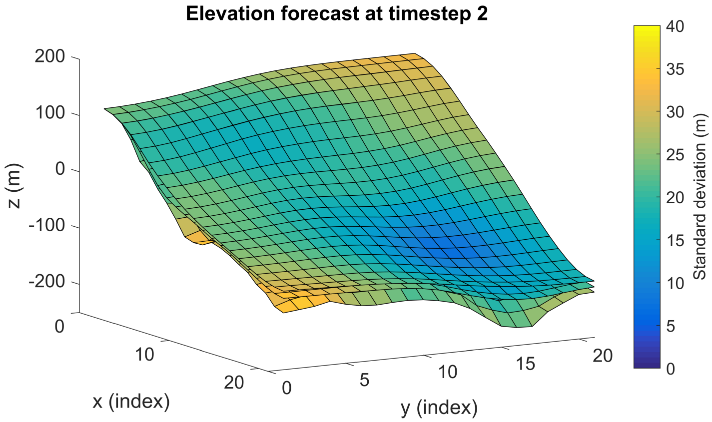
Time evolution of ensemble



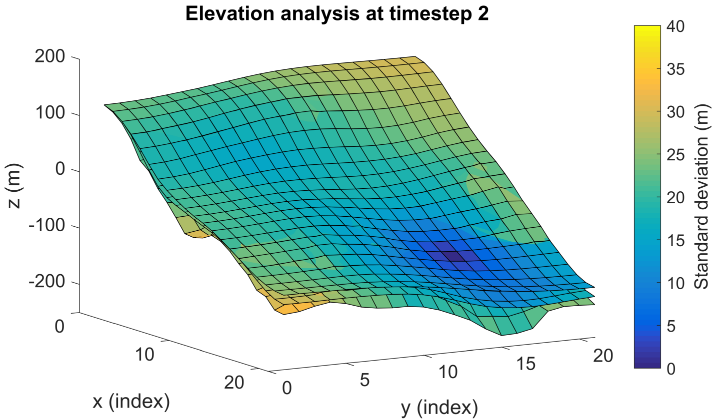
Time evolution of ensemble



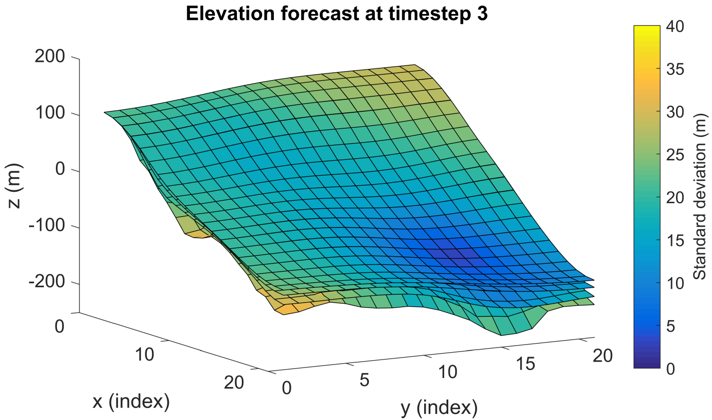
Time evolution of ensemble



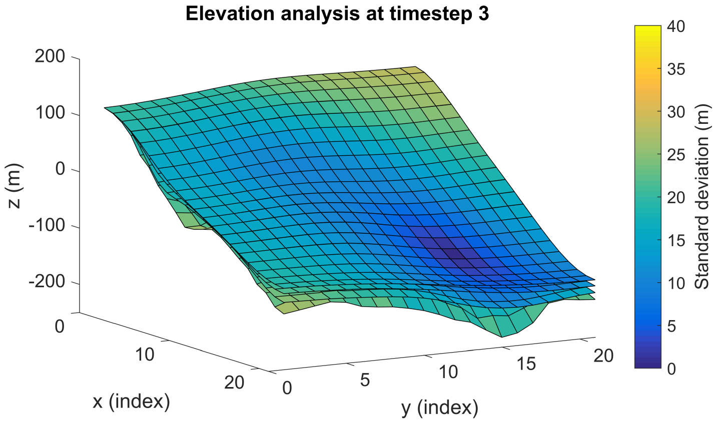
Time evolution of ensemble



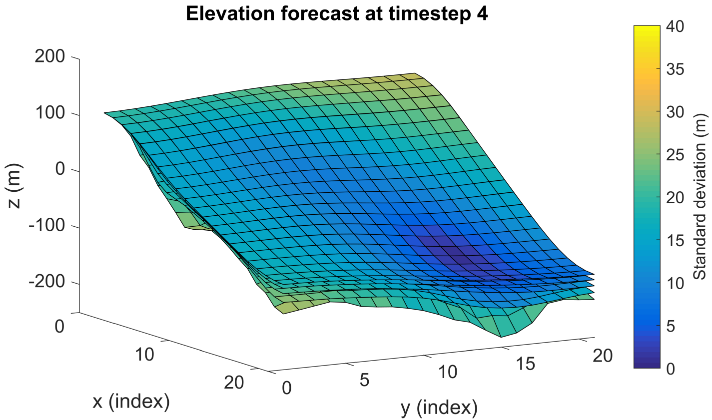
Time evolution of ensemble



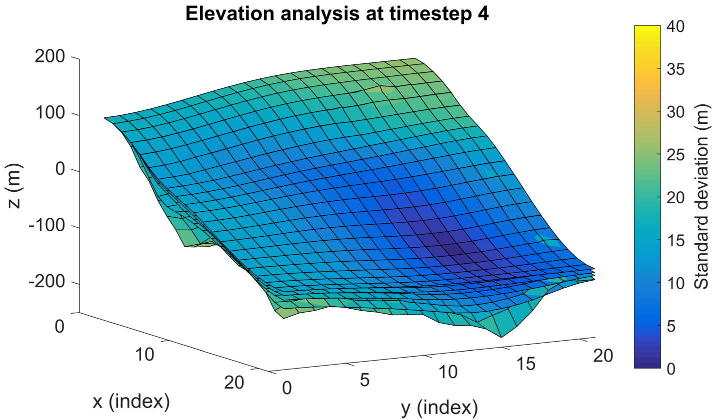
Time evolution of ensemble



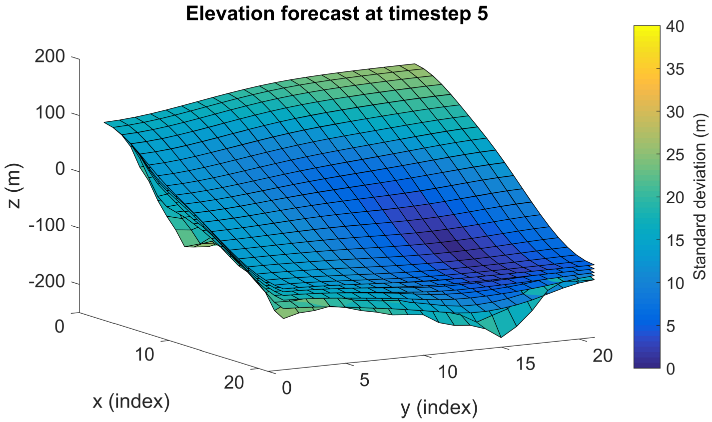
Time evolution of ensemble



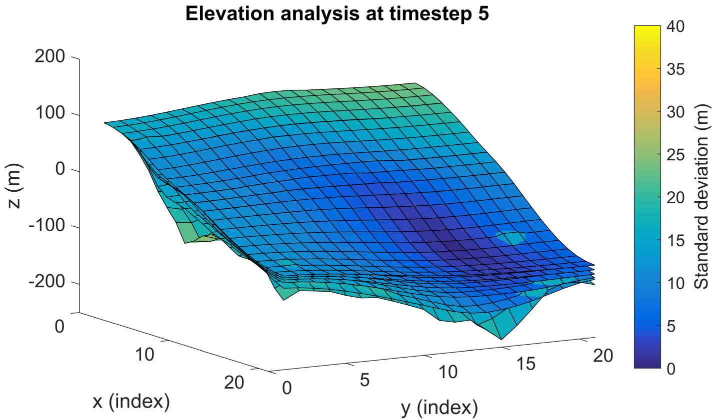
Time evolution of ensemble



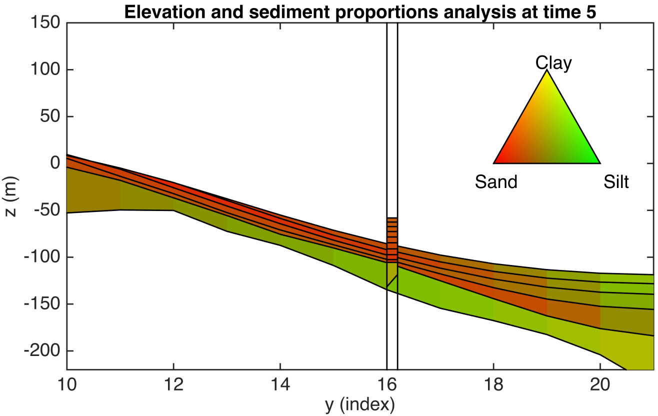
Time evolution of ensemble



Time evolution of ensemble



Time evolution of ensemble



Time evolution of ensemble

Sea level parameter - constant: $\theta(t) = \theta_0, \; t_{\text{start}} \leq t \leq t_{\text{end}}$