Data-Assimilation Fundamentals:

A Unified Formulation of the State and Parameter Estimation Problem

Geir Evensen, Femke C. Vossepoel, and Peter Jan van Leeuwen

NORCE TU Delft COLORADO STATE UNIVERSITY

Available from `https://github.com/geirev/Data-Assimilation-Fundamentals.git`

Springer Textbooks in Earth Sciences, Geography and Environment

Geir Evensen · Femke C. Vossepoel · Peter Jan van Leeuwen

Data Assimilation Fundamentals

This open-access textbook's significant contribution is the unified derivation of data-assimilation techniques from a common fundamental and optimal starting point, namely Bayes' theorem. Unique for this book is the "top-down" derivation of the assimilation methods. It starts from Bayes theorem and gradually introduces the assumptions and approximations needed to arrive at today's popular data-assimilation methods. This strategy is the opposite of most textbooks and reviews on data assimilation that typically take a bottom-up approach to derive a particular assimilation method. E.g., the derivation of the Kalman Filter from control theory and the derivation of the ensemble Kalman Filter as a low-rank approximation of the standard Kalman Filter. The bottom-up approach derives the assimilation methods from different mathematical principles, making it difficult to compare them. Thus, it is unclear which assumptions are made to derive an assimilation method and sometimes even which problem it aspires to solve. The book's top-down approach allows categorizing data-assimilation methods based on the approximations used. This approach enables the user to choose the most suitable method for a particular problem or application. Have you ever wondered about the difference between the ensemble 4DVar and the "ensemble randomized likelihood" (EnRML) methods? Do you know the differences between the ensemble smoother and the ensemble-Kalman smoother? Would you like to understand how a particle flow is related to a particle filter? In this book, we will provide clear answers to several such questions. The book provides the basis for an advanced course in data assimilation. It focuses on the unified derivation of the methods and illustrates their properties on multiple examples. It is suitable for graduate students, post-docs, scientists, and practitioners working in data assimilation.

ISBN 978-3-030-96708-6

9 783030 967086

▶ springer.com

Evensen · Vossepoel · Leeuwen

Data Assimilation Fundamentals

TEXTBOOK

Geir Evensen · Femke C. Vossepoel
Peter Jan van Leeuwen

Data Assimilation Fundamentals

A Unified Formulation of the State and Parameter Estimation Problem

OPEN ACCESS

Springer

Simple scalar example from Evensen (2009)

Simple scalar DA example

Given the problem

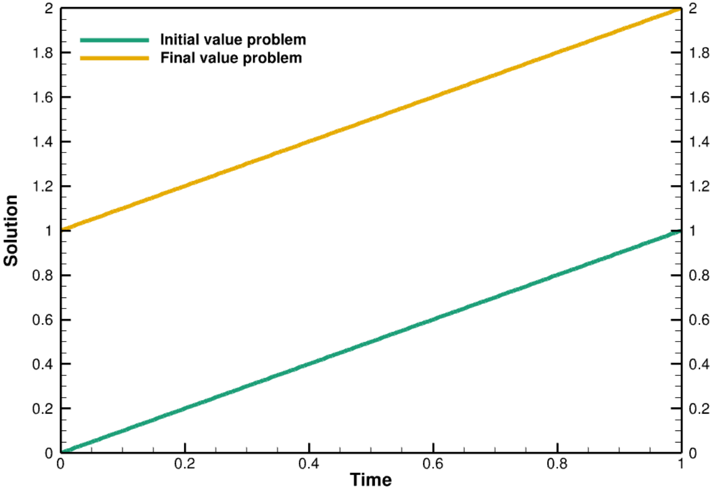
$$\frac{dx}{dt} = 1 \qquad \text{Linear model} \tag{1}$$

$$x(0) = 0 \qquad \text{Initial condition} \tag{2}$$

$$x(1) = 2 \qquad \text{Final condition} \tag{3}$$

- Overdetermined problem.
- No solution.

Initial- and final-value solutions



Available from: https://github.com/geirev/Data-Assimilation-Fundamentals/blob/main/presentations/uibcourse.pdf

Allowing for errors in model and conditions

$$\frac{dx}{dt} = 1 + q(t) \tag{4}$$

$$x(0) = 0 + a \tag{5}$$

$$x(1) = 2 + b \tag{6}$$

- Underdetermined problem.
- Infinitively many solutions.

Impose a statistical assumption on the error terms

The mean is zero: The variance is known: No cross correlations:

$$\overline{q(t)} = 0, \qquad \overline{q(t_1)q(t_2)} = C\delta(t_1 - t_2), \qquad \overline{q(t)a} = 0, \tag{7}$$

$$\overline{a} = 0, \qquad \overline{a^2} = C, \qquad \overline{ab} = 0, \tag{8}$$

$$\overline{b} = 0, \qquad \overline{b^2} = C, \qquad \overline{q(t)b} = 0. \tag{9}$$

We will search for a solution that

- is close to the conditions and almost satisfies the model,

by minimizing the error terms.

Define a quadratic cost function

$$\mathcal{J}[x] = C^{-1}\int_0^1 \left(\frac{dx}{dt} - 1\right)^2 dt + C^{-1}\left(x(0) - 0\right)^2 + C^{-1}\left(x(1) - 2\right)^2 \tag{10}$$

Then x is an extremum if

$$\delta\mathcal{J}[x] = \mathcal{J}[x + \delta x] - \mathcal{J}[x] = \mathcal{O}\left(\delta x^2\right) \tag{11}$$

when $\delta x \to 0$.

$$\mathcal{J}[x + \delta x] = C^{-1}\int_0^1 \left(\frac{dx}{dt} - 1 + \frac{d\delta x}{dt}\right)^2 dt + C^{-1}\left(x(0) - 0 + \delta x(0)\right)^2 + C^{-1}\left(x(1) - 2 + \delta x(1)\right)^2 \tag{12}$$

Variation of cost function gives

$$\int_0^1 \frac{d\delta x}{dt}\left(\frac{dx}{dt} - 1\right)dt \quad + \quad \delta x(0)\big(x(0) - 0\big) \quad + \quad \delta x(1)\big(x(1) - 2\big) = 0, \tag{13}$$

From integration by part we get

$$\delta x\left(\frac{dx}{dt} - 1\right)\Big|_0^1 \quad - \quad \int_0^1 \delta x \frac{d^2x}{dt^2}dt \quad + \quad \delta x(0)\big(x(0) - 0\big) \quad + \quad \delta x(1)\big(x(1) - 2\big) = 0. \tag{14}$$

Minimium of cost function

This gives the following system of equations

$$\delta x(0)\left(-\frac{dx}{dt}+1+x\right)\bigg|_{t=0}=0, \tag{15}$$

$$\delta x(1)\left(\frac{dx}{dt}-1+x-2\right)\bigg|_{t=1}=0, \tag{16}$$

$$\int_0^1 \delta x\left(\frac{d^2x}{dt^2}\right)dt=0, \tag{17}$$

or since δx is arbitrary....

Euler-Lagrange equation

The Euler–Lagrange equation

$$\frac{d^2x}{dt^2} = 0, \tag{18}$$

$$\frac{dx}{dt} - x = 1 \quad \text{for } t = 0, \tag{19}$$

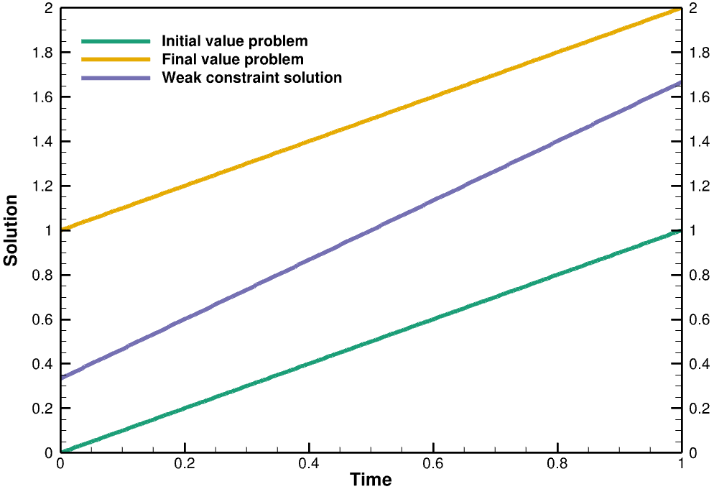
$$\frac{dx}{dt} + x = 3 \quad \text{for } t = 1, \tag{20}$$

- Elliptic boundary value problem in time.
- It has a unique solution.

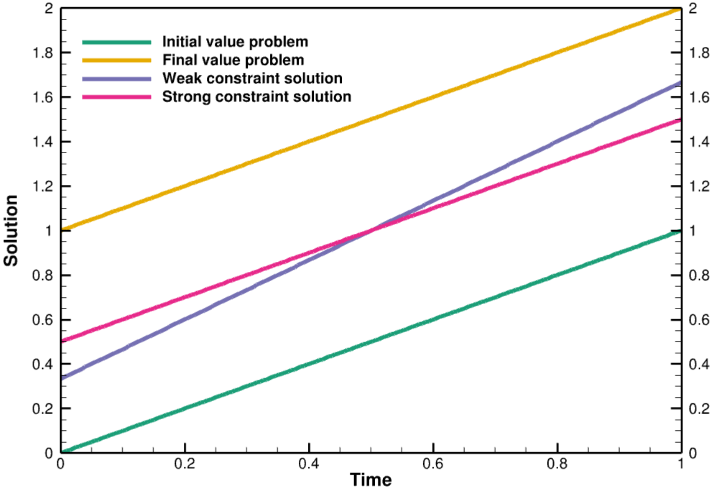
$$x = c_1 t + c_2, \tag{21}$$

with $c_1 = 4/3$ and $c_2 = 1/3$.

Weak-constraint solution



Strong-constraint solution



Available from: https://github.com/geirev/Data-Assimilation-Fundamentals/blob/main/presentations/uibcourse.pdf

Summary

- A well-posed model with conditions has a unique solution.
- Additional conditions makes the problem over determined.
- Allowing for errors gives infinitely many solutions.
- Specify mean and covariance for error terms.
- Define variational inverse problem for least-squares solution.
- The Euler-Lagrange equation defines the least-squares solution.
- The problem becomes a boundary-value problem in time.
- Weak-constraint solution: almost satisfies dynamics and data.
- Strong-constraint solution: satisfies dynamics, and close to data.

Probabilistic formulation with Gaussian priors

Initial conditions

$$f\big(x(0)\big) \propto \exp\left(-\frac{a^2}{C}\right) = \exp\left(-\frac{\big(x(0)-0\big)^2}{C}\right) \tag{22}$$

Model evolution

$$f\big(x|x(0)\big) \propto \exp\left(-\int_0^1 \frac{q^2}{C}dt\right) = \exp\left(-\int_0^1 \frac{1}{C}\left(\frac{dx}{dt}-1\right)^2 dt\right) \tag{23}$$

A measurement

$$f\big(d|x(1)\big) \propto \exp\left(-\frac{b^2}{C}\right) = \exp\left(-\frac{\big(x(1)-2\big)^2}{C}\right) \tag{24}$$

Bayes' theorem for the scalar model with Gaussian priors

$$
\begin{aligned}
f(x|d) &= \frac{f(d|x)\,f(x)}{f(d)} && (25)\\
&\propto f\left(d|x(1)\right)\left(f\left(x|x(0)\right) f\left(x(0)\right)\right) && (26)\\
&= \mathsf{exp}\left(-\frac{\left(x(1)-2\right)^2}{C}\right) \mathsf{exp}\left(-\int_0^1 \frac{1}{C}\left(\frac{dx}{dt}-1\right)^2 dt\right) \mathsf{exp}\left(-\frac{\left(x(0)-0\right)^2}{C}\right) && (27)\\
&= \mathsf{exp}\left(-\mathcal{J}[x]\right) && (28)
\end{aligned}
$$

Hence, maximizing the probability is equivalent to minimizing the cost function.

Why Bayes Theorem?

- Provides a fundamental *framework* for data assimilation.
- All data-assimilation methods can be derived from Bayes'.

Properties of a probability density function

- The graph of the density function is continuous, since it is defined over a continuous range over a continuous variable.
- The total probability

$$P(x) = \int_{-\infty}^{\infty} f(x)dx = 1 \tag{29}$$

- The probability of $x \in [a, b]$ is

$$P(x \in [a, b]) = \int_{a}^{b} f(x)dx \tag{30}$$

- And two special cases

$$P(x = c) = \int_{c}^{c} f(x)dx = 0 \qquad \text{and} \qquad \int_{-\infty}^{\infty} \delta(x)dx = 1 \tag{31}$$

Also, we have

- The joint probability
$$f(x,y) = f(x)f(y|x) = f(y)f(x|y) \tag{32}$$
- Solving for $f(x|y)$ gives Bayes' theorem
$$f(x|y) = \frac{f(x)f(y|x)}{f(y)} \tag{33}$$
- Bayes states that "the probability of x given y, is equal to the probability of x, times the likelihood of y given x, divided by the probability of y."
- Here $f(y)$ is a normalization constant so that the integral of $f(x|y)$ becomes one.

Bayes' theorem

Given (now using again $\mathbf{z}$ and $\mathbf{d}$ for x and y):

- A state variable $\mathbf{z}$ and its prior pdf: $f(\mathbf{z})$
- A vector of observations $\mathbf{d}$ and their likelihood: $f(\mathbf{d}|\mathbf{z})$
- Bayes' theorem defines the posterior pdf, $f(\mathbf{z}|\mathbf{d})$:

Bayes' theorem

$$f(\mathbf{z}|\mathbf{d}) = \frac{f(\mathbf{z})f(\mathbf{d}|\mathbf{z})}{f(\mathbf{d})} \tag{34}$$

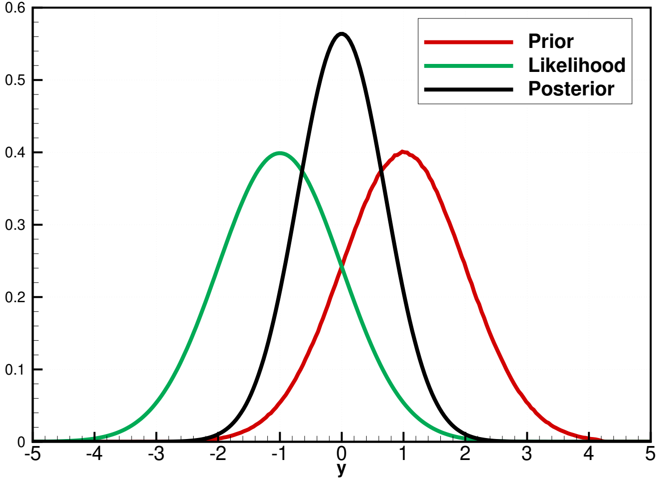
What is the likelihood function: $f(d|x)$

- The likelihood function $f(d|x)$ is the probability of the observed data d for various values of the unknown parameters x.
- The likelihood is used after data are available to describe a plausibility of a parameter value x.
- The likelihood does not have to integrate to one.

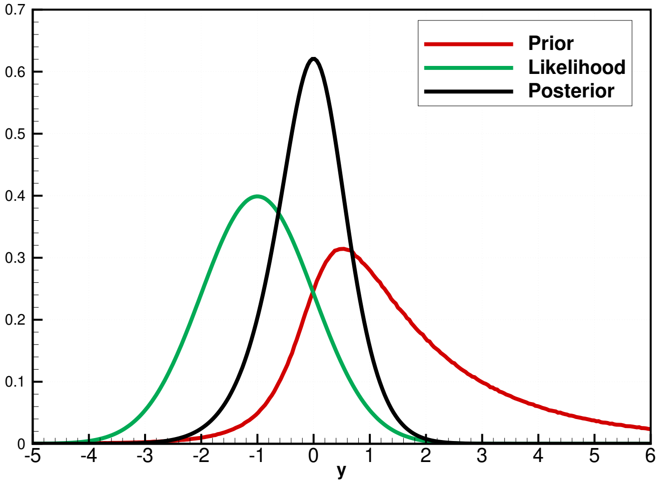
Likelihood is the plausability of a particular distribution explaining the given data. The higher the likelihood of a distribution, the more likely it is to explain the observed data.

Probability is how likely are the chances of a certain data to occur if the model parameters are fixed and Likelihood is the chances of a particular model parameter explaining the given observed data.

Example of using Bayes' theorem



Example of using Bayes' theorem



Bayes' formula is the optimal starting point

Bayes' formula arises as the first-order optimality condition from the joint minimization of the Kullback-Leibler (KL) divergence between a posterior and prior distribution and the mean-square errors of the data represented by the likelihood[1].

Bayes' formula elegantly shows how to update prior information when new information becomes available.

One of the strengths of Bayes' formula is that it does not try to solve the ill-defined problem of "inverting observations" but instead updates prior knowledge.

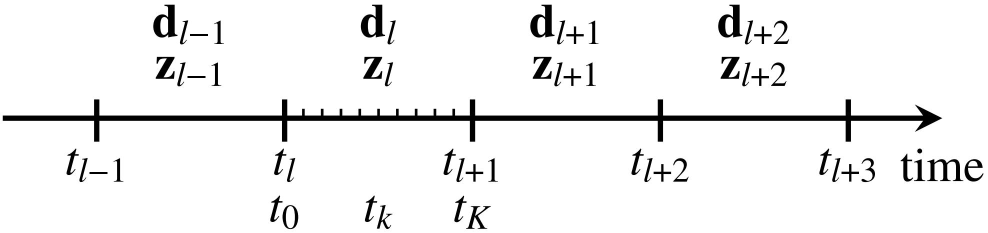
[1] Bui-Thanh, T. (2021): https://sinews.siam.org/Details-Page/the-optimality-of-bayes-theorem

We start from Bayes' theorem

$$f(\mathcal{Z}|\mathcal{D}) = \frac{f(\mathcal{D}|\mathcal{Z})f(\mathcal{Z})}{f(\mathcal{D})}. \tag{35}$$

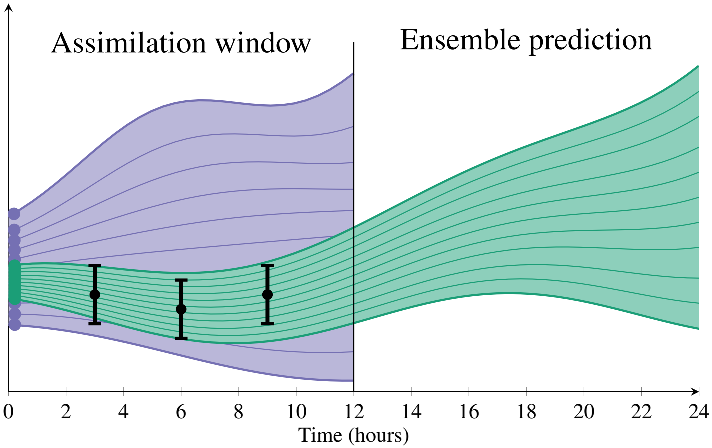
- $\mathcal{Z} = (\mathbf{z}_0, \mathbf{z}_1, \ldots, \mathbf{z}_L)$ is the vector of state variables on all the assimilation windows.
- $\mathcal{D} = (\mathbf{d}_1, \ldots, \mathbf{d}_L)$ is the vector containing all the measurements.

Split time into data-assimilation windows



- We consider the DA problem for one single window.
- Errors propagate from one window to the next.

Weather prediction configuration



Model is Markov process

Approximation 1 (Model is 1st-order Markov process)

We assume the dynamical model is a 1st-order Markov process.

$$f(\mathbf{z}_l|\mathbf{z}_{l-1},\mathbf{z}_{l-2},\dots,\mathbf{z}_0) = f(\mathbf{z}_l|\mathbf{z}_{l-1}), \tag{36}$$

Independent measurements

Approximation 2 (Independent measurements)

We assume that measurements are independent between different assimilation windows.

Independent measurements have uncorrelated errors

$$f(\mathcal{D}|\mathcal{Z}) = \prod_{l=1}^{L} f(\mathbf{d}_l|\mathbf{z}_l). \tag{37}$$

Bayes becomes

$$f(\mathcal{Z}|\mathcal{D}) \propto \prod_{l=1}^{L} f(\mathbf{d}_l|\mathbf{z}_l) \prod_{l=1}^{L} f(\mathbf{z}_l|\mathbf{z}_{l-1}) f(\mathbf{z}_0). \tag{38}$$

Recursive form of Bayes

$$f(\mathbf{z}_1, \mathbf{z}_0|\mathbf{d}_1) = \frac{f(\mathbf{d}_1|\mathbf{z}_1) f(\mathbf{z}_1|\mathbf{z}_0) f(\mathbf{z}_0)}{f(\mathbf{d}_1)}, \tag{39}$$

$$f(\mathbf{z}_2, \mathbf{z}_1, \mathbf{z}_0|\mathbf{d}_1, \mathbf{d}_2) = \frac{f(\mathbf{d}_2|\mathbf{z}_2) f(\mathbf{z}_2|\mathbf{z}_1) f(\mathbf{z}_1, \mathbf{z}_0|\mathbf{d}_1)}{f(\mathbf{d}_2)}, \tag{40}$$

$$\vdots$$

$$f(\mathcal{Z}|\mathcal{D}) = \frac{f(\mathbf{d}_L|\mathbf{z}_L) f(\mathbf{z}_L|\mathbf{z}_{L-1}) f(\mathbf{z}_{L-1}, \dots, \mathbf{z}_0|\mathbf{d}_{L-1}, \dots, \mathbf{d}_1)}{f(\mathbf{d}_L)}. \tag{41}$$

Filtering assumption

Approximation 3 (Filtering assumption)

We approximate the full smoother solution with a sequential data-assimilation solution. We only update the solution in the current assimilation window, and we do not project the measurement's information backward in time from one assimilation window to the previous ones.

Recursive Bayes' for filtering

$$f(\mathbf{z}_1|\mathbf{d}_1) = \frac{f(\mathbf{d}_1|\mathbf{z}_1)\ \int f(\mathbf{z}_1|\mathbf{z}_0) f(\mathbf{z}_0)\, d\mathbf{z}_0}{f(\mathbf{d}_1)} = \frac{f(\mathbf{d}_1|\mathbf{z}_1) f(\mathbf{z}_1)}{f(\mathbf{d}_1)}, \tag{42}$$

$$f(\mathbf{z}_2|\mathbf{d}_1, \mathbf{d}_2) = \frac{f(\mathbf{d}_2|\mathbf{z}_2)\ \int f(\mathbf{z}_2|\mathbf{z}_1) f(\mathbf{z}_1|\mathbf{d}_1)\, d\mathbf{z}_1}{f(\mathbf{d}_2)} = \frac{f(\mathbf{d}_2|\mathbf{z}_2) f(\mathbf{z}_2|\mathbf{d}_1)}{f(\mathbf{d}_2)}, \tag{43}$$

$$\vdots$$

Or just Bayes' for the assimilation window

$$\boxed{f(\mathbf{z}|\mathbf{d}) = \frac{f(\mathbf{d}|\mathbf{z})f(\mathbf{z})}{f(\mathbf{d})}, \qquad (44)}$$

Discrete dynamic model with uncertain inputs

$$\mathbf{x}_k = \mathbf{m}(\mathbf{x}_{k-1}, \boldsymbol{\theta}, \mathbf{u}_k, \mathbf{q}_k). \tag{45}$$

- $\mathbf{x}_k$ is the model state.
- $\boldsymbol{\theta}$ are model parameters.
- $\mathbf{u}_k$ are model controls.
- $\mathbf{q}_k$ are model errors.
- Define $\mathbf{x} = (\mathbf{x}_0, \ldots, \mathbf{x}_K)$ as model state over the assimilation window.
- Define $\mathbf{q} = (\mathbf{q}_0, \ldots, \mathbf{q}_K)$ as model errors over the assimilation window.
- Define $\mathbf{u} = (\mathbf{u}_0, \ldots, \mathbf{u}_K)$ as model forcing over the assimilation window.
- Define $\mathbf{z} = (\mathbf{x}, \boldsymbol{\theta}, \mathbf{u}, \mathbf{q})$ as state vector for assimilation problem.

Parameter estimation vs state estimation

Including the model errors in $\mathbf{z}$ allows us to consider the model and measurement operator as deterministic.

Example:

- Solve for uncertain input parameters.
- Condition on measurements distributed over the assimilation window.

$$\mathbf{y} = \mathbf{g}(\mathbf{z}) = \mathbf{h}\big(\mathbf{m}(\boldsymbol{\theta}, \mathbf{q})\big). \tag{46}$$

$$\mathbf{y} = \mathbf{g}(\mathbf{z}) = \mathbf{h}\big(\mathbf{m}(\mathbf{x}_0, \mathbf{q})\big). \tag{47}$$

Error propagation by Fokker-Planck equation

Stochastic model

$$d\mathbf{x} = \mathbf{m}(\mathbf{x})\,dt + d\mathbf{q}. \tag{48}$$

Fokker-Planck is an advection-diffusion equation in the state-space

$$\frac{\partial f(\mathbf{x})}{\partial t} + \sum_i \frac{\partial\big(m_i(\mathbf{x})\,f(\mathbf{x})\big)}{\partial x_i} = \frac{1}{2}\mathbf{C}_{qq}\sum_{i,j}\frac{\partial^2 f(\mathbf{x})}{\partial x_i\,\partial x_j}. \tag{49}$$

Error propagation by covariance evolution

Comparing the evolution of the true model state with that of our estimated model state

$$\mathbf{x}^{\mathrm{t}}_{k+1} = \mathbf{m}(\mathbf{x}^{\mathrm{t}}_k) + \mathbf{q}_k \approx \mathbf{m}(\mathbf{x}_k) + \mathbf{M}_k(\mathbf{x}^{\mathrm{t}}_k - \mathbf{x}_k) + \mathbf{q}_k, \tag{50}$$

$$\mathbf{x}_{k+1} = \mathbf{m}(\mathbf{x}_k), \tag{51}$$

Subtract Eq. (51) from Eq. (50), square the result, and take the expectation,

$$\mathbf{C}_{xx,k+1} \approx \mathbf{M}_k \mathbf{C}_{xx,k} \mathbf{M}_k^{\mathrm{T}} + \mathbf{C}_{qq,k}. \tag{52}$$

- $\mathbf{M}_k$ is the model's tangent-linear operator evaluated at $\mathbf{x}_k$.
- $\mathbf{C}_{qq}$ is the model error covariance matrix.

Error propagation by ensemble predictions

- Represent uncertainty by an ensemble of samples

$$\mathbf{x}_{j,0} \sim f(\mathbf{x}) \quad \text{and} \quad \mathbf{q}_{j,k} \sim f(\mathbf{x}_{k+1}|\mathbf{x}_k) \tag{53}$$

- Nonlinear propagation of uncertainty by ensemble integrations using the dynamical model.

$$\mathbf{x}_{j,k+1} = \mathbf{m}(\mathbf{x}_{j,k}, \mathbf{q}_{j,k}). \tag{54}$$

We can then compute statistics like mean and covariance, e.g.,

$$\mathbb{E}[\mathbf{x}] \approx \overline{\mathbf{x}} = \frac{1}{N}\sum_{j=1}^{N} \mathbf{x}_j \tag{55}$$

$$\mathbf{C}_{xx} \approx \overline{(\mathbf{x}-\overline{\mathbf{x}})(\mathbf{x}-\overline{\mathbf{x}})^{\mathrm{T}}} = \frac{1}{N-1}\sum_{j=1}^{N} (\mathbf{x}_j-\overline{\mathbf{x}})(\mathbf{x}_j-\overline{\mathbf{x}})^{\mathrm{T}} \tag{56}$$

Marginal posterior pdf for perfect models

Nonlinear "perfect" model and measurements

$$\mathbf{y} = \mathbf{g}(\mathbf{z}) \qquad \mathbf{d} = \mathbf{y} + \mathbf{e}$$

Bayesian formulation

$$f(\mathbf{z}, \mathbf{y}|\mathbf{d}) \propto f(\mathbf{d}|\mathbf{y}) f(\mathbf{y}|\mathbf{z}) f(\mathbf{z})$$

Model pdf

$$f(\mathbf{y}|\mathbf{z}) = \delta\big(\mathbf{y} - \mathbf{g}(\mathbf{z})\big)$$

Marginal pdf

$$f(\mathbf{z}|\mathbf{d}) \propto \int f(\mathbf{d}|\mathbf{y}) f(\mathbf{y}|\mathbf{z}) f(\mathbf{z}) d\mathbf{y} = f\big(\mathbf{d}|\mathbf{g}(\mathbf{z})\big) f(\mathbf{z})$$

Bayes' theorem related to the predicted measurements

We introduce nonlinearity through the likelihood

$$f(\mathbf{z}|\mathbf{d}) = \frac{f\big(\mathbf{d}|\mathbf{g}(\mathbf{z})\big)\, f(\mathbf{z})}{f(\mathbf{d})}. \tag{57}$$

The MAP solution

Gaussian assumption

Approximation 4 (Gaussian prior and likelihood)

We assume that the prior distributions of the state vector's components $\mathbf{z}$ *and observation errors* $\boldsymbol{\epsilon}$ *are both Gaussian distributed.*

$$f(\mathbf{z}|\mathbf{d}) \propto \exp\left\{-\mathcal{J}(\mathbf{z})\right\}, \tag{58}$$

Leads to a cost-function formulation for the MAP solution

Cost function

$$\mathcal{J}(\mathbf{z}) = \frac{1}{2}\left(\mathbf{z}-\mathbf{z}^{\mathrm{f}}\right)^{\mathrm{T}}\mathbf{C}_{zz}^{-1}\left(\mathbf{z}-\mathbf{z}^{\mathrm{f}}\right) + \frac{1}{2}\left(\mathbf{g}(\mathbf{z})-\mathbf{d}\right)^{\mathrm{T}}\mathbf{C}_{dd}^{-1}\left(\mathbf{g}(\mathbf{z})-\mathbf{d}\right). \tag{59}$$

The gradient set to zero

$$\mathbf{C}_{zz}^{-1}\left(\mathbf{z}^{\mathrm{a}}-\mathbf{z}^{\mathrm{f}}\right) + \nabla_{\mathbf{z}}\mathbf{g}\left(\mathbf{z}^{\mathrm{a}}\right)\mathbf{C}_{dd}^{-1}\left(\mathbf{g}\left(\mathbf{z}^{\mathrm{a}}\right)-\mathbf{d}\right) = 0. \tag{60}$$

- There is no explicit solution of the gradient equation.

Gauss-Newton methods solves for the MAP estimate

Gauss-Newton iteration

$$\mathbf{z}^{i+1} = \mathbf{z}^i - \gamma^i \left(\mathbf{C}_{zz}^{-1} + \mathbf{G}^{i\mathrm{T}}\mathbf{C}_{dd}^{-1}\mathbf{G}^i\right)^{-1} \left(\mathbf{C}_{zz}^{-1}(\mathbf{z}^i - \mathbf{z}^{\mathrm{f}}) + \mathbf{G}^{i\mathrm{T}}\mathbf{C}_{dd}^{-1}\left(\mathbf{g}(\mathbf{z}^i) - \mathbf{d}\right)\right). \tag{61}$$

- The incremental formulation is sometimes more convenient.

Incremental Gauss-Newton methods

Quadratic cost function for the increments

$$\mathcal{J}(\delta\mathbf{z}) = \frac{1}{2}\left(\delta\mathbf{z} - \boldsymbol{\xi}^i\right)^{\mathrm{T}} \mathbf{C}_{zz}^{-1} \left(\delta\mathbf{z} - \boldsymbol{\xi}^i\right) + \frac{1}{2}\left(\mathbf{G}^i\delta\mathbf{z} - \boldsymbol{\eta}^i\right)^{\mathrm{T}} \mathbf{C}_{dd}^{-1} \left(\mathbf{G}^i\delta\mathbf{z} - \boldsymbol{\eta}^i\right). \tag{62}$$

with

$$\mathbf{z}^{i+1} = \mathbf{z}^i + \delta\mathbf{z}, \tag{63}$$

$$\boldsymbol{\eta}^i = \mathbf{d} - \mathbf{g}\left(\mathbf{z}^i\right), \tag{64}$$

$$\boldsymbol{\xi}^i = \mathbf{z}^{\mathrm{f}} - \mathbf{z}^i. \tag{65}$$

- Sequence of linear iterates.
- Solved by SC-4DVar, WC-4DVar, and Representer method.

Standard strong constraint 4DVar

Standard SC-4DVar

Model with initial condition and poorly known parameter

$$\mathbf{x}_0 = \mathbf{x}_0^{\mathrm{f}} + \mathbf{x}_0', \tag{66}$$

$$\boldsymbol{\theta} = \boldsymbol{\theta}^{\mathrm{f}} + \boldsymbol{\theta}', \tag{67}$$

$$\mathbf{x}_{k+1} = \mathbf{m}(\mathbf{x}_k, \boldsymbol{\theta}), \tag{68}$$

Measurements

$$\mathbf{d} = \mathbf{h}(\mathbf{x}) + \mathbf{e}. \tag{69}$$

Problem formulation

State vector and covariance matrix

$$\mathbf{z} = \begin{pmatrix} \mathbf{x}_0 \\ \boldsymbol{\theta} \end{pmatrix} \quad \text{and} \quad \mathbf{C}_{zz} = \begin{pmatrix} \mathbf{C}_{x_0 x_0} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_{\theta\theta} \end{pmatrix}, \tag{70}$$

SC-4DVar costfunction

$$\mathcal{J}(\mathbf{z}) = \frac{1}{2}\left(\mathbf{z} - \mathbf{z}^{\mathrm{f}}\right)^{\mathrm{T}} \mathbf{C}_{zz}^{-1} \left(\mathbf{z} - \mathbf{z}^{\mathrm{f}}\right) + \frac{1}{2}\left(\mathbf{h}(\mathbf{x}) - \mathbf{d}\right)^{\mathrm{T}} \mathbf{C}_{dd}^{-1} \left(\mathbf{h}(\mathbf{x}) - \mathbf{d}\right), \tag{71}$$

Solve for initial condition and parameter that minimize the cost function

Lagrangian formulation

$$
\begin{aligned}
\mathcal{L}(\mathbf{x}_0, \ldots, \mathbf{x}_{K+1}, \boldsymbol{\theta}, \boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_{K+1}) = {} & \frac{1}{2}\left(\mathbf{x}_0 - \mathbf{x}_0^{\mathrm{f}}\right)^{\mathrm{T}} \mathbf{C}_{x_0 x_0}^{-1} \left(\mathbf{x}_0 - \mathbf{x}_0^{\mathrm{f}}\right) \\
& + \frac{1}{2}\left(\boldsymbol{\theta} - \boldsymbol{\theta}^{\mathrm{f}}\right)^{\mathrm{T}} \mathbf{C}_{\theta\theta}^{-1} \left(\boldsymbol{\theta} - \boldsymbol{\theta}^{\mathrm{f}}\right) \\
& + \frac{1}{2}\left(\mathbf{h}(\mathbf{x}) - \mathbf{d}\right)^{\mathrm{T}} \mathbf{C}_{dd}^{-1} \left(\mathbf{h}(\mathbf{x}) - \mathbf{d}\right) \\
& + \sum_{k=0}^{K} \boldsymbol{\lambda}_{k+1}^{\mathrm{T}} \left(\mathbf{x}_{k+1} - \mathbf{m}(\mathbf{x}_k, \boldsymbol{\theta})\right).
\end{aligned}
\tag{72}
$$

We include the perfect model by introducing a Lagrangian multiplier $\boldsymbol{\lambda}$.

Gradient of Lagrangian

$$
\begin{aligned}
\nabla_{\mathbf{x}_k}\mathcal{L}(\mathbf{x},\boldsymbol{\theta},\boldsymbol{\lambda}) &= \mathbf{H}_k^{\mathrm{T}}\,\mathbf{C}_{dd}^{-1}\left(\mathbf{h}(\mathbf{x})-\mathbf{d}\right)+\boldsymbol{\lambda}_k-\mathbf{M}_{x,k}^{\mathrm{T}}\boldsymbol{\lambda}_{k+1}, && (73)\\
\nabla_{\mathbf{x}_{K+1}}\mathcal{L}(\mathbf{z},\mathbf{x},\boldsymbol{\lambda}) &= \boldsymbol{\lambda}_{K+1}, && (74)\\
\nabla_{\mathbf{x}_0}\mathcal{L}(\mathbf{x},\boldsymbol{\theta},\boldsymbol{\lambda}) &= \mathbf{C}_{zz}^{-1}\left(\mathbf{x}_0-\mathbf{x}_0^{\mathrm{f}}\right)-\mathbf{M}_{x,0}^{\mathrm{T}}\boldsymbol{\lambda}_1 \\
&= \mathbf{C}_{zz}^{-1}\left(\mathbf{x}_0-\mathbf{x}_0^{\mathrm{f}}\right)-\boldsymbol{\lambda}_0, && (75)\\
\nabla_{\boldsymbol{\theta}}\mathcal{L}(\mathbf{x},\boldsymbol{\theta},\boldsymbol{\lambda}) &= \mathbf{C}_{\theta\theta}^{-1}\left(\boldsymbol{\theta}-\boldsymbol{\theta}^{\mathrm{f}}\right)-\sum_{k=0}^{K}\mathbf{M}_{\theta,k}^{\mathrm{T}}\boldsymbol{\lambda}_{k+1}, && (76)\\
\nabla_{\boldsymbol{\lambda}_k}\mathcal{L}(\mathbf{x},\boldsymbol{\theta},\boldsymbol{\lambda}) &= \mathbf{x}_{k+1}-\mathbf{m}(\mathbf{x}_k,\boldsymbol{\theta}). && (77)
\end{aligned}
$$

Euler-Lagrange equation(s)

Forward model

$$\mathbf{x}_0 = \mathbf{x}_0^{\mathrm{f}} + \mathbf{C}_{x_0 x_0} \boldsymbol{\lambda}_0, \tag{78}$$

$$\boldsymbol{\theta} = \boldsymbol{\theta}^{\mathrm{f}} + \mathbf{C}_{\theta\theta} \sum_{k=0}^{K} \mathbf{M}_{\theta,k}^{\mathrm{T}} \boldsymbol{\lambda}_{k+1}, \tag{79}$$

$$\mathbf{x}_{k+1} = \mathbf{m}(\mathbf{x}_k, \boldsymbol{\theta}), \tag{80}$$

Backward model for the adjoint variable

$$\boldsymbol{\lambda}_{K+1} = 0, \tag{81}$$

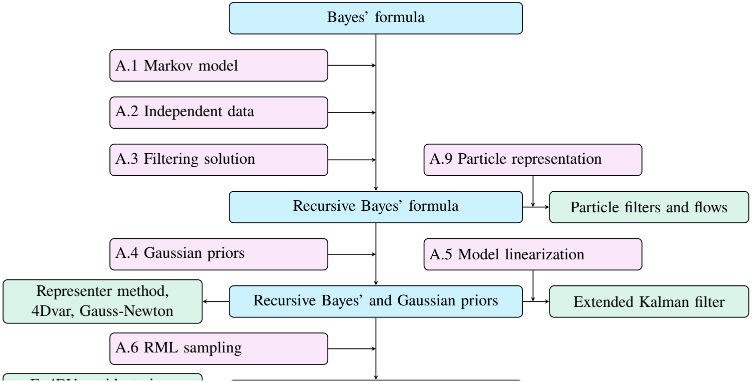
$$\boldsymbol{\lambda}_k = \mathbf{M}_{x,k}^{\mathrm{T}} \boldsymbol{\lambda}_{k+1} - \mathbf{H}_k^{\mathrm{T}} \mathbf{C}_{dd}^{-1} \big(\mathbf{h}(\mathbf{x}) - \mathbf{d}\big). \tag{82}$$

Coupled two-point boundary-value problem in time.

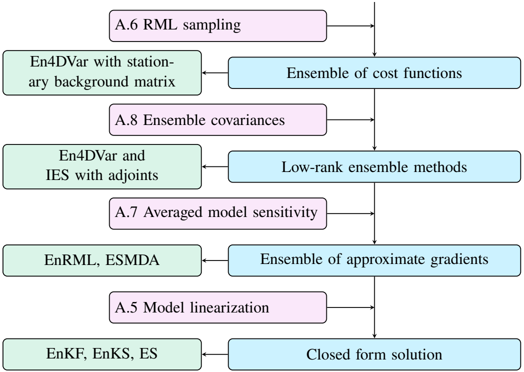
SC-4DVar algorithm

Input: $\mathbf{z}^{\mathrm{f}} \in \Re^{n}$; $\mathbf{d} \in \Re^{m}$ ▷ Prior initial conditions and observations
$\mathbf{x}_0 = \mathbf{x}_0^{\mathrm{f}}$ ▷ Initialization of $\mathbf{x}_0$
$\boldsymbol{\theta} = \boldsymbol{\theta}^{\mathrm{f}}$ ▷ Initialization of $\boldsymbol{\theta}$
repeat ▷ Iteration loop
  for $k = 0 : K$ **do** ▷ Integrate forward model
    $\mathbf{x}_{k+1} = \mathbf{m}(\mathbf{x}_k, \boldsymbol{\theta})$
  end for
  $\boldsymbol{\lambda}_{K+1} = 0$
  for $k = K : 0$ **do** ▷ Integrate backward adjoint model
    $\boldsymbol{\lambda}_k = \mathbf{M}_{x,k}^{\mathrm{T}} \boldsymbol{\lambda}_{k+1} - \mathbf{H}_k^{\mathrm{T}} \mathbf{C}_{dd}^{-1} (\mathbf{H}\mathbf{x} - \boldsymbol{\eta})$
  end for
  $\mathbf{x}_0 \leftarrow \mathbf{x}_0 - \gamma \mathbf{B} \nabla_{x_0} \mathcal{L}(\mathbf{x}, \boldsymbol{\theta}, \boldsymbol{\lambda})$ ▷ Update $\mathbf{x}_0$ using Eq. (75)
  $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \gamma \mathbf{B} \nabla_{\theta} \mathcal{L}(\mathbf{x}, \boldsymbol{\theta}, \boldsymbol{\lambda})$ ▷ Update $\boldsymbol{\theta}$ using Eq. (76)
until convergence

Overview of approximations and methods



Overview of approximations and methods



Linearization

Starting point

Start from the cost-function from Eq. (59)

$$\mathcal{J}(\mathbf{z}) = \frac{1}{2}\left(\mathbf{z} - \mathbf{z}^{\mathrm{f}}\right)^{\mathrm{T}} \mathbf{C}_{zz}^{-1}\left(\mathbf{z} - \mathbf{z}^{\mathrm{f}}\right) + \frac{1}{2}\left(\mathbf{g}(\mathbf{z}) - \mathbf{d}\right)^{\mathrm{T}} \mathbf{C}_{dd}^{-1}\left(\mathbf{g}(\mathbf{z}) - \mathbf{d}\right), \tag{83}$$

with the gradient in Eq. (60) set to zero

$$\mathbf{C}_{zz}^{-1}\left(\mathbf{z}^{\mathrm{a}} - \mathbf{z}^{\mathrm{f}}\right) + \nabla_{\mathbf{z}}\mathbf{g}\left(\mathbf{z}^{\mathrm{a}}\right)\mathbf{C}_{dd}^{-1}\left(\mathbf{g}\left(\mathbf{z}^{\mathrm{a}}\right) - \mathbf{d}\right) = 0. \tag{84}$$

- There is no explicit solution of the gradient equation.

Linearization leads to an approximate explicit solution

Approximation 5 (Linearization)

Linearize $\mathbf{g}(\mathbf{z})$ *around the prior estimate* $\mathbf{z}^{\mathrm{f}}$,

$$\mathbf{g}(\mathbf{z}) \approx \mathbf{g}(\mathbf{z}^{\mathrm{f}}) + \mathbf{G}(\mathbf{z} - \mathbf{z}^{\mathrm{f}}), \tag{85}$$

and approximate the gradient evaluated at the prior estimate

$$\nabla_{\mathbf{z}}\mathbf{g}(\mathbf{z}) \approx \mathbf{G}^{\mathrm{T}}, \tag{86}$$

where we have defined

$$\mathbf{G}^{\mathrm{T}} = \nabla_{\mathbf{z}}\mathbf{g}(\mathbf{z})\big|_{\mathbf{z}=\mathbf{z}^{\mathrm{f}}}. \tag{87}$$

$\mathbf{G}$ *is the tangent-linear operator of* $\mathbf{g}(\mathbf{z})$ *and* $\mathbf{G}^{\mathrm{T}}$ *is its "adjoint."*

$$\mathbf{M}_k^{\mathrm{T}} = \nabla_{\mathbf{z}}\mathbf{m}(\mathbf{z})\big|_{\mathbf{z}=\mathbf{z}_k} \quad \textit{and} \quad \mathbf{H}^{\mathrm{T}} = \nabla_{\mathbf{m}(\mathbf{z})}\mathbf{h}(\mathbf{m}(\mathbf{z}))\big|_{\mathbf{z}=\mathbf{z}_k}. \tag{88}$$

Solution by linearization

The linearization in Approx. 5 leads to the following form of (84)

$$\mathbf{C}_{zz}^{-1}\left(\mathbf{z}^{\mathrm{a}}-\mathbf{z}^{\mathrm{f}}\right)+\mathbf{G}^{\mathrm{T}}\mathbf{C}_{dd}^{-1}\left(\mathbf{g}(\mathbf{z}^{\mathrm{f}})+\mathbf{G}(\mathbf{z}^{\mathrm{a}}-\mathbf{z}^{\mathrm{f}})-\mathbf{d}\right)=0, \tag{89}$$

or

$$\left(\mathbf{C}_{zz}^{-1}+\mathbf{G}^{\mathrm{T}}\mathbf{C}_{dd}^{-1}\mathbf{G}\right)\left(\mathbf{z}^{\mathrm{a}}-\mathbf{z}^{\mathrm{f}}\right)=\mathbf{G}^{\mathrm{T}}\mathbf{C}_{dd}^{-1}\left(\mathbf{d}-\mathbf{g}(\mathbf{z}^{\mathrm{f}})\right). \tag{90}$$

which we can solve for $\mathbf{z}^{\mathrm{a}}-\mathbf{z}^{\mathrm{f}}$ and get

$$\mathbf{z}^{\mathrm{a}}=\mathbf{z}^{\mathrm{f}}+\left(\mathbf{C}_{zz}^{-1}+\mathbf{G}^{\mathrm{T}}\mathbf{C}_{dd}^{-1}\mathbf{G}\right)^{-1}\mathbf{G}^{\mathrm{T}}\mathbf{C}_{dd}^{-1}\left(\mathbf{d}-\mathbf{g}(\mathbf{z}^{\mathrm{f}})\right). \tag{91}$$

Alternative form using Woodbury

Woodbury corollaries

$$\left(\mathbf{C}^{-1} + \mathbf{G}^{\mathrm{T}}\mathbf{D}^{-1}\mathbf{G}\right)^{-1} = \mathbf{C} - \mathbf{C}\mathbf{G}^{\mathrm{T}}(\mathbf{G}\mathbf{C}\mathbf{G}^{\mathrm{T}} + \mathbf{D})^{-1}\mathbf{G}\mathbf{C}, \tag{92}$$

$$\left(\mathbf{G}^{\mathrm{T}}\mathbf{D}^{-1}\mathbf{G} + \mathbf{C}^{-1}\right)^{-1}\mathbf{G}^{\mathrm{T}}\mathbf{D}^{-1} = \mathbf{C}\mathbf{G}^{\mathrm{T}}\left(\mathbf{G}\mathbf{C}\mathbf{G}^{\mathrm{T}} + \mathbf{D}\right)^{-1}, \tag{93}$$

Using Eq. (93) we rewrite Eq. (91) as

Closed form solution by linearization

$$\mathbf{z}^{\mathrm{a}} = \mathbf{z}^{\mathrm{f}} + \mathbf{C}_{zz}\mathbf{G}^{\mathrm{T}}\left(\mathbf{G}\mathbf{C}_{zz}\mathbf{G}^{\mathrm{T}} + \mathbf{C}_{dd}\right)^{-1}\left(\mathbf{d} - \mathbf{g}\left(\mathbf{z}^{\mathrm{f}}\right)\right). \tag{94}$$

Predicted measurements

Assume linear model and measurement operator $\mathbf{G} = \mathbf{H}\boldsymbol{\mathcal{M}}$ and a state vector $\mathbf{z} = \mathbf{x}_0$.

The predicted measurements then become

$$\mathbf{y} = \mathbf{G}\mathbf{z} = \mathbf{H}\begin{pmatrix} \mathbf{z} \\ \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_K \end{pmatrix} = \mathbf{H}\begin{pmatrix} \mathbf{z} \\ \mathbf{M}_1\mathbf{z} \\ \vdots \\ \mathbf{M}_K \ldots \mathbf{M}_1\mathbf{z} \end{pmatrix} = \mathbf{H}\begin{pmatrix} \mathbf{I} \\ \mathbf{M}_1 \\ \vdots \\ \mathbf{M}_K \ldots \mathbf{M}_1 \end{pmatrix}\mathbf{z} = \mathbf{H}\boldsymbol{\mathcal{M}}\mathbf{z}, \tag{95}$$

Nonliner predicted measurements

Assume nonlinear model and measurement operator $\mathbf{g}(\mathbf{z}) = \mathbf{h}(\mathbf{m}(\mathbf{z}))$ and a state vector $\mathbf{z} = \mathbf{x}_0$.

The predicted measurements then becomes

$$\mathbf{y} = \mathbf{g}(\mathbf{z}) = \mathbf{h}\begin{pmatrix} \mathbf{z} \\ \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_K \end{pmatrix} = \mathbf{h}\begin{pmatrix} \mathbf{z} \\ \mathbf{m}_1(\mathbf{z}) \\ \vdots \\ \mathbf{m}_K(\cdots(\mathbf{m}_2(\mathbf{m}_1(\mathbf{z}))\cdots) \end{pmatrix} = \mathbf{h}(\mathbf{m}(\mathbf{z})) \tag{96}$$

Thus, we can compute the update of $\mathbf{z} = \mathbf{x}_0$ using data distributed over the assimilation window.

Update over the assimilation window

Multiply Eq. (94) by $\mathcal{M}$ to get

$$\mathcal{M}\mathbf{z}^{\mathrm{a}} = \mathcal{M}\mathbf{z}^{\mathrm{f}} + \mathcal{M}\mathbf{C}_{zz}\mathcal{M}^{\mathrm{T}}\mathbf{H}^{\mathrm{T}}\left(\mathbf{G}\mathbf{C}_{zz}\mathbf{G}^{\mathrm{T}} + \mathbf{C}_{dd}\right)^{-1}\left(\mathbf{d} - \mathbf{g}\left(\mathbf{z}^{\mathrm{f}}\right)\right). \tag{97}$$

We can write this equation as

$$\begin{pmatrix} \mathbf{z}^{\mathrm{a}} \\ \mathbf{x}_1^{\mathrm{a}} \\ \vdots \\ \mathbf{x}_K^{\mathrm{a}} \end{pmatrix} = \begin{pmatrix} \mathbf{z}^{\mathrm{f}} \\ \mathbf{x}_1^{\mathrm{f}} \\ \vdots \\ \mathbf{x}_K^{\mathrm{f}} \end{pmatrix} + \begin{pmatrix} \mathbf{C}_{zz} & \dots & \mathbf{C}_{zx_K} \\ \mathbf{C}_{x_1 z} & \dots & \mathbf{C}_{x_1 x_K} \\ \vdots & \ddots & \vdots \\ \mathbf{C}_{x_K z} & \dots & \mathbf{C}_{x_K x_K} \end{pmatrix} \mathbf{H}^{\mathrm{T}}\left(\mathbf{G}\mathbf{C}_{zz}\mathbf{G}^{\mathrm{T}} + \mathbf{C}_{dd}\right)^{-1}\left(\mathbf{d} - \mathbf{g}\left(\mathbf{z}^{\mathrm{f}}\right)\right). \tag{98}$$

This gives a smoother update of the model solution over the whole assimilation window

If we are only interested in the solution at the time t_K, we can compute

$$\begin{aligned} \mathbf{x}_K^{\mathrm{a}} &= \mathbf{x}_K^{\mathrm{f}} + \begin{pmatrix} \mathbf{C}_{x_K z} & \dots & \mathbf{C}_{x_K x_K} \end{pmatrix} \mathbf{H}^{\mathrm{T}}\left(\mathbf{G}\mathbf{C}_{zz}\mathbf{G}^{\mathrm{T}} + \mathbf{C}_{dd}\right)^{-1}\left(\mathbf{d} - \mathbf{g}\left(\mathbf{z}^{\mathrm{f}}\right)\right) && (99) \\ &= \mathbf{x}_K^{\mathrm{f}} + \mathbf{C}_{x_K y}\left(\mathbf{C}_{yy} + \mathbf{C}_{dd}\right)^{-1}\left(\mathbf{d} - \mathbf{g}\left(\mathbf{z}^{\mathrm{f}}\right)\right). && (100) \end{aligned}$$

Kalman Filter (KF) state update

- Linear dynamics and measurement operator.

$$\mathcal{J}(\mathbf{z}) = \frac{1}{2}\left(\mathbf{z} - \mathbf{z}^{\mathrm{f}}\right)^{\mathrm{T}} \mathbf{C}_{zz}^{-1} \left(\mathbf{z} - \mathbf{z}^{\mathrm{f}}\right) + \frac{1}{2}\left(\mathbf{H}\mathbf{z} - \mathbf{d}\right)^{\mathrm{T}} \mathbf{C}_{dd}^{-1} \left(\mathbf{H}\mathbf{z} - \mathbf{d}\right). \tag{101}$$

Gradient of the cost function equal to zero,

$$\mathbf{C}_{zz}^{-1}\left(\mathbf{z}^{\mathrm{a}} - \mathbf{z}^{\mathrm{f}}\right) + \mathbf{H}^{\mathrm{T}} \mathbf{C}_{dd}^{-1} \left(\mathbf{H}\mathbf{z}^{\mathrm{a}} - \mathbf{d}\right) = 0, \tag{102}$$

The Kalman filter state update

$$\mathbf{z}^{\mathrm{a}} = \mathbf{z}^{\mathrm{f}} + \mathbf{C}_{zz}\mathbf{H}^{\mathrm{T}}\left(\mathbf{H}\mathbf{C}_{zz}\mathbf{H}^{\mathrm{T}} + \mathbf{C}_{dd}\right)^{-1}\left(\mathbf{d} - \mathbf{H}\mathbf{z}^{\mathrm{f}}\right), \tag{103}$$

Kalman Filter (KF) error covariance update

Cost function

$$\mathcal{J}(\mathbf{z}) = \frac{1}{2}\left(\mathbf{z} - \mathbf{z}^{\mathrm{f}}\right)^{\mathrm{T}} \mathbf{C}_{zz}^{-1} \left(\mathbf{z} - \mathbf{z}^{\mathrm{f}}\right) + \frac{1}{2}\left(\mathbf{H}\mathbf{z} - \mathbf{d}\right)^{\mathrm{T}} \mathbf{C}_{dd}^{-1} \left(\mathbf{H}\mathbf{z} - \mathbf{d}\right). \tag{104}$$

Hessian

$$\nabla_z \nabla_z \mathcal{J}(\mathbf{z}) = \mathbf{C}_{zz}^{-1} + \mathbf{H}\mathbf{C}_{dd}^{-1}\mathbf{H}^{\mathrm{T}}. \tag{105}$$

The posterior is Gaussian, with covariance matrix $\mathbf{C}_{zz}^{\mathrm{a}}$, hence

$$\mathcal{J}(\mathbf{z}) = \frac{1}{2}\left(\mathbf{z} - \mathbf{z}^{\mathrm{a}}\right)^{\mathrm{T}} \left(\mathbf{C}_{zz}^{\mathrm{a}}\right)^{-1} \left(\mathbf{z} - \mathbf{z}^{\mathrm{a}}\right) + \text{constant} \tag{106}$$

Second derivative

$$\nabla_z \nabla_z \mathcal{J}(\mathbf{z}) = \left(\mathbf{C}_{zz}^{\mathrm{a}}\right)^{-1}. \tag{107}$$

Since the two expressions for the Hessian must be the same, we find for the posterior covariance

$$\left(\mathbf{C}_{zz}^{\mathrm{a}}\right)^{-1} = \mathbf{C}_{zz}^{-1} + \mathbf{H}\mathbf{C}_{dd}^{-1}\mathbf{H}^{\mathrm{T}}. \tag{108}$$

Kalman Filter error covariance update equation

Rewrite using the matrix identity (92) to find

The Kalman filter error-covariance update

$$\mathbf{C}^{\mathrm{a}}_{zz} = \mathbf{C}_{zz} - \mathbf{C}_{zz}\mathbf{H}^{\mathrm{T}}\left(\mathbf{H}\mathbf{C}_{zz}\mathbf{H}^{\mathrm{T}} + \mathbf{C}_{dd}\right)^{-1}\mathbf{H}\mathbf{C}_{zz}. \tag{109}$$

NORCE

Kalman Filter (KF)

Forward model

$$
\begin{aligned}
\mathbf{z}_{k+1} &= \mathbf{M}\mathbf{z}_k, && (110)\\
\mathbf{C}_{zz,k+1} &= \mathbf{M}\mathbf{C}_{zz,k}\mathbf{M}^\mathrm{T} + \mathbf{C}_{qq}. && (111)
\end{aligned}
$$

Update equations

$$
\begin{aligned}
\mathbf{z}^\mathrm{a} &= \mathbf{z}^\mathrm{f} + \mathbf{C}_{zz}\mathbf{H}^\mathrm{T}\left(\mathbf{H}\mathbf{C}_{zz}\mathbf{H}^\mathrm{T} + \mathbf{C}_{dd}\right)^{-1}\left(\mathbf{d} - \mathbf{H}\mathbf{z}^\mathrm{f}\right), && (112)\\
\mathbf{C}_{zz}^\mathrm{a} &= \mathbf{C}_{zz} - \mathbf{C}_{zz}\mathbf{H}^\mathrm{T}\left(\mathbf{H}\mathbf{C}_{zz}\mathbf{H}^\mathrm{T} + \mathbf{C}_{dd}\right)^{-1}\mathbf{H}\mathbf{C}_{zz}. && (113)
\end{aligned}
$$

One often defines the “Kalman gain matrix”

$$
\mathbf{K} = \mathbf{C}_{zz}\mathbf{H}^\mathrm{T}\left(\mathbf{H}\mathbf{C}_{zz}\mathbf{H}^\mathrm{T} + \mathbf{C}_{dd}\right)^{-1}, \tag{114}
$$

Extension to nonlinear problem

- Allows for nonlinear dynamics and measurement operator.

Gradient Eq. (??) becomes

$$\mathbf{C}_{zz}^{-1}\left(\mathbf{z}^{\mathrm{a}} - \mathbf{z}^{\mathrm{f}}\right) + \mathbf{H}^{\mathrm{T}}\,\mathbf{C}_{dd}^{-1}\left(\mathbf{h}(\mathbf{z}^{\mathrm{f}}) + \mathbf{H}(\mathbf{z}^{\mathrm{a}} - \mathbf{z}^{\mathrm{f}}) - \mathbf{d}\right). \tag{115}$$

Explicit solution

$$\mathbf{z}^{\mathrm{a}} = \mathbf{z}^{\mathrm{f}} + \mathbf{C}_{zz}\mathbf{H}^{\mathrm{T}}\left(\mathbf{H}\mathbf{C}_{zz}\mathbf{H}^{\mathrm{T}} + \mathbf{C}_{dd}\right)^{-1}\left(\mathbf{d} - \mathbf{h}\left(\mathbf{z}^{\mathrm{f}}\right)\right). \tag{116}$$

Extended Kalman Filter

$$\mathbf{z}_{k+1} = \mathbf{m}\big(\mathbf{z}_k\big) \tag{117}$$

$$\mathbf{C}_{zz,k+1} = \mathbf{M}_k \mathbf{C}_{zz,k} \mathbf{M}_k^{\mathrm{T}} + \mathbf{C}_{qq}. \tag{118}$$

Update equations

$$\mathbf{z}^{\mathrm{a}} = \mathbf{z}^{\mathrm{f}} + \mathbf{C}_{zz}\mathbf{H}^{\mathrm{T}}\Big(\mathbf{H}\mathbf{C}_{zz}\mathbf{H}^{\mathrm{T}} + \mathbf{C}_{dd}\Big)^{-1}\Big(\mathbf{d} - \mathbf{h}\big(\mathbf{z}^{\mathrm{f}}\big)\Big). \tag{119}$$

$$\mathbf{C}_{zz}^{\mathrm{a}} = \mathbf{C}_{zz} - \mathbf{C}_{zz}\mathbf{H}^{\mathrm{T}}\Big(\mathbf{H}\mathbf{C}_{zz}\mathbf{H}^{\mathrm{T}} + \mathbf{C}_{dd}\Big)^{-1}\mathbf{H}\mathbf{C}_{zz}. \tag{120}$$

Summary of KF, EKF, and 3DVar

- Used for recursive data assimilation.
- 3DVar assumes steady state $\mathbf{C}_{zz}$ and neglects uncertainty propagation.
- 3DVar minimizes full cost function using Gauss-Newton.
- KF applies for linear models and measurement operators.
- KF propagates uncertainty by solving covariance equation.
- KF solves for explicit update from the update equation.
- EKF linearizes model to obtain approximate uncertainty propagation.
- EKF linearizes measurement operator to use the explicit KF update equations.
- The storage and solution of the covariance equation is too computationally expensive.
- Optimal interpolation resembles 3DVar's steady state $\mathbf{C}_{zz}$ but uses KF update equation.

Randomized-maximum-likelihood sampling

Randomized Maximum Likelihood sampling

Approximation 6 (RML sampling)

In the weakly nonlinear case, we can approximately sample the posterior pdf with Gaussian priors by minimizing the ensemble of cost functions defined by Eq. (122).

ps: it's really Randomized MAP sampling, or rather just approximate sampling of the posterior pdf.

RML minimizes an ensemble of cost functions

We define realizations

$$\mathbf{z}_j^{\mathrm{f}} \leftarrow \mathcal{N}(\mathbf{z}^{\mathrm{f}}, \mathbf{C}_{zz}) \quad \text{and} \quad \mathbf{d}_j \leftarrow \mathcal{N}(\mathbf{d}, \mathbf{C}_{dd}) \tag{121}$$

Ensemble of cost functions

$$\mathcal{J}(\mathbf{z}_j) = \frac{1}{2}\left(\mathbf{z}_j - \mathbf{z}_j^{\mathrm{f}}\right)^{\mathrm{T}} \mathbf{C}_{zz}^{-1} \left(\mathbf{z}_j - \mathbf{z}_j^{\mathrm{f}}\right) + \frac{1}{2}\left(\mathbf{g}(\mathbf{z}_j) - \mathbf{d}_j\right)^{\mathrm{T}} \mathbf{C}_{dd}^{-1} \left(\mathbf{g}(\mathbf{z}_j) - \mathbf{d}_j\right), \tag{122}$$

Ensemble of gradients set to zero

$$\mathbf{C}_{zz}^{-1}\left(\mathbf{z}_j - \mathbf{z}_j^{\mathrm{f}}\right) + \nabla_{\mathbf{z}} \mathbf{g}(\mathbf{z}_j) \mathbf{C}_{dd}^{-1}\left(\mathbf{g}(\mathbf{z}_j) - \mathbf{d}_j\right) = 0. \tag{123}$$

Thus, we must solve N independent minimizations.

Solutions methods using the tangent linear model $\mathbf{G}$

Ensemble of incremental 4DVars

$$\mathcal{J}(\delta \mathbf{z}_j) = \frac{1}{2}\left(\delta \mathbf{z}_j - \boldsymbol{\xi}_j^i\right)^{\mathrm{T}} \mathbf{C}_{zz}^{-1}\left(\delta \mathbf{z}_j - \boldsymbol{\xi}_j^i\right) + \frac{1}{2}\left(\mathbf{G}_j^i \delta \mathbf{z}_j - \boldsymbol{\eta}_j^i\right)^{\mathrm{T}} \mathbf{C}_{dd}^{-1}\left(\mathbf{G}_j^i \delta \mathbf{z}_j - \boldsymbol{\eta}_j^i\right). \tag{124}$$

Ensemble of GN iterations

$$\mathbf{z}_j^{i+1} = \mathbf{z}_j^i - \gamma\left(\mathbf{C}_{zz}^{-1} + \mathbf{G}_j^{i\,\mathrm{T}} \mathbf{C}_{dd}^{-1} \mathbf{G}_j^i\right)^{-1}\left(\mathbf{C}_{zz}^{-1}\left(\mathbf{z}_j^i - \mathbf{z}_j^{\mathrm{f}}\right) + \mathbf{G}_j^{i\,\mathrm{T}} \mathbf{C}_{dd}^{-1}\left(\mathbf{g}\left(\mathbf{z}_j^i\right) - \mathbf{d}_j\right)\right), \tag{125}$$

The linear Approximation 5 leads to an Ensemble of Kalman-filter updates

$$\mathbf{z}_j^{\mathrm{a}} = \mathbf{z}_j^{\mathrm{f}} + \mathbf{C}_{zz}\mathbf{G}_j^{\mathrm{T}}\left(\mathbf{G}_j \mathbf{C}_{zz} \mathbf{G}_j^{\mathrm{T}} + \mathbf{C}_{dd}\right)^{-1}\left(\mathbf{d}_j - \mathbf{g}\left(\mathbf{z}_j^{\mathrm{f}}\right)\right). \tag{126}$$

Comments

- SC-4DVar and WC-4DVar solves for the MAP estimate over the assimilation window.
- RML sampling using (En)SC-4DVar and (En)WC-4DVar would aim to sample the posterior.
- It is possible to propagate error statistics using ensemble integrations.
- We could have a consistent method using ensemble “background” covariances.
- Still uses the tangent linear and adjoint models.
- What is the benefit of computing update over a finite data-assimilation window?

Replacing the adjoints with an averaged model sensitivity

Use an averaged model sensitivity to avoid adjoints

Approximation 7 (Best-fit averaged model sensitivity)

Interpret $\mathbf{G}_j$ in Eq. (126) and $\mathbf{G}_j^i$ in Eq. (125) as the sensitivity matrix in linear regression and represent them using the definition

$$\mathbf{G}_j \approx \mathbf{G} \triangleq \mathbf{C}_{yz}\mathbf{C}_{zz}^{-1}. \tag{127}$$

We approximate the individual model sensitivities with a common averaged sensitivity used for all realizations.

Explanation of the linear regression formula

Define Taylor expansion of $g(x)$ around the ensemble mean

$$g\left(x\right) \approx g\left(\overline{x}\right) + g'\left(\overline{x}\right)\left(x - \overline{x}\right). \tag{128}$$

$$\begin{aligned}
C^{\mathrm{e}}_{xy} &= \overline{\left(x - \overline{x}\right)\left(y - \overline{y}\right)} \\
&= \overline{\left(x - \overline{x}\right)\left(g\left(x\right) - \overline{g\left(x\right)}\right)} \\
&\approx \overline{\left(x - \overline{x}\right)\left(g\left(\overline{x}\right) + g'\left(\overline{x}\right)\left(x - \overline{x}\right) - \overline{\left(g\left(\overline{x}\right) + g'\left(\overline{x}\right)\left(x - \overline{x}\right)\right)}\right)} \\
&= g'\left(\overline{x}\right)\overline{\left(x - \overline{x}\right)^2} \\
&= g'\left(\overline{x}\right) C^{\mathrm{e}}_{xx},
\end{aligned} \tag{129}$$

Gauss-Newton iterations with averaged model sensitivity

Rewrite the Gauss-Newton iteration in Eq. (125) as

$$
\begin{aligned}
\mathbf{z}_j^{i+1} &= \mathbf{z}_j^i - \gamma\left(\mathbf{C}_{zz}^{-1} + \mathbf{G}_j^{i\mathrm{T}}\mathbf{C}_{dd}^{-1}\mathbf{G}_j^i\right)^{-1}\left(\mathbf{C}_{zz}^{-1}\left(\mathbf{z}_j^i - \mathbf{z}_j^{\mathrm{f}}\right) + \mathbf{G}_j^{i\mathrm{T}}\mathbf{C}_{dd}^{-1}\left(\mathbf{g}\left(\mathbf{z}_j^i\right) - \mathbf{d}_j\right)\right), && (130)\\
&\approx \mathbf{z}_j^i - \gamma\left(\mathbf{C}_{zz}^{-1} + \mathbf{G}^{i\mathrm{T}}\mathbf{C}_{dd}^{-1}\mathbf{G}^i\right)^{-1}\left(\mathbf{C}_{zz}^{-1}\left(\mathbf{z}_j^i - \mathbf{z}_j^{\mathrm{f}}\right) + \mathbf{G}^{i\mathrm{T}}\mathbf{C}_{dd}^{-1}\left(\mathbf{g}(\mathbf{z}_i) - \mathbf{d}_j\right)\right) && (131)\\
&= \mathbf{z}_j^i - \gamma\left(\mathbf{z}_j^i - \mathbf{z}_j^{\mathrm{f}}\right) + \gamma\mathbf{C}_{zz}\mathbf{G}^{i\mathrm{T}}\left(\mathbf{G}^i\mathbf{C}_{zz}\mathbf{G}^{i\mathrm{T}} + \mathbf{C}_{dd}\right)^{-1}\left(\mathbf{G}^i\left(\mathbf{z}_j^i - \mathbf{z}_j^{\mathrm{f}}\right) - \left(\mathbf{g}\left(\mathbf{z}_j^i\right) - \mathbf{d}_j\right)\right), && (132)
\end{aligned}
$$

where we have used the corollaries from Eqs. (92) and (93).

Expression in terms of covariances

We have

$$\mathbf{G}\mathbf{C}_{zz} = \mathbf{C}_{yz}, \tag{133}$$

$$\mathbf{G}\mathbf{C}_{zz}\mathbf{G}^{\mathrm{T}} = \mathbf{C}_{yz}\mathbf{C}_{zz}^{-1}\mathbf{C}_{zy} \neq \mathbf{C}_{yy}. \tag{134}$$

Gauss-Newton Eq. (132)

$$\mathbf{z}_j^{i+1} = \mathbf{z}_j^i - \gamma\left(\mathbf{z}_j^i - \mathbf{z}_j^{\mathrm{f}}\right) + \gamma\mathbf{C}_{zy}\left(\mathbf{C}_{yz}\mathbf{C}_{zz}^{-1}\mathbf{C}_{zy} + \mathbf{C}_{dd}\right)^{-1}\left(\mathbf{C}_{yz}\mathbf{C}_{zz}^{-1}\left(\mathbf{z}_j^i - \mathbf{z}_j^{\mathrm{f}}\right) - \left(\mathbf{g}(\mathbf{z}_j^i) - \mathbf{d}_j\right)\right), \tag{135}$$

EKF update Eq. (126)

$$\mathbf{z}_j = \mathbf{z}_j^{\mathrm{f}} + \mathbf{C}_{zy}\left(\mathbf{C}_{yz}\mathbf{C}_{zz}^{-1}\mathbf{C}_{zy} + \mathbf{C}_{dd}\right)^{-1}\left(\mathbf{d}_j - \mathbf{g}\left(\mathbf{z}_j^{\mathrm{f}}\right)\right). \tag{136}$$

First GN-step equals EKF update

First step of Gauss-Newton Eq. (132) with $\gamma = 1$ and $i = 1$:

$$\mathbf{z}_j^{i+1} = \mathbf{z}_j^i - \left(\mathbf{z}_j^i - \mathbf{z}_j^{\mathrm{f}}\right) + \mathbf{C}_{zy}\left(\mathbf{C}_{yz}\mathbf{C}_{zz}^{-1}\mathbf{C}_{zy} + \mathbf{C}_{dd}\right)^{-1}\left(\mathbf{C}_{yz}\mathbf{C}_{zz}^{-1}\left(\mathbf{z}_j^i - \mathbf{z}_j^{\mathrm{f}}\right) - \left(\mathbf{g}(\mathbf{z}_j^i) - \mathbf{d}_j\right)\right), \tag{137}$$

$$= \mathbf{z}_j^{\mathrm{f}} + \mathbf{C}_{zy}\left(\mathbf{C}_{yz}\mathbf{C}_{zz}^{-1}\mathbf{C}_{zy} + \mathbf{C}_{dd}\right)^{-1}\left(\mathbf{d}_j - \mathbf{g}\left(\mathbf{z}_j^{\mathrm{f}}\right)\right). \tag{138}$$

Ensemble methods

Ensemble representation of covariances

Approximation 8 (Ensemble approximation)

It is possible to approximately represent a covariance matrix by a low-rank ensemble of states with fewer realizations than the state dimension.

Ensemble representation of all covariances

Ensemble matrices

$$\mathbf{Z} = \left(\mathbf{z}_1,\ \mathbf{z}_2,\ \ldots\ ,\ \mathbf{z}_N\right), \tag{139}$$

$$\mathbf{D} = \left(\mathbf{d}_1,\ \mathbf{d}_2,\ \ldots\ ,\ \mathbf{d}_N\right), \tag{140}$$

$$\boldsymbol{\Upsilon} = \mathbf{g}(\mathbf{Z}). \tag{141}$$

Define the projection $\boldsymbol{\Pi} \in \Re^{N\times N}$

$$\boldsymbol{\Pi} = \left(\mathbf{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^{\mathrm{T}}\right) \Big/ \sqrt{N-1}. \tag{142}$$

Ensemble anomalies and covariances

$$\mathbf{A} = \mathbf{Z}\boldsymbol{\Pi}, \qquad \mathbf{C}_{zz} \approx \overline{\mathbf{C}}_{zz} = \mathbf{A}\mathbf{A}^{\mathrm{T}}, \tag{143}$$

$$\mathbf{E} = \mathbf{D}\boldsymbol{\Pi}, \qquad \mathbf{C}_{dd} \approx \overline{\mathbf{C}}_{dd} = \mathbf{E}\mathbf{E}^{\mathrm{T}}, \tag{144}$$

$$\mathbf{Y} = \boldsymbol{\Upsilon}\boldsymbol{\Pi}, \qquad \mathbf{C}_{zy} \approx \overline{\mathbf{C}}_{zy} = \mathbf{A}\mathbf{Y}^{\mathrm{T}}. \tag{145}$$

Ensemble Kalman Filter (EnKF) update

Identical to one iteration of Subspace EnRML with step length $\gamma = 1.0$.

$$\mathbf{Z}^{\mathrm{a}} = \mathbf{Z}^{\mathrm{f}} + \mathbf{A}\mathbf{Y}^{\mathrm{T}}\left(\mathbf{Y}\mathbf{Y}^{\mathrm{T}} + \mathbf{E}\mathbf{E}^{\mathrm{T}}\right)^{-1}\left(\mathbf{D} - \mathbf{g}\left(\mathbf{Z}^{\mathrm{f}}\right)\right) \tag{146}$$

Alternative interpretation using

$$\mathbf{W} = \mathbf{Y}^{\mathrm{T}}\left(\mathbf{Y}\mathbf{Y}^{\mathrm{T}} + \mathbf{E}\mathbf{E}^{\mathrm{T}}\right)^{-1}\left(\mathbf{D} - \mathbf{g}\left(\mathbf{Z}^{\mathrm{f}}\right)\right), \tag{147}$$

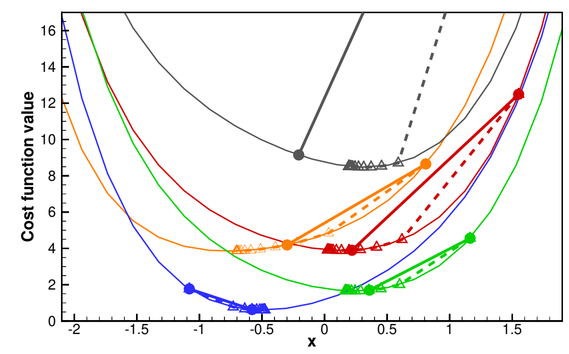
to get

$$\mathbf{Z}^{\mathrm{a}} = \mathbf{Z}^{\mathrm{f}}\left(\mathbf{I} + \mathbf{W}/\sqrt{N-1}\right) \tag{148}$$

But note that

$$\mathbf{Y} = \begin{cases} \mathbf{Y} & \text{for } n \geq N-1 \\ \mathbf{Y}\mathbf{A}^{\dagger}\mathbf{A} & \text{for } n < N-1. \end{cases} \tag{149}$$

EnKF and EnRML illustration: Non-linear model



- EnRML gets closer to minimum than the linear EnKF update.
- Approximate sampling of posterior pdf.

ESMDA uses tapering of likelihood

Approximate sampling of $f(\mathbf{x}|\mathbf{d})$ by gradually introducing the measurements (Neal, 1996)

$$
\begin{aligned}
f(\mathbf{x}|\mathbf{d}) &= f(\mathbf{d}|\mathbf{y}) f(\mathbf{x}) \\
&= f(\mathbf{d}|\mathbf{y})^{\left(\sum_{i=1}^{N} \frac{1}{\alpha_i}\right)} f(\mathbf{x}) \quad \text{with} \quad \sum_{i=1}^{N} \frac{1}{\alpha_i} = 1 \\
&= f(\mathbf{d}|\mathbf{y})^{\frac{1}{\alpha_N}} \cdots f(\mathbf{d}|\mathbf{y})^{\frac{1}{\alpha_2}} f(\mathbf{d}|\mathbf{y})^{\frac{1}{\alpha_1}} f(\mathbf{x})
\end{aligned}
$$

We compute N recursive ES/EnKF steps with "inflated" observation errors.

- Small updates reduce impact of the linear approximation.
- ESMDA is identical to ES in the linear case.
- Remember to resample measurement perturbations for each update step.

Some publications:

- Ensemble Randomized Maximum Likelihood EnRML (Chen and Oliver, 2013).
- Ensemble DA with multiple updates ESMDA (Emerick and Reynolds, 2013).
- Analysis of iterative ensemble smoothers (Evensen, 2018).
- IES with model errors (Evensen, 2019).
- Ensemble subspace RML (Evensen et al., 2019; Raanes et al., 2019).

Ensemble subspace RML implementation

We can use the algorithm for ES, ESMDA, and EnRML.

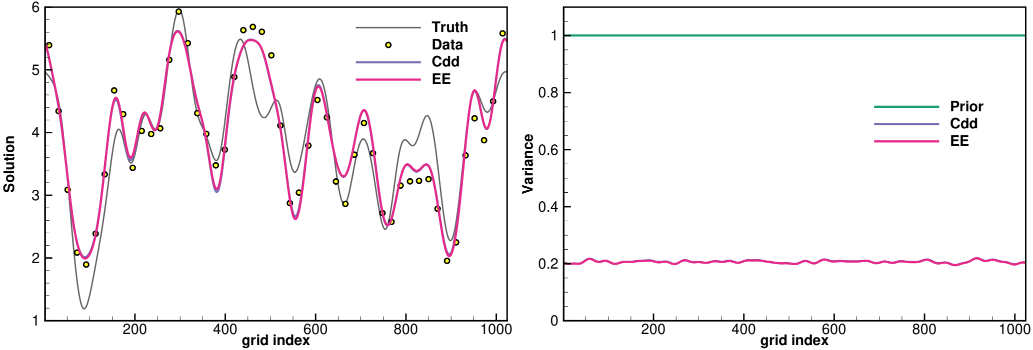
- For ES call once with step length $\gamma = 1$ and $\mathbf{W} = 0$.
- For ESMDA call in each MDA-step with resampled and inflated $\mathbf{D}$, $\gamma = 1$ and $\mathbf{W} = 0$.
- From Evensen et al. (2019); Raanes et al. (2019)

GPL licenced software

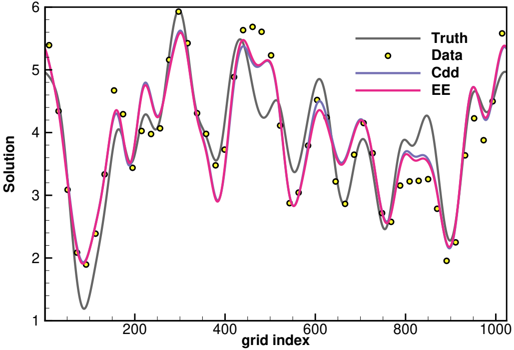
- `https://github.com/equinor/iterative_ensemble_smoother`
- `https://github.com/equinor/ert`
- `https://github.com/Python-Ensemble-Toolbox`

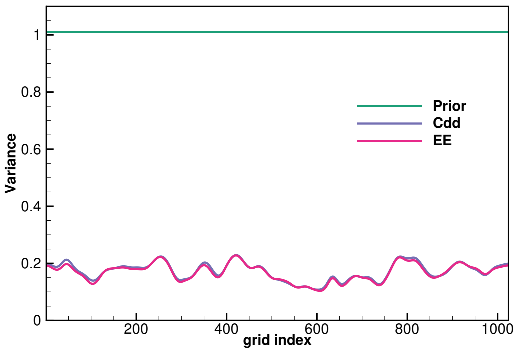
Linear EnKF update examples

EnKF update with large ensemble size $N = 2000$ and $m = 50$

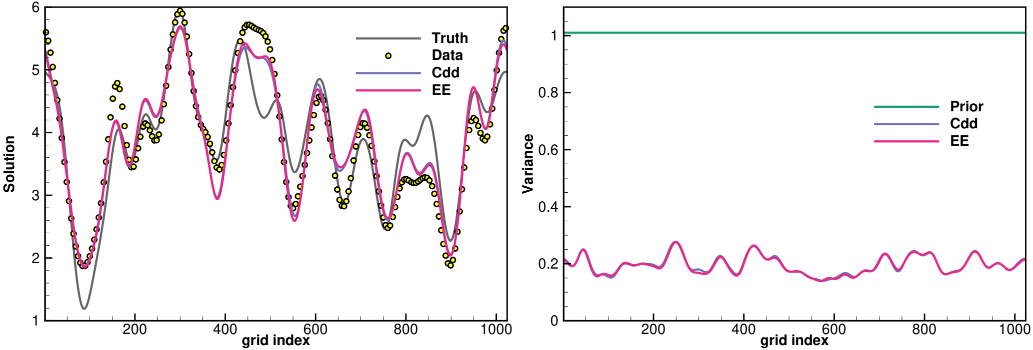


EnKF update with normal ensemble size $N = 100$ and $m = 50$



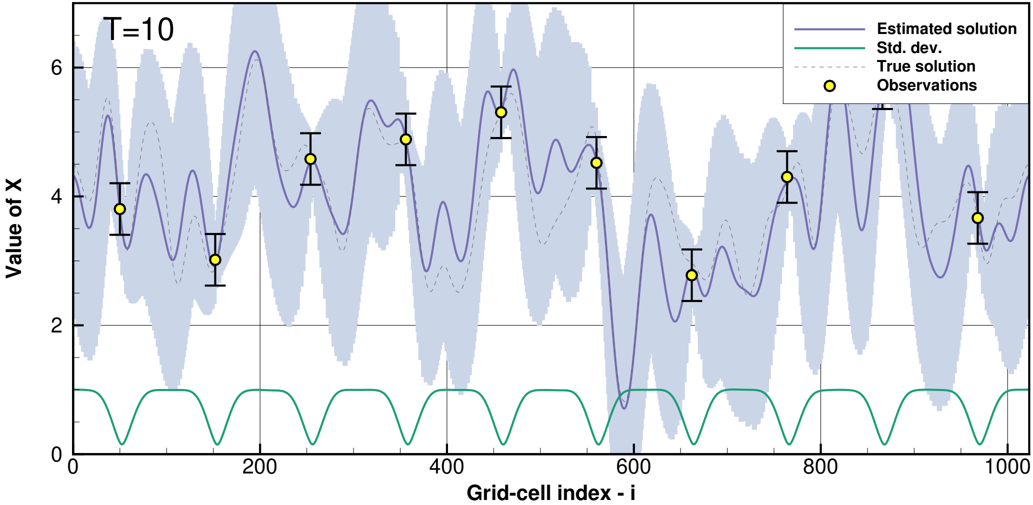


EnKF update with $N = 100$ and many measurements $m = 200$

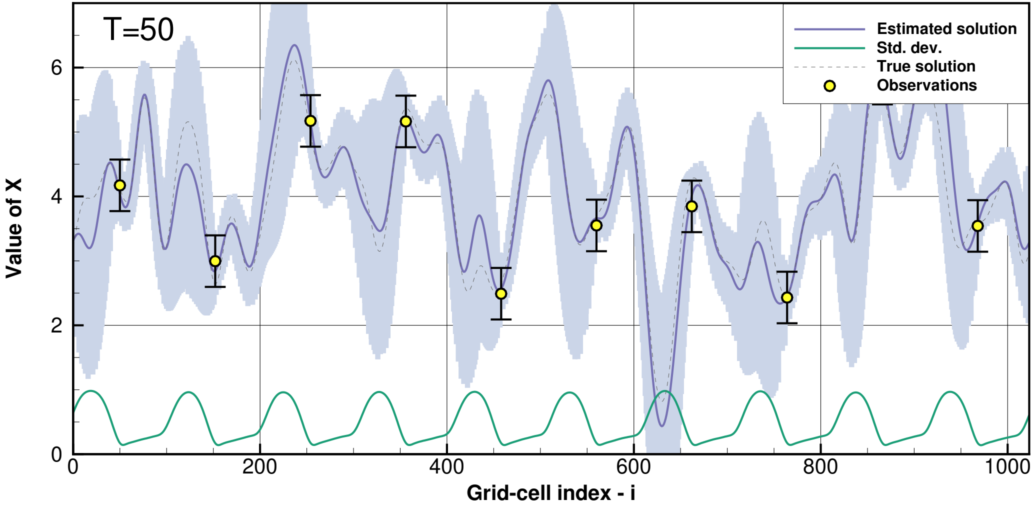


EnKF for an advection equation

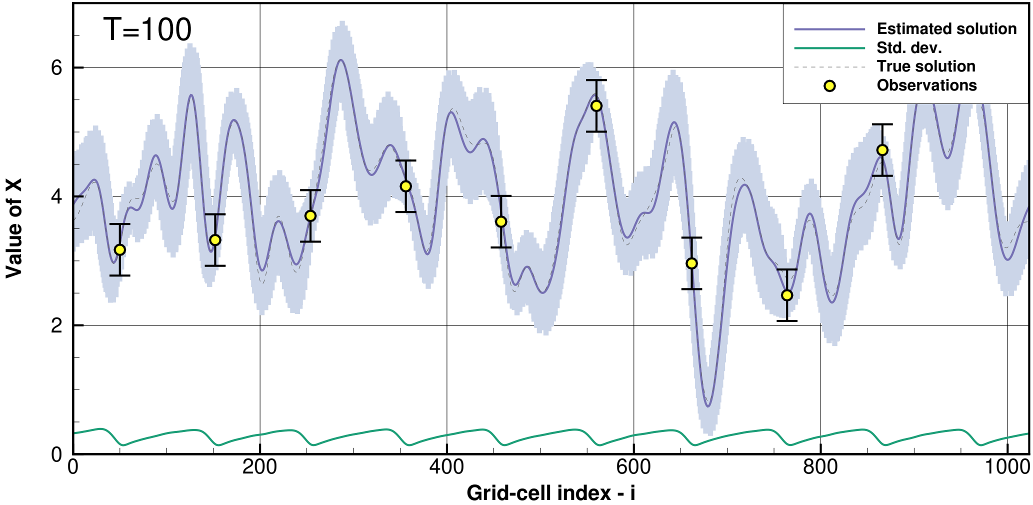
EnKF with the advection equation after two updates



EnKF with the advection equation after ten updates



EnKF with the advection equation after twenty updates



EnKF with the advection equation: Animation

Animation

EnKF with the Lorenz equations

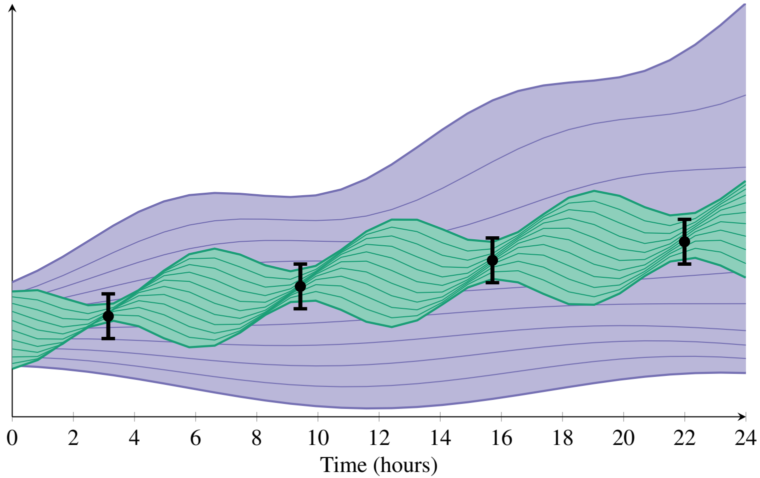
EnKF with the Lorenz model

$$\frac{\partial x}{\partial t} = \sigma(y - x), \tag{150}$$

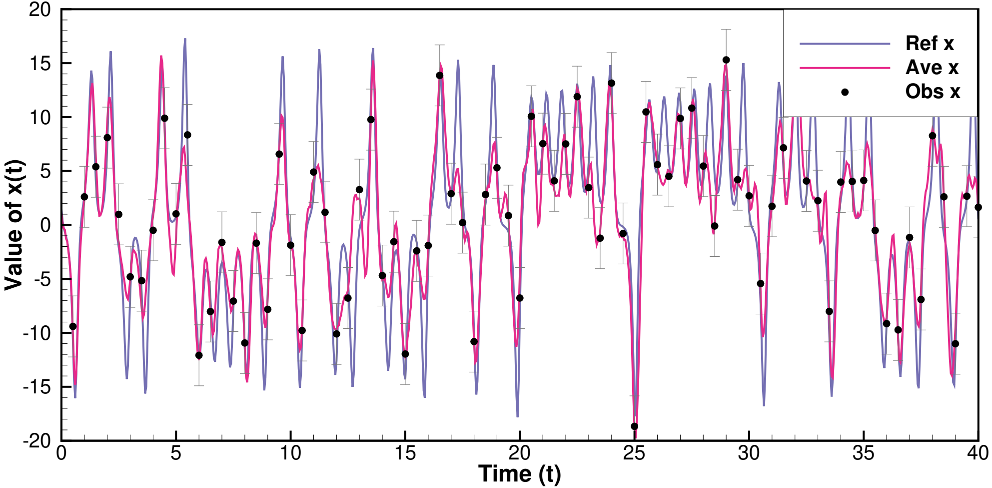
$$\frac{\partial y}{\partial t} = \rho x - y - xz, \tag{151}$$

$$\frac{\partial z}{\partial t} = xy - \beta z. \tag{152}$$

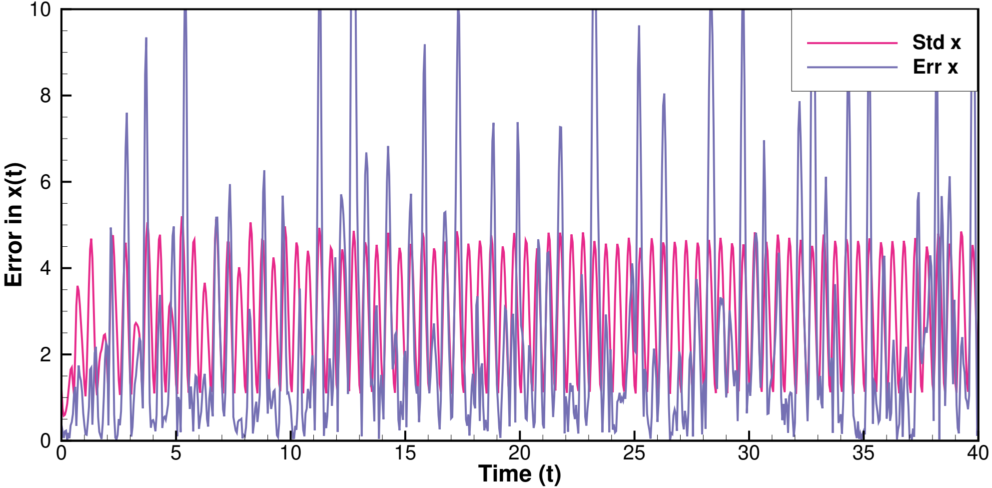
General smoother formulation



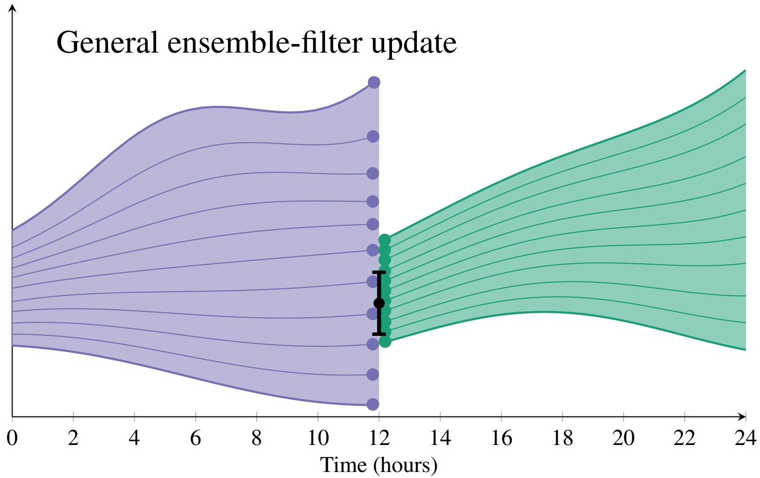
ES with the Lorenz model: estimate



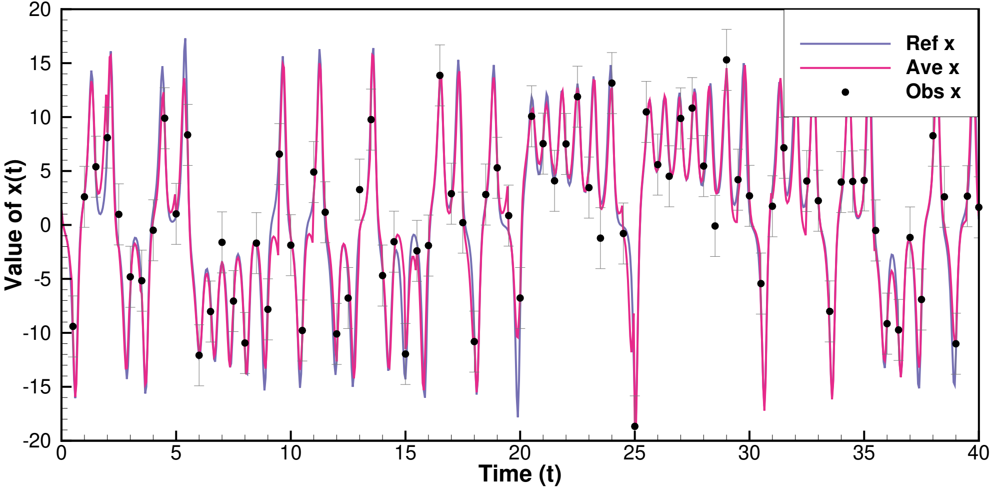
ES with the Lorenz model: error estimate



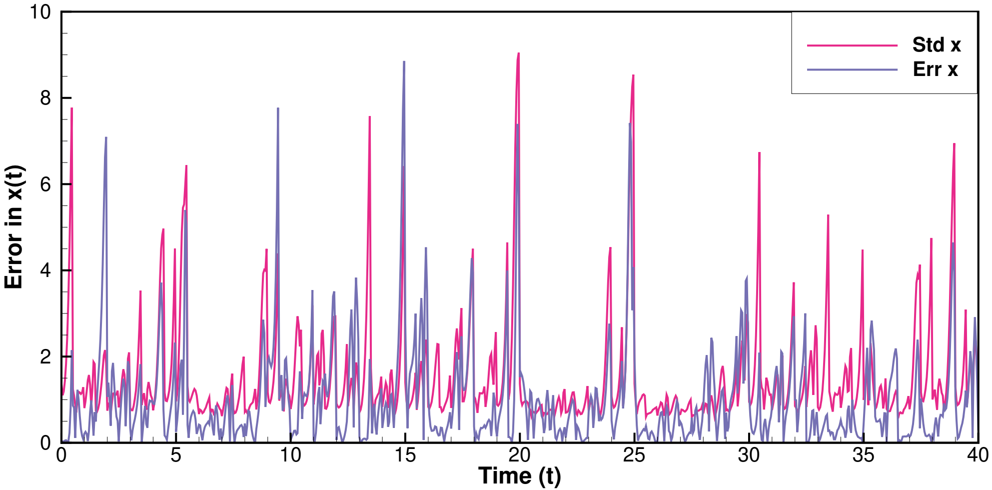
General filter formulation



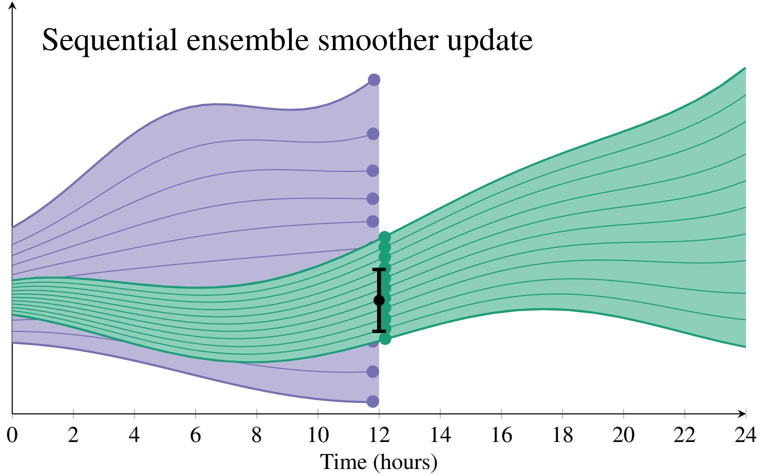
EnKF with the Lorenz model: estimate



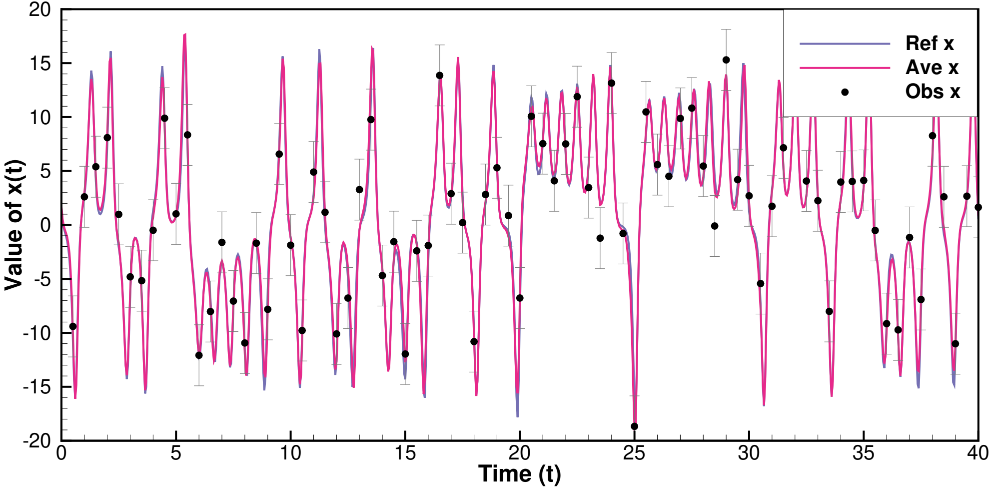
EnKF with the Lorenz model: error estimate



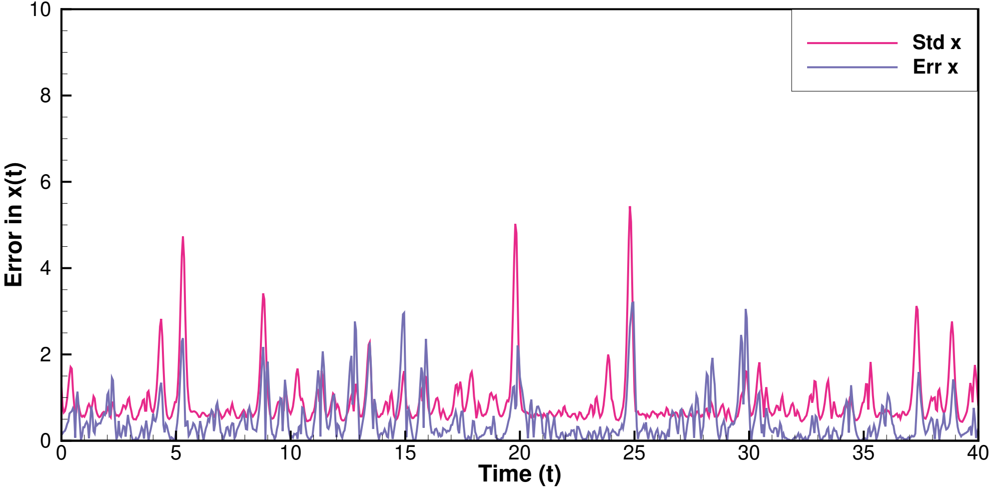
Recursive smoother formulation



EnKS with the Lorenz model: estimate



EnKS with the Lorenz model: error estimate



EnKF algorithms

Subspace EnKF update

subroutine EnKF_update$(\mathbf{Z}, \mathbf{D}, \boldsymbol{\Upsilon})$
Input: $\mathbf{Z} \in \Re^{n\times N}$ ▷ Prior state-vector ensemble
Input: $\mathbf{D} \in \Re^{m\times N}$ ▷ Perturbed measurements
Input: $\boldsymbol{\Upsilon} \in \Re^{m\times N}$ ▷ Predicted measurements
$\boldsymbol{\Pi} = \left(\mathbf{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^{\mathrm{T}}\right) / \sqrt{N-1}$ ▷ $\boldsymbol{\Pi} \in \Re^{N\times N}$
$\mathbf{E} = \mathbf{D}\boldsymbol{\Pi}$ ▷ $\mathbf{E} \in \Re^{m\times N}$
$\mathbf{Y} = \boldsymbol{\Upsilon}\boldsymbol{\Pi}$ ▷ $\mathbf{Y} \in \Re^{m\times N}$
if $n < N-1$ **then**
    $\mathbf{Y} = \mathbf{Y}\mathbf{A}^{\dagger}\mathbf{A}$
end if
$\mathbf{S} = \mathbf{Y}$ ▷ $\mathbf{S} \in \Re^{m\times N}$
$\mathbf{W} = \mathbf{S}^{\mathrm{T}}\left(\mathbf{S}\mathbf{S}^{\mathrm{T}} + \mathbf{E}\mathbf{E}^{\mathrm{T}}\right)^{-1}\left(\mathbf{D} - \boldsymbol{\Upsilon}\right)$ ▷ $\mathbf{W} \in \Re^{N\times N}$
$\mathbf{Z} \leftarrow \mathbf{Z}\left(\mathbf{I} + \mathbf{W}\big/\sqrt{N-1}\right)$ ▷ Update returned in $\mathbf{Z}$

Standard EnKF application

Input: $\mathbf{Z} \in \Re^{n \times N}$ ▷ Initial model state-vector ensemble
Input: $\mathbf{D}_l \in \Re^{m \times N}$ ▷ Perturbed measurements for each assimilation window
for $l = 1, \ldots$ **do** ▷ Loop over assimilation windows
    $\mathbf{X}_0 = \mathbf{Z}$
    for $k = 1, K$ **do** ▷ Ensemble integration
        $\mathbf{X}_k = \mathbf{m}(\mathbf{X}_{k-1})$
    end for
    $\mathbf{X} = [\mathbf{X}_0, \ldots, \mathbf{X}_K]$
    $\boldsymbol{\Upsilon} = \mathbf{h}(\mathbf{X})$ ▷ Predicted measurments
    call EnKF_update$(\mathbf{X}_K, \mathbf{D}_l, \boldsymbol{\Upsilon})$
    $\mathbf{Z} = \mathbf{X}_K$ ▷ Define state vector for next assimilation window
end for

EnKF updating $\mathbf{X}_0$

Input: $\mathbf{Z} \in \Re^{n \times N}$ ▷ Initial model state-vector ensemble
Input: $\mathbf{D}_l \in \Re^{m \times N}$ ▷ Perturbed measurements for each assimilation window
for $l = 1, \ldots$ **do** ▷ Loop over assimilation windows
  $\mathbf{X}_0 = \mathbf{Z}$
  for $k = 1, K$ **do** ▷ Ensemble integration
    $\mathbf{X}_k = \mathbf{m}(\mathbf{X}_{k-1})$
  end for
  $\mathbf{X} = [\mathbf{X}_0, \ldots, \mathbf{X}_K]$
  $\boldsymbol{\Upsilon} = \mathbf{h}(\mathbf{X})$ ▷ Predicted measurments
  call EnKF_update$(\mathbf{X}_0, \mathbf{D}_l, \boldsymbol{\Upsilon})$
  for $k = 1, K$ **do** ▷ Rerun ensemble integration
    $\mathbf{X}_k = \mathbf{m}(\mathbf{X}_{k-1})$
  end for
  $\mathbf{Z} = \mathbf{X}_K$ ▷ Define state vector for next assimilation window
end for

ES updating $\mathbf{X}$

Input: $\mathbf{Z} \in \Re^{n \times N}$ ▷ Initial model state-vector ensemble

Input: $\mathbf{D}_l \in \Re^{m \times N}$ ▷ Perturbed measurements for each assimilation window

for $l = 1, \ldots$ **do** ▷ Loop over assimilation windows

    $\mathbf{X}_0 = \mathbf{Z}$

    for $k = 1, K$ **do** ▷ Ensemble integration

        $\mathbf{X}_k = \mathbf{m}(\mathbf{X}_{k-1})$

    end for

    $\mathbf{X} = [\mathbf{X}_0, \ldots, \mathbf{X}_K]$

    $\boldsymbol{\Upsilon} = \mathbf{h}(\mathbf{X})$ ▷ Predicted measurments

    call EnKF_update$(\mathbf{X}, \mathbf{D}_l, \boldsymbol{\Upsilon})$

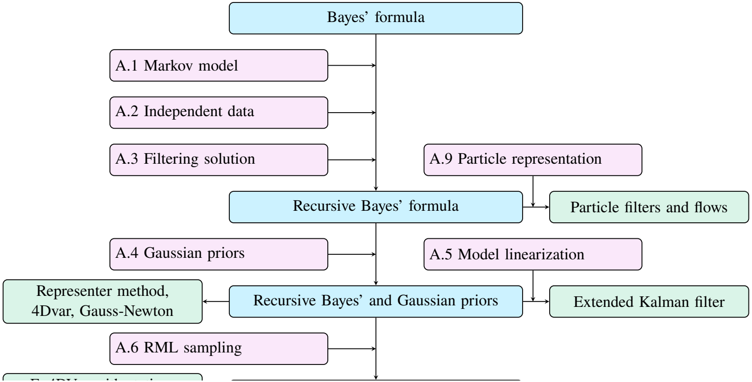
    $\mathbf{Z} = \mathbf{X}_K$ ▷ Define state vector for next assimilation window

end for

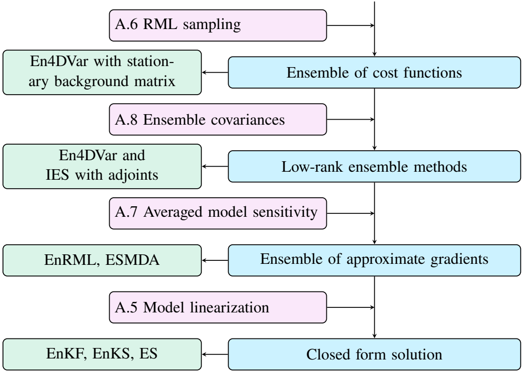
EnKS updating $[\mathbf{X}_l, \ldots, \mathbf{X}_0]$

Input: $\mathbf{Z} \in \Re^{n\times N}$ ▷ Initial model state-vector ensemble
Input: $\mathbf{D}_l \in \Re^{m\times N}$ ▷ Perturbed measurements for each assimilation window
for $l = 1, \ldots$ **do** ▷ Loop over assimilation windows
    $\mathbf{X}_0 = \mathbf{Z}$
    for $k = 1, K$ **do** ▷ Ensemble integration
        $\mathbf{X}_k = \mathbf{m}\big(\mathbf{X}_{k-1}\big)$
    end for
    $\mathbf{X} = \big[\mathbf{X}_0, \ldots, \mathbf{X}_K\big]$
    $\boldsymbol{\Upsilon} = \mathbf{h}\big(\mathbf{X}\big)$ ▷ Predicted measurments
    $\boldsymbol{\mathcal{X}}_l = \big[\boldsymbol{\mathcal{X}}_{l-1}, \mathbf{X}\big]$
    call EnKF_update$\big(\boldsymbol{\mathcal{X}}_l, \mathbf{D}_l, \boldsymbol{\Upsilon}\big)$
    $\mathbf{X} = \boldsymbol{\mathcal{X}}_l(l)$
    $\mathbf{Z} = \mathbf{X}_K$ ▷ Define state vector for next assimilation window
end for

Graphic overview



Graphic overview



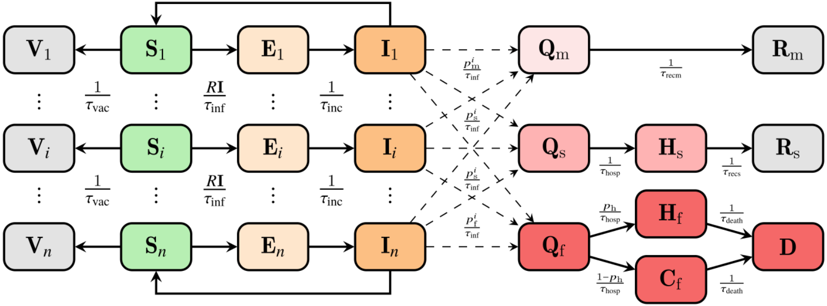
ESMDA with a SARS-COV-2 pandemic nmodel

Available observations

Hospitalized:	Within different agegroups and gender.
Deaths:	At hospitals or care homes.
Cases:	Positive tests (highly underreported).

1. Data uncertainty and availability varies in different countries.
2. The SEIR model doesn't predict deaths and hospitalizations.

Extended SEIR model



- We add age classes to model age-specific infection and death rates.
- We differentiate between mild, severe, and fatal symptoms.
- We model those with fatal symptoms who die in care homes.

Extended SEIR model

$$\frac{\partial \mathbf{S}_i}{\partial t} = -\left(\sum_{j=1}^{n} \frac{R_{ij}(t)\mathbf{I}_j}{\tau_{\mathrm{inf}}}\right)\mathbf{S}_i \tag{153}$$

$$\frac{\partial \mathbf{E}_i}{\partial t} = \left(\sum_{j=1}^{n} \frac{R_{ij}(t)\mathbf{I}_j}{\tau_{\mathrm{inf}}}\right)\mathbf{S}_i - \frac{1}{\tau_{\mathrm{inc}}}\mathbf{E}_i \tag{154}$$

$$\frac{\partial \mathbf{I}_i}{\partial t} = \frac{1}{\tau_{\mathrm{inc}}}\mathbf{E}_i - \frac{1}{\tau_{\mathrm{inf}}}\mathbf{I}_i \tag{155}$$

$$\frac{\partial \mathbf{Q}_{\mathrm{m}}}{\partial t} = \sum_{i=1}^{n} \frac{p^i_{\mathrm{m}}}{\tau_{\mathrm{inf}}}\mathbf{I}_i - (1/\tau_{\mathrm{recm}})\,\mathbf{Q}_{\mathrm{m}} \tag{156}$$

$$\frac{\partial \mathbf{Q}_{\mathrm{s}}}{\partial t} = \sum_{i=1}^{n} \frac{p^i_{\mathrm{s}}}{\tau_{\mathrm{inf}}}\mathbf{I}_i - (1/\tau_{\mathrm{hosp}})\,\mathbf{Q}_{\mathrm{s}} \tag{157}$$

$$\frac{\partial \mathbf{Q}_{\mathrm{f}}}{\partial t} = \sum_{i=1}^{n} \frac{p^i_{\mathrm{f}}}{\tau_{\mathrm{inf}}}\mathbf{I}_i - (1/\tau_{\mathrm{hosp}})\,\mathbf{Q}_{\mathrm{f}} \tag{158}$$

$$\frac{\partial \mathbf{H}_{\mathrm{s}}}{\partial t} = \frac{1}{\tau_{\mathrm{hosp}}}\mathbf{Q}_{\mathrm{s}} - \frac{1}{\tau_{\mathrm{recs}}}\mathbf{H}_{\mathrm{s}} \tag{159}$$

$$\frac{\partial \mathbf{H}_{\mathrm{f}}}{\partial t} = \frac{p_{\mathrm{h}}}{\tau_{\mathrm{hosp}}}\mathbf{Q}_{\mathrm{f}} - \frac{1}{\tau_{\mathrm{death}}}\mathbf{H}_{\mathrm{f}} \tag{160}$$

$$\frac{\partial \mathbf{C}_{\mathrm{f}}}{\partial t} = \frac{(1-p_{\mathrm{h}})}{\tau_{\mathrm{hosp}}}\mathbf{Q}_{\mathrm{f}} - \frac{1}{\tau_{\mathrm{death}}}\mathbf{C}_{\mathrm{f}} \tag{161}$$

$$\frac{\partial \mathbf{R}_{\mathrm{m}}}{\partial t} = \frac{1}{\tau_{\mathrm{recm}}}\mathbf{Q}_{\mathrm{m}} \tag{162}$$

$$\frac{\partial \mathbf{R}_{\mathrm{s}}}{\partial t} = \frac{1}{\tau_{\mathrm{recs}}}\mathbf{H}_{\mathrm{s}} \tag{163}$$

$$\frac{\partial \mathbf{D}}{\partial t} = \frac{1}{\tau_{\mathrm{death}}}\mathbf{H}_{\mathrm{f}} + \frac{1}{\tau_{\mathrm{death}}}\mathbf{C}_{\mathrm{f}} \tag{164}$$

Validity of the SEIR model

- Aggregated variables (statistical significance).
- Neglects import of cases (ok during lockdown).
- SEIR type models tend to successfully model epidemics.
- The simplicity is a huge advantage.

More complex models involve additional parameters.

Constant model parameters

1. Relative fractions p_{m}^{i}, p_{s}^{i}, p_{f}^{i} per age group.
2. Fractions dying in a Hospital p_{h} versus in a Care home $1 - p_{\mathrm{h}}$.

Age group	1	2	3	4	5	6	7	8	9	10	11
Age range	0–5	6–12	13–19	20–29	30–39	40–49	50–59	60–69	70–79	80–89	90–105
p–mild	1.00	1.00	0.99	0.99	0.97	0.96	0.93	0.90	0.84	0.81	0.81
p–severe	0.00	0.00	0.00	0.00	0.02	0.02	0.05	0.08	0.11	0.11	0.11
p–fatal	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.03	0.06	0.06

Model parameters estimated by DA

Parameter	First guess	Description
τ_{inc}	5.5	Incubation period
τ_{inf}	3.8	Infection time
τ_{recm}	14.0	Recovery time mild cases
τ_{recs}	5.0	Recovery time severe cases
τ_{hosp}	6.0	Time until hospitalization
τ_{death}	16.0	Time until death
p_{f}	0.009	Case fatality rate
p_{s}	0.039	Hospitalization rate (severe cases)
I_0		Initial number of infectious
E_0		Initial number of exposed
$R(t)$		Effective reproductive number

Effective reproductive number

$$\mathbf{R}(t) = R(t)\hat{\mathbf{R}}$$

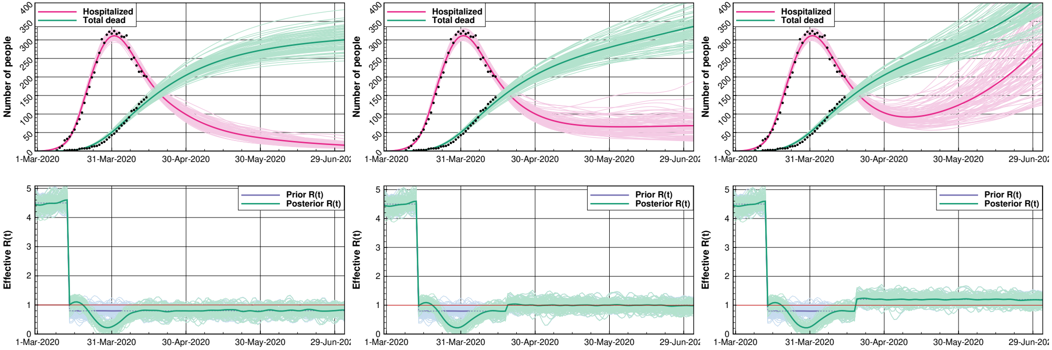
$\mathbf{R}(t)$ is a function of time (steered by how people isolate or interact).

- $R(t)$ is a scalar function of time.
- $\hat{\mathbf{R}}$ a constant matrix of transmissions between age classes..
- Behavior two weeks ago determines today's deaths and hospitalizations.
- We can estimate $R(t)$ for the past.
- We assume the value $R(t)$ for the future.

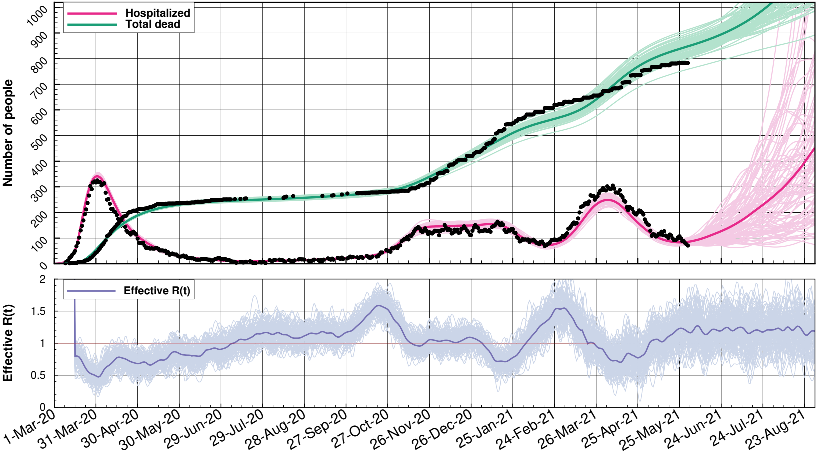
We used ESMDA

- Simple implementation and use.
- Efficient for large ensemble sizes.
- 5000 realizations and 32 ESMDA steps.

Back-to-school scenarios for Norway

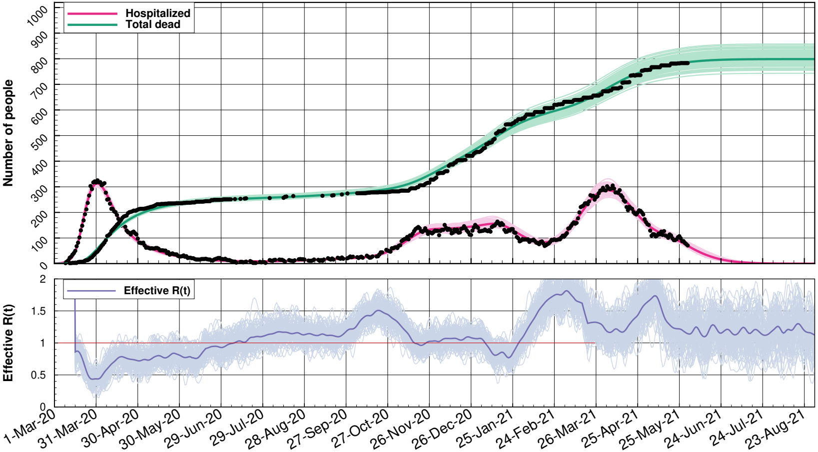


Norway: prediction without vaccinations



Available from: `https://github.com/geirev/Data-Assimilation-Fundamentals/blob/main/presentations/uibcourse.pdf`

Norway: prediction with vaccinations



Available from: `https://github.com/geirev/Data-Assimilation-Fundamentals/blob/main/presentations/uibcourse.pdf`

Summary EnKF_seir

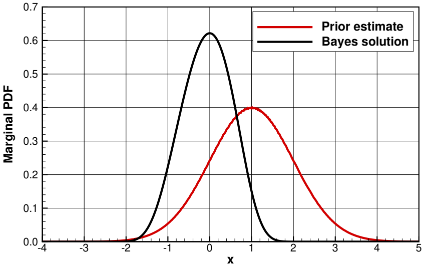
- The DA system tracks the epidemic accurately.
- We can estimate past $R(t)$.
- It is possible to quantify the impact of interventions.
- Short-term forecasting using R-persistence works well.
- Long-term scenario forecasting with specified future R.
- Code: https://github.com/geirev/EnKF_seir
- Paper: (Evensen et al., 2020) http://www.aimsciences.org/article/doi/10.3934/fods.2021001

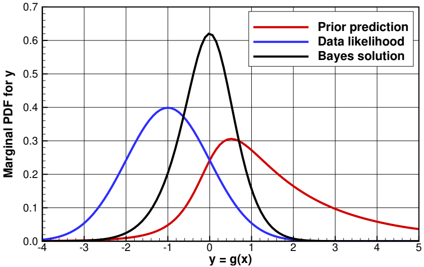
EnRML for History Matching Petroleum Models

Illustration of the parameter estimation problem

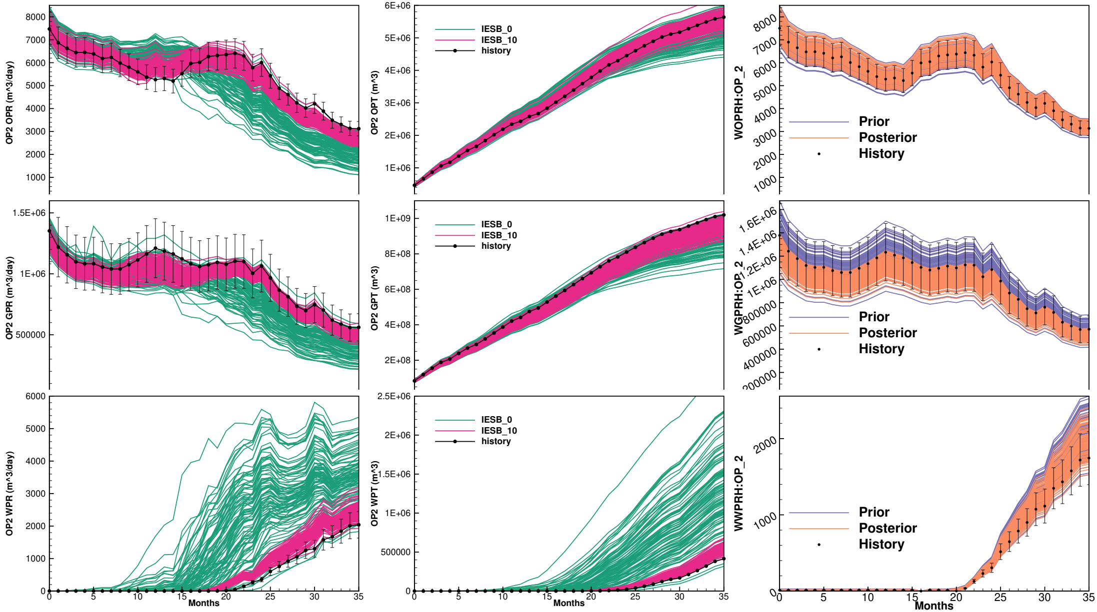
Bayes theorem gives posterior probability function for parameters $\mathbf{x}$

$$f(\mathbf{x}|\mathbf{d}) \propto f(\mathbf{d} \mid \mathbf{g}(\mathbf{x}))\, f(\mathbf{x})$$

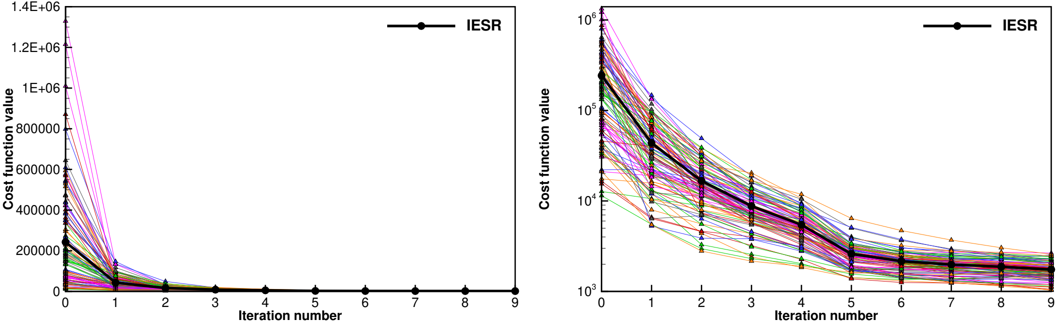




- $\mathbf{x}$ represents model input parameters like, porosity, permeability, fault multipliers
- $\mathbf{y}$ could represent predicted production of oil, gas, and water
- Prior pdf represents uncertainty of $\mathbf{x}$.
- Prior prediction pdf represents uncertainty of $\mathbf{y} = g(\mathbf{x})$.
- Data likelihood represents uncertainty of measurement $\mathbf{d}$.


OP2 OPR (m^3/day)
OP2 OPT (m^3)
WOPRH:OP_2
OP2 GPR (m^3/day)
OP2 GPT (m^3)
WGPRH:OP_2
OP2 WPR (m^3/day)
OP2 WPT (m^3)
WWPRH:OP_2
IESB_0
IESB_10
history
Prior
Posterior
History
Months

Ensemble of cost functions



Summary

Presentation follows new text book on data assimilation:

- Top-down approach for deriving the most popular methods from Bayes'.
- We introduced the important assumptions and applied approximations.
- We illustrated ensemble methods on simple problems.
- Codes for examples are available from https://github.com/geirev/

Chen, Y. and D. S. Oliver. Levenberg-Marquardt forms of the iterative ensemble smoother for efficient history matching and uncertainty quantification. *Computat Geosci*, 17:689–703, 2013. doi:10.1007/s10596-013-9351-5.

Emerick, A. A. and A. C. Reynolds. Ensemble smoother with multiple data assimilation. *Computers and Geosciences*, 55:3–15, 2013. doi:10.1016/j.cageo.2012.03.011.

Evensen, G. *Data Assimilation: The Ensemble Kalman Filter*. Springer, 2nd edition, 2009. doi:10.1007/978-3-642-03711-5.

Evensen, G. Analysis of iterative ensemble smoothers for solving inverse problems. *Computat Geosci*, 22(3):885–908, 2018. doi:10.1007/s10596-018-9731-y.

Evensen, G. Accounting for model errors in iterative ensemble smoothers. *Computat Geosci*, 23(4):761–775, 2019. doi:10.1007/s10596-019-9819-z.

Evensen, G., P. N. Raanes, A. S. Stordal, and J. Hove. Efficient implementation of an iterative ensemble smoother for data assimilation and reservoir history matching. *Frontiers in Applied Mathematics and Statistics*, 5:47, 2019. doi:10.3389/fams.2019.00047.

Evensen, G., J. Amezcua, M. Bocquet, A. Carrassi, A. Farchi, A. Fowler, P. L. Houtekamer, C. K. Jones, R. J. de Moraes, M. Pulido, C. Sampson, and F. C. Vossepoel. An international initiative of predicting the sars-cov-2 pandemic using ensemble data assimilation. *Foundations of Data Science*, page 65, 2020. doi:10.3934/fods.2021001.

Neal, R. M. Sampling from multimodal distributions using tempered transitions. *Statistics and Computing*, 6(4):353–366, 1996. doi:10.1007/BF00143556.

Raanes, P. N., A. S. Stordal, and G. Evensen. Revising the stochastic iterative ensemble smoother. *Nonlin. Processes Geophys*, 26:325–338, 2019. doi:10.5194/npg-2019-10.