Data-Assimilation Fundamentals:

A Unified Formulation of the State and Parameter Estimation Problem

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

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

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

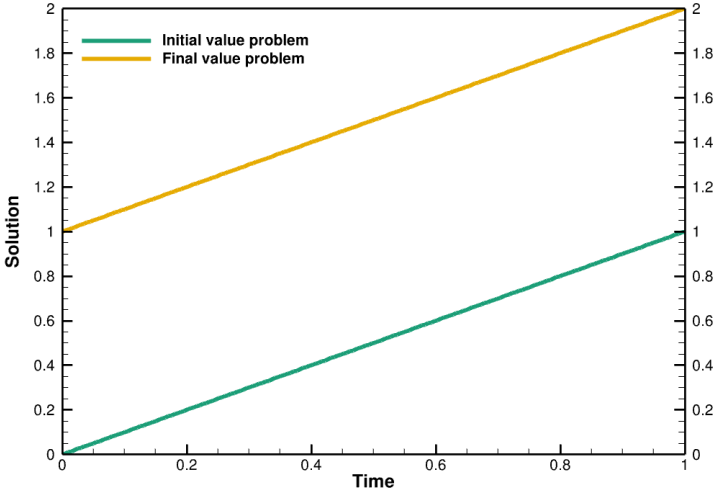
$$\frac{dx}{dt} = 1 \qquad \text{Linear model} \qquad (1)$$

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

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

- Overdetermined problem.
- No solution.

Initial- and final-value solutions



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:

$$\overline{q(t)} = 0,$$
$$\overline{a} = 0,$$
$$\overline{b} = 0,$$

The variance is known:

$$\overline{q(t_1)q(t_2)} = C\delta(t_1 - t_2),$$
$$\overline{a^2} = C,$$
$$\overline{b^2} = C,$$

No cross correlations:

$$\overline{q(t)a} = 0, \quad (7)$$
$$\overline{ab} = 0, \quad (8)$$
$$\overline{q(t)b} = 0. \quad (9)$$

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

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

Then x is an extremum if

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

when $\delta x \to 0$.

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

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

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

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

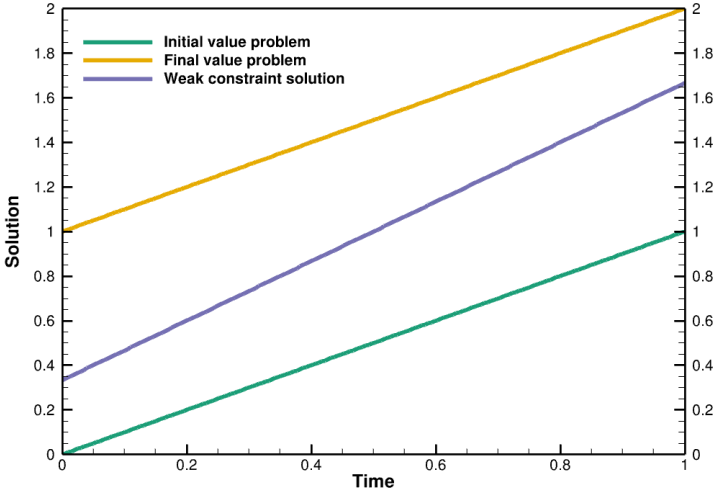
$$\frac{dx}{dt} + x = 3 \quad \text{for } t = 1, \tag{20}$$

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

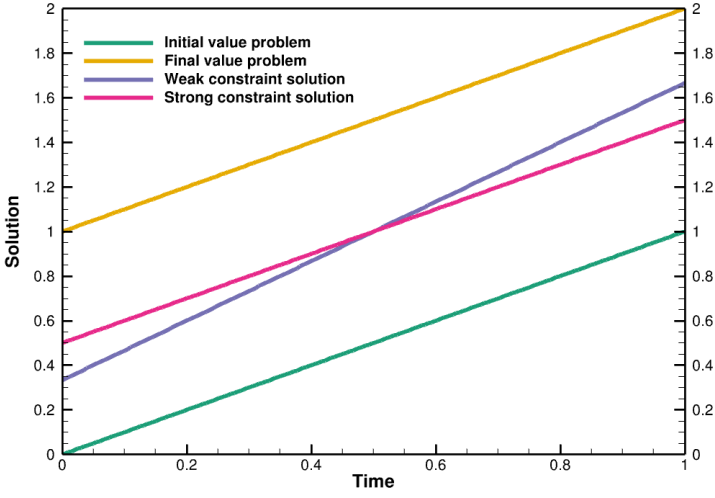
$$x = c_1 t + c_2, \tag{21}$$

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

Weak-constraint solution



Strong-constraint solution



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

$$f(x|d) = \frac{f(d|x)f(x)}{f(d)} \tag{25}$$

$$\propto f\big(d|x(1)\big)\left(f\big(x|x(0)\big)f\big(x(0)\big)\right) \tag{26}$$

$$= \exp\left(-\frac{\big(x(1)-2\big)^2}{C}\right)\exp\left(-\int_0^1 \frac{1}{C}\left(\frac{dx}{dt}-1\right)^2 dt\right)\exp\left(-\frac{\big(x(0)-0\big)^2}{C}\right) \tag{27}$$

$$= \exp\Big(-\mathcal{J}[x]\Big) \tag{28}$$

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 denisty 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:

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

Bayes' theorem

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

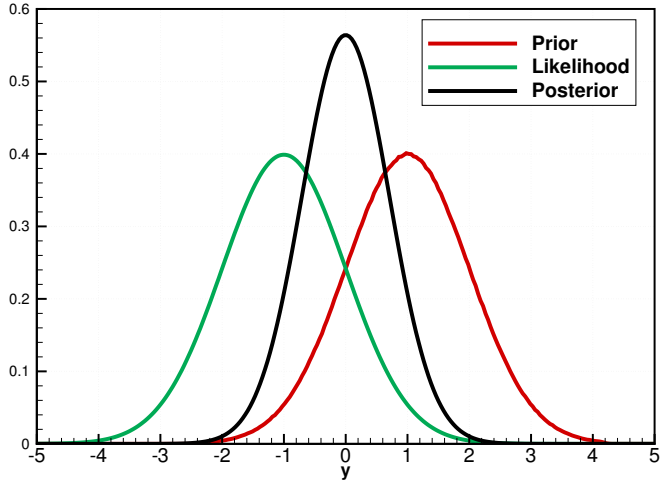
What is the likelihood function: $f(d|x)$

- 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.

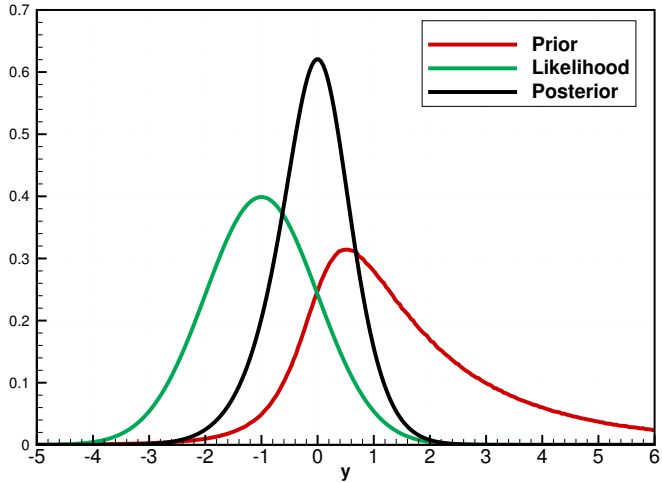
Likelihood is the plausability of a particular distribution explaining the given data. The higher the likelihood of a distribution, the more likely it is to explain the observed data.

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.

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

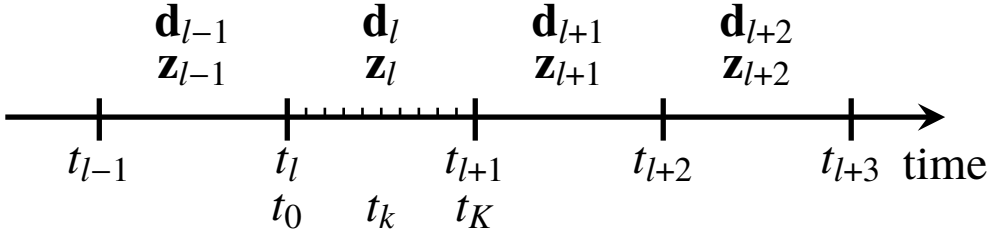
One of the strengths of Bayes' formula is that it does not try to solve the ill-defined problem of "inverting observations" but instead updates prior knowledge.

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 by ensemble integrations.

Model is Markov process

Approximation 1 (Model is 1st-order Markov process)
We assume the dynamical model is a 1st-order Markov process.

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

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

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

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

$$\vdots \tag{41}$$

$$f(\mathcal{Z}|\mathcal{D}) = \frac{f(\mathbf{d}_L|\mathbf{z}_L) f(\mathbf{z}_L|\mathbf{z}_{L-1}) f(\mathbf{z}_{L-1}, \ldots, \mathbf{z}_0|\mathbf{d}_{L-1}, \ldots, \mathbf{d}_1)}{f(\mathbf{d}_L)}. \tag{42}$$

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

$$f(\mathbf{z}_1|\mathbf{d}_1) = \frac{f(\mathbf{d}_1|\mathbf{z}_1)\ \int f(\mathbf{z}_1|\mathbf{z}_0) f(\mathbf{z}_0)\, d\mathbf{z}_0}{f(\mathbf{d}_1)} = \frac{f(\mathbf{d}_1|\mathbf{z}_1) f(\mathbf{z}_1)}{f(\mathbf{d}_1)}, \tag{43}$$

$$f(\mathbf{z}_2|\mathbf{d}_1, \mathbf{d}_2) = \frac{f(\mathbf{d}_2|\mathbf{z}_2)\ \int f(\mathbf{z}_2|\mathbf{z}_1) f(\mathbf{z}_1|\mathbf{d}_1)\, d\mathbf{z}_1}{f(\mathbf{d}_2)} = \frac{f(\mathbf{d}_2|\mathbf{z}_2) f(\mathbf{z}_2|\mathbf{d}_1)}{f(\mathbf{d}_2)}, \tag{44}$$

$$\vdots$$

$$f(\mathbf{z}_L|\mathcal{D}) = \frac{f(\mathbf{d}_L|\mathbf{z}_L)\ \int f(\mathbf{z}_L|\mathbf{z}_{L-1}) f(\mathbf{z}_{L-1}|\mathbf{d}_{L-1}, \ldots, \mathbf{d}_1)\, d\mathbf{z}_{L-1}}{f(\mathbf{d}_L)} \tag{45}$$

$$= \frac{f(\mathbf{d}_L|\mathbf{z}_L) f(\mathbf{z}_L|\mathbf{d}_{L-1})}{f(\mathbf{d}_L)}. \tag{46}$$

Or just Bayes' for the assimilation window

$$f(\mathbf{z}|\mathbf{d}) = \frac{f(\mathbf{d}|\mathbf{z})f(\mathbf{z})}{f(\mathbf{d})}, \tag{47}$$

Discrete model with uncertain inputs

$$\mathbf{x}_k = \mathbf{m}(\mathbf{x}_{k-1}, \boldsymbol{\theta}, \mathbf{u}_k, \mathbf{q}_k). \tag{48}$$

- $\mathbf{x}_k$ 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.

Error propagation by Fokker-Planck equation

Stochastic model

$$d\mathbf{x} = \mathbf{m}(\mathbf{x})\, dt + d\mathbf{q}. \tag{49}$$

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

$$\frac{\partial f(\mathbf{x})}{\partial t} + \sum_i \frac{\partial \big(m_i(\mathbf{x}) f(\mathbf{x})\big)}{\partial x_i} = \frac{1}{2}\mathbf{C}_{qq} \sum_{i,j} \frac{\partial^2 f(\mathbf{x})}{\partial x_i\, \partial x_j}. \tag{50}$$

Error propagation by covariance evolution

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

$$\mathbf{x}^{\mathrm{t}}_{k+1} = \mathbf{m}(\mathbf{x}^{\mathrm{t}}_k) + \mathbf{q}_k \approx \mathbf{m}(\mathbf{x}_k) + \mathbf{M}_k(\mathbf{x}^{\mathrm{t}}_k - \mathbf{x}_k) + \mathbf{q}_k, \tag{51}$$

$$\mathbf{x}_{k+1} = \mathbf{m}(\mathbf{x}_k), \tag{52}$$

Subtract Eq. (52) from Eq. (51), square the result, and take the expectation,

$$\mathbf{C}_{xx,k+1} \approx \mathbf{M}_k \mathbf{C}_{xx,k} \mathbf{M}_k^{\mathrm{T}} + \mathbf{C}_{qq,k}. \tag{53}$$

- $\mathbf{M}_k$ 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

$$\mathbf{x}_{j,0} \sim f(\mathbf{x}) \quad \text{and} \quad \mathbf{q}_{j,k} \sim f(\mathbf{x}_{k+1}|\mathbf{x}_k) \tag{54}$$

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

$$\mathbf{x}_{j,k+1} = \mathbf{m}(\mathbf{x}_{j,k}, \mathbf{q}_{j,k}). \tag{55}$$

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

$$\mathbb{E}[\mathbf{x}] \approx \overline{\mathbf{x}} = \frac{1}{N}\sum_{j=1}^{N} \mathbf{x}_j \tag{56}$$

$$\mathbf{C}_{xx} \approx \overline{(\mathbf{x}-\overline{\mathbf{x}})(\mathbf{x}-\overline{\mathbf{x}})^{\mathrm{T}}} = \frac{1}{N-1}\sum_{j=1}^{N} (\mathbf{x}_j-\overline{\mathbf{x}})(\mathbf{x}_j-\overline{\mathbf{x}})^{\mathrm{T}} \tag{57}$$

General smoother formulation

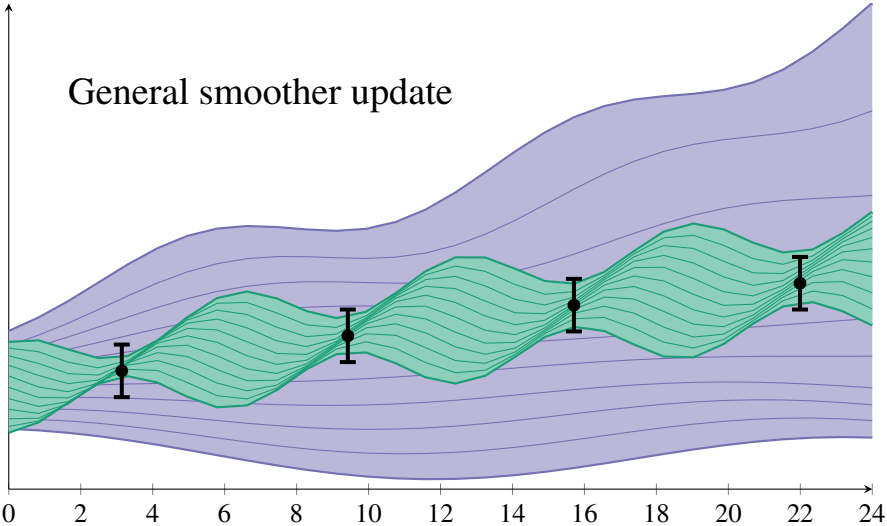
- Solve for model solution over an assimilation window $\mathbf{x}$.
- Condition on measurements distributed over the assimilation window.

Predicted measurements $\mathbf{y}$

$$\mathbf{y} = \mathbf{g}(\mathbf{z}) = \mathbf{h}(\mathbf{x}) = \mathbf{h}\big(\mathbf{m}(\mathbf{x}_0, \mathbf{q})\big) \tag{58}$$

- Measurement operator $\mathbf{h}$.
- Ensemble smoother (ES) solution, weak constraint 4DVar, Representer method.

General smoother formulation



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

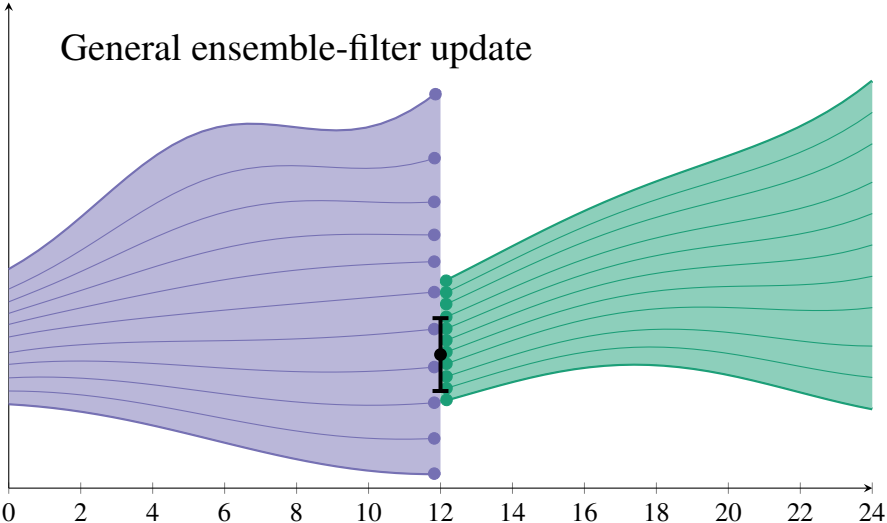
General filter formulation

- Solve for model solution at the end of an assimilation window $\mathbf{x}_K$.
- Condition on measurements at the end of the assimilation window.

$$\mathbf{y} = \mathbf{g}(\mathbf{z}) = \mathbf{h}(\mathbf{x}_K). \tag{59}$$

- Kalman filters
- EnKF (also allows for measurments distributed over the assimilation window)
- Particle filter

General filter formulation



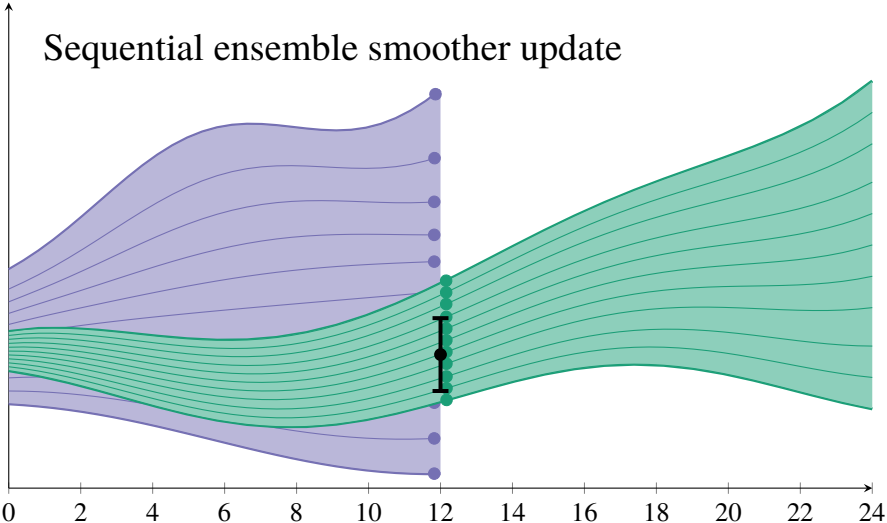
Recursive smoother formulation

- Solve for model solution in the whole (and previous) assimilation window(s) $\mathbf{x}$.
- Condition on measurements at the end of the assimilation window.

$$\mathbf{y} = \mathbf{g}(\mathbf{z}) = \mathbf{h}(\mathbf{x}_K), \tag{60}$$

- Ensemble Kalman Smoother (EnKS)

Recursive smoother formulation



Smoother for perfect models

- Solve for model solution at the beginning of an assimilation window $\mathbf{x}_0$.
- Condition on measurements distributed over the assimilation window.

$$\mathbf{y} = \mathbf{g}(\mathbf{z}) = \mathbf{h}\big(\mathbf{m}(\mathbf{x}_0)\big). \tag{61}$$

- Strong constraint 4DVar.
- Iterative ensemble smoothers (EnRML, ESMDA).

Nonlinear measurement functional

- Sequential DA with nonlinear data at the end of the assimilation window.
- Solve for model solution at the end of an assimilation window $\mathbf{x}_K$.

$$\mathbf{y} = \mathbf{g}(\mathbf{z}) = \mathbf{h}\big(\mathbf{x}_K\big). \tag{62}$$

- EnKF.

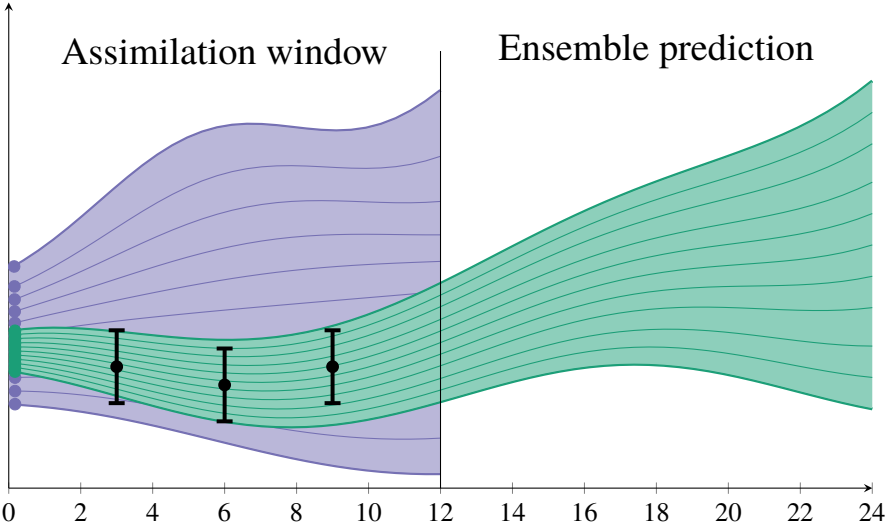
Parameter estimation

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

$$\mathbf{y} = \mathbf{g}(\mathbf{z}) = \mathbf{h}\big(\mathbf{m}(\boldsymbol{\theta})\big). \tag{63}$$

- Strong constraint 4DVar.
- Iterative ensemble smoothers (EnRML, ESMDA).
- Parameter estimation just replaces $\mathbf{x}_0$ with $\boldsymbol{\theta}$.

Smoother for perfect models



Deriving the marginal posterior pdf

Nonlinear "perfect" model and measurements

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

Bayesian formulation

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

Model pdf

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

Marginal pdf

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

Bayes' theorem related to the predicted measurements

We introduce nonlinearity through the likelihood

$$f(\mathbf{z}|\mathbf{d}) = \frac{f\big(\mathbf{d}|\mathbf{g}(\mathbf{z})\big)f(\mathbf{z})}{f(\mathbf{d})}. \tag{64}$$

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.*

$$f(\mathbf{z}|\mathbf{d}) \propto \exp\{-\mathcal{J}(\mathbf{z})\}, \tag{65}$$

Leads to a cost-function formulation for the MAP solution

Cost function

$$\mathcal{J}(\mathbf{z}) = \frac{1}{2}\left(\mathbf{z}-\mathbf{z}^{\mathrm{f}}\right)^{\mathrm{T}}\mathbf{C}_{zz}^{-1}\left(\mathbf{z}-\mathbf{z}^{\mathrm{f}}\right) + \frac{1}{2}\left(\mathbf{g}(\mathbf{z})-\mathbf{d}\right)^{\mathrm{T}}\mathbf{C}_{dd}^{-1}\left(\mathbf{g}(\mathbf{z})-\mathbf{d}\right). \tag{66}$$

The gradient set to zero

$$\mathbf{C}_{zz}^{-1}\left(\mathbf{z}^{\mathrm{a}}-\mathbf{z}^{\mathrm{f}}\right) + \nabla_{\mathbf{z}}\mathbf{g}\left(\mathbf{z}^{\mathrm{a}}\right)\mathbf{C}_{dd}^{-1}\left(\mathbf{g}\left(\mathbf{z}^{\mathrm{a}}\right)-\mathbf{d}\right) = 0. \tag{67}$$

- There is no explicit solution of the gradient equation.

Gauss-Newton methods solves for the MAP estimate

Gauss-Newton iteration

$$\mathbf{z}^{i+1} = \mathbf{z}^i - \gamma^i \left(\mathbf{C}_{zz}^{-1} + \mathbf{G}^{i\mathrm{T}} \mathbf{C}_{dd}^{-1} \mathbf{G}^i \right)^{-1} \left(\mathbf{C}_{zz}^{-1} (\mathbf{z}^i - \mathbf{z}^{\mathrm{f}}) + \mathbf{G}^{i\mathrm{T}} \mathbf{C}_{dd}^{-1} \left(\mathbf{g}(\mathbf{z}^i) - \mathbf{d} \right) \right). \tag{68}$$

- The incremental formulation is sometimes more convenient.

Incremental Gauss-Newton methods

Quadratic cost function for the increments

$$\mathcal{J}(\delta\mathbf{z}) = \frac{1}{2}\left(\delta\mathbf{z} - \boldsymbol{\xi}^i\right)^{\mathrm{T}} \mathbf{C}_{zz}^{-1} \left(\delta\mathbf{z} - \boldsymbol{\xi}^i\right) + \frac{1}{2}\left(\mathbf{G}^i\delta\mathbf{z} - \boldsymbol{\eta}^i\right)^{\mathrm{T}} \mathbf{C}_{dd}^{-1} \left(\mathbf{G}^i\delta\mathbf{z} - \boldsymbol{\eta}^i\right). \tag{69}$$

with

$$\mathbf{z}^{i+1} = \mathbf{z}^i + \delta\mathbf{z}, \tag{70}$$

$$\boldsymbol{\eta}^i = \mathbf{d} - \mathbf{g}\left(\mathbf{z}^i\right), \tag{71}$$

$$\boldsymbol{\xi}^i = \mathbf{z}^{\mathrm{f}} - \mathbf{z}^i. \tag{72}$$

- 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

$$\mathbf{x}_0 = \mathbf{x}_0^{\mathrm{f}} + \mathbf{x}_0', \tag{73}$$

$$\boldsymbol{\theta} = \boldsymbol{\theta}^{\mathrm{f}} + \boldsymbol{\theta}', \tag{74}$$

$$\mathbf{x}_{k+1} = \mathbf{m}(\mathbf{x}_k, \boldsymbol{\theta}), \tag{75}$$

Measurements

$$\mathbf{d} = \mathbf{h}(\mathbf{x}) + \mathbf{e}. \tag{76}$$

Problem formulation

State vector and covariance matrix

$$\mathbf{z} = \begin{pmatrix} \mathbf{x}_0 \\ \boldsymbol{\theta} \end{pmatrix} \quad \text{and} \quad \mathbf{C}_{zz} = \begin{pmatrix} \mathbf{C}_{x_0 x_0} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_{\theta\theta} \end{pmatrix}, \tag{77}$$

SC-4DVar costfunction

$$\mathcal{J}(\mathbf{z}) = \frac{1}{2}\left(\mathbf{z} - \mathbf{z}^{\mathrm{f}}\right)^{\mathrm{T}} \mathbf{C}_{zz}^{-1} \left(\mathbf{z} - \mathbf{z}^{\mathrm{f}}\right) + \frac{1}{2}\left(\mathbf{h}(\mathbf{x}) - \mathbf{d}\right)^{\mathrm{T}} \mathbf{C}_{dd}^{-1} \left(\mathbf{h}(\mathbf{x}) - \mathbf{d}\right), \tag{78}$$

Solve for initial condition and parameter that minimize the cost function

Lagrangian formulation

$$\begin{aligned}\mathcal{L}(\mathbf{x}_0,\ldots,\mathbf{x}_{K+1},\boldsymbol{\theta},\boldsymbol{\lambda}_1,\ldots,\boldsymbol{\lambda}_{K+1}) &= \frac{1}{2}\left(\mathbf{x}_0-\mathbf{x}_0^{\mathrm{f}}\right)^{\mathrm{T}}\mathbf{C}_{x_0x_0}^{-1}\left(\mathbf{x}_0-\mathbf{x}_0^{\mathrm{f}}\right)\\ &+\frac{1}{2}\left(\boldsymbol{\theta}-\boldsymbol{\theta}^{\mathrm{f}}\right)^{\mathrm{T}}\mathbf{C}_{\theta\theta}^{-1}\left(\boldsymbol{\theta}-\boldsymbol{\theta}^{\mathrm{f}}\right)\\ &+\frac{1}{2}\left(\mathbf{h}(\mathbf{x})-\mathbf{d}\right)^{\mathrm{T}}\mathbf{C}_{dd}^{-1}\left(\mathbf{h}(\mathbf{x})-\mathbf{d}\right)\\ &+\sum_{k=0}^{K}\boldsymbol{\lambda}_{k+1}^{\mathrm{T}}\left(\mathbf{x}_{k+1}-\mathbf{m}(\mathbf{x}_k,\boldsymbol{\theta})\right).\end{aligned} \tag{79}$$

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

Gradient of Lagrangian

$$\nabla_{\mathbf{x}_k} \mathcal{L}(\mathbf{x}, \boldsymbol{\theta}, \boldsymbol{\lambda}) = \mathbf{H}_k^{\mathrm{T}} \, \mathbf{C}_{dd}^{-1} \left(\mathbf{h}(\mathbf{x}) - \mathbf{d} \right) + \boldsymbol{\lambda}_k - \mathbf{M}_{x,k}^{\mathrm{T}} \boldsymbol{\lambda}_{k+1}, \tag{80}$$

$$\nabla_{\mathbf{x}_{K+1}} \mathcal{L}(\mathbf{z}, \mathbf{x}, \boldsymbol{\lambda}) = \boldsymbol{\lambda}_{K+1}, \tag{81}$$

$$\begin{aligned} \nabla_{\mathbf{x}_0} \mathcal{L}(\mathbf{x}, \boldsymbol{\theta}, \boldsymbol{\lambda}) &= \mathbf{C}_{zz}^{-1} \left(\mathbf{x}_0 - \mathbf{x}_0^{\mathrm{f}} \right) - \mathbf{M}_{x,0}^{\mathrm{T}} \boldsymbol{\lambda}_1 \\ &= \mathbf{C}_{zz}^{-1} \left(\mathbf{x}_0 - \mathbf{x}_0^{\mathrm{f}} \right) - \boldsymbol{\lambda}_0, \end{aligned} \tag{82}$$

$$\nabla_{\boldsymbol{\theta}} \mathcal{L}(\mathbf{x}, \boldsymbol{\theta}, \boldsymbol{\lambda}) = \mathbf{C}_{\theta\theta}^{-1} \left(\boldsymbol{\theta} - \boldsymbol{\theta}^{\mathrm{f}} \right) - \sum_{k=0}^{K} \mathbf{M}_{\theta,k}^{\mathrm{T}} \boldsymbol{\lambda}_{k+1}, \tag{83}$$

$$\nabla_{\boldsymbol{\lambda}_k} \mathcal{L}(\mathbf{x}, \boldsymbol{\theta}, \boldsymbol{\lambda}) = \mathbf{x}_{k+1} - \mathbf{m}(\mathbf{x}_k, \boldsymbol{\theta}). \tag{84}$$

Euler-Lagrange equation(s)

Forward model

$$\mathbf{x}_0 = \mathbf{x}_0^{\mathrm{f}} + \mathbf{C}_{x_0 x_0} \boldsymbol{\lambda}_0, \tag{85}$$

$$\boldsymbol{\theta} = \boldsymbol{\theta}^{\mathrm{f}} + \mathbf{C}_{\theta\theta} \sum_{k=0}^{K} \mathbf{M}_{\theta,k}^{\mathrm{T}} \boldsymbol{\lambda}_{k+1}, \tag{86}$$

$$\mathbf{x}_{k+1} = \mathbf{m}(\mathbf{x}_k, \boldsymbol{\theta}), \tag{87}$$

Backward model for the adjoint variable

$$\boldsymbol{\lambda}_{K+1} = 0, \tag{88}$$

$$\boldsymbol{\lambda}_k = \mathbf{M}_{x,k}^{\mathrm{T}} \, \boldsymbol{\lambda}_{k+1} - \mathbf{H}_k^{\mathrm{T}} \, \mathbf{C}_{dd}^{-1} \, \big(\mathbf{h}(\mathbf{x}) - \mathbf{d}\big). \tag{89}$$

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. (82)
  $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \gamma \mathbf{B} \nabla_{\theta} \mathcal{L}(\mathbf{x}, \boldsymbol{\theta}, \boldsymbol{\lambda})$ ▷ Update $\boldsymbol{\theta}$ using Eq. (83)
until convergence

Incremental strong constraint 4DVar

Incremental SC-4DVar

Nonlinear model

$$\mathbf{x}_0 = \mathbf{x}_0^{\mathrm{f}} + \mathbf{x}_0', \tag{90}$$

$$\mathbf{x}_{k+1} = \mathbf{m}(\mathbf{x}_k). \tag{91}$$

State variable

$$\mathbf{z} = \mathbf{x}_0 \tag{92}$$

We compute updates

$$\mathbf{z}^{i+1} = \mathbf{z}^i + \delta\mathbf{z}, \tag{93}$$

Solving for the increments

Inner incremental SC-4DVar costfunction

$$\mathcal{J}(\delta\mathbf{z}) = \frac{1}{2}\left(\delta\mathbf{z} - \boldsymbol{\xi}^i\right)^{\mathrm{T}} \mathbf{C}_{zz}^{-1}\left(\delta\mathbf{z} - \boldsymbol{\xi}^i\right) + \frac{1}{2}\left(\mathbf{H}^i\delta\mathbf{x} - \boldsymbol{\eta}^i\right)^{\mathrm{T}} \mathbf{C}_{dd}^{-1}\left(\mathbf{H}^i\delta\mathbf{x} - \boldsymbol{\eta}^i\right), \tag{94}$$

Linear model for the increments

$$\delta\mathbf{x}_0 = \delta\mathbf{z}^i, \tag{95}$$

$$\delta\mathbf{x}_{k+1} = \mathbf{M}_k^i\delta\mathbf{x}_k. \tag{96}$$

Prior increment

$$\boldsymbol{\xi}^i = \mathbf{x}_0^{\mathrm{f}} - \mathbf{x}_0^i, \tag{97}$$

Innovation

$$\boldsymbol{\eta}^i = \mathbf{d} - \mathbf{h}(\mathbf{x}^i). \tag{98}$$

Lagrangian formulation

$$\begin{aligned}\mathcal{L}(\delta\mathbf{z}, \delta\mathbf{x}, \delta\boldsymbol{\lambda}) = &\frac{1}{2}\left(\delta\mathbf{z} - \boldsymbol{\xi}^i\right)^{\mathrm{T}} \mathbf{C}_{zz}^{-1} \left(\delta\mathbf{z} - \boldsymbol{\xi}^i\right) \\ &+ \frac{1}{2}\left(\mathbf{H}^i\delta\mathbf{x} - \boldsymbol{\eta}^i\right)^{\mathrm{T}} \mathbf{C}_{dd}^{-1} \left(\mathbf{H}^i\delta\mathbf{x} - \boldsymbol{\eta}^i\right) \\ &+ \sum_{k=0}^{K} \delta\boldsymbol{\lambda}_{k+1}^{\mathrm{T}}\left(\delta\mathbf{x}_{k+1} - \mathbf{M}_k^i\delta\mathbf{x}_k\right).\end{aligned} \tag{99}$$

Introduces the linear model for the increments using a Lagrangian multiplier.

Gradient of Lagrangian

$$\nabla_{\delta \mathbf{x}_k} \mathcal{L}(\delta \mathbf{z}, \delta \mathbf{x}, \delta \boldsymbol{\lambda}) = \mathbf{H}_k^{i\mathrm{T}} \mathbf{C}_{dd}^{-1} \left(\mathbf{H}^i \delta \mathbf{x} - \boldsymbol{\eta}^i \right) + \delta \boldsymbol{\lambda}_k - \mathbf{M}_k^{i\mathrm{T}} \delta \boldsymbol{\lambda}_{k+1}, \tag{100}$$

$$\nabla_{\delta \mathbf{x}_K} \mathcal{L}(\delta \mathbf{z}, \delta \mathbf{x}, \delta \boldsymbol{\lambda}) = \delta \boldsymbol{\lambda}_{K+1}, \tag{101}$$

$$\begin{aligned} \nabla_{\delta \mathbf{z}} \mathcal{L}(\delta \mathbf{z}, \delta \mathbf{x}, \delta \boldsymbol{\lambda}) &= \mathbf{C}_{zz}^{-1} \left(\delta \mathbf{z} - \boldsymbol{\xi}^i \right) - \mathbf{M}_0^{i\mathrm{T}} \delta \boldsymbol{\lambda}_1 \\ &= \mathbf{C}_{zz}^{-1} \left(\delta \mathbf{z} - \boldsymbol{\xi}^i \right) - \delta \boldsymbol{\lambda}_0, \end{aligned} \tag{102}$$

$$\nabla_{\delta \boldsymbol{\lambda}_k} \mathcal{L}(\delta \mathbf{z}, \delta \mathbf{x}, \delta \boldsymbol{\lambda}) = \delta \mathbf{x}_{k+1} - \mathbf{M}_k^i \delta \mathbf{x}_k. \tag{103}$$

Euler-Lagrange equation(s)

Forward model

$$\delta \mathbf{x}_0 = \xi^i + \mathbf{C}_{zz}\, \delta \boldsymbol{\lambda}_0, \tag{104}$$

$$\delta \mathbf{x}_{k+1} - \mathbf{M}_k^i \delta \mathbf{x}_k = 0, \tag{105}$$

Adjoint model

$$\delta \boldsymbol{\lambda}_{K+1} = 0, \tag{106}$$

$$\delta \boldsymbol{\lambda}_k - \mathbf{M}_k^{i\,\mathrm{T}} \delta \boldsymbol{\lambda}_{k+1} = \mathbf{H}_k^{i\,\mathrm{T}} \mathbf{C}_{dd}^{-1} \left(\mathbf{H}^i \delta \mathbf{x} - \boldsymbol{\eta}^i \right). \tag{107}$$

Incremental 4DVar algorithm

Input: $\mathbf{x}_0^{\mathrm{f}} \in \mathfrak{R}^n$; $\mathbf{d} \in \mathfrak{R}^m$; $\mathbf{C}_{dd} \in \mathfrak{R}^{m \times m}$; $\mathbf{C}_{zz} \in \mathfrak{R}^{n \times n}$ ▷ Prior inputs

$\mathbf{x}_0^0 = \mathbf{x}_0^{\mathrm{f}}$ ▷ Initialization of $\mathbf{x}_0$

$\delta\mathbf{z} = 0$ ▷ Initialization of $\delta\mathbf{z}$

$i = 1$

repeat ▷ Iteration loop

  $\mathbf{x}_0^i = \mathbf{x}_0^{i-1} + \delta\mathbf{z}$ ▷ Update initial condition

  $\boldsymbol{\xi}^i = \mathbf{x}_0^{\mathrm{f}} - \mathbf{x}_0^i$ ▷ Current increment

  for $k = 0 : K$ **do** ▷ Integrate forward model

    $\mathbf{x}_{k+1}^i = \mathbf{m}(\mathbf{x}_k^i)$ ▷ Nonlinear model prediction

  end for

  $\boldsymbol{\eta}^i = \mathbf{d} - \mathbf{h}(\mathbf{x}^i)$ ▷ Current innovation vector

  repeat ▷ Iteration loop

    $\delta\mathbf{x}_0 = \delta\mathbf{z}$

    for $k = 0 : K$ **do** ▷ Integrate forward model

      $\delta\mathbf{x}_{k+1} = \mathbf{M}_k^i \, \delta\mathbf{x}_k$

    end for

    $\delta\boldsymbol{\lambda}_{K+1} = 0$

    for $k = K : 0$ **do** ▷ Integrate backward adjoint model

      $\delta\boldsymbol{\lambda}_k = {\mathbf{M}_k^i}^{\mathrm{T}} \delta\boldsymbol{\lambda}_{k+1} - {\mathbf{H}_k^i}^{\mathrm{T}} \mathbf{C}_{dd}^{-1} \left(\mathbf{H}^i \, \delta\mathbf{x} - \boldsymbol{\eta}^i \right)$

    end for

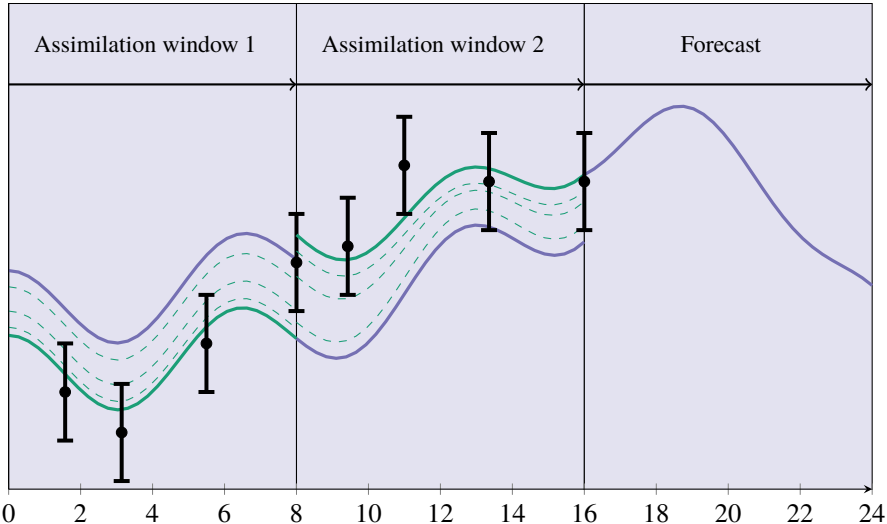
    $\delta\mathbf{z} \leftarrow \delta\mathbf{z} - \gamma \mathbf{B} \left(\delta\mathbf{z} - \boldsymbol{\xi}^i - \mathbf{C}_{zz} \, \delta\boldsymbol{\lambda}_0 \right)$ ▷ Update $\delta\mathbf{z}$ using gradient in Eq. (102)

  until convergence

  $i = i + 1$

until convergence

Incremental 4DVar algorithm



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

Comments

- Incremental SC-4DVar is now the standard approach in weather prediction.
- The method solves a data assimilation problem over an assimilation window.
- The solution is the initial condition for the window and it defines the MAP solution.
- Do we know that the incremental formulation converges in all cases?
- We do not evolve error statistics in time.
- Finally, it is a strong-constrant solution that assumes the model is perfect.

Weak constraint 4DVar

WC-4DVar: State-space formulation

Model with additive errors

$$\mathbf{x}_k = \mathbf{m}(\mathbf{x}_{k-1}) + \mathbf{q}_k. \tag{108}$$

Define state vector and model error covariance

$$\mathbf{z} = \mathbf{x} = \begin{pmatrix} \mathbf{x}_0 \\ \vdots \\ \mathbf{x}_K \end{pmatrix}, \quad \text{and} \quad \mathbf{C}_{qq} = \begin{pmatrix} \mathbf{C}_{q_1 q_1} & \cdots & \mathbf{C}_{q_1 q_K} \\ \vdots & \ddots & \vdots \\ \mathbf{C}_{q_K q_1} & \cdots & \mathbf{C}_{q_K q_K} \end{pmatrix}, \tag{109}$$

WC-4DVar cost function

The weak-constraint cost function (generalized inverse formulation)

$$\begin{aligned}\mathcal{J}(\mathbf{x}) = &\frac{1}{2}\left(\mathbf{x}_0 - \mathbf{x}_0^{\mathrm{f}}\right)^{\mathrm{T}} \mathbf{C}_{x_0 x_0}^{-1} \left(\mathbf{x}_0 - \mathbf{x}_0^{\mathrm{f}}\right) \\ &+ \frac{1}{2}\left(\mathbf{h}(\mathbf{x}) - \mathbf{d}\right)^{\mathrm{T}} \mathbf{C}_{dd}^{-1} \left(\mathbf{h}(\mathbf{x}) - \mathbf{d}\right) \\ &+ \frac{1}{2}\sum_{r=1}^{K}\sum_{s=1}^{K}\left(\mathbf{x}_r - \mathbf{m}(\mathbf{x}_{r-1})\right)^{\mathrm{T}} \mathbf{C}_{qq}^{-1}(r,s)\left(\mathbf{x}_s - \mathbf{m}(\mathbf{x}_{s-1})\right).\end{aligned} \tag{110}$$

WC-4DVar: incremental form

Define update

$$\mathbf{x}^{i+1} = \mathbf{x}^{i} + \delta\mathbf{x}. \tag{111}$$

Linearized model

$$\begin{aligned} \mathbf{m}\left(\mathbf{x}_k^{i+1}\right) &= \mathbf{m}\left(\mathbf{x}_k^{i} + \delta\mathbf{x}_k\right) \\ &\approx \mathbf{m}\left(\mathbf{x}_k^{i}\right) + \mathbf{M}_k\delta\mathbf{x}_k, \end{aligned} \tag{112}$$

Linearized measurement operator

$$\begin{aligned} \mathbf{h}\left(\mathbf{x}^{i+1}\right) &= \mathbf{h}\left(\mathbf{x}^{i} + \delta\mathbf{x}\right) \\ &\approx \mathbf{h}\left(\mathbf{x}^{i}\right) + \mathbf{H}\delta\mathbf{x}, \end{aligned} \tag{113}$$

WC-4DVar: incremental form

Write the model residual in Eq. (110) as

$$\begin{aligned}\mathbf{x}_k^{i+1} - \mathbf{m}\left(\mathbf{x}_{k-1}^{i+1}\right) &\approx \mathbf{x}_k^{i+1} - \mathbf{m}\left(\mathbf{x}_{k-1}^{i}\right) - \mathbf{M}_{k-1}\delta\mathbf{x}_{k-1} \\ &= \mathbf{x}_k^{i+1} - \mathbf{x}_k^{i} + \mathbf{x}_k^{i} - \mathbf{m}\left(\mathbf{x}_{k-1}^{i}\right) - \mathbf{M}_{k-1}\delta\mathbf{x}_{k-1} \\ &= \delta\mathbf{x}_k - \mathbf{M}_{k-1}\delta\mathbf{x}_{k-1} + \boldsymbol{\xi}_k^i,\end{aligned} \tag{114}$$

where

$$\boldsymbol{\xi}_k^i = \mathbf{x}_k^i - \mathbf{m}\left(\mathbf{x}_{k-1}^i\right). \tag{115}$$

Innovations becomes

$$\boldsymbol{\eta}^i = \mathbf{d} - \mathbf{h}\left(\mathbf{x}^i\right). \tag{116}$$

WC incremental cost function

The weak-constraint incremental cost function

$$\begin{aligned}\mathcal{J}(\delta\mathbf{x}) = &\frac{1}{2}\left(\delta\mathbf{x}_0 - \boldsymbol{\xi}_0^i\right)^{\mathrm{T}}\mathbf{C}_{x_0x_0}^{-1}\left(\delta\mathbf{x}_0 - \boldsymbol{\xi}_0^i\right)\\ &+\frac{1}{2}\left(\mathbf{H}^i\delta\mathbf{x} - \boldsymbol{\eta}^i\right)^{\mathrm{T}}\mathbf{C}_{dd}^{-1}\left(\mathbf{H}^i\delta\mathbf{x} - \boldsymbol{\eta}^i\right)\\ &+\frac{1}{2}\sum_{r=1}^{K}\sum_{s=1}^{K}\left(\delta\mathbf{x}_r - \mathbf{M}_{r-1}\delta\mathbf{x}_{r-1} + \boldsymbol{\xi}_r^i\right)^{\mathrm{T}}\mathbf{C}_{qq}^{-1}(r,s)\left(\delta\mathbf{x}_s - \mathbf{M}_{s-1}\delta\mathbf{x}_{s-1} + \boldsymbol{\xi}_s^i\right).\end{aligned} \tag{117}$$

Linear problem with Gaussian priors

Minimizing the cost function for the increment

Gradient of the cost function to the model state $\delta\mathbf{x}_k$ gives

$$\begin{aligned}\nabla_{\delta\mathbf{x}_k}\mathcal{J}(\delta\mathbf{x}) = &\mathbf{H}_k^{i\,\mathrm{T}}\mathbf{C}_{dd}^{-1}\left(\mathbf{H}^i\delta\mathbf{x}-\boldsymbol{\eta}^i\right)\\ &+\sum_{s=1}^{K}\mathbf{C}_{qq}^{-1}(k,s)\left(\delta\mathbf{x}_s-\mathbf{M}_{s-1}^i\delta\mathbf{x}_{s-1}+\boldsymbol{\xi}_s^i\right)\\ &-\mathbf{M}_k^{i\,\mathrm{T}}\sum_{s=1}^{K}\mathbf{C}_{qq}^{-1}(k+1,s)\left(\delta\mathbf{x}_s-\mathbf{M}_{s-1}^i\delta\mathbf{x}_{s-1}+\boldsymbol{\xi}_s^i\right).\end{aligned} \tag{118}$$

Define an adjoint vector for each time step as

$$\delta\boldsymbol{\lambda}_k=\sum_{s=1}^{K}\mathbf{C}_{qq}^{-1}(k,s)\left(\delta\mathbf{x}_s-\mathbf{M}_{s-1}^i\delta\mathbf{x}_{s-1}+\boldsymbol{\xi}_s^i\right), \tag{119}$$

and write Eq. (118) as

$$\nabla_{\delta\mathbf{x}_k}\mathcal{J}(\delta\mathbf{x})=\mathbf{H}_k^{i\,\mathrm{T}}\mathbf{C}_{dd}^{-1}\left(\mathbf{H}^i\delta\mathbf{x}-\boldsymbol{\eta}^i\right)+\delta\boldsymbol{\lambda}_k-\mathbf{M}_k^{i\,\mathrm{T}}\delta\boldsymbol{\lambda}_{k+1}. \tag{120}$$

Euler-Lagrange equation(s)

Forward model

$$\delta \mathbf{x}_0 = \boldsymbol{\xi}_0^i + \mathbf{C}_{x_0 x_0} \delta \boldsymbol{\lambda}_0, \tag{121}$$

$$\delta \mathbf{x}_k - \mathbf{M}_{k-1}^i \delta \mathbf{x}_{k-1} = -\boldsymbol{\xi}_k^i + \sum_{s=1}^{K} \mathbf{C}_{qq}(k,s) \delta \boldsymbol{\lambda}_s, \tag{122}$$

Backward model

$$\delta \boldsymbol{\lambda}_{K+1} = \mathbf{0}, \tag{123}$$

$$\delta \boldsymbol{\lambda}_k - \mathbf{M}_k^{i\mathrm{T}} \delta \boldsymbol{\lambda}_{k+1} = -\mathbf{H}_k^{i\mathrm{T}} \mathbf{C}_{dd}^{-1} \left(\mathbf{H}^i \delta \mathbf{x} - \boldsymbol{\eta}^i \right). \tag{124}$$

Representer method

We have a linear problem for the increments. Assume solution of the form

$$\delta\mathbf{x} = \delta\mathbf{x}^{\mathrm{f}} + \sum_{p=1}^{m} b_p \mathbf{r}_p = \delta\mathbf{x}^{\mathrm{f}} + \mathbf{R}\mathbf{b}, \tag{125}$$

$$\delta\boldsymbol{\lambda} = \delta\boldsymbol{\lambda}^{\mathrm{f}} + \sum_{p=1}^{m} b_p \mathbf{s}_p = \delta\boldsymbol{\lambda}^{\mathrm{f}} + \mathbf{S}\mathbf{b}. \tag{126}$$

Solution is a linear combination of influence functions (representers) $\mathbf{r}_p$, one for each measurement.

Euler-Lagrange equations: Prior

We get for the prior

$$\delta \mathbf{x}_0^{\mathrm{f}} = \boldsymbol{\xi}_0^i, \tag{127}$$

$$\delta \mathbf{x}_k^{\mathrm{f}} - \mathbf{M}_{k-1}^i \delta \mathbf{x}_{k-1}^{\mathrm{f}} = -\boldsymbol{\xi}_k^i, \tag{128}$$

$$\delta \boldsymbol{\lambda}_{K+1}^{\mathrm{f}} = 0, \tag{129}$$

$$\delta \boldsymbol{\lambda}_k^{\mathrm{f}} - \mathbf{M}_k^{i\,\mathrm{T}} \delta \boldsymbol{\lambda}_{k+1}^{\mathrm{f}} = 0. \tag{130}$$

which is just the prior model solution obtained from one model integration.

Derivation for the representers

$$\mathbf{R}_0\mathbf{b} = \mathbf{C}_{x_0x_0}\mathbf{M}_0^{i\,\mathrm{T}}\mathbf{S}_1\mathbf{b}, \tag{131}$$

$$\left(\mathbf{R}_k\mathbf{b} - \mathbf{M}_{k-1}^{i}\mathbf{R}_{k-1}\mathbf{b}\right) = \sum_{s=1}^{K}\mathbf{C}_{qq}(k,s)\mathbf{S}_s\mathbf{b}, \tag{132}$$

$$\mathbf{S}_{K+1}\mathbf{b} = \mathbf{0}, \tag{133}$$

$$\mathbf{S}_k\mathbf{b} - \mathbf{M}_k^{i\,\mathrm{T}}\mathbf{S}_{k+1}\mathbf{b} = \mathbf{H}_k^{i\,\mathrm{T}}\mathbf{C}_{dd}^{-1}\left(\boldsymbol{\eta}^i - \mathbf{H}^i(\delta\mathbf{x}^{\mathrm{f}} + \mathbf{R}\mathbf{b})\right). \tag{134}$$

Define $\mathbf{b}$ as

$$\mathbf{b} = \mathbf{C}_{dd}^{-1}\left(\boldsymbol{\eta}^i - \mathbf{H}^i(\delta\mathbf{x}^{\mathrm{f}} + \mathbf{R}\mathbf{b})\right), \tag{135}$$

such that Eq. (134) simplifies to

$$\mathbf{S}_k\mathbf{b} - \mathbf{M}_k^{i\,\mathrm{T}}\mathbf{S}_{k+1}\mathbf{b} = \mathbf{H}_k^{i\,\mathrm{T}}\mathbf{b}. \tag{136}$$

Euler-Lagrange equations: Representers

$$\mathbf{R}_0 = \mathbf{C}_{x_0 x_0} \mathbf{M}_0^{i\,\mathrm{T}} \mathbf{S}_1, \tag{137}$$

$$\mathbf{R}_k - \mathbf{M}_{k-1}^i \mathbf{R}_{k-1} = \sum_{s=1}^{K} \mathbf{C}_{qq}(k,s) \mathbf{S}_s, \tag{138}$$

$$\mathbf{S}_{K+1} = \mathbf{0}, \tag{139}$$

$$\mathbf{S}_k - \mathbf{M}_k^{i\,\mathrm{T}} \mathbf{S}_{k+1} = \mathbf{H}_k^{i\,\mathrm{T}}. \tag{140}$$

Requires $2m$ model integrations.

Euler-Lagrange equations: **b**

Linear system for the representer coefficients b

$$\left(\mathbf{H}^i\mathbf{R} + \mathbf{C}_{dd}\right)\mathbf{b} = \boldsymbol{\eta}^i - \mathbf{H}^i\delta\mathbf{x}^{\mathrm{f}}. \tag{141}$$

Results in minimizing solution from $2m + 1$ model integrations.

More efficient solution method

Given $\mathbf{b}$ we obtain the final solution from (124)

Adjoint equation forced by b

$$\delta\boldsymbol{\lambda}_k - \mathbf{M}_k^{i\,\mathrm{T}}\,\delta\boldsymbol{\lambda}_{k+1} = \mathbf{H}_k^{i\,\mathrm{T}}\mathbf{b}. \tag{142}$$

followed by a forward model integration.
Thus, no need to store the representers $\mathbf{R}$, (only $\mathbf{H}^i\mathbf{R}$ is needed to solve for $\mathbf{b}$).

An even more efficient solution method

Use an iterative equation solver:

$$\mathbf{C}\mathbf{b} = \boldsymbol{\nu}. \tag{143}$$

Solving Eq. (143) is equivalent to minimizing the functional

$$\phi(\mathbf{b}) = \frac{1}{2}\mathbf{b}^{\mathrm{T}}\mathbf{C}\mathbf{b} - \mathbf{b}^{\mathrm{T}}\boldsymbol{\nu}, \tag{144}$$

which has a gradient

$$\nabla_{\mathbf{b}}\phi(\mathbf{b}) = \boldsymbol{\rho} = \mathbf{C}\mathbf{b} - \boldsymbol{\nu}. \tag{145}$$

Iterative solution

$$\mathbf{b}^{i+1} = \mathbf{b}^{i} - \gamma\boldsymbol{\rho}, \tag{146}$$

We don't need to know $\mathbf{C}$ only the product $\mathbf{C}\mathbf{b}$.

Evaluate the product $\mathbf{Rv}$

Define $\mathbf{c} = \mathbf{Rv}$ and $\boldsymbol{\psi} = \mathbf{Sv}$, and write (131–134) as

$$\mathbf{c}_0 = \mathbf{C}_{x_0 x_0} \mathbf{M}_0^{i\,\mathrm{T}} \boldsymbol{\psi}_1, \tag{147}$$

$$\mathbf{c}_k = \mathbf{M}_{k-1}^{i} \mathbf{c}_{k-1} + \sum_{s=1}^{K} \mathbf{C}_{qq}(k,s) \boldsymbol{\psi}_s, \tag{148}$$

$$\boldsymbol{\psi}_{K+1} = \mathbf{0}, \tag{149}$$

$$\boldsymbol{\psi}_k = \mathbf{M}_k^{i\,\mathrm{T}} \boldsymbol{\psi}_{k+1} + \mathbf{H}_k^{i\,\mathrm{T}} \mathbf{v}. \tag{150}$$

By measuring this solution, we find for any nonzero vector $\mathbf{v}$

$$\mathbf{Hc} = \mathbf{HRv} = \mathcal{R}\mathbf{v}, \tag{151}$$

Comments

- Efficient solution for WC-4DVar problem by the Representer method.
- The method solves the data-assimilation problem over an assimilation window.
- The solution is the model state over the window and defines the MAP solution.
- Do we know that the incremental formulation converges in all cases?
- We do not evolve error statistics in time.
- A prediction should be initialized from the end of the assimilation window.
- An ensemble of 4DVars would allow for evolving error statistics in time.

Representer method with an Ekman flow model

Representer solution for an Ekman flow model

Model

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{k} \times \mathbf{u} = \frac{\partial}{\partial z}\left(A \frac{\partial \mathbf{u}}{\partial z}\right), \tag{152}$$

Initial condition

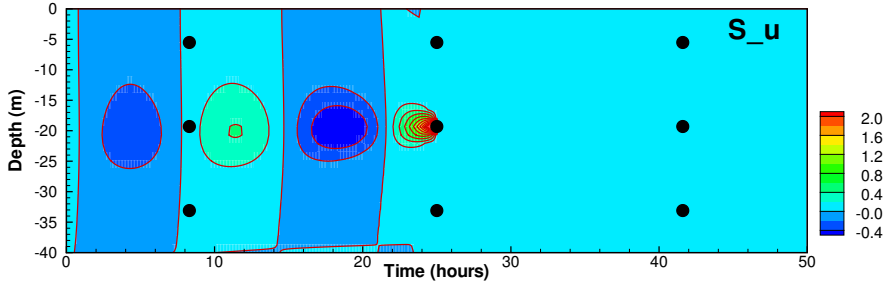
$$\mathbf{u}(z, 0) = \mathbf{u}_0, \tag{153}$$

boundary conditions

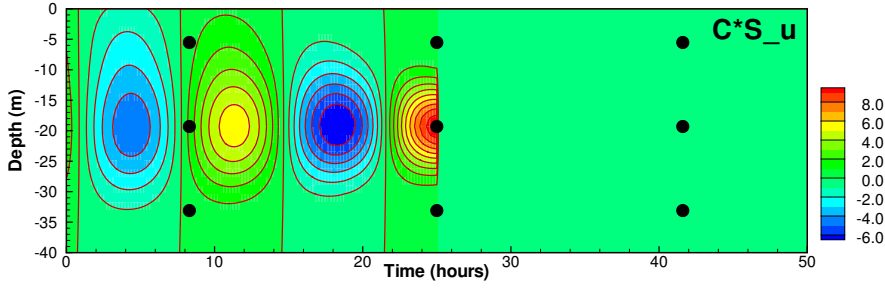
$$A \frac{\partial \mathbf{u}}{\partial z}\bigg|_{z=0} = \left(c_d \sqrt{u_a^2 + v_a^2}\right) \mathbf{u}_a, \tag{154}$$

$$A \frac{\partial \mathbf{u}}{\partial z}\bigg|_{z=-H} = \mathbf{0}, \tag{155}$$

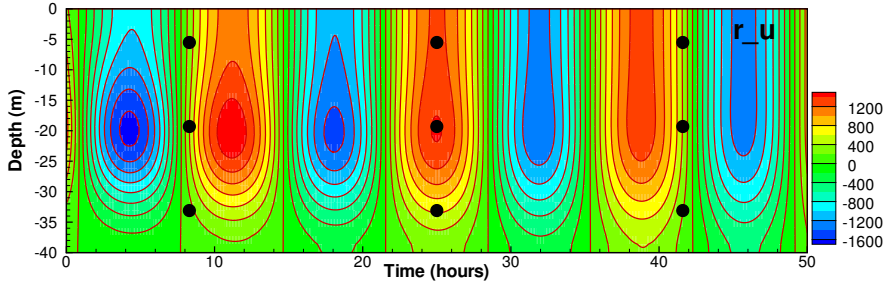
An adjoint representer s



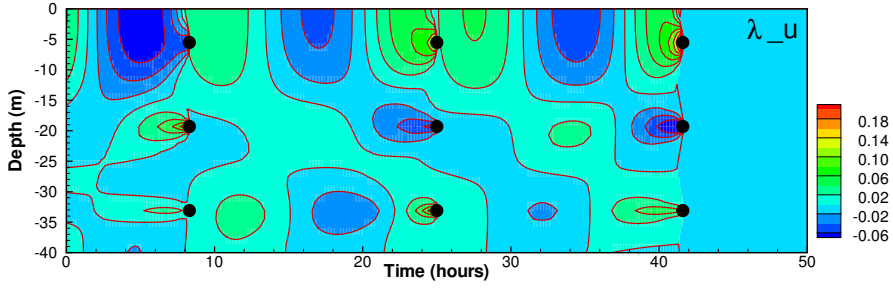
A convolution of an adjoint representer $\mathbf{C}_{qq} \circ \mathbf{s}$



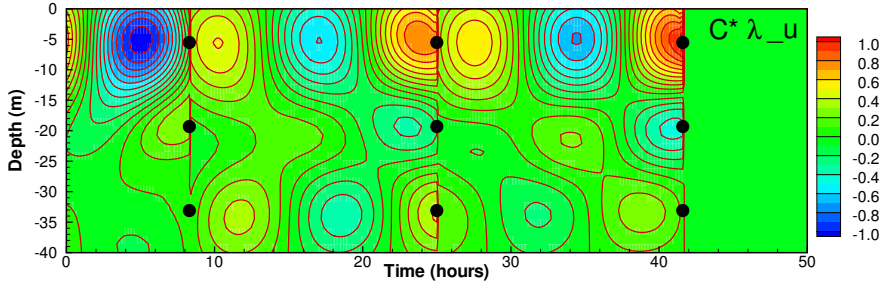
A representer $\mathbf{r}$



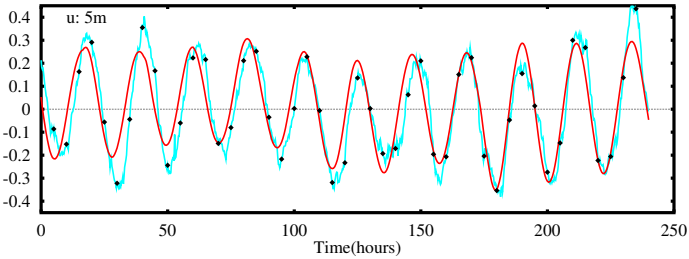
The adjoint λ

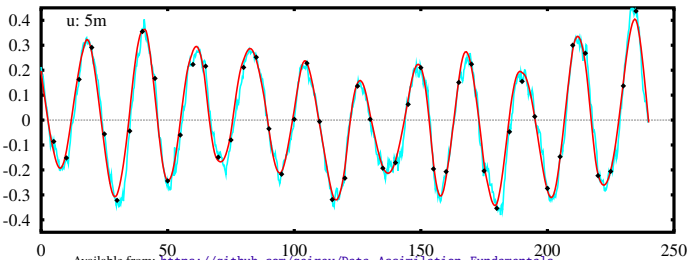


The convolution of the adjoint $\mathbf{C}_{qq} \circ \lambda$



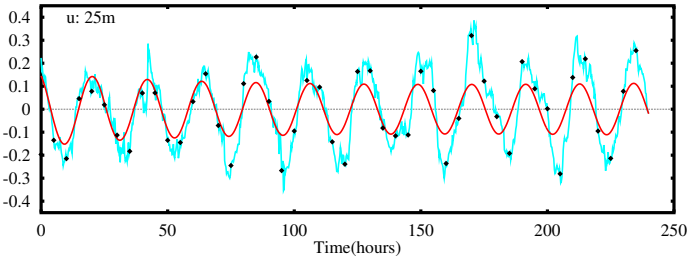
Strong and weak constraint solution

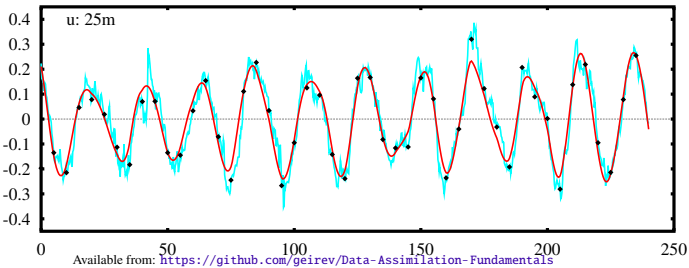




Available from: https://github.com/geirev/Data-Assimilation-Fundamentals

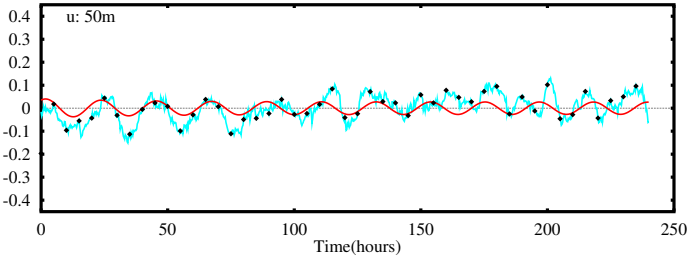
Strong and weak constraint solution

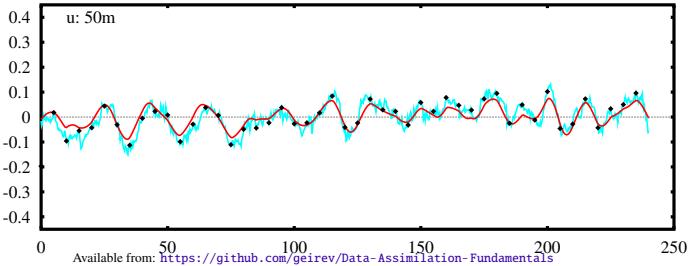




Available from: https://github.com/geirev/Data-Assimilation-Fundamentals

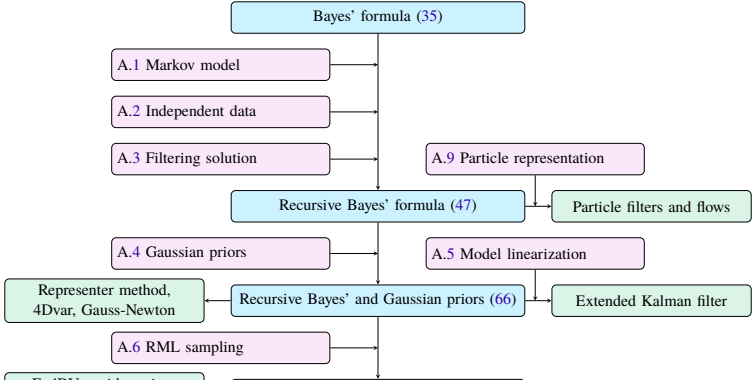
Strong and weak constraint solution



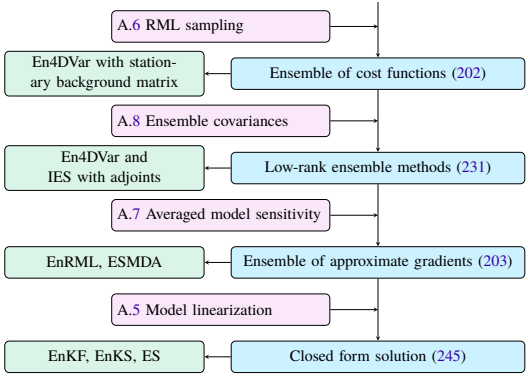


Available from: https://github.com/geirev/Data-Assimilation-Fundamentals

Overview of approximations and methods



Overview of approximations and methods



Kalman filters and 3D-Var

Starting point

Start from the cost-function from Eq. (66)

$$\mathcal{J}(\mathbf{z}) = \frac{1}{2}\left(\mathbf{z}-\mathbf{z}^{\mathrm{f}}\right)^{\mathrm{T}}\mathbf{C}_{zz}^{-1}\left(\mathbf{z}-\mathbf{z}^{\mathrm{f}}\right) + \frac{1}{2}\left(\mathbf{g}(\mathbf{z})-\mathbf{d}\right)^{\mathrm{T}}\mathbf{C}_{dd}^{-1}\left(\mathbf{g}(\mathbf{z})-\mathbf{d}\right), \tag{156}$$

with the gradient in Eq. (67) set to zero

$$\mathbf{C}_{zz}^{-1}\left(\mathbf{z}^{\mathrm{a}}-\mathbf{z}^{\mathrm{f}}\right) + \nabla_{\mathbf{z}}\mathbf{g}\left(\mathbf{z}^{\mathrm{a}}\right)\mathbf{C}_{dd}^{-1}\left(\mathbf{g}\left(\mathbf{z}^{\mathrm{a}}\right)-\mathbf{d}\right) = 0. \tag{157}$$

- 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* $\mathbf{z}^{\mathrm{f}}$,

$$\mathbf{g}(\mathbf{z}) \approx \mathbf{g}(\mathbf{z}^{\mathrm{f}}) + \mathbf{G}(\mathbf{z} - \mathbf{z}^{\mathrm{f}}), \tag{158}$$

and approximate the gradient evaluated at the prior estimate

$$\nabla_{\mathbf{z}}\mathbf{g}(\mathbf{z}) \approx \mathbf{G}^{\mathrm{T}}, \tag{159}$$

where we have defined

$$\mathbf{G}^{\mathrm{T}} = \nabla_{\mathbf{z}}\mathbf{g}(\mathbf{z})\big|_{\mathbf{z}=\mathbf{z}^{\mathrm{f}}}. \tag{160}$$

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

$$\mathbf{M}_k^{\mathrm{T}} = \nabla_{\mathbf{z}}\mathbf{m}(\mathbf{z})\big|_{\mathbf{z}=\mathbf{z}_k} \quad \textit{and} \quad \mathbf{H}^{\mathrm{T}} = \nabla_{\mathbf{m}(\mathbf{z})}\mathbf{h}(\mathbf{m}(\mathbf{z}))\big|_{\mathbf{z}=\mathbf{z}_k}. \tag{161}$$

Solution by linearization

The linearization in Approx. 5 leads to the following form of (157)

$$\mathbf{C}_{zz}^{-1}\left(\mathbf{z}^{\mathrm{a}}-\mathbf{z}^{\mathrm{f}}\right)+\mathbf{G}^{\mathrm{T}}\mathbf{C}_{dd}^{-1}\left(\mathbf{g}(\mathbf{z}^{\mathrm{f}})+\mathbf{G}(\mathbf{z}^{\mathrm{a}}-\mathbf{z}^{\mathrm{f}})-\mathbf{d}\right)=0, \tag{162}$$

or

$$\left(\mathbf{C}_{zz}^{-1}+\mathbf{G}^{\mathrm{T}}\mathbf{C}_{dd}^{-1}\mathbf{G}\right)\left(\mathbf{z}^{\mathrm{a}}-\mathbf{z}^{\mathrm{f}}\right)=\mathbf{G}^{\mathrm{T}}\mathbf{C}_{dd}^{-1}\left(\mathbf{d}-\mathbf{g}(\mathbf{z}^{\mathrm{f}})\right)=0. \tag{163}$$

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

$$\mathbf{z}^{\mathrm{a}}=\mathbf{z}^{\mathrm{f}}+\left(\mathbf{C}_{zz}^{-1}+\mathbf{G}^{\mathrm{T}}\mathbf{C}_{dd}^{-1}\mathbf{G}\right)^{-1}\mathbf{G}^{\mathrm{T}}\mathbf{C}_{dd}^{-1}\left(\mathbf{d}-\mathbf{g}(\mathbf{z}^{\mathrm{f}})\right). \tag{164}$$

Alternative form using Woodbury

Woodbury corollaries

$$\left(\mathbf{C}^{-1}+\mathbf{G}^{\mathrm{T}}\mathbf{D}^{-1}\mathbf{G}\right)^{-1}=\mathbf{C}-\mathbf{C}\mathbf{G}^{\mathrm{T}}(\mathbf{G}\mathbf{C}\mathbf{G}^{\mathrm{T}}+\mathbf{D})^{-1}\mathbf{G}\mathbf{C}, \tag{165}$$

$$\left(\mathbf{G}^{\mathrm{T}}\mathbf{D}^{-1}\mathbf{G}+\mathbf{C}^{-1}\right)^{-1}\mathbf{G}^{\mathrm{T}}\mathbf{D}^{-1}=\mathbf{C}\mathbf{G}^{\mathrm{T}}\left(\mathbf{G}\mathbf{C}\mathbf{G}^{\mathrm{T}}+\mathbf{D}\right)^{-1}, \tag{166}$$

Using Eq. (166) we rewrite Eq. (164) as

Closed form solution by linearization

$$\mathbf{z}^{\mathrm{a}}=\mathbf{z}^{\mathrm{f}}+\mathbf{C}_{zz}\mathbf{G}^{\mathrm{T}}\left(\mathbf{G}\mathbf{C}_{zz}\mathbf{G}^{\mathrm{T}}+\mathbf{C}_{dd}\right)^{-1}\left(\mathbf{d}-\mathbf{g}\left(\mathbf{z}^{\mathrm{f}}\right)\right), \tag{167}$$

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 becomes

$$\mathbf{y} = \mathbf{G}\mathbf{z} = \mathbf{H}\begin{pmatrix} \mathbf{z} \\ \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_K \end{pmatrix} = \mathbf{H}\begin{pmatrix} \mathbf{z} \\ \mathbf{M}_1\mathbf{z} \\ \vdots \\ \mathbf{M}_K \ldots \mathbf{M}_1\mathbf{z} \end{pmatrix} = \mathbf{H}\begin{pmatrix} \mathbf{I} \\ \mathbf{M}_1 \\ \vdots \\ \mathbf{M}_K \ldots \mathbf{M}_1 \end{pmatrix}\mathbf{z} = \mathbf{H}\boldsymbol{\mathcal{M}}\mathbf{z}, \tag{168}$$

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

$$\mathbf{y} = \mathbf{g}(\mathbf{z}) = \mathbf{h}\begin{pmatrix} \mathbf{z} \\ \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_K \end{pmatrix} = \mathbf{h}\begin{pmatrix} \mathbf{z} \\ \mathbf{m}_1(\mathbf{z}) \\ \vdots \\ \mathbf{m}_K(\cdots(\mathbf{m}_2(\mathbf{m}_1(\mathbf{z}))\cdots) \end{pmatrix} = \mathbf{h}(\mathbf{m}(\mathbf{z})) \tag{169}$$

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

Alternative formulation

$\mathbf{G}\mathbf{C}_{zz}$ and $\mathbf{G}\mathbf{C}_{zz}\mathbf{G}^{\mathrm{T}}$ is a propagation of error covariances over the assimilation interval.

Thus, we have

$$\mathbf{C}_{yz} = \mathbf{G}\mathbf{C}_{zz} \tag{170}$$

$$\mathbf{C}_{yy} = \mathbf{G}\mathbf{C}_{zz}\mathbf{G}^{\mathrm{T}} \tag{171}$$

We can then write Eq. (167) as

$$\mathbf{z}^{\mathrm{a}} = \mathbf{z}^{\mathrm{f}} + \mathbf{C}_{zy}\Big(\mathbf{C}_{yy} + \mathbf{C}_{dd}\Big)^{-1}\Big(\mathbf{d} - \mathbf{g}\big(\mathbf{z}^{\mathrm{f}}\big)\Big). \tag{172}$$

Update over the assimilation window

Multiply Eq. (167) by $\boldsymbol{\mathcal{M}}$ to get

$$\boldsymbol{\mathcal{M}}\mathbf{z}^{\mathrm{a}} = \boldsymbol{\mathcal{M}}\mathbf{z}^{\mathrm{f}} + \boldsymbol{\mathcal{M}}\mathbf{C}_{zz}\boldsymbol{\mathcal{M}}^{\mathrm{T}}\mathbf{H}^{\mathrm{T}}\left(\mathbf{G}\mathbf{C}_{zz}\mathbf{G}^{\mathrm{T}} + \mathbf{C}_{dd}\right)^{-1}\left(\mathbf{d} - \mathbf{g}\left(\mathbf{z}^{\mathrm{f}}\right)\right). \tag{173}$$

We can write this equation as

$$\begin{pmatrix} \mathbf{z}^{\mathrm{a}} \\ \mathbf{x}_1^{\mathrm{a}} \\ \vdots \\ \mathbf{x}_K^{\mathrm{a}} \end{pmatrix} = \begin{pmatrix} \mathbf{z}^{\mathrm{f}} \\ \mathbf{x}_1^{\mathrm{f}} \\ \vdots \\ \mathbf{x}_K^{\mathrm{f}} \end{pmatrix} + \begin{pmatrix} \mathbf{C}_{zz} & \cdots & \mathbf{C}_{zx_K} \\ \mathbf{C}_{x_1 z} & \cdots & \mathbf{C}_{x_1 x_K} \\ \vdots & \ddots & \vdots \\ \mathbf{C}_{x_K z} & \cdots & \mathbf{C}_{x_K x_K} \end{pmatrix} \mathbf{H}^{\mathrm{T}}\left(\mathbf{G}\mathbf{C}_{zz}\mathbf{G}^{\mathrm{T}} + \mathbf{C}_{dd}\right)^{-1}\left(\mathbf{d} - \mathbf{g}\left(\mathbf{z}^{\mathrm{f}}\right)\right). \tag{174}$$

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

$$\mathbf{x}_K^{\mathrm{a}} = \mathbf{x}_K^{\mathrm{f}} + \begin{pmatrix} \mathbf{C}_{x_K z} & \cdots & \mathbf{C}_{x_K x_K} \end{pmatrix} \mathbf{H}^{\mathrm{T}}\left(\mathbf{G}\mathbf{C}_{zz}\mathbf{G}^{\mathrm{T}} + \mathbf{C}_{dd}\right)^{-1}\left(\mathbf{d} - \mathbf{g}\left(\mathbf{z}^{\mathrm{f}}\right)\right) \tag{175}$$

$$= \mathbf{x}_K^{\mathrm{f}} + \mathbf{C}_{x_K y}\left(\mathbf{C}_{yy} + \mathbf{C}_{dd}\right)^{-1}\left(\mathbf{d} - \mathbf{g}\left(\mathbf{z}^{\mathrm{f}}\right)\right). \tag{176}$$

3DVar problem

- Prior and measurements available at beginning of DA window.
- Constant in time $\mathbf{C}_{zz}$.

3DVar costfunction

$$\mathcal{J}(\mathbf{z}) = \frac{1}{2}\left(\mathbf{z}-\mathbf{z}^{\mathrm{f}}\right)^{\mathrm{T}} \mathbf{C}_{zz}^{-1} \left(\mathbf{z}-\mathbf{z}^{\mathrm{f}}\right) + \frac{1}{2}\left(\mathbf{h}(\mathbf{z})-\mathbf{d}\right)^{\mathrm{T}} \mathbf{C}_{dd}^{-1} \left(\mathbf{h}(\mathbf{z})-\mathbf{d}\right). \tag{177}$$

3DVar solution by Gauss-Newton

Gradient

$$\nabla_{\mathbf{z}} \mathcal{J}(\mathbf{z}) = \mathbf{C}_{zz}^{-1}\left(\mathbf{z} - \mathbf{z}^{\mathrm{f}}\right) + \mathbf{H}^{\mathrm{T}}\,\mathbf{C}_{dd}^{-1}\left(\mathbf{h}(\mathbf{z}) - \mathbf{d}\right), \tag{178}$$

Hessian

$$\nabla_{\mathbf{z}} \nabla_{\mathbf{z}} \mathcal{J}(\mathbf{z}) \approx \mathbf{C}_{zz}^{-1} + \mathbf{H}^{\mathrm{T}} \mathbf{C}_{dd}^{-1} \mathbf{H}. \tag{179}$$

Gauss-Newton iterations

$$\begin{aligned} \mathbf{z}^{i+1} &= \mathbf{z}^{i} - \gamma^{i}\left(\mathbf{C}_{zz}^{-1} + \mathbf{H}^{i\mathrm{T}}\mathbf{C}_{dd}^{-1}\mathbf{H}^{i}\right)^{-1}\left(\mathbf{C}_{zz}^{-1}\left(\mathbf{z}^{i} - \mathbf{z}^{\mathrm{f}}\right) + \mathbf{H}^{i\mathrm{T}}\,\mathbf{C}_{dd}^{-1}\left(\mathbf{h}(\mathbf{z}^{i}) - \mathbf{d}\right)\right) \\ &= \mathbf{z}^{i} - \gamma^{i}(\mathbf{z}^{i} - \mathbf{z}^{\mathrm{f}}) + \gamma^{i}\mathbf{C}_{zz}\mathbf{H}^{\mathrm{T}}\left(\mathbf{H}\mathbf{C}_{zz}\mathbf{H}^{\mathrm{T}} + \mathbf{C}_{dd}\right)^{-1}\left(\mathbf{H}(\mathbf{z}^{i} - \mathbf{z}^{\mathrm{f}}) - \left(\mathbf{h}(\mathbf{z}^{i}) - \mathbf{d}\right)\right). \end{aligned} \tag{180}$$

Kalman Filter (KF) state update

- Prior and measurements available at beginning of DA window.
- Linear dynamics and measurement operator.

$$\mathcal{J}(\mathbf{z}) = \frac{1}{2}\left(\mathbf{z}-\mathbf{z}^{\mathrm{f}}\right)^{\mathrm{T}}\mathbf{C}_{zz}^{-1}\left(\mathbf{z}-\mathbf{z}^{\mathrm{f}}\right) + \frac{1}{2}\left(\mathbf{H}\mathbf{z}-\mathbf{d}\right)^{\mathrm{T}}\mathbf{C}_{dd}^{-1}\left(\mathbf{H}\mathbf{z}-\mathbf{d}\right). \tag{181}$$

Gradient of the cost function equal to zero,

$$\mathbf{C}_{zz}^{-1}\left(\mathbf{z}^{\mathrm{a}}-\mathbf{z}^{\mathrm{f}}\right) + \mathbf{H}^{\mathrm{T}}\mathbf{C}_{dd}^{-1}\left(\mathbf{H}\mathbf{z}^{\mathrm{a}}-\mathbf{d}\right) = 0, \tag{182}$$

The Kalman filter state update

$$\mathbf{z}^{\mathrm{a}} = \mathbf{z}^{\mathrm{f}} + \mathbf{C}_{zz}\mathbf{H}^{\mathrm{T}}\left(\mathbf{H}\mathbf{C}_{zz}\mathbf{H}^{\mathrm{T}} + \mathbf{C}_{dd}\right)^{-1}\left(\mathbf{d}-\mathbf{H}\mathbf{z}^{\mathrm{f}}\right), \tag{183}$$

Kalman Filter (KF) error covariance update

Cost function

$$\mathcal{J}(\mathbf{z}) = \frac{1}{2}\left(\mathbf{z}-\mathbf{z}^{\mathrm{f}}\right)^{\mathrm{T}}\mathbf{C}_{zz}^{-1}\left(\mathbf{z}-\mathbf{z}^{\mathrm{f}}\right) + \frac{1}{2}\left(\mathbf{H}\mathbf{z}-\mathbf{d}\right)^{\mathrm{T}}\mathbf{C}_{dd}^{-1}\left(\mathbf{H}\mathbf{z}-\mathbf{d}\right). \tag{184}$$

Hessian

$$\nabla_z\nabla_z\mathcal{J}(\mathbf{z}) = \mathbf{C}_{zz}^{-1} + \mathbf{H}\mathbf{C}_{dd}^{-1}\mathbf{H}^{\mathrm{T}}. \tag{185}$$

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

$$\mathcal{J}(\mathbf{z}) = \frac{1}{2}\left(\mathbf{z}-\mathbf{z}^{\mathrm{a}}\right)^{\mathrm{T}}\left(\mathbf{C}_{zz}^{\mathrm{a}}\right)^{-1}\left(\mathbf{z}-\mathbf{z}^{\mathrm{a}}\right) + \text{constant} \tag{186}$$

Second derivative

$$\nabla_z\nabla_z\mathcal{J}(\mathbf{z}) = \left(\mathbf{C}_{zz}^{\mathrm{a}}\right)^{-1}. \tag{187}$$

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

$$\left(\mathbf{C}_{zz}^{\mathrm{a}}\right)^{-1} = \mathbf{C}_{zz}^{-1} + \mathbf{H}\mathbf{C}_{dd}^{-1}\mathbf{H}^{\mathrm{T}}. \tag{188}$$

Kalman Filter error covariance update equation

Rewrite using the matrix identity (165) to find

The Kalman filter error-covariance update

$$\mathbf{C}_{zz}^{\mathrm{a}} = \mathbf{C}_{zz} - \mathbf{C}_{zz}\mathbf{H}^{\mathrm{T}}\left(\mathbf{H}\mathbf{C}_{zz}\mathbf{H}^{\mathrm{T}} + \mathbf{C}_{dd}\right)^{-1}\mathbf{H}\mathbf{C}_{zz}. \tag{189}$$

Kalman Filter (KF)

Forward model

$$\mathbf{z}_{k+1} = \mathbf{M}\mathbf{z}_k, \tag{190}$$

$$\mathbf{C}_{zz,k+1} = \mathbf{M}\mathbf{C}_{zz,k}\mathbf{M}^{\mathrm{T}} + \mathbf{C}_{qq}. \tag{191}$$

Update equations

$$\mathbf{z}^{\mathrm{a}} = \mathbf{z}^{\mathrm{f}} + \mathbf{C}_{zz}\mathbf{H}^{\mathrm{T}}\left(\mathbf{H}\mathbf{C}_{zz}\mathbf{H}^{\mathrm{T}} + \mathbf{C}_{dd}\right)^{-1}\left(\mathbf{d} - \mathbf{H}\mathbf{z}^{\mathrm{f}}\right), \tag{192}$$

$$\mathbf{C}_{zz}^{\mathrm{a}} = \mathbf{C}_{zz} - \mathbf{C}_{zz}\mathbf{H}^{\mathrm{T}}\left(\mathbf{H}\mathbf{C}_{zz}\mathbf{H}^{\mathrm{T}} + \mathbf{C}_{dd}\right)^{-1}\mathbf{H}\mathbf{C}_{zz}. \tag{193}$$

One often defines the “Kalman gain matrix”

$$\mathbf{K} = \mathbf{C}_{zz}\mathbf{H}^{\mathrm{T}}\left(\mathbf{H}\mathbf{C}_{zz}\mathbf{H}^{\mathrm{T}} + \mathbf{C}_{dd}\right)^{-1}, \tag{194}$$

Extension to nonlinear problem

- Allows for nonlinear dynamics and measurement operator.

Gradient Eq. (178) becomes

$$\mathbf{C}_{zz}^{-1}\left(\mathbf{z}^{\mathrm{a}}-\mathbf{z}^{\mathrm{f}}\right)+\mathbf{H}^{\mathrm{T}}\,\mathbf{C}_{dd}^{-1}\,\left(\mathbf{h}(\mathbf{z}^{\mathrm{f}})+\mathbf{H}(\mathbf{z}^{\mathrm{a}}-\mathbf{z}^{\mathrm{f}})-\mathbf{d}\right). \tag{195}$$

Explicit solution

$$\mathbf{z}^{\mathrm{a}}=\mathbf{z}^{\mathrm{f}}+\mathbf{C}_{zz}\mathbf{H}^{\mathrm{T}}\left(\mathbf{H}\mathbf{C}_{zz}\mathbf{H}^{\mathrm{T}}+\mathbf{C}_{dd}\right)^{-1}\left(\mathbf{d}-\mathbf{h}\left(\mathbf{z}^{\mathrm{f}}\right)\right). \tag{196}$$

Extended Kalman Filter

$$\mathbf{z}_{k+1} = \mathbf{m}(\mathbf{z}_k) \tag{197}$$

$$\mathbf{C}_{zz,k+1} = \mathbf{M}_k \mathbf{C}_{zz,k} \mathbf{M}_k^{\mathrm{T}} + \mathbf{C}_{qq}. \tag{198}$$

Update equations

$$\mathbf{z}^{\mathrm{a}} = \mathbf{z}^{\mathrm{f}} + \mathbf{C}_{zz}\mathbf{H}^{\mathrm{T}}\left(\mathbf{H}\mathbf{C}_{zz}\mathbf{H}^{\mathrm{T}} + \mathbf{C}_{dd}\right)^{-1}\left(\mathbf{d} - \mathbf{h}(\mathbf{z}^{\mathrm{f}})\right). \tag{199}$$

$$\mathbf{C}_{zz}^{\mathrm{a}} = \mathbf{C}_{zz} - \mathbf{C}_{zz}\mathbf{H}^{\mathrm{T}}\left(\mathbf{H}\mathbf{C}_{zz}\mathbf{H}^{\mathrm{T}} + \mathbf{C}_{dd}\right)^{-1}\mathbf{H}\mathbf{C}_{zz}. \tag{200}$$

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. (202).

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

$$\mathbf{z}_j^{\mathrm{f}} \leftarrow \mathcal{N}(\mathbf{z}^{\mathrm{f}}, \mathbf{C}_{zz}) \quad \text{and} \quad \mathbf{d}_j \leftarrow \mathcal{N}(\mathbf{d}, \mathbf{C}_{dd}) \tag{201}$$

Ensemble of cost functions

$$\mathcal{J}(\mathbf{z}_j) = \frac{1}{2}\left(\mathbf{z}_j - \mathbf{z}_j^{\mathrm{f}}\right)^{\mathrm{T}} \mathbf{C}_{zz}^{-1}\left(\mathbf{z}_j - \mathbf{z}_j^{\mathrm{f}}\right) + \frac{1}{2}\left(\mathbf{g}(\mathbf{z}_j) - \mathbf{d}_j\right)^{\mathrm{T}} \mathbf{C}_{dd}^{-1}\left(\mathbf{g}(\mathbf{z}_j) - \mathbf{d}_j\right), \tag{202}$$

Ensemble of gradients set to zero

$$\mathbf{C}_{zz}^{-1}\left(\mathbf{z}_j - \mathbf{z}_j^{\mathrm{f}}\right) + \nabla_{\mathbf{z}}\mathbf{g}(\mathbf{z}_j)\mathbf{C}_{dd}^{-1}\left(\mathbf{g}(\mathbf{z}_j) - \mathbf{d}_j\right) = 0. \tag{203}$$

Thus, we must solve N independent minimizations.

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

Ensemble of incremental 4DVars

$$\mathcal{J}(\delta\mathbf{z}_j) = \frac{1}{2}\left(\delta\mathbf{z}_j - \boldsymbol{\xi}_j^i\right)^{\mathrm{T}} \mathbf{C}_{zz}^{-1} \left(\delta\mathbf{z}_j - \boldsymbol{\xi}_j^i\right) + \frac{1}{2}\left(\mathbf{G}_j^i\delta\mathbf{z}_j - \boldsymbol{\eta}_j^i\right)^{\mathrm{T}} \mathbf{C}_{dd}^{-1} \left(\mathbf{G}_j^i\delta\mathbf{z}_j - \boldsymbol{\eta}_j^i\right). \tag{204}$$

Ensemble of GN iterations

$$\mathbf{z}_j^{i+1} = \mathbf{z}_j^i - \gamma\left(\mathbf{C}_{zz}^{-1} + \mathbf{G}_j^{i\mathrm{T}}\mathbf{C}_{dd}^{-1}\mathbf{G}_j^i\right)^{-1}\left(\mathbf{C}_{zz}^{-1}\left(\mathbf{z}_j^i - \mathbf{z}_j^{\mathrm{f}}\right) + \mathbf{G}_j^{i\mathrm{T}}\mathbf{C}_{dd}^{-1}\left(\mathbf{g}\left(\mathbf{z}_j^i\right) - \mathbf{d}_j\right)\right), \tag{205}$$

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

$$\mathbf{z}_j^{\mathrm{a}} = \mathbf{z}_j^{\mathrm{f}} + \mathbf{C}_{zz}\mathbf{G}_j^{\mathrm{T}}\left(\mathbf{G}_j\mathbf{C}_{zz}\mathbf{G}_j^{\mathrm{T}} + \mathbf{C}_{dd}\right)^{-1}\left(\mathbf{d}_j - \mathbf{g}(\mathbf{z}_j^{\mathrm{f}})\right). \tag{206}$$

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. (206) and $\mathbf{G}_j^i$ in Eq. (205) as the sensitivity matrix in linear regression and represent them using the definition

$$\mathbf{G}_j \approx \mathbf{G} \triangleq \mathbf{C}_{yz}\mathbf{C}_{zz}^{-1}. \tag{207}$$

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

$$g(x) \approx g(\overline{x}) + g'(\overline{x})(x - \overline{x}). \tag{208}$$

$$\begin{aligned} C^{\mathrm{e}}_{xy} &= \overline{\left(x - \overline{x}\right)\left(y - \overline{y}\right)} \\ &= \overline{\left(x - \overline{x}\right)\left(g(x) - \overline{g(x)}\right)} \\ &\approx \overline{\left(x - \overline{x}\right)\left(g(\overline{x}) + g'(\overline{x})(x - \overline{x}) - \overline{\left(g(\overline{x}) + g'(\overline{x})(x - \overline{x})\right)}\right)} \\ &= g'(\overline{x})\,\overline{(x - \overline{x})^2} \\ &= g'(\overline{x})\,C^{\mathrm{e}}_{xx}, \end{aligned} \tag{209}$$

Gauss-Newton iterations with averaged model sensitivity

Rewrite the Gauss-Newton iteration in Eq. (205) as

$$\mathbf{z}_j^{i+1} = \mathbf{z}_j^i - \gamma\left(\mathbf{C}_{zz}^{-1} + \mathbf{G}_j^{i\mathrm{T}}\mathbf{C}_{dd}^{-1}\mathbf{G}_j^i\right)^{-1}\left(\mathbf{C}_{zz}^{-1}\left(\mathbf{z}_j^i - \mathbf{z}_j^{\mathrm{f}}\right) + \mathbf{G}_j^{i\mathrm{T}}\mathbf{C}_{dd}^{-1}\left(\mathbf{g}\left(\mathbf{z}_j^i\right) - \mathbf{d}_j\right)\right), \tag{210}$$

$$\approx \mathbf{z}_j^i - \gamma\left(\mathbf{C}_{zz}^{-1} + \mathbf{G}^{i\mathrm{T}}\mathbf{C}_{dd}^{-1}\mathbf{G}^i\right)^{-1}\left(\mathbf{C}_{zz}^{-1}\left(\mathbf{z}_j^i - \mathbf{z}_j^{\mathrm{f}}\right) + \mathbf{G}^{i\mathrm{T}}\mathbf{C}_{dd}^{-1}\left(\mathbf{g}(\mathbf{z}_i) - \mathbf{d}_j\right)\right) \tag{211}$$

$$= \mathbf{z}_j^i - \gamma\left(\mathbf{z}_j^i - \mathbf{z}_j^{\mathrm{f}}\right) + \gamma\mathbf{C}_{zz}\mathbf{G}^{i\mathrm{T}}\left(\mathbf{G}^i\mathbf{C}_{zz}\mathbf{G}^{i\mathrm{T}} + \mathbf{C}_{dd}\right)^{-1}\left(\mathbf{G}^i\left(\mathbf{z}_j^i - \mathbf{z}_j^{\mathrm{f}}\right) - \left(\mathbf{g}(\mathbf{z}_j^i) - \mathbf{d}_j\right)\right), \tag{212}$$

where we have used the corollaries from Eqs. (165) and (166).

Expression in terms of covariances

We have

$$\mathbf{G}\mathbf{C}_{zz} = \mathbf{C}_{yz}, \tag{213}$$

$$\mathbf{G}\mathbf{C}_{zz}\mathbf{G}^{\mathrm{T}} = \mathbf{C}_{yz}\mathbf{C}_{zz}^{-1}\mathbf{C}_{zy} \neq \mathbf{C}_{yy}. \tag{214}$$

Gauss-Newton Eq. (212)

$$\mathbf{z}_j^{i+1} = \mathbf{z}_j^i - \gamma\left(\mathbf{z}_j^i - \mathbf{z}_j^{\mathrm{f}}\right) + \gamma\mathbf{C}_{zy}\left(\mathbf{C}_{yz}\mathbf{C}_{zz}^{-1}\mathbf{C}_{zy} + \mathbf{C}_{dd}\right)^{-1}\left(\mathbf{C}_{yz}\mathbf{C}_{zz}^{-1}\left(\mathbf{z}_j^i - \mathbf{z}_j^{\mathrm{f}}\right) - \left(\mathbf{g}(\mathbf{z}_j^i) - \mathbf{d}_j\right)\right), \tag{215}$$

EKF update Eq. (206)

$$\mathbf{z}_j = \mathbf{z}_j^{\mathrm{f}} + \mathbf{C}_{zy}\left(\mathbf{C}_{yz}\mathbf{C}_{zz}^{-1}\mathbf{C}_{zy} + \mathbf{C}_{dd}\right)^{-1}\left(\mathbf{d}_j - \mathbf{g}\left(\mathbf{z}_j^{\mathrm{f}}\right)\right). \tag{216}$$

First GN-step equals EKF update

First step of Gauss-Newton Eq. (212) with $\gamma = 1$ and $i = 1$:

$$\mathbf{z}_j^{i+1} = \mathbf{z}_j^i - \left(\mathbf{z}_j^i - \mathbf{z}_j^{\mathrm{f}}\right) + \mathbf{C}_{zy}\left(\mathbf{C}_{yz}\mathbf{C}_{zz}^{-1}\mathbf{C}_{zy} + \mathbf{C}_{dd}\right)^{-1}\left(\mathbf{C}_{yz}\mathbf{C}_{zz}^{-1}\left(\mathbf{z}_j^i - \mathbf{z}_j^{\mathrm{f}}\right) - \left(\mathbf{g}(\mathbf{z}_j^i) - \mathbf{d}_j\right)\right), \tag{217}$$

$$= \mathbf{z}_j^{\mathrm{f}} + \mathbf{C}_{zy}\left(\mathbf{C}_{yz}\mathbf{C}_{zz}^{-1}\mathbf{C}_{zy} + \mathbf{C}_{dd}\right)^{-1}\left(\mathbf{d}_j - \mathbf{g}(\mathbf{z}_j^{\mathrm{f}})\right). \tag{218}$$

Low-rank 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.

State vector ensemble

Define

$$\mathbf{Z} = \left(\mathbf{z}_1,\ \mathbf{z}_2,\ \ldots\ ,\ \mathbf{z}_N\right). \tag{219}$$

Define the projection $\mathbf{\Pi} \in \Re^{N\times N}$ with $\mathbf{1}^{\mathrm{T}} = (1, 1, \ldots, 1) \in \Re^N$

$$\mathbf{\Pi} = \left(\mathbf{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^{\mathrm{T}}\right)\Big/\sqrt{N-1}. \tag{220}$$

State vector ensemble anomalies

$$\mathbf{A} = \mathbf{Z}\mathbf{\Pi}. \tag{221}$$

State vector ensemble covariance

$$\overline{\mathbf{C}}_{zz} = \mathbf{A}\mathbf{A}^{\mathrm{T}}. \tag{222}$$

Measurement ensemble

Define ensemble of perturbed measurements

$$\mathbf{D} = \mathbf{d}\mathbf{1}^{\mathrm{T}} + \sqrt{N-1}\mathbf{E}. \tag{223}$$

Measurement perturbations

$$\mathbf{E} = \Big(\mathbf{e}_1,\ \mathbf{e}_2,\ \ldots\ ,\ \mathbf{e}_N\Big). \tag{224}$$

Measurement-error ensemble covariance matrix

$$\overline{\mathbf{C}}_{dd} = \mathbf{E}\mathbf{E}^{\mathrm{T}}. \tag{225}$$

Model ensemble predictions

Ensemble of model-predicted measurements

$$\mathbf{\Upsilon} = \mathbf{g}(\mathbf{Z}), \tag{226}$$

with anomalies

$$\mathbf{Y} = \mathbf{\Upsilon}\mathbf{\Pi}, \tag{227}$$

and we have

$$\overline{\mathbf{C}_{yy}} = \mathbf{Y}\mathbf{Y}^{\mathrm{T}} \quad \text{and} \quad \overline{\mathbf{C}_{yz}} = \mathbf{Y}\mathbf{A}^{\mathrm{T}} \tag{228}$$

Solution is confined to ensemble subspace

Cost function (202)

$$\mathcal{J}(\mathbf{z}_j) = \frac{1}{2}\left(\mathbf{z}_j - \mathbf{z}_j^{\mathrm{f}}\right)^{\mathrm{T}} \overline{\mathbf{C}}_{zz}^{-1} \left(\mathbf{z}_j - \mathbf{z}_j^{\mathrm{f}}\right) + \frac{1}{2}\left(\mathbf{g}(\mathbf{z}_j) - \mathbf{d}_j\right)^{\mathrm{T}} \overline{\mathbf{C}}_{dd}^{-1} \left(\mathbf{g}(\mathbf{z}_j) - \mathbf{d}_j\right), \tag{229}$$

From Eq. (215) we search for the solution in the ensemble subspace

$$\mathbf{Z}^{\mathrm{a}} = \mathbf{Z}^{\mathrm{f}} + \mathbf{A}\mathbf{W}. \tag{230}$$

Cost function in ensemble subspace

$$\mathcal{J}(\mathbf{w}_j) = \frac{1}{2}\mathbf{w}_j^{\mathrm{T}}\mathbf{w}_j + \frac{1}{2}\left(\mathbf{g}\left(\mathbf{z}_j^{\mathrm{f}} + \mathbf{A}\mathbf{w}_j\right) - \mathbf{d}_j\right)^{\mathrm{T}} \overline{\mathbf{C}}_{dd}^{-1} \left(\mathbf{g}\left(\mathbf{z}_j^{\mathrm{f}} + \mathbf{A}\mathbf{w}_j\right) - \mathbf{d}_j\right), \tag{231}$$

Gradient and Hessian

Gradient

$$\nabla_{\mathbf{w}}\mathcal{J}(\mathbf{w}_j) = \mathbf{w}_j + \left(\mathbf{G}_j\mathbf{A}\right)^{\mathrm{T}}\overline{\mathbf{C}}_{dd}^{-1}\left(\mathbf{g}\left(\mathbf{z}_j^{\mathrm{f}} + \mathbf{A}\mathbf{w}_j\right) - \mathbf{d}_j\right), \tag{232}$$

Approximate Hessian

$$\nabla_{\mathbf{w}}\nabla_{\mathbf{w}}\mathcal{J}\left(\mathbf{w}_j\right) \approx \mathbf{I} + \left(\mathbf{G}_j\mathbf{A}\right)^{\mathrm{T}}\overline{\mathbf{C}}_{dd}^{-1}\left(\mathbf{G}_j\mathbf{A}\right). \tag{233}$$

Tangent-linear model

$$\mathbf{G}_j = \left(\nabla_{\mathbf{w}}\mathbf{g}|_{\mathbf{z}_j^{\mathrm{f}}+\mathbf{A}\mathbf{w}_j}\right)^{\mathrm{T}} \in \mathfrak{R}^{m\times n}, \tag{234}$$

Gauss-Newton iterations in the ensemble subspace

Gauss-Newton iterations

$$\mathbf{w}_j^{i+1} = \mathbf{w}_j^i - \gamma\left(\mathbf{I} + \left(\mathbf{G}_j^i\mathbf{A}\right)^{\mathrm{T}}\overline{\mathbf{C}}_{dd}^{-1}\left(\mathbf{G}_j^i\mathbf{A}\right)\right)^{-1}\left(\mathbf{w}_j^i + \left(\mathbf{G}_j^i\mathbf{A}\right)^{\mathrm{T}}\overline{\mathbf{C}}_{dd}^{-1}\left(\mathbf{g}\left(\mathbf{z}_j^{\mathrm{f}} + \mathbf{A}\mathbf{w}_j^i\right) - \mathbf{d}_j\right)\right), \tag{235}$$

or using Woodbury we get

$$\mathbf{w}_j^{i+1} = \mathbf{w}_j^i - \gamma\left\{\mathbf{w}_j^i - \left(\mathbf{G}_j^i\mathbf{A}\right)^{\mathrm{T}}\left(\left(\mathbf{G}_j^i\mathbf{A}\right)\left(\mathbf{G}_j^i\mathbf{A}\right)^{\mathrm{T}} + \overline{\mathbf{C}}_{dd}\right)^{-1}\left(\left(\mathbf{G}_j^i\mathbf{A}\right)\mathbf{w}_j^i + \mathbf{d}_j - \mathbf{g}\left(\mathbf{z}_j^{\mathrm{f}} + \mathbf{A}\mathbf{w}_j^i\right)\right)\right\}. \tag{236}$$

with linear regression approximation for model sensitivities

$$\mathbf{G}_j^i \approx \overline{\mathbf{G}}^i \triangleq \overline{\mathbf{C}}_{yz}^i\overline{\mathbf{C}}_{zz}^{i\ \dagger} = \mathbf{Y}^i\mathbf{A}^{i\dagger}, \tag{237}$$

$\mathbf{G}_j^i\mathbf{A}$

Replace $\mathbf{G}_j^i$ with average sensitivity $\mathbf{G}^i$

Define the linear regression

$$\overline{\mathbf{G}}^i = \mathbf{Y}^i \mathbf{A}^{i\dagger}$$

Write

$$\begin{aligned}
\mathbf{G}_j^i\mathbf{A} \approx \overline{\mathbf{G}}^i\mathbf{A} &= \mathbf{Y}^i\mathbf{A}^{i\dagger}\mathbf{A} \\
&= \mathbf{Y}^i\mathbf{A}^{i\dagger}\mathbf{A}^i\boldsymbol{\Omega}^{i-1} && \mathbf{A} = \mathbf{A}^i\boldsymbol{\Omega}^{i-1} \\
&= \mathbf{Y}_i\boldsymbol{\Omega}_i^{-1} = \mathbf{S}_i && \text{when } n \geq N-1
\end{aligned}$$

$$\mathbf{Y}_i = \mathbf{g}(\mathbf{X}_i)\boldsymbol{\Pi}$$

$$\boldsymbol{\Omega}_i = \mathbf{I} + \mathbf{W}_i\boldsymbol{\Pi}$$

Definition of $\boldsymbol{\Omega}$

We have

$$\mathbf{Z}^i = \mathbf{Z}^{\mathrm{f}} + \mathbf{A}\mathbf{W}^i \tag{238}$$

$$= \mathbf{Z}^{\mathrm{f}}\left(\mathbf{I} + \boldsymbol{\Pi}\mathbf{W}^i\right) \tag{239}$$

$$= \mathbf{Z}^{\mathrm{f}}\left(\mathbf{I} + \left(\mathbf{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^{\mathrm{T}}\right)\mathbf{W}^i/\sqrt{N-1}\right) \tag{240}$$

$$= \mathbf{Z}^{\mathrm{f}}\left(\mathbf{I} + \mathbf{W}^i\right) \tag{241}$$

using that

$$\mathbf{1}^{\mathrm{T}}\mathbf{W}^i = 0 \tag{242}$$

The orthogonal projection $\mathbf{Y}^i\mathbf{A}^{i\dagger}\mathbf{A}^i$

Nonlinear case with $n \geq N-1$

$$\mathbf{Y}^i\mathbf{A}^{i\dagger}\mathbf{A}^i = \mathbf{Y}^i\left(\mathbf{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^{\mathrm{T}}\right) = \mathbf{Y}^i$$

Nonlinear case with $n < N-1$

$$\mathbf{Y}^i\mathbf{A}^{i\dagger}\mathbf{A}^i = \mathbf{Y}^i\mathbf{V}\begin{pmatrix}\mathbf{I}_{n\times n} & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{pmatrix}\mathbf{V}^{\mathrm{T}}.$$

Linear case

$$\begin{aligned}\mathbf{Y}^i\mathbf{A}^{i\dagger}\mathbf{A}^i &= \mathbf{H}\mathbf{M}\mathbf{X}^i\mathbf{\Pi}\mathbf{A}^{i\dagger}\mathbf{A}^i\\ &= \mathbf{H}\mathbf{M}\mathbf{A}^i\mathbf{A}^{i\dagger}\mathbf{A}^i\\ &= \mathbf{H}\mathbf{M}\mathbf{A}^i = \mathbf{Y}^i\end{aligned}$$

Final iteration

We can now write the iteration of Eq. (236) in matrix form as

$$\mathbf{W}^0 = \mathbf{0} \tag{243}$$

$$\mathbf{W}^{i+1} = \mathbf{W}^i - \gamma\Big(\mathbf{W}^i - \mathbf{S}^{i\mathrm{T}}\Big(\mathbf{S}^i\mathbf{S}^{i\mathrm{T}} + \overline{\mathbf{C}}_{dd}\Big)^{-1}\Big(\mathbf{S}^i\mathbf{W}^i + \mathbf{D} - \mathbf{g}(\mathbf{Z}^i)\Big)\Big), \tag{244}$$

Subspace EnRML

Input: $\mathbf{Z} \in \mathbb{R}^{n \times N}$ ▷ Prior state-vector ensemble
Input: $\mathbf{D} \in \mathbb{R}^{m \times N}$ ▷ Perturbed measurements
$\mathbf{W}^{(0)} = 0$ ▷ $\mathbf{W} \in \mathbb{R}^{N \times N}$
$\mathbf{\Pi} = \left(\mathbf{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^{\mathrm{T}}\right) / \sqrt{N-1}$ ▷ $\mathbf{\Pi} \in \mathbb{R}^{N \times N}$
$\mathbf{E} = \mathbf{D}\mathbf{\Pi}$ ▷ $\mathbf{E} \in \mathbb{R}^{m \times N}$
$i = 0$
repeat
  $\mathbf{Y}^i = \mathbf{g}(\mathbf{Z}^i)\mathbf{\Pi}$ ▷ $\mathbf{Y} \in \mathbb{R}^{m \times N}$
  if $n < N - 1$ **then**
    $\mathbf{Y}^i = \mathbf{Y}^i \mathbf{A}^{i\dagger} \mathbf{A}^i$
  end if
  $\mathbf{\Omega}^i = \mathbf{I} + \mathbf{W}^i \mathbf{\Pi}$ ▷ $\mathbf{\Omega} \in \mathbb{R}^{N \times N}$
  $\mathbf{S}^i = \mathbf{Y}^i \mathbf{\Omega}^{i-1}$ ▷ $\mathbf{S} \in \mathbb{R}^{m \times N}$
  $\mathbf{W}^{i+1} = \mathbf{W}^i - \gamma\left(\mathbf{W}^i - \mathbf{S}^{i\mathrm{T}}\left(\mathbf{S}^i\mathbf{S}^{i\mathrm{T}} + \mathbf{E}\mathbf{E}^{\mathrm{T}}\right)^{-1}\left(\mathbf{S}^i\mathbf{W}^i + \mathbf{D} - \mathbf{g}(\mathbf{Z}^i)\right)\right)$
  $\mathbf{Z}^{i+1} = \mathbf{Z}\left(\mathbf{I} + \mathbf{W}^{i+1} / \sqrt{N-1}\right)$
  $i = i + 1$
until convergence

Ensemble KF update

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

$$\mathbf{Z}^{\mathrm{a}} = \mathbf{Z}^{\mathrm{f}} + \mathbf{A}\mathbf{Y}^{\mathrm{T}}\left(\mathbf{Y}\mathbf{Y}^{\mathrm{T}} + \mathbf{E}\mathbf{E}^{\mathrm{T}}\right)^{-1}\left(\mathbf{D} - \mathbf{g}(\mathbf{Z}^{\mathrm{f}})\right) \tag{245}$$

Alternative interpretation using

$$\mathbf{W} = \mathbf{Y}^{\mathrm{T}}\left(\mathbf{Y}\mathbf{Y}^{\mathrm{T}} + \mathbf{E}\mathbf{E}^{\mathrm{T}}\right)^{-1}\left(\mathbf{D} - \mathbf{g}(\mathbf{Z}^{\mathrm{f}})\right), \tag{246}$$

to get

$$\mathbf{Z}^{\mathrm{a}} = \mathbf{Z}^{\mathrm{f}}\left(\mathbf{I} + \mathbf{W}/\sqrt{N-1}\right) \tag{247}$$

But note that

$$\mathbf{Y} = \begin{cases} \mathbf{Y} & \text{for } n \geq N-1 \\ \mathbf{Y}\mathbf{A}^{\dagger}\mathbf{A} & \text{for } n < N-1. \end{cases} \tag{248}$$

Ensemble Kalman Filter (EnKF)

Uses ensemble integrations to represent and predict uncertainty

$$\mathbf{Y} = \mathbf{g}(\mathbf{Z}^{\mathrm{f}})\mathbf{\Pi} \tag{249}$$

Updates the ensemble with measurements using

$$\mathbf{Z}^{\mathrm{a}} = \mathbf{Z}^{\mathrm{f}} + \mathbf{A}\mathbf{Y}^{\mathrm{T}}\left(\mathbf{Y}\mathbf{Y}^{\mathrm{T}} + \mathbf{E}\mathbf{E}^{\mathrm{T}}\right)^{-1}\left(\mathbf{D} - \mathbf{g}(\mathbf{Z}^{\mathrm{f}})\right) \tag{250}$$

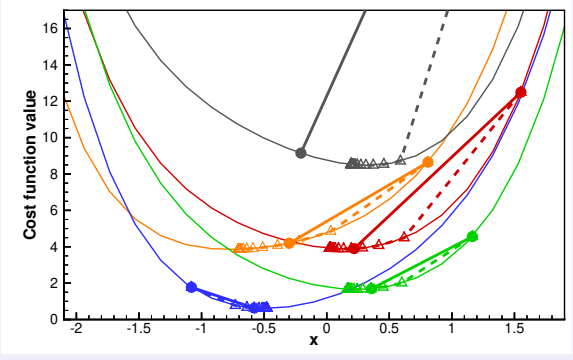
EnRML summary

- Represents errors statistics by ensembles of model and measurement realizations.
- Solves the DA problem over an assimilation window.
- Suitable for weather-prediction systems and parameter estimation.
- Nonlinear error propagation.
- Computes update by minimizing an ensemble of costfunctions
- Computes update in ensemble subspace.
- Can handle stronger nonlinearities than EnKF.

EnKF summary

- Represents errors statistics by ensembles of model and measurement realizations .
- Sequential data assimilation processing measurements recursively in time.
- Suitable for weather-prediction systems.
- Nonlinear error propagation.
- Linear update equation.
- Computes update in ensemble subspace.

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 ES steps with "inflated" observation errors.

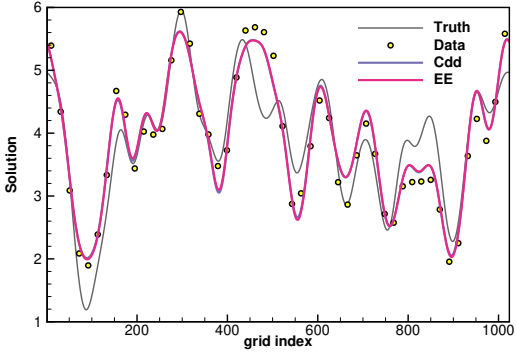
- Small updates reduce impact of the linear approximation.
- ESMDA is identical to ES in the linear case.
- Remember to resample measurement perturbations for each update step.

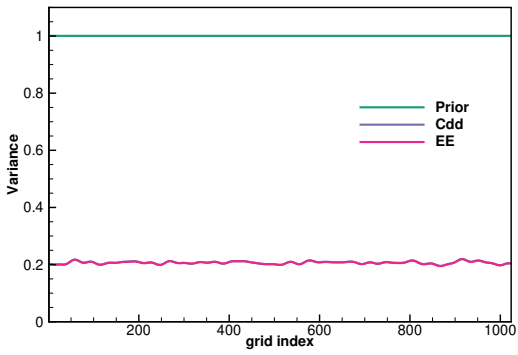
Some publications:

- 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).

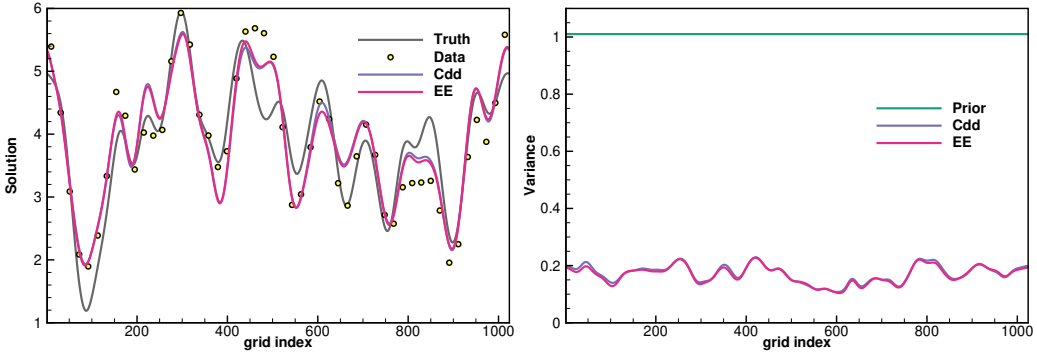
Linear EnKF update examples

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

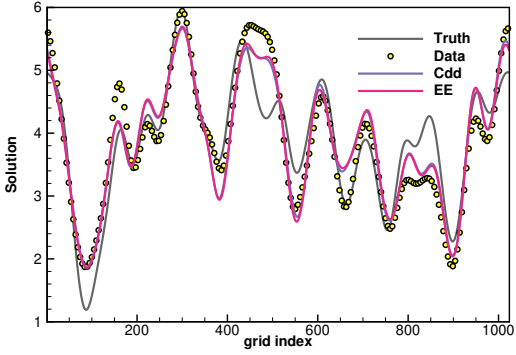


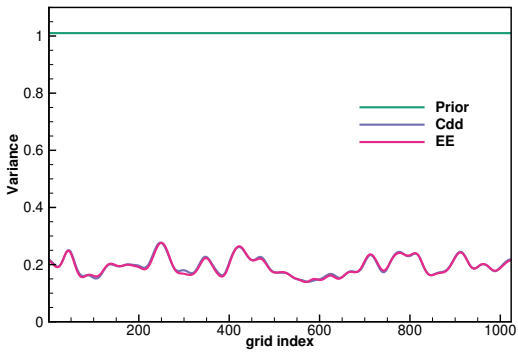


EnKF update with normal ensemble size $N = 100$ and $m = 50$



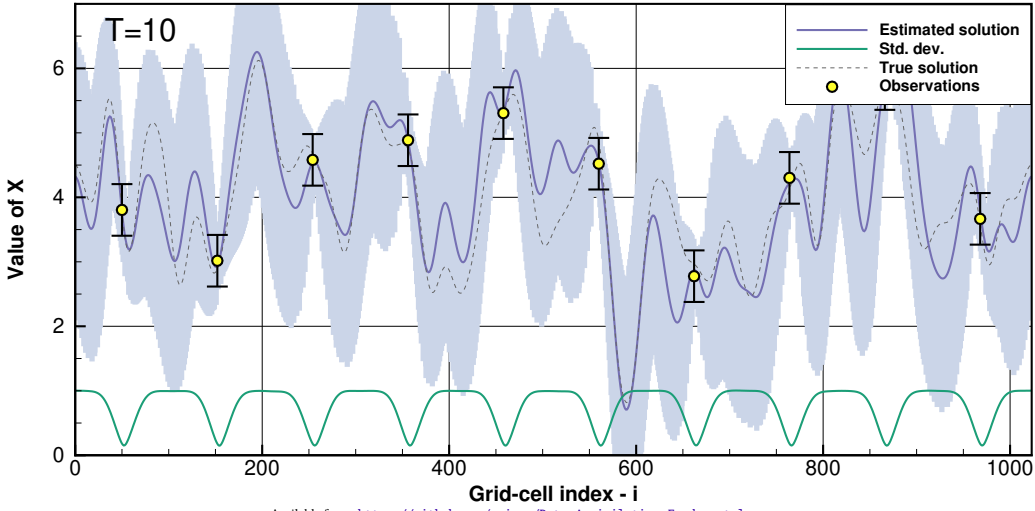
EnKF update with $N = 100$ and many measurements $m = 200$





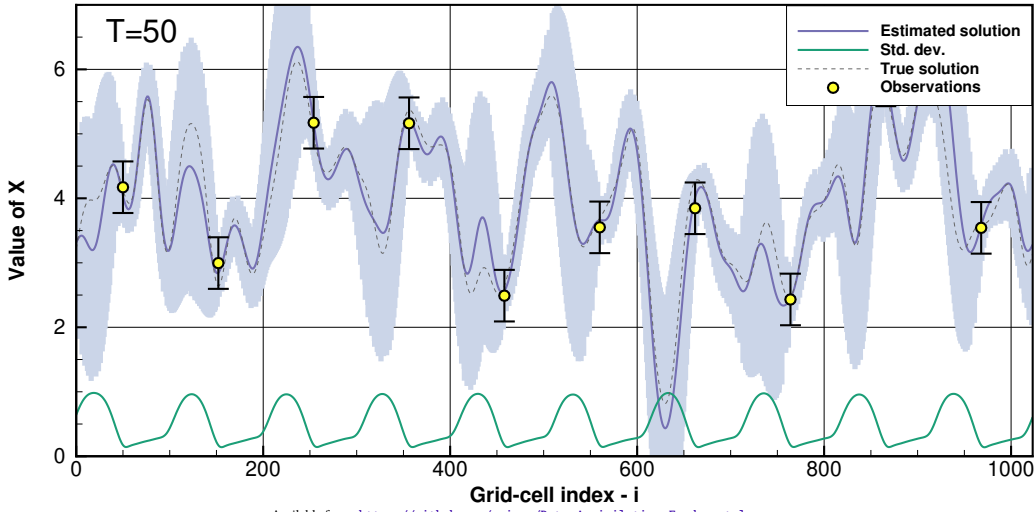
EnKF for an advection equation

EnKF with the advection equation after two updates

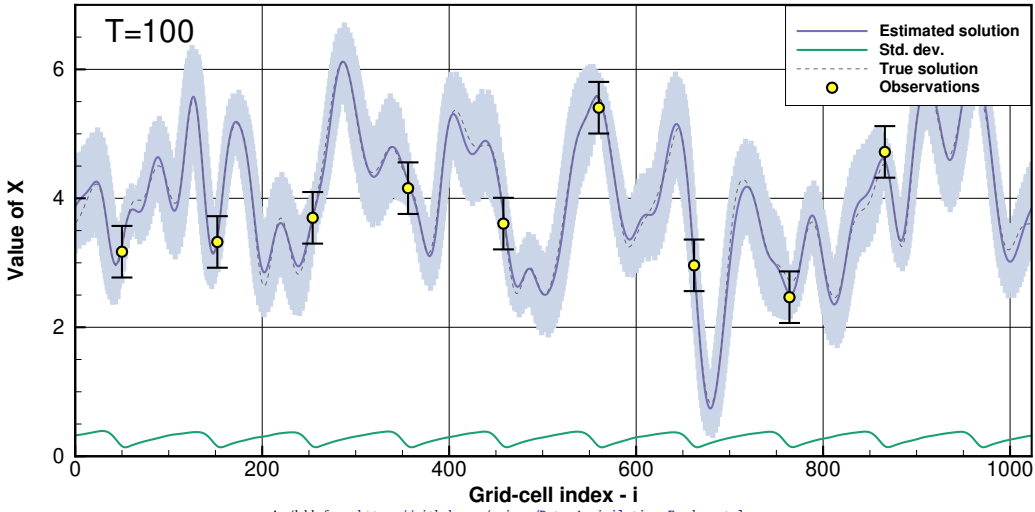


Available from: https://github.com/geirev/Data-Assimilation-Fundamentals

EnKF with the advection equation after ten updates



EnKF with the advection equation after twenty updates



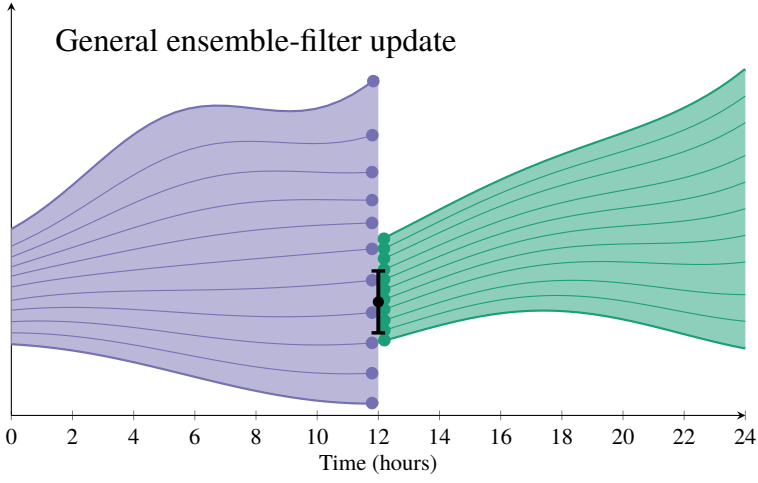
Available from: https://github.com/geirev/Data-Assimilation-Fundamentals

EnKF with the advection equation: Animation

Animation

EnKF with the Lorenz equations

General filter formulation



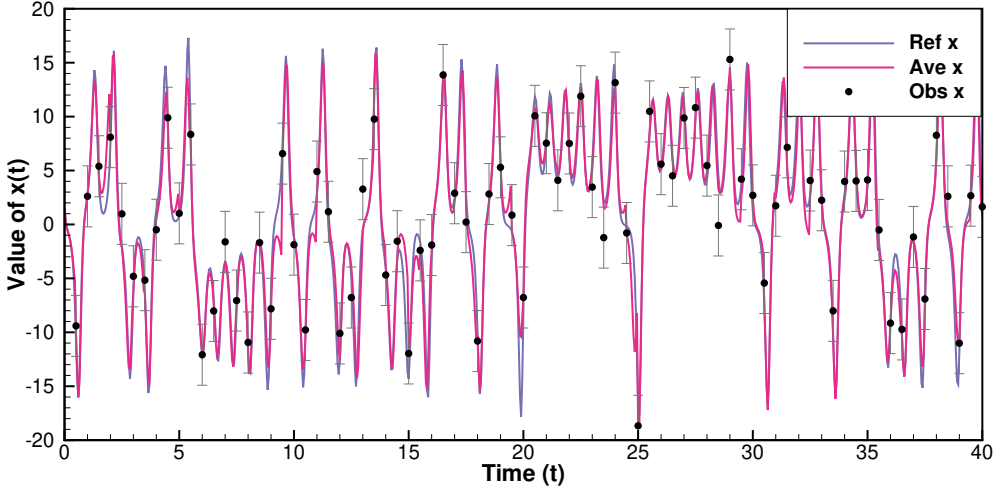
EnKF with the Lorenz model

$$\frac{\partial x}{\partial t} = \sigma(y - x), \tag{251}$$

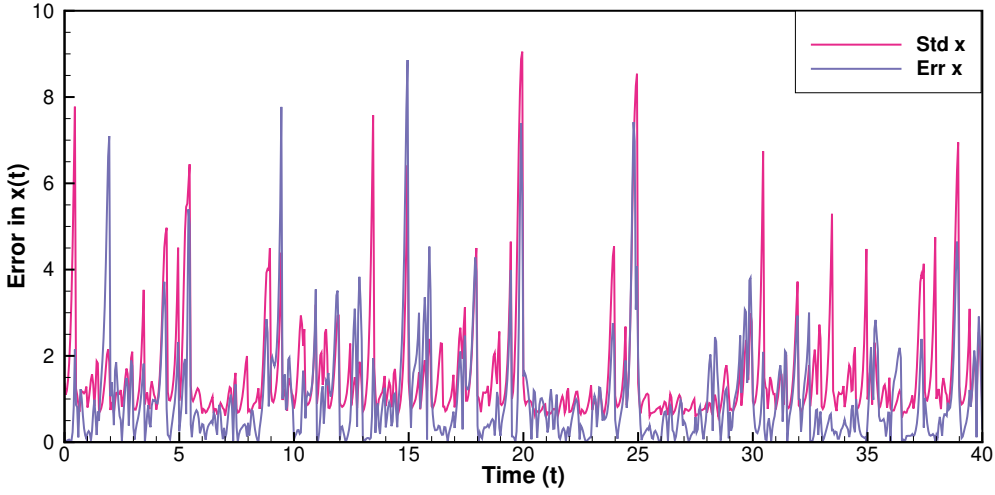
$$\frac{\partial y}{\partial t} = \rho x - y - xz, \tag{252}$$

$$\frac{\partial z}{\partial t} = xy - \beta z. \tag{253}$$

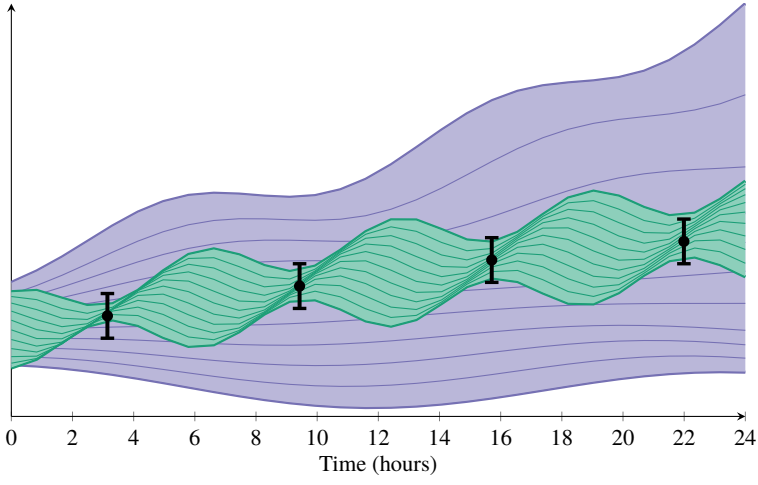
EnKF with the Lorenz model: estimate



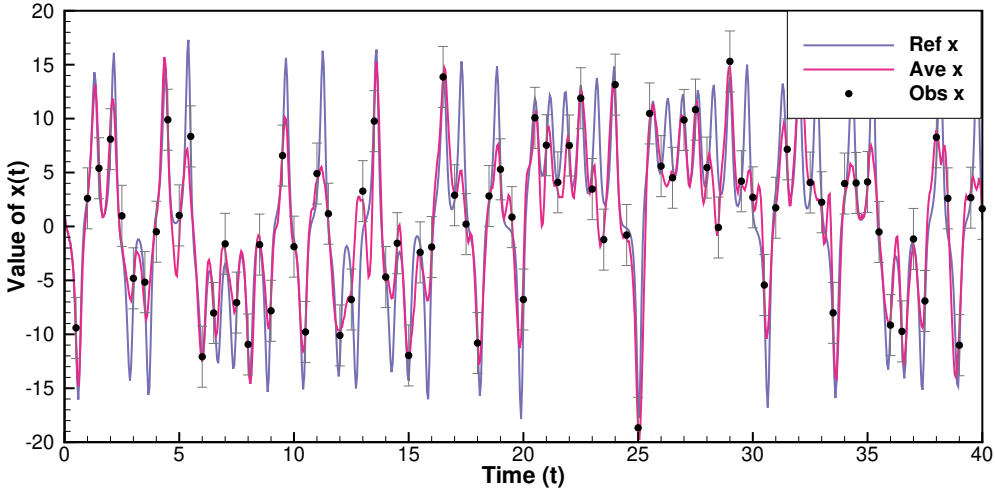
EnKF with the Lorenz model: error estimate



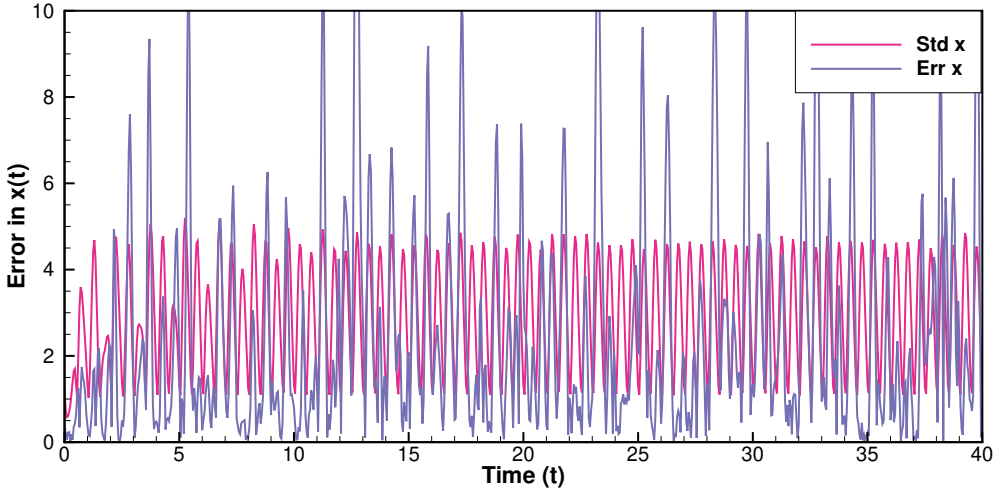
General smoother formulation



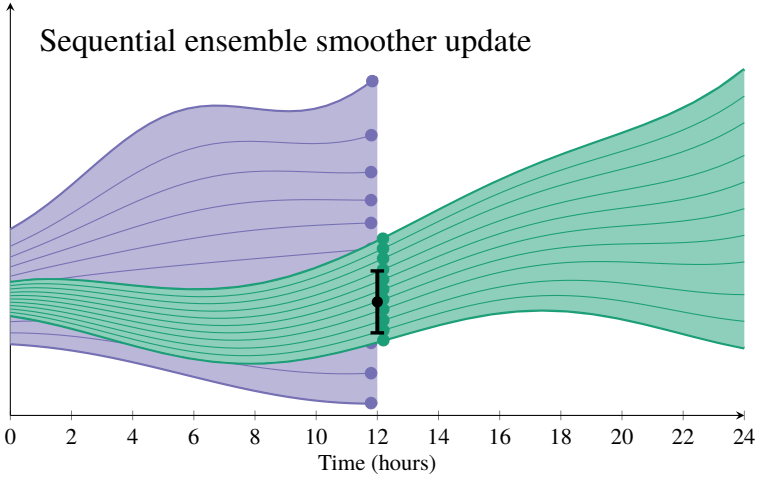
ES with the Lorenz model: estimate



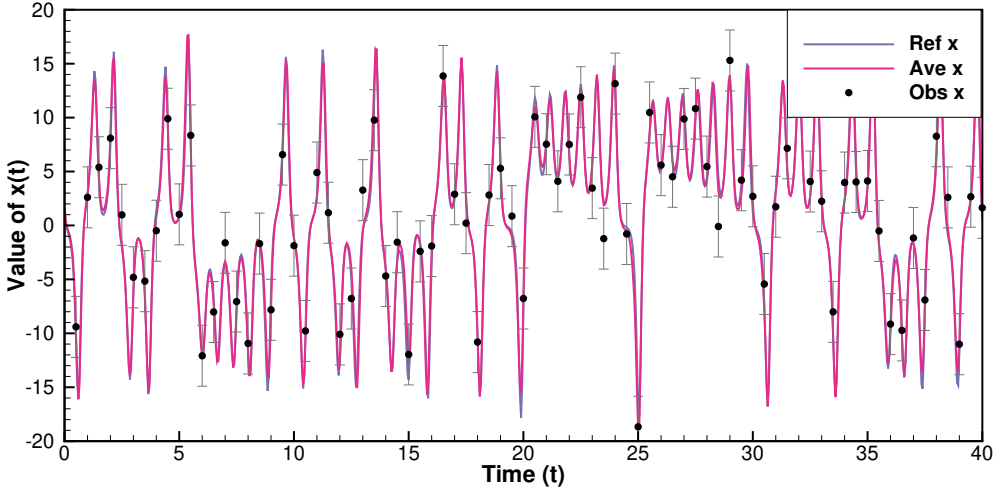
ES with the Lorenz model: error estimate



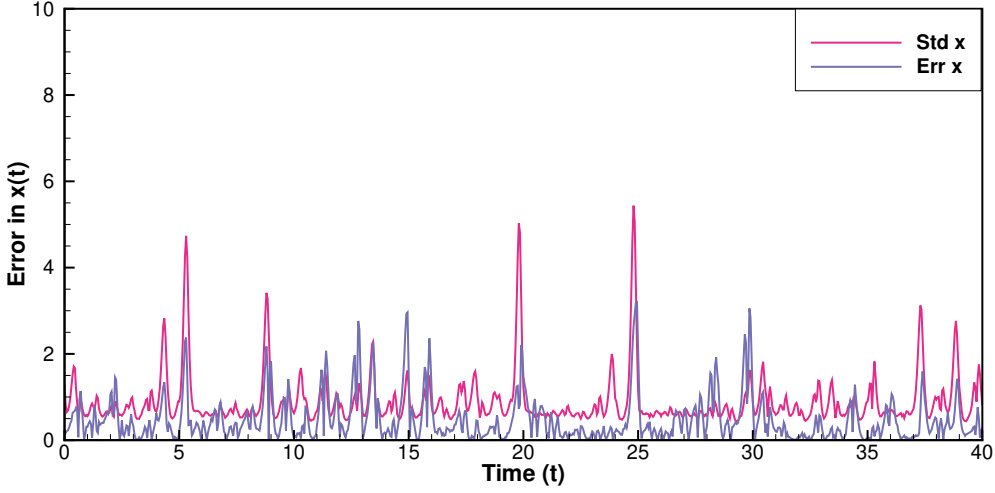
Recursive smoother formulation



EnKS with the Lorenz model: estimate



EnKS with the Lorenz model: error estimate



EnKF algorithms

Subspace EnKF update

1: subroutine EnKF_update$(\mathbf{Z}, \mathbf{D}, \boldsymbol{\Upsilon})$
2: Input: $\mathbf{Z} \in \mathfrak{R}^{n \times N}$ ▷ Prior state-vector ensemble
3: Input: $\mathbf{D} \in \mathfrak{R}^{m \times N}$ ▷ Perturbed measurements
4: Input: $\boldsymbol{\Upsilon} \in \mathfrak{R}^{m \times N}$ ▷ Predicted measurements
5: $\boldsymbol{\Pi} = \left(\mathbf{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^{\mathrm{T}}\right) / \sqrt{N-1}$ ▷ $\boldsymbol{\Pi} \in \mathfrak{R}^{N \times N}$
6: $\mathbf{E} = \mathbf{D}\boldsymbol{\Pi}$ ▷ $\mathbf{E} \in \mathfrak{R}^{m \times N}$
7: $\mathbf{Y} = \boldsymbol{\Upsilon}\boldsymbol{\Pi}$ ▷ $\mathbf{Y} \in \mathfrak{R}^{m \times N}$
8: **if** $n < N - 1$ **then**
9: $\quad \mathbf{Y} = \mathbf{Y}\mathbf{A}^{\dagger}\mathbf{A}$
10: **end if**
11: $\mathbf{S} = \mathbf{Y}$ ▷ $\mathbf{S} \in \mathfrak{R}^{m \times N}$
12: $\mathbf{W} = \mathbf{S}^{\mathrm{T}}\left(\mathbf{S}\mathbf{S}^{\mathrm{T}} + \mathbf{E}\mathbf{E}^{\mathrm{T}}\right)^{-1}\left(\mathbf{D} - \boldsymbol{\Upsilon}\right)$ ▷ $\mathbf{W} \in \mathfrak{R}^{N \times N}$
13: $\mathbf{Z} \leftarrow \mathbf{Z}\left(\mathbf{I} + \mathbf{W}\middle/\sqrt{N-1}\right)$ ▷ Update returned in $\mathbf{Z}$

Standard EnKF application

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

EnKF updating $\mathbf{X}_0$

1: Input: $\mathbf{Z} \in \Re^{n \times N}$ ▷ Initial model state-vector ensemble
2: Input: $\mathbf{D}_l \in \Re^{m \times N}$ ▷ Perturbed measurements for each assimilation window
3: **for** $l = 1, \ldots$ **do** ▷ Loop over assimilation windows
4: $\quad \mathbf{X}_0 = \mathbf{Z}$
5: $\quad$ **for** $k = 1, K$ **do** ▷ Ensemble integration
6: $\quad\quad \mathbf{X}_k = \mathbf{m}(\mathbf{X}_{k-1})$
7: $\quad$ **end for**
8: $\quad \mathbf{X} = [\mathbf{X}_0, \ldots, \mathbf{X}_K]$
9: $\quad \Upsilon = \mathbf{h}(\mathbf{X})$ ▷ Predicted measurments
10: $\quad$ **call EnKF_update**$(\mathbf{X}_0, \mathbf{D}_l, \Upsilon)$
11: $\quad$ **for** $k = 1, K$ **do** ▷ Rerun ensemble integration
12: $\quad\quad \mathbf{X}_k = \mathbf{m}(\mathbf{X}_{k-1})$
13: $\quad$ **end for**
14: $\quad \mathbf{Z} = \mathbf{X}_K$ ▷ Define state vector for next assimilation window
15: **end for**

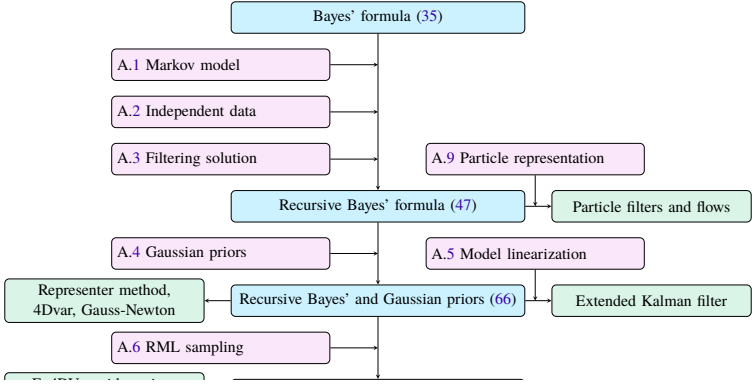
ES updating $\mathbf{X}$

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

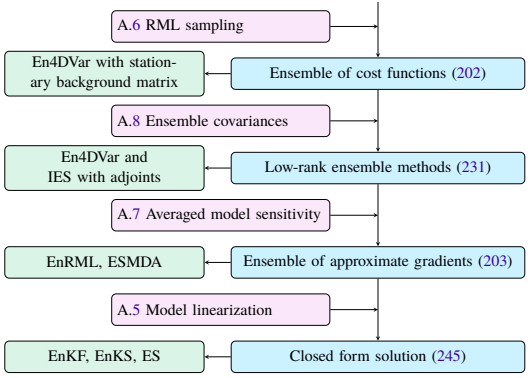
EnKS updating $[\mathbf{X}_l, \ldots, \mathbf{X}_0]$

1: Input: $\mathbf{Z} \in \Re^{n \times N}$ ▷ Initial model state-vector ensemble
2: Input: $\mathbf{D}_l \in \Re^{m \times N}$ ▷ Perturbed measurements for each assimilation window
3: **for** $l = 1, \ldots$ **do** ▷ Loop over assimilation windows
4: $\quad \mathbf{X}_0 = \mathbf{Z}$
5: $\quad$ **for** $k = 1, K$ **do** ▷ Ensemble integration
6: $\quad\quad \mathbf{X}_k = \mathbf{m}(\mathbf{X}_{k-1})$
7: $\quad$ **end for**
8: $\quad \mathbf{X} = [\mathbf{X}_0, \ldots, \mathbf{X}_K]$
9: $\quad \Upsilon = \mathbf{h}(\mathbf{X})$ ▷ Predicted measurments
10: $\quad \mathcal{X}_l = [\mathcal{X}_{l-1}, \mathbf{X}]$
11: $\quad$ **call EnKF_update**$(\mathcal{X}_l, \mathbf{D}_l, \Upsilon)$
12: $\quad \mathbf{X} = \mathcal{X}_l(l)$
13: $\quad \mathbf{Z} = \mathbf{X}_K$ ▷ Define state vector for next assimilation window
14: **end for**

Graphic overview



Graphic overview



Particle filters and flows

Particle filters and flows

Approximation 9 (Particle representation of the pdfs)

It is possible to approximate a probability density function by a finite ensemble of N model states (or particles) as

$$f(\mathbf{z}) \approx \sum_{j=1}^{N} \frac{1}{N} \delta(\mathbf{z} - \mathbf{z}_j), \tag{254}$$

where $\delta(\cdot)$ denotes the Dirac-delta function.

Comparison of methods on a scalar model

Scalar models

$$y = g(x, q) = x + 0.3x^3 + q, \tag{255}$$

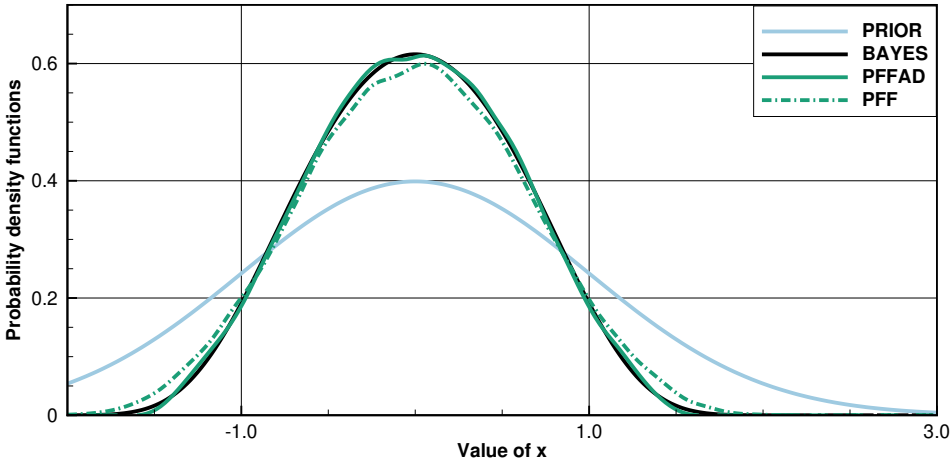
$$y = g(x, q) = 1 + \sin(x) + q, \tag{256}$$

Scalar examples

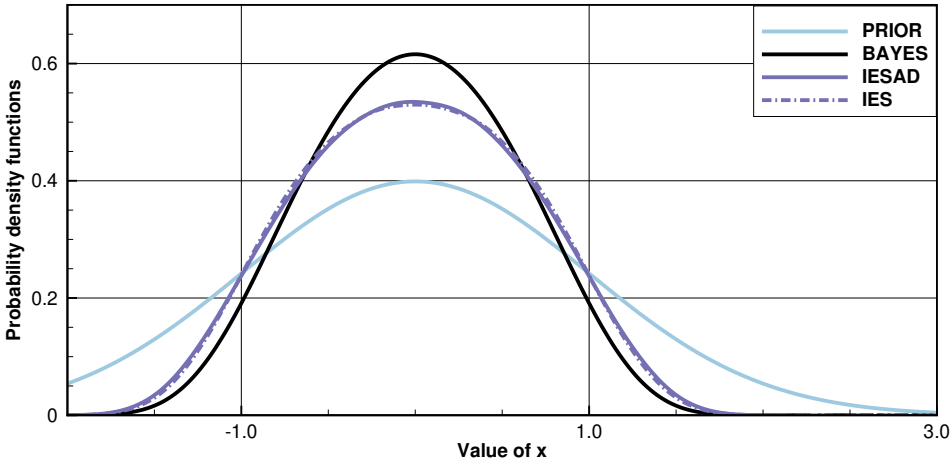
1. Model Eq. (255), $x_j^{\mathrm{f}} = \mathcal{N}(x^{\mathrm{f}} = 0.0, C_{xx} = 1.0)$, $d_j = \mathcal{N}(0.0, 1.0)$, and $q = \mathcal{N}(0, C_{qq} = 0.25)$.
2. Model Eq. (255), $x_j^{\mathrm{f}} = \mathcal{N}(x^{\mathrm{f}} = 1.0, C_{xx} = 1.0)$, $d_j = \mathcal{N}(0.0, -1.0)$, and $q = \mathcal{N}(0, C_{qq} = 0.25)$.
3. Model Eq. (256), $x_j^{\mathrm{f}} = \mathcal{N}(x^{\mathrm{f}} = 1.0, C_{xx} = 1.0)$, $d_j = \mathcal{N}(0.0, 1.0)$, and $q = \mathcal{N}(0, C_{qq} = 0.25)$.

Ensemble size $N = 10^7$.

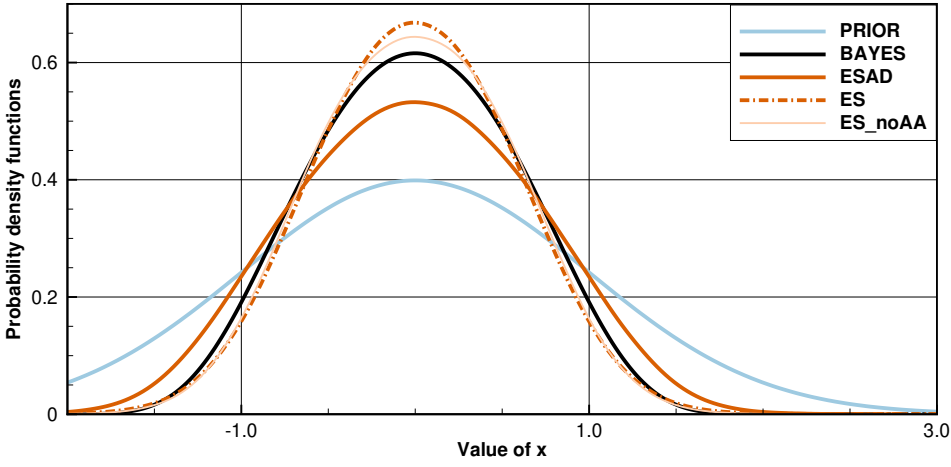
Case 1: Particle flow



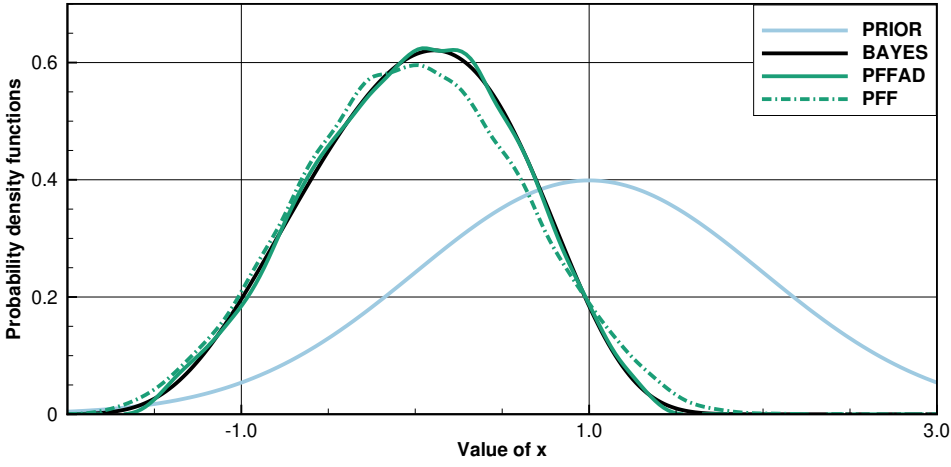
Case 1: IES


PRIOR
BAYES
IESAD
IES
Probability density functions
0.6
0.4
0.2
0
-1.0
1.0
3.0
Value of x

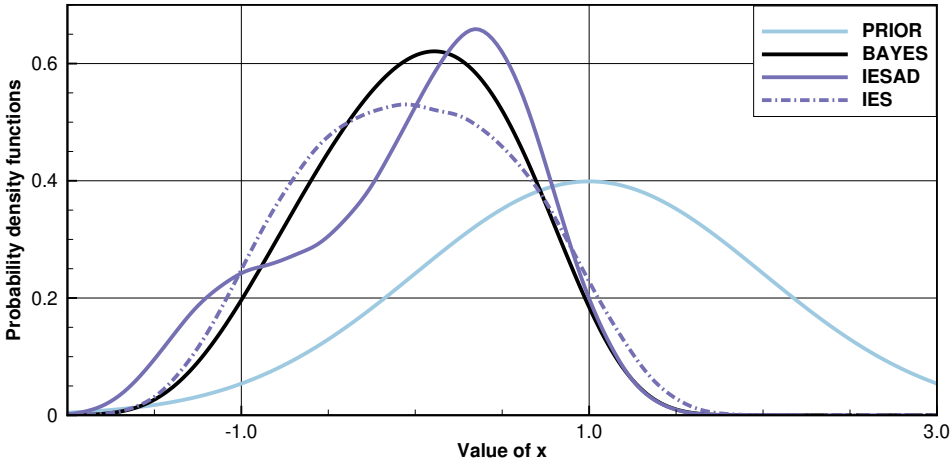
Case 1: EnKF


PRIOR
BAYES
ESAD
ES
ES_noAA
Probability density functions
0
0.2
0.4
0.6
-1.0
1.0
3.0
Value of x

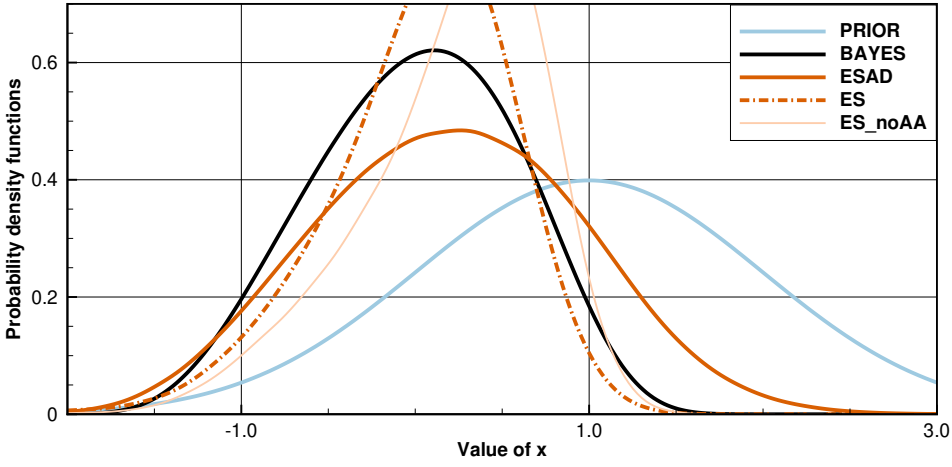
Case 2: Particle flow



Case 2: IES

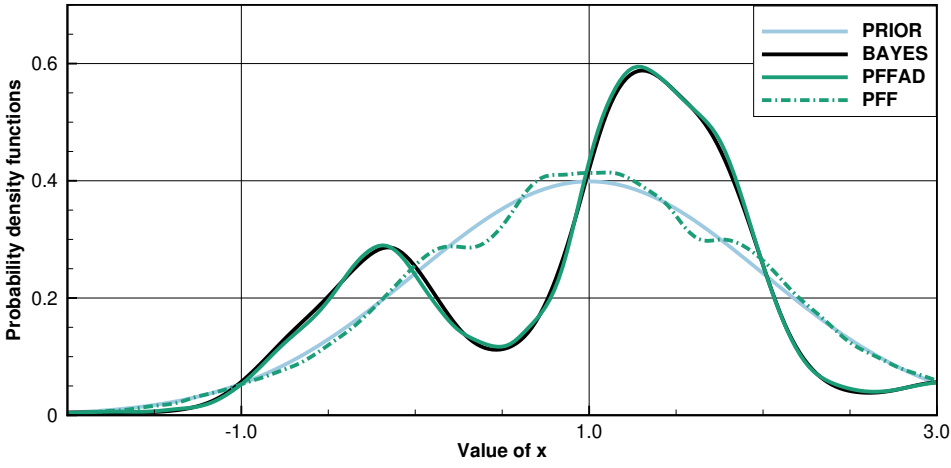

PRIOR
BAYES
IESAD
IES
Probability density functions
0
0.2
0.4
0.6
-1.0
1.0
3.0
Value of x

Case 2: EnKF

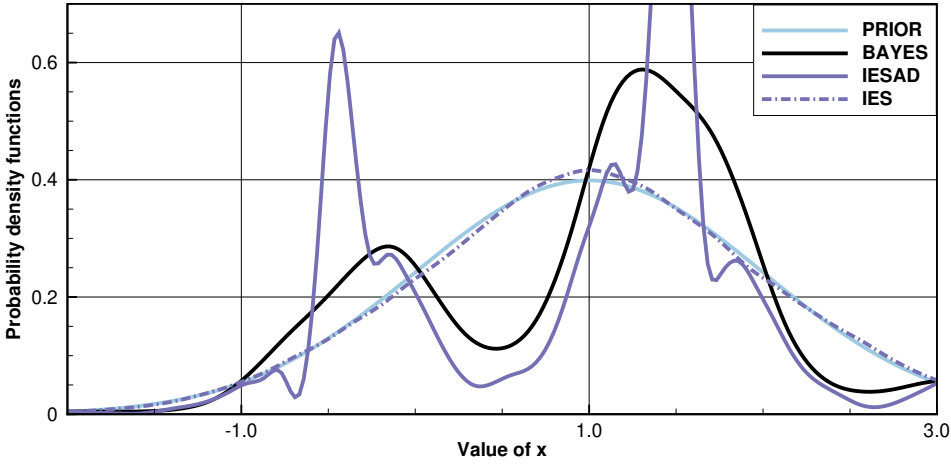


Available from: https://github.com/geirev/Data-Assimilation-Fundamentals

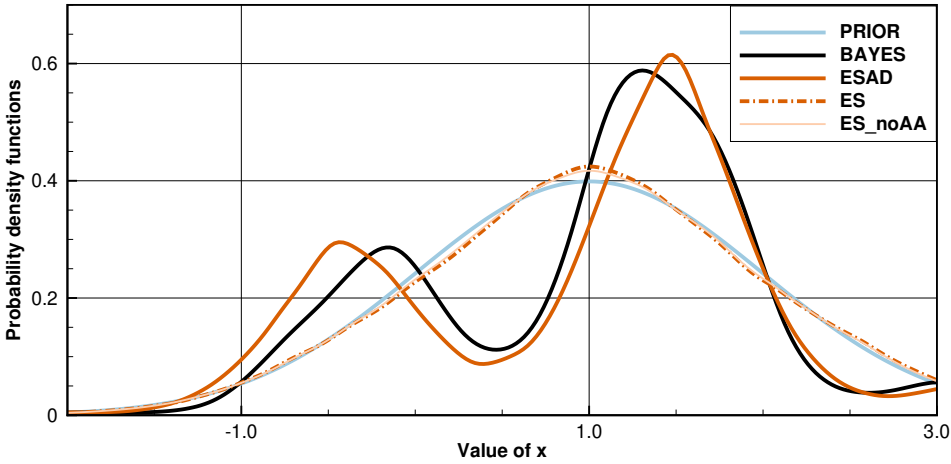
Case 3: Particle flow



Case 3: IES


PRIOR
BAYES
IESAD
IES
Probability density functions
0.6
0.4
0.2
0
-1.0
1.0
3.0
Value of x

Case 3: EnKF


PRIOR
BAYES
ESAD
ES
ES_noAA
Probability density functions
0.6
0.4
0.2
0
-1.0
1.0
3.0
Value of x

ESMDA with a SARS-COV-2 pandemic nmodel

Available observations

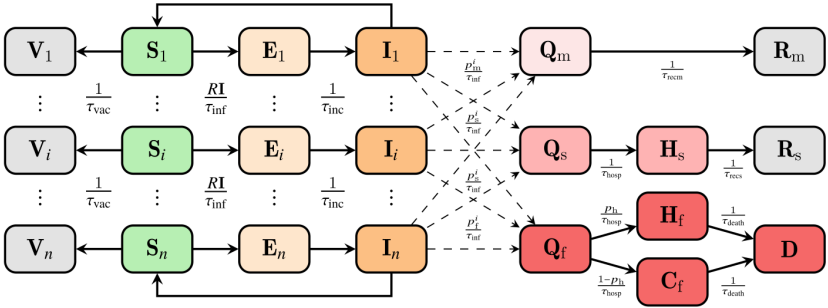
Hospitalized: Within different agegroups and gender.

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

$$\frac{\partial \mathbf{S}_i}{\partial t} = -\left(\sum_{j=1}^{n} \frac{R_{ij}(t)\mathbf{I}_j}{\tau_{\text{inf}}}\right)\mathbf{S}_i \tag{257}$$

$$\frac{\partial \mathbf{E}_i}{\partial t} = \left(\sum_{j=1}^{n} \frac{R_{ij}(t)\mathbf{I}_j}{\tau_{\text{inf}}}\right)\mathbf{S}_i - \frac{1}{\tau_{\text{inc}}}\mathbf{E}_i \tag{258}$$

$$\frac{\partial \mathbf{I}_i}{\partial t} = \frac{1}{\tau_{\text{inc}}}\mathbf{E}_i - \frac{1}{\tau_{\text{inf}}}\mathbf{I}_i \tag{259}$$

$$\frac{\partial \mathbf{Q}_{\text{m}}}{\partial t} = \sum_{i=1}^{n} \frac{p_{\text{m}}^i}{\tau_{\text{inf}}}\mathbf{I}_i - (1/\tau_{\text{recm}})\,\mathbf{Q}_{\text{m}} \tag{260}$$

$$\frac{\partial \mathbf{Q}_{\text{s}}}{\partial t} = \sum_{i=1}^{n} \frac{p_{\text{s}}^i}{\tau_{\text{inf}}}\mathbf{I}_i - (1/\tau_{\text{hosp}})\,\mathbf{Q}_{\text{s}} \tag{261}$$

$$\frac{\partial \mathbf{Q}_{\text{f}}}{\partial t} = \sum_{i=1}^{n} \frac{p_{\text{f}}^i}{\tau_{\text{inf}}}\mathbf{I}_i - (1/\tau_{\text{hosp}})\,\mathbf{Q}_{\text{f}} \tag{262}$$

$$\frac{\partial \mathbf{H}_{\text{s}}}{\partial t} = \frac{1}{\tau_{\text{hosp}}}\mathbf{Q}_{\text{s}} - \frac{1}{\tau_{\text{recs}}}\mathbf{H}_{\text{s}} \tag{263}$$

$$\frac{\partial \mathbf{H}_{\text{f}}}{\partial t} = \frac{p_{\text{h}}}{\tau_{\text{hosp}}}\mathbf{Q}_{\text{f}} - \frac{1}{\tau_{\text{death}}}\mathbf{H}_{\text{f}} \tag{264}$$

$$\frac{\partial \mathbf{C}_{\text{f}}}{\partial t} = \frac{(1-p_{\text{h}})}{\tau_{\text{hosp}}}\mathbf{Q}_{\text{f}} - \frac{1}{\tau_{\text{death}}}\mathbf{C}_{\text{f}} \tag{265}$$

$$\frac{\partial \mathbf{R}_{\text{m}}}{\partial t} = \frac{1}{\tau_{\text{recm}}}\mathbf{Q}_{\text{m}} \tag{266}$$

$$\frac{\partial \mathbf{R}_{\text{s}}}{\partial t} = \frac{1}{\tau_{\text{recs}}}\mathbf{H}_{\text{s}} \tag{267}$$

$$\frac{\partial \mathbf{D}}{\partial t} = \frac{1}{\tau_{\text{death}}}\mathbf{H}_{\text{f}} + \frac{1}{\tau_{\text{death}}}\mathbf{C}_{\text{f}} \tag{268}$$

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^i_{m}, p^i_{s}, p^i_{f} 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

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

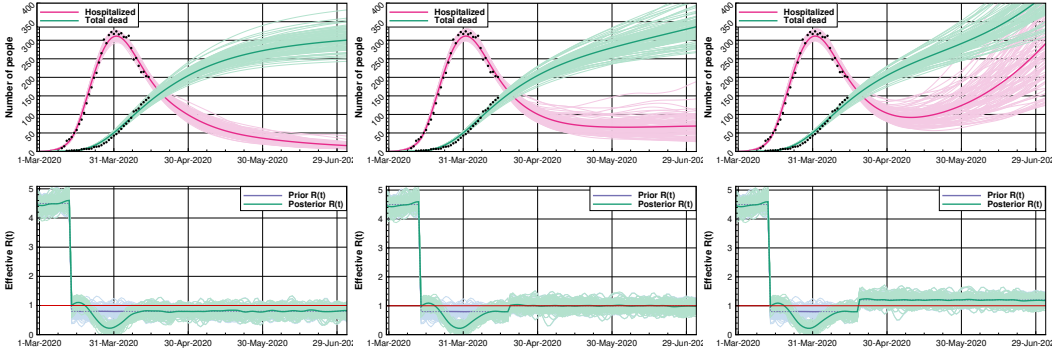
$\mathbf{R}(t)$ is a function of time (steered by how people isolate or interact).

- $R(t)$ 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.

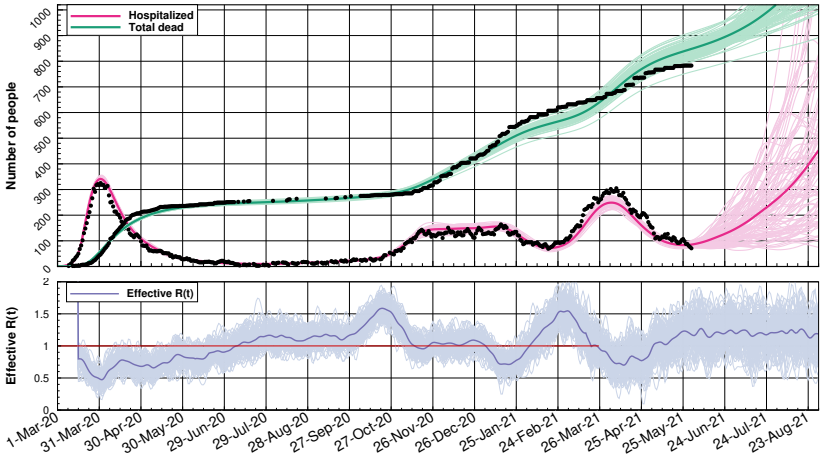
We used ESMDA

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

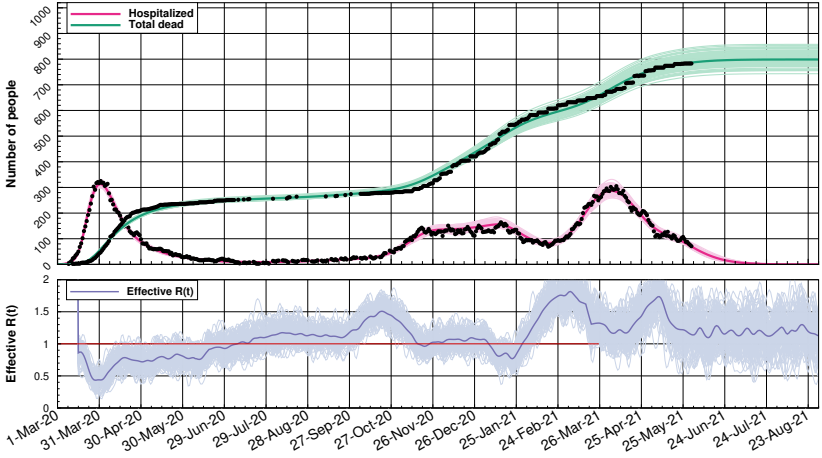
Back-to-school scenarios for Norway



Norway: prediction without vaccinations



Norway: prediction with vaccinations



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

Summary EnKF_seir

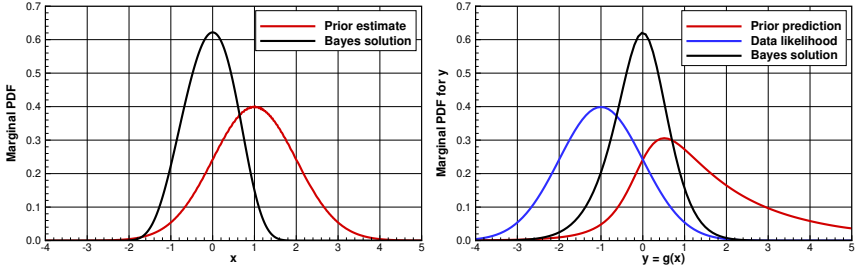
- The DA system tracks the epidemic accurately.
- We can estimate past $R(t)$.
- It is possible to quantify the impact of interventions.
- Short-term forecasting using R-persistence works well.
- Long-term scenario forecasting with specified future R.
- Code: https://github.com/geirev/EnKF_seir
- Paper: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

EnRML for History Matching Petroleum Models

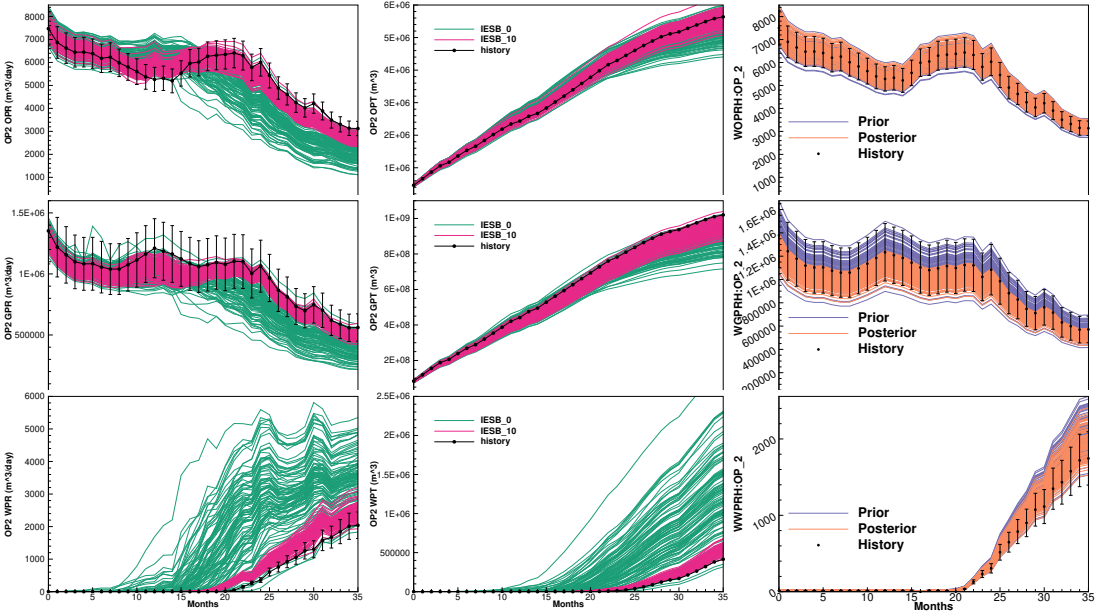
Illustration of the parameter estimation problem

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

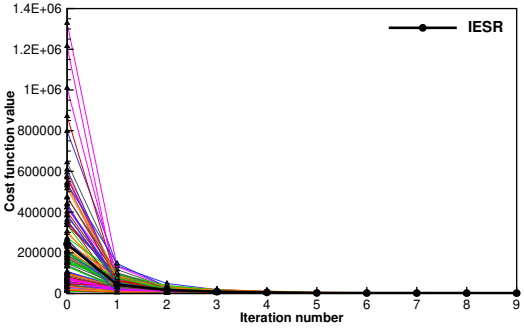
$$f(\mathbf{x}|\mathbf{d}) \propto f(\mathbf{d} \,|\, \mathbf{g}(\mathbf{x}))\, f(\mathbf{x})$$

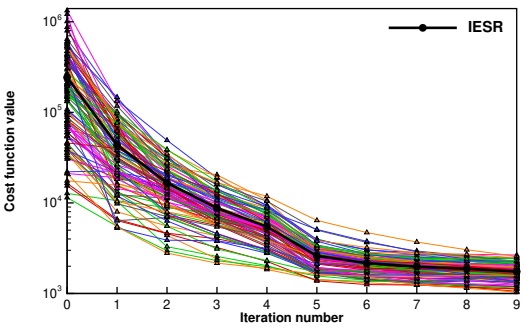


- $\mathbf{x}$ represents model input parameters like, porosity, permeability, fault multipliers
- $\mathbf{y}$ could represent predicted production of oil, gas, and water
- Prior pdf represents uncertainty of $\mathbf{x}$.
- Prior prediction pdf represents uncertainty of $\mathbf{y} = g(\mathbf{x})$.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


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

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.
- Next seminar will present iterative ensemble smoothers and their applications.
- 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.