Data-Assimilation Fundamentals:

A Unified Formulation of the State and Parameter Estimation Problem

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

NORCE

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

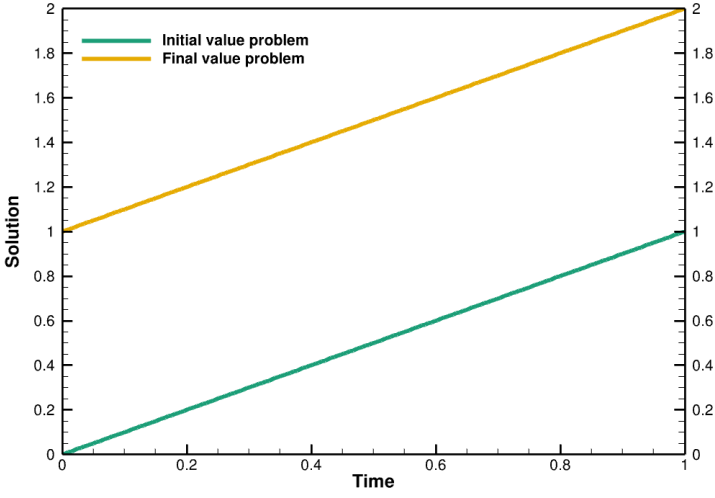
$$\frac{dx}{dt} = 1 \qquad \text{Linear model} \qquad (1)$$

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

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

- 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

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

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

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

or since δx is arbitrary....

Euler-Lagrange equation

The Euler–Lagrange equation

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

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

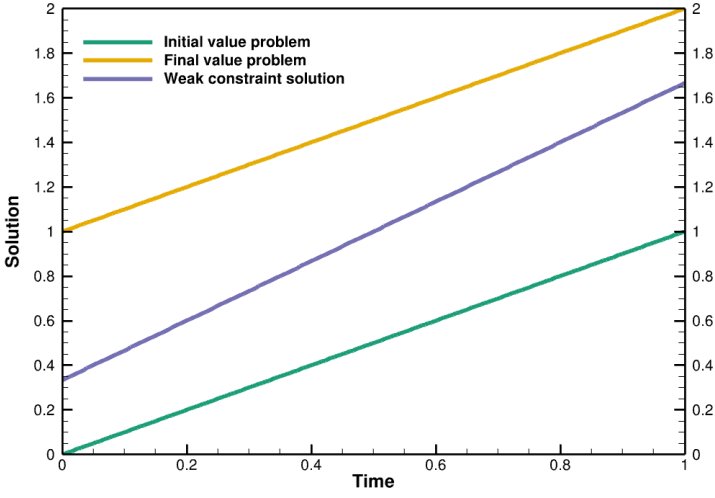
$$\frac{dx}{dt} + x = 3 \quad \text{for } t = 1, \tag{20}$$

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

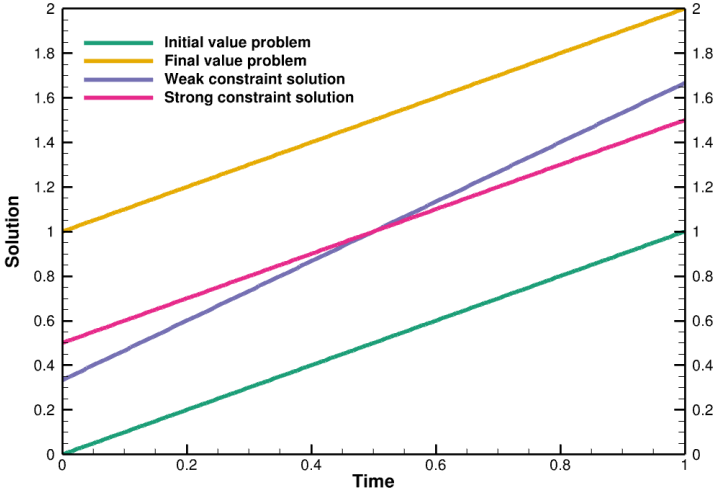
$$x = c_1 t + c_2, \tag{21}$$

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.

We start from Bayes formula

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.

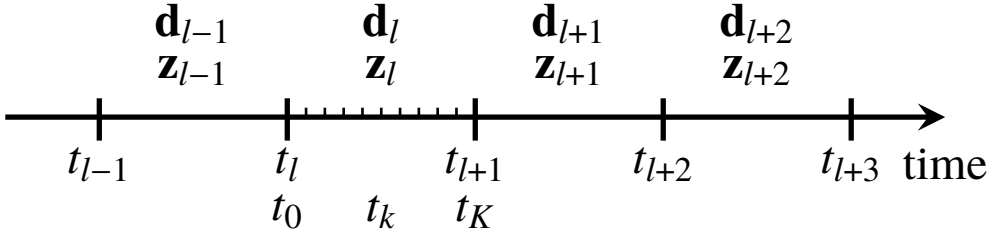
One of the strengths of Bayes' formula is that it does not try to solve the ill-defined problem of "inverting observations" but instead updates prior knowledge.

We start from Bayes' theorem

$$f(\mathcal{Z}|\mathcal{D}) = \frac{f(\mathcal{D}|\mathcal{Z})f(\mathcal{Z})}{f(\mathcal{D})}. \tag{29}$$

- $\mathcal{Z} = (\mathbf{z}_0, \mathbf{z}_1, \ldots, \mathbf{z}_L)$ 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.

$$f(\mathbf{z}_l|\mathbf{z}_{l-1}, \mathbf{z}_{l-2}, \ldots, \mathbf{z}_0) = f(\mathbf{z}_l|\mathbf{z}_{l-1}), \tag{30}$$

Independent measurements

Approximation 2 (Independent measurements)

We assume that measurements are independent between different assimilation windows.

Independent measurements have uncorrelated errors

$$f(\mathcal{D}|\mathcal{Z}) = \prod_{l=1}^{L} f(\mathbf{d}_l|\mathbf{z}_l). \tag{31}$$

Recursive form of Bayes

$$f(\mathbf{z}_1, \mathbf{z}_0|\mathbf{d}_1) = \frac{f(\mathbf{d}_1|\mathbf{z}_1) f(\mathbf{z}_1|\mathbf{z}_0) f(\mathbf{z}_0)}{f(\mathbf{d}_1)}, \tag{32}$$

$$f(\mathbf{z}_2, \mathbf{z}_1, \mathbf{z}_0|\mathbf{d}_1, \mathbf{d}_2) = \frac{f(\mathbf{d}_2|\mathbf{z}_2) f(\mathbf{z}_2|\mathbf{z}_1) f(\mathbf{z}_1, \mathbf{z}_0|\mathbf{d}_1)}{f(\mathbf{d}_2)}, \tag{33}$$

$$\vdots \tag{34}$$

$$f(\mathcal{Z}|\mathcal{D}) = \frac{f(\mathbf{d}_L|\mathbf{z}_L) f(\mathbf{z}_L|\mathbf{z}_{L-1}) f(\mathbf{z}_{L-1}, \ldots, \mathbf{z}_0|\mathbf{d}_{L-1}, \ldots, \mathbf{d}_1)}{f(\mathbf{d}_L)}. \tag{35}$$

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

$$f(\mathbf{z}_1|\mathbf{d}_1) = \frac{f(\mathbf{d}_1|\mathbf{z}_1)\ \int f(\mathbf{z}_1|\mathbf{z}_0) f(\mathbf{z}_0)\, d\mathbf{z}_0}{f(\mathbf{d}_1)} = \frac{f(\mathbf{d}_1|\mathbf{z}_1) f(\mathbf{z}_1)}{f(\mathbf{d}_1)}, \tag{36}$$

$$f(\mathbf{z}_2|\mathbf{d}_1, \mathbf{d}_2) = \frac{f(\mathbf{d}_2|\mathbf{z}_2)\ \int f(\mathbf{z}_2|\mathbf{z}_1) f(\mathbf{z}_1|\mathbf{d}_1)\, d\mathbf{z}_1}{f(\mathbf{d}_2)} = \frac{f(\mathbf{d}_2|\mathbf{z}_2) f(\mathbf{z}_2|\mathbf{d}_1)}{f(\mathbf{d}_2)}, \tag{37}$$

$$\vdots$$

$$f(\mathbf{z}_L|\mathcal{D}) = \frac{f(\mathbf{d}_L|\mathbf{z}_L)\ \int f(\mathbf{z}_L|\mathbf{z}_{L-1}) f(\mathbf{z}_{L-1}|\mathbf{d}_{L-1}, \ldots, \mathbf{d}_1)\, d\mathbf{z}_{L-1}}{f(\mathbf{d}_L)} \tag{38}$$

$$= \frac{f(\mathbf{d}_L|\mathbf{z}_L) f(\mathbf{z}_L|\mathbf{d}_{L-1})}{f(\mathbf{d}_L)}. \tag{39}$$

Or just Bayes' for the assimilation window

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

Discrete model with uncertain inputs

$$\mathbf{x}_k = \mathbf{m}(\mathbf{x}_{k-1}, \boldsymbol{\theta}, \mathbf{u}_k, \mathbf{q}_k). \tag{41}$$

- $\mathbf{x}_k$ 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.

General smoother formulation

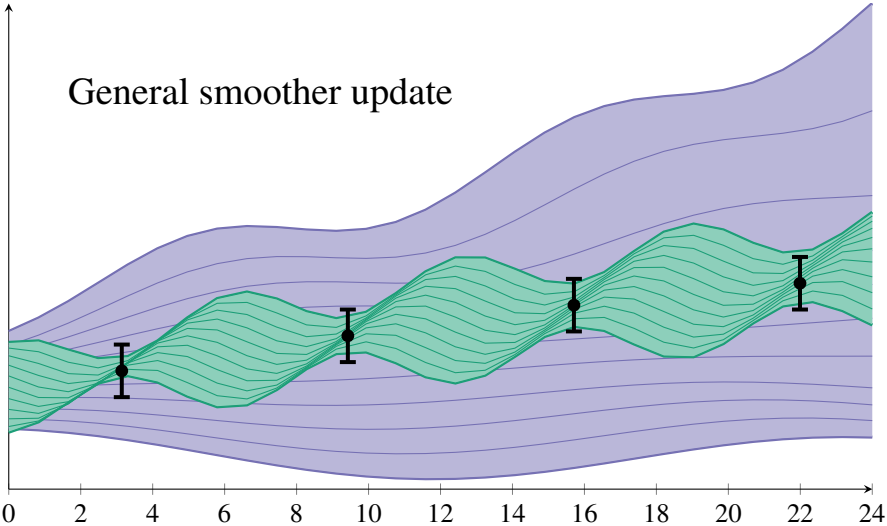
- Solve for model solution over an assimilation window $\mathbf{x}$.
- Condition on measurements distributed over the assimilation window.

Predicted measurements $\mathbf{y}$

$$\mathbf{y} = \mathbf{g}(\mathbf{z}) = \mathbf{h}(\mathbf{x}) = \mathbf{h}\big(\mathbf{m}(\mathbf{x}_0, \mathbf{q})\big) \tag{42}$$

- 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

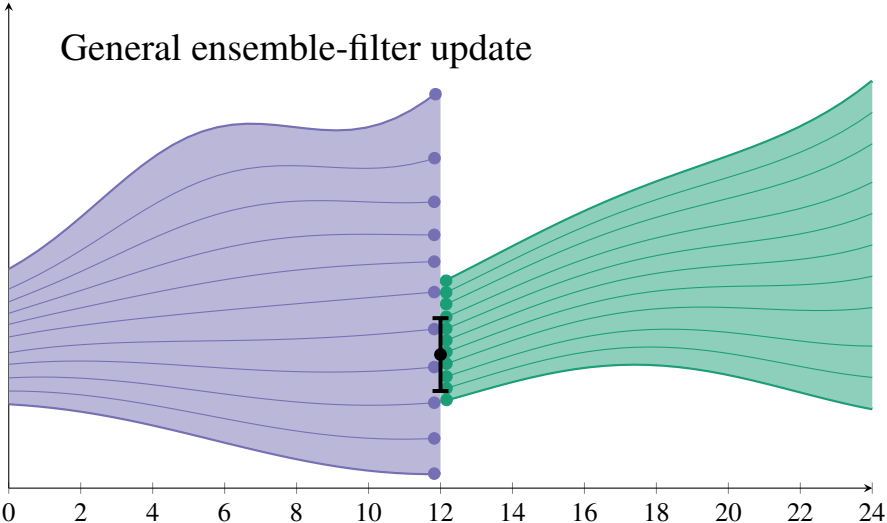
General filter formulation

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

$$\mathbf{y} = \mathbf{g}(\mathbf{z}) = \mathbf{h}(\mathbf{x}_K). \tag{43}$$

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

General filter formulation



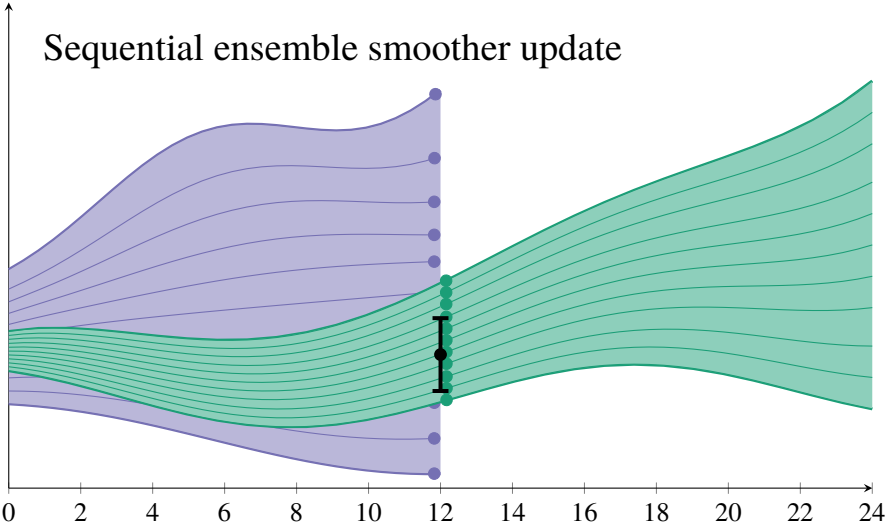
Recursive smoother formulation

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

$$\mathbf{y} = \mathbf{g}(\mathbf{z}) = \mathbf{h}(\mathbf{x}_K), \tag{44}$$

- Ensemble Kalman Smoother (EnKS)

Recursive smoother formulation



Iterative methods to handle nonlinearity

EKF (EnKF) uses linearization to compute one linear update step.

- Incremental SC-4DVar (using adjoint model).
- Incremental WC-4DVar (representer solution using adjoint model).
- Ensemble of incremental SC-4DVar (using adjoint model).
- Ensemble of Incremental WC-4DVar (representer solution using adjoint model).
- EnRML (using averaged model sensitivity).
- ESMDA (using averaged model sensitivity).

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.

$$\mathbf{y} = \mathbf{g}(\mathbf{z}) = \mathbf{h}\big(\mathbf{m}(\mathbf{x}_0)\big). \tag{45}$$

- 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 $\mathbf{x}_K$.

$$\mathbf{y} = \mathbf{g}(\mathbf{z}) = \mathbf{h}\big(\mathbf{x}_K\big). \tag{46}$$

- EnKF.

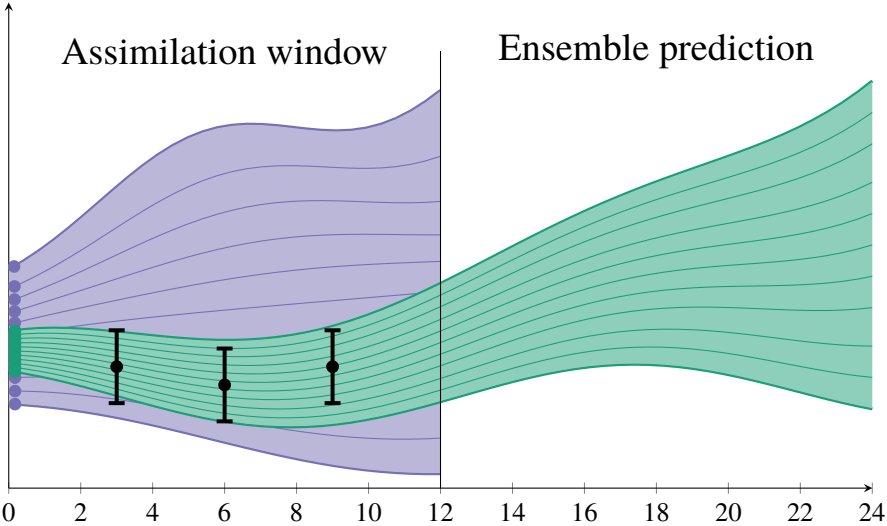
Parameter estimation

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

$$\mathbf{y} = \mathbf{g}(\mathbf{z}) = \mathbf{h}\big(\mathbf{m}(\boldsymbol{\theta})\big). \tag{47}$$

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

Smoother for perfect models



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

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

$$f(\mathbf{z}|\mathbf{d}) = \frac{f\big(\mathbf{d}|\mathbf{g}(\mathbf{z})\big)f(\mathbf{z})}{f(\mathbf{d})}. \tag{48}$$

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.*

$$f(\mathbf{z}|\mathbf{d}) \propto \exp\{-\mathcal{J}(\mathbf{z})\}, \tag{49}$$

Leads to a cost-function formulation for the MAP solution

Cost function

$$\mathcal{J}(\mathbf{z}) = \frac{1}{2}\left(\mathbf{z}-\mathbf{z}^{\mathrm{f}}\right)^{\mathrm{T}}\mathbf{C}_{zz}^{-1}\left(\mathbf{z}-\mathbf{z}^{\mathrm{f}}\right) + \frac{1}{2}\left(\mathbf{g}(\mathbf{z})-\mathbf{d}\right)^{\mathrm{T}}\mathbf{C}_{dd}^{-1}\left(\mathbf{g}(\mathbf{z})-\mathbf{d}\right). \tag{50}$$

The gradient set to zero

$$\mathbf{C}_{zz}^{-1}\left(\mathbf{z}^{\mathrm{a}}-\mathbf{z}^{\mathrm{f}}\right) + \nabla_{\mathbf{z}}\mathbf{g}\left(\mathbf{z}^{\mathrm{a}}\right)\mathbf{C}_{dd}^{-1}\left(\mathbf{g}\left(\mathbf{z}^{\mathrm{a}}\right)-\mathbf{d}\right) = 0. \tag{51}$$

- There is no explicit solution of the gradient equation.

Gauss-Newton methods solves for the MAP estimate

Gauss-Newton iteration

$$\mathbf{z}^{i+1} = \mathbf{z}^{i} - \gamma^{i}\left(\mathbf{C}_{zz}^{-1} + \mathbf{G}^{i\mathrm{T}}\mathbf{C}_{dd}^{-1}\mathbf{G}^{i}\right)^{-1}\left(\mathbf{C}_{zz}^{-1}(\mathbf{z}^{i} - \mathbf{z}^{\mathrm{f}}) + \mathbf{G}^{i\mathrm{T}}\mathbf{C}_{dd}^{-1}\left(\mathbf{g}(\mathbf{z}^{i}) - \mathbf{d}\right)\right). \tag{52}$$

- The incremental formulation is sometimes more convenient.

Incremental Gauss-Newton methods

Quadratic cost function for the increments

$$\mathcal{J}(\delta\mathbf{z}) = \frac{1}{2}\left(\delta\mathbf{z} - \boldsymbol{\xi}^i\right)^{\mathrm{T}} \mathbf{C}_{zz}^{-1} \left(\delta\mathbf{z} - \boldsymbol{\xi}^i\right) + \frac{1}{2}\left(\mathbf{G}^i\delta\mathbf{z} - \boldsymbol{\eta}^i\right)^{\mathrm{T}} \mathbf{C}_{dd}^{-1} \left(\mathbf{G}^i\delta\mathbf{z} - \boldsymbol{\eta}^i\right). \tag{53}$$

with

$$\mathbf{z}^{i+1} = \mathbf{z}^i + \delta\mathbf{z}, \tag{54}$$

$$\boldsymbol{\eta}^i = \mathbf{d} - \mathbf{g}\left(\mathbf{z}^i\right), \tag{55}$$

$$\boldsymbol{\xi}^i = \mathbf{z}^{\mathrm{f}} - \mathbf{z}^i. \tag{56}$$

- 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

$$\mathbf{x}_0 = \mathbf{x}_0^{\mathrm{f}} + \mathbf{x}_0', \tag{57}$$

$$\boldsymbol{\theta} = \boldsymbol{\theta}^{\mathrm{f}} + \boldsymbol{\theta}', \tag{58}$$

$$\mathbf{x}_{k+1} = \mathbf{m}(\mathbf{x}_k, \boldsymbol{\theta}), \tag{59}$$

Measurements

$$\mathbf{d} = \mathbf{h}(\mathbf{x}) + \mathbf{e}. \tag{60}$$

Problem formulation

State vector and covariance matrix

$$\mathbf{z} = \begin{pmatrix} \mathbf{x}_0 \\ \boldsymbol{\theta} \end{pmatrix} \quad \text{and} \quad \mathbf{C}_{zz} = \begin{pmatrix} \mathbf{C}_{x_0 x_0} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_{\theta\theta} \end{pmatrix}, \tag{61}$$

SC-4DVar costfunction

$$\mathcal{J}(\mathbf{z}) = \frac{1}{2}\left(\mathbf{z} - \mathbf{z}^{\mathrm{f}}\right)^{\mathrm{T}} \mathbf{C}_{zz}^{-1} \left(\mathbf{z} - \mathbf{z}^{\mathrm{f}}\right) + \frac{1}{2}\left(\mathbf{h}(\mathbf{x}) - \mathbf{d}\right)^{\mathrm{T}} \mathbf{C}_{dd}^{-1} \left(\mathbf{h}(\mathbf{x}) - \mathbf{d}\right), \tag{62}$$

Solve for initial condition and parameter that minimize the cost function

Lagrangian formulation

$$\begin{aligned}\mathcal{L}(\mathbf{x}_0, \ldots, \mathbf{x}_{K+1}, \boldsymbol{\theta}, \boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_{K+1}) = & \frac{1}{2}\left(\mathbf{x}_0 - \mathbf{x}_0^{\mathrm{f}}\right)^{\mathrm{T}} \mathbf{C}_{x_0 x_0}^{-1} \left(\mathbf{x}_0 - \mathbf{x}_0^{\mathrm{f}}\right) \\ & + \frac{1}{2}\left(\boldsymbol{\theta} - \boldsymbol{\theta}^{\mathrm{f}}\right)^{\mathrm{T}} \mathbf{C}_{\theta\theta}^{-1} \left(\boldsymbol{\theta} - \boldsymbol{\theta}^{\mathrm{f}}\right) \\ & + \frac{1}{2}\left(\mathbf{h}(\mathbf{x}) - \mathbf{d}\right)^{\mathrm{T}} \mathbf{C}_{dd}^{-1} \left(\mathbf{h}(\mathbf{x}) - \mathbf{d}\right) \\ & + \sum_{k=0}^{K} \boldsymbol{\lambda}_{k+1}^{\mathrm{T}} \left(\mathbf{x}_{k+1} - \mathbf{m}(\mathbf{x}_k, \boldsymbol{\theta})\right).\end{aligned} \tag{63}$$

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

Gradient of Lagrangian

$$\nabla_{\mathbf{x}_k} \mathcal{L}(\mathbf{x}, \boldsymbol{\theta}, \boldsymbol{\lambda}) = \mathbf{H}_k^{\mathrm{T}} \, \mathbf{C}_{dd}^{-1} \left(\mathbf{h}(\mathbf{x}) - \mathbf{d} \right) + \boldsymbol{\lambda}_k - \mathbf{M}_{x,k}^{\mathrm{T}} \boldsymbol{\lambda}_{k+1}, \tag{64}$$

$$\nabla_{\mathbf{x}_{K+1}} \mathcal{L}(\mathbf{z}, \mathbf{x}, \boldsymbol{\lambda}) = \boldsymbol{\lambda}_{K+1}, \tag{65}$$

$$\begin{aligned} \nabla_{\mathbf{x}_0} \mathcal{L}(\mathbf{x}, \boldsymbol{\theta}, \boldsymbol{\lambda}) &= \mathbf{C}_{zz}^{-1} \left(\mathbf{x}_0 - \mathbf{x}_0^{\mathrm{f}} \right) - \mathbf{M}_{x,0}^{\mathrm{T}} \boldsymbol{\lambda}_1 \\ &= \mathbf{C}_{zz}^{-1} \left(\mathbf{x}_0 - \mathbf{x}_0^{\mathrm{f}} \right) - \boldsymbol{\lambda}_0, \end{aligned} \tag{66}$$

$$\nabla_{\boldsymbol{\theta}} \mathcal{L}(\mathbf{x}, \boldsymbol{\theta}, \boldsymbol{\lambda}) = \mathbf{C}_{\theta\theta}^{-1} \left(\boldsymbol{\theta} - \boldsymbol{\theta}^{\mathrm{f}} \right) - \sum_{k=0}^{K} \mathbf{M}_{\theta,k}^{\mathrm{T}} \boldsymbol{\lambda}_{k+1}, \tag{67}$$

$$\nabla_{\boldsymbol{\lambda}_k} \mathcal{L}(\mathbf{x}, \boldsymbol{\theta}, \boldsymbol{\lambda}) = \mathbf{x}_{k+1} - \mathbf{m}(\mathbf{x}_k, \boldsymbol{\theta}). \tag{68}$$

Euler-Lagrange equation(s)

Forward model

$$\mathbf{x}_0 = \mathbf{x}_0^{\mathrm{f}} + \mathbf{C}_{x_0 x_0} \boldsymbol{\lambda}_0, \tag{69}$$

$$\boldsymbol{\theta} = \boldsymbol{\theta}^{\mathrm{f}} + \mathbf{C}_{\theta\theta} \sum_{k=0}^{K} \mathbf{M}_{\theta,k}^{\mathrm{T}} \boldsymbol{\lambda}_{k+1}, \tag{70}$$

$$\mathbf{x}_{k+1} = \mathbf{m}(\mathbf{x}_k, \boldsymbol{\theta}), \tag{71}$$

Backward model for the adjoint variable

$$\boldsymbol{\lambda}_{K+1} = 0, \tag{72}$$

$$\boldsymbol{\lambda}_k = \mathbf{M}_{x,k}^{\mathrm{T}} \, \boldsymbol{\lambda}_{k+1} - \mathbf{H}_k^{\mathrm{T}} \, \mathbf{C}_{dd}^{-1} \, \big(\mathbf{h}(\mathbf{x}) - \mathbf{d}\big). \tag{73}$$

Coupled two-point boundary-value problem in time.

SC-4DVar algorithm

1: Input: $\mathbf{z}^{\mathrm{f}} \in \Re^n$; $\mathbf{d} \in \Re^m$ ▷ Prior initial conditions and observations
2: $\mathbf{x}_0 = \mathbf{x}_0^{\mathrm{f}}$ ▷ Initialization of $\mathbf{x}_0$
3: $\boldsymbol{\theta} = \boldsymbol{\theta}^{\mathrm{f}}$ ▷ Initialization of $\boldsymbol{\theta}$
4: **repeat** ▷ Iteration loop
5: **for** $k = 0 : K$ **do** ▷ Integrate forward model
6: $\mathbf{x}_{k+1} = \mathbf{m}(\mathbf{x}_k, \boldsymbol{\theta})$
7: **end for**
8: $\boldsymbol{\lambda}_{K+1} = 0$
9: **for** $k = K : 0$ **do** ▷ Integrate backward adjoint model
10: $\boldsymbol{\lambda}_k = \mathbf{M}_{x,k}^{\mathrm{T}} \boldsymbol{\lambda}_{k+1} - \mathbf{H}_k^{\mathrm{T}} \mathbf{C}_{dd}^{-1} (\mathbf{H}\mathbf{x} - \boldsymbol{\eta})$
11: **end for**
12: $\mathbf{x}_0 \leftarrow \mathbf{x}_0 - \gamma \mathbf{B} \nabla_{x_0} \mathcal{L}(\mathbf{x}, \boldsymbol{\theta}, \boldsymbol{\lambda})$ ▷ Update $\mathbf{x}_0$ using Eq. (66)
13: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \gamma \mathbf{B} \nabla_{\theta} \mathcal{L}(\mathbf{x}, \boldsymbol{\theta}, \boldsymbol{\lambda})$ ▷ Update $\boldsymbol{\theta}$ using Eq. (67)
14: **until** convergence

Incremental strong constraint 4DVar

Incremental SC-4DVar

Nonlinear model

$$\mathbf{x}_0 = \mathbf{x}_0^{\mathrm{f}} + \mathbf{x}_0', \tag{74}$$

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

State variable

$$\mathbf{z} = \mathbf{x}_0 \tag{76}$$

We compute updates

$$\mathbf{z}^{i+1} = \mathbf{z}^i + \delta\mathbf{z}, \tag{77}$$

Solving for the increments

Inner incremental SC-4DVar costfunction

$$\mathcal{J}(\delta\mathbf{z}) = \frac{1}{2}\left(\delta\mathbf{z} - \boldsymbol{\xi}^i\right)^{\mathrm{T}} \mathbf{C}_{zz}^{-1}\left(\delta\mathbf{z} - \boldsymbol{\xi}^i\right) + \frac{1}{2}\left(\mathbf{H}^i\delta\mathbf{x} - \boldsymbol{\eta}^i\right)^{\mathrm{T}} \mathbf{C}_{dd}^{-1}\left(\mathbf{H}^i\delta\mathbf{x} - \boldsymbol{\eta}^i\right), \tag{78}$$

Linear model for the increments

$$\delta\mathbf{x}_0 = \delta\mathbf{z}^i, \tag{79}$$

$$\delta\mathbf{x}_{k+1} = \mathbf{M}_k^i \delta\mathbf{x}_k. \tag{80}$$

Prior increment

$$\boldsymbol{\xi}^i = \mathbf{x}_0^{\mathrm{f}} - \mathbf{x}_0^i, \tag{81}$$

Innovation

$$\boldsymbol{\eta}^i = \mathbf{d} - \mathbf{h}(\mathbf{x}^i). \tag{82}$$

Lagrangian formulation

$$\begin{aligned}\mathcal{L}(\delta\mathbf{z}, \delta\mathbf{x}, \delta\boldsymbol{\lambda}) = & \frac{1}{2}\left(\delta\mathbf{z} - \boldsymbol{\xi}^i\right)^{\mathrm{T}} \mathbf{C}_{zz}^{-1} \left(\delta\mathbf{z} - \boldsymbol{\xi}^i\right) \\ & + \frac{1}{2}\left(\mathbf{H}^i\delta\mathbf{x} - \boldsymbol{\eta}^i\right)^{\mathrm{T}} \mathbf{C}_{dd}^{-1} \left(\mathbf{H}^i\delta\mathbf{x} - \boldsymbol{\eta}^i\right) \\ & + \sum_{k=0}^{K} \delta\boldsymbol{\lambda}_{k+1}^{\mathrm{T}} \left(\delta\mathbf{x}_{k+1} - \mathbf{M}_k^i \delta\mathbf{x}_k\right).\end{aligned} \tag{83}$$

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

Gradient of Lagrangian

$$\nabla_{\delta \mathbf{x}_k} \mathcal{L}(\delta \mathbf{z}, \delta \mathbf{x}, \delta \boldsymbol{\lambda}) = \mathbf{H}_k^{i\mathrm{T}} \mathbf{C}_{dd}^{-1} \left(\mathbf{H}^i \delta \mathbf{x} - \boldsymbol{\eta}^i \right) + \delta \boldsymbol{\lambda}_k - \mathbf{M}_k^{i\mathrm{T}} \delta \boldsymbol{\lambda}_{k+1}, \tag{84}$$

$$\nabla_{\delta \mathbf{x}_K} \mathcal{L}(\delta \mathbf{z}, \delta \mathbf{x}, \delta \boldsymbol{\lambda}) = \delta \boldsymbol{\lambda}_{K+1}, \tag{85}$$

$$\begin{aligned} \nabla_{\delta \mathbf{z}} \mathcal{L}(\delta \mathbf{z}, \delta \mathbf{x}, \delta \boldsymbol{\lambda}) &= \mathbf{C}_{zz}^{-1} \left(\delta \mathbf{z} - \boldsymbol{\xi}^i \right) - \mathbf{M}_0^{i\mathrm{T}} \delta \boldsymbol{\lambda}_1 \\ &= \mathbf{C}_{zz}^{-1} \left(\delta \mathbf{z} - \boldsymbol{\xi}^i \right) - \delta \boldsymbol{\lambda}_0, \end{aligned} \tag{86}$$

$$\nabla_{\delta \boldsymbol{\lambda}_k} \mathcal{L}(\delta \mathbf{z}, \delta \mathbf{x}, \delta \boldsymbol{\lambda}) = \delta \mathbf{x}_{k+1} - \mathbf{M}_k^i \delta \mathbf{x}_k. \tag{87}$$

Euler-Lagrange equation(s)

Forward model

$$\delta \mathbf{x}_0 = \xi^i + \mathbf{C}_{zz}\, \delta \boldsymbol{\lambda}_0, \tag{88}$$

$$\delta \mathbf{x}_{k+1} - \mathbf{M}_k^i \delta \mathbf{x}_k = 0, \tag{89}$$

Adjoint model

$$\delta \boldsymbol{\lambda}_{K+1} = 0, \tag{90}$$

$$\delta \boldsymbol{\lambda}_k - {\mathbf{M}_k^i}^{\mathrm{T}} \delta \boldsymbol{\lambda}_{k+1} = {\mathbf{H}_k^i}^{\mathrm{T}} \mathbf{C}_{dd}^{-1} \left(\mathbf{H}^i \delta \mathbf{x} - \boldsymbol{\eta}^i \right). \tag{91}$$

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

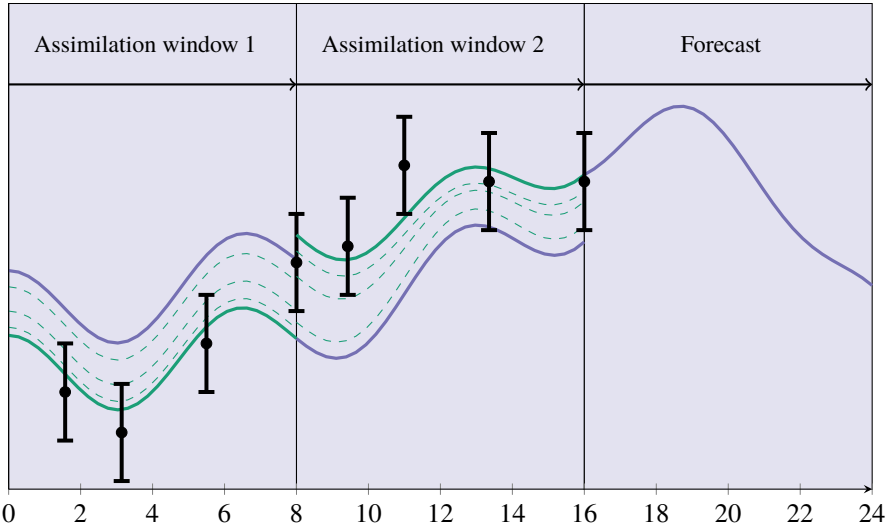
    $\delta\mathbf{z} \leftarrow \delta\mathbf{z} - \gamma \mathbf{B} \left(\delta\mathbf{z} - \boldsymbol{\xi}^i - \mathbf{C}_{zz} \, \delta\boldsymbol{\lambda}_0 \right)$ ▷ Update $\delta\mathbf{z}$ using gradient in Eq. (86)

  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

$$\mathbf{x}_k = \mathbf{m}(\mathbf{x}_{k-1}) + \mathbf{q}_k. \tag{92}$$

Define state vector and model error covariance

$$\mathbf{z} = \mathbf{x} = \begin{pmatrix} \mathbf{x}_0 \\ \vdots \\ \mathbf{x}_K \end{pmatrix}, \quad \text{and} \quad \mathbf{C}_{qq} = \begin{pmatrix} \mathbf{C}_{q_1 q_1} & \cdots & \mathbf{C}_{q_1 q_K} \\ \vdots & \ddots & \vdots \\ \mathbf{C}_{q_K q_1} & \cdots & \mathbf{C}_{q_K q_K} \end{pmatrix}, \tag{93}$$

WC-4DVar cost function

The weak-constraint cost function (generalized inverse formulation)

$$\begin{aligned}\mathcal{J}(\mathbf{x}) = &\frac{1}{2}\left(\mathbf{x}_0 - \mathbf{x}_0^{\mathrm{f}}\right)^{\mathrm{T}} \mathbf{C}_{x_0 x_0}^{-1} \left(\mathbf{x}_0 - \mathbf{x}_0^{\mathrm{f}}\right) \\ &+ \frac{1}{2}\left(\mathbf{h}(\mathbf{x}) - \mathbf{d}\right)^{\mathrm{T}} \mathbf{C}_{dd}^{-1} \left(\mathbf{h}(\mathbf{x}) - \mathbf{d}\right) \\ &+ \frac{1}{2}\sum_{r=1}^{K}\sum_{s=1}^{K}\left(\mathbf{x}_r - \mathbf{m}(\mathbf{x}_{r-1})\right)^{\mathrm{T}} \mathbf{C}_{qq}^{-1}(r,s)\left(\mathbf{x}_s - \mathbf{m}(\mathbf{x}_{s-1})\right).\end{aligned} \tag{94}$$

WC-4DVar: incremental form

Define update

$$\mathbf{x}^{i+1} = \mathbf{x}^{i} + \delta\mathbf{x}. \tag{95}$$

Linearized model

$$\begin{aligned} \mathbf{m}\left(\mathbf{x}_k^{i+1}\right) &= \mathbf{m}\left(\mathbf{x}_k^{i} + \delta\mathbf{x}_k\right) \\ &\approx \mathbf{m}\left(\mathbf{x}_k^{i}\right) + \mathbf{M}_k\delta\mathbf{x}_k, \end{aligned} \tag{96}$$

Linearized measurement operator

$$\begin{aligned} \mathbf{h}\left(\mathbf{x}^{i+1}\right) &= \mathbf{h}\left(\mathbf{x}^{i} + \delta\mathbf{x}\right) \\ &\approx \mathbf{h}\left(\mathbf{x}^{i}\right) + \mathbf{H}\delta\mathbf{x}, \end{aligned} \tag{97}$$

WC-4DVar: incremental form

Write the model residual in Eq. (94) as

$$\begin{aligned}\mathbf{x}_k^{i+1} - \mathbf{m}\left(\mathbf{x}_{k-1}^{i+1}\right) &\approx \mathbf{x}_k^{i+1} - \mathbf{m}\left(\mathbf{x}_{k-1}^{i}\right) - \mathbf{M}_{k-1}\delta\mathbf{x}_{k-1} \\ &= \mathbf{x}_k^{i+1} - \mathbf{x}_k^{i} + \mathbf{x}_k^{i} - \mathbf{m}\left(\mathbf{x}_{k-1}^{i}\right) - \mathbf{M}_{k-1}\delta\mathbf{x}_{k-1} \\ &= \delta\mathbf{x}_k - \mathbf{M}_{k-1}\delta\mathbf{x}_{k-1} + \boldsymbol{\xi}_k^i,\end{aligned} \tag{98}$$

where

$$\boldsymbol{\xi}_k^i = \mathbf{x}_k^i - \mathbf{m}\left(\mathbf{x}_{k-1}^i\right). \tag{99}$$

Innovations becomes

$$\boldsymbol{\eta}^i = \mathbf{d} - \mathbf{h}\left(\mathbf{x}^i\right). \tag{100}$$

WC incremental cost function

The weak-constraint incremental cost function

$$\begin{aligned}\mathcal{J}(\delta\mathbf{x}) = &\frac{1}{2}\left(\delta\mathbf{x}_0 - \boldsymbol{\xi}_0^i\right)^{\mathrm{T}}\mathbf{C}_{x_0x_0}^{-1}\left(\delta\mathbf{x}_0 - \boldsymbol{\xi}_0^i\right)\\ &+\frac{1}{2}\left(\mathbf{H}^i\delta\mathbf{x} - \boldsymbol{\eta}^i\right)^{\mathrm{T}}\mathbf{C}_{dd}^{-1}\left(\mathbf{H}^i\delta\mathbf{x} - \boldsymbol{\eta}^i\right)\\ &+\frac{1}{2}\sum_{r=1}^{K}\sum_{s=1}^{K}\left(\delta\mathbf{x}_r - \mathbf{M}_{r-1}\delta\mathbf{x}_{r-1} + \boldsymbol{\xi}_r^i\right)^{\mathrm{T}}\mathbf{C}_{qq}^{-1}(r,s)\left(\delta\mathbf{x}_s - \mathbf{M}_{s-1}\delta\mathbf{x}_{s-1} + \boldsymbol{\xi}_s^i\right).\end{aligned} \tag{101}$$

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

$$\begin{aligned}\nabla_{\delta\mathbf{x}_k}\mathcal{J}(\delta\mathbf{x}) = {\mathbf{H}_k^i}^{\mathrm{T}}\mathbf{C}_{dd}^{-1}\left(\mathbf{H}^i\delta\mathbf{x}-\boldsymbol{\eta}^i\right)\\ +\sum_{s=1}^{K}\mathbf{C}_{qq}^{-1}(k,s)\left(\delta\mathbf{x}_s-\mathbf{M}_{s-1}^i\delta\mathbf{x}_{s-1}+\boldsymbol{\xi}_s^i\right)\\ -{\mathbf{M}_k^i}^{\mathrm{T}}\sum_{s=1}^{K}\mathbf{C}_{qq}^{-1}(k+1,s)\left(\delta\mathbf{x}_s-\mathbf{M}_{s-1}^i\delta\mathbf{x}_{s-1}+\boldsymbol{\xi}_s^i\right).\end{aligned} \tag{102}$$

Define an adjoint vector for each time step as

$$\delta\boldsymbol{\lambda}_k=\sum_{s=1}^{K}\mathbf{C}_{qq}^{-1}(k,s)\left(\delta\mathbf{x}_s-\mathbf{M}_{s-1}^i\delta\mathbf{x}_{s-1}+\boldsymbol{\xi}_s^i\right), \tag{103}$$

and write Eq. (102) as

$$\nabla_{\delta\mathbf{x}_k}\mathcal{J}(\delta\mathbf{x})={\mathbf{H}_k^i}^{\mathrm{T}}\mathbf{C}_{dd}^{-1}\left(\mathbf{H}^i\delta\mathbf{x}-\boldsymbol{\eta}^i\right)+\delta\boldsymbol{\lambda}_k-{\mathbf{M}_k^i}^{\mathrm{T}}\delta\boldsymbol{\lambda}_{k+1}. \tag{104}$$

Euler-Lagrange equation(s)

Forward model

$$\delta \mathbf{x}_0 = \boldsymbol{\xi}_0^i + \mathbf{C}_{x_0 x_0} \delta \boldsymbol{\lambda}_0, \tag{105}$$

$$\delta \mathbf{x}_k - \mathbf{M}_{k-1}^i \delta \mathbf{x}_{k-1} = -\boldsymbol{\xi}_k^i + \sum_{s=1}^{K} \mathbf{C}_{qq}(k,s) \delta \boldsymbol{\lambda}_s, \tag{106}$$

Backward model

$$\delta \boldsymbol{\lambda}_{K+1} = \mathbf{0}, \tag{107}$$

$$\delta \boldsymbol{\lambda}_k - \mathbf{M}_k^{i\mathrm{T}} \delta \boldsymbol{\lambda}_{k+1} = -\mathbf{H}_k^{i\mathrm{T}} \mathbf{C}_{dd}^{-1} \left(\mathbf{H}^i \delta \mathbf{x} - \boldsymbol{\eta}^i \right). \tag{108}$$

Representer method

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

$$\delta\mathbf{x} = \delta\mathbf{x}^{\mathrm{f}} + \sum_{p=1}^{m} b_p \mathbf{r}_p = \delta\mathbf{x}^{\mathrm{f}} + \mathbf{R}\mathbf{b}, \tag{109}$$

$$\delta\boldsymbol{\lambda} = \delta\boldsymbol{\lambda}^{\mathrm{f}} + \sum_{p=1}^{m} b_p \mathbf{s}_p = \delta\boldsymbol{\lambda}^{\mathrm{f}} + \mathbf{S}\mathbf{b}. \tag{110}$$

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

$$\delta \mathbf{x}_0^{\mathrm{f}} = \boldsymbol{\xi}_0^i, \tag{111}$$

$$\delta \mathbf{x}_k^{\mathrm{f}} - \mathbf{M}_{k-1}^i \delta \mathbf{x}_{k-1}^{\mathrm{f}} = -\boldsymbol{\xi}_k^i, \tag{112}$$

$$\delta \boldsymbol{\lambda}_{K+1}^{\mathrm{f}} = 0, \tag{113}$$

$$\delta \boldsymbol{\lambda}_k^{\mathrm{f}} - \mathbf{M}_k^{i\,\mathrm{T}} \delta \boldsymbol{\lambda}_{k+1}^{\mathrm{f}} = 0. \tag{114}$$

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

Derivation for the representers

$$\mathbf{R}_0\mathbf{b} = \mathbf{C}_{x_0x_0}\mathbf{M}_0^{i\,\mathrm{T}}\mathbf{S}_1\mathbf{b}, \tag{115}$$

$$\left(\mathbf{R}_k\mathbf{b} - \mathbf{M}_{k-1}^{i}\mathbf{R}_{k-1}\mathbf{b}\right) = \sum_{s=1}^{K}\mathbf{C}_{qq}(k,s)\mathbf{S}_s\mathbf{b}, \tag{116}$$

$$\mathbf{S}_{K+1}\mathbf{b} = \mathbf{0}, \tag{117}$$

$$\mathbf{S}_k\mathbf{b} - \mathbf{M}_k^{i\,\mathrm{T}}\mathbf{S}_{k+1}\mathbf{b} = \mathbf{H}_k^{i\,\mathrm{T}}\mathbf{C}_{dd}^{-1}\left(\boldsymbol{\eta}^i - \mathbf{H}^i(\delta\mathbf{x}^{\mathrm{f}} + \mathbf{R}\mathbf{b})\right). \tag{118}$$

Define $\mathbf{b}$ as

$$\mathbf{b} = \mathbf{C}_{dd}^{-1}\left(\boldsymbol{\eta}^i - \mathbf{H}^i(\delta\mathbf{x}^{\mathrm{f}} + \mathbf{R}\mathbf{b})\right), \tag{119}$$

such that Eq. (118) simplifies to

$$\mathbf{S}_k\mathbf{b} - \mathbf{M}_k^{i\,\mathrm{T}}\mathbf{S}_{k+1}\mathbf{b} = \mathbf{H}_k^{i\,\mathrm{T}}\mathbf{b}. \tag{120}$$

Euler-Lagrange equations: Representers

$$\mathbf{R}_0 = \mathbf{C}_{x_0 x_0} \mathbf{M}_0^{i\,\mathrm{T}} \mathbf{S}_1, \tag{121}$$

$$\mathbf{R}_k - \mathbf{M}_{k-1}^i \mathbf{R}_{k-1} = \sum_{s=1}^{K} \mathbf{C}_{qq}(k,s) \mathbf{S}_s, \tag{122}$$

$$\mathbf{S}_{K+1} = \mathbf{0}, \tag{123}$$

$$\mathbf{S}_k - \mathbf{M}_k^{i\,\mathrm{T}} \mathbf{S}_{k+1} = \mathbf{H}_k^{i\,\mathrm{T}}. \tag{124}$$

Requires $2m$ model integrations.

Euler-Lagrange equations: **b**

Linear system for the representer coefficients b

$$\left(\mathbf{H}^i\mathbf{R} + \mathbf{C}_{dd}\right)\mathbf{b} = \boldsymbol{\eta}^i - \mathbf{H}^i\delta\mathbf{x}^{\mathrm{f}}. \tag{125}$$

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

More efficient solution method

Given $\mathbf{b}$ we obtain the final solution from (108)

Adjoint equation forced by b

$$\delta\boldsymbol{\lambda}_k - \mathbf{M}_k^{i\mathrm{T}}\delta\boldsymbol{\lambda}_{k+1} = \mathbf{H}_k^{i\mathrm{T}}\mathbf{b}. \tag{126}$$

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:

$$\mathbf{C}\mathbf{b} = \boldsymbol{\nu}. \tag{127}$$

Solving Eq. (127) is equivalent to minimizing the functional

$$\phi(\mathbf{b}) = \frac{1}{2}\mathbf{b}^{\mathrm{T}}\mathbf{C}\mathbf{b} - \mathbf{b}^{\mathrm{T}}\boldsymbol{\nu}, \tag{128}$$

which has a gradient

$$\nabla_{\mathbf{b}}\phi(\mathbf{b}) = \boldsymbol{\rho} = \mathbf{C}\mathbf{b} - \boldsymbol{\nu}. \tag{129}$$

Iterative solution

$$\mathbf{b}^{i+1} = \mathbf{b}^{i} - \gamma\boldsymbol{\rho}, \tag{130}$$

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 (115–118) as

$$\mathbf{c}_0 = \mathbf{C}_{x_0 x_0} \mathbf{M}_0^{i\,\mathrm{T}} \boldsymbol{\psi}_1, \tag{131}$$

$$\mathbf{c}_k = \mathbf{M}_{k-1}^{i} \mathbf{c}_{k-1} + \sum_{s=1}^{K} \mathbf{C}_{qq}(k,s) \boldsymbol{\psi}_s, \tag{132}$$

$$\boldsymbol{\psi}_{K+1} = \mathbf{0}, \tag{133}$$

$$\boldsymbol{\psi}_k = \mathbf{M}_k^{i\,\mathrm{T}} \boldsymbol{\psi}_{k+1} + \mathbf{H}_k^{i\,\mathrm{T}} \mathbf{v}. \tag{134}$$

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

$$\mathbf{Hc} = \mathbf{HRv} = \mathcal{R}\mathbf{v}, \tag{135}$$

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

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{k} \times \mathbf{u} = \frac{\partial}{\partial z}\left(A \frac{\partial \mathbf{u}}{\partial z}\right), \tag{136}$$

Initial condition

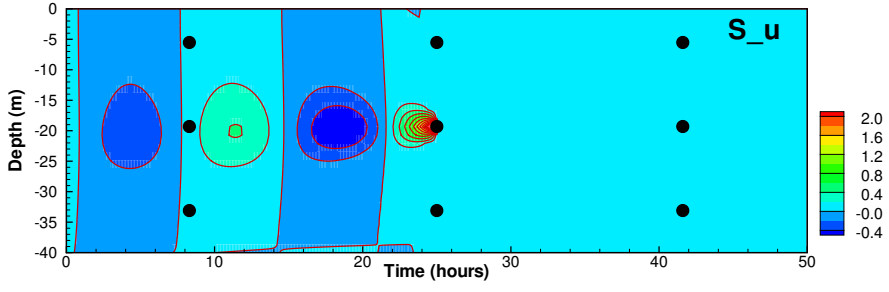
$$\mathbf{u}(z, 0) = \mathbf{u}_0, \tag{137}$$

boundary conditions

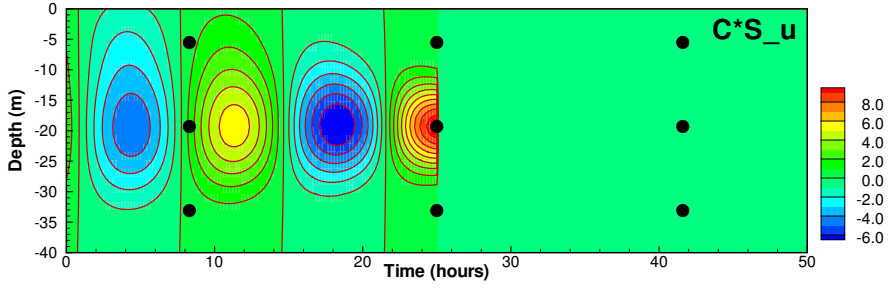
$$A \frac{\partial \mathbf{u}}{\partial z}\bigg|_{z=0} = \left(c_d \sqrt{u_a^2 + v_a^2}\right) \mathbf{u}_a, \tag{138}$$

$$A \frac{\partial \mathbf{u}}{\partial z}\bigg|_{z=-H} = \mathbf{0}, \tag{139}$$

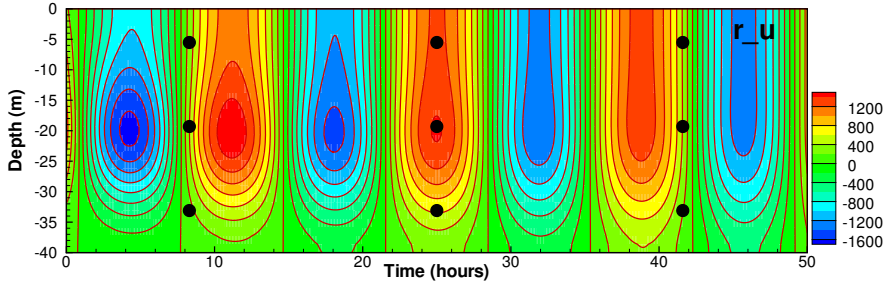
An adjoint representer s



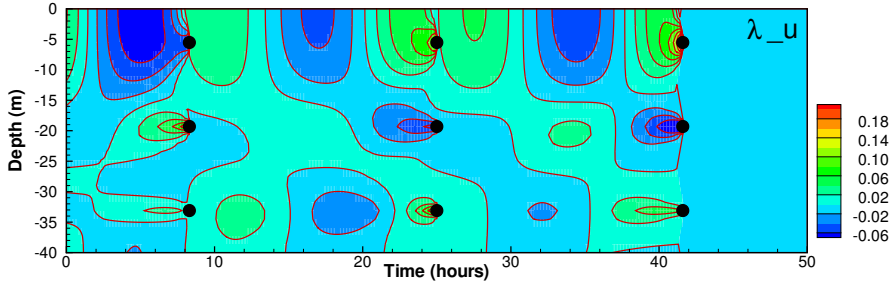
A convolution of an adjoint representer $\mathbf{C}_{qq} \circ \mathbf{s}$



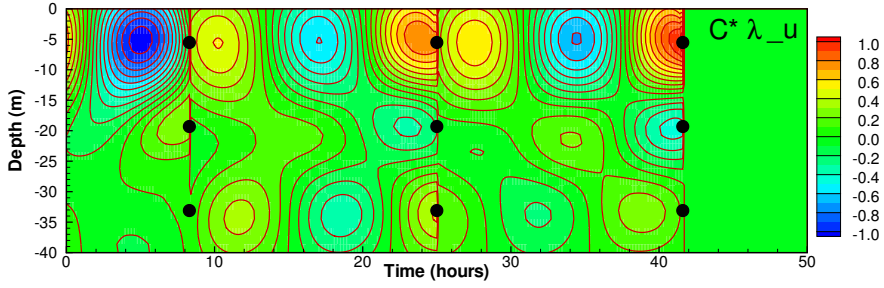
A representer **r**



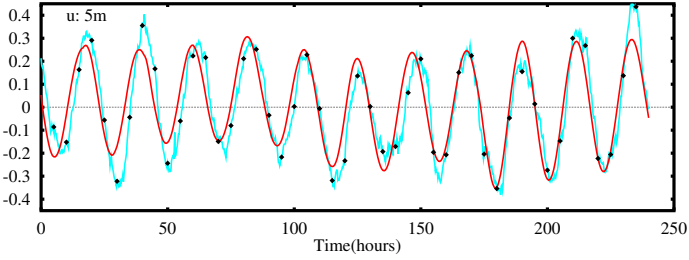
The adjoint λ

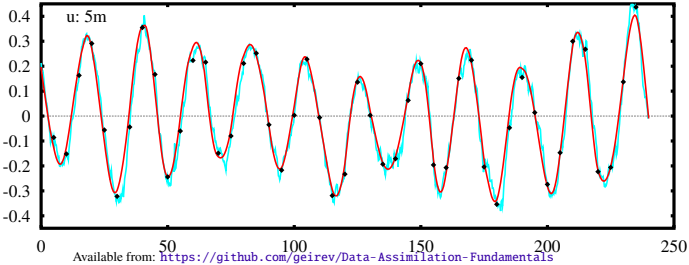


The convolution of the adjoint $\mathbf{C}_{qq} \circ \lambda$



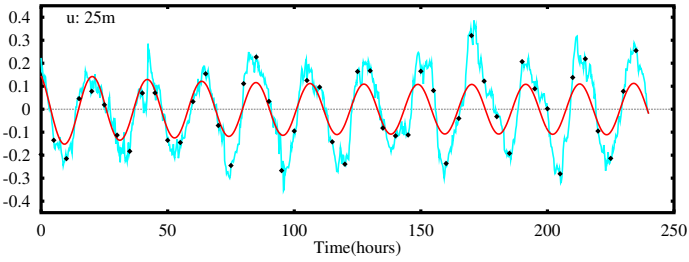
Strong and weak constraint solution

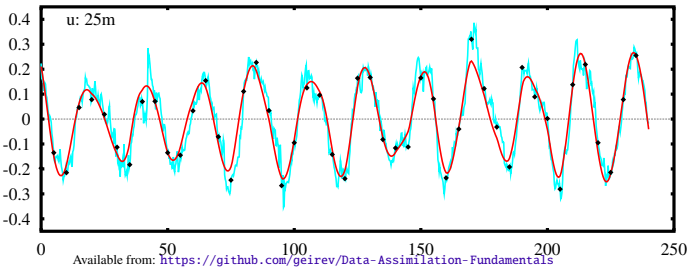




Available from: https://github.com/geirev/Data-Assimilation-Fundamentals

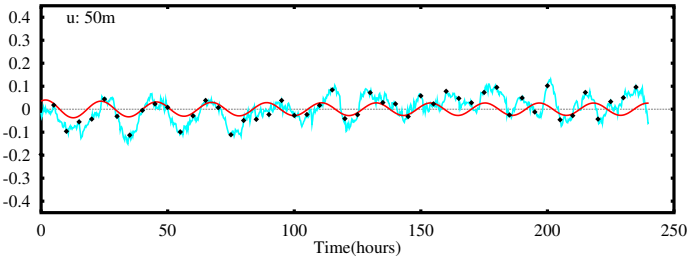
Strong and weak constraint solution

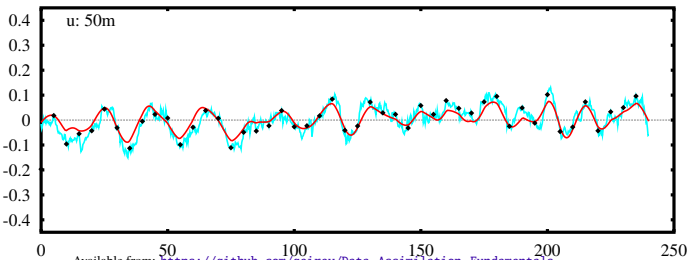




Available from: https://github.com/geirev/Data-Assimilation-Fundamentals

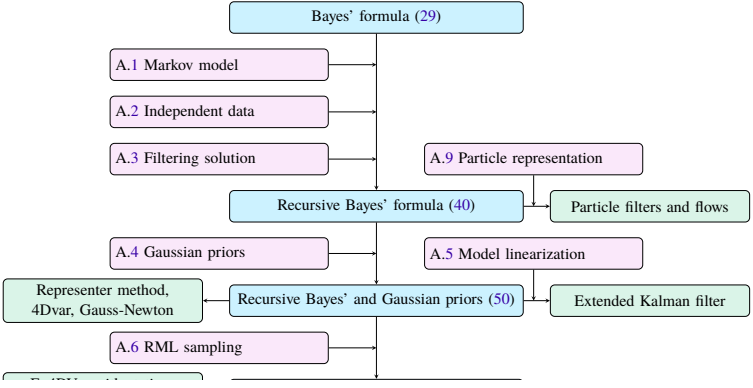
Strong and weak constraint solution



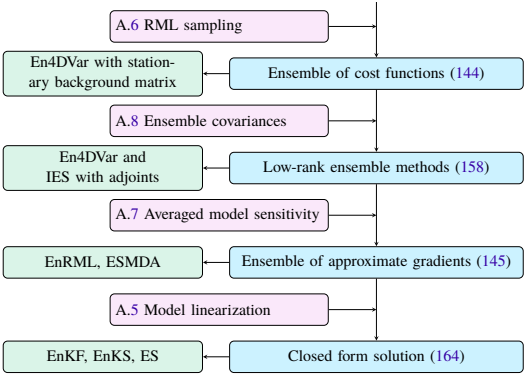


Available from: https://github.com/geirev/Data-Assimilation-Fundamentals

Overview of approximations and methods



Overview of approximations and methods



Kalman filters and 3D-Var

Linearization leads to an approximate explicit solution

Approximation 5 (Linearization)

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

$$\mathbf{g}(\mathbf{z}) \approx \mathbf{g}(\mathbf{z}^{\mathrm{f}}) + \mathbf{G}(\mathbf{z} - \mathbf{z}^{\mathrm{f}}), \tag{140}$$

and approximate the gradient evaluated at the prior estimate

$$\nabla_{\mathbf{z}}\mathbf{g}(\mathbf{z}) \approx \mathbf{G}^{\mathrm{T}}, \tag{141}$$

where we have defined

$$\mathbf{G}^{\mathrm{T}} = \nabla_{\mathbf{z}}\mathbf{g}(\mathbf{z})\big|_{\mathbf{z}=\mathbf{z}^{\mathrm{f}}}. \tag{142}$$

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

Extended Kalman Filter solution

The linearization in Approx. 5 leads to closed form solution of (51)

$$\mathbf{z}^{\mathrm{a}} = \mathbf{z}^{\mathrm{f}} + \mathbf{C}_{zz}\mathbf{G}^{\mathrm{T}}\left(\mathbf{G}\mathbf{C}_{zz}\mathbf{G}^{\mathrm{T}} + \mathbf{C}_{dd}\right)^{-1}\left(\mathbf{d} - \mathbf{g}\left(\mathbf{z}^{\mathrm{f}}\right)\right). \tag{143}$$

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. (144).

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

RML minimizes an ensemble of cost functions

Ensemble of cost functions

$$\mathcal{J}(\mathbf{z}_j) = \frac{1}{2}\left(\mathbf{z}_j - \mathbf{z}_j^{\mathrm{f}}\right)^{\mathrm{T}} \mathbf{C}_{zz}^{-1}\left(\mathbf{z}_j - \mathbf{z}_j^{\mathrm{f}}\right) + \frac{1}{2}\left(\mathbf{g}(\mathbf{z}_j) - \mathbf{d}_j\right)^{\mathrm{T}} \mathbf{C}_{dd}^{-1}\left(\mathbf{g}(\mathbf{z}_j) - \mathbf{d}_j\right), \tag{144}$$

Ensemble of gradients set to zero

$$\mathbf{C}_{zz}^{-1}\left(\mathbf{z}_j - \mathbf{z}_j^{\mathrm{f}}\right) + \nabla_{\mathbf{z}}\mathbf{g}(\mathbf{z}_j)\mathbf{C}_{dd}^{-1}\left(\mathbf{g}(\mathbf{z}_j) - \mathbf{d}_j\right) = 0. \tag{145}$$

Thus, we must solve N independent minimizations.

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

Ensemble of incremental 4DVars

$$\mathcal{J}(\delta\mathbf{z}_j) = \frac{1}{2}\left(\delta\mathbf{z}_j - \boldsymbol{\xi}_j^i\right)^{\mathrm{T}} \mathbf{C}_{zz}^{-1} \left(\delta\mathbf{z}_j - \boldsymbol{\xi}_j^i\right) + \frac{1}{2}\left(\mathbf{G}_j^i\delta\mathbf{z}_j - \boldsymbol{\eta}_j^i\right)^{\mathrm{T}} \mathbf{C}_{dd}^{-1} \left(\mathbf{G}_j^i\delta\mathbf{z}_j - \boldsymbol{\eta}_j^i\right). \tag{146}$$

Ensemble of GN iterations

$$\mathbf{z}_j^{i+1} = \mathbf{z}_j^i - \gamma\left(\mathbf{C}_{zz}^{-1} + \mathbf{G}_j^{i\mathrm{T}}\mathbf{C}_{dd}^{-1}\mathbf{G}_j^i\right)^{-1}\left(\mathbf{C}_{zz}^{-1}\left(\mathbf{z}_j^i - \mathbf{z}_j^{\mathrm{f}}\right) + \mathbf{G}_j^{i\mathrm{T}}\mathbf{C}_{dd}^{-1}\left(\mathbf{g}\left(\mathbf{z}_j^i\right) - \mathbf{d}_j\right)\right), \tag{147}$$

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

$$\mathbf{z}_j^{\mathrm{a}} = \mathbf{z}_j^{\mathrm{f}} + \mathbf{C}_{zz}\mathbf{G}_j^{\mathrm{T}}\left(\mathbf{G}_j\mathbf{C}_{zz}\mathbf{G}_j^{\mathrm{T}} + \mathbf{C}_{dd}\right)^{-1}\left(\mathbf{d}_j - \mathbf{g}(\mathbf{z}_j^{\mathrm{f}})\right). \tag{148}$$

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?

Use an averaged model sensitivity to avoid adjoints

Approximation 7 (Best-fit averaged model sensitivity)

Interpret $\mathbf{G}_j$ in Eq. (148) and $\mathbf{G}_j^i$ in Eq. (147) as the sensitivity matrix in linear regression and represent them using the definition

$$\mathbf{G}_j \approx \mathbf{G} \triangleq \mathbf{C}_{yz}\mathbf{C}_{zz}^{-1}. \tag{149}$$

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

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.

$$\mathbf{Z} = \left(\mathbf{z}_1,\ \mathbf{z}_2,\ \ldots\ ,\ \mathbf{z}_N\right). \tag{150}$$

Define the projection $\mathbf{\Pi} \in \mathfrak{R}^{N \times N}$ as

$$\mathbf{\Pi} = \left(\mathbf{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^{\mathrm{T}}\right) \Big/ \sqrt{N-1}, \tag{151}$$

Ensemble anomalies

$$\mathbf{A} = \mathbf{Z}\mathbf{\Pi}, \tag{152}$$

Ensemble matrices

Ensemble covariance

$$\overline{\mathbf{C}}_{zz} = \mathbf{A}\mathbf{A}^{\mathrm{T}}. \tag{153}$$

Measurement error covariance matrix

$$\overline{\mathbf{C}}_{dd} = \mathbf{E}\mathbf{E}^{\mathrm{T}}. \tag{154}$$

Ensemble of model-predicted measurements

$$\boldsymbol{\Upsilon} = \mathbf{g}(\mathbf{Z}), \tag{155}$$

with anomalies

$$\mathbf{Y} = \boldsymbol{\Upsilon}\boldsymbol{\Pi}, \tag{156}$$

Solution is confined to ensemble subspace

Search for the solution in the ensemble subspace

$$\mathbf{Z}^{\mathrm{a}} = \mathbf{Z}^{\mathrm{f}} + \mathbf{A}\mathbf{W}. \tag{157}$$

Cost function in ensemble subspace

$$\mathcal{J}(\mathbf{w}_j) = \frac{1}{2}\mathbf{w}_j^{\mathrm{T}}\mathbf{w}_j + \frac{1}{2}\left(\mathbf{g}\left(\mathbf{z}_j^{\mathrm{f}} + \mathbf{A}\mathbf{w}_j\right) - \mathbf{d}_j\right)^{\mathrm{T}}\overline{\mathbf{C}}_{dd}^{-1}\left(\mathbf{g}\left(\mathbf{z}_j^{\mathrm{f}} + \mathbf{A}\mathbf{w}_j\right) - \mathbf{d}_j\right), \tag{158}$$

Gradient and Hessian

Gradient

$$\nabla_{\mathbf{w}} \mathcal{J}(\mathbf{w}_j) = \mathbf{w}_j + \left(\mathbf{G}_j \mathbf{A}\right)^{\mathrm{T}} \overline{\mathbf{C}}_{dd}^{-1} \left(\mathbf{g}\left(\mathbf{z}_j^{\mathrm{f}} + \mathbf{A}\mathbf{w}_j\right) - \mathbf{d}_j\right), \tag{159}$$

Approximate Hessian

$$\nabla_{\mathbf{w}} \nabla_{\mathbf{w}} \mathcal{J}\left(\mathbf{w}_j\right) \approx \mathbf{I} + \left(\mathbf{G}_j \mathbf{A}\right)^{\mathrm{T}} \overline{\mathbf{C}}_{dd}^{-1} \left(\mathbf{G}_j \mathbf{A}\right). \tag{160}$$

Tangent-linear model

$$\mathbf{G}_j = \left(\nabla_{\mathbf{w}} \mathbf{g}|_{\mathbf{z}_j^{\mathrm{f}} + \mathbf{A}\mathbf{w}_j}\right)^{\mathrm{T}} \in \mathfrak{R}^{m \times n}, \tag{161}$$

Gauss-Newton iterations

$$\mathbf{w}_j^{i+1} = \mathbf{w}_j^i - \gamma\left\{\mathbf{w}_j^i - \left(\mathbf{G}_j^i\mathbf{A}\right)^{\mathrm{T}}\left(\left(\mathbf{G}_j^i\mathbf{A}\right)\left(\mathbf{G}_j^i\mathbf{A}\right)^{\mathrm{T}} + \overline{\mathbf{C}}_{dd}\right)^{-1}\left(\left(\mathbf{G}_j^i\mathbf{A}\right)\mathbf{w}_j^i + \mathbf{d}_j - \mathbf{g}\left(\mathbf{z}_j^{\mathrm{f}} + \mathbf{A}\mathbf{w}_j^i\right)\right)\right\}. \tag{162}$$

with linear regression approximation for model sensitivities

$$\mathbf{G}_j^i \approx \overline{\mathbf{G}}^i \triangleq \overline{\mathbf{C}}_{yz}^i \overline{\mathbf{C}}_{zz}^{i\,\dagger} = \mathbf{Y}^i \mathbf{A}^{i\dagger}, \tag{163}$$

Subspace EnRML

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

EnKF update

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

$$\mathbf{Z}^{\mathrm{a}} = \mathbf{Z}^{\mathrm{f}} + \mathbf{A}\mathbf{Y}^{\mathrm{T}}\left(\mathbf{Y}\mathbf{Y}^{\mathrm{T}} + \mathbf{E}\mathbf{E}^{\mathrm{T}}\right)^{-1}\left(\mathbf{D} - \mathbf{g}(\mathbf{Z}^{\mathrm{f}})\right) \tag{164}$$

Alternative interpretation using

$$\mathbf{W} = \mathbf{Y}^{\mathrm{T}}\left(\mathbf{Y}\mathbf{Y}^{\mathrm{T}} + \mathbf{E}\mathbf{E}^{\mathrm{T}}\right)^{-1}\left(\mathbf{D} - \mathbf{g}(\mathbf{Z}^{\mathrm{f}})\right), \tag{165}$$

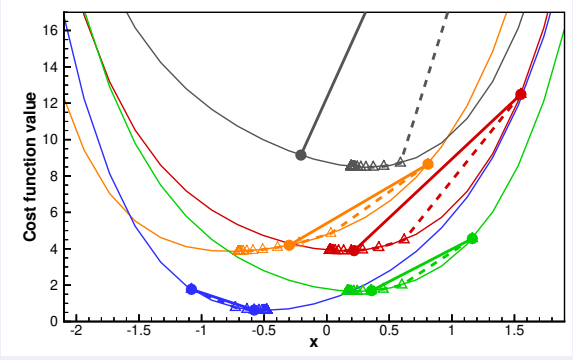
to get

$$\mathbf{Z}^{\mathrm{a}} = \mathbf{Z}^{\mathrm{f}}\left(\mathbf{I} + \mathbf{W}/\sqrt{N-1}\right) \tag{166}$$

But note that

$$\mathbf{Y} = \begin{cases} \mathbf{Y} & \text{for } n \geq N-1 \\ \mathbf{Y}\mathbf{A}^{\dagger}\mathbf{A} & \text{for } n < N-1. \end{cases} \tag{167}$$

ES and IES illustration: Non-linear model



- IES gets closer to minimum than ES
- Approximate sampling of posterior pdf.

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

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.

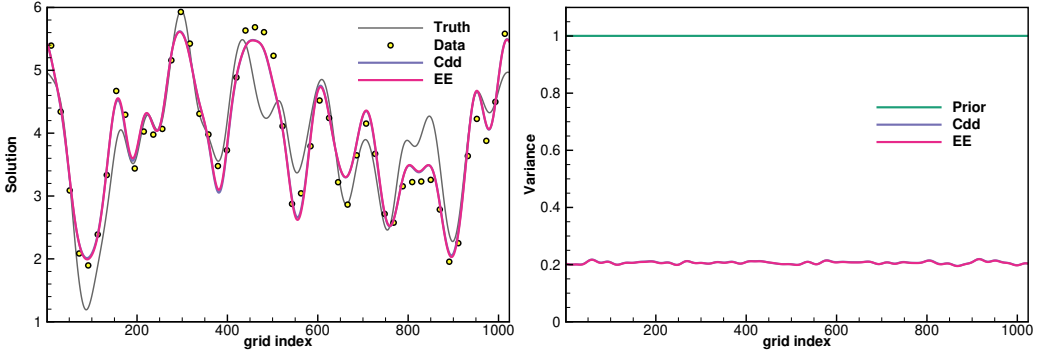
- Small updates reduce impact of the linear approximation.
- ESMDA is identical to ES in the linear case.
- Remember to resample measurement perturbations for each update step.

Some publications:

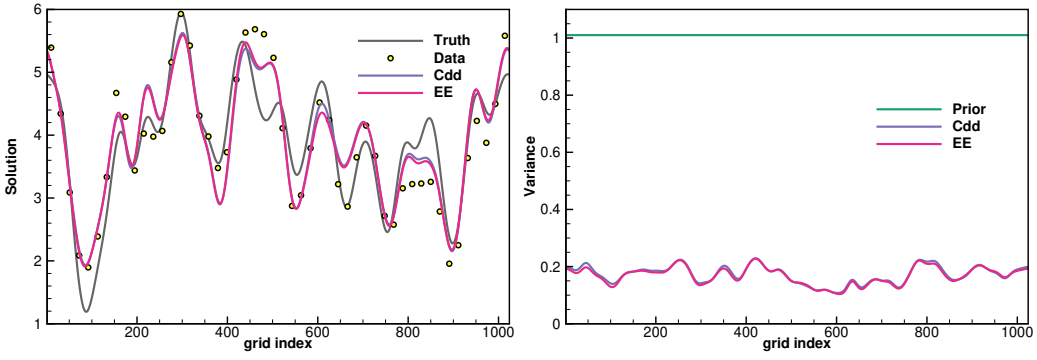
- Ensemble Randomized Maximum Likelihood EnRML (Chen and Oliver, 2013).
- Ensemble DA with multiple updates ESMDA (Emerick and Reynolds, 2013).
- Analysis of iterative ensemble smoothers (Evensen, 2018).
- IES with model errors (Evensen, 2019).
- Ensemble subspace RML (Evensen et al., 2019).

Linear EnKF update examples

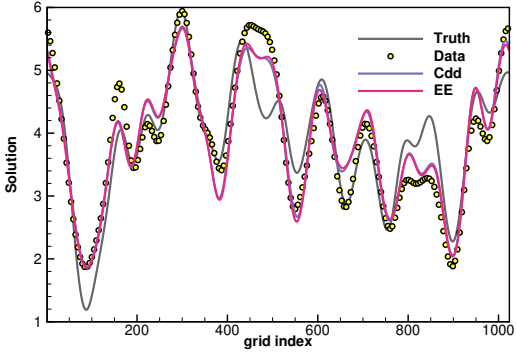
EnKF update with large ensemble size $N = 2000$ and $m = 50$

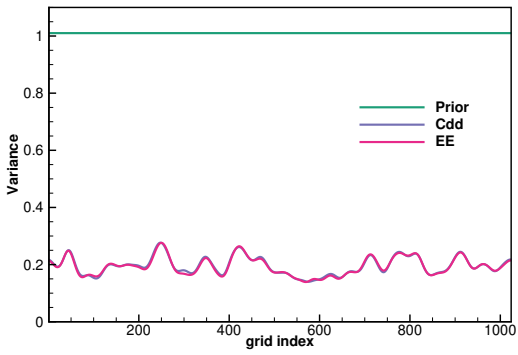


EnKF update with normal ensemble size $N = 100$ and $m = 50$



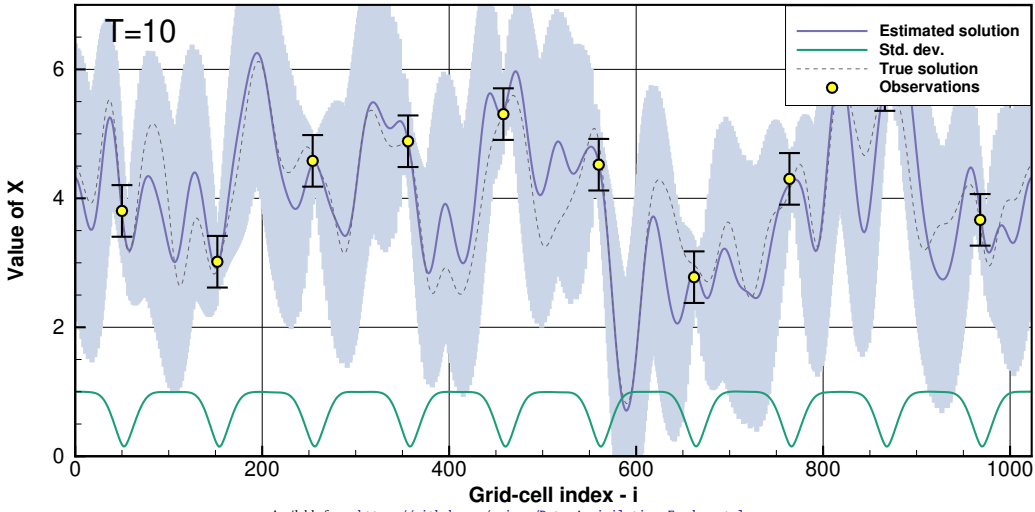
EnKF update with $N = 100$ and many measurements $m = 200$



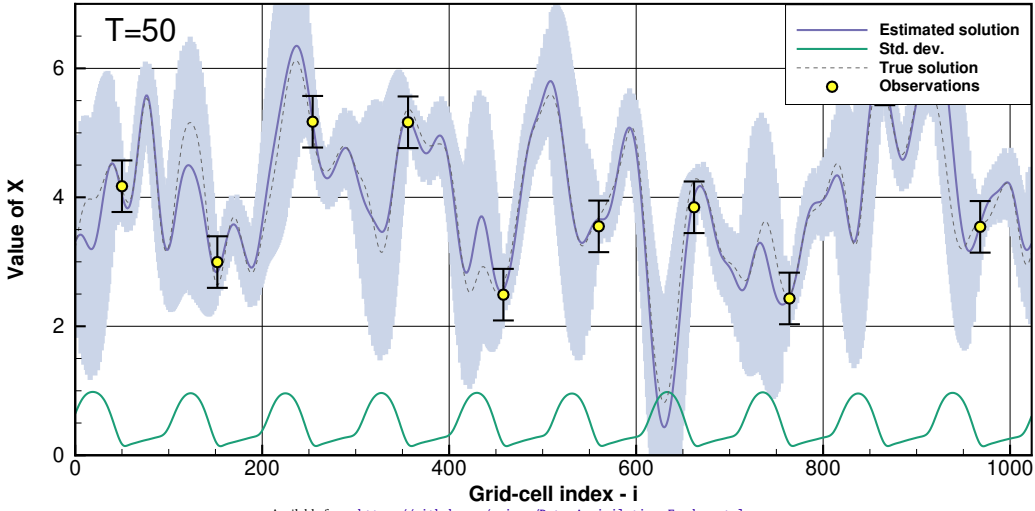


EnKF for an advection equation

EnKF with the advection equation after two updates

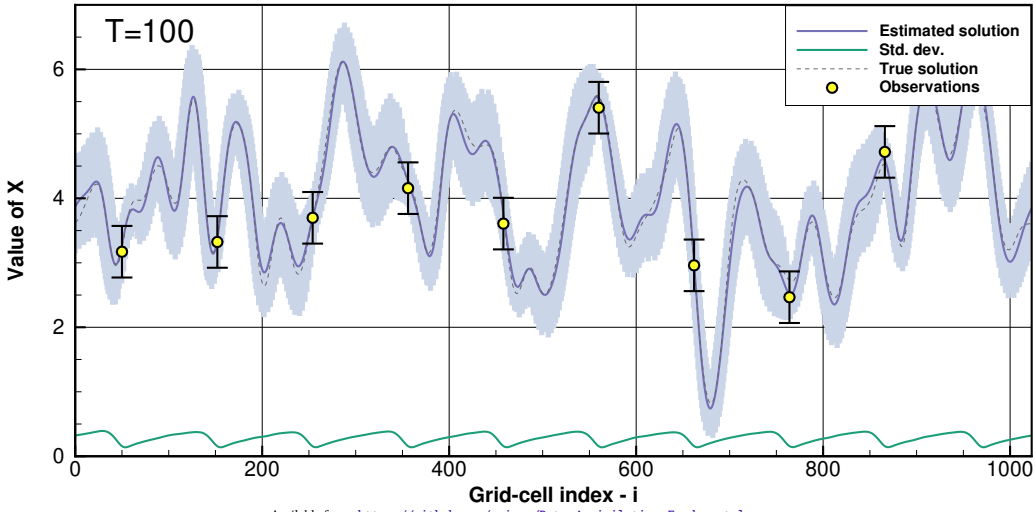


EnKF with the advection equation after ten updates



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

EnKF with the advection equation after twenty updates



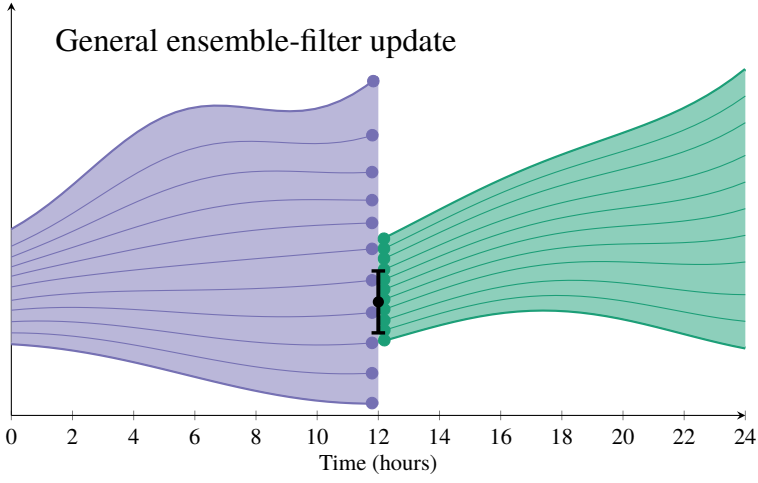
Available from: https://github.com/geirev/Data-Assimilation-Fundamentals

EnKF with the advection equation: Animation

Animation

EnKF with the Lorenz equations

General filter formulation



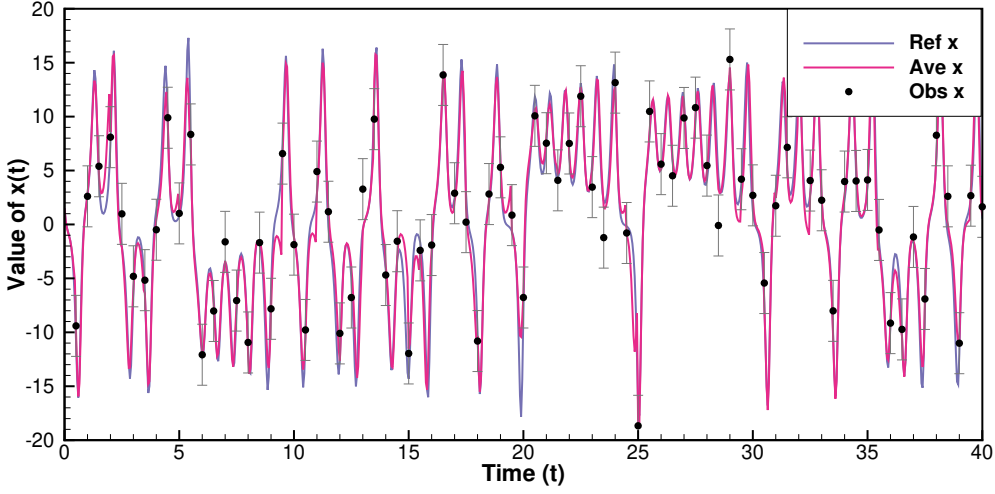
EnKF with the Lorenz model

$$\frac{\partial x}{\partial t} = \sigma(y - x), \tag{168}$$

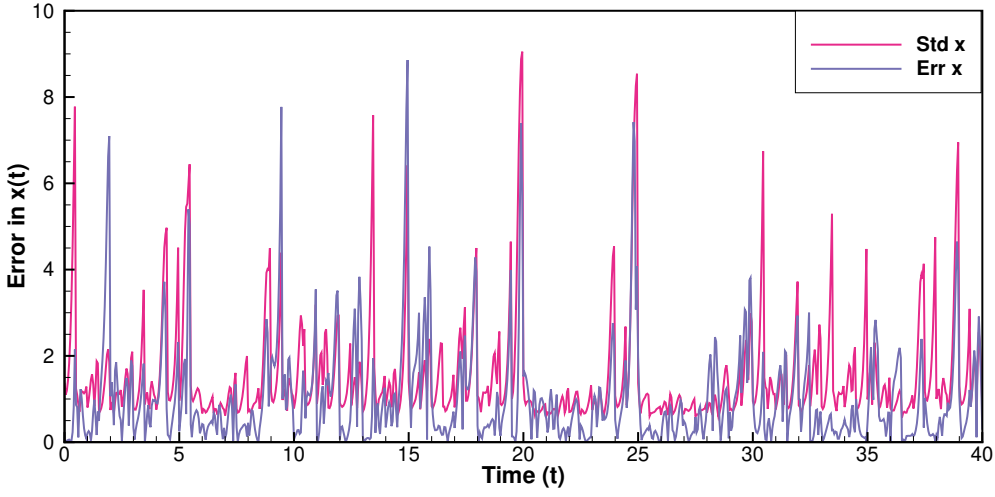
$$\frac{\partial y}{\partial t} = \rho x - y - xz, \tag{169}$$

$$\frac{\partial z}{\partial t} = xy - \beta z. \tag{170}$$

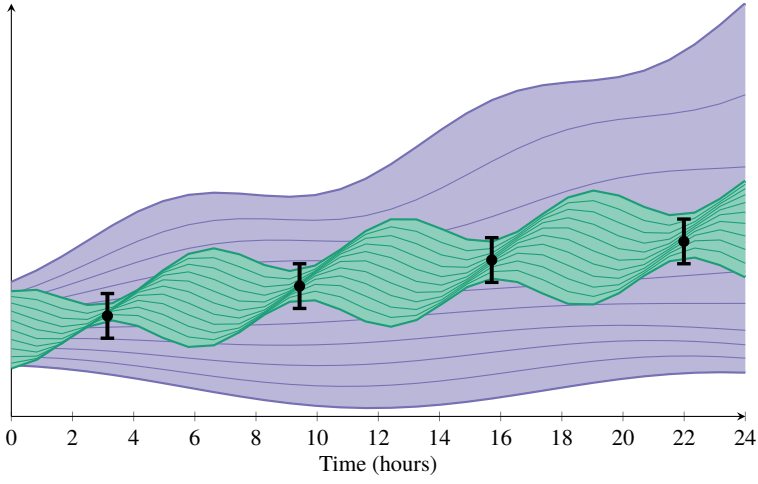
EnKF with the Lorenz model: estimate



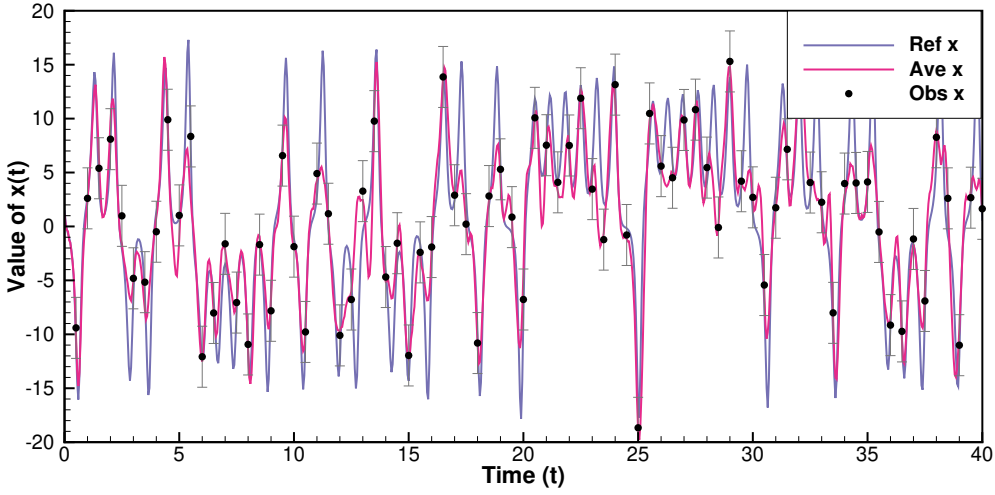
EnKF with the Lorenz model: error estimate



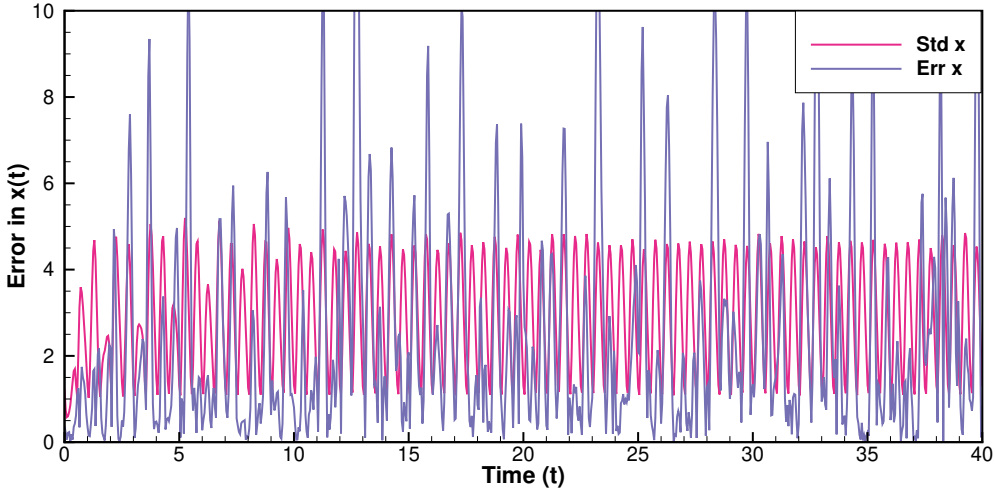
General smoother formulation



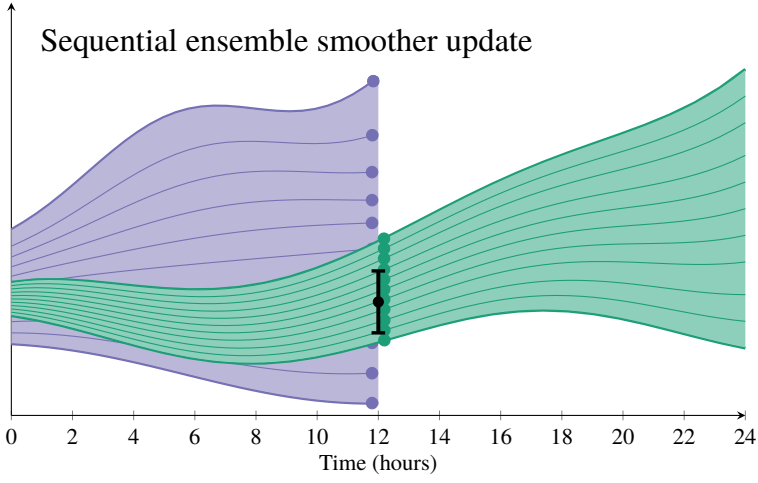
ES with the Lorenz model: estimate



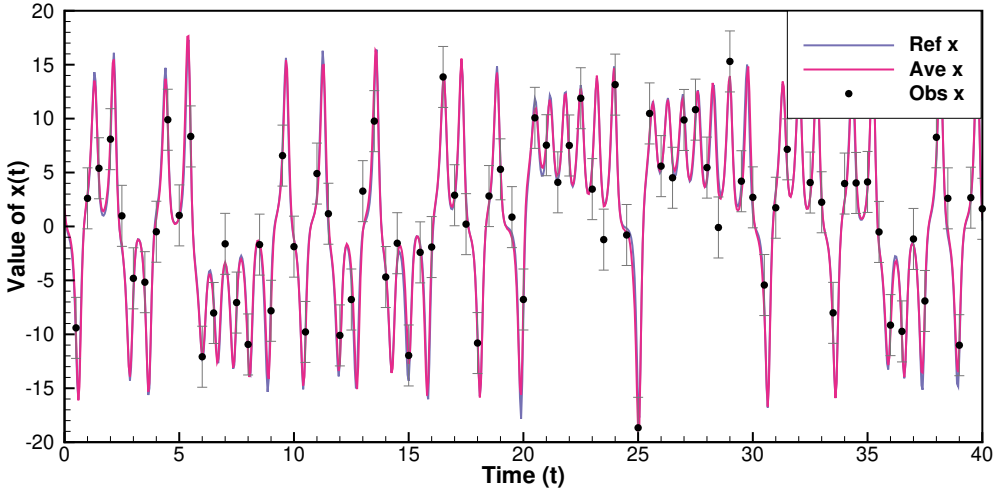
ES with the Lorenz model: error estimate



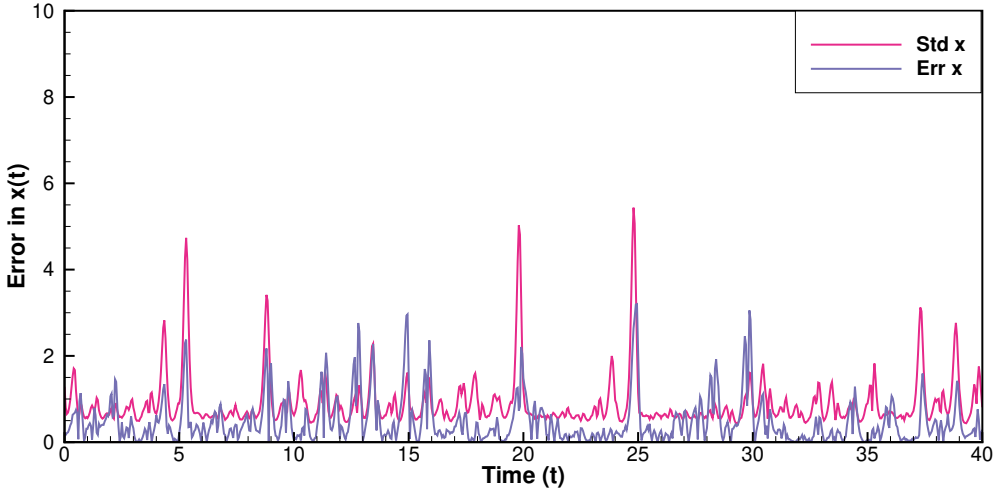
Recursive smoother formulation



EnKS with the Lorenz model: estimate



EnKS with the Lorenz model: error estimate



EnKF algorithms

Subspace EnKF update

1: subroutine EnKF_update$\left(\mathbf{Z}, \mathbf{D}, \boldsymbol{\Upsilon}\right)$
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: $\mathbf{Z} \leftarrow \mathbf{Z}\left(\mathbf{I} + \mathbf{W}\Big/\sqrt{N-1}\right)$ ▷ 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**

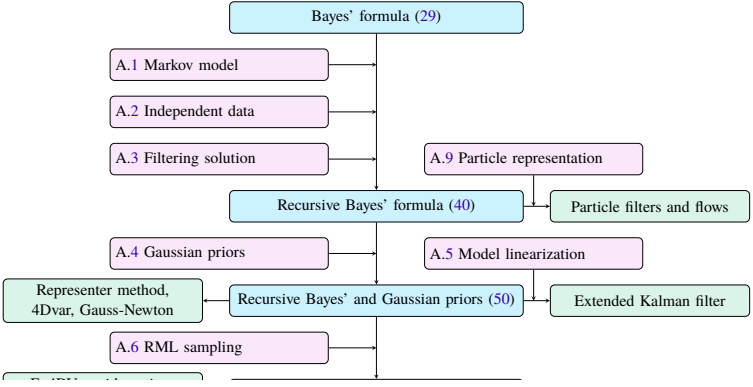
ES updating $\mathbf{X}$

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**

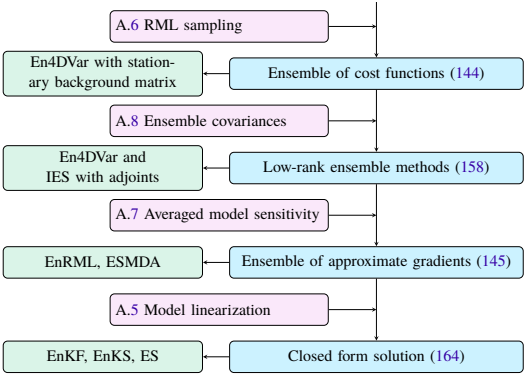
EnKS updating $[\mathbf{X}_l, \ldots, \mathbf{X}_0]$

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

$$f(\mathbf{z}) \approx \sum_{j=1}^{N} \frac{1}{N} \delta(\mathbf{z} - \mathbf{z}_j), \tag{171}$$

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

Comparison of methods on a scalar model

Scalar models

$$y = g(x, q) = x + 0.3x^3 + q, \tag{172}$$

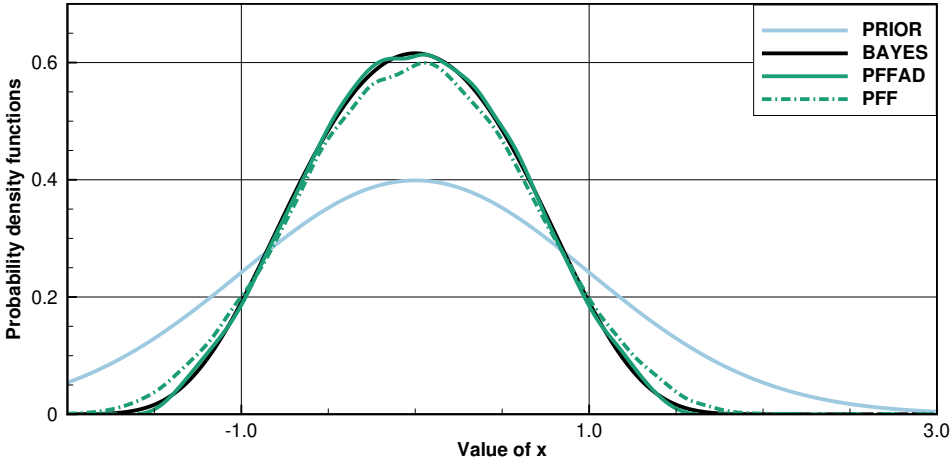
$$y = g(x, q) = 1 + \sin(x) + q, \tag{173}$$

Scalar examples

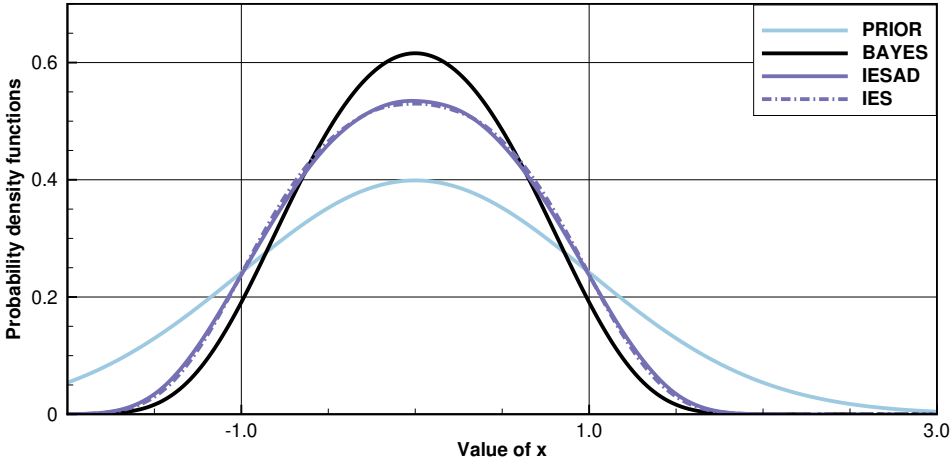
1. Model Eq. (172), $x_j^{\mathrm{f}} = \mathcal{N}(x^{\mathrm{f}} = 0.0, C_{xx} = 1.0)$, $d_j = \mathcal{N}(0.0, 1.0)$, and $q = \mathcal{N}(0, C_{qq} = 0.25)$.
2. Model Eq. (172), $x_j^{\mathrm{f}} = \mathcal{N}(x^{\mathrm{f}} = 1.0, C_{xx} = 1.0)$, $d_j = \mathcal{N}(0.0, -1.0)$, and $q = \mathcal{N}(0, C_{qq} = 0.25)$.
3. Model Eq. (173), $x_j^{\mathrm{f}} = \mathcal{N}(x^{\mathrm{f}} = 1.0, C_{xx} = 1.0)$, $d_j = \mathcal{N}(0.0, 1.0)$, and $q = \mathcal{N}(0, C_{qq} = 0.25)$.

Ensemble size $N = 10^7$.

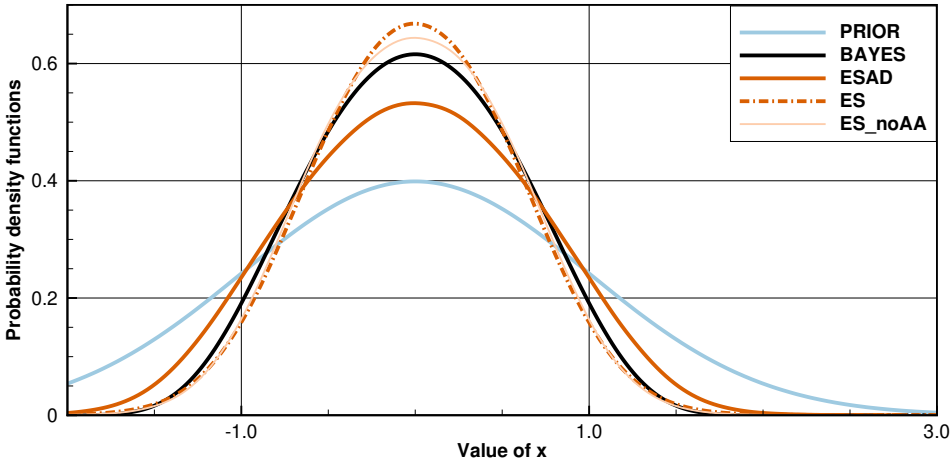
Case 1: Particle flow



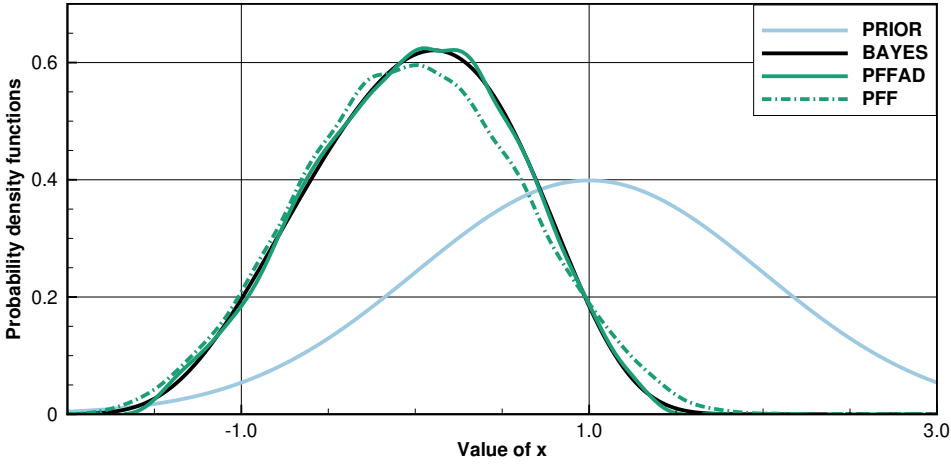
Case 1: IES


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

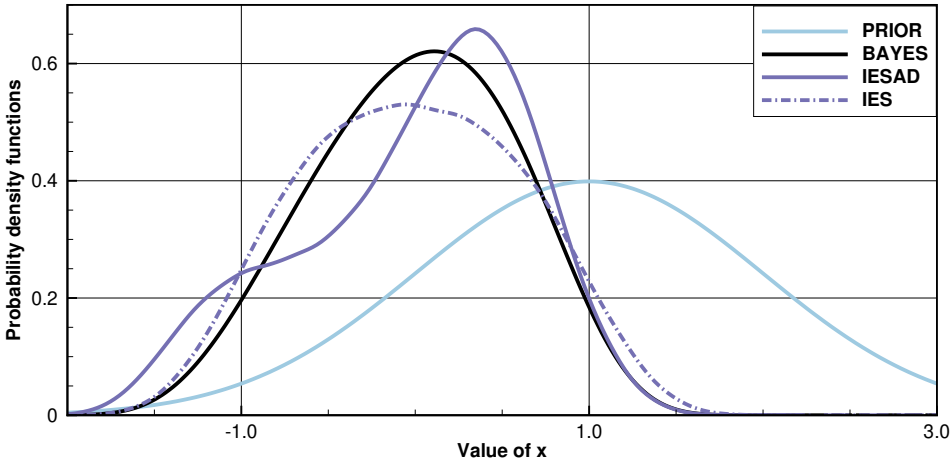
Case 1: EnKF


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

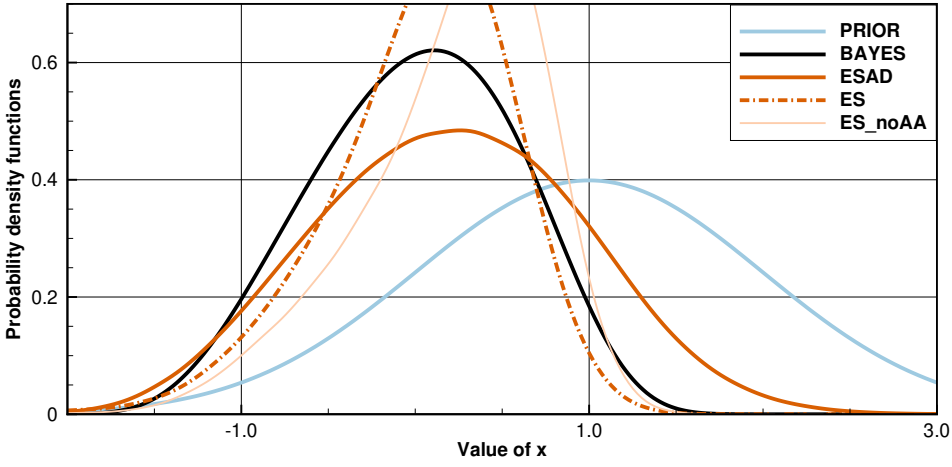
Case 2: Particle flow



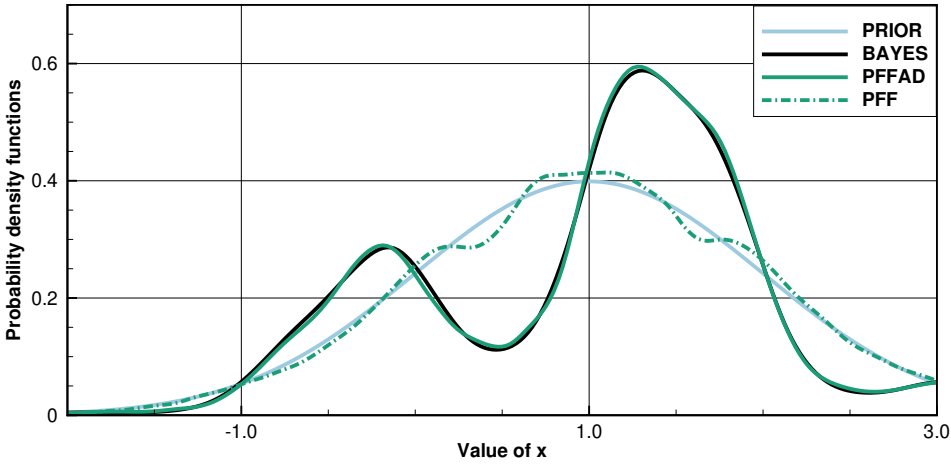
Case 2: IES


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

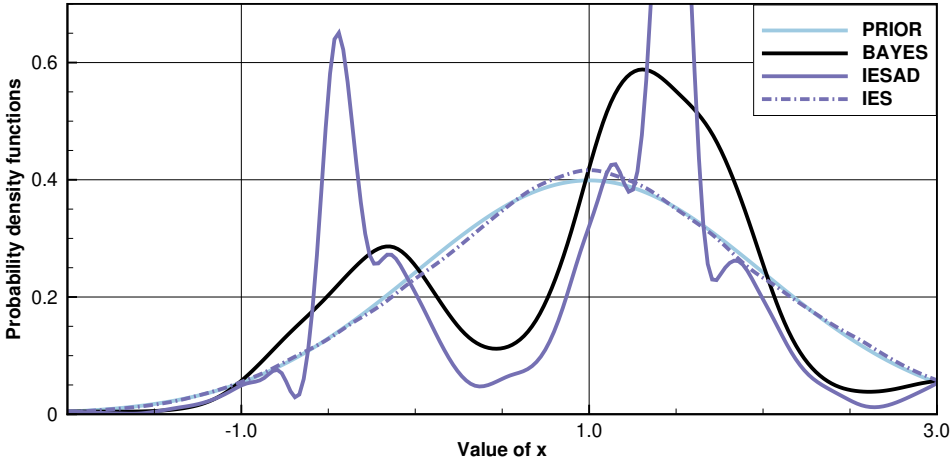
Case 2: EnKF


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

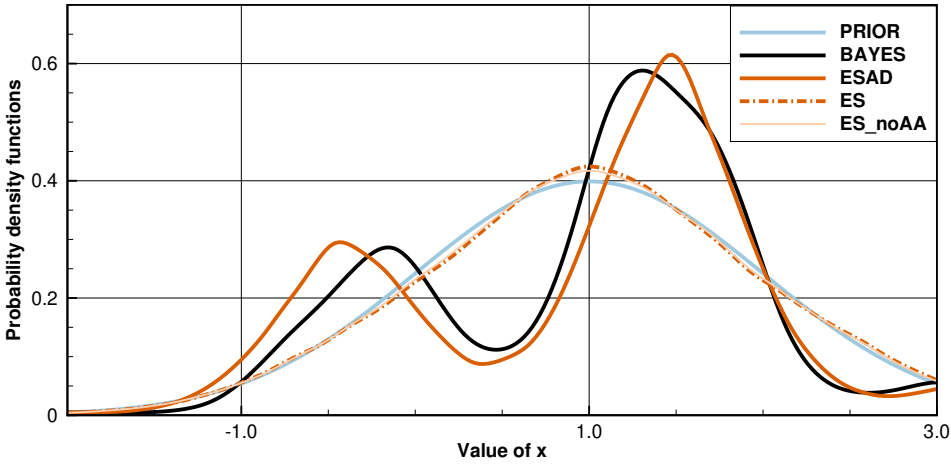
Case 3: Particle flow



Case 3: 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 3: 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

ESMDA with a SARS-COV-2 pandemic nmodel

Available observations

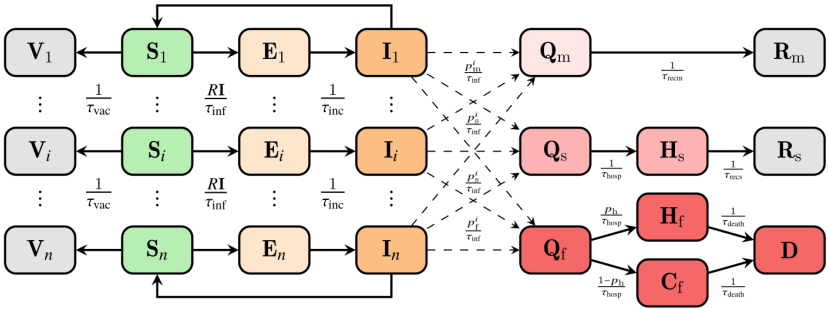
Hospitalized: Within different agegroups and gender.

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

$$\frac{\partial \mathbf{S}_i}{\partial t} = -\left(\sum_{j=1}^{n} \frac{R_{ij}(t)\mathbf{I}_j}{\tau_{\text{inf}}}\right)\mathbf{S}_i \tag{174}$$

$$\frac{\partial \mathbf{E}_i}{\partial t} = \left(\sum_{j=1}^{n} \frac{R_{ij}(t)\mathbf{I}_j}{\tau_{\text{inf}}}\right)\mathbf{S}_i - \frac{1}{\tau_{\text{inc}}}\,\mathbf{E}_i \tag{175}$$

$$\frac{\partial \mathbf{I}_i}{\partial t} = \frac{1}{\tau_{\text{inc}}}\,\mathbf{E}_i - \frac{1}{\tau_{\text{inf}}}\,\mathbf{I}_i \tag{176}$$

$$\frac{\partial \mathbf{Q}_{\text{m}}}{\partial t} = \sum_{i=1}^{n} \frac{p_{\text{m}}^i}{\tau_{\text{inf}}}\,\mathbf{I}_i - (1/\tau_{\text{recm}})\,\mathbf{Q}_{\text{m}} \tag{177}$$

$$\frac{\partial \mathbf{Q}_{\text{s}}}{\partial t} = \sum_{i=1}^{n} \frac{p_{\text{s}}^i}{\tau_{\text{inf}}}\,\mathbf{I}_i - (1/\tau_{\text{hosp}})\,\mathbf{Q}_{\text{s}} \tag{178}$$

$$\frac{\partial \mathbf{Q}_{\text{f}}}{\partial t} = \sum_{i=1}^{n} \frac{p_{\text{f}}^i}{\tau_{\text{inf}}}\,\mathbf{I}_i - (1/\tau_{\text{hosp}})\,\mathbf{Q}_{\text{f}} \tag{179}$$

$$\frac{\partial \mathbf{H}_{\text{s}}}{\partial t} = \frac{1}{\tau_{\text{hosp}}}\,\mathbf{Q}_{\text{s}} - \frac{1}{\tau_{\text{recs}}}\,\mathbf{H}_{\text{s}} \tag{180}$$

$$\frac{\partial \mathbf{H}_{\text{f}}}{\partial t} = \frac{p_{\text{h}}}{\tau_{\text{hosp}}}\,\mathbf{Q}_{\text{f}} - \frac{1}{\tau_{\text{death}}}\,\mathbf{H}_{\text{f}} \tag{181}$$

$$\frac{\partial \mathbf{C}_{\text{f}}}{\partial t} = \frac{(1-p_{\text{h}})}{\tau_{\text{hosp}}}\,\mathbf{Q}_{\text{f}} - \frac{1}{\tau_{\text{death}}}\,\mathbf{C}_{\text{f}} \tag{182}$$

$$\frac{\partial \mathbf{R}_{\text{m}}}{\partial t} = \frac{1}{\tau_{\text{recm}}}\,\mathbf{Q}_{\text{m}} \tag{183}$$

$$\frac{\partial \mathbf{R}_{\text{s}}}{\partial t} = \frac{1}{\tau_{\text{recs}}}\,\mathbf{H}_{\text{s}} \tag{184}$$

$$\frac{\partial \mathbf{D}}{\partial t} = \frac{1}{\tau_{\text{death}}}\,\mathbf{H}_{\text{f}} + \frac{1}{\tau_{\text{death}}}\mathbf{C}_{\text{f}} \tag{185}$$

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

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

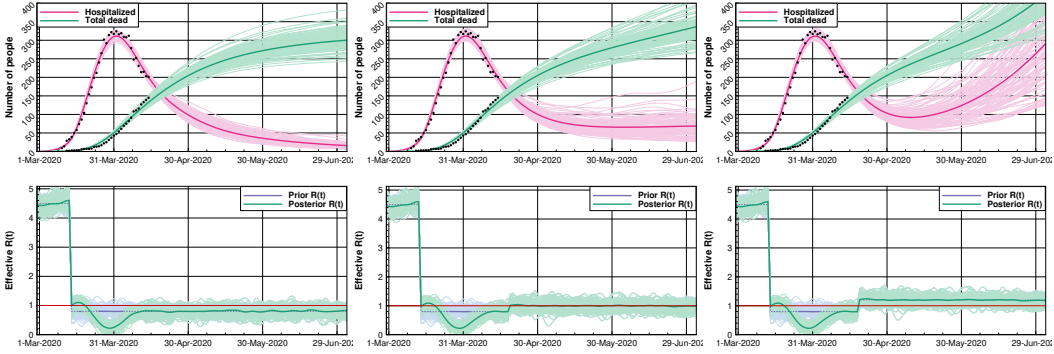
$\mathbf{R}(t)$ is a function of time (steered by how people isolate or interact).

- $R(t)$ 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

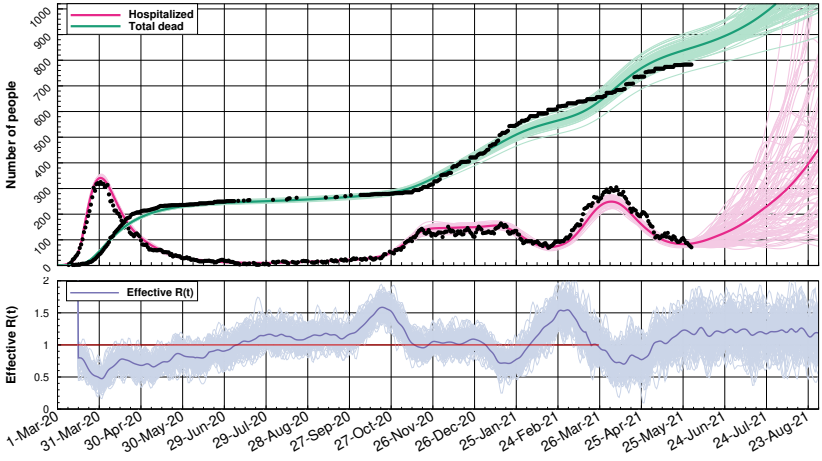
- Simple implementation and use.
- Efficient for large ensemble sizes.
- 5000 realizations and 32 ESMDA steps.

Back-to-school scenarios for Norway



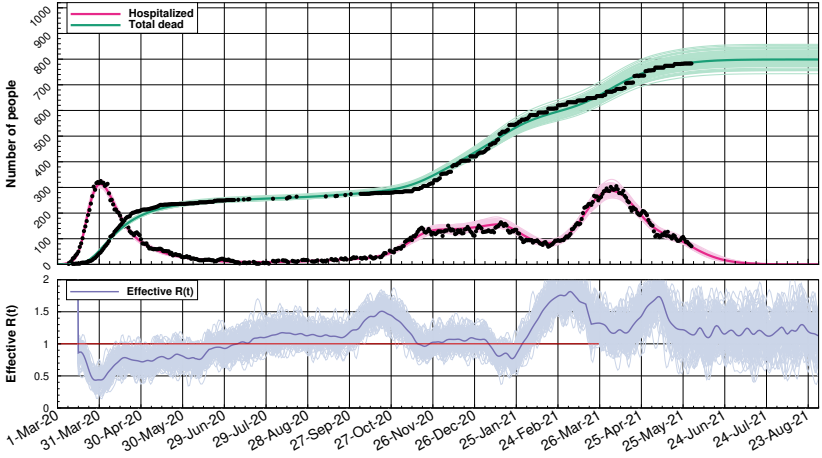
Available from: https://github.com/geirev/Data-Assimilation-Fundamentals

Norway: prediction without vaccinations



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

Norway: prediction with vaccinations



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

Summary EnKF_seir

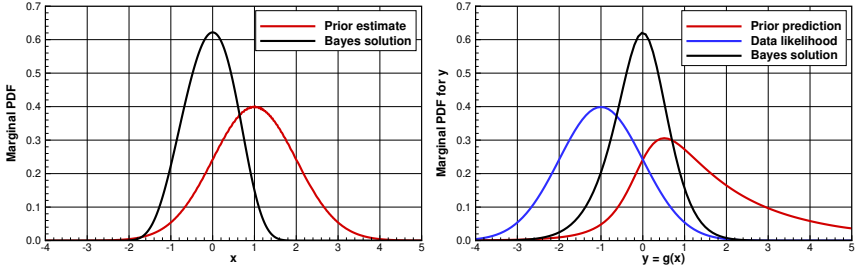
- The DA system tracks the epidemic accurately.
- We can estimate past $R(t)$.
- It is possible to quantify the impact of interventions.
- Short-term forecasting using R-persistence works well.
- Long-term scenario forecasting with specified future R.
- Code: https://github.com/geirev/EnKF_seir
- Paper: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

EnRML for History Matching Petroleum Models

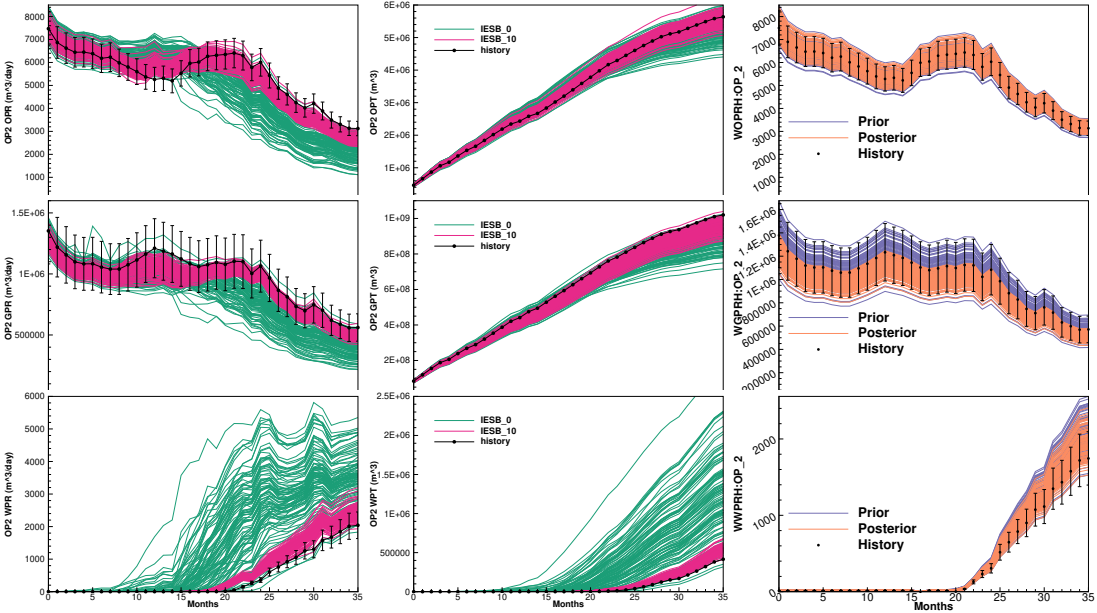
Illustration of the parameter estimation problem

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

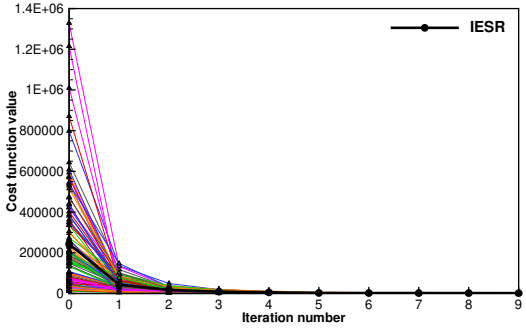
$$f(\mathbf{x}|\mathbf{d}) \propto f(\mathbf{d} \mid \mathbf{g}(\mathbf{x}))\, f(\mathbf{x})$$

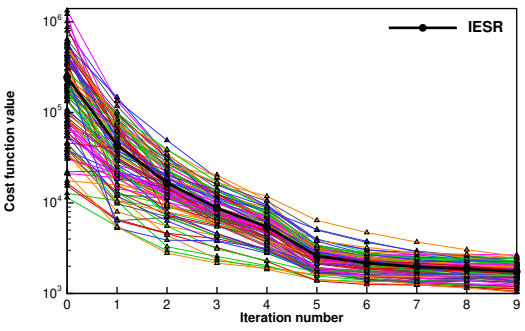


- $\mathbf{x}$ represents model input parameters like, porosity, permeability, fault multipliers
- $\mathbf{y}$ could represent predicted production of oil, gas, and water
- Prior pdf represents uncertainty of $\mathbf{x}$.
- Prior prediction pdf represents uncertainty of $\mathbf{y} = g(\mathbf{x})$.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


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

Ensemble of cost functions





Summary

Presentation follows new text book on data assimilation:

- Top-down approach for deriving the most popular methods from Bayes'.
- We introduced the important assumptions and applied approximations.
- We illustrated ensemble methods on simple problems.
- 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.