

# Data-Assimilation Fundamentals: Using iterative methods to handle nonlinearity

Geir Evensen



Seminar at DIFA of UNIBO Apr. 7th and May 19th, 2022

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

## 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.

Evensen · Vossepoel · Leeuwen



Data Assimilation Fundamentals

Except where otherwise noted, this book is licensed under a Creative Commons Attribution 4.0 International License. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>.



ISBN 978-3-030-96708-6



► [springer.com](http://springer.com)



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

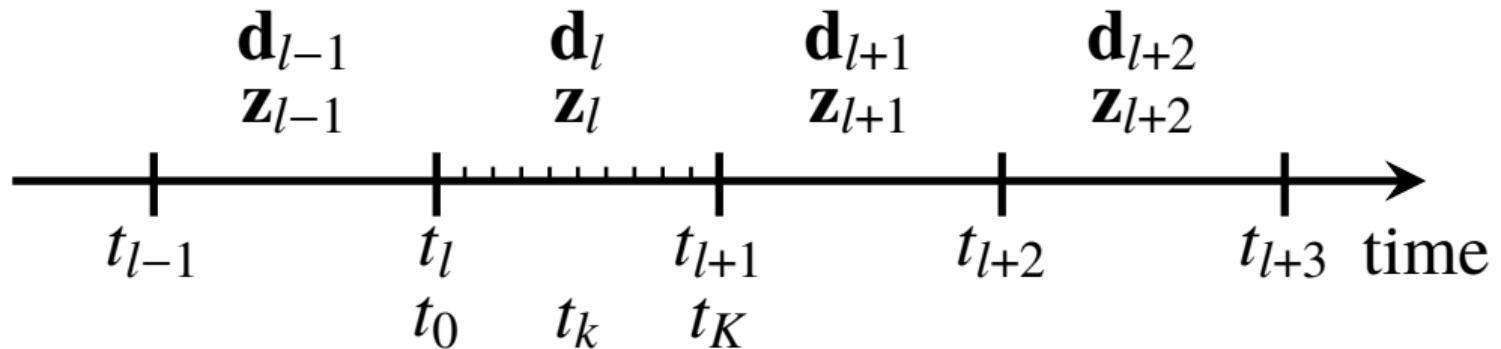
Springer

We start from Bayes' theorem

$$f(\mathcal{Z}|\mathcal{D}) = \frac{f(\mathcal{D}|\mathcal{Z})f(\mathcal{Z})}{f(\mathcal{D})}. \quad (1)$$

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

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

## Independent measurements

### Approximation 2 (Independent measurements)

*We assume that measurements are independent between different assimilation windows.*

Independent measurements have uncorrelated errors

$$f(\mathcal{D}|\mathcal{Z}) = \prod_{l=1}^L f(\mathbf{d}_l|\mathbf{z}_l). \quad (3)$$

## 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.*

## Discrete model with uncertain inputs

$$\mathbf{x}_k = \mathbf{m}(\mathbf{x}_{k-1}, \boldsymbol{\theta}, \mathbf{u}_k, \mathbf{q}_k). \quad (4)$$

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

## 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.

$$\mathbf{y} = \mathbf{g}(\mathbf{z}) = \mathbf{h}(\mathbf{m}(\mathbf{x}_0)). \quad (5)$$

- 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  $\mathbf{x}_K$ .

$$\mathbf{y} = \mathbf{g}(\mathbf{z}) = \mathbf{h}(\mathbf{x}_K). \quad (6)$$

- EnKF.

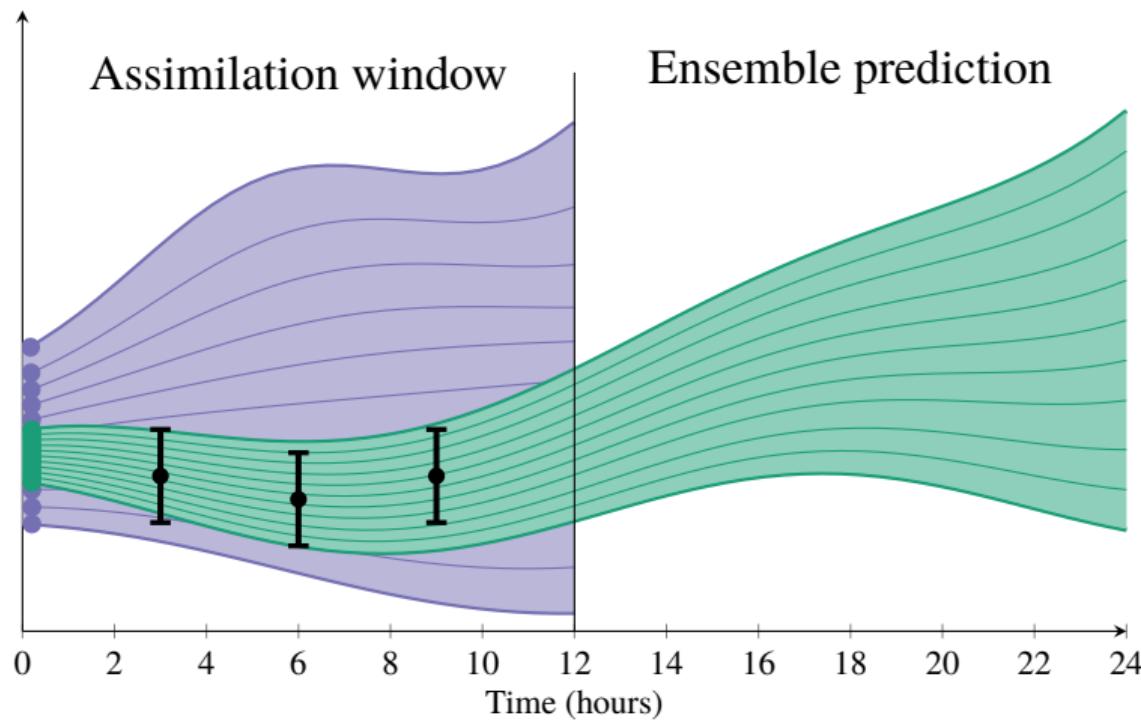
## Parameter estimation

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

$$\mathbf{y} = \mathbf{g}(\mathbf{z}) = \mathbf{h}(\mathbf{m}(\boldsymbol{\theta})). \quad (7)$$

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

## Smoothen for perfect models



# Deriving the marginal posterior pdf

Nonlinear “perfect” model and measurements

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

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

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

Marginal pdf

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

## Bayes' theorem related to the predicted measurements

We introduce nonlinearity through the likelihood

$$f(\mathbf{z}|\mathbf{d}) = \frac{f(\mathbf{d}|\mathbf{g}(\mathbf{z}))f(\mathbf{z})}{f(\mathbf{d})}. \quad (8)$$

## 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.*

$$f(\mathbf{z}|\mathbf{d}) \propto \exp\{-\mathcal{J}(\mathbf{z})\}, \quad (9)$$

Leads to a cost-function formulation for the MAP solution

### Cost function

$$\mathcal{J}(\mathbf{z}) = \frac{1}{2} (\mathbf{z} - \mathbf{z}^f)^T \mathbf{C}_{zz}^{-1} (\mathbf{z} - \mathbf{z}^f) + \frac{1}{2} (\mathbf{g}(\mathbf{z}) - \mathbf{d})^T \mathbf{C}_{dd}^{-1} (\mathbf{g}(\mathbf{z}) - \mathbf{d}). \quad (10)$$

### The gradient set to zero

$$\mathbf{C}_{zz}^{-1} (\mathbf{z}^a - \mathbf{z}^f) + \nabla_{\mathbf{z}} \mathbf{g}(\mathbf{z}^a) \mathbf{C}_{dd}^{-1} (\mathbf{g}(\mathbf{z}^a) - \mathbf{d}) = 0. \quad (11)$$

- There is no explicit solution of the gradient equation.

# Gauss-Newton methods solves for the MAP estimate

## Gauss-Newton iteration

$$\mathbf{z}^{i+1} = \mathbf{z}^i - \gamma^i \left( \mathbf{C}_{zz}^{-1} + \mathbf{G}^{i\top} \mathbf{C}_{dd}^{-1} \mathbf{G}^i \right)^{-1} \left( \mathbf{C}_{zz}^{-1} (\mathbf{z}^i - \mathbf{z}^f) + \mathbf{G}^{i\top} \mathbf{C}_{dd}^{-1} (\mathbf{g}(\mathbf{z}^i) - \mathbf{d}) \right). \quad (12)$$

- The incremental formulation is sometimes more convenient.

# Incremental Gauss-Newton methods

## Quadratic cost function for the increments

$$\mathcal{J}(\delta\mathbf{z}) = \frac{1}{2}(\delta\mathbf{z} - \boldsymbol{\xi}^i)^T \mathbf{C}_{zz}^{-1} (\delta\mathbf{z} - \boldsymbol{\xi}^i) + \frac{1}{2}(\mathbf{G}^i \delta\mathbf{z} - \boldsymbol{\eta}^i)^T \mathbf{C}_{dd}^{-1} (\mathbf{G}^i \delta\mathbf{z} - \boldsymbol{\eta}^i). \quad (13)$$

with

$$\mathbf{z}^{i+1} = \mathbf{z}^i + \delta\mathbf{z}, \quad (14)$$

$$\boldsymbol{\eta}^i = \mathbf{d} - \mathbf{g}(\mathbf{z}^i), \quad (15)$$

$$\boldsymbol{\xi}^i = \mathbf{z}^f - \mathbf{z}^i. \quad (16)$$

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

## Standard strong constraint 4DVar

## Incremental strong constraint 4DVar

# Incremental SC-4DVar

Nonlinear model

$$\mathbf{x}_0 = \mathbf{x}_0^f + \mathbf{x}'_0, \quad (17)$$

$$\mathbf{x}_{k+1} = \mathbf{m}(\mathbf{x}_k). \quad (18)$$

State variable

$$\mathbf{z} = \mathbf{x}_0 \quad (19)$$

We compute updates

$$\mathbf{z}^{i+1} = \mathbf{z}^i + \delta\mathbf{z}, \quad (20)$$

# Solving for the increments

## Inner incremental SC-4DVar costfunction

$$\mathcal{J}(\delta\mathbf{z}) = \frac{1}{2} (\delta\mathbf{z} - \boldsymbol{\xi}^i)^T \mathbf{C}_{zz}^{-1} (\delta\mathbf{z} - \boldsymbol{\xi}^i) + \frac{1}{2} (\mathbf{H}^i \delta\mathbf{x} - \boldsymbol{\eta}^i)^T \mathbf{C}_{dd}^{-1} (\mathbf{H}^i \delta\mathbf{x} - \boldsymbol{\eta}^i), \quad (21)$$

Linear model for the increments

$$\delta\mathbf{x}_0 = \delta\mathbf{z}^i, \quad (22)$$

$$\delta\mathbf{x}_{k+1} = \mathbf{M}_k^i \delta\mathbf{x}_k. \quad (23)$$

Prior increment

$$\boldsymbol{\xi}^i = \mathbf{x}_0^f - \mathbf{x}_0^i, \quad (24)$$

Innovation

$$\boldsymbol{\eta}^i = \mathbf{d} - \mathbf{h}(\mathbf{x}^i). \quad (25)$$

## Lagrangian formulation

$$\begin{aligned}\mathcal{L}(\delta\mathbf{z}, \delta\mathbf{x}, \delta\boldsymbol{\lambda}) = & \frac{1}{2} (\delta\mathbf{z} - \boldsymbol{\xi}^i)^T \mathbf{C}_{zz}^{-1} (\delta\mathbf{z} - \boldsymbol{\xi}^i) \\ & + \frac{1}{2} (\mathbf{H}^i \delta\mathbf{x} - \boldsymbol{\eta}^i)^T \mathbf{C}_{dd}^{-1} (\mathbf{H}^i \delta\mathbf{x} - \boldsymbol{\eta}^i) \\ & + \sum_{k=0}^K \delta\boldsymbol{\lambda}_{k+1}^T (\delta\mathbf{x}_{k+1} - \mathbf{M}_k^i \delta\mathbf{x}_k).\end{aligned}\tag{26}$$

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

# Gradient of Lagrangian

$$\nabla_{\delta \mathbf{x}_k} \mathcal{L}(\delta \mathbf{z}, \delta \mathbf{x}, \delta \boldsymbol{\lambda}) = \mathbf{H}_k^{i^T} \mathbf{C}_{dd}^{-1} (\mathbf{H}^i \delta \mathbf{x} - \boldsymbol{\eta}^i) + \delta \boldsymbol{\lambda}_k - \mathbf{M}_k^{i^T} \delta \boldsymbol{\lambda}_{k+1}, \quad (27)$$

$$\nabla_{\delta \mathbf{x}_K} \mathcal{L}(\delta \mathbf{z}, \delta \mathbf{x}, \delta \boldsymbol{\lambda}) = \delta \boldsymbol{\lambda}_{K+1}, \quad (28)$$

$$\begin{aligned} \nabla_{\delta \mathbf{z}} \mathcal{L}(\delta \mathbf{z}, \delta \mathbf{x}, \delta \boldsymbol{\lambda}) &= \mathbf{C}_{zz}^{-1} (\delta \mathbf{z} - \boldsymbol{\xi}^i) - \mathbf{M}_0^{i^T} \delta \boldsymbol{\lambda}_1 \\ &= \mathbf{C}_{zz}^{-1} (\delta \mathbf{z} - \boldsymbol{\xi}^i) - \delta \boldsymbol{\lambda}_0, \end{aligned} \quad (29)$$

$$\nabla_{\delta \boldsymbol{\lambda}_k} \mathcal{L}(\delta \mathbf{z}, \delta \mathbf{x}, \delta \boldsymbol{\lambda}) = \delta \mathbf{x}_{k+1} - \mathbf{M}_k^i \delta \mathbf{x}_k. \quad (30)$$

# Euler-Lagrange equation(s)

Forward model

$$\delta \mathbf{x}_0 = \xi^i + \mathbf{C}_{zz} \delta \lambda_0, \quad (31)$$

$$\delta \mathbf{x}_{k+1} - \mathbf{M}_k^i \delta \mathbf{x}_k = 0, \quad (32)$$

Adjoint model

$$\delta \lambda_{K+1} = 0, \quad (33)$$

$$\delta \lambda_k - \mathbf{M}_k^{i^T} \delta \lambda_{k+1} = \mathbf{H}_k^{i^T} \mathbf{C}_{dd}^{-1} (\mathbf{H}^i \delta \mathbf{x} - \boldsymbol{\eta}^i). \quad (34)$$

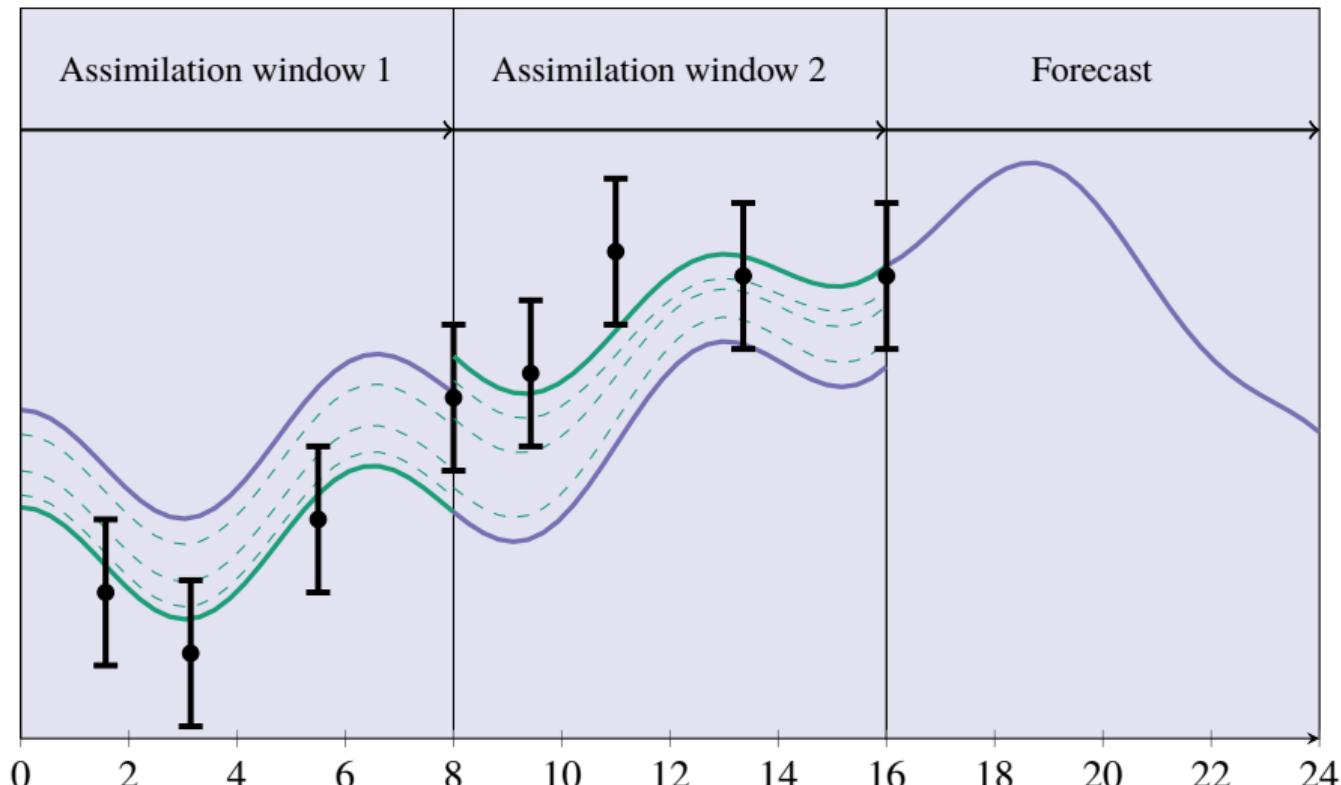
# Incremental 4DVar algorithm

```

1: Input:  $\mathbf{x}_0^f \in \mathbb{R}^n$ ;  $\mathbf{d} \in \mathbb{R}^m$ ;  $\mathbf{C}_{dd} \in \mathbb{R}^{m \times m}$ ;  $\mathbf{C}_{zz} \in \mathbb{R}^{n \times n}$                                 ▷ Prior inputs
2:  $\mathbf{x}_0^0 = \mathbf{x}_0^f$                                                                ▷ Initialization of  $\mathbf{x}_0$ 
3:  $\delta\mathbf{z} = 0$                                                                     ▷ Initialization of  $\delta\mathbf{z}$ 
4:  $i = 1$ 
5: repeat                                                                           ▷ Iteration loop
6:    $\mathbf{x}_0^i = \mathbf{x}_0^{i-1} + \delta\mathbf{z}$                                          ▷ Update initial condition
7:    $\xi^i = \mathbf{x}_0^f - \mathbf{x}_0^i$                                                  ▷ Current increment
8:   for  $k = 0 : K$  do                                                       ▷ Integrate forward model
9:      $\mathbf{x}_{k+1}^i = \mathbf{m}(\mathbf{x}_k^i)$                                          ▷ Nonlinear model prediction
10:    end for
11:     $\eta^i = \mathbf{d} - \mathbf{h}(\mathbf{x}^i)$                                          ▷ Current innovation vector
12:    repeat                                                                           ▷ Iteration loop
13:       $\delta\mathbf{x}_0 = \delta\mathbf{z}$ 
14:      for  $k = 0 : K$  do                                                       ▷ Integrate forward model
15:         $\delta\mathbf{x}_{k+1} = \mathbf{M}_k^i \delta\mathbf{x}_k$ 
16:      end for
17:       $\delta\lambda_{K+1} = 0$ 
18:      for  $k = K : 0$  do                                                       ▷ Integrate backward adjoint model
19:         $\delta\lambda_k = \mathbf{M}_k^{i^T} \delta\lambda_{k+1} - \mathbf{H}_k^{i^T} \mathbf{C}_{dd}^{-1} (\mathbf{H}^i \delta\mathbf{x} - \eta^i)$ 
20:      end for
21:       $\delta\mathbf{z} \leftarrow \delta\mathbf{z} - \gamma \mathbf{B} (\delta\mathbf{z} - \xi^i - \mathbf{C}_{zz} \delta\lambda_0)$  ▷ Update  $\delta\mathbf{z}$  using gradient in Eq. (29)
22:    until convergence
23:     $i = i + 1$ 
24:  until convergence

```

# Incremental 4DVar algorithm



## 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-constraint solution that assumes the model is perfect.

## Weak constraint 4DVar

## WC-4DVar: State-space formulation

Model with additive errors

$$\mathbf{x}_k = \mathbf{m}(\mathbf{x}_{k-1}) + \mathbf{q}_k. \quad (35)$$

Define state vector and model error covariance

$$\mathbf{z} = \mathbf{x} = \begin{pmatrix} \mathbf{x}_0 \\ \vdots \\ \mathbf{x}_K \end{pmatrix}, \quad \text{and} \quad \mathbf{C}_{qq} = \begin{pmatrix} \mathbf{C}_{q_1 q_1} & \cdots & \mathbf{C}_{q_1 q_K} \\ \vdots & \ddots & \vdots \\ \mathbf{C}_{q_K q_1} & \cdots & \mathbf{C}_{q_K q_K} \end{pmatrix}, \quad (36)$$

## WC-4DVar cost function

### The weak-constraint cost function (generalized inverse formulation)

$$\begin{aligned}\mathcal{J}(\mathbf{x}) = & \frac{1}{2} (\mathbf{x}_0 - \mathbf{x}_0^f)^T \mathbf{C}_{x_0 x_0}^{-1} (\mathbf{x}_0 - \mathbf{x}_0^f) \\ & + \frac{1}{2} (\mathbf{h}(\mathbf{x}) - \mathbf{d})^T \mathbf{C}_{dd}^{-1} (\mathbf{h}(\mathbf{x}) - \mathbf{d}) \\ & + \frac{1}{2} \sum_{r=1}^K \sum_{s=1}^K (\mathbf{x}_r - \mathbf{m}(\mathbf{x}_{r-1}))^T \mathbf{C}_{qq}^{-1}(r, s) (\mathbf{x}_s - \mathbf{m}(\mathbf{x}_{s-1})).\end{aligned}\tag{37}$$

## WC-4DVar: incremental form

Define update

$$\mathbf{x}^{i+1} = \mathbf{x}^i + \delta\mathbf{x}. \quad (38)$$

Linearized model

$$\begin{aligned} \mathbf{m}(\mathbf{x}_k^{i+1}) &= \mathbf{m}(\mathbf{x}_k^i + \delta\mathbf{x}_k) \\ &\approx \mathbf{m}(\mathbf{x}_k^i) + \mathbf{M}_k \delta\mathbf{x}_k, \end{aligned} \quad (39)$$

Linearized measurement operator

$$\begin{aligned} \mathbf{h}(\mathbf{x}^{i+1}) &= \mathbf{h}(\mathbf{x}^i + \delta\mathbf{x}) \\ &\approx \mathbf{h}(\mathbf{x}^i) + \mathbf{H} \delta\mathbf{x}, \end{aligned} \quad (40)$$

## WC-4DVar: incremental form

Write the model residual in Eq. (37) as

$$\begin{aligned}
 \mathbf{x}_k^{i+1} - \mathbf{m}(\mathbf{x}_{k-1}^{i+1}) &\approx \mathbf{x}_k^{i+1} - \mathbf{m}(\mathbf{x}_{k-1}^i) - \mathbf{M}_{k-1}\delta\mathbf{x}_{k-1} \\
 &= \mathbf{x}_k^{i+1} - \mathbf{x}_k^i + \mathbf{x}_k^i - \mathbf{m}(\mathbf{x}_{k-1}^i) - \mathbf{M}_{k-1}\delta\mathbf{x}_{k-1} \\
 &= \delta\mathbf{x}_k - \mathbf{M}_{k-1}\delta\mathbf{x}_{k-1} + \boldsymbol{\xi}_k^i,
 \end{aligned} \tag{41}$$

where

$$\boldsymbol{\xi}_k^i = \mathbf{x}_k^i - \mathbf{m}(\mathbf{x}_{k-1}^i). \tag{42}$$

Innovations becomes

$$\boldsymbol{\eta}^i = \mathbf{d} - \mathbf{h}(\mathbf{x}^i). \tag{43}$$

## WC incremental cost function

### The weak-constraint incremental cost function

$$\begin{aligned}\mathcal{J}(\delta\mathbf{x}) = & \frac{1}{2} (\delta\mathbf{x}_0 - \boldsymbol{\xi}_0^i)^T \mathbf{C}_{x_0 x_0}^{-1} (\delta\mathbf{x}_0 - \boldsymbol{\xi}_0^i) \\ & + \frac{1}{2} (\mathbf{H}^i \delta\mathbf{x} - \boldsymbol{\eta}^i)^T \mathbf{C}_{dd}^{-1} (\mathbf{H}^i \delta\mathbf{x} - \boldsymbol{\eta}^i) \\ & + \frac{1}{2} \sum_{r=1}^K \sum_{s=1}^K (\delta\mathbf{x}_r - \mathbf{M}_{r-1} \delta\mathbf{x}_{r-1} + \boldsymbol{\xi}_r^i)^T \mathbf{C}_{qq}^{-1}(r, s) (\delta\mathbf{x}_s - \mathbf{M}_{s-1} \delta\mathbf{x}_{s-1} + \boldsymbol{\xi}_s^i).\end{aligned}\tag{44}$$

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

$$\begin{aligned}\nabla_{\delta\mathbf{x}_k} \mathcal{J}(\delta\mathbf{x}) &= \mathbf{H}_k^{i^T} \mathbf{C}_{dd}^{-1} (\mathbf{H}^i \delta\mathbf{x} - \boldsymbol{\eta}^i) \\ &\quad + \sum_{s=1}^K \mathbf{C}_{qq}^{-1}(k, s) (\delta\mathbf{x}_s - \mathbf{M}_{s-1}^i \delta\mathbf{x}_{s-1} + \boldsymbol{\xi}_s^i) \\ &\quad - \mathbf{M}_k^{i^T} \sum_{s=1}^K \mathbf{C}_{qq}^{-1}(k+1, s) (\delta\mathbf{x}_s - \mathbf{M}_{s-1}^i \delta\mathbf{x}_{s-1} + \boldsymbol{\xi}_s^i).\end{aligned}\tag{45}$$

Define an adjoint vector for each time step as

$$\delta\lambda_k = \sum_{s=1}^K \mathbf{C}_{qq}^{-1}(k, s) (\delta\mathbf{x}_s - \mathbf{M}_{s-1}^i \delta\mathbf{x}_{s-1} + \boldsymbol{\xi}_s^i),\tag{46}$$

and write Eq. (45) as

$$\nabla_{\delta\mathbf{x}_k} \mathcal{J}(\delta\mathbf{x}) = \mathbf{H}_k^{i^T} \mathbf{C}_{dd}^{-1} (\mathbf{H}^i \delta\mathbf{x} - \boldsymbol{\eta}^i) + \delta\lambda_k - \mathbf{M}_k^{i^T} \delta\lambda_{k+1}.\tag{47}$$

# Euler-Lagrange equation(s)

## Forward model

$$\delta \mathbf{x}_0 = \boldsymbol{\xi}_0^i + \mathbf{C}_{x_0 x_0} \delta \boldsymbol{\lambda}_0, \quad (48)$$

$$\delta \mathbf{x}_k - \mathbf{M}_{k-1}^i \delta \mathbf{x}_{k-1} = -\boldsymbol{\xi}_k^i + \sum_{s=1}^K \mathbf{C}_{qq}(k, s) \delta \boldsymbol{\lambda}_s, \quad (49)$$

## Backward model

$$\delta \boldsymbol{\lambda}_{K+1} = \mathbf{0}, \quad (50)$$

$$\delta \boldsymbol{\lambda}_k - \mathbf{M}_k^{i^T} \delta \boldsymbol{\lambda}_{k+1} = -\mathbf{H}_k^{i^T} \mathbf{C}_{dd}^{-1} (\mathbf{H}^i \delta \mathbf{x} - \boldsymbol{\eta}^i). \quad (51)$$

## Reresenter method

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

$$\delta\mathbf{x} = \delta\mathbf{x}^f + \sum_{p=1}^m b_p \mathbf{r}_p = \delta\mathbf{x}^f + \mathbf{R}\mathbf{b}, \quad (52)$$

$$\delta\lambda = \delta\lambda^f + \sum_{p=1}^m b_p \mathbf{s}_p = \delta\lambda^f + \mathbf{S}\mathbf{b}. \quad (53)$$

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

$$\delta \mathbf{x}_0^f = \boldsymbol{\xi}_0^i, \quad (54)$$

$$\delta \mathbf{x}_k^f - \mathbf{M}_{k-1}^i \delta \mathbf{x}_{k-1}^f = -\boldsymbol{\xi}_k^i, \quad (55)$$

$$\delta \boldsymbol{\lambda}_{K+1}^f = 0, \quad (56)$$

$$\delta \boldsymbol{\lambda}_k^f - \mathbf{M}_k^{i^T} \delta \boldsymbol{\lambda}_{k+1}^f = 0. \quad (57)$$

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

## Derivation for the representers

$$\mathbf{R}_0 \mathbf{b} = \mathbf{C}_{x_0 x_0} \mathbf{M}_0^{i^T} \mathbf{S}_1 \mathbf{b}, \quad (58)$$

$$(\mathbf{R}_k \mathbf{b} - \mathbf{M}_{k-1}^i \mathbf{R}_{k-1} \mathbf{b}) = \sum_{s=1}^K \mathbf{C}_{qq}(k, s) \mathbf{S}_s \mathbf{b}, \quad (59)$$

$$\mathbf{S}_{K+1} \mathbf{b} = \mathbf{0}, \quad (60)$$

$$\mathbf{S}_k \mathbf{b} - \mathbf{M}_k^{i^T} \mathbf{S}_{k+1} \mathbf{b} = \mathbf{H}_k^{i^T} \mathbf{C}_{dd}^{-1} (\boldsymbol{\eta}^i - \mathbf{H}^i (\delta \mathbf{x}^f + \mathbf{R} \mathbf{b})). \quad (61)$$

Define  $\mathbf{b}$  as

$$\mathbf{b} = \mathbf{C}_{dd}^{-1} (\boldsymbol{\eta}^i - \mathbf{H}^i (\delta \mathbf{x}^f + \mathbf{R} \mathbf{b})), \quad (62)$$

such that Eq. (61) simplifies to

$$\mathbf{S}_k \mathbf{b} - \mathbf{M}_k^{i^T} \mathbf{S}_{k+1} \mathbf{b} = \mathbf{H}_k^{i^T} \mathbf{b}. \quad (63)$$

## Euler-Lagrange equations: Representers

$$\mathbf{R}_0 = \mathbf{C}_{x_0 x_0} \mathbf{M}_0^{i^T} \mathbf{S}_1, \quad (64)$$

$$\mathbf{R}_k - \mathbf{M}_{k-1}^i \mathbf{R}_{k-1} = \sum_{s=1}^K \mathbf{C}_{qq}(k, s) \mathbf{S}_s, \quad (65)$$

$$\mathbf{S}_{K+1} = \mathbf{0}, \quad (66)$$

$$\mathbf{S}_k - \mathbf{M}_k^{i^T} \mathbf{S}_{k+1} = \mathbf{H}_k^{i^T}. \quad (67)$$

Requires  $2m$  model integrations.

## Euler-Lagrange equations: $\mathbf{b}$

### Linear system for the representer coefficients $\mathbf{b}$

$$(\mathbf{H}^i \mathbf{R} + \mathbf{C}_{dd}) \mathbf{b} = \boldsymbol{\eta}^i - \mathbf{H}^i \delta \mathbf{x}^f. \quad (68)$$

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

## More efficient solution method

Given  $\mathbf{b}$  we obtain the final solution from (51)

### Adjoint equation forced by $\mathbf{b}$

$$\delta\lambda_k - \mathbf{M}_k^{i^T} \delta\lambda_{k+1} = \mathbf{H}_k^{i^T} \mathbf{b}. \quad (69)$$

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:

$$\mathbf{C}\mathbf{b} = \boldsymbol{\nu}. \quad (70)$$

Solving Eq. (70) is equivalent to minimizing the functional

$$\phi(\mathbf{b}) = \frac{1}{2} \mathbf{b}^T \mathbf{C} \mathbf{b} - \mathbf{b}^T \boldsymbol{\nu}, \quad (71)$$

which has a gradient

$$\nabla_{\mathbf{b}} \phi(\mathbf{b}) = \boldsymbol{\rho} = \mathbf{C}\mathbf{b} - \boldsymbol{\nu}. \quad (72)$$

Iterative solution

$$\mathbf{b}^{i+1} = \mathbf{b}^i - \gamma \boldsymbol{\rho}, \quad (73)$$

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 (58–61) as

$$\mathbf{c}_0 = \mathbf{C}_{x_0 x_0} \mathbf{M}_0^{i^T} \boldsymbol{\psi}_1, \quad (74)$$

$$\mathbf{c}_k = \mathbf{M}_{k-1}^i \mathbf{c}_{k-1} + \sum_{s=1}^K \mathbf{C}_{qq}(k, s) \boldsymbol{\psi}_s, \quad (75)$$

$$\boldsymbol{\psi}_{K+1} = \mathbf{0}, \quad (76)$$

$$\boldsymbol{\psi}_k = \mathbf{M}_k^{i^T} \boldsymbol{\psi}_{k+1} + \mathbf{H}_k^{i^T} \mathbf{v}. \quad (77)$$

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

$$\mathbf{H}\mathbf{c} = \mathbf{H}\mathbf{R}\mathbf{v} = \mathcal{R}\mathbf{v}, \quad (78)$$

## 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

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{k} \times \mathbf{u} = \frac{\partial}{\partial z} \left( A \frac{\partial \mathbf{u}}{\partial z} \right), \quad (79)$$

Initial condition

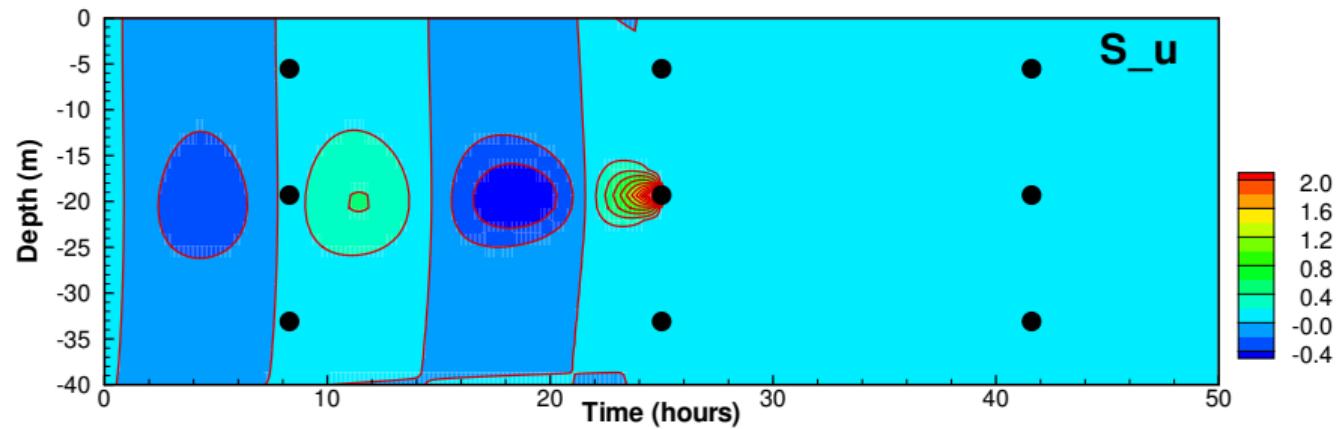
$$\mathbf{u}(z, 0) = \mathbf{u}_0, \quad (80)$$

boundary conditions

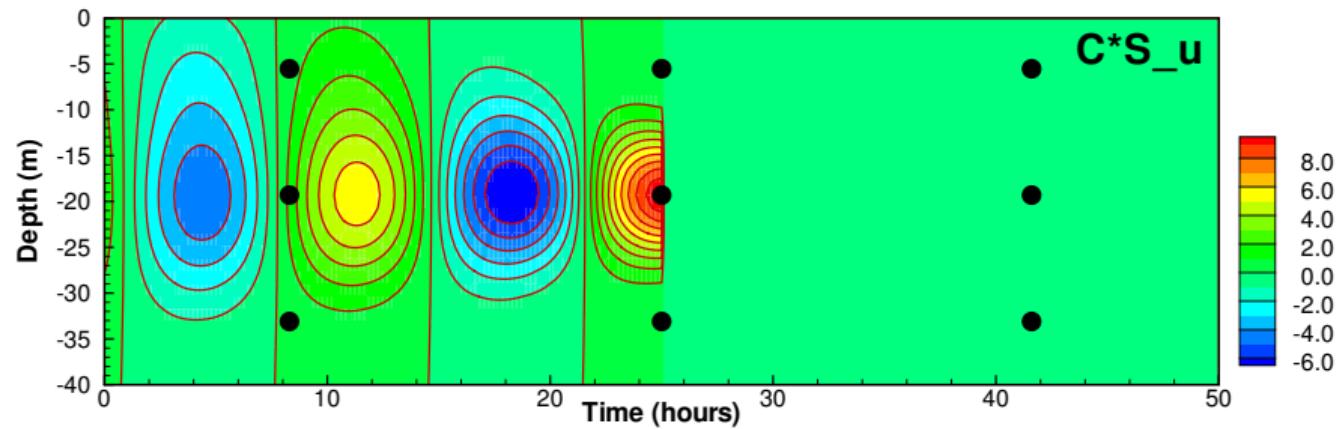
$$A \frac{\partial \mathbf{u}}{\partial z} \Big|_{z=0} = \left( c_d \sqrt{u_a^2 + v_a^2} \right) \mathbf{u}_a, \quad (81)$$

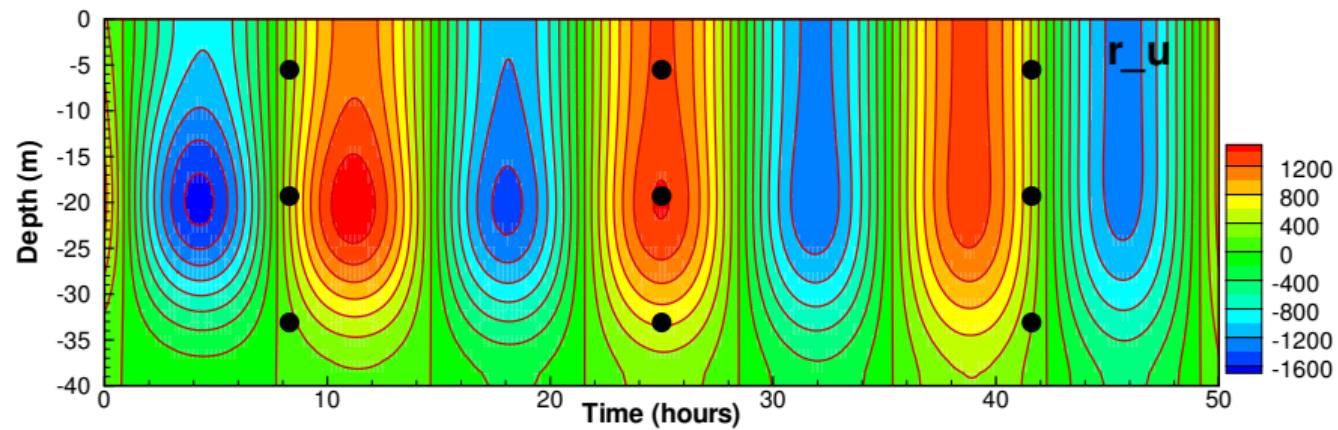
$$A \frac{\partial \mathbf{u}}{\partial z} \Big|_{z=-H} = \mathbf{0}, \quad (82)$$

# An adjoint representer $s$

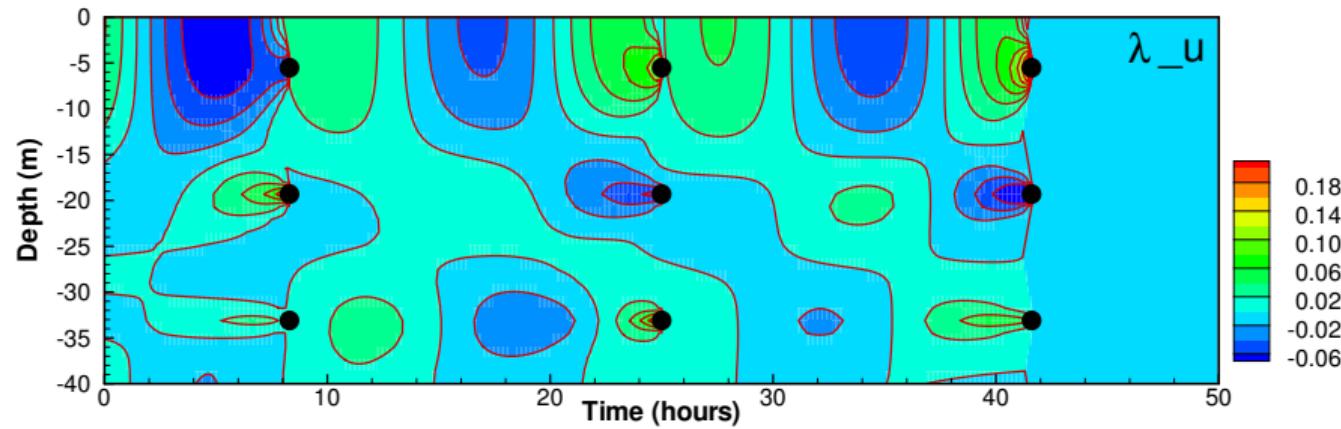


# A convolution of an adjoint representer $\mathbf{C}_{qq} \circ \mathbf{s}$

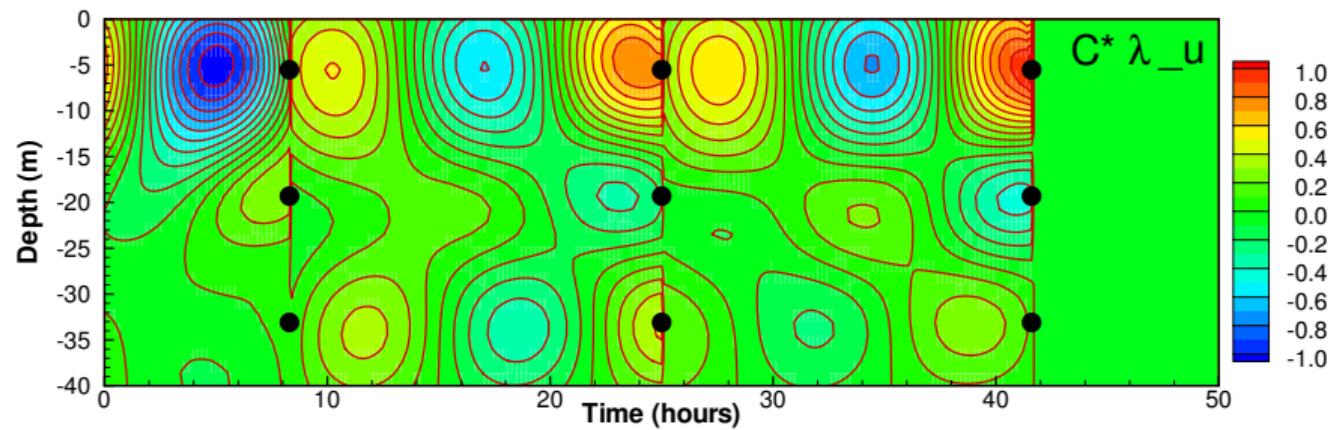


A representer  $\mathbf{r}$ 

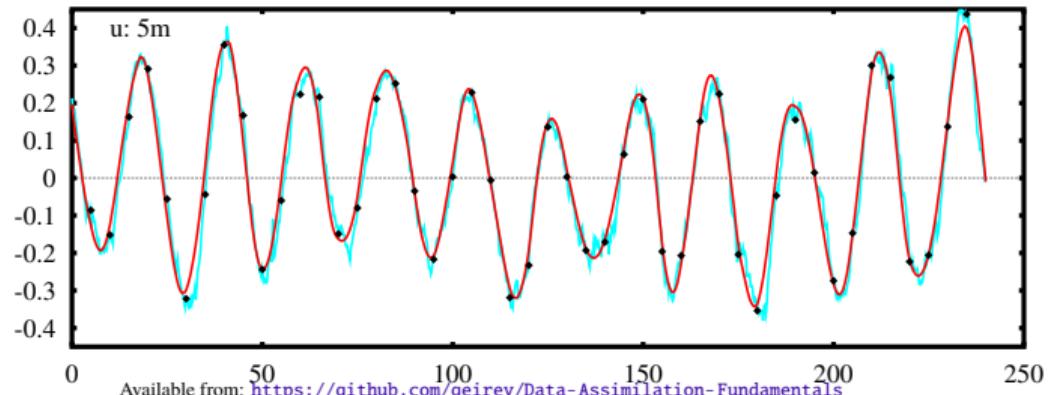
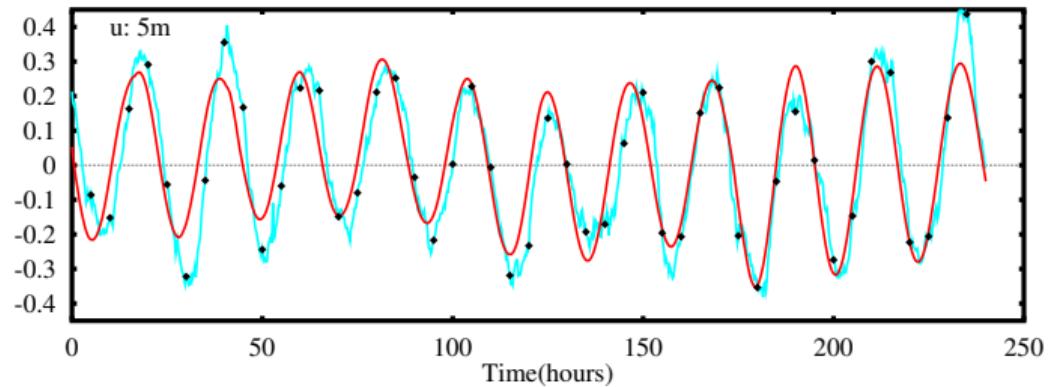
# The adjoint $\lambda$



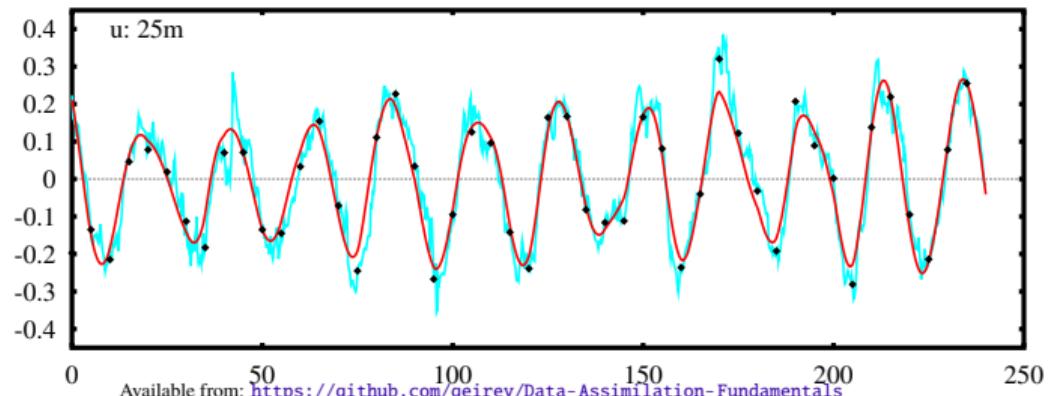
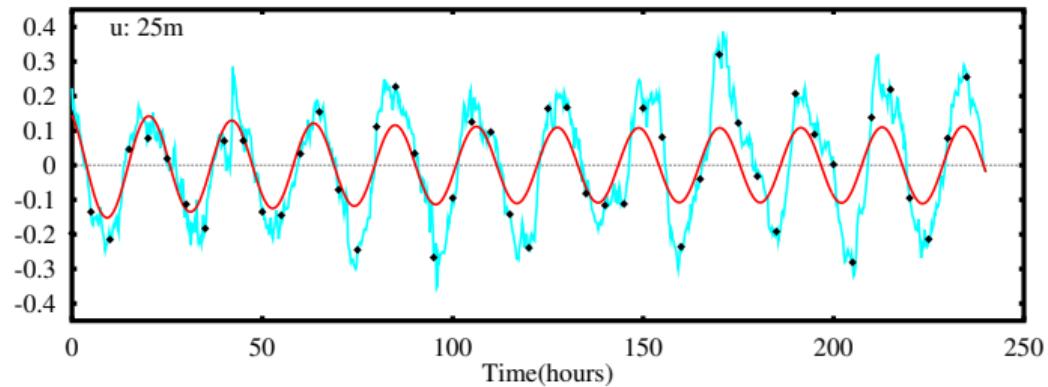
# The convolution of the adjoint $\mathbf{C}_{qq} \circ \lambda$



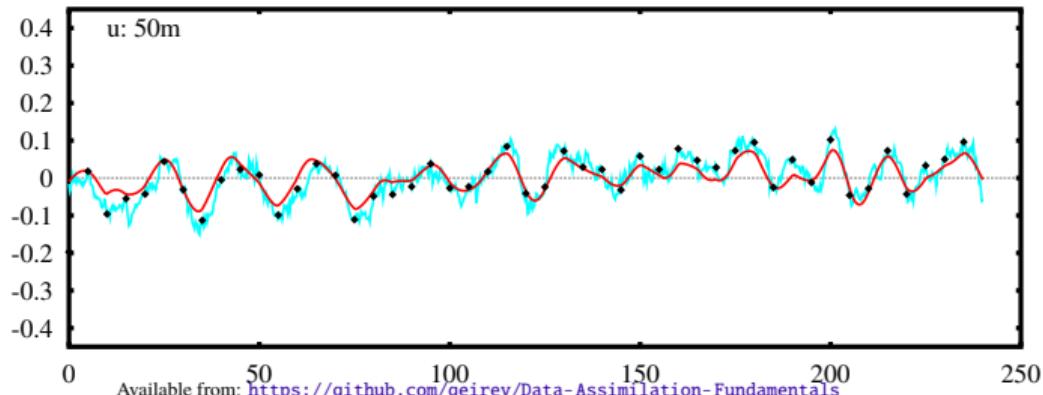
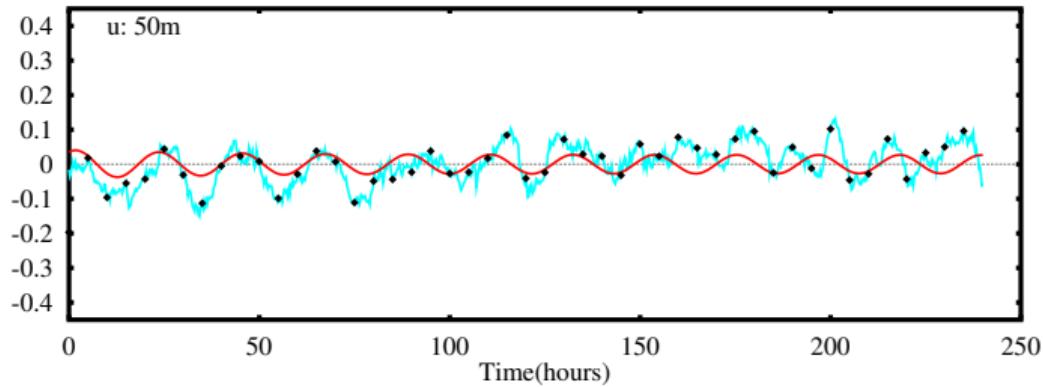
## Strong and weak constraint solution



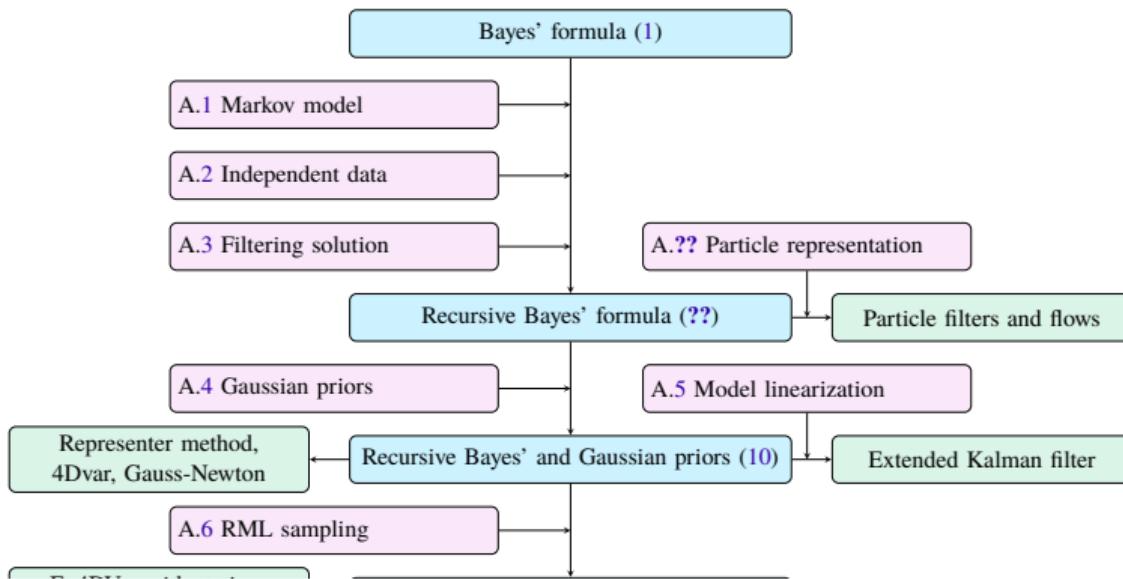
## Strong and weak constraint solution



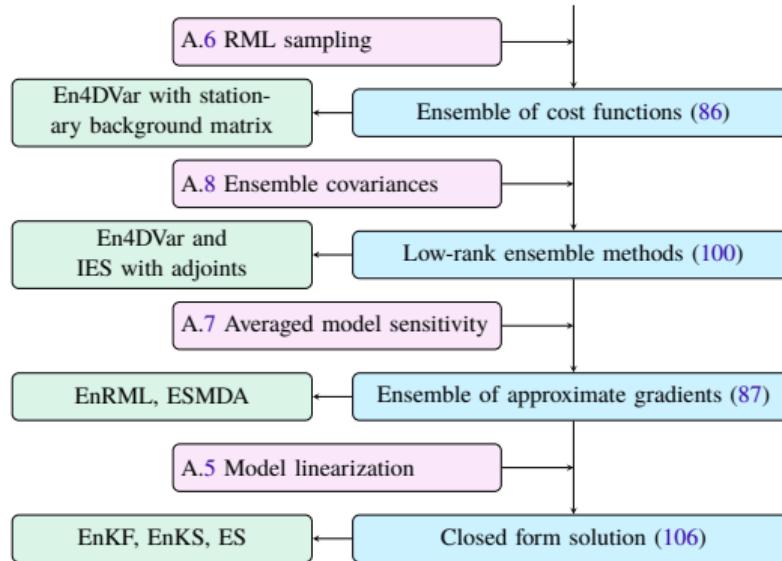
## Strong and weak constraint solution



# Overview of approximations and methods



# Overview of approximations and methods



## Linearization leads to an approximate explicit solution

### Approximation 5 (Linearization)

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

$$\mathbf{g}(\mathbf{z}) \approx \mathbf{g}(\mathbf{z}^f) + \mathbf{G}(\mathbf{z} - \mathbf{z}^f), \quad (83)$$

*and approximate the gradient evaluated at the prior estimate*

$$\nabla_{\mathbf{z}} \mathbf{g}(\mathbf{z}) \approx \mathbf{G}^T, \quad (84)$$

*where we have defined*

$$\mathbf{G}^T = \nabla_{\mathbf{z}} \mathbf{g}(\mathbf{z}) \Big|_{\mathbf{z}=\mathbf{z}^f}. \quad (85)$$

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

## 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. (86).*

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

$$\mathcal{J}(\mathbf{z}_j) = \frac{1}{2} (\mathbf{z}_j - \mathbf{z}_j^f)^T \mathbf{C}_{zz}^{-1} (\mathbf{z}_j - \mathbf{z}_j^f) + \frac{1}{2} (\mathbf{g}(\mathbf{z}_j) - \mathbf{d}_j)^T \mathbf{C}_{dd}^{-1} (\mathbf{g}(\mathbf{z}_j) - \mathbf{d}_j), \quad (86)$$

### Ensemble of gradients set to zero

$$\mathbf{C}_{zz}^{-1} (\mathbf{z}_j - \mathbf{z}_j^f) + \nabla_{\mathbf{z}} \mathbf{g}(\mathbf{z}_j) \mathbf{C}_{dd}^{-1} (\mathbf{g}(\mathbf{z}_j) - \mathbf{d}_j) = 0. \quad (87)$$

Thus, we must solve  $N$  independent minimizations.

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

## Ensemble of incremental 4DVars

$$\mathcal{J}(\delta \mathbf{z}_j) = \frac{1}{2} (\delta \mathbf{z}_j - \boldsymbol{\xi}_j^i)^T \mathbf{C}_{zz}^{-1} (\delta \mathbf{z}_j - \boldsymbol{\xi}_j^i) + \frac{1}{2} (\mathbf{G}_j^i \delta \mathbf{z}_j - \boldsymbol{\eta}_j^i)^T \mathbf{C}_{dd}^{-1} (\mathbf{G}_j^i \delta \mathbf{z}_j - \boldsymbol{\eta}_j^i). \quad (88)$$

## Ensemble of GN iterations

$$\mathbf{z}_j^{i+1} = \mathbf{z}_j^i - \gamma \left( \mathbf{C}_{zz}^{-1} + \mathbf{G}_j^{iT} \mathbf{C}_{dd}^{-1} \mathbf{G}_j^i \right)^{-1} \left( \mathbf{C}_{zz}^{-1} (\mathbf{z}_j^i - \mathbf{z}_j^f) + \mathbf{G}_j^{iT} \mathbf{C}_{dd}^{-1} (\mathbf{g}(\mathbf{z}_j^i) - \mathbf{d}_j) \right), \quad (89)$$

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

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

## 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. (90) and  $\mathbf{G}_j^i$  in Eq. (89) as the sensitivity matrix in linear regression and represent them using the definition*

$$\mathbf{G}_j \approx \mathbf{G} \triangleq \mathbf{C}_{yz} \mathbf{C}_{zz}^{-1}. \quad (91)$$

*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.*

$$\mathbf{Z} = (\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_N). \quad (92)$$

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

$$\mathbf{\Pi} = \left( \mathbf{I} - \frac{1}{N} \mathbf{1} \mathbf{1}^T \right) / \sqrt{N - 1}, \quad (93)$$

Ensemble anomalies

$$\mathbf{A} = \mathbf{Z} \mathbf{\Pi}, \quad (94)$$

## Ensemble matrices

Ensemble covariance

$$\bar{\mathbf{C}}_{zz} = \mathbf{A}\mathbf{A}^T. \quad (95)$$

Measurement error covariance matrix

$$\bar{\mathbf{C}}_{dd} = \mathbf{E}\mathbf{E}^T. \quad (96)$$

Ensemble of model-predicted measurements

$$\mathbf{\Upsilon} = \mathbf{g}(\mathbf{Z}), \quad (97)$$

with anomalies

$$\mathbf{Y} = \mathbf{\Upsilon}\boldsymbol{\Pi}, \quad (98)$$

## Solution is confined to ensemble subspace

Search for the solution in the ensemble subspace

$$\mathbf{Z}^a = \mathbf{Z}^f + \mathbf{A}\mathbf{W}. \quad (99)$$

### Cost function in ensemble subspace

$$\mathcal{J}(\mathbf{w}_j) = \frac{1}{2} \mathbf{w}_j^T \mathbf{w}_j + \frac{1}{2} (\mathbf{g}(\mathbf{z}_j^f + \mathbf{A}\mathbf{w}_j) - \mathbf{d}_j)^T \bar{\mathbf{C}}_{dd}^{-1} (\mathbf{g}(\mathbf{z}_j^f + \mathbf{A}\mathbf{w}_j) - \mathbf{d}_j), \quad (100)$$

# Gradient and Hessian

Gradient

$$\nabla_{\mathbf{w}} \mathcal{J}(\mathbf{w}_j) = \mathbf{w}_j + (\mathbf{G}_j \mathbf{A})^T \bar{\mathbf{C}}_{dd}^{-1} (\mathbf{g}(\mathbf{z}_j^f + \mathbf{A}\mathbf{w}_j) - \mathbf{d}_j), \quad (101)$$

Approximate Hessian

$$\nabla_{\mathbf{w}} \nabla_{\mathbf{w}} \mathcal{J}(\mathbf{w}_j) \approx \mathbf{I} + (\mathbf{G}_j \mathbf{A})^T \bar{\mathbf{C}}_{dd}^{-1} (\mathbf{G}_j \mathbf{A}). \quad (102)$$

Tangent-linear model

$$\mathbf{G}_j = \left( \nabla_{\mathbf{w}} \mathbf{g} \Big|_{\mathbf{z}_j^f + \mathbf{A}\mathbf{w}_j} \right)^T \in \Re^{m \times n}, \quad (103)$$

## Gauss-Newton iterations

$$\mathbf{w}_j^{i+1} = \mathbf{w}_j^i - \gamma \left\{ \mathbf{w}_j^i - (\mathbf{G}_j^i \mathbf{A})^T \left( (\mathbf{G}_j^i \mathbf{A})(\mathbf{G}_j^i \mathbf{A})^T + \bar{\mathbf{C}}_{dd} \right)^{-1} \left( (\mathbf{G}_j^i \mathbf{A})\mathbf{w}_j^i + \mathbf{d}_j - \mathbf{g}(\mathbf{z}_j^f + \mathbf{A}\mathbf{w}_j^i) \right) \right\}. \quad (104)$$

with linear regression approximation for model sensitivities

$$\mathbf{G}_j^i \approx \bar{\mathbf{G}}^i \triangleq \bar{\mathbf{C}}_{yz}^i \bar{\mathbf{C}}_{zz}^{i\dagger} = \mathbf{Y}^i \mathbf{A}^{i\dagger}, \quad (105)$$

# Subspace EnRML

```

1: Input:  $\mathbf{Z} \in \mathbb{R}^{n \times N}$ 
2: Input:  $\mathbf{D} \in \mathbb{R}^{m \times N}$ 
3:  $\mathbf{W}^{(0)} = 0$ 
4:  $\mathbf{\Pi} = \left(\mathbf{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^T\right) / \sqrt{N-1}$ 
5:  $\mathbf{E} = \mathbf{D}\mathbf{\Pi}$ 
6:  $i = 0$ 
7: repeat
8:    $\mathbf{Y}^i = \mathbf{g}(\mathbf{Z}^i)\mathbf{\Pi}$  ▷  $\mathbf{Y} \in \mathbb{R}^{m \times N}$ 
9:   if  $n < N - 1$  then
10:     $\mathbf{Y}^i = \mathbf{Y}^i \mathbf{A}^{i\dagger} \mathbf{A}^i$ 
11:   end if
12:    $\mathbf{\Omega}^i = \mathbf{I} + \mathbf{W}^i \mathbf{\Pi}$ 
13:    $\mathbf{S}^i = \mathbf{Y}^i \mathbf{\Omega}^{i-1}$ 
14:    $\mathbf{W}^{i+1} = \mathbf{W}^i - \gamma \left( \mathbf{W}^i - \mathbf{S}^{i^T} \left( \mathbf{S}^i \mathbf{S}^{i^T} + \mathbf{E} \mathbf{E}^T \right)^{-1} \left( \mathbf{S}^i \mathbf{W}^i + \mathbf{D} - \mathbf{g}(\mathbf{Z}^i) \right) \right)$ 
15:    $\mathbf{Z}^{i+1} = \mathbf{Z} \left( \mathbf{I} + \mathbf{W}^{i+1} \right) / \sqrt{N-1}$ 
16:    $i = i + 1$ 
17: until convergence

```

- ▷ Prior state-vector ensemble
- ▷ Perturbed measurements
- ▷  $\mathbf{W} \in \mathbb{R}^{N \times N}$
- ▷  $\mathbf{\Pi} \in \mathbb{R}^{N \times N}$
- ▷  $\mathbf{E} \in \mathbb{R}^{m \times N}$
- ▷  $\mathbf{\Omega} \in \mathbb{R}^{N \times N}$
- ▷  $\mathbf{S} \in \mathbb{R}^{m \times N}$

## EnKF update

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

$$\mathbf{Z}^a = \mathbf{Z}^f + \mathbf{A} \mathbf{Y}^T (\mathbf{Y} \mathbf{Y}^T + \mathbf{E} \mathbf{E}^T)^{-1} (\mathbf{D} - \mathbf{g}(\mathbf{Z}^f)) \quad (106)$$

Alternative interpretation using

$$\mathbf{W} = \mathbf{Y}^T (\mathbf{Y} \mathbf{Y}^T + \mathbf{E} \mathbf{E}^T)^{-1} (\mathbf{D} - \mathbf{g}(\mathbf{Z}^f)), \quad (107)$$

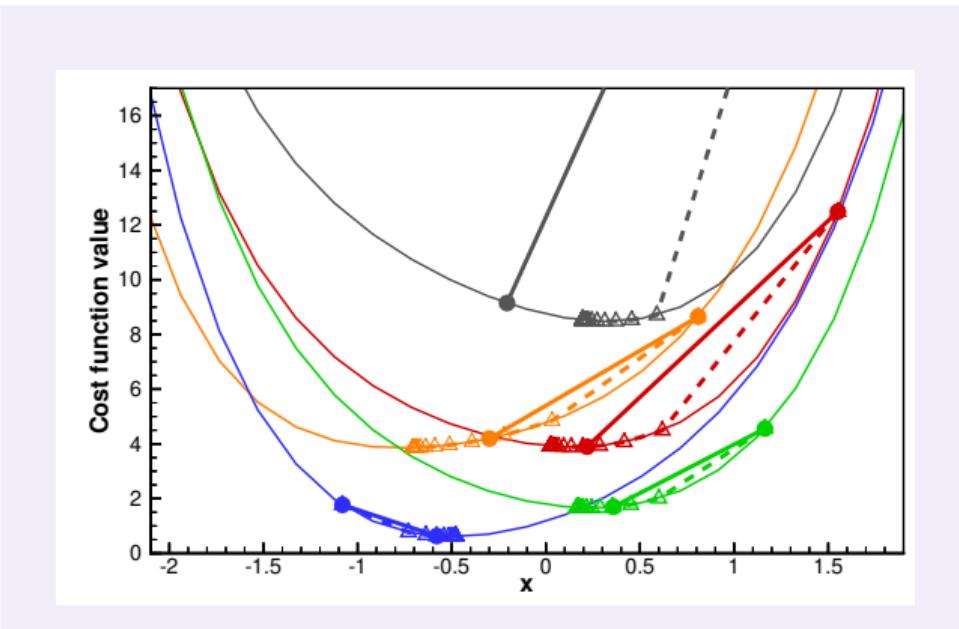
to get

$$\mathbf{Z}^a = \mathbf{Z}^f \left( \mathbf{I} + \mathbf{W} / \sqrt{N - 1} \right) \quad (108)$$

But note that

$$\mathbf{Y} = \begin{cases} \mathbf{Y} & \text{for } n \geq N - 1 \\ \mathbf{Y} \mathbf{A}^\dagger \mathbf{A} & \text{for } n < N - 1. \end{cases} \quad (109)$$

## ES and IES illustration: Non-linear model



- IES gets closer to minimum than ES
- 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).

## Comparison of methods on a scalar model

## Scalar models

$$y = g(x, q) = x + 0.3x^3 + q, \quad (110)$$

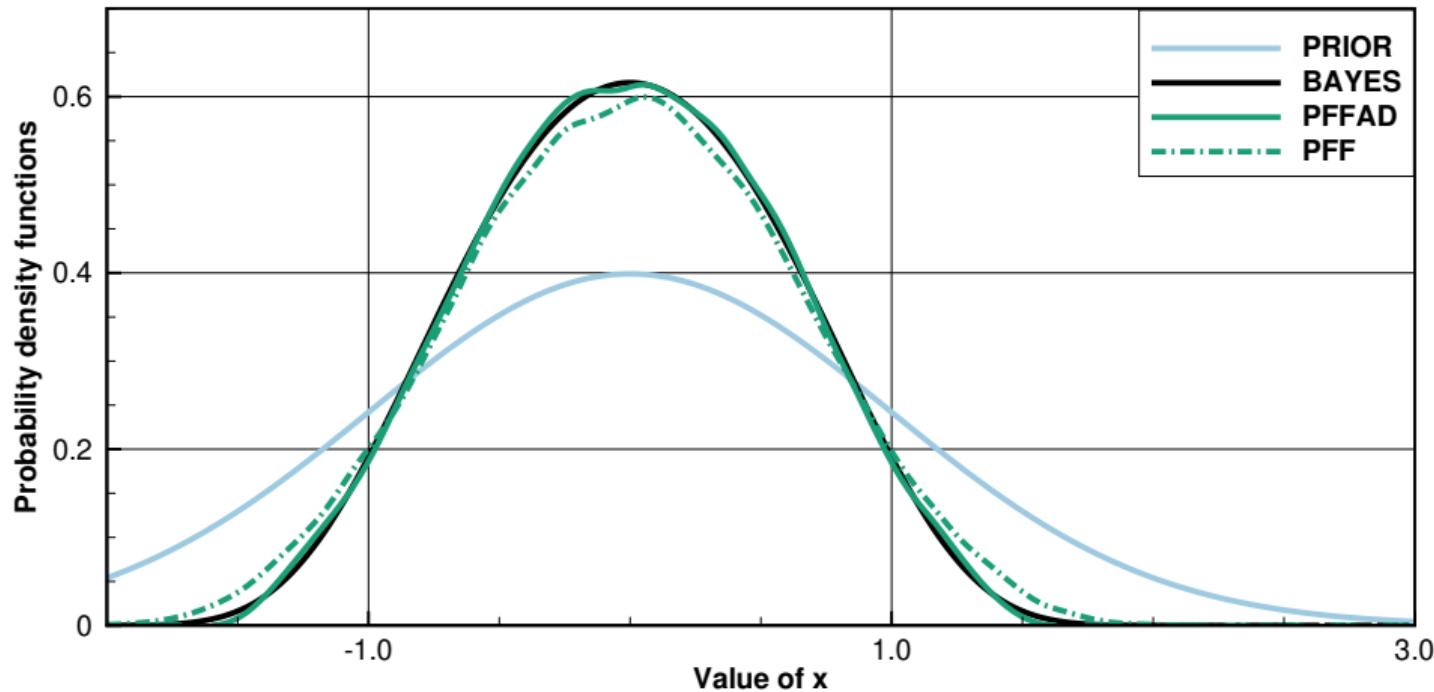
$$y = g(x, q) = 1 + \sin(x) + q, \quad (111)$$

## Scalar examples

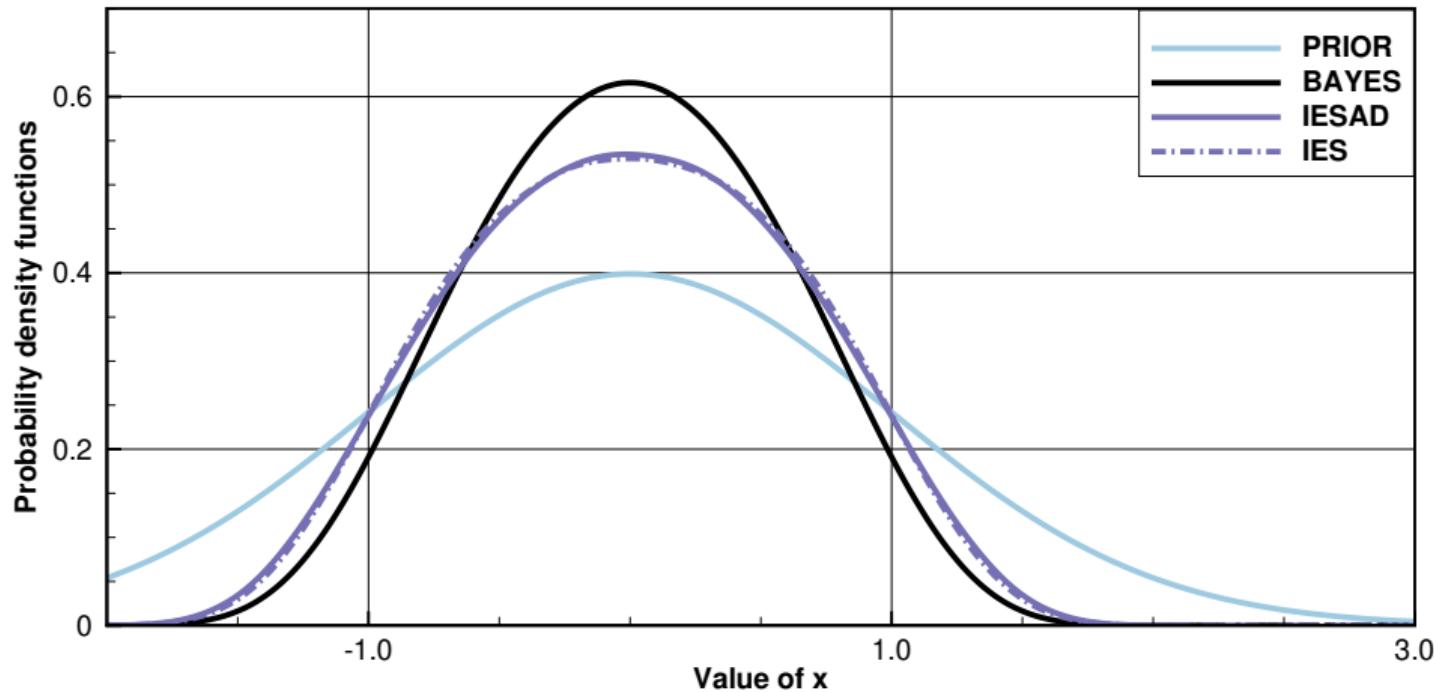
1. Model Eq. (110),  $x_j^f = \mathcal{N}(x^f = 0.0, C_{xx} = 1.0)$ ,  $d_j = \mathcal{N}(0.0, 1.0)$ , and  $q = \mathcal{N}(0, C_{qq} = 0.25)$ .
2. Model Eq. (110),  $x_j^f = \mathcal{N}(x^f = 1.0, C_{xx} = 1.0)$ ,  $d_j = \mathcal{N}(0.0, -1.0)$ , and  $q = \mathcal{N}(0, C_{qq} = 0.25)$ .
3. Model Eq. (111),  $x_j^f = \mathcal{N}(x^f = 1.0, C_{xx} = 1.0)$ ,  $d_j = \mathcal{N}(0.0, 1.0)$ , and  $q = \mathcal{N}(0, C_{qq} = 0.25)$ .

Ensemble size  $N = 10^7$ .

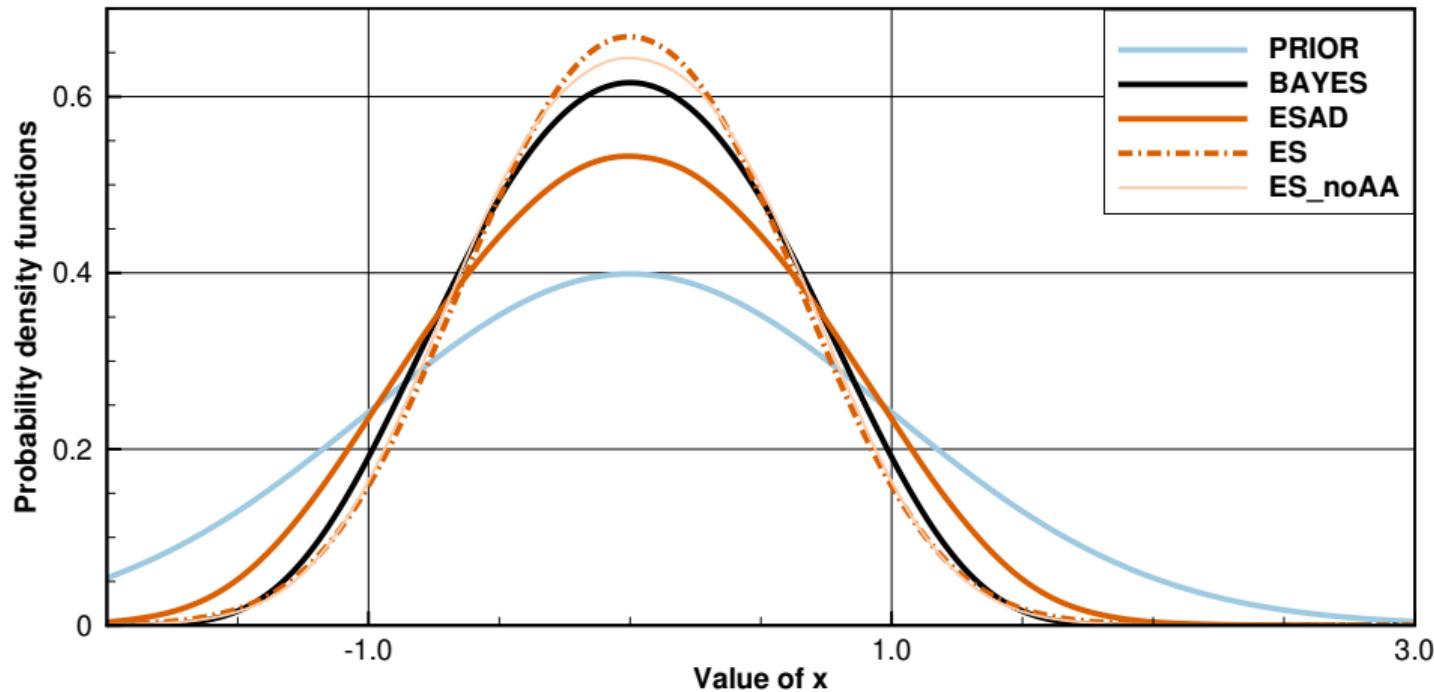
## Case 1: Particle flow



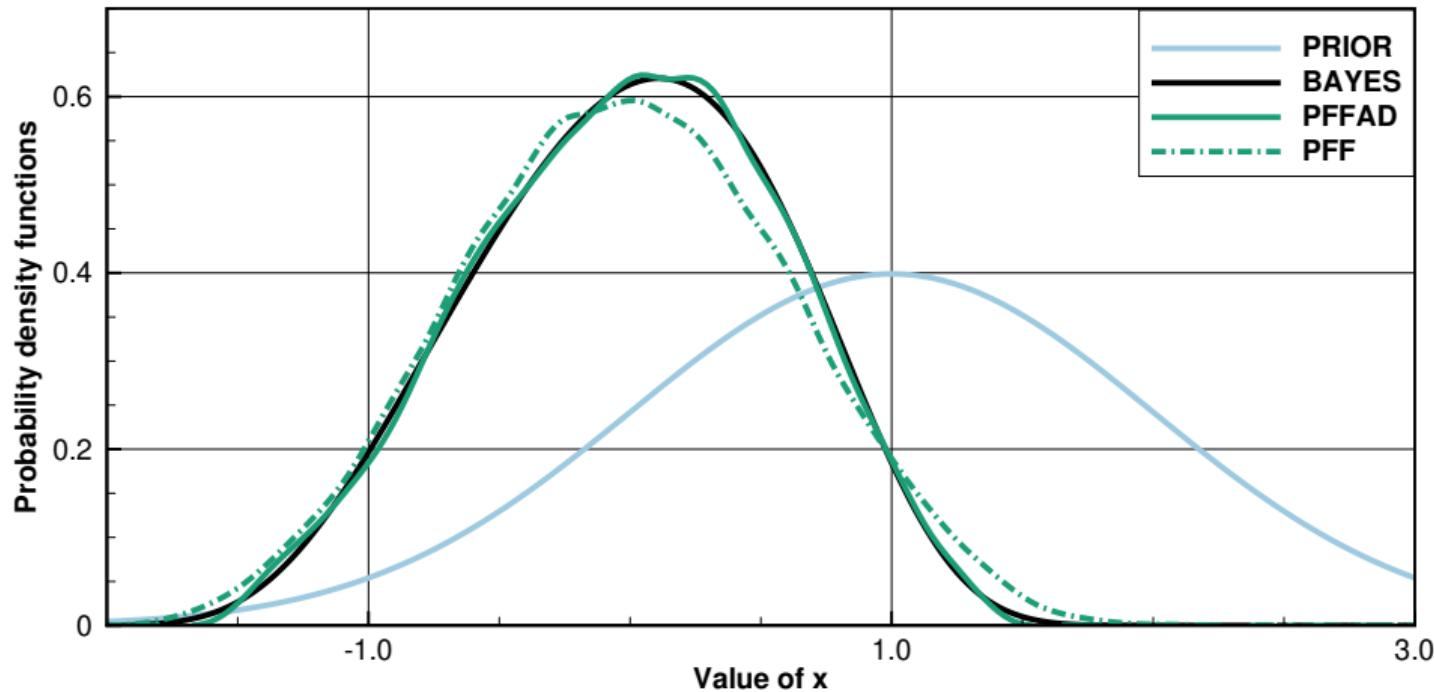
## Case 1: IES



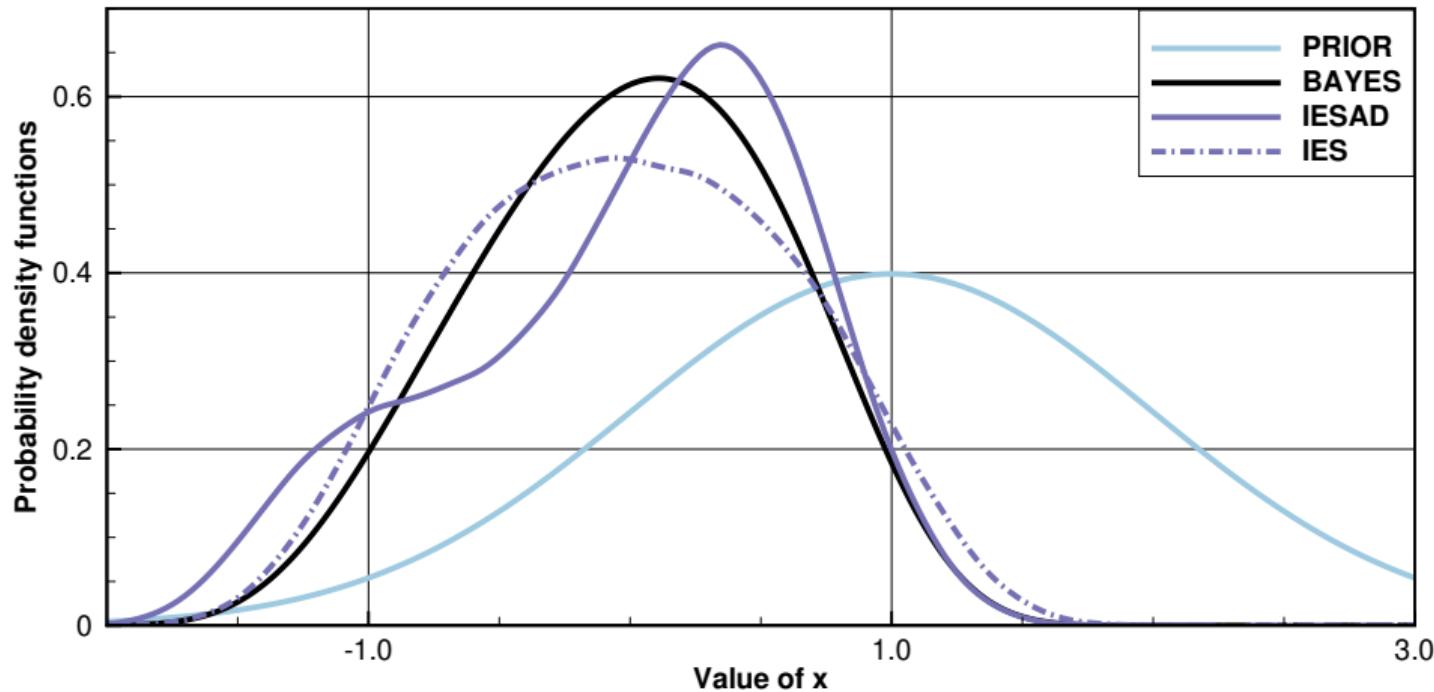
## Case 1: EnKF



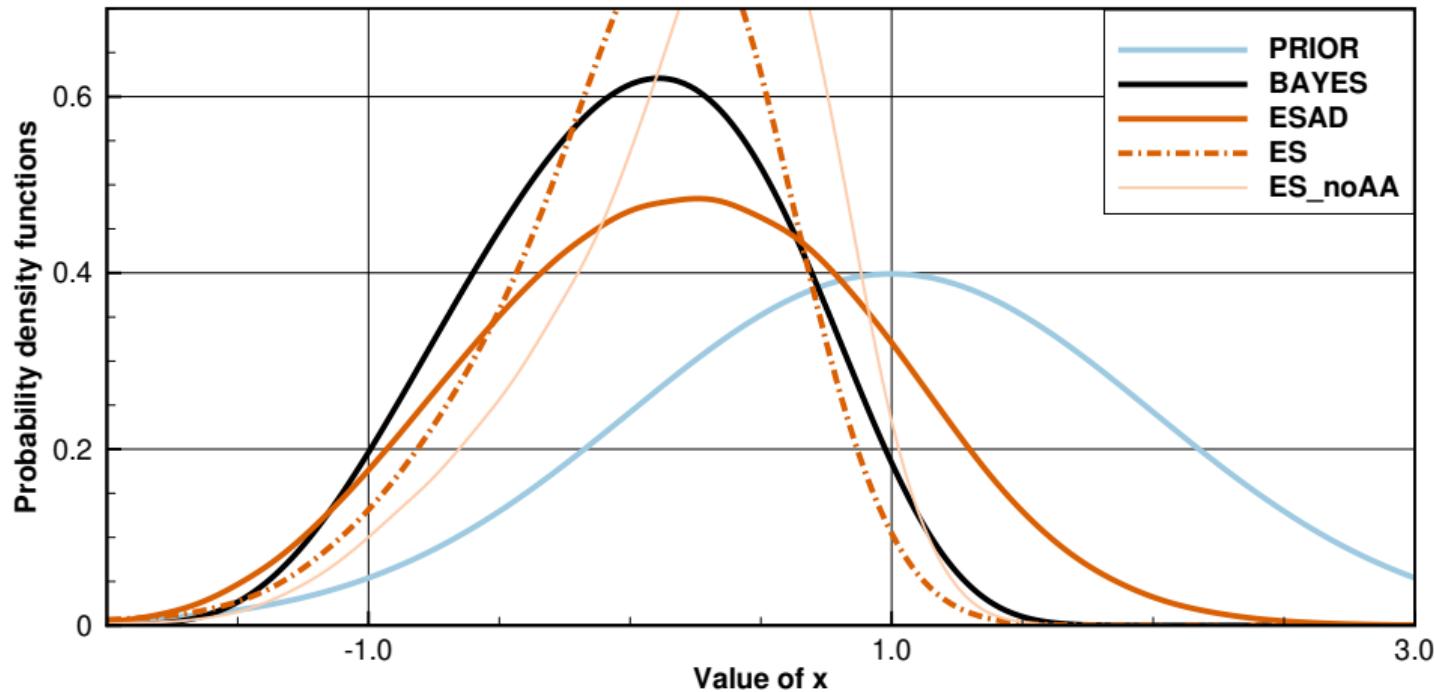
## Case 2: Particle flow



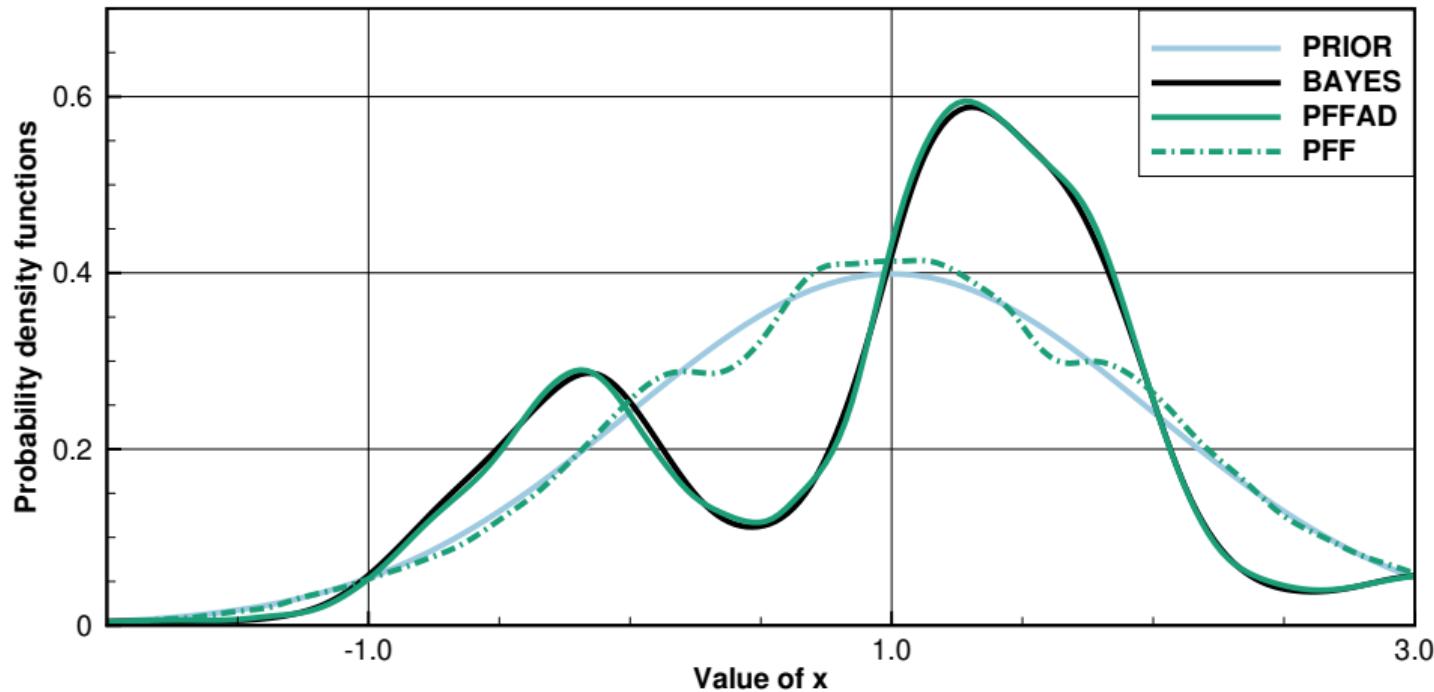
## Case 2: IES



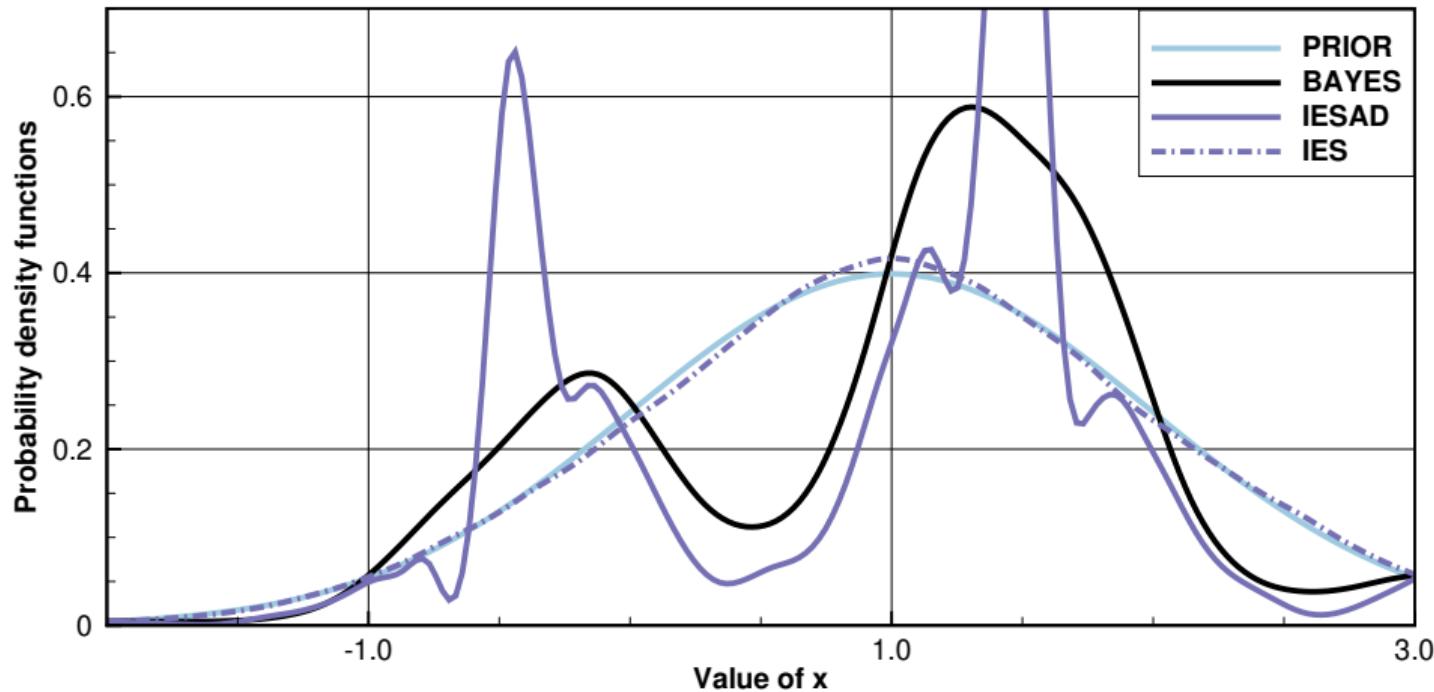
## Case 2: EnKF



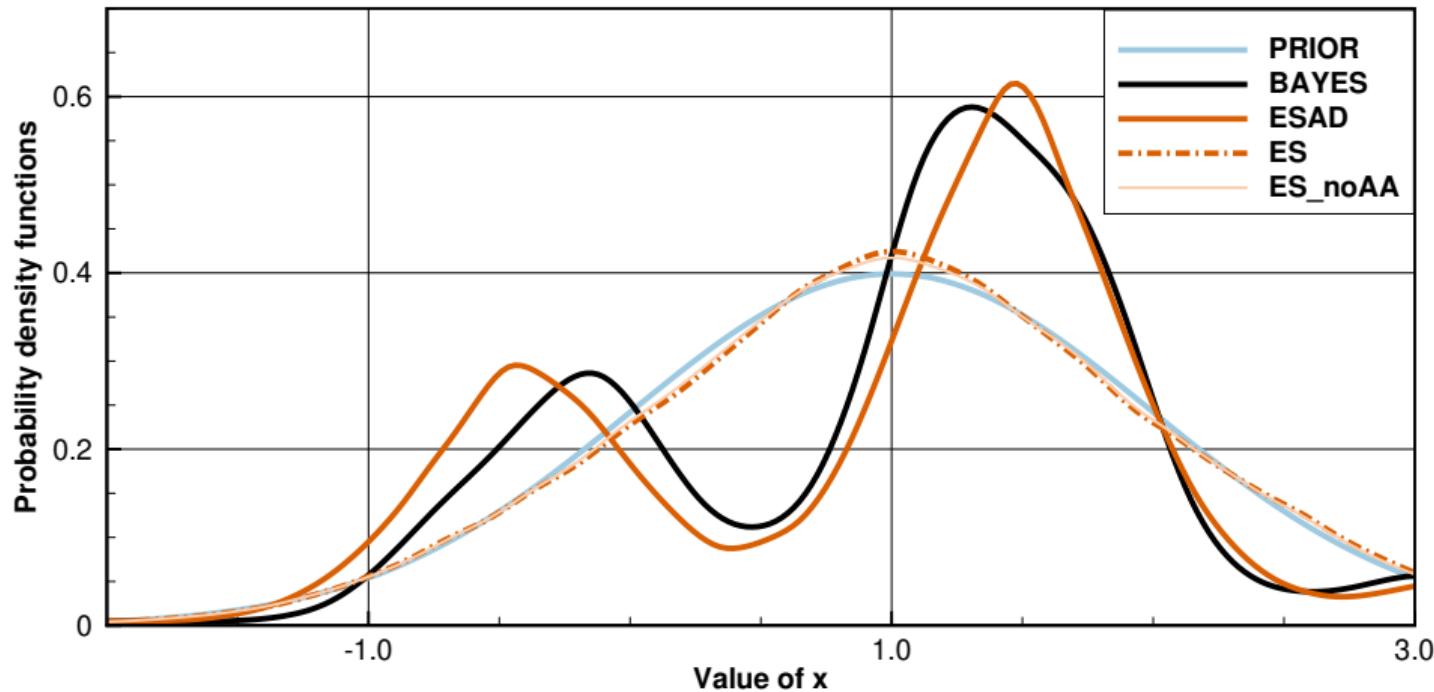
## Case 3: Particle flow



## Case 3: IES



## Case 3: EnKF



## ESMDA with a SARS-COV-2 pandemic nmodel

## Available observations

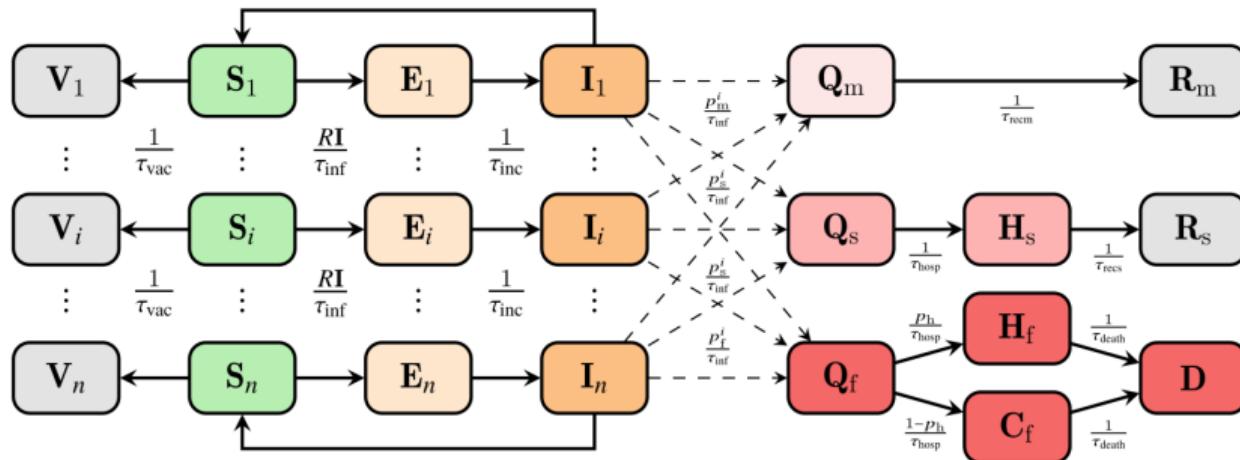
Hospitalized: Within different agegroups and gender.

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.

## Validity of the SEIR model

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

More complex models involve additional parameters.

## Constant model parameters

1. Relative fractions  $p_m^i, p_s^i, p_f^i$  per age group.
2. Fractions dying in a Hospital  $p_h$  versus in a Care home  $1 - p_h$ .

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
$\tau_{\text{inc}}$	5.5	Incubation period
$\tau_{\text{inf}}$	3.8	Infection time
$\tau_{\text{recm}}$	14.0	Recovery time mild cases
$\tau_{\text{recs}}$	5.0	Recovery time severe cases
$\tau_{\text{hosp}}$	6.0	Time until hospitalization
$\tau_{\text{death}}$	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

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

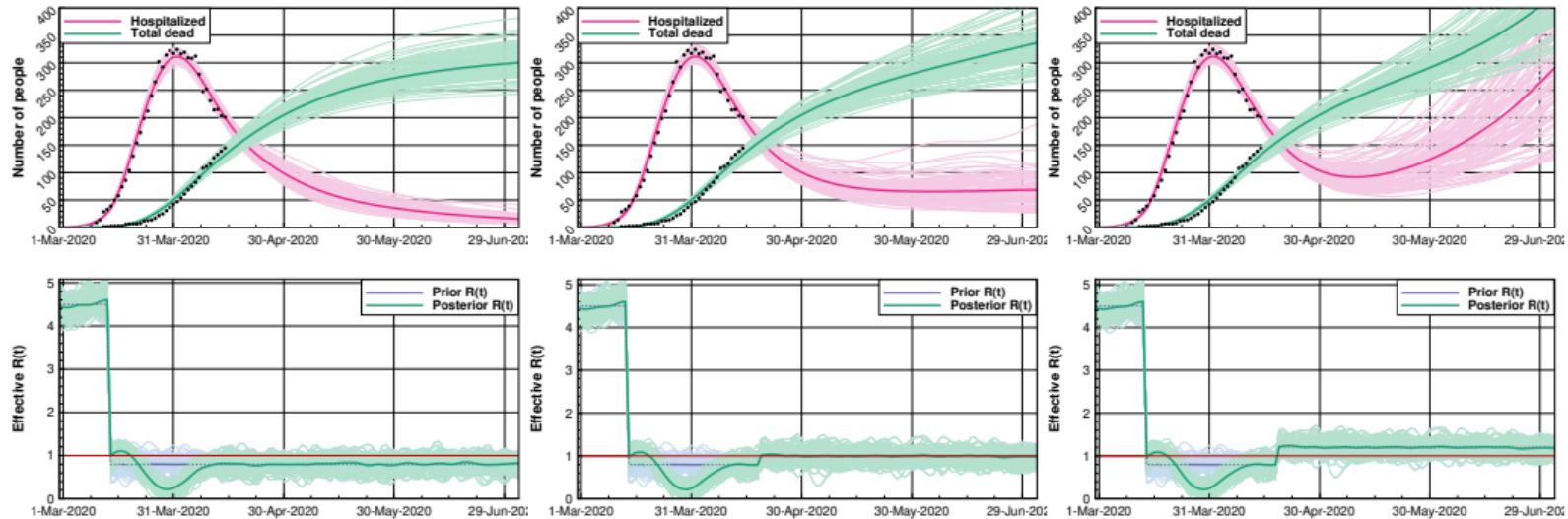
$\mathbf{R}(t)$  is a function of time (steered by how people isolate or interact).

- $R(t)$  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.

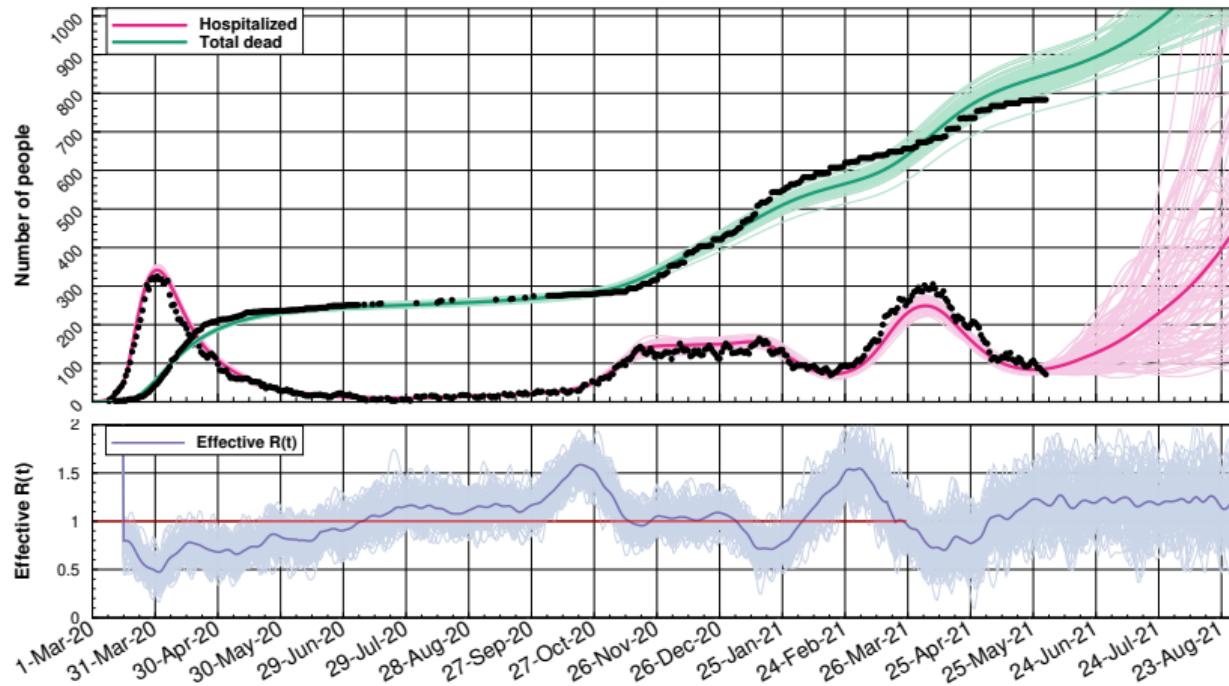
## We used ESMDA

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

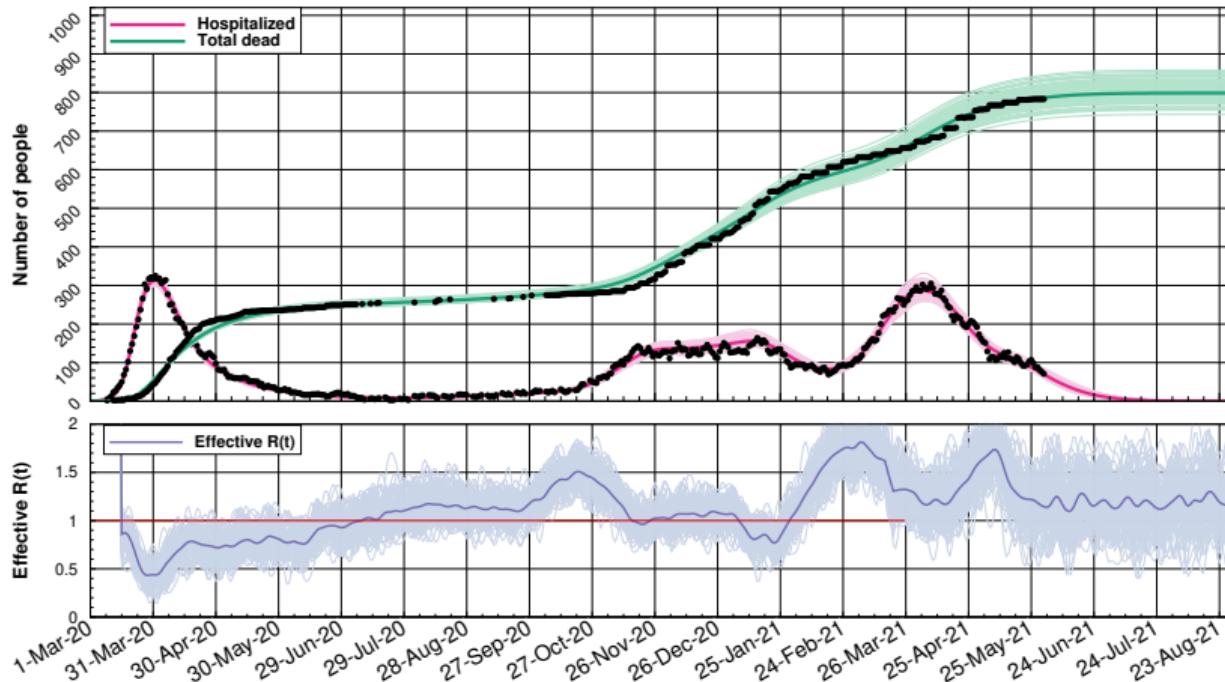
# Back-to-school scenarios for Norway



# Norway: prediction without vaccinations



# Norway: prediction with vaccinations



## Summary EnKF\_seir

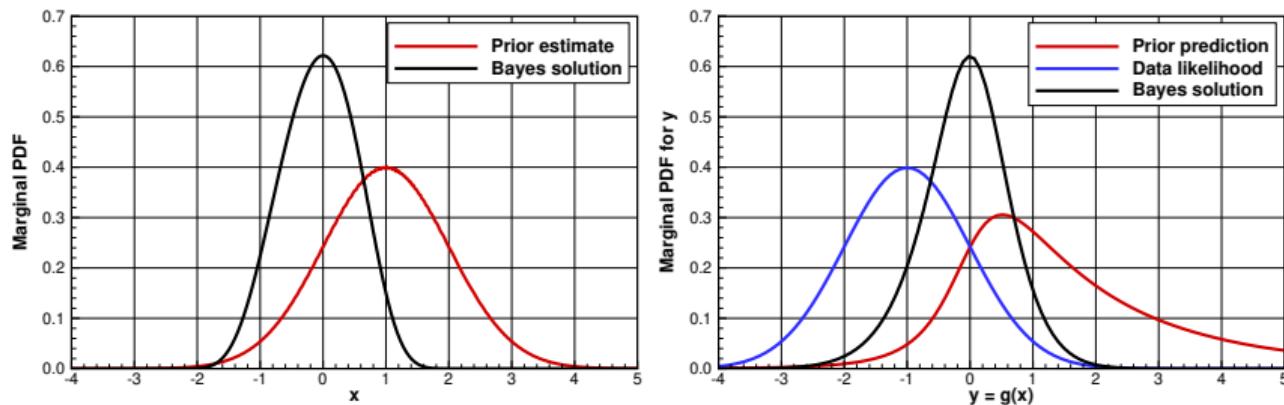
- The DA system tracks the epidemic accurately.
- We can estimate past  $R(t)$ .
- It is possible to quantify the impact of interventions.
- Short-term forecasting using  $R$ -persistence works well.
- Long-term scenario forecasting with specified future  $R$ .
- Code: [https://github.com/geirev/EnKF\\_seir](https://github.com/geirev/EnKF_seir)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

## EnRML for History Matching Petroleum Models

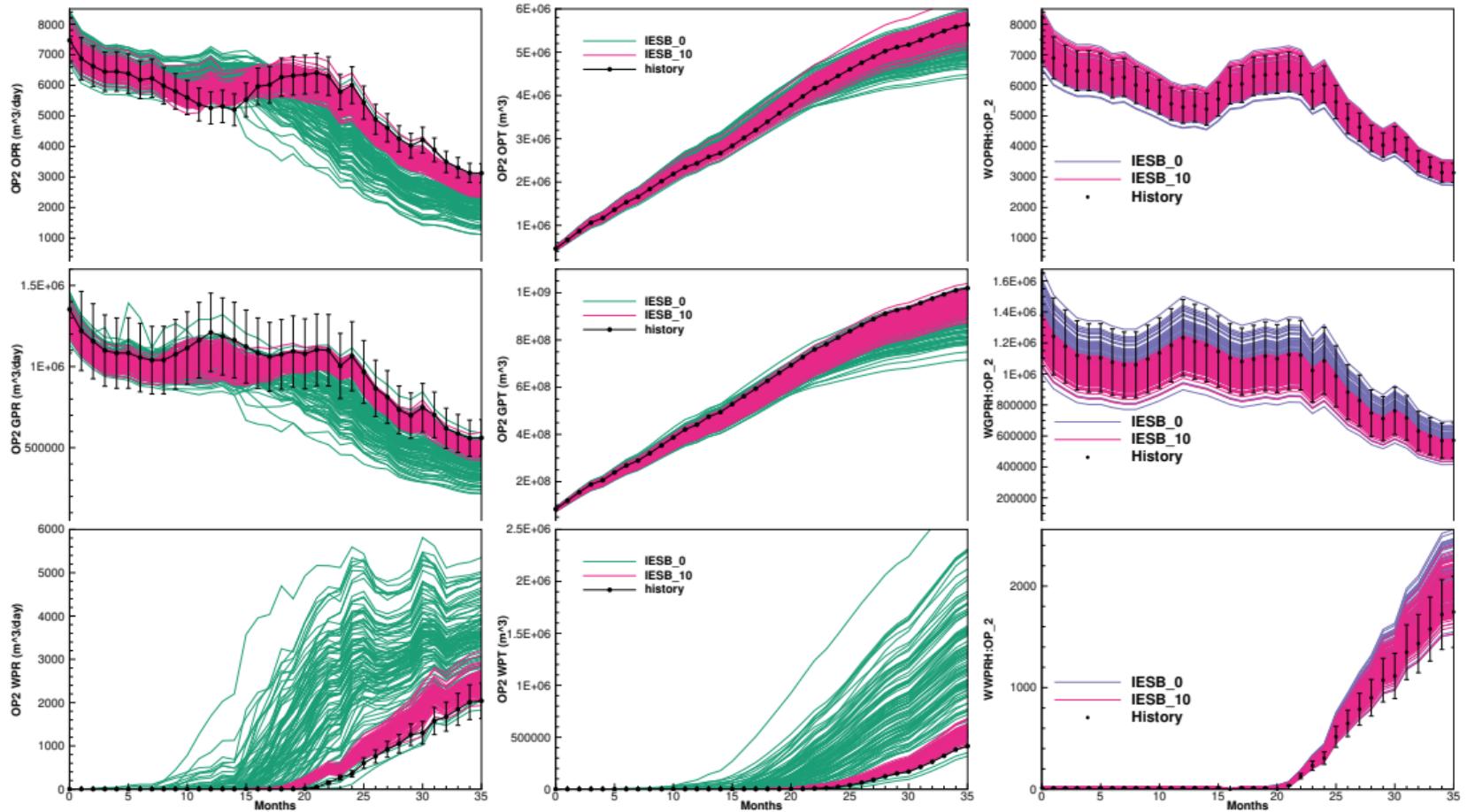
# Illustration of the parameter estimation problem

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

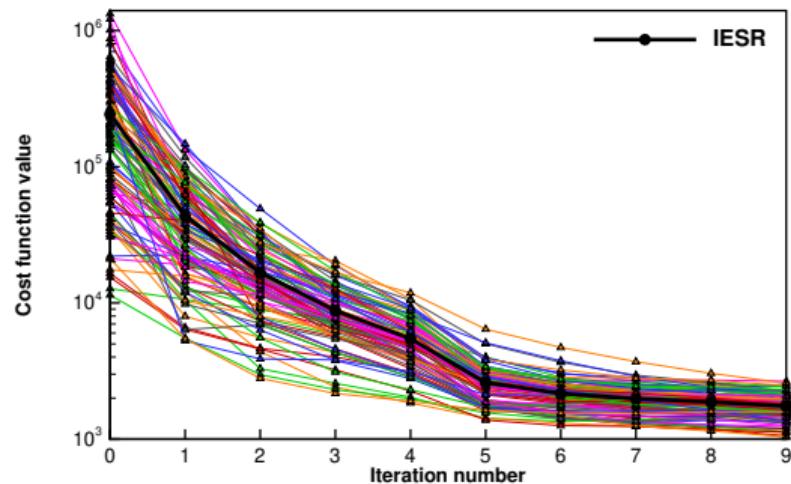
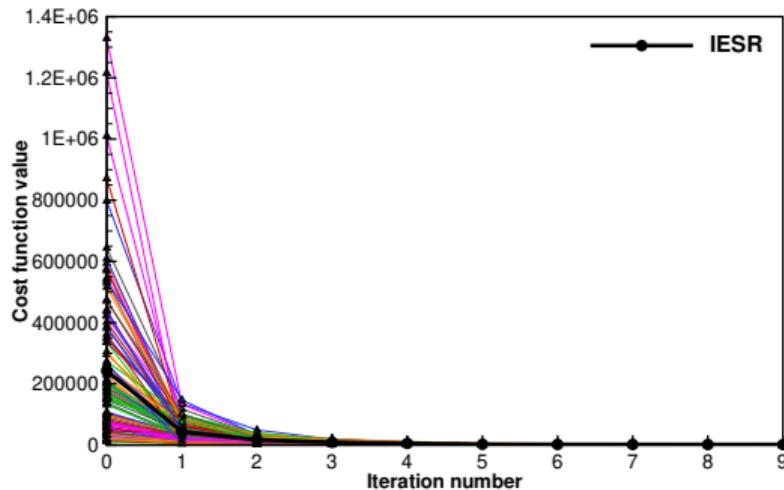
$$f(\mathbf{x}|\mathbf{d}) \propto f(\mathbf{d} | g(\mathbf{x})) f(\mathbf{x})$$



- $\mathbf{x}$  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  $\mathbf{d}$ .



## 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](https://doi.org/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](https://doi.org/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](https://doi.org/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](https://doi.org/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](https://doi.org/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](https://doi.org/10.3389/fams.2019.00047).
- Evensen, G., J. Amezcu, 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](https://doi.org/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](https://doi.org/10.1007/BF00143556).
- Van Leeuwen, P. J. and G. Evensen. Data assimilation and inverse methods in terms of a probabilistic formulation. *Mon. Weather Rev.*, 124: 2898–2913, 1996. doi:[10.1175/1520-0493\(1996\)124<2898:DAAIMI>2.0.CO;2](https://doi.org/10.1175/1520-0493(1996)124<2898:DAAIMI>2.0.CO;2).