Introduction to data assimilation
Kalman filters and ensembles

Vivien Mallet[1]
vivien.mallet@inria.fr

[1]INRIA

CEMRACS, Marseille, July 2016

Outline

Notation

$\boldsymbol{x}^b$	Background state vector, in $\mathbb{R}^{N_x}$
N_x	Size of the state vector
$\boldsymbol{x}^a$	Analysis state vector or best linear unbiased estimator (BLUE)
$\boldsymbol{x}^t$	True state vector
$\boldsymbol{\epsilon}^b$	Background error: $\boldsymbol{\epsilon}^b = \boldsymbol{x}^b - \boldsymbol{x}^t$
$\mathbf{B}$	Background error variance, or background error covariance matrix $\mathbf{B} = \mathrm{E}\left[(\boldsymbol{\epsilon}^b - \mathrm{E}(\boldsymbol{\epsilon}^b))(\boldsymbol{\epsilon}^b - \mathrm{E}(\boldsymbol{\epsilon}^b))^\top\right]$
$\boldsymbol{\epsilon}^a$	Analysis error: $\boldsymbol{\epsilon}^a = \boldsymbol{x}^a - \boldsymbol{x}^t$
$\mathbf{A}$	Analysis error variance $\mathbf{A} = \mathrm{E}\left[(\boldsymbol{\epsilon}^a - \mathrm{E}(\boldsymbol{\epsilon}^a))(\boldsymbol{\epsilon}^a - \mathrm{E}(\boldsymbol{\epsilon}^a))^\top\right]$
$\boldsymbol{y}$	Observation vector, in $\mathbb{R}^{N_y}$
N_y	Number of observations
$\mathcal{H}$	Observation operator maps the state space into observation space
$\boldsymbol{\epsilon}^y$	Observational error: $\boldsymbol{\epsilon}^y = \boldsymbol{y} - \mathcal{H}(\boldsymbol{x}^t)$
$\mathbf{R}$	Observational error variance: $\mathbf{R} = \mathrm{E}\left[(\boldsymbol{\epsilon}^y - \mathrm{E}(\boldsymbol{\epsilon}^y))(\boldsymbol{\epsilon}^y - \mathrm{E}(\boldsymbol{\epsilon}^y))^\top\right]$

Best linear unbiased estimator (BLUE)

Assumptions

- Unbiased background: $\mathrm{E}(\boldsymbol{\epsilon}^b) = \mathbf{0}$
- Unbiased observations: $\mathrm{E}(\boldsymbol{\epsilon}^y) = \mathbf{0}$
- Uncorrelated errors: $\mathrm{E}(\boldsymbol{\epsilon}^b \boldsymbol{\epsilon}^{y\top}) = \mathbf{0}$

BLUE

$$\boldsymbol{x}^a = \boldsymbol{x}^b + \mathbf{K}(\boldsymbol{y} - \mathbf{H}\boldsymbol{x}^b) \tag{1}$$

$$\mathbf{K} = \mathbf{B}\mathbf{H}^\top(\mathbf{H}\mathbf{B}\mathbf{H}^\top + \mathbf{R})^{-1} \tag{2}$$

$$\mathbf{K} = (\mathbf{B}^{-1} + \mathbf{H}^\top\mathbf{R}^{-1}\mathbf{H})^{-1}\mathbf{H}^\top\mathbf{R}^{-1} \tag{3}$$

Optimal interpolation

BLUE is computed every time new observations are available, and the time integration continues from this new state estimator.

Algorithm

1. At initial time step $h = 0$, the initial condition $\boldsymbol{x}_0^f$ is given.
2. To forecast time step $h + 1$,
   - 2.1 if observations $\boldsymbol{y}_h$ are available,
     the analysis is computed as the BLUE

$$\boldsymbol{x}_h^a = \boldsymbol{x}_h^f + \mathbf{K}_h(\boldsymbol{y}_h - \mathbf{H}_h\boldsymbol{x}_h^f) \tag{4}$$

     with

$$\mathbf{K}_h = \mathbf{P}_h^f\mathbf{H}_h^\top(\mathbf{H}_h\mathbf{P}_h^f\mathbf{H}_h^\top + \mathbf{R}_h)^{-1} \tag{5}$$

     and the forecast is

$$\boldsymbol{x}_{h+1}^f = \mathcal{M}_h(\boldsymbol{x}_h^a)\ ; \tag{6}$$

   - 2.2 if there are no observations,
     the forecast is

$$\boldsymbol{x}_{h+1}^f = \mathcal{M}_h(\boldsymbol{x}_h^f)\ . \tag{7}$$

Kalman filter: motivation

Optimal interpolation does not track error changes

- After the analysis step, optimal interpolation provides the analysis state vector $\boldsymbol{x}_h^a$ and its error variance

$$\mathbf{P}_h^a = (\mathbf{I} - \mathbf{K}_h \mathbf{H}_h)\mathbf{P}_h^f \tag{8}$$

or

$$\mathbf{P}_h^a = (\mathbf{I} - \mathbf{K}_h \mathbf{H}_h)\mathbf{P}_h^f(\mathbf{I} - \mathbf{K}_h \mathbf{H}_h)^\top + \mathbf{K}_h \mathbf{R}_h \mathbf{K}_h^\top \tag{9}$$

- After the forecast step, optimal interpolation only provides the forecast state vector $\boldsymbol{x}_{h+1}^f$, and one needs to prescribe the error variance $\mathbf{P}_{h+1}^f$ for later assimilation

In the linear case, Kalman filter tracks both the state and its error variance

- Analysis stage returns $(\boldsymbol{x}_h^a, \mathbf{P}_h^a)$
- Forecast stage returns $(\boldsymbol{x}_{h+1}^f, \mathbf{P}_{h+1}^f)$

Kalman filter: tracking the errors

- The estimation error is decreased after the assimilation, which translates into the analysis error variance
$$\mathbf{P}_h^a = (\mathbf{I} - \mathbf{K}_h \mathbf{H}_h)\mathbf{P}_h^f \tag{10}$$
- The model modifies the errors from one time step to the other. The imperfect model will increase the error, but it may also decrease the error in certain directions.
We introduce the model error $\boldsymbol{\epsilon}_h^m$, which is the missing part in the forecast state, starting from perfect state:
$$\boldsymbol{x}_{h+1}^t = \mathbf{M}_h \boldsymbol{x}_h^t + \boldsymbol{\epsilon}_h^m \tag{11}$$
  - Had we a perfect model $\mathbf{M}_h^t$, which satisfies $\boldsymbol{x}_{h+1}^t = \mathbf{M}_h^t \boldsymbol{x}_h^t$, the model error would be $\boldsymbol{\epsilon}_h^m = \mathbf{M}_h^t \boldsymbol{x}_h^t - \mathbf{M}_h \boldsymbol{x}_h^t$, which is the opposite sign to the natural definition.

Kalman filter: model error

Definition of model error:

$$\boldsymbol{x}_{h+1}^t = \mathbf{M}_h \boldsymbol{x}_h^t + \boldsymbol{\epsilon}_h^m$$

Assumptions

- The model error has zero mean
$$\mathrm{E}(\boldsymbol{\epsilon}_h^m) = \mathbf{0} \tag{12}$$
- The model error variance is denoted
$$\mathbf{Q}_h = \mathrm{E}(\boldsymbol{\epsilon}_h^m \boldsymbol{\epsilon}_h^{m\top}) \tag{13}$$
- We assume that the model error is uncorrelated with the analysis error:
$$\mathrm{E}(\boldsymbol{\epsilon}_h^m \boldsymbol{\epsilon}_h^{a\top}) = \mathbf{0} \tag{14}$$

Kalman filter: propagation of errors

Error vector

$$\boldsymbol{\epsilon}^f_{h+1} = \boldsymbol{x}^f_{h+1} - \boldsymbol{x}^t_{h+1} \tag{15}$$

$$\boldsymbol{\epsilon}^f_{h+1} = \mathbf{M}_h \boldsymbol{x}^a_h - \mathbf{M}_h \boldsymbol{x}^t_h - \boldsymbol{\epsilon}^m_h$$

$$\boldsymbol{\epsilon}^f_{h+1} = \mathbf{M}_h \boldsymbol{\epsilon}^a_h - \boldsymbol{\epsilon}^m_h \tag{16}$$

The mean forecast error is

$$\mathrm{E}(\boldsymbol{\epsilon}^f_{h+1}) = \mathbf{M}_h \, \mathrm{E}(\boldsymbol{\epsilon}^a_h) - \mathrm{E}(\boldsymbol{\epsilon}^m_h)$$

$$\mathrm{E}(\boldsymbol{\epsilon}^f_{h+1}) = \mathbf{0} \tag{17}$$

Kalman: propagation of errors

Error variance

$$\boldsymbol{\epsilon}^f_{h+1} = \mathbf{M}_h \boldsymbol{\epsilon}^a_h - \boldsymbol{\epsilon}^m_h$$

$$\mathrm{var}(\boldsymbol{\epsilon}^f_{h+1}) = \mathrm{E}\left(\boldsymbol{\epsilon}^f_{h+1} {\boldsymbol{\epsilon}^f_{h+1}}^\top\right)$$

$$\begin{aligned}\mathrm{var}(\boldsymbol{\epsilon}^f_{h+1}) = &\mathrm{E}(\mathbf{M}_h \boldsymbol{\epsilon}^a_h {\boldsymbol{\epsilon}^a_h}^\top \mathbf{M}_h^\top) + \mathrm{E}(\boldsymbol{\epsilon}^m_h {\boldsymbol{\epsilon}^m_h}^\top) \\ &- \mathrm{E}(\mathbf{M}_h \boldsymbol{\epsilon}^a_h {\boldsymbol{\epsilon}^m_h}^\top) - \mathrm{E}(\boldsymbol{\epsilon}^m_h {\boldsymbol{\epsilon}^a_h}^\top {\mathbf{M}_h}^\top)\end{aligned}$$

$$\mathrm{var}(\boldsymbol{\epsilon}^f_{h+1}) = \mathbf{M}_h \,\mathrm{E}(\boldsymbol{\epsilon}^a_h {\boldsymbol{\epsilon}^a_h}^\top)\mathbf{M}_h^\top + \mathrm{E}(\boldsymbol{\epsilon}^m_h {\boldsymbol{\epsilon}^m_h}^\top)$$

$$\mathbf{P}^f_{h+1} = \mathrm{var}(\boldsymbol{\epsilon}^f_{h+1}) = \mathbf{M}_h \mathbf{P}^a_h \mathbf{M}_h^\top + \mathbf{Q}_h \tag{18}$$

Kalman filter: algorithm

1. At initial time step $h = 0$, the initial condition $\boldsymbol{x}_0^f$ and its error variance $\mathbf{P}_0^f$ are given.
2. To forecast time step $h + 1$,
   2.1 if observations $\boldsymbol{y}_h$ are available, the analysis and its error variance are computed as

$$\begin{aligned} \mathbf{K}_h &= \mathbf{P}_h^f \mathbf{H}_h^\top (\mathbf{H}_h \mathbf{P}_h^f \mathbf{H}_h^\top + \mathbf{R}_h)^{-1} \\ \boldsymbol{x}_h^a &= \boldsymbol{x}_h^f + \mathbf{K}_h (\boldsymbol{y}_h - \mathbf{H}_h \boldsymbol{x}_h^f) \\ \mathbf{P}_h^a &= (\mathbf{I} - \mathbf{K}_h \mathbf{H}_h) \mathbf{P}_h^f \end{aligned} \tag{19}$$

   And the forecast or prediction follows:

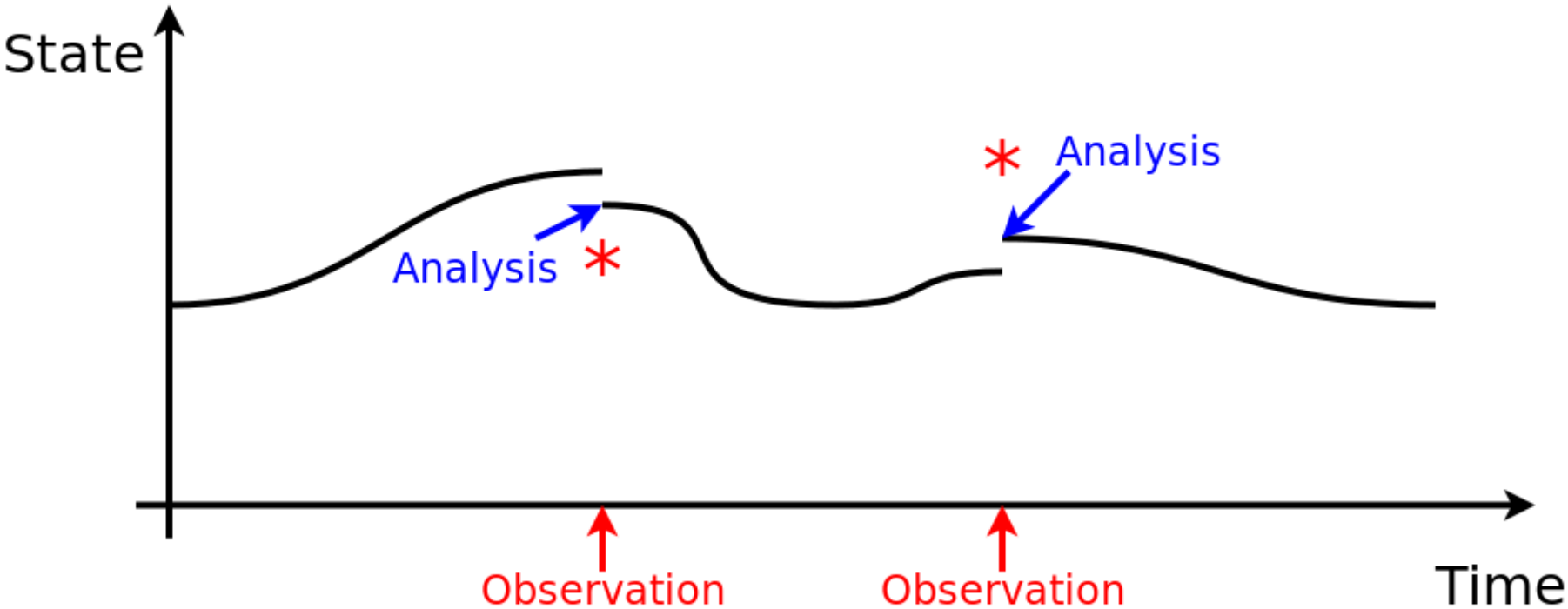
$$\begin{aligned} \boldsymbol{x}_{h+1}^f &= \mathbf{M}_h \boldsymbol{x}_h^a \\ \mathbf{P}_{h+1}^f &= \mathbf{M}_h \mathbf{P}_h^a \mathbf{M}_h^\top + \mathbf{Q}_h \end{aligned} \tag{20}$$

   2.2 if there are no observations, the forecast or prediction step is

$$\begin{aligned} \boldsymbol{x}_{h+1}^f &= \mathbf{M}_h \boldsymbol{x}_h^f \\ \mathbf{P}_{h+1}^f &= \mathbf{M}_h \mathbf{P}_h^f \mathbf{M}_h^\top + \mathbf{Q}_h \end{aligned} \tag{21}$$

Ensemble Kalman filter: time sequence

Time sequence



Kalman filter: a few notes

Link with Bayesian estimation

Under all previous assumptions on the errors, the filter computes the state mean and its error variance. In case the "input" errors are Gaussian, $\boldsymbol{\epsilon}_0^f \sim \mathcal{N}(\mathbf{0}, \mathbf{P}_0^f)$, $\boldsymbol{\epsilon}_h^m \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_h)$, $\boldsymbol{\epsilon}_h^y \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_h)$, the forecast and analysis errors remain Gaussian and the Kalman filter correctly computes their first two moments, $\boldsymbol{\epsilon}_h^f \sim \mathcal{N}(\mathbf{0}, \mathbf{P}_h^f)$ and $\boldsymbol{\epsilon}_h^a \sim \mathcal{N}(\mathbf{0}, \mathbf{P}_h^a)$.

Quick comparison with 4D-Var

Still in the linear case, at the end of the assimilation window, $h = T$, the 4D-Var and the Kalman analyses coincide. At other steps, $h < T$, the Kalman filter provides a state estimate given all observations until h, while the 4D-Var computes the state estimate given all observations in the assimilation window. The Kalman filter directly computes the state error variance. The 4D-Var is well adapted to non-linear models.

Extended Kalman filter

The Kalman filter is derived in the linear case. In practice, it is used for non-linear models as well. The algorithm remains similar, but optimality is lost. The probability density function of the state is probably not Gaussian anymore, even if all "input" errors are.

Non-linear model

The analysis step does not change.

The forecast step makes use of the non-linear model:

$$\boldsymbol{x}_{h+1}^f = \mathcal{M}_h(\boldsymbol{x}_h^a)$$

The main issue is the propagation of error variance, which depends on the model non-linearities. The propagation relies on the linearization of the model. It makes use of the Jacobian of the model, taken at $\boldsymbol{x}_h^a$:

$$\mathbf{M}_h = \left.\frac{d\mathcal{M}_h}{d\boldsymbol{x}}\right|_{\boldsymbol{x}_h^a} \tag{22}$$

Then the propagation remains

$$\mathbf{P}_{h+1}^f = \mathbf{M}_h \mathbf{P}_h^a \mathbf{M}_h^\top + \mathbf{Q}_h$$

Extended Kalman filter

Non-linear observation operator

The observation operator may also be non-linear. This changes the analysis step. The non-linear operator is used in the analysis step:

$$\boldsymbol{x}_h^a = \boldsymbol{x}_h^f + \mathbf{K}_h(\boldsymbol{y}_h - \mathcal{H}_h(\boldsymbol{x}_h^f))$$

The linearized observation operator

$$\mathbf{H}_h = \left.\frac{d\mathcal{H}_h}{d\boldsymbol{x}}\right|_{\boldsymbol{x}_h^f} \tag{23}$$

is used elsewhere:

$$\mathbf{K}_h = \mathbf{P}_h^f \mathbf{H}_h^\top (\mathbf{H}_h \mathbf{P}_h^f \mathbf{H}_h^\top + \mathbf{R}_h)^{-1}$$
$$\mathbf{P}_h^a = (\mathbf{I} - \mathbf{K}_h \mathbf{H}_h)\mathbf{P}_h^f$$

Extended Kalman filter: algorithm

1. At initial time step $h = 0$, the initial condition $\boldsymbol{x}_0^f$ and its error variance $\mathbf{P}_0^f$ are given.
2. To forecast time step $h + 1$,
   2.1 if observations $\boldsymbol{y}_h$ are available, the analysis and its error variance are computed as

$$\begin{aligned} \mathbf{K}_h &= \mathbf{P}_h^f \mathbf{H}_h^\top (\mathbf{H}_h \mathbf{P}_h^f \mathbf{H}_h^\top + \mathbf{R}_h)^{-1} \\ \boldsymbol{x}_h^a &= \boldsymbol{x}_h^f + \mathbf{K}_h (\boldsymbol{y}_h - \mathcal{H}_h(\boldsymbol{x}_h^f)) \\ \mathbf{P}_h^a &= (\mathbf{I} - \mathbf{K}_h \mathbf{H}_h) \mathbf{P}_h^f \end{aligned} \tag{24}$$

   And the forecast or prediction follows:

$$\begin{aligned} \boldsymbol{x}_{h+1}^f &= \mathcal{M}_h(\boldsymbol{x}_h^a) \\ \mathbf{P}_{h+1}^f &= \mathbf{M}_h \mathbf{P}_h^a \mathbf{M}_h^\top + \mathbf{Q}_h \end{aligned} \tag{25}$$

   2.2 if there are no observations, the forecast or prediction step is

$$\begin{aligned} \boldsymbol{x}_{h+1}^f &= \mathcal{M}_h(\boldsymbol{x}_h^f) \\ \mathbf{P}_{h+1}^f &= \mathbf{M}_h \mathbf{P}_h^f \mathbf{M}_h^\top + \mathbf{Q}_h \end{aligned} \tag{26}$$

Parameter estimation or inverse modeling with the Kalman filter

Inverse modeling consists in estimating input parameters $\boldsymbol{u}$ to the model, in addition or instead of the state.

4D-Var is especially well adapted to inverse modeling. It is possible to apply a Kalman filter too, using a so-called augmented state:

$$\widetilde{\boldsymbol{x}_h^f} = \begin{bmatrix} \boldsymbol{x}_h^f \\ \boldsymbol{u} \end{bmatrix} \tag{27}$$

The state equation reads

$$\begin{bmatrix} \boldsymbol{x}_{h+1}^f \\ \boldsymbol{u} \end{bmatrix} = \begin{bmatrix} \mathcal{M}_h(\boldsymbol{x}_h^f, \boldsymbol{u}) \\ \boldsymbol{u} \end{bmatrix} \tag{28}$$

The error variances are augmented as well, e.g., if $\boldsymbol{u}$ is unbiased and using $\boldsymbol{\epsilon}^u = \boldsymbol{u} - \boldsymbol{u}^t$:

$$\widetilde{\mathbf{P}_h^f} = \begin{bmatrix} \mathbf{P}_h^f & \mathrm{E}\left(\boldsymbol{\epsilon}_h^f \boldsymbol{\epsilon}^{u\top}\right) \\ \mathrm{E}\left(\boldsymbol{\epsilon}^u \boldsymbol{\epsilon}_h^{f\top}\right) & \mathbf{P}_h^u \end{bmatrix} \tag{29}$$

and $\widetilde{\mathbf{Q}_h} = \begin{bmatrix} \mathbf{Q}_h & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}$

Kalman filter: computational costs

Kalman filter is too expensive

The Kalman filter is so computationally intensive that it cannot be used for most high-dimensional systems. The showstopper is usually the propagation of the error variance:

$$\mathbf{P}_{h+1}^{f} = \mathbf{M}_h \mathbf{P}_h^a \mathbf{M}_h^\top + \mathbf{Q}_h$$

Assume the state vector is in $\mathbb{R}^{N_x}$ and $N_x = 10^6$. There are 10^6 calls to the linearized model $\mathbf{M}_h$, which is usually intractable.

Many variants of the Kalman filter rely on a different error variance propagation

- Dimension reduction in reduced-rank filters
- Deterministic sampling of the error
- Random sampling of the error

Introduction to data assimilation: references


F. Bouttier and P. Courtier (1999). *Data assimilation concepts and methods*. Meteorological training course lecture series. ECMWF


S. E. Cohn (1997). "An introduction to estimation theory". In: *Journal of the Meteorological Society of Japan* 75.1B, pp. 257–288


Lecture notes in French: "Introduction aux principes et méthodes de l'assimilation de données en géophysique", M. Bocquet — see `http://cerea.enpc.fr/HomePages/bocquet/teaching.html`

Ensemble Kalman filter: principle

Approximate the error variance with state samples

- An ensemble of states is generated and is supposed to sample the errors
- From the ensemble, an empirical error variance is computed
- The empirical error variance replaces the exact error variance $\mathbf{P}_h^f$

Tracking an ensemble of state trajectories

- Ensemble of forecast states at time step h: $\boldsymbol{x}_h^{(m)f}$, $m = 1 \dots M$
- Ensemble of analysis states at time step h: $\boldsymbol{x}_h^{(m)a}$, $m = 1 \dots M$

Ensemble Kalman filter: principle

Forecast and analysis

- The filter forecast and the analysis are taken as the average of all forecasts and analyses

$$\boldsymbol{x}_h^f = \frac{1}{M}\sum_{m=1}^{M} \boldsymbol{x}_h^{(m)f} \qquad \boldsymbol{x}_h^a = \frac{1}{M}\sum_{m=1}^{M} \boldsymbol{x}_h^{(m)a} \tag{30}$$

Empirical error variance

$$\mathbf{P}_h^f = \frac{1}{M-1}\sum_{m=1}^{M} \left(\boldsymbol{x}_h^{(m)f} - \boldsymbol{x}_h^f\right)\left(\boldsymbol{x}_h^{(m)f} - \boldsymbol{x}_h^f\right)^\top \tag{31}$$

$$\mathbf{P}_h^a = \frac{1}{M-1}\sum_{m=1}^{M} \left(\boldsymbol{x}_h^{(m)a} - \boldsymbol{x}_h^a\right)\left(\boldsymbol{x}_h^{(m)a} - \boldsymbol{x}_h^a\right)^\top \tag{32}$$

Ensemble Kalman filter: error variance

Classical formula:

$$\mathbf{P}_h^f = \frac{1}{M-1} \sum_{m=1}^{M} \left(\boldsymbol{x}_h^{(m)f} - \boldsymbol{x}_h^f \right) \left(\boldsymbol{x}_h^{(m)f} - \boldsymbol{x}_h^f \right)^\top \tag{33}$$

A formula in square-root form:

$$\mathbf{P}_h^{f\,1/2} = \frac{1}{\sqrt{M-1}} \left[\begin{array}{ccc} \boldsymbol{x}_h^{(1)f} - \boldsymbol{x}_h^f & \cdots & \boldsymbol{x}_h^{(M)f} - \boldsymbol{x}_h^f \end{array} \right] \in \mathbb{R}^{N_x \times M} \tag{34}$$

$$\mathbf{P}_h^f = \mathbf{P}_h^{f\,1/2} \mathbf{P}_h^{f\,\top/2} \in \mathbb{R}^{N_x \times N_x} \tag{35}$$

Ensemble Kalman filter: error variance

Classical formula:

$$\mathbf{P}_h^f = \frac{1}{M-1} \sum_{m=1}^{M} \left(\boldsymbol{x}_h^{(m)f} - \boldsymbol{x}_h^f \right) \left(\boldsymbol{x}_h^{(m)f} - \boldsymbol{x}_h^f \right)^\top \tag{33}$$

A formula in square-root form:

$$\mathbf{P}_h^{f\,1/2} = \frac{1}{\sqrt{M-1}} \left[\begin{array}{ccc} \boldsymbol{x}_h^{(1)f} - \boldsymbol{x}_h^f & \cdots & \boldsymbol{x}_h^{(M)f} - \boldsymbol{x}_h^f \end{array} \right] \in \mathbb{R}^{N_x \times M} \tag{34}$$

$$\mathbf{P}_h^f = \mathbf{P}_h^{f\,1/2} \mathbf{P}_h^{f\,\top/2} \in \mathbb{R}^{N_x \times N_x} \tag{35}$$

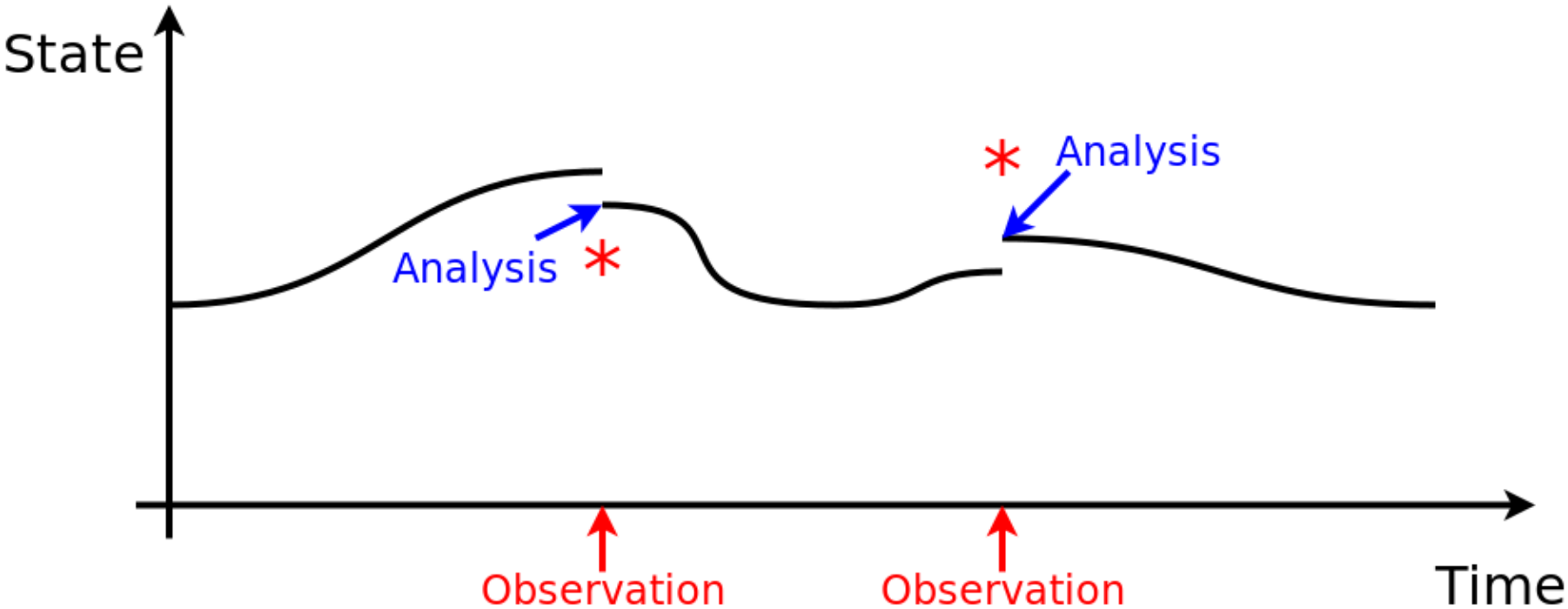
The linearized observation operator is not needed anymore, with the following approximation:

$$\mathbf{P}_h^f \mathbf{H}_h^\top = \frac{1}{M-1} \sum_{m=1}^{M} \left(\boldsymbol{x}_h^{(m)f} - \boldsymbol{x}_h^f \right) \left(\mathcal{H}_h(\boldsymbol{x}_h^{(m)f}) - \frac{1}{M} \sum_{m'=1}^{M} \mathcal{H}_h(\boldsymbol{x}_h^{(m')f}) \right)^\top \tag{36}$$

$$\mathbf{H}_h \mathbf{P}_h^f \mathbf{H}_h^\top = \frac{1}{M-1} \sum_{m=1}^{M} \left(\mathcal{H}_h(\boldsymbol{x}_h^{(m)f}) - \frac{1}{M} \sum_{m'=1}^{M} \mathcal{H}_h(\boldsymbol{x}_h^{(m')f}) \right) \left(\mathcal{H}_h(\boldsymbol{x}_h^{(m)f}) - \frac{1}{M} \sum_{m'=1}^{M} \mathcal{H}_h(\boldsymbol{x}_h^{(m')f}) \right)^\top \tag{37}$$

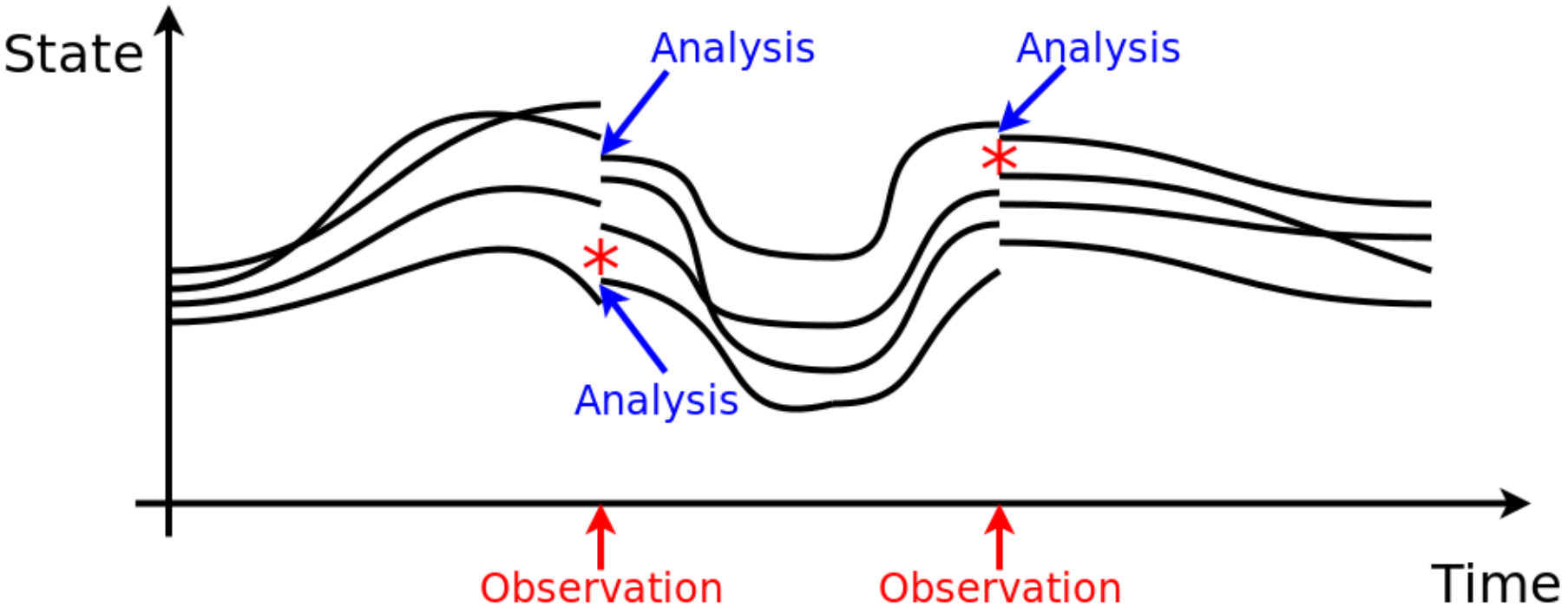
Ensemble Kalman filter: time sequence

Time sequence for a classical extended Kalman filter



Ensemble Kalman filter: time sequence

Time sequence for the ensemble Kalman filter



Ensemble Kalman filter: algorithm

1. At initial time step $h = 0$, an ensemble of initial conditions $\boldsymbol{x}_0^{(m)f}$ is given.
2. To forecast time step $h + 1$,
   2.1 if observations $\boldsymbol{y}_h$ are available, the analysis step is

$$\begin{aligned} \mathbf{K}_h &= \mathbf{P}_h^f \mathbf{H}_h^\top (\mathbf{H}_h \mathbf{P}_h^f \mathbf{H}_h^\top + \mathbf{R}_h)^{-1} \\ \boldsymbol{x}_h^{(m)a} &= \boldsymbol{x}_h^{(m)f} + \mathbf{K}_h (\boldsymbol{y}_h - \mathcal{H}_h(\boldsymbol{x}_h^{(m)f})) \end{aligned} \tag{38}$$

   where $\mathbf{K}_h$ is computed using (36) and (37).
   The forecast or prediction step is:

$$\boldsymbol{x}_{h+1}^{(m)f} = \mathcal{M}_h(\boldsymbol{x}_h^{(m)a}) \tag{39}$$

   2.2 if there are no observations, the forecast or prediction step is

$$\boldsymbol{x}_{h+1}^{(m)f} = \mathcal{M}_h(\boldsymbol{x}_h^{(m)f}) \tag{40}$$

Ensemble Kalman filter: algorithm with perturbed observations

1. At initial time step $h = 0$, an ensemble of initial conditions $\boldsymbol{x}_0^{(m)f}$ is given.
2. To forecast time step $h + 1$,

   2.1 if observations $\boldsymbol{y}_h$ are available, the analysis step is

$$\begin{aligned} \mathbf{K}_h &= \mathbf{P}_h^f \mathbf{H}_h^\top (\mathbf{H}_h \mathbf{P}_h^f \mathbf{H}_h^\top + \mathbf{R}_h)^{-1} \\ \boldsymbol{x}_h^{(m)a} &= \boldsymbol{x}_h^{(m)f} + \mathbf{K}_h (\boldsymbol{y}_h + \boldsymbol{\delta}_h^{(m)} - \mathcal{H}_h(\boldsymbol{x}_h^{(m)f})) \end{aligned} \tag{41}$$

   where $\mathbf{K}_h$ is computed using (36) and (37), and $\boldsymbol{\delta}_h^{(m)} \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_h)$. The forecast or prediction step is:

$$\boldsymbol{x}_{h+1}^{(m)f} = \mathcal{M}_h(\boldsymbol{x}_h^{(m)a}) \tag{42}$$

   2.2 if there are no observations, the forecast or prediction step is

$$\boldsymbol{x}_{h+1}^{(m)f} = \mathcal{M}_h(\boldsymbol{x}_h^{(m)f}) \tag{43}$$

Ensemble Kalman filter with perturbed observations

Why perturbations on the observations?

Let us consider the linear case and check whether the ensemble of analyses has the proper error variance, that is (9):

$$\mathbf{P}_h^a = (\mathbf{I} - \mathbf{K}_h\mathbf{H}_h)\mathbf{P}_h^f(\mathbf{I} - \mathbf{K}_h\mathbf{H}_h)^\top + \mathbf{K}_h\mathbf{R}_h\mathbf{K}_h^\top$$

We have

$$\boldsymbol{x}_h^{(m)a} = (\mathbf{I} - \mathbf{K}_h\mathbf{H}_h)\boldsymbol{x}_h^{(m)f} + \mathbf{K}_h(\boldsymbol{y}_h + \boldsymbol{\delta}_h^{(m)}) \tag{44}$$

With $\boldsymbol{\delta}_h^{(m)} = \mathbf{0}$, we obtain

$$\mathbf{P}_h^a = (\mathbf{I} - \mathbf{K}_h\mathbf{H}_h)\mathbf{P}_h^f(\mathbf{I} - \mathbf{K}_h\mathbf{H}_h)^\top \tag{45}$$

With $\boldsymbol{\delta}_h^{(m)} \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_h)$ and discarding correlations between $\boldsymbol{\delta}_h^{(m)}$ and $\boldsymbol{x}_h^{(m)f} - \boldsymbol{x}_h^f$, we indeed obtain (9):

$$\mathbf{P}_h^a = (\mathbf{I} - \mathbf{K}_h\mathbf{H}_h)\mathbf{P}_h^f(\mathbf{I} - \mathbf{K}_h\mathbf{H}_h)^\top + \mathbf{K}_h\mathbf{R}_h\mathbf{K}_h^\top$$

Ensemble Kalman filter: computational costs at forecast step

The total computational costs of the forecast steps are M times the cost of one simulation. In geosciences, the typical size of the ensemble is $M = 30$–100.

Each forecast step from the ensemble can be carried out independently of the others, which greatly eases the distribution of the computations. Btu the available number of cores on a HPC architecture is likely to be higher, in which case parallelization of the individual members of the ensemble is also necessary.

Ensemble Kalman filter: computational costs at analysis step

The computational cost of the analysis step depends on

- The cost of the observation operator,
- The size M of the ensemble,
- The number of observations, because of $(\mathbf{H}_h \mathbf{P}_h^f \mathbf{H}_h^\top + \mathbf{R}_h)^{-1}$.

The forecast and analysis error variances are never computed explicitly. Their square root form is used in the computations.

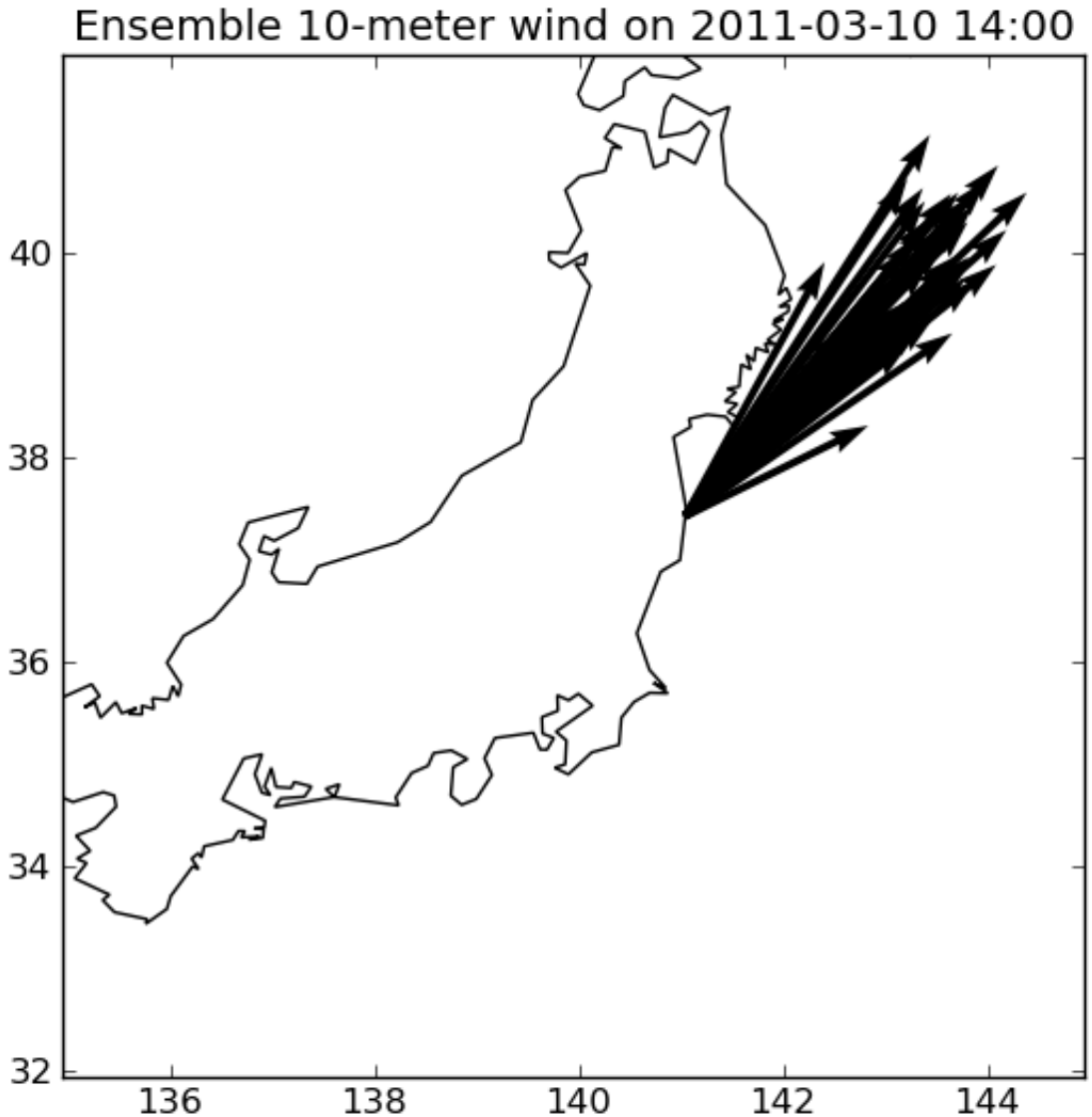
The analysis step couples all ensemble members through $\mathbf{P}_h^f$ and therefore $\mathbf{K}_h$.

Ensemble Kalman filter: the generation of the ensembles

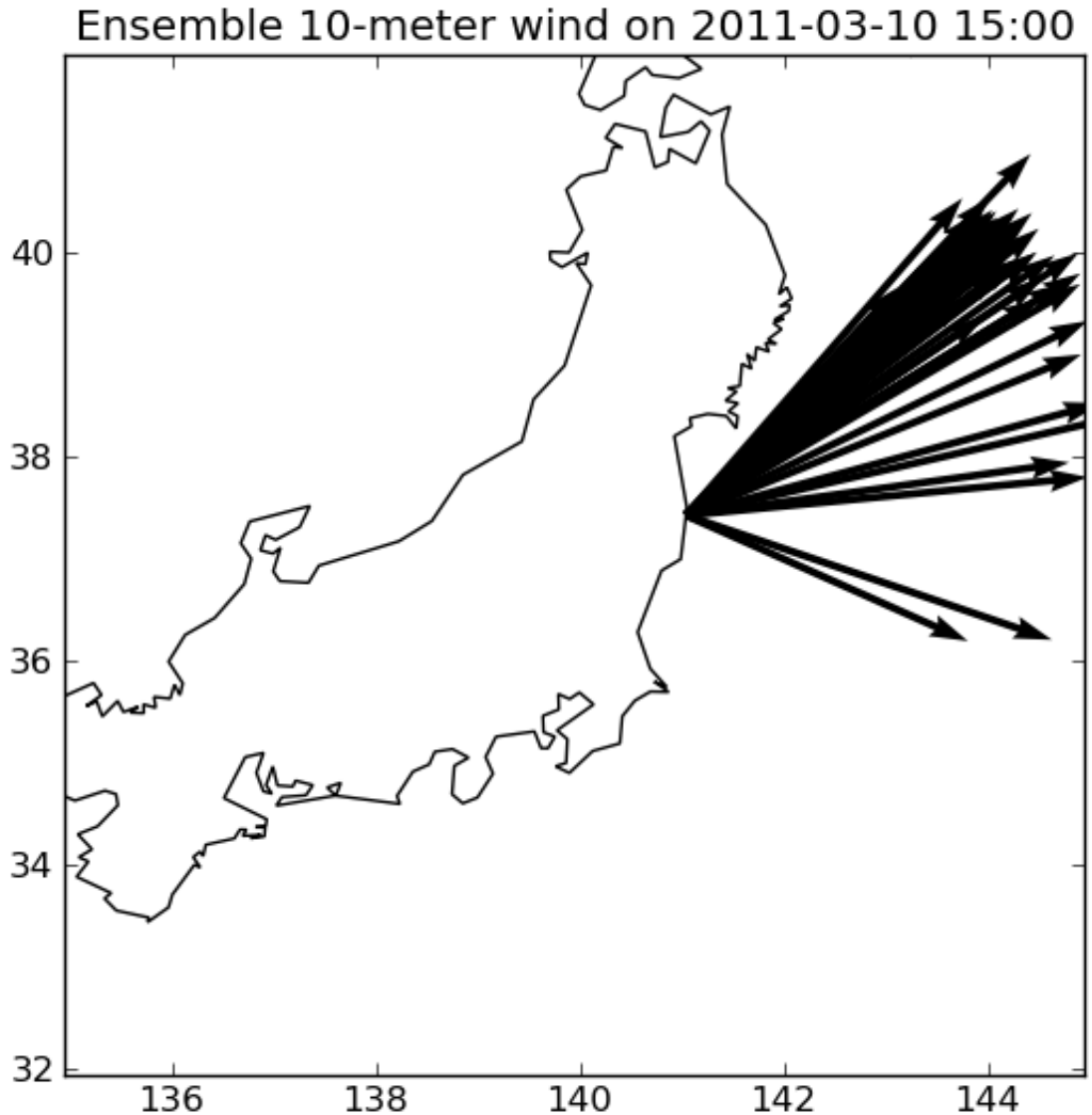
The ensemble should sample the errors. There are at least three error sources:

1. The input data to the model, which can be a time-varying vector $\boldsymbol{u}_h$ of parameters;
2. The biological/chemical/physical model is imperfect, e.g., may rely on empirical parameterizations related to unresolved phenomena;
3. The numerical solver introduces errors, which are unknown and therefore contribute to the global uncertainty.

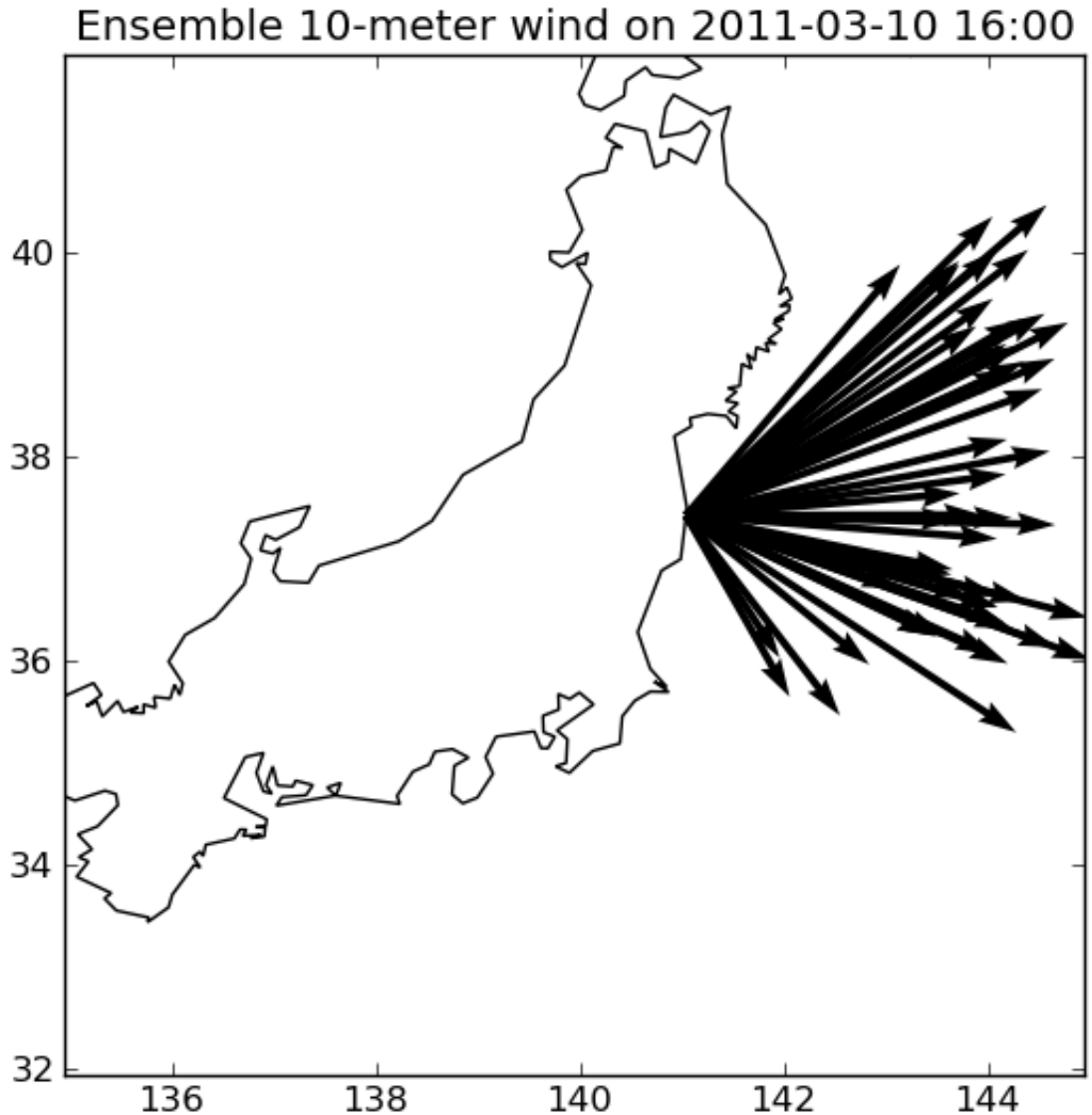
Ensemble simulations of wind at Fukushima



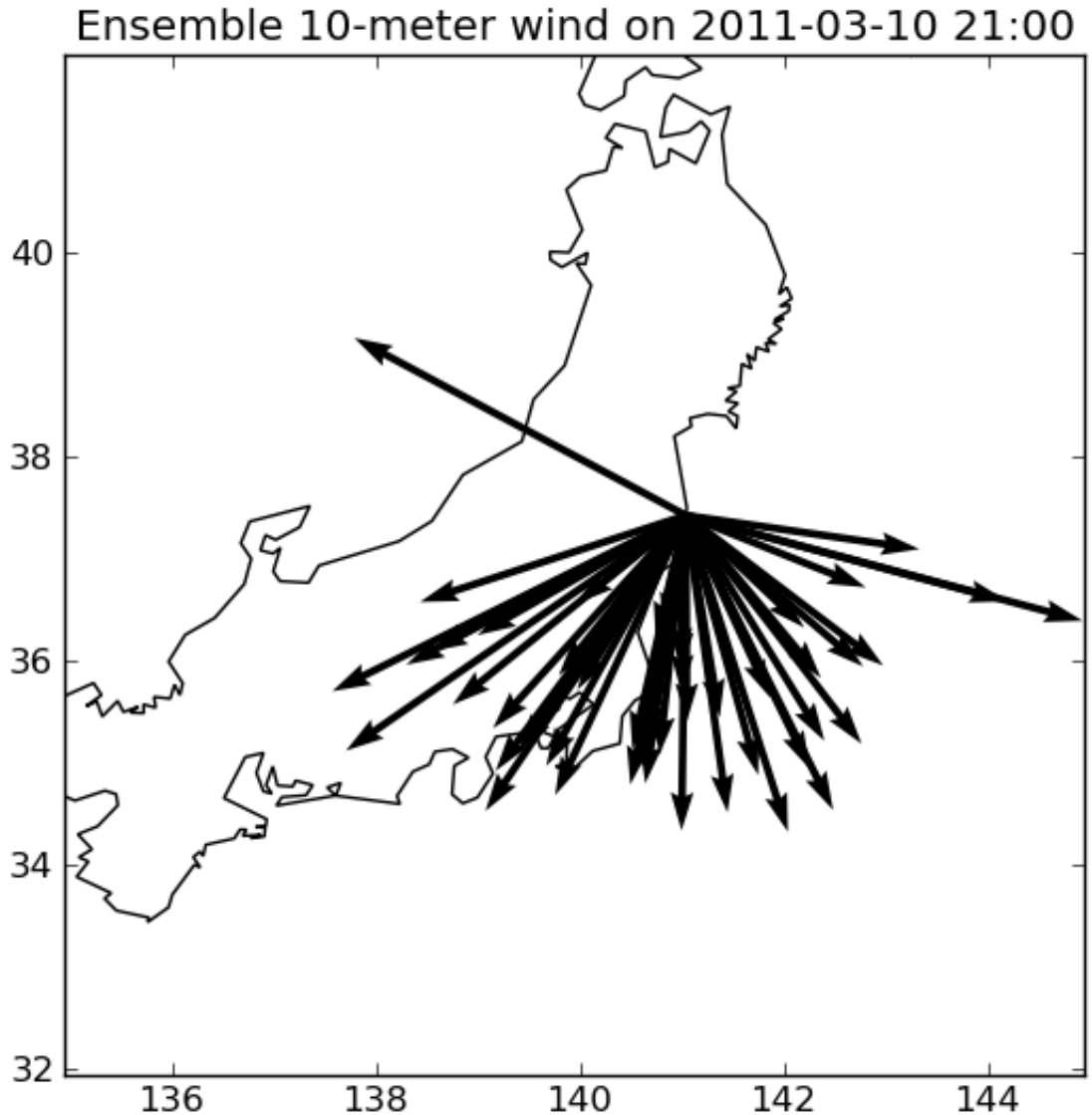
Ensemble simulations of wind at Fukushima



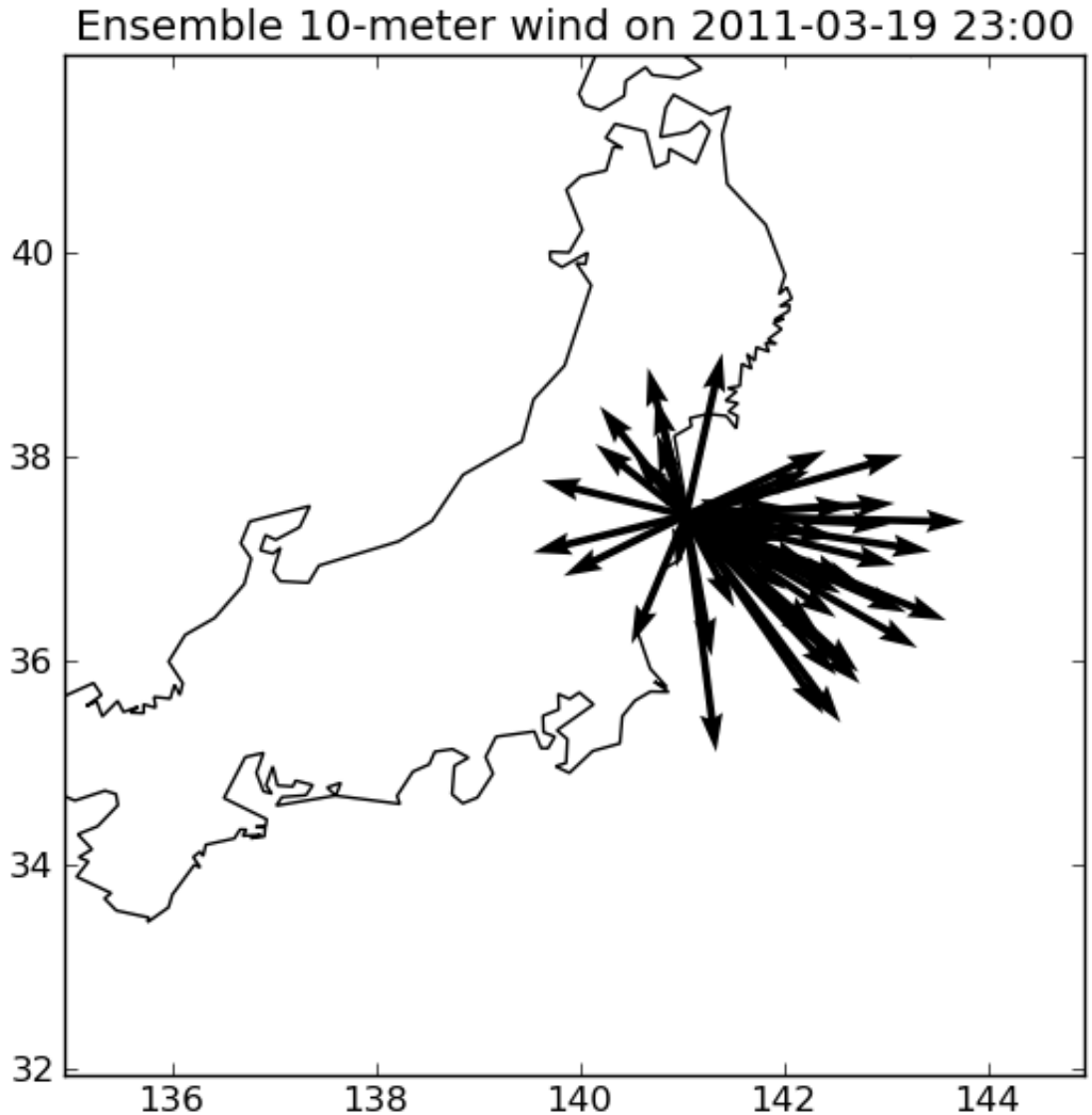
Ensemble simulations of wind at Fukushima



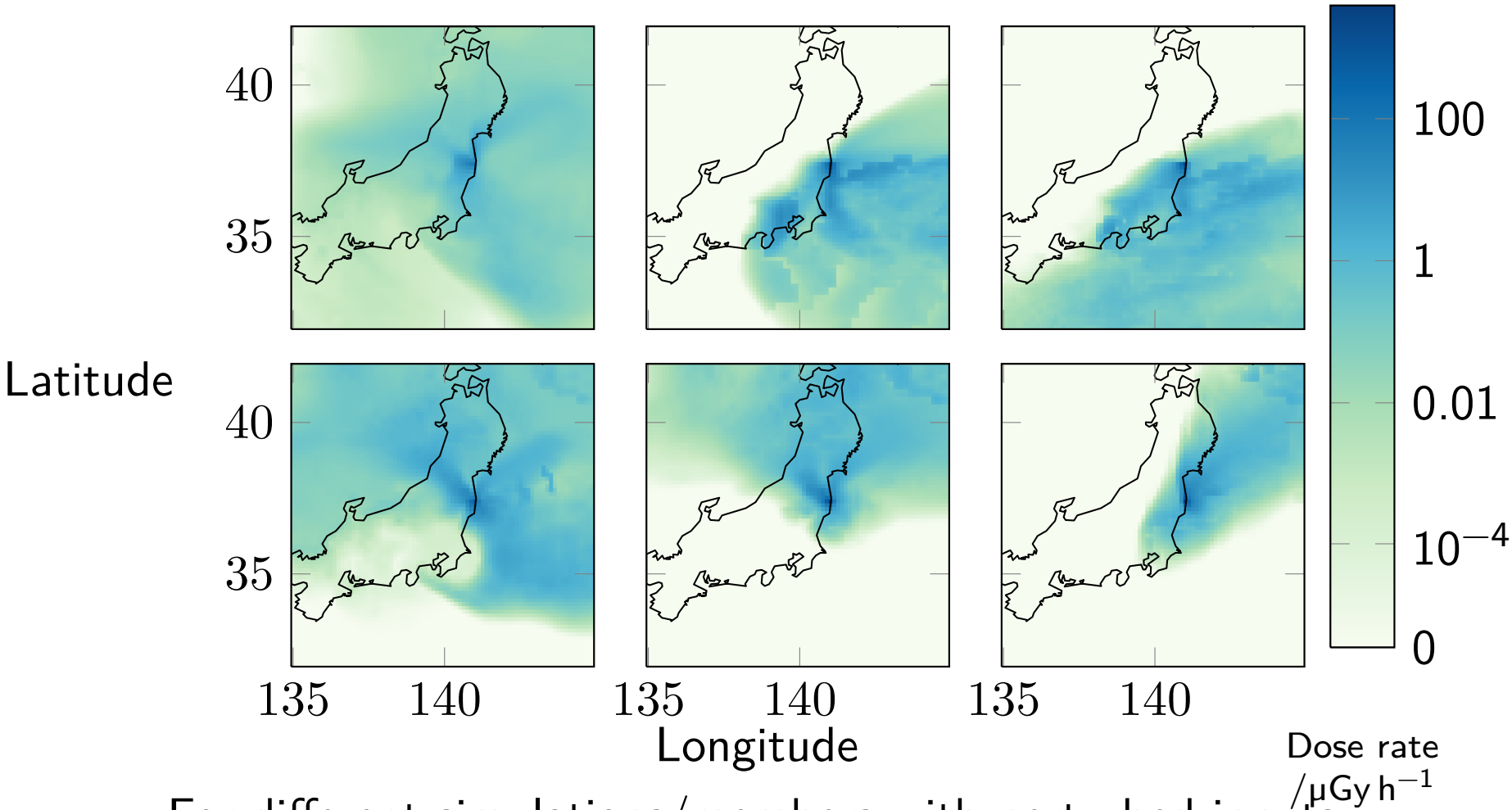
Ensemble simulations of wind at Fukushima



Ensemble simulations of wind at Fukushima



Maximum of gamma dose rate from the deposit, during Fukushima disaster



For different simulations/members with perturbed inputs.

Ensemble Kalman filter: algorithm with perturbed observations and imperfect model

1. At initial time step $h = 0$, an ensemble of initial conditions $\boldsymbol{x}_0^{(m)f}$ is given.
2. To forecast time step $h + 1$,

   2.1 if observations $\boldsymbol{y}_h$ are available, the analysis step is

$$\begin{aligned} \mathbf{K}_h &= \mathbf{P}_h^f \mathbf{H}_h^\top (\mathbf{H}_h \mathbf{P}_h^f \mathbf{H}_h^\top + \mathbf{R}_h)^{-1} \\ \boldsymbol{x}_h^{(m)a} &= \boldsymbol{x}_h^{(m)f} + \mathbf{K}_h (\boldsymbol{y}_h + \boldsymbol{\delta}_h^{(m)} - \mathcal{H}_h(\boldsymbol{x}_h^{(m)f})) \end{aligned} \tag{46}$$

   where $\mathbf{K}_h$ is computed using (36) and (37), and $\boldsymbol{\delta}_h^{(m)} \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_h)$. The forecast or prediction step is:

$$\boldsymbol{x}_{h+1}^{(m)f} = \mathcal{M}_h^{(m)}(\boldsymbol{x}_h^{(m)a}, \boldsymbol{u}_h^{(m)}) \tag{47}$$

   2.2 if there are no observations, the forecast or prediction step is

$$\boldsymbol{x}_{h+1}^{(m)f} = \mathcal{M}_h^{(m)}(\boldsymbol{x}_h^{(m)f}, \boldsymbol{u}_h^{(m)}) \tag{48}$$

Ensemble Kalman filter: references

G. Evensen (1994). "Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics". In: *Journal of Geophysical Research* 99.C5, pp. 10,143–10,162

G. Evensen (2003). "The Ensemble Kalman Filter: theoretical formulation and practical implementation". In: *Ocean Dynamics* 53, pp. 343–367. DOI: 10.1007/s10236-003-0036-9

Ensemble Kalman filter: example with a shallow water model

Time evolution of the water height $w(x)$ and the horizontal velocity $v(x)$:

$$\begin{cases} \partial_t w + \partial_x(wv) &= 0\,, \\ \partial_t(wv) + \partial_x(wv^2) + \frac{1}{2}g\partial_x w^2 &= 0\,. \end{cases} \tag{49}$$

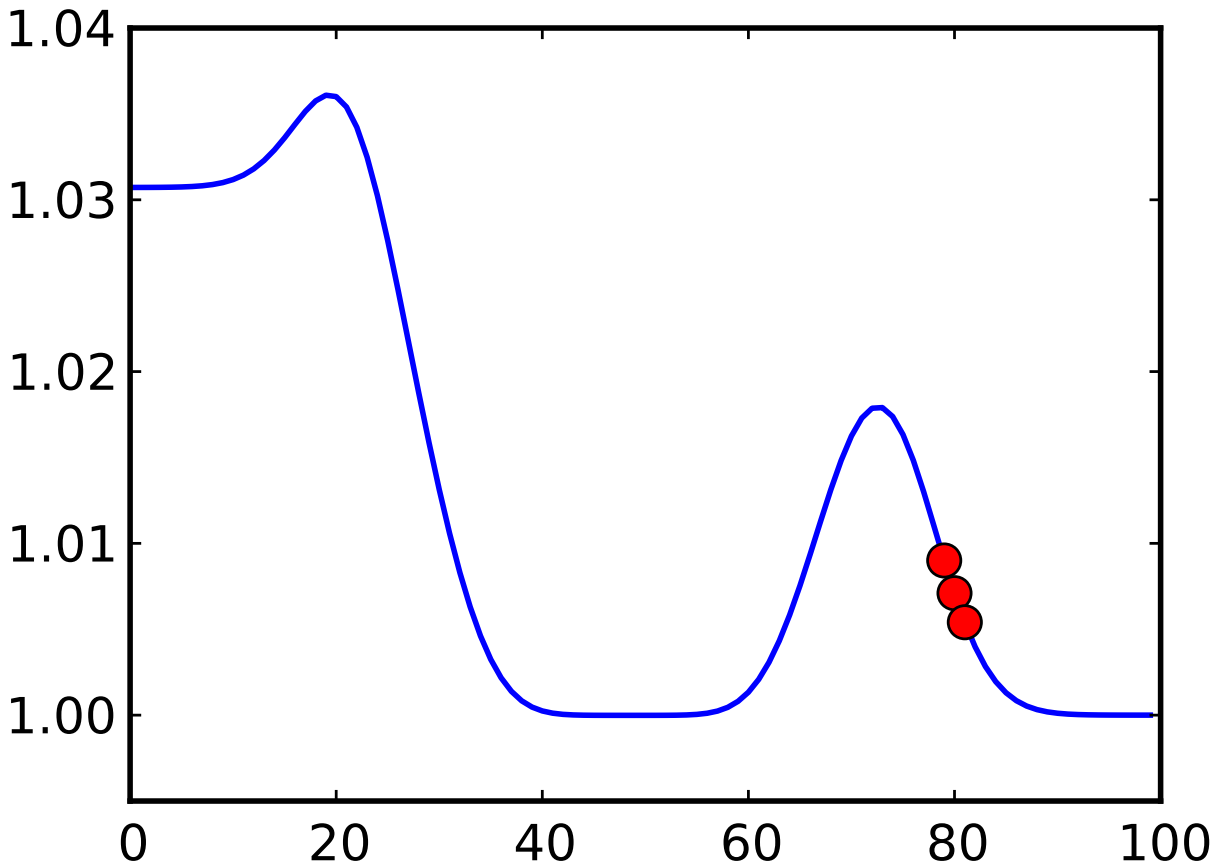
Boundary conditions:

1. on the left, a constant inflow $q_b = w_b v_b$;
2. on the right, a homogeneous Neumann condition for (w, v).

The simulation domain is made of 100 grid points. We show, observe and assimilate only water height—not the velocity.

The true state starts from a step function.

Ensemble Kalman filter: example with a shallow water model



Location of the observations.

Local ensemble transform Kalman filter (LETKF)

First idea: rewrite the analysis in the space spanned by the ensemble

Consider the analysis in the ensemble Kalman filter:

$$\boldsymbol{x}_h^{(m)a} = \boldsymbol{x}_h^{(m)f} + \mathbf{K}_h(\boldsymbol{y}_h - \mathcal{H}_h(\boldsymbol{x}_h^{(m)f}))$$

$$\mathbf{K}_h = \mathbf{P}_h^f \mathbf{H}_h^\top (\mathbf{H}_h \mathbf{P}_h^f \mathbf{H}_h^\top + \mathbf{R}_h)^{-1}$$

and

$$\mathbf{P}_h^{f\,1/2} = \frac{1}{\sqrt{M-1}} \left[\begin{array}{ccc} \boldsymbol{x}_h^{(1)f} - \boldsymbol{x}_h^f & \cdots & \boldsymbol{x}_h^{(M)f} - \boldsymbol{x}_h^f \end{array} \right] \in \mathbb{R}^{N_x \times M}$$

Notice that the correction on the forecast $(\boldsymbol{x}_h^{(m)a} - \boldsymbol{x}_h^{(m)f})$ is in the space spanned by the perturbations $\boldsymbol{x}_h^{(m)f} - \boldsymbol{x}_h^f$. LETKF makes use of this property to reformulate the analysis expression.

Local ensemble transform Kalman filter (LETKF)

First idea: rewrite the analysis in the space spanned by the ensemble

Departures (in columns) from the mean:

$$\mathbf{X}_h^f = \left(\boldsymbol{x}_h^{(1)f} - \boldsymbol{x}_h^f \ \cdots \ \boldsymbol{x}_h^{(M)f} - \boldsymbol{x}_h^f \right) \tag{50}$$

$$\mathbf{X}_h^a = \left(\boldsymbol{x}_h^{(1)a} - \boldsymbol{x}_h^a \ \cdots \ \boldsymbol{x}_h^{(M)a} - \boldsymbol{x}_h^a \right) \tag{51}$$

Similarly, at observed locations:

$$\mathbf{Y}_h^f = \left(\mathcal{H}_h(\boldsymbol{x}_h^{(1)f}) - \boldsymbol{y}_h^f \ \cdots \ \mathcal{H}_h(\boldsymbol{x}_h^{(M)f}) - \boldsymbol{y}_h^f \right) \tag{52}$$

where

$$\boldsymbol{y}_h^f = \frac{1}{M} \sum_{m=1}^{M} \mathcal{H}_h(\boldsymbol{x}_h^{(m)f}) \tag{53}$$

Local ensemble transform Kalman filter (LETKF)

First idea: rewrite the analysis in the space spanned by the ensemble

Intermediate matrix:

$$\mathbf{N}_h = \left((M-1)\mathbf{I} + \mathbf{Y}_h^{f\top} \mathbf{R}_h^{-1} \mathbf{Y}_h^f \right)^{-1} \tag{54}$$

Weight vector:

$$\boldsymbol{w}_h^a = \mathbf{N}_h \mathbf{Y}_h^{f\top} \mathbf{R}_h^{-1} (\boldsymbol{y}_h - \boldsymbol{y}_h^f) \tag{55}$$

Analysis (mean) and its perturbations:

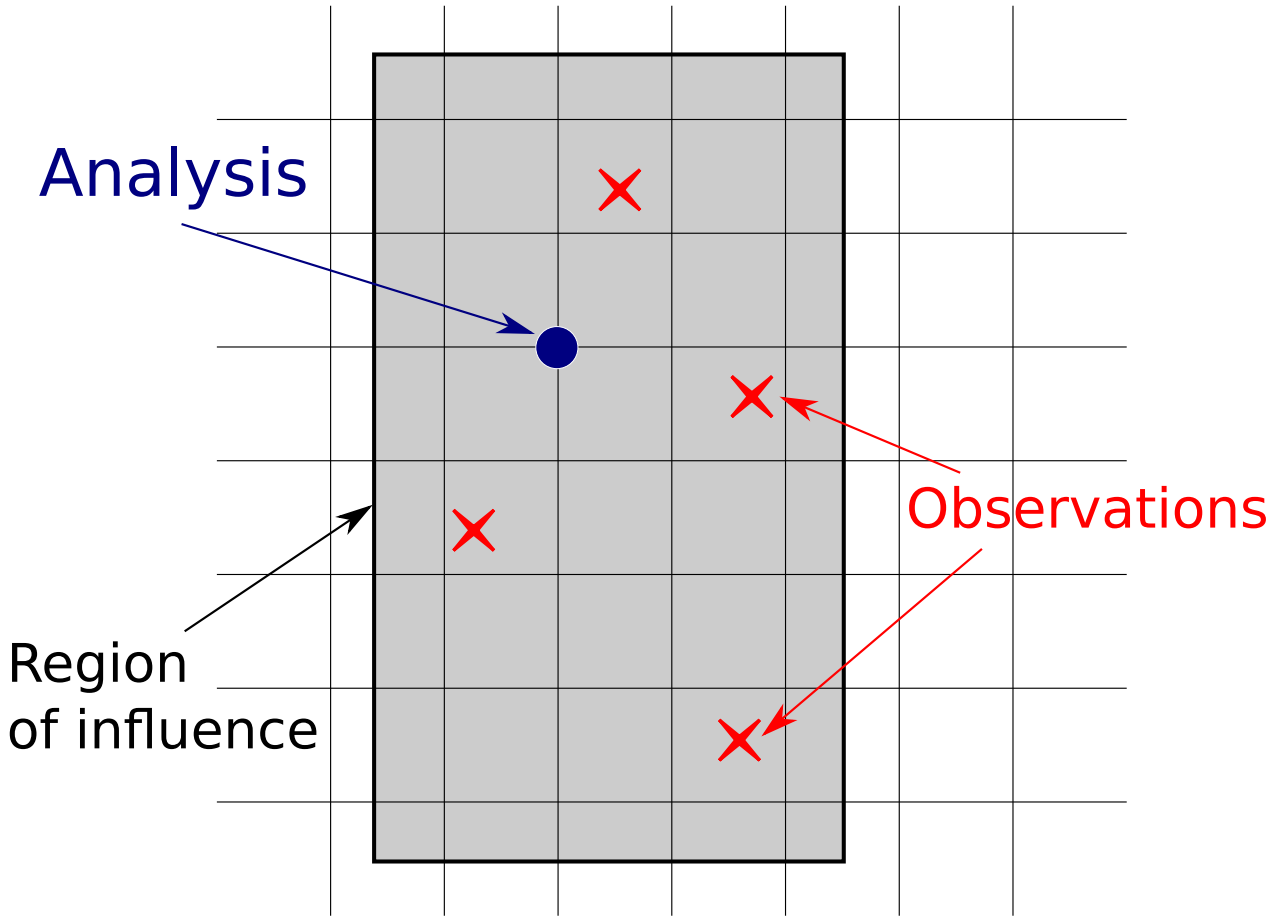
$$\boldsymbol{x}_h^a = \boldsymbol{x}_h^f + \mathbf{X}_h^f \boldsymbol{w}_h^a \tag{56}$$

$$\mathbf{X}_h^a = \sqrt{M-1}\mathbf{X}_h^f \mathbf{N}_h^{\frac{1}{2}} \tag{57}$$

Local ensemble transform Kalman filter (LETKF)

Second idea: compute the analysis for one state component at a time

- Include all observations that can have a significant influence on the analysis, because of significant forecast error covariance.



Local ensemble transform Kalman filter (LETKF): reference

B. R. Hunt, E. J. Kostelich, and I. Szunyogh (2007). “Efficient data assimilation for spatiotemporal chaos: A local ensemble transform Kalman filter”. In: *Physica D: Nonlinear Phenomena* 230, pp. 112–126

Unscented Kalman filter: deterministic filter

Deterministic filter

- The error variance is approximated with an "ensemble" of states that are deterministically chosen.
- The "ensemble" members are chosen so that they catch the transformation of the error variance due to non-linearity of the model.
- The mean $\boldsymbol{x}^f$ and the error variance $\mathbf{P}^f$ are replaced by empirical mean and empirical variance derived from these so-called $2N_x + 1$ sigma vectors:

$$
\begin{aligned}
\boldsymbol{x}^{(0)f} &= \boldsymbol{x}^f \\
\boldsymbol{x}^{(m)f} &= \boldsymbol{x}^f + \sqrt{(N_x + \lambda)\mathbf{P}^f}\Big|_m \qquad m = 1, \ldots, N_x \\
\boldsymbol{x}^{(m)f} &= \boldsymbol{x}^f - \sqrt{(N_x + \lambda)\mathbf{P}^f}\Big|_{m-N_x} \qquad m = N_x + 1, \ldots, 2N_x
\end{aligned}
\tag{58}
$$

where $\lambda = \alpha^2(N_x + \kappa) + N_x$, and $\cdot|_m$ refers to the mth column.

Unscented Kalman filter: use of sigma vectors

- The empirical mean is computed with the weights $w_0^m = \frac{\lambda}{N_x+\lambda}$ and $w_m^m = \frac{1}{2(\lambda+N_x)}$ for $m \leq 1$.
  - We can verify that

$$\boldsymbol{x}^f = \sum_{m=0}^{2N_x} w_m^m \boldsymbol{x}^{(m)f}$$

- The empirical error variance is computed with the weights $w_0^P = \frac{\lambda}{N_x+\lambda} + 1 - \alpha^2 + \beta$ and $w_m^m = \frac{1}{2(\lambda+N_x)}$ for $m \leq 1$.
  - We can verify that

$$\mathbf{P}^f = \sum_{m=0}^{2N_x} w_m^P \left(\boldsymbol{x}^{(m)f} - \boldsymbol{x}^f\right)\left(\boldsymbol{x}^{(m)f} - \boldsymbol{x}^f\right)^\top$$

Unscented Kalman filter: why using the sigma vectors?

Better approximations of mean and error variance in non-linear case

- The introduction of the sigma vectors (or sigma points) is called the unscented transform.
- The model is applied on these sigma vectors, and the empirical mean and error variance of the resulting "ensemble" of vectors are better approximations of the actual mean and error variance than those of the extended Kalman filter.
- Extended Kalman filter uses a first-order approximation at propagation stage, while the unscented Kalman filter enjoys a second-order approximation.

Unscented Kalman filter: algorithm (1/2)

1. At initial time step $h = 0$, $2N_x + 1$ sigma vectors $\boldsymbol{x}_0^{(m)f}$ are derived from the initial condition $\boldsymbol{x}_0^f$ and its error variance $\mathbf{P}_0^f$ (see the forecast step below).
2. The analysis step at h reads

$$\boldsymbol{y}_h^f = \sum_{m=0}^{2N_x} w_m^m \mathcal{H}_h(\boldsymbol{x}_h^{(m)f}) \tag{59}$$

The gain is given by

$$\mathbf{P}_h^{xy} = \sum_{m=0}^{2N_x} w_m^P \left(\boldsymbol{x}_h^{(m)f} - \boldsymbol{x}_h^f\right)\left(\mathcal{H}_h(\boldsymbol{x}_h^{(m)f}) - \boldsymbol{y}_h^f\right)^\top$$

$$\mathbf{P}_h^{yy} = \sum_{m=0}^{2N_x} w_m^P \left(\mathcal{H}_h(\boldsymbol{x}_h^{(m)f}) - \boldsymbol{y}_h^f\right)\left(\mathcal{H}_h(\boldsymbol{x}_h^{(m)f}) - \boldsymbol{y}_h^f\right)^\top \tag{60}$$

$$\mathbf{K}_h = \mathbf{P}_h^{xy}\mathbf{P}_h^{yy-1}$$

The analysis and its error variance are

$$\boldsymbol{x}_h^a = \boldsymbol{x}_h^f + \mathbf{K}_h(\boldsymbol{y}_h - \boldsymbol{y}_h^f) \tag{61}$$

$$\mathbf{P}_h^a = \mathbf{P}_h^f - \mathbf{K}_h\mathbf{P}_h^{yy}\mathbf{K}_h^\top \tag{62}$$

Unscented Kalman filter: algorithm (2/2)

1. To forecast time step $h+1$,
   1.1 $2N_x+1$ sigma vectors $\boldsymbol{x}_h^{(m)a}$ are defined as

$$\begin{aligned}
\boldsymbol{x}_h^{(0)a} &= \boldsymbol{x}_h^a \\
\boldsymbol{x}_h^{(m)a} &= \boldsymbol{x}_h^a + \sqrt{(N_x+\lambda)\mathbf{P}_h^a} \quad m = 1,\ldots,N_x \\
\boldsymbol{x}_h^{(m)a} &= \boldsymbol{x}_h^a - \sqrt{(N_x+\lambda)\mathbf{P}_h^a} \quad m = N_x+1,\ldots,2N_x
\end{aligned} \tag{63}$$

   where $\lambda = \alpha^2(N_x+\kappa)+N_x$.

   1.2 The propagation is carried out for all sigma vectors:

$$\boldsymbol{x}_{h+1}^{(m)f} = \mathcal{M}_h(\boldsymbol{x}_h^{(m)a}) \tag{64}$$

   1.3 The forecast mean and its error variance are taken as

$$\begin{aligned}
\boldsymbol{x}_{h+1}^f &= \sum_{m=0}^{2N_x} w_m^m \boldsymbol{x}_{h+1}^{(m)f} \\
\mathbf{P}_{h+1}^f &= \sum_{m=0}^{2N_x} w_m^P \left(\boldsymbol{x}_{h+1}^{(m)f} - \boldsymbol{x}_{h+1}^f\right)\left(\boldsymbol{x}_{h+1}^{(m)f} - \boldsymbol{x}_{h+1}^f\right)^\top + \mathbf{Q}_h
\end{aligned} \tag{65}$$

Unscented Kalman filter: references

S. J. Julier and J. K. Uhlmann (1997). "A new extension of the Kalman filter to nonlinear systems". In: *Proceedings of AeroSense: The 11th International Symposium on Aerospace/Defense Sensing, Simulation and Controls*

P. Moireau and D. Chapelle (2010). "Reduced-order unscented Kalman filtering with application to parameter identification in large-dimensional systems". In: *ESAIM: Control, Optimisation and Calculus of Variations*. DOI: 10.1051/cocv/2010006

Kalman filter

Ensemble Kalman filter

Other filters

- Local ensemble transform Kalman filter
- Unscented Kalman filter
- Reduced-rank Kalman filter

Generation and evaluation of ensembles

Advanced methods with ensembles

- Bayesian approach
- Meta-modeling with reduction and emulation
- Ensemble forecast with sequential aggregation

Reduced-rank square-root Kalman filter

Reduced Kalman filters

- The dimension reduction is applied to the error variance—reduced-rank Kalman filters.
- Dimension reduction directly on the state might be a way, but one needs to be carefully with respect to stability, at least.

Variance reduction and propagation

- The main idea is to write the error variance in reduced form, e.g., in square-root form with low-rank square-root

$$\mathbf{P}_h^a \simeq \mathbf{S}_h^a \mathbf{S}_h^{a\top} \tag{66}$$

  where $\mathbf{P}_h^a \in \mathbb{R}^{N_x \times N_x}$ and $\mathbf{S}_h^a \in \mathbb{R}^{N_x \times r}$ with $r \ll N_x$.
- Then the problematic variance propagation becomes tractable, e.g., without model error:

$$\mathbf{P}_{h+1}^f \simeq \mathbf{M}_h \mathbf{S}_h^a \mathbf{S}_h^{a\top} \mathbf{M}_h^\top \tag{67}$$

  or

$$\mathbf{S}_{h+1}^f = \mathbf{M}_h \mathbf{S}_h^a \tag{68}$$

Reduced-rank square-root Kalman filter: one algorithm

1. At initial time step $h = 0$, the state estimator $\boldsymbol{x}_0^f$ is given with the (reduced) square root $\mathbf{S}_0^f$ of its error variance.
2. The analysis step at h reads
$$\begin{aligned} \mathbf{K}_h &= \mathbf{S}_h^f \mathbf{S}_h^{f\top} \mathbf{H}_h^\top (\mathbf{H}_h \mathbf{S}_h^f \mathbf{S}_h^{f\top} \mathbf{H}_h^\top + \mathbf{R}_h)^{-1} \\ \boldsymbol{x}_h^a &= \boldsymbol{x}_h^f + \mathbf{K}_h (\boldsymbol{y}_h - \mathcal{H}_h(\boldsymbol{x}_h^f)) \end{aligned} \tag{69}$$
The square root of the analysis error variance is
$$\widetilde{\mathbf{S}_h^a} = \left[(\mathbf{I} - \mathbf{K}_h \mathbf{H}_h) \mathbf{S}_h^f \middle| \mathbf{K}_h \mathbf{R}_h^{\frac{1}{2}} \right] \tag{70}$$
but is truncated by some operator Π to keep a constant (or low enough) reduced dimension r:
$$\mathbf{S}_h^a = \Pi_r \widetilde{\mathbf{S}_h^a} \in \mathbb{R}^{N_x \times r} \tag{71}$$
3. The forecast step at h reads
$$\boldsymbol{x}_{h+1}^f = \mathcal{M}_h(\boldsymbol{x}_h^f) \tag{72}$$
The square root of the forecast error variance is truncated too because of the additional model error:
$$\mathbf{S}_{h+1}^f = \Pi_r \left[\mathbf{M}_h \mathbf{S}_h^a \middle| \mathbf{Q}_h^{\frac{1}{2}} \right] \tag{73}$$

Reduced-rank square-root Kalman filter: reference

S. E. Cohn and R. Todling (1996). "Approximate data assimilation schemes for stable and unstable dynamics". In: *Journal of the Meteorological Society of Japan* 74.1, pp. 63–75

M. Verlaan and A. W. Heemink (1995). "Reduced rank square root filters for large scale data assimilation problems". In: *Second International Symposium on Assimilation of Observations in Meteorology and Oceanography*. Japan, pp. 247–252

A. W. Heemink, M. Verlaan, and A. J. Segers (2001). "Variance reduced ensemble Kalman filtering". In: *Monthly Weather Review* 129, pp. 1,718–1,728

Other methods

1. Ensemble transform Kalman filter (ETKF)

   C. Bishop, B. Etherton, and S. Majumdar (2001). "Adaptive sampling with the ensemble transform Kalman filter, part I: the theoretical aspects". In: *Monthly Weather Review* 129, pp. 420–436

2. Reduced filters SEEK and SEIK

   D. T. Pham, J. Verron, and M. C. Roubaud (1996). "A singular evolutive extended Kalman filter for data assimilation in oceanography". In: *Journal of Marine Systems* 16, pp. 323–340

   D. T. Pham (2001). "Stochastic methods for sequential data assimilation in strongly nonlinear systems". In: *Monthly Weather Review* 129, pp. 1,194–1,207

3. Hybrid ensemble Kalman filter and 4D-Var
4. Minimax filtering
5. Back and forth nudging
6. Particle filtering
7. And so on and so forth

Generation of ensembles

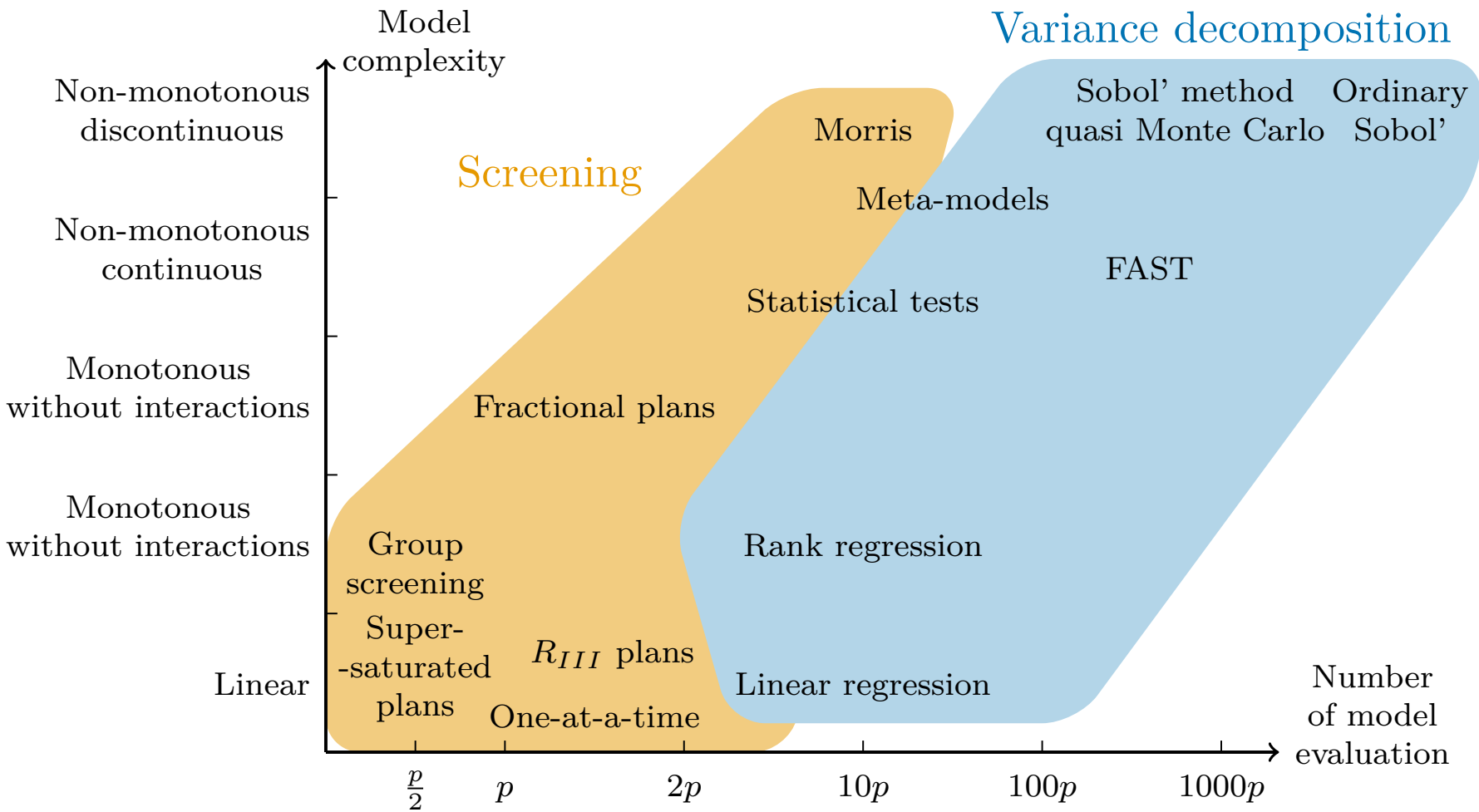
Perturbation of input data

- Monte Carlo method
- Random perturbations, with time and space correlations for fields
  - E.g.,  G. Evensen (2003). "The Ensemble Kalman Filter: theoretical formulation and practical implementation". In: *Ocean Dynamics* 53, pp. 343–367. DOI: 10.1007/s10236-003-0036-9
- Can be really difficult, in particular to preserve given properties in the perturbed fields

$$\boldsymbol{u}_h^{(m)} = \boldsymbol{u}_h + \boldsymbol{\zeta}_h^m$$

$$\boldsymbol{x}_{h+1}^{(m)f} = \mathcal{M}_h(\boldsymbol{x}_h^{(m)f}, \boldsymbol{u}_h^{(m)})$$

Generation of ensembles: sensitivity analysis



B. Iooss (2011). "Revue sur l'analyse de sensibilité globale de modèles numériques". In: *Journal de la Société Française de Statistique* 152.1, pp. 3–25 — Figure regenerated by Sylvain Girard

Generation of ensembles: multimodel ensemble

Model formulation can be changed

- In a model, there are configuration options
- It is possible to replace one parameterization or one submodel with an alternative
- It is therefore possible to build models $\mathcal{M}_h^{(m)}$ relying on different assumptions, physical/chemical/biological formulations and parameterizations, possibly less detailed or "accurate"
- The selection may be carried out randomly, providing some consistency is maintained

Numerical formulation can also be changed

- Same ideas, possibly using less accurate numerical schemes

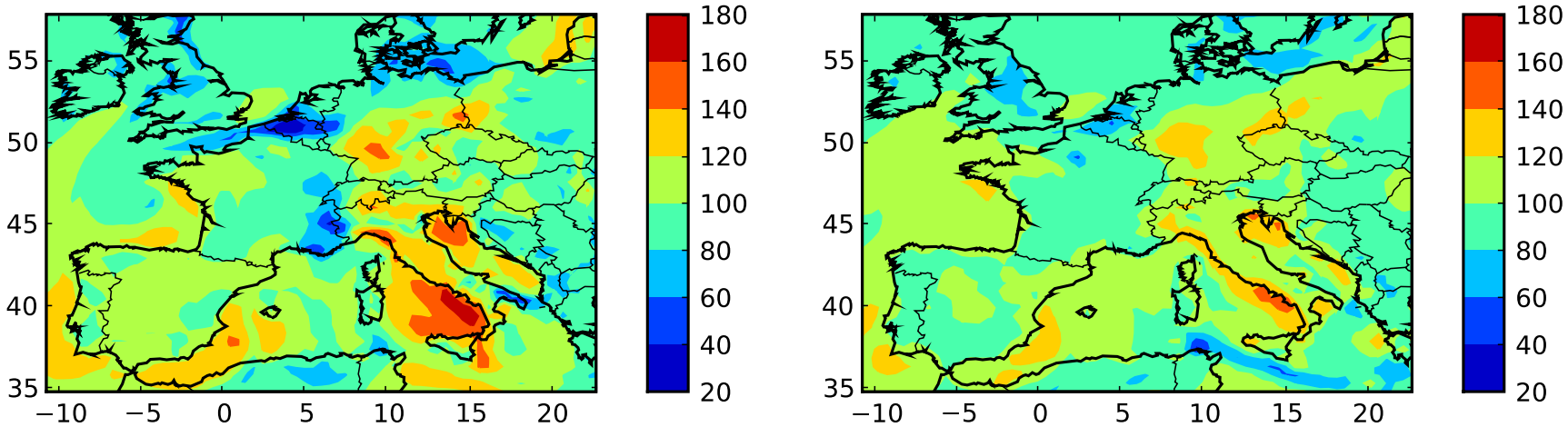
$$\boldsymbol{x}_{h+1}^{(m)f} = \mathcal{M}_h^{(m)}(\boldsymbol{x}_h^{(m)f}, \boldsymbol{u}_h^{(m)})$$

Example of options in ensemble simulations for air quality

#	*Parameterization*	*Reference*	*Alternative(s)*
	Physical parameterizations		
1.	Chemistry	RACM	RADM 2
2.	Vertical diffusion	Troen & Mahrt	Louis
			Louis in stable condition
3.	Deposition velocities	Zhang	Wesely
4.	Surface flux	Heat flux	Momentum flux
5.	Cloud attenuation	RADM method	Esquif
6.	Critical relative humidity	Depends on σ	Two layers
	Input data		
7.	Emissions vertical distribution	All in the first layer	All in the two first layers
8.	Land use coverage (dep.)	USGS	GLCF
9.	Land use coverage (bio.)	USGS	GLCF
10.	Exponent p in Troen & Mahrt	2	3
11.	Photolysis rates	JPROC	Depends on zenith angle
	Numerical issues		
12.	Time step	600 s	100 s
			1800 s
13.	Splitting method	First order	Strang splitting
14.	Horizontal resolution	0.5°	0.1°
			1.0°
15.	Vertical resolution	5 layers	9 layers
16.	First layer height	50 m	40 m

Note: already over 2×10^5 simulations possible.

On the benefit of multi-models



Ozone concentration maps in $\mu g\,m^{-3}$. Same date, but different models.

Large ensembles can bring a lot

- Currently, we explore low-dimensional error spaces
- Adequate methods can extract the best for
  - Uncertainty quantification
  - Improved forecasts
- Hence, an interesting strategy may be to generate very large ensembles (1000 members or more) as rich as possible ("shake" everything), and to process from that

Data assimilation and ensembles

Where are ensembles used?

- In the ensemble Kalman filter
- For uncertainty quantification, which is always needed in data assimilation
- For aggregation in order to improve forecasts
- To construct a metamodel which can be useful for uncertainty quantification and for Bayesian inference

How ensembles are evaluated?

Need to test the ensemble against observations

- Classical, deterministic evaluation compares model outputs and observations, which are almost directly comparable
- Comparing every individual forecast with the observations makes no sense in terms of uncertainty evaluation
- Yet we need to make use of observations, which is the only additional information we have for evaluation

Notation

- Ensemble of values $x^{(m)}$, where $m \in \{1, \ldots, M\}$
- The truth is unknown and therefore represented as a random variable Y
- Observation y which is a realization of Y
- Same, with time (or space) dependence: $x_h^{(m)}$, Y_h, y_h

What makes an ensemble adequate?

A key concept is "consistency"

- Hopefully, $x^{(m)}$ are independent and identically sampled from the same distribution Y
- y is drawn from Y as well
- Consequently the realizations y should not be distinguishable from the samples $x^{(m)}$
- This is the "consistency" property

J. L. Anderson (1997). "The impact of dynamical constraints on the selection of initial conditions for ensemble predictions: low-order perfect model results". In: *Monthly Weather Review* 125, pp. 2,969–2,983. DOI: 10.1175/1520-0493(1997)125<2969:TIODCO>2.0.CO;2

What makes an ensemble adequate?

The key value is the "rank"

- y is of rank m if $x^{(m)} \leq y < x^{(m+1)}$, with $x^{(m)}$ sorted
  - It is 0 if $y < x^{(1)}$
  - It is M if $x^{(M)} \leq y$
- I.e., y is of rank m if m members simulated a lower value
- The rank is in $\{0, \ldots, M\}$

Rank histogram or "Talagrand diagram"

Rank histogram

- Number of observations of a given rank
- Ideal histogram: flat histogram
- References:


J. L. Anderson (1996). "A Method for Producing and Evaluating Probabilistic Forecasts from Ensemble Model Integrations". In: *Journal of Climate* 9.7, pp. 1518–1530

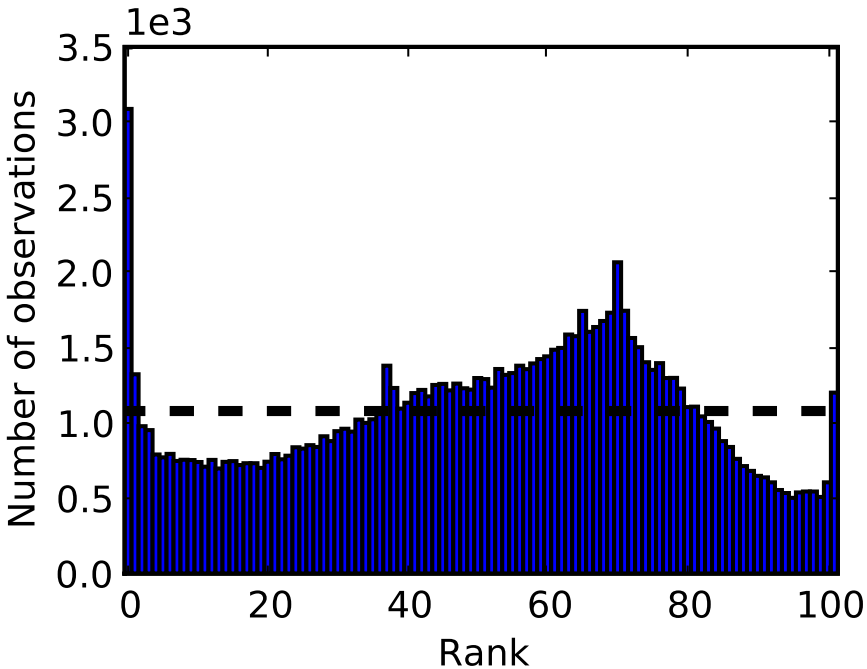

T. M. Hamill and S. J. Colucci (1997). "Verification of Eta/RSM Short-Range Ensemble Forecasts". In: *Monthly Weather Review* 125, pp. 1312–1327


O. Talagrand, R. Vautard, and B. Strauss (1999). *Evaluation of Probabilistic Prediction System*. Proceedings of the ECMWF Workshop on Predictability. Reading, United Kingdom

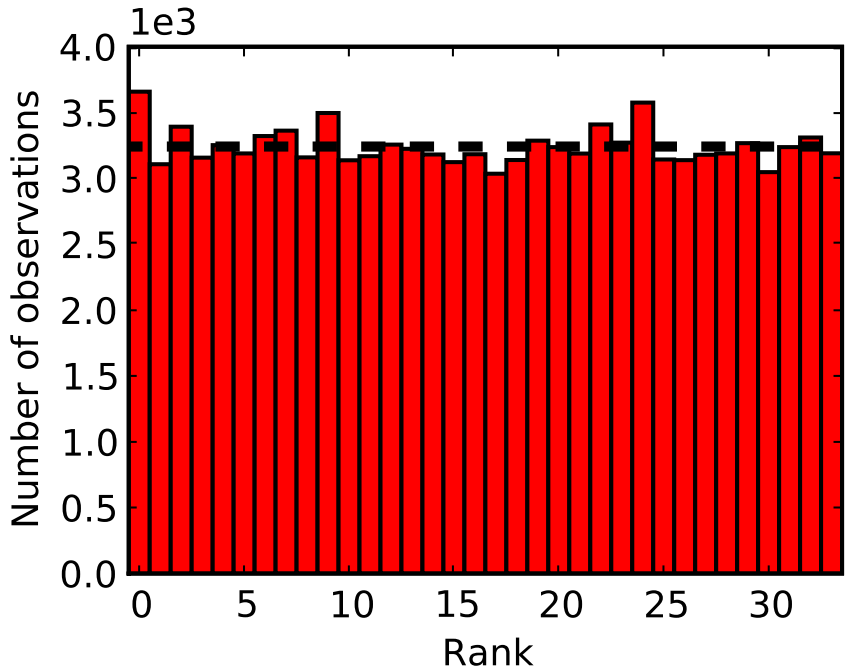
Multidimensional extension

- E.g., the minimum spanning trees

Calibration of an ensemble for air quality using the rank histogram

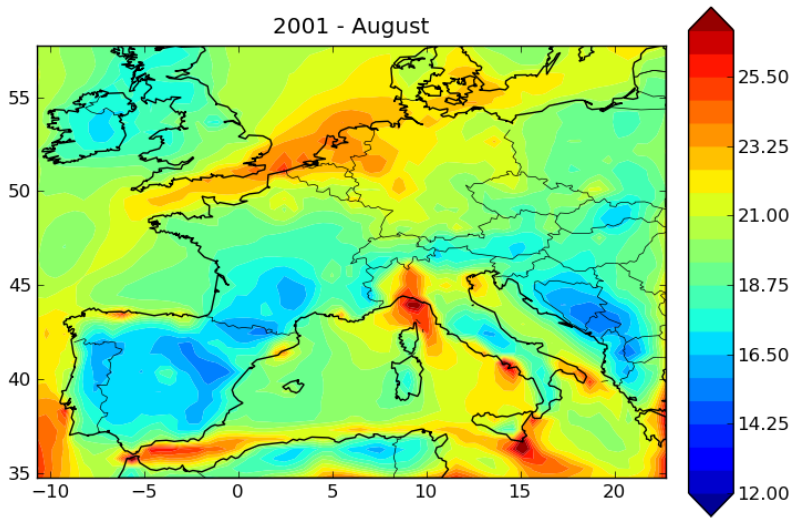


Complete ensemble

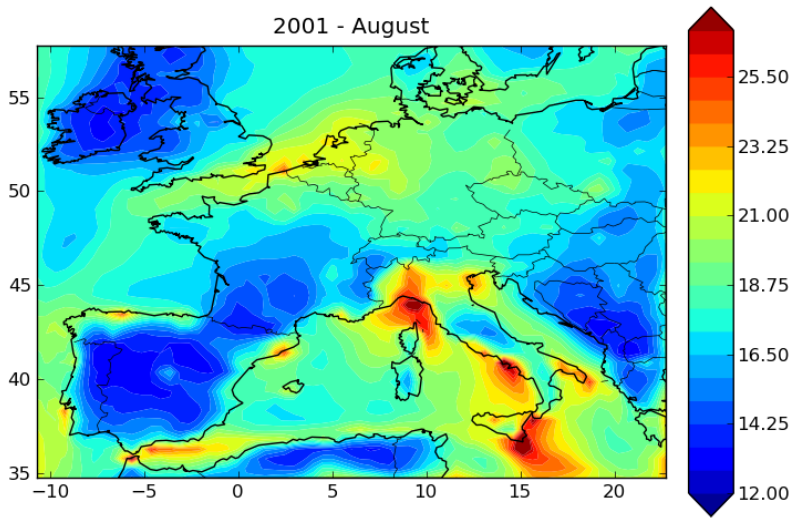


Selected sub-ensemble

Standard deviation of the ensemble



Complete ensemble



Selected sub-ensemble

What are the desirable properties for a good ensemble?

Reliability

- Accuracy of prediction, i.e., agreement between the probabilistic forecast and the observed frequency
- Sometimes called "calibration"

Sharpness

- Variability of the forecasts
- It can be seen as the amount of information available in the system

Resolution

- Ability of the ensemble to make distinct predictions for different subsets of events

Brier score

Definition

- Let p_h the probability that an event occurs at time/location h according to the ensemble (e.g., number of members that forecast the event, divided by M), and $o_h = 1$ if the event indeed occurs, or $o_h = 0$ if the event did not occur.
- $$\mathrm{BS} = \frac{1}{T} \sum_{h=1}^{T} (p_h - o_h)^2 \tag{74}$$

G. W. Brier (1950). "Verification of Forecasts Expressed in Terms of Probability". In: *Monthly Weather Review* 78.1, pp. 1–3. DOI: 10.1175/1520-0493(1950)078<0001:VOFEIT>2.0.CO;2

Discrete ranked probability score

- The ranked probability score generalizes the Brier score for a set of events $Y \geq \text{threshold}_k$, $k \in \{1, K\}$
- For every threshold k and date t, there is an observed "probability" $o_h^{|k} \in \{0, 1\}$ and a simulated probability $p_h^{|k}$
- The discrete ranked probability score reads

$$\text{DRPS} = \frac{1}{TK} \sum_{h=1}^{T} \sum_{k=1}^{K} (p_h^{|k} - o_h^{|k})^2$$

- It actually evaluates the full distribution

Continuous ranked probability score

- We consider cumulative distribution functions
  - Derived from the ensemble, hence simulated: $F_p(z) = \frac{1}{M}|\{m/x^{(m)} \leq z\}|$, where $|\cdot|$ stands for the cardinal
  - "Observed" and assumed to be a step function: $F_o(z) = \mathbf{1}_{[y,+\infty[}(z)$, where $\mathbf{1}_A(z) = 1$ if $z \in A$, and $\mathbf{1}_A(z) = 0$ otherwise
- The continuous ranked probability score reads

$$\text{CRPS} = \frac{1}{T}\sum_{h=1}^{T}\int_{\mathbb{R}}(F_p - F_o)^2$$

A few useful references


D. S. Wilks (2005). *Statistical Methods in the Atmospheric Sciences*. Second. International Geophysics Series. Academic Press


`http://www.cawcr.gov.au/projects/verification/`


I. T. Jolliffe and D. B. Stephenson, eds. (2003). *Forecast verification: A practitioner's guide in atmospheric science*. Wiley

Fully Bayesian approach

Bayes' theorem

$$p(\boldsymbol{x}|\boldsymbol{y}) = \frac{p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{x})}{p(\boldsymbol{y})}$$

BLUE is the maximum a posteriori (MAP) in Gaussian case

- Assume $\boldsymbol{x}^t \sim \mathcal{N}(\boldsymbol{x}^f, \mathbf{P}^f)$ and $\boldsymbol{y}|\boldsymbol{x}^t \sim \mathcal{N}(\mathbf{H}\boldsymbol{x}^t, \mathbf{R})$.
- We are looking for $\boldsymbol{x}^a$ that maximizes $p(\boldsymbol{x}^t|\boldsymbol{y})$, hence $p(\boldsymbol{y}|\boldsymbol{x}^t)p(\boldsymbol{x}^t)$ because $p(\boldsymbol{y})$ does not depend on $\boldsymbol{x}^t$.

$$p(\boldsymbol{x}^t) = (2\pi)^{-\frac{N_x}{2}} (\det \mathbf{P}^f)^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(\boldsymbol{x}^t - \boldsymbol{x}^f)^\top \mathbf{P}^{f\,-1}(\boldsymbol{x}^t - \boldsymbol{x}^f)\right)$$

$$p(\boldsymbol{y}|\boldsymbol{x}^t) = (2\pi)^{-\frac{N_y}{2}} (\det \mathbf{R})^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(\boldsymbol{y} - \mathbf{H}\boldsymbol{x}^t)^\top \mathbf{R}^{-1}(\boldsymbol{y} - \mathbf{H}\boldsymbol{x}^t)\right)$$

$$-2\log p(\boldsymbol{x}^t|\boldsymbol{y}) = (\boldsymbol{x}^t - \boldsymbol{x}^f)^\top \mathbf{P}^{f\,-1}(\boldsymbol{x}^t - \boldsymbol{x}^f) + (\boldsymbol{y} - \mathbf{H}\boldsymbol{x}^t)^\top \mathbf{R}^{-1}(\boldsymbol{y} - \mathbf{H}\boldsymbol{x}^t) + C$$

The minimum is attained at BLUE.

Bayesian inference

With parameters to estimate

$$\boldsymbol{x}^t = \mathcal{M}_{1\to T}(\boldsymbol{u})$$

$$p(\boldsymbol{u}|\boldsymbol{y}) = \frac{p(\boldsymbol{y}|\boldsymbol{u})p(\boldsymbol{u})}{p(\boldsymbol{y})}$$

- We may still assume, e.g., $\boldsymbol{y}|\boldsymbol{u} \sim \mathcal{N}(\mathbf{H}\mathcal{M}_{1\to T}(\boldsymbol{u}), \mathbf{R})$.
- We need a prior distribution on the parameters; e.g., $\boldsymbol{u} \sim \mathcal{U}([u_1^{\min}, u_1^{\max}] \times \ldots \times [u_K^{\min}, u_K^{\max}])$.
- We may not be able to compute the full distribution analytically. As a consequence, we sample it.

Bayesian inference

With parameters to estimate

$$\boldsymbol{x}^t = \mathcal{M}_{1\to T}(\boldsymbol{u})$$

$$p(\boldsymbol{u}|\boldsymbol{y}) = \frac{p(\boldsymbol{y}|\boldsymbol{u})p(\boldsymbol{u})}{p(\boldsymbol{y})}$$

- Once we are given $p(\boldsymbol{u})$ and $p(\boldsymbol{y}|\boldsymbol{u})$, we have access to $p(\boldsymbol{u}|\boldsymbol{y})$ up to a multiplicative constant, $p(\boldsymbol{y})$.
- Markov chain Monte Carlo (MCMC) methods are often used.
- A well known method is Metropolis-Hastings algorithm which can sample a distribution using a function proportional to the target probability density.
  - It generates a sequence $\boldsymbol{u}_i^{(MH)}$, which samples $\boldsymbol{u}|\boldsymbol{y}$ when i is large enough.
  - It requires a large number of calls to the model: $\boldsymbol{x}_i^{(MH)} = \mathcal{M}_{1\to T}(\boldsymbol{u}_i^{(MH)})$, to evaluate $p(\boldsymbol{y}|\boldsymbol{u}_i^{(MH)})$.

Meta-modeling with reduction and emulation

Two-step approach

- Dimension reduction: projection of inputs ($\boldsymbol{u}$) and outputs ($\boldsymbol{x}$) onto reduced subspaces.
- Emulation: replacing the relation between reduced inputs and reduced outputs with a fast function.
  - Regression and/or interpolation between training points.

Dimension reduction

$$\boldsymbol{x} = \mathcal{M}_{1\to T}(\boldsymbol{u})$$

- Reduced basis: $(\boldsymbol{\Psi}_j)_{j=1,\dots,q}$
  - May be derived from principal component analysis (PCA) on some sequence $\boldsymbol{x}_i$
  - $\boldsymbol{x} \simeq \bar{\boldsymbol{x}} + \sum_{j=1}^{n} \alpha_j \boldsymbol{\Psi}_j$, where $\alpha_j = (\boldsymbol{x} - \bar{\boldsymbol{x}})^\top \boldsymbol{\Psi}_j$
- Reduced model

  $$\boldsymbol{x} \simeq \bar{\boldsymbol{x}} + \boldsymbol{\Psi}\boldsymbol{\Psi}^\top(\mathcal{M}_{1\to T}(\boldsymbol{u}) - \bar{\boldsymbol{x}})$$

  where $\boldsymbol{\Psi}$ is the matrix whose jth column is $\boldsymbol{\Psi}_j$.
  - Note: if the dimension of the parameters is high, one can project $\boldsymbol{u}$ onto the subspace spanned by the columns of $\boldsymbol{\Phi}$:
    $\boldsymbol{x} \simeq \bar{\boldsymbol{x}} + \boldsymbol{\Psi}\boldsymbol{\Psi}^\top(\mathcal{M}_{1\to T}(\boldsymbol{\Phi}\boldsymbol{\Phi}^\top\boldsymbol{u}) - \bar{\boldsymbol{x}})$
- The reduced model includes $\boldsymbol{\Psi}^\top\mathcal{M}_{1\to T}(\boldsymbol{u})$ which is function of $\boldsymbol{u}$ with few outputs.

Emulation (response surface methods)

- We consider $f(\boldsymbol{u}) = \boldsymbol{\Psi}^\top \mathcal{M}_{1\to T}(\boldsymbol{u})$ and emulate independently every component $f_j(\boldsymbol{u})$ with the fast and approximate function $\widehat{f}_j(\boldsymbol{u})$.
- We use a training set $\boldsymbol{\Psi}^\top \mathcal{M}_{1\to T}(\boldsymbol{u}^{(i)})$, $i = \{1, L\}$

$$\widehat{f}_j(\boldsymbol{u}) = \underbrace{\sum_{k=1}^{K} \beta_{j,k} p_k}_{\text{Regression}} + \underbrace{\sum_{i=1}^{L} w_{j,i}(\boldsymbol{u}, \boldsymbol{u}^{(1)}, \ldots, \boldsymbol{u}^{(L)}) \left(f_j(\boldsymbol{u}^{(i)}) - \sum_{k=1}^{K} \beta_{j,k} p_k^{(i)} \right)}_{\text{Interpolation of the residuals}} .$$

- See radial basis functions (RBF), Kriging, random forests, LARS, ...
- Finally,

$$\boldsymbol{x} = \bar{\boldsymbol{x}} + \boldsymbol{\Psi}(\widehat{f}(\boldsymbol{u}) - \boldsymbol{\Psi}^\top \bar{\boldsymbol{x}})$$

Emulation (response surface methods): a few references

Managing Uncertainty in Complex Models (MUCM): `http://mucm.aston.ac.uk/MUCM/MUCMToolkit/`

DiceKriging: `http://cran.r-project.org/web/packages/DiceKriging/index.html`

COBRA: Nonlinear Aggregation of Predictors; `http://cran.r-project.org/web/packages/COBRA/index.html`

J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn (1989). "Design and analysis of computer experiments". In: *Statistical Science* 4.4, pp. 409–423

C. E. Rasmussen and C. K. I. Williams (2006). *Gaussian Processes for Machine Learning*. MIT Press

V. R. Joseph and L. Kang (2011). "Regression-based inverse distance weighting with applications to computer experiments". In: *Technometrics* 53.3, pp. 254–265. DOI: 10.1198/TECH.2011.09154

Ensemble forecast: online ensemble learning

Principle: to create a linear combination x^f of the members with best performance than any individual member.

$$x_h^f = \sum_{m=1}^{M} u_h^{(m)} \times x_h^{(m)f} \, .$$

$x_h^{(m)f}$	:	value of m-th member.
$u_h^{(m)}$	:	weight of m-th member.
y_h	:	observation or analysis to be forecasted.

The method is applied independently to each component of the state vector.

Ensemble forecast: online ensemble learning

$$x_h^f = \sum_{m=1}^{M} u_h^{(m)} \times x_h^{(m)f} \, .$$

$x_h^{(m)f}$: value of m-th member.
$u_h^{(m)}$: weight of m-th member.
y_h : observation (or verification) to be forecasted.

$h-2$	$h-1$	h	$h+1$
$x_{h-2}^{(m)f}$	$x_{h-1}^{(m)f}$	$x_h^{(m)f}$	$x_{h+1}^{(m)f}$
$u_{h-2}^{(m)}$	$u_{h-1}^{(m)}$	$u_h^{(m)}$	$u_{h+1}^{(m)}$
x_{h-2}^{f}	x_{h-1}^{f}	x_h^{f}	x_{h+1}^{f}
y_{h-2}	y_{h-1}	y_h	y_{h+1}

Ensemble forecast: online ensemble learning

$$x_h^f = \sum_{m=1}^{M} u_h^{(m)} \times x_h^{(m)f} \ .$$

$x_h^{(m)f}$: value of m-th member.
$u_h^{(m)}$: weight of m-th member.
y_h : observation (or verification) to be forecasted.

$h-2$	$h-1$	h	$h+1$
$x_{h-2}^{(m)f}$	$x_{h-1}^{(m)f}$	$x_h^{(m)f}$	$x_{h+1}^{(m)f}$
$u_{h-2}^{(m)}$	$u_{h-1}^{(m)}$	$u_h^{(m)}$	$u_{h+1}^{(m)}$
x_{h-2}^{f}	x_{h-1}^{f}	x_h^{f}	x_{h+1}^{f}
y_{h-2}	y_{h-1}	y_h	y_{h+1}

Ensemble forecast: weights computation

Approach: online regularized regression.

Loss function: $\ell_h(\boldsymbol{u}) = (\sum_{m=1}^{M} u^{(m)} x_h^{(m)f} - y_h)^2$.

Regularization term: $r(\boldsymbol{u}) = \lambda \|\boldsymbol{u}\|^2$.

The weight vector $\boldsymbol{u}_h = [u_h^{(1)}, u_h^{(2)}, ..., u_h^{(M)}]^\top$ is chosen as:

$$\boldsymbol{u}_h = \underset{\boldsymbol{w} \in \mathbb{R}^M}{\operatorname{argmin}} \left[r(\boldsymbol{w}) + \sum_{h'=1}^{h-1} \ell_{h'}(\boldsymbol{w}) \right] .$$

Ensemble forecast: weights computation

Approach: online regularized regression.

Loss function: $\ell_h(\boldsymbol{u}) = (\sum_{m=1}^{M} u^{(m)} x_h^{(m)f} - y_h)^2$.

Regularization term: $r(\boldsymbol{u}) = \lambda \|\boldsymbol{u}\|^2$.

The weight vector $\boldsymbol{u}_h = [u_h^{(1)}, u_h^{(2)}, ..., u_h^{(M)}]^\top$ is chosen as:

$$\boldsymbol{u}_h = \underset{\boldsymbol{w} \in \mathbb{R}^M}{\operatorname{argmin}} \left[r(\boldsymbol{w}) + \sum_{h'=1}^{h-1} \beta_{h,h'} \, \ell_{h'}(\boldsymbol{w}) \right] .$$

Focus on recent data with discount factors: $\beta_{h,h'} = \frac{\gamma}{(h-h')^2}$

In practice: low sensitivity around optimal parameters λ, γ.

Ensemble forecast: theoretical guarantee of robustness

Case of online ridge regression

Under essentially no assumptions:

- Tends to zero with increasing T. 
- Cumulated loss  of our forecasts.

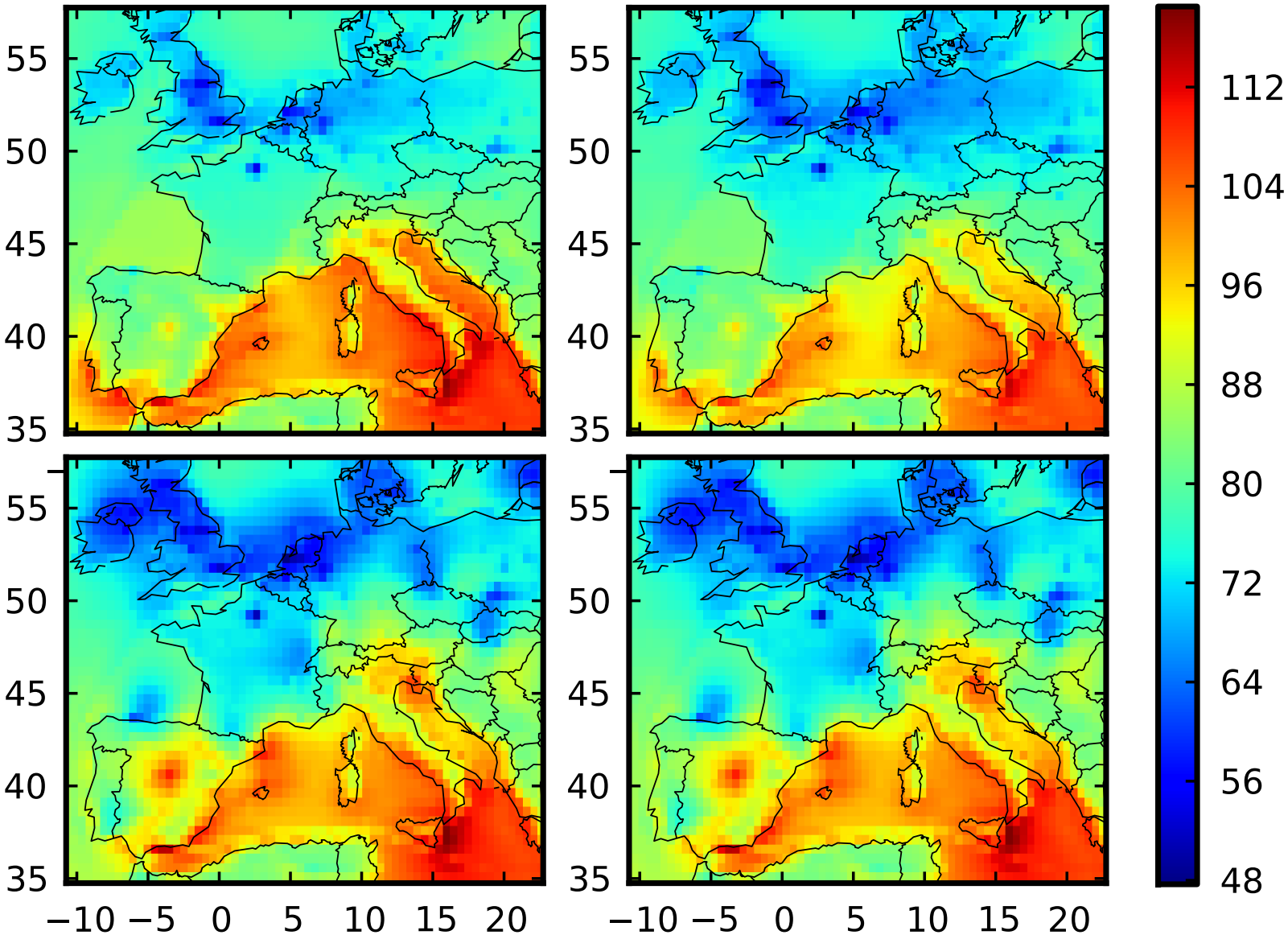
$$\frac{1}{T}\left(\sum_{h=1}^{T}\ell_h(\boldsymbol{u}_h) - \min_{\boldsymbol{w}\in\mathbb{R}^M}\sum_{h=1}^{T}\ell_h(\boldsymbol{w})\right) \leq \mathcal{O}\left(\frac{\ln T}{T}\right)$$

- Cumulated loss  of the best combination with constant weights.

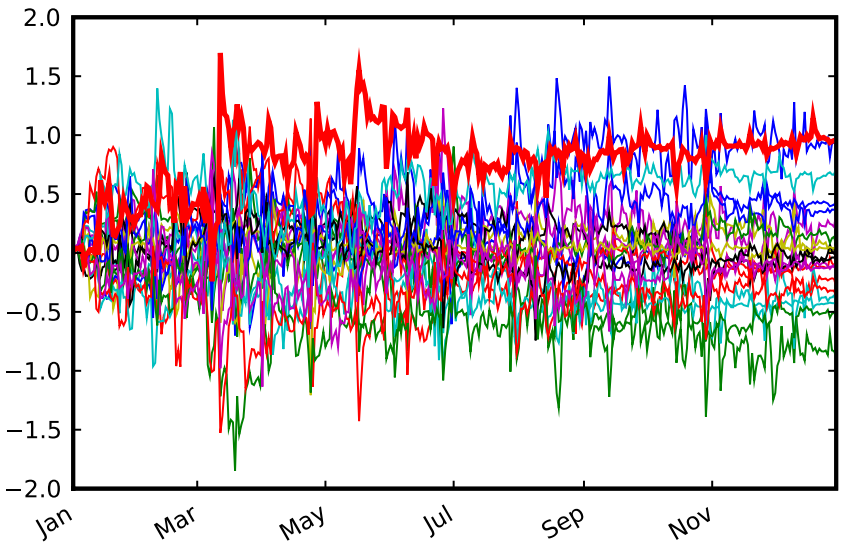
Other algorithms

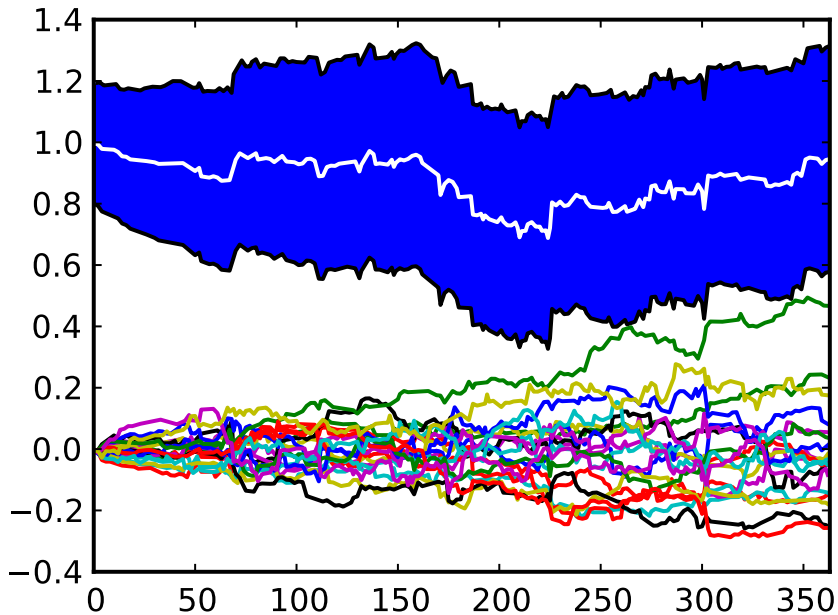
- Some algorithms carry out sparse aggregation, e.g., LASSO, which can be well adapted in a context with very large ensembles

Ensemble forecast: example results (ozone maps, $\mu g\,m^{-3}$)

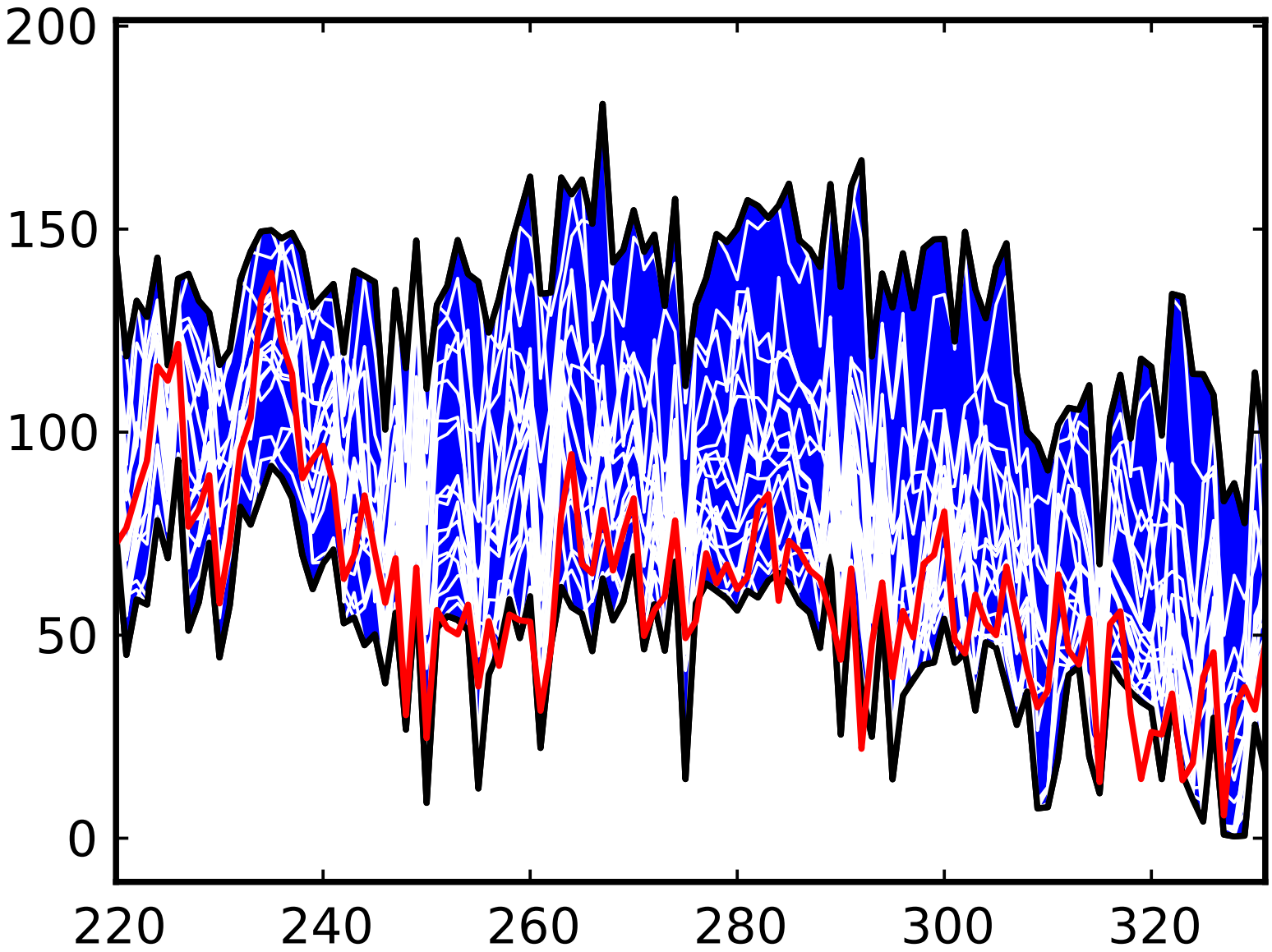


Ensemble forecast: example results (weights against time)

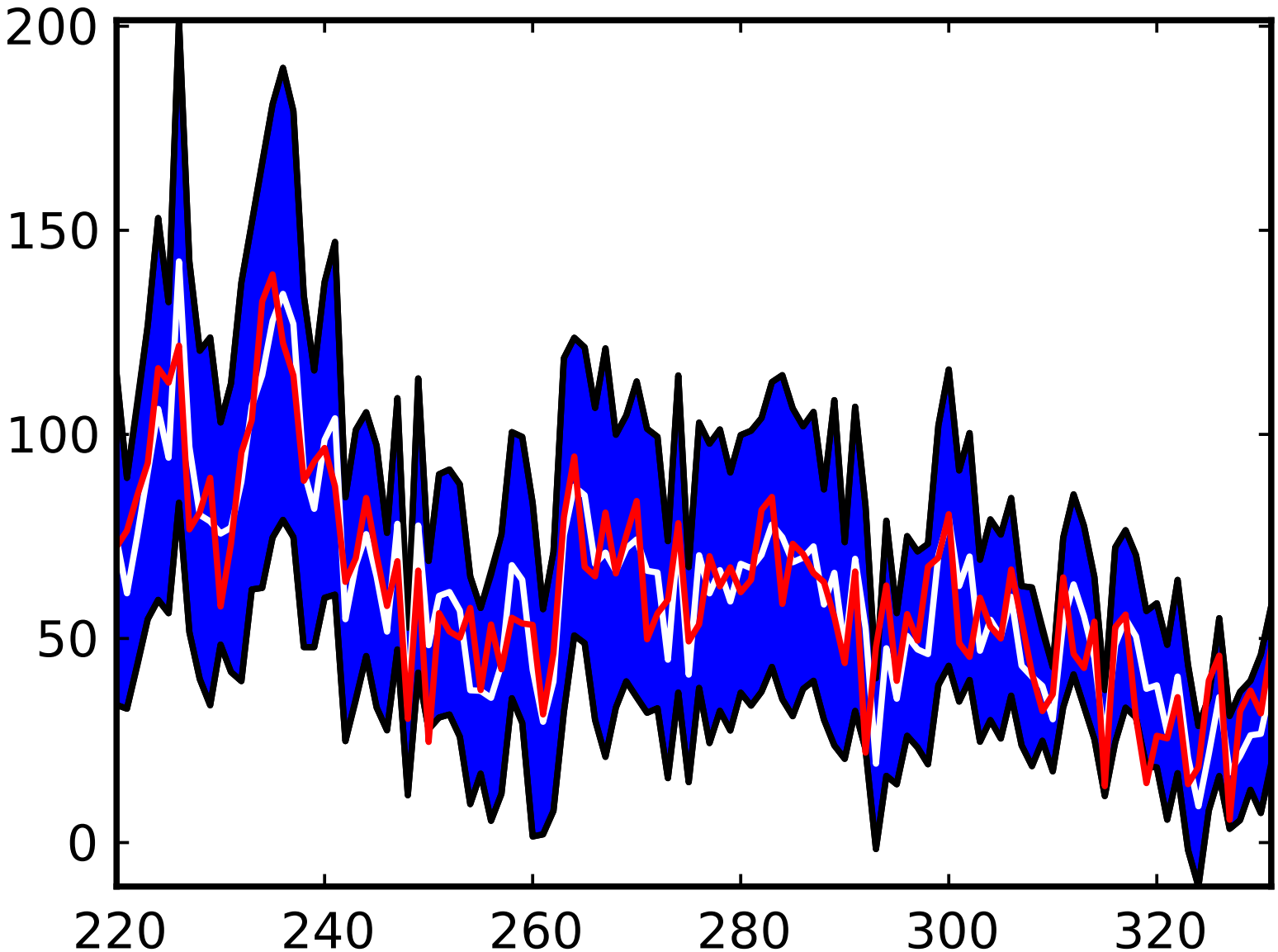




Ensemble forecast: example results (ozone, $\mu g\,m^{-3}$)



Ensemble forecast: example results (ozone, $\mu g\,m^{-3}$)



Ensemble forecast: references


V. Mallet, G. Stoltz, and B. Mauricette (2009). "Ozone ensemble forecast with machine learning algorithms". In: *Journal of Geophysical Research* 114.D05307. DOI: 10.1029/2008JD009978


V. Mallet (2010). "Ensemble forecast of analyses: Coupling data assimilation and sequential aggregation". In: *Journal of Geophysical Research* 115.D24303. DOI: 10.1029/2010JD014259


V. Mallet, A. Nakonechny, and S. Zhuk (2013). "Minimax filtering for sequential aggregation: Application to ensemble forecast of ozone analyses". In: *Journal of Geophysical Research* 118.11, pp. 11,294–11,303. DOI: 10.1002/jgrd.50751

Data assimilation library Verdandi

Software

- Open source library (LGPL)
- Generic C++ library and Python interface
- Available at `http://gforge.inria.fr/verdandi/`

Using of the library

- Relies a clear interface to the model and the observation operator: `GetState`, `GetParameter`, `GetParameterVariance`
- Parallelization in the assimilation method and support for parallelized models as well
- Applied to real applications (see next slides)

D. Chapelle, M. Fragu, V. Mallet, and P. Moireau (2013). "Fundamental principles of data assimilation underlying the Verdandi library: applications to biophysical model personalization within euHeart". In: *Medical & Biological Engineering & Computing* 51.11, pp. 1,221–1,233

Applications of data assimilation

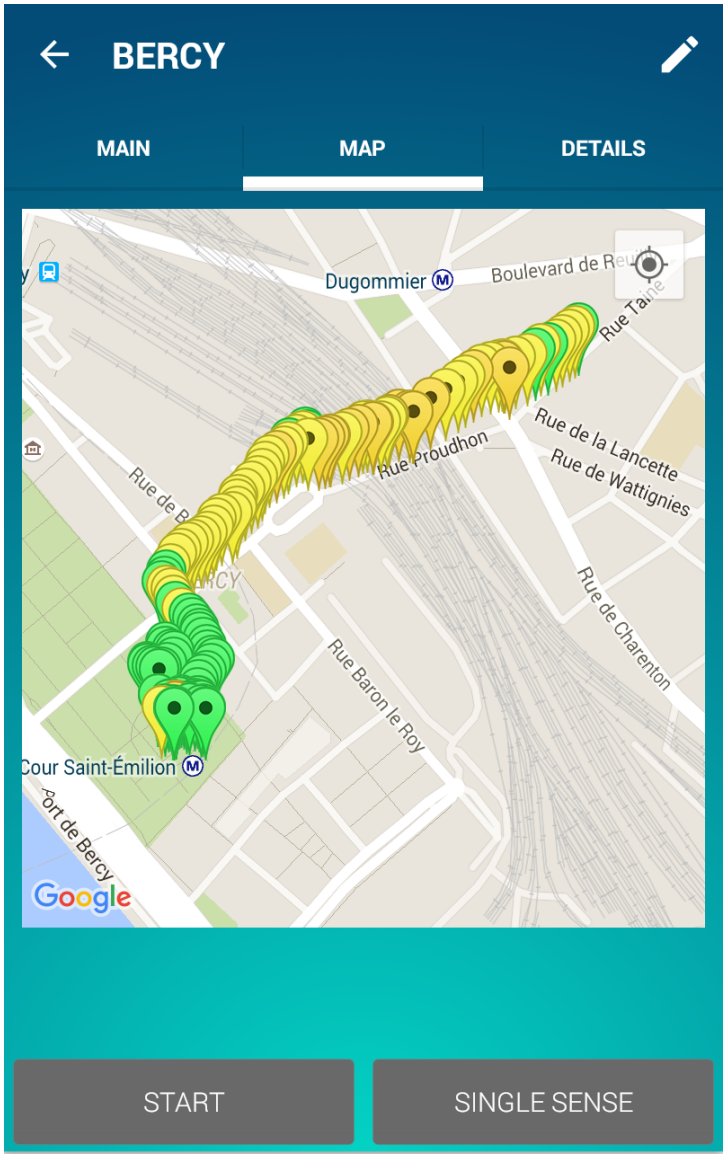
Classical applications

- Meteorology
- Geosciences

New range of applications

- Biology, medicine (see M3DISIM Inria team, D. Chapelle, P. Moireau)
- Urban pollutions (air quality, noise pollution, . . .)
- New sensors, mobile measurements, big data

New applications of data assimilation

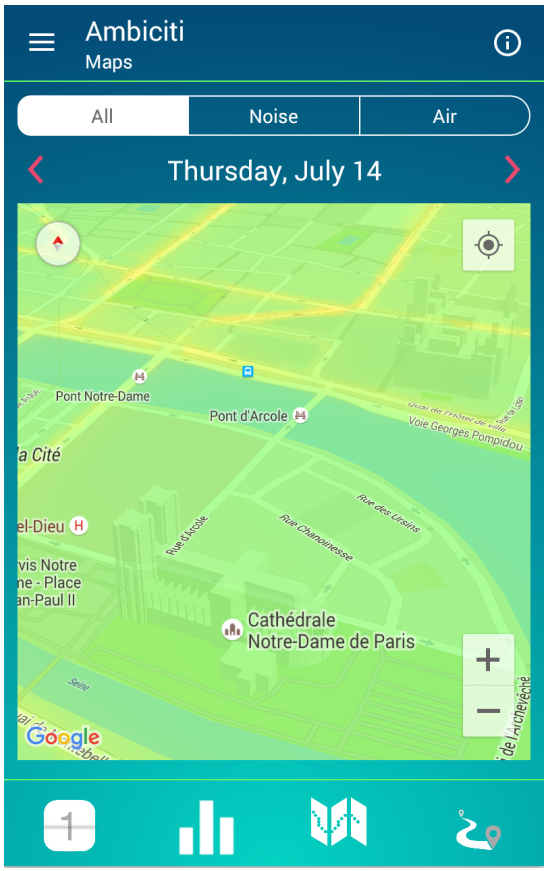


Noise measurements by the application

Example of the mobile application SoundCity

- Press conference with Paris City: `http://tiny.cc/soundcity-press`
- Freely available on Google Play and App Store
- Measures noise levels with the smartphone
- Soon will include air quality, and be renamed Ambiciti

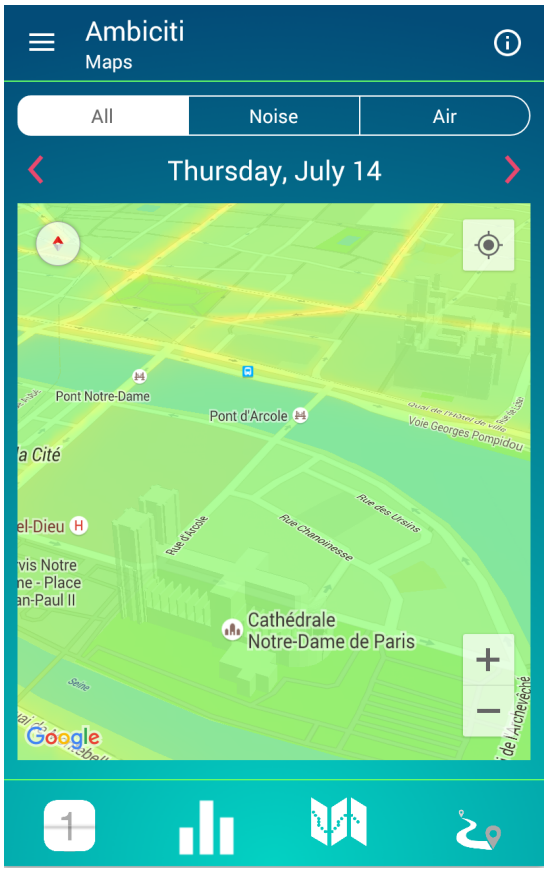
New applications of data assimilation



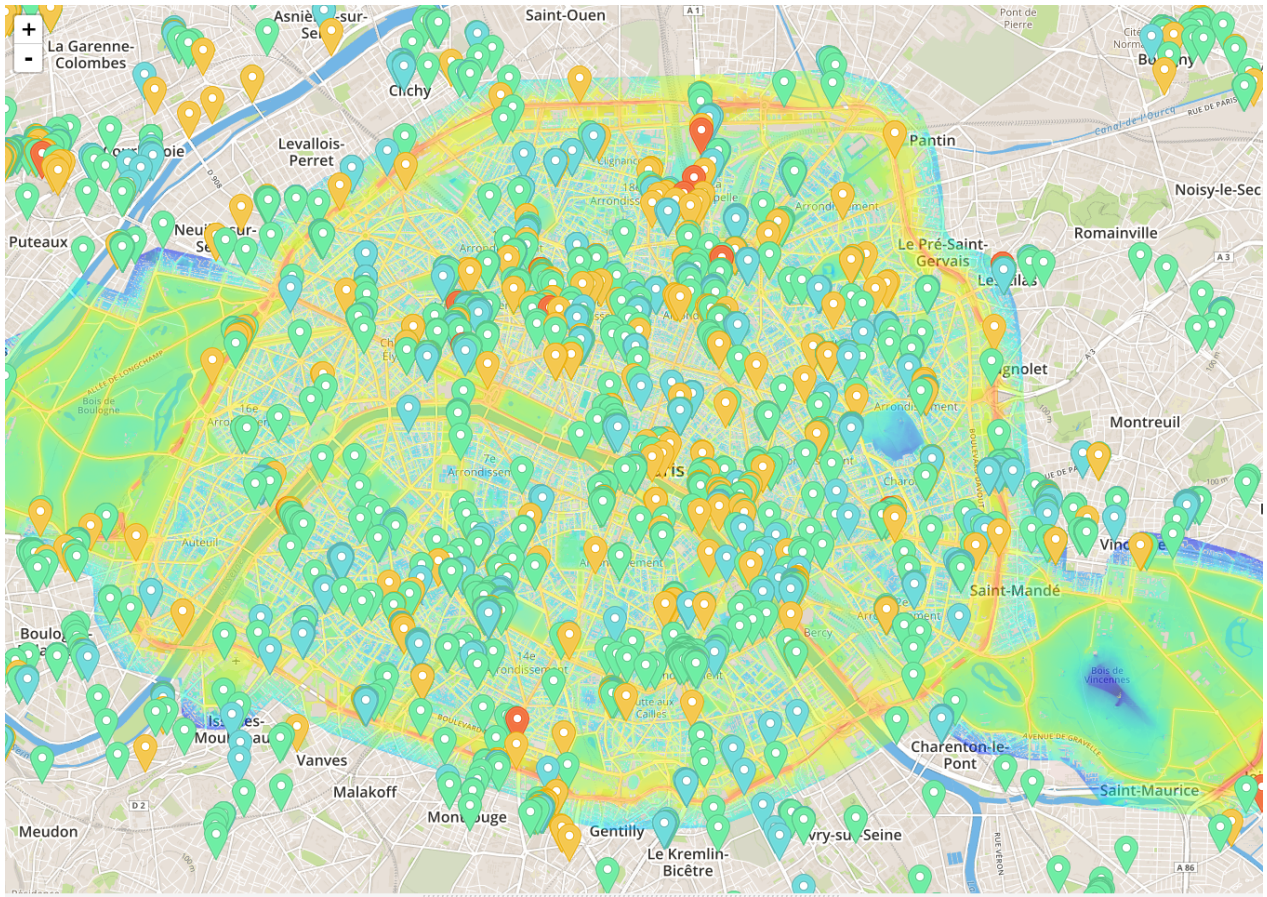
Air quality map
on the mobile application

Collected noise observations
during one hour in Paris

New applications of data assimilation



Air quality map
on the mobile application



Collected noise observations
during eight hours in Paris

Appendix

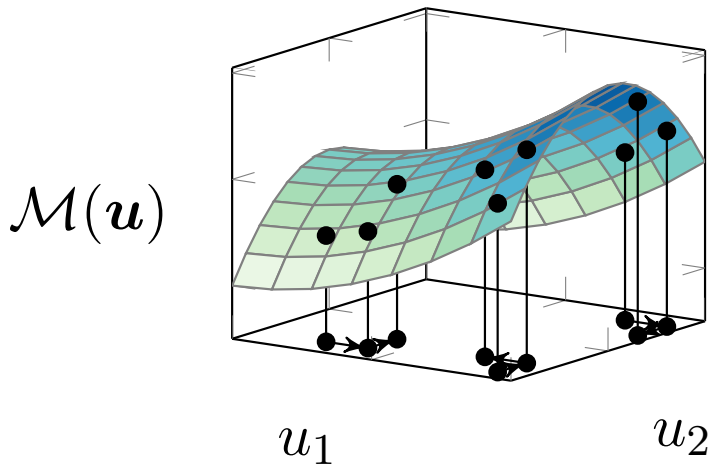
Morris screening method

- Elementary effects are computed by perturbing each input variable in turn, one at a time

- Elementary effect:

$$d_j = \frac{\mathcal{M}(\boldsymbol{u} + \zeta \boldsymbol{i}_j) - \mathcal{M}(\boldsymbol{u})}{\zeta}$$

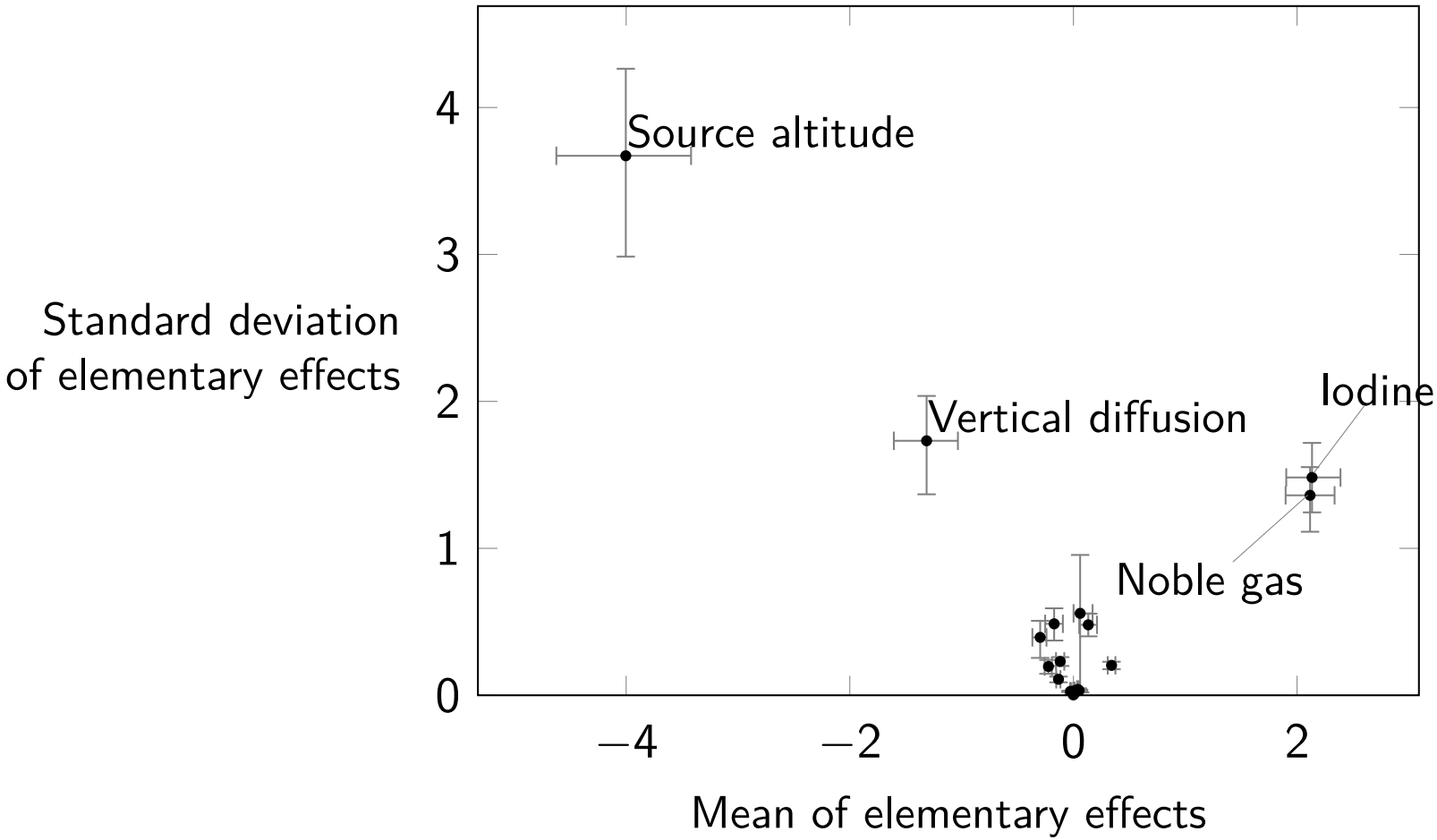
where $\boldsymbol{i}_j$ contains 1 at index j and 0 otherwise



- Starting points are randomized in order to sample the distribution of elementary effects
- The moments of this distribution are used as sensitivity measures

Example of results obtained with Morris method

For the atmospheric dispersion of radionuclides after Fukushima disaster

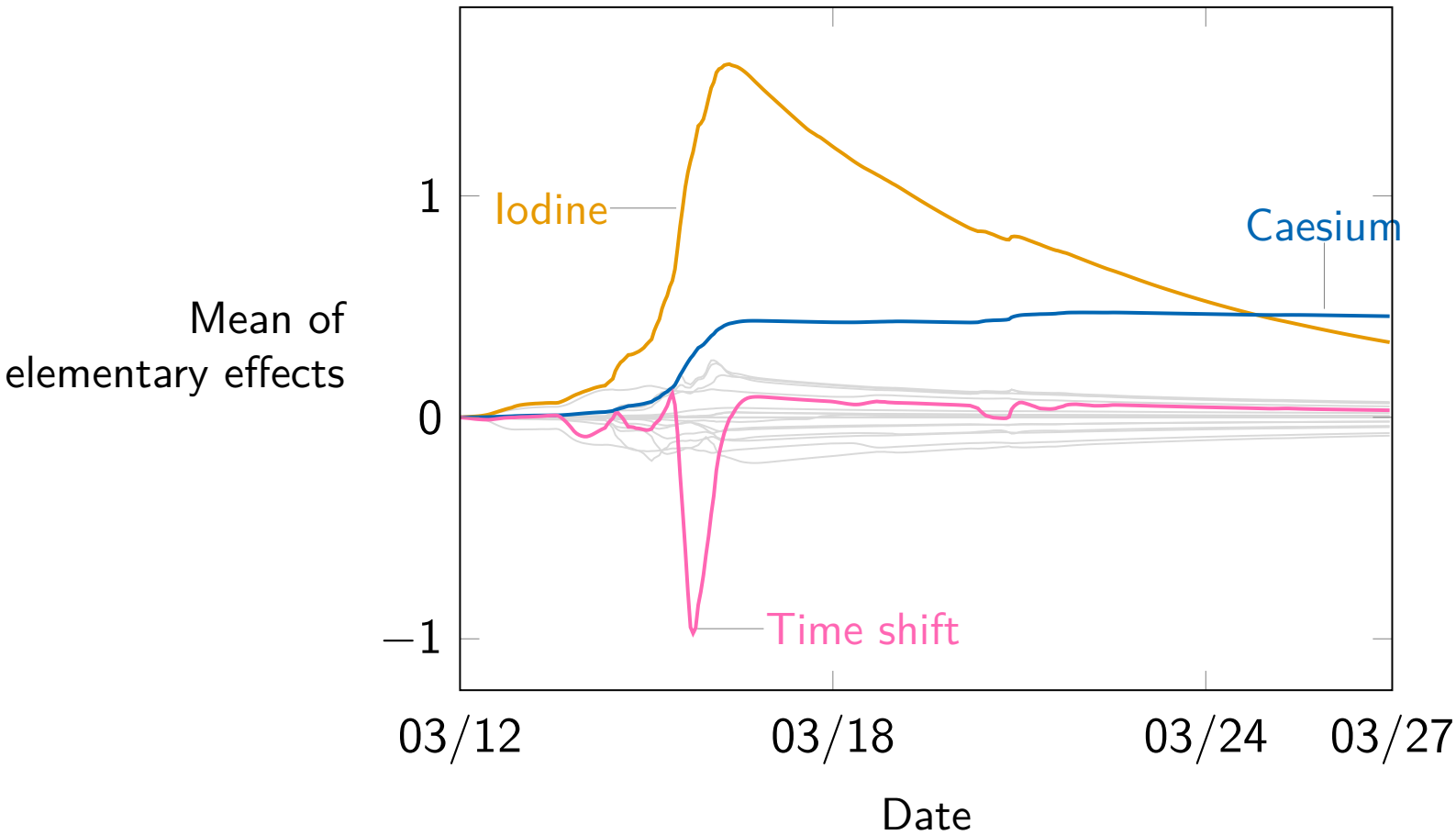


Sensitivity of the spatio-temporal average of atmospheric dose rate

Figure generated by Sylvain Girard (IRSN)

Example of results obtained with Morris method

For the atmospheric dispersion of radionuclides after Fukushima disaster



Sensitivity of the spatial average of deposit dose rate
Figure generated by Sylvain Girard (IRSN)

How to determine the variance $\mathbf{P}^f$?

Using a priori uncertainty quantification

- E.g., construct a matrix $\mathbf{P}^f$ independent of time

The covariance is often parameterized, independently of time

- Take the covariance between two components i and j in the state, which represent the same variable at two points in space.
- Assume the distance between the two points is d_{ij}.

Exponential: $P_{ij}^f = b \exp(-\frac{d_{ij}}{L})$
Second-order autoregressive function (SOAR):
$P_{ij}^f = b \left(1 + \frac{d_{ij}}{L}\right) \exp\left(-\frac{d_{ij}}{L}\right)$

 G. Gaspari and S. E. Cohn (1999). "Construction of correlation functions in two and three dimensions". In: *Quarterly Journal of the Royal Meteorological Society* 125, pp. 723–757

How to determine the variance $\mathbf{P}^f$?

Closer look at the innovation

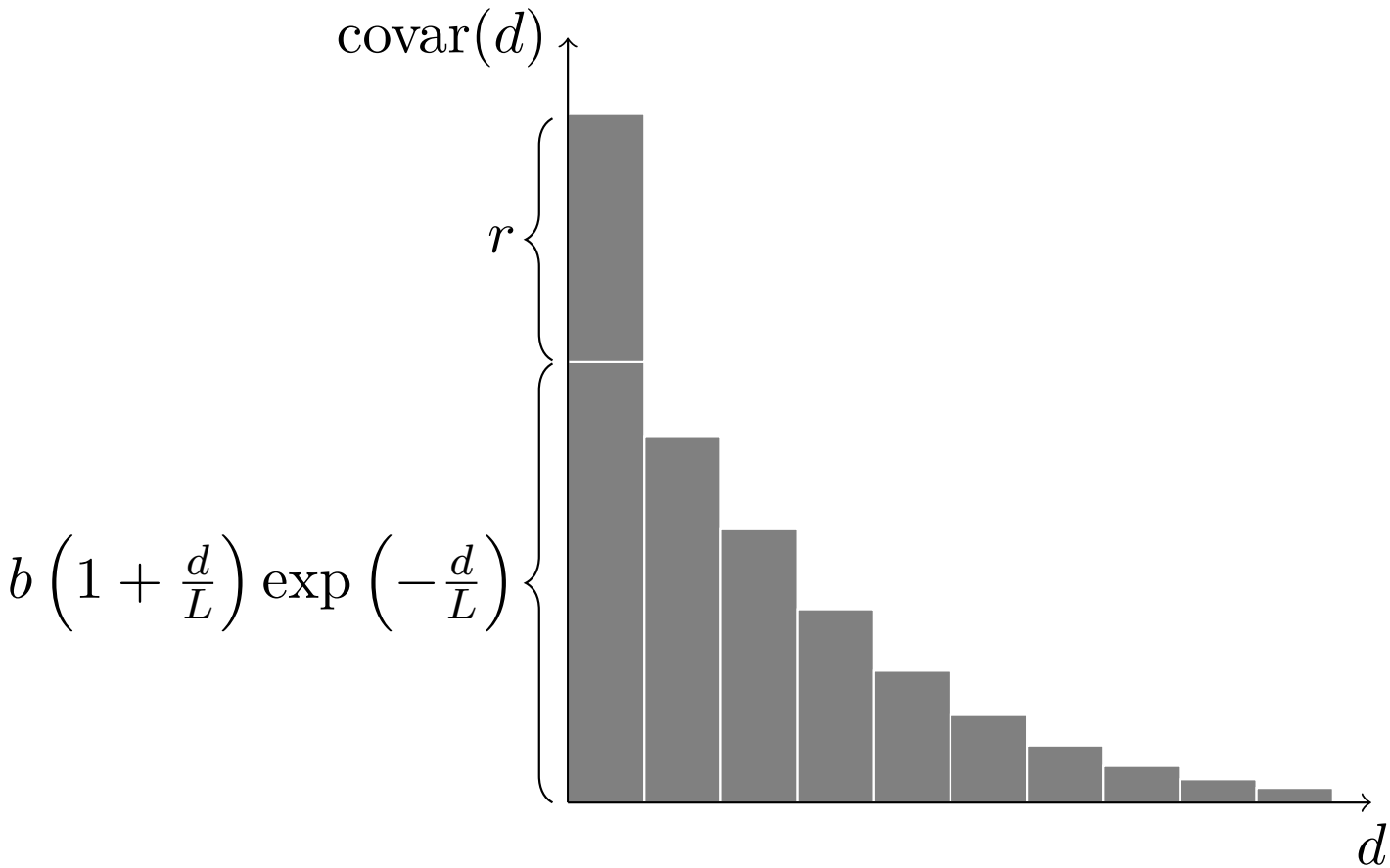
- Let us consider the innovation
$$\boldsymbol{y} - \mathbf{H}\boldsymbol{x}^f = \boldsymbol{y} - \mathbf{H}\boldsymbol{x}^t + \mathbf{H}(\boldsymbol{x}^t - \boldsymbol{x}^f)$$
- Because the observational error and the background error are uncorrelated, the variance of the innovation is $\mathbf{R} + \mathbf{H}\mathbf{P}^f\mathbf{H}^\top$.
- $\mathbf{R}$ is often assumed diagonal (no correlation between the errors of two observations): $\mathbf{R} = r\mathbf{I}$
- The innovation covariance between i and j: $rI_{ij} + \mathbf{H}_i\mathbf{P}^f\mathbf{H}_j^\top$ where $\mathbf{H}_i$ is the i-th row of $\mathbf{H}$.

How to determine the variance $\mathbf{P}^f$?
Example with $\mathbf{H} = \mathbf{I}$ and parameterized with SOAR

- Between i and j at distance d_{ij}, the covariance is

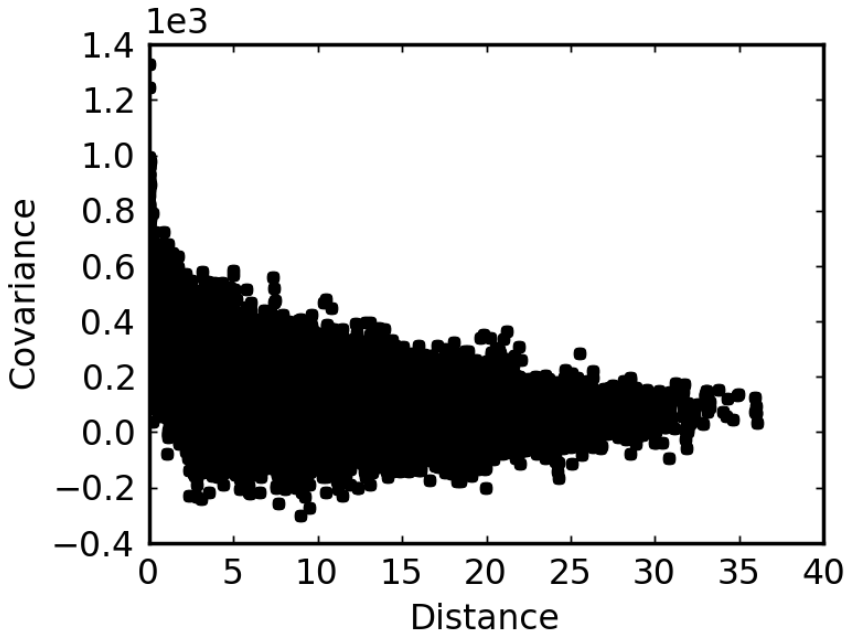
$$\mathrm{covar}(d_{ij}) = rI_{ij} + b\left(1 + \frac{d_{ij}}{L}\right)\exp\left(-\frac{d_{ij}}{L}\right)$$

where $I_{ij} = 1$ when $d_{ij} = 0$, and $I_{ij} = 0$ otherwise.



How to determine the variance $\mathbf{P}^f$ in practice?

- Computing the covariances between the innovations (time series)



- Fitting the parameters (with SOAR, b and L)

Example: H. J. Thiébaux, H. L. Mitchell, and D. W. Shantz (1986). "Horizontal Structure of Hemispheric Forecast Error Correlations for Geopotential and Temperature". In: *Monthly Weather Review* 114, pp. 1,048–1,066

Checking with the χ^2 diagnosis

Consistency between the innovations and their error statistics

- The variance of innovations is $\mathbf{R} + \mathbf{H}\mathbf{P}^f\mathbf{H}^\top$.
- One checks that
$$\chi^2 = \mathrm{E}\,(\boldsymbol{y} - \mathbf{H}\boldsymbol{x}^f)^\top(\mathbf{R} + \mathbf{H}\mathbf{P}^f\mathbf{H}^\top)^{-1}(\boldsymbol{y} - \mathbf{H}\boldsymbol{x}^f) = N_y$$
where N_y is the number of observations (i.e., the dimension of $\boldsymbol{y}$).
- Note that if the innovations are normally distributed, then χ^2 follows a chi-squared distribution with N_y degrees of freedom.
- Among other references, see

R. Ménard, S. E. Cohn, L.-P. Chang, and P. M. Lyster (2000). "Assimilation of stratospheric chemical tracer observations using a Kalman filter. Part I: Formulation". In: *Monthly Weather Review* 128, pp. 2,654–2,671