Image assimilation with the weighted ensemble Kalman filter

INRIA Rennes - Fluminance team
S. Beyou,S. Gorthi, E. Mémin
University of South Brittany
A. Cuzol

Ninth International Workshop on Adjoint Model Applications in Dynamic Meteorology
10-14 October 2011

Framework

Assimilation (filtering) for non linear and high-dimensional systems.

State-space model:

- Continuous stochastic dynamical model

$$\mathrm{d}\mathbf{x}(t) = f(\mathbf{x}(t))\mathrm{d}t + \sigma \mathrm{d}\mathbf{B}(t)$$

- Discrete-time observations (images)

$$\mathbf{y}(t_k) = g(\mathbf{x}(t_k)) + \gamma_{t_k}$$

Filtering problem

Filtering aims at estimating $p(\mathbf{x}_{t_k}|\mathbf{y}_{t_1:t_k})$ for all t_k.

Sequential Monte Carlo techniques:

- Ensemble Kalman filter:
  $\hat{p}(\mathbf{x}_{t_k}|\mathbf{y}_{t_1:t_k}) = \mathcal{N}(\mu_{t_k}, \Sigma_{t_k}) = \sum_{i=1}^{N} \delta_{\mathbf{x}_{t_k}^{(i)}}(\mathbf{x}_{t_k})$
- Particle filter:
  $\hat{p}(\mathbf{x}_{t_k}|\mathbf{y}_{t_1:t_k}) = \sum_{i=1}^{N} w_{t_k}^{(i)} \delta_{\mathbf{x}_{t_k}^{(i)}}(\mathbf{x}_{t_k})$

Filtering problem

Filtering aims at estimating $p(\mathbf{x}_{t_k}|\mathbf{y}_{t_1:t_k})$ for all t_k.

Sequential Monte Carlo techniques:

- Ensemble Kalman filter:
  $\hat{p}(\mathbf{x}_{t_k}|\mathbf{y}_{t_1:t_k}) = \mathcal{N}(\mu_{t_k}, \Sigma_{t_k}) = \sum_{i=1}^{N} \delta_{\mathbf{x}_{t_k}^{(i)}}(\mathbf{x}_{t_k})$
- Particle filter:
  $\hat{p}(\mathbf{x}_{t_k}|\mathbf{y}_{t_1:t_k}) = \sum_{i=1}^{N} w_{t_k}^{(i)} \delta_{\mathbf{x}_{t_k}^{(i)}}(\mathbf{x}_{t_k})$

Both are based on prediction and correction:

- EnKF: ensemble prediction (model), Kalman correction (Gaussian);
- Particle filter: importance sampling, particles weights correction.

Particle filter

- Prediction : importance sampling

$$\mathbf{x}_{t_k}^{(i)} \sim \pi(\mathbf{x}_{t_k}^{(i)} | \mathbf{x}_{t_0:t_{k-1}}, \mathbf{y}_{t_1:t_k})$$

- Correction : computation of importance weights

$$w_{t_k}^{(i)} \propto w_{t_{k-1}}^{(i)} \frac{p(\mathbf{y}_{t_k} | \mathbf{x}_{t_k}^{(i)}) p(\mathbf{x}_{t_k}^{(i)} | \mathbf{x}_{t_{k-1}}^{(i)})}{\pi(\mathbf{x}_{t_k}^{(i)} | \mathbf{x}_{t_0:t_{k-1}}^{(i)}, \mathbf{y}_{t_1:t_k})}$$

- Resampling

Particle filter

Advantages of particle filter:

- no Gaussian or linear hypotheses;
- theoretical convergence towards optimal Bayesian filter.

But particle filter in its simplest form:

- uses the transition $p(\mathbf{x}_{t_k}|\mathbf{x}_{t_{k-1}})$ as importance distribution
- $\Rightarrow$ not efficient for high-dimensional problems.

Particle filter

Advantages of particle filter:

- no Gaussian or linear hypotheses;
- theoretical convergence towards optimal Bayesian filter.

But particle filter in its simplest form:

- uses the transition $p(\mathbf{x}_{t_k}|\mathbf{x}_{t_{k-1}})$ as importance distribution
- $\Rightarrow$ not efficient for high-dimensional problems.

$\Rightarrow$ Weighted EnKF: tries to combine the efficiency of EnKF methods with the good properties of particle filters.

Weighted ensemble Kalman filter

- Idea of WEnKF: the importance distribution of the particle filter is given by the EnKF.

$$\pi(\mathbf{x}_{t_k}|\mathbf{x}_{t_0:t_{k-1}},\mathbf{y}_{t_1:t_k}) = \pi(\mathbf{x}_{t_k}|\mathbf{x}_{t_{k-1}},\mathbf{y}_{t_k}) = \mathcal{N}(\mu_{t_k},\Sigma_{t_k})$$

Weighted ensemble Kalman filter

- Idea of WEnKF: the importance distribution of the particle filter is given by the EnKF.

$$\pi(\mathbf{x}_{t_k}|\mathbf{x}_{t_{0:t_{k-1}}}, \mathbf{y}_{t_{1:t_k}}) = \pi(\mathbf{x}_{t_k}|\mathbf{x}_{t_{k-1}}, \mathbf{y}_{t_k}) = \mathcal{N}(\mu_{t_k}, \Sigma_{t_k})$$

- One WEnKF iteration, from $\hat{p}(\mathbf{x}_{t_{k-1}}|\mathbf{y}_{t_{1:t_{k-1}}})$ to $\hat{p}(\mathbf{x}_{t_k}|\mathbf{y}_{t_{1:t_k}})$:
  - Start with particles $\mathbf{x}_{t_{k-1}}^{(i),f}$ and weights $w_{t_{k-1}}^{(i)}$ for $i = 1, \ldots, N$
  - Prediction step of EnKF $\Rightarrow \mathbf{x}_{t_k}^{(i),f}$
  - Analysis step of EnKF $\Rightarrow \mathbf{x}_{t_k}^{(i),a}$
  - Computation of weights $w_{t_k}^{(i)}$
  - Resampling

Weighted ensemble Kalman filter

$\Rightarrow$ The WEnKF can be seen as:

- a particle filter with EnKF as importance distribution: guides particles towards observation, contrary to standard particle filters;
- an EnKF with ensemble weights $w_{t_k}^{(i)}$ for $i = 1 : N$: relaxation of the Gaussian assumption.

Weighted ensemble Kalman filter

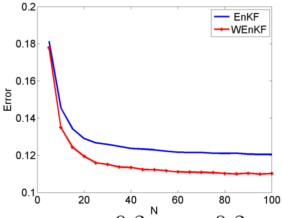
$\Rightarrow$ The WEnKF can be seen as:

- a particle filter with EnKF as importance distribution: guides particles towards observation, contrary to standard particle filters;
- an EnKF with ensemble weights $w_{t_k}^{(i)}$ for $i = 1 : N$: relaxation of the Gaussian assumption.

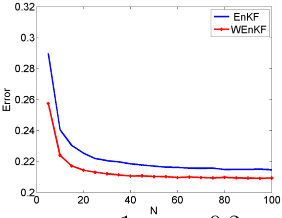
Data assimilation with the weighted ensemble Kalman filter. *Tellus Series A: Dynamical Meteorology and Oceanography, 2010* (N. Papadakis, E. Mémin, A. Cuzol, N. Gengembre)

Weighted ensemble Kalman filter

- Theoretical result: EnKF and particle filter do not have the same limit distribution (LeGland et al 2011).
- This can be observed in small dimension for a non linear model:



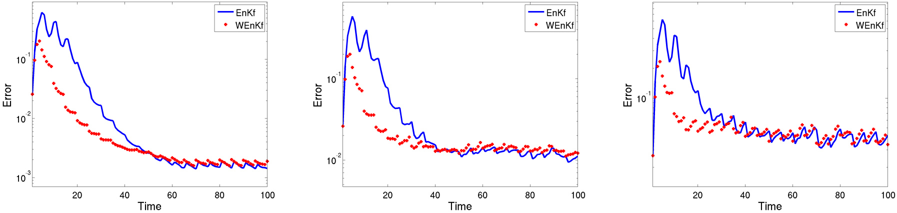
$\sigma_Q = 0.2, \sigma_R = 0.2$



$\sigma_Q = 1, \ \sigma_R = 0.2$

Weighted ensemble Kalman filter

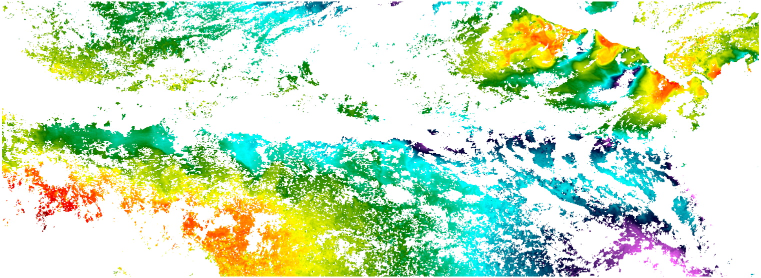
- High dimension: harder to highlight a difference in limit distributions.
- But WEnKF seems to converge faster than EnKF:



$\sigma_R = 0.1$, $\sigma_Q = 0.01$ $\sigma_R = 0.1$,$\sigma_Q = 0.05$ $\sigma_R = 0.1$,$\sigma_Q = 0.1$

SST image assimilation

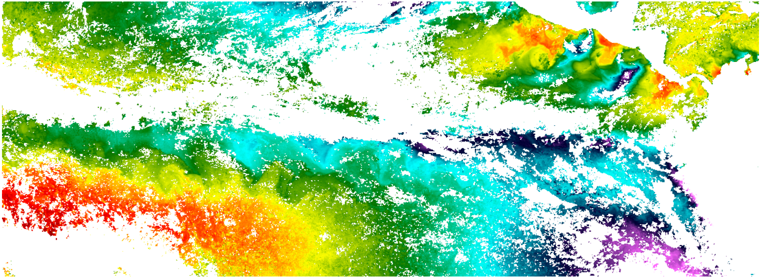
Practical application: assimilation of sea surface temperature images (El Niño phenomenon, January 2008).



05/01/2008

SST image assimilation

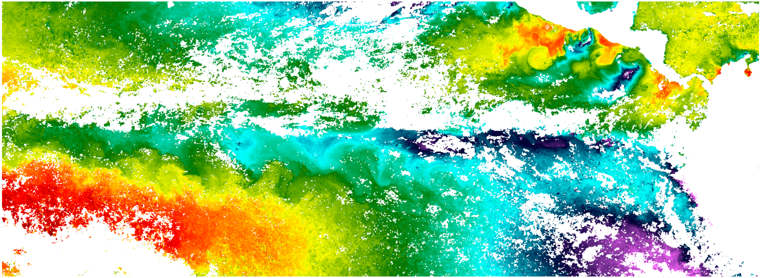
Practical application: assimilation of sea surface temperature images (El Niño phenomenon, January 2008).



06/01/2008

SST image assimilation

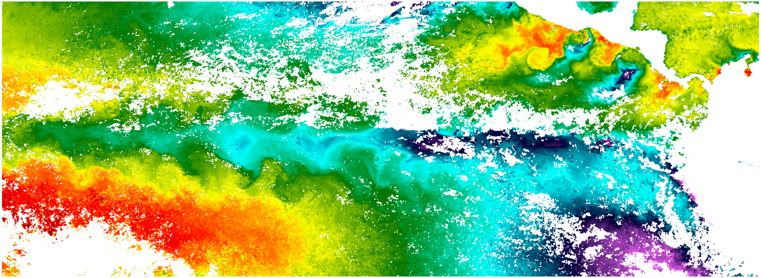
Practical application: assimilation of sea surface temperature images (El Niño phenomenon, January 2008).



07/01/2008

SST image assimilation

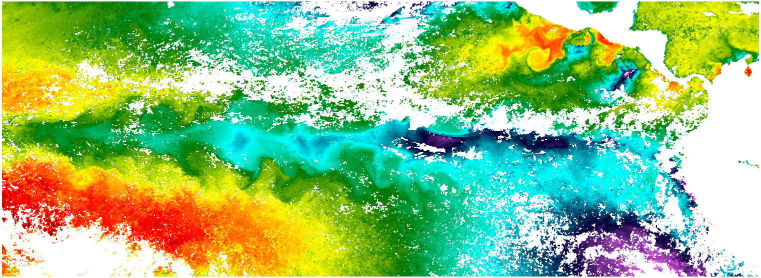
Practical application: assimilation of sea surface temperature images (El Niño phenomenon, January 2008).



08/01/2008

SST image assimilation

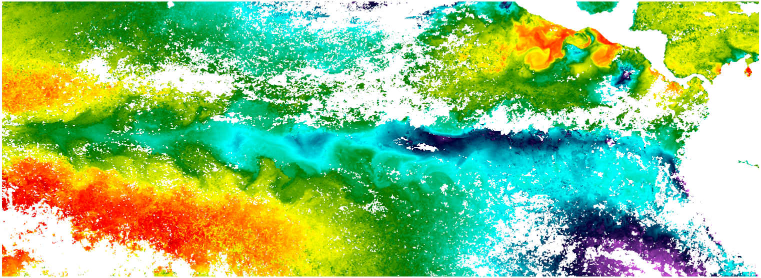
Practical application: assimilation of sea surface temperature images (El Niño phenomenon, January 2008).



09/01/2008

SST image assimilation

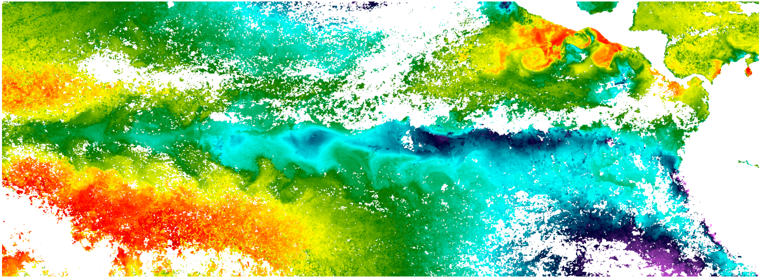
Practical application: assimilation of sea surface temperature images (El Niño phenomenon, January 2008).



10/01/2008

SST image assimilation

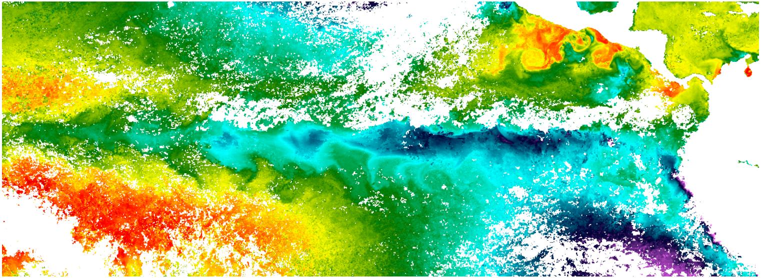
Practical application: assimilation of sea surface temperature images (El Niño phenomenon, January 2008).



11/01/2008

SST image assimilation

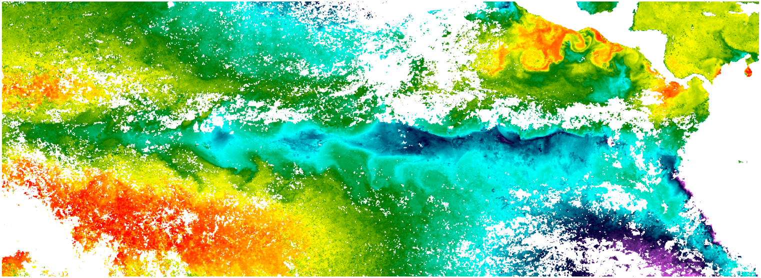
Practical application: assimilation of sea surface temperature images (El Niño phenomenon, January 2008).



12/01/2008

SST image assimilation

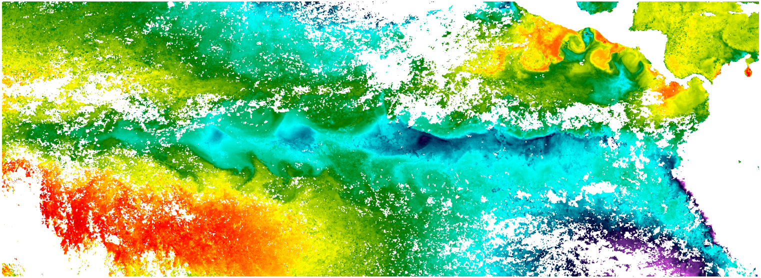
Practical application: assimilation of sea surface temperature images (El Niño phenomenon, January 2008).



13/01/2008

SST image assimilation

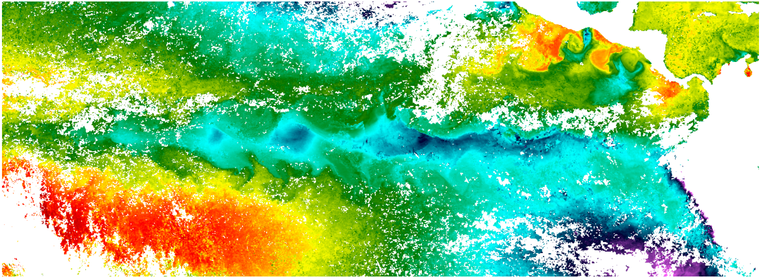
Practical application: assimilation of sea surface temperature images (El Niño phenomenon, January 2008).



14/01/2008

SST image assimilation

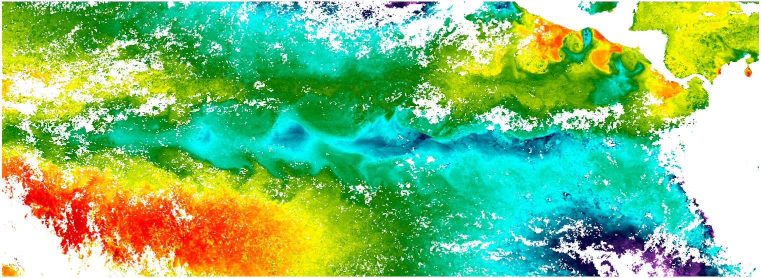
Practical application: assimilation of sea surface temperature images (El Niño phenomenon, January 2008).



15/01/2008

SST image assimilation

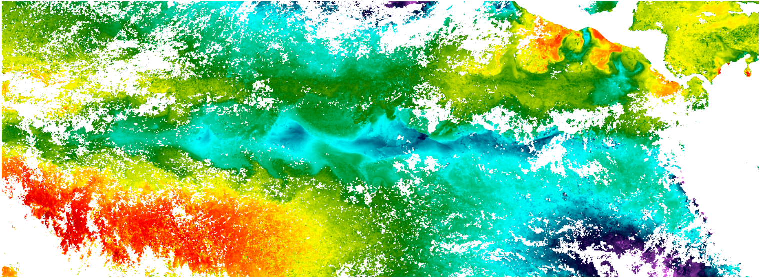
Practical application: assimilation of sea surface temperature images (El Niño phenomenon, January 2008).



16/01/2008

SST image assimilation

Practical application: assimilation of sea surface temperature images (El Niño phenomenon, January 2008).



17/01/2008

SST image assimilation - Model

- 2D velocity-vorticity dynamical model:

$$\mathrm{d}\boldsymbol{\xi}_t = -\nabla\boldsymbol{\xi}_t \cdot \mathbf{w}_t \mathrm{d}t + \nu\Delta\boldsymbol{\xi}_t \mathrm{d}t + \sigma \mathrm{d}\mathbf{B}_t$$

SST image assimilation - Model

- 2D velocity-vorticity dynamical model:

$$\mathrm{d}\boldsymbol{\xi}_t = -\nabla\boldsymbol{\xi}_t \cdot \mathbf{w}_t \mathrm{d}t + \nu\Delta\boldsymbol{\xi}_t \mathrm{d}t + \sigma \mathrm{d}\mathbf{B}_t$$

- Model perturbations:
  Gaussian random fields with covariance $\Sigma = \sigma\sigma^T$ (exponential covariance $\Sigma(\mathbf{x}_i, \mathbf{x}_j) = \eta \exp(-\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{\lambda}))$

SST image assimilation - Model

- Observation models:
  - Linear (external estimator $\tilde{\boldsymbol{\xi}}$) :

$$\tilde{\boldsymbol{\xi}}_{t_k} = \boldsymbol{\xi}_{t_k} + \boldsymbol{\gamma}_{t_k}$$

  - Non linear (directly from image data I) :

$$I(x, t_k) = I(x + \mathbf{d}(x), t_{k+1}) + \boldsymbol{\gamma}_{t_k}(x)$$

SST image assimilation - Details

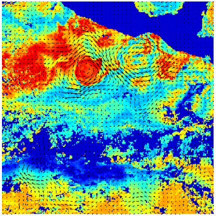
- ETKF is used as proposal step
- Non linear observation: $H\mathbf{x}_k$ replaced by $H(\mathbf{x}_k)$
- 48 images 256*256 (spatial resolution: 10km)
- Temporal resolution: one day
- Missing data $\Rightarrow$ high observation noise

SST image assimilation - Details

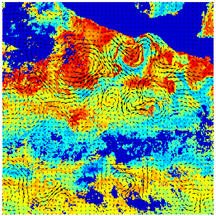
- ETKF is used as proposal step
- Non linear observation: $H\mathbf{x}_k$ replaced by $H(\mathbf{x}_k)$
- 48 images 256*256 (spatial resolution: 10km)
- Temporal resolution: one day
- Missing data $\Rightarrow$ high observation noise

Analysis of SST images by weighted ensemble transform Kalman filter. *IGARSS'11*(S.Beyou, S. Gorthi, E. Mémin)
Weighted ensemble transform Kalman filter for image assimilation. *In preparation* (S.Beyou, A. Cuzol, S. Gorthi, E. Mémin)

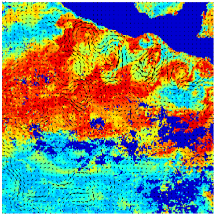
SST image assimilation - Results



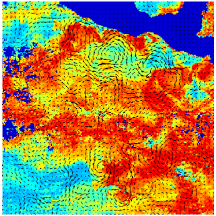
SST and velocity - Day 1



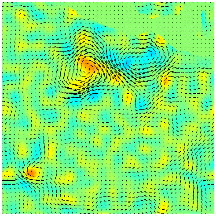
Day 10



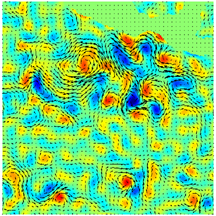
Day 39



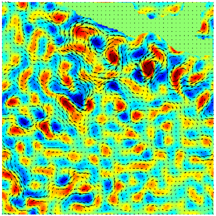
Day 48



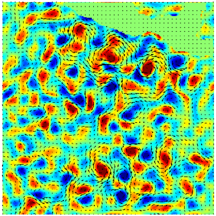
Vorticity and velocity - Day 1



Day 10



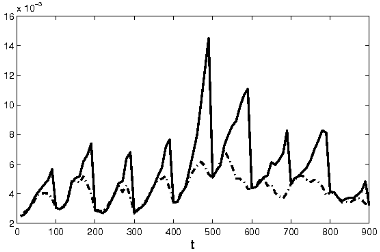
Day 39



Day 48

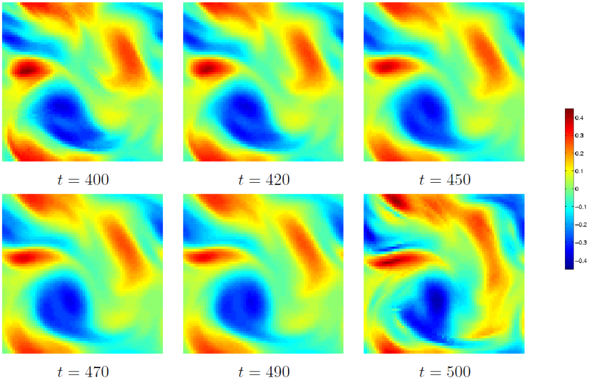
Filtering discontinuities

WEnKF (as EnKF) leads to temporal discontinuities (correction at observation times only):



Filtering discontinuities

Illustration for a given time interval between two observations:



Sequential trajectories smoothing

- Using conditional simulation of diffusions (Delyon et al 2006), one can sample new trajectories between t_{k-1} and t_k, once $\mathbf{y}_{t_k}$ is known.
  $\Rightarrow$ For each pair $\{\mathbf{x}_{t_{k-1}}^{(i)}, \mathbf{x}_{t_k}^{(i)}\}$, $i = 1, \ldots, N$, compute:

$$p(\mathbf{x}_t|\mathbf{x}_{t_{k-1}}^{(i)}, \mathbf{x}_{t_k}^{(i)}) \quad \text{for all} \quad t \in [t_{k-1}, t_k]$$

Sequential trajectories smoothing

- Using conditional simulation of diffusions (Delyon et al 2006), one can sample new trajectories between t_{k-1} and t_k, once $\mathbf{y}_{t_k}$ is known.
  $\Rightarrow$ For each pair $\{\mathbf{x}_{t_{k-1}}^{(i)}, \mathbf{x}_{t_k}^{(i)}\}$, $i = 1, \dots, N$, compute:

$$p(\mathbf{x}_t | \mathbf{x}_{t_{k-1}}^{(i)}, \mathbf{x}_{t_k}^{(i)}) \quad \text{for all} \quad t \in [t_{k-1}, t_k]$$

- The smoothing distribution writes:

$$\hat{p}(\mathbf{x}_t | \mathbf{y}_{t_1:t_k}) = \sum_{i=1}^{N} w_{t_k}^{(i)} p(\mathbf{x}_t | \mathbf{x}_{t_{k-1}}^{(i)}, \mathbf{x}_{t_k}^{(i)}) \quad \text{for all} \quad t \in [t_{k-1}, t_k]$$

  - Based on WEnKF trajectories weights;
  - No linearization or Gaussian assumption;
  - Respects the state model.

Sequential trajectories smoothing

- Using conditional simulation of diffusions (Delyon et al 2006), one can sample new trajectories between t_{k-1} and t_k, once $\mathbf{y}_{t_k}$ is known.
  $\Rightarrow$ For each pair $\{\mathbf{x}_{t_{k-1}}^{(i)}, \mathbf{x}_{t_k}^{(i)}\}$, $i = 1, \ldots, N$, compute:

$$p(\mathbf{x}_t | \mathbf{x}_{t_{k-1}}^{(i)}, \mathbf{x}_{t_k}^{(i)}) \quad \text{for all} \quad t \in [t_{k-1}, t_k]$$

- The smoothing distribution writes:

$$\hat{p}(\mathbf{x}_t | \mathbf{y}_{t_1:t_k}) = \sum_{i=1}^{N} w_{t_k}^{(i)} p(\mathbf{x}_t | \mathbf{x}_{t_{k-1}}^{(i)}, \mathbf{x}_{t_k}^{(i)}) \quad \text{for all} \quad t \in [t_{k-1}, t_k]$$

  - Based on WEnKF trajectories weights;
  - No linearization or Gaussian assumption;
  - Respects the state model.

Monte Carlo fixed lag smoothing in state-space models. *To be submitted* (A. Cuzol, E. Mémin)

Sequential trajectories smoothing

Illustration for a given time interval between two observations: