Applications to Nonlinear Filtering and Data Assimilation

Oana Lang

Babeș-Bolyai University, Cluj-Napoca, Romania

Faculty of Mathematics and Computer Science

UNIVERSITATEA BABEŞ-BOLYAI
BABEŞ-BOLYAI TUDOMÁNYEGYETEM
BABEŞ-BOLYAI UNIVERSITÄT
BABEŞ-BOLYAI UNIVERSITY
TRADITIO ET EXCELLENTIA

STOCHASTICA

Contents

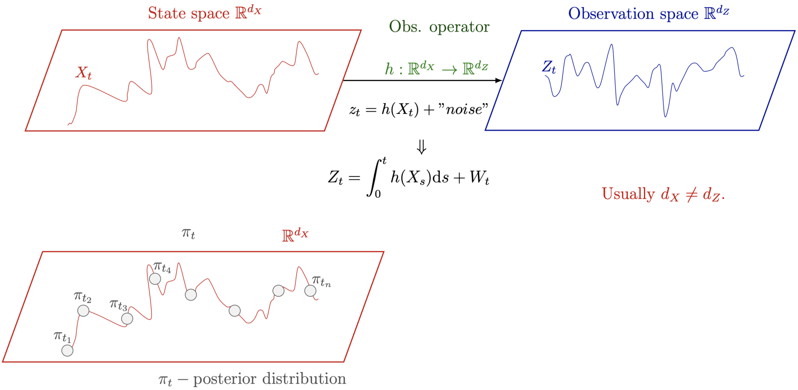
1. Introduction

Framework and Notations

- X (signal/truth) and Z (observation) are two stochastic processes defined on the probability space $(\Omega, \mathcal{F}, \mathbb{P})$.
- The pair of processes (X, Z) forms the basis of the nonlinear filtering problem which consists in finding the best approximation of the posterior distribution of the signal X_t given the observations $Z_1, Z_2, \ldots, Z_t$.
- π_t **is the posterior distribution of the signal at time** t.
- In our applications, X is the pathwise solution of either the Lorenz '63 model or a stochastic version of the rotating shallow water system.

For details on the nonlinear filtering problem see [4] D. Crisan & A. Bain, *Fundamentals of Stochastic Filtering*.

The Stochastic Filtering Problem



The process of using partial observations and a stochastic model to make inferences about an evolving dynamical system.

The Stochastic Filtering Problem

- X the signal process - "hidden component"
- Z the observation process - "the data"

The filtering problem: Find the conditional distribution of the *signal* X_t given $\mathcal{Z}_t = \sigma(Z_s, s \in [0,t])$, i.e.

$$\pi_t(A) = \mathbb{P}(X_t \in A | \mathcal{Z}_t), \qquad t \geq 0, \qquad A \in \mathcal{B}(\mathbb{R}^{d_X}).$$

Discrete framework:

$$(X_t)_{t\geq 0} \ \text{ Markov chain } \ \mathbb{P}(X_t \in A | X_{t-1} = x_{t-1}) = \mathcal{K}_t(x_{t-1}, A)dt,$$
$$(X_t, Z_t)_{t\geq 0} \quad \mathbb{P}(Z_t \in dz | X_t = x_t) = g_t(z|x_t)dz$$

Continuous framework:

$$dX_t = f(X_t)dt + \sigma(X_t)dW_t,$$
$$dZ_t = h(X_t)dt + dV_t.$$

The Stochastic Filtering Problem

- Let $\mathcal{F}_t^X$ be the filtration generated by the signal process $X = (X_t)_t$, that is
$$\mathcal{F}_t^X \triangleq \sigma(X_s,\ s \in [0,t]).$$
- We assume that X is a Markov process. That is, for all $t \in \mathbb{N}$ and $A \in \mathcal{B}(\mathbb{R}^d)$,
$$\mathbb{P}\left(X_t \in A \mid \mathcal{F}_{t-1}^X\right) = \mathbb{P}\left(X_t \in A \mid X_{t-1}\right).$$
- Let $\mathcal{K}_t$ be its transition kernel:
$$\mathcal{K}_t : \mathbb{R}^{d_X} \times \mathcal{B}(\mathbb{R}^{d_X}) \to [0,1],\ \mathcal{K}_t(x,B) = \mathbb{P}(X_t \in B | X_{t-1} = x_{t-1})$$
for any Borel measurable set $B \in \mathcal{B}(\mathbb{R}^{d_X})$ and $x_{t-1} \in \mathbb{R}^{d_X}$. The transition kernel $\mathcal{K}_t$ is required to have the following properties:
   i. $\mathcal{K}_t(x,\cdot)$ is a probability measure on $(\mathbb{R}^{d_X}, \mathcal{B}(\mathbb{R}^{d_X}))$, for all $t \in \mathbb{N}$ and $x \in \mathbb{R}^{d_X}$.
   ii. $\mathcal{K}_t(\cdot, A) \in \mathcal{B}_b(\mathbb{R}^{d_X})$, for all $t \in \mathbb{N}$ and $A \in \mathcal{B}(\mathbb{R}^{d_X})$.

The Stochastic Filtering Problem

- The process Z models noisy measurements of the truth, using the *observation operator* $h : \mathbb{R}^{d_X} \to \mathbb{R}^{d_Z}$:

$$Z_t = h(X_t) + V_t$$

  where $(V_t)_{t \geq 0}$ are i.i.d. random variables which represent the measurement noise, and h is a Borel-measurable function.

The Stochastic Filtering Problem

- The observations are incorporated into the system at *assimilation times*. In particle filtering, an ensemble of particles is evolved between assimilation times according to the law of the signal.
- At each assimilation time the observation is incorporated into the system through the *likelihood function*:

$$g_t^{z_t} : \mathbb{R}^{d_X} \to \mathbb{R}_+, \; g_t^{z_t}(x) = g_t(z_t - h(x_t)) = \mathbb{P}(Z_t \in dz_t | X_t = x_t)$$

$$\int_A g(z_t - h(x_t)) dz_t = \mathbb{P}(Z_t \in A | X_t = x_t),$$

where $A \in \mathcal{B}(\mathbb{R}^{d_Z})$ the σ-algebra of Borel measurable sets on $\mathbb{R}^{d_Z}$.

- The following recursion formula holds (see [4])

$$\pi_t = g_t \star \pi_{t-1} \mathcal{K}_t$$

where by '$\star$' we denoted the *projective product*.

The Stochastic Filtering Problem

- Schematically, the recursion formula $\pi_{t-1} \longrightarrow \pi_t$ can be described as

$$\pi_{t-1}^{a,z_{0:t-1}} \xrightarrow[\substack{model\\ forecast\\ prediction}]{\mathcal{K}_t} \pi_{t-1}^{a,z_{0:t-1}}\mathcal{K}_t =: \pi_t^b =: p_t \xrightarrow[\substack{assimilation\\ analysis\\ update}]{tempering,\ g_t^{z_t}\star} g_t^{z_t} \star \pi_t^b = \pi_t^{a,z_{0:t}}.$$

- We use the superscripts $z_{0:t-1} := (z_0, \ldots, z_{t-1})$ and $z_{0:t} := (z_0, \ldots, z_t)$ to emphasize the dependence on the fixed data.
- The indices a and b stand for *analysis* and *background*, respectively.
- $p_t := p_t^{Z_{0:t-1}}$ is the predictive distribution of the signal, that is the distribution of the signal X_t given the observations $Z_0, Z_1, \ldots, Z_{t-1}$.

The Stochastic Filtering Problem

- By the definition of the projective product, the recursion formula can be written in the more familiar form:

$$\pi_t(B) = \frac{\int_B g_t^{z_t}(x_t) p_t(dx_t)}{\int_{\mathbb{R}^{d_X}} g_t^{z_t}(x_t) p_t(dx_t)} = \beta_t^{-1} \int_B g_t^{z_t}(x_t) p_t(dx_t)$$

where $B \in \mathcal{B}(\mathbb{R}^{d_X})$ and $\beta_t := \int_{\mathbb{R}^{d_X}} g_t^{z_t}(x_t) p_t(dx_t)$ is a normalising constant.

The Stochastic Filtering Problem

Notations:

- posterior measure: the conditional distribution of the *signal* X_t given $\mathcal{Z}_t$

$$\pi_t\,(A) = \mathbb{P}(X_t \in A|\mathcal{Z}_t), \qquad t \geq 0, \qquad A \in \mathcal{B}(\mathbb{R}^d).$$

- predictive measure: the conditional distribution of the *signal* X_t given

$\mathcal{Z}_{t-1}$

$$p_t\,(A) = \mathbb{P}(X_t \in A|\mathcal{Z}_{t-1}), \qquad t \geq 0, \qquad A \in \mathcal{B}(\mathbb{R}^d).$$

- prior distribution: the distribution of the *signal* X_t

$$q_t(A) = \mathbb{P}(X_t \in A), \qquad t \geq 0, \qquad A \in \mathcal{B}(\mathbb{R}^d).$$

The Stochastic Filtering Problem

Bayes' recursion formula

The posterior distribution satisfies the following recursion formula ([4])

Prediction	$p_t = \mathcal{K}_t \pi_{t-1}$
Updating	$\pi_t = g_t \star p_t$

In other words, $\frac{d\pi_t}{dp_t} = C_t^{-1} g_t$, where $C_t \triangleq \int_{\mathbb{R}^{d_X}} g_t\left(z_t, x_t\right) p_t\left(dx_t\right)$.

$$\pi_{t-1} \xrightarrow[\substack{model \\ forecast \\ prediction}]{\mathcal{K}_t} \mathcal{K}_t \pi_{t-1} =: p_t \xrightarrow[\substack{assimilation \\ analysis \\ update}]{\textbf{non-linear}:\ g_t \star} g_t \star p_t = \pi_t$$

Particle Filters

- Signal: $X \to$ pathwise solution of an SPDE (for us either Lorenz '63 or SALT-SRSW)
- Observation process: Z
- π_t = posterior distribution of X_t given $Z_1, Z_2, \ldots Z_t \to$ approximated using a set of particles i.e. random measures:

$$\pi_t \approx \pi_t^N = \sum_{\ell=1}^{N} \mathrm{w}_t^\ell \delta(x_t^\ell)$$

  - $\mathrm{w}_t^1, \mathrm{w}_t^2, \ldots \to$ weights of the particles
  - $x_t^1, x_t^2, \ldots \to$ positions of the particles

Particle Filters

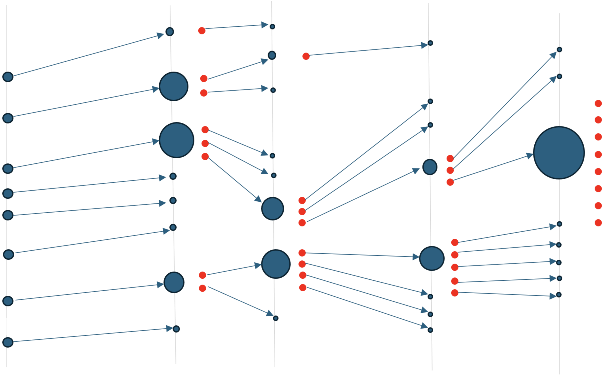
- We model the evolution of the signal discretely in time through a map $\mathcal{M}_t : \mathbb{R}^{dx} \to \mathbb{R}^{dx}$
- If $x^\ell_{t_1}$, $x^\ell_{t_2}$ is the position of the particle ℓ at time t_1 respectively t_2 then

$$x^\ell_{t_2} = \mathcal{M}_{t_2}(x^\ell_{t_1}).$$

and

$$\pi^a_{t_2} = \sum_{\ell=1}^{N} \mathrm{w}^\ell_{t_2} \delta(\mathcal{M}_{t_2}(x^\ell_{t_1})) = \sum_{\ell=1}^{N} \mathrm{w}^\ell_{t_2} \delta(x^\ell_{t_2}).$$

Particle Filters



Classical Particle Filter
The resampling procedure ensures that particles with low weights are replaced with particles with higher weights. Following the resampling, an ensemble of equal-weighted particles is obtained. In high-dimensional spaces: one particle gaining a weight close to one while all the others have weights close to zero and therefore are discarded.

2. Nonlinear Signals

2.1. A Stochastic Rotating Shallow Water (SRSW) Model

Rotating Shallow Water Model (deterministic)

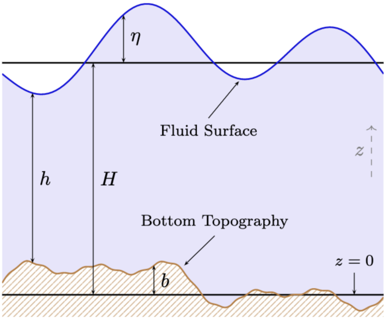


Figure 1: Illustration of the variables in the shallow water model. The scalar function h denotes the height of the fluid column and the scalar value H is the average height of the fluid column over the domain. We denote by η the scalar function that gives the elevation of the fluid surface relative to H and the scalar function b is the bottom topography. Thus, the z-coordinate of the fluid surface is given by $H + \eta = h + b$.

Picture produced by Alex Lobbe. See [2].

Rotating Shallow Water Model (deterministic)

The inviscid model:

$$\frac{D}{Dt}u_t + f\hat{z} \times u_t + g\nabla h_t = 0 \tag{1a}$$

$$\frac{\partial h_t}{\partial t} + \nabla \cdot (h_t u_t) = 0 \tag{1b}$$

- $\frac{D}{Dt} := \frac{\partial}{\partial t} + u \cdot \nabla$ is the material derivative.
- $u = (u^1, u^2)$ is the horizontal fluid velocity vector field
- h is the height of the fluid column
- f is the Coriolis parameter, $f = 2\Theta \sin\varphi$ where Θ is the rotation rate of the Earth and φ is the latitude; $f\hat{z} \times u = (-fu^2, fu^1)$, where $\hat{z}$ is a unit vector pointing away form the centre of the Earth; g is the gravitational acceleration

We can formally re-write a viscous version of the RSW system: $X := (u, h)$ and then

$$dX_t + F\left(X_t\right)dt = 0 \tag{2}$$

where $F\left(X_t\right)$ denotes

$$F\begin{pmatrix} u \\ h \end{pmatrix} = \begin{pmatrix} u \cdot \nabla u + f\hat{z} \times u + g\nabla h - \nu\Delta u \\ \nabla \cdot (hu) - \eta\Delta h \end{pmatrix}. \tag{3}$$

Nonlinear Stochastic Transport

Example: vorticity equation

- deterministic transport: the Lie form of the vorticity equation contains a Lie derivative which expresses the change of vorticity along the flow generated by the velocity vector field:

$$\partial_t \omega_t + \mathcal{L}_{u_t} \omega_t = 0 \quad \Leftrightarrow \quad d\omega_t + u_t \cdot \nabla \omega_t dt = 0$$

- vorticity: $\omega_t = curl\ u_t = \nabla \times u_t$
- stochastic transport: perturb the velocity vector field and investigate the case where vorticity is transported along the newly perturbed trajectory (Stochastic Advection by Lie Transport, Holm, 2015):

$$dy_t := u_t dt + \sum_i \xi_i \circ dW_t^i.$$

$$d\omega_t + u_t \cdot \nabla \omega_t dt + \sum_i \xi_i \cdot \nabla \omega_t \circ dW_t^i = 0$$

- $(\xi_i)_i$ vector fields:
  - divergence-free, time-independent, derived from the underlying physics
  - improved representation of the missing physics
  - induce variability in the particle filter ensemble

A Stochastic Rotating Shallow Water Model

Stochastic version:

$$du_t + \left[u_t \cdot \nabla u_t + f\hat{z} \times u_t + g\nabla h_t\right]dt + \sum_{i=1}^{\infty} \left[(\mathcal{L}_i + \mathcal{A}_i)u_t\right] \circ dW_t^i = \nu \Delta u_t dt \tag{4a}$$

$$dh_t + \nabla \cdot (h_t u_t)dt + \sum_{i=1}^{\infty} \left[\nabla \cdot (\xi_i h_t)\right] \circ dW_t^i = \eta \Delta h_t dt \tag{4b}$$

where ξ_i are divergence-free and time-independent vector fields,

$$\mathcal{L}_i u := \xi_i \cdot \nabla u,\ \ \mathcal{A}_i u := u_j \nabla \xi_i^j = \sum_{j=1}^{2} u_j \nabla \xi_i^j.$$

Re-written as:

$$dX_t + F\left(X_t\right)dt + \sum_{i=1}^{\infty} \mathcal{G}_i\left(X_t\right) \circ dW_t^i = 0 \tag{5}$$

where W^i are independent Brownian motions, and $\mathcal{G}_i$ are differential operators:

$$\mathcal{G}_i(X) = \mathcal{G}_i \begin{pmatrix} u \\ h \end{pmatrix} = \begin{pmatrix} \mathcal{L}_i u + \mathcal{A}_i u \\ \mathcal{L}_i h \end{pmatrix}.$$

A Stochastic Rotating Shallow Water Model

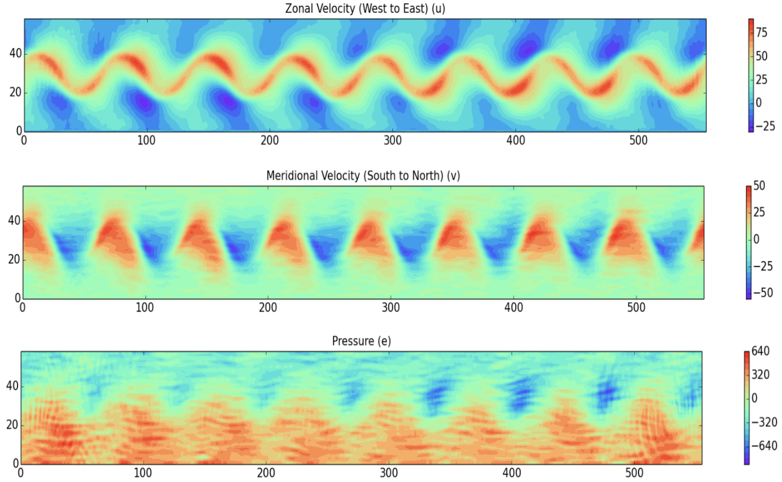
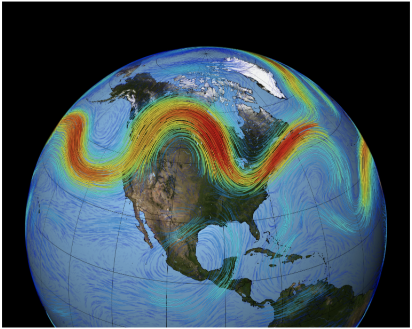


Figure 2: Instances of the zonal (west-east) velocity field, the meridional (south-north) velocity field, and the pressure field, for the SRSW model, after 200 time steps. The distance between any two grid points in both x and y direction is 50 km.

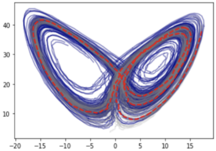
A Stochastic Rotating Shallow Water Model



Real jetstream, from NASA at https://svs.gsfc.nasa.gov/3864

2.2. The Lorenz '63 (L63) Model

The L63 Model

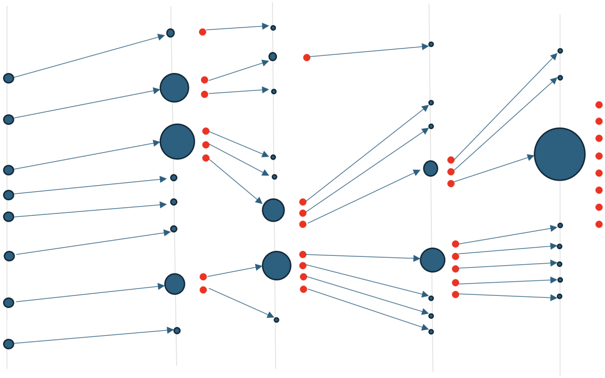


- L63 model:

$$dx = \alpha(y - x)dt + \epsilon dW_t^1$$
$$dy = ((\beta - z)x - y)\,dt + \epsilon dW_t^2$$
$$dz = (xy - \gamma z)\,dt + \epsilon dW_t^3$$

- α, β, γ are real positive parameters.
- Well-known for the broad spectrum of patterns displayed for different values of α, β, γ and its *butterfly attractor*. The original values chosen by Lorenz were $\alpha = 10, \beta = 28$ and $\gamma = \frac{8}{3}$.
- Implemented here using a Runge-Kutta scheme of order 4, with initial conditions $x_0 = 1.508870, y_0 = -1.531271, z_0 = 25.46091$.
- We first use the Runge-Kutta scheme to implement the three deterministic equations, then we generate a random field and perturb the system in a manner which is similar to the one described at "Jittering" ($\rho = 0.99$ here).

3. Tempering, Jittering, Calibration



Tempering

- In order to quantify the spread of the weights with respect to the posterior, we use the *effective sample size* statistic:

$$ess(\mathrm{w}) = \frac{1}{\sum_{\ell=1}^{N} (\mathrm{w}^{\ell})^2}$$

- If the ess is smaller than a representative threshold $N_{threshold} \leftrightarrow$ particles have drifted in a 'wrong' direction and many of them have small weights.
- The *ess* will be chosen to be above a certain threshold in order to keep the ensemble of particles in the right place $\leftrightarrow$ particles have comparable weights.
- Tempering [1]: flatten (gradually) the likelihood distribution such that $N_{threshold}$ is attained, then resample $\Rightarrow$ a more diverse ensemble of particles which are samples corresponding to a sequence of *altered* distributions $\rightarrow$ repeat until the original distribution is recovered.

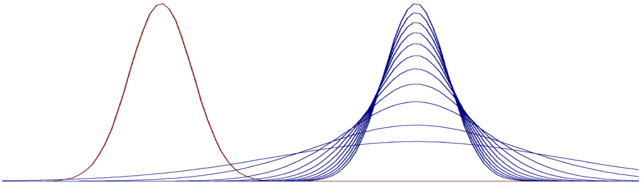
[1]Originally introduced in [5].

Tempering

- This is performed using a sequence of temperatures $0 = \phi_0 < \phi_1 < \ldots \phi_R = 1$ to ensure that the ess remains above the chosen threshold. Once the temperature is chosen, a resampling procedure is applied. The output is a sequence of *tempered posterior distributions* with corresponding normalised tempered weights.
- Intermediate tempered posterior distribution at the r^{th} tempering step:

$$\pi_t^r(B) = \frac{\int_B \left(g_t^{z_t}(x_t)\right)^{\phi_r} p_t(dx_t)}{\int_{\mathbb{R}^{d_X}} \left(g_t^{z_t}(x_t)\right)^{\phi_r} p_t(dx_t)}$$

for any $B \in \mathcal{B}(\mathbb{R}^{d_X})$.



Tempering

- Each intermediate step incorporates a resampling procedure followed by a jittering one. If we denote by $\mathbf{x}^r = (x^{\ell,r})_{\ell=1}^N$ the positions of the ensemble of particles at the beginning of the intermediate step $r = 0, 1, \ldots R$, then at the r-step we resample from

$$\pi_{t_i}^{r,N} = \frac{\sum_{\ell=1}^{N} \mathrm{w}_{t_i}^{r,\ell}(\phi_r, \mathbf{x}^r)\delta(x_{t_i}^{\ell})}{\sum_{\ell=1}^{N} \mathrm{w}_{t_i}^{r,\ell}(\phi_r, \mathbf{x}^r)},$$

  where $\mathrm{w}_{t_i}^{r,\ell}(\phi_r, \mathbf{x}^r) = \left(g_{t_i}^{z_{t_i}}(x_{t_i}^{\ell})\right)^{\phi_r - \phi_{r-1}}$,
- The corresponding effective sample size of $\pi_{t_i}^{r,N}$ is controlled by a suitable choice of the temperature increment $\phi_r - \phi_{r-1}$.
- $(\pi_t^{r,N})_N$ is an approx. for π_t^r.

Jittering

- Without jittering we have

$$x_{t_2}^{\ell} = \mathcal{M}(x_{t_1}^{\ell})$$

which can also be written as

$$x_{t_2}^{\ell} = \mathcal{M}(x_{t_1}^{\ell}, W(t_1 : t_2))$$

where W is the driving Brownian motion of the model and $W(t_1 : t_2)$ is the Brownian path between t_1 and t_2.

- Then the prior distribution is given by

$$p_{t_2} = \frac{1}{N} \sum_{\ell=1}^{N} \delta(x_{t_2}^{\ell}).$$

- **Issue**: after resampling the particles end up in the same place and we can have a large number of duplicates. $\Rightarrow$ An artificial predictive step is applied, through a Metropolis-Hastings procedure called jittering. This increases the spread of the ensemble whilst keeping the approximation asymptotically consistent.

Jittering

- We modify the last part of the particle trajectory using a *jittering parameter* ρ and a Gaussian random variable Z which is orthogonal to W.

- The new dynamics is given by

$$\tilde{x}^{\ell}_{t_2} = \mathcal{M}_{t_2}\left(x^{\ell}_{t_1}, \rho W(t_1 : t_2) + \sqrt{1-\rho^2} Z(t_1 : t_2)\right)$$

  where

$$x^{\ell}_{t_i} = \mathcal{M}_{t_i}\left(x^{\ell}_{t_{i-1}}, W(t_{i-1} : t_i)\right).$$

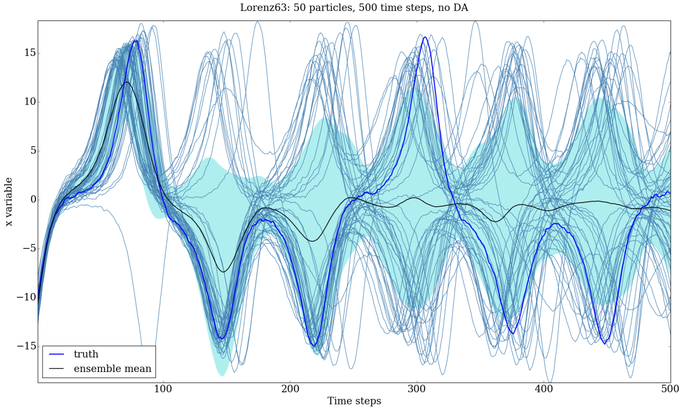
See [1] OL, D. Crisan, P. J. van Leeuwen, R. Potthast, *Bayesian Inference for Fluid Dynamics: A Case Study for the Stochastic Rotating Shallow Water Model*, Frontiers in Applied Mathematics and Statistics, 8 (2022).

Tempering and Jittering: The Algorithm

- $t = 0$: Sample N particles from the prior distribution.
- $(t_{i-1}, t_i]$: We have an ensemble $\mathbf{x}$ of particles with positions $(x^\ell_{t_{i-1}})_\ell$ where $x^\ell_{t_i} = \mathcal{M}(x^\ell_{t_{i-1}}, W(t_{i-1} : t_i))$. We want to assimilate observational data z_{t_i} in order to obtain a new ensemble $(x^\ell_{t_i})_\ell$ that defines $\pi^N_{t_i}$:
  - Evolve $x^\ell_{t_{i-1}} \xrightarrow[SRSW, L63]{SPDE} x^\ell_{t_i}$.
  - Set temperature $\phi = 1$.
  - While $ess_i(\phi, \mathbf{x}) < N_{threshold}$ do
    - Find $\phi' \in (1-\phi, 1)$ such that $ess_i(\phi' - (1-\phi), \mathbf{x}) \approx N_{threshold}$. Resample according to $\mathrm{w}^\ell_i(\phi' - (1-\phi), \mathbf{x})$ and apply MCMC with jittering if required (i.e. if there are duplicates) $\Rightarrow$ a new ensemble $\mathbf{x}(\phi')$.
    - Set $\phi = 1 - \phi'$ and $\mathbf{x} = \mathbf{x}(\phi)$.
  - If $ess_i \geq N_{threshold}$ then Stop and go to the $(i+1)^{th}$ filtering step with $(x^\ell_{t_i}, \mathrm{w}^\ell_i)_\ell$.
- Jittering: new dynamics given by

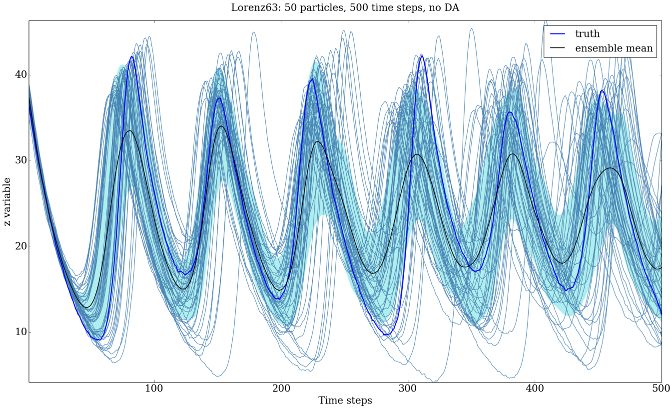
$$\tilde{x}^\ell_{t_2} = \mathcal{M}\left(x^\ell_{t_1}, \rho W(t_1 : t_2) + \sqrt{1-\rho^2} Z(t_1 : t_2)\right).$$

Tempering and Jittering for the Lorenz '63 Model



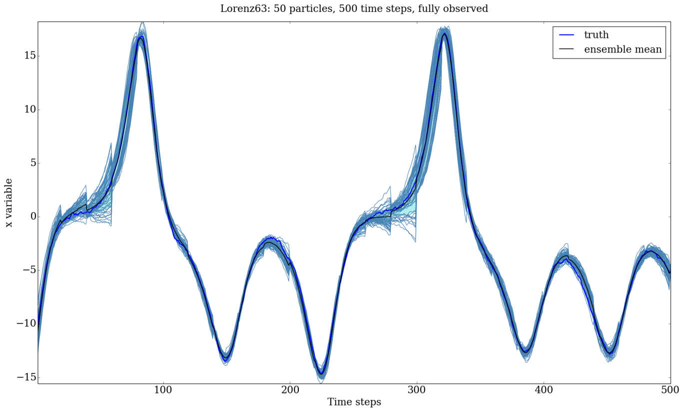
Evolution of the Lorenz '63 model for 500 time steps without any data assimilation, x variable.

Tempering and Jittering for the Lorenz '63 Model



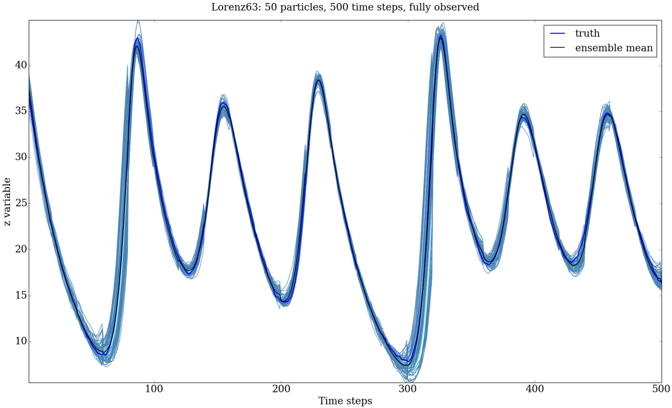
Evolution of the Lorenz '63 model for 500 time steps without any data assimilation, z variable.

Tempering and Jittering for the Lorenz '63 Model



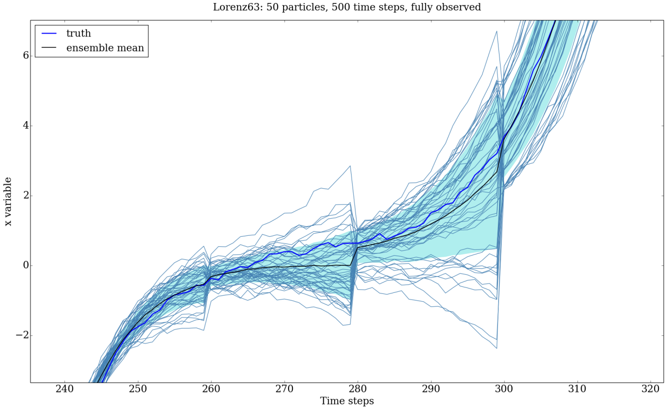
Evolution of the Lorenz '63 model for 500 time steps, all 3 variables are observed every 20 time steps. The initial uncertainty and the observational uncertainty are both equal to 1, the model error is equal to 0.1. Uncertainty is substantially reduced at assimilation times, x variable.

Tempering and Jittering for the Lorenz '63 Model



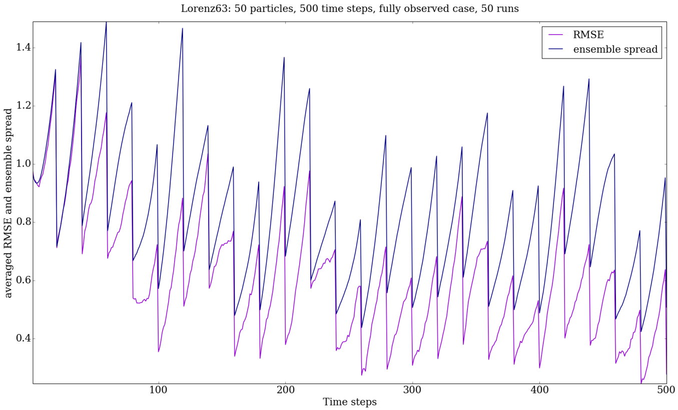
Evolution of the Lorenz '63 model for 500 time steps, all 3 variables are observed every 20 time steps. The initial uncertainty and the observational uncertainty are both equal to 1, the model error is equal to 0.1. Uncertainty is substantially reduced at assimilation times, z variable.

Tempering and Jittering for the Lorenz '63 Model



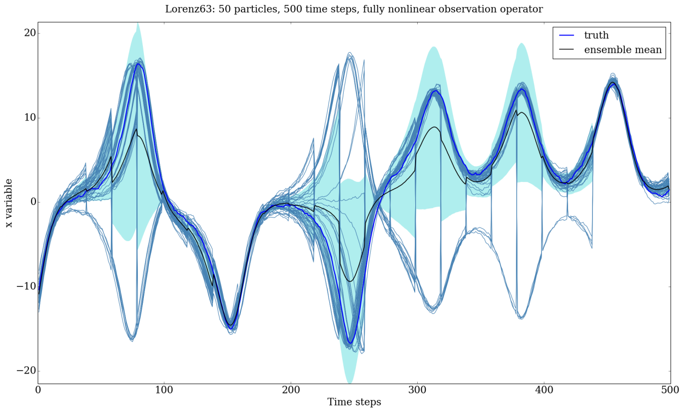
Evolution of the Lorenz '63 model for 500 time steps, all 3 variables are observed every 20 time steps. The initial uncertainty and the observational uncertainty are both equal to 1, the model error is equal to 0.1. Uncertainty is substantially reduced at assimilation times, x variable.

Tempering and Jittering for the Lorenz '63 Model



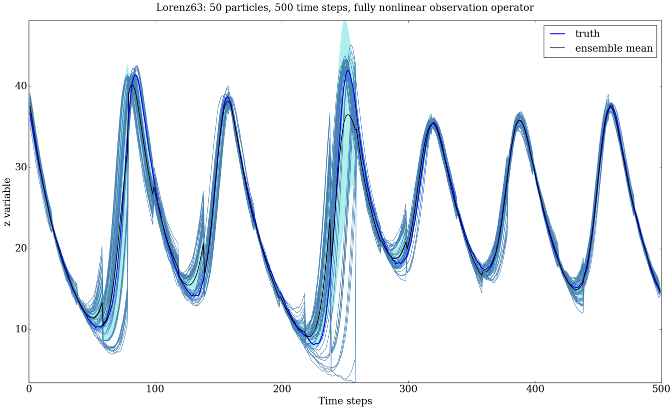
Evolution of the Lorenz '63 model for 500 time steps, all 3 variables are observed every 20 time steps. The initial uncertainty and the observational uncertainty are both equal to 1, the model error is equal to 0.1. Uncertainty is substantially reduced at assimilation times, RMSE and ES.

Tempering and Jittering for the Lorenz '63 Model



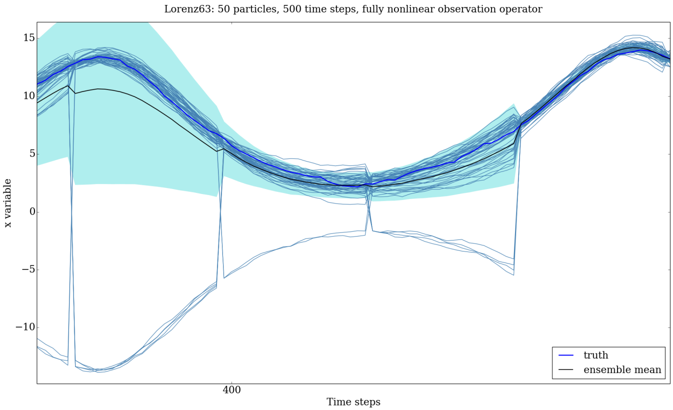
Evolution of the Lorenz '63 model for 500 time steps, all 3 variables are observed every 20 time steps. Observations are denoted by z_1, z_2, z_3. Fully nonlinear observation operator i.e. $z_1 = x^2, z_2 = y^2, z_3 = z^2$ plus noise, x variable.

Tempering and Jittering for the Lorenz '63 Model



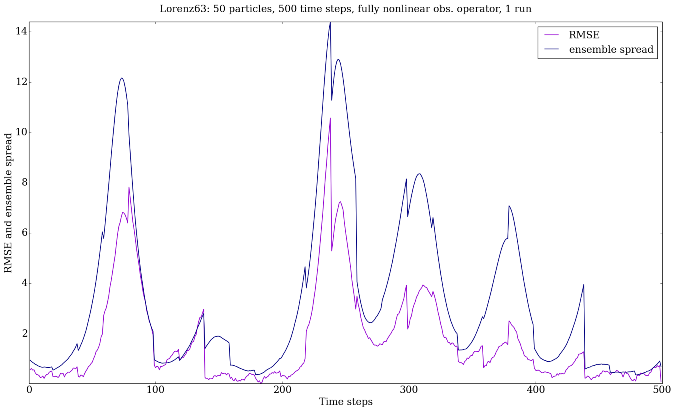
Evolution of the Lorenz '63 model for 500 time steps, all 3 variables are observed every 20 time steps. Observations are denoted by z_1, z_2, z_3. Fully nonlinear observation operator i.e. $z_1 = x^2, z_2 = y^2, z_3 = z^2$ plus noise, z variable.

Tempering and Jittering for the Lorenz '63 Model



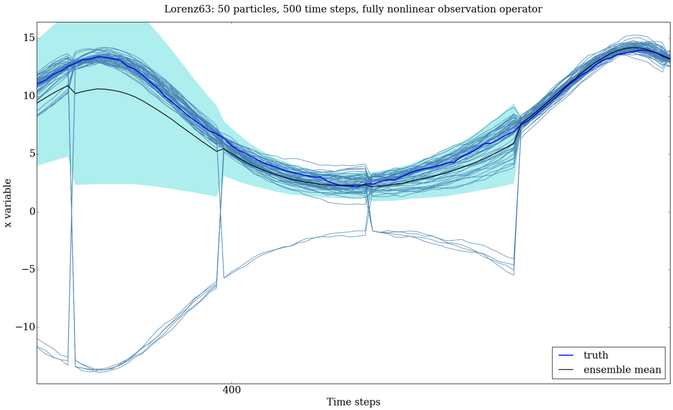
Evolution of the Lorenz '63 model for 500 time steps, all 3 variables are observed every 20 time steps. Observations are denoted by z_1, z_2, z_3. Fully nonlinear observation operator i.e. $z_1 = x^2, z_2 = y^2, z_3 = z^2$ plus noise, x variable.

Tempering and Jittering for the Lorenz '63 Model



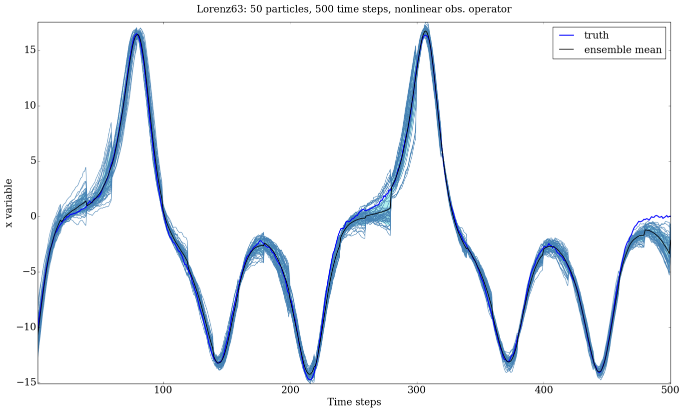
Evolution of the Lorenz '63 model for 500 time steps, all 3 variables are observed every 20 time steps. Observations are denoted by z_1, z_2, z_3. Fully nonlinear observation operator i.e. $z_1 = x^2, z_2 = y^2, z_3 = z^2$ plus noise, RMSE and ES.

Tempering and Jittering for the Lorenz '63 Model



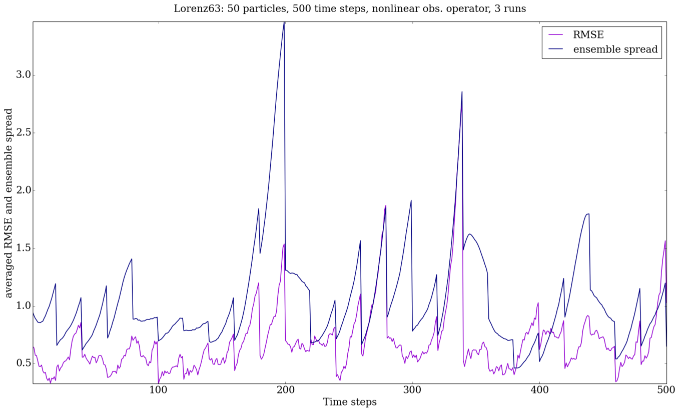
Evolution of the Lorenz '63 model for 500 time steps, all 3 variables are observed every 20 time steps. Observations are denoted by z_1, z_2, z_3. Fully nonlinear observation operator i.e. $z_1 = x^2, z_2 = y^2, z_3 = z^2$ plus noise, x variable.

Tempering and Jittering for the Lorenz '63 Model



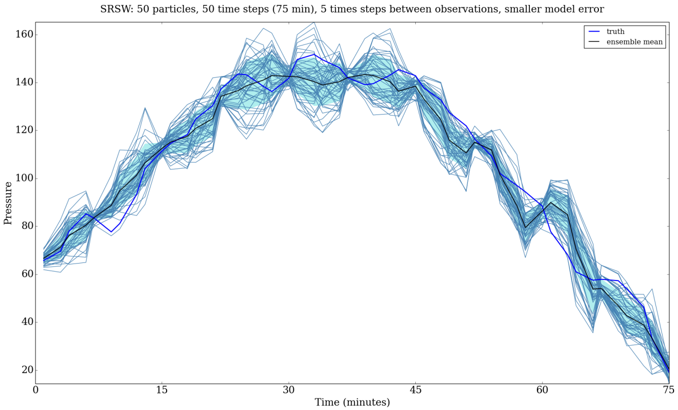
Evolution of the Lorenz '63 model for 500 time steps, all 3 variables are observed every 20 time steps. Observations are denoted by z_1, z_2, z_3. Partially nonlinear observation operator i.e. $z_1 = x^2, z_2 = y, z_3 = z$ plus noise.

Tempering and Jittering for the Lorenz '63 Model



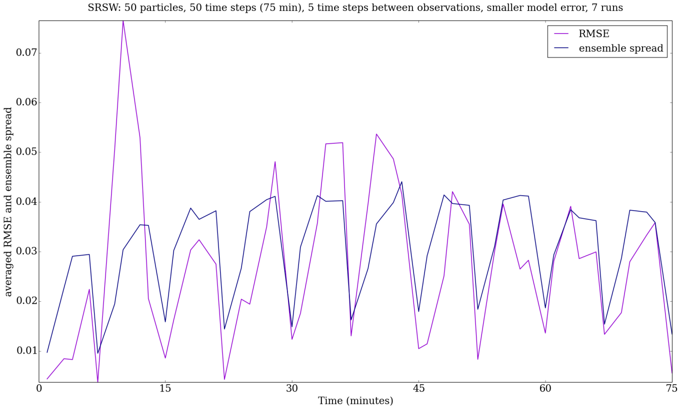
Evolution of the Lorenz '63 model for 500 time steps, all 3 variables are observed every 20 time steps. Observations are denoted by z_1, z_2, z_3. Partially nonlinear observation operator i.e. $z_1 = x^2, z_2 = y, z_3 = z$ plus noise. Displayed: averaged (3 runs) RMSE and ensemble spread.

Tempering and Jittering for the SRSW Model



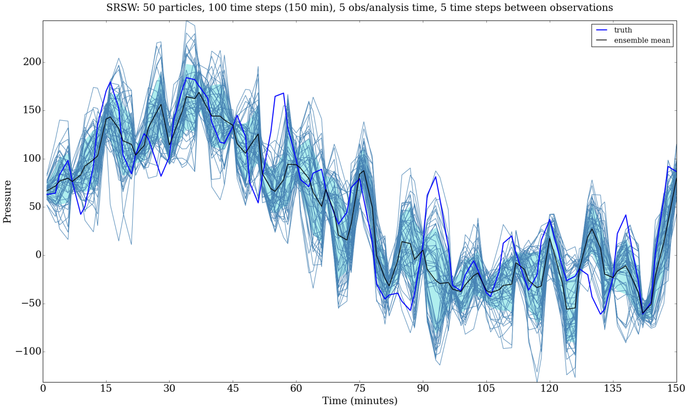
Evolution of the SRSW model for 50 time steps. The system is observed every 5 time steps. The particle filter efficacy is improved by the time window and the decreased model error.

Tempering and Jittering for the SRSW Model



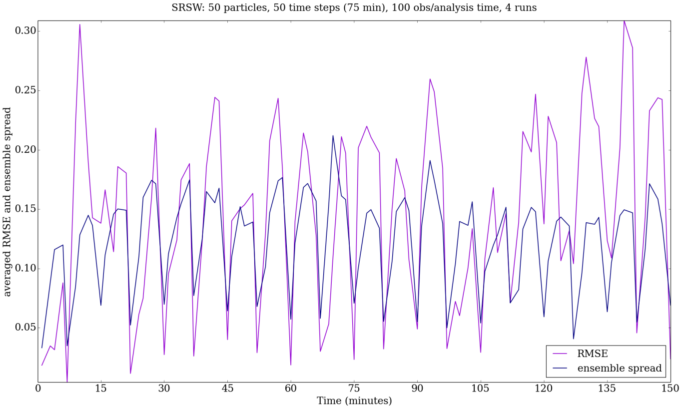
Evolution of the SRSW model for 50 time steps. The system is observed every 5 time steps. The particle filter efficacy is improved by the time window and the decreased model error, displayed: RMSE and ES.

Tempering and Jittering for the SRSW Model



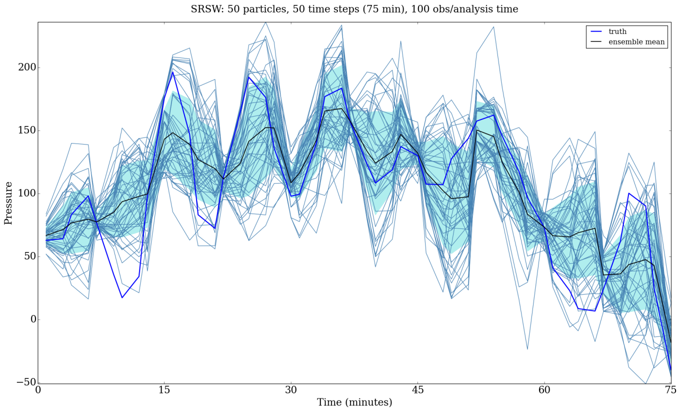
Evolution of the SRSW model for 100 time steps. The system is observed every 5 time steps, with 5 observations/analysis time. The cloud of particles follows the truth most of the time.

Tempering and Jittering for the SRSW Model



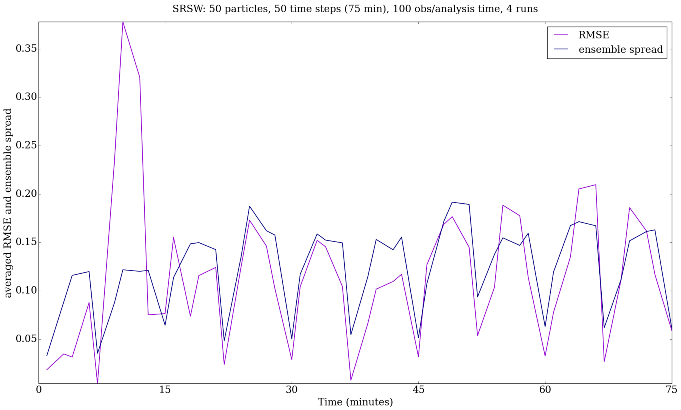
Evolution of the SRSW model for 100 time steps. The system is observed every 5 time steps, with 5 observations/analysis time. The cloud of particles follows the truth most of the time, displayed: RMSE and ES.

Tempering and Jittering for the SRSW Model



Evolution of the SRSW model for 50 time steps. The system is observed every 5 time steps, with 100 observations/analysis time. A large number of observations improves the performance of the particle filter. Displayed: pressure field.

Tempering and Jittering for the SRSW Model



Evolution of the SRSW model for 50 time steps. The system is observed every 5 time steps, with 100 observations/analysis time. A large number of observations improves the performance of the particle filter. Displayed: RMSE and ES.

4. Stochastic Calibration

Stochastic Calibration

Motivation

- chaotic variability of the multi-scale atmospheric and oceanic dynamics $\to$ certain small/sub-grid scale geophysical processes are still under-represented $\leftrightarrow$ introduce stochasticity $\Rightarrow$ improved representation of the missing physics.

Challenges

- general representations which capture a broad class of small-scale interactions
- the accurate resolution of small-scale features using numerical methods $\to$ a challenging task $\to$ reduce the model complexity using a coarser numerical resolution

Examples of approaches used to introduce stochasticity: Location Uncertainty (LU) (Mémin, 2014), Stochastic Advection by Lie Transport (SALT) (Holm, 2015).

$$\pi_{t-1} \xrightarrow[\substack{\textit{model}\\ \textit{forecast}\\ \textit{prediction}}]{K_t} K_t\pi_{t-1} =: p_t \xrightarrow[\substack{\textit{assimilation}\\ \textit{analysis}\\ \textit{update}}]{\textbf{non-linear}:\ g_t*} g_t * p_t = \pi_t$$

$$-\tau \xrightarrow[\textit{samples from the signal}]{\textit{model calibration}} 0 \xrightarrow[Y_t, t\in[0,T]]{\textit{filtering}} T$$

4.1. Stochastic Calibration - An Eulerian Approach

- The evolution of a model state m^f is represented using a deterministic partial differential equation

$$\frac{dm^f}{dt} = \mathcal{A}(m^f), \quad t \geq 0, \tag{6}$$

  where $\mathcal{A}$ is the model operator.
- We denote by m^c the coarse scale model.
- The effect of the *unresolved scales* can be mathematically modelled by a term of the form

$$\int_0^t \mathcal{M}(m_s^c) dW_s \tag{7}$$

  where $\mathcal{M}$ is a suitably chosen operator and $W_t = W(t, x)$ is a space-time Brownian motion.
- That is, the model run on the coarse scale will satisfy the stochastic (partial) differential equation

$$dm^c = \mathcal{A}(m^c)dt + \mathcal{M}(m^c)dW_t, \quad t \geq 0. \tag{8}$$

Eulerian Approach

- We decompose W as follows

$$W_t = \sum_{k=1}^{\infty} \xi_k W_t^k \tag{9}$$

where $(\xi_k)_k$ are space-dependent vector fields and $(W^k)_k$ are one-dimensional Brownian motions.

- So we want to model uncertainty using

$$\int_0^t \mathcal{M}(m_s^c) dW_s^N \tag{10}$$

on a sufficiently large time window $[0, T]$ where

$$W_t^N = \sum_{k=1}^{N} \xi_k W_t^k \tag{11}$$

and both ξ_k and N have to be estimated from data.

Eulerian Approach

- The data m^f can be coarsened using a low-pass filter. We denote by $C(m^f)$ the resulting coarsening of the data and

$$\hat{m} := m^f - C(m^f).$$

- We make the ansatz that the difference between the two processes $\hat{m} := m^f - C(m^f)$ has a stochastic representation given by (7).
- On a sufficiently small time interval $[t, t+\delta]$ we have

$$\hat{m}_{t+\delta} - \hat{m}_t = \int_t^{t+\delta} \mathcal{M}(C(m_s^f))dW_s^N \approx \sum_{k=1}^{N} \mathcal{M}(C(m_t^f))\xi_k \Delta W_t^k. \tag{12}$$

- Choose a partition of the time interval of the form $0 \leq t_1 < t_1 + \delta < \dots t_n < t_n + \delta \leq T$. Then we need to estimate the vector fields ξ_k from the data:

$$\hat{m}_{t_k+\delta} - \hat{m}_{t_k}, \quad k = 2, ..., n \tag{13}$$

  This will depend on the choice of perturbation chosen to account for the model uncertainty.

- Additive noise. In this case, $\mathcal{M}(C(m^f_s)) \equiv 1$, that is,

$$\hat{m}_{t+\delta} - \hat{m}_t \approx \sum_{k=1}^{N} \xi_k \Delta W_t^k. \tag{14}$$

and therefore $\hat{m}_{t_k+\delta} - \hat{m}_{t_k}$ can be interpreted as samples from a multi-dimensional Gaussian random variable M where

$$M \sim \mathcal{N}\left(0, \sum_{k=1}^{N} (\xi_k)(\xi_k)^T t\right). \tag{15}$$

Eulerian Approach

Some examples

- Multiplicative noise. In this case, $\mathcal{M}(C(m_t^f)) = f(C(m_t^f))$ with

$$\hat{m}_{t+\delta} - \hat{m}_t \approx f(C(m_t^f)) \sum_{k=1}^{N} \xi_k \Delta W_t^k. \tag{16}$$

in which $f(C(m_t^f))$ is a scalar function of $C(m_t^f)$, and $\frac{\hat{m}_{t_k+\delta} - \hat{m}_{t_k}}{f(C(m_t^f))}$ can be interpreted as samples from a multi-dimensional Gaussian random variable M with distribution (15).

Eulerian Approach

Some examples

- Transport noise.

$$\hat{m}_{t+\delta} - \hat{m}_t \approx \sum_{k=1}^{N} (\xi_k^1 \partial^1 C(m_s^f) + \xi_k^2 \partial^2 C(m_s^f)) \Delta W_t^k$$
$$= \sum_{k=1}^{N} (\partial^2 \psi_k \partial^1 C(m_s^f) - \partial^1 \psi_k \partial^2 C(m_s^f)) \Delta W_t^k$$
$$= \left(-\partial^2 C(m_s^f)\partial^1 + \partial^1 C(m_s^f)\partial^2\right) \left(\sum_{k=1}^{N} \psi_k \Delta W_t^k\right),$$

where we imposed the additional assumption that $\xi_k = \nabla^{\perp}\psi_k$. This ensures that $\nabla \cdot \xi_k = 0$. It follows that the solution of the linear hyperbolic equation

$$f = q \cdot \nabla \psi$$

with $f := -(\hat{m}_{t+\delta} - \hat{m}_t)$ and $q := -\nabla^{\perp} C(m_t^f)$ can be interpreted as samples from the multi-dimensional Gaussian random variable

$$M \sim \mathcal{N}\left(0, \sum_{k=1}^{N} (\psi_k)(\psi_k)^T t\right).$$

Application for the SRSW Model

- We use a simulation of the (deterministic) rotating shallow water model run on a fine grid of size 2224×320 (FD on a staggered Arakawa C-grid).
- Domain $\Omega = [0, L_x] \times [0, L_y]$ with $L_x = 27787.5$km and $L_y = 3975$ km.
- Periodic boundary condition in the East-West direction and slip-free boundary condition in the North-South direction.
- Starting condition

$$h = -a \arctan\left(0.05\left(\frac{y}{L_y} - 0.5\right)\pi\right) + \left[a \sin\left(16\pi\frac{x}{L_x}\right) + 0.5a \sin\left(2\pi\frac{x}{L_x}\right)\right] \sin\left(\pi\frac{y}{L_y}\right)^4 \quad a = 100.$$

- The starting velocity computed using geostrophic balance.
- Use the height of the fluid column $h^f = h$ as callibration data.

Application for the SRSW Model

Then calibrate using two coarser grids, one of size 556×80 and one of size 256×40. Then construct a mollified version of the fine grid trajectory using a low-pass filter.

1. Compute the time-increments of the discrepancy between the fine resolution and the coarse resolution trajectories.
2. Choose a *calibration time grid* in such a way that the data is decorrelated. For this, we first estimate the *decorrelation time* of the data corresponding to the fine grid trajectory.
3. Compute the sample noises corresponding to each of the times on the calibration time grid. In our case, this amounts to solving a sequence of hyperbolic equations.
4. Extract a basis for the stochastic noise together with the corresponding eigenvalues that explains a sufficiently large part of the variance in the data. For this we use the Principal Components Analysis algorithm.

See [2].

Application for the SRSW Model

For this SALT parametrization, $\hat{h} := h^f - C(h^f)$

$$\begin{aligned}
\hat{h}_{t+\delta} - \hat{h}_t &\approx \sum_{k=1}^{N} (\xi_k^1 \partial^1 C(h_s^f) + \xi_k^2 \partial^2 C(h_s^f))\Delta W_t^k. \\
&= \sum_{k=1}^{N} (\partial^2 \psi_k \partial^1 C(h_s^f) - \partial^1 \psi_k \partial^2 C(h_s^f))\Delta W_t^k \\
&= \left(-\partial^2 C(h_s^f)\partial^1 + \partial^1 C(h_s^f)\partial^2\right)\Psi \\
\Psi &:= \sum_{k=1}^{N} \psi_k \Delta W_t^k
\end{aligned}$$

where we assume that $\xi_k \equiv \nabla^\perp \psi_k$. This will ensure that $\nabla \cdot \xi_k = 0$.

Application for the SRSW Model

Hence Ψ is the solution of the linear hyperbolic equation

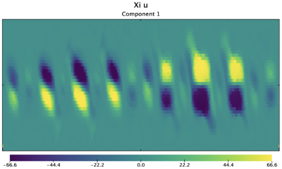
$$f = q \cdot \nabla \psi$$

with $f := -(\hat{h}_{t+\delta} - \hat{h}_t)$ and $q := -\nabla^{\perp} C(h_t^f)$ and

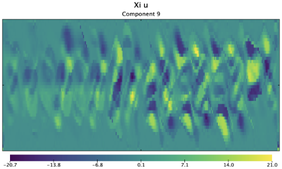
$$\frac{\Psi}{\sqrt{\delta}} \sim \mathcal{N}\left(0, \sum_{k=1}^{N} \psi_k \psi_k^T\right).$$

The vector fields ψ_k are recovered by PCA and then $\xi_k \equiv \nabla^{\perp} \psi_k$.

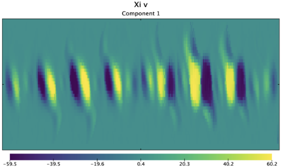
Application for the SRSW Model



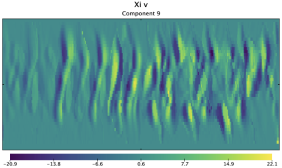
(a) ξ_1^u.



(b) ξ_9^u.



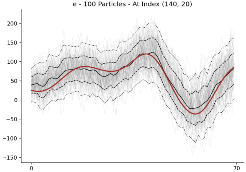
(c) ξ_1^v.



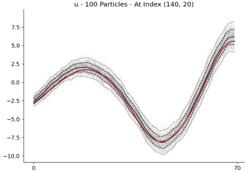
(d) ξ_9^v.

Figure: Estimated ξ's on the $c = 8$ grid. The left column depicts the first component and the right column the last component of the set explaining 99% of the variance in the data obtained from the solutions of the calibration equation.

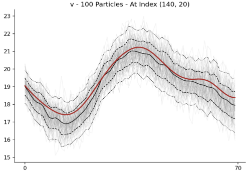
Application for the SRSW Model



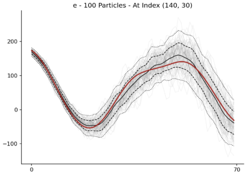
(a) Elevation η at $(140, 20)$.



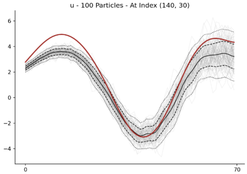
(b) Zonal Vel. u at $(140, 20)$.



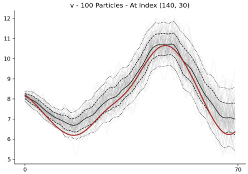
(c) Meridional Vel. v at $(140, 20)$.



(d) Elevation η at $(140, 30)$.



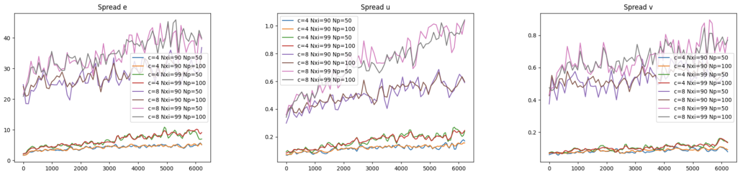
(e) Zonal Vel. u at $(140, 30)$.



(f) Meridional Vel. v at $(140, 30)$.

Figure: coarse grid with $c = 8$. time horizon is 560 fine PDE timesteps.

Application for the SRSW Model



(a) Elevation η. (b) Zonal Vel. u. (c) Meridional Vel. v.

Figure: Ensemble spread in all scenarios on the central grid point with index $(140, 20)$ in the case of $c = 8$ and index $(280, 40)$ in the $c = 4$ case.

4.2. Stochastic Calibration Using Generative Models

The world is not always Gaussian!

The assumption that the data increments are normally distributed is not always realistic. We can have

$$\hat{m}_{t_{n+1}} - \hat{m}_{t_n} = -\mathcal{M}(C(m^f_{t_n}))\nabla^\perp \tilde{N}_n \tag{17}$$

where $\tilde{N}_n$ is not necessarily normally distributed, that is, for the same fine-grid discretised version

$$m^f_{t_{n+1}} = m^f_{t_n} + \mathcal{A}\left(m^f_{t_n}\right)\Delta$$

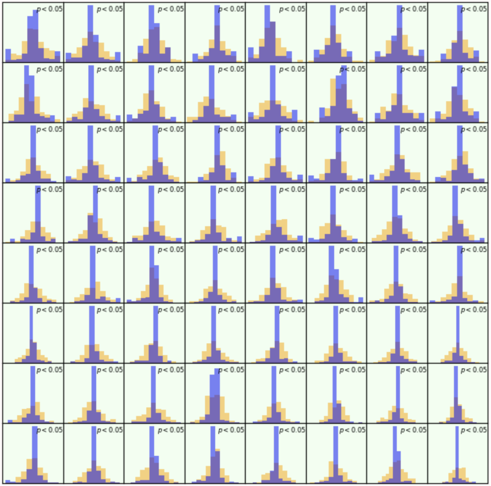
instead of

$$m^c_{t_{n+1}} = m^c_{t_n} + \mathcal{A}\left(m^c_{t_n}\right)\Delta + \sum_{k=1}^{M} \mathcal{M}(m^c_{t_n})\xi_k W^k_{t_n}\sqrt{\Delta}$$

we have

$$m^c_{t_{n+1}} = m^c_{t_n} + \mathcal{A}\left(m^c_{t_n}\right)\Delta + \mathcal{M}(m^c_{t_n})N_n$$

where $(N_n)_{n\geq 1}$ are independent identically distributed random variables, but not necessarily with a Gaussian distribution.

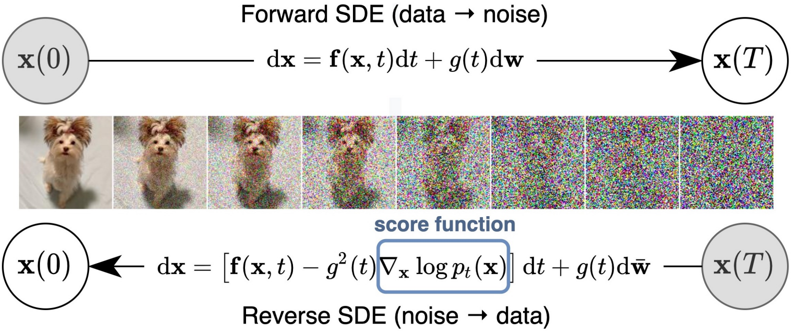


Distribution of the generated noise values compared to Gaussians at every pixel in a central region. A Kolmogorov-Smirnov one-sample test has been performed to check if the data come from a normal distribution. The hypothesis was rejected for all locations indicating that the generated noise does not come from a simple normal distribution.

Generative Modelling

- The previous methodology uses a PCA technique based on the ansatz that the data is normally distributed $\Rightarrow$ replace the PCA technique by a generative model technique, to model closer to the data and relax the Gaussian assumption.
- *Generative models* are a class of machine learning models designed to generate new data samples from an unknown distribution which to which we have access only through a dataset of samples.
- An important class of generative models are *diffusion models*.
- Key idea: iteratively transform the training data through a diffusion-like mechanism into samples from a known distribution (e.g. Gaussian). In the process, the forward and the backward diffusions are learnt using a neural network. Once the learning is complete, samples from the unknown distribution are obtained by running the backward diffusion initiated from samples from the Gaussian distribution.
- More precisely: we assume that N_n has an unknown distribution which will be modeled by a *score-based generative model* $\rightarrow$ Diffusion Schrödinger Bridge ([6]).

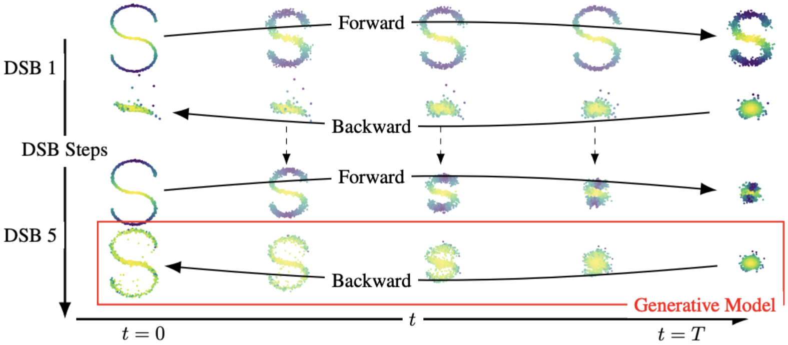
Diffusion Schrödinger Bridge



Score-based diffusion modelling

- Valentin De Bortoli, James Thornton, Jeremy Heng, and Arnaud Doucet, *Diffusion Schrödinger bridge with applications to score-based generative modeling*, Advances in Neural Information Processing Systems (2021).

Diffusion Schrödinger Bridge



DSB allows to iterate the training of the forward and backward diffusions.

- Valentin De Bortoli, James Thornton, Jeremy Heng, and Arnaud Doucet, *Diffusion Schrödinger bridge with applications to score-based generative modeling*, Advances in Neural Information Processing Systems (2021).

Diffusion Schrödinger Bridge

- Assume we have the following forward diffusion process (noising)

$$\mathrm{d}\mathbf{X}_t^n = f_t^n\left(\mathbf{X}_t^n\right)\mathrm{d}t + \sqrt{2}\,\mathrm{d}\mathbf{B}_t, \quad \mathbf{X}_0^n \sim p_{\text{data}}\ ,$$

- Then the corresponding backward dynamics is associated with the following (denoising) process

$$\mathrm{d}\mathbf{Y}_t^n = b_{T-t}^n\left(\mathbf{Y}_t^n\right)\mathrm{d}t + \sqrt{2}\,\mathrm{d}\mathbf{B}_t, \quad \mathbf{Y}_0^n \sim p_{\text{prior}}$$

  where

$$b_t^n(x) = -f_t^n(x) + 2\nabla \log p_t^n(x),$$

  and p_t^n is a density function.

- This is done iteratively, for the discretised equations.

Score-based generative models

- For a given distribution $p_{data}(x)$, the score function is given by

$$f(x) = \nabla_x \log p_{data}(x).$$

- In practice, the score function can be learnt using a neural network $s_\theta(x)$.
- Aim: minimise the distance between the score function and the *score network*

$$argmin_\theta \frac{1}{2}\mathbb{E}_{p_{data}(x)}\left[\|s_\theta(x) - \nabla_x \log p_{data}(x)\|^2\right]$$

$\rightarrow$ intractable as we don't know $p_{data}(x)$, but it has been shown ([7]) that this is equivalent to e.g.

$$argmin_\theta \frac{1}{2}\mathbb{E}_{p_{data}(x)}\left(tr(\nabla_x)s_\theta(x) + \frac{1}{2}\|s_\theta(x)\|_2^2\right)$$

which can be computed using Monte Carlo methods by sampling from $p_{data}(x)$ since the last quantity only depends on knowing $s_\theta(x)$.

Score-based generative models

- The standard approach requires many small steps to approximate the target (true) distribution.
- Schrödinger Bridge $\rightarrow$ a more efficient path between distributions i.e. a refinement of existing score-based methods by allowing to significantly reduce the number of stepsizes needed $\Rightarrow$ a faster and more accurate generative modelling.
  - We define a sequence of distributions π^n that alternate between:
    - Keeping p_{prior} fixed while adjusting towards p_{data}.
    - Keeping p_{data} fixed while adjusting towards p_{prior}.
  - Over many iterations, π^n converges to the optimal Schrödinger Bridge solution $\pi^\star$.

Diffusion Schrödinger Bridge

- Let $(\mathbf{X}_t)_{t\in[0,T]}$ be the noising SDE, with distribution $\mathbb{P}$. [2]
- We want to find $\pi^\star$ such that $\pi^\star_0 = p_{\text{data}}$ and $\pi^\star_T = p_{\text{prior}}$ and minimize the Kullback-Leibler divergence between $\pi^\star$ and $\mathbb{P}$:

$$\pi^\star = \text{argmin}_\pi \left\{\text{KL}(\pi \mid \mathbb{P}), \pi_0 = p_{\text{data}}, \pi_N = p_{\text{prior}}\right\}$$

  where N is the number of (discrete) time steps used in the approximation of the forward SDE ($t_N = T$) i.e. we want to find a new diffusion process π that transports the data distribution p_{data} to the prior p_{prior}, while staying as close as possible to the reference diffusion $\mathbb{P}$.
- Successively solve half-bridge problems: Define a sequence of distributions $(\pi^n)_{n\in\mathbb{N}}$ such that

$$\pi^{2n+1} = \text{argmin}\left\{\text{KL}\left(\pi \mid \pi^{2n}\right), \pi_N = p_{\text{prior}}\right\}$$
$$\pi^{2n+2} = \text{argmin}\left\{\text{KL}\left(\pi \mid \pi^{2n+1}\right), \pi_0 = p_{\text{data}}\right\}$$

  with initial condition $\pi^0 = \mathbb{P}$ the reference dynamics.
- For large n, π^n is close to the Schrödinger bridge $\pi^\star$. A generative model can be obtained by sampling from π^{2n+1}.

[2] [6]

Diffusion Schrödinger Bridge

- Assume that π^{2n} is the measure associated with the diffusion

$$\mathrm{d}\mathbf{X}_t^n = f_t^n\left(\mathbf{X}_t^n\right)\mathrm{d}t + \sqrt{2}\,\mathrm{d}\mathbf{B}_t, \quad \mathbf{X}_0^n \sim p_{\text{data}}\ ,$$

- Then in [6] it is shown that $\left(\pi^{2n+1}\right)^R$ is associated with the diffusion

$$\mathrm{d}\mathbf{Y}_t^n = b_{T-t}^n\left(\mathbf{Y}_t^n\right)\mathrm{d}t + \sqrt{2}\,\mathrm{d}\mathbf{B}_t, \quad \mathbf{Y}_0^n \sim p_{\text{prior}}$$

  where R denotes the time-reversal operation and

$$b_t^n(x) = -f_t^n(x) + 2\nabla\log p_t^n(x),$$

  with p_t^n the density of π_t^{2n}.
- If this procedure is repeated, one can obtain ([6]) that π^{2n+2} is associated with the diffusion

$$\mathrm{d}\mathbf{X}_t^{n+1} = f_t^{n+1}\left(\mathbf{X}_t^{n+1}\right)\mathrm{d}t + \sqrt{2}\,\mathrm{d}\mathbf{B}_t, \quad \mathbf{X}_0^{n+1} \sim p_{\text{data}}\ ,$$

  where

$$f_t^{n+1}(x) = -b_t^n(x) + 2\nabla\log q_t^n(x),$$

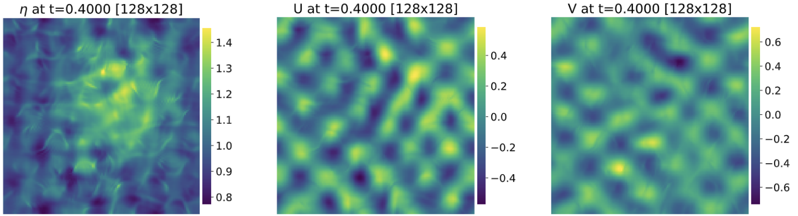
  with q_t^n the density of π_t^{2n+1}.

Application to the SRSW Model

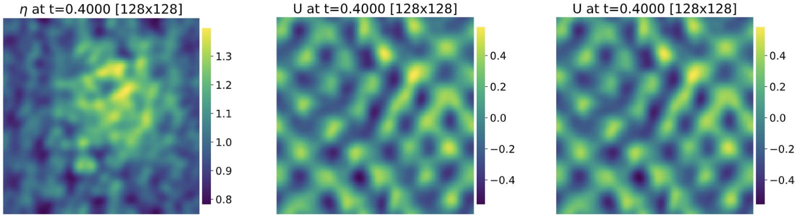
Calibration for the SRSW model

We use a version of the previous SRSW model, which contains also a wind forcing and a bottom friction term.

- Data for the calibration of the SPDE noise was generated from a simulation of the deterministic model on a fine grid of size 128×128 grid points over a total duration of $15,000$ timesteps.
- The data is collected by applying a low pass filter corresponding to a coarse grid resolution of 32×32 gridpoints.
- We isolate the high-frequency fluctuations by taking the difference between the original and filtered fields.
- The computed fluctuations of the height variable η are then time-differenced to yield increments and used in the calibration equation.
- A solution of the calibration equation is approximated using a finite-difference method constrained to yield a mean-zero solution.
- The approximated solutions represent a stream function for the noise we seek to generate. Iterating this process over time generates a sample from the distribution of the assumed stream function.



(a) Height η Fine. **(b)** Zonal Velocity U Fine. **(c)** Meridional Velocity V Fine.



(d) Height η Coarse. **(e)** Zonal Velocity U Coarse. **(f)** Meridional Velocity V Coarse.

The top row (a-c) shows the fully resolved fields and the bottom row (d-f) shows the coarsened fields after applying the low-pass filter. The solution is obtained after running 4000 timesteps.

Application for the SRSW Model

For the SRSW model, $\hat{\eta} := \eta^f - C(\eta^f)$

$$\begin{aligned}
\hat{\eta}_{t+\delta} - \hat{\eta}_t &\approx \sum_{k=1}^{N} (\xi_k^1 \partial^1 C(\eta_s^f) + \xi_k^2 \partial^2 C(\eta_s^f)) \Delta W_t^k. \\
&= \sum_{k=1}^{N} (\partial^2 \psi_k \partial^1 C(\eta_s^f) - \partial^1 \psi_k \partial^2 C(\eta_s^f)) \Delta W_t^k \\
&= \left(-\partial^2 C(\eta_s^f) \partial^1 + \partial^1 C(\eta_s^f) \partial^2\right) \Psi \\
\Psi &:= \sum_{k=1}^{N} \psi_k \Delta W_t^k
\end{aligned}$$

where we assume that $\xi_k \equiv \nabla^\perp \psi_k$. This will ensure that $\nabla \cdot \xi_k = 0$.

Application for the SRSW Model

Hence Ψ is the solution of the linear hyperbolic equation

$$f = q \cdot \nabla \psi$$

with $f := -(\hat{\eta}_{t+\delta} - \hat{\eta}_t)$ and $q := \nabla^\perp C(\eta^f)$ and

$$\frac{\Psi}{\sqrt{\delta}} \sim \mathcal{N}\left(0, \sum_{k=1}^{N} \psi_k \psi_k^T\right).$$

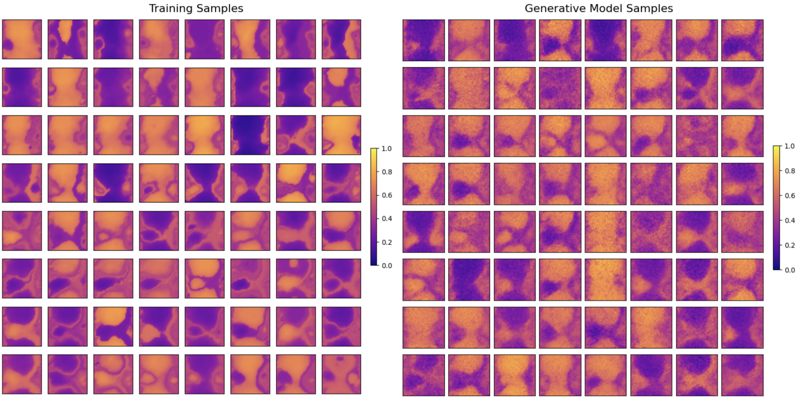
The vector fields ψ_k are recovered by PCA and then $\xi_k \equiv \nabla^\perp \psi_k$.

This assumption is not always realistic. We can have

$$\hat{m}_{t_{n+1}} - \hat{m}_{t_n} = -C(m^f_{t_n}) \nabla^\perp \tilde{N}_n$$

where $\tilde{N}_n$ is not necessarily normally distributed, that is

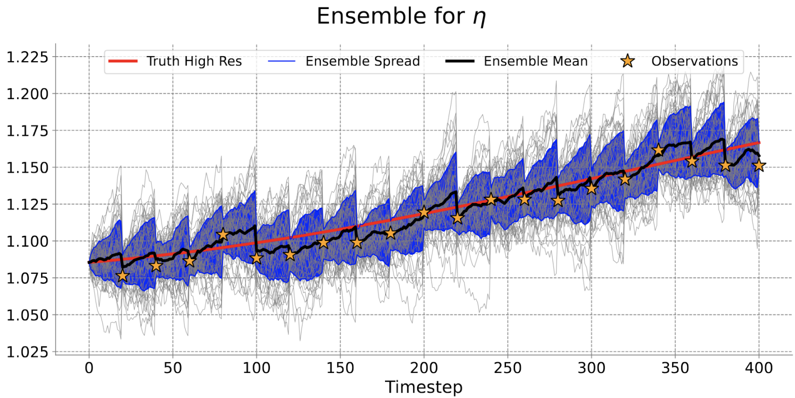
$$\frac{\Psi}{\sqrt{\delta}} \not\sim \mathcal{N}\left(0, \sum_{k=1}^{N} \psi_k \psi_k^T\right).$$



(a) Training Samples (b) Generated Samples

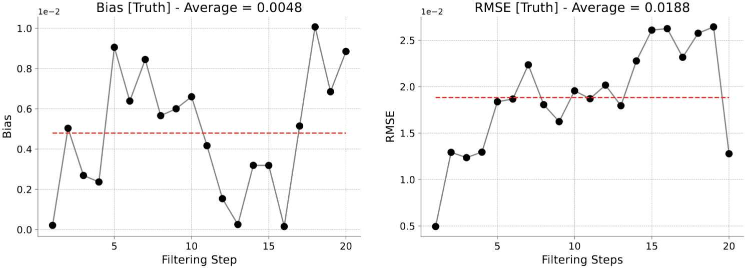
Training Samples and Samples from the generative model. Samples of the training data after transformation. The fields are outputs of the calibration equation thought of as stream functions for the velocity perturbations in the SPDE. Samples from the generative model.

Data Assimilation Results



Results of the filtering experiment over 400 total timesteps. The assimilated variable is η using an ensemble of 50 particles and an assimilation window of 20 forecast timesteps.

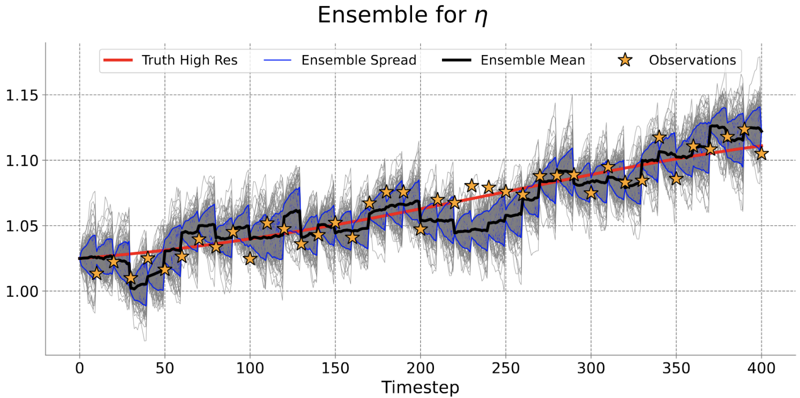
Data Assimilation Results



(b) Ensemble Bias **(c)** Ensemble RMSE

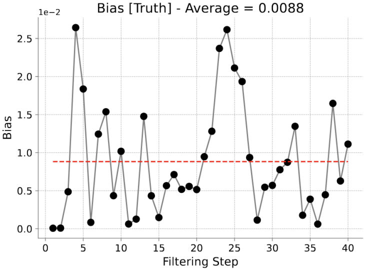
Figure 9: Filtering Results 1D. Results of the filtering experiment over 400 total timesteps. The assimilated variable is η using an ensemble of 50 particles and an assimilation window of 20 forecast timesteps. We show, at the observed gird location, (a) the ensemble evolution compared to the true deterministic fine-grid trajectory, (b) the ensemble bias over time, and (c) the ensemble RMSE.

Data Assimilation Results

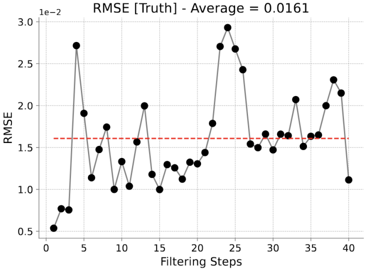


Results of the filtering experiment over 400 total timesteps. The assimilated variable is η using an ensemble of 100 particles and an assimilation window of 10 forecast timesteps.

Data Assimilation Results



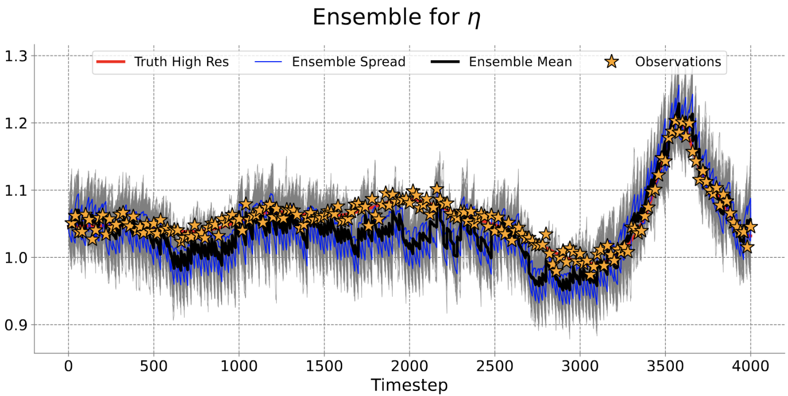
(b) Ensemble Bias



(c) Ensemble RMSE

Figure 11: Filtering Results 9D. Results of the filtering experiment over 400 total timesteps. The assimilated variable is η using an ensemble of 100 particles and an assimilation window of 10 forecast timesteps. We show, at the best observed gird location, (a) the ensemble evolution compared to the true deterministic fine-grid trajectory, (b) the ensemble bias over time, and (c) the ensemble RMSE.

Data Assimilation Results



Results of the filtering experiment over 4000 total timesteps. The assimilated variable is η using an ensemble of 200 particles and an assimilation window of 20 forecast timesteps.

Final remarks and future work

Data Assimilation

- The adaptive tempering based particle filter with jittering can be applied to nonlinear data assimilation problems where the the signal is given by a stochastic rotating shallow model.
- The single resampling step in the standard particle filter is replaced by a repeated application of a triple procedure: tempering, resampling and jittering, the number of applications being chosen adaptively by taking into account the effective sample size of the intermediate sample.
- The same particle filter is applicable also when the signal is given by L63 with nonlinear observations.
- We test the forecast reliability by comparing the forecast root mean square error (RMSE) with the forecast ensemble spread (ES).

Final remarks and future work

- The performance of the data assimilation methodology is influenced by: the linearity of the observation operator, DA time window, number of observations per analysis time.

Calibration

- We introduce a calibration method which can be applied to a large class of stochastic parametrisations and it is also agnostic as to the source of data (real or synthetic).

Future work: Use the calibrated model together with the particle filter procedure to solve a data assimilation problem where the truth is a model of a horizontal slice of the atmosphere and the data is given by a set of atmospheric pressure observations collected by DWD from pressure sensors carried by commercial aircrafts.

References

O. Lang, D. Crisan, P. J. van Leeuwen, R. Potthast, *Bayesian Inference for Fluid Dynamics: A Case Study for the Stochastic Rotating Shallow Water Model*, Frontiers in Applied Mathematics and Statistics, 8 (2022).

D. Crisan, O. Lang, A. Lobbe, P.J. van Leeuwen, R. Potthast, Noise calibration for the stochastic rotating shallow water model, Foundations of Data Science (2023).

D. Crisan, A. Lobbe, O. Lang, *Bayesian inference for geophysical fluid dynamics using generative models*, Philos Trans A of the Royal Society, Math Phys Eng Sci (2025).

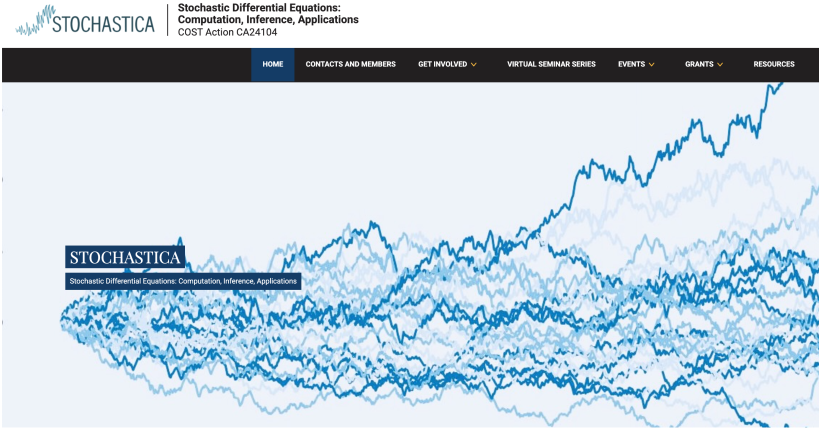
D. Crisan, A. Bain, *Fundamentals of Stochastic Filtering*.

Alexandros Beskos, Dan Crisan, Ajay Jasra, *On the stability of sequential Monte Carlo methods in high dimensions*, Ann. Appl. Probab. 24(4): 1396-1445 (August 2014). DOI: 10.1214/13-AAP951.

Valentin De Bortoli, James Thornton, Jeremy Heng, and Arnaud Doucet, Diffusion Schrödinger bridge with applications to score-based generative modeling, Advances in Neural Information Processing Systems (2021).

Hyvärinen, A. and Dayan, P. (2005), Estimation of non-normalized statistical models by score matching, Journal of Machine Learning Research, 6(4).

Vincent, P., A connection between score matching and denoising autoencoders, Neural Computation, 23(7):1661–1674.



www.ucc.ie/en/stochastica/

About STOCHASTICA

- European COST Action CA24104 (2025–2029) on **Stochastic Differential and Stochastic Partial Differential Equations**.
- Builds a pan-European network connecting **theory, computation, and applications**.
- Involves mathematicians, computational scientists, statisticians, and industry partners.
- Applications span health & medicine, renewable energy, finance, data science, and AI.

STOCHASTICA

Goals and Participation

- Advance the **theoretical analysis** of SDEs/SPDEs, including well-posedness, stability, and multiscale limits.
- Develop and benchmark **high-order and structure-preserving numerical schemes**.
- Establish rigorous frameworks for **statistical inference and parameter estimation** in stochastic systems.
- Create a shared **computational toolbox** and **repository of benchmark problems** for reproducible research.
- Promote synergy between theory, numerics, and real-world applications, and strengthen links between academia and industry.

Organisation

- Five Working Groups: **Theory, Numerics, Inference, Toolbox, Dissemination**.
- Open to **researchers at all career stages**, including PhD students and postdocs.

Thank you!

Diffusion Schrödinger Bridge

- The noising process is modelled by a forward Markov Chain $(X_k)_{k=0}^N$ on $\mathbb{R}^d$ such that

$$X_{k+1} = X_k + \gamma_{k+1} f(X_k) + 2\gamma_{k+1} V_{k+1}, \quad k = 0, \ldots, N,$$

where $(V_{k+1})_k \sim \mathcal{N}(0, \mathbf{1})$ are i.i.d. Gaussian random variables, $f : \mathbb{R}^d \to \mathbb{R}^d$ is a drift function and $(\gamma_k)_k$ are typically small stepsize parameters.

- The joint density p of the Markov Chain $X_{0:N} = (X_0, \ldots, X_N)$ can be decomposed into the corresponding forward transition densities, denoted by $(p_{k+1|k})_k$, as follows

$$p\left(x_{0:N}\right) = p_0(x_0) \prod_{k=0}^{N-1} p_{k+1|k}(x_{k+1}|x_k)$$

where

$$x_{0:N} = (x_k)_{k=0}^N \in \left(\mathbb{R}^d\right)^{N+1}$$

and we assume the initial density $X_0 \sim p_0 = p_{data}$.

Diffusion Schrödinger Bridge

Similarly we can write down the backward decomposition

$$p(x_{0:N}) = p_N(x_N) \prod_{k=0}^{N-1} p_{k|k+1}(x_k|x_{k+1}) = p_N(x_N) \prod_{k=0}^{N-1} \frac{p_k(x_k)p_{k+1|k}(x_{k+1}|x_k)}{p_{k+1}(x_{k+1})},$$

where $(p_k)_k$ are the marginal densities and $(p_{k|k+1})_k$ are the reverse transition densities. The last equality above is a result of the application of Bayes' theorem to the reverse transition densities $(p_{k|k+1})_k$.

- The methodology is based on sampling from p_{data} using the reverse decomposition initialized at $p_N = p_{\text{prior}}$.
- For this, we need to approximate the reverse transition densities.
- Assumption: the backward transitions are normally distributed as

$$p_{k+1|k}(x_{k+1}|x_k) = \mathcal{N}(x_{k+1}; x_k + \gamma_{k+1} f(x_k), 2\gamma_{k+1}\mathbf{1}).$$

Diffusion Schrödinger Bridge

- We apply a Taylor approximation ([6]) to get

$$
\begin{aligned}
&p_{k|k+1}\left(x_k \mid x_{k+1}\right)=p_{k+1|k}\left(x_{k+1} \mid x_k\right) \exp \left[\log p_k\left(x_k\right)-\log p_{k+1}\left(x_{k+1}\right)\right] \\
&\approx \mathcal{N}\left(x_k ; x_{k+1}-\gamma_{k+1} f\left(x_{k+1}\right)+2 \gamma_{k+1} \nabla \log p_{k+1}\left(x_{k+1}\right), 2 \gamma_{k+1} \mathbf{1}\right)
\end{aligned}
$$

- The backward transitions are then Gaussian, with a drift depending on the parameters f and $(\gamma_k)_k$ and on the *score functions* $(\nabla \log p_k)_k$.
- We can integrate out the initial density from the marginals such that

$$
p_{k+1}(x_{k+1}) = \int p_0(x_0) p_{k+1|0}(x_{k+1}|x_0) dx_0
$$

and thus

$$
\nabla \log p_{k+1}(x_{k+1}) = \mathbb{E}_{p_{0|k+1}}[\nabla_{x_{k+1}} \log p_{k+1|0}(x_{k+1}|X_0)].
$$

Diffusion Schrödinger Bridge

- The conditional expectation above is intractable, but the joint distribution is available through samples, so we can use regression to find it

$$s_{k+1} = \arg\min_s \mathbb{E}_{p_{0,k+1}} \left[\left\| s\left(X_{k+1}\right) - \nabla_{x_{k+1}} \log p_{k+1|0}\left(X_{k+1} \mid X_0\right) \right\|^2 \right],$$

where $\|\cdot\|$ denotes the L_2-norm on $\mathbb{R}^d$.

- We can learn a parametrised approximation of the score (all scores simultaneously)

$$s_{\theta^\star}(k, x_k) \approx \nabla \log p_k(x_k)$$

via Denoising Score Matching (Vincent 2011) as

$$\theta^\star = \arg\min_\theta \sum_{k=1}^{N} \mathbb{E}_{p_{0,k}}[\|s_\theta(k, X_k) - \nabla_{x_k} \log p_{k|0}(X_k|X_0)\|^2].$$

- That is, we estimate the score function and then sample $X_0 \overset{approx}{\sim} p_{data}$ using the diffusion started at $p_N \approx p_{prior}$ such that

$$X_k = X_{k+1} - \gamma_{k+1} f(X_{k+1}) + 2\gamma_{k+1} s_{\theta^\star}(k+1, X_{k+1}) + \sqrt{2\gamma_{k+1}}\mathcal{N}(0, \mathbf{1}).$$

Diffusion Schrödinger Bridge

- Let $\mathcal{P}_{N+1}$ be the space of sequences of probability densities of length $N+1$. In the Schrödinger Bridge framework, we consider the joint density $p \in \mathcal{P}_{N+1}$ of the Markov Chain X and we want to find a density $\pi^\star \in \mathcal{P}_{N+1}$ such that

$$\pi^\star = \{\mathrm{KL}(\pi \mid p) : \pi \in \mathcal{P}_{N+1}, \pi_0 = p_{\text{data}}, \pi_N = p_{\text{prior}}\} \tag{18}$$

where for any two probability densities p and q over a space $\mathcal{X}$,

$$KL(p||q) := \int_{\mathcal{X}} p(x) \log\left(\frac{p(x)}{q(x)}\right) dx.$$

- Assuming $\pi^\star$ is available, the generative model is obtained by sampling $X_N \sim p_{\text{prior}}$, followed by the reverse-time dynamics $X_k \sim \pi^\star_{k|k+1}(\cdot \mid X_{k+1})$ for $k \in \{N-1, \ldots, 0\}$.
- To find a minimum of (18), use IPF: initialise at $\pi^0 = p(x_{0:N})$ and define the following iterative process

$$\pi^{2n+1} = \left\{\mathrm{KL}\left(\pi \mid \pi^{2n}\right) : \pi \in \mathcal{P}_{N+1}, \pi_N = p_{\text{prior}}\right\}$$
$$\pi^{2n+2} = \left\{\mathrm{KL}\left(\pi \mid \pi^{2n+1}\right) : \pi \in \mathcal{P}_{N+1}, \pi_0 = p_{\text{data}}\right\}.$$

Diffusion Schrödinger Bridge

Proposition (Proposition 2 in [6])

Assume that $\mathrm{KL}\left(p_{\text{data}} \otimes p_{\text{prior}} \mid p_{0,N}\right) < +\infty$. Then for any $n \in \mathbb{N}, \pi^{2n}$ and π^{2n+1} admit positive densities w.r.t. the Lebesgue measure denoted as p^n resp. q^n and for any $x_{0:N} \in \mathcal{X}$, we have $p^0\left(x_{0:N}\right) = p\left(x_{0:N}\right)$ and

$$
\begin{aligned}
q^n\left(x_{0:N}\right) = p_{\text{prior}}\left(x_N\right) \prod_{k=0}^{N-1} p_{k \mid k+1}^n\left(x_k \mid x_{k+1}\right), p^{n+1}\left(x_{0:N}\right) \\
= p_{\text{data}}\left(x_0\right) \prod_{k=0}^{N-1} q_{k+1 \mid k}^n\left(x_{k+1} \mid x_k\right).
\end{aligned}
$$

- In practice we have access to $p_{k+1 \mid k}^n$ and $q_{k \mid k+1}^n$. Hence, to compute $p_{k \mid k+1}^n$ and $q_{k+1 \mid k}^n$ we use

$$
\begin{aligned}
p_{k \mid k+1}^n\left(x_k \mid x_{k+1}\right) &= \frac{p_{k+1 \mid k}^n\left(x_{k+1} \mid x_k\right) p_k^n\left(x_k\right)}{p_{k+1}^n\left(x_{k+1}\right)} \\
q_{k+1 \mid k}^n\left(x_{k+1} \mid x_k\right) &= \frac{q_{k \mid k+1}^n\left(x_k \mid x_{k+1}\right) q_{k+1}^n\left(x_{k+1}\right)}{q_k^n\left(x_k\right)}.
\end{aligned}
$$

Stochastic Calibration - Application for the SRSW Model

- Solve the sequence of hyperbolic equations

$$\Delta_{t^n_{\text{calib}}} = \eta^c_x(t^n_{\text{calib}})\frac{\partial \psi_i}{\partial y} - \eta^c_y(t^n_{\text{calib}})\frac{\partial \psi_i}{\partial x}$$

  with the same boundary condition as the PDE.
- Rewrite it as

$$f = q \cdot \nabla \psi$$

  with $f := \Delta_{t^n_{\text{calib}}}$ and $q := -\nabla^{\perp}\eta^c(t^n_{\text{calib}})$.
- Use the Finite-Element Method to solve the PDE.

Stochastic Calibration - Application for the SRSW Model

- We use the h-field on $\mathcal{T}_{\text{calib}} \cup \mathcal{T}_{\text{PDE}}$ where $\mathcal{T}_{\text{calib}} = \mathcal{T}_{\text{PDE}} + \delta t_{\text{calib}}$, $\delta t_{\text{calib}} = 4$.
- Each h_{t_i} and $h_{t_i + \delta t_{\text{calib}}}$ is *coarsened* according to the low-pass filter given by local averages, through a convolution kernel (adapted to the coarsening)
- Compute the time-increments of the discrepancy between the high resolution fields and coarsened fields.
- Calibration time-grid t^n_{calib} is obtained as a sub-grid of the PDE time grid as described below:

$$\Delta_{t^n_{\text{calib}}} = h^c_{t^n_{\text{calib}} + \delta t_{\text{calib}}} - h^f_{t^n_{\text{calib}} + \delta t_{\text{calib}}} - (h^c_{t^n_{\text{calib}}} - h^f_{t^n_{\text{calib}}}).$$

Stochastic Calibration - Application for the SRSW Model

- Solve the sequence of hyperbolic equations

$$\Delta_{t^n_{\text{calib}}} = h^c_x(t^n_{\text{calib}})\frac{\partial \psi_i}{\partial y} - h^c_y(t^n_{\text{calib}})\frac{\partial \psi_i}{\partial x}$$

  with the same boundary condition as the PDE.
- Rewrite it as

$$f = q \cdot \nabla \psi$$

  with $f := \Delta_{t^n_{\text{calib}}}$ and $q := -\nabla^{\perp} h^c(t^n_{\text{calib}})$.
- Use the Finite-Element Method to solve the PDE.

Stochastic Calibration - Application for the SRSW Model

- The solutions to the sequence of calibration equations can be thought of as *stream functions* for the perturbation fields $\tilde{u}$, $\tilde{v}$, so that $(\tilde{u}, \tilde{v})^\top = \nabla^\perp \psi$. This grid based data $(\tilde{u}_{ij}, \tilde{v}_{ij})$ is vectorized as $\Psi = ((\tilde{u}_{ij}), (\tilde{v}_{ij}))^\top$ and represented in the following form

$$\frac{\Psi_i - \bar{\Psi}}{\sqrt{\delta}} = \sum_{j=1}^{N} \zeta^j \Delta W_i^j, \quad \Delta W_i^j \quad \sim N(0,1)$$

- N is decided by using empirical orthogonal functions (EOFs).
- The EOFs are given by a PCA algorithm based on the singular value decomposition (SVD).