Particle Filters and Flows

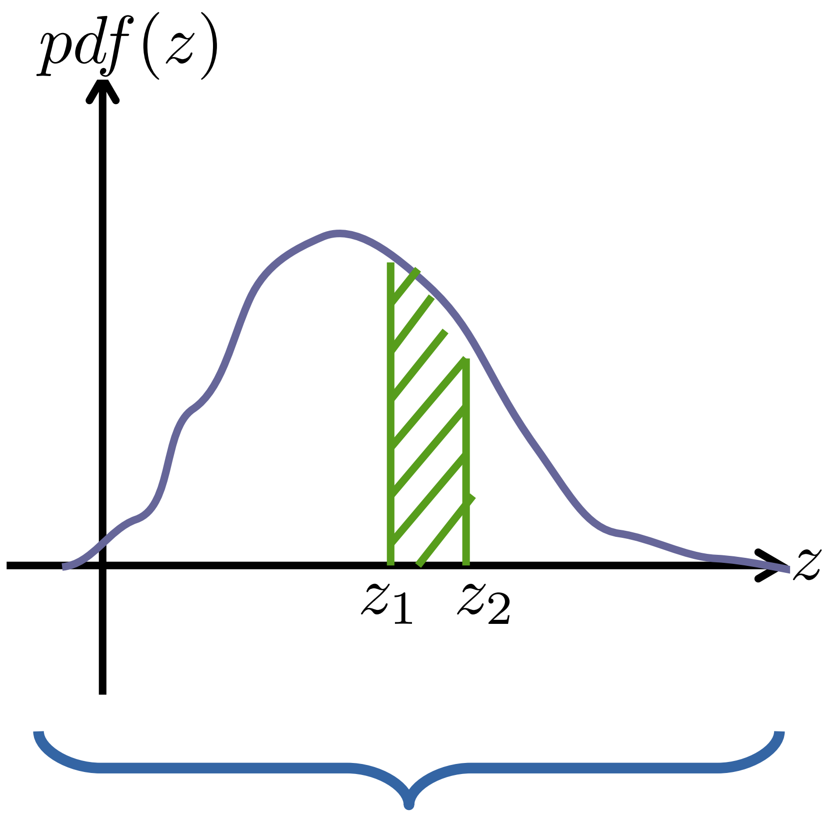
Peter Jan van Leeuwen
(using many slides from Javier Amezcua

NCEO/ECMWF training course

Data Assimilation Research Centre

University of Reading

Probability density functions



Statistical **support** of the variable

Probability density function:

$$pdf(z) \geq 0 \;\; z \in \mathcal{R}^1$$

$$pdf(\mathbf{z}) \geq 0 \;\; \mathbf{z} \in \mathcal{R}^N$$

Probability:

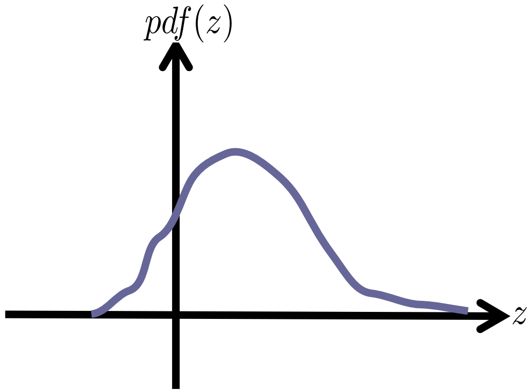
$$p\left(z_1 \leq z \leq z_2\right) = \int_{z_1}^{z_2} pdf(z)dz$$

$$0 \leq p\left(z_i \leq z \leq z_j\right) \leq 1$$

Cumulative density function

$$cdf(z_1) = \int_{-\infty}^{z_1} pdf(z)dz$$

Properties and operations



Normalization:

$$\int_{-\infty}^{+\infty} pdf(z)dz = 1$$

Expected value: **center of mass of the distribution (barycentre/centroid)**

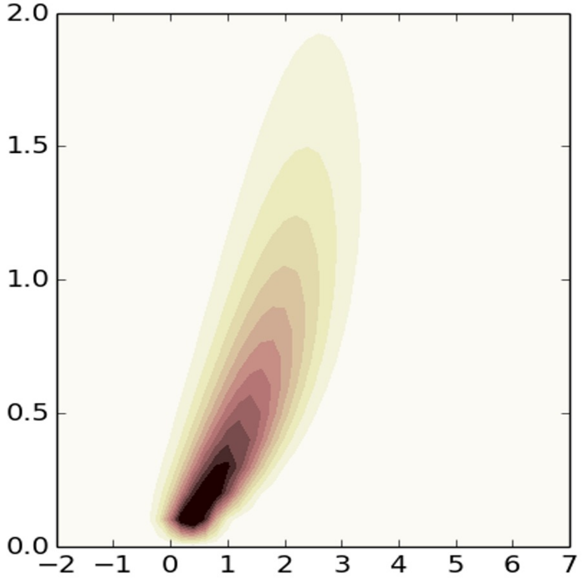
$$\mu_z = E[z] = \int_{-\infty}^{+\infty} z\,pdf(z)dz \qquad \mu_g(z) = E[g(z)] = \int_{-\infty}^{+\infty} g(z)pdf(z)dz$$

Variance: **mean quadratic deviation from the expected value**

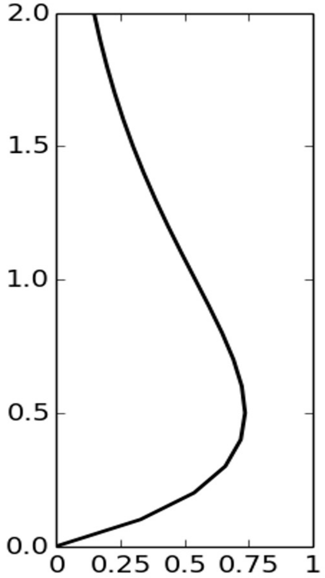
$$\sigma_z^2 = Var[z] = \int_{-\infty}^{+\infty} (z-\mu_z)^2 pdf(z)dz$$

Joint, conditionals, marginals

joint $pdf(x, y)$

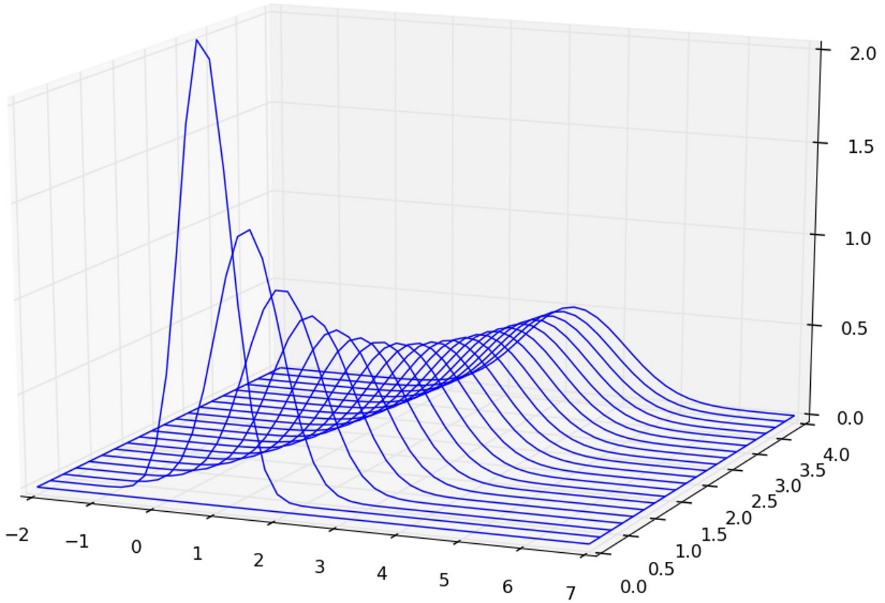


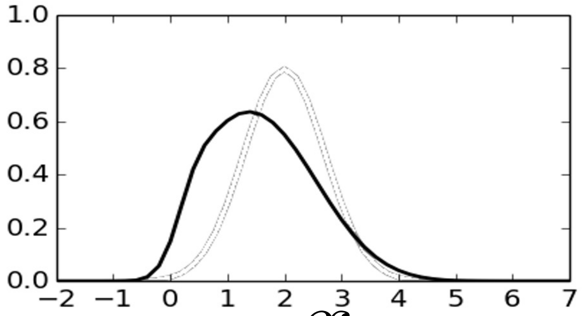
marginal $pdf(y)$



y

conditional $pdf(y|x)$





x

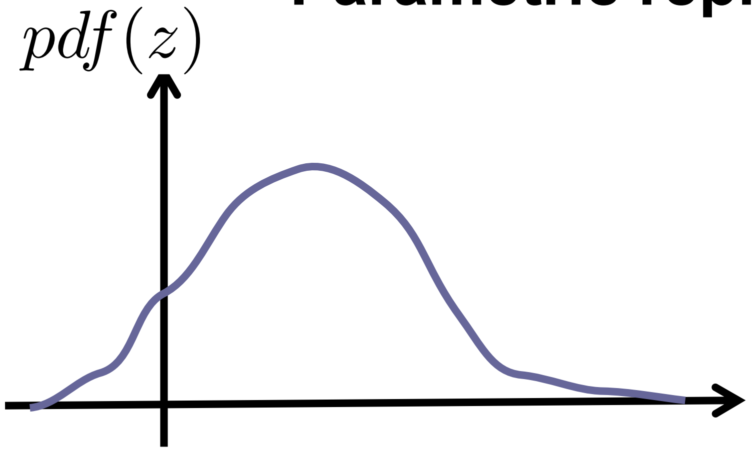
marginal $pdf(x)$

$$pdf(y|x) = \frac{pdf(x, y)}{pdf(x)}$$

$$pdf(y) = \int pdf(x, y) dx$$

Representing pdf's

Parametric representation

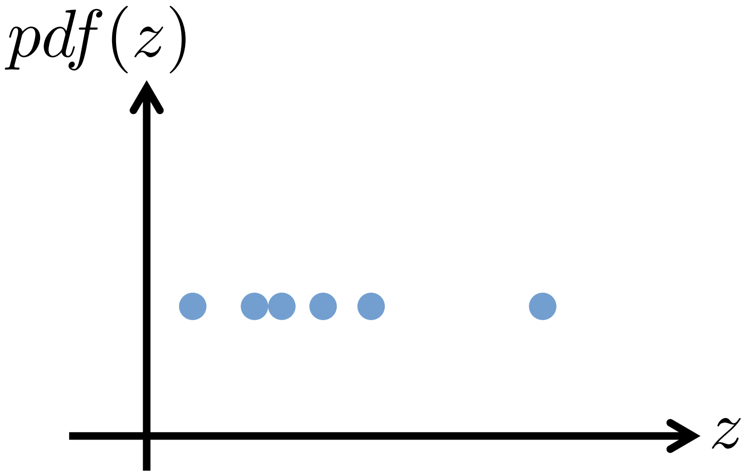


$$pdf(z;\theta)$$

Eg:

$N(\mu,\sigma)$ $Exp(\theta)$ $\Gamma(\alpha,\beta)$

Empirical or weak representation (sample)



$$pdf(z) = \frac{1}{N_e}\sum_{n_e=1}^{Ne}\delta(z - z_{n_e})$$

The Dirac delta function

$$pdf(z) = \frac{1}{N_e} \sum_{n_e=1}^{Ne} \delta(z - z_{n_e})$$

Properties:

$$\delta(z - z^*) = 0 \;\forall z \neq z^*$$

$$\int_{-\infty}^{+\infty} \delta(z - z^*) dz = 1$$

The Dirac delta 'kills' integrals:

$$\int_{-\infty}^{+\infty} g(z)\delta(z - z^*) dz = g(z^*)$$

Bayes theorem

Likelihood of the state to give these observations

Prior distribution of the state variable .

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

Marginal distribution of the **observations**.

Posterior probability distribution of the state variables **given** the observations.

How to get these elements?

Nonlinear data assimilation

3DVar and 4Dvar and (Ensemble) Kalman Filters and Smoothers assume Gaussian pdfs, so we only need the mean and covariance. For nonlinear data assimilation we need the whole pdf.

Assume the model equation reads

$$dx = f(x)dt + d\beta \qquad \text{with} \qquad d\beta \sim N(0, Qdt)$$

then the corresponding evolution equation for the pdf of *x* reads:

$$\frac{\partial p(x,t)}{\partial t} = \nabla_x f(x)p(x,t) + \nabla_x^2 Q p(x,t)$$

Nonlinear data assimilation

Forecast between observations:

Continuous system:

- With model error (Wiener process): **Fokker-Plank equation**.
- Without model error: **Liouville equation**

Discrete system:

- With model error. Transition probabilities. **Chapman-Kolmogorov equation.**
- Without model error. **Chapman-Kolmogorov equation using Dirac deltas**.

Analysis at observation times:

- Bayes Theorem

But how to do this in high-dimensional systems???

The simplest particle filter

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

Use ensemble

$$p(x) = \sum_{i=1}^{N} \frac{1}{N}\delta(x - x_i)$$

$$p(x|y) = \sum_{i=1}^{N} w_i \delta(x - x_i)$$

with

$$w_i = \frac{p(y|x_i)}{\sum_j p(y|x_j)}$$

the weights.

What are these weights?

The weight w_i is the normalised value of the likelihood of the state x_i to give these observations.

For Gaussian distributed variables it is given by:

$$\begin{aligned} w_i &\propto p(y|x_i) \\ &\propto \exp\left[-\frac{1}{2}\left(y - H(x_i)\right) R^{-1} \left(y - H(x_i)\right)\right] \end{aligned}$$

One can just calculate this value

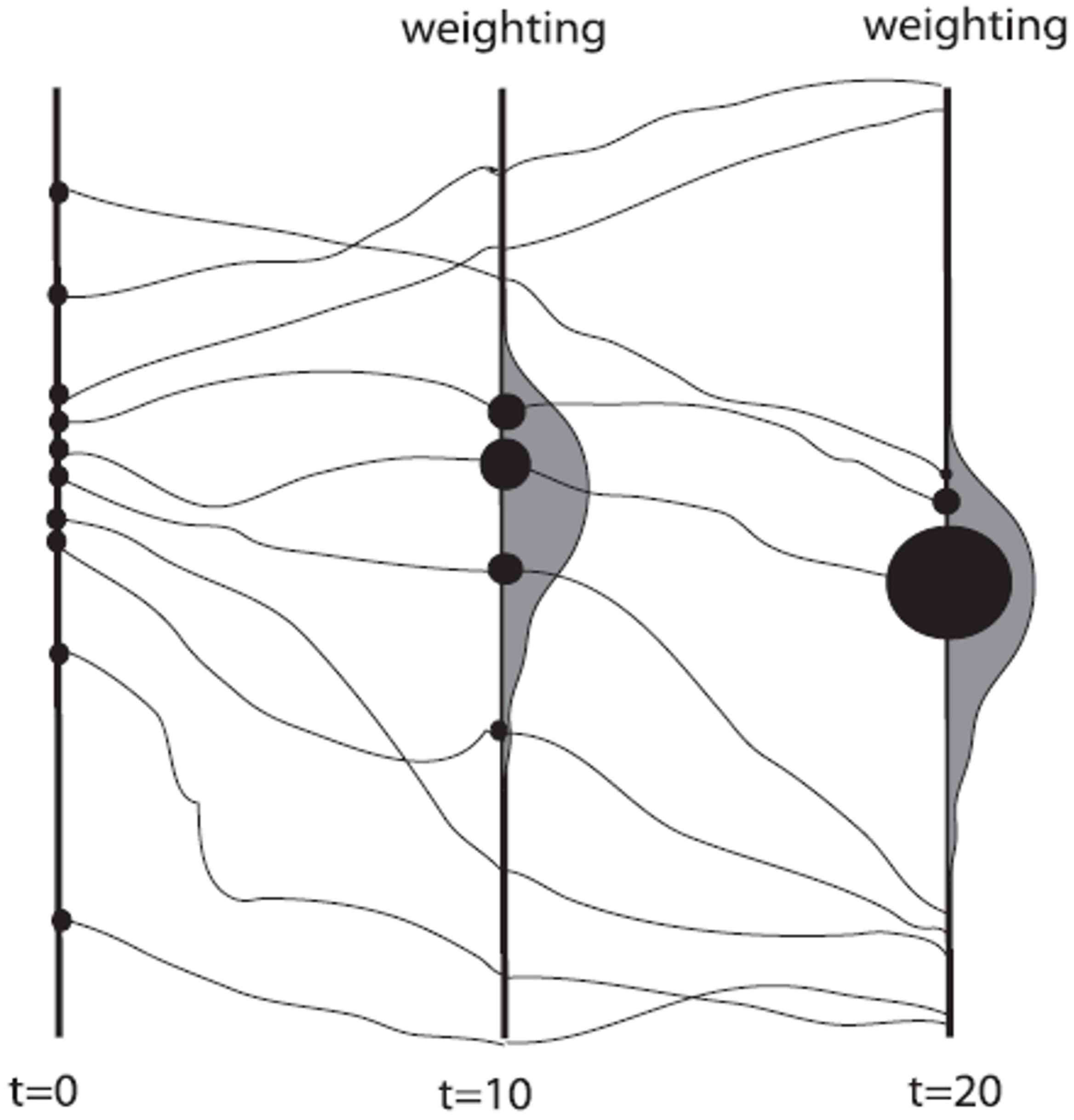
That is all !!!

Or is it? More needed for high-dimensional problems…

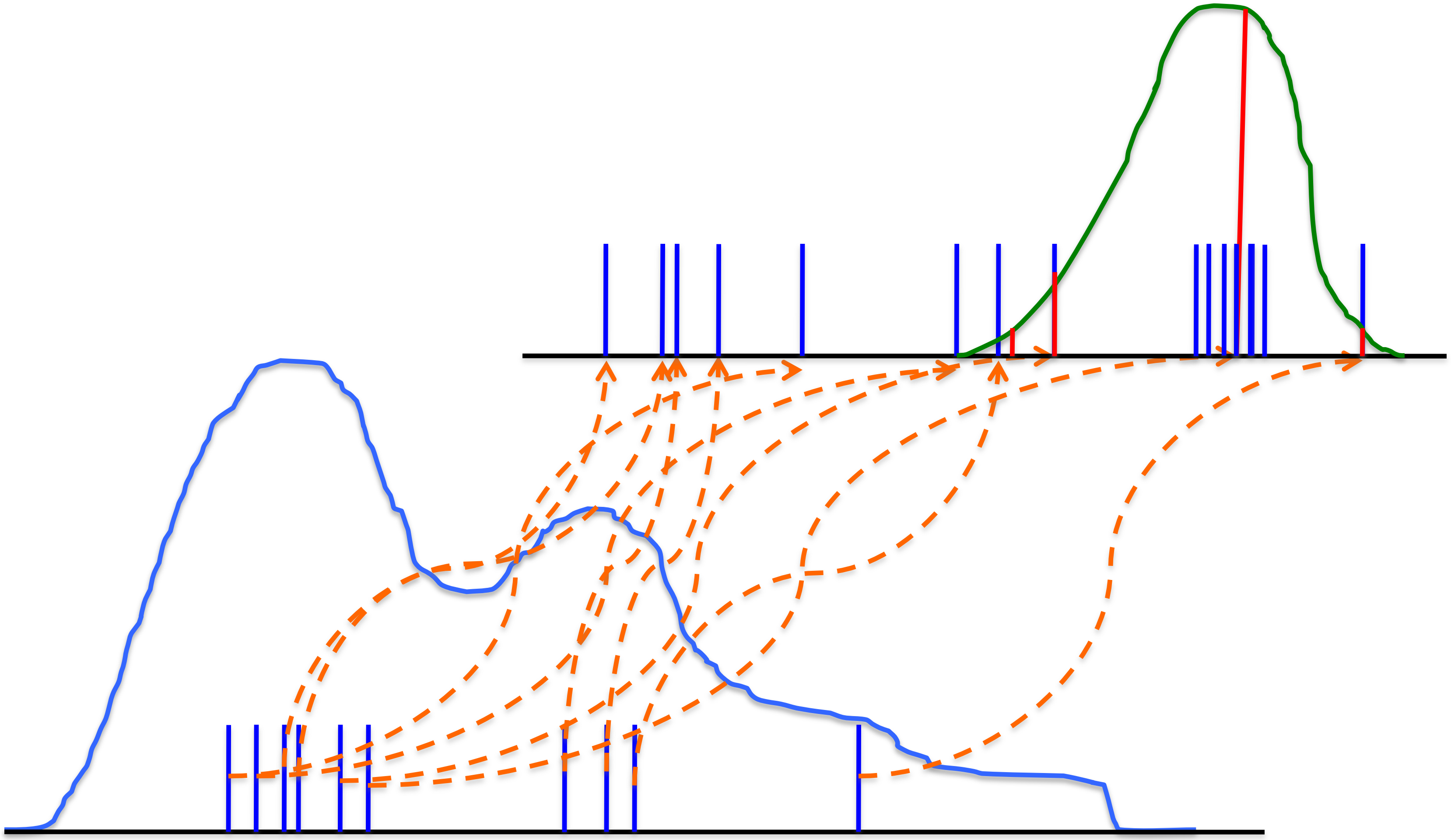
Particle filtering in time

If we iterate this in time we need to weight every time there is an observation.

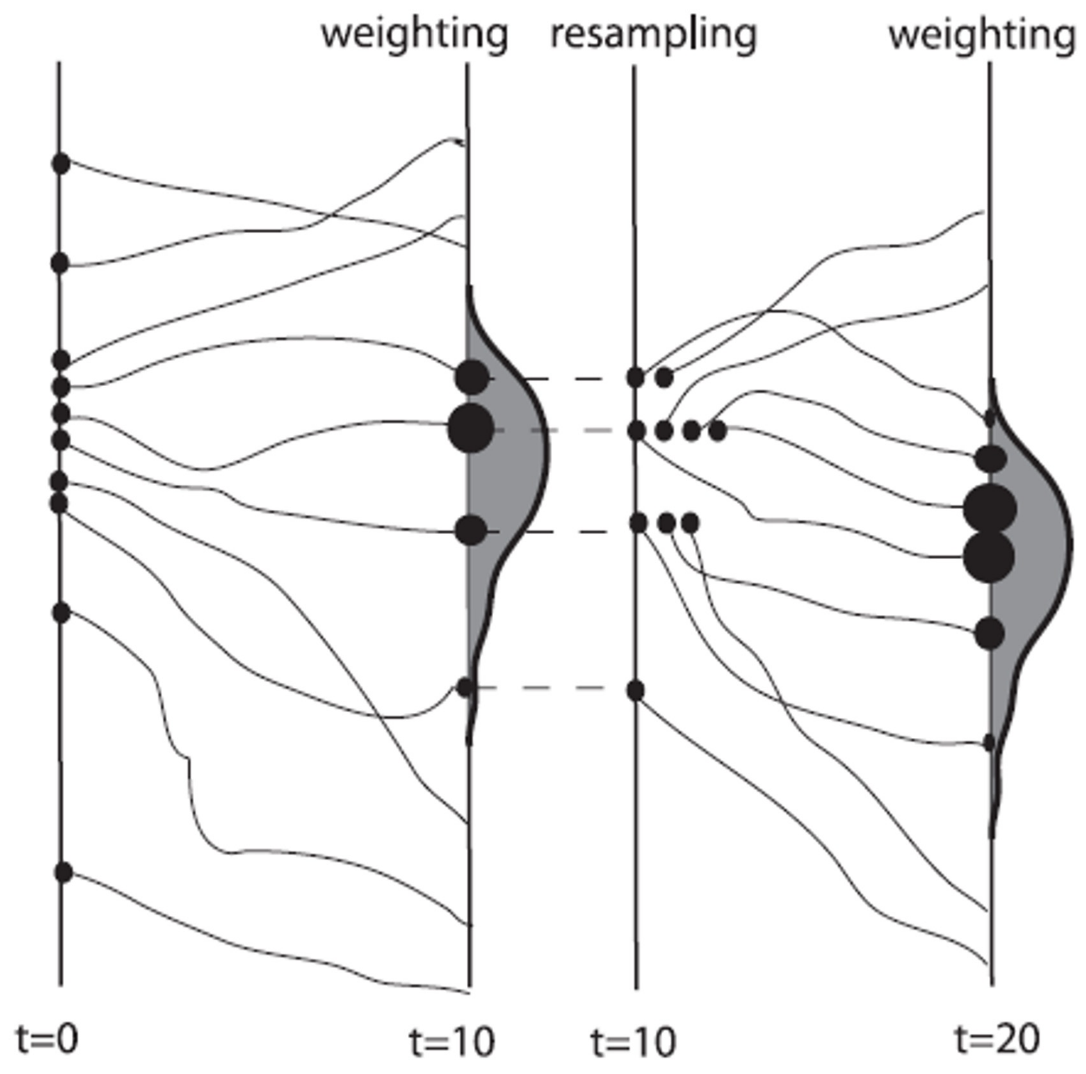
Note that after some time one particle gets all the weight…



Weight degeneracy and resampling



Simple Importance Resampling



Particle filters in high-dimensional systems are degenerate even with resampling:

$$N_e \propto exp[D_{eff}^2]$$

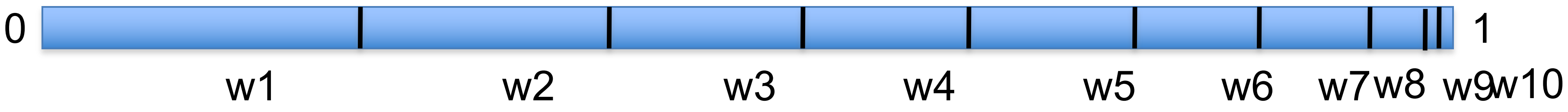
Snyder et al 2008

$$D_{eff} \propto N_{y,indep}$$

Ades and Van Leeuwen 2013

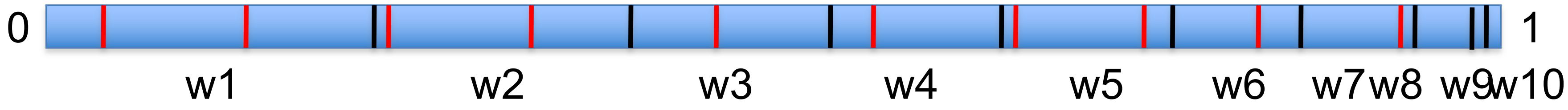
A simple resampling scheme

1. Put all weights after each other on the unit interval:



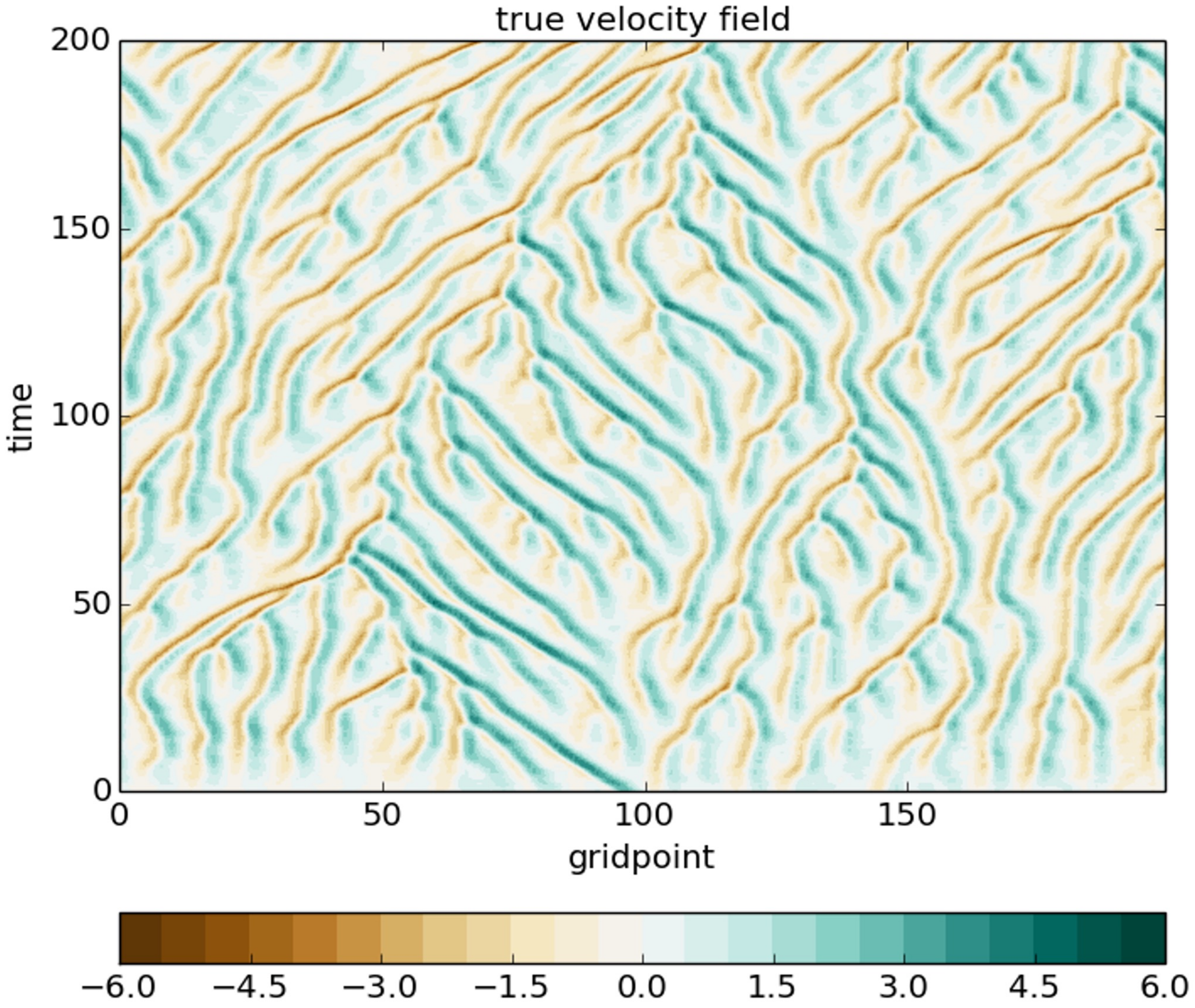
2. Draw a random number from the uniform distribution over [0,1/N], in this case with 10 members over [0,1/10].
3. Put that number on the unit interval: its end point is the first member drawn.

4. Add 1/N to the end point: the new end point is our second member. Repeat this until N new members are obtained.

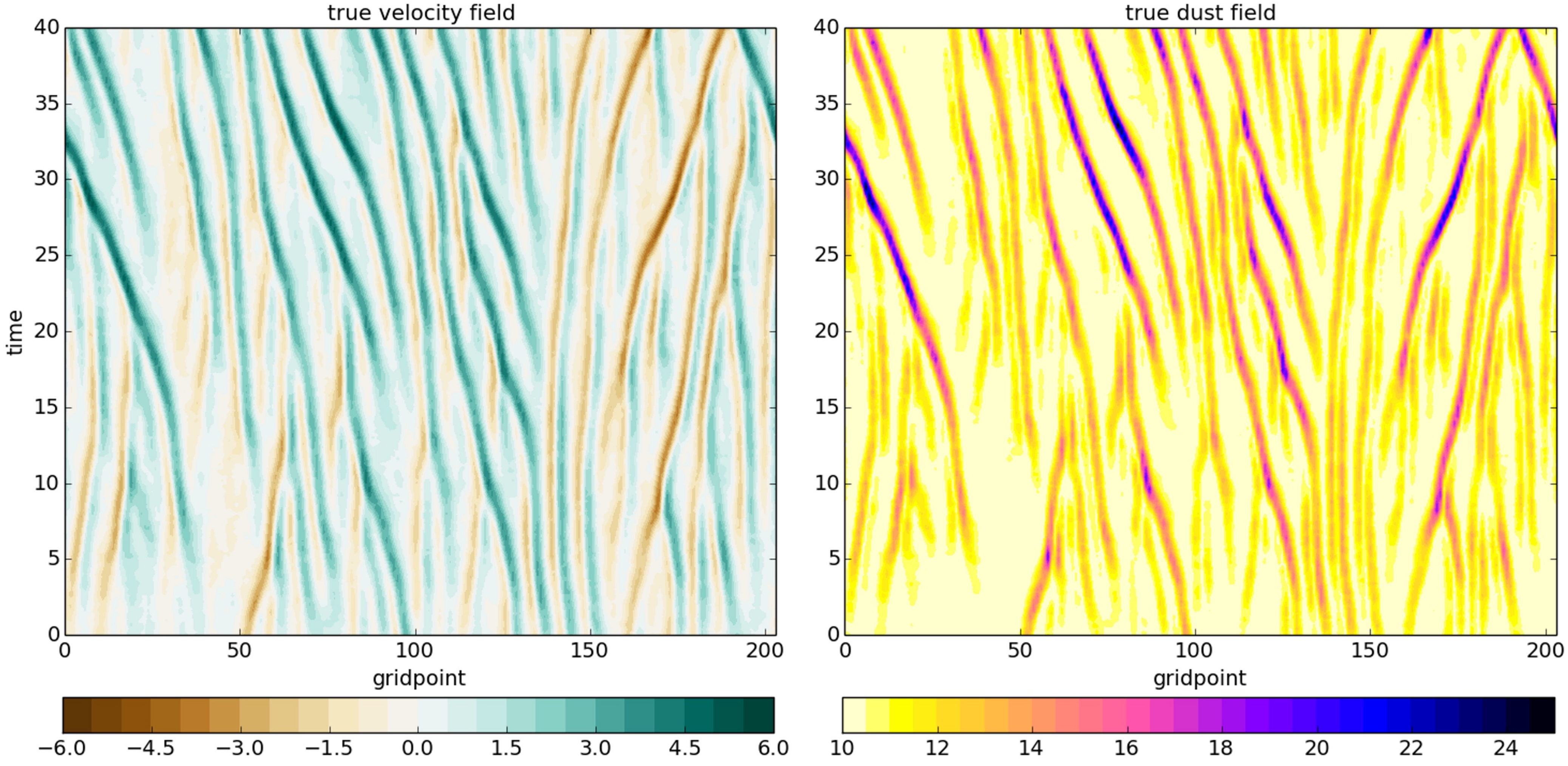


5. In our example we choose m1 2 times, m2 2 times, m3, m4, m5 2 times, m6 and m7.

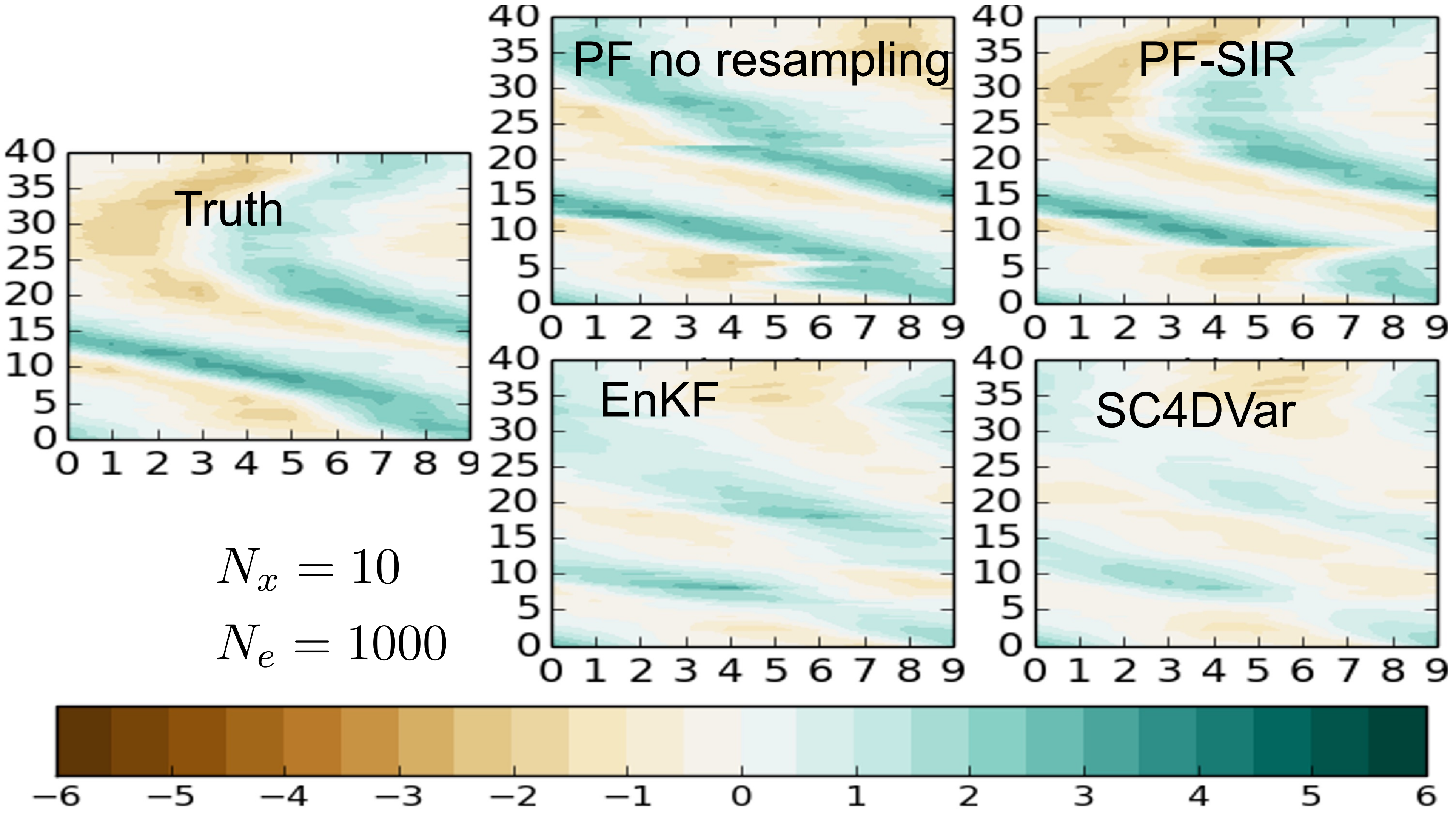
Example: Kuramoto-Shivashinski



Non-linear observations



Example: Kuramoto-Shivashinski



$$N_x = 10$$

$$N_e = 1000$$

What can we do?

Three options have been explored:

1. Localization
2. Proposal densities
3. Particle Flows

Localization in Particle Filters

Easy to make weights spatially varying, similar to observation-space localisation in ETKF.

Two issues:

- How to combine particles from different areas in the domain.
- Number of observations is often too large in local area, still degenerate.

Ensemble Transform Particle Filter (ETPF, Reich, 2014)

Poterjoy (2014, 2022) Complicated scheme that mixes prior and posterior samples and sets minimum weight(!)

Proposal densities

Rewrite Bayes Theorem as

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

in which *q(x|y)* is a so-called proposal density **that depends on the observations!**

Now draw samples x_i from the proposal density instead of from *p(x),* leading to weights

$$w_i = \frac{p(y|x_i)}{p(y)} \frac{p(x_i)}{q(x_i|y)}$$

The samples x_i know about the observations so the likelihood will be larger, leading to more equal weights.

How to choose the proposal density?

The proposal density depends on the observations, **hence we can use any other data-assimilation method** to generate the samples:

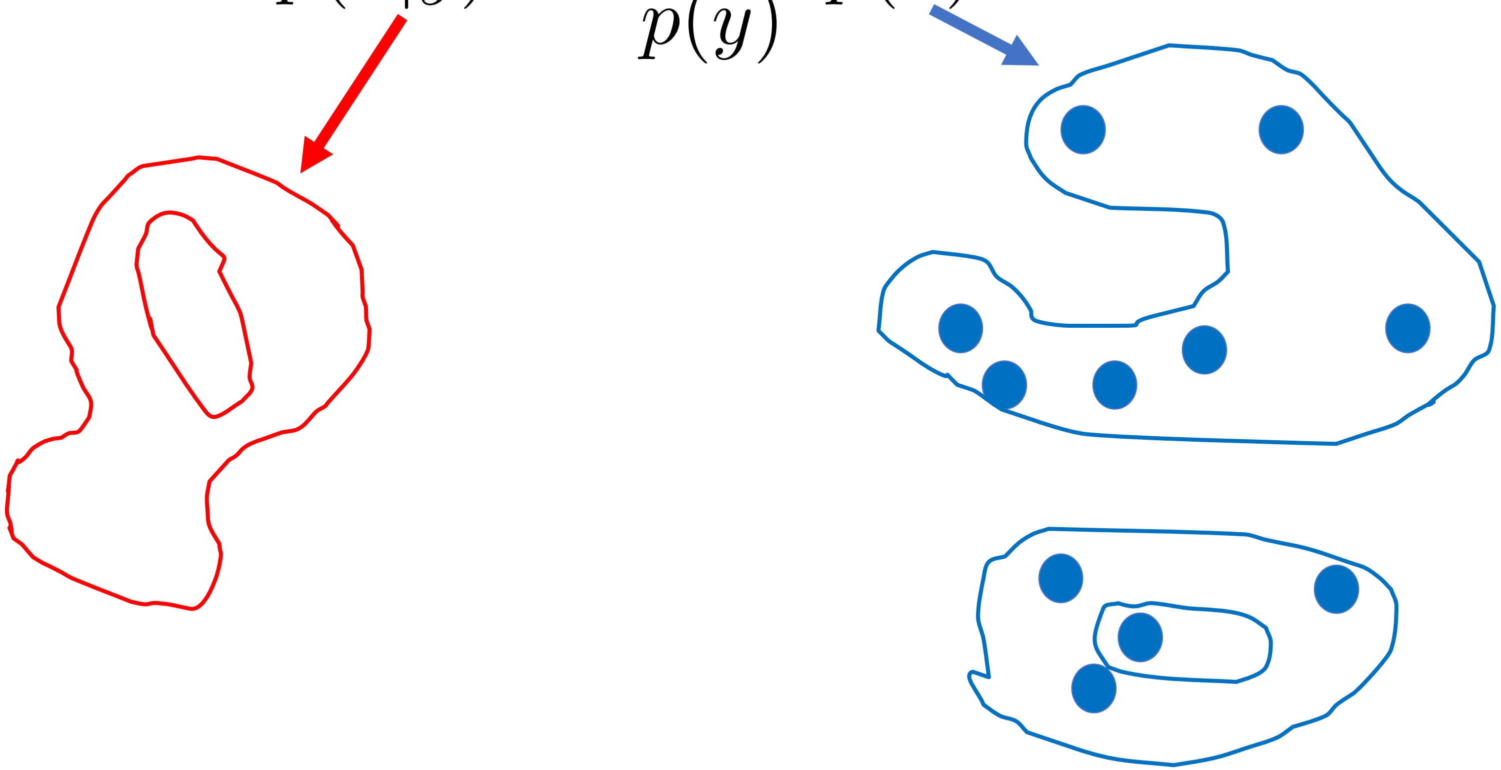
- 3Dvar, gives the so-called optimal proposal density
- 4DVar
- LETKF
- something simple, such as nudging
- synchronization

This is a very natural and principled way to build hybrid schemes without approximations (apart from the finite ensemble).

This has not been explored to the full…

Nonlinear data assimilation: Particle flows

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

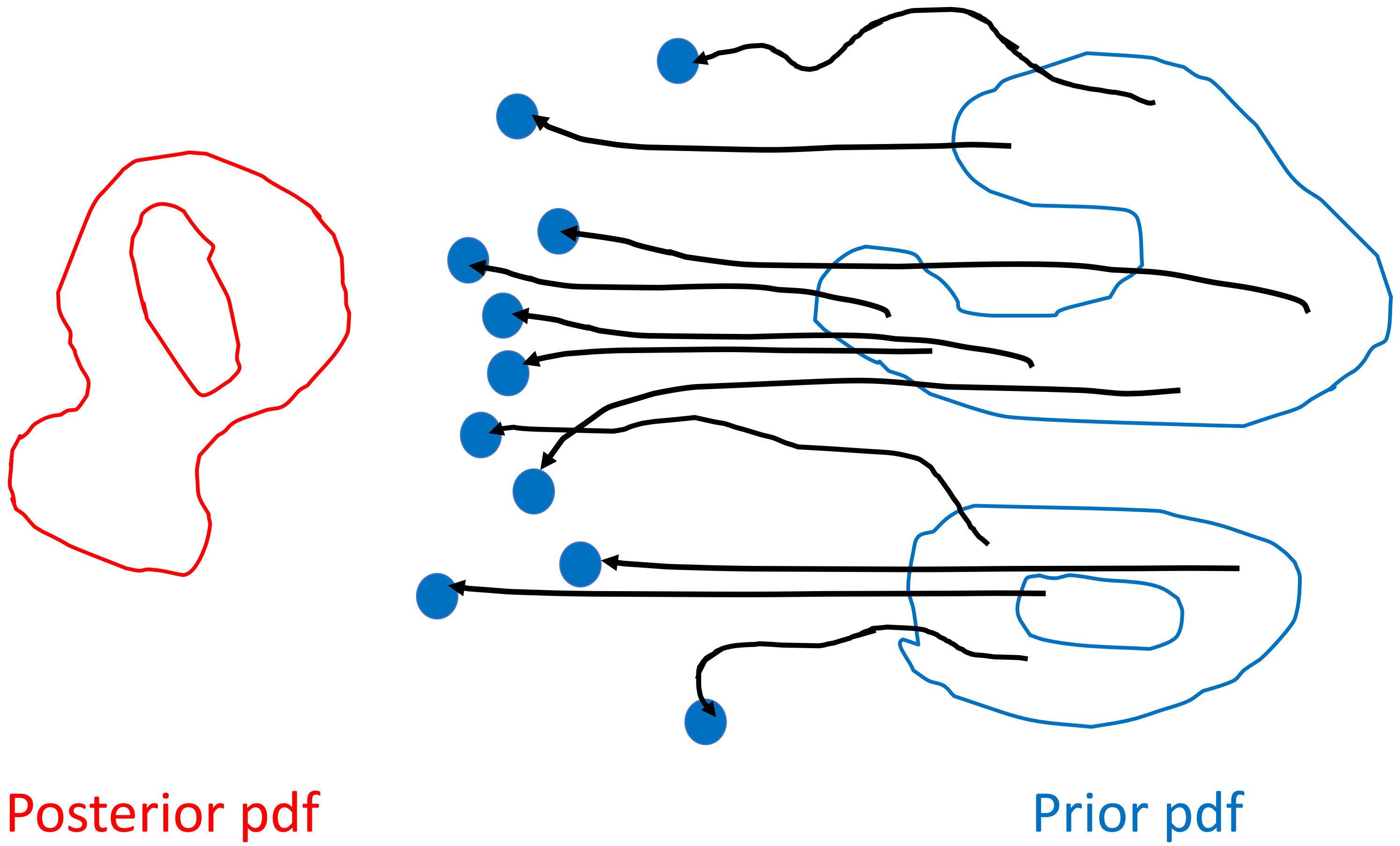


Posterior pdf

Prior pdf

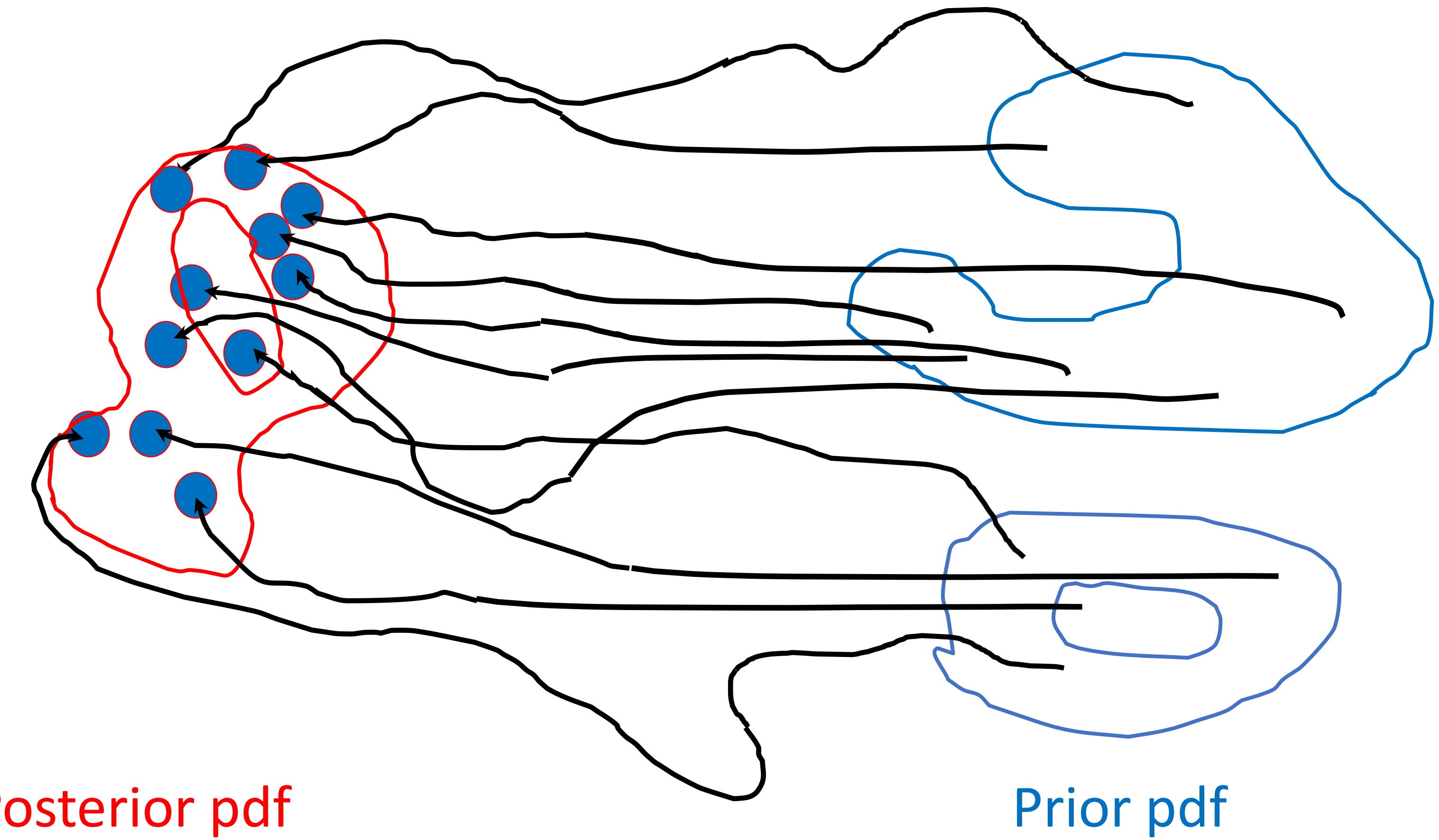
The prior and posterior can be for a model state (filter) or a model trajectory (smoother) or a set of parameters, or a combination of these.

Particle flow in *pseudo time*



Particle Flow converged on posterior pdf

Not degenerate by construction



Hu, C-C, and P.J. van Leeuwen (2021) A particle flow filter for fully nonlinear high-dimensional data assimilation., Q.J. Royal Meteorol. Soc., doi: 10.1002/qj.4028

What is a particle flow filter/smoother?

- A particle flow moves particles iteratively from samples of the prior to samples of the posterior with a flow field $f_s(x)$ in which x is a model state/trajectory over the time window

$$\frac{dx}{ds} = f_s(x) \quad \text{hence, for pdf:} \qquad \frac{\partial p_s(x)}{\partial s} = -\nabla_x \cdot (p_s(x) f_s(x))$$

- The flow field $f_s(x)$ can be chosen in many ways, let's choose an optimal way.
- 'Optimal' depends on distance metric, we use Kullback-Leibler divergence between intermediate pdf $p_s(x)$ and the posterior pdf $p(x|y)$.
- To solve the problem we embed the flow field in a Reproducing Kernel Hilbert Space (Liu & Wang 2016).

Finding the flow field

Kullback-Leibler divergence (or relative entropy):

$$KL(p_s(x) \| p(x|y)) = \int p_s(x) \log \frac{p_s(x)}{p(x|y)} \, dx$$

Hence

$$\frac{dKL}{ds} = \int \frac{dp_s(x)}{ds} \left[1 + \log \frac{p_s(x)}{p(x|y)} \right] \, dx$$

Use FP equation for the intermediate pdf $p_s(x)$:

$$\frac{dKL}{ds} = -\int \nabla(p_s(x) f_s(x)) \left[1 + \log \frac{p_s(x)}{p(x|y)} \right] \, dx$$

A practical method for the flow field

We found:

$$\frac{dKL}{ds} = -\int p_s(x) f_s(x) \cdot \nabla_x \log\left(\frac{p_s(x)}{p(x|y)}\right)\, dx$$

Assume $f_s(x)$ is in a Reproducing Kernel Hilbert Space:

$$f_s(x) = \langle K(x,.), f(.)\rangle$$

Since we have 'total' freedom on $f_s(x)$ we still solve full problem.
Use this in the expression above to find:

$$\frac{dKL}{ds} = \left\langle \int p_s(x) K(x',.) \nabla_x \log\left(\frac{p_s(x)}{p(x|y)}\right)\, dx, f_s(.)\right\rangle_{\mathcal{F}}$$

$$= \langle \nabla KL, f_s(.)\rangle_{\mathcal{F}}$$ with ∇KL the gradient of *KL*.

The flow field

The distance reduces if $$f_s(x) = -a\nabla KL(x)$$

for some scalar *a.*
We can write:

$$\nabla KL(x) = \int p_s(x')K(x',x)\nabla_{x'}\log\left(\frac{p_s(x')}{p(x'|y)}\right)\,dx'$$
$$= -\int p_s(x')\left[K(x',x)\nabla_{x'}\log p(x'|y) + \nabla_{x'}K(x',x)\right]\,dx'$$

Using a particle representation for $p_s(x)$ gives for the flow field:

$$f_s(x) = \frac{1}{N}\sum_{i=1}^{N}\left[k(x_i^s,x)\nabla_{x_i^s}\log p(x_i^s|y) + \nabla_{x_i^s}k(x_i^s,x)\right]$$

Particle implementation

The evolution equation *for each particle* is now

$$\frac{dx_i}{ds} = f_s(x_i) = -a\nabla KL$$

This leads to an iterative scheme

$$x_i^{(j)} = x_i^{(j-1)} + \epsilon f_s(x_i)$$

until

$$|\nabla KL| \leq \epsilon_{KL}$$

The flow field in a Particle Flow Filter/Smoother

- The flow field is found as

$$f_s(x) = \frac{B}{N}\sum_{i=1}^{N} \underbrace{\boxed{K(x^i, x)}}_{\text{kernel}} \underbrace{\boxed{\nabla_{x^i} \log p(x^i|y)}}_{\text{grad posterior pdf}} + \nabla_{x^i} K(x^i, x)$$

kernel — grad posterior pdf

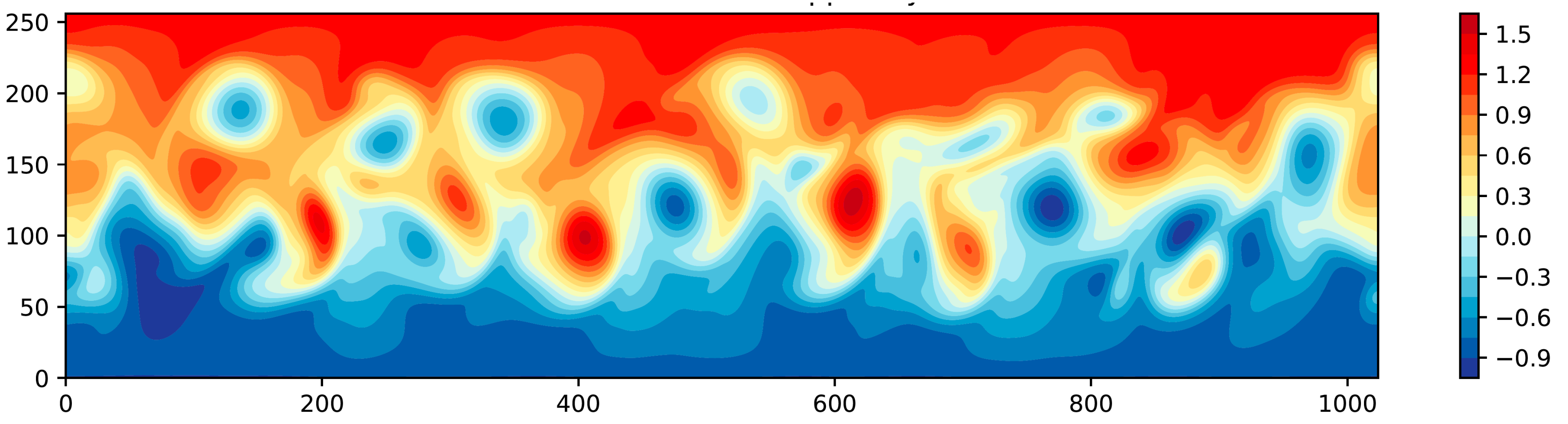
attracting term — *repelling term*

- The attracting term pulls the particles towards the posterior mode
- The repelling term ensures coverage of the posterior pdf
- We use a matrix-valued kernel as in Hu and Van Leeuwen (QJRMS, 2021)

Particle Flow Filter on a high dimensional ocean model

- 2-layer Quasi-geostrophic model of the Antacrtic Circumpolar Current, 526,850 gridpoints
- Assimilation of upper-layer variables once a day, of every 5th gridpoint, 9600 observations
- $H(x) = x^2$, Observation error 5 cm^2,
- 50 particles, initialized from samples from long model run

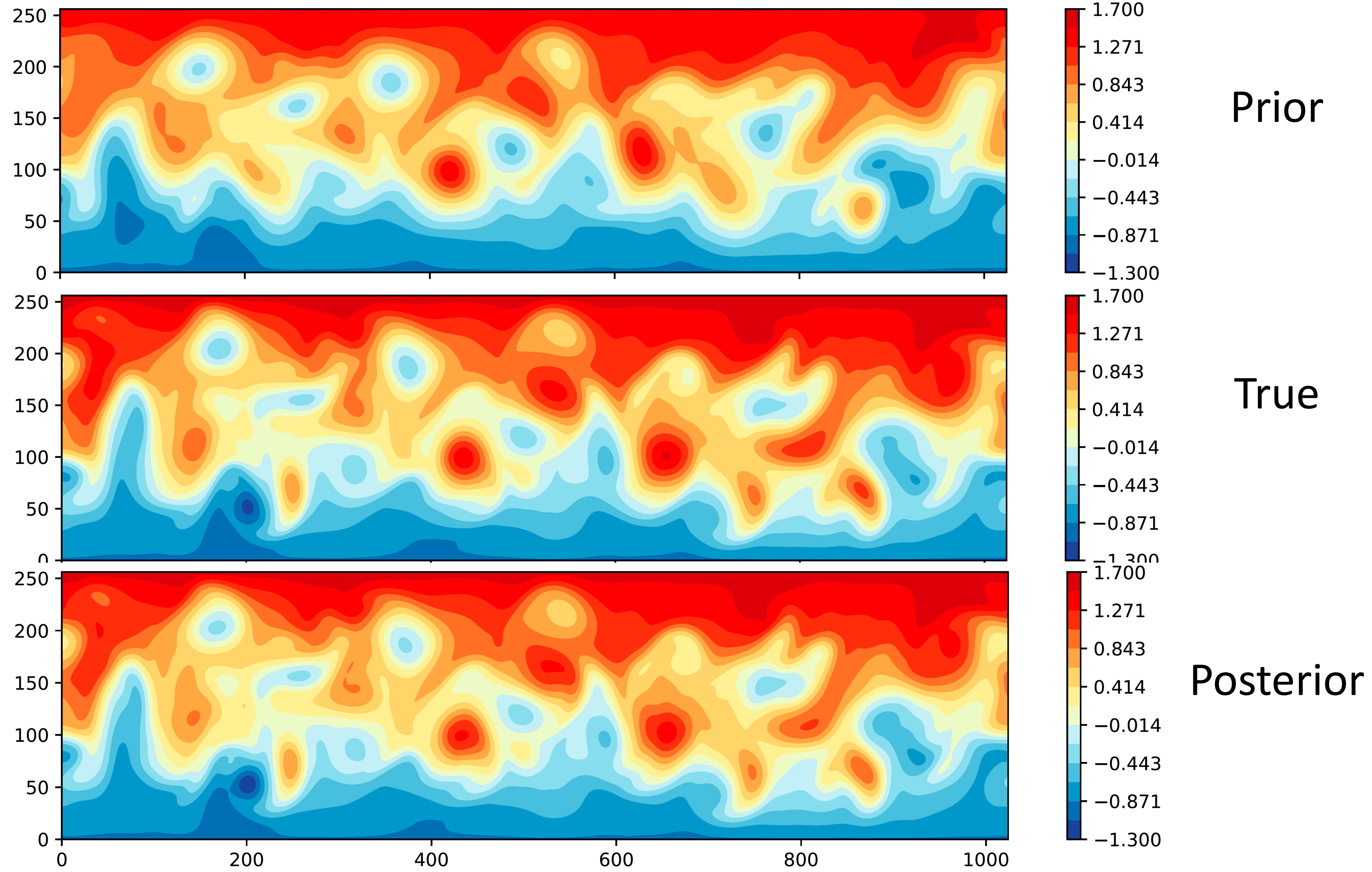
Example of SSH field (m), grid spacing 5 km, 5125 km by 1285 km.



Update mean SSH field (m)

Note:

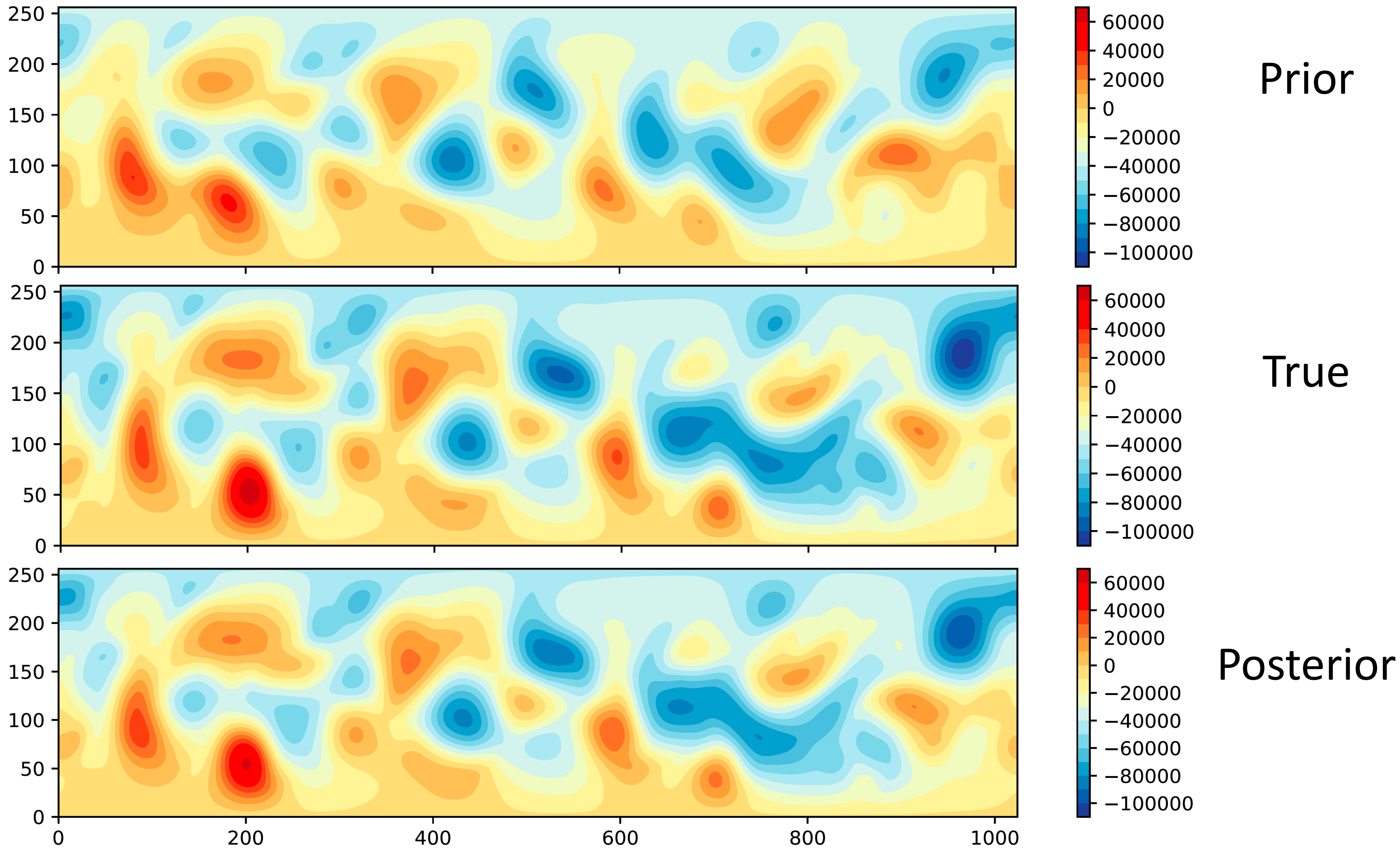
1) The observation operator is SSH squared !
2) The large difference between prior and true field.
3) The close resemblance between posterior and true fields.



Update mean streamfunction lower layer (m^2s^{-1})

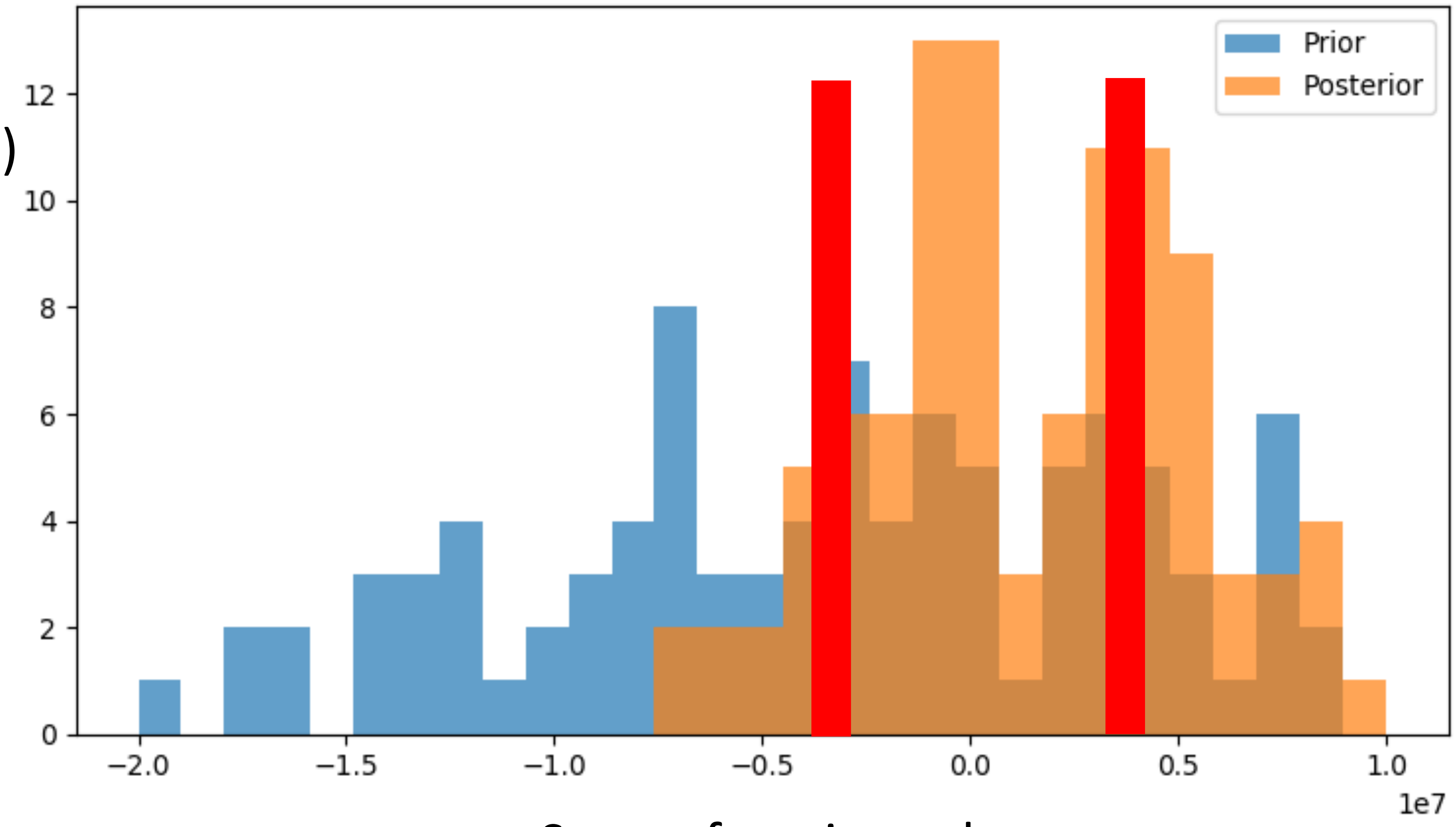
Note:

1) This layer is not observed
2) The large difference between prior and true field.
3) The close resemblance between posterior and true fields.



Example of histogram for H(x) = x^2, at one grid point

Probability p(x)



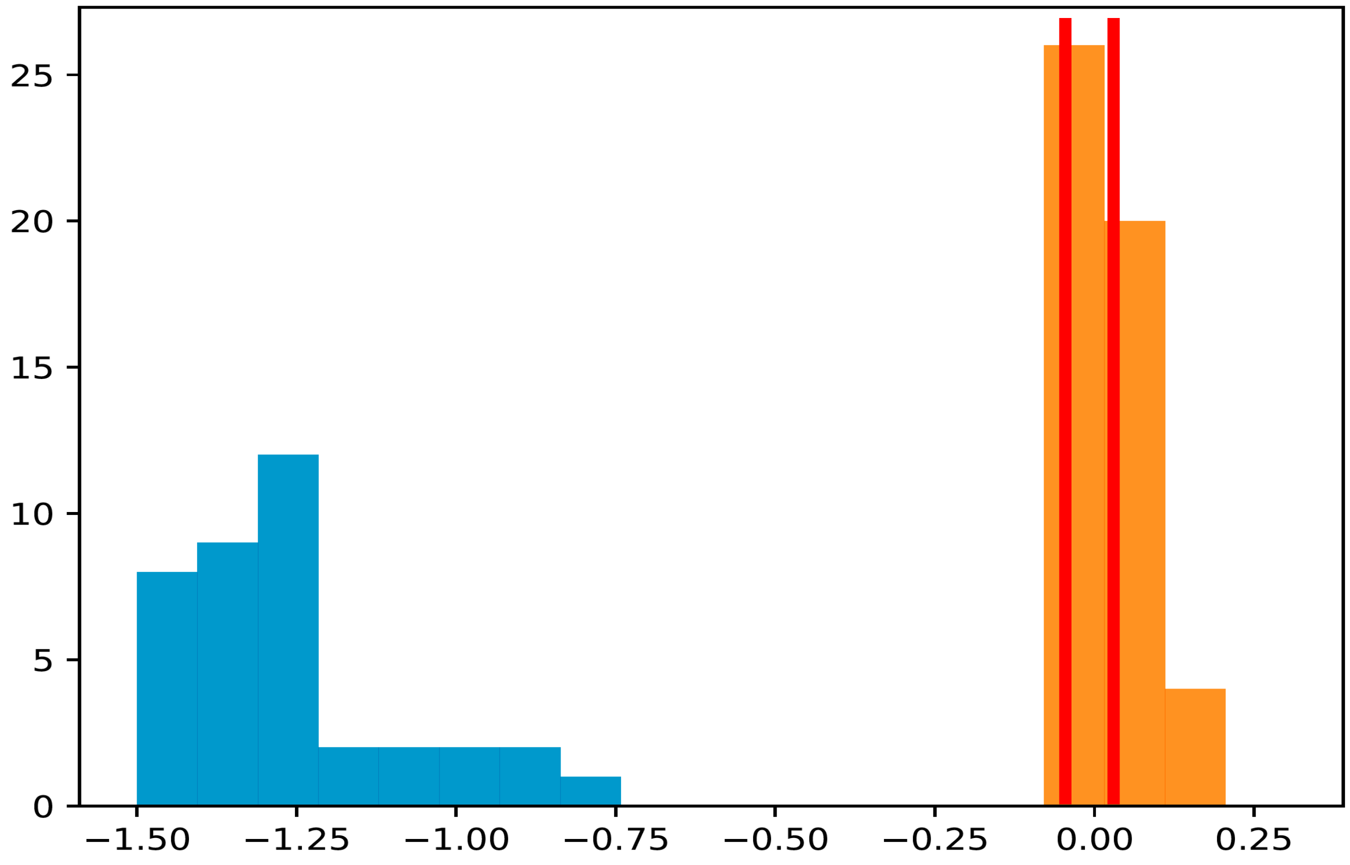
Streamfunction value

Red bars denote this single observation in state space

Example of histogram for H(x) = x^2, at one grid point

Probability p(x)

Red bars denote this single observation in state space



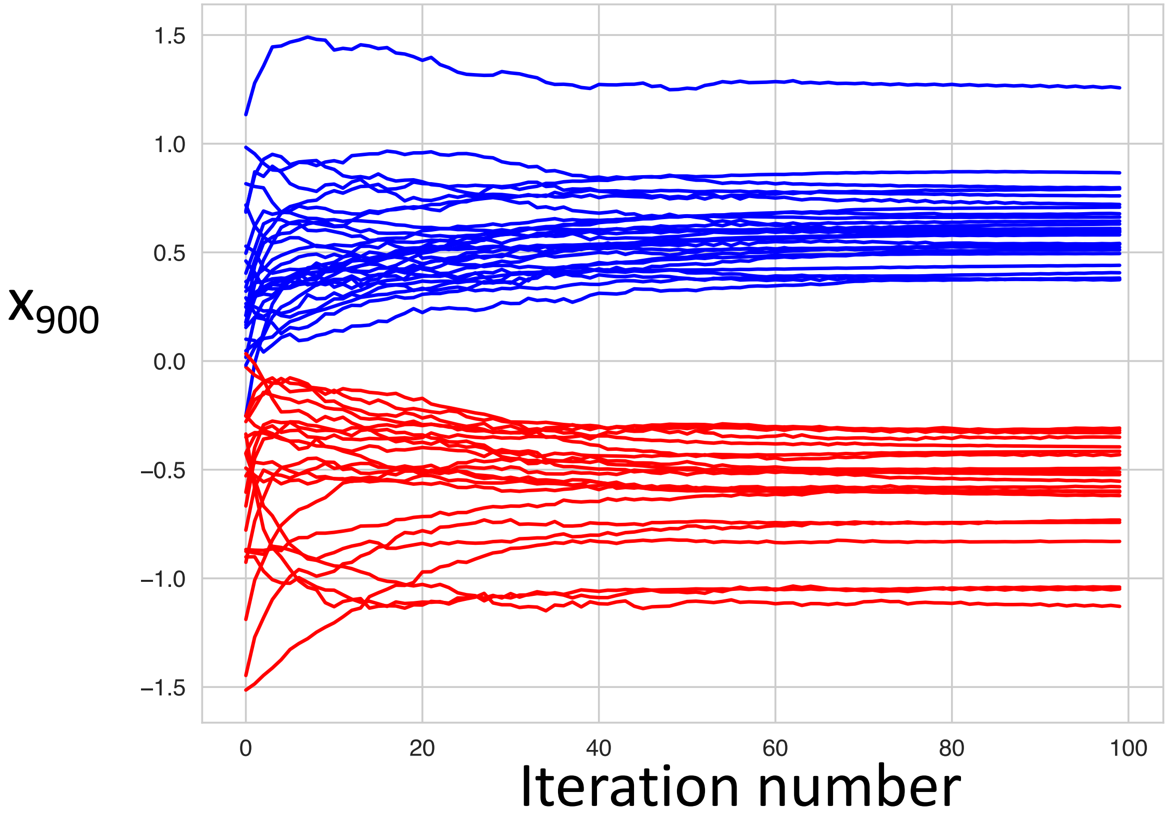
SSH value

Example of Particle Flow Smoother on Lorenz 1996 model

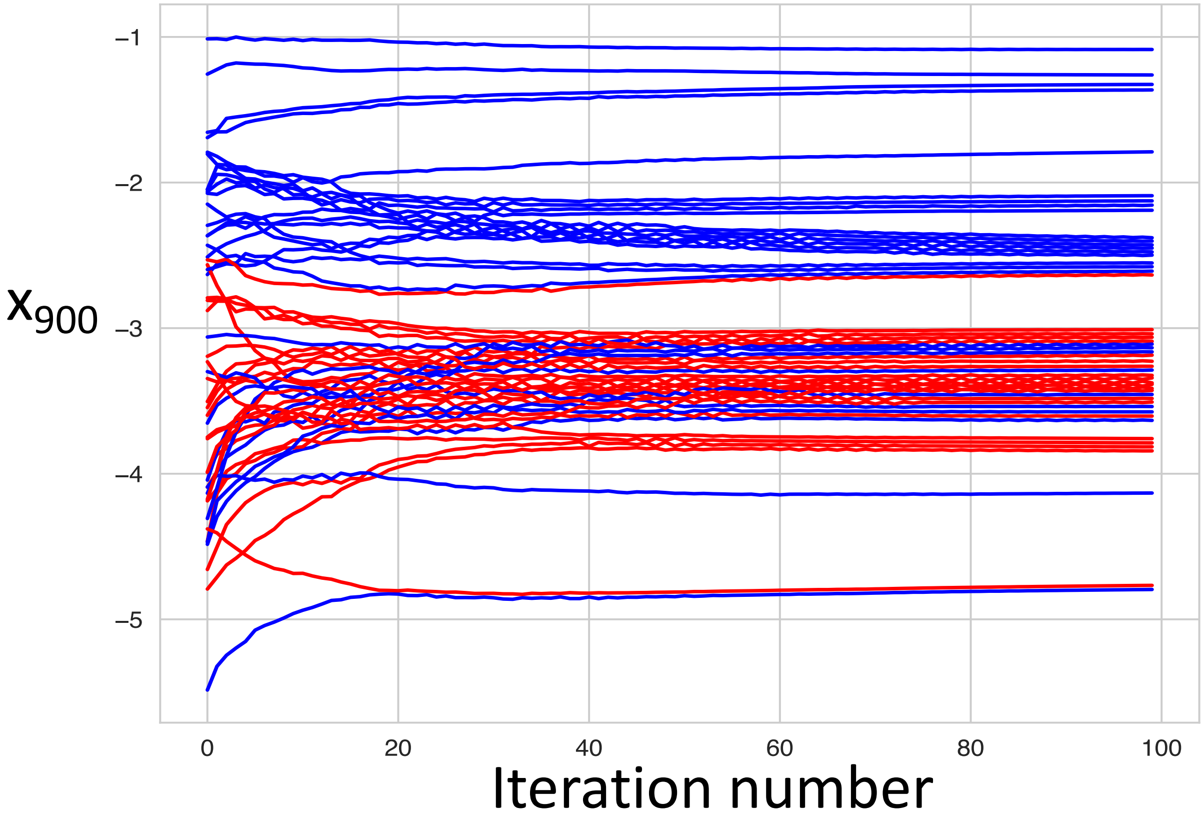
$$\frac{dx_i}{dt} = (x_{i+1} - x_{i-2})x_{i-1} - x_i + F$$

- 1000 dimensional
- Assimilation window 10 time steps of time step 0.02
- Initial state error 1.0
- Observation every other gridpoint at end of assimilation window
- $H(x) = |x|$
- Observation error 0.1
- 50 particles
- Comparison with perturbed-observation ensemble of 4DVars

Convergence of Particle Flow Smoother

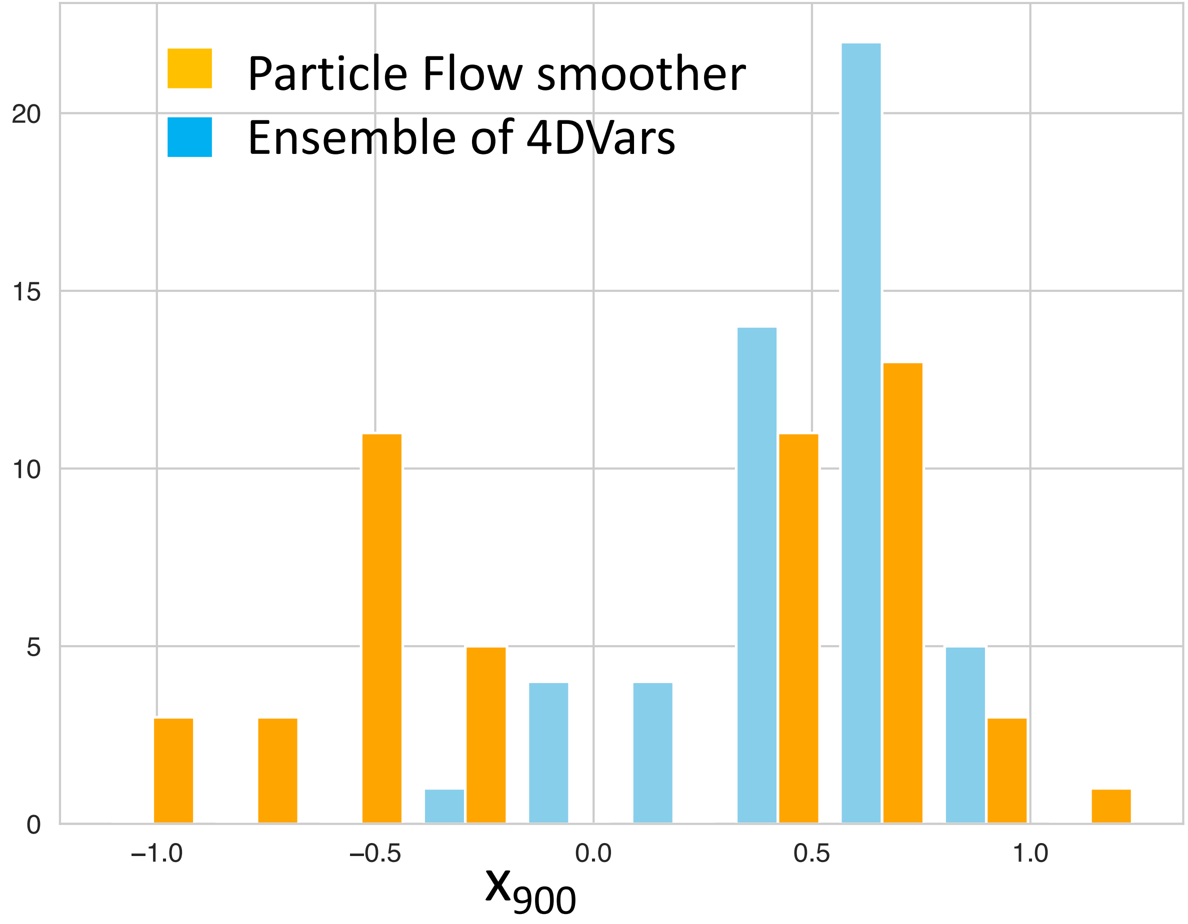


Convergence of particles at *end* of assimilation window

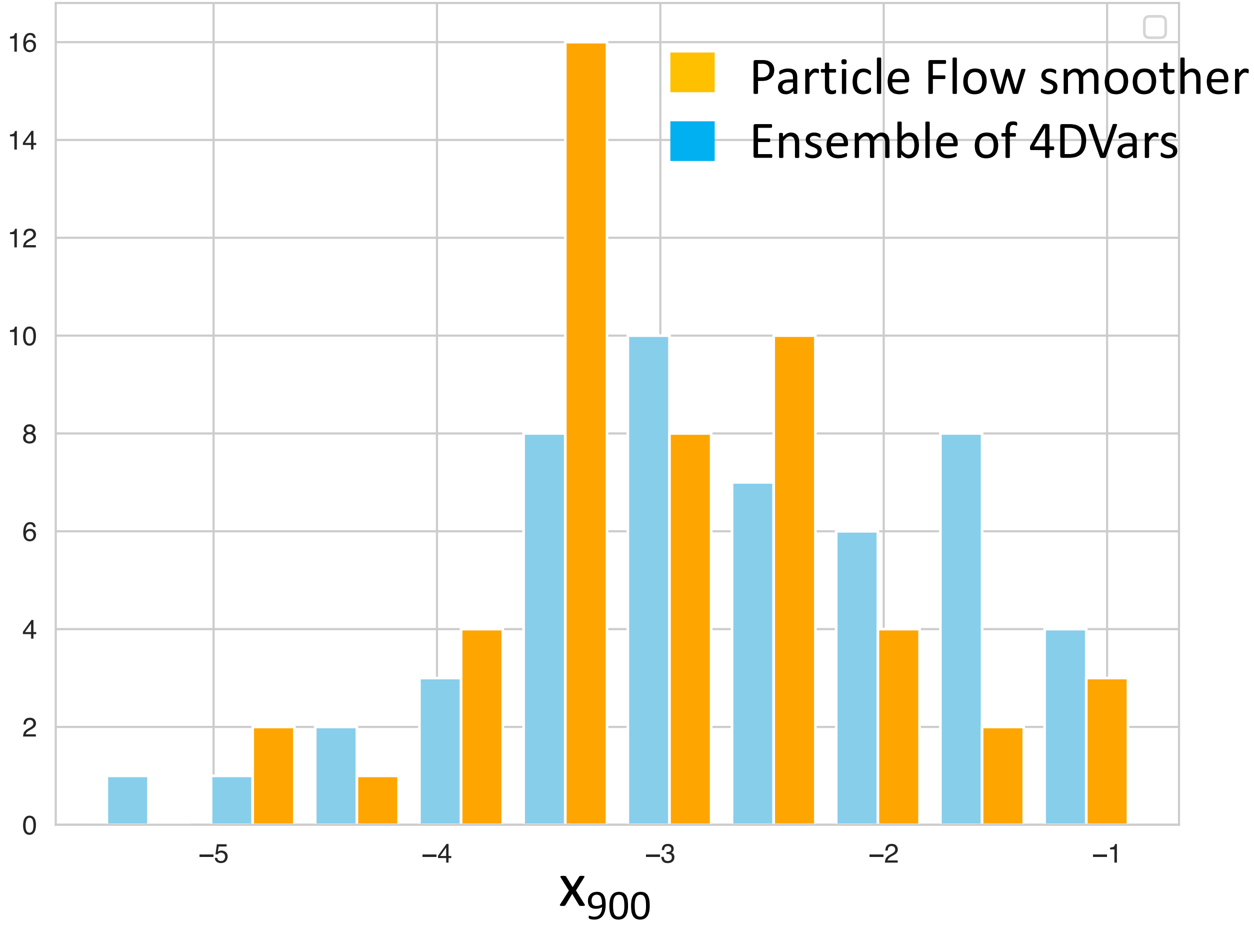


Convergence of particles at *start* of assimilation window

Histograms comparison with Ensemble of 4DVars

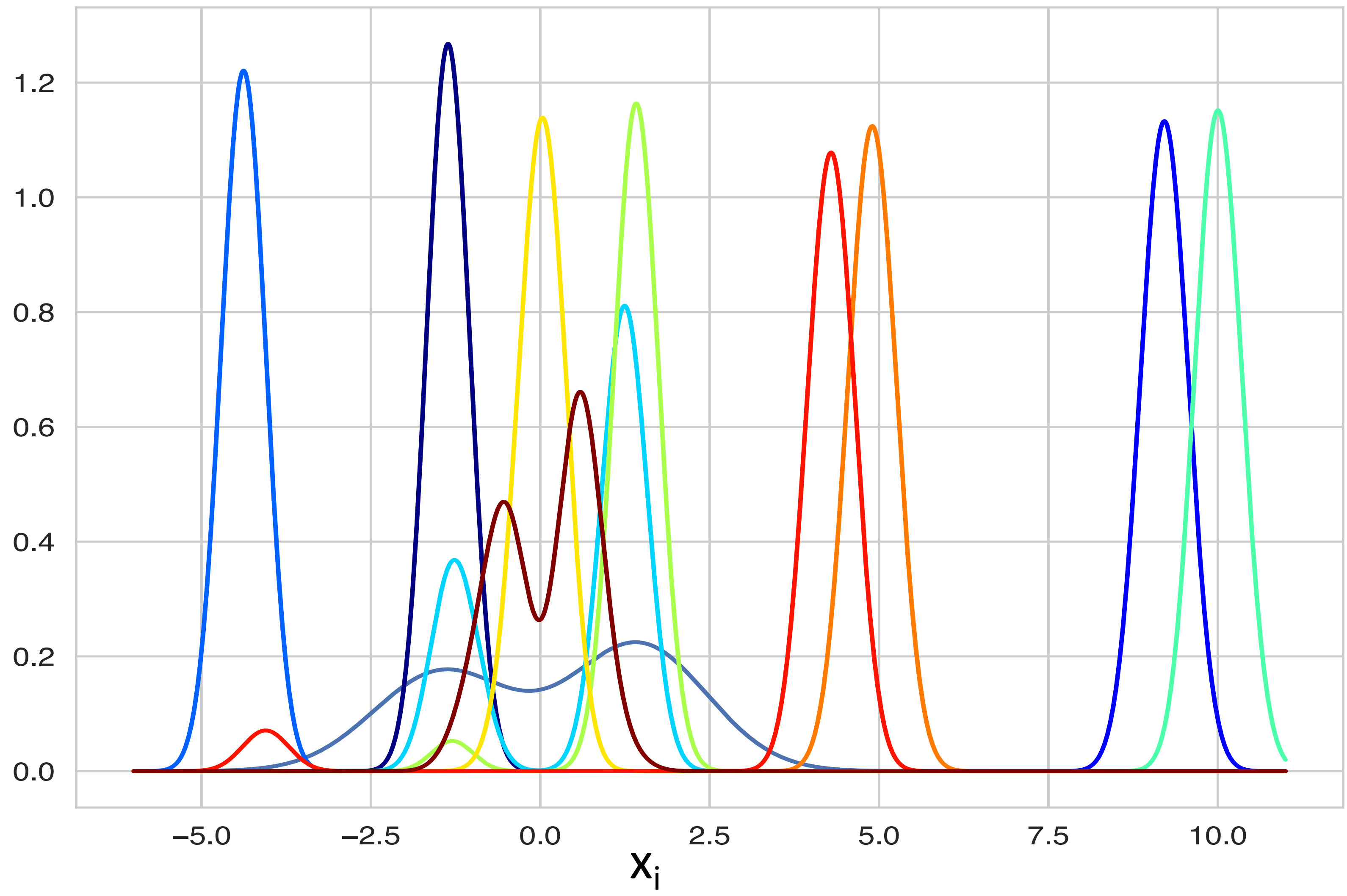


Histograms at *end* of assimilation window



Histograms at *start* of assimilation window

Example pdfs at final time for different grid points



250 bimodal marginal pdfs using 50 particles

These will be used for forecasts!

Conclusions

- Fully nonlinear data-assimilation methods exist and are slowly becoming mainstream.
- Standard Particle Filers are extremely flexible and easy to implement, but suffer from weight degeneracy
- Localization in particle filters is maturing, but weight collapse in local areas remains a problem
- Particle Flow filters (and smoothers) are not degenerate by construction.
- Very promising in high-dimensional applications
- Cost equivalent to ensemble of 3DVars (filter) or strong-constraint 4DVars (smoother).
- Working on implementation in DART (filter) and JEDI (smoother).
- Much need for more scientists working in. nonlinear DA!