Particle Filters
Part I:
Theory

Peter Jan van Leeuwen

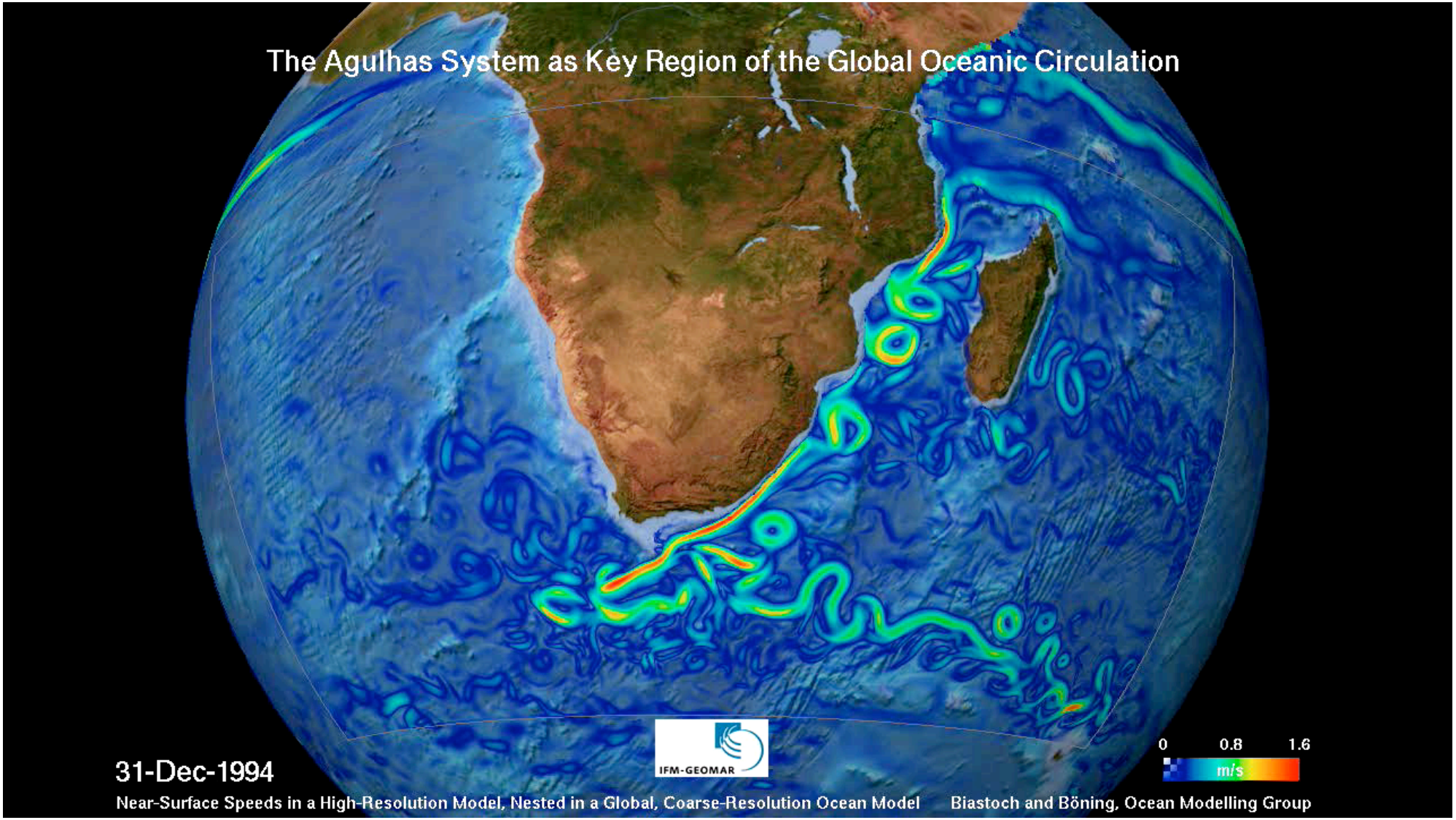
Data-Assimilation Research Centre DARC

University of Reading

IIRS, Dehra Dun, December 2012

Why Data Assimilation

- Prediction
- Model improvement:
  - Parameter estimation
  - Parameterisation estimation
- Increase our understanding


The Agulhas System as Key Region of the Global Oceanic Circulation
IFM-GEOMAR
0
0.8
1.6
m/s
31-Dec-1994
Near-Surface Speeds in a High-Resolution Model, Nested in a Global, Coarse-Resolution Ocean Model
Biastoch and Böning, Ocean Modelling Group

Data Assimilation Ingredients

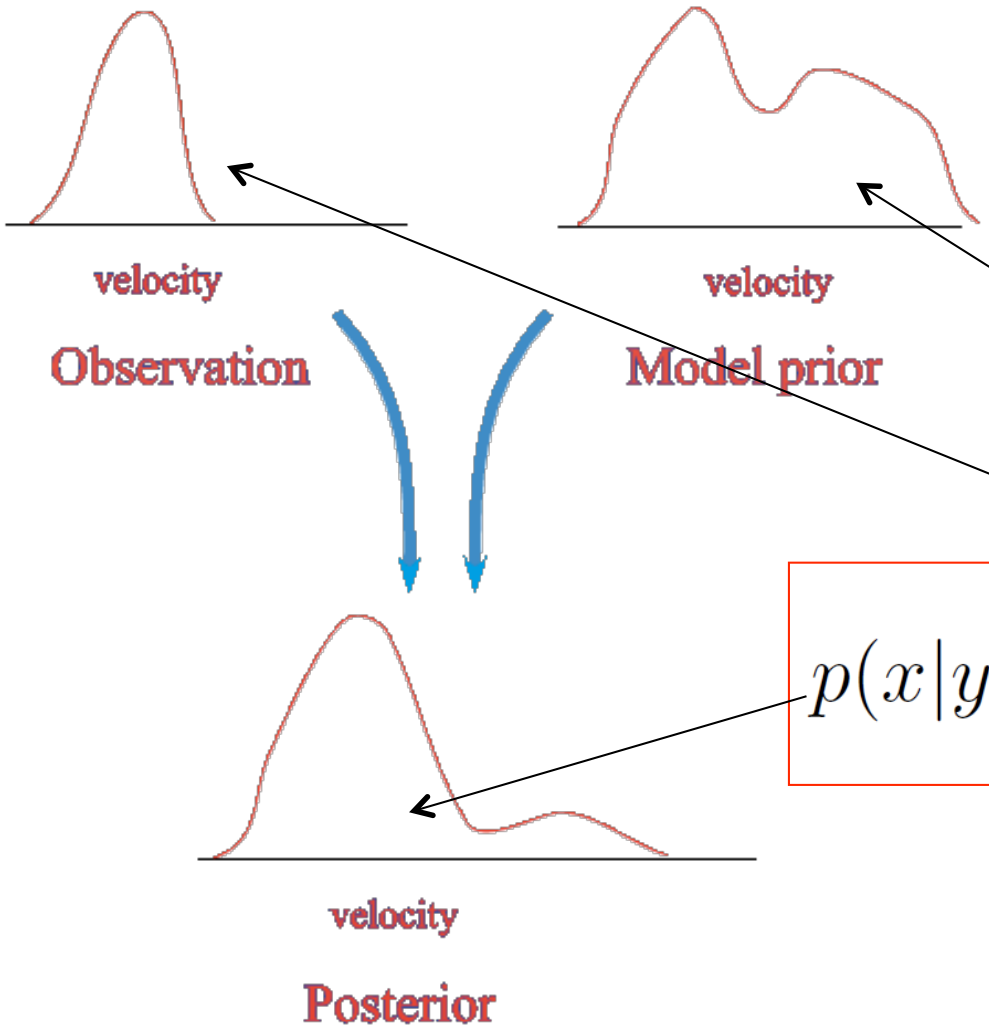
- Prior knowledge, the Stochastic model:

$$x^n = f(x^{n-1}) + \beta^{n-1}$$

- Observations: y^n

- Relation between the two: $y^n = H(x^n) + \epsilon^n$

Data assimilation: general formulation



Bayes theorem:

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

Solution is pdf!

NO INVERSION !!!

Parameter estimation:

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

with

$$y = H(\theta) + \epsilon$$

Again, no inversion but a direct point-wise multiplication.

Propagation of pdf in time: Kolmogorov's equation

Model equation:

$$x^n = f(x^{n-1}) + \beta^{n-1}$$

Pdf evolution: Kolmogorov's equation
(Fokker-Planck equation)

$$\frac{\partial p(x,t)}{\partial t} = -\frac{\partial f(x,t)p(x,t)}{\partial x} + \frac{1}{2}\frac{\partial^2 p(x,t)Q}{\partial x^2}$$

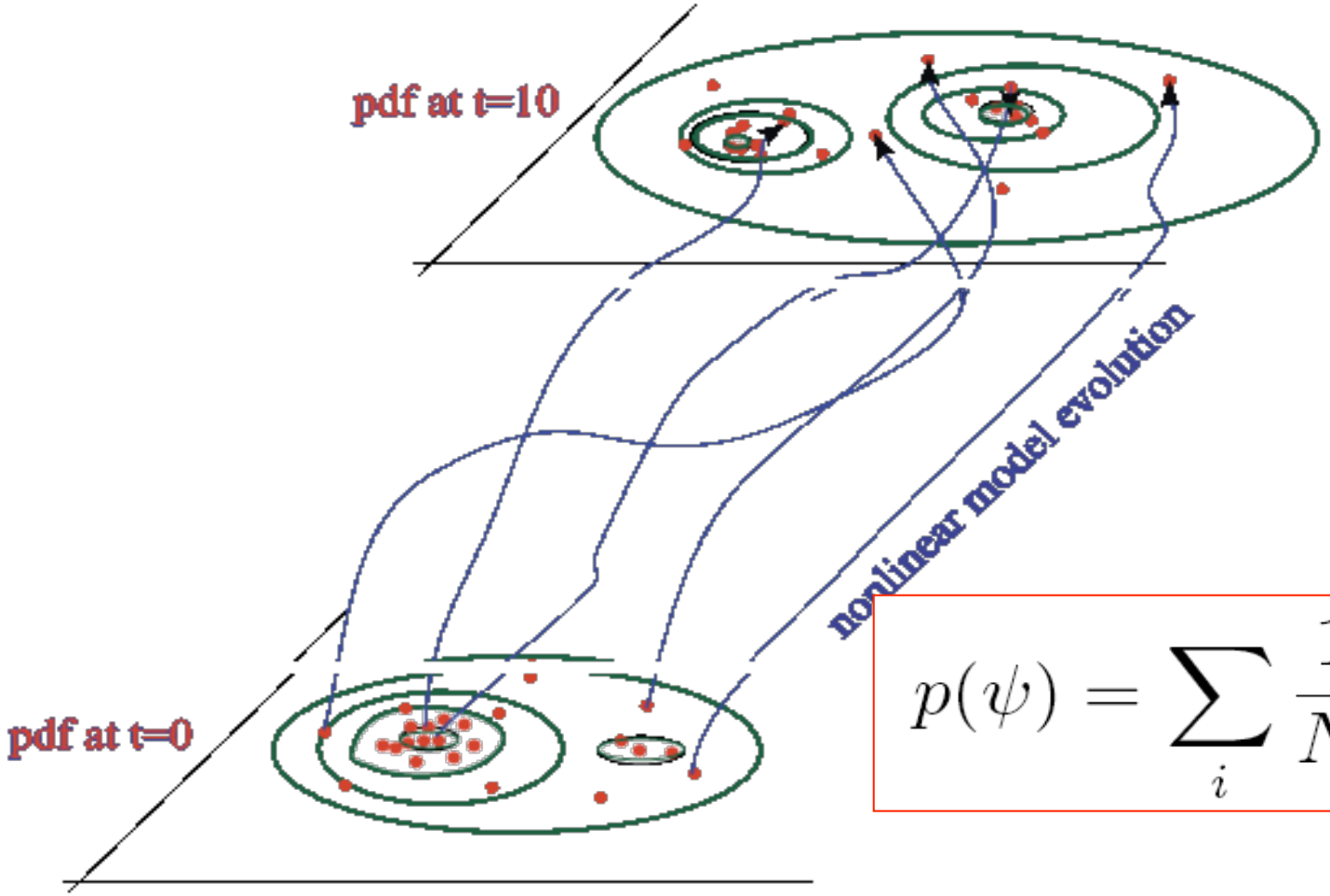
advection

diffusion

Too expensive !!!

Motivation ensemble methods:

'Efficient' propagation of pdf in time (for nonlinear models)



$$p(\psi) = \sum_i \frac{1}{N} \delta(\psi - \psi_i)$$

How is DA used today in geosciences?

Present-day data-assimilation systems are based on linearizations and state covariances are essential.

4DVar, Representer method (PSAS):

- Gaussian pdf's , solves for posterior mode, needs error covariance of initial state (B matrix), 'no' posterior error covariances

(Ensemble) Kalman filter:

- assumes Gaussian pdf's for the state, approximates posterior mean and covariance, doesn't minimize anything in nonlinear systems, needs inflation and localisation

Combinations of these: hybrid methods (!!!)

Non-linear Data Assimilation

- Metropolis-Hastings
- Langevin sampling
- Hybrid Monte-Carlo
- Particle Filters/Smoothers
- Combinations of MH and PF

All try to sample from the posterior pdf, either the joint-in-time, or the marginal. Only the particle filter/smoother does this sequentially in time.

Nonlinear filtering: 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 pdf of the observations given model state x_i.
- For Gaussian distributed variables is is given by:

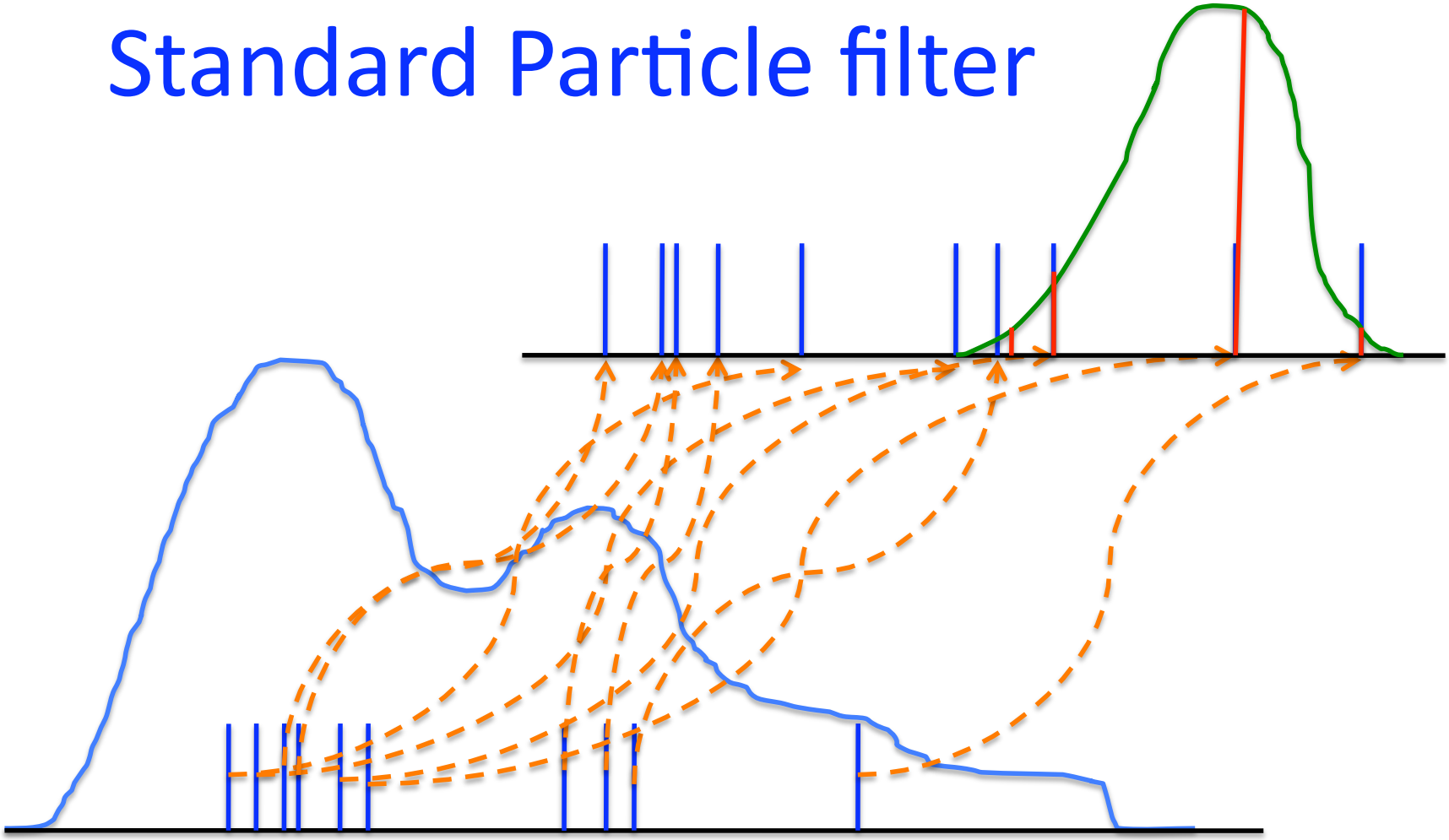
$$\begin{aligned} w_i &\propto p(y|x_i) \\ &\propto \exp\left[-\frac{1}{2}\left(y - H(x_i)\right) R^{-1} \left(y - H(x_i)\right)\right] \end{aligned}$$

- One can just calculate this value
- That is all !!!

No explicit need for state covariances

- 3DVar and 4DVar need a good error covariance of the prior state estimate: complicated
- The performance of Ensemble Kalman filters relies on the quality of the sample covariance, forcing artificial inflation and localisation.
- Particle filter doesn't have this problem, but…

Standard Particle filter



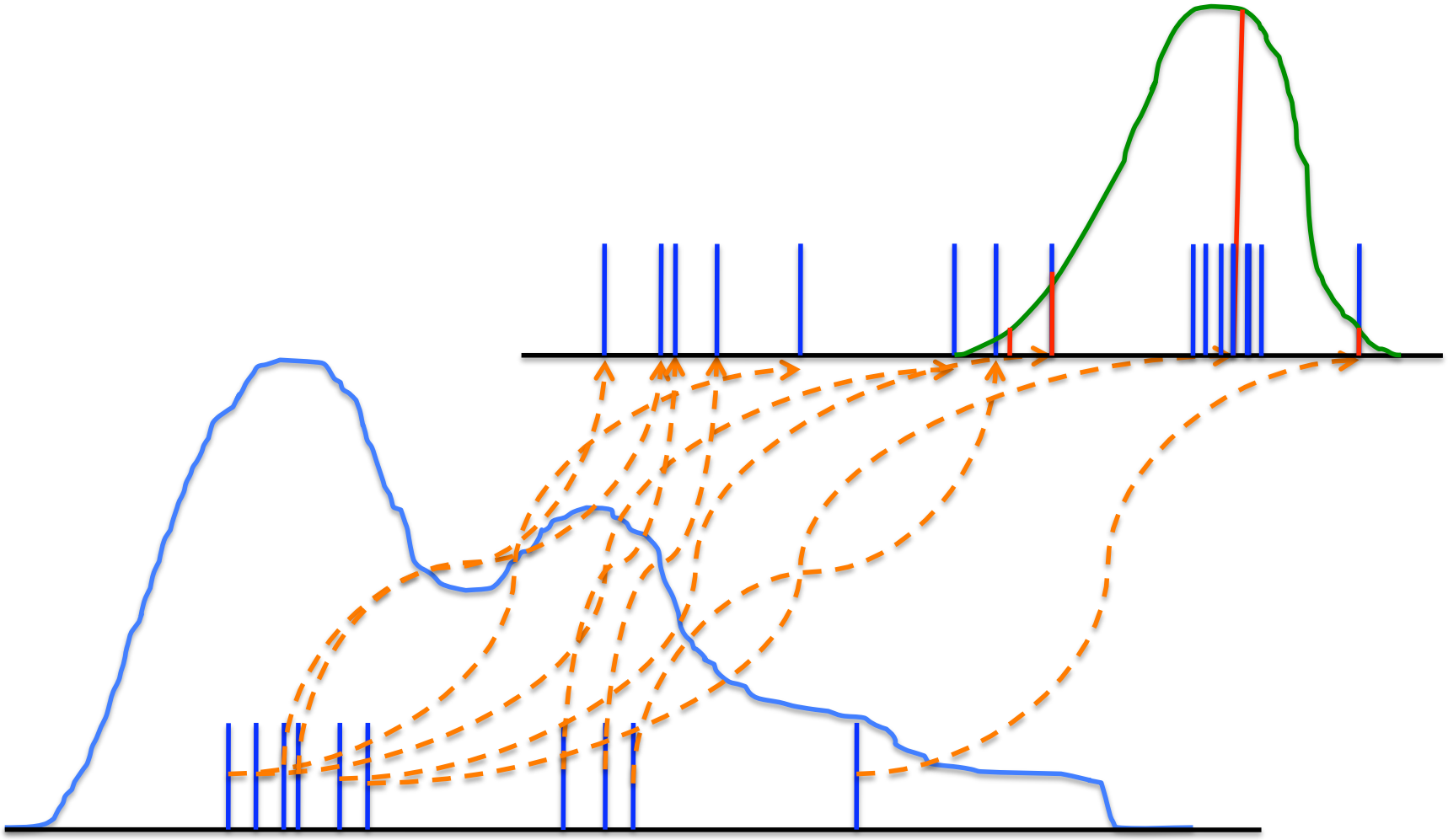
The standard particle filter is degenerate for moderate ensemble size in moderate-dimensional systems.

Particle Filter degeneracy: resampling

- With each new set of observations the old weights are multiplied with the new weights.
- Very soon only one particle has all the weight…
- Solution:

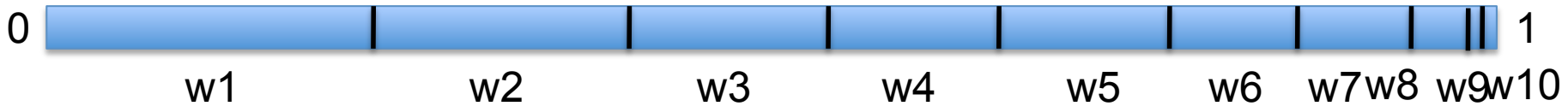
  Resampling: duplicate high-weight particles and abandon low-weight particles

Standard Particle filter

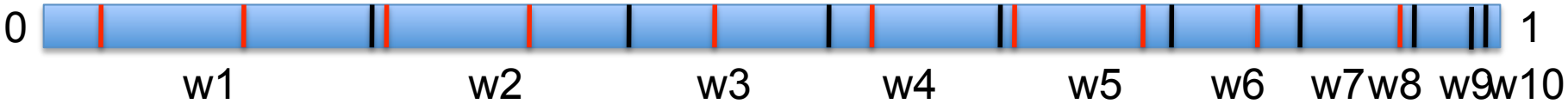


A simple resampling scheme

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



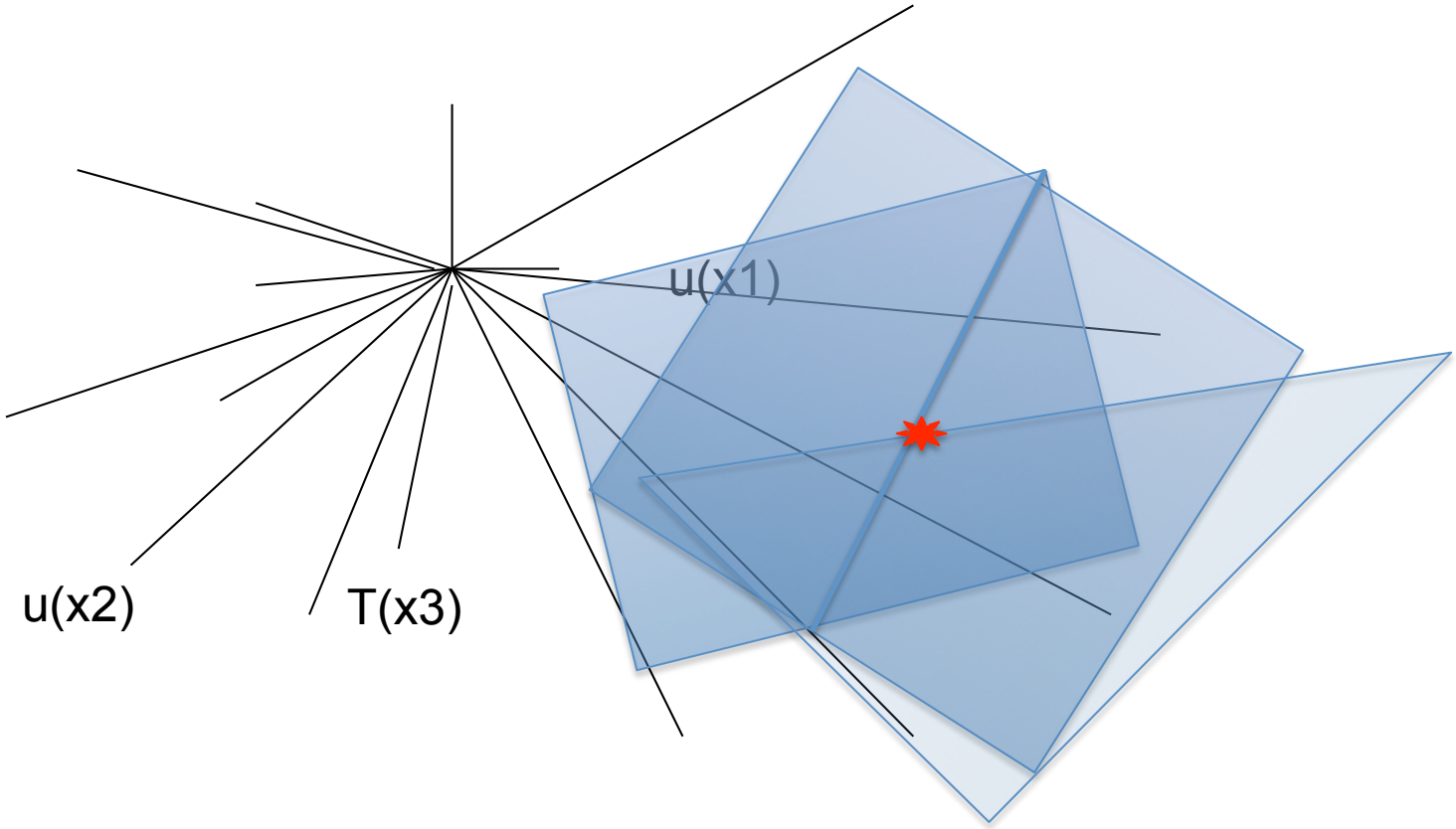
2. Draw a random number from the uniform distribution over [0,1/N], in this case with 10 members over [0,1/10].
3. Put that number on the unit interval: its end point is the first member drawn.
4. Add 1/N to the end point: the new end point is our second member. Repeat this until N new members are obtained.



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

A closer look at the weights I

Probability space in large-dimensional systems is ‘empty’: the curse of dimensionality



A closer look at the weights II

Assume particle 1 is at 0.1 standard deviations *s* of M independent observations.
Assume particle 2 is at 0.2 *s* of the M observations.

The weight of particle 1 will be

$$w_1 \propto \exp\left[-\frac{1}{2}\ (y - H(x_i))\, R^{-1}\, (y - H(x_i))\right] = exp(-0.005M)$$

and particle 2 gives

$$w_2 \propto \exp\left[-\frac{1}{2}\ (y - H(x_i))\, R^{-1}\, (y - H(x_i))\right] = exp(-0.02M)$$

A closer look at the weights III

The ratio of the weights is

$$\frac{w_2}{w_1} = exp(-0.015M)$$

Take M=1000 to find

$$\boxed{\frac{w_2}{w_1} = exp(-15) \approx 3\ 10^{-7}}$$

Conclusion: the number of independent observations is responsible for the degeneracy in particle filters.

A closer look at the weights IV

- The volume of a hypersphere of radius *r* in an *M* dimensional space is

$$V \propto \frac{r^M}{\Gamma(M/2-1)}$$

- Taking for the radius $r \approx 3\sigma_y$ we find, using Stirling:

$$V \propto \left[\frac{9\sigma_y}{M/2}\right]^{M/2}$$

- So very small indeed.

How can we make particle filters useful?

The joint-in-time prior pdf can be written as:

$$p(x^n, x^{n-1}) = p(x^n|x^{n-1})p(x^{n-1})$$

So the marginal prior pdf at time *n* becomes:

$$p(x^n) = \int p(x^n|x^{n-1})p(x^{n-1})\ dx^{n-1}$$

We introduced the **transition densities**

$$p(x^n|x^{n-1})$$

Meaning of the transition densities

Stochastic model:

$$x^n = f(x^{n-1}) + \beta^{n-1}$$

Transition density:

$$p(x^n|x^{n-1}) \propto p(\beta^{n-1})$$

So, draw a sample from the model error pdf, and use that in the stochastic model equations.
For a deterministic model this pdf is a delta function centered around the the deterministic forward step.
For a Gaussian model error we find:

$$p(x^n|x^{n-1}) = N\left(f(x^{n-1}), Q\right)$$

Bayes Theorem and the proposal density

Bayes Theorem now becomes:

$$
\begin{aligned}
p(x^n|y^n) &= \frac{p(y^n|x^n)p(x^n)}{p(y)} \\
&= \frac{p(y^n|x^n)}{p(y)} \int p(x^n|x^{n-1})p(x^{n-1})\, dx^{n-1}
\end{aligned}
$$

Multiply and divide this expression by a proposal transition density *q*:

$$
p(x^n|y^n) = \frac{p(y^n|x^n)}{p(y)} \int \frac{p(x^n|x^{n-1})}{q(x^x|x^{n-1}, y^n)} q(x^n|x^{n-1}, y^n)p(x^{n-1})\, dx^{n-1}
$$

The magic: the proposal density

We found:

$$p(x^n|y^n) = \frac{p(y^n|x^n)}{p(y)} \int \frac{p(x^n|x^{n-1})}{q(x^x|x^{n-1}, y^n)} q(x^n|x^{n-1}, y^n) p(x^{n-1})\, dx^{n-1}$$

Note that the transition pdf *q* can be conditioned on the future observation y^n.

The trick will be to draw samples from transition density *q* instead of from transition density *p.*

How to use this in practice?

Start with the particle description of the conditional pdf at *n-1* (assuming equal weight particles):

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

Leading to:

$$p(x^n|y^n) = \frac{p(y^n|x^n)}{p(y)}\frac{1}{N}\sum_{i=1}^{N}\frac{p(x^n|x_i^{n-1})}{q(x^n|x_i^{n-1}, y^n)}q(x^x|x_i^{n-1}, y^n)$$

Practice II

- The standard Particle Filter propagates the original model by drawing from $p(x^n|x^{n-1})$.
- Now we draw from $q(x^n|x^{n-1},y^n)$, *so we propagate the state using a different model.*
- This model can be anything, e.g.

$$x^n = g(x^{n-1}, y^n) + \hat{\beta}^n$$

Examples proposal transition density

The proposal transition density is related to a proposed model.
In theory, this can be any model!

For instance, add a relaxation term and change random forcing:

$$x^n = f(x^{n-1}) + \hat{\beta}^{n-1} + K\left(y^n - H(x^{n-1})\right)$$

Or, run a 4D-Var on each particle (implicit particle filter).
This is a special 4D-Var:
- initial condition is fixed
- model error essential
- needs extra random forcing

Or use the EnKF as proposal density.

Practice III

For each particle at time n-1 draw a sample from the proposal transition density *q*, to find:

$$p(x^n|y^n) = \frac{1}{N}\sum_{i=1}^{N} \frac{p(y^n|x_i^n)}{p(y)} \frac{p(x_i^n|x_i^{n-1})}{q(x_i^n|x_i^{n-1}, y^n)} \delta(x^n - x_i^n)$$

Which can be rewritten as:

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

with weights

$$w_i = \frac{p(y^n|x_i^n)}{p(y^n)} \frac{p(x_i^n|x_i^{n-1})}{q(x_i^n|x_i^{n-1}, y^n)}$$

Likelihood weight

Proposal weight

How to calculate *p/q*?

Let's assume that the original model has Gaussian distributed model errors:

$$p(x^n|x^{n-1}) = N\left(f(x^{n-1}), Q\right)$$

To calculate the value of this term realise it is the probability of moving from x_i^{n-1} *to* x_i^n. Since x_i^n and x_i^{n-1} are known from the proposed model we can calculate directly:

$$p(x_i^n|x_i^{n-1}) \propto exp\left[-\frac{1}{2}\left(x_i^n - f(x_i^{n-1})\right)^T Q^{-1}\left(x_i^n - f(x_i^{n-1})\right)\right]$$

Example calculation of *p*

- Assume the proposed model is

$$x^n = f(x^{n-1}) + \hat{\beta}^{n-1} + K\left(y^n - H(x^{n-1})\right)$$

- Then we find

$$p(x_i^n | x_i^{n-1}) \propto \exp\left[-\frac{1}{2}\left(K(y^n - H(x_i^{n-1})) + \hat{\beta}^n\right)^T Q^{-1}\left(K(y^n - H(x_i^{n-1})) + \hat{\beta}^n\right)\right]$$

- We know all the terms, so this can be calculated

And the *q* term…

- The deterministic part of the proposed model is:

$$x^n = f(x^{n-1}) + \quad K\left(y^n - H(x^{n-1})\right)$$

- So the probability becomes

$$q(x_i^n | x_i^{n-1}, y^n) \propto exp\left[-\frac{1}{2}\hat{\beta}_i^{n-1T}\hat{Q}^{-1}\hat{\beta}_i^{n-1}\right]$$

- We did draw the stochastic terms, so we know what they are, so this term can be calculated too.

The weights

- We can calculate *p/q* and we can calculate the likelihood so we can calculate the weights:

$$w_i = \frac{p(y^n|x_i^n)}{p(y^n)} \frac{p(x_i^n|x_i^{n-1})}{q(x_i^n|x_i^{n-1}, y^n)}$$

Example: EnKF as proposal

EnKF update:

$$x_i^n = x_i^* + K^e \left(y^n + \epsilon_i - H(x_i^*)\right)$$

Use model equation:

$$x_i^n = f(x_i^{n-1}) + \beta_i^n + + K^e \left(y^n + \epsilon_i - H\left(\left(f(x_i^{n-1}) + \beta_i^n\right)\right)\right)$$

Regroup terms:

$$x_i^n = f(x_i^{n-1}) + K^e \left(y^n - H\left(f(x_i^{n-1})\right)\right) + (1 - K^e H)\beta_i^n + K^e \epsilon_i$$

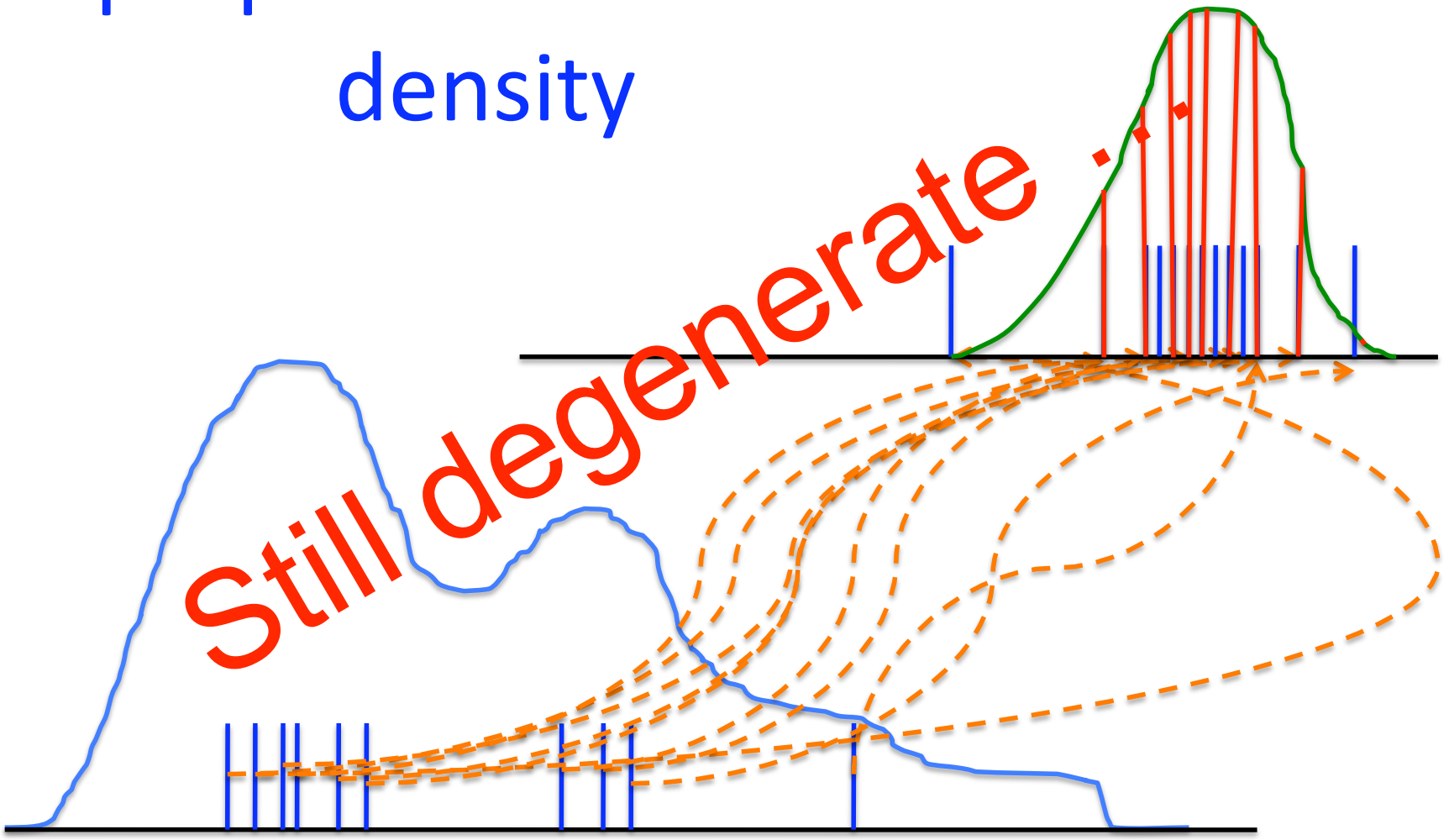
Leading to:

$$x_i^n = \qquad g(x_i^{n-1}, y^n) \qquad + \qquad \hat{\beta}_i^n$$

Algorithm

- Generate initial set of particles
- Run proposed model conditioned on next observation
- Accumulate proposal density weights p/q
- Calculate likelihood weights
- Calculate full weights and resample
- Note, the original model is never used directly.

Particle filter with proposal transition density



Equivalent-weights I

1. We know:

$$w_i = \frac{p(y^n|x_i^n)}{p(y^n)} \frac{p(x_i^n|x_i^{n-1})}{q(x_i^n|x_i^{n-1}, y^n)}$$

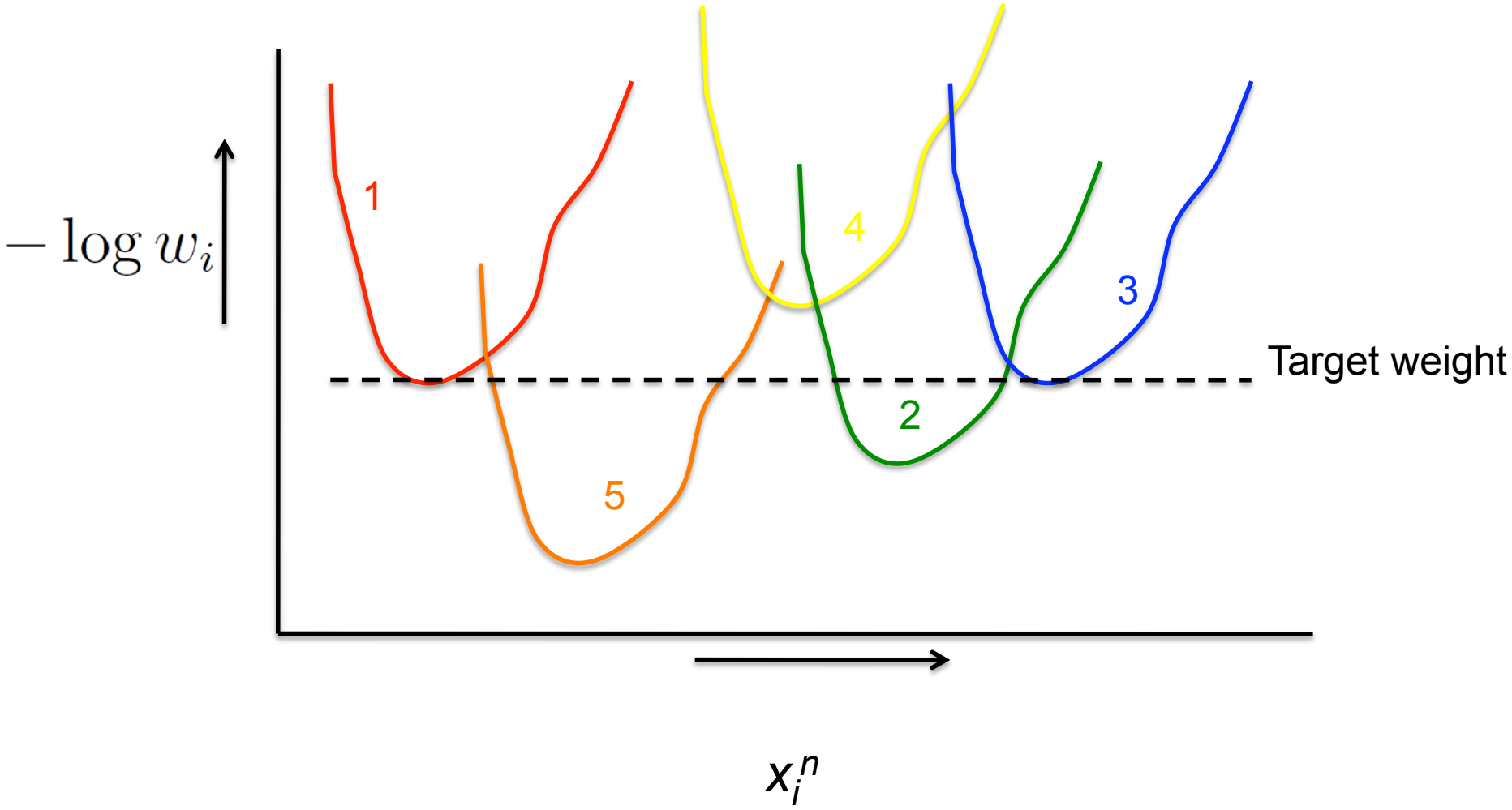
2. Write down expression for each weight ignoring *q* for now:

$$w_i \propto w_i^{rest} \exp \left[-\frac{1}{2} \left(x_i^n - f(x_i^{n-1}) \right)^T Q^{-1} \left(x_i^n - f(x_i^{n-1}) \right) - \frac{1}{2} (y^n - H(x_i^n))^T R^{-1} (y^n - H(x_i^n)) \right]$$

3. When H is linear this is a quadratic function in x_i^n for each particle. Otherwise linearize.

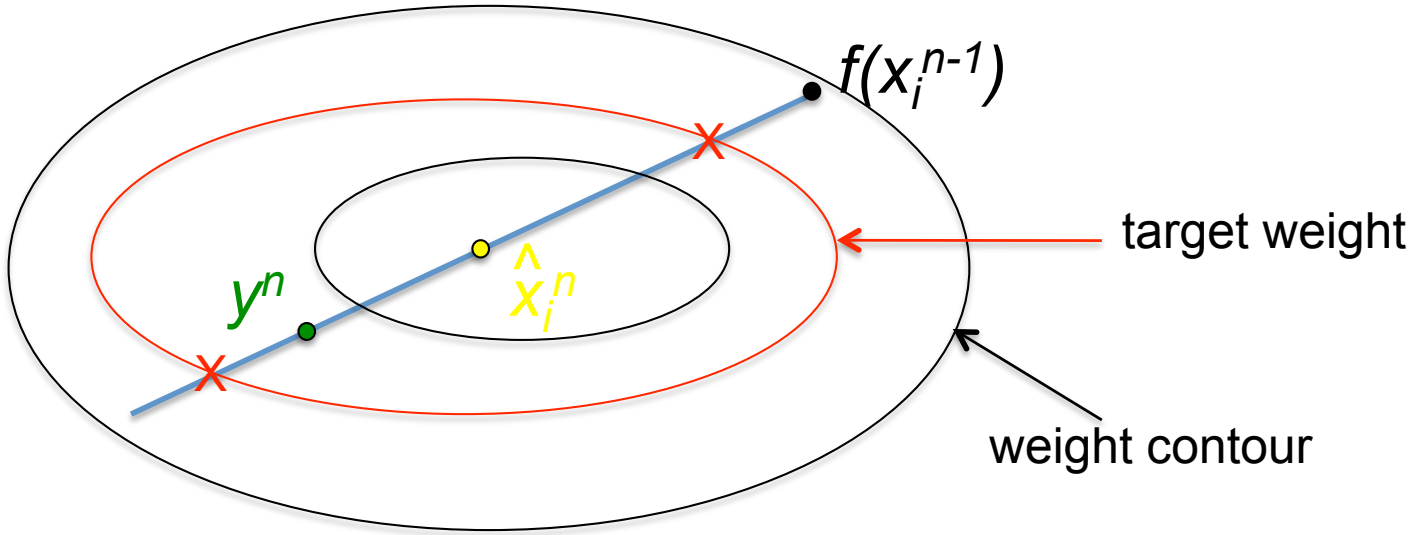
Equivalent-weights II

4. Determine a target weight



Equivalent-weights III

5. Determine corresponding model states, infinite number of solutions.



Determine α at crossing of line with target weight contour in:

$$x_i^n = f(x_i^{n-1}) + \alpha K \left(y^n - H f(x_i^{n-1}) \right)$$

with $$K = Q H^T (H Q H^T + R)^{-1}$$

Equivalent-Weights IV

- So, by construction 80% of the particles have equal weight!
- Hence PF not degenerate by construction.
- However, we still need a stochastic move. (Why?)

Almost equal weights IV

6. The previous is the deterministic part of the proposal density.

The stochastic part of *q* should not be Gaussian because we divide by *q*, so an unlikely value for the random vector $\hat{\beta}_i^{n-1}$ will result in a huge weight:

$$w_i = \frac{p(y^n|x_i^n)}{p(y^n)} \frac{p(x_i^n|x_i^{n-1})}{q(x_i^n|x_i^{n-1}, y^n)}$$

A uniform density will leave the weights unchanged, but has limited support.

Hence we choose $\hat{\beta}_i^{n-1}$ from a mixture density:

$$p(\hat{\beta}_i^{n-1}) \propto (1-a)U[-b,b] + aN(0,\hat{Q})$$

with
a,*b*,*Q* small

Almost equal weights V

The full scheme is now:

- Use modified model up to last time step
- Set target weight (e.g. 80%)
- Calculate deterministic moves:

$$x_i^n = f(x_i^{n-1}) + \alpha K \left(y^n - H f(x_i^{n-1}) \right)$$

- Determine stochastic move

$$p(\hat{\beta}_i^{n-1}) \propto (1-a)U[-b,b] + aN(0,\hat{Q})$$

- Calculate new weights and resample ‘lost’ particles

Parameter Estimation I

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

with

$$y = H(\theta) + \epsilon$$

Again, no inversion but a direct point-wise multiplication.
But how to do this?

Parameter estimation II

Typically we have a set of observations over time, so we want to know:

$$p(\theta|y^{1:n})$$

The connection between the parameters and the observations is via the model:

$$x_i^n = f(x_i^{n-1}, \theta) + \beta_i^n$$

Parameter estimation III

The likelihood can be written as:

$$p(y^{1:n}|\theta) = \int p(y^{1:n}, x^{0:n}|\theta)\, dx^{0:n}$$

Using the conditional pdf we find:

$$p(y^{1:n}|\theta) = \int p(y^{1:n}|x^{0:n}, \theta)\, p(x^{0:n}|\theta)\, dx^{0:n}$$

Exploring that observations depend directly on the state:

$$p(y^{1:n}|\theta) = \int p(y^{1:n}|x^{0:n})\, p(x^{0:n}|\theta)\, dx^{0:n}$$

Parameter estimation IV

Explore Bayes theorem:

$$p(\theta|y^{1:n}) = \frac{\int p(y^{1:n}|x^{0:n})\ p(x^{0:n}|\theta)\ dx^{0:n}}{p(y)} p(\theta)$$

So the standard procedure is:

1 Draw a θ from its prior pdf
2 Draw an initial model state
3 Run the model, drawing stochastic terms at each time step
4 Calculate the likelihood for this run
5 Repeat 2-4 N times and add the likelihoods for this θ
6 Repeat 1-5 M times to find weighted ensemble of θ's

Parameter estimation V

- The particle filters for the model runs can be made more efficient using a proposal density, like the Equivalent-weights Particle Filter.

How to test the accuracy of DA scheme's?

- If the DA problem is linear one can use that the costfunction is a chi-squared variable with M, the number of independent observations, as degrees of freedom. So the value of the costfucntion should lie in $$M \pm \sqrt{M}$$
- Other consistency tests for linear DA test relations between the statistics of the forecast and analysis innovations, and the covariances. From linear theory we find relations like:

$$E\left[(x_a - y)(x_a - y)^T\right] = R - HP_aH^T$$

$$E\left[(x_a - y)(x_b - y)^T\right] = R$$

that can be checked.

And nonlinear DA scheme's?

- Compare the ensemble spread with the RMSE in idealised experiments
- One can use other measures, like a rank histogram or talagrand diagram:

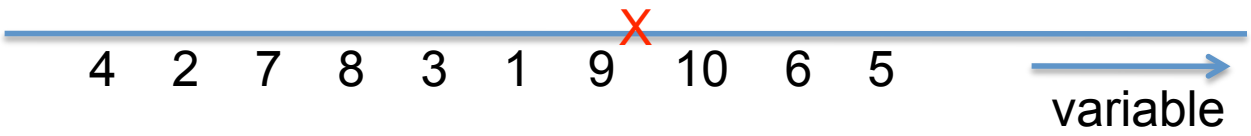
  - Take one observation *y.*

  - Add observation noise to model equivalents $H(x_i)$ and sort them in magnitude.

  - Rank the position of the observation in this sorted ensemble

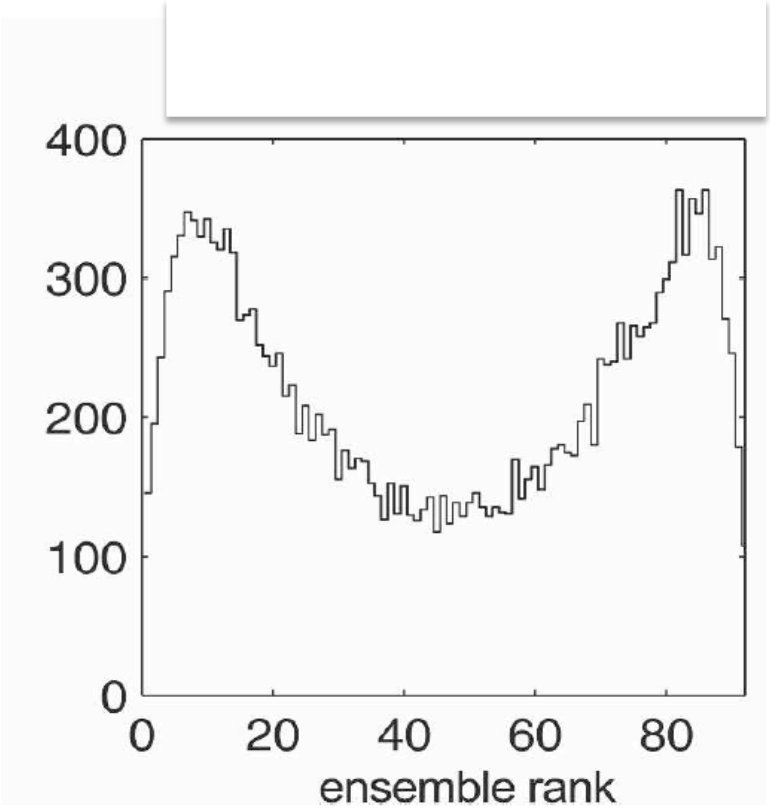
  - Do this every several assimilation cycles (or over similar independent observations)

  - Produce a histogram of the rankings.

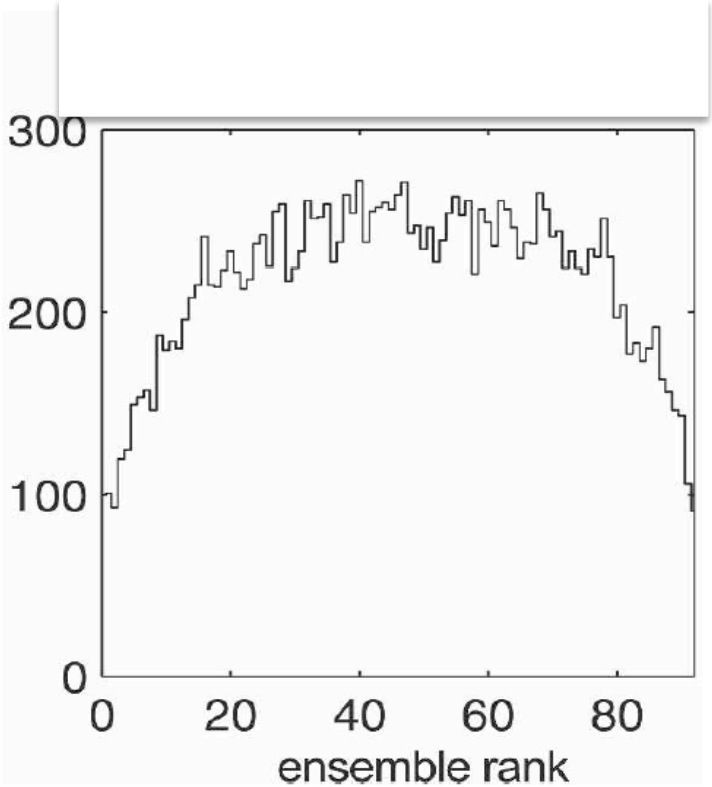


Examples of rank histograms

Ideally the rank histogram is flat: the observation is
Indistinguishable from any ensemble member, so any
Ensemble member could be responsible for the observations



Under-dispersive ensemble



Over-dispersive ensemble