Introduction to Particle Filters

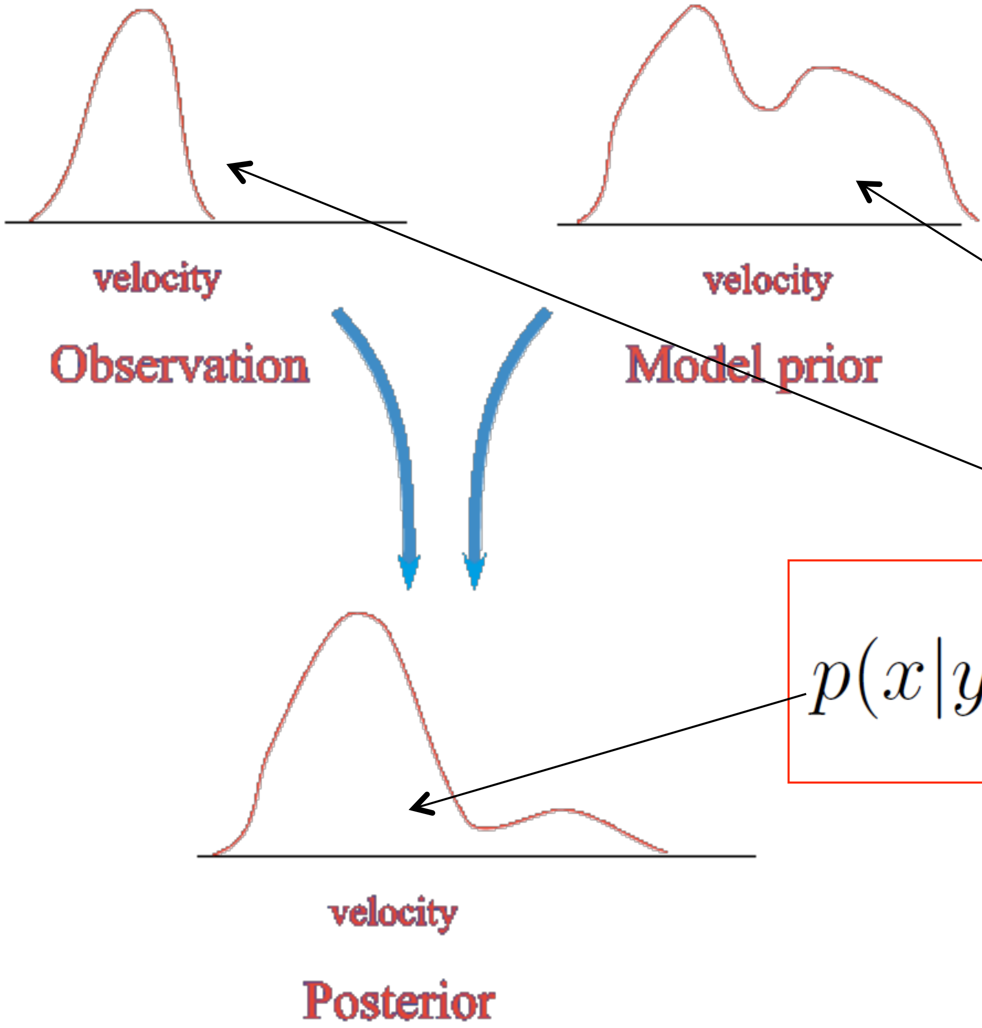
Peter Jan van Leeuwen and Mel Ades

Data-Assimilation Research Centre DARC

University of Reading

Adjoint Workshop 2011

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.

How is this used today in geosciences?

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

4DVar:

- smoother
- Gaussian pdf for initial state, observations (and model errors)
- allows for nonlinear observation operators
- solves for posterior mode.
- needs good error covariance of initial state (B matrix)
- ‘no’ posterior error covariances

How is this used today in geosciences?

Representer method (PSAS):

- solves for posterior mode in observation space

(Ensemble) Kalman filter:

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

Combinations of these: hybrid methods (!!!)

Non-linear Data Assimilation

- Metropolis-Hastings
- Langevin sampling
- Hybrid Monte-Carlo
- Particle Filters/Smoothers

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

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

Not very efficient !

A simple resampling scheme

1. Put all weights on the unit interval [0,1]:

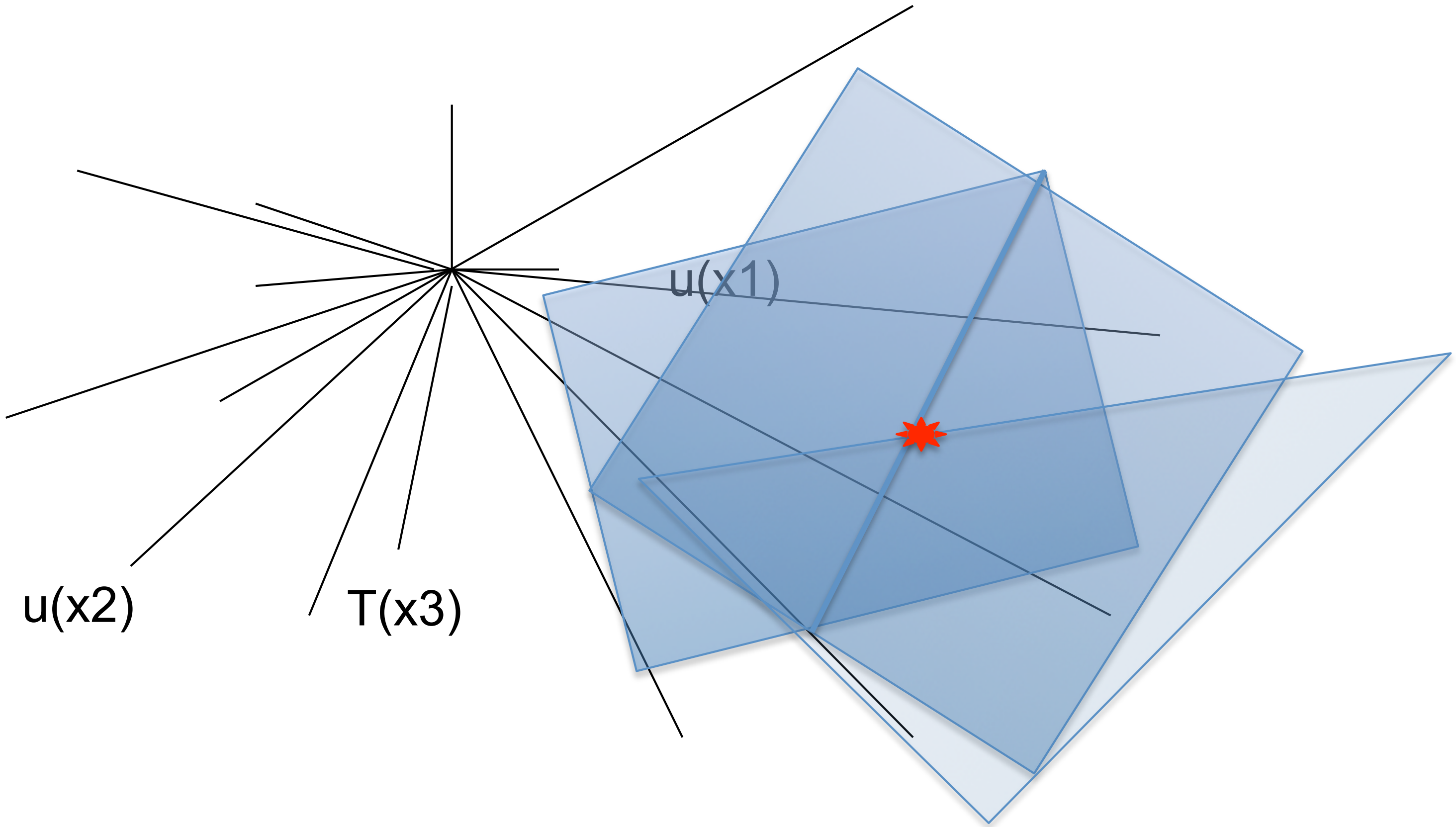
2. Draw a random number from U[0,1/N] (= U[1,1/10] in this case). Put it on the unit interval: this is the first resampled particle.

3. Add 1/N : this is the second resampled particle. Etc.

In this example we choose old particle 1 three times, old particle 2 two times, old particle 3 two times etc.

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.

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

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

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

What is the proposal transition density?

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

For instance, add a nudging 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. This is a special 4D-Var:

- initial condition is fixed
- model error essential
- needs extra random forcing (perhaps perturbing obs?)

How to calculate *p/q*?

Let's assume

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

Since x_i^n and x_i^{n-1} are known from the proposed model we can calculate directly:

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

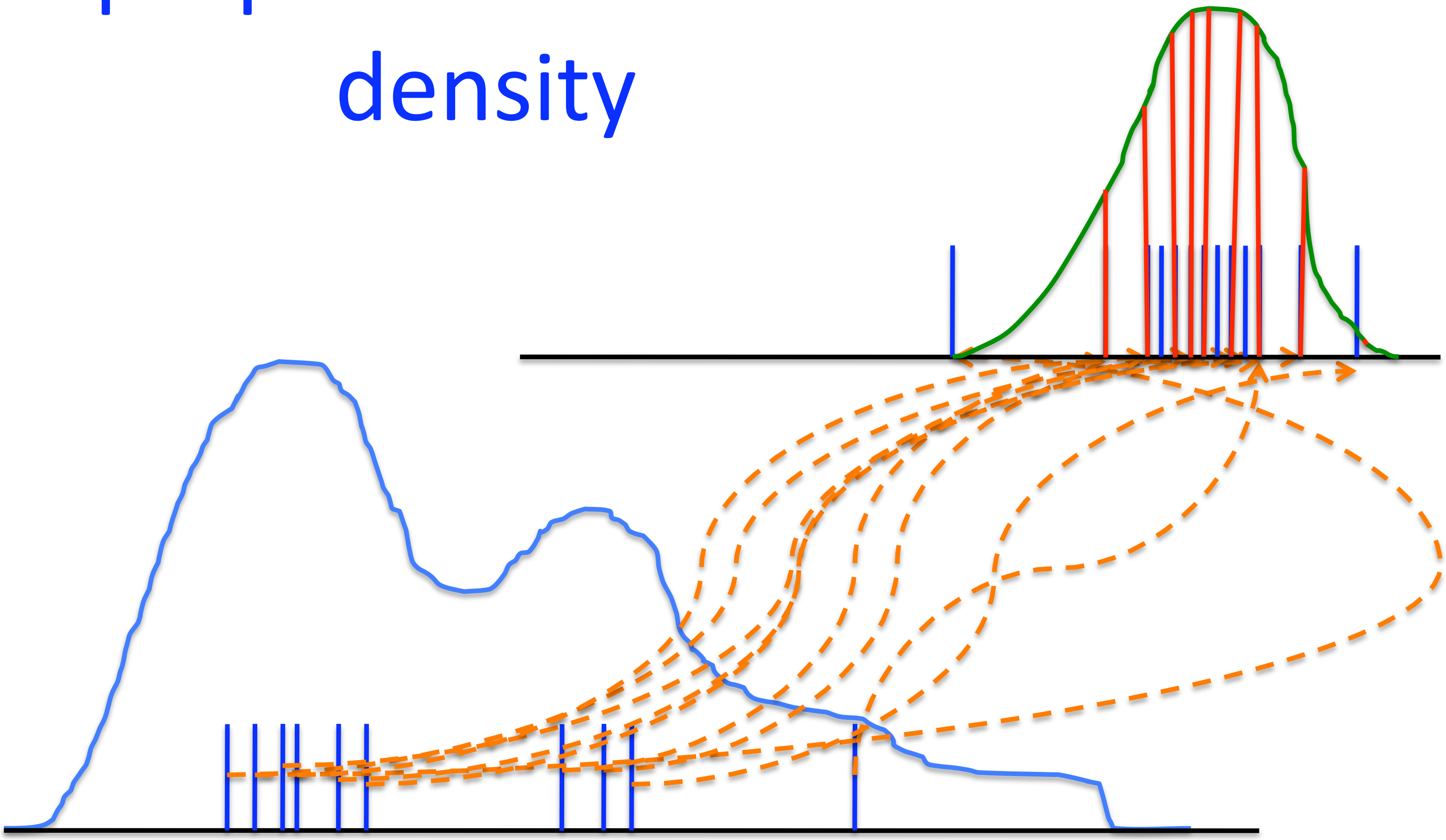
Similarly, for the proposal transition density:

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

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



However: degeneracy

For large-scale problems with lots of observations this method is still degenerate:

Only a few particles get high weights; the other weights are negligibly small.

Recent ideas

- ‘Optimal’ proposal transition density: is not optimal. This method is explored by Chorin and Tu (2009), and Miller (the ‘Berkeley group’).
- Other particle filters use interpolation (Anderson, 2010; Majda and Harlim, 2011), can give rise to balance issues. Proposal not used (yet).
- Briggs (2011) explores a spatial marginal smoother at analysis time. Needs copula for joint pdf, chosen as an elliptical density.
- Can we do better?

Almost equal weights I

1. We know:

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

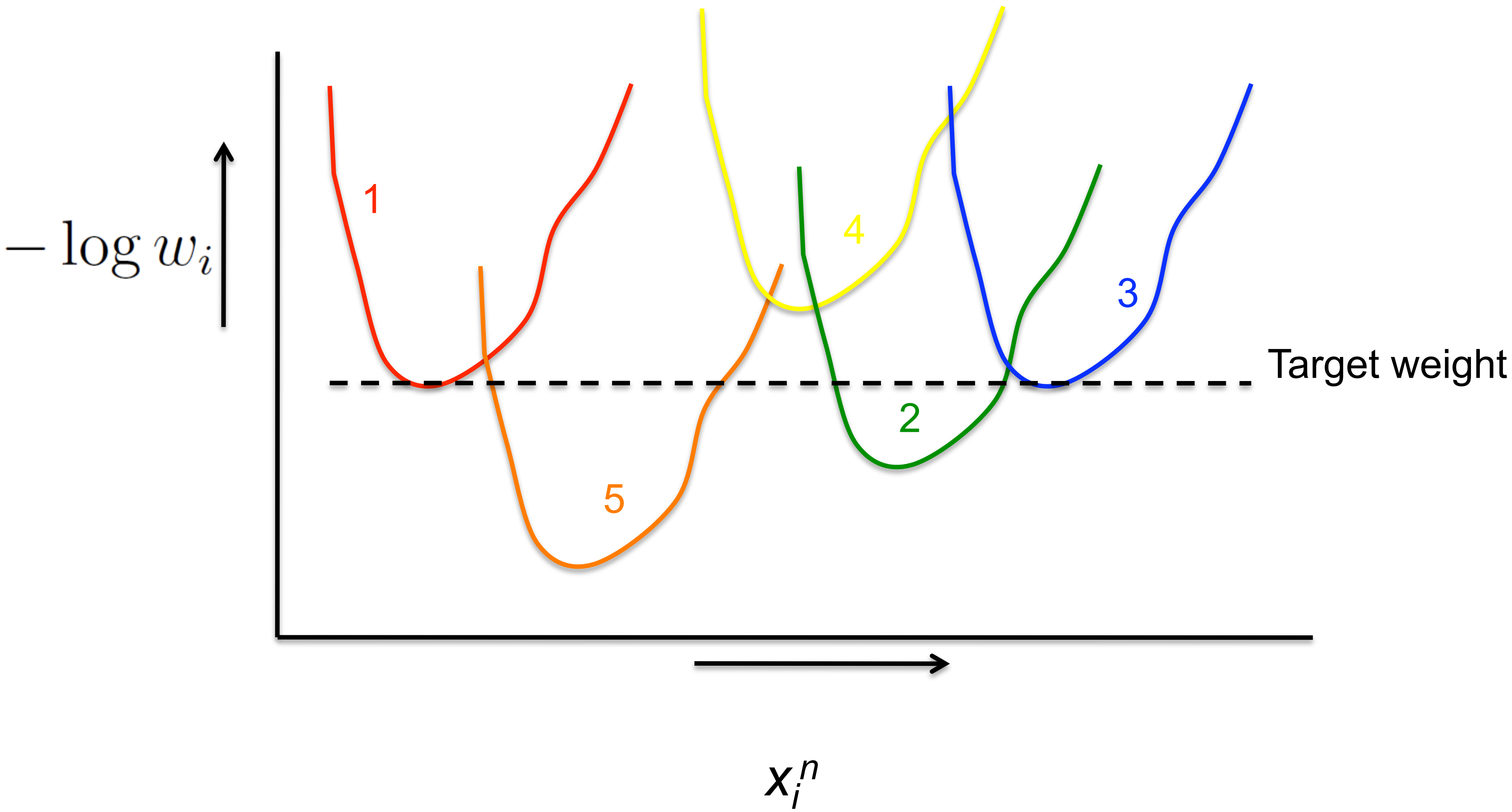
2. Write down expression for each weight ignoring *q* for now:

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

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

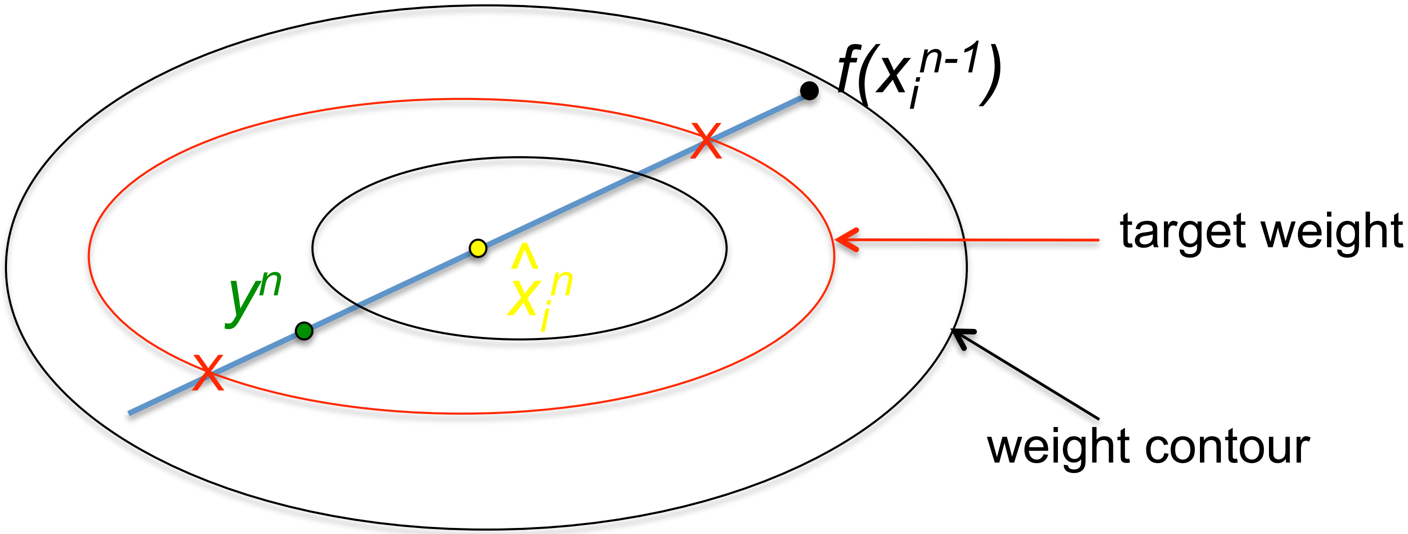
Almost Equal weights II

4. Determine a target weight



Almost equal 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
$$K = QH^T(HQH^T + R)^{-1}$$

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

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

- Calculate new weights and resample ‘lost’ particles

Conclusions

- Particle filters do not need state covariances.

- Observations do not have to be perturbed.

- Degeneracy is related to number of observations, not to size of the state space.

- Proposal density allows enormous freedom

- Almost-equal-weight scheme is scalable => high-dimensional problems.

- Other efficient schemes are being derived.

We need more people !

- In Reading only we expect to have 10 new PDRA positions available in the this year
- We also have PhD vacancies
- And we still have room in the

Data Assimilation and Inverse Methods

in Geosciences MSc program

Gaussian-peak weight scheme

The weights are given by:

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

and our goal is to make these weights almost equal by choosing a good proposal density, and a natural limit for *N* --> infinity.

We start by writing

$$\begin{aligned} -2\log\left(p(y|x_i^n)p(x_i^n|x_i^{n-1})\right) &= \left(x_i^n - f(x_i^{n-1})\right)^T Q^{-1}\left(x_i^n - f(x_i^{n-1})\right) \\ &\quad + (y^n - H(x_i^n))^T R^{-1}(y^n - H(x_i^n)) \end{aligned}$$

Which can be rewritten as (completing the square on x_i^n):

$$-2\log\left(p(y|x_i^n)p(x_i^n|x_i^{n-1})\right) \propto (x_i^n - m_i)^T P^{-1} (x_i^n - m_i) + \phi_i$$

With the constant

$$\phi_i = \left(y - Hf(x_i^{n-1})\right)(HQH^T + R)^{-1}\left(y - Hf(x_i^{n-1})\right)$$

Write the proposal transition density as:

$$-2\log\left(q(x_i^n|x_i^{n-1}, y^n)\right) \propto (x_i^n - m_i)^T \hat{Q}_i^{-1} (x_i^n - m_i) + \phi_i$$

So we draw samples from this Gaussian density.
The normalisation of *q* leads to the relation

$$|\hat{Q}_i|^{1/2} \propto \exp[-\phi_i]$$

To control the weights write:

$$-2\log\left(p(y|x_i^n)p(x_i^n|x_i^{n-1})\right) \propto (x_i^n - m_i)^T\, P^{-1}\, (x_i^n - m_i) + \phi_i$$

as

This is q

$$-2\log\left(p(y|x_i^n)p(x_i^n|x_i^{n-1})\right) \quad \propto \quad (x_i^n - m_i)^T\, \hat{Q}_i^{-1}\, (x_i^n - m_i) + \phi_i$$
$$+ (x_i^n - m_i)^T\, S_i^{-1}\, (x_i^n - m_i)$$

To find weights:

$$w_i \propto \exp\left[-\frac{1}{2}\, (x_i^n - m_i)^T\, S_i^{-1}\, (x_i^n - m_i)\right]$$

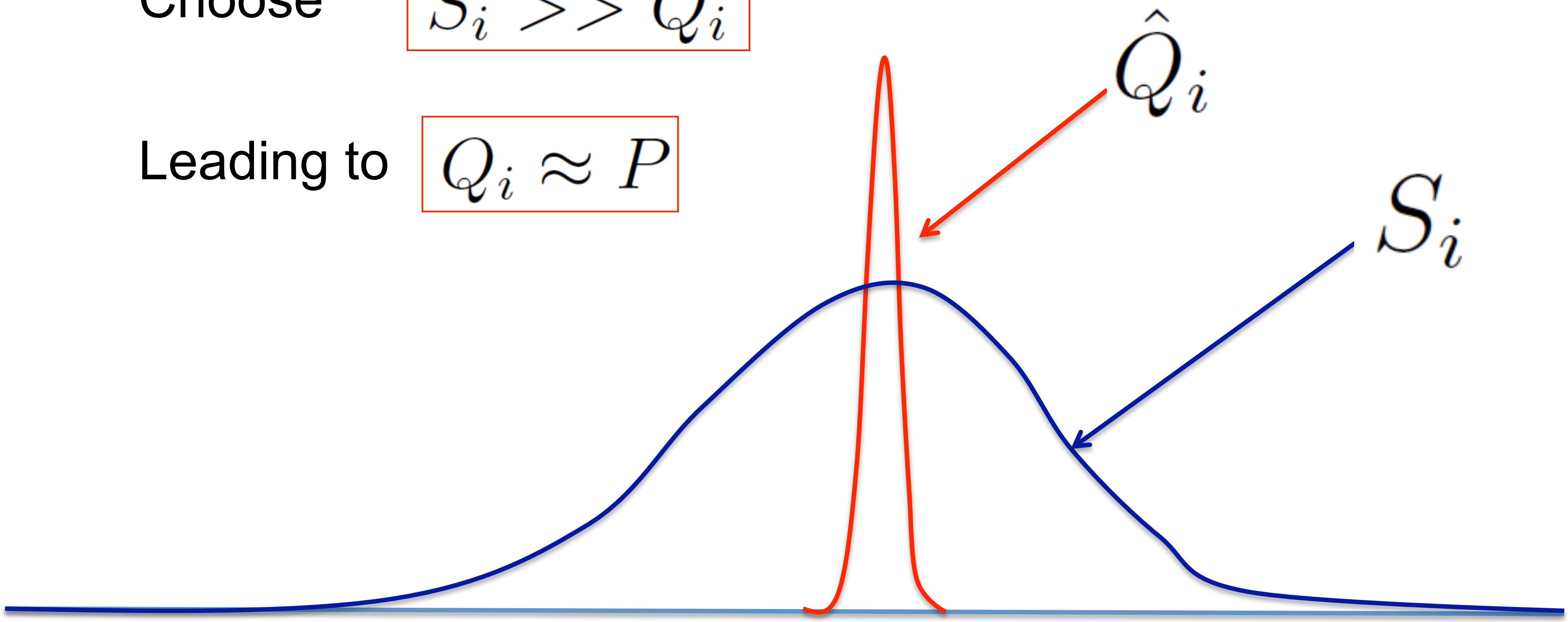
And a relation between the covariances:

$$P = \hat{Q}_i(\hat{Q}_i + S_i)^{-1} S_i$$

The final idea…

Choose $S_i >> \hat{Q}_i$

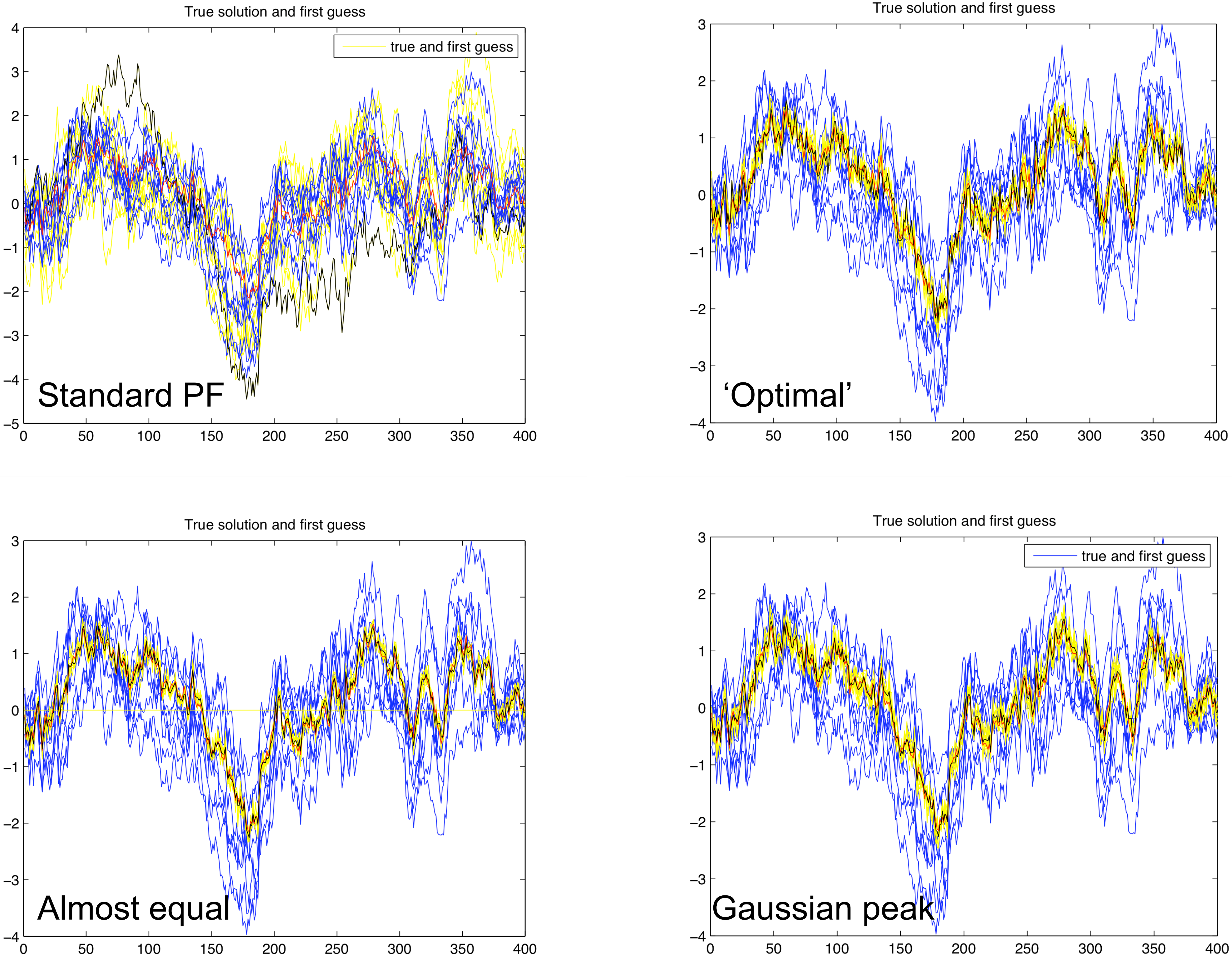
Leading to $Q_i \approx P$



So, the idea is to draw from $N(0,\hat{Q}_i)$ and the weights come out as drawn from $N(0,S_i)$.

Example: one step, with equal weight ensemble at time *n-1*

- 400 dimensional system, Q = 0.5
- 200 observations, sigma = 0.1
- 10 particles
- Four Particle filters:
  - Standard PF
  - ‘Optimal’ proposal density
  - Almost equal weight scheme
  - Gaussian-peak weight scheme



True solution and first guess
true and first guess
Standard PF
True solution and first guess
'Optimal'
True solution and first guess
Almost equal
True solution and first guess
true and first guess
Gaussian peak

Performance measures

Effective ensemble size

$$N_{eff} = \frac{1}{\sum_{i=1}^{N} w_i^2}$$

Filter:	Squared difference from truth:	Effective ensemble size:
PF standard error	1.3931	1
PF-'optimal' error	0.10889	1
PF-Almost equal error	0.073509	8.8
PF-Gaussian Peak error	0.083328	9.4

'Optimal' proposal density has no pdf information, new schemes performing well.