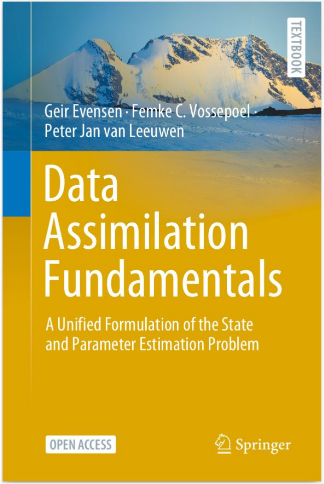
Nonlinear data assimilation:

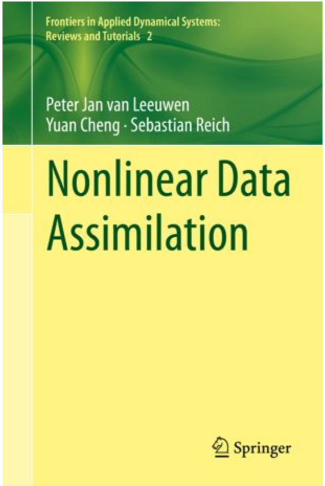
Particle filters from a Bayesian perspective

Femke C. Vossepoel, based on the book of Geir Evensen, Femke C. Vossepoel and Peter Jan van Leeuwen

Available from `https://github.com/geirev/Data-Assimilation-Fundamentals.git`

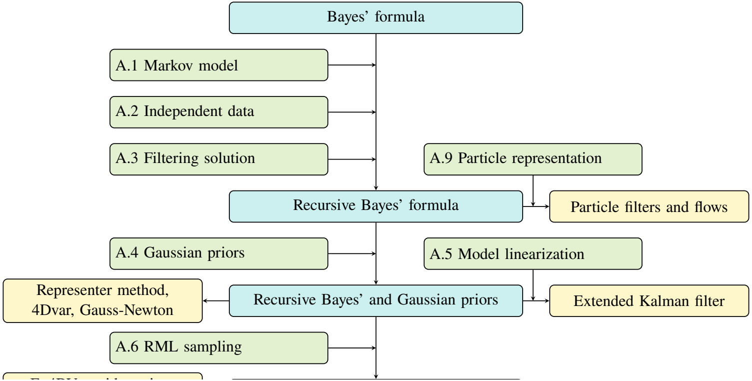
Full details in:



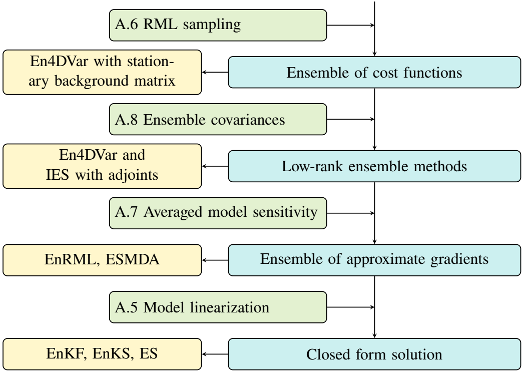


Available from: `https://github.com/geirev/Data-Assimilation-Fundamentals`

Overview of approximations and methods



Overview of approximations and methods



Particle filters for nonlinear data assimilation

Approximation 9 (Particle representation of the pdfs)

It is possible to approximate a probability density function by a finite ensemble of N model states (or particles) as

$$f(\mathbf{z}) \approx \sum_{j=1}^{N} \frac{1}{N} \delta(\mathbf{z} - \mathbf{z}_j), \tag{1}$$

where $\delta(\cdot)$ denotes the Dirac-delta function.

Importance sampling Monte Carlo

We can now use what we derived about the probability of $\mathcal{Z}$ given the data, and use Monte Carlo samples to approximate the probability. This involves:

- generate N pseudo-random realisations z_j from $f(\mathbf{z}|\mathbf{d})$ with $j = 1, \ldots, N$.
- evaluate for each realisation the outcome of the forward model and compute the arithmetic average of the results.

Importance sampling Monte Carlo

So, we can approximate the distribution $f(\mathbf{z}|\mathbf{d})$ by:

$$f(\mathbf{z}|\mathbf{d}) = \sum_{j=1}^{N} w_j \delta(\mathbf{z} - \mathbf{z}_j), \tag{2}$$

where $\delta_{\mathbf{z}_j}$ is a Dirac delta, and the so-called likelihood weights w_j given by

$$w_j = \frac{f(\mathbf{d}|\mathbf{z}_j)}{f(\mathbf{d})} = \frac{f(\mathbf{d}|\mathbf{z}_j)}{\sum_{j=1}^{N} f(\mathbf{d}|\mathbf{z}_j))}. \tag{3}$$

denominator: standard self normalization to ensure the weights add up to one, $f(\mathbf{d}) = \int f(\mathbf{d}|\mathbf{z}) f(\mathbf{z})\, d\mathbf{z} \approx \sum_{j=1}^{N} f(\mathbf{d}|\mathbf{z}_j)$.

Basic Importance Sampling

In practice, sequential scheme

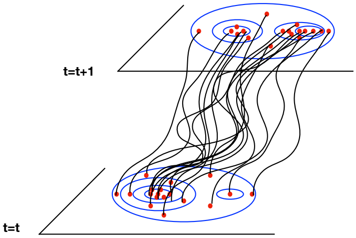
1. Sample N particles $\mathbf{z}_j$ from initial model probability density $f(\mathbf{z}(t=0))$
2. Integrate all particles forward to measurement time: $f(\mathbf{z}_t|\mathbf{z}_{t-1,j})$ for time t, for each j.
3. Calculate weights w_j, for each sample, and normalize
4. Increase t by 1, and repeat steps 2 and 3

Note: particles are not modified, only their weights.

Pseudo algorithm

Input: $\mathbf{Z} \in \Re^{n \times N}$ ▷ Initial model state-vector ensemble
Input: $\mathbf{d} \in \Re^{m}$ ▷ Measurements vector
for $j = 1, N$ **do** ▷ Log Particle weights
    $\tilde{w}_j = \ln f(\mathbf{d}|\mathbf{z}_j)$
end for
$\tilde{w}_{\max} = \max_j(\tilde{w}_j)$ ▷ Max of log weights
for $j = 1, N$ **do** ▷ Unnormalized Particle weights
    $\tilde{w}_j = \exp(\tilde{w}_{\max} - \tilde{w}_j)$
end for
for $j = 1, N$ **do** ▷ Particle weights
    $w_j = \frac{\tilde{w}_j}{\sum_i \tilde{w}_i}$
end for
call Resample$(\mathbf{w}, \mathbf{I})$ ▷ Resampling step
for $j = 1, N$ **do** ▷ Resample ensemble
    $\mathbf{z}_j = \mathbf{z}_{I_j}$
end for

Concept of particle filtering



Ensemble of N realisations (particles) to estimate pdf evolution

Posterior is proportional to prior times likelihood

Prior:

$$f(\mathbf{z}) = \Sigma_{j=1}^{N} \frac{1}{N} \delta(\mathbf{z} - \mathbf{z}_j)$$

Likelihood:

$f(\mathbf{d}|\mathbf{z}) = \frac{1}{\sigma\sqrt{2\pi}} \exp^{-\frac{1}{2}(\frac{\mathbf{d}-\mathbf{z}}{\sigma})^2}$

(Gaussian, can also be Lorentz function)

Posterior:

$f(\mathbf{z}) \propto f(\mathbf{d}|\mathbf{z}) f(\mathbf{z})$

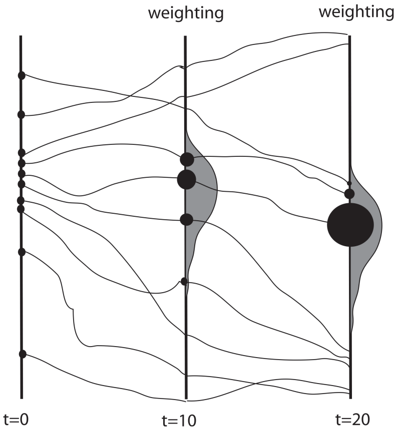
Effective ensemble size

$$N_{\text{eff}} = \frac{1}{\sum_{i=1}^{N} w_i^2}, \tag{4}$$

Effective ensemble size: measure for particle divergence.
w_i: normalized weights
resampling: typically when $N_{\text{eff}} \leq 0.8N$

Ensemble degeneracy in particle filtering

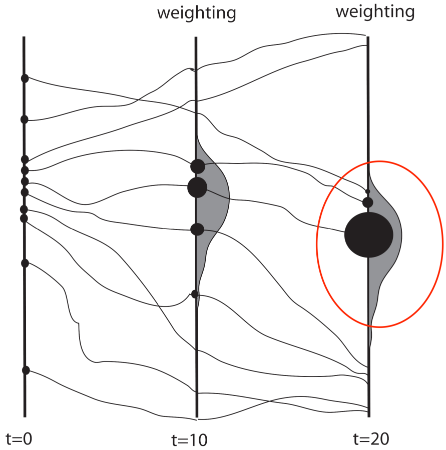
Size of circles indicates weight



Samples that are closest to observations obtain largest weight.

Degeneracy

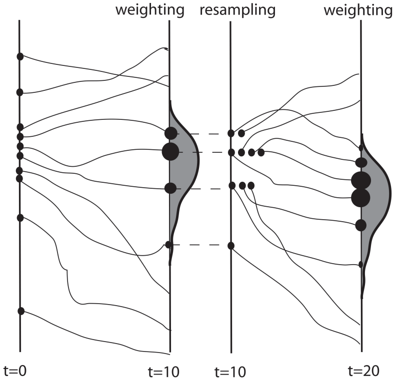
Size of circles indicates weight



Some samples move very far from the observations and obtain a low weight. This means that effectively, there are only few samples left!

Importance Resampling

Size of circles indicates weight



Samples that are closest to observations obtain largest weight and are being duplicated. Low weight samples are removed from the ensemble.

Resampling in practice

1. Sample N particles $\mathbf{z}_j$ from initial model probability density $f(\mathbf{z}(t=0))$
2. Integrate all particles forward to measurement time: $f(\mathbf{z}_t|\mathbf{z}_{t-1,j})$ for time t, for each j.
3. Calculate weights w_j, for each sample, and normalize
4. Resample, while choosing for each particle a weight equal to $1/N$
5. Repeat steps 2, 3, 4 until all observations have been processed

Note: some particles are lost: analysis not smooth anymore!

Methods of resampling

For step 4, the resampling, there are three common approaches:

- **probabilistic resampling:** sample from distribution of ensemble members. Straightforward, but introducing sampling noise.
- **residual resampling:** multiply weights by N, take integer part of resulting weight (call this n) and take n copies of these particles. Subtract all integer parts of Nw_j, and to complete the set of N samples, sample randomly from the resulting distribution. Lower sampling noise than probabilistic resampling.
- **stochastic universal resampling:** Place all weights after each other between $[0, 1]$. Draw a random number between $[0, 1/N]$ and start with a sample at this place. Then sample again at each $1/N$ distance from the previous sample. Samples with a large weight will have a higher chance of being 'hit'. Lowest sampling noise of three options.

Stochastic Universal Resampling

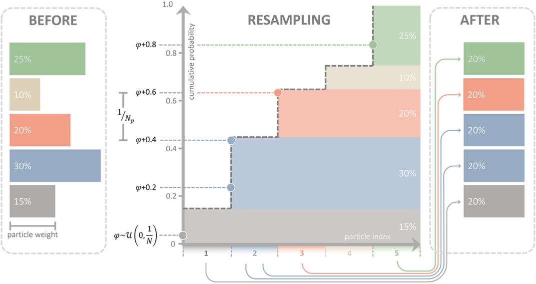


Figure 1. Schematic illustration of stochastic universal resampling for $\boldsymbol{N = 5}$. Starting with an ensemble of nonuniformly weighted particles (left), we construct a cumulative probability function (dashed gray line, center). After drawing a random offset $\boldsymbol{\varphi}$ from a uniform distribution $\boldsymbol{\mathcal{U}}(\boldsymbol{min}=\mathbf{0}, \boldsymbol{max}=\mathbf{1}/\boldsymbol{N})$, we obtain resampled particle indices by sampling this function in additive increments of $\mathbf{1}/\boldsymbol{N}$. Finally, each particle slot inherits the variables of its respective resampled particle index ($\mathbf{1} \rightarrow \mathbf{1,2} \rightarrow \mathbf{2,3} \rightarrow \mathbf{2,4} \rightarrow \mathbf{3,5} \rightarrow \mathbf{5}$) and the weights are reset (right, equation 15).

Ramgraber et al. 2019, 10.1029/2019WR026777

What factors influence degeneracy?

- **Size of state vector**, see next slide on asymptotic limit of *Snyder et al, 2018*.
- **Number of independent observations**. Hypothetical example of one observation at 0.1σ, and another at 0.2σ from model, and a total of N_y observations with variance 1σ. Using a Gaussian likelihood, it can be shown that the weight of the second observation is $3 \cdot 10^{-7}$ times smaller than weight of first observation when number of observations is 1000. This becomes $0,22$ when N_y is 100. So the larger the number of independent observations, the higher the chance of degeneracy, even with inaccurate observations.
- **Proposal densities**. Instead of drawing samples from our prior, we can also draw samples from a density function that minimises degeneracy: the proposal density.

Condition for collapse - asymptotic limit of Snyder et al (2008)

$$log_{10} N_e = 0.05 N_x + 0.78$$

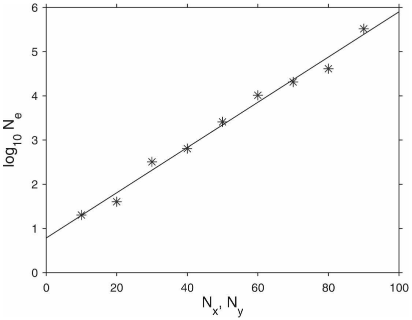


FIG. 2. The ensemble size N_e as a function of N_x (or N_y) required if the posterior mean estimated by the particle filter is to have average squared error less than the prior or observations, in the simple example considered in the text. Asterisks show the simulation results, averaged over 400 realizations. The best-fit line is given by $\log_{10} N_e = 0.05 N_x + 0.78$.

Snyder et al. 2008, 10.1175/2008MWR2529.1

Proposal densities

Instead of sampling from the prior, we sample form another pdf, the *proposal pdf* $q(\mathbf{z})$. Rewriting Bayes:

$$f(\mathbf{z}|\mathbf{d}) = \frac{f(\mathbf{d}|\mathbf{z})}{f(\mathbf{d})} f(\mathbf{z}) = \frac{f(\mathbf{d}|\mathbf{z})}{f(\mathbf{d})} \frac{f(\mathbf{z})}{q(\mathbf{z})} q(\mathbf{z}). \tag{5}$$

Using samples from $q(\mathbf{z})$ we can write

$$q(\mathbf{z}) = \sum_{i=1}^{N} \frac{1}{N} \delta(\mathbf{z} - \mathbf{z}_i), \tag{6}$$

Proposal density

Again, we write:

$$f(\mathbf{z}|\mathbf{d}) = \sum_{i=1}^{N} w_i \delta(\mathbf{z} - \mathbf{z}_i), \tag{7}$$

but now with weights

$$w_i = \frac{f(\mathbf{d}|\mathbf{z}_i)}{N \sum_{j=1}^{N} f(\mathbf{d}|\mathbf{z}_j)} \frac{f(\mathbf{z}_i)}{q(\mathbf{z}_i)}. \tag{8}$$

likelihood weights are multiplied by *proposal weights*.

Example of proposal density

Let's assume Gaussian additive model errors with mean zero and covariance $\mathbf{C}_{qq}$.
Transition density:

$$f(\mathbf{z}_k|\mathbf{z}_{k-1}) \propto \exp\left(-\frac{1}{2}\left(\mathbf{z}_k - \mathbf{m}(\mathbf{z}_{k-1})\right)^{\mathrm{T}}\mathbf{C}_{qq}^{-1}\left(\mathbf{z}_k - \mathbf{m}(\mathbf{z}_{k-1})\right)\right). \tag{9}$$

Now, we can rewrite the prior pdf as:

$$f(\mathbf{z}_k) = \int f(\mathbf{z}_k, \mathbf{z}_{k-1})\, d\mathbf{z}_{k-1} = \int f(\mathbf{z}_k|\mathbf{z}_{k-1}) f(\mathbf{z}_{k-1})\, d\mathbf{z}_{k-1}. \tag{10}$$

Invoke particle representation at time t_{k-1}, and prior becomes:

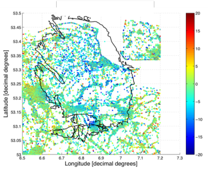
$$f(\mathbf{z}_k) = \int f(\mathbf{z}_k|\mathbf{z}_{k-1}) \frac{1}{N}\sum_{j=1}^{N} \delta(\mathbf{z}_{k-1} - \mathbf{z}_{j,k-1})\, d\mathbf{z}_{k-1} = \sum_{j=1}^{N} \frac{1}{N} f(\mathbf{z}_k|\mathbf{z}_{j,k-1}). \tag{11}$$

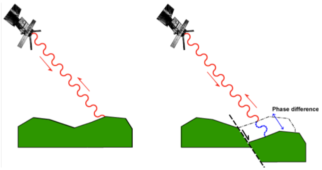
Proposal density based on observations

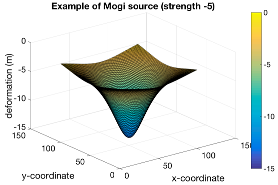
$$
\begin{aligned}
f\left(\mathbf{z}_k|\mathbf{d}_k\right) &= \frac{f\left(\mathbf{d}_k|\mathbf{z}_k\right)}{f\left(\mathbf{d}_k\right)} f\left(\mathbf{z}_k\right) \\
&= \frac{f\left(\mathbf{d}_k|\mathbf{z}_k\right)}{f\left(\mathbf{d}_k\right)} \sum_{j=1}^{N} \frac{1}{N} f\left(\mathbf{z}_k|\mathbf{z}_{j,k-1}\right) \\
&= \frac{f\left(\mathbf{d}_k|\mathbf{z}_k\right)}{f\left(\mathbf{d}_k\right)} \sum_{j=1}^{N} \frac{1}{N} \frac{f\left(\mathbf{z}_k|\mathbf{z}_{j,k-1}\right)}{q\left(\mathbf{z}_k|\mathbf{z}_{k-1},\mathbf{d}\right)} q\left(\mathbf{z}_k|\mathbf{z}_{k-1},\mathbf{d}\right).
\end{aligned} \tag{12}
$$

Case I: Subsidence in Groningen

InSAR data of 2009-2010 subsidence (mm)



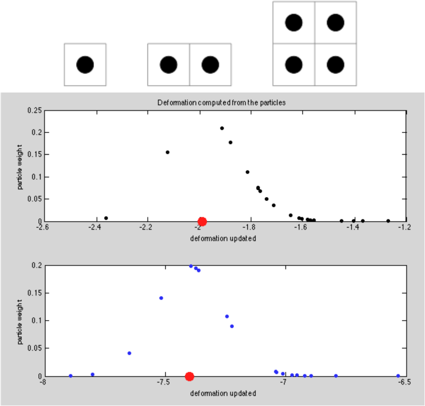




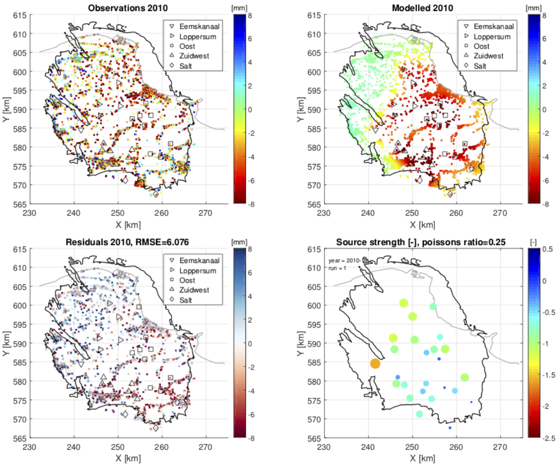
- Studying induced subsidence over the Groningen gas field
- Methodology: particle filter with importance resampling. note: strictly speaking not a filter, because the model is (quasi-)static!
- Estimating the strength using a nucleus of strain (Mogi source) with uncertain strength at the locations of producing wells
- Assimilating InSAR data

Mogi-source strength estimation

- The approach was tested on synthetic experiments
- In each experiment, the number of Mogi sources was increased (1,2,4,16,...)
- If no resampling is applied, degeneracy starts to occur with ... numbers of Mogi sources, and with ... ensemble sizes
- You will experience this yourself in the practical tomorrow

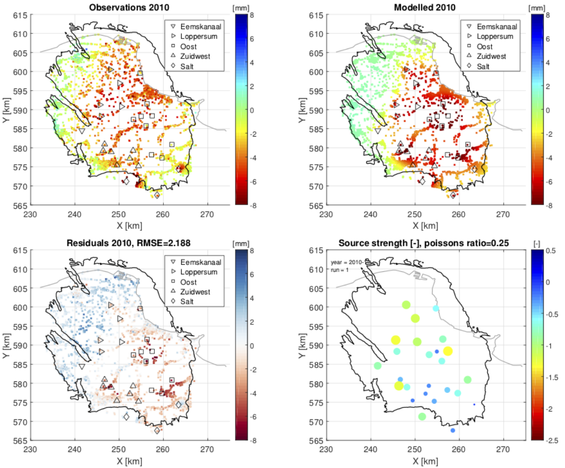


Assimilation actual InSAR data (unfitted)



Available from: `https://github.com/geirev/Data-Assimilation-Fundamentals`

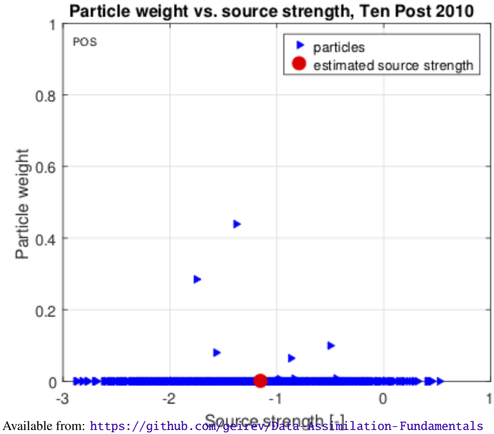
Assimilation actual InSAR data (fitted)



Available from: `https://github.com/geirev/Data-Assimilation-Fundamentals`

Particle weights

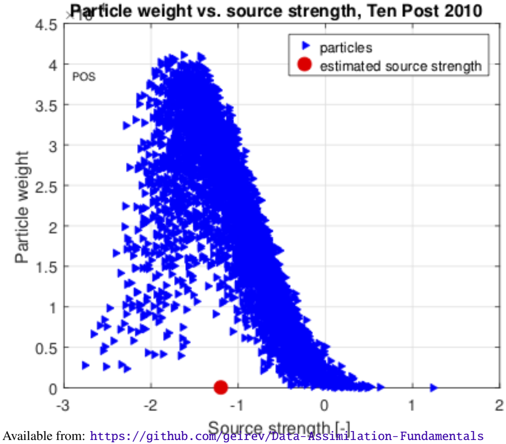
Even with 5000 ensemble members, we observe degeneracy



Available from: https://github.com/geirev/Data-Assimilation-Fundamentals

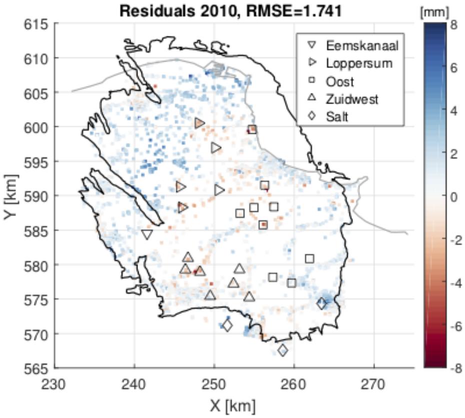
Particle weights with localisation

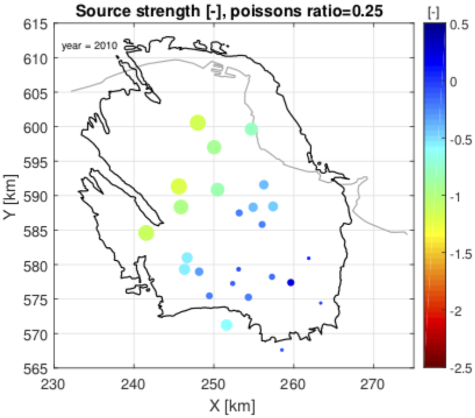
Degeneracy solved with localisation



Available from: https://github.com/geirev/Data-Assimilation-Fundamentals

Assimilation actual InSAR data (fitted, localised)





Available from: https://github.com/geirev/Data-Assimilation-Fundamentals

Case I: Subsidence in Groningen

- Particle filter can be used to estimate Mogi-source strengths as a representation of reservoir compaction
- For synthetic experiments, increasing the number of particles helps to avoid degeneracy
- For realistic experiments, an ensemble size of 5000 particles still leads to degeneracy
- Localisation can help to overcome this