Auto-tuning Hamiltonian Monte Carlo

for high-dimensional model space exploration

Andreas Fichtner
ETH Zurich

Lars Gebraad
ETH Zurich

Andrea Zunino
University of Copenhagen

Christian Boehm
ETH Zurich

Thank you very much for taking the time to read these slides. The text below the figures, such as the one you are reading right now, will provide a little bit more context and explanations. Do not hesitate to contact me if you have questions or comments.

ETH Seismology & Wave Physics

Auto-tuning Hamiltonian Monte Carlo

introduced as *hybrid* Monte Carlo in quantum mechanics [Duane et al. 1987]
more recently in geophysics [Sen & Biswas, 2017; Fichtner et al., 2018]

- **Markov chain sampling** method [e.g., to sample posterior of an inverse problem].
- taking advantage of derivative information
- **long-distance moves + high acceptance rate**
- solve **high-dimensional** inverse problems + **uncertainty quantification**

Hamiltonian Monte Carlo is a Markov-chain Monte Carlo method originally introduced in quantum mechanics under the name hybrid Monte Carlo. It has only recently gained popularity in the solution of geophysical inverse problems. By taking advantage of derivative information, it enables the combination of long-distance moves in model space and a high acceptance rate. This is attractive for high-dimensional Bayesian inverse problems.

ETH Seismology & Wave Physics

Conceptual Introduction

The mathematical theory behind Hamiltonian Monte Carlo is very beautiful and elegant. Yet, to keep it short, we limit ourselves to a conceptual introduction of this method. For more details, you may want to look at this open-access article (Fichtner et al., 2019) https://academic.oup.com/gji/article/216/2/1344/5199200 .

ETH Seismology & Wave Physics

- initial model
  - treated as particle in n-dimensional space
  - equipped with artificial mass matrix **M**

- maximum-likelihood model

MODEL SPACE

Tne basic idea is that a model of dimension n is treated as a partical in n-dimensional space. To this particle we assign a totally artifical mass matrix. This mass matrix can be chosen arbitrarily, but it needs to be symmetric and positive definite. Later we will see that the choice of the mass matrix has a big impact on the performance of the Hamiltonian Monte Carlo sampler.

- initial model

1. Initial model **m**, chosen randomly.
2. Misfit U(**m**) defines potential energy.

e.g.: $U(m) = \sum (d_i(m) - d_i^{obs})^2$

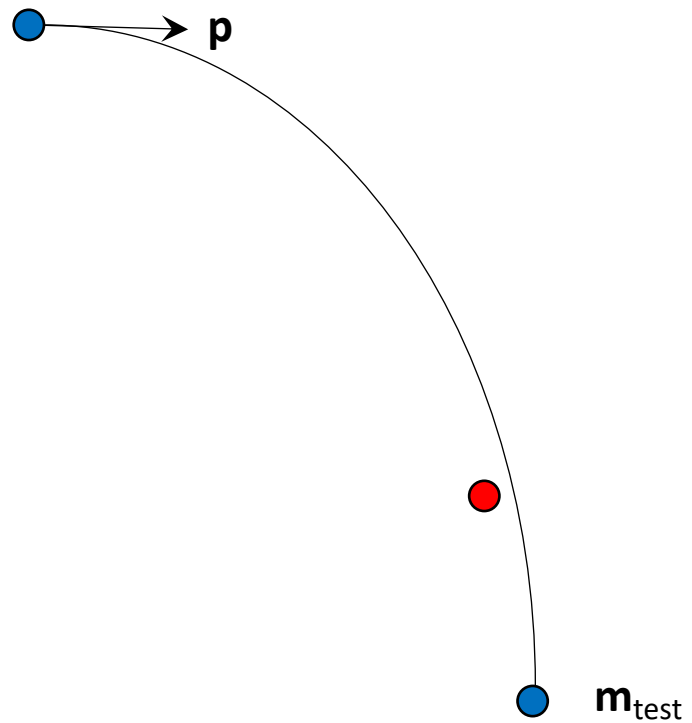
- maximum-likelihood model

Now assume that we have some randomly chosen initial model. This model can be used to compute synthetic data, and the comparison of the synthetic to observed data gives some misfit. We denote this misfit by U(**m**) and we equate it with the potential energy of the particle (model).

1. Initial model **m**, chosen randomly.
2. Misfit U(**m**) defines potential energy.
3. Random momentum **p** [auxiliary quantity].
4. Defines kinetic energy K(**p**).

- maximum-likelihood model

Now that the model (particle) has an artificial mass and an artificial potential energy, we also give it some randomly chosen momentum. The momentum and the mass define the kinetic energy of the particle.

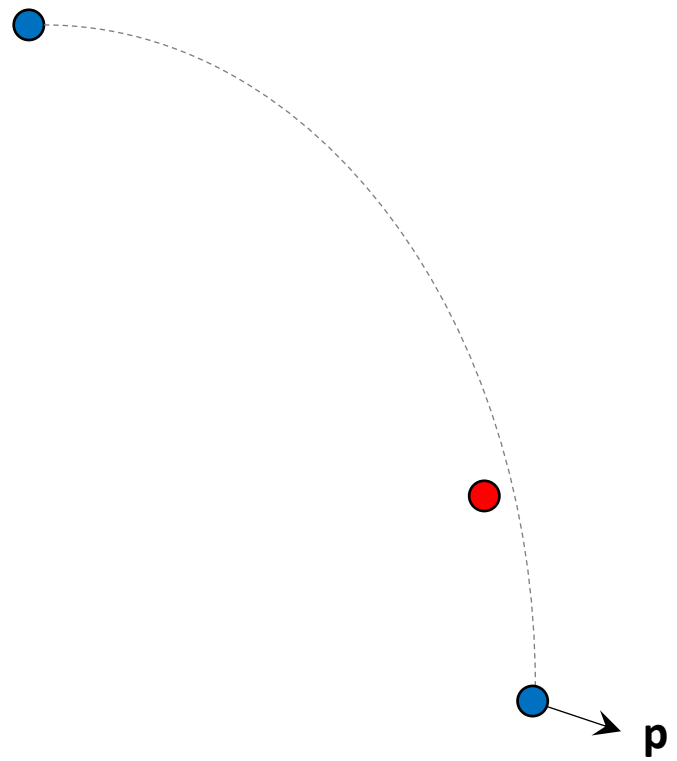


1. Initial model **m**, chosen randomly.
2. Misfit U(**m**) defines potential energy.
3. Random momentum **p** [auxiliary quantity].
4. Defines kinetic energy K(**p**).
5. Solve Hamilton's equations with Hamiltonian

   $H(\mathbf{m},\mathbf{p}) = U(\mathbf{m}) + K(\mathbf{p})$.

6. Move towards a new test model, $\mathbf{m}_{test}$.

With these ingredients, the particle has a Hamiltonian H, i.e., a total energy composed of potential energy (misfit) and kinetic energy. The Hamiltonian enters Hamilton's equations of classical mechanics, which define the trajectory of the particle through model space. Hence, our particle starts moving towards new positions in model space. After some time, we stop the trajectory, and this new position is our new test model.

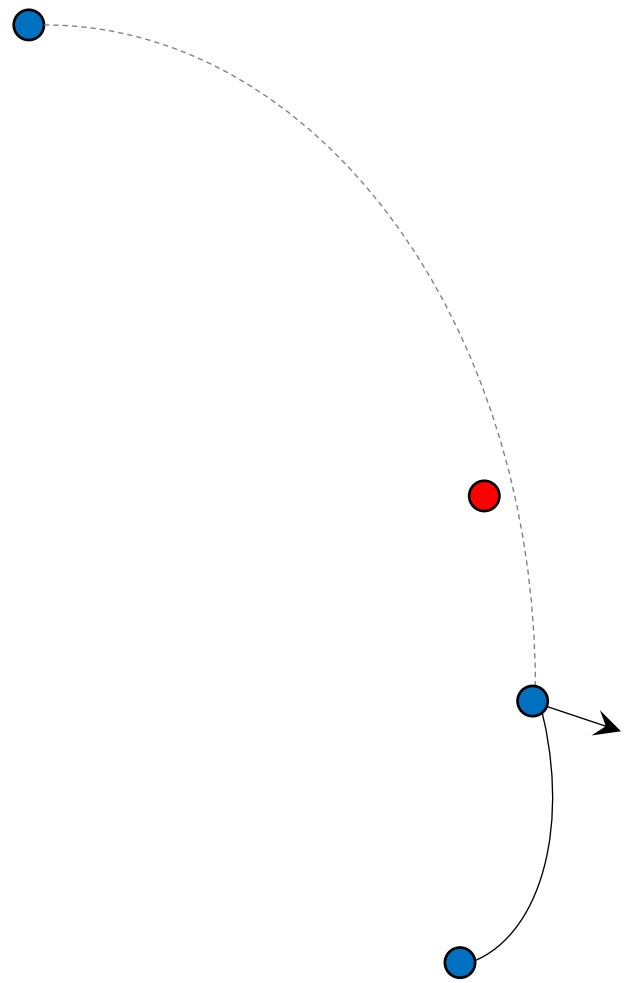


1. Initial model **m**, chosen randomly.
2. Misfit U(**m**) defines potential energy.
3. Random momentum **p** [auxiliary quantity].
4. Defines kinetic energy K(**p**).
5. Solve Hamilton's equations with Hamiltonian

   $H(\mathbf{m},\mathbf{p}) = U(\mathbf{m}) + K(\mathbf{p})$.

6. Move towards a new test model, $\mathbf{m}_{test}$.
7. Evaluate Metropolis rule:
   - If rejected: go back.
   - If accepted: move on.

Then, we evaluate the Metropolis rule. If the test model is rejected, we go back to the previous model. Otherwise, we move on by choosing a new random momentum, solving Hamilton's equations, etc.

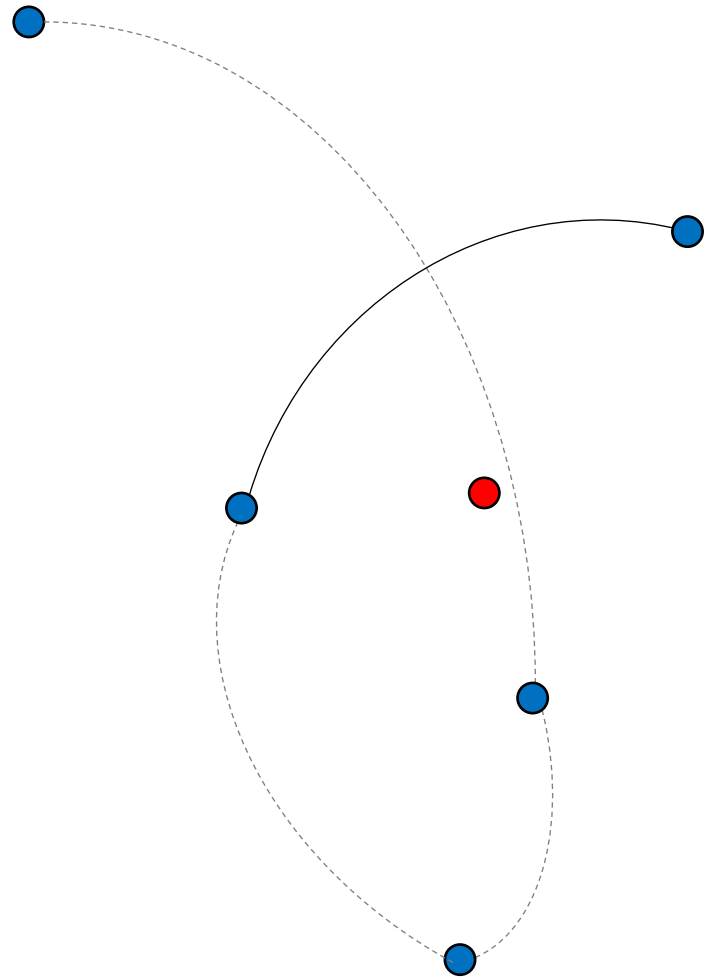


1. Initial model **m**, chosen randomly.
2. Misfit U(**m**) defines potential energy.
3. Random momentum **p** [auxiliary quantity].
4. Defines kinetic energy K(**p**).
5. Solve Hamilton's equations with Hamiltonian

   $H(\mathbf{m},\mathbf{p}) = U(\mathbf{m}) + K(\mathbf{p})$.

6. Move towards a new test model, $\mathbf{m}_{test}$.
7. Evaluate Metropolis rule:
   - If rejected: go back.
   - If accepted: move on.

This gives a new trajectory, and the end of which we have a new test model.



1. Initial model **m**, chosen randomly.
2. Misfit U(**m**) defines potential energy.
3. Random momentum **p** [auxiliary quantity].
4. Defines kinetic energy K(**p**).
5. Solve Hamilton's equations with Hamiltonian

   $H(\mathbf{m},\mathbf{p}) = U(\mathbf{m}) + K(\mathbf{p})$.

6. Move towards a new test model, $\mathbf{m}_{test}$.
7. Evaluate Metropolis rule:
   - If rejected: go back.
   - If accepted: move on.

And so on and so forth.

Key features

Pros:

- Acceptance of long-distance moves [thanks to use of derivatives].
- Fast model space exploration.
- **Cost of generating independent model $\propto n^{1.2}$** [compared to n^2 for Metropolis-Hastings].

Cons:

- Requires derivatives of the forward problem [compensated for high dimensions].

The outstanding advantages of Hamiltonian Monte Carlo are the acceptance of long-distance moves through model space, which in turn enables a fast exploration of model space. This can be expressed quantitatively in terms of the low numerical cost of generating an independent model. Of course, the downside is the need to actually compute derivatives, needed for the solution of Hamilton's equations.

Key features

Pros:

- Acceptance of long-distance moves.
- Fast model space exploration.
- **Cost of generating independent model $\propto n^{1.2}$** [compared to n^2 for Metropolis-Hastings].
- Can be tuned for high efficiency [n(n+1)/2 elements of the mass matrix **M**]

Cons:

- Requires derivatives of the forward problem [compensated for high dimensions].
- Must be tuned to avoid inefficiency.

Another key advantage of Hamiltonian Monte Carlo is that is can be tuned to increase efficiency. In fact, there are n(n+1)/2 tuning parameters in the mass matrix, where n is the model space dimension. So there are very many tuning parameters. This advantage implies a disadvantage: One actually has to tune. Otherwise the method can be rather inefficient. (Two additional tuning parameters are the integration step length and the length of the trajectory.)

Example: 2D Full-Waveform Inversion

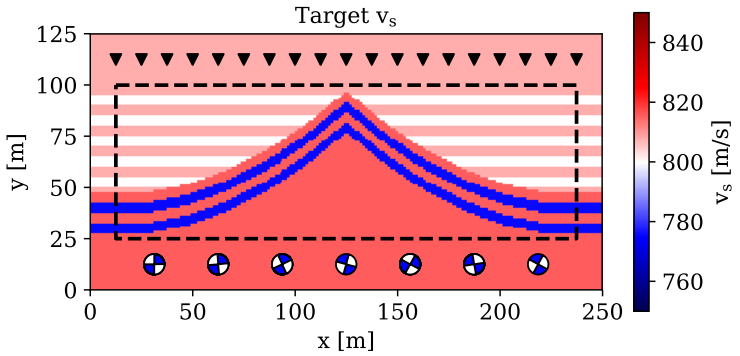
Now let‘s see how this can be useful in the solution of Bayesian full-waveform inversion. For more details, have a look at this JGR article (Gebraad et al., 2020) https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2019JB018428 . (Let us know if you cannot access the full text.)

- **2D elastic wave propagation** [staggered-grid FD, f_{max}=50 Hz].
- Model parameters: v_p, v_s, ρ.
- Grid points: 10’800.
- **Model space dimension: 32’400**

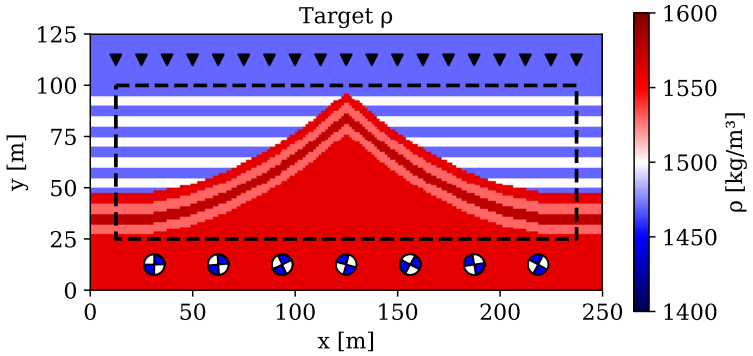
We consider a 2D elastic wave propagation setup. The model parameters that we try to constrain are the P velocity v_p, the S velocity v_s, and density ρ. In our 2D domain, we have 10’800 grid points, which then gives a total of 32’400 model parameters.

- **2D elastic wave propagation** [staggered-grid FD, f_{max}=50 Hz].
- Model parameters: v_p, v_s, ρ.
- Grid points: 10’800.
- **Model space dimension: 32’400**

S velocity



density



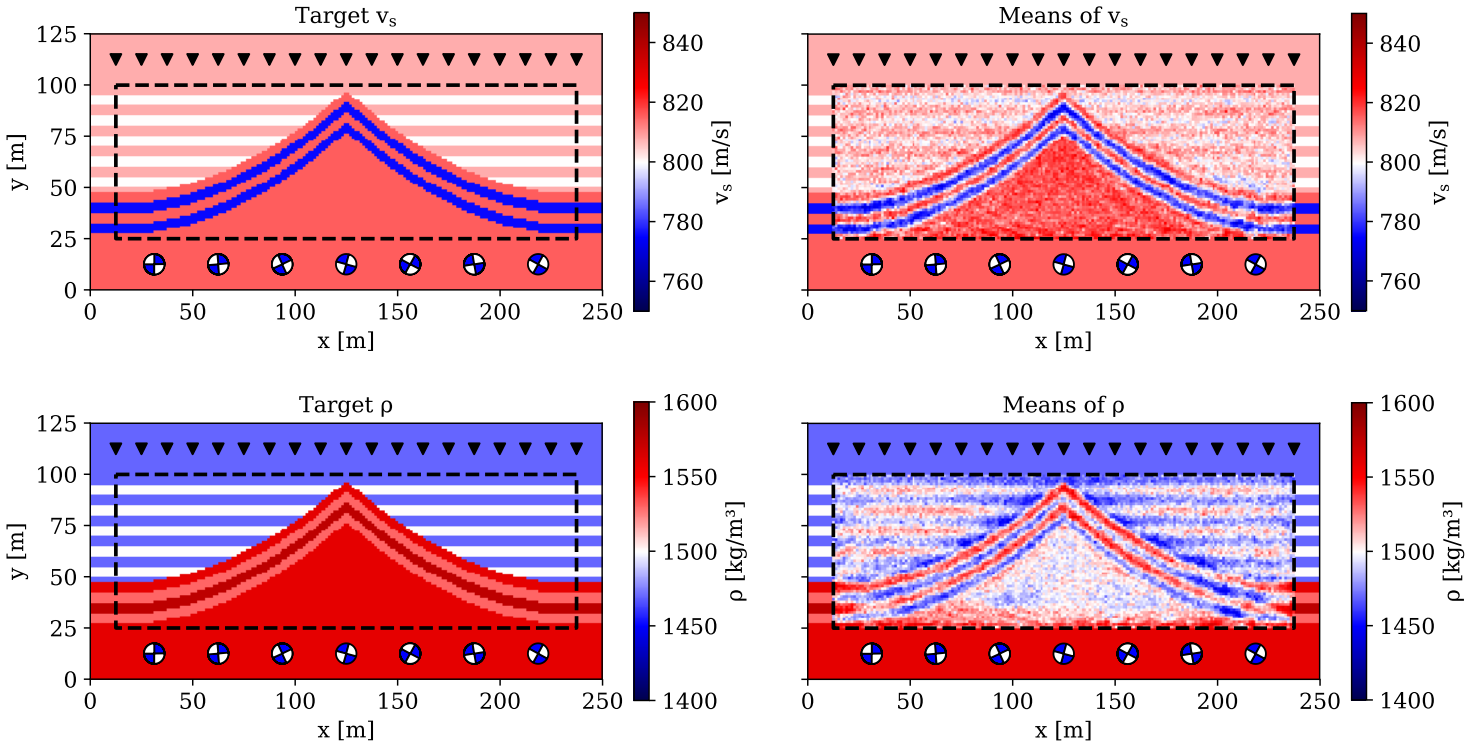
On this slide you can see the target models of S velocity and density. We use these target models to compute artificial data for a synthetic inversion based on Hamiltonian Monte Carlo sampling.

ETH Seismology & Wave Physics

- **2D elastic wave propagation** [staggered-grid FD, f_{max}=50 Hz].
- Model parameters: v_p, v_s, ρ.
- Grid points: 10’800.
- **Model space dimension: 32’400**

S velocity

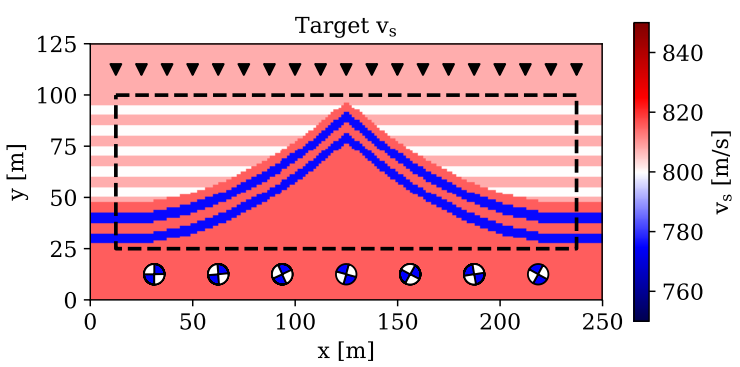
density

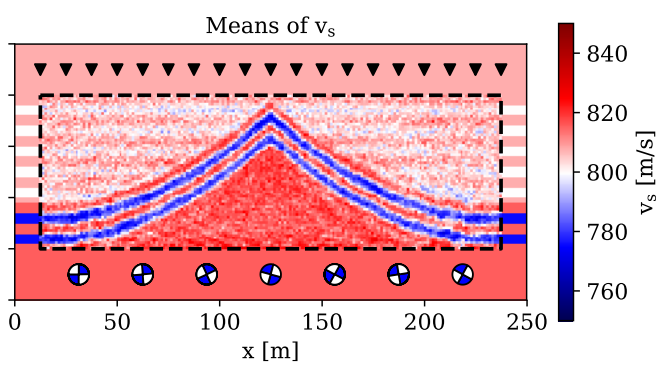


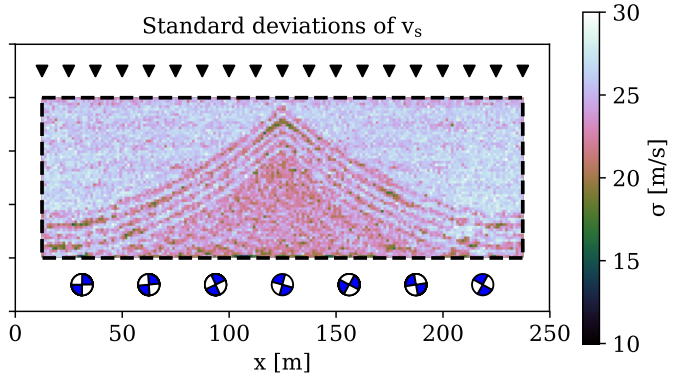
The result of the sampling is a large number of models that can be used to compute various aspects of the complete posterior probability densities. One of these aspects is the mean model, shown here in the middle. It closely resembles the target model, providing some first indications that the method actually works.

- **2D elastic wave propagation** [staggered-grid FD, f_{max}=50 Hz].
- Model parameters: v_p, v_s, ρ.
- Grid points: 10’800.
- **Model space dimension: 32’400**

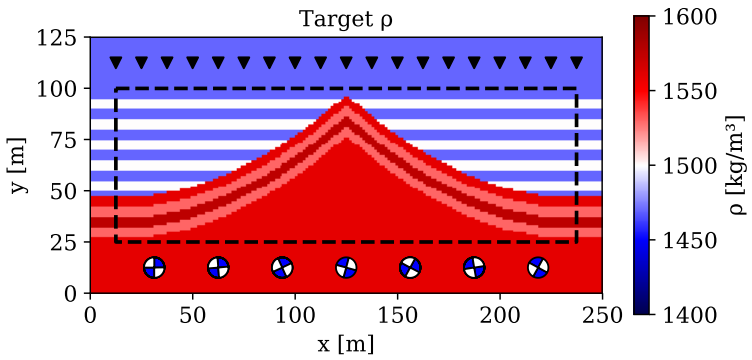
S velocity

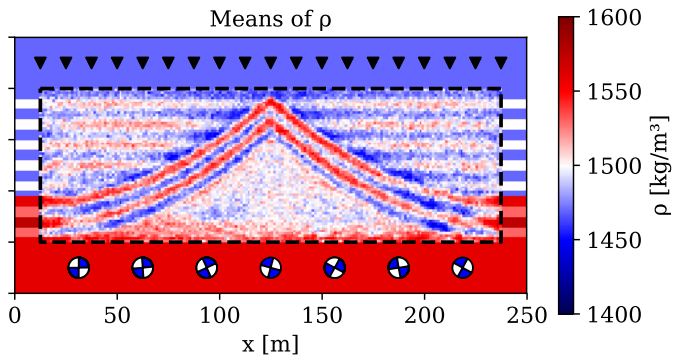


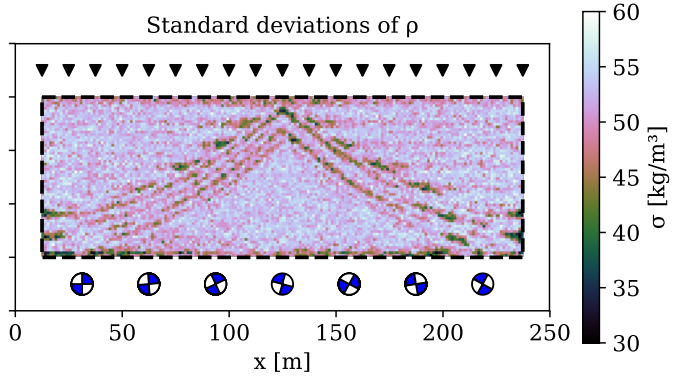




density







Furthermore, we may compute the posterior standard deviation, or other proxies of model resolution. **The most important point here is that we obtain the full posterior distribution without any need for subjective regularisation.**

ETH Seismology & Wave Physics

Drawing independent models

So far so good. Now let us look a bit more at the efficiency of the method. It very much depends on the ability to draw independent models. The more quickly samples become independent, the more quickly we actually learn something new about model space, and the more quickly the algorithm converges.

Drawing independent models

important for

Efficient model- or null-space exploration

Convergence error of Monte Carlo integrals $\propto 1/\sqrt{\mathrm{N}_{\mathrm{independent}}}$

This can in fact be made quantitative using the well-known convergence error of Monte Carlo methods, which depends on the number of *independent* samples.

Example 1: 1000-D Gaussian

Model covariance matrix: $C_{1,1}=0.100$, $C_{2,2}=0.101$, ..., $C_{1000,1000}=1.100$

Mass matrix: $\mathbf{M}=\mathbf{I}$

auto-correlation of sample chain

measure of the independence of successive samples

Let us look, in the interest of illustration, at a simple example: The sampling of a Gaussian. Of course, this may not be the most fascinating problem, but in 1000 dimensions, this actually becomes non-trivial. As a measure of model independence we consider the auto-correlation of the sequence of samples.

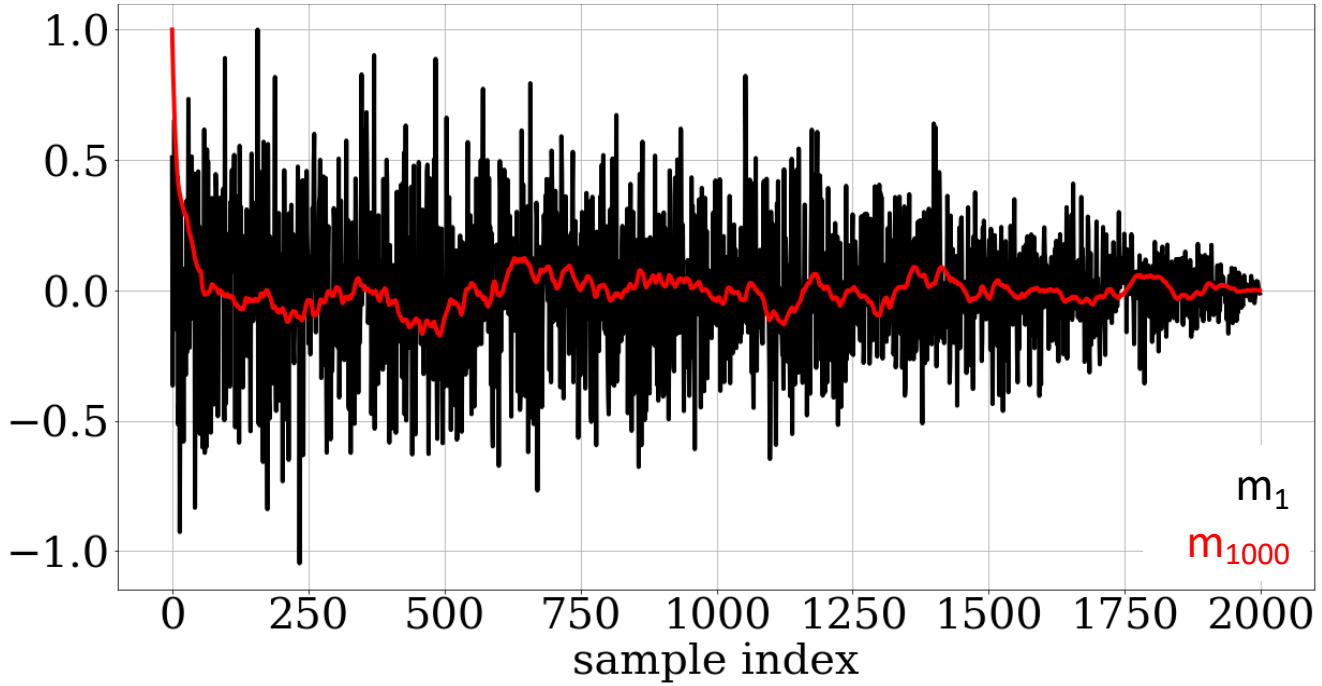
Example 1: 1000-D Gaussian

Model covariance matrix: $C_{1,1}=0.100$, $C_{2,2}=0.101$, ..., $C_{1000,1000}=1.100$

Mass matrix: **M**=**I**

auto-correlation of sample chain

measure of the independence of successive samples



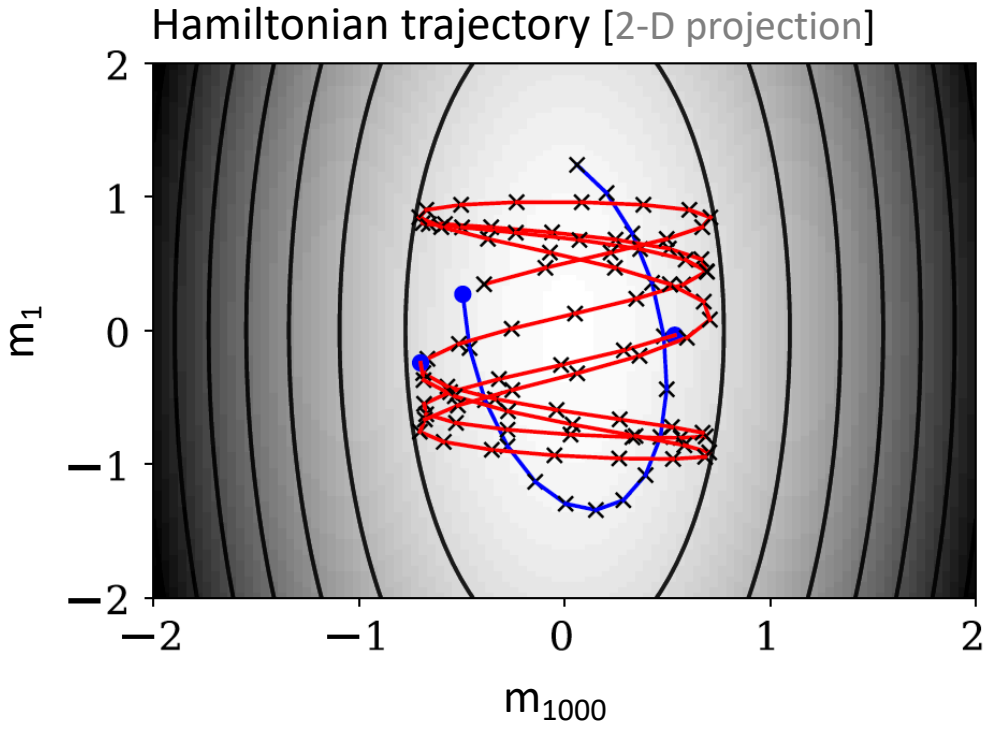
First, we simplistically use the identity matrix **I** as mass matrix. As we see in the figure, the samples of parameter m_1 remain highly correlated for a long time. Hence, we waste computational power on drawing thousands of samples, but these samples all look very similar.

Example 1: 1000-D Gaussian

Model covariance matrix: $C_{1,1}=0.100$, $C_{2,2}=0.101$, ..., $C_{1000,1000}=1.100$

Mass matrix: **M**=**I**

Mass matrix: **M**=$\mathbf{C}^{-1}$



This unpleasant behaviour can be explained by looking at the Hamiltonian trajectories. Shown in red is a typical example. Basically, we see that progress in m_1 direction is much slower than in m_{1000} direction. Hence, the high correlation of the 1-component of the samples. Shown in blue is the case where the mass matrix is the inverse covariance matrix. Now this looks much more balanced. We progress equally fast in all directions. Can we use this in practice?

Ideal: $\mathbf{M}=\mathbf{C}^{-1}=\mathbf{H}$ [mass matrix = Hessian]

Problems: Hessian cannot be computed or stored explicitly

Autotuning: Approximate the Hessian on the fly

- Use last couple of samples to approximate **H·vector**.
- Closely related to L-BFGS method from nonlinear optimisation [Nocedal, 1980].
- Use approximate **H** as **M** in computation of Hamiltonian trajectories.

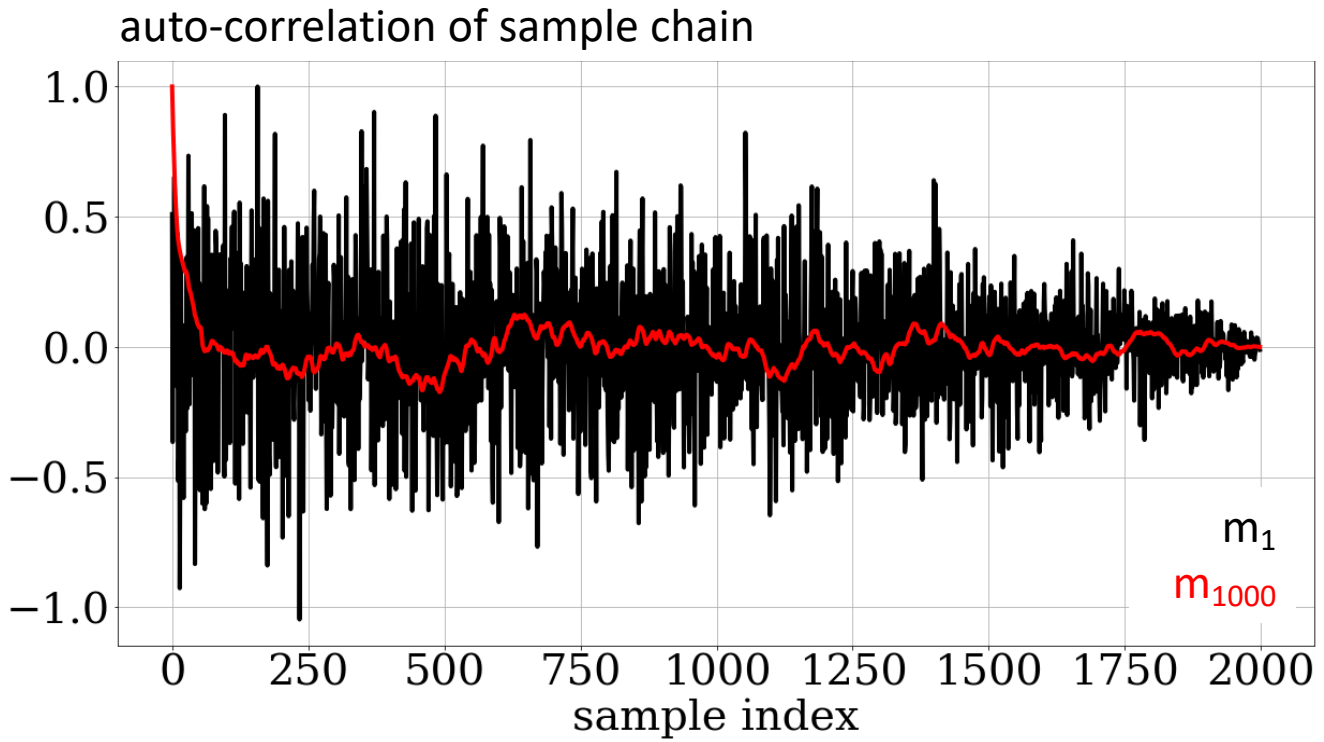
The inverse covariance matrix is the same as the Hessian. Of course, the problem is that – for high-dimensional problems – the Hessian cannot be computed or stored explicitly. **The solution is auto-tuning, where we approximate the Hessian on the fly.** This slide summarises the basic concept, and the next two slides show the effect for our Gaussian toy problem.

Return to Example 1:

Autotuning: Approximate the Hessian on the fly

auto-tuning off

[mass matrix: **M**=**I**]

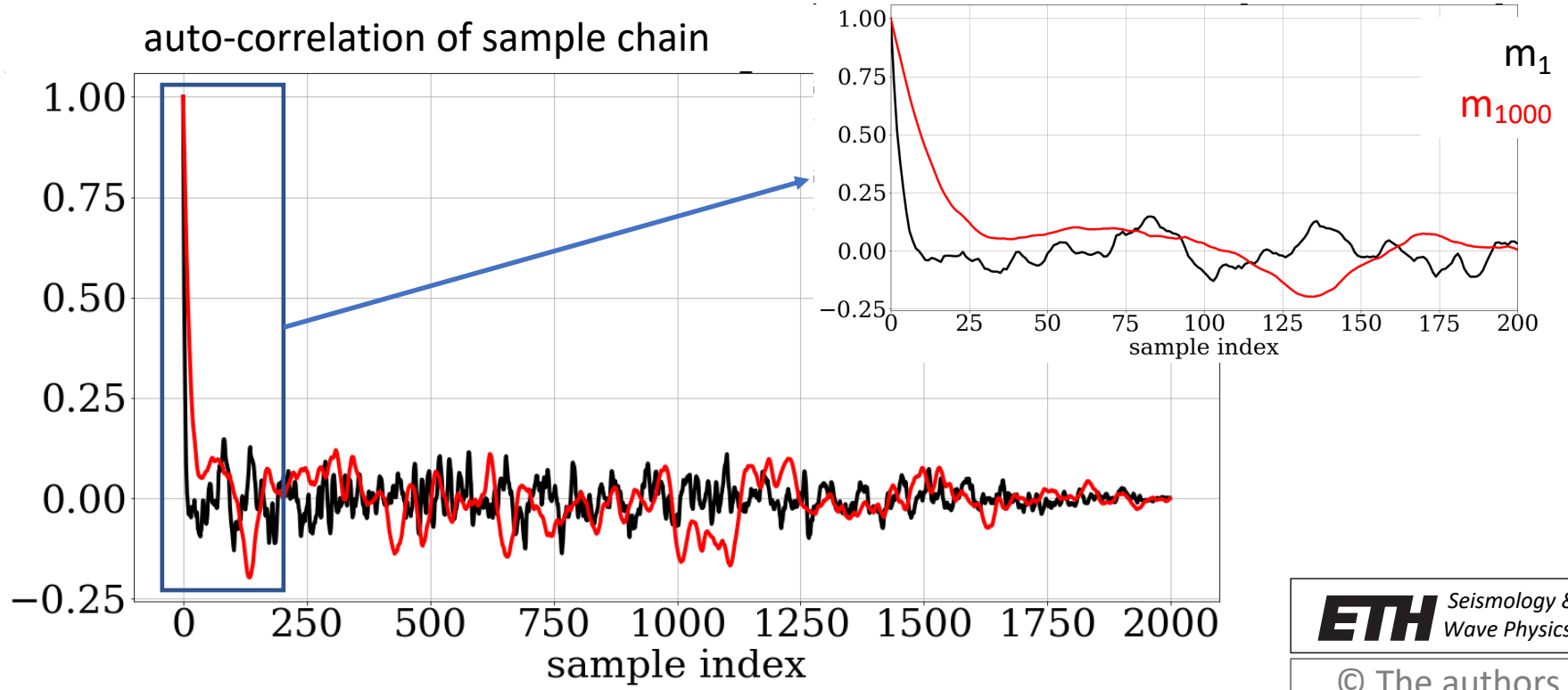


Return to Example 1:

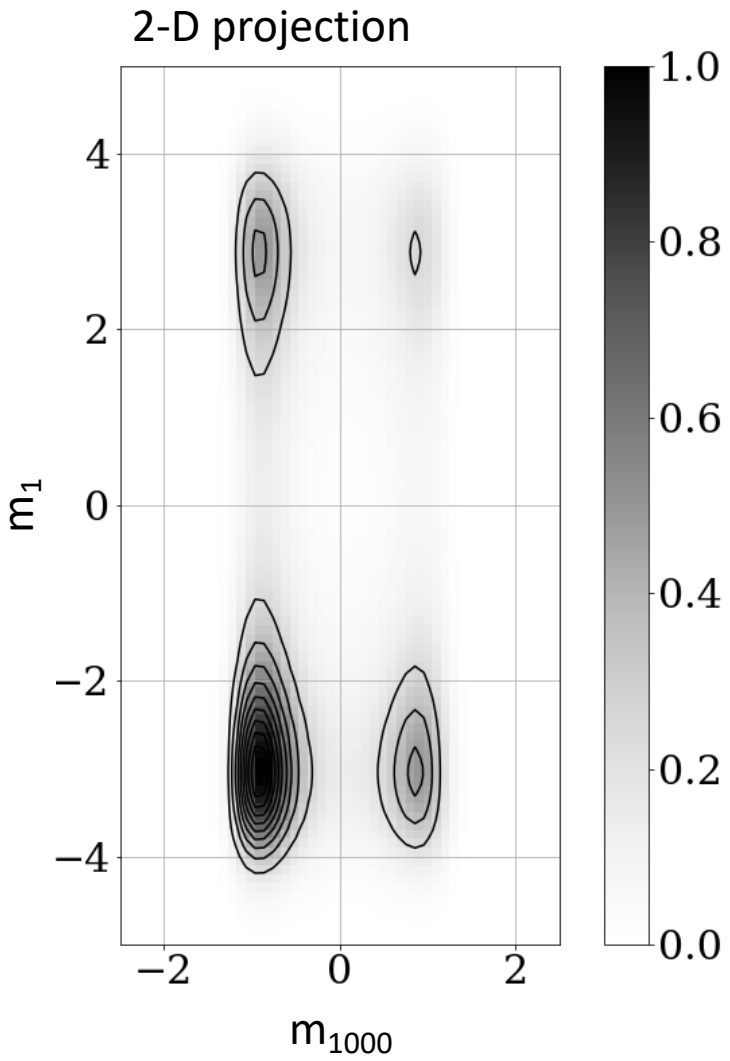
Autotuning: Approximate the Hessian on the fly

auto-tuning on:

[mass matrix: **M**≈**H**]

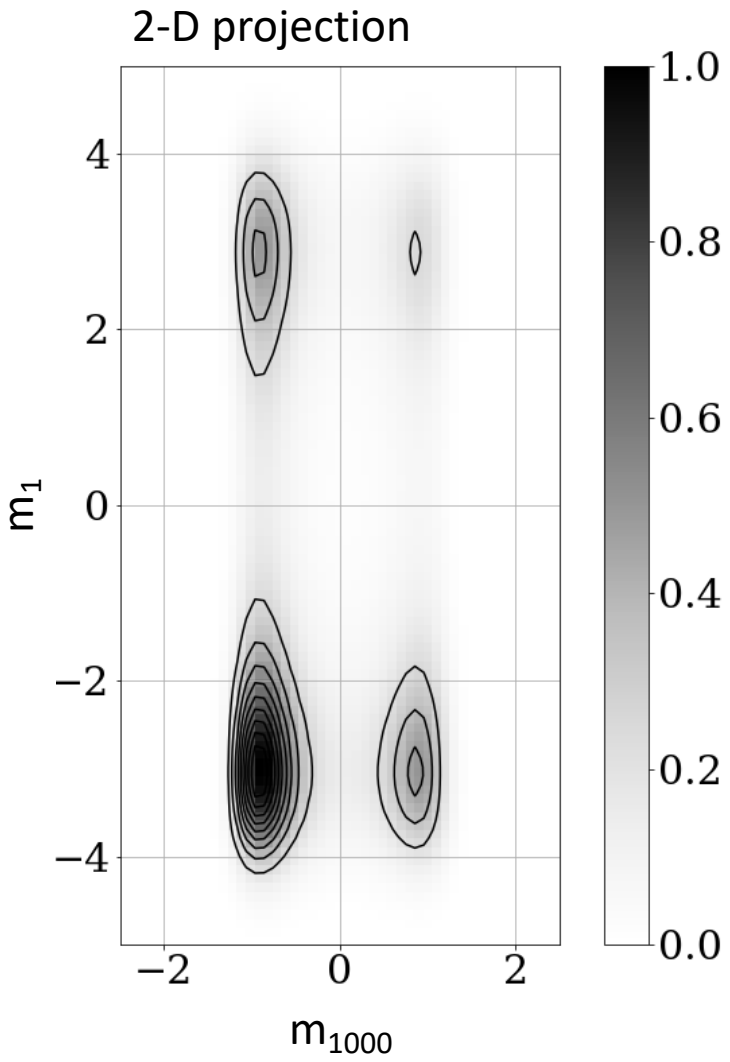


Example 2: 1000-D Styblinski-Tang function [4^{500} local minima]



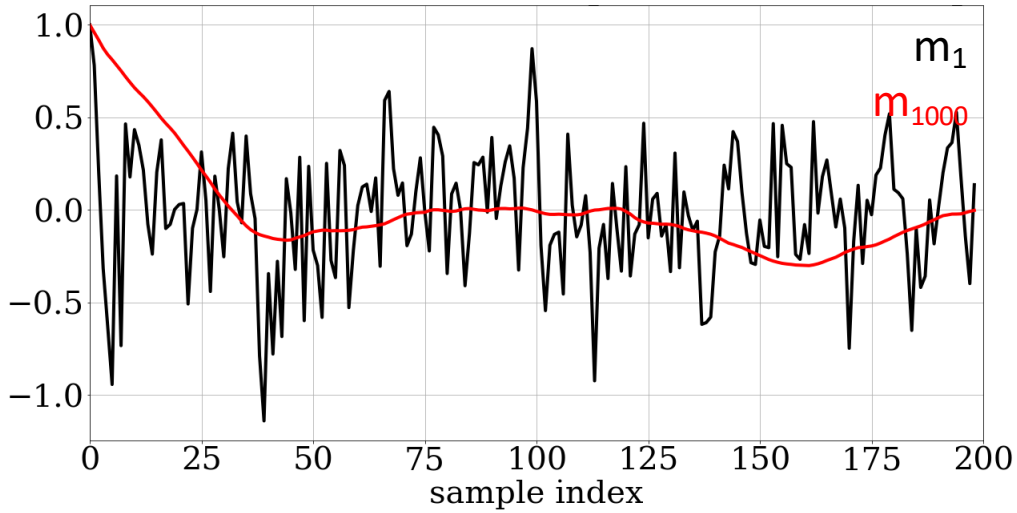
Of course, the whole concept also works for more complicated cases, such as the 1000-dimensional Styblinski-Tang function. This function, often used as test function in numerical optimisation, has 4^{500} local minima.

Example 2: 1000-D Styblinski-Tang function [4^{500} local minima]



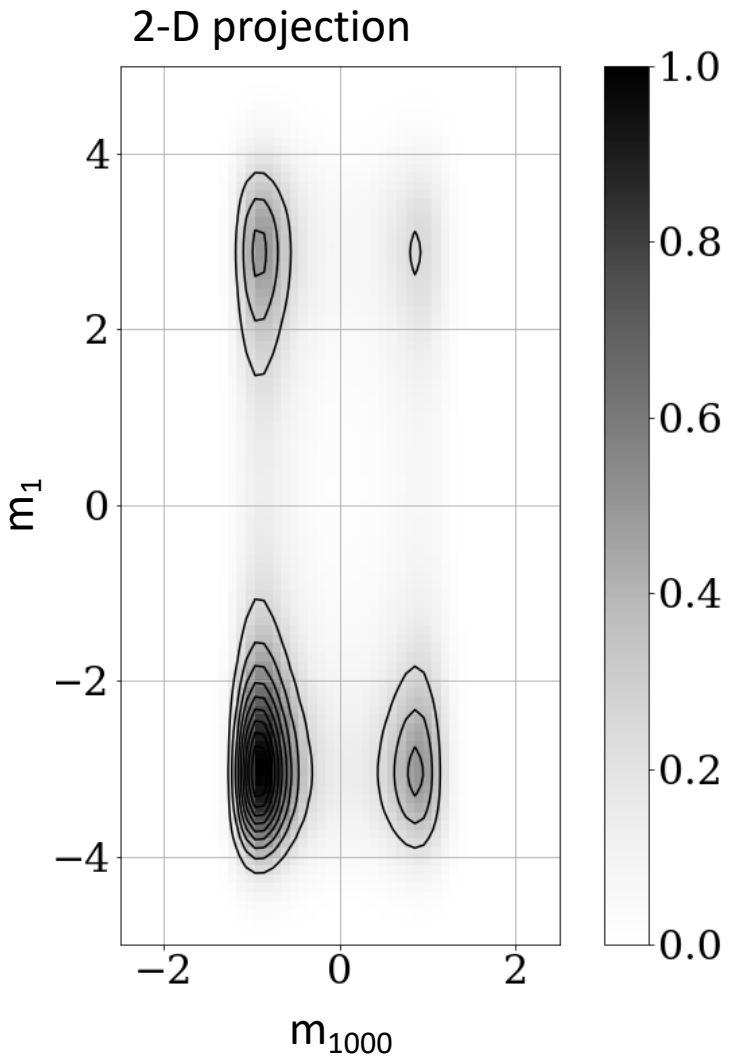
auto-tuning off

highly correlated m_1



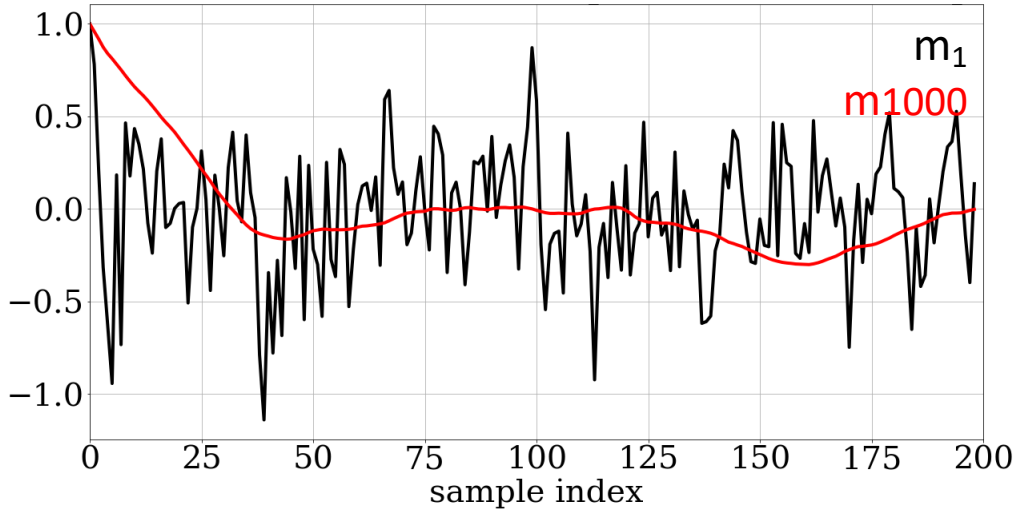
As in the Gaussian example, we observe a strong correlation of the samples when auto-tuning is off, and when the mass matrix is the identity matix. In contrast, when auto-tuning is used, the samples decorrelate very quickly. This can be seen on the following slide.

Example 2: 1000-D Styblinski-Tang function [4^{500} local minima]



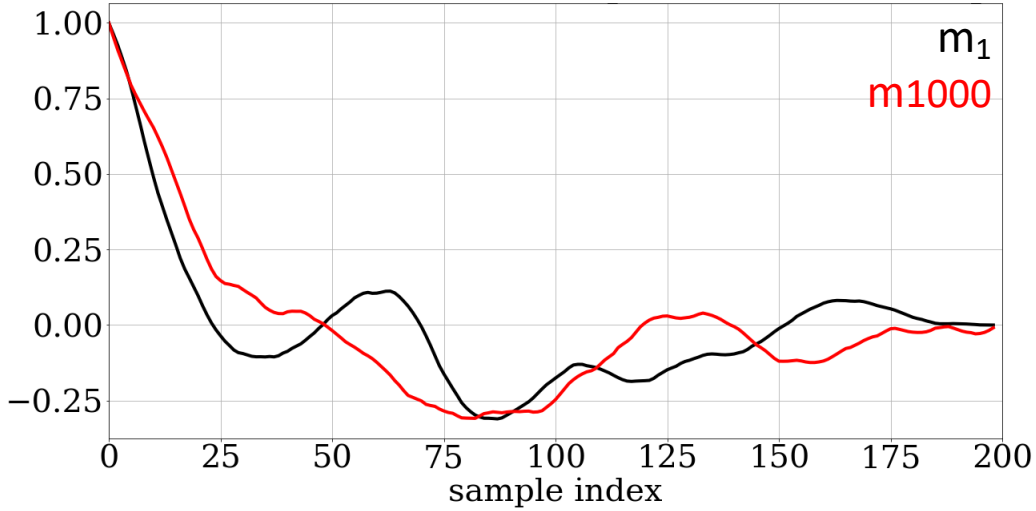
auto-tuning **off**

highly correlated m_1



auto-tuning **on**

largely uncorrelated m_1



Conclusions

1) HMC *status quo*: Realistic inverse problems with *O*(10‘000) parameters
   - computing derivatives via adjoint techniques
   - in particular 2D acoustic and elastic full-waveform inversion

2) Tuning is critical
   - mostly the mass matrix **M**
   - controls the independence of successive samples

3) Auto-tuning
   - HMC + L-BFGS
   - Very rapid generation of independent samples
   - **Great promise for realistic applications**

For questions, please do not hesitate to enter the discussion or to contact me via email: andreas.fichtner@erdw.ethz.ch.

ETH Seismology & Wave Physics