Variational Inversion

Andrew Curtis
Xin Zhang

University of Edinburgh

Scotland

Generalised **Optimisation** for Probability Distributions

- **Variational Bayesian Inference**
- Synthetic tests
- Application to Grane array data

– *arXiv: Zhang & Curtis, 2019*

- Application to Full Waveform Inversion problem – *ESSOAR: Zhang & Curtis, 2020*

Variational Inference

- Bayesian solution

$$p(\mathbf{m}|\mathbf{d}) = \frac{p(\mathbf{d}|\mathbf{m})p(\mathbf{m})}{p(\mathbf{d})}$$

$\mathbf{d}$=data, $\mathbf{m}$=parameters

- Variational methods replace **stochastic sampling** with **optimisation** of functions

- Strategy 1: **fit** semi-analytic functions to $p(\mathbf{m}|\mathbf{d})$
  - Choose family of functions $q(\mathbf{m}, \varphi)$, φ=parameters [c.f. φ=Gaussian mean & covar]
  - Optimise φ s.t. $q(\mathbf{m}|\varphi) \cong p(\mathbf{m}|\mathbf{d})$
- Define a measure of difference between q and p; then minimize it.

- Strategy 2: generate a set of **samples** of $p(\mathbf{m}|\mathbf{d})$ by **optimisation**

arXiv: Zhang & Curtis, 2019

Variational Inference

- **Kullback-Liebler divergence** measures difference between q and p:

log(evidence) : intractable

$$KL[q||p] = E_{q(\mathbf{m})}[\log q(\mathbf{m}|\varphi)] - E_{q(\mathbf{m})}[\log p(\mathbf{m}|\mathbf{d})] + \log p(\mathbf{d})$$

- KL ≥ 0 and KL $= 0$ when $q = p$. Rearrange…

log(evidence) : constant w.r.t. q

$$KL[q||p] + E_{q(\mathbf{m})}[\log p(\mathbf{m}|\mathbf{d})] - E_{q(\mathbf{m})}[\log q(\mathbf{m}|\varphi)] = \log p(\mathbf{d})$$

- Evidence lower bound (ELBO)

Bayes Rule, prior, likelihood
→ Efficient, case-specific analytical methods (Nawaz & Curtis 2018/19/20)

$$ELBO(q) = E_{q(\mathbf{m})}[\log p(\mathbf{m}|\mathbf{d})] - E_{q(\mathbf{m})}[\log q(\mathbf{m}|\varphi)]$$

Expectations w.r.t. q which we choose

To Maximise ELBO ➔ Minimise KL divergence (difference) between q and p.

Automatic Differential Variational Inference (ADVI)

- Transform constrained parameter $\mathbf{m}$ to real space: $\boldsymbol{\theta} = T(\mathbf{m})$
- In transformed space assume a Gaussian distribution:

$$q(\boldsymbol{\theta}; \varphi) = Normal(\boldsymbol{\theta}|\boldsymbol{\mu}, \mathbf{L}\mathbf{L}^T)$$

- Standardize the normal distribution: $\boldsymbol{\eta} = S_\varphi(\boldsymbol{\theta})$

$$q(\boldsymbol{\eta}) = Normal(\boldsymbol{\eta}|\mathbf{0}, \mathbf{I})$$

- Gradient of ELBO = $\mathcal{L}$ can be calculated:

$$\nabla_{\boldsymbol{\mu}}\mathcal{L} = E_{N(\boldsymbol{\eta})}[\nabla_{\mathbf{m}} \log p(\mathbf{d}, \mathbf{m}) \nabla_{\boldsymbol{\theta}} T^{-1}(\boldsymbol{\theta}) + \nabla_{\boldsymbol{\theta}} \log|det \mathbf{J}_{T^{-1}}(\boldsymbol{\theta})|]$$

$$\nabla_{\mathbf{L}}\mathcal{L} = E_{N(\boldsymbol{\eta})}[\nabla_{\mathbf{m}} \log p(\mathbf{d}, \mathbf{m}) \nabla_{\boldsymbol{\theta}} T^{-1}(\boldsymbol{\theta}) + \nabla_{\boldsymbol{\theta}} \log|det \mathbf{J}_{T^{-1}}(\boldsymbol{\theta})| \boldsymbol{\eta}^T] + (\mathbf{L}^{-1})^T$$

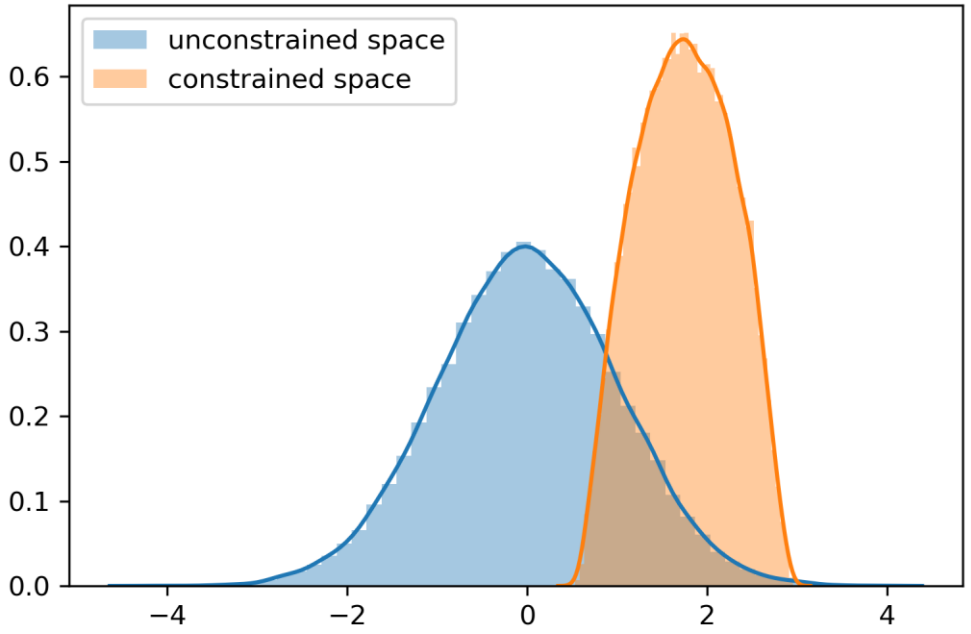
Expectations calculated by MC. But sampling N(0,1) → low number of samples (even 1)

- Maximize ELBO $\mathcal{L}$ using **gradient ascent**

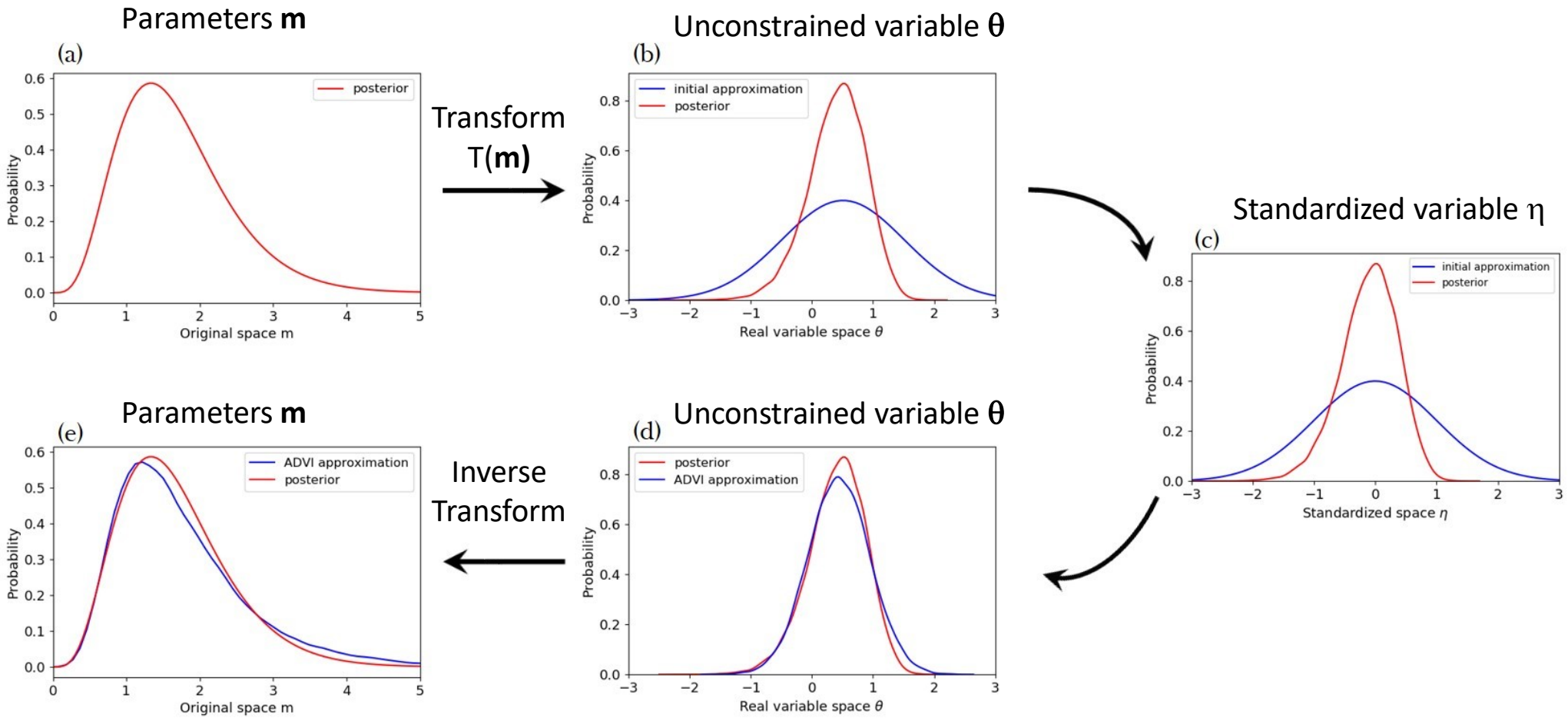
Kucukelbir et al., 2017 JMLR

Transform

$\theta_i = \log(m_i - a) - \log(b - m_i)$ where a and b are lower and upper bounds



Automatic Differential Variational Inference (ADVI)



Variational Inference based on **Invertible Transforms**

- Approximate posterior p using a series of transforms:

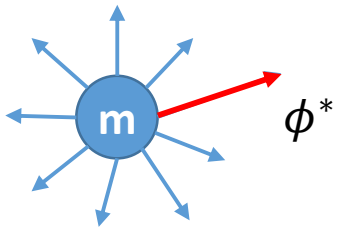
$$q_n = T_n \cdots T_1 T_0(q_0)$$

- Optimize transform T_i by minimizing KL divergence.

- *Normalizing flow, Stein variational methods, etc.*

Stein Variational Gradient Descent(SVGD)

- Assume a transformation $T(\mathbf{m}) = \mathbf{m} + \epsilon\phi(\mathbf{m})$



$$\nabla_\epsilon KL[q_{[T]}||p]|_{\epsilon=0} = -E_{\mathbf{m}\sim q}[trace\left(A_p\phi(\mathbf{m})\right)]$$

where $A_p\phi(\mathbf{m}) = \nabla_{\mathbf{m}}\log p(\mathbf{m}|\mathbf{d})\phi(\mathbf{m})^T + \nabla_{\mathbf{m}}\phi(\mathbf{m})$

SAMPLES

Standard gradients (as used in linearised inversion)

- ϕ^* that maximize the negative gradient:

$$\phi^*_{q,p}(\mathbf{m}) = E_{\mathbf{m}'\sim q}[k(\mathbf{m}',\mathbf{m})\nabla_{\mathbf{m}'}\log p(\mathbf{m}'|\mathbf{d}) + \nabla_{\mathbf{m}'}k(\mathbf{m}',\mathbf{m})]$$

where k is a kernel, e.g. a RBF kernel:

$$k(\mathbf{m},\mathbf{m}') = \exp(-\frac{1}{h}\|\mathbf{m}-\mathbf{m}'\|^2)$$

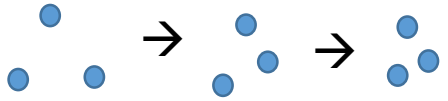
Liu and Wang, 2016 arXiv.

Stein Variational Gradient Descent(SVGD)

- **Input:** A target distribution $p(\mathbf{m})$ and a set of initial particles $\{\mathbf{m}_i^0\}_{i=1}^n$

for iteration j do

$$\mathbf{m}_i^{j+1} \leftarrow \mathbf{m}_i^j + \epsilon_j \phi^*(\mathbf{m}_i^j)$$



where

$$\phi^*(\mathbf{m}) = \frac{1}{n}\sum_{k=1}^{n}[k\left(\mathbf{m}_k^j, \mathbf{m}\right)\nabla_{\mathbf{m}_k^j}\log p\left(\mathbf{m}_k^j\right) + \nabla_{\mathbf{m}_k^j} k(\mathbf{m}_k^j, \mathbf{m})]$$

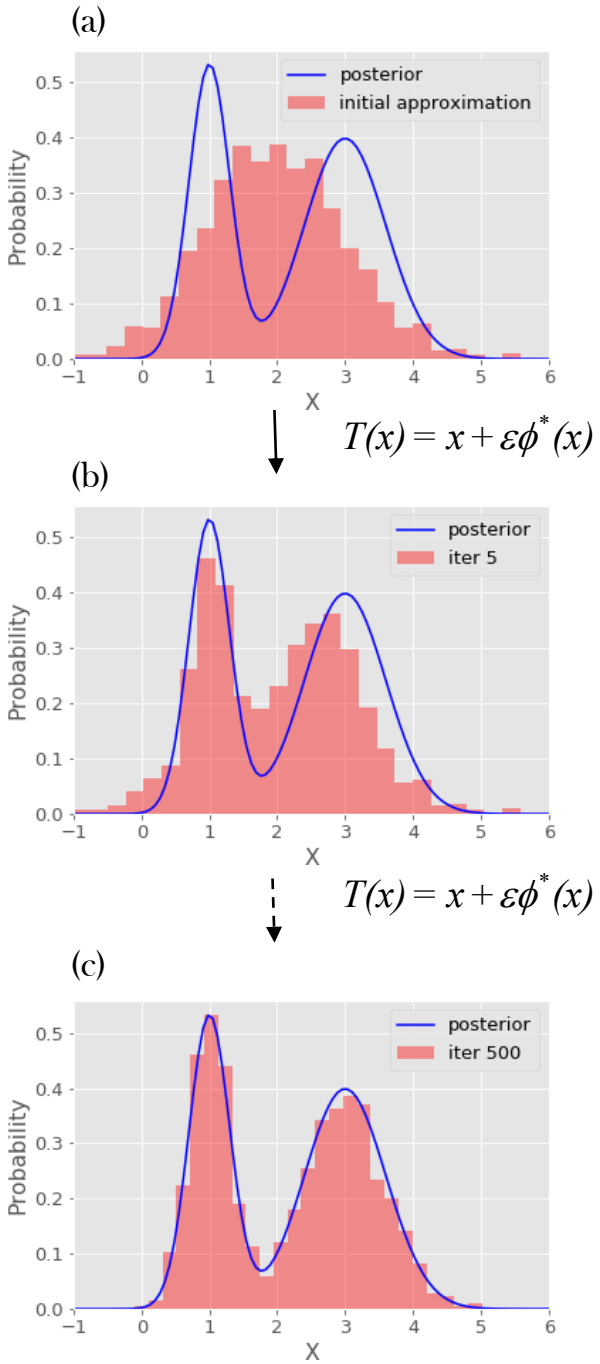
- **Output:** A set of particles $\{\mathbf{m}_i\}_{i=1}^n$ whose density approximates the target distribution
  Also a set of transforms $q_n = T_n \cdots T_1 T_0(q_0)$

- Found by **optimisation** rather than stochastic sampling.
  - Faster, parallelizable, more scalable → **optimal** set of model space samples

Liu and Wang, 2016 arXiv.

SVGD

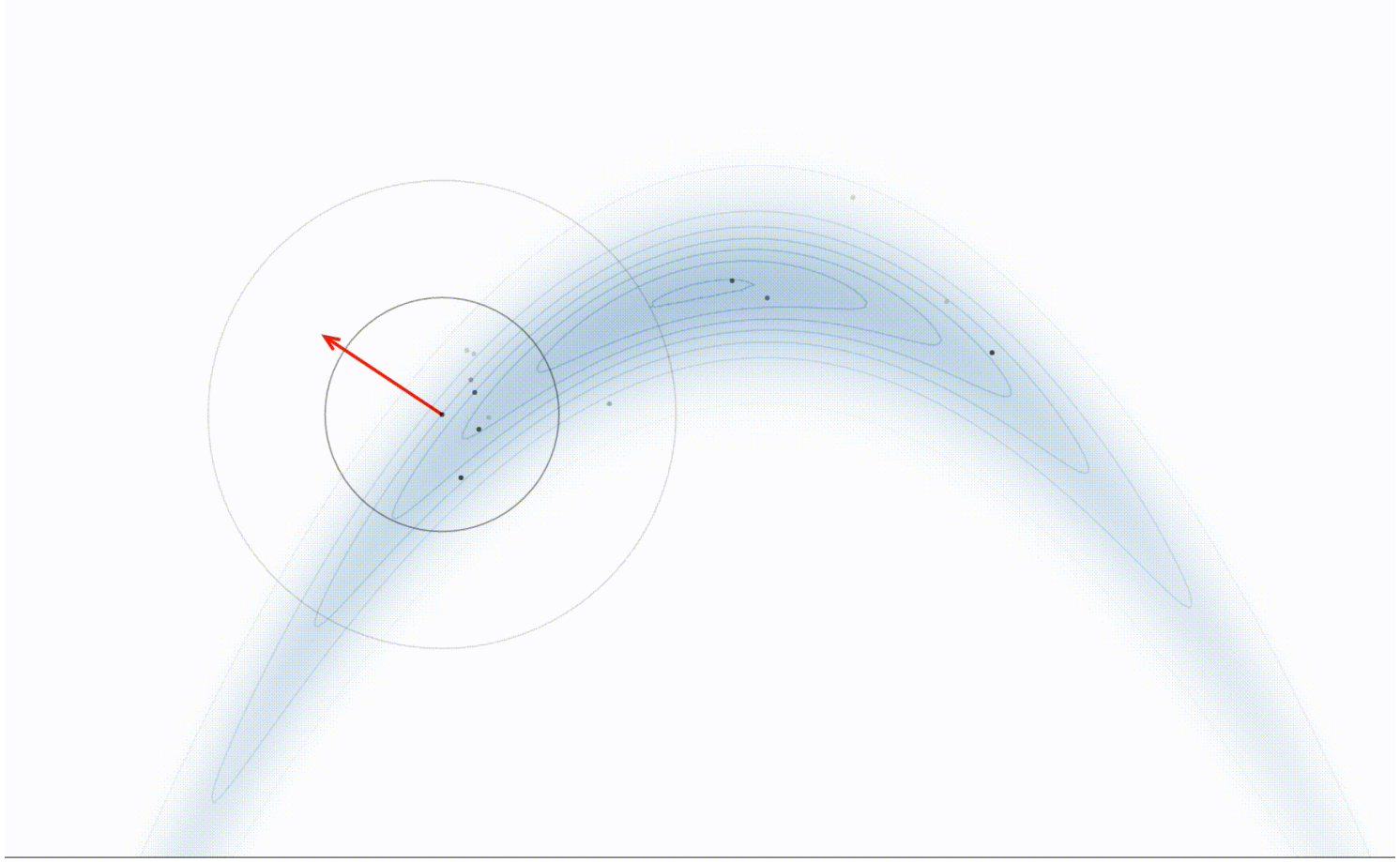
- Target probability
- Histogram of particles



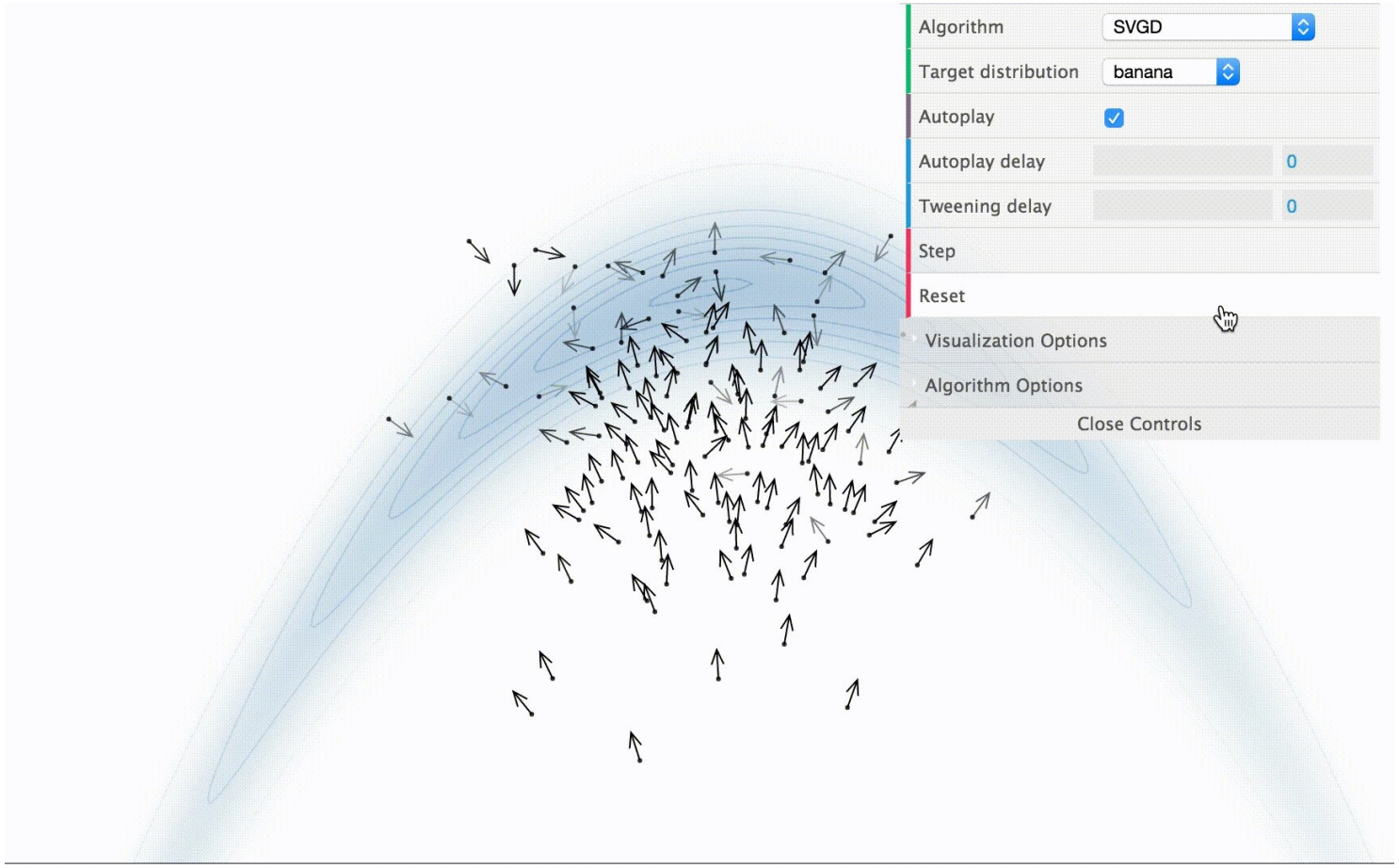
Try this:

https://chi-feng.github.io/mcmc-demo/app.html

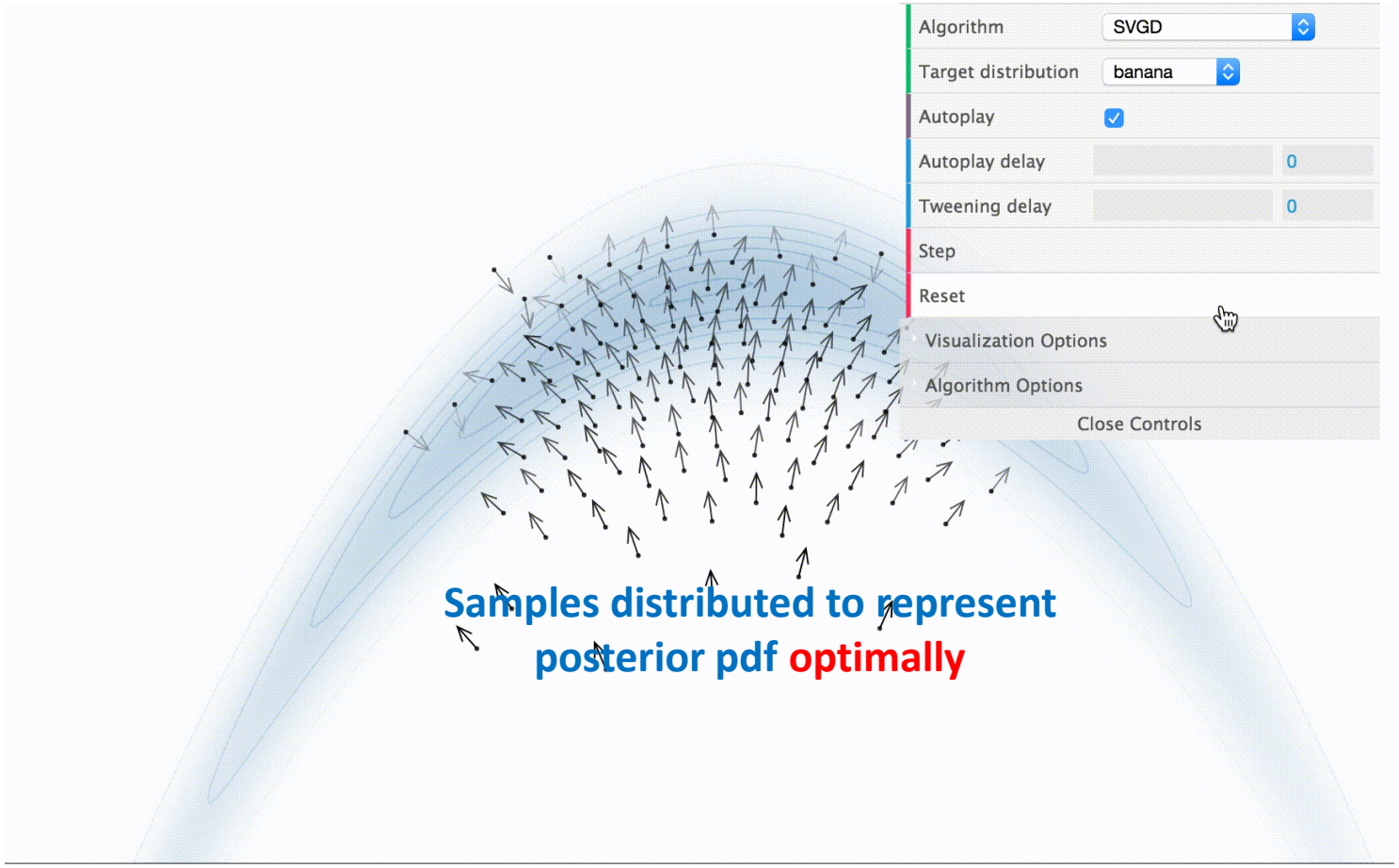
Metropolis-Hastings Monte Carlo



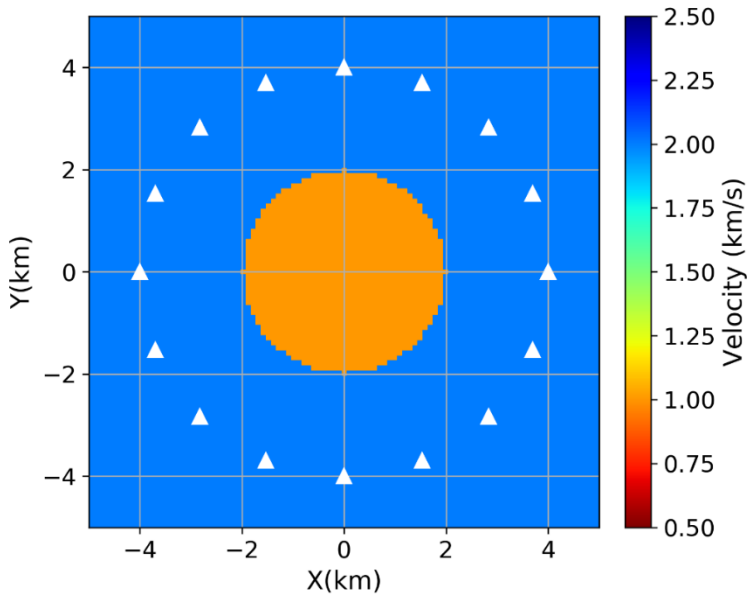
SVGD


Algorithm
SVGD
Target distribution
banana
Autoplay
Autoplay delay
0
Tweening delay
0
Step
Reset
Visualization Options
Algorithm Options
Close Controls

SVGD

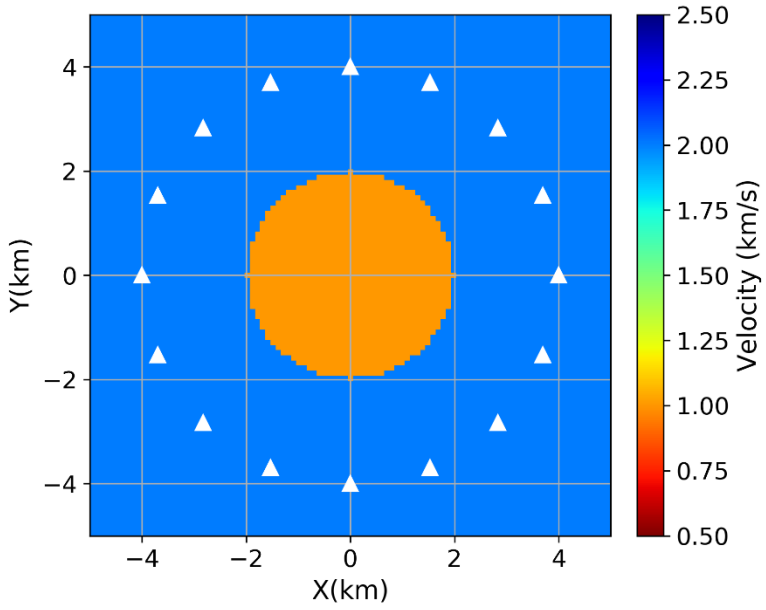

Algorithm
SVGD
Target distribution
banana
Autoplay
Autoplay delay
0
Tweening delay
0
Step
Reset
Visualization Options
Algorithm Options
Close Controls
Samples distributed to represent
posterior pdf optimally

Synthetic tests

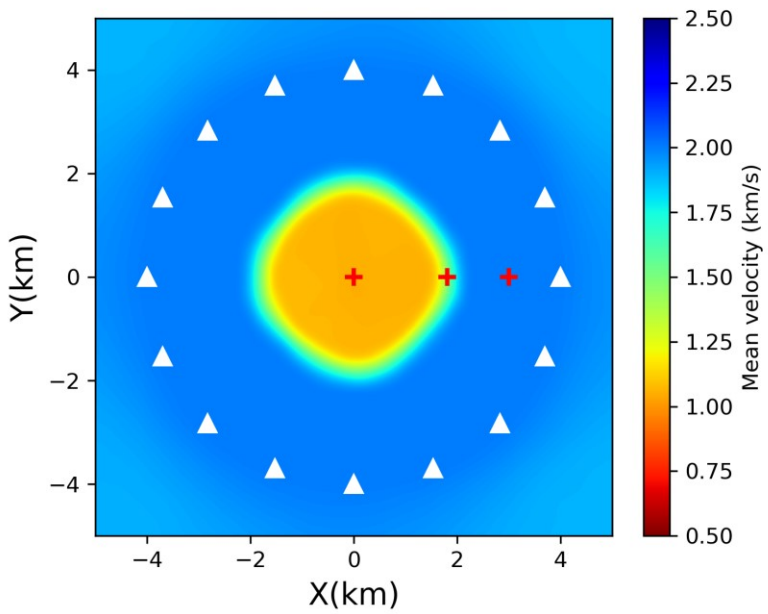


Reversible jump McMC

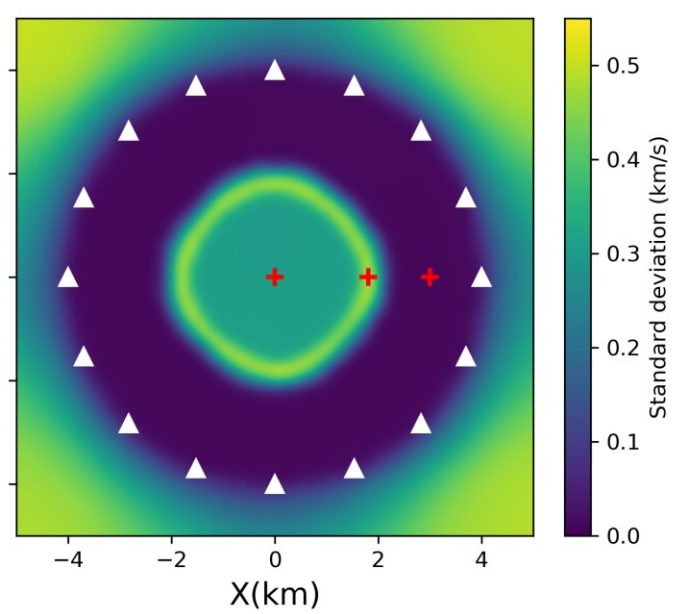
True model



Mean



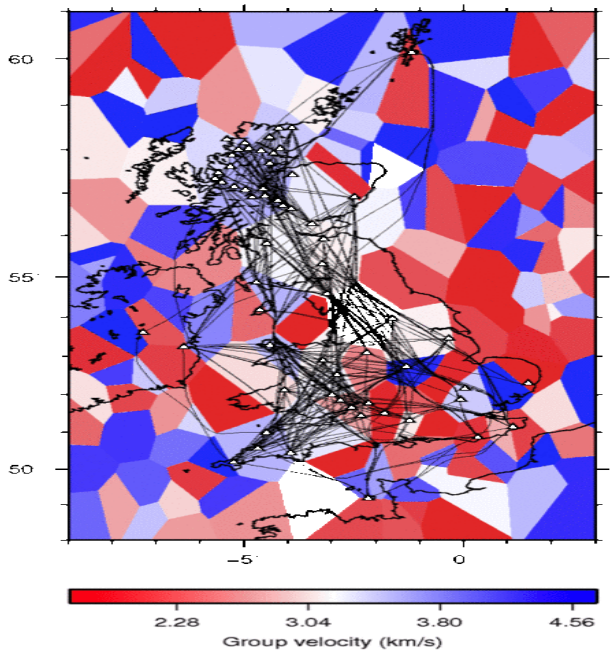
Stdev



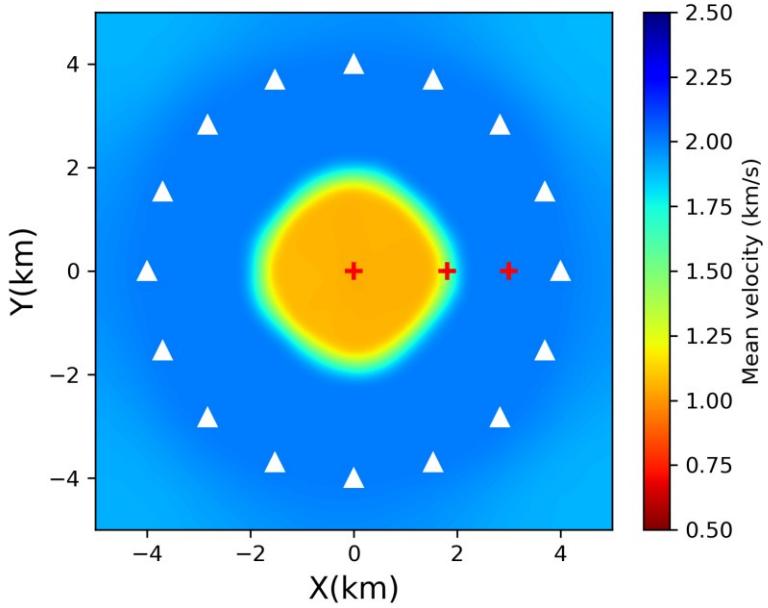
Parameterized by Voronoi cells

Reversible jump McMC

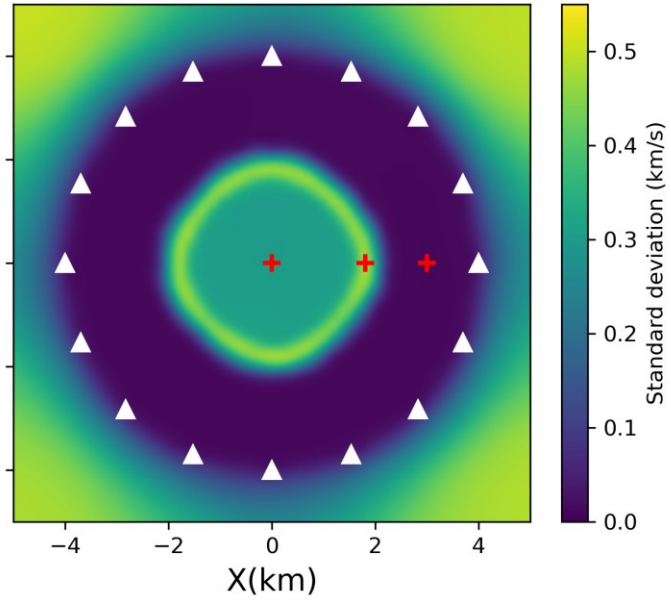
True model



Mean



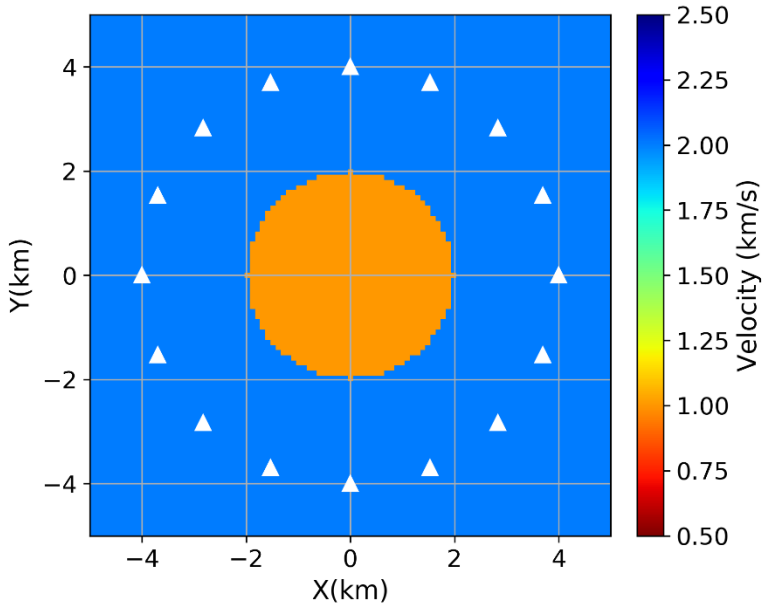
Stdev



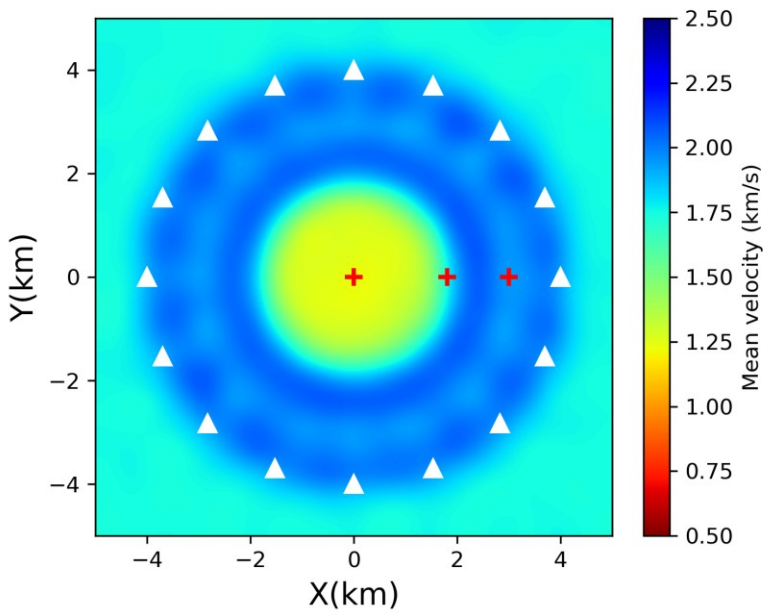
Parameterized by Voronoi cells

ADVI results

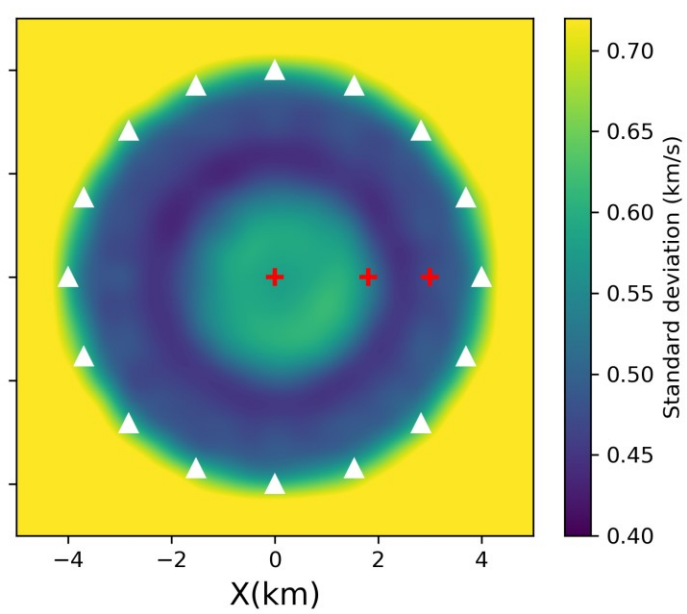
True model



Mean



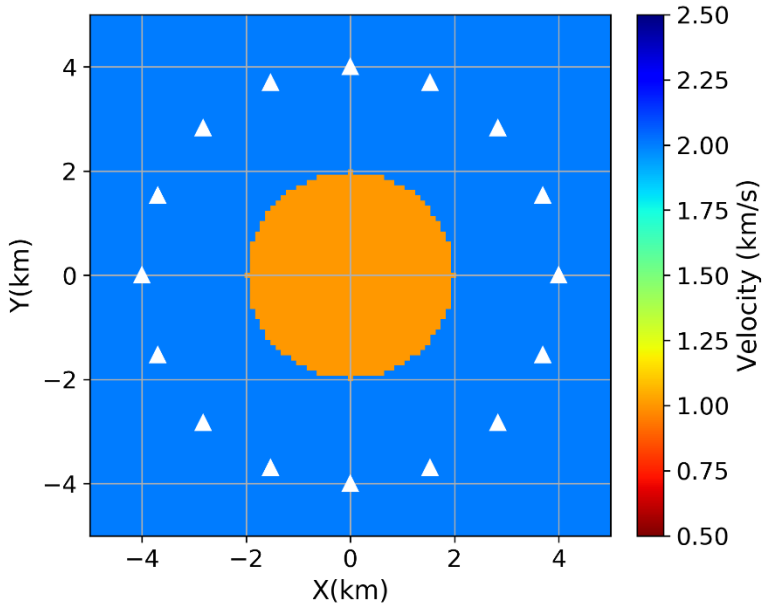
Stdev



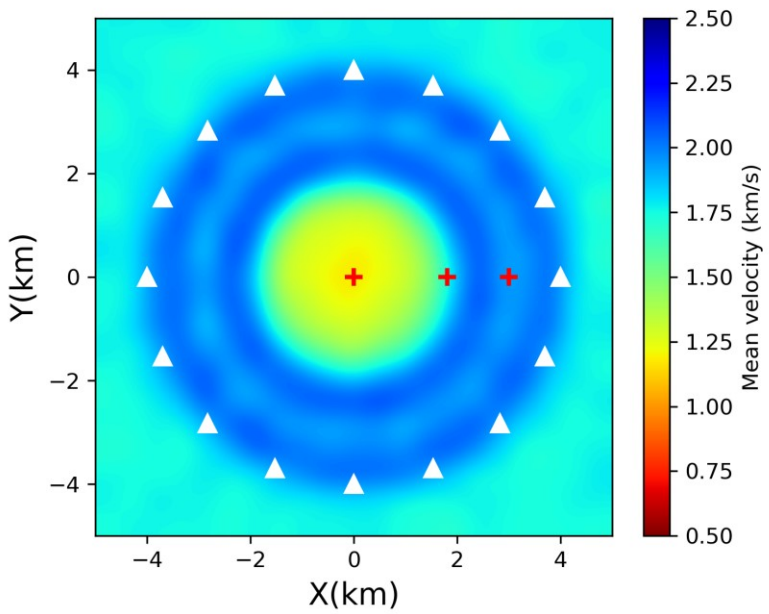
Parameterized by a 21*21 grid

SVGD results

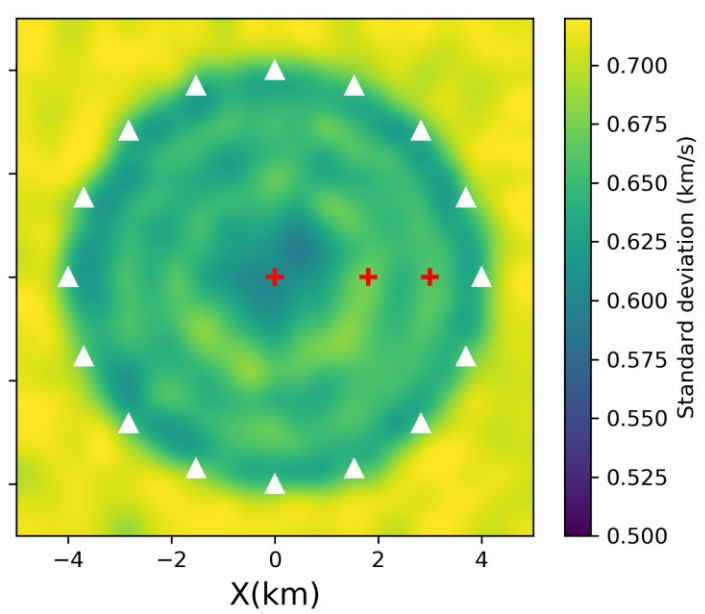
True model



Mean



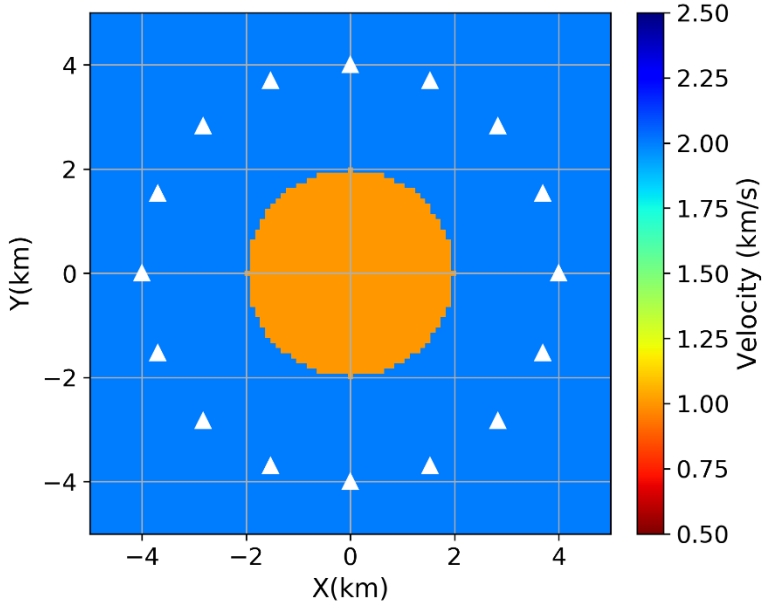
Stdev



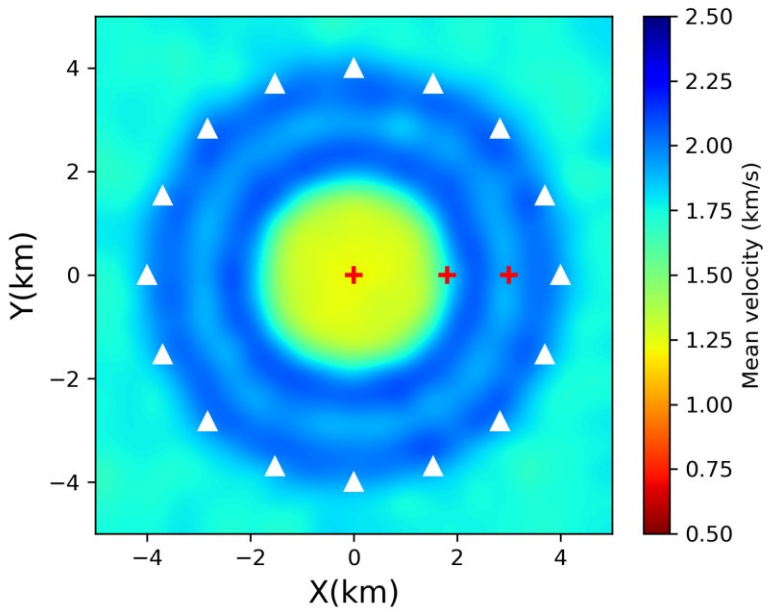
Parameterized by a 21*21 grid

Metropolis-Hastings McMC

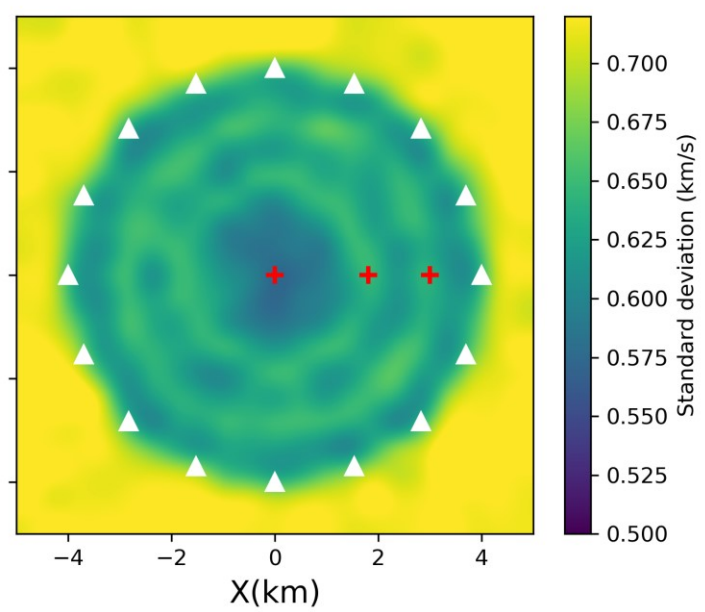
True model



Mean



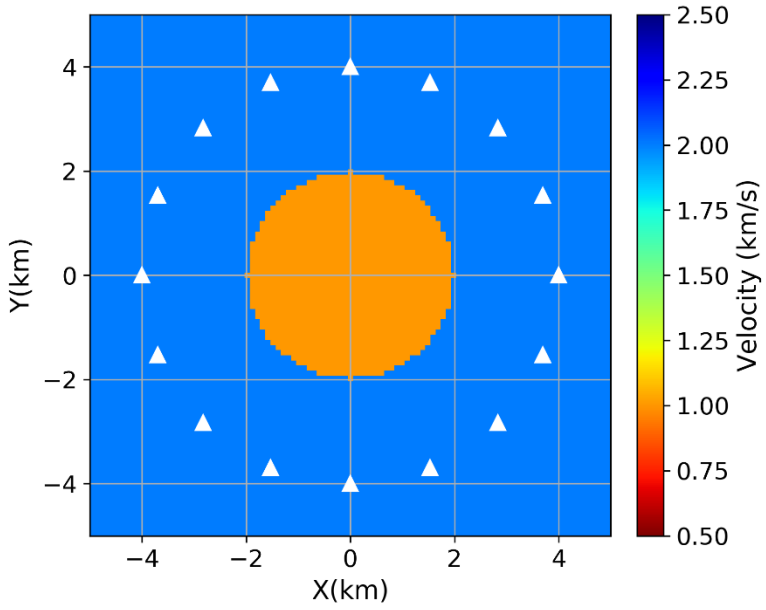
Stdev



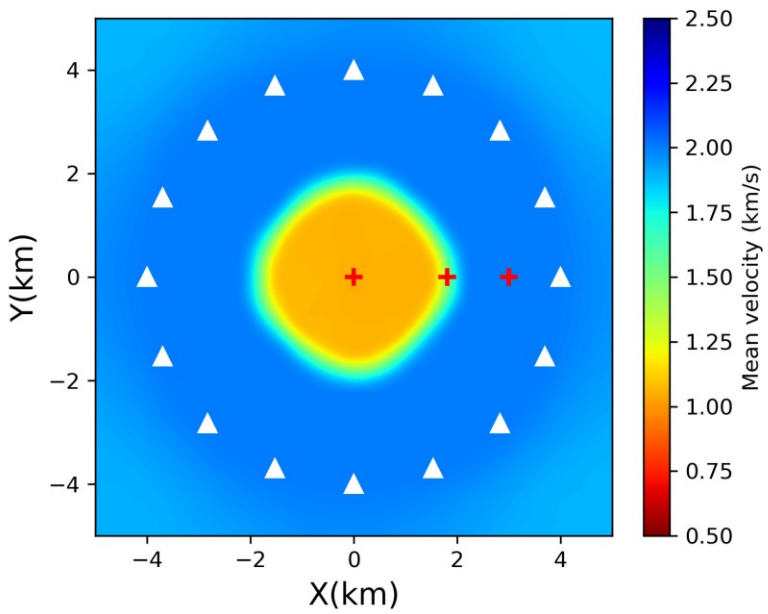
Parameterized by a 21*21 grid

Reversible jump McMC

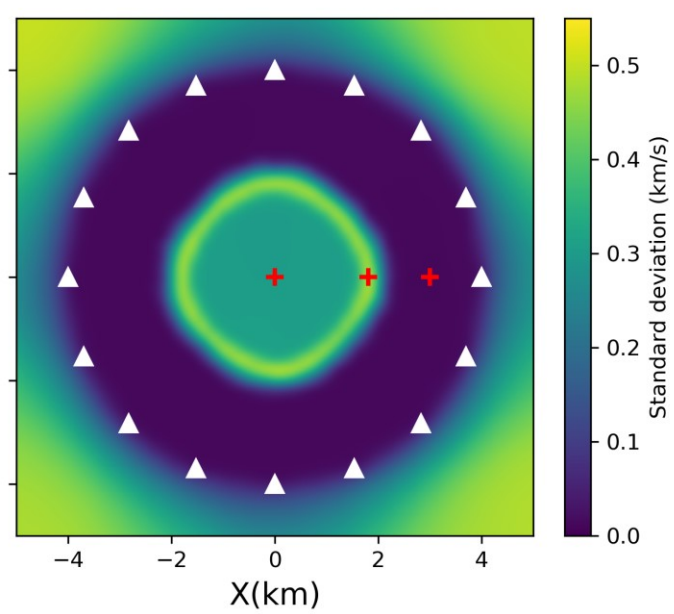
True model



Mean



Stdev



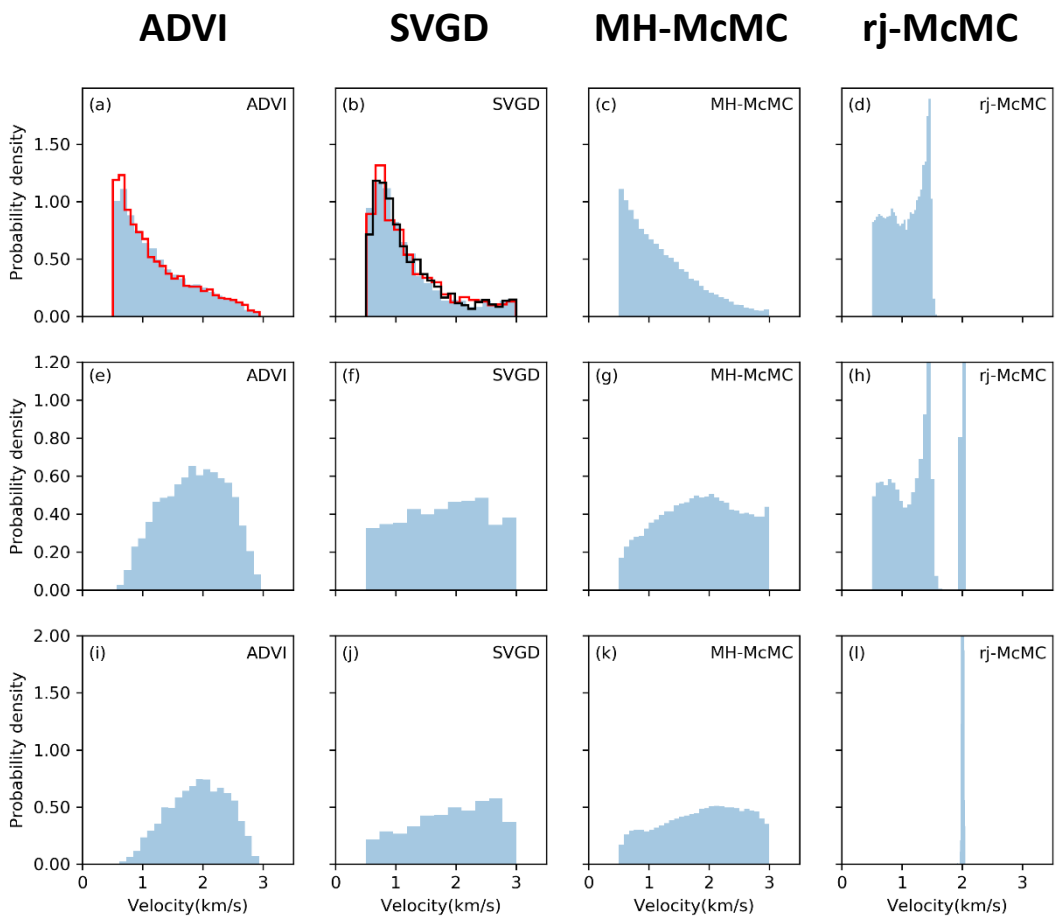
Parameterized by Voronoi cells

Marginal distribution

Interior
Point (0.0,0)

Boundary
Point (1.8,0)

Exterior
Point (3.0,0)



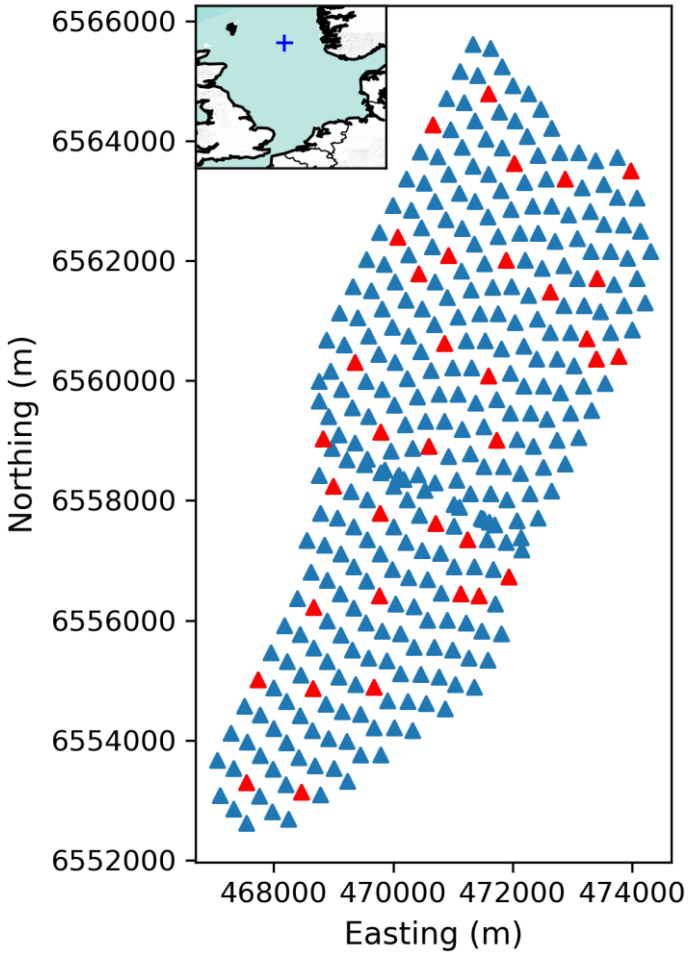
Computational cost

Methods	Number of simulations	CPU hours	Real time (hours)
ADVI	10,000	0.45	0.45
SVGD	400,000	8.53	0.97
MH-McMC	12,000,000	410.3	68.4
Rj-McMC	3,000,000	102.6	17.1

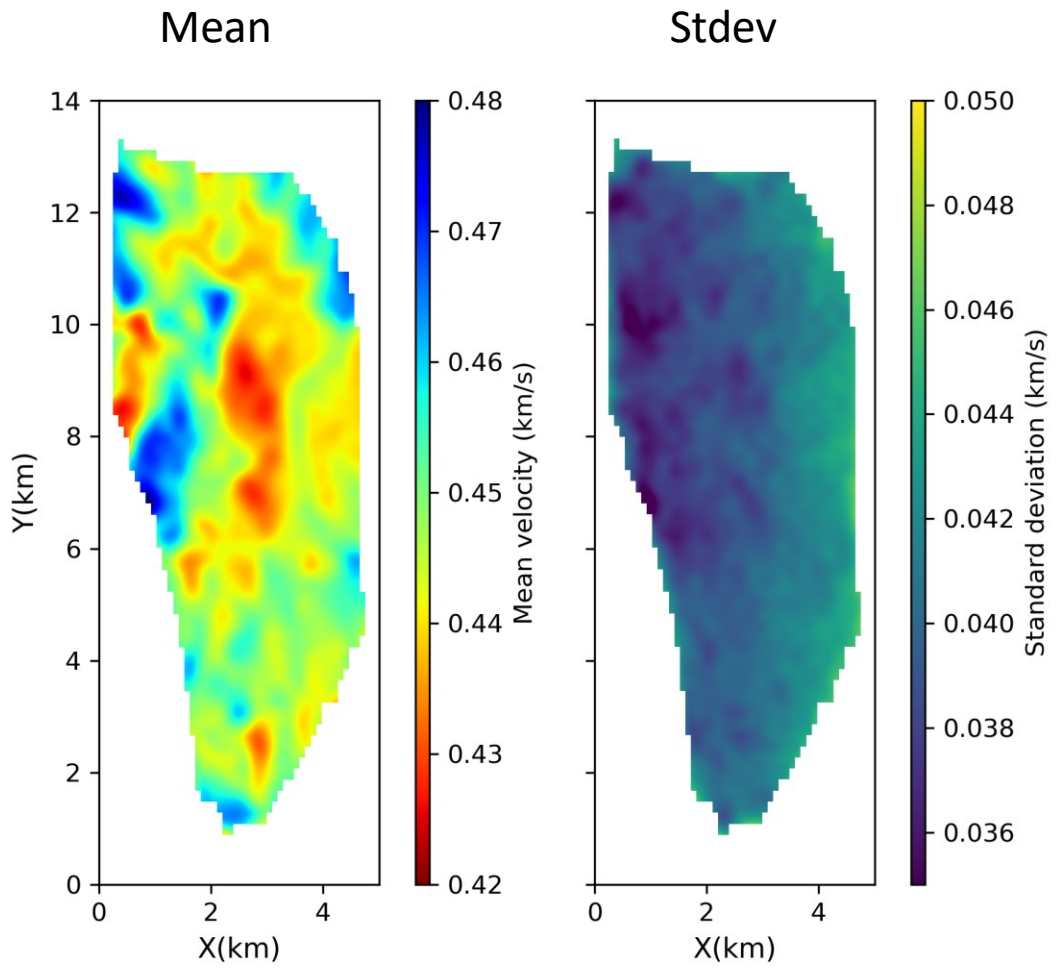
Application to Grane field

346 receivers

35 virtual sources

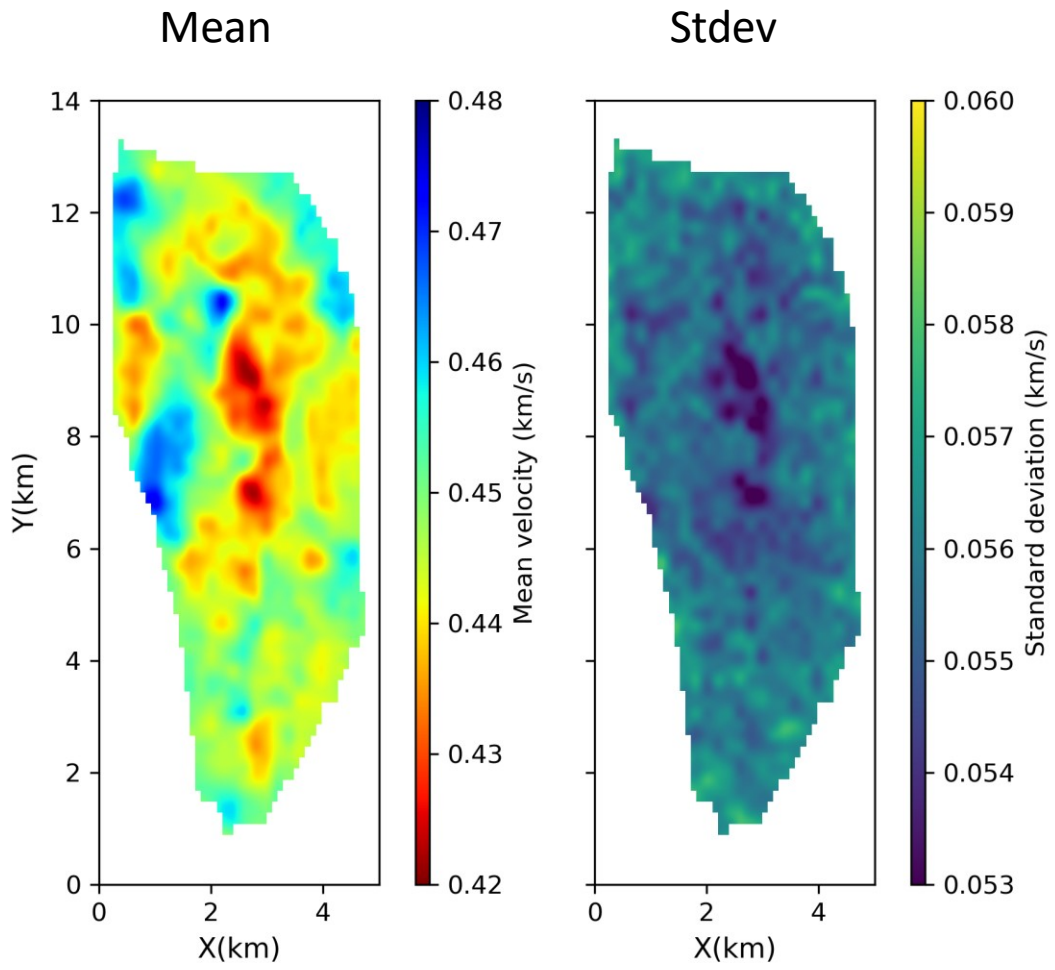


ADVI



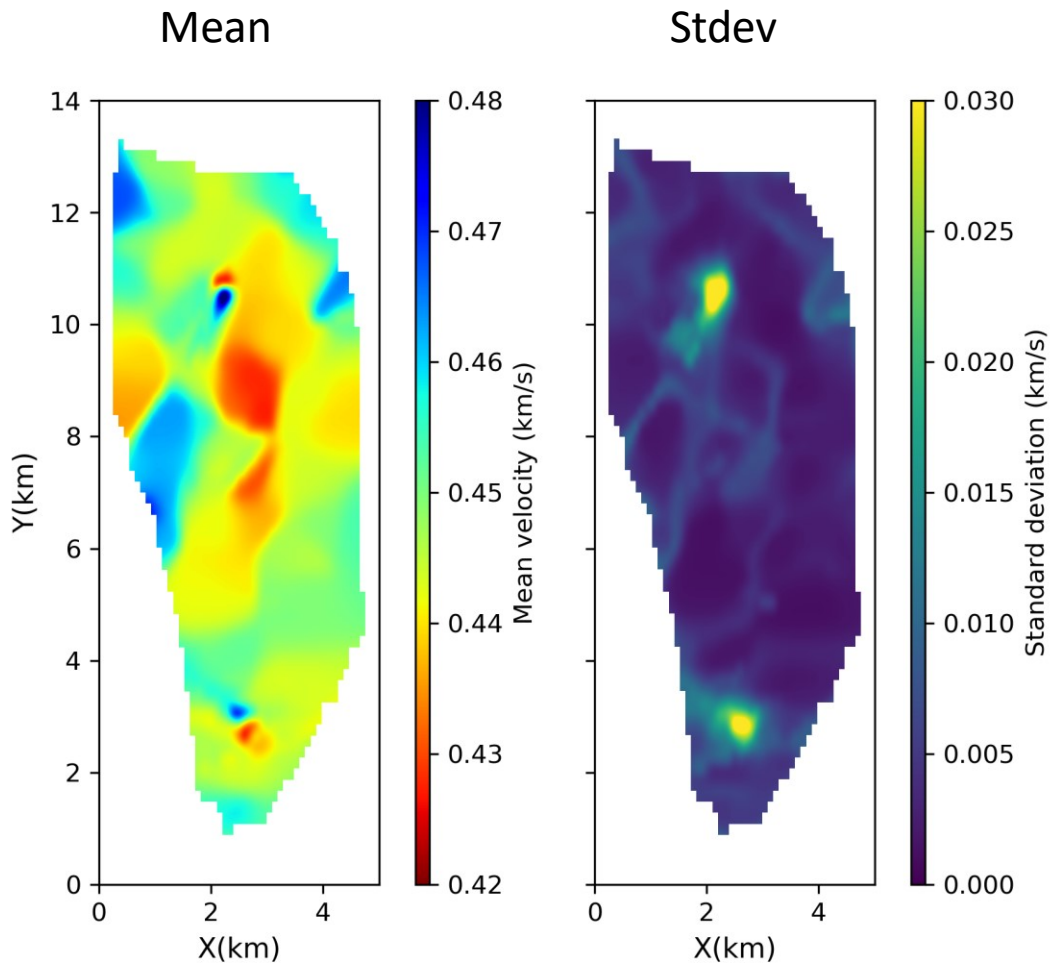
10,000 forward modelling, 5.1 hours

SVGD



500,000 forward modelling, 12.1 hours parallelized using 12 cores

rj-McMC



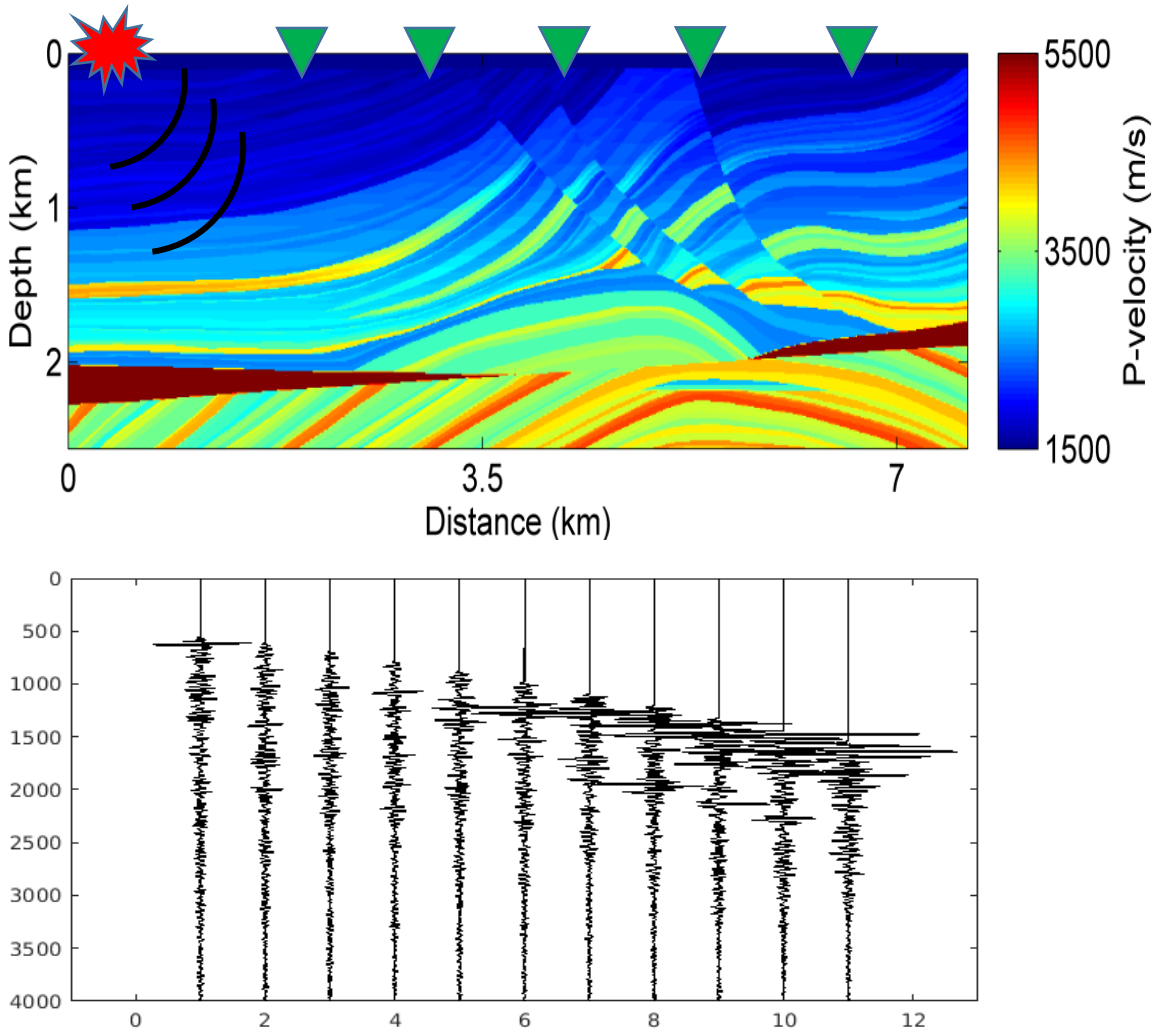
12,800,000 forward modelling, 5 days running on 16 cores

Variational Full-Waveform Inversion

Xin Zhang and Andrew Curtis

ESSOAR, 2020
https://doi.org/10.1002/essoar.10502012.1

Full-waveform Inversion



Forward

Inversion

Choose $\mathbf{m}$ to minimize: $\|F(\mathbf{m}) - \mathbf{d}\|$

Bayesian Solution

- Bayes' theorem

$$p(\mathbf{m} \mid \mathbf{d}_{obs}) = \frac{p(\mathbf{d}_{obs} \mid \mathbf{m})\, p(\mathbf{m})}{p(\mathbf{d}_{obs})}$$

$\mathbf{m}$ = model , $\mathbf{d_{obs}}$ = data

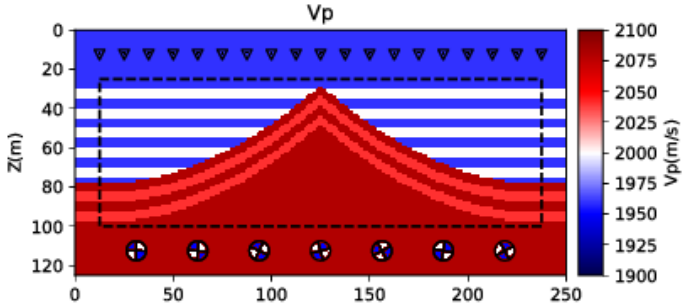
$p(\mathbf{d}_{obs}|\mathbf{m}) \propto \boldsymbol{e}^{-\boldsymbol{\varphi}(\mathbf{m})}$, where $\boldsymbol{\varphi}(\mathbf{m})$ is a misfit function

Synthetic examples

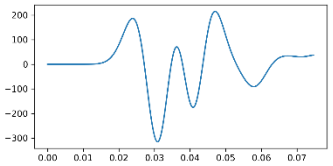
- Fully **elastic** model
- Two component data (x,z)
  7 earthquake sources, 19 receivers
- 180*60*3 free parameters (Vp, Vs, density)
- Uniform priors
  - Vp: 2000 $\pm 100\ m/s$
  - Vs: 800 $\pm 50\ m/s$
  - Density: 1500 $\pm 100\ kg/m^3$

Test case chosen for comparison with Hamiltonian-MC study (*Gebraad et al., 2019*)

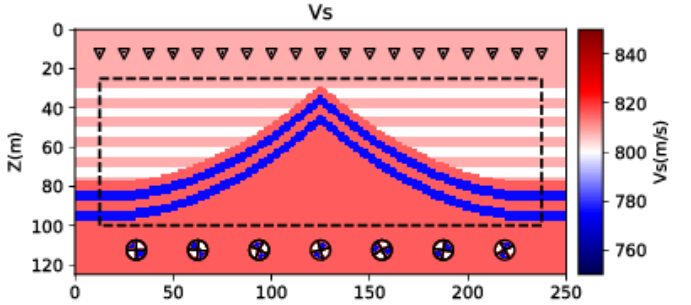
Vp



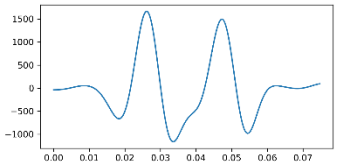
50 Hz



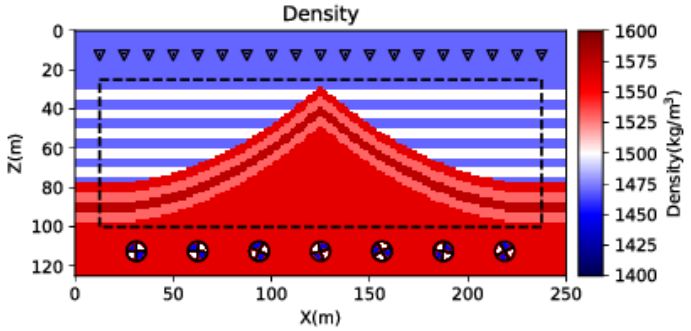
Vs



50 Hz



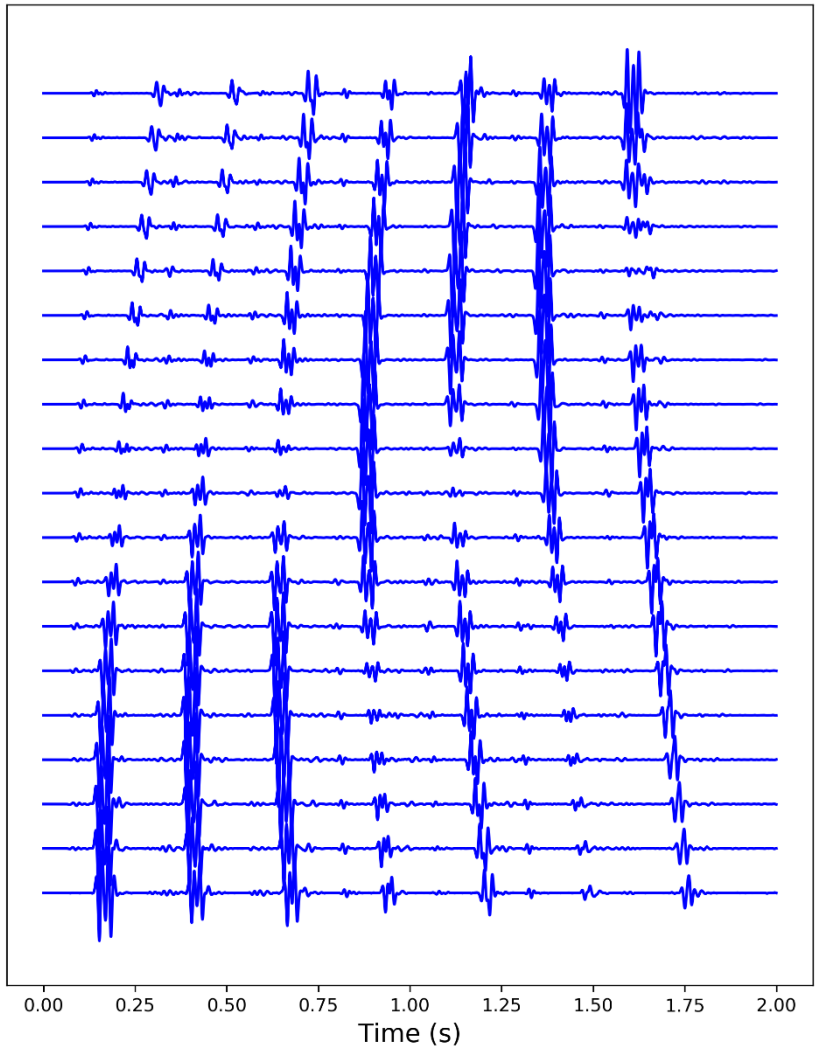
Density



Gebraad et al., 2019 EarthArXiv

Synthetic examples

X-component Data



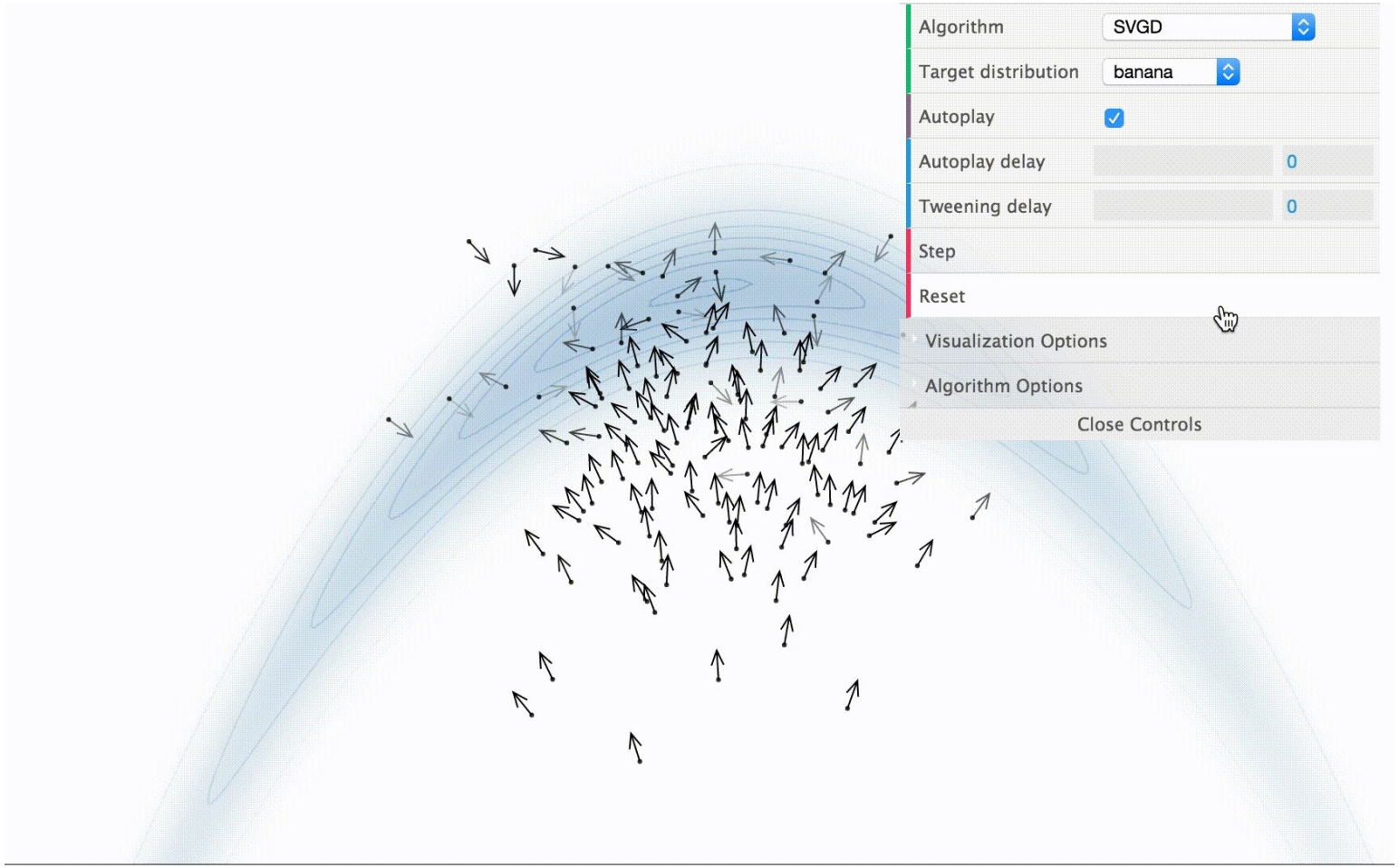
Likelihood: (L2 norm in time domain)

$$p(\mathbf{d}_{obs}|\mathbf{m}) \propto \boldsymbol{e}^{-\boldsymbol{\varphi}(\mathbf{m})}$$

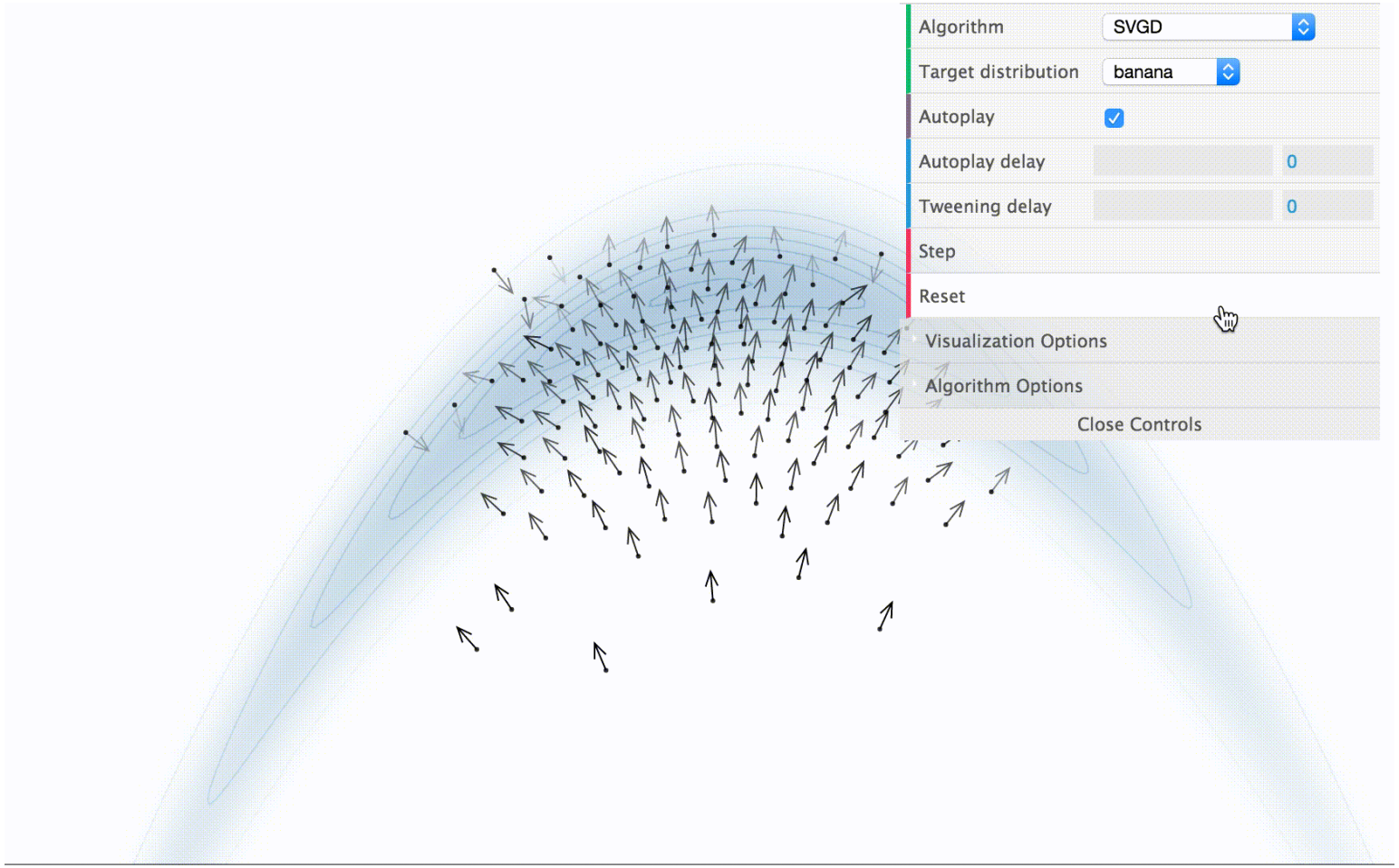
$$\varphi(\mathbf{m}) = \frac{1}{2}\sum_{i,j}\left(\frac{d_{ij,obs} - d_{ij}(\mathbf{m})}{\sigma_{ij}}\right)^2$$

where *i* is receiver number and *j* denotes time samples and $\sigma_{ij} = 1$.

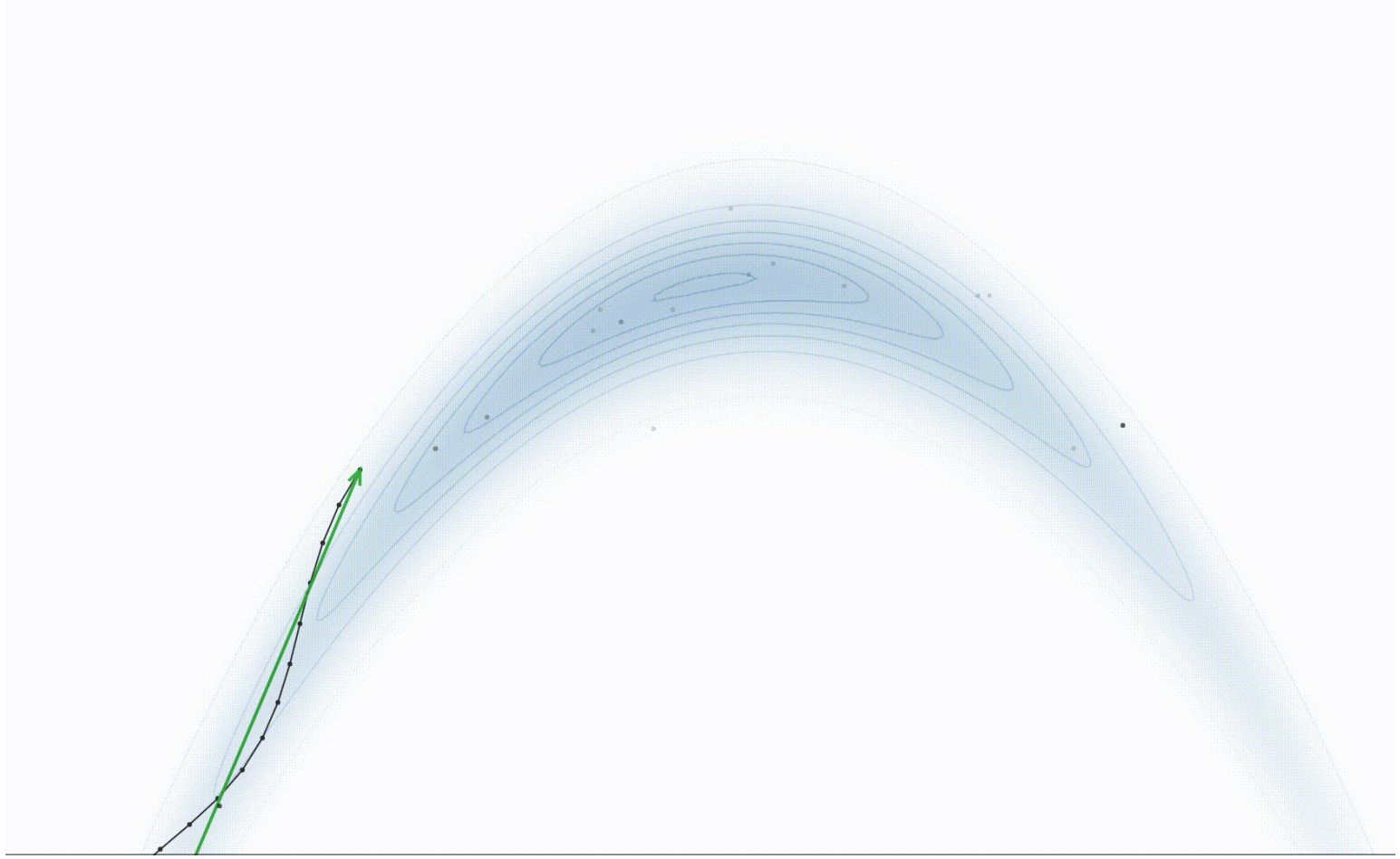
SVGD


Algorithm
SVGD
Target distribution
banana
Autoplay
Autoplay delay
0
Tweening delay
0
Step
Reset
Visualization Options
Algorithm Options
Close Controls

SVGD

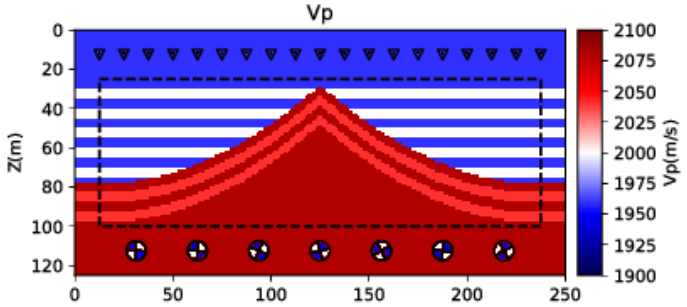

Algorithm
SVGD
Target distribution
banana
Autoplay
Autoplay delay
0
Tweening delay
0
Step
Reset
Visualization Options
Algorithm Options
Close Controls

Hamiltonian Monte Carlo

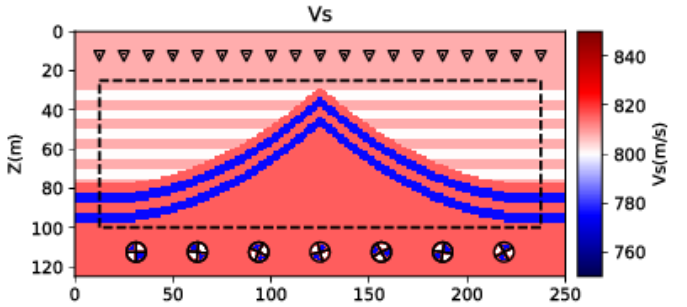


Synthetic examples

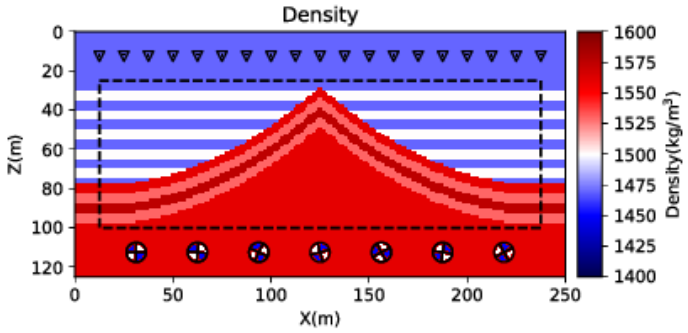
Vp



Vs



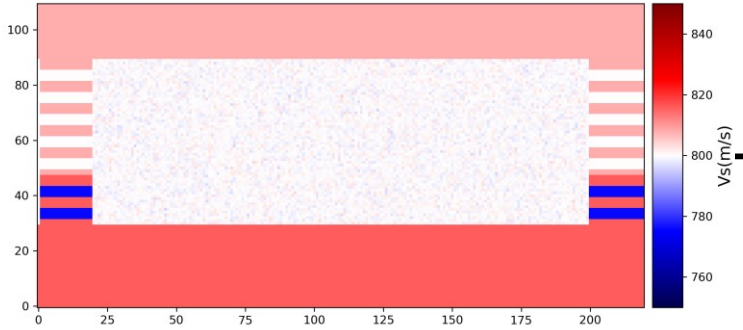
Density



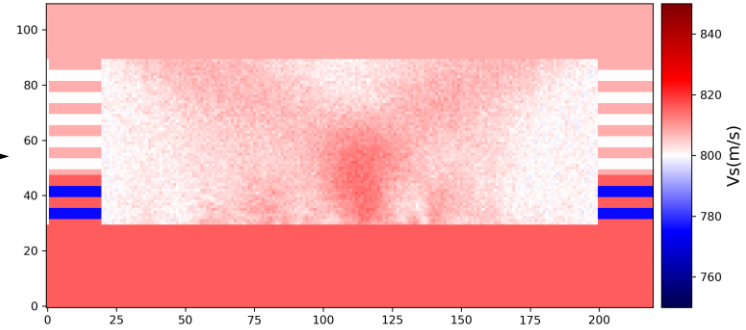
Gebraad et al., 2019 EarthArXiv

Iterations of SVGD

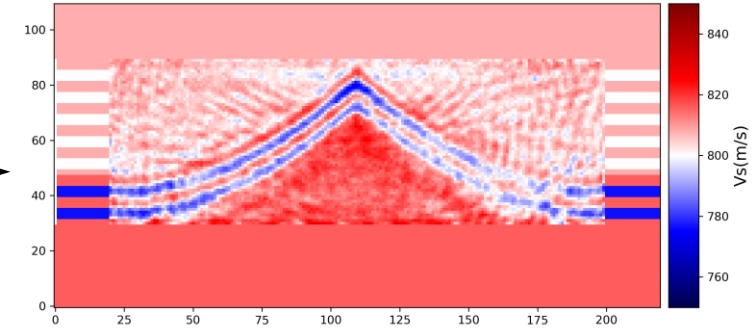
Prior Vs mean



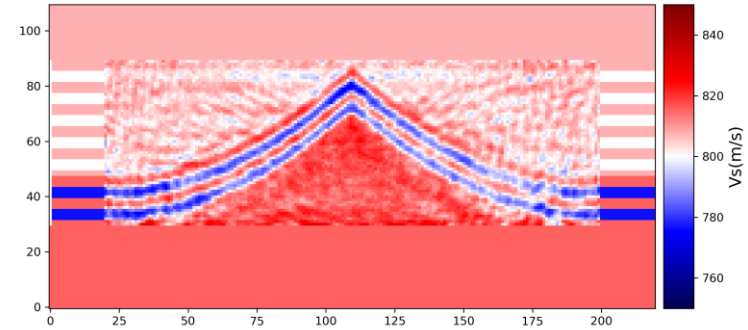
Vs mean of iteration 100



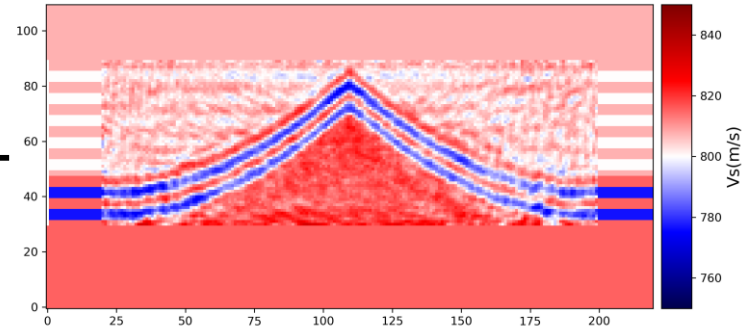
Vs mean of iteration 200



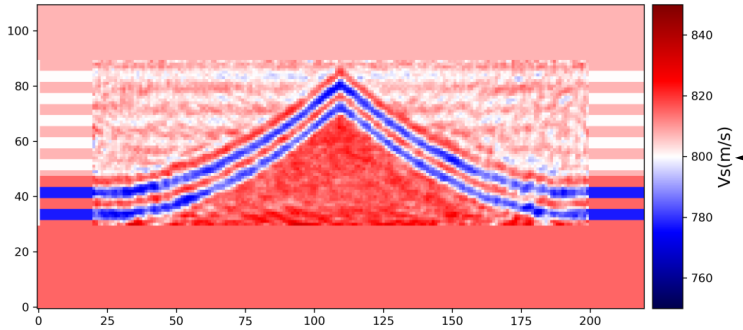
Vs mean of iteration 300



Vs mean of iteration 400



Vs mean of iteration 600

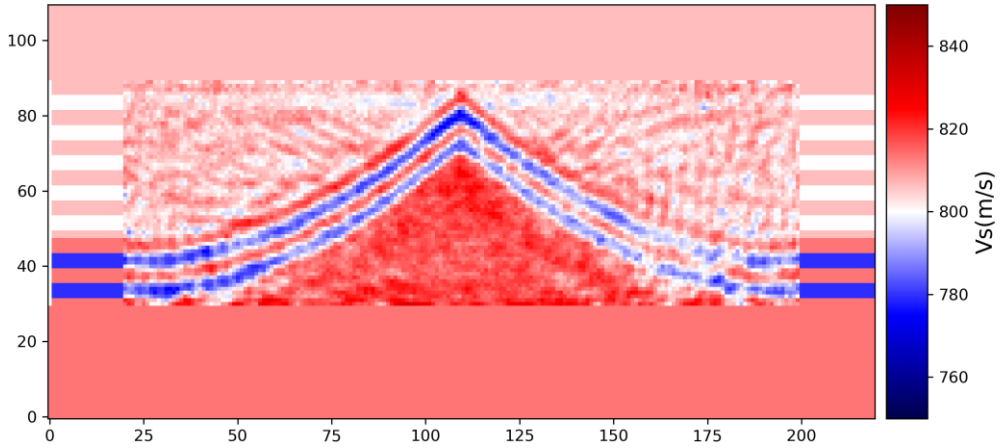


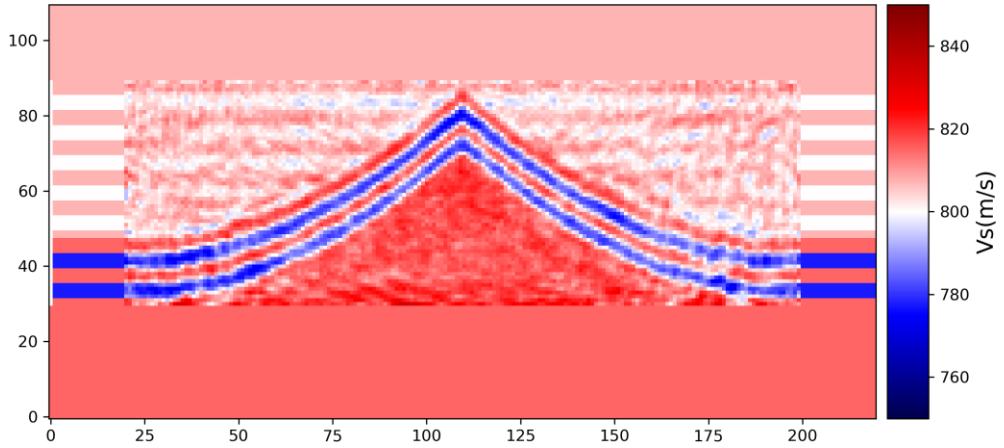
Number of particles

400 particles

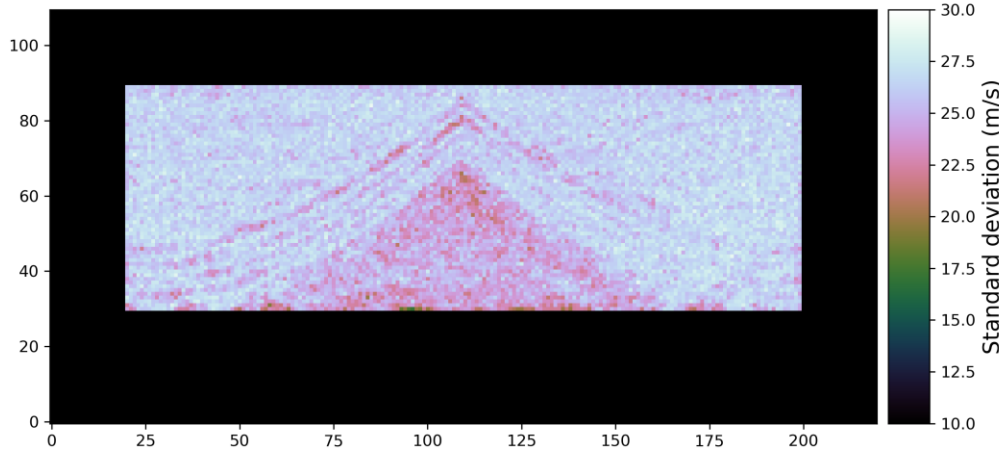
600 particles

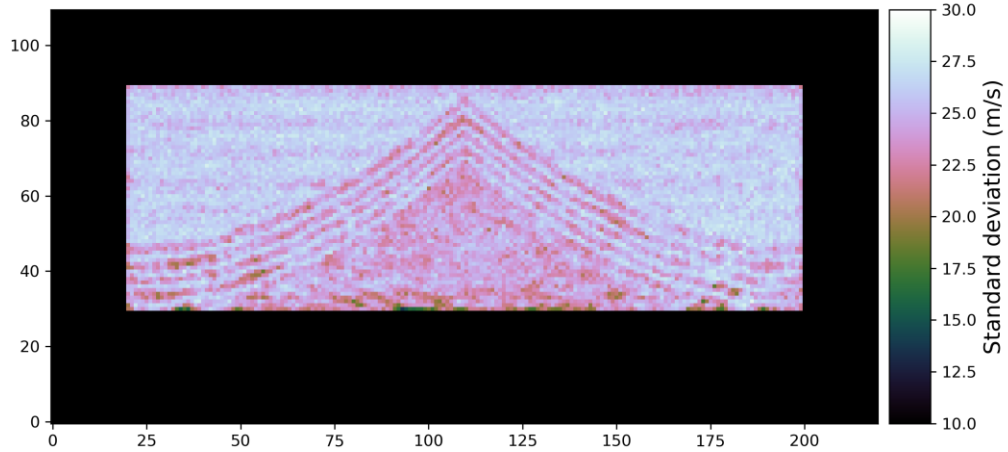
Mean Vs



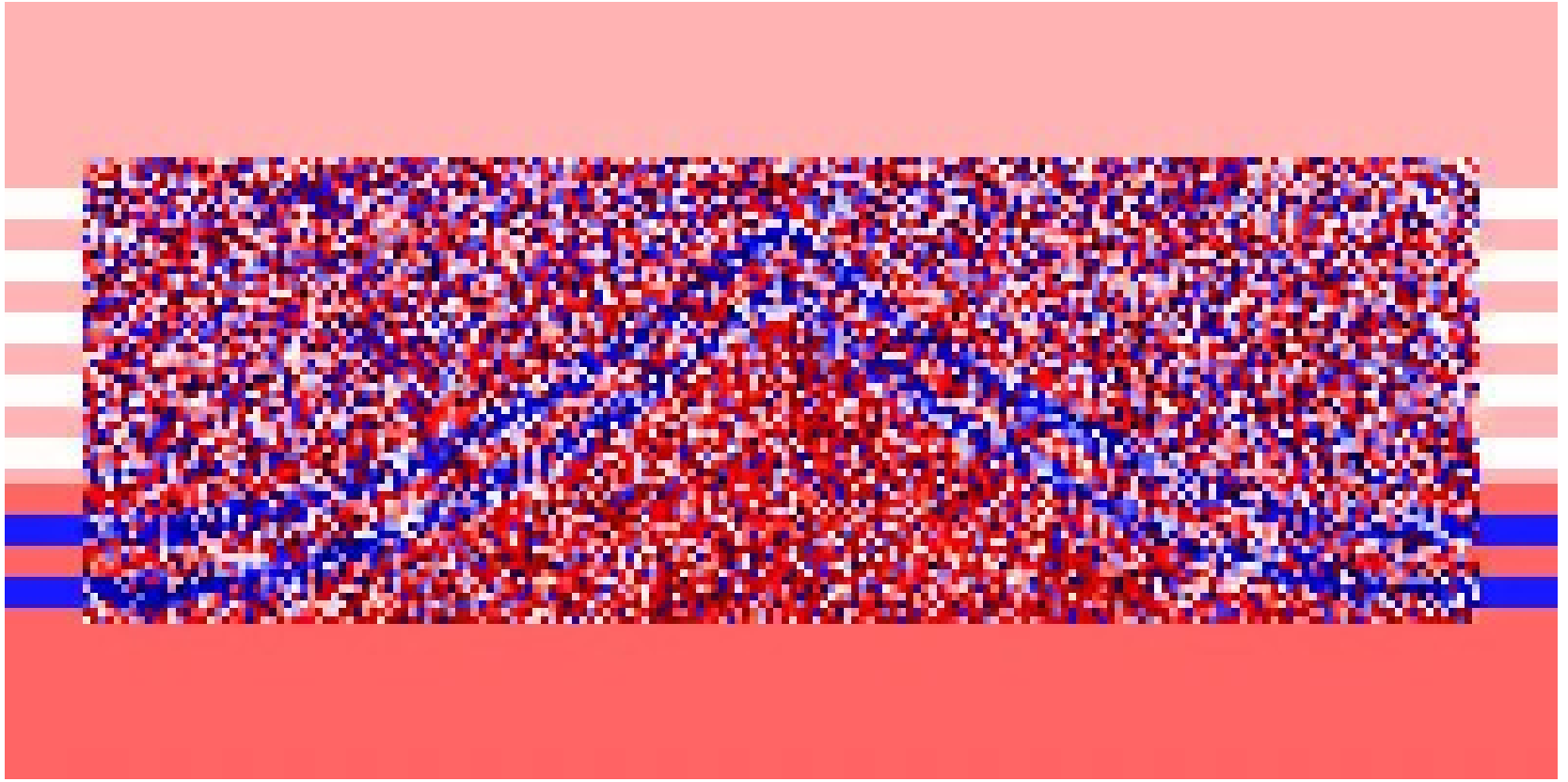


Stdev Vs

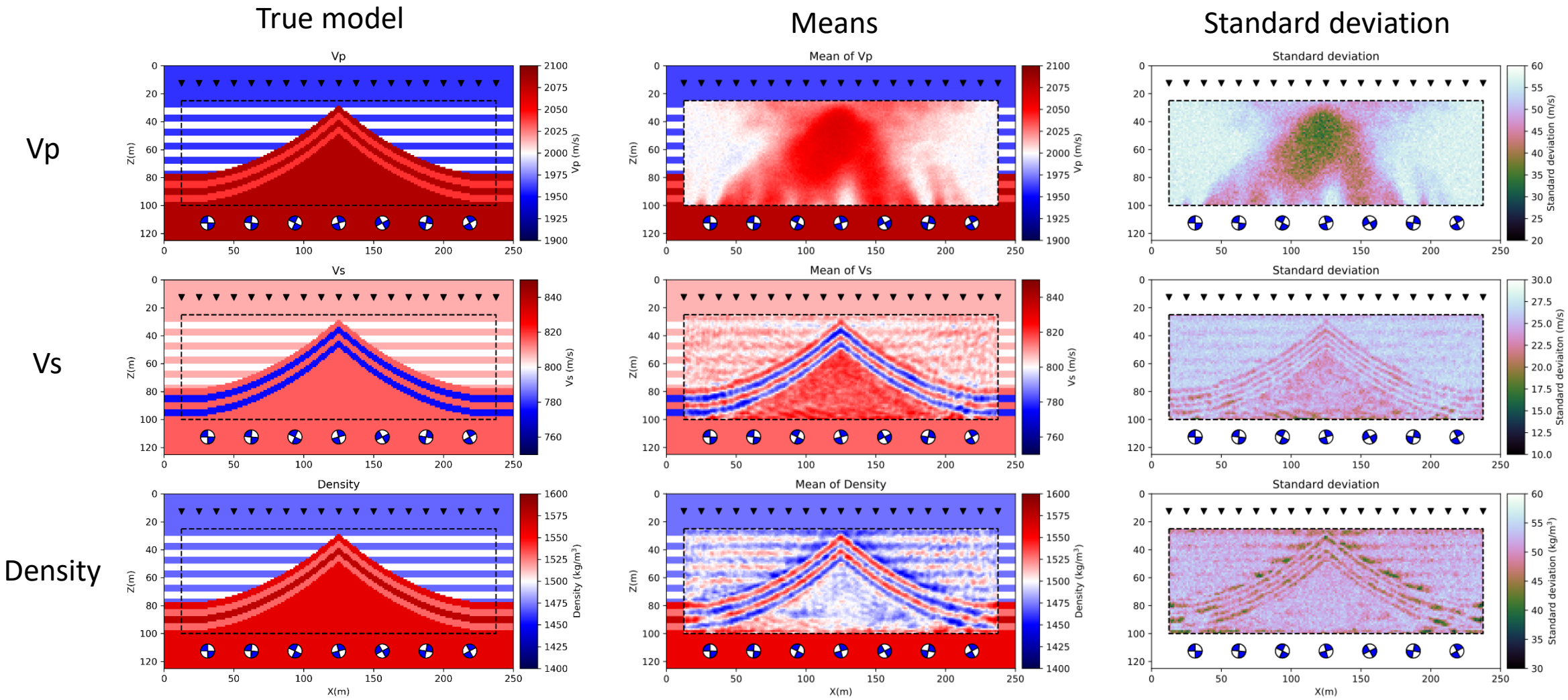




Particles of SVGD



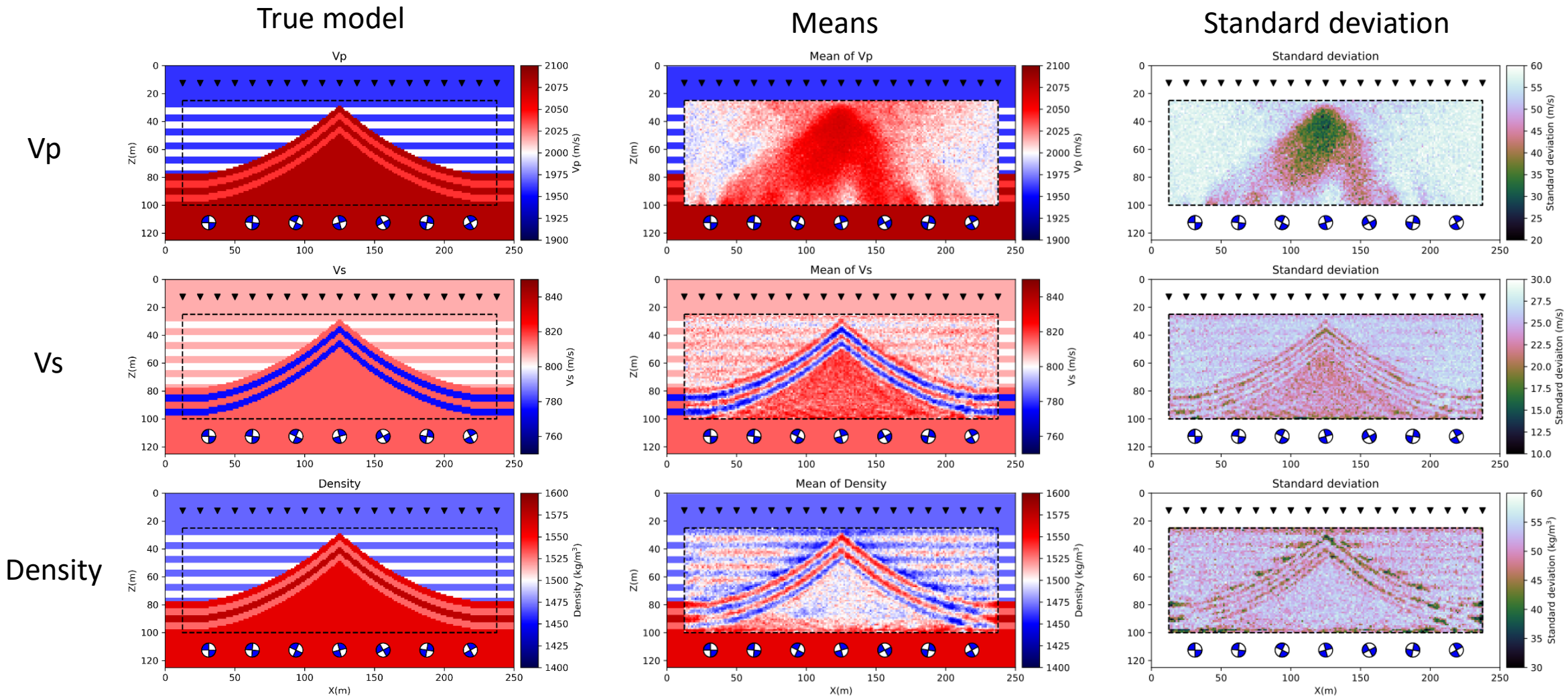
Results of SVGD



600 particles, 600 iterations, parallelized using 16 cores

Zhang & Curtis, 2019 ESSOAR

Results of Hamiltonian Monte Carlo

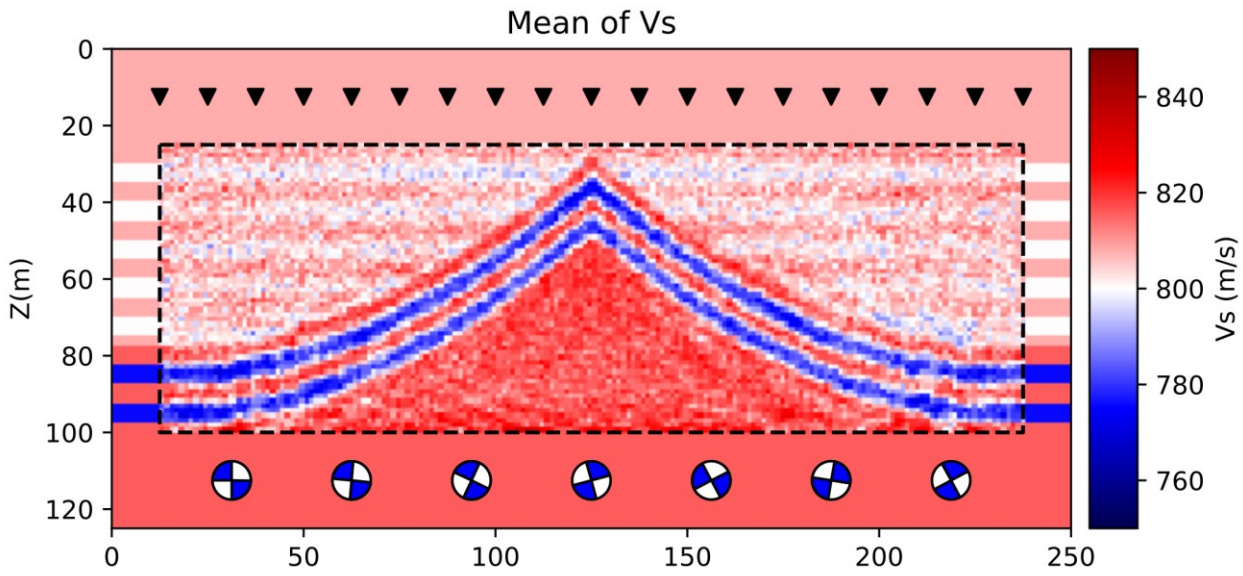


1 chain, about 100,000 simulations, not parallelized

Gebraad et al., 2019 EarthArXiv

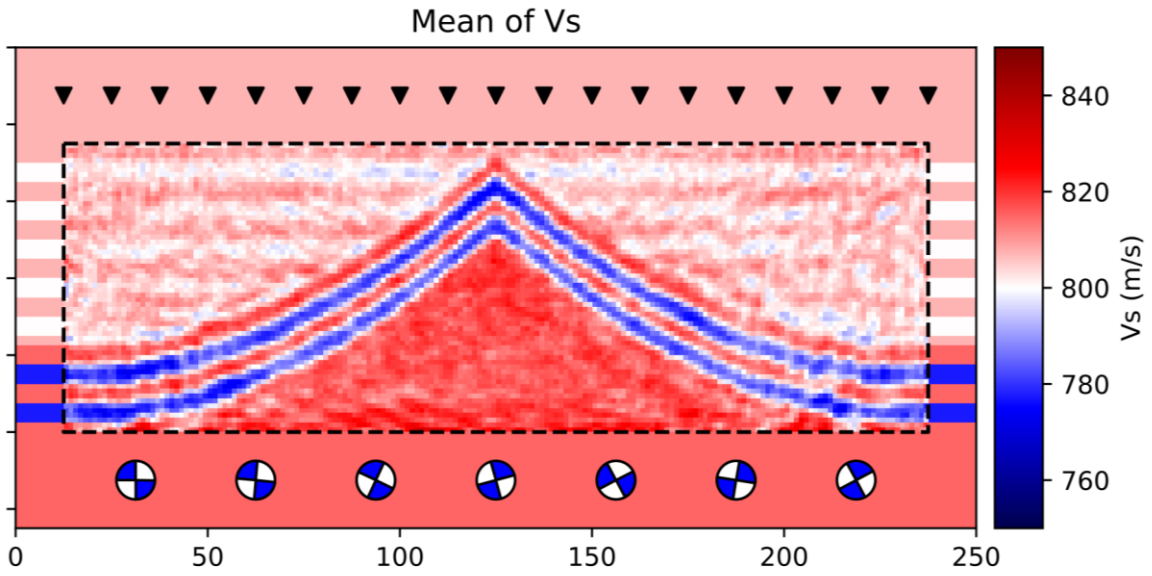
Comparison

Mean Vs from **HMC**



10,000 samples

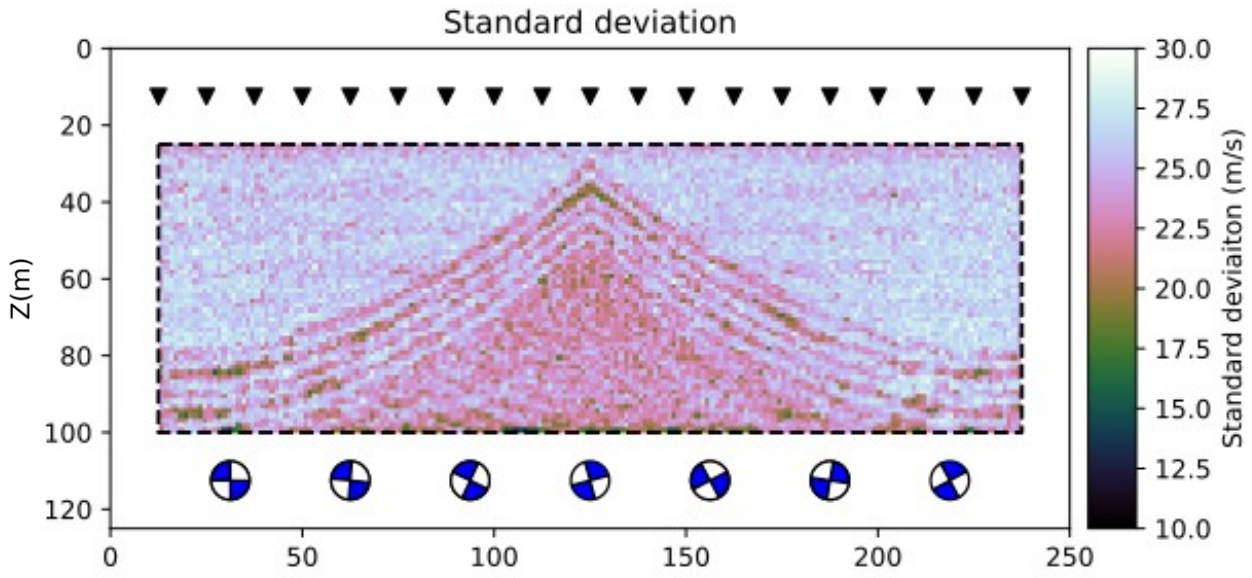
Mean Vs from **SVGD**



600 samples

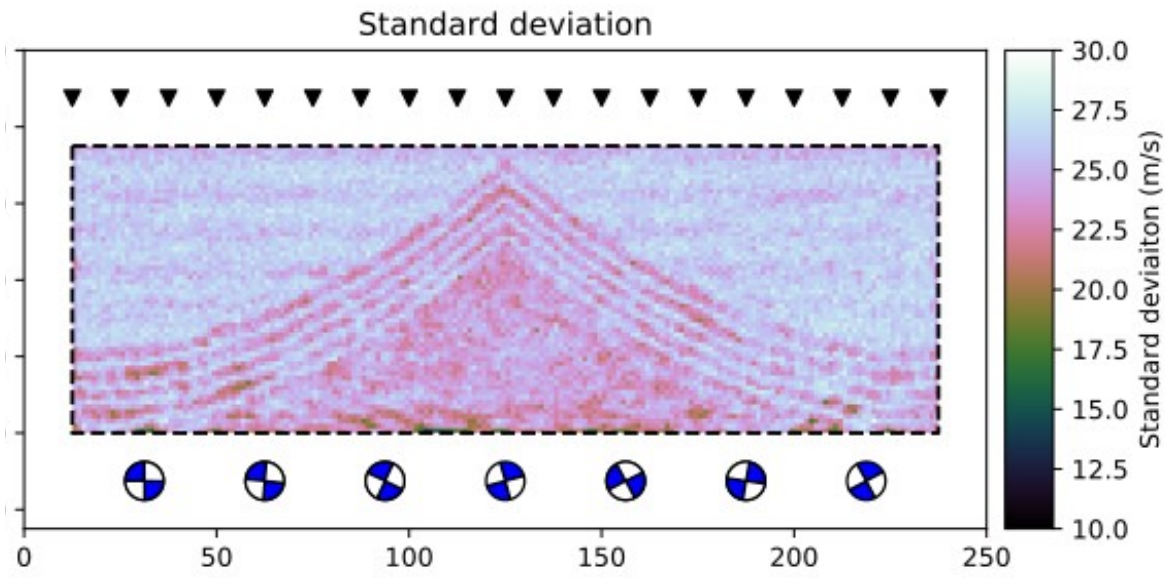
Comparison

Stdev Vs from **HMC**



10,000 samples

Stdev Vs from **SVGD**



600 samples

Optimally Selected to Represent Posterior pdf

Comparison between SVGD and HMC

SVGD	**HMC**
Gradient based	**Gradient based**
Optimisation	Sampling
Accurate (converges asymptotically)	Accurate (converges asymptotically)
Easily parallelized	Cannot be fully parallelized within one chain
Can be applied to large data sets - minibatch optimisation	Cannot use minibatches
Easy to tune	Hard to tune

Comparison between SVGD and HMC

SVGD	**HMC**
Gradient based	Gradient based
Optimisation	**Sampling**
Accurate (converges asymptotically)	Accurate (converges asymptotically)
Easily parallelized	Cannot be fully parallelized within one chain
Can be applied to large data sets - minibatch optimisation	Cannot use minibatches
Easy to tune	Hard to tune

Comparison between SVGD and HMC

SVGD	**HMC**
Gradient based	Gradient based
Optimisation	Sampling
Accurate (converges asymptotically)	**Accurate (converges asymptotically)**
Easily parallelized	Cannot be fully parallelized within one chain
Can be applied to large data sets - minibatch optimisation	Cannot use minibatches
Easy to tune	Hard to tune

Comparison between SVGD and HMC

SVGD	HMC
Gradient based	Gradient based
Optimisation	Sampling
Accurate (converges asymptotically)	Accurate (converges asymptotically)
Easily parallelized	**Cannot be fully parallelized within one chain**
Can be applied to large data sets - minibatch optimisation	Cannot use minibatches
Easy to tune	Hard to tune

Comparison between SVGD and HMC

SVGD	**HMC**
Gradient based	Gradient based
Optimisation	Sampling
Accurate (converges asymptotically)	Accurate (converges asymptotically)
Easily parallelized	Cannot be fully parallelized within one chain
Can be applied to large data sets - minibatch optimisation	**Cannot use minibatches**
Easy to tune	Hard to tune

Comparison between SVGD and HMC

SVGD	**HMC**
Gradient based	Gradient based
Optimisation	Sampling
Accurate (converges asymptotically)	Accurate (converges asymptotically)
Easily parallelized	Cannot be fully parallelized within one chain
Can be applied to large data sets - minibatch optimisation	Cannot use minibatches
Easy to tune	**Hard to tune**

Conclusion

- Applied Stein variational gradient descent (SVGD) to full waveform inversion

- Compared the results with HMC
  - SVGD provides accurate approximations to the results of HMC
    (although HMC has not converged – only used 100,000 samples – neither is ‘correct’)

- SVGD can be applied to large data sets
  - Parallelization
  - Minibatches
  - Provides optimal samples to represent posterior pdf


THE UNIVERSITY
of EDINBURGH


eip
edinburgh imaging project

Thank you

Summary

- Introduced two variational inference methods to seismic tomography
  - Automatic differential variational inference (ADVI)
  - Stein variational gradient descent (SVGD)

- Compared with Metropolis-Hastings and reversible jump McMC
  - Variational methods provide efficient alternatives to McMC

- Variational methods almost unexplored in Geophysics – **Try them!**

➔**FOR PAPERS ON ALL TOPICS SEE: www.ed.ac.uk/homes/acurtis**

➔**Or email Andrew.Curtis@ed.ac.uk**