Uncertainty quantification in geomodeling by Hessian informed MCMC

Zhouji Liang[1,2]   Prof. Florian Wellmann [1]

[1]Computational Geoscience and Reservoir Engineering(CGRE), RWTH Aachen University
[2]International Research Training Group – Modern Inverse Problems (IRTG-MIP)

Computational Geoscience and Reservoir Engineering | MIP Modern Inverse Problems | RWTH AACHEN UNIVERSITY

Summary Page

Abstract: Uncertainty quantification is an important aspect of geological modelling and model interpretation. Recent developments in geological modelling allow us to view the inversion as a problem in Bayesian inference, incorporating the uncertainties in the observations, the forward models and the prior knowledge from geologists. The sampling method Markov chain Monte Carlo (MCMC) is then often applied to solve this inference problem.

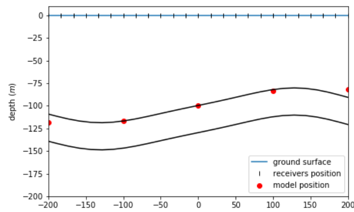
Methods:

- The presented work is about applying recent developed Hessian informed MCMC method to accelerate the uncertainty quantification during geomodelling.
- We utilize state-of-the-art differentiable programming to efficiently calculate the gradient and Hessian of the negative log posterior respect to the parameters of interest.
- We implement Adaptive Moment Estimation (Adam) to find the Maximum a Posterior point(MAP), and then, construct the Laplace approximation of the posterior at MAP to achieve more efficient posterior search.

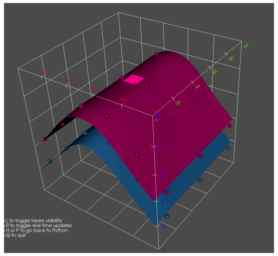
Results: A gravity inversion is conducted in this study. The sampling chain by the novel Hessian informed MCMC shows a better mixing and lower autocorrelation, which demonstrate a more efficient estimation of the uncertainties.

Methods[1]	Computation time for 10000 samples
RMH	71.60s
HMC	215.10s
gpCN	Finding MAP by Adam: 61.89s+Hessian Calculation[2]: 0.43s +run chain:38.90s = Total time cost: 101.22 s

2D model example



3D model example

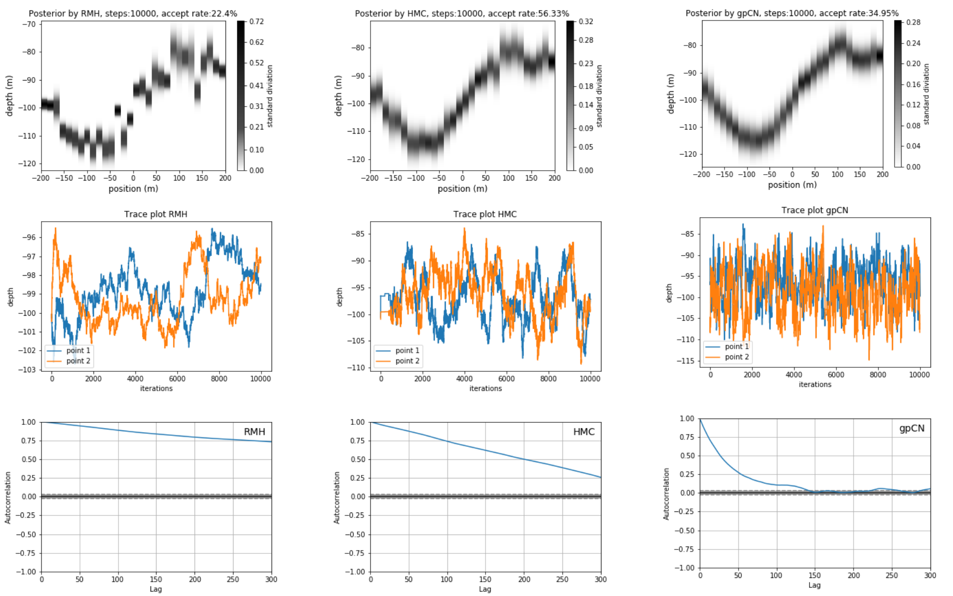


software: TensorFlow, GemPy

a) Random walk

b) HMC

c) our approach gpCN



Computational Geoscience and Reservoir Engineering | Modern Inverse Problems | RWTH AACHEN UNIVERSITY

Outline

Motivation

Dimension independent MCMC

Automatic Differentiation

Application / Synthetic test cases

Results

Discussion

Summary

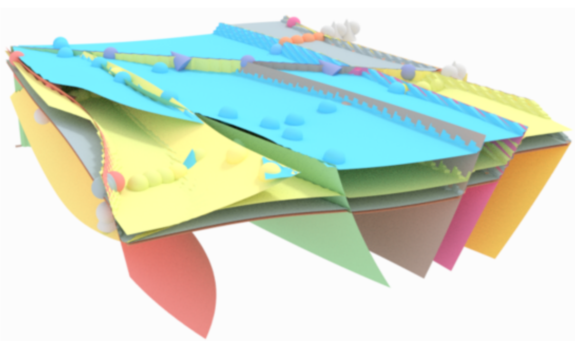
Uncertainties in Geomodeling

Difficulties:

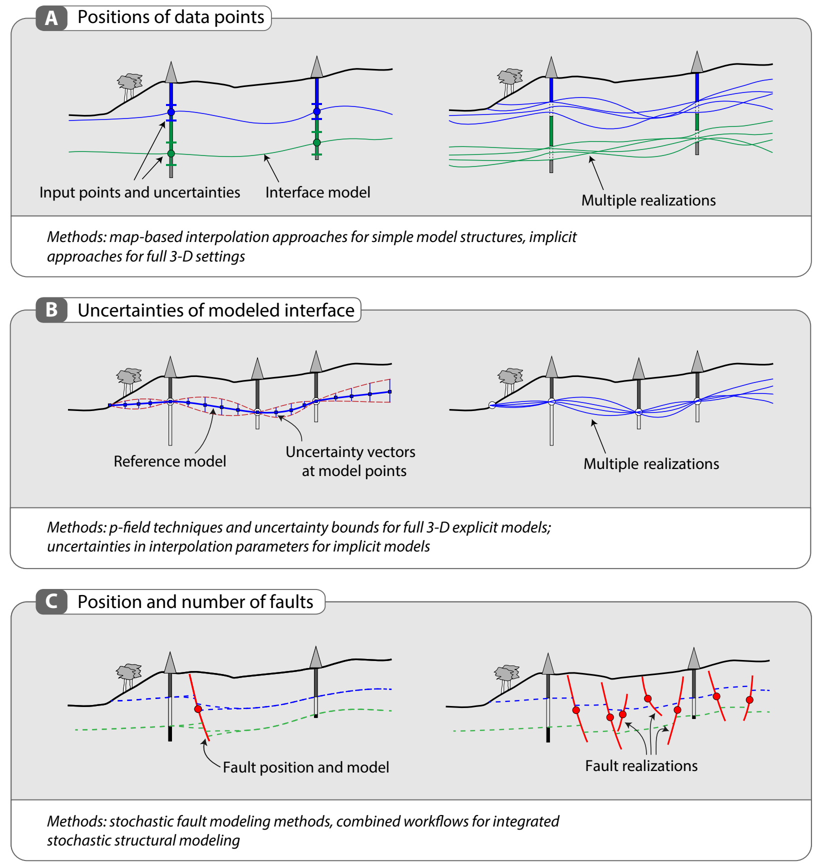
- Data is often noisy
- Hyper-parameters

But:

- Knowledge about the expected structure setting often available



Example of a 3D geological model built in Gempy (de la Varga et. al., 2019)

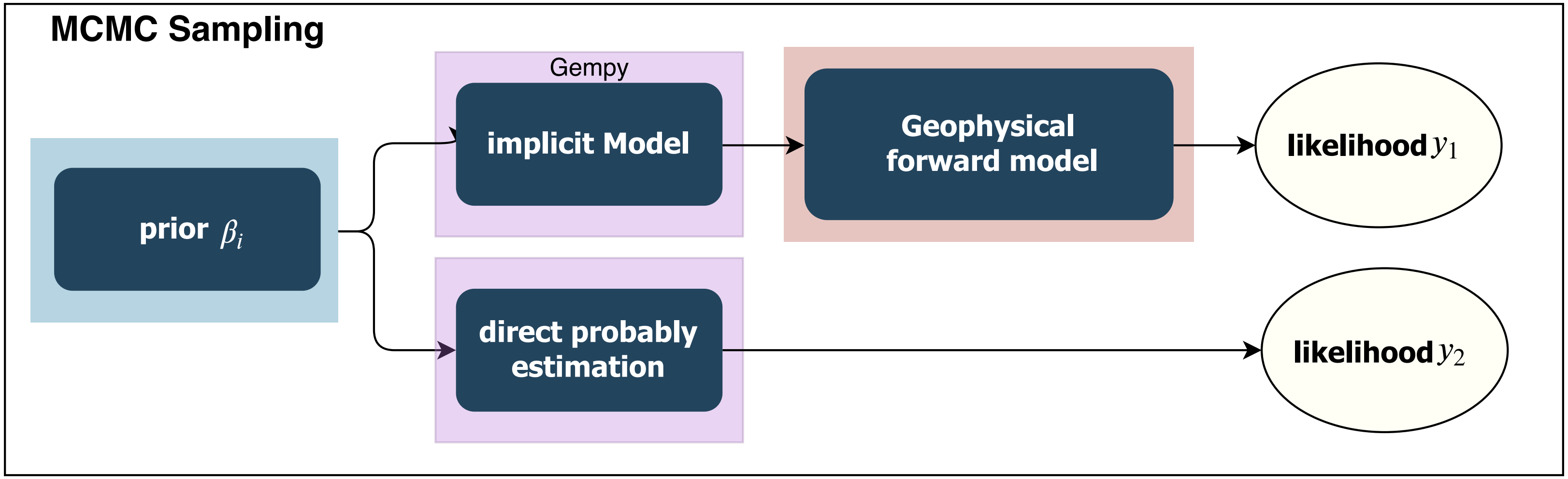


Uncertainties in Geomodeling,(Wellmann and Caumon, 2018)

Motivation

Bayesian inference:

- **Not only** the point estimation of the best-fit parameters, **but** a complete statistical description of the parameter values that is consistent with the data
- Incorporate our knowledge of the geology, and combine observations



Computational Geoscience and Reservoir Engineering | Modern Inverse Problems | RWTH AACHEN UNIVERSITY

Motivation

Bayes' theorem

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

where the denominator, marginal likelihood $p(d)$ is normally not computable, one common solution is to use Markov Chain Monte Carlo (MCMC)

Standard random walk MCMC [1]

- Set $k = 0$ and pick $u^{(0)}$.
- Propose $v^{(k)} = u^{(k)} + \beta\xi^{(k)}, \xi^{(k)} \sim N(0, \mathcal{C})$.
- Set $u^{(k+1)} = v^{(k)}$ with probability $a\left(u^{(k)}, v^{(k)}\right)$.
- Set $u^{(k+1)} = u^{(k)}$ otherwise.
- $k \to k + 1$.

[1] Cotter, Simon L., et al. "MCMC methods for functions: modifying old algorithms to make them faster." Statistical Science (2013): 424-446.

Motivation

Problem: Atandard random walk MCMC method works fine in low dimension problems but is not efficient in high dimesional problem, therefore leads to ineffcient posterior search

Our approach: Applying a recently-developed Hessian informed MCMC

Difficulties: Implementation for efficient calculation of Hessian

Outline

Motivation

Dimension independent MCMC

Automatic Differentiation

Application / Synthetic test cases

Results

Discussion

Summary

Dimension independent MCMC

Based on Rudolf and Sprungk[1] (2018):

Algorithm 1 Generalized Preconditioned Crank-Nicolson

Initialize $m^{(0)} \sim q(m)$

for iteration : $k = 1, 2, \ldots$ **do**

Propose : $x^{\text{cand}} = m_v + \sqrt{(1-\beta^2)}(x^{(k)} - m_v) + \beta\xi^{(k)}, \quad \xi^{(k)} \sim \mathcal{N}(0, C_v)$

Acceptance Probability :

$a_\nu\left(m_{\text{i-1}}, m_{\text{cand}}\right) := \min\left\{1, \exp\left(\Delta\left(m_{\text{i-1}}\right) - \Delta\left(m_{\text{cand}}\right)\right)\right\}$

Where $\Delta(m) = \Phi\left(m, \mathbf{d}_{\text{obs}}\right) + \frac{1}{2}\left\|\mathrm{m} - \mathrm{m}_{\text{prior}}\right\|^2_{C^{-1}_{\text{pior}}} - \frac{1}{2}\left\|\mathrm{m} - \mathrm{m}_\nu\right\|^2_{\mathrm{C}^{-1}_\nu}$

and $\Phi\left(m, \mathbf{d}_{\text{obs}}\right) = \frac{1}{2}\left\|\mathbf{f}(\mathrm{m}) - \mathbf{d}_{\text{obs}}\right\|^2_{\Gamma^{-1}_{noise}}$

$u \sim$ Uniform $(u; 0, 1)$

if $u < a_\nu$ **then**

Accept the proposal : $m^{(i)} \leftarrow m^{\text{cand}}$

else

Reject the proposal : $m^{(i)} \leftarrow m^{(i-1)}$

end if

end for

Question:

- How to determine covariance of posterior C_v ?

[1] Rudolf, Daniel, and Björn Sprungk. "On a generalization of the preconditioned Crank–Nicolson Metropolis algorithm." Foundations of Computational Mathematics 18.2 (2018): 309-343.

Computational Geoscience and Reservoir Engineering
Modern Inverse Problems
RWTH AACHEN UNIVERSITY

Dimension independent MCMC

Construct the covariance of posterior[1]

1. Compute the Maximum a posteriori (MAP), by minimize the negative log posterior

$$m_\nu := \arg\min_m \mathcal{J}(m) := \left(\frac{1}{2} \|\mathbf{G}(m) - \mathbf{d}_{\text{obs}}\|^2_{\Gamma^{-1}_{\text{noise}}} + \frac{1}{2} \|\mathrm{m} - \mathrm{m}_{\text{prior}}\|^2_{C^{-1}_{\text{prior}}} \right)$$

   available methods:
   - **First-order methods:** Stochastic gradient descent, **Adaptive Moment Estimation (Adam)**, etc.
   - **Second-order methods:** Newtown-CG method, etc.
2. Compute the Hessian of negative log of the posterior at MAP

$$\mathcal{H}_{\text{misfit}}\,(m_\nu)$$

3. Construct the Laplace approximation to the posterior:

$$\mathcal{C}_{\text{post}} = \left(\mathcal{H}_{\text{misfit}}\,(m_\nu) + \mathcal{C}^{-1}_{\text{prior}} \right)^{-1}$$

Question:

- How to efficiently calculate Hessian $\mathcal{H}$?

[1] Villa, Umberto, Noemi Petra, and Omar Ghattas. "hIPPYlib: An Extensible Software Framework for Large-Scale Inverse Problems Governed by PDEs; Part I: Deterministic Inversion and Linearized Bayesian Inference." arXiv preprint arXiv:1909.03948 (2019).

Outline

Motivation

Dimension independent MCMC

Automatic Differentiation

Application / Synthetic test cases

Results

Discussion

Summary

Automatic Differentiation

A technique to evaluate the derivative of a function by a computer program.
Implementation software library: **Tensorflow**, Theano etc.

$$\frac{\partial y}{\partial x} = \frac{\partial y}{\partial w_1}\frac{\partial w_1}{\partial x} = \left(\frac{\partial y}{\partial w_2}\frac{\partial w_2}{\partial w_1}\right)\frac{\partial w_1}{\partial x} = \left(\left(\frac{\partial y}{\partial w_3}\frac{\partial w_3}{\partial w_2}\right)\frac{\partial w_2}{\partial w_1}\right)\frac{\partial w_1}{\partial x} = \cdots \tag{1}$$

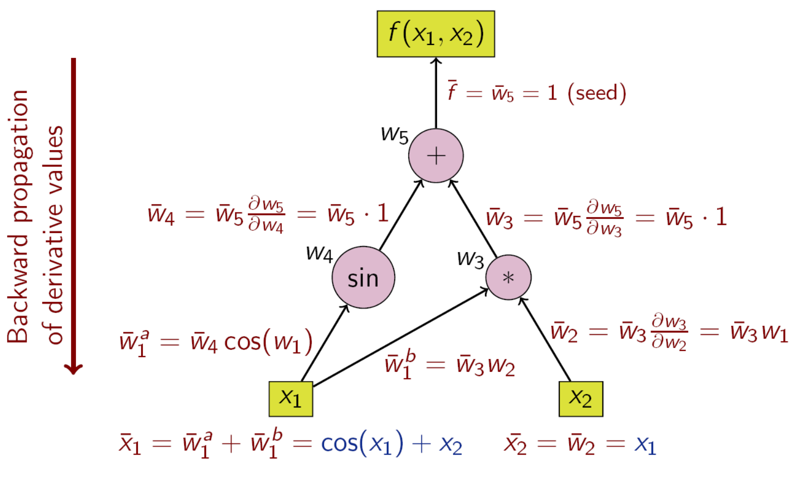
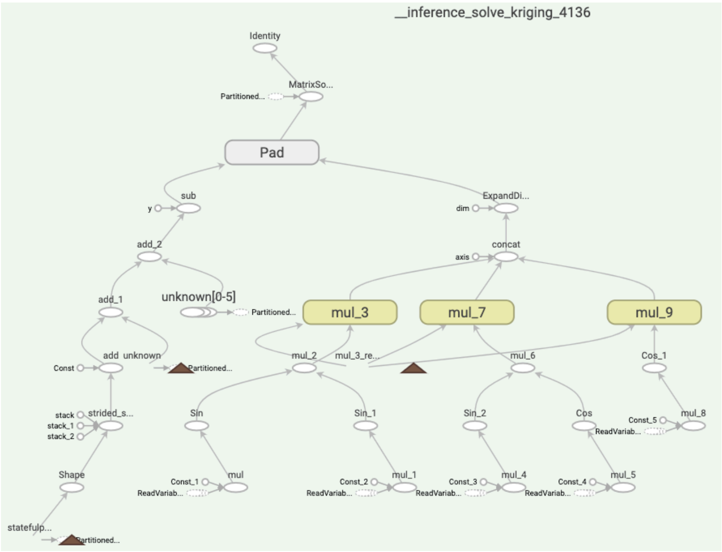


illustration of auto-diff of backward propogation (Wikipedia, Berland, 2007)



part of computational graph in Gempy-Tensorflow

Outline

Motivation

Dimension independent MCMC

Automatic Differentiation

Application / Synthetic test cases

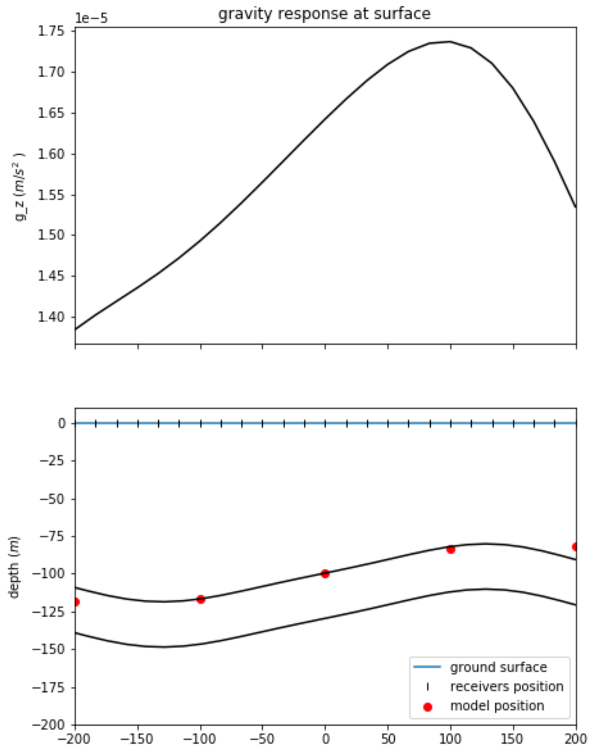
Results

Discussion

Summary

Gravity simulation

2D polygon gravity

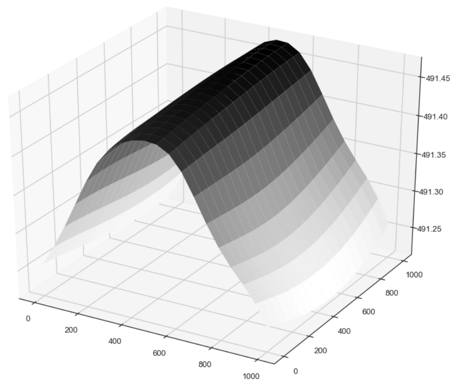


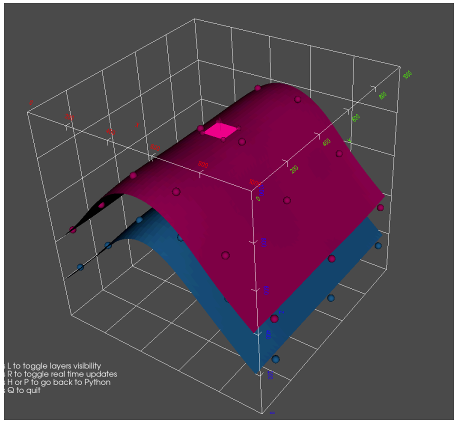
- A simple 2D polygon layer model is built in Tensorflow
- The model is defined by several control points (5 in the left example) and interpolated by Gaussian Process
- A constant layer thickness is added to generate the lower bound, and a constant density is assigned to the layer for simplicity
- The gravity response at the surface is calculated following the line integration algorithm for a polygon[1]

[1] Won, I. J., and Michael Bevis. "Computing the gravitational and magnetic anomalies due to a polygon: Algorithms and Fortran subroutines." Geophysics 52.2 (1987): 232-238.

Application / Synthetic test cases

3D polygon gravity





- 3D geological model is build in Tensorflow as a submodule of Gempy[1]
- The model is defined by several surface points and orientation points (12 surface points and 4 orientation points in the left example)
- An implicit method based on universal co-kriging is used to interpolate scaler fields
- Different density is assigned to each layer and the gravity is calculated at the surface, the details of the algorithm can be fund in the documentation of Gempy

[1] de la Varga, Miguel, Alexander Schaaf, and Florian Wellmann. "GemPy 1.0: open-source stochastic geological modeling and inversion." Geoscientific Model Development (2019).

Outline

Motivation

Dimension independent MCMC

Automatic Differentiation

Application / Synthetic test cases

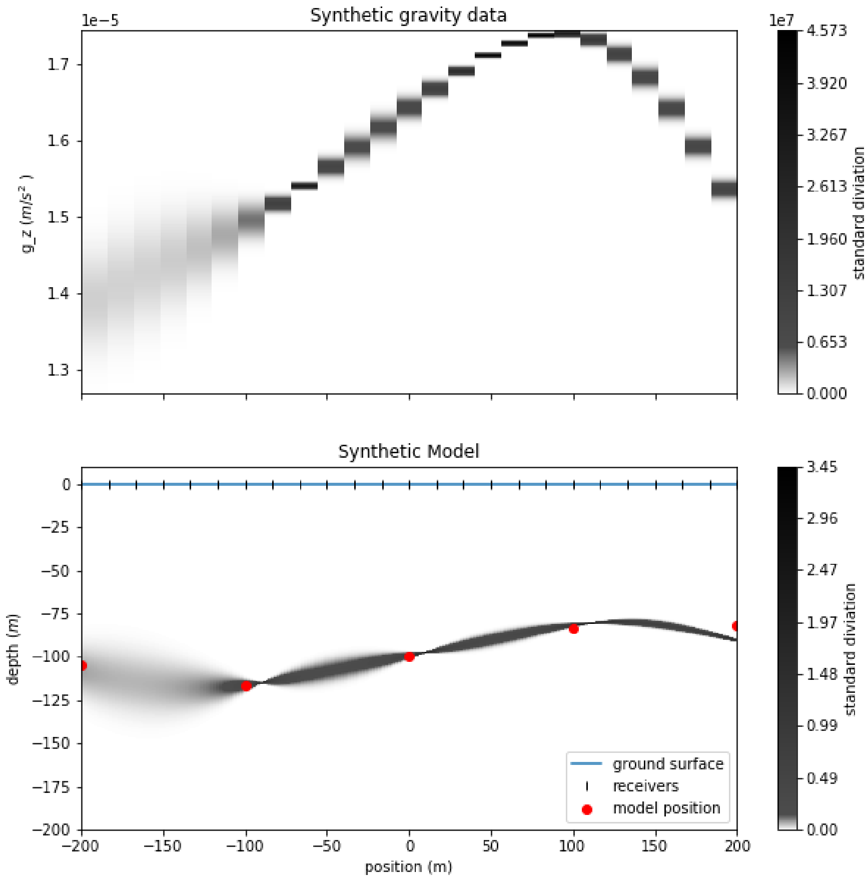
Results

Discussion

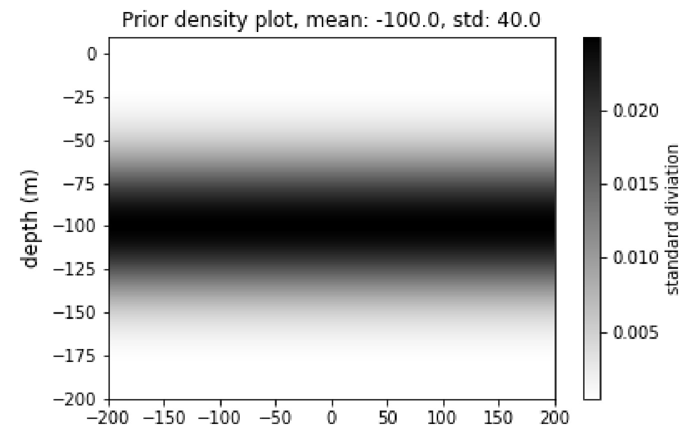
Summary

Results

2D gravity forward simulation



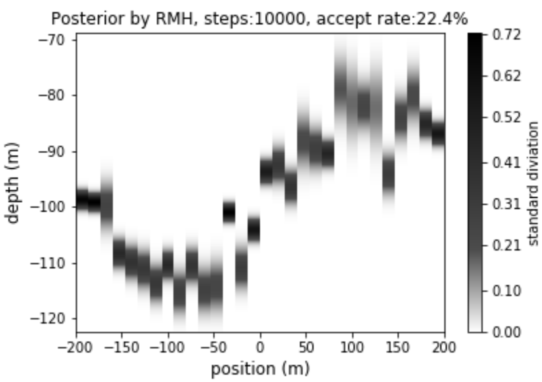
Prior

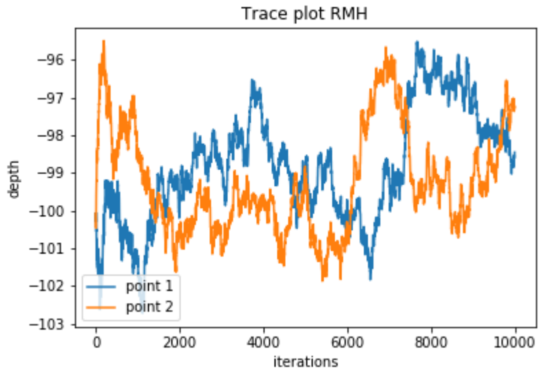


- control points are **mutable in z direction**
- Synthetic gravity data is generated by 1000 forward simulations with high variation on the left end control points (shown on the left)
- Prior probability distribution is defined as normal around $-100\ m$
- An oversampled model with 30 control points is used for the inversion

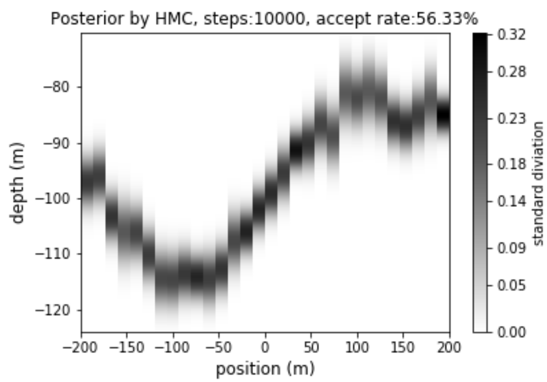
Posterior estimation by different MCMC methods

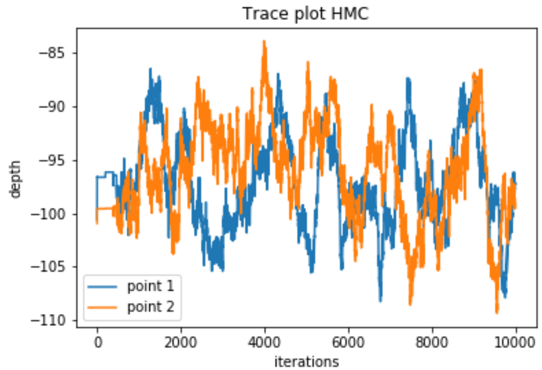
a) Random walk



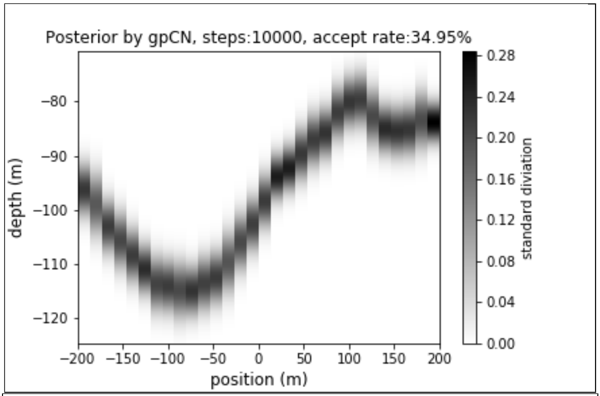


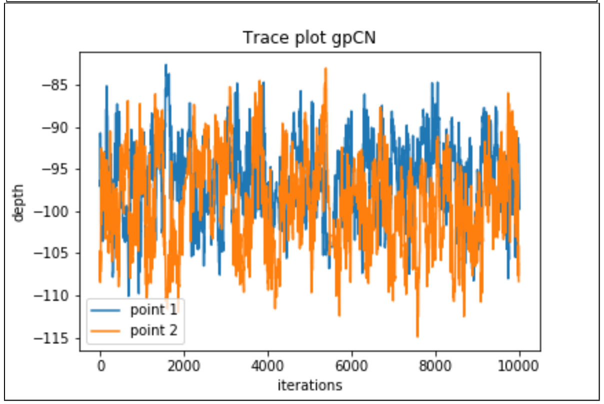
b) HMC





c) our approach gpCN





[2]RHM: Random Walk Metropolis Hastings ; HMC: Hamiltonian Monte Carlo; gpCN: generalized Preconditioned Crank–Nicolson.
[1]Trace plot: the left 2 points

Computational Geoscience and Reservoir Engineering | MIP Modern Inverse Problems | RWTH AACHEN UNIVERSITY

Outline

Motivation

Dimension independent MCMC

Automatic Differentiation

Application / Synthetic test cases

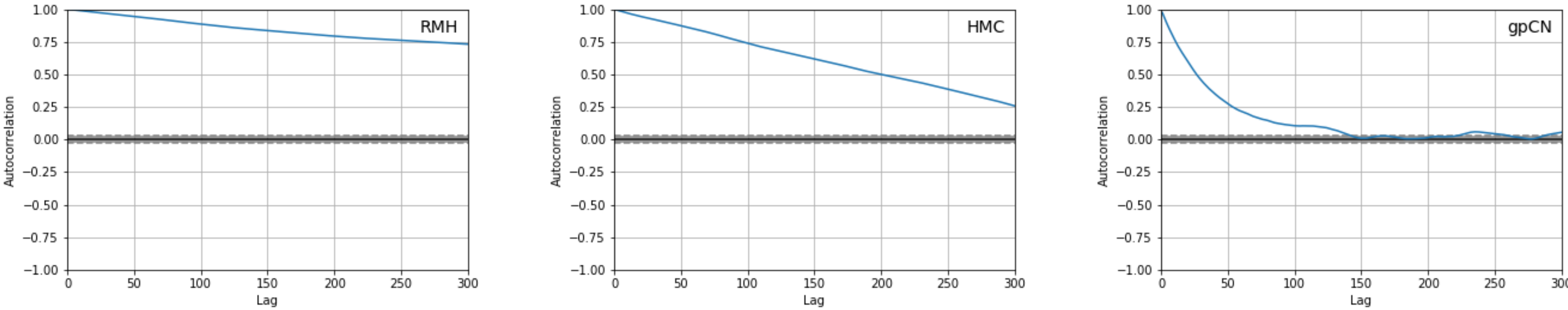
Results

Discussion

Summary

Discussion

Autocorrelation



- no burn-in step is used in this example just to show the initial behaviour of the traces of different methods
- gpCN shows a better mixing very quickly, while RMH and HMC would need to run longer to have a better mixing results
- we can always adjust step size to have higher acceptance rate, but small step size normally will lead to higher autocorrelation between samples which leads to slow convergence

Discussion

Computational cost for 2D gravity example

Methods[1]	Computation time for 10000 samples
RMH	71.60s
HMC	215.10s
gpCN	Finding MAP by Adam: 61.89s+Hessian Calculation[2]: 0.43s +run chain:38.90s = Total time cost: 101.22 s

The computation time highly depend on the forward simulation. Although a properly graphed Tensorflow program has a relatively fast calculation of gradient ($\sim 10ms$ for the 2D example), calculating gradient at each step of the chain boosted the time cost for HMC. However, gpCN we only calculate the Hessian once, therefore it has a better time-efficiency as the number of samples grow.

[1] Runinng on a single core of 2,3 GHz Intel Core i5
[2] Excuting computational graph after first compiling/tracing

Outline

Motivation

Dimension independent MCMC

Automatic Differentiation

Application / Synthetic test cases

Results

Discussion

Summary

Summary

This work demonstrate the efficiency of using novel Hessian informed MCMC in geomodeling to obtain a more efficient estimation of the uncertainties.

Future Work

- implementation on 3D model is still under progress
- application on real cases
- detailed comparison with mroe advanced HMC algorithm (e.g. NUTS)
- investigation of identified model covariance, as an additional interesting aspect
- …

Computational Geoscience and Reservoir Engineering
Modern Inverse Problems
RWTH AACHEN UNIVERSITY

References

- Won, I. J., and Michael Bevis. ”Computing the gravitational and magnetic anomalies due to a polygon: Algorithms and Fortran subroutines.” Geophysics 52, no. 2 (1987): 232-238.
- Lajaunie, Christian, Gabriel Courrioux, and Laurent Manuel. ”Foliation fields and 3D cartography in geology: principles of a method based on potential interpolation.” Mathematical Geology 29, no. 4 (1997): 571-584.
- de la Varga, Miguel, and J. Florian Wellmann. ”Structural geologic modeling as an inference problem: A Bayesian perspective.” Interpretation 4.3 (2016): SM1-SM16.
- de la Varga, Miguel, Alexander Schaaf, and Florian Wellmann. ”GemPy 1.0: open-source stochastic geological modeling and inversion.” Geoscientific Model Development (2019).
- Villa, Umberto, Noemi Petra, and Omar Ghattas. ”hIPPYlib: An Extensible Software Framework for Large-Scale Inverse Problems Governed by PDEs; Part I: Deterministic Inversion and Linearized Bayesian Inference.” arXiv preprint arXiv:1909.03948 (2019).
- Rudolf, Daniel, and Björn Sprungk. ”On a generalization of the preconditioned Crank–Nicolson Metropolis algorithm.” Foundations of Computational Mathematics 18.2 (2018): 309-343.
- Cotter, Simon L., et al. ”MCMC methods for functions: modifying old algorithms to make them faster.” Statistical Science (2013): 424-446.
- Wellmann, Florian, and Guillaume Caumon. ”3-D Structural geological models: Concepts, methods, and uncertainties.” Advances in Geophysics. Vol. 59. Elsevier, 2018. 1-121.
- Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Rafal Jozefowicz, Yangqing Jia, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Mike Schuster, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

Acknowledgment

Financial support from the
Deutsche Forschungsgemeinschaft (DFG)
through grant IRTG-2379 is
gratefully acknowledged