Derivative-informed Bayesian inference for trainable geological modeling -- in modern Machine-Learning framework

From the Faculty of Georesources and Materials Engineering of the
RWTH Aachen University

to obtain the academic degree of

Doktor der Naturwissenschaften
Doctor of Natural Science
approved thesis

submitted by

Zhouji Liang, M.Sc.

Advisors: **Univ.-Prof. Florian Wellmann, Ph.D.**
Univ.-Prof. Omar Ghattas, Ph.D.

Date of the oral examination: 02/06/2023

500000
Fakultät für Georessourcen und Materialtechnik
S. Frenzel-Gumlich
Intzestr. 1
52056 Aachen

Fax: 0241-80-92370
E-Mail: promotionen@fb5.rwth-aachen.de

Eidesstattliche Erklärung

1.)

Hiermit versichere ich eidesstattlich, dass ich die Dissertation selbstständig verfasst und alle in Anspruch genommenen Hilfen in der Dissertation angegeben habe. ☒

____
Unterschrift

____
Ort, Datum

2.) Sofern die Dissertation in einer Einrichtung außerhalb der RWTH entstanden ist:

Hiermit erkläre ich, dass mit Veröffentlichung der Dissertation keine bestehenden Betriebsgeheimnisse verletzt werden. ☐

____
Unterschrift

____
Ort, Datum

RWTH AACHEN UNIVERSITY

Kurzzusammenfassung

Fakultät für Georessourcen und Materialwissenschaften
Computational Geoscience and Reservoir Engineering

Dr.re.nat

Derivativ-informierte Bayes'sche Inferenz für trainierbare geologische Modellierung - in einem modernen Machine-Learning-Rahmen

von Zhouji Liang

Der Untergrund der Erde ist nach wie vor die wichtigste Energie- und Mineralienquelle für die Menschheit. Diese unterirdischen Ressourcen können jedoch nicht ohne ein umfassendes Verständnis dessen, was unter unseren Füßen liegt, gewonnen werden. Geowissenschaftler haben erhebliche Anstrengungen unternommen, den Untergrund zu charakterisieren, um Entscheidungen über die Erschließung von Ressourcen zu treffen, und haben sich dabei zunehmend auf Computermodellierungssoftware verlassen. Ein geologisches 3D-Strukturmodell dient als ein solches Werkzeug, das das Wissen der Geowissenschaftler bündelt und eine effiziente Visualisierung, Kommunikation und fortschrittliche Analyse ermöglicht. Die Sicherstellung einer getreuen Darstellung des Untergrunds ist eine entscheidende Aufgabe für das geologische Modell, sowohl aus finanzieller Sicht als auch aus Sicherheitsgründen. Üblicherweise wird ein einzelnes Modell auf der Grundlage des besten Wissens des Modellierers entwickelt. Jedes Kriterium, auf das sich das Modell über die tatsächliche Beobachtung hinaus stützt, ist jedoch mit gewissen Unsicherheiten behaftet, z. B. die Eingangsdaten, die Interpolation und das fehlende Wissen. Eine gute Quantifizierung dieser Unsicherheiten ist von grundlegender Bedeutung für den Erfolg der Anwendung von geologischen Modellen.

Der Bayes'sche Rahmen bietet einen systematischen Ansatz zur gleichzeitigen Berücksichtigung der Unsicherheiten im Vorwissen und zusätzlicher Beobachtungen innerhalb der Wahrscheinlichkeitsfunktion. Die abgeleitete Wahrscheinlichkeit unter Berücksichtigung zusätzlicher Beobachtungen wird als Posteriorwahrscheinlichkeit bezeichnet. Dieses Inferenzproblem kann oft nicht analytisch gelöst werden. Die Bewertung der Posterior-Verteilung ist gleichbedeutend mit der Erkundung des Posterior-Raums und wird häufig mit Markov Chain Monte Carlo Methoden (MCMC) gelöst.

In den Geowissenschaften werden geophysikalische Daten in großem Umfang zur Charakterisierung des Untergrunds durch die Erfassung physikalischer Signale verwendet. Das angesammelte wissenschaftliche Fachwissen und die gesammelten geophysikalischen Erhebungen machen geophysikalische Daten zu einem attraktiven Kandidaten für die Quantifizierung der Unsicherheit. Die Integration geophysikalischer

Beobachtungen in den Bayes'schen Rahmen ist jedoch eine Herausforderung, da es schwierig ist, verkettete Ableitungen von geologischen Daten zu berechnen, gefolgt von geophysikalischen Simulationen. In dieser Arbeit wird eine End-to-End-Methode vorgestellt, um das Problem der Integration potenzieller Felddaten, insbesondere der Gravimetrie, in den Bayes'schen Inferenzrahmen zu lösen, indem fortschrittliche abgeleitete Inferenzmethoden genutzt werden, die in einer Software für maschinelles Lernen - TensorFlow - implementiert sind.

Zunächst werden Methoden zur Simulation von Schwerfelddaten aus einem geologischen Modell unter Verwendung der impliziten Modellierungsmethode vorgestellt. Die vorgeschlagenen Kernel-Methoden für die Schwerfeldsimulation nutzen den Vorteil des Wertes, der an jeder beliebigen Position im Raum auf der Grundlage der impliziten Modellierungsmethoden abgefragt werden kann, um eine effizientere Schwerberechnung zu erreichen. Darüber hinaus wird eine Verfeinerungsstrategie vorgeschlagen, um eine bessere Genauigkeit im Rahmen der probabilistischen Modellierung zu erreichen.

Anschließend werden Methoden zur Erstellung trainierbarer geologischer Modelle vorgestellt. Die vorgeschlagenen Methoden führen eine glatte Steigungsfunktion ein, um Ableitungsdiskontinuitäten zwischen geologischer Modellierung und Schwerfeldsimulation zu überbrücken. Die vorgeschlagene Methode ermöglicht die Berechnung aussagekräftiger Ableitungen aus einer geologischen Inversion, um die Anwendung fortgeschrittener Inferenzmethoden zu ermöglichen. Eine Visualisierungsmethode, die die Technik der Ordnungsreduktion verwendet, wird zur Visualisierung des trainierbaren Posteriorraums eingesetzt.

Durch die Kombination der vorgestellten Schwerfeldsimulationsmethoden und der trainierbaren geologischen Modellierung ist diese Studie der erste Versuch, fortgeschrittene Inferenzmethoden zu verwenden, darunter die Hessian-informierte MCMC (generalisierte vorkonditionierte Crank-Nicolson, gpCN) und gradienteninformierte Variationsmethode (Stein Variational Gradient Descent, SVGD) für die Anwendung probabilistischer geologischen Modellierung. Es wird ein Ansatz zur effizienten Auswertung des Hessian-Matrix Informationen auf der Grundlage des trainierbaren geologischen Konzepts vorgestellt. Die vorgeschlagene Methodik mit gpCN führt zu eine überlegenen Posterior-Exploration im Vergleich zur State-of-the-Art MCMC-Methode sowohl in synthetischen Beispielen als auch in realen Fallstudien und zeigt das Potenzial für kompliziertere Szenarien. Der SVGD Algorithmus wird vorgeschlagen, um das multimodale Posterior zu bewältigen, was eine Herausforderung für viele MCMC Algorithmen darstellt. Das vorläufige Ergebnis zeigt, dass der Posteriorraum bei der geologischen Inversion mit Schwer als Wahrscheinlichkeit komplex sein könnte und die multimodale Verteilung in der aktuellen Konfiguration schwer aufzulösen ist. Dies zeigt, dass diese Methode weiter verbessert werden muss.

Der SVGD-Algorithmus wird vorgeschlagen, um das multimodale Posterior zu bewältigen, was für viele MCMC-Algorithmen eine Herausforderung darstellt. Vorläufige Ergebnisse zeigen, dass der Posterior-Raum bei der geologischen Inversion mit Schwer

als Wahrscheinlichkeit komplex sein könnte und dass die multimodale Verteilung in der aktuellen Konfiguration schwer zu lösen ist. Dies soll die Möglichkeit der Anwendung von SVGD zur Lösung des modellbasierten Inversionsproblems aufzeigen und auf die Notwendigkeit einer weiteren Verbesserung hinweisen.

RWTH AACHEN UNIVERSITY

Abstract

Faculty of Georesources and Materials Engineering
Computational Geoscience and Reservoir Engineering

Dr.re.nat

Derivative-informed Bayesian inference for trainable geological modeling – in modern Machine-Learning framework

by Zhouji Liang

Earth's subsurface remains the most vital source of energy and minerals for humankind. However, these subsurface resources cannot be extracted without a comprehensive understanding of what lies beneath our feet. Geoscientists have dedicated significant efforts characterizing the subsurface to guide resource development decisions and have increasingly relied on computer modeling software. A 3D structural geological model serves as such a tool that encapsulates the knowledge of geoscientists and provides an efficient visualization, communication, and advanced analysis. Ensuring a faithful representation of the subsurface is a crucial task for the geological model, both from the financial point of view and for safety reasons. Conventionally, a single model is developed based on the modeler's best knowledge. Yet, any criteria the model was based on beyond our actual observation is subject to certain uncertainties, for example, the input data, the interpolation, and the missing knowledge. A good quantification of these uncertainties is fundamental for the success of the application of geological models.

The Bayesian framework offers a systematic approach to simultaneously consider the uncertainties in the prior knowledge and additional observations within the likelihood function. The inferred probability with the consideration of additional observations is referred to as the posterior probability. This inference problem often cannot be solved analytically. Evaluating the posterior distribution is equivalent to the exploration of the posterior space and is often solved using Markov Chain Monte Carlo (MCMC) methods.

In the geosciences, geophysical data is widely used to characterize the subsurface through collection of observations of physical signals. The accumulated scientific expertise and collected geophysical surveys make geophysical data an attractive candidate for uncertainty quantification. However, integrating geophysical observations in the Bayesian Framework is challenging due to difficulties in calculating chained derivatives of geological followed by geophysical simulations. This thesis presents an end-to-end methodology to solve the problem of integrating potential field data, specifically gravity, into the Bayesian inference framework by leveraging advanced

derivative-informed inference methods, implemented in a Machine Learning framework - TensorFlow.

First, methods are introduced to simulate gravity data from a geological model using the implicit modeling method. The proposed kernel methods for gravity simulation take advantage of the value that can be queried at any position in space based on the implicit modeling methods to achieve a more efficient gravity calculation. In addition, a refinement strategy is proposed to achieve better accuracy in the probabilistic modeling framework.

Then, methods to create trainable geological models are introduced. The proposed methods introduce a smooth slope function to bridge derivative discontinuities between geological modeling and gravity simulation. The proposed method enables meaningful derivatives to be calculated from a geological inversion to allow the application of advanced inference methods. A visualization method using the order-reduction technique is adopted to visualize the trainable posterior space.

Finally, by combining the introduced gravity simulation methods and trainable geological modeling technique, this study is the first attempt to adopt advanced inference methods, including the Hessian-informed MCMC (generalized preconditioned Crank-Nicolson, gpCN) and gradient-informed variational method (Stein Variational Gradient Descent, SVGD) to the application of probabilistic geological modeling. The approach to efficiently evaluate Hessian information based on the trainable geological concept is introduced. The proposed methodology using gpCN has demonstrated the superior performance of posterior exploration compared to the state-of-the-art MCMC method in both synthetic examples and real case studies and shows the potential for more complex scenarios.

The SVGD algorithm is proposed in an attempt to tackle the multimodal posterior, which is a challenge for many MCMC-type algorithms. Preliminary results show that the posterior space in geological inversion with gravity as the likelihood could be complex, and the multimodal distribution is difficult to resolve in the current configuration. This is intended to show the possibility of applying SVGD to solve the model-based inversion problem and indicates the need for further improvement.

Acknowledgements

First, I would like to express my deepest gratitude to Prof. Florian Wellmann for his constant support and encouragement during my doctoral study. I would like to thank both my thesis advisors, Prof. Florian Wellmann, and Prof. Omar Ghattas, for their inspiring discussions, stimulating ideas and generous guidance to my study. Their scientific insights into the topic of computational geosciences served as a strong motivator throughout the thesis's preparation.

I would like to thank my ex-colleague at CGRE group at RWTH, Miguel de la Varga, for his preliminary and pioneering work in the geological modeling framework. Miguel is always generous and patient in sharing his experience and insights on this topic with me and spends time meeting me for scientific discussion during his busy start-up period. His work provides a substantial foundation for this thesis and is deeply appreciated. I also thank Dr. Denise Degen for the insightful discussions in process simulation.

I would also like to thank Professor Florian Wagner for providing generous and constructive advice for this work with his expertise in geophysics.

I would like to thank my senior colleague and officemate in AICES, and also friends Dr. Hu Zhao, Dr. Ankit Chakraborty, and Eshwar Savitha. Hanging out with them for lunch always relaxes me from the stressed work. I thank all the fellows at CGRE. Working with these fellows with intelligence and enthusiasm in such a friendly group environment is a great pleasure.

I would like to give special thanks to my parents for their support of my studies over the years. I would like to thank my girlfriend, Luying Hua, for her support. Only with her, study life in Germany becomes this colorful! I would also like to thank my friends Min Liu, Zhaohui Ning and Ruikun Cao for their support. Ruikun has always supported me when I needed advice, and his profound knowledge and expertise in computer science have greatly helped me in my studies.

Last but not least, I would like to thank Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – 333849990/GRK2379 (IRTG Modern Inverse Problems) for funding this work. I am also grateful to Prof. Marek Behr and Dr. Nicole Faber for the organization of IRTG at RWTH. The organization provides an excellent research environment with numerous scientific seminars and lectures. The support from the IRTG service team is also grateful.

Contents

List of Abbreviations

AD	**A**utomatic **D**ifferentiation
FD	**F**inite **D**ifference
FE	**F**inite **E**lement
gpCN	**g**eneralized **p**reconditioned **C**rank-**N**Nicolson
GD	**G**radient **D**escent
HMC	**H**amiltonian **M**Monte **C**arlo
MAP	**M**aximum **A** **P**osteriori
MCMC	**M**arkov **C**hain **M**onte **C**arlo
PDE	**P**artial **D**ifferential **E**quation
RMH	**R**andom-walk **M**etropolis-**H**astings
RMSE	**R**oot **M**ean **S**quare **E**rror
RWTH	**R**heinisch **W**estfälische **T**echnische **H**ochschule
SVGD	**S**tein **V**ariational **G**radient **D**escent

Chapter 1

Introduction

1.1 Background and Motivation

The Earth's lithosphere consists of highly heterogeneous materials, which bring human society both rich natural resources (e.g., mineral deposits, hydrocarbon reservoirs) and natural hazards (e.g., earthquakes, volcanic eruptions) in the meantime. In contrast to the traditional ways of utilizing the subsurface, such as hydrocarbon extraction and tunnels development, there are now many more ways, such as carbon capture and storage (Boot-Handford et al., 2014), geothermal energy extraction (Rohit et al., 2023), etc. Therefore, evaluating the spatial distribution of the properties is crucial for many geological applications (e.g., Ross et al., 2005; Culshaw, 2005; Guglielmetti et al., 2013; Rongier et al., 2014; Høyer et al., 2019).

Over the past decades, Geoscientists have put substantial efforts into understanding the subsurface for both practical and scientific purposes. Reliable visualization tools to faithfully represent and communicate geological knowledge are essential (Wang et al., 2015). The 3D structural geological model (also referred to as the geological model), which represents the geometric elements, is able to capture large-scale heterogeneity. Hence, as an extension of the conventional geological mapping (Jones et al., 2004; Carranza, 2011), 3D geological models are often used as an tool to visualize and encapsulate the knowledge of the surface (Wu et al., 2005; Wellmann and Caumon, 2018; Caumon et al., 2009; Hillier et al., 2014). Various modeling schemes have been developed for this purpose (Jessell et al., 2014).

Geological models are often based on the interpolation of observational data. They are inevitably subject to uncertainties due to the input data uncertainties, inherent randomness, and inadequate understanding of structural existence (Wellmann et al., 2010). According to the previous studies on the uncertainties of the geological model, it has a significant impact on the modeling results (Tacher et al., 2006; Wellmann et al., 2010; Pirot et al., 2015; Linde et al., 2017; Wellmann and Caumon, 2018). Hence, quantifying the uncertainties is crucial for decision making, and risk assessment (Dominy et al., 2002; Singer and Menzie, 2010; Jessell et al., 2014). Methods have been developed in recent years to simulate and quantify the uncertainties of a geological model (Tacher et al., 2006; Wellmann et al., 2010; Lindsay et al., 2012;

Jessell et al., 2014). Notably, Wellmann et al. (2010) proposed an approach for uncertainty quantification using geological inversion, which generates realizations based on simulated data sets, which provides the foundation for geological data-driven ensemble modeling and inversion in the following development of Bayesian framework in geological inversion by de la Varga and Wellmann (2016). The Bayesian framework is also the methodology we follow in this study.

The Bayesian framework provides a systematic way to incorporate geological modeling and additional data. Multiple sources of additional information have been introduced in recent years as an external constraint to the model uncertainties. de la Varga and Wellmann (2016) used layer thickness in the pioneering work of Bayesian geological modeling. (Wellmann et al., 2018) used gravity data as an additional constraint to the model uncertainties. Stamm et al. (2019) used multiple likelihoods, including seal thickness, reservoir thickness, and Shale smear factor, as the likelihood function for the study of reservoir assessment. Schaaf et al. (2021) constrains the model with topology information as the likelihood. Each of these is useful for specific additional information. Among these, potential-field data (e.g.,gravity, magnetic data) is especially attractive to geophysics due to the abundant available data collected over years (Jessell et al., 2014; Nabighian et al., 2005a,b; Giraud et al., 2019b; Güdük et al., 2021).

For a faithful and efficient uncertainty quantification with potential-field data in the Bayesian framework, efforts must be made to address two aspects of the numerical model, the forward modeling and inversion methods. On the one hand, the link between geological modeling and forward gravity simulation has long been attracting researchers (Jessell et al., 1993; Jessell and Valenta, 1996; Jessell, 2001; Jessell et al., 2010; Lindsay et al., 2012). On the other hand, a comprehensive statistical analysis of the uncertainties often requires characterizing the posterior probability to provide in the Bayesian problem.

Until recently, many uncertainty quantification methods under the framework of geological modeling were limited to algorithms such as Monte Carlo type algorithms (Lindsay et al., 2013; Yamamoto et al., 2014; Thiele et al., 2016; Schweizer et al., 2017; Pakyuz-Charrier et al., 2018; Giraud et al., 2019a; Schaaf et al., 2021), or derivative-free Markov Chain Monte Carlo type algorithms (de la Varga and Wellmann, 2016; Wellmann et al., 2018; Stamm et al., 2019; Scalzo et al., 2021). While these methods are appropriate for problems with lower dimensionality, as the dimensionality increases due to more complex models and input data, one has to face the curse of dimensionality, and the conventional methods are inefficient. To address this problem, de la Varga et al. (2019) introduced the Automatic Differentiation (AD) framework to the modeling software GemPy. AD aims to provide end-to-end differentiable geological modeling to allow derivative information to be efficiently evaluated through the geological model. This work has provided a perfect pavement, but improvement in this framework is still required to achieve a successful and efficient uncertainty quantification using potential-field data in the geological inversion. Therefore, the aim of this thesis is to improve the previous work and further develop under the current AD and Bayesian framework to achieve a more feasible UQ for 3D geological modeling.

1.2 Research Objectives

As the entry to industry 4.0 (Wellmann, 2022), the number of data we are collecting is growing, and in the meantime, the demand for efficiency in the workflow is also increasing. Hence, there is a rising need for a more efficient and intelligent uncertainty quantification workflow. In fact, advanced techniques and algorithms have been rapidly developed in the community of applied mathematics and computer science. However, applying these advanced algorithms to geological modeling problems is not straightforward and requires some special tuning of the forward model.

To address the research gap, this thesis aims to extend the current probabilistic geological inverse framework work and develop novel schemes for combined geological modeling and geophysical inversion. A graph representation of this thesis is given in Figure 1.1. The topic of the thesis consists of two essential components under the derivative-informed Bayesian inference framework, namely the forward part and inverse part. In the forward part, we extend the previous differentiable modeling framework of GemPy (de la Varga et al., 2019) and aim to provide efficient and accurate forward gravity simulation (Chapter 3) in the probabilistic modeling framework. The inverse part aims to use advanced statistical tools to efficiently solve the Bayesian inference problem (Chapter 5,6). The derivative information provided by the trainable forward modeling method (Chapter 4) links the two parts and is essential for the success of the inverse methods. More specifically, the research objects are listed as follows:

- Develop a new methodology to enable an efficient and accurate gravity simulation, by taking advantage of the fact that properties can be queried at any locations using the meshless implicit modeling method and innovate new meshing refinement strategy for probability modeling framework.
- Develop an efficient methodology which can provide appropriate derivatives information for the inversion. Investigate the posterior surface using novel approaches.
- Develop an efficient methodology which adopting advanced derivative-informed inference methods to solve the geological inversion problems with gravity data as an additional constraint.

1.3 Outline

Chapter 2 presents a review of the implicit modeling method and the associated uncertainties. A literature review is presented first, followed by a discussion about the

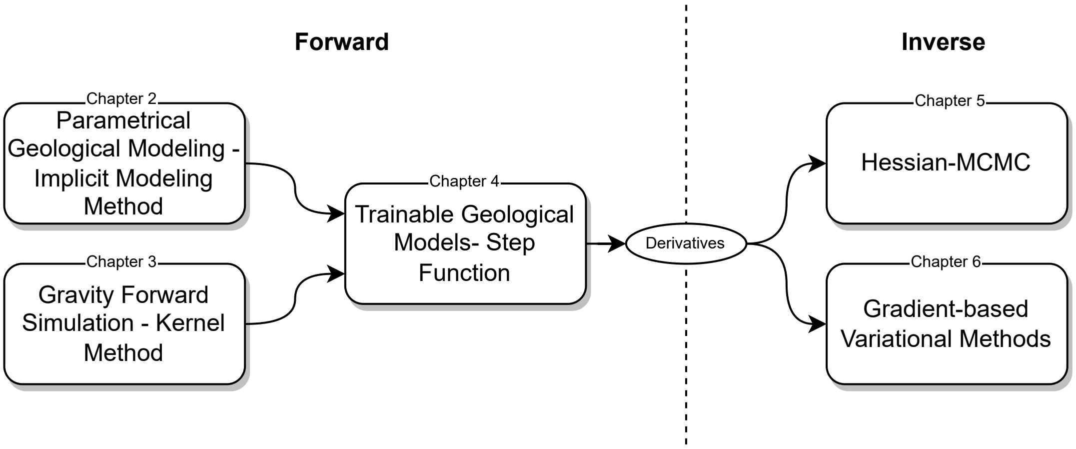


FIGURE 1.1: Structure of research questions addressed in this thesis.

uncertainties in the geological models. Then I will review the mathematical details of the co-kriging modeling method used in the study. After that, a review of the Bayesian framework to reduce and quantify the uncertainties in the geological modeling is given. The method to use of additional geophysical data in the Bayesian framework as an additional constraint to the modeling construction is introduced.

Chapter 3 presents a new methodology refered to as the kernel method for gravity simulation in implicit modeling. The concept of the method is first demonstrated, and the proposed subsequent refinement strategy is then presented. A numerical example with a 3D model of the Greenstone-belt in Western Australia is presented for the validation of the proposed methodology. Finally, computational efficiency is discussed at the end. This chapter is slightly modified from the manuscript under review: "*Kernel method for gravity forward simulation in implicit probabilistic geological modeling.*" submitted to Geophysics, (under review).

Chapter 4 introduces a new methodology to provide a trainable model in the framework of gravity constrained probability geological modeling. The method is first explained using synthetic models and the application of the proposed step-function to deal with different types of discontinuities in the implicit modeling method. Finally, a reduced-dimensional method is introduced to provide an intuitive and concise visualization of the trainable posterior.

Chapter 5 proposes a new methodology for efficient posterior exploration using the Hessian-informed MCMC method in a probabilistic geological modeling framework. First, the MCMC method - Generalized Preconditioned Crank-Nicolson (gpCN) is presented. Then the method to efficiently calculate the Hessian using Automatic-Differentiation through the geological model is introduced. Finally, numerical examples are presented and compared with other methods. This chapter is slightly modified from the published paper: Liang, Zhouji, Florian Wellmann, and Omar Ghattas. "*Uncertainty quantification of geological model parameters in 3D gravity inversion by Hessian-informed Markov Chain Monte Carlo.*" Geophysics 88.1 (2022): 1-78.

Chapter 6 discussed the multimodal posterior in the probabilistic geological modeling problem with gravity data as the likelihood function. The multimodality problem is first presented, and the Stein-Variational gradient descent (SVGD) method is adopted to solve the multimodal problem based on the trainable geological model introduced in Chapter 4. Finally, the proposed method is demonstrated by using a synthetic example.

Chapter 7 and 8 summarizes the thesis and gives recommendations to future research.

Chapter 2

Geological Modeling and Uncertainty Quantification

2.1 Introduction

3D structural geological models have become increasingly important as tools to provide a comprehensive representation of subsurface geological structures. In various fields of applications, e.g., geothermal energy exploration, groundwater, mining, and hydrocarbon exploration, a faithful geological model is crucial for decision-making. However, a geological model is naturally subject to inevitable uncertainties arising from different sources (Wellmann and Caumon, 2018). These include bias and imprecision in the observation data, incomplete knowledge about the structures, and inefficiency and imperfectness of the modeling methods to capture the randomness and stochasticity in interpolation and extrapolations (Wellmann et al., 2010). Therefore, a better understanding and quantification of the uncertainties associated with the geological model is crucial to the applications and attracted many researchers' attention in recent years (Lindsay et al., 2012; Rawlinson et al., 2014; Bond, 2015; de la Varga and Wellmann, 2016; Jessell et al., 2018; Wellmann and Caumon, 2018). As the center of quantifying the uncertainty associated with geological modeling, the accuracy and also efficiency of the modeling methods is the foundation of the problem. An excellent review of the recent development of geological modeling and uncertainty quantification in geological modeling is given by (Wellmann and Caumon, 2018). In this Chapter, I will discuss the geological modeling methods and the link to uncertainties. I will start by briefly reviewing the recent development of geological modeling methods. Then I will review the implicit modeling method used in this study, the universal co-kriging method, in theoretical detail. Finally, I will review the recent development of the Bayesian inference framework for stochastic geological modeling.

2.2 Review of Recent Development of Geological Modeling

Various approaches to model the subsurface structures have been developed in recent decades (see review: Caumon et al., 2009; Jessell et al., 2014; Wellmann and Caumon, 2018). Despite the differences in the use of input data and geological knowledge in different models, the common objective of geological modeling is seeking a faithful numerical representation of the subsurface boundaries, which distinguishes rock units with distinctive properties. These boundaries are generated due to various reasons, including changing sedimentary environment, erosion process, tectonic movement, etc. (Wellmann and Caumon, 2018). Therefore, rigorous surface modeling is fundamental to represent the modelers' understanding of the underlying geological process.

These surface modeling approaches can be generally classified into two categories: explicit and implicit methods. Explicit modeling methods directly construct the surface geometry by either triangulation or interpolation of points on the surface (Mallet, 1992; Wu et al., 2005; Caumon et al., 2009). This naturally requires detailed surface data and is often used in the petroleum industry, where, more often, abundant seismic data are accessible. However, topological relationships of the surfaces often require to be handled with special care to satisfy the presumed structural constraints (Caumon et al., 2004, 2009; Jessell et al., 2014). Here, topology is a term that refers to the relationships of connection between the different domains of a geological model. Therefore, the model developing process using explicit methods can be time-consuming and requires tedious work for modelers to carefully pick and tune the surfaces to force a correct topological relationship (Jessell et al., 2014; Wellmann et al., 2018). Recent development in explicit modeling methods attempts to find a more automatic way of the modeling process while maintaining the topological relationships using a wide range of methods, for example: automatically removing horizon data adjacent to the fault (see review: Wellmann and Caumon, 2018), Voronoi polyhedra (Hale and Emanuel, 2003; Merland et al., 2014), and more recent NURBS (non-uniform rational B-splines) and subdivision surfaces (Moulaeifard et al., 2022)

In contrast, implicit modeling methods interpolate a scalar function based on the interface locations and attitudes data together with additional spatial constraints. The structural surfaces are represented by the iso-surfaces based on the interpolated scalar field. Implicit modeling enables a fully automated model construction and represents complex structures while considering the topological constraints implicitly. Therefore, implicit methods have drawn increasing attention from the modelers, and diverse variants have been developed. These variants of the implicit modeling methods can be divided into two branches: discrete and meshless. In the discrete branch, a predefined mesh is required for computation. A notable method in the discrete branch of implicit methods is using volumetric discrete smooth interpolation (DSI) on tetrahedral meshes (Frank et al., 2007; Massiot and Caumon, 2010; Caumon et al., 2012). Another branch is the meshless method which is widely adopted in various forms in several modeling softwares due to its convenience and flexibility. Among these meshless methods, different methods can be distinguished from each other in the way of data interpolation and the underlying criteria the interpolation follows. The software Noddy (Jessell, 1981) and its Python distribution PyNoddy (Wellmann et al., 2016) use kinematic models to represent the tectonic events by the transformation of the scalar field. The commercial software GeoModeller and the recently developed open-source package

GemPy utilize the co-kriging method (Lajaunie et al., 1997) to interpolate the observational data. Turk and O'brien (2005); Carr et al. (2001); Hillier et al. (2014) utilized radial basis functions for the scalar field interpolation. The more recently released package LoopStructural attempts to combine both the co-kriging and kinematic methods (Grose et al., 2021). Other studies used Machine Learning methods to determine the boundaries between scattering data belonging to different lithology groups (Gonçalves et al., 2017; Hillier et al., 2021). After obtaining the scalar function, often a level-set or similar technique is used to define the surface boundaries. Further discretization is then often required to extract the iso-surfaces for visualization (e.g., the Marching cube algorithm) and geophysical simulations purposes. In this study, we will focus on the co-kriging method due to its abilities of fast model construction and the close link to stochastic geological modeling.

2.3 Universal Co-kriging Method

The kriging method has long been used in Geostatistics in many different applications (Chiles and Delfiner, 2009). Kriging provides an unbiased estimation of the unknown data based on the correlation between input data. Different kriging methods mainly differ in the way of taking the values at unknown positions: simple kriging (zero value), ordinary kriging (the mean value), and universal kriging (drifting functions). Co-kriging allows the correlation between different types of data to be considered simultaneously. The kriging method has been introduced to geological modeling since the work by (Lajaunie et al., 1997) and drawn more attentions in recent development (e.g., Calcagno et al., 2008; de la Varga et al., 2019).

In this study, we follow the implicit modeling method using universal co-kriging implemented in GemPy (Lajaunie et al., 1997; de la Varga et al., 2019). Similar to many other implicit methods discussed above, the co-kriging methods seek an artificial auxiliary scalar function $\mathbf{Z}$ to represent interfaces of the subsurface. The scalar value at a given position in three-dimensional space $\boldsymbol{x} = (x, y, z) \in \mathbb{R}^3$ is given by $\mathbf{Z}(\boldsymbol{x})$. The block of strata with similar lithology are grouped by level-set methods based on the scalar field. This method takes two types of input data into the interpolation:

1. The surface point data, which contains the data location $\boldsymbol{x}$ and the corresponding reference scalar value $\mathbf{Z}_{\boldsymbol{x}}$ of the interface of interest.

2. The orientation points, which contains of the data location $\boldsymbol{x}$ and the corresponding attitudes data $\mathbf{G}$ (where $\mathbf{G} = (G^z, G^y, G^z)$ represents the gradient in 3D case).

One should note that the input data for surface points and orientation points do not have to be at the same locations. We define surface points belonging to any specific interface k as $\boldsymbol{x}^k$. According to the constraining rules in the co-kriging approach (Lajaunie et al., 1997), the scalar function at $\mathbf{Z}(\boldsymbol{x}^k)$ (simplified as $\mathbf{Z}^k$) should share the same scalar value $Z_{\boldsymbol{x}^k}$. The gradient of the scalar filed at $\boldsymbol{x}$ is defined as $\frac{\partial \mathbf{Z}}{\partial \mathbf{u}}(\boldsymbol{x})$ with $\mathbf{u}$ being any unit vector (simplified as $\nabla \mathbf{Z}$), and takes the following analytical form:

$$\frac{\partial \mathbf{Z}}{\partial \mathbf{u}}(\boldsymbol{x}) = \lim_{\varrho \to 0} \frac{\mathbf{Z}(\boldsymbol{x} + \varrho \mathbf{u}) - \mathbf{Z}(\boldsymbol{x})}{\varrho} \tag{2.1}$$

The gradient at $\nabla \mathbf{Z}(\boldsymbol{x})$ should be consistent with the attitudes data $\mathbf{G}(\boldsymbol{x})$ at the given locations and should be pointing the stratigraphy younging direction of the lithologies (perpendicular to the tangent of scalar field). The constants for the co-kriging system can be summarized by the following equations (Lajaunie et al., 1997):

$$\begin{cases} \nabla \mathbf{Z}(\boldsymbol{x}_i) = \mathbf{G}(\boldsymbol{x}_i) & \forall i \in I \\ \langle \nabla \mathbf{Z}(\boldsymbol{x}_{i'}), \boldsymbol{\tau}(\boldsymbol{x}_{i'}) \rangle = 0 & \forall i' \in I' \\ \mathbf{Z}(\boldsymbol{x}_j) - \mathbf{Z}(\boldsymbol{x}_{j'}) = 0 & \forall j, j' \in J \end{cases} \tag{2.2}$$

where I and J are data sets at a specific iso-surface for orientation points and surface points, respectively. We denote arbitrary independent pair of point in $jj' \in \mathcal{P}\left(J_k\right)$, with $\mathcal{P}$ denotes one set of independent pairs at surface k', and surfaces with orientation data as $i \in I$ and $i' \in I'$, and $\boldsymbol{\tau}$ is the tangent of the scalar field. The absolute values of the scalar function at $\mathbf{Z}(\boldsymbol{x})$ have no significance to our application, while only the relative values are needed to represent subsequential relationships between the surfaces. Therefore, in practice, an arbitrary origin $\boldsymbol{x}_0$ from the input data is often selected as a reference point for the calculation.

As a variant to the original co-kriging method, Lajaunie et al. (1997) introduced a universal drift function as an external constraint to the scalar field to maintain a subparallel pattern. The external drifting function is often defined using polynomial functions f_d (Lajaunie et al., 1997; de la Varga et al., 2019) and should also follow the zero increment criteria, meaning:

$$f_d(\boldsymbol{x}_i) - f_d(\boldsymbol{x}_{i'}) = 0 \tag{2.3}$$

We will follow the notation for covariance and cross-covariance functions in Lajaunie et al. (1997) and de la Varga et al. (2019). We denote the covariance function of $\mathbf{Z}$ as $\boldsymbol{K}_Z$, and denote the vector joining an arbitrary points $\boldsymbol{x}_j$ and $\boldsymbol{x}_{j\prime}$ as $\boldsymbol{r} = \boldsymbol{x}' - \boldsymbol{x}$. Then we can write the definition of the covariance function:

$$\boldsymbol{K}_Z(\boldsymbol{r}) = C_Z(r) \tag{2.4}$$

Where $r = |\boldsymbol{r}|$ is the modules, C_Z is a chosen covariance kernel. We chose to use a cubic covariance kernel in this study due to its twice-differentiability, robustness, and its coherent geological description of the space (de la Varga et al., 2019).

By taking all the above-described constraints into consideration, the kriging system takes the following form:

$$\begin{bmatrix} \mathbf{C} & \mathbf{U}^t \\ \mathbf{U} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{k_a} \\ \mathbf{k_b} \\ \mathbf{k_c} \\ \mathbf{k_d} \end{bmatrix} = \begin{bmatrix} G_i^x \\ G_i^y \\ G_i^z \\ \mathbf{0} \end{bmatrix} \tag{2.5}$$

where $\mathbf{C}$ is the covariance matrix, $\mathbf{U}$ is the universal drift matrix, $\mathbf{k_a}$,$\mathbf{k_b}$,$\mathbf{k_c}$,$\mathbf{k_d}$ contain the kriging parameters we solve. The covariance matrix $\mathbf{C}$ takes the following form:

$$\mathbf{C} = \begin{bmatrix} \boldsymbol{K}_{G_x}^{i'i} & \boldsymbol{K}_{G_xG_y}^{i'i} & \boldsymbol{K}_{G_xG_z}^{i'i} & \boldsymbol{K}_{G_xZ}^{ji} - \boldsymbol{K}_{G_xZ}^{j'i} \\ \boldsymbol{K}_{G_yG_x}^{i'i} & \boldsymbol{K}_{G_y}^{i'i} & \boldsymbol{K}_{G_yG_z}^{i'i} & \boldsymbol{K}_{G_yZ}^{ji} - \boldsymbol{K}_{G_yZ}^{j'i} \\ \boldsymbol{K}_{G_zG_x}^{i'i} & \boldsymbol{K}_{G_zG_y}^{i'i} & \boldsymbol{K}_{G_z}^{i'i} & \boldsymbol{K}_{G_zZ}^{ji} - \boldsymbol{K}_{G_zZ}^{j'i} \\ \boldsymbol{K}_{ZG_x}^{i'j''} - \boldsymbol{K}_{ZG_x}^{i'j'''} & \boldsymbol{K}_{ZG_y}^{i'j''} - \boldsymbol{K}_{ZG_y}^{i'j'''} & \boldsymbol{K}_{ZG_z}^{i'j''} - \boldsymbol{K}_{ZG_z}^{i'j'''} & \boldsymbol{K}_{Z}^{j'j''} - \boldsymbol{K}_{Z}^{j'j'''} - \boldsymbol{K}_{Z}^{j'j''} + \boldsymbol{K}_{Z}^{j'j'''} \end{bmatrix} \tag{2.6}$$

where each of the entry in the covariance matrix is defined as follows. The covariance function of the scalar function is defined and denoted as:

$$\boldsymbol{K}_{Z}^{j'j} = \boldsymbol{K}_Z(\boldsymbol{x}_j - \boldsymbol{x}_{j'}) = \operatorname{Cov}\left(Z_x(\boldsymbol{x}_i), Z_x(\boldsymbol{x}_{i'})\right) \tag{2.7}$$

The covariance function of the gradient is defined as:

$$\boldsymbol{K}_{G_x}^{i'i} = \boldsymbol{K}_{G_x}(\boldsymbol{x}_i - \boldsymbol{x}_{i'}) = \operatorname{Cov}\left(Z'_x(\boldsymbol{x}_i), Z'_x(\boldsymbol{x}_{i'})\right) \tag{2.8}$$

The cross-covariance function of the gradient is defined as:

$$\boldsymbol{K}_{G_xG_y}^{i'i} = \boldsymbol{K}_{G_xG_y}(\boldsymbol{x}_i - \boldsymbol{x}_{i'}) = \operatorname{Cov}\left(Z'_x(\boldsymbol{x}_i), Z'_y(\boldsymbol{x}_{i'})\right) \tag{2.9}$$

The cross-variance function between scalar field and its gradient is defined as:

$$\boldsymbol{K}_{ZG_x}^{ij} = \boldsymbol{K}_{ZG_x}^{ij}(\boldsymbol{x}_j - \boldsymbol{x}_{j'}) = \operatorname{Cov}\left(Z(\boldsymbol{x}_j), Z'_x(\boldsymbol{x}_i)\right) \tag{2.10}$$

where $j'j'' \in \mathcal{P}\left(J_{k'}\right)$. A comprehensive derivation for the kriging system is given in Lajaunie et al. (1997).

The universal drift matrix $\mathbf{U}$ is defined as

$$\mathbf{U} = \begin{bmatrix} 1 & 0 & 0 & x_j - x_{j'} \\ 0 & 1 & 0 & y_j - y_{j'} \\ 0 & 0 & 1 & z_j - z_{j'} \\ 2x_j & 0 & 0 & x_j^2 - x_{j'}^2 \\ 0 & 2y_j & 0 & y_j^2 - y_{j'}^2 \\ 0 & 0 & 2z_j & z_j^2 - z_{j'}^2 \\ y_j & x_j & 0 & x_j y_j - x_{j'} y_{j'} \\ z_j & 0 & x_j & x_j z_j - x_{j'} z_{j'} \\ 0 & z_j & y_j & y_j z_j - y_{j'} z_{j'} \end{bmatrix} \tag{2.11}$$

A continuous scalar field can be obtained by solving the co-kriging system. A 2D scalar field example is given in Figure 2.1. Additionally, faults can be represented by an extra term in the kriging function, and unconformities can be achieved by combining several scalar fields, which are interpreted separately (de la Varga et al., 2019). The geological model is represented by the iso-surfaces extracted from the scalar field by further using a discretization method. In addition, the interpreted scalar field function allows the scalar value at an arbitrary location to be queried individually, and this is fundamental for the proposed kernel method described in Chapter 3.

2.4 Stochastic Geological Modeling and Bayesian Framework

2.4.1 Stochastic Geological Modeling

Conventional 3D structural modeling often constructs only a single model based on the input data. However, both the input data and the modeling methods are prone to including mismatches to the truth, especially at the early stage of exploration when the data are sparse. Therefore, quantifying uncertainties in 3D structural models has drawn increasing attention in recent years (Wellmann et al., 2010; Wellmann and Regenauer-Lieb, 2012; Lindsay et al., 2012; Jessell et al., 2014). One method to quantify these uncertainties is based on stochastic geological modeling (Wellmann et al., 2010). In the stochastic modeling method, the input data is treated as stochastic variables. Instead of considering the raw data directly in the final model, the input

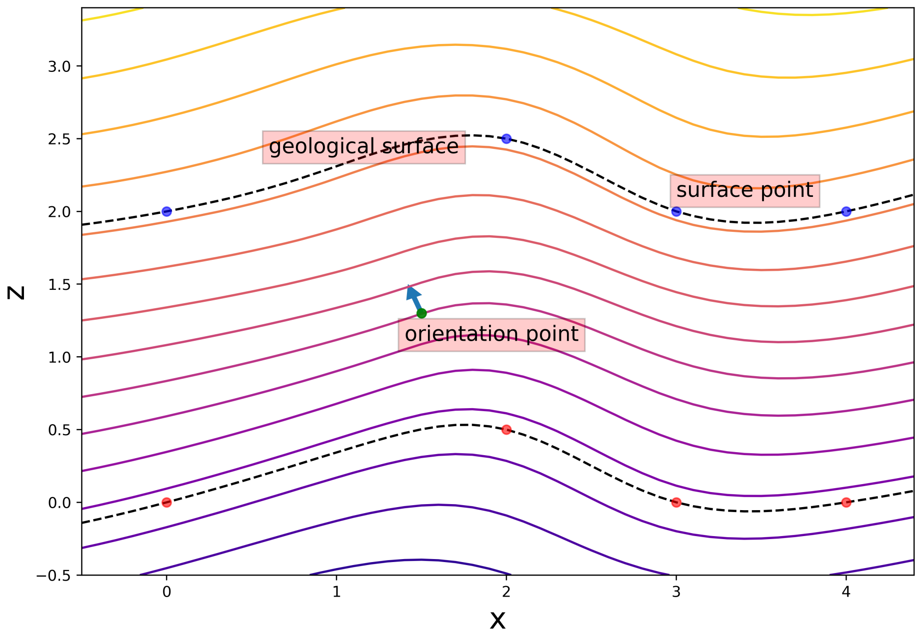


FIGURE 2.1: Interpolated scalar filed using co-kriging of interface points and orientation values. Isocurves represents the scalar values based on the interpolation of the surface points and orientations.

data is taken into the modeling procedure, and the geological model is constructed by simulation. The geological modeling process is treated as a deterministic forward process. Realizations of geological models can be generated by sampling from the input data distributions. Then, the ensembles of geological models can be used to evaluate the uncertainties in either a visual and qualitative manner (Suzuki et al., 2008; Mallet and Tertois, 2010; de la Varga et al., 2019) or a quantitative manner (Wellmann et al., 2010; Wellmann and Regenauer-Lieb, 2012). The implicit modeling method using the co-kriging method is naturally a good candidate for the uncertainty quantification purpose due to its low-dimensional parametric representation and efficiency in the model construction.

2.4.2 Bayesian Framework

Based on the stochastic modeling methods, de la Varga and Wellmann (2016) proposed a method to consider geological modeling as an inference problem. The Bayesian framework is proposed to consider additional information as well as the prior knowledge of the modeling parameters be the geological model construction. In the Bayesian framework, this additional information could come from diverse resources, including topological relationships (Schaaf et al., 2021), process simulations (Degen et al., 2022b), geophysical observations (Wellmann et al., 2018), deterministic functions on the constructed model (e.g., layer thickness) (de la Varga and Wellmann, 2016), other criteria based on decision theory (Stamm et al., 2019), etc. In this study, we focus on the geophysical observations due to their significance linked to geophysics.

The geophysical observations $\boldsymbol{d}_{obs} \in D$ in any forward problem can be described by $\boldsymbol{d}_{obs} = \boldsymbol{f}(\boldsymbol{m}) + \boldsymbol{e}_d$, where $\boldsymbol{m} \in M$ is the set of model parameters, $\boldsymbol{f}$ is the parameter-to-observation map that represents the modeling or simulation procedure. The error $\boldsymbol{e}_d$ denotes the misfit between the observation and the forward simulation. Various sources could cause the misfit, for example, sparse and erroneous data raised from $\boldsymbol{d}_{obs}$, imperfect forward modeling, and loss of information in the parameter-to-observation map in $\boldsymbol{f}$. This often leads to ill-posed Geophysical inversion problems (Tikhonov and Arsenin, 1977). Therefore, in many cases, we are interested in not only the best-fit parameter sets but also other plausible solutions that fit the observations within an uncertainty range and include our prior geological knowledge. Solving such inversion problems then amounts to exploring the distribution of plausible model parameters.

Bayesian inference starts from Bayes' theorem,

$$p\left(\boldsymbol{m} \mid \boldsymbol{d}_{obs}\right) = \frac{p\left(\boldsymbol{d}_{obs} \mid \boldsymbol{m}\right) p(\boldsymbol{m})}{p\left(\mathbf{d}_{obs}\right)} \tag{2.12}$$

where $p\left(\boldsymbol{d}_{obs} \mid \boldsymbol{m}\right)$ is known as the likelihood of the parameters, and $p\left(\boldsymbol{m}\right)$ represents the prior probability of the model parameters $\boldsymbol{m}$. The posterior is normalized by the denominator $p\left(\boldsymbol{d}_{obs}\right)$ so that the integral of the right-hand side is equal to one. The denominator is called the marginal likelihood or evidence. However, the denominator, is intractable in most cases because of the infinite possibility of observations. Sampling methods, such as MCMC, are therefore often used to approximate the integrals by sampling the posterior space. Here we briefly review two commonly used MCMC methods, random-walk Metropolis-Hasting (RMH), which is derivative-free, and Hamilton Monte Carlo (HMC), which is based on the first-order derivatives. These two methods have been widely adopted in many applications in other fields and are state-of-the-art methods in the geological modeling field. I will have a brief review of the two algorithms.

2.4.3 Markov Chain Monte Carlo (MCMC)

In the RMH algorithm, which is the most well-known and intuitively simple MCMC algorithm, a chain of samples is drawn based on a Gaussian proposal distribution, and by an accept/reject step, the sampling chain can move from the previous state to a new state based on an acceptance probability a. The accept/reject step restores the balance of the reversibility condition, which makes the Markov kernel invariant with respect to the probability of interest; therefore, the target distribution can be represented by the chain of samples. The steps of the RMH can be expressed as follows:

```
Set $k = 0$ and initiate with $\boldsymbol{m}^{(0)}$
while $k <$ number of samples do
    Propose $\boldsymbol{m}^{cand} = \boldsymbol{m}^k + \beta\boldsymbol{\xi}^k$, $\boldsymbol{\xi} \sim \mathcal{N}(0, C)$
    Set $\boldsymbol{m}^{k+1} = \boldsymbol{m}^{cand}$ with acceptance probability $a$
    Set $\boldsymbol{m}^{k+1} = \boldsymbol{m}^k$ otherwise
end
```

Algorithm 1: METROPOLIS-HASTING ALGORITHM

where acceptance probability $a(\boldsymbol{m}^{cand}, \boldsymbol{m}^k) = \min\left(1, \frac{p(\boldsymbol{m}^k|\boldsymbol{d}_{obs})}{p(\boldsymbol{m}^{cand}|\boldsymbol{d}_{obs})}\right)$, β is the step size, and C is the covariance matrix of the proposal distribution. In the random-walk method, a symmetric proposal is used. Therefore C is an identity matrix.

The random-walk method has limitations in cases of high dimensionality and high correlations. Gradient information is therefore employed to assist in posterior exploration. The most popular gradient-based method, the HMC algorithm, is designed to draw independent samples and, therefore, efficiently explore the state space (Duane et al., 1987; Neal, 1993; Chen et al., 2014; Betancourt, 2017). HMC introduces auxiliary momentum variables $\boldsymbol{r}$, with which we can express the Hamiltonian of a particle as:

$$H(\boldsymbol{m}, \boldsymbol{r}) = K(\boldsymbol{r}) + V(\boldsymbol{m}) \tag{2.13}$$

where K and V are kinetic and potential energies respectively and are defined as:

$$K(\boldsymbol{r}) = \frac{1}{2}\sum_{i,j=1}^{n} r_i \boldsymbol{M}_{ij}^{-1} r_j \tag{2.14}$$

$$V(\boldsymbol{m}) = -\log p(\boldsymbol{m} \mid \boldsymbol{d}) \tag{2.15}$$

here, the mass matrix $\boldsymbol{M}$ is often set as the identity matrix (Chen et al., 2014)

Hamiltonian dynamics can be simulated by following Hamilton's equations with an artificially introduced time variable t:

$$\begin{aligned} \frac{\mathrm{d}\boldsymbol{m}}{\mathrm{d}t} &= \frac{\partial H}{\partial \boldsymbol{r}} \\ \frac{\mathrm{d}\boldsymbol{r}}{\mathrm{d}t} &= -\frac{\partial H}{\partial \boldsymbol{m}} \end{aligned} \tag{2.16}$$

The HMC algotithm requires that Hamilton's equations are integrated using a symplectic integrator. In practice, the "leapfrog" integrator is often used. The algorithm can be summarized as follows (Chen et al., 2014):

```
Set $k = 0$ and initiate with $\boldsymbol{m}^{(0)}$ and step size $\epsilon$
while $k <$ number of samples do
    Sample momentum $\boldsymbol{r}^k \sim \mathcal{N}(0, \boldsymbol{M})$
    Set $(\boldsymbol{m}_0, \boldsymbol{r}_0) = (\boldsymbol{m}^k, \boldsymbol{r}^k)$
    Simulate discretization of Hamiltonian dynamics in Eq. 2.16:
    $\boldsymbol{r}_0 = \boldsymbol{r}_0 - \frac{\epsilon}{2}\nabla V\left(\boldsymbol{m}_0\right)$
    for i=1,i< leapfrog steps $n$ do
        $\boldsymbol{m}_i = \boldsymbol{m}_{(i-1)} + \epsilon \boldsymbol{M}^{-1}\boldsymbol{r}_{i-1}$
        $\boldsymbol{r}_i = \boldsymbol{r}_{i-1} - \epsilon\nabla V\left(\boldsymbol{m}_i\right)$
    end
    Set $\boldsymbol{r}_n = \boldsymbol{r}_n - \frac{\epsilon}{2}\nabla V\left(\boldsymbol{m}_n\right)$
    Propose $(\boldsymbol{m}^{cand}, \boldsymbol{r}^{cand}) = (\boldsymbol{m}_n, \boldsymbol{r}_n)$
    Set $\boldsymbol{m}^{k+1} = \boldsymbol{x}^{cand}$ with acceptance probability $a$
    Set $\boldsymbol{m}^{k+1} = \boldsymbol{m}^k$ otherwise
end
```

Algorithm 2: HAMILTONIAN MONTE CARLO ALGORITHM

where the acceptance probability $a(\boldsymbol{m}^{cand}, \boldsymbol{m}^k) = \min\left(1, e^{-H(\boldsymbol{m}^{cand}|r^{cand})+H\left(\boldsymbol{m}^k|r^k\right)}\right)$

RMH is simple to implement and gradient-free, while HMC uses gradient information to obtain low autocorrelated samples; however, both methods are popular and widely applied, and we will test our method using these two algorithms in the following work.

Chapter 3

Gravity Simulation

A manuscript with the text of this chapter has been published in *Geophysics*.

3.1 Introduction

The gravity method as a fundamental geophysical method has been actively developed in recent decades and has been playing an important role in subsurface exploration (Nabighian et al., 2005a; Hinze et al., 2013; Klitzke et al., 2016; Rodriguez Piceda et al., 2021). A typical workflow of gravity exploration involves the collaboration of both geophysicists and geologists. A 3D geological model is often combined with the observed geophysical data. The model is then used to perform numerical simulations for validation and predictions of the model developed. The process of constructing the 3D geological model and conducting geophysical simulation on this model is called the forward process. The inverse process means inferring the properties of the subsurface from observations (e.g., gravity data).

The uncertainties from developing the 3D model and from conducting simulations on this model have recently received increasing attention (Wellmann and Regenauer-Lieb, 2012; Pirot et al., 2015; Linde et al., 2017). The success of gravity exploration requires a better control of the sources of uncertainties arising from both the forward modeling and the inversion process.

Forward gravity modeling often begins with the construction of a 3D subsurface structural model. The rock densities are essential for the preparation of the gravity simulation. A widely used approach is the so-called voxel-based method, in which voxel meshes or unstructured grids represent the subsurface units and rock properties are directly assigned to each of the voxels or grid cells. Gravity can then be simulated through using closed-form solutions for a regular voxel (Nagy, 1966) or numerical solutions such as finite difference (FD) (e.g., Farquharson and Mosher, 2009) finite volume (FVM) and finite element (FEM) methods (e.g., Jahandari and Farquharson, 2013; Rücker et al., 2017). Although the directly implemented forward gravity simulation on voxels has the advantage of representing the heterogeneities of the density distribution in the subsurface, the inversion conducted directly on voxels, however, suffers from the high-dimensional parameters, and the smoothness of the property field due to pool regularizations; therefore a direct geometrical interpretation is unsuitable and additional regulations are often required (Li and Oldenburg, 1998).

An alternative to the voxel-based method and a widely applied approach to modeling the subsurface follows the intuition of stratigraphy geology interpretation, which constructs the boundaries of individual stratigraphy units (Chapter 2). The continuous model is then projected onto a discrete volumetric mesh for geophysical simulation. In contrast to the direct inversion on the voxels, surface-based methods will naturally satisfy the geological interpretation. Surface-based modeling is suitable for structural modeling, in which the each stratigraphy has distinct properties. Geophysical inversion then act as a constraint on the 3D structure in surface-based modeling (Jessell, 2001).

As described previously in Chapter 2, the Bayesian perspective allows the prior geological knowledge and additional geophysical data to be combined in the modeling procedure in a probabilistic manner. Therefore, uncertainties of the geological model considering the geophysical observation can be estimated by evaluating the posterior distribution (Wellmann et al., 2010; Jessell et al., 2010; Lindsay et al., 2012). With the goal of accelerating the posterior exploration, further advances in modeling software introduce Automatic Differentiation (AD) (de la Varga et al., 2019), a computational technique that enables a rapid evaluation of derivative, into geological modeling and make the derivative informed Markov Chain Monte Carlo (MCMC) methods (e.g., Betancourt, 2017; Liu and Wang, 2016) feasible in geological modeling (Güdük et al., 2021). Our study follows this implicit probabilistic structural modeling method through universal co-Kriging (Lajaunie et al., 1997).

Implicit modeling method as a 'mesh-free' method, a pre-defined mesh is not required during the early interpolation of the scalar field. However, projection on a discretized mesh is necessary for the later stage of visualization and geophysical simulation. Due to the simplicity and easily accessible analytical solution of potential field simulation, the tensor grid is often adopted in the potential field simulation applications (Li and Oldenburg, 1998; Boulanger and Chouteau, 2001; Fournier, 2019; Scalzo et al., 2021, e.g.,) and software, e.g., SimPEG (Cockett et al., 2015), Geomodeller (Intrepid Geophysics), pynoddy (Wellmann et al., 2016) etc., using either voxel-based or implicit modeling. We refer to the specific subclass of the tensor grid with an identical shape for each voxel as a regular grid, which is an even more common and easier option due to the simplicity and intuition (Scalzo et al., 2021). Throughout, cell and voxel will be used interchangeably to represent the unit element of the grid. When a regular grid is used, the user generally assumes no prior knowledge of the voxel alignment with the interpreted geological structure. The lithologies at each grid point of the model can be queried by the coordinates of the centers of the voxels based on the interpolated scalar field. Computational visualization techniques such as marching cubes are then applied to generate relatively smooth surfaces. Based on the interpreted rock density, gravity response can also be simulated by integrating the grid points using the closed-form solution (Nagy, 1966). To eliminate the boundary effect, an additional padding zone is often required to be added at the boundary of the model (e.g., Boulanger and Chouteau, 2001; Rashidifard et al., 2021; Cockett et al., 2015). The gravity anomaly is then computed sequentially at each receiver at the ground surface. However, in a real-world application, gravity data are sometimes sparse or highly correlated. A surface gravity field is then often interpolated based on scattered point observations (e.g., Chasseriau and Chouteau, 2003; Rashidifard et al., 2021). Therefore we argue that not all the data are available or necessary to be included in the inversion due to the data correlation. Hence, to remove the requirement to perform a 2-D interpolation first, we propose a kernel method. The kernel method constructs an independent grid centered at each measurement location. This method benefits from the fact that lithologies can be independently queried at locations of interest based on the interpolated scalar field in the implicit modeling method. The kernel method can save computational resources by only conducting computation within the regions which are relevant to the gravity receivers. In addition, conventional gravity methods using a regular grid are prone to slicing errors (described in the following sections). This will lead to a imprecise forward simulation and therefore a poor inversion result. Improvements in gravity forward algorithm to reduce the slicing errors are being actively developed, e.g., in the space domain, Ren et al. (2018); Zhang and Jiang (2017) used closed-form solutions for prismatic and polyhedra cells with variable density; in the Fourier-domain, Wu and Chen (2016); Wu (2018) used a hybrid rectangular-Gaussian grid to reduce the slicing errors. The kernel method also has more faithful simulation results compared to conventional methods by naturally alleviating the slicing errors, and padding zones (which are often required to eliminate the boundary effect) are embedded in each individual kernel. This concept of the local kernel has been adopted previously by applications in the field of airborne electromagnetic (AEM) surveys. For example, Cox et al. (2010) introduced the 'footprint' method where only the local sensitivities matrix within limited spatial extent is constructed and used in the inversion for each iteration; following that Yang et al. (2014); Haber and Schwarzbach (2014) used a global mesh and a local mesh for each sounding. Apart from the difference in the properties of interest, the main difference here is instead of directly using the local sensitivities matrix in the inversion, there is an implicit structural modeling step involved. Also, thanks to the implicit modeling method, there is no need for constructing a global mesh and projecting on the local mesh, and there is a lot more flexibility.

While the quality of the simulation is heavily dependent on the chosen resolution, a high resolution can be prohibitive due to the limitation of available computational resources, especially in the probabilistic geological modeling framework using AD. Refinement of the grid becomes a suitable solution to relocate the limited resources to focus on more impactful regions. Refinement strategies have been intensively studied in the field of gravity inversion for the past decades, mainly in the context of voxel-based inversion (Fournier, 2019; Astic et al., 2020). However, many of these methods require prior knowledge of the geometry of the subsurface and some intermediate remeshing step will increase the numerical challenges and difficulties embedded in the AD framework. In the case of implicit geological modeling, refining the grid based on the geometry is not straightforward before interpolating the data. Additionally, the probabilistic geological method will iteratively update the structural model. Hence, we introduce an optional subsequent refinement strategy based on the proposed kernel method. We refine a tensor mesh based on the sensitivity of each voxel to the individual receiver. This method assigns high resolution to the more sensitive part of each kernel and lower resolution to the distant region. We will use two numerical examples to show the efficiency and improved accuracy of the proposed kernel methods.

3.2 Methods

3.2.1 Gravity forward simulation

Gravity data has been one of the most important geophysical data and there is increasing amount of data been collected and available in recent years. As previously described, gravity data can be included in the modeling procedure as the observational data $\boldsymbol{d} \in \mathcal{D}$ in a probabilistic or optimization framework to assess the quality of a geological model. The gravity forward model can be described as

$$\boldsymbol{d} = \boldsymbol{f}(\boldsymbol{m}) + \boldsymbol{e}_d \tag{3.1}$$

where $\boldsymbol{m} \in \Theta$ is the model parameters, $\boldsymbol{f}$ is the parameter-to-observation map that represents the modeling or simulation procedure. The error $\boldsymbol{e}_d$ denotes the misfit between the observation and the forward simulation. Various sources could cause the misfit, for example, sparse and erroneous data raised from $\boldsymbol{d}$, imperfect forward modeling and loss of information in the parameter-to-observation map in $\boldsymbol{f}$. Hence, the quality of the forward simulation is an essential factor for the success of the stochastic geological modelling. In the following sections, we will introduce our methodology to improve the $\boldsymbol{f}$ under the framework of implicit modeling methods.

A typical geophysical simulation of a geological model can be summarized by the following Figure 3.1. Voxel-based modeling methods, as described by the name, are initialized with discretized voxels. Rock properties are then assigned to each cell to calculate the geophysical simulation. In contrast, surface modeling methods initiated with constructing the interfaces of the model and then a projection of the continuous surfaces on the discretized grid. The geophysical simulation can then be solved by numerical methods (FEM, FD) or by breaking the problem into several discrete object modeling problems and solved by integrals of direct analytical solutions. The forward model $\boldsymbol{f}$ can be described in a chain of two forward models (Scalzo et al., 2021) as $\boldsymbol{f} := \iota \circ q$: the map q: $\Theta \to \mathcal{G}$ which projects the parameter space Θ into a discrete volumetric representation in space $\mathcal{G}$ followed by the map ι: $\mathcal{G} \to \mathcal{D}$ which projects the discretizations into the observational data space. One of the characteristics that

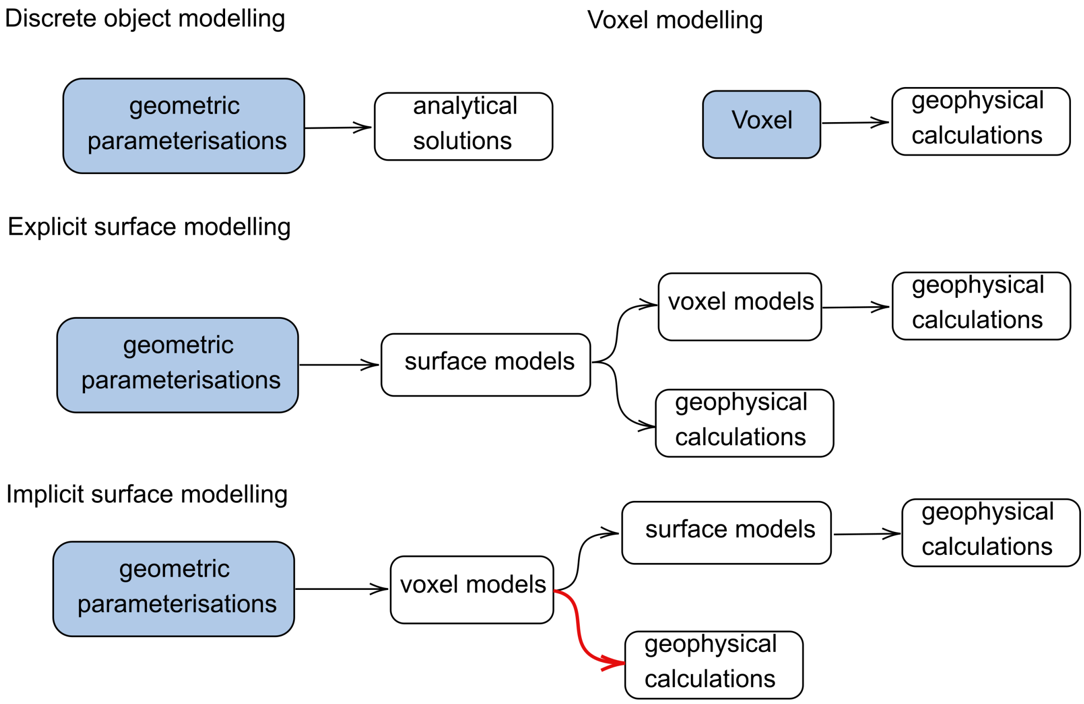


FIGURE 3.1: Graph representation of typical gravity forward approaches. The top two methods are the elementary methods for gravity simulation. Here the 'geophysical calculations' represents either analytical solution or numerical solutions. In explicit surface modeling, the surface model will firstly be constructed, and then either a discretization will be conducted or the gravity can be directly calculated through methods, for example, line integrals (Zhou, 2009), the Gauss-fast Fourier transform (FFT) method (Wu and Chen, 2016) and analytical formulas (Zhang and Jiang, 2017; Ren et al., 2018). Implicit modeling is a 'mesh-free' method, and discretization is only involved during visualization or geophysical computation. The surface model can be extracted from the implicit function, and the same technique as in explicit modeling can be used for geophysical calculation. Whereas implicit modeling can avoid explicit surface construction by directly performing geophysical calculations on the discretization (as shown by the red arrow).

differentiate the implicit method from other surface modeling methods is that the step of explicit surface construction is not necessary during forward modeling. One could directly find the map q by, for example, by utilizing the co-kriging methods described above.

Therefore, in cases of many-query problems, computation resources used for the step of explicit surface construction can be saved by using directly implicit modeling methods until for the visualization for the final result (Figure 3.1). This is fundamental for the proposed method.

The gravitational field is the conservative field where its norm $\boldsymbol{g}$ is often referred to as gravity in geophysics. $\boldsymbol{g}$ can be calculated following the integration of an elementary volume dV and density ρ

$$\boldsymbol{g}(\boldsymbol{r}) = -G\int_V \rho(\tilde{\boldsymbol{r}})\frac{\boldsymbol{r}-\tilde{\boldsymbol{r}}}{|\boldsymbol{r}-\tilde{\boldsymbol{r}}|^3}dV \tag{3.2}$$

where $\boldsymbol{r} = (x, y, z)$ is position vector of the computation point, $\tilde{\boldsymbol{r}} = (\tilde{x}, \tilde{y}, \tilde{z})$ is position vector of the running integration point; G is the gravitational constant. Although

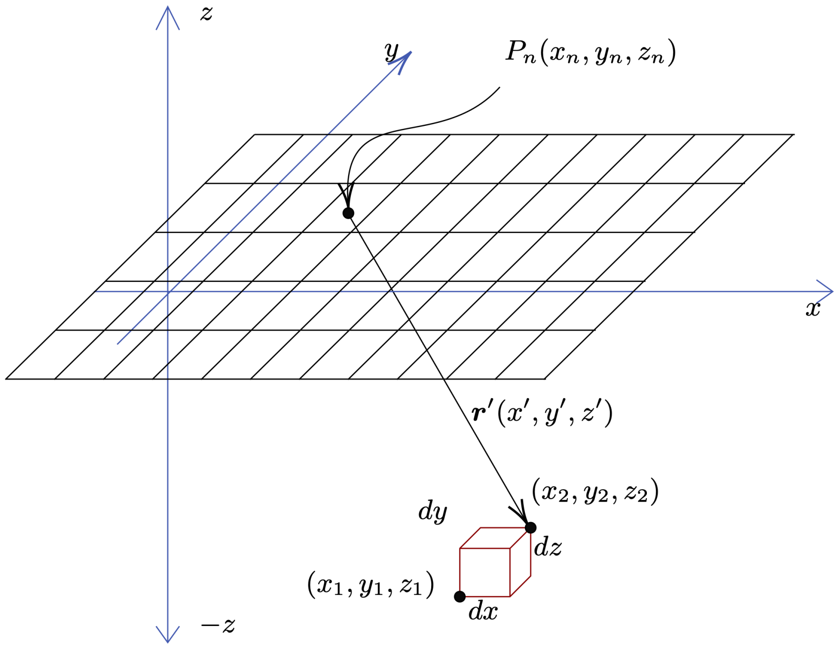


FIGURE 3.2: Computing the gravity contribution of a rectangular prism. Here the positive z are chosen to be pointing upwards for the consistency with elevation input data.

numerical methods (e.g., FEM, FVM) can be applied based on an irregular mesh (e.g. Jahandari and Farquharson, 2013) for 3D gravity inversion, regular grids are mostly adopted by modelers. Regular grids are dominating the application in gravity field simulation due to their simplicity in data storage and visualization (Fedi and Rapolla, 1999; Li and Oldenburg, 1998; Portniaguine and Zhdanov, 1999; Commer, 2011; Čuma et al., 2012; Chen and Liu, 2019). Because the existence of the closed-form solution, it can conserve the AD framework. A rectangular prism at location $\boldsymbol{x_m}$ can be defined by the two diagonal corners $\boldsymbol{x_{m1}} = (x_{m1}, y_{m1}, z_{m1})$ and $\boldsymbol{x_{m2}} = (x_{m2}, y_{m2}, z_{m2})$ (shown in Figure 3.2). The closed-form solution of the vertical component of the gravitational attraction at the nth measurement location $P_n = (x_n, y_n, z_n)$ caused by the rectangular prism with limit $x_1 \leq x' \leq x_2$, $y_1 \leq y' \leq y_2$ and $z_1 \leq z' \leq z_2$ can be calculated by the the following equations (Nagy, 1966).

$$g_{zmn} = t_{zm}\rho_m \tag{3.3}$$

$$t_{zm} = -|||x' \ln(y' + r') + y' \ln(x' + r') - z' \arctan\left(\frac{x'y'}{z'r'}\right) |_{x_1}^{x_2}|_{y_1}^{y_2}|_{z_1}^{z_2} \tag{3.4}$$

where

$$r' = |\boldsymbol{r}'| = \sqrt{x'^2 + y'^2 + z'^2} \tag{3.5}$$

and ρ is the density of the cell, (x', y', z') are the Cartesian components from the measurement point P_n to $\boldsymbol{x_m}$. All the coordinates in this study will have the vertical axis z pointing upwards to keep consistency with the input elevation data. The synthetic gravitational attraction at the nth receiver can then be calculated as the summation of the contribution of all the cells in the modeling domain.

$$g_{zn} = \boldsymbol{t_z}\boldsymbol{\rho} \tag{3.6}$$

where $\boldsymbol{t_z} = (t_{z1}, t_{z2}, \ldots, t_{zM})$ and $\boldsymbol{\rho} = (\rho_1, \rho_2, \ldots, \rho_M)^T$.

All the existing methods construct the 3D grids for the entire domain $\boldsymbol{D}$ as a whole, regardless of the discretization methods. Additionally, if the region outside the modeling area is considered to have zero density, this will generate an artificial anomaly at the boundaries. A padding zone is often used to alleviate the distortion caused at the boundaries, which further extends the domain of modeling, hence increasing the computational resource consumption (e.g., Figure 3.3a). A conventional method to simulate the gravity from the grid geological model is either using all cells in the model to do the summation in Equation 3.6 or to choose a range of cells around the receiver to be included in the corresponding summation. To make a fair comparison of our method, we implement in this study the spatial convolution scheme (Jessell and Valenta, 1996; Wellmann et al., 2016) illustrated in Figure 3.3a. The computation in Equation 3.6 is conducted repeatedly through each receiver by summing the contribution in a windowed subdomain D_n, where $n \in N$, N is the number of receivers. This leads to the following problems:

1. the receiver positions need to be configured at a distance from the edge of the cell to avoid stability issues. This added more conditional operations in AD and is also error-prone.

2. the actual range of cells included in the computation is dependent on the resolution. This will result in a row of cells being abruptly excluded or included in the subdomain D_n and therefore add error to the computation. This is more significant when the grid is coarse.

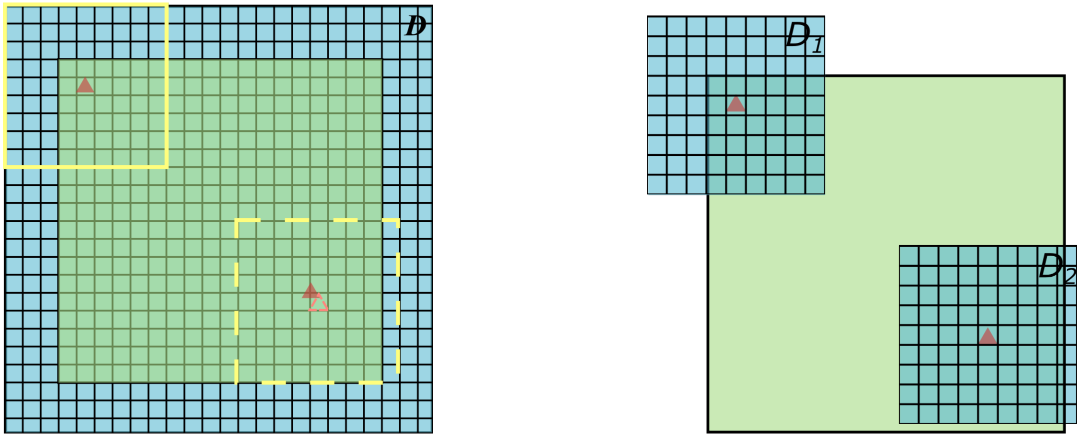
3. in the case of sparse data, for example, as shown by the sketch in Figure 3.3a, many cells are not included in the computation of gravity, however still initialized in the interpolation and cached in memory and therefore the computation resource on those cells is wasted.

4. Equation 3.6 is revisited N times.

Hence we propose the kernel method aiming to improve the existing gravity simulation algorithm described in the following section.

3.2.2 Kernel methods

We develop the kernel based on individual receiver locations (as shown in Figure 3.3b). In this method, the mesh is only constructed for a subdomain, or a kernel, D_n in a certain defined range around each receiver P_n. The name is given after the kernel of



(a) Spatial convolution scheme

(b) Kernel method

FIGURE 3.3: Sketch comparison of spatial convolution scheme and kernel method. Green box is the modeling area. Blue box is the actual meshing area. Red triangles are the receiver locations (a) In the spatial convolution scheme, a padding zone is often added around the model block to eliminate border effects. Gravity is simulated by a defined convolution window represented by the yellow box. The dashed-line objects represents cases where the receiver is not aligned at the center of a cell, then the actual simulated window is deviated. (b) Kernel method do not need to generate grid for the entire model, but around the area of a receiver.

image preprocessing, where the kernel denotes the convolution matrix or mask used to convolute through the image (Lecarme and Delvare, 2013). It can also be interpreted by the meaning of 'kernelization' in computer science, which stands for algorithms that achieve efficiency by a preprocessing stage to replace the inputs by a smaller input (Abu-Khzam et al., 2017). The new modeling domain is then a collection of each subdomain $\boldsymbol{D'} = \{D_1, D_2, ..., D_n\} \subseteq \boldsymbol{D}$.

Then, considering all the N receivers in a matrix form, we can augment Equation 3.3 as:

$$\boldsymbol{g_z} = \boldsymbol{T_z}\boldsymbol{\rho} \tag{3.7}$$

$\boldsymbol{T_z} \in \mathbb{R}^{N\times M}$ is the linear forward operator maps the density $\boldsymbol{\rho} \in \mathbb{R}^M$ of M number of cells over the gravity measured at N number of observation locations $\boldsymbol{g_z} \in \mathbb{R}^N$. $\boldsymbol{T_z} = (\boldsymbol{t_{z1}}, \boldsymbol{t_{z2}}; \ldots, \boldsymbol{t_{zN}})^t$, where $\boldsymbol{t_{zn}} = (t_{z1}, t_{z2}; \ldots, t_{zM})^t$ is also referred to as the collection of sensitivity of each subdomain D_n. With the choice of identical kernel for each receiver, the sensitivity matrix $\boldsymbol{t_z}$ is also identical. The choice of identical kernel allows the $\boldsymbol{T_z}$ to be pre-computed outside of the computational graph and reused at each iteration of a many-query application (e.g., MCMC).

The kernel method improves the conventional spatial convolution algorithm in the following aspects. First, when available data is sparse or highly correlated, or in some applications (Güdük et al., 2021), only a limited amount of data need to be considered in the simulation, the collection $\boldsymbol{D'}$ is smaller than $\boldsymbol{D}$. Also, the padding zone is embedded in each kernel so that a global padding zone is not needed. Second, this vectorized computation converts the previous sequential computation to a single calculation and can simply parallelized. Third, the grid centered around the receiver in the kernel method avoids the singularity caused by a receiver accidentally aligning with the edge of a cell and eliminates the error of slicing the cells. In this study, we use an identical kernel for each gravity receiver, therefore, in addition to the easy preparation of $\boldsymbol{T_z}$ described above, it also makes further mesh refinement respecting each receiver possible (introduced in the next section).

3.2.3 Optimized centered kernel

As discussed above, different sources can cause the misfit e_d between the data and forward model. In many studies, a perfect forward model is assumed and therefore only the error from the data is considered in the likelihood. However, it is not possible to have a perfect model to represent reality. One source of error is the limited resolution in the discretized volumetric representation of the continuous geological structures, this is also referred to as the aliasing effect in some studies (Scalzo et al., 2021). With limited computation resources, simulation accuracy and simulation efficiency are not easy to be achieved simultaneously. Some refinement strategies are used to alleviate such an error. For example, an octree mesh is used to refine the mesh around the geometry boundaries and to assign coarser meshes to the non-boundaries area (Wellmann and Caumon, 2018; Fournier, 2019). However, aligning mesh cells along the boundaries of the geometry of interest is not easy before evaluating the model. Existing refinement strategies are also often case-based (Fournier, 2019; Scalzo et al., 2021).

We construct our grid in a tensor grid, which means the grid is divided into rectangular prisms by straight lines/surfaces across the whole grid. There are several reasons to use the tensor grid. As above discussed, it is challenging to align the mesh with the subsurface geometry before interpreting the model. Hence, it is not only easier to implement and store the data, but the tensor grid is also more general for an unknown geometry. The easily accessible analytical solution is suitable for the AD framework. The ability to easily parameterize and therefore optimize adds a further advantage. As a specific configuration of the tensor grid, the regular grid is the simplest one to implement and visualize. However, in the application of gravity simulation, using a regular grid assigns the same resolution to the close and distant regions which have different sensitivities to a receiver at a given location. This results in that, with a limited resolution, the area with high sensitivity to the receiver is treated the same as the lower sensitivity region. As described in Equation 3.1, e_d is consists of uncertainties caused by the measurement device e_{dm} and forward model e_{df}. With a given measuring device, the error from the data measurement e_{dm} is fixed. The total error is then determined by the error of forward model e_{df}. When the boundary of the density contrast is not known while building the grid, this will result in that the risk of an aliasing effect appearing close to the receiver is the same as the distant region, while the impact of the aliasing is more significant to the result at the former and trivial at the later.

To address this problem, an intuitive way is to assign a higher resolution to the more sensitive region. The kernel method centers the mesh according to the location of the receiver, therefore allowing a lateral grid refinement in addition to the depth-dependent refinement. With the assumption of no prior knowledge of geometry before interpolating the model, we focus only on the $\boldsymbol{t}_z$ in Equation 3.3. Therefore, averaging the risk of an aliasing effect is essentially finding a configuration of the grid, which has a more uniform sensitivity $\boldsymbol{t}_z$ for all the cells. Then the objective is to distribute the limited resolution accordingly to the sensitivity $\boldsymbol{t}_z$, hence the whole subdomain D_n will have a more balanced risk over all the cells. We can formulate the objective function as the standard deviation σ_t of $\boldsymbol{t}_z$, then the optimized solution is the grid with minimized σ_t. In the ideal case, every cell will have the same $\boldsymbol{t}_z$, therefore $\sigma_t = 0$. In the 3D scenario of the tensor grid, finding the optimal configuration is challenging. Hence, we introduce the parametric representation of the grid and the Machine Learning (ML) methods to solve the optimization problem.

First, we introduce how we parameterize the grid and then how we could optimize the grid. For simplicity and legibility, we use a 1D grid as an example for demonstration. Due to the grid symmetry, we can consider only half of the axis. Given user-defined resolution $2k - 1$, with $k > 0$ and extent R. We define a vector $\boldsymbol{a} = \{a_0, a_1, \ldots, a_k\}$. The origin $a_0 = 0$ is at the middle of the center cell. We denote the distance from the $(q-1)$th node to the qth node as $a_q \geq 0$. Therefore in one direction of the grid, there are k number of cells $\{a_1, a_2, \ldots, a_k\}$, as illustrated in Figure 3.4. By parameterizing only the spacing for each dimension, the grid will be subject to a tensor grid automatically in any degree of dimensionality. Then following this parameterization, the 1D regular grid with uniform cell size can be then written as:

$$a_q = \begin{cases} \frac{2R}{(2k-1)} & , q = 2, 3, \cdots k \\ \frac{R}{(2k-1)} & , q = 1 \end{cases} \tag{3.8}$$

Inspired by the half-sphere geometry of the sensitivity matrix and the radial decaying property of gravity respecting the distance to the object in Equation 3.2, an empirical exponential refinement of the grid with the cell length as a geometric progression of sequence has been successfully adopted in the recent work of Güdük et al. (2021). This empirical exponential refinement strategy can be summarized as following:

$$\sum_{q=1}^{k} a_q = \epsilon \cdot 10^{\left(\frac{\log_{10} R - \log_{10} \epsilon}{k}\right) q} \tag{3.9}$$

where $\varepsilon \ll R$ is a small number to maintain the stability.

Although the exponential grid distributes more resolution to the center part of the grid (Figure 3.5), it has not reached the optimal criteria for $\boldsymbol{t}_z$ we described above. In order to find an optimized configuration of the grid, following the parameterization

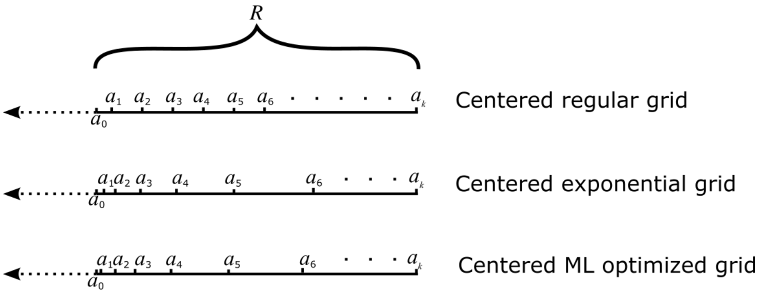


FIGURE 3.4: Simple sketch illustration of grid parameterization of different grid scheme.

above, we can formulate this as the following optimization problem:

$$\begin{aligned} &\text{Min} \quad \sigma(S(\boldsymbol{a})) = \sigma(\boldsymbol{t_z}) = \sqrt{\tfrac{\sum(t_{zq}-\bar{t}_z)^2}{k}} \\ &\text{s.t.} \quad \textstyle\sum_{q=1}^{k} a_q = R \quad, q = 1, 2, \cdots, k \\ &\qquad\ a_q \geq 0 \quad, q = 1, 2, \cdots, k \end{aligned} \tag{3.10}$$

where t_{zq} denotes the sensitivity $\boldsymbol{t_z}$ of qth cell, which can be calculated following Equation 3.4 and Equation 3.5 for the 3D cases. Here for simplicity, $\boldsymbol{t_z}$ does not exist in a 1D grid, therefore we denote the map from the parameter of the grid $\boldsymbol{a}$ to the sensitivity $\boldsymbol{t_z}$ using a symbolic notation S. We denote the standard deviation as σ. Solving this optimization or finding a deterministic solution is not easy, especially in 3D where there are two more dimensions to consider. Therefore, we program the grid in AD and utilize the gradient evaluated from AD to solve this optimization problem. In order to further simplify the computation, instead of defining a sequence of positive numbers which subject to the constrain of the summation $\sum_{q=1}^{k} a_q = R \quad, q = 1, 2, \cdots, k$, we design it using the softmax function ξ, which is used widely in Machine learning applications of probability (e.g. Howard et al., 2017). The softmax matrix has the following formula:

$$\xi_q(\boldsymbol{u}) = \frac{e^{u_q}}{\sum_{\psi=1}^{k} e^{u_\psi}} \tag{3.11}$$

where $q = 1, \ldots, k$, and $\boldsymbol{u}$ is the input vector which has element $\{u_1, u_2, \ldots, u_k\}$, k is the total number of elements same as above. The softmax function transfers an arbitrary sequence of numbers to a positive sequence with a sum of 1. This property allows the resulting sequence from the softmax function automatically fulfill the constraints in our optimization problem and result in one loss function to optimize for. By adopting the softmax function, the elements of the vector can be represented by

$$a_q = R \cdot \xi_q(\boldsymbol{u}) \tag{3.12}$$

Next, we can compute the sensitivity matrix and loss function S. The gradient of loss S with respect to the input vector $\boldsymbol{u}$ can be calculated by AD. Then we can use any robust iterative optimization algorithm to solve this optimization problem. Here we adopted Adam (Kingma and Ba, 2014), which is a gradient-based optimization method to update $\boldsymbol{u}$. We summarize the optimization algorithm below

```
Initialize a random vector $\boldsymbol{u} = \{u_1, u_2, \ldots, u_k\}$, $u \sim U(-1, 1)$
while step < Number of steps do
    Convert $\boldsymbol{u}$ to $\boldsymbol{a}$ by using softmax function $a_i = R \cdot \xi(\boldsymbol{u})_i$
    Compute the loss $S(\boldsymbol{a})$
    Compute the gradient $\frac{\partial S}{\partial \boldsymbol{u}}$ through AD
    Update $\boldsymbol{u}$ following Adam algorithm (Kingma and Ba, 2014)
end
```

Algorithm 3: OPTIMIZATION OF THE TENSOR GRID

In a 3D case, we simply extend $\boldsymbol{u}$ to $\boldsymbol{u} = (\boldsymbol{u_x}, \boldsymbol{u_y}, \boldsymbol{u_z})$ and $\boldsymbol{a}$ to $\boldsymbol{a} = (\boldsymbol{a_x}, \boldsymbol{a_y}, \boldsymbol{a_z})$.

The user will need to define the resolution of the kernel and the window size, and the algorithm will optimize the grid spacing automatically until reaching a designed error or the maximum number of iterations. For a many-query application that uses a fixed grid scheme, this optimization will only need to be conducted once for each given resolution and window size and the resulting configuration can be reused without additional cost. To illustrate the grid configuration computed by the optimization algorithm, here we compare the centered optimized grid with a centered regular grid and a centered exponential grid. The cross-sections of two 3-D grid examples with resolutions of $21 \times 21 \times 10$ and $51 \times 51 \times 30$ are shown in Figure 3.5 and Figure 3.6. In both examples, both exponential and optimal refinement strategies provide a smaller standard deviation of $\boldsymbol{t_z}$ compared to that of an unrefined regular grid. This illustrates a more evenly distributed resolution respecting sensitivity. Moreover, the optimized grid keeps a more balanced distribution of the grid, avoids over-dense resolution at the top part compared to the empirical exponential refinement, and can eliminate the individual cells with large $\boldsymbol{t_z}$, which appear at the closest area to the receiver and is most sensitive to the simulation result.

3.3 Results

In this section, we first provide a synthetic gravity forward simulation example of a spherical geobody, for which the analytical solution is available, to demonstrate the kernel methods and the centered optimized grid, and compare it with the spatial convolution scheme. Then, we investigate the gravity forward simulation with different methods and grids on the Sandstone greenstone belt in Western Australia (Wellmann et al., 2018).

3.3.1 Synthetic spherical object

There are many different ways to describe the surface of a spherical object using the implicit method. For simplicity, here we represent the scalar field of a spherical object

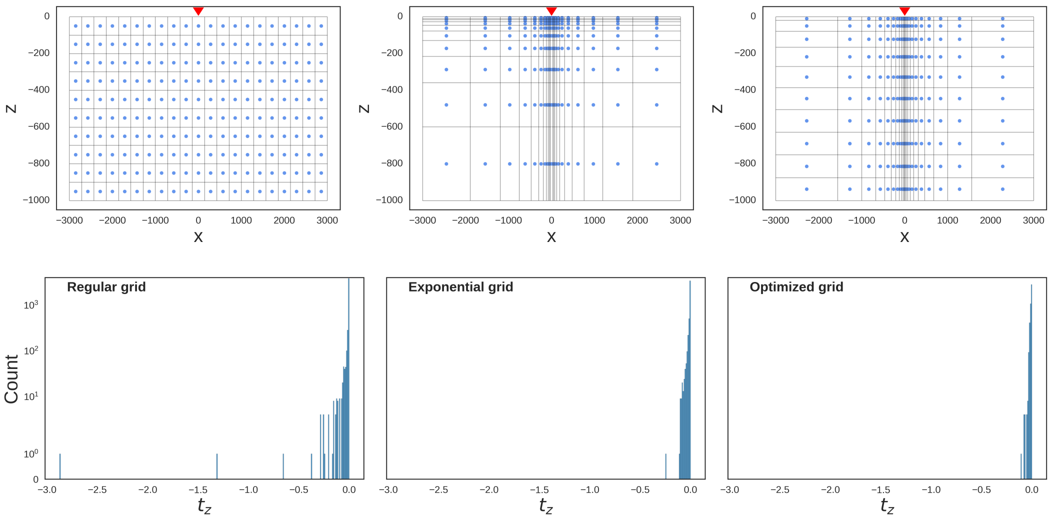


FIGURE 3.5: Upper row shows the cross section of the three different grid scheme with resolution $21 \times 21 \times 10$. The red triangle presents the position of receiver at the ground surface. The blue dots illustrates the center of the cell. Lower row shows the corresponding t_z histogram. The vertical axis is in log scale.

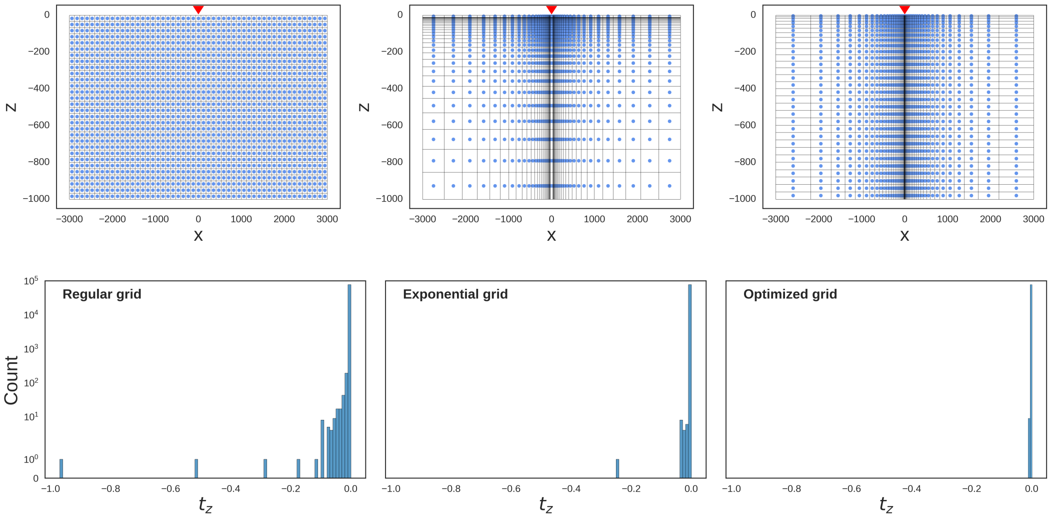


FIGURE 3.6: Upper row shows the cross section of the three different grid scheme with resolution $51 \times 51 \times 30$. The red triangle presents the position of receiver at the ground surface. The blue dots illustrates the center of the cell. Lower row shows the corresponding t_z histogram. The vertical axis is in log scale.

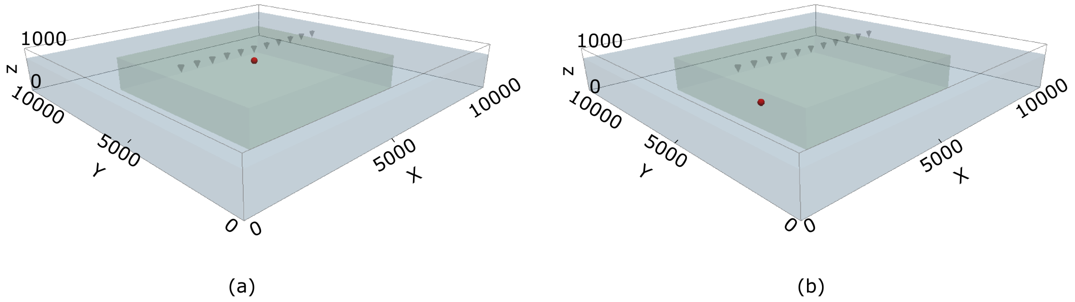


FIGURE 3.7: 3D synthetic spherical geobody (red) with identical y coordinates at 5000 m, radius of 100 m and different x and elevation, the inner green box is the modeling domain, the outer blue box represents the actual meshing domain with an additional padding zone. Receivers are placed at the top surface

in cartesian coordinates centered at x_c, y_c, z_c by a radial function

$$\mathbf{Z}_{sphere} = \sqrt{(x - x_c)^2 + (y - y_c)^2 + (z - z_c)^2} \tag{3.13}$$

where x, y, z denotes the coordinates of an arbitrary position. Then the surface of a spherical object with radius r is simply $\mathbf{Z}_{sphere} = r$.

To simulate the situation of gravity anomalies received at the top surface due to a spherical object of radius $100\ \mathrm{m}$ and density contrast $\Delta\rho$ of $1\,\mathrm{g/cm^3}$, we define a square prism modeling region with a total extent of of $10\,000\,\mathrm{m} \times 10\,000\,\mathrm{m} \times 1000\,\mathrm{m}$ including a padding zone extended in both x and y direction. We found a window radius approximately $2000\,\mathrm{m}$-$5000\,\mathrm{m}$,depending on different input configuration, yields reasonable simulation results while keeping sufficient resolution. We chose a window radius of $2000\,\mathrm{m}$ for the gravity simulation of this model. The synthetic receivers and the spherical object are all set at $5000\,\mathrm{m}$. The spherical object is set at different location of $x_c = 5000\,\mathrm{m}$, $z_c = 870\,\mathrm{m}$ and $x_c = 3000\,\mathrm{m}$, $z_c = 100\,\mathrm{m}$, as shown in Figure 3.7.

To make a fair comparison of the convolution scheme and kernel methods, the equivalent window size used in the convolution scheme is used to define the centered grid extent in the kernel method. The equivalent resolution is also applied. We fix the vertical resolution of 40 and only vary the xy resolution and placed 11 synthetic receivers evenly distributed between $x = 2300\,\mathrm{m}$,$z = 1000\,\mathrm{m}$ and $x = 7000\,\mathrm{m}$,$z = 1000\,\mathrm{m}$. The configuration of the model is is shown in Figure 3.8 and Figure 3.9.

The analytical solution of a sphere is given by

$$g_n = -10^8\,\frac{4}{3}\,\pi\,\frac{r^3\,\Delta\rho\,G\,z'_n}{\sqrt{(x'^2_n + y'^2_n + z'^2_n)^3}} \tag{3.14}$$

where x'_n, y'_n, z'_n are the relative position of the center of the sphere to the ith receiver, g_n is the vertical component of the gravitational vector in mGal, r is the radius, $\Delta\rho$ is the density contrast to the surrounding domain of unit $\mathrm{g/cm^3}$. The negative sign

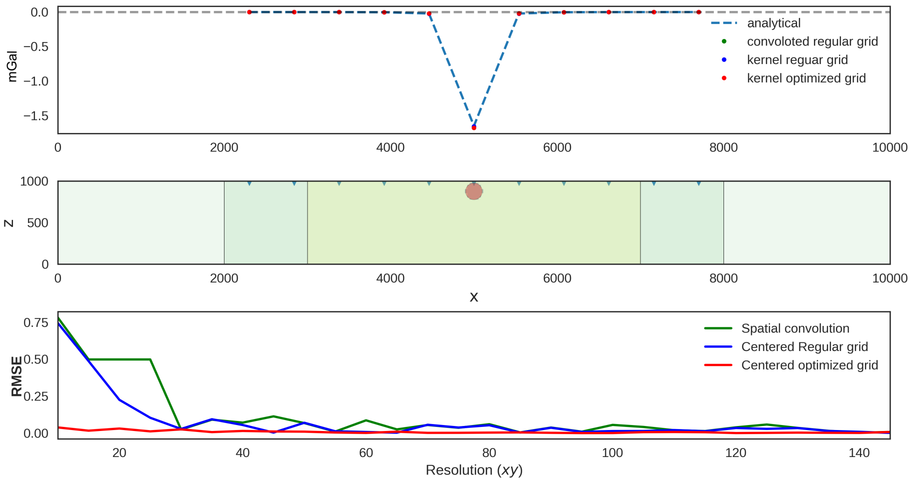


FIGURE 3.8: Simulation results of the model shown in Figure 7(a) Upper row: forward gravity simulated by three methods all with the window size equivalent to 150×150 resolution convolution scheme; Middle row: cross-section of the model illustrating the subsurface geometry; Lower row: RMSE to the analytical solution with different equivalent resolution.

is due to the chosen z direction.

The Root Mean Square Error (RMSE) is calculated between the analytical solution and the simulation results by the three different methods (spatial convolution scheme, kernel method with centered regular grid, and kernel method with centered optimal grid) along all receivers

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{n=1}^{N} (g_n - \hat{g}_n)^2}{N}} \tag{3.15}$$

g_n denotes the analytical solution at the nth receiver, $\hat{g}_n$ represents the simulated gravity at the nth receiver. N is the total number of receivers.

Results with a simultaneously increasing xy window resolution from 10 to 140 are demonstrated to show the performance difference and convergence. The positive density anomalies leads to a negative gravity anomalies at the surface due to the choice of positive z pointing upwards,

In both examples, all three methods show convergence or the tendency to converge to the analytical solution at a high resolution. In the case where the object is placed close to the top surface ($z_c = 870\,\mathrm{m}$), kernel method with optimized grid shows significantly smaller RMSE starting from low resolution. Kernel method with regular grid shows slightly smaller RMSE than that of convolution scheme, which demonstrates the error due to slicing in convolution scheme. The superior performance of the optimized grid is not surprising as the top part of each grid is refined. Accordingly, a less superior performance is expected in the case of the object appearing at the lower part of the model. However, because the sensitivity of the lower part of the model is orders of magnitude lower compared to the previous example, the RMSE difference

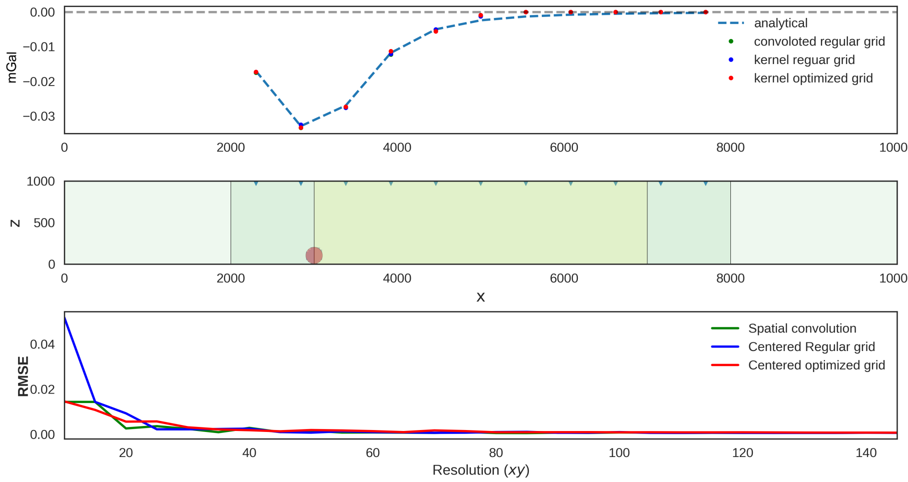
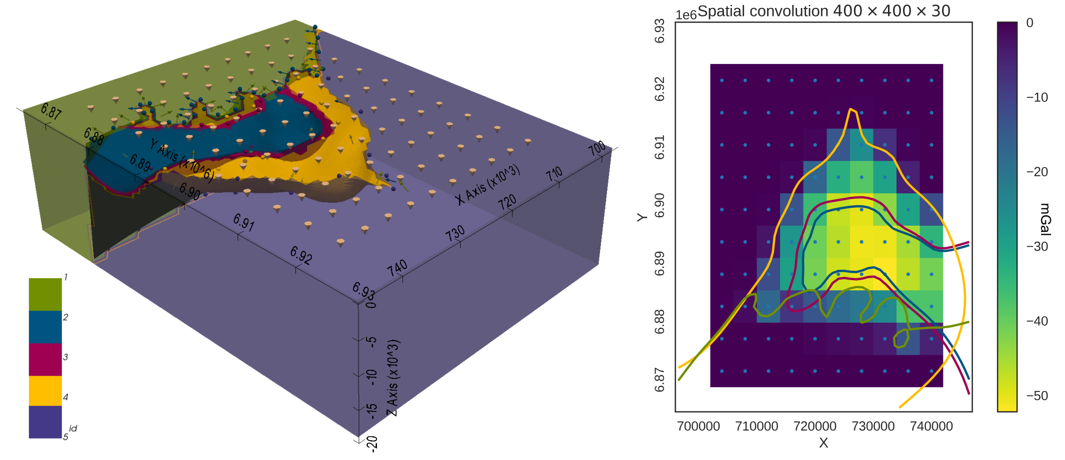


FIGURE 3.9: Simulation results of the model shown in Figure 7(b) Upper row: forward gravity simulated by three methods all with the window size equivalent to 150×150 resolution convolution scheme, dashed lines shows the linear interpolation of the analytical solution; Middle row: cross-section of the model illustrating the subsurface geometry; Lower row: RMSE to the analytical solution with different equivalent resolution.

is also less significant than in the previous scenario, all three methods show similar performance. These two examples demonstrate the superior performance of kernel methods with the kernel optimized grid to the most sensitive part of the model while preserving the same performance for the lower part of the model. The slight wobble at the beginning of the spatial convolution scheme shows again the error introduced by slicing when the grid is coarse.

3.3.2 Sandstone Greenstone Belt

In the second example, we apply our methods to the Sandstone Greenstone Belt (SSGB) in Western Australia. A comprehensive review of the regional setting and exploration was given by Davies et al. (2018). The SSGB is an Archaean greenstone belt located between the conjunction of two major north–south-trending Younami shear zone and Edale shear zone (Chen, 2005). The belt's stratigraphy consists of greenschist-facies metamorphism of mafic, ultramafic, and minor sedimentary sequences. The gold-rich deposits have received much interest in mineral exploration (Davies et al., 2020). Due to the presence of heavy minerals, the gravity contrast makes it a perfect case to test our methods. The previous study was done to model this region based on surface mapping, shallow drill holes, and a seismic survey by Wellmann et al. (2018). In this work, we adopt the model developed in (Wellmann et al., 2018) and focus on the forward gravity simulation. We reproduce the 3D geological model of the greenstone belt in GemPy (Figure 3.10) and simulate the gravity using the three different schemes, similar to the previous example.



(a) (b)

Figure 3.10: (a) 3D GemPy model of greenstone belt, lithology units id: 1: Early Granite; 2: Simple Mafic II; 3: Simple BIF; 4: Simple Mafic I; 5: Merchison (b) forward simulated gravity anomaly by spatial convolution scheme with a total resolution of $400 \times 400 \times 30$, with window resolution of 79×59. The value is negative due to positive z is defined pointing upwards. The white area represents area with no computation data and the approximate padding zone

The simplified lithology units and their corresponding densities are given as the following: Early Granite $2.61\,\mathrm{g/cm^3}$, Simple Mafic II $2.92\,\mathrm{g/cm^3}$, Simple BIF $3.1\,\mathrm{g/cm^3}$, Simple Mafic I $2.92\,\mathrm{g/cm^3}$, Murchison $2.61\,\mathrm{g/cm^3}$ (Figure 3.10). The model has an input consisting of 69 surface points and 40 orientation points. It has an East-West extent of $51\,\mathrm{km}$, a North-South extent of $87\,\mathrm{km}$ and a depth of $20\,\mathrm{km}$. The padding zone required for the convolution scheme is included in the modeling extent. A window radius of $5000\,\mathrm{m}\ \times 5000\,\mathrm{m}$ is used for the gravity simulation. 100 gravity receivers are evenly placed at the top surface as shown in Figure 3.10. We found $400 \times 400 \times 30$ is a high enough resolution for a model with this scale while taking reasonable computational resources, therefore we chose the forward gravity simulated using the spatial convolution scheme with this resolution as a benchmark.

To have a fair comparison, we use an equivalent window resolution for both spatial convolution schemes and kernel methods. The Residuals of simulated gravity from the benchmark high-resolution simulation are shown in Figure 3.11. When a low window resolution is applied (upper row of Figure 3.11), the slicing error has a significant impact on the convolution scheme, hence a large RMSE. Both kernel methods show closer simulated results to the benchmark, while the optimized grid shows a smaller overall RMSE. In the case of medium resolution example (lower row of Figure 3.11). Both kernel methods show comparable and superior results while the optimized grid resolves the region of complex surface structure (e.g. southeast corner) slightly better.

3.4 Discussion

We provide in this work a perspective to mesh a geological model depending on the positioning of the receivers for the gravity simulation in implicit geological modeling

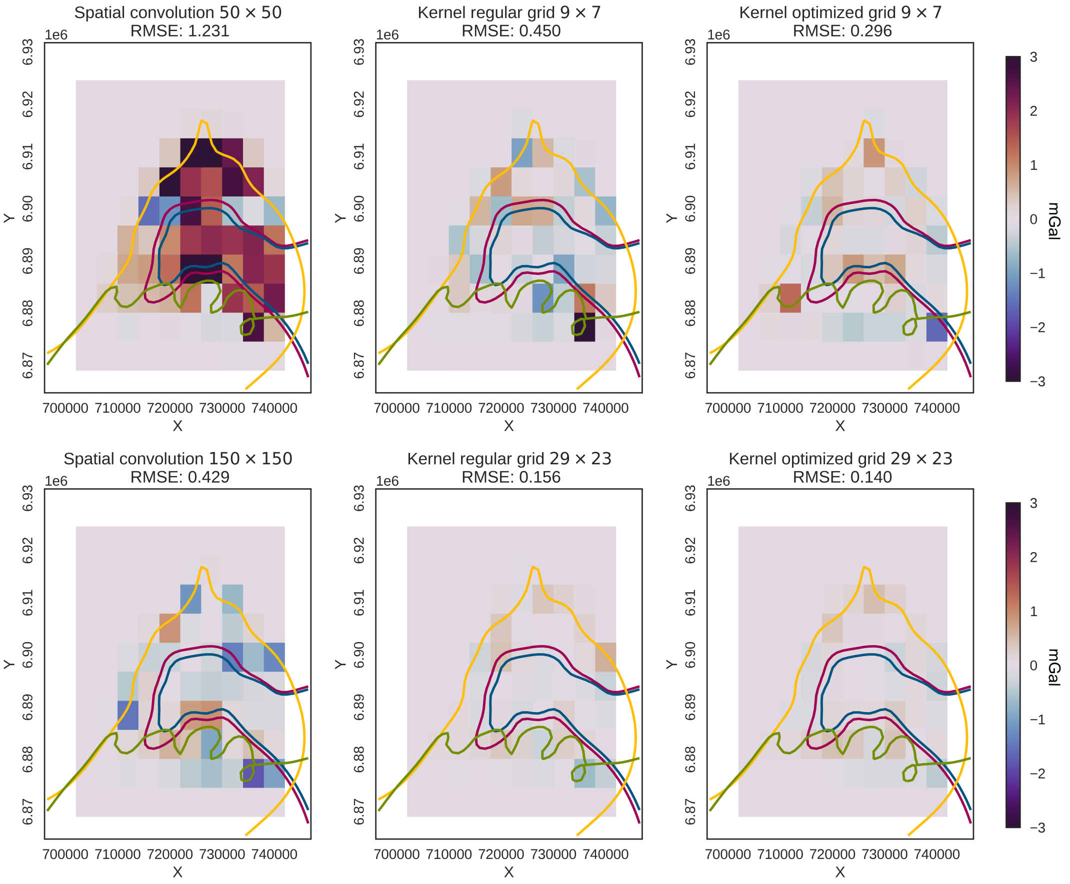


FIGURE 3.11: Residuals of simulated gravity from high resolution simulation with equivalent window resolution of 79×59. Upper row: equivalent low window resolution of 9×7. Lower row: equivalent medium window resolution of 29×23

and a subsequent refinement strategy based on the kernel methods. In this section, we will summarize the advantages and some open questions.

The kernel method is developed based on the fact that a property can be queried at an arbitrary location based on the interpolated scalar field in implicit geological modeling methods. This allows us to construct meshes individually respecting each receiver without an explicit surface construction. The closed-form solution of gravity is used to achieve a fast computation and no requirement for an external numerical solver. It also allows us to skip the step of explicit surface construction which saves computational time. We used the same grid for each kernel in this work so that it is easier to prepare the sensitivity matrix as a static input, however, each kernel does not have to be identical. One can design the kernel based on individual measurement receivers. For example, a higher resolution grid can be given to more precise receivers, and this can be easily done in a kernel method. In addition, kernel methods diminish the error arising from grid slicing as shown in the previous two examples. One could also potentially define the window radius based on the receiver precision. This study has constructed grids with flat top surfaces and the topography is ignored, which is appropriate for Bouguer anomaly with terrain correction. However, we believe the proposed method should also show similar results for topography effect.

The kernel method naturally provides a conformal grid that can be easily vectorized and therefore takes advantage of parallelized computation implemented in modern software packages. When the number of observations is large, the sequential computation to loop through the devices can be reduced to a single computation. The conventional spatial convolution scheme is implemented in this study in an iterative manner, which is adopted by most modeling softwares. Although this does not prevent it from vectorized computation, extra processes might be required to obtain a conformal grid. In cases where the number of observations is sparse on a large model, although the computation for gravity is not conducted at regions without data, the grid is initialized in those regions when using a spatial convolution scheme. The kernel method saves memory by not constructing meshes in those regions where data is unavailable, and also the padding zone is embedded in each kernel. As previously described, a core objective of chaining the modeling and geophysical simulation is enabling the use of derivative-informed Bayesian inference methods (e.g., HMC, SVGD, etc.). The introduced kernel method also provides a correspondingly simpler program structure with less conditional statements and sequential computation steps. This simpler structure improves the efficiency under the framework of AD in the context of probabilistic geological modeling (de la Varga et al., 2019). While the key values of interest are often the derivatives of posterior probability (or log posterior probability) with respect to the parameters of interests, this can often be reduced to the derivative of likelihood (Equation 3.2) when the prior has a known analytical form. This further reduced the problem to evaluating the derivative of simulation results (here is gravity) with respect to the parameters through AD. Therefore, in this work, we compare the computation time of evaluating the first-order derivative (jacobian) of the simulated

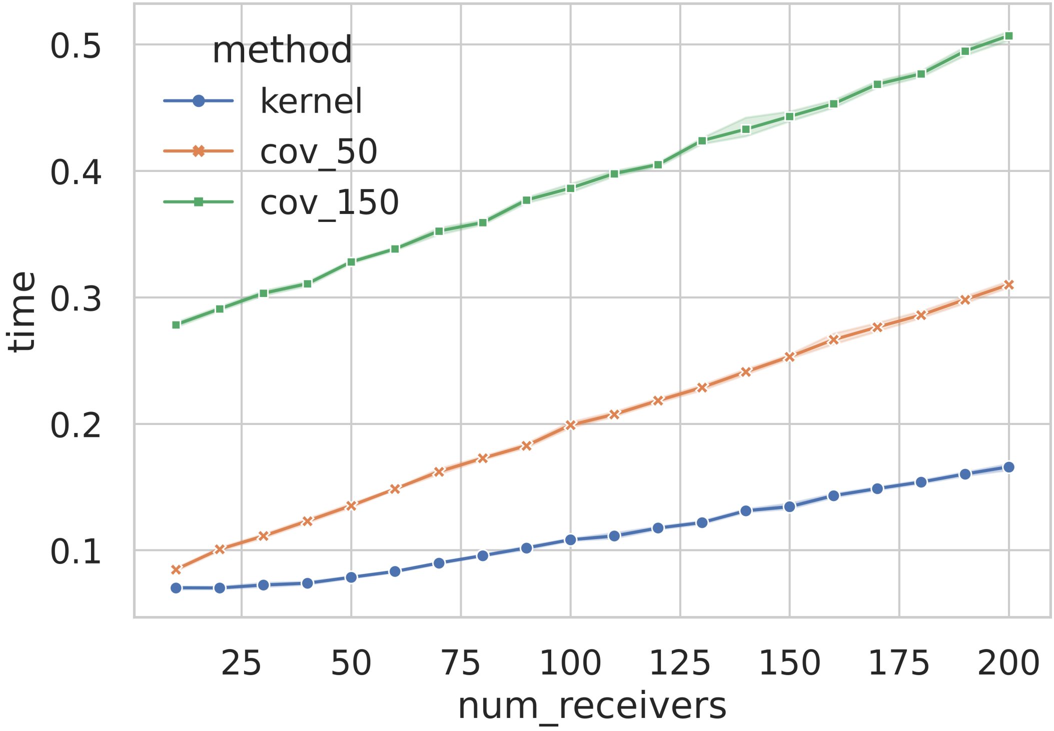


FIGURE 3.12: Computation time (in seconds) of evaluating jacobian of gravity over the input surface points(69×3 parameters) using backpropagation algorithm (AD) using kernel method and convolution scheme respecting to different number of devices on the Green stone model.

gravity at each receiver with respect to the input surface points position of the Greenstone model through a backpropagation algorithm, and the results are shown in Figure 3.12. The experiment is performed on a single Tesla P100 GPU with a different number of receivers. Although a growing matrices size due to the vectorization in the kernel method introduces some computational overhead and leads to a non-constant computational time, the 9×7 kernel is still twice faster than a 50×50 convolution scheme and four times faster than a 150×150 convolution scheme, while the kernel methods could have less error depending on the grid chosen (Figure 3.11). The vectorized computation compared to the loop structure in AD and the corresponding memory usage and compute speedup trade-off is still actively investigated by researchers (Bagby et al., 2018; Agarwal, 2019). We believe that the memory overhead can be addressed with the fast-evolving hardware or batching algorithm in the future.

The subsequent refinement strategy we suggested is on the basis of the centered grid of the kernel method. A similar idea of receiver-dependent refinement was adopted in the study of the gravitational terrain effect (Capponi et al., 2018), where the authors define an inner and outer regime based on the receiver location and applied different simulation methods and grid, and also the application of AEM by (Yang et al., 2014). The main difference of the refinement strategy in our work is that the refinement is based on the sensitivity T_z rather than simply two different regimes. The refinement strategy aiming to evenly relocating the limited resolution according to the sensitivity is based on the assumption that the geological interfaces are not known before interpolating the model and the underlying structure could vary due to the inference procedure. The choice of a fixed grid (which is independent of the interpreted geological model) allows reusing the same grid repeatedly in a many-query problem without embedding the mesh adaptation in the probabilistic framework. In this work, we chose a tensor mesh as a subsequent extension to the regular grid. However, choices of other types of mesh are possible, for example, an octree mesh. More advanced mesh adaptation methods can be applied while still keeping a rectangular mesh (Fuster et al., 2009; Fournier, 2019) and therefore apply the closed-form solution.

Although a more sophisticated refinement strategy will provide a more evenly distributed grid respecting T_z, we have shown in this study a more evenly distributed grid can lower the risk of aliasing effect to some extent for a geological model with unknown structure before interpolation, we want to emphasize that the final quality of the forward simulation still depends on the interpreted geometry. For example, in cases where subsurface structure with features appear only at the distant region to the receiver or coincidentally align poorly with the cell, a better-optimized grid does not guarantee a better result. Therefore extra effort might not gain more benefit in this sense. A refinement stratigraphy respecting the fuzzy prior knowledge of the geological model is a potential future study topic.

The numerical examples provided in this study are meant to demonstrate the general principle of the proposed methods, rather than an exhaustive proof of the validation. With different geometry, the refinement strategy performs differently according to the actual structure. We argue that if the subsurface geometry is known before interpretation, the best refinement strategy should be combined or guided by prior knowledge of the structure, but the resulting grid should also be applicable for the purpose of varying model in probabilistic modeling.

In this Chapter, we focused on the forward simulation and backward propagation, however, we can expect the same better performance on the inversion which is directly based on the results of many-query forward simulations.

3.5 Conclusion

We proposed in this study a kernel method for gravity computation in the implicit modeling framework to improve the conventional gravity forward method. The kernel method differs from the conventional methods by only meshing around the receiver position. This method provides higher simulation accuracy and simpler program complexity which is beneficial for an integration into probabilistic geological modelling frameworks. The kernel method potentially saves memory in scenarios where data is sparse. To further improve the accuracy of the simulation, we provide an optional subsequent refinement strategy which refines the tensor mesh in each individual kernel based on the sensitivity. This refinement allows better resolution to be assigned to more sensitive area under the assumption that the subsurface structure is not know prior to the interpolation. The numerical results show that the kernel methods together with the refinement strategy provides a superior performance compare to the conventional spatial convolution scheme and is more suitable for the gradient based probabilistic geological modelling framework.

Chapter 4

Trainable Geological Modeling

4.1 Introduction

Recent developments in probabilistic inversion workflows incorporate forward geophysics based on forward-modeled 3-D structural geology. Bayesian inference provides a systematic way to quantify the uncertainties in geological modeling. In the framework of Bayesian inference, Geophysical data is introduced in the modeling procedure as the likelihood function (see Section 2.4.2). The likelihood function acts as an additional constant to the prior knowledge of the model, which can further control the uncertainty range. Instead of finding the best-fit solution, Bayesian inference often seeks the full geometry of the high-probability region. Therefore, solving the uncertainty quantification problem in the Bayesian inference problem means a full exploration of the posterior probability space. A common technique called Markov Chain Monte Carlo (MCMC) is often adopted to efficiently draw chains of samples from the posterior space based on the probability relationships between the current status and the previous status in the chain. The simple but also widely used random-walk Metropolis-Hastings (RMH) algorithm is often used. However, as the parameter dimension increases, the convergency rate of RMH dramatically slows down, which will require drawing more samples (Cotter et al., 2013). Many algorithms have been proposed recently to accelerate MCMC. While some algorithms make simple modifications to the original algorithms without additional information (Green, 1995; Andrieu and Thoms, 2008; Cotter et al., 2013, e.g.,), other algorithms achieve better efficiency by using the derivative information of the log posterior space. The most popular methods using the first-order derivatives are the Metropolis-adjusted Langevin algorithm (MALA) (Roberts et al., 1996) and Hamiltonian Monte Carlo (HMC) (Betancourt, 2017; Beskos et al., 2013), which uses Langevin dynamics and Hamiltonian dynamics, respectively, to guide the proposed sample moving towards the high probability region which keeps the target density invariant. Recent advances in Bayesian inference methods introduced higher-order derivatives to accelerate the posterior exploration by capturing the posterior geometry (Qi and Minka, 2002; Petra et al., 2014; Isaac et al., 2015; Villa et al., 2021). The other branch of solving a Bayesian inference problem is by using the so-called variational inference methods (e.g., Rezende and Mohamed, 2015; Tran et al., 2015a; Liu and Wang, 2016; Detommaso et al., 2018, etc.). The variational inference methods approximate the target distribution by the transformation of a predefined simpler distribution. While different variational methods mainly differ from each other by choice of the predefined distribution, derivative information is generally required.

Although a handful of inference methods are available to solve the Bayesian problem, methods are limited to derivative-free algorithms in the application of geological modeling (e.g., Wellmann et al., 2018; Pollack et al., 2021). This is mainly due to the extra cost of deriving the derivatives using numerical methods. To solve this issue, de la Varga et al. (2019) introduced Automatic Differentiation (AD) to geological modeling and subsequent geophysical simulation in the open-source software GemPy. AD constructs a computational graph that keeps a record of the operations in the forward calculation and applies the chain rule to calculate the derivatives. Although the AD framework provides the possibility to efficiently calculate the derivative, that does not guarantee that the calculated derivatives from the geological model properly fit our purpose. This is mainly due to the discontinuities handled by level-set like methods in implicit modeling, which will be further discussed in detail in this Chapter.

Hence, in this study, we introduce the method which can make the geological model end-to-end trainable. The term ’trainable’ is adopted from the field of machine learning (ML). In recent advances in ML, the widely applied Artificial Neural Networks (ANN) technique seeks the underlying relationships between the input features and the output prediction criteria by constructing a sequence of interconnected artificial neurons. The neurons are connected by some parameters (weights, bias). The typical problem in ANN is optimizing the parameters to find the best-fit map according to the learning rule (Goodfellow et al., 2016). This optimization process is also referred to as the training process and often requires the derivative information of the learning criteria with respect to the parameters. Hence, a ’trainable’ ANN can be interpreted as its ability to be optimized by the derivative information (Shi et al., 2016; Frankle and Carbin, 2018; Li et al., 2018). Similarly, we are seeking a method to efficiently evaluate the derivatives of the modeling parameters, and we can use the calculated derivatives to guide optimization and inference. This leads to a smooth transition between the discontinuities. Therefore, we introduce a smooth function to replace the conventional level-set methods in the implicit modeling procedure, and we will discuss in detail how the proposed method can be used in dealing with different types of discontinuities in the co-kriging method. A relevant study by Scalzo et al. (2021) introduced a method where they attempt to solve the so-called aliasing effect by fitting an anti-aliasing function. The method seeks an interpolation between the values in the discretized mesh and aims to generate faithful simulation results without an aliasing effect. While their method produces good simulation results, the method requires the calculation of the norm of the surface at the discretized mesh. Our approach differs from their work by directly interacting with the scalar function. Therefore there is no need to calculate the norm independently. Although with different initiatives, both methods are aiming for smoothness at the discretization.

Finally, we propose a method to visualize the high-dimensional posterior (or likelihood) landscape in an intuitive and concise way. We adopt the method of visualizing the loss landscape in ANN (Li et al., 2018). The proposed method can be used to qualitatively evaluate the ability of 'trainable' in geological modeling, and intuitive visualization of the effect of the smooth function on the posterior (or likelihood)

4.2 Automatic Differentiation

AD is the fundamental to the trainable geological modeling, therefore we will have a brief review of the AD technique.

4.2.1 Automatic Differentiation principle

Derivative information is often required in many traditional learning algorithms. Generally speaking, there are four categories of methods to evaluate the derivatives in computational science: (1) manually derive and code the derivatives; (2) numerical differentiation using finite-difference approximation; (3) symbolic differentiation using expression manipulation; (4) automatic differentiation (Baydin et al., 2018). Manually deriving the derivatives is time-consuming and is not flexible to different model configurations for our application. Numerical differentiation is subject to precisions limit due to round-off and truncation errors (Jerrell, 1997). Symbolic differentiation addresses the issues in both manual and numerical methods but often results in complex expressions, which will lead to the so-called problem of 'Expression Swell' (Corliss, 1988) and requires a closed-form expression which is the same as the manual method. AD compensates this by performing a partly numerical and partly symbolical style technique (Griewank, 2003). AD has been widely adopted in several fields of applications due to its accuracy, efficiency, and flexibility. Many general-purpose AD software package are also developed in recent years, for example: Theano (Bastien et al., 2012), TensorFlow (Abadi et al., 2015), autograd (Maclaurin et al., 2015) and more recent such as PyTorch (Paszke et al., 2017) and JAX (Bradbury et al., 2018).

AD performs an operator overloading of the original computer program and constructs a computational graph that will keep track of all the intermediate operations from the parameters of interest to the output value. Then, the derivatives of the output with respect to the parameters of interest can be evaluated by applying the chain rules of differential calculus. Consider an arbitrary function $\mathbf{y} = F(\mathbf{x})$, where F is the mapping function which maps the input parameters $\mathbf{x} = x_1, x_2, ...x_i, (\mathbf{x} \in R^{D_0})$ to the cost function $\mathbf{y} = y_1, y_2, ..., y_j, (\mathbf{y} \in R^{D_1})$. The first-order partial derivative, also known as the Jacobian matrix, is given as follows:

$$(\mathcal{J}_F)^i_j (x) = \frac{\partial (F)^i}{\partial x^j}(x) \tag{4.1}$$

by constructing the composite function $F = F_N \circ F_{N-1} \circ \cdots \circ F_2 \circ F_1$ and ignoring i, the Jacobian matrix can be represented by iteratively applying the chain rule:

$$\mathcal{J}_F(x) = \frac{\partial (F)}{\partial x^j}(x) = \frac{\partial (F_N)}{\partial (F_{N-1})} \cdots \frac{\partial (F_2)}{\partial (F_1)} \frac{\partial (F_1)}{\partial x}(x) \tag{4.2}$$

AD records the derivative of each fundamental operator in a program and builds a computational graph to record the dependency of each operator. The derivative of the output with respect to the input parameter of interest is evaluated by a forward or a backward propagation using the chain rule based on the computational graph. Forward mode and backward mode both have their advantages and disadvantages. By combining these two methods, we get an efficient method for second-order derivative evaluation. The approach is summarized in the following.

To evaluate the gradient in the forward mode AD, a tangent vector v is selected at the evaluation point x,

$$\mathcal{J}_F(x) \cdot v = \left(\frac{\partial (F_N)}{\partial (F_{N-1})} \cdots \left(\frac{\partial (F_2)}{\partial (F_1)} \left(\frac{\partial (F_1)}{\partial x}(x) \cdot v \right) \right) \right) \tag{4.3}$$

For example, $\frac{\partial (F)^i}{\partial (x_1)}$ can be evaluated by a tangent vector $v = (1, 0, \ldots, 0), v \in R^i$. Thus, the forward mode AD provides the directional derivative.

In contrast, backward mode AD evaluation is based on a fixed dependent variable, and the quantity of interest is the adjoint

$$\mathcal{J}_F^i(x) = \left(\left(\left(\frac{\partial (F_N)}{\partial (F_{N-1})} \right) \cdots \frac{\partial (F_2)}{\partial (F_1)} \right) \frac{\partial (F_1)}{\partial x}(x) \right) \tag{4.4}$$

Evidently, for a problem $\mathbb{R}^{D_I} \to \mathbb{R}^{D_O}$, where D_I denotes the dimension of the input, and D_o denotes the dimension of the output, if $D_I \gg D_O$, backward propagation is more efficient, which matches the cases of many machine learning problems and the Bayesian inference in our study, where the output is a single cost function, so $D_O = 1$. In contrast, forward propagation is more efficient when $D_I \ll D_O$, requiring less memory.

AD has been widely adopted as an efficient and accurate way to calculate derivatives in many different applications in geophysics, for example, seismic wave (Sambridge et al., 2007) , electromagnetics (Enciu et al., 2009), groundwater flow (Rath et al., 2006), gravity (Abad and Lacruz, 2013) etc. Only recently, de la Varga et al. (2019) introduced AD to geological modeling in the software package GemPy. In this study, we seek improvements following the workflow of GemPy.

4.2.2 Computational Graph

To fully adapt to AD, we implement the program in TensorFlow and TensorFlow Probability for the probabilistic modeling purpose. The forward model chains the implicit geological modeling and geophysical simulation in a computational graph. The implemented model is best visualized in the form of a network graph, analogous to typical ANN graphs, and shown in Figure 4.1. The graph mainly consists of three blocks: geological modeling, geophysical simulation, and Bayesian inference. The inputs to the graph are also the parameters of interest describing the parametric representation of the structural geological model. Each of the input parameters can be treated as a probabilistic variable or fixed as a static tensor, depending on the problem formulation. The inputs are passed to the calculation of the aforementioned implicit modeling algorithm (Sec. 2.3). The main elements in the modeling graph are the kernel matrix blocks in Equation 2.6 (or covariance matrix in geostatistics nomenclature), the kernel hyperparameters (or kriging parameters in geostatistics), and the mapping values for the scalar field values (e.g., rock unit IDs or density). These elements are calculated through a sequence of deterministic computations implemented using TensorFlow operations and represented by the black lines. The elements can be seen as the counterpart to nodes in an ANN. The grid node represents different discretization methods applied to visualize the surfaces or to obtain the tesselation for the geophysical simulation on the implicit scalar function and is here treated as a constant parameter in the graph. In this gravity example, The forward gravity is then evaluated based on the properties and scalar values based on the summation of the analytical solution of discretized elements Nagy (1966). In analogy to the training data in an ANN, the observation data from the field is another constant value to the graph. At this stage, different loss functions can be constructed depending on the problem. In a deterministic inversion case, one can directly seek the optimal solution by minimizing the differences between the simulation result and data, for example, by minimizing the 2-norm

$$\min \left\| \bar{\boldsymbol{d}} - \boldsymbol{d}_{\text{obs}} \right\|^2 . \tag{4.5}$$

In a probabilistic framework, the inputs are implemented using TensorFlow Probability distributions. The loss function can be constructed in a Bayesian manner either to optimize the maximum likelihood estimate or, in combination with the prior distributions of the parameters, to explore the full posterior distribution $p_{\text{post}}(\boldsymbol{m} \mid \boldsymbol{d}_{\text{obs}})$ where $\boldsymbol{m}$ is the parameter set and can be any subset of the parameters $\boldsymbol{m} \in \{I, G, F, \cdots, \rho\}$.

By constructing the computational graph in the TensorFlow framework, the derivatives of the loss function with respect to the parameters of interest can be traced through the computational graph by repeated application of the chain rules. The implementation, therefore, opens-up the path to an application of derivative-based sampling methods such as HMC for joint geological modeling and geophysical inversion in a probabilistic machine learning framework.

4.3 Trainable Geological Models

By adopting AD in geological modeling, one can calculate derivatives of the output with respect to the parameters of interest efficiently through the computational graph with minor modifications to the forward program. However, AD does not guarantee the calculated derivatives are feasible for training and inference purposes. This is mainly due to the level-set during the discretization process.

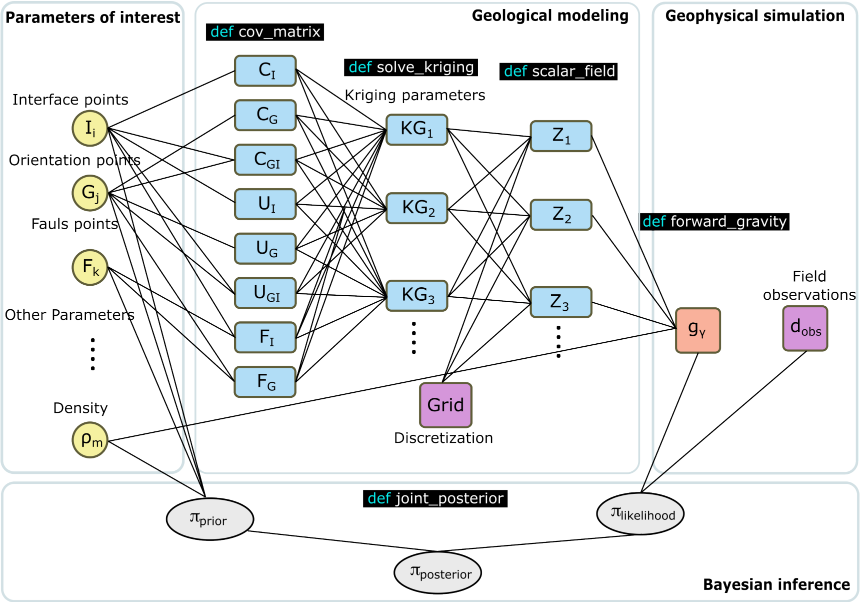


FIGURE 4.1: Illustration of the TensorFlow graph structure. Characteristic parameter are represented in blue nodes. Grey and yellow nodes represents the inference and input parameter implemented in TFP. Purple nodes denotes constant input to the model.

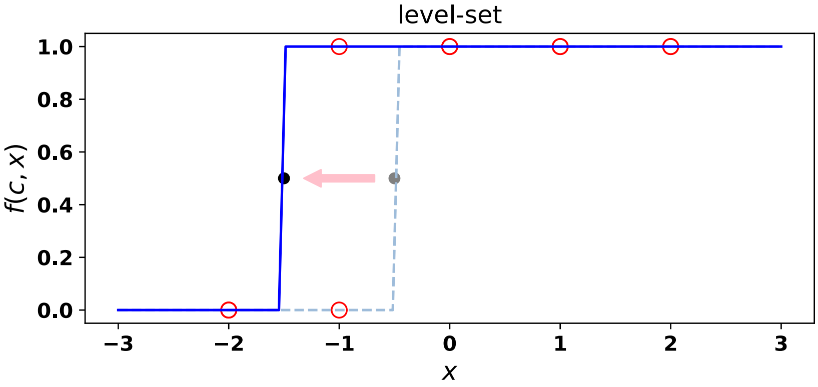
4.3.1 Vanishing Gradient

As previously discussed in Section 3.2.1, the forward modeling consists of two chaining maps: $\boldsymbol{f} := \iota \circ q$. The first map ι mapping the parameter $\boldsymbol{m} \in \Theta$ into a discrete volumetric representation $\boldsymbol{\rho} \in \mathcal{G}$. This mapping ι can be further divided into two parts as $\iota := \iota_1 \circ \iota_2$, where the first part ι_1 maps the parameters $\boldsymbol{m}$ into a function Z, and the second part ι_2 maps the discontinuous function into a discrete representation $\boldsymbol{\rho}$. The first map ι_1 is the co-kriging, where all the calculation is continuous. However, the second map often involves level-set-style methods, which utilizes a heavy-side function and leads to a discontinuous and non-differentiable function. To better illustrate the reason for the vanishing gradient in the level-set method, we demonstrate a simplified 1D dummy problem. We construct a dummy function with one variable c and define the minimization problem in the following form:

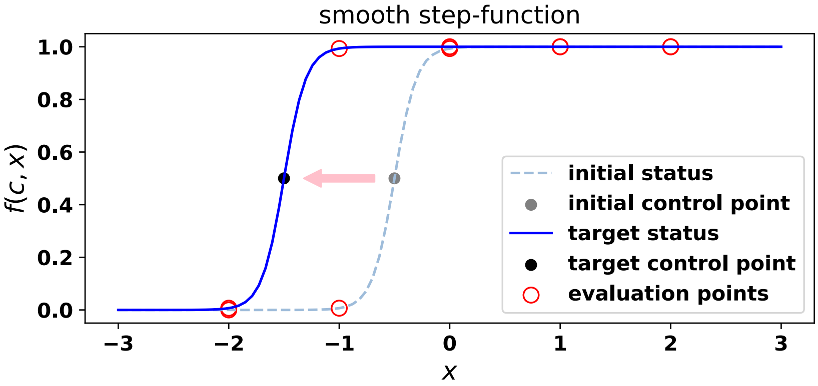
$$\text{Min } L = \| \sum_{x \in \boldsymbol{x}_s} f(c,x) - S_{\text{ target }} \| \tag{4.6}$$

where $\boldsymbol{x}_s$ is a fixed set of evaluation positions, $S_{\text{ target }}$ is an arbitrary target value we are trying to fit to the sum of function f evaluated at $\boldsymbol{x}_s$.The aim of the problem is to find the optimized control point c, which minimizes the loss L. Here the evaluation points are analogous to the property matrix in a geological model, and the controlling point c is analogous to the surfaces point or orientation points. When the function f is a heavy-side function and has the following form:

$$f(c,x) = \begin{cases} 0 & \text{if } x \leq c \\ 1 & \text{if } x > c \end{cases} \tag{4.7}$$



(a) Simple function with level-set method



(b) Simple function with step-function

FIGURE 4.2: Illustration of vanishing gradient with a simple function with level-set method and with smooth step-function.

If we take $\boldsymbol{x}_s = \{-2, -1, 0, 1, 2\}$, the controlling point with initial value of $c = -0.5$ and target $S_{\text{target}} = 4$, we can illustrate the problem in Figure 4.2a. It is obvious that one cannot find a smooth transition from the initial position to the target position by using gradient-informed optimization methods such as gradient descent because the gradient $\frac{\partial L}{\partial c}$ does not exist.

To solve this problem, one can replace this heavy-side function with a smooth step-function, similar to the activation function in a neural network (Sharma et al., 2017). Here we chose a sigmoid function that extends to infinity:

$$f(c, x) = \frac{1}{1 + e^{l(-x+c)}} \tag{4.8}$$

where l is the slope of the sigmoid function. We illustrate the smoothed function with $l = 10$ in Figure 4.2b. The step-function provides a smooth gradient $\frac{\partial L}{\partial c}$. Hence one could solve the above-described problem by using off-the-shelf gradient-informed optimization techniques.

4.3.2 Vanishing Gradient in implicit geological modeling

There are mainly three types of discontinuities affected in the current implementation of implicit modeling using the co-kriging method, namely layer interfaces, unconformities, and faults. These discontinuities interpreted by the heavy-side function directly cause the vanishing derivatives (here, we focus on the gradient). Therefore, in this section, I will first explain the reason for the vanishing gradient in each of the discontinuity cases and demonstrate the methods to keep the gradient using a smooth the step-function.

Layering model

In the simplest case of a layer cake model without unconformities and faults, the calculation involves only one single series of co-kriging calculation. Following the notation in Section 2.3, where the artificial auxiliary scalar field is denoted as Z, and scalar value at any given location $\boldsymbol{x} = (x, y, z)$ in Cartesian coordinates as $Z(\boldsymbol{x})$. We further denote the lithological layers from younger to older by using $i = 1, 2, \ldots, \mathbb{L}$ in an ascending order, where $\mathbb{L}$ is the total number of layers. An identical rock property value ρ_i is assigned to the corresponding ith layer. Then we denote the scalar value at the interface between the ith layer and the $(i+1)$th layer as Z_i. We can simply express the properties ρ at location $\boldsymbol{x}$ by using the level-set method in the implicit geological modeling as follows:

$$\rho(\boldsymbol{x}) = \begin{cases} \rho_1, & \text{if } Z(\boldsymbol{x}) \geqslant Z_1 \\ \rho_i, & \text{if } Z_{i-1} > Z(\boldsymbol{x}) \geqslant Z_i, \forall i = 2, 3, ..., \mathbb{L}-1 \\ \rho_{\mathbb{L}}, & \text{if } Z_{\mathbb{L}-1} > Z(\boldsymbol{x}) \end{cases} \tag{4.9}$$

An example of the resulting discrete volumetric representation is shown in Figure 4.3.

Unconformities model

An unconformity can be implemented by stacking scalar fields from different depositional series in the co-kriging modeling method (de la Varga et al., 2019). Considering two contacting series with the younger rock series I intercepting, the older rock series II by the erosional surface at the bottom of series I. The scalar field of the two series Z_I and Z_{II} can be interpolated individually with the input geological data at each series following the method described in the layering model. The combination of the two series is done by interpreting a binary mask from the bottom layer of series I. The binary mask is obtained by using the same level-set method to separate only the volume which is above or below the last surface $\mathbb{L}_I$ in series I. The implemented level-set method for the discretization of unconformities can be summarized as:

$$\rho(\boldsymbol{x}) = \begin{cases} \rho_I(\boldsymbol{x}), & \text{if } Z_I(\boldsymbol{x}) \geqslant Z_{I\mathbb{L}-1} \\ \rho_{II}(\boldsymbol{x}), & \text{if } Z_{I\mathbb{L}-1} > Z_I(\boldsymbol{x}) \end{cases} \tag{4.10}$$

where ρ_I and ρ_{II} are individual functions ruling the discretization in its own domain following Equation 4.9. A more complex sequence of unconformities can be implemented by repeating Equation 4.10. An example of an unconformity using this method is shown in Figure 4.4.

Fault model

Currently, only infinite faults are implemented. The infinite faults can be implemented as an extension to the co-kriging framework (Equation 2.5) by adding a drifting term to the kriging system (Marechal, 1984; de la Varga et al., 2019), shown as follows:

$$\mathbb{C}_{d\times d} = \begin{bmatrix} \mathbf{C} & \mathbf{U}^t & \mathbf{F}^t \\ \mathbf{U} & \mathbf{0} & \mathbf{0} \\ \mathbf{F} & \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{4.11}$$

$$\mathbb{K}_{d\times 1} = \begin{bmatrix} \mathbf{k_a} \\ \mathbf{k_b} \\ \mathbf{k_c} \\ \mathbf{k_d} \\ \mathbf{k_f} \end{bmatrix} \tag{4.12}$$

$$\mathbb{G}_{d\times 1} = \begin{bmatrix} G_i^x \\ G_i^y \\ G_i^z \\ \mathbf{0} \end{bmatrix} \tag{4.13}$$

$$\mathbb{C}\mathbb{K} = \mathbb{G} \tag{4.14}$$

where $d = n_c \times n_u \times n_f$ presents the dimension of the matrix, n_c is the length of the covariance matrix in Equation 2.6, n_u is 9 for a second order universal drifting (Equation 2.11), n_f is the number of series of faults, $\mathbf{F} = [F_Z, F_G]$ is the fault matrix, F_Z is the fault drifting function evaluated at the surface points locations and F_G is the derivative of the drifting function evaluated at the orientation points; $\mathbf{k_f}$ is additional unknown kriging parameter to be solved. The drifting function itself can be computed by using the same co-kriging method with the fault interface points and orientation points in the 'fault series' prior to the interpolation in Equation 4.14. A fault matrix is obtained at the grid locations from the drifting function. The same level-set method is applied to segment locations $\boldsymbol{x}$ from the foot-wall and hanging-wall depending on the fault compartment. A unique constant value is applied to the locations at each side of the fault based on the segmentation. The impact of the level-set method appears in both the two series of the scalar field. An example of this fault construction workflow is illustrated in Figure 4.5.

4.3.3 Method: Smooth Step-Function

To keep the gradient in the geological modeling at the discretization procedure, we introduce a smooth step-function. The introduced step-function can be applied to all three types of discontinuities described above.

Layering model

We construct the step-function with the sigmoid function described above. For the interface between two lithologies, the smooth step-function is constructed by connecting two sigmoid functions which takes the general form:

$$\rho(\boldsymbol{x}) = \frac{\rho_i}{1 - e^{l(Z(\boldsymbol{x}) - Z_i)}} + \frac{\rho_{i+1}}{1 - e^{-l(Z(\boldsymbol{x}) - Z_i)}} \tag{4.15}$$

Where l is the slope of the step-function, ρ_i and ρ_{i+1} are the property of the contacting layers i and $i+1$. A sequence of layering can be done by repeating this formula. The step-function naturally interacts in the orthogonal directions to the interfaces. Hence the step-function is also tangent to the norm of the interfaces and requires no additional approximations and calculations. The effect of the smooth step-function to the properties matrix compared to level-set method is demonstrated in a simple 3D anticline model and shown in Figure 4.3.

Unconformities model

The step-function can be easily extended to unconformity models. By adopting the step-function, the binary masking between the sequential series is replaced by a mask with values smoothly transferred from one status to another. The Equation 4.10 is then replaced by the form:

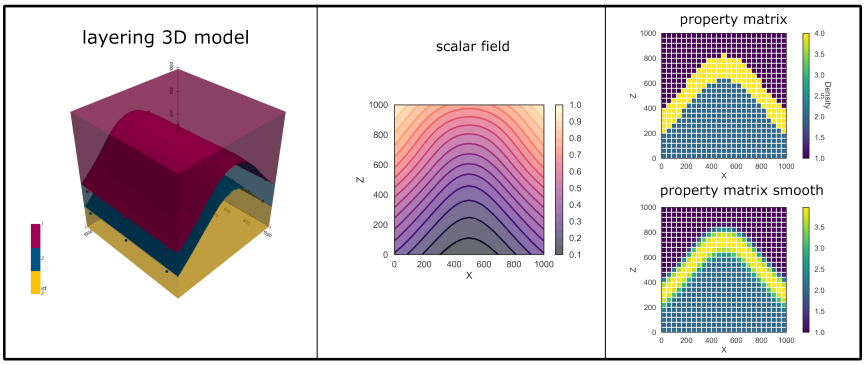


FIGURE 4.3: Workflow of 3D layering model construction.

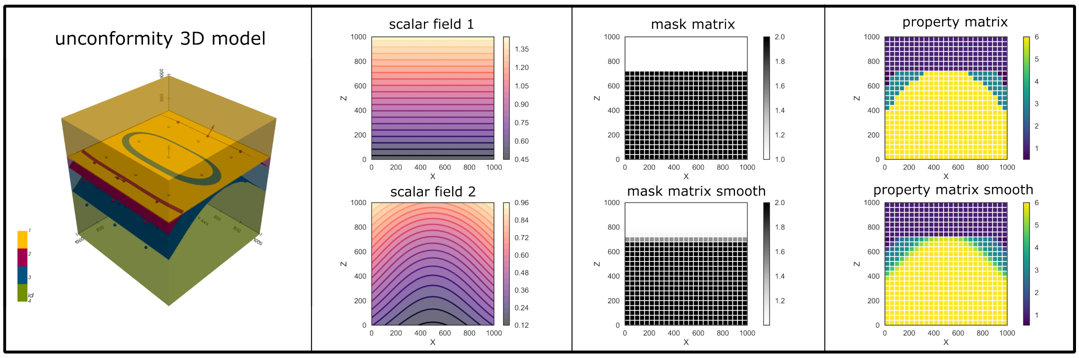


FIGURE 4.4: Workflow of 3D unconformities model construction.

$$\rho(\boldsymbol{x}) = \frac{\rho_I}{1 - e^{l(Z(\boldsymbol{x}) - Z_i)}} + \frac{\rho_{II}}{1 - e^{-l(Z(\boldsymbol{x}) - Z_i)}} \tag{4.16}$$

A 3D example of an unconformity between two series are demonstrated to shown the different property blocks using different methods in Figure 4.4.

Fault model

As discussed previously, a fault in the implicit modeling method using the kriging algorithm is implemented by an extra drifting function. The introduced step-function takes action in both series of kriging interpolation. First, the fault matrix is extracted at the first series of interpolation to distinguish the evaluation points on the footwall and hanging wall. Instead of a binary fault matrix obtained using the level-set method, a fault matrix with a smooth transition from the footwall to the hanging wall is obtained using the smooth step-function (example in Figure 4.5). Then the drifting term obtained from the first interpolation is integrated into the second interpolation. The resulting second scalar field has a curved shape which is similar to the 'dragging folds' and 'clay smearing' of a fault in a realistic scenario (Grasemann et al., 2005). The properties matrix is smoothed at both the faults and interfaces. By using the step-function, the gradient of the property blocks with respect to both the fault and surface input data are conserved.

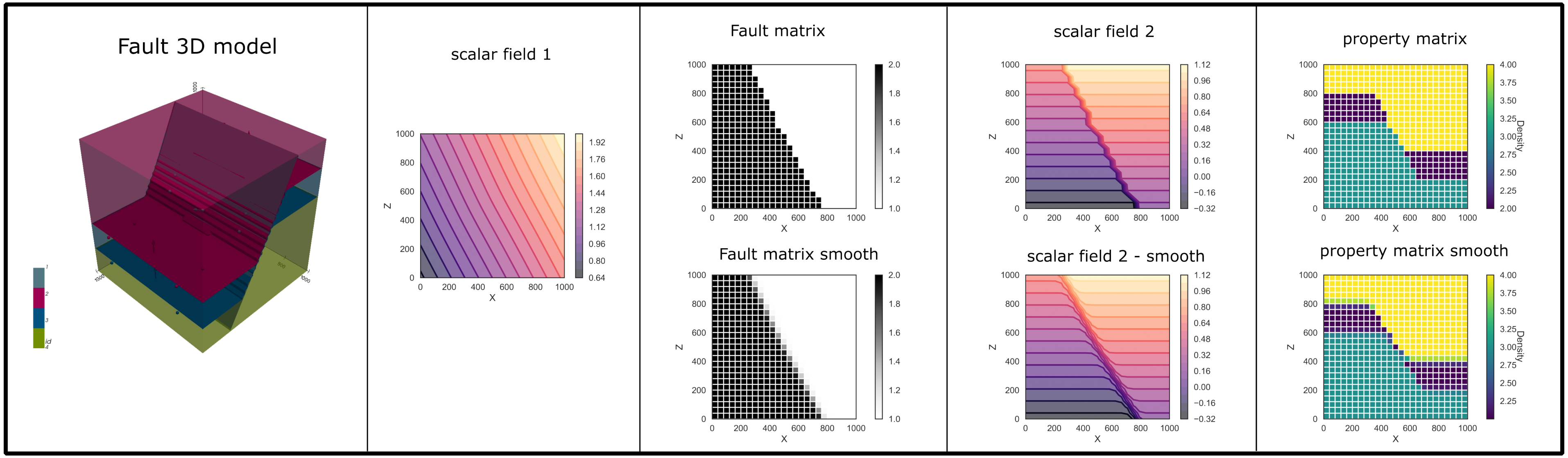


FIGURE 4.5: Workflow of 3D fault model construction. First, the fault points (surface points and orientation points) are interpolated in the first series of scalar field 1. A fault relationship matrix is constructed based on the interpolated fault scalar field 1. The fault matrix is added to the co-kriging equation as an additional drifting term to interpolate scalar field 2. The final property matrix is obtained based on scalar field 2. Upper and Lower rows demonstrates the effect of heavy-side function and step-function to the model construction, respectively. Both the workflow construct the first scalar field identically. The scalar field 2 using the level-set method is plotted using a contour plot, therefore the sharp discontinuities are connected due to the same scalar value on two sides of the fault with an off-set. The apparent perturbation is due to the limited resolution used to generate the plot.

4.3.4 Effect on Gravity Forward Simulation

The introduced smooth step-function generates geological models which are end-to-end trainable from the gravity simulation to the input parameters. However, the resulting property block is also smoothed with the artificially introduced step-function and therefore impacts the forward gravity simulation results. Hence in this section, we will discuss the impact of the step-function on the simulation results.

We use the same anticline model demonstrated previously in Figure 4.3. The model has a simple anticline geometry with three layers. A 5×5 grid of receivers is deployed at the top surface of the model. The densities are set for demonstration purpose as $1\,\mathrm{g/cm^3}$, $4\,\mathrm{g/cm^3}$ and $2\,\mathrm{g/cm^3}$ from top to bottom. We simulated the gravity at a high resolution with the cell size of $1.43\,\mathrm{m}\ \times 1.43\,\mathrm{m}\ \times 1.43\,\mathrm{m}$ using the method without smoothing as a benchmark. The model contains 36 surface points in total and is loosely constrained by two orientation points. The 3D model is demonstrated in an extended extent to illustrate the slightly curved surface at a distant area along the y axis without data constraints (Figure 4.6). This curved surface explains the gravity drop in the simulation results in an apparent 2.5D model. The gravity is calculated using the kernel method with a regular grid (as described in Chapter 3).

We compare the forward simulation result using both the level-set method with heavy-side function and the smooth step-function simulated in a lower resolution model with the cell size of $5\,\mathrm{m}\ \times 5\,\mathrm{m}\ \times 5\,\mathrm{m}$. While a more accurate forward simulation is not our primary objective, the simulated gravity with the smooth model gives a lower RMSE compared to the simulated result by using the level-set method. This is mainly due to the fact that the voxels which are at the interfaces are classified based on the value evaluated at the center of the voxel, and the interface is poorly interpreted due to the limited resolution. This is also known as the aliasing problem (Scalzo et al., 2021). While in the smoothed matrix, the properties of the voxels close to the interface will take an intermediate value and, therefore, could be more faithful to the simulation for a potential field problem, for example, gravity and magnetic. This result is also consistent with the result given by the anti-aliasing method used by Scalzo et al. (2021).

4.3.5 Choice of the Step Function

In the above example, the simulation result shows better accuracy when using step-function. The choice of the slope l has a significant impact on the simulation and trainability of the model. It is obvious that when l is chosen to be unreasonably large, which means the slope is very steep, the step-function basically degenerates into a heavy-side function and loses the trainability. Although the sigmoid function extends to infinity, the gradient value can not reach infinity in practice. Using the previous 1D example in Figure 4.2. The gradient is $\frac{\partial L}{\partial c} = \sum_{x \in xs} f(c, x)$. When the evaluation point is at a large distance to the control point (i.e., $|x_i - c| >> 0$), the gradient $\frac{\partial x_i}{\partial c}$ might have a numerical value of NaN. This causes problems in some AD software (e.g., Theano) that a single NaN will turn the whole result into an error, although these gradients are trivial and have no impact on the learning. A solution to keep the stability in the previous AD framework is to add an artificial activation function $\tanh$ at the tail of the step-function (de la Varga et al., 2019). In the modern AD frameworks (e.g., TensorFlow, PyTorch, etc.), this is solved by simply assigning NaN values to zero.

In contrast, when the slope l is extremely small, the model has a high trainability, but the property matrix will be smoothed out, and the simulation result is no longer faithful.

Therefore, we propose a subsequent optimization method to have a faithful simulation result and keep the trainability at the same time. We construct this in another optimization problem, where we generate data from a high-resolution model as the target simulation result we would like to achieve. We define the upper bound of the slope l_{max} taking the following form:

$$l_{max} = \lambda \cdot 2/d_{\text{diag}} \cdot \text{rf} \tag{4.17}$$

where rf is the scaling factor in GemPy to scale the input extent to unit scale for computation and d_{diag} is the diagonal distance of a rectangular prism cell. We found a scale value $\lambda = 1.5$ is sufficient to keep the gradient when dealing with most problems using a double float data type. The slope l can be then defined using another sigmoid function with variable δ_l as:

$$l = l_{max} \cdot \sigma(\delta_l) \tag{4.18}$$

Benefitions from the fact that the whole model is constructed in a computational graph (Section 4.2.2), we can further consider the variable δ_l as a TensorFlow variable and use AD to calculate the gradient of the misfit with respect to the variable. Here we use the gradient descent method to update δ, and the result is shown in Figure 4.7. A slight improvement of the forward simulation is gained by the optimization.

4.4 Posterior Visualization - a dimension reduction method

4.4.1 Morphology of Posterior Space

Similar to the loss function in an ANN discussed above, the posterior probability function could have a complex geometry in high-dimensional space. The qualitative characterization of the posterior landscape and the impact of the step-function on the posterior is helpful to comprehend the inference problem we are solving. While the prior is often chosen to be a well-defined distribution (e.g., normal distribution, uniform distribution), the effect of the prior on the posterior is known to be smooth. The smoothness of posterior landscape is mainly controlled by the likelihood function. Hence, in the following description, we will only use the likelihood for simplicity

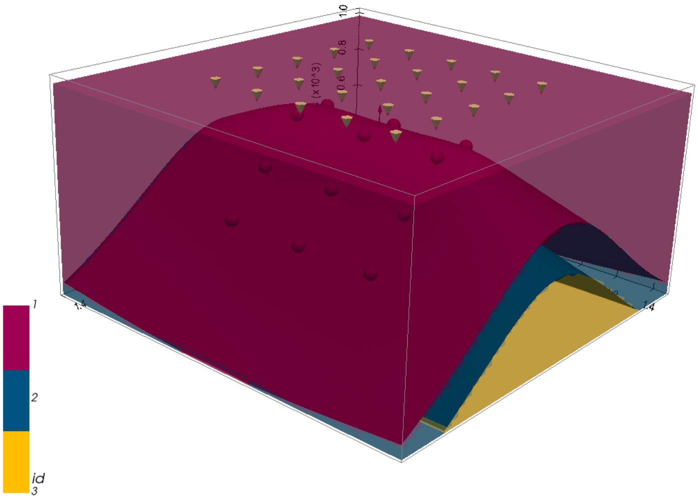


Figure 4.6: 3D plot of the anticline model

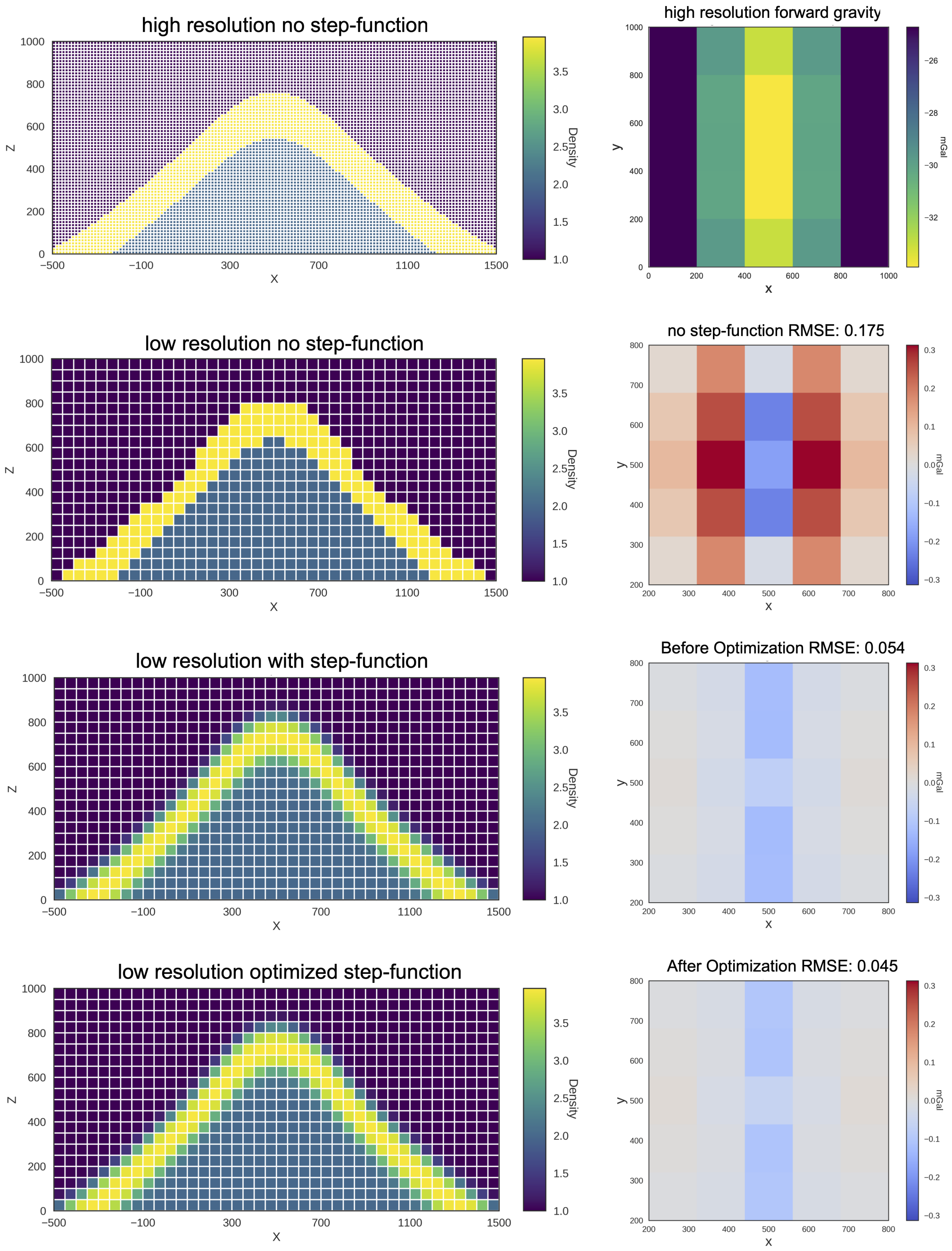
and to emphasize the impact of the step-function, but the method applies equally to posterior functions.

The two common approaches to represent the posterior distribution are the corner plot and the cross-section plot. A corner plot provides complete pictures of the posterior space by plotting the marginal distributions of each pair of parameters. It is suitable for plotting the empirical distribution of samples generated from the distribution. However, when it comes to plotting the exhaustive posterior distribution with grid-based analytical solutions, plotting conner plots becomes expensive. For distribution with $\mathcal{P}$ numbers of parameters, a corner plot will contain $(\mathcal{P}-1) \times (\mathcal{P}-1)$ number of subplots. If each plot has a grid dimension of $\mathcal{K} \times \mathcal{K}$, the total number of evaluations will be $\mathcal{K}^{(}\mathcal{P}-1)$. This exponentially growing computational cost is also the reason for the use of sampling methods like the Monte Carlo method.

As an alternative, in the lower dimension, cross-section plots are often used (Scalzo et al., 2021). Unlike the corner plot evaluates the posterior exhaustively, the cross-section plot fixes the other dimension (often at the origin) and plots the posterior only at two chosen dimensions. This method is suitable for dealing with low-dimensional distribution. However, it could be misleading in higher dimensional distributions with more complex geometry. To illustrate this complexity and potential bias caused by a cross-section plot, we construct a modified banana-shape distribution in 4-dimension:

$$\begin{aligned} y \mid \boldsymbol{\theta} &\sim \mathcal{N}\left(\mu_y, \sigma_y^2\right), \quad \mu_y := \theta_1 + \theta_2^2 + 0.3\theta_3 - 0.5\theta_4^2 \\ \theta_i &\overset{iid}{\sim} \mathcal{N}\left(0, \sigma_\theta^2\right) \end{aligned} \tag{4.19}$$

We generate $N = 100$ number of data $\{y_n\}_{n=1}^N$ with $\mu_y = 1$, $\sigma_y = 2$ and $\sigma_\theta = 1$. The resulting distribution is shown in the cross-section plots in Figure 4.8. Although a simple distribution, the morphology is quite complex: the first and third plots show two modes in the distribution, while the mid-plot shows a single mode. However,



high resolution no step-function

high resolution forward gravity

low resolution no step-function

no step-function RMSE: 0.175

low resolution with step-function

Before Optimization RMSE: 0.054

low resolution optimized step-function

After Optimization RMSE: 0.045

Figure 4.7: Gravity forward simulations. The first row shows the benchmark model with a high-resolution level-set method. The second row shows the low-resolution geological model and its forward gravity error with level-set method. Third row is the low-resolution geological model and forward gravity error with a non-optimized slope. The forth row shows is the low-resolution geological model and forward gravity error with the optimized slope.

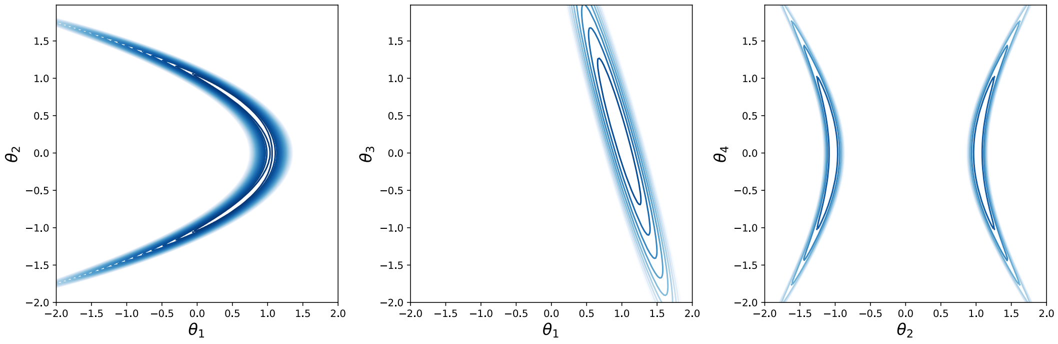


FIGURE 4.8: Cross section of the distribution described in Equation 4.19 with $N = 100$,$\mu_y = 1$, $\sigma_y = 2$ and $\sigma_\theta = 1$

only by combining the three plots can one comprehend the actual geometry. The parameters in a geological model are higher-dimensional and could correlate in a more complex manner. This will be difficult to visualize. Hence, to have a concise and comprehensive visualization of the posterior, we propose a method to visualize the posterior by adapting the dimension-reduction visualization method of loss surfaces in ANN (Li et al., 2018) as a qualitative way to evaluate a trainable geological model.

4.4.2 Loss Landscape Method

We propose to adopt the 2D random direction method as a dimensional-reduction visualization method for the posterior distribution. This original method plots the loss function with the following form:

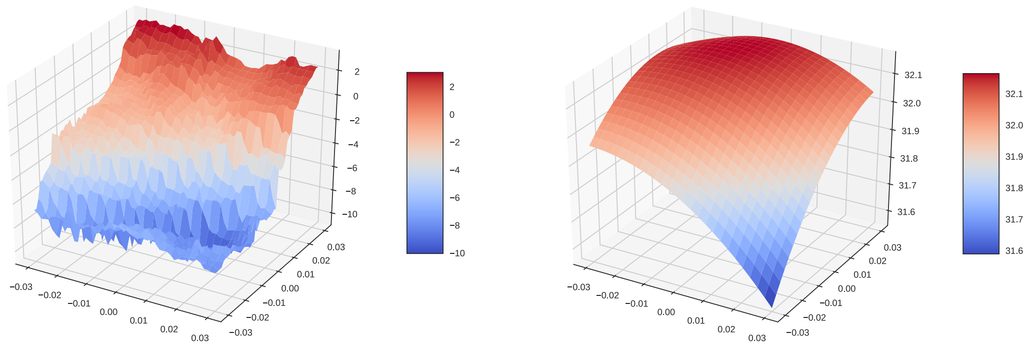
$$f(\alpha, \beta) = L\left(\boldsymbol{m}^* + \alpha\boldsymbol{\delta} + \beta\boldsymbol{\eta}\right) \tag{4.20}$$

where $\boldsymbol{\delta}$ and $\boldsymbol{\eta}$ are two randomly chosen direction vectors, α and β are two scalar parameters, $\boldsymbol{m}^*$ is the center point, and L is the loss function. This method has been used to provide an empirical analysis of the deep neural network loss function, and the optimization algorithms (Goodfellow et al., 2014; Im et al., 2016). This 2D method has advantages over its 1D counterpart to better visualize the non-convexity in the loss function (Goodfellow et al., 2014). A variant of this method with a "filter normalization" is used to analyze the trainability of neural networks (Li et al., 2018).

In the case of Bayesian inference in a geological model, the loss function is replaced by the posterior probability function p and plots the following function:

$$f(\alpha, \beta) = p\left(\boldsymbol{m}^* + \alpha\boldsymbol{\delta} + \beta\boldsymbol{\eta} \mid \boldsymbol{d}_{obs}\right) \tag{4.21}$$

A numerical example of the posterior (likelihood) surface is plotted using this method and discussed in the following section.



(a) likelihood without step-function

(b) likelihood wit step-function

FIGURE 4.9: The likelihood surfaces with/without step-function.

4.5 Numerical examples

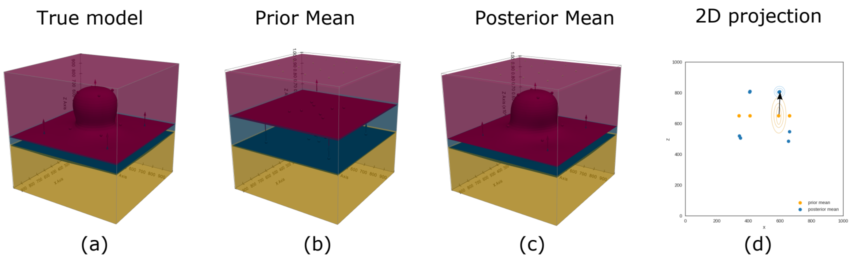
4.5.1 Posterior Landscape of the geological model

We use the above-described visualization method to plot the posterior landscape of the geological model in Figure 4.7. We use the synthetic data from the high-resolution model as the observational data for the inference and analyze the posterior surface of the model with level-set method and step-function. The parameters of interest are the z values of the surface points, which are 36 in total. To emphasize the effect of the aliased effect, we chose to only plot the likelihood function, which is essentially equivalent to using a uniform prior distribution. The resulting plots are shown in Figure 4.9.

The likelihood function surface of the model using the level-set method (Figure 4.9a) displays strong non-convexities. In contrast, the model using a step-function shows a smooth surface and displays convex contours in the region of plot. The value scale difference between the two plots is evident in the simulation misfit discussed in Figure 4.7. However, we should emphasize that the displayed likelihood surface is continuous in the contour plot, although it shows significant non-convexity. This continuity does not imply continuity in the actual likelihood function. The derivative information does not exist in the level-set models (as previously discussed in Section 4.3.1).

4.5.2 Numerical example of probabilistic gravity inversion

We evaluate the above-described 'trainable' concept in a simplified proof of concept resembling a 3D dome structure. These types of structures often exist in sedimentary basins, for example, in the Gulf of Mexico (Stern et al., 2011), Gulf Coast (Barton, 1933). In our simplified representation, the dome structure consists of a basement at the bottom, with the dense dome body in the middle and overlain by lighter rocks. To prove the concept, we only take the eight surface points at the upper surface as



(a) (b) (c) (d)

FIGURE 4.10: 3D example of a dome structure. (a) synthetic true model (b) prior mean and initial position (c) posterior mean by 1000 iterations HMC (d) 2D project of the surface points, black arrow illustrates the movement of distribution after inversion

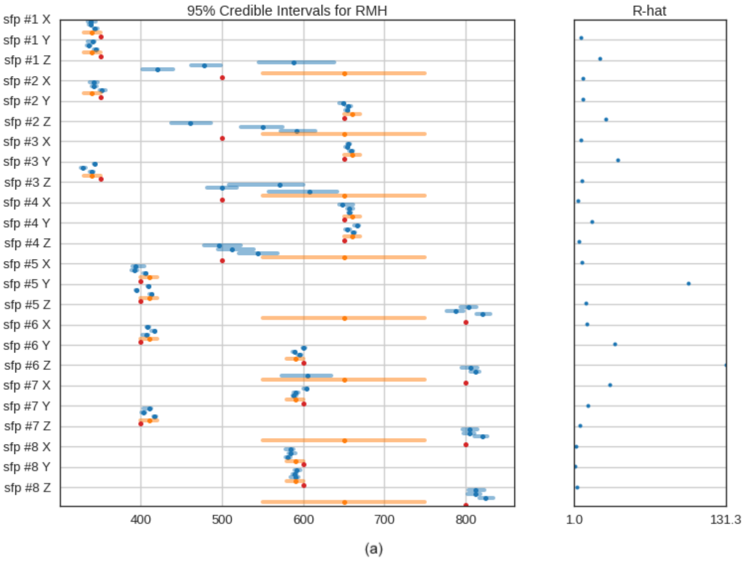
probabilistic variables while we keep the other parameters fixed. We generate synthetic gravity data by forward simulation, and subsequently invert for the x, y, and z coordinates of the 8 surface points at the upper surface, resulting in a total of 24 probabilistic parameters implemented by TensorFlow Probability distributions.

We assigned the model with a weak prior with the mean position to be at a flat surface at $z = 650$, and all x- and y-positions deviated from the truth. The prior distribution is given as an independent multivariate normal distribution with the mean shown in Figure 4.10. The synthetic observation data is in a grid at the top surface. We perform three runs with individual chains using both RMH and HMC methods. Each chain with 1000 iterations. 1000 Burn-in steps for RMH and 100 Burn-in for HMC, resulting in chains of 1000 samples from the posterior distribution.

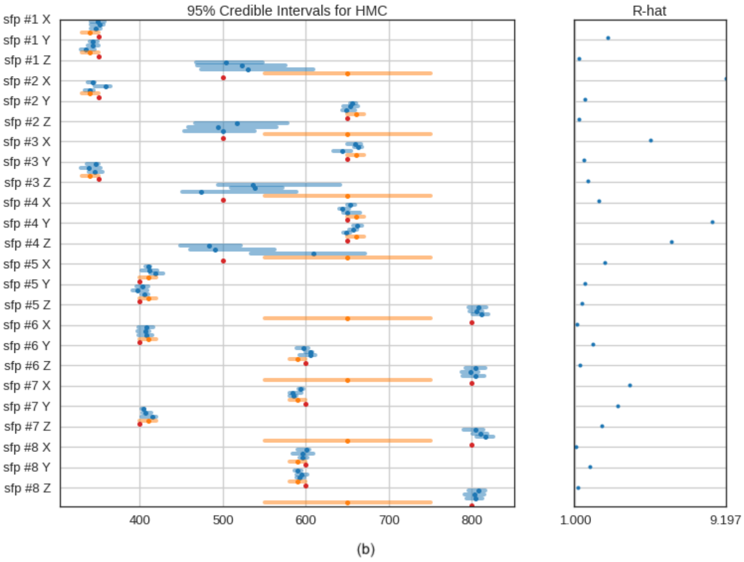
The result shows that, even with an extremely weak prior knowledge of the input parameters, with Monte Carlo methods, the general geometric setting can be retrieved. We compared the potential scale reduction factor $\hat{R}$ (Gelman and Rubin, 1992) as a measure of chain convergence using the implementation in TensorFlow Probability (Lao et al., 2020) over the three individual chains shown in Figure 4.11. The comparison shows that HMC generally leads to better-mixed chains when compared to the derivative-free RMH, resulting in more efficient posterior exploration and a better ability to capture the diversity of the realizations. This observation is consistent with findings by other researchers using HMC (Izmailov et al., 2021). However, the deviation of the posterior distribution also shows the limitation of HMC to jump between poorly connected modes in multimodal density (Mangoubi et al., 2018). This implies that good prior knowledge from geologists to the input of the model is indispensable. The purpose of this example is to show how the gradient information can assist in the sampling of the posterior distribution in the context of geological modeling and geophysical inversion.

4.6 Discussion

We have shown in the chapter that the introduced smooth step-function provides a smooth posterior landscape and grants trainability to the geological model. Our method is different from the anti-aliasing method proposed by (Scalzo et al., 2021).



(a)



(b)

FIGURE 4.11: Inversion results of three runs with individual chains using RMH (a) and HMC (b) . sfp $\#n$ stands the nth surface points on the top surface in the dome inversion example. The x coordinates represent the value of the parameters. The blue bar shows the resulting mean value (dots) and 95% credible intervals. The orange bars present the prior distribution. The red dots represent the parameters for the true model

While Scalzo et al. (2021) also aimed to achieve a smooth posterior surface, the method is based on the interpolation of the values close to the surface boundaries. Our method directly interacts with the scalar field, and no additional cost is required to approximate the partial volumes and calculate the unit normal of the surfaces. One could choose different slopes for different interfaces and could even adapt the slope to the sensitivity matrix based on their choice.

However, there are some limitations to the proposed method. First, although the numerical example shows that the smoothed model gives more faithful simulation results in the low-resolution case, better results are not guaranteed. In the case where the geological interface is exactly aligned with the surfaces of the regular grid, the level-set method will provide the exact result with no aliasing effect, regardless of the resolution. On the contrary, the step-function will generate a smooth surface by choice of an upper-bound value l_{max}. This is mainly because the step-function is not designed primarily to provide a more faithful gravity simulation. Rather, it is designed to keep the derivatives over the sharp interface. A unique slope value can not satisfy the optimal simulation results with a varying structure. Second, the defined slope function is suitable for a regular tensor grid. For grids with more complex configurations, for example, the ML optimized grid (in Chapter 3) and the octree mesh, the unique slope value could lead to over-smoothing in the part with smaller cells and gradient-vanishing in the other part with larger cells. Hence, the unique slope value choice is no longer the best option in this scenario, and future study is needed to address the step-function in more complex grid configurations.

The choice of the step-function in this study is a sigmoid function that extends to the infinite. Compared to other smooth step-functions with limited extension, for example, cubic polynomial and quartic polynomial, the main advantage of this type of infinite step function is the flexibility of the chosen grid type. The step-function acts directly on the scalar field, which means that any choices of the grid can be used to query the property values after the interpolation of the scalar field. Although that won't guarantee that the simulation result can be kept faithful, and the problem of vanishing gradient could appear with a badly chosen grid, but it opens the probability of choosing an irregular grid.

The optimization method for the slope function used here is gradient-descent. It does not guarantee a global minima to be found. However, the slope function is a 1D problem. One could certainly use a more expensive grid-based search method for the optimization. Nevertheless, the optimization of a pre-computation of the high-resolution model, and the optimized slope might not be optimal once the geometry is updated. Also, the improvement given by the optimization depends heavily on the geometry at that state. One should consider the use of an optimization based on the actual application scenarios. With a fixed, regular grid configuration, other options of finite step-functions are also possible and will provide similar results with a proper choice of scaling, etc. However, when it comes to the application of second-order derivatives, the chosen step-function should be twice-differentiable at the parameter locations, which is the origin of the step-function in the current configuration. Hence the infinitely differentiable sigmoid function is a good choice for this purpose.

The posterior surface visualization method proposed in this study provides an intuitive and concise view of the posterior in high dimensions. However, with the dramatic dimension reduction, the plot is rather a qualitative characterization of the posterior function rather than a rigorous display of the posterior convexity. The apparent convexity in the reduced-dimension surface does not mean true convexity in the high-dimensional function. Rather it shows the function is dominated by positive curvatures or a positive mean curvature (Li et al., 2018). However, when non-convexity is present in the reduced-dimensional plot (e.g., Figure 4.9a), the non-convexities must be present in the full-dimensional function. In the case of a multimodal posterior surface, the introduced smooth step-function also helps to remove the 'fake' multimodality caused by the discretization.

4.7 Conclusion

In this chapter, we have introduced the use of smooth step-function to generate trainable geological modeling in a gravity inference problem. The model is implemented end-to-end trainable in the AD package TensorFlow. The primary goal of the step-function is to provide the trainability to the geological to allow updating using derivative information. Trainability is the foundation for the application of the advanced derivative-informed inference method, and we have shown a successful application of the HMC method on a synthetic dome model. The step-function also provides better gravity simulation results in the cases of aliased low-resolution models.

Chapter 5

Hessian-informed MCMC

The text in this chapter is published in Geophysics (Liang et al., 2022).

5.1 Introduction

In many geoscience applications, inversion methods are used to estimate subsurface properties (e.g., structure, density and porosity) from observed geophysical data. Conventional geophysical inversion aims to find the best-fit parameter sets that minimize the error between observed geophysical data and simulation results. However, in practical cases, observational data has the general difficulties of sparsity and noise. The uncertainties in the data are combined with the geometry errors from the geological model constructions, leading to numerous possible solutions within the uncertainty range, and hence ill-posedness. Geoscientists are therefore interested in not only the best-fit model parameters but also a quantification of uncertainties associated with these parameters (e.g., Sen and Stoffa, 1996; Wellmann et al., 2018; Witter et al., 2019; Chauhan et al., 2021).

In this study, we focus on the uncertainty quantification in model-based inversion problems. Developments in geomodeling have provided us with tools to generate 3D structural geological models based on interface and orientation information from observed or inferred data (Caumon et al., 2009; Caumon, 2010; Wellmann and Caumon, 2018). In contrast to directly inverting the property field (e.g. density or thermal conductivity field), parametric geological models have the advantage of easy interpretability and low parameter dimensionality. Geophysical data can be used as an additional constrained to the developed model. The gravity method has been widely applied in geophysics to investigate subsurface geometries and properties (Nabighian et al., 2005a). While conventional gravity inversion often suffers from ill-posedness due to insufficient data and intrinsic ambiguity (Skeels, 1947; Parker, 1975, 1974), it is suitable as additional data to constrain a geological model.

Recent developments in structural geomodeling methods allow geoscientists to quantify the uncertainties in geological models based on prior geological knowledge and additional geophysical data in a Bayesian inference approach (Wellmann et al., 2010; de la Varga and Wellmann, 2016; Wellmann et al., 2018). The Bayesian approach has long been used as a tool to quantitatively combine uncertainties from various sources in a probabilistic workflow (Tarantola and Valette, 1982; Mosegaard and Tarantola, 1995; Sambridge and Mosegaard, 2002; Sambridge et al., 2013). In a Bayesian inference problem, the expert knowledge of the geological model (prior information) and the observational data (likelihood function) are combined probabilistically. The inference provides us the updated uncertainty range of the parameters of interest given the observational data. This output uncertainty range is known as the posterior, which is also a distribution. However, it is often not possible to calculate the posterior uncertainty analytically. The Markov chain Monte Carlo (MCMC) is used to sample from the posterior distribution by generating Markov chains of samples e.g. according to the Metroplis-Hastings algorithm (Metropolis et al., 1953; Hastings, 1970). The posterior distribution can then be approximated by the density distribution of the generated samples. The samples can also provide useful statistical information about the posterior probability density function (PDF) (e.g., mean and standard deviation).

The random walk Metropolis-Hasting (RMH) algorithm is likely the most popular MCMC method. It is the simplest algorithm and has been widely adopted from the pioneering applications in traditional geophysical inversion (Mosegaard and Tarantola, 1995; Malinverno and Leaney, 2000; Malinverno, 2002) through the recent applications of geological modeling inversion (de la Varga and Wellmann, 2016; Wellmann et al., 2018; de la Varga et al., 2019). While the classical RMH algorithm is simple in its intuition and implementation, it is often inefficient for complex posterior distributions. Complexity may result from high dimensionality or strong nonlinearities of the model. This leads to slow convergence of the MCMC chain (Ruggeri et al., 2015). The inefficiency results primarily from the mismatch between the proposal distribution used in the random walk method and the actual posterior distribution. The recent work of Scalzo et al. (2019) used a preconditioned Crank–Nicolson (pCN) method to address the high dimensionality problem in the context of the joint geophysical inversion problems.

In cases where the gradient of the negative logarithm of the posterior with respect to the parameters can be obtained easily and efficiently, gradient information can accelerate a MCMC method by biasing samples toward higher probability regions. The Metropolis adjusted Langevin algorithm (MALA) (Rossky et al., 1978; Roberts et al., 1996), which involves Langevin diffusions, utilizes gradient information to confine the induced pseudodynamics (Betancourt, 2019). MALA can significantly increase the efficiency of MCMC, however, its performance deteriorates for strongly anisotropic distributions since the gradient may no longer point in a globally useful direction. A recent example of the application of this algorithm to the seismic inversion problem can be found in the work of Mosser et al. (2020). Another popular gradient-informed algorithm, the Hamiltonian Monte Carlo (HMC) method, also known as hybrid Monte Carlo, was first introduced by Duane et al. (1987). HMC utilizes gradient information to find the next independent sampling points to achieve a more efficient exploration of the typical set at the cost of numerical integrator for solving the Hamiltonian equation at each step. (Betancourt, 2019). In addition, the parameters for the leapfrog iterations require careful tuning, and the computational cost of leapfrog steps can be substantial (Girolami and Calderhead, 2011). Recent studies have shown the applications of HMC in geophysical inversion (Muir and Tkalcic, 2015; Fichtner et al., 2019).

The use of higher-order derivative information in MCMC, in particular the Hessian of the negative log-posterior has been shown to lead to more rapid exploration of the posterior by adapting to the posterior curvature and anisotropy (Geweke and Tanizaki, 1999; Qi and Minka, 2002). However, explicitly constructing the Hessian for high dimensional inverse problems is prohibitive. Instead, for many ill-posed inverse problems, the Hessian of the negative log-likelihood admits a low rank approximation, and this can be efficiently computed using a randomized eigensolver along with higher order adjoints (Martin et al., 2012; Petra et al., 2014; Isaac et al., 2015). A number of advanced Hessian-based MCMC methods for large-scale inverse problems have appeared in the past decade, for example, the stochastic Newton MCMC method (Martin et al., 2012), dimension-independent, likelihood-informed MCMC (Cui et al., 2016) and adaptive Gaussian process emulated geometric Monte Carlo (Lan et al., 2016). These Hessian-based methods overcome the slow convergence of gradient-only MCMC for ill-posed problems and are applicable in high-dimensional parameter space.

Developing first and higher order adjoints for complex simulations can be challenging. As an alternative, automatic differentiation (AD) can be employed. The earlier development of geological modeling methods by de la Varga et al. (2019) utilized automatic differentiation (AD) to make gradient evaluations easily tractable. Recent work of Güdük et al. (2021) presented the successful application of this technique to model-based probabilistic inversion using HMC. In this work, we introduce the Hessian-based MCMC method as a tool to conduct efficient uncertainty quantification in model-based geophysical simulation. We follow the implicit geological modeling method and forward geophysical simulation introduced by de la Varga et al. (2015, 2019) and adapted the Hessian-based MCMC algorithm developed by Villa et al. (2018). Here, we first find the maximum a posteriori (MAP) point by using Adam, a adaptive gradient-based optimization method (Kingma and Ba, 2014). Then we construct the Laplace approximation of the posterior PDF by equating the posterior covariance to the inverse of the Hessian of the negative-log posterior at the MAP point. A generalized preconditioned Crank-Nicolson proposal (gpCN) (Pinski et al., 2015) is then applied to conduct the MCMC sampling. Most importantly, to efficiently calculate the second-order derivatives, we further extend the geomodeling framework GemPy (de la Varga et al., 2019) to an implementation in TensorFlow (Abadi et al., 2015), a machine-learning framework that allows fast derivative evaluation through AD. The second-order derivative is calculated by a forward-over-backward propagation to achieve time and memory efficiency. We establish a conceptual 3D fold model and a case study of Kevitsa deposit 3D model in order to compare the efficiency of the proposed method using gpCN algorithm against the RMH and HMC methods. The results show the potential of Hessian-based MCMC methods in real-world geophysical inversion applications.

5.2 Method: Generalized Preconditioned Crank-Nicolson (gpCN)

In this section, we conducted an end-to-end procedure from the model construction to the geophysical simulation and finally to Bayesian inference. We first generated a geological model by using the implicit modeling method. A synthetic gravity field was then simulated based on the model. Finally, we apply three different MCMC algorithms, namely RMH, HMC, and gpCN to solve the Bayesian inference on the same synthetic model. In this section, we describe the automatic differentiation method which links the geological modeling and Bayesian inference framework and makes the high-order derivative evaluation tractable, and present, how the high-order derivative information can be used to accelerate MCMC.

Although it is intuitively simple, the RMH algorithm suffers from the problem of dimensionality (Cotter et al., 2013; Betancourt, 2017). As the dimension increases, the naive symmetric proposal used in the RMH algorithm will result in most of the samples being rejected, and thereby resulting in an inefficient exploration of the posterior. A decent volumetric explanation was established by Betancourt (2017). Thus, researchers have been searching for an alternative to the RMH algorithm. Cotter et al. (2013) introduced the preconditioned Crank–Nicolson (pCN) proposal, which is a slight modification to the original RMH algorithm but provides its scalability to high-dimensional problems.

In recent years, the idea of employing the geometry of the posterior to accelerate the exploration has attracted many researchers (Girolami and Calderhead, 2011; Martin et al., 2012; Law, 2014; Cui et al., 2016), where the first-order or second-order derivative information or either both (Martin et al., 2012) are utilized. Among those studies, Pinski et al. (2015) and Villa et al. (2018) introduced the proposal preconditioned by the Gaussian approximations of the probability measure of interest (which we referred as the gpCN here) to improve the pCN proposal. This is a modified version of gpCN which differs from the original gpCN (Rudolf and Sprungk, 2018) by the way of keeping the reversibility of the algorithms.

Villa et al. (2018) introduced the method to evaluate the posterior covariance by using a Laplacian approximation (Stigler, 1986; Wong, 2001; Evans and Swartz, 2000; Kerkering, 2003; Tierney and Kadane, 1986) with the Hessian of the log likelihood $\mathcal{H}_{misfit}$ at the MAP point $\boldsymbol{m}_\nu$ and covariance of the prior C_{prior}

$$C_\nu = \mathcal{H}\left(\boldsymbol{m}_\nu\right)^{-1} = \left(\mathcal{H}_{misfit}\left(\boldsymbol{m}_\nu\right) + C_{prior}^{-1}\right)^{-1} \tag{5.1}$$

where $\mathcal{H}\left(\boldsymbol{m}_\nu\right)$ is the Hessian of the log posterior evaluated at the MAP point $\boldsymbol{m}_\nu$,

$$\boldsymbol{m}_\nu := \arg\min \mathcal{J}(\boldsymbol{m}) := \left(\Phi(\boldsymbol{m}, \boldsymbol{d}_{obs}) + \frac{1}{2}\left\|\boldsymbol{m} - \boldsymbol{m}^{prior}\right\|^2_{C_{prior}^{-1}}\right) \tag{5.2}$$

Here, we applied the recently developed first-order gradient optimization algorithm Adam (Kingma and Ba, 2014) to find the MAP. The update rule of Adam is given as follows:

$$
\begin{aligned}
\boldsymbol{b}_t &\leftarrow \beta_1 \cdot \boldsymbol{b}_{t-1} + (1-\beta_1) \cdot g_t \\
\boldsymbol{v}_t &\leftarrow \beta_2 \cdot \boldsymbol{v}_{t-1} + (1-\beta_2) \cdot g_t^2 \\
\hat{b}_t &\leftarrow \boldsymbol{b}_t / \left(1-\beta_1^t\right) \\
\hat{v} &\leftarrow \boldsymbol{v}_t / \left(1-\beta_2^t\right) \\
\boldsymbol{m}_{t+1} &\leftarrow \boldsymbol{m}_t - \frac{\alpha}{\sqrt{\hat{v}_t} + \epsilon} \hat{b}_t
\end{aligned}
$$

where α is the learning rate, t denotes the time step. β_1, β_2 and ϵ are parameters. A normal configuration of the parameters is $\beta_1 = 0.9$,$\beta_2 = 0.999$ and $\epsilon = 10^{-8}$. $\boldsymbol{b}$ and $\boldsymbol{v}$ are the bias-corrected first and second momentum vector estimates of the gradients, respectively. The momentum term averages the past gradient and thus accelerates the convergence rate compared with the standard gradient descent algorithm, while the second momentum which averages the past squared gradients adapts its learning rate.

Here we follows the original form of gpCN introduced by Pinski et al. (2015). The key idea of this algorithm is to employ the covariance of the posterior C_ν to construct a proposal distribution that adapts the posterior geometry (Figure 5.1). The algorithm can be summarized as follows:

```
Set $k = 0$ and initiate with $\boldsymbol{m}^{(0)}$
while $k <$ number of samples do
    Propose $\boldsymbol{m}^{cand} = \boldsymbol{m}_\nu + \sqrt{(1-\beta^2)}(\boldsymbol{m}^k - \boldsymbol{m}_\nu) + \beta\boldsymbol{\xi}^k$, $\boldsymbol{\xi} \sim \mathcal{N}(0, C_\nu)$
    Set $\boldsymbol{m}^{k+1} = \boldsymbol{m}^{cand}$ with acceptance probability $a$
    Set $\boldsymbol{m}^{k+1} = \boldsymbol{m}^k$ otherwise
end
```

Algorithm 4: GENERALIZED PRECONDITIONED CRANK–NICOLSON ALGORITHM ALGORITHM (Pinski et al., 2015; Villa et al., 2018)

where the acceptance probability $a(\boldsymbol{m}^{cand}, \boldsymbol{m}^k) = \min\left\{1, \exp\left(\Delta\left(\boldsymbol{m}^k\right) - \Delta\left(\boldsymbol{m}^{cand}\right)\right)\right\}$, and $\Delta(\boldsymbol{m}) = \boldsymbol{m}_\nu - \frac{1}{2}\left\|\boldsymbol{m} - \boldsymbol{m}_\nu\right\|_{C_\nu^{-1}}^2$,

$\Phi(\boldsymbol{m}, \boldsymbol{d}_{obs})$ is the negative log likelihood function: $\Phi(\boldsymbol{m}, \boldsymbol{d}_{obs}) = \frac{1}{2}\left\|\boldsymbol{f}(\boldsymbol{m}) - \boldsymbol{d}_{obs}\right\|_{\Gamma_{noise}^{-1}}^2$

5.3 Efficient Hessian Calculation

Implementing derivative-informed MCMC methods such as HMC and gpCN in geomodeling requires the derivative information in several steps such as finding the MAP point and constructing the posterior covariance approximation. This derivative refers to the derivative of the posterior (or negative log posterior) with respect to the model

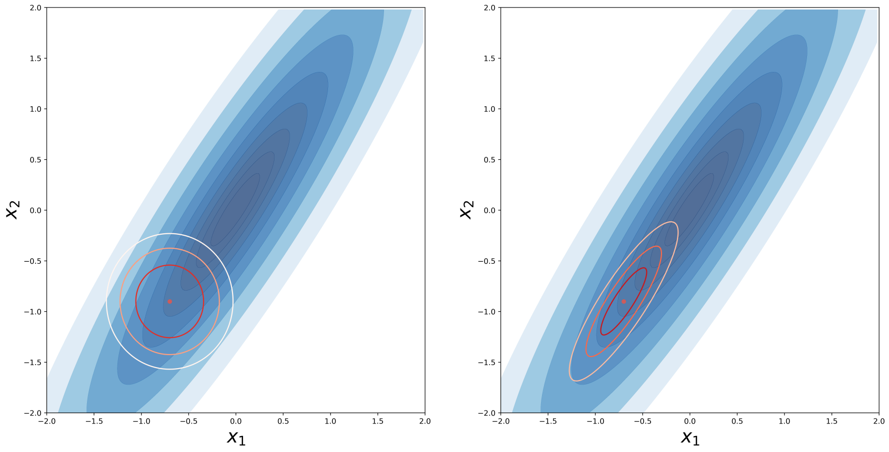


FIGURE 5.1: Different proposal distribution illustrated on the 2D Gaussian posterior distribution $\boldsymbol{d}|\boldsymbol{m} \sim \mathcal{N}(\mu, I), \mu := x_1 - 0.7x_2$, where $x_i \overset{iid}{\sim} \mathcal{N}(0, I)$. Left shows the Gaussian proposal without preconditioning. Right shows the proposal distribution with Laplacian approximation at MAP.

input parameters (e.g., surface points and orientation points). By tracing the derivative from the Bayesian inference through the geophysical simulation, geological modeling of the input parameters is nontrivial. Although evaluating the derivatives from the co-Kriging and gravity simulation is simple individually, the combination of them into a geological modeling framework is not straightforward. An analytical solution of the derivatives is difficult or even impossible to find. Using numerical methods such as finite difference (FD) is not only computationally costly but also can suffer from numerical inaccuracy. Therefore, we adopted the automatic differentiation (AD) technique, which is widely applied in the field of artificial intelligence and is critical to the success of training neural networks. Here we briefly introduce how AD works and how gradient and higher-order derivatives can be evaluated efficiently using AD.

Efficient second-order derivate calculation for multivariate input and single output problems is given by the combination of a forward-over-backward propagation. Applying backward propagation gives us the Jacobian matrix $\mathcal{J}_F(x) = [\frac{\partial F}{\partial x_1}, \frac{\partial F}{\partial x_2}, \ldots, \frac{\partial F}{\partial x_j}]$. The second derivative then evaluates the partial derivative over the Jacobian, which is an $\mathbb{R}^{D_I} \rightarrow \mathbb{R}^{D_I}$ problem. Considering the memory efficiency, we can evaluate the Hessian vector product efficiently by an additional forward sweep over the backward propagation. Each forward-over-backward iteration will return a column of the Hessian matrix; therefore, this method has an $\mathcal{O}(n)$ complexity, where n is the dimension of the parameter of interest. It is significantly more efficient than evaluating the Hessian by the FD method, which has a complexity of $\mathcal{O}(n^2)$. A comprehensive review of AD and its implementation can be found in Betancourt (2018) and Margossian (2019).

Programming the geological model by adopting AD, a second-order derivative can be efficiently evaluated. The numerical comparison of the computational efficiency and precision are discussed in the following sections.

5.4 Results

5.4.1 Synthetic Models

To configure such a problem in a Bayesian inference framework as described above, both the prior information and the likelihood must be expressed in terms of a probability distribution. Three 3D geological models with a gradually increasing complexity are established using the methods described above and the numerical results are presented below. The numerical implementation is based on the previously developed modeling methods of the GemPy software (`https://www.gempy.org/`) and specific extension in the differentiable programming framework TensorFlow (`https://www.tensorflow.org/`). All the presented results were run on a single Tesla-P100 GPU.

Model A: Simple Dome

In the first example, a simple model is established to validate the proposed method. The example model has a simple dome geometry, replicating a conceptual setting that is often observed in real-world geology. It consists of three layers, representing three distinct lithologies. The model has an extent from 0 m to 1,000 m for all three axes. The positions (x-y coordinates) of the interface points are fixed, and the depth z values of the top layer (rock 2) interface points are the uncertain parameters in this study. Both the ground truth layout of the interface points and forward simulated gravity is shown in Figure 5.2. To simplify the problem, we assume a constant thickness between the two surfaces at configuration interfaces positions and only vary the upper interface points. In other words, the lower surface moves parallel in dependence of the upper surface; therefore, a roughly constant thickness of the middle layer is expected. Interpolation could introduce some variation to the thickness at locations where no interface points are given. Sixteen gravity receivers are evenly located at the ground surface, which is the top surface of the model. The grid resolution for gravity calculation was selected to be $10 \times 10 \times 30$ to have a higher resolution in the vertical axis because the parameters of interest vary along the z axis. Our goal is to estimate the updated uncertainties of the depth position of the interface points of the upper surface considering the additional gravity information.

We assume a simple Gaussian prior to all eight interface points with the mean at a flat surface as $m_0 = 780$ and a standard deviation σ_0 of 100 m. Therefore, the prior distribution can be expressed as a multivariate normal distribution centered at m_0 and a diagonal covariance matrix assuming no correlations between the parameters in our prior information as $p_{\boldsymbol{m}} \overset{iid}{\sim} \mathcal{N}(m_0, \sigma_0)$. An uniform density for each layer is used for simplification. The density of each lithology is assigned as 2.6 $\mathrm{g/cm^3}$, 3.5 $\mathrm{g/cm^3}$

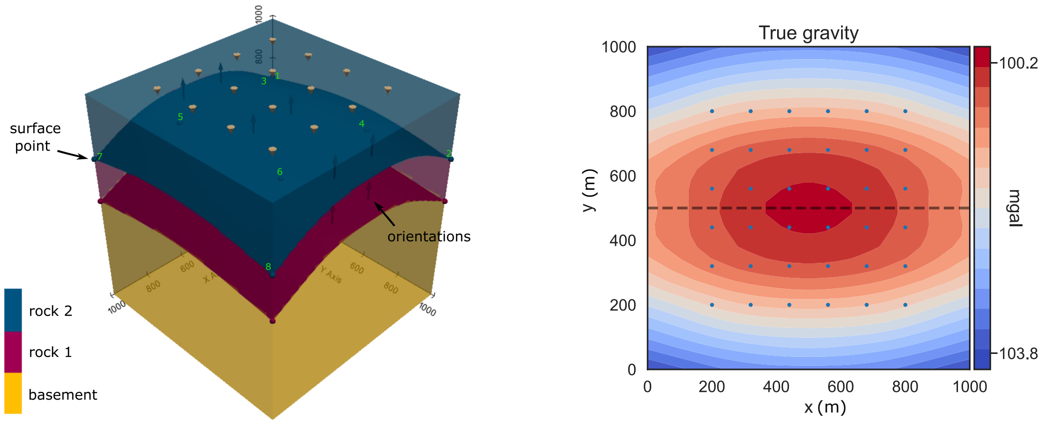


FIGURE 5.2: Left: 3D plot of the base-case geological model. The ground truth z value of the top layer interface points are placed at depth 705 m and 805 m. Gravity receivers are placed at the top denoted by the cones. Surface points are numbered and denoted as spheres. Orientations are denoted as arrows. Right: forward gravity simulation of the ground truth model. The dashed line denotes the cross-section to evaluate entropy. Gravity data used in the inference at a 6 × 6 grid receivers are denoted by the blue dots. A dense 30 × 30 grid of gravity is evaluated for visualization and evaluation.

and 2.0 $\mathrm{g/cm^3}$ respectively from top to bottom. The observation data $\boldsymbol{d_{obs}}$ was generated by a forward simulation of the ground truth model. The likelihood function is defined as another multivariate normal distribution $p\left(\boldsymbol{d}_{obs} \mid \boldsymbol{m}\right) \sim \mathcal{N}(f(\boldsymbol{m}), 0.01)$ with the mean at the forward simulation of a given parameter sets $\boldsymbol{m}$.

To solve the Bayesian inference by MCMC, specifically by using the gpCN, we first search for the Maximum a Posteriori (MAP) point $\boldsymbol{m}_{MAP}$ in the posterior space. $\boldsymbol{m}_{MAP}$ can be sufficiently found by Adam with the assistance of the gradient of the target negative log posterior with respect to the variable (depth value). The gradient is evaluated through AD. The time cost for each gradient evaluation is within 1 s depending on the model size. The initial status is generated by randomly sampling the prior distribution. The convergence speed depends on the model configuration, stepsize, and the initial status. The full Hessian matrix of the target negative log posterior with respect to the variable is evaluated at the MAP point through the forward-over-backward propagation described above.

To compare the performance of gpCN, we implemented the two other most commonly used methods, namely RMH and HMC, to solve the same inference problem. We used 3 and 10 leapfrog steps for HMC, with the stepsize of 0.7. To have a fair comparison, we tune the acceptance rate of RMH and gpCN at both around 80% and neglect the high acceptance rate of HMC due to the algorithm design. The resulting sampling chain, posterior, and efficiency are compared below. Figure 5.3 shows the trace plot of the same two interface points for each algorithm. We observe that better-mixed chains are obtained by both HMC and gpCN, but the chain of RMH is poorly mixed, and far from convergent. The lag k autocorrelation is often used to quantitatively assess the Markov chains which is defined as the autocorrelation between samples at k steps apart (Cowles and Carlin, 1996; Roy, 2020). Here the lag k autocorrelation of

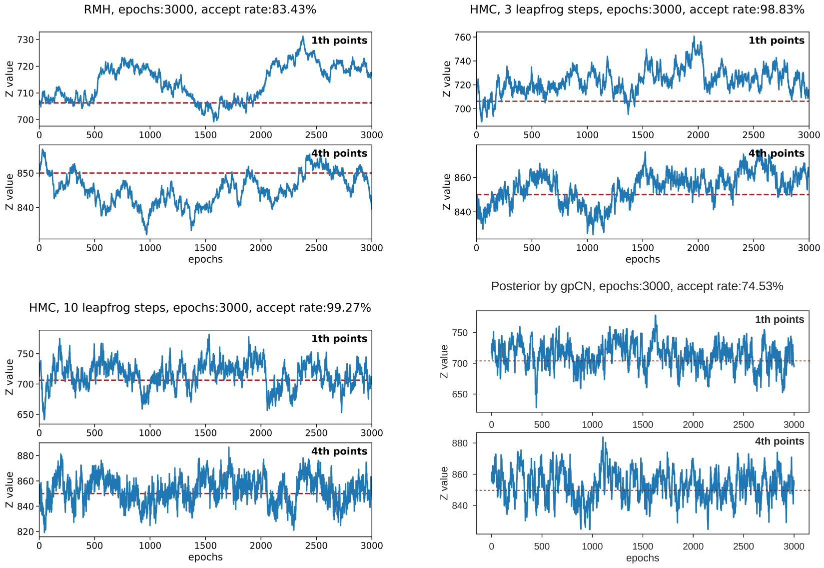


FIGURE 5.3: Trace plot of RMH, HMC, and gpCN. HMC with 3 and 10 leapfrog steps and stepsize of 0.7. The true values are denoted by dashed lines.

the results from all three methods are plotted in Figure 5.4. The plotted autocorrelation shows the mean and variation of all dimensions. gpCN and HMC with 10 leapfrog steps show generally less correlated samples, while samples by RMH and HMC with few leapfrog steps are highly autocorrelated with a similar acceptance rate. One can always tune the stepsize of MCMC to achieve fewer autocorrelated samples, but this will also lead to a lower acceptance rate.

While gpCN and HMC have comparable results in terms of autocorrelation, what is often not mentioned in the application of HMC is the cost of the numerical integration (leapfrog integrator used in this study). The efficiency of HMC depends on the efficiency of the gradient evaluation. While the gradient is easily accessible in some applications, in the case of geological modeling described in this work, the cost of the leapfrog integrator is a computationally demanding process even using AD, when the cost accumulated through a large number of leapfrog steps and MCMC iterations. The same demanding leapfrog integration to use HMC is also described in literatures (Neal et al., 2011, e.g.). The choice of stepsize and the number of steps used for the leapfrog integration are essential for the performance and efficiency of HMC. Large stepsize results in large integration error, while small stepsize will require more steps to draw uncorrelated samples. Because HMC is intended to find an uncorrelated point by using several evaluations, the theoretical acceptance rate of HMC is 100%, but slightly lower due to the imperfection of numerical integration. The impact of the error of integration caused by too large stepsize will also be reflected in the deviation of the acceptance rate. In terms of the number of steps, too few steps will result in a bad integration, and therefore, more correlated samples, while an increased number of steps will increase the computational cost. gpCN is more efficient in terms of computational cost with no accumulated cost for derivative evaluation. In order to assess the computational efficiency, in addition to the acceptance rate and the autocorrelation,

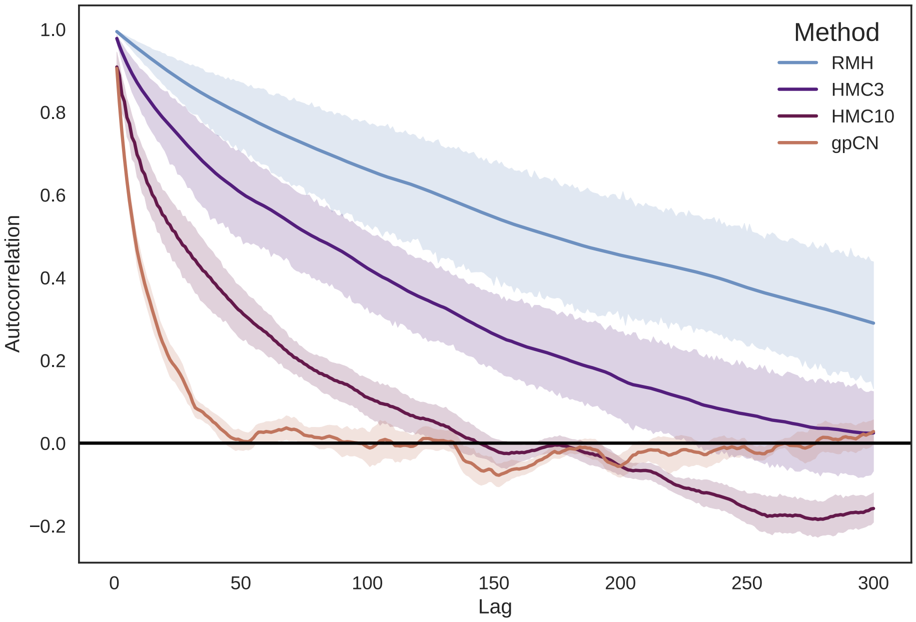


FIGURE 5.4: Autocorrelation plot of the MCMC chain of all three methods on Model A. The solid lines represent the mean autocorrelation among different variables, and the shadow represents the 95% confidence interval of the autocorrelation on different dimensions.

we also compared the effective sample size n_{eff} (Liu, 2008, p. 125) per unit time. The effective sample size n_{eff} is defined as follows:

$$n_{\text{eff}} = \frac{n_{\max}}{1 + 2\sum_{k=1}^{\infty} \rho_k} \tag{5.3}$$

where ρ_k is the autocorrelation coefficient at lag k. A higher number of n_{eff} means a better mixing Markov chain. Our numerical experiments indicate that gpCN outperformed carefully tuning HMC in terms of the $AvgN_{eff}$ (Table 5.1), while preserving the computational efficiency.

The main reason for the superior performance of gpCN is the correlated posterior. To demonstrate that, we plot the adopted proposal distribution of the gpCN on the posterior distribution (Figure 5.5), we can see that the proposal distribution captured the dominating posterior geometry, therefore a higher acceptance rate is expected in gpCN.

Finally, we represent the uncertainties following the information entropy method introduced by Wellmann and Regenauer-Lieb (2012). In Figure 5.6, we compared the uncertainties of the prior and posterior. We generated 1000 samples from the prior distribution and build the geological models based on the sampled parameters. The samples drawn by the gpCN MCMC are used to generate geological models for posterior representation. By converting the lithology changes into information entropy, we can see that the uncertainty range of the position of the interfaces is significantly

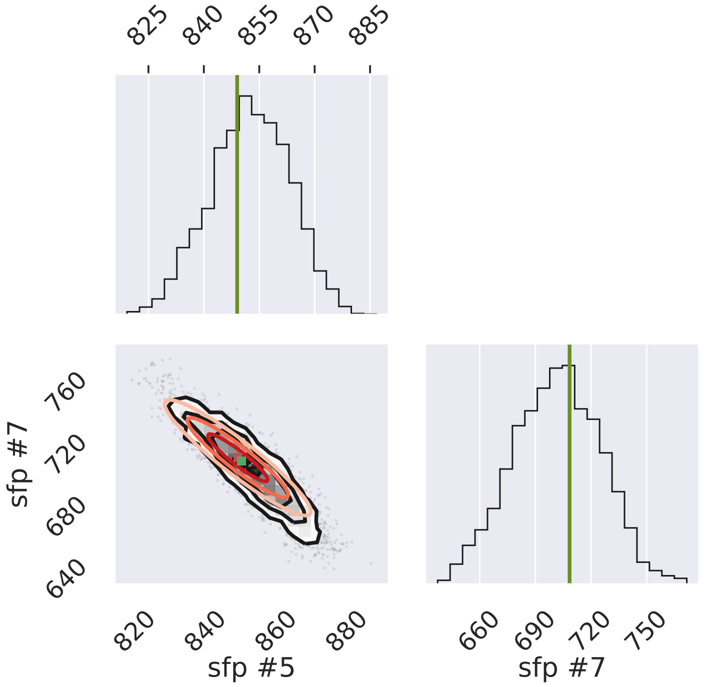


FIGURE 5.5: Illustration of the samples of surface points 5 and 7 by gpCN (grey dots). Black contour denotes the approximate probability density. The red contour denotes the proposal distribution used in gpCN which captures the geometry of posterior distribution. The green square and green lines denote the ground truth positions. The red square denotes the MAP points.

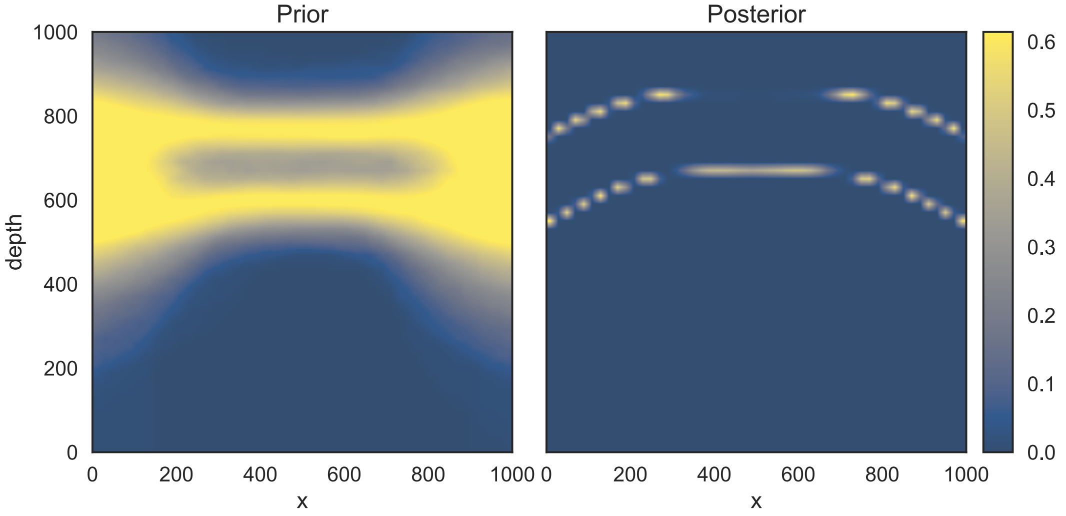


FIGURE 5.6: Information entropy plot of prior and posterior of model A evaluated on cross-section shown in Figure 5.2. High entropy represents high uncertainties.

reduced by additional gravity data which also successfully captures the true geometry.

Model B: Thickness as Additional Variables and Prior

In the next example we consider the uncertainty of the positions of both layer surface points. The ground truth model is the same as the previous example. The same 6×6 grid of receivers are deployed at the top surface for the evaluation. The prior distribution of the upper layer surface points is defined as $p_{\boldsymbol{mu}} \overset{iid}{\sim} \mathcal{N}(m_u, \sigma_u)$, where $m_u = 780$ and $\sigma_u = 60$. The lower surface points are defined as $p_{\boldsymbol{ml}} \overset{iid}{\sim} \mathcal{N}(m_l, \sigma_l)$, where $m_l = 200 + \zeta$, $sigma_l = 50$ and $\zeta \overset{iid}{\sim} \mathcal{N}(0, 30)$. The prior distribution $p_{\boldsymbol{m}}$ is then defined as the joint distribution $p_{\boldsymbol{m}} = p_{\boldsymbol{mu}} p_{\boldsymbol{ml}}$. The likelihood is defiend as a normal distribution with the standard deviation set to 0.5. Normalization is applied to the parameters according to the prior before MCMC due to the difference in the scales. In total 16 parameters are considered in the inversion problem. The initial states are randomly chosen in the prior distribution, and the same MCMC methods as in the previous example (Model A) are applied.

Similar superior efficiency is observed in the inference results (Figure 5.8 and Table 5.1). We present instances sampled from prior and posterior as well as the corresponding gravity in Figure 5.9. Each instance represents one realization generated by the modeling parameters randomly sampled from the prior and posterior. Compared to the ground truth model and gravity observation (Figure 5.2) forward simulation, the gravity simulation from the prior samples largely deviates from the truth. This large deviation illustrates poorly-constrained prior models and large uncertainties in the prior. The inferred posterior samples, however, captured the main geometry of the ground truth and reduced the uncertainty range. The posterior shows that the upper surface has a lower uncertainty than the lower layer, which results from the fact that the sensitivity of the gravity is high in the near-surface regions. The resulting forward

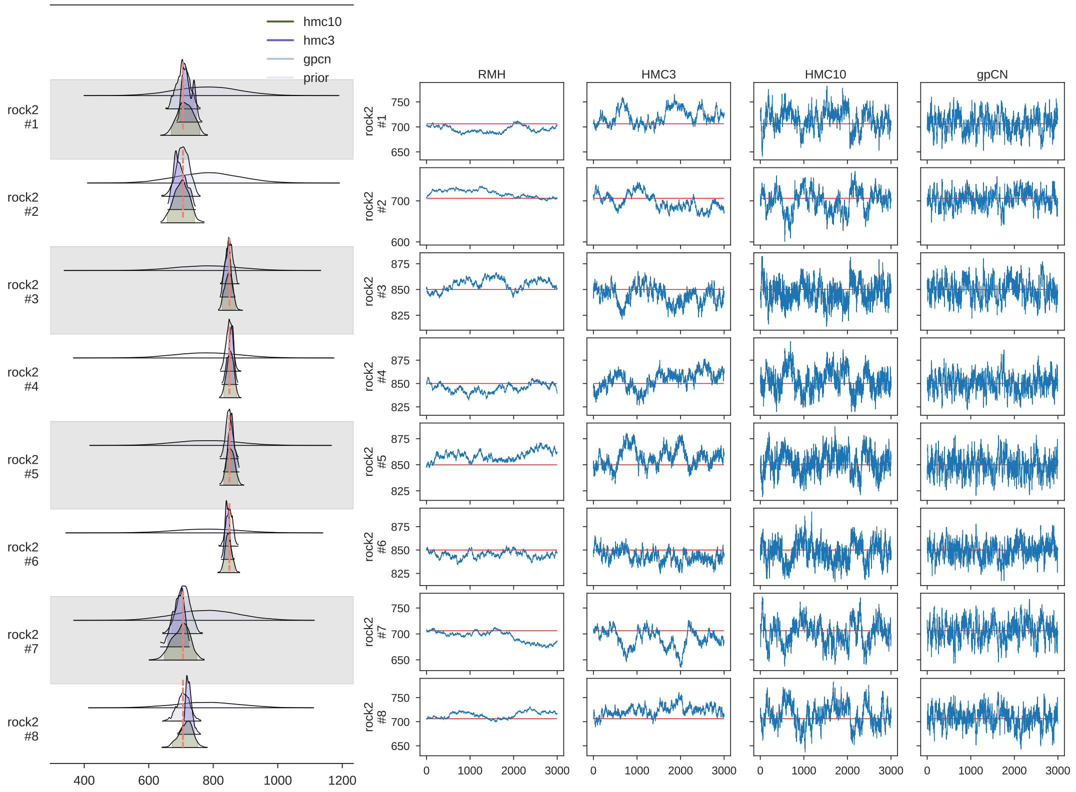


FIGURE 5.7: Empirical density by gpCN and HMC and the MCMC chains for model A. RMH acceptance rate: 83.4 %, HMC with 3 leapfrog steps acceptance rate 98.8 %, HMC with 10 leapfrog steps acceptance rate 99.3 %, gpCN acceptance rate 70.2%

gravity from the posterior samples does not fit perfectly with the ground truth value due to the less restricted likelihood function chosen. This allows us to keep the information of the prior knowledge and prevent it from reducing back to a maximum likelihood problem. This is the fundamental difference between Bayesian inference and minimizing the data misfit in conventional gravity inversion. The exhaustive chains and marginal distribution are summarized in Figure 5.10.

5.4.2 Case Study: Kevitsa Deposit

The Kevitsa deposit is located in northern Finland with a large amount of nickel/-copper reserves. The main ore body is hosted by an ultramatic unit and overlain by younger carbonaceous phyllite and basaltic komatiites units. This geologically complex region is interesting to both the mining industry as well as structural geological studies: e.g., Kevitsa 3D models (Koivisto et al., 2015; Güdük et al., 2021), Kevitsa gravity inversion (Fournier, 2019). The Kevitsa main intrusion mainly consists of olivine pyroxenites and olivine websterites. We chose this region to test the proposed method due to the interesting geological structure and rich available gravity data for validation.

Although a large number of drill holes are available in the study area, the depth of

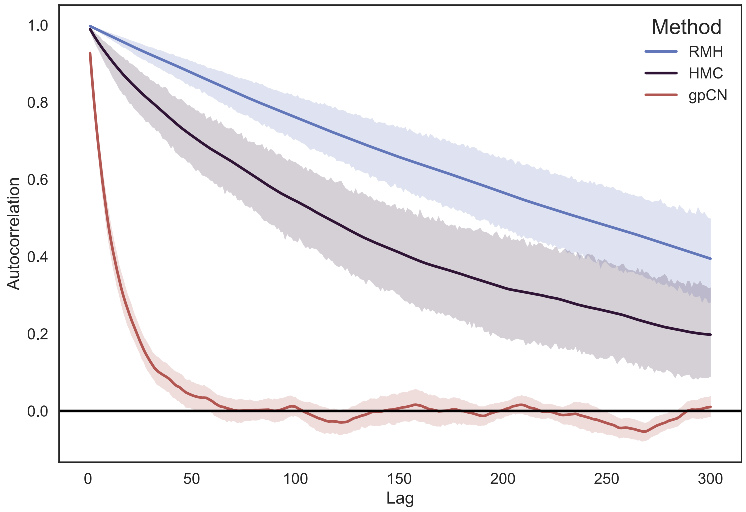


FIGURE 5.8: Autocorrelation plot of the MCMC chain of all three methods on Model B. The solid lines represent the mean autocorrelation among different variables, and the shadow represents the 95% confidence interval of the autocorrelation on different dimensions.

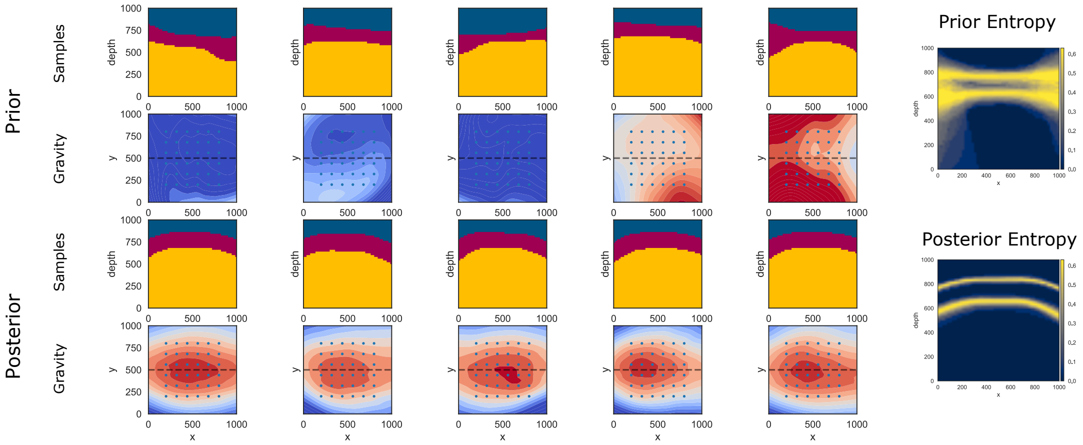


FIGURE 5.9: Left: Model instances sampled from prior and posterior with the corresponding forward gravity plotted below each instance. Forward simulations are evaluated on 50 by 50 grid. The data locations used in the inference are denoted as blue dots. The cross-sections used to plot the entropy are denoted by the black dashed lines; Right: the entropy plot of prior and posterior on cross-sections at the dashed line. Gravity is using the same color scale as the measurement in Figure 5.2

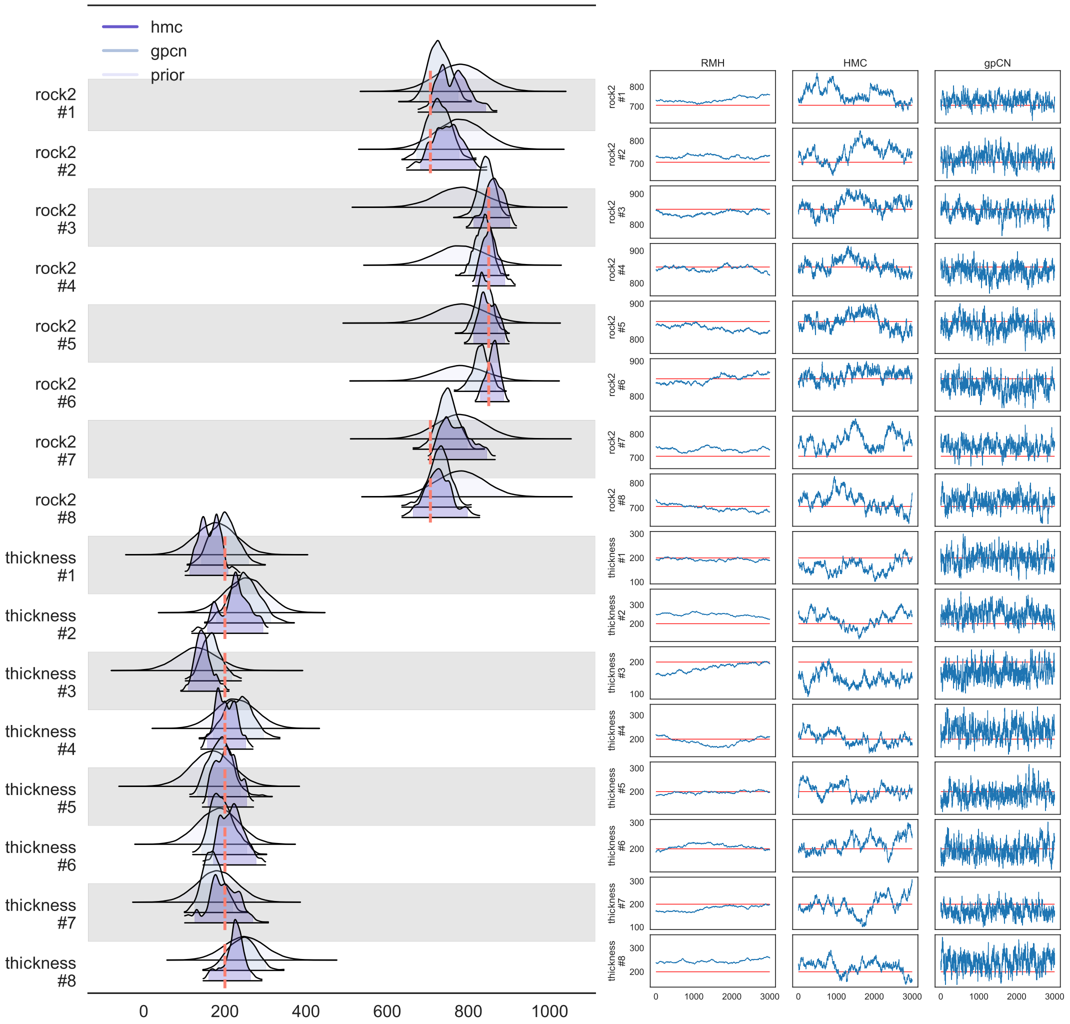


FIGURE 5.10: Empirical density by gpCN and HMC and the MCMC chains for model B. RMH acceptance rate: 93.8%, HMC with 3 leapfrog steps acceptance rate 99.8 %, gpCN acceptance rate 75.0 %. The RMH chain is not shown due to the poor mixing.

these drill holes is limited in shallow areas. The deepest available drill hole KV297 located at the center of the deposit only reaches 1366 m which left the lower boundaries with large uncertainty. The 3D geological model is developed based on the stratigraphy described in two geological cross-sections by along southwest–northeast (A) and southeast–northwest (B) directions which are interpreted based on seismic profiles Koivisto et al. (2015), and the 2D plot is evaluated on cross-section (A). The model is represented on the intrusion scale by simplifying following the same lithology category used by Güdük et al. (2021) with a different parameter configuration from the work of Güdük et al. (2021). The subsurface lithology is represented by three components: the overburden (OVB), the group of ultramafic pyroxenite (UPX), and the host rock and Paleoproterozoic basement as one unit. The OVB is modeled as an unconformity unit in the described implicit modeling method which increases the complexity of the model compared to previous two examples. Faults are not considered in this case for simplification. The modeling area is centered at the deposit with an extent 4500 m in North-South direction and 6300 m in the East-West direction. The density of each lithology has a large range of variation (Malehmir et al., 2011; Koivisto et al., 2015; Fournier, 2019), which makes the modeling using a single density for the whole entire simplified unit challenging to capture the reality. We use the estimated mean rock density in the literature for the forward simulation. We configure the surface points as a multivariate normal distribution with the mean as the interpretation and assigning with a large standard deviation of 100 to provide the model the room to transform. Different from the previous model, where we know the ground truth gravity, here we use the field measurement extrapolated at a regular grid receiver positions shown in Figure 5.11. Instead of absolute values used in the previous examples, we use the relative Bouguer anomaly. A wider likelihood function with a standard deviation of 1.9 is used to consider the simplification of the model and the complexity of the reality. A 10×10 grid of receivers is evaluated in the inference. In total 14 parameters including the z values of 5 interface points for OVB surface and 9 interface points for UPX surface are considered in the inversion problem.

As shown in Figure 5.12, the instances sampled from the prior can already capture some geometrical information about the gravity field as well as the ellipsoid geometry of the ore body. By constraining the model using the same inference methodology described above, the instances sampled from the posterior show a shallower OVB layer to the East of the ore body at cross-section (A). Although the forward gravity simulated from the simplified model shows a deviation from the measurement. The posterior models show a better fit to the gravity data. This is evident in the southeast region, central region, and west region. Due to the simplified model configuration, the ability to represent reality is limited, but one can already see how gravity data can be included to improve the geological model and reduce uncertainties. The 3D model of the mean posterior is shown in Figure 5.13.

Similar MCMC efficiency results are shown by comparing the Markov chains and autocorrelation plots of different inference methods (Figure 5.15 and Figure 5.14).This

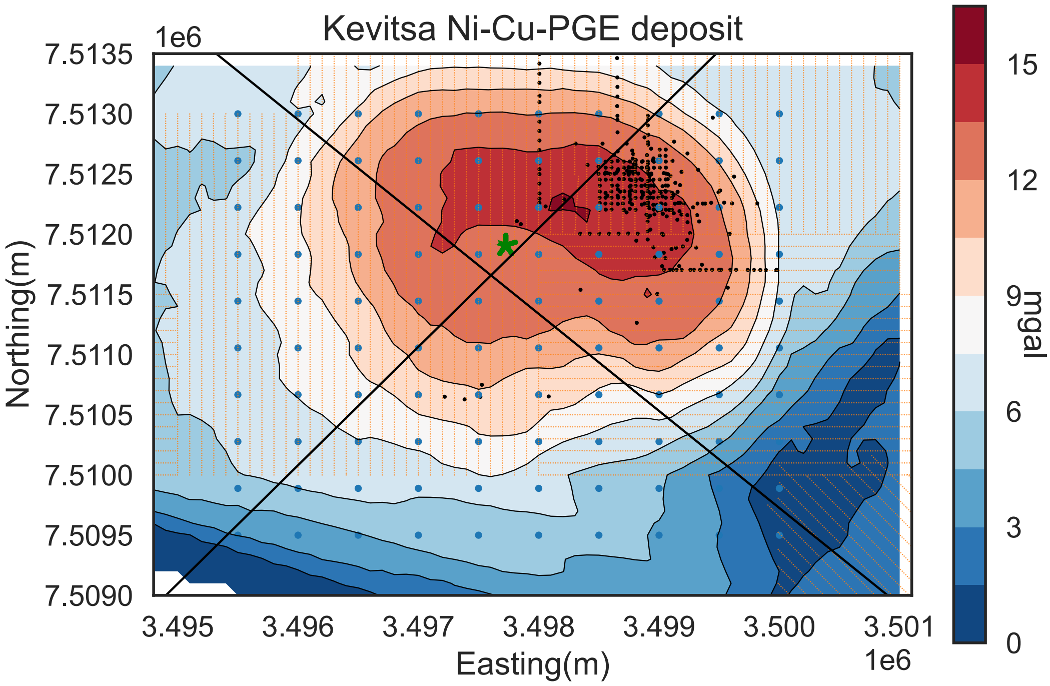


FIGURE 5.11: The gravity data in the modeling area. The geological data locations used in the inference are denoted as blue dots. The field gravity measurement locations are denoted as orange dots. The drill hole locations are denoted as black dots. The deepest drill hole KV297 is shown by the green star. Cross-sections are denoted by black lines.

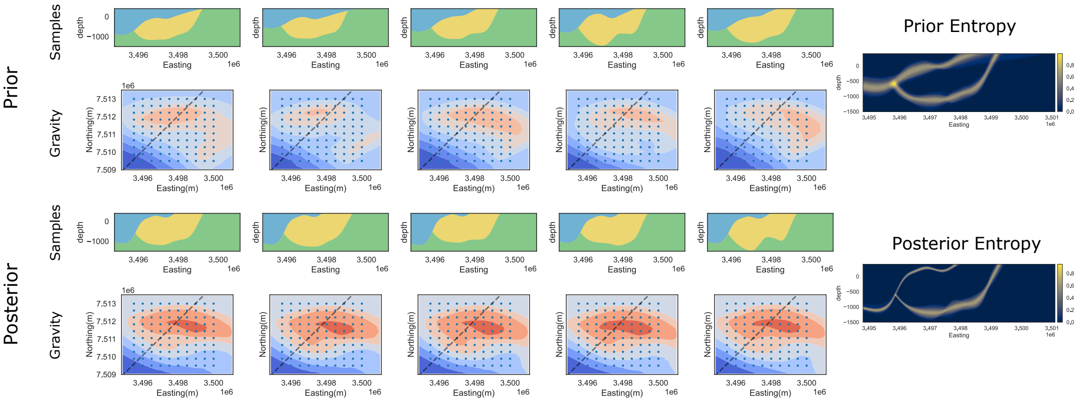


FIGURE 5.12: Left: Model instances of Kevitsa deposit sampled from prior and posterior with the corresponding forward gravity plotted below each instance. Forward simulations are evaluated on a 50 by 50 grid. The data locations used in the inference are denoted as blue dots. The cross-sections (A) used to plot the entropy are denoted by the black dashed lines; Right: the entropy plot of prior and posterior evaluated at cross-section (A). Gravity is using the same color scale as the measurement if Figure 5.11

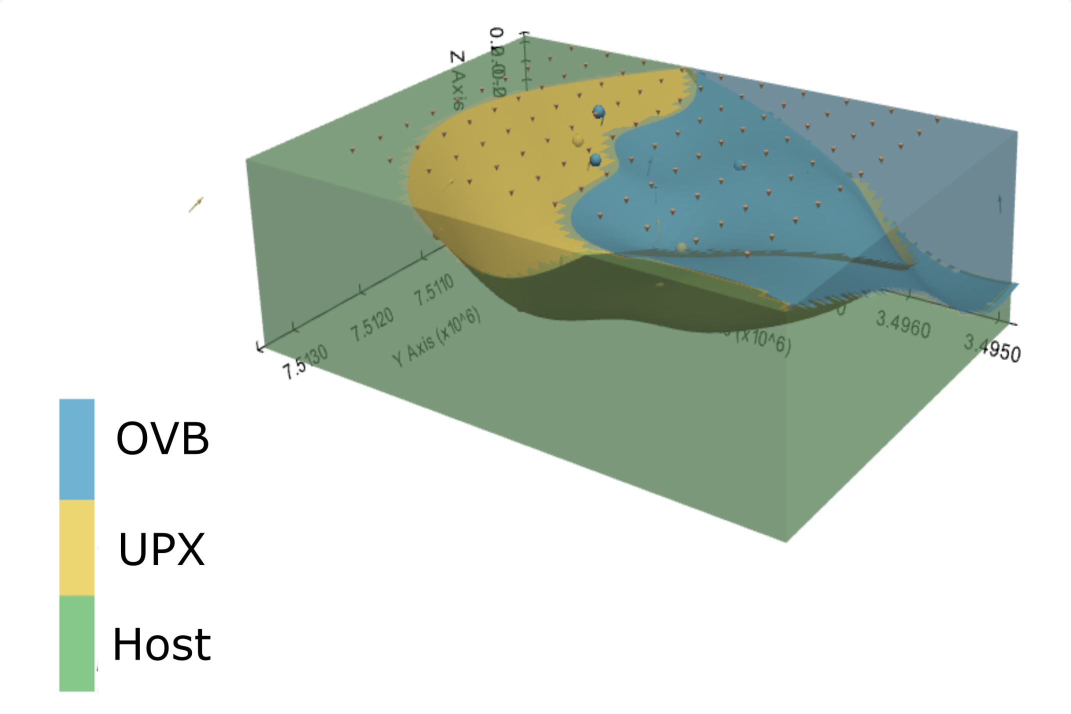


FIGURE 5.13: 3D Kevitsa model of the mean posterior.

example demonstrates that better-mixed chains of sampling can be obtained by gpCN while still preserving the computational efficiency in the meantime.

5.5 Discussion

5.5.1 Comparison to Conventional Gravity Inversion

The prior model used in this study is favorable to the inversion compared to conventional gravity inversion problems. We argue that, in the presence of prior knowledge of the structure as well as the geophysical data, constraining uncertainties by additional data is the more relevant goal in the framework of Bayesian inference. The model-based gravity inversion in the framework of Bayesian inference differs from the conventional gravity inversion in several ways. The implicit modeling method seeks a low-dimensional representation of the subsurface structure. Without a proper constrain a reasonable subsurface structure is difficult to be retrieved. In this work, we have experimented with the inversion under poor prior knowledge (in models A and B). We observe the impact of poor prior knowledge on the inversion results. However, in a realistic application (e.g., Model C in this work), one should consider configuring the prior information as close as possible to the knowledge of geological experts. We did not seek the maximum likelihood solution to allow the prior knowledge to be considered comprehensively. A small number of receivers are used in the inversion in this study to replicate cases where gravity data is sparse or correlated. The variation

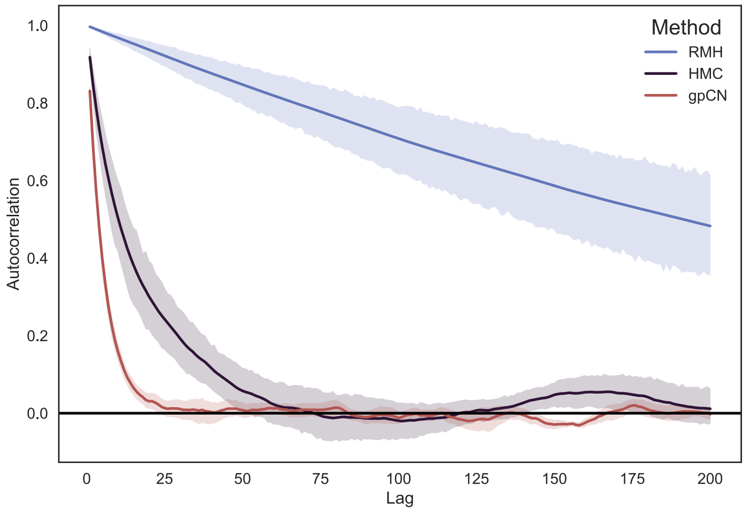


FIGURE 5.14: Autocorrelation plot of the MCMC chain of all three methods on Model C. The solid lines represent the mean autocorrelation among different variables, and the shadow represents the 95% confidence interval of the autocorrelation on different dimensions.

Models	Methods	Computation time for 3000 samples	n_{eff}	$AvgN_{eff}$ per 100s
model A	RMH	67s	8.5	12.5
	HMC 3	593s	14.0	2.3
	HMC 10	1824s	40.1	2.2
	gpCN	40s	113.9	284.8
model B	RMH	106.8	5.8	5.5
	HMC 3	2245.0	9.3	0.4
	gpCN	81.5	94.3	116.5
model C	RMH	209.0	7.5	3.6
	HMC 3	3349.4	108.8	3.2
	gpCN	187.5	261.5	139.4

TABLE 5.1: Computational efficiency of different methods. The number after HMC denotes the leapfrog steps used. The computational time for different models with the same method is related to the grid resolution, number of receivers, parameters dimensionality and number of leapfrog steps.

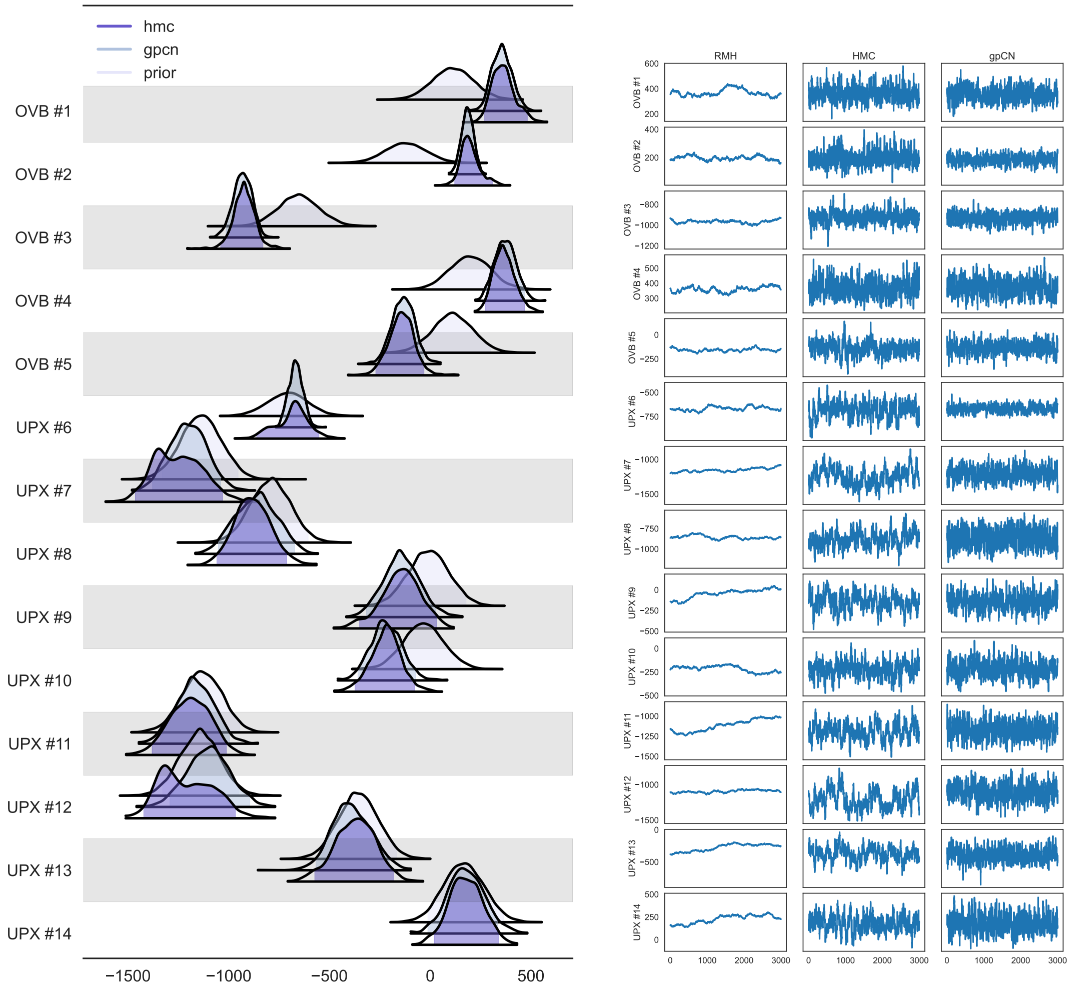


FIGURE 5.15: Empirical density by gpCN and HMC and the MCMC chains for model C. RMH acceptance rate: 76.2 %, HMC with 50 leapfrog steps acceptance rate 98.8 %, gpCN acceptance rate 80.0 %

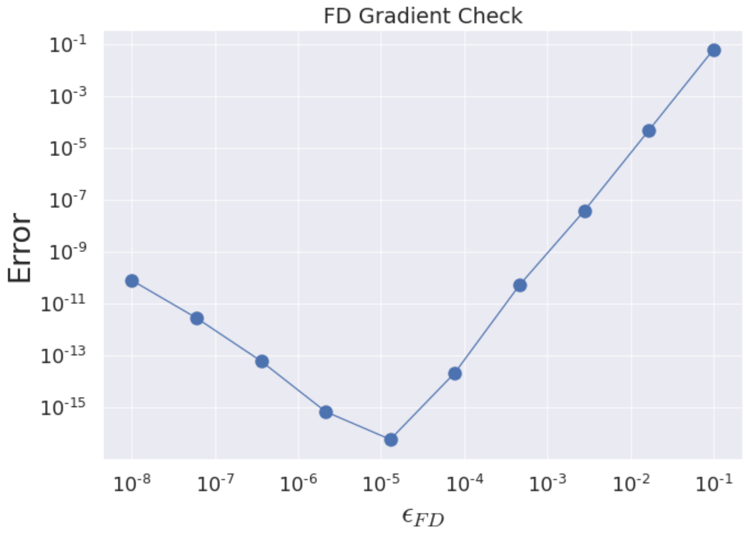


FIGURE 5.16: Finite difference gradient check

of likelihood plays a role in weighting the importance of prior knowledge and data in the final results.

5.5.2 Verification of Derivatives

To verify the correctness of AD, we compared the Jacobian and Hessian with the central finite difference scheme. We use the same configuration as model B with the exclusion of density parameters to alleviate the effect of sensitivity to the results. We illustrate the mean square error between AD and FD with different finite difference step size ϵ_{FD} in Figure 5.16 and Figure 5.17. Comparable results are obtained by AD and FD. The difference shown can be evidence of the typical pattern of truncation error, and roundoff error of numerical differentiation methods (Flannery et al., 1992; Seidl and Granzow, 2022). The larger error of Hessian evaluated by the FD method shows its difficulty of choosing ϵ_{FD} and insufficiency to preserve precision through the complex computational graph.

5.5.3 Computational Efficiency

The gpCN method we adopted in this study is a MAP-based method. These MAP-based methods utilize a global Hessian evaluated at the MAP point. According to our experiments, the main obstacle for HMC in probabilistic geomodeling is the computational cost for leapfrog steps. While each gradient evaluation is efficient through AD, the cumulative cost is detrimental to the efficiency of HMC. The primary advantage of gpCN compared with the gradient-based methods such as HMC is that the Hessian matrix is only required to be evaluated once. In addition to the adoption of pCN and the adapted proposal, gpCN outperforms RMH in high-dimension problems and a highly correlated posterior. Additionally, finding the MAP and starting the MCMC from the MAP saves the burn-in computation typically required in MCMC because the MAP is

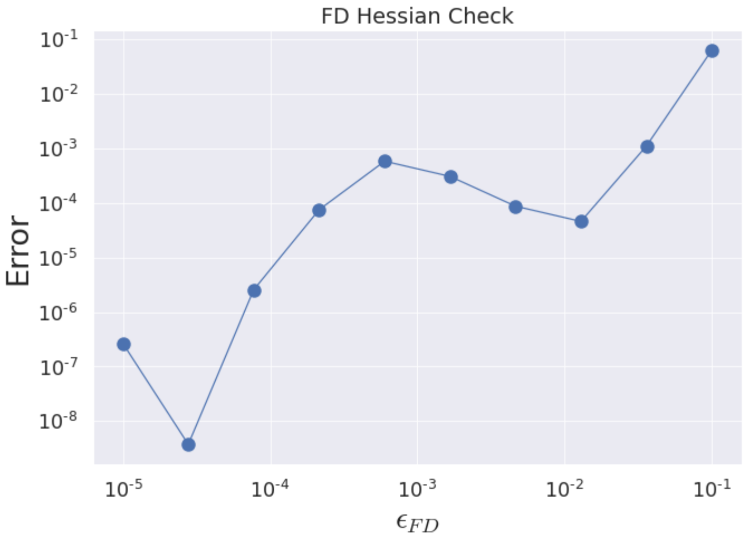


FIGURE 5.17: Finite difference Hessian check

already in the typical set. While finding the MAP seems essential for the success of the application of gpCN, it depends on the actual geometry of the posterior space in practice.

Our work is different from other works where the derivatives are accessible through an analytical solution. For example, Pankratov and Kuvshinov (2016) adopted the stochastic Newton MCMC method developed by Martin et al. (2012) to the application of electro-magnetic inversion. However, the work is similar to other successful adoption of Hessian information in the application of seismic inversion where the forward model has an analytical form, and therefore, the adjoint method can be applied for the Hessian evaluation. It utilizes the same concept of utilizing Hessian information to incorporate information on the measure of interest.

The high computational cost of the higher-order derivative is the main obstacle to applying advanced MCMC methods involving higher-order derivatives. The efficient computation of higher-order derivatives remains an open research problem (Laue et al., 2018; Nilsen et al., 2019; Margossian, 2019). In our work, the computation time required for a single Hessian calculation with the current implementation depends on the complexity of the computational graph used in AD and the number of parameters of interest. Although the full Hessian matrix evaluation is still more expensive than the gradient evaluation using a single backward propagation in AD, it is significantly faster than computing the Hessian by the FD method, regardless of the precision. The numerical analysis is conducted on model A, and the results are shown in Figure 5.18. Although AD has an $\mathcal{O}(M)$ complexity in theory, the overhead of memory due to the large matrix operations in the computational graph harms the computational time and deviates the resulting computation time away from a straight line. The numerical results of the central FD scheme match the theoretical $\mathcal{O}(M^2)$ complexity (Figure 5.18). A full Hessian matrix requires $\frac{M^2+M}{2} \times 4 \times t$ by using FD, where M represents the dimension of the parameter of interest, and t is the time cost for a single forward simulation.

In addition to the computational efficiency, the computational graph employed in AD also allows any variable in the graph to be traced with a derivative with minor modifications to the computer program.

5.5.4 Limitations

While the global Hessian method is efficient in high-dimensional problems and has the advantage of the single Hessian evaluation, it is likely to perform less optimally than methods with derivatives evaluated (once or several times) at each iteration (e.g. HMC) in a highly nonlinear posterior. Hessian evaluated at the MAP point are not the same representative in those scenarios, and therefore, leading to a poor exploration in regions far away from the MAP. In such cases, the method can be extended into the state-dependent local approximation with additional computational costs (Petra et al., 2014; Rudolf and Sprungk, 2018).

Another difficulty is raised by the multimodal posterior. While in many cases where a geological likelihood (e.g. Stamm et al., 2019; Wellmann et al., 2018; de la Varga and Wellmann, 2016), or a simple geophysical likelihood as described in this study is used, the posterior distribution is subject to a single modal distribution. The design of the likelihood function can lead to a multimodal posterior. Most MCMC methods will be trapped in isolated modes. Small stepsizes will localize the chain and prevent it from jumping between different modes, while larger steps will cause the acceptance rate to be extremely low. The ability of gradient-based methods (e.g., HMC) to explore the multimodal distribution depends on the connectivity of different modes. The proposed methods can be potentially combined with the existing framework to address multimodal posterior, for example parallel-tempered MCMC scheme (Robert et al., 2018; Scalzo et al., 2019) and adaptive MCMC (Pompe et al., 2020) or stacking independent chains (Yao et al., 2020). The proposed method is believed to explore each isolated mode efficiently compared to other MCMC methods.

5.6 Conclusion

In summary, this study extended the previous development of stochastic geological modeling methods and is the first attempt to exploit the higher-order derivatives to solve the probabilistic model-based gravity inversion. More specifically, we used the automatic differential technique implemented in TensorFlow to allow second-order derivative information to be efficiently evaluated in geological models. We applied the recently developed Hessian-informed MCMC, the generalized preconditioned Crank-Nicolson (gpCN), to solve the Bayesian inference problem on a synthetic three-layer geological model and a real-case model of the Kevitsa deposit. We compared gpCN

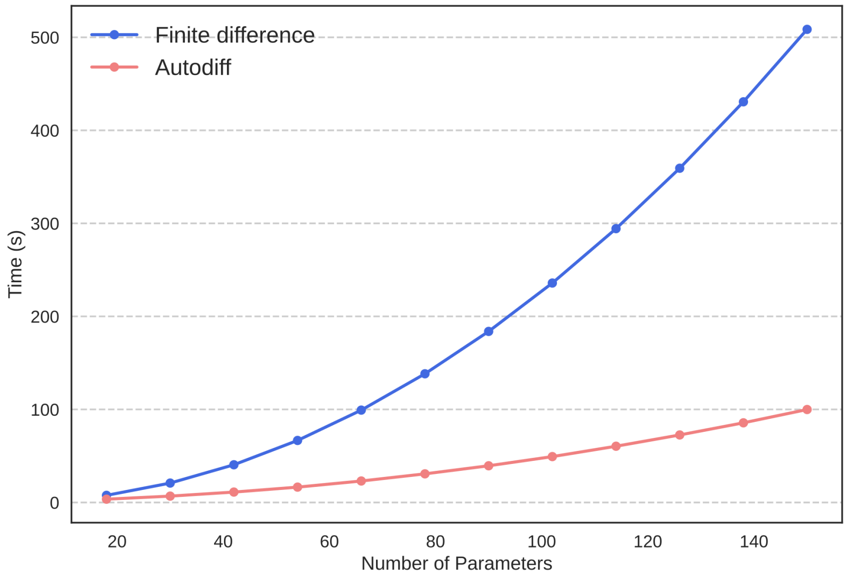


FIGURE 5.18: Comparison of computation time of a full Hessian matrix by Finite Difference(FD) method and Auto-Diff(AD). FD computation time is approximated by extrapolating the computation time for a single forward simulation. AD computation time is simulated with the same configuration used in the previous example. Experiments conducted on Tesla-P100 GPU

with the two commonly used MCMC methods, including RMH and the state-of-the-art HMC. The results demonstrate that with a single Hessian evaluation, gpCN outperforms RMH while preserving the computational efficiency, with no additional accumulated computational cost at each sampling step, which has the potential to be generalized to more complex models.

Chapter 6

Multimodality in Geological Modeling

6.1 Multimodality Problem

In the previous studies, we have introduced the smooth step-function to remove the 'artificial' multimodality in the posterior function of a probabilistic geological inference problem using gravity data in Chapter 4. However, this does not guarantee that the posterior is a unimodal modal distribution, and that the maximum a posteriori probability (MAP) point can always be found using gradient descent (GD). As previously discussed, the interpolation of gravity data is well-known to be ambiguous (Skeels, 1947) and methods have been developed over the last decades to constrain gravity inversion by adding regularization to the inverse problem (e.g., Reamer and Ferguson, 1989; Li and Oldenburg, 1998; Boulanger and Chouteau, 2001; Fournier, 2019). In the model-based geological inversion problem, the ambiguity emerged in two procedures: first, due to the use of a low-dimensional deterministic parametric representation of the geological model using an implicit modeling method, different parameter sets could construct similar or identical geological models; second, due to the intrinsic ambiguity of gravity interpolation, different geological structures could generate comparable gravity data. Hence, without a proper constraint, the model-based inverse problem is prone to ambiguities. If the discrepancy between the possible realizations is connected by transitional structures with lower possibility, the posterior distribution can exhibit multimodality.

In fact, the posterior could be highly multimodal due to the lack of identifiability of the parameters in the general Bayesian model (Stephens, 2000). In the Bayesian framework, the posterior probability is proportional to the product of the prior and likelihood function. The additional geophysical data and the a priori information are mutual constraints to each other. Hence both prior and likelihood both could introduce multimodality to the posterior distribution. If the prior distribution is defined by a multimodal distribution, the posterior is distinctly possible to be a multimodal distribution. A multimodal prior distribution is often used in the inversion problem involved with petrophysical data (e.g., Sun and Li, 2016; de Figueiredo et al., 2019).

In the case of the model-based gravity inversion, prior are often the structural geological data. Multimodal prior distribution might be used in cases where the density distribution is heterogeneous or the input structural data ambiguous. This prior distribution is, however, more often based on the modelers' choices. On the other hand, the likelihood function is controlled by the observational gravity data and the forward modeling and gravity simulation. Even with the prior distribution defined using unimodal distributions, the posterior could result in multimodality due to the low-dimensional parametric representation and the above-mentioned intrinsic ambiguity. This multimodal phenomenon in gravity inverse problems has been investigated in the previous studies (e.g., Scalzo et al., 2019, 2021).

Although one could use MCMC methods to characterize the statistical properties of the multimodal posterior distribution in theory, the majority of MCMC methods using localized proposal are tuned toward local approximate optimality (Gelman et al., 1997; Roberts and Rosenthal, 2001; Rudoy and Wolfe, 2006) and have been shown to mix slowly for multimodal distributions separated by low probability regions (Holmes et al., 2017; Aleardi et al., 2018; Scalzo et al., 2019). Methods utilizing gradient information tend to guide the chain toward local high probability area, and therefore opposite to the multimodality setting (Robert et al., 2018). For example, HMC is shown to have similar performance compared to RMH in multimodality settings (Mangoubi et al., 2018). Other methods combining local maxima information and MCMC are also inefficient for the multimodal problem, for example, the Hessian-informed MCMC (Chapter 5) and stochastic sampling using genetic algorithms and MCMC (Galley et al., 2020).

A simple alternative solution is sampling the posterior using multiple MCMC chains. However, in high-dimensional posterior space, the target distribution could be highly localized by separated modes leading to an inefficient sampling. Mixing sampling chains trapped in the local maxima is also challenging (Yao et al., 2020). Some MCMC methods specifically tuned for resolving the multimodal distribution include: Orthogonal MCMC methods run parallel chains using random-walk proposal and periodically exchange information between the chains to achieve the global exploration of multimodal distribution (Martino et al., 2016); The power tempering methods, including the simulated tempering algorithm, which uses a single Markov-chain, as well as its multi-chain variant parallel tempering (Geyer, 1991; Marinari and Parisi, 1992; Scalzo et al., 2019, 2021; Olierook et al., 2021), utilize an auxiliary temperature parameter to assist the sampling in a multimodality setting. However, the tempering method can be inefficient as the dimension increases (Robert et al., 2018).

Another branch of inference methods is variational inference. The variational inference method seeks an approximation to the target distribution within a predetermined family of probabilistic probability distributions by minimizing the Kullback-Leibler (KL) divergence. KL divergence is a measurement of the difference between two distributions (Kullback and Leibler, 1951). Various variational inference methods have been developed and mainly differ in the choice of those approximating families.

For example, mean-field variational inference (Bishop and Nasrabadi, 2006; Blei et al., 2017), or structured stochastic variational inference (Saul and Jordan, 1995; Hoffman and Blei, 2015). Recent development in variational methods seeks more generally applicable methods than the ones using simple families. For example, "black box" variational inference methods (Kingma and Welling, 2013; Ranganath et al., 2014), Kucukelbir et al. (2017) proposed the automatic variational inference method which uses a Gaussian variational family. Another group of methods which seeks a probability transformations has also been proposed (Rezende and Mohamed, 2015; Tran et al., 2015b; Marzouk et al., 2016). Liu and Wang (2016) proposed the Stein variational gradient descent (SVGD), which is based on a sequence of incremental transformations to minimize the KL divergence based on the gradient information. In addition to the scalability, SVGD has also the flexibility of resolving multimodal posterior distribution and adapting to different applications. Therefore, SVGD has been adopted in some recent studies of uncertainty quantification (Zhang and Curtis, 2020a,b; Zhu and Zabaras, 2018; Zhang et al., 2019).

In this study, we attempt to apply SVGD to explore the multimodal posterior in a geomodeling inversion problem. The multimodality introduced by the prior is relatively simple to analyze and control. Therefore, here we focus on the multimodality introduced by the likelihood function, and the proposed method is generally applicable to both scenarios. I will first review the SVGD, which closely follows the original description of SVGD in (Liu and Wang, 2016). Then I will show some preliminary results on geomodeling inversion applications.

6.2 Method: Stein Variational Gradient Descent (SVGD)

To introduce the SVGD algorithm, let us first recall the Bayes' theorem from Chapter 2:

$$p\left(\boldsymbol{m} \mid \boldsymbol{d}_{obs}\right)=\frac{p\left(\boldsymbol{d}_{obs} \mid \boldsymbol{m}\right) p(\boldsymbol{m})}{p\left(\mathbf{d}_{obs}\right)} \tag{6.1}$$

where $\boldsymbol{m}=[m_1, m_2, \ldots, m_M] \in \mathbb{R}^M$ is the set of parameters.

At the core of the variational methods, is to approximate the posterior $p\left(\boldsymbol{m} \mid \boldsymbol{d}_{obs}\right)$ with a family (set) of predefined distributions $\mathcal{Q}=\{q(\boldsymbol{m})\}$ by minimizing the KL divergence. The KL divergence is a measure of the difference between two distribution which is defined as (Kullback and Leibler, 1951):

$$\mathrm{KL}\left[q(\boldsymbol{m}) \| p\left(\boldsymbol{m} \mid \boldsymbol{d}_{obs}\right)\right]=\mathbb{E}_q[\log q(\boldsymbol{m})]-\mathbb{E}_q\left[\log p\left(\boldsymbol{m} \mid \boldsymbol{d}_{obs}\right)\right] \tag{6.2}$$

By combining Equation 6.1 and 6.2, the we get:

$$\mathrm{KL}\left[q(\boldsymbol{m}) \| p\left(\boldsymbol{m} \mid \boldsymbol{d}_{obs}\right)\right]=\mathrm{E}_q[\log q(\boldsymbol{m})]-\mathrm{E}_q\left[\log p\left(\boldsymbol{m}, \boldsymbol{d}_{obs}\right)\right]+\log p\left(\boldsymbol{d}_{obs}\right) \tag{6.3}$$

Liu and Wang (2016) proposed the SVGD algorithm to minimize the KL divergence by a sequence of smooth transformations $\mathcal{T}$. The transformation $\mathcal{T}$ takes the following form:

$$\mathcal{T}(\boldsymbol{m}) = \boldsymbol{m} + \epsilon\phi(\boldsymbol{m}) \tag{6.4}$$

where ϵ is a small step size, and $\phi(\boldsymbol{m}) \in \mathbb{R}^M$ is the direction the perturbation transformation takes.

Let's take the target distribution as $\hat{p}(\boldsymbol{m})$ supported on $\mathcal{X} \subseteq \mathbb{R}^M$. If we take a different probability distribution of $\hat{q}(\boldsymbol{m})$ also supported in $\mathcal{X}$, and its transformation by $\mathcal{T}$ as $q_{\mathcal{T}}(\boldsymbol{m})$, one can find the maximal decreases of the KL divergence by the gradient (see derivation in: Liu and Wang, 2016; Zhang and Curtis, 2020a):

$$\nabla_\epsilon \mathrm{KL}\left[\hat{q}_T \| \hat{p}\right]\big|_{\epsilon=0} = -\mathrm{E}_q\left[\operatorname{trace}\left(\mathcal{A}_p \phi(\boldsymbol{m})\right)\right] \tag{6.5}$$

where $\mathcal{A}_p$ is the Stein operator and has the form:

$$\mathcal{A}_p \boldsymbol{\phi}(\boldsymbol{m}) = \nabla_{\boldsymbol{m}} \log \hat{p}(\boldsymbol{m}) \boldsymbol{\phi}(\boldsymbol{m})^T + \nabla_{\boldsymbol{m}} \boldsymbol{\phi}(\boldsymbol{m}) \tag{6.6}$$

Liu and Wang (2016) proposed to use the kernelized Stein discrepancy to maximize the negative gradient of the KL divergence. The Stein discrepancy takes the following form:

$$\mathbb{D}(\hat{q}, \hat{p}) = \max_{\phi \in \mathcal{F}} \left\{\mathbb{E}_{\boldsymbol{m} \sim \hat{q}}\left[\operatorname{trace}\left(\mathcal{A}_p \phi(\boldsymbol{m})\right)\right]\right\} \tag{6.7}$$

where the proper choice of the function set $\mathcal{F}$ for ϕ is critical (Liu and Wang, 2016). The above optimization problem in Equation 6.7 can be solved in a simpler way in a closed form by maximizing ϕ in the unit ball of a reproducing kernel Hilbert space (RKHS) (Liu and Wang, 2016), where the Stein discrepancy becomes the Kernelized Stein discrepancy (KSD):

$$\mathbb{D}(\hat{q}, \hat{p}) = \max_{\phi \in \mathcal{H}^M} \left\{\mathbb{E}_{\boldsymbol{m} \sim \hat{q}}\left[\operatorname{trace}\left(\mathcal{A}_p \phi(\boldsymbol{m})\right)\right] \quad, \text{s.t.} \quad \|\phi\|_{\mathcal{H}^M} \leq 1\right\} \tag{6.8}$$

where $\mathcal{H}^M$ is the Hilbert space of M-dimensional vector functions. if we define a kernel function $k(x, y) =< \varphi(\boldsymbol{m}), \varphi(\boldsymbol{m}\prime) >_{\mathcal{H}}$, then, its RKHS $\mathcal{H}$ is defined as the closure of linear span $\{f : f(x) = \sum_{i=1}^m a_i k\left(x, x_i\right), \quad a_i \in \mathbb{R}, \quad m \in \mathbb{N}, \quad x_i \in \mathcal{X}\}$ with inner products $\langle f, h\rangle_{\mathcal{H}} = \sum_{ij} a_i b_j k\left(x_i, x_j\right)$ for $h(x) = \sum_i b_i k\left(x, x_i\right)$. The so-called reproducing property of RKHS is referred to as: $f(x) = \left\langle f\left(\boldsymbol{m}'\right), k\left(\boldsymbol{m}', \boldsymbol{m}\right)\right\rangle_{\mathcal{H}}$. The right-hand side has the optimal solution (Liu et al., 2016; Oates et al., 2017; Chwialkowski et al., 2016):

$$\phi(x) = \phi_{\hat{q},\hat{p}}^*(x) / \left\|\phi_{\hat{q},\hat{p}}^*\right\|_{\mathcal{H}^M} \tag{6.9}$$

where

$$\phi_{\hat{q},\hat{p}}^*(\boldsymbol{m}) = \mathbb{E}_{\boldsymbol{m}' \sim \hat{q}}\left[\mathcal{A}_p k(\boldsymbol{m}', \boldsymbol{m})\right] \tag{6.10}$$

and the KSD becomes:

$$\mathbb{D}(\hat{q}, \hat{p}) = \left\| \phi^*_{\hat{q},\hat{p}} \right\|_{\mathcal{H}^M} \tag{6.11}$$

Hence, the optimal perturbation direction ϕ that maximize the negative gradient in Equation 6.4 is $\phi^*_{\hat{q},\hat{p}}(x)$ and $\phi^*_{\hat{q},\hat{p}}(x)$ takes the form:

$$\phi^*_{\hat{q},\hat{p}}(\boldsymbol{m}) = \mathbb{E}_{\boldsymbol{m}' \sim \hat{q}} \left[\nabla'_{\boldsymbol{m}} \log \hat{p}(\boldsymbol{m}') k(\boldsymbol{m}\prime, \boldsymbol{m}) + \nabla'_{\boldsymbol{m}} k(\boldsymbol{m}', \boldsymbol{m}) \right] \tag{6.12}$$

The resulting Equation 6.12 shows that one can use a sequence of transformations $\mathcal{T}^*_\ell(\boldsymbol{m}) = \boldsymbol{m} + \epsilon_\ell \cdot \phi^*_{\hat{q}_\ell,\hat{p}}(\boldsymbol{m})$ to transform an initial set of particles taken from reference distribution $\hat{q}_0$ to the target $\hat{p}$, where ℓ is the number of iterations.

The Stein variational gradient descent algorithm is given below:

```
Initialize a set ($\mathcal{W}$ number) of particles $\{\boldsymbol{m}_i^0\}_{i=1}^{\mathcal{W}}$
while $\ell$ < Max number of iterations do
    $\boldsymbol{m}_i^{\ell+1} \leftarrow \boldsymbol{m}_i^\ell + \epsilon_\ell \hat{\boldsymbol{\phi}}^* \left(\boldsymbol{m}_i^\ell\right)$
    where $\hat{\phi}^*(\boldsymbol{m}) = \frac{1}{\mathcal{W}} \sum_{j=1}^{\mathcal{W}} \left[ k\left(\boldsymbol{m}_j^\ell, \boldsymbol{m}\right) \nabla_{\boldsymbol{m}_j^\ell} \log p\left(\boldsymbol{m}_j^\ell\right) + \nabla_{\boldsymbol{m}_j^\ell} k\left(\boldsymbol{m}_j^\ell, \boldsymbol{m}\right) \right]$,
    and $\epsilon_\ell$ is the step size at iteration $\ell$
end
```

Algorithm 5: STEIN VARIATIONAL GRADIENT DESCENT (Liu and Wang, 2016)

6.3 Results

6.3.1 Dummy 2D-Example

To demonstrate exploring a multimodal distribution using SVGD, we construct a dummy 2D Gaussian-mixture model. The three-mode mixture model with variable $\boldsymbol{m} = [m_1, m_2] \in \mathbb{R}^2$ (shown in Figure 6.1) is defined as :

$$p(\boldsymbol{m}) = \sum_{i=1}^{K} \zeta_i \frac{1}{\sqrt{(2\pi)^K |\boldsymbol{C}_i|}} \exp\left(-\frac{1}{2} (\boldsymbol{m} - \boldsymbol{\mu}_i)^{\mathrm{T}} \boldsymbol{C}_i^{-1} (\boldsymbol{m} - \boldsymbol{\mu}_i) \right) \tag{6.13}$$

where K is the total number of mode, ζ is the weights, μ is the mean and $\boldsymbol{C}$ is the covariance matrix.

We explore the distribution with RMH, HMC, and SVGD, and the results are shown below (gpCN is not included here as it's designed specifically for singal modal distribution). In both the MCMC methods, the sampling chains are trapped in a local mode with the chosen parameters (Figure 6.2). One could potentially use a larger step size in both methods and more leapfrog numbers in HMC to achieve a better posterior exploration, but the acceptance ratio will also be affected and might lead to a less efficient sampling. Based on the experiments, the MCMC chains can potentially jump

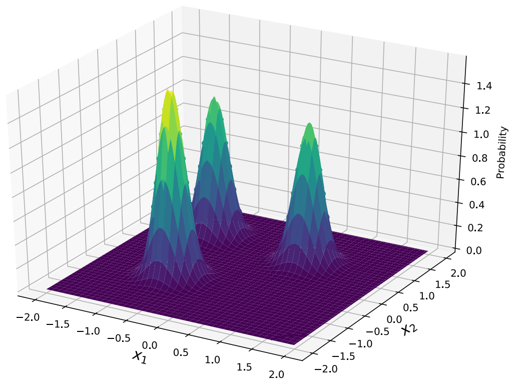


FIGURE 6.1: Probability of a multimodal Gaussian mixture model with $K = 3$, $[\zeta_1, \zeta_2, \zeta_3] = [0.3, 0.4, 0.3]$, $[\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \boldsymbol{\mu}_3] = [(0.7, 0.7), (-0.7, -0.7), (-1, 1)]$. $[\boldsymbol{C}_1, \boldsymbol{C}_2, \boldsymbol{C}_3] = [\mathrm{diag}(0.2, 0.2), \mathrm{diag}(0.2, 0.2), \mathrm{diag}(0.2, 0.2)]$.

between different modes but achieving a convergence takes a long time. This problem will be more significant when the dimension increases.

The SVGD methods randomly initialize 250 particles from a multivariate Gaussian distribution $\mathcal{N} \sim (\mu_0, \mathcal{C}_0)$, with $\mu_0 = (0, 0)$ and $\mathcal{C}_0 = \mathrm{diag}(1, 1)$. The particles move towards the probability high area and can achieve a relatively good representation of the target distribution within tens of steps with all modes recovered (shown in Figure 6.3). While the demonstrated example is simplified, it shows the ability of SVGD to deal with multimodal distributions, and we can expect the same performance in a real geological inverse problem.

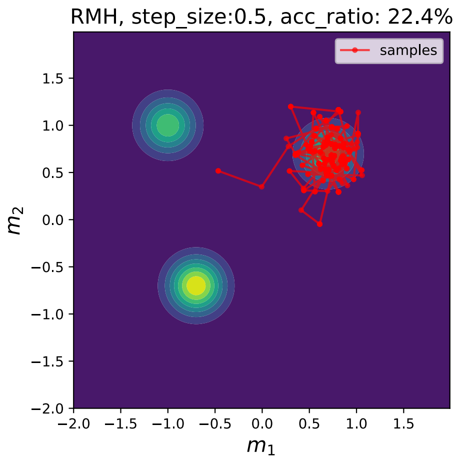
6.3.2 Application in Geological Modeling Inversion

To demonstrate the ability of SVGD to solve a multimodal distribution in a geological inversion, the geological model with a normal fault described in Section 4.3.3 is used to test the SVGD algorithm. The model configuration is shown in Figure 6.4. The smooth step-function introduced in Chapter 4 is applied. The ground truth model has two interfaces defined by 4 surface points and 2 orientation points, respectively. The normal fault is defined with 2 fault surface points 1 orientation point, all at the same elevation to construct a flat fault surface.

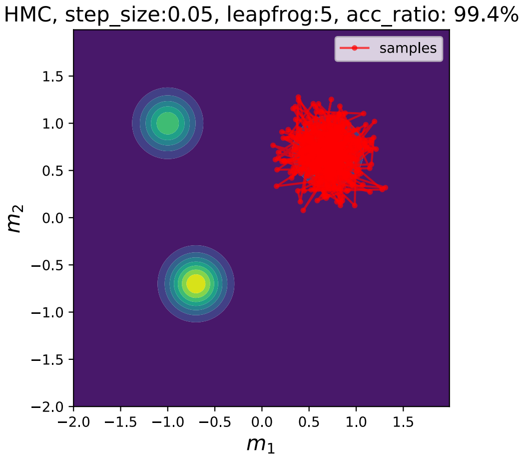
The variables $\boldsymbol{m}$ which we treat as probabilistic parameters contain the elevation z of the surface points, the dip angle of the fault ϖ and the density values for the three lithology units ρ:

$$\boldsymbol{m} = [z_1, z_1, \dots, z_8, \varpi, \rho_1, \rho_2, \rho_3] \tag{6.14}$$

To avoid negative values in density and preserve the normal fault (which means dip angle), the parameters are defined using constrained distributions which are transformed from the unbounded normal distributions. The transformation Υ follows the

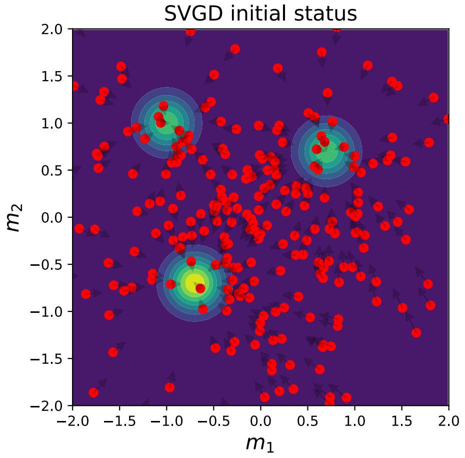


(a) RMH

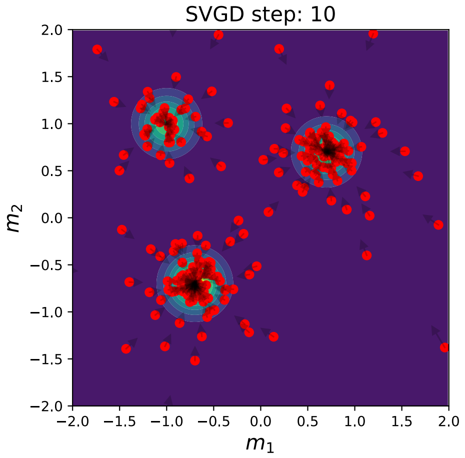


(b) HMC

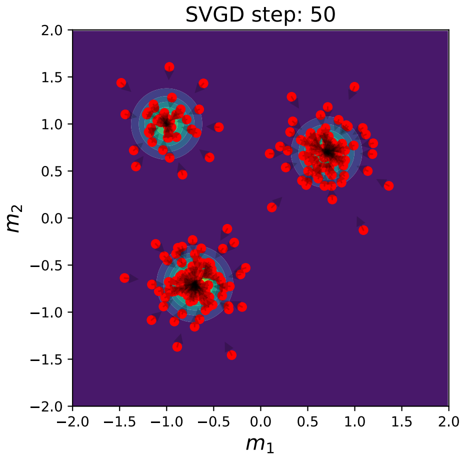
FIGURE 6.2: Sampling the dummy multimodal distribution with MCMC methods: RMH and HMC. 500 samples in both methods. MCMC methods could miss unseen mode complete with finite number of samples.



(a) step: 0



(b) step: 10



(c) step: 50

FIGURE 6.3: Inference result on the dummy multimodal distribution using SVGD with 250 particles showing at iteration 0, 10 and 50. Step size $\epsilon = 0.2$. Black arrows show the particle moving direction.

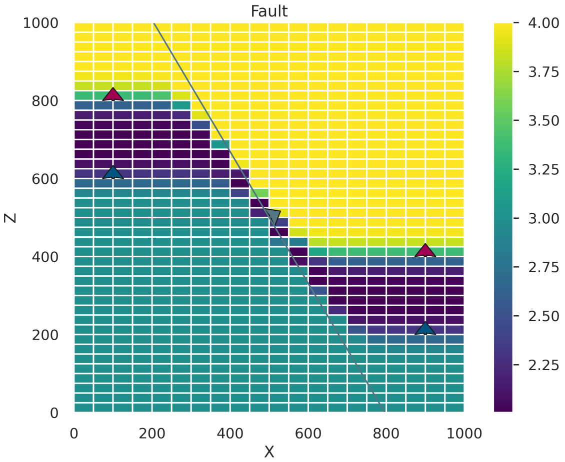


FIGURE 6.4: Cross-section of the density matrix of the fault model.

widely applied invertible logarithmic transform (Team et al., 2016; Zhang and Curtis, 2020a):

$$\theta_i = \Upsilon\left(m_i\right) = \log\left(m_i - lb_i\right) - \log\left(ub_i - m_i\right) \tag{6.15}$$

$$m_i = \Upsilon^{-1}\left(\theta_i\right) = lb_i + \frac{\left(ub_i - lb_i\right)}{1 + \exp\left(-\theta_i\right)} \tag{6.16}$$

where lb and ub are the lower bound and upper bound, θ is the unconstrained transformation of parameters.

The model configuration is summarized in Table 6.1, and the mean value of the variables are set to the true values, which are highlighted in boldface 6.1. A 3×3 grid of gravity receivers is set at the top of the model, and gravity is calculated using the gravity kernel method (Chapter 3). The likelihood function is defined as a multivariate normal distribution with covariance $\mathcal{C} = \mathrm{diag}[0.05, 0.05, \ldots, 0.05]$. The prior distribution is defined as a multivariate standard normal distribution. The bounds are set to $\pm 60\ \mathrm{m}$ for all the surface points, $\pm 20\ ^\circ$ for the fault dig angle and $\pm 0.6\ \mathrm{g/cm}^3$ for the densities. The constrained and unconstrained prior distribution is illustrated in Figure 6.5

We first use the GD method with random initial status sampled from the prior to find the MAP points. Each initial status converges to different local minima, which suggests the existence of the multimodality in the posterior distribution (shown in Figure 6.6). One can further confirm that the Hessian matrices at these MAP points are positive definite by using the method described in Chapter 5.

We initialized 500 particles in the prior distribution and applied the SVGD method to infer the posterior probability. The SVGD is run for 100 iterations with the learning rate ϵ set to 0.08. The result is shown in Figure 6.6.

We can observe that particles have shown a significantly larger movement in the

surface name	surface points			orientations			dip	density
	x	y	z	x	y	z		in g/cm^3
rock unit 1	200	200	**800**	100	500	800	0	**4**
	200	800	**800**					
	800	200	**400**	900	500	400	0	
	800	800	**400**					
rock unit 2	200	200	**600**	100	500	600	0	**2**
	200	800	**600**					
	800	200	**200**	900	500	200	0	
	800	800	**200**					
basement	\	\	\	\	\	\	\	**3**
fault surface	500	200	500	500	500	500	**60**	\
	500	800	500					

TABLE 6.1: Input data for the fault model. Probabilistic variables means are shown in boldface.

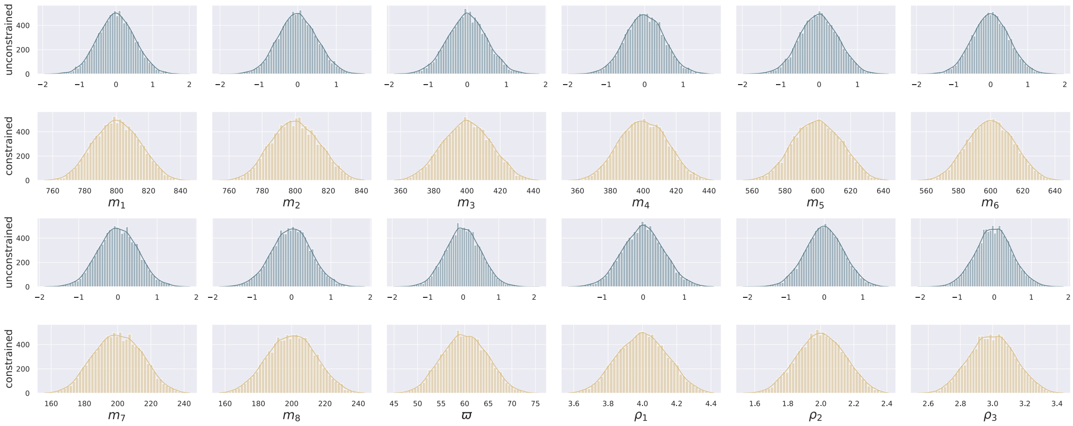


FIGURE 6.5: Illustration of the prior distribution in the original domain (in yellow) and transformed domain (in green). Each pair of the plot have the unconstrained plot above and the constrained plot below. The probability distribution is illustrated using Monte Carlo samples

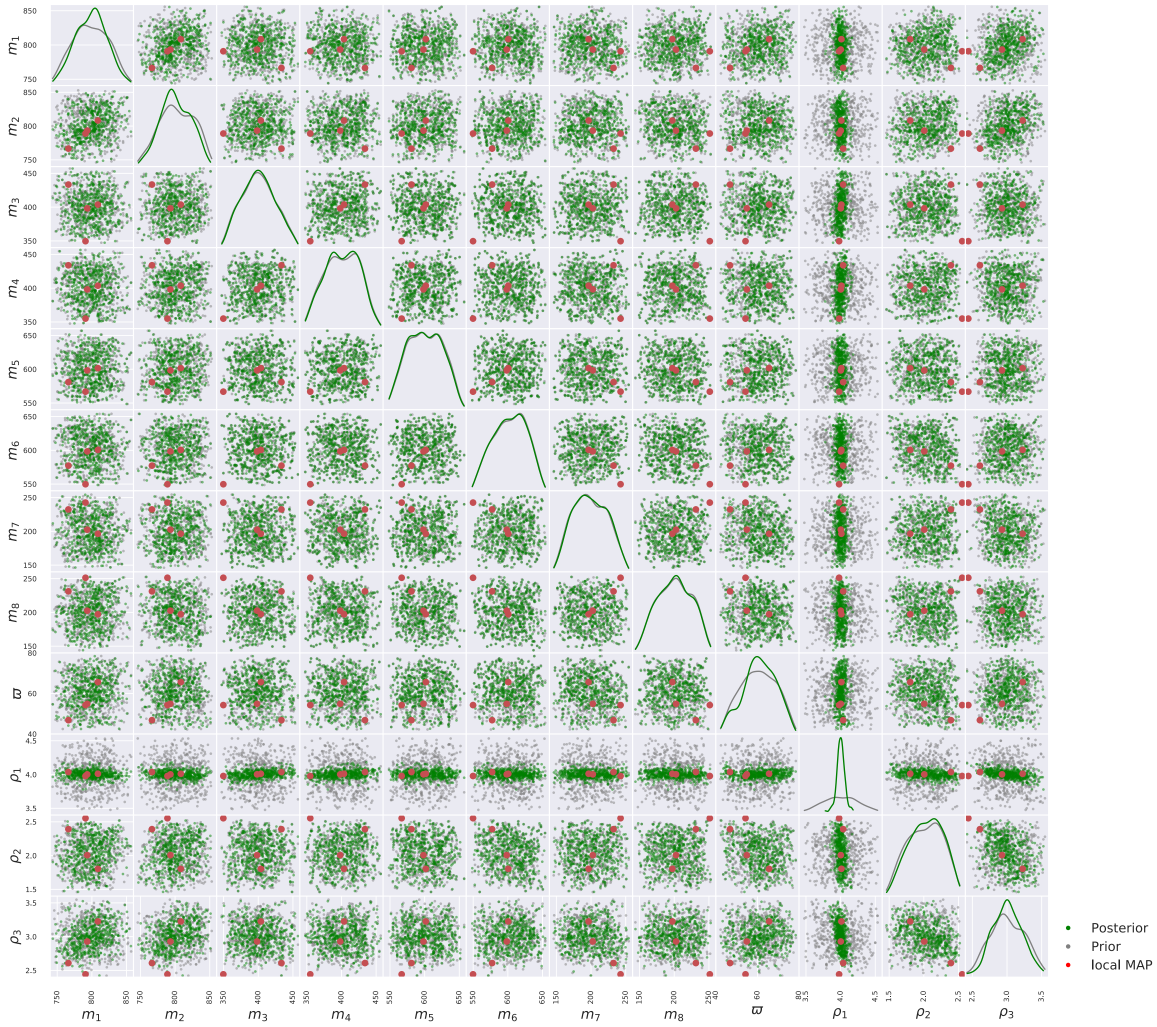


FIGURE 6.6: Corner plot of the prior and posterior distribution. The kernel density estimation is given in the diagonal plots. Red dots are the MAP points found by GD method.

dimension of the density value of the top layer ρ_1. This indicates that the posterior is most sensitive to the top layer density. The movement in the other dimensions is subtle in comparison. The individual local modes are not clearly seen in the posterior particles, but the MAP points found by GD all lie in the high posterior region in the ρ_1 dimension.

The above observations suggest that both GD and SVGD are largely impacted by the significant sensitivity difference in the parameters. This sensitivity difference might lead to slow convergence of both GD and SVGD, where most of the movement happens in the dominating direction. This might lead SVGD to require a large number of iterations to resolve the posterior on the other dimensions. In addition, only 500 particles are considered in this experiment. The number of particles directly links to the resolution of the estimated distribution. When only one particle is used, the SVGD algorithm degenerates to a GD (Liu and Wang, 2016). This might suggest more particles are needed to resolve a more detailed posterior in higher dimensions.

Another point worth noticing is the choice of prior and likelihood function uncertainty range. In this study, we have chosen to set the likelihood function with an extremely small uncertainty range to construct a multimodal scenario for research. This leads to the likelihood function dominating the posterior probability, and the prior has a weak constraint to the inversion. Without a proper constraint of the prior, the posterior is dominantly controlled by the likelihood function in high-dimensional space and could have complex structures. This complexity might emerge due to the relatively low-dimensional parametric representation of the geological model (as described in Chapter 2). One should also consider if the existing modes make sense. For example, some of the MAP points found in this study have a lower probability compared to others, is this mode important to the result is an open question and needs further analysis. Overall, to clearly explain the complexity of the posterior morphology and the application of SVGD to solve the multimodal posterior problem in geological modeling, further research is needed. One might find a model more balanced sensitivity parameter configuration and more clear multimodality to better demonstrate the use of SVGD.

6.4 Conclusion

In conclusion, this chapter proposed using the SVGD algorithm to solve the Bayesian inference problem in geological modeling using gravity as the likelihood. The presented experiment was designed to solve the multimodal posterior distribution problem, which is challenging for conventional MCMC methods. The preliminary result indicates that SVGD can be successfully applied to explore the posterior space to some extent, but the ability to resolve the multimodality is unclear. A large number of iterations and a better-constraint posterior might be needed. This study is the first attempt to adopt a gradient-based variational inference method to solve the model-based geological inversion. The presented work intended to provide guidance to future research in this direction.

Chapter 7

Discussion and Outlook

In the presented work, I have demonstrated an end-to-end trainable geological inversion framework. The framework mainly contains of two parts, forward modeling and inversion. The two fundamental elements of forward modeling are the implicit geological modeling and gravity forward simulation. In the inversion part, I focus on uncertainty quantification with the Bayesian inference method. Both the forward and inverse parts play important roles in the success of the introduced framework. In this chapter, I will first present some discussions from both the forward modeling perspective and the inversion perspective. Some recommendations for future research will also be given from my own perspective based on the conducted experiments.

7.1 Forward Modeling

The introduced forward modeling contains two chaining parts, geological modeling and gravity simulation, and it is end-to-end trainable, meaning that continuous derivatives can be evaluated through the model. However, the introduced UQ framework and the inference methods introduced to the geological inversion problem are generally applicable. The main obstacle to the application using other geological modeling approaches and other geophysical simulations is the difficulties and efficiency in the evaluation of derivatives. In this study, we have shown an AD framework to evaluate the derivatives and shown that AD is more efficient and accurate compared to the numerical method - FD (Chapter 5). However, we have also shown that by just blindly applying AD, a meaningful derivative is not guaranteed to be used for training purposes. The current workflow is demonstrated with gravity field forward simulation, but it can also be extended to other potential-field methods, for example, magnetic field measurements (Güdük et al., 2021). However, it is still limited in the potential-field methods due to the simplicity of the forward simulation. The combination of geological modeling with other geophysical simulations (e.g., seismic tomography, geothermic, electrical resistivity tomography, etc.) is extremely interesting and important for the application of geological modeling and uncertainty quantification study to a broader set of problems. The potential link to these PDE-based problems is through the adjoint. However, the PDE problems often require a redefined mesh and are solved by numerical methods using FDM/FEM, which is difficult to be integrated into the current workflow. Attempts have been made to bypass the numerical solution by using computationally cheaper surrogate models, for example, Physics-informed neural networks (PINN) (Raissi et al., 2019; Liang et al., 2021; Degen et al., 2022a), a Gaussian process emulation (Zhao and Kowalski, 2022). However, the link to structural geological modeling is still unclear and it would be interesting to address this approach in future studies.

The geological modeling method we adopted in this study is the co-kriging method. While the co-kriging method has many advantages (see Chapter 2), the current implementation of co-kriging has limitations in representing, for example, finite faults, strike-slip faults, complex objects, and non-stationary models. A possible solution for the issue of strike-slip faults and non-stationary models might be using an additional external drift function similar to the universal drift described in Chapter 2. However, future research is still needed to allow a wider application of the proposed framework.

The computational graph (in Figure 4.1) is an analogue to an ANN. One of the advantages of ANN using AD technique is that batched data can be passed into the model to fully take advantage of the GPU architecture (Paszke et al., 2017). However, this is limited in the current geological modeling framework implemented using the AD techniques and accelerated using GPU. This limitation prevents parallel MCMC chains from being executed on a non-distributed scheme and the application in SVGD. A batched implementation would be significantly beneficial for the SVGD methods, where the particles can be evaluated simultaneously. The implementation issue associated with memory usage will be discussed in the following section.

7.2 Bayesian Inference, Uncertainty Quantification

In this study, the Bayesian inference framework is adopted to simultaneously consider the prior knowledge of the input parameters and additional gravity data. Hence both the likelihood and the prior are important to the Bayesian inference.

Prior

The parameters considered in this study are mainly the position (x, y, and z) of the surface points, the dipping angle of the fault, and the density values for the lithologies. The proposed framework can be easily extended to other parameters in the geological model, for example, the position, azimuth, polarity (which controls the scalar field changing rate), and other parameters in co-kriging (range, nuggets effect, covariance), and also the parameters in gravity calculation. However, the current implementation still requires some manual configuration of the inference model but is also more flexible. Considering more parameters in the inference is possible but also makes the inversion more challenging. Also, the number of input data can vary. For example, one can find infinite numbers of points on the same flat surface to produce the same structural geological model. The increasing number of parameters increases the number of possible combinations of parameter sets and therefore requires better constraints by either geophysical data or prior knowledge. The increasing number of parameters also reduces the model's ability to learn as it is more constrained. For example, the surface structure is more and more constrained with an increasing number of surface points, and it will end up defining the surface without any interpolation in between the data.

Therefore, it is crucial for the modeler to decide which parameters should and should not be taken into the inversion. For example, considering the x y coordinates in a model with only Horizontal surfaces, varying the x and y coordinates have no impact on the resulting structure, and so is the gravity simulation. This will add meaningless dimensions to the parameter space and cost more computational resources. However, deciding the parameters to be included is non-trivial. One possible solution is to analyze the eigenvalues in the Hessian matrix, which gives the sensitivity of the parameters.

Likelihood function

For the likelihood function, this study focused on gravity data. This is mainly due to the simplicity of the gravity calculation and widely available data. The same concept for choosing the prior parameters applies to the choices of data points taking into the inversion. New data could be expensive to collect and taking the full collection of data could introduce unnecessary correlations and increase the computational cost. One might consider the use of optimal experimental design to better select the included data (Aretz-Nellesen et al., 2021; Wu et al., 2022).

Considering other types of likelihoods, including other types of geophysical simulation in the current framework, is challenging due to the difficulties in evaluating derivatives, as discussed above. Another issue linked to the use of additional likelihoods is the choice of the definition of the likelihood function. In this study, the gravity likelihood is defined as a multivariate normal distribution. However, other types of distributions are used in the gravity likelihood function. For example, Wellmann et al. (2018) used the squared error likelihood with a half Cauchy distribution, (Scalzo et al., 2021) defined each likelihood function as a t-distribution and calculated the joint probability. All these likelihood definitions are essentially equivalent to the exact choice of distributions. The choice of the uncertainty range of the likelihood is mentioned in Chapter 3, which should not be decided only based on the uncertainty in the observational data as in the conventional gravity inversion problems. In the model-based inversion, the misfit in the simulation result and the observational data comes from both the noise data, the simulation approach (e.g., aliased effect in Chapter 2), and the simplification of the model. The latter arguably has more impact on the uncertainty of the likelihood function. Other types of likelihood functions are possible to be included in the framework (see Chapter 2). Here, I would specifically mention the topological constraints. Schaaf et al. (2021) proposed to use topology information in the Bayesian inference and the method has recently been applied to analyse 3D model of complex fold-and-thrust belts (Brisson et al., 2023). Topology provides an important constraint so that the meaningful structure of the geological model can be preserved. However, the method is limited to derivatives-free inference problems due to the discretized topology matrix. One could potentially adopt the smooth step function (introduced in Chapter 4) to generate a smooth version of the topology matrix and allow the derivative-informed inference method to be applied.

7.3 Implementation

The forward geological modeling and gravity simulation is implemented in TensorFlow (Abadi et al., 2015) as an AD tool, but the implementation is not limited to the choice of software packages. Here, I would like to discuss some of the bottlenecks in the current implementation.

Software Packages

TensorFlow, as one of the most popular ML frameworks, has been actively developed since the recent surge of ML applications. The choice of TensorFlow is due to its good readability, good community support and scalability, GPU support, and parallelization ability (Abadi et al., 2015)

The current implementation of the forward modeling in TensorFlow is not optimal. This is mainly presented in two aspects. The first aspect is the memory efficiency in the current implementation. The second aspect is the implementation of the derivative implementation with the TensorFlow Autograph function.

In this study, we have demonstrated the inverse problem with observational gravity data. We have developed the kernel method, which takes advantage of the implicit modeling methods and constructs meshes only around the receivers' locations to reduce memory usage in the case of sparse observational data. However, there are still many cases where gravity data is abundant. Whether these data are informative to the inversion or not is an open question, which we will discuss in the following section. Modeling all the data itself is a challenge in the current implementation. On the other hand, in the conventional convolution scheme implemented in an iterative manner, the sequential computation is time-consuming (see Chapter 3). By using the vectorized computation, for example, the kernel methods in Chapter 3 consumes a large amount of memory and limits the number of data points we can consider. One reason for this demanding memory consumption emerged from the inflexible tensor manipulation in TensorFlow. A tensor is a data type that contains a multi-dimensional array in TensorFlow and does not support value assignment. The alternative data type, TensorFlow Variable, also does not support assigned values partially (by the time of writing, with TensorFlow version 2.10.0). This is mainly required in the geological modeling parts where we need to dynamically vary the matrix. The current implementation bypasses this difficulty by using concatenation operators. However, the concatenation operators seem to allocate new memories for the newly constructed tensor, and due to the use of a computational graph, the old tensor is not cleared. The matrix contains the dimension of the resolution of the mesh and could scale with the number of receivers depending on the gravity method used. Therefore it demands a large amount of memory, especially on a GPU. This problem becomes more significant with the second-order derivatives. This is less of an issue for other software, such as Theano (Bastien et al., 2012) and Pytorch (Paszke et al., 2017) , where the tensor is more flexible. However, other unknown issues might emerge. Alternative methods are currently being explored, and a flexible framework that can be adapted to different software packages is currently being developed.

Some potential solutions for the current implementation issues with Tensorflow are discussed here. The first is to use an alternative operation to concatenation in TensorFlow or to use other software. The second is to use an octree mesh. However, this still has difficulties in adopting the stochastic framework, as the geometry varies at each step, and the mesh would need to be refined at each iteration. Third is using TPU (Tensor processing unit), which is hardware developed by Google specifically for tensor processing (Abadi et al., 2016). Accordingly the preliminary experiments on TPU, the TPU overcomes the memory issue by using more efficient parallelization and could achieve more efficient calculations.

Computational Time

The second limitation in the current implementation in TensorFlow is the use of the Autograph function. As discussed in Chapter 4, AD constructs a computational graph to track the adjoint of each operator in the computation in the forward path to allow for efficient derivative calculation. This graph construction process is called 'tracing'. Then, the constructed graph is revisited in each iteration without construction, and this process is called 'executing'. 'tracing' takes significantly more computational resources than 'executing'. Therefore an efficient framework requires a stationary computational graph to be executed. In TensorFlow, this is called graph mode execution. On the opposite, one has the option to execute the code with 'eager execution', where the computational graph is not stationary and will be constructed at each iteration. This is referred to as the eager mode. The Autograph function in TensorFlow converts the program automatically in the graph code and allows the graph mode execution. This is done by using the decoration function 'tf.function'. The use of decoration is still unclear and under experiment. I found that a nested style of decoration (decorating each inner function) could achieve a roughly 10% boost in performance. However, in the current implementation, the decoration is limited only to first-order derivatives. The reason

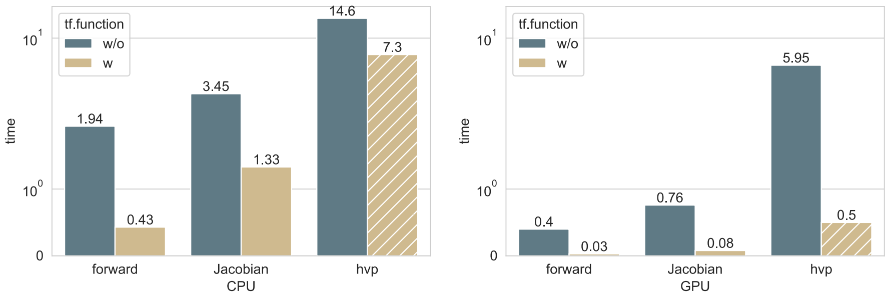


FIGURE 7.1: Comparison of computation time (in seconds) for different functions - Forward modeling and gravity (forward), Jacobian and Hessian vector product (hvp), on GPU and CPU, and with (w) and without (w/o) TensorFlow autograph decoration. The computation time plotted in striped pattern is an estimated value based on the ratio of efficiency gain of Jacobian computation.

for the failure in the application of the Hessian calculation is still unknown. This problem only occurs when a specific control flow is used in TensorFlow. This is probably due to the limited functionality in the implemented forward-mode AD. Further investigation is needed. Here, we compare the performance of graph model execution on the forward function, the first-order derivative (Jacobian with back-propagation), and the second-order derivatives (Hessian vector production, using the method described in Chapter 5).

The computation time is compared based on the Greenstone model with four receivers and the low-resolution kernel method described in Chapter 3. The result is shown in Figure 7.1. The same code was run with Intel Xeon CPU and Tesla K80 GPU on Google Colab. The forward simulation, Jacobian and Hessian-vector product (hvp, described in Chaprter 5) are conducted with and without the Autograph technique. We can see that the static graph generated greatly improved the efficiency in all the computations. The acceleration is greater for the GPU, which is consistent with the advantage in large tensor computation in GPU. The hvp with Autograph failed in the experiment with both CPU and GPU. Based on the improvement in the computation of the forward function and the Jacobian, the predicted computation time is given at $\sim \times 2$ in CPU and $\sim \times 12$ in GPU. Hence, successful implementation of a static computational graph would greatly improve the Hessian computation and provide the potential for the application of inference method with step-wise Hessian calculation, for example, the Stein Variation Newton method (Detommaso et al., 2018).

All the points discussed above are important for the application of the stochastic geological modeling. Future research is needed to address these issues.

Chapter 8

Conclusion

This thesis presents an end-to-end trainable geological modeling and uncertainty quantification framework that allows advanced derivative-informed Bayesian inference methods to be efficiently applied. The thesis presents a leap forward from current Bayesian practices, with the ability to explore large parameter spaces orders of magnitude faster than traditional MCMC based methods. This speed computationally enables the use of Bayesian methods in real life applications that often have larger parameter spaces (and where it would be computationally infeasible to use trational methods). Is is an important step forward for the application of automated uncertainty quantification in geological modeling.

The framework contains three essential components, namely the forward geological modeling using the implicit cokriging method; the gravity simulation in the implicit modeling framework; and efficient Bayesian inference approaches utilizing the liklihood's derivatives to accelerate the posterior exploration based on the trainable modeling approach. The contributions to these three components are detailed below.

Chapter 3 introduced a novel approach for gravity simulation for the implicit modeling method. In the framework of stochastic geological modeling, gravity data serves as an additional constraint to the model construction. The gravity data can be included in the modeling process as the likelihood function in a probabilistic joint inversion framework and allows the quantification of uncertainty in geological modeling directly. A fast but also precise forward gravity simulation is essential to the success of the probabilistic inversion. The proposed kernel method is based on the widely adopted analytical solution on a discretized grid. As opposed to a globally refined regular mesh, we construct local tensor grids for individual gravity receivers, respecting the gravimeter locations and the local sensitivities. I have demonstrated that the kernel method is efficient in terms of both computing and memory use for mesh-free implicit geological modeling approaches. This design makes the method well-suited for many-query applications like Bayesian machine learning using gradient information calculated from Automatic Differentiation (AD). Optimal grid design without knowing the underlying geometry is not straightforward before evaluating the model. Therefore, a novel perspective on a refinement strategy for the kernel method based on the sensitivity of the cell to the corresponding receiver is developed. Numerical results are presented and show superior performance compared to the conventional spatial convolution method.

Chapter 4 investigated the posterior landscape and introduced a method using step-functions to allow continuous derivatives to be evaluated in the geological model. The presented work bridged the gap between the application of geological modeling using a AD framework and gravity simulation. The introduced method is the foundation of the successful application of the derivative-informed Bayesian inference method in the model-based gravity inversion. The introduced dimension-reduction visualization method provides an intuitive way to analyze the posterior probability distribution, which is often in high dimensions. It provides a qualitative indication of the trainability of a geological inverse problem.

To achieve a more efficient posterior exploration, advances in MCMC methods utilize derivative information. Hence, Chapter 5 has introduced an approach to efficiently evaluate second-order derivatives in geological modeling based on the introduced trainable approach and adopt a Hessian-informed MCMC method, the generalized preconditioned Crank-Nicolson (gpCN), as a tool to solve the 3D model-based gravity Bayesian inversion problem. The result is compared with two other widely applied MCMC methods, random walk Metropolis-Hasting and Hamiltonian Monte Carlo, on a synthetic geological model and a realistic structural model of the Kevitsa deposit. The experiment demonstrates that superior performance is achieved by the gpCN compared to the other two state-of-the-art sampling methods. This indicates the potential of the proposed method to be generalized to more complex models.

Preliminary results of the experiment using SVGD to solve the model-based gravity inversion problem is also shown. The results of a successful application of SVGD to the uncertainty quantification, but the ability to resolve a multimodal posterior distribution is not clear. This indicates improvements and further research are required in the future.

This thesis demonstrated novel methods to address the uncertainty quantification problems in 3D geological modeling and geophysical simulation. It is a fundamental step towards the wider application of advanced Bayesian inference methods for geomodel-based inversions in various areas of geosciences.

List of Figures

List of Tables

Bibliography

Abad, A. and Lacruz, E. (2013). Computing derivatives of a gravity potential by using automatic differentiation. *Celestial Mechanics and Dynamical Astronomy*, 117(2):187–200.

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al. (2016). Tensorflow: A system for large-scale machine learning. In *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)*, pages 265–283.

Abu-Khzam, F. N., Collins, R. L., Fellows, M. R., Langston, M. A., Suters, W. H., and Symons, C. T. (2017). Kernelization algorithms for the vertex cover problem.

Agarwal, A. (2019). Static automatic batching in tensorflow. In *International Conference on Machine Learning*, pages 92–101. PMLR.

Aleardi, M., Ciabarri, F., and Gukov, T. (2018). A two-step inversion approach for seismic-reservoir characterization and a comparison with a single-loop markov-chain monte carlo algorithmts and sl inversion algorithms. *Geophysics*, 83(3):R227–R244.

Andrieu, C. and Thoms, J. (2008). A tutorial on adaptive mcmc. *Statistics and computing*, 18(4):343–373.

Aretz-Nellesen, N., Chen, P., Grepl, M. A., and Veroy, K. (2021). A sequential sensor selection strategy for hyper-parameterized linear bayesian inverse problems. In *Numerical Mathematics and Advanced Applications ENUMATH 2019*, pages 489–497. Springer.

Astic, T., Fournier, D., and Oldenburg, D. W. (2020). Joint inversion of potential-fields data over the do-27 kimberlite pipe using a gaussian mixture model prior. *Interpretation*, 8(4):SS47–SS62.

Bagby, T., Rao, K., and Sim, K. C. (2018). Efficient implementation of recurrent neural network transducer in tensorflow. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 506–512. IEEE.

Barton, D. C. (1933). Mechanics of formation of salt domes with special reference to gulf coast salt domes of texas and louisiana. *AAPG Bulletin*, 17(9):1025–1083.

Bastien, F., Lamblin, P., Pascanu, R., Bergstra, J., Goodfellow, I., Bergeron, A., Bouchard, N., Warde-Farley, D., and Bengio, Y. (2012). Theano: new features and speed improvements. *arXiv preprint arXiv:1211.5590*.

Baydin, A. G., Pearlmutter, B. A., Radul, A. A., and Siskind, J. M. (2018). Automatic differentiation in machine learning: a survey. *Journal of machine learning research*, 18.

Beskos, A., Pillai, N., Roberts, G., Sanz-Serna, J.-M., and Stuart, A. (2013). Optimal tuning of the hybrid monte carlo algorithm. *Bernoulli*, 19(5A):1501–1534.

Betancourt, M. (2017). A conceptual introduction to hamiltonian monte carlo. *arXiv preprint arXiv:1701.02434*.

Betancourt, M. (2018). A geometric theory of higher-order automatic differentiation. *arXiv preprint arXiv:1812.11592*.

Betancourt, M. (2019). The convergence of markov chain monte carlo methods: from the metropolis method to hamiltonian monte carlo. *Annalen der Physik*, 531(3):1700214.

Bishop, C. M. and Nasrabadi, N. M. (2006). *Pattern recognition and machine learning*, volume 4. Springer.

Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. (2017). Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877.

Bond, C. E. (2015). Uncertainty in structural interpretation: Lessons to be learnt. *Journal of Structural Geology*, 74:185–200.

Boot-Handford, M. E., Abanades, J. C., Anthony, E. J., Blunt, M. J., Brandani, S., Mac Dowell, N., Fernández, J. R., Ferrari, M.-C., Gross, R., Hallett, J. P., et al. (2014). Carbon capture and storage update. *Energy & Environmental Science*, 7(1):130–189.

Boulanger, O. and Chouteau, M. (2001). Constraints in 3d gravity inversion. *Geophysical prospecting*, 49(2):265–280.

Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., and Zhang, Q. (2018). JAX: composable transformations of Python+NumPy programs.

Brisson, S., Wellmann, F., Chudalla, N., von Harten, J., and von Hagke, C. (2023). Estimating uncertainties in 3-d models of complex fold-and-thrust belts: A case study of the eastern alps triangle zone. *Applied Computing and Geosciences*, 18:100115.

Calcagno, P., Chilès, J.-P., Courrioux, G., and Guillen, A. (2008). Geological modelling from field data and geological knowledge: Part i. modelling method coupling 3d potential-field interpolation and geological rules. *Physics of the Earth and Planetary Interiors*, 171(1-4):147–157.

Capponi, M., Mansi, A. H., and Sampietro, D. (2018). Improving the computation of the gravitational terrain effect close to ground stations in the gte software. *Studia Geophysica et Geodaetica*, 62(2):206–222.

Carr, J. C., Beatson, R. K., Cherrie, J. B., Mitchell, T. J., Fright, W. R., McCallum, B. C., and Evans, T. R. (2001). Reconstruction and representation of 3d objects with radial basis functions. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 67–76.

Carranza, E. J. (2011). From predictive mapping of mineral prospectivity to quantitative estimation of number of undiscovered prospects. *Resource Geology*, 61(1):30–51.

Caumon, G. (2010). Towards stochastic time-varying geological modeling. *Mathematical Geosciences*, 42(5):555–569.

Caumon, G., Collon-Drouaillet, P., De Veslud, C. L. C., Viseur, S., and Sausse, J. (2009). Surface-based 3d modeling of geological structures. *Mathematical Geosciences*, 41(8):927–945.

Caumon, G., Gray, G., Antoine, C., and Titeux, M.-O. (2012). Three-dimensional implicit stratigraphic model building from remote sensing data on tetrahedral meshes: Theory and application to a regional model of la popa basin, ne mexico. *IEEE Transactions on Geoscience and Remote Sensing*, 51(3):1613–1621.

Caumon, G., Lepage, F., Sword, C. H., and Mallet, J.-L. (2004). Building and editing a sealed geological model. *Mathematical Geology*, 36(4):405–424.

Chasseriau, P. and Chouteau, M. (2003). 3d gravity inversion using a model of parameter covariance. *Journal of applied geophysics*, 52(1):59–74.

Chauhan, M. S., Pierri, I., Sen, M. K., and Fedi, M. (2021). Assessing model uncertainty for the scaling function inversion of potential fields. *Geophysics*, 86(6):G89–G98.

Chen, L. and Liu, L. (2019). Fast and accurate forward modelling of gravity field using prismatic grids. *Geophysical Journal International*, 216(2):1062–1071.

Chen, S. F. (2005). *Geology of the Atley, Rays Rocks, and Southern Sandstone 1: 100 000 Sheets*. Geological Survey of Western Australia.

Chen, T., Fox, E., and Guestrin, C. (2014). Stochastic gradient hamiltonian monte carlo. In *International conference on machine learning*, pages 1683–1691.

Chiles, J.-P. and Delfiner, P. (2009). *Geostatistics: modeling spatial uncertainty*, volume 497. John Wiley & Sons.

Chwialkowski, K., Strathmann, H., and Gretton, A. (2016). A kernel test of goodness of fit. In *International conference on machine learning*, pages 2606–2615. PMLR.

Cockett, R., Kang, S., Heagy, L. J., Pidlisecky, A., and Oldenburg, D. W. (2015). Simpeg: An open source framework for simulation and gradient based parameter estimation in geophysical applications. *Computers & Geosciences*, 85:142–154.

Commer, M. (2011). Three-dimensional gravity modelling and focusing inversion using rectangular meshes. *Geophysical Prospecting*, 59(Modelling Methods for Geophysical Imaging: Trends and Perspectives):966–979.

Corliss, G. F. (1988). Application of differentiation arithmetic, volume 19 of perspectives in computing.

Cotter, S. L., Roberts, G. O., Stuart, A. M., and White, D. (2013). Mcmc methods for functions: modifying old algorithms to make them faster. *Statistical Science*, pages 424–446.

Cowles, M. K. and Carlin, B. P. (1996). Markov chain monte carlo convergence diagnostics: a comparative review. *Journal of the American Statistical Association*, 91(434):883–904.

Cox, L. H., Wilson, G. A., and Zhdanov, M. S. (2010). 3d inversion of airborne electromagnetic data using a moving footprint. *Exploration Geophysics*, 41(4):250–259.

Cui, T., Law, K. J., and Marzouk, Y. M. (2016). Dimension-independent likelihood-informed mcmc. *Journal of Computational Physics*, 304:109–137.

Culshaw, M. (2005). From concept towards reality: developing the attributed 3d geological model of the shallow subsurface. *Quarterly Journal of Engineering Geology and Hydrogeology*, 38(3):231–284.

Čuma, M., Wilson, G. A., and Zhdanov, M. S. (2012). Large-scale 3d inversion of potential field data. *Geophysical Prospecting*, 60(6):1186–1199.

Davies, R. S., Groves, D. I., Trench, A., Dentith, M., and Sykes, J. P. (2020). Appraisal of the usgs three-part mineral resource assessment through estimation of the orogenic gold endowment of the sandstone greenstone belt, yilgarn craton, western australia. *Mineralium Deposita*, 55(5):1009–1028.

Davies, R. S., Groves, D. I., Trench, A., Sykes, J., and Standing, J. G. (2018). Entering an immature exploration search space: Assessment of the potential orogenic gold endowment of the sandstone greenstone belt, yilgarn craton, by application of zipf's law and comparison with the adjacent agnew goldfield. *Ore Geology Reviews*, 94:326–350.

de Figueiredo, L. P., Grana, D., Roisenberg, M., and Rodrigues, B. B. (2019). Multimodal markov chain monte carlo method for nonlinear petrophysical seismic inversion. *Geophysics*, 84(5):M1–M13.

de la Varga, M., Schaaf, A., and Wellmann, F. (2019). Gempy 1.0: open-source stochastic geological modeling and inversion. *Geoscientific Model Development*.

de la Varga, M., Wellmann, F., and Murdie, R. (2015). Adding geological knowledge to improve uncertain geological models: a bayesian perspective. *Geotectonic Research*, 97(1):18–20.

de la Varga, M. and Wellmann, J. F. (2016). Structural geologic modeling as an inference problem: A bayesian perspective. *Interpretation*, 4(3):SM1–SM16.

Degen, D., Cacace, M., and Wellmann, F. (2022a). 3d multi-physics uncertainty quantification using physics-based machine learning.

Degen, D., Veroy, K., and Wellmann, F. (2022b). Uncertainty quantification for basin-scale geothermal conduction models. *Scientific reports*, 12(1):1–10.

Detommaso, G., Cui, T., Marzouk, Y., Spantini, A., and Scheichl, R. (2018). A stein variational newton method. *Advances in Neural Information Processing Systems*, 31.

Dominy, S. C., Noppé, M. A., and Annels, A. E. (2002). Errors and uncertainty in mineral resource and ore reserve estimation: The importance of getting it right. *Exploration and Mining Geology*, 11(1-4):77–98.

Duane, S., Kennedy, A. D., Pendleton, B. J., and Roweth, D. (1987). Hybrid monte carlo. *Physics letters B*, 195(2):216–222.

Enciu, P., Wurtz, F., Gerbaud, L., and Delinchant, B. (2009). Automatic differentiation for electromagnetic models used in optimization. *COMPEL-The international journal for computation and mathematics in electrical and electronic engineering*.

Evans, M. and Swartz, T. (2000). Approximating integrals via monte carlo and deterministic methods.

Farquharson, C. and Mosher, C. (2009). Three-dimensional modelling of gravity data using finite differences. *Journal of Applied Geophysics*, 68(3):417–422.

Fedi, M. and Rapolla, A. (1999). 3-d inversion of gravity and magnetic data with depth resolution. *Geophysics*, 64(2):452–460.

Fichtner, A., Zunino, A., and Gebraad, L. (2019). Hamiltonian monte carlo solution of tomographic inverse problems. *Geophysical Journal International*, 216(2):1344–1363.

Flannery, B. P., Press, W. H., Teukolsky, S. A., and Vetterling, W. (1992). Numerical recipes in c. *Press Syndicate of the University of Cambridge, New York*, 24(78):186–189.

Fournier, D. (2019). *Advanced potential field data inversion with lp-norm regularization*. PhD thesis, University of British Columbia.

Frank, T., Tertois, A.-L., and Mallet, J.-L. (2007). 3d-reconstruction of complex geological interfaces from irregularly distributed and noisy point data. *Computers & Geosciences*, 33(7):932–943.

Frankle, J. and Carbin, M. (2018). The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv preprint arXiv:1803.03635*.

Fuster, D., Bagué, A., Boeck, T., Le Moyne, L., Leboissetier, A., Popinet, S., Ray, P., Scardovelli, R., and Zaleski, S. (2009). Simulation of primary atomization with an octree adaptive mesh refinement and vof method. *International Journal of Multiphase Flow*, 35(6):550–565.

Galley, C. G., Lelièvre, P. G., and Farquharson, C. G. (2020). Geophysical inversion for 3d contact surface geometry. *Geophysics*, 85(6):K27–K45.

Gelman, A., Gilks, W. R., and Roberts, G. O. (1997). Weak convergence and optimal scaling of random walk metropolis algorithms. *The annals of applied probability*, 7(1):110–120.

Gelman, A. and Rubin, D. B. (1992). Inference from iterative simulation using multiple sequences. *Statistical science*, 7(4):457–472.

Geweke, J. and Tanizaki, H. (1999). On markov chain monte carlo methods for nonlinear and non-gaussian state-space models. *Communications in Statistics-Simulation and Computation*, 28(4):867–894.

Geyer, C. J. (1991). Markov chain monte carlo maximum likelihood.

Giraud, J., Lindsay, M., Ogarko, V., Jessell, M., Martin, R., and Pakyuz-Charrier, E. (2019a). Integration of geoscientific uncertainty into geophysical inversion by means of local gradient regularization. *Solid Earth*, 10(1):193–210.

Giraud, J., Ogarko, V., Lindsay, M., Pakyuz-Charrier, E., Jessell, M., and Martin, R. (2019b). Sensitivity of constrained joint inversions to geological and petrophysical input data uncertainties with posterior geological analysis. *Geophysical Journal International*, 218(1):666–688.

Girolami, M. and Calderhead, B. (2011). Riemann manifold langevin and hamiltonian monte carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(2):123–214.

Gonçalves, Í. G., Kumaira, S., and Guadagnin, F. (2017). A machine learning approach to the potential-field method for implicit modeling of geological structures. *Computers & Geosciences*, 103:173–182.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning*. MIT press.

Goodfellow, I. J., Vinyals, O., and Saxe, A. M. (2014). Qualitatively characterizing neural network optimization problems. *arXiv preprint arXiv:1412.6544*.

Grasemann, B., Martel, S., and Passchier, C. (2005). Reverse and normal drag along a fault. *Journal of Structural Geology*, 27(6):999–1010.

Green, P. J. (1995). Reversible jump markov chain monte carlo computation and bayesian model determination. *Biometrika*, 82(4):711–732.

Griewank, A. (2003). A mathematical view of automatic differentiation. *Acta Numerica*, 12:321–398.

Grose, L., Ailleres, L., Laurent, G., and Jessell, M. (2021). Loopstructural 1.0: time-aware geological modelling. *Geoscientific Model Development*, 14(6):3915–3937.

Güdük, N., de la Varga, M., Kaukolinna, J., and Wellmann, F. (2021). Model-based probabilistic inversion using magnetic data: A case study on the kevitsa deposit. *Geosciences*, 11(4):150.

Guglielmetti, L., Comina, C., Abdelfettah, Y., Schill, E., and Mandrone, G. (2013). Integration of 3d geological modeling and gravity surveys for geothermal prospection in an alpine region. *Tectonophysics*, 608:1025–1036.

Haber, E. and Schwarzbach, C. (2014). Parallel inversion of large-scale airborne time-domain electromagnetic data with multiple octree meshes. *Inverse Problems*, 30(5):055011.

Hale, D. and Emanuel, J. (2003). Seismic interpretation using global image segmentation. In *SEG Technical Program Expanded Abstracts 2003*, pages 2410–2413. Society of Exploration Geophysicists.

Hastings, W. K. (1970). Monte carlo sampling methods using markov chains and their applications.

Hillier, M., Wellmann, F., Brodaric, B., de Kemp, E., and Schetselaar, E. (2021). Three-dimensional structural geological modeling using graph neural networks. *Mathematical Geosciences*, 53(8):1725–1749.

Hillier, M. J., Schetselaar, E. M., de Kemp, E. A., and Perron, G. (2014). Three-dimensional modelling of geological surfaces using generalized interpolation with radial basis functions. *Mathematical Geosciences*, 46(8):931–953.

Hinze, W. J., Von Frese, R. R., Von Frese, R., and Saad, A. H. (2013). *Gravity and magnetic exploration: Principles, practices, and applications*. Cambridge University Press.

Hoffman, M. D. and Blei, D. M. (2015). Structured stochastic variational inference. In *Artificial Intelligence and Statistics*, pages 361–369.

Holmes, C., Krzysztof, L., and Pompe, E. (2017). Adaptive mcmc for multimodal distributions. Technical report, Technical report.

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.

Høyer, A.-S., Klint, K., Fiandaca, G., Maurya, P., Christiansen, A., Balbarini, N., Bjerg, P., Hansen, T., and Møller, I. (2019). Development of a high-resolution 3d geological model for landfill leachate risk assessment. *Engineering Geology*, 249:45–59.

Im, D. J., Tao, M., and Branson, K. (2016). An empirical analysis of the optimization of deep network loss surfaces. *arXiv preprint arXiv:1612.04010*.

Isaac, T., Petra, N., Stadler, G., and Ghattas, O. (2015). Scalable and efficient algorithms for the propagation of uncertainty from data through inference to prediction for large-scale problems, with application to flow of the antarctic ice sheet. *Journal of Computational Physics*, 296:348–368.

Izmailov, P., Vikram, S., Hoffman, M. D., and Wilson, A. G. (2021). What are bayesian neural network posteriors really like? *arXiv preprint arXiv:2104.14421*.

Jahandari, H. and Farquharson, C. G. (2013). Forward modeling of gravity data using finite-volume and finite-element methods on unstructured grids. *Geophysics*, 78(3):G69–G80.

Jerrell, M. E. (1997). Interval arithmetic for input-output models with inexact data. *Computational Economics*, 10(1):89–100.

Jessell, M. (2001). Three-dimensional geological modelling of potential-field data. *Computers & Geosciences*, 27(4):455–465.

Jessell, M., Aillères, L., De Kemp, E., Lindsay, M., Wellmann, J. F., Hillier, M., Laurent, G., Carmichael, T., and Martin, R. (2014). Next generation three-dimensional geologic modeling and inversion. *Society of Economic Geologists Special Publication*, 18(18):261–272.

Jessell, M., Pakyuz-Charrier, E., Lindsay, M., Giraud, J., de Kemp, E., Arribas, A., and Mauk, J. (2018). Assessing and mitigating uncertainty in three-dimensional geologic models in contrasting geologic scenarios. *Metals, Minerals, and Society*, 21:63–74.

Jessell, M., Valenta, R., Jung, G., Cull, J., and Geiro, A. (1993). Structural geophysics. *Exploration Geophysics*, 24(4):599–602.

Jessell, M. W. (1981). Noddy: an interactive map creation package. *unpublished MSc thesis, University of London*.

Jessell, M. W., Ailleres, L., and De Kemp, E. A. (2010). Towards an integrated inversion of geoscientific data: What price of geology? *Tectonophysics*, 490(3-4):294–306.

Jessell, M. W. and Valenta, R. K. (1996). Structural geophysics: integrated structural and geophysical modelling. 15:303–324.

Jones, R. R., McCaffrey, K. J., Wilson, R. W., and Holdsworth, R. E. (2004). Digital field data acquisition: towards increased quantification of uncertainty during geological mapping. *Geological Society, London, Special Publications*, 239(1):43–56.

Kerkering, J. C. (2003). Subjective and objective bayesian statistics: Principles, models, and applications.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kingma, D. P. and Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Klitzke, P., Sippel, J., Faleide, J. I., and Scheck-Wenderoth, M. (2016). A 3d gravity and thermal model for the barents sea and kara sea. *Tectonophysics*, 684:131–147.

Koivisto, E., Malehmir, A., Hellqvist, N., Voipio, T., and Wijns, C. (2015). Building a 3d model of lithological contacts and near-mine structures in the kevitsa mining and exploration site, northern finland: constraints from 2d and 3d reflection seismic data. *Geophysical Prospecting*, 63(4-Hard Rock Seismic imaging):754–773.

Kucukelbir, A., Tran, D., Ranganath, R., Gelman, A., and Blei, D. M. (2017). Automatic differentiation variational inference. *Journal of machine learning research*.

Kullback, S. and Leibler, R. A. (1951). On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86.

Lajaunie, C., Courrioux, G., and Manuel, L. (1997). Foliation fields and 3d cartography in geology: principles of a method based on potential interpolation. *Mathematical Geology*, 29(4):571–584.

Lan, S., Bui-Thanh, T., Christie, M., and Girolami, M. (2016). Emulation of higher-order tensors in manifold monte carlo methods for bayesian inverse problems. *Journal of Computational Physics*, 308:81–101.

Lao, J., Suter, C., Langmore, I., Chimisov, C., Saxena, A., Sountsov, P., Moore, D., Saurous, R. A., Hoffman, M. D., and Dillon, J. V. (2020). tfp. mcmc: Modern markov chain monte carlo tools built for modern hardware. *arXiv preprint arXiv:2002.01184*.

Laue, S., Mitterreiter, M., and Giesen, J. (2018). Computing higher order derivatives of matrix and tensor expressions. In *Advances in Neural Information Processing Systems*, pages 2750–2759.

Law, K. J. (2014). Proposals which speed up function-space mcmc. *Journal of Computational and Applied Mathematics*, 262:127–138.

Lecarme, O. and Delvare, K. (2013). *The book of GIMP: A complete guide to nearly everything*. No Starch Press.

Li, H., Xu, Z., Taylor, G., Studer, C., and Goldstein, T. (2018). Visualizing the loss landscape of neural nets. *Advances in neural information processing systems*, 31.

Li, Y. and Oldenburg, D. W. (1998). 3-d inversion of gravity data. *Geophysics*, 63(1):109–119.

Liang, Z., Degen, D., and Wellmann, F. (2021). The application of neural operator in subsurface process simulation. In *EGU General Assembly Conference Abstracts*, pages EGU21–12940.

Liang, Z., Wellmann, F., and Ghattas, O. (2022). Uncertainty quantification of geological model parameters in 3d gravity inversion by hessian-informed markov chain monte carlo. *Geophysics*, 88(1):1–78.

Linde, N., Ginsbourger, D., Irving, J., Nobile, F., and Doucet, A. (2017). On uncertainty quantification in hydrogeology and hydrogeophysics. *Advances in Water Resources*, 110:166–181.

Lindsay, M. D., Aillères, L., Jessell, M. W., de Kemp, E. A., and Betts, P. G. (2012). Locating and quantifying geological uncertainty in three-dimensional models: Analysis of the gippsland basin, southeastern australia. *Tectonophysics*, 546:10–27.

Lindsay, M. D., Perrouty, S., Jessell, M. W., and Ailleres, L. (2013). Making the link between geological and geophysical uncertainty: geodiversity in the ashanti greenstone belt. *Geophysical Journal International*, 195(2):903–922.

Liu, J. S. (2008). *Monte Carlo strategies in scientific computing*. Springer Science & Business Media.

Liu, Q., Lee, J., and Jordan, M. (2016). A kernelized stein discrepancy for goodness-of-fit tests. In *International conference on machine learning*, pages 276–284. PMLR.

Liu, Q. and Wang, D. (2016). Stein variational gradient descent: A general purpose bayesian inference algorithm. *Advances in neural information processing systems*, 29.

Maclaurin, D., Duvenaud, D., and Adams, R. (2015). Gradient-based hyperparameter optimization through reversible learning. In *International conference on machine learning*, pages 2113–2122. PMLR.

Malehmir, A., Juhlin, C., Wijns, C., Urosevic, M., Valasti, P., Koivisto, E., Paananen, M., Kukkonen, I., and Heikkinen, P. (2011). 3d reflection seismic investigation for mine planning and exploration in the kevitsa ni-cu pge deposit, northern finland. In *2011 SEG Annual Meeting*. OnePetro.

Malinverno, A. (2002). Parsimonious bayesian markov chain monte carlo inversion in a nonlinear geophysical problem. *Geophysical Journal International*, 151(3):675–688.

Malinverno, A. and Leaney, S. (2000). A monte carlo method to quantify uncertainty in the inversion of zero-offset vsp data. In *SEG Technical Program Expanded Abstracts 2000*, pages 2393–2396. Society of Exploration Geophysicists.

Mallet, J.-L. (1992). Discrete smooth interpolation in geometric modelling. *Computer-aided design*, 24(4):178–191.

Mallet, J.-L. and Tertois, A.-L. (2010). Solid earth modeling and geometric uncertainties. In *SPE annual technical conference and exhibition*. OnePetro.

Mangoubi, O., Pillai, N. S., and Smith, A. (2018). Does hamiltonian monte carlo mix faster than a random walk on multimodal densities? *arXiv preprint arXiv:1808.03230*.

Marechal, A. (1984). Kriging seismic data in presence of faults. In *Geostatistics for natural resources characterization*, pages 271–294. Springer.

Margossian, C. C. (2019). A review of automatic differentiation and its efficient implementation. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 9(4):e1305.

Marinari, E. and Parisi, G. (1992). Simulated tempering: a new monte carlo scheme. *EPL (Europhysics Letters)*, 19(6):451.

Martin, J., Wilcox, L. C., Burstedde, C., and Ghattas, O. (2012). A stochastic newton mcmc method for large-scale statistical inverse problems with application to seismic inversion. *SIAM Journal on Scientific Computing*, 34(3):A1460–A1487.

Martino, L., Elvira, V., Luengo, D., Corander, J., and Louzada, F. (2016). Orthogonal parallel mcmc methods for sampling and optimization. *Digital Signal Processing*, 58:64–84.

Marzouk, Y., Moselhy, T., Parno, M., and Spantini, A. (2016). An introduction to sampling via measure transport. *arXiv preprint arXiv:1602.05023*.

Massiot, C. and Caumon, G. (2010). Accounting for axial directions, cleavages and folding style during 3d structural modeling. In *30th Gocad Meeting Proceedings, 30th Gocad meeting*.

Merland, R., Caumon, G., Lévy, B., and Collon-Drouaillet, P. (2014). Voronoi grids conforming to 3d structural features. *Computational Geosciences*, 18(3):373–383.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092.

Mosegaard, K. and Tarantola, A. (1995). Monte carlo sampling of solutions to inverse problems. *Journal of Geophysical Research: Solid Earth*, 100(B7):12431–12447.

Mosser, L., Dubrule, O., and Blunt, M. J. (2020). Stochastic seismic waveform inversion using generative adversarial networks as a geological prior. *Mathematical Geosciences*, 52(1):53–79.

Moulaeifard, M., Wellmann, F., Bernard, S., de la Varga, M., and Bommes, D. (2022). Subdivide and conquer: Adapting non-manifold subdivision surfaces to surface-based representation and reconstruction of complex geological structures. *Mathematical Geosciences*, pages 1–31.

Muir, J. B. and Tkalcic, H. (2015). Probabilistic joint inversion of lowermost mantle p-wave velocities and core mantle boundary topography using differential travel times and hierarchical hamiltonian monte-carlo sampling. *AGUFM*, 2015:S14A–03.

Nabighian, M. N., Ander, M., Grauch, V., Hansen, R., LaFehr, T., Li, Y., Pearson, W., Peirce, J., Phillips, J., and Ruder, M. (2005a). Historical development of the gravity method in exploration. *Geophysics*, 70(6):63ND–89ND.

Nabighian, M. N., Grauch, V., Hansen, R., LaFehr, T., Li, Y., Peirce, J. W., Phillips, J. D., and Ruder, M. (2005b). The historical development of the magnetic method in exploration. *Geophysics*, 70(6):33ND–61ND.

Nagy, D. (1966). The gravitational attraction of a right rectangular prism. *Geophysics*, 31(2):362–371.

Neal, R. M. (1993). Bayesian learning via stochastic dynamics. In *Advances in neural information processing systems*, pages 475–482.

Neal, R. M. et al. (2011). Mcmc using hamiltonian dynamics. *Handbook of markov chain monte carlo*, 2(11):2.

Nilsen, G. K., Munthe-Kaas, A. Z., Skaug, H. J., and Brun, M. (2019). Efficient computation of hessian matrices in tensorflow. *arXiv preprint arXiv:1905.05559*.

Oates, C. J., Girolami, M., and Chopin, N. (2017). Control functionals for monte carlo integration. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 79(3):695–718.

Olierook, H. K., Scalzo, R., Kohn, D., Chandra, R., Farahbakhsh, E., Clark, C., Reddy, S. M., and Müller, R. D. (2021). Bayesian geological and geophysical data fusion for the construction and uncertainty quantification of 3d geological models. *Geoscience Frontiers*, 12(1):479–493.

Pakyuz-Charrier, E., Lindsay, M., Ogarko, V., Giraud, J., and Jessell, M. (2018). Monte carlo simulation for uncertainty estimation on structural data in implicit 3-d geological modeling, a guide for disturbance distribution selection and parameterization. *Solid Earth*, 9(2):385–402.

Pankratov, O. and Kuvshinov, A. (2016). Applied mathematics in em studies with special emphasis on an uncertainty quantification and 3-d integral equation modelling. *Surveys in Geophysics*, 37(1):109–147.

Parker, R. L. (1974). Best bounds on density and depth from gravity data. *Geophysics*, 39(5):644–649.

Parker, R. L. (1975). The theory of ideal bodies for gravity interpretation. *Geophysical Journal International*, 42(2):315–334.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. (2017). Automatic differentiation in pytorch.

Petra, N., Martin, J., Stadler, G., and Ghattas, O. (2014). A computational framework for infinite-dimensional bayesian inverse problems, part ii: Stochastic newton mcmc with application to ice sheet flow inverse problems. *SIAM Journal on Scientific Computing*, 36(4):A1525–A1555.

Pinski, F. J., Simpson, G., Stuart, A. M., and Weber, H. (2015). Algorithms for kullback–leibler approximation of probability measures in infinite dimensions. *SIAM Journal on Scientific Computing*, 37(6):A2733–A2757.

Pirot, G., Renard, P., Huber, E., Straubhaar, J., and Huggenberger, P. (2015). Influence of conceptual model uncertainty on contaminant transport forecasting in braided river aquifers. *Journal of Hydrology*, 531:124–141.

Pollack, A., Cladouhos, T. T., Swyer, M. W., Siler, D., Mukerji, T., and Horne, R. N. (2021). Stochastic inversion of gravity, magnetic, tracer, lithology, and fault data for geologically realistic structural models: Patua geothermal field case study. *Geothermics*, 95:102129.

Pompe, E., Holmes, C., and Łatuszyński, K. (2020). A framework for adaptive mcmc targeting multimodal distributions. *The Annals of Statistics*, 48(5):2930–2952.

Portniaguine, O. and Zhdanov, M. S. (1999). Focusing geophysical inversion images. *Geophysics*, 64(3):874–887.

Qi, Y. and Minka, T. P. (2002). Hessian-based markov chain monte-carlo algorithms.

Raissi, M., Perdikaris, P., and Karniadakis, G. E. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378:686–707.

Ranganath, R., Gerrish, S., and Blei, D. (2014). Black box variational inference. In *Artificial intelligence and statistics*, pages 814–822. PMLR.

Rashidifard, M., Giraud, J., Lindsay, M., Jessell, M., and Ogarko, V. (2021). Constraining 3d geometric gravity inversion with 2d reflection seismic profile using a generalized level-set approach: application to eastern yilgarn craton. *Solid Earth Discussions*, pages 1–35.

Rath, V., Wolf, A., and Bücker, H. (2006). Joint three-dimensional inversion of coupled groundwater flow and heat transfer based on automatic differentiation: sensitivity calculation, verification, and synthetic examples. *Geophysical Journal International*, 167(1):453–466.

Rawlinson, N., Fichtner, A., Sambridge, M., and Young, M. K. (2014). Seismic tomography and the assessment of uncertainty. *Advances in geophysics*, 55:1–76.

Reamer, S. K. and Ferguson, J. F. (1989). Regularized two-dimensional fourier gravity inversion method with application to the silent canyon caldera, nevada. *Geophysics*, 54(4):486–496.

Ren, Z., Zhong, Y., Chen, C., Tang, J., and Pan, K. (2018). Gravity anomalies of arbitrary 3d polyhedral bodies with horizontal and vertical mass contrasts up to cubic order. *Geophysics*, 83(1):G1–G13.

Rezende, D. and Mohamed, S. (2015). Variational inference with normalizing flows. In *International conference on machine learning*, pages 1530–1538. PMLR.

Robert, C. P., Elvira, V., Tawn, N., and Wu, C. (2018). Accelerating mcmc algorithms. *Wiley Interdisciplinary Reviews: Computational Statistics*, 10(5):e1435.

Roberts, G. O. and Rosenthal, J. S. (2001). Optimal scaling for various metropolis-hastings algorithms. *Statistical science*, 16(4):351–367.

Roberts, G. O., Tweedie, R. L., et al. (1996). Exponential convergence of langevin distributions and their discrete approximations. *Bernoulli*, 2(4):341–363.

Rodriguez Piceda, C., Scheck Wenderoth, M., Gomez Dacal, M. L., Bott, J., Prezzi, C. B., and Strecker, M. R. (2021). Lithospheric density structure of the southern central andes constrained by 3d data-integrative gravity modelling. *International Journal of Earth Sciences*, 110(7):2333–2359.

Rohit, R., Kiplangat, D. C., Veena, R., Jose, R., Pradeepkumar, A., Kumar, K. S., et al. (2023). Tracing the evolution and charting the future of geothermal energy research and development. *Renewable and Sustainable Energy Reviews*, 184:113531.

Rongier, G., Collon-Drouaillet, P., and Filipponi, M. (2014). Simulation of 3d karst conduits with an object-distance based method integrating geological knowledge. *Geomorphology*, 217:152–164.

Ross, M., Parent, M., and Lefebvre, R. (2005). 3d geologic framework models for regional hydrogeology and land-use management: a case study from a quaternary basin of southwestern quebec, canada. *Hydrogeology Journal*, 13(5):690–707.

Rossky, P. J., Doll, J., and Friedman, H. (1978). Brownian dynamics as smart monte carlo simulation. *The Journal of Chemical Physics*, 69(10):4628–4633.

Roy, V. (2020). Convergence diagnostics for markov chain monte carlo. *Annual Review of Statistics and Its Application*, 7:387–412.

Rücker, C., Günther, T., and Wagner, F. M. (2017). pyGIMLi: An open-source library for modelling and inversion in geophysics. *Computers and Geosciences*, 109:106–123.

Rudolf, D. and Sprungk, B. (2018). On a generalization of the preconditioned crank–nicolson metropolis algorithm. *Foundations of Computational Mathematics*, 18(2):309–343.

Rudoy, D. and Wolfe, P. J. (2006). Monte carlo methods for multi-modal distributions. In *2006 Fortieth Asilomar Conference on Signals, Systems and Computers*, pages 2019–2023. IEEE.

Ruggeri, P., Irving, J., and Holliger, K. (2015). Systematic evaluation of sequential geostatistical resampling within mcmc for posterior sampling of near-surface geophysical inverse problems. *Geophysical Journal International*, 202(2):961–975.

Sambridge, M., Bodin, T., Gallagher, K., and Tkalčić, H. (2013). Transdimensional inference in the geosciences. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 371(1984):20110547.

Sambridge, M. and Mosegaard, K. (2002). Monte carlo methods in geophysical inverse problems. *Reviews of Geophysics*, 40(3):3–1.

Sambridge, M., Rickwood, P., Rawlinson, N., and Sommacal, S. (2007). Automatic differentiation in geophysical inverse problems. *Geophysical Journal International*, 170(1):1–8.

Saul, L. and Jordan, M. (1995). Exploiting tractable substructures in intractable networks. *Advances in neural information processing systems*, 8.

Scalzo, R., Kohn, D., Olierook, H., Houseman, G., Chandra, R., Girolami, M., and Cripps, S. (2019). Efficiency and robustness in monte carlo sampling for 3-d geophysical inversions with obsidian v0. 1.2: setting up for success. *Geoscientific Model Development*, 12(7):2941–2960.

Scalzo, R., Lindsay, M., Jessell, M., Pirot, G., Giraud, J., Cripps, E., and Cripps, S. (2021). Blockworlds 0.1. 0: A demonstration of anti-aliased geophysics for probabilistic inversions of implicit and kinematic geological models. *Geoscientific Model Development Discussions*, pages 1–33.

Schaaf, A., de la Varga, M., Wellmann, F., and Bond, C. E. (2021). Constraining stochastic 3-d structural geological models with topology information using approximate bayesian computation in gempy 2.1. *Geoscientific Model Development*, 14(6):3899–3913.

Schweizer, D., Blum, P., and Butscher, C. (2017). Uncertainty assessment in 3-d geological models of increasing complexity. *Solid Earth*, 8(2):515–530.

Seidl, D. T. and Granzow, B. N. (2022). Calibration of elastoplastic constitutive model parameters from full-field data with automatic differentiation-based sensitivities. *International Journal for Numerical Methods in Engineering*, 123(1):69–100.

Sen, M. K. and Stoffa, P. L. (1996). Bayesian inference, gibbs' sampler and uncertainty estimation in geophysical inversion 1. *Geophysical Prospecting*, 44(2):313–350.

Sharma, S., Sharma, S., and Athaiya, A. (2017). Activation functions in neural networks. *Towards Data Sci*, 6(12):310–316.

Shi, B., Bai, X., and Yao, C. (2016). An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *IEEE transactions on pattern analysis and machine intelligence*, 39(11):2298–2304.

Singer, D. and Menzie, W. D. (2010). *Quantitative mineral resource assessments: An integrated approach*. Oxford University Press.

Skeels, D. C. (1947). Ambiguity in gravity interpretation. *Geophysics*, 12(1):43–56.

Stamm, F. A., de la Varga, M., and Wellmann, F. (2019). Actors, actions, and uncertainties: optimizing decision-making based on 3-d structural geological models. *Solid Earth*, 10(6).

Stephens, M. (2000). Dealing with multimodal posteriors and non-identifiability in mixture models. *Journal of the Royal Statistical Society, Series B, xx (xx)*.

Stern, R. J., Anthony, E. Y., Ren, M., Lock, B. E., Norton, I., Kimura, J.-I., Miyazaki, T., Hanyu, T., Chang, Q., and Hirahara, Y. (2011). Southern louisiana salt dome xenoliths: First glimpse of jurassic (ca. 160 ma) gulf of mexico crust. *Geology*, 39(4):315–318.

Stigler, S. M. (1986). Laplace's 1774 memoir on inverse probability. *Statistical Science*, 1(3):359–363.

Sun, J. and Li, Y. (2016). Joint inversion of multiple geophysical data using guided fuzzy c-means clustering. *Geophysics*, 81(3):ID37–ID57.

Suzuki, S., Caumon, G., and Caers, J. (2008). Dynamic data integration for structural modeling: model screening approach using a distance-based model parameterization. *Computational Geosciences*, 12(1):105–119.

Tacher, L., Pomian-Srzednicki, I., and Parriaux, A. (2006). Geological uncertainties associated with 3-d subsurface models. *Computers & Geosciences*, 32(2):212–221.

Tarantola, A. and Valette, B. (1982). Generalized nonlinear inverse problems solved using the least squares criterion. *Reviews of Geophysics*, 20(2):219–232.

Team, S. D. et al. (2016). Stan modeling language users guide and reference manual. *Team SD*.

Thiele, S. T., Jessell, M. W., Lindsay, M., Wellmann, J. F., and Pakyuz-Charrier, E. (2016). The topology of geology 2: Topological uncertainty. *Journal of Structural Geology*, 91:74–87.

Tierney, L. and Kadane, J. B. (1986). Accurate approximations for posterior moments and marginal densities. *Journal of the american statistical association*, 81(393):82–86.

Tikhonov, A. N. and Arsenin, V. Y. (1977). Solutions of ill-posed problems. *New York*, pages 1–30.

Tran, D., Blei, D., and Airoldi, E. M. (2015a). Copula variational inference. *Advances in neural information processing systems*, 28.

Tran, D., Ranganath, R., and Blei, D. M. (2015b). The variational gaussian process. *arXiv preprint arXiv:1511.06499*.

Turk, G. and O'brien, J. F. (2005). Shape transformation using variational implicit functions. In *ACM SIGGRAPH 2005 Courses*, pages 13–es.

Villa, U., Petra, N., and Ghattas, O. (2018). hippylib: An extensible software framework for large-scale inverse problems. *Journal of Open Source Software*, 3(30):940.

Villa, U., Petra, N., and Ghattas, O. (2021). hippylib: An extensible software framework for large-scale inverse problems governed by pdes: Part i: Deterministic inversion and linearized bayesian inference. *ACM Transactions on Mathematical Software (TOMS)*, 47(2):1–34.

Wang, G., Li, R., Carranza, E. J. M., Zhang, S., Yan, C., Zhu, Y., Qu, J., Hong, D., Song, Y., Han, J., et al. (2015). 3d geological modeling for prediction of subsurface mo targets in the luanchuan district, china. *Ore Geology Reviews*, 71:592–610.

Wellmann, F. (2022). Geological modeling 4.0. In *Handbook Industry 4.0*, pages 807–819. Springer.

Wellmann, F. and Caumon, G. (2018). 3-d structural geological models: Concepts, methods, and uncertainties. 59:1–121.

Wellmann, J. F., de la Varga, M., Murdie, R. E., Gessner, K., and Jessell, M. (2018). Uncertainty estimation for a geological model of the sandstone greenstone belt, western australia–insights from integrated geological and geophysical inversion in a bayesian inference framework. *Geological Society, London, Special Publications*, 453(1):41–56.

Wellmann, J. F., Horowitz, F. G., Schill, E., and Regenauer-Lieb, K. (2010). Towards incorporating uncertainty of structural data in 3d geological inversion. *Tectonophysics*, 490(3-4):141–151.

Wellmann, J. F. and Regenauer-Lieb, K. (2012). Uncertainties have a meaning: Information entropy as a quality measure for 3-d geological models. *Tectonophysics*, 526:207–216.

Wellmann, J. F., Thiele, S. T., Lindsay, M. D., and Jessell, M. W. (2016). pynoddy 1.0: an experimental platform for automated 3-d kinematic and potential field modelling. *Geoscientific Model Development*, 9(3):1019–1035.

Witter, J. B., Trainor-Guitton, W. J., and Siler, D. L. (2019). Uncertainty and risk evaluation during the exploration stage of geothermal development: A review. *Geothermics*, 78:233–242.

Wong, R. (2001). Asymptotic approximations of integrals. society for industrial and applied mathematics.

Wu, K., O'Leary-Roseberry, T., Chen, P., and Ghattas, O. (2022). Derivative-informed projected neural network for large-scale bayesian optimal experimental design. *arXiv preprint arXiv:2201.07925*.

Wu, L. (2018). Efficient modeling of gravity fields caused by sources with arbitrary geometry and arbitrary density distribution. *Surveys in Geophysics*, 39(3):401–434.

Wu, L. and Chen, L. (2016). Fourier forward modeling of vector and tensor gravity fields due to prismatic bodies with variable density contrastvariable density contrast. *Geophysics*, 81(1):G13–G26.

Wu, Q., Xu, H., and Zou, X. (2005). An effective method for 3d geological modeling with multi-source data integration. *Computers & geosciences*, 31(1):35–43.

Yamamoto, J. K., Koike, K., Kikuda, A. T., da Cruz Campanha, G. A., and Endlen, A. (2014). Post-processing for uncertainty reduction in computed 3d geological models. *Tectonophysics*, 633:232–245.

Yang, D., Oldenburg, D. W., and Haber, E. (2014). 3-d inversion of airborne electromagnetic data parallelized and accelerated by local mesh and adaptive soundings. *Geophysical Journal International*, 196(3):1492–1507.

Yao, Y., Vehtari, A., and Gelman, A. (2020). Stacking for non-mixing bayesian computations: The curse and blessing of multimodal posteriors. *arXiv preprint arXiv:2006.12335*.

Zhang, J. and Jiang, L. (2017). Analytical expressions for the gravitational vector field of a 3-d rectangular prism with density varying as an arbitrary-order polynomial function. *Geophysical Journal International*, 210(2):1176–1190.

Zhang, X. and Curtis, A. (2020a). Seismic tomography using variational inference methods. *Journal of Geophysical Research: Solid Earth*, 125(4):e2019JB018589.

Zhang, X. and Curtis, A. (2020b). Variational full-waveform inversion. *Geophysical Journal International*, 222(1):406–411.

Zhang, Y. et al. (2019). Bayesian semi-supervised learning for uncertainty-calibrated prediction of molecular properties and active learning. *Chemical science*, 10(35):8154–8163.

Zhao, H. and Kowalski, J. (2022). Bayesian active learning for parameter calibration of landslide run-out models. *Landslides*, pages 1–13.

Zhou, X. (2009). 3d vector gravity potential and line integrals for the gravity anomaly of a rectangular prism with 3d variable density contrast. *Geophysics*, 74(6):I43–I53.

Zhu, Y. and Zabaras, N. (2018). Bayesian deep convolutional encoder–decoder networks for surrogate modeling and uncertainty quantification. *Journal of Computational Physics*, 366:415–447.