Theses and Dissertations

2023-06-07

Hamiltonian Monte Carlo for Reconstructing Historical Earthquake-Induced Tsunamis

Jacob Paul Callahan
Brigham Young University

Follow this and additional works at: https://scholarsarchive.byu.edu/etd

Part of the Physical Sciences and Mathematics Commons

BYU ScholarsArchive Citation
Callahan, Jacob Paul, "Hamiltonian Monte Carlo for Reconstructing Historical Earthquake-Induced Tsunamis" (2023). Theses and Dissertations. 9950.
https://scholarsarchive.byu.edu/etd/9950

This Thesis is brought to you for free and open access by BYU ScholarsArchive. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of BYU ScholarsArchive. For more information, please contact ellen_amatangelo@byu.edu.
Hamiltonian Monte Carlo for Reconstructing Historical Earthquake-Induced Tsunamis

Jacob Paul Callahan

A thesis submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of
Master of Science

Jared P. Whitehead, Chair
Tyler J. Jarvis
Mark C. Kempton
L. Clark Callahan

Department of Mathematics
Brigham Young University

Copyright © 2023 Jacob Paul Callahan
All Rights Reserved
ABSTRACT

Hamiltonian Monte Carlo for Reconstructing Historical Earthquake-Induced Tsunamis

Jacob Paul Callahan
Department of Mathematics, BYU
Master of Science

In many areas of the world, seismic hazards pose a great risk to both human and natural populations. In particular, earthquake-induced tsunamis are especially dangerous to many areas in the Pacific. The study and quantification of these seismic events can both help scientists better understand how these natural hazards occur and help at-risk populations make better preparations for these events. However, many events of interest occurred too long ago to be recorded by modern instruments, so data on these earthquakes are sparse and unreliable. To remedy this, a Bayesian method for reconstructing the source earthquakes for these historical tsunamis based on anecdotal data, called TsunamiBayes, has been developed and used to study historical events that occurred in 1852 and 1820.

One drawback of this method is the computational cost to reconstruct posterior distributions on tsunami source parameters. In this work, we improve on the TsunamiBayes method by introducing higher-order MCMC methods, specifically the Hamiltonian Monte Carlo (HMC) method to increase sample acceptance rate and therefore reduce computation time. Unfortunately the exact gradient for this problem is not available, and so we make use of a surrogate gradient via a neural network fitted to the forward model. We examine the effects of this surrogate gradient HMC sampling method on the posterior distribution for an 1852 event in the Banda Sea, compare results to previous results collected using random walk, and note the benefits of the surrogate gradient in this context.

Keywords: Bayesian statistics, Markov chain Monte Carlo, Hamiltonian Monte Carlo, inverse problems, earthquakes, tsunamis, seismic hazard analysis
ACKNOWLEDGEMENTS

I am greatly indebted to my advisor, Dr. Jared Whitehead. His patience, encouragement, and countless hours of mentorship changed the way I look at both mathematics as a subject and at myself as a mathematician. Any success I have achieved as a graduate student I owe directly to him.

I am also grateful to my committee members: Dr. Mark Kempton for being a fantastic teacher and helping me push my limits as a student and Dr. Tyler Jarvis for taking me on to his research group when I was an undergraduate and helping me discover a love for research.

I am thankful for my time in the Department of Mathematics at BYU and for the many professors whose mentorship and instruction I have received. In particular, I owe thanks to Dr. Blake Barker and Dr. Emily Evans for their undeserved support for me, somewhat delinquent student that I was, as well as to the rest of the professors in the ACME program for providing me with such a wonderful and challenging undergraduate experience that forced me far beyond my comfort zone.

I am grateful for the friends and collaborators who made both my research and graduate experience a success: James Griffin, Taylor Paskett, Clay Williams, Eric Steadman, Grant White, and many others with whom I had the privilege to spend my time. I’m grateful to the other students who collaborated with me on this research, without whom this thesis would not be possible: Raelynn Wonnacott, Chelsey Noorda, Paul Smith, Claire Ashcraft, and many others. Additionally, I am thankful to Drs. Justin Krometis and Nathan Glatt-Holtz for their time spent helping me understand the intricacies of Hamiltonian Monte Carlo both in theory and in application.

Finally, I owe my deepest gratitude my family. My wife, Elizabeth, who I love dearly, made this work possible through her love, encouragement, and sacrifice. This thesis is as much a product of her hard work as it is mine. I am grateful to my two wonderful boys, Benjamin and Cooper, for sharing their dad with his work. My heart will always belong to them.
# Contents

## List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
</table>

## List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
</table>

## 1 Introduction

## 2 The TsunamiBayes Method

2.1 Bayesian Inference .......................... 3
2.2 Forward Model ............................ 6
2.3 Prior Distribution and Parameter Space .......... 9
2.4 Total likelihood (observational distributions) .... 11

## 3 Sampling Methods

3.1 Markov Chain Monte Carlo (MCMC) ............... 14
3.2 Hamiltonian Monte Carlo (HMC) ................. 18

## 4 Gradient Methods

4.1 Analytic gradient .................................. 30
4.2 Surrogate model choice and architecture ........ 34

## 5 Results

5.1 Performance of HMC .......................... 37
5.2 Effects of the surrogate gradient ............... 39

## 6 Conclusion and Future Work

## Bibliography


List of Tables

2.1 Prior distributions for the 1852 Banda Arc earthquake . . . . . . . . . . . . 11

5.1 Acceptance rate across MCMC schemes for 1852 Banda Arc earthquake . . . 37
## List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>A map of Indonesia demonstrating many of the most impactful seismic events in the Indonesian archipelago [1, Figure 12].</td>
</tr>
<tr>
<td>2.1</td>
<td>All selected observation locations for the 1852 Banda Arc earthquake and tsunami.</td>
</tr>
<tr>
<td>2.2</td>
<td>A picture of the area described in the account on page 42 of the Wichmann catalog, originally found in [2, Figure 9]. Lines are drawn indicating the inundation inferred from the descriptions in the text for several historical events. These inferences are made based on historical knowledge combined with knowledge about seasonal storms and assumed standard flood level. The 1852 runup line is drawn to match the description of water reaching “the base of the hill on which Fort Belgica is built.”</td>
</tr>
<tr>
<td>2.3</td>
<td>All likelihood densities for the observations at each observation location. Each density represents an interpretation of an account from the Wichmann catalog.</td>
</tr>
<tr>
<td>4.1</td>
<td>Regularized neural network prediction results at several gauge locations for a spread of input earthquake parameters. Each distribution represents the outputs of a model at a discrete grid of inputs across the sample domain. The surrogate model achieves good coverage of the distributions at each gauge, although the network output does appear to be tighter, i.e., less variance in the depicted histograms than those of Geoclaw. As we are more interested in the overall distribution of each of these observables, this relative comparison between the neural network output and the output generated from Geoclaw is of more interest than variations in a single value.</td>
</tr>
</tbody>
</table>
5.1 Visualization of how chains explore the sample space in several different variables. Each line represents a chain that was started at the same point close to the mode of the distribution and has a different integration time. With a long enough integration time, the chains appear to explore the sample space well. Even though each chain began at the same point, most moved to different areas of sample space and avoided getting stuck.

5.2 Acceptance rates as a function of integration time. Each chain was run for 100 samples starting at the same point (near the mode of the distribution found in [3]). Clearly, step quantity and size don’t affect acceptance rate nearly as much as total integration time (Note that for integration times of .075 and .1, the acceptance rate for both time step sizes was exactly the same).
Chapter 1. Introduction

In many areas of the world, seismic hazards pose a great risk to both human and natural populations. In particular, earthquake-induced tsunamis are especially dangerous to many areas in the Pacific. Indonesia is one such area. In fact, it is one of the most seismically active areas in the world.

![Figure 1.1: A map of Indonesia demonstrating many of the most impactful seismic events in the Indonesian archipelago [1, Figure 12]](image)

Due to the danger posed by seismic events and tsunamis in particular, it is important to conduct seismic risk analyses in this area. However, as most seismological data was recorded after 1900 (accurate readings of seismic activity have only been available since the Cold War), analysis is hindered by the fact that no mega-thrust earthquakes have been recorded in many potential source areas near Indonesia [4]. These data are only available since the advent of the Cold War, so risk estimates can be biased by relying only on seismological data: using only events from the last 70 years ignores that the temporal scale for a given fault zone can span hundreds or thousands of years. Thus, risk estimates can be greatly improved by incorporating earthquake data recorded prior to modern instruments being available.

While accurate seismological data are sparse, there are many pre-instrumental tsunami
events in the Indonesian archipelago recorded in various historical records dating back to the 17th century ([5], [6]). [3] developed a method for reconstructing the source earthquake of tsunami events from sparse anecdotal data, and applied it to an event that occurred in 1852 in the Banda Sea of Eastern Indonesia. Their method uses Markov chain Monte Carlo sampling to recover a Bayesian posterior distribution of earthquake parameters based on historical accounts of past tsunamis and the destruction caused by said waves. While powerful, this method is limited by the computational cost of recovering the source parameter distributions. In the years since this method and the associated tsunamibayes code-base were published in [3], work has been done in order to speed up this process. In particular, [7] attempted to use the Metropolis-Adjusted Langevin Algorithm (MALA) method to achieve a sampling speedup. The approach taken in [7] relied on an extremely simple surrogate model for the generative forward model, and thus is limited in its ability to recover an accurate posterior for the source parameters. In fact, [7] found that the surrogate model only performed well if the initial set of parameters was far removed from a realistic value that matched the data well. On the other hand, the surrogate was unable to distinguish the difference between two different “reasonable” sets of parameters so that resolution of the full posterior was severely restricted. [8] proposed a method for using the gradient of a surrogate model to approximate the true gradient to enable using Hamiltonian Monte Carlo (HMC), which is a general case of MALA. This method was meant for situations with complex forward models, but was only evaluated on relatively simple problems.

In this work, we improve on the methods developed in [3, 7] by implementing the Hamiltonian Monte Carlo sampling method proposed in [8] to replace the random walk sampling method used originally, while using a far more accurate surrogate gradient than that used in [7].

The rest of this thesis will proceed as follows: Chapter 2 gives an overview of the TsunamiBayes method developed in [3] including application to the Banda 1852 event. Next, in Chapter 3 we provide a brief overview of the theoretical underpinnings of Hamiltonian
Monte Carlo. In order to use HMC, we require a gradient of the generative forward model, so in Chapter 4 we discuss using surrogate modeling in order to recover a computationally tractable gradient. Finally, we present the results of using our surrogate-gradient HMC approach in Chapter 5 to collect samples from the 1852 event. Chapter 6 concludes with future work and lessons learned.

Chapter 2. The TsunamiBayes Method

The TsunamiBayes method (and relevant Python software package) aims to reconstruct earthquakes that caused them historical tsunamis from sparse anecdotal data. This method is well suited for events about which there exists no recorded data from seismic instruments, but there are written or oral accounts of the tsunami or earthquake. This method is a Bayesian approach, and as such requires three components:

- A forward model
- A prior distribution
- A likelihood

In the following sections, we provide a summary of the Bayesian method and discuss the construction of each of these components in detail as applied to the 1852 Banda Sea earthquake and tsunami. Using the 1852 event, which was already studied in detail in 3, will allow us to compare the performance of HMC to an existing study. The end goal of TsunamiBayes is to produce a posterior distribution on seismic parameters describing the tsunami-generating earthquake.

2.1 Bayesian Inference

We first provide a brief overview of Bayesian probability and inference. We will follow an outline of the material in 9, where the interested reader may find more details. Under
the frequentist approach, probability is treated as a measure of randomness and seeks to measure the inherent variability in a system or process. In contrast, the Bayesian framework treats probability as a measure of *epistemic* uncertainty. Epistemic uncertainty refers to uncertainty due to lack of knowledge. Under the Bayesian framework, randomness is treated as a consequence of imperfect knowledge, and thus probability distributions are interpreted as a quantification of how uncertain the observer is about the root cause of a process. Thus, Bayesian probability can be used to reason about processes we do not believe to be truly random; we only need to be uncertain about some aspect of the process for Bayesian inference to be useful.

In settings where the goal is to recover a “true” set of parameters, this makes Bayesian inference a natural choice. Bayesian inference offers a wide set of tools that make quantifying, reasoning about, and reducing uncertainty possible. We provide here a brief overview of Bayesian inference.

Suppose we wish to model some process that depends on a set of parameters $\theta$. Further suppose that we also have a set of data $\mathcal{D}$ we believe to result from the underlying process. A question that arises naturally from this setting is which set of parameters produced the data we have observed, which we can think of as the “true parameters.” While we cannot know with 100% certainty which parameters produced the observed data, we can reason about which parameters are more likely to be the true parameters. If we can assign a likelihood to each possible set of parameters, we can get an idea of which ones would be good candidates for the true parameters.

To state this mathematically, we are interested in a probability distribution on $\theta$ that is conditioned on the data we have observed. We refer to the conditional probability density function (PDF) of this distribution as the posterior distribution $P(\theta|\mathcal{D})$. This distribution can be computed via Bayes’ Theorem:

**Theorem 2.1** (Bayes’ Theorem). For two events $A$ and $B$, the conditional probability
\[ P(A|B) \text{ is given by} \]
\[ P(A|B) = \frac{P(B|A)P(A)}{\int P(B|A')P(A')dA'}. \]

Bayes’ Theorem gives a method both for computing conditional probabilities for single events and for computing conditional probability distributions. In Bayesian inference we are typically interested in the latter case. In the parameter estimation setting which we have been discussing, we are interested in computing the distribution \( P(\theta|\mathcal{D}) \). This is referred to as the posterior distribution, and can be thought of as a distribution describing belief about the likelihood of possible parameters after we’ve observed a dataset \( \mathcal{D} \) that was generated by some unknown set of parameters. For notational ease, we will hereafter refer to this distribution as \( \pi_{\text{post}} \).

To compute \( \pi_{\text{post}} \), we need several things. We need a distribution on data conditioned on a given set of parameters, i.e. \( P(B|A) \) in the formula above. This is referred to as the likelihood, and we notate it as \( \mathcal{L}(\mathcal{D}|\theta) \). We also need a distribution describing the probability of a given set of parameters independent of the observed data, i.e. \( P(A) \). This is referred to as the prior, and it can be thought of as a distribution describing a priori belief about the likelihood of possible parameters. We will notate it as \( \pi_{\text{prior}} \). Finally, to directly compute the posterior, we would need a distribution describing the probability of data independent of model parameters (the denominator in the formula above). In practice, since this requires taking an integral of functions that are often complex and high-dimensional, this term is typically intractable to compute. In Chapter 3 we discuss methods for getting around this limitation.

In order to find the distribution describing how likely a set of parameters is to be the true parameter set, we are thus interested in computing
\[ \pi_{\text{post}}(\theta|\mathcal{D}) = \frac{\mathcal{L}(\mathcal{D}|\theta)\pi_{\text{prior}}(\theta)}{\int \mathcal{L}(\mathcal{D}|\theta')\pi_{\text{prior}}(\theta')d\theta'}. \tag{2.1} \]

In particular, we are interested in describing the earthquake that generated the 1852 Banda Arc tsunami, so \( \theta \) is a set of parameters that describe this earthquake. In the following sections we discuss the forward model, which models a tsunami generated by a given set of
earthquake parameters, the prior distribution on earthquake parameters, and observational probabilities which, when combined with the forward model, will yield the likelihood on observed tsunami data $D$.

2.2 Forward Model

The forward model for this problem takes as input the set of parameters $\theta$ that describes a seismic event (things like centroid location, magnitude, and geometry) and maps them to observed tsunami characteristics at a given location (potential observations such as wave height, wave arrival time, and inundation). This process involves modeling the seafloor deformation caused by the seismic event and the propagation of the resultant tsunami across the ocean to the observation points of interest. We use a numerical model called GeoClaw \cite{10} to perform this process. GeoClaw computes seafloor deformation from input parameters and then solves the nonlinear shallow water equations using a dynamically adaptive mesh finite-volume solver.

2.2.1 Seafloor Deformation: The Okada Model. We provide a brief overview of how the seafloor deformation is computed. GeoClaw uses a model of surface displacement for a given earthquake called the Okada model \cite{11}, \cite{12}. This model makes several simplifying assumptions: including assuming the Earth is made of a homogeneous isotropic elastic material, that the earth is a flat surface, and that it has infinite surface area. These are reasonable assumptions for small areas (relative to the size of the entire globe), and this model is widely accepted as a good approximation for seafloor deformation in this context.

The Okada model, which we denote as $O$, accepts 9 input parameters that parameterize the source earthquake and maps them to the perturbed seafloor deformation across the entire domain (in this case we restrict the domain to a rectangular set of latitude-longitude coordinates). These parameters are

- Centroidal latitude ($c_{lat}$)
• Centroidal longitude ($c_{lon}$)

• Earthquake depth ($d$)

• Strike angle, indicating the direction parallel to the fault ($strike$)

• Rake angle, indicating the direction the rupture moves relative to strike ($rake$)

• Dip angle, indicating how far from horizontal the fault is inclined ($dip$)

• Length of rectangular fault ($l$)

• Width of rectangular fault ($w$)

• Slip of rectangular fault ($s$).

In practice, we typically work with a second set of parameters that can be mapped to the Okada input parameters as explained in [3]. These parameters are

• Centroidal latitude ($c_{lat}$)

• Centroidal longitude ($c_{lon}$)

• Earthquake magnitude ($m$)

• Difference from predicted log-length of slab ($\Delta \ell$)

• Difference from predicted log-width of slab ($\Delta w$)

• Depth offset (difference from predicted values from the SLAB2 data [13]), ($d_0$)

These parameters, to which we refer collectively as sample parameters and which we notate as $q$, are the parameters on which we wish to recover a posterior distribution. We map $q$ to the Okada input parameters, to which we collectively refer as model parameters and notate as $z$, using a simple map we call the Map-to-Okada (M2O) function, i.e., $z = M2O(q)$. Details on this map and the choice of sample parameters can be found in [3]. Finally, we
denote the seafloor deformation as $h_0(x)$ where $x$ refers to the rectangular latitude-longitude coordinates of the domain. We thus write the Okada model as

$$h_0(x) = \mathcal{O}(z) = \mathcal{O}(M2O(q)).$$

### 2.2.2 Forward Propagation: shallow water solver.

Once the seafloor deformation $h_0(x)$ has been computed, GeoClaw then propagates the tsunami forward in time, computing wave heights and arrival times at each observation point of interest. We denote this map as $\mathcal{G}$, i.e. the wave height at time $t$ and latitude-longitude location $x$ is calculated via $h(x,t) = \mathcal{G}(h_0(x))$. We are specifically interested in $h$ at specific values of $x$ (where the observations were made) and when the maximal value of $h$ is reached at each of these locations, which we call the arrival time and denote as $a(x)$. The propagation of $h(x,t)$ is performed by solving the nonlinear shallow water equations in GeoClaw.

The solver in GeoClaw uses an adaptive mesh refinement method for a finite volume scheme [10]. The adaptive mesh strategy solves a linearized adjoint equation backward in time for each observation location. The solution produces waves that are then propagated backward to determine which parts of the wave influence each observation point. See [10, 14] for details. For a discussion of how the bathymetry and topography of the region of interest influence the solver and how those parameters were chosen for our application, see [3]. The full forward model is thus given by

$$\mathbf{h}, \mathbf{a} = \mathcal{G}(h_0(x,t)), \quad (2.2)$$

$$h_0(x,t) = \mathcal{O}(M2O(q)), \quad (2.3)$$

where $\mathbf{h}$ and $\mathbf{a}$ are vectors of wave heights and arrival times (respectively) at each observation point.
2.3 Prior Distribution and Parameter Space

In order to perform Bayesian inference, we need to define a prior probability for each sample. These priors are typically informed by data, domain knowledge, or other information known *a priori*. In this section we briefly discuss the choice of prior distribution for our experiments. We assign a prior distribution $\pi_i$ to each parameter we are sampling. A summary of all prior distributions and their parameters can be found in Table 2.1.

2.3.1 Latitude and longitude. It is only physically possible for tsunami-generating earthquakes to occur in a certain depth range. In locations that are too shallow, the subduction interface would protrude above the surface and the colliding plates are too plastic for the necessary stress to build up, but in locations that are too deep the earthquake will not produce a sufficient seafloor deformation to instigate a tsunami. This means that centroidal location is constrained by the subduction depth interface geometry of the plate tectonics in the region. This means that any probability distribution on depth implies a distribution on latitude and longitude as well. Based on observed data and on domain expert advice, this leads us to choose a truncated normal distribution for depth. We use the SLAB2 data \[13\], a dataset that converts latitude-longitude coordinate pairs into depth of major subduction zones across the globe, to evaluate this distribution for latitude and longitude, thereby providing us a prior distribution on location. This distribution is supported on $[2.5, 50]$ kilometers, with a mean of 30km and a standard deviation of 5km, all selected to be consistent with tsunami-generating earthquakes in the modern USGS catalog.

2.3.2 Magnitude. Earthquake magnitude has been observed to follow an exponential distribution \[15\], and it is generally modeled in seismology as such. We choose to follow this precedent for our prior, and use an exponential distribution that is right-truncated at magnitude 9.5 with an exponential distribution parameter of $\lambda = .5$ \[16\].
2.3.3 Delta log parameters. The parameters $\Delta \ell$ and $\Delta w$ are defined as residuals of a best-fit line (see [3]), and so it is natural to model them as Gaussian-distributed variables with mean 0. The standard deviations for each of these Gaussian distributions were recovered from the sample variances of the residuals collected from an augmented Wells-Coppersmith dataset [17] against the log-linear fit. These values were $\sigma_{\Delta \ell} = 0.188$ and $\sigma_{\Delta w} = 0.172$. See [3] for more details.

2.3.4 Depth offset. The prior distribution for depth offset was chosen to be a mean-zero Gaussian distribution with standard deviation of 5. This value was chosen by analyzing the reported depth uncertainties from the SLAB2 dataset, in which the average reported uncertainty was 5km.

2.3.5 Total prior. Finally, we assume independence between all five of the aforementioned distributions as there is not sufficient knowledge to determine the actual dependence of each of these parameters on each other. This means that our total prior distribution is given by

$$
\pi_{\text{prior}} = \prod_i \pi_i(q)
$$

where $q$ is the vector of sample parameters and $\pi_i(q)$ refers to the prior distribution defined on the $i^{th}$ component of $q$. 

10
Table 2.1: Prior distributions for the 1852 Banda Arc earthquake

<table>
<thead>
<tr>
<th>Parameter name(s)</th>
<th>Kind</th>
<th>Distribution Parameters</th>
</tr>
</thead>
</table>
| Latitude & longitude       | pre-image of truncated normal via depth | • $\mu = 30\text{km}$  
• $\sigma = 5\text{km}$  
• $(a, b) = (2.5\text{km},50\text{km})$ |
| Magnitude                  | truncated exponential       | • $\lambda = .5$  
• $(a, b) = (6.5,9.5)$ |
| $\Delta \log L$            | normal                      | • $\mu = 0$  
• $\sigma = .188$ |
| $\Delta \log W$            | normal                      | • $\mu = 0$  
• $\sigma = .172$ |
| depth offset               | normal                      | • $\mu = 0$  
• $\sigma = 5\text{km}$ |

2.4 TOTAL LIKELIHOOD (OBSERVATIONAL DISTRIBUTIONS)

The full likelihood of this event is the combination of the output of the forward model plugged into assigned observational probability distributions, which we now discuss. Observational distributions are constructed by analyzing historical accounts at selected observation locations. These observations are selected from the Wichmann [5,6] catalog, a collection of Dutch historical records of tsunamis and earthquakes in and around Indonesia. These accounts are selected based on whether they contain identifiable locations that can be translated to a (relatively) precise latitude/longitude coordinate. They must also contain enough information about the tsunami that some level of confidence can be ascribed to the observation. We identified 9 observation points around the Banda arc for the 1852 earthquake and tsunami. These observation locations are depicted in Figure 2.1.

From these observations, we extract two types of observable:

- Arrival time, which describes the time the first noticeable wave arrived at the observation point.
Figure 2.1: All selected observation locations for the 1852 Banda Arc earthquake and tsunami

- Maximum wave height, which describes the height of the tallest observed wave.

We use the information recorded about these observables to construct observational distributions for arrival time and wave height at each selected observation point (in [3] an inundation length was also used as an observable, but we have found since then that the inundation introduces no additional information relative to the maximum wave height).

One example of an observation can be found on page 42 of the Wichmann catalog: “Barely had the ground been calm for a quarter of an hour when the flood wave crashed in...The water rose to the roofs of the storehouses and homes...[the wave] reached the base of the hill on which Fort Belgica is built on Banda Neira.” From this account we can use knowledge about the layout of Fort Belgica and the geography of Banda Neira to infer an observation latitude of $-4.524^\circ$ and an observation longitude of $129.8965^\circ$. From historical
knowledge about Indonesian building construction in the mid 1800s and knowledge about seasonal storms and assumed standard flood level, we can model wave height using a normal distribution with mean 6.5 meters and standard deviation of 1.5 meters. See Figure 2.2 for a visual representation of this process.

We follow this same process for wave height and arrival time across all 9 selected observation locations. For more details on this process, see [3]. The final observational distributions are depicted in Figure 2.3. The likelihood of a certain set of parameters \( q \) is then computed by first computing \( h, a \) via (2.2) and (2.3) and then inserting these values into the observational distributions depicted in Figure 2.3. The total likelihood is then computed as a product of all of the observational probabilities (we again are assuming that all the observations are statistically independent).
Figure 2.3: All likelihood densities for the observations at each observation location. Each density represents an interpretation of an account from the Wichmann catalog.

CHAPTER 3. SAMPLING METHODS

In this chapter we describe the sampling methods used to recover the posterior distribution on earthquake parameters. We use Markov Chain Monte-Carlo methods to explore the space of possible earthquakes and sample from that space according to the posterior distribution. We describe the Metropolis Hastings algorithm used to perform this sampling, and we discuss the Hamiltonian Monte Carlo algorithm which we use to perform the sampling in a more efficient way than a simple random walk which was previously exploited in [3, 18].

3.1 MARKOV CHAIN MONTE CARLO (MCMC)

The overall goal of this research is to construct the posterior distribution on earthquake parameters $\theta$ given our observed data $D$, as given by Bayes’ Rule:

$$
\pi_{post}(\theta|D) = \frac{p(D|\theta)\pi_{prior}(\theta)}{p(D)} = \frac{p(D|\theta)\pi_{prior}(\theta)}{\int p(D|\theta')d\theta'}.
$$
In practice, computing this distribution analytically is infeasible, as integrating the normalizing constant in the denominator is computationally intractable if not impossible. Instead, we use Markov chain Monte-Carlo (MCMC) sampling to sample from the posterior distribution in such a way that we recover a set of samples that sufficiently represent the distribution in question. We do this by constructing a Markov chain whose stationary distribution is the posterior distribution and sampling from the full distribution for a sufficiently large enough number of samples will then provide a good approximation of the overall distribution.

In this section we discuss Markov chains and their properties and the method for constructing a Markov chain with the desired stationary distribution for further details see [19] for example.

3.1.1 Markov chains.

Definition 3.1. Let $X = \{ X_0, X_1, \cdots, X_n \}$ be a sequence of random variables defined on a probability space $(\Omega, \mathcal{A}, P)$. $X$ is said to be a Markov chain if it satisfies the Markov property:

$$
P(X_{n+1} \in A|X_n = x_n, \cdots, X_1 = x_1) = P(X_{n+1} = x_{n+1} \in A|X_n = x_n) \quad \forall A \in \mathcal{A}
$$

In other words, a Markov chain is a sequence of random variables where the probability of transitioning to the next element of the sequence (the transition probability) depends only on the value of the current element of the sequence. This property is sometimes referred to as memorylessness. We are interested in a particular type of Markov chain called time-homogeneous Markov chains.

Definition 3.2. A Markov chain is said to be time-homogeneous if

$$
P(X_{n+k} \in A|X_n) = P(X_k \in A|X_0) \quad \forall n, k \in \mathbb{R}.
$$

This means that the transition probabilities of a time-homogeneous Markov chain do not depend on the time index. This type of Markov chain has well known properties surrounding its stationary distribution that we can exploit to construct a Markov chain with the posterior distribution as its stationary distribution.
3.1.2 The stationary distribution.

**Definition 3.3.** A distribution $\pi$ for $X_n$ is called **stationary** if the subsequent marginal distribution for $X_{n+1}$ is also $\pi$.

For a time-homogeneous Markov chain with finite, discrete states, this is equivalent to the statement

$$Q\pi = \pi$$

where $Q$ is the matrix recording the transition probabilities from $X_n$ to $X_{n+1}$, and so we can identify the stationary distribution by finding the eigenvector of $Q$ with corresponding eigenvalue 1.

In the continuous case, identifying this distribution is more difficult. However, as shown in [20], we can construct a Markov chain with the desired stationary distribution by using the *detailed balance* condition. Consider the *probability transition kernel* for a continuous-state Markov chain, which is the infinite-dimensional analogue to the transition matrix of finite-state Markov chains. This transition kernel has the form

$$P(X_{n+1} \in A | X_n = x) = \int_A K(x, y)dy + r(x)\chi_A(x)$$

where $r(x)$ is the probability of remaining at state $x$ and $K(x, y)$ is the probability of transitioning from state $x$ to state $y$. The balance equation states that if a distribution $\pi(x)$ is the stationary distribution for a Markov chain with transition density $K(x, y)$ then

$$\int K(x, y)\pi(x)dx = \int K(y, x)\pi(y)dy.$$

The *detailed balance* equation, which is a stronger condition and hence implies the balance equation, is given by

$$K(x, y)\pi(x) = K(y, x)\pi(y).$$

We can thus construct a Markov chain with the desired stationary distribution by choosing a transition probability density that satisfies the detailed balance equation.
3.1.3 Metropolis-Hastings algorithm. The Metropolis-Hastings algorithm is a commonly-used method for constructing and sampling from Markov chains whose stationary distributions match a target distribution that we want to sample from. We first begin by letting $\pi$ represent the distribution from which we wish to sample and rewriting the detailed balance equation:

\[
K(x, y)\pi(x) = K(y, x)\pi(y)
\]

\[
\Rightarrow \frac{K(x, y)}{K(y, x)} = \frac{\pi(y)}{\pi(x)}.
\]

Now, we choose the transition distribution to be the product of two different distributions: the proposal distribution, denoted as $Q(x, y)$, and the acceptance probability, denoted as $A(x, y)$. The proposal distribution represents the probability of proposing a new state $y$ given the current state $x$ and the acceptance probability represents the probability of accepting the proposal $y$ given the current state $x$. We write this choice as

\[
K(x, y) = Q(x, y)A(x, y)
\]

and can thus rewrite the detailed balance equation:

\[
\frac{K(x, y)}{K(y, x)} = \frac{\pi(y)}{\pi(x)}
\]

\[
\Rightarrow \frac{Q(x, y)A(x, y)}{Q(y, x)A(y, x)} = \frac{\pi(y)}{\pi(x)}
\]

\[
\Rightarrow \frac{A(x, y)}{A(y, x)} = \frac{\pi(y) Q(y, x)}{\pi(x) Q(x, y)}.
\]

It remains to choose $A$ such that this equation is satisfied. The Metropolis-Hastings algorithm chooses

\[
A(x, y) = \min \left( 1, \frac{\pi(y) Q(y, x)}{\pi(x) Q(x, y)} \right),
\]

which we can see satisfies the detailed balance equation.

Intuitively, the idea of the Metropolis-Hastings algorithm is to propose a new state by obtaining a proposal from $q(x, \cdot)$. Ideally, $Q$ would provide a computationally cheap method for proposing new states in the Markov Chain; in many cases, for example, a random walk proposal is used. Then, the algorithm accepts or rejects this sample according to the correction
probability defined by $A$. The full algorithm is given by Algorithm 1. Metropolis-Hastings is a well-studied method, and many thorough treatments may be found in the literature. See, for example, [21, 22, 23, 24] for further details.

**Algorithm 1: Metropolis-Hastings**

Initialize: $\theta_0$

for $i = 0, \ldots$ do

Propose $\tilde{\theta} \sim Q(\theta_i, \cdot)$

Set $a = \min \left( 1, \frac{\pi(\theta)Q(\theta, x_i)}{\pi(x_i)Q(\tilde{\theta}, \theta)} \right)$

Draw $u \sim U(0, 1)$

if $u < a$ then

Set $\theta_{i+1} = \tilde{\theta}$

else

Set $\theta_{i+1} = \theta_i$

end

3.2 **Hamiltonian Monte Carlo (HMC)**

An MCMC scheme utilizing a random walk as its proposal method is sufficient for many applications. However, in some cases there are factors that make using a simple random walk undesirable or even intractable. A “good” acceptance rate for random walk MCMC is considered to be at or above 23.4% of samples [25]. This means that in the ideal setting over 75% of the computational time does not go toward resolving the desired target distribution. In cases where a single sample takes significant computational resources to generate, this rate is too low to guarantee convergence in a reasonable amount of time. This is exactly what we are encountering for the tsunami-earthquake inversion problem.

Depending on the size of the area of interest, a single forward run of the Geoclaw model could take from several seconds to several hours. In order to generate a posterior with sufficient confidence in convergence, thousands or even hundreds of thousands of samples must be drawn for a random walk proposal kernel. In order to apply our method to larger events, like the Lisbon earthquake and tsunami of 1755 [26] we must use an MCMC scheme that accepts samples at a higher rate. We thus turn to a method called Hamiltonian Monte
Carlo. We will provide a summary here of the description given by Neal in [27], where a more detailed discussion of the Hamiltonian Monte Carlo method can be found.

Hamiltonian Monte Carlo, which is a special case of the Metropolis-Hastings algorithm that defines a specific involved proposal kernel \( Q \), draws inspiration from Hamiltonian dynamics, which describe the motion of a particle over a frictionless surface. This motion, simulated over some time interval \( T \), is what acts as the proposal function in the Metropolis-Hastings update step. In this section we describe the equations of motion governing the Hamiltonian system and discuss how these equations are used as a proposal function for Metropolis-Hastings.

3.2.1 Equations of motion. The motion in a Hamiltonian system is parameterized in terms of position \( \tilde{q} \) (traditionally the position is \( q \), but we have already used \( q \) above) and momentum \( p \). We call the function describing the system the Hamiltonian \( H(\tilde{q}, p) \). The Hamiltonian is usually written as a sum of potential energy \( U \) and kinetic energy \( K \):

\[
H(\tilde{q}, p) = U(\tilde{q}) + K(p)
\]  

(3.2)

The equations governing motion in this system are known as Hamilton’s equations and are given by

\[
\frac{d\tilde{q}_i}{dt} = \frac{\partial H}{\partial p_i}
\]

(3.3)

\[
\frac{dp_i}{dt} = -\frac{\partial H}{\partial \tilde{q}_i}.
\]

(3.4)

for \( i = 1, 2, \cdots, D \) where \( D \) is the number of dimensions in the system.

3.2.2 Solving Hamilton’s equations. Depending on how the Hamiltonian is defined, it can be difficult to solve Hamilton’s equations analytically, as is the case in our setting. We do this by discretizing the system in time and iteratively computing the state at time \( t = 0, \varepsilon, 2\varepsilon, 3\varepsilon, \cdots \) for some small value of \( \varepsilon \). Euler’s method (see [28] for more details on numerical methods in general) is the most well-known way of performing these updates. In this setting Euler’s method gives an approximate update from time \( t \) to time \( t + \varepsilon \) in both
position and momentum by the following:

\[ p_i(t + \varepsilon) = p_i(t) + \varepsilon \frac{dp_i}{dt}(t) \]  
(3.5)

\[ \tilde{q}_i(t + \varepsilon) = \tilde{q}_i(t) + \varepsilon \frac{d\tilde{q}_i}{dt}(t). \]  
(3.6)

Using Hamilton’s equations to make substitutions, we obtain

\[ p_i(t + \varepsilon) = p_i(t) - \varepsilon \frac{\partial H}{\partial \tilde{q}_i}(t) \]  
(3.7)

\[ \tilde{q}_i(t + \varepsilon) = \tilde{q}_i(t) + \varepsilon \frac{\partial H}{\partial p_i}(t). \]  
(3.8)

If we further assume that the kinetic energy of the system is dictated by

\[ K(p) = \frac{1}{2} p^T M^{-1} p, \]

where \( M \) is a (as yet unspecified) mass matrix (we further assume below that \( M = mI \) is a constant multiple of the identity matrix), then the update becomes

\[ p_i(t + \varepsilon) = p_i(t) - \frac{\partial U}{\partial \tilde{q}_i}(t), \]  
(3.9)

\[ \tilde{q}_i(t + \varepsilon) = \tilde{q}_i(t) + \varepsilon \frac{p_i(t)}{m}. \]  
(3.10)

3.2.3 The Leapfrog Method. Unfortunately, when integrating a Hamiltonian system, the Euler method performs poorly due to the fact that the Hamiltonian is not conserved under the Euler update equations. Instead, a modification to this method, called the leapfrog method, can be used. The update equations are given by

\[ p_i(t + \frac{\varepsilon}{2}) = p_i(t) - \frac{\varepsilon}{2} \left( \frac{\partial U}{\partial \tilde{q}_i}(\tilde{q}(t)) \right) \]  
(3.11)

\[ \tilde{q}_i(t + \varepsilon) = \tilde{q}_i(t) + \varepsilon \left( \frac{p_i(t + \frac{\varepsilon}{2})}{m_i} \right) \]  
(3.12)

\[ p_i(t + \varepsilon) = p_i(t + \frac{\varepsilon}{2}) - \frac{\varepsilon}{2} \left( \frac{\partial U}{\partial \tilde{q}_i}(\tilde{q}(t + \varepsilon)) \right). \]  
(3.13)

This integration scheme has several useful properties that make it an ideal choice for HMC, which we discuss in Section 3.2.5.

3.2.4 Using HMC as a proposal. We wish to use Hamilton’s equations to sample from our posterior. Intuitively, this works by imagining that the position in sample space
is marked by a particle. The particle moves around in sample space, periodically stopping. Its position when it stops is a new proposed sample. Sometimes this proposed sample is accepted and evolving position continues, and other times the proposal is rejected and instead the evolution of the particle restarts from the previous state value. We suppose that this particle follows Hamiltonian dynamics for a randomly selected momentum \( p \), which governs the motion of the particle through sample space. We will now describe exactly how to define a proposal kernel for the Metropolis-Hastings algorithm that samples from our desired posterior distribution. We begin with a brief overview of a concept from statistical mechanics.

**Definition 3.4.** The **Boltzmann distribution**, sometimes called the Gibbs distribution, is a probability distribution that describes the probability of a system being at a certain state. It is a function of the system’s energy, \( E(x) \) and of the system’s temperature \( T \), and is given by

\[
P(x) = \frac{1}{Z} \exp \left( -\frac{E(x)}{T} \right)
\]

where \( Z \) is a normalizing constant.

We will use the Boltzmann distribution to build a distribution which we can explore using Hamilton’s equations. First, recall that the Hamiltonian, \( H(\tilde{q}, p) \) is a joint energy function of position \( \tilde{q} \) and momentum \( p \). If we let \( T = 1 \) and define \( H \) as in equation \( 3.2 \), we see that it has a corresponding Boltzmann distribution given by

\[
P(\tilde{q}, p) \propto \exp(-H(\tilde{q}, p)).
\]

(3.14)

Expanding this equation gives us

\[
P(\tilde{q}, p) \propto \exp(-U(\tilde{q})) \exp(-K(p)).
\]

(3.15)

This implies that \( \tilde{q} \) and \( p \) are independent, and that they have respective energy functions \( U \) and \( K \) with corresponding Boltzmann distributions

\[
P(\tilde{q}) \propto \exp(-U(\tilde{q}))
\]

(3.16)
Finally, to connect this back to our desired sampling strategy, we set the potential energy function to be the negative log of the target posterior distribution:

$$U(\tilde{q}) = -\log(\pi_{prior}(\tilde{q})\mathcal{L}(D|\tilde{q})).$$  \hfill (3.18)

Substituting back into the Boltzmann distribution, we see that

$$P(\tilde{q}) \propto \exp\left(-\log(\pi_{prior}(\tilde{q})\mathcal{L}(D|\tilde{q}))\right)$$

$$= \exp(\log(\pi_{prior}(\tilde{q})\mathcal{L}(D|\tilde{q})))$$

$$= \pi_{prior}(\tilde{q})\mathcal{L}(D|\tilde{q})$$

$$\propto \pi_{post}(\tilde{q}|D).$$

Thus, if we equip a state space with an energy function equal to the negative log of the posterior, the distribution on position of a particle in that space will be equal to the posterior distribution.

Returning to our intuitive explanation, this means that a particle wandering this space will be in position $\tilde{q}$ with probability proportional to $\pi_{post}(\tilde{q}|D)$, and thus repeatedly sampling this particle’s movement should give us an accurate reconstruction of the posterior distribution.

### 3.2.5 Deriving an acceptance probability for HMC.

In order to use Metropolis-Hastings to sample the particle’s movement we must define an acceptance probability

$$A(q^*, p^*, q, p)$$

that satisfies detailed balance $\theta$ going forward). If we were able to integrate our Hamiltonian system exactly, this would define a symmetric proposal function and deriving an acceptable acceptance function would be trivial. However, our numerical integrator (3.2.2) only provides approximations to the true solution, and therefore we cannot say our acceptance function is symmetric. We thus require a measure-theoretic argument to derive the correct acceptance probability function for Hamiltonian Monte Carlo. We here
outline the argument provided by [29], and encourage the interested reader to refer to the full paper for more details.

First we briefly review some key concepts from measure theory.

**Definition 3.5 (Pushforward measure).** Let \((\mathcal{X}, \Sigma_\mathcal{X})\) and \((\mathcal{Y}, \Sigma_\mathcal{Y})\) be measurable spaces. Given a measurable function \(\phi : \mathcal{X} \rightarrow \mathcal{Y}\) and a measure \(\mu\) on \((\mathcal{X}, \Sigma_\mathcal{X})\), the **pushforward measure** of \(\nu\) by \(\phi\) is defined as the measure on \(\mathcal{Y}\) given by

\[
\phi^*\nu(A) := \nu(\phi^{-1}(A)) \quad \text{for any } A \in \Sigma_\mathcal{Y}.
\] (3.19)

The pushforward measure is a useful concept that allows us to define probability measures in a natural way. The following theorem discusses how to represent the pushforward in terms of the original variables.

**Theorem 3.6 (Change of variables).** Let \(w\) be a random variable sampled from a probability measure \(\nu\), so that \(\phi(w)\) is distributed according to \(\phi^*\nu\). Further, let \(\nu\) be a Borel measure on \(\mathbb{R}^n\) and let it be absolutely continuous (See Def. 3.7) with respect to the Lebesgue measure \(\rho\). Then for any diffeomorphism \(\phi : \mathbb{R}^n \rightarrow \mathbb{R}\), we have

\[
\phi^*\nu(dw) = p(\phi^{-1}(w))|\det \nabla \phi^{-1}(w)dw|
\] (3.20)

for some density function \(p\).

In order for this theorem to apply, the measure \(\nu\) must be absolutely continuous with respect to the Lebesgue measure, a concept that we define now.

**Definition 3.7 (Absolute continuity).** Let \(\nu\) and \(\rho\) be measures on a measurable space \((\mathcal{X}, \Sigma_\mathcal{X})\). We say \(\nu\) is **absolutely continuous** with respect to \(\rho\) and write \(\nu \ll \rho\) if \(\nu(A) = 0\) if \(\rho(A) = 0\) for all \(A \in \Sigma_\mathcal{X}\). Equivalently, we may say \(\nu \ll \rho\) if there exists some density function \(p : \mathbb{R}^n \rightarrow \mathbb{R}\) such that

\[
\nu(dw) = p(w)\rho(dw).
\]

Finally, we define the Radon-Nikodym derivative.
**Definition 3.8** (Radon-Nikodym derivative). Let $\nu$ and $\rho$ be sigma-finite measures on $(\mathfrak{X}, \Sigma_\mathfrak{X})$ such that $\nu \ll \rho$. Then there exists a function $\frac{d\nu}{d\rho} \in L^1(\rho)$ such that

$$\nu(A) = \int_A \frac{d\nu}{d\rho}(w) \rho(dw), \quad \forall A \in \Sigma_\mathfrak{X}$$

which we call the **Radon-Nikodym** derivative of $\nu$ with respect to $\rho$.

In particular, if $\nu_1$, $\nu_2$, and $\rho$ are sigma-finite measures on $(\mathfrak{X}, \Sigma_\mathfrak{X})$ with $\nu_1 \ll \rho$ and $\nu_2 \ll \rho$, such that

$$\nu_1 = \phi_1(w) \rho(dw), \quad \nu_2 = \phi_2(w) \rho(dw),$$

with $\phi_1 = \frac{d\nu_1}{d\rho}$ and $\phi_2 = \frac{d\nu_2}{d\rho}$ and $\phi_2 > 0$ $\rho$-a.e., then $\nu_1 \ll \nu_2$ and

$$\frac{d\nu_1}{d\nu_2}(w) = \frac{\phi_1}{\phi_2}(w), \quad \text{for } w \in \mathfrak{X}.$$  \hspace{1cm} (3.21)

This result allows us to directly compute the Radon-Nikodym derivative in cases where both measures of interest are absolutely continuous with respect to a third measure (typically the Lebesgue measure). We now use the Radon-Nikodym derivative to create an acceptance function for a general Markov (proposal) kernel that satisfies detailed balance.

**Theorem 3.9.** Let $(X, \Sigma_X)$ and $(Y, \Sigma_Y)$ be measurable spaces. Let $\mu$ be a probability measure on $X$, and let $V : X \times \Sigma_Y \rightarrow [0, 1]$ be a Markov kernel. Let $\mathcal{M}$ be the probability measure on $X \times Y$ defined as

$$\mathcal{M}(dq, dv) = V(q, dv)\mu(dq).$$  \hspace{1cm} (3.22)

Suppose there exists a measurable mapping $S : X \times Y \rightarrow X \times Y$ satisfying the following properties:

(i) $S$ is an involution, i.e. $S^2 = I$

(ii) $S^*\mathcal{M}$ is absolutely continuous with respect to $\mathcal{M}$.

Let $\hat{\alpha} : X \times Y \rightarrow \mathbb{R}$ be the function defined by

$$\hat{\alpha}(q, v) := 1 \wedge \frac{dS^*\mathcal{M}}{d\mathcal{M}}(q, v), \quad (q, v) \in X \times Y,$$  \hspace{1cm} (3.23)
and let $P : X \to [0, 1]$ be the Markov kernel defined as

$$P(q, dq) = \int_Y \hat{\alpha}(q, v) \delta_{\Pi \circ S(q, v)}(dq, dv) \mathcal{V}(q, dv) + \delta_q(dq) \int_Y (1 - \hat{\alpha}(q, v)) \mathcal{V}(q, dv),$$  

(3.24)

for $q \in X$. Then, $P$ satisfies detailed balance with respect to $\mu$, i.e.

$$P(q, dq) \mu(dq) = P(\bar{q}, dq) \mu(d\bar{q}),$$  

(3.25)

so that, in particular, $\mu$ is $P$ invariant.

We omit the proof (see [29]), but provide intuition. Under the provided assumptions, an $S$ that maps $(q, v)$ to $(\bar{q}, \bar{v})$ will also map $(\bar{q}, \bar{v})$ to $(q, v)$. This fact means that if we can use the Markov kernel $P$ to generate a proposal $\bar{q}$ starting from a point $q$, then we can equally likely also use $P$ to start at a point $q$ and generate a proposal $\bar{q}$. If these assumptions are satisfied, Theorem 3.9 gives us a way to choose an acceptance probability that guarantees detailed balance will be achieved.

Recall that we are interested in sampling from the distribution on particle position given in 3.16. Since it is a probability distribution, we can write it as a measure $\mu$:

$$\mu(dq) = \frac{1}{Z_U} e^{-U(q)} dq.$$

(3.26)

However, since modeling the position of the particle also involves modeling the velocity, we need to work with the joint Boltzmann distribution on both position $q$ and velocity $v$:

$$\mathcal{M}(dq, dv) = \frac{1}{Z_H} e^{-H(q, v)} dv.$$

(3.27)

Note that up to now we have been working with momentum $p$ and not velocity $v$. Since momentum is defined as mass times velocity, we can work through the following with either variable. In order to make the marginal of $\mathcal{M}$ exactly equal to $\mu$, we have to choose a slightly different Hamiltonian function, namely

$$\mathcal{H} = U(q) + K(q, v) + \ln(Z_k(q)), \quad Z_k(q) := \int \limits_{\mathbb{R}^n} e^{-K(q, v)} dv.$$

(3.28)
Using this Hamiltonian, we will now write this joint distribution in terms of $\mu$.

$$
\mathcal{M}(dq, dv) = \frac{1}{Z} e^{-H(q, v)}
\quad = \frac{1}{Z} e^{U(q)} e^{-K(q,v)}
\quad = \frac{1}{Z} \cdot \frac{1}{Z_K} e^{U(q)} e^{-K(q,v)}
\quad = \frac{1}{Z} \frac{1}{Z_K} e^{U(q)} e^{-K(q,v)}.
$$

Letting $Z = \int e^{-U(q)} dq$ and $V(q, v) = \frac{1}{Z_K} e^{-K(q,v)}$, we can write

$$
\mathcal{M}(dq, dv) = \mu(q) V(q, v).
$$

(3.29)

Denoting $\mathcal{B}(\mathbb{R}^n)$ as the Borel sets on $\mathbb{R}^n$, $\mathcal{V} : \mathbb{R}^n \times \mathcal{B}(\mathbb{R}^n) \to [0, 1]$ must be a Markov kernel by construction (see [29]). Since $\mu$ is a probability measure, $\mathcal{M}$ then defines a probability measure on $\mathbb{R}^{2N}$ in the form required by Theorem 3.9.

We now consider the transition probability kernel. Intuitively, we wish to propose a new sample by moving our “particle” around sample space for a certain amount of time $T$ and then taking its new position as a proposal. This movement is performed by integrating the Hamiltonian dynamics over $T$. We will refer to this integrator as $\hat{S}$. In practice, we use the leapfrog integration scheme (3.11), but this integrator can be generalized to any symplectic integration scheme. We now consider some properties of this integrator necessary for deriving the Hamiltonian acceptance probability.

**Definition 3.10 (Symplectic).** Let $\hat{S} : \mathbb{R}^{2N} \to \mathbb{R}^{2N}$ be a $C^1$ map. We say $\hat{S}$ is **symplectic** with respect to an invertible matrix $J$ if

$$
(\nabla \hat{S}(z))^* J \nabla \hat{S}(z) = J
$$

(3.30)

for all $z \in \mathbb{R}^{2N}$ with $A^*$ denoting the conjugate transpose of $A$. Let $R : \mathbb{R}^{2N} \to \mathbb{R}^{2N}$ be a linear invertible map. We say that $\hat{S}$ is **reversible with respect to** $R$ if $\hat{S}$ is invertible and

$$
R \circ \hat{S}(z) = \hat{S}^{-1} \circ R(z).
$$

(3.31)

**Theorem 3.11 (Properties of symplectic mappings).** Let $\hat{S}$ be a symplectic mapping and let $R$ be an involution. Then:
(i) Let $\hat{S}$ be reversible with respect to $R$. Then $R \circ \hat{S}$ is an involution:

$$R \circ \hat{S} \circ (R \circ \hat{S}) = \hat{S}^{-1} \circ R \circ R \circ \hat{S} = I$$  \hspace{1cm} (3.32)

(ii) $\hat{S}$ is volume preserving in $\mathbb{R}^{2N}$, i.e.

$$| \det \nabla \hat{S}(z) | = 1$$  \hspace{1cm} (3.33)

(iii) if $\hat{S}_1$ and $\hat{S}_2$ are both symplectic mappings, then their composition is also symplectic. Similarly, if both are volume preserving, then their composition is volume preserving.

We are now ready to combine all of the preceding machinery to derive an acceptance probability for the HMC algorithm.

**Theorem 3.12** (Acceptance probability for HMC). Define a probability measure $\mu(dq) = Z_U^{-1} e^{-U(q)} dq$ as in 3.26 and a Markov kernel $\mathcal{V}(q, dv) = Z_K(q)^{-1} e^K(q, v) dv$ as in 3.27. Consider the associated Hamiltonian $H = U + K + \ln Z_U$ as defined in 3.28. Let $\hat{S} : \mathbb{R}^{2N} \to \mathbb{R}^{2N}$ be a volume-preserving mapping which is reversible with respect to a linear involution $R$ as in 3.10. Finally, let $H$ be invariant under $R$, namely

$$H(q, v) = H(R(q, v)).$$

Then the kernel $P$ defined as in 3.9 with $S = R \circ \hat{S}$ and with $\hat{\alpha}$ defined as

$$\hat{\alpha}(q, v) = \min \left( \exp(-H(\hat{S}(q), v)) + H(q, v) \right)$$  \hspace{1cm} (3.34)

satisfies detailed balance with respect to $\mu$.

**Proof.** Let $\mathcal{M} = \mathcal{V}(q, dv) \mu(dq) = \frac{1}{Z_H} e^{-H(q, v)}$ be the probability measure defined in 3.29. First, note that $\mathcal{M}$ has strictly positive density with respect to the Lebesgue measure, meaning that the pushforward measure $S^* \mathcal{M}$ must be absolutely continuous with respect to $\mathcal{M}$. Additionally, since $R$ is an involution and $\hat{S}$ is symplectic, Theorem 3.11 gives that $S = R \circ \hat{S}$ must be an involution. Thus, by Theorem 3.9 we know that the acceptance probability given by

$$\hat{\alpha}(q, v) = \min \left( 1, \frac{dS^* \mathcal{M}}{d\mathcal{M}}(q, v) \right)$$
with the associated proposal kernel $P$ satisfies detailed balance with respect to $\mu$. It remains to compute $\frac{dS^* \mathcal{M}}{d\mathcal{M}}$. We use equation 3.21:

$$\frac{dS^* \mathcal{M}}{d\mathcal{M}} = \exp(-H(S^{-1}(q, v)) + H(q, v))|\det \nabla S^{-1}(q, v)|$$

$$= \exp(-H(S^{-1}(q, v)) + H(q, v))|\det \nabla R(\hat{S}(q, v)||\det \hat{S}(qv)|.$$

Since $R$ is a linear involution, $|\det \nabla R| = 1$, and since $\hat{S}$ is volume-preserving by assumption, Theorem 3.11 (ii) gives that $|\det \nabla \hat{S}| = 1$, we have that:

$$\frac{dS^* \mathcal{M}}{d\mathcal{M}} = \exp(-H(R \circ \hat{S}(q, v)) + H(q, v)).$$

Since $H$ is invariant under $R$, we are left with:

$$\frac{dS^* \mathcal{M}}{d\mathcal{M}} = \exp(-H(\hat{S}(q, v) + H(q, v)),$$

completing the proof.

In order to use this result, we need a volume-preserving mapping $\hat{S}$ and a linear involution $R$ under which $H$ is invariant. Luckily, Hamiltonian dynamics satisfy these properties (see [27]) when computed exactly. In fact, when computed exactly, the acceptance probability given by Equation 3.34 will always be 1. Unfortunately, as discussed in Section 3.2.2, exact computation is not typically possible.

Instead, we choose to approximate Hamiltonian dynamics using the leapfrog method. While the leapfrog method doesn’t perfectly conserve the Hamiltonian, it is volume-preserving and symmetric, meaning it can be made reversible by negating $p$, applying the same number of steps, and negative $p$ again. Thus we have an operator $S$ that satisfies the desired criteria for Theorem 3.12 and we can use Equation 3.34 as the acceptance probability for the Hamiltonian Monte Carlo scheme.

3.2.6 The HMC sampling algorithm. We use Metropolis-Hastings to sample the particle’s movement around the space. We do this by choosing an initial position $q$ and sampling a random momentum $p$, typically from a Gaussian distribution, solving for the particle’s motion in time for a predetermined time interval, and then proposing its new position $\tilde{q}$ as a
new sample. This proposal is accepted or rejected according to \[ 3.34 \]. If accepted, \( \tilde{q} \) becomes the starting position. If not, \( q \) is retained as the new starting position. A new momentum \( p \) is chosen, and the process repeats. The resampling of \( p \) at each iteration is what allows movement to points of different probability density, since Hamiltonian dynamics would otherwise leave the probability density unchanged. The full algorithm is presented in Algorithm 2 below.

**Algorithm 2:** Hamiltonian Monte Carlo

1. **Initialize:** \( q_0 \)
   
   2. for \( i = 0, \ldots \) do
      
      2.1 Draw \( p_i \sim \mathcal{N}(\mu, \Sigma) \)
      
      2.2 Set \( \tilde{q}_0 = q_i \)
      
      2.3 Set \( \tilde{p}_0 = p - \varepsilon \cdot \nabla U \)
      
      3. for \( k = 1, \ldots, T \) do
         
         3.1 Set \( \tilde{q}_k = q + \varepsilon \cdot p_{k-1} \)
         
         3.2 if \( k \neq T \) then
            
            3.3 \( \tilde{p}_k = \tilde{p}_{k-1} - \varepsilon \cdot \nabla U \)
         
         3.4 end
         
         3.5 \( \tilde{p}_T = \tilde{p}_{T-1} - \varepsilon \cdot \nabla U \)
         
         3.6 Set \( \alpha = \min (1, \exp(-H(\tilde{q}_T, \tilde{p}_T) + H(q_i, p_i))) \)
         
         3.7 Draw \( \beta \sim \text{Unif}(0, 1) \)
         
         3.8 if \( \beta < \alpha \) then
            
            3.9 Set \( q_{i+1} = \tilde{q}_T \)
         
         3.10 else
            
            3.11 Set \( q_{i+1} = q_i \)
         
         3.12 end

---

**Chapter 4. Gradient Methods**

A key piece in implementing the HMC algorithm as outlined in section 3.2 is the gradient \( \nabla U \) of the potential energy function \( U \). In this chapter we describe this gradient analytically and then discuss why calculating it is infeasible. [8] outlines a method for using the gradient of a surrogate model to approximate the true gradient to enable using HMC for situations with complex forward models. In this section we discuss this method and apply it to the tsunami problem.
4.1 Analytic gradient

In order to propose a new sample for each iteration of the HMC algorithm, we use the Leapfrog method (see section 3.2.2), which requires a gradient calculation at each step. We need the gradient of the potential energy function, which is given in Equation 3.18. Here we calculate this gradient in terms of each of the sample parameters. First, note that

\[
\frac{\partial U}{\partial q_i} = -\frac{\partial \log (\pi_{\text{prior}}(q))}{\partial q_i} - \frac{\partial \log (L(q|D))}{\partial q_i}.
\]

We will consider each term in the sum separately.

4.1.1 Prior gradient. We will here provide an overview of the terms of the gradient of the negative log-prior. We omit quite a few details, and refer the interested reader to [7] where a complete discussion may be found. The negative log-prior distribution is given by

\[
-\log(\pi_{\text{prior}}) = -\sum_i \log(\pi^{(i)}_{\text{prior}}(q_i))
\]

where \(\pi^{(i)}_{\text{prior}}\) is the prior distribution defined on the \(i^{th}\) sample parameter. Since we are dealing with a sum, each entry in the gradient will be given by

\[
-\frac{\partial \log(\pi_{\text{prior}})}{\partial q_i} = \frac{-1}{\log(\pi^{(i)}_{\text{prior}}(q_i))} \cdot \frac{\partial \pi^{(i)}_{\text{prior}}}{\partial q_i}.
\]

In the tsunami problem, we have six sample parameters: latitude \(c_{\text{lat}}\), longitude \(c_{\text{lon}}\), magnitude \(m\), delta log-width \(\Delta w\), delta log-length \(\Delta \ell\), and depth offset \(d_0\). This means we have six partial derivatives that make up the gradient:

\[
-\nabla \log(\pi_{\text{prior}}) = \left[ \frac{\partial \pi^{c_{\text{lat}}}_{\text{prior}}}{\partial c_{\text{lat}}}, \frac{\partial \pi^{c_{\text{lon}}}_{\text{prior}}}{\partial c_{\text{lon}}}, \frac{\partial \pi^{m}_{\text{prior}}}{\partial m}, \frac{\partial \pi^{\Delta w}_{\text{prior}}}{\partial \Delta w}, \frac{\partial \pi^{\Delta \ell}_{\text{prior}}}{\partial \Delta \ell}, \frac{\partial \pi^{d_0}_{\text{prior}}}{\partial d_0} \right]. \quad (4.1)
\]
These partial derivatives are given by the following equations:

\[
\begin{align*}
\frac{\partial \pi^\text{clat}_{\text{prior}}}{\partial d_0} &= \text{depth\_map}(\text{clat}, \text{lon}) + 1000 \cdot d_0 - \mu_d \cdot \frac{\partial d}{\partial \text{clat}} + \sigma_d d_0 - \mu_d \sigma_d \cdot \frac{\partial d}{\partial \text{clat}} \quad (4.2) \\
\frac{\partial \pi^\text{clon}_{\text{prior}}}{\partial d_0} &= \text{depth\_map}(\text{clat}, \text{lon}) + 1000 \cdot d_0 - \mu_d \cdot \frac{\partial d}{\partial \text{clon}} + \sigma_d d_0 - \mu_d \sigma_d \cdot \frac{\partial d}{\partial \text{clon}} \quad (4.3) \\
\frac{\partial \pi^\text{m}_{\text{prior}}}{\partial m} &= \frac{1}{\sigma_m} \quad (4.4) \\
\frac{\partial \pi^\Delta w_{\text{prior}}}{\partial \Delta w} &= \frac{\Delta w - \mu_{\Delta w}}{\sigma^2_{\Delta w}} \quad (4.5) \\
\frac{\partial \pi^\Delta \ell_{\text{prior}}}{\partial \Delta \ell} &= \frac{\Delta \ell - \mu_{\Delta \ell}}{\sigma^2_{\Delta \ell}} \quad (4.6) \\
\frac{\partial \pi^d_{\text{prior}}}{\partial d_0} &= \frac{d_0 - \mu_{d_0}}{\sigma^2_{d_0}} + 1000. \quad (4.7)
\end{align*}
\]

Here, \( \mu_d \), \( \mu_{\Delta \ell} \), and \( \mu_{\Delta w} \) are the means of the depth, delta log-\( \ell \), and delta log-w distributions, respectively, and \( \sigma_d \) and \( \sigma_{d_0} \) are the standard deviations of the depth and depth offset distributions, respectively. Note here that we are calculating the derivatives of latitude and longitude by taking the derivative of the depth function with respect to each variable, since in practice we use the depth prior to evaluate the prior probability for both latitude and longitude. The \text{depth\_map} function is a centered difference scheme that calculates the derivative of the function used to calculate depth, and \text{step} refers to the stepsize of the centered difference calculation.

### 4.1.2 Likelihood gradient.

Next, we consider the gradient of the log-likelihood. Recall that the likelihood \( \mathcal{L} \) takes the form of

\[
\mathcal{L}(D|q) = \prod_k \ell_k(h_k|D),
\]

and thus the negative log-likelihood is given by

\[
-\log (\mathcal{L}(D|q)) = -\sum_k \log (\ell_k(h_k|D)).
\]

Each partial derivative of the negative log-likelihood is therefore

\[
\frac{\partial [-\log (\mathcal{L}(D|q))]}{\partial q_i} = -\sum_k \frac{1}{\ell_k(h_k|D)} \cdot \frac{\partial h_k}{\partial q_i},
\]

where \( h_k \) is the wave height at observation location \( k \).
We are now left to compute the partial derivative \( \frac{\partial h_k}{\partial q_i} \). Recall that the vector of generated wave heights \( h \) is the output of a series of composed functions:

\[
h = G(h_0(x, t))
\]

where

\[
h_0(x, t) = O(M2O(q)).
\]

For notational simplicity, we will now denote the \( n \)-dimensional output of the sample-to-model parameter map M2O as \( z \), with each entry of this vector denoted as \( z_n \), and so we can rewrite the equation above as

\[
h_0(x, t) = O(z)
\]

Now, we are interested in the partial derivatives of each \( h_k \), not the partials of \( h \) as a whole. A wave height at a particular gauge location \( k \) is obtained by giving a specific location value \( x_k \) and time value \( t_k \) as inputs to the seafloor deformation map \( h_0 \). Thus, each \( h_k \) is given by the equation

\[
h_k = G(h_0(x_k, t_k)).
\]

We are now ready to calculate the partial derivatives of \( h_k \). A careful application of the Chain Rule gives us that

\[
\frac{\partial h_k}{\partial q_i} = \frac{\partial G}{h_0(x_k, t_k)} \cdot \frac{\partial h_0}{\partial z} \cdot \frac{\partial z}{\partial q_i} = \frac{\partial G}{h_0(x_k, t_k)} \sum_n \frac{\partial h_0}{\partial z_n} \cdot \frac{\partial z_n}{\partial q_i},
\]

where \( h_0 = O(z) \).

We see that this requires us to be able to take derivatives of the Geoclaw model. Unfortunately, taking a gradient of the Geoclaw model is computationally intractable. In its current implementation, using any sort of automatic differentiation scheme to take a derivative of Geoclaw is impossible, and rewriting the model code to enable automatic differentiation is a project that would require years of work. Additionally, we have no way of saying whether Geoclaw is close enough to the continuous PDE that it models to make using the derivative
of the PDE itself a reasonable alternative.

To overcome this, we choose to use a surrogate model for Geoclaw, which we denote as \( N \). For simplicity’s sake, we choose \( N \) as a surrogate for both the Okada map and Geoclaw, such that

\[
N \approx G(O(z)). \tag{4.8}
\]

This surrogate will take the M2O outputs \( z \) (hereafter referred to as model parameters, as opposed to sample parameters \( q \)) as inputs, and produces a vector of approximate wave heights \( \hat{h} \) and arrival times \( \hat{a} \) as output. \( \frac{\partial \hat{h}_k}{\partial q_i} \) and \( \frac{\partial \hat{a}_k}{\partial q_i} \) will be given by:

\[
\frac{\partial \hat{h}_k}{\partial q_i} = \sum_n \frac{\partial N^h_k}{\partial z_n} \cdot \frac{\partial z_n}{q_i}, \tag{4.9}
\]
\[
\frac{\partial \hat{a}_k}{\partial q_i} = \sum_n \frac{\partial N^a_k}{\partial z_n} \cdot \frac{\partial z_n}{q_i}, \tag{4.10}
\]

where \( N^h_k \) corresponds to the height output of \( N \) at location \( k \) and \( N^a_k \) corresponds to the arrival time output of \( N \) at location \( k \). The total partial gradient \( \frac{\partial N}{\partial q_i} \) will thus be given by:

\[
\frac{\partial N}{\partial q_i} = \sum_n \frac{\partial N^h_k}{\partial z_n} \cdot \frac{\partial z_n}{q_i} + \sum_n \frac{\partial N^a_k}{\partial z_n} \cdot \frac{\partial z_n}{q_i}. \tag{4.11}
\]

It is important to note that this surrogate gradient will not be a perfect representation of the true gradient; in fact, it will likely differ significantly in some ways. However, it is likely “good enough” that our gradient points in a generally accurate direction: even if the gradient is not highly accurate, if it points the integration in a reasonable direction, the resultant samples will be accepted at a higher rate than those proposed by random walk.

4.1.3 Surrogate model gradient. Using equation 4.11, we can now explicitly define the gradient of our surrogate model in terms of the sample parameters of the tsunami problem. The surrogate model takes in 8 (for our setting we have fixed the rake of the rupture to 90°) parameters, many of which are functions of the sample parameters. It takes latitude \( c_{lat} \), longitude \( c_{long} \), length \( l \), width \( w \), and slip \( s \), which are each functions of \( \Delta \log_l(\Delta l), \Delta \log_w(\Delta w) \) and magnitude \( m \), dip \( \alpha \) and strike \( \beta \), which are both functions of latitude and longitude,
and depth $d$, which in turn is a function of latitude, longitude, and depth offset $d_0$. Thus the surrogate is given by:

$$
\hat{h} = N(c_{lat}(\Delta \log l, \Delta \log W, m), c_{lon}(\Delta \log l, \Delta \log W, m),
\nonumber
(l(\Delta \log l, \Delta \log W, m), w(\Delta \log l, \Delta \log W, m),
\nonumber
s(\Delta \log l, \Delta \log W, m), \alpha(c_{lat}, c_{long}),
\nonumber
\beta(c_{lat}, c_{long}), d(c_{lat}, c_{long}, d_0)).
\nonumber
$$

We need the gradient with respect to the sample parameters $c_{lat}, c_{long}, \Delta \log l, \Delta \log W, m,$ and $d_0$. This gradient is given by the following:

$$
\nabla N = \left[ \frac{\partial N}{\partial m}, \frac{\partial N}{\partial \Delta l}, \frac{\partial N}{\partial \Delta w}, \frac{\partial N}{\partial \Delta l'}, \frac{\partial N}{\partial \Delta w'}, \frac{\partial N}{\partial c_{lat}}, \frac{\partial N}{\partial c_{long}}, \frac{\partial N}{\partial d_0} \right]
\nonumber
$$

(4.12)

where

$$
\frac{\partial N}{\partial m} = \frac{\partial N}{\partial l} \frac{\partial l}{\partial m} + \frac{\partial N}{\partial w} \frac{\partial w}{\partial m} + \frac{\partial N}{\partial s} \left( \frac{\partial s}{\partial l} \frac{\partial l}{\partial m} + \frac{\partial s}{\partial w} \frac{\partial w}{\partial m} \right),
\nonumber
$$

(4.13)

$$
\frac{\partial N}{\partial \Delta l} = \frac{\partial N}{\partial l} \frac{\partial l}{\partial \Delta l} + \frac{\partial N}{\partial w} \frac{\partial w}{\partial \Delta l} + \frac{\partial N}{\partial s} \left( \frac{\partial s}{\partial l} \frac{\partial l}{\partial \Delta l} + \frac{\partial s}{\partial w} \frac{\partial w}{\partial \Delta l} \right),
\nonumber
$$

(4.14)

$$
\frac{\partial N}{\partial \Delta w} = \frac{\partial N}{\partial l} \frac{\partial l}{\partial \Delta w} + \frac{\partial N}{\partial w} \frac{\partial w}{\partial \Delta w} + \frac{\partial N}{\partial s} \left( \frac{\partial s}{\partial l} \frac{\partial l}{\partial \Delta w} + \frac{\partial s}{\partial w} \frac{\partial w}{\partial \Delta w} \right),
\nonumber
$$

(4.15)

$$
\frac{\partial N}{\partial c_{lat}} = \frac{\partial N}{\partial l} \frac{\partial l}{\partial c_{lat}} + \frac{\partial N}{\partial w} \frac{\partial w}{\partial c_{lat}} + \frac{\partial N}{\partial \alpha} \frac{\partial \alpha}{\partial c_{lat}} + \frac{\partial N}{\partial \beta} \frac{\partial \beta}{\partial c_{lat}} + \frac{\partial N}{\partial d} \frac{\partial d}{\partial c_{lat}},
\nonumber
$$

(4.16)

$$
\frac{\partial N}{\partial c_{long}} = \frac{\partial N}{\partial l} \frac{\partial l}{\partial c_{long}} + \frac{\partial N}{\partial w} \frac{\partial w}{\partial c_{long}} + \frac{\partial N}{\partial \alpha} \frac{\partial \alpha}{\partial c_{long}} + \frac{\partial N}{\partial \beta} \frac{\partial \beta}{\partial c_{long}} + \frac{\partial N}{\partial d} \frac{\partial d}{\partial c_{long}},
\nonumber
$$

(4.17)

$$
\frac{\partial N}{\partial d_0} = \frac{\partial N}{\partial d} \frac{\partial d}{\partial d_0}.
\nonumber
$$

(4.18)

These derivatives are straightforward calculations when built into a neural network.

4.2 Surrogate Model Choice and Architecture

We require a model whose gradient is easy to take and reflects the gradient of Geoclaw reasonably well. A natural choice of surrogate model for Geoclaw is a neural network. Due to their construction and training method, it is extremely easy to take a gradient of a neural network. Additionally, their flexibility in the complexity and type of functions they are able to model make them a natural choice as a surrogate for a model as complex as Geoclaw.
In [30], a surrogate model for Geoclaw is built via a neural network. Their architecture is chosen using an extensive hyperparameter search and consists of:

- 13 hidden layers.
- tanh activation function.
- learning rate of 0.01.
- 150 nodes per layer.
- RMSE cost function.

Data generated from [3] and [7] is used to train the network. This data consists of over 300,000 simulated tsunamis (using Geoclaw as the forward model) with both the input earthquake parameters and output wave height and arrival time observables.

We use this architecture but make a slight modification. Our input space is a subset of $\mathbb{R}^8$ and our output space is a subset of $\mathbb{R}^1$. With input and output spaces of such low dimensionality, it would be easy for a neural network to overfit to this data, producing a model with an unreasonably large $H^1$ norm. Effectively, such a fitted model would oscillate rapidly in order to match the data leading to wildly misbehaving gradients.

Since we are concerned with using the gradient of the neural network as a proxy for Geoclaw’s gradient, we wish to avoid errors in the gradient (typically due to overfitting) and ensure that the gradient acts “normal”—namely, that the gradient is relatively smooth and robust to step size in centered finite difference derivative approximations. We attempt to achieve this through two means: First, by using a tanh activation function instead of the traditional ReLU, and second by adding a Tikhonov regularization term to the loss function. We choose a weight on this term of 0.05. This weight is chosen in an attempt to maximize the regularization applied to the network while maintaining the accuracy of the model. We observe that with a high Tikhonov regularization weight, the neural network will regress to predicting the mean as the ideal setting will be to minimize the size of all the weights which
can most readily be achieved by fitting the data to a single value or bias term. With a low Tikhonov regularization weight, the neural network will overfit and the gradient will become unstable. We find that with a weight of .05, we achieve relatively strong predictions while maintaining a stable gradient (see Figure 4.1).

![Figure 4.1: Regularized neural network prediction results at several gauge locations for a spread of input earthquake parameters. Each distribution represents the outputs of a model at a discrete grid of inputs across the sample domain. The surrogate model achieves good coverage of the distributions at each gauge, although the network output does appear to be tighter, i.e., less variance in the depicted histograms than those of Geoclaw. As we are more interested in the overall distribution of each of these observables, this relative comparison between the neural network output and the output generated from Geoclaw is of more interest than variations in a single value.](image)

**CHAPTER 5. RESULTS**

We ran HMC chains starting both near the mode and in areas of lower probability density of the distribution reconstructed in [3]. We found that we could achieve an acceptance rate
Table 5.1: Acceptance rate across MCMC schemes for 1852 Banda Arc earthquake

<table>
<thead>
<tr>
<th>MCMC Scheme</th>
<th>Acceptance Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random Walk</td>
<td>23.9%</td>
</tr>
<tr>
<td>Surrogate MALA</td>
<td>21.8%</td>
</tr>
<tr>
<td>Surrogate Gradient HMC</td>
<td></td>
</tr>
<tr>
<td>• Chains starting near the assumed mode: 45.9%</td>
<td></td>
</tr>
<tr>
<td>• Chains starting away from the assumed mode: 60.1%</td>
<td></td>
</tr>
</tbody>
</table>

of 60.1% while still sampling the necessary areas to approximate a full posterior. While not quite reaching the 80-85% acceptance rate anticipated for an exact gradient HMC method, the current approach does yield an acceptance rate that compares very favorably to the rates achieved in [3] and [7] (see Table 5), performing approximately twice as efficiently. This is a significant improvement from the surrogate gradient MALA approach taken in [7] and is likely because the neural network is a much more accurate surrogate than the Tsunami By Formula (TBF, see [31, 32, 33]) approach implemented in [7].

5.1 PERFORMANCE OF HMC

5.1.1 Probability density of starting point. Following intuition gained in [7] we anticipated seeing a different acceptance rate when starting chains in an area of presumed high density in sample space (chosen based on the results in [3]) and an area of presumed lower density.

In fact, we found this to be true. For a chain starting approximately at the mode of the distribution found in [3], the acceptance rate was 45.9%. For a chain starting in a lower-density area of the distribution found in [3], the acceptance rate was 60.1%.

It is possible that the surrogate gradient is not accurate at a high enough resolution to point the proposal in the right direction when operating in a high density area. In other
words, it is likely that the surrogate gradient gives a “generally correct” direction at a large scale, but doesn’t give a direction that’s correct locally. This behavior was amplified in [7] where the surrogate gradient was a significantly poorer approximation than the neural network used here.

5.1.2 Exploration of sample space. We found that, as long as the step size is large enough, the chains do a relatively good job of exploring the sample space. Chains with a long enough integration time \( T \) tend not to get stuck and, if they do, eventually get unstuck by accepting a sample far away from the current position. While these results are preliminary and more investigation should be done, it seems likely that chains built by the HMC method would do a good job of reconstructing and accurate posterior distribution. See Figure 5.1 for a visualization of the chains that were run and how they explored sample space.

We note here that, while we haven’t yet collected enough samples to create a good approximation of a posterior distribution or to check convergence of chains, the samples that have been collected so far appear to be a good subsampling of the chains built in [3]. This suggests that the chains will do a good job recreating the posterior found in that study.

It is hard to say in general whether HMC will do a good job at creating the “true” posterior; while the posterior found by [3] appears to be a good suggestion of what the true 1852 Banda Sea earthquake looked like, there is no guarantee this is true. We don’t currently have access to data on an event that occurred after the advent of modern seismic instruments, and using our forward model to create such a dataset is computationally infeasible. As such, we don’t have an ability to judge any posteriors created by our methods against a “ground truth.” Despite all of this, it appears that the posterior that is being subsampled by our surrogate HMC method provides a reasonable interpretation of the 1852 Banda Sea event.
Figure 5.1: Visualization of how chains explore the sample space in several different variables. Each line represents a chain that was started at the same point close to the mode of the distribution and has a different integration time. With a long enough integration time, the chains appear to explore the sample space well. Even though each chain began at the same point, most moved to different areas of sample space and avoided getting stuck.

5.2 EFFECTS OF THE SURROGATE GRADIENT

While we ultimately achieved a good acceptance rate on chains that explored the sample domain reasonably well, we also observed several ways in which the surrogate model negatively
affected MCMC performance.

5.2.1 Acceptance rate. We found that acceptance rate depends almost entirely on the integration time scale length. As integration time is increased, we see a drop off in acceptance rate. This effect is independent of time step size, meaning a leapfrog scheme using 100 time steps of size .001 will result in approximately the same acceptance rate as a leapfrog scheme using 10 time steps of size .1. This is most likely explained by the usage of the surrogate gradient: since the surrogate gradient is not exact, it’s possible that if allowed to integrate long enough, the integrator is “falling off” the constant Hamiltonian curves that HMC is trying to follow. Once this happens, acceptance rates begin to drop.

![Figure 5.2: Acceptance rates as a function of integration time. Each chain was run for 100 samples starting at the same point (near the mode of the distribution found in [3]). Clearly, step quantity and size don’t affect acceptance rate nearly as much as total integration time (Note that for integration times of .075 and .1, the acceptance rate for both time step sizes was exactly the same).](image-url)
5.2.2 Disagreement with prior. We found that in some cases, the chains repeatedly drifted into areas that were invalid according to the prior distribution. For example, the prior set a maximum depth at 50 kilometers, with all depth values greater than the maximum depth being assigned a log-prior of $-\infty$. When starting in certain areas of the domain (typically those regions where the centroid location didn’t match the observational data well), earthquake depths were frequently proposed exceeding 50 kilometers. As the maximum depth was increased, depths were proposed exceeding the maximum depth every time, no matter how deep the maximum was set.

Since the prior used a maximum depth and was thus discontinuous, the posterior distribution should also be discontinuous and so its true gradient is undefined. Hence, we anticipate that the surrogate gradient will perform poorly for deeper earthquakes. This is likely due to the inability of the surrogate gradient to correctly capture the true gradient in all situations, and is a direct result of the discontinuous definition of the prior as mentioned.

Future studies will investigate which centroidal locations yield proposals of this type and how those effects are dependent on the discontinuous nature of the prior. It would also be beneficial to replace the discontinuous prior with a continuous one (since the choices of prior are somewhat arbitrary, this choice can be easily justified). However, in doing this we lose the ability to compare HMC’s performance to that of the method used by [3].

5.2.3 Gradient of arrival time. In an attempt to examine which observation types were influenced by the input earthquake parameters, we excluded the gradient of the arrival times with respect to the observations for a handful of chains. This effectively assumes that the arrival time is a constant function with respect to the input earthquake parameters (as it turns out, this is a rather poor assumption). Under this significantly simplifying assumption, the acceptance rates for these chains nearly vanished unless the Hamiltonian integration time was drastically reduced as well. In practice, the kinetic energy would grow while most components of the sample position would hardly budge. This led to a significant change in the Hamiltonian $H$ from the current sample to the proposal which led to rejecting
the proposed sample almost every time.

While most of the sample parameters being proposed seemed to be relatively unaffected by the change in gradient, the latitude and longitude values being proposed in the absence of information on arrival times, were highly unlikely, i.e., the latitude and longitude appeared to critically depend on the arrival time observations. Interestingly, [7] did not use arrival time gradients in their method but didn’t see this phenomenon (likely because they never completed a base case where the arrival times were used). This indicates that location information is being learned almost exclusively from the arrival time data, and when that piece of the gradient is excluded, the surrogate gradient we are currently using is not accurate enough to compensate for this lack of information.

CHAPTER 6. CONCLUSION AND FUTURE WORK

Despite some of the quirks arising from using a surrogate gradient, we conclude that we have successfully developed an implementation of Hamiltonian Monte Carlo that provides significant speedup over previous methods. At the acceptance rates currently being achieved, this HMC method would allow chains to converge to the posterior distribution while collecting half as many samples as required in [3]. This is a significant speedup and would allow further work to be done on tsunamis spanning much larger areas than previously possible.

In addition, we have uncovered some potential pitfalls arising from using a surrogate gradient on a problem of this complexity. We observed that the acceptance rate of a surrogate gradient HMC scheme is volatile and highly dependent on the chosen integration time. We also observed that a surrogate gradient can lead to misleading proposals, particularly when the posterior distribution is not everywhere continuous.

Additional study should be done on both of these observations—if it were possible to know how the initial starting point of a chain influences how feasible many of the chain’s samples would be, it would allow more careful selection of starting points and therefore a more efficient sampling process. Additionally, a more careful investigation of precisely how
long the HMC scheme can be integrated before the sample begins to fall off the constant Hamiltonian curves would be useful in informing integration time choices in the future. Further work will also consider regularization of the prior distribution in hopes of avoiding the discontinuity enforced by the current setup.


