2021-11-23

Exploring Improvements to the Convergence of Reconstructing Historical Destructive Earthquakes

Kameron Lightheart
Brigham Young University

Exploring Improvements to the Convergence of Reconstructing Historical Destructive Earthquakes

Kameron Lightheart

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
Christopher Grant
Blake Barker

Department of Mathematics

Brigham Young University

ABSTRACT

Exploring Improvements to the Convergence of Reconstructing Historical Destructive Earthquakes

Kameron Lightheart
Department of Mathematics, BYU
Master of Science

Determining risk to human populations due to natural disasters has been a topic of interest in the STEM fields for centuries. Earthquakes and the tsunamis they cause are of particular interest due to their repetition cycles. These cycles can last hundreds of years but we have only had modern measuring instruments for the last century or so which makes analysis difficult. In this document, we explore ways to improve upon an existing method for reconstructing earthquakes from historical accounts of tsunamis. This method was designed and implemented by Jared P Whitehead's research group over the last 5 years. The issue of this method that we address is the relatively slow convergence. One reason for this slow convergence is caused by the random walk proposal step in the Markov Chain Monte Carlo (MCMC) sampling.

We explore ways of constructing an approximate gradient of the model in order to apply a more robust MCMC Method called MALA that uses a gradient combined with some randomness to propose new samples. The types of approximate gradients we explore were a heuristic gradient, a data driven gradient and a gradient of a surrogate model. We chose to use the gradient of a simplified tsunami formula for our implementation. Our MALA algorithm under performed the previous random walk method which we believe implies that the simplified tsunami model didn't give sufficient information to guide the proposed samples in the optimal direction. Further experimentation would be needed to confirm this and we are confident that there are other ways we can improve our convergence as specified in the future work section.

Our method is built into the existing Python package tsunamibayes. It is available, open-source, on GitHub: https://github.com/jwp37/tsunamibayes.

Keywords: earthquake, tsunami, BYU, applied, math, MCMC, MALA

Acknowledgements

First, I would like to thank my advisor, Jared Whitehead, for allowing me to join his research group that piqued my interest a couple years ago. I also want to thank everyone who I have worked with on the tsunami research team, as well as those who came before me. Next I want to express my appreciation to Justin Krometis for his guidance in coming up with an approximate gradient, and learning how to use MCMC methods other than the simple random walk proposal.

Second, I want to thank my wonderful wife, Aubri for her love and support throughout my undergraduate and graduate experiences. I also want to thank my parents for raising me in a way that allowed me to progress to where I am today. In particular my Mom, Wendy Lightheart, for fueling my interest in the field of Mathematics.

Lastly, I would like to thank all the BYU faculty, especially those in the ACME program. It was never easy, but their motivation and superb teaching abilities were a major help in getting to this point.

Contents

List of Tables

List of Figures

Chapter 1. Introduction

Among the many challenges we face in this world, perhaps one of the most heartbreaking is when natural disasters strike. What makes matters even worse is when we don't see it coming. That is why our tsunami research team at BYU has set out to contribute to the effort of modeling and prediction of a subset of these natural disasters, particularly tsunamis. Tsunamis are mainly caused by any one of four things: earthquakes, landslides/seamount collapse, lava entering the sea or meteorite impacts [1]. It is worth noting that there are rare occurrences that have unknown causes. Our main focus has been on tsunamis caused by earthquakes, but we are also planning on looking into some of these other causes, particularly seaside land collapse since it is the second most common cause whereas lava and meteorite impact are much more rare.

There are many geographic locations in this world that are in danger of tsunamis, but the one that we have centered our work around is Indonesia. Indonesia is the most tectonically active region on earth, and it has a highly dense population. Unfortunately this is a setup for mass casualties and destruction when large events strike [2]. This was made evident when the 2004 earthquake hit off the coast of Sumatra, causing a massive tsunami that devastated many neighboring islands and shorelines. This event in particular spurred an enormous research effort and has become the basis of many geological theories and improved understanding due to the well documented reports and seismic data generated from precise instruments. Unfortunately, we haven't always had these seismic measuring tools. There were some primitive instruments, but nothing capable of producing data essential to the modeling and prediction of earthquakes necessary to make informed decisions about preventative measures [3]. To that end, we have pursued a method of determining key characteristics and details about tsunamis and the earthquakes that caused them for events that occurred previous to the invention of precise measuring instruments.

The first event we focused on was the 1852 Banda Arc earthquake due to a large amount of historical writings that give basic details, like how long the tsunami took to arrive after the earthquake hit, how high the waves reached, and how far inland the waves reached [4]. This data is given in very imprecise ways, which makes it very difficult to determine the magnitude of the earthquake or other details about it. Using Bayesian inference, our research group came up with and implemented an algorithm that made valuable discoveries about this event [5, 6].

The bare bones of the algorithm include Bayesian inference, the Metropolis-Hastings Markov Chain Monte Carlo (MCMC) method [14, 15], the Okada Model [16] and the Geoclaw tsunami simulation package [10, 11, 12, 13]. More details on these will be given in the next chapter. While our group was successful in discovering the nature and details of this event within a measure of error, unfortunately it was quite costly computationally. The reason it was so expensive has to do with the random walk implementation of MCMC that was used. Essentially to find the desired set of parameters, a random guess is taken, evaluated against historical details and either accepted or rejected depending on if the guess was an improvement to the previous guess. Normally using a random walk to propose each new set of parameters isn't a problem when used to run a MCMC algorithm, but in our case for each iteration we needed to run a quite expensive simulation (Geoclaw) to simulate a tsunami based on the input parameters which is then compared to the historical accounts we gathered. This led to months of running chains before we could confidently determine that the algorithm converged to a solution. This is not ideal since we want to apply our same method to other tsunamis that occurred at different times and in different places throughout history.

This is where derivative based MCMC methods come into play. The idea is that if we can determine a gradient of the tsunami simulation, instead of randomly guessing at each step, we can step in the right direction each time resulting in a drastic improvement in convergence time. Unfortunately, the simulation package that we use (Geoclaw) can not be used to determine such a gradient. Luckily these derivative based MCMC methods can use an approximation of that gradient and still reap the convergence improvements. On a higher level this is because stepping in roughly the right direction is better than randomly guessing. Some of these derivative based methods are Preconditioned Crank-Nicolson (pCN), Metropolis-Adjusted Langevin (MALA), and Hamiltonian or Hybrid Monte Carlo (HMC) [7, 8, 9]. In this thesis we focus on the MALA method, but future work can pursue other methods as well.

Chapter 2. Overview of Previous Work

2.1 Background

In this section we aim to cover the necessary background topics needed to understand the algorithm, as well as introduce the notation we will use. The main topics of interest here are Bayesian statistics, a forward model, a likelihood distribution and the MCMC algorithm.

Bayesian statistics is one of the main branches of statistics opposite frequentist's theory. Frequentists interpret random variables to not represent a truly random process, but instead to represent a true set of probabilities that can be found via lengthy experimentation. This approach follows the idea that this true set of probabilities exists and with enough samples the experimental results will approach the true probabilities. The focus is on the probability of given data based on a fixed parameter $P(data|parameter)$. The Bayesian approach measures this too, but also asks the question of the probability of a given parameter based on a fixed data set $P(parameter|data)$.

The foundation of Bayesian statistics relies on Bayes theorem which relates these two probabilities in this way:

$$p(x|\theta) = \frac{p(\theta|x)p(x)}{p(\theta)} \tag{2.1}$$

Using the law of total probability and assuming we have continuous random variables we can extend this to:

$$p(x|\theta) = \frac{p(\theta|x)p(x)}{\int p(\theta|x)p(x)dx} \tag{2.2}$$

This formula can also be interpreted in the context of an inversion problem where we have an unknown distribution that we can approximate by starting with an initial guess we call the prior distribution that is represented by $p(x)$, and a likelihood function which is represented by $p(\theta|x)$. The integral in the denominator is known as a normalizing constant that scales the prior times the likelihood. Finally when we compute the right side of the equation the result is called the posterior distribution. Successively computing this formula in an iterative fashion while updating the sample x is the bare-bones of MCMC.

2.2 MCMC

In many situations even though we can compute the prior and likelihoods of a given sample, the normalizing constant is expensive or impossible to compute or even estimate. Luckily there are ways to sample from the posterior without needing to compute it. Some of the most famous of these sampling methods are called Markov chain Monte Carl (MCMC) methods. We focus on the Metropolis-Hastings algorithm, originally developed by Nicholas Metropolis [14], and extended by W.K. Hastings [15].

A Markov chain is defined to be a sequence of random variables X_1,X_2,... where any given state X_n only depends on the previous state X_{n-1} this property is stated as:

$$P(X_{n+1}|X_n = x_n, ..., X_1 = x_1) = P(x_{n+1}|X_n = x_n) \tag{2.3}$$

A stationary distribution $\pi(x)$ of a Markov chain is a probability distribution for that Markov chain that remains unchanged as time progresses. In other words, as time goes to infinity, it is a distribution of how long the chain will spend at each state.

Let $K(x, y)$ be the probability density formula to transition from one state to another. This is called the Markov kernel. We can compute the probability of the next state using this kernel in the following way:

$$P(X_{n+1}|X_n = x) = \int K(x,y)dy + r(x)\chi(x) \tag{2.4}$$

where x is the current state, r(x) is the probability of remaining at the state x, and y is the integrating variable that represents the next state.

Another important definition which combines the stationary distribution and the transition kernel is called detailed balance. Detailed balance ensures that the Markov chain is reversible with respect to the stationary distribution. It holds when the following equation is true:

$$\pi(y)K(y,x) = \pi(x)K(x,y) \tag{2.5}$$

The algorithm for Metropolis-Hastings MCMC is

Algorithm 1 Metropolis Hastings MCMC

```
Initialize: $x_0$
for i=0,... do
    Propose $p \sim q(x_i)$
    Set $\alpha = min(1, \frac{\pi(p)q(p,x_i)}{\pi(x_i)q(x_i,p)})$
    Draw $p \sim U(0,1)$
    if $p < \alpha$ then
        Set $x_{i+1} = p$
    else
        Set $x_{i+1} = x_i$
```

Our problem requires a Bayesian inversion implementation of MCMC where we have a prior, likelihood and forward model as the three main components. The main difference is in how we decide whether or not to accept a given proposal sample. We use a predetermined likelihood function to help make this decision. The likelihood in our case is a function that computes how closely a set of outputs (wave height, arrival time, inundation distance) match the historical accounts we have. The forward model is a function that takes in the Okada parameters as input (latitude, longitude, magnitude, delta log-length, delta log-width, depth offset) and outputs wave height, arrival time and inundation. This is implemented using the

Geoclaw tsunami simulation model we mentioned before [10, 11, 12, 13]. Finally the prior encapsulates both general knowledge of earthquakes/tsunamis as well as knowledge about the specific region around the Banda Arc. We will define the distributions and parameters chosen for the prior and likelihood in the MALA implementation of the MCMC section.

2.3 Previous Findings

After years of building and adjusting the implementation of this algorithm, we began running chains to gather samples in hopes of converging to a solution. We started several chains along the Banda Arc as the epicenter of the earthquake, and with varying starting magnitudes to try to get an idea of where to focus our efforts. We made some interesting discoveries and re-sampled at the parameters that the chain were approaching.

One interesting finding was that all the chains that we started along the southern half of the arc never made their way North, even though the samples in the North had a higher likelihood. For this reason, we decided to re-sample in the North. The magnitude was clearly tending toward 8.8, so we re-sampled there too. After a total of 168,000 samples collected, we felt confident that our algorithm had converged. This is validated by the Gelman-Rubin diagnostic which settled just below 1.1 as shown in Figure 2.1.

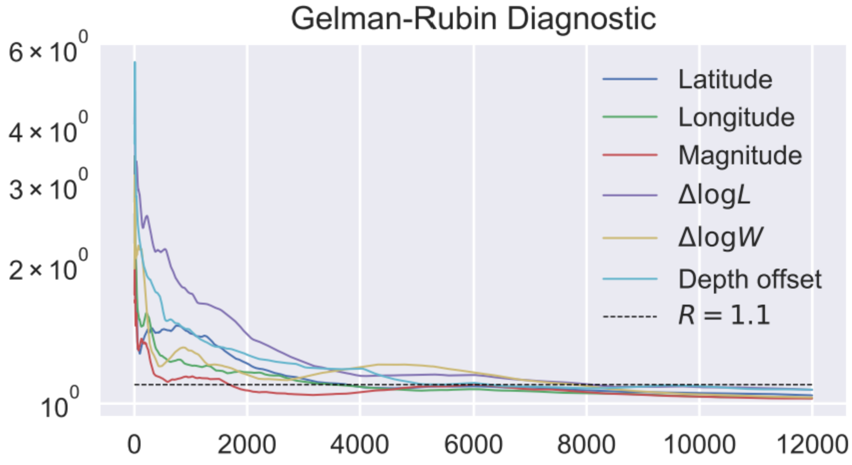


Figure 2.1: Gelman-Rubin Diagnostic Plot of Original Random Walk Chains

The region of convergence settled around latitude -4.5, longitude 131.8, magnitude 8.8. The latitude and longitude results were as expected based on discussions with geologists, but the magnitude ended up being higher than expected. This gave us confidence that our setup could extend to other historical events that we can discover new facts about.

Chapter 3. Construction of an Approximate Gradient

3.1 Heuristic Gradient

Within the realm of earthquakes and tsunamis there are some logical inferences we can make that can be useful for approximating our gradient. The simplest such inference might be the arrival time of a tsunami wave since we would expect that time to decrease the closer the epicenter of the earthquake is to the observation point. Thus we start with a simple equation of:

$$A(x_1, y_1, x_2, y_2) = \frac{d((x_1, y_1), (x_2, y_2))}{z} \tag{3.1}$$

where x_1, y_1 are the latitude and longitude of the epicenter and x_2, y_2 are the latitude and longitude of the shoreline of interest, d is euclidean distance, z is the wave velocity, and A represents the arrival time. For wave velocity we can either choose a constant from domain knowledge, or have it be a function of other model parameters like magnitude, etc.

This should work since the units on distance are a length such as meters, and wave velocity will have units of meters per second. Thus as long as our wave velocity is reasonable, this will yield a first order approximation. We did consider the fact that some of the gauge locations are inside of bays or have pieces of land between them and the epicenter. To account for this, we could have introduced another geographic point between the gauge and epicenter, and measure the distance as point-wise from one point to another. Another option would be to draw a smooth contour in a path similar to what we might expect a tsunami wave to take. Both are improvements that could be pursued, but we expect that other assumptions below will indroduce just as much error, and so we opted to take the simplest route.

Another consideration is to have arrival time be a function of magnitude, i.e. the higher the magnitude, the shorter the arrival time. This one is more complicated though, because it would require some sort of scaling to make sure any magnitude below 10 has an above 0 arrival time. This could be done by applying domain knowledge about how fast tsunami waves are based on an earthquake's magnitude, and reusing the formula above, but making it a function of magnitude only, with a fixed epicenter, and the magnitude would determine the wave velocity. In fact, combining these 2 ideas might be the best approach as we could then take a gradient with respect to latitude, longitude and magnitude with reference to the wave arrival time.

3.2 Linear Fit to Data

Due to our background in machine learning, and the large amount of data we had already collected, we hoped that there could be something done with the data from previously run models, in order to imitate the complex and expensive Geoclaw simulations. Since we had hundreds of thousands of data points from previous random walk MCMC runs, we started there. The simplest approach would be a linear fit to the data, so that is what we started with. The idea was to graph different input parameters against wave height and arrival time, and try to find if there are any parameters that have a good linear fit. If they aren't linear, then other simple curves could be used like an exponential, logarithmic or polynomial fit. The idea is that if any parameters have a differentiable curve that approximates them well, we can use that fitting function in the gradient approximation.

We plotted all the input parameters against the output parameters and did linear regression plots for each of them. Most of the plots were clearly not linear and by narrowing down to the plots with the highest R-Squared coefficients Figure 3.1 shows the top 6 fits. The latitude vs Pulu Ai height, latitude vs Banda Neira height and magnitude vs Pulu Ai height look the best out of these, but they still aren't by any means linear. In fact most of the plots were too spread out to be approximated by any simple curve let alone a linear fit. Therefore we can conclude that a curve fit won't help with any of our parameters and we can move on to other possibilities.

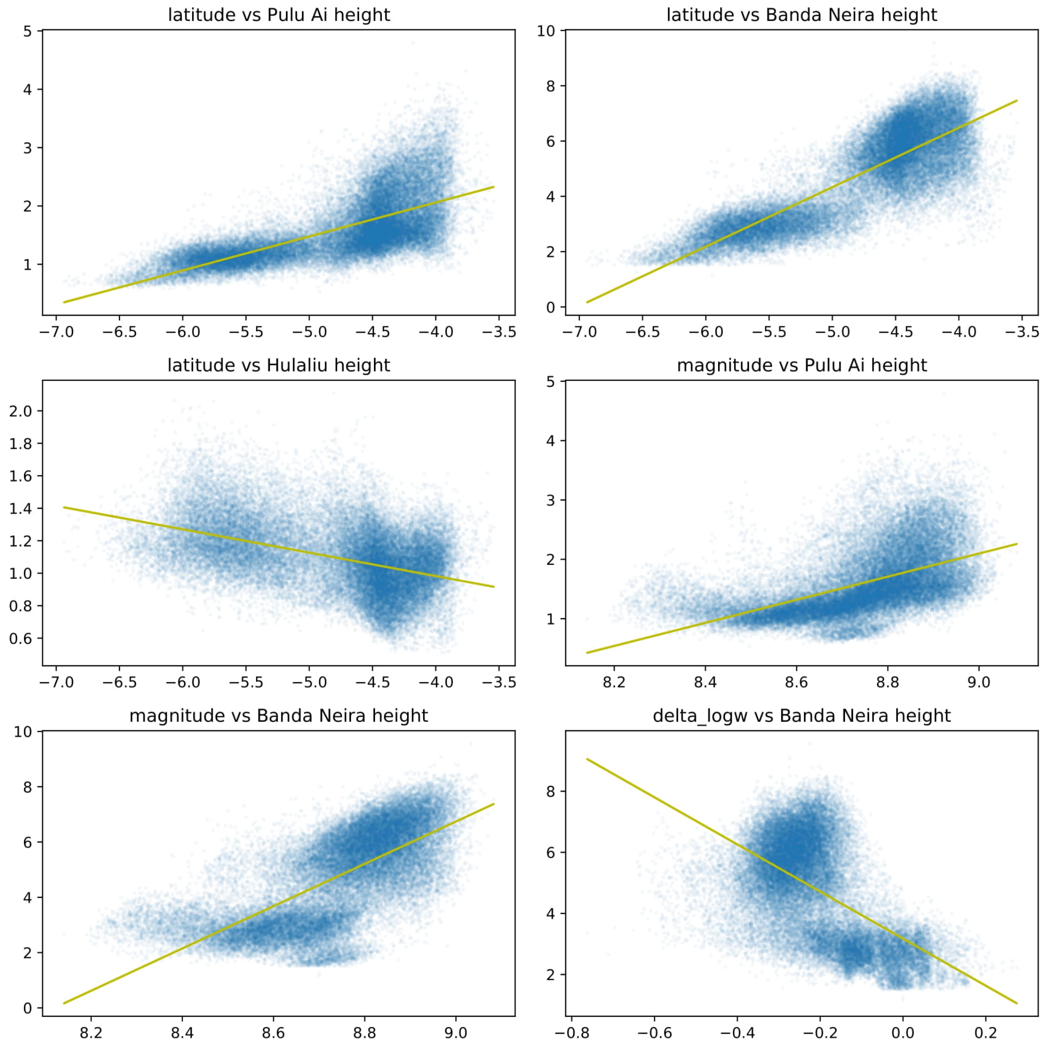


Figure 3.1: Best linear fit Okada parameters vs observation plots

3.3 Data driven Gradient

The next step after curve fitting was to try training some machine learning models on the data. The models that best fit the format and behavior of the data were stochastic gradient descent regressor, Deep Neural multi-layer perceptron regressor, support vector regressor, K-nearest neighbors regressor, and the random forest regressor. As Table 3.1 shows, the results were pretty scattered, but the random forest regressor did the best job impersonating the forward model. The problem is that there isn't a good way to compute a gradient on a random forest which is fundamentally built on non-differentiable functions.

Model Type	Score $(-\infty, 1]$	MSE	Train Time
Random Forest	0.92	0.05	13.8 s
K-Nearest Neighbors	0.58	0.24	0.04 s
MLP Neural Network	0.38	0.27	10.8 s
Suport Vector Machine	-0.33	0.42	13.5 s
Stochastic Gradient Descent	-∞	∞	0.97 s

Table 3.1: Data driven forward model replacement results

3.4 Simplified Formula

One idea to overcoming the dilemma of not being able to take a gradient of the forward model is to replace it with a simplified formula that takes the same parameters and gives the same result, just with much less accuracy. This idea came to us because we were in discussions with Steve Ward who has done previous research and published papers that put together such a formula [17]. The nice thing about derivative based MCMC methods is that we only need to approximate the derivative, so as long as such a formula gives us more valuable information than a random guess, it will be an improvement over random walk.

The formula taken from Steve Ward's research that we used is the shoaled tsunami amplitude $A_S(R)$ (essentially wave height) at a shore location R distance away from the epicenter of an earthquake which is given by

$$A_S(R) = A_0 P S_L \tag{3.2}$$

where A_0 is the initial tsunami amplitude, P is the propagation loss and S_L is the shoaling correction. The computation of these three forms is quite involved, but the only non-constant variables used are magnitude M; the ocean depth at the source H_0; the distance from the epicenter R; and the mean ocean depth $\bar{H}$.

Let's start breaking this down by first approximating A_0 or the initial tsunami height by the formula:

$$A_0 = \frac{\alpha \nabla u(M)}{cosh\left[\frac{4\pi H_0}{W(M)+L(M)}\right]} \tag{3.3}$$

where α is the fraction of earthquake slip that transforms into tsunami-making uplift calculated using $\theta =$ fault dip and $\varphi =$ fault rake angles given in degrees. It is given by:

$$\alpha = \left(1 - \frac{\theta}{180}\right) sin(\theta)|sin(\varphi)| \tag{3.4}$$

$\nabla u(M)$ is the fault slip which is a function of magnitude M, W(M) and L(M) are the fault width and length respectively which are also functions of M.

Next we define propagation loss which is the wave decay that tsunamis face as they travel due to geometrical spreading and frequency dispersion. It is a function of R, L(M) and $\bar{H}$ given by

$$P = \left(1 + \frac{2R}{L(M)}\right)^{-\psi} \tag{3.5}$$

where $\psi = 0.5 + 0.575e^{-0.0175\frac{L(M)}{\bar{H}}}$.

Finally S_L is the shoaling correction which reduces to Green's Law:

$$S_L = \left(\frac{H_0}{\bar{H}}\right)^{\frac{1}{4}} \tag{3.6}$$

Combining 3.3, 3.5 and 3.6 together gives our combined simplified wave height equation to be:

$$A_s(R) = A_0 P S_L = \frac{\alpha \nabla u(M)}{cosh\left[\frac{4\pi H_0}{W(M)+L(M)}\right]} \left(1 + \frac{2R}{L(M)}\right)^{-\psi} \left[\frac{H_0}{H}\right]^{\frac{1}{4}} \tag{3.7}$$

This is great, but the input parameters to the Okada model [16] we use in our algorithm are slightly different, so let's talk about how to go between the two. As a reminder, the parameters used in our model are latitude and longitude of the epicenter, change in log length and log width and the depth offset of the fault. From this point forward we will use the variables $l_1, l_2, m, d_{ll}, d_{lw}, d_o$ for latitude, longitude, magnitude, delta log-length, delta log-width and depth offset respectively. Equation (3.8) gives a breakdown of each of the variables from the simplified formula using these 5 parameters. Haversine is the well known formula for computing the distance between two points on a sphere (the earth in this case), dip_map is a mesh grid that maps from latitude and longitude to the dip angle of the fault [19], and μ is a scaling factor we set to 4e10. For simplicity we set the height variables H and $\bar{H}$ as $H = 1$ and $\bar{H} = 2,000$ while H_0 is looked up using another mesh grid of data we have called the depth_map [19].

$$
\begin{gathered}
R = Haversine(l_1, l_2) \\
\theta = dip\_map(l_1, l_2), \quad \varphi = 90 \\
\nabla u(m) = 10^{1.5*m+9.05-log(\mu*L(m)*W(m))} \qquad (3.8) \\
W(m) = 10^{0.2992*m+2.60873+d_{lw}} \\
L(m) = 10^{0.52339*m+1.09744+d_{ll}}
\end{gathered}
$$

Thus the simplified tsunami formula can be written as a function of l_1, l_2, m, d_{ll}, d_{lw} and d_o (depth offset isn't explicitly used, and is not important enough to involve, thus we ignore it when taking derivatives and instead will update it using a random walk, more on that later).

Now that we have a simplified tsunami equation defined, we proceed to compute the gradient of it analytically. This is so that we can compute a direction to go at each step of our Markov chain, rather than stepping in a random direction. While it is possible to take these derivatives by hand, the faster approach was to use symbolic programming to build the equation and take subsequent derivatives. Further details will be covered after we introduce MALA.

Chapter 4. MALA

The first derivative based MCMC algorithm we implemented was the Metropolis-adjusted Langevin Algorithm (MALA) [9]. This algorithm uses the same basic structure and steps presented in the MCMC section of the background chapter above, except for the proposal and accept-reject are implemented using Langevin dynamics. Langevin dynamics involve modeling molecular systems using simplified mathematical models that account for omitted degrees of freedom through stochastic differential equations. They can be used for MCMC because they involve both a gradient flow toward higher probability in a given region and a random walk portion which together gives the convergence properties needed.

In random walk MCMC the proposal step uses a sum of the negative log-likelihood and the negative log-prior. We are going to define this function as U(q) where q is the current sample. The new proposal is:

$$\tilde{q} = q - \frac{1}{2}\delta^2 \nabla U(q) + \delta\nu \tag{4.1}$$

where ν is drawn from the unit normal distribution, $\delta << 0$ is the step size which is typically small and ∇ is the gradient operator. This gives us our proposal $\tilde{q}$ which is then passed into the new acceptance probability formula which as required by detailed balance is given by:

$$\alpha(q, \tilde{q}) := 1 \wedge exp(-U(\tilde{q}) - \frac{1}{2\delta^2}|q - \tilde{q} + \frac{\delta^2}{2}\nabla U(\tilde{q})|^2 + U(q) + \frac{1}{2\delta^2}|\tilde{q} - q + \frac{\delta^2}{2}\nabla U(q)|^2). \tag{4.2}$$

Using this proposal and acceptance probability, it has been proven that we obtain a reversible Markov kernel and thus the MCMC algorithm will converge.

The reason this method is useful to us is because we only have an approximation of the gradient, but since there is a random term involved in Langevin dynamics, along with the ability of the algorithm to reject samples that worsen the probability distribution, even with a exact opposite value for the gradient, the chain will still work its way toward the correct solution over time.

4.1 MALA Applied to our Tsunami Problem

As stated in the previous section, we have that U(q) is the sum of the log-likelihood and the log-prior. In the context of our problem the likelihood equation for a given sample q is determined by a set of wave height distributions that were constructed with the help of the colleagues in Geology. These height distributions vary for each observation location so we will outline those here:

Gauge Location	Distribution	Mean	Standard Deviation	Shape Parameter (df)
Amahai	Normal	3.5	1.0	N/A
Ambon	Normal	1.8	0.4	N/A
Ameth	Normal	3.0	1.0	N/A
Banda Neira	Normal	6.5	1.5	N/A
Buru	Chi-square	0.5	1.5	1.01
Hulaliu	Chi-square	0.5	2.0	1.01
Kulur	Normal	3.0	1.0	N/A
Pulu Ai	Normal	3.0	0.8	N/A
Saparua	Normal	5.0	1.0	N/A

Table 4.1: Wave height prior distributions by gauge location

where the Normal and Chi distributions are defined as:

$$N(\mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{1}{2}\left(\frac{h-\mu}{\sigma}\right)^2} \tag{4.3}$$

$$Chi(\mu, \sigma, df) = \frac{1}{2^{\frac{df}{2}}\Gamma\left(\frac{df}{2}\right)} \left(\frac{h-\mu}{\sigma}\right)^{\frac{df}{2}-1} e^{\frac{-h}{2}} \tag{4.4}$$

and where h is the wave height at the shoreline.

Next we substitute our simplified tsunami wave height equation in for h to create a composite function since $A_S(M)$ defined above is equivalent to wave height. Thus given an input of latitude, longitude, magnitude, delta log-length, delta log-width and depth offset, these equations with given parameters per gauge give the probability (or likelihood) that those 5 parameters could have caused the wave that hit that particular gauge location. The combination of them gives us a picture of how likely a set of 5 parameters could have caused the 1852 tsunami.

Now for the prior, we again use domain knowledge with the help of colleagues in Geology to come up with prior distributions for latitude, longitude, magnitude, delta log-length, delta log-width and depth offset. These are essentially educated guesses of what reasonable values for each should be. Here is a table of the prior distributions.

Parameter	Distribution	Mean	Standard Deviation	Shape
Depth	Truncated Normal	$\mu_d = 30000$	$\sigma_d = 5000$	N/A
Magnitude	Truncated Exponential	$\mu_m = 6.5$	$\sigma_m = 1.0$	$b_m = 3.0$
Delta-logl	Normal	$\mu_{\delta logl} = 0.0$	$\sigma_{\delta logl} = 0.188$	N/A
Delta-logw	Normal	$\mu_{\delta logw} = 0.0$	$\sigma_{\delta logl} = 0.172$	N/A
Depth-offset	Normal	$\mu_{d_o} = 0.0$	$\sigma_{d_o} = 5.0$	N/A

Table 4.2: Okada model parameter prior distributions

Here the truncated normal is the normal distribution, but zero everywhere outside of a given interval [c,d], We fix these values to be

$$c = \frac{mindepth - \mu_d}{\sigma_d}, \quad d = \frac{maxdepth - \mu_d}{\sigma_d} \tag{4.5}$$

where we fix mindepth=2500 and maxdepth=50000 and μ_d, σ_d are the mean and standard deviation for depth given in the table above. The truncated exponential distribution is

$$truncexpon(m, b_m, \mu_m, \sigma_m) = \frac{e^{\frac{-(m-\mu_m)}{\sigma_m}}}{1 - e^{-b_m}} \tag{4.6}$$

You might notice that depth is in place of latitude and longitude from our original 6 parameters. In order to compute the gradient in terms of latitude and longitude, we can use this equation to relate depth to latitude and longitude:

$$depth = depth\_map(l_1, l_2) + 1000 * d_o \tag{4.7}$$

where depth-map is a grid of depths within a region taken from SLAB2 dataset, in this case a grid that contains all the gauge locations, the region of plausible epicenters and a buffer outside of all those. We aren't able to compute an analytic derivative, so instead we use a finite difference method for computing the derivatives with respect to latitude and longitude, and analytically solve for the other parameters. We use the centered difference schemes

$$\frac{ddepth}{dl_1}(l_1, l_2) = \frac{1}{2 * step}(depth\_map(lat + step, l_2) - depth\_map(lat - step, l_2)) \tag{4.8}$$

$$\frac{ddepth}{dl_2}(l_1, l_2) = \frac{1}{2 * step}(depth\_map(l_1, l_2 + step) - depth\_map(l_1, l_2 - step)) \tag{4.9}$$

Now the combined prior is:

$$prior(l_1, l_2, m, d_{ll}, d_{lw}, d_o) = latlon\_prior * m\_prior * d_{ll}\_prior * d_{lw}\_prior * d_o\_prior \quad (4.10)$$

We are interested in the negative log-prior so we end up with:

$$\begin{aligned} -log(prior(l_1, l_2, m, d_{ll}, d_{lw}, d_o)) = &- log(depth\_prior(l_1, l_2)) \\ &- log(m\_prior) \\ &- log(d_{ll}\_prior) \\ &- log(d_{lw}\_prior) \\ &- log(d_o\_prior) \end{aligned} \quad (4.11)$$

Thus when we take the gradient of the negative log-prior, the derivatives distribute over the subtraction and thus we can take the derivatives individually without worrying about the product rule. Thus our gradient is:

$$-\nabla log(prior) = \left[\frac{dprior}{dl_1}, \frac{dprior}{dl_2}, \frac{dprior}{dm}, \frac{dprior}{d\delta log\_l}, \frac{dprior}{d\delta log\_w}, \frac{dprior}{dd_o}\right] \quad (4.12)$$

Conveniently all of our prior distributions involve an exponential, and since the logarithm is the inverse of the exponential, the distributions simplify quite nicely and it can be easily verified that the derivatives are:

$$\frac{dprior}{dl_1}(l_1, l_2, d_o) = \frac{(\mu_d - (depth\_map(l_1, l_2, step) + 1000 * d_o)) * \frac{ddepth}{dl_1}}{\sigma_d} \quad (4.13)$$

$$\frac{dprior}{dl_2}(l_1, l_2, d_o) = \frac{(\mu_d - (depth\_map(l_1, l_2, step) + 1000 * d_o)) * \frac{ddepth}{dl_2}}{\sigma_d} \quad (4.14)$$

$$\frac{dprior}{dm}(m) = 1 \quad (4.15)$$

$$\frac{dprior}{d\delta log\_l} = \frac{\delta log\_l}{\mu^2_{\delta log\_l}} \quad (4.16)$$

$$\frac{dprior}{d\delta log\_w} = \frac{\delta log\_w}{\mu^2_{\delta log\_w}} \quad (4.17)$$

$$\frac{dprior}{dd_o} = \frac{(\mu_d - (depth\_map(l_1, l_2, step) + 1000 * d_o))}{\sigma_{d_o}} + \frac{d_o}{\sigma_{d_o}} \quad (4.18)$$

With that, we now have all the pieces necessary to compute the gradient of U(q)

$$\nabla U(q) = -\nabla log(likelihood(q)) - \nabla log(prior(q)) \tag{4.19}$$

and we can now compute the proposal and acceptance probability for our MCMC algorithm.

Chapter 5. Experimentation and Results

5.1 Initial MALA Tests and Delta Search

When it comes time to run MALA, the value of delta must be chosen between 0 and 1. Delta is the step size of both the gradient and the random portion of the proposal. As with many tunable parameters in various algorithms, there usually isn't a predetermined way of choosing delta. Most of the time the most efficient way of choosing those parameters is to run experiments with different combinations and keep the ones that perform the best. That is why we started several different chains with varying values of delta, and varying Okada parameters. We choose 0.3, 0.1, 0.01, 0.001, 0.0001 as our values of delta and had 3 different starting locations of the North-West, South and North-East areas of the Banda Arc. The North-West and South locations were chosen to see if the gradient would help those chains make their way over to the posterior distribution that we know is found in the North-East area near -4.5 latitude, 131.5 longitude. The Southern region is of particular interest since previous experiments never made their way North due to a local minimum of the negative log-likelihood. The hope was that the gradient might help the chains jump their way out of the local minimum and make their way towards the global minimum. The chains we started near previously identified maximal posterior from the random walk were chosen to start where the re-sampling had started during the random walk experiments. These 15 chains were run for between 9,000 and 11,000 samples each, and some interesting results came from them.

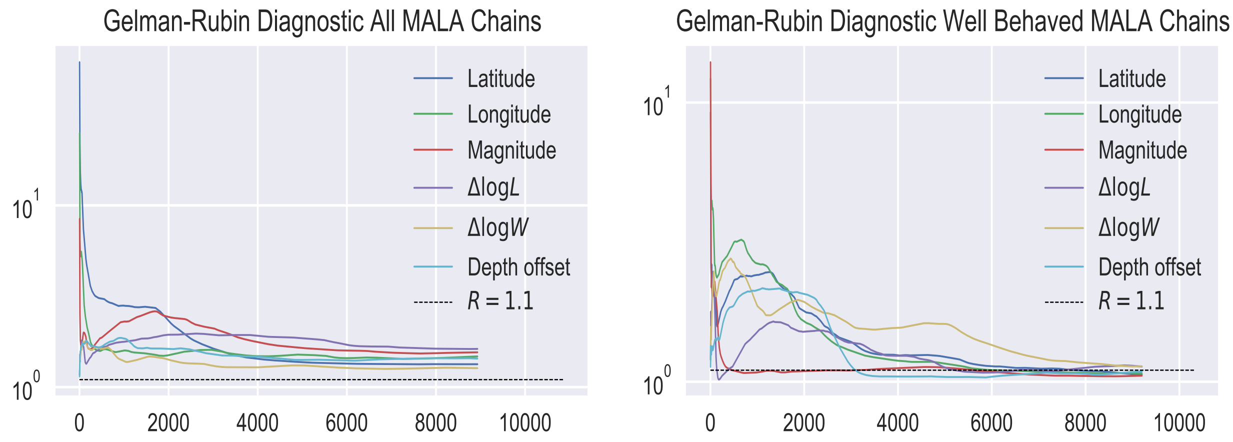
One important metric of any MCMC method is the acceptance rate of samples. For random walk, there is a general rule of 23% being the optimal acceptance ratio. For MALA, the general consensus is that an acceptance rate closer to 50% is ideal, but this is really a rule of thumb, and has not been rigorously justified. In the table below, we list the average acceptance ratio for each of the 15 chains. It is worth noting that starting the alternate locations at a magnitude of 8.0 was likely not ideal since that is well outside of the posterior distribution found from previous work. This is adjusted in the second run of chains in the next section.

Delta	Latitude	Longitude	Magnitude	Acceptance Ratio
0.0001	-4.4	131.8	8.5	0.202
0.0001	-3.85	130.9	8.0	0.198
0.0001	-7.75	130.0	8.0	0.417
0.001	-4.4	131.8	8.5	0.096
0.001	-3.85	130.9	8.0	0.705
0.001	-7.75	130.0	8.0	0.4
0.01	-4.4	131.8	8.5	0.248
0.01	-3.85	130.9	8.0	0.234
0.01	-7.75	130.0	8.0	0.43
0.1	-4.4	131.8	8.5	0.229
0.1	-3.85	130.9	8.0	0.048
0.1	-7.75	130.0	8.0	0.394
0.3	-4.4	131.8	8.5	0.03
0.3	-3.85	130.9	8.0	0.17
0.3	-7.75	130.0	8.0	0.15

Table 5.1: MALA delta parameter search results

The first observation is that the acceptance ratio varied wildly throughout these tests ranging from 0.03 to 0.705. There were 2 values that were close to 0 and both of them were with a larger delta of 0.1 and 0.3 which likely means those values of delta are too high and lead to instability. When the step size is too large, proposals can jump into unlikely samples repeatedly. An expected result is that the acceptance ratio of those samples starting in the Northwest and South were greater than or equal to those of the Northeast. This is because the Northeast location is already near the region of maximal posterior probability, so moving around near there leads to small if any improvements in the log-likelihood. The concerning finding is that none of the delta values could consistently a get higher than 20% acceptance rate. Based on these acceptance rates, along with some findings we'll discuss in the next few paragraphs led to us choosing 0.0001 as our delta value for our direct comparison to random walk.

One important metric for convergence of MCMC chains is the Gelman-Rubin diagnostic [18]. It was shown previously that our random walk method converged based on the Gelman-Rubin diagnostic scores reaching below 1.1 for all of the parameters at around 8000 samples. Figure 5.1a shows a plot of the Gelman-Rubin scores for all the MALA chains of this initial experiment. Although the values decrease for the first 8000 samples, they appear to level out well above the desired score of 1.1. This is expected since the Southern chains never made their way North to the maximum posterior probability region. Figure 5.1b shows the same plot omitting the Southern, near-zero acceptance, and poorly behaved chains. Now the chains appear to converge by around 6000 samples, other than the delta-log width which for some reason takes until roughly 9000 samples to converge.



(a)

(b)

Figure 5.1: Gelman-Rubin diagnostic test for delta parameter search chains

Next we consider the latitude, longitude and magnitude plots of the subset of chains described in the previous paragraph. See Figure 5.2. We can see from where the chains level out that they appear to converge to approximately latitude -4.5, longitude 131.6 and magnitude 8.8. This almost exactly matches up to our previous findings. This is good news as it means that our implementation of MALA is at the very least comparable to the random walk implementation in both the rate of convergence and the location of the maximal posterior probability. We mentioned how we started some chains with a magnitude of 8.0, which was probably a bad decision since the convergence zone centers tightly around 8.8 in magnitude, but notice that those chains were able to shoot up to near 8.8 within only a few hundred samples. A similar result can be found in the longitude plot of the chains that started in the northeast corridor. It took quite a bit longer with close to 2000-3000 samples to get to the maximum probability region, but notice how there were big jumps up at around 1900 and 2200 samples. This could be an indication of the gradient helping to point the proposals in the right direction and causing faster convergence. It will require more experiments to determine how much the approximate gradient is actually helping compared to random proposals.

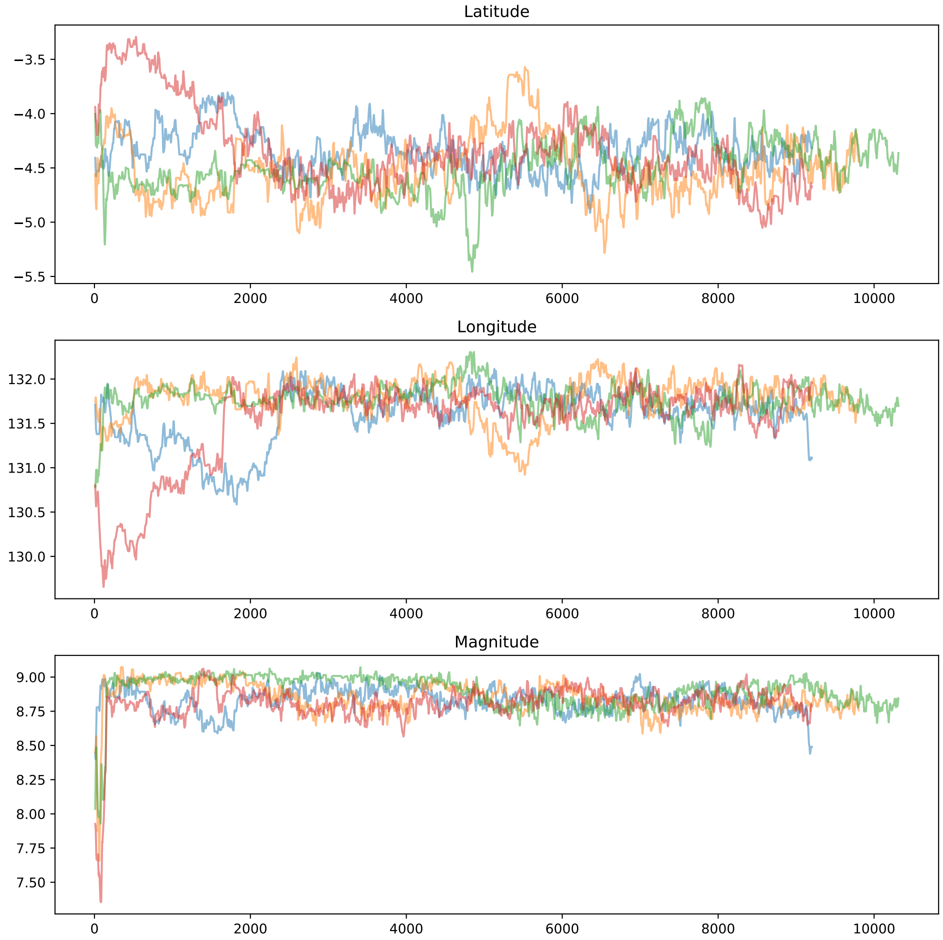
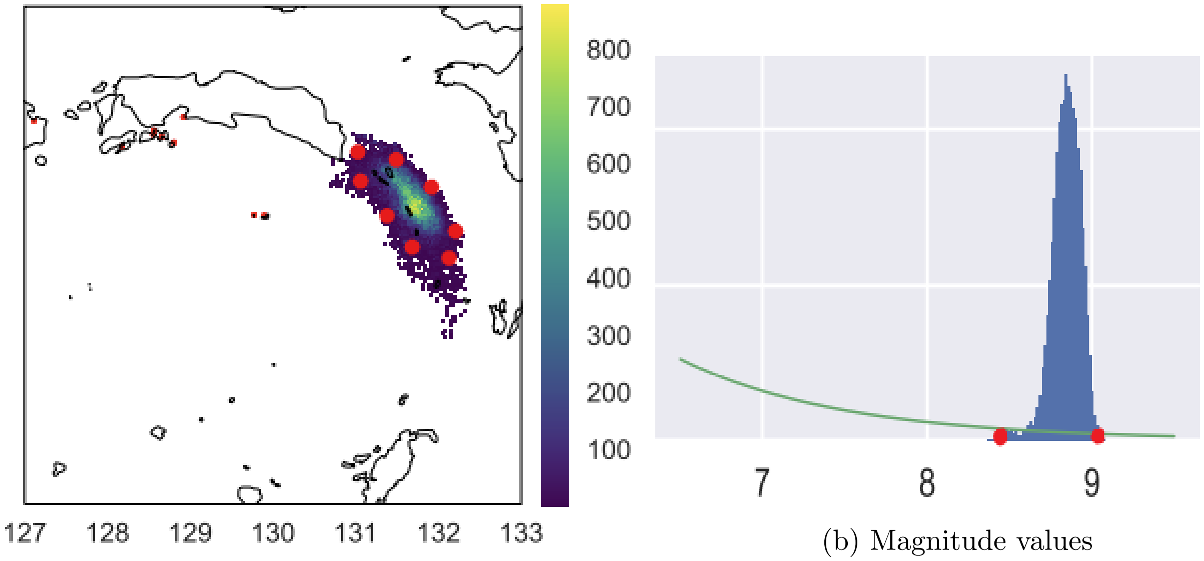


Figure 5.2: MALA experiment 1 latitude, longitude and magnitude graphs

5.2 MALA vs Random Walk Side-by-side Comparison

Now that we have established similar behavior, we hope to discover if approximate gradient MALA can outperform random walk. To do this we designed an experiment to directly compare the two algorithms. The setup of this experiment will consist of choosing some starting epicenters and magnitudes on the edges of the maximum probability region and then running chains for both algorithms simultaneously from the same starting parameters. Starting near the maximum probability region should remove some of the erratic behavior we saw with our first experiment. Since the only values of delta that didn't have a near zero acceptance rate were 0.0001 and 0.01, we decided to use the more conservative 0.0001 value of delta for our direct comparison. This decision was based on our observation that the bigger values of delta tended to have poor behaving chains in our first experiment.



(a) Epicenter locations

(b) Magnitude values

Figure 5.3: Starting parameters for MALA vs random walk direct comparison experiment

The red dots are the starting locations, and the heat map (latitude and longitude) and histogram (magnitude) are from the previously computed data set via random walk.

The next step was to choose the starting epicenters and magnitudes. This was easy for the magnitudes since the plot is 2d so we simply chose the tail ends of the distribution at 8.5 and 9.0 as shown in Figure 5.2. The process for choosing the epicenter starting points was less scientific due to the shape of the convergence region as shown in Figure 5.3b. We settled on 8 different epicenter points listed in Table 5.2 along with the two magnitude edge cases which should give us enough variety and structure for a good comparison experiment. For the remaining parameters of depth offset, delta-log length and width we left them at 0.0 for simplicity.

Latitude	-4.9	-4.9	-4.55	-4.2	-3.9	-3.92	-4.2	-4.6	-4.5	-4.5
Longitude	132.0	131.65	131.4	131.1	131.1	131.5	131.8	132	131.8	131.8
Magnitude	8.8	8.8	8.8	8.8	8.8	8.8	8.8	8.8	8.5	9.0

Table 5.2: Starting parameters for MALA vs random walk direct comparison experiment

Our first thought was to compare the Gelman-Rubin diagnostic between the two since both algorithms were started in the exact same places. See Figure 5.4 The point at which all the parameters crossed below 1.1 was roughly 16,000 for MALA and 13,000 for random walk. Using this metric alone, it would appear that random walk performed better than MALA. It is interesting that the MALA graph appears to shoot down fast within the first 500 samples, before then jumping back up and gradually going down whereas the random walk took until 5000 samples before hitting a steady decline. This could indicate that our implementation of MALA does a better job of getting close to the posterior region, but struggles to converge once it gets there whereas the random walk might take longer to wander near the posterior region, but performs better once it gets close. Essentially, our approximate gradient works well to get in the right neighborhood, but performs poorly for small changes/adjustments.

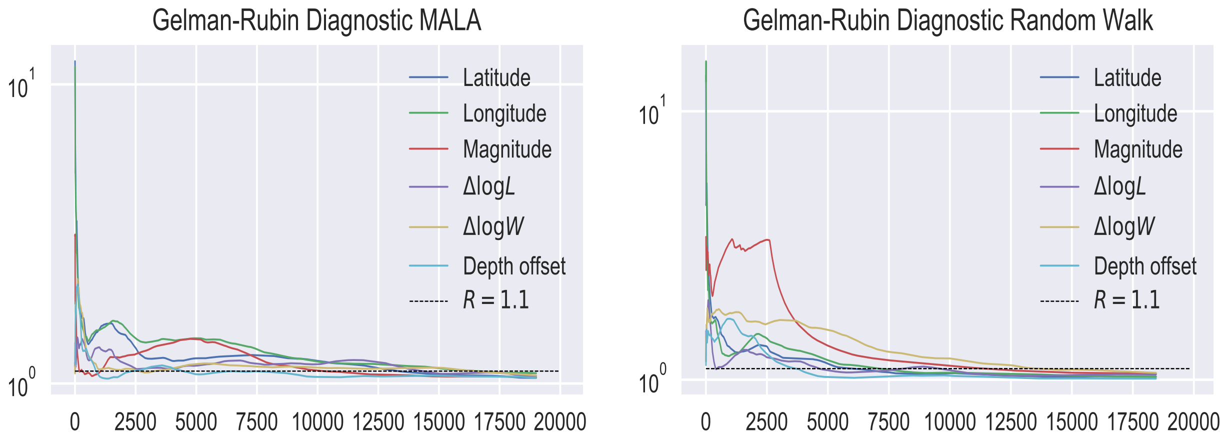


Figure 5.4: Gelman-Rubin diagnostic plots for MALA vs random walk experiment

Another thing to note is that the average acceptance rate for all the MALA chains was 21.8% while the random walk chains averaged 23.9%. This accounts for part of why MALA took longer to converge since it took more samples for MALA to accept proposals on average. This is also disappointing since we would typically like to see a higher acceptance ratio with a gradient method of MCMC since it should propose samples that would improve the likelihood. We think this means that our simplified tsunami formula isn't giving good enough information for the gradient to help. In fact it is making things worse than randomly proposing as we can see from Figure 5.5. The random walk chains stayed consistently above MALA in acceptance ratio hovering above 25% until around 5,000 samples and then slowly declining towards 23%. The MALA chains rather quickly dropped close to 22% and reached 20% by sample 3,000 and sat there until about 10,000 samples. The odd thing is how MALA manages to increase in acceptance ratio slowly from 10,000 to 18,000 samples. Sadly it still didn't surpass random walk at the end and even if it did, the chains had converged before this point so it wouldn't matter in this comparison.

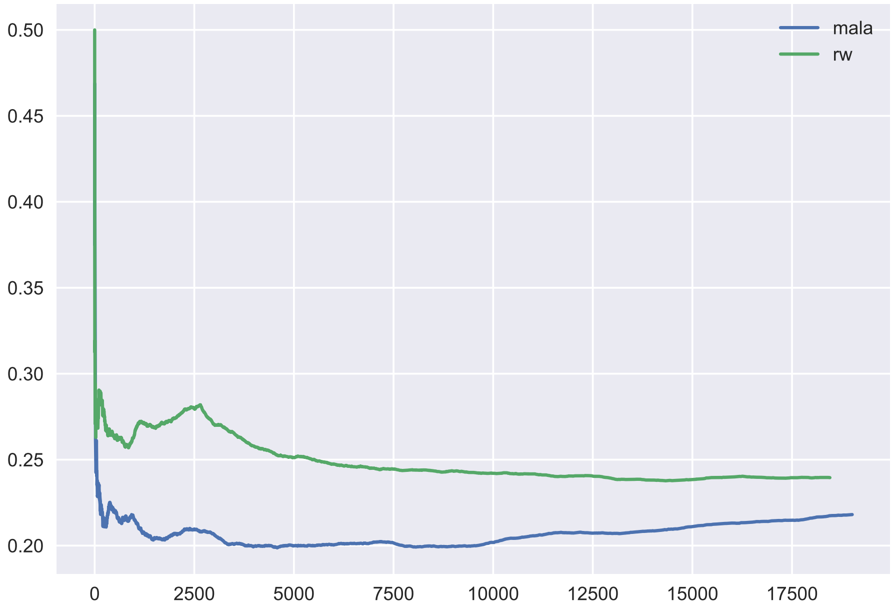


Figure 5.5: Acceptance Ratio Plot of MALA vs Random Walk

The final plot we will discuss are the resulting distributions of samples for MALA and random walk shown in Figure 5.6. These plots show that MALA and random walk found very similar posterior distributions. Random walk appears to have lower variance when compared to MALA in the latitude, longitude, magnitude and delta-log width parameters. This might be because MALA found some local maxima as indicated by the small secondary peak at around -3.7 latitude, the taller tail near 131.0 longitude and the peaks on either tail end of the delta-log width. This could indicate that our implementation of MALA with the set delta parameter we chose is more prone to getting stuck in local maximums than the random walk implementation, or MALA may have identified alternative regions of the parameter space that the random walk missed.

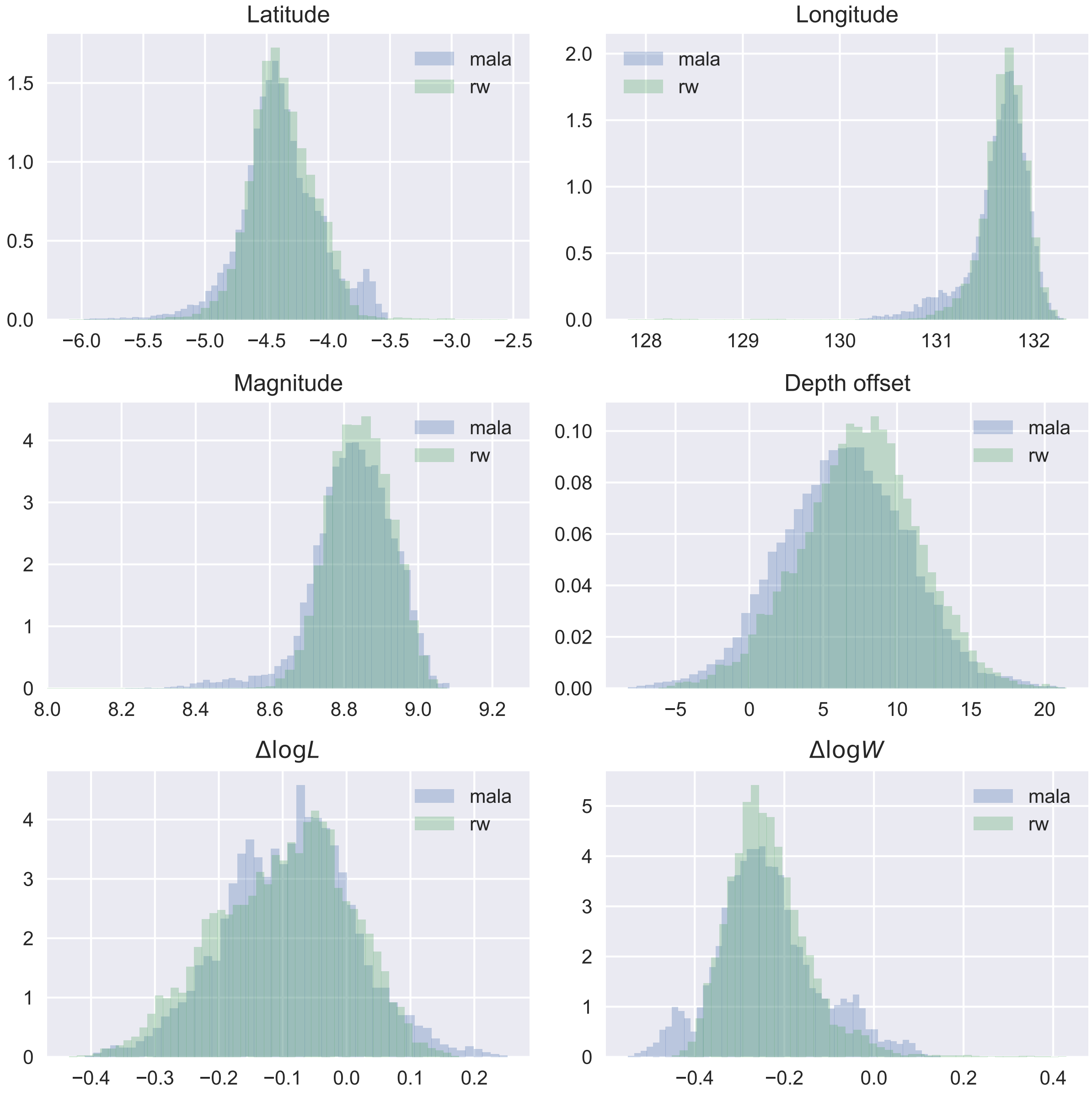


Figure 5.6: Parameter Distribution Plots of MALA vs Random Walk

Chapter 6. Conclusion and Future Work

6.1 Conclusion

We have explored many methods of approximating the gradient of our forward model including curve fitting, data driven models, heuristic models and simplified tsunami formulas. The curve fitting was clearly not viable since none of the mapping from input parameters to end results had shapes that could be approximated by a curve. Data driven models seemed promising, but not for a gradient method specifically since the best fit model wasn't differentiable. Heuristic models were either too simple that the gradient was meaningless, or would require much more research and domain knowledge to construct. This all led us to a simplified model that was built by an expert that we hoped would make a good stand in for our forward model. We knew that this simplified tsunami formula would not be able to replace the forward model entirely due to its oversimplifications, but our hope was that the gradient of this surrogate model would be useful in pushing our proposals in the right direction.

We chose to implement the MALA MCMC method since it was the easiest to implement, and we weren't even sure if our approximate gradient would be useful enough, so we didn't want to spend more time on a more complex gradient MCMC method. The goal of our first experiment was to determine an optimal delta parameter to use to compare our MALA method to previous random walk attempts. This first experiment brought some doubt since the acceptance rates of our chains didn't appear to correlate with a change in delta, which we would expect in the same way that changing the step size in the random walk method directly affected the acceptance rate. After choosing a value of delta our second experiment was a direct comparison between MALA and random walk where we started each chain at the same spots near the maximum posterior probability region. This experiment solidified our hypothesis that something was wrong with either the MALA implementation or the approximate gradient because MALA both took longer to converge, and had a consistently lower acceptance rate than the random walk chains. We conclude that the gradient simplified formula does not give enough information to point our proposals in the right direction, although it may give a good direction for burn-in i.e. sampling that starts from an initial guess far from the maximum posterior region in the probability space.

6.2 Future Work

There is much more work that can be done from here. Further analysis needs to be done to investigate why varying the delta parameter in the MALA algorithm didn't seem to correlate with the acceptance ratio. One huge improvement would come from rewriting the Geoclaw tsunami simulation in a language that has automatic differentiation built in, that way an exact gradient could be used which would greatly improve convergence. There are many other MCMC methods that use a gradient other than MALA such as Hamiltonian Monte Carlo (HMC), Preconditioned Crank-Nicolson (pCN) that could be implemented using the same approximate gradient we have outlined, or an improved gradient. Other ideas include using a MCMC method that doesn't involve a gradient at all like Gibbs sampling, importance sampling, etc.

Another idea to improve our algorithm, is to either replace Geoclaw entirely with a less expensive formula or data driven model, or do an alternating approach where we switch between using Geoclaw and a less expensive forward model at given intervals. That way more samples can be pumped out in the same amount of time due to the less expensive formula iterations, but the full Geoclaw simulation should still ensure that convergence is reached eventually. It is unclear if this would be an improvement, so further investigation would be necessary to implement and test this theory.

Bibliography

[1] Know what is a Tsunami? CDEMA Web Portal - We Ready! (n.d.) http://weready.org/tsunami/index.php?option=com_content&amp;view=article&amp;id=8&amp;It

[2] Ron Harris and Jonathan Major. Waves of destruction in the East Indies: the Wichmann catalogue of earthquakes and tsunami in the Indonesian region from 1538 to 1877. Geological Society, London, Special Publications, 441:SP441–2, 2016.

[3] Dewey, J. and Byerly, P. (1979). The History of Seismometry to 1900. Earthquake Information Bulletin (USGS), 11(2), 64–70.

[4] Jacob Swart. Verhandelingen en Berigten Betrekkelijk het Zeewegen en de Zeevaartkunde (English: Treatises and Reports Related to the Seaways and Nautical Sciences). 13:257–274, 1853.

[5] Ringer, Hayden J., ”A Method for Reconstructing Historical Destructive Earthquakes Using Bayesian Inference” (2020). Theses and Dissertations. 8678.

[6] Ringer, H., Whitehead, J. P., Krometis, J., Harris, R. A., GlattHoltz, N., Giddens, S., et al. (2021). Methodological reconstruction of historical seismic events from anecdotal accounts of destructive tsunamis: A case study for the great 1852 Banda arc megathrust earthquake and tsunami. Journal of Geophysical Research: Solid Earth, 126, e2020JB021107. https://doi. org/10.1029/2020JB021107

[7] Glatt-Holtz, Nathan, Krometis, Justin and Mondaini, Cecilia F. “On the accept-reject mechanism for Metropolis-Hastings algorithms.” ArXiv abs/2011.04493 (2020): n. pag.

[8] Krometis, J. (2018). A bayesian approach to estimating background flows from a passive scalar (dissertation).

[9] Roberts, G. O., &amp; Tweedie, R. L. (1996). Exponential convergence of Langevin distributions and their discrete approximations. Bernoulli, 2(4), 341. https://doi.org/10.2307/3318418

[10] Randall J LeVeque and David L George. High-resolution finite volume methods for the shallow water equations with bathymetry and dry states. In Advanced Numerical Models for Simulating Tsunami Waves and Runup, pages 43– 73. World Scientific, 2008. 58

[11] Randall J. LeVeque, David L. George, and Marsha J. Berger. Tsunami modelling with adaptively refined finite volume methods. Acta Numerica, 20:211–289, 2011.

[12] Frank I González, Randall J LeVeque, Paul Chamberlain, Bryant Hirai, Jonathan Varkovitzky, and David L George. Validation of the GeoClaw Model. In NTHMP MMS Tsunami Inundation Model Validation Workshop. GeoClaw Tsunami Modeling Group, 2011.

[13] Marsha J Berger, David L George, Randall J LeVeque, and Kyle T Mandli. The GeoClaw software for depth-averaged flows with adaptive refinement. Advances in Water Resources, 34(9):1195–1206, 2011.

[14] Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. Equation of state calculations by fast computing machines. The Journal of Chemical Physics, 21(6):1087–1092, 1953

[15] W Keith Hastings. Monte Carlo sampling methods using Markov chains and their applications. Biometrika, 57(1):97–109, 1970.

[16] Yoshimitsu Okada. Surface deformation due to shear and tensile faults in a half-space. Bulletin of the Seismological Society of America, 75(4):1135–1154, 1985.

[17] Ward S.N. (2021) Tsunami. In: Gupta H.K. (eds) Encyclopedia of Solid Earth Geophysics. Encyclopedia of Earth Sciences Series. Springer, Cham. https://doi.org/10.1007/978-3-030-58631-7_22

[18] Andrew Gelman and Donald B. Rubin. Inference from Iterative Simulation Using Multiple Sequences. Statistical Science, 7(4):457–472, 11 1992.

[19] Dr. Gavin P. Hayes (2018) USGS Slab 2 Global Subduction Zone Geometries. https://github.com/usgs/slab2