Interim Report 1: Incorporating Stepping-Stone Sampling Into BayesWave

Seth Moriarty
(Dated: July 7 2022)

`BayesWave` is a library of code used to analyze data from LIGO's gravitational wave detections. `BayesWave` uses Bayesian statistics to reconstruct signals and determine possible sources. The likelihoods of various models can be compared, such that `BayesWave` can determine the most likely sizes, locations, and types of sources that could produce a certain detected signal. Currently, `BayesWave` uses Thermodynamic Integration (TI) to calculate the likelihoods of various models. An alternative method is called Stepping-Stone (SS) sampling. In other fields, SS has been shown to be as accurate as TI while also being less computationally expensive. This project explores the comparison between TI and SS methods when each is applied inside `BayesWave`, to determine if SS is a viable replacement for TI to be used for analysis of LIGO's fourth detection run in 2023.

I. INTRODUCTION/BACKGROUND

A. LIGO

Gravitational waves are ripples in space-time caused by high-energy events in outer space, such as supernovae and the collisions of black holes. LIGO [1], which stands for Laser Interferometer Gravitational-wave Observatory, is a large ground-based interferometer used to detect those ripples. It first detected gravitational waves in 2015, and has been carrying out observational runs since. LIGO detects gravitational-waves produced by distant, massive, compact objects by measuring slight fluctuations in its interferometric "arm" length. Once LIGO has collected data from an event, that data needs to be processed and analyzed computationally.

B. Bayesian Statistics

Bayesian statistics is a form of statistics in which earlier probability distributions (priors) can be updated to account for new data to produce new distributions (posteriors). This is done using Bayes' Theorem, given by equation 1.

$$P(A|B) = \frac{\mathcal{L}(B|A)\pi(A)}{P(B)} \tag{1}$$

Some $P(A|B)$ represents the probability of some event A given that B is true, and $\pi(A)$ is the probability of some event A independent of other events. Here, $P(A|B)$ is the posterior, the prior is $\pi(A)$, and the likelihood is $\mathcal{L}(B|A)$. When you integrate such a function over the entire parameter space, the evidence is produced. Evidences can be compared using the Bayes' factor, a ratio of evidences, like so:

$$\frac{P(B_1)}{P(B_2)} \tag{2}$$

Such evidences can be calculated with methods such as thermodynamic integration or a stepping-stone algorithm. Being able to compare models is crucial to determining the most likely sources of gravitational wave signals that LIGO detects. The part of BayesWave which calculates evidences and compares them is the part relevant to this project. This project focuses on using BayesWave to calculate evidences and compare them.

C. `BayesWave`

`BayesWave` [2] is a library of code which analyzes LIGO data using Bayesian statistical methods. is able to account for noise and glitches as it reconstructs observed signals. It reconstructs detected waveforms as a series of wavelets which form a model [2]. Models can then be compared with their Bayes' Factors, to determine which are good fits for the observed data.

When `BayesWave` is used for signal matching, it can execute different types of runs that will match the observed signal with different types of waveforms. `BayesWave` can produce matches that are coherent wavelet models (signal), incoherent wavelet model (glitch), waveform template model (CBC), and many combinations of those. When one of these runs has been completed, `BayesWave` creates and stores model data and diagrams in a directory, useful for analysis. Figure 1 shows a reconstruction of the 150914 signal at the Hanford detector using a coherent wavelet model. Figure 2 shows the reconstructed spectrogram from the same signal.

II. EVIDENCE CALCULATION

To calculate marginalized likelihoods, or evidences, with thermodynamic integration or stepping-stone methods, we start with Bayes' theorem. We rewrite equation 3 into more relevant notation. Here $p_i(\theta)$ is the posterior probability density for some model, $\pi(\theta|M_i)$ is the prior distribution $\mathcal{L}(D|\theta, M_i)$ is the likelihood function of some data, D, given that the prior is true. z_i the evidence (Eq. 4), a normalizing constant also referred to as the marginal likelihood [3].

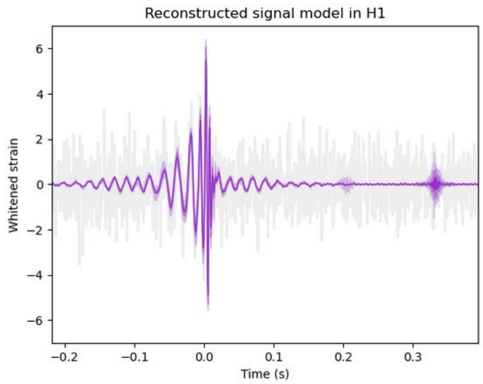

FIG. 1: reconstructed waveform using `BayesWave`

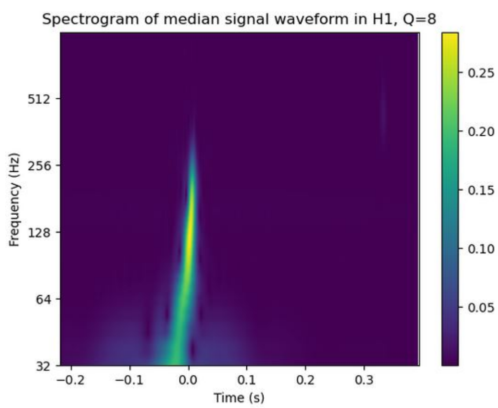

FIG. 2: reconstructed signal spectrogram using `BayesWave`

$$p_i(\theta) = \frac{\mathcal{L}(D|\theta, M_i)\pi(\theta|M_i)}{z_i} \tag{3}$$

$$z_i = p(D|M_i) = \int \mathcal{L}(D|\theta, M_i)\pi(\theta|M_i)d\theta \tag{4}$$

The likelihoods of two models can be compared using a ratio between their evidences known as a Bayes' factor (Eq. 3). this ratio, which we can call μ, is generally scaled logarithmically, as in equation 5.

$$\mu = ln(\frac{z_1}{z_0}) \tag{5}$$

What we want to do is rewrite equation 5 in terms of the symbol β. For this, we need more context about how `BayesWave` solves for evidences. It does so with a Markov chain Monte Carlo (MCMC) process [4]. A chain is a series of points that move through a parameter space to take samples of the probability distribution of the space. It moves in a way determined by the chain's "temperature" until reaching an equilibrium state.

A high temperature corresponds to a chain which, similar to the behavior of a thermodynamic system, has a higher chance of jumping to less likely states. This causes the likelihood function to approach the prior distribution (flattening out in parameter space). Likewise, a low temperature corresponds to chains that is more likely to stay in areas of high probability [3], resulting in a peaked probability distribution approaching that of the posterior. Because β is the inverse of the temperature, a β value moving from zero to one corresponds to chains "cooling down", while starting at one and moving to zero corresponds to a chain "heating up".

the result of a series of chains is sample expectations, which can then be used to estimate the integrals we need. An example of this is figure 3, where the changing β value is on the x axis, and the sample expectation values of each chain are on the y axis. The evidence of a model can then be rewritten in terms of β, in equation 6

$$\mu = ln(\frac{z_1}{z_0}) = ln(z_1) - ln(z_0) = \int_0^1 \frac{\partial ln(z_\beta)}{\partial \beta} d\beta \tag{6}$$

III. THERMODYNAMIC INTEGRATION

Currently `BayesWave` uses thermodynamic integration [5] to calculate evidences for potential models. The goal of thermodynamic integration and the stepping-stone algorithm is to do this by estimating the integral form of equation 6. TI is also known as path sampling, because it involves taking samples along a path of temperatures. β begins at one extreme, either zero or one, and travels along a path to the other extreme as the temperature changes.

The goal of thermodynamic integration and the stepping-stone algorithm is to estimate the integral form of equation 6. This is done with TI by taking many discrete steps as β moves between 0 and 1 and taking samples at each temperature. The samples can then be used to estimate the integral, as we can see in figure 3 [3]. In figure 4, a curve estimated with TI using 1000 sample points is shown.

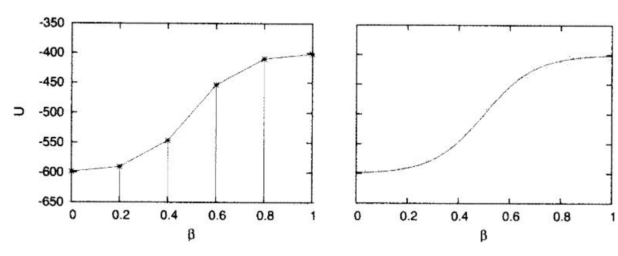

FIG. 3: Using discrete points to estimate an integral

Thermodynamic integration estimates the integral in equation 6 using general path sampling [6]. this involves using the expectation E. This is given in equation 7, read as the expectation of the likelihood, given β. For

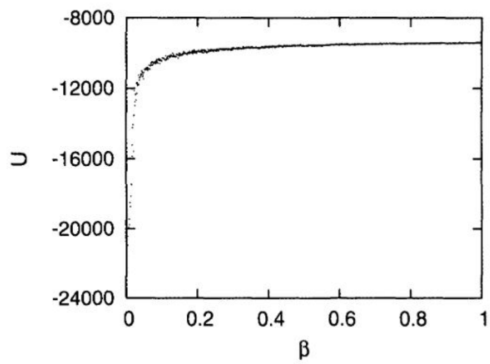

FIG. 4: Estimation using 1000 points

each chain location along β's path, samples will be taken and a sample average acquired. Then, those values can be used to estimate the integral [6], as per equation 8. This takes some time, as each β chain is run one at a time, rather than in parallel.

$$E_\beta[log(\mathcal{L}(D|\theta, M))] = \int log(\mathcal{L}(D|\theta, M))\pi(\theta)d\theta \quad (7)$$

$$log(z) = \int_0^1 E_\beta[log(\mathcal{L}(D|\theta, M))]d\beta \quad (8)$$

Thermodynamic integration has proven to be a very reliable method, provided enough samples are taken to minimize bias and produce accurate estimations of the desired curve. TI experiences thermic lag bias [3] as it changes β values and adjusts to each new value. This causes TI to slightly underestimate marginal likelihood values if β begins at zero, and a slight overestimate if it starts at one. It also experiences discretization bias, because of the limits on the accuracy with which a discrete number of values can estimate a continuous integral. Despite these points, and the fact that it is computationally expensive, TI is significantly more accurate than simpler methods, and thus a very helpful tool.

IV. THE STEPPING-STONE METHOD

The stepping-stone algorithm [6] is a method for finding evidences that is similar to TI but is less computationally costly. like TI, SS calculates marginal likelihoods directly, producing similar, and actually slightly more accurate [7] estimates.

the stepping-stone method calculates evidences differently than TI. Rather than calculating the average likelihood at each β value and summing them to estimate the integral, SS compares marginal likelihoods between each discrete β_i value and that of the one before it, in a process called importance sampling. Then the product of those ratios can be used to estimate the evidence [6]. This is shown in equation 9, where K is the number of chains (β values) used.

$$z = \frac{z_1}{z_0} = \prod_{k=1}^{K-1} \frac{z_{\beta_k}}{z_{\beta_{k-1}}} \quad (9)$$

A benefit of this method is that every chain does not need to sample from the posterior distribution, as it refers to the distribution immediately before it instead. This is more accurate than the TI method of averaging samples at each step in comparison to the posterior [7].

LIGO's fourth observational run is expected to detect significantly more events than previous runs, so it is possible that the SS algorithm will make an important addition to the tools `BayesWave` has at its disposal to analyze that data.

V. OBJECTIVES

The main goal of this project is to run `BayesWave` using both TI and SS methods, to compare the speed and accuracy of the two methods. This will help determine if the stepping-stone algorithm would be a worthwhile replacement to TI for LIGO's upcoming run. This will mean doing runs in bulk and comparing both the runtime and evidence convergence.

To determine this, many runs must be done using condor, which automates the run process. The data from these runs must then be organized so that the TI and SS data can be compared. If SS proves to be faster or more accurate, it may be incorporated into the master branch of `BayesWave` for future data analysis.

VI. PROGRESS

The first three weeks of this project were primarily dedicated to familiarizing myself with the background required for the project, and learning to run BayesWave from my computer.

I began in week one by working through the GWOSC open data workshop, in order to understand some of the steps that go into signal matching, noise reduction, and glitch removal. Because I am working in data analysis, and BayesWave is used for signal matching, this was an important basis of understanding for me to have that has made working with `BayesWave` and the data it outputs more intuitive.

To be able to work with `BayesWave`, I learned how to enter the LIGO computing cluster from my computer, and make a branch of `BayesWave` that I can edit and work with, based on the master branch. Additionally, I needed a branch off of Meg Millhouse's version of `BayesWave`, which includes the stepping-stone algorithm that I am working with. After I had a branch of `BayesWave` with the stepping-stone algorithm, I learned how to configure runs for it, changing parts of the run. This was to familiarize myself with the process of doing runs, and learn to analyze `BayesWave` outputs. These runs used signal data from the 150914 detection. The familiar signal made it easier to identify that the `BayesWave` outputs were as expected.

My next step was to write a python script which could access the stored evidence data from `BayesWave` runs. This included files which contained final evidence estimations using TI and the standard deviations thereof, as well as evidence estimations using SS. I could also access information about the sampling chains used in the run. This file included the number of temperatures at which samples were taken, the corresponding β value at each temperature, the average log likelihood at each temperature, and the standard deviations of those likelihoods. From this, I am able to produce scatter plots such as figure 5.

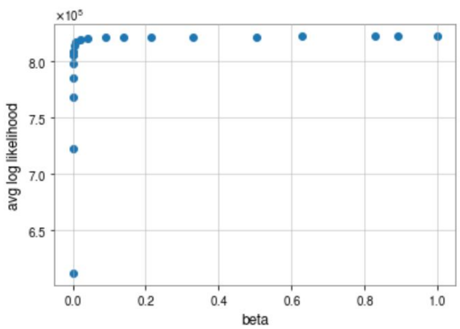

FIG. 5: Average log likelihood over a range of β values

Next, I remotely ran many runs with `BayesWave`, using simulated data from an injection of the 150914 signal. For these runs, I varied the number of sampling chains, beginning with runs where chains were used at five β values between zero and one, and increasing by fives to runs which used chains at 40 β values. Additionally, the number of iterations each sampling chain uses has been varied. For each number of chains used, I did runs with 1000000, 2000000, 3000000, and 4000000 iterations each. These runs all make use of both TI and SS sampling, so that the outputs can be compared.

Figure 6 shows the varying number of chains used for the runs along the x axis, and the evidences calculated by each run along the y axis. The green dots represent evidences found using thermodynamic integration, and the pink represent evidences found with the stepping-stone algorithm. For thermodynamic integration, when the number of chains used is small, the calculated evidences are underestimated. As the number of chains is increased, the evidences stabilize towards a very good estimate. It seems that as the number of chains is increased, evidences calculated using SS converge on a good estimate much more quickly than those from TI.

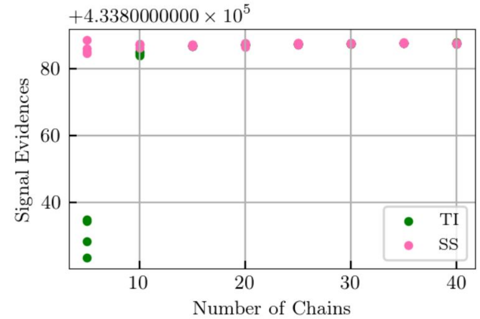

FIG. 6: Number of chains used for sampling vs calculated evidence values

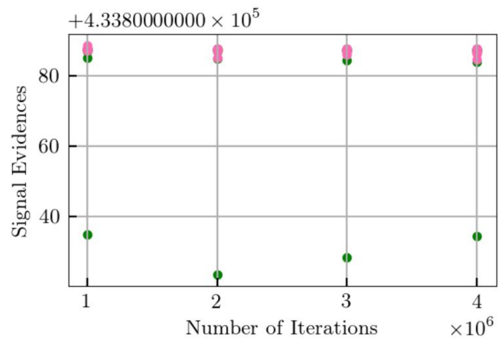

FIG. 7: Number of iterations sampled vs calculated evidence values

Another way to see this is with figure 7. The number of iterations used for sampling is along the x axis and the evidences along the y axis. All of the SS evidences are near the same value, while you can see that the TI evidences from the 5-chain runs are significantly lower than the rest.

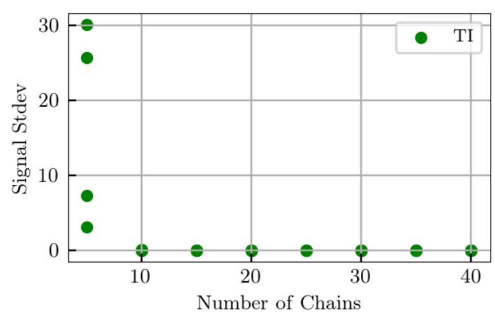

FIG. 8: Number of chains used for sampling vs standard deviation of the calculated evidences

Figure 8 plots the number of chains vs the standard deviation of the calculated evidences. As you can see, the error if very large when the number of chains used is small, and stabilizes to a near-zero value as the number of chains used increases. The evidence files currently provide standard deviation values for thermodynamic integration but not for stepping-stone. In the future, it would be helpful to be able to plot error for SS as well.

In the next steps of this project, I am going to do runs using 3, 7, and 12 chains, so that the shape of the curve given in figure 6 is more clear. Hopefully it will become easier to tell how much more quickly SS estimates stabilize compared to TI. In addition, I will be plotting graphs which compare signal models compared to glitch models for these runs. The figures produced so far are signal model evidences. In addition, it will be helpful to do runs on different waveforms other than 150914, to see how TI and SS compare across the board.

VII. ACKNOWLEDGEMENTS

I'd like to thank Sophie Hourihane and Katerina Chatziioannou for their tremendous support in this project, as well as Meg Millhouse for her work on the stepping-stone algorithm. I also gratefully acknowledge the support from the National Science Foundation Research Experience for Undergraduates (NSF REU) program, the California Institute of Technology, and the LIGO Summer Undergraduate Research Fellowship."

[1] LIGO- A Gravitational-Wave Interferometer.

[2] N. J. Cornish, T. B. Littenberg, B. Bécsy, K. Chatziioannou, J. A. Clark, S. Ghonge, and M. Millhouse, BayesWave analysis pipeline in the era of gravitational wave observations, Phys. Rev. D **103**, 044006 (2021), arXiv:2011.09494 [gr-qc].

[3] N. Lartillot and H. Phillipe, Computing Bayes' Factors Using Thermodynamic Integration (2006).

[4] S. Carstens, Introduction to Markov Chain Monte Carlo (MCMC) Sampling (2020).

[5] J. Annis, Thermodynamic Integration and Steppingstone Sampling Methods for Estimating Bayes Factors: A Tutorial, Journal of mathematical psychology **89** (2019).

[6] P. Maturana-Russel, R. Meyer, J. Veitch, and N. Christensen, Stepping-stone sampling algorithm for calculating the evidence of gravitational wave models, Phys. Rev. D **99**, 084006 (2019), arXiv:1810.04488 [physics.data-an].

[7] W. Xie, P. Lewis, Y. Fan, L. Kuo, and M.-H. Chen, Improving Marginal Likelihood Estimation for Bayesian Phylogenetic Model Selection", url =.