Gravitational Wave Signal Analysis Using Monte Carlo Markov Chain Algorithms

Nico O'Neill* and Chance Jackson†
Syracuse University Department of Physics
(Dated: November 24, 2025)

In this work, we present our replication of the parameter estimation results for the gravitational wave event GW150914 using the techniques of Bayesian inference, estimating that the component masses involved in the event were $M_1 = 38.26^{+13.45}_{-10.00} M_\odot$ and $M_2 = 29.61^{+10.92}_{-6.73} M_\odot$ at a luminosity distance of $d_L = 201.34^{+5.86}_{-1.30}$. We also discuss the convergence of our Markov chains via both visual inspection of a sample chain for each parameter and the direct computation of the Gelman-Rubin statistic. With both methods we observed significant evidence for the convergence of our chains.

1. INTRODUCTION

Within the field of gravitational wave data analysis, the estimation of the parameters of gravitational wave signals (i.e, the component masses involved, the distance to the source, etc.) is paramount, and as such, many methods have been devised for this purpose. Likely the most ubiquitous method for parameter estimation is matched filtering, which filters the stream of data from our detector networks against a template bank of expected waveforms, returning the waveforms with the highest match [1]. Matched filtering is highly accurate, but incredibly slow, as the template banks used in the filtering process can contain hundreds of thousands of waveforms.

In this work, we'll investigate an alternative approach, using a Monte Carlo Markov Chain algorithm to explore the posterior distribution of a gravitational wave waveform model. We apply this method to GW150914, the first detected gravitational wave event, and compare our results to the published matched-filtering estimates from LIGO [2].

1.1. Monte Carlo Markov Chain methods

Monte Carlo Markov Chain (MCMC) methods provide a powerful tool for drawing samples from a statistical distribution which might otherwise be prohibitively difficult to sample from [3]. The particular MCMC method employed in this paper is the Metropolis-Hastings algorithm, where a randomly placed walker is allowed to explore the parameter space of a distribution. The idea behind the Metroplis-Hastings algorithm is that the final distribution of walkers will eventually converge to the desired distribution given sufficient iterations, since the path that the walkers take through the parameter space is decided by the "transition probability", which depends upon the choice of proposal distribution (which "proposes" the next step in a chain) and the underlying distribution being sampled [3].

Since the distribution of our walkers only reaches the target distribution within sufficient iterations, it's important to ask how one might tell whether or not this convergence has occurred. Unfortunately, no statistic exists which can definitively state whether or not the chains have converged, but there are several methods which can provide useful evidence for the chains having converged. In this work, we'll rely on both visual inspection of the chains and the Gelman-Rubin statistic to provide evidence for convergence. The former method is perhaps intuitively obvious: if the chains have converged, then when the parameter values at each step are plotted, one should observe that they become roughly stationary, moving in small steps around a fixed value.

The Gelman-Rubin (GR) statistic, on the other hand, compares the variance between chains within an ensemble (a collection of many walkers) to the variance within a single chain [4]. The GR statistic for an ensemble of J chains each of length L is defined as:

$$R = \frac{\frac{L-1}{L}W + \frac{1}{L}B}{W}, \tag{1}$$

where:

$$W = \frac{1}{J}\sum_{j=1}^{J}\left(\frac{1}{L-1}\sum_{i=1}^{L}\left(x_i^{(j)} - \bar{x}_j\right)^2\right) \tag{2}$$

is the average variance within the chains of the ensemble, and:

$$B = \frac{L}{J-1}\sum_{j=1}^{J}\left(\bar{x}_j - \bar{x}_*\right)^2, \tag{3}$$

is the average variance between chains, with $\bar{x}_*$ the mean of the mean chain values. If all chains converge to the same posterior, we anticipate that the between-chain variance would approach zero (since if they converge, they've converged to the same distribution) [4]. In this limit, and as the length L of the chains approaches infinity, the GR statistic approaches 1. In practice, R being less than 1.1 or 1.05 is used as evidence that the chains have converged [4].

*Electronic address: neoneill@syr.edu
†Electronic address: cjacks34@syr.edu

2. SIMULATION

Our analysis consists of two primary components: a custom MCMC sampler and a Newton-informed optimization code.

2.1. PyCBC Library

PyCBC is an open-source Python package that was designed for gravitational wave data analysis [5]. In this code, PyCBC handled two main tasks before the MCMC Implementation. First, PyCBC was used to generate a waveform model. This function is where parameters were fit to and handled the physics backend of the fitting. This utilized the "get_fd_waveform" Function, which takes in mass, luminosity distance, coalescence phase, and minimum frequency to output a frequency domain waveform. This waveform went to the likelihood function to find the mismatch between the model and observed strain data. The next function from PyCBC we used was the "welch" function, which estimates the PSD from the data off-source. To reduce high-frequency artifacts in the inverse PSD¡ we use an inverse spectrum truncation, which limits the length of the time-domain whitening filter[6]

2.2. Optimization Functions

To provide good initial conditions and aid in converging, we used a multidimensional implementation of Newton's method, which iteratively minimizes a function by effectively fitting it with a parabola and moving towards the minima of that fitted parabola. To minimize some function $f(\vec{x})$, we begin at some point $\vec{x}_n$. Newton's method then proposes a step to the point $\vec{x}_{n+1}$, with the two related by:

$$\vec{x}_{n+1} = \vec{x}_n - H_f^{-1}(\vec{x}_n)\nabla f(\vec{x}_n), \tag{4}$$

where $H_f^{-1}(\vec{x_n})$ is the inverse Hessian matrix (the matrix of second partial derivatives) of the function $f(\vec{x})$, and $\nabla f(\vec{x})$ the gradient, both evaluated at the point $\vec{x}_n$. Because the cost function for gravitational Wave parameters is so curved, this method provides an accurate best initial guess before running the MCMC sampler, seen in Sec. 2.3 [7].

2.3. MCMC Implementation

To predict the physical parameters of GW150914, we constructed a Metropolis-Hastings Markov Chain Monte Carlo sampler[8]. The posterior density is:

$$p(\theta|d) \propto p(d|\theta)p(\theta), \tag{5}$$

where $p(\theta)$ is the prior and $p(d|\theta)$ is the likelihood. This postieror used priors chosen to reflect physically motivated constraints, such as assuming the mass is in a common range of 20-60$M_\odot$, the luminosity distance is between 200 and 800 Mpc, and assuming both masses are around the same.

For gravitational wave analysis, using log-likelihood functions is the standard Gaussian choice.

$$\log \mathcal{L}(\theta) = -\frac{1}{2}(\tilde{d} - \tilde{h}|\tilde{d} - \tilde{h}) \tag{6}$$

where $\tilde{d}$ is the strain data and $\tilde{h}$ is the model waveform from PyCBC.
The inner product is computed as:

$$(a|b) = 4\Delta f \sum \frac{a(f)b*(f)}{S_n(f)}, \tag{7}$$

where $S_n(f)$ is the noise PSD.

This likelihood is the same form commonly used in PyCBC integrators, such as PyCBC Inference[9].

To weigh the different frequency bins, we estimated the PSD using PyCBC's Welch's method and pass it through inverse spectrum truncation to prevent overfitting, which then is applied to the likelihood through the inner product. This also ensured the time-domain filtering has a finite length and avoids high-frequency ringing [6].

Our MCMC Implementation has two proposal mechanisms: a Gaussian random-walk or a Newton-informed proposal. The Gaussian walk proposal functioned as

$$\theta' = \theta + \sigma \tag{8}$$

where σ is the normal distribution. The Newton informed proposal method is outlined in 2.2. Our best results come from a mixed method using both, with an 80-20 mix. This mixed method is the most effective because it avoids getting trapped in a local minima or other instabilities. To improve sampling efficiency and reduce sensitivity to poor initial guesses, we ran multiple walker chains. Each walker is initialized near the result of a Newton-optimized chirp mass.

3. RESULTS AND VALIDATION

In this section, we present the results of our parameter estimation using a 10 walker 5000 step ensemble MCMC sampler, and show how they compare to those found by the LIGO data analysis pipeline and those estimated by the external MCMC package Emcee. We also analyze the convergence of our Markov chains through both visual inspection and the computation of the Gelman-Rubin statistic. Finally, we estimate the autocorrelation time of our Markov chains, which we find indicates our samples are highly correlated. We briefly discuss potential explanations for this.

3.1. Predicted Values

As GW150914 passed through the LIGO data analysis pipeline, the parameters of the signal were estimated using the technique of Matched Filtering, which filters the signal through a large bank of anticipated gravitational wave waveforms and returns the best match. The maximum likelihood values for the component masses and the luminosity distance published by the LIGO collaboration are given in Table I [2], along with their 95% credible interval. We use these values to benchmark the performance of our inference method.

Parameter	Value
M_1	$36^{+5}_{-4}\ M_\odot$
M_2	$29^{+4}_{-4}\ M_\odot$
d_L	410^{+160}_{-180} Mpc

TABLE I: Maximum likelihood parameter values returned by LIGO matched filter analysis.

3.2. Estimated Values

A corner plot showing the posterior probability distributions for each of the 5 parameters in our model is shown in Fig. 1. We note that the median component mass values of $M_1 = 38.26^{+13.45}_{-10.00} M_\odot$ and $M_2 = 29.61^{+10.92}_{-6.73} M_\odot$ are both contained within the 95% confidence interval of the maximum likelihood values returned by the LIGO collaboration, indicating agreement between these and our estimated values. On the other hand, our estimate of the luminosity distance, $d_L = 201.34^{+5.86}_{-1.30}$ Mpc, is not contained within the LIGO reported confidence interval. The luminosity distance can be a uniquely difficult parameter to estimate, since it's degenerate with the inclination angle of the source, which we neither specify or fit for [10]. It's possible that the inclusion of the inclination angle in our waveform model would be able to break this degeneracy, and our inference methods would be able to better estimate the luminosity distance.

Beyond the luminosity distance-inclination angle degeneracy, there's another feature of note in these distributions. In the two-dimensional distribution of the component masses, we observe an apparent multi-modality (the set of peaks in the center of the distribution). This is an important reflection of a particular symmetry in our problem. From Christensen et al., our waveform model is only uniquely defined by the total mass and mass ratio [11]. Hence, as long as these two values are kept constant, multiple values of the component masses can produce the same waveform model. This appears in the distribution of component masses returned by our inference, as we see two large peaks placed symmetrically above and below the central peak, meaning that increasing one mass while decreasing the other results in waveforms that appear equally likely.

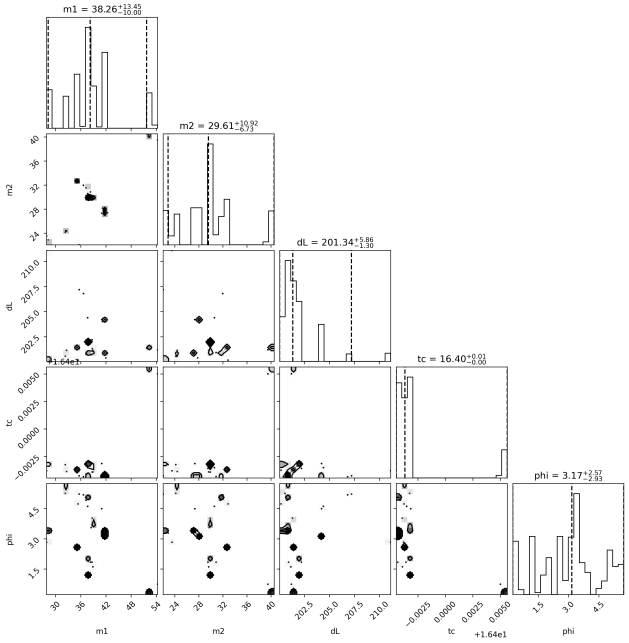

FIG. 1: Corner plot of posterior distributions after a 10 walker, 5000 step run. We quote the median parameter value, along with the 95% credible interval. Note the apparent multi-modality present in the two-dimensional mass distributions. This is expected, since unique waveforms are not specified by the individual values of the component masses, but rather the total mass and the mass ratio [11]. Hence, as long as the total mass and mass ratio are kept constant, the values of the component masses are free to switch, reflecting an important symmetry of our problem.

3.3. Comparison to Emcee Implementation

In order to provide a point of comparison for our handwritten MCMC implementation, we used an ensemble sampler from the external Emcee package. Figure 2 shows a sample chain chosen from both the Emcee ensemble sampler and our implementation. For the most part, they appear quite similar, appearing to converge at roughly equivalent rates. We do notice a large amount of variability later in the phi chain in the Emcee implementation. It's possible this is simply due to the nature of the phi parameter, which is 2π-periodic. If the walker reached a value slightly beyond 2π, it could've wrapped back around to 0, and vice versa.

3.4. Convergence

3.4.1. Visual Inspection

Figure 3 shows a sample chain (selected at random from our 10 walker ensemble) for each of our 5 parameters. For each, we observe a highly variable initial period, which extends to just below step 2000. Based upon

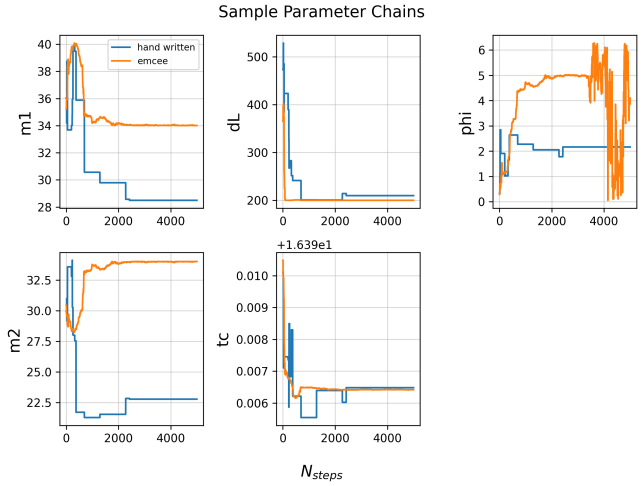

FIG. 2: Figure showing sample chains for each parameter, chosen randomly from our ensemble and a 10 walker, 5000 step ensemble run with the external Emcee package. For both, we observe the apparent convergence of most parameters with the exception of the coalescence phase in the Emcee ensemble. This is due, potentially, to the nature of the coalescence phase, which is 2π-periodic, hence it's possible that this walker reached a value near 2π and the phase wrapped back around to 0 and vice versa.

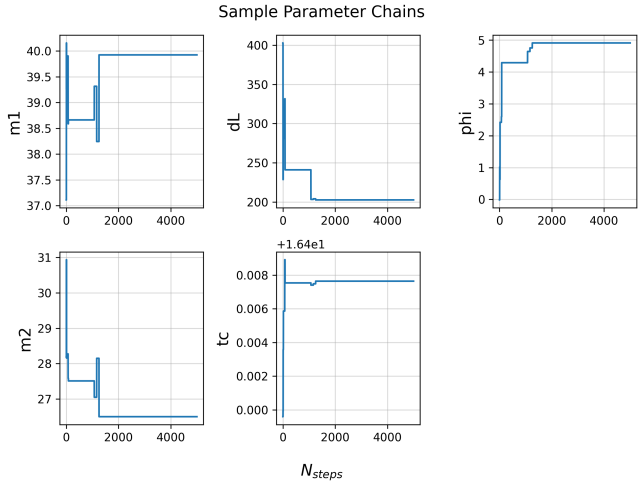

FIG. 3: Sample chains for each parameter randomly chosen from our ensemble. We observe the high variance early in the chain, characteristic of the "burn-in" period. We also observe that each chain seems to settle around a fixed value, which points to our chains having converged. Based on these traces, we conservatively chose to eliminate the first 50% of samples as the "burn-in" period in all analyses.

these plots, we elected to eliminate the first 50% of samples as burn-in, which appears to be a rather conservative estimate of the actual burn-in period. These trace plots show that each parameter appears to settle around a fixed value after the initial burn-in period, indicative of our chains converging.

3.4.2. Gelman-Rubin Statistic and Autocorrelation Length

In addition to the visual methods of Sec. 3.4.1, we also directly calculated the Gelman-Rubin statistic and the autocorrelation length for each of our parameter chains within our ensemble. These values are given in Tab. II. The estimated Gelman-Rubin statistic for each of our parameters were well below the typically cited cut-off value of $R < 1.1$, providing further evidence for the convergence of our chains (and validating what we observed visually in Sec. 3.4.1 [12]. On the other hand, the autocorrelation length for several of our parameters was higher than the typical threshold value of $L > 50\tau$, with τ the autocorrelation length. This does not necessarily indicate non-convergence of our chains, and could have several possible explanations. For instance, it's possible that our walkers were not exploring our parameter space sufficiently fast enough, resulting in many nearby samples and hence a large autocorrelation length. This could potentially be corrected in future analyses by tuning the acceptance ratio, which was consistently ¡10% for our analysis, allowing the walkers to explore more of the parameter space.

Parameter	Gelman-Rubin Stat. (R)	Autocorrelation Length (τ)
M_1	1.002	324.5
M_2	1.008	200.8
d_L	1.002	24.15
t_c	1.003	142.7
phi	1.003	99.12

TABLE II: Gelman-Rubin statistic and Autocorrelation Length for 10 walker 5000 step ensemble run. These values were computed for the same run which produced the chain plot shown in Fig. 3. We see that the Gelman-Rubin statistic for all of our parameters is well below the typical cut-off value of $R < 1.1$, providing strong evidence for the convergence of our chains. We also note that the autocorrelation length of several of our chains are long compared to the chain length ($L < 50\tau$). Potential explanations for this are discussed within Sec. 3.4.2

4. CONCLUSIONS

Using MCMC methods, we were able to accurately predict the component masses of GW1509514, with our values falling within the 95% confidence interval reported by the LIGO collaboration. Despite this, our estimation of the luminosity distance fell outside of the interval, potentially due to a degeneracy between it and the inclination angle of the binary. Through both visual assessment and the estimation of the Gelman-Rubin statistic, we provided strong evidence for the convergence of our chains.

[1] N. Christensen and R. Meyer, Reviews of Modern Physics **94** (2022), ISSN 1539-0756, URL http://dx.doi.org/10.1103/RevModPhys.94.025001.

[2] B. Abbott, R. Abbott, T. Abbott, M. Abernathy, F. Acernese, K. Ackley, C. Adams, T. Adams, P. Addesso, R. Adhikari, et al., Physical Review Letters **116** (2016), ISSN 1079-7114, URL http://dx.doi.org/10.1103/PhysRevLett.116.061102.

[3] J. S. Speagle, *A conceptual introduction to markov chain monte carlo methods* (2020), 1909.12313, URL https://arxiv.org/abs/1909.12313.

[4] R. D. Peng, *7.4 Monitoring Convergence — Advanced Statistical Computing* (2022), URL https://bookdown.org/rdpeng/advstatcomp/monitoring-convergence.html.

[5] D. B. C. M. B. J. W. T. D. C. C. C. T. D. L. P. G. S. C. D. S. D. M. C. S. W. A. R. W. B. M. D. M. F. P. P. K. Alex Nitz, Ian Harry, *gwastro/pycbc: v2.3.3 release of pycbc* (2024), URL doi:10.5281/zenodo.10473621.

[6] Usman, Syracuse University Honors Program Capstone Projects (2016), URL https://scispace.com/pdf/the-pycbc-search-for-gravitational-waves-from-compact-bpdf.

[7] Rodriguez, Phys. Rev. (2013), URL https://arxiv.org/pdf/1308.1397.

[8] Biscoveanu, DCC (2020), URL https://dcc.ligo.org/public/0168/G2000800/003/intro_to_pe.pdf.

[9] C. B. N. R. Biwer, Capano, Arxiv (2018), URL https://doi.org/10.48550/arXiv.1807.10312.

[10] S. A. Usman, J. C. Mills, S. Fairhurst, S. A. Usman, J. C. Mills, and S. Fairhurst, The Astrophysical Journal **877**, 82 (2019), URL https://doi.org/10.3847/1538-4357/ab0b3e.

[11] N. Christensen and R. Meyer, Physical Review D **64** (2001), ISSN 1089-4918, URL http://dx.doi.org/10.1103/PhysRevD.64.022001.

[12] D. Vats and C. Knudson, *Revisiting the gelman-rubin diagnostic* (2020), 1812.09384, URL https://arxiv.org/abs/1812.09384.