Estimation of Parameters of Compact Binaries from Gravitational-wave Observations

A thesis submitted towards partial fulfilment of
BS-MS Dual Degree Programme

by

Siddharth Rajiv Mohite

under the guidance of

Prof. P.Ajith
Dr.Archisman Ghosh

International Centre for Theoretical Sciences
(ICTS-TIFR),Bengaluru

Indian Institute of Science Education and Research,Pune

Certificate

This is to certify that this thesis entitled "Estimation of Parameters of Compact Binaries from Gravitational-wave Observations" submitted towards the partial fulfilment of the BS-MS dual degree programme at the Indian Institute of Science Education and Research,Pune represents original research carried out by "Siddharth Rajiv Mohite" at the "International Centre for Theoretical Sciences (ICTS-TIFR),Bengaluru", under the supervision of "Prof. P.Ajith and Dr. Archisman Ghosh" during the academic year 2014-2015.

Student
SIDDHARTH MOHITE

Supervisor
PROF.P.AJITH

Declaration

I hereby declare that the matter embodied in the report entitled "Estimation of Parameters of Compact Binaries from Gravitational-wave Observations" are the results of the investigations carried out by me at the International Centre for Theoretical Sciences (ICTS-TIFR),Bengaluru, under the supervision of Prof. P.Ajith and Dr. Archisman Ghosh and the same has not been submitted elsewhere for any other degree.

Student
SIDDHARTH MOHITE

Supervisor
PROF. P.AJITH

Acknowledgements

I would first and foremost like to express my sincere gratitude towards Prof. P.Ajith for giving me the great oppurtunity to work on this project. It has made a field which I was already really interested in, even more interesting. I am always encouraged to work harder by looking at his motivation towards research in this field. I am thankful to him for guiding me throughout and always being there to discuss whenever I had any doubts or problems.

Secondly, I sincerely wish to thank Dr. Archisman Ghosh who has also majorly guided me through this project. I thank him for all the useful discussions that we had and bearing with me when I used to bug him at random times. I am thankful to him for dedicatedly giving his time towards helping me, whenever I was stuck with any problem.

I would also like to specially thank the rest of the Astrorel Group at ICTS. The environment in the group has been most friendly and supportive and the suggestions I got from the group have always been helpful. I am also thankful for the weekly seminars, paper discussions and journal clubs that take place. They have really fuelled my interest in this field and increased the breadth of my knowledge greatly.

Finally, I would like to thank my family and friends, for their constant support and care.

Abstract

Gravitational-wave astronomy promises to open a new window to viewing the universe and a great amount of interesting physics is thought to come from the direct observation of gravitational waves (GWs).Coalescences of binaries comprising of compact objects(neutron stars or black holes) called CBCs are one of the most powerful and promising sources for gravitational-wave detection. In order to understand and make sense of the physical information we receive from such sources, it is important and imperative to accurately estimate their characteristic parameters. Parameter estimation of such sources, involves estimating quantities related to the binary, such as, the individual masses, spins, luminosity distance of the binary, its sky location, polarization etc. Second-generation interferometric detectors such as Advanced LIGO will be operational in the coming year and the first direct detection of graviational waves (GWs) is expected. Thus, once the GW signals are detected, parameter estimation becomes the next logical step in the analysis of the obtained gravitational-wave data. The aim of this project, therefore, is to develop an efficient Bayesian parameter estimation code using an efficient Markov Chain Monte Carlo (MCMC) algorithm, that can be used to get posterior distributions of the binary parameters from GW observations.

We started out by writing a code to obtain the two-dimensional posterior distributions for the chirp mass, defined later, and the luminosity distance of the binary, in the simple case when the signal contains a Newtonian (the leading order approximation to the GW signal) waveform added to white noise. The posteriors were obtained for signals in both the time and frequency domains and they were compared. We then incorporated more realistic and accurate waveforms with noise models and obtained posteriors for component masses and equivalently for the chirp mass and symmetric mass ratio. The single chain MCMC code was then parallelized to give data from multiple chains, starting at different initial positions in the parameter space, using the Multiprocessing module in Python, the programming language. Finally, we made the code run with a more sophisticated and adaptive algorithm, called the ensemble sampler and obtained posteriors for the chirp mass and mass ratio. We also found that this code is much more efficient and accurate than the single-chain MCMC code.

Contents

Chapter 1

Introduction

Gravitational waves, as predicted by Einstein's Theory of Relativity [12], are freely propagating perturbations of the gravitational field, travelling at the speed of light. Just like light travels as oscillations of the electromagnetic field. But these perturbations are so feeble that detectable amounts of gravitational radiation can only be obtatined from massive and energetic astrophysical systems such as compact binary coalescences of black holes and neutron stars. The first indirect evidence for the existence of gravitational waves was provided by the observations of Hulse and Taylor [10] in the year 1974, where they discovered the first binary pulsar. Since then, efforts have and are being made world-wide to directly detect these perturbations and as a result of these efforts, a number of detectors such as the Laser Interferometric Gravitational-wave Observatory(LIGO), VIRGO, GEO, KAGRA etc. have been set up at different locations around the globe. These detectors are expected to detect signals of gravitational waves in the near future and this will thus open a new window onto the universe.

Compact binary coalescences (CBCs) can be divided into three phases with respect to their evolution - inspiral, merger and ringdown. Due to numerous efforts in analytical and numerical relativity, there has been great progress in modelling the gravitational waveforms emitted by these coalescing compact binaries [6] in each of these phases. Analytical or Post-Newtonian methods as they are popularly known are valid only upto the inspiral part of the evolution. Modelling the remaining two phases requires numerical relativity. Current gravitational waveforms are generated by combining the results from Post-Newtonian methods and numerical relativity.

A plethora of interesting and exciting physics is expected to come out from the direct observation of gravitational waves [16] right from testing Einstein's theory in strong-field regimes to observations that may corroborate or change our ideas of various theories in Quantum Gravity. But in order to make any conclusions, it is first necessary to analyze the gravitational-wave signal or data that we obtain. An essential or rather the main goal of this analysis would be to estimate the characteristic parameters of such binary sources which are emitting gravitational radiation. These parameters include ones intrinsic to the source such as the masses of the components in the binary and their spins, and also extrinsic ones such as the sky location of the binary, its inclination angle with respect to our line of sight etc. There are two main methods that have been developed for parameter estimation from gravitational-wave data - the covariance matrix method and Markov Chain Monte Carlo(MCMC) techniques. But as pointed out in [1, 2, 18], covariance matrix methods do not give a very reliable estimate of the parameters. MCMC methods on the other hand, can give more reliable results even though they can be computationally expensive. We thus set out to create and implement efficient MCMC techniques which can be used in parameter estimation and thus obtain the posterior distributions for these parameters.

As seen above, there are a number of parameters that need to be estimated accurately when it comes to gravitational wave data analysis. This implies that the space of these parameters is multi-dimensional and really large. Sampling such a space upto a high resolution becomes practically impossible and we thus have to resort to stochastic methods such as the MCMC [7]. There have been studies to see how MCMC methods can be used for parameter estimation previously [3, 5, 14, 15]. MCMC methods are essentially based on the principles of Bayesian inference [17], where the probability of detecting the parameters is updated as additional data is acquired. This requires placing initial probability bounds on the parameters called the prior probability.

The MCMC method that we have used in our work is based on the algorithm given by Metropolis [13] and Hastings [9]. The exact working of this algorithm and how it is implemented in our code is explained in later sections. Our initial code used a single MCMC chain to sample the parameter space and return the posteriors. This is a highly inefficient method compared to one in which several chains have been initiated. We test and check this fact by incorporating multiple chains by using the Multiprocessing module in Python. Further, when we found that the posteriors we obtained were not accurate, inspite of using Multiprocessing, due to the likelihhod being biased towards the boundary of the prior, we resorted to the sophisticated and efficient method of the ensemble sampler which is discussed in [8]. The results of this method are again discussed later. In this work we have tried to obtain posteriors on parameters such as the component masses, chirp mass, mass ratio and luminosity distance.

The following section describes the theory of Markov Chain Monte Carlo methods and how they are based on the principles of Bayesian statistics. It also gives a brief description of the Metropolis-Hastings algorithm and also of the moves involved in the ensemble sampler technique. Section 3 gives the methods used in this project to generate the gravitational-wave data, the implementation of the single chain and multiple chain MCMC codes and also of the ensemble sampler. Section 4 shows the posteriors obtained from our codes for these different methods and the discussion for the same. We also discuss how the ensemble sampler method is more efficient than its single chain counterpart. The sampling and computing challenges faced in this project are also described here.

Chapter 2

Theory

Parameter estimation, as stated, is carried out in this work using the stochastic methods of Markov Chain Monte Carlo (MCMC). These MCMC methods are based on the principles of Bayesian inference. Before understanding how these methods work and what are the principles the rely on, it is essential to state certain preliminaries with respect to gravitational waves from coalescing compact binaries and their detection.

2.1 Gravitational Waves

Analogous to light travelling as oscillations in the electromagnetic field, gravitational waves are propagating perturbations of the gravitational field. They are generated by accelerating masses just as electromagnetic waves are generated by accelerating charges. Coalescing compact binaries are expected to generate large amounts of gravitational radiation. The evolution of such binaries can be divided into three stages namely, the inspiral, merger and ringdown. Three major approximation methods are used to model the gravitational waves generated by such sources during the mentioned phases of their evolution. [16]:

- **Post-Newtonian Scheme**: This involves deriving solutions to Einstein's equations under the low-velocity (v/c) as well as weak-field (M/R) expansion. The zero Post-Newtonian(PN) term describes a Newtonian binary system. In recent years, the PN approximation has been evaluated upto very high orders in v/c since in the late stages[1] of the evolution of the binary, the departure from Newtonian dynamics is significant and we need to model this phase as accurately as possible. The PN approximation breaks down before the coalescence of the binary and hence cannot be used to model the further stages, merger and ringdown, of the system. We thus need to resort to simulations from numerical relativity.

- **Perturbation Theory**: The expansion here is carried out in terms of the mass ratio of the binary components. The approximation is being used to study the GW signals from coalescing Stellar Mass Black Hole(SMBH) binaries.

[1] upto inspiral

- **Numerical Simulations**: Numerical relativity is used to model the final stages of the evolution of the binary, where Post-Newtonian analytic expressions are not available.

2.1.1 Post-Newtonian waveform in the Fourier Domain

To get an idea of how a gravitational wave from an astrophysical source, such as a compact binary, depends on its parameters, let us look at the mathematical form of this dependance. From Post-Newtonian theory, the waveform of a binary in the Fourier or frequency domain, $H(f)$, can be written as -

$$H(f) = \mathcal{A} f^{-7/6} \exp[i\Psi(f) + i\frac{\pi}{4}] = h_+(f) + ih_X(f) \tag{2.1}$$

where, the Fourier amplitude $\mathcal{A}$ and phase $\Psi(f)$ are given as,

$$\mathcal{A} = \frac{\mathcal{C}}{D\pi^{2/3}}\sqrt{\frac{5\nu}{24}}M^{5/6}, \Psi(f) = 2\pi f t_C + \Phi_C + \sum_k \alpha_k (\pi M f)^{\frac{(k-5)}{3}} \tag{2.2}$$

Here, $M = m_1 + m_2$, is the total mass of the binary, ν is the symmetric mass ratio defined as $\nu = \frac{m_1 m_2}{M^2}$, $\mathcal{C}$ is a function of angles, and t_C and Φ_C denote the epoch of merger and phase of the GW from the binary at that particular epoch respectively. The equation can also be expressed in terms of a quantity called the chirp mass M, which is a combination of the masses m_1 and m_2 and gives the leading order approximation to the gravitational wave signal. $\mathrm{M} = \frac{(m_1 m_2)^{3/5}}{(m_1+m_2)^{1/5}}$. The α_k's are the PN coefficients in the expansion of the phase in the Fourier domain and expressions for them are given in [16].

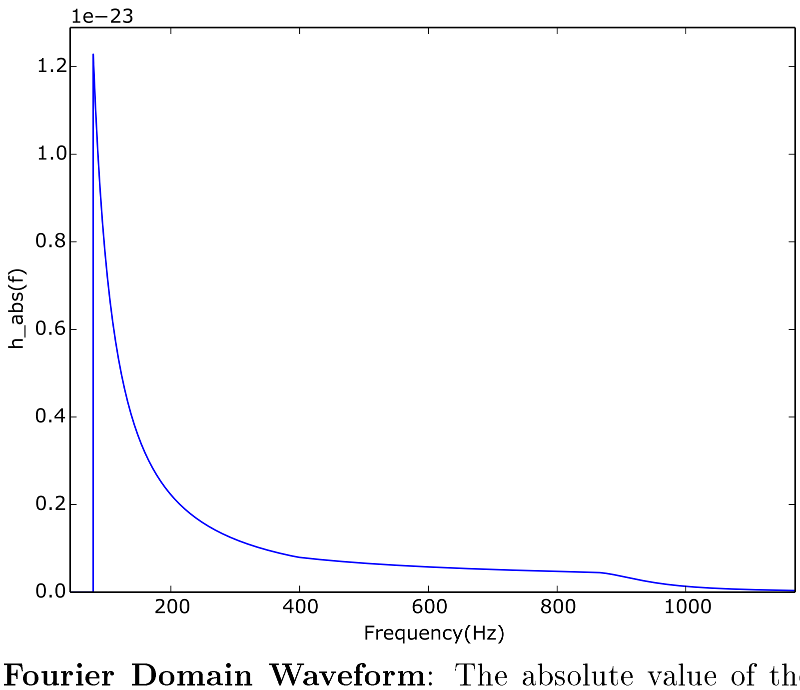


Figure 2.1: **Fourier Domain Waveform**: The absolute value of the Fourier domain waveform for a binary with component masses of $10M_{Sun}$ each.

2.1.2 Detector Noise and Power Spectral Density

The detection of gravitational waves is highly sensitive to the noise present in the detectors used for detecting them. As stated in [16], the main sources of this noise are thermal noise, sensor noise and quantum noise. The amplitude of the resultant noise in the detector typically ranges from the order of that of the signal to two or three orders of magnitude higher. Thus, we can essentially say that the GW signal we want to detect, is "buried" in the noise. It is therefore important to model and characterize this noise as accurately as possible, so that a correspondingly accurate detection can be claimed. The following figure shows a noise realization for the detector and the corresponding power-spectral density (PSD). The PSD is colored and one-sided.

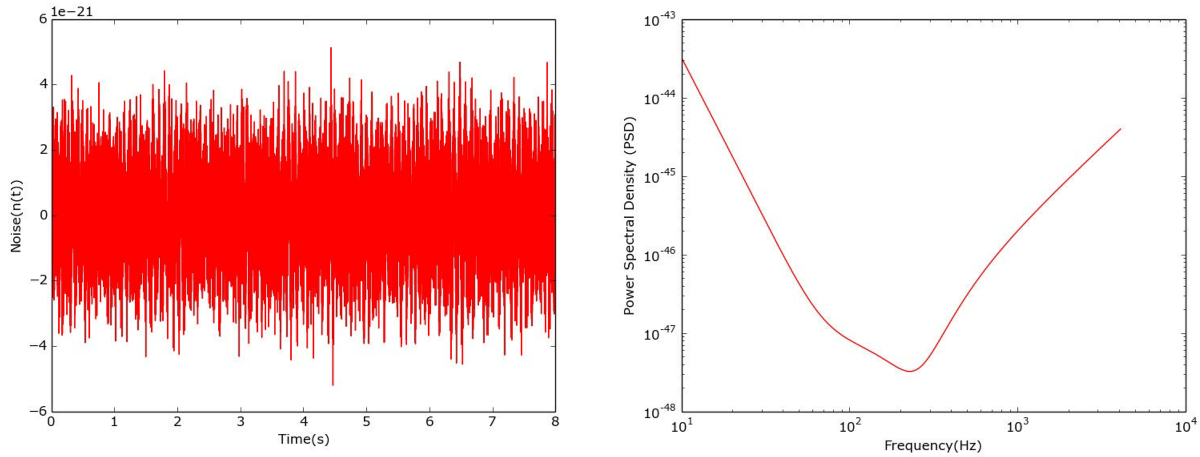


Figure 2.2: **Noise and Power Spectral Density for a detector**: The above figures show the noise of the LIGO detector in the time domain (left) and the corresponding power spectral density (right).

In the following sections, the Fourier domain noise of the detector will be denoted by $n(f)$ and the PSD by $S(f)$.

2.2 Bayesian Inference

The idea behind Bayesian inference simply follows from Bayes' rule. Consider two hypotheses A and B. Then, the joint probability of both A and B being true is given as,

$$P(A,B) = P(A)P(B|A) or P(A,B) = P(B)P(A|B) \tag{2.3}$$

From this we can derive Bayes' rule as:

$$P(A|B) = \frac{P(A)P(B|A)}{P(B)} \tag{2.4}$$

The left-hand side of the above equation, $P(A|B)$, is called the posterior probabilty of A given B. The probability $P(B|A)$, on the right, is the probabilty of B given A and is called the likelihood. $P(A)$ is the prior probability for A while $P(B)$ serves as a normalization constant, which is usually ignored.

Applying this analysis to our case, the hypotheses A and B become the parameters, $\vec{\theta}$[2], and data, X, respectively. So, the posterior probability that we want to obtain, that is $P(\vec{\theta}|X)$, will be given as,

$$P(\vec{\theta}|X) = \frac{P(\vec{\theta})P(X|\vec{\theta})}{P(X)} \tag{2.5}$$

Here, the likelihood $P(X|\vec{\theta})$, denotes the probability of the data given the parameters, $P(\vec{\theta})$ is the prior probability on the parameters which, in this work, is taken to be uniform over the range of parameters.

If $h(f)$ and $n(f)$ denote the gravitational-waveform template[3] and Fourier noise of the detector respectively, then the data $x(f)$ is simply,

$$x(f) = h(f) + n(f) \tag{2.6}$$

$x(f)$ is also called an injection. The functional form of the likelihood $L(\tilde{\theta}|x)$ that we use in our analysis, is then given as,

$$L(\vec{\theta}|x) = e^{-\int \frac{(x(f)-h_{\vec{\theta}}(f))^2}{S(f)} df} \tag{2.7}$$

where, $h_{\vec{\theta}}(f)$ is the waveform template for a particular value of parameters, $\vec{\theta}$, and $S(f)$ is the power-spectral density for the noise realization $n(f)$.

2.3 Markov Chain Monte Carlo (MCMC)

Markov Chain Monte Carlo, or MCMC for short, as the name suggests is Monte Carlo integration carried out over Markov chains. MCMC methods are one of the most reliable and efficient for parameter estimation, especially when the space of parameters to be sampled is large and multi-dimensional. The book by Gilks et al [7] provides an excellent introduction to the same and the concepts defined below are referred from it. We begin by stating what a Markov chain is.

2.3.1 Markov Chain

A sequence of random variables $X_0, X_1, X_2, X_3, ...$, such that at each time $t \geq 0$, the state X_t is sampled from a distribution $P(X_t|X_{t-1})$ which depends only on the previous state of the chain, X_{t-1}, is called a Markov chain.

[2] where the arrow denotes more than one parameter

[3] for a particular value of the parameters

2.3.2 Monte Carlo integration

Lets say that we have to evaluate the expectation value, $E[f(x)]$, of a function f. Monte Carlo integration calculates this by drawing samples $X_t, t = 1, ..., n$ from a proposal distribution and then approximating,

$$E[f(x)] \approx \frac{1}{n}\sum_{i=1}^{n} f(X_t) \tag{2.8}$$

When the samples X_t are independent, the approximate sample mean approaches the true expectation value as n increases. This is the basic theme for Monte Carlo integrations.

2.3.3 Metropolis-Hastings Algorithm

The MCMC method used in the parameter estimation pipeline for this work depends on the famous algorithm first given by Metropolis [13], and Hastings [9] who later extended it to a general case. The algorithm is fairly simple yet it can sample and give accurate posteriors, for most distributions, efficiently. The algorithm can be described as follows: Let L be the likelihood function for a given problem.

1. Choose a point θ_1 at random in the parameter space.
2. Choose another point θ_2 based on a proposal distribution $q(\theta_2|\theta_1)$, dependent only on θ_1. (This ensures the chain is Markovian. Usually the proposal distribution is a Gaussian centered around θ_1)
3. Generate a random number $r \in (0, 1)$ from a uniform distribution.
4. If $r < min.(1, \frac{L(\theta_2)}{L(\theta_1)})$, then make a transition to θ_2, else stay at θ_1.
5. Repeat steps $2 - 4$.

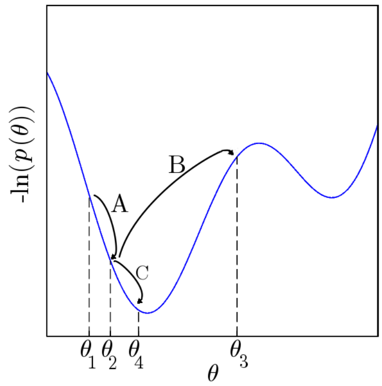


Figure 2.3: **Implementing MCMC**:The Metropolis-Hastings algorithm showing how the moves are proposed from a current point θ_1 to the next point θ_2 and so on.`https://inspirehep.net/record/1283811/plots`

2.4 Ensemble MCMC Methods

Instead of employing a single MCMC chain to obtain the posterior distribution for the parameters, another method or extension to this, which is used in this work, is to employ multiple MCMC chains to sample the parameter space. Such a set of MCMC chains, having different initial locations in the parameter space, is called an ensemble MCMC sampler. This approach has two major advantages-Firstly, having more than one chain ensures that a larger area of the parameter space is sampled. Secondly, and more importantly, we can be more sure of the fact that the posterior actually includes the true maximum likelihood. This is because with a single chain it is probable, and also as will be later discussed, that it converges to a local maximum and not the global one which corresponds to the injected parameters.Also, as shown in [8], ensemble methods computationally outperform their single-chain MCMC counterparts. The ensemble-sampling method used here is such that the updation of the state of any chain in the ensemble, depends on the state of the other chains in the same ensemble. The specific moves proposed in this ensemble-sampler method are described in general below. The specific values assumed and results obtained will be discussed later.
Let $\vec{X}$ be an ensemble of L MCMC chains or walkers. That is, $\vec{X} = (X_1, X_2, X_3, ..., X_L)$. One MCMC step of the ensemble consists of updating all the L walkers. At any stage, the next position of the k^{th} walker, X_k, depends on the current positions of the other walkers in the ensemble. The other walkers form a complementary ensemble,

$$\vec{X_{[k]}}(t) = (X_1(t+1), X_1(t+1), ..., X_{k-1}(t+1), X_{k+1}(t), ...X_L(t)) \tag{2.9}$$

There are three types of moves or updations given in [8] and used in the ensemble-sampler here:

1. **Stretch**: In a stretch move, the walker X_k is updated using a walker from the complementary ensemble $\vec{X_{[k]}}$, say X_j. The walker X_j is chosen at random from the complementary set. The proposed move is of the form,

$$X_k(t) \rightarrow Y = X_j + Z(X_k(t) - X_j) \tag{2.10}$$

   Here Z is a scaling factor (>1) which can be adjusted to improve performance.

2. **Walk**: The walk move involves choosing a subset S ($|S| \geq 2$) from $\vec{X_{[k]}}$. We calculate the mean, $\bar{X_S}$, of the walker positions in S as,

$$\bar{X_S} = \frac{1}{|S|} \sum_{X_j \in S} X_j \tag{2.11}$$

   We then construct W, with mean zero and covariance same as walkers $X_j \in S$ as,

$$W = \sum_{X_j \in S} Z_j (X_j - \bar{X_S}) \tag{2.12}$$

   where Z_j are univariate standard normals. The proposed move is then,

$$X_k(t) \rightarrow Y = X_k(t) + W \tag{2.13}$$

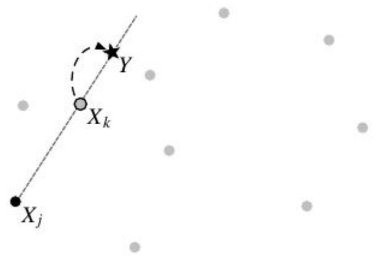


Figure 2.4: **The Stretch move:** Picture from Goodman and Weare(2010)

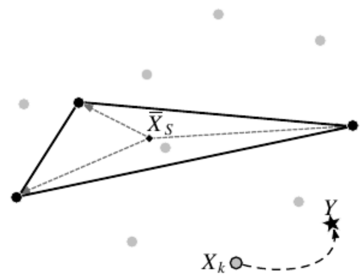


Figure 2.5: **The Walk move:** Picture from Goodman and Weare(2010)

3. **Replace**: The third type of move is the replace move. This move is the same as the usual Metropolis-Hastings MCMC move as defined above.

Along with proposing these different types of moves, the ensemble sampler is different from the single-chain MCMC in that it is adaptive with respect to the step size or variance of the Gaussian proposal density used to propose the next move in the chain.

Chapter 3

Methods

The parameter estimation pipeline developed here completely relies on the MCMC algorithm previously described. So we will look at, in detail, how the algorithm is exactly used in this case. The use of single-chain MCMC routines, restricts the range of the parameter space sampled and, as shown later, turns out to be inaccurate and inefficient. In order to overcome this, we try to make use of the Multiprocessing module from Python, whereby we can employ multiple MCMC chains. These chains have different initial positions in the same parameter space and help in sampling more of the space. Finally, since the results from using the Multiprocessing module were not adequate enough we also implemented the ensemble MCMC sampler as described in the previous chapter. Let us begin by looking at how the gravitational-wave data is generated and used in the pipeline.

3.1 Creating an injection

3.1.1 Generating the Waveform

Gravitational-wave data is generated by combining the waveform template with the noise from the detector. Waveforms are obtained by using appropriate packages in the LIGO Algorithm Library(LAL) [4], which contains routines in the language C, for carrying out data analysis in gravitational-wave astronomy. These packages in LAL require an input from the user for the values of the source parameters and return the waveform, $h_{template}(f)$[1], corresponding to these values. Some of the parameter inputs required by the code in LAL are as follows,

- m_1, m_2 : The component masses of the binary.
- $S_{1x}, S_{1y}, S_{1z}, S_{2x}, S_{2y}, S_{2z}$: The spins of the individual components along all 3 axes.
- d_L : Luminosity distance of the binary..
- i : Inclination angle of the binary with respect to the line of sight.
- θ : Declination of the binary.
- ϕ : Right ascension of the binary.

[1] both time and frequency domain waveforms can be generated

- ψ : Polarization of the GW signal.
- ϕ_c : Reference phase.
- t_c : Reference time of coalescence.

In principal, these are all the parameters that one can try and get estimates for. In addition, there are certain other arguments that one is expected to pass in LAL. These include the approximant to be used in generating the waveform, the PN order upto which the template needs to be evaluated and the domain[2] of the waveform.

In the work done so far, we have considered non-spinning binaries only. This implies that the six spin parameters stated above are set to zero. Also we have majorly used the highly accurate phenomenological waveform 'IMRPhenomB' upto the highest available PN order, that is 3.5PN. All our calculations have been carried out in the Fourier domain. A typical waveform looks as follows,
figure

3.1.2 Generating the Detector Noise

Once again, in order to create a noise realization, $n(f)$, corresponding to a detector, we use the packages in LAL. Information about various noise models is built-in in LAL and we just need to provide the name of the noise model we need to use. For this work we resorted to using the Advanced LIGO detector noise, whose PSD is as shown in the previous section.

3.1.3 Data

Finally, we create the data, $x(f)$ by summing the template $h_{template}(f)$ and noise $n(f)$ obtained from above.

$$x(f) = h_{template}(f) + n(f) \tag{3.1}$$

This data is called an injection.

3.2 Implementing the MCMC algorithm

Once the data is generated, the next step is to implement the Metropolis-Hastings algorithm to generate the MCMC chain for the parameter estimation. As discussed in the previous section, we need to consider a certain prior probability, $P(\theta)$, for the parameters and also need to have a functional form for the likelihood $L(\theta|x)$ (eq.2.7). We assume the prior probability to be uniform over a fixed range of the individual parameters and zero for any value outside this range.

Thus, $P(\theta) = Uniform(\theta_{min}, \theta_{max})^3$ and $L(\vec{\theta}|x) = e^{-\int \frac{(x(f)-h_{\vec{\theta}}(f))^2}{S(f)} df}$. Since the numbers we obtain for the likelihood in a gravitational-wave analysis are extremely small and we

[2] time or frequency

[3] where we specify $\theta_{min}, \theta_{max}$ for each parameter

might encounter underflow errors, it is best to work with logarithm of the likelihood which is what we use. The Loglikelihood will simply be the exponent in the expression given above.The algorithm then proceeds as:

1. We choose a random point[4] to begin with in the multi-dimensional parameter space, say $\vec{\theta_1}$.
2. We use a proposal distribution which is a Gaussian having a specified variance[5] along each dimension of the parameter space and a mean equivalent to the position $\vec{\theta_1}$.
3. The next proposal point $\vec{\theta_2}$ is chosen using this Gaussian proposal distribution.
4. A random number $r \in Uniform(0, 1)$ is chosen.
5. We evaluate Loglikelihood values at the positions $\vec{\theta_1}$ and $\vec{\theta_2}$.
6. The comparison step in the MCMC algorithm then turns out to be - If $log(r) < logL(\vec{\theta_2}) - logL(\vec{\theta_1})$, then make a transition to $\vec{\theta_2}$. Else we stay at $\vec{\theta_1}$.
7. We repeat this, each time using the proposal distribution to propose a new point from the current point, for a large number of steps such that the chain will converge.

3.3 Multiprocessing from Python

Implementing the MCMC with a single chain can lead to inaccurate results as will be shown later. This suggests that we try and use multiple MCMC chains, all having different starting positions in the parameter space. This is most efficiently done by using the Multiprocessing module in Python [11].
Multiprocessing allows the user to parallely use multiple processors of a computer and one can submit multiple jobs with different initial values. This is usually done using the Pool object of the module, which asks the user for the number of jobs to be submitted and runs instances of the same program parallely. Thus, by supplying the number of MCMC chains we need as an argument to the Pool object, we can run those many chains on the same parameter space simultaneously. The results obtained by using this module will be shown in the results section.

3.4 Ensemble Sampler

Moving on from using the Multiprocessing module, we implemented the Ensemble sampler, so that the parameter space could be sampled more efficiently. The proposal moves used in this ensemble sampler have been explained in the previous section. Another aspect of this method is that it is adaptive with respect to the step sizes used in the Gaussian proposal distribution as explained earlier. The steps for the updation of each chain are similar to that of the simple single chain MCMC, except for the type of proposal moves - the stretch,walk or replace. As said earlier, one step for the ensemble sampler

[4]The random number generator was used from Python's NUMPY module
[5]Also referred to as the MCMC step size

involves updating the positions of all the walkers one-by-one in a cycle. In this algorithm, a counter serially selects the walkers for updation. In our case, if the counter is a multiple of 2, the walker corresponding to the counter undergoes a stretch move. If it is a multiple of 3, it undergoes a walk move, else a replace move is carried out.

Chapter 4

Results

4.1 A simple exercise

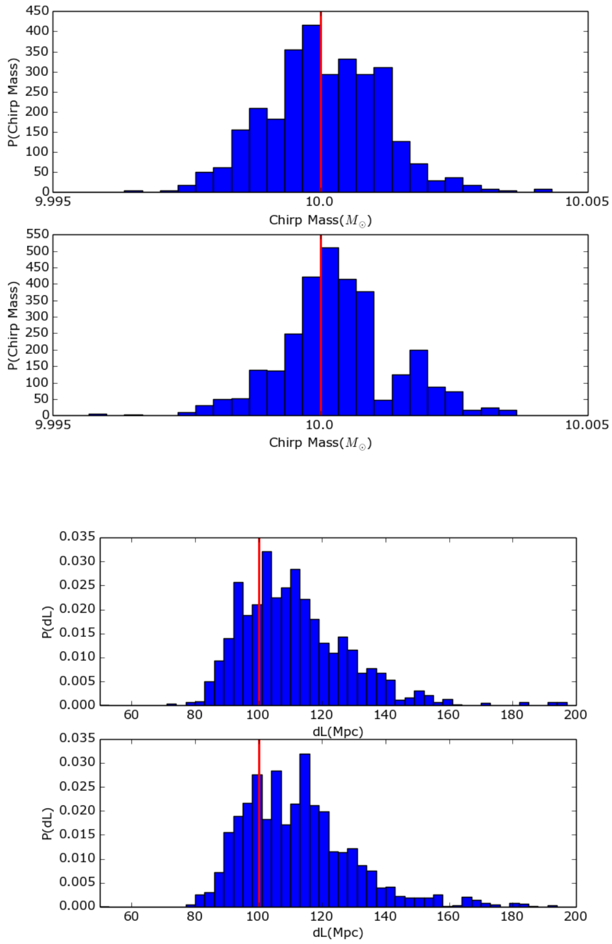


Figure 4.1: **Time-Frequency domain comparison:**Plots for the comparison between the time(top panels) and frequency domain(bottom panels) posteriors for the chirp mass (left) and luminosity distance (right).

The above plots are the 1-d posterior distributions obtained for the chirp mass and luminosity distance in the case of the data being a Newtonian i.e. 0PN or leading order waveform with added Gaussian white noise.

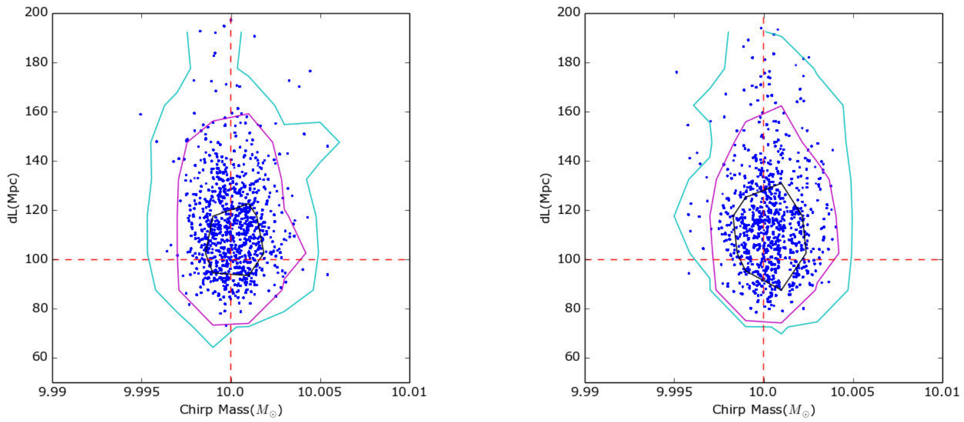
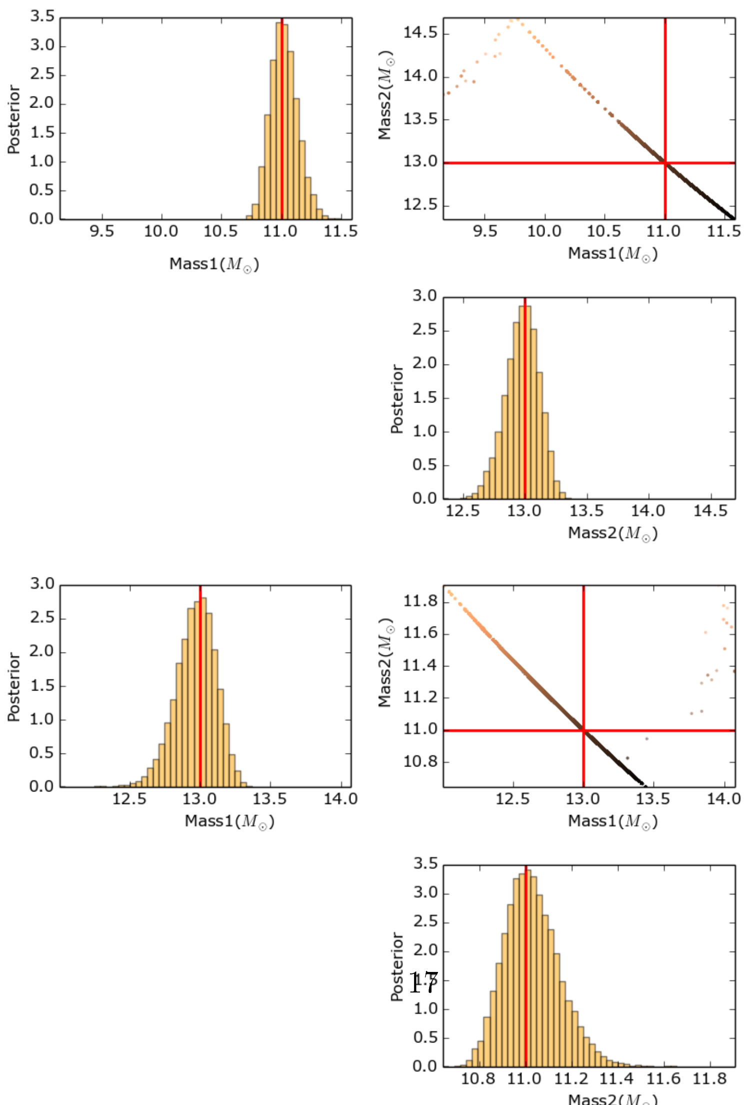


Figure 4.2: **Time-Frequency domain comparison:**Plots for the comparison between the time(left) and frequency domain(right) 2-d posteriors of the chirp mass and luminosity distance for the same system as above.The confidence levels are shown by colored lines:1σ(black),2σ(magenta),3σ(cyan)

4.2 Single Chain MCMC



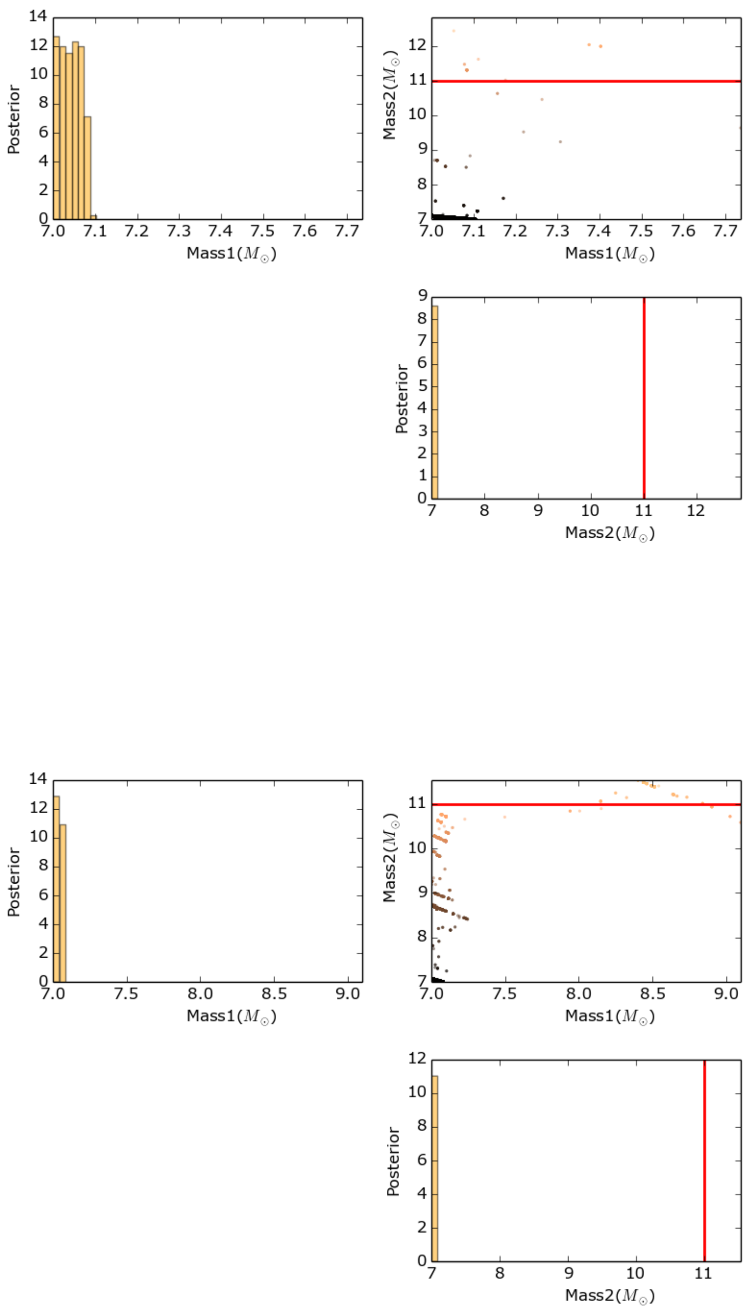


Figure 4.3: **Single chain MCMC posteriors:**Posterior plots for the component masses for a system with $m_1 = 13M_\odot$ and $m_2 = 11M_\odot$. The first two plots show that the chain has converged to the injected values while the next two have converged to the boundary.

4.3 Multiprocessing Runs

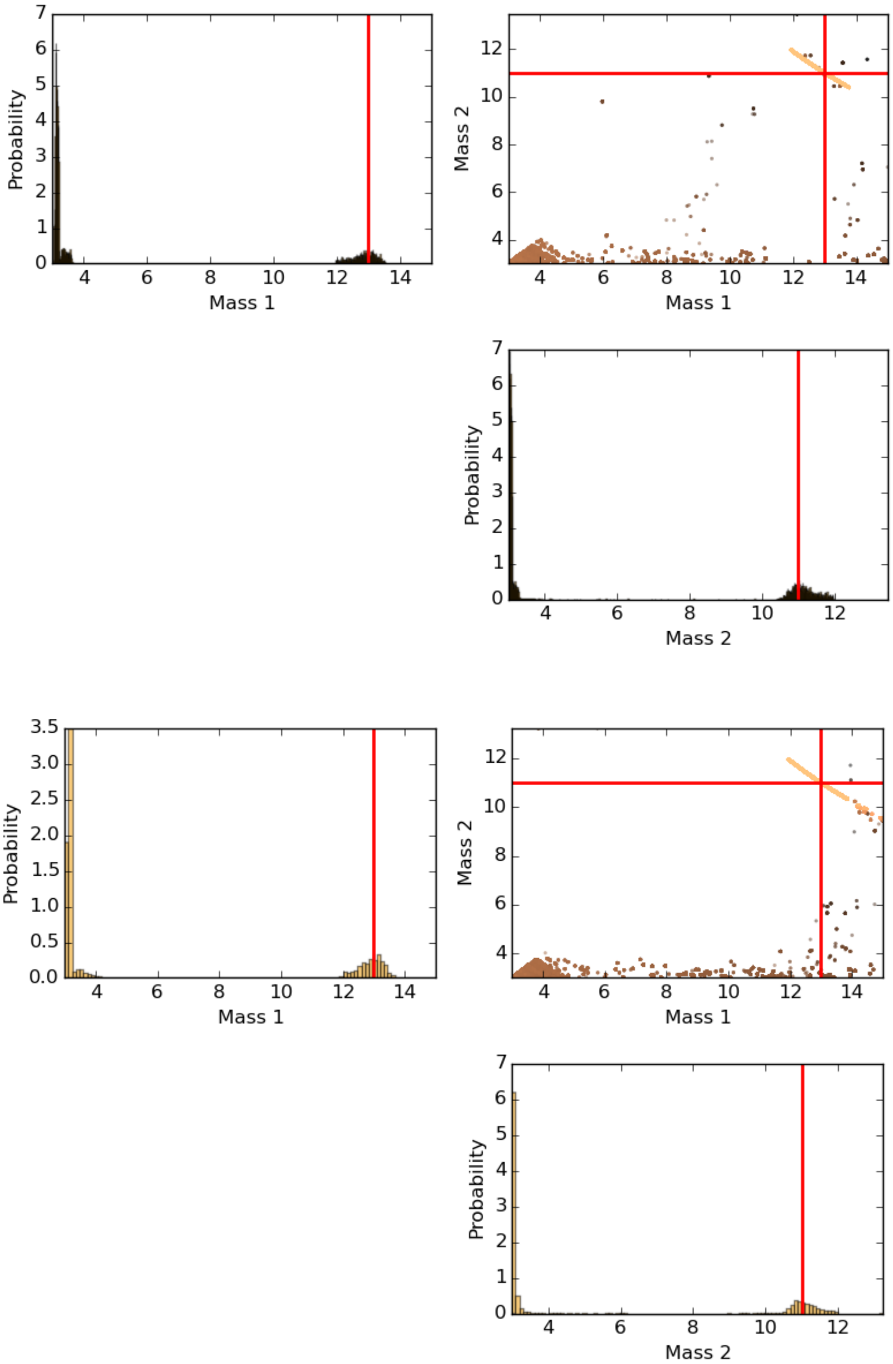


Figure 4.4: **MCMC using Multiprocessing**Multiple chains are used via the Multiprocessing module on the same system. Most of the chains have again converged to the boundary while a few have converged to the injected values

4.4 Ensemble Sampler

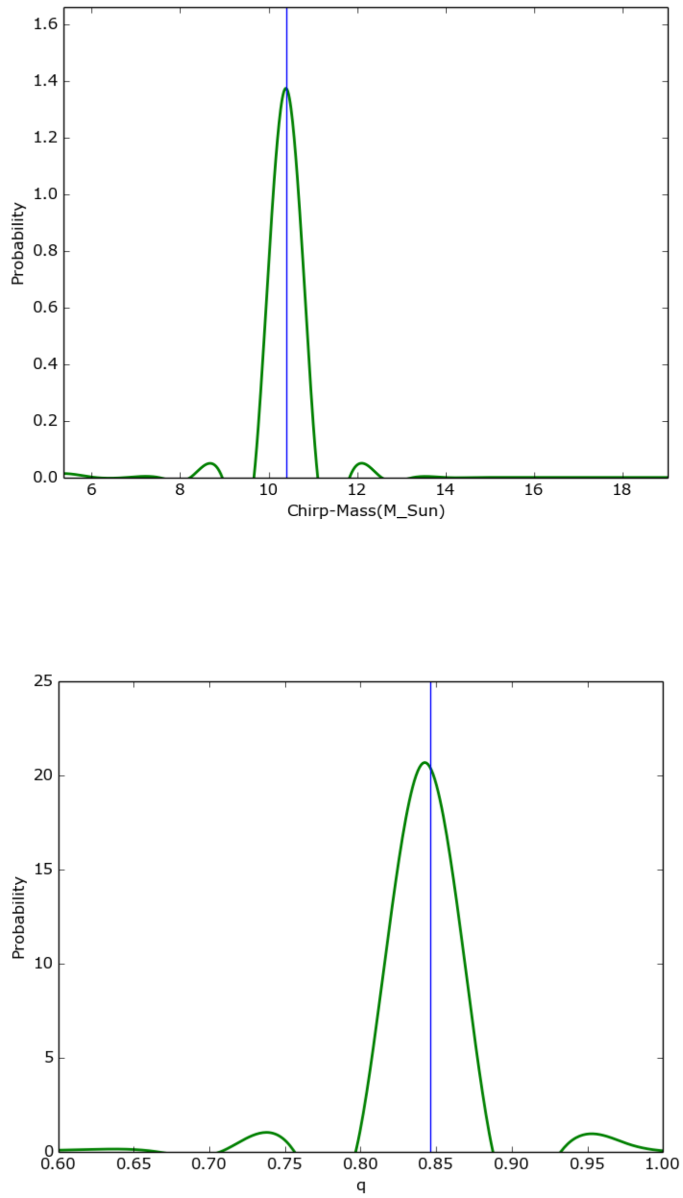
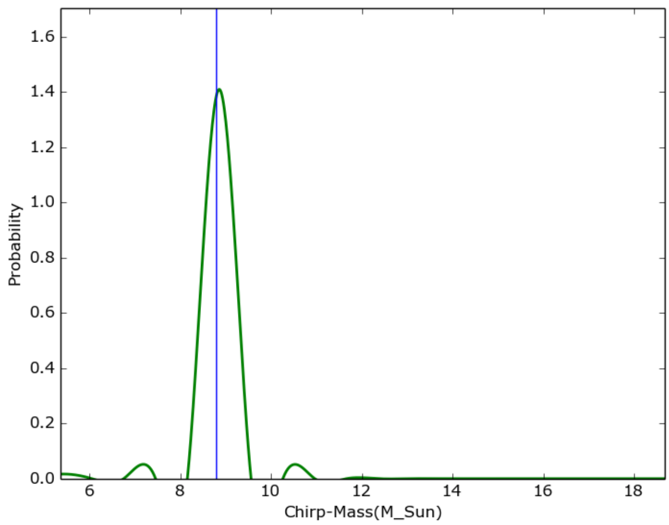


Figure 4.5: **Chirp Mass and Mass ratio posteriors:** Posteriors for the $13M_{\odot}$,$11_{\odot}$ system



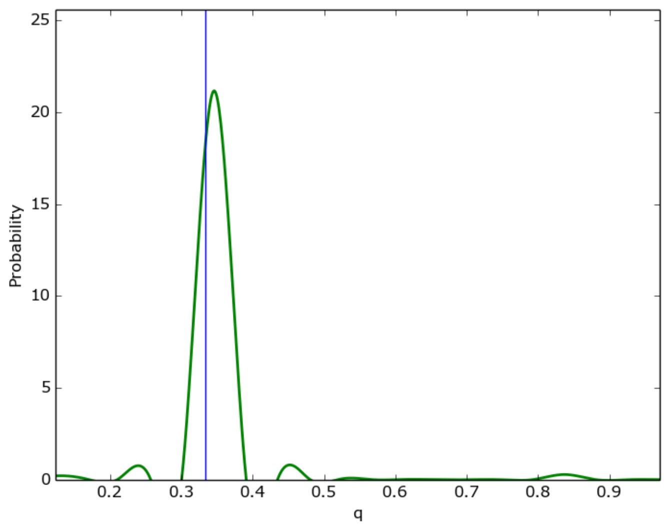
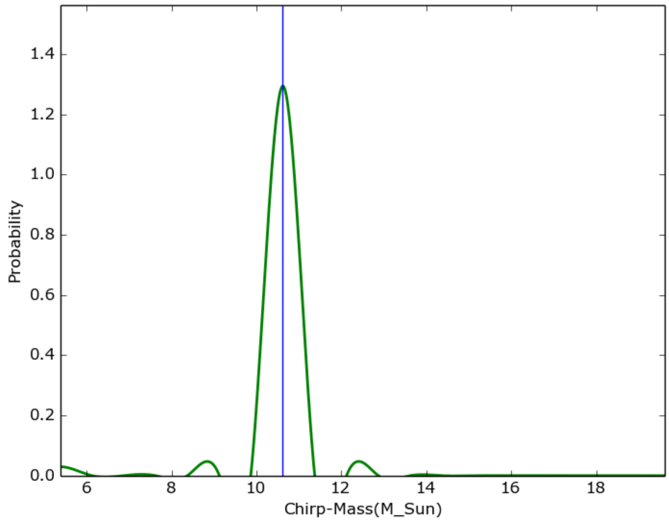


Figure 4.6: **Chirp Mass and Mass ratio posteriors:** Posteriors for the $6M_{\odot}$,$18_{\odot}$ system



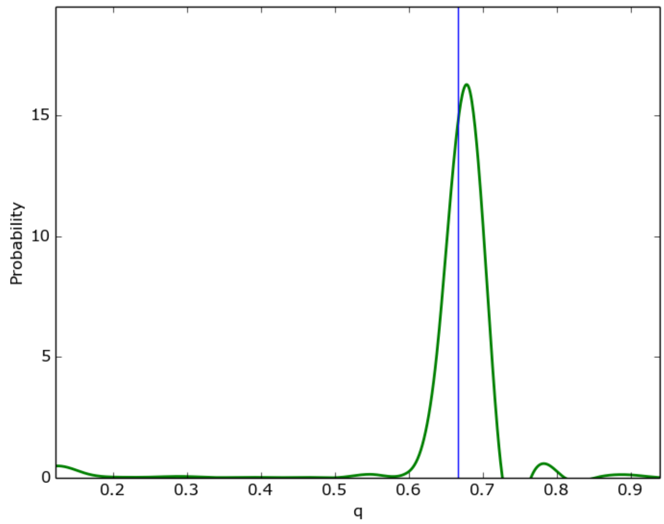


Figure 4.7: **Chirp Mass and Mass ratio posteriors:** Posteriors for the $10M_{\odot}$,$15_{\odot}$ system

Chapter 5

Discussion

We discuss our results shown in the previous section and also the reasons for some of the inaccurate posteriors obtained. As shown in Section 4.1, the posteriors in the time as well as frequency domains for the same system is similar. This is expected since doing a parameter estimation in either domains[1], does not change the parameters itself. The minor differences that we can see in the plots, arise from the fact that the number of samples taken is small, 10^5 in this case. As we increase the number of samples the two posteriors will look more and more similar.

After having done this analysis, we moved onto the more realistic waveforms and noise models, in particular the IMRPhenomB waveform and the colored noise of the Advanced LIGO detector, respectively. The posteriors obtained from using these waveforms are shown in section 4.2. We found that while some of the chains did converge to the injected values, most of the other chains converged to the boundary of the prior as shown in the other plots of the same section. This led to investigating the nature of the likelihood being sampled for these MCMC chains. The following figure shows the numerical and analytical plots of the likelihood for the same system.

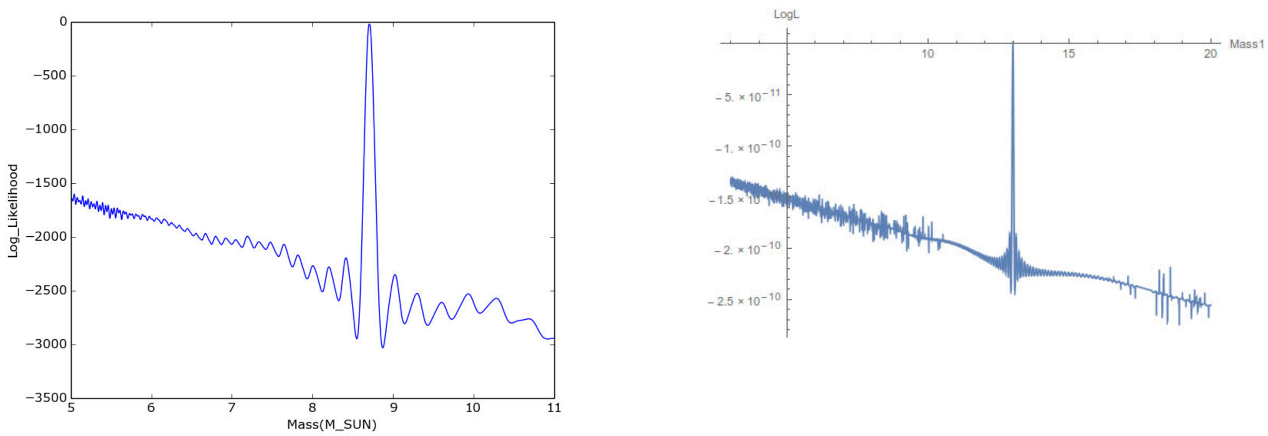


Figure 5.1: **Numerical and analytical likelihoods**

[1] Here it is possible only in the case of white noise

The analytical likelihood in the above figure has been computed in the leading PN order. Even in this case, we can see that the likelihood does rise towards the boundary of the prior and hence it seems that this is not a numerical artifact. In other words, the boundary behaves as a local maxima during the sampling of the mass parameter space, causing many chains to converge and get “stuck” in this region. This is also the reason why most of the runs using multiple chains from the Multiprocessing module converged to the boundary of the prior. Thus, in order to get over this problem, we had to resort to a more sophisticated, adaptive algorithm which is the one offered by the ensemble sampler.

We have provided the plots of the posteriors on the chirp mass and mass ratio, obtained from the ensemble sampler, in Section 4.4. We can clearly see that now almost all the chains or MCMC walkers converge to the injected value with great accuracy. This is because, during the updation of walkers in the ensemble sampler, the proposed moves of stretching and walking force the state of one walker to be dependent on that of the others. Thus, even if a few walkers are converging to the injected values of the parameters, they can actually “pull” the other walkers stuck in other regions by this algorithm. We also tested to see what is the approximate, optimum number of walkers needed to sample the parameter space accurately and efficiently. The posterior plots using 10, 50, 100 and 1000 walkers is given below.

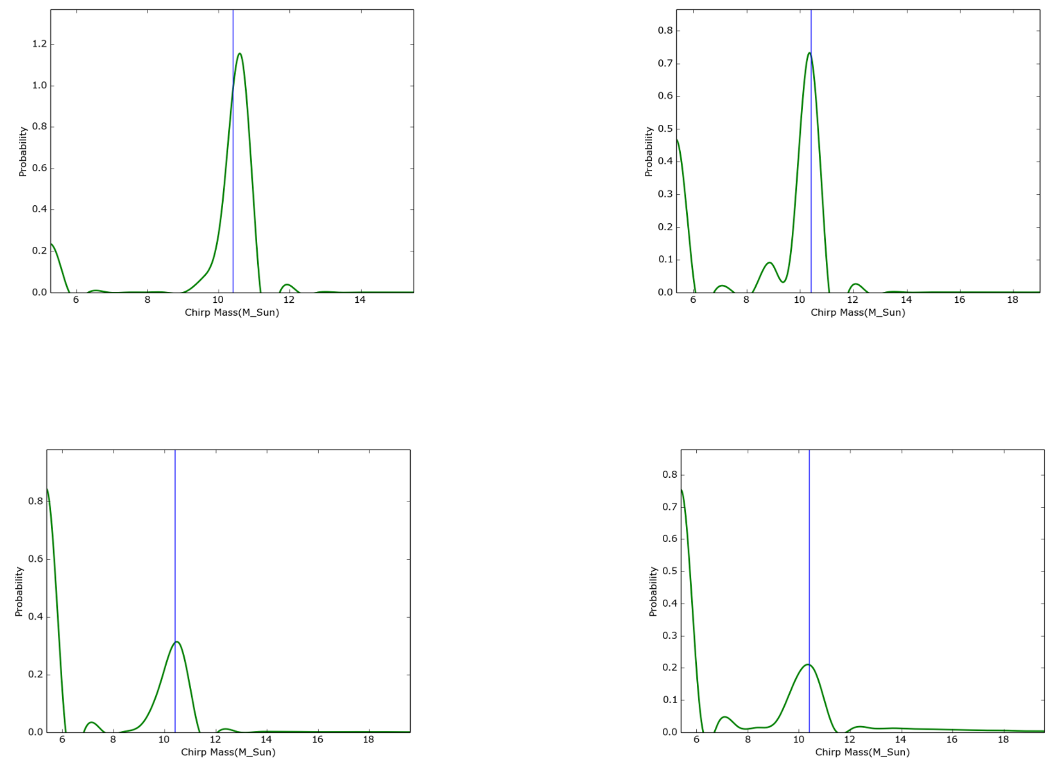


Figure 5.2: **Chirp Mass posteriors for different number of walkers:**10 Walkers(top left),50 Walkers(top right),100 Walkers(bottom left),1000 Walkers(bottom right)

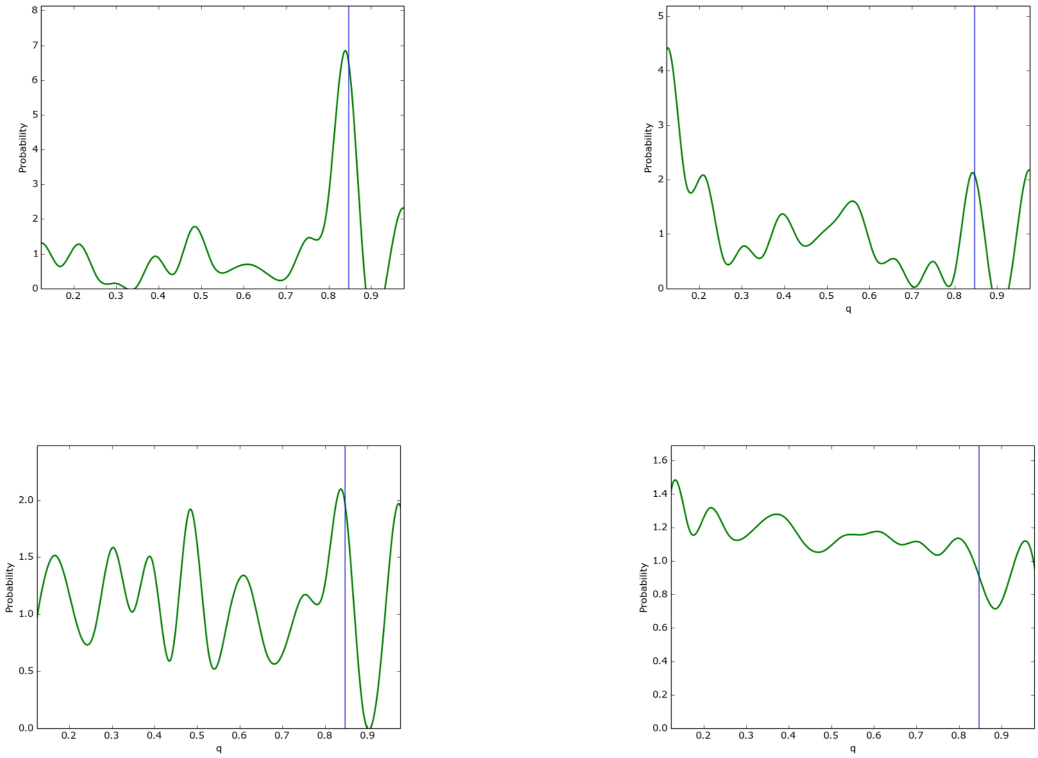


Figure 5.3: **Mass ratio posteriors for different number of walkers:**10 Walkers(top left),50 Walkers(top right),100 Walkers(bottom left),1000 Walkers(bottom right)

For the above plots, we kept the number of MCMC steps to be fixed and varied the number of walkers. Thus, more the number of walkers meant that each walker underwent fewer steps. From these plots it is clear that, the accuracy of the ensemble sampler is higher when there are a fewer number of walkers undergoing greater number of steps.

The ensemble sampler has turned out to be the most efficient and accurate way to estimate the parameters so far and we will be using the same to carry out parameter estimation over even higher-dimensional spaces to estimate more parameters.

Conclusion

To summarize, we created and implemented a Markov Chain Monte Carlo (MCMC) parameter-estimation code based on the principles of Bayesian inference, and obtained posteriors for parameters such as the chirp mass, luminosity distance and the component masses of compact binaries. This single-chain MCMC code was then parallelized using the Multiprocessing module in Python. In order to sample the parameter space more accurately and efficiently, we implemented the ensemble sampling technique as described by Goodman and Weare [8].

Now that we have obtained posteriors for some of the intrinsic parameters such as the component masses etc., the next step would be to try and estimate some extrinsic parametrs such as the sky location, inclination angle and polarization. In order to estimate the sky location, we will need to extend this code to incorporate a multi-detector analysis. For parameters such as the polarization, we need to fold in the antenna pattern functions for the detector and the carry out the analysis. The final goal will be to implement the estimation pipeline for the full 9-parameter space, in case of non-spinning binaries. Another possible way of improving the estimation would be to search for even more efficient techniques, compared to the ensemble sampler, to sample a large and multi-dimensional space.

References

[1] R. Balasubramanian and S.V. Dhurandhar. Estimation of parameters of gravitational wave signal from coalescing binaries. *Phys.Rev.*, D57:3408–3422, 1998.

[2] R. Balasubramanian, B.S. Sathyaprakash, and S.V. Dhurandhar. Estimation of parameters of gravitational waves from coalescing binaries. *Pramana*, 45:L463–L470, 1995.

[3] Nelson Christensen and Renate Meyer. Markov chain monte carlo methods for bayesian gravitational radiation data analysis. *Phys. Rev. D*, 58:082001, Sep 1998.

[4] LIGO Scientific Community. Ligo algorithm library. `https://www.lsc-group.phys.uwm.edu/daswg/projects/lalsuite.html`.

[5] Neil J. Cornish and Tyson B. Littenberg. Tests of bayesian model selection techniques for gravitational wave astronomy. *Phys. Rev. D*, 76:083006, Oct 2007.

[6] Guillaume Faye, Sylvain Marsat, Luc Blanchet, and Bala R. Iyer. The third and a half post-Newtonian gravitational wave quadrupole mode for quasi-circular inspiralling compact binaries. *Class.Quant.Grav.*, 29:175004, 2012.

[7] W Richardson Gilks, Sylvia Richardson, and David J Spiegelhalter. (1996). markov chain monte carlo in practice.

[8] Jonathan Goodman and Jonathan Weare. Ensemble samplers with affine invariance. *Communications in Applied Mathematics and Computational Science*, 5(1):65–80, 2010.

[9] W. K. HASTINGS. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

[10] R. A. Hulse and J. H. Taylor. Discovery of a pulsar in a binary system. *Astrophysical Journal*, 195:L51–L53, January 1975.

[11] Python Standard Library. Multiproccessing in python. `https://docs.python.org/2/library/multiprocessing.html`.

[12] Michele Maggiore. *Gravitational Waves: Volume 1: Theory and Experiments*, volume 1. Oxford university press, 2007.

[13] 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), 1953.

[14] Christian Rover, Renate Meyer, and Nelson Christensen. Bayesian inference on compact binary inspiral gravitational radiation signals in interferometric data. *Class.Quant.Grav.*, 23:4895–4906, 2006.

[15] Christian Rover, Renate Meyer, and Nelson Christensen. Coherent Bayesian inference on compact binary inspirals using a network of interferometric gravitational wave detectors. *Phys.Rev.*, D75:062004, 2007.

[16] B. S. Sathyaprakash and B. F. Schutz. Physics, astrophysics and cosmology with gravitational waves. *Living Reviews in Relativity*, 12:2, Mar 2009.

[17] DS Sivia and John Skilling. Data analysis: a bayesian tutorial. *AMC*, 10:12, 1996.

[18] Michele Vallisneri. Use and abuse of the Fisher information matrix in the assessment of gravitational-wave parameter-estimation prospects. *Phys.Rev.*, D77:042001, 2008.