Laboratory and Computational Physics 3: Bayesian Inference and Gravitational-wave data analysis

---

Authors:
Christine Y.S. Lee
Kok Hong (Jacky) Thong
Yutong (Tracy) Bu

Part 3 Laboratories, School of Physics
The University of Melbourne

August 17, 2023

Contents

1 Introduction

Gravitational waves (GW) are gravitational fluctuations propagating across the fabric of spacetime as a consequence of accelerating massive bodies. This phenomenon was postulated by Albert Einstein in 1915 and exactly 100 years later in September 2015, the two detectors of the Laser Interferometer Gravitational-Wave Observatory (LIGO) at Hanford and Livingston detected the very first gravitational-wave signal (GW150914) in history during their first observing run, O1. This signal originated from a merger between two black holes at a luminosity distance of 410^{+160}_{-180} Mpc, with initial masses of $36^{+5}_{-4}M_{\odot}$ and $29^{+4}_{-4}M_{\odot}$ [1]. A Nobel prize was awarded for this ground-breaking discovery in 2017 for it has opened up countless windows of opportunities for new research in astronomy.

For many years, scientists have been exploiting electromagnetic waves emitted by astrophysical sources to image and understand astrophysical processes that are happening in the vast universe. One of many challenges in traditional astronomy, which observes using electromagnetic radiation, is that telescopes must be able to capture sufficient photons emitted by the target source to obtain decent-quality spectra or images for analysis. This makes it extremely difficult to image and study faraway sources given that photons propagating from such distances will encounter numerous obstructions throughout their journey towards earth. Fortunately, GW does not suffer from this problem; GW does not get absorbed or modulated by the intervening material, although it still loses flux (power over area) with the square of the distance like all waves do. Nevertheless, by exploiting the unobstructed propagation of GW, astronomers are now able to explore a new regime of space previously inaccessible by electromagnetic radiation and retrieve information on obscure astrophysical sources, like black holes which do not emit photons. This can lead to new and profound discoveries.

Between 2015 and 2020, the Advanced LIGO detectors in Hanford and Livingston have completed three observing runs: O1, O2 and O3 [2–4], and in addition conducted joint searches with Italian partner Virgo in the late stages of O2 and the whole of O3. The Kamioka Gravitational-Wave Detector (KAGRA) also began observing in February 2020, towards the end of O3. In all, 90 GW events from compact binary coalescences (CBCs) are collectively observed throughout the three completed observing runs. The transients detected so far include mergers of binary black holes, binary neutron stars and neutron star-black hole binaries.

Over the course of this lab, you will develop an understanding of Bayesian inference, the Markov chain Monte Carlo algorithm, and gravitational-wave data analysis where you work with real LIGO data.

2 Background

2.1 Generation of GW: Quadrupole moment

Before we describe GW mathematically with general relativity, let's discuss its origin by comparing it with electromagnetic radiation.

You may recall from your electrodynamics courses that one can approximate potentials at large distances using multipole expansions [5]. For a given charge configuration with potential

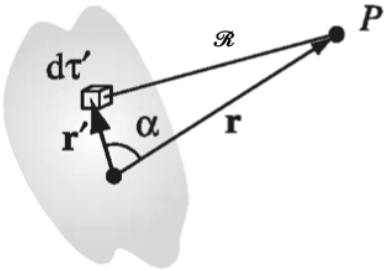

Figure 1: An arbitrary charge configuration as seen by an observer at point P (Figure adapted from Griffiths: Introduction to Electrodynamics).

(see Figure 1)

$$V(\mathbf{r}) = \frac{1}{4\pi\varepsilon_0}\int \frac{1}{\mathcal{R}}\rho(\mathbf{r}')d\tau' \tag{1}$$

where $\rho(r)$ is the charge density and

$$\mathcal{R} = r^2 + (r'^2) - 2rr'\cos\alpha \tag{2}$$

is the distance between the point of interest, P and infinitesimal volume $d\tau'$. One can systematically expand $V(\mathbf{r})$ of *any* localised charge distribution in powers of $1/r$. This is known as the **multipole expansion** and is given by

$$V(\mathbf{r}) = \frac{1}{4\pi\varepsilon_0}\left[\frac{1}{r}\int \rho(\mathbf{r}')d\tau' + \frac{1}{r^2}\int r'\cos\alpha\rho(\mathbf{r}')d\tau' + \frac{1}{r^3}\int (r')^2\left(\frac{3}{2}\cos^2\alpha - \frac{1}{2}\right)\rho(\mathbf{r}')d\tau' + ...\right], \tag{3}$$

or more compactly

$$V(\mathbf{r}) = \frac{1}{4\pi\varepsilon_0}\sum_{n=0}^{\infty}\frac{1}{r^{(n+1)}}\int (r')^n P_n(\cos\alpha)\rho(\mathbf{r}')d\tau' \tag{4}$$

where $P_n(\cos\alpha)$ are Legendre polynomials. The integral in the equation above, typically denoted as M_n i.e.

$$M_n = \int (r')^n P_n(\cos\alpha)\rho(\mathbf{r}')d\tau' \tag{5}$$

is n-th multipole moment of charge distribution (of the source).

Now let's break down the terms in Equation 3. You can see that each term has increasing powers of 1/r. At large distances, the multipole expansion is usually dominated by the first term ($n = 0$), otherwise known as the monopole term, which typically appears in the form

$$V_{mono}(\mathbf{r}) = \frac{1}{4\pi\varepsilon_0}\frac{Q}{r} \tag{6}$$

where $Q \equiv \int \rho d\tau$ is the total charge of the system, as you would have seen in first year. The total charge is also known as the **monopole moment** which is conserved and therefore electromagnetic monopolar radiation does not exist.

The second term ($n = 1$) is the dipole potential. According to Figure 1, we can write $r' \cos\alpha = \hat{\mathbf{r}} \cdot \mathbf{r}'$ and therefore we can rewrite the dipole potential as:

$$V_{dip}(\mathbf{r}) = \frac{1}{4\pi\varepsilon_0} \frac{\mathbf{p} \cdot \hat{\mathbf{r}}}{r^2} \tag{7}$$

where

$$\mathbf{p} \equiv \int \mathbf{r}' \rho(\mathbf{r}') d\tau' \tag{8}$$

is the **dipole moment** of the charge distribution. There is no conservation law for dipole (and higher order) moments, thus electromagnetic dipole (and higher order) radiation is possible. Accelerating charges produce changing electric and magnetic fields. The change in electric field induces magnetic fields and vice versa, resulting in propagating electromagnetic waves.

We use the same multipole expansion described above for gravitational potentials, except now $\rho(\mathbf{r})$ denotes the mass-energy density function. In this context, the **monopole moment** is the total mass-energy, which cannot vary in a closed system by mass-energy conservation and therefore monopolar gravitational radiation cannot exist. The **dipole moment** (as seen in Equation 7) is then the center of mass-energy of the system. In the center-of-mass frame, the dipole moment is invariant (zero, if we choose the right center-of-mass frame). Furthermore, the first-order time derivative of the mass dipole is the momentum of the system, which is also a conserved quantity. So again, there is no dipolar gravitational radiation in the centre of mass frame, and therefore in all other frames too since existence of radiation is frame-independent.

Next up is the **quadrupole moment** ($n = 2$ term in Equation 3[1]. Again we use $r' \cos\alpha = \hat{\mathbf{r}} \cdot \mathbf{r}'$ to rewrite the quadrupole potential as:

$$V_{quad}(\mathbf{r}) = -\frac{G}{r^3} \sum_{i,j} Q_{ij} \hat{r}_i \hat{r}_j \tag{9}$$

where Q_{ij} is the quadrupole moment tensor given by[2]

$$Q_{ij} = \int \rho(\mathbf{r}) \left(\frac{3}{2} r'_i r'_j - \frac{1}{2} (r')^2 \delta_{ij} \right) d\tau'. \tag{10}$$

Note the change of constants in Equation 9 and the addition of a negative sign since masses are always positive and the force is attractive, hence negative potential. Each term of the quadrupole tensor measures the distribution of mass (or charge, depending on context) in a plane. Since the quadrupole moment is not a conserved quantity i.e. it can change with time, the quadrupole gravitational radiation is the lowest-order gravitational radiation to exist. The change in quadrupole moment is induced by accelerating masses, and this is what we call GRAVITATIONAL WAVES!

QUESTION 1: Derive Equation 9 from the quadrupole term of Equation 3. *Hint:* use $r' \cos\alpha = \hat{\mathbf{r}} \cdot \mathbf{r}'$ and $\hat{\mathbf{r}} \cdot \hat{\mathbf{r}} = \sum_{i,j} \hat{r}_i \hat{r}_j \delta_{ij} = 1$.

[1] Although not exactly the moment of inertia, the quadrupole moment encodes this information.
[2] δ_{ij} is the Kronecker delta.

Based on our discussion above, we can speculate the types of GW sources. Spherically symmetric variations in mass distribution (e.g. isotropic explosions or collapse) are monopolar/dipolar and therefore do not produce gravitational radiation regardless of how catastrophic the event it. Axisymmetric rotations of massive bodies also do not emit gravitational radiation since the quadrupolar and high moments are not changing with time. Therefore the systems that we are interested are those with asymmetries e.g. coalescing binaries, rotating stars with non-axisymmetric mountains, asymmetric collapses (e.g. supernovae) etc. We discuss these systems and the types of GWs they produce in Section 2.4.

2.2 General relativity: GWs as metric perturbation (EXTENSION)

[**NOTE**: The following 2 paragraphs ***VERY*** briefly summarises the mathematics of GWs - we do not have time, over the course of 2 weeks, to provide you with a crash course on general relativity. So we have added some references at the end if you'd like to do some further reading.]

The existence of gravitational waves is postulated by Einstein's Theory of General Relativity. Einstein's field equations in natural units ($c = G = 1$) are given by

$$R_{\mu\nu} - \frac{1}{2}Rg_{\mu\nu} = 8\pi T_{\mu\nu}, \tag{11}$$

where the left-hand side (LHS) encodes information about the curvature of spacetime while the right-hand side (RHS) has the matter distribution and fields described by the stress-energy tensor, $T_{\mu\nu}$. If the gravitational field is weak, Einstein's field equations can be linearised by assuming that the spacetime metric tensor is given by $g_{\mu\nu} = \eta_{\mu\nu} + h_{\mu\nu}$, where $\eta_{\mu\nu}$ is the Minkowski (flat space) metric and $h_{\mu\nu}$ is a perturbation term. Linearisation in this context arises from the fact that $|h_{\mu\nu}| \ll 1$ such that all perturbation terms with $|h_{\mu\nu}|^k$ with $k > 1$ can be neglected.

This linearised theory then leads to Einsein's weak-field equations

$$\Box\,\bar{h}_{\mu\nu} = -16\pi T_{\mu\nu} \tag{12}$$

for an unknown $\bar{h}_{\mu\nu} = h_{\mu\nu} - \frac{1}{2}\eta_{\mu\nu}h_{\alpha\beta}\eta^{\alpha\beta}$, where $\Box = \eta_{\mu\nu}\partial^\mu\partial^\nu$ is the D'Alembertian (wave) operator. In vacuum (or far outside the source field), $T_{\mu\nu} = 0$ (approximately). One can show that the solution to $\Box\,\bar{h}_{\mu\nu} = 0$ takes the form $\bar{h}_{\mu\nu} = A_{\mu\nu}\exp(ik_\alpha x^\alpha)$. In other words, the perturbations of Minkowski space-time are plane waves tavelling at the speed of light, with propagation vectors k_α. This is the so-called gravitational waves, which is what we observe today from our ground-based detectors like LIGO and Virgo. (For more details, see Chapter 9.1 of [6] or Page 686 onward of [7] OR, do an MSc in physics and take General Relativity).

2.3 LIGO

Gravitational-wave detectors in the global network are large-scale Fabry-Pérot interferometers (see Figure 2). The two LIGO detectors in Hanford and Livingston each have arms of length 4 km, where as the Virgo and KAGRA detectors each have two arms of length 3 km. These detectors are sensitive to minuscule deformation in space-time caused by the propagation of gravitational waves. With equal arm lengths, light deconstructively interferes at the output sensor. In presence of gravitational waves, the change in arm length causes the light beams from both arms to no longer interfere perfectly at the output sensor. The interference pattern of light

from the two beams encodes information on space-time deformation.

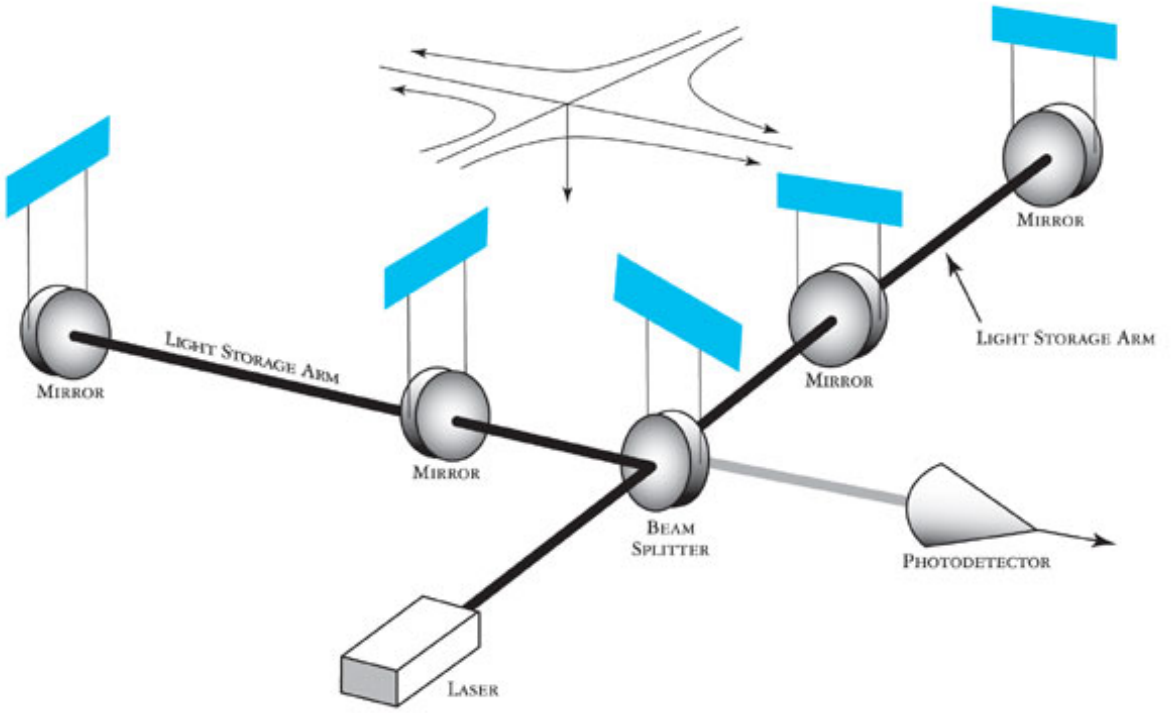

Figure 2: A simplified schematic of an interferometer [8]

The degree of space-time deformation is quantified by *strain*, a dimensionless parameter defined as the fractional change in arm lengths of each detector

$$h(t) = \frac{\Delta L(t)}{L} \tag{13}$$

where ΔL is the change in arm length and L is the original arm length. In other words, strain characterises the amplitude of gravitational-wave signals. The 1D strain[3], h is an array of strain for a range of discrete timesteps and $h(t)$ is the strain at a specific timestep, t. In the case of the first detection, GW150914, the peak strain was 1.0×10^{-21}, a remarkably small change in length [1]. This underlines the level of precision that gravitational-wave detectors have to achieve in order to make significant detections.

With strain sensitivities of up to 10^{-23}, gravitational-wave detectors are prone to various sources of noise, e.g. noises induced by thermal fluctuations and seismic motion; shot noise due to quantum fluctuations in the local light field etc. On top of that, results from the initial LIGO-Virgo science runs indicated non-stationary and non-Gaussian detector noises, which includes short-duration noise transients called 'glitches'. Glitches can be of environmental origin or due to instrumental sources, but in many cases the source of the glitch is unknown. From time to time, noise artefacts like glitches can resemble gravitational waves and consequently limit the ability of detecting low-amplitude signals. This raises a challenge for the search of astrophysical signals embedded within detector data.

So the big question in gravitational-wave data analysis is: Does the detector data contain a real, astrophysical signal? If there is, what type of waveform is it? What is the signal

[3]Usually h is a vector but we'll discuss it in 1D here for simplicity.

source/origin?

2.4 Types of GWs

GWs signals can be classified into two subgroups based on their duration: (i) transient signals (bursts) which have durations ranging from a few milliseconds up to a few seconds; (ii) long duration (or continuous) signals which can last up to hundreds of seconds or more. Since gravitational-wave signals originate from a large variety of sources, each of the two groups can be further divided into well-modelled sources, and sources that remain relatively unmodelled (see Figure 3). We describe the 4 types of GW as follows.

2.4.1 Modelled transients: Compact-binary coalescences (CBCs)

CBC is a general term used to describe a system of two inspiralling and merging compact objects like neutron stars and black holes. CBCs are the most well-understood gravitational-wave source to date and there is an extensive collection of well-developed models available to predict the waveforms emitted from such systems. For that reason, searches for CBCs can be done using a technique called matched-filtering, which (in simple terms) compares the data to a template bank i.e. a range of modelled waveforms, to find the best match to the potential GW signal in the data.

QUESTION 2: The quadrupole formula:

$$h_{ij}(t,r) = \frac{2G}{c^4 r}\ddot{I}_{ij}\left(t - \frac{r}{c}\right) \tag{14}$$

describes the rate at which GWs are emitted from a system of masses, where r is the distance to the system and $\ddot{I}_{ij}$ is the second order time derivative of the mass quadrupole moment. Assuming the bodies in the binary system have similar masses, M, we can write

$$\ddot{I}_{ij}\left(t - \frac{r}{c}\right) \sim M\Omega^2(2R^2) \tag{15}$$

where Ω is the orbital angular frequency and R is the orbital radius of the binary. Estimate the order-of-magnitude of the GW strain of a CBC, and hence how much the LIGO arms compress when the GW passes through Earth. (Hint: You can approximate the orbital radius via the Schwarzschild radius: $R_S = 2GM/c^2$.)

2.4.2 Unmodelled transients: Bursts

Although CBC signals are the only type of gravitational-wave signal with confirmed detection so far, there are many other transient astrophysical processes like core-collapse supernovae (SNe) of massive stars, pulsar glitches and cusps in cosmic strings which can be potential candidates for future burst signal detection. Unlike CBCs, the nature of these burst sources, and therefore their GW signals, are complex and non-deterministic so robust waveform templates are not available. In consequence, match-filtering cannot be used to search for these types of signals. It difficult to identify unfamiliar astrophysical signals when buried amongst noise in detector data.

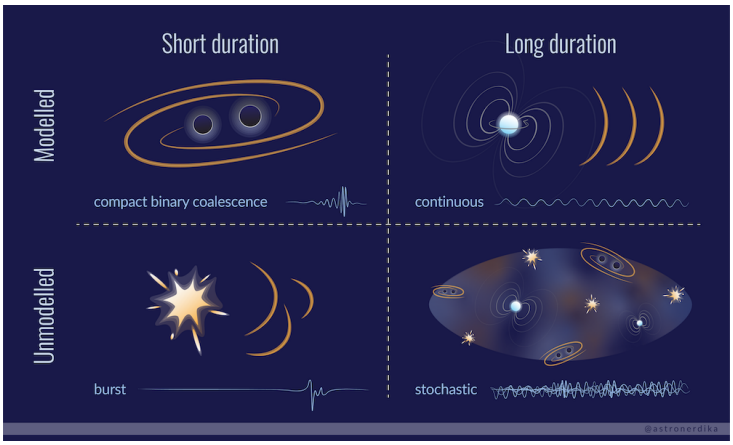

Figure 3: Types of gravitational-wave sources: transient (short duration) OR persistent (long duration), and modelled OR unmodelled. (Credit: Shanika Galaudage).

Moreover, unmodelled signals from unmodelled sources could be more easily misidentified as glitches[4]. Despite the challenges, it is worthwhile to search for generic gravitational-wave burst signals as they can provide remarkable insights to the physics of the originating source: from the dynamics of its surrounding environments to the centre of the object itself. This can improve our understanding of obscure astrophysical sources and processes, or even make prospective discoveries of sources previously unknown to humankind.

The alternative to matched-filter searches are unmodelled searches. The distinctive feature between the two methods is that unmodelled searches reconstructs waveform from detector data from scratch instead of fitting detector data using pre-determined templates, with no prior knowledge of the source. (In Section 6.1 of this project, we will introduce an unmodelled search algorithm called *BayesWave*.)

2.4.3 Modelled long-duration: Continuous gravitational waves (CWs)

Continuous waves refer to persistent, quasi-monochromatic gravitational waves produced by neutron stars, including isolated neutron stars and neutron stars in binaries. Some models for dark matter also produce a continuous gravitational-wave signal, motivating searches for continuous waves from ultralight bosons surrounding black holes [9]. No continuous gravitational waves have been detected to date.

2.4.4 Unmodelled long-duration: Stochastic gravitational waves

Stochastic gravitatioanl waves also referred to as the stochastic gravitational-wave background (SGWB), are unmodelled, persistent gravitational-wave signals of astrophysical and cosmological origins. The astrophysical component is produced by the superposition of other gravitational-wave signals, including CBCs, supernovae, and continuous wave signals. The cosmological component consists of gravitational waves produced during and immediately after inflation and from early-universe phase transitions. Searches for the SGWB are conducted using both LIGO data [10] and pulsar timing arrays [11].

[4]Glitches are non-astrophysical, non-Gaussian, transient power spikes in the detector data, which can mimic or mask short-duration GW signals (bursts).

3 Project Overview

3.1 Bayesian inference and MCMC

In research areas which heavily involve data (e.g. astrophysics, particle physics), Bayesian inference is commonly used for parameter estimation and statistical modelling. In the first half of this project, we explore Bayesian inference and its applications in generic physics problems both analytically and numerically. We introduce the Metropolis-Hasting algorithm, a variant of the Monte Carlo Markov Chain (MCMC) methods, which is a widely applicable numerical method used to sample probability distributions for the parameters of interest.

3.2 OPTION 1: Signal processing - Waveform whitening

GW detectors are extremely susceptible to noise given their high sensitivities to minuscule changes in space-time. In the second part of the lab, you will learn about noise whitening, that is to normalise the amplitudes of a non-uniform frequency spectrum of noise (i.e coloured noise). This is a procedure in signal processing, applicable beyond GW searches - it is applicable to any device/sensor/instrument etc. with non-uniform response in the frequency domain.

3.3 OPTION 2: Data analysis - Digging the data for gravitational waves

You will work with data from a Bayesian GW burst detection algorithm called *BayesWave*. The algorithm uses MCMC to sample parameters for GW waveform reconstructions. The main goal of this exercise is to give you the experience of managing large data sets (data analysis!). But more excitingly, you will be recovering GW waveforms from the data yourself i.e. finding a gravitational wave!

4 Bayesian Inference

One of the common routines in scientific research is to statistically infer the models and model parameters that best describe observations. The models along with their parameters are then used to make further testifiable predictions for a given scientific phenomenon. There exist various types of statistical inferences. In this lab we introduce Bayesian inference along with frequentist inference for comparison.

The frequentist interpretation is probably the type of statistics that you are familiar with (from what you've learned in high school and undergraduate maths). For frequentists, a parameter of interest (e.g. a model parameter) has a true, fixed value and any discrepancies in the observed value is due to experimental/random errors. Frequentists design estimators (which can be calculated from observations) for such a parameter so that it converges to the true parameter value as the number of observations increases. An example of such an estimator is the average.

In the Bayesian interpretation, on the other hand, our understanding of the parameter is limited, hence its value is random and typically follow a distribution. That is, Bayesian interpretation treats a parameter as a random variable sampled from an underlying distribution, as opposed to having fixed value. To summarise, frequentist probabilities fundamentally depend

on the frequencies of events whereas Bayesian probabilities are based on our knowledge of the events.

4.1 Bayes' Theorem

Now let's discuss Bayesian inference mathematically via Bayes' theorem, which describes the probability of a proposition (e.g., event, parameter, model), based on prior knowledge of conditions that might be related to the proposition. The probability of the proposition obtained under such conditions/constraints is known as the posterior probability. Essentially, Bayes' theorem uses one proposition to evaluate the probability of the other. Propositions are statements asserting that something is true. Examples of propositions include: "It is raining today", "James has COVID", and "Modified Newtonian Dynamics (MOND) is correct". Given two arbitrary propositions A and B, Bayes' theorem states

$$p(A \mid B) = \frac{p(A) \cdot p(B \mid A)}{p(B)}, \tag{16}$$

where:

- $p(A \mid B)$ is the probability of A occurring given that B is true (a.k.a. **posterior probability**);
- $p(B \mid A)$ is the probability of B occurring given that A is true (a.k.a. **likelihood**);
- $p(A)$ and $p(B)$ are the probabilities of observing A and B respectively without any conditions.

A video by 3Blue1Brown discusses the Bayes' Theorem in a very neat way, with incredible illustrations: https://www.youtube.com/watch?v=HZGCoVF3YvM - we highly recommend you to watch this in your own time!

> QUESTION 3: Currently living in the Coronavirus (COVID) pandemic, James, an astrophysics PhD student, still attends conferences in person for science. He has recently returned to Melbourne from a Gravitational Wave conference in Canberra. He was feeling fine and hasn't heard about any positive COVID cases from the conference, but he did a rapid antigen test (RAT) just in case. The RAT he used has a 98% true positive rate, a 15% false positive rate, while the COVID prevalence rate is 0.5% in Australia. Use Bayes' theorem to calculate the probability that James has COVID if he tested RAT positive. (These numbers are not real.)
>
> Note: Think carefully about what your propositions are before proceeding.

In astrophysics, Bayes' Theorem can be used for model parameter estimation. Let's look at a specific example - say we are trying to study the parameter θ_α of a one-dimensional model (i.e.hypothesis) $\mathcal{M}_\alpha$, based observed data D. In this scenario, our propositions are θ_α and $\mathcal{M}_\alpha$. So the question we are trying to answer is: given the data D and that $\mathcal{M}_\alpha$ is true, how would the model parameter θ_α be distributed. In other words, we are trying to evaluate the (posterior)

distribution of θ_α based on our observations (D) and a given hypothesis. Using Bayes' Theorem, we write

$$p(\theta_\alpha|D, \mathcal{M}_\alpha) = \frac{p(\theta_\alpha \mid \mathcal{M}_\alpha) \cdot p(D \mid \theta_\alpha, \mathcal{M}_\alpha)}{p(D \mid \mathcal{M}_\alpha)}. \tag{17}$$

Definitions of each term in this context:

- $p(D \mid \theta_\alpha, \mathcal{M}_\alpha)$ is the **likelihood function** i.e. the probability of observing D if our $\mathcal{M}\alpha$ is parameterised by some given θ_α;
- $p(\theta_\alpha \mid \mathcal{M}_\alpha)$ is the **prior probability** of θ_α which specifies our initial beliefs of θ_α prior to observing the data D i.e. how did we think θ_α is distributed before we were given the data D. Priors are formed in light of (or from the lack of) the data from past experiments

$$p(\theta_\alpha \mid \mathcal{M}_\alpha) = \int p(\theta_\alpha|D_{\text{past}}, \mathcal{M}_\alpha)\, p(D_{\text{past}}|\mathcal{M}_\alpha)\, dD_{\text{past}}. \tag{18}$$

- $p(D \mid \mathcal{M}_\alpha)$ is the **marginalised likelihood**. As its name suggest, we are marginalising (in a way summing) the likelihood function over the parameter space of the model $\mathcal{M}_\alpha$. For this reason, $p(D \mid \mathcal{M}_\alpha)$ is otherwise known as the "model evidence". In this example, our parameter space is all possible values of parameter θ_α, so we write:

$$p(D \mid \mathcal{M}_\alpha) = \int p(D \mid \theta_\alpha, \mathcal{M}_\alpha)\, p(\theta_\alpha \mid \mathcal{M}_\alpha)\, d\theta_\alpha \tag{19}$$

Everything discussed above is also applicable to a multi-dimensional model (models with more than one parameter), except we change the notation: $\theta_\alpha \to \vec{\theta}_\alpha$. The vector notation, $\vec{\theta}_\alpha$ in this context refers to a set of variables that parameterises the multi-dimensional model. These variables don't have to form a single vector-like physical quantity and they can have different units. For multi-dimensional models, the posterior distribution is now a **joint (posterior) distribution** for all parameters, given by exactly the same equation as Equation 17 except $\theta_\alpha \to \vec{\theta}_\alpha$). However we note that the model evidence is no longer just a single variable integral as we saw in Equation 19. It now involves a multi-variate integral (note the $d\vec{\theta}_\alpha$)

$$p(D \mid \mathcal{M}_\alpha) = \int p(D \mid \vec{\theta}_\alpha, \mathcal{M}_\alpha)\, p(\vec{\theta}_\alpha \mid \mathcal{M}_\alpha)\, d\vec{\theta}_\alpha \tag{20}$$

where we marginalise over the parameter spaces of *all* parameters associated with model $\mathcal{M}_\alpha$.

Another great thing about Bayes' Theorem is its flexibility i.e. it can be used for pretty much *anything*, not just to predict the posterior of a single model parameter or the joint posterior of a set of model parameters. For example, the parameter θ can be replaced with a model, $\mathcal{M}_\alpha$ such that

$$p(\mathcal{M}_\alpha \mid D) = \frac{p(\mathcal{M}_\alpha) \cdot p(D \mid \mathcal{M}_\alpha)}{p(D)}. \tag{21}$$

Notice that the likelihood of $\mathcal{M}_\alpha$ i.e. $p(D \mid \mathcal{M}_\alpha)$ is marginalised over the prior probabilities of its parameters as given by Equation 19. By definition, the evidence $p(D)$ is the marginalised likelihood i.e. the likelihood function integrated over all of the parameter space. In this case, our parameter space includes all possible models $\mathcal{M}_\alpha$ that could represent the data D. However the space of all possible models is generally unknown. Therefore $p(D)$ is a typically expressed

as discrete sum taken over the subset of models that are being considered:

$$p(D) = \sum_{\alpha=1} p(D \mid \mathcal{M}_\alpha)\, p(\mathcal{M}_\alpha), \tag{22}$$

as opposed to an integral i.e. a continuous sum over all possible **model parameters** as we saw in 17.

4.2 Bayesian Model Selection: Bayes factor

Given two competing models (i.e. hypotheses), $\mathcal{M}_\alpha$ and $\mathcal{M}_\beta$, we can determine which of the two is more strongly supported by the data by considering the posterior odds ratio

$$\mathcal{O}_{\alpha,\beta} = \frac{p(\mathcal{M}_\alpha \mid D)}{p(\mathcal{M}_\beta \mid D)}. \tag{23}$$

By Equation 21, the above can be rewritten as

$$\mathcal{O}_{\alpha,\beta} = \frac{p(\mathcal{M}_\alpha)}{p(\mathcal{M}_\beta)} \frac{p(D \mid \mathcal{M}_\alpha)}{p(D \mid \mathcal{M}_\beta)}. \tag{24}$$

We note that the evidence, $p(D)$, from Equation 21 conveniently cancels. The first term on the RHS is the prior odds ratio. Often we have no information regarding which prior ($p(\mathcal{M}_\alpha)$ or $p(\mathcal{M}_\beta)$) takes precedence, hence the prior odds ratio is taken to unity. The second term is what we call the **Bayes factor**,

$$\mathcal{B}_{\alpha,\beta} = \frac{p(D \mid \mathcal{M}_\alpha)}{p(D \mid \mathcal{M}_\beta)}. \tag{25}$$

Since the prior odds ratio is typically equal to 1, the Bayes factor is effectively the ratio of posterior probabilities of the two models i.e. $O_{\alpha,\beta} = \mathcal{B}_{\alpha,\beta}$.. In which case $\mathcal{B}_{\alpha,\beta} > 1$ suggests that $\mathcal{M}_\alpha$ is more strongly supported by the data (i.e. more favourable) than $\mathcal{M}_\beta$. Intuitively, using the Bayes factor, one can iteratively compare models from the space of all models until the model that best describe observations are found. In other words, we are trying *maximise the posterior odds ratio* to select the best model to represent the data.

You may have also noticed that the Bayes factor between two competing **models** is effectively the ratio of marginalised likelihood i.e. **model** evidence as defined in Equation 19 (or Equation 20 in the case of multi-dimensional models). However, if we are trying to compare simpler hypotheses e.g. two specific **parameter values** θ_1 and θ_2 of the same model $\mathcal{M}_\alpha$ (as in Equation 17), then the Bayes factor is simply the likelihood ratio (instead of *marginalised* likelihood) i.e.

$$\mathcal{B}_{\theta_1,\theta_2} = \frac{p(D \mid \theta_1,\ \mathcal{M}_\alpha)}{p(D \mid \theta_2,\ \mathcal{M}_\alpha)}, \tag{26}$$

again assuming uniform prior $p(\theta_1 \mid \mathcal{M}_\alpha) = p(\theta_2 \mid \mathcal{M}_\alpha)$ i.e. no prior knowledge of θ. This is a special case where maximising the likelihood ratio is equivalent to maximising the posterior odds ratio.

QUESTION 4: For a large number of identical, non-interacting, and non-relativistic classical particles in thermal equilibrium (ideal gas), the probability density function (PDF), $f(v)$, is given by the Maxwell-Boltzmann distribution

$$f(v) = \left(\frac{m}{2\pi kT}\right)^{3/2} 4\pi v^2 e^{-\frac{mv^2}{2kT}}, \tag{27}$$

where $\int_0^\infty f(v)\, dv = 1$, v is the particle speed, m is the particle mass, k is the Boltzmann constant, and T is the temperature. $f(v)$ is our model and we wish to estimate its parameter, the most probable particle speed, v_p, of our particles.

(a) Show that $v_p = \sqrt{\frac{2kT}{m}}$ and

$$f(v) = \frac{4}{\sqrt{\pi}} \frac{v^2}{v_p^3} e^{-\frac{v^2}{v_p^2}}. \tag{28}$$

(b) The speed of a single particle chosen arbitrarily from the distribution is measured to be 23 ms^{-1}. Given two hypotheses for v_p: (i) $v_{p1} = 10$ ms^{-1} and (ii) $v_{p2} = 50$ ms^{-1}, which hypothesis is favoured given this measurement? Assume a uniform prior i.e. our prior beliefs say that all hypotheses are equally likely.

(c) The speed of nine other randomly chosen particles are measured to be 24, 26, 12, 60, 3, 15, 30, 27, 35 ms^{-1}. Given the same two hypotheses as above (with the same prior assumptions), use the Bayes' factor to find out which hypothesis is favoured given these additional measurements i.e. you now have 10 samples: 23 ms^{-1} from part (b) plus the 9 samples above.

Hint: $p(D_1, D_2, ..., D_N) = p(D_1)p(D_2)...p(D_N)$ if all D's are independent of each other.

Note: You should also consider writing a script (in Jupyter notebook or whatever you feel comfortable with) to do your calculations. It is much easier than punching a bunch of numbers into a calculator which is hard to keep track of. The script for this question will not be graded, but please include it in your submissions so the marker can refer back to your working if need be.

(d) Given an arbitrary number of measure velocity samples, N, analytically estimate the parameter v_p that best fits these measurements.

Hint: You should find that

$$v_p = \sqrt{\frac{2\sum_{i=1}^{N} v_i^2}{3N}}, \tag{29}$$

where N is the total number of data points.

In general, determining the best-fitting models and model parameters analytically as in part (d) is difficult or even impossible. However, we can solve these problems using numerical techniques, such as using the Markov chain Monte Carlo (MCMC) algorithm, as we shall explore in the next section.

5 Markov chain Monte Carlo (MCMC)

Now with the knowledge of Bayesian statistics in mind, we will introduce the popular algorithm - Markov chain Monte Carlo (MCMC). MCMC is typically used to obtain information about distributions e.g to estimate posterior distributions in Bayesian inference, where usually cannot be directly calculated or derive analytically.

The name of MCMC suggests the algorithm combines two properties: Markov chain and Monte Carlo. Monte Carlo estimates the properties of a distribution by examining random samples from the distribution. A classic example is estimating the value of π by generating a large number of random points and see how many fall in the circle enclosed by the unit square. The benefit of Monte Carlo is that random samples are easy to draw and calculate the statistics with, when the distribution's equations are hard to work with in other ways, such as direct calculations. The Markov chain aspect is that the random samples are generated in sequence, such that the next sample depends *only* on the one before it.

5.1 Metropolis-Hastings algorithm

The Metropolis-Hastings algorithm is a MCMC method for obtaining a random sequence of samples from a probability distribution where direct sampling is difficult. Before discussing how this algorithm is used in Bayesian inference, let's discuss its in broader terms. Say we want to draw samples of some variable x from some probability distribution with probability density $P(x)$. However, we may not know exactly what $P(x)$ is, but we know a function $f(x) \propto P(x)$ such that the values of $f(x)$ can be calculated. We can therefore sample from $f(x)$ (instead of directly from $P(x)$) to achieve a distribution of samples which closely approximates the desired distribution as if it was sampled from $P(x)$ (the desired distribution).

In order to obtain a sequence of x samples to achieve the desired distribution, the Metropolis-Hastings algorithm consists of two steps: the proposal step and the acceptance-rejection step. In the proposal step, a random sample $x_{proposed}$ is generated from a proposal distribution $q(x_{proposed}|x_{current})$, centered about the current sample value, $x_{current}$. This is known as a Markov process, where the next state in a sequence of possible events ($x_{proposed}$) depends only on the state of the previous event ($x_{current}$).

The acceptance-rejection step then decides, with a given probability, if $x_{proposed}$ is accepted and added into the MCMC chain. The probabilistic decision is a Monte Carlo method and hence the name MCMC. The probability of acceptance is determined by comparing the values of $f(x)$ of $x_{proposed}$ and $x_{current}$ via the Metropolis ratio, r

$$r = \frac{f(x_{proposed})}{f(x_{current})} \cdot \frac{q(x_{current}|x_{proposed})}{q(x_{proposed}|x_{current})}. \tag{30}$$

Although proposal distributions can essentially take any form, we will only deal with symmetric distributions in this lab i.e. $q(x|y) = q(y|x)$ and so the proposal ratio (i.e. the second factor in Equation 30) equals unity.

Now let's discuss how r decides if our proposed sample $x_{proposed}$ gets accepted into the MCMC chain. If $r \geq 1$, we set $x_{current} = x_{proposed}$ because $f(x_{proposed})$ takes precedence over $f(x_{current})$. If $r < 1$, $x_{proposed}$ has a lower likelihood of occurrence compared to $x_{current}$ according to $f(x)$.

Therefore the acceptance of $x_{proposed}$ into the chain is not guaranteed, rather we accept it with a probability r. If $x_{proposed}$ is not accepted, then $x_{current}$ remains as the previous accepted value. The overall acceptance probability can be expressed mathematically as

$$\alpha = \min(1, r). \tag{31}$$

In Bayesian statistics, the goal is typically to obtain the posterior distribution of a variable/parameter based on the information we have from the prior, the data and the hypothesis. However, when the models involve multiple parameters as in Equation 19, the Bayesian evidence involves multi-variate integrals and therefore analytically determining the posterior distribution may be impossible, if not extremely challenging. Therefore, in such cases, we turn to the Metropolis-Hasting algorithm to numerically estimate the posterior distribution. In this case, the posterior distribution is our desired distribution, analogous to $P(x)$ above. So the requirement is to find a function proportional to the posterior distribution (analogous to $f(x)$) to make the Metropolis-Hasting algorithm work. This is made easy with Bayes' Theorem, where the posterior is proportional to product of the prior and the likelihood function. So let's say we are trying to estimate the posterior of the model parameter θ, given the model $\mathcal{M}$ and the data D i.e. $p(\theta|D, \mathcal{M})$. The Metropolis-Hastings ratio following Equation 30 is then given by

$$r = \frac{p(D|\theta_{proposed}, \mathcal{M})p(\theta_{proposed} \mid \mathcal{M})}{p(D|\theta_{current}, \mathcal{M})p(\theta_{current} \mid \mathcal{M})}, \tag{32}$$

which is essentially the ratio of the Bayes' Theorem numerator for $\theta_{proposed}$ and $\theta_{current}$ (see Equation 17). Note that the denominator (the model evidence) of Bayes' Theorem conveniently gets ignored, or technically cancelled out, as it simply is a constant of proportionality. Also note that we have taken the proposal ratio in Equation 32 unity under the assumption that we are using symmetric proposal distributions.
The Metropolis-Hastings algorithm can be summarized as follows:

1. Start with a random initial sample $X_{current}$.
2. Randomly draw a sample $X_{proposed}$ from the proposal distribution.
3. Find a function, $f(x)$ that is proportional to your desired distribution.
4. Calculate r and hence α.
5. Randomly draw a sample U_1 from a uniform distribution between 0 to 1 i.e. $U_1 \sim U(0, 1)$.
6. If $U_1 \leq \alpha$, then $X_{proposed}$ is accepted and added into MCMC chain. Set $X_{current} = X_{proposed}$;
   Otherwise add $X_{current}$ into MCMC chain and discard $X_{proposed}$.
7. Repeat steps 2-5 for a large number of samples, so that the MCMC samples produces the target distribution.

5.2 Writing your own MCMC

We will now attempt to implement an MCMC for a given data sample "`mcmc/mcmc_data.csv`". In the following instructions, we discuss the general logistics and things to consider for writing the algorithm - Refer to "`mcmc/mcmc.py`" for skeleton code:

1. Plot the distribution of the dataset "`mcmc/mcmc_data.csv`". The data contains 1000 samples. Can you see a pattern in the data? What does the distribution look like i.e. what do you think is the underlying distribution from which the samples are drawn?

2. **TASK 1: Toy Example**
   Write a Metropolis-Hasting algorithm to obtain the posterior distribution of the data sample provided. This exercise might sound very trivial i.e. why are we reconstructing our dataset...when we ALREADY have the dataset? As discussed previously, MCMC is typically used to obtain a sequence of samples from a probability distribution where direct sampling is difficult (e.g. when the problem involve large number of parameters). But the purpose of this exercise is to help you understand how the MCMC algorithm works fundamentally before we use it for anything more sophisticated. So hopefully we have convinced you to proceed with this seemingly trivial exercise.

   The dataset provided consists of 1000 samples of an unknown variable, X. With the MCMC algorithm, we can reconstruct the distribution of X with as many samples as we want using the existing 1000 samples. The distribution of X obtained from the MCMC will be a more refined version of the 1000 samples (but they are essentially the 'same' distribution).

   Here's a list of what you should consider:

   - What are the arguments of your MCMC function? (They are given to you in the skeleton code, but make sure you know what they all mean!)
   - What is the output of your MCMC algorithm?
   - At each iteration of the MCMC, you propose a new sample $X_{proposed}$, which you draw from a proposal distribution. Your proposal distribution should be centred at your current sample, $X_{current}$. To help with your intuition, we use a Gaussian distribution as our proposal distribution for all the MCMC exercises, such that it is less probable to draw a proposed sample further away from the current sample. **Extension** (do this only if you have time): The width of your proposal distribution (i.e. how far can your proposed sample be from your current value) also affects your MCMC. In the skeleton code, this can be adjusted using the `jump_size` argument. Have a look at the initial dataset which you are trying to recreate, start testing with different `jump_size` to see what it does.
     Hint: for a given width, calculate the "acceptance ratio" which is the ratio of samples accepted into the MCMC chain to the total number of iterations in the chain.
   - To obtain the Metropolis ratio, we need to find $f(X)$ that is proportional to our desired distribution, $P(X)$. Well...in this toy example, we are trying to reconstruct

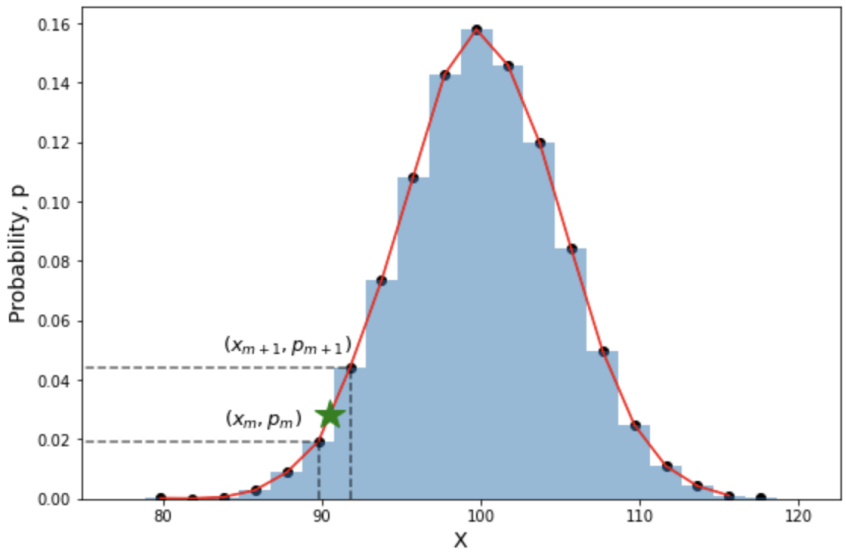

Figure 4: Visualising interpolation.

the distribution of data using the probability density function generated using the data samples itself. So technically $f(X) = P(X)$.

- However $f(X)$ must be a continuous function, such that values of $f(X)$ must exist for any value of $X_{proposed}$. In order to construct a continuous $f(X)$, we first obtain the discrete probability distribution of the data (blue histogram in Figure 4) using the `get_distribution()` function such that $p_1, p_2, \cdots, p_n$ are probabilities associated with $X_1, X_2, \cdots X_n$ (black points in Figure 4). We can then obtain the likelihood of any given X (e.g. green star in Figure 4) by interpolating the discrete probability distribution such that:
  If $X_m \leq X \leq X_{m+1}$, we linearly interpolate (i.e. draw a straight line - red lines in Figure 4) between (X_m, X_{m+1}) and (p_m, p_{m+1}) to obtain $p_m \leq p(X) \leq p_{m+1}$.
  This is mostly written for you in the `interpolate_distribution()` function, there are just a couple of blanks to fill in (see the function description in the code for more information).
- The remaining steps are simply to decide whether or not $X_{proposed}$ gets accepted into the MCMC chain. The conditions for acceptance follow the Metropolis-Hastings algorithm rules described above in Section 5.1.
- Plot your MCMC chain using the function `plot_mcmc_output`. The argument called "`burn_in`" is a casual term used to refer to the group samples that are discarded at the start of the MCMC chain[5].
- Compare your posterior distribution with the data distribution. Do they look alike? Do the distributions look like anything you've seen?

[5]Since the initially proposed sample, `x0` can be pretty much anything, the MCMC chain may start its iterations in low probability regions in the parameter space. It is typically expected that MCMC chains will eventually find its way into the high probability region, but whether or not it happens is all up to probability. But generally speaking, the more iterations we have, the closer we get to the true posterior. So the idea of burn-in is to discard an arbitrary number of samples at start the MCMC chain as it is still stabilising and finding the 'right footing' in the parameter space.

3. **TASK 2:**

You might have noticed that the distribution of the data you saw in the last exercise looks suspiciously like the Maxwell-Boltzmann distribution discussed in Question 4. If we trust that the Maxwell-Boltzmann distribution describes our data, we can then choose use the Maxwell-Boltzmann equation as the model, $\mathcal{M}$ for our data[6]. As you saw in Question 4(a), the Maxwell-Boltzmann distribution is characterised by a single parameter, v_p. We can then infer the posterior distribution of v_p i.e. for the given model $\mathcal{M}$ (Maxwell-Boltzmann), what is the probability that the a given value of model parameter v_p generates the underlying (Maxwell-Boltzmann) distribution from which our data is drawn from. Using Bayes' Theorem the posterior distribution of v_p is given by:

$$p(v_p|D,\mathcal{M}) = \frac{p(D|v_p,\mathcal{M})p(v_p|\mathcal{M})}{p(D|\mathcal{M})} \tag{33}$$

where $p(D|\mathcal{M}) = \int p(D|v_p,\mathcal{M})p(v_p|\mathcal{M})\,dv_p$ is the model evidence. The Metropolis-Hastings algorithm implementation is identical to the one you write in Task 1. However, instead of generating $f(X)$ from the data, you $f(X)$ is now the denominator of Equation 33 for reasons discussed in Section 5.1. Here are a few tips to help you get started:

- The likelihood, $p(D|v_p,\mathcal{M})$, is the probability of obtaining our data D with the given model $\mathcal{M}$ parameterised by v_p (think carefully about what this means, chat with your demonstrators, do not blindly proceed without understanding what the likelihood is).
- The prior, $p(v_p|\mathcal{M})$ is our initial belief of how v_p is distributed, before D is observed. In the skeleton code, `prior=None` means that we have a uniform prior i.e. means all values of X are equally probable. The reason why we can exclude the prior completely from the algorithm when we have a uniform prior is because $p(v_{p,current}|\mathcal{M}) = P(p(v_{p,proposed}|\mathcal{M})$ and they cancel out in the Metropolis ratio, r.
  If a prior is given, it should be a scipy object (e.g. `sp.stats.norm`, `sp.stats.maxwell`) which can inherit the method `.pdf()`.
- Let's start with the `prior=None` setting (i.e. assuming uniform prior), which simplifies our Metropolis ratio to the ratio of likelihoods between $v_{p,proposed}$ and $v_{p,current}$
- To obtain the likelihood, we first obtain the PDF for the model for the v_p of interest i.e. $\mathcal{M}(v_p)$. The `maxwell_boltzmann()` function in the skeleton code does this for you (i.e. you don't have to write this yourself). We can then use the `.pdf()` method to find the probability of each sample in our dataset, D. This will return an array of probabilities, and $p(D|v_p,\mathcal{M})$ i.e. the overall likelihood will be the product of all the probabilities in the array.
- You may run into an issue when trying to take the product of all the probabilities in the array. What is happening? How do you fix it? Chat with your demonstrator!
- As before, plot your results using the function `plot_mcmc_output`. What does your

[6]The question of whether or not one trusts a particular model can be asked and answered rigously. It, however, is outside the scope of this lab

posterior distribution look like? Based on your MCMC, what is the most probable v_p of the underlying Maxwell-Boltzmann distribution?

6 GW Data analysis

In this short project, we cannot cover all aspects of GW data analysis. So we introduce two key ideas: (i) spectral whitening and (ii) waveform recovery. We will explain what each of this means as we go. The main aim of these exercises is to give you a taste of signal processing and data analysis in the context of GW astronomy. However, the concepts and skills that you learn from this lab are widely applicable in disciplines beyond GW (and astrophysics). Hopefully you can get something useful out of this project regardless if you decide to continue in GW research.

6.1 *BayesWave* and your analysis dataset

Before we proceed with the exercises, let's first introduce the algorithm which produced the datasets that you will be using for the remaining of this lab. In Section 2.4, we introduced various types of GW sources. Recall that there are two types of transient sources, CBCs and unmodelled bursts. As discussed in 2.4.1, the match-filtering technique is typically used to search for CBCs as they are well-understood and hence well-modelled. However, unmodelled bursts cannot be searched for the same way as CBCs due to their stochastic and complex nature. Therefore we now introduce an unmodelled burst search algorithm used to generate your analysis datasets - *BayesWave*.

The *BayesWave* algorithm is designed for the joint detection of generic, unmodelled GW bursts and non-astrophysical transients (instrumental glitches). *BayesWave* uses a sum of sine-Gaussian wavelets to reconstruct any non-Gaussian, transient features in the detector data, with no prior assumption on its origin or morphology. As its name suggests, sine-Gaussian wavelets are essentially sine wave signals modulated by a Gaussian envelope. A single sine-Gaussian wavelet has five parameters: t_0, f_0, Q, A, ϕ_0 which denote central time, central frequency, quality factor[7], amplitude and phase offset respectively. The mathematical representation of a sine-Gaussian wavelet is given by

$$\Psi(t : t_0, f_0, Q, A, \phi_0) = Ae^{-\Delta t^2/\tau^2} \cos(2\pi f_0 \Delta t + \phi_0) \tag{34}$$

where $\tau \equiv Q/(2\pi f_0)$ and $\Delta t \equiv t - t_0$. Figure 5 shows an example of a single sine-Gaussian wavelet and Figure 6 shows how a waveform is reconstructed by summing over multiple sine-Gaussian wavelets.

Since *BayesWave* has no prior knowledge of the transient feature's origin, it reconstructs all non-Gaussian transient features in the detector data as three seperate models: (i) Gaussian noise-only model, (ii) Gaussian noise + GW signal model, $\mathcal{S}$ and (iii) Gaussian noise + instrumental glitch model, $\mathcal{G}$. The algorithm then compares the Bayes factor between the models to select the one that best represents the data. (Note: from here on, we will refer to $\mathcal{S}$ and $\mathcal{G}$ as the 'signal' and 'glitch' models respectively)

[7]Quality factor is the ratio of f_0 to the frequency bandwidth of the wavelet. Essentially higher quality factor means the wavelet is more spread out in time. See this link for an interactive demonstration.

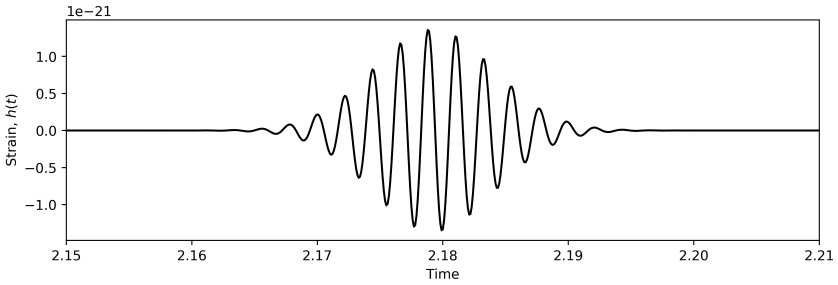

Figure 5: A single sine-Gaussian wavelet with $t_0 = 2.18$s, $f_0 = 451.06$Hz, $A = 1.37 \times 10^{-21}$, $Q = 19.34$ and $\phi_0 = 1.19$.

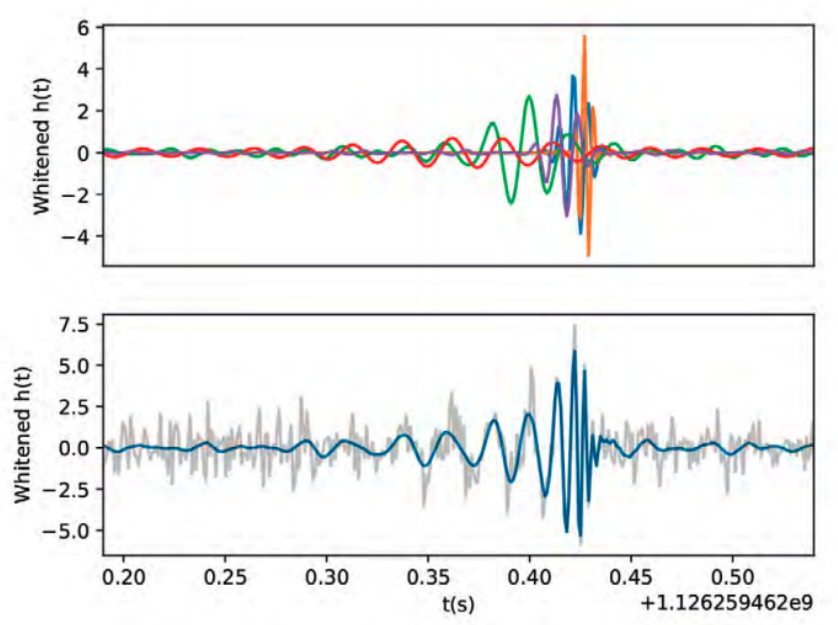

Figure 6: The top panel shows individual sine-Gaussian wavelets, each of different colour, used in the reconstruction of the first detection event GW150914 in the time-domain. The bottom panel shows the resulting waveform as a sum of all wavelets, overlaid on the actual data shown in grey. [Image adapted from: LIGO Magazine, Issue 16, March 2020]

BayesWave is specifically designed to search for unmodelled transients and reconstruct the signal buried in the data. However, because of the flexibility of the frames (i.e. sine-Gaussian wavelet), *BayesWave* can also be used to search for and reconstruct modelled transients like CBCs. The difference between *BayesWave* and match-filter searches is that *BayesWave* constructs the signal from scratch instead of matching the data with ready-made templates, and the parameters of the sine-Gaussian wavelets are determined via MCMC (hence why we introduced the concept in the first part of the lab).

Even though we have been hyping up unmodelled bursts, the data in this project is obtained by injecting simulated CBC signals into detector noise, and then recovering them using *BayesWave*. The CBC signals are simulated using similar parameters to the first detection, GW150914. You will not be learning how to use *BayesWave* in the lab due to time constrains, but you will be analysing the output of the algorithm.

OBTAINING BAYESWAVE DATA:

1. **Copy the zipped data to your working directory via Terminal.**
   - Path to zipped data:
     `/home/student.unimelb.edu.au/PhysicsCommon/3rd_year/GW_CBC_lab/BayesWave_data.tar.gz`
   - Path to your working directory depends on where you saved the skeleton code zip file (`gwCBC.tar.xz`) you downloaded from Canvas. You want to copy the zipped data above into `gwCBC` and **NOT** into `gwCBC/python/` (which is the directory containing the code that you will be submitting and you don't want to be submitting the $\sim$ 5GB worth of *BayesWave* data).
   - You can copy using the terminal command
     `cp <path-to-zipped-data> <path-to-gwCBC>`
2. Command line for unzipping a `.tar.gz` file in Terminal:
   `tar -xvf BayesWave_data.tar.gz`

PLEASE READ: As mentioned earlier, there are two exercises in the following sections: (i) spectral whitening and (ii) waveform recovery. We suggest you pick one to start with and if you have time, you can do the other one as an extension. Please read through the notes of both exercises so you have a brief idea of what they involve, discuss with your lab partners and pick the same exercise so you can work on it together. Let your demonstrators know which one you have decided on.

6.2 EXERCISE 1 - Signal processing: Waveform whitening

Like all detectors, GW detectors are susceptible to noise. However, noise does not occur at a single frequency, rather they are present throughout the frequency spectrum with various amplitudes. We call this coloured noise (as opposed to white noise). This means that whatever signal we find, it is "coloured" by detector noise.

We note that noise spectrum is different in each individual detector, so the detectors will "see" the same signal with different amplitudes. In order to marginalise the effects of the varying noise spectra across detectors, we "whiten" the signal by removing the coloured noise contributions. As the name suggests, the whitened signal is the signal as if it would be seen in white noise (i.e. noise with equal intensities at all frequencies).

Colored noise is typically described using a power spectral density (PSD), $S_n(f)$, which shows the distribution of signal power across a range of frequency, f. For a dimensionless noise strain amplitude $\tilde{h}_n(f)$ in the frequency domain[8], the PSD is given by

$$S_n(f) = \frac{h_n(f)^2}{f}. \tag{35}$$

[8]Note on notation: for the remaining of this text, 'tilde' (the $\sim$ symbol) implies that the function is in the frequency domain.

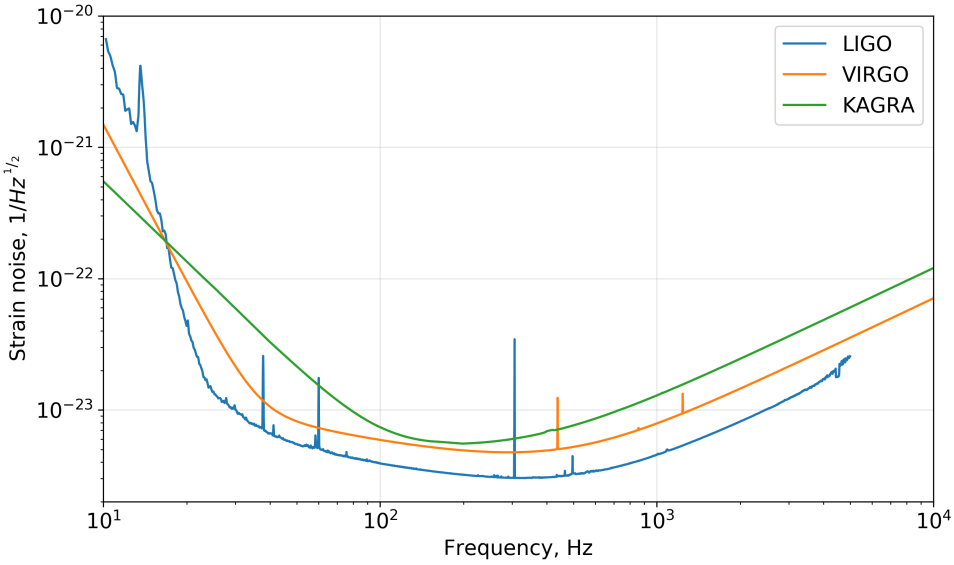

Figure 7: Simulated LIGO, Virgo and KAGRA strain amplitude spectral densities. (Data from: https://dcc.ligo.org/LIGO-T2000012/public).

Since $S_n(f)$ scales as noise strain amplitude squared, it is not directly comparable to the data measurements. Therefore by convention, the amplitude spectral density (ASD), which is simply $\sqrt{S_n(f)}$, is a more commonly used to quantity for measuring detector noise strain. To help visualise this and show how ASDs are typically plotted in literature, Figure 7 shows the simulated ASDs of LIGO, Virgo and KAGRA.

Given a coloured, time domain waveform $h(t)$ the procedure of whitening is as follows [12]:

$$h(t) \xrightarrow{\text{FFT}} \tilde{h}(f) \xrightarrow{\text{whiten}} \tilde{h}_w(f) = \frac{\tilde{h}(f)}{\sqrt{S_n(f)}} \xrightarrow{\text{iFFT}} h_w(t). \tag{36}$$

The whitening procedure involves dividing the Fourier coefficients by the estimated noise amplitude spectral density. By down-weighting frequency bins where the noise is loud, the data in each frequency bin is ensured to have equal significance. The whitened samples in the time domain have unit of variance.

Now that you have a gist of waveform whitening, you will be writing a script to whiten two types of waveforms: (i) a sine-Gaussian wavelet coloured by arbitrarily generated noise and (ii) a CBC signal coloured by simulated detector noise (*BayesWave* output). The notebook `GW_whitening.ipynb` has all the instructions and skeleton code to help you get started.

6.3 EXERCISE 2 - Searching for CBC signals: Waveform recovery

In this exercise, you will be looking for relevant data from the *BayesWave* output to plot the following:

1. **Whitened detector data:** raw whitened detector data which contains noise and the GW signal (if you did not do Exercise 1, refer to its notes for definition of 'whitened' data/signal - In your lab report, you are expected to explain what whitened data means and its relevance to your analysis dataset);

2. **Injected waveform:** simulated BBH waveform (with similar parameters to GW150914) that we artificially added into detector noise to create a mock dataset for the *BayesWave* analysis;
3. **Recovered waveforms:** a posterior distribution of waveforms reconstructed by *BayesWave*.

You will be plotting these for each individual interferometer in the network i.e. LIGO Hanford (H1), LIGO Livingston (L1), Virgo (V1) and KAGRA (K1). Recall that *BayesWave* reconstructs non-Gaussian, transient features in the data regardless of origin using three seperate models as discussed in Section 6.1. We will focus on the signal plus Gaussian noise model ($\mathcal{S}$) first, and if you have time you can look into the glitch plus Gaussian noise model ($\mathcal{G}$) later. Figure 8 shows a sample of what you should achieve at the end of this exercise.

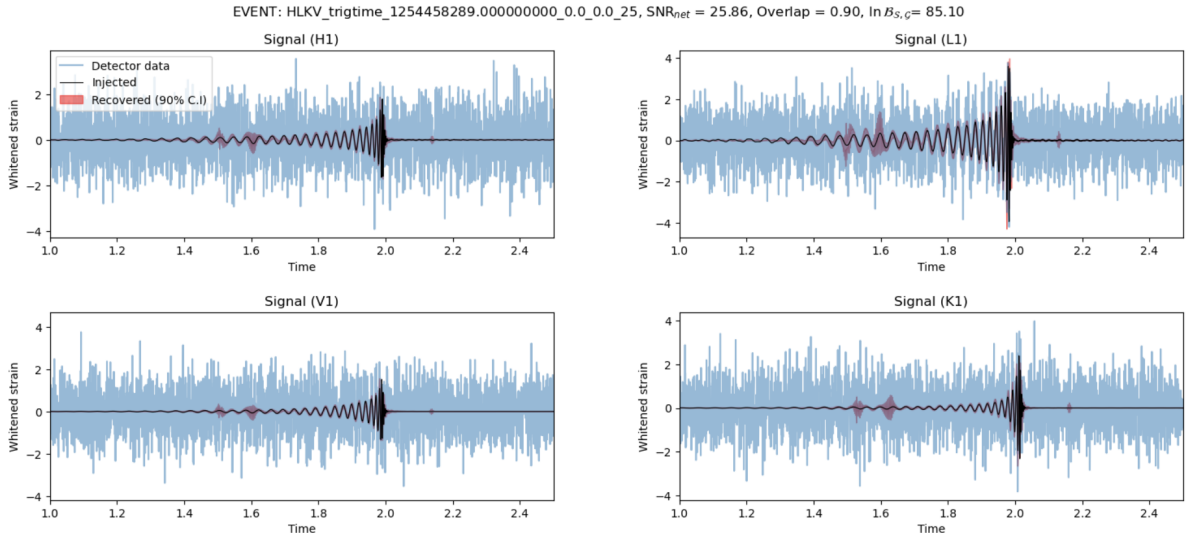

Figure 8: Sample plot for signal model $\mathcal{S}$.

Let's talk a bit more about the recovered waveforms. If you take a close look at Figure 8, the recovered waveform (labelled "Recovered (90% C.I.)") is not a single, solid line. As discussed previously, *BayesWave* reconstructs constructs the data using three separate models. For each of the models, *BayesWave* outputs a posterior distribution of the recovered waveforms, instead of a single waveform to represent the non-Gaussian, transient feature in the data. Therefore the "recovered waveform" seen in Figure 8 encloses a range of strain at a given time, within the 90% credible interval.

Now you may ask what does "credible interval" mean? In Bayesian statistics, the credible interval is simply the interval within which a parameter value would lie with a given probability. This interval exists in the domain of the posterior distribution of the parameter. For example, the 95% credible interval is the central portion of the posterior distribution which contains 95% of the parameter values (see Figure 9). The credible interval is not to be confused with *confidence* interval used in frequentist statistics. Recall that frequentists assume the observed parameter has a fixed true value. The *confidence* interval is the interval which contains the "true" value of the parameter to a specified level of confidence. The *confidence* interval is obtained from the distribution of a large number of observed samples (i.e. it is purely based on the data). Conversely, the credible interval accounts for the prior beliefs of the parameter distribution **and**

the data via Bayes' theorem.

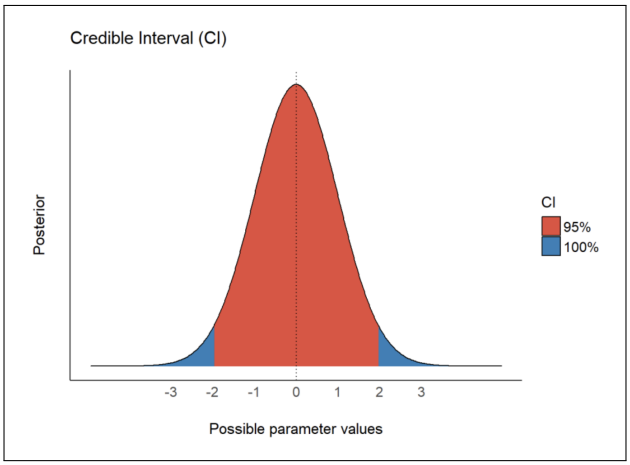

Figure 9: 95% and 100% credible interval of an arbitrary posterior distribution. Image from https://www.r-bloggers.com/2020/05/in-defence-of-the-95-ci/.

Finally, there are a couple of numbers in the title of Figure 8. Let's discuss what they mean:

1. $\mathbf{SNR_{net}}$ is the network signal-to-noise ratio (SNR) at which the simulated signal is injected into the network. For a multi-detector with $\mathcal{I}$ detectors, $\mathbf{SNR_{net}}$ is given by:

$$\mathrm{SNR}^2_{\mathrm{net}} = \sum_{\mathrm{i}=1}^{\mathcal{I}} \mathrm{SNR}^2_{\mathrm{i}} \tag{37}$$

   where SNR_i denotes the SNR of the signal in detector i.

2. **Overlap** quantifies the agreement between the true signal waveform and the ones recovered by *BayesWave*. Overlap= 1 implies a perfect match between the injected and recovered waveform; overlap= 0 implies that there is no match at all; overlap= -1 implies a perfect anticorrelation.

3. $\ln \mathcal{B}_{\mathcal{S},\mathcal{G}}$ is the log signal-versus-glitch model Bayes factor (see Equation 26). As discussed previously, the Bayes factor compares two competing hypothesis (models) via the strengths of their evidences (marginalised likelihood). In BayesWave, the hypotheses are the different types of waveform models e.g. $\mathcal{S}$ and $\mathcal{G}$. When $\ln \mathcal{B}_{\mathcal{S},\mathcal{G}} >= 0$ (i.e. when $\mathcal{B}_{\mathcal{S},\mathcal{G}} >= 1$), $\mathcal{S}$ is more strongly supported by the data compared to $\mathcal{G}$ and vice versa. In *BayesWave*, $\ln \mathcal{B}_{\mathcal{S},\mathcal{G}}$ is used to measure detection confidence of a signal!

In the first part of this exercise, you are given the *BayesWave* output of 3 different simulated injection events. For each of these events, you should produce plot as shown in Figure 8 by digging the *BayesWave* data. The notebook `GW_reconstruction.ipynb` has instructions on where you can find all the information described above (i.e. the time-domain detector data, injected waveform, recovered waveform posterior, $\mathrm{SNR}_{\mathrm{net}}$, overlap and $\ln \mathcal{B}_{\mathcal{S},\mathcal{G}}$). Once you have obtain the waveform plots for the events, qualitatively speculate how $\mathrm{SNR}_{\mathrm{net}}$ affects the overlap and *BayesWave*'s detection confidence (based on the 3 events you have).

In the second part of this exercise, you will be modifying the code for the 3 simulated events to reconstruct the first ever GW detection: GW150914. Basically you'll be reproducing your

own version of Figure 6. The instructions are given in the notebook, but we emphasise that the data of simulated and real GW events have minor but crucial differences. One of them (as mentioned in the notebook) is that real GW events do not have 'injected' waveforms i.e. we don't know what the actual signal looks like. There are a couple more differences, but we'll let you figure them out! When writing up your report, make sure to discuss these differences.

7 Acknowledgements

This lab is supported by The University of Melbourne School of Physics via the Laby Foundation. The authors thank Prof. Martin Sevior, Prof. Andrew Melatos, Tong Cheunchitra for their highly valued feedback on the lab notes and exercises; as well as Dr. Meg Millhouse for assisting with the GW150914 *BayesWave* data and Dr. Lucy Strang for their contributions.

References

[1]LIGO Scientific Collaboration and Virgo Collaboration, "Observation of Gravitational Waves from a Binary Black Hole Merger", Phys. Rev. Lett. **116**, 061102, 061102 (2016).

[2]LIGO Scientific Collaboration et al., "All-sky search for short gravitational-wave bursts in the first Advanced LIGO run", Phys. Rev. D. **95**, 042003, 042003 (2017).

[3]LIGO Scientific Collaboration, Virgo Collaboration, B. P. Abbott, et al., "All-sky search for short gravitational-wave bursts in the second Advanced LIGO and Advanced Virgo run", Phys. Rev. D. **100**, 024017, 024017 (2019).

[4]LIGO Scientific Collaboration et al., "All-sky search for short gravitational-wave bursts in the third Advanced LIGO and Advanced Virgo run", arXiv e-prints **100**, arXiv:2107.03701, arXiv:2107.03701 (2021).

[5]D. J. Griffiths et al., *Introduction to electrodynamics*, 4th ed., Vol. 100, 2 (Cambridge University Press, July 2017), p. 024017.

[6]B. F. Schutz et al., *A first course in General Relativity*, Vol. 100, 2 (Cambridge Univ. Pr., Cambridge, UK, July 1985), p. 024017.

[7]Hill, C. D and Nurowski P et al., "The Mathematics of Gravitational Waves", American Mathematical Society **1551**, 024017, 686 (2017).

[8]LIGO Scientific Collaboration et al., *Introduction to LIGO and gravitational waves*, July 2022.

[9]M. Sieniawska, M. Bejger, et al., "Continuous Gravitational Waves from Neutron Stars: Current Status and Prospects", Universe **5**, 024017, 217 (2019).

[10]B. P. Abbott et al., "Search for the isotropic stochastic background using data from Advanced LIGO's second observing run", Physical Review D **100**, 061101, 061101 (2019).

[11]J. Antoniadis, Z. Arzoumanian, et al., "The International Pulsar Timing Array second data release: Search for an isotropic gravitational wave background", Monthly Notices of the Royal Astronomical Society **510**, 024017, 4873–4887 (2022).

[12]B. P. Abbott et al., "A guide to LIGO-Virgo detector noise and extraction of transient gravitational-wave signals", Classical and Quantum Gravity **37**, 055002, 055002 (2020).