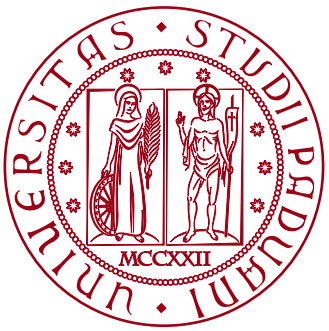



# UNIVERSITÀ DEGLI STUDI DI PADOVA

**Dipartimento di Fisica e Astronomia "Galileo Galilei"**

**Master Degree in Astrophysics and Cosmology**


**Final Dissertation**

## Machine Learning for Gravitational Waveforms from Binary Neutron Star mergers


**Thesis supervisors**

**Prof. Michela Mapelli**

**Prof. Sebastiano Bernuzzi**

**Thesis co-supervisor**

**Prof. Roberto Turolla**

**Candidate**

**Jacopo Tissino**


**Academic Year 2020/2021**

## Abstract

The analysis of interferometric data corresponding to the gravitational waves emitted in a binary neutron star merger requires the generation of numerous theoretical waveforms. In this thesis, I develop a machine learning algorithm trained on frequency-domain waveforms generated through the effective one-body technique, which is able to reconstruct them accurately and with a speed improvement.

# Introduction

[INTRODUCTION IS STILL PROVISIONAL]

The detection of a gravitational wave (GW) signal from the binary neutron star (BNS) merger GW170817 [Abb+17] opened a new avenue for the direct study of neutron star properties, such as their masses, spins and equations of state.

The study of an observed signal is done through Bayesian parameter estimation algorithms, which require the comparison of the experimental signal with several million [Lac+17] simulated waveforms or more (see section 1.3). Speed in the generation of simulated waveforms is, therefore, crucial.

There are three main strategies for the generation of these waveforms. In order of decreasing evaluation time, as well as (typically) decreasing accuracy, they are:

1. full numerical relativity (NR) simulations: these are our most accurate possible avenue for the study of BNS mergers, but they are so computationally expensive that it is reasonable to simulate at most a few tens of initial condition setups — these are then used to validate the other methods (see section 1.4.2);

2. effective one-body (EOB) simulations: these consist in constructing an effective Hamiltonian for a test particle as a proxy for the two-body system and simulating its evolution — they can be calibrated by comparison to NR simulations (see section 1.4.3);

3. post-Newtonian (PN) waveforms: here we expand the equations of motion of the system in powers of $1/c^2$ and analytically calculate the evolution of the amplitude and phase of the waveform (see section 1.4.1).

The EOB and PN formalisms can be complemented with a tidal term, which is relevant for BNS mergers: neutron stars (as opposed to black holes) can be deformed by each other's gravitational field in a way which depends on the specifics of their equation of state.

The evaluation times for EOB waveforms are dependent on the duration of the waveforms: GW170817 lasted for about two minutes, and the generation of waveforms of this length takes on the order of 100ms with state-of-the-art systems, but as the noise in ground-based interferometers decreases we expect to be able to see a longer and longer section of the waveform, which might bring the evaluation times back above 1s.

Also, the waveforms ought to be generated in the frequency domain, since that form is the one which is used in data analysis; EOB models, on the other hand, natively generate waveforms in the time domain.

The work in this thesis is an attempt to make progress in this direction, by developing a machine learning system trained on the (Fourier transforms of the) EOB waveforms. This builds on the work of Schmidt et al. [Sch+20], who developed a similar system to generate Binary Black Hole merger waveforms in the time domain.

The algorithm's basic components are as follows:

- a training dataset is generated by considering uniformly distributed tuples of BNS parameters in the allowed ranges and calculating the corresponding waveforms with the EOB model `TEOBResumS` [Nag+18];

- each time-domain waveform is Fourier-transformed and decomposed into phase and amplitude;
- each EOB waveform is compared to a PN one with the same parameters, and a dataset is constructed from the residuals of the EOB waveform from the PN one;
- the dimensionality of the training dataset is reduced through principal component analysis (PCA);
- a feed-forward neural network is trained to reconstruct the map between the BNS system parameters and the PCA components of the waveforms.

The resulting model must be evaluated in terms of both speed and accuracy. Regarding the former, proper benchmarking will begin at a later stage, but the results obtained by Schmidt et al. [Sch+20] indicate the possibility of obtaining a speedup of one to two orders of magnitude compared to EOB algorithms.

Regarding the latter, preliminary results indicate the possibility of the model reconstructing EOB waveforms with three parameters (the mass ratio $q$ and the two tidal deformabilities $\Lambda_1$ and $\Lambda_2$) with a maximal unfaithfulness of the order of $\mathcal{F} \simeq 10^{-3}$. This is comparable to typical values of the unfaithfulness of EOB waveforms compared to NR ones ($\mathcal{F} \simeq 2.5 \times 10^{-3}$ [Nag+18]).

This indicates that it is possible for this system to become a part of a gravitational wave data analysis pipeline, enabling a significant speedup at the cost a small, tolerable loss in accuracy.

Possible applications of the software developed in this work include modelled signal searches and parameter estimation for both current interferometers (such as LIGO and Virgo) and for future ones, such as Einstein Telescope.

# Chapter 1

# Gravitational wave theory

## 1.1 Linearized gravity

The simplest way to discuss gravitational radiation is to consider linearized gravity on a flat Minkowskian background.

This means that we assume that our spacetime admits a reference frame for which the metric is in the form

$$g_{\mu\nu} = \eta_{\mu\nu} + h_{\mu\nu}\,, \tag{1.1}$$

where the value of the components $h_{\mu\nu}$ is small enough that we can work to first order in them. Any equation in this section includes an implicit "$+\mathcal{O}\left(h^2\right)$". We work in this *global inertial frame*.

In order to study the evolution of the perturbation $h_{\mu\nu}$ we need to solve the Einstein Field Equations for it to linear order. In a vacuum, they can be written as

$$G_{\mu\nu} = R_{\mu\nu} - \frac{1}{2}\eta_{\mu\nu}R = 0 \tag{1.2}$$

$$R_{\mu\nu} = g^{\alpha\beta}R_{\alpha\mu\beta\nu} \tag{1.3}$$

$$R = g^{\mu\nu}R_{\mu\nu}\,, \tag{1.4}$$

where $R_{\alpha\mu\beta\nu} \sim \partial\Gamma + \Gamma\Gamma$ is the Riemann tensor, which is written in terms of derivatives and squares of Christoffel symbols $\Gamma$:

$$\Gamma^{\rho}_{\mu\nu} = \frac{1}{2}g^{\rho\lambda}\left(\partial_\mu g_{\nu\lambda} + \partial_\nu g_{\lambda\mu} - \partial_\lambda g_{\mu\nu}\right) \tag{1.5}$$

$$= \frac{1}{2}\eta^{\rho\lambda}\left(\partial_\mu h_{\nu\lambda} + \partial_\nu h_{\lambda\mu} - \partial_\lambda h_{\mu\nu}\right). \tag{1.6}$$

We use the fact that $\partial\eta = 0$, and keep only linear order terms.

Since the Christoffel symbols are of first order in the perturbation, the term $\Gamma\Gamma$ in the Riemann tensor is of second order and can be neglected. Therefore, the relevant components are

$$R_{\mu\nu\rho\sigma} = 2\eta_{\mu\lambda}\Gamma^{\lambda}_{\nu[\sigma,\rho]} = h_{\mu[\sigma,\rho]\nu} - h_{\nu[\sigma,\rho]\mu}\,, \tag{1.7}$$

which gives us the following expression for the Ricci tensor:

$$R_{\mu\nu} = \frac{1}{2}\left(h^{\sigma}{}_{\mu,\sigma\nu} + h^{\sigma}{}_{\nu,\sigma\mu} - h_{,\mu\nu} - \Box h_{\mu\nu}\right), \tag{1.8}$$

where $h = \eta^{\mu\nu} h_{\mu\nu}$ is the trace of the perturbation (computed with respect to the flat metric), while $\Box = \eta^{\mu\nu}\partial_\mu\partial_\nu$ is the flat space d'Alambertian.

This, in turn, allows us to write out the Einstein tensor:

$$G_{\mu\nu} = \frac{1}{2}\left(h^{\sigma}{}_{\mu,\sigma\nu} + h^{\sigma}{}_{\nu,\sigma\mu} - h_{,\mu\nu} - \Box h_{\mu\nu} - \eta_{\mu\nu} h^{\rho\lambda}{}_{,\rho\lambda} + \eta_{\mu\nu}\Box h\right). \tag{1.9}$$

This can be greatly simplified with two steps: first, we change variable from the perturbation $h_{\mu\nu}$ to the *trace-reversed* perturbation $\bar{h}_{\mu\nu} = h_{\mu\nu} - \eta_{\mu\nu} h/2$ — the name comes from the fact that $\eta^{\mu\nu}\bar{h}_{\mu\nu} = -h$.

This substitution allows us to write the Einstein tensor as

$$G_{\mu\nu} = -\frac{1}{2}\Box\bar{h}_{\mu\nu} + \bar{h}_{\alpha(\mu,\nu)}{}^{\alpha} - \frac{1}{2}\eta_{\mu\nu}\bar{h}_{\alpha\beta}{}^{,\alpha\beta}\,. \tag{1.10}$$

We will shortly show that it is possible, as a *gauge choice*, to set the divergence of the trace-reversed perturbation to zero: $\partial^\mu\bar{h}_{\mu\nu} = 0$ (see section 1.1.2). The gauge imposed by this choice is called the *Hilbert Gauge*,[1] which in terms of the regular perturbation reads

$$\partial^\mu h_{\mu\nu} - \frac{1}{2}\partial_\nu h = 0\,. \tag{1.11}$$

With this choice the Einstein tensor becomes simply

$$G_{\mu\nu} = -\frac{1}{2}\Box\bar{h}_{\mu\nu}\,, \tag{1.12}$$

so the general form of the Einstein equations to linear order will be

$$\boxed{\Box\bar{h}_{\mu\nu} = -16\pi G\; T_{\mu\nu}\Big|_{\text{linear}}\,,} \tag{1.13}$$

where the stress-energy tensor is computed up to first order in the metric perturbation.

### 1.1.1 Transformations of the perturbation

The theory of General Relativity is constructed to be invariant under smooth changes of coordinates: under a map in the form $x \to x' = x'(x)$ (where $x'(x)$ is a diffeomorphism[2]).

Under such a coordinate transformation the metric transforms like any $(0,2)$ tensor:

$$g'_{\alpha\beta}(x') = \frac{\partial x^\mu}{\partial x'^\alpha}\frac{\partial x^\nu}{\partial x'^\beta} g_{\mu\nu}(x)\,. \tag{1.14}$$

A general transformation of this kind may break the condition that $g = \eta + h$ where $h$ is small, so in order to preserve our framework we restrict ourselves to a small class of transformations.

[1] Despite the name, this choice was first suggested by De Sitter to Einstein [Ken07], who had been previously trying to impose the gauge $|g| = 1$ [Ein16, page 688].

[2] In the physics parlance this property is known as "diffeomorphism invariance", while a mathematician would call the kinds of transformations considered "isometries", since we ask that they preserve the metric structure of the manifold.

One possibility is to consider infinitesimal transformations in the form

$$x^\mu \to x'^\mu = x^\mu + \xi^\mu(x)\,, \tag{1.15}$$

where $\xi^\mu$ is a vector field such that $\left|\partial_\mu \xi_\nu\right|$ is small — specifically, the condition to impose is that the first order in $\partial_\mu \xi_\nu$ should match the first order in $h_{\mu\nu}$.

This condition is all we need in order to write the transformation law for the perturbation: the full equation reads

$$\eta'_{\mu\nu} + h'_{\mu\nu} \approx \left(\delta^\alpha_\mu - \partial^\alpha \xi_\mu\right)\left(\delta^\beta_\nu - \partial^\beta \xi_\nu\right)\left(\eta_{\alpha\beta} + h_{\mu\nu}\right), \tag{1.16}$$

Used $1/(1+x) = 1 - x + \mathcal{O}\left(x^2\right)$.

so the zeroth order contribution is $\eta'_{\mu\nu} = \eta_{\mu\nu}$, while the first order one is

$$\boxed{h'_{\mu\nu} = h_{\mu\nu} - 2\partial_{(\mu}\xi_{\nu)}\,,} \tag{1.17}$$

which is our transformation law for the metric perturbation.

We will also need a transformation law for the trace-reversed perturbation $\overline{h}_{\mu\nu}$: the trace transforms as $h' \to h - 2\partial_\mu \xi^\mu$, therefore the required law is

$$\overline{h}'_{\mu\nu} = \overline{h}_{\mu\nu} - 2\partial_{(\mu}\xi_{\nu)} + \eta_{\mu\nu}\partial_\alpha \xi^\alpha\,. \tag{1.18}$$

A second class of transformations which sometimes preserve the structure $g = \eta + h$ is a subset of Lorentz boosts and rotations: substituting $\partial x^\mu / \partial x'^\nu = \Lambda_\nu{}^\mu$ into the transformation law (1.14) we find that the flat metric is unchanged, while

$$h'_{\mu\nu}(x') = \Lambda_\mu{}^\alpha \Lambda_\nu{}^\beta h_{\alpha\beta}\,, \tag{1.19}$$

which may remain in the class of small metric perturbations: this is not guaranteed, but it is true for a certain subset of boosts and for all rotations [Mag07].

Finally, the perturbation is invariant under shifts in the form $x'^\mu = x^\mu + a^\mu$.

### 1.1.2 Gauge fixing

Now that we know how the perturbation $h_{\mu\nu}$ transforms under an infinitesimal transformation, we can use this to impose the condition we want — specifically, the Hilbert gauge (1.11).

The way to show that this is possible is to write out the way $\partial^\mu \overline{h}_{\mu\nu}$ transforms for an arbitrary choice of $\xi$, and to see that with an appropriate choice of $\xi$ we can always map it to zero. The transformation reads

$$\partial^\mu \overline{h}'_{\mu\nu} = \partial^\mu\left(h_{\mu\nu} - 2\partial_{(\mu}\xi_{\nu)}\right) - \frac{1}{2}\partial_\nu\left(\eta^{\alpha\beta}\left(h_{\alpha\beta} - 2\partial_{(\alpha}\xi_{\beta)}\right)\right) \tag{1.20}$$

$$= \partial^\mu \overline{h}_{\mu\nu} - \Box\xi_\nu - \partial_\nu\left(\partial^\mu \xi_\mu\right) + \partial_\nu\left(\partial^\mu \xi_\mu\right) \tag{1.21}$$

$$= \partial^\mu \overline{h}_{\mu\nu} - \Box\xi_\nu\,. \tag{1.22}$$

Therefore, from any starting gauge we must only find a $\xi_\nu$ such that $\partial^\mu \overline{h}_{\mu\nu} = \Box\xi_\nu$, and we will be in the correct gauge. This can always be done, since the D'Alambert equation $\Box f = g$ can always be solved for $f$ — if we needed to compute $\xi_\nu$ explicitly

(which we typically do not) we could use the Green's function $G(z)$ for the operator, defined by $\Box G(z) = \delta^{(4)}(z)$.

While the equation is solvable, the solution is not unique: if we were to define an "inverse" of the D'Alambertian it would not be a function but a one-to-many relation. Specifically, while keeping fixed the value of $\Box\xi_\nu$ we can add any function $\zeta_\nu$ to $\xi_\nu$ as long as $\Box\zeta_\nu = 0$. A trivial example is $\zeta_\nu = \text{const}$, but other wave-like choices are of more interest.

These still induce a transformation on $h_{\mu\nu}$ according to the usual law (1.17), and they can be used to further specify the form of the gravitational radiation.

In terms of **degrees of freedom**, the full perturbation $h_{\mu\nu}$ starts with 10 as any symmetric 4D, rank-2 tensor; the four Hilbert gauge conditions (1.11) reduce them to 6, while the four residual gauge conditions will allow us, under some specific assumptions, to reduce them to 2.

Let us fix the residual gauge in the specific context of a plane wave solution to the wave equation $\Box\overline{h}_{\mu\nu} = 0$.

### 1.1.3 Plane gravitational waves

A general solution to the wave equation may be written as a superposition of plane waves in the form

$$\overline{h}_{\mu\nu} = A_{\mu\nu} e^{ik_\alpha x^\alpha}\,, \tag{1.23}$$

where $A_{\mu\nu}$ is a constant symmetric tensor. Let us then consider a single plane wave.

Imposing the wave equation sets $k_\alpha k^\alpha = 0$, and the Hilbert gauge (1.11) condition can be written as $A_{\mu\nu}k^\mu = 0$, where $A = \eta^{\mu\nu}A_{\mu\nu}$.

In this framework we can impose the residual gauge condition explicitly: a function which satisfies $\Box\zeta_\mu = 0$ is $\zeta_\mu = B_\mu \exp(id_\alpha x^\alpha)$, where $d_\alpha$ is a null vector ($d_\alpha d^\alpha = 0$) while $B_\mu$ is a generic constant vector.

In these terms, the transformation equation (1.18) reads

$$\overline{h}_{\mu\nu} \to \overline{h}_{\mu\nu} + \left(-2iB_{(\mu}d_{\nu)} + i\eta_{\mu\nu}B_\beta d^\beta\right)e^{id_\alpha x^\alpha} \tag{1.24}$$

$$A_{\mu\nu}e^{ik_\alpha x^\alpha} \to A_{\mu\nu}e^{ik_\alpha x^\alpha} + \left(-2iB_{(\mu}d_{\nu)} + i\eta_{\mu\nu}B_\beta d^\beta\right)e^{id_\alpha x^\alpha}\,. \tag{1.25}$$

This tells us that if we set the vector $d_\alpha$ to be equal to $k_\alpha$ the amplitude $A_{\mu\nu}$ will transform according to the algebraic system

$$A_{\mu\nu} \to A_{\mu\nu} - 2iB_{(\mu}k_{\nu)}\,, \tag{1.26}$$

which allows us to impose four conditions on $A_{\mu\nu}$, one for each of the free components of $B_\mu$. It is customary to choose $A = 0 = A_{0i}$: these are known together as the transverse-traceless (TT) gauge.

The condition $A = 0$ also means that $h = 0 = \overline{h}$: thus, form this point onward we can stop distinguishing between $h_{\mu\nu}$ and $\overline{h}_{\mu\nu}$, and for simplicity's sake we write the former.

If we orient our axes such that $\vec{k} = k^i$ is along the $\hat{z}$ direction (which means $k^\mu = (k, 0, 0, k)^\top$) the conditions can be written as

1. Hilbert gauge + traceless: $A_{\mu 0} + A_{\mu 3} = 0$;

2. traceless: $A = -A_{00} + A_{11} + A_{22} + A_{33} = 0$;

3. transverse: $A_{0i} = 0$.

The Hilbert gauge combined with the transverse conditions show that $A_{00} = 0$ as well, followed by $A_{13} = 0$, $A_{23} = 0$ and $A_{33} = 0$.

Finally, the traceless condition imposes $A_{11} = -A_{22}$. These conditions tell us that the plane gravitational wave must have the form

$$h^{TT}_{\mu\nu}(x) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & h_+ & h_\times & 0 \\ 0 & h_\times & -h_+ & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \exp(ik_\alpha x^\alpha)\,, \tag{1.27}$$

where $h_+ = A_{11}$ and $h_\times = A_{12}$.

In a later discussion we will use the fact that a general perturbation $h_{\mu\nu}$ can be *projected* into the TT gauge by using a certain projection tensor $\Lambda_{ijkl}$; this is not trivial to show, but the general idea follows from a decomposition of the full perturbation into scalar, vector and tensor components [Car19]. Specifically, the number of free components before gauge fixing is 4 scalar, 4 vector and 2 tensor components. The tensor components are gauge invariant, while the other 8 can each be separately set to zero in an appropriate gauge. After applying the same decomposition to the stress-energy tensor and writing the Einstein equations we find that the tensor perturbations (which are precisely the $h_+$ and $h_\times$ polarization degrees of freedom) are the only ones which propagate according to a wave equation. The projection, which yields a perturbation with two degrees of freedom, is thus only a way to recover the "radiative" part of the metric.

### 1.1.4 Effect on test masses

The TT gauge we defined has the rather peculiar characteristic of "moving with the wave", so that the position of any observer initially at rest (so, such that $u^\mu(\tau = 0) = (1, \vec{0})^\top$) is unchanged: the geodesic equation evaluated at $\tau = 0$ reads

$$\left.\frac{\mathrm{d}^2 x^\mu}{\mathrm{d}\tau^2}\right|_{\tau=0} + \left.\Gamma^\mu_{\nu\rho} u^\nu u^\rho\right|_{\tau=0} = 0 \tag{1.28}$$

$$\left.\frac{\mathrm{d}^2 x^\mu}{\mathrm{d}\tau^2}\right|_{\tau=0} = -\Gamma^\mu_{00} = 0\,, \tag{1.29}$$

since the Christoffel symbols computed $\Gamma^\mu_{00}$ with the TT gauge perturbation (1.27) all vanish: $\Gamma^\mu_{00} = \eta^{\mu\nu}(2g_{\nu 0,0} - g_{00,\nu})/2 = 0$.

Does this mean that gravitational waves are merely an artefact, and have no effect on particles? No, since while in the TT gauge the *positions* of the points do not change, the *distance* among them does.

In order to understand this effect we can make use of the geodesic deviation equation [Car19, section 3.10], which states that if we take two geodesics whose four-velocities are

both approximately $u^\mu$, separated by a short vector $\xi^\mu$,[3] they might diverge or converge, and the evolution of $\xi^\mu$ will be described by

$$\ddot{\xi}^\mu = R^\mu_{\nu\rho\sigma} u^\nu u^\rho \xi^\sigma \,. \tag{1.30}$$

Let us consider the same geodesics as before, whose four-velocity is uniformly $u^\mu \equiv \left(1, \vec{0}\right)^\top$: the "acceleration" will then be given by the matrix product $R^\mu_{00\sigma}\xi^\sigma$.

These components of the Riemann tensor read [Car19, eq. 7.106]:

$$R^\mu_{00\sigma} = \frac{1}{2}\ddot{h}^\mu_\sigma \,. \tag{1.31}$$

Therefore, the temporal component of the acceleration is $\ddot{\xi}^0 \propto \ddot{h}^0_\sigma = 0$, while the spatial components read

$$\ddot{\xi}^i = \frac{1}{2}\ddot{h}^i_j \xi^j \,. \tag{1.32}$$

These equations can then be explicitly solved. It is important to remark that $\ddot{\xi}^i$ is *not* coordinate acceleration: the points are *stationary* in the TT gauge coordinates, but the distance among them changes. Intuitively, we can interpret this as due to our carefully constructed gauge choice, which was determined through residual gauge, so it satisfied a wave equation $\Box\zeta_\mu = 0$: it precisely oscillates with the wave, allowing the position of any point to not change (to linear order). Fortunately GWs are physical, so they must still manifest regardless of our gauge, and they do so through the distortion of distances as described here.

The distances of points at a fixed position $(x, y, z)$ from the center change (up to a phase) according to the expression

$$x(t) = x_0\left(1 + h_+ e^{i\omega t}\right) + y_0 h_\times e^{i\omega t} \tag{1.33}$$

$$y(t) = y_0\left(1 - h_+ e^{i\omega t}\right) + x_0 h_\times e^{i\omega t} \,, \tag{1.34}$$

which is shown graphically in figure 1.1 for four configurations: only $h_+$ or $h_\times$ being nonzero, or only $h_R \propto h_+ + ih_\times$ or $h_L \propto h_+ - ih_\times$ being nonzero. The last two are called *circular polarizations*.

This description is not the one an experimentalist might use: if we need to consider the noise due to other sources of gravitation, it is convenient to move from the TT gauge to the *proper detector frame*, in which the metric is expanded around a fiducial point of our detector — for instance, around the beam-splitter in an interferometer. In such a frame, it can be shown that the effect of GWs can be described as a Newtonian force on points a certain distance away from it — for instance, the mirrors of the interferometer. This comes in at second order in the distance, and it is given by [Mag07, eq. 1.96]

$$F_i = \frac{m}{2}\ddot{h}^{TT}_{ij}\xi^j \,, \tag{1.35}$$

[3] There is a technical note to be made here: in GR the notion of a vector between two points in the manifold is meaningless, however we can get around this problem by considering a one-parameter family of geodesics, and identifying the separation vector between them to be the tangent vector associated to the parameter.

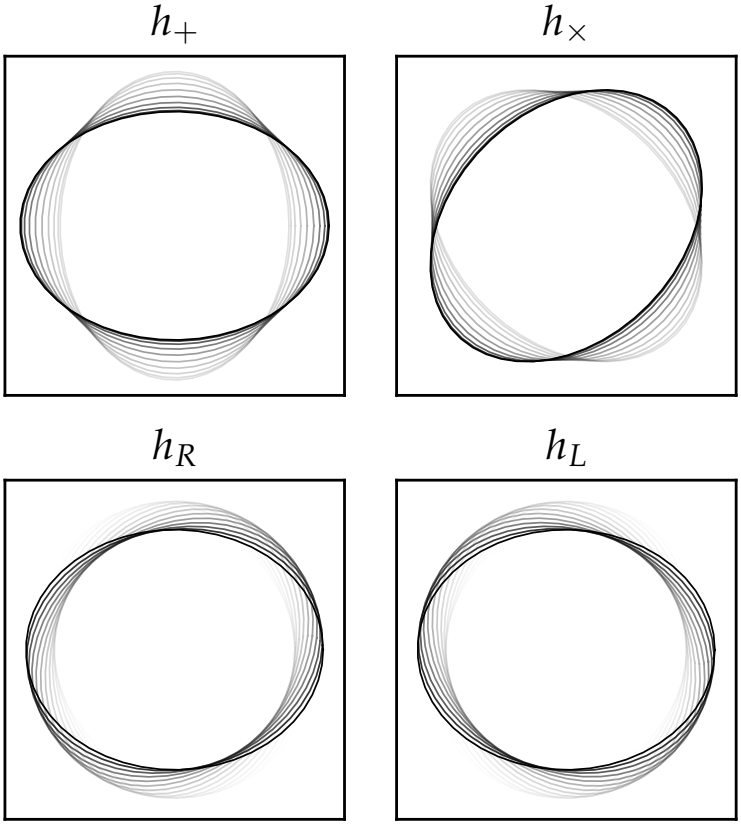


Figure 1.1: Polarizations of gravitational waves. Time evolution is represented though transparency: darker ellipses correspond to later points in time. The four cases are described in the text. We stress that in the TT gauge the ellipses do not represent changes in coordinate position but only in distance among points. In all cases the starting configuration is circular. The code generating this figure is available here.

which unsurprisingly closely mirrors the TT gauge variation of the distance: the physically measurable effect must be independent of gauge choice. The reason the TT gauge perturbation appears here as well is that the Riemann tensor, which is used in these computations, is *invariant* (a stronger condition than covariant — the values of the components are actually unchanged) under gauge transformations.

### 1.1.5 The quadrupole formula

The lowest order contribution to the generation of gravitational waves can be calculated starting from the linearized Einstein equations (1.13). The Green's function method for the inversion of the D'Alambertian is a common technique: if we can find a function $G(z)$ such that $\Box G(z) = \delta^{(4)}(z)$, then we can do the following manipulation (denoting the gravitational constant as $G_N$ for clarity):

$$\Box_x \bar{h}_{\mu\nu}(x) = -16\pi G_N T_{\mu\nu}(x) \tag{1.36}$$

$$= -16\pi G_N \int \mathrm{d}^4 y \, T_{\mu\nu}(y)\delta^{(4)}(x-y) \tag{1.37}$$

$$= -16\pi G_N \int \mathrm{d}^4y\, T_{\mu\nu}(y)\Box_x G(x-y) \tag{1.38}$$

$$= \Box_x\left(-16\pi G_N \int \mathrm{d}^4y\, T_{\mu\nu}(y) G(x-y)\right) \tag{1.39}$$

$$\bar{h}_{\mu\nu} = -16\pi G_N \int \mathrm{d}^4y\, T_{\mu\nu}(y) G(x-y) \tag{1.40}$$

$$= 4G_N \int \mathrm{d}^3y\, T_{\mu\nu}(x^0 - |\vec{x}-\vec{y}|, \vec{y}) \frac{1}{|\vec{x}-\vec{y}|}\,, \tag{1.41}$$

Removing the D'Alambertian is valid insamuch as it will give us *a* solution for the wave equation, which will not be unique: $\Box f = \Box g$ does not imply $f = g$, but the reverse is true.

where in the last step we introduced the explicit expression for the Green's function of the D'Alambertian: $G(z) = -\Theta(z^0)/4\pi|\vec{z}|$ [Mag07, eq. 3.6].

We then have a formula for the trace-reversed perturbation, and as we discussed in section 1.1.3 far from the source we can recover the TT gauge perturbation by projecting this equation thanks to a tensor $\Lambda_{ij,kl}(\hat{n})$ (we return to $G \equiv G_N$ denoting the gravitational constant):

$$h^{TT}_{ij}(t,\vec{x}) = 4G\Lambda_{ij,kl}(\hat{n}) \int \frac{\mathrm{d}^3y}{|\vec{x}-\vec{y}|} T_{kl}\Big(t - |\vec{x}-\vec{y}|, \vec{y}\Big)\,, \tag{1.42}$$

where the tensor $\Lambda$ is constructed to be a projector which sends 2-tensors into the subspace of traceless tensors, which are transverse to the direction defined by $\hat{n} = \vec{x}/|\vec{x}|$:

$$\Lambda_{ij,kl} = P_{ik}P_{jl} - \frac{1}{2}P_{ij}P_{kl} \qquad \text{where} \qquad P_{ij} = \delta_{ij} - n_i n_j\,. \tag{1.43}$$

This expression can be further simplified by making use of the assumption that the source is far away: the integral over $\mathrm{d}^3y$ ranges over a region which has a maximum size of $R$, the scale of the source, while the wave is observed at a distance $r = |\vec{x}| \gg R$. This allows us to expand and neglect terms of the order $R^2/r^2$ and over;[4] the result we get is

$$h^{TT}_{ij} = \frac{4G}{r}\Lambda_{ij,kl}(\hat{n}) \int \mathrm{d}^3y\, T_{kl}\big(t - r + \vec{y}\cdot\hat{n}, \vec{y}\big)\,. \tag{1.45}$$

This equation uses the stress-energy tensor as computed without considering the higher-order effects of gravity on matter (this is the meaning of "linear" in the linear wave equation (1.13)); thus it is only a good approximation in the case of objects whose typical scale $R$ is much larger, than, say, their Schwarzschild radius $2GM$. The problem with this assumption is that it fails to hold precisely for the sources which we are most interested in since, as we shall see, they give out some of the most easily detectable gravitational radiation: binary compact objects near coalescence.[5]

For a gravitationally bound source like a binary, with total mass $M$, reduced mass $\mu$, moving with speed $v$ and with a relative separation $r$, the virial theorem dictates that

$$\frac{1}{2}\mu v^2 = \frac{1}{2}G\frac{\mu M}{r} \implies v^2 = \frac{GM}{r} = \frac{R_s}{r}\,, \tag{1.46}$$

[4] Specifically, we expand

$$|\vec{x}-\vec{y}| = \sqrt{x^2+y^2+2x\cdot y} = r\sqrt{1 - 2\frac{\hat{n}\cdot\vec{y}}{r} + \frac{y^2}{r^2}} \approx r\left(1 - 2\frac{\hat{n}\cdot\vec{y}}{r} + \mathcal{O}\left(R^2/r^2\right)\right). \tag{1.44}$$

[5] As we shall discuss later, the amplitude of the waves emitted rises as they get closer and closer; also, BNSs and binary black holes (BBHs) often reach their peak amplitude in the frequency range where our detectors are most sensitive. Nevertheless, the "Newtonian" stage of the inspiral (a long time before the merger) will be described quite well by the expressions discussed here.

where $R_s$ is the Schwarzschild radius corresponding to the total mass of the system.

This means that the source not being very compact[6] ($R_s \ll r$) is equivalent to it moving slowly ($v \ll 1$). Neither of these assumptions will hold in the end, but since we are using one we should use both.

In Fourier space, this amounts to saying that the typical frequencies $\omega$ for which the amplitude of the Fourier transform of the stress-energy tensor is large will satisfy $\omega\vec{y}\cdot\hat{n} \lesssim \omega R \ll 1$. This means that we can expand the exponential in the Fourier transform; in the time domain this amounts to expanding in time around the retarded time $t - r$:

$$T_{kl}(t - r + \vec{y}\cdot\hat{n}, \vec{y}) \approx T_{kl}(t - r, \vec{y}) + \vec{y}\cdot\hat{n}\partial_t T_{kl} + \frac{1}{2}(\vec{y}\cdot\hat{n})^2\partial_{tt}^2 T_{kl} + \mathcal{O}\left((\vec{y}\cdot\hat{n})^3\right). \tag{1.47}$$

This allows us to write the resulting wave as a multipole expansion:

$$h_{ij}^{TT}(t, \vec{x}) = \frac{4G}{r}\Lambda_{ij,kl}\left(S_{kl} + \sum_{L=1}^{\infty}\frac{1}{L!}\partial_t^L S_{kl|i_1\ldots i_L} n_{i_1}\ldots n_{i_L}\right), \tag{1.48}$$

To be computed at redarded time.

where the moments of the stress tensor $S_{kl|i_1\ldots i_L}$ are defined as

$$S_{kl|i_1\ldots i_L} = \int \mathrm{d}^3 y\, T_{kl} y_{i_1}\ldots y_{i_L}. \tag{1.49}$$

We can also analogously calculate moments for the energy density $T_{00}$, which we denote $M_{i_1\ldots i_L}$, and for the momentum density $T_{0i}$, which we denote $P_{k|i_1\ldots i_L}$.

With integration by parts combined with the (flat space!) conservation of the stress-energy tensor $\partial_\mu T^{\mu\nu} = 0$ we can relate the $M$, $P$ and $S$ with equations such as [Mag07, eqs. 3.45–51]

$$\dot{M} = 0 \tag{1.50}$$
$$\dot{P}_i = 0 \tag{1.51}$$
$$\dot{M}_{ij} = 2P_{(i|j)} \tag{1.52}$$
$$S_{ij} = P_{i|j}, \tag{1.53}$$

where dots denote time derivatives.

These all tell us something interesting: $\dot{M} = 0$ and $\dot{P}_i = 0$ are energy and momentum conservation, which seem to tell us that there is no energy nor momentum loss from GW emission. This is an artifact, due to the assumptions of linear theory which neglect back-action on the source; fortunately we will still be able to compute the energy loss of the system even in this approximation.

The second useful fact is $S_{ij} = \ddot{M}_{ij}/2$: this allows us to write the lowest-order approximation of the expression of the wave from the source in terms of moments (1.48) as

$$h_{ij}^{TT}(t, \vec{x}) = \frac{2G}{r}\Lambda_{ij,kl}\ddot{M}_{kl}, \tag{1.54}$$

where

$$M_{kl}(t - r) = \int \mathrm{d}^3 y\, T_{00}(t - r, \vec{y}) y_k y_l. \tag{1.55}$$

[6] We are being loose with terminology: the two stars in the binary may be compact themselves, but the system as a whole isn't.

since this expression only depends on the trace-free part of the moment $M$ we can write it as a function of the traceless **quadrupole moment**

$$Q_{kl} = M_{kl} - \delta_{kl} M_{nn}/3 = \int \mathrm{d}^3 y\, \rho(t, \vec{y}) \left( y^i y^j - \frac{1}{2} \delta^{ij} y^2 \right). \tag{1.56}$$

This gives rise to the quadrupole formula

$$h_{ij}^{TT}(t, \vec{x}) = \frac{2G}{r} \Lambda_{ij,kl} \ddot{Q}_{kl}(t - r). \tag{1.57}$$

### 1.1.6 Energy loss through gravitational radiation

The problem of quantifying the energy carried by gravitational radiation is thorny: first of all, there is no universally valid way to split the perturbation from the background in the general case; also, at each point we can always apply the equivalence principle to recover flat spacetime up to first order. It is impossible to construct a *local*, gauge invariant stress-energy tensor $T_{\mu\nu}^{GW}$ for GW radiation: the limit of the energy density contained in any volume will always vanish as that volume goes to zero.

However, we *can* define a tensor through an averaging procedure over many wavelengths and periods of the wave. There are different ways to do so, but a common one is the Landau-Lifshitz pseudotensor:

$$t_{\mu\nu} = -\frac{1}{8\pi G} \left\langle R_{\mu\nu}^{(2)} - \frac{1}{2} \overline{g}_{\mu\nu} R^{(2)} \right\rangle_{\text{mesoscopic}} \tag{1.58}$$

$$= \frac{1}{32\pi G} \left\langle \partial_\mu h_{\alpha\beta} \partial_\nu h^{\alpha\beta} \right\rangle, \tag{1.59}$$

where $R_{\mu\nu}^{(2)}$ and $R^{(2)}$ are the components of the Ricci tensor and scalar which are quadratic in the perturbation $h_{\mu\nu}$; $\overline{g}_{\mu\nu}$ is the background metric, and the averaging procedure is done on scales (wavelengths/periods) which are (much) larger than the typical wavelengths of the gravitational radiation considered, but (much) smaller than the typical wavelengths of the background [Mag07, sections 1.4.2, 1.4.3].

This pseudotensor can be used to describe the way in which, on large enough scales, the presence of GWs does indeed curve spacetime. Also, we can compute the energy flux passing through a surface a large distance from the source: if we use the quadrupole formula (1.57) for the gravitational perturbation, we can give an expression for the emitted power in terms of the third derivatives of the quadrupole as [Mag07, eq. 3.98]

$$\frac{\mathrm{d}E}{\mathrm{d}t} = -\frac{G}{5} \left\langle \dddot{Q}_{ij} \dddot{Q}_{ij} \right\rangle. \tag{1.60}$$

Dimensionally, $Q \sim \int \mathrm{d}^3 y\, \rho r^2$ has units of $\mathrm{kgm}^2$; so $\dddot{Q}$ has units of $\mathrm{kgm}^2/\mathrm{s}^3 = \mathrm{W}$.

This means that the prefactor, which we wrote using $c = 1$, must really be $Gc^n/5$ with units of inverse power, which implies $n = -5$. Numerically, it is the inverse of $5c^5/G \approx 2 \times 10^{53}\,\mathrm{W}$. This means that in order for GW emission to be efficient we must have a large value for $\dddot{Q}$: let us estimate it in terms of the typical size of the system, $R$, of its typical velocity $v = \Omega R$ and of its mass $M$. Each time derivative will roughly correspond to multiplication by a factor $\Omega$, so $\dddot{Q} \sim \Omega^3 M R^2 = v^3 M/R$.

The power can then be estimated as

$$-\frac{\mathrm{d}E}{\mathrm{d}t} \sim \frac{G}{5c^5} v^6 \frac{M^2}{R^2} = \underbrace{\frac{1}{5}\frac{c^5}{G}}_{\sim 10^{52}\,\mathrm{W}} \left(\frac{v}{c}\right)^6 \left(\frac{GM}{c^2 R}\right)^2, \tag{1.61}$$

which tells us that the most significant sources of GW will be relativistic and compact. This prediction has been validated in 2015 with the first detection of GW from a BBH system [LIG+16], and again in 2017 with the first detection of GW from a BNS system [Abb+17].

## 1.2 Compact binaries

We focus our attention towards a pair of inspiraling (compact) objects, which we will initially model as point masses in quasi-circular orbits. The first thing we need to do is to write an expression the amplitudes in the two polarizations $h_{+,\times}$ of the waves generated by a generic mass distribution with a mass moment $M_{ij}$ [Mag07, eqs. 3.67–68]:[7]

$$h_+(t,\hat{n}) = \frac{G}{r}\left(\ddot{M}'_{11} - \ddot{M}'_{22}\right) \tag{1.62}$$

$$h_+(t,\hat{n}) = \frac{G}{r}\left(2\ddot{M}'_{12}\right), \tag{1.63}$$

where $M'_{ij}$ are the components of the mass moment tensor in a frame whose $z'$ axis is aligned to the observation direction.

This frame is not the natural one with which to describe a binary system: typically, we would want to align the $z$ axis of the coordinates with the rotation axis. The rotation matrix between two systems depends on two angles, however one of these is more important than the other: aligning the $z$ axes can be accomplished by a rotation of angle $\iota$ (dubbed the *inclination*), while another rotation of angle $\phi$ is needed in order to align the $x$ and $y$ axes as well. However, since the system is rotating around its axis in a quasi-circular orbit, the second rotation only amounts to a time shift, an additional phase term in the oscillatory functions. For simplicity, we will neglect this phase freedom here and only include a variable phase at the end of the computations.

If the position of the bodies in the center-of-mass frame is

$$\vec{x}(t) = R \begin{bmatrix} \cos(\Omega t) \\ \sin(\Omega t) \\ 0 \end{bmatrix}, \tag{1.64}$$

then the mass moments read $M_{ij} = \mu x_i(t) x_j(t)$ (where $\mu = m_1 m_2/(m_1+m_2)$ is the reduced mass of the system), and going through the computation yields [Mag07, eq. 3.332]

$$h_+(t) = \underbrace{\frac{4G\mu\Omega^2 R^2}{r}}_{A} \left(\frac{1+\cos^2\iota}{2}\right) \cos(2\Omega t) \tag{1.65}$$

[7] We could also write these in terms of the traceless quadrupole moment $Q_{ij}$; we follow the convention set forward by Maggiore [Mag07].

$$h_\times(t) = \underbrace{\frac{4G\mu\Omega^2R^2}{r}}_{A}\cos\iota\sin(2\Omega t) \tag{1.66}$$

$$A = \frac{4}{r}\left(\frac{G\mathcal{M}_c}{c^2}\right)^{5/3}\left(\frac{\pi f_{\rm gw}}{c}\right)^{2/3}. \tag{1.67}$$

Approximating the motion of the bodies as circular during an orbit is called the *adiabatic approximation*, since it assumes that no energy is lost in each orbit (as that would deform the circle). The variation of the orbital frequency $\Omega$, which we shall compute shortly, must satisfy $\dot\Omega \ll \Omega^2$. It is a rather good approximation for the early stages of the inspiral, as we shall see.

The last expression we wrote for the amplitude of the emission reintroduces factors of $c$, and it is expressed in terms of the emission frequency $f_{\rm gw} = \omega_{\rm gw}/2\pi = \Omega/\pi$ and of the *chirp mass*

$$\mathcal{M}_c = \frac{(m_1m_2)^{3/5}}{(m_1+m_2)^{1/5}} = \nu^{3/5}M \qquad \text{where} \qquad \nu = \frac{\mu}{M}. \tag{1.68}$$

The parameter $\nu$ is called the *symmetric mass ratio*, and it takes values from $\nu \to 0$ (one mass vanishing) to $\nu = 1/4$ (equal masses).

### 1.2.1 Energy evolution

Combining equations (1.65) with the quadrupole emission formula (1.60) yields an expression for the emitted power

$$P = \frac{32}{5}\frac{c^5}{G}\left(\frac{G\mathcal{M}_c\omega_{\rm gw}}{2c^3}\right)^{10/3}. \tag{1.69}$$

This allows us to describe the evolution of the orbit in this quasi-adiabatic context: the total energy of the binary system, by the virial theorem combined with Kepler's third law $\Omega^2R^3 = GM$, reads

$$E = -\frac{Gm_1m_2}{2R} = -\frac{Gm_1m_2}{2}\frac{\Omega^{2/3}}{G^{1/3}M^{1/3}} = -\left(\frac{G^2\mathcal{M}_c^5\omega_{\rm gw}^2}{32}\right)^{1/3}. \tag{1.70}$$

With these two expressions at hand, we can impose $P = -\dot E$, which we can write in the form $\dot\omega_{\rm gw} \propto \omega^n$ for some $n$: since $\dot E \propto \omega_{\rm gw}^{-1/3}\dot\omega_{\rm gw}$, $n$ will be $11/3$, and the specific expression will look like

$$\dot\omega_{\rm gw} = \frac{12}{5}\sqrt[3]{2}\left(\frac{G\mathcal{M}_c}{c^3}\right)^{5/3}\omega_{\rm gw}^{11/3}. \tag{1.71}$$

This equation is in the form $\omega^{-11/3}\,\mathrm{d}\omega = K\,\mathrm{d}t$; integrating from point 0 to point 1 we find $3K(t_1-t_0)/8 = \omega_0^{-8/3} - \omega_1^{-8/3}$; if we treat this as an initial value problem we can see that as $t_1$ increases there is only a finite "budget" on the right-hand side: when $3K(t_1-t_0)/8 = \omega_0^{-8/3}$, $\omega_1^{-8/3}$ must go to 0 so $\omega_1$ must diverge. The divergence itself is unphysical since, besides corresponding to a breakdown of some of our approximations,

it comes about when considering point masses, while compact objects have a finite size and at some point collide.

Nevertheless, the moment of divergence happens close enough to the actual merger of the objects, therefore it is useful to call it $t_c$ and to define a time coordinate as $\tau = t_c - t$. In terms of this, the solution to the equation reads

$$f_{\mathrm{gw}}(t) = \frac{\omega_{\mathrm{gw}}(t)}{2\pi} = \frac{1}{2\pi}\left(\frac{3K\tau}{8}\right)^{-3/8} = \frac{1}{\pi}\left(\frac{5}{256\tau}\right)^{3/8}\left(\frac{G\mathcal{M}_c}{c^3}\right)^{-5/8}. \tag{1.72}$$

This expression for $f_{\mathrm{gw}}$ can be substituted into the gravitational waveforms (1.65); we must also update the phase term, since $2\Omega t$ is the phase for a uniformly circular orbit: we will want to use integrate the angular velocity, to find

$$\Phi(t) = \int^t \omega_{\mathrm{gw}}(\widetilde{t})\,\mathrm{d}\widetilde{t} = 2\int^t \Omega(\widetilde{t})\,\mathrm{d}\widetilde{t} \tag{1.73}$$

$$= -2\left(\frac{5G\mathcal{M}_c}{c^3}\right)^{-5/8}\tau^{-5/8} + \mathrm{const}\,. \tag{1.74}$$

In terms of this, the amplitude in the two polarizations will read [Mag07, eqs. 4.31–32]

$$h_+(t) = \frac{1}{r}\left(\frac{G\mathcal{M}_c}{c^2}\right)^{5/4}\left(\frac{5}{c\tau}\right)^{1/4}\left(\frac{1+\cos^2\iota}{2}\right)\cos\Phi(t) \tag{1.75}$$

$$h_\times(t) = \frac{1}{r}\left(\frac{G\mathcal{M}_c}{c^2}\right)^{5/4}\left(\frac{5}{c\tau}\right)^{1/4}\cos\iota\sin\Phi(t)\,. \tag{1.76}$$

As we will discuss, it is convenient to have expressions for the Fourier transforms of these. An analytic computation of the integrals is intractable, but we can make use of a technique known as stationary phase approximation (SPA), which is discussed in detail in appendix A. The final expressions for the Fourier-domain waveforms are equations (A.18).

### 1.2.2 Spherical harmonics decomposition

The waveform we described so far comes from the quadrupole of the mass distribution, but higher-order moments also generate GWs. In general, in units where $G = c = 1$, one can write the waveform as [Aji+11, eq. II.6]

$$h_+ + ih_\times = \frac{M}{r}\sum_{l=2}^{\infty}\sum_{m=-\ell}^{\ell} h_{\ell m}(t)\,{}_{(-2)}Y_{\ell m}(\iota,\varphi)\,, \tag{1.77}$$

where the functions ${}_{-2}Y_{\ell m}$ are called *spin-weighted spherical harmonics*, a generalization of the regular spherical harmonics. The only assumption in this expression is the $1/r$ scaling of the amplitude of the waveform. The expansion parameters $h_{\ell m}$ are time-dependent and complex, and the spin-weighted harmonics depend on the orientation of the basis we choose (see Sky position and polarization).

In the case of aligned spins, the harmonics which give the largest contribution are the $(\ell = 2, m = 2)$ and $(\ell = 2, m = -2)$ ones, which are conjugates of each other. The algorithm machine learning gravitational waves from binary neutron stars (`mlgw_bns`) currently only reconstructs this mode — considering more modes will be needed in order to treat the effects of precession, which happens when the spins are not aligned.

### 1.2.3 Parameters for a compact binary coalescence (CBC) waveform

The waveform we wrote explicitly depends on:

1. the chirp mass $\mathcal{M}_c$, defined in (1.68);
2. the distance $r$, which in a more general cosmological context should be replaced by the *luminosity distance* $D_L$;
3. the inclination $\iota$, which is the angle between the observation direction and the total angular momentum of the system.

These are only some of the parameters which must be considered when discussing a CBC. We will now discuss the full set of parameters which can be used to fully describe the binary system generating the waveform [BGB21, eq. 21].

**Arrival time and initial phase** The time at which any given waveform arrives at Earth is arbitrary, as is the global phase of the waveform. In practice, one will typically analyze the output from a detector in batches, and in each of these the analysis will be performed in Fourier space. Therefore, it is relevant to see how the Fourier-domain waveform responds to a time and phase shift: what is the transform of $f(t-t_0)e^{i\varphi_0}$? It comes out to be

$$\int_{\mathbb{R}} f(t-t_0)e^{i\varphi_0}e^{i\omega t}\,\mathrm{d}t = e^{i\omega t_0+i\varphi_0}\underbrace{\int_{\mathbb{R}} f(t-t_0)e^{i\omega(t-t_0)}\,\mathrm{d}(t-t_0)}_{\tilde{f}(\omega)}, \tag{1.78}$$

so a time shift corresponds to the addition of a linear term to the phase, while the phase can be directly moved from the time to the frequency domain.

**Sky position and polarization** The wave will be coming from a sky position, which we can describe with respect to a given coordinate system through two angles; it is convenient to use right ascension $\alpha$ and declination $\delta$. Also, the polarization of the gravitational waves can have an arbitrary angle with respect to the observation direction: we denote this angle as $\psi$.

In more technical terms, a general waveform arriving our detector will be a superposition of GWs in the TT gauge (1.27) with varying $\alpha$, $\delta$, $\psi$ as well as different frequencies; we can write it as [Mag07, eq. 1.58]

$$h_{ab}(t,\vec{x}) = \sum_{\mathrm{pol}=+,\times}\int_{-\infty}^{\infty}\mathrm{d}f\int \mathrm{d}^2\hat{n}\,\tilde{h}_{\mathrm{pol}}(f,\hat{n})e^{\mathrm{pol}}_{ab}(\hat{n},\psi)e^{-2\pi i f(t-\hat{n}\cdot\vec{x}/c)}, \tag{1.79}$$

where $\hat{n}=\hat{n}(\alpha,\delta)$ is the vector describing the propagation direction of a specific component — in the context of a CBC the distribution of the $e^{\mathrm{pol}}_{ab}(\hat{n},\psi)$ will include a Dirac delta centered on the position of the source in the sky, while this more general expression can be useful, for example, in the context of a stochastic gravitational wave background (SGWB).

The tensors $e_{ab}$ are basis tensors: in a frame where $\hat{n}$ and $\psi$ are chosen such that the $+$ polarization stretches the $x$ and $y$ axes, they read

$$e^{+}_{ij}(\hat{n}=\hat{z},\psi=0) = u_iu_j - v_iv_j = \begin{bmatrix}1&0&0\\0&-1&0\\0&0&0\end{bmatrix} \tag{1.80}$$

$$e^{\times}_{ij}(\hat{n} = \hat{z}, \psi = 0) = u_i v_j + v_i u_j = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \tag{1.81}$$

where $u$ and $v$ are two unit vectors defining the polarization direction; they are always chosen to be orthogonal, so fixing the angle $\psi$ is enough to determine them both once we give $\hat{n}$.

The aforementioned expansion glosses over a technical issue: the three-dimensional Fourier transform of the TT gauge waveform is in the form $h_{ij}(x) \sim \int \mathrm{d}^3k\, A_{ij}(k) e^{ikx}$, so in order to write it like we did we need to define

$$\frac{f^2}{c^3} A_{ij}(f, \hat{n}) = \sum_{\mathrm{pol}=+,\times} \widetilde{h}_{\mathrm{pol}}(f, \hat{n}) e^{\mathrm{pol}}_{ij}(\hat{n}), \tag{1.82}$$

where we reparametrized the wavevector $\vec{k}$ through frequency $f$ and direction $\hat{n}$, and exploited the decomposition $\mathrm{d}^3k = k^2\, \mathrm{d}k\, \mathrm{d}^2\hat{n}$.

**Masses and mass ratio** The waveform to the lowest order we described only depends on the chirp mass $\mathcal{M}_c$, but as we will see later at higher order there appears a dependence on the ratio of the two masses: $q = m_1/m_2$,. It is customary to fix one of the two masses to always be the largest, therefore constraining $q \lesseqgtr 1$; we choose the convention $q \geq 1$.

A possible parametrization for the two masses is $(\mathcal{M}_c, q)$, another is $(M, q)$: these are equivalent, but in the following work we will typically use the latter.

Fix!

At this point we can notice an interesting scaling property: in these units lengths, times, energies, masses and momenta have the same dimensions. What happens, then, if we rescale all these quantities by the same amount? This means we are mapping $M \to \alpha M$ and $r \to \alpha r$ in the previous expression, so we can see that the amplitude $h$ is preserved. This is consistent with a time rescaling of the time $t \to \alpha t$, as can be seen in equation (1.75), and with a rescaling of the frequency $f \to f/\alpha$, as can be seen in equation (A.18). The result also holds at higher orders than the quadrupole, as we will see by the parametrization of the waveform in section 1.4.1.

**Spin** Compact objects can spin, and the spin of each of the two is a vector with three independent components. These vectors $\vec{S}_i$ have units of angular momentum, but they are typically rescaled as

$$\vec{\chi}_i = \frac{c\vec{S}_i}{Gm_i} \in [-1, 1] \text{ in magnitude}. \tag{1.83}$$

The constraint of $|\chi_i| < 1$ can be approached by realistic models of BHs, while realistic models of NSs typically are constrained to spin much less, at the most $|\chi_i| \lesssim 0.7$ [LL11]. This constraint is hard to experimentally verify since one is faced with the degeneracy between the components $\chi_{z,i}$ aligned with the direction of the angular momentum and the mass ratio $q$; in the analysis of the merger GW170807 two sets of prior distributions for the spins were used, $|\chi| < 0.05$ and $|\chi| < 0.89$, for this reason [Abb+17]. We have theoretical reasons to believe that the low-spins priors might correspond to a

more physically meaningful scenario, but the degeneracy means that the high-spin case is not experimentally excluded, so we must still consider it a possibility.

The qualitative effect of spin is due to both spin-orbit and spin-spin interactions.

In the simplest spin-aligned case, $\vec{\chi}_i \propto \vec{L}$, the effect is to make the interaction less attractive (when the spins are aligned, $\vec{\chi}_1 \cdot \vec{\chi}_2 > 0$) or to make it more attractive in the alternate case.

The dynamics when the spins are not aligned are significantly more complicated, since the spins precess and give rise to modulations of the amplitude. Because of this, often models restrict themselves to the aligned-spin scenario.

**Tidal polarizability of neutron stars** If our compact objects are not black holes, they might be able to deform. The astrophysically motivated scenario we think of is that of neutron stars, but the following characterization could also apply to any extended compact objects. For concreteness' sake, we will refer to NSs.

This is a complicated process, but we can try to capture its most significant part by limiting ourselves to the quadrupole order of the deformation of the star. The discussion of this effect is clearest in the Newtonian context, but it can be also formalized in general relativity (GR) [Mag18, section 14.4.1].

Tidal effects are described by the tidal tensor $\mathcal{E}_{ij} = -\partial_i \partial_j U$, the traceless[8] Hessian of the Newtonian potential $U$. In the weak-field relativistic case, this corresponds to part of the Riemann tensor: $\mathcal{E}_{ij} = c^2 R_{0i0j}$.

We can describe the effects on the deformed star of such a tidal stress by looking at its quadrupole moment $Q_{ij}$ (see equation (1.56)). To linear order and neglecting any time dependence, we will have the relation

$$\mathcal{E}_{ij} = -\lambda Q_{ij} \,. \tag{1.84}$$

This parameter $\lambda$ describes the deformability of the star; if it is smaller the star deforms less in response to tidal perturbations. In practice the parameter used is not $\lambda$ itself but one of two common ways to rescale it: the $\ell = 2$ (quadrupole) **Love number**

$$k_2 = \frac{3}{2} \frac{G\lambda}{R^5} \,, \tag{1.85}$$

where $R$ is the radius of the deformed neutron star (NS), or the **tidal deformability**

$$\Lambda = \frac{2}{3} k_2 \left( \frac{Rc^2}{Gm} \right)^5 \,, \tag{1.86}$$

where $m$ is the mass of the deformed NS. The value of these parameters depends on the specific equation of state used in the model; $k_2$ is typically of the order of $10^{-1}$, and in the expression for $\Lambda$ we have a fifth power of the *compactness* $\sigma = Rc^2/Gm$ of the NS — this is expected to be a small number, but larger than 3,[9] so its fifth power will be of order $10^{3 \div 4}$. This means that the deformability $\Lambda$ will typically be of the order $10^{2 \div 3}$.

[8] This tidal tensor describes the effect of one star's gravitational field on the other, so its source is not where we compute it, which is a convoluted way of saying that it should be taken to be a solution of $\nabla^2 U = 4\pi\rho$ in vacuum: therefore, $\nabla^2 U = -\operatorname{Tr}\left[\mathcal{E}_{ij}\right] = 0$.

[9] A limit on the maximum redshift of radiation from the surface of a NS was calculated by Lindblom [Lin84] under the assumptions of causality and stability for the nuclear matter, which is equivalent to a compactness limit of $\sigma \geq 2.83$ [LP07].

The Love number is useful since it is an expansion parameter for the gravitational field of the deformed NS. This field has contributions from both its own quadrupole moment, which is due to the tidal deformation from the other star, and from the gravitational field of the other star itself, which we can expand in terms of the tidal field around the center of the deformed star: $U_{\rm ext} \approx -\mathcal{E}_{ij} x_i x_j / 2$.

The gravitational field of the deformed star also changes under the effect of the deformation: including the first term in its expansion, which depends on the quadrupole moment, the potential reads

$$U_{\rm int} \approx \frac{Gm}{r} + \frac{3G}{2r^3} \frac{x_i}{r} \frac{x_j}{r} Q_{ij} \,, \tag{1.87}$$

where $r = |x|$ is the radius from the center of the deformed star.

We can then see that under the assumption of linear dependence of the quadrupole on the tidal tensor the two potentials can be added (since we are still in the realm of linear theory):[10]

$$U \approx \frac{Gm}{r} - \frac{1}{2} \mathcal{E}_{ij} x_i x_j \left[ 1 + 2k_2 (R/r)^5 \right] . \tag{1.88}$$

This is the starting point for the relativistic generalization of the result as well, where $k_2$ is an expansion parameter for a metric component.

The meaning of $\Lambda$ is less immediate — the first correction to the phasing of the waveform which appears is[11] [Mag18, eq. 14.231; FH08]

$$\Delta\Psi^{\rm tidal}_{\rm 5PN} = -\frac{117}{256} \frac{m^2}{m_1 m_2} \widetilde{\Lambda} \left( \frac{v}{c} \right)^5 \tag{1.89}$$

$$\widetilde{\Lambda} = \frac{16}{13} \frac{(m_1 + 12m_2) m_1^4 \Lambda_1 + (m_2 + 12m_1) m_2^4 \Lambda_2}{(m_1 + m_2)^5} \,. \tag{1.90}$$

The parameter $\widetilde{\Lambda}$, which depends on the tidal deformabilities of the two stars $\Lambda_{1,2}$, is called the *reduced tidal parameter*.

One would expect to see $\Lambda = 0$ for black holes, and this is true in the nonspinning case; on the other hand, Kerr black holes can exhibit small but nonvanishing deformability. Le Tiec and Casals [LC21] recently showed that, for example, BHs with dimensionless spins of the order $\chi \sim 0.1$ can exhibit Love numbers on the order of $k_\ell \sim 2 \times 10^{-3}$, around two orders of magnitude less than NS values. The effect this will have on GW emission is yet to be determined, but one can suspect it will be small.

**Eccentricity** A general binary system will orbit in an ellipse with eccentricity $e \in [0, 1)$; the two semiaxes of the ellipse read [Mag07, eqs. 4.51]

$$a = \frac{R}{1 - e^2} \qquad \text{and} \qquad b = \frac{R}{\sqrt{1 - e^2}} \,. \tag{1.91}$$

[10] A technical fact to remember here is that the expansions do not naturally match: the expansion of the internal potential is in orders of $1/r$, while the expansion of the external one is in orders of $r$.

[11] The meaning of the PN expansion will be discussed in section 1.4.1 — for now, one can think of it simply as an expansion in orders of $v/c$.

It can be shown that, for a Keplerian orbit, the semimajor axis $a$ and the eccentricity $e$ depend on the total energy $E$ and angular momentum $L$ as [Mag07, eqs. 4.50, 4.53]

$$e = \sqrt{1 + \frac{2EL^2}{G^2 m^2 \mu^3}} \qquad \text{and} \qquad a = \frac{Gm\mu}{2|E|} \,. \tag{1.92}$$

In order to fully describe the GW emission from a binary system we must then also account for eccentricity; this parameter can also change as time progresses, and in order to describe this process we need a second evolution equation. This can be found thanks to the quadrupole angular momentum emission formula, which is analogous to the energy emission one (1.60) [Mag07, eq. 3.99]:[12]

$$\frac{\mathrm{d}L^i}{\mathrm{d}t} = -\frac{2G}{5c^5}\epsilon^{ikl}\left\langle \ddot{M}_{ka}\dddot{M}_{la}\right\rangle \,. \tag{1.93}$$

This allows us to make a system of two equations, $\mathrm{d}E/\mathrm{d}t$ and $\mathrm{d}L/\mathrm{d}t$, which we can reparametrize as equations for $\dot{a}$ and $\dot{e}$ [Mag07, eqs. 4.116–17]:

$$\dot{a} = -\frac{64}{5}\frac{G^3\mu m^2}{c^5 a^3}\frac{1}{(1-e^2)^{7/2}}\left(1 + \frac{73}{24}e^2 + \frac{37}{96}e^4\right) \tag{1.94}$$

$$\dot{e} = -\frac{304}{15}\frac{G^3\mu m^2}{c^5 a^4}\frac{e}{(1-e^2)^{5/2}}\left(1 + \frac{121}{304}e^2\right) , \tag{1.95}$$

which can be (nontrivially) analytically solved, yielding the result shown in figure 1.2.

It can be clearly seen that eccentricity will decrease as the semimajor axis shrinks due to GW emission. Typical astrophysical binaries need to shrink by several orders of magnitude before merging — the notorious pulsar in a binary system detected by Hulse and Taylor [HT75] has a semimajor axis of $a \approx 2 \times 10^9\,\mathrm{m}$ and an eccentricity of $e \approx 0.617$, while the frequency of the GWs it emits is $f_{\mathrm{gw}} = 2/P \approx 7 \times 10^{-5}\,\mathrm{Hz}$ [TW82].

If we were to wait until this reached a frequency range where next generation ground-based detectors might hope to detect it, $f \sim 10\,\mathrm{Hz}$, we would expect to see (using Kepler's third law $a^3 f^2 = \mathrm{const}$) a reduction of semimajor axis by a factor $\sim 2700$: it would go off-scale to the left in figure 1.2, and we would expect to basically have $e \approx 0$. One can then see that this result holds for all but the most extreme initial values of $e$.

This is the reason why eccentricity is often not considered in GW modelling, while it is of crucial importance in astrophysical population studies, since it has a large effect on the emitted power (enhancing it) in the early stages of the inspiral.

Eccentric binaries are by no means excluded by this line of argument; there are astrophysical contexts in which they might be generated with high eccentricities combined with already small radii, which would allow $e > 0$ to be detectable even in the laser interferometric gravitational-wave observatory (LIGO)-Virgo band. Eccentric binary GW models do exist [Fav+21], and ignoring eccentricities may lead to bias in parameter estimation.[13]

[12] The angular momentum emitted in this expression is considered to be both spin and orbital angular momentum.

[13] It is interesting to note that these models reach into "relativistic territory", and therefore must also include effects such as periastron precession — a good resource on these, based on Favata [Fav11], is the https://www.soundsofspacetime.org/elliptical-binaries.html website, which also provides audio renditions of the waveforms.

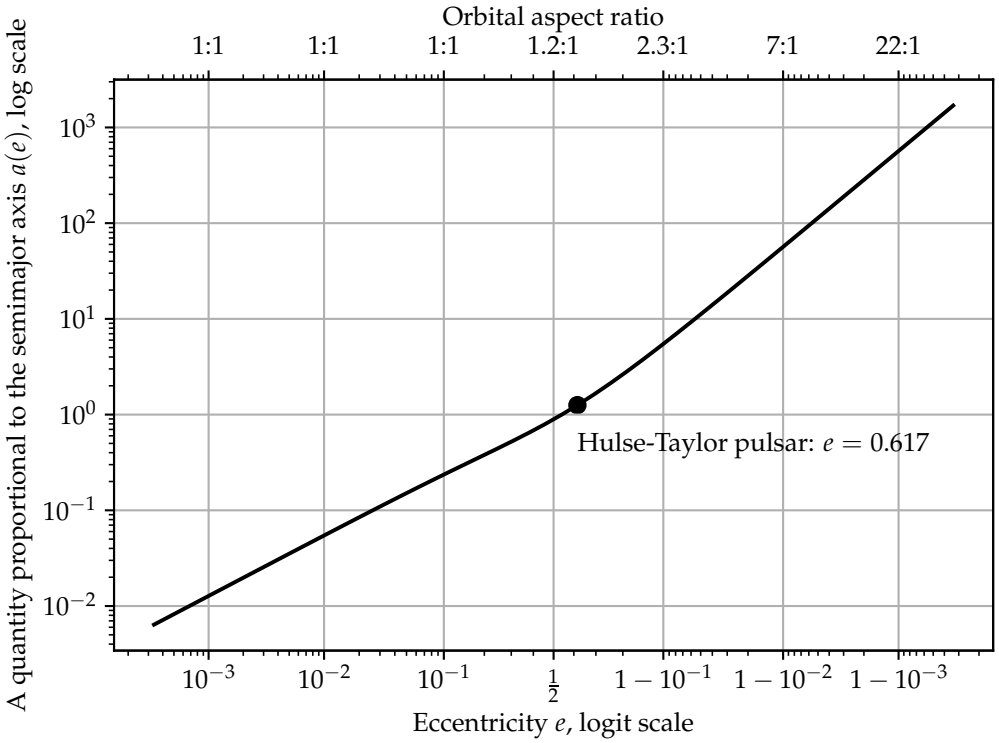


Figure 1.2: We show the behaviour of the analytic solution for the semimajor axis $a(e)$ [Mag07, eq. 4.126]; the $x$ axis is on a logit scale, meaning that it is proportional to $\log\big(e/(1-e)\big)$. The quantity on the $y$ axis is not $a(e)$, but $a$ is proportional to it: ratios in the form $a(e_1)/a(e_2)$ are correctly represented in this graph.

**Summary** The parameter vector $\vec{\theta}$ describing a binary system can be divided into internal and external parameters as

$$\vec{\theta} = (\underbrace{q, \vec{\chi}_1, \vec{\chi}_2, \Lambda_1, \Lambda_2}_{\theta_{\text{int}}}, \underbrace{M, D_L, \iota, \alpha, \delta, \psi, t_0, \phi_0}_{\theta_{\text{ext}}}) \,, \tag{1.96}$$

where the dependence of the waveform on the external parameters is well-understood and easy to analytically calculate, while the dependence on the internal parameters (barring $M$) is complicated.

The model developed in this work is built to reconstruct the dependence on $q$, $\Lambda_1$, $\Lambda_2$ and the aligned-spin components $\chi_{1,z}$ and $\chi_{2,z}$.

### 1.2.4 Natural units and simplifications

Restricting ourselves to the $(\ell = 2, m = 2)$ mode is equivalent to writing the Fourier-domain waveform as

$$\widetilde{h}_+(f) = \frac{M}{r} \frac{1 + \cos^2 \iota}{2} \, \widetilde{h}(f) \tag{1.97}$$

$$\widetilde{h}_\times(f) = \frac{M}{r} \cos \iota \, \widetilde{h}(f) \,. \tag{1.98}$$

Also, we work in natural units: we treat the waveform as a function of $Mf$, which is

dimensionless if $G = c = 1$. We must be careful, however: the time-domain strain $h(t)$ is dimensionless, but its Fourier transform $\widetilde{h}(f)$ has the dimensions of a time, or a mass.

Let us rewrite the SPA, 0PN expression for the amplitude (the absolute value of equation A.18 for $\iota = 0$) to illustrate this point:

$$\frac{\left|\widetilde{h}_+(f)\right|}{M_s} = \frac{1}{\pi^{2/3}}\sqrt{\frac{5}{24}}(M_s f)^{-7/6}\frac{M_m}{r}\sqrt{\nu}\,, \tag{1.99}$$

where we use employ the relation $\mathcal{M}_c = M\nu^{3/5}$ (1.68), as well as *almost* using natural units: $M_m$ and $M_s$ denote the values of the total mass as expressed in terms of length (meters) or time (seconds), respectively $M_m = GM/c^2$ and $M_s = GM/c^3$.

Conventionally (as well as in `mlgw_bns`), the waveforms expressed in "natural units" are provided as

$$\widetilde{h}_{NU}(fM) = \frac{\widetilde{h}(f)}{M_s}\frac{r}{\nu M_m} \overset{\text{0PN}}{\sim} \frac{1}{\pi^{2/3}}\sqrt{\frac{5}{24}}(Mf)^{-7/6}\frac{1}{\sqrt{\nu}}\,. \tag{1.100}$$

The amplitude of these is $\nu$-dependent even in the Newtonian limit — the division by $\nu$ is convenient in the context of EOB calculations.

When reconstructing a waveform, `mlgw_bns` will first compute $\widetilde{h}_{NU}(fM)$, and then the International System of Units (SI)-units version as

$$\widetilde{h}(f) = \widetilde{h}_{NU}(fM)\frac{\mathtt{M}^2}{\mathtt{r}}\nu\underbrace{\left(\frac{GM_\odot}{c^3}\right)\left(\frac{GM_\odot}{c^2\mathrm{Mpc}}\right)}_{\approx 2.36\times 10^{-25}\,\mathrm{s}}\,, \tag{1.101}$$

where `M` is the numeric value of the total mass, expressed in solar masses $M_\odot$, and `r` is the numeric value of the distance, expressed in Mpc.

## 1.3 Interferometers and data analysis

Having seen how a gravitational waveform from a CBC might look to lowest order, we move to a discussion of the detection of these waveforms with interferometric techniques.

The response of any detector to a GW is a[14] scalar output, which will be in the form

$$h(t) = D_{ij}h_{ij}\,, \tag{1.102}$$

where $D_{ij}$ is known as the *detector tensor*. We want to apply this expression to the generic one for a gravitational wave (1.79), which we can simplify by

1. removing the integral in $\mathrm{d}^2\hat{n}$: in the case of a CBC this is an excellent approximation;

[14] Current interferometric detectors have a single scalar response because of their Michelson-Morley design with arms at 90°; planned detector such as the Einstein Telescope will exhibit multiple scalar inputs [Tea11, section 5.3.2]. The following analysis still applies, each scalar input can be treated analogously; having multiple (from one multi-output detector or from a network) is incredibly beneficial for the accuracy of measurements.

2. removing the dependence on $\vec{x}$: this is called the *short-arm* approximation, which is warranted by the fact that our detectors have arms with lengths $L \sim 3\,\mathrm{km}$ and are most sensitive for GW frequencies of $f \sim 100\,\mathrm{Hz}$, while the frequencies corresponding to $L$ are $f \sim 100\,\mathrm{kHz}$.[15]

This leads to the following expression for the observed signal:

$$h(t) = \sum_{\mathrm{pol}=+,\times} \underbrace{D^{ij} e_{ij}(\hat{n}, \psi)}_{F_{\mathrm{pol}}} \underbrace{\int_{-\infty}^{\infty} \mathrm{d}f\, \widetilde{h}_{\mathrm{pol}} e^{-2\pi i f t}}_{h_{\mathrm{pol}}(t)} \tag{1.103}$$

$$= F_+ h_+(t) + F_\times h_\times(t)\,. \tag{1.104}$$

The *detector pattern functions* $F_{+,\times}$ can be computed explicitly; for a Michelson-Morley interferometer they read

$$F_+ = \frac{1}{2}\left(1 + \cos^2\theta\right)\cos 2\phi \cos 2\psi - \cos\theta \sin 2\phi \sin 2\psi \tag{1.105}$$

$$F_\times = \frac{1}{2}\left(1 + \cos^2\theta\right)\cos 2\phi \sin 2\psi + \cos\theta \sin 2\phi \cos 2\psi\,, \tag{1.106}$$

where $\theta$, $\phi$ are the two angles describing the direction the GW is coming from in a frame aligned with the axes of the detector — they will depend on the sky position of the source ($\alpha$, $\delta$) as well as the orientation of the detector in space (which depends on well-known parameters such as its latitude, the orientation of the earth at each time and so on).

### 1.3.1 Matched filtering

We have seen how the GW signal will look to our detector: $h(t)$, but in practical experiments what we will measure will be in the form $s(t) = h(t) + n(t)$,[16] and typically the magnitude of the noise timeseries $n(t)$ will be much larger than the magnitude of the signal.

This poses an issue both for the detection of a signal and for the analysis of a signal which has been identified as such. The technique we will describe here, matched filtering, has applications in both branches of GW data analysis.

The idea is to define a *filter*, a map from the signal timeseries to $\mathbb{R}$, in such a way that its value is low if there is no signal, and it is high if there is a signal of a certain shape. A common choice because of its computational simplicity is a *linear* filter, written as

$$s(t) \to \hat{s} = \int \mathrm{d}t\, s(t) K(t) \tag{1.107}$$

[15] It is actually possible to not use this approximation, and in fact it is advisable not to: modern interferometers use power recycling techniques, which allow for the effective length of the arms to be much longer than their physical one. In fact, the optimal detection strategy for any given GW frequency is to have a detector whose arms are a quarter of the GW wavelength long [Mag07, eq. 9.33] — this balances the effect of the deformation due to the GW changing sign during the time of flight of any specific photon with the "stacking" effect of the photon taking a longer path through the deformed space.

[16] The "noise" described here is not actually what is measured: the output of the detector is not $h(t)$ but it is a linear function of it — even without accounting for the technical details of the measurement, the quantity measured is the intensity of the light at the dark fringe of the detector, not directly $h$. Nevertheless, if we know the transfer function of the detector we can refer the measured noise to an "effective noise" $n(t)$, which would be the noise we would need to add to $h(t)$ in order to see the signal we see. This effective noise is what we will call $n$ hereafter.

for some filter function $K(t)$, which we can select arbitrarily. How can we determine the best choice of $K$? We want to maximize the *distinguishability* between true signals and random noise, which we can quantify through the signal-to-noise ratio (SNR):[17]

$$\mathrm{SNR} = \frac{\mathbb{E}(\hat{s}|\text{presence of } h)}{\sqrt{\delta\hat{s}|\text{absence of } h}} = \frac{S}{N}\,, \tag{1.108}$$

where we compute the root of a variance for $N$ since if there is only noise we expect $\hat{s} = \int \mathrm{d}t\, n(t)K(t)$ to be a random variable.

In order to properly express this, let us discuss our assumptions about the statistical properties of the noise: the simplest noise we can characterize is

1. stationary: its statistical properties are unchanging in time. This is not true in real detectors, but if the variation is slow enough one can work with "local" properties, on the scale of hours or days.

2. zero-mean: $\langle n(t)\rangle = 0$; this can be enforced by subtracting an offset.

3. uncorrelated in Fourier space: this can be stated simultaneously with the definition of the variance of each Fourier component, which is expressed through the single sided **power spectral density (PSD)**[18] $\langle \tilde{n}^*(f)n(f)\rangle = \delta(f-f')S_n(f)/2$.

4. Gaussian: each Fourier component is normally distributed around zero, with a variance described by the PSD.

The PSD, as defined, has the dimension of an inverse frequency; since it describes a variance it is often useful to discuss its square root, the *spectral strain sensitivity*, or *amplitude spectral density* $\sqrt{S_n}$, with dimensions $1/\sqrt{\mathrm{Hz}}$.

With these assumptions, we can write the SNR, moving to Fourier space, as [Mag07, eq. 7.45]

$$\frac{S}{N} = \frac{\int_{\mathbb{R}} \mathrm{d}f\, \tilde{h}(f)\tilde{K}^*(f)}{\sqrt{\int_{\mathbb{R}} \mathrm{d}f\, (S_n(f)/2)\left|\tilde{K}(f)\right|^2}} = \frac{(u|h)}{\sqrt{(u|u)}}\,, \tag{1.110}$$

where we defined the **Wiener product** between two real-valued signals $a$ and $b$ as the Fourier-space expression [Fin92; Mag07, eq. 7.46]

$$(a|b) = 4\,\mathrm{Re}\int_0^\infty \mathrm{d}f\, \frac{\tilde{a}^*(f)b(f)}{S_n(f)}\,. \tag{1.111}$$

The reason for the presence of the real part is that we want this to match the previous expression, where we know the signal $S$ to be real-valued; the factor 4 is a combination of the factor of 2 in the definition of the PSD and the fact that we restrict the integral to

[17] Properly speaking, this quantifies the distinguishability only under the assumption of zero-mean noise, otherwise we could make it arbitrarily large by adding a constant to $s(t)$.

[18] The distinction between the single- and double-sided PSD depends on whether we want to use negative frequencies in the integral to recover the variance at each time or not:

$$\left\langle n^2(t)\right\rangle = \int_0^\infty \mathrm{d}f\, S_n^{\text{single-sided}}(f) = \int_{-\infty}^\infty \mathrm{d}f\, S_n^{\text{double-sided}}(f)\,. \tag{1.109}$$

Since the noise is real-valued, these two are simply related by $S_n^{\text{single-sided}} = S_n^{\text{double-sided}}/2$.

positive frequencies only, using the fact that the negative-frequency part gives the same contribution for real-valued signals.

We also defined the modified filter $u$, which is defined so that its Fourier transform reads $\widetilde{u}(f) = \widetilde{K}(f)S_n(f)/2$, which allows the expression with the Wiener product to match the previous one.

This expression can then be written as $S/N = (\hat{u}|h)$, where $\hat{u} = u/\sqrt{(u|u)}$ This is then maximized by $\hat{u}$ parallel to $h$ with respect to the metric defined by the Wiener product: $\hat{u} \propto h$ means that

$$\widetilde{K}(f) \propto \frac{\widetilde{h}(f)}{S_n(f)}\,. \tag{1.112}$$

In other words, the best way to find a signal buried in noise is to scale the Fourier-domain expression for the filter by the amplitude of the noise.

**Whitening** An alternative way to write the same expression is through the concept of *whitening*: if $S_n(f)$ is known, we can transform any signal $a(t)$ into

$$a_w(t) \qquad \text{such that} \qquad \widetilde{a}_w(f) = \frac{\widetilde{a}(f)}{\sqrt{S_n(f)/2}}\,. \tag{1.113}$$

In other words, we are mapping a signal into another signal where all the noise Fourier components are uniformly scaled: white noise.

In terms of the whitened signals, the Wiener product just reads

$$(a|b) = 2\,\mathrm{Re}\int_0^\infty \mathrm{d}f\, \widetilde{a}_w^*(f)\widetilde{b}_w(f)\,. \tag{1.114}$$

**Optimal SNR** The expression for the SNR, if we are using the optimal filter, is then (in terms of an arbitrary constant $C$):

$$\text{optimal SNR} = \frac{(Ch|h)}{\sqrt{(Ch|Ch)}} = \sqrt{(h|h)} = 4\int_0^\infty \mathrm{d}f\, \frac{\left|\widetilde{h}(f)\right|^2}{S_n(f)}\,. \tag{1.115}$$

Amplitude strain profiles, as well as Fourier transforms of signals, are often plotted with log-scales: in order to have an intuition for this quantity we can reframe it as

$$\text{optimal SNR} = \int_{-\infty}^{\infty} \mathrm{d}\log f\, \frac{h_c^2(f)}{h_n^2(f)} \tag{1.116}$$

Used $\mathrm{d}\log f = \mathrm{d}f\,/f$.

$$h_c(f) = 2f\left|\widetilde{h}(f)\right| \tag{1.117}$$

$$h_n(f) = \sqrt{fS_n(f)}\,, \tag{1.118}$$

where the quantities $h_c$ and $h_n$ are called the **characteristic strains** of signal and noise [MCB15, eqs. 17–19]. They are both dimensionless, the way this integral is expressed allows us to integrate "by eye": if we plot log-characteristic strain against log-frequency, the positive area between $h_c$ and $h_n$ will be proportional to the optimal SNR.

This is shown in figure 1.3, with specific reference to the event GW170817. We both show the PSDs computed from the data near the actual event and the two design

sensitivities for the current-generation advanced LIGO (aLIGO) and the planned, third generation detector Einstein telescope (ET). The optimal SNR for a signal with the parameters of GW170817, naïvely computed, is much larger than the actual value of the measured SNR. This is due to the inclination of the source, its position in the sky and the specific realization of the experimental noise. These all act to decrease the experimental SNR, and they also differ among the various detectors for each signal, since they are differently oriented.

These effects can be analytically accounted for in the analysis of a signal; in the making of figure 1.3 we simply divide the signal characteristic strain by a constant, selected in order to match the measured SNR.

The fact that integrating $(h_c/h_n)^2$ yields the SNR is represented in more detail in figure 1.4.

Alternatively, we can define an "amplitude signal strain spectral density":

$$\sqrt{S_h(f)} = 2\sqrt{f}\left|\widetilde{h}(f)\right|, \tag{1.119}$$

which is comparable to $\sqrt{S_n(f)}$, in the sense that it also has units of $\mathrm{Hz}^{-1/2}$.

**The (optimal) mismatch** The scalar product $(a|b)$ between waveforms allows us to define a distance between them which is meaningful in terms of their distinguishability in data analysis. Specifically, we use the fact that in any inner product space we can write the product as $(a|b) = \sqrt{(a|a)(b|b)}\cos\theta$, where $\theta$ can be interpreted as an angle between $a$ and $b$. Therefore, we can define the *fidelity*, or *mismatch* "distance":[19]

$$M(a,b) = 1 - \frac{(a|b)}{\sqrt{(a|a)(b|b)}} = 1 - \cos\theta\,. \tag{1.120}$$

This mismatch is computed among two waveforms with all of their parameters fixed; in practice, it is often useful to let the initial phase and coalescence time of the waveforms free to vary, and compute the *optimal mismatch*

$$\mathcal{F}(a,b) = 1 - \max_{\phi_0,t_0}\frac{(a|b(\phi_0,t_0))}{\sqrt{(a|a)(b|b)}}\,, \tag{1.121}$$

where either of the two waveforms is shifted in time and in phase.

In practice, this does not affect the norms of the waveforms in the denominator, but instead it means that the scalar product between $a = A_a(f)e^{i\phi_a(f)}$ and $b = A_b(f)e^{i\phi_b(f)}$ reads

$$\max_{\phi_0,t_0}(a|b(\phi_0,t_0)) = \max_{\phi_0,t_0} 4\,\mathrm{Re}\int_0^\infty \mathrm{d}f\,\frac{A_a(f)A_b(f)e^{i\phi_a - i\phi_b + 2\pi i f t_0 + \phi_0}}{S_n(f)} \tag{1.122}$$

$$= \max_{t_0} 4\left|\int_0^\infty \mathrm{d}f\,\frac{A_a(f)A_b(f)e^{i\phi_a - i\phi_b + 2\pi i f t_0}}{S_n(f)}\right|. \tag{1.123}$$

[19] This is not a distance in the mathematical sense: it does not satisfy the identity of indiscernibles principle since $M(a,Ka) = 0$ for any constant $K$, and it does not satisfy the triangle inequality $M(a,b) \leq M(a,c) + M(b,c)$ — think, for example, of $a$ and $c$ at right angles to one another (so $M(a,c) = 1$) and $b$ in the middle, at 45° from either, so $M(a,b) = M(b,c) = 1 - \sqrt{2}/2 \approx 0.3$. Then, $M(a,c) = 1 > M(a,b) + M(b,c) \approx 0.6$, in violation of the inequality.

This is not a problem for us: this mismatch is still a useful metric even if it is not a distance.

This relation also means that while the mismatch $M(a,b)$ could in principle be as large as 2 (for $a(f) = -b(f)$) the optimal mismatch is bounded by 1, since there will always be a phase choice to make $(a|b)$ positive, albeit small in magnitude.

### 1.3.2 Target fidelity

We defined the fidelity as a distance induced by the Wiener product, but we have yet to show that it actually measures what we need. We shall use it to evaluate the accuracy of our waveform templates against certain reference templates; this needs to satisfy two independent criteria [LOB08].

1. In a matched-filtering search for signals, they need to be accurate enough that we do not miss a large fraction of the signals — this is called *effectualness* [DNT11];

2. in the data analysis of a signal, they need to be accurate enough that they do not induce a significant bias in the estimation of parameters — this is called *accuracy*.

It turns out that fidelity is a useful metric for both of these, but the thresholds we need to set are chosen differently.

**Effectualness** In the matched-filtering context, we search for signals by comparing (with the Wiener product) the measured data against a bank of templates with varying parameters, and checking whether any of them surpass a certain SNR threshold.

The ineffectualness of our templates is one factor which makes it more likely for us to miss a signal; another source of error in this regard is given by the grid size of the template bank — if it is too sparse, the true parameters of the system might lie quite far from any grid point. Let us neglect the second of these aspects, since it is independent of the template-making efforts of this work, and focus on the first.

How does a certain template fidelity $\mathcal{F}$ relate to the fraction of signals we will miss because of the ineffectualness of our templates, supposing that our template bank has a vanishing grid size?

We already discussed the fact that if the true signal is $h$, then the optimal SNR is $\mathrm{SNR} = \sqrt{(h|h)} = (s|h)/\sqrt{(h|h)}$; if instead the signal is $h$ while our template includes $h'$, the SNR will be $\mathrm{SNR}' = (h|h')/\sqrt{(h'|h')} \approx (1-\mathcal{F})\mathrm{SNR}$. This also applies in the noisy case: having imprecise templates will lead to a decrease in the effective SNR.

For very loud signals this will not be an issue, but the distribution of the sources we detect is peaked at the edge of detectability; if we make the simplifying assumption that the distribution of sources is uniform in space and the volume scales as the luminosity distance cubed[20] we find that the fraction of sources we miss due to template ineffectualness is roughly $1-(1-\mathcal{F})^3$ [DNT11, eq. 18; Owe96, eq. 2.21].

A commonly used threshold by the LIGO-Virgo collaboration is to accept missing 10 % of the signals, therefore setting $\mathcal{F}_{\max} \approx 0.035$ [LOB08].

[20] This is not true in a cosmological context, i.e. at high redshift: the comoving volume scales as $d_L^3/(1+z)^3$. For BNSs detections have only ever been achieved so far with $z \lesssim 0.05$, meaning that the difference is of the order of 15 %, while for BBH the bound is $z \lesssim 1$, for which the calculation with $d_L^3$ alone is wrong by almost an order of magnitude. Specifically, the comoving volume is *smaller* than $\propto d_L^3$, which makes the bound for the ineffectualness of templates more forgiving in the high-redshift context.

**Accuracy** The condition needed for data analysis is stricter. We can quantify the bias given by our waveform in terms of the error in parameter estimation we have from the intrinsic detector noise [LOB08, sec. A; DNT11, sec. C]. If we allow for the bias due to waveform inaccuracy to be comparable to the one due to intrinsic statistical uncertainty (i.e. smaller than $1\sigma$) we find that in the Gaussian approximation, for a signal detected with a certain SNR and for which we want to estimate $D$ intrinsic parameters we will need our inaccuracy to be bounded by [Cha+17, appendix G; Gam+21]:[21]

$$\mathcal{F} \leq \frac{D}{2\mathrm{SNR}^2} \,. \tag{1.124}$$

As an example, GW170817 was detected with a combined SNR $\approx 32.4$ [Abb+17], meaning that the fidelity needed to analyze it is $\mathcal{F} \lesssim 5 \times 10^{-3}$ if we are considering five intrinsic parameters.

**Surrogates** The previous considerations regard the accuracy of a waveform compared to the true signal; however, `mlgw_bns` is a surrogate model: it does not model any physics itself, relying instead on a dataset generated with another model.

This means that the final accuracy which can be reached is bounded by the accuracy of the underlying model, and we must seek not to worsen it by a large margin.

Comparing the EOB model currently used in the training of `mlgw_bns`, `TEOBResumS`, to numerical relativity simulations we get maximal values for the mismatch of $\mathcal{F} \simeq 2.5 \times 10^{-3}$ [Nag+18, fig. 1], while the bulk of the distribution lies in the $\mathcal{F} \sim 10^{-4} \div 10^{-3}$ range. We shall discuss in section 3.6 how this compares to the reconstruction errors by `mlgw_bns`.

### 1.3.3 Parameter inference: studying the signal

Modern data analysis techniques for GW signals are Bayesian by necessity: we need to extract estimates for the parameters generating a signal of which we only have one measurement.

The main quantity we want to extract from our analysis is the **posterior probability density function** $\mathbb{P}(\vec{\theta}|s)$, where $\vec{\theta}$ is the parameter vector while $s$ represents the data from the detector. If we integrate it in a certain hyper-volume $\Omega$, we get $\int_\Omega \mathrm{d}^n\vec{\theta}\, \mathbb{P}(\vec{\theta}|s)$: the answer to the question "given the data we measured, what is the probability that the parameters of the system were contained in the region $\Omega$?"

The way we write these probabilities is a compactification of notation: after the "given" symbol we should also always ideally include all the other assumptions about the signal, the noise, and the way they are combined into $s = h + n$: in the context of GW data analysis, this also includes the choice of waveform approximant. We will leave this implicit, but we understand that the probabilities calculated will always be model-dependent.

A Bayesian approach starts by applying Bayes' theorem:

$$\underbrace{\mathbb{P}(\vec{\theta}|s)}_{\text{posterior}} = \underbrace{\frac{1}{\mathbb{P}(s)}}_{\text{evidence}} \underbrace{\mathbb{P}(s|\vec{\theta})}_{\text{likelihood}} \underbrace{\mathbb{P}(\vec{\theta})}_{\text{prior}} \,. \tag{1.125}$$

[21] It has been suggested [PH20, sec. III] that in practical applications this criterion is too strict, and that the biases in parameter estimation it suggests are an overestimate.

The normalization factor $\mathbb{P}(s)$ is called the *evidence*, and it can be expressed as

$$\mathbb{P}(s) = \int \mathbb{P}(s|\vec{\theta})\mathbb{P}(\theta)\,\mathrm{d}^n\theta\ , \tag{1.126}$$

since the posterior must be normalized to have unit integral in order to be a probability distribution.

The prior, which encodes our prior belief about the potential values of the parameters, is often chosen to be uninformative,[22] so that we do not introduce bias in the analysis. In certain cases it might be warranted to use "biased" priors: for example, the analysis for GW170817 offers results based on a low-spin and a high-spin prior [Abb+17]: the first is theoretically motivated and might be more meaningful, but the second is still allowed by the data.

The second ingredient is the likelihood $\mathbb{P}(s|\vec{\theta})$: this is where signal modelling comes in, since the likelihood needs to include the way the theoretical signal $h_\theta(t)$ depends on the parameters.

The posterior distribution $\mathbb{P}(\vec{\theta}|s)$ contains all the information we can gather about the parameters, and it can be explored without computing the evidence, since certain algorithms (such as Metropolis-Hastings (MH), algorithm 1) can explore unnormalized distributions.

However, this makes the assumption that the model — which is practically represented by the choice of parameters, likelihood and prior — is correct; as mentioned before we should really be writing $\mathbb{P}(s) = \mathbb{P}(s|\text{model})$. If one only shows the posterior, the results cannot be directly compared with ones corresponding to a different model. Therefore, the results of a full data analysis should include the posterior as well as the evidence [Ski06; Knu+15].

Specifically, if we have the evidence for a specific signal corresponding to two models, $\mathbb{P}(s|M_1)$ and $\mathbb{P}(s|M_2)$, we can compute the *Bayes factor*, which allows us to update our belief about the relative probabilities of the two models

$$\underbrace{\frac{\mathbb{P}(M_1|s)}{\mathbb{P}(M_2|s)}}_{\text{posterior odds}} = \underbrace{\frac{\mathbb{P}(s|M_1)}{\mathbb{P}(s|M_2)}}_{\text{Bayes factor}}\underbrace{\frac{\mathbb{P}(M_1)}{\mathbb{P}(M_2)}}_{\text{prior odds}}\ . \tag{1.127}$$

**Gravitational wave likelihoods** What is the probability of observing $s = h_\theta + n$ if we fix $\theta$? The theoretical signal $h_\theta$ can be computed and is thereafter fixed, likewise $s$, so this is just the probability of observing a certain realization of the noise: under the assumption of Gaussianity, it will read

$$\mathbb{P}(h_\theta + n|\vec{\theta}) \propto \exp\left(-\frac{1}{2}\int_\infty^\infty \mathrm{d}f\ \frac{|n(f)|^2}{S_n(f)/2}\right) \tag{1.128}$$

$$\propto \exp\left(-\frac{(n|n)}{2}\right) = \exp\left(-\frac{(s-h_\theta|s-h_\theta)}{2}\right) \tag{1.129}$$

[22] This does not necessarily mean "uniform": there is a method, known as Jeffrey's prior, which allows one to maximize the "ignorance" about a parameter once a likelihood is given, through what is basically an argument for reparametrization invariance. For example, an uninformative prior on the mean of a Gaussian is uniform, while an uninformative prior on its standard deviation is log-uniform.

$$\propto \exp\left((s|h_\theta) - \frac{(s|s)}{2} - \frac{(h_\theta|h_\theta)}{2}\right) \tag{1.130}$$

$$\propto \exp\left((s|h_\theta) - \frac{(h_\theta|h_\theta)}{2}\right). \tag{1.131}$$

This is a closed-form expression for our posterior probability density, so if computational power were no issue we would be done: the function $\mathbb{P}(\theta|s)$ includes all the possible information we could extract about the parameters from our observation. We can use *estimators* for single parameters and their variances in order to better understand what we have measured: these are functions which take the full posterior and return an estimate for a single quantity, such as the value for a specific parameter. These will always be a simplification of the true distribution, but they are very useful by virtue of being simple to understand.

We ask of these estimators that they be

1. consistent: they should converge to the true value as more data is included;
2. efficient: they should minimize the variance[23] of the estimate;
3. robust: they should not be very sensitive to small fluctuations in the posterior distribution.

A choice which satisfies these criteria quite well is the **Bayes estimator**, which estimates a parameter $\theta^i$ and the covariance matrix component $\Sigma_{ij}$ as

$$\hat{\theta}^i = \int \mathrm{d}^n\vec{\theta}\,\theta^i \mathbb{P}(\vec{\theta}|s) \tag{1.132}$$

$$\hat{\Sigma}_{ij} = \int \mathrm{d}^n\vec{\theta}\left(\theta^i - \hat{\theta}^i\right)\left(\theta^j - \hat{\theta}^j\right)\mathbb{P}(\vec{\theta}|s)\,. \tag{1.133}$$

Depending on the way the posterior distribution is calculated and parametrized, computing this may be impractical. Other choices include estimating the parameters by the values which maximize the likelihood or the posterior, and the covariance from the Hessian of the probability density function (PDF) around that point.

### 1.3.4 Posterior sampling

So, how do we compute the posterior distribution $\mathbb{P}(\vec{\theta}|s)$ in practice? We know how to evaluate it at a single point $\vec{\theta}$, so we could evaluate it on some sort of grid, say, sampling $N$ equally spaced points in a reasonable interval for each parameter. The curse of dimensionality makes this infeasible: if we have $n$ parameters we would need $N^n$ evaluations; even with a very small $N \sim 10$ this quickly becomes unreasonably large, since we typically have $n \sim 15$ parameters.

[23] Here it is important to be careful: this is not the variance of the parameter, an estimate of the error we assign to it based on our single experiment, but the variance of the *estimates* taken over the space of possible experimental results we could have gotten while keeping the value of the true parameter fixed to a certain value

**Monte Carlo Markov Chains** One solution to this issue is to stochastically sample $\mathbb{P}(\theta|s)$, taking steps in randomly determined directions using a rule which includes information about the probability distribution. If the rule is appropriately selected, the sequence of points touched by this random walk will approximate a set of samples for the true distribution. This is known as **Monte Carlo** sampling. A common way to construct an appropriate rule is to make it memory-less, so that each new point is determined only based on the probability distribution and the current point. This is known as a **Markov Chain**.

Several algorithms implement Monte Carlo markov chain (MCMC) sampling; a common choice is the Metropolis-Hastings one, which is shown as algorithm 1; the function $p(\theta)$ should be chosen to be proportional to $\mathbb{P}(\theta|s)$ (the normalization is irrelevant), and we need to provide a probability density function $g(\theta_{\text{new}}|\theta)$ such that samples from $g$ can be easily be drawn. A common choice for $g$ is a Gaussian distribution centered in $\theta$, which has the advantage of being symmetric: $g(\theta_{\text{new}}|\theta) = g(\theta|\theta_{\text{new}})$. Further, we need a randomly chosen initial point $\theta_0$.

**Algorithm 1** Metropolis-Hastings algorithm.

**Require:** $p(\theta)$, $g(\theta_{\text{new}}|\theta)$, $\theta_0$, $N_{\text{samples}}$
  $i \leftarrow 0$
  $S \leftarrow \emptyset$
  **while** $i < N_{\text{samples}}$ **do**
    $\theta_{\text{new}} \leftarrow$ sample from $g(\theta_{\text{new}}|\theta_i)$
    $\alpha \leftarrow p(\theta_{\text{new}})/p(\theta_i) \times g(\theta_i|\theta_{\text{new}})/g(\theta_{\text{new}}|\theta_i)$
    with probability $\min(\alpha, 1)$ accept
    **if** accept **then**
      $\theta_{i+1} \leftarrow \theta_{\text{new}}$
    **else**
      $\theta_{i+1} \leftarrow \theta_i$
    **end if**
    $S \leftarrow S \cup \theta_{i+1}$
    $i \leftarrow i + 1$
  **end while**
  **return** sample set $S$.

This random walk will eventually (for $N_{\text{samples}} \to \infty$) converge to the true distribution; there are several heuristics which can be used to check whether this is actually happening in a specific number of steps, such as measuring the autocorrelation of the chain, or running several chains starting at different points and checking that they "mix" appropriately.

**Nested sampling** This technique allows for the simultaneous calculation of the evidence $\mathbb{P}(s) = \int \mathscr{L}(s|\theta)\mathbb{P}(\theta)\,\mathrm{d}^n\theta$ as well as the likelihood [Ski06; SS06; BGB21, app. B; Bet11].

The idea of this algorithm is to define the *prior mass* variable

$$X(\lambda) = \int_{\mathbb{P}(s|\theta)>\lambda} \mathbb{P}(\theta)\,\mathrm{d}^n\theta\ , \tag{1.134}$$

which quantifies the "mass" in parameter space (with the prior distribution as "density") of the region in which the likelihood $\mathbb{P}(s|\theta)$ is larger than $\lambda$.

Since the prior must be normalized $X$ is bounded to lie in $[0, 1]$, and the evidence can be written as

$$\mathbb{P}(s) = \int_0^1 \mathbb{P}(s|\theta)\,\mathrm{d}X\,. \tag{1.135}$$

If we knew the values of the likelihood at certain points $X_i$ with corresponding likelihood values $L_i$, then we could approximate the evidence as

$$\mathbb{P}(s) \approx \frac{1}{2}\sum_{i=1}^{N}(X_{i-1} - X_{i+1})L_i\,. \tag{1.136}$$

The practical problem in the evaluation of this integral in the high-dimensional context of parameter estimation is the fact that the region where the likelihood is high is typically very small in terms of prior mass; this is especially severe in the most desirable scenarios, where we are learning a great deal of new information compared to the non-informative prior, i.e. constraining our parameters very well [Ski06, sec. 4].

Ideally, we'd want to sample with a uniform distribution in $\log X$; the nested sampling method allows us to do so stochastically, with the procedure outlined as algorithm 2.

The way the points are selected forces them to be distributed close to uniformly in $\log X$, which means it is a good approximation to just set $\log X_i = -i/N$ for the samples. Alternatively, one could sample the prior distribution to get a more accurate answer.

**Algorithm 2** Nested sampling algorithm.

**Require:** set of $N$ points $\theta_j$ sampled from the prior, likelihood function $L(\theta)$
  evidence $Z \leftarrow 0$
  $i \leftarrow 0$
  prior mass sample $X_0 \leftarrow 0$
  **while** stopping condition not reached **do**
    $X_i \leftarrow \exp(-i/N)$
    $L_i \leftarrow \min(L(\theta_j))$
    $Z \leftarrow Z + L_i(X_{i-1} - X_{i+1})/2$
    replace the $\theta_j$ corresponding to $L_i$ with a new value,
    drawn from the prior restricted to $[L(\theta) > L_i]$
    $i \leftarrow i + 1$
  **end while**
  $Z \leftarrow Z + (X_{\text{last}}/N)\sum_j L(\theta_j)$
  **return** evidence $Z$
  **return** all computed samples $\theta_k$, weighted by $(X_{k-1} - X_{k+1})/2 \times L(\theta_k)/Z$.

The stopping condition in algorithm 2 must be defined so that it detects the point at which all the $N$ points are lying very close to the maximum of the likelihood; this is typically done by setting a lower threshold for the likelihood improvement at each step.

Besides estimating the evidence, this algorithm provides us with a sampling of the posterior distribution as long as we are careful to weigh them appropriately, as outlined in the last line of algorithm 2.

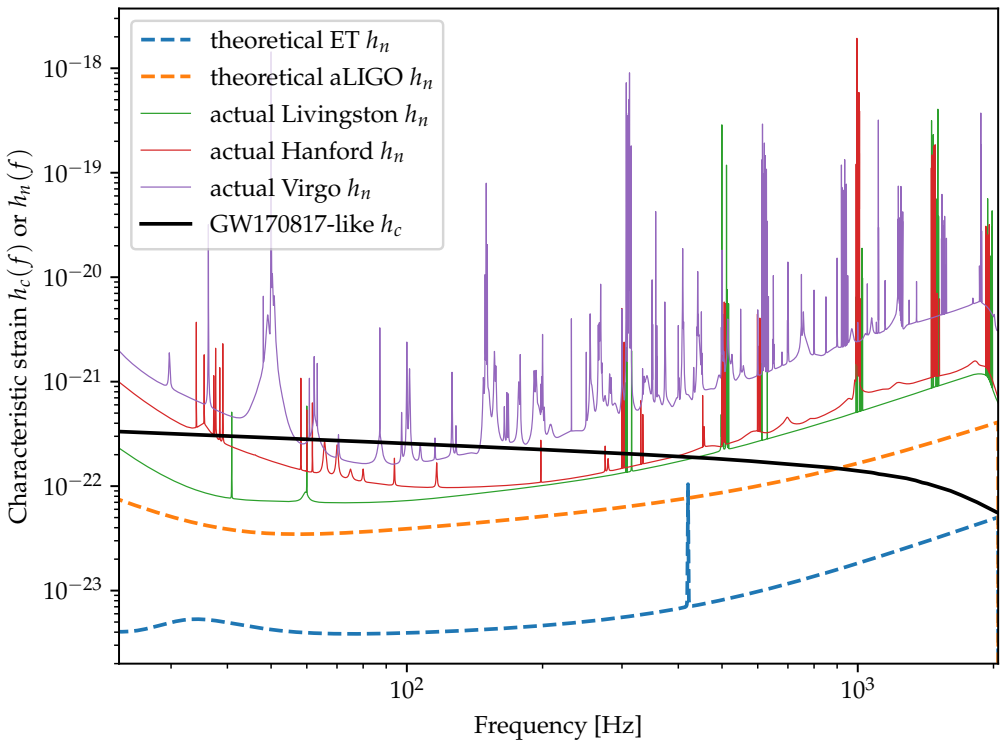


Figure 1.3: We show several comparable quantities: the characteristic noise strains corresponding to the estimated PSDs of aLIGO and ET, the ones corresponding to the actual PSDs in the three detectors at the time of the detection of GW170817, and the characteristic signal strain corresponding to a theoretical waveform calculated with GW170817 best-fit parameters. Comparing this single waveform to three different detectors' PSDs is not completely warranted, since their different locations and orientations give rise to different detector patterns. The waveform shown is lowered by a factor 8 compared to the value it would have with perfect detector orientation and zero inclination. The value is chosen so that the Livingston SNR of $\sim 26.4$ is matched by the graph; the Hanford SNR was $\sim 18.8$ while this graph shows $\sim 10$; the Virgo SNR of $\sim 2$ is matched by the graph [Abb+17].

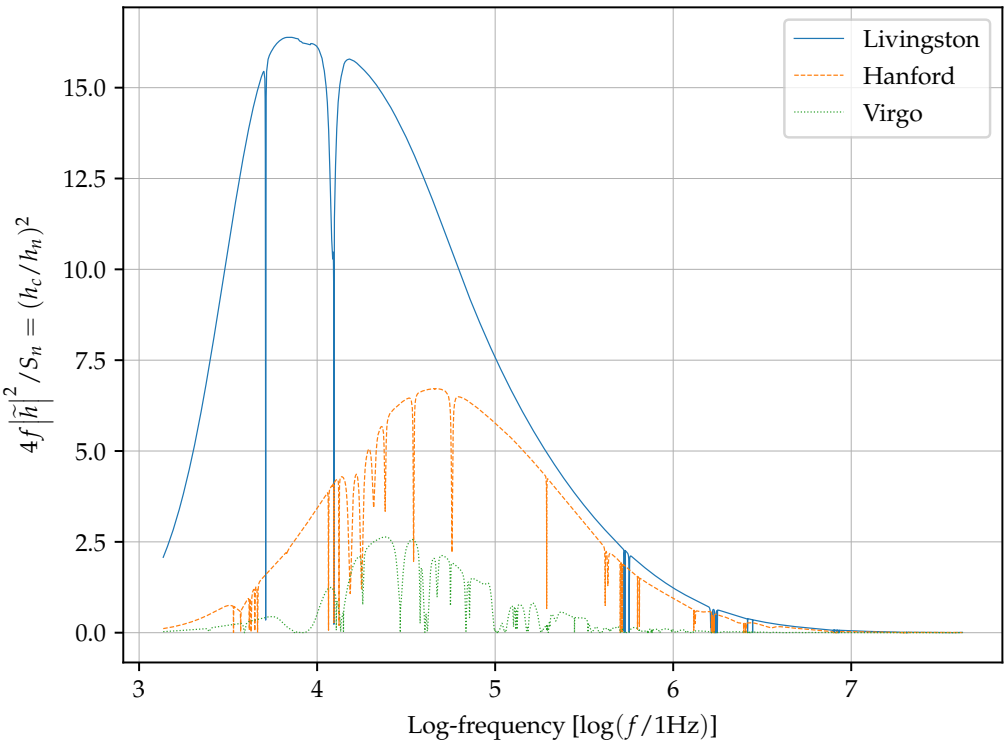


Figure 1.4: Integrand in the calculation of the optimal SNR for a GW170817-like signal with the three detectors of the LIGO-Virgo collaboration. The plotted data corresponds to figure 1.3: we show the square ratio of $h_c$ to the three actual PSDs.

## 1.4 Higher order waveforms

The treatment of GW production in linear gravity discussed in section 1.2 is useful but incomplete: as equation (1.61) shows, the systems emitting the most are *relativistic* ($v/c$ is large) and *compact* ($R_s/R$ is large). This has been verified in practice by the fact by the binary systems we detected: mostly BBHs, with some BNSs.

The quadrupole approximation, as well as the linearization of gravity, breaks down in the case we are interested in: what to do?

In the following sections we will discuss the main strategies[24] used to model CBC waveforms during all of their stages.

Qualitatively, a waveform from a CBC involves a long *inspiral* phase which terminates when the two compact bodies *merge*, and finally a *ringdown* phase in which the single body remaining exhibits damped vibrations.

### 1.4.1 Post-Newtonian

The assumptions which we made in order to derive the quadrupole formula for the emission of gravitational radiation (1.57) were to consider a non-compact, slow-moving source ($v \ll c$), on a flat background.

One might wish to expand the equations of motion in orders of $v/c$, and this is the idea behind the PN expansion, but there are technical difficulties associated with this. In terms of notation, a $n$PN expansion will be up to order $(v/c)^{2n}$ in this work.

**Technical difficulties** The strength of the gravitational field scales with the ratio of the Schwarzschild radius of an object to the distance between the objects, so as the system becomes relativistic in speed the gravity also becomes strong. The flat background metric needs to be substituted; a common choice is a Post-Minkowskian expansion, which is an expansion in orders of $R_s/r$, where $R_s$ is the Schwarzschild radius for the combined system. This expansion and the PN one have different domains of validity (one far from the source, one near it), so one needs to find an *overlap region* and match them.

GWs of low order source higher-order ones, as well as back-reacting and subtracting energy from the source. The latter of these effects comes about at the 2.5PN order: equating the emitted power estimated in equation (1.61) to the derivative of the energy (which by the virial theorem has the same magnitude as the kinetic energy) $\dot{E} = -Mv\dot{v}$ we find $\dot{v} \propto (v/c)^5$.

We can expand the retarded-time argument of the stress-energy tensor, $t - r/c$, around $r = 0$; this, however, is only valid if we are relatively near the source and breaks down if we try to extend the result to the "radiation zone" where our detector lies — we need to treat that zone separately. This expansion allows us to express the Dalambertian as $\Box \approx \left(1 + \mathcal{O}\left((v/c)^2\right)\right)\nabla^2$, and to turn the Einstein equations into Poisson equations

[24] One strategy which is not particularly relevant in this work but which is useful in other contexts is the gravitational self-force approach [Wal09], in which one works in orders of $m_1/m_2$, by making successive corrections to geodesic motion in a fixed background. This is not applicable to neutron stars, whose mass ratios are never extreme, but it is useful to discuss extreme mass ratio inspirals.

of the form

$$\triangle g_{\mu\nu}^{(n)} = \text{matter source} + \text{metric source from } g_{\mu\nu}^{(n-1)}\,. \tag{1.137}$$

After computing the $n$PN equations of motion we will need to compute the appropriate number of terms in the multipole expansion (1.48). This may also lead to divergencies: as we expand $\frac{1}{|\vec{x}-\vec{y}|}$ we get terms of type $(\vec{x}\cdot\vec{y})^\ell$ with large values for $\ell$, which rapidly diverge as we get further from the source. Specifically, the Poisson integrals, which solve $g(\vec{x}) = \nabla^2 f(\vec{x})$ with

$$f(\vec{x}) = -\frac{1}{4\pi}\int_{\mathbb{R}^3}\frac{\mathrm{d}^3 y}{|\vec{x}-\vec{y}|}g(\vec{y}) \tag{1.138}$$

will not always converge. Fortunately, this only reflects a limitation of the Poisson integral, and through analytic continuation one can recover a physical solution which respects the boundary conditions.

Solutions to these problems have been developed between the '80s and the early 2000s, and currently there exist general formalisms, such as the Blanchet-Damour one, which allow for high-order computations of PN waveforms [Bla14].

**PN waveforms** The typical notation used for PN waveforms expresses them [Mag07, sec. 5.6.1] as a function of the dimensionless frequency

$GM/r \sim v^2$ and $r\Omega \sim v$.

$$x = \left(\frac{GM\Omega}{c^3}\right)^{2/3} = \mathcal{O}\Big((v/c)^2\Big)\,, \tag{1.139}$$

the symmetric mass ratio $\nu = \mu/M$ and the dimensionless time

$$\Theta = \frac{\nu c^3}{5GM}(t_c - t)\,. \tag{1.140}$$

These are related by $x = \Theta^{-1/4}/4$ (a reframing of equation (1.72)) at Newtonian order, and we can also write the phase as

$$\phi - \phi_0 = -\frac{\Theta^{5/8}}{\nu} = -\frac{x^{-5/2}}{32\nu}\,. \tag{1.141}$$

Corrections to this expression are then written multiplicatively as a series in $x$ (or, really, in $x^{1/2}$, but the PN order is the exponent of $x$):

$$\phi - \phi_0 = -\frac{x^{5/2}}{32\nu}\Big(1 + c_{1\text{PN}}x + c_{1.5\text{PN}}x^{1.5} + \dots\Big)\,. \tag{1.142}$$

The error in the phase is therefore of order $x^{-1/2} \gg 1$ if we do not go beyond 2PN order: we need a high-order computation in order to recover an accurate expression for the phase. Current results include very high order waveforms, such as 5.5PN [BDG20].

**The PN order of tidal stresses** We can give a crude estimate, which turns out to be accurate, for the PN order at which the effects of quadrupolar tidal deformation come in [Mag07, pagg. 288–289].

The quadrupole moment can be estimated as $Q_{ij} \sim \epsilon m r^2$ where $\epsilon$ is the induced ellipticity of the NS while $m$, $r$ is its mass and radius. This ellipticity comes from the tidal effect of the companion; it can be estimated as the ratio of the tidal force to the restoring self-gravity of the star:

$$\epsilon \sim \frac{F_{\text{tidal}}}{F_{\text{self}}} \sim \frac{Gmr/d^3}{Gm/r^2} = \left(\frac{r}{d}\right)^3 , \tag{1.143}$$

where $d$ is the separation between the two stars, and we assume that their masses are the same.

The inter-body force induced by the quadrupolar deformation depends on the quadrupole moment like [Mag07, eqs. 5.219, 5.235]

$$F_k^{\text{induced}} \approx Q_{ij}\partial_i\partial_j\partial_k U^{\text{Newton}} \sim \frac{Q_{ij}}{m}\partial_i\partial_j F_k^{\text{Newton}} \sim \left(\frac{r}{d}\right)^3 \frac{r^2}{d^2} F_k^{\text{Newton}} , \tag{1.144}$$

so $F^{\text{induced}} \sim (r/d)^5 F^{\text{Newton}}$. This ratio can be reframed in terms of $v/c$: the radius $r$ of a star is proportional to its Schwarzschild radius $Gm/c^2$; while the separation of the two is given by the virial theorem: $mv^2 \sim Gm^2/d$, so $d \sim Gm/v^2$. This means that

$$F^{\text{induced}} \sim (v/c)^{10} F^{\text{Newton}} , \tag{1.145}$$

and since the Newtonian order is the 0PN one tidal forces come in at 5PN order.

While this is a very high order, the dimensionless tidal deformability parameters $\Lambda$ are numerically large: therefore, it can be useful to include higher-order tidal terms in a lower-order calculation.

The effect of spin-orbit interactions, on the other hand, comes about at 1.5PN order, while spin-spin interactions are a 2PN effect.

### 1.4.2 Numerical Relativity

The PN expansion, even if it goes to high order, is still not enough when we approach the merger.

Thus, we need to compare our results to NR simulations, in which the spacetime metric and the stress-energy tensor are evolved according to the Einstein equations. This is possible but time-consuming: each simulation requires many days of supercomputer time.

The gravitational signal can be extracted from the simulated spacetime by computing the Weyl scalar $\Psi_4 = C_{\alpha\beta\gamma\delta} n^\alpha m^\beta n^\gamma m^\delta$,[25] where $n$ and $m$ are component vectors of a null tetrad [Löf+12, sec. 5.6.3]. This scalar is related to gravitational wave emission by

$$\Psi_4 = \ddot{h}_+ - i\ddot{h}_\times . \tag{1.146}$$

The advantage is that we can incorporate the full machinery of GR, and account for the internal dynamics of neutron stars, as well as other phenomena such as the formation of an accretion disk after the merger [Ned+20].

[25] Here $C_{\alpha\beta\gamma\delta}$ is the Weyl tensor, the fully traceless part of the Riemann tensor, whose definition can be found for example in Gourgoulhon [Gou07, eq. 2.18].

Besides using the waveforms as validation for any model, an approach we can use for waveform generation is called "Inspiral-Merger-Ringdown (IMR) phenomenological": a PN waveform is complemented with a power series fitted to NR simulations [Kum+15]. Several variations on this idea are implemented in the LIGO Algorithm Library (LAL) [Col18].

### 1.4.3 Effective One Body

The EOB framework allows for the generation of waveforms encompassing inspiral, merger and ringdown, by mapping the relativistic two-body problem onto the motion of a test particle in an effective metric. This problem is described through an effective Hamiltonian, whose equations of motion are then numerically solved. Let us describe the procedure in a simplified case, including neither spin nor tidal interactions.

The Hamiltonian of the real system is approximated with a PN one, written in relative Arnowitt, Deser, Misner (ADM) coordinates and made dimensionless [Dam14, eqs. 4–5]:

$$\hat{H} = \frac{H}{\mu} = \underbrace{\frac{p^2}{2} - \frac{1}{q}}_{\text{Newtonian}} + \underbrace{\frac{1}{8}(3\nu - 1)(p^2)^2 - \frac{1}{2}\Big[(3+\nu)p^2 + \nu(n\cdot p)^2\Big] + \frac{1}{2q^2}}_{\text{1PN}} + \dots , \tag{1.147}$$

where $q$ and $p$ are dimensionless variables related to relative distance and the corresponding momentum by $q = rc^2/GM$ and $p = p'/\mu$, where $p'$ is the momentum with the correct dimensions.

This is then matched[26] to the Hamiltonian of a particle moving in an effective metric

$$g_{\text{eff}} = A(u,\nu)\,\mathrm{d}T^2 + B(u,\nu)\,\mathrm{d}R^2 + R^2\,\mathrm{d}\Omega^2 \tag{1.148}$$

$$A(u,\nu) \approx 1 + \widetilde{a}_1(\nu)u + \widetilde{a}_2(\nu)u^2 + \dots \tag{1.149}$$

$$B(u,\nu) \approx 1 + \widetilde{b}_1(\nu)u + \widetilde{b}_2(\nu)u^2 + \dots , \tag{1.150}$$

where $u$ is the inverse radial coordinate $u = GM/c^2R$. The way to do so is to write both Hamiltonians in Delaunay (action-angle) coordinates. The metric potentials $A$ and $B$, to 3PN order, read:

$$A_{\text{3PN}} = 1 - 2u + 3\nu u^2 + \left(\frac{94}{3} - \frac{41\pi}{32}\right)\nu u^3 \tag{1.151}$$

$$B_{\text{3PN}} = \frac{1 - 6\nu u^2 + 2(3\nu - 26)\nu u^3}{A_{\text{3PN}}} . \tag{1.152}$$

Note that the $\nu \to 0$ case corresponds to one of the masses vanishing: therefore, we are looking at a test particle in Schwarzschild geometry, and we indeed recover $A = 1 - 2u = 1/B$. The other extreme case is $\nu = 1/4$, which corresponds to $m_1 = m_2$.

[26] The way this "matching", this construction of a dictionary between the two- and the one-body problems, is performed is through a quantum-mechanical analogy. The idea is to consider the quantized energy levels corresponding to the classical Hamiltonians, which will depend on the quantum numbers $n$ (principal quantum number) and $\ell$ (total angular momentum), while the quantum number $m$, describing the $z$ component of the angular momentum, is irrelevant because of the spherical symmetry of the problem. One can then determine the correspondence by establishing a rule to translate between the energy levels corresponding to the same $(n, \ell)$ pairs [Dam14].

A strategy which is then used is Padé resummation: the idea is to take a Taylor expansion (such as those for the PN potentials) and match it to a ratio of polynomial functions, so that

$$f^{(N)}(x) = \sum_{n=1}^{N} c_n x^n \approx P_L^M[f^{(n)}(x)] = \frac{\sum_{n=1}^{M} a_n x^n}{\sum_{n=1}^{L} b_n x^n}, \tag{1.153}$$

where the coefficients $c_n$ are fixed, while the $a_n$ and $b_n$ are determined algebraically by having equal-order terms match in the two series.

A Padé approximant $P_L^M$ can then be applied to any polynomial function. This may seem pointless, since it reproduces the PN result at the same order, however in practice the Padé-resummed expression often exhibits desirable behaviours, such as avoiding divergences or having faster convergence to the true potential.

The Hamiltonian whose equations of motion are actually solved is expressed in terms of the potentials [BB21, eq. 8.22]:

$$H_{\text{EOB}} = Mc^2\sqrt{1 + 2\nu(\hat{H}_{\text{eff}} - 1)} \tag{1.154}$$

$$\hat{H}_{\text{eff}} = \frac{H_{\text{eff}}}{\mu} = \sqrt{A(u,\nu)\left(1 + p_\varphi^2 u^2 + 2\nu(4-3\nu)u^2 p_{r*}^4\right) + p_{r*}^2}, \tag{1.155}$$

where $p_{r*} = \mu^{-1}\int (B/A)\,\mathrm{d}R$ is the momentum corresponding to a rescaled radial variable, while $p_\varphi = P_\varphi/(\mu G M)$ is a rescaled angular momentum.

This formalism can then be complemented by including some further elements: one is the radiation reaction force, which can be expressed as an influence decreasing the momentum $p_\varphi$ according to the emitted flux. Including this term means we are not considering an *adiabatic* approximation anymore.

Further, next-to-quasi-circular terms are introduced: these are multiplicative corrections to the last orbits which account for the fact that these are getting further and further from being circular. Finally, the waveform is smoothly connected to a model for the ringdown, which is informed by black hole (BH) perturbation theory and NR simulations.

EOB models can also incorporate tidal and spin effects: the waveform model `TEOBResumS` [Nag+18] is applicable to spin-aligned, tidally interacting compact objects.

These EOB models can reproduce NR results quite well: for example, in the case of spinning BBH binaries Nagar, Bonino, and Rettegno [NBR21, fig. 2] find mismatches between the EOB and GWs from NR simulations to be at most of the order of $F \sim 10^{-2}$ and typically even less, $F \sim 10^{-3}$.

The EOB models natively output time-domain waveforms, while as discussed in section 1.3 we need them in the frequency domain in order to perform data analysis. One must therefore find a way to move to the frequency domain. An option is to use a SPA [GBN20]; this has been found to be quite successful in providing fast waveform evaluations even for low initial frequencies. However, it is still an approximation: ideally, we would want to numerically Fourier-transform the time-domain waveform. The issues with This becomes quite slow as we reach lower initial frequencies: the duration of the waveform for an equal-mass BNS with $M \approx 2.8 M_\odot$ is typically [Mag07, eq. 4.21]

$$\text{duration} \approx 3\,\text{min}\left(\frac{f_0}{20\,\text{Hz}}\right)^{-8/3}, \tag{1.156}$$

which is graphically shown in figure 1.5.

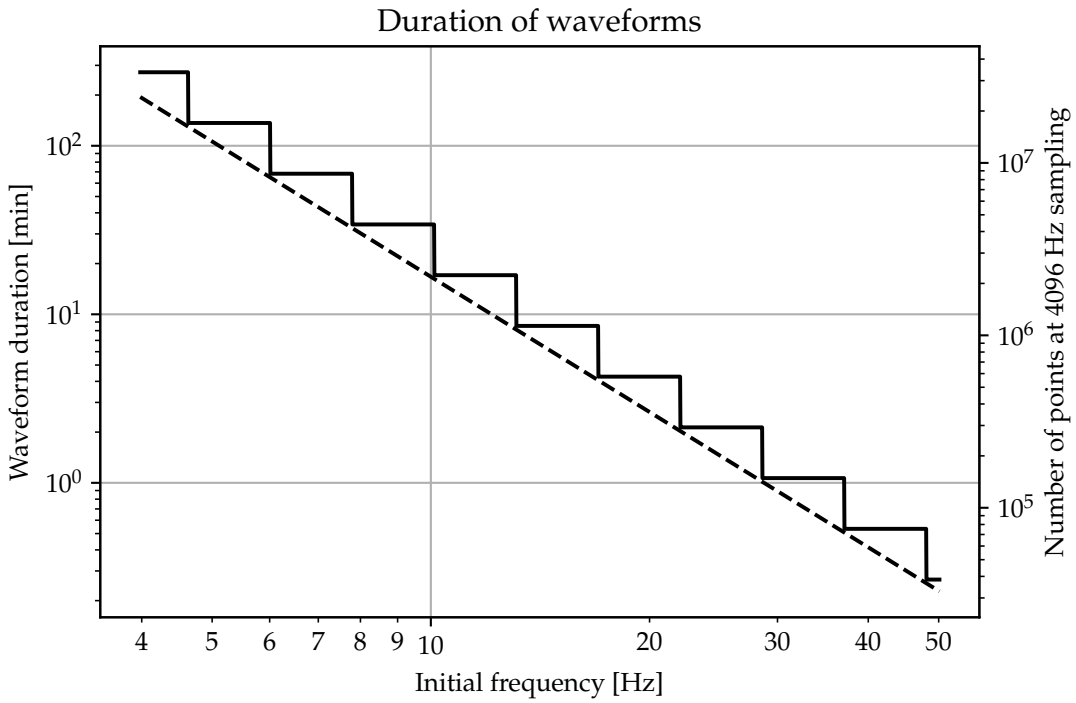


Figure 1.5: Duration of time-domain waveforms for a $M = 2.8M_\odot$ BNS. The dashed line shows the output of equation (1.156), while the thick line shows the durations one might concretely use, imposing the number of samples in the time domain is an integer power of 2 and at least slightly larger than the typical duration, so as to have a margin.

# Chapter 2

# Machine Learning

## 2.1 Supervised learning and optimization

Machine Learning is about having an algorithm improve through the use of *training data*, so that is it able to make predictions or decisions automatically, without any specific outcome being explicitly programmed in.

There are many kinds of problems this general approach can be used to solve, one of which is known as *supervised learning*. The idea is to start from a set of training data consisting of pairs $(x, y)$, where $x$ and $y$ are typically high-dimensional objects, and to train a system so that it is able to reconstruct $y_{\text{new}}$ from a given $x_{\text{new}}$ to within some tolerable margin of error.

Supervised learning differs from *unsupervised* learning, in which data is provided without labels of any sort, and the algorithm must seek some sort of structure in it.

The software `mlgw_bns` uses both kinds: the neural network (NN) is trained with supervised learning, while the PCA dimensionality reduction technique and the down-sampling are unsupervised.

### 2.1.1 Dataset management

The way a supervised learning system typically works is to build a model $y_{\text{pred}} = f(x; \alpha, \beta)$ which depends on variable parameters $\alpha$ and fixed hyperparameters $\beta$. For concreteness, we can think of polynomial regression: $\beta$ may then be the degree of the polynomial with which we fit the data, while $\alpha$ is the set of coefficients of this polynomial.

We can evaluate the performance of this model by comparing $y_{\text{pred}}$ with the known label $y$ for all the data in the training dataset. This allows us to compute some cost function $C(\alpha; \beta)$ — the choice of $C$ is not simple in general, let us keep it abstract for now.

Then, we can use some optimization procedure to find

$$\overline{\alpha} = \underset{\alpha}{\operatorname{argmax}}\, C(\alpha; \beta)\,. \tag{2.1}$$

This sounds good, but there is a problem: a procedure as described can get arbitrarily low costs by learning *specific features* of the training dataset which will not generalize. This is known as *overfitting*, and it is characterized by a low error in the training dataset but a high error on new data.

Its counterpart, *underfitting*, may happen when our model is not "detailed" enough to capture some feature of the data. Here, the model will not only perform badly on new data, but also on its training data.

The parameters $\beta$ can typically be adjusted to find an optimum between these two extremes. A concrete example is shown in figure 2.1: some noisy data is fitted with polynomials of differing degrees — this corresponds to minimizing $C(\alpha;\beta)$ for different values of $\beta$. Only the data in the central region is used for the fit, while the rest is interpreted as validation data.

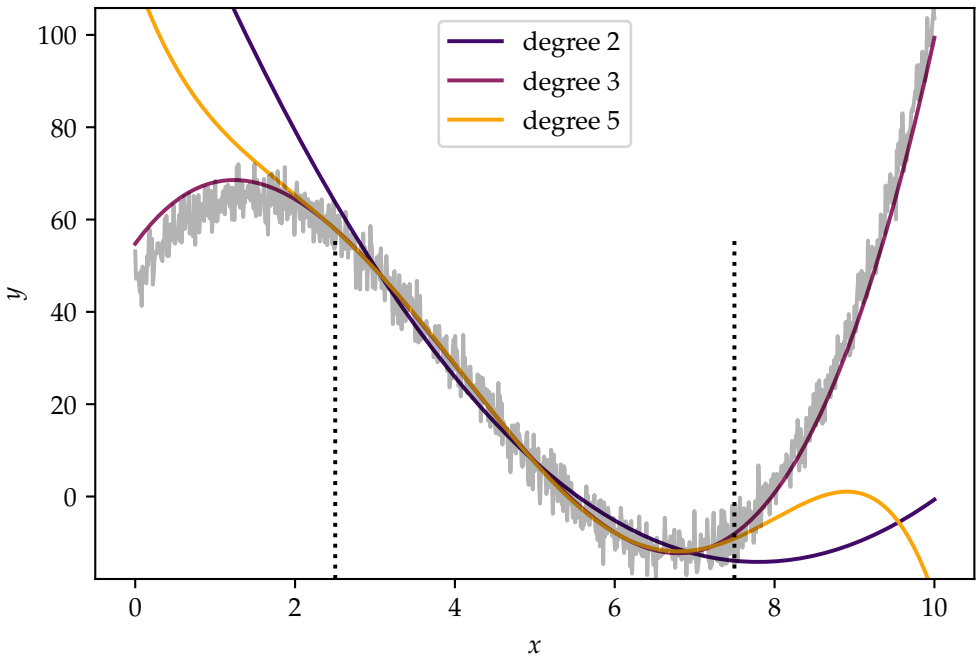


Figure 2.1: Demonstration of under- and over-fitting: the training data here are the central region between the two vertical lines; the models are then validated in the remaining part of the region. See figure 2.2 for the exact value of the training and validation errors at various polynomial degrees.

Note that this split of training and validation data, while visually clear, is undesirable in real circumstances: we can achieve much better accuracy if the training data spans every part of the region of interest. This is the case for acmb, since we can generate data at will.

If we denote the error on the validation data as $C_V(\alpha;\beta)$, the full procedure can then be schematically be written as

$$\overline{\beta} = \underbrace{\underset{\beta}{\operatorname{argmax}}}_{\text{validation}} \; C_V \left( \underbrace{\underset{\alpha}{\operatorname{argmax}} \, C(\alpha;\beta)}_{\text{training}};\; \beta \right) . \tag{2.2}$$

The best ways to calculate these two optimums differ, both among each other and on

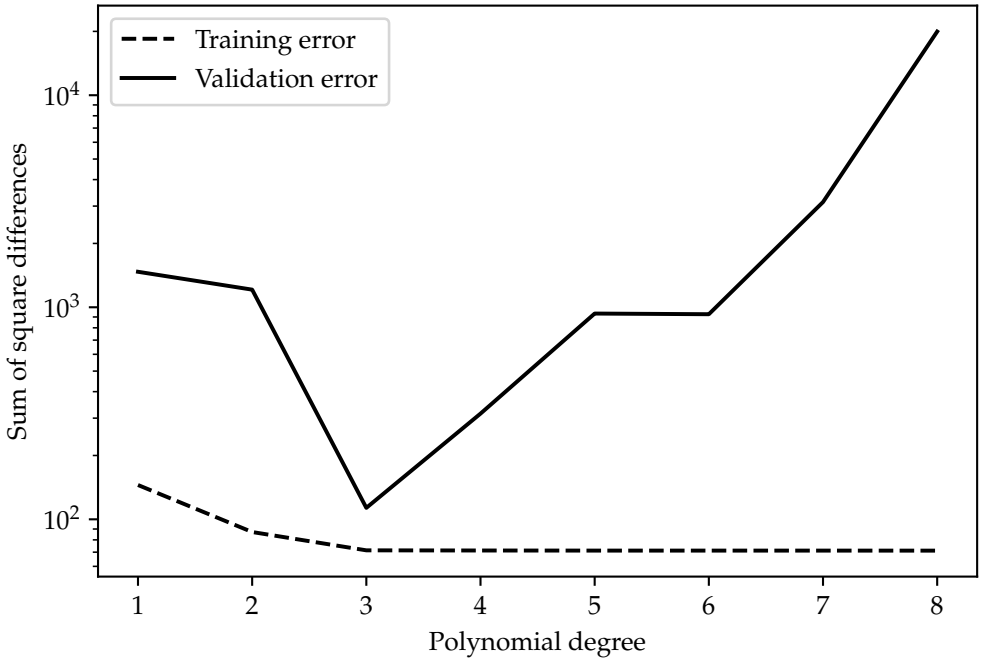


Figure 2.2: Training and validation errors for figure 2.1. The best choice for the degree here is clearly 3, since it minimizes the validation error as well as having a very low training error, but without the validation procedure we could not appreciate it being a better choice than higher values.

a case-by-case basis: the validation procedure must account for the fact that the training procedure for each value of $\beta$ analyzed can take some time.

In particular, the algorithm used in `mlgw_bns` for training is discussed in section 2.3.4, while the one used for validation is discussed in section 2.4.

The procedure described here makes use of a training and a validation dataset, but typically also we want to have some objective measure of the quality of the fit. This cannot be provided by only using training and validation data information, since the parameters of the model depend on them: we need a third separate *testing* dataset which does not inform the model in any way.[1]

## 2.2 Principal Component Analysis

After downsampling, a waveform used by `mlgw_bns` is described by several hundred points. It is convenient to reduce this number in order for the neural network to be faster. We are able to do so by making use of the fact that the components of the high-dimensional vector representing the waveform are correlated.

The technique of PCA is quite general,[2] so let us describe it in general terms, and

[1] Confusingly, machine learning (ML) literature sometimes inverts the labels "validation" and "testing" — we shall stick with the definition given above.

[2] This technique is quite old: it was developed in 1901 by Karl Pearson [Pea01; MN17], but it started to

then apply it to our specific problem.

Dimensionality reduction algorithms such as PCA may be considered as a kind of unsupervised learning, since they can allow for the detection of relevant features in high-dimensional datasets. Here we will not use it for this purpose though, it will be more of a pre-processing step.

### 2.2.1 General method

We start with a dataset of $N$ points in $\mathbb{R}^D$, which we denote by $\{x^i\}_{i=0}^{N-1}$. We need $D$ floating point numbers to represent each of these points.

If we can find a $k$-dimensional hyperplane in $\mathbb{R}^D$, with $k \ll D$, such that our points are never very far from this subspace, we can substitute the $D$-dimensional parametrization of the points for a $k$-dimensional one by approximating each point by its orthogonal projection onto the $k$-dimensional hyperplane. We will make a certain error in this process: specifically, if $P_k(x_i)$ denotes the projection of the point onto this hyperplane, the error (computed according to the Euclidean distance among points) can be quantified by

$$\text{error}(k) = \sum_{i=0}^{N-1} \left\| x_i - P_k(x_i) \right\|^2 . \tag{2.3}$$

The algorithm of PCA allows us to determine which hyperplane minimizes this error.

The first step is to center the data: we compute their mean $\overline{x}$, and work with the dataset $y_i = x_i - \overline{x}$. Because of this, we can say that the $k$-dimensional hyperspace is now a *subspace* with respect to $y$. Computationally, we keep the mean $\overline{x}$ saved and add it to the reconstructed data $y$.

Let us now consider the $k = 1$ case: we want to project the data onto a single line, which we can parametrize as the span of a unit vector $w$. Therefore, what we want to minimize is $\sum_i \left\| y_i - (y_i \cdot w)w \right\|^2 = \sum_i \left( \left\| y_i \right\|^2 - (y_i \cdot w)^2 \right)$, and we can do so by maximizing $\sum_i (y_i \cdot w)^2$.

Therefore, the best 1-dimensional subspace is:

$$w = \underset{w \in \mathbb{S}^{D-1}}{\text{argmax}} \sum_i (y_i \cdot w)^2 . \tag{2.4}$$

Now comes the clever idea of PCA: we can reformulate this argmax problem as an eigenvalue problem for the covariance matrix of the data:

$$C = \frac{1}{N} \sum_i y_i y_i^\top . \tag{2.5}$$

A unit eigenvector $w$ of this matrix will satisfy $Cw = \lambda w$ for its eigenvalue $\lambda$, and we can recover the eigenvalue $\lambda$ from this equation by computing $w^\top C w = \lambda w^\top w = \lambda$; making the covariance matrix explicit allows us to see that

$$\lambda = w^\top C w = \frac{1}{N} \sum_i (y_i \cdot w)^2 ; \tag{2.6}$$

see broad use once availability of computers became widespread.

which is precisely the quantity we wanted to maximize: therefore, the best one-dimensional subspace is precisely the largest eigenvector of the covariance matrix, the direction of maximum variance.

If we make the further observation that the covariance matrix is symmetric and positive definite, and can therefore be orthogonally diagonalized, we are almost done: we can generalize to arbitrary $k$ moving one vector at a time. To find the second vector to span the subspace we can restrict ourselves to the subspace $w^{\perp}$ and apply the same procedure as before, this tells us that the optimal two-dimensional subspace is the span of the first two eigenvectors of the covariance matrix, and so on.

In order to perform a reduction onto a $k$-dimensional subspace, then, we need to calculate the unit eigenvectors $\{w_i\}_{i=0}^{k-1}$ corresponding to the $k$ largest eigenvalues; we can understand these as the columns of a $D \times k$ matrix $V$, which we can then use to construct the projection matrix onto the $k$-dimensional subspace.

In terms of the $D$-dimensional coordinates, the projection matrix is $VV^{\top}$: its application to a vector $y$ can be written as

$$P_k(y) = VV^{\top}y = \begin{bmatrix} w_1 & \cdots & w_k \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_k \end{bmatrix} y = \sum_{i=0}^{k-1} (w_i \cdot y) w_i \in \mathbb{R}^D . \tag{2.7}$$

For the purpose of dimensionality reduction, however, we are not interested in this projection, which still yields a $D$-dimensional vector: all the "reduced" vectors lie in a $k$-dimensional subspace, therefore we express them with only $k$ coordinates, by computing $V^{\top}y \in \mathbb{R}^k$.

If we approach the problem by diagonalizing the covariance matrix $C$, the computational complexity in the worst case scenario is $\mathcal{O}\left(D^3\right)$, since it involves the diagonalization of a $D \times D$ matrix.

### 2.2.2 PCA for waveforms

After downsampling and computing the residuals from the PN waveform, we are left with a waveform described by a vector of a few hundred points for the amplitude and similarly for the phase.

It would be possible to perform PCA separately for these two vectors; however if we combine them into a single one we can exploit any existing correlation between the amplitude and phase residuals. In the worst case scenario — amplitude and phase residuals being completely uncorrelated — this procedure will perform exactly like separating the PCA into two. So, we want to combine amplitude and phase residuals into a single vector.

A simple way to do so is to simply "append" one vector to the other, and therefore consider the waveform as a vector in $\mathbb{R}^{D_A + D_\varphi}$. There is an issue with this procedure: PCA optimizes Euclidean distance, so the scaling of the amplitude and phase residuals becomes relevant. Fortunately, this carries no physical meaning: because of the way we define the residuals 3.3, it is equivalent to a change in the basis for the logarithm.

Heuristically, we want the typical scale of the phase residuals to roughly match that of the amplitude residuals. This can be simply achieved by multiplying either part of the vector by a constant, which can be tuned in order to optimize the reconstruction error.

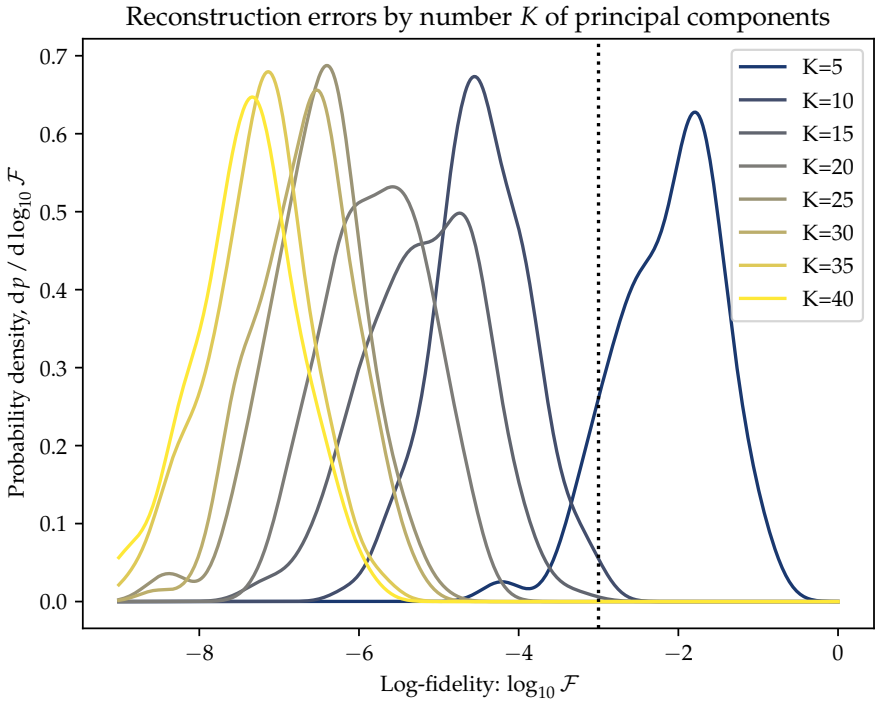


Figure 2.3: The reconstruction error is computed after projecting onto a $K$-dimensional subspace a number $N_w = 64$ waveforms. The parameters for the PCA model are fitted with an independent, $N_{\mathrm{PCA}} = 128$ waveform dataset. A KDE is performed for each set of corresponding log-fidelities, in order for the probability curve to look smoother. Here the EOB waveforms are calculated with a SPA. A vertical dashed line indicates a "desirable" maximal reconstruction error, $\mathcal{F} \sim 10^{-3}$, which roughly corresponds to the accuracy of the underlying EOB model to NR simulations.

## 2.3 Neural Networks

After the reduction of dimensionality through PCA, we are left with the task of approximating the map between the parameters of the system generating the waveform, $\theta_i$, and the $k$ principal components.

In the work of Schmidt et al. [Sch+20] this was accomplished through a Mixture of Experts model, which amounts to a fit to a polynomial expression of the parameters; here instead we reconstruct the function with a neural network.

NNs are known to be able to approximate arbitrary functions [Nie15], and in practice they are quite versatile and usually not prone to overfitting. In the next section we will describe the architecture of a multi-layer perceptron regressor, the kind of network used by `mlgw_bns`.

### 2.3.1 Multi-layer perceptrons

This architecture is built in order to solve the problem of reconstructing a map from an input $x_i \in \mathbb{R}^n$, to an output $y_i \in \mathbb{R}^k$.

We introduce a *hidden layer* between the input and the output. This consists of a

certain number $m$ of "neurons", which can be more or less activated as a function of the inputs. Specifically, the $j$-th neuron in the hidden layer will have an activation level given by the expression:

$$z_j = \sigma\left(w_{ji}^{(1)} x_i + b_j^{(1)}\right), \tag{2.8}$$

where $\sigma\colon \mathbb{R} \to \mathbb{R}$ is called the *activation function*, while the parameters $w_{ji}^{(1)}$ and $b_j^{(1)}$ are respectively called the *weights* and the *biases* (for the first hidden layer).

We shall discuss the reasons one might have for choosing different activation functions later; for now let us say that it is typically differentiable almost everywhere, and it achieves low values for low inputs and high values for high inputs. A common choice, for example, is a logistic sigmoid in the form $\sigma(z) = (1 + e^{-z})^{-1}$.

The weights and biases of the network are free parameters, real numbers which will need to be tuned by the training process.

Once the network has computed the activations $z_j$ for our single hidden layer, it can compute the activations for the output: in this last stage we use no activation function, and the output of the network is simply

$$y_\ell = w_{\ell j}^{(2)} z_j + b_\ell^{(2)} = w_{\ell j}^{(2)} \sigma\left(w_{ji}^{(1)} x_i + b_j^{(1)}\right) + b_\ell^{(2)}. \tag{2.9}$$

In the end, therefore, the number of free parameters of the network is $nm + m$ for the first layer and $mk + k$ for the second, so $m + k + m(n + k)$ in total.

Adding more layers is not conceptually different, it only amounts to applying the procedure described by equation 2.8 again to the result of the first layer, and so on; each time we use the activation function, except for the output layer.

The activation function is crucial for our network to be able to capture nonlinearities: if we were to remove it, with any number of layers the network would still be a linear function of the inputs.

The implementation we use for `mlgw_bns` is the one provided by `scikit-learn` [Ped+11]; specifically, we choose an `MLPRegressor`.[3]

### 2.3.2 Training

The network is able to reconstruct our function as long as the weights and biases are appropriately set: how do we train it to ensure this?

We need to assign a loss function to the output of the network. Here, simplicity trumps accuracy — the "true" error we might like to work with is given by the Wiener distance among waveforms reconstructed from their PCA components, but in order to efficiently train the network we need something easier to compute. The typical error chosen is quadratic in the Euclidean distance, since its analytical derivative is easy to compute:

$$\text{error}(y) \propto \sum_{\text{training data}} \left\| y_{\text{predicted}} - y_{\text{true}} \right\|^2. \tag{2.10}$$

We also add an error term in the form $\alpha \|W\|^2$, where $\alpha$ is a non-negative hyperparameter (typically chosen to be small) and $\|W\|^2$ is the L2 norm of the weight tensor:

[3] The documentation for this network can be found at the url https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPRegressor.html.

this term is known as a "regularizer", it penalizes complex models. The reason why the L2 norm specifically is often chosen is because of the simplicity and efficiency of its implementation.[4] The parameter $\alpha$ is optimized in the hyperparameter training procedure.

We are then using the Euclidean distance among the $y$s (which for us will be PCA component vectors) as a measure of the performance of the network; this works well enough, but we can make an improvement by noticing that the first PCA components are responsible for more of the variance (and thus more of the distance) between data points. Therefore, we can improve the performance by having the network learn the distance among the rescaled

$$\mathrm{PC}_i \lambda_i^\kappa \,, \tag{2.11}$$

where $\lambda_i$ are the eigenvalues corresponding to the principal components, while $\kappa > 0$ is a hyperparameter. The prior distribution for $\kappa$ is a log-uniform one between $10^{-3}$ and 1.

### 2.3.3 Backpropagation

Once we have our cost function, we need a rule to change the weights $w_{ij}^l$ and biases $b_j^l$ of our network[5] according to the variation of the cost function. We would like to implement some sort of gradient descent algorithm, updating weights and biases by

$$\Delta w_{ij}^l = -\eta \frac{\partial C}{\partial w_{ij}^l} \qquad \text{and} \qquad \Delta b_j^l = -\eta \frac{\partial C}{\partial b_j^l} \,, \tag{2.12}$$

where $\eta$ is called the *learning rate*.

So, we need to compute the gradients $\partial C \Big/ \partial w_{ij}^l$ and $\partial C \Big/ \partial b_j^l$. The first idea one might have to do so is to approximate them to first order, doing something like

$$\frac{\partial C}{\partial w_{ij}^l} \approx \frac{C(w + \epsilon e_{ij}^l) - C(w)}{\epsilon} \,, \tag{2.13}$$

where $w$ is the full weight tensor, while $\epsilon e_{ij}^l$ represents a small increment to that particular weight, $e$ being a "unit tensor".

This strategy turns out to be unfeasible because of its computational complexity: the computation of the cost requires a full pass-through of the network, requiring at least $M$ floating point operations (additions or multiplications) where $M$ is the number of free parameters. This would need to be done to compute the update of each of these parameters, so the number of operations needed to perform a single step of the gradient descent for the full network would be at least $M^2$.

The backpropagation algorithm is a clever idea which allows us to compute the gradient with only a forward pass through the network followed by a backward pass, exploiting the chain rule and the way the weight matrices affect each other iteratively.

[4] In the backpropagation equations, which we will shortly introduce, this eventually translates to a penalty on the update of each weight proportional to the weight itself. The overhead needed in order to compute this is very small.

[5] We have added an index $l$ for the layer: so, $w_{ij}^l$ is the weight that the $j$-th neuron in the $l$-th layer gives to input $i$.

The algorithm can be summarized by the following equations [Nie15, chapter 2]:

$$\frac{\partial C}{\partial w^l_{jk}} = a^{l-1}_k \delta^l_j \tag{2.14}$$

$$\frac{\partial C}{\partial b^l_j} = \delta^l_j \tag{2.15}$$

$$\delta^l_j = \left( \sum_i w^{l+1}_{ij} \delta^{l+1}_i \right) \sigma'(z^l_j) \tag{2.16}$$

$$\delta^L_j = \frac{\partial C}{\partial a^L_j}, \tag{2.17}$$

where $a^l_j = w^l_{ji} z_i + b^l_j$ is the activation for the $j$-th neuron of layer $l$, while $z^l_j = \sigma(a^l_j)$ is the output of the activation function for $a^l_j$.

It is important to note that the backpropagation equations are written without the Einstein summation convention: "$x_j = y_j z_j$" means that the $j$-th component of the vector $x$ is calculated by multiplying the $j$-th components of the vectors $y$ and $z$. This means that we are not computing matrix products, but instead the element-wise Hadamard product.

Working backwards, the last equation tells us how to compute the error corresponding to the output layer, denoted with $L$. We can compute the derivative of the cost function with respect to the activation analytically if, for example, we are using a simple quadratic cost function as described in equation 2.10.[6]

The second-to-last equation tells us how to compute the error $\delta^l_j$ of a layer if we know the error of the following layer. It is an application of the chain rule. We compute the $\delta^l_j$ iteratively going backwards through the network, and having done so we can recover the derivative of the cost with respect to the weights and biases by combining the error $\delta^l_j$ with the activations of each layer, $a^{l-1}_k$, as described by the first two equations.

### 2.3.4 Stochastic gradient descent

The method for gradient descent described by equation 2.12 is still slow if we try to compute the gradient of the cost function by using all the training data we have. It turns out to be more efficient to compute the gradient by only looking at a single example or a small batch of them, chosen randomly: this idea is known as *stochastic gradient descent*.

This typically allows for much faster convergence of the training process.

The algorithm used for the training of the network in `mlgw_bns` is Adam [KB17; Rud16], short for "Adaptive Moment Estimation", as implemented in `scikit-learn` [Ped+11].

The general idea of this algorithm is, first, to not move directly in the direction of the gradient, but instead to keep a running, exponentially weighted average of it, and move in *that* direction. An example where we can imagine this could be useful is if the cost function landscape exhibits a "canyon", with a low slope in a long and narrow central region and steep walls. A direct move in the direction of the gradient might mean we "bounce" between the walls a lot without being able to settle in the middle,

[6] The expression in [Nie15] differs from this one since he applies the activation function to the last layer, which one should do for a classification algorithm but not for a regression algorithm.

while averaging allows the direction of our movement to be smoothed and perhaps fall in the middle region.

The second aspect is to make the step-size adaptive, as opposed to it being strictly proportional to the gradient. A "signal-to-noise ratio" is estimated through the square of the gradient, and it is used to scale the step based on how confident we might be in it being meaningful or not.

Both of these exponential decays are regulated by a hyperparameter — $\beta_1$ and $\beta_2$ respectively, which are commonly set to $\beta_1 = 0.9$ and $\beta_2 = 0.999$, but which can be optimized.

## 2.4 Hyperparameter optimization

Our network will depend on several hyperparameters, such as the number and size of the layers or the learning rate; we want to get as close as possible to the optimal choice of these for the reconstruction of the function mapping the binary system parameters to the principal components to be fast as well as accurate.

This optimization is accomplished through a multi-objective tree-structured Parzen estimator (MOTPE), as described in Ozaki et al. [Oza+20] and as implemented through the Optuna API [Aki+19].

Here we will summarize the mechanism through which a single objective tree-structured Parzen estimator (TPE) works, as originally described in Bergstra et al. [Ber+11, section 4], since the generalization to the multi-objective case [Oza+20] is rather mathematically involved but not too conceptually dissimilar from the single-objective case.

We can abstract away the neural network as a function $f$ which, after being given a set of hyperparameters $\vec{x}$, outputs a cost $y$, which we want to minimize. The evaluation of $f$ is quite costly: it involves the training of the network and a full evaluation of its performance (in our case, the reconstruction of the waveforms and a computation of the Wiener distance to their true counterparts).

So, we want to find a value of $\vec{x}$ which minimizes $y$ with as few evaluations of $f$ as possible. The parameters $\vec{x}$ will be given certain prior distributions initially, from which their values will be drawn randomly. Let us then suppose we already have a set of observations $\{(\vec{x}_i, y_i)\}_i$, and we want to find the best possible new value of $\vec{x}$.

We choose a certain quantile $\gamma$, say 15%, and select a fraction $\gamma$ of the best observations we have. This allows us to find a $y^*$ such that $\mathbb{P}(y < y^*) = \gamma$.

Then, we approximate the probability density $p(\vec{x}|y)$ as follows:

$$p(\vec{x}|y) = \begin{cases} \ell(\vec{x}) & y < y^* \\ g(\vec{x}) & y \geq y^* \,. \end{cases} \tag{2.18}$$

We are condensing the $y$-dependence onto a binary choice between "good" observations, modelled by $\ell(\vec{x})$, and "bad" observations, modelled by $g(\vec{x})$. These two functions can then be estimated by making use of the observations we have in the $y < y^*$ and $y \geq y^*$ cases.

This, crucially, allows us to calculate the **expected improvement** associated with a certain parameter set $\vec{x}$:

There is no improvement if $y > y^*$.

$$\mathrm{EI}_{y^*}(\vec{x}) = \int_{-\infty}^{\infty} \max\left(y^* - y, 0\right) p(y|\vec{x})\, \mathrm{d}y \tag{2.19}$$

$$= \int_{-\infty}^{y^*} (y^* - y) p(y|\vec{x})\, \mathrm{d}y \tag{2.20}$$

$$= \int_{-\infty}^{y^*} (y^* - y) \underbrace{p(\vec{x}|y)}_{=\ell(\vec{x})} p(y) \frac{1}{p(\vec{x})}\, \mathrm{d}y \tag{2.21}$$

Used $p(x,y) = p(x|y)p(y) = p(y|x)p(x)$.

$$= \frac{\ell(\vec{x})}{\gamma\ell(\vec{x}) + (1-\gamma)g(\vec{x})} \underbrace{\int_{-\infty}^{y^*} (y^* - y) p(y)\, \mathrm{d}y}_{\text{independent of } \vec{x}} \tag{2.22}$$

Expanded $p(\vec{x})$ as $\int p(x|y)p(y)\,\mathrm{d}y$ and split the $\ell$ and $g$ cases.

$$\propto \frac{1}{\gamma + (1-\gamma)\frac{g(\vec{x})}{\ell(\vec{x})}}, \tag{2.23}$$

where the term we neglected can be interpreted as the average improvement over all choices of $\vec{x}$, which we hope is large but which we cannot affect with a good choice of $\vec{x}$.

What this tells us is the rather intuitive fact that we want to select points which are favored by the "good" distribution $\ell$ and not by the "bad" distribution $g$, so that $g/\ell$ is small, which means that the expected improvement will be large. The formula also says that we should prefer smaller values of $\gamma$, but it does not take into account the fact that we need a reasonably large selection of points in order to properly model $\ell(\vec{x})$ without a large error. The balance among these two contrasting desires will dictate our choice of $\gamma$.

Once we are able to compute the expected improvement, the task is simple: we just need to generate a selection of points $\vec{x}_i$ from the distribution $\ell(\vec{x})$, evaluate their expected improvement (or just $g/\ell$, really) and pick the one with the largest expected improvement as the one for which to evaluate the NN.

The one aspect still missing in the algorithm is how to estimate a probability distribution from a small set of samples, and how to calculate a new sample from it. There are different ways to do so; the one implemented in the TPE algorithm fits a gaussian mixture model to estimate $\ell(\vec{x})$ and $g(\vec{x})$ [BYC13, section 5].

## 2.5 Greedy downsampling

The residual waveforms generated with the EOB / PN models have on the order of a few times $10^5$ sampling points. This is too large a number to effectively use PCA on: as we will see, it requires writing a covariance matrix which would have $\sim 10^{11}$ entries, way beyond the RAM of most computers.

Therefore, we need to start by applying a simple dimensionality reduction technique: downsampling, applied to the unwrapped phase and amplitude of waveforms (see section 3.2) after the subtraction of the "baseline" PN ones.

Doing so on a uniform grid, however, is suboptimal: there is more detail to be captured in some areas than in others. It is difficult to determine *a priori* which areas will be more relevant than others, therefore `mlgw_bns` implements a greedy algorithm which selects a set of indices which optimize the reconstruction accuracy, inspired by `romspline`, which was introduced by Galley and Schmidt [GS16].

The algorithm is detailed as algorithm 3; it includes certain improvements over `romspline`. First, it allows for a full training dataset as opposed to reconstruction of a single waveform. The points are greedily chosen until the error over the whole dataset is below the threshold. This allows us to be more confident of the fact that the indices

are not capturing the specifics of a single waveform, but instead they are representative of the whole dataset.

**Algorithm 3** Greedy downsampling algorithm.

**Require:** $x_i$, $y_i^{(j)}$, tolerance $\epsilon$.
**Require:** "Seed" indices $I$
  Error $e \leftarrow \infty$
  **while** $e > \epsilon$ **do**
    **for** each $j$ **do**
      $I' \leftarrow \emptyset$
      $e_j \leftarrow 0$
      $y^{(j)}_{\mathrm{rec},i} \leftarrow$ interpolation of $x_I$, $y^{(j)}_I$
      **for** $k_1, k_2$ successive pair in $I$ **do**
        $n \leftarrow \mathrm{argmax}_{i \in [k_1, k_2]} \left| y^{(j)}_{\mathrm{rec},i} - y^{(j)}_i \right|$
        $e_j \leftarrow \max\left(e_j, \left| y^{(j)}_{\mathrm{rec},n} - y^{(j)}_n \right|\right)$
        **if** $\left| y_{\mathrm{rec},n} - y^{(j)}_n \right| > \epsilon$ **then**
          $I' \leftarrow I' \cup n$
        **end if**
        $I \leftarrow I \cup I'$
      **end for**
      $e \leftarrow \min(e, e_j)$
    **end for**
  **end while**
  **return** indices $I$

Whether this is actually happening can be checked with a validation dataset: the validation errors are shown in figure 2.4 as the size of the training dataset increases. The number of indices found corresponding to a fixed tolerance increases with the size of the training dataset, as can be seen in figure 2.5.

The size of the training dataset is limited by memory availability, more so than computational complexity: each training waveform must be fully kept in memory, with its $\sim 10^6$ floating point numbers, each needing 8 bytes of memory. Therefore, we need $\sim 10\,\mathrm{MiB}$ of memory per stored waveform, and a hundred of them will take up $\sim 1\,\mathrm{GiB}$ of storage.

The interpolation method choice is important: higher-order interpolants are slower but need less points. Here we use cubic splines, which have been found to be a good compromise for surrogate models [Lac+19]. As we shall see in section 3.5, the interpolation accounts for the most significant part of the overhead `mlgw_bns` exhibits over the 3.5PN TaylorF2 approximant it uses (roughly 65 % of a 40 % overhead) in the case of high numbers of points to interpolate. This may seem like a reason to switch to a lower-order interpolation mechanism, but it is balanced by the fact that the lower number of points allows for a larger training dataset.

Care must also be put into the choice of the tolerances, since they have a secondary, perhaps unexpected effect: the greedy downsampling algorithm implemented in `mlgw_bns` yields different numbers of downsampling points for amplitude and phase based on how many are needed in order to get the training dataset below the toler-

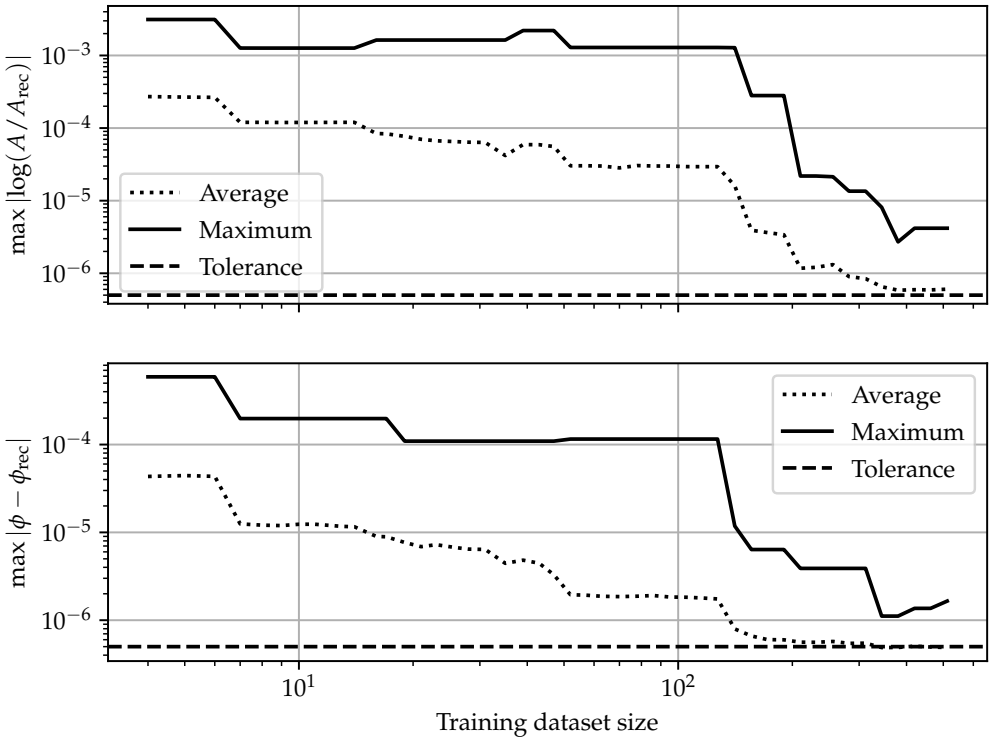


Figure 2.4: Validation errors for downsampled waveforms. Downsampling indices are calculated by applying algorithm 3 to a dataset of varying size, and the error is calculated by computing the maximum reconstruction error for each of $N_{\rm val} = 128$ waveforms. Note that "average" and "maximum" here refer to the difference between taking an average or a maximum over the $N_{\rm val}$-long array of maximum reconstruction errors per validation waveform. The validation waveforms are stochastically generated each time, hence the slight up-ticks one can see at certain points — still, the overall trend is downward.

ance, therefore the combined amplitude+phase vector will have $N_{\rm amp} + N_{\rm phi}$ components, where the numbers of points $N_{\rm amp}$ and $N_{\rm phi}$ depend on the tolerances $\epsilon_{\rm amp}$ and $\epsilon_{\rm phi}$.

There is no natural relation between $\epsilon_{\rm amp}$ ad $\epsilon_{\rm phi}$: one measures a variation in the logarithm of the amplitude, the other a variation in angle. However, there is more to the choice of these than just selecting the error we are willing to accept. The vectors whose distances are computed when doing PCA will have more components for amplitude or for phase, and this will affect the reconstruction error, biasing the reconstruction towards one or the other.

Heuristically, it was found to be typically best to tune the tolerances so that the number of points describing the phase is slightly larger than the ones describing the amplitude.

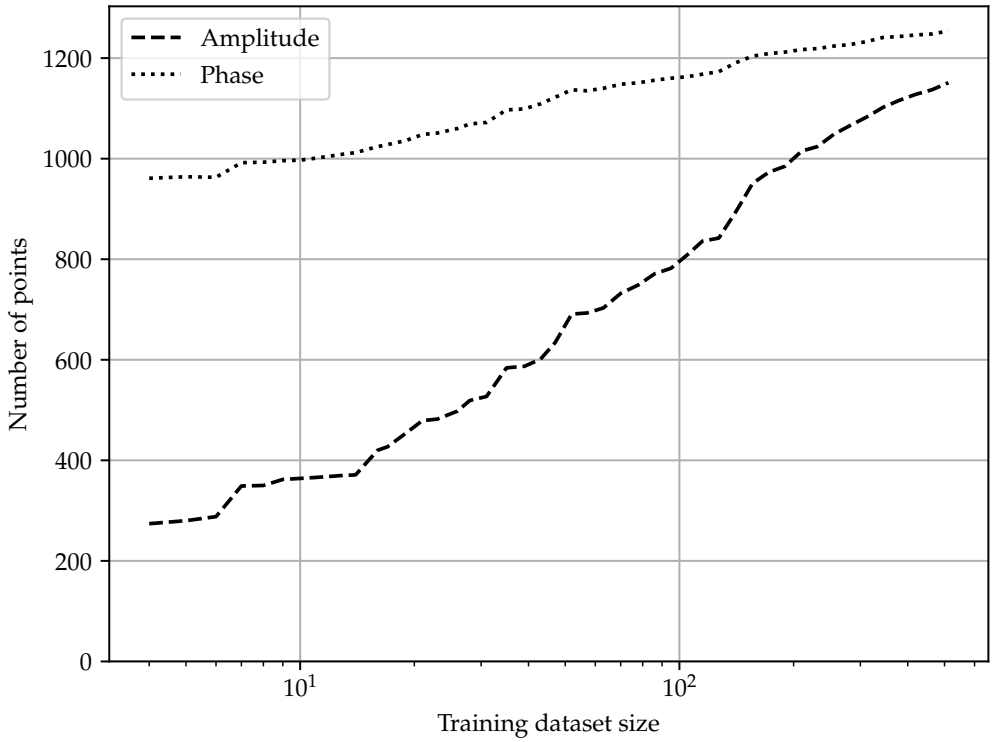


Figure 2.5: Size of the downsampling index array deriving from the application of algorithm 3 to training datasets of varying sizes.

# Chapter 3

# MLGW_BNS

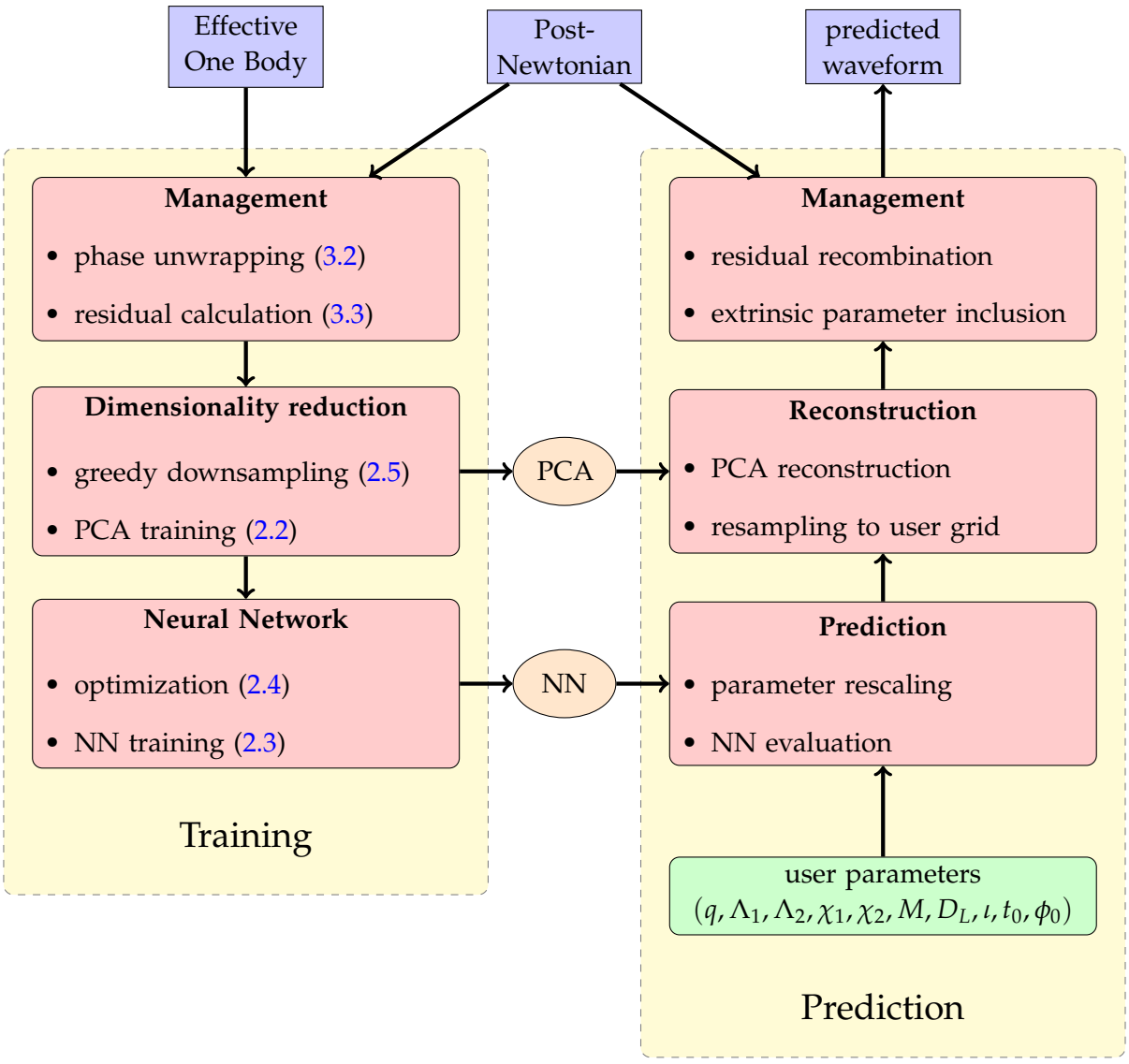


Figure 3.1: Flowchart for `mlgw_bns`.

Let us start with an overview of the algorithm employed by `mlgw_bns` before getting into the technical details.

It is based on the use of two systems for the generation of a theoretical waveform from a CBC in the frequency domain, one being faster but less accurate than the other. The "fast" system will be the PN model `TaylorF2`, as implemented in `bajes` [BGB21], while the "slow" system will be the EOB model `TEOBResumS`.

As we shall see shortly, a crucial feature of the "fast" system is the capability to generate waveforms in the frequency domain at select frequencies, without needing to perform a full Fourier transform from a time-domain waveform. This feature is shared by all PN approximants, which base their Fourier-domain representations on the analytic SPA.

As illustrated in figure 3.1, the main idea is then to

1. compute the residuals of the "slow" waveforms from the "fast" ones, and with these build a dataset;

2. train a machine learning system on this dataset, so that it is able to reconstruct the map from the parameters $\vec{\theta}$ to the residuals.

If this task is accomplished with tolerable errors and with a fast enough execution time (shorter than the time taken by the "slow" generation method) we will have a working *surrogate model*.

The generation of datasets needs us to define a probability distribution over the parameter space: the density of points in each region will bias the reconstruction, but this might actually be desirable if we need to sample from certain regions more than others. A good choice, therefore, could be to generate the training dataset according to a prior distribution over the parameters.

The implementation currently active in `mlgw_bns`, however, simply generates waveforms with a uniform distribution over the selected parameter ranges: $q \in [1, 2]$, $\Lambda_i \in [5, 5000]$[1] and $\chi_i \in [-0.5, 0.5]$.

The nonspinning case, which will be discussed later as a simplification for which the reconstruction errors improve significantly, simply amounts to restricting the prior to the three-dimensional subspace $\chi_i \equiv 0$. The set

## 3.1 Training waveform generation

As discussed in section 1.2.4,

If we reconstruct it, we can recover $h_+$ and $h_\times$ for arbitrary inclination using the two aforementioned expressions.

Fix this bit, citing 1.2.4.

## 3.2 Phase unwrapping

The waveforms generated by the EOB system are complex-valued, in the form $h(f) = A(f)e^{i\phi(f)}$.

[1] The reason why $\Lambda \to 0$ is not included in the range is that the EOB model used is unstable very small $\Lambda$, since certain computations include divisions by it.

It is useful for the later stages of this algorithm to decompose such a waveform into $A(f)$ and $\phi(f)$, however in doing one runs into a complication. The amplitude can be simply calculated as $A(f) = \left|h(f)\right|$, but the phase computed as $\angle h(f)$ is bounded (between $-\pi$ and $\pi$ in the `numpy` implementation), therefore it is discontinuous when $h(f)$ crosses the negative real axis.

In order to overcome this issue we need to make some assumptions about the waveform: we ask that the phase $\phi(f)$ is smoothly varying, densely sampled and almost monotonic.

The first two conditions, in practice, refer to the fact that the variation of the phase between two sample points should be small compared to $2\pi$ — otherwise, we would not be able to tell whether it increased by $x$ or $2\pi n + x$ for some integer $n$.

If $\phi(f)$ were very densely sampled, such that $\Delta\phi$ between two successive points were always $< \pi$, we could drop the condition of almost-monotonicity: an algorithm could simply compute $\angle h(f)$ for each sample point, and add $2\pi n$ as needed in order to make the differences between successive points $< \pi$. This is the algorithm implemented, for example, by the `unwrap` function in the `numpy` library [Har+20].

We can do slightly better, however, by using the quasi-monotonicity assumption and treating decreasing phase differently than increasing phase: we add $2\pi n$ to each point, determining $n$ so that $\phi_{i+1} - \phi_i \in [-\epsilon, 2\pi - \epsilon]$, where $\epsilon$ is some small number quantifying the maximum decrease in phase we expect to see. If $\epsilon = \pi$ this is equivalent to the `numpy` algorithm.

The monotonicity and magnitude of the phase are not invariant: we can add a general linear term $2\pi f t_0 + \phi_0$ to $\phi$ by changing the initial phase and the coalescence time. However, within the convention we are using (where the coalescence happens at the edge of the sample range) the phase of $h(f)$ is indeed almost monotonic, such that $\epsilon \sim 10^{-1}$ is typically a good choice.

Since this implementation is custom-made it is also slightly faster: it takes $\sim 30\,\mathrm{ms}$ to unwrap a waveform with $\sim 5 \times 10^5$ sampling points, as opposed to the $\sim 40\,\mathrm{ms}$ needed for the `numpy` implementation.

## 3.3 Residual calculation

The `mlgw_bns` model reconstructs the residuals of EOB waveforms from PN ones: these are specifically expressed as

$$\Delta A(f;\theta) = \log\left(\frac{A_{\mathrm{EOB}}(f;\theta)}{A_{\mathrm{PN}}(f;\theta)}\right) \tag{3.1}$$

$$\Delta\Phi(f;\theta) = \Phi_{\mathrm{EOB}}(f;\theta) - \Phi_{\mathrm{PN}}(f;\theta)\,, \tag{3.2}$$

where $A$ and $\Phi$ are the amplitude and phase of the respective Fourier domain (FD) waveforms.

The computation of the residuals is made easy by the fact that we have closed-form expressions for the PN waveforms to high order, including tidal effects.

Using higher-order waveforms means that the residuals to fit are smaller, but this comes with a large drawback: the evaluation time of the PN waveforms drastically rises with the order — they are complicated analytic waveforms.

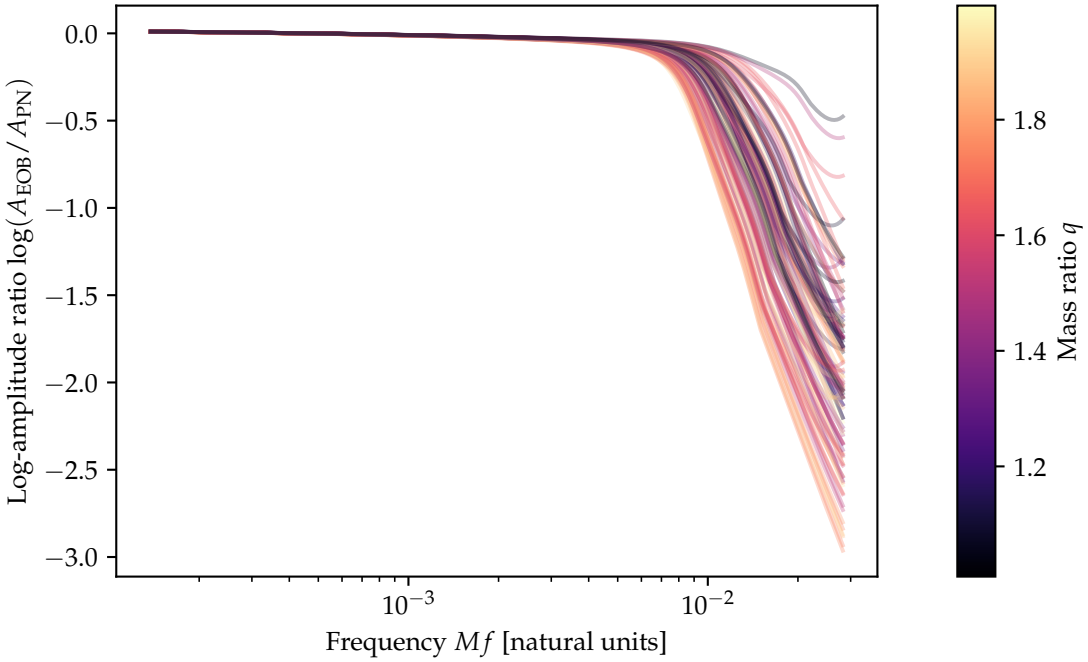


Figure 3.2: Residuals in amplitude of the EOB waveforms versus the PN ones. The waveforms are uniformly distributed across all five parameters.

As we will discuss later, the reconstruction time for a single waveform is dominated by the recovery of the full waveform from the residuals: the computation of $A_{\rm PN}(f;\theta)$ and $\Phi_{\rm PN}(f;\theta)$ in

$$A_{\rm rec}(f;\theta) = \exp\big(\Delta A_{\rm rec}(f;\theta)\big) A_{\rm PN}(f;\theta) \tag{3.3}$$
$$\Phi_{\rm rec}(f;\theta) = \Phi_{\rm rec}(f;\theta) + \Phi_{\rm PN}(f;\theta)\,. \tag{3.4}$$

Therefore, a relatively low-order PN approximant is chosen; specifically, the amplitude is computed to 3.5PN order, and the phase is computed to 3.5PN order including tidal contributions to 6PN order [Fav14].[2]

[2] Specifically, the functions used are `Af3hPN` and `Phif3hPN` implemented in the `bajes` module [BGB21].

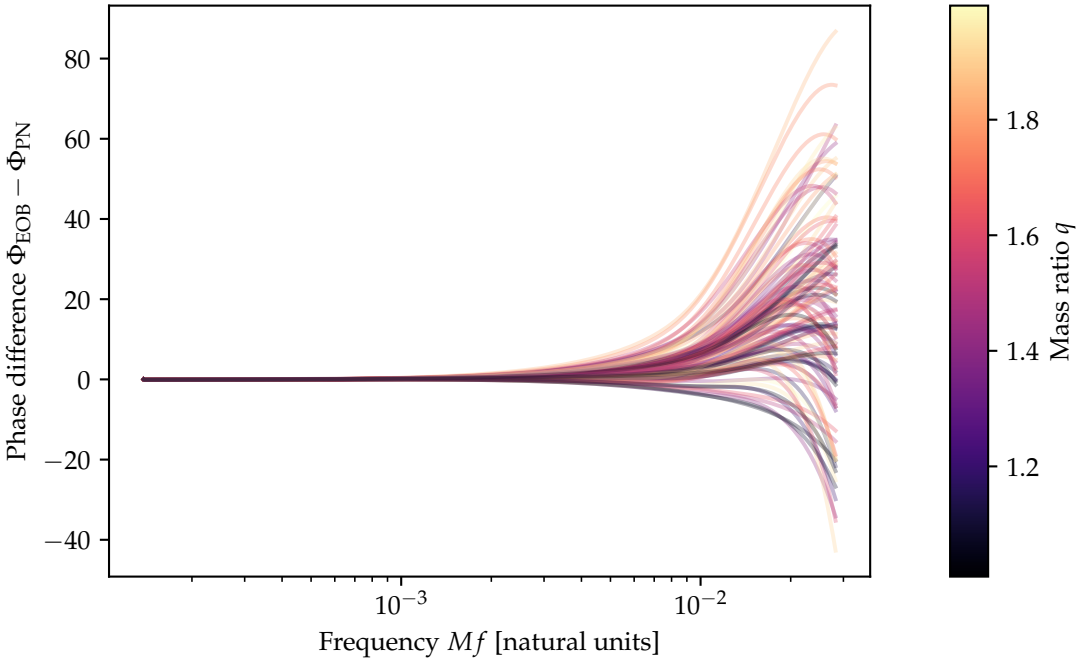


Figure 3.3: Residuals in phase of the EOB waveforms versus the PN ones. The waveforms are uniformly distributed across all five parameters. A linear term has been subtracted from all of these for the plot: this does not affect the physical waveform (it only amounts to a time shift), and it is needed because the time-alignment of the EOB waveform is not exact.

## 3.4 The reconstruction of a waveform

We describe the procedure taken by a trained `mlgw_bns` model in order to reconstruct a waveform starting from a set of parameters $\theta = [q, \Lambda_1, \Lambda_2, \chi_1, \chi_2]$, extrinsic parameters (for simplicity we consider only $\theta_{\text{ext}} = [M, D_L, \iota]$), as well as a set of new frequencies $f$ to re-interpolate the waveform at.

1. The parameters $\theta$ are reduced to $\theta_r$ by a scaler, which is trained so that they each have zero mean and unit variance.

2. The trained NN makes its prediction based on the parameters, let us denote it as $(y_i)_{i=0}^{K-1}$.

3. The prediction is divided by the corresponding PCA eigenvalues, to get $x_i = y_i / \lambda_i^{\alpha}$.

4. The PCA model reconstructs combined vector representing the amplitude and phase residuals: $\text{comb}_i$.

5. The combined vector is split into the amplitude and phase residual vectors, corresponding to their respective amplitude and phase downsampled frequencies.

6. The frequencies $f$ are converted into natural units by dividing them by $c^3/GM \approx (M_\odot/M) \times 2 \times 10^5$ Hz; this yields a set of natural-units frequencies $f_n$.

7. The amplitude and phase are reconstructed from the residuals according to equations (3.3).

8. The amplitude and phase are resampled to the frequencies $f_n$.

9. The waveform is recombined as $h = Ae^{i\Phi}$.

10. The waveform is rescaled according to $M$ and $D_L$, so that it is expressed in SI units (seconds).

11. The two polarizations are recovered by multiplying $h$ by their respective functions of $\iota$ and a phase shift by $\pi/2$ for $h_\times$.

Add new benchmarks

## 3.5 Evaluation time

In broad strokes, on standard benchmarks the evaluation time of a waveform by `mlgw_bns` is slightly better than that of the model it is trained on, `TEOBResumS` with SPA; while in simpler but realistic scenarios its evaluation time is up to an order of magnitude faster.

The standard benchmarks for the evaluation time are shown in figure 3.4, and the ratios between the times for the three models are shown in more detail in figure 3.5.

The assumption underlying all these benchmarks is that, as the initial frequency decreases and the duration of the waveform increases (as shown in figure 1.5), the number of samples for the Fourier transform must roughly match the number of samples in the time domain; specifically, if $r$ is the sample rate and $T$ is the length of the time-series then in the time domain we will need $rT$ samples to represent the signal. Typically, for BNS analysis, the sample rate is set to $2^{12}\text{Hz} = 4096\,\text{Hz}$, while the duration of the signal depends on the initial frequency we want to consider — for example, a BNS with $M = 2.8M_\odot$ starting at 12 Hz can be described within a time series lasting 1024 s, leading to $2^{22}$ samples in the time domain.

On the other hand, in the frequency domain the standard approach is to use uniform frequency sampling, where the lowest frequency we can describe, $\Delta f = 1/T$, will determine the sampling rate. The highest frequency it is useful to record is the Nyquist frequency $f_N = r/2$. Since our signal is real-valued, negative frequency samples are redundant — they are the conjugates of the corresponding positive frequency ones. We also do not need to record components at frequencies lower than $f_0$, the initial frequency of the signal. Therefore, the total number of frequency samples is

We sample both endpoints, hence the +1.

$$N = \frac{f_N - f_0}{\Delta f} + 1 = \left(\frac{r}{2} - f_0\right) T + 1 \approx \frac{r}{2} T\,. \tag{3.5}$$

In the aforementioned case, this corresponds to roughly $2^{21} \approx 2.1 \times 10^6$ samples. In terms of storage, each of these samples is a complex number, typically represented with

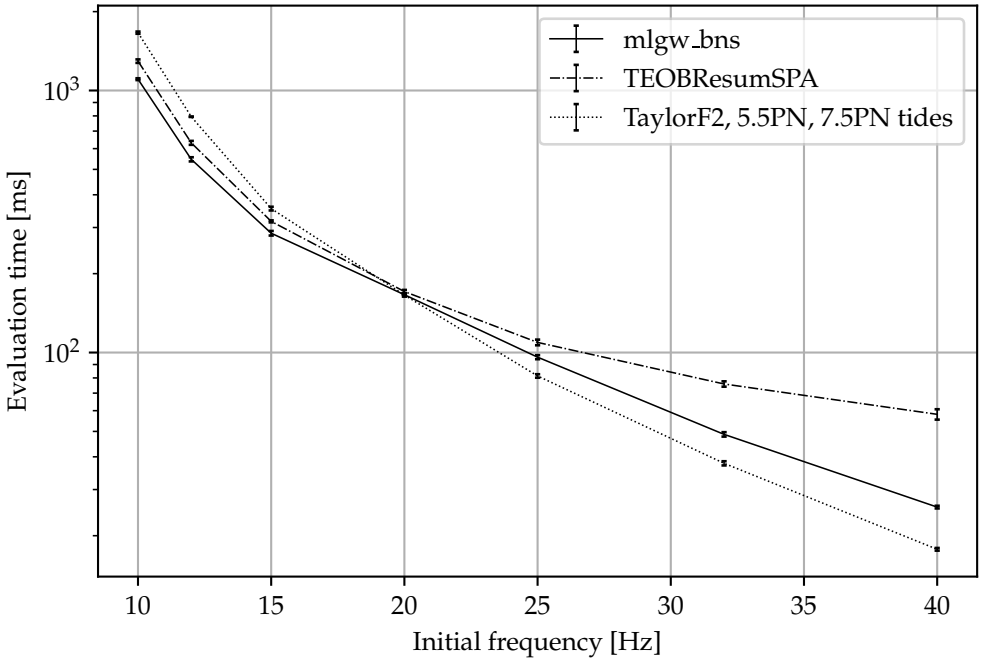


Figure 3.4: Evaluation times. The frequencies at which the models are evaluated are chosen so that they each correspond to a doubling of waveform duration.

two 64-bit floating points Cartesian components, therefore a waveform represented in this way will take up 32 MiB of memory.

Neither `mlgw_bns` nor `TEOBResumS` natively work with so many points, and both need to interpolate their results on a fine grid such as this one. This is the reason why `mlgw_bns`, as shown in figure 3.5, is only slightly faster than `TEOBResumS` when the initial frequency is very low: both algorithms are bottle-necked by resampling, which can be performed in linear time with respect to the number of points in the new, densely sampled array.

### 3.5.1 Downsampled waveform evaluation time

In order to illustrate this point, let us discuss the breakdown of an evaluation of `mlgw_bns` in three different cases. In all three it reconstructs both polarizations for a BNS waveform with $f_0 = 12\,\text{Hz}$, with a time-domain sampling rate of 4096 Hz, total mass $M = 2.8 M_\odot$, mass ratio $q \approx 1.96$, aligned spins $\chi_{1z} \approx 0.38$ and $\chi_{2z} \approx 0.04$, tidal deformabilities $\Lambda_1 \approx 3600$ and $\Lambda_2 \approx 2500$, luminosity distance $d_L = 25\,\text{Mpc}$, and inclination $\iota = 3\,\text{rad}$.

The times we give refer to a single evaluation of `mlgw_bns`, so they cannot be considered fully representative, but they were found to be fairly consistent among different runs; the most significant difference between different runs was found to be the overall scaling of the times, which depends on system factors such as other processes running and memory availability, while the ratios were generally preserved.

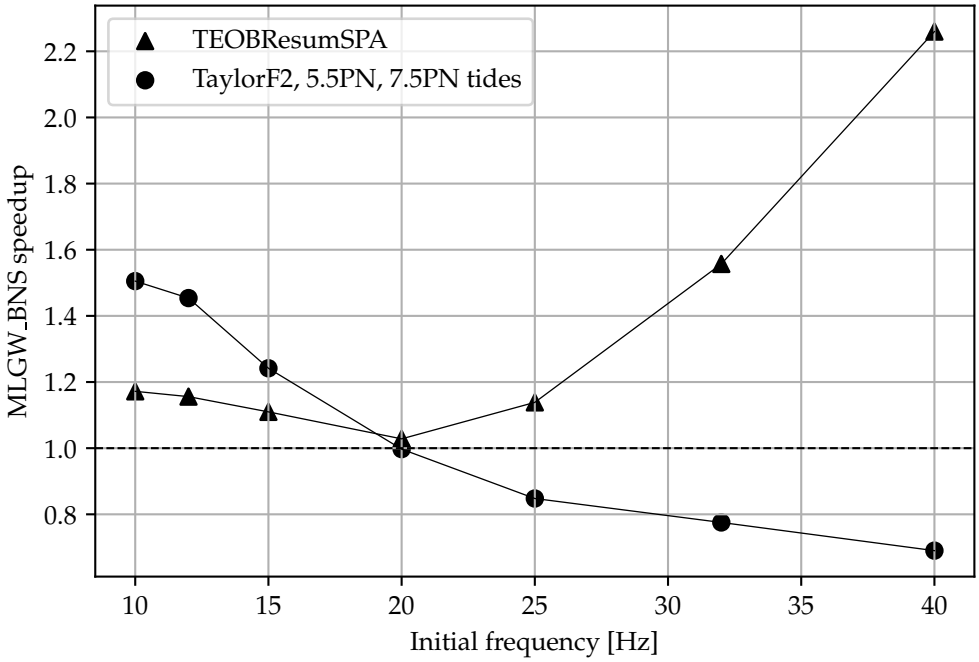


Figure 3.5: Ratios of evaluation times: these are the same data as in figure **??**, viewed as ratios between the two approximants and `mlgw_bns`.

**Full resampling** Here the waveform is resampled to the full number of points, which equals $(f_N - f_0)/\Delta f + 1 = 2084865 \approx 2 \times 10^6$. The total time taken by `mlgw_bns` is 385 ms,[3] of which

1. 246 ms (69 %) are taken by the interpolation (point 8);
2. 103 ms (27 %) are taken by extrinsic parameter operations (points 9 through 11);
3. 0.9 ms (0.2 %) are taken by the NN + PCA routines (points 1 through 5);
4. 0.5 ms (0.15 %) are taken by the evaluation of the PN waveform (point 9);
5. the rest of the time (16 ms, 4 %) is taken by other post-processing operations (such as point 6).

The evaluation time of the baseline PN model, as well as the PCA and NN steps, are basically negligible.

In a comparable scenario, `TEOBResumSPA` took 452 ms, or around 17 % more.

[3] This is slightly faster than the typical time shown in figure 3.4 for an initial frequency of 12 Hz: this comes down to the state of the system at the time the benchmarks were taken — the data for the figure was computed at a different time than the following benchmarks, but the overall slowdown preserves ratios, so the speedups we discuss are rather consistent. Still, the breakdown in the following section refers to a single trial, so it should be taken with a grain of salt.

**16-fold downsampling** Here the waveform provided by `mlgw_bns` is evaluated on a grid made 16 times sparser, so consisting of 130305 points. These were taken to be still equally spaced for simplicity, even though it is not a realistic usage case: the `mlgw_bns` model can interpolate to an arbitrary frequency array, and providing an efficiently chosen, non-uniform frequency array does not change its performance.

In this case the total time taken by `mlgw_bns` is 34 ms, of which

1. 22 ms (64 %) are taken by the interpolation (point 8);
2. 10 ms (30 %) are taken by extrinsic parameter operations (points 9 through 11);
3. 0.6 ms (1.8 %) are taken by the NN + PCA routines (points 1 through 5);
4. 0.6 ms (1.7 %) are taken by the evaluation of the PN waveform (point 9);
5. the rest of the time (0.7 ms, 2 %) is taken by other post-processing operations (such as point 6).

In the same scenario, `TEOBResumSPA` took 98 ms, or around three times more.

**256-fold downsampling** Here the waveform provided by `mlgw_bns` is evaluated on a grid made 256 ($= 16^2$) times sparser, so consisting of 8145 points.

In this case the total time taken by `mlgw_bns` is 4.2 ms. Here we give a more detailed breakdown of the times taken by some subroutines, since they can be more clearly interpreted as percentages of this total.

1. 2 ms (49 %) are taken by the interpolation (point 8);
   (a) of these, 0.7 ms (16 % of the total) are taken by the creation of the cubic spline — this absolute time is consistent, regardless of the number of points it is then evaluated on;
   (b) 1.4 ms (33 % of the total) are taken by the evaluation of the spline.
2. 0.8 ms (19 %) are taken by the NN + PCA routines (points 1 through 5);
   (a) of these, 0.6 ms (15 % of the total) are taken by the NN prediction (point 2), while
   (b) 0.1 ms (3 % of the total) are taken by the PCA reconstruction (point 4);
3. 0.6 ms (15 %) are taken by extrinsic parameter operations (points 9 through 11);
4. 0.5 ms (13 %) are taken by the evaluation of the PN waveform (point 9);
5. the rest of the time (0.1 ms, 3 %) is taken by other post-processing operations (such as point 6).

In the same scenario, `TEOBResumSPA` typically takes 70 ms, or around 15 times more than `mlgw_bns`.

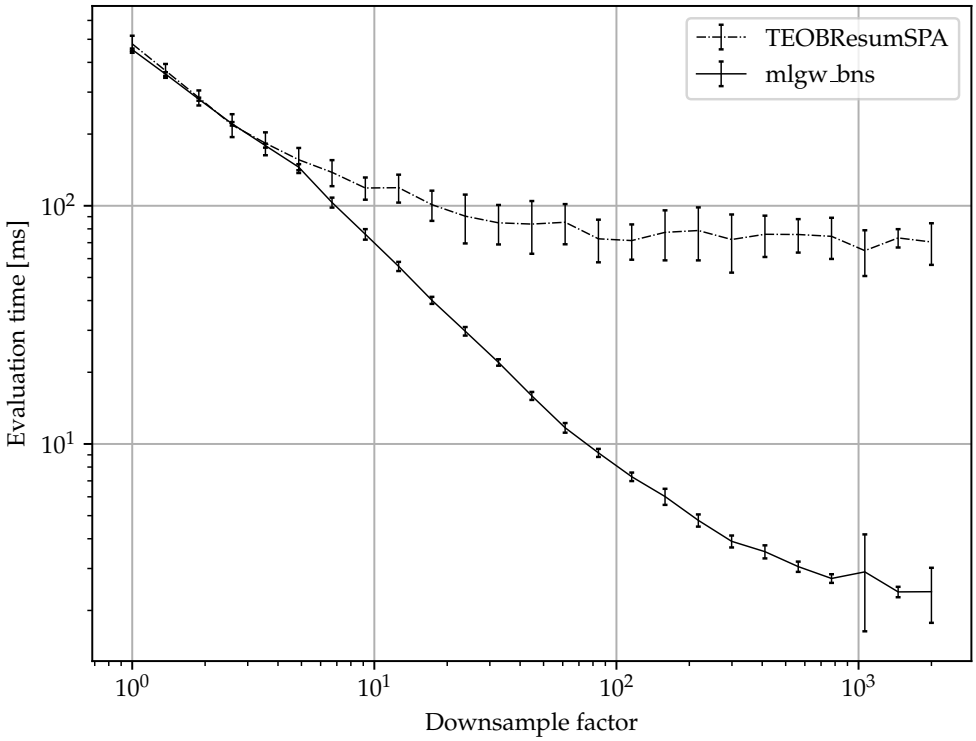


Figure 3.6: Evaluation time for waveforms starting at 12 Hz, as detailed in section 3.5.1; at the very left of the graph the waveforms are evaluated at roughly two million points, while at the very right they are evaluated at roughly one thousand points.

**Reduced order modelling** The improvement given by `mlgw_bns` as opposed to `TEOBResumS` is shown in more detail in figure 3.6: if the waveform is sampled at fewer points, `mlgw_bns` yields a significant speedup.

How many points do we actually need to sample at? Not that many! Vinciguerra, Veitch, and Mandel [VVM17], for example, outline a multi-banding approach: the chirping behavior of the signal means that we know *a priori* that the content at high frequencies is only relative to the short section of time relative to the end of the timeseries.

They subdivide the frequency series into bands with varying frequency resolution, but they keep the resolution constant within each band. This allows for a reduction of the number of points in frequency space by a factor of roughly 150 for $f_0 = 12\,\text{Hz}$ (the same value as that used in figure 3.6), with very low, mismatches of a few times $10^{-7}$.

This already would allow for millisecond-scale `mlgw_bns` evaluations.

Alternative techniques for fast likelihood evaluation exist: an example is reduced order quadratures (ROQ) [Smi+16; Can+15], in which a basis of $N$ waveforms is constructed (typically through a greedy algorithm), among which scalar products can be pre-computed; then, waveforms can be represented in this basis by sampling them at $N$ specific points. The size of these bases typically ranges from $10^3$ to $10^4$.

Another technique is relative binning [ZDV18], in which ratios of the frequency domain waveform to a fiducial, less precise one are downsampled.

## 3.6 Accuracy

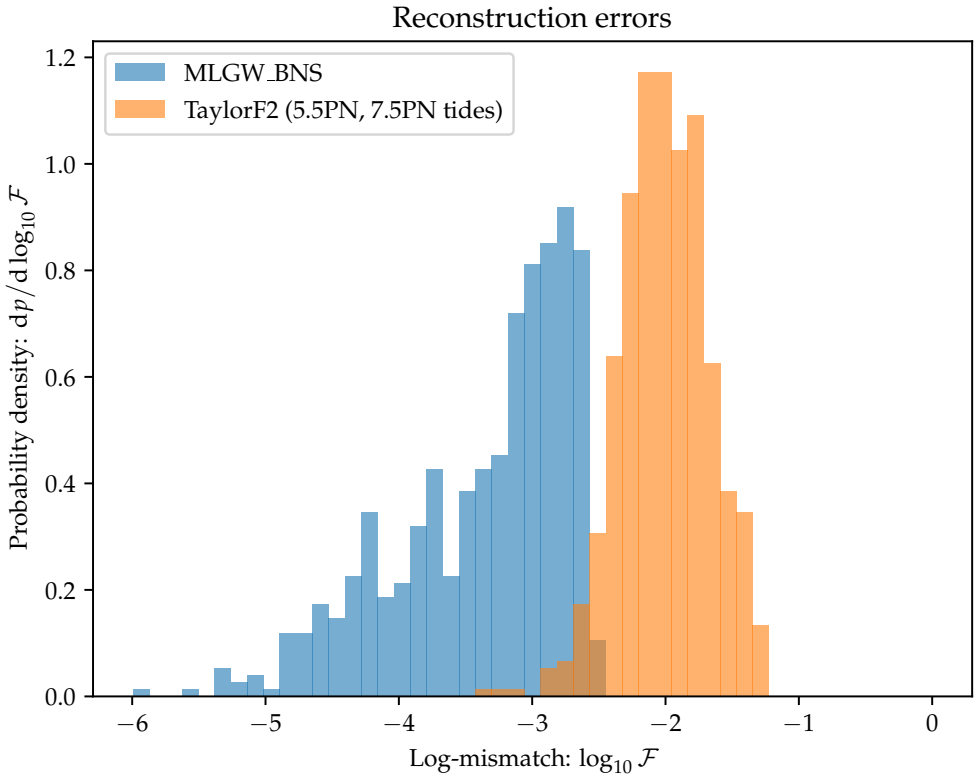


Figure 3.7: Mismatches between reconstructed and EOB waveforms in the spinning case; we also show the mismatches between the EOB waveforms and the original PN ones (i.e. we set the residuals to zero). All waveforms are reconstructed starting from 12 Hz, and the mismatches are computed according to the aLIGO PSD.

Section 1.3.2 discussed some order-of-magnitude estimates for the fidelity required for matched filtering and for parameter estimation.

Figure 3.7 shows the mismatches between the waveforms reconstructed by `mlgw_bns` and the reference ones generated by `TEOBResumS`.

Since the algorithm is based on PN waveforms, we also show the corresponding mismatches computed between the PN model and the EOB one. We can clearly see an improvement by a factor 10 or more — the histograms barely overlap, and the `mlgw_bns` has a much heavier tail into very low $\mathcal{F}$.

The corresponding amplitude and phase residuals are shown in figures 3.8 and 3.9.

Figure 3.10 shows how the mismatches computed with the aLIGO PSD change if we use another one. Specifically, they are almost unchanged with the predicted ET PSD, while they are lower with a reference, unphysical constant noise PSD.

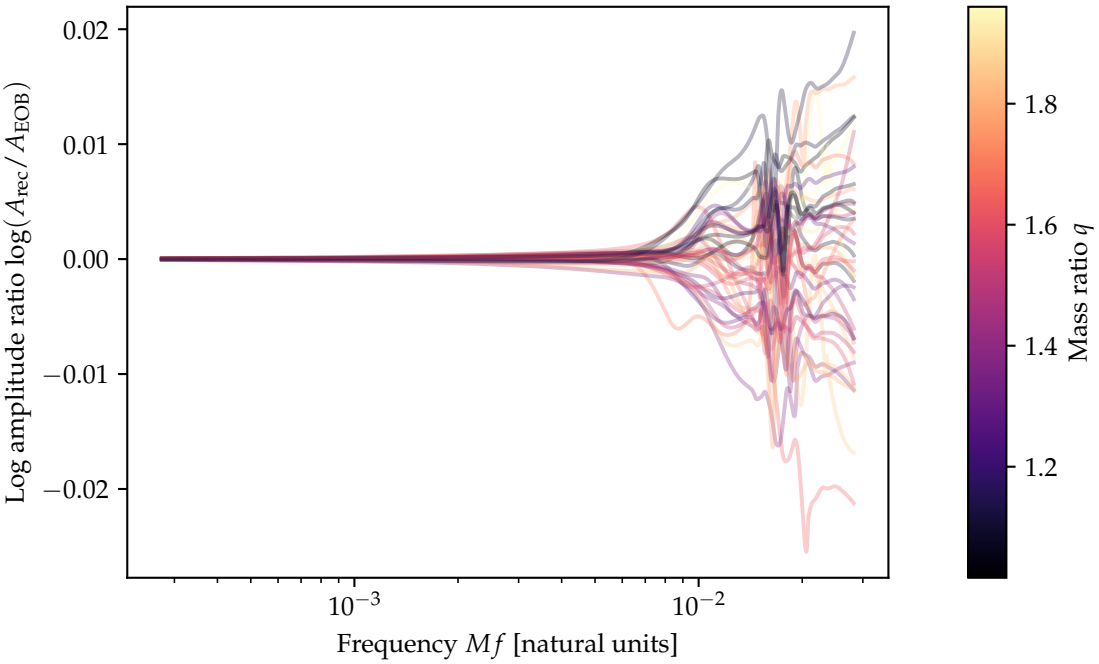


Figure 3.8: Log-amplitude reconstruction residuals for waveforms starting at 12 Hz.

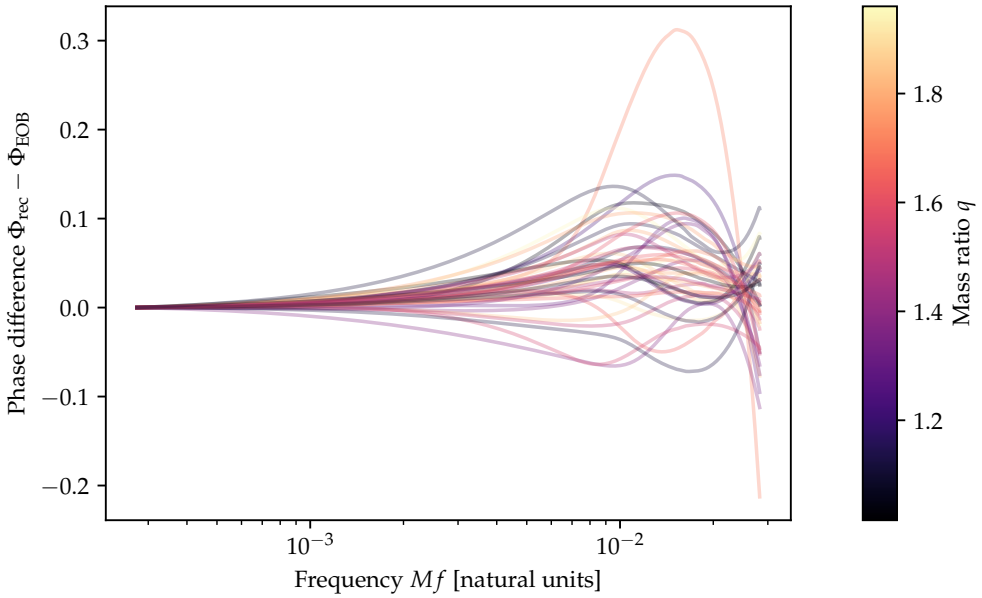


Figure 3.9: Phase reconstruction residuals for waveforms starting at 12 Hz. A linear fit is subtracted from all of these; this is arbitrary, however it shows the magnitude of the phase residuals for a close-to-optimal time alignment.

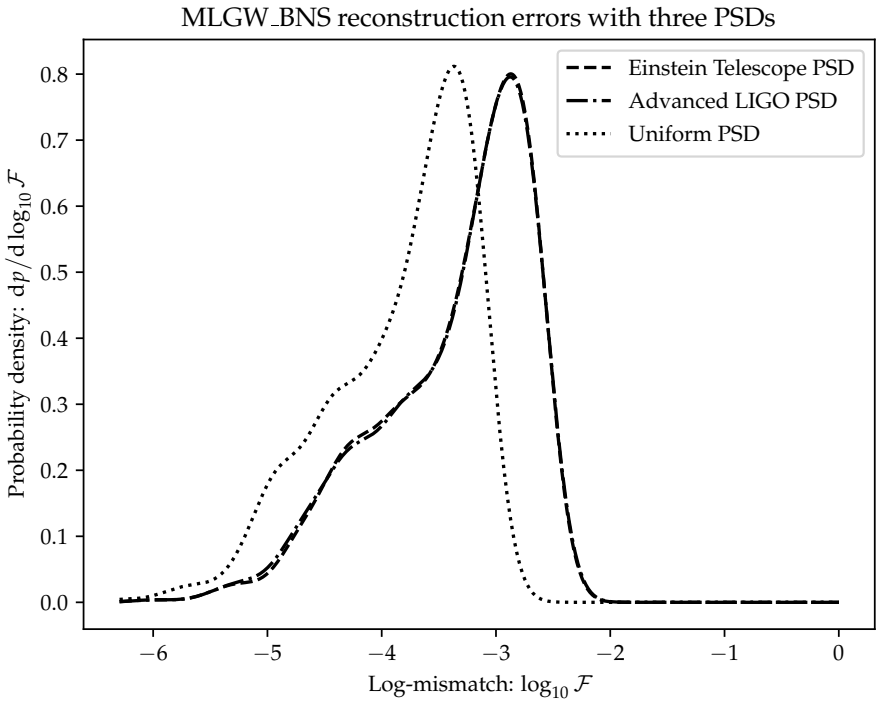


Figure 3.10: KDE plot of the mismatches between the same waveforms reconstructed by `mlgw_bns` as figure 3.7, varying the PSD: we use the aLIGO and ET PSDs, as well as a uniform $S_n \equiv$ const for reference.

## 3.7 Conclusions and outlook

We did this stuff, it could be improved by

# Appendix A

# The Stationary Phase Approximation

We can apply SPA to approximate the Fourier transform of an oscillatory function $f(t) = A(t)e^{-i\Phi(t)}$ as long as [CF98]

1. $\dot{\Phi}$ is monotonic;
2. the amplitude $A(t)$ is slowly changing compared to the phase: $\left|\dot{A}/A\right| \ll \left|\dot{\Phi}\right|$;
3. the phase derivative is slowly varying: $\left|\ddot{\Phi}\right| \ll \dot{\Phi}^2$.

If these conditions hold, then in the computation of the transform

$$\widetilde{f}(\omega) = \int_{-\infty}^{\infty} \mathrm{d}t\, A(t)e^{-i\Phi(t)+i\omega t} \tag{A.1}$$

the exponential term will be quickly oscillating, and therefore destructively interfering, almost everywhere; while near the time in which $\omega = \dot{\Phi}$ it will have a positive contribution.

With this in mind, we can approximate the integral: specifically, looking at the local integral contribution to the integral in figure A.1 we can see that it approaches a Gaussian,[1] which suggests expanding the argument of the exponential to second order around its stationary point,

$$-i\Phi(t) + i\omega t \approx \left(-i\Phi(t) + i\omega t\right)\Big|_{t_s} - \frac{i}{2}\,\ddot{\Phi}\Big|_{t_s}(t-t_s)^2\,, \tag{A.2}$$

where the stationary time $t_s$ is precisely the one such that $\dot{\Phi}(t_s) = \omega$ (which means that the linear term of the expansion vanishes).

The Fourier integral then looks like

$$\widetilde{f}(\omega) \approx \int_{-\infty}^{\infty} \mathrm{d}t\, A(t) \exp\left(-i\Phi(t_s) + i\omega t_s\right) \exp\left(-\frac{i}{2}\ddot{\Phi}(t-t_s)^2\right), \tag{A.3}$$

so the $t$-dependent part of the integral looks like the product of the amplitude and a "complex Gaussian". There is an analytic formula for a complex Gaussian integral

$$\int e^{i\alpha x^2}\,\mathrm{d}x = \sqrt{\frac{\pi i}{\alpha}}\,, \tag{A.4}$$

[1] The noise is mostly due to the window choice — a rectangular window is clearest for the visualization, but smoother choices yield less noise.

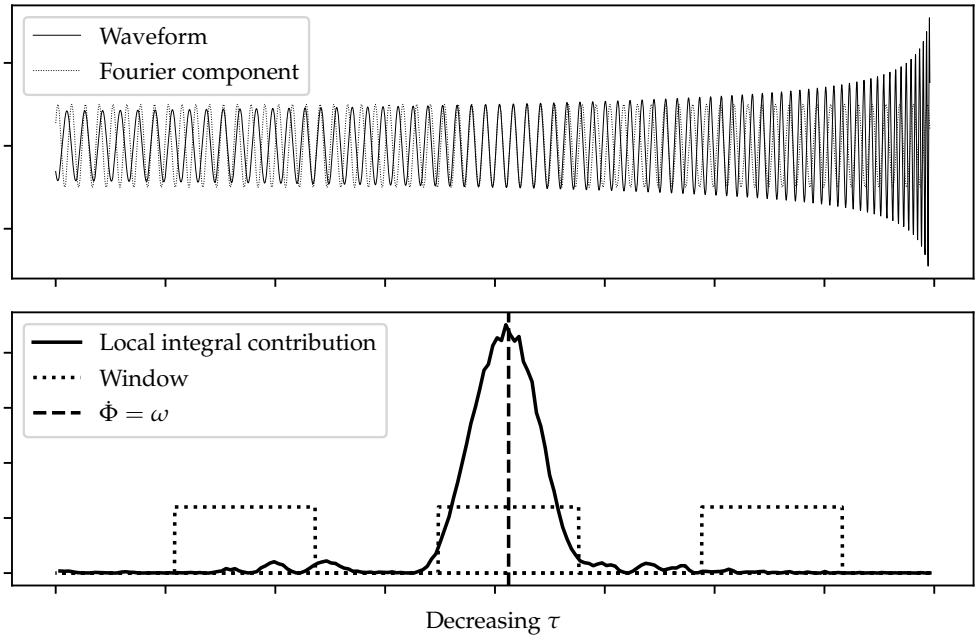


Figure A.1: Visualization of the idea behind SPA. In the upper plot we show a typical waveform for a CBC together with a sinusoid corresponding to the ones used in the computation of the Fourier transform. The lower plot shows a windowed Fourier-like integral, in the form $I(t_0) = \int h(t)\cos(\omega t)w(t-t_0)\,\mathrm{d}t$ for some window function $w$ centered on $t - t_0 = 0$, as well as three copies of the window. The value plotted with a solid line is $\left|I(t_0)\right|^2$. It can be clearly seen that in the regions where there is a frequency mismatch the contribution vanishes.

however, in order to apply it, we need to approximate the amplitude as constant: $A(t) \approx A(t_s)$. This will be valid as long as the variation of the amplitude is small in the width of the Gaussian we are integrating it against, which is $\sigma = 1/\sqrt{\ddot{\Phi}}$, so a condition we might write is $(\dot{A}/A)(1/\sqrt{\left|\ddot{\Phi}\right|}) \ll 1$. Notice that the combination of the conditions we imposed at the beginning ensures this.

If this is the case, we can write the Fourier integral as

$$\widetilde{f}(\omega) \approx A(t_s)e^{-i\Phi(t_s)+i\omega t_s}\sqrt{\frac{-2\pi i}{\ddot{\Phi}}} \tag{A.5}$$

$$\left|\widetilde{f}(\omega)\right| \approx A(t_s(\omega))\sqrt{\frac{2\pi}{\ddot{\Phi}(t_s(\omega))}} \tag{A.6}$$

$$\angle\widetilde{f}(\omega) \approx -\Phi(t_s(\omega)) + \omega t_s(\omega) - \frac{\pi}{4}, \tag{A.7}$$

where the $-\pi/4$ comes from the $\sqrt{-i}$.

We want to apply this to the specific case of $A$ and $\Phi$ being given by polynomial expressions like the formulas describing quadrupole emission: (1.75) and (1.73); for simplicity, let us write these as

$$\Phi(t) = K_\Phi(-t)^\alpha \qquad \text{and} \qquad A(t) = K_A(-t)^\beta\,, \tag{A.8}$$

where the reason for writing $(-t)$ is that the aforementioned equations are written in terms of the time until merger $\tau$ — $t = -\tau$ is increasing and equal to 0 at the merger. We do not specify the polarization, the argument will apply for both. The true values for the exponents in the quadrupole approximation are $\alpha = 5/8$ and $\beta = -1/4$.

The first thing to do is to find $t_s(\omega)$: it is given by

$$\omega = \dot{\Phi} = \alpha K_\Phi (-t_s)^{\alpha-1} \tag{A.9}$$

$$t_S = -\left(\frac{\omega}{\alpha K_\Phi}\right)^{1/(\alpha-1)} . \tag{A.10}$$

The other quantities we need to compute are:

$$A(t_s) = K_A \left(\frac{\omega}{\alpha K_\Phi}\right)^{\beta/(\alpha-1)} \tag{A.11}$$

$$\Phi(t_s) = K_\Phi \left(\frac{\omega}{\alpha K_\Phi}\right)^{\alpha/(\alpha-1)} \tag{A.12}$$

$$\ddot{\Phi}(t_s) = K_\Phi \alpha(\alpha-1)(-t_s)^{\alpha-2} = -K_\Phi \alpha(\alpha-1)\left(\frac{\omega}{\alpha K_\Phi}\right)^{(\alpha-2)/(\alpha-1)} , \tag{A.13}$$

therefore the phase reads

$$\angle\widetilde{f}(\omega) = -K_\Phi \left(\frac{\omega}{\alpha K_\Phi}\right)^{\alpha/(\alpha-1)} - \omega\left(\frac{\omega}{\alpha K_\Phi}\right)^{1/\alpha} - \frac{\pi}{4} \tag{A.14}$$

$$= -\left(\frac{\omega^\alpha}{K_\Phi}\left(\alpha^{-\alpha} + \alpha^{-1}\right)\right)^{1/(\alpha-1)} - \frac{\pi}{4} , \tag{A.15}$$

while the amplitude reads

$$\left|\widetilde{f}(\omega)\right| = K_A \left(\frac{\omega}{\alpha K_\Phi}\right)^{\beta/(\alpha-1)} \sqrt{2\pi} \underbrace{\left(K_\Phi \alpha(\alpha-1)\left(\frac{\omega}{\alpha K_\Phi}\right)^{(\alpha-2)/(\alpha-1)}\right)^{-1/2}}_{1/\sqrt{\ddot{\Phi}}} \tag{A.16}$$

$$= K_A (\alpha K_\Phi)^{\frac{\beta-1/2}{\alpha-1}} \omega^{\frac{\beta-(\alpha-2)/2}{\alpha-1}} \sqrt{\frac{2\pi}{\alpha-1}} . \tag{A.17}$$

The last complication is the fact that the phase appears in the expression for the waveform (1.75) not with a complex exponential, but instead with a cosine or a sine: this is not a problem, we just need to use the relations $\cos x = (e^{ix} + e^{-ix})/2$ and $\sin x = (e^{ix} - e^{-ix})/2i$. The factor of $i$ in the sine will correspond to a $\pi/2$ phase difference between the two polarizations, while the average of the two different exponentials will correspond to an amplitude reduction of a factor 2 if we restrict ourselves to positive frequencies.[2]

[2] Since the Fourier transform is linear the two exponentials can be treated separately. The transform of a real signal is fully encoded by its positive-frequency components; in order for the integral not to vanish in the Gaussian approximation there needs to be a stationary point; the phase of the GW signal is strictly monotonic. These three assumptions lead to the fact that only one of the two exponentials will give a nonzero contribution while the other will have no stationary point, so only the division by 2 will matter in the average.

Substituting the values for $\alpha$, $\beta$, $K_\Phi$ and $K_A$ we get the following expressions [Mag07, eqs. 4.34–37], written in terms of $f = \omega/2\pi$:

$$\widetilde{h}_+(f) = \frac{1}{\pi^{2/3}}\sqrt{\frac{5}{24}}\frac{c}{r}\left(\frac{G\mathcal{M}_c}{c^3}\right)^{5/6} f^{-7/6}\exp\left(\frac{3i}{4}\left(\frac{G\mathcal{M}_c}{c^3}8\pi f\right)^{-5/3} - i\frac{\pi}{4}\right)\left(\frac{1+\cos^2\iota}{2}\right) \tag{A.18}$$

$$\widetilde{h}_\times(f) = \frac{1}{\pi^{2/3}}\sqrt{\frac{5}{24}}\frac{c}{r}\left(\frac{G\mathcal{M}_c}{c^3}\right)^{5/6} f^{-7/6}\exp\left(\frac{3i}{4}\left(\frac{G\mathcal{M}_c}{c^3}8\pi f\right)^{-5/3} + i\frac{\pi}{4}\right)\cos\iota\,. \tag{A.19}$$

The frequency dependence of the amplitude comes out to be $\left|\widetilde{f}(\omega)\right| \propto \omega^{-7/6} \propto f^{-7/6}$: it might be surprising to see that this is decreasing while the chirping waveform rises in amplitude! The reason for this fact is that each portion of the Fourier integral is weighted not only by the amplitude of the envelope $A(t)$ but also by the time the waveform "spends" in that frequency region, which decreases with time. These two effects compete, but $A \sim \tau^{-1/4}$ while the time spent in the frequency region is measured by $1/\sqrt{\ddot{\Phi}} \sim 1/\sqrt{\tau^{-11/8}} = \tau^{+11/16}$, so the latter wins: $\left|\widetilde{h}(f)\right| \sim \tau^{7/16} \propto \omega^{-7/6}$.

# Appendix B

# Acronyms

**aLIGO** advanced LIGO

**ADM** Arnowitt, Deser, Misner

**BBH** binary black hole

**BH** black hole

**BNS** binary neutron star

**CBC** compact binary coalescence

**EOB** effective one-body

**EMRI** extreme mass ratio inspiral

**ET** Einstein telescope

**FD** Fourier domain

**FFT** fast Fourier transform

**IMR** Inspiral-Merger-Ringdown

**KDE** Kernel Density Estimate

**LAL** LIGO Algorithm Library

**LIGO** laser interferometric gravitational-wave observatory

**GR** general relativity

**GW** gravitational wave

**MCMC** Monte Carlo markov chain

**ML** machine learning

**MH** Metropolis-Hastings

**MOTPE** multi-objective tree-structured Parzen estimator

**NN** neural network

**NR** numerical relativity

**NS** neutron star

**PDF** probability density function

**PN** post-Newtonian

**PCA** principal component analysis

**PSD** power spectral density

**ROQ** reduced order quadratures

**SI** International System of Units

**TPE** tree-structured Parzen estimator

`mlgw_bns` machine learning gravitational waves from binary neutron stars

**SGWB** stochastic gravitational wave background

**SNR** signal-to-noise ratio (see section 1.3)

**SPA** stationary phase approximation (see appendix A)

**TT** transverse-traceless

# Contents

# Bibliography


[Abb+17] B. P. Abbott et al. "GW170817: Observation of Gravitational Waves from a Binary Neutron Star Inspiral".
In: *Physical Review Letters* 119.16 (Oct. 16, 2017), p. 161101.
ISSN: 0031-9007, 1079-7114. DOI: 10.1103/PhysRevLett.119.161101.
URL: https://link.aps.org/doi/10.1103/PhysRevLett.119.161101
(visited on 2021-04-04) (cit. on pp. 3, 15, 19, 30, 31, 35).

[Aji+11] P. Ajith et al. *Data Formats for Numerical Relativity Waves*. Feb. 24, 2011.
arXiv: 0709.0093 [gr-qc].
URL: http://arxiv.org/abs/0709.0093 (visited on 2021-03-03)
(cit. on p. 17).

[Aki+19] Takuya Akiba et al.
*Optuna: A Next-Generation Hyperparameter Optimization Framework*.
July 25, 2019. arXiv: 1907.10902 [cs, stat].
URL: http://arxiv.org/abs/1907.10902 (visited on 2021-04-30)
(cit. on p. 52).

[BYC13] J Bergstra, D Yamins, and D D Cox.
"Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures". In: (2013), p. 9
(cit. on p. 53).

[Ber+11] James Bergstra et al. "Algorithms for Hyper-Parameter Optimization".
In: *Advances in Neural Information Processing Systems*. Vol. 24.
Curran Associates, Inc., 2011. URL: https://papers.nips.cc/paper/2011/hash/86e8f7ab32cfd12577bc2619bc635690-Abstract.html (visited on 2021-08-15) (cit. on p. 52).

[BB21] Sebastiano Bernuzzi and Matteo Breschi. *Notes on Gravitational Waves*. 2021.
URL: http://sbernuzzi.gitpages.tpi.uni-jena.de/gw/ (cit. on p. 41).

[Bet11] Michael Betancourt.
"Nested Sampling with Constrained Hamiltonian Monte Carlo".
In: *AIP Conference Proceedings* 1305.1 (Mar. 14, 2011), pp. 165–172.
ISSN: 0094-243X. DOI: 10.1063/1.3573613.
URL: https://aip.scitation.org/doi/abs/10.1063/1.3573613 (visited on 2021-10-01) (cit. on p. 33).

[BDG20] Donato Bini, Thibault Damour, and Andrea Geralico. "Binary Dynamics at the Fifth and Fifth-and-a-Half Post-Newtonian Orders". In: (Mar. 26, 2020).
DOI: 10.1103/PhysRevD.102.024062.

URL: https://arxiv.org/abs/2003.11891v1 (visited on 2021-07-21) (cit. on p. 38).

[Bla14] Luc Blanchet. "Gravitational Radiation from Post-Newtonian Sources and Inspiralling Compact Binaries". In: *Living Reviews in Relativity* 17.1 (Dec. 2014), p. 2. ISSN: 2367-3613, 1433-8351. DOI: 10.12942/lrr-2014-2. URL: http://link.springer.com/10.12942/lrr-2014-2 (visited on 2021-04-04) (cit. on p. 38).

[BGB21] Matteo Breschi, Rossella Gamba, and Sebastiano Bernuzzi. *${\tt Bajes}$: Bayesian Inference of Multimessenger Astrophysical Data, Methods and Application to Gravitational-Waves*. Jan. 29, 2021. arXiv: 2102.00017 [astro-ph, physics:gr-qc]. URL: http://arxiv.org/abs/2102.00017 (visited on 2021-05-03) (cit. on pp. 18, 33, 58, 60).

[Can+15] Priscilla Canizares et al. "Accelerated Gravitational Wave Parameter Estimation with Reduced Order Modeling". In: *Physical Review Letters* 114.7 (Feb. 20, 2015), p. 071104. DOI: 10.1103/PhysRevLett.114.071104. URL: https://link.aps.org/doi/10.1103/PhysRevLett.114.071104 (visited on 2021-09-22) (cit. on p. 66).

[Car19] Sean M. Carroll. *Spacetime and Geometry: An Introduction to General Relativity*. New York: Cambridge University Press, 2019. 1 p. ISBN: 978-1-108-77038-5 (cit. on pp. 9, 10).

[CF98] E. Chassande-Mottin and P. Flandrin. "On the Stationary Phase Approximation of Chirp Spectra". In: *Proceedings of the IEEE-SP International Symposium on Time-Frequency and Time-Scale Analysis (Cat. No.98TH8380)*. International Symposium on Time-Frequency and Time-Scale Analysis. Pittsburgh, PA, USA: IEEE, 1998, pp. 117–120. ISBN: 978-0-7803-5073-1. DOI: 10.1109/TFSA.1998.721375. URL: http://ieeexplore.ieee.org/document/721375/ (visited on 2021-08-24) (cit. on p. 71).

[Cha+17] Katerina Chatziioannou et al. "Constructing Gravitational Waves from Generic Spin-Precessing Compact Binary Inspirals". In: *Physical Review D* 95.10 (May 5, 2017), p. 104004. DOI: 10.1103/PhysRevD.95.104004. URL: https://link.aps.org/doi/10.1103/PhysRevD.95.104004 (visited on 2021-10-01) (cit. on p. 30).

[Col18] LIGO Scientific Collaboration. *LIGO Algorithm Library*. LIGO Scientific Collaboration, 2018. DOI: 10.7935/GT1W-FZ16. URL: https://git.ligo.org/lscsoft/lalsuite (visited on 2021-08-27) (cit. on p. 40).

[DNT11] T. Damour, A. Nagar, and M. Trias.
"Accuracy and Effectualness of Closed-Form, Frequency-Domain Waveforms for Non-Spinning Black Hole Binaries".
In: *Physical Review D* 83.2 (Jan. 7, 2011), p. 024006.
ISSN: 1550-7998, 1550-2368. DOI: 10.1103/PhysRevD.83.024006.
arXiv: 1009.5998.
URL: http://arxiv.org/abs/1009.5998 (visited on 2021-09-28)
(cit. on pp. 29, 30).

[Dam14] Thibault Damour. *The General Relativistic Two Body Problem and the Effective One Body Formalism*. 2014. DOI: 10.1007/978-3-319-06349-2_5.
arXiv: 1212.3169 [gr-qc, physics:hep-th].
URL: http://arxiv.org/abs/1212.3169 (visited on 2021-08-28)
(cit. on p. 40).

[Ein16] Albert Einstein.
"Näherungsweise Integration Der Feldgleichungen Der Gravitation".
In: *Sitzungsberichte der Königlich Preußischen Akademie der Wissenschaften (Berlin* (Jan. 1, 1916), pp. 688–696.
URL: https://ui.adsabs.harvard.edu/abs/1916SPAW.....688E (visited on 2021-08-17) (cit. on p. 6).

[Fav11] Marc Favata. "The Gravitational-Wave Memory from Eccentric Binaries".
In: *Physical Review D* 84.12 (Dec. 6, 2011), p. 124013.
ISSN: 1550-7998, 1550-2368. DOI: 10.1103/PhysRevD.84.124013.
arXiv: 1108.3121.
URL: http://arxiv.org/abs/1108.3121 (visited on 2021-08-22)
(cit. on p. 22).

[Fav14] Marc Favata. *Systematic Parameter Errors in Inspiraling Neutron Star Binaries*.
Mar. 17, 2014. DOI: 10.1103/PhysRevLett.112.101101.
arXiv: 1310.8288 [astro-ph, physics:gr-qc].
URL: http://arxiv.org/abs/1310.8288 (visited on 2021-09-08)
(cit. on p. 60).

[Fav+21] Marc Favata et al. *Constraining the Orbital Eccentricity of Inspiralling Compact Binary Systems with Advanced LIGO*. Aug. 12, 2021.
arXiv: 2108.05861 [gr-qc].
URL: http://arxiv.org/abs/2108.05861 (visited on 2021-08-22)
(cit. on p. 22).

[Fin92] Lee Samuel Finn. *Detection, Measurement and Gravitational Radiation*.
Sept. 24, 1992. DOI: 10.1103/PhysRevD.46.5236. arXiv: gr-qc/9209010.
URL: http://arxiv.org/abs/gr-qc/9209010 (visited on 2021-08-23)
(cit. on p. 26).

[FH08] Eanna E. Flanagan and Tanja Hinderer. "Constraining Neutron Star Tidal Love Numbers with Gravitational Wave Detectors".
In: *Physical Review D* 77.2 (Jan. 22, 2008), p. 021502.
ISSN: 1550-7998, 1550-2368. DOI: 10.1103/PhysRevD.77.021502.
arXiv: 0709.1915.

URL: http://arxiv.org/abs/0709.1915 (visited on 2020-11-13)
(cit. on p. 21).

[GS16] Chad R. Galley and Patricia Schmidt. *Fast and Efficient Evaluation of Gravitational Waveforms via Reduced-Order Spline Interpolation*. Nov. 22, 2016.
arXiv: 1611.07529 [gr-qc].
URL: http://arxiv.org/abs/1611.07529 (visited on 2021-08-28)
(cit. on p. 53).

[GBN20] Rossella Gamba, Sebastiano Bernuzzi, and Alessandro Nagar.
*Fast, Faithful, Frequency-Domain Effective-One-Body Waveforms for Compact Binary Coalescences*. Nov. 30, 2020.
arXiv: 2012.00027 [astro-ph, physics:gr-qc].
URL: http://arxiv.org/abs/2012.00027 (visited on 2021-02-27)
(cit. on p. 41).

[Gam+21] Rossella Gamba et al.
"Waveform Systematics in the Gravitational-Wave Inference of Tidal Parameters and Equation of State from Binary Neutron Star Signals".
In: *Physical Review D* 103.12 (June 7, 2021), p. 124015.
ISSN: 2470-0010, 2470-0029. DOI: 10.1103/PhysRevD.103.124015.
arXiv: 2009.08467.
URL: http://arxiv.org/abs/2009.08467 (visited on 2021-09-08)
(cit. on p. 30).

[Gou07] Eric Gourgoulhon. *3+1 Formalism and Bases of Numerical Relativity*.
Mar. 6, 2007. arXiv: gr-qc/0703035.
URL: http://arxiv.org/abs/gr-qc/0703035 (visited on 2021-04-13)
(cit. on p. 39).

[Har+20] Charles R. Harris et al. "Array Programming with NumPy".
In: *Nature* 585.7825 (7825 Sept. 2020), pp. 357–362. ISSN: 1476-4687.
DOI: 10.1038/s41586-020-2649-2.
URL: https://www.nature.com/articles/s41586-020-2649-2 (visited on 2021-09-07) (cit. on p. 59).

[HT75] R. A. Hulse and J. H. Taylor. "Discovery of a Pulsar in a Binary System".
In: *The Astrophysical Journal Letters* 195 (Jan. 1, 1975), pp. L51–L53.
ISSN: 0004-637X. DOI: 10.1086/181708.
URL: http://adsabs.harvard.edu/abs/1975ApJ...195L..51H (visited on 2020-07-13) (cit. on p. 22).

[Ken07] Daniel Kennefick. *Traveling at the Speed of Thought*. 2007.
ISBN: 978-0-691-11727-0. URL: https://press.princeton.edu/books/hardcover/9780691117270/traveling-at-the-speed-of-thought (visited on 2021-08-17) (cit. on p. 6).

[KB17] Diederik P. Kingma and Jimmy Ba.
*Adam: A Method for Stochastic Optimization*. Jan. 29, 2017.
arXiv: 1412.6980 [cs].
URL: http://arxiv.org/abs/1412.6980 (visited on 2021-08-15)
(cit. on p. 51).

[Knu+15] Kevin H. Knuth et al. “Bayesian Evidence and Model Selection”. In: *Digital Signal Processing* 47 (Dec. 2015), pp. 50–67. ISSN: 10512004. DOI: 10.1016/j.dsp.2015.06.012. arXiv: 1411.3013. URL: http://arxiv.org/abs/1411.3013 (visited on 2021-09-15) (cit. on p. 31).

[Kum+15] Prayush Kumar et al. “Accuracy and Precision of Gravitational-Wave Models of Inspiraling Neutron Star – Black Hole Binaries with Spin: Comparison with Numerical Relativity in the Low-Frequency Regime”. In: *Physical Review D* 92.10 (Nov. 4, 2015), p. 102001. ISSN: 1550-7998, 1550-2368. DOI: 10.1103/PhysRevD.92.102001. arXiv: 1507.00103. URL: http://arxiv.org/abs/1507.00103 (visited on 2021-03-31) (cit. on p. 40).

[Lac+17] Benjamin D. Lackey et al. “Effective-One-Body Waveforms for Binary Neutron Stars Using Surrogate Models”. In: *Physical Review D* 95.10 (May 30, 2017), p. 104036. ISSN: 2470-0010, 2470-0029. DOI: 10.1103/PhysRevD.95.104036. arXiv: 1610.04742. URL: http://arxiv.org/abs/1610.04742 (visited on 2020-11-20) (cit. on p. 3).

[Lac+19] Benjamin D. Lackey et al. “Surrogate Model for an Aligned-Spin Effective One Body Waveform Model of Binary Neutron Star Inspirals Using Gaussian Process Regression”. In: *Physical Review D* 100.2 (July 3, 2019), p. 024002. ISSN: 2470-0010, 2470-0029. DOI: 10.1103/PhysRevD.100.024002. arXiv: 1812.08643. URL: http://arxiv.org/abs/1812.08643 (visited on 2021-02-26) (cit. on p. 54).

[LP07] James M. Lattimer and Maddapa Prakash. “Neutron Star Observations: Prognosis for Equation of State Constraints”. In: *Physics Reports* 442.1-6 (Apr. 2007), pp. 109–165. ISSN: 03701573. DOI: 10.1016/j.physrep.2007.02.003. arXiv: astro-ph/0612440. URL: http://arxiv.org/abs/astro-ph/0612440 (visited on 2020-09-26) (cit. on p. 20).

[LC21] Alexandre Le Tiec and Marc Casals. “Spinning Black Holes Fall in Love”. In: *Physical Review Letters* 126.13 (Mar. 30, 2021), p. 131102. DOI: 10.1103/PhysRevLett.126.131102. URL: https://link.aps.org/doi/10.1103/PhysRevLett.126.131102 (visited on 2021-08-21) (cit. on p. 21).

[LIG+16] LIGO Scientific Collaboration and Virgo Collaboration et al. “Observation of Gravitational Waves from a Binary Black Hole Merger”. In: *Physical Review Letters* 116.6 (Feb. 11, 2016), p. 061102. DOI: 10.1103/PhysRevLett.116.061102. URL: https://link.aps.org/doi/10.1103/PhysRevLett.116.061102 (visited on 2020-03-23) (cit. on p. 15).

[Lin84] L. Lindblom. “Limits on the Gravitational Redshift Form Neutron Stars”. In: *The Astrophysical Journal* 278 (Mar. 1, 1984), pp. 364–368. ISSN: 0004-637X. DOI: 10.1086/161800. URL: https://ui.adsabs.harvard.edu/abs/1984ApJ...278..364L (visited on 2021-08-20) (cit. on p. 20).

[LOB08] Lee Lindblom, Benjamin J. Owen, and Duncan A. Brown. “Model Waveform Accuracy Standards for Gravitational Wave Data Analysis”. In: *Physical Review D* 78.12 (Dec. 29, 2008), p. 124020. ISSN: 1550-7998, 1550-2368. DOI: 10.1103/PhysRevD.78.124020. arXiv: 0809.3844. URL: http://arxiv.org/abs/0809.3844 (visited on 2021-09-08) (cit. on pp. 29, 30).

[LL11] Ka-Wai Lo and Lap-Ming Lin. “The Spin Parameter of Uniformly Rotating Compact Stars”. In: *The Astrophysical Journal* 728.1 (Jan. 2011), p. 12. ISSN: 0004-637X. DOI: 10.1088/0004-637X/728/1/12. URL: https://doi.org/10.1088/0004-637x/728/1/12 (visited on 2021-08-20) (cit. on p. 19).

[Löf+12] Frank Löffler et al. “The Einstein Toolkit: A Community Computational Infrastructure for Relativistic Astrophysics”. In: *Classical and Quantum Gravity* 29.11 (June 7, 2012), p. 115001. ISSN: 0264-9381, 1361-6382. DOI: 10.1088/0264-9381/29/11/115001. arXiv: 1111.3344. URL: http://arxiv.org/abs/1111.3344 (visited on 2021-08-27) (cit. on p. 39).

[Mag07] Michele Maggiore. *Gravitational Waves: Volume 1: Theory and Experiments*. 1 edition. Oxford: Oxford University Press, Nov. 24, 2007. 576 pp. ISBN: 978-0-19-857074-5 (cit. on pp. 7, 10, 12–15, 17, 18, 21–23, 25, 26, 38, 39, 41, 74).

[Mag18] Michele Maggiore. *Gravitational Waves: Volume 2: Astrophysics and Cosmology*. Oxford: Oxford University Press, May 22, 2018. 848 pp. ISBN: 978-0-19-857089-9 (cit. on pp. 20, 21).

[MN17] Bryan F. J. Manly and Jorge A. Navarro Alberto. *Multivariate Statistical Methods: A Primer*. Fourth edition. Boca Raton: CRC Press, Taylor & Francis Group, 2017. 253 pp. ISBN: 978-1-4987-2896-6 (cit. on p. 45).

[MCB15] Christopher J. Moore, Robert H. Cole, and Christopher P. L. Berry. “Gravitational-Wave Sensitivity Curves”. In: *Classical and Quantum Gravity* 32.1 (Jan. 8, 2015), p. 015014. ISSN: 0264-9381, 1361-6382. DOI: 10.1088/0264-9381/32/1/015014. arXiv: 1408.0740. URL: http://arxiv.org/abs/1408.0740 (visited on 2020-04-19) (cit. on p. 27).

[NBR21] Alessandro Nagar, Alice Bonino, and Piero Rettegno.
*All in One: Effective One Body Multipolar Waveform Model for Spin-Aligned, Quasi-Circular, Eccentric, Hyperbolic Black Hole Binaries*. Jan. 26, 2021.
arXiv: 2101.08624 [gr-qc].
URL: http://arxiv.org/abs/2101.08624 (visited on 2021-02-26)
(cit. on p. 41).

[Nag+18] Alessandro Nagar et al.
*Time-Domain Effective-One-Body Gravitational Waveforms for Coalescing Compact Binaries with Nonprecessing Spins, Tides and Self-Spin Effects*.
Oct. 28, 2018. DOI: 10.1103/PhysRevD.98.104052.
arXiv: 1806.01772 [gr-qc].
URL: http://arxiv.org/abs/1806.01772 (visited on 2021-02-27)
(cit. on pp. 3, 4, 30, 41).

[Ned+20] Vsevolod Nedora et al.
*Numerical Relativity Simulations of the Neutron Star Merger GW170817: Long-Term Remnant Evolutions, Winds, Remnant Disks, and Nucleosynthesis*.
Aug. 12, 2020. arXiv: 2008.04333 [astro-ph, physics:gr-qc].
URL: http://arxiv.org/abs/2008.04333 (visited on 2020-11-16)
(cit. on p. 39).

[Nie15] Michael A. Nielsen. *Neural Networks and Deep Learning*.
Determination Press, 2015.
URL: http://neuralnetworksanddeeplearning.com (visited on 2021-03-23)
(cit. on pp. 48, 51).

[Owe96] Benjamin J. Owen. "Search Templates for Gravitational Waves from Inspiraling Binaries: Choice of Template Spacing".
In: *Physical Review D* 53.12 (June 15, 1996), pp. 6749–6761.
ISSN: 0556-2821, 1089-4918. DOI: 10.1103/PhysRevD.53.6749.
arXiv: gr-qc/9511032.
URL: http://arxiv.org/abs/gr-qc/9511032 (visited on 2021-09-28)
(cit. on p. 29).

[Oza+20] Yoshihiko Ozaki et al. "Multiobjective Tree-Structured Parzen Estimator for Computationally Expensive Optimization Problems".
In: *Proceedings of the 2020 Genetic and Evolutionary Computation Conference*. GECCO '20.
New York, NY, USA: Association for Computing Machinery, June 25, 2020, pp. 533–541. ISBN: 978-1-4503-7128-5. DOI: 10.1145/3377930.3389817.
URL: https://doi.org/10.1145/3377930.3389817 (visited on 2021-08-15)
(cit. on p. 52).

[Pea01] Karl Pearson.
"LIII. On Lines and Planes of Closest Fit to Systems of Points in Space".
In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 2.11 (Nov. 1, 1901), pp. 559–572. ISSN: 1941-5982.
DOI: 10.1080/14786440109462720. URL:
https://doi.org/10.1080/14786440109462720 (visited on 2021-08-13)
(cit. on p. 45).

[Ped+11] Fabian Pedregosa et al. “Scikit-Learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12.85 (2011), pp. 2825–2830. URL: http://jmlr.org/papers/v12/pedregosa11a.html (visited on 2021-08-14) (cit. on pp. 49, 51).

[PH20] Michael Pürrer and Carl-Johan Haster. “Gravitational Waveform Accuracy Requirements for Future Ground-Based Detectors”. In: *Physical Review Research* 2.2 (May 11, 2020), p. 023151. DOI: 10.1103/PhysRevResearch.2.023151. URL: https://link.aps.org/doi/10.1103/PhysRevResearch.2.023151 (visited on 2021-10-01) (cit. on p. 30).

[Rud16] Sebastian Ruder. “An Overview of Gradient Descent Optimization Algorithms”. In: (Sept. 15, 2016). URL: https://arxiv.org/abs/1609.04747v2 (visited on 2021-08-15) (cit. on p. 51).

[Sch+20] Stefano Schmidt et al. *Machine Learning Gravitational Waves from Binary Black Hole Mergers*. Nov. 3, 2020. arXiv: 2011.01958 [gr-qc]. URL: http://arxiv.org/abs/2011.01958 (visited on 2020-11-13) (cit. on pp. 3, 4, 48).

[SS06] Devinderjit Sivia and John Skilling. *Data Analysis: A Bayesian Tutorial*. Oxford University Press, June 2006. 259 pp. ISBN: 978-0-19-856831-5. Google Books: lYMSDAAAQBAJ (cit. on p. 33).

[Ski06] John Skilling. “Nested Sampling for General Bayesian Computation”. In: *Bayesian Analysis* 1.4 (Dec. 2006), pp. 833–859. ISSN: 1936-0975, 1931-6690. DOI: 10.1214/06-BA127. URL: https://projecteuclid.org/journals/bayesian-analysis/volume-1/issue-4/Nested-sampling-for-general-Bayesian-computation/10.1214/06-BA127.full (visited on 2021-09-15) (cit. on pp. 31, 33, 34).

[Smi+16] Rory Smith et al. “Fast and Accurate Inference on Gravitational Waves from Precessing Compact Binaries”. In: *Physical Review D* 94.4 (Aug. 15, 2016), p. 044031. ISSN: 2470-0010, 2470-0029. DOI: 10.1103/PhysRevD.94.044031. arXiv: 1604.08253. URL: http://arxiv.org/abs/1604.08253 (visited on 2021-09-14) (cit. on p. 66).

[TW82] J. H. Taylor and J. M. Weisberg. “A New Test of General Relativity - Gravitational Radiation and the Binary Pulsar PSR 1913+16”. In: *The Astrophysical Journal* 253 (Feb. 1, 1982), pp. 908–920. ISSN: 0004-637X. DOI: 10.1086/159690. URL: http://adsabs.harvard.edu/abs/1982ApJ...253..908T (visited on 2020-07-13) (cit. on p. 22).

[Tea11] ET Science Team.
*Einstein Gravitational Wave Telescope Conceptual Design Study*. 2011.
URL: http://www.et-gw.eu (cit. on p. 24).

[VVM17] Serena Vinciguerra, John Veitch, and Ilya Mandel.
"Accelerating Gravitational Wave Parameter Estimation with Multi-Band Template Interpolation".
In: *Classical and Quantum Gravity* 34.11 (June 8, 2017), p. 115006.
ISSN: 0264-9381, 1361-6382. DOI: 10.1088/1361-6382/aa6d44.
arXiv: 1703.02062.
URL: http://arxiv.org/abs/1703.02062 (visited on 2021-09-14)
(cit. on p. 66).

[Wal09] Robert M. Wald. *Introduction to Gravitational Self-Force*. July 2, 2009.
arXiv: 0907.0412 [gr-qc].
URL: http://arxiv.org/abs/0907.0412 (visited on 2021-06-09)
(cit. on p. 37).

[ZDV18] Barak Zackay, Liang Dai, and Tejaswi Venumadhav. *Relative Binning and Fast Likelihood Evaluation for Gravitational Wave Parameter Estimation*.
Aug. 2, 2018. arXiv: 1806.08792 [astro-ph, physics:gr-qc].
URL: http://arxiv.org/abs/1806.08792 (visited on 2021-09-14)
(cit. on p. 66).