

Gravitational Waveforms for Extreme Mass Ratio Inspirals (EMRIs)

An Honours Thesis

submitted in partial fulfilment of the requirements for the degree of
Bachelor of Science with Honours in Physics

presented to
the Department of Physics
Faculty of Science
National University of Singapore
A/P Alvin Chua, Supervisor



by
Irwin Tay
A0199200J

April 2023

This Honours Thesis represents my own work and due acknowledgement is given whenever information is derived from other sources. No part of this Honours Thesis has been or is being concurrently submitted for any other qualification at any other university.

Signed: _____

Acknowledgement

I am honored to extend my heartfelt appreciation to my esteemed supervisor, **A/P Alvin Chua**, whose unparalleled guidance, unwavering support, and extensive expertise have been integral in shaping my research and enabling the successful completion of this thesis. My gratitude also extends to **Soichiro** and **Josh**, both postdoctoral fellows in the NUS Gravity Group, whose generosity in sharing their skills, knowledge, and enthusiasm has been invaluable in my academic growth and development.

Moreover, I am deeply grateful to my **parents**, whose unconditional love, encouragement, and sacrifices have been instrumental in supporting me throughout my academic journey. To my **friends**, I am thankful for their constant friendship, humor, and unwavering moral support, as well as useful insights, which have been a constant source of motivation and inspiration for me. Together, their collective contributions have enriched my life in countless ways, motivating me to pursue academic excellence with passion, determination, and gratitude.

Abstract

The upcoming LISA space mission will detect sources of gravitational waves in the millihertz frequency range. Among these sources, the extreme mass ratio inspirals (EMRIs) of compact objects into supermassive black holes pose a challenge due to their long-lived signals and the complexity of their strong fields. The data obtained from LISA requires a large number of accurate waveform templates for comparison in order to detect and characterise the signals. Therefore, the generation of accurate waveform templates is crucial in reducing the time required for LISA data analysis. We generate adiabatic waveforms for EMRIs by combining waveform snapshots of each geodesic orbit to form a long inspiral waveform. We use the multipolar expansion method, which demonstrates that the complex total waveform can be decomposed into a sum of individual harmonic modes, each of which evolves simply over long timescales. This characteristic can be used to efficiently produce waveform models that accurately capture the properties of EMRI systems. We provide an example of an EMRI waveform for a circular equatorial Kerr orbit using three different fitting methods: deep neural network, interpolation, and polynomial fitting. We also compare the accuracy and efficiency of these three methods for generating a full evolving EMRI waveform, and determine that, within the scope of a circular equatorial Kerr orbit, interpolation is sufficiently fast to generate waveform templates quickly enough to be used directly in LISA analysis algorithms. This specific waveform template can also be used as a reference for our generic model in the limits of a circular and equatorial orbit.

Contents

Acknowledgement	v
Abstract	vii
Contents	ix
List of Figures	xi
1 Introduction	1
1.1 Background	1
1.2 LISA	2
1.3 Extreme Mass Ratio Inspiral (EMRI)	3
1.4 Modelling Techniques	4
1.5 Motivation and Purpose	6
1.6 Thesis Outline	7
2 Theory	9
2.1 Perturbation in General Relativity	9
2.2 Bound Kerr Geodesics	13
2.3 Teukolsky Formalism	15
2.4 Full Waveform Evolution	20
3 Methodology	23
3.1 Generation of Data - GRELIM	23
3.2 Implementation of Neural Network	24
3.3 Implementation of Interpolation	25
3.4 Implementation of Polyfit	26
3.5 Data Formats	27
3.6 Error Metrics	31
4 Results	33

4.1	Neural Network	33
4.2	Interpolation	41
4.3	Polyfit	45
4.4	Mismatch of the Three Methods	48
4.5	Final Waveform Evolution	49
4.6	Computation Speed	50
5	Discussion	51
5.1	Results from the Neural Network	51
5.2	Results from Interpolation	52
5.3	Results from Polyfit	53
5.4	Full Waveform Evolution	53
5.5	General Discussion	53
5.6	Future Work	54
6	Conclusion	57
A	0PA and 1PA Adiabatic Waveform	59
A.1	How Accurate is Enough?	59
B	Proofs for Theory	61
B.1	Proofs for Eqn 2.7-2.9	61
C	Python Code	63
C.1	Script Used to Generate Data	63
C.2	Neural Network	66
C.3	Mode-Distribution Error Code	67
C.4	Mismatch Code	67
	Bibliography	69

List of Figures

1.1	An artist’s rendition of two black holes merging. Illustration credit: Simulating eXtreme Spacetimes (SXS) Project.	1
1.2	Strain and Detectors	2
1.3	Different Modelling Techniques in GR	4
2.1	Waveform decomposition	19
3.1	Illustration of how K-Fold validation works	25
3.2	Transformation graph of u to p	30
4.1	Neural network loss plot for the toy dataset	33
4.2	Teukolsky amplitudes in mode space index for comparison between NN and toy dataset.	34
4.3	Neural network loss plot for the 20×21 training dataset.	35
4.4	Teukolsky amplitudes in mode space for comparison between NN and the 20×14 test dataset B.	36
4.5	Teukolsky amplitudes in parameter space for comparison between NN and the 20×14 test dataset B	36
4.6	The mode-distribution error of NN and test dataset B	37
4.7	Teukolsky amplitudes in mode space for a particular worst-performing parameter to compare NN and test dataset B.	37
4.8	Neural network loss plot for the 10×30 final training dataset.	38
4.9	Teukolsky amplitudes in mode space for comparison between NN and the final test set.	39
4.10	Teukolsky amplitudes in parameter space for comparison between NN and the final test set	39
4.11	Mode-distribution error of NN and the final test set	40
4.12	Amplitude mismatch between the NN-fitted waveform and the final test set waveform	40

4.13	Waveform snapshot constructed by using amplitudes of the NN as well as a comparison with the waveform snapshot constructed by the final test set's amplitudes	41
4.14	Teukolsky amplitudes in parameter space constructed by interpolating the amplitudes in the training set. This is compared to test set A.	42
4.15	Mode-distribution error of the interpolation and test dataset A.	42
4.16	Teukolsky amplitudes in parameter space constructed by interpolating the amplitudes in the final training set. This is compared with the final test set.	43
4.17	Mode-distribution error of the interpolation and the final test dataset.	44
4.18	Amplitude mismatch between interpolated data and the final test set waveform	44
4.19	Waveform snapshot constructed using amplitudes from the interpolation and a comparison with the waveform snapshot using the test set's amplitudes is made.	45
4.20	Matrix plot of parameter space for the Teukolsky real amplitude.	46
4.21	Teukolsky amplitudes in parameter space constructed by polyfitting the amplitudes. This is compared with the final test set.	46
4.22	Mode-distribution error of the polyfit and final test set	47
4.23	Amplitude mismatch between polyfitted waveform and the final test set waveform	47
4.24	Waveform snapshot constructed using amplitudes from polyfit and a comparison with the waveform snapshot using the test set's amplitudes is done	48
4.25	The mismatch of the snapshot waveform across the parameter space for the three different methods - NN (blue), Interpolation (orange), and Polyfit (green).	48
4.26	Full evolving waveform constructed using the three methods.	49
4.27	Wall time of the three different methods proposed for generating amplitudes.	50

1. Introduction

1.1 Background

In a distant galaxy far away from Earth, there was a spectacular astronomical event that took place a long time ago. Two black holes, with their massive gravitational fields, orbited each other with fierce intensity, each determined to collide and unleash a cataclysmic amount of energy. This energy, in the form of gravitational waves, rippled through the very fabric of spacetime, waiting to be discovered.

For years, this incredible event went unnoticed until September 14, 2015. The signal of this incredible collision was detected on Earth by two detectors [1, 2]: the Laser Interferometer Gravitational-Wave Observatory (LIGO), ushering in the dawn of a new era of gravitational wave astronomy.

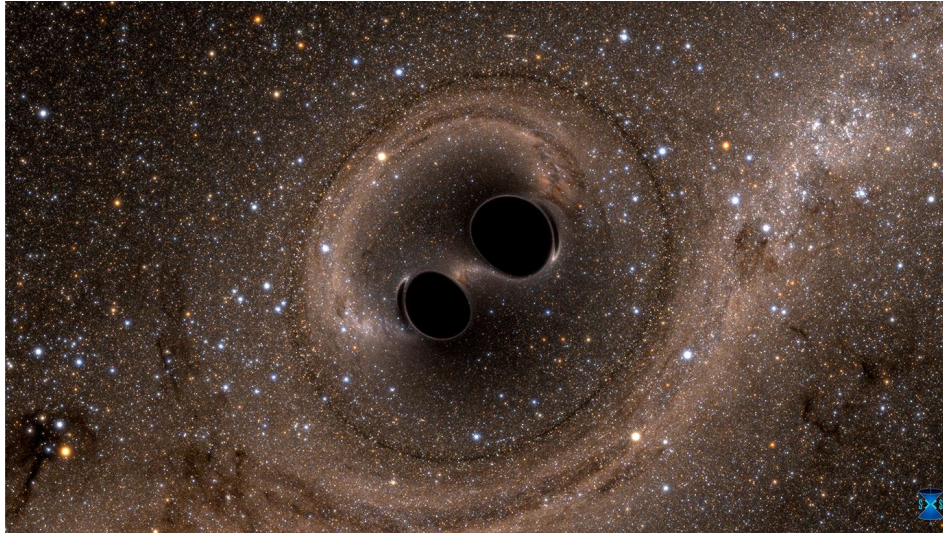


Figure 1.1: An artist's rendition of two black holes merging. Illustration credit: Simulating eXtreme Spacetimes (SXS) Project.

From that moment on, scientists could finally begin to explore and study these incredible waves, which have the power to reveal a wealth of information about our universe, including the nature of black holes, the evolution of galaxies, and the fundamental laws of physics. The discovery of gravitational waves has opened up a whole new world of possibilities and has given us an unprecedented glimpse into the mysteries of the cosmos.

Although the concept of gravitational waves is not new, there has been much theoretical work before the 2015 observation by LIGO. However, to this day, one of the biggest challenges in General Relativity (GR) is providing a closed-form solution to the gravitational two-body problem [3].

1.2 LISA

The Laser Interferometer Space Antenna (LISA) is a proposed space-based observatory that will consist of three spacecraft arranged in a triangle, with each spacecraft separated by millions of kilometres. Compared to our ground-based detectors, LISA has access to an abundance of sources from our cosmos, including massive binaries, galactic binaries, as well as extreme mass ratio binaries. This is because LISA is designed to detect gravitational waves in the millihertz frequency range, within which the gravitational waves produced by those sources fall, as seen in **Figure 1.2**. The mission aims to study a wide range of astrophysical phenomena, including the coalescence of supermassive black holes, the formation of binary star systems, and the earliest moments of the universe.

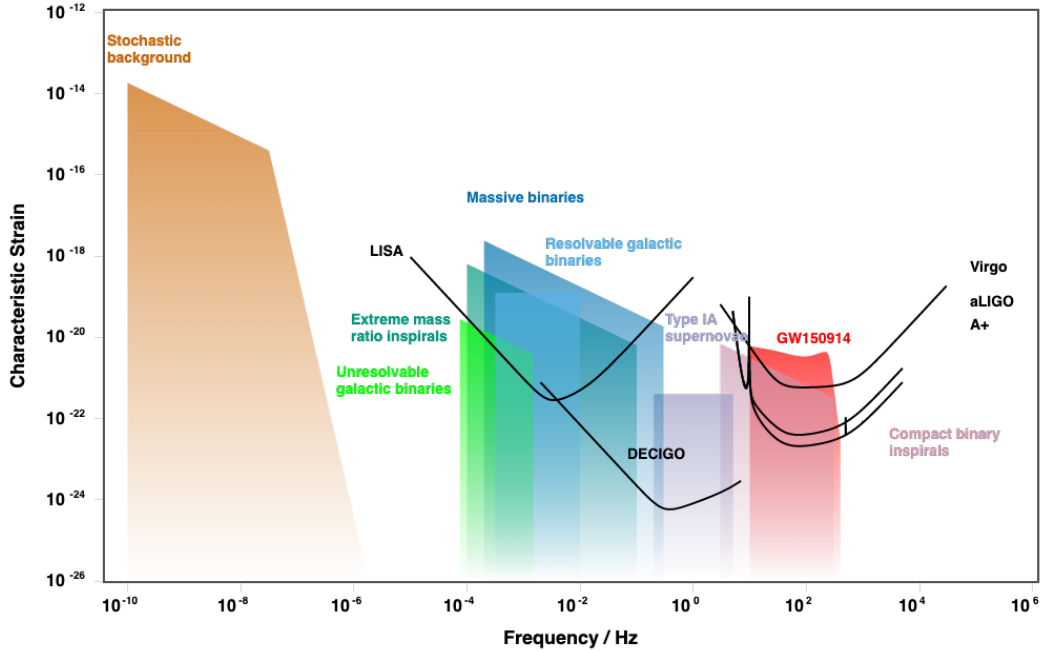


Figure 1.2: The sensitivity plot for the different detectors. We can see that LISA falls primarily in the millihertz region where the gravitational waveforms are primarily emitted by EMRIs. Credits: <http://gwplotter.com/>

As LISA is expected to be sensitive to the gravitational waves produced by extreme mass

ratio inspirals (EMRIs), which are one of the most promising sources of gravitational waves in this frequency range, the study of EMRIs is closely related to LISA. The detection and characterisation of these sources will be one of the primary scientific goals of the observatory. However, without a proper waveform template, it is difficult for LISA to search for and characterise these signals. With the launch planned for 2037 [4], researchers are expected to develop modelling work for LISA before the mission's launch.

1.3 Extreme Mass Ratio Inspiral (EMRI)

Located at the core of most galaxies, including our own Milky Way, there exists an EMRI source due to the presence of a supermassive black hole. We will now focus on EMRI, which is an important process in astrophysics and cosmology. It involves a compact object (CO), such as a black hole (BH) of $10M_{\odot}$ or a neutron star, spiralling into a much larger object, such as a supermassive black hole (SMBH) of $10^6 M_{\odot}$, due to the emission of gravitational waves. This phenomenon provides a unique opportunity to test the predictions of GR in the strong-field regime, probe the properties of SMBHs, and contribute to the emerging field of gravitational wave astronomy.

To achieve the ambitious scientific objectives for EMRI measurements, it is crucial to create waveform models or templates that can accurately match the signals present in the data throughout their entire duration. These templates provide guidance to algorithms that locate EMRI signals in detector noise and help characterise astrophysical sources.

The nature of EMRIs poses difficult and challenging problems for researchers today. This is due to their characteristics. For EMRIs, the inspirals can take a very long time before the CO plunges into the SMBH (between 10^4 and 10^5 strong field orbits [5]). This makes modelling EMRIs a non-trivial task. Additionally, it is already non-trivial to solve a GR two-body problem. However, it is precisely because EMRIs are long-lived that they can be studied extensively. Another challenge is that even the most luminous EMRI signals are expected to be weaker than the instrumental noise of LISA in its current design [6]. Therefore, extracting these signals would require filtering them against precise theoretical waveform templates. To fully access the valuable scientific information embedded within the signals, a comprehensive model of the radiative evolution and emitted radiation in EMRI systems is essential.

1.4 Modelling Techniques

When modelling general relativity (GR), there are several approaches, which are illustrated graphically in **Figure 1.3**:

1. Post-Newtonian (PN) theory
2. Numerical relativity (NR)
3. Black hole perturbation theory/Gravitational self-force (GSF)

PN theory has proven successful in modelling the evolution of binaries when the bodies are widely separated [7], but it breaks down when the bodies are close. Numerical relativity (NR) is required to accurately evolve and understand the dynamics of compact binaries in the strong field regime. While NR can theoretically solve the two-body system numerically, the time required to compute the results is a limiting factor. In the latest NR research, computing the results for a $1/128$ mass ratio binary for 13 orbits took 41250 node hours [8]. If run on a normal computer, the simulation would take at least one year to complete. Within NR, there are multiple methods for investigating the gravitational waveform emitted from binary systems, such as surrogate modelling, where PN approximations are used in the earlier parts of the inspiral and the surrogate model during the last few inspirals. However, this work does not explore such methods.

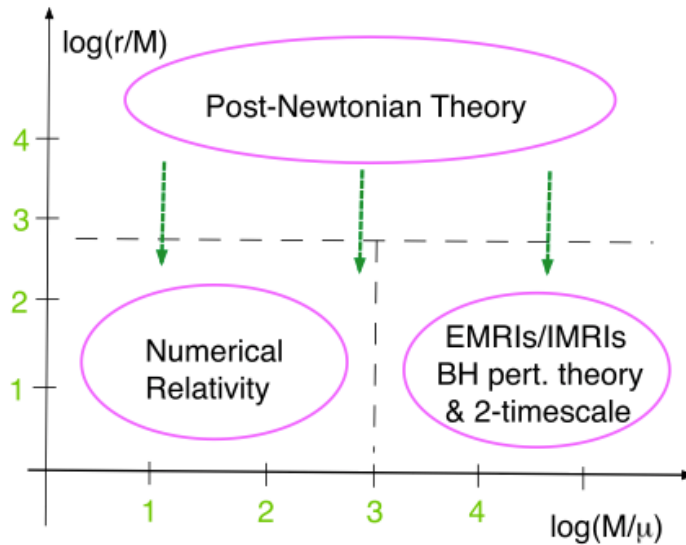


Figure 1.3: The parameter space of inspiralling compact binaries in GR, in terms of the inverse mass ratio $M/\mu = 1/\epsilon$ and the orbital radius r , showing the different regimes and the modelling techniques required in each regime. Individual binaries evolve downwards in the diagram (green dashed arrows). Credit: [9]

Therefore, we turn to black hole perturbation theory or gravitational self-force (GSF) theory. In the limit of mass ratios, researchers can solve the two-body problem with high accuracy using current tools. This precision is possible precisely in the case of extreme mass-ratio inspirals (EMRIs). Currently, one of the primary goals of GSF theory research is to create EMRI waveform models, which involve formulating equations that describe the motion of objects in specified background spacetimes and the objects' interactions with their own perturbations to the spacetime. The theory utilizes tools from black hole perturbation theory to develop GSF, where the mass ratio $\mu/M = \epsilon$ (where μ is the mass of the compact object and M is the mass of the supermassive black hole) serves as a parameter for counting perturbative order.

However, GSF approaches are still ongoing, and there is not yet a concrete analytical answer. Therefore, we turn to adiabatic waveforms from EMRIs.

1.4.1 Adiabatic Waveform, 0PA Leading-Order

EMRI waveforms inspiral very 'slowly', and because the inspirals are slow, we can approximate them adiabatically. The adiabatic approximation is used to model the slow change in the orbit of a CO caused by gravitational radiation. According to this approximation, the orbit is thought to evolve gradually in comparison to the period of time during which gravitational radiation is released. As such, one can interpret it as a two-timescale expansion [9]. This means that the orbital parameters (such as the semi-latus rectum axis and eccentricity) change gradually over many orbits and can be approximated as constant in each geodesic. Hence, the timescale at which the CO inspirals and the timescale at which it orbits can be treated differently. Therefore, one can think of the system evolving slowly from one geodesic to the next. For the purpose of detecting a sufficiently 'luminous' EMRI, the leading-order (also known as 0PA) of this adiabatic approximation may well suffice [3].

The adiabatic waveform can be obtained by solving the equations of motion for the orbiting CO in the background spacetime of the supermassive black hole, taking into account the emission of gravitational radiation or also known as the backreaction that arises from the orbit-averaged self-interaction. The gravitational wave signal emitted by the system can be obtained from the Teukolsky equation [10] and can be expressed as a sum over harmonics, each of which corresponds to a particular frequency and amplitude. The amplitude and

frequency of each harmonic depend on the orbital parameters and the mass and spin of the black holes. This method is also known as the multipolar expansion. It is necessary to know the orbital parameters as a function of time. These parameters can be obtained by integrating the equations of motion numerically, taking into account the energy and angular momentum loss due to gravitational radiation. The resulting waveform is a theoretical prediction of what the gravitational wave signal from an EMRI system would look like in an adiabatic approximation.

However, the adiabatic waveform only takes into account the leading-order of the adiabatic approximation. To consider the next order, the waveform is now called a post-1-adiabatic waveform, or also a 1PA waveform. Though 1PA waveforms are more accurate, they have not been realised yet as they are still research in progress. A deeper analysis of the accuracy of the adiabatic approximation is discussed further in **Appendix A**. For the purpose of our project, we will only be dealing with the 0PA adiabatic waveform.

1.5 Motivation and Purpose

In this work, we use the leading-order of the adiabatic approximation derived from the Teukolsky equation. To construct a full waveform trajectory, we employ the PN approximation to our adiabatic waveform to trace the change in parameters over time for our adiabatic waveform.

Our project aims to create accurate and efficient snapshot waveform templates to test how they behave after being stitched together to produce a full evolving EMRI waveform for a specific EMRI system: a circular and equatorial orbit system. Although progress has been made in simulating generic EMRI waveforms using the adiabatic approximation over the past few decades [5, 11], there are still very few of these waveforms being simulated. Most are either equatorial or spherical orbits, with none considering circular and equatorial orbits.

Furthermore, it is worthwhile to investigate different model fitting methods on a simple case before moving on to a generic case. Having a specific case allows us to check our generic models. By using the limits of a circular equatorial orbit for our generic model, we can test if it converges to our model as well.

The main challenge in creating a model for adiabatic waveform is the high-dimensional

parameter space of generic EMRI [11]. In addition, for a generic EMRI, the number of mode amplitudes needed to construct an accurate waveform is very large (approximately $10^4 - 10^5$) [12]. Therefore, it is computationally expensive to construct this many modes for an entire full waveform evolution.

Hence, our goal is to construct waveforms to a desirable accuracy and in a computationally efficient manner. To achieve this, we employ several methods to fit the amplitudes:

1. Neural Network (NN)
2. Spline Interpolation
3. Polynomial Fitting (Polyfit)

Afterwards, we compare the error and computation time of each method to determine the most effective one.

1.6 Thesis Outline

The following is an outline of this thesis:

1. **Introduction** (Chapter 1) - A brief introduction to the project and some background information.
2. **Theory** (Chapter 2) - A review of relevant literature and how it relates to this project.
3. **Methodology** (Chapter 3) - An explanation of the steps taken in this project and an introduction to the error metric used.
4. **Results** (Chapter 4) - Presentation of the study's findings.
5. **Discussion** (Chapter 5) - Analysis of the results and suggestions for future work.
6. **Conclusion** (Chapter 6) - Concluding remarks on the project.

2. Theory

2.1 Perturbation in General Relativity

When modelling EMRIs, the Post-Newtonian approach is not applicable due to their highly relativistic nature. Although the use of numerical relativity (NR) theory is theoretically possible, it is computationally expensive due to the large number of cycles/orbits required to model EMRIs. Therefore, black hole perturbation theory/gravitational self-force (GSF) theory is used to understand the mechanics behind EMRIs.

To begin, consider a compact object of small mass μ moving in a spacetime with a much larger mass M . Suppose that the exact metric $\tilde{g}_{\alpha\beta}(\epsilon)$ depends on ϵ . To approximate $\tilde{g}_{\alpha\beta}$ in the limit of $\mu/M \ll 1$, Taylor expansion can be used. The metric can then be expressed as:

$$\tilde{g}_{\alpha\beta} = g_{\alpha\beta} + \epsilon h_{\alpha\beta}^{(1)} + \epsilon^2 h_{\alpha\beta}^{(2)} + O(\epsilon^3) \quad (2.1)$$

where ϵ is introduced as a formal expansion parameter of which $\epsilon = \mu/M$. The zeroth-order term in **Eq. (2.1)**, $g_{\alpha\beta}$, is referred to as the background metric, which in the case of EMRI is also referred as the metric of the large black hole. The correction $h_{\alpha\beta}^{(n)}$ refers to the gravitational perturbations created by the small object where $h_{\alpha\beta}^{(n)} = \frac{1}{n!} \frac{d^n}{d\epsilon^n} \tilde{g}_{\alpha\beta} \big|_{\epsilon=0}$. We define:

$$h_{\alpha\beta} \equiv \tilde{g}_{\alpha\beta} - g_{\alpha\beta} = \sum_{n \geq 1} \epsilon^n h_{\alpha\beta}^{(n)} \quad (2.2)$$

Therefore, **Eq. (2.1)** can then be expressed as:

$$\tilde{g}_{\alpha\beta} = g_{\alpha\beta} + h_{\alpha\beta} \quad (2.3)$$

The metric must satisfy the Einstein equation:

$$G_{\alpha\beta}[\tilde{g}] = 8\pi T_{\alpha\beta} \quad (2.4)$$

where $G_{\mu\nu}[\tilde{g}]$ is the Einstein tensor of spacetime and $T_{\alpha\beta}$ is the stress-energy tensor of the

system's matter content. For simplicity, we suppose that the small object represents the only matter such that the stress-energy tensor $T_{\alpha\beta}$ is of the small object itself. Similarly, the stress-energy tensor can also be expanded as:

$$T_{\alpha\beta} = \epsilon T_{\alpha\beta}^{(1)} + \epsilon^2 T_{\alpha\beta}^{(2)} + O(\epsilon^3) \quad (2.5)$$

Substituting **Eq. (2.1)** into the Einstein equation, we would obtain:

$$G_{\alpha\beta}[\tilde{g}] = G_{\alpha\beta}[g] + \epsilon \delta G_{\alpha\beta}[h^{(1)}] + \epsilon^2 \left(\delta G_{\alpha\beta}[h^{(2)}] + \delta^2 G_{\alpha\beta}[h^{(1)}] \right) + O(\epsilon^3) \quad (2.6)$$

where $\delta G_{\alpha\beta}[h^{(n)}]$ is linear in $h_{\alpha\beta}^{(n)}$ and $\delta^2 G_{\alpha\beta}[h^{(1)}]$ has the schematic form $\partial h_{\alpha\beta}^{(1)} \partial h_{\mu\nu}^{(1)} + h_{\alpha\beta}^{(1)} \partial^2 h_{\mu\nu}^{(1)}$. The reason would become clear as we follow up with the proper derivation.

We continue to derive the explicit form of $\delta G_{\alpha\beta}[h^{(1)}]$ and $\delta^2 G_{\alpha\beta}[h^{(1)}]$. To begin, we define $C^\alpha_{\beta\gamma}$ such that $(\tilde{\nabla}_\beta - \nabla_\beta)\omega^\alpha = C^\alpha_{\beta\gamma}\omega^\gamma$, where $\tilde{\nabla}$ is the covariant derivative compatible with \tilde{g} and ∇ compatible with g . We can show that:

$$C^\alpha_{\beta\gamma} = \tilde{\Gamma}^\alpha_{\beta\gamma} - \Gamma^\alpha_{\beta\gamma} \quad (2.7)$$

$$C^\alpha_{\beta\gamma} = \frac{1}{2} \tilde{g}^{\alpha\mu} (\nabla_\beta h_{\gamma\mu} + \nabla_\gamma h_{\beta\mu} - \nabla_\mu h_{\beta\gamma}) \quad (2.8)$$

$$R^\alpha_{\beta\gamma\delta}[\tilde{g}] = R^\alpha_{\beta\gamma\delta}[g] + 2\nabla_{[\gamma} C^\alpha_{\delta]\beta} + 2C^\alpha_{\mu[\gamma} C^\mu_{\delta]\beta} \quad (2.9)$$

The proof of **Eqs. (2.7) to (2.9)** can be found in **Appendix B**. The $[\]$ in $\nabla_{[\gamma} C^\alpha_{\delta]\beta}$ represents antisymmetric part of the tensor such that $\nabla_{[\gamma} C^\alpha_{\delta]\beta} = \frac{1}{2} (\nabla_\gamma C^\alpha_{\delta\beta} - \nabla_\delta C^\alpha_{\gamma\beta})$. By the use of contraction, we can arrive:

$$R_{\beta\delta}[\tilde{g}] = R^\alpha_{\beta\alpha\delta}[\tilde{g}] \quad (2.10)$$

$$= R_{\beta\delta}[g] + 2\nabla_{[\alpha} C^\alpha_{\delta]\beta} + 2C^\alpha_{\mu[\alpha} C^\mu_{\delta]\beta} \quad (2.11)$$

The Einstein Field Equation is defined as:

$$G_{\alpha\beta}[\tilde{g}] = R_{\alpha\beta}[\tilde{g}] - \frac{1}{2} \tilde{g}_{\alpha\beta} \tilde{g}^{\mu\nu} R_{\mu\nu}[\tilde{g}] = 8\pi T_{\alpha\beta} \quad (2.12)$$

Now, if we were to do the inverse of the exact metric to obtain $\tilde{g}^{\alpha\beta}$, we find that:

$$\tilde{g}^{\alpha\beta} = g^{\alpha\beta} - h^{\alpha\beta} \quad (2.13)$$

Substituting this into **Eq. (2.8)**, we see that:

$$C^\alpha_{\beta\gamma} = \frac{1}{2}g^{\alpha\mu}(\nabla_\beta h_{\gamma\mu} + \nabla_\gamma h_{\beta\mu} - \nabla_\mu h_{\beta\gamma}) + O(h^2) \quad (2.14)$$

is now expanded with a leading order and some higher order h^2 .

Since **Eq. (2.14)** has a leading-order linear in h , we can easily see that in **Eq. (2.11)**, the last term contains only higher order term $O(h^2)$. As such we can rewrite **Eq. (2.11)** into:

$$R_{\alpha\beta}[\tilde{g}] = R_{\alpha\beta}[g] + 2\nabla_{[\mu}C^\mu_{\alpha]\beta} + O(h^2) \quad (2.15)$$

Next, we can see that by linearising the Ricci Tensor, using **Eq. (2.14)** and dropping higher order terms:

$$\delta R_{\alpha\beta}[h] = 0 + 2\nabla_{[\mu}\delta C^\mu_{\alpha]\beta} \quad (2.16)$$

$$= -\frac{1}{2}\square h_{\alpha\beta} - R_{\alpha}{}^{\mu}{}_{\beta}{}^{\nu}h_{\mu\nu} + R_{(\alpha}{}^{\mu}h_{\beta)\mu} + \nabla_{(\alpha}\nabla^{\mu}\bar{h}_{\beta)\mu} \quad (2.17)$$

Where $\square = g^{\mu\nu}\nabla_\mu\nabla_\nu$, $\bar{h}_{\alpha\beta} = h_{\alpha\beta} - \frac{1}{2}g_{\alpha\beta}g^{\mu\nu}h_{\mu\nu}$ and $(\)$ just represents the symmetric part of the tensor such that $\nabla_{(\alpha}\nabla^{\mu}\bar{h}_{\beta)\mu} = \frac{1}{2}(\nabla_\alpha\nabla^{\mu}\bar{h}_{\beta\mu} + \nabla_\beta\nabla^{\mu}\bar{h}_{\alpha\mu})$.

We want to find the first order linearised Einstein Field Equation and using the exact background metric we can show:

$$\delta G_{\alpha\beta}[h] = \delta\left(R_{\alpha\beta}[\tilde{g}] - \frac{1}{2}\tilde{g}_{\alpha\beta}\tilde{g}^{\mu\nu}R_{\mu\nu}[\tilde{g}]\right) \quad (2.18)$$

$$= \delta R_{\alpha\beta}[h] - \frac{1}{2}h_{\alpha\beta}R[g] - \frac{1}{2}g_{\alpha\beta}\delta g^{\mu\nu}R_{\mu\nu}[g] - \frac{1}{2}g_{\alpha\beta}g^{\mu\nu}\delta R_{\mu\nu}[h] \quad (2.19)$$

With specialisation to the vacuum background where $R_{\mu\nu}[g] = 0$ and using **Eq. (2.17)** we can finally get:

$$\delta G_{\alpha\beta}[h] = -\frac{1}{2}\square\bar{h}_{\alpha\beta} - R_{\alpha}{}^{\mu}{}_{\beta}{}^{\nu}\bar{h}_{\mu\nu} + \nabla_{(\alpha}\nabla^{\mu}\bar{h}_{\beta)\mu} - \frac{1}{2}g_{\alpha\beta}\nabla_\mu\nabla_\nu\bar{h}^{\mu\nu} \quad (2.20)$$

Now, the same procedure can be done for $\delta^2 R_{\alpha\beta}[h]$ and we would arrive:

$$\delta^2 G_{\alpha\beta}[h] = \delta^2 \left(R_{\alpha\beta}[\tilde{g}] - \frac{1}{2} \tilde{g}_{\alpha\beta} \tilde{g}^{\mu\nu} R_{\mu\nu}[\tilde{g}] \right) \quad (2.21)$$

$$= \delta^2 R_{\alpha\beta}[h] - \frac{1}{2} g_{\alpha\beta} g^{\mu\nu} \delta^2 R_{\mu\nu}[h] - \frac{1}{2} h_{\alpha\beta} g^{\mu\nu} \delta R_{\mu\nu}[h] + \frac{1}{2} g_{\alpha\beta} h^{\mu\nu} \delta R_{\mu\nu}[h] \quad (2.22)$$

where

$$\begin{aligned} \delta^2 R_{\alpha\beta}[h] &= \frac{1}{4} \nabla_\alpha h^{\mu\nu} \nabla_\beta h_{\mu\nu} + \frac{1}{2} \nabla^\nu h^\mu{}_\beta (\nabla_\nu h_{\mu\alpha} - \nabla_\mu h_{\nu\alpha}) \\ &\quad - \frac{1}{2} \nabla_\nu \bar{h}^{\mu\nu} \left(2 \nabla_{(\alpha} h_{\beta)\mu} - \nabla_\mu h_{\alpha\beta} \right) - \frac{1}{2} \left(2 \nabla_\nu \nabla_{(\alpha} h_{\beta)\mu} - \nabla_\nu \nabla_\mu h_{\alpha\beta} - \nabla_\beta \nabla_\alpha h_{\mu\nu} \right) \end{aligned}$$

We now substitute **Eq. (2.5)** and **Eq. (2.6)**, we would get:

$$\epsilon \delta G_{\alpha\beta}[h^{(1)}] + \epsilon^2 (\delta G_{\alpha\beta}[h^{(2)}] + \delta^2 G_{\alpha\beta}[h^{(1)}]) = 8\pi \left(\epsilon T_{\alpha\beta}^{(1)} + \epsilon^2 T_{\alpha\beta}^{(2)} + O(\epsilon^3) \right) \quad (2.23)$$

Comparing term by term, it is easy to see that:

$$\delta G_{\alpha\beta}[h^{(1)}] = 8\pi T_{\alpha\beta}^{(1)} \quad (2.24)$$

$$\delta G_{\alpha\beta}[h^{(2)}] = 8\pi T_{\alpha\beta}^{(2)} - \delta^2 G_{\alpha\beta}[h^{(1)}] \quad (2.25)$$

Eq. (2.24) presents the linearised Einstein equation specifically for a point-particle source. There has been a long history of using first-order perturbation theory to compute gravitational waveforms from particles in geodesic orbits around black holes [13, 14, 15, 16]. These computations have recently been extended to fully generic orbits [17, 18]. However, first-order perturbation theory is limited to producing “snapshot” waveforms that neglect radiation reaction.

Eq. (2.25) is the second-order term in the linearised Einstein equation. It is evident that the second-order term is sourced by the non-linear terms of $h^{(1)}$ in $\delta^2 G_{\alpha\beta}[h^{(1)}]$. As it is constructed from a non-linear operation on an integrable function, it is not well-defined at all. The other source term $T_{\alpha\beta}^{(2)}$, if even well-defined, must be a distribution solely supported by the particle’s worldline. Hence, it is unable to resolve the problems arising from $\delta^2 G_{\alpha\beta}[h^{(1)}]$. Therefore, the whole **Eq. (2.25)** is not well-defined. Additionally, at

this order, the particle’s geodesic motion must be corrected by self-force effects describing its interaction with its own spacetime distortion. Although such formalism exists [19, 20], it has been challenging to translate this expression into a practical computational scheme for Kerr black holes because of the mathematical complexity of the self-field regularisation required [9, 21]. As a result, this matter has been the focus of ongoing research at the research frontiers.

The complexity of the terms increases as we go higher orders, as seen from **Eq. (2.25)**. However, due to the nature of EMRIs, to obtain a sufficiently accurate waveform for the LISA mission, it is possible to express up to the second order (more will be explained later). In this project, we will only consider up to the first order of the linearised Einstein equation to produce ‘snapshots’ of the gravitational waveform and attempt to stitch the snapshots together to produce the adiabatic waveform of an EMRI binary.

2.2 Bound Kerr Geodesics

Before we begin, let us denote M and S as the mass and spin parameters of the Kerr black hole, with $a = S/M$ representing its dimensionless spin magnitude and $\mu \ll M$ the mass of the small body (hence, $\epsilon = \mu/M$). In our work, we will be dealing with prograde orbits.

To start our analysis, we will discuss bound Kerr geodesics. The Kerr geodesic in Boyer-Lindquist coordinates [22] (with $G = c = 1$) is expressed as:

$$ds_{\text{Kerr}}^2 = - \left(1 - \frac{2Mr}{\Sigma} \right) dt^2 + \frac{\Sigma}{\Delta} dr^2 + \Sigma d\theta^2 + \left(r^2 + a^2 + \frac{2Ma^2r}{\Sigma} \sin^2 \theta \right) \sin^2 \theta d\phi^2 - \frac{4Mar}{\Sigma} \sin^2 \theta dt d\phi \quad (2.26)$$

where $\Sigma = r^2 + a^2 \cos^2 \theta$ and $\Delta = r^2 - 2Mr + a^2$.

Kerr orbits are governed by three constants of motion ¹ related to symmetries: E , L_z , and Q . They represent the orbit’s energy, axial angular momentum, and Carter’s constant, respectively. Since these three constants of motion are independent, the equations of motion of a test particle in Kerr spacetime are integrable. Thus, we can uniquely parametrise all geodesic orbits with the parameters $\{E, L_z, Q\}$ in Kerr spacetime. Without incorporating radiation reaction, Kerr geodesics are governed by four equations [16]:

¹Although there is a fourth equation of motion, μ the small CO mass, $\{E, L_z, Q\}$ are sufficient to completely and uniquely parametrise all geodesic orbits in Kerr spacetime.

$$\begin{aligned}
\Sigma^2 \left(\frac{dr}{d\tau} \right)^2 &= [E(r^2 + a^2) - aL_z]^2 - \Delta [r^2 + (L_z - aE)^2 + Q] \equiv R, \\
\Sigma^2 \left(\frac{d\theta}{d\tau} \right)^2 &= Q - \cot^2 \theta L_z^2 - a^2 \cos^2 \theta (1 - E^2) \equiv \Theta^2 \\
\Sigma \left(\frac{d\phi}{d\tau} \right) &= \csc^2 \theta L_z + aE \left(\frac{r^2 + a^2}{\Delta} - 1 \right) - \frac{a^2 L_z}{\Delta}, \\
\Sigma \left(\frac{dt}{d\tau} \right) &= E \left[\frac{(r^2 + a^2)^2}{\Delta} - a^2 \sin^2 \theta \right] + aL_z \left(1 - \frac{r^2 + a^2}{\Delta} \right).
\end{aligned} \tag{2.27}$$

Next, we will gain some geometrical intuition on a bound Kerr orbit. A bound Kerr geodesic can be thought of as being confined to a toroidal region given by:

$$r_{\min} \leq r \leq r_{\max} \tag{2.28}$$

$$\theta_{\min} \leq \theta \leq \pi - \theta_{\min} \tag{2.29}$$

As such, when we represent the constants of motion in quasi-Keplerian orbital elements (such as the semi-latus rectum distance p , eccentricity e , as well as inclination ι), we obtain [13, 23]:

$$r_{\min} = \frac{pM}{1 + e} \tag{2.30}$$

$$r_{\max} = \frac{pM}{1 - e} \tag{2.31}$$

$$\cos \iota = \frac{L_z}{\sqrt{L_z^2 + Q}} \tag{2.32}$$

Similarly we can show how E and L_z are related to the new parameters [13]:

$$E^2 = \frac{(p - 2 - 2e)(p - 2 + 2e)}{p(p - 3 - e^2)} \tag{2.33}$$

$$L_z^2 = \frac{p^2 M^2}{(p - 3 - e^2)} \tag{2.34}$$

Because of this parameterisation, we can conveniently describe the orbit's triperiodic motion with these three constants of motion: $\{p, e, \iota\}$, as choosing p , e , and ι is equivalent to choosing the integrals of motion E , L_z , and Q . Additionally, we are able to describe the three orbital phases by $\{\Omega_r, \Omega_\theta, \Omega_\phi\}$ [24], which are the fundamental frequencies in a Kerr geodesic.

2.2.1 Circular, Equatorial Kerr Orbit

However, since we are dealing with a specific type of orbit, both the eccentricity e and inclination ι will be 0. The only remaining intrinsic parameterisation will be the semi-latus rectum distance p and the SMBH dimensionless spin parameter a . In this study, we will focus on this set of parameters $\{p, a\}$ when constructing our waveform. Since it is a circular equatorial orbit, the orbit will remain circular in the equatorial plane throughout the inspiral [25]. Therefore, there can only be one fundamental frequency describing the orbit, namely Ω_ϕ . We can solve for Ω_ϕ by using **Eq. (2.27)** and imposing these conditions: $d\theta/d\tau = 0$, $R = 0 = R'$ and $\iota = 0^\circ$ or 180° , and we obtain [16]:

$$\Omega_\phi = \frac{d\phi}{dt} = \frac{d\phi}{d\tau} \frac{d\tau}{dt} = \pm \frac{M^{\frac{1}{2}}}{(pM)^{\frac{3}{2}} \pm aM^{\frac{3}{2}}} \quad (2.35)$$

where the upper sign refers to prograde orbits, and the lower sign refers to retrograde ones.

2.3 Teukolsky Formalism

As we have seen in our linearised Einstein equations (**Eqs. (2.24) to (2.25)**), the parameters are all coupled together in partial differential equations (PDEs), which increases the complexity of solving the equations. Teukolsky devised a brilliant method to decouple the PDEs into ordinary differential equations (ODEs) [10, 26]. In this section, we will outline the essential steps he took and derive the equation we will be using in our project.

To begin with, Teukolsky decomposed the metric perturbation into a new basis using the Newman-Penrose (NP) formalism [27]. These new bases essentially introduce new null tetrad vectors:

$$\left\{ e_{(a)}^\alpha \right\} = \{ l^\alpha, n^\alpha, m^\alpha, \bar{m}^\alpha \} \quad (2.36)$$

With normalisation:

$$l^\alpha n_\alpha = -1, \quad (2.37)$$

$$m^\alpha \bar{m}_\alpha = 1 \quad (2.38)$$

With all other inner products vanishing, the metric perturbation can be expressed in the null tetrad vectors as shown:

$$g_{\alpha\beta} = -2l_{(\alpha}n_{\beta)} + 2m_{(\alpha}\bar{m}_{\beta)} \quad (2.39)$$

To our interest, we will be targeting only one of the Weyl scalars, ψ_4 . ψ_4 is related to the fluxes of energy and angular momentum of the system [28]. It always represents the dissipative piece of the self-force. Therefore, by solving the Teukolsky equation (which is essentially solving the first-order linearised Einstein equation), we will always obtain the first-order dissipative piece.

In our project, we are considering perturbations only to the first order in the mass ratio. Thus, ψ_4 will always represent the leading $O(\epsilon)$ contribution to the curvature perturbation, and by focusing on ψ_4 , we are mainly dealing with outgoing perturbations. This treatment allows us to obtain the solution for our leading-order adiabatic piece - 0PA waveform.

Continuing via the NP-formalism we can write:

$$\psi_4 = -C_{\alpha\beta\gamma\delta}n^\alpha\bar{m}^\beta n^\gamma\bar{m}^\delta \quad (2.40)$$

where $C_{\alpha\beta\gamma\delta}$ is the Weyl curvature tensor, which is equals to the Riemann tensor in vacuum.

By using the NP-formalism and working in the Boyer-Lindquist coordinates [22] to solve for the Weyl scalar, Teukolsky has shown that the scalars can be expressed in:

$$\begin{aligned} & \left[\frac{(r^2 + a^2)^2}{\Delta} - a^2 \sin^2 \theta \right] \partial_t^2 \Psi - 4 \left[r + ia \cos \theta - \frac{M(r^2 - a^2)}{\Delta} \right] \partial_t \Psi \\ & + \frac{4Mar}{\Delta} \partial_\phi \partial_t \Psi - \Delta^2 \partial_r (\Delta^{-1} \partial_r \Psi) - \frac{1}{\sin \theta} \partial_\theta (\sin \theta \partial_\theta \Psi) + \left[\frac{a^2}{\Delta} - \frac{1}{\sin^2 \theta} \right] \partial_\phi^2 \Psi \\ & + 4 \left[\frac{a(r - M)}{\Delta} + \frac{i \cos \theta}{\sin^2 \theta} \right] \partial_\phi \Psi + (4 \cot^2 \theta + 2) \Psi = 4\pi \Sigma \mathcal{T} \end{aligned} \quad (2.41)$$

where $\Psi = (r - ia \cos \theta)^4 \psi_4$ and \mathcal{T} is a source term that whose explicit form can be found in [10, 16].

In this expression, we see that the parameters r and θ are coupled together. Although theoretically one can still use this master equation **Eq. (2.41)** to numerically solve the PDEs and obtain a solution in the time domain, this treatment will not be accurate enough for our EMRI treatment, as time domain treatments are accurate only to about 10% for EMRI sources [29]. Therefore, we will use the frequency domain approach, following Teukolsky's

non-source-free ($\mathcal{T} \neq 0$) Fourier and multipolar expansion:

$$\psi_4 = \frac{1}{(r - ia \cos \vartheta)^4} \int_{-\infty}^{\infty} d\omega \sum_{l=2}^{\infty} \sum_{m=-l}^l R_{lm}(r; \omega) S_{lm}(\vartheta; a\omega) e^{i[m\varphi - \omega t]} \quad (2.42)$$

From **Eq. (2.42)**, we can see that the master equation has been decoupled into R_{lm} and S_{lm} . R_{lm} is considered as the radial part of the Teukolsky equation, and S_{lm} is the angular part of the Teukolsky equation. S_{lm} is also known as the -2 spin-weighted spheroidal harmonics, defined by Teukolsky. Essentially, the functions R_{lm} and S_{lm} are solutions obtained by solving the ODEs from the Teukolsky equations. The explicit form of the source-free ODEs can be found in [10].

2.3.1 Hughes' Approach

Following Teukolsky's work, Hughes further developed a computational and generic approach for EMRI waveforms. In his work, he derived a more general solution to the Teukolsky equation through the use of Green's functions [16]. Specifically, he obtained the expression:

$$R_{lm\omega}(r) = Z_{lm\omega}^H(r) R_{lm\omega}^{\infty}(r) + Z_{lm\omega}^{\infty}(r) R_{lm\omega}^H(r) \quad (2.43)$$

A detailed description of how one can compute $Z_{lm\omega}^H(r)$, $Z_{lm\omega}^{\infty}(r)$, $R_{lm\omega}^{\infty}(r)$, and $R_{lm\omega}^H(r)$ and how they relate to the source function $\mathcal{T}_{lm\omega}$ can be found in [16]. Similarly, a detailed computational method for S_{lm} the -2 spin-weighted spheroidal harmonics can be found in [16].

One important note is the existence of a symmetry in the Teukolsky amplitude Z [5]:

$$Z_{l,-m}^{\infty} = (-1)^l \bar{Z}_{l,m}^{\infty} \quad (2.44)$$

where \bar{Z} refers to the complex conjugate of Z .

By exploiting this symmetry, we can cut down on our computation time which we would only need to focus on the positive m modes amplitude during computation and fit.

2.3.2 Circular Equatorial Orbit Waveform Snapshot

Circular orbits remain circular [25]. Additionally, since this is also an equatorial orbit, there will be no precession. Our orbit's source and all quantities derived from it will be described

very simply by one fundamental frequency Ω_ϕ [16]. As ω is only affected by one mode m , when we decompose ω into its harmonics, we obtain [30]:

$$Z_{lm\omega}^{H,\infty} = Z_{lm}^{H,\infty} \delta(\omega - \omega_m) \quad (2.45)$$

where $\omega_m = m\Omega_\phi$ and Ω_ϕ described in **Eq. (2.35)**.

As such, when we consider events far from source [5]:

$$R_{lm\omega}(r) \rightarrow Z_{lm\omega}^\infty r^3 e^{i\omega r^*} \quad r \rightarrow \infty \quad (2.46)$$

where r^* is the Kerr tortoise coordinates [31].

In the limits of $r \rightarrow \infty$, where the observer is far from the source ψ_4 becomes [5]:

$$\psi_4 = \frac{1}{2} \frac{d^2}{dt^2} (h_+ - ih_\times) \quad \text{as } r \rightarrow \infty \quad (2.47)$$

In this expression, we can see that when we are far from the source ψ_4 encodes the gravitational waves. By putting together **Eq. (2.42)**, **Eq. (2.45)**, **Eq. (2.46)** and **Eq. (2.47)**, we can obtain:

$$h_+ - ih_\times \propto \frac{1}{r} \sum_{l=2}^{\infty} \sum_{m=-l}^l \frac{Z_{lm}^\infty}{\omega_m^2} S_{lm}(\vartheta, a\omega_m) e^{i[m\varphi - \omega(t-r^*)]} \quad (2.48)$$

Performing a change of basis from the -2 spin-weighted spheroidal harmonics to the -2 spin-weighted spherical harmonics, as described in more detail in [32] (but the general idea is in the limits of $a\omega \rightarrow 0$, ${}_{-2}Y_{lm} = {}_{-2}S_{lm} e^{im\varphi}$ [10]), and for computational ease and notation, we will treat the retarded time as just t . Thus, we have:

$$h \equiv h_+ - ih_\times \equiv \frac{1}{r} \sum_{l=2}^{\infty} \sum_{m=-l}^l A_{lm} Y_{lm}(\vartheta, \varphi) e^{-i\omega_m t} \quad (2.49)$$

where $A_{lm} = \frac{-2Z_{lm}^\infty}{\omega_m^2}$ and $Y_{lm}(\vartheta, \varphi)$ is the -2 spin-weighted spherical harmonics.

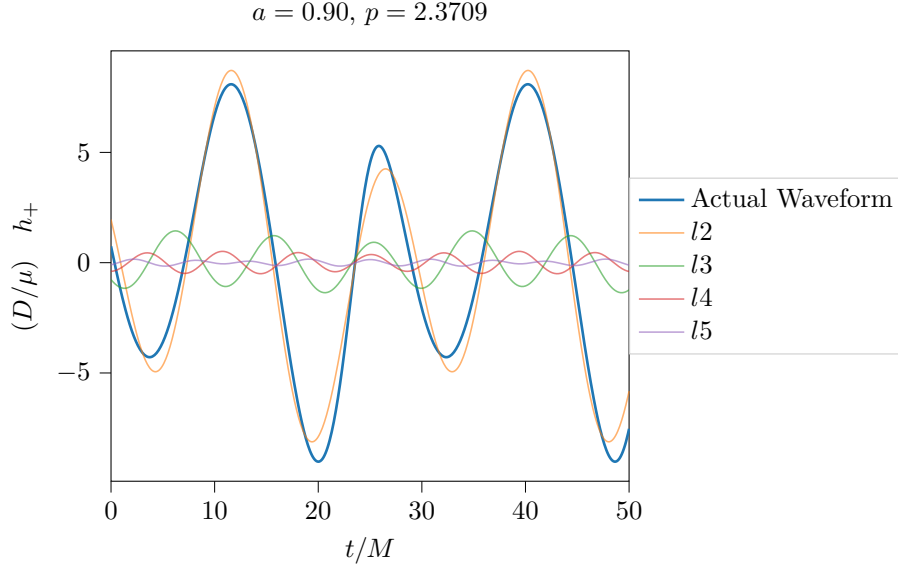


Figure 2.1: Example waveform snapshot decomposed into its l modes. The modes are only summed over m . Here we can see that after decomposing the modes, we can gauge how the higher modes will affect the overall waveform.

We will use **Eq. (2.49)** to construct our waveform snapshot. The equation contains non-evolving terms, such as A_{lm} and ω_m , implicitly assuming a small mass μ in the limiting condition $\mu \rightarrow 0$ while generating the gravitational waves in the fixed geodesic. Thus, the essence of a snapshot waveform is to treat μ as negligible mass but still part of a geodesic to the SMBH.

We can understand how the mode sum decomposition works by examining **Figure 2.1**. The higher modes contribute smaller amplitudes to the overall waveform. Nevertheless, we need to sum them to a desired accuracy as they are higher in frequency and could potentially cause a phase difference in the actual waveform if we do not sum enough modes. However, we cannot sum to infinity as it would take a long computational time. The truncation of the l modes in our project is determined by **GREMLIN** (which we will explore later). A proper study and assessment of how to truncate these sums should be done before using these ideas to create ‘production quality’ waveforms for the LISA mission. However, for the purpose of our project, since we are not investigating on that matter, for simplicity, we will solely use what is provided in **GREMLIN**.

2.4 Full Waveform Evolution

Once we have pieced together our waveform snapshots, we will need to find a way to ‘stitch’ them together into a complete, evolving waveform. To do so, we must solve for the trajectory of our small CO. In our study, we will use a Post-Newtonian (PN) approximation to our adiabatic waveform to model the trajectory of our small CO. One might wonder why we are not using the proper adiabatic waveform approach. This is mainly because our project primarily focuses on methods for fitting the Teukolsky amplitude, and the trajectory falls outside the scope of our project. Therefore, for simplicity, we will use a form that is widely available to us to construct our trajectory. However, in Chapter 5, we will outline a possible future direction to incorporate the proper construction of the adiabatic waveform’s trajectory.

The general 3PN- e^6 expression for the trajectory as obtained from [33] is:

$$\begin{aligned}
\left\langle \frac{dv}{dt} \right\rangle_t^\infty = & \left(\frac{dv}{dt} \right)_N \left[1 + \frac{7}{8}e^2 + \left\{ -\frac{743}{336} - \frac{55}{21}e^2 + \frac{8539}{2688}e^4 \right\} v^2 \right. \\
& + \left\{ 4\pi - \frac{133}{12}Yq + \left(\frac{97}{8}\pi - \frac{379}{24}Yq \right) e^2 \right. \\
& + \left. \left(\frac{49}{32}\pi - \frac{475}{96}Yq \right) e^4 - \frac{49}{4608}\pi e^6 \right\} v^3 \\
& + \left\{ \frac{34103}{18144} - \frac{329}{96}q^2 + \frac{815}{96}Y^2q^2 + \left(-\frac{526955}{12096} - \frac{929}{96}q^2 + \frac{477}{32}Y^2q^2 \right) e^2 \right. \\
& + \left. \left(-\frac{1232809}{48384} - \frac{1051}{768}q^2 + \frac{999}{256}Y^2q^2 \right) e^4 + \frac{105925}{16128}e^6 \right\} v^4 \\
& + \left\{ -\frac{4159}{672}\pi - \frac{1451}{56}Yq + \left(-\frac{48809}{1344}\pi - \frac{1043}{96}Yq \right) e^2 \right. \\
& + \left. \left(\frac{679957}{43008}\pi - \frac{15623}{336}Yq \right) e^4 + \left(\frac{4005097}{774144}\pi - \frac{35569}{1792}Yq \right) e^6 \right\} v^5 \\
& + \left\{ \frac{16447322263}{139708800} + \frac{16}{3}\pi^2 - \frac{1712}{105}\gamma - \frac{3424}{105}\ln(2) - \frac{331}{192}q^2 \right. \\
& - \frac{289}{6}\pi Yq + \frac{145759}{1344}Y^2q^2 \\
& + \left(\frac{8901670423}{11642400} + \frac{229}{6}\pi^2 - \frac{24503}{210}\gamma + \frac{1391}{30}\ln(2) - \frac{78003}{280}\ln(3) \right. \\
& + \left. \frac{2129}{42}q^2 - \frac{4225}{24}\pi Yq + \frac{27191}{224}Y^2q^2 \right\} e^2 \\
& + \left(\frac{269418340489}{372556800} + \frac{109}{4}\pi^2 + \frac{3042117}{1120}\ln(3) - \frac{11663}{140}\gamma \right. \\
& \left. - \frac{418049}{84}\ln(2) - \frac{56239}{10752}q^2 - \frac{17113}{192}\pi Yq + \frac{414439}{3584}Y^2q^2 \right) e^4
\end{aligned} \tag{2.50}$$

$$\begin{aligned}
& + \left(\frac{174289281}{862400} - \frac{1044921875}{96768} \ln(5) + \frac{23}{16} \pi^2 - \frac{42667641}{3584} \ln(3) \right. \\
& + \frac{94138279}{2160} \ln(2) - \frac{2461}{560} \gamma - \frac{3571}{3584} q^2 - \frac{108577}{13824} \pi Y q \\
& + \left. \frac{41071}{1536} Y^2 q^2 \right) e^6 \\
& - \left(\frac{1712}{105} + \frac{24503}{210} e^2 + \frac{11663}{140} e^4 + \frac{2461}{560} e^6 \right) \ln(v) \Big\} v^6 \Big]
\end{aligned}$$

where $Y = \cos \iota$, $v = \sqrt{1/p}$, $q = a$, γ is the Euler constant and

$$\left(\frac{dv}{dt} \right)_N = \frac{32}{5} \left(\frac{\mu}{M^2} \right) v^9 (1 - e^2)^{3/2} \quad (2.51)$$

However, since we are interested in finding the circular equatorial orbit, we let $e = 0$ and $Y = 1$ for prograde orbit, we obtain:

$$\begin{aligned}
\left\langle \frac{dv}{dt} \right\rangle_t^\infty &= \left(\frac{32}{5} \left(\frac{m}{M^2} \right) v^9 \right) \left[1 - \frac{743}{336} v^2 \right. \\
& + \left(4\pi - \frac{133}{12} q \right) v^3 \\
& + \left(\frac{34103}{18144} - \frac{329}{96} q^2 + \frac{815}{96} q^2 \right) v^4 \\
& + \left(-\frac{4159}{672} \pi - \frac{1451}{56} q \right) v^5 \\
& + \left(\frac{16447322263}{139708800} + \frac{16}{3} \pi^2 - \frac{1712}{105} \gamma - \frac{3424}{105} \ln(2) \right. \\
& \left. \left. - \frac{331}{192} q^2 - \frac{289}{6} \pi q + \frac{145759}{1344} q^2 - \frac{1712}{105} \ln(v) \right) v^6 \right]
\end{aligned} \quad (2.52)$$

By solving the ordinary differential equation in **Eq. (2.52)**, we obtain a function of $p(t)$. Once we have this function, we can set the initial condition p_0 and determine our trajectory. We know that at each geodesic, the frequency changes. At each value of p , we obtain a corresponding Ω_ϕ using **Eq. (2.35)**.

With knowledge of how p evolves over time, we can use a set of parameters $\{p(t), a\}$ (since the spin parameter does not change, it will remain constant throughout the trajectory) to determine the A_{lm} as a function of t as well.

Finally, the evolving waveform is given by [11, 5]:

$$h \equiv h_+ - ih_\times \equiv \frac{1}{r} \sum_{l=2}^{\infty} \sum_{m=-l}^l A_{lm}(t) Y_{lm}(\vartheta, \varphi) e^{-i\Phi_m(t)} \quad (2.53)$$

where we have replaced the snapshot phase $\omega_m t$ by the adiabatic phase:

$$\Phi_m(t) = \int_{t_0}^t \omega_m(t') dt' \quad (2.54)$$

Using this set of equations, we can plot the full evolving waveform. However, due to limitations in computation time and plot resolution, we have restricted our plots to show only the first 2 hours and last 2 hours of a one-year inspiral before the plunge. This will be discussed further in Chapter 4.

3. Methodology

The general outline of our project can be summarised as follows:

1. Generating a toy dataset
2. Implementing a neural network (NN) and training it on the toy dataset to obtain a desirable accuracy
3. Generating a larger dataset, as well as some test sets
4. Implementing interpolation while working on training the NN on the new dataset
5. Implementing polyfit on the larger training dataset
6. Obtaining the final dataset and then partitioning it into training and test sets
7. Implementing all three methods on the final dataset to obtain results as shown in Chapter 4

In this chapter, we will discuss the implementation of all three methods (NN, interpolation, and polyfit), as well as how we obtained and structured our dataset. Finally, we will explain the error metric used in this project.

3.1 Generation of Data - GRELIM

Work has been done by incorporating the Teukolsky equation into code and releasing it for open source usage. The ‘Black Hole Perturbation Toolkit’ is a leading resource for individuals or groups interested in research in the area of black hole perturbation theory. For generating our data, we utilised **GREMLIN**, a tool from the ‘Black Hole Perturbation Toolkit’ [34]. We were particularly interested in using two of its executables to generate our Teukolsky amplitudes, namely `Circ_Eq_Seq2` and `Cir_Eq`. Details of the functions can be found in the documentation of **GREMLIN** [35].

We created personal scripts to automate and expedite the generation process. These scripts can be found in **Appendix C.1**. The toy dataset was generated in a mere 10 minutes, while generating the other dataset could take up to 2 hours for ~ 400 parameters and 119 modes. In full generic EMRI cases, the number of modes can reach up to 1×10^5 . Thus, our work is particularly relevant and important as we can already see that generating data

for this few numbers are already taking a toll to our computation time.

We also utilised the transformation of parameters from p to u and u' , as described in § 3.5.6, to produce data on a uniform grid.

3.2 Implementation of Neural Network

A neural network (NN) is a type of machine learning algorithm that is inspired by the structure and function of the human brain. It consists of interconnected processing units called neurons that work together to learn and recognise patterns in data. Neural networks can be used for a variety of tasks such as image and speech recognition, natural language processing, and recommendation systems.

During training, the network adjusts the weights between neurons to minimise the difference between its output and the desired output. The process involves forward propagation of input data through the network, followed by backpropagation of errors to adjust the weights. Neural networks can have one or more hidden layers between the input and output layers, allowing for more complex and abstract representations of the input data.

Implementing a neural network model is not a straightforward process and sometimes requires a bit of guesswork and creativity to produce the perfect model. As the dimensionality of our model increases, an NN becomes something like a black box where it is extremely difficult to understand what the network is doing. In such cases, the best we can do is tweak some values here and there to try and obtain a functioning model to our desired accuracy. Following previous work [36], we created an NN with similar features. The following is a summary of our NN model:

1. Our NN is constructed with 20 hidden layers a_n where n is the layer number ($n = 1$ for the first layer), and the first 6 layers each contain 2^{n+1} nodes where $n = 1, 2, \dots, 6$, and the remaining layers each contain 256 nodes.
2. In our first implementation of our NN, we used basic cross-validation where 20% of the training sample is held out for validation. This validation method is used for training of the toy dataset and the 20×21 training dataset.
3. K-fold validation is used for the final training dataset. The training set is divided into 5 blocks. Every 2000 epochs, one block will be left out for validation while the network trains on the other 4 blocks. This iteration runs through 5 times until all blocks have

been validated. **Figure 3.1** shows an illustration of how it would work. Each split would correspond to 2000 epochs that the NN would train on until all 5 splits were done training.

4. The mini-batch size is 80, 210, and 60, respectively, for the training of the toy dataset, the 20×21 training dataset, and the final training dataset.
5. The loss function we used is the mean-squared error where $\text{loss} = (y_{\text{true}} - y_{\text{pred}})^2$ averaged over each mini-batch.
6. Our NN is trained on 6000, 4000, and finally 10000 epochs, respectively, for the toy dataset, the 20×21 training dataset, and the final training dataset.
7. The learning rate starts at 1×10^{-4} and eventually decreases to 1×10^{-5} after 1000 epochs.
8. Our network is designed to take in two inputs, $\{p, a\}$, and produce two outputs: the real and imaginary components of the Teukolsky amplitude.

The final model can be found in **Appendix C.2**.

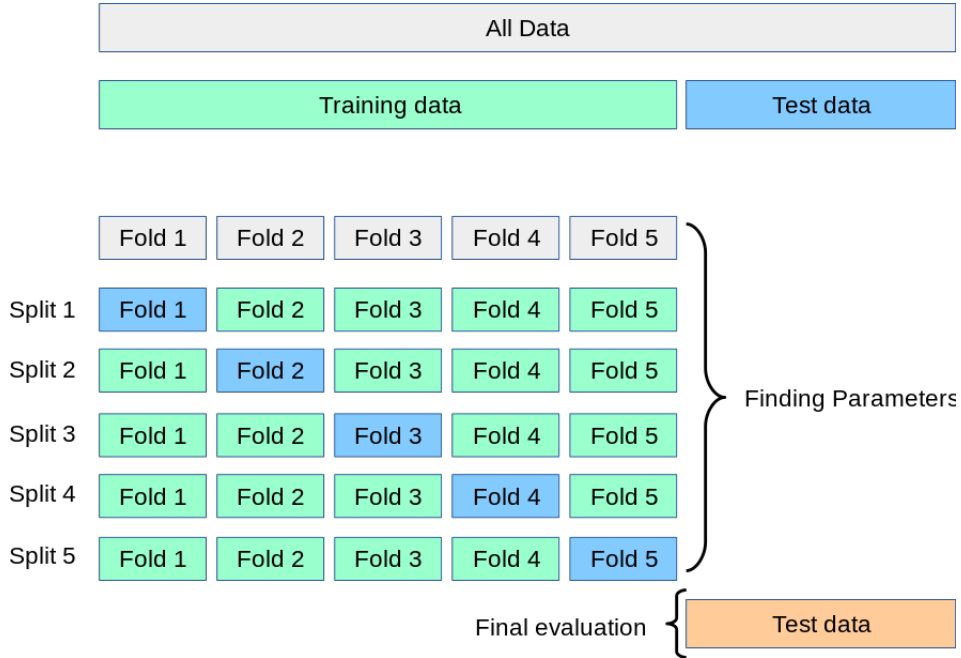


Figure 3.1: Illustration of how K-Fold validation works. Credits: [37]

3.3 Implementation of Interpolation

Spline interpolation is a mathematical technique used to estimate values between known data points. It works by fitting a piecewise polynomial curve to the data, allowing for smooth and continuous interpolation between points.

Since we are dealing with a specific case of EMRI waveforms, we only need to work in two dimensions. Therefore, it is possible to interpolate the amplitudes over our parameter space. To implement the process, we need to construct splines using our training dataset. After obtaining the splines, we use them to construct the curves for our amplitudes in the test set parameter space. This requires the use of bivariate splines, an extension of spline interpolation to two-dimensional data. Like spline interpolation in one dimension, bivariate spline interpolation fits a piecewise polynomial surface to the data, allowing for smooth and continuous interpolation between points in two dimensions.

For this project, we used the tools provided by `scipy` [38]. Specifically, we used `scipy.interpolation.RectBivariateSpline()` to construct our splines. The splines were constructed for each mode, with the number of splines equal to the number of modes in each amplitude.

3.4 Implementation of Polyfit

Two-dimensional (2D) polynomial fitting is a mathematical technique used to approximate a function using a polynomial surface in two dimensions. It works by fitting a polynomial function to a set of data points that are defined in a two-dimensional space, which allows for the estimation of new data points that fall between the known data points. The polynomial function can be of any degree, and the degree of the polynomial determines the complexity and accuracy of the fit.

Constructing a polynomial fit is not a simple task as there are no existing packages that provide a 2D fit, which is required in this case. However, one can still write their own functions for a polynomial 2D fit as shown in **Listing C.4**.

The main challenge in implementing a polyfit is to determine the optimal degree of the polynomial. Since amplitudes at high p are simpler than the ones near p_{ISCO} , we cannot simply increase the degree of the polynomial as we wish. There has to be an optimum degree that gives desirable accuracy across the entire parameter space.

After some experimentation, we have decided to use 9 and 25 for the degree of p and a , respectively. The fit is then constructed for each mode and each component of the amplitudes. Thus, the number of fits we have is twice the number of total modes in one parameter set.

3.5 Data Formats

In total, we have generated four datasets: the Toy dataset, a 20×21 Training dataset, Test Dataset A, and Test Dataset B and a Final Dataset that was split into two which we used in our project. The following summarises what the datasets consist of:

3.5.1 A 10×10 Toy Dataset

In the toy dataset, we have the following:

1. A non-uniform grid of 10×10 , with a against p .
2. p runs from p_{ISCO} to $p_{\text{ISCO}} + 10$ in steps determined automatically by `Circ_Eq_Seq2`.
3. a runs from 0.0 to 0.9 in steps of 0.1.
4. Some modes are artificially added with $[0, 0]$ amplitudes to match the parameter with the highest number of modes so that each set of parameters' amplitude can have the same number of modes.
5. A total of 119 modes, e.g. $l2m1, \dots, l15m15$, were generated for each set of parameters.

The data is generated with GREMLIN's `Circ_Eq_Seq2` using a self-written script in **Listing C.1**.

3.5.2 A 20×21 Training Dataset

In the training dataset, we have the following:

1. A uniform grid of 20×21 , with a against u .
2. The transformation of p to u is given by **Eq. (3.1)**.
3. u runs from 0 to 10 in steps of 0.5.
4. a runs from 0.00 to 0.95 in steps of 0.05.
5. To maintain consistency, amplitudes are generated with modes of $l2$ to $l_{\text{max}} = 15$, keeping the total number of modes to 119 for each set of parameters.

The data is generated with `Circ_Eq` and the script is given in **Listing C.2**.

3.5.3 10×9 Test Dataset A

In test set A, we have the following:

1. A uniform grid of 10×9 , with a plotted against $p - p_{\text{ISCO}}$
2. $p - p_{\text{ISCO}}$ runs from 0.5 to 9.5 in steps of 1.0

3. a runs from 0.05 to 0.85 in steps of 0.1
4. A total of 119 modes were generated for each set of parameters

The data is generated with the `Circ_Eq` code. The script to produce this test set is not included as it takes a similar form to that shown in **Listing C.2**.

3.5.4 20×14 Test Dataset B

In test set B, we have increased the size of our test set. It consists of the following:

1. A uniform grid of 20×14 , with a plotted against u
2. Similarly, the transformation of p to u is given by **Eq. (3.1)**.
3. u runs from 0.2 to 9.95 in steps of 0.75
4. a runs from 0.430 to 0.905 in steps of 0.025
5. A total of 119 modes were generated for each set of parameters.

The data is generated with the `Circ_Eq` code. The script is shown in **Listing C.3**.

3.5.5 Final Dataset 10×60

The problem with our previous datasets was that the amplitudes were generated in the spin-weighted spheroidal harmonics basis. This posed a problem for us since we do not have the tools to construct the $S_{lm}(\vartheta, a\omega_m)$. Therefore, the new dataset enables us to generate waveforms using the spin-weighted spherical harmonics basis.

The dataset contains the amplitude vector A , where $A = -2Z^\infty/\omega^2$, which contains the Teukolsky amplitudes in the spin-weighted spherical harmonic basis. The dataset provided was for a fully generic EMRI geodesic. The first step is to extract only the relevant data specifically for a circular, equatorial orbit EMRI waveform. After extraction, we now have what we consider our final dataset. The final dataset is arranged as follows:

1. An approximately uniform grid of 10×60 , with a against u' , where u' is a transformation defined in **Eq. (3.5)**.
2. u' runs from 0.000 to approximately 2.065 in steps of 0.035.
3. a runs from 0.0 to 0.9 in steps of 0.1.
4. A total of 629 modes were generated.
5. Amplitudes generated from the data are in the spherical harmonics basis.

The training set is constructed by taking all even indices (starting from index 0) in the

parameter space, while the test set is constructed by taking all odd indices (starting from index 1) in the parameter space.

3.5.5.1 Final Training Dataset 10×30

Thus, for the final training set, we have:

1. An approximately uniform grid of 10×30 .
2. u' runs from approximately 0.000 to 2.030 in steps of 0.070.
3. a runs from 0.0 to 0.9 in steps of 0.1.
4. A total of 629 modes were generated.

3.5.5.2 Final Test Dataset 10×30

And for the final test set:

1. An approximately uniform grid of 10×30 .
2. u' runs from approximately 0.035 to 2.065 in steps of 0.070.
3. a runs from 0.0 to 0.9 in steps of 0.1.
4. A total of 629 modes were generated.

3.5.6 Reparameterisation

In EMRI waveforms, the waveform generated when the small mass is very close to the SMBH (low p) will be more complicated compared to the waveform during the start of inspiral. Therefore, we would require more points in the lower p region where the data varies more rapidly. However, methods such as interpolation do not work well on a non-uniform grid where more points are generated at low p . Though one solution is to generate more points to cover the grid, it is not ideal. Hence, we will introduce new parameters, a transformation of p , to produce a uniform grid with more points in the complicated waveform regions. As such, we have introduced u which takes the form:

$$u = \frac{\ln(p - p_{\text{ISCO}} + 1)}{0.23979} \quad (3.1)$$

where p_{ISCO} stands for the semilatus rectum distance p at its innermost stable circular orbit (ISCO) and is obtained by solving **Eq. (2.27)** with the condition ($\iota = 0$) as shown in [25].

We obtain:

$$p_{\text{ISCO}} = 3 + Z_2 - [(3 - Z_1)(3 + Z_1 + 2Z_2)]^{\frac{1}{2}} \quad (3.2)$$

$$Z_1 = 1 + [1 - a^2]^{\frac{1}{3}} \left[(1 + a)^{\frac{1}{3}} + (1 - a)^{\frac{1}{3}} \right] \quad (3.3)$$

$$Z_2 = [3a^2 + Z_1^2]^{\frac{1}{2}} \quad (3.4)$$

If we were to use a uniform grid in u ranging from 1 to 10 in steps of 1, we can observe from **Figure 3.2** that more points are squeezed into the smaller p region, which is precisely the transformation we require.

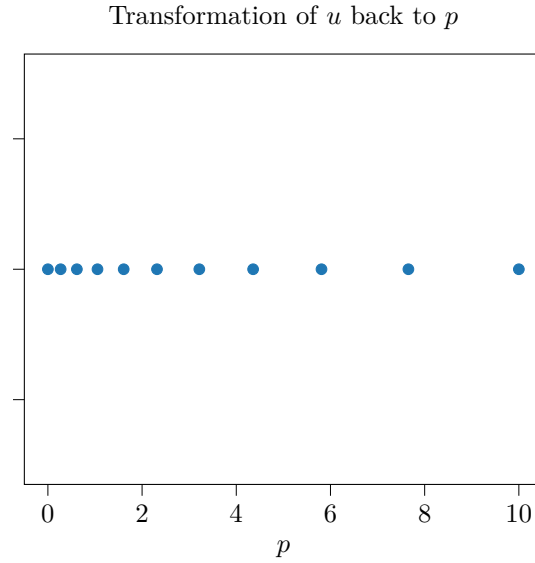


Figure 3.2: An example of how our transformation enables more points in smaller p values. For u ranging from 1 to 10 in steps of 1, these will be the corresponding p values we obtain. A uniform grid in u corresponds to a non-uniform grid in p , resulting in more points in smaller p .

In our final dataset, the parameter transformation is given by:

$$u' = \ln\left(\frac{p - p_{\text{ISCO}} + 3.95}{4}\right) \quad (3.5)$$

Similarly, this transformation would result in more points being located in smaller p regions in a uniform grid of u .

Different choices could be used; for instance, a grid with a different algorithm for increasing density near p_{ISCO} was utilised in [36]. The choice of parameterisation will be further

discussed in Chapter 5.

3.6 Error Metrics

To evaluate our results, we will be using two main error metrics. We are primarily concerned with errors in the phase space of the waveform, which is the most important aspect of EMRI modelling. The waveform evolves over time, and errors accumulate, eventually becoming significant over thousands of cycles. Therefore, we want to focus on the direction of the vector rather than its magnitude.

Considering only the final dataset, the Teukolsky amplitudes are expressed on a uniform grid in $\{u', a\}$. Thus, we can express our vectorised amplitudes as $A_i = \text{vec}(A_{lm}) \in \mathbb{C}^{629} \cong \mathbb{R}^{1258}$. These are the vectorised amplitudes in mode space that we will consider when presenting our results. Similarly, we can express our amplitudes in parameter space as $A_{jk} = \text{vec}(A_i(u'_j, a_k)) \in \mathbb{C}^{300} \cong \mathbb{R}^{600}$. Once we have the vectorised amplitudes, we can compute the normalised inner dot products of the vectors, essentially looking at the difference in the angles between two vectors. Thus, our first error metric is the mode-distribution error [36]. This metric finds the difference in the vector angles between the analytical model's amplitude A^{ana} and our dataset's amplitude A^{num} .

$$\text{Mode-Distribution Error} = 1 - \frac{\Re(\langle A^{\text{ana}}, A^{\text{num}} \rangle)}{|A^{\text{ana}}||A^{\text{num}}|} \quad (3.6)$$

Similarly, our second error metric quantifies the agreement between any pair of waveforms by measuring the overlap, also known as the mismatch [6].

$$\text{Mismatch} = 1 - \frac{\langle h_1, h_2 \rangle}{\sqrt{\langle h_1, h_1 \rangle \langle h_2, h_2 \rangle}} \quad (3.7)$$

The mismatch measures how ‘out of phase’ the two waveforms are and is dominated by the amplitude error [36]. This metric also directly indicates the fraction of signal strength that can be recovered, making it important for the study of EMRI and the LISA mission.

4. Results

4.1 Neural Network

4.1.1 Toy Dataset 10×10

In the preliminary study of this project, the NN was tested on the toy dataset. The details of the toy dataset have already been mentioned in § 3.5. The results obtained from the preliminary run have shown the success of the NN. However, the accuracy could still be improved. Nonetheless, with a small dataset, it has indeed shown potential in the usage of NN in generating fits for EMRI waveforms.

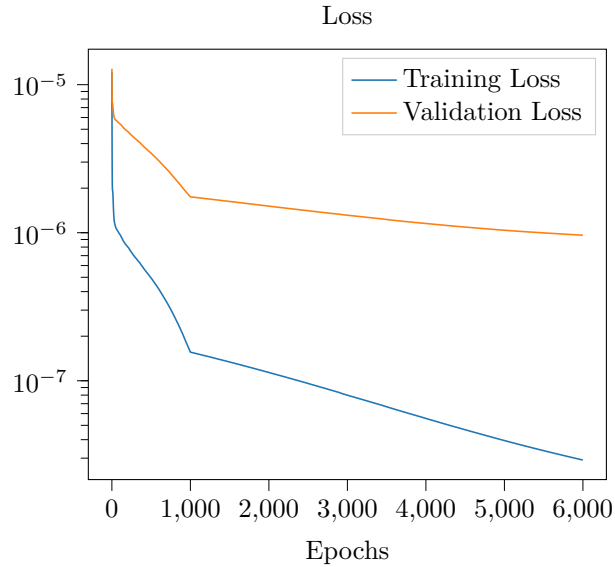


Figure 4.1: The mean-squared loss error plotted for a total of 6000 epochs. The training data used was the toy dataset. Validation loss is obtained by holding out 20% of the data sent to be trained for validation. As the epoch increases, we can see that the loss graphs eventually plateau, indicating that the training has reached saturation.

In the initial stage of creating the model for our NN, plotting out the loss against epoch graph such as **Figure 4.1** is always important in determining if our NN is working. An indication of successful training would be seeing the loss graph decreasing until it plateaus. In our case, we are not too worried about overfitting as with most cases with EMRIs, the fit generated by NN would underfit more often than overfit due to the high dimensionality of EMRI waveforms. With a desired model coded up, as seen in **Appendix C.2**, we are

able to test our model.

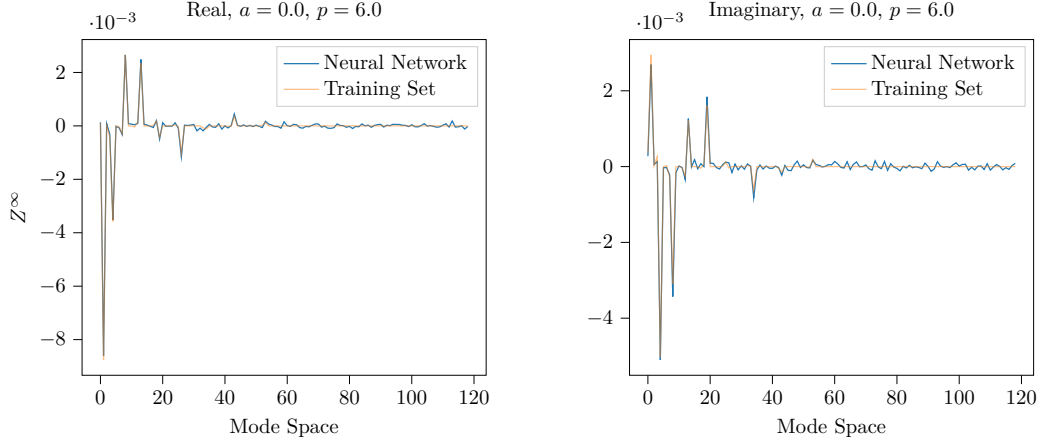


Figure 4.2: The Teukolsky ‘to infinity’ amplitudes of the real (left) and imaginary (right) components are plotted against the mode space for $a = 0.0$ and $p = 6.0$. The mode space is akin to the indexing of the modes, whose order goes according to the summation as seen in **Eq. (2.49)**. As such, the zeroth index would represent $l2m1$, the first index would represent $l2m2$, etc. The NN fit (blue) captures most of the large amplitude peaks in the mode space. However, in the higher mode regimes, the noise created by the NN fit is expected.

To begin, owing to the structure of our training results output, it is easier for us to compare the amplitudes generated across the mode space. We have constructed **Figure 4.2** to enable a visual comparison between how our NN performed and the amplitudes of the training dataset. By visual inspection, it is evident that our NN has effectively captured the major ‘spikes’ - a good indication that it has been trained. In the higher modes regime, we can see that the noise created by our NN is visually observable compared to the training amplitudes. A potential reason for this is that the higher modes regime contains very small amplitudes. As the NN is providing fit across all modes, it is understandable why it would fail when the amplitudes become small. The NN is attempting to fit compensation across all modes, thus creating this noise in the higher mode regions.

We used the mode distribution error as our error metric, as mentioned in **Eq. (3.6)**. For $a = 0.0$ and $p = 6.0$, we obtained an error of 4.3633×10^{-3} . As this is a toy dataset, we did not extend our calculation to the entire parameter space, as the primary purpose of the dataset is to construct an NN model.

4.1.2 20×21 Training Dataset and 20×14 Test Dataset B

In our secondary approach, we increased the training set to a 20×21 uniform grid, which is essentially 4.2 times larger than our toy dataset. In addition, we generated a test set that is a 20×14 uniform grid to evaluate our results.

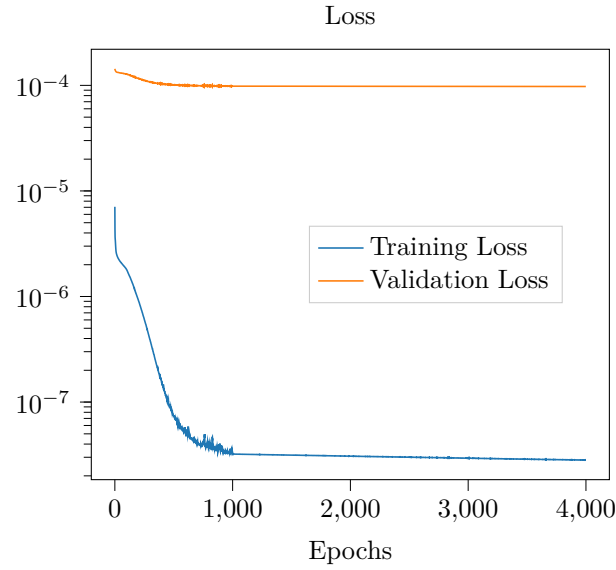


Figure 4.3: The mean-squared error loss plotted against a total of 4000 epochs. In this new dataset, the NN finishes training early on. A validation set is created by holding out 20% of the training data.

By training and testing on a different dataset, we can accurately determine the effectiveness of our NN. As with our previous dataset, we plotted **Figure 4.4** to visually inspect if the NN captured most of the prominent spikes. In fact, we can see that for the case of $\{0.2, 0.43\}$, the fit generated by our NN is particularly good.

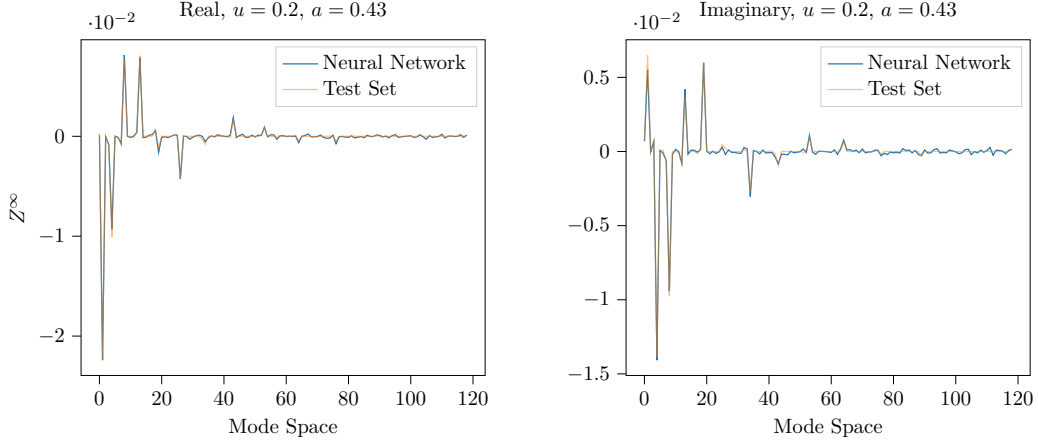


Figure 4.4: The Teukolsky ‘to infinity’ amplitudes of the real (left) and imaginary (right) components are plotted in the mode space (a total of 119 modes) for the set of parameters $\{u, a\} = \{0.2, 0.43\}$.

There are multiple ways to visualize our results. We can also plot our amplitudes in the parameter space to visualize the potential errors in our NN.

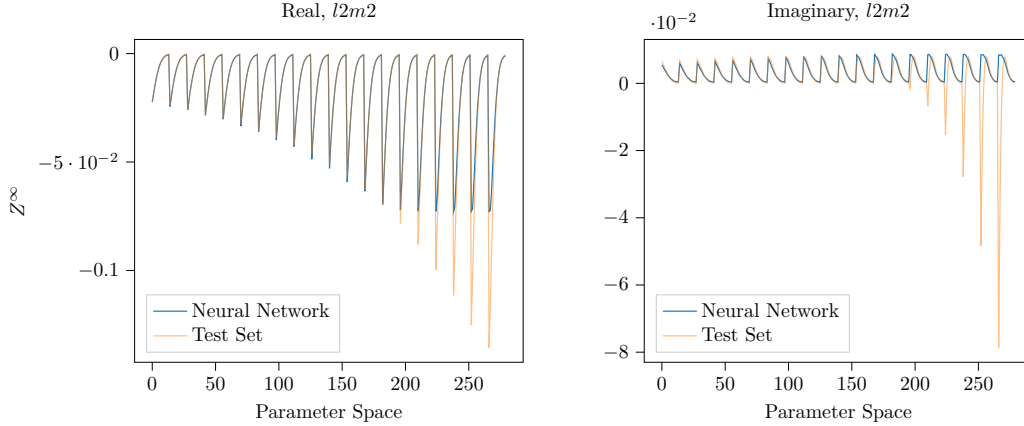


Figure 4.5: The Teukolsky ‘to infinity’ amplitudes of the real (left) and imaginary (right) components in the parameter space index. The parameter space is indexed from increasing u to increasing a . Therefore, when one loop of u finishes, a will then increase, e.g., $\{0.2, 0.43\}, \dots, \{9.95, 0.43\}, \{0.2, 0.455\}, \dots$ and so on. A total of 280 sets of parameters are present. At high parameter space, which indicates high spin regions, the model fails to capture the amplitudes well. The mode chosen is $l2m2$.

By changing our visualization to the parameter space, we can identify potential issues that our NN may have. It is clear that there might have been some problems when it comes to fitting. However, because of our error metric, we are only concerned with the mode distribution error shown in **Eq. (3.6)**.

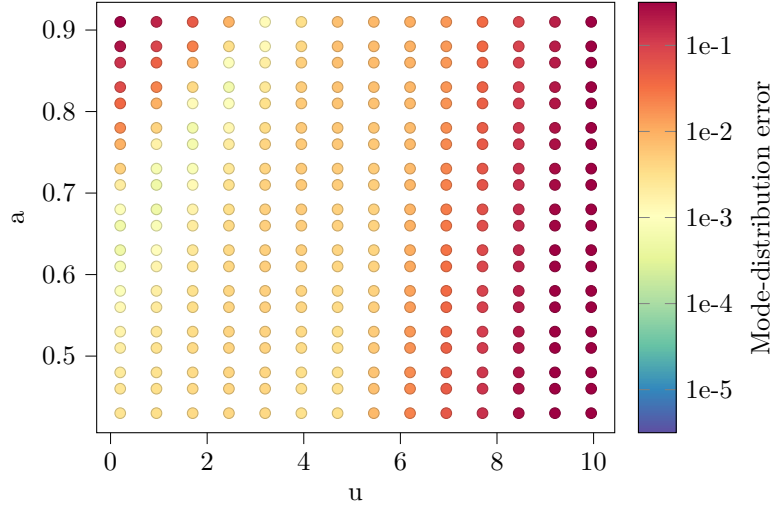


Figure 4.6: Mode distribution error of the Teukolsky ‘to infinity’ amplitudes relative to the test set. The error plot is plotted on the entire uniform parameter space grid. We can see that in regions of high a and low u , as well as regions of high u , the error becomes drastically high. More will be discussed in Chapter 5.

An error plot was constructed over the entire parameter space of the test set, as seen in **Figure 4.6**. By truncating the worst-performing regions, it is still possible to conclude that our NN produces results of decent accuracy. In fact, we have some points on the grid with errors less than 1×10^{-3} .

The failure of our NN in regions of high error is investigated in **Figure 4.7**, which was constructed to investigate one of the reddest regions in **Figure 4.6**.

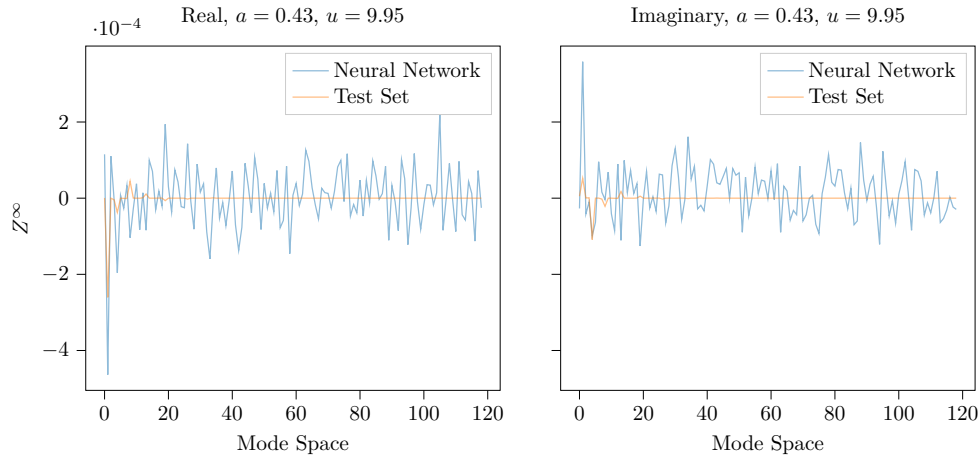


Figure 4.7: The Teukolsky ‘to infinity’ amplitudes of the real (left) and imaginary (right) components plotted in mode space for parameters $\{9.95, 0.43\}$. As the amplitudes for higher p , and therefore u , are significantly smaller, the noise due to the NN will be much more significant, as shown. This would drastically reduce the accuracy of our NN model.

It is clear that the reason for the large error is the noise created by our NN. Further discussion will be presented in Chapter 5.

4.1.3 Final Training Dataset 10×30 , Final Test Dataset 10×30

In our final dataset, which was described in § 3.5.5, we employed K-Fold validation. The results we obtained were much more promising compared to other runs.

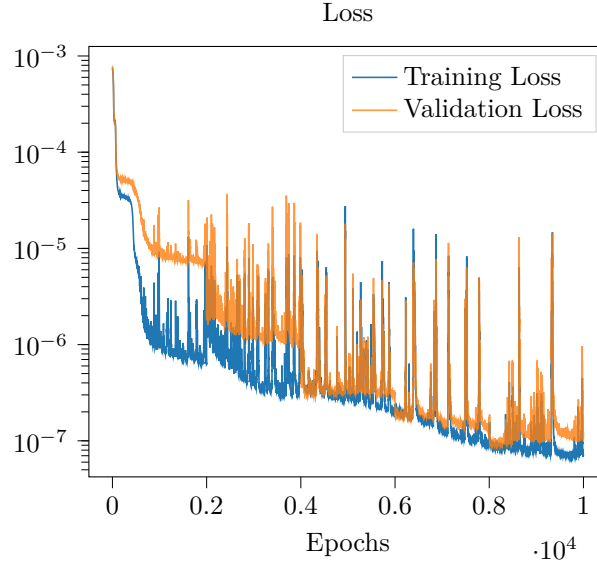


Figure 4.8: The mean-squared error loss plotted against a total of 10,000 epochs. K-fold validation is used in this NN model, where 20% of the training data is used for validation for each fold. We can also observe that the validation loss and the training loss converge near the end of the epochs, which is an indication of no overfitting and underfitting. Generally speaking, this is a well-trained NN.

The loss graph we obtained this time shows promising results of a well-trained NN. As a follow-up, we plotted **Figure 4.9** and **Figure 4.10** to visualise how our NN performed.

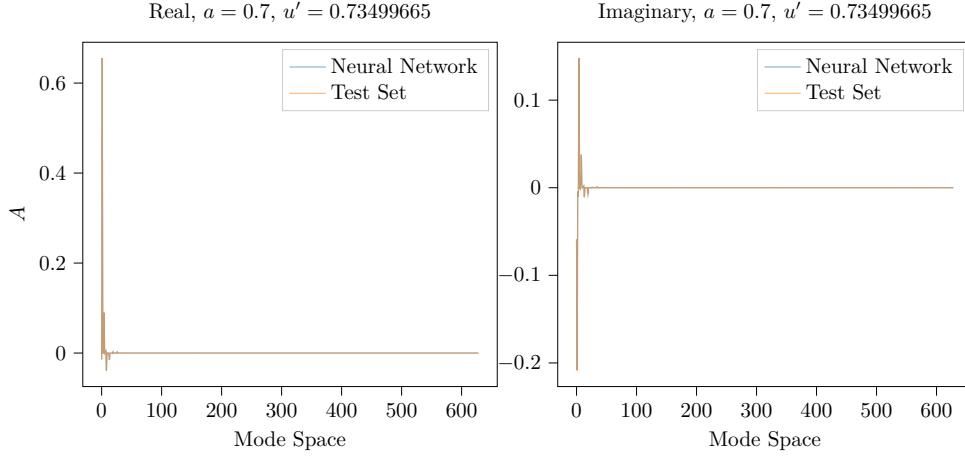


Figure 4.9: The Teukolsky ‘to infinity’ amplitudes of the real (left) and imaginary (right) components are plotted in mode space index. In this new dataset, there are a total of 629 modes, with the first few modes contributing the most to the amplitudes of the waveform. The chosen parameter space is $\{0.734996, 0.7\}$, which is an arbitrary choice. Upon initial inspection, the NN seems to fit quite nicely to our test set.

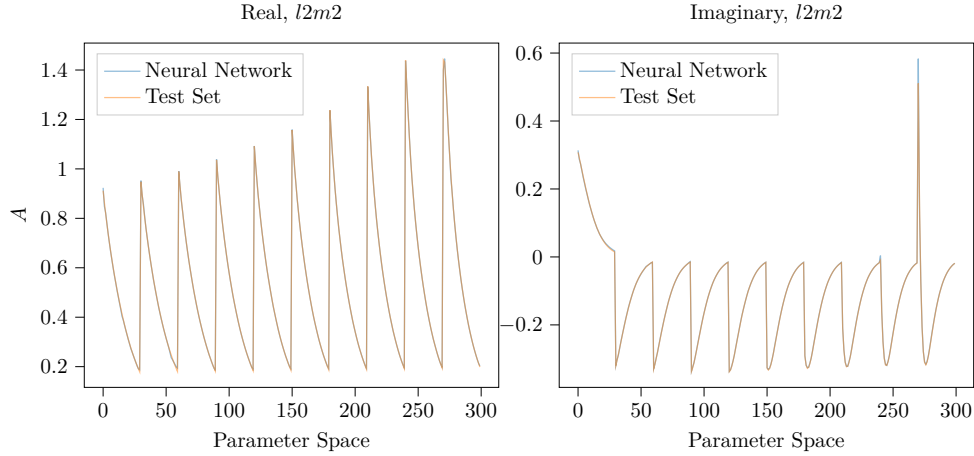


Figure 4.10: The Teukolsky ‘to infinity’ amplitudes of the real (left) and imaginary (right) components are plotted in parameter space index. The indexes for parameter space loop through u' first and then a . The chosen mode for the plots is $l2m2$, and a total of 300 sets of parameters are present. By initial visual inspection, the NN fit (blue) and the test set (orange) appear to coincide well.

So far, the results are promising, and we have not encountered any major issues. Next, we plotted the mode-distribution error to visualise the individual error at each parameter.

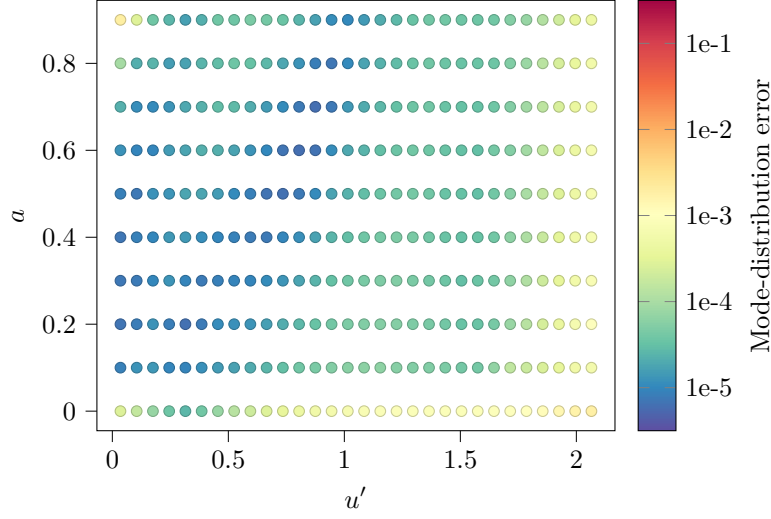


Figure 4.11: Mode-distribution error of the Teukolsky ‘to infinity’ amplitudes relative to the test set. The error plot is plotted on a uniformly transformed parameter space grid. The error obtained by our NN fit is significantly better, with some of the best-performing points having errors of less than 1×10^{-5} and the worst-performing point at only 1×10^{-3} .

The results we obtained are of desirable accuracy. Using **Eq. (3.7)**, we plotted **Figure 4.12**, which shows the amplitude mismatch of the waveform snapshot produced from that particular geodesic.

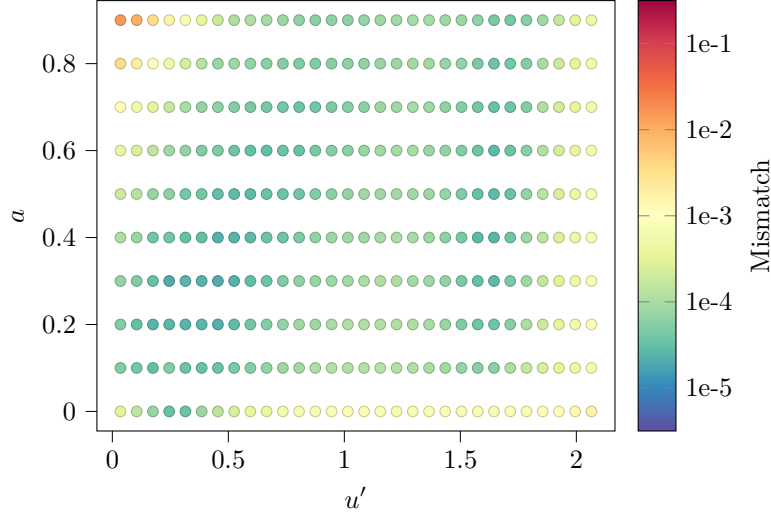


Figure 4.12: Amplitude mismatch between our NN-fitted waveform and the test set constructed waveform from $\{u'_0, a\}$ to $\{u', a\}$ with $M = 10^6 M_\odot$. These results are for $\{\vartheta, \varphi\} = \{\frac{\pi}{3}, \frac{\pi}{4}\}$, but do not depend strongly on the viewing angles. In our worst case, we have approximately a 1×10^{-2} mismatch.

Furthermore, we have investigated the worst-performing point in **Figure 4.12**. The point $\{0.03500511, 0.9\}$ was chosen for investigation by plotting the snapshot waveform. By

expressing u' back into p using **Eq. (3.5)**, we obtain $\{2.51338304, 0.9\}$, which is a particular geodesic right before its plunge into the SMBH.

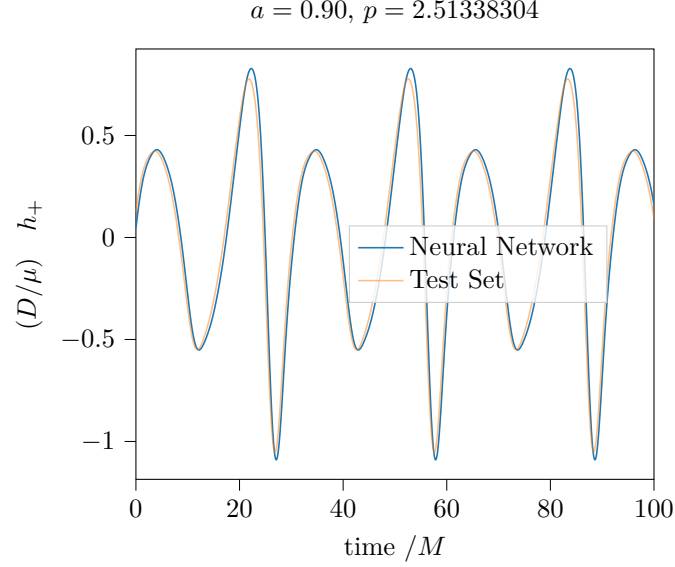


Figure 4.13: A waveform snapshot that was constructed for the geodesic $\{2.51338304, 0.9\}$ with $M = 10^6 M_\odot$ using amplitudes from the NN fit (blue) and from the test set (orange). The result is for $\{\vartheta, \varphi\} = \{\frac{\pi}{3}, \frac{\pi}{4}\}$. The waveform is plotted over a period of $100M$, and this particular geodesic would show a late-stage EMRI waveform.

With **Figure 4.13** plotted, we can visually see the amplitude mismatch. However, as the amplitudes contain both real and imaginary terms, closer to the plunge, this can also lead to a phase mismatch, which can be detrimental to a waveform model. If we were to zoom into the plot, there are hints of phase mismatch present too. However, the error is acceptable for our purpose. More will be discussed in Chapter 5.

Overall, the NN has proven to be an effective and successful tool for generating amplitudes to construct our waveform models.

4.2 Interpolation

In our next investigation, we also probed the effectiveness of interpolating our amplitudes over the parameter space.

4.2.1 20×21 Training Dataset, 10×9 Test Dataset A

Details of the format of the training dataset and test set A can be found in § 3.5. The methodology of interpolation can also be found in § 3.3. After running through the steps of interpolating, we obtain our splines. The splines were then evaluated at each point in our

test set's parameter space. Since our test set parameters are $\{p - p_{\text{ISCO}}, a\}$, we transformed our parameters into $\{u, a\}$ using **Eq. (3.1)** since our splines were constructed in the $\{u, a\}$ domain.

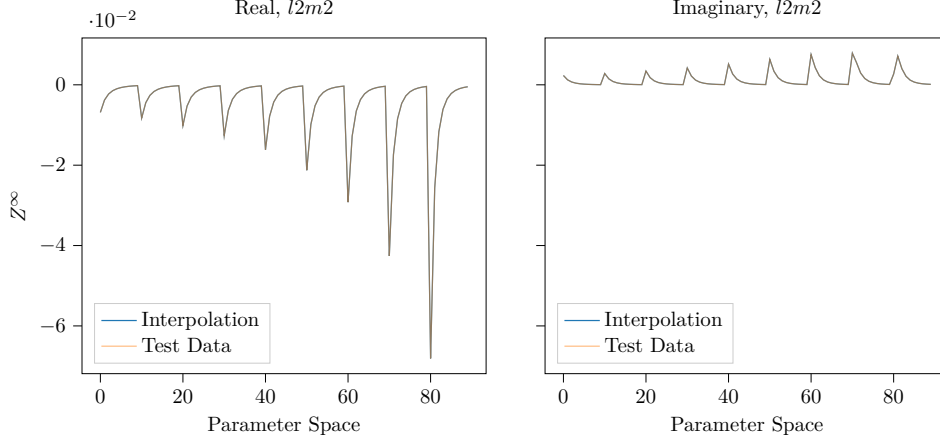


Figure 4.14: The Teukolsky 'to infinity' amplitudes of the real (left) and imaginary (right) components in the parameter space index. The parameter index loops through u first, then a , and a total of 90 sets of parameters are present.

The results we obtain from interpolating seem to interpolate very nicely on our test set. Upon visual inspection, it seems almost like a perfect match. Further investigation is needed by plotting out the mode-distribution error over the entire parameter space as shown in **Figure 4.15**.

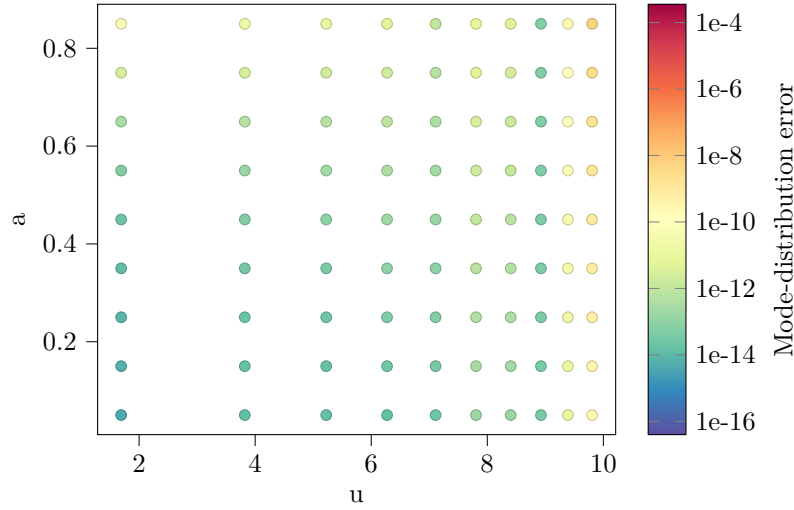


Figure 4.15: Mode-distribution error of the Teukolsky 'to infinity' amplitudes relative to the test set. Since we have used a test set that was transformed from $p - p_{\text{ISCO}}$ to u , we can see that due to the transformation, more points are accumulated in high u regions. The performance of the interpolation did much better than our NN, with the worst-performing parameter only having an error of approximately 1×10^{-7} .

With a non-uniform grid, we already achieved a worst error of 1×10^{-7} . This shows more promising results than our NN. Therefore, our next target is to investigate the performance of this method in our final dataset.

4.2.2 Final Training Dataset 10×30 , Final Test Dataset 10×30

We started by constructing our splines using the final training dataset, and then used these splines to fit over the parameter space of the final test dataset for the amplitudes. We compared the obtained amplitudes to our final test dataset and present the results in **Figure 4.16**.

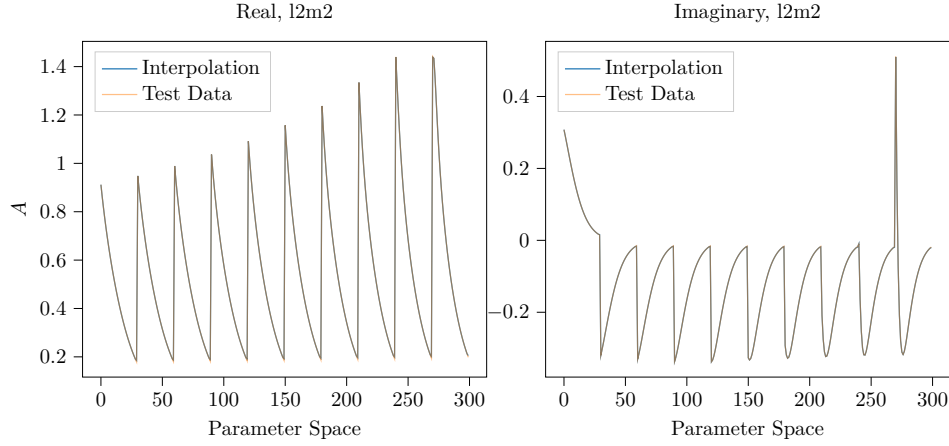


Figure 4.16: The Teukolsky ‘to infinity’ amplitudes of the real (left) and imaginary (right) components plotted in parameter space index. Visually, the two interpolations (blue) and the final test set (orange) seem to coincide perfectly.

We then constructed the mode-distribution error, and subsequently the mismatch, as shown in **Figures 4.17** to **4.18**. We observed a line of points, specifically $u' = 2.065$, with very high errors, which will be discussed in Chapter 5.

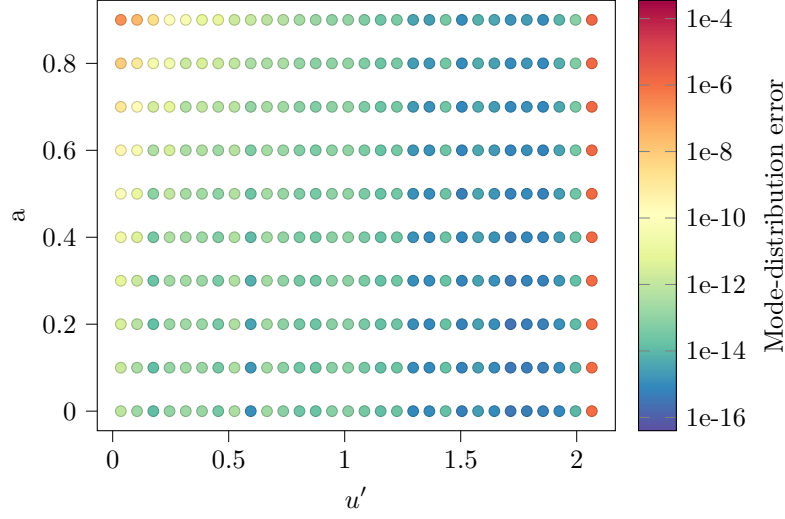


Figure 4.17: Mode-distribution error of the Teukolsky ‘to infinity’ amplitudes relative to the final test set. A drastic increase in error can be seen for $u' = 2.065$, as those points fall out of our interpolation range. We can also see that the error increases for high a and low u' . By ignoring the points that exceed the range, we see that our worst performing error is approximately 1×10^{-8} , which is by far the best model we have produced.

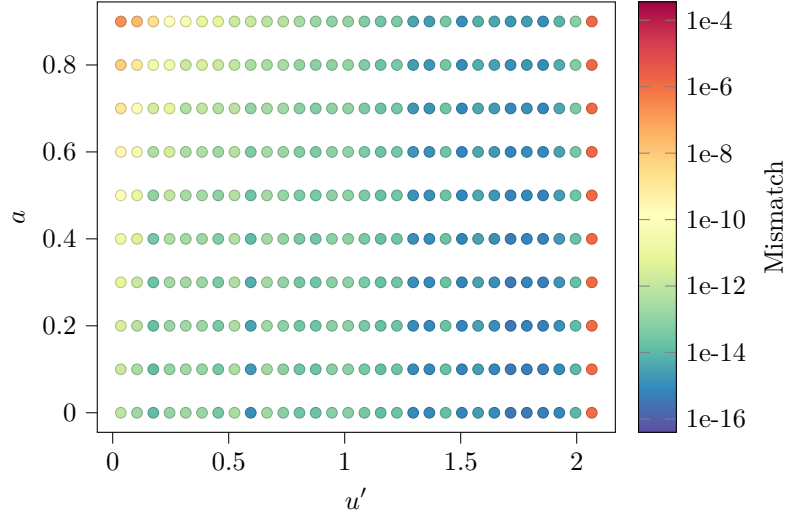


Figure 4.18: Amplitude mismatch between our interpolated waveform and the test set’s constructed waveform from $\{u'_0, a\}$ to $\{u', a\}$ with $M = 10^6 M_\odot$. These results are for $\{\vartheta, \varphi\} = \{\frac{\pi}{3}, \frac{\pi}{4}\}$, but they do not depend strongly on the viewing angles. In our worst case, we have an approximate mismatch of 1×10^{-7} , ignoring points outside of the interpolation range.

Selecting one of our worst-performing points, $\{u', a\} = \{0.035, 0.9\}$, we constructed a waveform snapshot (**Figure 4.19**). However, despite being the worst-performing point, the constructed snapshot does not have any visible flaws.

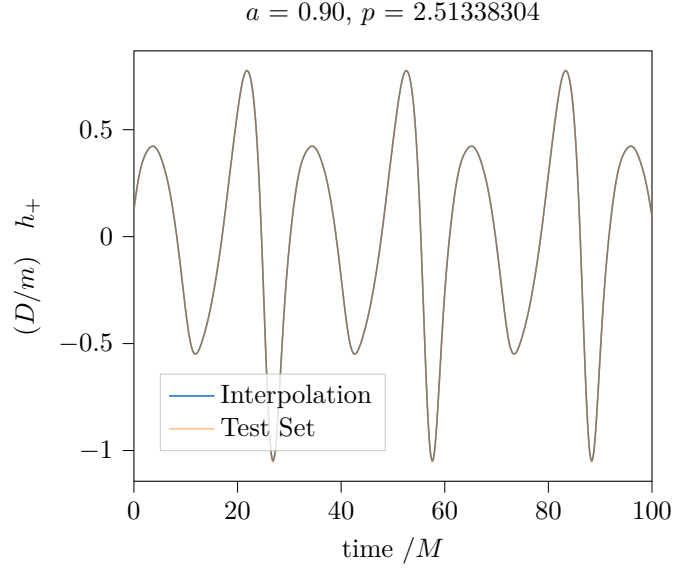


Figure 4.19: Waveform snapshot constructed for $\{2.51338304, 0.9\}$ with $M = 10^6 M_\odot$ using amplitudes from the interpolated dataset (blue) and the test set (orange). The result is for $\{\vartheta, \varphi\} = \{\frac{\pi}{3}, \frac{\pi}{4}\}$. The waveform is plotted over a period of $100M$.

Therefore, we can conclude the efficiency and effectiveness of interpolating through the parameter space, which has so far surpassed the performance of our NN.

4.3 Polyfit

We have decided to investigate the effectiveness of polyfitting in our project. Polyfitting came with its own challenges which have already been mentioned in § 3.4. Therefore, one of the first preliminary investigations of polyfitting is to fit the correct polynomials to our waveform. After we found the polynomials that give the desired accuracy to our result, we then performed the fit on our final dataset.

4.3.1 Final Training Dataset 10×30 , Final Test Dataset 10×30

Figure 4.20 was plotted for the preliminary visual check of our fit compared to the test set. In this case, we see that the points in the matrix for our polyfitted amplitudes match well with the final test set's amplitudes.

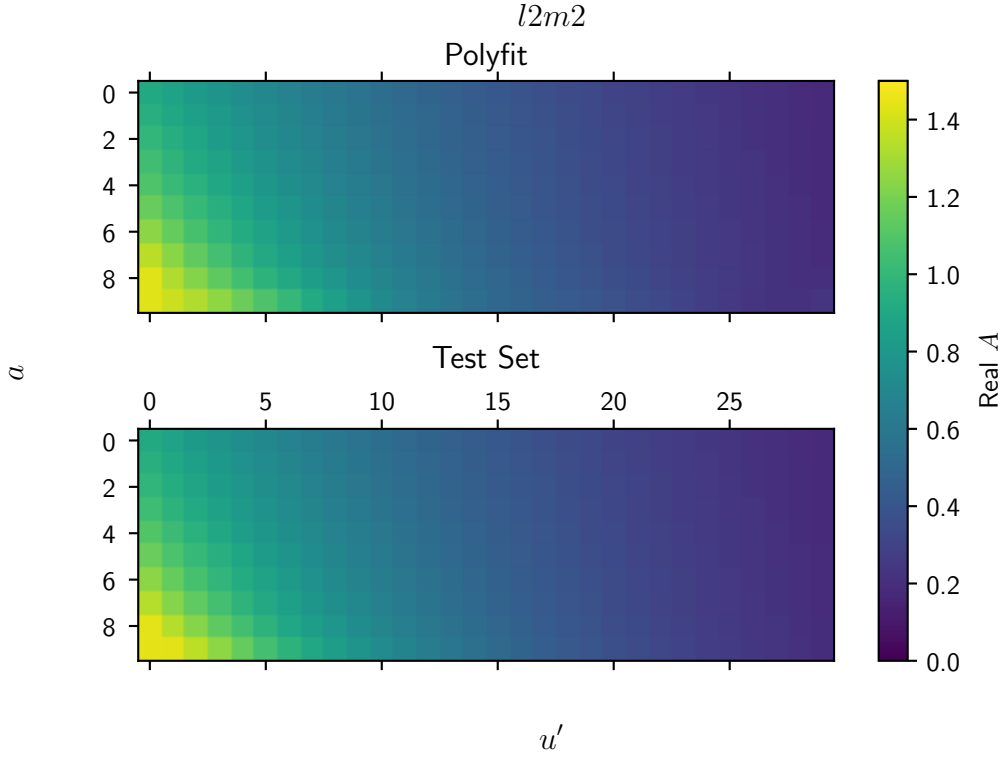


Figure 4.20: Matrix plot of the parameter space for the Teukolsky's real amplitude. The chosen mode is $l2m2$. The matrix plot is useful as we are able to have a first-hand gauge on how our polyfit performed compared to the test set.

As **Figure 4.20** matches our results so well, it is not surprising that **Figure 4.21** shows no visual discrepancy between our polyfitted amplitudes and the test set's amplitudes.

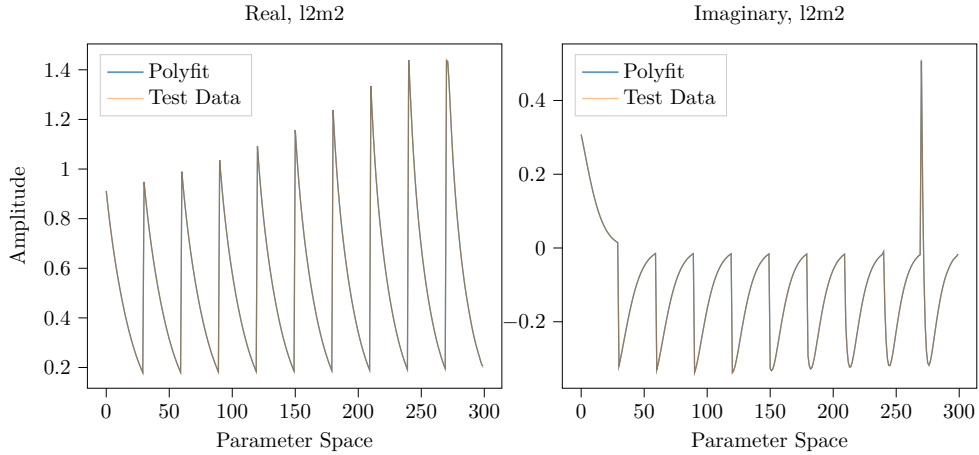


Figure 4.21: The Teukolsky 'to infinity' amplitudes of the real (left) and imaginary (right) components plotted in the parameter space index. This is for polyfit, and the chosen mode is $l2m2$.

From our mode-distribution error, we can see that the polynomials chosen for our polyfit do not fit the high a well. However, the errors fall within the range of around 1×10^{-4} , which still performs better than our NN.

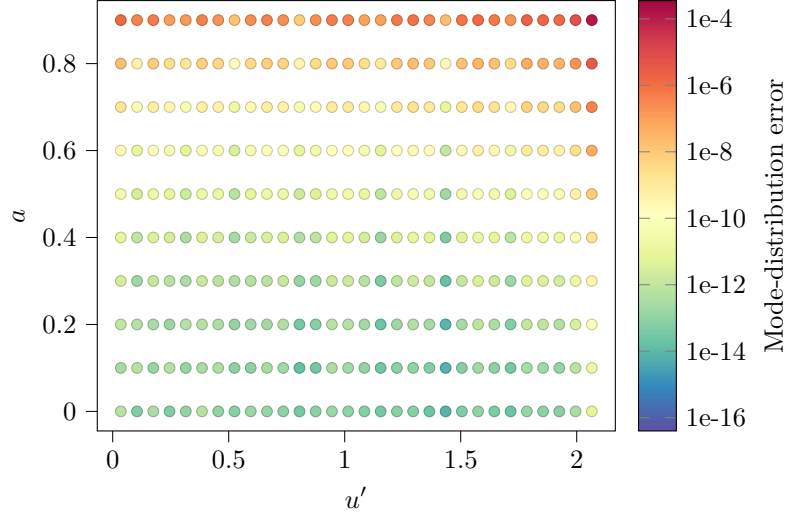


Figure 4.22: Mode-distribution error of the Teukolsky ‘to infinity’ amplitudes relative to the final test set. The polyfit seems to be effective in fitting parameters with low a . The worst performing point has an error of approximately 1×10^{-4} , which still seems to perform better than our NN model.

Although we noticed that the point with the highest error ($\{u', a\} = \{2.065, 0.90\}$), we still chose $\{u', a\} = \{0.035, 0.90\}$ for our waveform snapshot evaluation, as these points were chosen for both NN and interpolation evaluations as well.

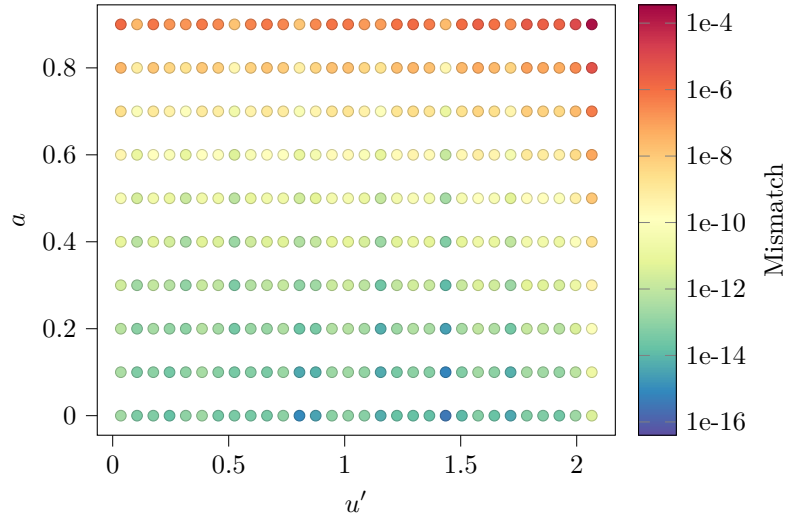


Figure 4.23: Amplitude mismatch between our polyfitted waveform and the test set constructed waveform from $\{u'_0, a\}$ to $\{u', a\}$ with $M = 10^6 M_\odot$. These results are for $\{\vartheta, \varphi\} = \{\frac{\pi}{3}, \frac{\pi}{4}\}$, but the mismatch does not depend strongly on the viewing angles. In our worst case, we have approximately 1×10^{-4} mismatch.

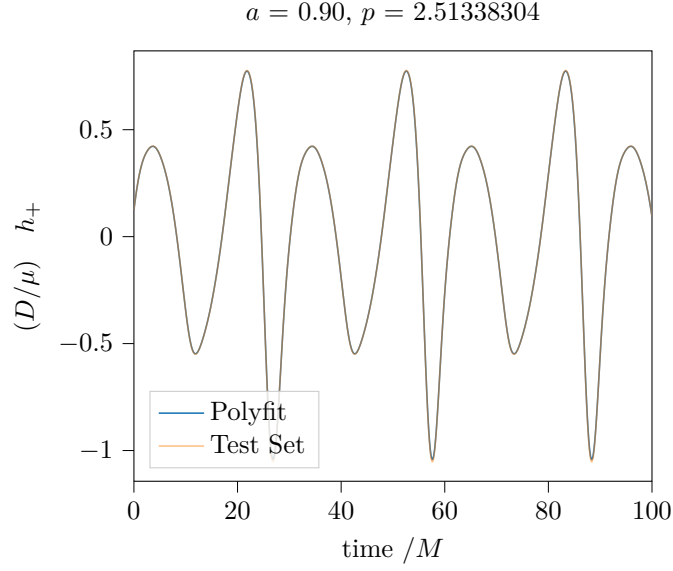


Figure 4.24: Though the point $\{p, a\} = \{2.51338304, 0.90\}$ is not one of the worst-performing points, it is still relatively bad. We still chose this point to allow us to compare it with the other two methods that were mentioned earlier. The waveform snapshot constructed with $M = 10^6 M_\odot$ using amplitudes from polyfit (blue) and the test set (orange). The result is for $\{\vartheta, \varphi\} = \{\frac{\pi}{3}, \frac{\pi}{4}\}$ and the waveform is plotted over a period of $100M$.

Visually, it is difficult to differentiate the two snapshot waveforms. Despite performing worse than our interpolation, an error of 1×10^{-4} still puts us in a region of good accuracy; therefore, the amplitudes fitted by polyfitting are still viable for waveform generation.

4.4 Mismatch of the Three Methods

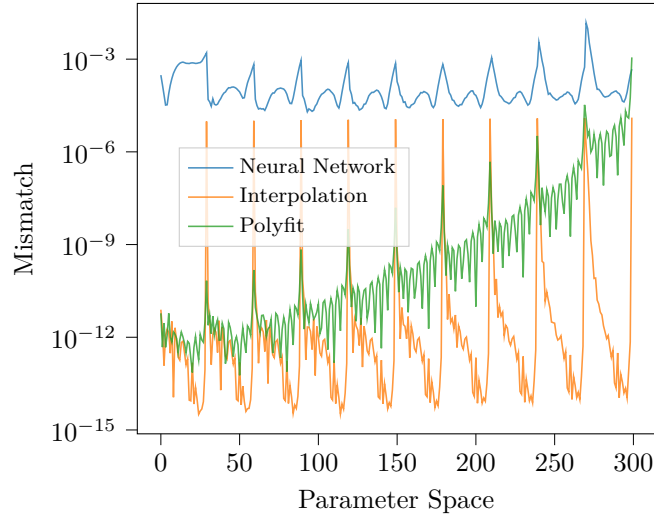


Figure 4.25: The mismatch of the snapshot waveform across the parameter space for the three different methods - NN (blue), Interpolation (orange), and Polyfit (green).

Figure 4.25 shows the mismatch for the three methods across the parameter space. If we remove the points that are out of range for interpolation, we can see that interpolation results in the smallest mismatch and is the clear winner.

4.5 Final Waveform Evolution

In this section, we compare the three methods by constructing the full waveform evolution using **Eq. (2.53)**.

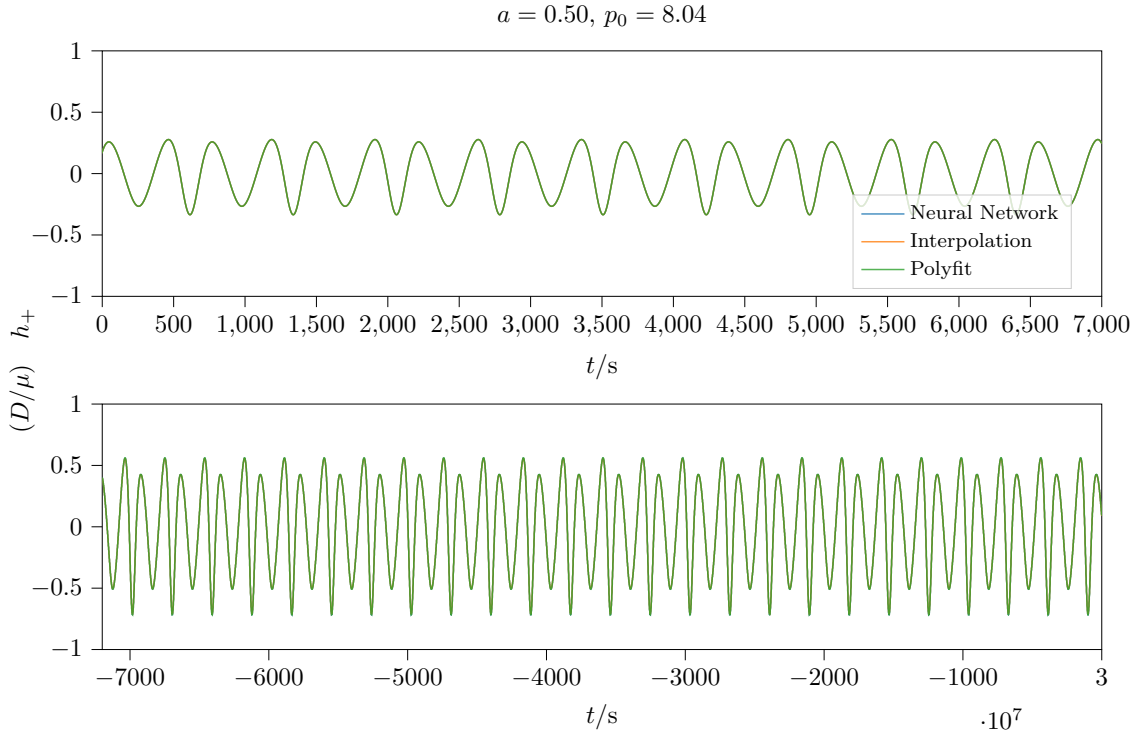


Figure 4.26: We plot the first 2 hours (top) and the last 2 hours (bottom) of the evolving waveform at viewing angle $\{\vartheta, \varphi\} = \{\frac{\pi}{3}, \frac{\pi}{4}\}$, generated by the three different methods: NN (blue), interpolation (orange), and polyfit (green). We chose an arbitrary spin $a = 0.50$ and a p_0 value such that the CO plunges into the SMBH in approximately one year. As computing the full waveform evolution of one year with a good enough resolution is computationally expensive, we only plot the first and last 2 hours of the waveform. Throughout the evolution of the waveform, the three methods coincide nicely, demonstrating that they are all good modelling techniques for a full waveform.

We chose an arbitrary spin $a = 0.50$ and a p_0 value such that the CO plunges into the SMBH in approximately one year. Computing the full waveform evolution of one year with sufficient resolution is computationally expensive, so we only plot the first and last 2 hours of the waveform to compare the waveform evolution throughout the year. As shown in **Figure 4.26**, the waveforms generated by the three methods coincide nicely throughout

the evolution and all seem to be good modelling techniques for a full waveform.

4.6 Computation Speed

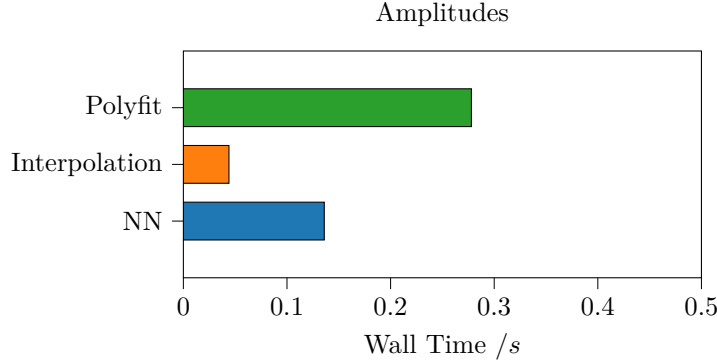


Figure 4.27: Computational wall time for the generation of Teukolsky amplitude by the three different methods. All times were averaged over five evaluations of the final test set parameter space. Evaluations were done on an Intel 2.4GHz Quad-Core i5 CPU.

In our final comparison between the three methods, we examine the computation time for generating the amplitude for our test set. In our previous result for the full waveform evolution, we see that despite having different snapshot waveform mismatches in the three methods, the final full evolving waveform coincides perfectly visually. Hence, another key comparison we perform is to investigate the speed at which they construct the amplitudes for us. Evidently, from **Figure 4.27**, interpolation took the least time to construct the amplitudes and is the clear winner in our study.

The computation speed only takes into account the fitting/constructing process, and we did not include any of the training processes. However, among the three methods, constructing the splines for the amplitudes took the least time, whereas NN took the longest time to train. Hence, regardless of the training time, interpolation is the most efficient and accurate method among the three methods we have investigated so far.

5. Discussion

5.1 Results from the Neural Network

A neural network (NN) can be thought of as a black box. However, our results allow us to understand which variables affect how well a NN trains. These variables are:

1. Learning rate. The learning rate is one of the most important factors when designing how effectively a NN can train. If the learning rate is too high, the weights may be updated too aggressively, causing the optimisation process to overshoot the optimal point and bounce back and forth, resulting in instability and slow convergence. On the other hand, if the learning rate is too low, the optimisation process may be too slow, and the NN may take a long time to converge or even converge to a suboptimal solution.
2. Size of data. An increasingly dense grid allows our NN to train better. With a larger training dataset, the NN has access to more diverse examples and can learn more representative features, resulting in better generalisation performance. It also reduces the chance of overfitting, as the model is less likely to memorise the training data and more likely to capture the underlying patterns.
3. Validation techniques. When we changed our validation approach, we finally observed the convergence between our validation set and our training set, as seen in **Figure 4.8**. If the training loss and validation loss are both decreasing and eventually converge, it means that the model is learning from the training data and also generalizing well to new data. This is a good sign that the model is not overfitting to the training data and is able to make accurate predictions on new data.
4. Reparameterisation. Evidently, when we compare the results of **Figure 4.6** and **Figure 4.11**, we see that the error for **Figure 4.6** is a lot higher. One main difference, besides the validation technique, is how we reparameterised our p to the new parameter. Though we did not investigate carefully how reparameterising our parameters would affect our results, we could potentially tell that there is some sort of correlation to it. A careful investigation should be done to study how the new parameters would affect the training of our results.

Even though **Figure 4.6** can be considered one of our ‘failures’ in training our NN, we can still learn a lot from it. From **Figure 4.6**, we can see that there are some peculiarities in our results from the grid. The NN seems to not fit well in the high spin region (high a), as well as when the CO is far from the SMBH (high p). A possible explanation is that in the high spin regions, the data varies more rapidly. In high p regions, the gravitational waveform is too simple, and since our NN does a fit across all parameters, the noise created to fit the simpler region may become too large, causing the error to increase tremendously. To resolve such issues, we can exclude the regions where our NN fails to produce waveforms with desirable accuracy. We can then train another NN to specifically tackle the high spin region as well as the high p region. Then, we can construct a validity region in which the NN is valid. Thus, when we construct our waveform, it might be possible to use multiple NNs to fit across one full evolving waveform. However, this was not needed, as shown in our final dataset, where our NN has reached the desired accuracy in all regions of the parameter space.

5.2 Results from Interpolation

Interpolation indeed gave us the best result in terms of accuracy as well as speed. However, it came at a cost. For lower-dimensional analysis, interpolation would seem to be the go-to for any numerical modelling studies. However, going beyond two dimensions, interpolating becomes tremendously expensive, if there is even a way to interpolate over higher-dimensional space. Because our work only requires two parameters, p and a , interpolation is a viable method. If we were to extend this study to a generic EMRI waveform containing four parameters, p , e , ι , and a , there is simply no existing way to interpolate all four parameters at once. Even in the `scipy` package for interpolation, the maximum dimension they offer is 2D [38].

In addition, interpolation works best with a uniform grid. The initial parameter p has to undergo some form of reparameterisation to u before we can interpolate smoothly without errors. Therefore, a deeper analysis has to be done on how we reparameterise p to investigate any potential errors that can stem from it.

We can also observe that interpolation only works well within the range of the data given. This is evident from **Figure 4.17** and **Figure 4.18**, where the last point in u' has a spike in error. Therefore, to consider the validity of our results, we will exclude the points with

$u' = 2.065$.

5.3 Results from Polyfit

Polyfit is one of the most unconventional methods out of all. This is mainly because we took a guess at determining the best degree for the polynomials for the fit. Though it is possible to investigate how the degree of the polynomial would affect the fit, it would be computationally expensive as the computation time increases with the degree of polynomial. Thus, ultimately, it is just a good guess at the degrees of polynomials we will be using. Furthermore, there is not any existing convenient way of fitting 2D polynomials with existing packages, thus we had to use some methods that exist elsewhere to aid us in doing the fit. More studies could have been done in implementing a more efficient polyfit process for EMRI, and then we could possibly cut down on some extra computational time. More studies could have also been done to investigate the correct degree of the polynomial needed to fit our amplitudes across the parameter space.

5.4 Full Waveform Evolution

We have decided to use $a = 0.5$ for our full waveform evolution. The worst mismatches along the evolution snapshots for each of the three methods are: 1×10^{-3} , 1×10^{-10} , and 1×10^{-8} for NN, Interpolation, and Polyfit, respectively. However, when we examine **Figure 4.26**, there does not appear to be any discrepancy between the three methods. This is mainly because, upon investigating the order of waveform mismatches, [36] has found that, on a crude analysis, a mismatch of 1×10^{-4} is sufficient to prevent the waveforms from deviating significantly from each other. In another study [39], it was determined that having a mismatch of 1×10^{-2} would be adequate.

Our NN snapshot waveform mismatch lies in the range of 1×10^{-3} , which is evidently accurate enough to successfully construct a full OPA adiabatic EMRI waveform. Therefore, all of our models have sufficient accuracy to successfully construct a full OPA adiabatic EMRI waveform.

5.5 General Discussion

Our results have ultimately shown that interpolation should be the main method for constructing a circular equatorial orbit EMRI waveform, as it is the fastest and most accurate method. However, it is likely that we cannot use this method to construct a

generic waveform, as interpolation does not scale well with dimensions. Theoretically, NN should be able to scale well with dimensions, and thus, for generic EMRI waveforms, NN would be a viable method. However, we have not come across any research that trained an NN with all four parameters $\{p, e, \iota, a\}$ at once. Hence, it may be worthwhile to investigate if training an NN with all parameters would produce a waveform of desirable accuracy.

Ultimately, for the purpose of this project, we will be using interpolation for future similar work.

5.6 Future Work

The most immediate task we can undertake is implementing our circular equatorial waveform template model into **FastEMRIWaveform**, an open-source package for plotting EMRI waveforms and containing useful tools [39].

Regarding improvements to our work, we suggest considering the following:

5.6.1 Trajectory

As we use a 3PN- e^6 approximation to construct our trajectory, we are essentially making an approximation on top of another approximation (0PA). However, according to the definition of an adiabatic waveform, the trajectory's proper construction involves using $\{E, L_z, Q\}$.

At each instant of p , there exists a corresponding set of $\{E, L_z, Q\}$. Using these values, we can obtain both ψ_4 and Z^∞ . To determine the backreaction to our system, $\left(\frac{dE}{dt}\right)^\infty$, we can use the following formula [5]:

$$\left(\frac{dE}{dt}\right)^\infty = \sum_{lmkn} \frac{|Z_{lmkn}^\infty|^2}{4\pi\omega_{mkn}^2}. \quad (5.1)$$

With the backreaction, we could evolve our system over a small time step and form our next set of $\{E_{\text{new}}, L_{z,\text{new}}, Q_{\text{new}}\}$ and obtain the backreaction to the new parameters. Continuing this method, we could theoretically obtain a trajectory for the full waveform. Thus, we could obtain a flux-driven trajectory in this way.

Furthermore, since $\left(\frac{dE}{dt}\right)^\infty$ is obtainable through **GREMLIN**, we can generate a series of $\left(\frac{dE}{dt}\right)^\infty$ values across the parameter space and interpolate between them. By doing so, we can save computational time compared to solving for $\left(\frac{dE}{dt}\right)^\infty$ directly. In this way, we could generate a full waveform trajectory without relying on PN approximations.

5.6.2 Constructing a more accurate waveform

Our project achieves a high level of accuracy up to adiabatic order. However, we note that our current work is limited to the 0PA region and 1PA waveforms, which consider the second-order self-force term, are expected to be more accurate. Generating waveforms using 1PA waveforms would be the next step in improving the accuracy of our work. However, 1PA waveforms are still a work in progress and lie beyond the scope of this project.

5.6.3 Generic waveform

Constructing a fully generic EMRI waveform is challenging due to the large number of modes and dimensions involved, making it difficult to use interpolation. Nevertheless, we can still investigate different sets of 2D parameters when generating the waveform using interpolation. For example, we can study the effect of varying $\{p, e\}$ or $\{e, \iota\}$ while keeping the other two parameters constant. By limiting the number of dimensions to ≤ 2 , we can use interpolation to create new waveform possibilities. It is also possible to connect the different parameters themselves through the use of splines. This approach helps us address the dimensionality problem and apply the best method (interpolation) to investigate new waveform possibilities.

5.6.4 Spin of Small Mass

In all of our work, we have not considered any spin for our small CO. However, applying a spin to the small CO is an ongoing area of research, and there is no concrete answer yet. Adding a spin will result in extremely complex dynamical behaviour within the system, and the ongoing research seeks to better describe such a system.

5.6.5 Environmental Conditions

In actual LISA missions, we will not receive waveforms that are free of noise. Therefore, when creating our templates for waveforms, we must consider environmental factors such as accretion disks [40] or massive perturbers [41, 42]. We need to consider how these factors will affect our waveform eventually.

6. Conclusion

In this project, we have constructed the OPA adiabatic waveform using three different methods: NN, interpolation, and polyfit, by building the trajectory using a 3PN- e^6 approximation. We have concluded that interpolation is the most suitable method to fit over the amplitudes when the dimensions are ≤ 2 , specifically for circular, equatorial orbits, where the modes are significantly fewer than the generic EMRI waveform. Our present waveform model is accurate at the adiabatic order for circular equatorial Kerr orbits, and thus we are now able to construct EMRI search templates with this specific orbit for a supermassive Kerr black hole. However, for actual LISA usage, we must extend our template to a generic waveform model that is sufficiently accurate, and possibly consider environmental factors or even the small mass spin.

With the approaching deadline of the LISA mission, we need to create a large number of waveform templates necessary for the mission. Therefore, the purpose of this project is to provide new waveform templates, albeit for a specific case. This project can also serve as a study that provides insight into other waveform templates consisting of 2D parameters, where we can use the methods developed in this project to construct other waveform templates. Moreover, we can use this project to check the generic model in the limits of a circular and equatorial orbit.

In conclusion, this project has been successful in developing new waveform templates and has opened up possibilities for many other applications.

A. 0PA and 1PA Adiabatic Waveform

A.1 How Accurate is Enough?

Without delving too deeply into analytical analysis, let us first examine the equation of motion with the expansion of the self-force:

$$\frac{D^2 z^\alpha}{d\tau^2} = \epsilon f_1^\alpha + \epsilon^2 f_2^\alpha + \dots \quad (\text{A.1})$$

where $\frac{D}{d\tau} = u^u \nabla_u$, $u^u = \frac{dz^u}{d\tau}$, z^α represents the position of the small mass, and f is the self-force term. We know that μ loses energy at a rate proportional to the gravitational waves carrying energy out of the system. As such, $\dot{E} \propto \text{GW Flux} \propto \dot{h}^2 \propto \epsilon$. But the energy itself is $E \propto \mu \propto \epsilon$. So the relative rate at which the energy is changing is thus $\dot{E}/E \propto \epsilon$ [12]. This defines for us a characteristic timescale at which the energy is changing:

$$\text{timescale of inspiral} = \frac{E}{\dot{E}} \propto \frac{1}{\epsilon}. \quad (\text{A.2})$$

Suppose we only know f_1^α but not f_2^α . The error in $\frac{D^2 z^\alpha}{d\tau^2}$ is then [43]:

$$\delta \frac{D^2 z^\alpha}{d\tau^2} \propto \epsilon^2 \delta z^\alpha \qquad \propto \epsilon^2 \tau^2 \delta z^\alpha \propto \epsilon^0. \quad (\text{A.3})$$

By considering only the first-order self-force term, we will arrive at an error of approximately 1 radian, but we would want our error to be much smaller than 1 radian [12]. An error in position will translate to an error in phase. Thus, an error of order 1 in position is equivalent to an error of order 1 in phase. Hence, the second-order term is needed. The same argument could be used to show why the third term is not needed, as the higher-order terms scale with ϵ , and their contribution will be extremely small.

However, as we are dealing with 0PA waveforms, we will not require any second-order self-force terms. Mino's realization helped us understand that in the adiabatic limit, one only

needs the time-averaged dissipative piece of the first-order self-force [44]. Hinderer and Flanagan’s analysis showed that 1PA waveforms are needed to make precise measurements of binary systems [9]. These waveforms incorporate the second-order self-force term, which produces waveforms more accurate than the 0PA. To obtain a 1PA waveform, one would need both the (not averaged) dissipative and conservative first-order self-force terms, as well as the dissipative second-order self-force term, and this is highly challenging. The computation of 1PA waveforms is at the frontier of the research that is still ongoing, as we still do not have a concrete answer to second-order self-force terms. Furthermore, the number of waveforms for 0PA is limited in quantity, and more work has to be done.

Therefore, for the purpose of the project, even though we are lacking in accuracy compared to other methods of waveform modelling, a 0PA waveform enables us to effectively investigate the methods of modelling which we want to test, as we can produce waveforms using the 0PA approach.

In Chapter 3, we have explored the error metric that we will be using for our study. In Chapter 5, we have also talk about how small an error is sufficient for the purpose of creating a template waveform.

B. Proofs for Theory

B.1 Proofs for Eqn 2.7-2.9

The proof for **Eq. (2.7)**:

$$\begin{aligned}
 C^\alpha_{\beta\gamma}\omega^\gamma &= \tilde{\nabla}_\beta\omega^\alpha - \nabla_\beta\omega^\alpha \\
 &= \partial_\beta\omega^\alpha + \tilde{\Gamma}^\alpha_{\beta\gamma}\omega^\gamma - \partial_\beta\omega^\alpha - \Gamma^\alpha_{\beta\gamma}\omega^\gamma \\
 &= (\tilde{\Gamma}^\alpha_{\beta\gamma} - \Gamma^\alpha_{\beta\gamma})\omega^\gamma \quad \square
 \end{aligned}$$

The proof for **Eq. (2.8)**:

We first choose a local inertia frame (w.r.t $g_{\alpha\beta}$), we have $\Gamma^\alpha_{\beta\gamma} = 0$.

$$\begin{aligned}
 C^\alpha_{\beta\gamma} &= \tilde{\Gamma}^\alpha_{\beta\gamma} \\
 &= \frac{1}{2}\tilde{g}^{\alpha\mu}(\partial_\beta\tilde{g}_{\gamma\mu} + \partial_\gamma\tilde{g}_{\beta\mu} - \partial_\mu\tilde{g}_{\beta\gamma}) \\
 &= \frac{1}{2}\tilde{g}^{\alpha\mu}(\nabla_\beta\tilde{g}_{\gamma\mu} + \nabla_\gamma\tilde{g}_{\beta\mu} - \nabla_\mu\tilde{g}_{\beta\gamma})
 \end{aligned}$$

Since $\Gamma^\alpha_{\beta\gamma} = 0$, the covariant derivative of g will be 0 and since $\tilde{g} = g + h$, we then have:

$$C^\alpha_{\beta\gamma} = \frac{1}{2}\tilde{g}^{\alpha\mu}(\nabla_\beta h_{\gamma\mu} + \nabla_\gamma h_{\beta\mu} - \nabla_\mu h_{\beta\gamma}) \quad \square$$

The proof for **Eq. (2.9)**:

By definition:

$$\tilde{R}^\alpha_{\beta\gamma\delta}\omega^\beta = (\tilde{\nabla}_\gamma\tilde{\nabla}_\delta - \tilde{\nabla}_\delta\tilde{\nabla}_\gamma)\omega^\alpha$$

and

$$\begin{aligned}
\tilde{\nabla}_\gamma \tilde{\nabla}_\delta \omega^\alpha &= \nabla_\gamma (\nabla_\delta \omega^\alpha + C^\alpha_{\delta\mu} \omega^\mu) \\
&\quad - C^\mu_{\gamma\delta} (\nabla_\mu \omega^\alpha + C^\alpha_{\delta\mu} \omega^\mu) \\
&\quad + C^\alpha_{\delta\mu} (\nabla_\delta \omega^\mu + C^\mu_{\delta\nu} \omega^\nu)
\end{aligned}$$

Thus,

$$\begin{aligned}
\tilde{\nabla}_\gamma \tilde{\nabla}_\delta \omega^\alpha - \tilde{\nabla}_\delta \tilde{\nabla}_\gamma \omega^\alpha &= 2\tilde{\nabla}_{[\gamma} \tilde{\nabla}_{\delta]} \omega^\alpha \\
&= R^\alpha_{\beta\gamma\delta} \omega^\beta + 2\nabla_{[\gamma} C^\alpha_{\delta]\mu} \omega^\mu + 2C^\alpha_{\mu[\delta} \nabla_{\gamma]} \omega^\mu + 2C^\alpha_{\mu[\gamma} \nabla_{\delta]} \omega^\mu + 2C^\alpha_{\mu[\gamma} C^\alpha_{\delta]\nu} \omega^\nu \\
&= R^\alpha_{\beta\gamma\delta} \omega^\beta + 2\nabla_{[\gamma} C^\alpha_{\delta]\mu} \omega^\mu + 2C^\alpha_{\mu[\gamma} C^\alpha_{\delta]\nu} \omega^\nu \\
\tilde{R}^\alpha_{\beta\gamma\delta} &= R^\alpha_{\beta\gamma\delta} + 2\nabla_{[\gamma} C^\alpha_{\delta]\beta} + 2C^\alpha_{\mu[\gamma} C^\alpha_{\delta]\beta} \quad \square
\end{aligned}$$

C. Python Code

C.1 Script Used to Generate Data

```
1 from ast import arg
2 import subprocess
3 from sys import stdout
4
5 lst = [0.25, 0.25248870865677786, 0.25512629051784125, 0.2579389410194883,
6        0.26096225322869604,
7        0.2642469474445845, 0.26787014760635847, 0.2719606950529427,
8        0.276768703936038, 0.28294198220896544]
9
10 for arg1, v in zip([x * 0.1 for x in range(0, 10)], lst):
11     number = "{:.1f}".format(arg1)
12     with open(f'{number} correct.csv', 'w') as data_file:
13         subprocess.run(['./bin/Circ_Eq_Seq2', f'{v}',
14                        f'{number}', '1', '0.001', '4', '10', '1', '10', f'{
15 number}'+ '_out_'+ 'corrected'], stdout=data_file)
```

Listing C.1: Script used to generate the toy dataset

```
1 from ast import arg
2 import subprocess
3 from sys import stdout
4 import os
5 import numpy as np
6
7 p_isco = [6.0, 5.835703087865181, 5.669302571208653, 5.500618391277777,
8          5.329443296434935, 5.1555367298123205, 4.97861683057595,
9          4.798349785163281, 4.6143353705640235, 4.426086882222835,
10         4.233002529530826, 4.034323403998572, 3.829069418813151,
11         3.615937243849511, 3.3931284701816296, 3.158039166325728,
12         2.9066438544641957, 2.632100685754583, 2.320883041761887,
13         1.9372378781396624]
14
15 x = np.arange(0, 1, 0.05)
16 y = np.arange(0, 10.5, 0.5)
```

```

12
13 for arg1, p in zip(x, p_isco):
14     for u in y:
15         u1 = float("{:.1f}".format(u))
16         n = np.exp(u1*0.23979) - 1
17         for l in range(2, 15 + 1):
18             for m in range(1, l+1):
19                 number = "{:.2f}".format(arg1)
20
21                 with open('out', 'w') as data_file:
22                     subprocess.run(['./bin/Circ_Eq', f'{p+n}',
23                                     f'{number}', '1', f'{l}', f'{m}', 'a_'+f'
{number}'+ '_u_'+f'{u1}'+ '_out_'+f'l'+f'{l}'+f'm'+f'{m}'], stdout=data_file)

```

Listing C.2: Script used to generate the 20 by 21 dataset

```

1 from ast import arg
2 import subprocess
3 from sys import stdout
4 import os
5 import numpy as np
6
7 p_isco = [4.501929891613884, 4.407007183524613, 4.310861009479744,
8           4.2134005592566, 4.114523239768497, 4.0141124218603315,
9           3.9120346018896015, 3.8081357811300873, 3.702236782641949,
10          3.5941270982990616,
11          3.4835566604432793, 3.3702246141456507, 3.253763637329264,
12          3.1337174445965297, 3.0095074690253765, 2.8803816019812727,
13          2.7453315824194204, 2.6029519155547147, 2.4511802574368193,
14          2.2867692899162364]
15
16 x = np.arange(0.43, 0.92, 0.025)
17 y = np.arange(0.2, 10, 0.75)
18 for arg1, p in zip(x, p_isco):
19     for u in y:
20         u1 = float("{:.2f}".format(u))
21         n = np.exp(u1*0.23979) - 1
22         for l in range(2, 15 + 1):
23             for m in range(1, l+1):
24                 number = "{:.2f}".format(arg1)

```

```

20         with open('out', 'w') as data_file:
21             subprocess.run(['./bin/Circ_Eq', f'{p+n}',
22                             f'{number}', '1', f'{l}', f'{m}', 'a_'+f'
{number}'+f'_u_'+f'{u1}' + '_out_'+f'l'+f'{l}'+f'm'+f'{m}'], stdout=
data_file)

```

Listing C.3: Script used to generate the 20 by 14 dataset

```

1 def polyfit2d(x, y, z, kx=3, ky=3, order=None):
2     '''
3     Two dimensional polynomial fitting by least squares.
4     Fits the functional form f(x,y) = z.
5
6     Notes
7     ----
8     Resultant fit can be plotted with:
9     np.polynomial.polynomial.polygrid2d(x, y, soln.reshape((kx+1, ky+1)))
10
11     Parameters
12     -----
13     x, y: array-like, 1d
14         x and y coordinates.
15     z: np.ndarray, 2d
16         Surface to fit.
17     kx, ky: int, default is 3
18         Polynomial order in x and y, respectively.
19     order: int or None, default is None
20         If None, all coefficients up to maximum kx, ky, ie. up to and
21         including xkxyky, are considered.
22         If int, coefficients up to a maximum of kx+ky <= order are considered
23         .
24
25     Returns
26     -----
27     Return parameters from np.linalg.lstsq.
28
29     soln: np.ndarray
30         Array of polynomial coefficients.
31     residuals: np.ndarray
32     rank: int
33     s: np.ndarray

```

```

32
33     '''
34
35     # grid coords
36     x, y = np.meshgrid(x, y)
37     # coefficient array, up to x^kx, y^ky
38     coeffs = np.ones((kx+1, ky+1))
39
40     # solve array
41     a = np.zeros((coeffs.size, x.size))
42
43     # for each coefficient produce array x^i, y^j
44     for index, (i, j) in enumerate(np.ndindex(coeffs.shape)):
45         # do not include powers greater than order
46         if order is not None and i + j > order:
47             arr = np.zeros_like(x)
48         else:
49             arr = coeffs[i, j] * x**i * y**j
50         a[index] = arr.ravel()
51
52     # do leastsq fitting and return leastsq result
53     return np.linalg.lstsq(a.T, np.ravel(z), rcond=None)

```

Listing C.4: A coded up function that enables us to do a polynomial 2D fit. The code is heavily inspired from [45]

C.2 Neural Network

```

1 #Creating the model
2 inputs = keras.Input(shape = (2,))
3 dense = keras.layers.Dense(4, activation = "relu")
4 x = dense(inputs)
5 x = keras.layers.Dense(8, activation = "relu")(x)
6 x = keras.layers.Dense(16, activation = "relu")(x)
7 x = keras.layers.Dense(32, activation = "relu")(x)
8 x = keras.layers.Dense(64, activation = "relu")(x)
9 x = keras.layers.Dense(128, activation = "relu")(x)
10 x = keras.layers.Dense(256, activation = "relu")(x)
11 x = keras.layers.Dense(256, activation = "relu")(x)
12 x = keras.layers.Dense(256, activation = "relu")(x)
13 x = keras.layers.Dense(256, activation = "relu")(x)

```

```

14 x = keras.layers.Dense(256, activation = "relu")(x)
15 x = keras.layers.Dense(256, activation = "relu")(x)
16 x = keras.layers.Dense(256, activation = "relu")(x)
17 x = keras.layers.Dense(256, activation = "relu")(x)
18 x = keras.layers.Dense(256, activation = "relu")(x)
19 x = keras.layers.Dense(256, activation = "relu")(x)
20 x = keras.layers.Dense(256, activation = "relu")(x)
21 x = keras.layers.Dense(256, activation = "relu")(x)
22 x = keras.layers.Dense(256, activation = "relu")(x)
23 x = keras.layers.Dense(256, activation = "relu")(x)
24 output_1 = keras.layers.Dense(119, activation = "linear", name = "real")(x)
25 output_2 = keras.layers.Dense(119, activation = "linear", name = "imaginary")
    (x)
26
27
28 model = keras.Model(inputs=inputs, outputs = [output_1, output_2], name="test"
    )

```

Listing C.5: Neural Network Model

C.3 Mode-Distribution Error Code

```

1 def mode_dis_acc(fit, test, no_para):
2     #fit = array of fitted case, test = array of test case (in shape of [
3     no_para, modes])
4     lst = []
5     for i in range(no_para):
6         acc = fit[i]@test[i]/ np.linalg.norm(test[i]) / np.linalg.norm(fit[i]
7         ])
8         lst.append(acc)
9     return 1- np.array(lst)

```

Listing C.6: Mode-distribution error definition

C.4 Mismatch Code

```

1 def waveform_acc(fit_real, fit_img, test_real, test_img):
2     lst = []
3     hh = np.inner(fit_real, fit_real) + np.inner(fit_img, fit_img)
4     h0h0 = np.inner(test_real, test_real) + np.inner(test_img, test_img)
5     hh0 = np.inner(fit_real, test_real) + np.inner(fit_img, test_img)
6     acc = hh0 / np.sqrt(hh * h0h0)

```

```
7     lst.append(acc)
8     return 1 - np.array(lst)
```

Listing C.7: Mismatch definition

Bibliography

- [1] B. P. Abbott et al. Observation of Gravitational Waves from a Binary Black Hole Merger. *Physical Review Letters*, 116(6):061102, February 2016. ISSN 0031-9007, 1079-7114. doi: 10.1103/PhysRevLett.116.061102. URL <https://link.aps.org/doi/10.1103/PhysRevLett.116.061102>.
- [2] T. L. S. Collaboration et al. The basic physics of the binary black hole merger GW150914. *Annalen der Physik*, 529(1-2):1600209, January 2017. ISSN 0003-3804, 1521-3889. doi: 10.1002/andp.201600209. URL <http://arxiv.org/abs/1608.01940>.
- [3] L. Barack and A. Pound. Self-force and radiation reaction in general relativity. *Reports on Progress in Physics*, 82(1):016904, January 2019. ISSN 0034-4885, 1361-6633. doi: 10.1088/1361-6633/aae552. URL <http://arxiv.org/abs/1805.10385>.
- [4] LISA Factsheet. URL https://www.esa.int/Science_Exploration/Space_Science/LISA_factsheet.
- [5] S. A. Hughes et al. Adiabatic waveforms for extreme mass-ratio inspirals via multivoice decomposition in time and frequency. *Physical Review D*, 103(10):104014, May 2021. ISSN 2470-0010, 2470-0029. doi: 10.1103/PhysRevD.103.104014. URL <http://arxiv.org/abs/2102.02713>.
- [6] S. Babak, A. Taracchini, and A. Buonanno. Validating the effective-one-body model of spinning, precessing binary black holes against numerical relativity. *Physical Review D*, 95(2):024010, January 2017. ISSN 2470-0010, 2470-0029. doi: 10.1103/PhysRevD.95.024010. URL <https://link.aps.org/doi/10.1103/PhysRevD.95.024010>.
- [7] L. Blanchet et al. Gravitational-Radiation Damping of Compact Binary Systems to Second Post-Newtonian Order. *Physical Review Letters*, 74(18):3515–3518, May 1995. ISSN 0031-9007, 1079-7114. doi: 10.1103/PhysRevLett.74.3515. URL <https://link.aps.org/doi/10.1103/PhysRevLett.74.3515>.
- [8] C. O. Lousto and J. Healy. Exploring the Small Mass Ratio Binary Black Hole Merger via Zeno’s Dichotomy Approach. *Physical Review Letters*, 125(19):191102,

- November 2020. ISSN 0031-9007, 1079-7114. doi: 10.1103/PhysRevLett.125.191102. URL <https://link.aps.org/doi/10.1103/PhysRevLett.125.191102>.
- [9] T. Hinderer and É. É. Flanagan. Two-timescale analysis of extreme mass ratio inspirals in Kerr spacetime: Orbital motion. *Physical Review D*, 78(6):064028, September 2008. ISSN 1550-7998, 1550-2368. doi: 10.1103/PhysRevD.78.064028. URL <https://link.aps.org/doi/10.1103/PhysRevD.78.064028>.
- [10] S. A. Teukolsky. Perturbations of a Rotating Black Hole. I. Fundamental Equations for Gravitational, Electromagnetic, and Neutrino-Field Perturbations. *The Astrophysical Journal*, 185:635, October 1973. ISSN 0004-637X, 1538-4357. doi: 10.1086/152444. URL <http://adsabs.harvard.edu/doi/10.1086/152444>.
- [11] S. Isoyama et al. Adiabatic waveforms from extreme-mass-ratio inspirals: An analytical approach. *Physical Review Letters*, 128(23):231101, June 2022. ISSN 0031-9007, 1079-7114. doi: 10.1103/PhysRevLett.128.231101. URL <http://arxiv.org/abs/2111.05288>.
- [12] A. Pound and B. Wardell. Black hole perturbation theory and gravitational self-force. pages 1–119. 2021. doi: 10.1007/978-981-15-4702-7_38-1. URL <http://arxiv.org/abs/2101.04592>.
- [13] C. Cutler, D. Kennefick, and E. Poisson. Gravitational radiation reaction for bound motion around a Schwarzschild black hole. *Physical Review D*, 50(6):3816–3835, September 1994. ISSN 0556-2821. doi: 10.1103/PhysRevD.50.3816. URL <https://link.aps.org/doi/10.1103/PhysRevD.50.3816>.
- [14] M. Shibata. Gravitational waves by compact star orbiting around rotating supermassive black holes. *Physical Review D*, 50(10):6297–6311, November 1994. ISSN 0556-2821. doi: 10.1103/PhysRevD.50.6297. URL <https://link.aps.org/doi/10.1103/PhysRevD.50.6297>.
- [15] K. Glampedakis and D. Kennefick. Zoom and whirl: Eccentric equatorial orbits around spinning black holes and their evolution under gravitational radiation reaction. *Physical Review D*, 66(4):044002, August 2002. ISSN 0556-2821, 1089-4918. doi: 10.1103/PhysRevD.66.044002. URL <https://link.aps.org/doi/10.1103/PhysRevD.66.044002>.

- [16] S. A. Hughes. Evolution of circular, nonequatorial orbits of Kerr black holes due to gravitational-wave emission. *Physical Review D*, 61(8):084004, March 2000. ISSN 0556-2821, 1089-4918. doi: 10.1103/PhysRevD.61.084004. URL <https://link.aps.org/doi/10.1103/PhysRevD.61.084004>.
- [17] S. Drasco, E. E. Flanagan, and S. A. Hughes. Computing inspirals in Kerr in the adiabatic regime. I. The scalar case. *Classical and Quantum Gravity*, 22(15):S801–S846, August 2005. ISSN 0264-9381, 1361-6382. doi: 10.1088/0264-9381/22/15/011. URL <http://arxiv.org/abs/gr-qc/0505075>.
- [18] S. A. Hughes, S. Drasco, E. E. Flanagan, and J. Franklin. Gravitational radiation reaction and inspiral waveforms in the adiabatic limit. *Physical Review Letters*, 94(22):221101, June 2005. ISSN 0031-9007, 1079-7114. doi: 10.1103/PhysRevLett.94.221101. URL <http://arxiv.org/abs/gr-qc/0504015>.
- [19] Y. Mino, M. Sasaki, and T. Tanaka. Gravitational radiation reaction to a particle motion. *Physical Review D*, 55(6):3457–3476, March 1997. ISSN 0556-2821, 1089-4918. doi: 10.1103/PhysRevD.55.3457. URL <https://link.aps.org/doi/10.1103/PhysRevD.55.3457>.
- [20] T. C. Quinn and R. M. Wald. Axiomatic approach to electromagnetic and gravitational radiation reaction of particles in curved spacetime. *Physical Review D*, 56(6):3381–3394, September 1997. ISSN 0556-2821, 1089-4918. doi: 10.1103/PhysRevD.56.3381. URL <https://link.aps.org/doi/10.1103/PhysRevD.56.3381>.
- [21] E. Poisson, A. Pound, and I. Vega. The Motion of Point Particles in Curved Spacetime. *Living Reviews in Relativity*, 14(1):7, December 2011. ISSN 2367-3613, 1433-8351. doi: 10.12942/lrr-2011-7. URL <http://link.springer.com/10.12942/lrr-2011-7>.
- [22] R. H. Boyer and R. W. Lindquist. Maximal Analytic Extension of the Kerr Metric. *Journal of Mathematical Physics*, 8(2):265–281, February 1967. ISSN 0022-2488, 1089-7658. doi: 10.1063/1.1705193. URL <http://aip.scitation.org/doi/10.1063/1.1705193>.
- [23] S. A. Hughes. The evolution of circular, non-equatorial orbits of Kerr black holes due to gravitational-wave emission. *Physical Review D*, 78(10):109902, November

2008. ISSN 1550-7998, 1550-2368. doi: 10.1103/PhysRevD.78.109902. URL <http://arxiv.org/abs/gr-qc/9910091>.
- [24] W. Schmidt. Celestial mechanics in Kerr spacetime. *Classical and Quantum Gravity*, 19(10):2743–2764, May 2002. ISSN 02649381. doi: 10.1088/0264-9381/19/10/314. URL <http://arxiv.org/abs/gr-qc/0202090>.
- [25] S. A. Hughes. Evolution of circular, non-equatorial orbits of Kerr black holes due to gravitational-wave emission: II. Inspiral trajectories and gravitational waveforms. *Physical Review D*, 64(6):064004, August 2001. ISSN 0556-2821, 1089-4918. doi: 10.1103/PhysRevD.64.064004. URL <http://arxiv.org/abs/gr-qc/0104041>.
- [26] W. H. Press and S. A. Teukolsky. Perturbations of a Rotating Black Hole. II. Dynamical Stability of the Kerr Metric. *The Astrophysical Journal*, 185:649, October 1973. ISSN 0004-637X, 1538-4357. doi: 10.1086/152445. URL <http://adsabs.harvard.edu/doi/10.1086/152445>.
- [27] E. Newman and R. Penrose. An Approach to Gravitational Radiation by a Method of Spin Coefficients. *Journal of Mathematical Physics*, 3(3):566–578, May 1962. ISSN 0022-2488, 1089-7658. doi: 10.1063/1.1724257. URL <http://aip.scitation.org/doi/10.1063/1.1724257>.
- [28] S. Drasco and S. A. Hughes. Gravitational wave snapshots of generic extreme mass ratio inspirals. *Physical Review D*, 73(2):024027, January 2006. ISSN 1550-7998, 1550-2368. doi: 10.1103/PhysRevD.73.024027. URL <http://arxiv.org/abs/gr-qc/0509101>.
- [29] G. Khanna. Teukolsky evolution of particle orbits around Kerr black holes in the time domain: Elliptic and inclined orbits. *Physical Review D*, 69(2):024016, January 2004. ISSN 1550-7998, 1550-2368. doi: 10.1103/PhysRevD.69.024016. URL <https://link.aps.org/doi/10.1103/PhysRevD.69.024016>.
- [30] R. Fujita and H. Tagoshi. New Numerical Methods to Evaluate Homogeneous Solutions of the Teukolsky Equation. *Progress of Theoretical Physics*, 112(3):415–450, September 2004. ISSN 0033-068X, 1347-4081. doi: 10.1143/PTP.112.415. URL <http://arxiv.org/abs/gr-qc/0410018>.
- [31] S. A. Teukolsky and W. H. Press. Perturbations of a rotating black hole. III - Interaction of the hole with gravitational and electromagnetic radiation. *The*

- Astrophysical Journal*, 193:443, October 1974. ISSN 0004-637X, 1538-4357. doi: 10.1086/153180. URL <http://adsabs.harvard.edu/doi/10.1086/153180>.
- [32] C. Kavanagh, A. C. Ottewill, and B. Wardell. Analytical high-order post-Newtonian expansions for spinning extreme mass ratio binaries. *Physical Review D*, 93(12):124038, June 2016. ISSN 2470-0010, 2470-0029. doi: 10.1103/PhysRevD.93.124038. URL <http://arxiv.org/abs/1601.03394>.
- [33] N. Sago and R. Fujita. Calculation of radiation reaction effect on orbital parameters in Kerr spacetime. *Progress of Theoretical and Experimental Physics*, 2015(7):073E03, July 2015. ISSN 2050-3911. doi: 10.1093/ptep/ptv092. URL <http://arxiv.org/abs/1505.01600>.
- [34] Black hole perturbation toolkit. URL <http://bhptoolkit.org/>.
- [35] Gremlin - Teukolsky equation solver for a point particle on a bound, timelike orbit. Black Hole Perturbation Toolkit, November 2021. URL <https://github.com/BlackHolePerturbationToolkit/GremlinEq>.
- [36] A. J. K. Chua, M. L. Katz, N. Warburton, and S. A. Hughes. Rapid generation of fully relativistic extreme-mass-ratio-inspiral waveform templates for LISA data analysis. *Physical Review Letters*, 126(5):051102, February 2021. ISSN 0031-9007, 1079-7114. doi: 10.1103/PhysRevLett.126.051102. URL <http://arxiv.org/abs/2008.06071>.
- [37] F. Pedregosa et al. Scikit-learn: Machine Learning in Python, June 2018. URL <http://arxiv.org/abs/1201.0490>.
- [38] P. Virtanen et al. SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nature Methods*, 17(3):261–272, March 2020. ISSN 1548-7091, 1548-7105. doi: 10.1038/s41592-019-0686-2. URL <http://www.nature.com/articles/s41592-019-0686-2>.
- [39] M. L. Katz et al. FastEMRIWaveforms: New tools for millihertz gravitational-wave data analysis. *Physical Review D*, 104(6):064047, September 2021. ISSN 2470-0010, 2470-0029. doi: 10.1103/PhysRevD.104.064047. URL <http://arxiv.org/abs/2104.04582>.

- [40] B. Kocsis, N. Yunes, and A. Loeb. Observable Signatures of EMRI Black Hole Binaries Embedded in Thin Accretion Disks. *Physical Review D*, 84(2):024032, July 2011. ISSN 1550-7998, 1550-2368. doi: 10.1103/PhysRevD.84.024032. URL <http://arxiv.org/abs/1104.2322>.
- [41] H. Yang and M. Casals. General Relativistic Dynamics of an Extreme Mass-Ratio Binary interacting with an External Body. *Physical Review D*, 96(8):083015, October 2017. ISSN 2470-0010, 2470-0029. doi: 10.1103/PhysRevD.96.083015. URL <http://arxiv.org/abs/1704.02022>.
- [42] B. Bonga, H. Yang, and S. A. Hughes. Tidal resonance in extreme mass-ratio inspirals. *Physical Review Letters*, 123(10):101103, September 2019. ISSN 0031-9007, 1079-7114. doi: 10.1103/PhysRevLett.123.101103. URL <http://arxiv.org/abs/1905.00030>.
- [43] J. Miller and A. Pound. Two-timescale evolution of extreme-mass-ratio inspirals: Waveform generation scheme for quasicircular orbits in Schwarzschild spacetime. *Physical Review D*, 103(6):064048, March 2021. ISSN 2470-0010, 2470-0029. doi: 10.1103/PhysRevD.103.064048. URL <http://arxiv.org/abs/2006.11263>.
- [44] Y. Mino. Perturbative approach to an orbital evolution around a supermassive black hole. *Physical Review D*, 67(8):084027, April 2003. ISSN 0556-2821, 1089-4918. doi: 10.1103/PhysRevD.67.084027. URL <https://link.aps.org/doi/10.1103/PhysRevD.67.084027>.
- [45] P. Harrison. Answer to "Equivalent of 'polyfit' for a 2D polynomial in Python", September 2019. URL <https://stackoverflow.com/a/57923405>.