WHITE DWARF BINARY ORBIT EVOLUTION USING STELLAR RESONANCE AND GRAVITATIONAL WAVE EMISSION

By

Thomas S. Cane

Thesis Project
Submitted in Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE IN PHYSICS AND ASTRONOMY

August 17 2020

Dr. Shane L. Larson , First Reader

Dr. Giles Novak , Second Reader

ABSTRACT

This thesis presents research done to better develop the understanding of compact binary objects orbital evolution. Relevant background information is introduced to familiarize the reader with the requisite background knowledge. A companion to gravitation wave emission for binary orbital evolution in the form of resonant modes is discussed at length with the requirements and effect being fully parameterized by several equations developed by McKernan and Ford 2016. The orbital evolution and orbital frequency histograms for two compact binary systems that undergo gravitational wave emission are presented with an accompanying discussion on the affect that Dual Evolution will have on orbital evolution. The code used to filter and evolve sources is given in the appendices.

ACKNOWLEDGEMENTS

While researching and writing this work I received a great deal of support.

Foremost I would like to thank Professor Shane Larson, my research advisor at Northwestern, for giving me the opportunity to conduct research under him, helping me adapt to graduate school life, and making sure that throughout these tumultuous times I was staying healthy and taking care of myself.

Thank you to Professor Giles Novak for stimulating my curiosity of interstellar processes even further and for being a member of my thesis committee.

Thank you Professor Mary Odekon, my research advisor at Skidmore College, for allowing me to join her research group as a freshman and showing me how fantastic astronomy research is.

To anyone who pursues their curiosities about the universe, $\chi\alpha\iota\rho\epsilon$

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1
INTRODUCTION AND BACKGROUND

1.1 Gravitational Waves

First predicted by Einstein in his theory of general relativity and quantified by the Laser Interferometer Gravitational-Wave Observatory (LIGO) in 2015, Gravitational Waves are a phenomena that allow astronomers to observe previously invisible events throughout the universe. In 1974 the Arecibo Radio Telescope observed a pulsar binary system for the first time. After nearly eight years of collecting data the astronomers there were able to say that the orbit of the system was indeed shrinking as predicted by general relativity, but it was not until LIGO that the existence of gravitational waves would be confirmed by direct detection. Up until this discovery the usual method of analyzing far away objects was using various forms of electromagnetic waves such as galaxy luminosity, x-ray observation of stellar mass black holes, and other forms of EM wavelength observation. After the discovery of gravitational waves astronomers were able to begin looking at events that are either too dim to be seen with, or completely invisible in electromagnetic waves. The most numerous strong source of gravitational waves are compact binary systems, where the binary components are stellar remnants: black holes, neutron stars, and white dwarfs.

Gravitational waves are generated from the orbital motion of any binary system, whether it be two stellar mass black holes, white dwarfs, or any other compact object. As the binary system evolves the orbit will shrink which points to a loss of energy from the system, and from conservation of energy we are able to say that the energy lost from the system has been deposited into the emitted gravitational wave. These waves are transverse waves, will oscillate perpendicular to the direction of motion and travel at the speed of light, but will be unimpeded by the usual sources of disruption that distort EM waves. Currently these waves are measured by ground based observatories, such as LIGO, but there are plans to have space based observatories launched in the early

2030's.

1.2 LISA

Ground based observatories have problems with background noise and operate at high frequencies meaning that they can only observe binary systems at the tail end of their lifetime, but space based observatories like the Laser Interferometer Space Antenna (LISA) will be able to have simpler noise and observes at a much lower frequency range, so they will be able to detect a multitude of binary systems and their corresponding gravitational waves. Terrestrial observatories operate from approximately 10 Hz to 1000 Hz and are only able to observe the latter portion of the orbital lifetime of compact binaries along with rotating neutron stars. Space interferometers observe sources with much lower frequencies ranging from 0.1 mHz to 1 Hz which covers the final orbital evolution of binary Supermassive Blackholes (SMBHs), the early and middle evolutionary lifetime of compact binaries, and the majority of the evolution for compact objects orbiting SMBHs.

Currently planned for launch in the early 2030's, LISA is one such gravitational wave observatory. An ESA funded program with NASA partnership this mission is designed to measure the aforementioned binary phenomena with extreme precision using a unique design of three separate spacecraft. After LISA is fully deployed these spacecraft will form an equilateral triangle in space with a distance, arm length, between points being 2.5 million km long. These detectors will measure gravitational waves as though they were measuring ripples on a still lake. Imagine three leaves floating on the surface of the water and a stone is dropped into the water creating ripples causing the leaves to move relative to one another. This is the process that the LISA detectors will undergo. The binary system will create gravitational waves that will ripple across spacetime. Once the wave reaches one of the detectors it will perturb the detector out of alignment from the other two detectors. This will be recorded and the source will be identified by the characteristic pattern from the the disruption of the beam communicating between detectors.

In the four years that LISA operates the most prolific binary system predicted to be observed are White Dwarf binaries. The orbital evolution of these systems will be directly considered in this work.

1.3 White Dwarf Binaries

When sun-like stars reach the end of their life by depleting their nuclear fuel the star will eject matter surrounding its core creating a planetary nebula leaving behind a very hot and dense core. This new object is known as a White Dwarf (WD), having a surface temperature of nearly 100,000K and generally comprised of ultra-dense carbon and oxygen. When fusion is no longer able to occur the WDs internal structure is only supported by degenerate electron pressure, so the self-gravity will collapse the star further leaving behind an extremely dense core. The typical mass and radius of WD's range from 0.5-0.7$M_{\odot}$ with the radius ranging just 0.8-2$\%R_{\odot}$ yielding an average density of $\sim 10^{9}kg/m^{3}$. Some WDs will be massive enough to fuse carbon into neon, but for this research we only consider WDs composed of carbon and oxygen. As the WD cools theorists have predicted it will transition to a black dwarf, a crystallized core with very low surface temperature and little to no light emission, but because of the temperature of WDs it is generally accepted that the time to cool into a black dwarf is longer than the current age of the universe.

Several evolutionary and structural changes occur to WDs when they are in a binary system. When two WDs are orbiting each other mass-transfer between them will occur. The more massive object will exert a stronger tide, gravitational pull, on the secondary object causing the primary to gain both mass and density. However, if the WD accretes enough matter to surpass the Chandrasekhar mass of 1.4$M_{\odot}$ the extreme self gravity could cause carbon ignition leading to a thermalization of the core resulting in a type 1a supernova event that would destroy the system. In very rare cases the two WDs could accrete onto each other on approximately equal time scales so no single WD's gravity dominates the system causing the binary to merge together making the mass greater than the Chandrasekhar mass leading to the aforementioned type 1a supernova event. The orbit of a binary is generally described by the usual equations by Peters and Matthews (1963) which stipulate that the energy lost from the orbit is accounted for by the emission of gravitational waves. This thesis will amend the assumption of gravitational wave exclusive evolution and test a proposed model (McKernan et al. 2016) that says the orbital evolution of WD-WD binary is also dependent on the energy deposited in various resonant modes of the WDs.

1.4 Resonant Modes

When a WD binary orbit shrinks tidal forcing will occur periodically between the WDs causing resonance between this tidal force and the quadrupolar moment of the WD. If a tide is raised it follows that there will be an associated tidal potential being driven by a tidal torque that can put the WD into resonance with its fundamental frequency or harmonic.

When any oscillatory system is driven by resonance there must be some restoring force that tries to bring the system back to equilibrium. There are many different methods by which this can occur, but this thesis will only focus on the restoring forces of pressure and gravity in the context of WD binary systems. P modes, or pressure modes, are resonant modes defined when the WD is brought out of excitation by compressing waves analogous to standing sound waves. The second mode called a g mode, or gravity mode, is when buoyancy restores the WD to equilibrium via radial gravitational force analogous to convective waves. Each of these modes are also described by low order harmonics derived from the fundamental solid body vibrational frequency, generally dubbed the fundamental frequency of the star, given that the harmonics might also experience resonance via tidal forcing.

This thesis will focus heavily on energy deposition in resonant p and g modes along with gravitational wave emission as methods of driving binary orbital evolution. Chapter 2 focuses on these resonant modes, the energy deposited in them, and a full mathematical explanation about how astrophysical processes are incorporated into gravitational wave evolution. Chapter 3 discusses the results found once a synthesized population has undergone a dual astrophysical and gravitational wave evolution. The two cases considered are that of a single system and how the new process affects the orbital frequency of the source, and a statistical treatment of a large population's evolution to better understand the confusion background sources predicted to be seen by LISA. Chapter 4 will discuss the relevance of the modal energy and how significant of a contribution it makes to our overall understanding of binary system evolution. Chapter 5 will conclude the thesis with a discussion on the potential refinements to the astrophysical model and code used to implement it, and what future work will be done to further expand on the discoveries relayed in this work.

CHAPTER 2

METHODS

2.1 Including Astrophysical Evolution

Generally binary system's orbits are evolved using gravitational waves exclusively using Peter's et al. 1964. The orbital energy lost from gravitational wave emission is described as

$$\langle\frac{dE}{dt}\rangle = -\frac{32}{5}\frac{G^4 m_1^2 m_2^2 (m_1 + m_2)}{c^5 a^5 (1-e^2)^{\frac{7}{2}}}(1 + \frac{73}{24}e^2 + \frac{37}{96}e^4) \tag{2.1}$$

Where G is Newton's gravitational constant, m_1 and m_2 are the masses of the binary objects measured in $M_\odot$, c is the speed of light in m/s, a is the semi-major axis in $R_\odot$, and e is the eccentricity of the orbit. There are similar equations that describe the changes in semi-major axis and eccentricity, but for this research we consider the eccentricity will remain constant and the semi-major axis will change via the inverse energy relationship seen in equation 5.2 of Peters et al. 1964.

$$a_b = \frac{-G m_1 m_2}{2E_{tot}} \tag{2.2}$$

Where the total energy E_{tot}, is measured in joules. The orbital period P_{orb} is initially a measured quantity, but is evolved using Kepler's third law.

$$P_{orb} = \sqrt{\frac{4\pi^2 a_b^3}{G(m_1 + m_2)}} \tag{2.3}$$

Where m_c is the chirp mass of the binary measured in $M_\odot$ and defined as

$$m_c = \frac{(m_1 m_2)^{\frac{3}{5}}}{(m_1 + m_2)^{\frac{1}{5}}} \tag{2.4}$$

The incorporation of the astrophysical evolution described my McKernan et al. 2016 is outlined by Figure 2.1 below.

This process begins with the population synthesis using COSMIC discussed heavily in Section 3 of this chapter. The created sources undergo multiple filtration steps to leave only the relevant ones needed for astrophysical and gravitational wave evolution. First, any stellar mass black hole, pulsar, neutron star, or other stellar remnant generated by COSMIC is discarded leaving white dwarfs remaining. Next, the chemical composition of the WD's is examined, keeping only those composed of carbon and oxygen (CO) electrons and removing helium or oxygen-neon based WD's. The last part of the population filtration is making sure the binary is actively orbiting, so any sources with either zero mass or zero orbital period, meaning the binary has collided or merged, is discarded from the data set.

The beginning of astrophysical evolution in tandem with gravitational wave evolution, or dual evolution, begins once a WD is depositing energy in resonant modes. The resonance requirement is given by Ford and McKernan 2016 as

$$\frac{\omega_*}{k} - \frac{1}{t_d} < n\Omega_{orb} - m\omega_{spin} < \frac{\omega_*}{k} + \frac{1}{t_d} \tag{2.5}$$

where $\Omega_{orb} = 2\pi/P_{orb}$, the Keplerian orbital angular velocity, and ω_* is the break up frequency of the WD given by

$$\omega_* = \sqrt{\frac{GM_*}{R_*^3}} \tag{2.6}$$

k denotes the harmonic of the mode stipulating $0 < k < 10$, t_d is the dissipation timescale of the mode measured in seconds with an initial value of $\sim 10^{12}s$ and having subsequent values calculated using $t_d = t_{j,kT} + t_{j,GW}$, the summation of the cooling time-scales of the jth mode, p or g, via thermal emission and the cooling time via gravitational wave emission. The full description of these time-scales requires a lengthy chain of thought that will be covered, but the resonance relation needs more attention to finish its description. Looking at the coefficients of the forcing frequency expression $\omega_F = n\Omega_{orb} - m\omega_{spin}$, n and m are the usual quantum numbers with n being the principle quantum number 1,2,3...etc. and m equaling 2 to describe the quadrupolar moment

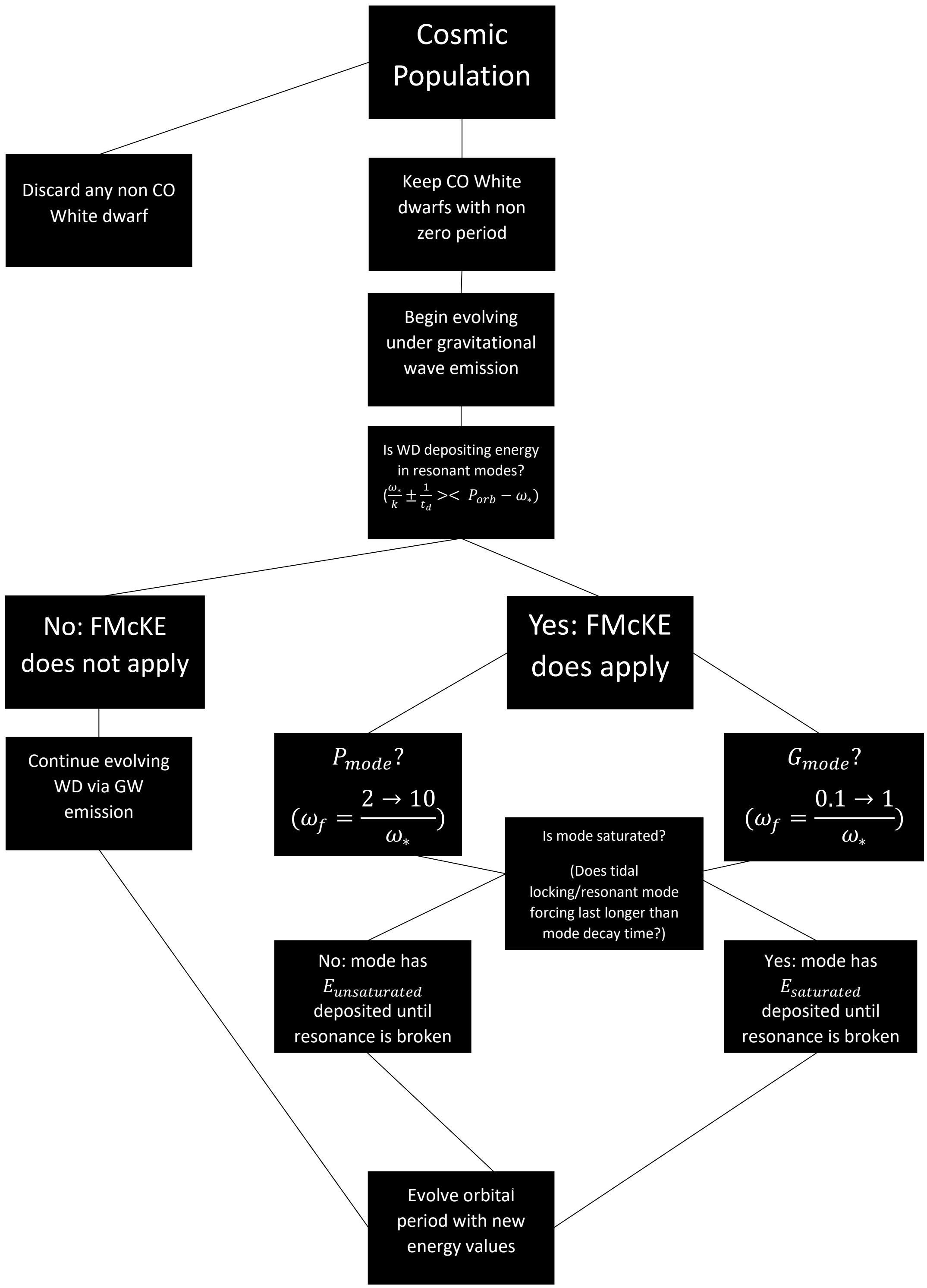

Figure 2.1: Outline of astrophysical and gravitational wave orbital evolution

of the WDs.

The mode dissipation time-scale, decay time, requires the thermal decay time-scale, $t_{j,kT}$ (Kumar and Goodman 1996), and the GW dissipation time-scale, $t_{j,GW}$. The GW time-scale does not have mode dependence and is described as

$$t_{j,GW} = (\frac{5c^5}{G})(\frac{1}{E_j\omega_j^2}) \tag{2.7}$$

Where ω_j^2 is the mode frequency where $\omega_j = 0.1 \rightarrow 1\omega_*$ for a g mode and $\omega_j = 2 \rightarrow 10\omega_*$ for a p mode. The energy of the jth mode, E_j, will be described later to account for the multiple dependencies that it has. The thermal low-order decay time-scale, $t_{j,kT}$ is quantified depending on the resonant mode the WD is in. For a g mode;

$$t_{j,kT} = \frac{4\pi}{\omega_j}(\frac{E_*}{E_j})^{\frac{1}{2}} \tag{2.8}$$

where E_* is the binding energy of the WD described by $E_* = \frac{GM_*^2}{R_*}$, and E_j is the energy deposited into the mode. The thermal decay time-scale for a low-order p mode is;

$$t_{j,kT} = 2\frac{t_p}{N_p}\frac{E_*}{E_j} \tag{2.9}$$

where t_p is the typical decay time of the mode calculated from Kumar and Goodman 1996, and N_p is the number of low-order radial modes.

The energy deposited into mode of the WD depends on two separate facets. If the WD is in a saturated state, and what mode is the WD in. Saturation of a mode occurs when the resonant mode is forced for longer than the decay time-scale of that mode. Mode forcing is able to occur when tidal locking is in effect, $\dot{\Omega}_{orb} = \dot{\omega}_{spin}$, the forcing frequency is described in Equation 2.5, but can also be described using the convenience adopted by McKernan et al. 2016 of jth mode frequencies with $\omega_F = \omega_j + \delta\omega$ where $\delta\omega = \frac{1}{r2t_d}$ with $r = 1, 2, 3...$ being a multiple of the half-width half-max (HWHM) of the resonance peak. Using this convenience the time the mode is forced for is calculated by saying $t_F = \frac{1}{\sqrt{\dot{\omega}_F}}$, so saturation is only able to occur when $t_F > t_d$. If the saturation requirement is not met we say the mode is unsaturated which does not depend on the type of resonant mode. The type of mode, either p or g, is determined by the relationship between forcing frequency and WD break up frequency. A p(ressure) mode has a frequency lower than the fundamental ω_*, so a p mode is designated as having $\omega_F = 2 \to 10\omega_*$. A g(ravity) mode has a frequency higher than the fundamental, so $\omega_F = 0.1 \to 1\omega_*$.

Knowing these two differentiation's we can fully describe the energy deposited in a resonant mode. Starting with the saturated g mode case McKernan et al 2016 derived

$$E_{sat,g} = 10^{-5} E_* (\frac{\frac{\omega_j}{\omega_*}}{0.1})^{\frac{-2}{3}} (\frac{\chi_j}{10^{-3}}) (\frac{r}{1})^{-1} (\frac{\frac{M_c}{M_*}}{1})^{\frac{4}{3}} (\frac{\frac{a_b}{R_*}}{10})^{-4} \tag{2.10}$$

where $\chi_j = 27(\frac{\omega_j}{\omega_*})^{3.7}$ is the overlap integral describing the region where resonant modes overlap as the binary orbit shrinks. The energy deposited in a saturated p is quantified by:

$$E_{sat,p} = 0.1 E_* (\frac{\omega_*}{0.3s^{-1}})^{\frac{4}{5}} (\frac{\frac{\omega_j}{\omega_*}}{4})^{\frac{2}{5}} (\frac{k}{1})^{-2} (\frac{\frac{M_c}{M_*}}{1})^{\frac{4}{5}} (\frac{\frac{a_b}{R_*}}{10})^{\frac{-12}{5}} (\frac{\chi_j}{10^{-3}})^{\frac{3}{5}} (\frac{t_p}{10^6 s})^{\frac{4}{5}} (\frac{f_{GW}}{0.5})^{\frac{-4}{5}} (\frac{N_p}{10^2})^{\frac{-4}{5}} (\frac{r}{1})^{-1} \tag{2.11}$$

where

$$f_{GW} = 2\Omega_{orb} \tag{2.12}$$

is the gravitational wave emission frequency of the binary. If the mode is unsaturated the energy described in equations 2.10 and 2.11 can be decreased by a factor of $10^{-6}(\frac{t_f}{10^3 s})^{\frac{2}{3}}(\frac{t_d}{10^{12} s})^{\frac{-2}{3}}$ for an unsaturated g mode, and for an unsaturated p mode the energy will be decreased by a factor of $0.06(\frac{t_f}{10^3 s})^{\frac{2}{5}}(\frac{t_d}{10^6 s})^{-\frac{2}{5}}$.

The total energy of the binary system is a combination of orbital potential energy, energy added to the WDs via resonant modes, and energy lost from GW emission.

$$E_{tot} = E_{orb} + E_{GW} + E_{res} \tag{2.13}$$

where

$$E_{orb} = -\frac{G(m_1 + m_2)}{2a_b} \tag{2.14}$$

E_{GW} is described by $\langle\frac{dE}{dt}\rangle * dt$ from Equation 2.1, and E_{res} described by Equation 2.10 and 2.11. Once the total energy of the system is known Equation is applied resulting the semi-major axis being calculated as

$$a_b = \frac{-Gm_1m_2}{2E_{tot}} \tag{2.15}$$

2.2 Code

The process described in section 2.1 is translated into code for the eventual application to a COSMIC galaxy described in section 2.3. The code is shown in Appendix A and was written in Python 3.3 using Jupyter Notebook. The basis for the logic forms from the outline in Figure 1. The code begins by reading in a csv file and transforming it into a pandas data frame while filtering out circular binary systems that have merged and begun mass transfer, keeping only actively orbiting eccentric CO-CO WD sources. Once the proper sources are gathered all values are converted into SI units along with initial values being supplied for the resonant energy deposited into the stars, denoted in the code as $E1$ and $E2$, initial energy lost via GW emission, zero, and initial decay time of the resonant mode, $td1$ and $td2$, as given by our collaborators at CUNY Barry McKernan Ph.D and Saavik Ford Ph.D. The basis of the dual evolution segment of the code is founded on the LISA observation time of four years, which is converted from sidereal years to seconds, while taking time steps, dt, set equal to the orbital period of the WD binary. As the system evolves through time it will go through a system of checks to make sure it still has a viable orbit and has not begun mass-transfer because of the WDs merging. The two main checks that will disqualify the source are based on the Roche-Lobe,RL, and the orbital period. The RL is the region around one of the objects in the binary system where matter is gravitationally bound to that object and is well defined by Eggelton 1983 as

$$r_{L1,2} = \frac{0.49q_{1,2}^{\frac{2}{3}}}{0.6q_{1,2} + ln(1 + q_{1,2}^{\frac{1}{3}})} \tag{2.16}$$

Where $q_{1,2} = M_{1,2}/M_{2,1}$ the mass ratio of the binary objects. The RL check is the indicator if the two WDs have begun mass transfer, because the sources kept have small eccentricity the distance between them can be approximated by the semi-major axis thus by stating when $r_{L1} + r_{L2} \leq a_b$ the binary system is transferring mass. There is one smaller check for the Ford McKernan Energy, FMcKE, that restricts the number of harmonics represented in the resonant mode set to be $k < 10$ which leads to only low order radial modes being present in the calculation.

2.3 Application to COSMIC Galaxy

Testing the relevance of dual evolution requires a galaxy sized population to run through the code and see what percentage of sources are affected by the FMcKE. The population is modeled after the Milky Way and created using the Compact Object Synthesis and Monte Carlo Investigation Code, COSMIC, an adaptation to the Fortran Binary Stellar Evolution, BSE, code created by Hurley et al 2002. The population for this thesis was created in the Google Collab environment with the exact parameters posted in Appendix B.

CHAPTER 3

RESULTS

3.1 White Dwarf-Neutron Star Systems

The White Dwarf-Neutron Star, WD-NS, system evolution via gravitational wave emission is shown in this section. First we consider a single binary system as seen in Figure 3.1 shown to quickly evolve out the LISA band corresponding to the very rapid orbital frequency achievable by WD-NS systems.

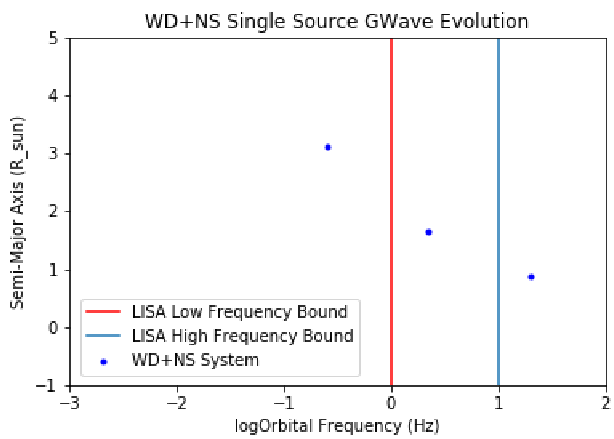

Figure 3.1: The orbital evolution of a single WD-NS binary via gravitational wave emission

Seeing the behavior of one such system it follows that a population of systems must be observed to fully understand the expected behaviour. Figure 3.2 and 3.3 do just that.

Interestingly Figure 3.4 shows that WD-NS systems will begin within the LISA band and evolve out of it whereas the WD-WD systems shown in section 3.3 will evolve into the LISA band. These evolutionary nuances will be fully discussed in Chapter 4. The frequency histogram of the WD-NS population seen in Figure 3.4 shows the vast majority of orbits have very small frequency value as anticipated.

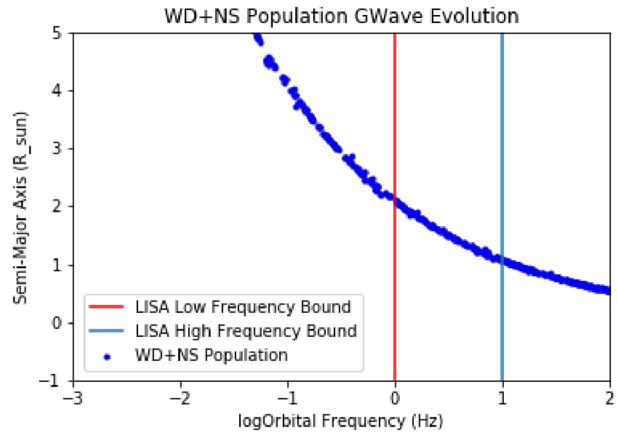

Figure 3.2: The orbital evolution for a population of WD-NS binary systems via gravitational wave emission

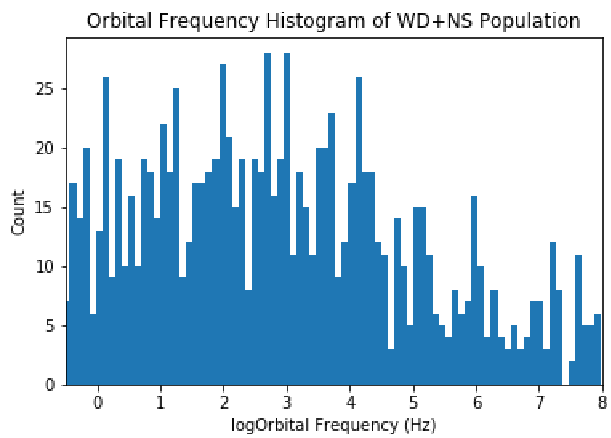

Figure 3.3: Histogram of orbital frequency for a population of WD-NS binary systems via gravitational wave emission

3.2 White Dwarf-White Dwarf Systems

The second type of systems analyzed are White Dwarf-White Dwarf, WD-WD, binaries which display a much longer evolutionary lifetime. These systems undergo the same treatment as the WD-NS systems seen in section 3.1. In a similar fashion we begin by observing the evolution of a single source via gravitational wave emission.

The population distribution for WD-WD systems follows a similar trend to the WD-NS systems, but they begin their evolution at a much lower frequency outside the LISA band and will

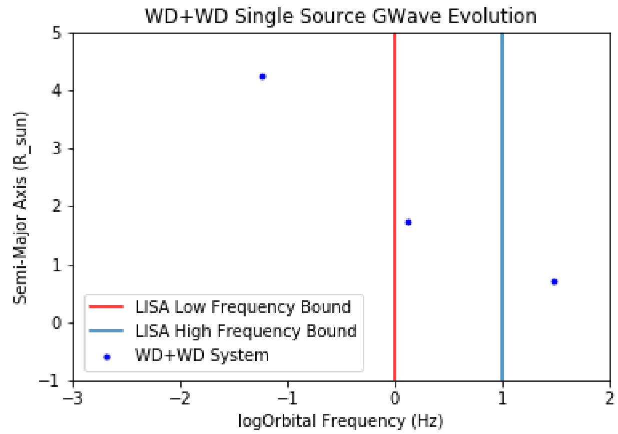

Figure 3.4: The orbital evolution for a single WD-WD binary system via gravitational wave emission

eventually evolve through the band. The histogram in Figure 3.8 requires a log scale because of the precision needed for discerning bin frequencies.

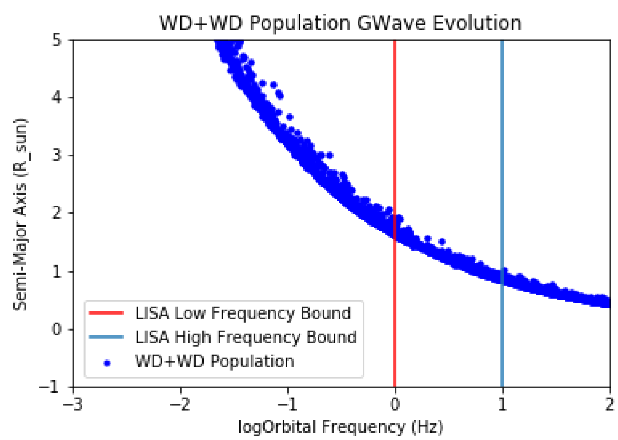

Figure 3.5: The orbital evolution for a population of WD-WD binary systems via gravitational wave emission

Translating the orbital evolution into a histogram to understand the count of frequencies results in Figure 3.8

Gravitational wave emission as a mechanism for binary orbital evolution is a very well understood topic, but the implementation of the resonant energy should have an non-negligible effect on the evolution.

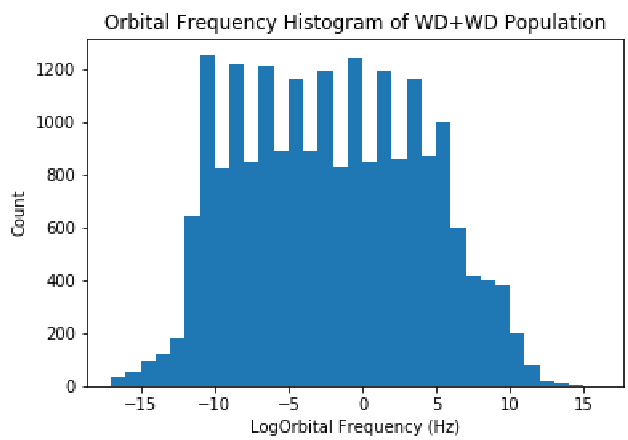

Figure 3.6: Histogram of the logged orbital frequency for a population of WD-WD systems

CHAPTER 4
DISCUSSION

4.1 Relevance of Resonant Modes for Orbital Evolution

Resonant modes require a very high level of numerical precision to properly quantify, but the impact resonance will have on orbital evolution is something that can be well understood. As stated in Chapter 2 the WD will begin depositing energy into a resonant mode once Equation 4 is satisfied with the amount of energy being regulated by the mode type and whether or not the mode is saturated. The energy evolution of the system is the starting point to fully understand how resonant modes affect a binary system or population.

From basic mechanics the orbital potential energy is what will drive the binary system to shrink over time. The orbital energy will become even more negative once gravitational waves are emitted which remove energy from the system further shrinking the binary orbit. When resonant modes are introduced into the system they will also extract energy from the orbit similar to gravitational wave emission, but the energy is deposited into the compact objects themselves instead of leaving the system entirely. Returning to basic orbital mechanics, as the orbital energy of the system grows more negative the semi-major axis will shrink causing the orbital period to also shrink that will make the orbital frequency grow. Gravitational wave emission as described by Peters 1964 will have a smooth orbit averaged change in energy which causes the orbital evolution seen in Chapter 3. Once resonance is considered the evolution will not be smooth. The resonant modes are not constantly removing energy from the orbit like gravitational waves, but only come into effect once resonance conditions are met. This creates spikes for the change of energy in the orbit. Energy will continuously decrease from gravitational wave emission and once resonance occurs additional energy will be extracted for as long as the WD remains in resonance. If the WD manages to reach a saturated mode an even greater amount of energy will be removed from orbit into the object, causing evolution curves to exhibit much more dynamic behavior.

Instead of the usual smooth curve from gravitational wave emission a source that goes through resonance will have bursts of increased energy loss leading to sudden orbital shrinking that will change the evolution curve to show steeper declines in semi-major axis once resonance is achieved, and once the source leaves resonance the curve will flatten out as the source resumes exclusively gravitational wave emission. The histograms of orbital frequency will change based on the binning of frequencies, because once resonance is met there will be sudden increase in the count of orbital frequencies in the ’resonance bin’ which will shift the histograms to show spikes where resonance is met as oppose to the relatively smooth distribution seen from gravitational wave emission. The orbital evolution could also be affected by the WD reaching a certain amount of deposited resonant energy which is discussed further in Chapter 5.

CHAPTER 5
CONCLUSION

5.1 Summarize application of code and results gathered

Several codes were developed and applied to the COSMIC populations for this research. First the population synthesis code using COSMIC found in appendix A is used to create the population of either WD-WD or WD-NS systems. In appendix B there is the code used to filter through the COSMIC csv file to find the relevent evolutionary states needed for analysis. The code used for orbital evolution via gravitational wave emission exclusively is seen in Appendix C. Appendix D has a second filtration code used for taking the gravitationally evolved sources and finding any sources that meet the initial resonance requirement for dual evolution, and Appendix E contains the dual evolution code used on sources found from the resonance filtration code. The results of WD-WD and WD-NS evolution via gravitational wave emission is presented for a single system and a population with corresponding orbital frequency histograms. The affect that Dual Evolution will have on the behavior of orbital frequency for these systems is discussed.

5.2 Future work

The incorporation and usage of the FMcKE in the dual evolution model for WD binaries has some simplifications and nuances that need to be expanded on in future work to have a more accurate model. One issue that needs attention is the strictness of the resonant condition, the current initial resonant limit set by $\pm 1/t_d$ requires the source to fall within a two picoHertz range causing most sources to pass over resonance in evolution timestep within the COSMIC dataframe. Another interesting issue is that the forcing frequency term is currently negative for the vast majority of sources synthesized, but the lower limit of the resonant condition has a positive value. Equations 2.10 and 2.11 for the energy deposited into a resonant mode are only applicable in the small,

< 0.05, eccentricity range, for higher eccentricity values such as those found in first evolutionary stages generated by COSMIC these equations require modification. The numerical expense of the harmonics and primary quantum numbers is less impactful but still relevant to a higher level of accuracy for the model. As previously discussed in Chapter 2, the dual evolution code truncates the process once the tenth harmonic or tenth low order radial mode has been reached, but in future models with more powerful computational ability higher order modes could be incorporated. Another computational recommendation is to have this dual evolution process incorporated into the COSMIC engine, so the sources do not need to be extracted and evolved seperately until mass-transfer begins and then placed back into COSMIC.

The post evolution analysis has several aspects that should be addressed. There might be an environmental dependence on the resonance condition of the WDs causing either an increase or decrease on resonant forcing time leading to changes in the amount of energy deposited into a mode. A second method involves the end state of the WD. The only two end states considered in this thesis are from the previously described Roche Lobe limit being reached making the binary begin mass transfer or the observation time of LISA has been reached. The third option that needs to be expanded on is the possibility that the star undergoes carbon-fusion via ignition. This process occurs when energy is deposited into the WD with a change of energy being released into the Helium atmosphere causing a dynamical equilibrium to be reached between energy deposited into the WD and the energy within the atmosphere. If this equilibrium condition is met the atmosphere will ignite causing a nova or supernova, but the exact calculations on how to reach this equilibrium have not been performed yet.

The most immediate future goal for this work is to mesh the COSMIC and Dual Evolution processes together, so that the compact binary systems undergo a full astrophysical and gravitational wave treatment without needing to extract a dataframe of sources to test. This would allow COSMIC to find and test sources immediately without any third party interference in the evolutionary process.

5.3 Thanks

Thank you to my committee members for reading and providing feedback on this thesis. Thank you Northwestern University for providing me the opportunity to conduct my thesis research at this institution.

REFERENCES

[1] Abbott, B. P. "Observation of Gravitational Waves from a Binary Black Hole Merger." Centennial of General Relativity, 2017, pp. 291–311.

[2] Burkart J., Quataert E., Arras P., Weinberg N. N., 2013, MNRAS, 433, 332

[3] Eggleton P. P., 1983, ApJ, 268, 368

[4] J. R. Hurley, C. A. Tout, and O. R. Pols. Evolution of Binary Stars and the Effect of Tides on Binary Populations. Monthly Notices of the Royal Astronomical Society, 329(4):897–928, Feb 2002

[5] Kumar P., Goodman J., 1996, ApJ, 466, 946

[6] Mckernan, B., and K. E. S. Ford. "On the Resonant Detonation of Sub-Chandrasekhar Mass White Dwarfs during Binary Inspiral." Monthly Notices of the Royal Astronomical Society, vol. 463, no. 2, 2016, pp. 2039–2045

[7] Peters P. C., 1964, Phy. Rev., 136, 1224

Appendices

WHITE DWARF BINARY ORBIT EVOLUTION USING STELLAR RESONANCE AND GRAVITATIONAL WAVE EMISSION

Approved by:

Dr. Larson, Advisor
The Graduate School CIERA
Northwestern University

Dr. Novak
The Graduate School CIERA
Northwestern University

Date Approved: August 17, 2020

APPENDIX A

COSMIC POPULATION CREATION

```
!pip install -q cosmic-popsynth==3.3.0

In [1]: from cosmic.sample.initialbinarytable import InitialBinaryTable

In [2]: from cosmic.evolve import Evolve
from cosmic.sample.sampler import independent
from google.colab import files
import pandas as pd
from math import *
from scipy import interpolate
import matplotlib.pyplot as plt
import numpy as np
final_kstar1=[11]
final_kstar2=[14]
BSEDict = {’xi’: 1.0, ’bhflag’: 1, ’neta’: 0.5, ’windflag’: 3, ’wdflag’: 1, ’alpha1’: 1.0, ’pts1’: 0.001, ’pts3’: 0.02,
        ’pts2’: 0.01, ’epsnov’: 0.001, ’hewind’: 0.5, ’ck’: 1000, ’bwind’: 0.0, ’lambdaf’: 0.5,
               ’mxns’: 2.5, ’beta’: 0.125, ’tflag’: 1, ’acc2’: 1.5, ’nsflag’: 3, ’ceflag’: 0,
               ’eddfac’: 1.0, ’ifflag’: 0, ’bconst’: 3000, ’sigma’: 265.0, ’gamma’: -1.0, ’pisn’: 45.0,
               ’natal_kick_array’ : [[-100.0,-100.0,-100.0,-100.0,0.0],[-100.0,-100.0,-100.0,-100.0,0.0]],
               ’bhsigmafrac’ : 1.0, ’polar_kick_angle’ : 90, ’qcrit_array’ : [0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0],
               ’cekickflag’ : 2, ’cehestarflag’ : 0, ’cemergeflag’ : 0, ’
```

```
ecsn' : 2.5, 'ecsn_mlow' : 1.4,
                 'aic' : 1, 'ussn' : 0, 'sigmadiv' :-20.0, 'qcflag' : 2, '
eddlimflag' : 0,

                 'fprimc_array' :
[2.0/21.0,2.0/21.0,2.0/21.0,2.0/21.0,2.0/21.0,2.0
                 /21.0,2.0/21.0,2.0/21.0,2.0

/21.0,2.0/21.0,2.0/21.0,2.0/21.0,2.0/21.0,2.0/21.0,2.0/21.0,2.0/21.0],
                 'bhspinflag' : 0, 'bhspinmag' : 0.0, 'rejuv_fac' : 1.0, '
rejuvflag' : 0, 'htpmb' : 1, 'ST_cr' : 1, 'ST_tide' : 0, 'bdecayfac' : 1,
'remnantflag' : 3, 'zsun' : 0.014,'kickflag': 0,'rembar_massloss' : 0.5}
InitialBinaries, mass_singles, mass_binaries, n_singles, n_binaries =
    InitialBinaryTable.sampler('independent', final_kstar1, final_kstar2,
    binfrac_model=0.5, primary_model='kroupa93', ecc_model='thermal',
    SFH_model='const', porb_model='sana12', SF_start=13700.0, SF_duration=0,
    component_age=10000.0, met=0.02, size=100000)
bpp, bcm, initC, kick_info  = Evolve.evolve(initialbinarytable=InitialBinaries
    , BSEDict=BSEDict)

bpp

bpp.to_csv('NameFile')
files.download('NameFile')
```

APPENDIX B

COSMIC FILTRATION

```
import pandas as pd
from math import *
from scipy import interpolate
import matplotlib.pyplot as plt
import numpy as np
import time, threading

pd.set_option('display.max_columns', None)
#read in data from cosmic using path to file
cosmicpop = pd.read_csv(r"PathToFile")
print(cosmicpop.columns.values)

#drop zero mass sources
cosmicpop_nom1_0=cosmicpop[(cosmicpop['mass_1'] ==0)].index
cosmicpop.drop(cosmicpop_nom1_0 , inplace=True)
cosmicpop_nom2_0=cosmicpop[(cosmicpop['mass_2'] ==0)].index
cosmicpop.drop(cosmicpop_nom2_0 , inplace=True)
#drop 0,-1 porb sources
cosmicpop_f1=cosmicpop[(cosmicpop['porb'] ==0)].index
cosmicpop.drop(cosmicpop_f1, inplace=True)
cosmicpop_f0=cosmicpop[(cosmicpop['porb'] ==-1)].index
cosmicpop.drop(cosmicpop_f0, inplace=True)
#define detectable sources
#detectable=cosmicpop[(cosmicpop['porb']<1461.03)]
#drop 0,-1 ecc sources
cosmicpop_ecc0=cosmicpop[(cosmicpop['ecc'] ==0)].index
cosmicpop.drop(cosmicpop_ecc0, inplace=True)
```

```
cosmicpop_ecc1=cosmicpop[(cosmicpop['ecc'] ==-1)].index
cosmicpop.drop(cosmicpop_ecc1, inplace=True)
#drop ecc>0.1 sources
#cosmicpop_eccbig=cosmicpop[(cosmicpop['ecc'] >0.1)].index
#cosmicpop.drop(cosmicpop_eccbig, inplace=True)
#drop non CO-CO White dwaves k_star1,2=11
cosmicpop_kstar1=cosmicpop[(cosmicpop['kstar_1'] != 11)].index
cosmicpop.drop(cosmicpop_kstar1, inplace=True)
cosmicpop_kstar2=cosmicpop[(cosmicpop['kstar_2'] != 13)].index
cosmicpop.drop(cosmicpop_kstar2, inplace=True)

#read in variables from COSMIC dataframe
#orbital period drops
porb=cosmicpop.porb
porb=porb.replace([0,-1], np.nan)
porb.dropna(inplace=True)
mass_1=cosmicpop.mass_1
mass_2=cosmicpop.mass_2
mass_1=mass_1.replace([0], np.nan)
mass_1.dropna(inplace=True)
mass_2=mass_2.replace([0], np.nan)
mass_2.dropna(inplace=True)
#WD radii
mass1_r=cosmicpop.rad_1
mass2_r=cosmicpop.rad_2
#eccentricity of orbit
ecc=cosmicpop.ecc
kstar_1=cosmicpop.kstar_1
kstar_2=cosmicpop.kstar_2
#separation is semi-major
a_b=cosmicpop.sep
#angular velocity of WD
w_star1=cosmicpop.omega_spin_1
```

```
w_star2=cosmicpop.omega_spin_2

#Convert quantities into SI units
#convert orbital period from days to seconds
PorbSI=porb*86400
#PorbYear=porb/356.256
#PorbDet=detectable
#convert mass1/2 from solar mass to kg
mass_1SI=mass_1*1.98847e30
mass_2SI=mass_2*1.98847e30
#convert radii to m
mass_1rSI=mass1_r*6.96e8
mass_2rSI=mass2_r*6.96e8
#chirp mass
#drop values that are 0
mc=((mass_1SI*mass_2SI)**(3/5)/((mass_1SI+mass_2SI)**(1/5)))
mc=mc.replace([0],np.nan)
mc.dropna(inplace=True)
#semi-major for R solar to m
a_b=a_b*6.96e8
#angular velocity from yr-1 to s-1
w_star1=w_star1*3.156e7
w_star2=w_star2*3.156e7

#write filtered and converted values to new csv
data=[PorbSI, mass_1SI, mass_2SI, mass_1rSI, mass_2rSI, mc, ecc, a_b, w_star1,
     w_star2, kstar_1, kstar_2]
COSMICfilteredPop=pd.DataFrame(data)
cosmicfilterpop=COSMICfilteredPop.transpose()
cosmicfilterpop=cosmicfilterpop.rename(columns={"porb": "PorbSI", "mass_1": "
    mass_1SI", "mass_2": "mass_2SI", "rad_1": "mass_1rSI", "rad_2": "mass_2rSI
    ", "Unnamed 0": "mc", "sep": "a_b", "omega_spin_1": "w_star1", "
    omega_spin_2": "w_star2"})
```

```
cosmicfilterpop.to_csv(r"PathToFile")
```

APPENDIX C

GRAVITATIONAL WAVE EVOLUTION

```
import pandas as pd
from math import *
from scipy import interpolate
import matplotlib.pyplot as plt
import numpy as np
import decimal

#read in data from cosmic using path to file
cosmicpop = pd.read_csv(r"PathToFile", index_col=False)
PorbSI=cosmicpop.PorbSI
mass_1SI=cosmicpop.mass_1SI
mass_2SI=cosmicpop.mass_2SI
mass_1rSI=cosmicpop.mass_1rSI
mass_2rSI=cosmicpop.mass_2rSI
a_b=cosmicpop.a_b
ecc=cosmicpop.ecc
mc=cosmicpop.mc

#re convert values for sensitivity
############
# PorbDay=PorbSI/86400
# #PorbYear=porb/356.256
# #PorbDet=detectable
# #convert mass1/2 from solar mass to kg
# mass_1Solar=mass_1SI/1.98847e30
# mass_2Solar=mass_2SI/1.98847e30
# #convert radii to m
```

```
# mass_1rSolar=mass_1rSI/6.96e8
# mass_2rSolar=mass_2rSI/6.96e8
# #chirp mass
# #drop values that are 0
# mc=((mass_1Solar*mass_2Solar)**(3/5)/((mass_1Solar+mass_2Solar)**(1/5)))
# #semi-major for R solar to m
# a_b=a_b/6.96e8
# w_star1=cosmicpop.w_star1
# w_star2=cosmicpop.w_star2
# #angular velocity from s^-1 to yr^-1
# w_star1=w_star1/86400
# w_star2=w_star2/86400

#constants
#gravitational converted from AU**3*yr**-2*Msolar**-1 to Solar Radii*day**-2*
    Msolar**-1
G=3100670.345563776
#speed of light in solar radii per day
c=37231.7
#define roche lobe radii
q_1=mass_1Solar/mass_2Solar
r_l_1=(0.49*q_1**(2/3))/(0.6*q_1**(2/3)+np.log(1+q_1**(1/3)))
q_2=mass_2Solar/mass_1Solar
r_l_2=(0.49*q_2**(2/3))/(0.6*q_2**(2/3)+np.log(1+q_2**(1/3)))
r_l_tot=r_l_1+r_l_2
#reduced mass
mu=(mass_1Solar*mass_2Solar)/(mass_1Solar+mass_2Solar)
#times
#t=0
#tobs=1.262*10**8
t_gw=(5/128)*((c**5)/(G**3))*((a_b**4)/((mass_1Solar+mass_2Solar)**2*mu))*(1-
    ecc**2)**(7/2)
#Constants required for gwave evolution
```

```
Beta=(64/5)*c**(-5)*G**3*mass_1Solar*mass_2Solar*(mass_1Solar+mass_2Solar)

a0=a_b

c0=a_b*(1-ecc**2)*ecc**(-12/19)*(1+(121/304)*ecc**2)**(-870/2299)

cn=a_b*(1-ecc**2)*ecc**(-12/19)*(1+(121/304)*ecc**2)**(-870/2299)

porb0=(a0**3/(mass_1Solar+mass_2Solar))**(1/2)

porb=(a_b**3/(mass_1Solar+mass_2Solar))**(1/2)

freq0=2/porb0

Eorb=-(G*(mass_1Solar+mass_2Solar))/(2*a_b)

E_GW=0

eorbtot=Eorb+E_GW

n=1
#evolve values
semi=[]
newecc=[]
con=[]
NewOrb=[]
orbitalperiod=[]
orbitalfrequency=[]
chirpmass=[]
mass1=[]
mass2=[]
mass1r=[]
mass2r=[]
```

```
wstar1=[]
wstar2=[]
for i in range(1):
    con.append(c0)
    semi.append(a0)
    newecc.append(ecc)
    orbitalperiod.append(porb0)
    orbitalfrequency.append(freq0)
    NewOrb.append(eorbtot)
    chirpmass.append(mc)
    mass1.append(mass_1Solar)
    mass2.append(mass_2Solar)
    mass1r.append(mass_1rSolar)
    mass2r.append(mass_2rSolar)
    wstar1.append(w_star1)
    wstar2.append(w_star2)


for i in range(1,len(cosmicpop)):
    while n<=100:
        timestep_new=n*t_gw/100
        timestep_old=(n-1)*t_gw/100

        #dt=timestep_new-timestep_old
        dt=porb

        decc=(-19/12)*(Beta/cn**4)*((ecc**(-29/19))*(1-ecc**2)**(3/2))
   /((1+(121/304)*ecc**2)**(1181/2299))
        ecc+=decc*dt
        newecc.append(ecc)

        #gwave energy
        dE_GW=-(32/5)*(((G**4*mass_1Solar**2*mass_2Solar**2)*(mass_1Solar+
```

```
   mass_2Solar))/(c**5*a_b**5*(1-ecc**2)**(7/2)))*(1+(73/24)*ecc**2+(37/96)*
   ecc**4)
        E_GW+=dE_GW*dt

        Eorb=-(G*(mass_1Solar+mass_2Solar))/(2*a_b)

        eorbtot=Eorb+E_GW

        #a_b=((cn*ecc**(12/19))/(1-ecc**2))*(1+(121/304)*ecc**2)**(870/2299)
        a_b=(-G*mass_1Solar*mass_2Solar)/(2*eorbtot)
        semi.append(a_b)

        porb=(a_b**3/(mass_1Solar+mass_2Solar))**(1/2)
        orbitalperiod.append(porb)

        fre=2/porb
        orbitalfrequency.append(fre)

        cn=a_b*(1-ecc**2)*(ecc**(-12/19))*(1+(121/304)*ecc**2)**(-870/2299)
        con.append(cn)
        chirpmass.append(mc)
        mass1.append(mass_1Solar)
        mass2.append(mass_2Solar)
        mass1r.append(mass_1rSolar)
        mass2r.append(mass_2rSolar)
        wstar1.append(w_star1)
        wstar2.append(w_star2)
        n=n+1
    if np.any(r_l_tot>=a_b):
        break

#zip files to flatten
mesemi=list(zip(*semi))
```

```
meorb=list(zip(*orbitalperiod))
mefre=list(zip(*orbitalfrequency))
mechirp=list(zip(*chirpmass))
mecc=list(zip(*newecc))

data=[mesemi, meorb, mefre,mechirp, mass1, mass2, mass1r, mass2r, wstar1,
    wstar2, mecc]
#flatten list of lists into list
x=[item for sublist in mesemi for item in sublist]
y=[item for sublist in meorb for item in sublist]
z=[item for sublist in mefre for item in sublist]
xx=[item for sublist in mechirp for item in sublist]
yy=[item for sublist in mass1 for item in sublist]
zz=[item for sublist in mass2 for item in sublist]
xxx=[item for sublist in mass1r for item in sublist]
yyy=[item for sublist in mass2r for item in sublist]
zzz=[item for sublist in wstar1 for item in sublist]
xyz=[item for sublist in wstar2 for item in sublist]
zzzz=[item for sublist in mecc for item in sublist]

data=[x,y,z,xx, yy, zz, xxx, yyy, zzz, xyz, zzzz]
#create new dataframe
df=pd.DataFrame(data)
df=df.transpose()
df=df.rename(columns={0: 'SemiMajorAxis', 1: 'OrbitalPeriod', 2: '
    OrbitalFrequency', 3:'Chirpmass', 4:'mass1',5:'mass2',6:'mass1r',7:'mass2r
    ',8:'wstar1',9:'wstar2', 10:'ecc'})


#read in values from new data frame
sma=df.SemiMajorAxis
sma.replace(np.nan,0)
op=df.OrbitalPeriod
```

```
op.replace(np.nan,0)
of=df.OrbitalFrequency
of.replace(np.nan, 0)
cm=df.Chirpmass
mass1=df.mass1
mass2=df.mass2
mass1r=df.mass1r
mass2r=df.mass2r
wstar1=df.wstar1
wstar2=df.wstar2
eccen=df.ecc
eccen.replace(np.nan,0)

#create new data frame for export
newdata=[sma, op, of, cm, mass1, mass2, mass1r, mass2r, wstar1, wstar2,eccen]
newdf=pd.DataFrame(newdata)
newdf=newdf.transpose()
newdf=newdf.rename(columns={’sma’: ’SemiMajorAxis’, ’op’: ’OrbitalPeriod’, ’of’: ’OrbitalFrequency’, ’cm’:’Chirpmass’, ’eccen’:"eccentricity"})

newdf.to_csv(r"PathToFile")
```

APPENDIX D

RESONANT SOURCE FINDER

```
import pandas as pd
from math import *
from scipy import interpolate
import matplotlib.pyplot as plt
import numpy as np
from decimal import *
getcontext().prec=20

#import data
cosmicpop = pd.read_csv(r"YourPathway", index_col=False)

#define values from df
porb=cosmicpop.OrbitalPeriod
mass_1Solar=cosmicpop.mass1
mass_2Solar=cosmicpop.mass2
mass_1rSolar=cosmicpop.mass1r
mass_2rSolar=cosmicpop.mass2r
a_b=cosmicpop.SemiMajorAxis
#ecc=cosmicpop.ecc
mc=cosmicpop.Chirpmass
w_star1=cosmicpop.wstar1
w_star2=cosmicpop.wstar2
ecc.cosmicpop.ecc
#constants
#gravitational converted from AU**3*yr**-2*Msolar**-1 to Solar Radii*day**-2*
   Msolar**-1
G=3100670.345563776
```

```
#speed of light in solar radii per day
c=37231.7

#define roche lobe radii
q_1=mass_1Solar/mass_2Solar
r_l_1=(0.49*q_1**(2/3))/(0.6*q_1**(2/3)+np.log(1+q_1**(1/3)))
q_2=mass_2Solar/mass_1Solar
r_l_2=(0.49*q_2**(2/3))/(0.6*q_2**(2/3)+np.log(1+q_2**(1/3)))
r_l_tot=r_l_1+r_l_2
#reduced mass
mu=(mass_1Solar*mass_2Solar)/(mass_1Solar+mass_2Solar)

#break up spin fraction
fspin1=-1
w_spin1=fspin1*w_star1


#principal quantum numbers
q=1
m=2
#harmonic counter
k=2
#forcing frequency
w_force=q*porb-m*w_spin1

#initial decay time (days)
tdp=11574074.074074075
tdg=11574074.074074075

#find the resonnat limits to high precision
for i in range(len(cosmicpop)):
    newhi=Decimal(w_star1[i]/k+1/tdp)
    newlo=Decimal(w_star1[i]/k-1/tdp)
```

```


#initial mode coefficients
d_res_mode_p=np.float32(2)
d_res_mode_g=0.1

#forcing frequency range by varying break spin fraction
rangeforce=[]
breakup=[]
for i in range(len(cosmicpop)):
    while fspin1<=1:
        w_spin1=fspin1*w_star1
        w_force=q*porb-m*w_spin1
        fspin1+=0.01
        rangeforce.append(w_force)
        breakup.append(fspin1)

#flatten list of lists
spinbreakrange=list(zip(*rangeforce))

newforcerange=[item for sublist in spinbreakrange for item in sublist]

#source meets resonance criteria
resonantsource=[]
breakfrac=[]
for i in range(len(cosmicpop)):
    if (Decimal(newlo)-Decimal(1*10**-5)) <=newforcerange[i] and newforcerange
    [i]<=(Decimal(newhi)+Decimal(1*10**-5)):
        resonantsource.append(newforcerange[i])
        breakfrac.append(breakup[i])

modecoeffg=[]
modecoeffp=[]
```

```
#find the type of mode by finding the coefficient using division
for i in range(len(resonantsource)):
    starcoeff=resonantsource[i]/w_star1[i]
    #pmode
    if starcoeff>=2:
        modecoeffp.append(starcoeff)
    #gmode
    elif starcoeff<=1:
        modecoeffg.append(starcoeff)
```

APPENDIX E

DUAL EVOLUTION

```
import pandas as pd
from math import *
from scipy import interpolate
import matplotlib.pyplot as plt
import numpy as np
from decimal import *
getcontext().prec=20


#import data
cosmicpop = pd.read_csv(r"PathToFile", index_col=False)

#define values from df
PorbSI=cosmicpop.PorbSI
mass_1SI=cosmicpop.mass_1SI
mass_2SI=cosmicpop.mass_2SI
mass_1rSI=cosmicpop.mass_1rSI
mass_2rSI=cosmicpop.mass_2rSI
a_b=cosmicpop.a_b
ecc=cosmicpop.ecc
mc=cosmicpop.mc
w_spin1=cosmicpop.w_spin1
w_spin2=cosmicpop.w_spin2

#constants
#gravitational converted from AU**3*yr**-2*Msolar**-1 to Solar Radii*day**-2*
    Msolar**-1
```

```
G=3100670.345563776
#speed of light in solar radii per day
c=37231.7
p=0
#define roche lobe radii
q_1=mass_1Solar/mass_2Solar
r_l_1=(0.49*q_1**(2/3))/(0.6*q_1**(2/3)+np.log(1+q_1**(1/3)))
q_2=mass_2Solar/mass_1Solar
r_l_2=(0.49*q_2**(2/3))/(0.6*q_2**(2/3)+np.log(1+q_2**(1/3)))
r_l_tot=r_l_1+r_l_2
#reduced mass
mu=(mass_1Solar*mass_2Solar)/(mass_1Solar+mass_2Solar)
#times
#t=0
tobs=1461.03
t_gw=(5/128)*((c**5)/(G**3))*((a_b**4)/((mass_1Solar+mass_2Solar)**2*mu))*(1-
    ecc**2)**(7/2)

#variables for gwave evolution
Beta=(64/5)*c**(-5)*G**3*mass_1Solar*mass_2Solar*(mass_1Solar+mass_2Solar)

a0=a_b

c0=a_b*(1-ecc**2)*ecc**(-12/19)*(1+(121/304)*ecc**2)**(-870/2299)

cn=a_b*(1-ecc**2)*ecc**(-12/19)*(1+(121/304)*ecc**2)**(-870/2299)

porb0=(a0**3/(mass_1Solar+mass_2Solar))**(1/2)

porb=(a_b**3/(mass_1Solar+mass_2Solar))**(1/2)



freq0=(2*pi)/porb0
```

```
E_GW=0

Eorb=-(G*(mass_1Solar+mass_2Solar))/(2*a_b)

eorbtot=Eorb+E_GW

#principal quantum numbers
q=1
m=2
#
#forcing frequency
w_force=q*freq0-m*w_spin1
#mode counter
d_res_mode_g=0.1
d_res_mode_p=2
#energy of WD and NS
Estar_1=(G*mass_1Solar**2)/mass_1rSolar
Estar_2=(G*mass_2Solar**2)/mass_2rSolar
Esys=Estar_1+Estar_2
#initial mode test
pmode0=d_res_mode_p*w_star1
gmode0=d_res_mode_g*w_star1

#harmonic counter
k=2
r=1
#typical decay time (seconds)
t_p=1*10**7
#number of radial modes
Np=10
#initial decay time of mode (p or g) measured in days
tdp=1*10**12
```

```
tdg=1*10**12
#initial energy deposited into modes
Ep0=0
Eg0=0
Ep=Ep0
Eg=Eg0
#total resonant energy
Ert0=Ep0+Eg0
Ert=Ep+Eg
#initial total energy
#Etotal0=Ert0+Esys+Eorb0+E_GW0

#grav wave lists
orbital_period=[]
Grav_wave_energy=[]
semi_major=[]
frequency=[]
newecc=[]
#lists for FMcKE relevant sources
E_res_p=[]
E_res_g=[]
f_res_p=[]
f_res_g=[]
E_res_tot=[]
#total energy list
E_tot=[]
#evolved orbital energy
correct_orbital_energy=[]
#forcing freq
force_freq=[]
#res conds
hires=[]
lowres=[]
```

```
#
delres=high_end-w_force
del_res_cond=[]
#
pmode=[]
gmode=[]
orbyengy=[]


#bin counter
n=1
#append intial values into lists
for i in range(1):
    #normal stuff
    semi_major.append(a0)
    newecc.append(ecc)
    orbital_period.append(porb0)
    frequency.append(freq0)
    #fmcke stuff
    E_res_p.append(Ep0)
    E_res_g.append(Eg0)
    Grav_wave_energy.append(E_GW)
    E_res_tot.append(Ert0)
    ###
    #E_tot.append(Etotal0)
    ####
    #changed orbital energy
    correct_orbital_energy.append(eorbtot)
    #record forcing frequency
    force_freq.append(w_force)
    hires.append(high_end)
    lowres.append(low_end)
    del_res_cond.append(delres)
```

```
    #mode check
    pmode.append(pmode0)
    gmode.append(gmode0)
    orbyengy.append(Eorb)
#begin the dual evolution loop
for i in range(1,len(cosmicpop)):
    while n<=200:
        timestep_new=n*t_gw/100
        timestep_old=(n-1)*t_gw/100

        #define time step for bins
        #dt=timestep_new-timestep_old

        dt=porb
        #fchirp=(96/5)*((c**3)/G)*(f/mc)*((G/c**3)*pi*f*mc)**(8/3)



        #begin FMcKE if resonant condition is met
        while low_end[i]<=w_force[i]<=high_end[i]:
            k=0
            r=0
            #is the resonant mode a gravity mode?
            while w_force[i]==d_res_mode_g*w_star1[i]:
                #truncate mode if harmonic/quantum# is too high
                if k==10 or q==10:
                    break

                #freqency of mode
                w_jg=d_res_mode_g*w_star1
                #overlap term for overlapping resonance
                X_jg=27*(w_jg/w_star1)**3.7
                #Satured g mode terms
```

```
            w_force_dotg=w_jg-(1/(r*2*tdg**2))
            t_forceg=1/((w_force_dotg)**1/2)


            #is the mode saturated?
            if t_forceg>tdg:
                #energy deposited
                Eg=(10**-5)*Estar_1*((w_jg/w_star1)/0.1)**(-2/3)*(X_jg/10**-3)*((mc/mass_1Solar)**(4/5))*((a_b/mass_1rSolar)/10)**-4
                E_res_g.append(Eg)
            #if not saturated use unsaturated energy
            else:
                Eg=((10**-5)*Estar_1*((w_jg/w_star1)/0.1)**(-2/3)*(X_jg/10**-3)*((mc/mass_1Solar)**(4/5))*((a_b/mass_1rSolar)/10)**-4)*(10**(-6)*(t_forceg/0.115)**(2/3)*(tdg/(115.74*10**6))**(-2/3))
                E_res_g.append(Eg)


            #cooling time of mode via thermal process
            t_j_ktg=(((4*pi)/w_j1)*(Estar_1/Eg)**(1/2))
            #cooling time of mode via grav wave
            t_j_GWg=((5*c**5)/G)*(1/(Eg*w_j1**2))
            #decay time of mode
            tdg=t_j_ktg+t_j_GWg
            w_force1+=w_force_dot1*dt


            #update mode
            d_res_mode_g+=0.1
            r+=1
            #harmonic count up
            k+=1

```

```

            #update resonant conditions
            low_end=w_star1/k-1/tdg
            high_end=w_star1/k+1/tdg


        #is the resonant mode a pressure mode?
        while w_force[i]==d_res_mode_p*w_star1[i]:
            #truncate
            if k==10 or q==10:
                break

            w_jp=d_res_mode_p*w_star1
            X_jp=27*(w_jp/w_star1)**3.7
            Np=5


            #Satured p_mode
            w_force_dotp=w_jp-(1/(r*2*tdp**2))
            t_forcep=1/((w_force_dotp)**1/2)
            if t_force1>tdp:
                Ep=0.1*Estar_1*((w_star1/(3.4722*10**(-6)))**(4/5))*(((w_jp/wstar_1))/4)**(2/5)*((k/1)**(-2))*((mc/mass_1Solar)**(4/5))*(((a_b/mass_1rSolar)/10)**(-12/5))*((X_jp/(10**(-3)))**(3/5))*((t_p/115.74)**(4/5))*((fchirp/0.5)**(-4/5))*(((Np/10**2))**(-4/5))
                E_res_p.append(Ep)
            #unsaturated energy
            else:
                Ep=0.1*Estar_1*((w_star1/(3.4722*10**(-6)))**(4/5))*(((w_jp/w_star1))/4)**(2/5)*((k/1)**(-2))*((mc/mass_1Solar)**(4/5))*(((a_b/mass_1rSolar)/10)**(-12/5))*((X_jp/(10**(-3)))**(3/5))*((t_p/(115.74)**(4/5)))*((fchirp/0.5)**(-4/5))*(((Np/10**2))**(-4/5))*(0.06*(t_forcep/0.1157)**(2/5)*(tdp/115.74)**(-2/5))
```

```
                    E_res_p.append(Ep)

                #decay time of pmode
                t_j_ktp=2*(t_p/Np)*(Estar_1/E1)
                t_j_GWp=((5*c**5)/G)*(1/(E1*w_j1**2))
                tdp=t_j_ktp+t_j+GWp

                #harmonic counter
                k+=1
                r+=1
                d_res_mode_p+=1

                #update resonant conditions

                low_end=w_star1/k-1/tdp
                high_end=w_star1/k+1/tdp

        #change eccentricity
        decc=(-19/12)*(Beta/cn**4)*((ecc**(-29/19))*(1-ecc**2)**(3/2))
    /((1+(121/304)*ecc**2)**(1181/2299))
        ecc+=decc*dt
        newecc.append(ecc)



        #calc total energy of binary
        #gravitational wave energy
        dE_GW=-(32/5)*(((G**4*mass_1Solar**2*mass_2Solar**2)*(mass_1Solar+
    mass_2Solar))/(c**5*a_b**5*(1-ecc**2)**(7/2)))*(1+(73/24)*ecc**2+(37/96)*
    ecc**4)
        E_GW+=dE_GW*dt
        Grav_wave_energy.append(E_GW)

        #energy deposited into resonant mode
```

```
        Ert=Eg+Ep
        E_res_tot.append(Ert)

        #orbital energy
        Eorb=-(G*(mass_1Solar+mass_2Solar))/(2*a_b)
        orbyengy.append(Eorb)

        #new orbital energy accounting for GWave Emission/Resonant deposits
        if Ert !=0:
            eorbtot=Eorb+E_GW+Ert
        else: eorbtot=Eorb+E_GW
        correct_orbital_energy.append(eorbtot)

        #calculate semi-major using total energy or Kepler III
        a_b=(-G*mass_1Solar*mass_2Solar)/(2*eorbtot)
        #a_b=((porb**2*G*mass_1Solar*mass_2Solar)/(4*pi**2*mu))**(1/3)
        semi_major.append(a_b)



        #orbital period using energy or Kepler III
        porb=(4*pi**2*a_b**3/(G*(mass_1Solar+mass_2Solar)))**(1/2)
        #porb=pi*G*mass_1Solar*mass_2Solar*(mu/2)**(1/2)*(abs(Etotal)**(-3/2))
        orbital_period.append(porb)



        f=(2*\pi)/porb
        frequency.append(f)

        cn=a_b*(1-ecc**2)*(ecc**(-12/19))*(1+(121/304)*ecc**2)**(-870/2299)



        #forcing frequency of WD
        w_force=q*f-m*w_spin1
```

```
        force_freq.append(w_force)

        hires.append(high_end)
        lowres.append(low_end)

        #how far away from res
        delres=high_end-w_force
        del_res_cond.append(delres)

        ppmode=d_res_mode_p*w_star1
        ggmode=d_res_mode_g*w_star1
        pmode.append(ppmode)
        gmode.append(ggmode)
        n=n+1
        if np.all(r_l_tot>=a_b):
            break
####################
```