

Binary Population Simulation and Gravitational Wave Merger Modelling

Devika Chandiran - MSc Astrophysics, Queen Mary University of London

October 2025

Abstract

This report describes a numerical simulation of compact binary systems and their evolution due to gravitational wave emission. The work extends my MSc research project at Queen Mary University of London. Using a simple model based on Peters (1964), the simulation estimates merger times, chirp masses and statistical properties of the population. All equations and results are explained in clear and direct terms for accessibility.

1 Introduction

Gravitational waves provide a direct way to study compact binaries such as black holes and neutron stars. When two massive bodies orbit one another, they emit energy as gravitational radiation, which gradually reduces the orbital separation until they merge. The rate at which this happens depends mainly on the component masses and their initial distance. In this project I created a Python model that generates a random population of binaries and computes their merger times and chirp masses, showing which systems would merge within the lifetime of the Universe.

2 Physical Background

For a circular binary with component masses m_1 and m_2 , and an initial orbital separation a_0 , the time until coalescence due to gravitational wave radiation is given by Peters (1964) as:

$$t_{\text{merge}} = \frac{5}{256} \frac{c^5 a_0^4}{G^3 m_1 m_2 (m_1 + m_2)}. \quad (1)$$

This equation shows that tighter and more massive systems merge faster. The *chirp mass*, which affects the observed gravitational wave frequency and amplitude, is defined as:

$$\mathcal{M} = \frac{(m_1 m_2)^{3/5}}{(m_1 + m_2)^{1/5}}. \quad (2)$$

These two quantities are key to understanding gravitational wave signals and the detectability of sources.

3 Numerical Implementation

3.1 Setup and Constants

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 from scipy.constants import G, c
4 import os
5
6 # Physical constants
7 YEAR = 365.25 * 24 * 3600          # seconds in a year
8 MSUN = 1.98847e30                  # kilograms in one solar mass
9 RSUN = 6.957e8                     # metres in one solar radius
10
11 plt.rcParams.update({'figure.figsize': (7, 5), 'axes.grid': True})
12 print("Environment ready")
```

3.2 Sampling Binary Systems

```
1 def sample_m1(size, mmin=5, mmax=40, alpha=2.35):
2     """
3     Draws the primary mass using a Salpeter power law.
4     """
5     r = np.random.rand(size)
6     a1 = 1 - alpha
7     return ((r * (mmax**a1 - mmin**a1) + mmin**a1)) ** (1 / a1)
8
9 N = 5000 # number of binaries
10 m1 = sample_m1(N)
11 q = np.random.uniform(0.3, 1.0, N) # mass ratio
12 m2 = q * m1
13 a0 = np.exp(np.random.uniform(np.log(1), np.log(100), N)) #
14     separation in solar radii
15 print(f"Generated {N} binary systems")
```

3.3 Merger Time and Chirp Mass

```
1 def merger_time_s(m1_msun, m2_msun, a0_rsun):
2     """
3     Calculates the merger time for circular binaries in seconds.
4     """
5     m1, m2, a0 = m1_msun * MSUN, m2_msun * MSUN, a0_rsun * RSUN
6     num = 5 * c**5 * a0**4
7     den = 256 * G**3 * m1 * m2 * (m1 + m2)
8     return num / den
9
10 tmerge_s = merger_time_s(m1, m2, a0)
11 tmerge_yr = tmerge_s / YEAR
```

```

12
13 def chirp_mass(m1, m2):
14     M = m1 + m2
15     return (m1 * m2)**(3/5) / (M**(1/5))
16
17 chirp = chirp_mass(m1, m2)
18
19 HUBBLE_TIME_YR = 13.8e9
20 merge_mask = tmerge_yr < HUBBLE_TIME_YR
21 print(f"{merge_mask.sum()} binaries merge within a Hubble time")

```

3.4 Visualisation of Results

```

1 os.makedirs("results", exist_ok=True)
2
3 # Plot 1: Merger time distribution
4 plt.figure()
5 plt.hist(np.log10(tmerge_yr[merge_mask]), bins=40,
6          colour="steelblue", edgecolour="black")
7 plt.xlabel("log10 Merger Time [years]")
8 plt.ylabel("Number of systems")
9 plt.title("Merger Time Distribution (within 13.8 Gyr)")
10 plt.tight_layout()
11 plt.savefig("results/merger_times_hist.png", dpi=160)
12 plt.show()
13
14 # Plot 2: Chirp mass versus merger time
15 plt.figure()
16 plt.scatter(tmerge_yr[merge_mask], chirp[merge_mask],
17             s=6, alpha=0.7, colour="darkorange")
18 plt.xscale("log")
19 plt.xlabel("Merger Time [years] (log scale)")
20 plt.ylabel("Chirp Mass [Solar Masses]")
21 plt.title("Chirp Mass vs Merger Time")
22 plt.tight_layout()
23 plt.savefig("results/chirp_vs_tmerge.png", dpi=160)
24 plt.show()
25
26 # Plot 3: Component masses
27 plt.figure()
28 plt.scatter(m1, m2, s=4, alpha=0.4, colour="seagreen")
29 plt.xlabel("m1 [Solar Masses]")
30 plt.ylabel("m2 [Solar Masses]")
31 plt.title("Component Masses of All Systems")
32 plt.tight_layout()
33 plt.savefig("results/mass_scatter.png", dpi=160)
34 plt.show()
35
36 print("Simulation complete")

```

4 Results and Discussion

Figure 1 shows the distribution of merger times for all systems that merge within the age of the Universe. Most binaries have very long inspiral times, while a smaller fraction merge in less than a billion years. Figure 2 plots chirp mass against merger time and shows that heavier systems merge more quickly. Figure 3 shows the range of component masses for all simulated binaries.

These patterns are consistent with physical expectations and with more detailed population synthesis results in the literature. This simple model therefore captures the essential physical relationships in a clear way.

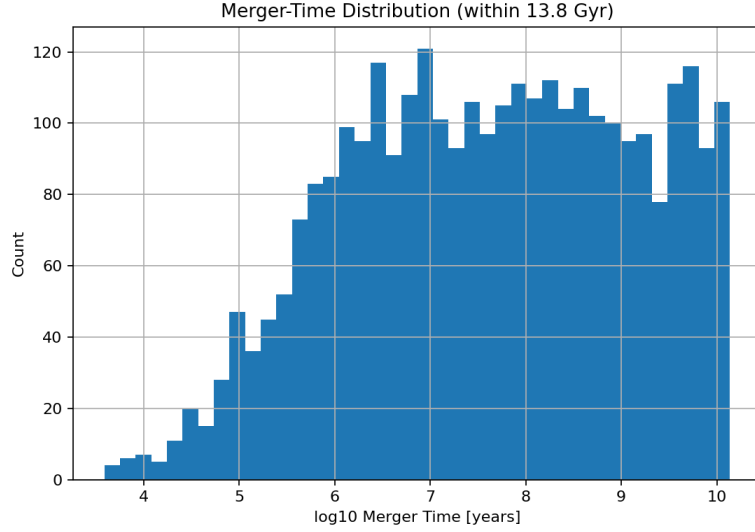


Figure 1: Distribution of merger times (log scale).

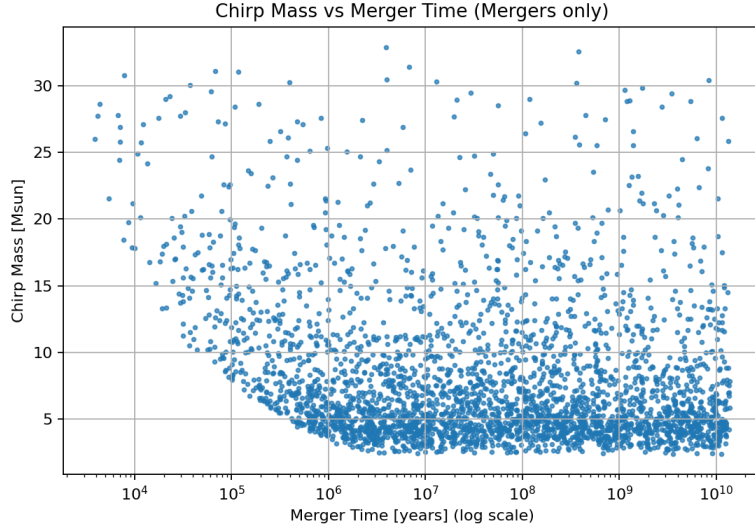


Figure 2: Chirp mass as a function of merger time.

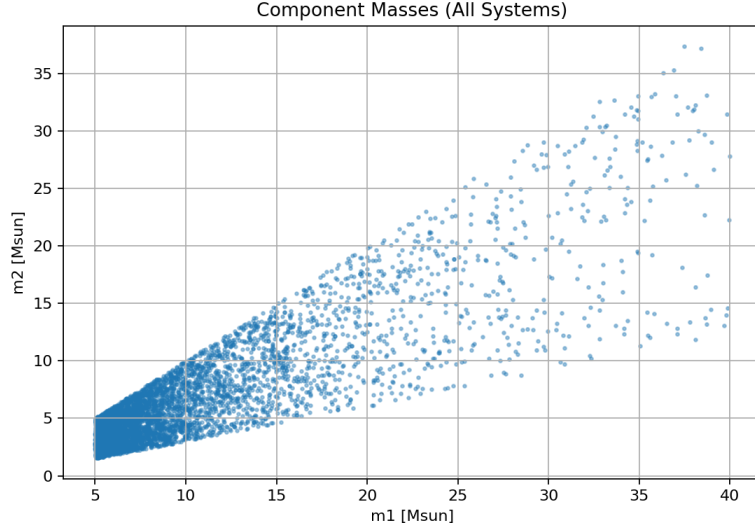


Figure 3: Component masses of all systems.

5 Conclusion

This project extends my MSc research and demonstrates how gravitational wave radiation influences the evolution of binary systems. Although the model is simple, it reproduces the main physical trends and builds a foundation for further work. In future, this can be developed to include metallicity, stellar winds and eccentricity, which would connect closely with the population synthesis research carried out at Radboud University. I am particularly interested in expanding this work to explore how binary evolution models can inform gravitational wave detections and source formation studies within the IMAPP group.

Acknowledgement

This work builds upon my MSc research at Queen Mary University of London. It reflects my continuing interest in computational astrophysics and gravitational wave cosmology.

References

- Peters, P.C. (1964). *Gravitational Radiation and the Motion of Two Point Masses*. Physical Review, 136(4B), B1224.
- Cutler, C. and Thorne, K.S. (2002). *An Overview of Gravitational Wave Sources*. arXiv:gr-qc/0204090.
- Maggiore, M. (2008). *Gravitational Waves: Volume 1*. Oxford University Press.