Matched Filtering of a Gravitational Wave Signal

Lucas Romero Fernández

4^{th} of June 2025

Introduction and Motivation

The further development of the detection methods of Gravitational Waves (GWs) has been one of the main objectives of the astrophysical and General Relativity (GR) scientific community in recent years. In particular, the search for and study of GWs generated from coalescing compact binaries consisting of either black holes (BHs) or neutron stars (NSs) using Earth-based interferometric detectors such as LIGO, Advanced LIGO and VIRGO, among others, has significantly garnered global attention in the research community. Unfortunately, major problems and setbacks have been encountered, especially regarding the crucial aspect of top-notch instrumental sensitivity due to the intensely weak nature of GW signals far from gravitational sources. Given these circumstances and the impossibility of complete isolation of the detectors, the totality of the collected data consists of the addition of the GW signal itself and the undesired contaminating external noise.

In terms of the GW signal in the data, in the case of the coalescence of compact binaries, this event can be divided into three stages: the inspiral, the merger and the ringdown. The main focus will be on the inspiral epoch, where the distance between the stars diminishes with time and correspondingly, the orbital frequency increases, as it is optimally modeled for low-mass, non-spinning compact binary systems with circular orbits using the Post-Newtonian (PN) approximation to GR, ignoring past second-order terms, which greatly simplifies the study and does not affect the results significantly.

In essence, the main objective of this work will revolve around the study of the GW signal (with its noise) of the inspiral coalescence of a compact binary BH system using the matched filtering process, fundamentally based on the computation and plotting of the Signal-To-Noise Ratio (SNR), the horizon distance and the overlap and Fitting Factors (FF) for different signal phases and BH mass cases.

General Methodology

Before the detailed analysis of this process, it should be noted that the nature and presentation of this work are a combination of theoretical concepts of signal processing, complemented by numerical computing shown in full code form in the Julia Programming Language [1], which can be found explicitly at the end of the document.

Beginning with some basic considerations, for convenience purposes, according to the convention commonly used in GR to express mass and length in units of time in the computations, these unit conversions are applied:

$$1\ \mathrm{Gpc} = 10^9 \frac{\mathrm{pc}}{\mathrm{c}}, \tag{1}$$

and

$$1\ \frac{\mathrm{M}_\odot \mathrm{G}}{\mathrm{c}} \approx 4.93\ \mu\mathrm{s}, \tag{2}$$

where 1 pc ("parsec") = $3.0857 \cdot 10^{16}$ m, the gravitational constant $G = 6.674 \cdot 10^{-11} \frac{\mathrm{m}^3}{\mathrm{kg} \cdot \mathrm{s}^2}$, 1 $\mathrm{M}_\odot$ ("solar mass") = $1.98847 \cdot 10^{30}$ kg, and the speed of light in vacuum c = 299792458.0 $\frac{\mathrm{m}}{\mathrm{s}}$.

For the noise computation, due to the simplified scope of this study, a stationary, Gaussian, and zero mean noise is considered with a theoretical noise curve constructed for the Advanced LIGO detectors [2] in the form of a one-sided PSD $S_n(f)$ in the frequency f domain, expressed as follows [3]:

$$S_n(f) = \begin{cases} \infty & \text{if} \quad f < f_c, \\ S_0 \left[x^{-4.14} - 5x^{-2} + \frac{111\left(1 - x^2 + \frac{x^4}{2}\right)}{1 + \frac{x^2}{2}} \right] & \text{if} \quad f \geq f_c, \end{cases} \tag{3}$$

where $x = \frac{f}{f_0}$ with the convenient scaling factor $f_0 = 215$ Hz, $f_c = 10$ Hz is the lower cut-off frequency and $S_0 = 10^{-49}$ Hz^{-1} is the noise curve "amplitude".

A Computation and plotting of the SNR at a distance of 1 Gpc for an equal mass case

Starting with the task of computing and plotting the averaged SNR for the event mentioned above, detected by the Advanced LIGO detectors at a luminosity distance $r = 1$ Gpc for the case of BHs of equal mass ($m_1 = m_2$), with the Fourier transforms of an arbitrary function $x(t)$ following the convention:

$$\tilde{x}(f) = \int_{-\infty}^{\infty} e^{i2\pi ft} x(t) dt, \tag{4}$$

the Fourier transform of the inspiral waveform $\tilde{h}_{\mathrm{INS}}(f)$ can be defined accordingly, using the stationary phase and the PN approximations, yielding [3]:

$$\tilde{h}_{\mathrm{INS}}(f) = \begin{cases} 0 & \text{if} \quad f < f_{\mathrm{ISCO}}, \\ \mathcal{A}_{\mathrm{INS}} f^{-\frac{7}{6}} e^{i\Psi(f)} & \text{if} \quad f \geq f_{\mathrm{ISCO}}, \end{cases} \tag{5}$$

with the phase $\Psi(f)$ and the inspiral "amplitude" $\mathcal{A}_{\mathrm{INS}}$, respectively:

$$\Psi(f) = 2\pi f t_c - \phi_c - \frac{\pi}{4} + \frac{3}{128} \sum_{k=0}^{4} A_k u^{k-5}, \tag{6}$$

$$\mathcal{A}_{\mathrm{INS}} = -\frac{\mathcal{M}^{\frac{5}{6}}}{r} \sqrt{\frac{5\pi}{96}} \pi^{-\frac{7}{6}} \sqrt{F_+^2 (1 + c^2)^2 + F_\times^2 4c^2}, \tag{7}$$

where $M = m_1 + m_2$ is the total mass, $\mu = \frac{m_1 m_2}{M}$ is the reduced mass, $\eta = \frac{\mu}{M}$ is the symmetric mass ratio, $\mathcal{M} = \mu^{\frac{3}{5}} M^{\frac{2}{5}} = M\eta^{\frac{3}{5}}$ is the chirp mass, $f_{\mathrm{ISCO}} = \frac{1}{6^{\frac{3}{2}} \pi M}$ is the frequency of the Innermost Stable Circular Orbit, or ISCO (considering that it occurs at a separation of $6M$), t_c is the coalescence time, ϕ_c is the coalescence instantaneous phase, $c = \cos(t)$, $u = (\pi \mathcal{M} f)^{\frac{1}{3}}$, F_+ and $F_\times$ are the plus and cross antenna pattern functions, respectively, and the coefficients A_k have the following expressions:

$$A_0 = 1, \tag{8}$$

$$A_1 = 0, \tag{9}$$

$$A_2 = \frac{20}{9}\left(\frac{743}{336}+\frac{11}{4}\eta\right)\eta^{-\frac{2}{5}}, \tag{10}$$

$$A_3 = -16\pi\eta^{-\frac{3}{5}}, \tag{11}$$

$$A_4 = 10\left(\frac{3058673}{1016064}+\frac{5429}{1008}\eta+\frac{617}{144}\eta^2\right)\eta^{-\frac{4}{5}}. \tag{12}$$

Following that, defining the "inner product" $(g|h)$ between two arbitrary signals $h(t)$ and $g(t)$ this way:

$$(g|h) \equiv 2\int_0^\infty \frac{\tilde{g}^*(f)\tilde{h}(f)+\tilde{g}(f)\tilde{h}^*(f)}{S_n(f)}df, \tag{13}$$

where "*" represents the conjugate, the expression for the optimal and standard squared SNR (ρ^2) can be obtained [3]:

$$\rho^2 = (h|h) = 4\int_0^\infty \frac{|\tilde{h}(f)|^2}{S_n(f)}df. \tag{14}$$

Finally, introducing (5)-(12) into (14) for the studied case and considering the lower limit f_c and upper limit $f_{\rm ISCO}$ for the pertinent computations, the following expression is acquired:

$$\rho^2 = (h_{\rm INS}|h_{\rm INS}) = 4\int_{f_c}^{f_{\rm ISCO}} \frac{|\mathcal{A}_{\rm INS}f^{-\frac{7}{6}}|^2}{S_n(f)}df \approx 4|\mathcal{A}_{\rm INS}^{\rm rms}|^2\int_{f_c}^{f_{\rm ISCO}}\frac{f^{-\frac{7}{3}}}{S_n(f)}df = 4\left|\frac{\mathcal{M}^{\frac{5}{6}}}{r\pi^{\frac{2}{3}}\sqrt{30}}\right|^2\int_{f_c}^{f_{\rm ISCO}}\frac{f^{-\frac{7}{3}}}{S_n(f)}df, \tag{15}$$

where $\mathcal{A}_{\rm INS}^{\rm rms} = \sqrt{\langle\mathcal{A}_{\rm INS}^2\rangle} = \frac{\mathcal{M}^{\frac{5}{6}}}{r\pi^{\frac{2}{3}}\sqrt{30}}$ is the approximated root-mean-square (rms) version of $\mathcal{A}_{\rm INS}$ for the numerical computations using the ensemble average $\langle(...)\rangle$ [3]. The non-square computed form of (15) for a range of different M values, with all the parameters and unit conversions previously mentioned defined, can be observed in Figure 1, which correctly reproduces the same case portrayed in Figure 1(b) of [3], successfully corroborating the results.

B Computation and plotting of the horizon distance for an SNR = 8 and an equal mass case

Continuing with the task of computing and plotting the horizon distance r for an equivalent event as before with the same detectors, and for the case of BHs of equal mass ($m_1 = m_2$) and an SNR of $\rho = 8$, (15) can simply be reorganized, and, as a result, an expression for r can be obtained in this manner:

$$\rho = 2\frac{\mathcal{M}^{\frac{5}{6}}}{r\pi^{\frac{2}{3}}\sqrt{30}}\sqrt{\int_{f_c}^{f_{\rm ISCO}}\frac{f^{-\frac{7}{3}}}{S_n(f)}df} \Rightarrow r = 2\frac{\mathcal{M}^{\frac{5}{6}}}{\rho\pi^{\frac{2}{3}}\sqrt{30}}\sqrt{\int_{f_c}^{f_{\rm ISCO}}\frac{f^{-\frac{7}{3}}}{S_n(f)}df}. \tag{16}$$

The result of the computation of (16) for the same range of M values as in the previous section can be observed in Figure 2, which maintains the same shape as the result for $r = 1$ Gpc, only being reduced in magnitude by $\rho = 8$.

C Computation of the overlap and Fitting Factors of the signal with a phase at 2 PN with the one at 1.5 PN for different mass cases

The last task of this work requires the computation of the overlap $\mathcal{O}$ and Fitting Factors FF of the signal with a phase of second Post-Newtonian order (2 PN or PN2), considering all coefficients A_k, with the one of 1.5 Post-Newtonian order (1.5 PN or PN1.5), only considering up to the included coefficient A_3, for three different mass cases: $m_1 = m_2 = 1.4\mathrm{M}_\odot$, $m_1 = m_2 = 10\mathrm{M}_\odot$, and $m_1 = 10\mathrm{M}_\odot$, $m_2 = 1.4\mathrm{M}_\odot$.

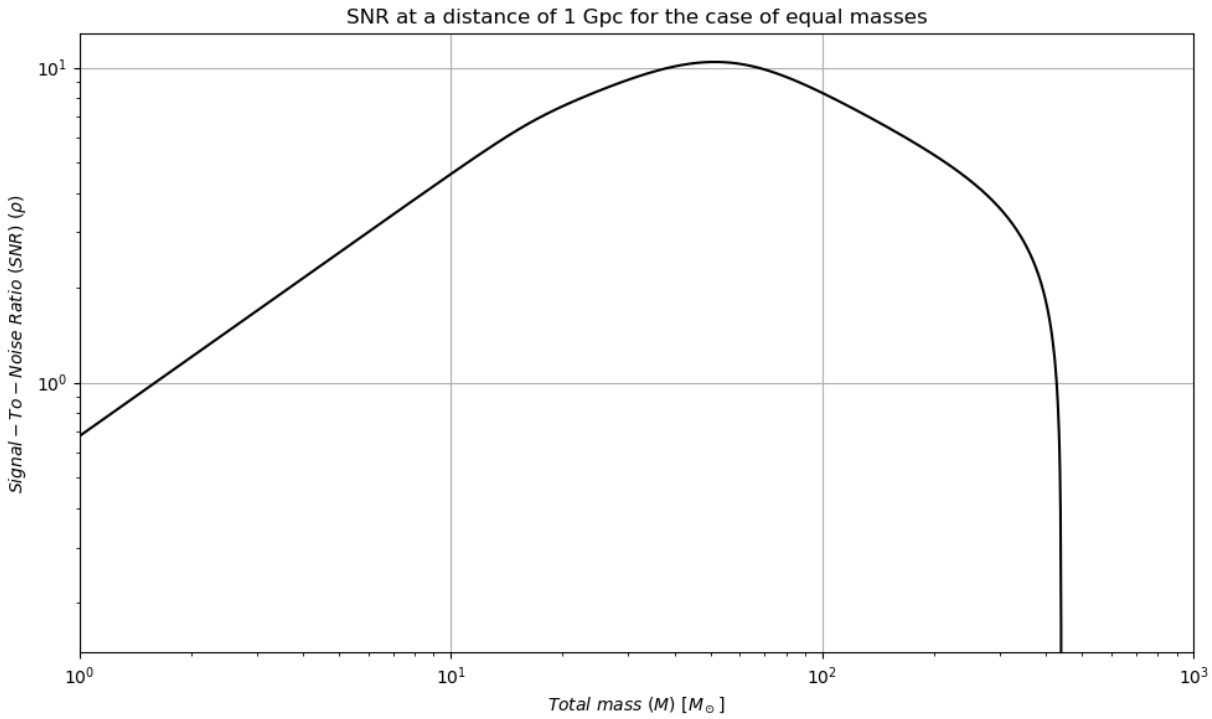


Figure 1: Signal-To-Noise Ratio (SNR) at $r = 1$ Gpc for an equal mass case scenario in binary inspiral coalescence, represented in a range of $M = [1,10^3]$ $M_\odot$, and in common logarithmic scale for easier visualization.

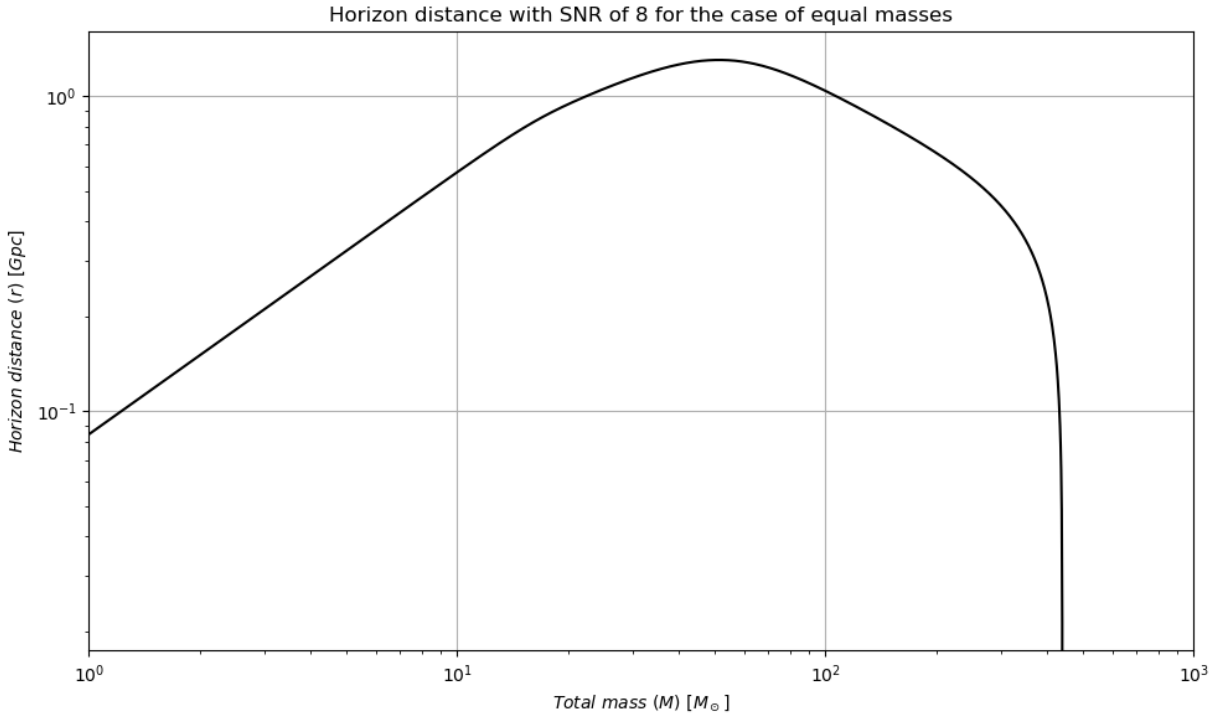


Figure 2: Horizon distance (r) for an equal mass and $\rho = 8$ case scenario in binary inspiral coalescence, represented in a range of $M = [1,10^3]$ $M_\odot$, and in common logarithmic scale for easier visualization.

It is deduced that the overlap $\mathcal{O}$ between two waveforms in the pertinent case can be expressed in the following manner, specifically, using (5), (6), (13), and (15):

$$\mathcal{O}(h_{\rm PN1.5}, h_{\rm PN2}) = \frac{(h_{\rm PN1.5}|h_{\rm PN2})}{\sqrt{(h_{\rm PN1.5}|h_{\rm PN1.5})(h_{\rm PN2}|h_{\rm PN2})}}, \tag{17}$$

with

$$(h_{\rm PN1.5}|h_{\rm PN1.5}) = 4\left|\frac{\mathcal{M}_{\rm PN1.5}^{\frac{5}{6}}}{r\pi^{\frac{2}{3}}\sqrt{30}}\right|^2 \int_{f_c}^{f_{\rm ISCO}^{\rm PN1.5}} \frac{f^{-\frac{7}{3}}}{S_n(f)} df, \tag{18}$$

$$(h_{\rm PN2}|h_{\rm PN2}) = 4\left|\frac{\mathcal{M}_{\rm PN2}^{\frac{5}{6}}}{r\pi^{\frac{2}{3}}\sqrt{30}}\right|^2 \int_{f_c}^{f_{\rm ISCO}^{\rm PN2}} \frac{f^{-\frac{7}{3}}}{S_n(f)} df, \tag{19}$$

$$(h_{\rm PN1.5}|h_{\rm PN2}) = 4\left|\frac{1}{r\pi^{\frac{2}{3}}\sqrt{30}}\right|^2 \mathcal{M}_{\rm PN1.5}^{\frac{5}{6}}\mathcal{M}_{\rm PN2}^{\frac{5}{6}} \int_{f_c}^{min(f_{\rm ISCO}^{\rm PN1.5}, f_{\rm ISCO}^{\rm PN2})} \frac{f^{-\frac{7}{3}}}{S_n(f)} \cos(\Psi_{\rm PN2}(f) - \Psi_{\rm PN1.5}(f)) df. \tag{20}$$

Thus, inserting (18)-(20) in (17) results in:

$$\mathcal{O}(h_{\rm PN1.5}, h_{\rm PN2}) = \frac{1}{\sqrt{\int_{f_c}^{f_{\rm ISCO}^{\rm PN1.5}} \frac{f^{-\frac{7}{3}}}{S_n(f)} df}\sqrt{\int_{f_c}^{f_{\rm ISCO}^{\rm PN2}} \frac{f^{-\frac{7}{3}}}{S_n(f)} df}} \int_{f_c}^{min(f_{\rm ISCO}^{\rm PN1.5}, f_{\rm ISCO}^{\rm PN2})} \frac{f^{-\frac{7}{3}}}{S_n(f)} \cos(\Psi_{\rm PN2}(f) - \Psi_{\rm PN1.5}(f)) df, \tag{21}$$

with

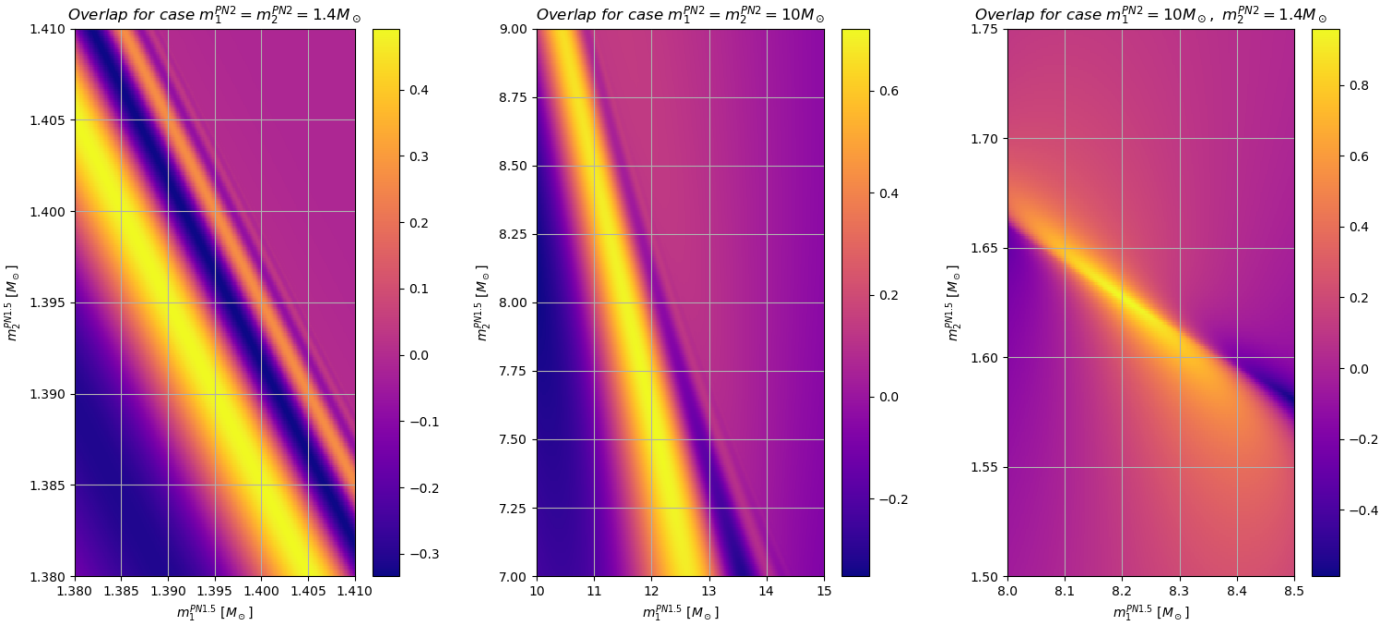
$$\Psi_{\rm PN2}(f) - \Psi_{\rm PN1.5}(f) = \frac{3}{128}\left(\sum_{k=0}^{4}[A_k u^{k-5}]_{\rm PN2} - \sum_{k=0}^{3}[A_k u^{k-5}]_{\rm PN1.5}\right), \tag{22}$$

being the phase difference between the two considered waveforms. The outcome of the computation of (21) with (22) for the three mass cases, with all the previous parameters implemented, can be observed in Table 1.

m_1 [$M_\odot$]	m_2 [$M_\odot$]	$\mathcal{O}$
1.4	1.4	+0.00059898
10	10	+0.13142914
10	1.4	−0.01714072

Table 1: Resulting overlaps $\mathcal{O}$ for the three different mass cases studied.

Subsequently, for convenience and visualization purposes, the overlaps $\mathcal{O}$ have been calculated (using (21) and (22)) with respect to the same different cases of fixed mass values for the signal in 2 PN, for a range of m_1 and m_2 values in 1.5 PN, resulting in the corresponding 2D color plots that can be seen in Figure 3.



(a) $m_1^{\mathrm{PN2}} = m_2^{\mathrm{PN2}} = 1.4\mathrm{M}_\odot$ case. (b) $m_1^{\mathrm{PN2}} = m_2^{\mathrm{PN2}} = 10\mathrm{M}_\odot$ case. (c) $m_1^{\mathrm{PN2}} = 10\mathrm{M}_\odot$; $m_2^{\mathrm{PN2}} = 1.4\mathrm{M}_\odot$ case.

Figure 3: 2D color mesh plots of the overlap $\mathcal{O}$ vs $(m_1^{\mathrm{PN1.5}}, m_2^{\mathrm{PN1.5}})$ for the three different mass cases in a range close to the maximum values of $\mathcal{O}$.

Lastly, using (21) with (22), a simple iterative nested sampling method, and the values of Figure 3, the FFs can be computed as [4]:

$$\mathrm{FF} = \max_{h_{\mathrm{PN1.5}}} \mathcal{O}(h_{\mathrm{PN1.5}}, h_{\mathrm{PN2}}), \tag{23}$$

with respect to the waveform in 1.5 PN, which produces the following results in Table 2:

m_1^{PN2} [$\mathrm{M}_\odot$]	m_2^{PN2} [$\mathrm{M}_\odot$]	$m_1^{\mathrm{PN1.5}}$ [$\mathrm{M}_\odot$]	$m_2^{\mathrm{PN1.5}}$ [$\mathrm{M}_\odot$]	FF
1.4	1.4	1.393	1.393	0.492
10	10	12.004	7.594	0.722
10	1.4	8.209	1.626	0.959

Table 2: Resulting FFs and corresponding masses in 1.5 PN for the three different mass cases studied in 2 PN.

References

[1] Jeff Bezanson, Alan Edelman, Stefan Karpinski, and Viral B. Shah. The julia programming language. https://julialang.org/. Accessed the 02/06/25.

[2] The LIGO Scientific Collaboration: J. Aasi et al. Advanced ligo. *Classical and Quantum Gravity*, 32(7):074001, mar 2015.

[3] Manuel Luna and Alicia M Sintes. Parameter estimation of compact binaries using the inspiral and ringdown waveforms. *Classical and Quantum Gravity*, 23(11):3763, may 2006.

[4] Alicia M. Sintes and Alberto Vecchio. Detection of gravitational waves from inspiraling compact binaries using nonrestricted postNewtonian approximations. In *34th Rencontres de Moriond: Gravitational Waves and Experimental Gravity*, pages 73–78, 1999.

Full code listing

```
# Matched Filtering of a Gravitational Wave (GW) Signal
# Computed in Julia 1.11.3
# author: Lucas Romero Fernández 2025
# Start with "julia Matched_Filtering_of_a_Gravitational_Wave_Signal.jl"

using QuadGK # For the primitives
using LinearAlgebra # For the vector and matrix operations
using PyPlot # For the plots of the results

# Definition of general constants, variables, arrays and functions

parsec = 3.0857e16 # Conversion from 1 parsec (pc) to meters (Units: m)
G = 6.674e-11 # Gravitational constant (Units: m^(3)kg^(-1)s^(-2))
M_sun = 1.98847e30 # Conversion from 1 solar mass to kilograms (Units: kg)
c = 299792458.0 # Speed of light in vaccuum (Units: m/s)
Gpc = 1e9*(parsec/c) # Conversion of gigaparsecs (length) to units of time (pc/c) (Units: s)
m_sun_time = M_sun*(G/(c^3)) # Conversion of solar masses (mass) to units of time (M_sun*G*(c^(-3)) (Units: s)
f_c = 10 # Lower cut-off frequency and lower limit of the SNR integral (Units: Hz)
Masses_array = range(0,stop = 3,length = 2000) .|> x->10^x # Total masses array (Units: Solar masses)
f_ISCO(M) = 1/(M*pi*6^(3/2)) # Mass dependent f_ISCO function (superior limit of the SNR integral) (Units: Hz)
function Sn(f) # Theoretical noise/sensitivity curve (PSD) for Advanced LIGO (Units: Hz^(-1))
    f_0 = 215 # Scaling factor (Units: Hz)
    S_0 = 1e-49 # "Amplitude" (Units: Hz^(-1))
    x = f./f_0
    return ifelse.(f .>= f_c,S_0*(x.^(-4.14) .- 5*x.^(-2) .+ 111*(1 .- x.^2 .+ x.^4/2)./(1 .+ x.^2/2)),Inf)
    end

# Section A: Computation and plotting of the SNR at a distance of 1 Gpc for an equal mass case

# Definition of specific functions

A_rms_INS(chirpM,r) = (chirpM^(5/6))/(r*sqrt(30)*pi^(2/3)) # Mass dependent A^rms_INS function (Units: kg^(5/6)/s)
function SNR(m_1,m_2,r) # Signal-to-Noise Ratio (SNR) function with masses m_1 and m_2 and horizon distance r
    M = (m_1 + m_2) # Total mass (Units: kg)
    mu = (m_1*m_2)/M # Reduced mass (Units: kg)
    eta = mu/M # Symmetric mass ratio (No units)
    chirpM = M^(2/5)*mu^(3/5) # Chirp mass (Units: kg)
    to_integrate(f) = (f^(-7/3))/Sn(f) # Function to integrate
    integral = quadgk(to_integrate,f_c,f_ISCO(M))[1] # Integration computation
    return sqrt(4*A_rms_INS(chirpM,r)^2*integral)
    end

# main_program

SNR_array = [SNR((m/2)*m_sun_time,(m/2)*m_sun_time,Gpc) for m in Masses_array] # r = 1 Gpc

# Plot

```

```
figure(figsize = (10,6))
plot(Masses_array,SNR_array,color = "black")
xscale("log")
yscale("log")
xlim(1,1000)
xlabel(L"Total\ mass\ (M)\ [M_\odot]")
ylabel(L"Signal-To-Noise\ Ratio\ (SNR)\ (\rho)")
grid()
title("SNR at a distance of 1 Gpc for the case of equal masses")
tight_layout()
savefig("Results_Section_A.png")
show()

# Section B: Computation and plotting of the horizon distance for an SNR = 8 and an equal mass case

# Definition of specific functions

function r(m_1,m_2,rho) # Horizon distance (r) function,
    # obtained by rewriting the SNR (rho) function from section A
    M = (m_1+m_2) # Total mass (Units: kg)
    mu = (m_1*m_2)/M # Reduced mass (Units: kg)
    eta = mu/M # Symmetric mass ratio (No units)
    chirpM = M^(2/5)*mu^(3/5) # Chirp mass (Units: kg)
    to_integrate(f) = f^(-7/3)/Sn(f) # Function to integrate
    integral = quadgk(to_integrate,f_c,f_ISCO(M))[1] # Integration computation
    return (((2*chirpM^(5/6))/(rho*sqrt(30)*pi^(2/3)))*sqrt(integral))/Gpc
    end

# main_program

r_array = [r((m/2)*m_sun_time,(m/2)*m_sun_time,8) for m in Masses_array] # SNR = 8

# Plot

figure(figsize = (10,6))
plot(Masses_array,r_array,color = "black")
xscale("log")
yscale("log")
xlim(1,1000)
xlabel(L"Total\ mass\ (M)\ [M_\odot]")
ylabel(L"Horizon\ distance\ (r)\ [Gpc]")
grid()
title("Horizon distance with SNR of 8 for the case of equal masses")
tight_layout()
savefig("Results_Section_B.png")
show()

# Section C: Computation of the overlap and Fitting Factors of the signal with a phase at 2 PN with the one at 1.5
# PN for different mass cases (for 2 PN, all terms have been considered, for 1.5 PN, only up to A_3).
# Mass cases to be studied:
```

```
# - m_1 = m_2 = 1.4 M_sun
# - m_1 = m_2 = 10 M_sun
# - m_1 = 10 M_sun and m_2 = 1.4 M_sun

# Definition of specific arrays and functions

x_array1 = range(1.38,stop = 1.41,length = 250) # For the first case
x_array2 = range(10.,stop = 15.,length = 500) # For the second case
x_array3 = range(8.0,stop = 8.5,length = 250) # For the third case
y_array1 = range(1.38,stop = 1.41,length = 250) # For the first case
y_array2 = range(7.,stop = 9.,length = 250) # For the second case
y_array3 = range(1.5,stop = 1.75,length = 250) # For the third case
function varPhase(m_1,m_2,f,PN) # Variable phase function with masses m_1 and m_2 and frequency f,
# between both phases considered of different Post-Newtonian (PN) orders
    M = m_1 + m_2 # Total mass (Units: kg)
    mu = (m_1*m_2)/M # Reduced mass (Units: kg)
    eta = mu/M # Symmetric mass ratio (No units)
    chirpM = M*eta^(3/5) # Chirp mass (Units: kg)
    u = (pi*chirpM*f)^(1/3) # u term
    A_0 = 1 # Coefficient of u of order 0
    A_1 = 0 # Coefficient of u of order 1
    A_2 = 20/9*(743/336 + (11/4)*eta)*eta^(-2/5) # Coefficient of u of order 2
    A_3 = -16*pi*eta^(-3/5) # Coefficient of u of order 3
    A_4 = 10*(3058673/1016064 + (5429/1008)*eta + (617/144)*eta^2)*eta^(-4/5) # Coefficient of u of order 4
    if PN == 2 # Variable phase for PN = 2 case
        return (3/128)*(A_0/u^5 + A_1/u^4 + A_2/u^3 + A_3/u^2 + A_4/u)
    elseif PN == 1.5 # Variable phase for PN = 1.5 case
        return (3/128)*(A_0/u^5 + A_1/u^4 + A_2/u^3 + A_3/u^2)
    else
        println("Variable phase not specified for this PN")
    end
    end
function overlap(m_1,m_2) # Overlap function with masses m_1 and m_2 with the specifics of this section
    m_1 = m_1*m_sun_time # Conversion of mass units to time units (Units: s)
    m_2 = m_2*m_sun_time # Conversion of mass units to time units (Units: s)
    f_ISCO_sp = f_ISCO(m_1+m_2) # f_ISCO function computation for the specific case of this section (Units: Hz)
    # Numerator computation
    to_integrate_top(f) = (cos(varPhase(m_1,m_2,f,2) - varPhase(m_1,m_2,f,1.5)))*(f^(-7/3)/Sn(f)) # I.Function
    numerator = quadgk(to_integrate_top,f_c,min(f_ISCO_sp,f_ISCO_sp))[1] # Integration computation
    # Denominator computation
    to_integrate_bottom(f) = f^(-7/3)/Sn(f) # Function to integrate
    integral_bottom1 = quadgk(to_integrate_bottom,f_c,f_ISCO_sp)[1] # Integrations computations
    integral_bottom2 = quadgk(to_integrate_bottom,f_c,f_ISCO_sp)[1] # Integrations computations
    denominator = sqrt(integral_bottom1*integral_bottom2)
    return numerator/denominator
    end
function meshgrid(x_in_array,y_in_array) # Mesh grid function for the 2D sampling,
# with array x_in_array and array y_in_array
    n_x = length(x_in_array) # Number of points in direction x
    n_y = length(y_in_array) # Number of points in direction y
```

```
    x_out_array = zeros(n_y,n_x) # Construction of resulting array in direction x
    y_out_array = zeros(n_y,n_x) # Construction of resulting array in direction y
    for j_x = 1:n_x
        for i_x = 1:n_y
            x_out_array[i_x,j_x] = x_in_array[j_x] # Mesh grid
            y_out_array[i_x,j_x] = y_in_array[i_x] # Mesh grid
            end
        end
    return (x_array = x_out_array,y_array = y_out_array)
    end
function overlap_plots(m_1b,m_2b,m_1a,m_2a) # Overlaps for the plots function,
# with masses m_1 and m_2 and 2D sampling for both PNs.
    m_1a = m_1a*m_sun_time # Conversion of mass units to time units (Units: s)
    m_2a = m_2a*m_sun_time # Conversion of mass units to time units (Units: s)
    m_1b = m_1b*m_sun_time # Conversion of mass units to time units (Units: s)
    m_2b = m_2b*m_sun_time # Conversion of mass units to time units (Units: s)
    f_ISCO1 = f_ISCO(m_1a + m_2a) # f_ISCO function computation for the specific cases of this section (Units: Hz)
    f_ISCO2 = f_ISCO(m_1b + m_2b) # f_ISCO function computation for the specific cases of this section (Units: Hz)
    # Numerator computation
    to_integrate_top(f) = (cos(varPhase(m_1a,m_2a,f,2) - varPhase(m_1b,m_2b,f,1.5)))*(f^(-7/3)/Sn(f)) # I.Function
    numerator = quadgk(to_integrate_top,f_c,min(f_ISCO1,f_ISCO2))[1] # Integration computation
    # Denominator computation
    to_integrate_bottom(f) = f^(-7/3)/Sn(f) # Function to integrate
    integral_bottom1 = quadgk(to_integrate_bottom,f_c,f_ISCO1)[1] # Integrations computations
    integral_bottom2 = quadgk(to_integrate_bottom,f_c,f_ISCO2)[1] # Integrations computations
    denominator = sqrt(integral_bottom1*integral_bottom2)
    return numerator/denominator
    end

# main_program

# Computation of the overlap for the three cases already indicated
println("Overlap for the m_1 = m_2 = 1.4 M_sun case: ",overlap(1.4,1.4))
println("Overlap for the m_1 = m_2 = 10 M_sun case: ",overlap(10,10))
println("Overlap for the m_1 = 10 M_sun and m_2 = 1.4 M_sun case: ",overlap(10,1.4))
# Final nested sampling for the different cases overlaps
X_array1,Y_array1 = meshgrid(x_array1,y_array1) # For the first case
overlap_array1 = overlap_plots.(X_array1,Y_array1,1.4,1.4) # For the first case
X_array2, Y_array2 = meshgrid(x_array2,y_array2) # For the second case
overlap_array2 = overlap_plots.(X_array2,Y_array2,10,10) # For the second case
X_array3, Y_array3 = meshgrid(x_array3,y_array3) # For the third case
overlap_array3 = overlap_plots.(X_array3,Y_array3,10,1.4) # For the third case

# Plots

# Overlap results for the first case
figure(figsize = (5,7))
pcolormesh(x_array1,y_array1,overlap_array1,cmap =:plasma)
xlabel(L"m^{PN1.5}_1\ [M_\odot]")
ylabel(L"m^{PN1.5}_2\ [M_\odot]")
```

```
xlim(minimum(x_array1),maximum(x_array1))
ylim(minimum(y_array1),maximum(y_array1))
grid()
colorbar()
title(L"Overlap\ for\ case\ m^{PN2}_1 = m^{PN2}_2 = 1.4 M_\odot")
tight_layout()
savefig("Results_Section_C_1.png")
show()
# Overlap results for the second case
figure(figsize = (5,7))
pcolormesh(x_array2,y_array2,overlap_array2,cmap =:plasma)
xlabel(L"m^{PN1.5}_1\ [M_\odot]")
ylabel(L"m^{PN1.5}_2\ [M_\odot]")
xlim(minimum(x_array2),maximum(x_array2))
ylim(minimum(y_array2),maximum(y_array2))
grid()
colorbar()
title(L"Overlap\ for\ case\ m^{PN2}_1 = m^{PN2}_2 = 10 M_\odot")
tight_layout()
savefig("Results_Section_C_2.png")
show()
# Overlap results for the third case
figure(figsize = (5,7))
pcolormesh(x_array3,y_array3,overlap_array3,cmap =:plasma)
xlabel(L"m^{PN1.5}_1\ [M_\odot]")
ylabel(L"m^{PN1.5}_2\ [M_\odot]")
xlim(minimum(x_array3),maximum(x_array3))
ylim(minimum(y_array3),maximum(y_array3))
grid()
colorbar()
title(L"Overlap\ for\ case\ m^{PN2}_1 = 10 M_\odot,\ m^{PN2}_2 = 1.4 M_\odot")
tight_layout()
savefig("Results_Section_C_3.png")
show()

# Computation of the searched Fitting Factors as the maximum values in the graphs

# FFs results for the first case
FF1 = maximum(overlap_array1) # Maximum value in the graph
index_max1 = findfirst(x_array1->x_array1 == maximum(overlap_array1),overlap_array1) # Index of the grid with FF1
m_1_index1 = x_array1[index_max1[2]] # Mass m_1 for 1.5 PN with that index
m_2_index1 = y_array1[index_max1[1]] # Mass m_2 for 1.5 PN with that index
println("For the 2 PN, m_1 = m_2 = 1.4 M_sun case: ")
println("Fitting Factor (FF): ",FF1," with masses for 1.5 PN m_1 = ",m_1_index1,"; m_2 = ",m_2_index1)
# FFs results for the second case
FF2 = maximum(overlap_array2) # Maximum value in the graph
index_max2 = findfirst(x_array2->x_array2 == maximum(overlap_array2),overlap_array2) # Index of the grid with FF2
m_1_index2 = x_array2[index_max2[2]] # Mass m_1 for 1.5 PN with that index
m_2_index2 = y_array2[index_max2[1]] # Mass m_2 for 1.5 PN with that index
println("For the 2 PN, m_1 = m_2 = 10 M_sun case: ")
```

```
println("Fitting Factor (FF): ",FF2," with masses for 1.5 PN m_1 = ",m_1_index2,"; m_2 = ",m_2_index2)
# FFs result for the third case
FF3 = maximum(overlap_array3) # Maximum value in the graph
index_max3 = findfirst(x_array3->x_array3 == maximum(overlap_array3),overlap_array3) # Index of the grid with FF3
m_1_index3 = x_array3[index_max3[2]] # Mass m_1 for 1.5 PN with that index
m_2_index3 = y_array3[index_max3[1]] # Mass m_2 for 1.5 PN with that index
println("For the 2 PN, m_1 = 10 M_sun and m_2 = 1.4 M_sun case: ")
println("Fitting Factor (FF): ",FF3," with masses for 1.5 PN m_1 = ",m_1_index3,"; m_2 = ",m_2_index3)
```