

Simulation of Linearized Relativistic Spatial Phenomena and Weak-field Approximated Gravitational Waves

Nishanth Shubhakar Mandala, Kevin Bojia Zhang

New York University

nishanth.mandala@nyu.edu, zhangkevin@nyu.edu

February 26, 2025

Contents

1	Introduction	4
2	Theoretical Background	4
3	Governing Equations	6
3.1	The Schwarzschild Single Star Transformations	6
3.2	The Radiation Reaction	7
3.3	Gravitational Waves	7
4	Numerical Method	8
4.0.1	Gravitational Acceleration	9
4.0.2	Damping (Radiation Reaction)	9
4.0.3	Total Acceleration	9
4.1	Symplectic Euler	9
5	Validation	10
5.1	Conservation of Total Energy	10
5.2	Conservation of Total Angular Momentum	12
6	Implementation and Code Considerations	15
6.1	N-Body Setup for Binary Stars and Planet	15
6.1.1	1. Simulation Setup	15
6.1.2	2. N-Body Initialization	15
6.1.3	3. Preallocation for Data Storage	16
6.1.4	4. Main Integration Loop (Symplectic Euler Method)	17
6.2	Schwarzschild Spatial Stretching	19
6.3	Simulating the Inspiral using a Drag Force	22
6.4	Gravitational Waves and TT Gauge Transformation	24
6.5	Discussion of Key Considerations	29
7	Results and Discussion	29
7.1	Parametric Studies: Varying the Star Separation	30
7.2	Comparison of With vs. Without the Planet	31
7.3	Interpretation and Limitations	31
7.4	Figures and Animations	31

8	Conclusions	33
9	Appendix	34
9.1	Solving the Schwarzschild Solution	34
9.1.1	Realizing Radial Stretching from the Schwarzschild Solution	34
9.1.2	Choice of Compression Function	35
9.2	Solving for Drag as a Post-Newtonian Correction	36
9.2.1	Expressing the Inspiral as a Drag Acceleration	37
9.3	Gravitational Wave Coordinate Transformations	38
9.3.1	The Minkowski Metric Tensor and TT Gauge	38
9.3.2	Wave Coordinate Transformations	41
9.4	Pattern Functions	44
9.4.1	Pattern Functions	44
9.4.2	Time Dependent Functions h_+ and h_x	44

1. Introduction

This project explores the differences between Newton's and Einstein's views of gravity by simulating how mass warps spacetime and produces gravitational waves.

The Newtonian view is that gravity is an instantaneous force acting at a distance, whereas the Relativistic perspective views gravity as emerging from the curvature of spacetime. This project is particularly interesting because it allows us to explore these differences, providing insights that are otherwise inaccessible through purely analytical methods.

This simulation serves several purposes:

1. It simulates the abstract concepts of general relativity, helping us visualize how spacetime is warped by massive objects and how gravitational waves propagate.
2. Understanding and Feasibility: In order to model the complex Einstein field equations, we use several approximations as dictated by scholarly literature.
3. Analysis: By implementing the simulation in MATLAB, we can easily modify variables to experiment with different scenarios.

2. Theoretical Background

We begin our efforts by first trying our hand at the simplest possible case, modeling the curvature of space around a single massive object, a star in this case.

The first description of this curvature comes from Einstein's field equations, a set of nonlinear partial differential equations. Needless to say, exact solutions to Einstein's field equations are either impossible or incredibly difficult to obtain for most real-life scenarios. The exact solutions that do exist typically describe highly idealized or simplified conditions, rather than the complexity of real-world astrophysical environments. [16]

In the context of space and its properties, as well as the simulation, Einstein's theory of general relativity describes a universe in which distances stretch in the presence of massive objects [15].

The solution used for simulation of the single star, the Schwarzschild Metric, falls under the same category, and the results of it should not be taken authoritatively, but as one method of many to visualize the abstractions of general relativity.

The Schwarzschild Metric is a solution to Einstein's field equations in general relativity [4] that describes the spacetime geometry around a spherically symmetric, non-rotating, uncharged massive object. It is an exact vacuum solution, meaning it describes empty space around such an object.

This solution provides a relationship between the proper distance (the actual physical distance in curved spacetime) and the radial coordinate distance (the distance as measured by an observer at infinity), giving insight into how spacetime is stretched by the gravitational field of the massive object, taking note that this “stretching” increases the closer the radial coordinate is to the star.

The next goalpost to introduce complexity is to visualize the same stretching in a dynamic environment and examine the effects of it on other celestial bodies. Our goal now is to attempt to simulate gravitational waves.

In many existing simulations of general relativity, the effort to approximate it for numerical solving usually results in what are known as post-Newtonian corrections [17]. As the name suggests, these corrections add on relativistic treatment to existing Newtonian models. Our first sight should be towards acknowledging that, according to general relativity, binary systems are unstable and tend to radiate their orbital energy in the form of gravitational waves [14]. The second sight being how to represent these emitted waves in our 3D grid.

When we examine what actually dictates the generation of these waves, literature points us in the direction of the ”mass quadrupole moment”. It is not necessary to know exactly what this quantification is, except that it is a measure of deviations from spherical symmetry, and the change of it results in the generation of gravitational waves [5].

A quick surf through more literature finds an approximated linearized form of this, a metric known as the ”transverse-traceless (TT) gauge” [8]. Instead of directly evaluating this metric as before, we need a little more context.

The passing of gravitational waves through space is evaluated as a time dependent perturbation in flat space. Flat space is dictated by yet another metric, known as the ”Minkowski metric” [7]. The combination of both yields an elegant solution for the coordinate transformations for these perturbations.

In this writeup, we examine physical distortions, varying initial conditions, and real time energy quantifications in these systems.¹

¹Please note that the explanations and derivations provided in the Appendix are pulled from a collection of academic literature and do not necessarily represent novelty or rigor, but a representation optimized and simplified for the sake of simulation.

3. Governing Equations

3.1. The Schwarzschild Single Star Transformations

From the Schwarzschild solution, we derive the following expression (1) for the proper distance corresponding to a radial coordinate (see Appendix 9.1.1):

$$R_{\text{stretched}} = R + R_s \log \left(\frac{\sqrt{R} + \sqrt{R - R_s}}{\sqrt{R_s}} \right) \quad (1)$$

Where:

- $R_{\text{stretched}}$ is the proper distance, which measures the actual physical (new) distance from the center due to gravitational warping.
- R is the radial coordinate in the Schwarzschild metric, which measures the Euclidean (original) distance from the center of the mass.
- R_s is the Schwarzschild radius, given by: $R_s = \frac{2GM}{c^2}$.
- G is the universal gravitational constant, approximately $6.67430 \times 10^{-11} \text{ m}^3 \text{ kg}^{-1} \text{ s}^{-2}$.
- M is the mass of the central object (such as a star or black hole) generating the gravitational field.
- c is the speed of light in a vacuum, approximately $3 \times 10^8 \text{ m/s}$.

We can now examine how these radial distances become increasingly stretched as they approach the star (up to the Schwarzschild radius). This is an interesting result, and we must choose an appropriate method to rearrange our 3D grid points so that they reflect this behavior.

One method involves choosing a suitable compression function for our grid points; the more the distances are stretched, the denser the points will appear.

We approach this method using an analogy from a 2D perspective rather than a 3D one. Imagine a dot-gridded latex sheet, level and elevated. Now, introduce a heavy object at the center of the sheet so that the latex is stretched downward (a three-dimensional warp in our two-dimensional world). When we observe the sheet from above, we notice that the dots in the vicinity of the heavy object appear more densely packed. We apply similar intuition for a 3D space being warped into its adjacent dimension. (See Appendix 9.1.2 for the chosen compression function.)

3.2. The Radiation Reaction

As Peters discusses in [14], in relativity, binary systems lose orbital energy over time. We introduce this loss of energy, also known as the Radiation Reaction, in our Newtonian Model with a drag force on the binary stars (2). (See Appendix 9.2)

$$F_{\text{drag}} = -D v_{\text{rel}} \quad (2)$$

Where,

- $D = \frac{32}{5} \frac{G^3}{c^5} \frac{\mu^2 M_{\text{tot}}^2}{r^4}$.
- $v_{\text{rel}} = \sqrt{\frac{GM_{\text{tot}}}{r}}$
- G is the gravitational constant, approximately $6.67430 \times 10^{-11} \text{ m}^3 \text{ kg}^{-1} \text{ s}^{-2}$.
- c is the speed of light in a vacuum, approximately $3 \times 10^8 \text{ m/s}$.
- $\mu = \frac{m_1 m_2}{m_1 + m_2}$ is the reduced mass of the two binary stars with masses m_1 and m_2 .
- $M_{\text{tot}} = m_1 + m_2$ is the total mass of the system.
- r is the separation distance between the two bodies.

3.3. Gravitational Waves

From the solution of the combined effects of the Minkowski Metric and the TT Gauge, we obtain time-dependent coordinate transformations for each 3D grid point (See Appendix 9.3.1).

$$X_{\text{new}} = \left(X + \frac{1}{2} ((P(\theta_1)h_{+1} + P(\theta_2)h_{+2}) X + (P(\theta_1)h_{\times 1} + P(\theta_2)h_{\times 2}) Y) \right), \quad (3)$$

$$Y_{\text{new}} = \left(Y - \frac{1}{2} ((P(\theta_1)h_{+1} + P(\theta_2)h_{+2}) Y - (P(\theta_1)h_{\times 1} + P(\theta_2)h_{\times 2}) X) \right), \quad (4)$$

$$Z_{\text{new}} = (Z). \quad (5)$$

Where,

- $X_{\text{new}}, Y_{\text{new}}, Z_{\text{new}}$ represent the new transformed coordinates,
- X, Y, Z represent the original coordinates,

-
- $h_{+1} = A \cos(2\omega t_{\text{ret}1})$ represents the plus polarization produced by the first binary star (See Appendix 9.4.2),
 - $h_{+2} = A \cos(2\omega t_{\text{ret}2})$ represents the plus polarization produced by the second binary star (See Appendix 9.4.2),
 - $h_{\times 1} = A \sin(2\omega t_{\text{ret}1})$.. represents the cross polarization produced by the first binary star (See Appendix 9.4.2)
 - $h_{\times 2} = A \sin(2\omega t_{\text{ret}2})$.. represents the cross polarization produced by the second binary star (See Appendix 9.4.2)
 - ω represents the orbital frequency (See Appendix 9.4.2)
 - t_{ret} = is the retarded time (See Appendix 9.4.2)
 - $P(\theta_1)$ = is the pattern function produced by the first binary star (See Appendix 9.4.1)
 - $P(\theta_2)$ = is the pattern function produced by the second binary star (See Appendix 9.4.1)

In the case of a planet under the influence of these gravitational waves, we apply the same wave coordinate transformations to its position.

4. Numerical Method

We are solving the equation for three bodies.

- **Star 1** at position $\mathbf{r}_1(t)$ with velocity $\mathbf{v}_1(t)$
- **Star 2** at position $\mathbf{r}_2(t)$ with velocity $\mathbf{v}_2(t)$
- **A planet (smaller mass)** at position $\mathbf{r}_3(t)$ with velocity $\mathbf{v}_3(t)$

The two stars form an inspiraling binary, meaning they lose orbital energy due to a damping term that approximates gravitational-wave emission. The planet is treated as a test particle that only experiences Newtonian gravity from the two stars (and does not have its own damping).

4.0.1 Gravitational Acceleration

Each body experiences gravitational attraction from the others. In our setup, we take $G = 1$. For example, Star 1 (mass m_1) is pulled by Star 2 (mass m_2) and the planet (mass m_3):

$$\mathbf{a}_1^{(\text{gravity})} = -m_2 \frac{(\mathbf{r}_1 - \mathbf{r}_2)}{\|\mathbf{r}_1 - \mathbf{r}_2\|^3} - m_3 \frac{(\mathbf{r}_1 - \mathbf{r}_3)}{\|\mathbf{r}_1 - \mathbf{r}_3\|^3}.$$

Similar expressions hold for Star 2 and for the planet.

4.0.2 Damping (Radiation Reaction)

Only the two stars have an additional “damping” or “drag” term to model the inspiral from gravitational-wave emission. For Star 1:

$$\mathbf{a}_1^{(\text{damping})} = -\frac{D}{m_1} (\mathbf{v}_1 - \mathbf{v}_{\text{CM}}),$$

where D is a damping coefficient that depends on the distance between Star 1 and Star 2, and \mathbf{v}_{CM} is the center-of-mass velocity of the binary. A similar term applies to Star 2.

4.0.3 Total Acceleration

The total acceleration on Star 1 is:

$$\mathbf{a}_1 = \mathbf{a}_1^{(\text{gravity})} + \mathbf{a}_1^{(\text{damping})},$$

and similarly for Star 2. The planet’s total acceleration includes only gravity from both stars.

4.1. Symplectic Euler

We discretize time into small steps Δt , where $t^n = n \Delta t$. Then we define:

$$\mathbf{r}_i^n = \mathbf{r}_i(t^n), \quad \mathbf{v}_i^n = \mathbf{v}_i(t^n).$$

We use the Symplectic Euler method to update t^n to t^{n+1} :

1. Compute accelerations \mathbf{a}_i^n using positions and velocities at time t^n . That is,

$$\mathbf{a}_i^n = \mathbf{a}_i(\mathbf{r}_1^n, \mathbf{r}_2^n, \mathbf{r}_3^n, \mathbf{v}_1^n, \mathbf{v}_2^n).$$

2. Update velocity:

$$\mathbf{v}_i^{n+1} = \mathbf{v}_i^n + \mathbf{a}_i^n \Delta t.$$

3. Update position using the new velocity:

$$\mathbf{r}_i^{n+1} = \mathbf{r}_i^n + \mathbf{v}_i^{n+1} \Delta t.$$

This "velocity-first, then position" update is unique to the Symplectic Euler scheme, which is better at preserving orbital energy than the standard Euler methods. These steps are repeated until we reach the desired final time.

5. Validation

In this section, we verify that our simulation code produces physically consistent results. We focus on two key conservation laws in an N -body system: total energy and total angular momentum [3].

5.1. Conservation of Total Energy

Our numerical method, being symplectic Euler, is not fully energy-conserving. To prove the accuracy of total orbital energy, we first removed the post-Newtonian corrections of the Radiation Reaction, as it causes the binary's orbital energy to decrease over time. When the Radiation Reaction is removed, the total energy (kinetic + potential of all bodies) in a purely Newtonian system should be conserved exactly by the true solution.

- **Without radiation reaction:** We comment out the damping terms so the system is pure Newtonian. In that scenario, the total energy remains nearly constant. (Figure 1) displays the binary orbital energy, planet energy, and total system energy. It can be observed that when the binary orbital energy spikes, the planet energy troughs, and vice versa. The total system energy, however, decreases within marginal error, attributed to errors with the symplectic Euler estimation.

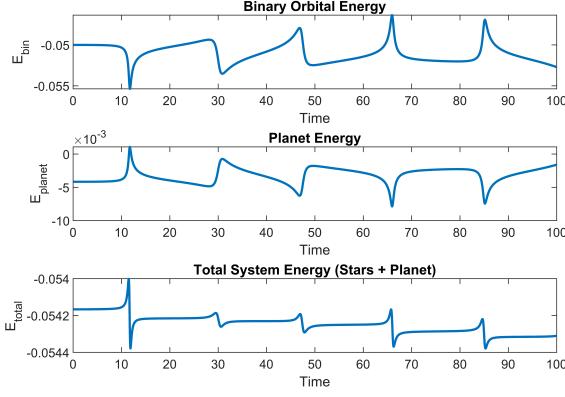


Figure 1: Newtonian orbital energy without post-Newtonian corrections

- **With Radiation Reaction + Gravitational Waves:** We uncomment the damping terms. The binary’s orbital energy decreases over time, but for the entire system, the net energy also declines because we are effectively including the energy carried away by gravitational waves. (Figure 2) shows energy for this case. We see a steady decay as the radiation reaction decreases orbital energy. We expect that the planet’s energy should increase by the amount that was lost with the emission of the gravitational waves. However, the planet’s energy approximately the same, suggesting that gravitational wave emission primarily affects the binary while having a negligible impact on the planet. This makes intuitive sense as the energy is radiated away in all directions and the planet may not be able to capture all of it.

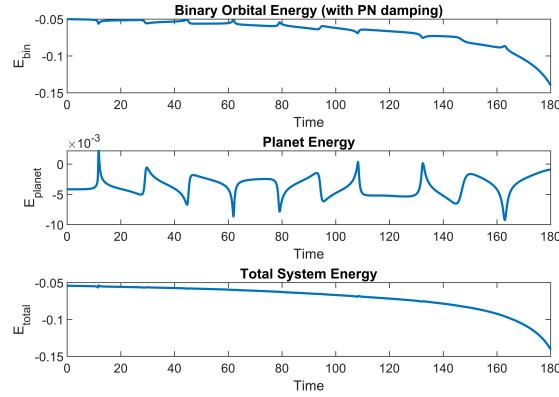


Figure 2: Newtonian orbital energy with post-Newtonian corrections

Thus, conservation of total energy is validated for both scenarios. Without the Radiation

Reaction, the total energy remains near constant, which is expected. With the Radiation Reaction, the energy decays while the planet’s energy remains unaffected, confirming that radiation losses are localized to the binary.

5.2. Conservation of Total Angular Momentum

To check conservation of angular momentum, we tracked the total angular momentum of the system over time,

$$\mathbf{L}_{\text{tot}}(t) = (\mathbf{r}_1 \times m_1 \mathbf{v}_1) + (\mathbf{r}_2 \times m_2 \mathbf{v}_2) + (\mathbf{r}_p \times m_p \mathbf{v}_p).$$

In a purely Newtonian N -body problem with no external torques, \mathbf{L}_{tot} should remain constant (apart from small integration errors).

However the post-Newtonian radiation-reaction term removes orbital energy from the binary. Real gravitational waves carry away both energy and angular momentum, but we only model the energy loss via a velocity-proportional drag. Consequently, we see two notable effects:

1. **Net Decrease in L_{tot} :** Because the code’s Radiation Reaction term indirectly subtracts orbital angular momentum from the stars without simultaneously adding that angular momentum to an outgoing Gravitational-Wave field, L_{tot} decreases steadily. Physically, the system is radiating angular momentum in the form of gravitational waves, but we do not track that radiation explicitly.
2. **Spikes or Jumps:** We observe large spikes in L_{bin} and L_{tot} when the planet passes very close to a star, inducing large torques in a single timestep. These near encounters and the first-order integration scheme can cause instantaneous overshoots in velocity updates, leading to abrupt changes in the measured angular momentum.

To better understand these effects, we first analyze the system without the planet to isolate the binary’s behavior.

(Figure 3) shows the total angular momentum over time in a Newtonian simulation without post-Newtonian corrections. As expected, the angular momentum remains nearly constant, with only small numerical fluctuations due to integration errors.

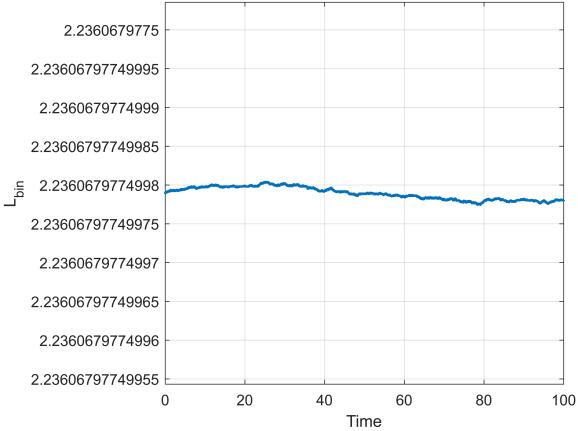


Figure 3: Total angular momentum over time in a Newtonian simulation without post-Newtonian corrections.

In contrast, (Figure 4) shows the total angular momentum with post-Newtonian corrections applied. Here, we observe a steady decrease in angular momentum, reflecting the emission of gravitational waves, which radiate momentum away from the system. This loss is not explicitly tracked in our simulation but is inferred from the decline in L_{bin} .

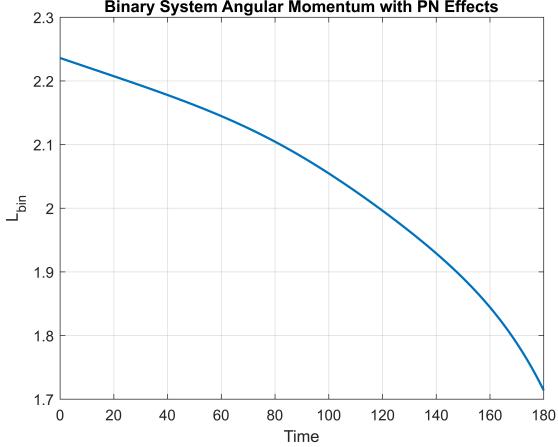


Figure 4: Total angular momentum over time with post-Newtonian corrections applied

After establishing the behavior of the binary system alone, we reintroduce the planet to observe its influence on angular momentum conservation. The planetary perturbations introduce additional angular momentum exchange between the bodies, leading to more complex fluctuations in L_{tot} .

(Figure 5) shows angular momentum of the system without post-Newtonian corrections.

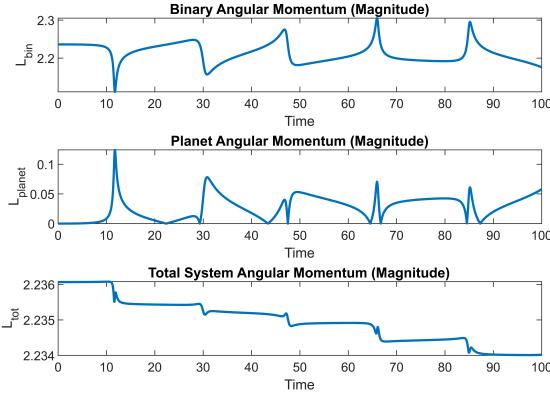


Figure 5: Newtonian angular momentum without post-Newtonian corrections

In this purely Newtonian case, total angular momentum remains nearly constant over time, with only minor fluctuations due to numerical integration errors. These small variations are expected from the first-order symplectic Euler method.

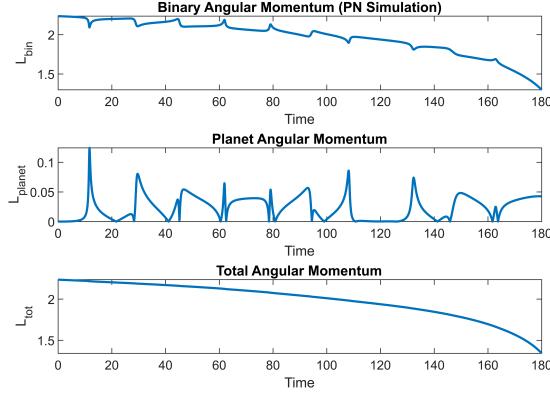


Figure 6: Newtonian angular momentum with post-Newtonian corrections

In contrast, (Figure 6) shows the angular momentum loss when the post-Newtonian damping term is included. The total system angular momentum steadily declines over time, reflecting the angular momentum carried away by gravitational waves. Additionally, occasional abrupt changes correspond to close encounters, where numerical discrepancies from integration cause sudden angular momentum shifts between the planet and the binary star system.

Thus, while the purely Newtonian part of the code conserves angular momentum up to small numerical drift, the post-Newtonian drag unavoidably violates angular momentum

conservation by design, since we neglect the wave-carried angular momentum.

6. Implementation and Code Considerations

This section details the key implementation steps and computational choices made for our simulations. The project comprises two major parts: (1) simulating spatial distortion due to a Schwarzschild metric and (2) modeling gravitational waves in an inspiraling binary system using the transverse-traceless (TT) gauge.

6.1. N-Body Setup for Binary Stars and Planet

Before incorporating the full complexities of gravitational wave effects and spatial distortions, we begin with a standard N-body simulation that models the orbits of a binary star system and a test planet.

6.1.1 1. Simulation Setup

This chunk defines the simulation parameters such as the time step, total duration, and creates the time vector.

Listing 1: Simulation Setup

```
%% ===== Step 0: Simulation Setup =====
dt      = 0.01;          % time step
T_total = 100;           % total simulation time
time    = 0:dt:T_total; % time vector from 0 to T_total
numSteps = length(time); % number of simulation steps
```

Explanation: Here, `dt` sets the simulation resolution, while `T_total` is the total duration. The vector `time` and the variable `numSteps` determine the number of iterations for our integration loop.

6.1.2 2. N-Body Initialization

This section initializes the physical parameters for the two stars and the test planet. The stars are placed along the x-axis with their center of mass at the origin, and the velocities are set for a circular orbit.

Listing 2: N-Body Initialization

```
%% ===== Step 1: N-Body Setup (Binary Stars + Planet) =====
% Define masses
```

```

m1 = 1;
m2 = 1;
M_tot = m1 + m2;      % Total mass
mu     = (m1*m2) / M_tot; % Reduced mass

% Set initial separation
r_sep = 10;
% Place stars along x-axis with center-of-mass at origin:
x1 = +r_sep/2; y1 = 0; z1 = 0;    % Star 1 (right)
x2 = -r_sep/2; y2 = 0; z2 = 0;    % Star 2 (left)

% Compute relative orbital speed for a circular orbit
v_rel = sqrt(M_tot / r_sep);
% Set initial velocities (orbit in the xy-plane)
v1x = 0;   v1y = (m2/M_tot)*v_rel;   v1z = 0;
v2x = 0;   v2y = -(m1/M_tot)*v_rel;   v2z = 0;

% Define test planet properties (small mass)
planetMass = 0.01;
xp = 1;   yp = 0;   zp = 0;      % Initial planet position
vxp = 0;  vyp = 0;  vzp = 0;    % Initial planet velocity

```

Explanation: - The masses m_1 and m_2 are defined along with the total mass and reduced mass. - The stars are symmetrically placed about the origin and given velocities that yield a circular orbit in the xy -plane. - A test planet is initialized with a small mass at a specified position.

6.1.3 3. Preallocation for Data Storage

Arrays are preallocated to store the positions of the stars and the planet at each time step for later visualization and analysis.

Listing 3: Data Storage Preallocation

```

%% ===== Step 2: Arrays for Storing Data =====
x1save = zeros(1, numSteps);  y1save = zeros(1, numSteps);  z1save = zeros(1,
x2save = zeros(1, numSteps);  y2save = zeros(1, numSteps);  z2save = zeros(1,
xpSave = zeros(1, numSteps);  ypSave = zeros(1, numSteps);  zpSave = zeros(1,

```

Explanation: These arrays will be populated during the simulation loop to record the trajectories of the binary stars and the planet.

6.1.4 4. Main Integration Loop (Symplectic Euler Method)

The simulation is evolved using a symplectic Euler method. The loop is broken down into sections: computing forces, updating velocities and positions, and saving the data.

a. Compute Forces and Accelerations For each time step, the gravitational interactions among the bodies are computed.

Listing 4: Force Computation for Stars

```
for k = 1:numSteps
    % Compute vector and distance between the two stars
    r12_vec = [x1 - x2, y1 - y2, z1 - z2];
    r12 = norm(r12_vec) + eps; % Avoid division by zero

    % Acceleration on star 1 from star 2:
    a1_star2 = - m2 * r12_vec / r12^3;
    % Acceleration on star 1 from the planet:
    r1p_vec = [x1 - xp, y1 - yp, z1 - zp];
    r1p = norm(r1p_vec) + eps;
    a1_planet = - planetMass * r1p_vec / r1p^3;
    % Total acceleration for star 1:
    a1 = a1_star2 + a1_planet;

    % Acceleration on star 2 from star 1:
    a2_star1 = - m1 * (-r12_vec) / r12^3;
    % Acceleration on star 2 from the planet:
    r2p_vec = [x2 - xp, y2 - yp, z2 - zp];
    r2p = norm(r2p_vec) + eps;
    a2_planet = - planetMass * r2p_vec / r2p^3;
    % Total acceleration for star 2:
    a2 = a2_star1 + a2_planet;
```

Explanation: - `r12_vec` and `r12` compute the separation between the stars. - The gravitational accelerations due to mutual interactions and the influence of the planet are computed separately and then summed for each star.

b. Update Velocities and Positions for Stars The velocities and positions are updated using the symplectic Euler method.

Listing 5: Updating Star Velocities and Positions

```
% Update star 1 velocities and position:  
v1x = v1x + a1(1)*dt;  
v1y = v1y + a1(2)*dt;  
v1z = v1z + a1(3)*dt;  
x1 = x1 + v1x*dt;  
y1 = y1 + v1y*dt;  
z1 = z1 + v1z*dt;  
  
% Update star 2 velocities and position:  
v2x = v2x + a2(1)*dt;  
v2y = v2y + a2(2)*dt;  
v2z = v2z + a2(3)*dt;  
x2 = x2 + v2x*dt;  
y2 = y2 + v2y*dt;  
z2 = z2 + v2z*dt;
```

Explanation: Each star's velocity is updated based on its acceleration and then its position is updated using the new velocity.

c. Update Planet Motion The gravitational acceleration on the planet from both stars is computed, and then its velocity and position are updated.

Listing 6: Updating Planet Motion

```
% Compute planet 's acceleration from star 1:  
r_p1_vec = [xp - x1, yp - y1, zp - z1];  
r_p1 = norm(r_p1_vec) + eps;  
% Compute planet 's acceleration from star 2:  
r_p2_vec = [xp - x2, yp - y2, zp - z2];  
r_p2 = norm(r_p2_vec) + eps;  
a_p1 = - m1 * r_p1_vec / r_p1^3;  
a_p2 = - m2 * r_p2_vec / r_p2^3;  
% Total acceleration for the planet:  
a_p = a_p1 + a_p2;
```

```
% Update planet velocity and position:
vxp = vxp + a_p(1)*dt;
vyp = vyp + a_p(2)*dt;
vzp = vzp + a_p(3)*dt;
xp = xp + vxp*dt;
yp = yp + vyp*dt;
zp = zp + vzp*dt;
```

Explanation: The planet's acceleration is the sum of the gravitational forces from both stars, and its motion is updated in the same manner as the stars.

d. Save Positions for Post-Processing Finally, the current positions of the stars and planet are stored for later visualization.

Listing 7: Saving Positions

```
% Save the current positions
x1save(k) = x1; y1save(k) = y1; z1save(k) = z1;
x2save(k) = x2; y2save(k) = y2; z2save(k) = z2;
xpSave(k) = xp; ypSave(k) = yp; zpSave(k) = zp;
end
```

Explanation: Storing the position data at each time step allows us to later animate or analyze the orbits.

6.2. Schwarzschild Spatial Stretching

In order to illustrate how a massive object warps space, we begin by creating a uniform 3D grid. We then convert the Cartesian grid to spherical coordinates, apply the Schwarzschild stretching, and finally convert back to Cartesian coordinates.

Our full transformation for a single grid point follows three steps:

- **Convert from Cartesian to Spherical Coordinates:** Given the Cartesian coordinates (X, Y, Z) , we first compute the radial distance R and the corresponding spherical coordinates θ and ϕ :

$$R = \sqrt{X^2 + Y^2 + Z^2}, \quad (6)$$

$$\theta = \arccos\left(\frac{Z}{R}\right), \quad (7)$$

$$\phi = \arctan\left(\frac{Y}{X}\right). \quad (8)$$

Here:

- X , Y , and Z are the original Cartesian coordinates.
- R is the radial distance from the origin.
- θ is the polar angle measured from the positive z -axis.
- ϕ is the azimuthal angle in the xy -plane, measured counterclockwise from the positive x -axis.

Code:

Listing 8: Conversion from Cartesian to Spherical Coordinates

```
clc; clear; close all;
%% ===== Step 0: Grid and Simulation Setup =====
N = 30;
L = 20;
[X, Y, Z] = meshgrid(linspace(-L, L, N), linspace(-L, L, N), linspace(0, pi, N));
R = sqrt(X.^2 + Y.^2 + Z.^2);
Theta = acos(Z. / (R+eps));
Phi = atan2(Y,X);
```

- **Apply the Schwarzschild Stretching and Compression:** Using the Schwarzschild metric solution, we compute the stretched radial distance:

$$R_{\text{stretched}} = R + R_s \log \left(\frac{\sqrt{R} + \sqrt{R - R_s}}{\sqrt{R_s}} \right), \quad (9)$$

and then apply the compression function to obtain the transformed coordinate:

$$R_{\text{new}} = \frac{R^2}{R_{\text{stretched}}}. \quad (10)$$

Here, R_s is the Schwarzschild radius, defined as:

$$R_s = \frac{2GM}{c^2}, \quad (11)$$

where:

- G is the gravitational constant, approximately $6.67430 \times 10^{-11} \text{ m}^3 \text{ kg}^{-1} \text{ s}^{-2}$,
- M is the mass of the central object,

-
- c is the speed of light in a vacuum, approximately 3×10^8 m/s.

Code:

Listing 9: Schwarzschild Stretching and Compression

```
% Define Schwarzschild parameters
G = 6.67430e-11; M = 1e30; c = 3e8;
Rs = 2*G*M/c^2;
% Compute stretched radial distance and compress:
R_stretched = R + Rs * log((sqrt(R) + sqrt(max(R-Rs,0)))/sqrt(Rs));
R_new = R.^2 ./ R_stretched;
```

- **Convert Back to Cartesian Coordinates:** Finally, using the transformed radial distance R_{new} along with the original spherical angles, we map back to Cartesian coordinates:

$$X_{\text{new}} = R_{\text{new}} \sin \theta \cos \phi, \quad (12)$$

$$Y_{\text{new}} = R_{\text{new}} \sin \theta \sin \phi, \quad (13)$$

$$Z_{\text{new}} = R_{\text{new}} \cos \theta. \quad (14)$$

Code:

Listing 10: Conversion Back to Cartesian Coordinates

```
X_new = R_new .* sin(Theta) .* cos(Phi);
Y_new = R_new .* sin(Theta) .* sin(Phi);
Z_new = R_new .* cos(Theta);
```

(Figure 7) shows the Comparison of original 3D grid and the Schwarzschild stretched 3D grid:

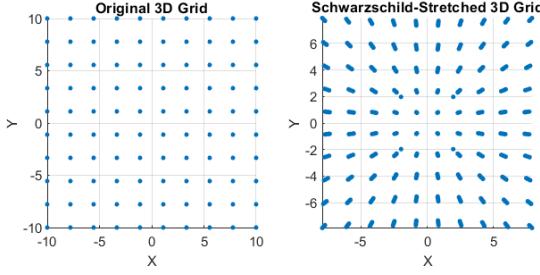


Figure 7: Comparison of original 3D grid and the Schwarzschild stretched 3D grid (top-down)

6.3. Simulating the Inspiral using a Drag Force

In our simulation, the drag force is evaluated as follows.

- **Determine the Instantaneous Separation and Compute the Damping Coefficient:**

First, we compute the instantaneous separation vector between the two stars:

$$\mathbf{r}_{12} = (x_1 - x_2, y_1 - y_2, z_1 - z_2)$$

and its magnitude:

$$r_{12} = \|\mathbf{r}_{12}\| .$$

The damping coefficient D is then calculated using the Peters formula:

$$D = \frac{32}{5} \frac{G^3}{c^5} \frac{\mu^2 M_{\text{tot}}^2}{r_{12}^4} .$$

Here:

- G is the gravitational constant,
- c is the speed of light,
- $\mu = \frac{m_1 m_2}{m_1 + m_2}$ is the reduced mass,
- $M_{\text{tot}} = m_1 + m_2$ is the total mass of the binary system.

Code:

Listing 11: Computing Damping Coefficient

```
r12_vec = [x1 - x2, y1 - y2, z1 - z2];
r12 = norm(r12_vec) + eps;
D = (32/5) * (mu^2 * M_tot^2) / (r12 ^ 4);
```

- **Calculate the Center-of-Mass Velocity:**

The center-of-mass (CM) velocity is computed as the mass-weighted average of the stars' velocities:

$$\mathbf{v}_{\text{cm}} = \frac{m_1 \mathbf{v}_1 + m_2 \mathbf{v}_2}{M_{\text{tot}}},$$

where $\mathbf{v}_1 = (v_{1x}, v_{1y}, v_{1z})$ and $\mathbf{v}_2 = (v_{2x}, v_{2y}, v_{2z})$.

Code:

Listing 12: Calculating Center-of-Mass Velocity

```
vcm = [(v1x*m1 + v2x*m2)/M_tot, (v1y*m1 + v2y*m2)/M_tot, (v1z*m1 + v2z*m2)/M_tot];
```

- **Compute the Relative (Orbital) Velocities:**

The velocities contributing to the orbital dynamics are obtained by subtracting the center-of-mass velocity from the individual velocities:

$$\Delta \mathbf{v}_1 = \mathbf{v}_1 - \mathbf{v}_{\text{cm}}, \quad \Delta \mathbf{v}_2 = \mathbf{v}_2 - \mathbf{v}_{\text{cm}}.$$

- **Apply the Drag (Radiation Reaction) Acceleration:**

The drag acceleration due to gravitational wave emission is proportional to the relative orbital velocities. For each star, the drag acceleration is:

$$\mathbf{a}_{1,\text{rad}} = -\frac{D}{m_1} \Delta \mathbf{v}_1, \quad \mathbf{a}_{2,\text{rad}} = -\frac{D}{m_2} \Delta \mathbf{v}_2.$$

This acceleration acts opposite to the stars' motion relative to the center of mass, effectively slowing the orbital motion.

Code:

Listing 13: Applying Radiation Reaction Acceleration

```
a1_rad = - (D/m1) * ([v1x, v1y, v1z] - vcm);
a2_rad = - (D/m2) * ([v2x, v2y, v2z] - vcm);
```

-
- **Combine with the Newtonian Gravitational Acceleration:**

The total acceleration acting on each star is the sum of the Newtonian gravitational acceleration and the drag acceleration:

$$\mathbf{a}_1 = \mathbf{a}_{1,\text{newt}} + \mathbf{a}_{1,\text{rad}}, \quad \mathbf{a}_2 = \mathbf{a}_{2,\text{newt}} + \mathbf{a}_{2,\text{rad}}.$$

These total accelerations are then used to update the stars' velocities and positions within the integration loop.

Code:

Listing 14: Updating Velocities and Positions

```
a1_total = a1_newt + a1_rad;
a2_total = a2_newt + a2_rad;
```

```
% Update star 1:
v1x = v1x + a1_total(1)*dt;
v1y = v1y + a1_total(2)*dt;
v1z = v1z + a1_total(3)*dt;
x1 = x1 + v1x*dt;
y1 = y1 + v1y*dt;
z1 = z1 + v1z*dt;
```

```
% Update star 2:
v2x = v2x + a2_total(1)*dt;
v2y = v2y + a2_total(2)*dt;
v2z = v2z + a2_total(3)*dt;
x2 = x2 + v2x*dt;
y2 = y2 + v2y*dt;
z2 = z2 + v2z*dt;
```

6.4. Gravitational Waves and TT Gauge Transformation

For each grid point, the TT gauge transformation is applied to account for the gravitational wave strains produced by the inspiraling binary. The procedure is as follows:

- **Compute Distances and Retarded Times:**

For each grid point, we calculate its Euclidean distance from each star. For star 1, the

distance is computed as

$$d_1 = \sqrt{(X - x_1)^2 + (Y - y_1)^2 + (Z - z_1)^2},$$

and similarly for star 2:

$$d_2 = \sqrt{(X - x_2)^2 + (Y - y_2)^2 + (Z - z_2)^2}.$$

Using these distances, the retarded times for each star are determined by:

$$t_{1,\text{ret}} = t - \frac{d_1}{c}, \quad t_{2,\text{ret}} = t - \frac{d_2}{c},$$

where c is the speed of light.

Code:

Listing 15: Computing Distances and Retarded Times for TT Transformation

```
d1 = sqrt((X - x1).^2 + (Y - y1).^2 + (Z - z1).^2);
d2 = sqrt((X - x2).^2 + (Y - y2).^2 + (Z - z2).^2);
t1_ret = t - d1/c;
t2_ret = t - d2/c;
```

- **Determine the Angles and Pattern Factors:**

At each grid point, the angle between the star-to-grid vector and the z -axis is computed.

For star 1, this is given by:

$$\theta_1 = \arccos\left(\frac{Z - z_1}{d_1}\right),$$

and similarly for star 2:

$$\theta_2 = \arccos\left(\frac{Z - z_2}{d_2}\right).$$

From these angles, the quadrupole pattern factors (which determine the angular dependence of the wave) are computed as:

$$\text{pattern}_1 = \frac{3 \cos^2 \theta_1 - 1}{2}, \quad \text{pattern}_2 = \frac{3 \cos^2 \theta_2 - 1}{2}.$$

Code:

Listing 16: Calculating Angles and Pattern Factors

```

theta1 = acos((Z - z1) ./ (d1 + eps));
theta2 = acos((Z - z2) ./ (d2 + eps));
pattern1 = (3*cos(theta1).^2 - 1) / 2;
pattern2 = (3*cos(theta2).^2 - 1) / 2;

```

- **Compute the Polarization Strains:**

The precomputed orbital frequency ω is used along with the retarded times to calculate the gravitational wave strains at each grid point. For star 1, the strains are:

$$h_{+1} = A \cos(2\omega t_{1,\text{ret}}) \cdot \text{pattern}_1, \quad h_{\times 1} = A \sin(2\omega t_{1,\text{ret}}) \cdot \text{pattern}_1,$$

and similarly for star 2:

$$h_{+2} = A \cos(2\omega t_{2,\text{ret}}) \cdot \text{pattern}_2, \quad h_{\times 2} = A \sin(2\omega t_{2,\text{ret}}) \cdot \text{pattern}_2.$$

The total strains at a grid point are then obtained by summing the contributions from both stars:

$$h_+ = h_{+1} + h_{+2}, \quad h_\times = h_{\times 1} + h_{\times 2}.$$

Code:

Listing 17: Computing Polarization Strains

```

omega = omegaSave(k);
hplus1 = A * cos(2*omega*t1_ret) .* pattern1;
hcross1 = A * sin(2*omega*t1_ret) .* pattern1;
hplus2 = A * cos(2*omega*t2_ret) .* pattern2;
hcross2 = A * sin(2*omega*t2_ret) .* pattern2;

hplus_total = hplus1 + hplus2;
hcross_total = hcross1 + hcross2;

```

- **Apply the TT Gauge Transformation:**

With the total gravitational wave strains known, the spatial coordinates of each grid

point are modified. The TT gauge transformation for the coordinates is:

$$X_{\text{new}} = P(\theta) \left(X + \frac{1}{2} (h_+ X + h_\times Y) \right), \quad (15)$$

$$Y_{\text{new}} = P(\theta) \left(Y - \frac{1}{2} (h_+ Y - h_\times X) \right), \quad (16)$$

$$Z_{\text{new}} = P(\theta) Z, \quad (17)$$

where $P(\theta)$ is a pattern function that may further modulate the transformation with time.

Code:

```
Listing 18: TT Gauge Transformation on Grid
X_new = X + 0.5 * (hplus_total .* X + hcross_total .* Y);
Y_new = Y - 0.5 * (hplus_total .* Y - hcross_total .* X);
Z_new = Z;

xGridTT(:, k) = X_new(:);
yGridTT(:, k) = Y_new(:);
zGridTT(:, k) = Z_new(:);
```

- **Apply the Transformation to the Planet:**

In addition to the grid points, the same coordinate transformations are applied to the planet's background position. This ensures that the planet's motion is consistently influenced by the gravitational waves.

Code:

```
Listing 19: TT Gauge Transformation on Planet
xp_bg = xpSave(k);    yp_bg = ypSave(k);    zp_bg = zpSave(k);

d1p = sqrt((xp_bg - x1save(k))^2 + (yp_bg - y1save(k))^2 + (zp_bg - z1save(k))^2);
d2p = sqrt((xp_bg - x2save(k))^2 + (yp_bg - y2save(k))^2 + (zp_bg - z2save(k))^2);

t1_ret = t - d1p/c;
t2_ret = t - d2p/c;

theta1p = acos((zp_bg - z1save(k)) / (d1p + eps));
```

```

theta2p = acos(( zp_bg - z2save(k)) / (d2p + eps));

pattern1p = (3*cos(theta1p)^2 - 1)/2;
pattern2p = (3*cos(theta2p)^2 - 1)/2;

omega_k = omegaSave(k);
hplus1 = A * cos(2*omega_k*t1_ret) * pattern1p;
hcross1 = A * sin(2*omega_k*t1_ret) * pattern1p;
hplus2 = A * cos(2*omega_k*t2_ret) * pattern2p;
hcross2 = A * sin(2*omega_k*t2_ret) * pattern2p;

hplus_total = hplus1 + hplus2;
hcross_total = hcross1 + hcross2;

xp_new = xp_bg + 0.5*( hplus_total*xp_bg + hcross_total*yp_bg );
yp_new = yp_bg - 0.5*( hplus_total*yp_bg - hcross_total*xp_bg );
zp_new = zp_bg;

xpTT(k) = xp_new;
ypTT(k) = yp_new;
zpTT(k) = zp_new;

```

(Figure 8) shows the gravitational waves at the beginning of the simulation:

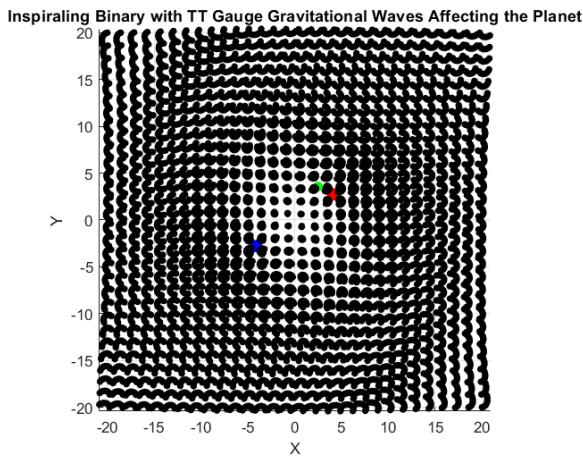


Figure 8: Gravitational waves at beginning (top-down)

(Figure 9) shows the gravitational waves at the end of the simulation:

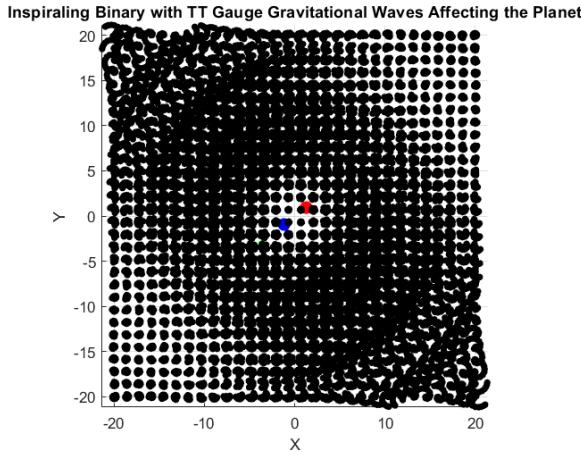


Figure 9: Gravitational waves at end (top-down)

6.5. Discussion of Key Considerations

Throughout the implementation, several critical choices were made:

- **Numerical Stability:** Small values (using `eps`) are added to denominators to prevent division by zero.
- **Normalization:** The real-life constants such as c , G , etc. have been normalized to equal one in some cases for the sake of simulation and visualization.
- **Symplectic Euler Integration:** This method updates velocities before positions, which helps preserve energy properties over long simulations.

7. Results and Discussion

In this section, we present and interpret the outcomes of our simulations under various initial conditions. Specifically, we focus on varying the initial binary star separation, r_{sep} , among four values (5, 10, 15, and 20). For each of these separations, we run two sets of simulations:

- **Binary-Only:** A pure two-body system consisting of the two stars with post-Newtonian radiation reaction.
- **Binary + Planet:** The same two stars, but now including a small planet orbiting the system.

Our goal is to compare how the presence of the planet, as well as different initial separations, affects four quantities over time:

-
1. $E_{\text{bin}}(t)$: The total orbital energy of the binary (kinetic + potential).
 2. $F_{\text{gw}}(t)$: A leading-order estimate of the gravitational-wave (GW) power.
 3. $r_{12}(t)$: The instantaneous separation of the two stars.
 4. $\omega_{\text{GW}}(t)$: The orbital (or GW) frequency of the binary.

We visualize each of these quantities vs. time for each separation value in Figures 10, 11, 12, and 13. Additionally, we provide animations of each scenario (linked in Section 7.4).

7.1. Parametric Studies: Varying the Star Separation

Initial Separation of 5. When $r_{\text{sep}} = 5$, the two stars begin in a relatively tight orbit. The orbital velocity is high, and we observe:

- **Binary-Only:** The binary quickly loses orbital energy due to radiation reaction, leading to a rapid inspiral. Consequently, $E_{\text{bin}}(t)$ drops more steeply than for larger separations, $r_{12}(t)$ shrinks faster, and $\omega_{\text{GW}}(t)$ rises quickly.
- **Binary + Planet:** The planet exerts additional gravitational tugs on both stars. Although the general inspiral trend remains, the binary energy and frequency show small oscillations superimposed on the overall decay/increase, due to three-body interactions.

Initial Separation of 10. For $r_{\text{sep}} = 10$, the stars orbit at more moderate speeds:

- **Binary-Only:** The curves of $E_{\text{bin}}(t)$, $F_{\text{gw}}(t)$, $r_{12}(t)$, and $\omega_{\text{GW}}(t)$ appear smoother. The energy still declines steadily, and the frequency grows gradually.
- **Binary + Planet:** As before, the planet induces small perturbations, but the energy/frequency fluctuations are milder than the $r_{\text{sep}} = 5$ case.

Initial Separation of 15 and 20. When r_{sep} increases further, the initial orbital velocity is even lower:

- **Binary-Only:** The inspiral remains smooth, but proceeds more slowly. This is reflected by a gentle decline in E_{bin} and a slower rise in ω_{GW} .
- **Binary + Planet:** The perturbations are more pronounced, especially for the orbital energy and frequency, observed when the planet's orbit nears the star.

7.2. Comparison of With vs. Without the Planet

To highlight the role of the planet, we overlay the binary-only and binary+planet results in the same plots for each r_{sep} .

- **Binary Energy:** Without the planet, E_{bin} decreases smoothly; with the planet, oscillations appear. These oscillations are more pronounced at larger separations.
- **GW Power:** In all cases, $F_{\text{gw}}(t)$ grows over time as the orbit shrinks. The presence of the planet mainly causes minor deviations from the smooth curve.
- **Separation:** $r_{12}(t)$ declines, but in the planet runs we see tiny “wiggles,” indicating transient star–planet interactions.
- **Orbital Frequency:** ω_{GW} consistently increases as the stars spiral in, though the planet runs show brief frequency dips before continuing the upward trend.

Overall, the planet causes moderate perturbations in the inspiral. As the orbit becomes extremely tight, the GW emission dominates, and the planet’s effect is overshadowed by the strong radiation reaction.

7.3. Interpretation and Limitations

The key takeaways are:

1. **Closer initial separations** lead to a faster inspiral, causing more rapid changes in E_{bin} and ω_{GW} .
2. **Adding a planet** introduces extra three-body dynamics, resulting deviations from the two-body inspiral curves.

7.4. Figures and Animations

Figures 10 through 13 show the main results:

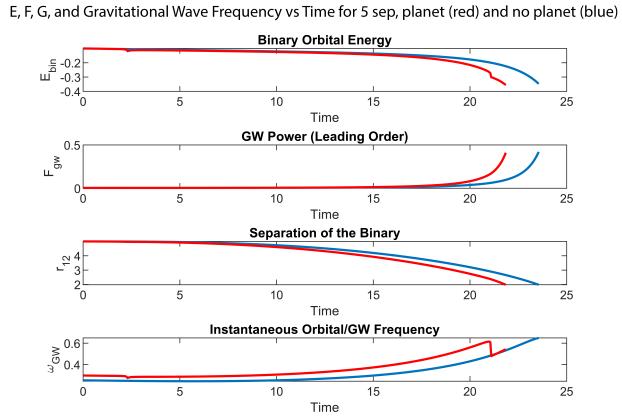


Figure 10: Comparison of 5 separation with and without planet

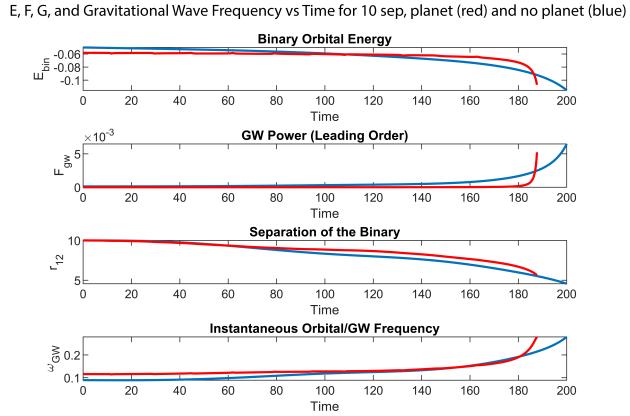


Figure 11: Comparison of 10 separation with and without planet

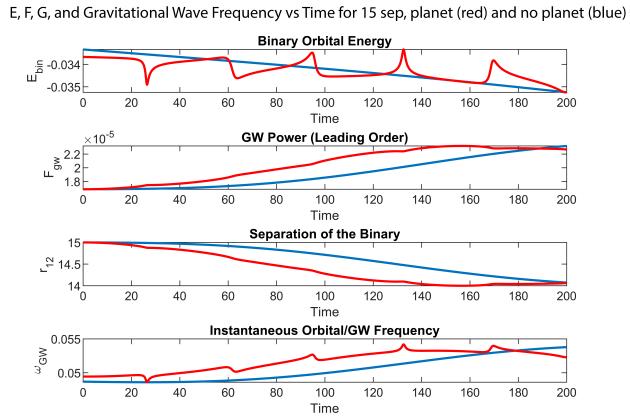


Figure 12: Comparison of 15 separation with and without planet

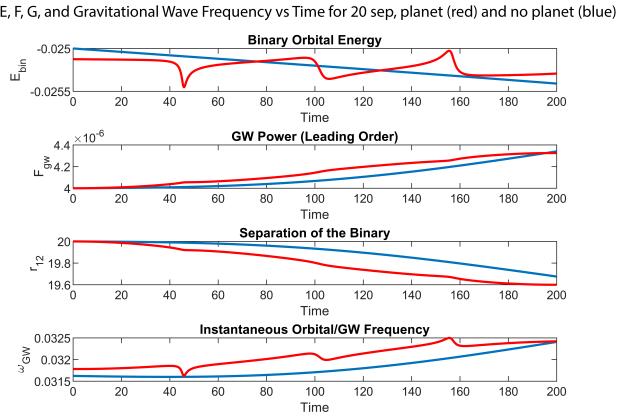


Figure 13: Comparison of 20 separation with and without planet

For each scenario, we also recorded a short animation showing the top-down view of the orbital motion and the TT-gauge warping of the background grid. These can be accessed at the following links:

- $r_{\text{sep}} = 5$: <https://www.youtube.com/watch?v=79-58tyOzgg>
- $r_{\text{sep}} = 10$: <https://www.youtube.com/watch?v=zQcaeUaXGG4>
- $r_{\text{sep}} = 15$: <https://www.youtube.com/watch?v=IDVIfRW4nGQ>
- $r_{\text{sep}} = 20$: <https://www.youtube.com/watch?v=Bg8fXXjGaCg>

8. Conclusions

This project set out to simulate the effects of relativistic gravity by exploring how mass warps space and generates gravitational waves. We combined the Schwarzschild metric to model the radial stretching of space with post-Newtonian corrections—implemented as a drag force—to approximate gravitational radiation from an inspiraling binary system. Additionally, we incorporated transverse-traceless (TT) gauge transformations to visualize the propagation of gravitational waves across a three-dimensional grid, affecting both the binary system and an orbiting test planet.

Through the development and analysis of our simulation in MATLAB, we deepened our understanding of several key concepts in gravitational physics. In particular, we learned how general relativity’s abstract descriptions of spacetime curvature can be translated into concrete numerical models. We also gained valuable insights into the conservation of energy and angular momentum in systems where gravitational waves carry energy away, and how

subtle numerical integration methods, like the symplectic Euler scheme, can be effectively used to capture these dynamics.

Overall, the project successfully bridges theoretical formulations with computational practice, demonstrating that complex relativistic phenomena can be approximated and visualized in a tractable and insightful manner.

9. Appendix

9.1. Solving the Schwarzschild Solution

9.1.1 Realizing Radial Stretching from the Schwarzschild Solution

This is the Schwarzschild solution (18), which describes the spacetime geometry around a spherically symmetric, non-rotating, uncharged massive object [4].

$$ds^2 = - \left(1 - \frac{2GM}{c^2r}\right) dt^2 + \left(1 - \frac{2GM}{c^2r}\right)^{-1} dr^2 + r^2 d\theta^2 + r^2 \sin^2 \theta d\phi^2 \quad (18)$$

Here,

- ds is the line element, representing an infinitesimal interval of spacetime.
- dt is an infinitesimal element of time (coordinate time).
- dr is an infinitesimal element of the radial coordinate, representing the distance from the center of the mass.
- $d\theta$ is an infinitesimal change in the polar angle, measured from the z -axis down toward the xy -plane.
- $d\phi$ is an infinitesimal change in the azimuthal angle, measured in the xy -plane counterclockwise from the positive x -axis.
- G is the universal gravitational constant, approximately $6.67430 \times 10^{-11} \text{ m}^3 \text{ kg}^{-1} \text{ s}^{-2}$.
- M is the mass of the central object (such as a star or black hole) generating the gravitational field.
- c is the speed of light in a vacuum, approximately $3 \times 10^8 \text{ m/s}$.
- r is the radial coordinate in the Schwarzschild metric, which measures the distance from the center of the mass.

Ignoring the time component, we isolate the radial portion of this solution, as radial transformations preserve their respective spherical angles. This leaves us with 19

$$dR_{\text{stretched}}^2 = \left(1 - \frac{2GM}{c^2 r}\right)^{-1} dr^2, \quad (19)$$

where $dR_{\text{stretched}}$ represents the stretched distance element as predicted by the Schwarzschild solution.

Taking the square root of both sides gives (20).

$$dR_{\text{stretched}} = \frac{dr}{\sqrt{1 - \frac{2GM}{c^2 r}}}. \quad (20)$$

We integrate from 0 to a radial point R to calculate the total stretched distance.

$$R_{\text{stretched}} = \int_0^R \frac{dr}{\sqrt{1 - \frac{2GM}{c^2 r}}}. \quad (21)$$

With the change of variable (22),

$$r = \frac{2GM}{c^2}(1 + u^2), \quad (22)$$

we obtain (23)

$$R_{\text{stretched}} = R + \frac{2GM}{c^2} \log \left(\frac{\sqrt{R} + \sqrt{R - \frac{2GM}{c^2}}}{\sqrt{\frac{2GM}{c^2}}} \right), \quad (23)$$

and finally (24),

$$R_{\text{stretched}} = R + R_s \log \left(\frac{\sqrt{R} + \sqrt{R - R_s}}{\sqrt{R_s}} \right), \quad (24)$$

where R_s is defined as the Schwarzschild radius, given by (25).

$$R_s = \frac{2GM}{c^2}. \quad (25)$$

9.1.2 Choice of Compression Function

To simulate the spatial distortion caused by a massive object, we apply a coordinate transformation that follows the radial solution of the Schwarzschild metric along with a compression function. The compression function is introduced as a tool to visualize the warping of space

into a higher dimension. It is designed so that the more a region is stretched, the denser the grid points become, while still preserving the enclosed volume of the original space.

In our analysis, we define the following variables:

- R is the original radial coordinate in the Schwarzschild metric.
- $R_{\text{stretched}}$ is the radial coordinate after stretching, as determined by the Schwarzschild solution.
- R_{new} is the new, compressed radial coordinate used for the grid.

For regions where the stretching increases, we expect an inverse proportional relationship:

$$R_{\text{new}} \propto \frac{1}{R_{\text{stretched}}}.$$

Moreover, since volume elements in spherical coordinates scale approximately as:

$$dV \sim R^2 dR,$$

Compressing the radial coordinates while keeping the volume distortion proportional suggests:

$$R_{\text{new}} \propto \frac{R^2}{R_{\text{stretched}}}.$$

This choice ensures that:

- $R_{\text{new}} \approx R$ far from the mass (i.e., no warping at infinity),
- $R_{\text{new}} \rightarrow 0$ near the Schwarzschild radius (indicating extreme warping near the central mass).

Thus, we select the compression function as

$$R_{\text{new}} = \frac{R^2}{R_{\text{stretched}}}, \quad (26)$$

which preserves radial contraction while keeping the overall shape of the spatial grid intuitive.

9.2. Solving for Drag as a Post-Newtonian Correction

According to General Relativity, a binary system is not stable and loses orbital energy, which is re-emitted as gravitational waves. These waves carry away energy, leading to the gradual inspiral of the binary system. The total power radiated via quadrupole emission is given by the Peters formula (27) [14]:

$$P_{\text{GW}} = -\frac{32}{5} \frac{G^4}{c^5} \frac{\mu^2 M_{\text{tot}}^3}{r^5} \quad (27)$$

Where:

- G is the gravitational constant, approximately $6.67430 \times 10^{-11} \text{ m}^3 \text{ kg}^{-1} \text{ s}^{-2}$.
- c is the speed of light in a vacuum, approximately $3 \times 10^8 \text{ m/s}$.
- $\mu = \frac{m_1 m_2}{m_1 + m_2}$ is the reduced mass of the two binary stars with masses m_1 and m_2 .
- $M_{\text{tot}} = m_1 + m_2$ is the total mass of the system.
- r is the separation distance between the two bodies.

This equation describes the rate at which the binary system loses energy due to gravitational wave emission.

The total orbital energy of the binary system is given by (28).

$$E = -\frac{G m_1 m_2}{2r} \quad (28)$$

Taking the time derivative,

$$\frac{dE}{dt} = \frac{G m_1 m_2}{2r^2} \frac{dr}{dt} \quad (29)$$

Setting (29) equal to the power loss due to gravitational waves (30).

$$\frac{G m_1 m_2}{2r^2} \frac{dr}{dt} = -\frac{32}{5} \frac{G^4}{c^5} \frac{\mu^2 M_{\text{tot}}^3}{r^5} \quad (30)$$

Solving for $\frac{dr}{dt}$,

$$\frac{dr}{dt} = -\frac{64}{5} \frac{G^3}{c^5} \frac{\mu M_{\text{tot}}^2}{r^3} \quad (31)$$

This result illuminates the fact that as the binary system loses orbital energy, its radial separation decreases over time, a process known as inspiral.

9.2.1 Expressing the Inspiral as a Drag Acceleration

One way to model the loss of orbital energy is by applying a drag force on the binary system.

The orbital velocity of each star in a circular orbit is given by: For a circular orbit, the relative orbital speed is given by (32)

$$v_{\text{rel}} = \sqrt{\frac{G M_{\text{tot}}}{r}}, \quad (32)$$

such that

$$v_{\text{rel}}^2 = \frac{G M_{\text{tot}}}{r}. \quad (33)$$

In the numerical simulation, we model the effect of gravitational radiation by introducing a damping (or drag) force that acts only on the relative motion (since gravitational waves carry away energy from the orbital motion relative to the center of mass) [14]. We define the drag force as (34).

$$F_{\text{drag}} = -D v_{\text{rel}}, \quad (34)$$

where D is the damping coefficient to be determined.

The power (rate of energy loss) due to this drag force is given by (35).

$$P_{\text{drag}} = F_{\text{drag}} v_{\text{rel}} = D v_{\text{rel}}^2. \quad (35)$$

To mimic gravitational wave energy loss, we equate the two power expressions.

$$D v_{\text{rel}}^2 = \frac{32 G^4 \mu^2 M_{\text{tot}}^3}{5 c^5 r^5}. \quad (36)$$

Substituting $v_{\text{rel}}^2 = \frac{G M_{\text{tot}}}{r}$ into (36) yields (37).

$$D \frac{G M_{\text{tot}}}{r} = \frac{32 G^4 \mu^2 M_{\text{tot}}^3}{5 c^5 r^5}. \quad (37)$$

Solving for D by multiplying both sides by $\frac{r}{G M_{\text{tot}}}$.

$$D = \frac{32 G^4 \mu^2 M_{\text{tot}}^3}{5 c^5 r^5} \cdot \frac{r}{G M_{\text{tot}}} = \frac{32 G^3 \mu^2 M_{\text{tot}}^2}{5 c^5 r^4}. \quad (38)$$

So our damping coefficient is given by (39)

$$D = \frac{32 G^3 \mu^2 M_{\text{tot}}^2}{5 c^5 r^4}. \quad (39)$$

This is the final form of the damping coefficient.

9.3. Gravitational Wave Coordinate Transformations

9.3.1 The Minkowski Metric Tensor and TT Gauge

The Minkowski metric tensor, $\eta_{\mu\nu}$ (40), encodes the structure of spacetime in special relativity by defining how distances are measured in a flat spacetime. A key consequence of this formulation is that the more one travels through space, the less one moves through time in

a relativistic sense. [7]

$$\eta_{\mu\nu} = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (40)$$

The elements of the Minkowski metric describe how different spacetime coordinates relate to one another. It provides a method to compute the spacetime interval between two four-dimensional events with coordinates x^μ and x'^μ . Given two four-dimensional position vectors (41):

$$x^\mu = (ct, x, y, z), \quad x'^\mu = (ct', x', y', z'), \quad (41)$$

The squared spacetime interval (or proper distance) between them is given by the contracted form (42)

$$s^2 = \eta_{\mu\nu}(x^\mu - x'^\mu)(x^\nu - x'^\nu) \quad (42)$$

or, in expanded form (43).

$$s^2 = -(ct - ct')^2 + (x - x')^2 + (y - y')^2 + (z - z')^2. \quad (43)$$

The Minkowski metric does not compare two four-dimensional coordinates in a Euclidean sense; rather, it determines how the separation between two events behaves in relativistic spacetime.

Gravitational waves are experienced as perturbations arising from a changing mass quadrupole moment added to the flat spacetime described by the Minkowski metric tensor. In the effort to linearize general relativity for the sake of simulation, we must use the weak-field approximation for the mass quadrupole moment tensor [10]. This involves combining the Minkowski metric tensor, which describes flat spacetime [7], with the weak-field mass quadrupole moment tensor. It must be noted that this linearization technique is accurate as long as the wavelength is much longer than the source length, or the orbital separation of the binary stars in this case [11]. As a consequence, the accuracy of our model will decrease as higher-frequency waves are emitted.

The weak-field equation for the mass quadrupole moment tensor is given by

$$h^{\mu\nu} = \frac{2G}{rc^4} \ddot{Q}_{\text{TT}}^{\mu\nu} \left(t - \frac{r}{c} \right).$$

We can further approximate the full mass quadrupole moment tensor with the transverse-traceless (TT) gauge. The TT gauge (44) simplifies calculations for numerous reasons, the most notable being that it reduces the number of polarization states to two and, due to its transverse property, ensures there are no time-time or time-space components in the wave [1] [8]:

$$h^{\mu\nu} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & h_+ & h_\times & 0 \\ 0 & h_\times & -h_+ & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (44)$$

We assume that the dominant wave propagation is along the $+z$ -axis for two reasons:

1. It simplifies calculations, as the TT gauge is easiest to impose when the wave travels in a single direction.
2. It aligns with astrophysical scenarios, since many gravitational wave observations (such as LIGO detections) involve waves approaching from a dominant direction.

Here, h_+ and h_\times represent the two time-dependent polarization states. The h_+ polarization squeezes space along one principal axis (e.g., the x -axis) while simultaneously stretching it along the perpendicular axis (e.g., the y -axis). The h_\times polarization produces the same effect, but rotated by 45 degrees, causing stretching and squeezing along diagonal directions [8]. We select sinusoidal polarization functions that are 90 degrees out of phase in accordance with literature [12].

Implementing gravitational waves as TT-gauge perturbations in the flat Minkowski metric is achieved by writing

$$g_{\mu\nu} = \eta_{\mu\nu} + h_{\mu\nu}. \quad (45)$$

Thus, the full metric becomes (46).

$$g_{\mu\nu} = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 + h_+ & h_\times & 0 \\ 0 & h_\times & 1 - h_+ & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (46)$$

The modified line element under this perturbation is:

$$ds^2 = -c^2 dt^2 + (1 + h_+) dx^2 + 2h_\times dx dy + (1 - h_+) dy^2 + dz^2. \quad (47)$$

9.3.2 Wave Coordinate Transformations

Assume that a point originally at (X, Y, Z) is displaced by the gravitational wave such that:

$$X' = X + \xi_x, \quad Y' = Y + \xi_y, \quad Z' = Z. \quad (48)$$

Taking the differentials, we obtain:

$$dX' = dX + \frac{\partial \xi_x}{\partial X} dX + \frac{\partial \xi_x}{\partial Y} dY, \quad (49)$$

$$dY' = dY + \frac{\partial \xi_y}{\partial X} dX + \frac{\partial \xi_y}{\partial Y} dY. \quad (50)$$

The goal is to determine the displacements ξ_x and ξ_y .

In our transformation to Euclidean space, the distance between two nearby points is given by:

$$ds^2 = dX^2 + dY^2. \quad (51)$$

Squaring both differentials and retaining only first-order terms yields:

$$(dX')^2 = dX^2 + 2\frac{\partial \xi_x}{\partial X} dX^2 + 2\frac{\partial \xi_x}{\partial Y} dX dY, \quad (52)$$

$$(dY')^2 = dY^2 + 2\frac{\partial \xi_y}{\partial X} dX dY + 2\frac{\partial \xi_y}{\partial Y} dY^2. \quad (53)$$

Rearranging, we have:

$$dX^2 = -(dX')^2 + 2\frac{\partial \xi_x}{\partial X} dX^2 + 2\frac{\partial \xi_x}{\partial Y} dX dY, \quad (54)$$

$$dY^2 = -(dY')^2 + 2\frac{\partial \xi_y}{\partial X} dX dY + 2\frac{\partial \xi_y}{\partial Y} dY^2. \quad (55)$$

Substituting these into the expression for ds^2 (47), we obtain (56).

$$ds^2 = -(dX')^2 + 2\frac{\partial \xi_x}{\partial X} dX^2 + 2\frac{\partial \xi_x}{\partial Y} dX dY - (dY')^2 + 2\frac{\partial \xi_y}{\partial X} dX dY + 2\frac{\partial \xi_y}{\partial Y} dY^2. \quad (56)$$

Under the gravitational wave, the perturbed metric is given by:

$$ds^2 = (1 + h_+) dX^2 + 2h_\times dX dY + (1 - h_+) dY^2. \quad (57)$$

Matching terms with the perturbed metric, we find:

$$\frac{\partial \xi_x}{\partial X} = \frac{1}{2} h_+, \quad (58)$$

$$\frac{\partial \xi_y}{\partial Y} = -\frac{1}{2} h_+, \quad (59)$$

$$\frac{\partial \xi_x}{\partial Y} + \frac{\partial \xi_y}{\partial X} = h_\times. \quad (60)$$

To solve for ξ_x , we integrate (61).

$$\xi_x = \int \frac{1}{2} h_+ dX = \frac{1}{2} h_+ X + f(Y). \quad (61)$$

Similarly, for ξ_y we integrate (62).

$$\xi_y = \int -\frac{1}{2} h_+ dY = -\frac{1}{2} h_+ Y + g(X). \quad (62)$$

To determine $f(Y)$ and $g(X)$, we use the relation (63).

$$\frac{\partial \xi_x}{\partial Y} + \frac{\partial \xi_y}{\partial X} = h_\times. \quad (63)$$

Substituting, we obtain (64).

$$\frac{df}{dY} + \frac{dg}{dX} = h_\times. \quad (64)$$

Choosing:

$$f(Y) = \frac{1}{2} h_\times Y, \quad g(X) = \frac{1}{2} h_\times X, \quad (65)$$

we then have:

$$\xi_x = \frac{1}{2} (h_+ X + h_\times Y), \quad (66)$$

$$\xi_y = -\frac{1}{2} (h_+ Y - h_\times X). \quad (67)$$

Thus, the coordinate transformations become:

$$X' = X + \frac{1}{2} (h_+ X + h_\times Y), \quad (68)$$

$$Y' = Y - \frac{1}{2} (h_+ Y - h_\times X), \quad (69)$$

$$Z' = Z. \quad (70)$$

These equations describe how our approximated gravitational wave modifies the spatial

coordinates.²

It should be noted that during computation, each source (star 1 and star 2) contributes waves that are summed together.

In order to visualize the quadrupole nature and angular dependence of the waves [6], we finally multiply our coordinate transformations by a quadrupole pattern function $P(\theta)$ [13].

Thus,

$$X_{\text{new}} = \left(X + \frac{1}{2} ((P(\theta_1)h_{+1} + P(\theta_2)h_{+2}) X + (P(\theta_1)h_{\times 1} + P(\theta_2)h_{\times 2}) Y) \right), \quad (71)$$

$$Y_{\text{new}} = \left(Y - \frac{1}{2} ((P(\theta_1)h_{+1} + P(\theta_2)h_{+2}) Y - (P(\theta_1)h_{\times 1} + P(\theta_2)h_{\times 2}) X) \right), \quad (72)$$

$$Z_{\text{new}} = (Z). \quad (73)$$

Where,

- $X_{\text{new}}, Y_{\text{new}}, Z_{\text{new}}$ represent the new transformed coordinates,
- X, Y, Z represent the original coordinates,
- $h_{+1} = A \cos(2\omega t_{\text{ret}1})$ represents the plus polarization produced by the first binary star (See Appendix 9.4.2),
- $h_{+2} = A \cos(2\omega t_{\text{ret}2})$ represents the plus polarization produced by the second binary star (See Appendix 9.4.2),
- $h_{\times 1} = A \sin(2\omega t_{\text{ret}1})$.. represents the cross polarization produced by the first binary star (See Appendix 9.4.2)
- $h_{\times 2} = A \sin(2\omega t_{\text{ret}2})$.. represents the cross polarization produced by the second binary star (See Appendix 9.4.2)
- ω represents the orbital frequency (See Appendix 9.4.2)
- t_{ret} = is the retarded time (See Appendix 9.4.2)
- $P(\theta_1)$ = is the pattern function produced by the first binary star (See Appendix 9.4.1)
- $P(\theta_2)$ = is the pattern function produced by the second binary star (See Appendix 9.4.1)

²Please note that this is an expansion of the coordinate transformations provided by Bernuzzi [2].

9.4. Pattern Functions

9.4.1 Pattern Functions

The gravitational wave is not uniform across space—it exhibits a directional dependence. The strain experienced at any given point depends on its angle relative to the binary system [6]. This is where the quadrupole pattern functions come into play. They quantify how much of the gravitational wave's effect is felt at a given angle θ from a star in the system.

These functions are incorporated into our equations because gravitational waves are generated by a time-varying quadrupole moment (not a monopole or dipole), and hence are not strictly planar.

We define the pattern functions based on the Landau & Lifshitz quadrupole moment expressions [13]:

$$P(\theta) = \mu r^2(3 \cos^2 \theta - 1) \quad (74)$$

$$P(\theta) = \frac{(3 \cos^2 \theta - 1)}{2}. \quad (75)$$

In the second equation, the prefactor μr^2 has been removed to emphasize the angular dependence of the radiation in a dimensionless form, and the factor of $\frac{1}{2}$ was introduced as a convenient normalization, ensuring consistency with standard quadrupole forms and facilitating easier comparisons in simulation results.

9.4.2 Time Dependent Functions h_+ and h_\times

We choose sinusoidal functions that are 90° out of phase for h_+ and h_\times , which is the standard choice in General Relativity for a binary system [12]. Thus, we take:

$$h_+ = A \cos(2\omega t) \quad (76)$$

$$h_\times = A \sin(2\omega t). \quad (77)$$

This choice reflects the quadrupole oscillation pattern of the wave strain. Key points to note are:

- The factor 2ω arises because gravitational waves oscillate at twice the binary's orbital frequency. [9]
- A cosine is used for h_+ and a sine for h_\times to represent the two polarization states, which are 90° out of phase.

Before implementing these functions in the code, we change the time parameter t to the retarded time t_{ret} to account for the time it takes for the wave to propagate to the point of

interest:

$$t_{\text{ret}} = t - \frac{d}{c}, \quad (78)$$

Where:

- t_{ret} is the time when the wave was emitted from the binary,
- d is the distance from the binary to the observer (or grid point),
- c is the speed of light.

Our final equations are:

$$h_+ = A \cos(2\omega t_{\text{ret}}) \quad (79)$$

$$h_\times = A \sin(2\omega t_{\text{ret}}). \quad (80)$$

References

- [1] Sebastiano Bernuzzi. Gravitational waves: Theory and data analysis lecture notes, 2023. Accessed: February 21, 2025, p. 13.
- [2] Sebastiano Bernuzzi. Gravitational waves: Theory and data analysis lecture notes, 2023. Accessed: February 21, 2025, p. 16.
- [3] James Binney and Scott Tremaine. *Galactic Dynamics*. Princeton University Press, Princeton, NJ, 2nd edition, 2008.
- [4] Tarun Biswas. Physical interpretation of coordinates for the schwarzschild metric, 2008.
- [5] Teviet Creighton. Gravitational waves: A detailed explanation, 2025. Accessed: 2025-02-19.
- [6] Giulia Cusin, Cyril Pitrou, and Jean-Philippe Uzan. The signal of the gravitational wave background and the angular correlation of its energy density. *Phys. Rev. D*, 97:123527, Jun 2018.
- [7] Paul D'Alessandris. 3.1: Minkowski metric. [https://phys.libretexts.org/Bookshelves/Modern_Physics/Book%3A_Spiral_Modern_Physics_\(D%27Alessandris\)/3%3A_Spacetime_and_General_Relativity/3.1%3A_Minkowski_Metric](https://phys.libretexts.org/Bookshelves/Modern_Physics/Book%3A_Spiral_Modern_Physics_(D%27Alessandris)/3%3A_Spacetime_and_General_Relativity/3.1%3A_Minkowski_Metric), 2018. Accessed: February 19, 2025.
- [8] EigenChris. Relativity 109d: Gravitational waves - transverse-traceless gauge (plus and cross polarizations), 2023. Accessed: 2025-02-19.

-
- [9] M. R. V. Gardner and B. Mashhoon. Gravitational waves from orbiting binaries without quadrupole approximation. *American Journal of Physics*, 86(3):186–196, 2018.
 - [10] Kostas Glampedakis. Gravitational waves: Theory - sources - detection, 2009. Lecture notes, page 6.
 - [11] Kostas Glampedakis. Gravitational waves: Theory - sources - detection, 2009. Lecture notes, page 14.
 - [12] Maximiliano Isi. Parametrizing gravitational-wave polarizations. *Classical and Quantum Gravity*, 40(20):203001, September 2023.
 - [13] Robert M. L. Baker Jr. Precursor proof-of-concept experiments for high-frequency gravitational waves. *American Institute of Physics Conference Proceedings*, page 3, 2003.
 - [14] P. C. Peters. Gravitational radiation and the motion of two point masses. *Physical Review*, 136(4B):B1224–B1232, 1964.
 - [15] Stanford University. Space and time in einstein’s universe. <https://einstein.stanford.edu/SPACETIME/spacetime2.html#:~:text=Space%20and%20time%20in%20Einstein%27s,stretched%20and%20warped%20by%20matter.>, n.d. Accessed: 2025-02-16.
 - [16] Hans Stephani, Dietrich Kramer, Malcolm MacCallum, Cornelius Hoenselaers, and Eduard Herlt. *Exact Solutions of Einstein’s Field Equations*. Cambridge University Press, Cambridge, 2nd edition, 2003. See pp. 33–34.
 - [17] Jinye Yang. Post-newtonian theory. Online, 2020. Course Report, University of Florida.