Computing Methods for Physics – 14 June 2022

Your exam material (code files, plots, datafiles, etc.) must be submitted via google classroom by 13:00 as a single `zip` file.

C++ evaluation will be based on: correct syntax, proper return types, proper arguments of functions, data members and class interfaces, comments throughout the code, separation of class implementations and interfaces.

Python evaluation will be based on: correct syntax, avoiding C-style loops, using Python features in general, comments throughout the notebook/scripts, labels, legends and plot styling and clarity in general.

Part 1 – Binary Black Holes in C++

Implement a C++ class to represent black holes. A black hole is fully described by its gravitational mass m, which we will handle as a multiple of the solar mass, and its dimensionless rotational angular momentum parameter $\chi \in [0, 1]$. Given these two parameters, one can define the irreducible mass of a black hole,

$$m_{\mathrm{irr}}^2 = \frac{m^2}{2}\left[1 + \sqrt{1-\chi^2}\right] ,$$

and the energy $E_{\mathrm{Penrose}} = m - m_{\mathrm{irr}}$ that may be extracted from it. Your `BH` class must have:

- Constructors (when the user only provides m assume $\chi = 0$, when the user only provides χ assume $m = 1$); the user should be allowed to construct a black hole using any two parameters in the set $\{m, m_{\mathrm{irr}}, \chi\}$.
- A copy constructor.
- A `Print()` method that informs about the values of $\{m, m_{\mathrm{irr}}, \chi, E_{\mathrm{Penrose}}\}$.
- Setter and getter methods.

Build a second C++ class capable of representing a binary black hole system. This must be able to handle a pair of `BH` instances. The requirements for the `BBH` class are the following:

- It must include a copy constructor.
- It must include a constructor that takes as arguments two gravitational masses, two dimensionless rotational angular momentum parameters, and two angles θ_1 and θ_2. The angles are between 0 and π. There is no request to implement other constructors.
- It must have an attribute that represents the effective rotational angular momentum

of the binary black hole. This is defined as

$$\chi_{\rm eff} = \frac{\chi_1 m_1 \cos\theta_1 + \chi_2 m_2 \cos\theta_2}{m_1 + m_2}\,.$$

- It must have a `Print()` method that informs about the individual black holes as well as $\{\chi_{\rm eff}, \theta_1, \theta_2\}$.

Your submitted material must include a file `app.cpp` that showcases the classes you implemented.

Part 2 – Binary Black Holes in Python

Use a Python notebook or Python scripts to complete the following tasks **with dictionaries**.

1. Draw $N = 10^4$ random values of $\{m_1, \chi_1, m_2, \chi_2, \theta_1, \theta_2\}$, which have the same meaning as in the previous sections. Draw the χ_i's uniformly, the θ_i's uniformly in their cosine, and the m_i's uniformly between 10 and 100 (solar masses).

2. Calculate and store the values of: $\chi_{\rm eff}$, the total mass $M = m_1 + m_2$, the symmetric mass ratio

   $$\eta = \frac{1}{4}\frac{m_1 m_2}{M^2}\,,$$

   the chirp mass

   $$\mathcal{M} = \frac{(m_1 m_2)^{3/5}}{M^{1/5}}\,,$$

   and the time spent (in seconds) by the binary black hole in the sensitivity band of current gravitational-wave detectors:

   $$t_c = \frac{50}{256}\frac{r_0^4}{m_1 m_2 M}\nu_{10} \qquad \text{where} \qquad r_0 = \left(\frac{M}{\pi^2 \nu_{10}^2}\right)^{1/3} \qquad \text{and} \qquad \nu_{10} = 4.916\cdot 10^{-5}\,.$$

3. Plot the distributions of these 5 quantities.

4. Use `scipy.optimize.curve_fit` to fit the $\chi_{\rm eff}$ data with a Lorentzian Function

   $$L(x) = \frac{1}{\pi}\frac{\Gamma}{(x - x_0)^2 + \Gamma^2}\,.$$

   Plot your data and your fit.