Gravitational-Wave Course — Homework Sheet 5

Instructor: Alessandra Buonanno (alessandra.buonanno@hu-berlin.de)
Tutor: Raj Patil (raj.patil@aei.mpg.de)
Tutor*(corresponding for this sheet):* Lorenzo Pompili (lorenzo.pompili@aei.mpg.de)

Course webpage: https://imprs-gw-lectures.aei.mpg.de/2022-gravitational-waves/

Homework due date: Homeworks must be emailed by Monday, December 5 2022 to the corresponding Tutor for this homework.
Homework rules: Homeworks must be neat, and must either be typed or written in pen (not pencil!). Please do not turn in homework that is messy or that has anything that's been erased and written over (or written over without erasing), making it harder to read.

Grading system: The homework sheet will be graded with an overall score within *0, 1, 2*.

0: not sufficient, the student has done less than half of the problems and did not attempt all of them.

1: sufficient, the student has done more than half of the problems and she/he tried to solve almost all of them.

2: good, the student correctly solved almost all the problems.

RECOMMENDED READINGS:

1. Gravitational waveform catalogs:
   - Boyle et al, CQG **36** 195004 (2019), https://arxiv.org/abs/1904.04831
2. Black hole kicks
   - Gonzalez et al, PRL **98** 091101 (2007), https://arxiv.org/abs/gr-qc/0610154
   - Ruiz et al, GRG **40** 1705 (2008), https://arxiv.org/abs/0707.4654

EXERCISES:

Instructions to install a working environment (Jupyter Lab) for Python are at the end of the problem sheet. For people not familiar with Python, Lorenzo and Raj would be available to arrange a session to describe what is needed to do the exercise.

I. BLACK HOLE KICKS

The goal of this exercise is to gain some familiarity with numerical relativity waveforms. We use the NR catalog of the Simulating eXtreme Spacetimes collaboration (SXS), which is described in detail in Boyle et al (2019). To have a concrete goal, let us calculate the kick-velocity of the remnant black hole in mergers of non-spinning black holes, as presented in Gonzalez et al (2007).
Please develop and hand in an Jupyter python notebook covering the following tasks.

1. *Finding & obtaining simulations*

   The most convenient means to interact with the SXS catalog is through the python package `sxs`, about which you can learn here:

   - documentation: https://sxs.readthedocs.io/en/main/

- tutorials: `https://mybinder.org/v2/gh/moble/sxs_notebooks/master`

Install `sxs` as described in the documentation (this is often as easy as `pip install sxs` in a terminal). We will use the SXS simulations with the numbers 1143, 1222, 1178, 1220, 192, and 1108.

Download these six simulations with the following code snippets:

```
import sxs

# enable caching
sxs.write_config(download=True, cache=True)

# download metadata
catalog = sxs.load("catalog")

# list of simulations to work with
IDs=['SXS:BBH:1143','SXS:BBH:1222','SXS:BBH:1178',
     'SXS:BBH:1220','SXS:BBH:0192','SXS:BBH:1108']

# download waveforms into 'wf'
wf={}
for k in IDs:
    wf[k]=sxs.load(k+"/Lev/rhOverM", extrapolation_order=2)
```

Once the waveforms are downloaded, the following python code illustrates how the relevant data can be accessed from '`wf`' and '`catalog`':

```
# illustrate how to use the waveforms from the catalog:
for sxs_id in wf:
    print("sxs_id=",sxs_id)

    # retrieve metadata
    sim=catalog.simulations[sxs_id]
    q = sim['reference_mass_ratio']
    mf = sim['remnant_mass']
    print(f"    mass-ratio {q},   remnant mass {mf}")

    # retrieve time and Re(h22) mode
    # the surrounding 'np.array' makes error messages
    # more familiar should errors occur in your code
    w = wf[sxs_id]
    t=np.array(w.t)
    l=2; m=2;
    Reh22=np.array(np.real(w[:,w.index(l,m)]))

    # print some info about waveform:
    tstart=t[0]
    tend=t[-1] # time at end of waveform
    print(f"    time range covered by waveform: [{tstart}, {tend}]")
```

Add this second code snippet into your output notebook and execute.

2. *Some practice with GW modes*

Gravitational waveforms are provided as modes $h^{l,m}(t)$ in a spin-weighted spherical harmonic expansion (Eq. (11) of Boyle et al),

$$h_+(\theta,\phi,t) - ih_\times(\theta,\phi,t) = \sum_{l,m} h^{l,m}(t)\,{}^{-2}Y_{l,m}(\theta,\phi), \tag{1}$$

where the modes $h^{l,m}(t)$ provide the time-dependence of the data, whereas the spherical harmonics with spin-weight -2 encode the spatial structure of the emission. If you don't know about spin-weights yet, don't worry – knowledge of them is not needed for the exercise. [1]

For *one* of your simulations, plot the $(l,m) = (2,2), (3,3), (4,4)$ modes as a function of time. Make **three plots**, one showing the entire time, one zooming into the early evolution, $t < 500$, and one showing the time-interval around merger and ringdown. You will notice that at early times, the NR waveforms have extraneous features often referred to as 'junk radiation'. Also, the (2,2) mode is dominant. Also, near merger higher-order modes are more pronounced than in the early inspiral.

For *all* your simulations, compute the maximum of the amplitude of the (2,2), (3,3), and (4,4) modes (i.e. three real numbers for each waveform). Plot max $A^{l,m}$ vs mass-ratio q for your simulations.

Your plot should show that (i) $\max A^{2,2}$ decreases with increasing mass-ratio. (ii) the higher modes become more important relative to (2,2) at increasing mass-ratio.

Demonstrate (through a plot or otherwise), that $\max A^{2,2}$ decreases with mass-ratio approximately proportional to the *symmetric mass-ratio* $\nu = q/(1+q)^2$.

[1] Also, if you wonder about the decay of the amplitude of gravitational waves with the distance to the source: The overall distance dependence $1/r$ is already factored out of the $h^{l,m}$ modes. Therefore, to obtain the correct magnitude of a mode at a distance r from the source, one would have to multiply Eq. (1) by M/r. This factor $1/r$ is also not needed in this problem.

3. *Black hole kicks*

   Now we are ready to embark on the main goal of this exercise, to compute the velocity imparted on the remnant black hole during a BBH merger. Black hole kicks arise through asymmetric emission of linear momentum in the gravitational waves. Therefore, we need to compute the momentum flux $d\vec{P}/dt$ carried in gravitational waves.

   The binaries we are interested in are non-spinning and have a $z \to -z$ symmetry. This implies that GW emission above and below the equatorial plane is equal so that the z-component of the kick will vanish, and that we only need to compute the linear momentum emission tangential to the orbital plane.

   Moreover, in order to utilize the GW modes $h^{l,m}$, we need the GW linear momentum flux expressed in terms of the $h^{l,m}(t)$ modes. This calculation is performed in Ruiz et al (see specifically the passage from Eq.(1.1) to Eq.(1.11), and from Eq. (3.5) to Eq. (3.17), using the equation numbering from the arXiv-version). Specifically, Eq. (3.14) of Ruiz gives the relevant formula. This formula is given in terms of the Newman-Penrose scalar Ψ_4 whereas we have modes of the gravitational wave strain h. The two are related by two time-derivatives, as indicated by Eq. (3.3) in Ruiz et al. Because we already have h (which is the double-time-integral of Ψ_4), there is no need for the time-integrations in Eq. (3.14), and the formula becomes

$$\frac{dP_+}{dt} = \frac{dP_x}{dt} + i\frac{P_y}{dt} = \frac{1}{8\pi}\sum_{l,m} \dot{h}^{l,m}\left[a_{l,m}\left(\dot{h}^{l,m+1}\right)^* + b_{l,-m}\left(\dot{h}^{l-1,m+1}\right)^* - b_{l+1,m+1}\left(\dot{h}^{l+1,m+1}\right)^*\right]. \quad (2)$$

   Note that this equation contains first time-derivatives of the $h^{l,m}$ modes, as well as complex conjugates indicated with the star. The coefficients $a_{l,m}$ and $b_{l,m}$ are given in Ruiz et al.

   Double-check Eq. (2) against Eq. (3.14) in Ruiz et al, to verify that all indices are correctly transcribed into this problem set. The overall factor $1/r^2$ in Ruiz et al is already absorbed in the NR definition of the h^{lm} modes, and therefore does not appear in Eq. (2).

   The time-derivatives $\dot{h}^{l,m}$ can be computed with various methods (e.g. finite-differencing in time). But since the `sxs` package already contains functionality to compute the time-derivative, we will just use this: If `w` is a waveform-object downloaded with `sxs.load`, then `wdot=w.dot` is the corresponding waveform object holding the first time-derivatives.

   For *one* of your simulations, plot as a function of time $h^{2,2}$ and $\dot{h}^{2,2}$ on the same plot. Convince yourself by visual inspection that $\dot{h}^{2,2}$ is indeed the time-derivative of $h^{2,2}$.

   Now we are ready for the main task of this problem: **Implement Eq. (2) as a python function that computes the GW linear momentum flux dP_+/dt as a time-series, given a waveform object.** When implementing this function be aware of the following:

   (a) SXS-waveforms only have modes up to $l <= 8$; be sure you do not access modes with $l > 8$.

   (b) Spherical harmonics modes satisfy $|m| \leq l$. Set all terms in Eq. (2) to zero, which have $|m| > l$.

   (c) Pay attention to the placement of the square-roots in the definitions of the coefficients $a_{l,m}$ and $b_{l,m}$ in Eqs. (3.16) and (3.17) of Ruiz et al.

   dP_+/dt is the linear momentum flux; we need to integrate this to arrive at the total linear momentum emitted in gravitational waves. When doing this integral, however, we have to be careful with the transient features at the start of the waveform (which appeared in your plot of $h^{2,2}$) above). As a first step, let us consider the linear momentum imparted *up to* time t:

For *one* of your simulations: Evaluate dP_+/dt with your function, and compute a cummulative integral of the emitted linear momentum up to time t, as a function of time t:

$$P_+(t) = \int_{t_{\rm start}}^{t} \frac{dP_+(t')}{dt'}\, dt', \tag{3}$$

where $t_{\rm start}$ is the start of the data-set [2]. **Plot $P_x(t) = \mathrm{Re}(P_+(t))$ and $P_y(t) = \mathrm{Im}(P_+(t))$ vs. time t.** This plot will show a significant change in the emitted P_x and P_y near merger, which is the kick we are looking for. However, the plot will also show an unphysical change in $P_{x/y}$ at the start of the data. This change in P_{xy} near the start of the data arises from the junk-radiation that is always present in a NR simulation. This part must be excluded in order to get the correct result. Because the SXS simulations are very long, and because there is little accumulation of P_{xy} during the inspiral, we can simply set any time reasonably far after the junk-radiation to compute the physical BH kicks. For concreteness, let's use t=1000.

For all your simulations, compute the cummulative linear momentum emitted from $t =$ 1000 to the end of the waveform:

$$P_x + iP_y = \int_{1000}^{t_{\rm end}} \frac{dP_+(t')}{dt'}\, dt'. \tag{4}$$

The emitted linear momentum must equal the linear momentum $v_{\rm kick} m_f$ of the remnant black hole, which has mass m_f. Therefore, the remnant kick velocity is given by

$$v_{\rm kick} = \frac{\sqrt{P_x^2 + P_y^2}}{m_f}. \tag{5}$$

Evaluate Eq. (5) for your simulations and plot $v_{\rm kick}$ vs. q. What is the maximum kick-velocity for non-spinning BBH, and for what mass-ratio is it obtained? Compare your result to Fig. 2 of Gonzalez et al.

[2] Hint: python's function `numpy.cumsum` can be very useful here.

INSTALLING A WORKING ENVIRONMENT (JUPYTER LAB) FOR PYTHON

1. Install the **Anaconda Packet Manager** on your laptop (https://www.anaconda.com/distribution/) Please keep in mind that the Conda-Setup has to fit your operating system (LIN/WIN/MAC - 32bit/64bit). Please install Python only in Version 3 and fill all checkboxes.

2. The command `conda - -help` should be available in a Terminal/CMD/BASH after the setup. Create a **Virtual Environment** within conda which will be your working environment. The command is: `conda create -n gwcourse2022 python numpy scipy matplotlib seaborn jupyterlab`

3. Now activate your working environment with the command: `conda activate gwcourse2022`. When active, the beginning of the line will change e.g. `(gwcourse2022) Laptop-xyz:...` where (`gwcourse2022`) is the name of the working environment.

4. In the virtual environment we can now use PIP (the package installer for Python) to install more software, command: `pip install - - upgrade pip`. For this problem you should install the `sxs` package with: `pip install sxs`.

5. Test your setup after the installation: start a jupyter lab instance (type: `jupyter lab` in the terminal) and open a new Python3 notebook. Go to the first cell and type `import numpy as np` confirm by pressing the SHIFT+RETURN Keys. Make sure that these imports work without errors.

6. To end the working environment: **Save** and close the Browser-Tab of JupyterLab, in the CMD/Terminal/BASH press the CTRL+c Keys and confirm with Y. Then deactivate the virtual environment by: `source deactivate` or `conda deactivate`. To start working again go back to step 3) (activating your working environment).