DPT/Talent school: Thursday, 01 August 2024

Albino Perego

01 August 2024

1 Exercise 1: conditions to produce lanthanides and/or actinides in r-process nucleosynthesis

In this exercise, we want to find the conditions for which a fluid element expelled by a compact binary merger will produce lanthanides or actanides though r-process nucleosynthesis.

To do that, we will consider the outcome of a large set of nuclear network calculations (performed with the SkyNet code, Lippuner & Roberts ApJS 2017). In particular, we considered a 3D grid in the (τ, s, Y_e) space, where

- τ, expansion timescale in ms, $\tau \in [0.5, 200]$;
- s, entropy per baryon in k_{B} baryon^{-1}, $s \in [1.5, 300]$;
- Y_e, electron fraction, $Y_e \in [0.01, 0.48]$.

The provided script (`NN_grid.py`) is able to read both the mass fractions and the abundances on this 3D grid, as a function of the mass number, A, or of the atomic number, Z. The user can

- select one or more Z's or A's to select some specific fractions or abundances;
- select to fix one of the three variables in the 3D parameter space;
- visualize the abundances or fractions as a function of the other two variables.

To do that,

- go to the `ex1` folder;
- open the file `NN_grid.py` and modify the parts that are contained inside the lines of ===

Goal: find the conditions in the τ, s and Y_e space for which lanthanides and/or actanides can be significantly produced.

Many thanks to Leonardo Chiesa, Paolo Somenzi and to the Core Database for providing the data and the script for this exercise!

2 Exercise 2: Abundances for a BNS simulation

In this exercise, we want to compute the abundances as produced by a BNS merger. To do that, it is necessary to make the convolution of the outcome of nuclear network calculations (see exercise 1) together with histograms of the ejecta properites.

We will first analyse histograms of the ejecta properties, as recorded on a sphere located at a fixed radius (usually, $\sim 300km$ from the remnant). By running `NR_histo.py`, one can read the ejecta histograms. These histograms have been computed in the (τ, s, Y_e) space and they retain information about the polar angle θ (while the dependence on the azimuthal angle was integrated out).

In practice, the space (τ, s, Y_e) is broken in $N_\tau \times N_s \times N_{Y_e}$ cubes, characterized by bins of central values τ_i, s_j and $Y_{e,k}$ with $i \in [1, N_\tau]$ and $j \in [1, N_s]$ and $k \in [1, N_{Y_e}]$. The quantity contained in the histogram is the mass of the ejecta whose properties lay in the cube.

The goal of the first part of the exercise is to compute the average value of τ, s and Y_e in the ejecta, as a function of the polar angle, θ. Use the script `NT_histo.py`. At the moment, it simply read the histograms and the grids. One can change the ejecta extraction criteria:

- `geo`: geodesic criterion;
- `bern`: Bernuolli criterion.

as well as the simulation name:

- `DD2_M13641364_M0_SR`;
- `SFHo_M11461635_M0_LK_SR`.

`geo` is a robust criterion to extract dynamical ejecta; `bern` is a "generous" criterion to extract dynamical and wind ejecta. My suggestion is: fix one of the variables and marginalize over the others (but not over θ), then compute the average as a function of θ.

In the second part of the exercise, we will compute the abundances as they come out from a simulation, by the convolution of the nuclear network calculation and of the ejecta properties. To do that, use the file `BNS_nucleo.py`. Specify the ejecta extraction criterion and the simulation. The script plots the abundances as a function of A, for example, for a user-selected angle. **The goal of the second part of the exercise is to plot the abundances at different polar angles, θ, for different simulations and for different extraction cirteria.**

Many thanks to Leonardo Chiesa, Paolo Somenzi and to the Core Database for providing the data and the script for this exercise!

3 Exercise 3: kilonova light curves

In this exercise, we want to compute kilonova light curves and compare them with data from AT2017gfo, the kilonova associated to GW170817. To do that, we will use `xkn`, a kilonova framework mostly developed by a group of people from Trento, Jena and Darmstadt. The framework was recently documented in this paper https://ui.adsabs.harvard.edu/abs/2024MNRAS.529..647R/abstract from G. Ricigliano and collaborators.

The framework allows the modeling of kilonova emission from multicomponent, anisotropic ejecta, assuming axisymmetry and discretizing the polar angle in a ray-by-ray fashion.

We will model 3 different ejecta components: dynamical ejecta, wind ejecta and viscous/secular ejecta.

- for the dynamical ejecta, we will consider an anisotropic distribution in the mass ($\propto \sin\theta$) and in the photon opacity (κ_γ, a step function separated by $\theta = \pi/4$), and an isotropic average speed;
- for the wind ejecta, we will consider an anisotropic distribution in the mass ($\propto \sin\theta$) and in the photon opacity (a step function separated by $\theta = \pi/4$), and an isotropic average speed;
- for the viscous/secular ejecta, we will consider an isotropic distribution in the mass, in the photon opacity and in the average speed.

For model consistency, one should make sure that

$$v_{\mathrm{sec}} \lesssim v_{\mathrm{wind}} \lesssim v_{\mathrm{dyn}}$$

Moreover, for the opacity,

- if there are no lanthanides, $\kappa_\gamma \in [0.1, 2]\mathrm{cm}^2/\mathrm{g}$
- if there are a few lanthanides, $\kappa_\gamma \in [2, 10]\mathrm{cm}^2/\mathrm{g}$
- if there are a lot of lanthanides, $\kappa_\gamma \in [10, 50]\mathrm{cm}^2/\mathrm{g}$

To solve the exercise,

- clone or download the xkn code from here:
  https://github.com/GiacomoRicigliano/xkn
- the code is in `python3` and it just requires the following pakages to be installed:
  - astropy;
  - matplotlib;
  - mpmath;
  - numpy;

  - scipy;
  - numba.

  Make sure to have all of them, otherwise install them using
  `pip install astropy matplotlib mpmath numpy scipy numba`

- move into the code folder (where the `setup.py` file is located) and install it:
  `pip install .`

- remove the original files `example.py` and `kn_config.py` inside the folder `examples`, and replace them with the ones you can find in the exercise folder;

- do not modify the `kn_config.py` file, while edit the `example.py` file. In particular, you have to provide 13 quantities:

  - the viewing angle, as an angle $\in [0, \pi/2]$;
  - the kilonova distance in Mpc;
  - the mass of the dynamical ejecta, in $M_\odot$;
  - the average speed of the dynamical ejecta, in unit of c;
  - the photon opacity of the dynamical ejecta at high latitudes in cm^2/g;
  - the photon opacity of the dynamical ejecta at low latitudes in cm^2/g;
  - the mass of the secular ejecta, in $M_\odot$;
  - the average speed of the secular ejecta, in unit of c;
  - the photon opacity of the secular ejecta in cm^2/g;
  - the mass of the wind ejecta, in $M_\odot$;
  - the average speed of the wind ejecta, in unit of c;
  - the photon opacity of the wind ejecta at high latitudes in cm^2/g;
  - the photon opacity of the wind ejecta at low latitudes in cm^2/g.

To run the code, simply type:
`python examples/example.py` and it will produce a pdf file with the light curves and the observed magnitudes. **Goal: provide a set of the 13 quantities that reproduce well the data from AT2017gfo**.