Progress Report on HD Extension Project

July 2023

Jose Arita-Escalante

1 Introduction

The Monte Carlo Radiative Transfer (MCRaT) code conducts photon scatterings within a hydrodynamically (HD) simulated Gamma-ray burst (GRB) jet to create mock observables from said jet.

The HD simulation has a spatial domain that is limited to the size of any given frame during the simulation. Its temporal domain is also limited to a given Δt. Therefore, for coordinates $(\vec{x}, t)$ that lay outside of the simulation domain, photons scatter without interacting with any matter from the jet.

The main goal of this project is to extend the HD simulation domain in space and time, such that the mock observables generated by MCRaT account for the parts of the jet that lay outside the photosphere, and outside of the simulation spatial and temporal domain.

2 Method

We make an assumption that the jet is in a homologous outflow, meaning the position and velocity unit vectors are parallel to each other. At this point, each jet shell scatters according to the differential equations used in Piran et al. (1993).

If each HD cell is in homologous outflow, their properties evolve according to

$$\begin{cases} \frac{\partial}{\partial r} r^2 n \Gamma = 0 \\ \frac{\partial}{\partial r} r^2 e^{3/4} \Gamma = 0 \\ \frac{\partial}{\partial r} r^2 \Gamma^2 (n + \frac{4}{3} e) = 0. \end{cases}$$

Under this assumption, we use the approximation used by Gottlieb et al. (2019), in which we assume each fluid element propagates forwards in time at speed c in the direction of its position vector.

We can normalize these quantities following the directives given by Gottlieb et al. (2019), such that $\tilde{r} = r/r_i$, $\tilde{n} = n/n_i = n/n(r_i)$, $\tilde{e} = e/e_i = e/e(r_i)$ and $\tilde{\Gamma} = \Gamma/\Gamma_i = \Gamma/\Gamma(r_i)$ such that

$$\begin{cases} \frac{\partial}{\partial \tilde{r}} \tilde{r}^2 \tilde{n} \tilde{\Gamma} = 0 \\ \frac{\partial}{\partial \tilde{r}} \tilde{r}^2 \tilde{e}^{3/4} \tilde{\Gamma} = 0 \\ \frac{\partial}{\partial \tilde{r}} \tilde{r}^2 \tilde{\Gamma}^2 (\tilde{n} + \frac{4}{3}\tilde{e}) = 0. \end{cases}$$

We are able to show how these variables behave for different values of initial enthalpy h_i, given by

$$h_i = 1 + \frac{4\tilde{e}}{3\tilde{n} m_p c^2}.$$

We then use the 4th order Runge-Kutta method to numerically solve this set of differential equations for initial conditions given by each HD cell in the last frame of the simulation, in which the jet is assumed to be in its homologous phase, up until any given radius or time. With this, we can propagate each cell forward in time, and calculate its HD properties at coordinates $(\vec{x}, t)$ outside of the simulation domain.

We use the yt Python library to read the last FLASH HD simulation of a GRB jet given by the 16TI model. Once this frame and its HD properties are loaded, we can use each grid element to determine which ones are in a homologous outflow.

PLUTO NUANCES

We tried using yt to read in a PLUTO file and repeat the same method for this different type of HD simulation file, but yt currently has a hard time reading the file, considering it has polar coordinates. yt tries to read it in as cartesian, and therefore, we can't retreive the right coordinates of each HD cell.

A potential workaround would be to use pyPLUTO or processMCRaT to try to do this, since both of these libraries can read a polar grid.

3 Notebook Order

The Jupyter Notebooks in this repository are named accordingly to what each one does, instead of what order should they be read in to catch up. Here, we present a quick overview of each notebook, along with what each one does and which order should they be read or analyzed in.

1. **recap_may_2023.ipynb**

   This notebook is the first one that should be read. Section 1 in the notebook walks through the homologous differential equations and the recreation of Figure A1 in Gottlieb et al. (2019).

   We then analyze the evolution of the HD grid shape, assuming all the cells scatter radially according to Section 3.2 in Gottlieb et al. (2019).

After taking a look at the grid shape evolution, we test out the homologous equations on an analytic spherical outflow grid.

2. **omega_test.ipynb**

   This notebook is the second one that should be read. Here, we discuss how in our spherical outflow case, in order to achieve homologous scattering of the HD cells, we used their position vector as the velocity vector. Clearly, this is only true for a spherical outflow case, and hence, our script should be updated. This notebook show the steps taken to update the script, such that we use the cells' velocity vector for the direction of their scattering instead of their position vector.

3. **adapting_homo_out_for_numpy.ipynb**

   This notebook is the third one that should be read. Up to this point, we had been using native Python and list comprehension to keep track of each grid element's properties. Although this works, this is a very inefficient process, as it not only takes a lot of memory, but also takes a long time (specially when dealing with thousands of elements, as one would see in a real HD frame). This notebook adapts all the functions used thus far to take full advantage of NumPy's capabilities. Doing this, we were able to cut our run time by half when running a solver on a spherical outflow frame.

4. **homologous_outflow_on_hd_frame.ipynb**

   This notebook is the fourth and last notebook thus far. In this notebook, we try our method shown in the previous notebooks on an HD frame. We use the last frame in a FLASH HD simulation to read in the properties of a GRB jet which is expected to be in its homologous outflow stage. We see that some velocity vectors have directions such that the new grid shape is not what we see in the spherical outflow case. Because of this, we need to identify which cells are part of the jet where the outflow is homologous.

4 Future Steps

We fully understand how the homologous outflow equations work for a shell in this outflow type. We understand how to numerically integrate the HD properties of these cells, assuming that the shell is moving at speed c.

The next step is to determine which cells are in their homologous expansion phase.

Following Dr. Gottlieb's method, even though the jet is never really homologous due to its stochastic nature, the second best thing to do is to find that the jet is in the fireball acceleration regime.

One way to check that is to plot the radial profiles of h (Γ), and see that the profile decreases (increases) linearly with r. Alternatively, we can also check that that $h \cdot \Gamma$ is constant.

Verifying any of these behaviors would indicate that mixing is no longer important at these radii, thus the fireball (homologous) equations can be applied to the outflow.

Once the part of the jet with a fireball expansion is located, these elements can be propagated homologously using our set of equations. This part of the jet is expected to be the one moving along with the MCRaT photons, so these will interact with the homologous HD medium, until the medium is optically thin enough for interactions (Compton scattering) to stop.

Once we understand how to implement this expansion to the fireball expansion part of the HD frame, we can start implementing it to MCRaT. This involved translating our Python work to C in such a way that MCRaT will be able to calculate the HD properties of these now evolved HD cells.

References

Gottlieb, O., Levinson, A., & Nakar, E. 2019, , 488, 1416, doi: `10.1093/mnras/stz1828`

Piran, T., Shemi, A., & Narayan, R. 1993, , 263, 861, doi: `10.1093/mnras/263.4.861`