

Black Hole Cluster with Blender

July 9, 2025

1 Generating the data

1.1

2 Before You Run

2.1 Setting up Blender

First, make sure you are using a supercomputer that has access to blender. In your `/.bashrc` file, you want to add something like this

```
export PATH=$PATH:/data/shared/blender-4.2.3-linux-x64/  
alias blender='/data/shared/blender-4.2.3-linux-x64/blender'
```

Now you should be able to run blender

2.2 Generated memory map

If you ever want to plot the memory effect, first make sure you have the memory data. In the VTK data you have, you should have two things:

1. SCALARS GW-FIELD float 1
2. SCALARS GW-MEM float 1

Run "createmap.py" if you want to plot *memory*, and run "createmap_r.py" if you want to plot $r * memory$. These codes will output a directory that contains the images of memory effect in each frame.

2.3 Check Data

There are several things you could plot:

1. Directory that contain memory map from 2.2
2. Directory that contain obj data for the plane. If you don't have this, run the following:
 - (a) "make_obj.py", if you just want to convert vtk to obj files, without cutting the hole
 - (b) "convert_vtk_to_obj.py", if the VTK data only has one set of data
 - (c) "convert_vtk_to_obj_mememory.py", if the VTK data contain more than one set of data
3. Directory that contains frames for the neutron star(or what ever you are plotting). This should has transparent background, and in the same view of blender. These frames are generated by `abid_bot` and `visit`, since we haven't find a way to make blender read hdf5 data. To make the background transparent, you want to have:

```
s = SaveWindowAttributes()  
s.format = s.PNG  
s.SetPixelData(2) <- this line
```

For the view, you want to do make the view as close as possible and make sure that it doesn't cut any of the parts of the neutron star. **Also make sure when you run the blender code, the size of the neutron star matches with cutoff radius!** (you might need to do some math and trails/errors.)

4. A file that contains the radius of the black hole. This is only used if the black hole data is missing and there isn't black holes in the frames made by abid_bot. The file can be created with "BH_diagnostics.ah1.gp", which contains the mean radius of the black holes. You want to first remove the gaps in this file, which are the headers and repeated lines, then run "generated_bh.py". The output data should be a file contains first column time and second column radius.
5. A blender file that contains a white plane. This is used because Stu think the shadow of the dark background is too dark, and making the background light will change the color of the plane. So the solution is put a white plane. **Also note, if you are making a gravitational wave movie for BH cluster, you want to have the cluster file here. The cluster file is the blender file you created with GUI for the cluster movie. Moreover, the size of the black holes should be modified since in the view here is farer away from the cluster movie.**

2.4 Color bar

The color bar are all generated in the GUI, and used node to python node to convert to python. In the "plot_single.py" or other plotting script you are using, use from "color_bar_file_name.py" import "color_bar_name" to import color bar, and assign the color bar to different objects. Currently for the memory effect, you should have a "change_color.py" file that contains all the color bar.

Moreover, if you are creating new color bar and the color bar code is not working, ask GPT. Usually it's because the index number doesn't match.

3 Run the Script

3.1 Make evolution movie

Pretty easy. Just run ".runSingle.sh". You want to **change the path in this file**. Also read the annotations since it should tell you how the flags work. The "DESIRED_FRAMES" is the frame you want to run, this number correspond to xxxx in hplus_00xxxx.obj.

If you want to run a full movie, you can do "runMulti.sh" or "runSingle.sh", but in the runSingle.sh, use

```
DESIRED_FRAMES=()
for i in $(seq -w 0 2 1914); do
    DESIRED_FRAMES+=("$i")
done
```

Personally speaking, I found runMulti(Also it's not up to date) doesn't increase the speed of the movie by a lot especially when other people are also using Riemann; thus, I suggest using runSingle majority of the times.

3.2 Make flyaround movie

If you want to make some type of flyaround, I would suggest download the blender file and do it in the GUI.

Yes, we do have a file called plot_flyaround.py, and it works similarly with plot_single.py. The only difference it has is that it animates the camera at two different time. By default it is 0 and 120, so 120 frames of flyaround. If you want to change this number, go to plot_flyaround.py and look for the section named flyaround. In runSingle.sh, you can replace plot_single with plot_flyaround. The problem of making flyaround with script is that the camera might look drunked. This is due to even though the endpoints looks good, the camera rotated 360 degree on the way to the final frame.

If you want to do it in GUI, turn the SAVE_BLENDER_FILE flag to 1, run the frame you want to rotate, download the blender file, and open in GUI. In the time bar = 0 in the GUI, click camera and press I to set a key frame for the camera. Rotate the camera and press I again at frame 120 or whichever frame you want as the final frame. Render the whole movie.

Some times, the movie takes a long time to render on your local. It might be helpful if you upload it to Riemann or other supercomputer that has blender first. Find the `render_all.py` in useful script, and run this. This python file will generate the all the frames in the blender file.

4 Debug

1. The neutron star are in rotated in different direction with gravity wave.

This is because that neutron star images generated by visit are viewed in different axis with the gravitational waves. For example, the things that look rotating clockwise on $+y$ -axis will look rotating counterclockwise on the $-y$ -axis.

To solve the problem, you can either reflect the gravitational wave plane or change the view.

2. You don't know what went wrong but it's not working

I would suggest turn the `SAVE_BLENDER_FILE` flag to 1, so it saves a blender file of the scene in the directory of the output images. Then you can download the file and take a look with GUI