

# SPIN 2023 Exercise

## Project

This exercise aims to provide a flavor of the expertise required for the SPIN project as well as giving you an idea of the work expected of you. If you choose to apply to work with me on this project then I will ask you to work through the exercise and send in your results before applying. I will **not** consider you for the project unless I have received the exercise results. Please keep in mind that the exercise mostly serves to give **you** an idea of the skills required to avoid signing up for more than you can handle.

Modern scientific simulations have enabled us to study non-linear phenomena that are impossible to study otherwise. Among the most challenging problems is the study of Einstein's theory of relativity which predicts the existence of gravitational waves detected very recently by the LIGO collaboration. The [Einstein Toolkit](#) is a community driven framework for astrophysical simulations. I am interested in recruiting a student interested in improving the quality of gravitational waveform templates describing colliding black holes produced with the in the [Einstein Toolkit](#).

[CarpetX](#), a new driver for the Einstein Toolkit based on AMReX, is now available for testing. A driver in the Einstein Toolkit is responsible for basic computational algorithms such as adaptive mesh refinement, parallelism, inter-processes communication, or GPU offloading. Thorns can then implement the physics bits and discretization methods, relying on the driver to stitch everything together into a single application. CarpetX offers a range of new features for the Einstein Toolkit that are interesting for hydrodynamics or magnetic fields (staggered grids, refluxing), improved performance (multi-threading, GPUs, scalability, I/O), and additional safety features that prevent or catch programming errors (uninitialized grid points, inconsistent definitions). It uses the [openPMD](#) file format for 3D output during simulation and checkpointing.

This project will be to implement a reader plugin for the 3D visualization package [Visit](#) for use with Carpet and openPMD.

The successful applicant will be involved with both the [Gravity Group](#) at NCSA and will be invited to participate in the weekly group meetings and discussions of their research projects.

Preferred skills:

- Familiarity with Linux, git, make
- Strong working knowledge of C++ in application development and using STL. No experience using OpenGL or 3D rendering is required.
- Working knowledge of Python, matplotlib

Helpful Material (though none is required for the exercise):

- [Nano](#) tutorial
- [Visit Python](#) user guide and [tutorial](#)
- unofficial [Docker overview](#)

## Prerequisites

### Setup

Install Docker using either your package manager (eg `sudo apt-get install docker.io` on Ubuntu / Debian) or following the instructions on <https://docs.docker.com/get-docker/>. Alternatively, you can try and directly install [Visit](#), [demo code](#) and the required packages onto your system (Linux will likely be the only one you succeed), by following what the [Dockerfile](#) does.

Pull the exercise docker container image and create a container:

```
sudo docker pull rhaas/spin2023:20230111v0
sudo docker create --name spin2023 rhaas/spin2023:20230111v0
```

### Check your setup

Make sure that you can run Visit in the container by first starting the container:

```
sudo docker start spin2023
```

and getting a shell in it

```
sudo docker exec -ti spin2023 /bin/bash -i
```

then running visit in it

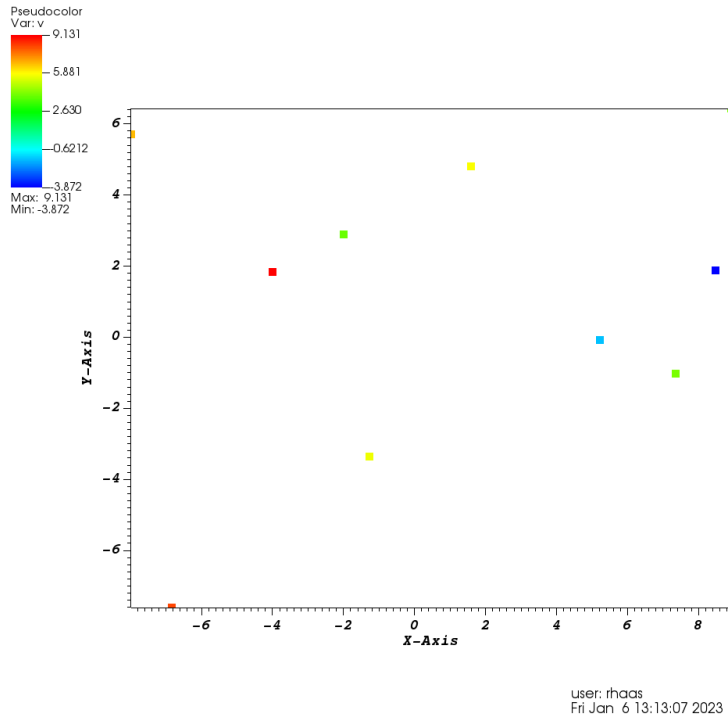
```
cd XYV
/home/visit/visit3_3_1.linux-x86_64/bin/visit -cli -nowin -s plot_XYV.py
```

Make sure you can copy files out of the container using:

```
sudo docker cp spin2023:/home/visit/XYV/data.png ./data.png
```

and that the image data.png looks like this:

DB: data.xyv



## Close down docker

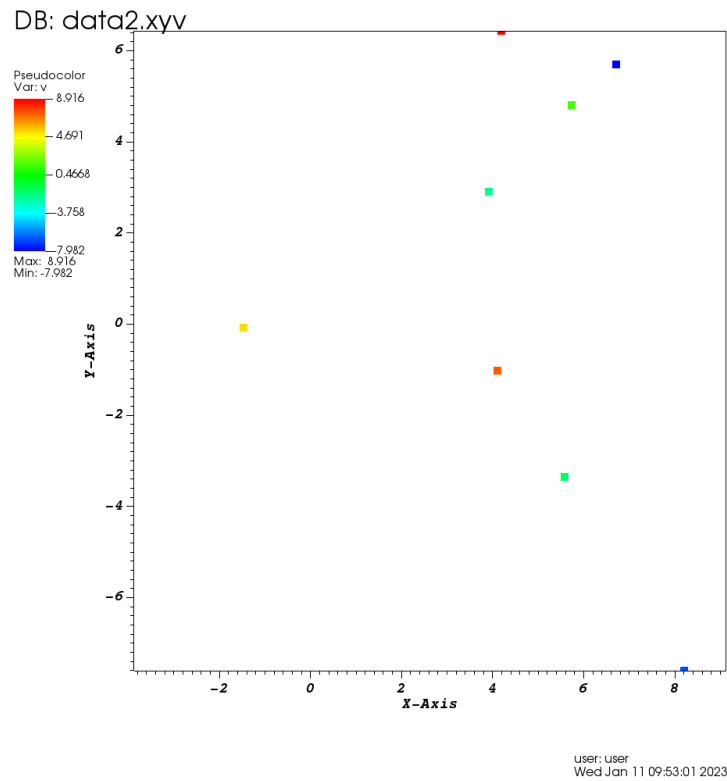
Finally stop the container

```
sudo docker stop spin2023
```

## Exercise

- Start and log in to the container as above
- Edit the file `XYV/avtXYVFileFormat.C` such that instead of using space separated files like in `data.xyv` it accepts comma separated files `data2.xyv`.
- There are `vim`, `emacs`, and `nano` editors present. If you are unfamiliar with all of them, `nano` will be the easiest one to use since it shows all commands at the bottom of the screen.
- Once you are done editing, recompile the code using `make`.
- Then run the script `plot_XYV2.py` to produce (if all goes well) `data2.png`
- Finally copy `avtXYVFileFormat.C` and `data2.png` out of the container and send me (rhaas@illinois.edu) both of them as an attachment in an email.

- The file `avtXYVFileFormat-sol.C` contains the final solution (a 3 character change). Your produced image file should look like this:



## Commands required

```
sudo docker start spin2023
sudo docker exec -ti spin2023 /bin/bash -i
cd XYV
nano avtXYVFileFormat.C
make
/home/visit/visit3_3_1.linux-x86_64/bin/visit -cli -nowin -s plot_XYV2.py
sudo docker cp spin2023:/home/visit/XYV/data2.png ./data2.png
sudo docker cp spin2023:/home/visit/XYV/avtXYVFileFormat.C ./avtXYVFileFormat.C
sudo docker stop spin2023
```

## Cleaning up

You can remove the docker image and container using:

```
sudo docker rm spin2023
sudo docker rmi rhaas/spin2023:20230111v0
sudo docker rmi debian:bullseye
```

## Sending in results

Once done, please send me the code (`avtXYVFileFormat.C`) and the output produced (`data2.png`) to [rhaas@illinois.edu](mailto:rhaas@illinois.edu) as an email attachment

## Solution

[avtXYVFileFormat-sol.C](#)

[data2.png](#)