Visualization toolkit: yt

Hsi-Yu Schive & Kuo-Chuan Pan

Introduction to yt

- Visualization and analysis toolkit for 3D data
- Support a variety of simulation codes
  - E.g., Athena, Pluto, Enzo, FLASH, Gadget, GAMER
  - Same analysis script, different simulation codes → allow you to forget about the data format and grids of your simulations
- Python- and script-based
  - Make your scientific results reproducible and shareable
- Unit-aware calculations
  - Otherwise unit conversion can be a headache…
- Very active and friendly community
- Open source

Useful Links

- Official website: https://yt-project.org/
- Source on GitHub: https://github.com/yt-project/yt
- Mailing list
  - User: https://mail.python.org/mailman3/lists/yt-users.python.org/
  - Developers: https://mail.python.org/mailman3/lists/yt-dev.python.org/
- Slack: https://yt-project.org/slack.html
- Installation: https://yt-project.org/#getyt
- Sample data: https://yt-project.org/data/

Other Features

- Support non-Cartesian coordinates
- Parallelization
  - Analyze multiple datasets in parallel
  - https://yt-project.org/docs/dev/analyzing/parallel_computation.html
- Extensions for connecting to external packages
  - E.g., pyXSIM (mock X-ray observations), Trident (mock absorption spectra), ytree (halo merger tree), unyt (unit manipulation in python)
  - https://yt-project.org/extensions.html
- yt hub: host and share your data
  - https://yt-project.org/docs/dev/sharing_data.html

Setup your visualization environment

- Request an interactive job

```
qsub -I -X -N name -l nodes=1:ppn=1,pmem=2gb,walltime=1:00:00
```

- Activate your python environment for yt

```
conda activate yt
```

Slice plot

- One command to plot all data

```
yt plot -f density my_sim_hdf5_plt_cnt_*
```

- Make a movie with ffmpeg

```
ffmpeg -r 10 -pattern_type glob -i '*.png' -vcodec libx264
-s 782x662 -pix_fmt yuv420p movie_sedov.mp4
```

Command line tools

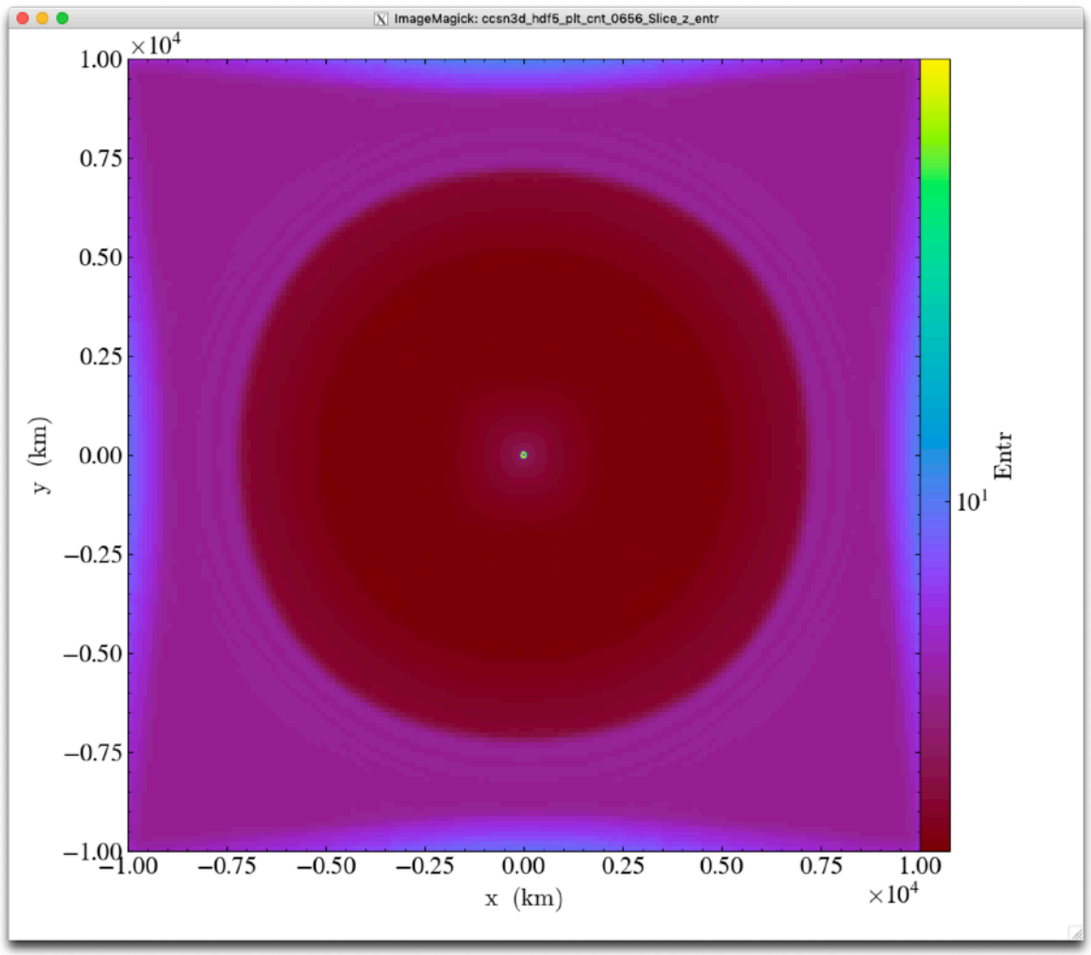
- yt plot -h

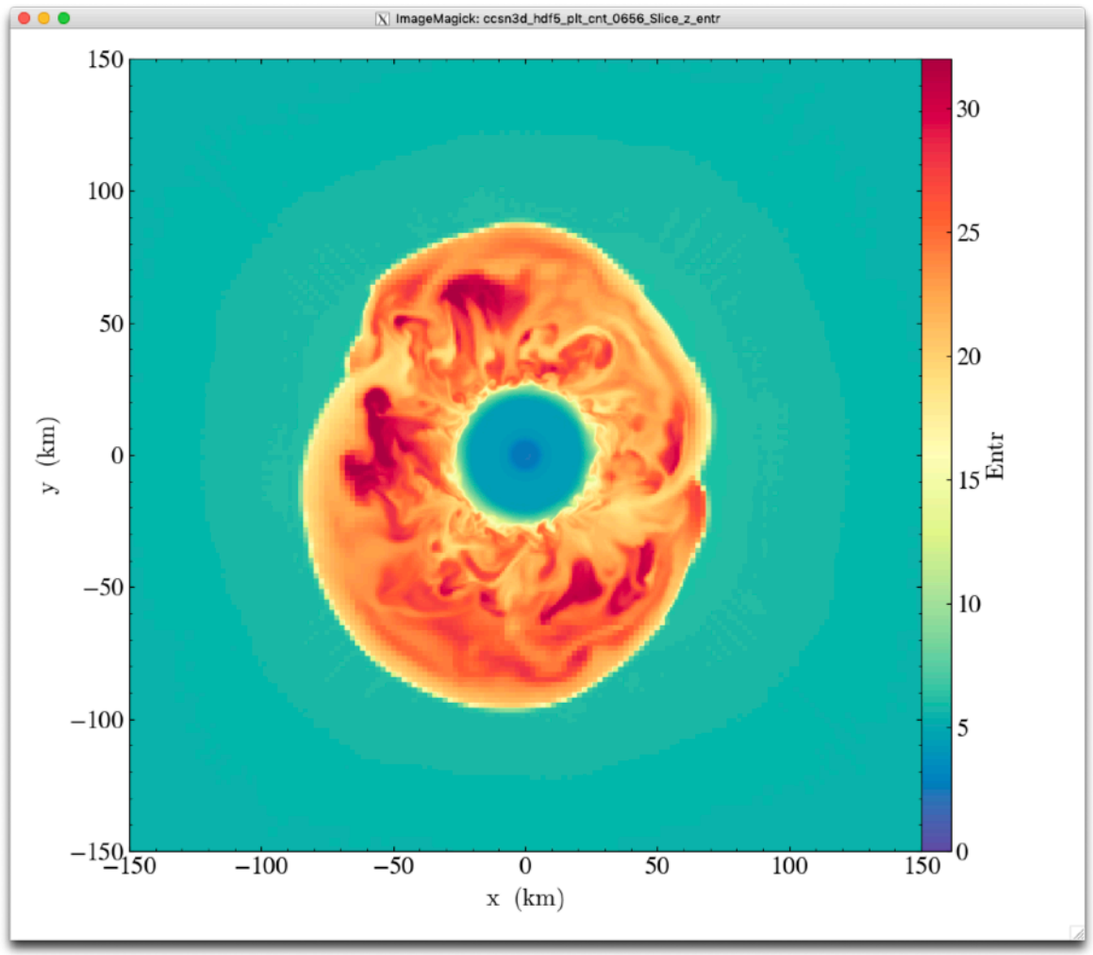
```
(yt) ➜  ~ yt plot -h
usage: yt plot [-h] [-w WIDTH] [-u UNIT] [-b BASENAME] [-p]
               [-c CENTER CENTER CENTER] [-z ZLIM ZLIM] [-a AXIS] [-f FIELD]
               [-g WEIGHT] [-s SKIP] [--colormap CMAP] [-o OUTPUT]
               [--show-grids] [--time] [-m] [-l] [--linear] [-fu FIELD_UNIT]
               [--show-scale-bar]
               ds [ds ...]

Create a set of images

positional arguments:
  ds                    datasets to run on
```

Customize the plot

- yt plot -h

```
yt plot -a 2 -f entr -w 300 -u km --linear --colormap Spectral_r -z 0 32
ccsn3d_hdf5_plt_cnt_0656
```

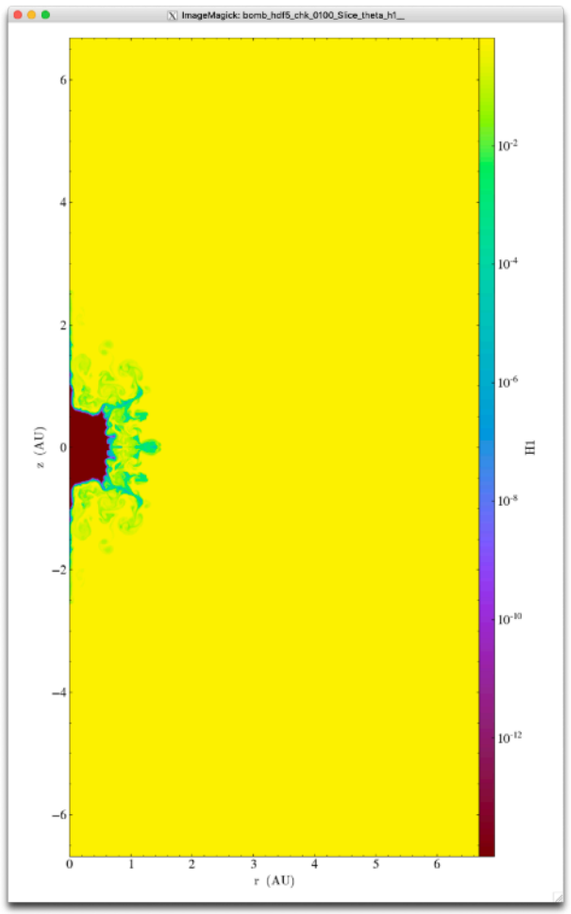


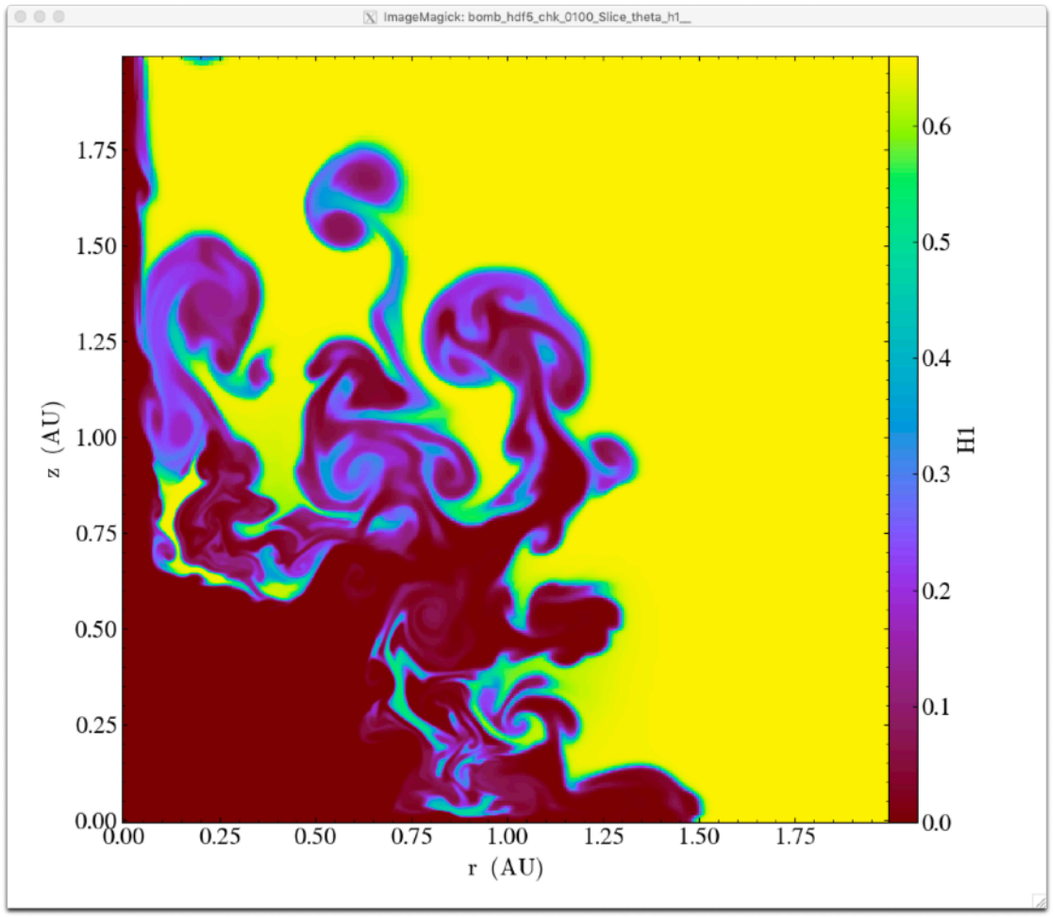


Customize the plot (conti.)

- Working with cylindrical data

```
yt plot -f "h1  " --linear -w 2 -u AU -c 1.49e13 1.49e13 0
bomb_hdf5_chk_0100
```





Import yt in a python script

```
import yt
import numpy as np

# set the root path
ROOT_PATH="./sample_data/"

# the file name for demo
fn = ROOT_PATH+"2d_cartesian/sedov_hdf5_chk_0010"

# Load the file to a dataset
ds = yt.load(fn)

# A simple slice plot
slice = yt.SlicePlot(ds,'z',"density")
slice.save("fig_tutorial_01.png")
```