Visualizing Einstein Toolkit Data with yt

Jonah M. Miller

with Erik Schnetter, Matthew Turk, and Kacper Kowalik

Perimeter Institute for Theoretical Physics
and the University of Guelph

Einstein Toolkit Workshop, 2015
Volume Rendering

http://yt-project.org/data/
import yt
source = "./ galaxy0030"
d = "density"
t = "temperature"
vm = "velocty_magnitude"
ds = yt.load(source)
ad = ds.all_data()
yt.PhasePlot(ad,d,t,vm)

http://yt-project.org/data/
With Python, yt is Easily Extendable

```python
# function defining
# new quantity
fname = "thermal_energy_density"
def therm_en_dens(field, data):
n = data['gas', 'number_density']
kT = data['gas', 'kT']
return (3/2)*n*kT

# add it to the dataset
ds.add_field(("gas", fname),
             units="erg/cm**3",
             function=therm_en_dens)

# plot
ad = ds.all_data()
yt.ProjectionPlot(...)
```

http://yt-project.org/data/
yt Extended for Discontinuous Galerkin Methods

Miller and Schnetter. In Prep.
\[ ds^2 = -H dt^2 + H dx^2 + dy^2 dz^2, \quad H = 1 - A \sin \left( \frac{2\pi (x - t)}{d} \right) \]
Extensions are Encouraged!

How To Read The Source Code

If you just want to look at the source code, you may already have it on your computer. If you build yt using the install script, the source is available at $YT_DEST/src/yt-hq. See Installing yt Using pip or from Source for more details about to obtain the yt source code if you did not build yt using the install script.

The root directory of the yt mercurial repository contains a number of subdirectories with different components of the code. Most of the yt source code is contained in the yt subdirectory. This directory itself contains the following subdirectories:

**frontends**

This is where interfaces to codes are created. Within each subdirectory of yt/frontends/ there must exist the following files, even if empty:

- data_structures.py, where subclasses of AMRGridPatch, Dataset and AMRHierarchy are defined.
- io.py, where a subclass of IOHandler is defined.
- fields.py, where fields we expect to find in datasets are defined
- misc.py, where any miscellaneous functions or classes are defined.
- definitions.py, where any definitions specific to the frontend are defined. (i.e., header formats, etc.)

**fields**

This is where all of the derived fields that ship with yt are defined.

**geometry**

This is where geometric helper routines are defined. Handlers for grid and oct data, as well as
\[ \ddot{u} = c^2 \nabla^2 u \]
Alternative Tools

- Visit and Paraview
- See Wolfgang Kastaun’s talk
- See Roland Haas’s talk:
  - Simulation Tools (Ian Hinder): http://simulationtools.org
  - scivis/scidata (David Radice): https://bitbucket.org/dradice/
  - Cactus reader thorns
  - Replay Thorn (Roland Haas)
Advantages
- Python module:
  - Scripting interface
  - Easily extendable
  - Full power of Python available
  - Trivial installation
- Community developed
  - Inclusive and Accessible

Disadvantages
- No GUI to speak of
- Infrastructure
  - Inflexible
  - (Currently) not ready for numerical relativity
- Reading Einstein Toolkit data in its preliminary stages