Visualizing Einstein Toolkit Data with yt

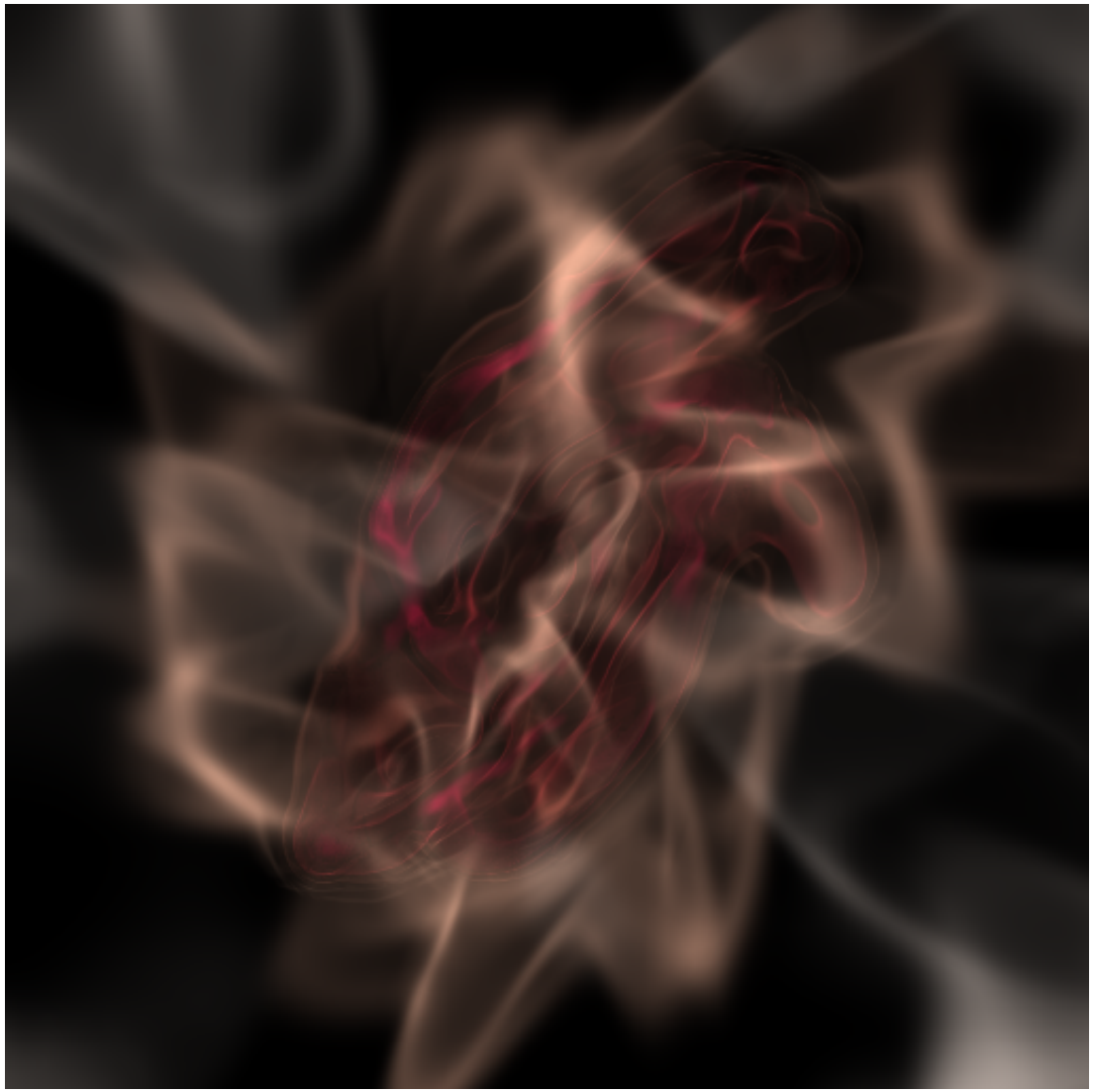
Jonah M. Miller

with Erik Schnetter, Matthew Turk, and Kacper Kowalik

Perimeter Institute for Theoretical Physics
and the University of Guelph

Einstein Toolkit Workshop, 2015

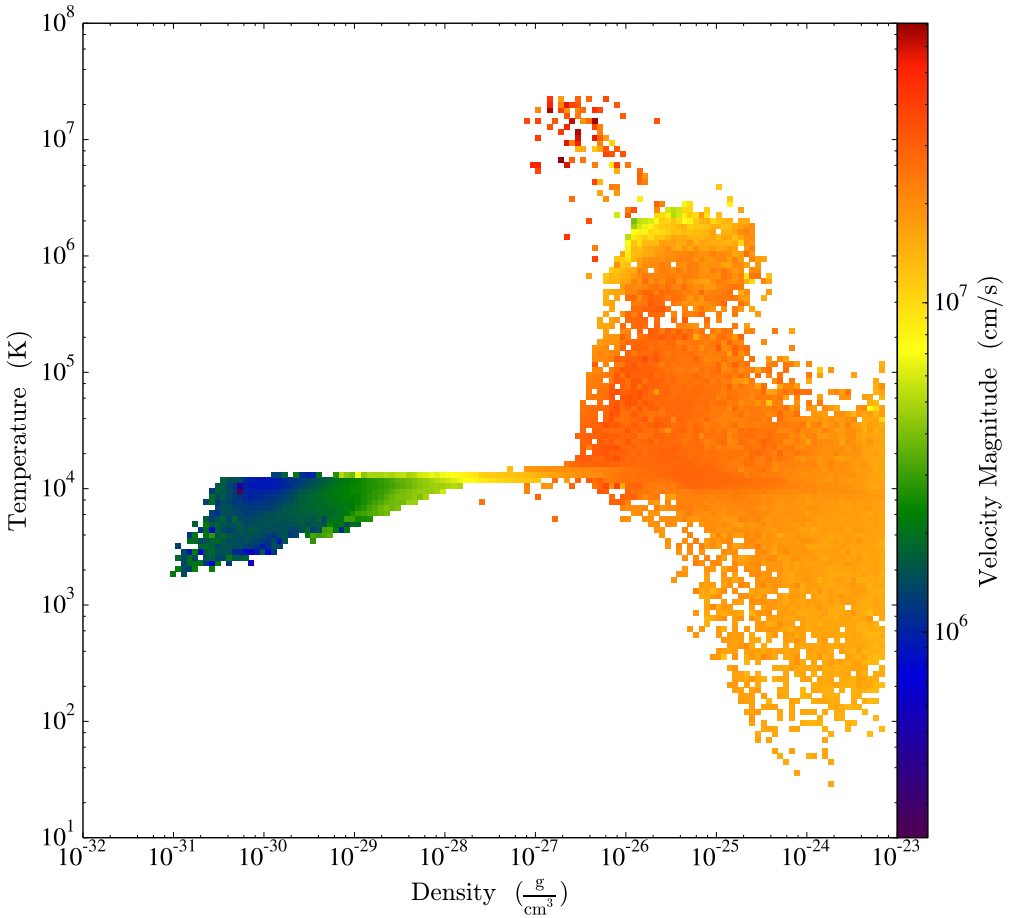
Volume Rendering



http://yt-project.org/data/

Data Analysis

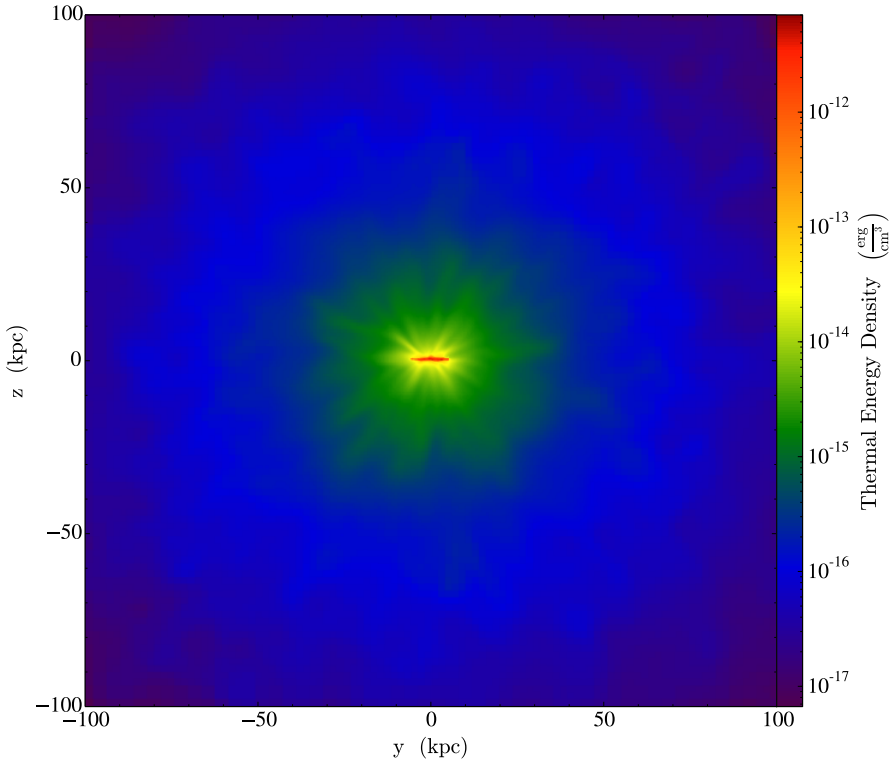
```
import yt
source = "./galaxy0030"
d = "density"
t = "temperature"
vm = "velocty_magnitude"
ds = yt.load(source)
ad = ds.all_data()
yt.PhasePlot(ad,d,t,vm)
```



http://yt-project.org/data/

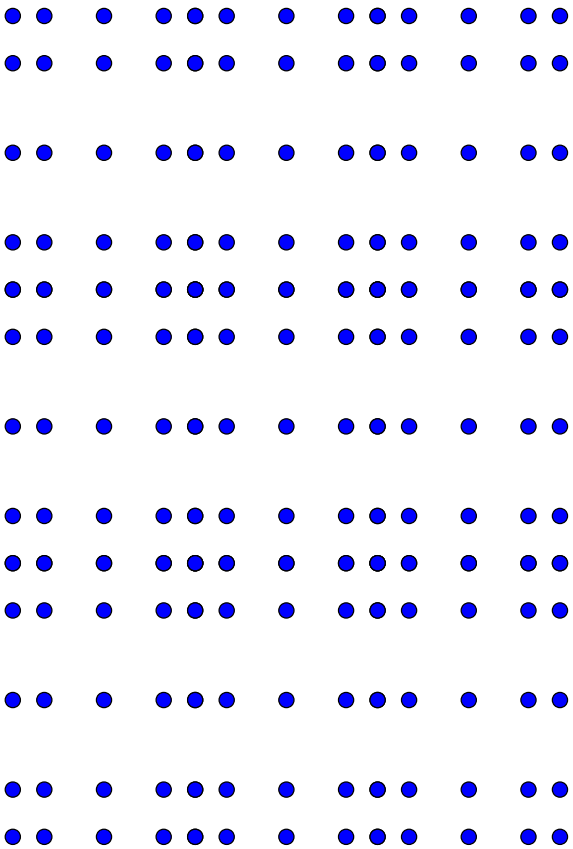
With Python, yt is Easily Extendable

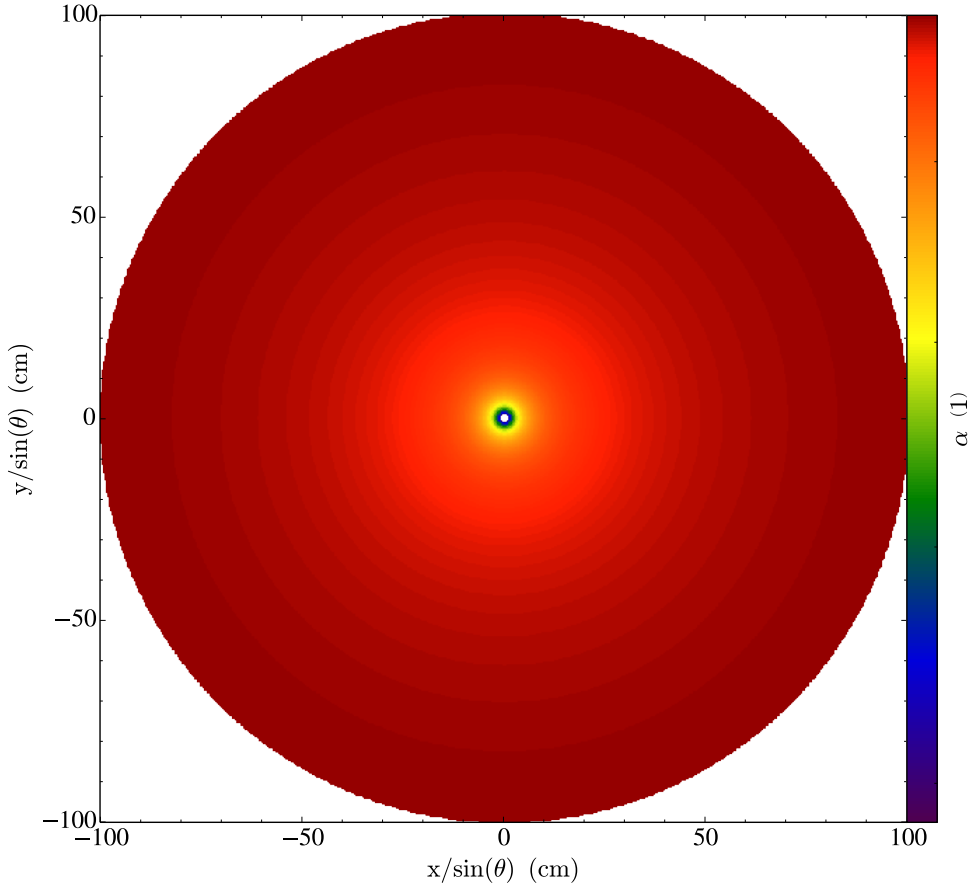
```
# function defining
# new quantity
fname = "thermal_energy_density"
def therm_en_dens(field,data):
    n = data['gas',
             'number_density']
    kT = data['gas', 'kT']
    return (3/2)*n*kT

# 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

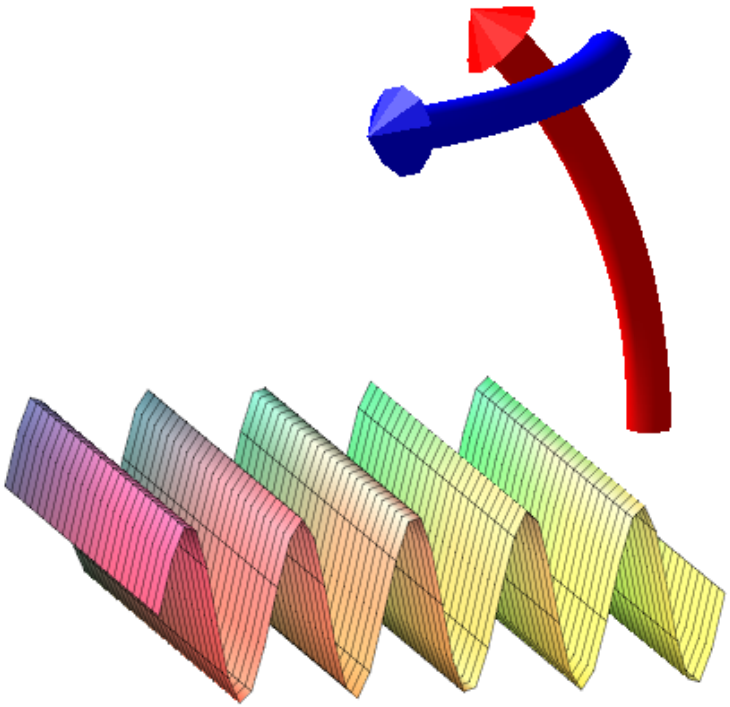


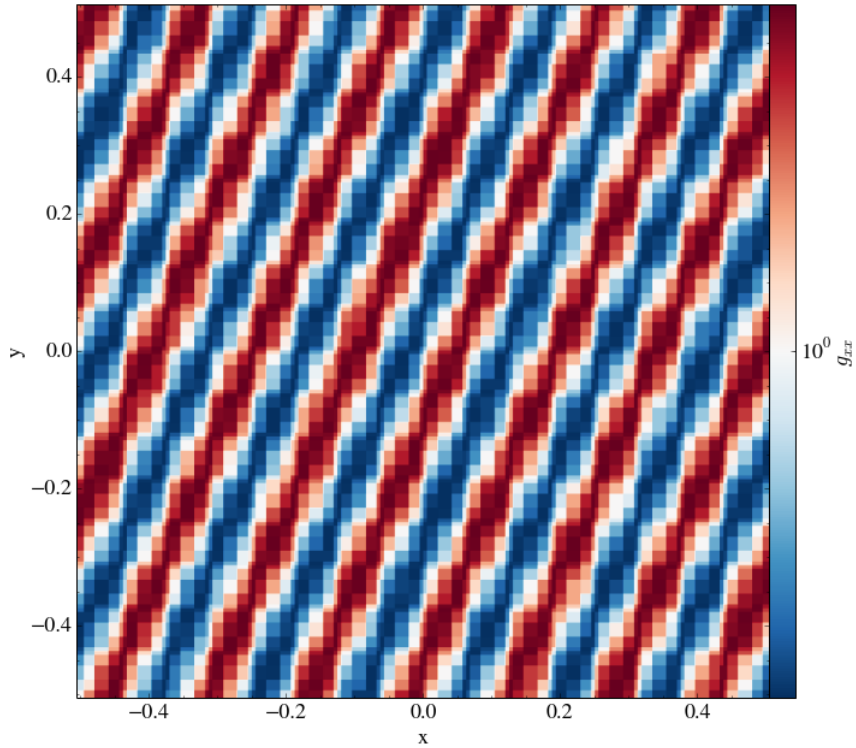


Miller and Schnetter. In Prep.

yt Extended for Discontinuous Galerkin Methods

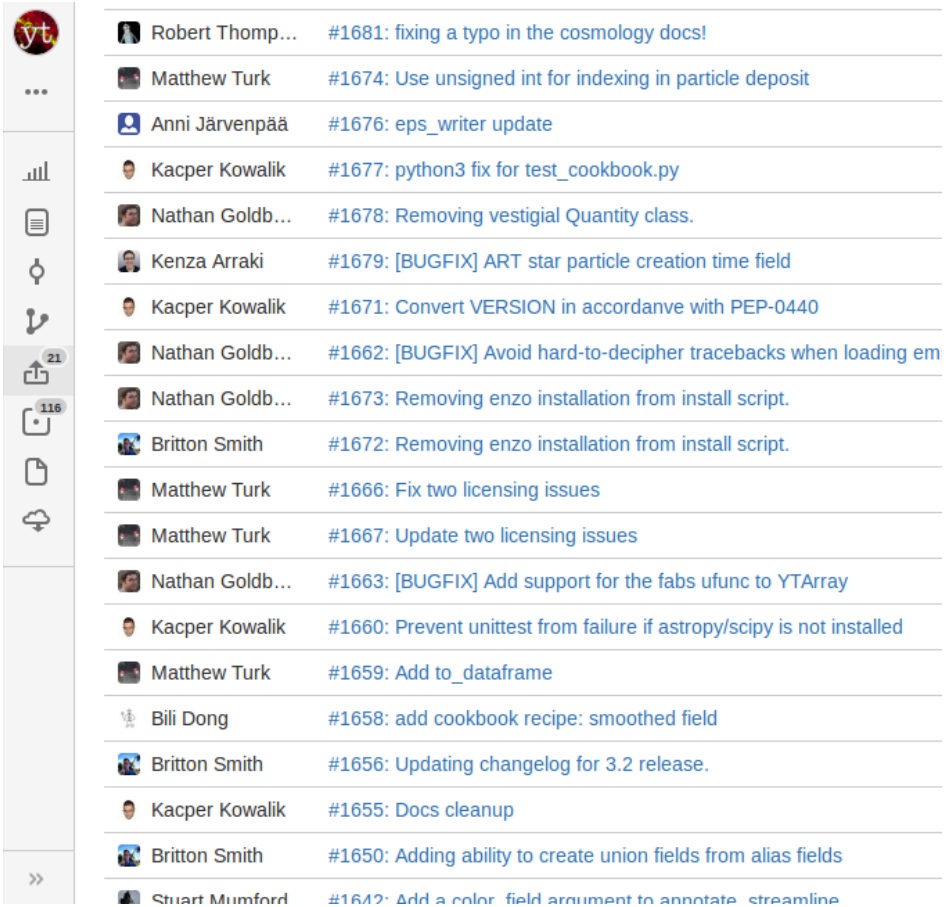
$$ds^2 = -H dt^2 + H dx^2 + dy^2 dz^2, \; H = 1 - A \sin\left(\frac{2\pi(x-t)}{d}\right)$$





Miller and Schnetter. In Prep.

Extensions are Encouraged!



Robert Thomp…	#1681: fixing a typo in the cosmology docs!
Matthew Turk	#1674: Use unsigned int for indexing in particle deposit
Anni Järvenpää	#1676: eps_writer update
Kacper Kowalik	#1677: python3 fix for test_cookbook.py
Nathan Goldb…	#1678: Removing vestigial Quantity class.
Kenza Arraki	#1679: [BUGFIX] ART star particle creation time field
Kacper Kowalik	#1671: Convert VERSION in accordanve with PEP-0440
Nathan Goldb…	#1662: [BUGFIX] Avoid hard-to-decipher tracebacks when loading em
Nathan Goldb…	#1673: Removing enzo installation from install script.
Britton Smith	#1672: Removing enzo installation from install script.
Matthew Turk	#1666: Fix two licensing issues
Matthew Turk	#1667: Update two licensing issues
Nathan Goldb…	#1663: [BUGFIX] Add support for the fabs ufunc to YTArray
Kacper Kowalik	#1660: Prevent unittest from failure if astropy/scipy is not installed
Matthew Turk	#1659: Add to_dataframe
Bili Dong	#1658: add cookbook recipe: smoothed field
Britton Smith	#1656: Updating changelog for 3.2 release.
Kacper Kowalik	#1655: Docs cleanup
Britton Smith	#1650: Adding ability to create union fields from alias fields
Stuart Mumford	#1642: Add a color_field argument to annotate_streamline

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-hg`. 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 its self 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`

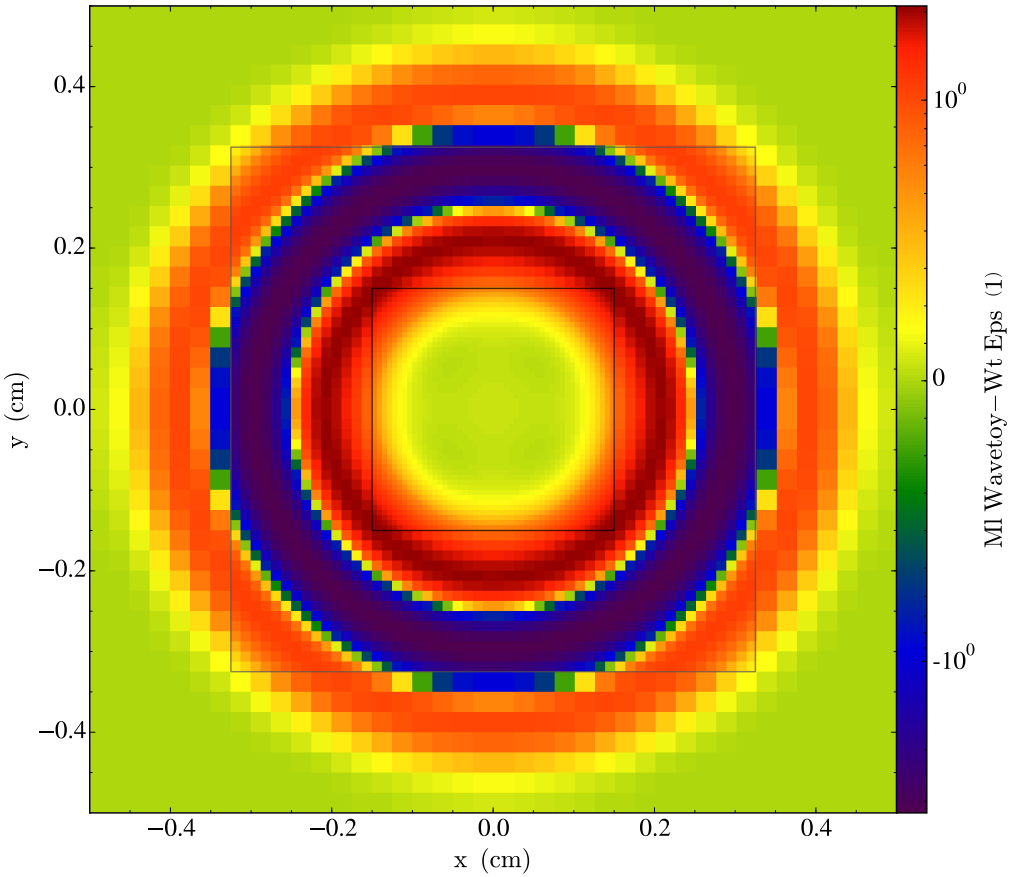
This is where all of the derived fields that ship with yt are defined.

`geometry`

This is where geometric helpler routines are defined. Handlers for grid and oct data, as well as

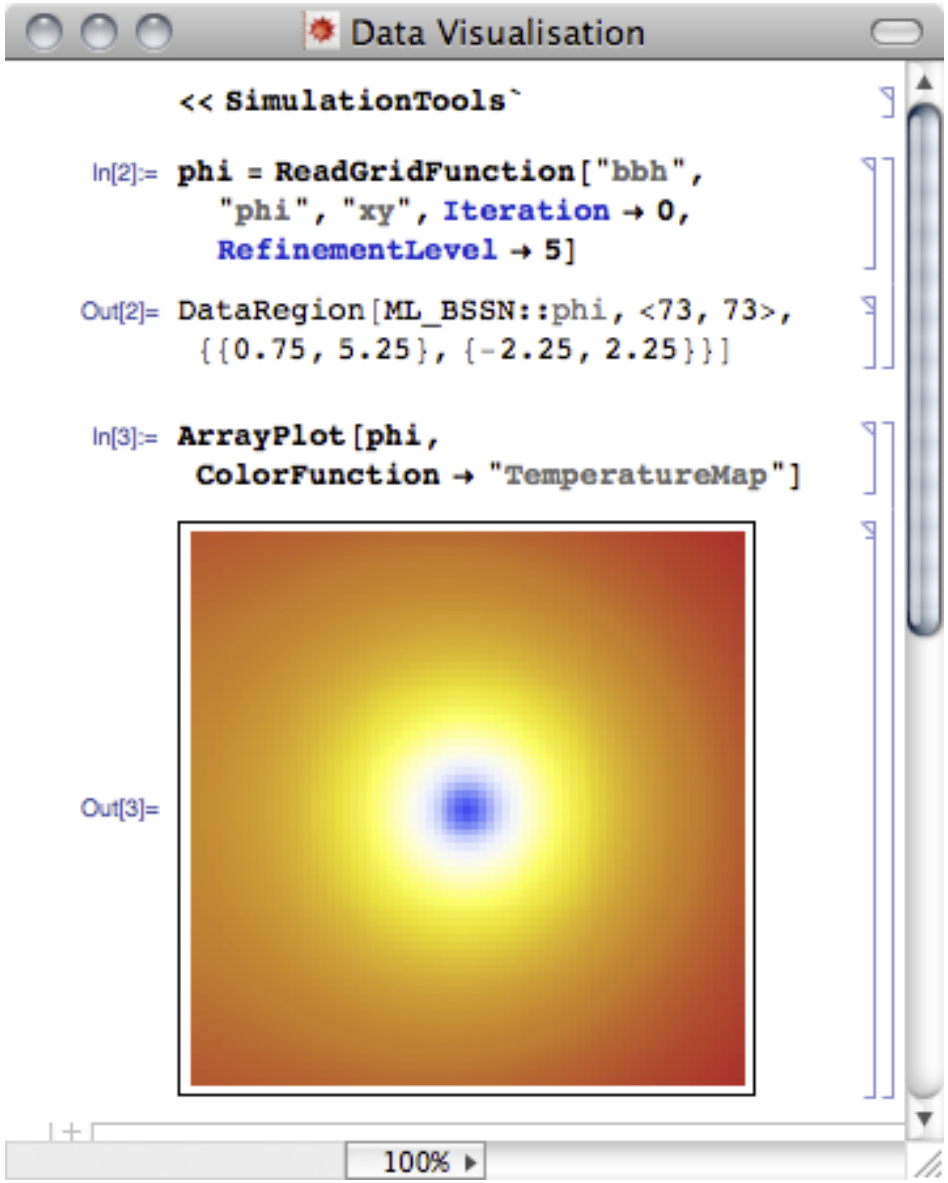
Preliminary Einstein Toolkit Visualizations

$$\ddot{u} = c^2 \nabla^2 u$$



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