The yt Project: Visualizing and Analyzing Volumetric Data Across Domains

Nathan Goldbaum,
Matthew Turk, Sam Walkow
NCSA, University of Illinois
ngoldbau@illinois.edu
yt-project.org
Data-exp-lab.github.io

Expanding the yt Community

The yt Project aims to produce an integrated science environment based on the Python programming language for collaboratively asking and answering questions about volumetric data. Currently yt is mostly used by astrophysicists who run and analyze simulations of astrophysical phenomena but we are in the process of adding support for data from sources across the physical sciences.

yt is designed to guide scientific inquiry (analysis, visualization, simulation) through physically-motivated understanding. It is released under the BSD 3-clause license, developed completely in the open, and is designed to present a library of loosely-coupled components that can be easily integrated with other Python tools.

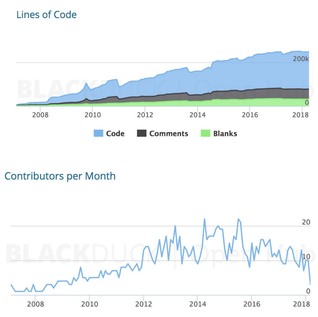
In a Nutshell, yt...

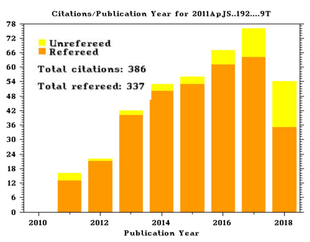
... has had 24,056 commits made by 162 contributors representing 170,615 lines of code

... is mostly written in Python with an average number of source code comments

... has a well established, mature codebase maintained by a very large development team with decreasing Y-O-Y commits

... took an estimated 45 years of effort (COCOMO model) starting with its first commit in February, 2007 ending with its most recent commit about 1 month ago





Citation Statistics from NASA ADS

JOSS The Journal of Open Source Software

unyt: Handle, manipulate, and convert data with units in Python

Nathan J. Goldbaum[1], John A. ZuHone[2], Matthew J. Turk[1], Kacper Kowalik[1], and Anna L. Rosen[2]

1 National Center for Supercomputing Applications, University of Illinois at Urbana-Champaign. 1205 W Clark St, Urbana, IL USA 61801 2 Harvard-Smithsonian Center for Astrophysics. 60 Garden St, Cambridge, MA USA 02138

yt.units is Now unyt

Since the release of yt 3.0, the yt.units module has provided a system for tracking unit consistency and converting data between different units and unit systems. This substantially improved the yt user experience by making clear the units of data returned by yt and has eliminated an entire class of bugs in user scripts and in yt's internals due to mismatched units and incorrect unit conversions.

The unit system is very popular but using it requires installing and importing the full yt package, which can be heavyweight for applications that do not need yt's other capabilities. To ease the use of the yt unit system in other applications, we have begun work on extracting yt.units from the yt package and created a new package named unyt for the code to live in. We are planning on making yt itself depend on unyt for the yt 4.0 release. We have separately released unyt as a standalone package, along with an article in the Journal of Open Source Software providing a description of the package and a set of performance benchmarks.

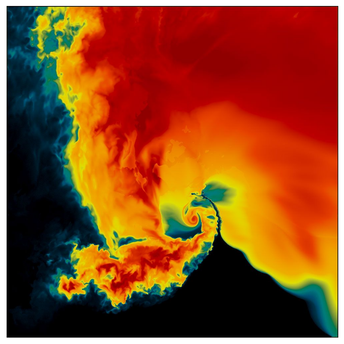
Domain Context System

Learning a new software package requires significant cognitive overhead. We do not want to add to that cognitive overhead by using unfamiliar jargon or terminology in the yt user interface or in the way yt presents data to the user. Instead, we have an opportunity to make learning the software easier by making yt adapt the terminology, data presentation, and visualization style to the user's expectations in their field of study.

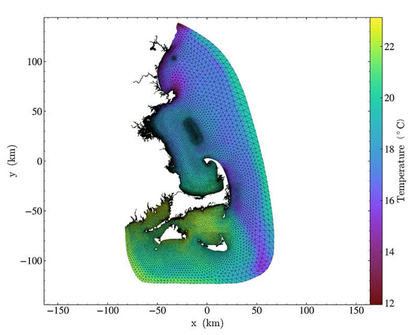
We are planning to implement domain contexts for the following research areas:

- Whole Earth Seismology and Tomography (Ben Holtzmann, Columbia University, funded)
- Weather simulations (Leigh Orf, University of Wisconsin, funded)
- Nuclear Engineering (Katy Huff, University of Illinois and Alex Lindsay, Idaho National lab)
- PIC Plasma Simulations (Michael Bussman, Helmholtz Zentrum Dresden-Rossendorf)
- Observational Astronomy (Adam Ginsburg, NRAO and Sarah Tuttle, University of Washington)
- Oceanography and Hydrology (Chris Barker, NOAA and Richard Signell, USGS)

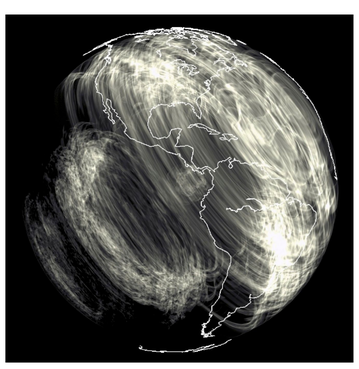
Our goal is to make the domain context system pluggable and easily extensible without requiring knowledge of yt internals. We expect to integrate community-developed domain contexts for other fields in parallel.



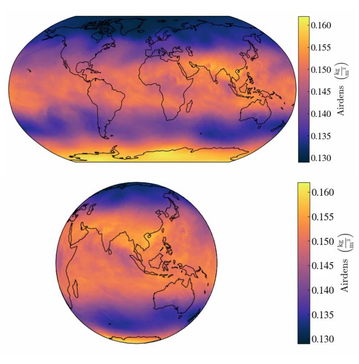
Predicted Weather Radar from Tornadogenesis Simulation
Leigh Orf (University of Wisconsin),
Image Credit: NCSA AVL



FVCOM Ocean Forecast Model of the North Atlantic Coast
(NOAA, UMass Dartmouth)



Volume Rendering of Seismic Waves Produced by the Tohoku Earthquake
Ben Holtzman (Columbia University)



GEOS-5 Global Weather Model visualized with yt using map projections from Cartopy
Madicken Munk, (UIUC)

Current status of the yt Project

yt is designed to live within the broader scientific python ecosystem and enable researchers to utilize technologies and tools in a platform-neutral way to advanced scientific inquiry. yt is focused on developing physically-motivated inquiry and understanding, rather than computationally-focused inquiry. This will enable direct technology transfer between individuals and research groups, as well as facilitate deeper study of simulation data, as well as 3D remote sensing data.

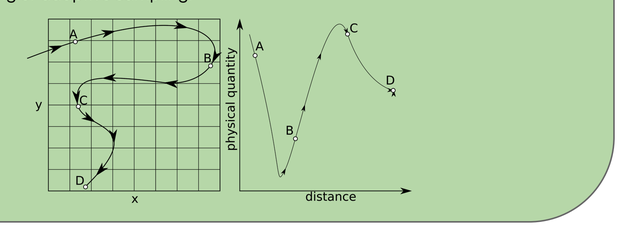
This project was awarded an NSF SI2-SSI grant which began in earnest in late 2017. The full proposal can be found at `goo.gl/6w25zy` and the development of all functionality occurs in public view. This grant enables development of new functionality described below and support for data from across the physical sciences.

We are currently working on modularizing the yt package, and have released `yt_astro_analysis` containing astrophysics-specific functionality as its own package. We expect to release yt 4.0 with a domain context system implemented, including initial support for several domains, with additional domains being added in subsequent releases.

Path Traversal and Non-local Analysis

Currently the yt field system requires all input fields to be defined at the location of a data element or its neighbors. We will add an API and associated infrastructure for traversing a path through a volume while simultaneously accumulating data. This will generalize our current support for volume rendering via ray tracing.

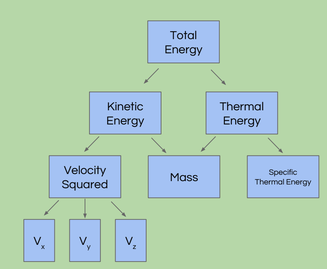
- A path traversal can accumulate, deposit, or sample data along the path
- The path can be algorithmically updated or pre-set
- Path traversal operations will be controlled at a high level via a Python API
- Ultimately this will enable domain convolutions via large-scale Monte Carlo sampling or adaptive sampling.



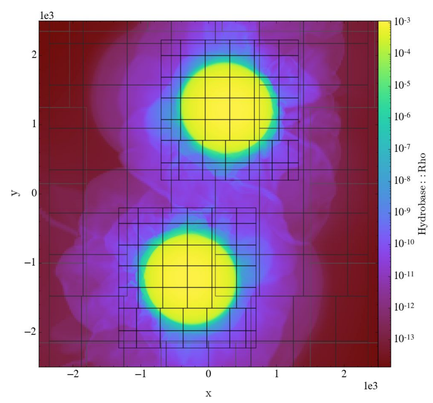
Symbolic Fields

We are rewriting the yt field system to calculate the definition of a field in terms of a field dependency graph rather than use chained python functions as in the current implementation.

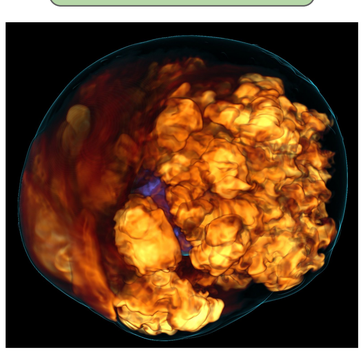
This will enable algebraic optimizations, such as common subexpression elimination, as well as performance optimization via code generation or offloading computation to JIT compiler like numba or distributed computation engines like Dask.



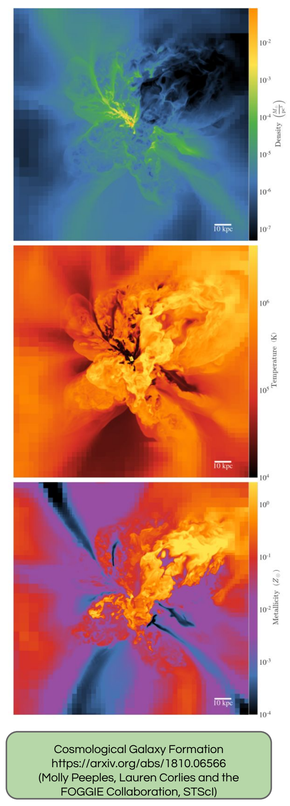
Community Developments



Einstein Toolkit Relatavistic Hydrodynamics
Colliding Neutron Star Simulation
(Jonah Miller, LANL)



Supernova Ignition Simulation
https://arxiv.org/abs/1807.07579
(Evan O'Connor and Sean Couch, MSU)



Cosmological Galaxy Formation
https://arxiv.org/abs/1810.06566
(Molly Peeples, Lauren Corlies and the FOGGIE Collaboration, STScI)

yt Extension Ecosystem

As part of our effort to make yt more modular we have created a system for community-maintained extensions to yt. The goal is that these extensions will enable domain-specific analysis that requires extensive domain knowledge to maintain. This allows the yt team to concentrate on maintaining core algorithms and functionality while domain experts can separately release and maintain functionality the need for their science. For more details about the yt extension system, see YTEP-0029.

Trident (trident-project.org)
Maintainer: Cameron Hummels

Generate synthetic absorption spectra from simulation data loadable by yt.

ytini (www.ytini.com)
Maintainer: Jill Naiman

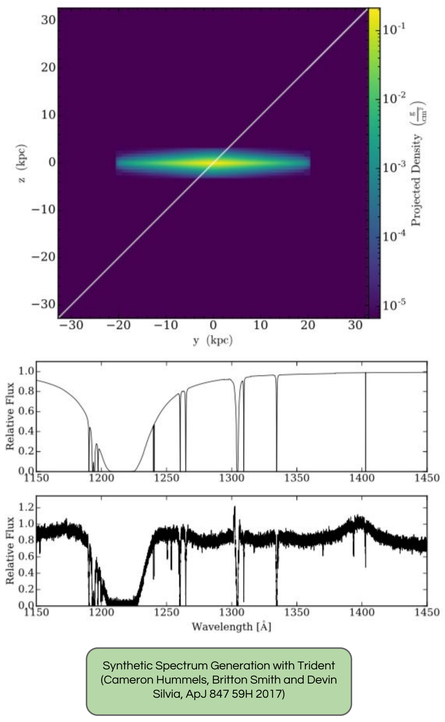
Interface between yt and the Houdini commercial volume rendering and special effects suite..

- ytree (ytree.readthedocs.io)
  Maintainer: Britton Smith

  Load and process halo merger trees in yt

PyXSIM (hea-www.cfa.harvar.edu/~jzuhone/pyxsim)
Maintainer: John ZuHone

Generate synthetic x-ray observations for data loadable by yt.



Synthetic Spectrum Generation with Trident
(Cameron Hummels, Britton Smith and Devin Silvia, ApJ 847 59H 2017)

This work is supported by NSF SI2-SSI OAC-1663914 (goo.gl/6w25zy)