yt: An Integrated Science Environment for Astrophysical Simulations

Matthew Turk
There is only one sky.

(but there are many simulation codes)
Different methods, data structures, assumptions, IO methods, units, variable names, ...

PHYSICALLY-MOTIVATED VISUALIZATION

Simulation of dwarf galaxies by Wise & Cen

THE YT PROJECT

ASTROPHYSICAL SIMULATION ANALYSIS AND VISUALIZATION

DETAILED DATA ANALYSIS AND VISUALIZATIONS, WRITTEN BY WORKING ASTROPHYSICISTS AND DESIGNED FOR PRAGMATIC ANALYSIS NEEDS.

DATA-DRIVEN
Inspect your data

yt is designed to provide a consistent, cross-code interface to

COMMUNITY
Participants welcome!

yt is composed of a friendly community of users and

FREE SOFTWARE
Open Source, Open Science

yt is developed completely in the open, released under the GPL

HOW DO I CITE YT?

WHICH CODES ARE SUPPORTED?
yt Overview

yt is a community-developed analysis and visualization toolkit for astrophysical simulation data. yt provides full support for the Enzo, Orion, Nyx, and FLASH codes, with preliminary support for RAMSES, ART, and Maestro. It runs both interactively and non-interactively, and has been designed to support as many operations as possible in parallel. For more detailed information, see our ApJS paper.

To install yt, see yt Orientation Session or Installing yt. To see what's new since the last version, check out Changelog.

If you use yt in a paper, you are highly encouraged to submit the repository containing the scripts you used to analyze and visualize your data to the yt Hub, and we ask that you consider citing our method paper, as well. If you are looking to use yt, then check out the yt Hub for ideas of how other people used yt to generate worthwhile analysis. We encourage you to explore the source code and even consider contributing your enhancements and scripts.

For more information, please visit our homepage and for help, please see Asking for Help.

Analysis and Visualization with yt

Welcome to yt!
What's yt all about?

Interacting with yt
Different ways -- scripting, GUIs, prompts, explorers -- to explore your data.

Visualizing Data
An overview of different ways to visualize data: projections, slices, phase plots, and volume rendering.

yt Orientation
Quickly get up and running with yt: zero to sixty.

Analyzing Data
An overview of different ways to handle and process data

Analysis Modules
Discussions of some provided procedures for astrophysical analysis like halo finding and synthetic imaging.
yt is a python package for analyzing and visualizing astrophysical simulation output from a couple simulation platforms. Check out the homepage (there's a link just above!) where we have documentation, a cookbook, and some user community information.

Clone this repository (size: 40.4 MB): HTTPS / SSH
$ hg clone ssh://hg@bitbucket.org/yt_analysis/yt

<table>
<thead>
<tr>
<th>Filename</th>
<th>Size</th>
<th>Date modified</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>doc</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>scripts</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>tests</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>yt</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.hignore</td>
<td>239 B</td>
<td>2011-03-28</td>
<td>Updated .hignore.</td>
</tr>
<tr>
<td>.hgtags</td>
<td>248.5 KB</td>
<td>2010-08-30</td>
<td>Removing SVN tags</td>
</tr>
<tr>
<td>CREDITS</td>
<td>2.1 KB</td>
<td>2011-09-12</td>
<td>Adding Elizabeth to CREDITS -- thanks to Elizabeth for her work</td>
</tr>
<tr>
<td>FUNDING</td>
<td>1.2 KB</td>
<td>2011-01-17</td>
<td>Adding a FUNDING document. Please feel free to add to this.</td>
</tr>
<tr>
<td>LICENSE.txt</td>
<td>34.3 KB</td>
<td>2007-08-25</td>
<td>[svn r237] Added GPL-3 to repo, and included appropriate</td>
</tr>
<tr>
<td>MANIFEST.in</td>
<td>142 B</td>
<td>2011-08-26</td>
<td>Adding distribute_setup.py explicitly.</td>
</tr>
<tr>
<td>README</td>
<td>1.0 KB</td>
<td>2011-08-21</td>
<td>enzotools -&gt; yt-project.</td>
</tr>
</tbody>
</table>
install script:

- Full dependency stack
- Source code
- Development environment
- GUI
- Sample data
yt has been designed to address physical, not computational, entities.
The Universe is full of gas, dark matter and stars. yt makes it easy to access that material.
Transparent IO, masking of overlapping data, load-on-demand, geometric and non-geometric selection, field generation, and common interfaces to different datatypes.
Enzo, Orion, CASTRO, FLASH

Chombo, Tiger, Athena, ART, RAMSES

yt is designed to be the lingua franca of astrophysical codes.
## Objects

<table>
<thead>
<tr>
<th>Category</th>
<th>Dimension</th>
</tr>
</thead>
<tbody>
<tr>
<td>Orthogonal Rays</td>
<td>1D</td>
</tr>
<tr>
<td>Non-orthogonal Rays</td>
<td>1D</td>
</tr>
<tr>
<td>Slices</td>
<td>2D</td>
</tr>
<tr>
<td>Oblique Slices</td>
<td>2D</td>
</tr>
<tr>
<td>Projections</td>
<td>2D</td>
</tr>
<tr>
<td>Spheres</td>
<td>3D</td>
</tr>
<tr>
<td>Rectangular Prisms</td>
<td>3D</td>
</tr>
<tr>
<td>Disks/Cylinders</td>
<td>3D</td>
</tr>
<tr>
<td>Inclined Boxes</td>
<td>3D</td>
</tr>
<tr>
<td>Clumps</td>
<td>3D</td>
</tr>
<tr>
<td>Extracted Regions</td>
<td>3D</td>
</tr>
<tr>
<td>Boolean combinations</td>
<td>3D</td>
</tr>
</tbody>
</table>
from yt.mods import *

pf = load('galaxy0030/galaxy0030')
p = SlicePlot(pf, 2, 'Density', 'c', (200, 'kpc'))
p.save('Galaxy')
from yt.mods import *
pf = load('galaxy0030/galaxy0030')
p = SlicePlot(pf, 2, 'Density', 'c', (200,'kpc'))
p.set_width(20, 'kpc')
p.save("GalaxyZoom")
from yt.mods import *

pf = load('galaxy0030/galaxy0030')

p = ProjectionPlot(pf, 2, 'Density', 'c', (20,'kpc'))

p.save('Galaxy')
\[
\frac{dI_\nu}{ds} = j_\nu - \alpha_\nu I_\nu
\]

Designed around integrating through a volume: visualization is a side effect.
$10^{-14} \text{ g/cc}$
Absorption Spectrum
Coordinate Transformations
Halo Finding
Mass Functions
Merger Trees
Halo Profiling
Level Sets
Light Cones
Light Rays
Time Series
Star Analysis
Two-Point Analysis
Level Sets
Synthetic Spectra
Two-Point Functions

Temperature Difference Probability Distribution Function for some length $L$
Analysis Modules

Extensions take yt fundamentals and run with them. For certain analysis needs, these tools make life a lot easier.

- Halo Finding
  - HOP
  - Friends-of-Friends
  - Running HaloFinder
  - Halo Data Access
  - Halo List Data Access
  - Loading Haloes Off Disk
  - General Parallel Halo Analysis
  - Parallel HOP
  - Halo Finding In A Subvolume

- Analyzing an Entire Simulation
  - EnzoSimulation Options
  - The Dataset List
  - Cosmology Splices
  - Planning a Cosmological Simulation

- Halo Mass Function: Start to Finish
  - Halo Finding
  - Halo Profiling
  - Halo Mass Function
  - Putting it All Together
Parallelism
Multi-level parallelism: dynamic workgroups, communicators, subgroups and task queues
Parallelism

<table>
<thead>
<tr>
<th>Embarassingly Parallel</th>
<th>Spatial Decomposition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decomposed by load or IO characteristics</td>
<td>Helper functions to decompose the domain</td>
</tr>
</tbody>
</table>
Developing as a Team
Forky development:

very low barrier to entry;
everything comes in the box.
Testing:

answer as well as integration tests get run every 30 minutes.
Community

- Mailing lists: community, development
- IRC (#yt on irc.freenode.net)
- Cookbook, docs
- Curriculum from workshop
In a Nutshell, yt...

...has had 6,037 commits made by 34 contributors representing 88,943 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 large development team with stable year-over-year commits

...took an estimated 23 years of effort (COCOMO model) starting with its first commit in February, 2007 ending with its most recent commit 5 days ago

Languages

- Python 72%
- C 16%
- C++ 5%
- Other 7%

Lines of Code

- Code
- Comments
- Blanks
80,000 lines of code
Python, Cython, C
20 contributors (60+ users)
Contributors from 10+ institutions
Contributors

Tom Abel
David Collins
Andrew Cunningham
Nathan Goldbaum
Cameron Hummels
Ji-hoon Kim
Steffen Klemer
Kacper Kowalik
Michael Kuhlen
Eve Lee
Chris Malone
Chris Moody
Andrew Myers
Jeff Oishi

Jean-Claude Passy
Thomass Robitaille
Anna Rosen
Anthony Scopatz
Devin Silvia
Sam Skillman
Stephen Skory
Britton Smith
Geoffrey So
Casey Stark
Elizabeth Tasker
Rick Wagner
John Wise
John ZuHone
No more grids, only chunks
Grids  Chunks
Particle segregation, multiple fluids
Address technical debt
Strategies

- Patch: chunks are collections of grids
- Octree: direct octree parsing
- SPH/N-body: octree indexing
3.0 Milestones

1. Patch-based analysis and 2D viz
2. Patch-based volume rendering
3. Halo Finding
4. Octree analysis and 2D viz
5. Particle analysis and 2D viz
Caveats!
SPH Support

- Particle support only at first
- No first light (ETA: 3-6 weeks)
- Many free parameters, many output formats
Octree Support

- Volume rendering not ready yet
- Subtleties may lurk
- ART half-finished
I volunteer!
(demo)
Thank you.

yt-project.org