Adaptive Mesh Refinement
Data Analysis in Python

Matthew Turk (KIPAC / SLAC / Stanford)
with Jeff Oishi (UC Berkeley)
SciPy 2008
“Nothing”
“Nothing”
“Nothing”  Stuff.
Size
Visible Universe
Visible Universe

$10^{23}$ km

Springel et al. 2005
Galaxy Cluster

$10^{19}$ km
10^{17} \text{ km}
Star Cluster
Star Cluster

10^{15} km
Star
Star

$10^6$ km
Earth
Earth

6000 km
835 km
You & Me
Universe
<table>
<thead>
<tr>
<th>BORING</th>
<th>BORING</th>
</tr>
</thead>
<tbody>
<tr>
<td>INTERESTING</td>
<td>BORING</td>
</tr>
</tbody>
</table>
Adaptive Mesh Refinement (patch-based)
N-body gravity, DM and Stars
Magnetohydrodynamics
Radiative Cooling
Radiative Transfer
12-species chemistry model

Black holes through galaxy clusters!

Developed and used at UCSD, Stanford, Columbia, CITA, Cambridge, ...
Adaptive Mesh Refinement (patch-based)
N-body gravity, DM and Stars
Magnetohydrodynamics
Radiative Cooling
Radiative Transfer
12-species chemistry model

Black holes through galaxy clusters!

Developed and used at UCSD, Stanford, Columbia, CITA, Cambridge, ...
Requirements
Processing
Speed
Plotting
Stability
Stable-ility
Free.
Free.

(completely.)
projection, please.
slice, please.
sphere, please.
yt

box, please.
Output

Hierarchy

(grids)
What is distinguishing?
Data Object Protocol

>>> grid[“Density”]
>>> sphere[“RadialVelocity”]
>>> disk[“Temperature”]
>>> sphere.quantities[“CenterOfMass”]()
def HydrogenNuclei(field, data):
    return (data["HI_Density"] +
            data["HII_Density"])) / m_h

add_field("HydrogenNuclei",
          function=HydrogenNuclei,
          units = r"\text{cm}^{-3}"")
Derived Fields

Ghost Zones

Real Zones
Derived Fields

Ghost Zones

Real Zones

Finite difference stencils!
Units
Analysis

Data Selection
Profiling
Halo Finding
Contour Finding
Derived Fields
Topologically Connected Sets
Plotting

Pixelization

(traited) VTK
Particles
HDF5 Wrappers

Discovery
Slicing
Reading
Parallel Analysis
Parallel Analysis Interface

partition_hierarch_y_2d()
Parallel Analysis Interface

get_grids()

GridIterator

ParallelGridIterator
>>> pf = EnzoStaticOutput("my_data")
>>> pc = PlotCollection(pf)

>>> pc.add_projection("Density", 0)
>>> pc.save("my_data")
yesterday morning:

```python
>>> pf = EnzoStaticOutput("my_data")
>>> pc = PlotCollection(pf)

>>> pc.add_projection("Density", 0)
>>> pc.save("my_data")
```
yesterday morning:

```python
>>> pf = EnzoStaticOutput("my_data")
>>> pc = PlotCollection(pf)

>>> pc.add_projection("Density", 0)
>>> pc.save("my_data")
```

this morning:

```python
>>> "my_data" | projector_x
>>> "my_data" | slicer(field="Temperature")
```
Preparing for Release
yt Overview

yt is a toolkit designed to analyze, manage and plot adaptive mesh refinement data from the Enzo code.

Contents:

- Introduction
  - History
  - What yt is and is not
  - What functionality does yt offer?
- Getting Started
  - Maintained Installations
  - Binary Packages
  - Installing From Source
- The Quick Guide to yt
  - Starting Python
  - Opening Your Data File
  - Making Plots
  - Saving Plots
  - A Few More Plots
- The Tutorial
  - Getting Started
Making Timeseries Plots

**Note:** Unlike the other scripts in this cookbook, these do not serve as standalone executables. You must edit them with your data information.

I don’t think anyone will argue that timeseries data is important. It is possible to create it with yt; however, a few things should be noted.

Python is notorious for leaking memory via circular references – if you have a variable `var1` that references variable `var2`, when one is deleted, is the other? Efforts have been made to identify and remove all leaking references, but if references are created in your code that generates the timeseries data, you may find that more memory is used than is desired.

The process of generating time series data is fairly simple: you iterate over the datasets, and generate a value for each. (`cookbook_timeseries_max_dens.py`)

```python
from yt.mods import *
max_rho = []
max_pos = []
times = []
for i in range(30):
    pf = lagos.EnzoStaticOutput("my_output@04i" % (i))
    v, c = pf.h.find_max("Density")
    max_rho.append(v)
    max_pos.append(c)
```
Interfaces
IPython
wxPython
In[1]: import sys
sys.path.insert(0, "/home/mturk/Development/yt/trunk/")
from yt.mods import *
pf = EnzoStaticOutput("/home/mturk/Research/data/RD0014/RedshiftOutput0014")
pc = PlotCollectionInteractive(pf)
pc.pylab.show = show
pc.add_slice("Density", 0)

Out[1]: Kernel Error: The interpreter might still be starting up.

In[2]: pc.pylab.show()
## Select your criteria

<table>
<thead>
<tr>
<th>Parameterfile</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>DataDump0039</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DataDump0040</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DataDump0041</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DataDump0042</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DataDump0043</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Type</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Projection</td>
<td>All</td>
<td>None</td>
</tr>
<tr>
<td>Slice</td>
<td>All</td>
<td>None</td>
</tr>
<tr>
<td>ThreePhase</td>
<td>All</td>
<td>None</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Width</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>All</td>
<td>None</td>
</tr>
<tr>
<td>10.0</td>
<td>All</td>
<td>None</td>
</tr>
<tr>
<td>100.0</td>
<td>All</td>
<td>None</td>
</tr>
<tr>
<td>1000.0</td>
<td>All</td>
<td>None</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Unit</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>au</td>
<td>All</td>
<td>None</td>
</tr>
<tr>
<td>kpc</td>
<td>All</td>
<td>None</td>
</tr>
<tr>
<td>pc</td>
<td>All</td>
<td>None</td>
</tr>
<tr>
<td>rsun</td>
<td>All</td>
<td>None</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Density</th>
<th></th>
</tr>
</thead>
</table>

192 images will be returned

[List Parameter Files]
Generalization
Things `yt` Can Do

- Slices
- Projections
- 1-, 2- and 3-D Profiles
- Halo finding
- Fixed Resolution Extraction
- Generic 3D objects
- Extracted Sets
- Wrap Enzo Fortran modules
- Topologically Connected Sets
- Derived Quantities
- Particle support
- Movies
- Time series analysis
- Galaxy merger trees
- Simple data management
- Web gallery
- Parallel Analysis
- Primitive problem generation support
- X-Ray spectrum (from ‘Cloudy’)
What next?
Committed to Open Science
Committed to Open Science

No Private Branches
Committed to Open Science

- No Private Branches
- No Secret Features
Committed to Open Science

- No Private Branches
- No Secret Features
- Free Repository Space
Acknowledgments

Enzo: Greg Bryan, Mike Norman, Tom Abel, Brian O’Shea (and on and on)
YT: Jeff Oishi, Britton Smith, David Collins

Sponsored by: KIPAC, Stanford, and LANL
Thank you.