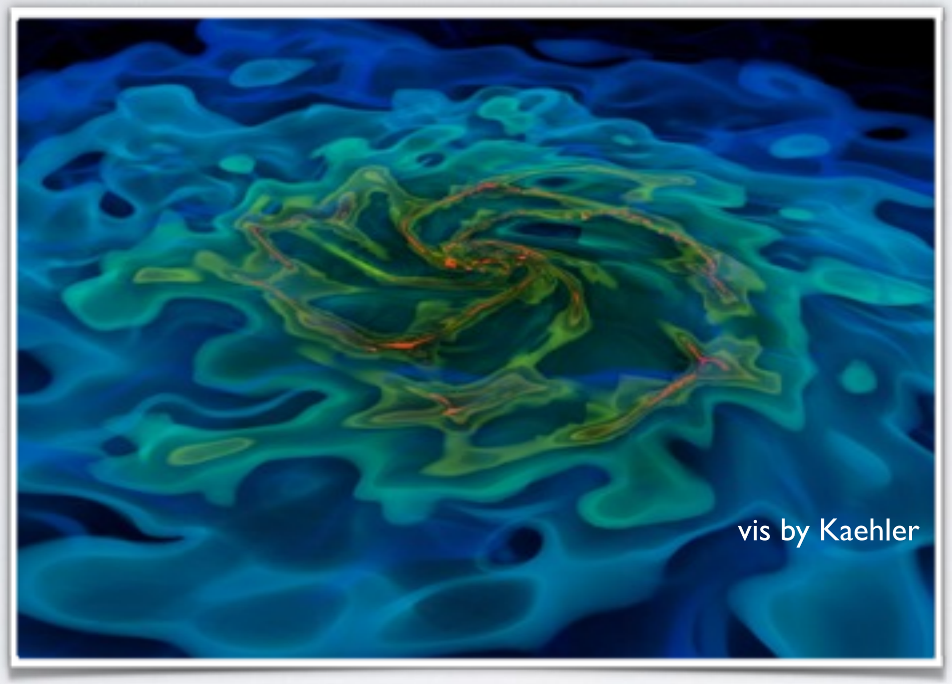
The Assembling Galaxies of Resolved Anatomy (AGORA) Project

Two Years After Initiation



Joel Primack (UCSC) and Ji-hoon Kim (Caltech)

on behalf of the AGORA Collaboration

Special thanks in the past year to: O. Agertz (MPIA), A. Bogert (UCSC), N. Gnedin (Fermilab), O. Hahn (ETH), C. Hummels (Arizona), P. Madau (UCSC), M. Rocha (UCSC) B. Smith (Edinburgh), R. Teyssier (Zurich), M. Turk (NCSA), J. Wadsley (McMaster), and many others

University of California
High-Performance
AstroComputing Center
(UC-HiPACC)
Joel Primack, Director

University of California
Santa Cruz
Next Telescope Science
Institute (NEXSI)
Piero Madau, Director

Assembling Galaxies of Resolved Anatomy
AGORA High-Resolution Galaxy Simulation Comparison Project Steering Committee

Piero Madau & Joel R. Primack, UCSC, Co-Chairs
Tom Abel, Stanford
Nick Gnedin, Chicago/Fermilab
Lucio Mayer, University of Zurich
Romain Teyssier, Saclay & Zurich
James Wadsley, McMaster

Ji-hoon Kim, UCSC (Coordinator)

108 astrophysicists have joined AGORA from 50 institutions in 8 countries using 11 simulation codes

www.AGORAsimulations.org

AGORA Galaxy Simulations Comparison

- 110 participants from 57 institutions worldwide as of today!

  - webpage: http://www.AGORAsimulations.org

- Common ICs for all codes (cosmological and isolated)

  - Expedited IC construction by MUulti-Scale Initial Conditions (MUSIC, Hahn & Abel 2011)
  - 4 halo masses ($\sim 10^{10,\,11,\,12,\,13}\,M_\odot$ @ $z=0$) + 2 merger histories (violent vs. quiescent)

- Common astrophysics all codes will include in high-resolution runs

  - Metal-dependent gas cooling, UV background, metal and energy yields for SNe, etc.
  - $\lesssim$100 pc resolution + high SF threshold : keys to realistic galaxies (e.g., *Eris* - Guedes+2011)

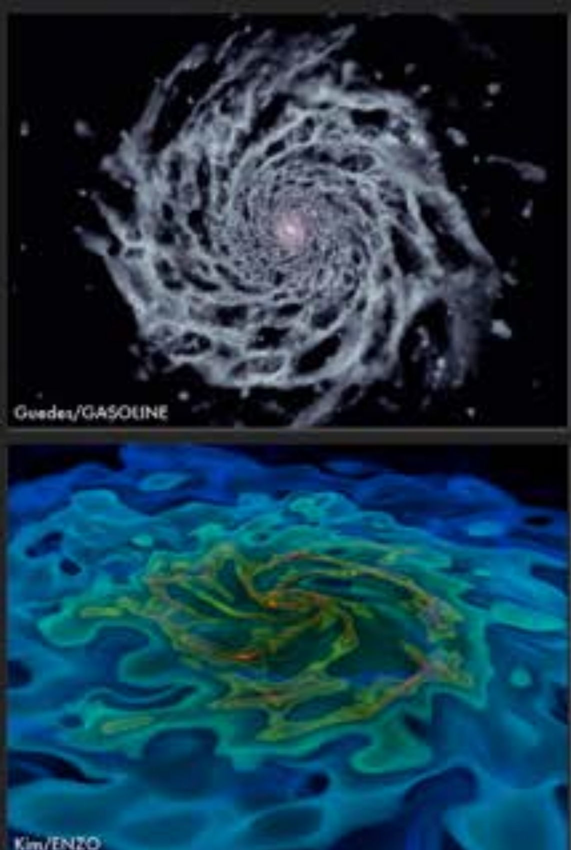
- Common analysis platform for all participating codes

  - Now possible on the yt-3.0 platform (Turk et al. 2011, Turk 2013, http://yt-project.org)

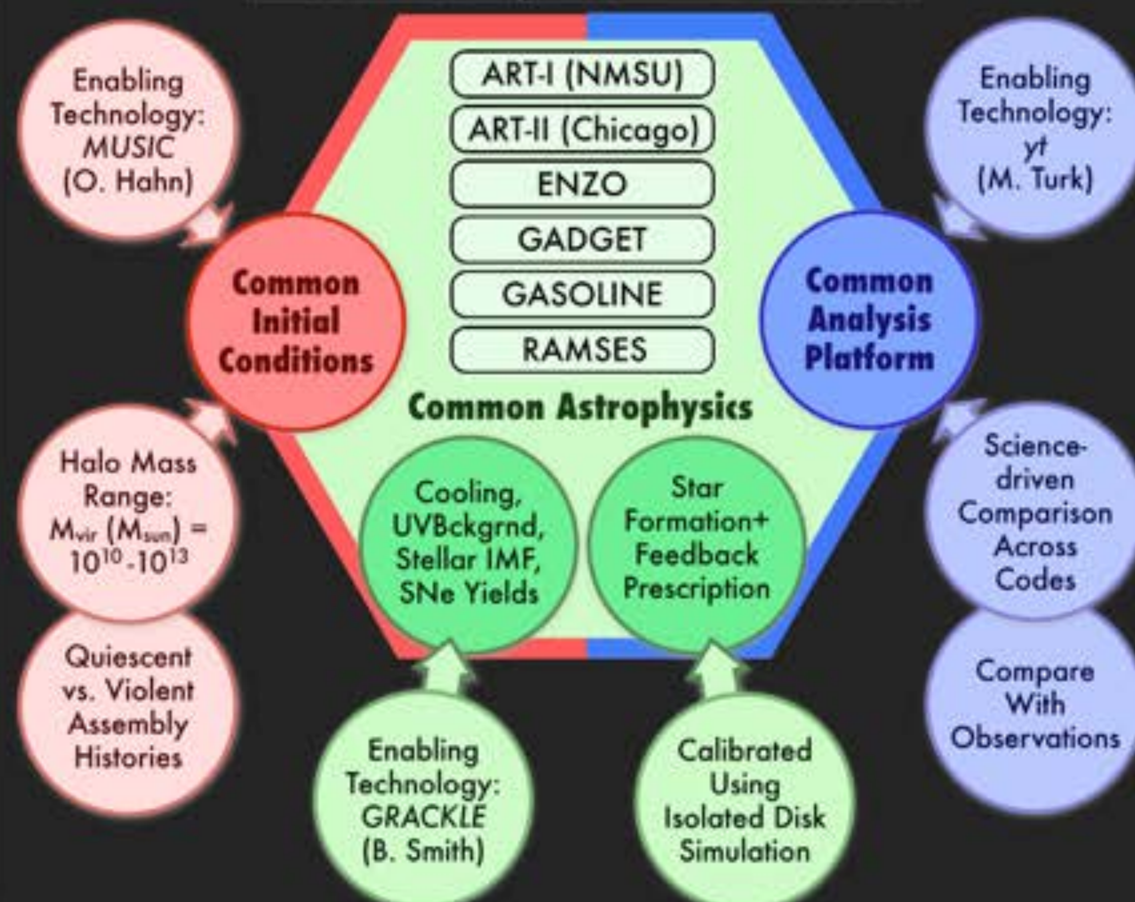
AGORA

A High-resolution Galaxy Simulations Comparison Initiative: www.AGORAsimulations.org

High-res Galaxy Simulations



AGORA Comparison Infrastructure



AGORA Goal & Team

- GOAL: A collaborative, multi-platform study to raise the realism and predictive power of galaxy formation simulations

- TEAM: 110 participants from 57 institutions worldwide, Aug. 2014

- DATA SHARING: Simulations outputs and analysis softwares will be shared with the community

- Contact: santacruzgalaxy@gmail.com
- AGORA First light: First paper by Ji-hoon Kim et al. (arXiv:1308.2669)
- Project funded in part by:

Official 1-page intro thanks to Kim and Governato

AGORA Initiative: Multi-platform Approach

- Common ICs publicly available via portable MUSIC parameters

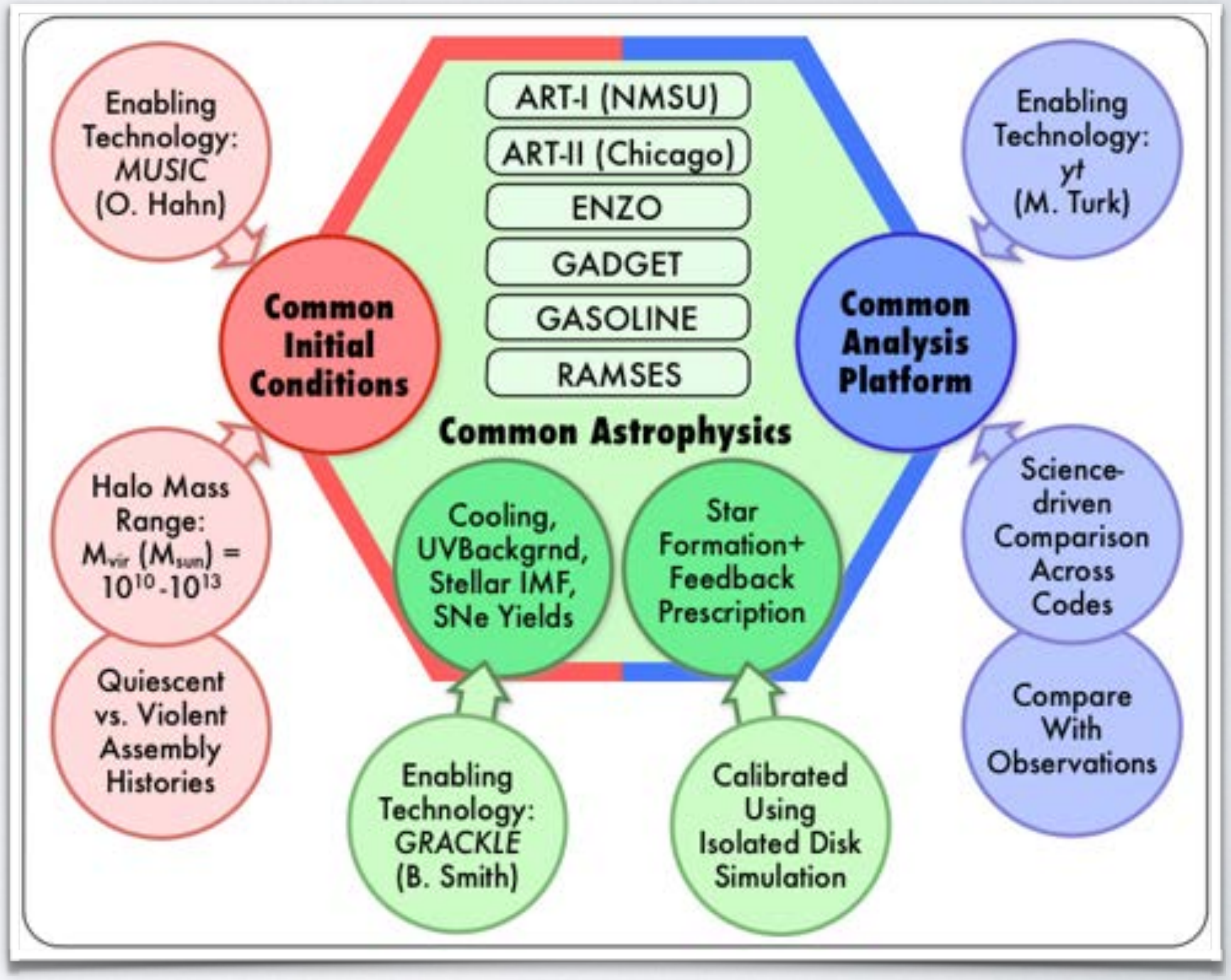
  - *Previously:* Reproducibility of numerical experiments has often been ignored

- Comparative analysis made easy by common analysis platform yt

  - *Previously:* Oftentimes one person had to try collecting analysis data while ensuring that the tests and analyses were identical across platforms

- Specifically designed with astrophysical questions in mind

  - *Previously:* Comparison ended when a single paper was out even though the framework could have been used to explore many other problems



AGORA Flagship Paper

- First light of the Project: Kim et al. published in January 2014

THE ASTROPHYSICAL JOURNAL SUPPLEMENT SERIES, 210:14 (20pp), 2014 January doi:10.1088/0067-0049/210/1/14

THE AGORA HIGH-RESOLUTION GALAXY SIMULATIONS COMPARISON PROJECT

JI-HOON KIM[1], TOM ABEL[2], OSCAR AGERTZ[3,4], GREG L. BRYAN[5], DANIEL CEVERINO[6], CHARLOTTE CHRISTENSEN[7], CHARLIE CONROY[1], AVISHAI DEKEL[8], NICKOLAY Y. GNEDIN[3,9,10], NATHAN J. GOLDBAUM[1], JAVIERA GUEDES[11], OLIVER HAHN[11], ALEXANDER HOBBS[11], PHILIP F. HOPKINS[12,13], CAMERON B. HUMMELS[7], FRANCESCA IANNUZZI[14], DUSAN KERES[15], ANATOLY KLYPIN[16], ANDREY V. KRAVTSOV[3,10], MARK R. KRUMHOLZ[1], MICHAEL KUHLEN[1,13], SAMUEL N. LEITNER[17], PIERO MADAU[1], LUCIO MAYER[18], CHRISTOPHER E. MOODY[1], KENTARO NAGAMINE[19,20], MICHAEL L. NORMAN[15], JOSE ONORBE[21], BRIAN W. O'SHEA[22], ANNALISA PILLEPICH[1], JOEL R. PRIMACK[23], THOMAS QUINN[24], JUSTIN I. READ[4], BRANT E. ROBERTSON[7], MIGUEL ROCHA[21], DOUGLAS H. RUDD[10,25], SIJING SHEN[1], BRITTON D. SMITH[22], ALEXANDER S. SZALAY[26], ROMAIN TEYSSIER[18], ROBERT THOMPSON[7,19], KEITA TODOROKI[19], MATTHEW J. TURK[5], JAMES W. WADSLEY[27], JOHN H. WISE[28], AND ADI ZOLOTOV[8]
FOR THE AGORA COLLABORATION[29]

ABSTRACT

We introduce the Assembling Galaxies Of Resolved Anatomy (AGORA) project, a comprehensive numerical study of well-resolved galaxies within the ΛCDM cosmology. Cosmological hydrodynamic simulations with force resolutions of ~100 proper pc or better will be run with a variety of code platforms to follow the hierarchical growth, star formation history, morphological transformation, and the cycle of baryons in and out of eight galaxies with halo masses $M_{\rm vir} \simeq 10^{10}$, 10^{11}, 10^{12}, and $10^{13}\,M_\odot$ at $z = 0$ and two different ("violent" and "quiescent") assembly histories. The numerical techniques and implementations used in this project include the smoothed particle hydrodynamics codes GADGET and GASOLINE, and the adaptive mesh refinement codes ART, ENZO, and RAMSES. The codes share common initial conditions and common astrophysics packages including UV background, metal-dependent radiative cooling, metal and energy yields of supernovae, and stellar initial mass function. These are described in detail in the present paper. Subgrid star formation and feedback prescriptions will be tuned to provide a realistic interstellar and circumgalactic medium using a non-cosmological disk galaxy simulation. Cosmological runs will be systematically compared with each other using a common analysis toolkit and validated against observations to verify that the solutions are robust—i.e., that the astrophysical assumptions are responsible for any success, rather than artifacts of particular implementations. The goals of the AGORA project are, broadly speaking, to raise the realism and predictive power of galaxy simulations and the understanding of the feedback processes that regulate galaxy "metabolism." The initial conditions for the AGORA galaxies as well as simulation outputs at various epochs will be made publicly available to the community. The proof-of-concept dark-matter-only test of the formation of a galactic halo with a $z = 0$ mass of $M_{\rm vir} \simeq 1.7 \times 10^{11}\,M_\odot$ by nine different versions of the participating codes is also presented to validate the infrastructure of the project.

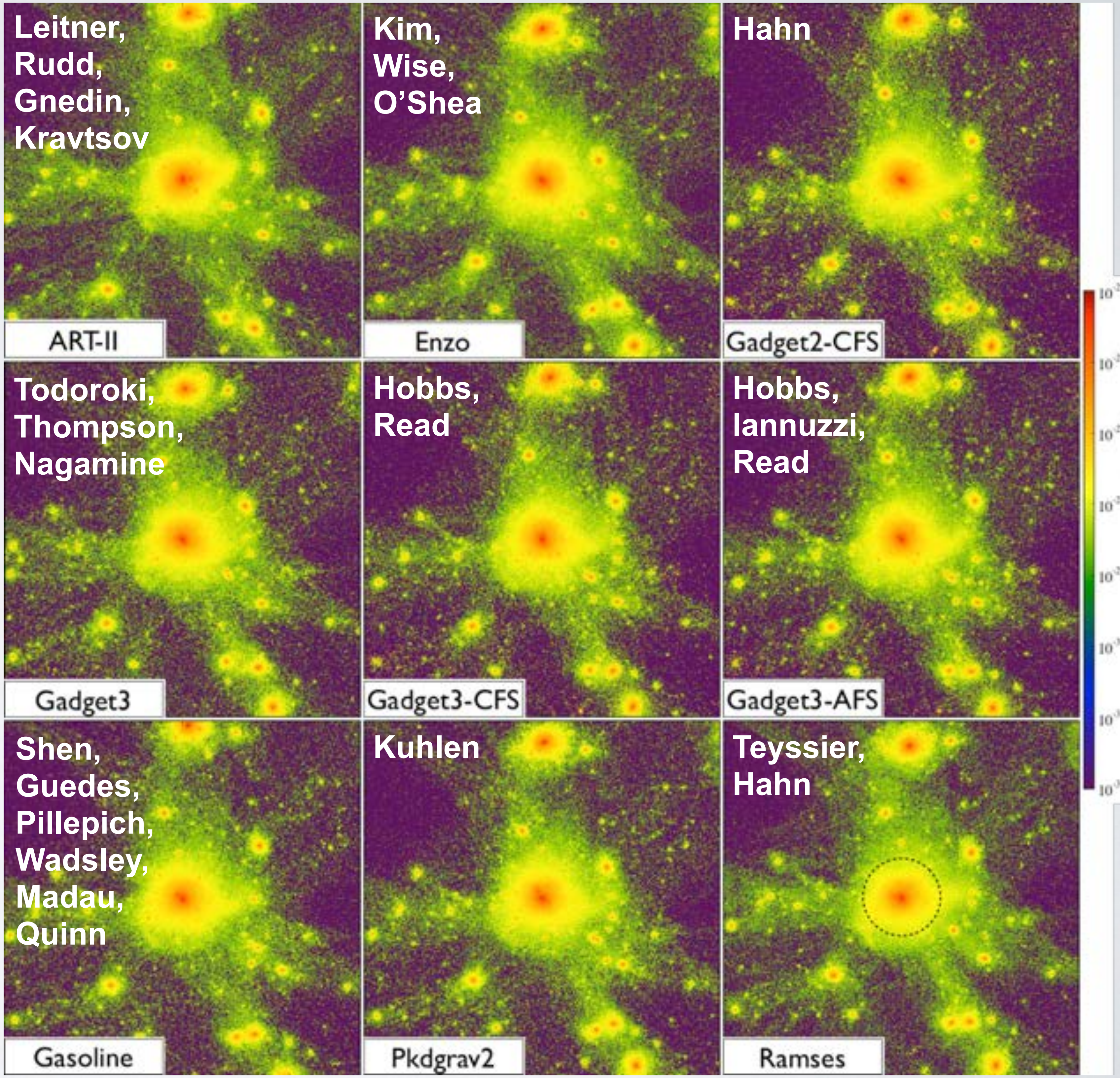
AGORA Proof-of-Concept Test

- DM-only test run

  - ~300 pc resolution runs by 9 different variations of 6 participating codes

  - Runtime parameters identified that make codes compatible with one another

  - ART-I (NMSU) has now also performed this test thanks to Ceverino, Klypin, Hahn



Projected DM density (1 Mpc/h), Kim et al. (2014)

AGORA ICs: Ready for Your Experiments

- Highly portable MUSIC ICs

  - A single parameter file is all you need.

  - No data conversion necessary between codes; therefore, the trickiest part of comparison is done!

- Already building a library of AGORA simulations making future comparisons trivially simple

  - Cosmological ICs open for use in any projects (AGORA or not) thanks to Hahn, Kim, Onorbe

  - Isolated Disk ICs also on the Workspace now thanks to Agertz, Teyssier, et al.

ics_AGORA_L12_1e11q_500kpc.conf_example

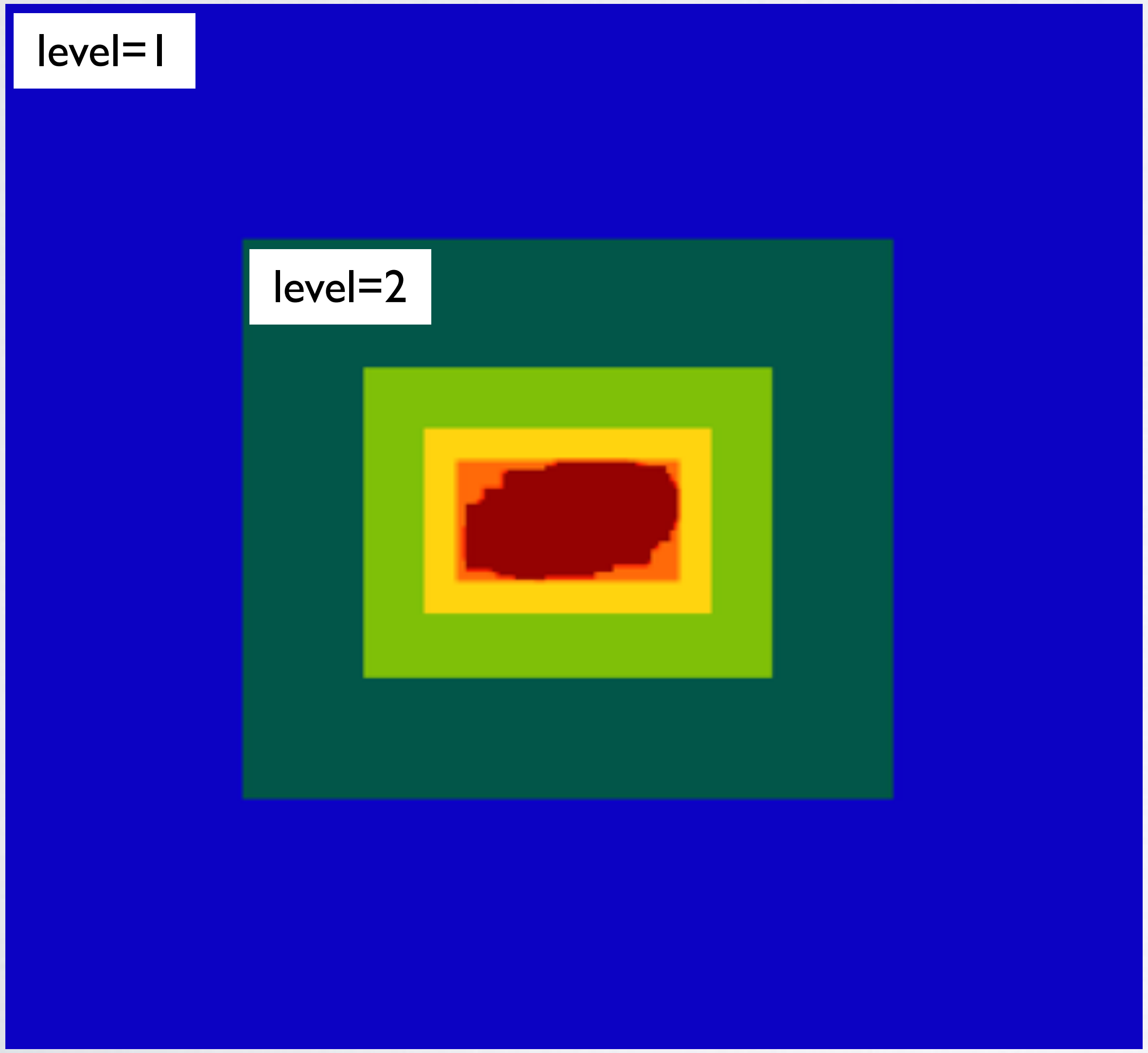
```
[setup]
boxlength                      = 60
zstart                         = 100
levelmin                       = 7
levelmin_TF                    = 9
levelmax                       = 12
padding                        = 16
overlap                        = 4
align_top                      = no
baryons                        = no
use_2LPT                       = no
use_LLA                        = no
region                         = ellipsoid
region_ellipsoid_matrix[0]     = 2710.833984, -498.042755, -260.366791
region_ellipsoid_matrix[1]     = -498.042755, 1496.330933, 864.111267
region_ellipsoid_matrix[2]     = -260.366791, 864.111267, 5030.364746
region_ellipsoid_center        = 0.638273, 0.576312, 0.447929

[cosmology]
Omega_m                        = 0.272
Omega_L                        = 0.728
Omega_b                        = 0.0455
H0                             = 70.2
sigma_8                        = 0.807
nspec                          = 0.961
transfer                       = eisenstein
#YHe                           = 0.248
#gamma                         = 1.6667

[random]
cubesize                       = 256
seed[8]                        = 95064
seed[9]                        = 31415
seed[10]                       = 27183

[output]
## ENZO
format                  = enzo
filename                = ic.enzo
#enzo_refine_region_fraction = 0.8

## TIPSY
#format                 = tipsy
#filename               = ics_tipsy.dat
##tipsy_eps             = 0.02
##tipsy_native          = no

[poisson]
fft_fine                = yes
accuracy                = 1e-6
grad_order              = 6
laplace_order           = 6
```

MUSIC parameter file with region_ellipsoid

MUSIC Improvements: Ellipsoidal ICs

- IC with a highest-resolution minimum-volume ellipsoid enclosing the Lagrangian region now available, thanks to Hahn et al.

  - Helps to reduce computational costs in large calculations for future projects
  - To be employed in Papers “2” and “3” (as opposed to cuboidal ICs used in the Flagship paper)



Paper “2” - 1×10^{11} $M_\odot$ DM only, with resolution < 100 pc

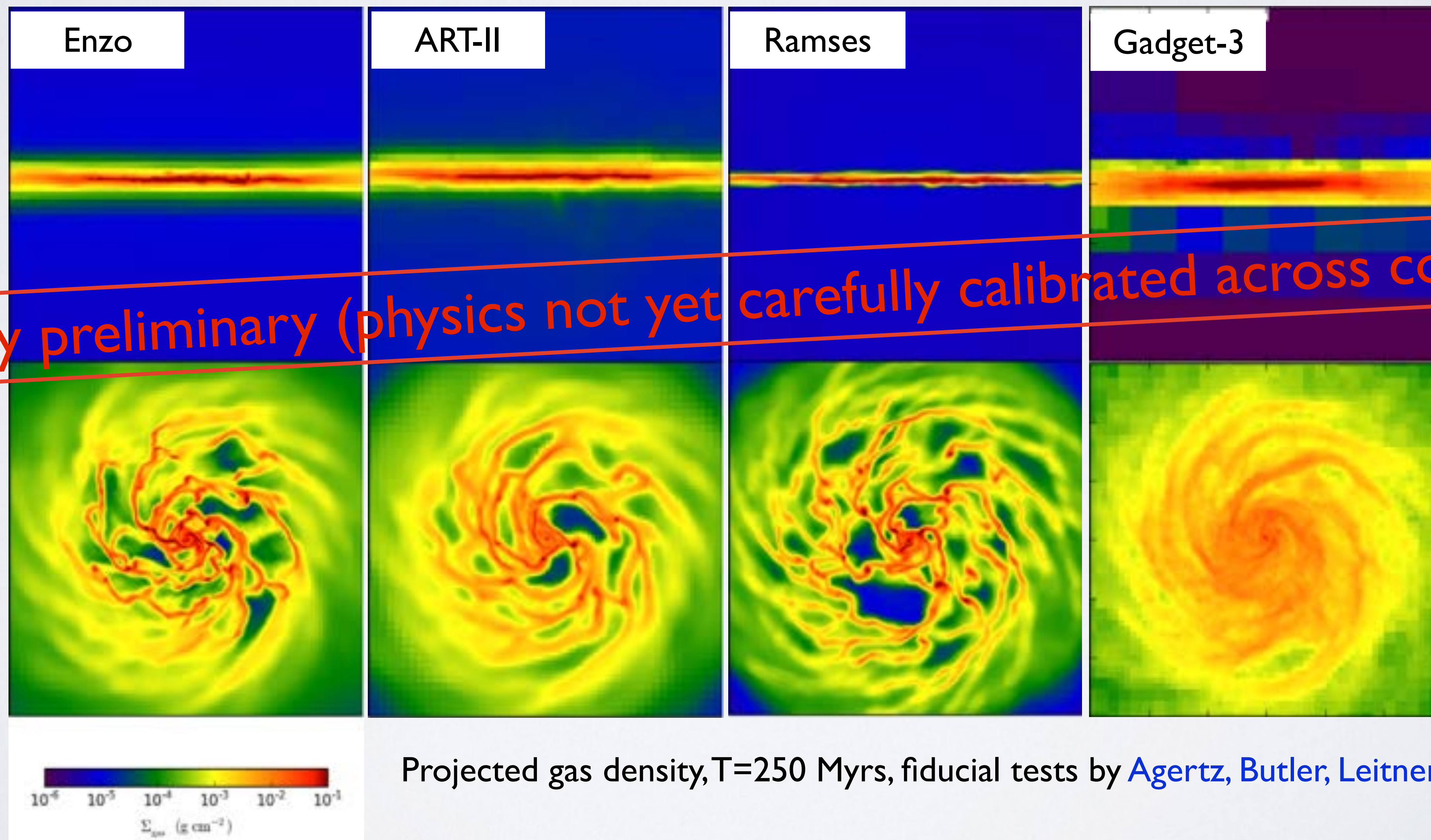
Paper “3” - 1×10^{11} $M_\odot$ DM+hydro including Grackle cooling code

Paper “4” - Calibrate common SF/FB recipes using isolated disk IC

Maximum grid level along LOS, z=50, Enzo test by Simpson et al.

Isolated Disk ICs: Built and Running

- ICs of a MW-size isolated disk galaxy (built w/ Springel's MakeDisk) now on Workspace, thanks to Agertz, Teyssier, et al.
  - 4-component galaxy, 3 resolution choices (low/med/high), to be employed in Paper "4"



Only preliminary (physics not yet carefully calibrated across codes)

Projected gas density, T=250 Myrs, fiducial tests by Agertz, Butler, Leitner, Todoroki, et al.

yt-3.0 Just Released: Aug. 4, 2014

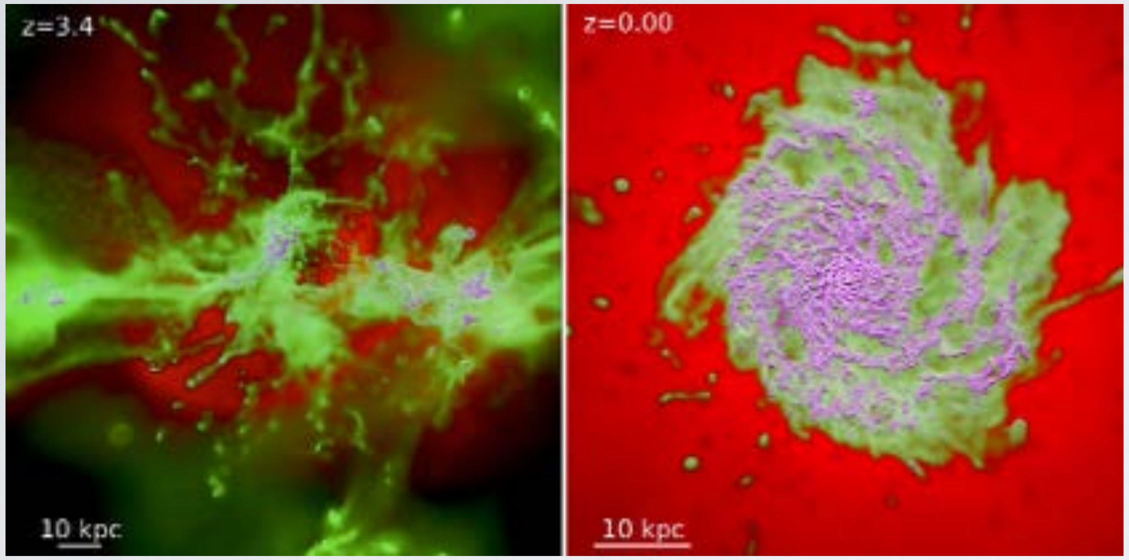
- Supports many codes, including all participating in AGORA

… This release of yt features an entirely rewritten infrastructure for data ingestion, indexing, and representation. While past versions of yt were focused on analysis and visualization of data structured as regular grids, this release features full support for particle (discrete point) data such as N-body and SPH data, irregular hexahedral mesh data, and data organized via octrees. This infrastructure will be extended in future versions for high-fidelity representation of unstructured mesh datasets.

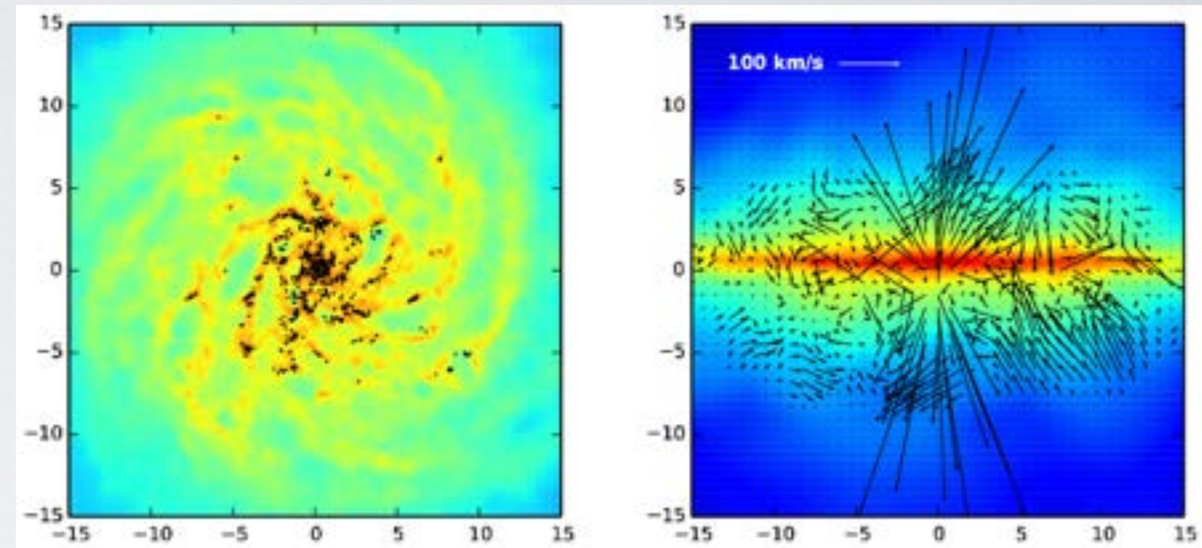
Highlighted changes in yt 3.0:

- Units now permeate the code base, enabling self-consistent unit transformations of all arrays and quantities returned by yt.
- Particle data is now supported using a lightweight octree. SPH data can be smoothed onto an adaptively-defined mesh using standard SPH smoothing
- Support for octree AMR codes
- Preliminary Support for non-Cartesian data, such as cylindrical, spherical, and geographical
- Revamped analysis framework for halos and halo catalogs, including direct ingestion and analysis of halo catalogs of several different formats
- Support for multi-fluid datasets and datasets containing multiple particle types
- Flexible support for dynamically defining new particle types using filters on existing particle types or by combining different particle types.
- Vastly improved support for loading generic grid, AMR, hexahedral mesh, and particle without hand-coding a frontend for a particular data format.
- New frontends for ART, ARTIO, Boxlib, Chombo, FITS, GDF, Subfind, Rockstar, Pluto, RAMSES, SDF, Gadget, OWLS, PyNE, Tipsy, as well as rewritten frontends for Enzo, FLASH, Athena, and generic data.
- First release to support installation of yt on Windows
- Extended capabilities for construction of simulated observations, and new facilities for analyzing and visualizing FITS images and cube data
- Many performance improvements

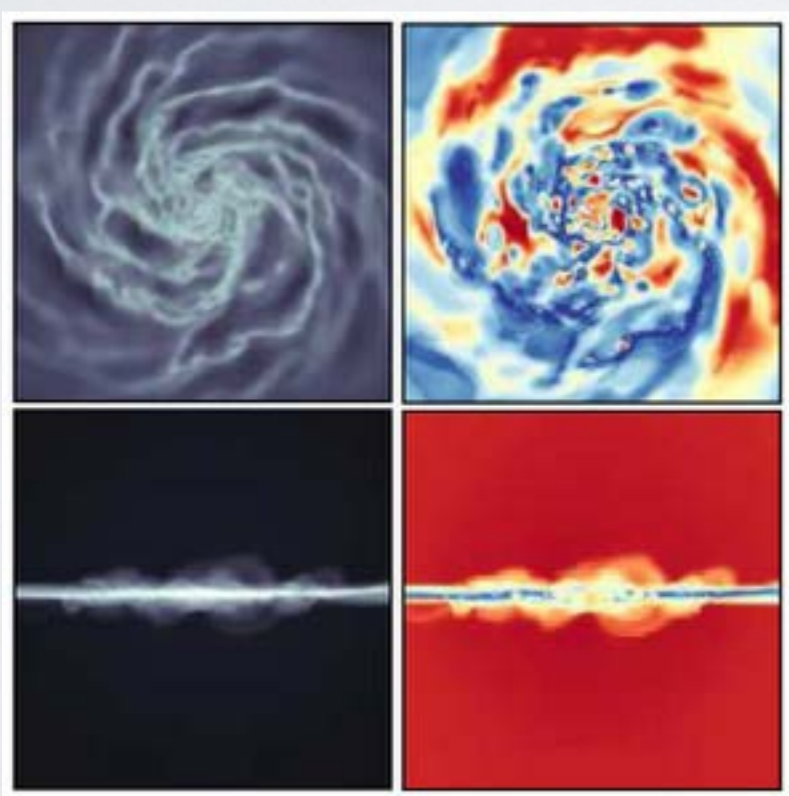
Researches Using AGORA ICs



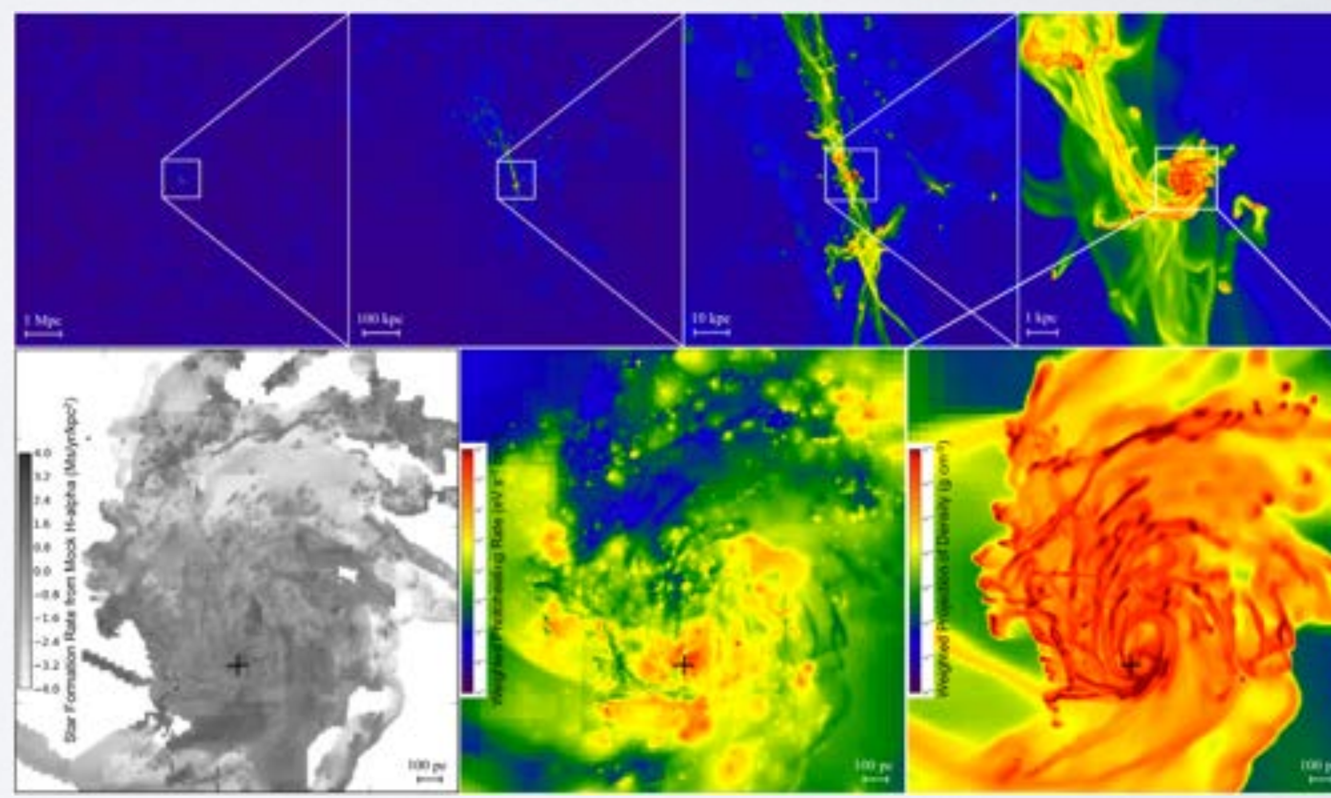
Hopkins et al. 2013, van de Voort et al. 2014,
MW-size halo (~10^{12} $M_\odot$ @z=0) with 50 pc/h softening



Keller et al. 2014,
MW-size halo (1.3×10^{12} $M_\odot$ @z=0) with 20 pc softening



Agertz et al. 2013,
MW-size halo with 70 pc resolution



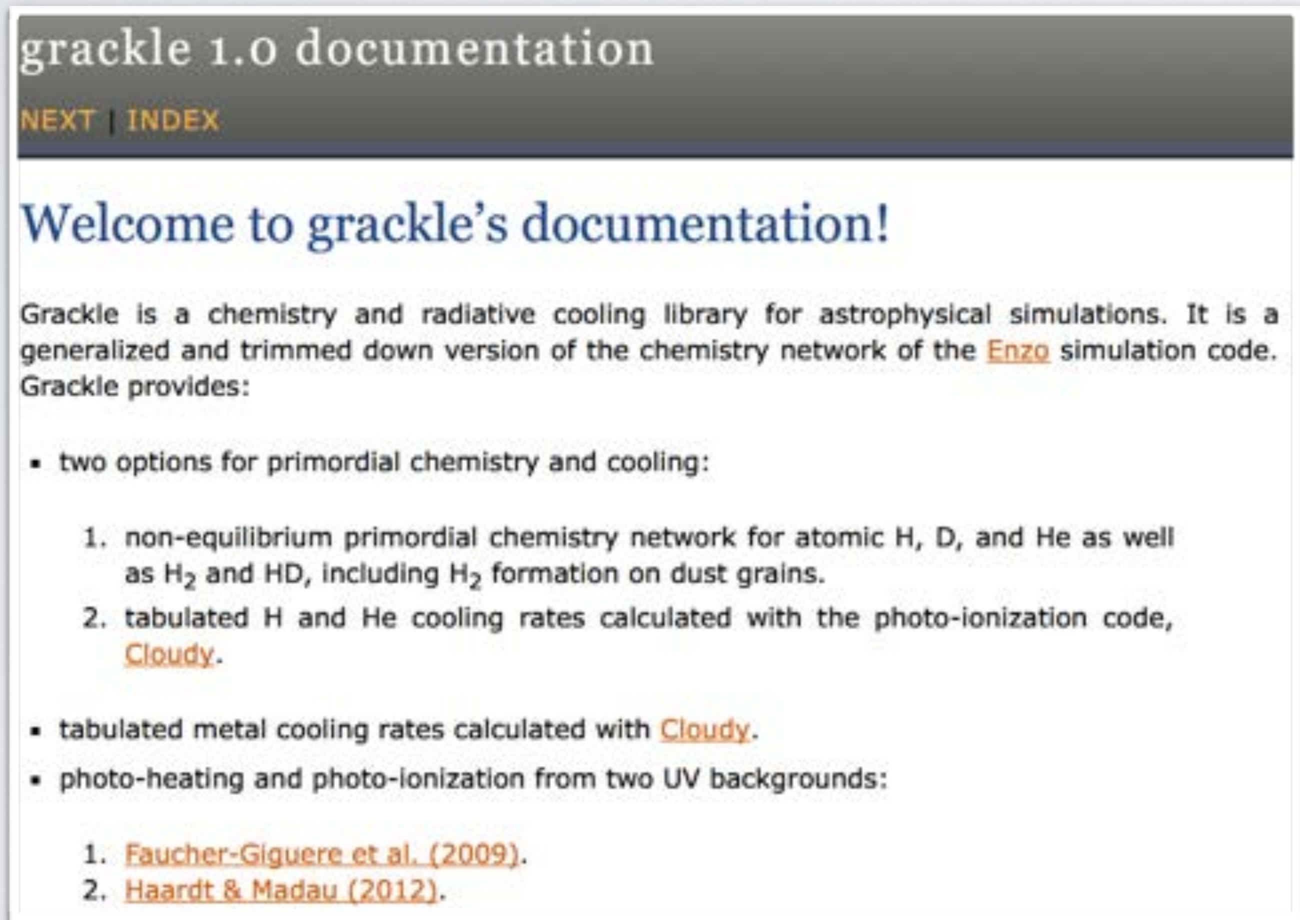
Kim et al. 2014 in prep., high-z quasar host
(7×10^{10} $M_\odot$ @z~7.5 → ~10^{13} $M_\odot$ @z=0) with 4.8 pc resolution

GRACKLE: Ready and Implemented

- Now available in two modes: non-equilibrium and equilibrium

  - Implementation completed or underway in all codes participating in AGORA thanks to Smith, Turk, Gnedin, Agertz, Choi, Nagamine, Teyssier, Madau, Shen, Quinn, Hummels, et al.

  - Fortran and C APIs now publicly available in what will soon become GRACKLE 2.0



grackle 1.0 documentation

NEXT | INDEX

Welcome to grackle's documentation!

Grackle is a chemistry and radiative cooling library for astrophysical simulations. It is a generalized and trimmed down version of the chemistry network of the Enzo simulation code. Grackle provides:

- two options for primordial chemistry and cooling:
  1. non-equilibrium primordial chemistry network for atomic H, D, and He as well as H_2 and HD, including H_2 formation on dust grains.
  2. tabulated H and He cooling rates calculated with the photo-ionization code, Cloudy.
- tabulated metal cooling rates calculated with Cloudy.
- photo-heating and photo-ionization from two UV backgrounds:
  1. Faucher-Giguere et al. (2009).
  2. Haardt & Madau (2012).

GRACKLE documentation: http://grackle.readthedocs.org

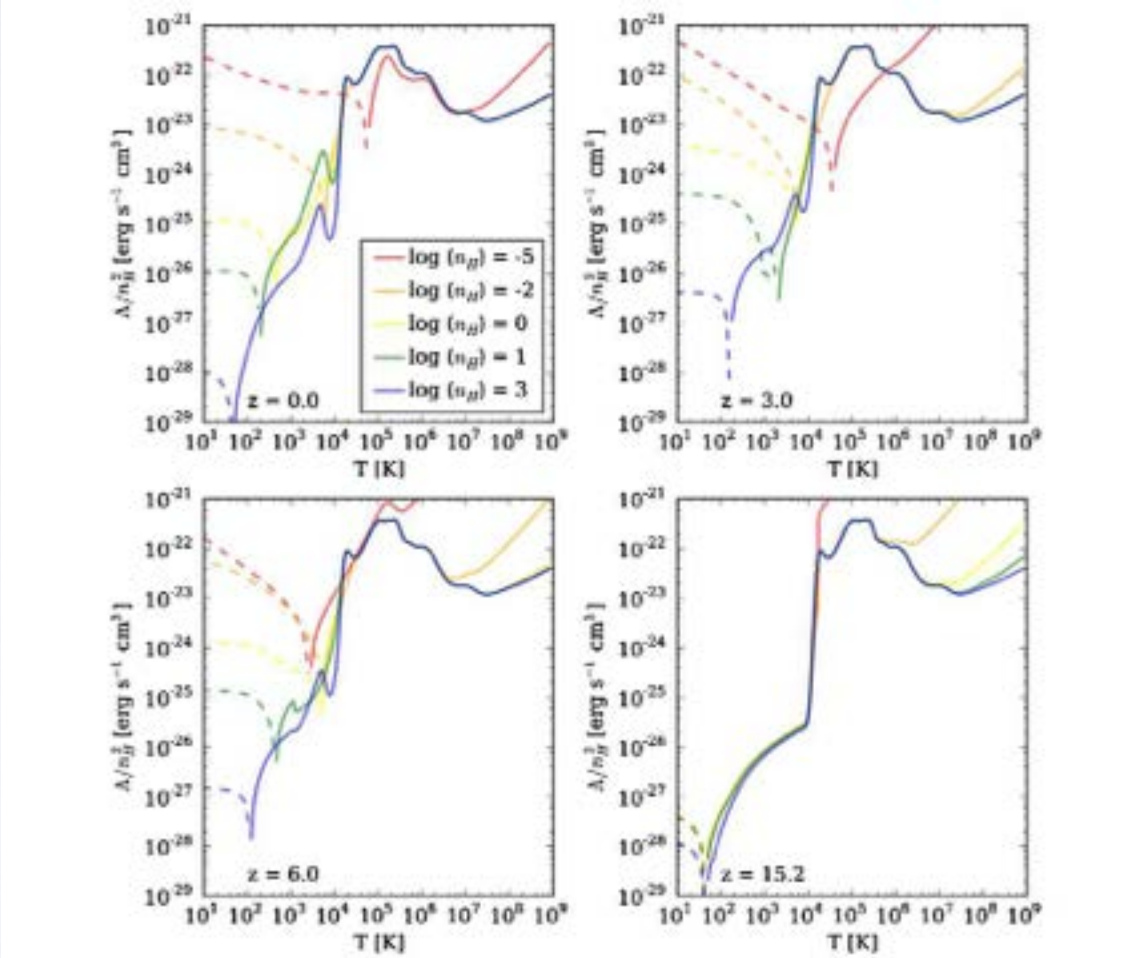
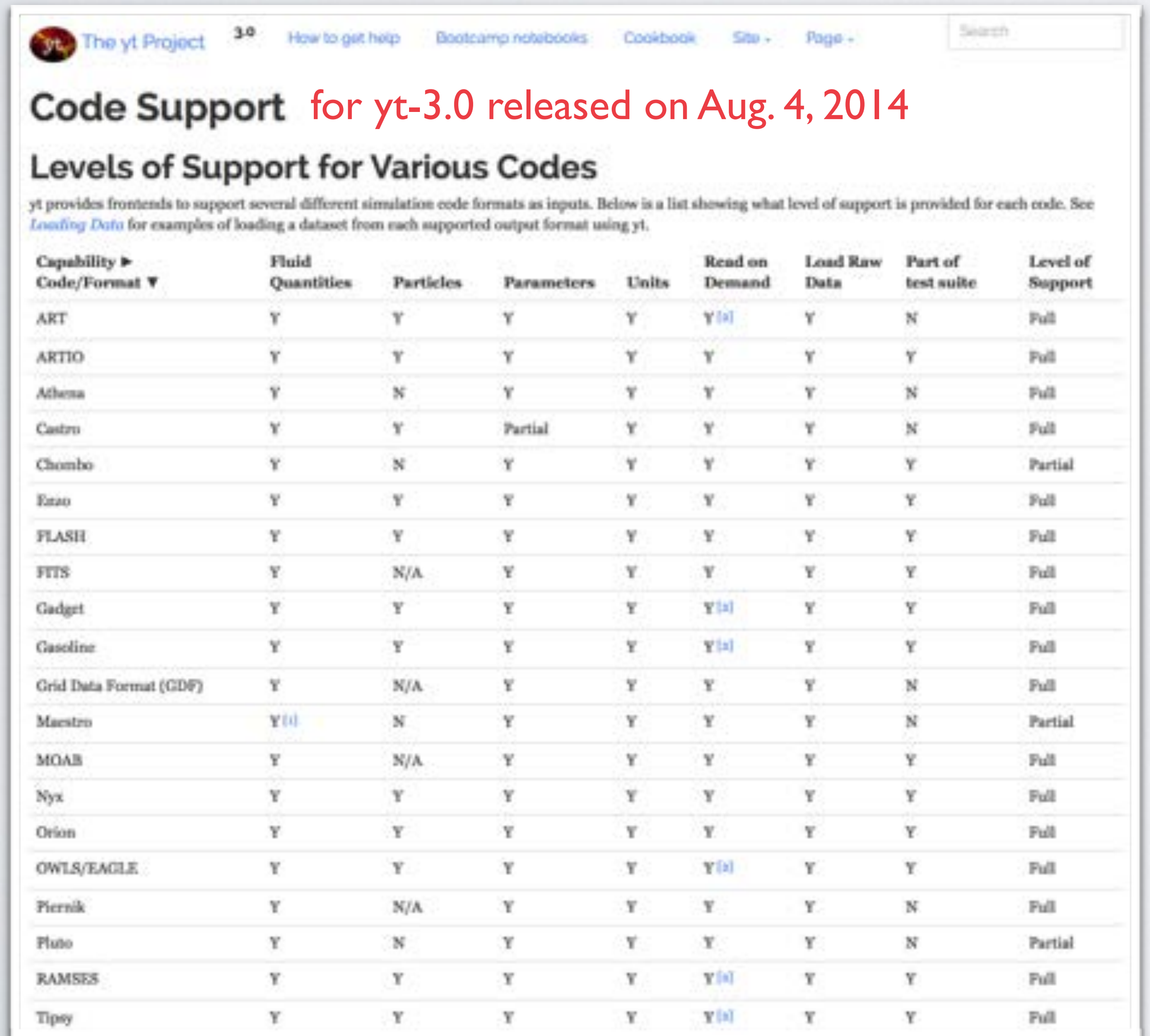


Figure 1. Gas cooling in the AGORA simulations. Equilibrium cooling rates normalized by n_H^2 calculated with the GRACKLE cooling library for 11 number densities of 10^{-5} (red), 10^{-2} (orange), 1 (yellow), 10 (green), and 10^3 (blue) cm^{-3} at redshifts z = 0, 3, 6, and 15.2 (just before the UV background turns on) and solar metallicity gas. Solid lines denote net cooling and dashed lines denote net heating. The curves plotted are made with the non-equilibrium chemistry network of H, He, H_2, and HD with tabulated metal cooling assuming the presence of a UV metagalactic background from Haardt & Madau (2012).

Gas cooling in AGORA, Kim et al. (2014)

yt Analysis Toolkit: Ever Improving

- Supports many codes, including all participating in AGORA
  - SPH code support was among main topics at the yt-AGORA mini-workshop (Mar. 23-25, 2014 @UCSC), organized by Turk, Rocha, Bogert, and Goldbaum
  - yt-3.0 released thanks to AGORA team: Arraki (ART-I), Rudd (ART-II), Leitner (ART-II), Altay (Gadget), Keller (Gasoline), Hummels (Enzo), Turk (All), etc.



The yt Project 3.0 How to get help Bootcamp notebooks Cookbook Site Page Search

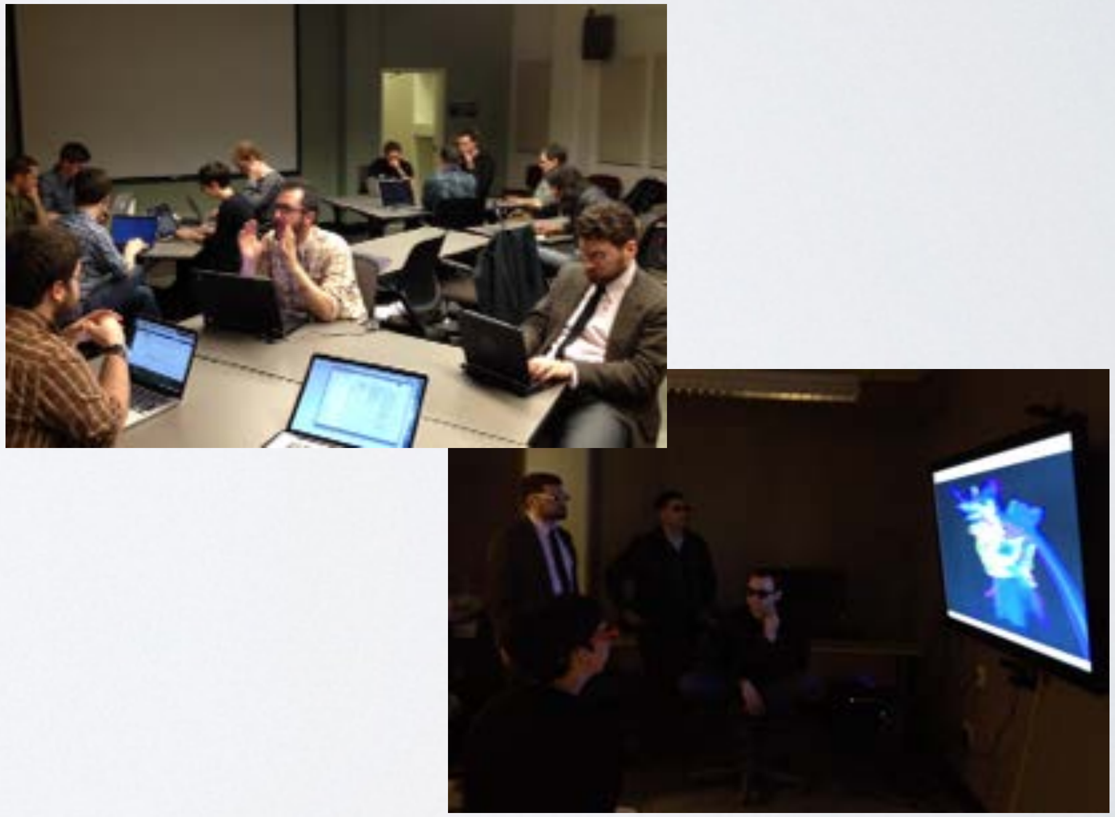
Code Support for yt-3.0 released on Aug. 4, 2014

Levels of Support for Various Codes

yt provides frontends to support several different simulation code formats as inputs. Below is a list showing what level of support is provided for each code. See *Loading Data* for examples of loading a dataset from each supported output format using yt.

Capability ► Code/Format ▼	Fluid Quantities	Particles	Parameters	Units	Read on Demand	Load Raw Data	Part of test suite	Level of Support
ART	Y	Y	Y	Y	Y [a]	Y	N	Full
ARTIO	Y	Y	Y	Y	Y	Y	Y	Full
Athena	Y	N	Y	Y	Y	Y	N	Full
Castro	Y	Y	Partial	Y	Y	Y	N	Full
Chombo	Y	N	Y	Y	Y	Y	Y	Partial
Enzo	Y	Y	Y	Y	Y	Y	Y	Full
FLASH	Y	Y	Y	Y	Y	Y	Y	Full
FITS	Y	N/A	Y	Y	Y	Y	Y	Full
Gadget	Y	Y	Y	Y	Y [a]	Y	Y	Full
Gasoline	Y	Y	Y	Y	Y [a]	Y	Y	Full
Grid Data Format (GDF)	Y	N/A	Y	Y	Y	Y	N	Full
Maestro	Y [1]	N	Y	Y	Y	Y	N	Partial
MOAB	Y	N/A	Y	Y	Y	Y	Y	Full
Nyx	Y	Y	Y	Y	Y	Y	Y	Full
Orion	Y	Y	Y	Y	Y	Y	Y	Full
OWLS/EAGLE	Y	Y	Y	Y	Y [a]	Y	Y	Full
Piernik	Y	N/A	Y	Y	Y	Y	N	Full
Pluto	Y	N	Y	Y	Y	Y	N	Partial
RAMSES	Y	Y	Y	Y	Y [a]	Y	Y	Full
Tipsy	Y	Y	Y	Y	Y [a]	Y	Y	Full

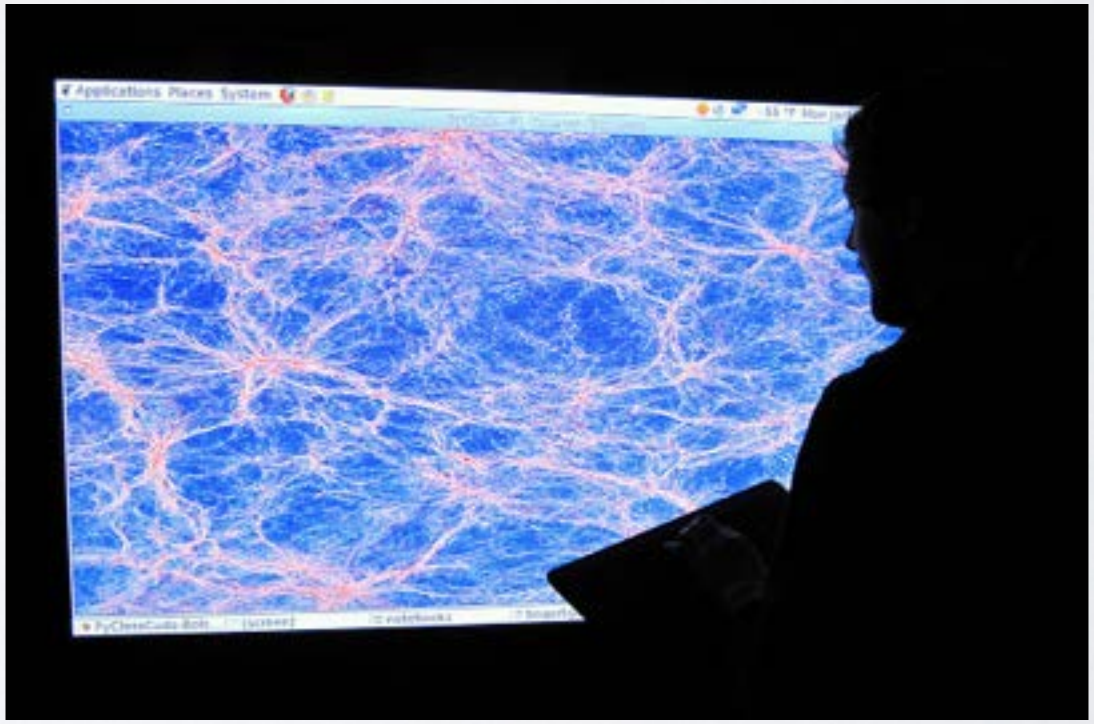
Talk to other yt users in AGORA:
Benincasa (Gasoline), Butler (Ramses), DeGraf (Ramses)



yt-AGORA mini-workshop@UCSC (Mar. 2014)

AGORA Analysis Pipeline on NERSC

- AGORA data analysis pipeline using yt has been built on NERSC thanks to Rocha, Bogert, Steffens, Turk, et al.
    - GPU volume rendering possible with yt thanks to Bogert, Turk, et al.
    - Remotely rendered images can be streamed via iPython notebook or flash video streaming (e.g. images rendered on Kepler cards on NERSC, then streamed to your laptop in real time)
    - yt output can also be fed into SUNRISE thanks to Moody, Turk, et al.
    - As AGORA members on NERSC, you don't need to install anything! (Rocha's talk on Sun)



From Bogert et al.'s GTC poster; see WG IV Workspace page for more information

AGORA Mass Storage on NERSC

- Data Pilot Program allocation by NERSC (PI: Primack, Madau)

  - To be used as one of the mass storages for AGORA

  - 5M cpu-hours (XT4-equivalent MPP hours for data analysis, mainly with yt and SUNRISE) + 0.6M storage resource units (SRUs) enough to transfer ~100 TB in and out of their HPSS

  - Storage and managing policies have been established during the web conference on Mar. 25, 2014 (talk to Rocha about NERSC system access)

Cray XE6 Hopper @NERSC

Cray XC30 Edison @NERSC

Upcoming Papers This Year

- Papers "2", "3" and "4": main agenda for this weekend sessions

<table>
<tr><th></th><th>PAPER "2"</th><th>PAPER "3"</th><th>PAPER "4"</th></tr>
<tr><td>GOALS</td><td>• Run Proof-of-concept-DM (~1e11 Ms halo)</td><td>• Implement and test common astrophysics package
• Run Proof-of-concept-NOSUBGRID (~1e11 Ms halo)</td><td>• Calibrate SF/Fbck recipes using common disk ICs</td></tr>
<tr><td>LEADERS</td><td>• Oliver Hahn (ETH)</td><td>• Nick Gnedin (Fermilab)
• Piero Madau (UCSC)
• Britton Smith (Edinburgh)</td><td>• Oscar Agertz (Surrey)
• Romain Teyssier (Zurich)
• James Wadsley (McMaster)</td></tr>
<tr><td>PARTICIPANTS</td>
<td>• Working Groups engaged:
– III (common cosmo ICs)
– IV (common analysis)
– VI (dwarf galaxies)

• People in charge:
<table>
<tr><th>Code</th><th>Contacts</th></tr>
<tr><td>ART-I</td><td>Ceverino, Klypin</td></tr>
<tr><td>ART-II</td><td>Gnedin, (Leitner)</td></tr>
<tr><td>CHANGA</td><td>Quinn</td></tr>
<tr><td>ENZO</td><td>Kim, Simpson, Wise, et al.</td></tr>
<tr><td>GADGET</td><td>Hahn, Hobbs, Nagamine, Todoroki</td></tr>
<tr><td>GASOLINE</td><td>Shen</td></tr>
<tr><td>PKDGRAV2</td><td>(Guedes, Kuhlen)</td></tr>
<tr><td>RAMSES</td><td>Teyssier</td></tr>
</table></td>
<td>• Working Groups engaged:
– I (common physics)
– III (common cosmo ICs)

• People in charge:
<table>
<tr><th>Code</th><th>Contacts</th></tr>
<tr><td>ART-I</td><td>Ceverino, Klypin</td></tr>
<tr><td>ART-II</td><td>Gnedin, (Letiner)</td></tr>
<tr><td>CHANGA</td><td>Quinn</td></tr>
<tr><td>ENZO</td><td>Hummels, Kim, Simpson, Smith, Turk, Wise</td></tr>
<tr><td>GADGET</td><td>Choi, Hobbs, Nagamine, Thompson, Todoroki</td></tr>
<tr><td>GASOLINE</td><td>Madau, Mayer, Shen, Wadsley</td></tr>
<tr><td>RAMSES</td><td>Agertz, Teyssier</td></tr>
</table></td>
<td>• Working Groups engaged:
– II (common disk ICs)
– IV (common analysis)
– V (subgrid physics)

• People in charge:
<table>
<tr><th>Code</th><th>Contacts</th></tr>
<tr><td>ART-I</td><td>Klypin</td></tr>
<tr><td>ART-II</td><td>Gnedin, (Leitner)</td></tr>
<tr><td>ENZO</td><td>Butler, Goldbaum, et al.</td></tr>
<tr><td>GADGET</td><td>Hobbs, Nagamine, Thompson, Todoroki</td></tr>
<tr><td>GASOLINE</td><td>Shen, Wadsley, Benincasa</td></tr>
<tr><td>RAMSES</td><td>Agertz, Teyssier, Nickerson</td></tr>
</table></td></tr>
</table>

AGORA Science: Much More to Come

- Launchpad to initiate a series of science-oriented comparisons

  - 4 task-oriented working groups to establish the framework of the Project

Table 1
Task-oriented Working Groups of the *AGORA* Project

Working Group	Objectives and Tasks†
Common Cosmological ICs	Determine common initial conditions for cosmological high-resolution zoom-in galaxies (Section 2.1)
Common Isolated ICs	Determine common initial conditions for an isolated low-redshift disk galaxy (Section 2.2)
Common Astrophysics	Define common physics including UV background, gas cooling, stellar IMF, energy and metal yields from SNe (Section 3)
Common Analysis	Support common analysis tools, define physical and quantitative comparisons across all codes (Section 4.2)

WG Leaders

Hahn, Kim, Onorbe
Teyssier
Madau, Smith, Gnedin
Turk

  - 9(+) science-oriented working groups to perform original research using the framework

Table 2
Science-oriented Working Groups of the *AGORA* Project

Working Group	Science Questions (includes, but are not limited to)†
Isolated Galaxies and Subgrid Physics	Tune subgrid models across codes to yield similar results for similar astrophysical assumptions (Section A.1)
Dwarf Galaxies	Simulate cosmological $\sim 10^{10}$ $M_\odot$ halos and compare results across all participating codes (Section A.2)
Dark Matter	Radial profile, shape, substructure, core-cusp problem (Section A.3)
Satellite Galaxies	Effects of environment, UV background, tidal disruption (Section A.4)
Galactic Characteristics	Surface brightness, disks, bulges, metallicity, images, spectral energy distributions (Section A.5)
Outflows	Galactic outflows, circumgalactic medium, metal absorption systems (Section A.6)
High-redshift Galaxies	Cold flows, clumpiness, kinematics, Lyman-limit systems (Section A.7)
Interstellar Medium	Galactic ISM, thermodynamics, kinematics (Section A.8)
Massive Black Holes	Growth and feedback of massive black holes in a galactic context (Section A.9)

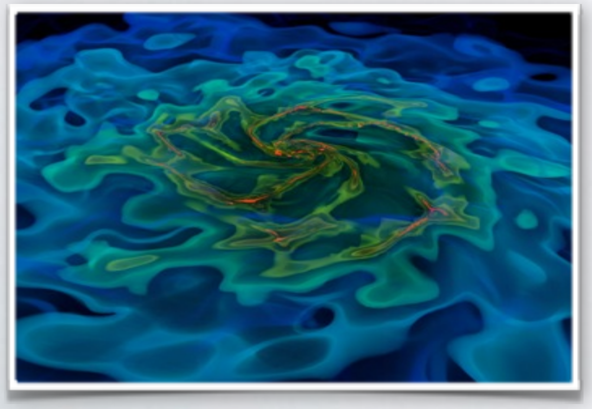
WG Leaders

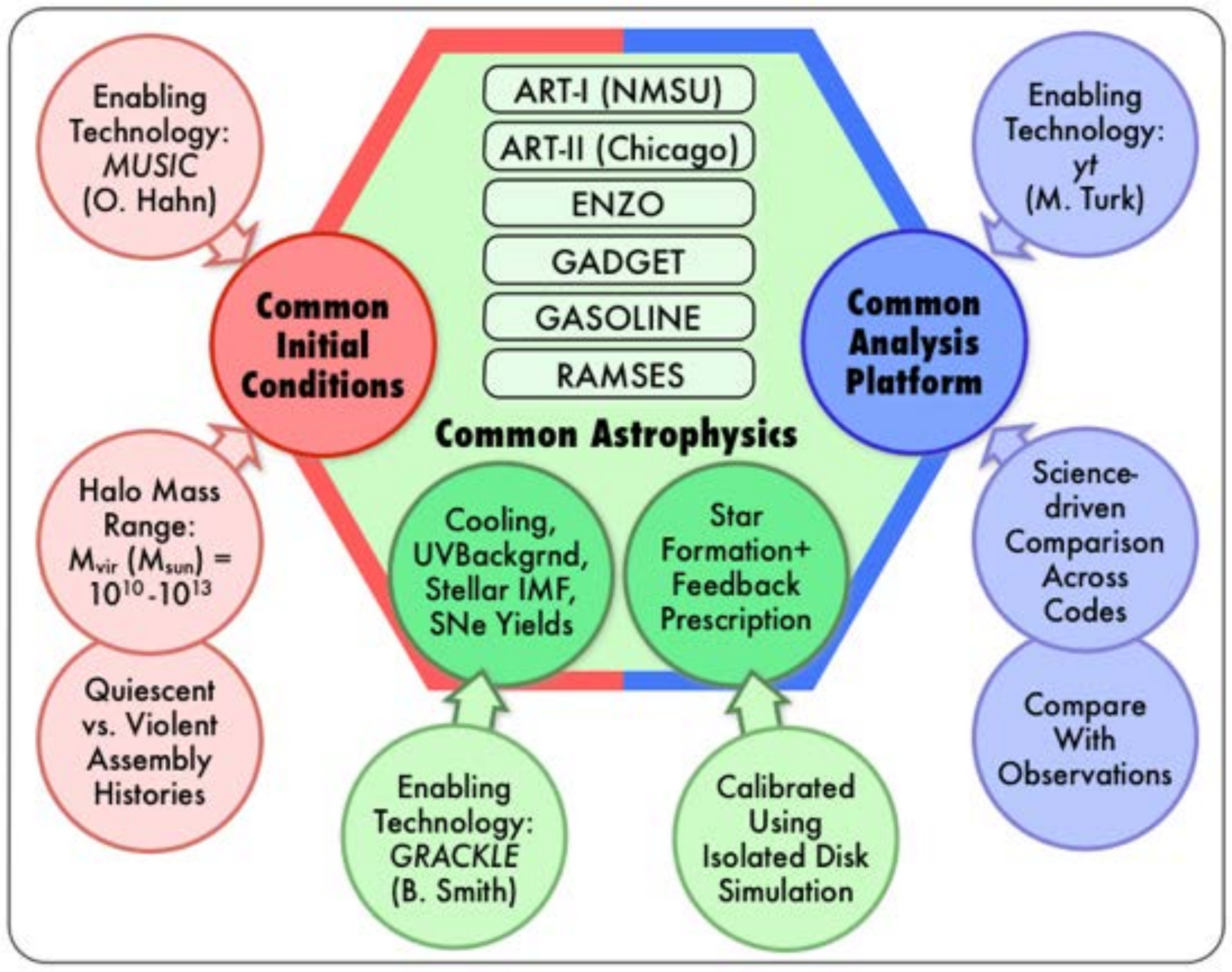
Agertz & Teyssier
Onorbe
Rocha
Zolotov
Hummels
Shen
Ceverino
Leitner
Hobbs, DeGraf

Comparing Resolved Galaxies Across Code Platforms

- The AGORA high-resolution galaxy simulations comparison project provides a unique opportunity to validate our answers to long-standing problems in galaxy formation

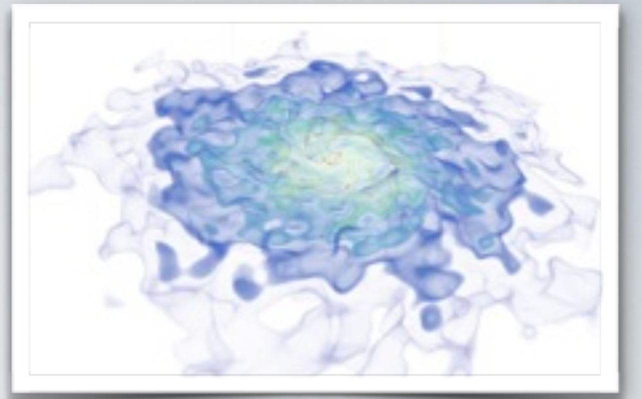
- We are at a critical junction in numerically understanding the process of galaxy formation and evolution





Enabling Technology: MUSIC (O. Hahn)
Halo Mass Range: Mvir (Msun) = 10^10-10^13
Quiescent vs. Violent Assembly Histories
Common Initial Conditions
ART-I (NMSU)
ART-II (Chicago)
ENZO
GADGET
GASOLINE
RAMSES
Common Astrophysics
Cooling, UVBackgrnd, Stellar IMF, SNe Yields
Star Formation+ Feedback Prescription
Enabling Technology: GRACKLE (B. Smith)
Calibrated Using Isolated Disk Simulation
Common Analysis Platform
Enabling Technology: yt (M. Turk)
Science-driven Comparison Across Codes
Compare With Observations

Thank you!



[Supplemental Slides]

Interactive Visualization of Astrophysical Data

Alex Bogert, John Holdener, and Nicholas Smith

Departments of Physics and Computer Science, UCSC

GPU Technology Conference

The yt-project

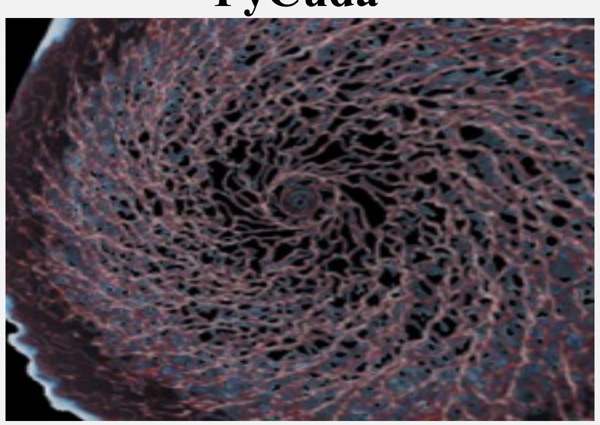
yt [1] is an analysis and visualization system for astrophysical volumetric data that is openly developed and freely available. At its core, yt provides a method of describing physical rather than computational objects inside an astrophysical simulation. yt provides methods for selecting regions, applying analysis to regions, visualizing (including volume rendering, projections, slices, phase plots) and exporting data to external analysis packages.

As an example, Adaptive Mesh Refinement (AMR) data consists of cells or grid patches that may be refined to higher-resolution and overlap with coarser objects. yt, when invoked on an AMR dataset, will transparently select data by masking the coarse cells, converting code units to physical units, process that data, optionally decomposing to multiple processors for parallel analysis, and then return the reduced data product or visualization to the user. yt allows astrophysicists to think about physical questions that drive their research, rather than necessary computational and technical steps to answer these questions.

yt has been designed to enable separation between data indexing and selection (such as patch-based AMR, N-body datasets, octrees, and irregular or unstructured meshes) and the processing of data to produce astrophysically-meaningful outputs. This enables researchers to utilize identical routines for generating simulated spectra and observations independent of the underlying simulation code, and provides direct cross-code comparisons and technology transfer. A fundamental component of yt is its investment in community: by providing different levels of contribution (infrastructure, code interfaces, analysis modules, documentation, community engagement) it has scaled to contributions from 50 different people, with an active development process and user community from across the globe.

Originally applied to astrophysical simulations, yt has been extended to work with radio telescope data, seismological simulations, high energy-density physics data, and nuclear engineering simulations.

PyCuda



PyCUDA[2] is an open source python package developed by Andreas Klöckner. This package offers a rich variety of tools that both simplify the use of CUDA and make it far more synergistic with Python. Since our research primarily entails using a python interpreter in order to facilitate computation, this method saves us from needing to write interoperability code between CUDA and Python. Preventing this step gives us more time to focus on our ray-tracing algorithms. In addition, strictly python computation would have not allowed us to take advantage of the GPU, and would therefore hinder our results.

Specifically this method allows us to focus on the development of CUDA kernel functions only, allowing us to create a diverse visualization library for use with Yt.

Volume Rendering With CUDA

In the UCSC Astrophysics 3D Visualization lab, we are developing a yt module that performs volume rendering on an Nvidia graphics card. We use CUDA to create ray-casting algorithms that move the volume rendering of the numerous data structures yt can parse onto the graphics card allowing users to interact with the data in a 3D space in real time.

We've successfully implemented a software toolchain that allows a user to process data with yt and then perform volume rendering on CUDA cores. The results of the volume rendering are then passed to OpenGL allowing users to interact with the data in real-time. Using a Quadro K6000, which was donated by Nvidia, 4.5 Gbytes of data was displayed at 50 frames per second. Alternatively, the results of the CUDA volume renderer can be displayed remotely in a users browser via the ipython notebook. This allows users to take advantage of GPU clusters to create high resolution videos.

Our results show that providing yt with a graphics suite that takes advantage of Nvidia CUDA cores will be a powerful contribution to the scientific community. We hope to extend visualization support to numerous simulation codes. Once completed, our module will equip yt users with stunning visuals of star birth, galaxy formation, the cosmic web of dark matter, and many other astrophysical phenomena. We believe both the public and the scientific community will benefit greatly from these advances in visualization software. Because yt is used for scientific work providing high-quality visuals, our work will enable easier public outreach.

IPython Notebook

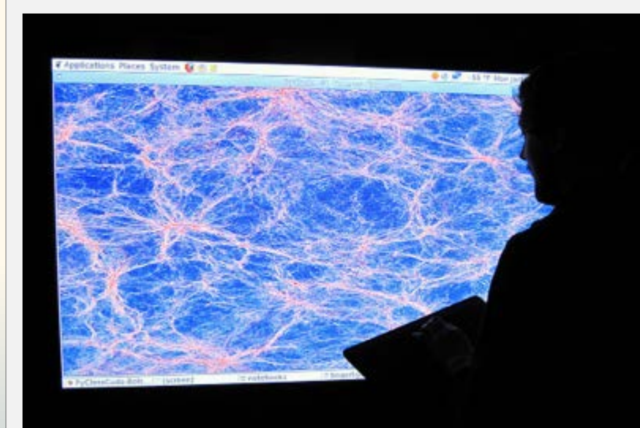
A real time streaming app that is embedded directly into IPython [3] Notebook is possible with the latest websocket protocol. This allows the user to make renderings on a remote server and stream images of those renderings to a window inside of IPython Notebook in real time. By simply calling Stream() on the CUDA context an HTML5 canvas is dropped into the output window and live images rendered on the server are displayed to the canvas. The mouse position is passed back to the server to update the rendering allowing the user to drag their mouse and rotate the data in real time. We have this kind of real time interaction with OpenGL, however, it requires you to be working on the machine locally. Streaming directly to the IPython Notebook allows the user to work from anywhere with an internet connection.

Streaming is done by forwarding an extra port when establishing a connection to the server via ssh; the IPython Notebook server is then started normally. From inside a notebook, the user creates a CUDA rendering context and then calls Stream() on that object. Stream() creates a new thread that starts a websocket server for communication with the browser while the parent thread returns a small piece of HTML5 and Javascript to set up the canvas inside of the IPython Notebook output window. The browser then sends a request to the server with the current mouse position, relative to the canvas window, and the server makes a rendering, encodes it into a jpeg image, serializes it for websocket communication and sends it to the browser. Finally, the browser deserializes the incoming jpeg and writes the image to the HTML5 canvas. The entire streaming system is all done with HTML and Javascript on the client side and uses Tornado web framework in Python on the server side.

PyOpenGL

Because our volume renderer relies on yt and PyCuda we naturally use a python implementation of OpenGL to create an interactive user experience. PyOpenGL[4] is capable of binding a pointer supplied by PyCuda to a texture representing the results of the ray casting. Then the user can use i/o devices such as a keyboard or mouse to control the view of the data.

The PyCuda to PyOpenGL interop allows the data to persist in device texture memory. The result of the ray casting through the data can be displayed at 50 fps on a 1080p screen. This gives the user the feeling that they are exploring the data in real time. This also allows the user to highlight interesting features of the data by changing color and brightness.



Stereoscopic Rendering

With the latest high performance Nvidia cards, such as the Quadro K6000, interactive stereoscopic volume rendering can be achieved in real-time. Though the frame rate suffers dropping to 20 fps from 50 fps, the user can still manipulate the data. Stereoscopic rendering gives viewers another piece of information to observe in his or her data. Combined with rotations this can reveal new information about three dimensional structures not previously seen in a two dimensional slice.

References:

[1] Turk, M. J., Smith, B. D., Oishi, J. S., Skory, S., Skillman, S. W., Abel, T., & Norman, M. L. "yt: A multi-code analysis toolkit for astrophysical simulation data." The Astrophysical Journal Supplement Series 192.1 (2011): 9.

[2] PyCuda Main Page, http://mathema.tician.de/software/pycuda/, retrieved January 27, 2014

[3] IPython Main Page, http://ipython.org/notebook.html, retrieved January 27, 2014

[4] PyOpenGL Main Page, http://pyopengl.sourceforge.net/, retrieved January 27, 2014

Thanks to:

Physics Professor Joel Primack, Astronomy & Astrophysics Professor Enrico Ramirez-Ruiz, University of California High-performance AstroComputing Center (UC-HiPACC), Matt Turk and the yt community, Nvidia for the donation of a Quadro K6000, and Nathan Goldbaum

yt-3.0 Just Released: Aug. 4, 2014

- Supports many codes, including all participating in AGORA

… This release of yt features an entirely rewritten infrastructure for data ingestion, indexing, and representation. While past versions of yt were focused on analysis and visualization of data structured as regular grids, this release features full support for particle (discrete point) data such as N-body and SPH data, irregular hexahedral mesh data, and data organized via octrees. This infrastructure will be extended in future versions for high-fidelity representation of unstructured mesh datasets.

Highlighted changes in yt 3.0:

- Units now permeate the code base, enabling self-consistent unit transformations of all arrays and quantities returned by yt.
- Particle data is now supported using a lightweight octree. SPH data can be smoothed onto an adaptively-defined mesh using standard SPH smoothing
- Support for octree AMR codes
- Preliminary Support for non-Cartesian data, such as cylindrical, spherical, and geographical
- Revamped analysis framework for halos and halo catalogs, including direct ingestion and analysis of halo catalogs of several different formats
- Support for multi-fluid datasets and datasets containing multiple particle types
- Flexible support for dynamically defining new particle types using filters on existing particle types or by combining different particle types.
- Vastly improved support for loading generic grid, AMR, hexahedral mesh, and particle without hand-coding a frontend for a particular data format.
- New frontends for ART, ARTIO, Boxlib, Chombo, FITS, GDF, Subfind, Rockstar, Pluto, RAMSES, SDF, Gadget, OWLS, PyNE, Tipsy, as well as rewritten frontends for Enzo, FLASH, Athena, and generic data.
- First release to support installation of yt on Windows
- Extended capabilities for construction of simulated observations, and new facilities for analyzing and visualizing FITS images and cube data
- Many performance improvements