INTRODUCTION TO ENZO

Britton Smith

OUTLINE

I. Available Modules

II. Problem Types

III. Enzo Resources

IV. Development

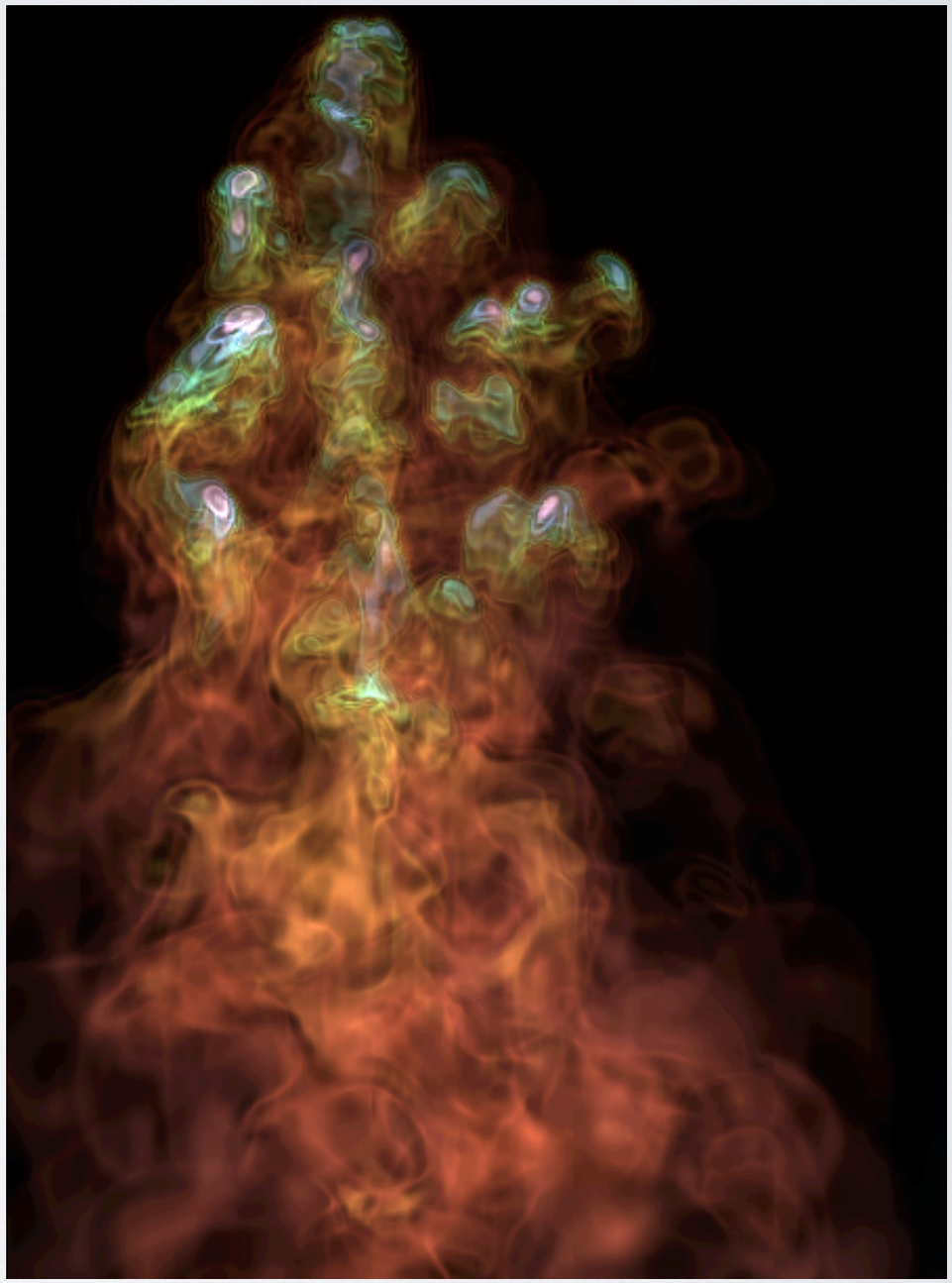


Image: Devin Silvia

ENZO METHOD PAPER

ENZO: AN ADAPTIVE MESH REFINEMENT CODE FOR ASTROPHYSICS

The Enzo Collaboration: Greg L. Bryan[1], Michael L. Norman[2,3], Brian W. O'Shea[4,5], Tom Abel[6, 20], John H. Wise[7], Matthew J. Turk[1], Daniel R. Reynolds[8], David C. Collins[9], Peng Wang[6], Samuel W. Skillman[10,11], Britton Smith[4], Robert P. Harkness[12], James Bordner[2], Ji-hoon Kim[13], Michael Kuhlen[14,15], Hao Xu[2], Nathan Goldbaum[15], Cameron Hummels[16], Alexei G. Kritsuk[2], Elizabeth Tasker[17], Stephen Skory[10], Christine M. Simpson[1], Oliver Hahn[18], Jeffrey S. Oishi[19], Geoffrey C So[2], Fen Zhao[20], Renyue Cen[21], and Yuan Li[1]

Draft version July 22, 2013

ABSTRACT

This paper describes the open-source code Enzo, which uses block-structured adaptive mesh refinement to provide high spatial and temporal resolution for modeling astrophysical fluid flows. The code is Cartesian, can be run in 1, 2, and 3 dimensions, and supports a wide variety of physics including hydrodynamics, ideal and non-ideal magnetohydrodynamics, N-body dynamics (and, more broadly, self-gravity of fluids and particles), primordial gas chemistry, optically-thin radiative cooling of primordial and metal-enriched plasmas (as well as some optically-thick cooling models), radiation transport, cosmological expansion, and models for star formation and feedback in a cosmological context. In addition to explaining the algorithms implemented, we present solutions for a wide range of test problems, demonstrate the code's parallel performance, and discuss the Enzo collaboration's code development methodology.

Keywords: methods: numerical — hydrodynamics

http://arxiv.org/abs/1307.2265

WHAT IS ENZO?

Enzo is a cosmological, adaptive-mesh refinement, hydrodynamics + N-body simulation code.

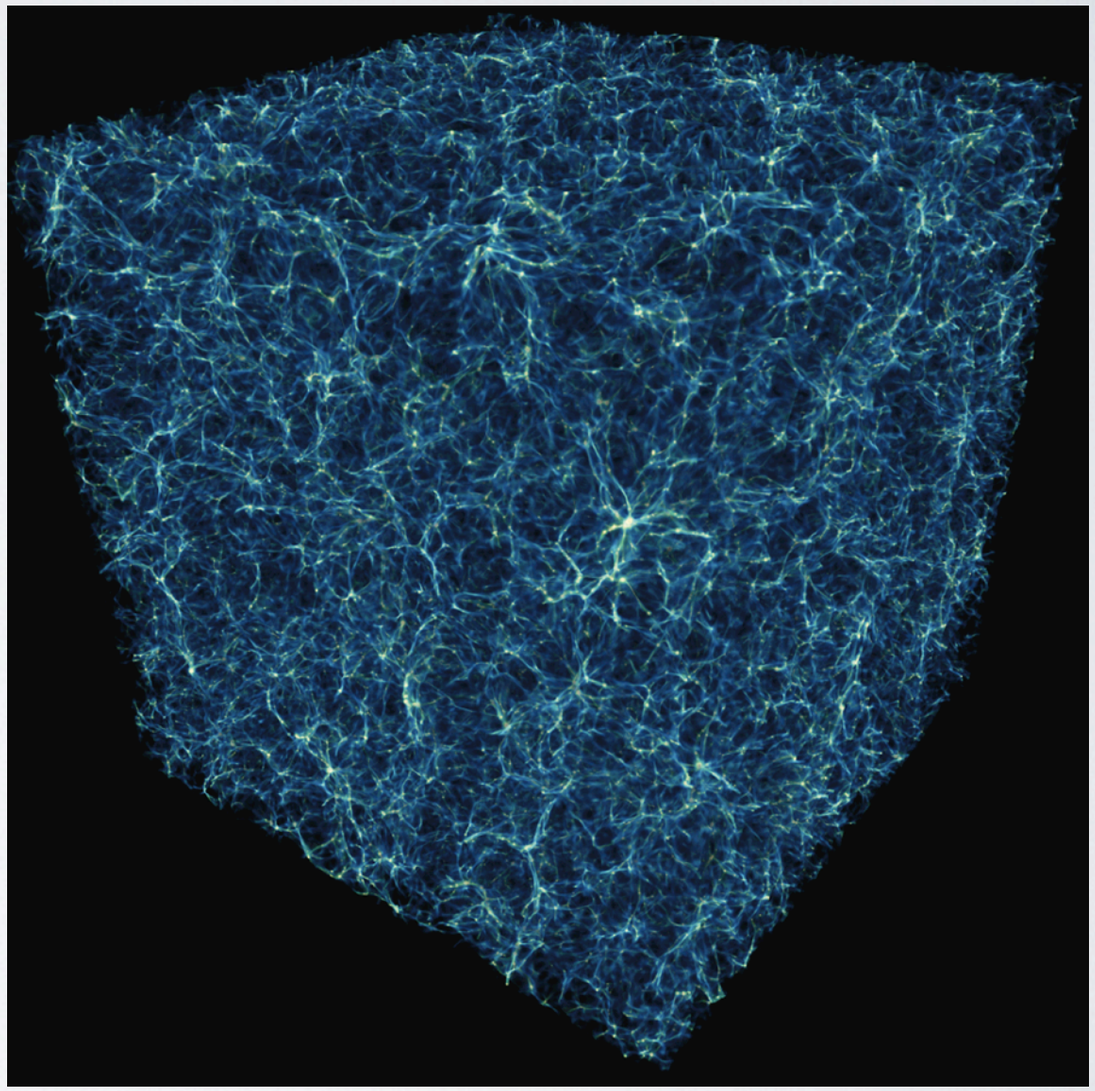
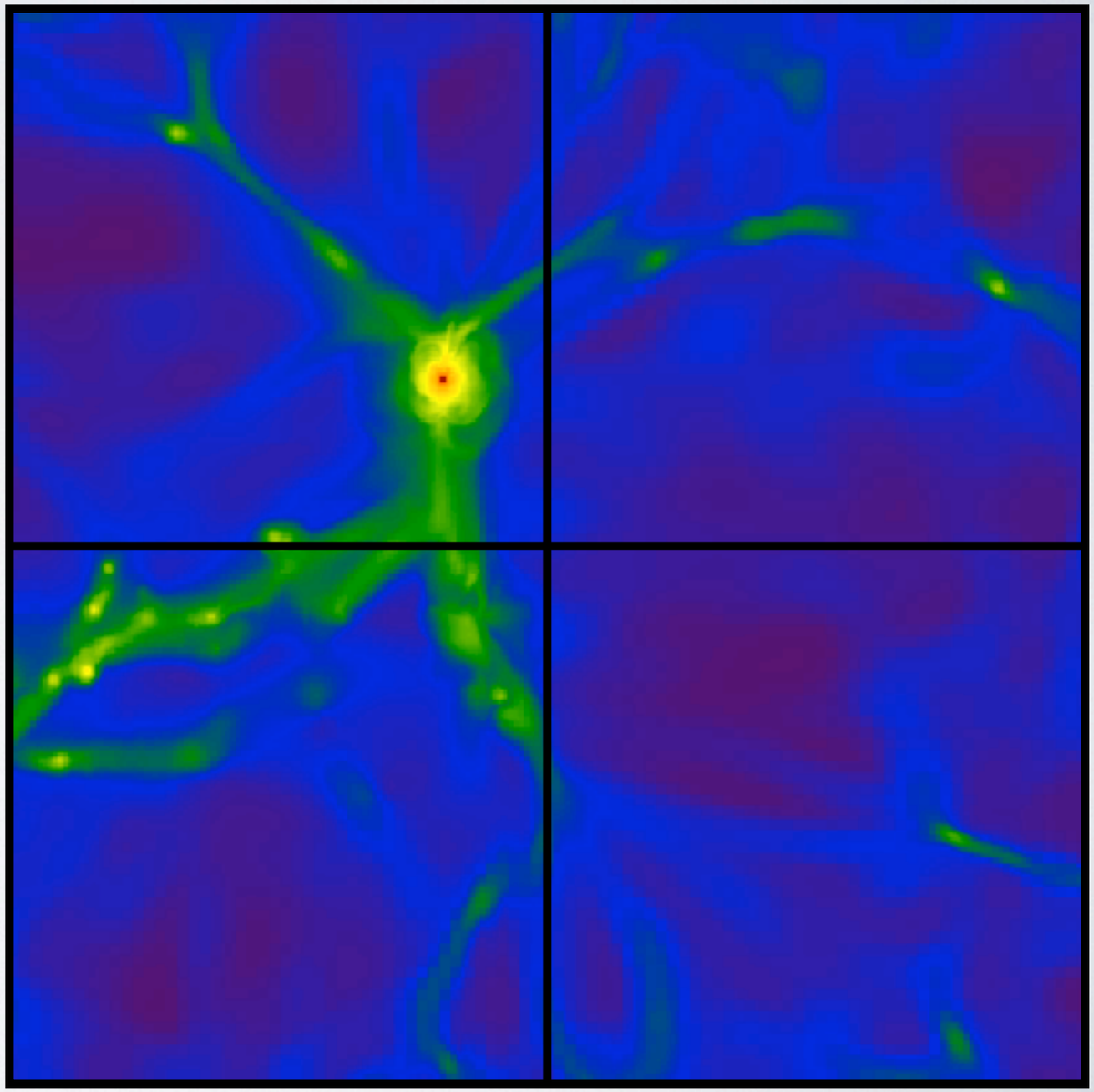


Image: Eric Hallman, Brian O'Shea

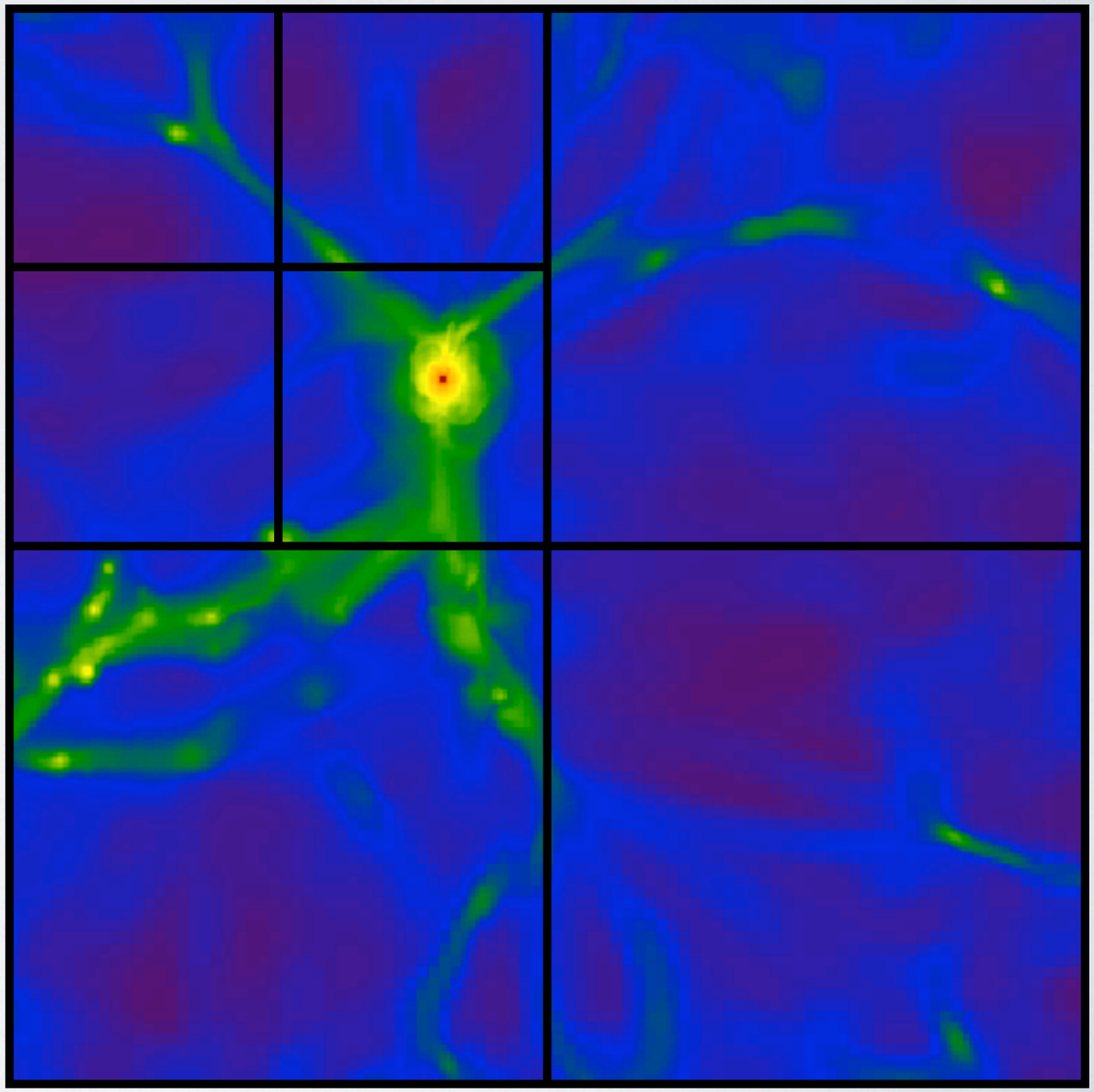
AMR:
ADAPTIVE-MESH REFINEMENT

- create and destroy grid patches dynamically (block-structured)
- grids at multiple resolutions
- multiple refinement criteria:
  - density (gas or dark matter)
  - gradients, shocks
  - cooling time
  - Jeans length
  - refine regions around particles
- easy to create new criteria



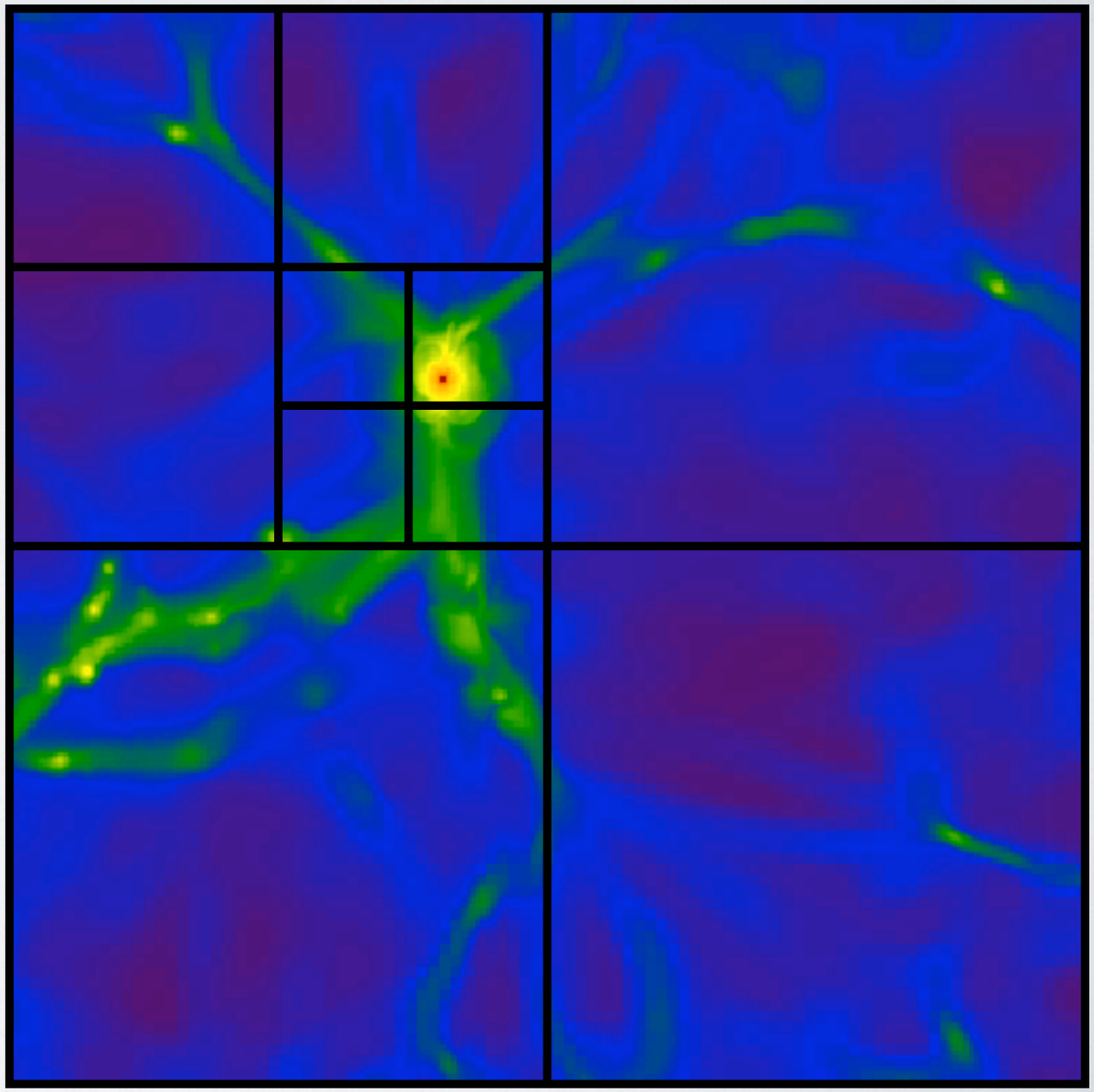
AMR:
ADAPTIVE-MESH REFINEMENT

- create and destroy grid patches dynamically (block-structured)
- grids at multiple resolutions
- multiple refinement criteria:
  - density (gas or dark matter)
  - gradients, shocks
  - cooling time
  - Jeans length
  - refine regions around particles
- easy to create new criteria



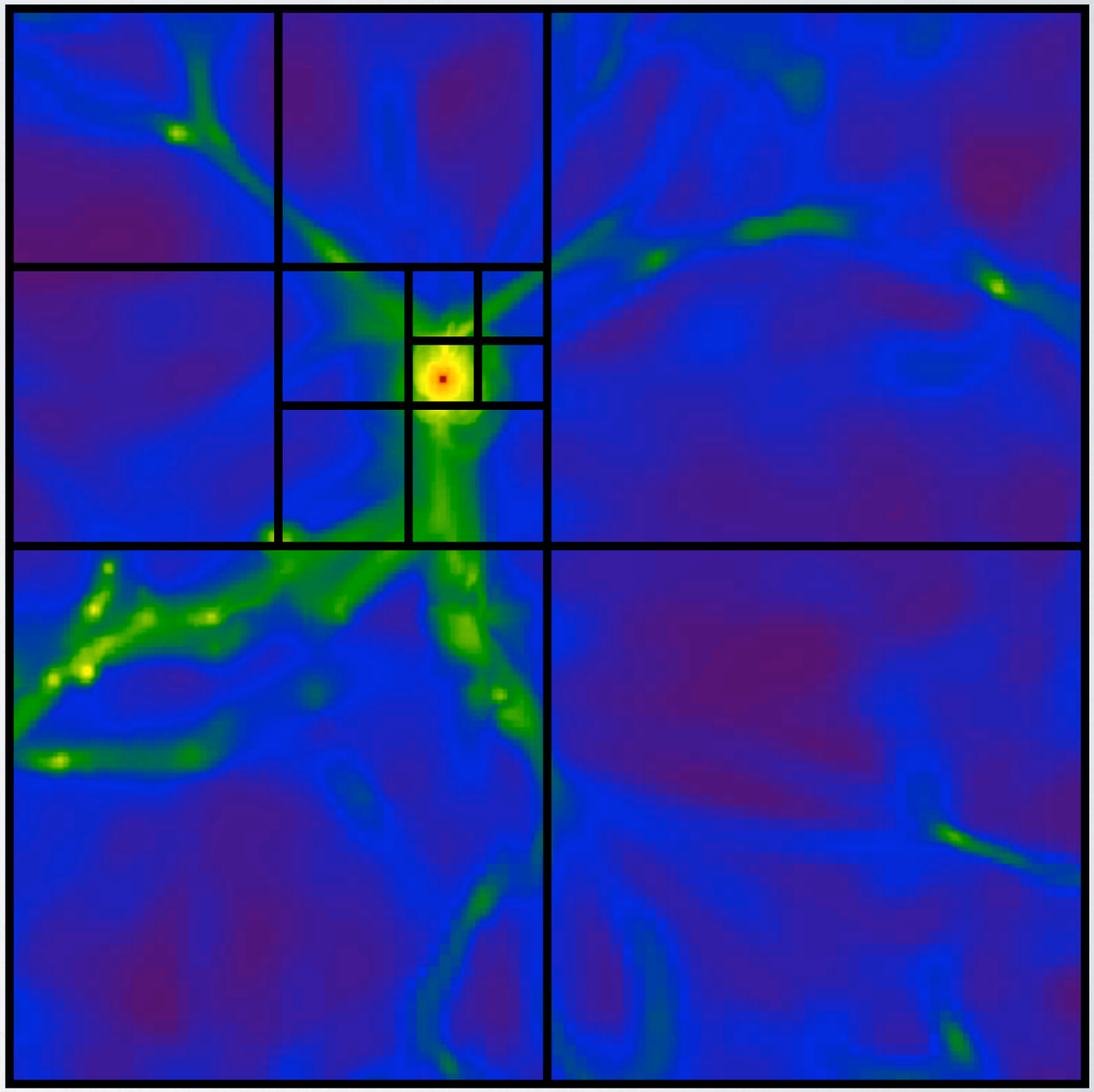
AMR:
ADAPTIVE-MESH REFINEMENT

- create and destroy grid patches dynamically (block-structured)
- grids at multiple resolutions
- multiple refinement criteria:
  - density (gas or dark matter)
  - gradients, shocks
  - cooling time
  - Jeans length
  - refine regions around particles
- easy to create new criteria



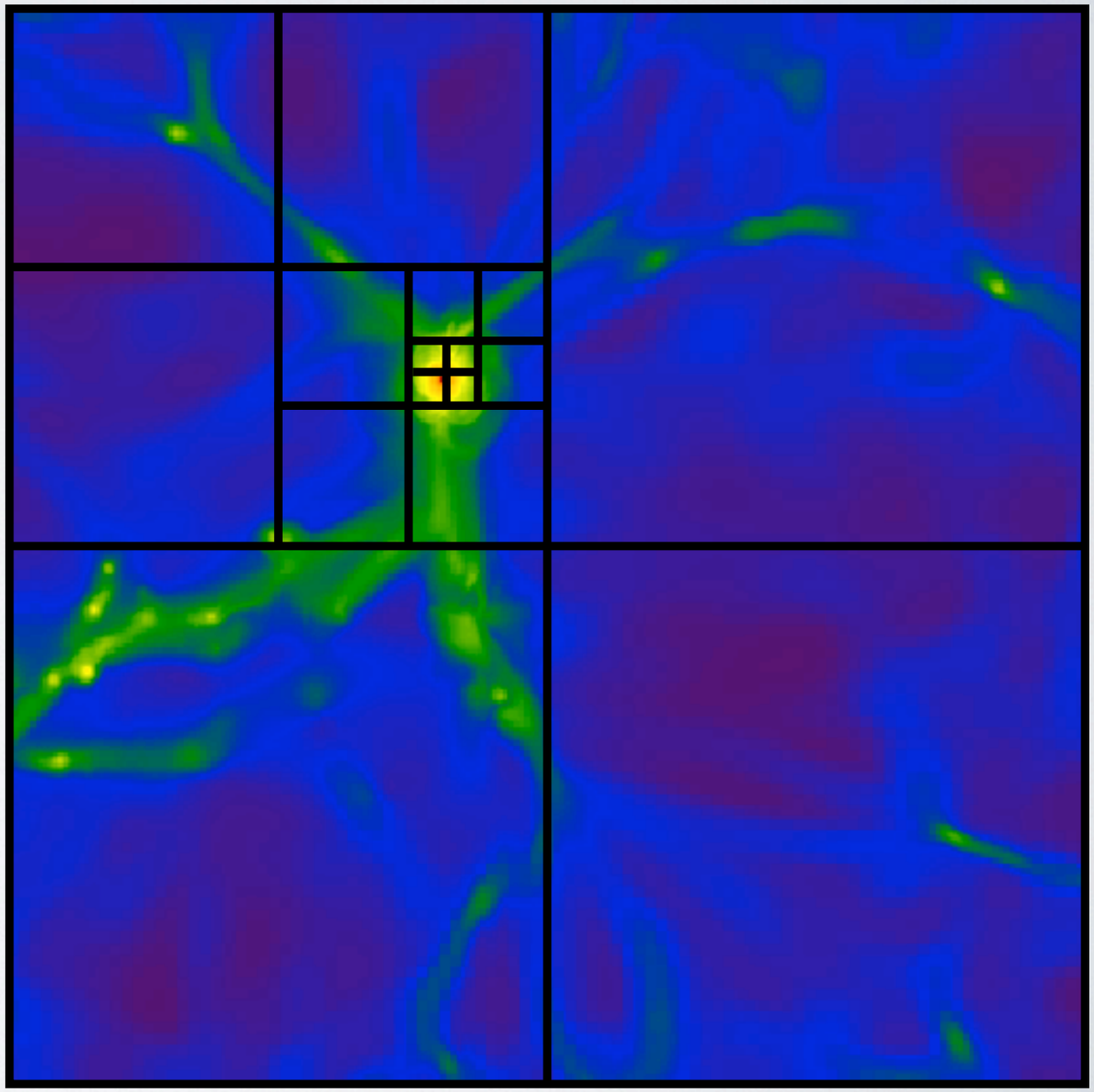
AMR:
ADAPTIVE-MESH REFINEMENT

- create and destroy grid patches dynamically (block-structured)
- grids at multiple resolutions
- multiple refinement criteria:
  - density (gas or dark matter)
  - gradients, shocks
  - cooling time
  - Jeans length
  - refine regions around particles
- easy to create new criteria



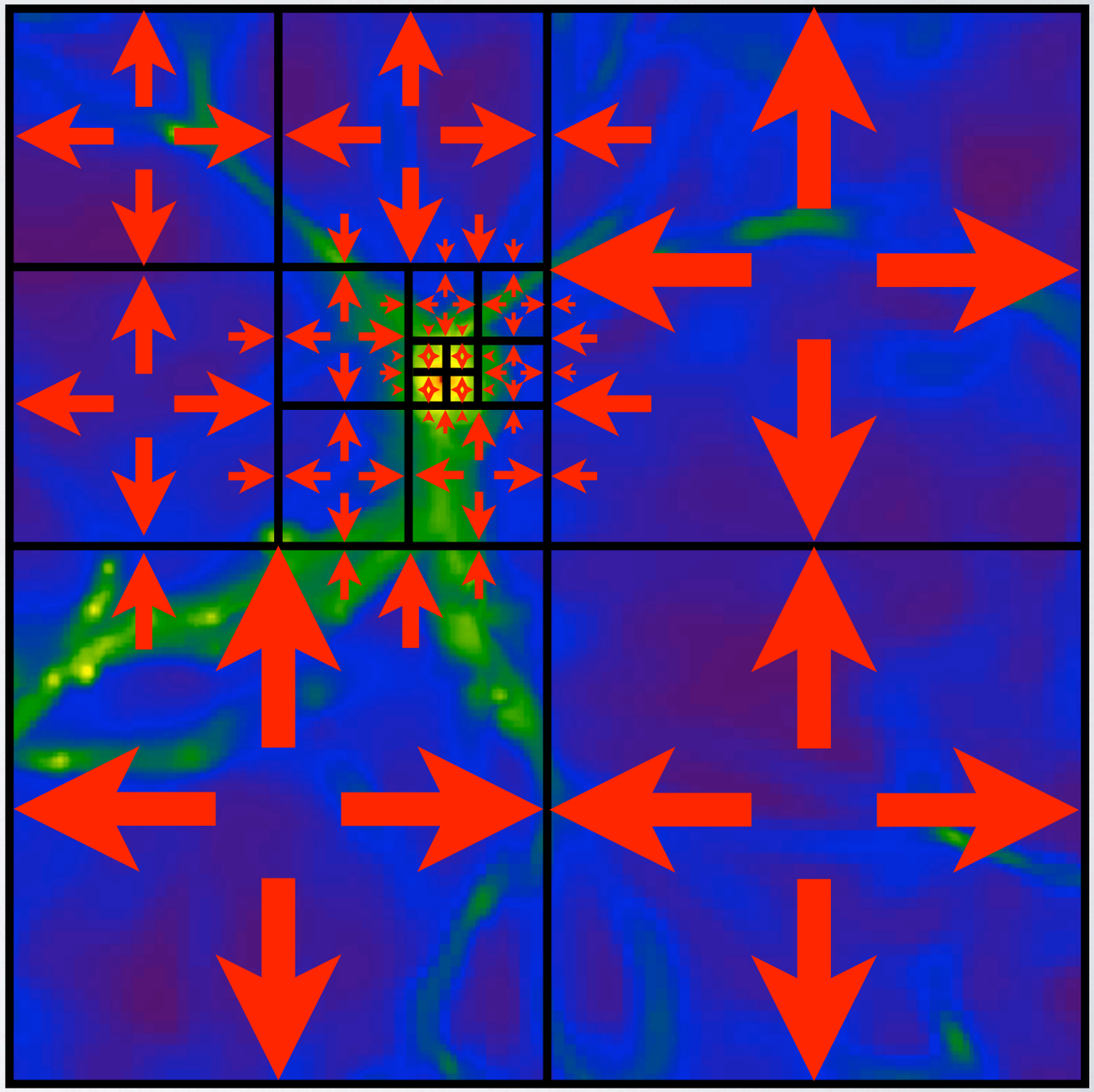
AMR:
ADAPTIVE-MESH REFINEMENT

- create and destroy grid patches dynamically (block-structured)
- grids at multiple resolutions
- multiple refinement criteria:
  - density (gas or dark matter)
  - gradients, shocks
  - cooling time
  - Jeans length
  - refine regions around particles
- easy to create new criteria



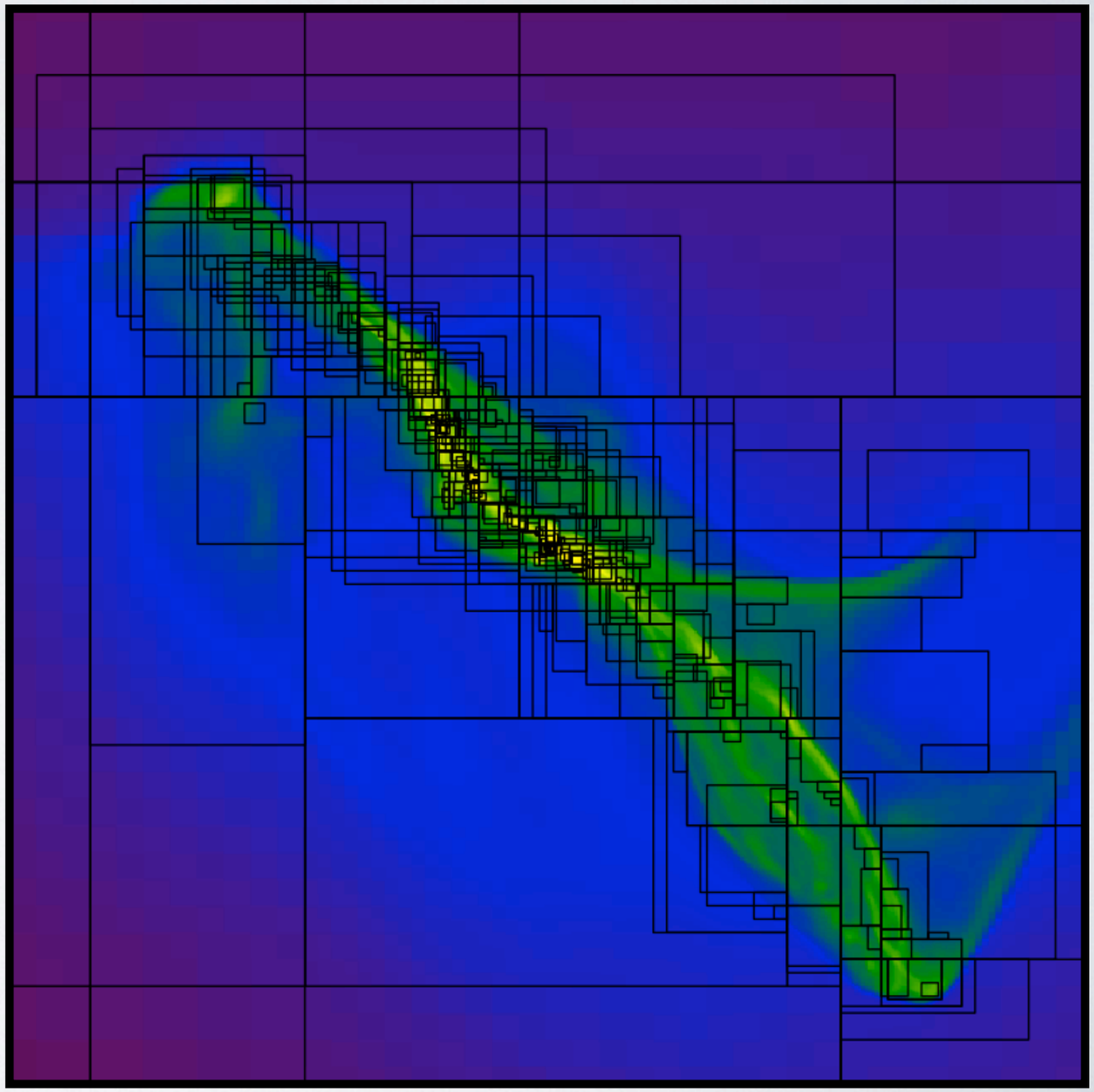
AMR:
ADAPTIVE-MESH REFINEMENT

- create and destroy grid patches dynamically (block-structured)
- grids at multiple resolutions
- multiple refinement criteria:
  - density (gas or dark matter)
  - gradients, shocks
  - cooling time
  - Jeans length
  - refine regions around particles
- easy to create new criteria



AMR:
ADAPTIVE-MESH REFINEMENT

- create and destroy grid patches dynamically (block-structured)
- grids at multiple resolutions
- multiple refinement criteria:
  - density (gas or dark matter)
  - gradients, shocks
  - cooling time
  - Jeans length
  - refine regions around particles
- easy to create new criteria



ADAPTIVE-MESH REFINEMENT
EXTREME DYNAMIC RANGE

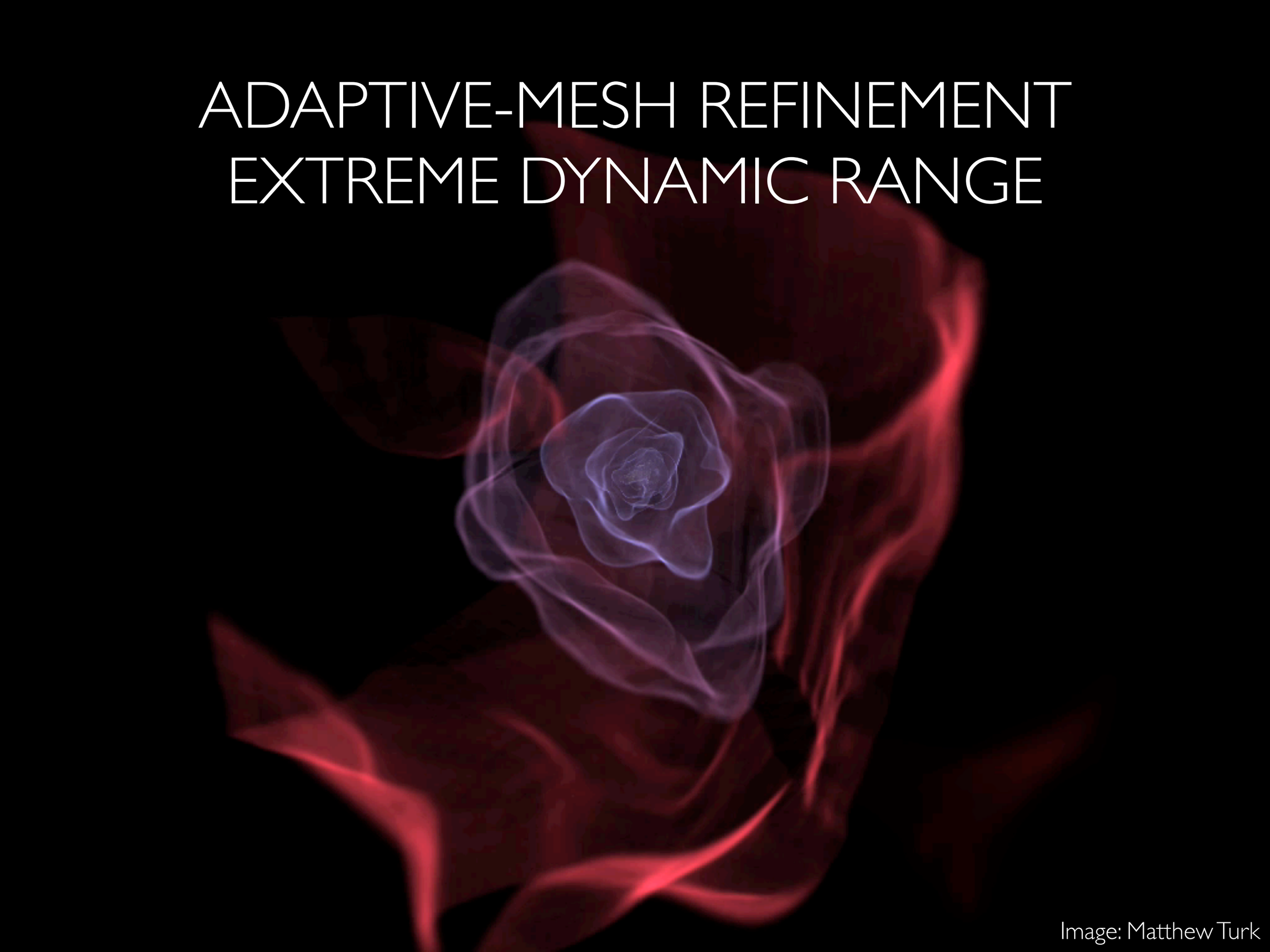


Image: Matthew Turk

GRAVITY

- dark matter treated as collision-less particles
- adaptive particle-mesh method
  - solve Poisson eqn.: $\nabla^2\phi = 4\pi G\rho$
  - particles interpolated onto grid to create density field, then gas densities added
  - multigrid relaxation for refined grids
- advantage: very fast!
- disadvantage: force res. is $2\Delta x$ (not great)

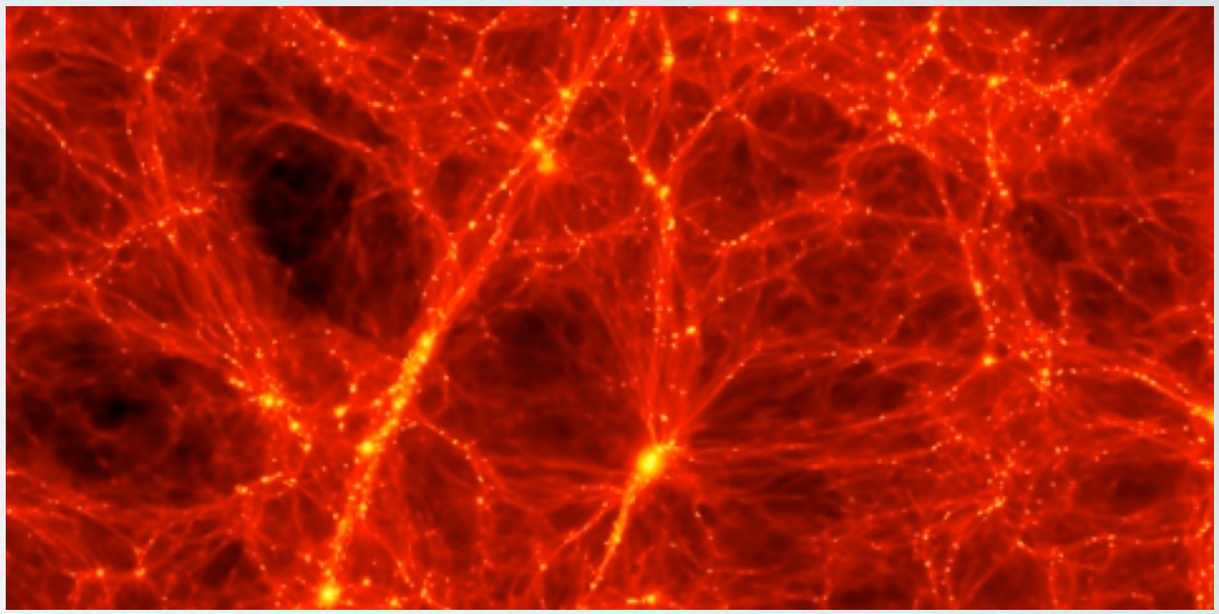


Image: Michael Norman et. al.

HYDRODYNAMICS

Multiple Hydro Methods

- Piecewise Parabolic Method
  - fits state variables to 3rd order parabolic
  - nonlinear Riemann solver for excellent shock capturing
  - can be unstable with cosmology or cooling
- Zeus
  - less accurate, more diffusive
  - extremely robust (excellent for cosmology)

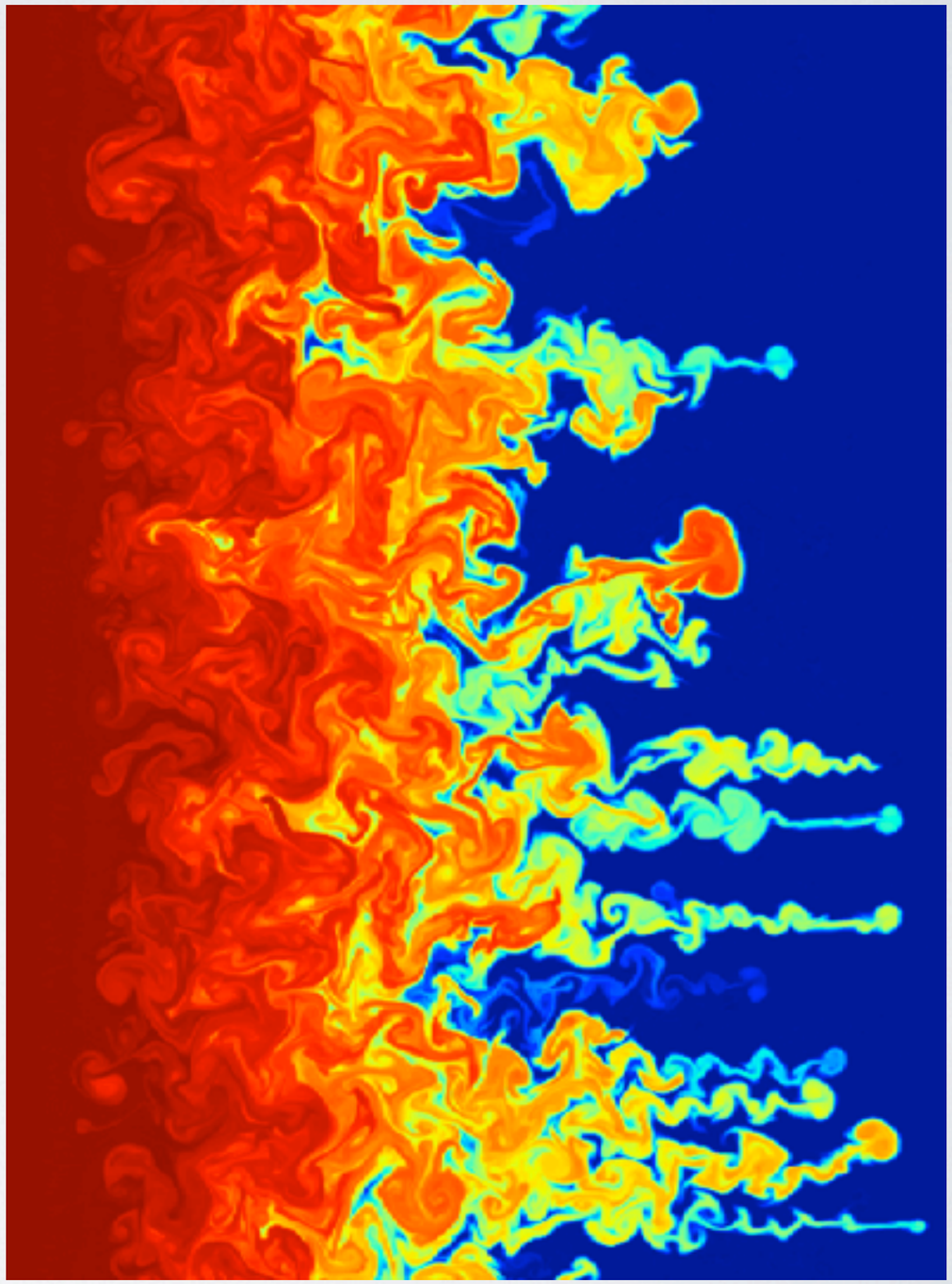


Image: Nick Earl

HYDRODYNAMICS

Multiple Hydro Methods

- MUSCL
  - 2nd order accurate Godunov solver
  - 2nd order Runge-Kutta time integration
  - multiple Riemann solvers and interpolation methods available

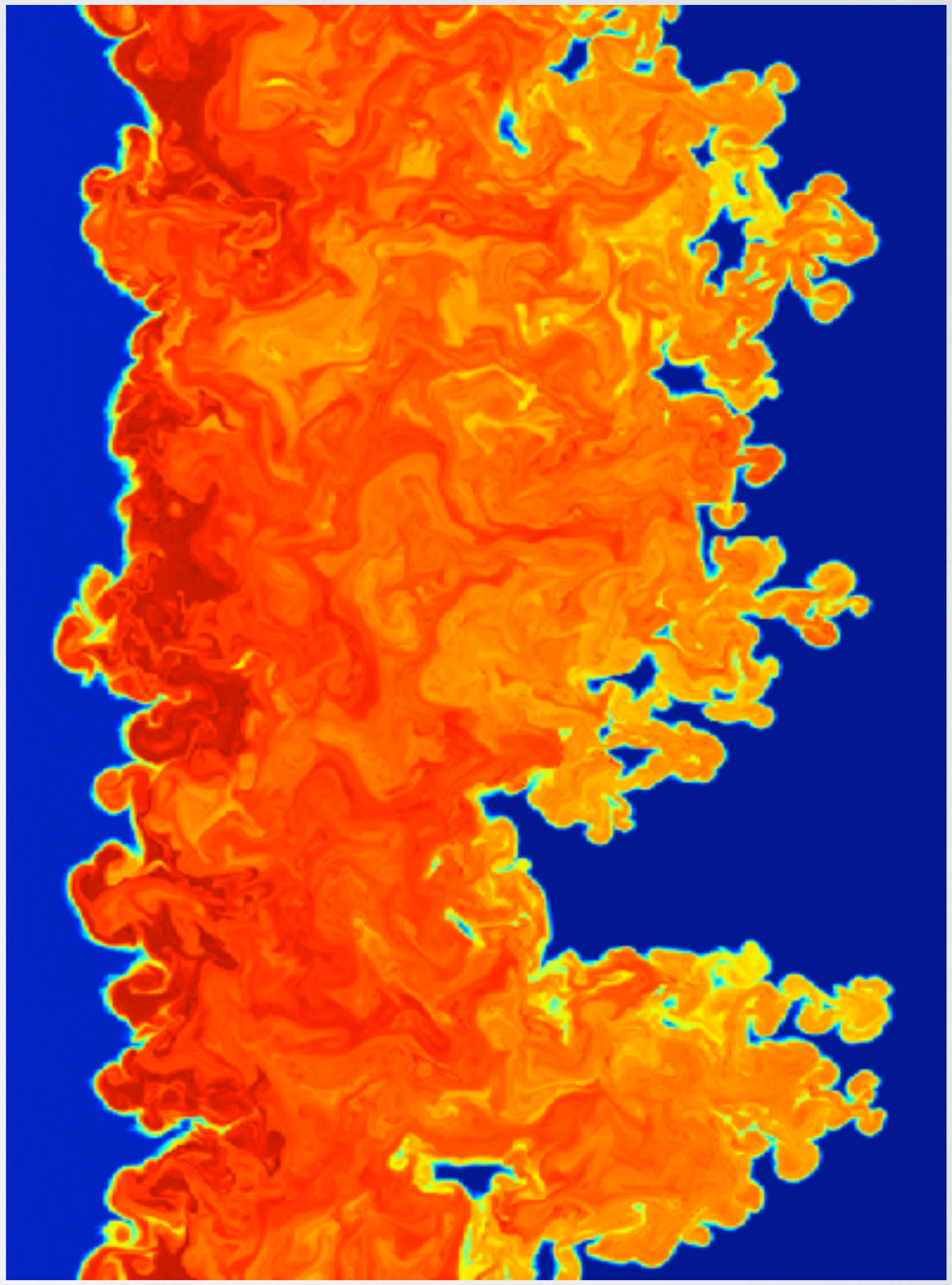
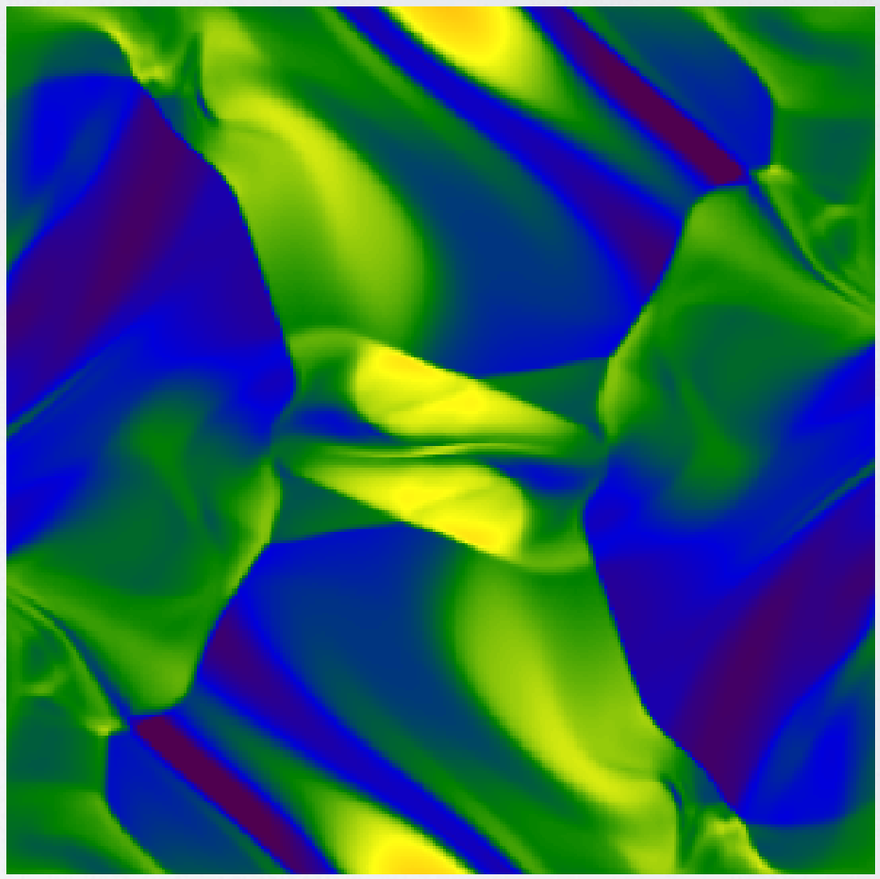


Image: Nick Earl

MHD

Multiple Hydro Methods

- Dedner
  - uses MUSCL framework above
  - hyperbolic divergence cleaning method to ensure $\nabla \cdot B = 0$
  - uses cell-centered B field
- Constrained Transport
  - magnetic field updated as the curl of the electric field
  - preserves $\nabla \cdot B = 0$ (since $\nabla \cdot (\nabla \times \mathbf{F}) = 0$)
  - needs face and edge-centered fields: more complicated



Enzo Method Paper

RADIATIVE TRANSFER

Two Rad. Transfer Methods

- Adaptive Ray Tracing
  - radiation from discreet sources (star and black hole particles)
  - adaptive ray splitting and merging
  - fully coupled to chemistry network
- Flux Limited Diffusion
  - treats radiation like a fluid
  - couple to atomic chemistry
  - highly scalable
  - unigrid and AMR versions available

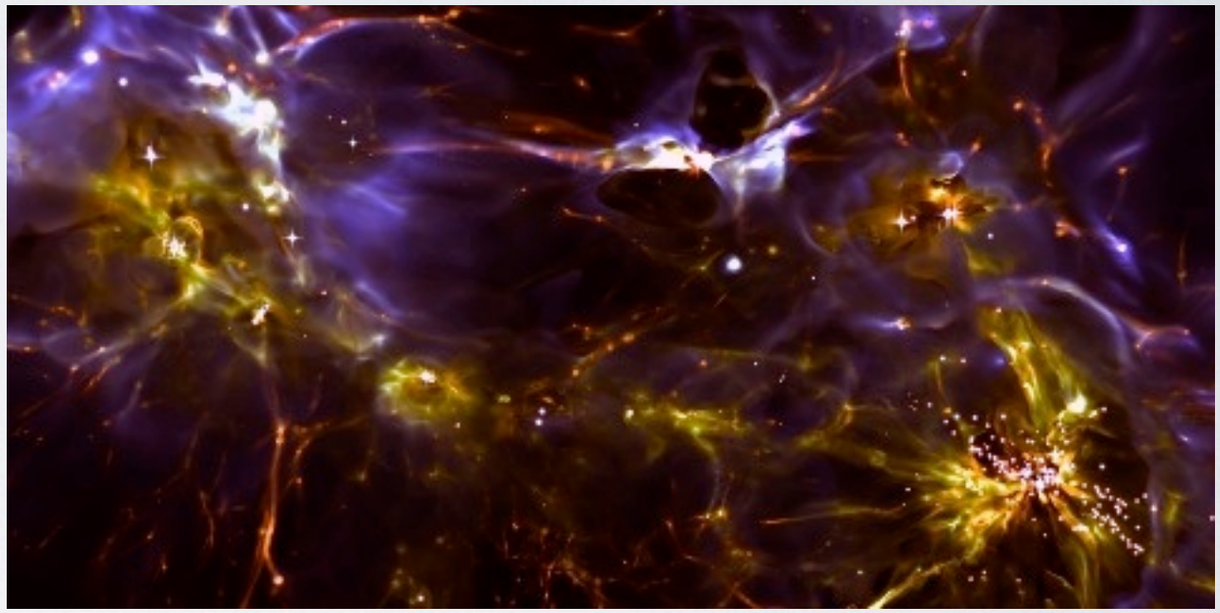


Image: John Wise

RADIATION BACKGROUNDS

- spatially smooth, time-dependent radiation fields
- UV metagalactic, ionizing backgrounds for photo-heating and ionization during Reionization (multiple models)
- Lyman-Werner soft UV fields represent radiation from first stellar sources and photo-dissociate H_2

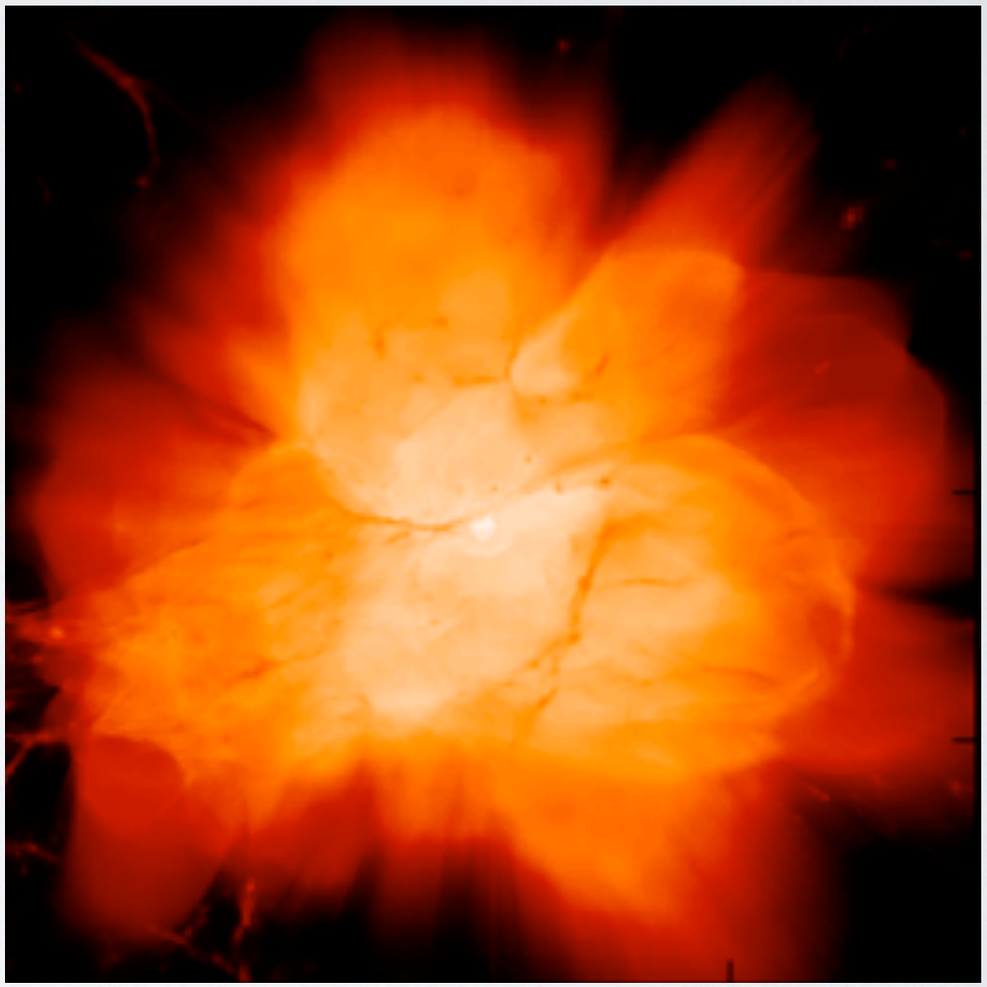
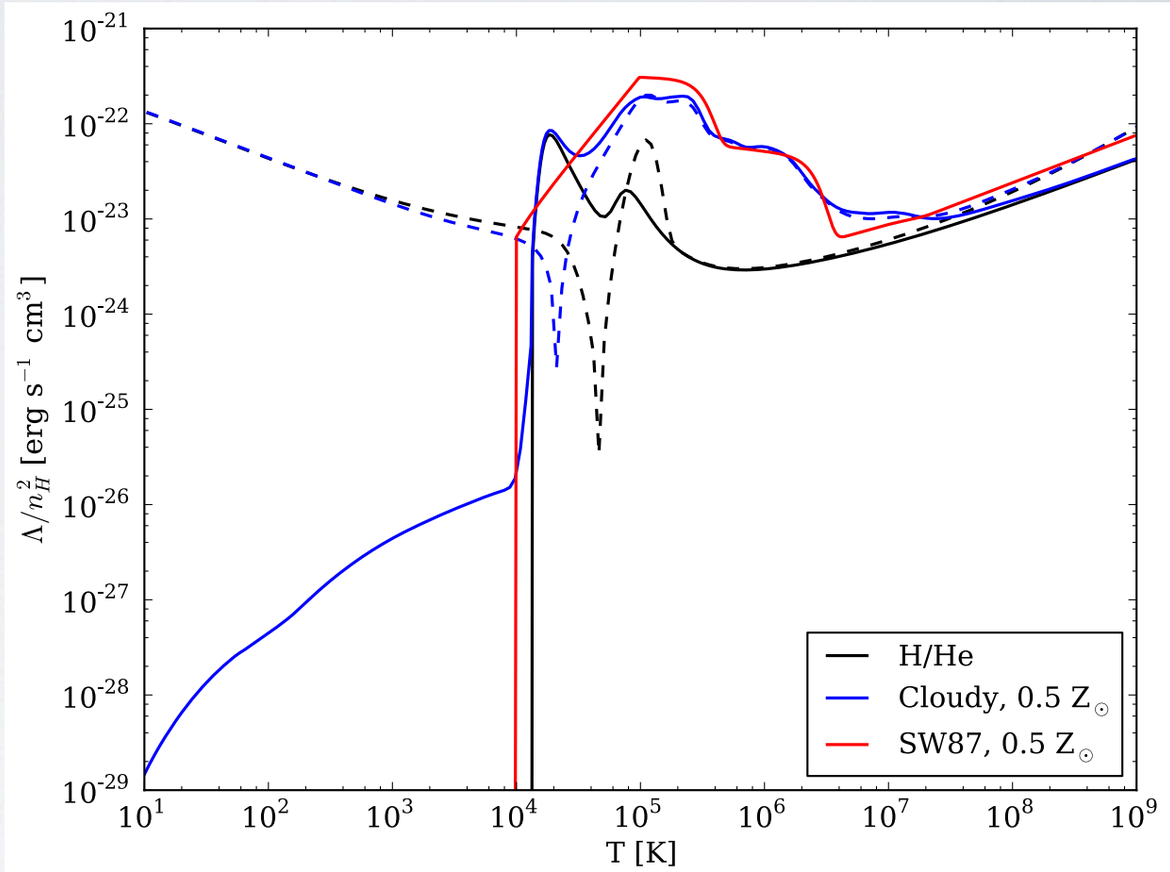


Image: Britton Smith

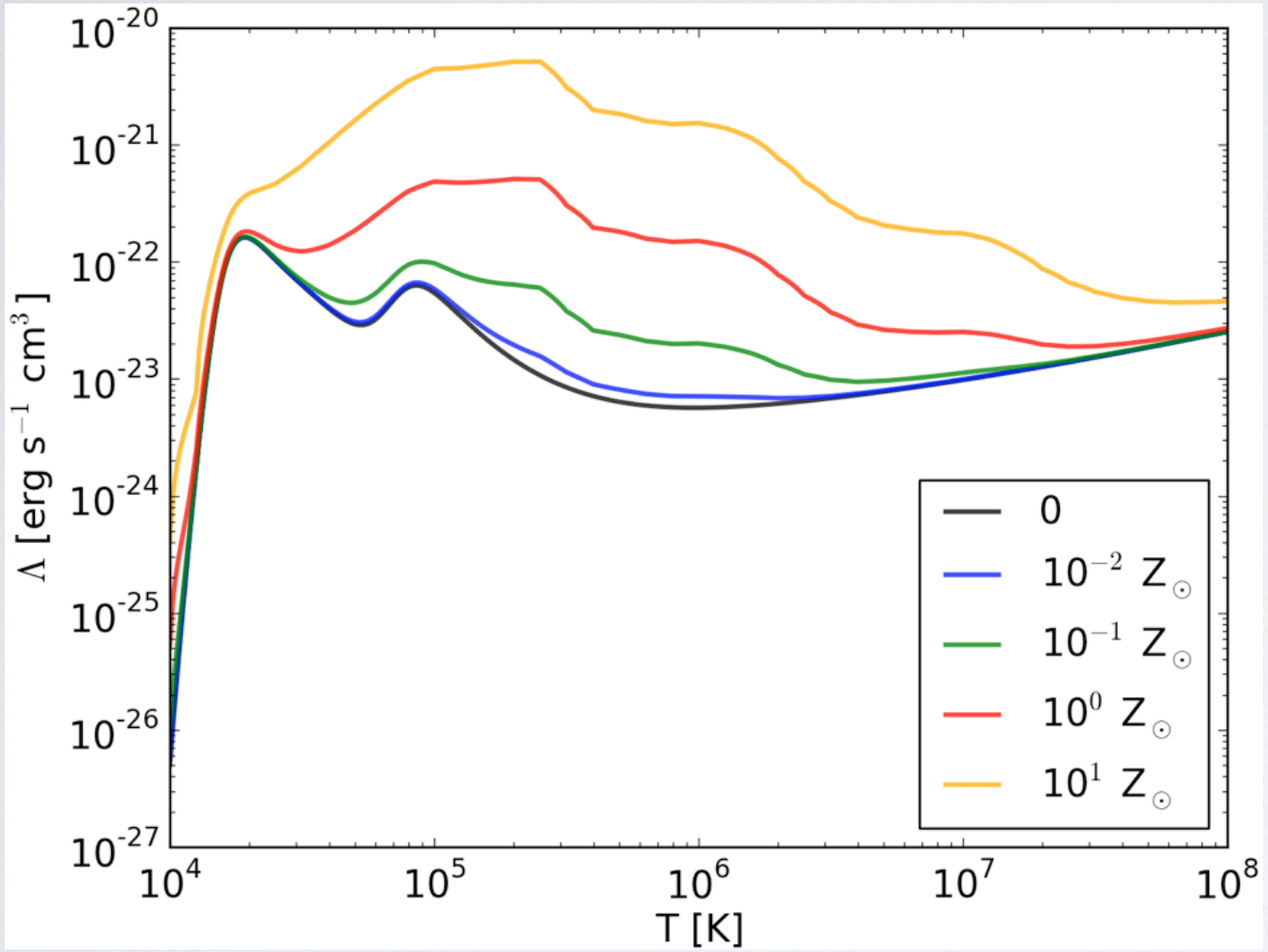
CHEMISTRY AND COOLING

- Non-equilibrium primordial chemistry
  - H, H^+, H^-, He, He^+, He^{++}, H_2, H_2^+, D, D^+, HD, e^-
  - H_2 chemistry: 2-body, 3-body channels, dust grains, chemical heating/cooling
- Metal cooling
  - simple tabulated rates ($T > 10^4$ K)
  - atomic fine-structure lines
  - Cloudy tables: density, metallicity, temperature, electron fraction, background redshift

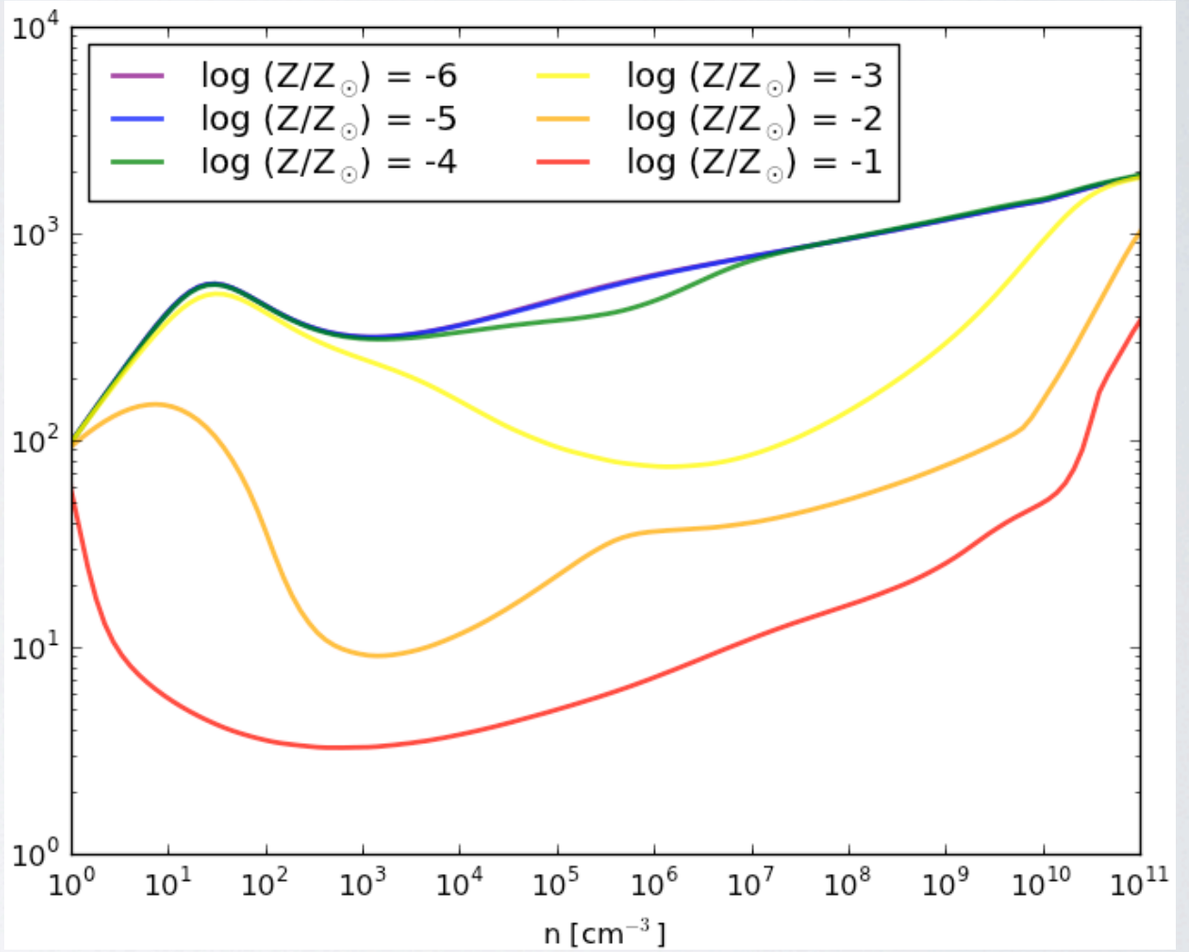


Enzo Method Paper

CHEMISTRY AND COOLING



High temperature cooling rates at various metallicities.

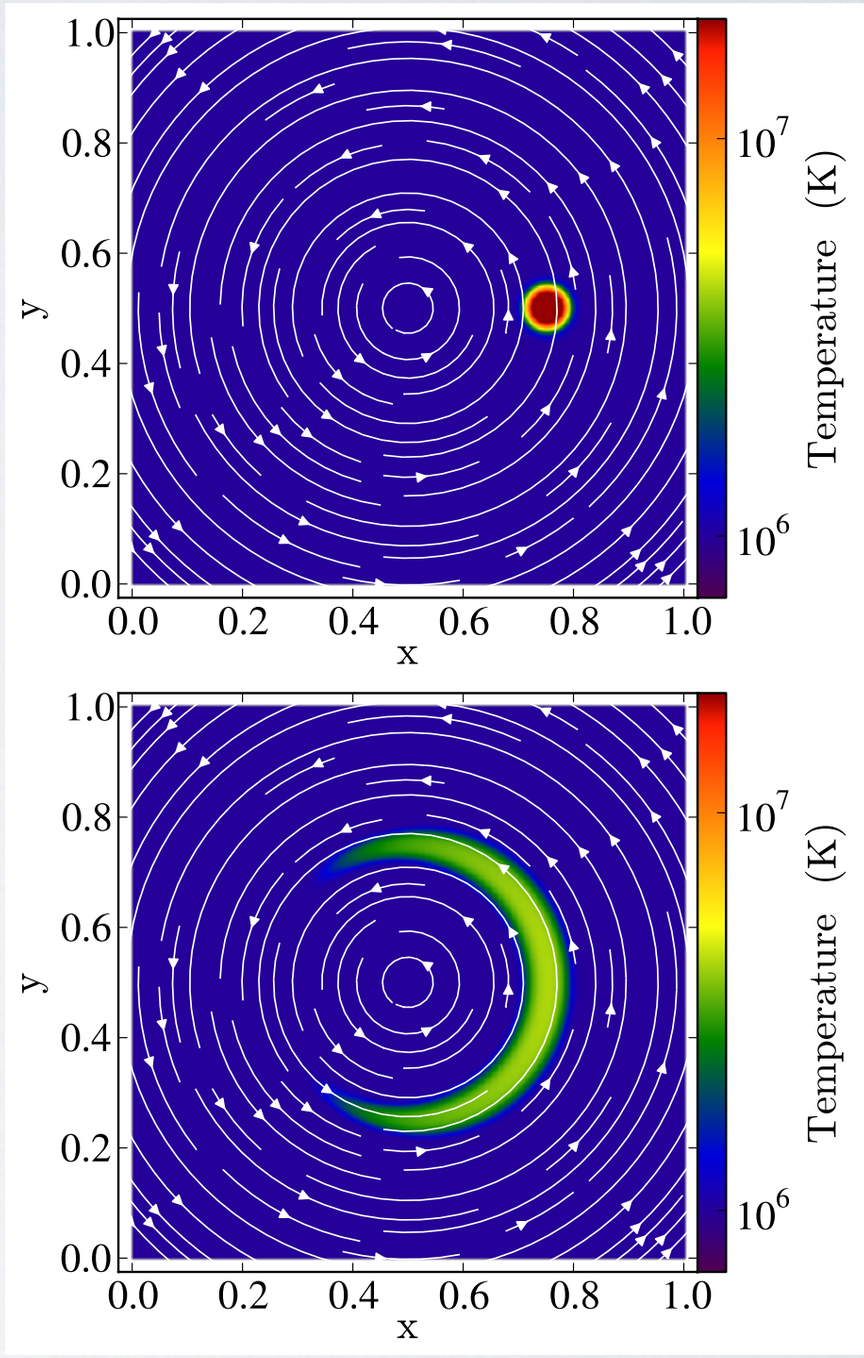


One-zone model of gas collapse at various metallicities.

THERMAL CONDUCTION

Spitzer Conduction

- heat transfer through electron Coulomb interactions
- $\kappa \sim T^{5/2}$, with saturation for sharp temperature gradients
- significant for $T > 10^7$ K (galaxy clusters)
- isotropic (hydro only) and anisotropic (MHD) available
- explicit solver: $dt \sim dx^2\, n / \kappa$: short timesteps!



Enzo Method Paper

ACTIVE PARTICLES

- act on the grid by adding or removing gas, energy, and momentum
- non-radiating star particles
  - form in dense, collapsing, cooling gas
  - inject thermal energy, metals into nearby grid cells
- radiating star and black-hole particles form the same way and emit radiation
- sink particles accrete nearby gas like collapsing protostars

Image: John Wise

TRACER PARTICLES

- can be placed anywhere in a simulation
- used to trace hydrodynamic flow
- output field values in which particles exist
- output separately from main dataset: can be output with higher frequency

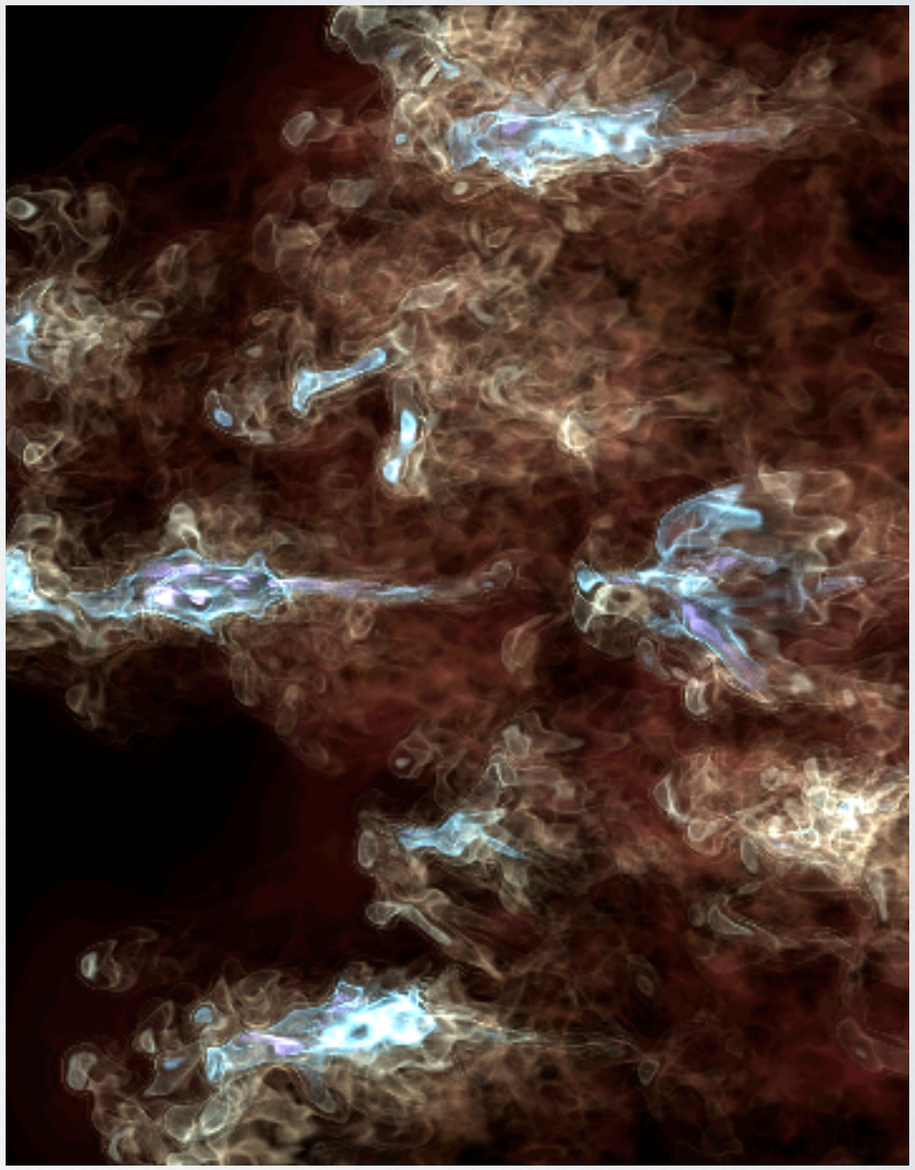
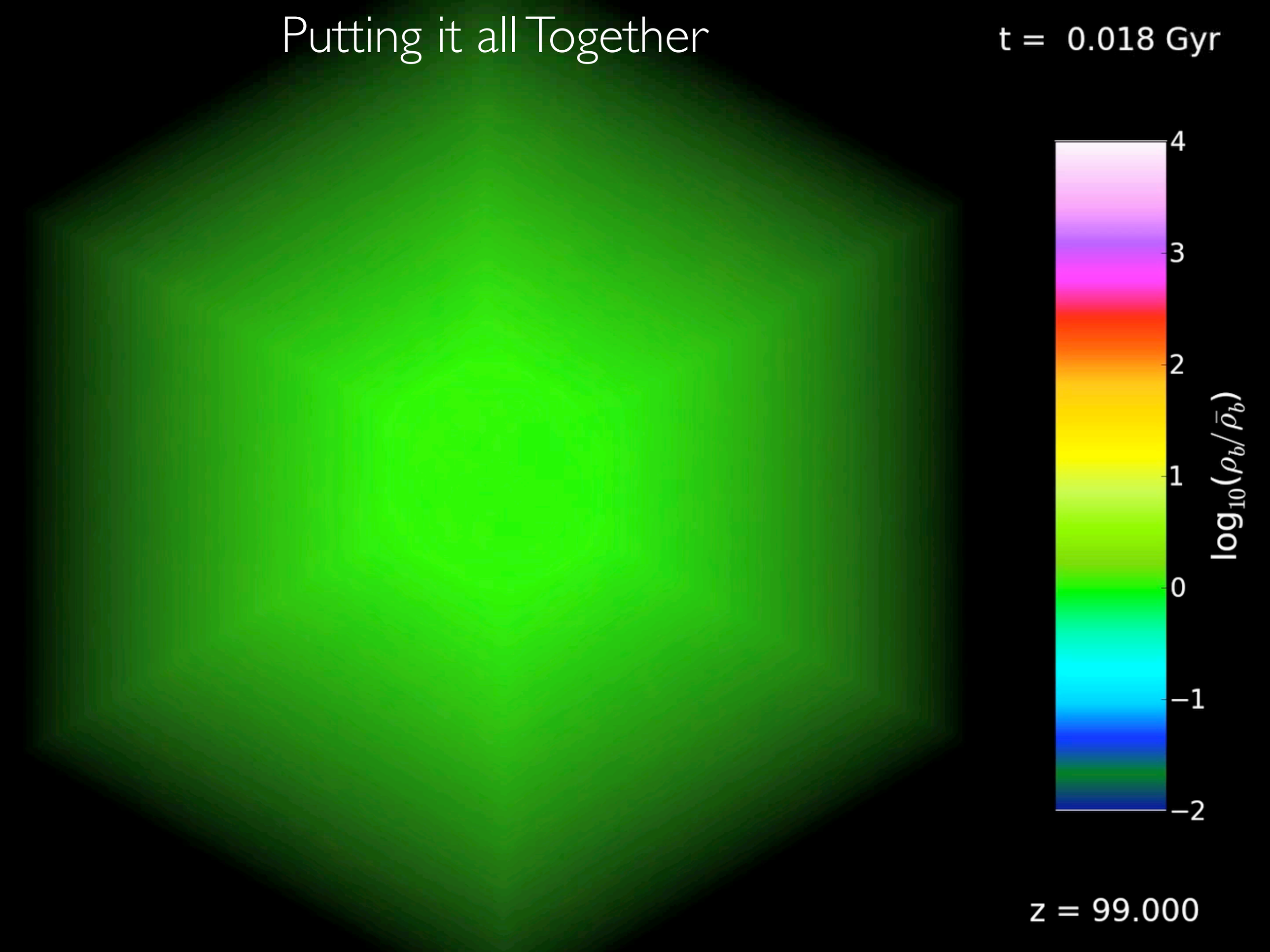


Image: Devin Silvia

Putting it all Together



PROBLEM TYPES

- Need external initial conditions files
  - Cosmology
  - Turbulence
- Enzo initializes everything
  - spheres: rotating, collapsing, colliding
  - galactic disks
  - shock tubes
  - cloud crushing
  - gravity, hydro tests
  - many, many more

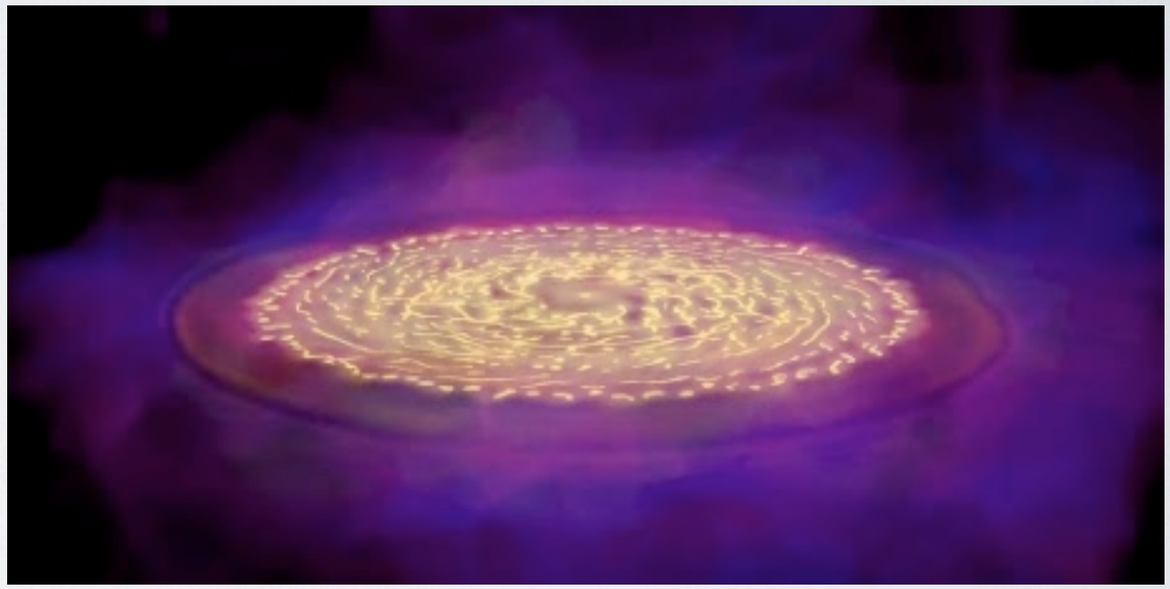


Image: Elizabeth Tasker

RESOURCES

- Enzo Webpage: enzo-project.org
- Documentation
- Email List
- IRC Channel
- yt Webpage: yt-project.org
- Mercurial tutorial: hginit.com

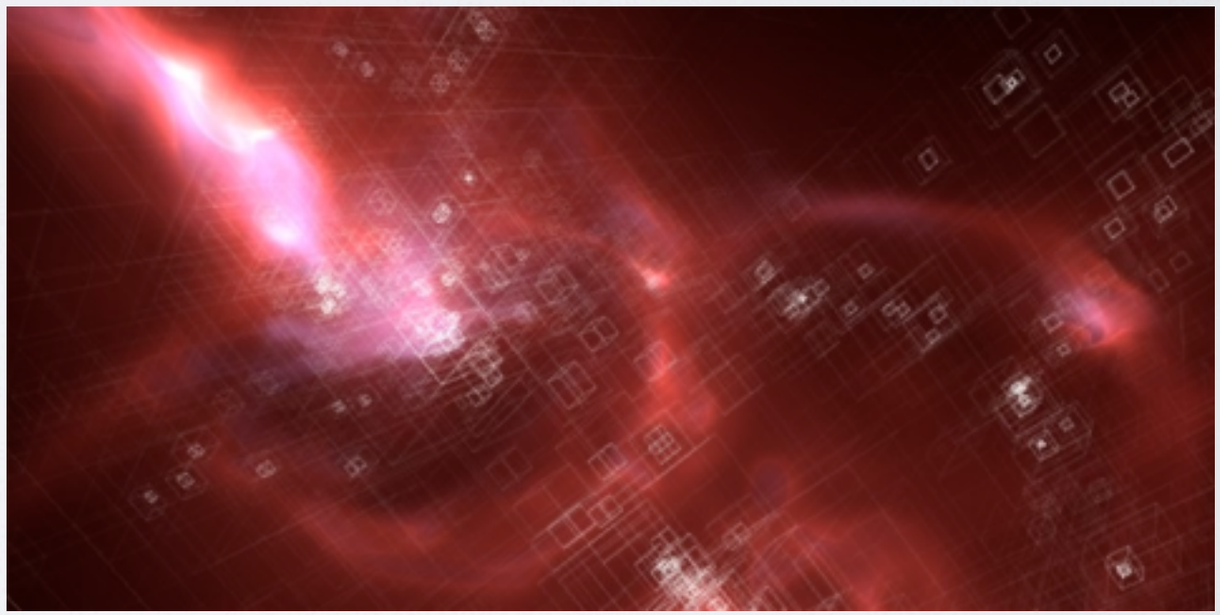


Image: Abel, Wise, Kahler

enzo-project.org

Enzo | Quick Links | Home | Get Enzo | Help! | Development | Community | Enzo Docs

Documentation

The Enzo Project

Help!

July 10 2013: Enzo 2.3 has been released. View the Release Notes!

July 10 2013: Enzo method paper entitled *Enzo: An Adaptive Mesh Refinement Code for Astrophysics* has been released. Get it here.

What is Enzo?

Enzo is a community-developed adaptive mesh refinement simulation code, designed for rich, multi-physics hydrodynamic astrophysical calculations.

Enzo is freely available, developed in the open, with a strong support structure for assistance. Simulations conducted with Enzo have been featured in numerous refereed journal articles, and it is capable of running on computers from laptop to Top500.

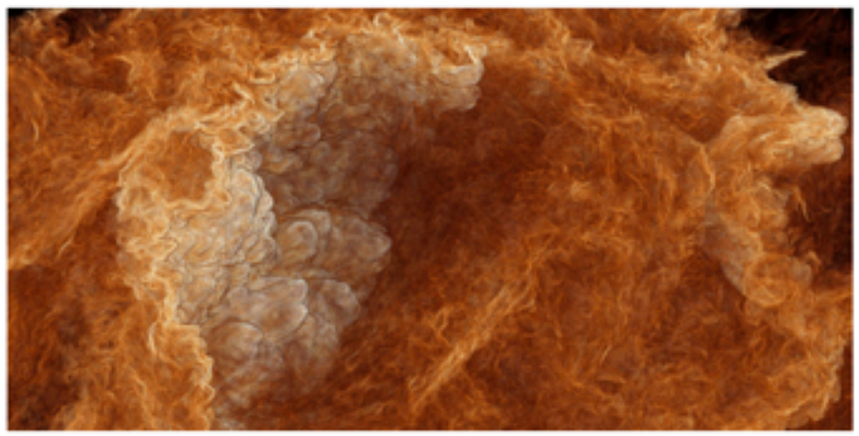


Image credit: Alexei Kritsuk, Paolo Padoan & Mike Norman

Getting Enzo

Enzo can be obtained in several places, corresponding to the degree of stability and development accessibility.

Let's go! »

Help!

There are several places to get help with Enzo, from mailing lists to documentation to online tutorials and recordings of workshop presentations.

Help me out! »

Developing

Enzo is developed in the open by a community of developers from different institutions. Contributions, fixes, and changes are all welcomed!

Develop! »

Community

There are several places to get help with Enzo, from mailing lists to documentation to online tutorials and recordings of workshop presentations.

Engage »

enzo-project.org

Enzo | Quick Links | Home | Get Enzo | Help! | Development | Community | Enzo Docs

The Enzo Project

July 10 2013: Enzo 2.3 has been released. View the Release Notes!

July 10 2013: Enzo method paper entitled *Enzo: An Adaptive Mesh Refinement Code for Astrophysics* has been released. Get it here.

What is Enzo?

Enzo is a community-developed adaptive mesh refinement simulation code, designed for rich, multi-physics hydrodynamic astrophysical calculations.

Enzo is freely available, developed in the open, with a strong support structure for assistance. Simulations conducted with Enzo have been featured in numerous refereed journal articles, and it is capable of running on computers from laptop to Top500.

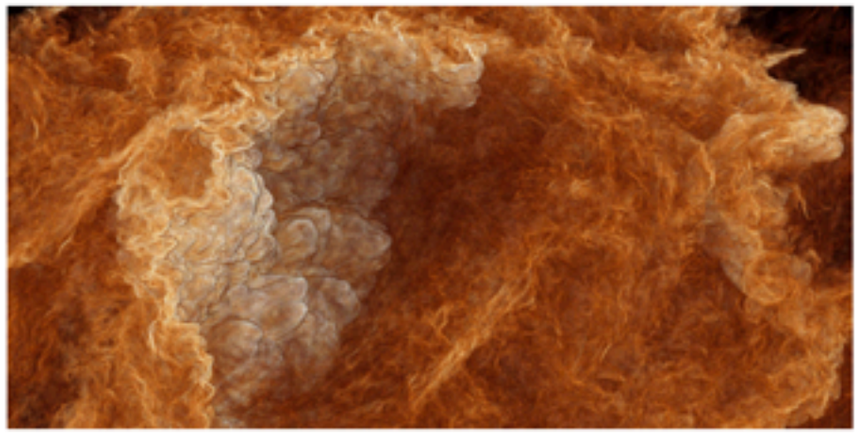


Image credit: Alexei Kritsuk, Paolo Padoan & Mike Norman

Getting Enzo

Enzo can be obtained in several places, corresponding to the degree of stability and development accessibility.

Let's go! »

Help!

There are several places to get help with Enzo, from mailing lists to documentation to online tutorials and recordings of workshop presentations.

Help me out! »

Developing

Enzo is developed in the open by a community of developers from different institutions. Contributions, fixes, and changes are all welcomed!

Develop! »

Community

There are several places to get help with Enzo, from mailing lists to documentation to online tutorials and recordings of workshop presentations.

Engage »

DOCUMENTATION

Enzo 2.3 documentation

Welcome to Enzo's documentation!

This is the development site for Enzo, an adaptive mesh refinement (AMR), grid-based hybrid code (hydro + N-Body) which is designed to do simulations of cosmological structure formation. Links to documentation and downloads for all versions of Enzo from 1.0 on are available.

Enzo development is supported by grants AST-0808184 and OCI-0832662 from the National Science Foundation.

- Enzo Public License
- Getting Started with Enzo
  - Obtaining and Building Enzo
  - How to run an Enzo test problem
  - How to run a cosmology simulation
  - Sample inits and Enzo parameter files
  - Writing Enzo Parameter Files
  - Data Analysis Basics
  - Controlling Enzo data output
- User Guide
  - Executables, Arguments, and Outputs
  - Running Enzo
  - Running Enzo with CUDA
  - Enzo Test Suite
  - Creating Cosmological Initial Conditions
  - Running Large Simulations
  - Enzo Output Formats
  - Analyzing With YT
  - Simulation Names and Identifiers
  - Embedded Python
  - The Enzo Hierarchy File - Explanation and Usage
  - Enzo Flow Chart, Source Browser

TABLE OF CONTENTS

Enzo Public License
Getting Started with Enzo
User Guide
Enzo Parameter List
Physics Modules in Enzo
Developer's Guide
Reference Information
Presentations Given About Enzo

SEARCH

Go

Enter search terms or a module, class or function name.

DOCUMENTATION

Enzo 2.3 documentation

NEXT

Welcome to Enzo's documentation!

This is the development site for Enzo, an adaptive mesh refinement (AMR), grid-based hybrid code (hydro + N-Body) which is designed to do simulations of cosmological structure formation. Links to documentation and downloads for all versions of Enzo from 1.0 on are available.

Enzo development is supported by grants AST-0808184 and OCI-0832662 from the National Science Foundation.

- Enzo Public License
- Getting Started with Enzo
  - Obtaining and Building Enzo
  - How to run an Enzo test problem
  - How to run a cosmology simulation
  - Sample inits and Enzo parameter files
  - Writing Enzo Parameter Files
  - Data Analysis Basics
  - Controlling Enzo data output
- User Guide
  - Executables, Arguments, and Outputs
  - Running Enzo
  - Running Enzo with CUDA
  - Enzo Test Suite
  - Creating Cosmological Initial Conditions
  - Running Large Simulations
  - Enzo Output Formats
  - Analyzing With YT
  - Simulation Names and Identifiers
  - Embedded Python
  - The Enzo Hierarchy File - Explanation and Usage
  - Enzo Flow Chart, Source Browser

TABLE OF CONTENTS

Enzo Public License
Getting Started with Enzo
User Guide
Enzo Parameter List
Physics Modules in Enzo
Developer's Guide
Reference Information
Presentations Given About Enzo

SEARCH

Go

Enter search terms or a module, class or function name.

on the internet or
on your computer

EMAIL LIST

Quick Help

Send to:

enzo-users@googlegroups.com

Archives:

http://groups.google.com/group/enzo-users

IRC CHANNEL

Live Help

Come say hello!
http://enzo-project.org/irc.html

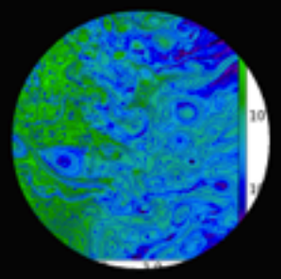
yt-project.org

yt project | About | Docs | Community | Develop | Gallery | Data Hub | Quick Links

Volumetric Data Analysis

yt is a python package for analyzing and visualizing volumetric, multi-resolution data from astrophysical simulations, radio telescopes, and a burgeoning interdisciplinary community.

Get yt



Data-Driven

yt is designed to provide a consistent, cross-code interface to analyzing and visualizing astrophysical simulation data from a physical perspective.

Community

yt is composed of a friendly community of users and developers. We want to make it easy to use and develop — we'd love it if you got involved!

Free Software

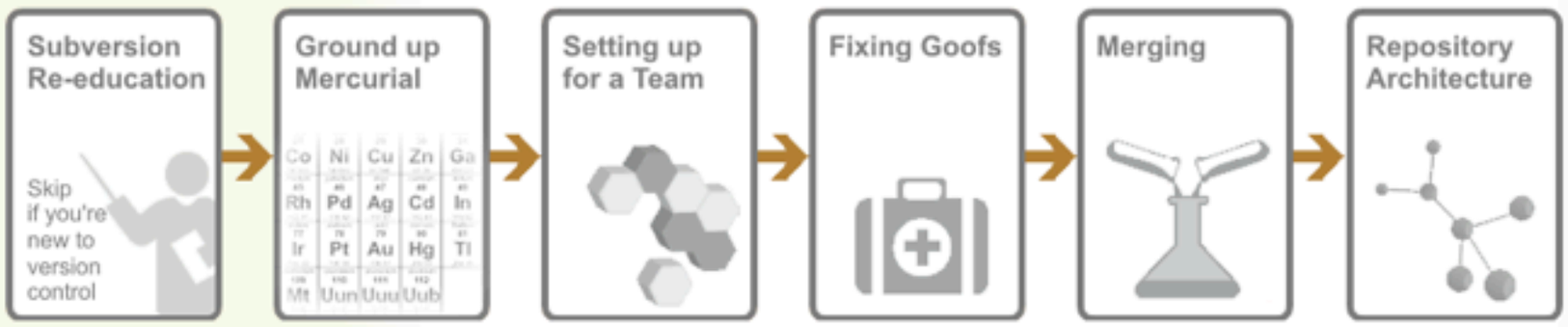
yt is developed completely in the open under a 3-clause BSD license. The developers are committed to open source practices and fidelity of scientific results.

hginit.com

80
Hg
Init
200.59

Hg Init: a Mercurial tutorial

Mercurial is a modern, open source, distributed version control system, and a compelling upgrade from older systems like Subversion. In this user-friendly, six-part tutorial, Joel Spolsky teaches you the key concepts. Also, Fog Creek offers free monthly webinars that go over the basics of Mercurial.



DEVELOPMENT

Bitbucket Repositories Create owner/repository

enzo-dev

enzo Share

Clone Branch Pull request ...

Overview Source Commits Branches Pull requests 3 Issues 15 Downloads

```
= ENZO =

ENZO IS AN OPEN SOURCE CODE.  We encourage you to take it, inspect it, use it,
and contribute back any changes you have made.  We strive to make the the Enzo
community a community of *developers*.

== RESOURCES ==

Enzo's main webpage is:

 * http://enzo-project.org

Enzo is developed in the open on bitbucket.org:

 * Stable Version: https://bitbucket.org/enzo/enzo-stable
 * Development Version: https://bitbucket.org/enzo/enzo-dev

Documentation, including instructions for compilation, can be found at:

 * http://enzo-project.org/docs/2.2/

Please subscribe to the Enzo Users' mailing list at:

 * https://mailman.ucsd.edu/mailman/listinfo/enzo-users-l

You can also follow the Enzo Project on google+ at:

 * https://plus.google.com/115923030596894217717

If you have received this source code through an archive, rather than the
mercurial version control system, we highly encourage you to upgrade to the
version controlled source, as no support can be provided for archived
("tarball") sources.

== DEVELOPERS ==

Many people have contributed to the development of Enzo -- here's just a short
list of the people who have recently contributed, in alphabetical order:

   * Tom Abel              tabel@stanford.edu
   * James Bordner         jobordner@ucsd.edu
```

HTTPS https://brittonsmith@bitbucket

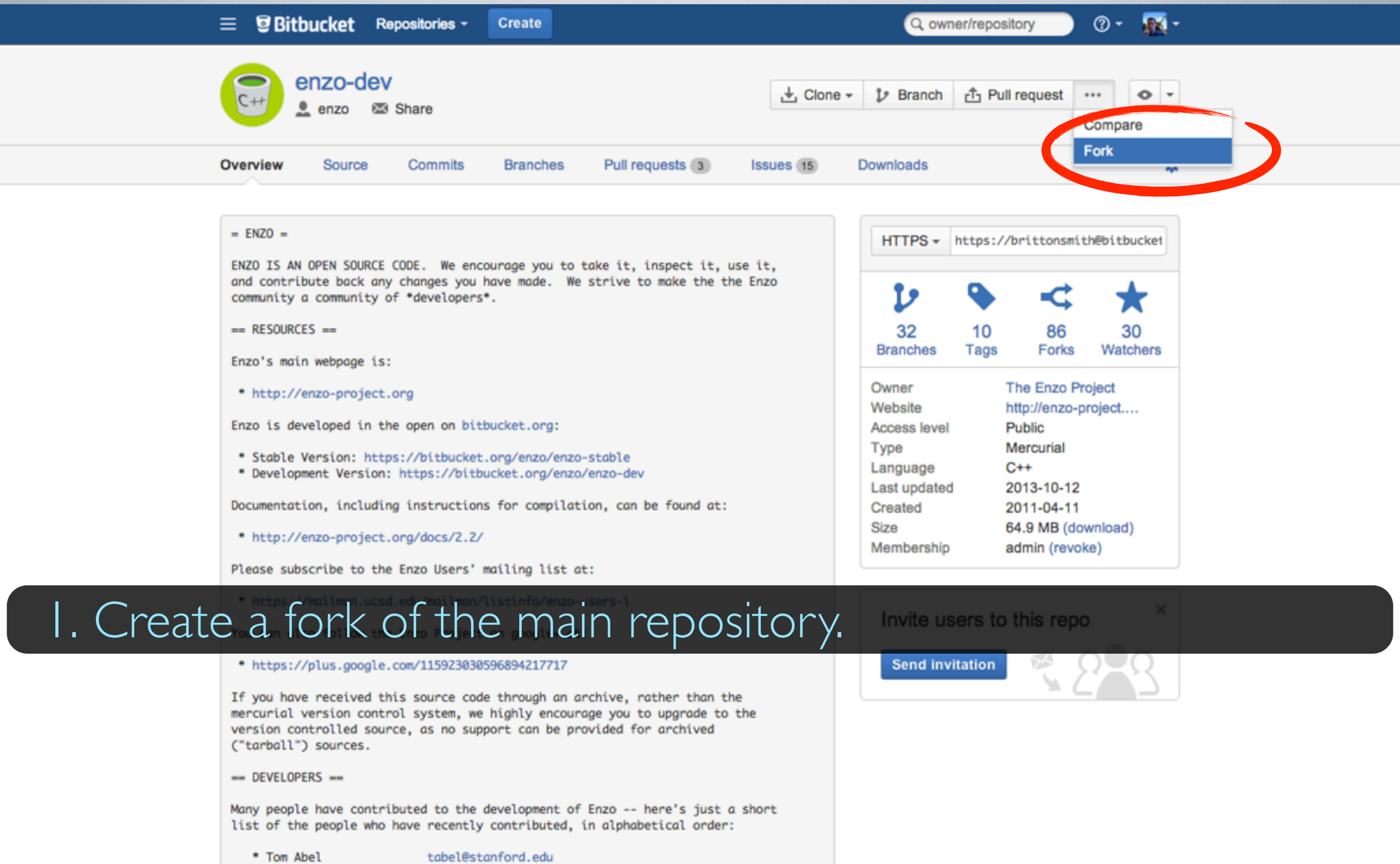
32	10	86	30
Branches	Tags	Forks	Watchers

Owner	The Enzo Project
Website	http://enzo-project....
Access level	Public
Type	Mercurial
Language	C++
Last updated	2013-10-12
Created	2011-04-11
Size	64.9 MB (download)
Membership	admin (revoke)

Invite users to this repo

Send invitation

DEVELOPMENT



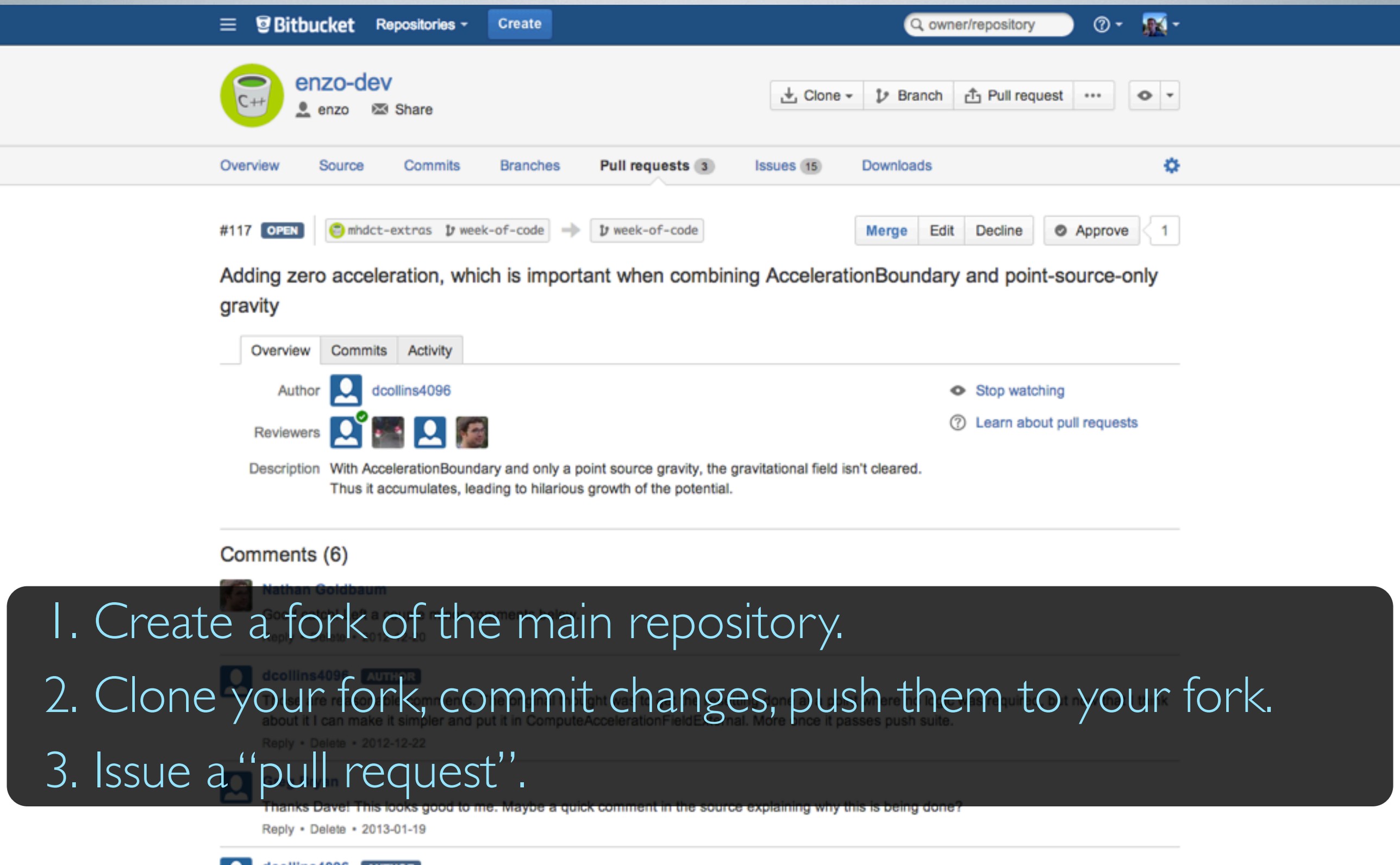
1. Create a fork of the main repository.

DEVELOPMENT

```
grackle:~[1] hg clone https://bitbucket.org/brittonsmith/enzo-dev
warning: bitbucket.org certificate with fingerprint 24:9c:45:8b:9c:aa:ba:55:4e:01:6d:
eck hostfingerprints or web.cacerts config setting)
destination directory: enzo-dev
requesting all changes
adding changesets
changesets [
```

1. Create a fork of the main repository.
2. Clone your fork, commit changes, push them to your fork.

DEVELOPMENT



1. Create a fork of the main repository.
2. Clone your fork, commit changes, push them to your fork.
3. Issue a "pull request".

DEVELOPMENT

1. Create a fork of the main repository.
2. Clone your fork, commit changes, push them to your fork.
3. Issue a "pull request".
4. Other developers review changes, make comments, accept.

DEVELOPMENT

1. Create a fork of the main repository.
2. Clone your fork, commit changes, push them to your fork.
3. Issue a "pull request".
4. Other developers review changes, make comments, accept.

DEVELOPERS

Many people have contributed to the development of Enzo -- here's just a short list of the people who have recently contributed, in alphabetical order:

* Tom Abel tabel@stanford.edu
* James Bordner jobordner@ucsd.edu
* Greg Bryan gbryan@astro.columbia.edu
* Renyue Cen cen@astro.princeton.edu
* Dave Collins dcollins@physics.ucsd.edu
* Brian Crosby crosby.bd@gmail.com
* Nathan Goldbaum goldbaum@ucolick.org
* Oliver Hahn hahn@phys.ethz.ch
* Robert Harkness harkness@sdsc.edu
* Elizabeth Harper-Clark h-clark@astro.utoronto.ca
* Cameron Hummels chummels@gmail.com
* Ji-hoon Kim mornkr@ucolick.org
* Daegene Koh dkoh30@gatech.edu
* Alexei Kritsuk akritsuk@ucsd.edu
* Michael Kuhlen mqk@astro.berkeley.edu
* Eve Lee elee@cita.utoronto.ca
* Yuan Li yuan@astro.columbia.edu
* Michael Norman mlnorman@ucsd.edu
* JS Oishi jsoishi@gmail.com
* Brian O'Shea oshea@msu.edu
* Pascal Paschos ppaschos@minbari.ucsd.edu
* Carolyn Peruta perutaca@msu.edu
* Alex Razoumov razoumov@gmail.com
* Dan Reynolds reynolds@smu.edu
* Munier Salem msalem@astro.columbia.edu
* Christine Simpson csimpson@astro.columbia.edu
* Samuel Skillman samskillman@gmail.com
* Stephen Skory s@skory.us
* Britton Smith brittonsmith@gmail.com
* Geoffrey So gsiisg@gmail.com
* Elizabeth Tasker tasker@astro1.sci.hokudai.ac.jp
* Matthew Turk matthewturk@gmail.com
* Rick Wagner rwagner@physics.ucsd.edu
* Peng Wang pengw@slac.stanford.edu
* John Wise jwise@physics.gatech.edu
* Hao Xu haxu@ucsd.edu
* Fen Zhao fenzhao@stanford.edu

DEVELOPERS

Many people have contributed to the development of Enzo -- here's just a short list of the people who have recently contributed, in alphabetical order:

* Tom Abel	tabel@stanford.edu
* James Bordner	jobordner@ucsd.edu
* Greg Bryan	gbryan@astro.columbia.edu
* Renyue Cen	cen@astro.princeton.edu
* Dave Collins	dcollins@physics.ucsd.edu
* Brian Crosby	crosby.bd@gmail.com
* Nathan Goldbaum	goldbaum@ucolick.org
* Oliver Hahn	hahn@phys.ethz.ch
* Robert Harkness	harkness@sdsc.edu
* Elizabeth Harper-Clark	h-clark@astro.utoronto.ca
* Cameron Hummels	chummels@gmail.com
* Ji-hoon Kim	mornkr@ucolick.org
* Daegene Koh	dkoh30@gatech.edu
* Alexei Kritsuk	akritsuk@ucsd.edu
* Michael Kuhlen	mqk@astro.berkeley.edu
* Eve Lee	elee@cita.utoronto.ca
* Yuan Li	yuan@astro.columbia.edu
* Michael Norman	mlnorman@ucsd.edu
* JS Oishi	jsoishi@gmail.com
* Brian O'Shea	oshea@msu.edu
* Pascal Paschos	ppaschos@minbari.ucsd.edu
* Carolyn Peruta	perutaca@msu.edu
* Alex Razoumov	razoumov@gmail.com
* Dan Reynolds	reynolds@smu.edu
* Munier Salem	msalem@astro.columbia.edu
* Christine Simpson	csimpson@astro.columbia.edu
* Samuel Skillman	samskillman@gmail.com
* Stephen Skory	s@skory.us
* Britton Smith	brittonsmith@gmail.com
* Geoffrey So	gsiisg@gmail.com
* Elizabeth Tasker	tasker@astro1.sci.hokudai.ac.jp
* Matthew Turk	matthewturk@gmail.com
* Rick Wagner	rwagner@physics.ucsd.edu
* Peng Wang	pengw@slac.stanford.edu
* John Wise	jwise@physics.gatech.edu
* [illegible] Xu	haxu@ucsd.edu
* Fen Zhao	fenzhao@stanford.edu

- You!

THANK YOU