INTRODUCTION TO ENZO

Britton Smith

OUTLINE

I. Code Overview

II. Obtaining

III. Building

IV. Running

V. The Enzo Community

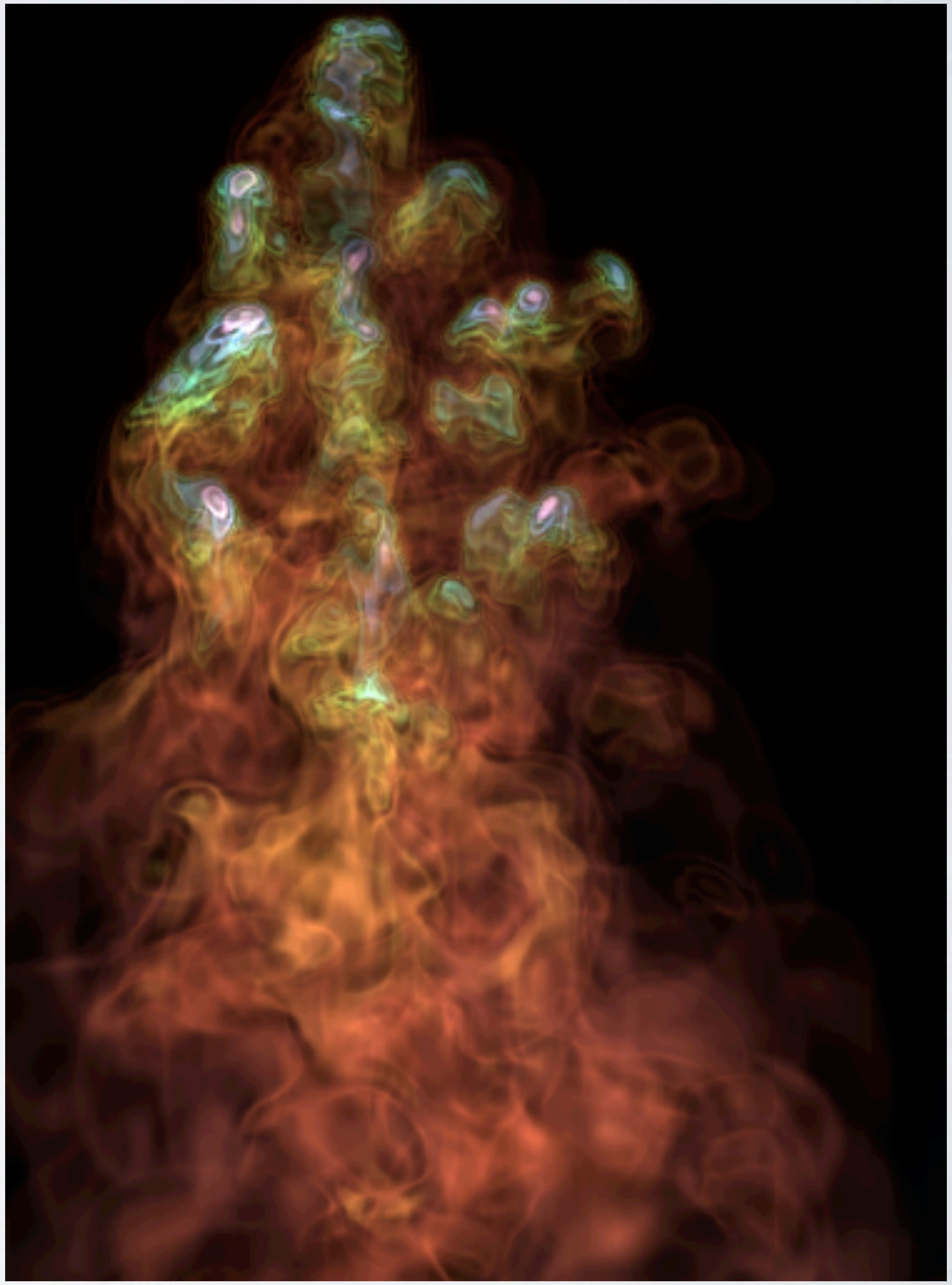


Image: Devin Silvia

WHAT IS ENZO?

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

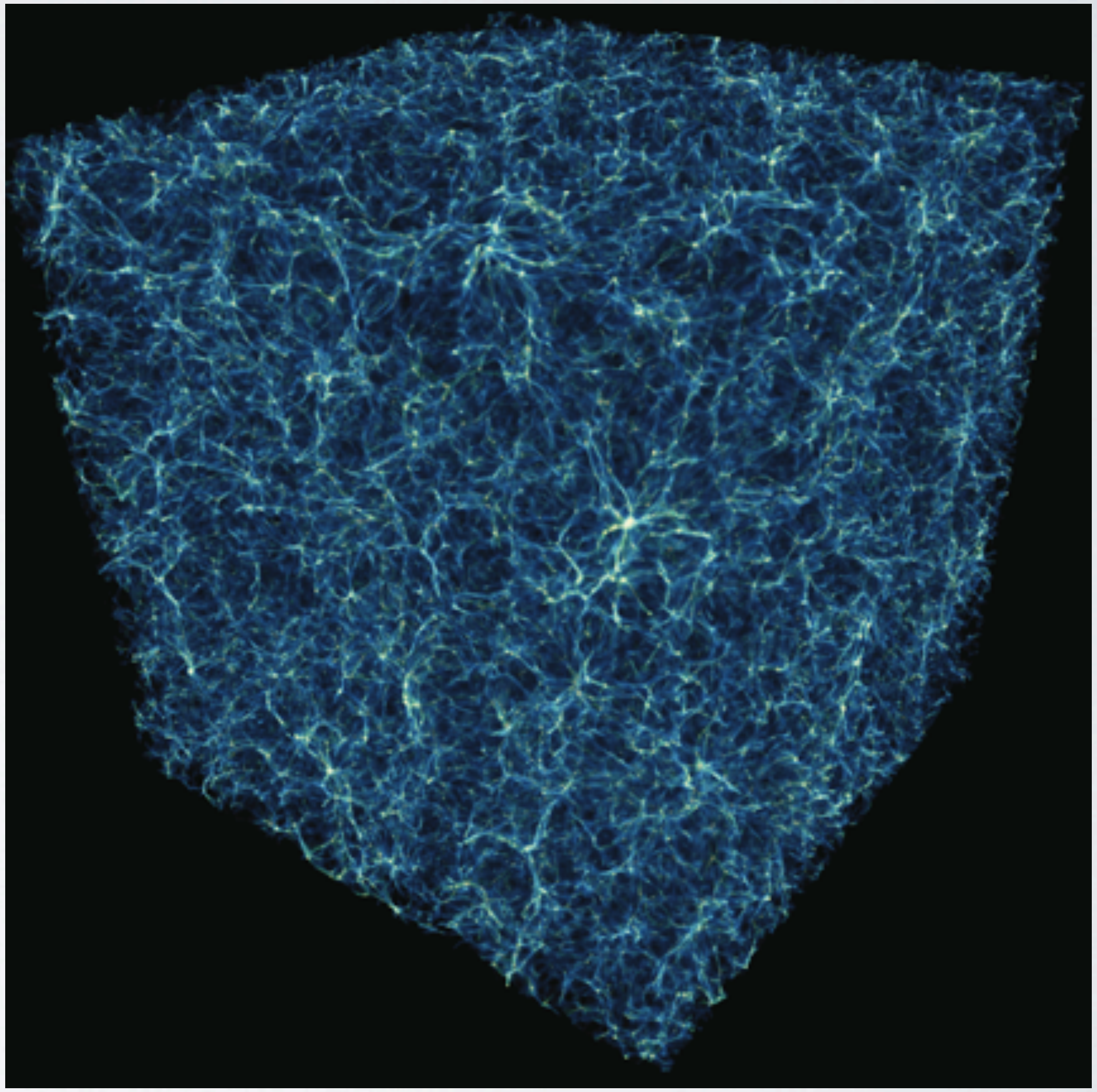
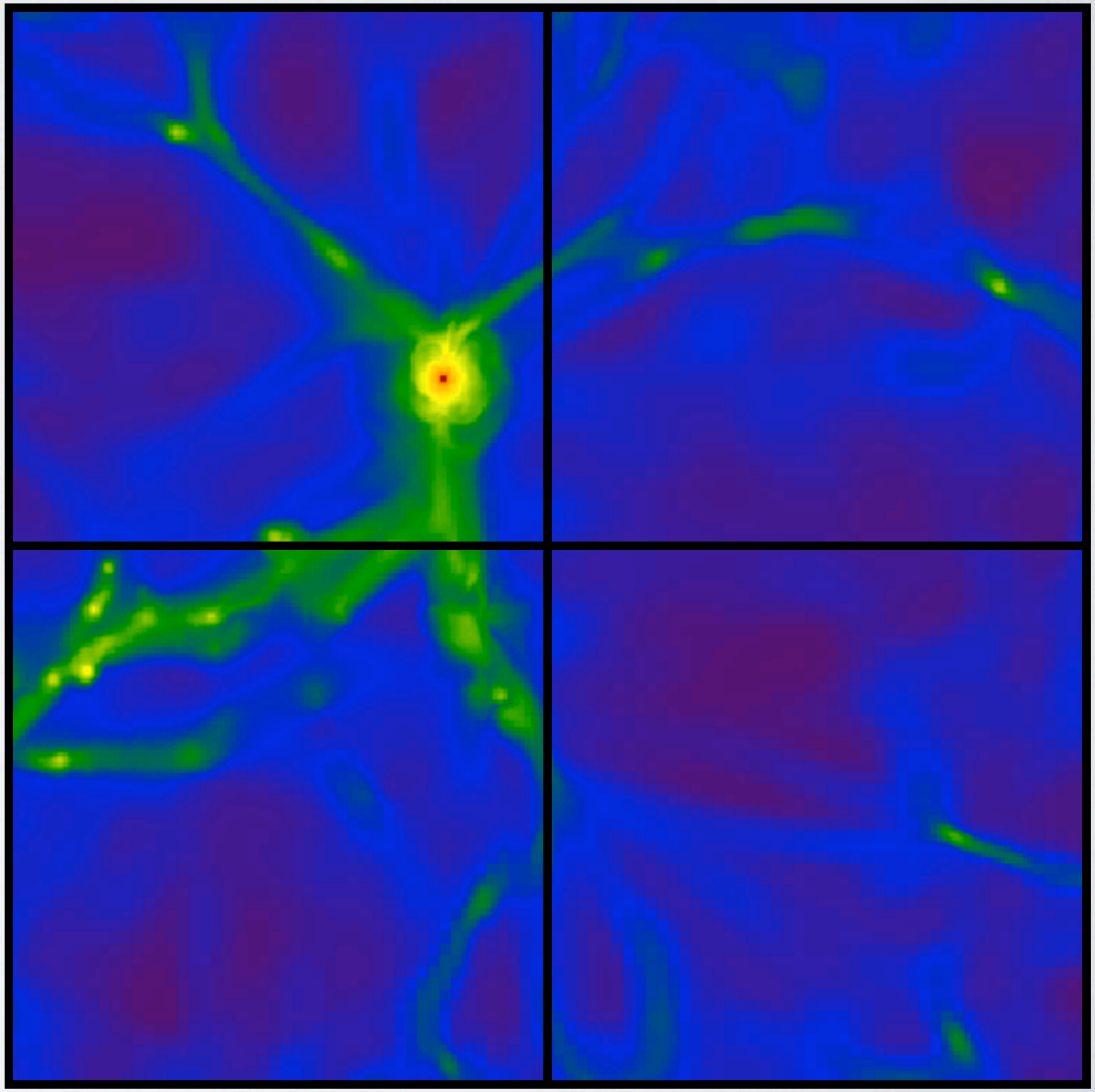


Image: Eric Hallman, Brian O'Shea

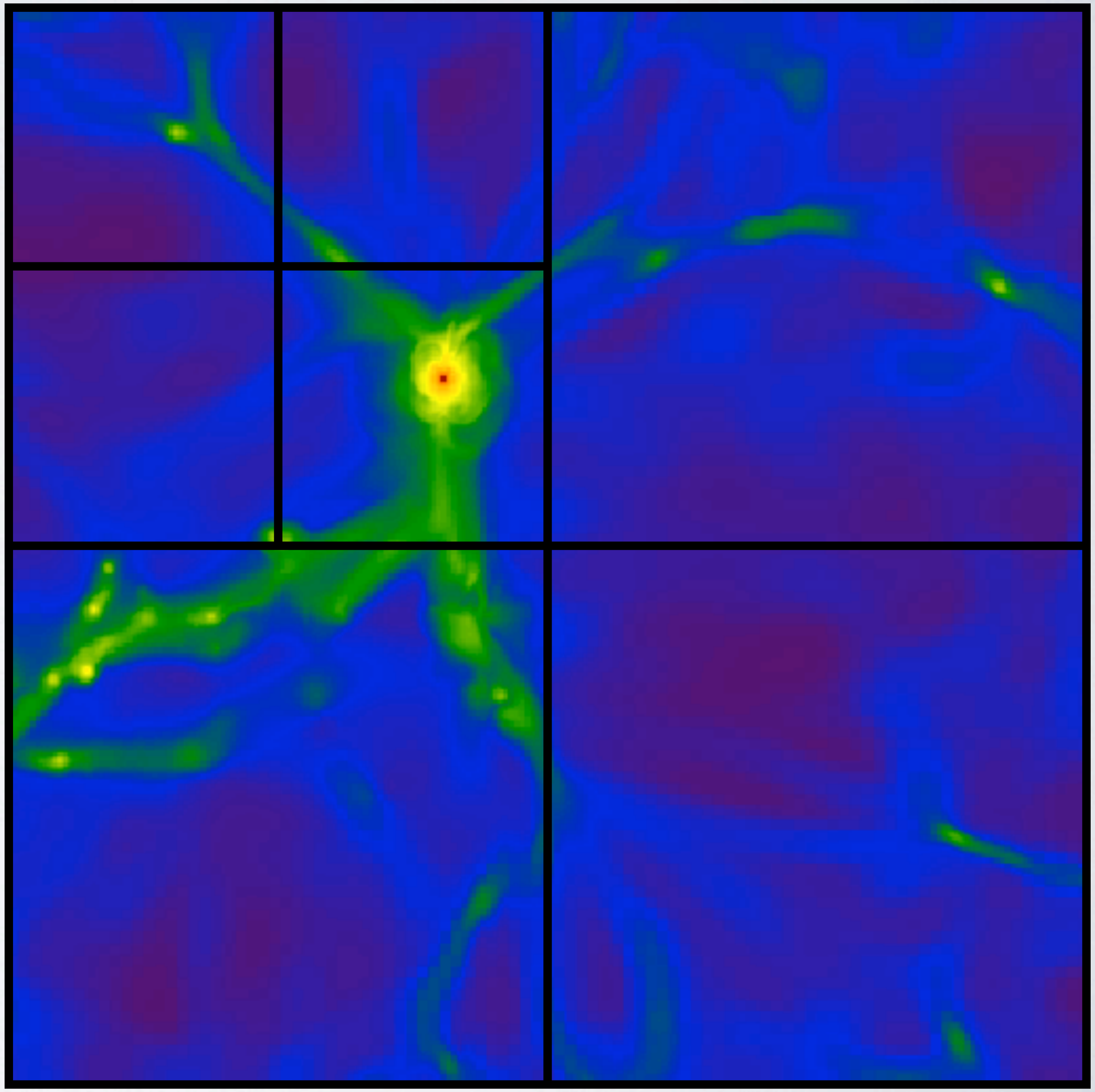
AMR:
ADAPTIVE-MESH REFINEMENT

- 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



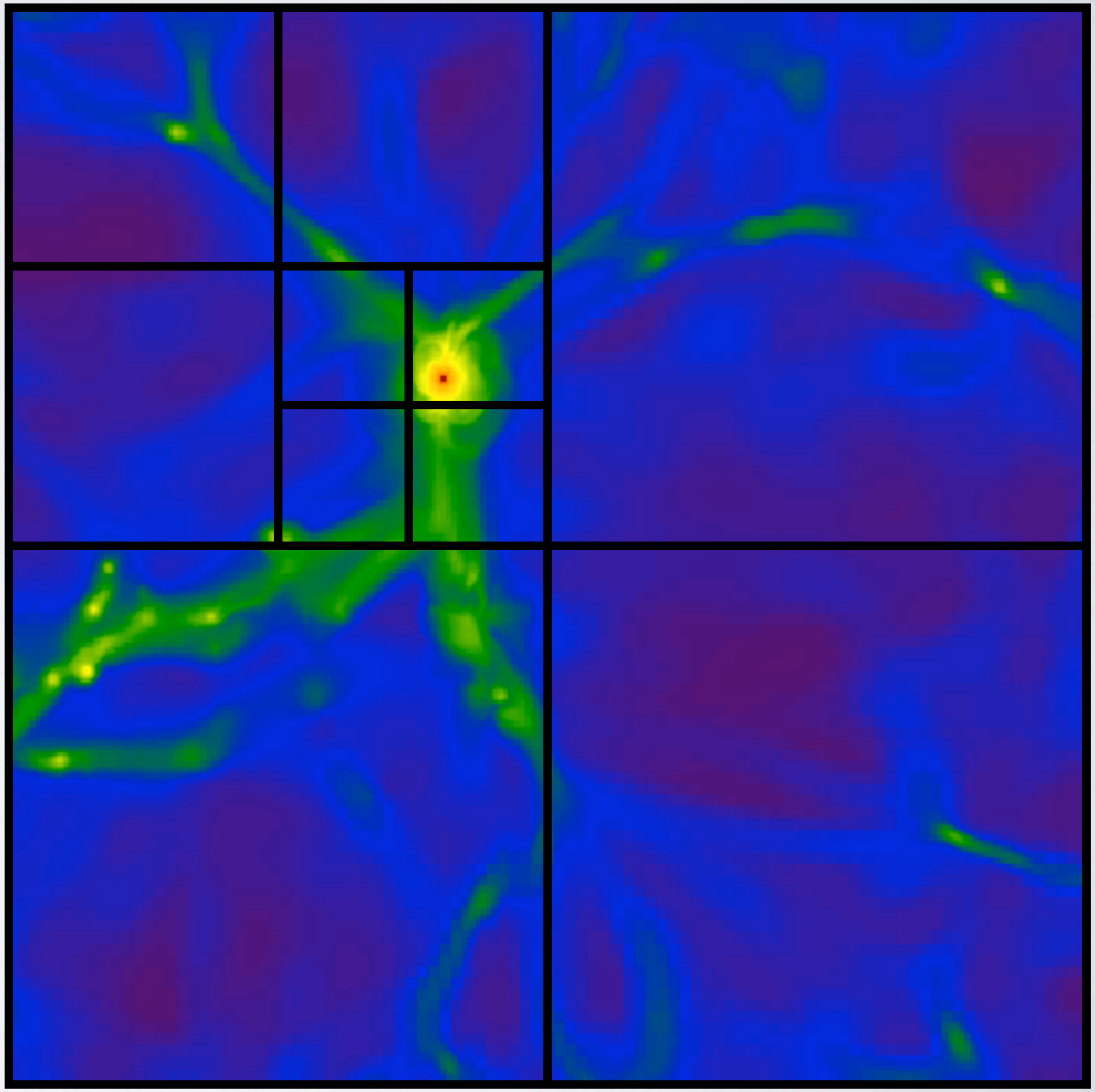
AMR:
ADAPTIVE-MESH REFINEMENT

- 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



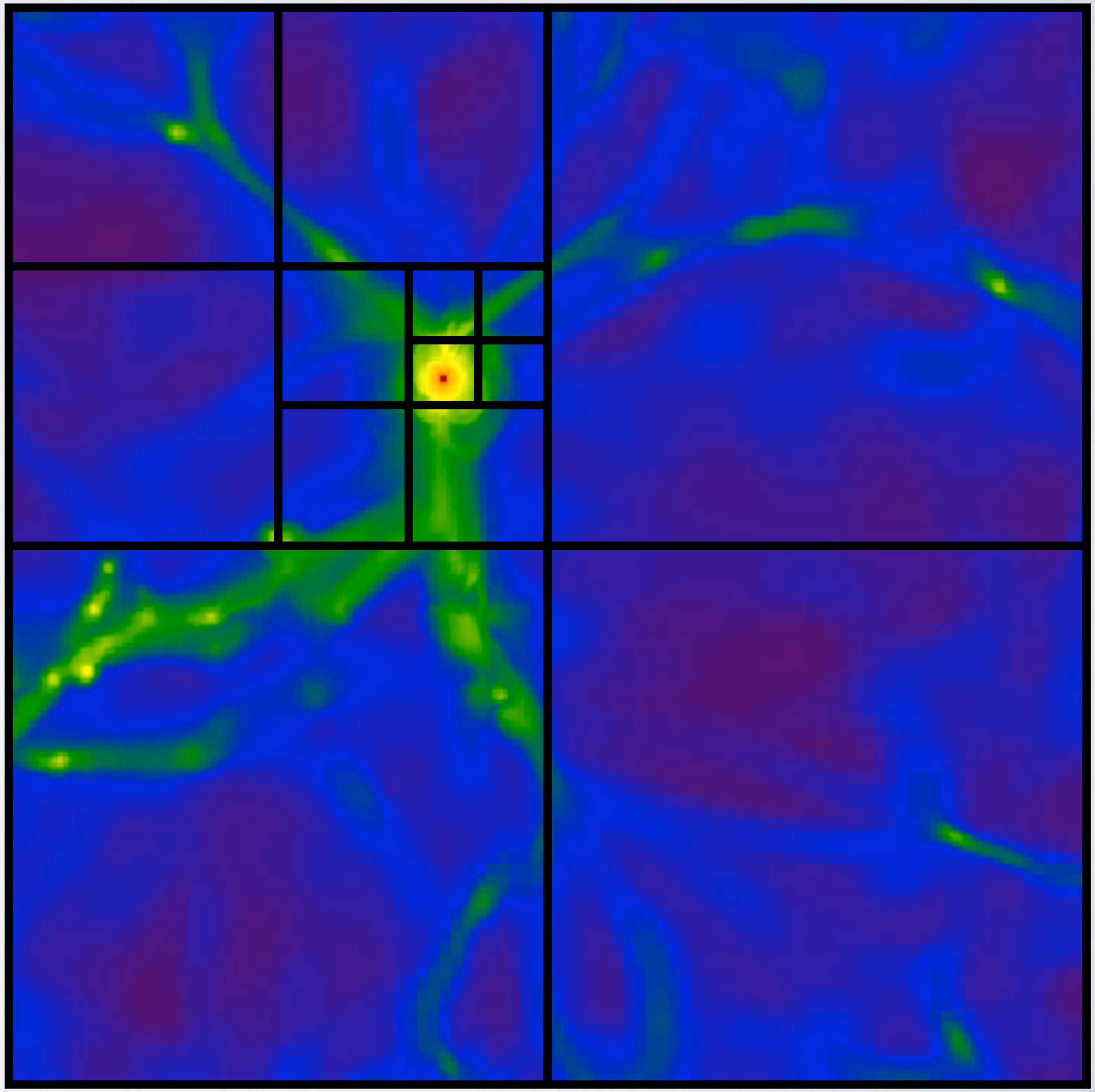
AMR:
ADAPTIVE-MESH REFINEMENT

- 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



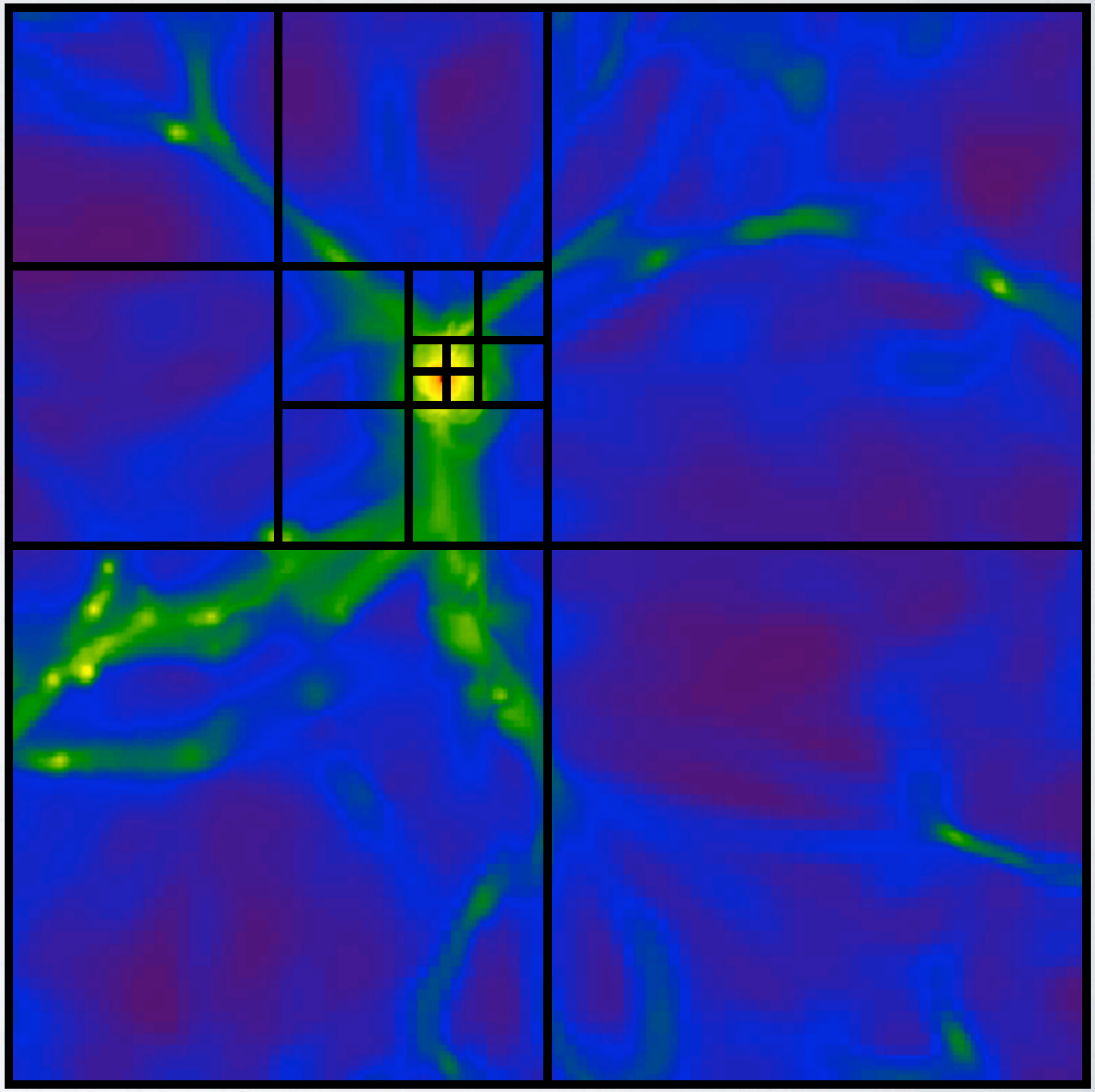
AMR: ADAPTIVE-MESH REFINEMENT

- 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



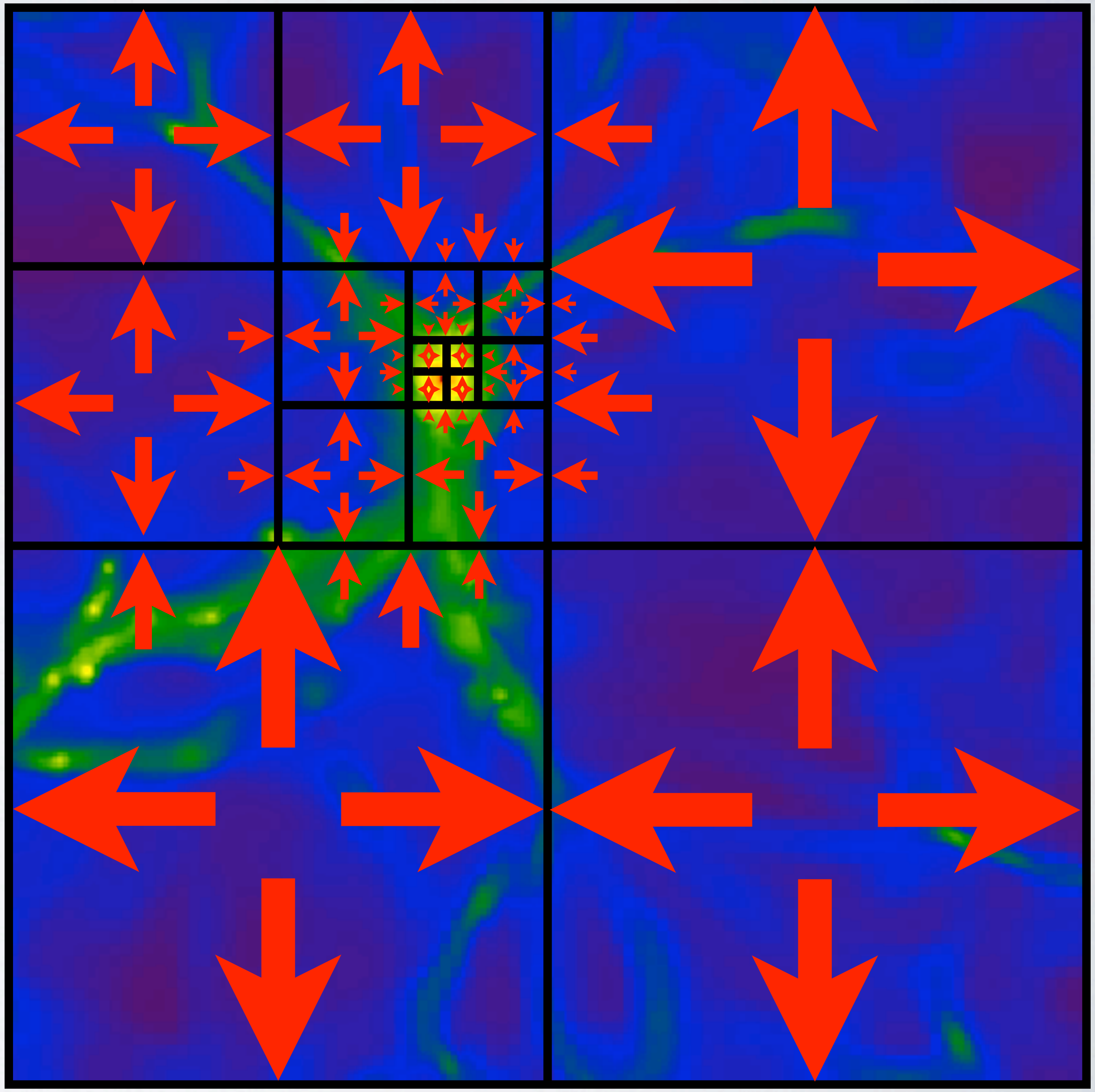
AMR:
ADAPTIVE-MESH REFINEMENT

- 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



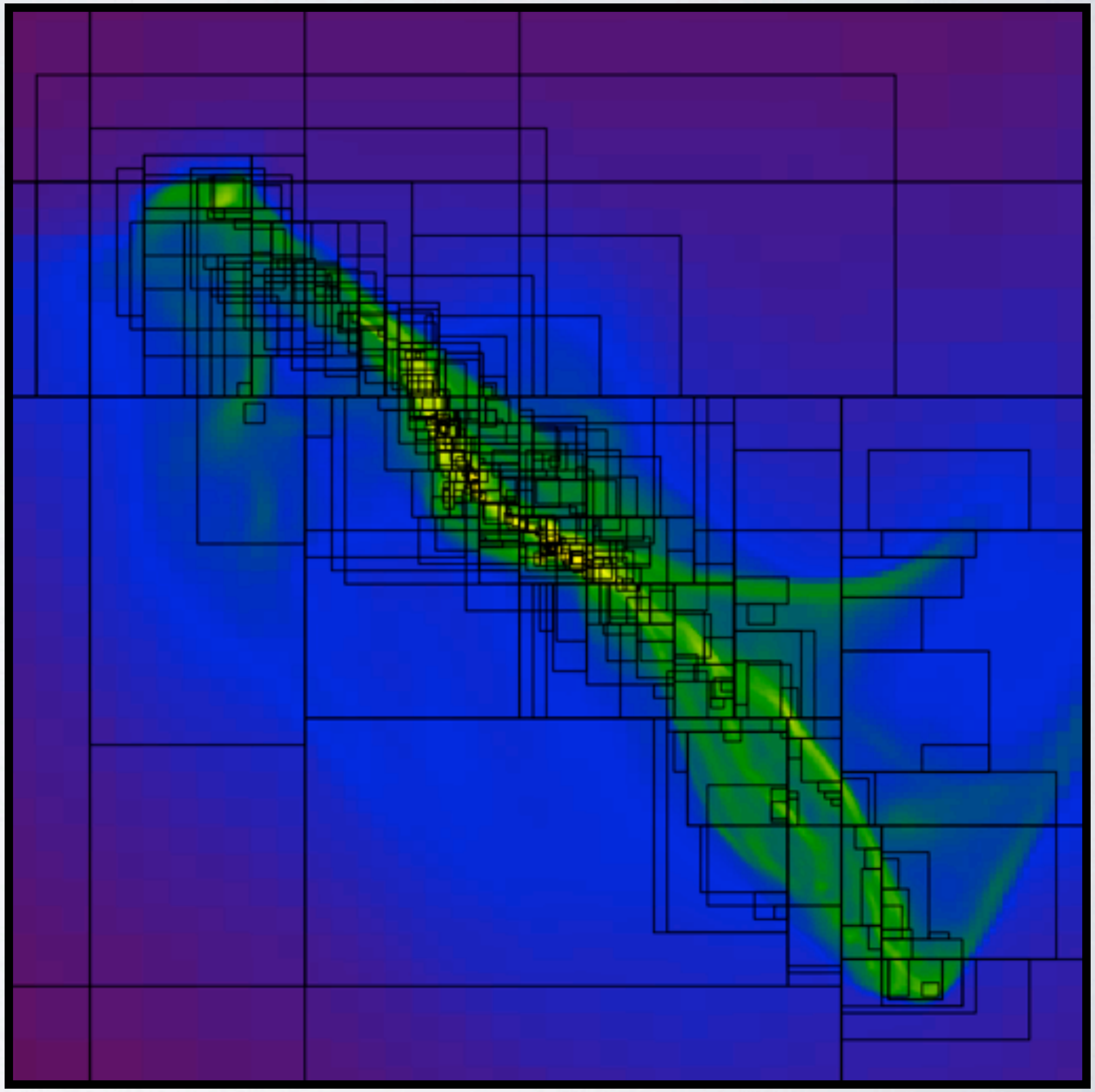
AMR:
ADAPTIVE-MESH REFINEMENT

- 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



Credit: 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)

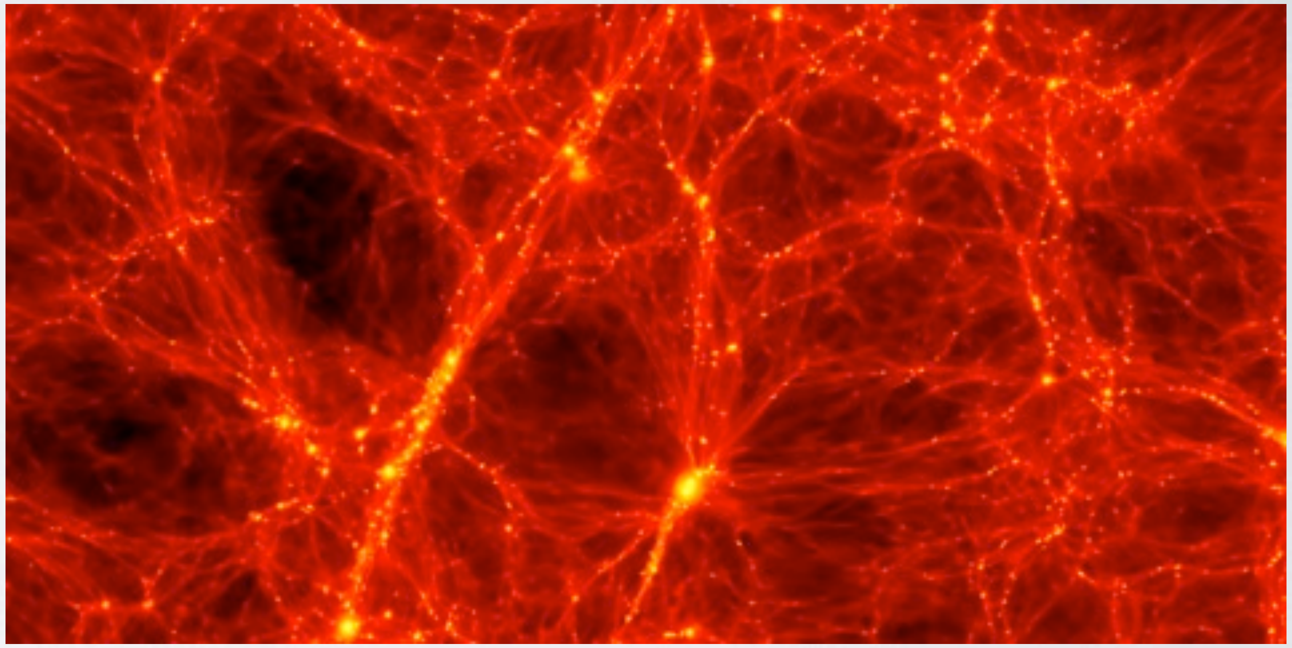


Image: Michael Norman et. al.

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 (now more stable thanks to improvements by John Wise)
- Zeus
  - less accurate, more diffusive
  - extremely robust (excellent for cosmology)

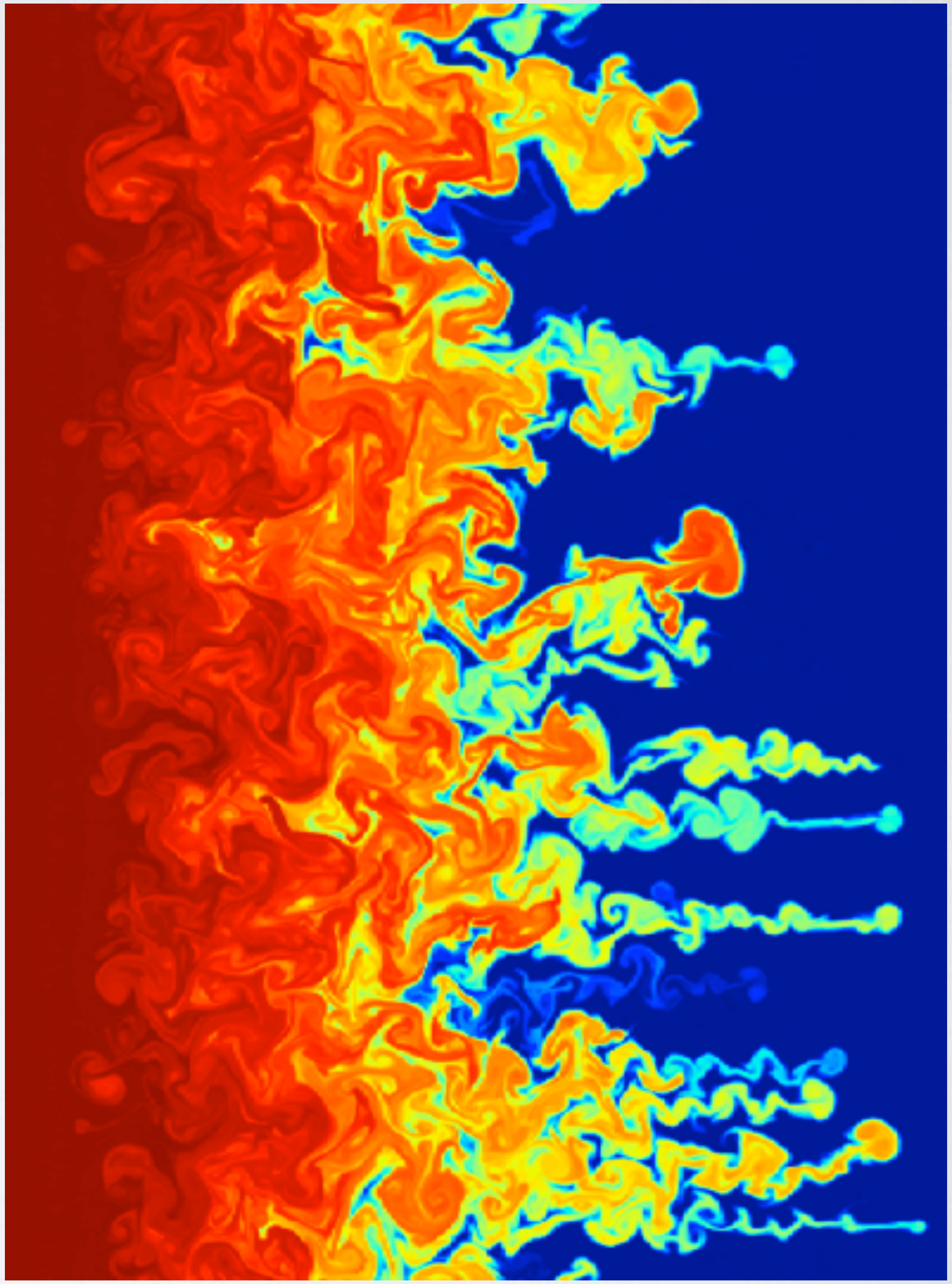


Image: Nick Earl

HYDRODYNAMICS

Multiple Hydro Methods

- MUSCL
  - 2^{nd} order accurate Godunov solver
  - 2^{nd} order Runge-Kutta time integration
  - multiple Riemann solvers and interpolation methods available
- MHD using MUSCL
  - uses MUSCL framework above
  - hyperbolic divergence cleaning method to ensure $\nabla \cdot B = 0$

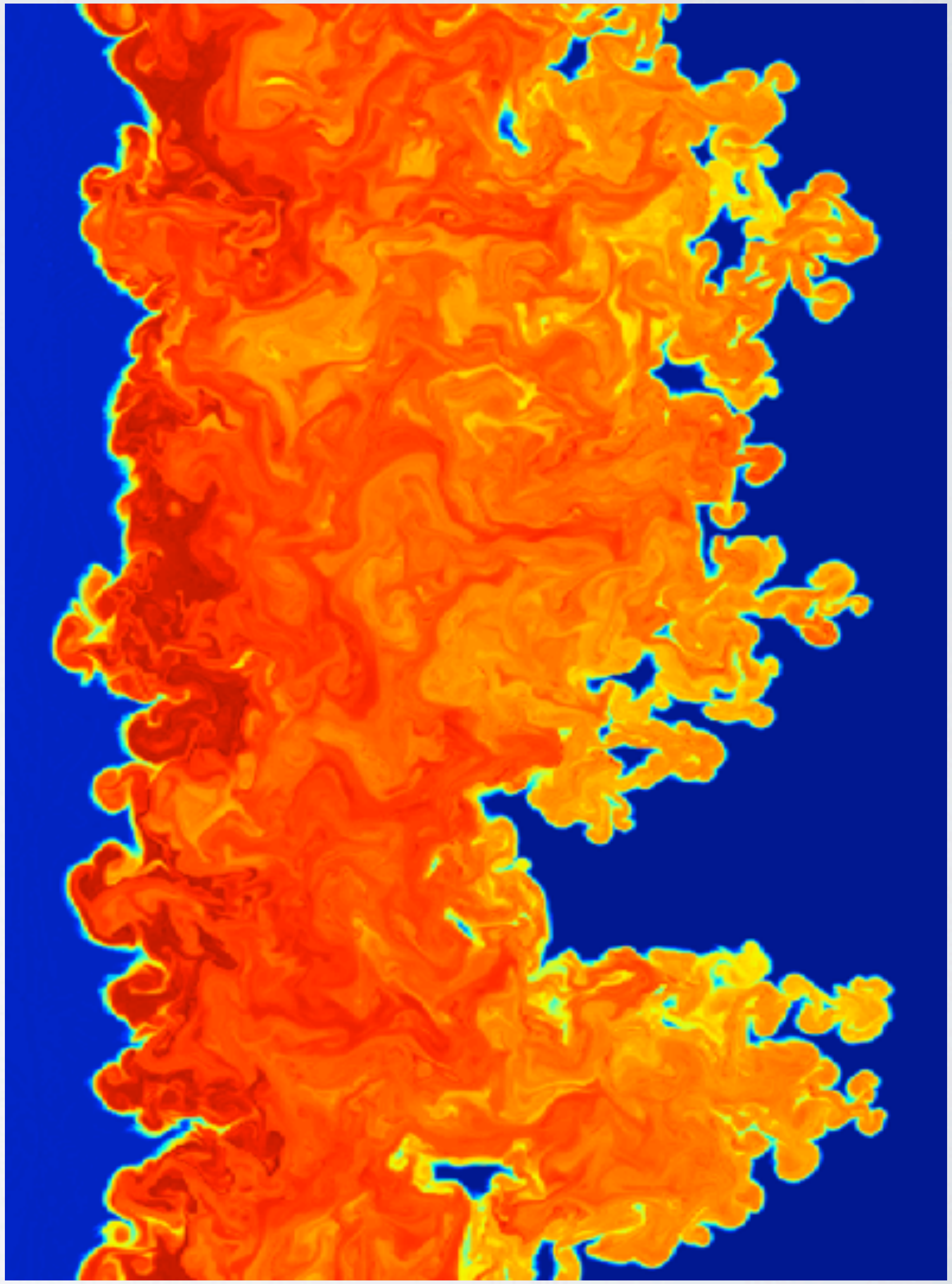


Image: Nick Earl

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

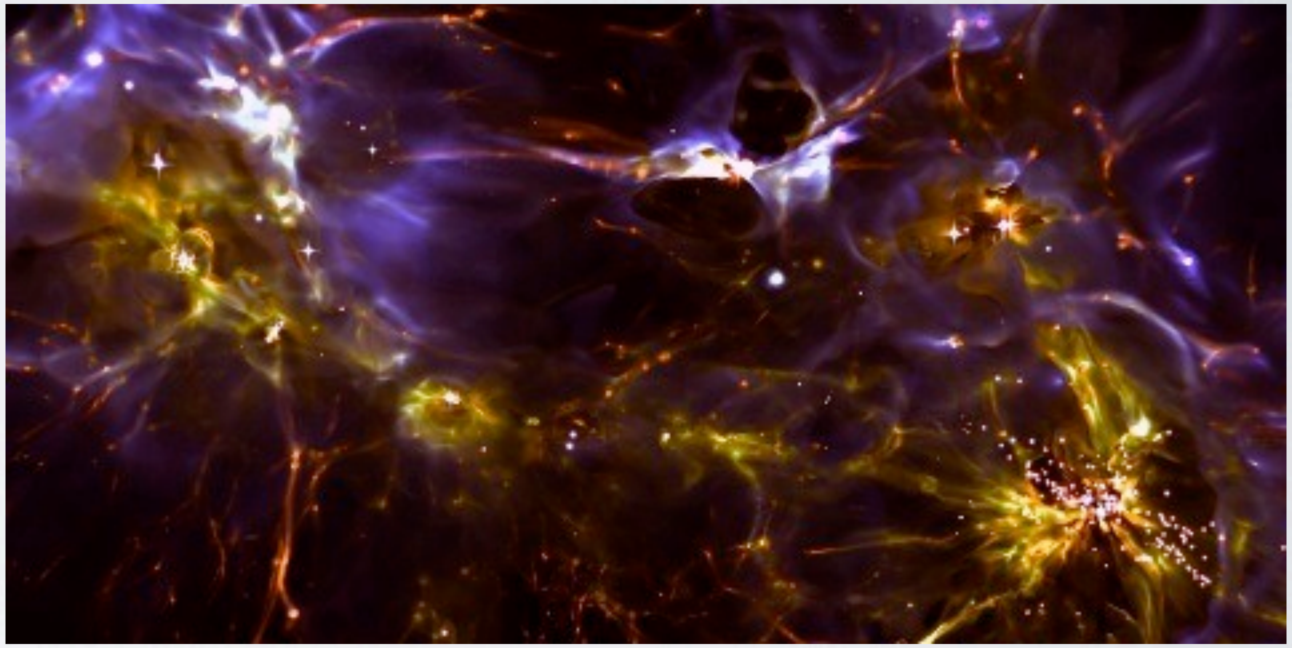


Image: John Wise

RADIATION BACKGROUNDS

- spatially smooth, time-dependent radiation fields
- UV metagalactic, ionizing backgrounds mimic effects of photo-heating and ionization during Reionization
- Lyman-Werner soft UV fields represent radiation from first stellar sources and photo-dissociate H_2

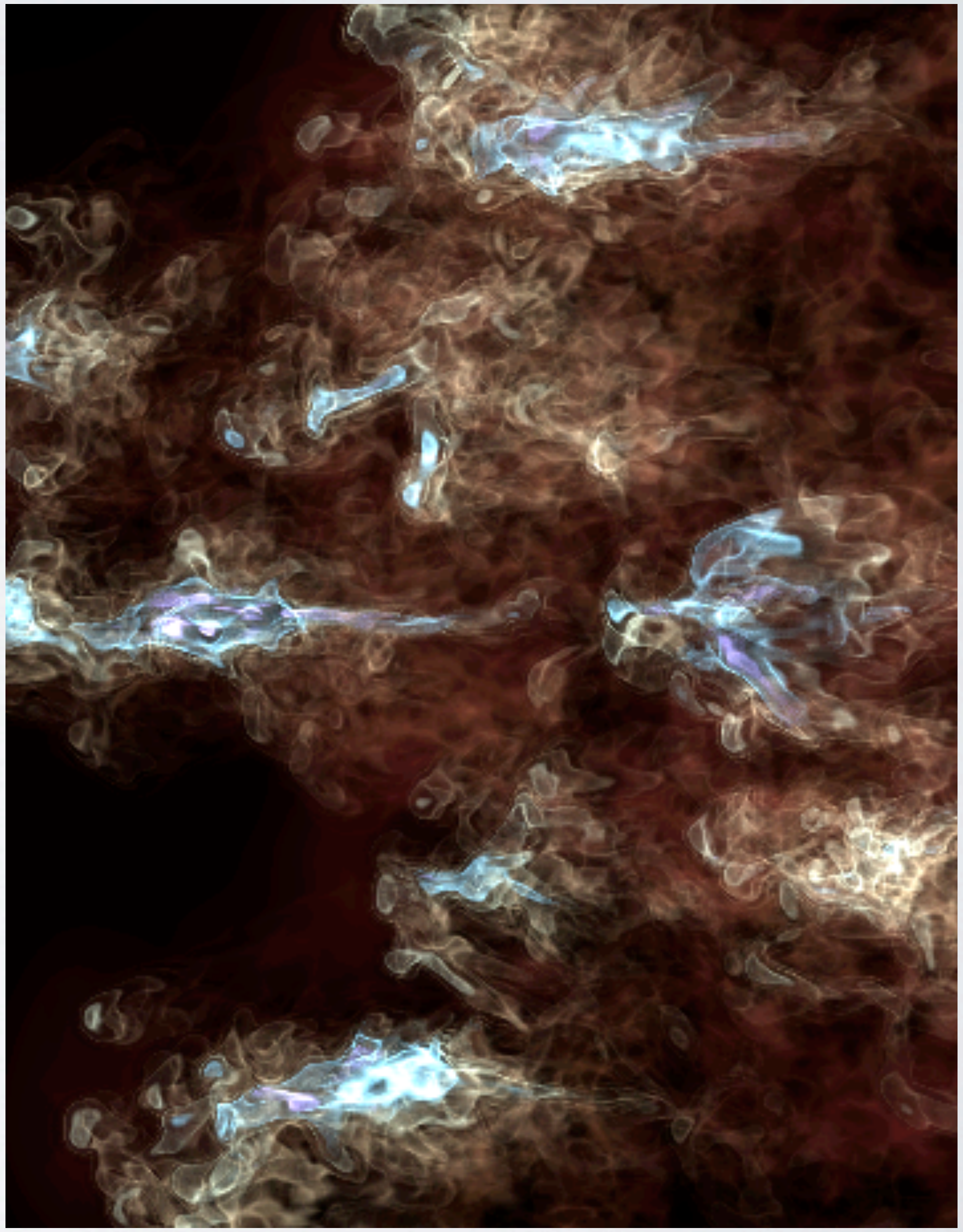


Image: Devin Silvia

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, chemical heating/cooling
- Metal cooling
  - simple tabulated rates ($T > 10^4$ K)
  - atomic fine-structure lines
  - multidimensional Cloudy tables
    - density, metallicity, temperature, electron fraction, background redshift
    - new tables can be made (different abundance patterns, input spectra)

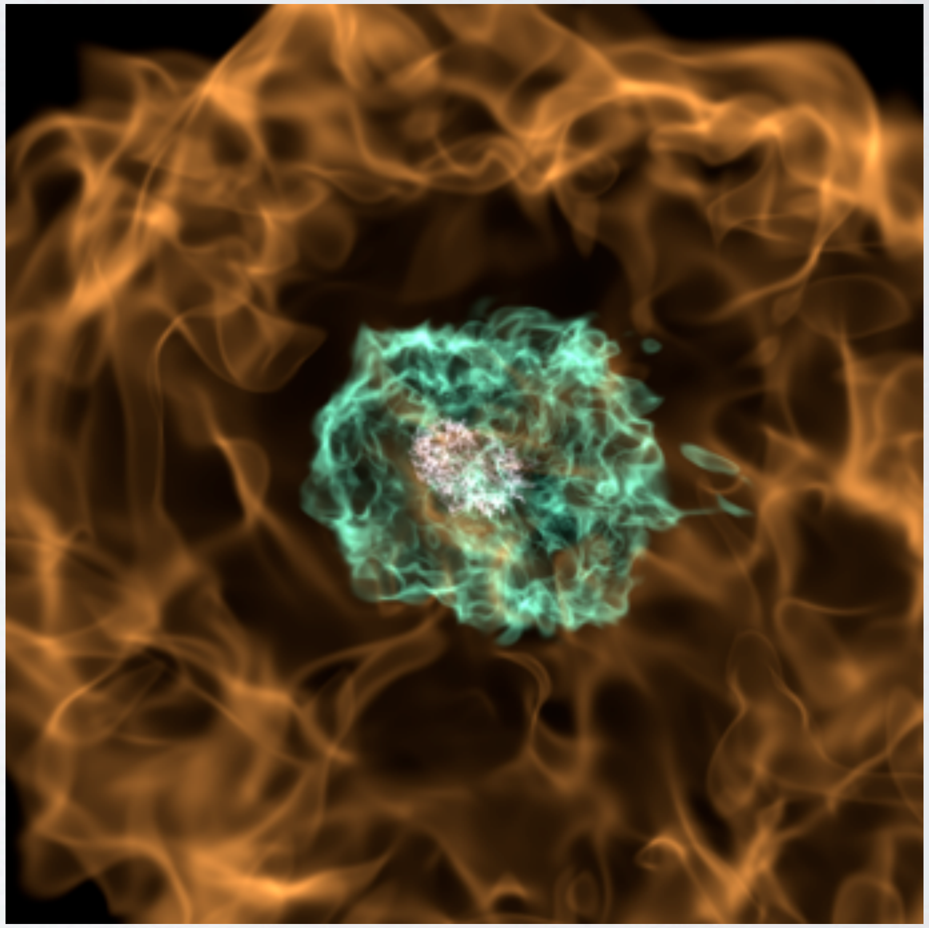
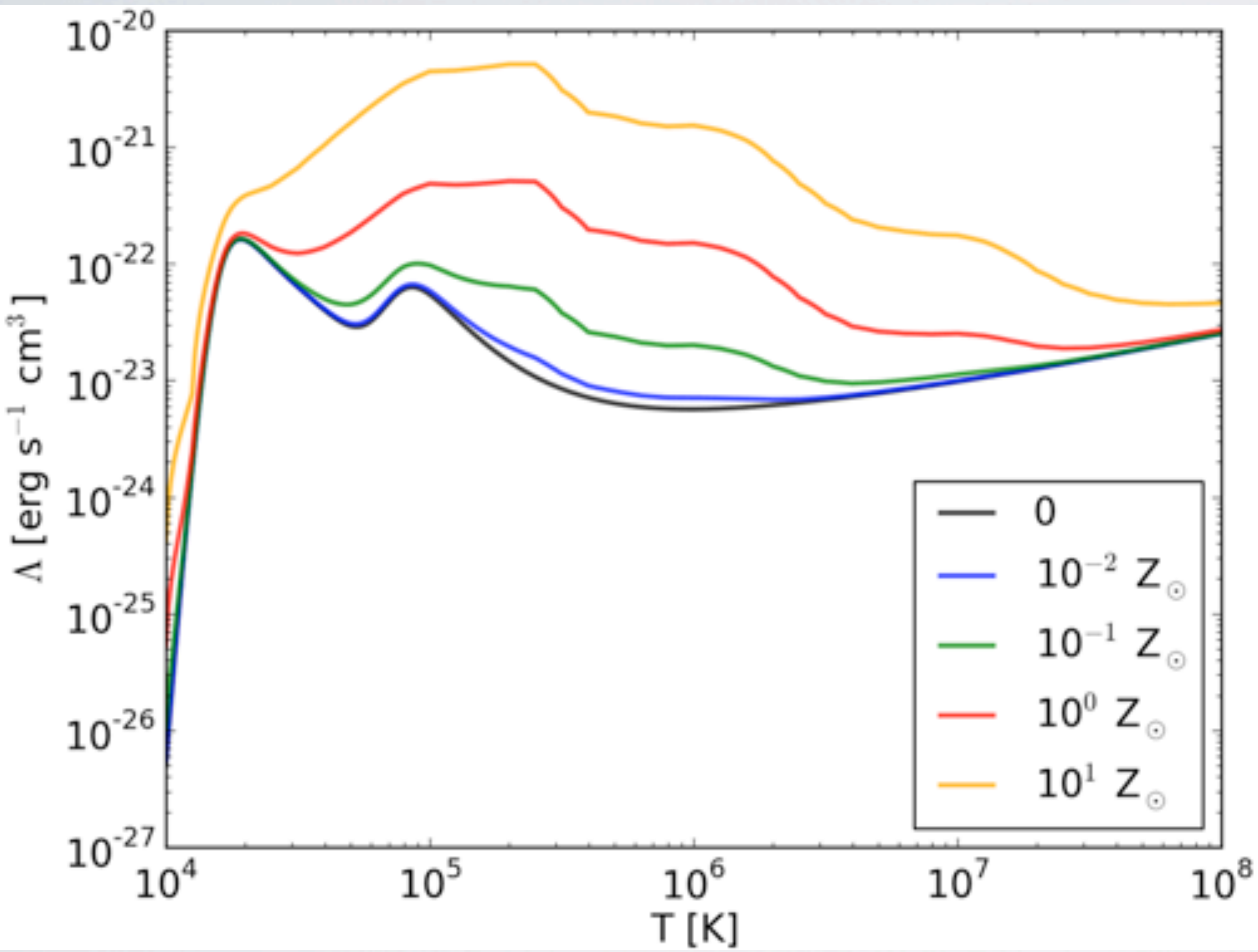
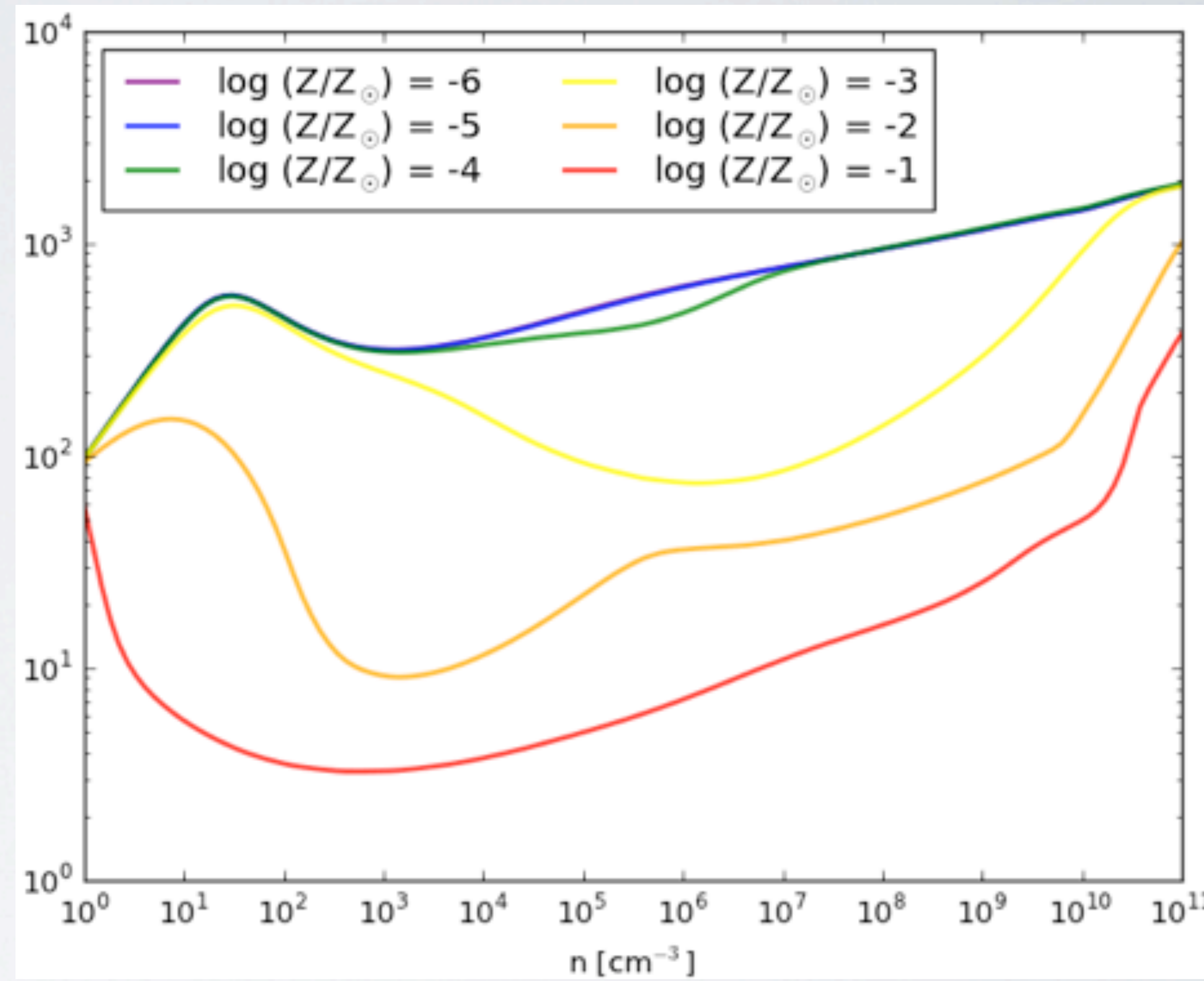


Image: Matthew Turk

CHEMISTRY AND COOLING



High temperature cooling rates at various metallicities.



One-zone model of gas collapse at various metallicities.

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

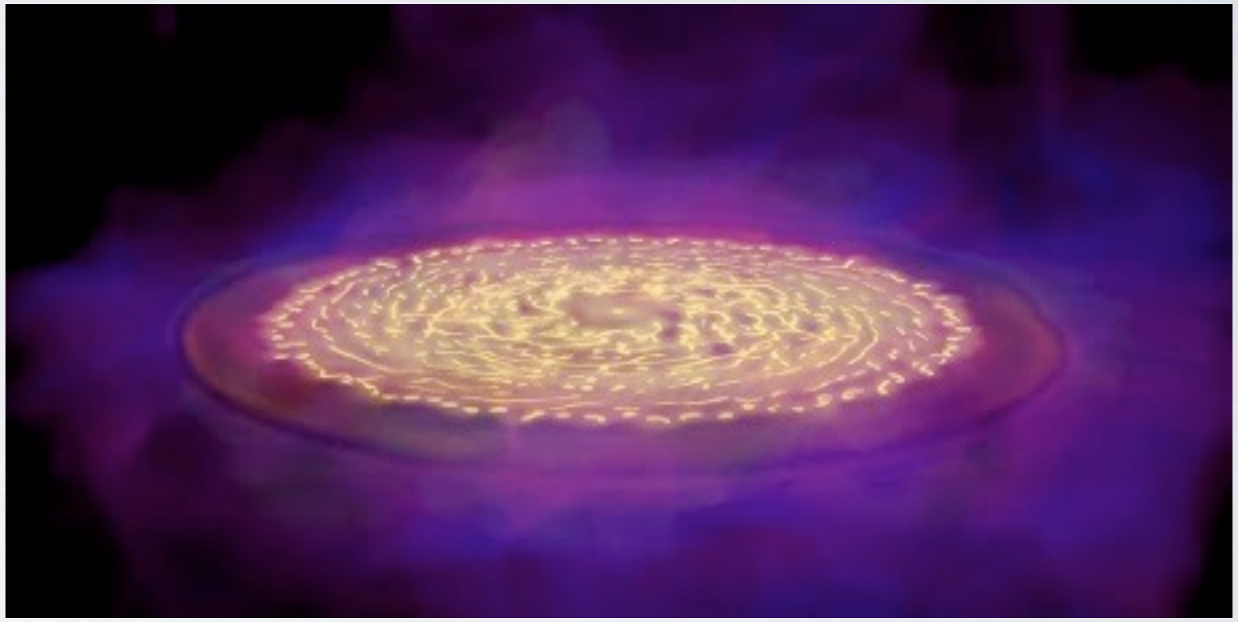
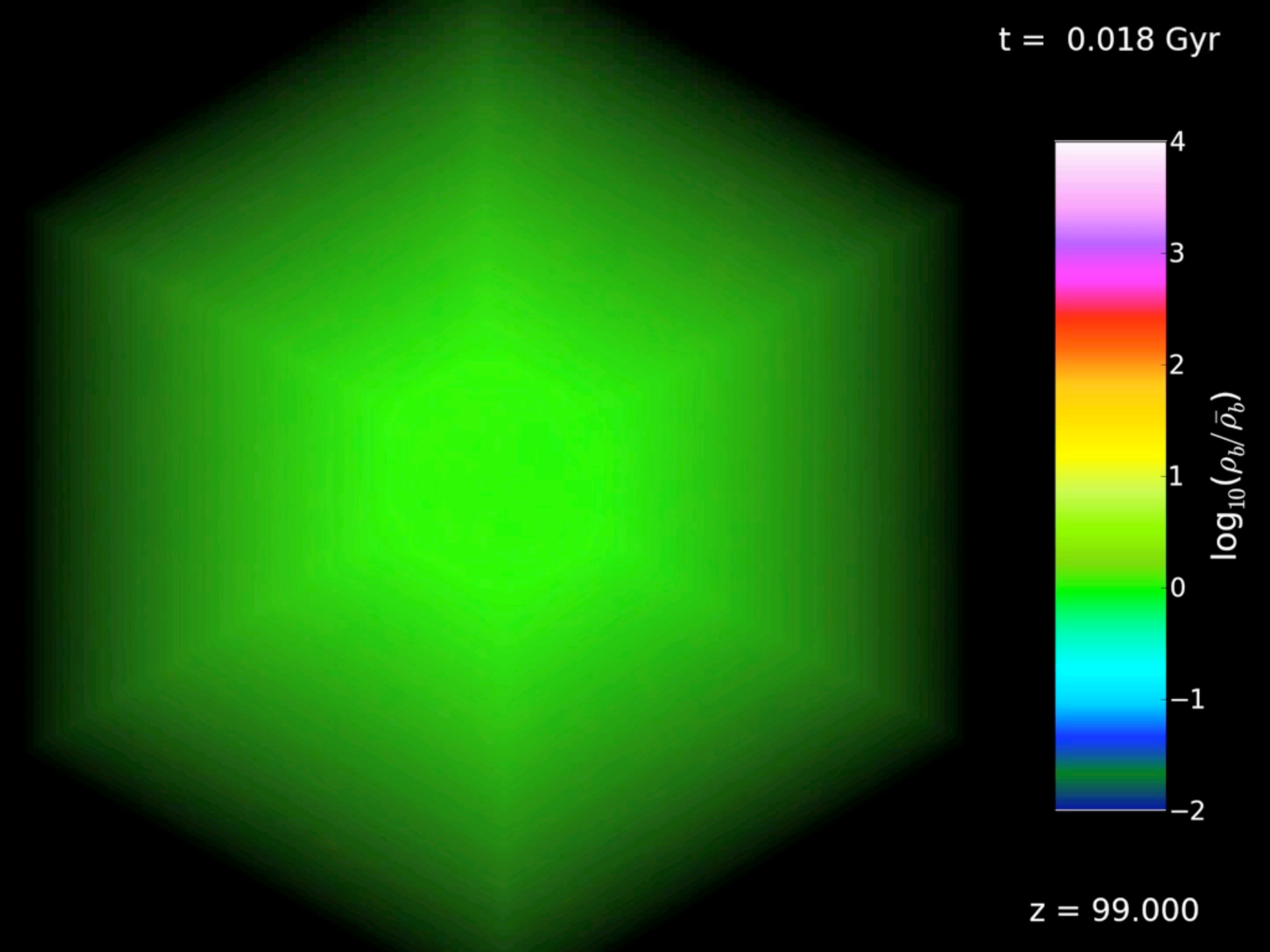


Image: Elizabeth Tasker

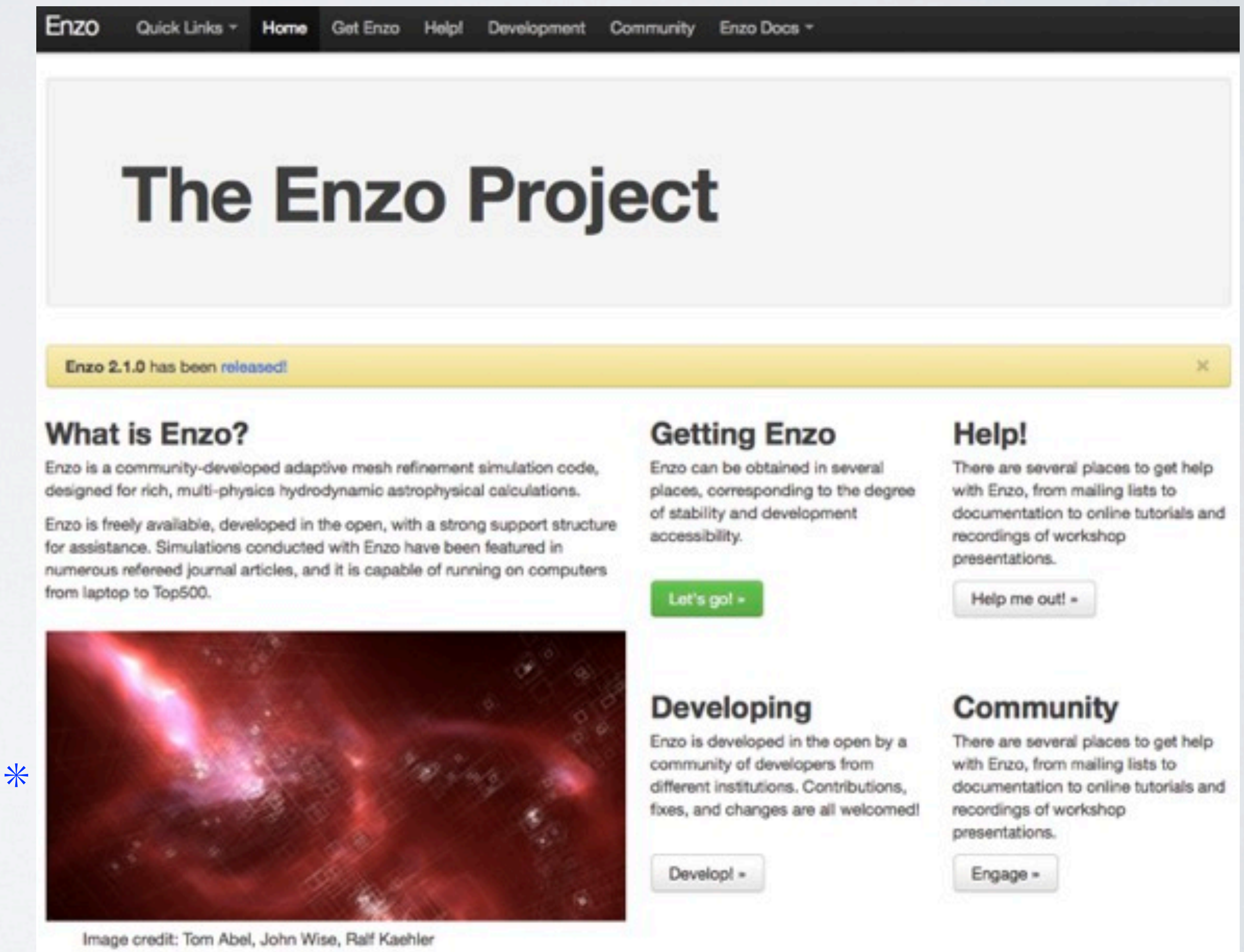


t = 0.018 Gyr
4
3
2
1
0
−1
−2
$\log_{10}(\rho_b/\bar{\rho}_b)$
z = 99.000

GETTING ENZO

enzo-project.org

- Go to enzo-project.org!
  - links to the source code
  - documentation
  - community resources
- Two major versions:
  - Stable: v2.1.1 (Oct 27, 2011)
    - ~1-2 major updates per year
    - periodic bug-fixes
  - Development *recommended*
    - ~1-2 updates per month
    - changes reviewed by developers

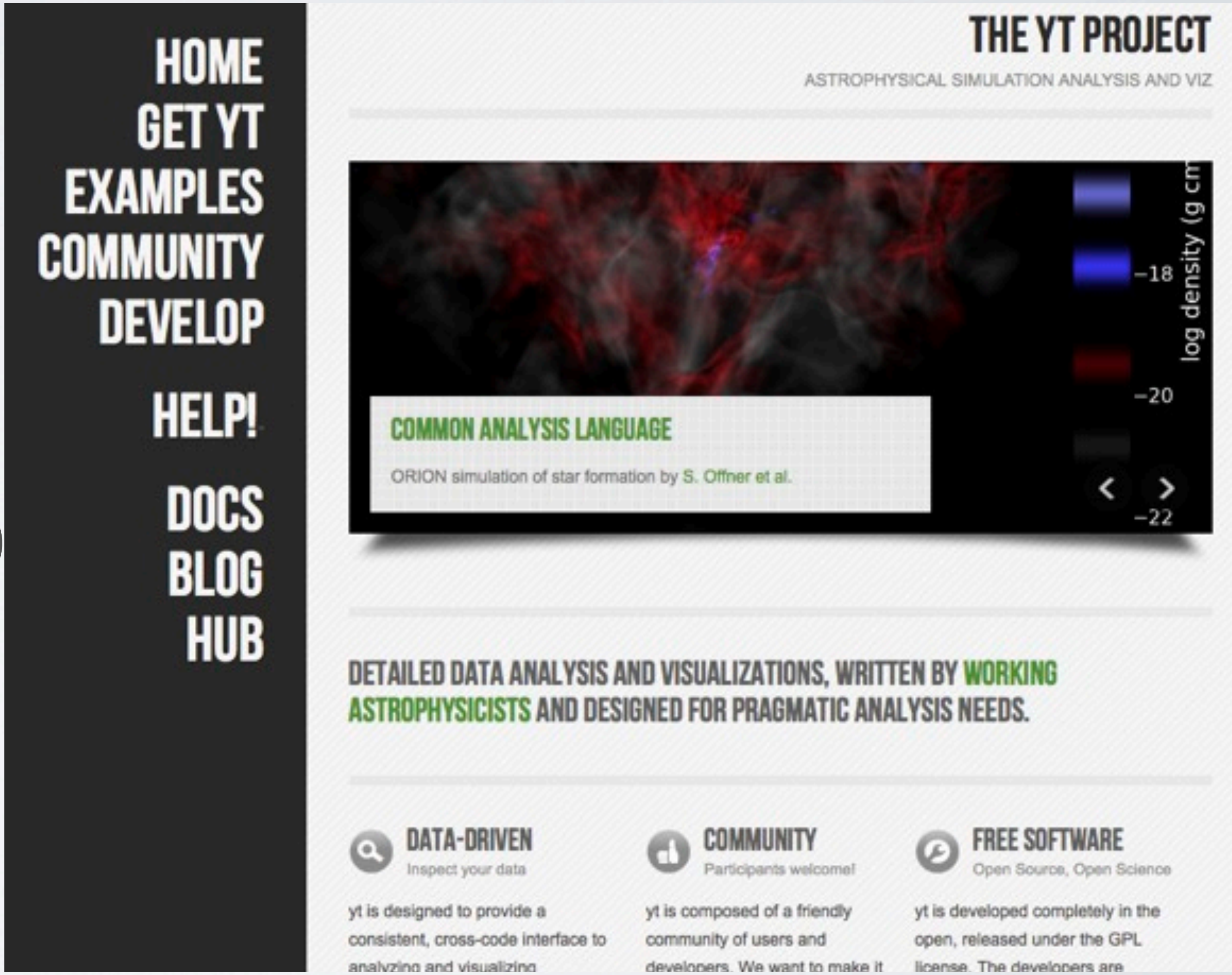


DEPENDENCIES

- hdf5 (Hierarchical Data Format) - version 1.8.x
- mpi (Message Passing Interface) - OpenMPI suggested
- Mercurial - version control system

RECOMMENDED METHOD

1. Go to yt-project.org
   - download and run install script
   - installs hdf5 and mercurial
2. Download and install OpenMPI (open-mpi.org)
   1. ./configure
   2. make
   3. make install



VERSION CONTROL
WITH MERCURIAL

- Distributed version control
  - no need for a central repository
  - changes can be pushed from any repository to any repository
  - merging changes from multiple branches is easy (at least easier)
- Mercurial tutorial: http://hginit.com

GETTING ENZO
WITH MERCURIAL

Check out a copy of Enzo (clone the repository):

```
hg clone https://bitbucket.org/enzo/enzo-dev
```

creates a directory on your computer called "enzo-dev"

Update your repository with the latest changes:

```
hg pull <source>
hg update
```

← pulls changes into the local repository

← updates the working copy with the latest changes

Add your new changes:

```
hg commit
hg push <destination>
```

← adds changes to the local repository

← pushes changes to another repository

COMPILING

```
cd enzo-dev
./configure
cd src/enzo
```

- Choose the make file that is right for your machine.
- Type "`ls Make.mach.*`" to see the available options.
- For Mac OS, choose "Make.mach.darwin".

COMPILING

- Edit Make.mach.darwin
- Change the following variables to the correct values:
  - LOCAL_MPI_INSTALL
  - LOCAL_HDF5_INSTALL
- Type:

```
make machine-darwin
make
```

- If everything is right, Enzo is now compiled!

COMPILE OPTIONS

- Enzo has many additional compile options.
- Type: `make show-config` to see the current settings.
- Type: `make help-config` for a description of each parameter.
- Example: `make opt-high` to compile with basic optimizations. Recommended!
- Enzo must be recompiled after options are changed.

EXTRA TIPS

- Custom make files can be stored the .enzo directory in your home directory.
- Compiler settings can be saved with:

```
make save-config-<keyword>
```

- Reload custom settings with:

```
make load-config-<keyword>
```

- Settings files saved in ~/.enzo/Make.settings.<keyword>

COMPILING ON CONIVAL

```
[guest13@Conival ~]$ ls
enzo-dev   mpiexamples   workshop
[guest13@Conival ~]$ cd enzo-dev/
[guest13@Conival enzo-dev]$ cd src/enzo
[guest13@Conival enzo]$ cp ~/workshop/Make.mach.conival .
[guest13@Conival enzo]$ cd ../..
[guest13@Conival enzo-dev]$ ./configure
Configure complete.
[guest13@Conival enzo-dev]$ cd src/enzo
[guest13@Conival enzo]$ make machine-conival

 *** Execute 'gmake clean' before rebuilding executables ***

   MACHINE: conival
   MACHINE-NAME: conival
[guest13@Conival enzo]$ make opt-high

 *** Execute 'gmake clean' before rebuilding executables ***

   CONFIG_OPT  [opt-{warn,debug,cudadebug,high,aggressive}]  : high

[guest13@Conival enzo]$ make

Updating DEPEND
make: [dep] Error 1 (ignored)
Compiling enzo.C
Compiling acml_st1.F
Compiling AdiabaticExpansionInitialize.C
Compiling AdjustRefineRegion.C
Compiling AdjustMustRefineParticlesRefineToLevel.C
Compiling AMRH5writer.C
Compiling AnalysisBaseClass.C
Compiling AnalysisBaseClass_HDF5Utils.C
```

- `cd enzo-dev/` — Move into enzo directory.
- `cd src/enzo` — Move into source directory.
- `cp ~/workshop/Make.mach.conival .` — Copy the make file into the current directory.
- `cd ../..` — Move back to the top directory.
- `./configure` — Run ./configure.
- `cd src/enzo` — Move back to the source directory.
- `make machine-conival` — Type this to select this make file for compiling.
- `make opt-high` — Configure to compile with high optimization.
- `make` — Compile the code.

RUNNING A SIMULATION

- Simulations are configured with a parameter file.
- Run a new simulation:

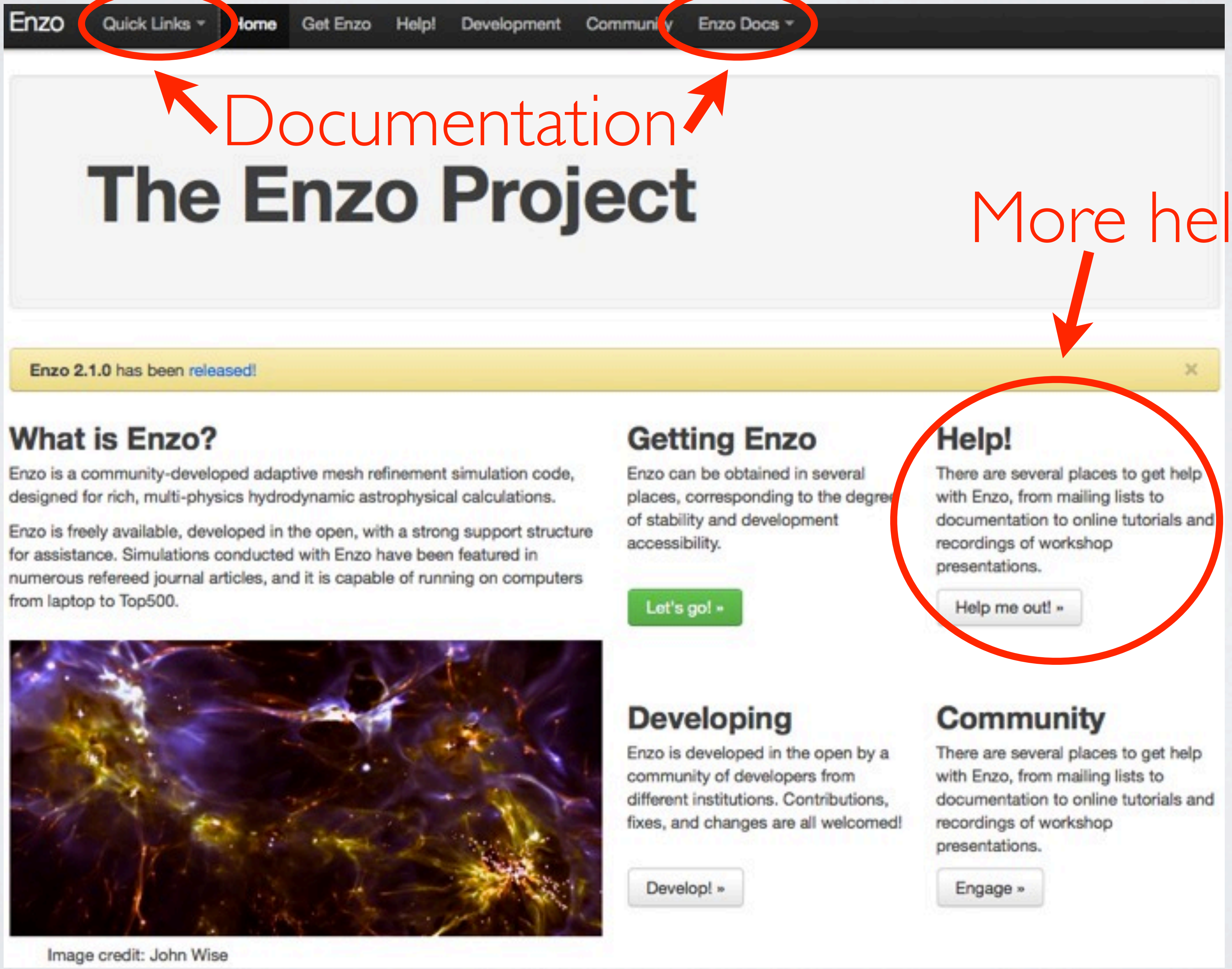
```
mpirun -np <#> ./enzo.exe -d <parameter_file>
```

- Restart a simulation:

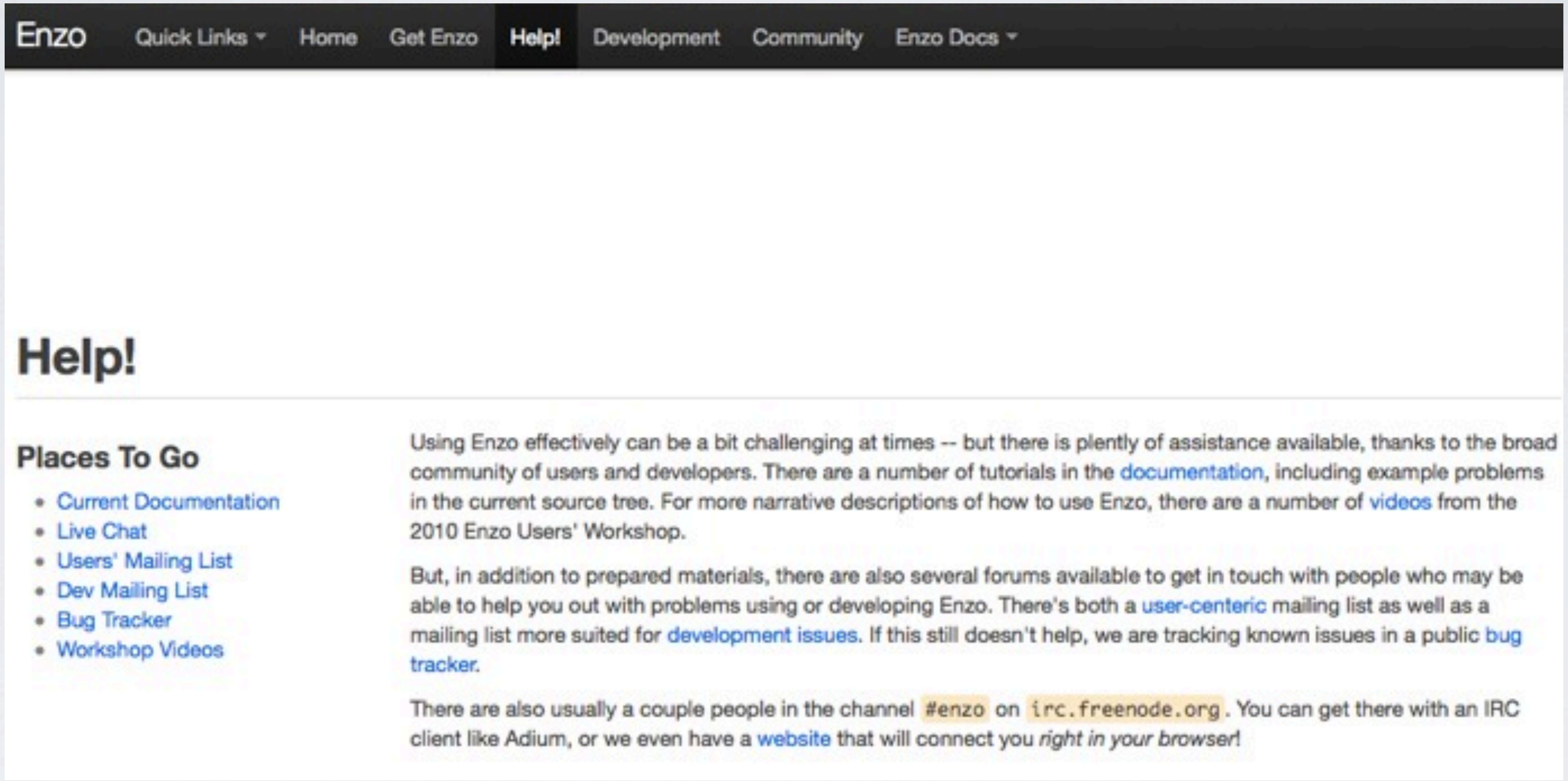
```
mpirun -np <#> ./enzo.exe -d -r <dataset>
```

- Many sample parameter files in enzo-dev/run

GETTING HELP

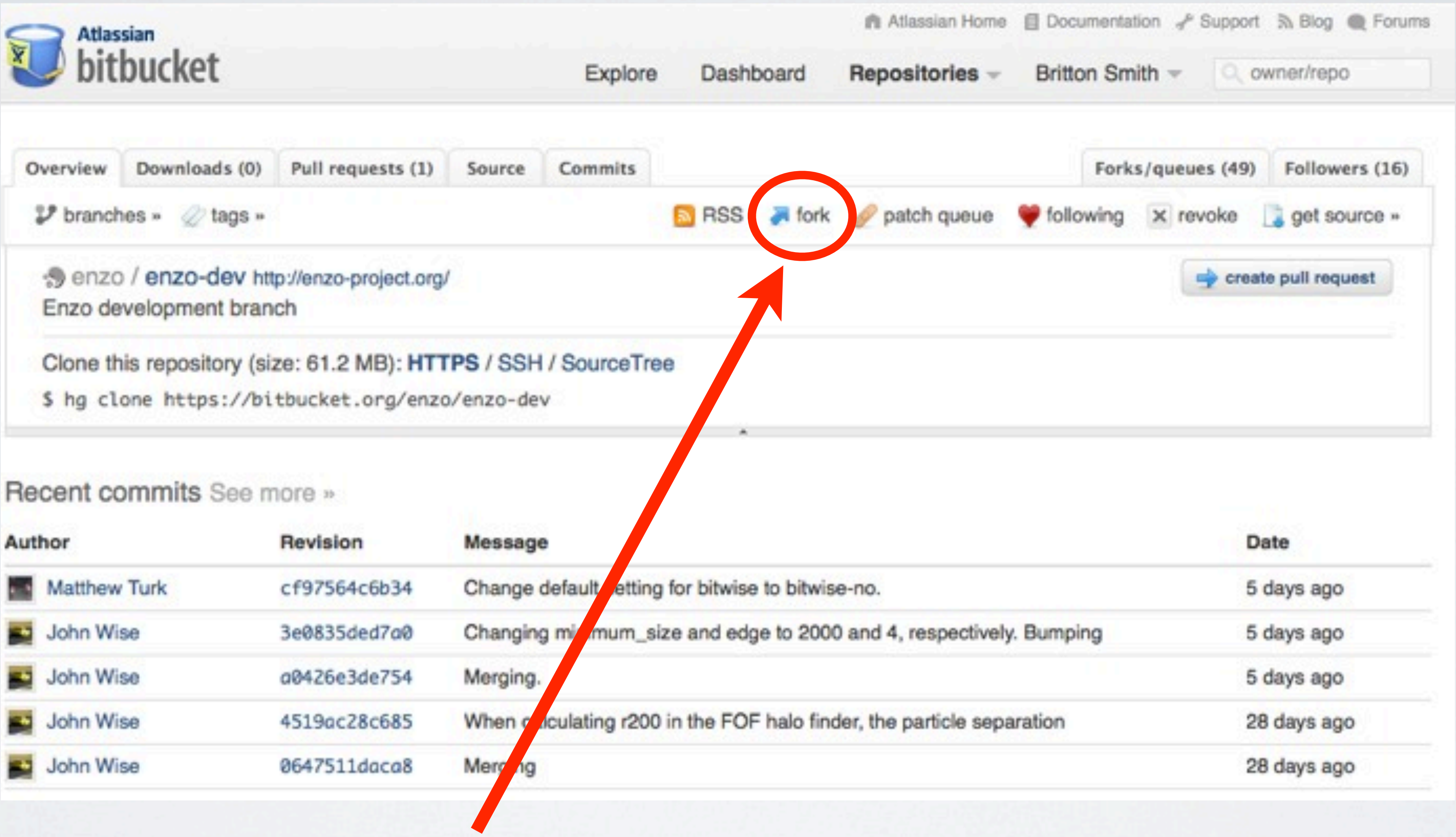


GETTING HELP



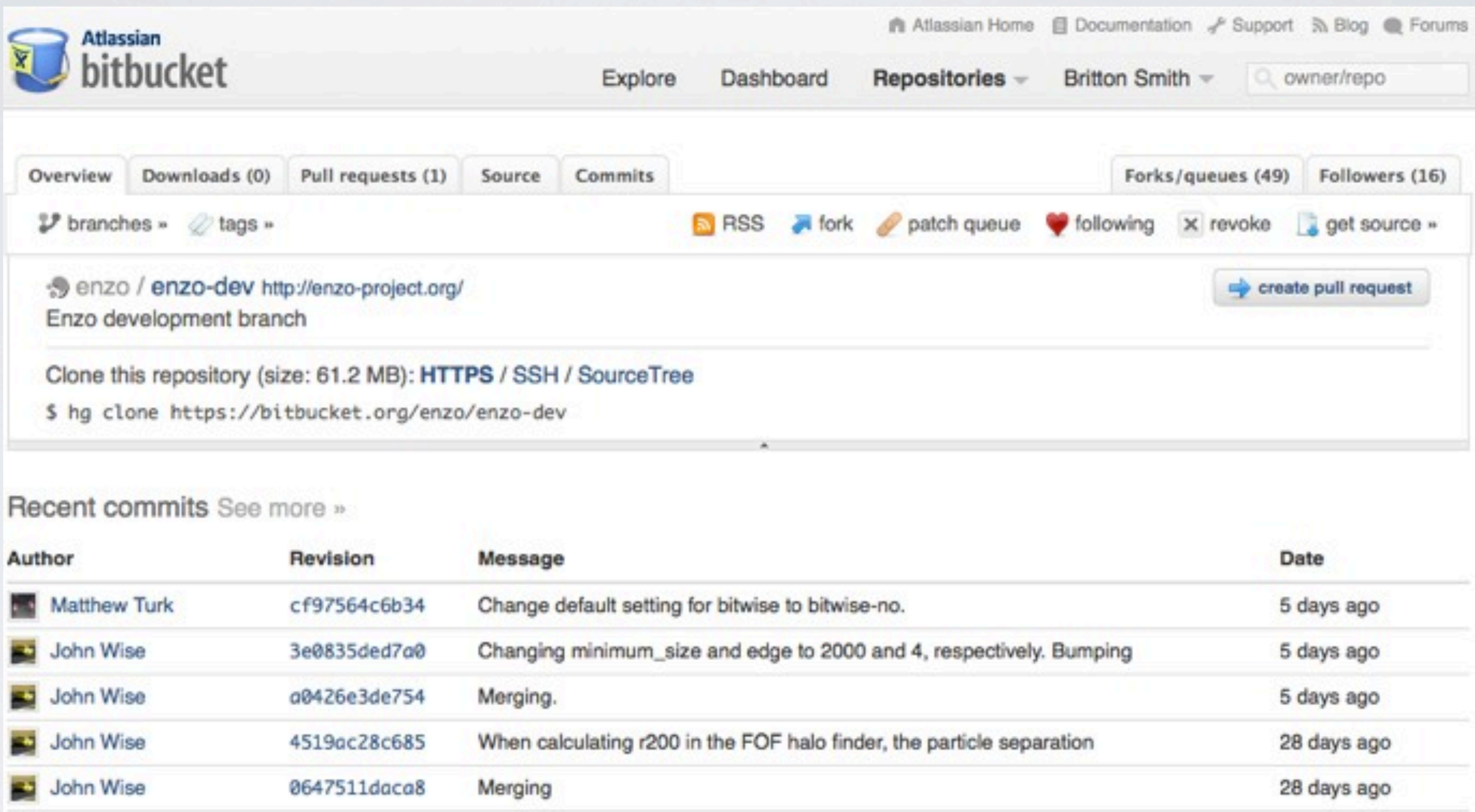
- mailing lists for users and developers
- IRC channel for live help
- workshop videos

DEVELOPMENT



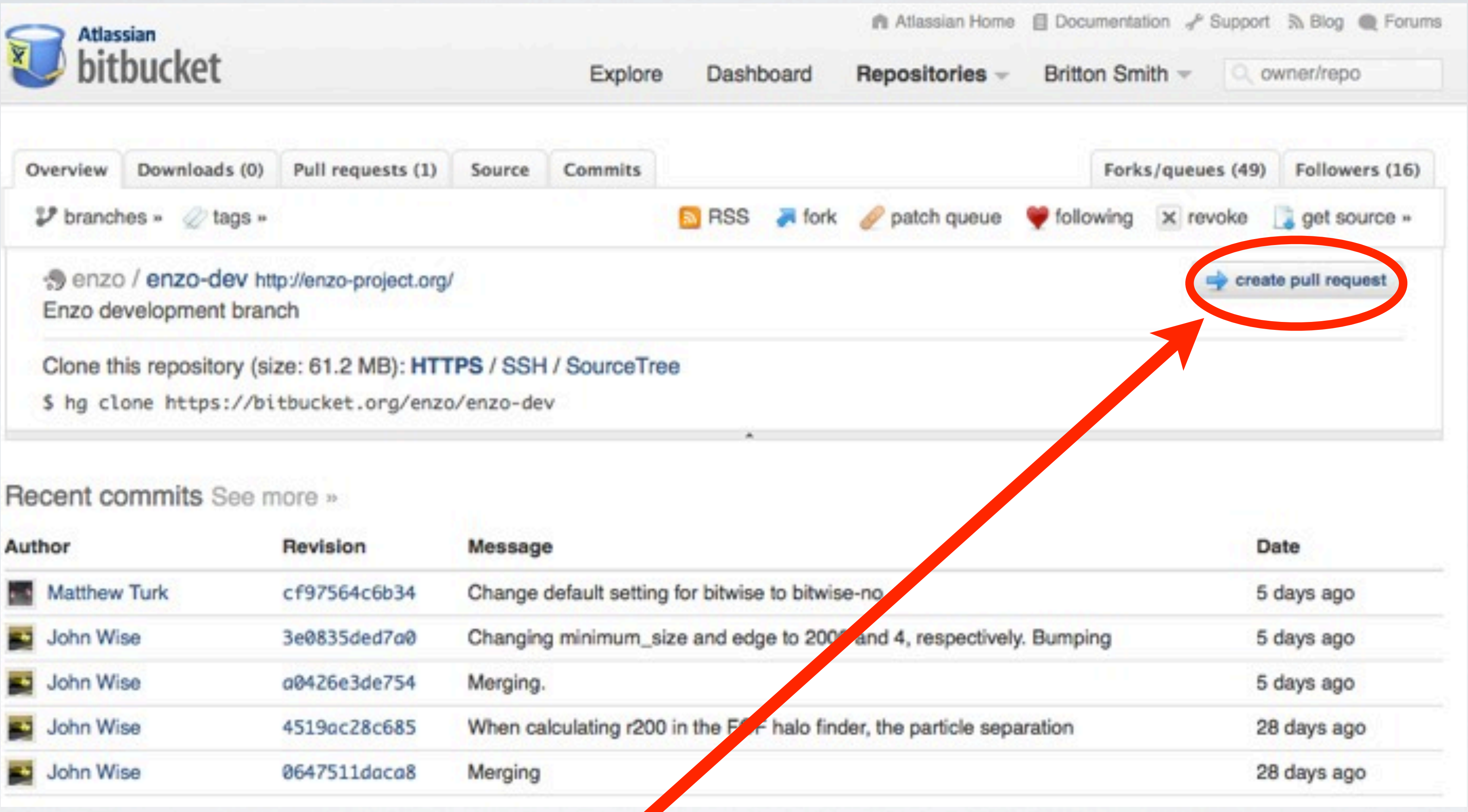
1. Create a fork of the main repository.

DEVELOPMENT



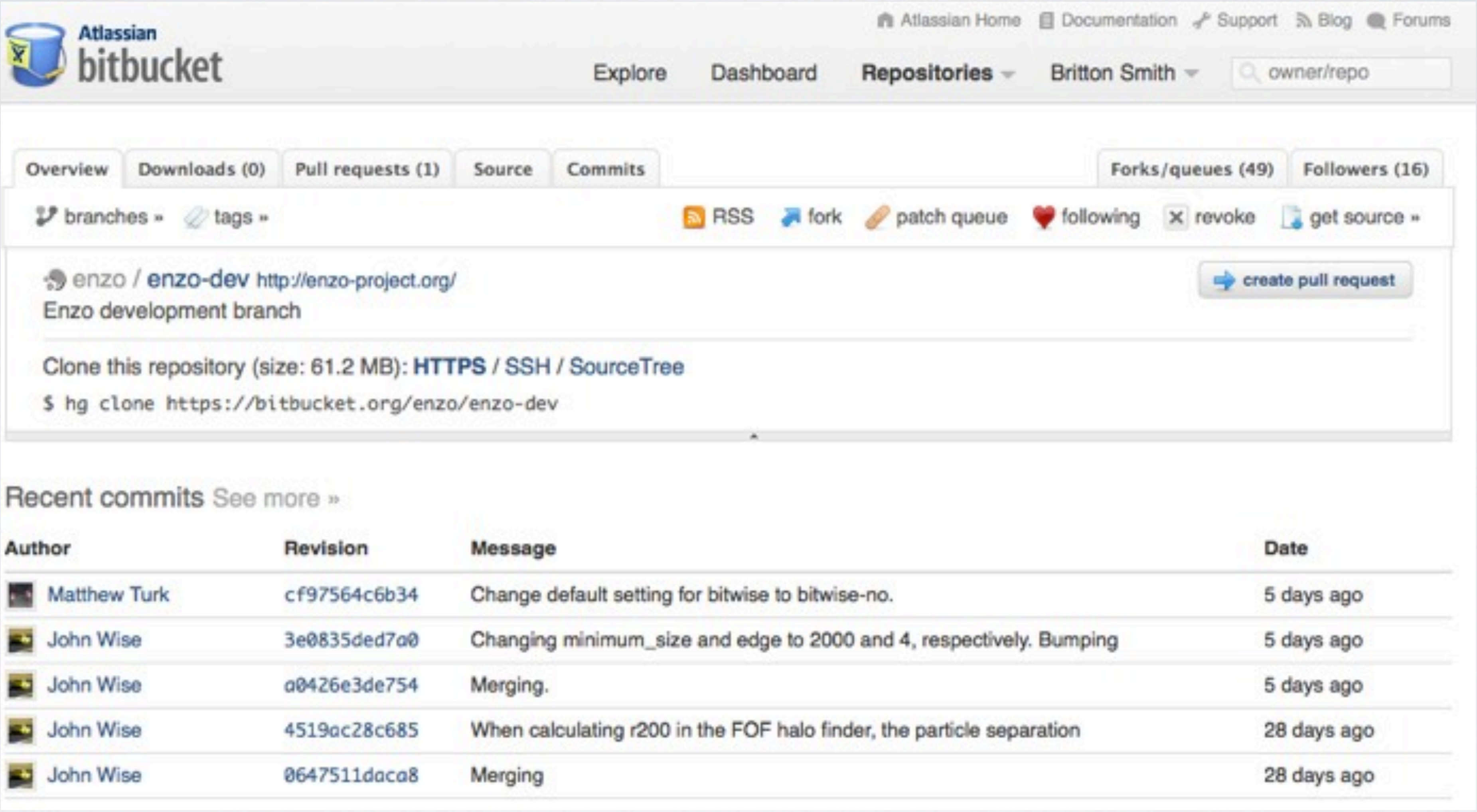
1. Create a fork of the main repository.
2. Clone your fork, commit changes, push them to your fork.

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.

DEVELOPERS

```
== DEVELOPERS ==

Many people have contributed to the development of Enzo -- here's just a short
list of the people who have recently contributed.

   * Greg Bryan       gbryan@astro.columbia.edu
   * Tom Abel         tabel@stanford.edu
   * Michael Norman   mlnorman@ucsd.edu
   * John Wise        jwise@physics.gatech.edu
   * Dan Reynolds     reynolds@smu.edu
   * Michael Kuhlen   mqk@astro.berkeley.edu
   * Matthew Turk     matthewturk@gmail.com
   * Brian O'Shea     oshea@msu.edu
   * Robert Harkness  harkness@sdsc.edu
   * Alexei Kritsuk    akritsuk@ucsd.edu
   * Elizabeth Tasker taskere@mcmaster.ca
   * Dave Collins     dcollins@physics.ucsd.edu
   * Britton Smith    brittonsmith@gmail.com
   * Elizabeth Harper-Clark h-clark@astro.utoronto.ca
   * Peng Wang        pengw@slac.stanford.edu
   * Fen Zhao         fenzhao@stanford.edu
   * James Bordner    jobordner@ucsd.edu
   * Pascal Paschos   ppaschos@minbari.ucsd.edu
   * Stephen Skory    sskory@physics.ucsd.edu
   * Rick Wagner      rwagner@physics.ucsd.edu
   * Renyue Cen       cen@astro.princeton.edu
   * Alex Razoumov    razoumov@gmail.com
   * Cameron Hummels chummels@astro.columbia.edu
   * JS Oishi         jsoishi@gmail.com
   * Christine Simpson csimpson@astro.columbia.edu
   * Samuel Skillman samskillman@gmail.com
```

DEVELOPERS

== DEVELOPERS ==

Many people have contributed to the development of Enzo -- here's just a short
list of the people who have recently contributed.

```
   * Greg Bryan      gbryan@astro.columbia.edu
   * Tom Abel        tabel@stanford.edu
   * Michael Norman  mlnorman@ucsd.edu
   * John Wise       jwise@physics.gatech.edu
   * Dan Reynolds    reynolds@smu.edu
   * Michael Kuhlen  mqk@astro.berkeley.edu
   * Matthew Turk    matthewturk@gmail.com
   * Brian O'Shea    oshea@msu.edu
   * Robert Harkness harkness@sdsc.edu
   * Alexei Kritsuk   akritsuk@ucsd.edu
   * Elizabeth Tasker taskere@mcmaster.ca
   * Dave Collins    dcollins@physics.ucsd.edu
   * Britton Smith   brittonsmith@gmail.com
   * Elizabeth Harper-Clark h-clark@astro.utoronto.ca
   * Peng Wang       pengw@slac.stanford.edu
   * Fen Zhao        fenzhao@stanford.edu
   * James Bordner   jobordner@ucsd.edu
   * Pascal Paschos  ppaschos@minbari.ucsd.edu
   * Stephen Skory   sskory@physics.ucsd.edu
   * Rick Wagner     rwagner@physics.ucsd.edu
   * Renyue Cen      cen@astro.princeton.edu
   * Alex Razoumov   razoumov@gmail.com
   * Cameron Hummels chummels@astro.columbia.edu
   * JS Oishi        jsoishi@gmail.com
   * Christine Simpson csimpson@astro.columbia.edu
   * Samuel Skillman samskillman@gmail.com
```

- You?

THANK YOU