FIRST STEPS WITH ENZO

Britton Smith

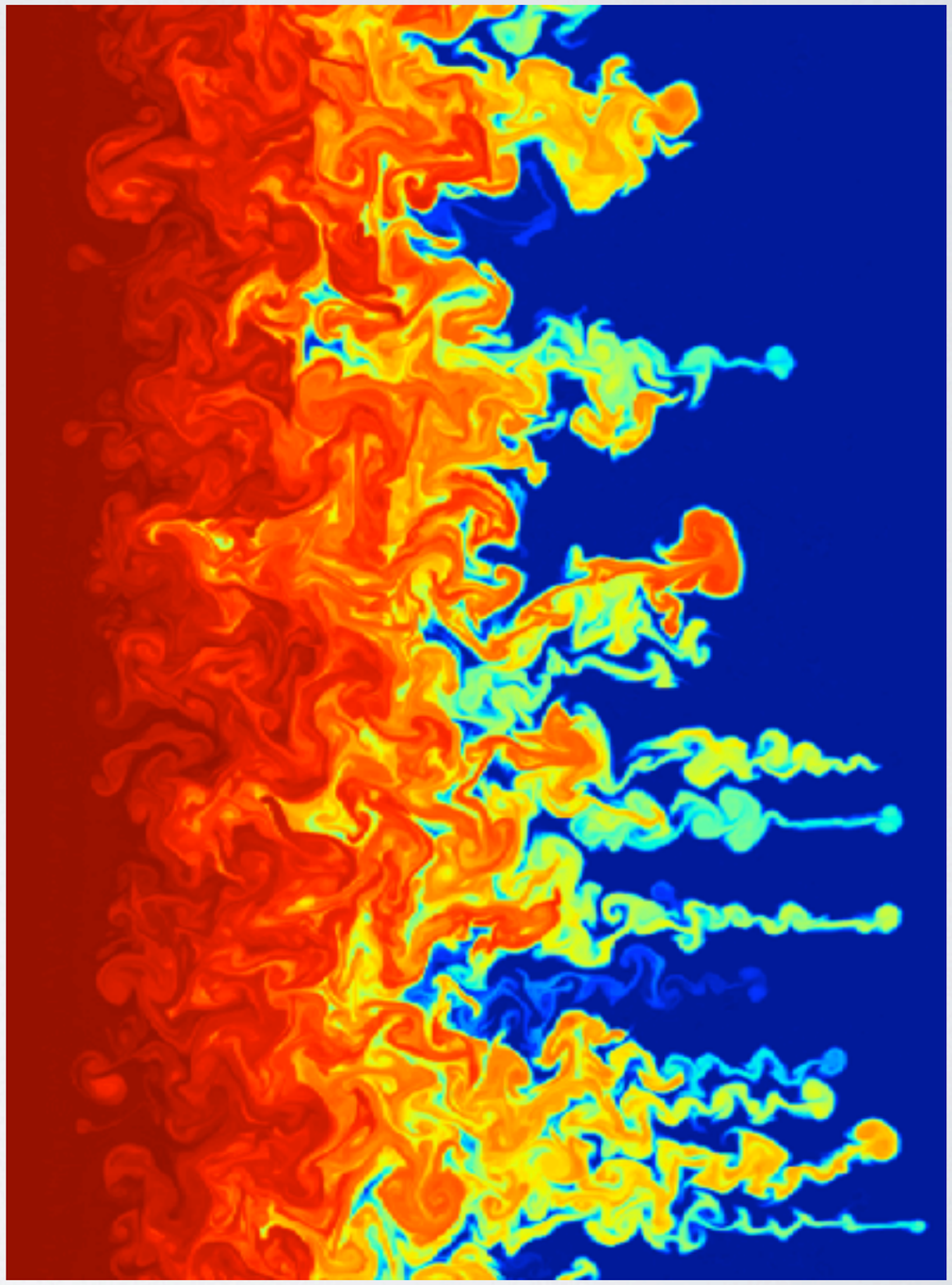
GOALS

I. Download

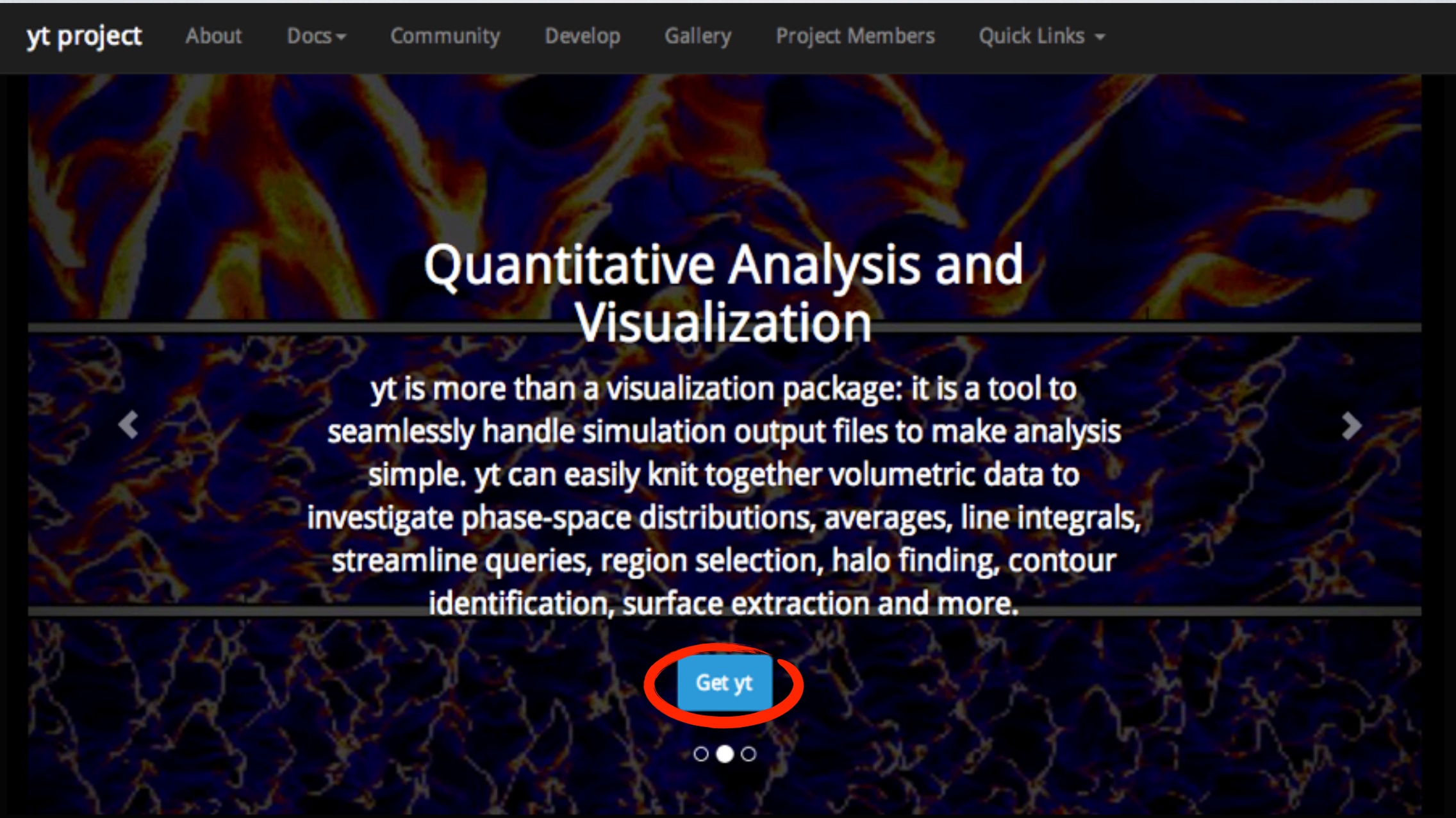
II. Understand the source

III. Compile

IV. Run simple problems



Install yt from yt-project.org

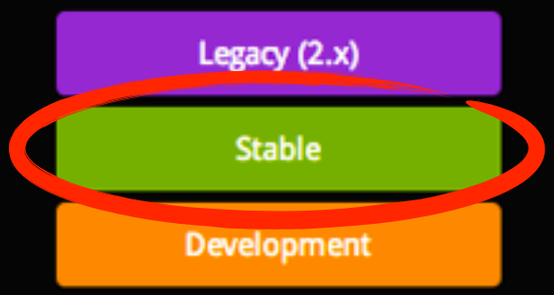


Install yt from yt-project.org

yt project | About | Docs | Community | Develop | Gallery | Project Members | Quick Links

can be upgraded and operated independently of the host operating system.

Usually getting yt is as simple as running the installation script. Simply download the legacy, stable, or development version of the install script and run it. You can do this using **wget** or **curl**, or even just right click and choose **Save As**. Carefully read the instructions the script prints to your terminal since there might be special instructions for your operating system.



Once you've downloaded it, just run:

```
$ bash install_script.sh
$ source YT_DEST/bin/activate
```

```
$ conda install yt
```

Get yt: from source.

If you are comfortable installing Python packages and have a build environment set up, you can install yt via **pip**:

```
$ pip install yt
```

If you would like to install the development version of yt, first clone the repository:

```
$ hg clone
https://bitbucket.org/yt_analysis/yt
```

...d in the

This will give us mercurial and hdf5.

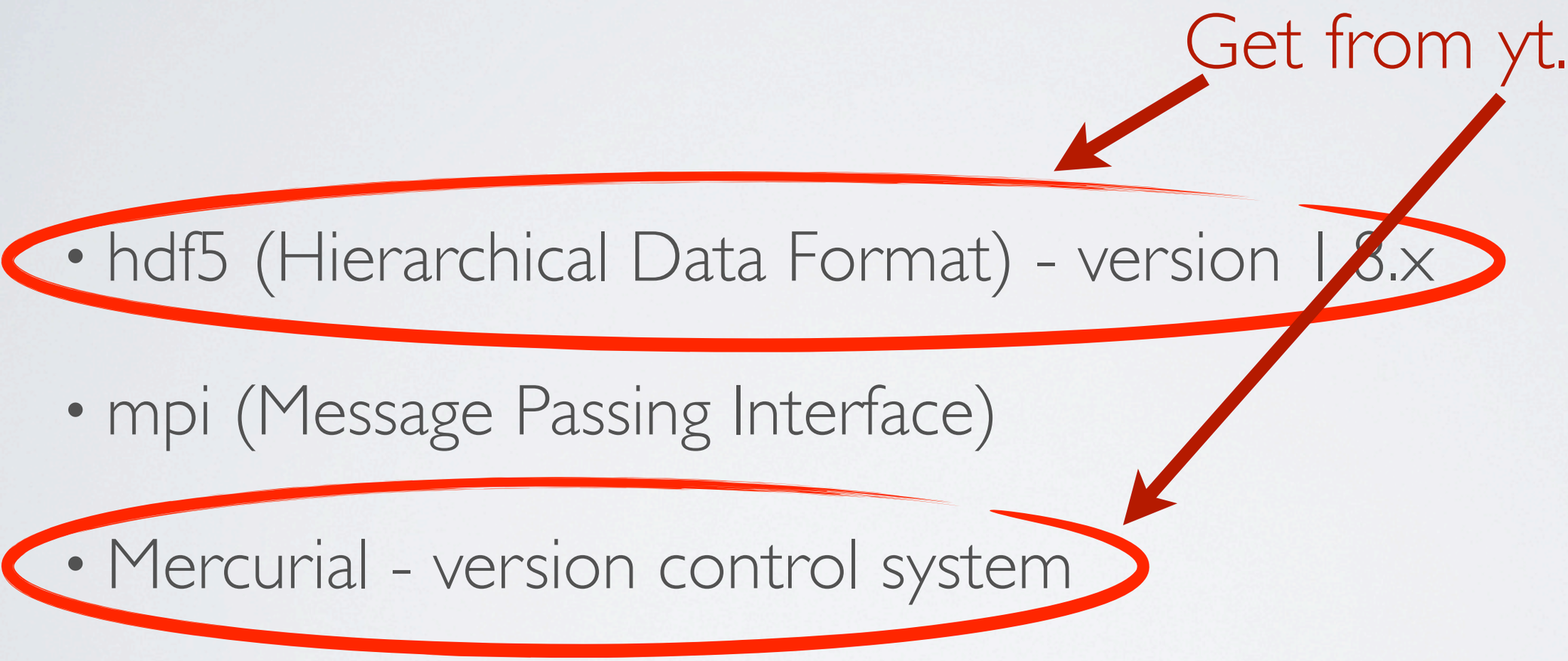
Then do the following:

```
$ python setup.py develop
```

DEPENDENCIES

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

DEPENDENCIES



DEPENDENCIES

- hdf5 (Hierarchical Data Format) - version 1.8.x
- mpi (Message Passing Interface)
- Mercurial - vers

Download and install OpenMPI (open-mpi.org)

1. ./configure
2. make
3. make install

Go to enzo-project.org

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

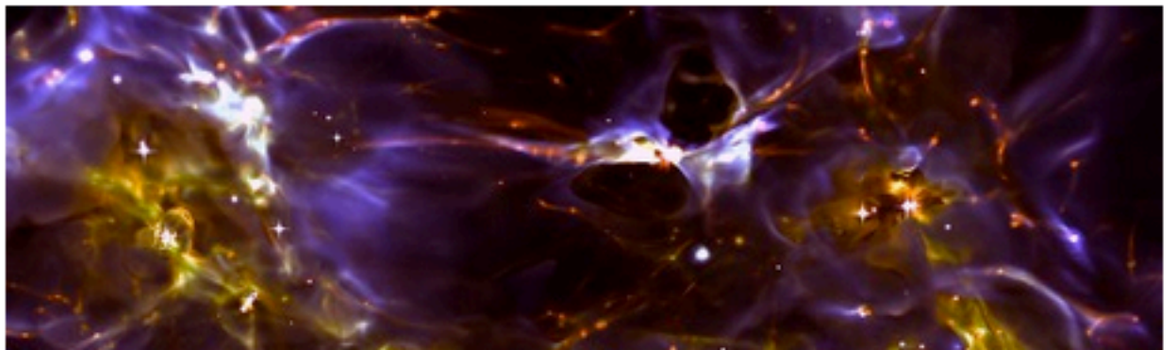
The Enzo Project

Aug 8 2013: Enzo 2.4 has been released. View the Release Notes!

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.



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,

Community

There are several places to get help with Enzo, from mailing lists to documentation to online tutorials and

GETTING ENZO

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

Getting Enzo

Places To Go

- Bitbucket: Stable Version
- Bitbucket: Development Version
- Tarfile Downloads
- Release Notes
- Enzo Boot Camp

The easiest way to get up and running with Enzo is to follow our Enzo Boot Camp, which walks you through installation, running test problems, and making some simple images.

Get up and running! »

Enzo is provided through several channels: a public repository of code that is under active development, as well as a stable channel that is carefully curated and that corresponds to releases. Have a look at the Release Notes, if you're interested in what features have recently made it into the stable version. Tarfiles of stable releases are also available, but are discouraged. If you want to use the development version, it's probably a good idea to check out the development guide.

The simplest way to get a copy of the current stable source code is to clone the repository using Mercurial:

```
$ hg clone https://bitbucket.org/enzo/enzo-stable
```

Development version

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

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

WHAT'S INSIDE?

```
grackle:enzo_workshop[14] cd enzo-dev/
grackle:enzo-dev[15] ls
CHANGELOG README    configure input     run
LICENSE   bin       doc       patch     src
grackle:enzo-dev[16]
```

WHAT'S INSIDE?

Documentation in
doc/manual/

```
grackle:enzo_workshop[14] cd enzo-dev/
grackle:enzo-dev[15] ls
CHANGELOG README    configure input     run
LICENSE   bin       doc       patch     src
grackle:enzo-dev[16]
```

WHAT'S INSIDE?

Documentation in
doc/manual/

```
grackle:enzo-dev[32] ls
CHANGELOG README    configure input      run
LICENSE   bin       doc       patch     src
grackle:enzo-dev[33] cd doc/manual/
grackle:manual[34] ls
Makefile README   build    source
grackle:manual[35]
```

WHAT'S INSIDE?

Build the documentation.

First do:"pip install sphinx"

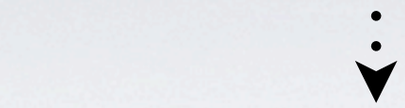
```
grackle:enzo-dev[32] ls
CHANGELOG README    configure input      run
LICENSE   bin       doc       patch     src
grackle:enzo-dev[33] cd doc/manual/
grackle:manual[34] ls
Makefile README   build    source
grackle:manual[35]
```

WHAT'S INSIDE?

Build the documentation.

First do: "pip install sphinx"

```
grackle:enzo-dev[33] cd doc/manual/
grackle:manual[34] ls
Makefile README   build     source
grackle:manual[35] make html
sphinx-build -b html -d build/doctrees   source build/html
Running Sphinx v1.2b1
```

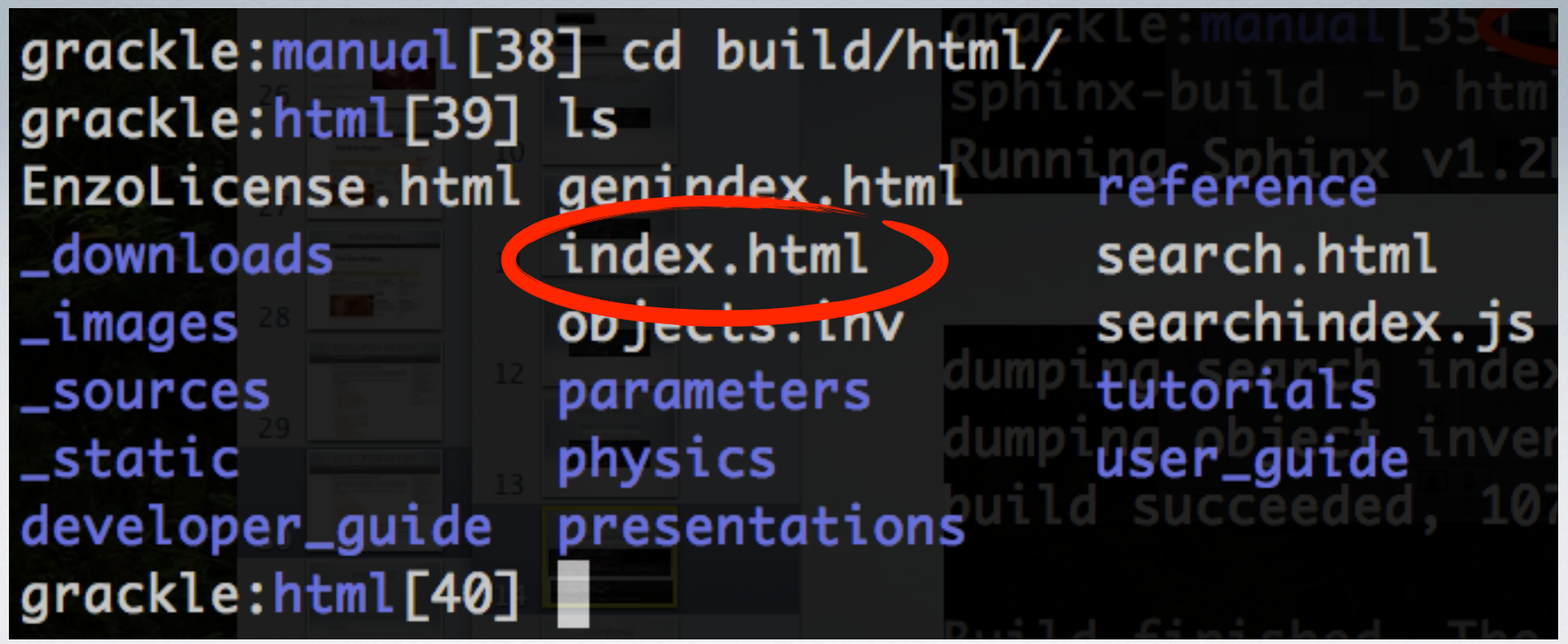


```
dumping search index... done
dumping object inventory... done
build succeeded, 107 warnings.

Build finished. The HTML pages are in build/html.
grackle:manual[36]
```

WHAT'S INSIDE?

The documentation
is now built just like
on the internet.



WHAT'S INSIDE?

Cooling tables in input/

```
grackle:enzo-dev[44] ls
CHANGELOG README    configure input    run
LICENSE   bin       doc       patch    src
grackle:enzo-dev[45] ls input/
ATOMIC.DAT                      make_Zcool_table.pro
LW_J21.in                       metal_cool.dat
TREECOOL                        metal_cool.dat_z=15
cool_rates.in                   metal_cool_pop3.dat
cool_rates.in_300K              metal_cool_ratios.dat
cooling.pro                     metal_cooling.pro
cosmic_ray.dat                  restart.sh
enzo_parameter_conflicts.txt    resubmit.sh
hm12_photorates.dat             zcool_sd93.dat
lookup_metal0.3.data
grackle:enzo-dev[46]
```

WHAT'S INSIDE?

Simulation parameter files in run/

```
grackle:enzo-dev[47] ls
CHANGELOG README    configure input  run
LICENSE   bin       doc       patch
grackle:enzo-dev[48] ls run/
Cooling                 RadiationTransport
Cosmology               RadiationTransportFLD
CosmologySimulation     index.html
DrivenTurbulence3D      run_templates
FLD                     test_makespreadsheet.py
GravitySolver           test_runner.py
Hydro                   test_type.py.template
MHD
grackle:enzo-dev[49]
```

Explore further!

WHAT'S INSIDE?

Enzo source in
src/enzo/

```
grackle:enzo-dev[59] ls
CHANGELOG README    configure input      run
LICENSE   bin       doc       patch     src
grackle:enzo-dev[60] ls src/enzo
```

```
Grid_FastSiblingLocatorFindSiblings.C
Grid_FinalizeRadiationFields.C
Grid_FindAllStarParticles.C
Grid_FindMassiveParticles.C
Grid_FindMinimumParticleMass.C
Grid_FindNewStarParticles.C
Grid_FindPhotonNewGrid.C
Grid_FindShocks.C
Grid_FinishFFT.C
```

COMPILING

1. run configure script

This will prepare the environment.

```
grackle:enzo-dev[81] ls
CHANGELOG README    configure input     run
LICENSE   bin       doc       patch     src
grackle:enzo-dev[82] ./configure
Configure complete.
```

COMPILING

1. run configure script
2. go into src/enzo

```
grackle:enzo-dev[81] ls
CHANGELOG README    configure input     run
LICENSE   bin       doc       patch     src
grackle:enzo-dev[82] ./configure
Configure complete.
grackle:enzo-dev[83] cd src/enzo
```

COMPILING

1. run configure script
2. go into src/enzo
3. find your make file

```
grackle:enzo-dev[81] ls
CHANGELOG README    configure input     run
LICENSE   bin       doc       patch     src
grackle:enzo-dev[82] ./configure
Configure complete.
grackle:enzo-dev[83] cd src/enzo
grackle:enzo[84] ls Make.mach.*
Make.mach.arizona                 Make.mach.nics-kraken-gnu-yt
Make.mach.darwin                  Make.mach.nics-nautilus
Make.mach.gtech-pace              Make.mach.orange
Make.mach.hotfoot-condor          Make.mach.ornl-jaguar-pgi
Make.mach.kolob                   Make.mach.scinet
```

for Macs, use Make.mach.darwin

COMPILING

1. run configure script
2. go into src/enzo
3. find your make file

   Make.mach.conival ← on Conival

```
grackle:enzo-dev[81] ls
CHANGELOG README     configure input      run
LICENSE   bin        doc       patch      src
grackle:enzo-dev[82] ./configure
Configure complete.
grackle:enzo-dev[83] cd src/enzo
grackle:enzo[84] ls Make.mach.*
Make.mach.arizona                Make.mach.nics-kraken-gnu-yt
Make.mach.darwin                 Make.mach.nics-nautilus
Make.mach.gtech-pace             Make.mach.orange
Make.mach.hotfoot-condor         Make.mach.ornl-jaguar-pgi
Make.mach.kolob                  Make.mach.scinet
```

for Macs, use Make.mach.darwin

COMPILING

4. Edit LOCAL_PACKAGES to point to your yt installation.

```
File Edit Options Buffers Tools Help
# Install paths (local variables)
#-------------------------------------------------------------------

LOCAL_PACKAGES   = /Users/britton/Desktop/enzo_workshop/yt-x86_64

# This will not work on OSX Lion or newer.  You may wany to try installin
# openmpi via macports.
LOCAL_MPI_INSTALL     = /usr/local
LOCAL_FC_INSTALL      = /usr/local
LOCAL_HDF5_INSTALL    = $(YT_DEST)
LOCAL_SZIP_INSTALL    = $(LOCAL_PACKAGES)
LOCAL_HYPRE_INSTALL   = $(HOME)
LOCAL_PYTHON_INSTALL = $(YT_DEST)

-uu-:---F1  Make.mach.darwin    24% L38     (Fundamental)--------------------
```

COMPILING

5. Type "make machine-<your machine>"

```
[guest03@Conival enzo]$ make machine-conival

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

   MACHINE: conival
   MACHINE-NAME: conival

[guest03@Conival enzo]$
```

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.

COMPILING

6. compile!

```
grackle:enzo[95] make

Updating DEPEND
Compiling enzo.C
Compiling acml_st1.F
Compiling AdiabaticExpansionInitialize.C
Compiling AdjustRefineRegion.C
Compiling AdjustMustRefineParticlesRefineToLevel.C
```

```
Linking enzo executable. Type  cat out.compile  in case it fails.
Success!
Compiled enzo from
Mercurial Branch   week-of-code
Mercurial Revision 4e0e0267f3b0+
grackle:enzo[101]
```

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>

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

RUN A SIMULATION

```
grackle:enzo-dev[125] ls
CHANGELOG README    configure input     run
LICENSE   bin       doc       patch     src
grackle:enzo-dev[126] cd run/Hydro/Hydro-3D/CollapseTestNonCosmological/
grackle:CollapseTestNonCosmological[127] ls
CollapseTestNonCosmological.enzo     notes.txt
CollapseTestNonCosmological.enzotest plot.py
grackle:CollapseTestNonCosmological[128]
```

RUN A SIMULATION

Choose units for the scale of your simulation.

```
#
#  units
#
DensityUnits          = 1.673e-20     // 10^4 g cm^-3
LengthUnits           = 3.0857e+18    // 1 pc in cm
TimeUnits             = 3.1557e+11    // 10^4 yrs
GravitationalConstant = 1.39698e-3    // 4*pi*G_{cgs}*DensityUnits*TimeUnits^2
```

RUN A SIMULATION

Run it!

```
mpirun -np 2 ./enzo.exe -d CollapseTestNonCosmological.enzo
```

PROGRESS METER

```
grackle:enzo_workshop[171] ls
CollapseTestNonCosmological enzo-dev
grackle:enzo_workshop[172] cp enzo-dev/bin/np CollapseTestNonCosmological/
grackle:enzo_workshop[173]
```

pipe output to estd.out

```
mpirun -np 2 ./enzo.exe -d CollapseTestNonCosmological.enzo >& estd.out &
grackle:CollapseTestNonCosmological[180] ./np
```

```
+-------------------------------------- . ----------------------------------------+
| Wed Oct 16 09:43:16 2013                    Status: 3.886% complete.             |
+------------------- Time -------------------+ +--------------- Output ----------------+
|  Initial  |  Current  |   Final   | Units | | |      |   Time    |  Name  | Completed |
+-----------+-----------+-----------+-------+ +------+-----------+--------+-----------|
| 2.000e-01 | 4.642e-01 | 7.000e+00 | code  | | Last | 4.000e-01 | DD0004 |  -------  |
| 2.000e+03 | 4.642e+03 | 7.000e+04 | years | | Next | 5.000e-01 | DD0005 | 64.22530% |
+-----------------------------------------+ +-----------------------------------------+
+-------------------------- Hierarchy ---------------------------+
| L | Grids |  Volume   |    dt     |  Sub  | Completed | Iter |R|
+---+-------+-----------+-----------+-------+-----------+------+-+
| 0 |     2 | 1.000e+00 | 1.000e-01 | 1.000 | 1.0000000 |    5 | |
| 1 |     2 | 1.250e-01 | 8.654e-02 | 0.865 | 0.8654120 |   10 | |
| 2 |     2 | 5.273e-02 | 4.362e-02 | 1.000 | 0.8654130 |   15 | |
```