PyGadget

A Framework for Analyzing GADGET Simulation Data in Pandas

PyGadget is a python library for reading GADGET-HDF5 files into a pandas DataFrame.

Pandas is the de-facto standard for data analysis in python, with broad support across the scientific python ecosystem.

- If we can painlessly get our data into pandas, we can more effectively leverage the rest of the tools in the ecosystem.

PyGadget is *NOT* a full-fledged astrophysical data analysis package like `yt` or `pynbody`.

A little background...

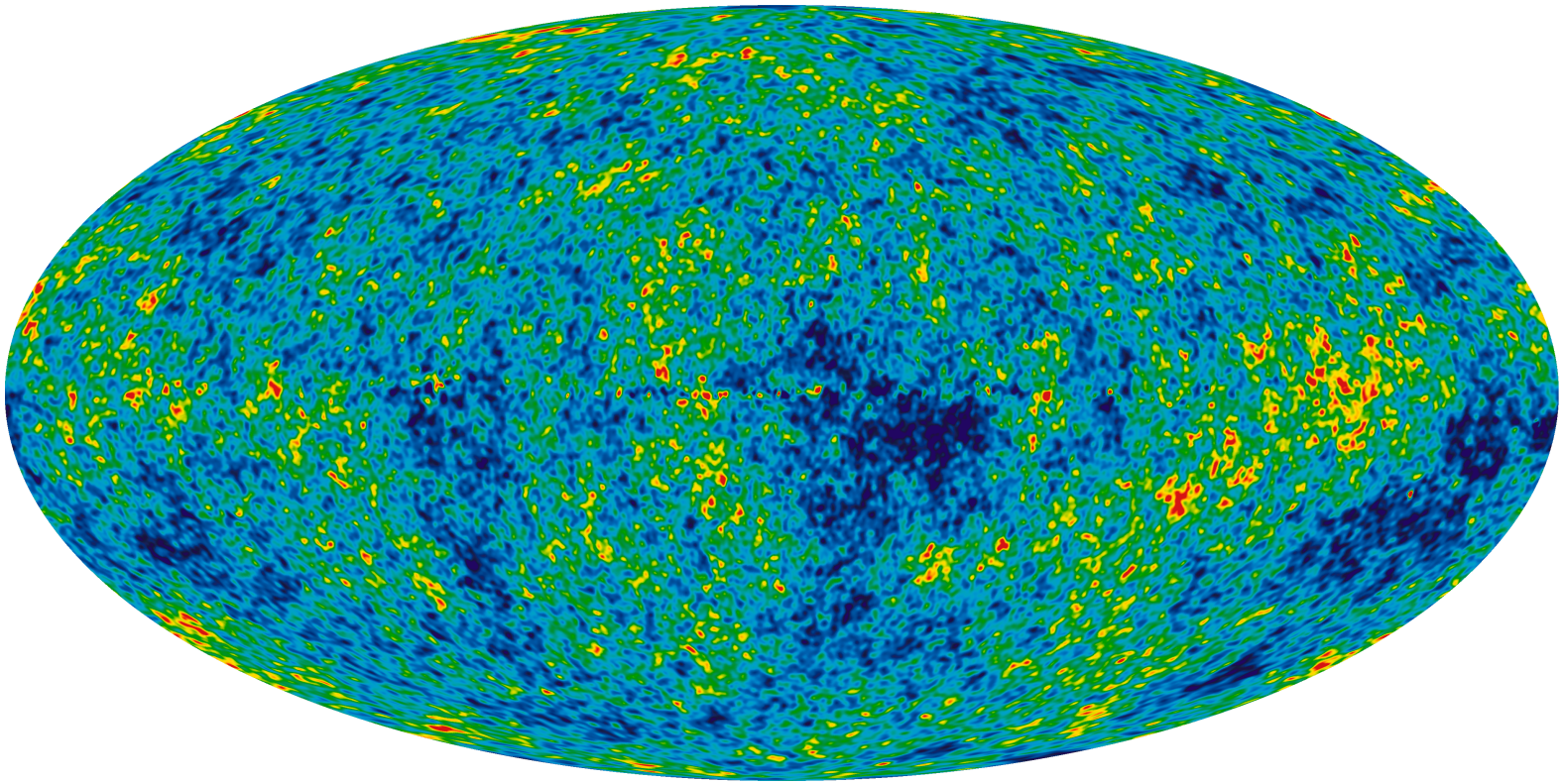
My ultimate goal is to understand how the first galaxies form.

The first stars represent a bottleneck of sorts in the evolution of the Universe.

- Their properties are sensitive to the initial conditions of the Universe.
- They have an outsized impact on everything that happens after them.

This is a very hard problem covering (literally) an astronomical range of physical scales.

How do we get from this:



To this?

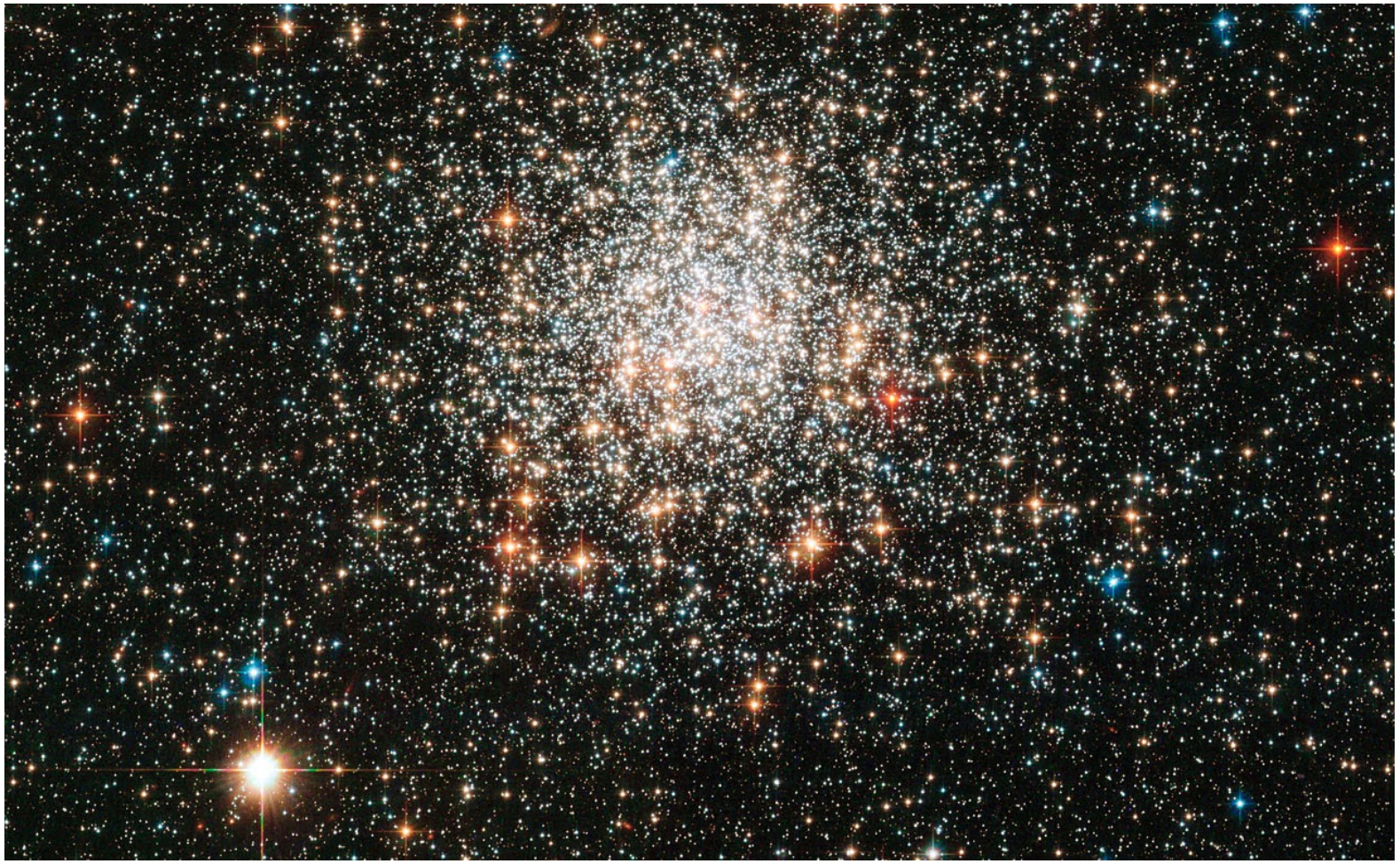


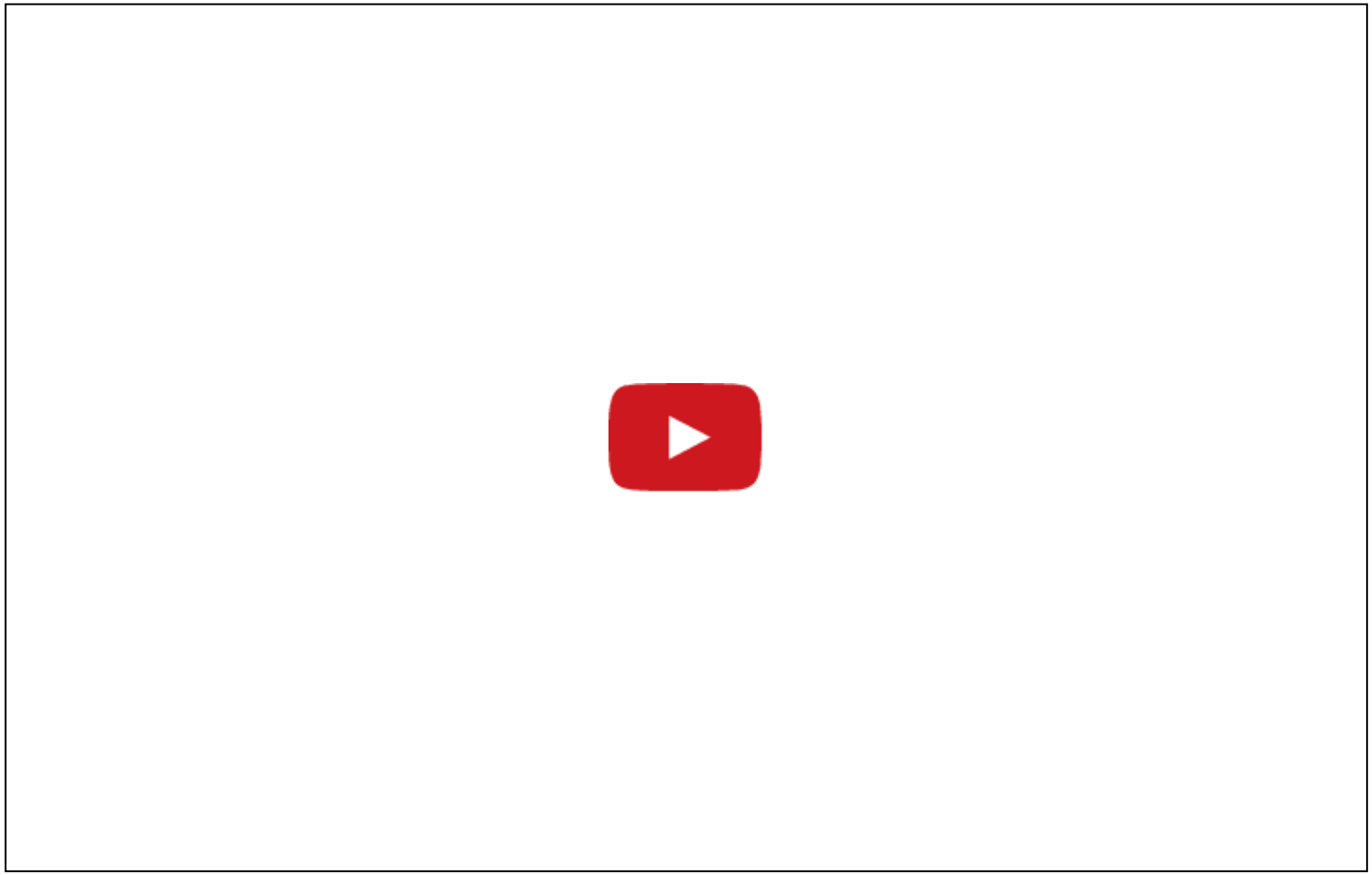
Image Credit: NASA

Break one big complex problem into millions of small simple ones by discretizing the system in time and space.

One way to do this is using a method called Smoothed-Particle Hydrodynamics

In [1]:
```
from IPython.display import YouTubeVideo
YouTubeVideo('N2Fah8RoX5M', width=853, height=480)
```

Out[1]:



GADGET-2 (http://www.mpa-garching.mpg.de/gadget/)

- Massively MPI parallel N-body + Smoothed-Particle Hydrodynamics code
- Standard code written in C
- Widely used to study a variety of astrophysical problems (mostly star/galaxy formation)
- Simulations use from 100,000 to 10,000,000,000 particles depending on the problem.
- Can take from 1,000 to 10,000,000 CPU hours
- Produces regular snapshots (full system state) can be GBs to TBs in size.
- Easily customizable, relatively speaking

Snapshots are saved in HDF5 files, or a code-specific binary format

My Simulations:

- ~25,000,000 particles
- 36-48 hours running on 256 cores (~10,000 CPU hours)
- ~2000 snapshots of 2-3 GB, totalling ~5 TB

Stampede: 100,000-core supercomputer located at TACC, #7 in the world.

Now let's talk about the data...

GADGET snapshot files are composed of several HDF5 groups:

One for each particle type, plus a header.

- `Header`: Necessary simulation metadata, stored as HDF5 attributes.
- `PartType0`: Particle Type 0, typically gas (SPH particles)
- `PartType1`: Particle Type 1, typically dark matter (N-body particles)
- et cetera

Particle fields include:

- Particle ID
- Mass
- Cordinates
- Velocities
- Density
- Smoothing Length
- Internal Energy
- Adiabatic Index
- + other custom fields

Simulation data is often big, but not unmanageable.

PyGadget is designed with this sort of 'medium' data in mind.

- HDF5 allows us to load only the data we're interested in.
- Library is designed to refine the data as it is loaded
- DataFrames help keep data aligned as we load, refine, and drop fields to save memory.
- Parallel batch processing via `multiprocessing`

PyGadget also provides tools for coordinate transformation...

- Conversion from comoving to physical coordinates
- Box centering
- Rotation around arbitrary axes
- Conversion to cylindrical/spherical coordinates

And SPH visualization.

- Primary routine written in C, paralellized with OpenMP, and wrapped using `scipy.weave`
- Additional pure python routine JIT-compiled using Numba.


WE'LL DO IT
LIVE!

PyGadget is available on github at:

github.com/hummel/pyGadget

Thank you!