The yt project aims to produce an integrated science environment for collaboratively asking and answering astrophysical questions. To do so, it will encompass the creation of initial conditions, the execution of simulations, and the detailed exploration and visualization of the resultant data. It will also provide a standard framework based on physical quantities interoperability between codes.

yt is designed to guide scientific inquiry (analysis, visualization, simulation) through physically-motivated understanding. It is released under the BSD license, developed completely in the open, and is designed to present a library of loosely-coupled components.

Mission Statement

The yt project aims to produce an integrated science environment for collaboratively asking and answering astrophysical questions. To do so, it will encompass the creation of initial conditions, the execution of simulations, and the detailed exploration and visualization of the resultant data. It will also provide a standard framework based on physical quantities interoperability between codes.

yt is designed to guide scientific inquiry (analysis, visualization, simulation) through physically-motivated understanding. It is released under the BSD license, developed completely in the open, and is designed to present a library of loosely-coupled components.

Interface

- **Year One**: Interactive HTML5 data exploration widgets for Jupyter notebook, including 2D and 3D exploration of reduced data products generated by yt. Initial implementation is available and undergoing UX testing.

- **Year Two**: A "data hub" for uploading reduced data products that can be accessed remotely and shared between individuals. This is now available and openly available at use.yt, built on Jupyterhub server and Girder.

- **Year Three**: "Outreach hub" for curating and providing access to visualizations constructed and designed for public consumption. This is being developed on the Girder platform.

Infrastructure

- **Year One**: Support for non-Eulerian datasets such as halo and galaxy catalogs, as well as SPH and N-body simulation datasets. This functionality is in general release and is being used in production.

- **Year Two**: Full integration of non-Cartesian datasets, such as cylindrical and spherical coordinate systems with non-uniform volume elements. This is now in general release and is being used in production.

- **Year Three**: Unstructured mesh analysis and visualization fully supported, including Voronoi tessellation techniques. Unstructured mesh (including higher order meshes) is now available in beta.

Instrumentation

- **Year One**: Development of a generic "fluid container" and interfaces to microphysical solvers utilizing this object. This has been implemented within the Grackle and Dengo libraries.

- **Year Two**: A command-server library for communication with yt, including in situ analysis and visualization. Protocol is under development, but not yet implemented.

- **Year Three**: IO library for reading and writing fluid containers to disk, accessible through command-server interface. Python version of this library is functional and C/F interface is in beta.

Community Growth

Growing a community of invested users and developers is key to yt's success. In the SI2 proposal, we have identified several metrics by which to measure community growth and strength. We will measure the number of papers using yt, the growth of the mailing list, and the number of contributors to the project.

Numerous bootcamps and workshops have been conducted, including upcoming workshops in Illinois and Boston.

This work was supported by NSF SI2-SSE ACI-1339624 (goo.gl/qcVx3o).