11th World Congress on Computational Mechanics (WCCM XI)
5th European Conference on Computational Mechanics (ECCM V)
6th European Conference on Computational Fluid Dynamics (ECFD VI)
July 20–25, 2014, Barcelona, Spain

THE IMPACT OF COMMUNITY SOFTWARE IN ASTROPHYSICS

A. Dubey[1], M. Turk[2] and B. O'Shea[3]

[1] Lawrence Berkeley National Laboratory, 1 Cyclotron Road, Berkeley, CA 94720, USA, adubey@lbl.gov and http://crd.lbl.gov/groups-depts/ANAG/about/staff/anshu-dubey/
[2] Astronomy and Astrophysics, Columbia University, Mail Code 5246, 550 W 120th St, New York, NY 10027, matthewturk@gmail.com
[3] Department of Physics and Astronomy, Lyman Briggs College, and the Institute for Cyber-Enabled Research, Michigan State University, East Lansing, MI 48824, USA, oshea@msu.edu

Key words: *Community Astrophysics Codes, Impact on Communities*

The astrophysics community has been ahead of many other science communities in making research codes publicly available, and therefore in the development and adoption of community codes. ZEUS-2D [5] was one of the earliest codes to become public, and it has been followed by several others such as FLASH [1], Gadget [3], Enzo [2] and Athena [4]. This list is not exhaustive, but it provides a glimpse of the open software development that prevails in the astrophysical community. It is this culture of openness and sharing of code that is further reflected in the development and rapid adoption of *yt* [6], an analysis and visualization package. In this contribution, we highlight the impact of three of these packages – FLASH, Enzo and *yt* – on the astrophysics community.

FLASH has been in existence since 1999 and first became public in 2000. Enzo's development started in 1996 and the code was made publicly available in 2003. FLASH has had over 50 significant developers during its lifetime, while Enzo has had approximately 25. FLASH has had centralized development at the University of Chicago's Flash Center, while Enzo currently has 12 active developers spread over 10 institutions. FLASH development has been predominantly supported by NNSA-DOE, with some funding from NSF, while Enzo has had multiple streams of funding. FLASH has been downloaded more than $3,000$ times with several of the downloads by the same person or group, indicating continued use of the code by corresponding users. Over 850 papers have been published that used FLASH to obtain all or part of their results. Enzo has similar statistics. Together, these codes annually account for the usage of several hundred million cpu hours on supercomputers run by DOE, NSF, NASA and other international agencies. Several PhD dissertations have used one of these codes as their primary simulation tool. Furthermore, involvement in these communities has had strong positive impact on the career trajectories of young scientists.

The analysis framework *yt* was instrumented in a similar fashion to FLASH; while initially designed to support only the Enzo simulation code, it has been designed to be modular in its intake of data while retaining a consistent public data access API and internal data representation. This enables identical analysis scripts to be applied to data produced by simulation codes such as FLASH, Enzo, and Athena. Recently added support for octree and particle-based codes such as RAMSES and Gadget has led to significant increase in the size of the community. Over the three years since the *yt* method paper [6] has been published, it has been cited by approximately 95 different papers. During the lifetime of the code, code contributions from nearly 50 different authors (ranging from very small to very large) have been accepted to the main code base and the user and developer mailing list have 220 and 70 members, respectively. Community growth has been important to all three of these projects. An active user and developer community serves as an extremely important function of reducing the barrier to entry for new researchers using HPC resources for scientific simulations. This has been a particularly high priority for *yt*, as it is segmented into both fundamental infrastructure (typically contributed by a core of individuals) and analysis modules and/or code frontends, which are often contributed by individuals that are newer to the project or who have developed technology for their own use. The *yt* project itself is designed to remove barriers to direct technology sharing between researchers by providing a common analysis platform. These, and similar projects in other fields, show that if reliable and robust codes exist, and efforts are made to promote their use, they facilitate higher overall scientific output. They also help foster open science in their corresponding communities, which has wide-ranging benefits.

REFERENCES

[1] A. Dubey, K. Antypas, M.K. Ganapathy, L.B. Reid, K. Riley, D. Sheeler, A. Siegel, and K. Weide. Extensible component-based architecture for FLASH, a massively parallel, multiphysics simulation code. *Parallel Computing*, 35(10-11):512–522, 2009.

[2] The Enzo Collaboration, Bryan, G. L., Norman, M. L., et al. Enzo: An Adaptive Mesh Refinement Code for Astrophysics *arXiv:1307.2265*, July 2013

[3] V. Springel. The cosmological simulation code GADGET-2. *MNRAS*, 364:1105–1134, December 2005.

[4] J. M. Stone, T. A. Gardiner, P. Teuben, J. F. Hawley, and J. B. Simon. Athena: A new code for astrophysical MHD. *APJS*, 178:137–177, September 2008.

[5] J. M. Stone and M. L. Norman. ZEUS-2D: A radiation magnetohydrodynamics code for astrophysical flows in two space dimensions. I - The hydrodynamic algorithms and tests. *APJS*, 80:753–790, June 1992.

[6] Matthew J. Turk, Britton D. Smith, Jeffrey S. Oishi, Stephen Skory, Samuel W. Skillman, Tom Abel, and Michael L. Norman. yt: A multi-code analysis toolkit for astrophysical simulation data. *The Astrophysical Journal Supplement Series*, 192(1):9, 2011.