PyNE Progress Report

A. Scopatz[1] for
C. R. Bates[2]
[1]Univ. of Wisconsin Madison
[2]Univ. of Cal. Berkeley / LLNL

ANS Winter Meeting, November 11, 2014

Outline

- PyNE [1]: what is it? (Python for Nuclear Engineering)
- New Features
- Improved Usability
- Future plans
- Expanding the community

What is PyNE?

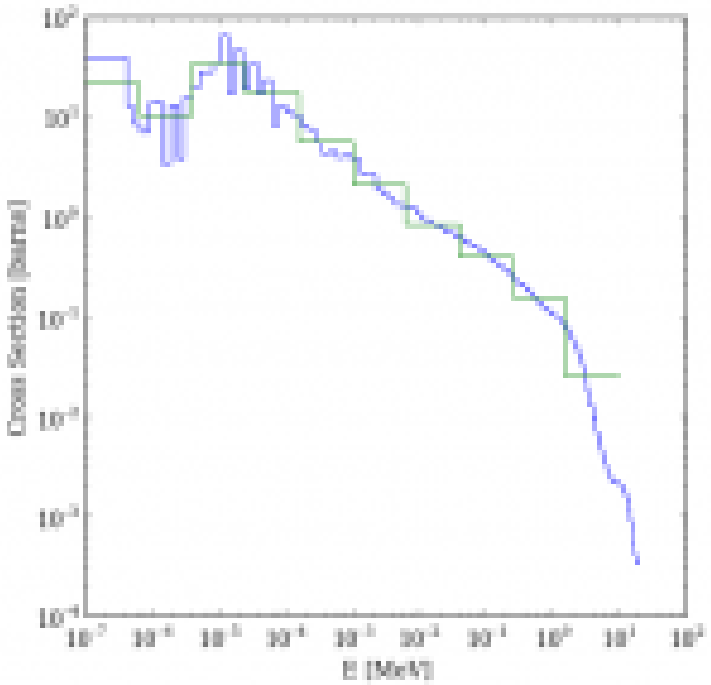
PyNE is the open source nuclear engineering toolkit.

- PyNE is a library of composable tools used to build nuclear science and engineering applications
- It is permissively licensed (2-clause BSD)
- It supports both a C++ and a Python API
- The name 'PyNE' is a bit of a misnomer since most of the code base is in C++ but most daily usage happens in Python
- v0.4 is the current, stable release
- As an organization, PyNE was born in April 2011 (however, core parts of PyNE have existed since 2007)

What are the Goals of PyNE?

To help nuclear engineers:

- be more productive (don't reinvent the wheel!)
- have the best solvers
- have a clear and useful API
- write really great code
- teach the next generation



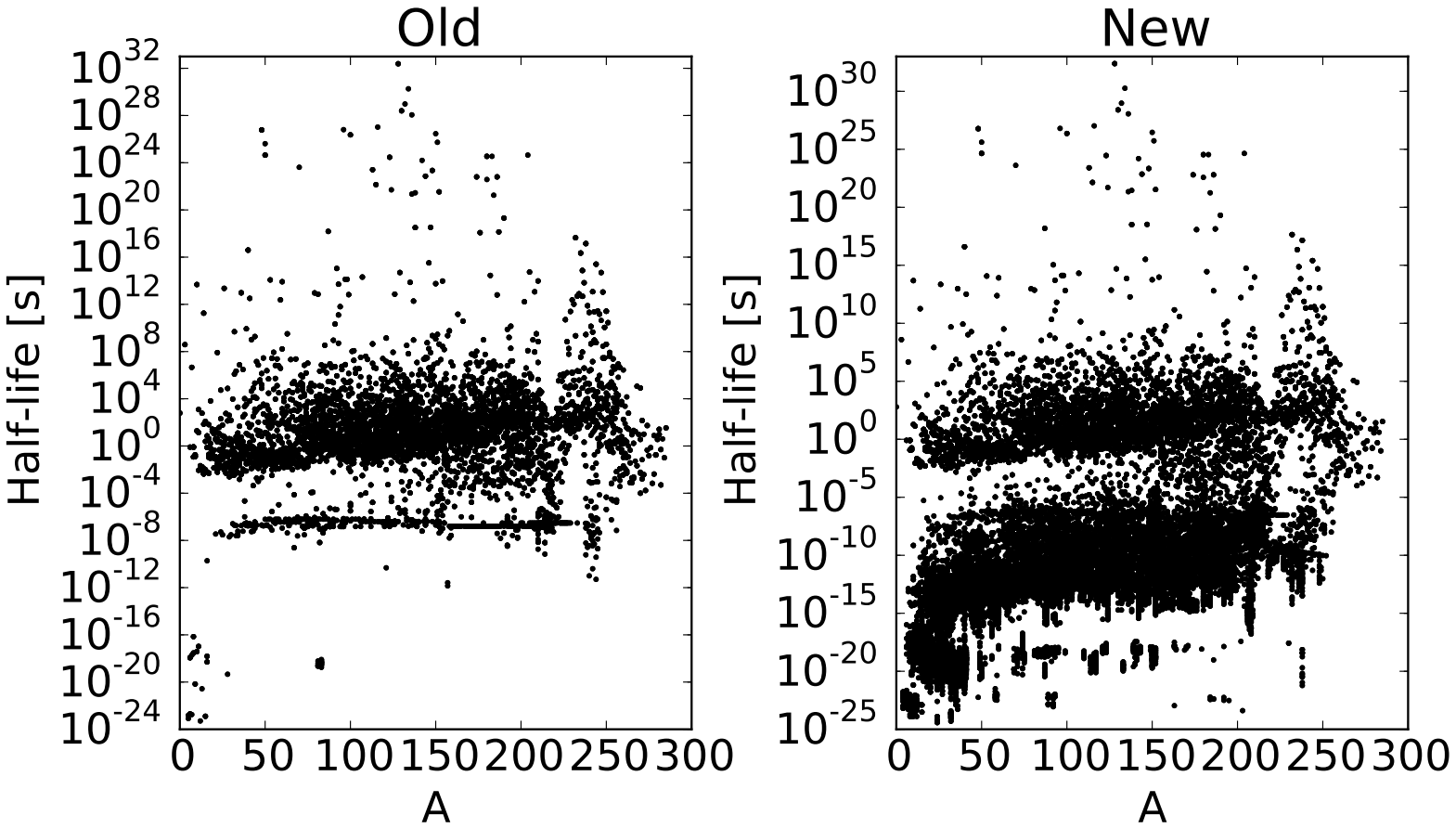
What Can PyNE Do?

The idea is to be able to easily combine components and avoid redeveloping utilities someone else has developed. **new/improved**

- Nuclear data and cross-section reading/processing
  - **ENSDF**, ENDF, **ACE**
- Material management
  - **FLUKA**, MCNP
- Canonical nuclide, particle and reaction naming conventions
- **Mesh operations**
  - **DAGMC, ALARA, CADIS**
- MCNP and Serpent input/output parsing
- Fuel cycle functionality (transmutation, enrichment)
- **AHOT**
- **Rigorous Two-Step Activation**
- **MORE!**

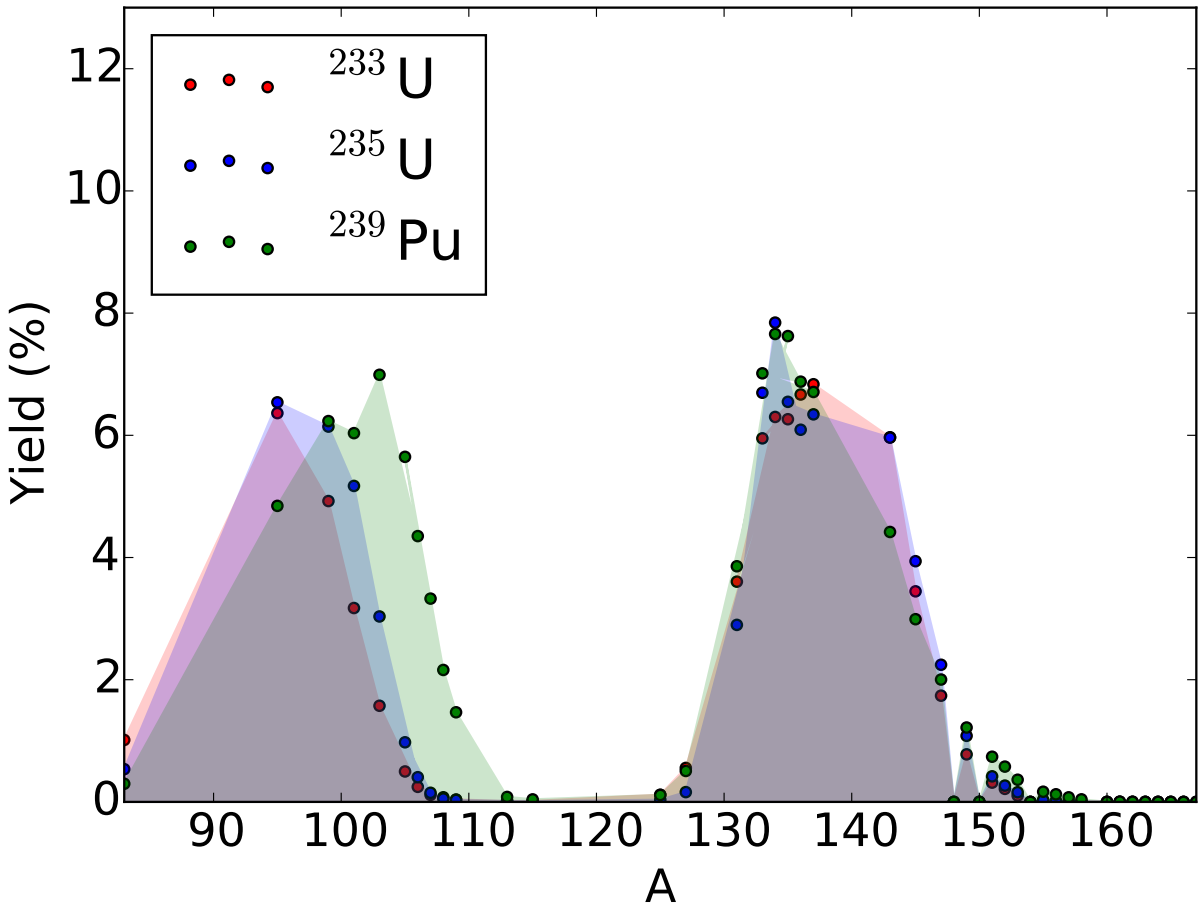
Improvements in ENSDF

- Completely re-written parser
- Reads all level and decay information into HDF5



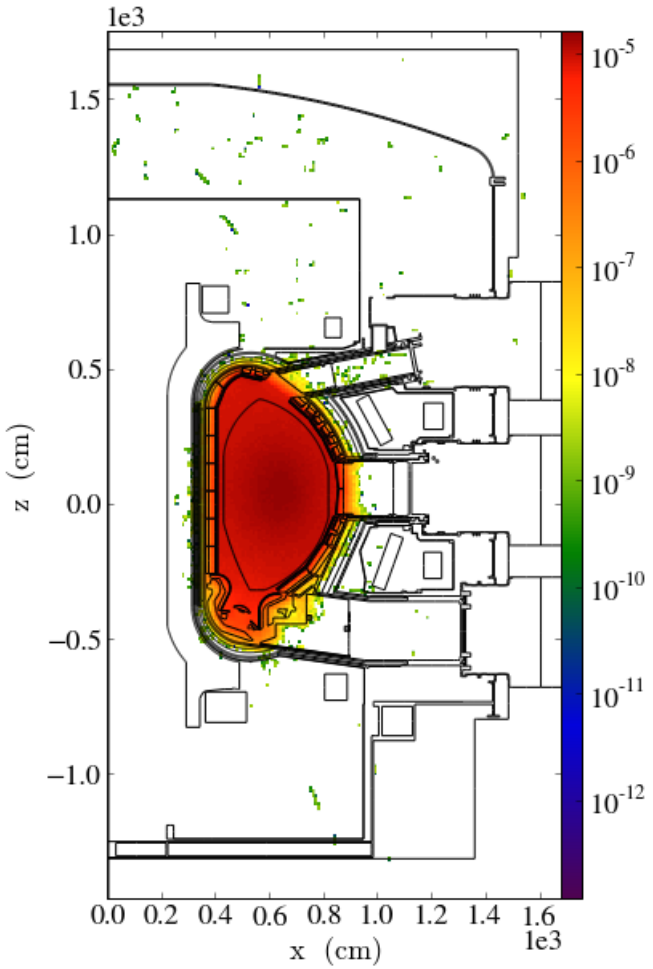
Addition of Fission Yield data

- WIMSD and NDS fission product yield data



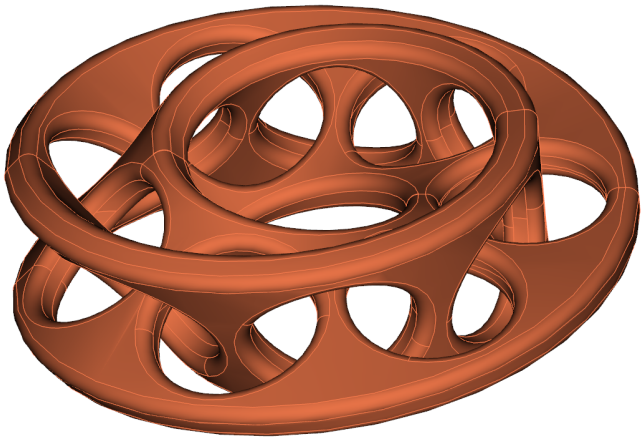
Meshes in PyNE

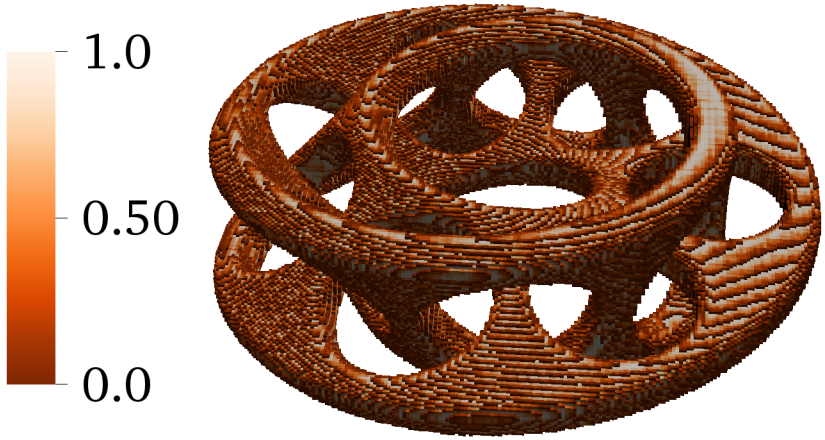
- Facilitates manipulating and plotting of meshes (with Yt's help)
- Reads MCNP mesh tallies
- Writes MCNP mesh inputs



Discretizing Geometry

- Discretize geometry on tetrahedral or cartesian mesh for transport
- Redistributes volume averaged properties





Rigorous Two-Step Activation

- Neutrons activate system components, gamma source persists after shutdown.
- Important for rad safety at ITER, JET, NIF, etc.

Even More Features!

- fluka support
- C++ amalgamation
- Tally container
- Python 3 support and CI
- ACE 2.0.0 Header support
- Conda binaries

Future Plans

- Variance Reduction Workflows??
- Plug and play transport tools/nuclear data
- Wrap ENSDF utility/analysis codes
- GND/FUDGE integration (Pending BSD license change)

Tutorials

- UC Berkeley Nov. 2013
  - Engage Profs. and Grad + Educate Undergrads
- ANS RPSD 2014
  - Engage national labs + industry

Hackathons

- Development sprints
- Congregate in one geographic location
- Feb. 2014 - 0.4 sprint
- Nov. 2014 - 0.5 sprint

Why Would I Get Involved?

As a user:

- You could do your work or research with PyNE
- Even if you have your own software that looks and behaves similarly to some aspects of PyNE, using PyNE will mean that you no longer have to develop AND maintain that functionality

As a developer:

- You should be selfish
- Contribute to PyNE in ways that support the work that you are doing
- If a feature you want is not in PyNE right now, chances are that other people want to see that feature too
- This will help your future self as much as future other people

How Can I Get Involved?

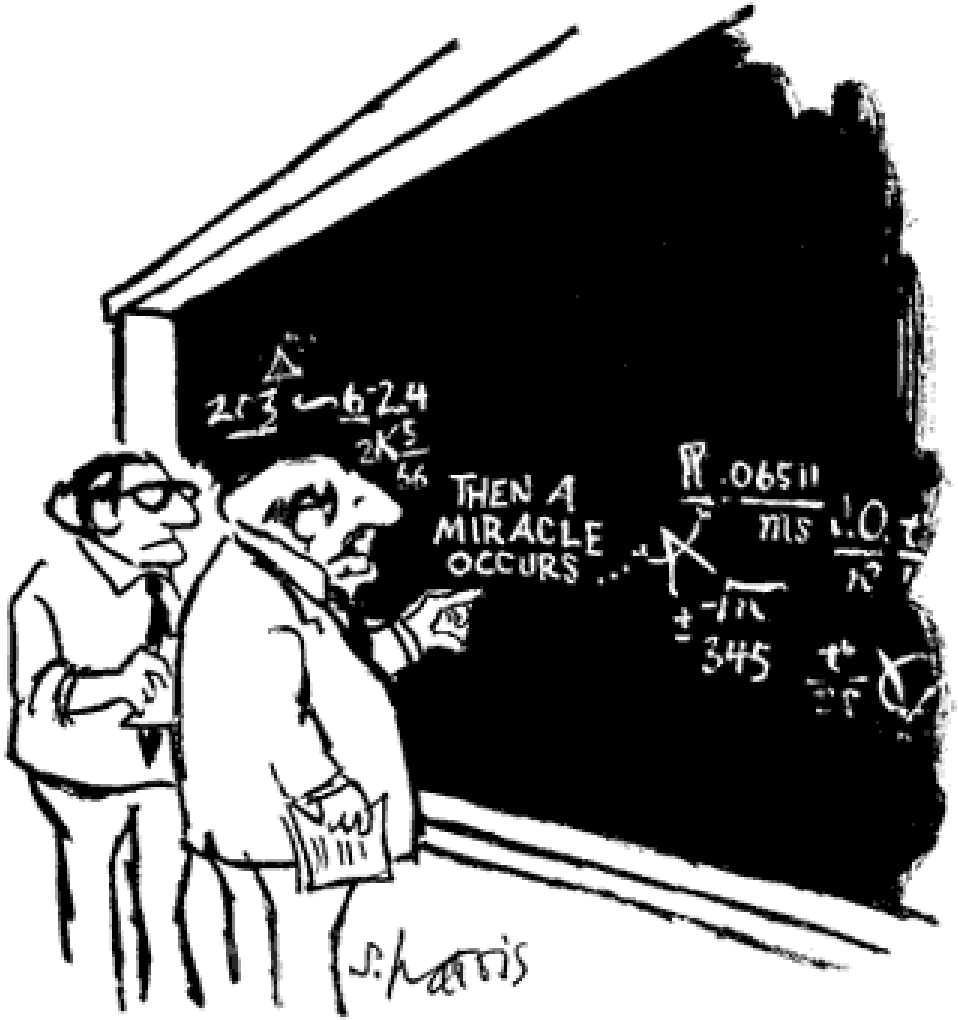
Contact PyNE

- Website: `http://pyne.io/`
- User's Mailing List: `pyne-users@googlegroups.com`
- Developer's List: `pyne-dev@googlegroups.com`
- GitHub: `https://github.com/pyne/pyne`
- Tutorial: `http://pyne.io/tutorial/index.html`

What goes into PyNE?

Anything that is not export controllable, proprietary, or under HIPPA restrictions! (If you have questions, *ask*)

QUESTIONS?



"I think you should be more explicit here in step two."

PyNE In the Literature

- Intro: "PyNE: Python For Nuclear Engineering" [2]
- Progress reports: [3], [4]
- In research: [5], [6], [7]
- V&V: "Quality Assurance within the PyNE Open Source Toolkit" [8]
- Poster at SciPy: [9]

References I


the PyNE Development Team.
PyNE: The Nuclear Engineering Toolkit, 2014.


Anthony Scopatz, Paul K. Romano, Paul P.H. Wilson, and Kathryn D. Huff.
PyNE: Python for Nuclear Engineering.
In *Am. Nuc. Soc. Winter Meeting 2012*, volume 107, San Diego, CA, USA, November 2012.


Anthony Scopatz, Elliott D. Biondo, Carsten Brachem, John Xia, and Paul P. H. Wilson.
PyNE Progress Report.
In *Am. Nuc. Soc. Winter Meeting 2013*, volume 109, Washington, D.C., USA, November 2013.

References II


Cameron Bates, Elliott Biondo, Kathryn Huff, and et al.
PyNE Progress Report.
In *Am. Nuc. Soc. Winter Meeting 2014*, volume 111, Anaheim, CA, USA, November 2014.


E. Biondo, A. Davis, A. Scopatz, and P. P. H. Wilson.
Rigorous Two-Step Activation for Fusion Systems with PyNE.
In *Proc. of the 18th Topical Meeting of the Radiation Protection & Shielding Division of ANS*, Knoxville, TN, 2014.


J.I. Mrquez Damin, J.R. Granada, and D.C. Malaspina.
{CAB} models for water: A new evaluation of the thermal neutron scattering laws for light and heavy water in endf-6 format.
Annals of Nuclear Energy, 65(0):280 – 289, 2014.

References III


Anthony Scopatz.
First & second order approximations to stage numbers in multicomponent enrichment cascades.
In *International Conference on Mathematics and Computational Methods Applied to Nuclear Science & Engineering (M&C 2013)*, Sun Valley, ID, USA, May 2013.


Elliott Biondo, Anthony Scopatz, Matthew Gidden, Rachel Slaybaugh, and Cameron Bates.
Quality Assurance within the PyNE Open Source Toolkit.
In *Am. Nuc. Soc. Winter Meeting 2014*, volume 111, Anaheim, CA, USA, November 2014.


Anthony Scopatz, Paul Romano, Paul Wilson, Rachel Slaybaugh, Katy Huff, and Eric Relson.
PyNE: Python for Nuclear Engineering.
In *SciPy 2012*, Austin, TX, USA, July 2012.