PyNE Progress Report

Anthony Scopatz[1] for Cameron R. Bates[2,3]

Elliott Biondo[1], Kathryn Huff[2], Kalin Kiesling[1], Robert Carlsen[1], Andrew Davis[1], Matthew Gidden[1], Tim Haines[1], Joshua Howland[2], Blake Huff[2], Kevin Manalo[4], Arielle Opotowsky[1], Rachel Slaybaugh[2], Eric Relson[1], Paul Romano[5], Patrick Shriwise[1], John D. Xia[6], Paul Wilson[1], Julie Zachman[1]

[1] The University of Wisconsin-Madison
[2] The University of California, Berkeley
[3] Lawrence Livermore National Laboratory
[4] Georgia Institute of Technology
[5] Massachusetts Institute of Technology
[6] University of Chicago

Outline

- PyNE [1]: what is it? (Python for Nuclear Engineering)
- New Features
- Improved Usability
- 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 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!**

MESH-BASED SOURCE SAMPLING

Randomly sample Monte Carlo birth parameters (position, energy, statistical weight) from a mesh-based source.

The PyNE `Sampler` class:

- Cartesian and tetrahedral mesh
- Utilizes alias sampling method [2]
- Sampling modes:
  - analog – no source biasing
  - uniform – all space sampled equally, statistical weight adjusted accordingly
  - user-specified – any bias scheme specified by the user
    - Consistent Adjoint-Driven Importance Sampling (CADIS) [3] in PyNE `variance_reduction` module

Using the Sampler class

- Written in C++
- Intended to be integrated into physics codes
  - C/C++, Fortran, Python interfaces
- Use with MCNP5 via a PyNE source.F90 file

CADIS

Consistent Adjoint-Driven Importance Sampling [3]

- A Monte Carlo variance reduction technique that uses a deterministic estimate of the adjoint flux.
  - Source biasing
  - Weight windows
- PyNE has a mesh-based implementation:
  - Uses the PyNE source sampling capabilities with MCNP for mesh-based source biasing.
  - Adjoint fluxes from must be supplied, e.g. from the Denovo 3D S_N code [4].

Installing PyNE

- Conda binary available (no compiler needed!)
- Ubuntu shell script
- Conda recipe (for developers)
- Single library (easier linking)
- Instructions for multiple platforms
- Relative links set during build

Even More Features!

- `fluka` support
- C++ amalgamation
- Tally container
- Python 3 support
- Continuous Integration (CI) testing
- Standardized style
- ACE 2.0.0 Header support

Tutorials + Hackathons

- Tutorial - UC Berkeley Nov. 2013
  - Engage Profs. and Grad + Educate Undergrads
- 0.4 Sprint - UC Berkeley Feb. 2014
  - Homogenizing code style
- Tutorial - ANS RPSD 2014
  - Engage national labs + industry
- Hackathon - UC Berkeley Nov. 14-16
  - Verification and Validation

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*)

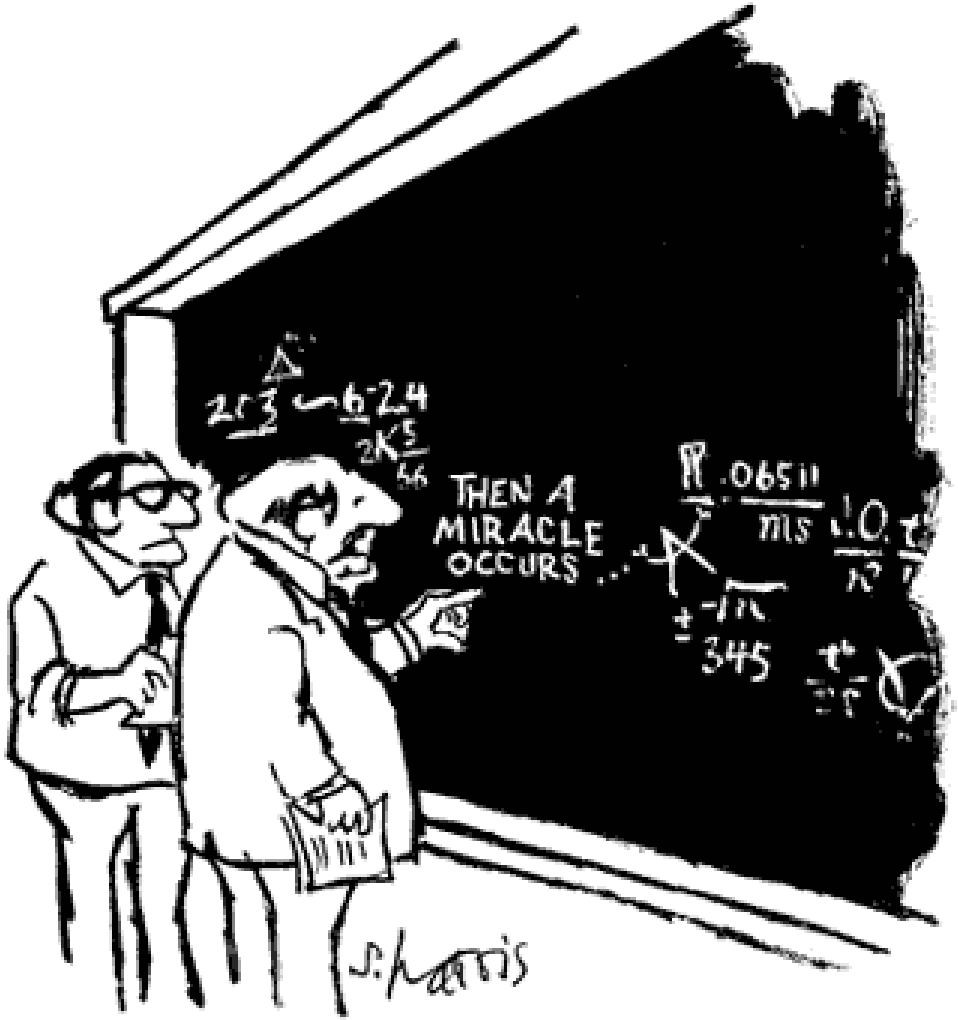
Acknowledgements

- PyNE is a diverse collection of knowledge
- We need lots of specialist to fill in the gaps:

Anthony Scopatz[1], Cameron R. Bates[2,3], Elliott Biondo[1], Kathryn Huff[2], Kalin Kiesling[1], Robert Carlsen[1], Andrew Davis[1], Matthew Gidden[1], Tim Haines[1], Joshua Howland[2], Blake Huff[2], Kevin Manalo[4], Arielle Opotowsky[1], Rachel Slaybaugh[2], Eric Relson[1], Paul Romano[5], Patrick Shriwise[1], John D. Xia[6], Paul Wilson[1], Julie Zachman[1]

[1] The University of Wisconsin-Madison
[2] The University of California, Berkeley
[3] Lawrence Livermore National Laboratory
[4] Georgia Institute of Technology
[5] Massachusetts Institute of Technology
[6] University of Chicago

Questions?



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

PyNE In the Literature

- Intro: "PyNE: Python For Nuclear Engineering" [5]
- Progress reports: [6], [7]
- In research: [8], [9], [10]
- V&V: "Quality Assurance within the PyNE Open Source Toolkit" [11]
- Poster at SciPy: [12]

References I


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


J. Cole Smith and Sheldon H. Jacobson.
An analysis of the alias method for discrete random-variate generation.
INFORMS Journal on Computing, 17(3):321–327, 2005.


Alireza Haghighat and John C. Wagner.
Monte carlo variance reduction with deterministic importance functions.
Progress in Nuclear Energy, 42(1):25–53, 2003.


T. M. Evans, A. S. Stafford, R. N. Slaybaugh, and K. T. Clarno.
Denovo: A new three-dimensional parallel discrete ordinates code in scale.
Nuclear Technology, 171:171–200, 2010.

References II


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.


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.

References III


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.


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.

References IV

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.