WHY PYTHON IS THE NEXT WAVE IN EARTH SCIENCES COMPUTING

BY JOHNNY WEI-BING LIN

So, why all the fuss about Python? Perhaps you have heard about Python from a coworker, heard a reference to this programming language in a presentation at a conference, or followed a link from a page on scientific computing, but wonder what extra benefits the Python language provides given the suite of powerful computational tools the Earth sciences already has. This article will make the case that Python is the next wave in Earth sciences computing for one simple reason: Python enables users to do more and better science. We'll look at features of the language and the benefits of those features. This article will describe how these features provide abilities in scientific computing that are currently less likely to be available with existing tools, and highlight the growing support for Python in the Earth sciences as well as events at the upcoming 2013 AMS Annual Meeting that will cover Python in the Earth sciences.

Python is a modern, interpreted, object-oriented, open-source language used in all kinds of software engineering. Though it has been around for two decades, it exploded into use in the atmospheric sciences just a few years ago after the development community converged upon the standard scientific packages (e.g., array handling) needed for atmospheric sciences work. Python is now a robust integration platform for all kinds of atmospheric sciences work, from data analysis to distributed computing, and graphical user interfaces to geographical information systems. Among its salient features, Python has a concise but natural syntax for both arrays and nonarrays, making programs exceedingly clear and easy to read; as the saying goes, "Python is executable pseudocode." Also, because the language is interpreted, development is much easier; you do not have to spend extra time manipulating a compiler and linker. In addition, the modern data structures and object-oriented nature of the language makes Python code more robust and less brittle. Finally, Python's open-source pedigree, aided by a large user and developer base in industry as well as the sciences, means that your programs can take advantage of the tens of thousands of Python packages that exist. These include visualization, numerical libraries, interconnection with compiled and other languages, memory caching, Web services, mobile and desktop graphical user interface programming, and others. In many cases, several packages exist in each of the above domain areas. You are not limited to only what one vendor can provide or even what only the scientific community can provide!

A number of other languages have some of Python's features: Fortran 90, for instance, also supports array syntax. Python's unique strengths are the interconnectedness and comprehensiveness of its tool suite and the ease with which one can apply innovations from other communities and disciplines. Consider a typical Earth sciences computing workflow. We want to investigate some phenomena and decide to either analyze data or conduct model experiments. So, we visit a data archive and download the data via a Web request, or we change parameters in the (probably Fortran) source code of a model and run the model. With the dataset or model output file in hand, we write an analysis program, perhaps using IDL or MATLAB, to conduct statistical analyses of the data. Finally, we visualize the data via a line plot or contour plot. In general, we accomplish this workflow using a kludge of tools: shell scripting for Web requests and file management, the Unix tool Make for code and compilation management, compiled languages for modeling, and IDL or MATLAB for data analysis and visualization. Each tool is isolated from every other tool, and communication between tools occurs through files.

AFFILIATION: LIN—Physics Department, North Park University, Chicago, Illinois
CORRESPONDING AUTHOR: Johnny Wei-Bing Lin, Physics Department, North Park University, 3225 W. Foster Ave., Chicago, IL 60625
E-mail: johnny@johnny-lin.com

DOI:10.1175/BAMS-D-12-00148.1

Interested in Learning More about Python?

The following are three places among many with details about acquiring and using Python: The PyAOS blog and mailing list (http://pyaos.johnny-lin.com) is a resource and community for Python users in the atmospheric and oceanic sciences (AOS). The Python Tutorial (http://docs.python.org/tutorial) is the standard introduction to Python. New AOS users may find the book, *A Hands-On Introduction to Using Python in the Atmospheric and Oceanic Sciences,* to be more tailored to their workflow; it is available online in a free version at www.johnny-lin.com/pyintro.

In Python, every tool is used in the same interpreted environment. And data flow does not have to occur through files; we can access any variable in any tool at any (logical) time in the workflow. As a result, our workflow becomes much more robust and flexible. Additionally, the breadth of the Python tool set means the most important capabilities we need will very likely be available even if a given vendor changes its priorities. Python users also have much greater ability to access innovations from industries outside of the Earth sciences. This latter advantage will become increasingly more important as we push inexorably into the world of cloud computing, big data, and mobile computing; in the next decade, the Earth sciences cannot afford to ignore these innovations.

To be fair, Python has real disadvantages, including the fact that pure Python code runs much slower than compiled code, there are fewer scientific libraries compared to Fortran, and documentation and support for new science users is relatively sparse. There are tools to overcome the speed penalty, the collection of scientific libraries is growing, and science support resources are becoming more robust; nevertheless, these are real issues. For most Earth science applications, the strengths of Python outweigh the weaknesses.

The Earth sciences community has begun to recognize the unique benefits of Python, and as a result its population of users is growing. Adoption of Python by Earth sciences users is no longer confined to "early adopters." Institutional support includes groups at Lawrence Livermore National Laboratory's Program for Climate Model Diagnosis and Intercomparison, NCAR's Computer Information Systems Laboratory, and the British Atmospheric Data Centre. Several packages exist that handle file formats used by atmospheric and oceanic sciences users, including netCDF, HDF4, HDF-EOS 2, and GRIB 1 and 2 (e.g., via the netCDF4-Python, UV-CDAT, and PyNIO packages). And Python's utility as a common glue is being leveraged by both open-source and proprietary vendors: Python can be used as a scripting language for applications such as Unidata's Integrated Data Viewer, the NCAR Command Language (via PyNGL), GrADS (via OpenGrADS), and Esri's ArcGIS.

Over the last two years, the AMS has helped support the development of Python in the Earth sciences by sponsoring the Symposiums on Advances in Modeling and Analysis Using Python, as well as short courses on using Python in the atmospheric and oceanic sciences. At the 2013 AMS Annual Meeting in Austin, Texas, we will feature the Third Python Symposium; short courses for beginning, intermediate, and advanced users; and a Town Hall meeting to provide answers to questions nonprogrammers and decision makers may have about how Python can help institutions become more productive.

The computational tools the Earth sciences have traditionally used are powerful and have enabled us to accomplish much. With modern tools, however, we can accomplish more. Python, in particular, can help us do more science and do it better. Python enables better science through enabling clear code: its elegant syntax and broad and robust tool set helps Earth scientists write less buggy code. Python also enables more science by helping us to take advantage of the recent explosion of computing resources by giving us access to an amazing set of software tools: Python gives us new capabilities through its modern, object-oriented structure but also gives us access to the new capabilities being generated daily from industries outside of the sciences. More and more, Earth scientists are becoming aware of Python's ability to help us do better and more science. With this "perfect storm" of a clear and concise language, access to a previously unfathomable range of software tools, and a growing scientific user community, Python is poised to become the next wave in the computational Earth sciences.

ACKNOWLEDGMENTS. Thanks to Nick Barnes for suggesting the "clear code" and "better and more" science formulations, and Grant Petty for the title. Discussions with the members of the PyAOS atmospheric and oceanic sciences Python user community are greatly appreciated, as are comments from two anonymous reviewers.