First Pan American Congress on Computational Mechanics
PANACM 2015
April 27-29, 2015, Buenos Aires, Argentina

RESEARCH AND EDUCATION IN COMPUTATIONAL ENGINEERING WITH SCIENTIFIC PYTHON

LORENA A. BARBA,[*] ANDY R. TERREL[†]

[*] Department of Mechanical and Aerospace Engineering, The George Washington University
labarba@gwu.edu

[†] Chief Computatinal Scientist, Continuum Analytics
aterrel@continuum.io

Key words: Computational Mechanics, Education, Python.

ABSTRACT

The computational science community has been using Python for prototyping ideas, visualizing and presenting results, and computing education for many years. But a recent surge in adoption follows the introduction of "killer apps" like IPython Notebooks, interactive parallel computing with IPython, CUDA Python libraries, and more.

Providing a fully open-source ecosystem for all stages of the computational science workflow —from idea, to prototype, computation, data analysis, presentation and documentation all the way through—many scientists are also relying on Python to make their science practice more open and reproducible. Finally, Python offers an ideal environment for teaching computational skills to student scientists and engineers.

In this mini symposium, we want to showcase leading work in accelerating all stages of scientific knowledge building with the Python ecosystem of open-source tools, with special application to computational mechanics. As the ultimate stage in knowledge creation, computational science enters the educational arena, where nursing the next generation of students with low-barrier learning platforms is another advantage of Python. Therefore, the minisymposium will also include talks on educational innovations in the field of computational science and engineering, using Python as the enabling technology.

One common concern of computational scientists that are not yet enthusiastic about Python is the expectation that performance will be deficient, due to the overhead of an interpreted language. The concern is valid, but sometimes overlooks the many ways in which high performance can be achieved with Python. The NumPy library for array computing provides large performance improvements over interpreted iterations, while IPython parallel allows for loosely coupled parallel workloads in interactive mode. Finally, CUDA Python using the Numba Python compiler gives access to GPU acceleration directly from the high-level language. The minisymposium will present use-cases for all these.