The Journal of Open Source Software

DOI: DOIunavailable

Software

- Review
- Repository
- Archive

Editor: Pending Editor

Reviewers:

- @Pending Reviewers

Submitted: N/A
Published: N/A

title: ‘pymplot: An open-source, lightweight plotting package for efficient visualization of multi-dimens
tags:
- Python
- geophysics
- visualization
- multi-dimensional data
- volume
- slice
authors:
- name: Kai Gao[1]
affiliation: 1
- name: Lianjie Huang
affiliation: 1
affiliations:
- name: Geophysics Group, Los Alamos National Laboratory, Los Alamos, NM 87645, USA
index: 1
bibliography: refs.bib
date: 15 April, 2021

Summary

We develop a lightweight, easy-to-use plotting package based on Python and matplotlib for efficient visualization of multi-dimensional geophysical data. The package contains eight plotting functions that can efficiently and conveniently render 1D, 2D and 3D regular-grid scalar data into publication-quality figures of various formats. These plotting functions include plotting 1D scalar data as a curve or a set of colored scatter points, showing 2D regular-grid scalar data as an image, wiggles, or contours, and displaying 3D regular-grid scalar data as a volume or three orthogonal slices in the image or contour form. We develop this package to facilitate command-line-based, quick rendering of 1D, 2D and 3D scalar data into visually decent forms with simple commands and options. Our package is also capable of rendering various fonts, subscripts, superscripts, and mathematical symbols on different types of plots in a consistent manner.

Statement of Need

Data visualization is gaining increasing importance in modern scientific research (**Schroeder_etal_2000?**; **Ahrens_etal_2005?**; **Fogal_etal_2010?**; **Ramachandran_Varoquaux_2011?**). The reason behind such transition is that modern scientific research, particularly those associated with data analysis, image analysis, and experimental result analysis, essentially relies on descent rendering of data, images, or results, in various forms to convey important information of the research. Scientific disciplines such as geophysics, biology, astrophysics, experimental physics, computer vision, imaging science, among others, always require proper and effective data visualization for publication. In some fields, neat graphical representations of principles and results could explain ideas/findings much more clearly than complex equations and lengthy text descriptions.

Starting from simple dot, line and curve representations in early days, scientific data visualization nowadays enjoys a vast pool of plotting tools than ever. The most notable ones include matplotlib (**Hunter_2007?**), Mathematica (**Mathematica?**),

[1]Corresponding author; Email: kaigao@lanl.gov

The Journal of Open Source Software

MATLAB (**MATLAB?**), gnuplot (**gnuplot?**), VTK (**Schroeder_etal_2000?**), Mayavi (**Ramachandran_Varoquaux_2011?**), Paraview (**Ahrens_etal_2005?**), VisIt (**VisIt?**), OpenGL (**Woo_etal_1999?**), Surfer (**Surfer?**), to name a few without a specific order. For geophysical particularly earthquake and seismic studies, generic mapping tools (GMT) (**Wessel_etal_2019?**), SeismicUnix (SU) (**seismicunix?**), and Madagascar (**Madagascar?**) are probably the most popular packages.
The package MinesJTK (**MinesJTK?**) based on Java programming language with Python interfaces also gains attention in the seismic research community recently.

The aforementioned plotting tools are based on drastically different fundamental-layer libraries, programming languages, orientations, and goals, and they are not necessarily compatible with one another. For example, Madagascar uses its own special data I/O format, which is not widely adopted in any other aforementioned software or libraries. More importantly, some of these tools are generic plotting tools that are applicable to a wide range of visualization tasks, but are not necessarily convenient for rendering geophysical data. For instance, both MATLAB and Mathematica are fancy scientific computational tools, but neither of them is designed primarily for data visualization. Some of these tools require heavy or at least an intermediate level of programming effort to render even a simple dataset. For example, OpenGL and VTK are undoubtedly very comprehensive and powerful, but are generally considered to be not user-friendly for usual users, and both of them require a very steep learning curve. GMT partially shares this drawback of a steep learning curve until probably the most recent version (**Wessel_etal_2019?**). Another reason that motivates us to develop a plotting package for routine plotting tasks is that many of these aforementioned plotting tools are commercial software, and require high initial and annual subscription fees to use.

Though geophysical data may be in any modern data storage forms, such as regularly sampled, irregularly sampled, unstructured, multi-component (vector or tensor), the most frequently used data form is perhaps the regularly sampled scalar data in 2D and 3D shapes, e.g., seismic velocity, density, and subsurface images. More than frequently, researchers in the geophysics community only need to render these 2D and 3D data into simple forms such as 2D images, contours, 3D image volumes, or multiple slices. In such a case, some of these aforementioned tools may be overkill for these plotting tasks.

With consideration of convenience, simplicity, and lightweight in mind, we develop a plotting package based on Python and matplotlib for visualizing 1D scalar data and 2D/3D regularly sampled scalar data. We have no intention to develop our plotting package, `pymplot` as we name it for now, to surpass any existing plotting software. Our intention is to provide a convenient application layer to the Python-based plotting library, matplotlib. We choose Python and matplotlib because both of them are free, open-source, de facto most widely used, and portable on different operating system platforms. Our package is not an independent, generic, or fundamental layer or library. We design and implement the `pymplot` package mainly to render regularly-sampled scalar data to a limited number of visualization forms (images, contours, wiggles, volumes, etc.). It is therefore not comprehensive enough to rendering complex data forms other than regularly sampled scalar data, including unstructured (e.g., unstructured 2D or 3D mesh) or multi-component (e.g., vector or tensor). Properly rendering these types of complex data usually require significant advanced programming efforts, particularly for 3D unstructured or multi-component data. In short, our plotting package is a collection of *applications* with specific purposes, instead of a *library* with generic functions and interfaces.

To address these needs, we develop eight plotting functionalities: - `showgraph`: To show 1D scalar data as curves or scatter points with or without colors representing their attributes. - `showmatrix`: To show 2D regularly sampled scalar data as a colored image. - `showcontour`: To show 2D regularly sampled scalar data as contours. - `showwiggle`: To show 2D regularly sampled scalar data as wiggles. The 2D data should be regularly sampled along at least one of the two axes. - `showslice`: To show 3D regularly sampled scalar data using three orthogonal slices, each with in a colored image. - `showslicon`: To show 3D regularly sampled scalar data using three orthogonal slices, each in the form of contours.

- `showvolume`: To shown 3D regularly sampled scalar data using a volume in the orthogonal projection, with a sub-volume cropped out. - `showvolcon`: To shown 3D regularly sampled scalar data using a volume in the orthogonal projection, with a sub-volume cropped out.

Acknowledgments

This work was supported by the U.S. Department of Energy (DOE) through the Los Alamos National Laboratory (LANL). LANL is operated by Triad National Security, LLC, for the U.S. DOE National Nuclear Security Administration (NNSA) under Contract No. 89233218CNA000001. We thank Dave Hale of Colorado School of Mines for helpful discussions.

References