psyplot: Visualising rectangular and triangular Climate Model Data with Python

Philipp S. Sommer

UNIL | Université de Lausanne
Institut des dynamiques de la surface terrestre

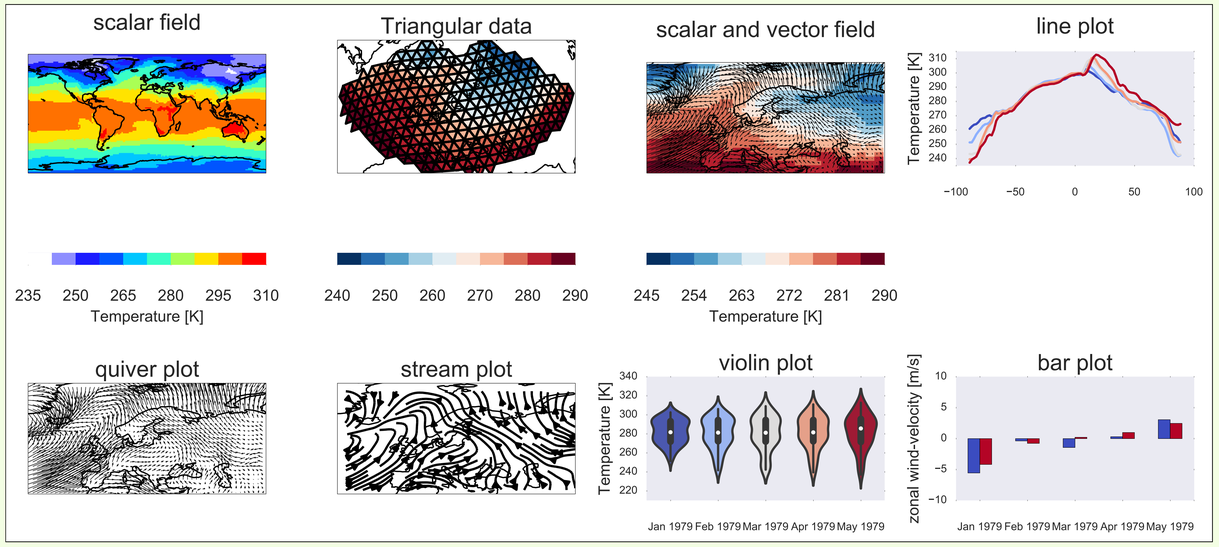
CC BY

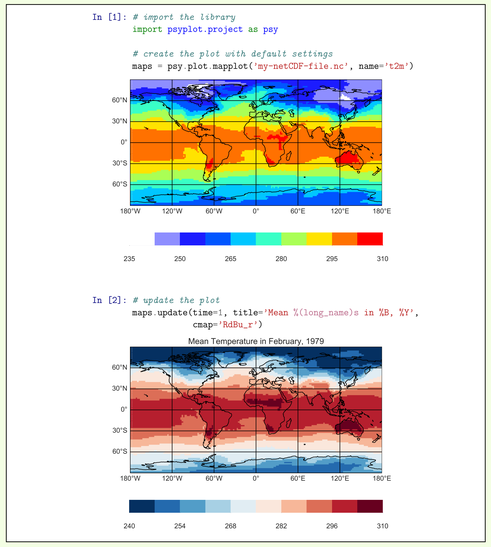
Abstract

The development and usage of climate models often requires the visualisation of the data on the globe. This visualisation should ideally be nice looking, simple in application, fast, easy reproducible and flexible. There exist a wide range of software tools to visualise rastered data which however often lack in their ability of being (easy) scriptable, have low flexibility or simply are far too complex for a quick look into the data. Therefore, we developed the new visualisation framework psyplot that aims to cover the visualisation in the daily work of earth system scientists working with data of the climate system. It is build (mainly) upon the python packages matplotlib, cartopy and xarray and supports (besides other functionalities) the visualisation of data on the globe. This data can either be stored in a NetCDF, GeoTIFF, any other format that is handled by the xarray package. Or it may be used with data that is currently processed and not already stored on the hard disk. Furthermore, the data may be on a rectangular grid (following or not following the CF Conventions) or on a triangular grid (following the CF Conventions (like the earth system model ICON).

The package visualises scalar and vector fields, enables to easily manage and formatting multiple plots at the same time and can export the plots to all formats covered by the matplotlib package. Psyplot can either be used with only a few lines of code from the command line in an interactive python session, via python scripts or from a graphical user interface (GUI). The framework developed in this package enables a very flexible configuration, an easy integration into other scripts using matplotlib and it is flexible for further development.

Examples



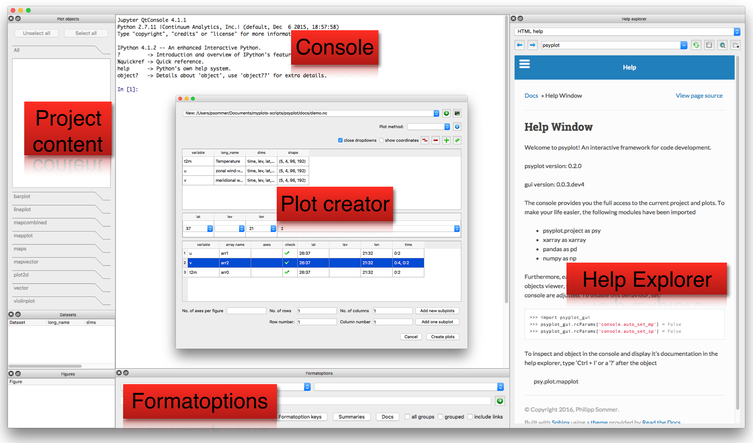


Module description

The graphical user interface

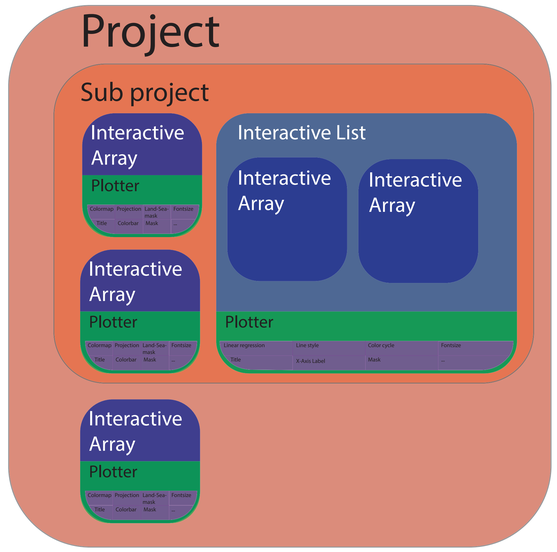
The psyplot_gui package provides a GUI for an easy access of the plotting features in psyplot. It bridges the gap between fast creation of plots and more or less complicate customisation using the capabilities of the matplotlib library.
The main parts of the GUI are

- **The console:** An in-process IPython console that provides the possibility to communicate with the psyplot package via the command line and to load any other module or to run any other script
- **The help explorer:** Motivated by the Object inspector of the Scientific PYthon Development EnviRonment Spyder, this widget provides a simple online browser and the possibility to show the documentation of Python objects as an html webpage. The help explorer is connected to several widgets of the GUI and especially to the console, to show the documentation of any python object.
- **The plot creator:** A widget to select data from a dataset and visualise it with a plot method of the psyplot package
- **The project content:** Shows the data arrays, the open datasets and the open figures in the current project



The Package structure

The psyplot framework consists mainly of five base classes that interact with each other:



- **The Project class** (or rather instances of it) contains multiple data objects that are (or are not) visualised. It mainly spreads update commands to the contained plot objects but also contains plot methods which are defined by the registered plotters.
  A project may be split up into sub projects which then only control a specific part of the main project.
- **The InteractiveArray class** is an enhanced version of the xarray.DataArray representing the data of one variable with reference to it's coordinates. It may be one or multidimensional depending on the chosen visualisation method
- **The InteractiveList class** is a collection of interactive arrays that are visualised by one plot method (e.g. multiple lines or a scalar field with overlying vector field)
- **The Plotter class** is a collection of multiple formatoptions. Each plotter subclass is designed to visualise the data in a specific manner (e.g. via line plots, violin plots, or map plots). It is completely defined through it's formatoptions.
- **Formatoptions** are the core of the visualisation in the psyplot framework. Each plotter is set up through it's formatoptions where each formatoption has a unique formatoption key inside the plotter. This formatoption key (e.g. *title* or *cmap*) is what is used for updating the plot, manipulate the data or anything else. Formatoptions might also interact with other formatoptions inside the plotter or from other plotters. This concept of formatoptions allows to easily use the same formatoption with all different kinds of plotters and the interaction of multiple plots with each other. It also allows a very easy integration and development of own formatoptions on a low or high level of complexity.
  The number of formatoptions is unlimited. The mapplot plot method alone holds about 45 formatoptions to control the appearance of the plot.

Installation

The package is available through PyPi

```
$ pip install psyplot
```

and anaconda

```
$ conda install -c chilipp psyplot_gui
```

More information on
http://psyplot.readthedocs.org

Outlook

Further development in the near-term will cover

- a plotter for visualising shape files
- a plotter for making scatter plots
- an automatic movie creation
- widgets for a fast control of the formatoptions via the GUI
- Implementations of the unstructured grid conventions (UGRID)
- more test routines (especially for the GUI)
- more examples
- guide for implementing new formatoptions

Contact

Email: philipp.sommer@unil.ch
Web: http://arve.unil.ch
Address: University of Lausanne, Institute of Earth Surface Dynamics (IDYST), ARVE Group, Bâtiment Géopolis, 1015 Lausanne, Switzerland

References

- Hunter, J. D.: Matplotlib: A 2D graphics environment. In: *Computing In Science & Engineering* 9 (2007), Nr. 3, S. 90–95
- Met Office: Cartopy: a cartographic python library with a matplotlib interface, Exeter, Devon , 2010 - 2015