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 complicated 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 Environmet 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 kind 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 map 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 philipp 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