

Visual

Updated description coming soon!

Todo List:

- ☒ Make the y-axis in logarithmic scale.
 - ☒ Show the location of the cross-section
 - ☒ Able to scan along a front
 - ☒ Add better cmaps for variables (Namely, making small magnitude invisible.)
 - ☐ Use the mouse cursor to draw an area or a line for plotting.
- .
- .

Version 0.1.0

- Updated the tutorial.
- Minor improvements.

Version 0.0.9 (Timelapse)

- A new method for plotting timelapse on an isosurface.
- All planned features are now implemented, except the interaction with the mouse cursor.

Version 0.0.8 (curve)

- A new method is now introduced to draw curves (Meaning we can scan along a front now).

Version 0.0.6 (Pressure)

- A new feature on scanning the atmosphere through pressure levels.
- Now there are optional choices available for the plotfile, namely 'extend' and 'linestyle'.
- Optimized Visualization for cross-section.

Version 0.0.4 (Tracking)

- Output can now show the location of each cross-section
- (Specifically, the estimation file now returns two objects, one is the list of coordinates for the scanner, and another is the information of the plot used in estimation() and can be used in the scanner by setting prec = (your input). If "prec=" is not specified, the cross-section track will not show up.)

- Changed the default omega and pv color bar
-

Version 0.0.3 (Trial release)

- I am making the package to be more formal
 - First release with .py files
-

Version 0.2 (I know there is no actual versioning, will be added formally with the first .py file release):

- Making the plot's vertical axis in logarithm scale.
 - Adding a requirement of keeping pyproj version updated.
-

Version 0.1

- First release.
-

There are several files in this repo:

-/data: The place storing the data.

-/data/12z.grib: The file used in the notebook in .grib format (Retrieved from [ERA5 dataset](#)).

-README.md: The readme file (Or what you are looking at now).

-environment.yml: The environment file used in this project (The same as the AOS573 environment.) -

Scanner.ipynb: A notebook includes all the procedures and functions with more detailed explanations of the code.

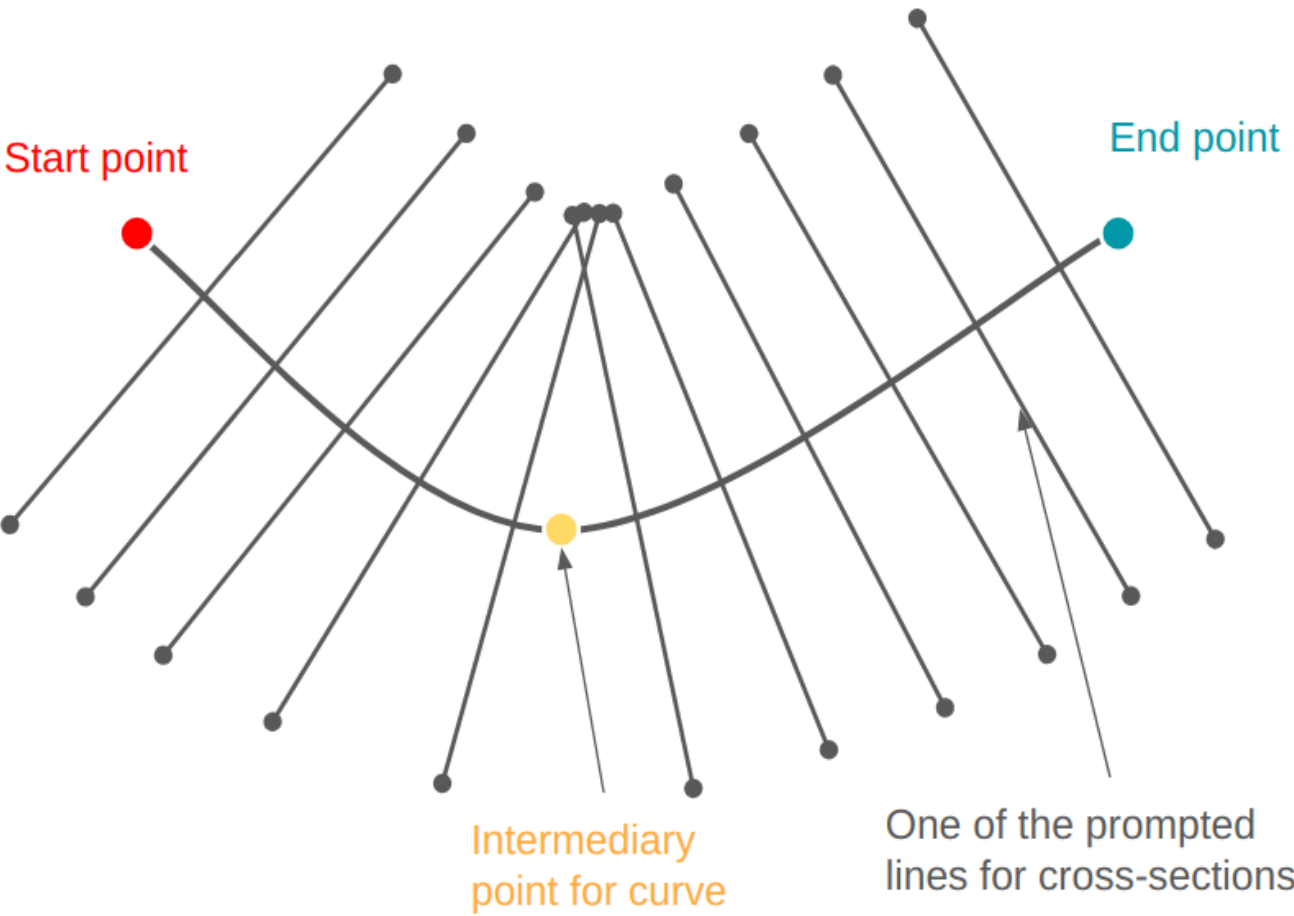
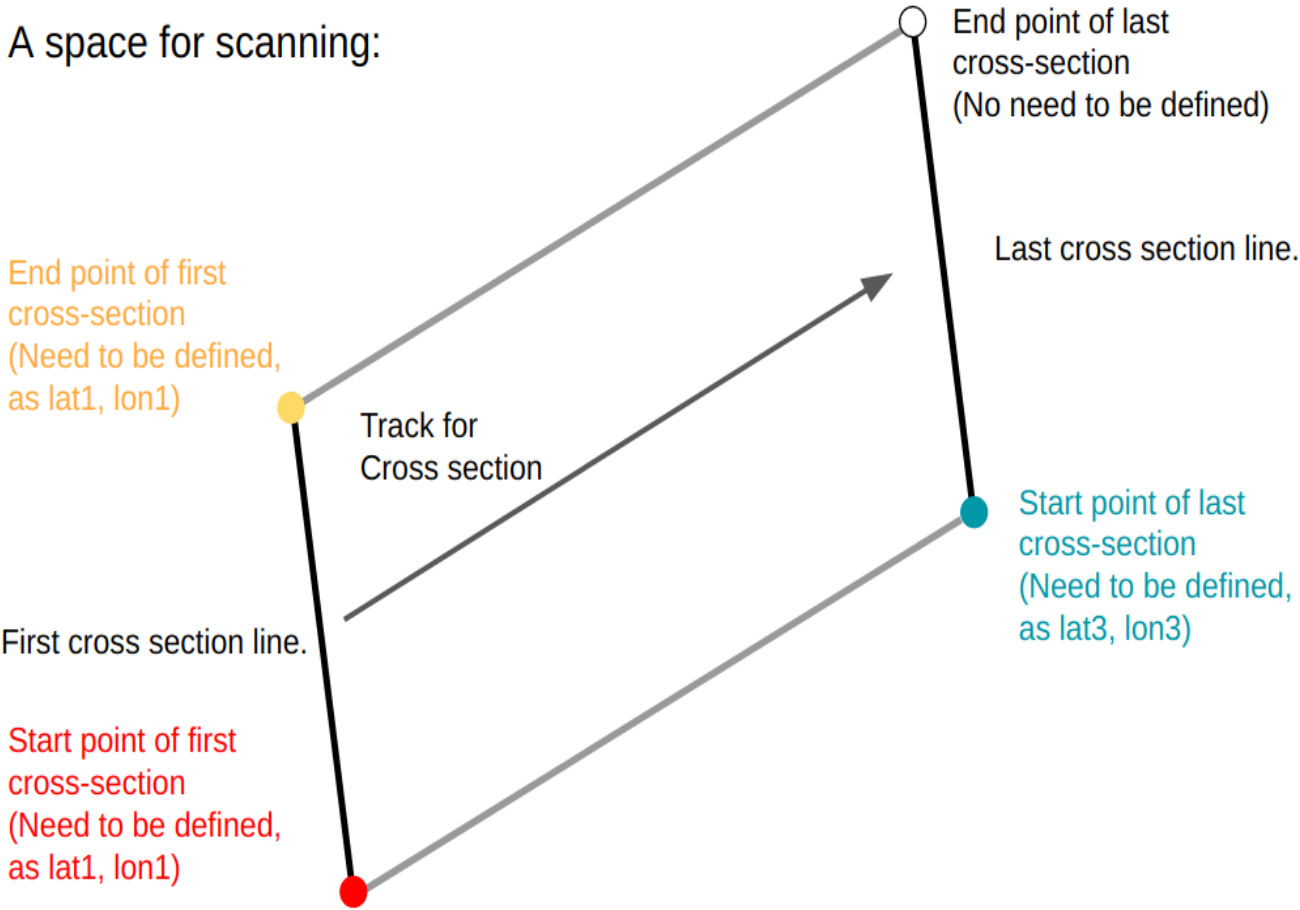
-plotfile.png & Function illustration.png: images embedded in markdown file.

-trial.gif: A gif as a result of running the whole notebook (as an alternate solution for Jupyterhub doesn't have ffmpeg).

-effect.mp4: A video generated by local jupyterlab with exactly the same function in the notebook in this repository (Again, since FFmpeg is unavailable to use here, this video is for people to see what the function should return if FFmpeg is working.)

One of the functionality in the package is to draw cross sections from the ERA5 dataset in two ways as shown by the following two figures:

A space for scanning:



Such a goal is achieved by the following:

First, present a base plot to choose the following points by running `estimation()`:

Then, run `scanner()` to generate a video, which plays the cross sections from the starting line to the final line.

A selection function(`selection()`) is also defined for extracting the desired dataset.

Other functions are defined for visualizing ERA5 data in other ways, such as plotting a timelapse on an isosurface, or plotting the atmosphere through atmospheric levels.

One important component of the notebook is a dictionary called **plotfile** that includes parameters for plotting inside of the `estimation()` and the `scanner()` function. Here is the plotfile used in the notebook (For new parameters, `level`, (`color&linewidth`)/`cmap`, and `title` are required, and the rest are optional.):

```
plotfile={'contour':{
    'thta':{'level': np.arange(250, 450, 3),
            'color':'red',
            'linewidth':1,
            'title':"Potential temperature (K)",
            'linestyle': 'dashed'},
    'z':{'level': np.arange(0, 150000, 60),
        'color':'black',
        'linewidth':1,
        'title': "Geopotential height (m)"},
    't':{'level': np.arange(0, 400, 3),
        'color':'black',
        'linewidth':1,
        'title': "Temperature (K)"}},
    'fill':{
        'vo':{'level': np.arange(5e-5, 40e-5, 5e-5),
            'cmap':plt.cm.YlOrRd,
            'title': "Relative vorticity(s^-1)",
            'extend': "max"},
        'pv':{'level': np.arange(-1e-6, 9.1e-6, 0.5e-6),
            'cmap':pv_cmap,
            'title': "Potential vorticity(PVU)"},
        'w':{'level': np.arange(-4, 4.1, 0.5),
            'cmap':w_cmap,
            'extend': 'both',
            'title': "Omega(Pa/s)}}}
```

and the structure looks like such:

```
plotfile={'contour':{ #Don't change the name "contour"
    'variable1':{
        'level': #A list generated by `np.arange()`,
        'color':#color used in matplotlib,
        'linewidths':#width of the contour,
        'title':#Used for the title of the plot
```

```
        },
        'variable2':{
            #...
            #...
            #...
            #...
        }}
'fill':{ #Don't change the name "fill"
    'variable1':{
        'level': #A list generated by `np.arange()`,
        'cmap':#colormap used in matplotlib,
        'title':#Used for the title of the plot
    },
    'variable2':{
        #...
        #...
        #...
    }}}
}}
```

Now, if you haven't seen the notebook yet, please take a look at it.