Mission Support System : Updating Geographical Plotting Routines

About Me

Name	Anveshan Lal
Github	https://www.github.com/funnelferry/
Email	anveshanrx8@gmail.com
IRC/Slack	funnelferry
Timezone	GMT +5:30

Education and Background

Currently, I am a freshman at Cluster Innovation Centre, University of Delhi pursuing Bachelors of Technology in Information Technology and Mathematical Innovation, I have done a variety of things involving python:

- GSoC Scrapper
- Reddit bot of a TV series 'Dark'
- Launched an alexa skill whose backend is written in python,

Other than that I also hone my coding skills at codewars.com, further, through my coursework I have also become familiar with C and MATLAB. I expect to graduate in the year of 2022.

Cluster Innovation Centre: https://www.ducic.ac.in
University of Delhi: https://www.du.ac.in
Location: New Delhi, India

Contributions to Mission Support System

Contribution	Issue number	Pull Request
Fixed Issue - Change comments for function "flightlevel2pressure_a"	#384	#606
Fixed Issue - Support of vertical cross sections beyond 30 km altitude	#379	#621
Reported Issue - Crashes on selecting Northern Hemisphere (stereo)	#388	-
Assisted in Issue - map zoom function on the user info does not work properly -> needs redraw	#364	-

Project Details

Sub-org: Mission Support System (MSS)

Abstract : MSS currently relies on the python basemap package for supplying non-cylindrical projections and plotting of geographical features with the use of EPSG Codes. This package has been deprecated and been supplanted in the community by the cartopy package.

In this project I will first completely change the existing code and replace the relevant parts by Cartopy, further optimization might be needed after which support for more geographical projections will also be added as mandated by WMS Standard.

Detailed description : At present the MSS project makes use of basemap for its plotting functions, this package has become obsolete. Cartopy with its superior API design is a viable solution, a code written in basemap:

```python
from mpl_toolkits.basemap import Basemap
import matplotlib.pyplot as plt
import numpy as np

map = Basemap(projection='merc', lat_0=57, lon_0=-135,
    resolution = 'l', area_thresh = 1000.0,
    llcrnrlon=-136.25, llcrnrlat=56,
    urcrnrlon=-134.25, urcrnrlat=57.75)

map.drawcoastlines()
map.drawcountries()
map.fillcontinents(color='coral')
map.drawmapboundary()

map.drawmeridians(np.arange(0, 360, 30))
map.drawparallels(np.arange(-90, 90, 30))

plt.show()
```

Can be rewritten in Cartopy as follows:

```python
import cartopy.crs as ccrs
import matplotlib.pyplot as plt
plt.figure(figsize=(5.12985642927, 3))
ax = plt.axes(projection=ccrs.PlateCarree(central_longitude=-35))
ax.set_extent([0, 50, -4, 10], crs=ccrs.PlateCarree())
ax.coastlines(resolution='110m')
gl = ax.gridlines(crs=ccrs.PlateCarree(), draw_labels=True,
            linewidth=1, color='gray', alpha=0.5, linestyle='-')
plt.show()
```

The zoom function demonstrated in above code can be controlled by changing the parameters of `ax.set_extent` to the desired value which will result in the map being cropped and zoomed accordingly while the same feature in the basemap written code will require you to change the values of `llcrnrlon, llcrnrlat, urcrnrlon, urcrnrlat`.

Majority of first part of this project will require to similarly rewrite existing code and making proper tests with update in documentation accordingly.

Further, The users require minimal rendering time for the maps provided by the web service. Thus, this work package shall measure the performance of the individual steps

of the cartopy solution and optimize the implementation, e.g. by using caches, as necessary.

The last stage will involve addition of new features specified in OGC Web Service standard currently lacking in MSS with the use of cartopy package. As of now, the code translates EPSG codes to Matplotlib basemap projection parameters which includes:

- EPSG:4326, the standard cylindrical lat/lon projection.
- EPSG:777llLLL, north polar stereographic projections with lat_0=ll and lon_0=LLL.

In the project I will also define a way to interpret the EPSG codes using Cartopy package and following that I will also focus on adding support for more types of projections like south polar projection

---

Milestones

1. Migration of codebase from basemap to cartopy
2. Completion of Test Suite and fulfillment of performance criteria
3. Implementation of missing OGC WS projections

Weekly Timeline

Upto May 7

- Make more contributions through issues and features to further my understanding of codebase
- Understand and write tests through pytest to be comfortable with it
- Be well versed with PEP8 style of coding

Community Bonding

- Thoroughly go through the documentation of Cartopy and Basemap
- Implement various functionalities of Cartopy standalone to get familiarity
- Learn PyQT framework into which matplotlib and Cartopy need to be embedded

Week 1

- Start rewriting basic functions and identify parts of code that need migration to Cartopy

Week 2-3

- Continue on to replace and rewrite code written in Basemap to Cartopy

- Update existing Tests and write from scratch for functionalities added through cartopy exclusively

Week 4

- Finish the migration of code
- Run tests and solve bugs that might have arisen during the change of package

Week 5

- Continue with writing tests and updating the existing ones
- Improve documentation and ensure it conforms to the newly introduced changes in codebase.

Week 6-8

- Clear any previous issues that had basemap addressed in them like #364
- Measure the performance of the individual steps of the cartopy code
- Optimizing the identified parts and make code modular

Week 9-10

- Begin expanding the support for more types of Geographical projections
- Testing the features through interface
- Write unit tests for new features

Week 11

- Continue and finish adding further support
- Making proper tests for all the work done
- Polishing the code for minor discrepancies
- Freeze Code

Week 12

- Update the whole documentation and ensure no part is left vague or undone.

Final Week

- Prepare final summary and organize the work into a presentable form

Communication

- I will be available on the IRC/Slack and the mail during the daytime.
- I will be regularly reporting to my mentor(s) about progress about the project and impediments that I may encounter.

- I am planning to write weekly blog posts about my progress on the project.

Commitment to Project

Since I have no other work commitment, I will be ready and make sure that I dedicate at least 5-7 hours a day to this project.

Exams and Other Summer plans

I have my university end-semester examinations from 8th May to 18th May. Other than that, I have no exclusive plans for this summer, I will stay at my home and dedicate most of my time to this project.

Why this project and Why I am best suited for it?

I have been fascinated with the world of open source since I was introduced to it, After looking at many projects I chose MSS as I found the codebase very approachable, also I was able to understand it fairly quickly, the whole concept of mapping was very familiar to me as I had previously done a project involving (GIS) ILWIS software which also involved use of different layers and plotting of maps. I think this project is right up my alley and will be an amazing learning opportunity for me this summer.

I think I am aptly suited for this project because I have the technical skill required to complete this project which is Python and Git with which I am enough experienced to tackle problems, Moreover I am willing to learn anything that this project demands and I believe within time I can pick up any skill, further being a freshman I will dedicate all of my time to this and though not necessarily but being new to development community I can also bring new perspective of doing things.

Are you applying for other projects in GSoC?

No, I am only applying for this project.