Mission Support System (MSS): Advanced CLI Plotting

About me

Name: Sreelakshmi P Jayarajan

Slack: swsrkty

GitHub: https://github.com/swsrkty

Timezone: GMT +5:30 (Indian Standard Time)

University: Government Engineering college, Sreekrishnapuram

Degree: Bachelor of Technology (B.Tech)

Major: Information Technology

Current Year: 2nd year

Expected Year of Graduation: 2024

Place: Kerala, India

Email: sreelakshmipj555@gmail.com

Link to resume: sreelakshmi_jayarajan_resume.pdf

I came to know about open source when I was in my first year of college. I was fascinated by it and I wanted to be part of a community solving an interesting problem. Since I only knew C++ at that time, I started contributing to SerenityOS, which is an open source operating system. I joined their community which had a lot of experienced people. After spending time understanding the codebase and getting some help from a few developers there, I opened 2 PRs. I wrote a blog post describing my journey as a beginner too. As I didn't get proper mentorship there, I found it a bit hard contributing there as a beginner and after some time, I realised that I need to gain more knowledge of operating systems and more experience in C++ to further contribute there. Then I decided to learn Python and built two projects: a TCP Chat Room using socket programming and an ISS-Tracker using a public API and PyQt. I also

looked into open source projects using Python and I came to know about MSS through my friend.

I find MSS interesting because of its application and how it helps researchers study about the processes in the atmosphere. It is fulfilling to contribute to an organisation which helps researchers with solving some of the global problems like climate change faced by people. Additionally, the people here have been really helpful throughout, the codebase is very approachable and I've got to learn a lot already.

The primary reason I chose the advanced CLI plotting project is because it would help users by making MSS more efficient. I would also get to learn a lot about MSS, build an interesting CLI tool with a GUI, learn best practices and learn how to work as a team. I look forward to contributing more to MSS, helping out new contributors, writing blogs about my wonderful experience here, being a part of the community and meeting the MSS team one day!

Contributions

- https://github.com/Open-MSS/MSS/pull/1399
- https://github.com/Open-MSS/MSS/pull/1339
- https://github.com/Open-MSS/MSS/issues/1181
- https://github.com/Open-MSS/MSS/pull/1383
- https://github.com/Open-MSS/MSS/pull/1301

Project information

Sub-org name: Mission Support System (MSS)

Abstract

In atmospheric research flight planning, one task that users have to perform is retrieving a set of plots for several flights or hundreds of

time-steps during post-campaign analysis or when compiling an overview over flights of a campaign during hindcasting. Another task is retrieving similar plots of the same parameters such as map section, level etc., on a daily basis. Both of these tasks cannot be accomplished in an efficient manner using the current MSS UI.

In this project, a feature would be built to let users download a number of plots in Jupyter notebooks and user scripts in an automated fashion according to the given settings. A CLI tool would also be provided for the same.

Detailed Description

The project can be divided into two main parts:

- Automated plotting module
- Building CLI tool using the automated plotting module

1. Automated plotting module

The MSS program has been designed for interactive, exploratory flight track planning. As such, it is focused on providing a wide set of display options to the user. This configurability quickly becomes tedious when a large number of plots shall be generated repeatedly in a standardised fashion due to the amount of clicking and configuration necessary.
The retriever.py script is a prototypical answer to this problem. It can be used to create plots without using the user interface. It loads details such as flights, layer names, pressure, time, etc., from the configuration file and produces top and side view plots in an ad-hoc manner. Concepts of the retriever.py script and functionalities after refactoring the MSS codebase would be used to build an auto plotting module, which would allow plotting graphs of top, side and linear view as a configured file format, e.g. png. The module can be used in user scripts and Jupyter notebooks, and is also used to build the CLI tool.

2. Building CLI tool using the automated plotting module

This tool (built using Click) would be used to download a number of plots according to the given settings. The settings would include the following parameters:

- View
- URL of the WMS server
- URL of mscolab server
- List of layers to be plotted
- Style
- Elevation
- List of initial time and forecast hours
- List of flight paths
- List of operations (from MSCOLAB)
- Level
- Map section
- Resolution of the plot
- List of shortened layer names to be used as the plots' titles
- Naming convention of output plot files
- Output file format
- Path of output folder

The user would enter the default values for all the parameters above into a new config file. The user can also maintain multiple config files. The path of the required config file needs to be passed as an argument to the CLI tool, which would read the default settings from the config file unless the user passes some arguments to overwrite those parameters' values.

```
mss autoplot --config_path mss_autoplot1.json
```

When the tool is used in the above way, the user would be able to download the required number of plots with the default settings into the output folder.

There are currently two major use cases for this tool:

Creating a daily "standard set" of plots with/without an actual flight track, e.g., for daily morning briefings

```
mss autoplot --config_path mss_autoplot1.json --time
2022-03-29T00:00:00 12
```

The above command will download all the plots configured in mss_autoplot1.json from initial time 2022-03-29T00:00:00 upto valid time 12 hrs.

Creating a "standard set" of plots for all actual flights of a campaign

```
mss autoplot --config_path mss_autoplot1.json
--ftrack example1.ftml --ftrack example2.ftml
```

The above command will download plots of the given flight tracks. The user can also change the flight tracks by directly editing the config file. If the user wants to use an operation from a mscolab server, they can set the mscolab URL in the autoplot config file and use the CLI tool as follows:

```
mss autoplot --config_path mss_autoplot1.json
--operation example1 --operation example2
```

All the methods in this module would be thoroughly tested and documented with examples and tutorials in the best way possible.

Weekly Timeline

I would like to go for the extended timeline, i.e. from May 21 - November 21 since I have regular classes throughout the year and no vacations (because of schedule changes due to covid). My estimated timeline to complete the project would be as follows:

Week	**Work**
May 20 - June 12	Community bonding period ● Discuss with mentors about the project ● Understand retriever.py and parts of the MSS codebase which is related to this feature in detail
(Week 1 - Week 12) June 13 - September 02	Autoplot module ● Build the autoplot module with functionalities for top, side, table and linear views, refactor parts of the MSS codebase and manually test the module in Jupyter notebook.
September 02	**Phase 1 evaluation deadline**
(Week 13 - Week 14) September 03 - September 17	Testing autoplot module ● Add tests and documentation for the auto plotting module.
(Week 15 - Week 21) September 18 - November 4	Building CLI tool ● Build CLI tool structure. ● Add functionality to read configuration from the config file. ● Extend the CLI tool to handle arguments from the user and configure the settings accordingly. ● Add tests and documentation.
(Week 22) November 5 - November 13	Tests and documentation ● Work on improvements, bug fixes and ensure everything works well. ● Work on remaining tests and documentation.
November 14 - November 21	Final work report ● Write the final work report.
November 21	**Final work report deadline**

Stretch goals

1. GUI enhancement and integrating with MSS UI

After building the CLI tool, a GUI will be built and integrated with it. The user can enter the path to the config file and optionally enter custom parameters to overwrite default parameters, instead of passing arguments in the CLI tool. The GUI would contain a section to configure the set of layers, a dropdown to set time steps and text fields for the rest of the parameters. The GUI would reuse the parts of the refactored MSS GUI, which would lead into the next package. The GUI can be accessed as follows:

```
mss autoplot --gui
```

This feature would also be integrated with MSS UI, where it would be added as a menu entry in the main menu to let the users access the GUI tool for automated plotting.

2. Add tutorials and extend documentation

Tutorials would be added for the users to quickly understand how to use the feature along with extending the documentation.

3. Add support for multilayering

The autoplot module would be extended to include the multi layering feature in which the user can specify a number of layers in the config file along with the other parameters to download a multilayered set of plots according to the given settings.

Commitments

- MSS is the only organisation I’m applying to.

- Although I would have classes, I assure that I'll work on MSS for at least 5 hours daily and 8 hours on weekends.
- I don't have any other commitments or summer plans during the GSoC period. In case something comes up, I will make sure to inform my mentors as soon as possible.
- I look forward to working on more interesting projects at MSS, being part of the community and mentoring future contributors after my GSoC.

Communication

- I will be available on Slack and on email.
- I will be regularly communicating with my mentors about my progress in the project, or about any issues that I may encounter.
- I'm also planning to write weekly blog posts about my progress and the things I learnt about the project.
- I would also be happy to join any discussions/video calls during daytime.