Software User Guide - Investigating Ionospheric Plasma Drifts and Thermospheric Neutral Winds Using CubeSats

Rifat Mahammod - Supervisor: Dr. Aruliah Anasuya

University College London,Gower Street, London WC1E 6BT, United Kingdom

E-mail: zcapmah@ucl.ac.uk

Contents

1 Introduction

This document describes the implementation of the software used in this project in order to analyse and evaluate the INMS data. This report aims to provide a step-by-step guide on how to reproduce some of the visual results concluded in this project. All the programs and datesets can be found in the "Rifat's project" folder and in the subsequent subfolders. The complete directory is shown in figure 1. The INMS textfiles are located in the "CorrectINMSdata" folder. The "Plots" folder contains some of the project subplots and figures.

The project is mainly written in MATLAB. The main.m file merges all the other MATLAB files and runs them to produce self consistent plots and figures. Individual functions were created in separate files, each with a different purpose. Each function is given an appropriate name and within the scripts, brief commented introductions are given ahead of the main lines of code, explaining the expected output. Most of the scripts use MATLAB inbuilt functions. For more details about these functions you may use the following link, replacing FUNCTION with the appropriate function name: https://uk.mathworks.com/help/matlab/ref/FUNCTION.html. One function that is frequently used in this project is the "atmoshwm" inbuilt function. This is a function from the Aerospace Toolbox, which calculates the meridional and zonal components from the Horizontal Wind Model (HWM). For more information on this function and the relevant toolbox please refer to https://uk.mathworks.com/help/aerotbx/index.html?s_tid=CRUX_lftnav.

Please note that the project uses MATLAB R2020b. Older MATLAB versions may not execute some of the scripts properly.

2 Data and Programs

As demonstrated in figure 1, the "Rifat's project" folder contains all the programs and datasets. Within that folder, there is a python file, a word document and few excel files. These files are not essential as they only contain a mixture of evaluations and calculations. The important files, however,

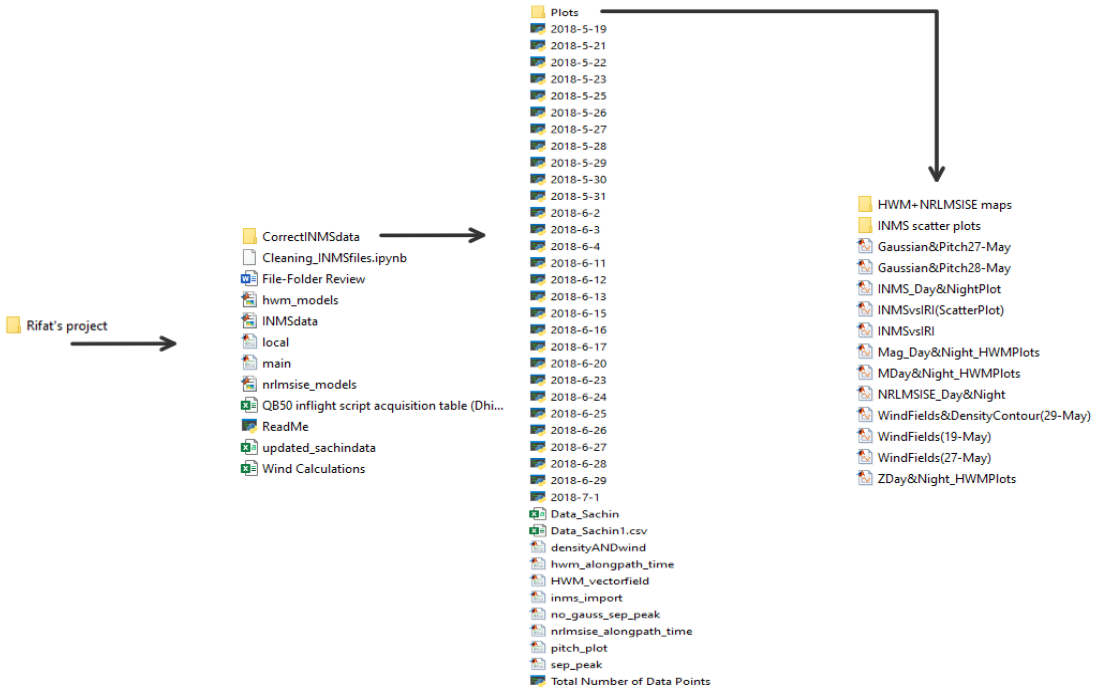

Figure 1. How to download the required modules/add-ons in order to run the scripts in this project.

are the MATLAB files. The main.m merges other custom functions and allows them to execute all at once, to produce comprehensive plots and figures. The other MATLAB functions are in the .mlx format. These are MATLAB notebooks, which were used to analyse and evaluate the INMS data. All the results can be inspected by opening those files in the editor. Please note that these notebooks are exploratory files that contributed to my results, although certain sections of code may present errors. It is advised to avoid running the code in full using the run button under the editor tab, but to execute the code section by section using ctrl+enter. In the "CorrectINMSdata" subfolder, all the INMS datafiles are saved in textfile format. The excel files contain some INMS counts data that is used in the MATLAB notebooks located in the previous directory. The rest of the files are MATLAB functions used for this project's data analysis. Each of these will be looked into in more details in section 4. The final subfolder, "Plots", contains saved figures in MATLAB'S .fig format and two folders with saved JPEG figures.

3 Prerequisites

In order to run this project a MATLAB development environment of version R2020b or higher is required. It is a requirement to download the following MATLAB Add-Ons from the Add-Ons drop down menu under the Home tab, as shown in figure 2:

- Curve Fitting Toolbox
- Mapping Toolbox
- Aerospace Toolbox

- Aerospace Blockset
- Symbolic Math Toolbox

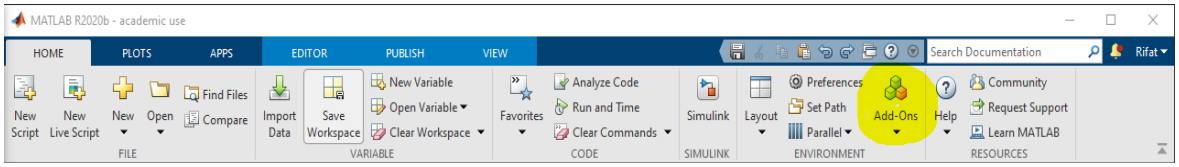

Figure 2. How to download the required modules/add-ons in order to run the scripts in this project.

4 Functions

In addition to MATLAB inbuilt functions, custom functions were also created. Brief descriptions of these functions are given in the following subsection of this report.

4.1 inms_import.m

This function imports the INMS textfiles and plots Gaussian distributions on the scattered INMS data. It takes the file name in date format 'yyyy-m-d.txt' as an input. For example, if interested on the data from the 27th of May, execute inms_import("2018-5-27.txt"). While the function iterates over the time column and INMS count column, zero count errors are removed. The INMS counts are then plotted against time to produce a scatter plot fitted by Gaussian distributions.

4.2 no_gauss_sep_peak.m

This function is very similar to the first one, but the only difference is that it uses array slicing to separate each peak for a given full run of counts. This function was utilized to isolate the high geometric factor peak in order to make comparisons between different dates.

4.3 HWM_vectorfield.m

This file computes the HWM vector fields using the 'atmoshwm' inbuilt function. In the same way as previously mentioned, the function extracts all the columns from the textfiles. The atmoshwm function takes in parameters such as latitude, longitude, time, date and Ap index. Once the extracted parameters were passed through this function, two matrices were filled to produce quiver plots. On top of these vector fields, the location data was also plotted to visualise the path of the CubeSat.

4.4 densityANDwind.m

This function produces plots of the HWM vector field overlaid on top of the NRLMSISE Oxygen density contour map. The function computes the NRLMSISE Oxygen densities by using the 'atmosnrlmsise00' inbuilt function from the Aerospace Toolbox. This takes the location, time and f10.7 data to compute a list of particle densities. The oxygen density is then extracted from the array and plotted on a meshgrid to produce a contour map. The HWM_vectorfield function is executed to overlay the vector field and the CubeSat path onto the same graph.

5 Running The Scripts

This section describes the necessary steps required to run the function scripts.

1. Open MATLAB.
2. Navigate to the directory where ”Rifat’s project” is saved.

If interested in running individual functions on their own, such as the ones mentioned in the previous section, first navigate to the directory in which the function is saved, either manually or using cd. For example, if you want to execute HWM_vectorfield.m to produce wind fields for the data on the 27th of May 2018, use the run() command in the command window, as shown below:

- run(HWM_vectorfield(”2018-5-27.txt”, 3))

where the second argument (3) is a chosen Ap index. This is shown in figure 3. Similarly, for the other functions you can execute them using the same method, but being careful to change the name of the function and its respective arguments. To find out more about the function itself double click on the file, under the ”Current Folder” window, as demonstrated in figure 3.

5.1 Running the main.m script

The main.m file is a script that merges all the individual functions into a main file. If executed, all the plots and figures for a certain given dataset are produced simultaneously. The main.m file also runs in a similar fashion to the method mentioned earlier.

1. Navigate to the directory where main.m is saved.
2. Open the file by double clicking it.

   There are two ways to execute this function. The easiest method is to press the run button under the editor tab, as shown in figure 4. The other method is to use the run command from the command window in the following way:

   - run(”main.m”)

   It is extremely important that the current directory is the same directory as the main.m file, otherwise the function will not run.
3. Upon pressing enter, a prompt on the command window will appear displaying ”Please Choose date with format ’yyyy-m-d.txt’:”. This is asking the user to choose the dataset. All the textfiles containing the location, time and INMS counts are saved in the ”correctINMSdata” subfolder. Each of these files correspond to the oxygen counts collected by the CubeSat during each day. The naming format of each file states the date on which the counts were gathered. The files can be seen in the left hand side of figure 3, under the Current Folder window.
4. Once a date is chosen out of the available files, use the requested format to run the script. For example, if you want to analyse the data for the 27th of May 2018, type ’2018-5-27.txt’ and press enter. You must include the speech marks around the file name. This is shown in figure 5.

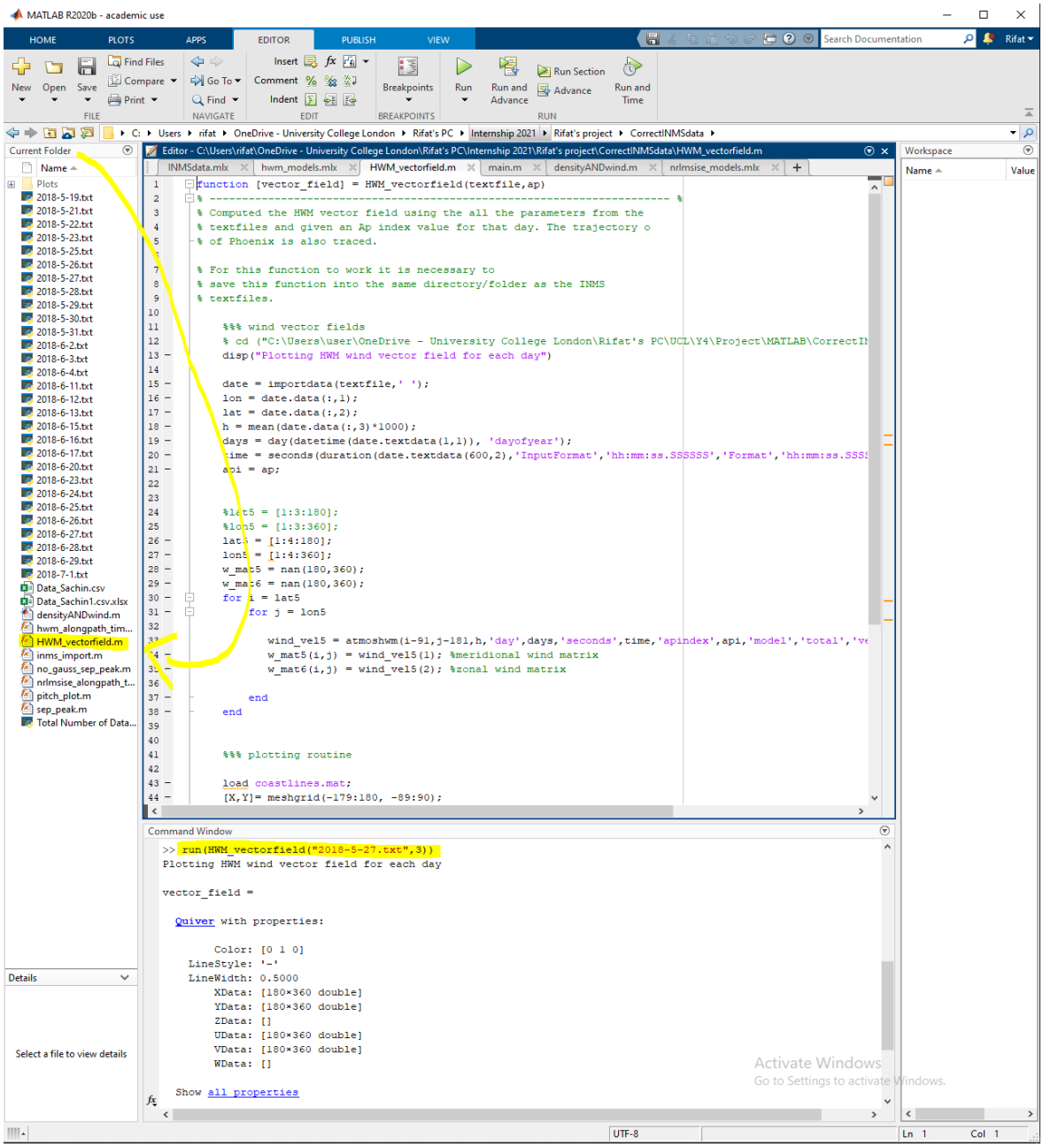

Figure 3. Demonstrating the location of the file in the current folder as highlighted, and how it is executed through the command line.

5. The program should run for no more than 60 seconds. It will produce 7 different figures. The first two figures displaying INMS scatter plots, should pop up instantaneously in separate windows. The other more complicated plots, such as vector fields and contour maps, will follow.

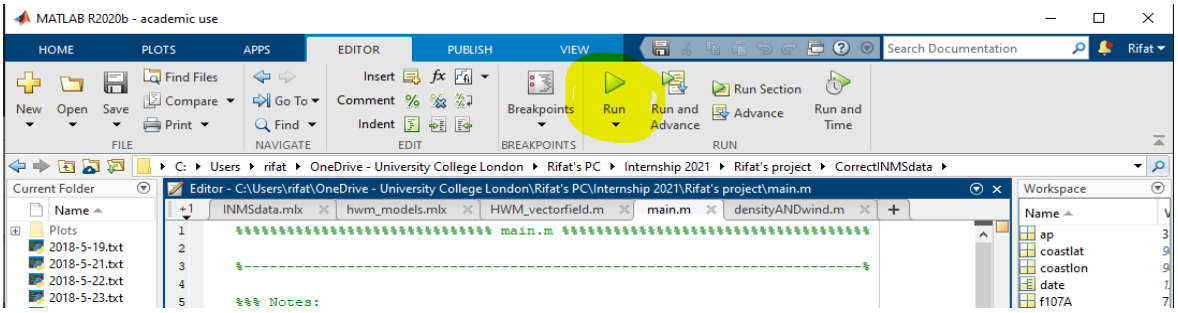

Figure 4. Showing the run button on the MATLAB editor.

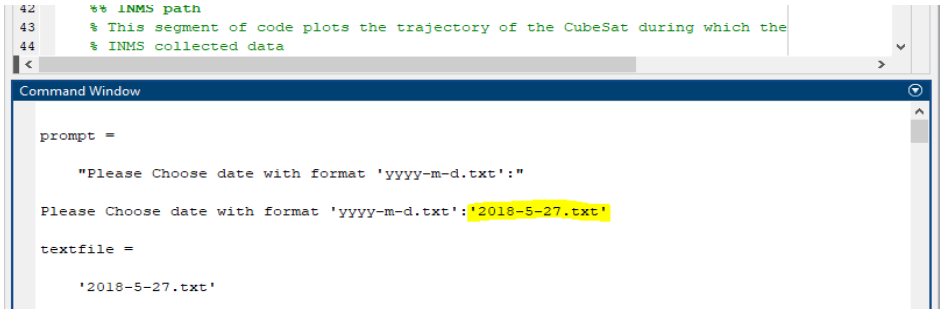

Figure 5. Demonstrating how to run the main.m using the command line

6 Errors and Warnings

Frequent warnings may involve the following ones:

1. Warning: Inputs must be character vectors, cell arrays of character vectors, or string arrays.
2. Error in run: Input must be a row vector of characters, or a string scalar, or a cellstr, or a string matrix.
3. Error in run: [fileDir,script,ext] = fileparts(scriptname);

These warnings can usually be ignored. However, to overcome this, individual functions can be executed without the run command. Simply navigate to the function directory and enter the command in the command window, as shown in the example below:

- HWM_vectorfield(”2018-5-27.txt”,3)

Additional errors may arise for not being in the current working directory of the file being executed. For example, if you want to run main.m, your current folder window must be in the same directory as your main.m file. This is explained visually in figure 6. The error that is displayed on the screen may look like this:

1. ’densityANDwind’ is not found in the current folder or on the MATLAB path, but exists in:

   `C:\Users\rifat\OneDrive-UniversityCollegeLondon\Rifat'sPC\Internship2021\Rifat'sproject\CorrectINMSdata`

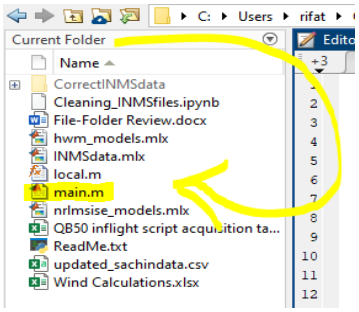

Figure 6. Demonstrating the current folder window.

A trivial but important step that needs to be considered when executing the programs in this project is that the functions have all been adapted to the directory of the creator. You must check every script and identify every lines of code which begin with cd. This is the MATLAB ”change directory” command. It is required to adapt the code to the directory in which the ”Rifat’s project” folder is saved in the user’s computer.

Therefore replace `C:\Users\rifat\OneDrive-UniversityCollegeLondon\Rifat'sPC\Internship2021\Rifat'sproject` with your own directory. If this step is not completed before running any file, then the following errors are going to appear in the command line:

1. Error using cd: Cannot CD to ... (Name is nonexistent or not a directory).