Visualization of data from the HOPE spacecraft

Prisha Jain

June-July 2022

1 EXI Data

1.1 Plotting the EXI data and opening .fits files

The initial tasks were to learn how to open a .fits file, and look at the image data contained in these files. This was done using python, and also cross verified using the FV viewer, which can be used to open and view the data contained in a .fits file in a very simple manner. Apart from opening the files and plotting the data, small tasks like zooming in and out were also done and the code for the same can be found in the notebooks.

1.2 Producing RGB Composites

The next task was to produce RGB composites to obtain an actual image of Mars. The data we had were images from 3 filters: red, blue and green filters. There were different modes of the data of all these images. The two modes from which we obtained the images mostly were the XOS1 and XOS2 mode. We scaled the values of the 3 channels from 0 to 255, relative to the maximum of all 3, and we assigned weights to the R, G and B components of the images obtained from the 3 filters to combine them all and produce the final RGB composite. The weights were different for different times of the days, but turned out to be nearly the same for the same time on different days and images taken from the same mode. The Q and stretch values were varied as well to produce the final images. An example of the kind of values we used for the different parameters of the python function "make_lupton_rgb" is as follows:

- Weights of R,G and B:

  R =1

  G =0.95

  B =0.8

- Q = 1
- Stretch = 1

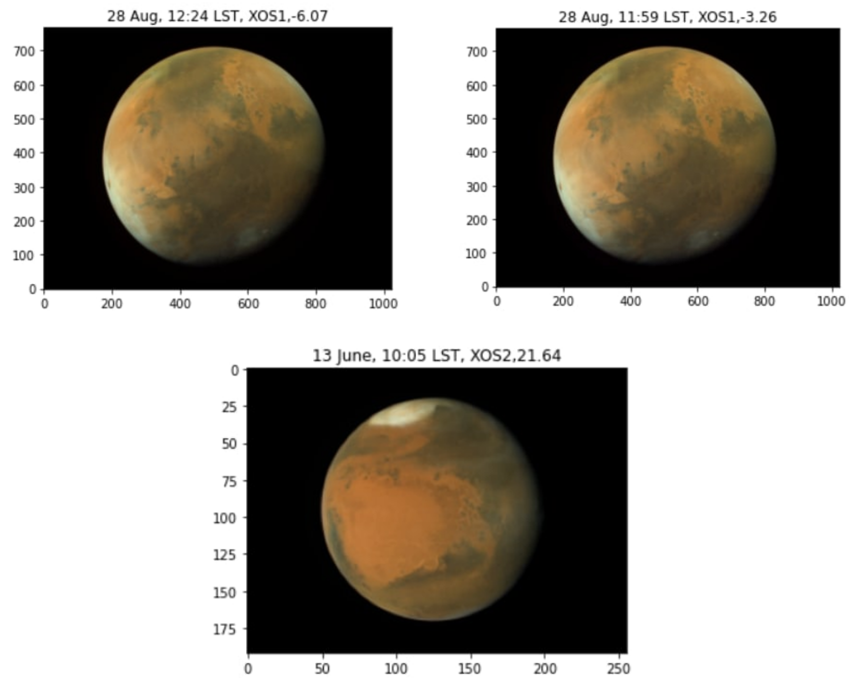


Figure 1: RGB Composite images of Mars

Some images of the data plotted with these values are shown in Fig 1: (The title on the plots follows this naming convention: Date on which the image was captured, Local Solar Time, Mode of the image, SC Latitude) A few observations about the images captured using different modes:

- In the XOS1 mode, the numpy array which represents the image from each filter don't all have the same shape. Their shapes are as follows:

  Red : 1536 x 2048

  Blue : 768 x 1024

  Green : 768 x 1024

- In the XOS2 mode, the dimensions of the numpy array representing the images from different modes were as follows:

  Red : 256 x 192

  Blue : 256 x 192

  Green : 256 x 192

The full disc images of Mars were obtained when images with LST around 12 noon were combined together. Using the data for one particular day (28 August

2021), we also made an animation of Mars by combining the different images (taken from around the same LST, which was around noon)

1.3 Plotting the figures in correct orientation

One thing to think about was, the plots we are showing, how do we know they are with the correct orientation? (that is, how do we know that the parameter “origin” in the plt.imshow function is to be given the value of ‘lower’ or ‘upper’?) Upon plotting a few images, it seems that the images plotted with origin as lower and flipped images plotted with origin as upper represented the correct plots.

In the code, if you input all the files of the data taken from the red, green and blue filters, you can obtain an image of the planet, in which you can see the north pole of the image in the upper half of the image (done for the sake of convention, we often like to view the north pole in the upper half of the image) So this code would orient the images correctly by setting the value of the parameter “origin” appropriately. The logic behind this script was as follows: We plot the data from the LAT of any one of the filters, and locate the north pole and the south pole using this image. The point where we get the highest value (against the color bar) is where we have the North pole and the lowest value point corresponds to the south pole. (See fig. 2 For reference)

```
In [52]: plt.imshow(lat,origin='upper')
         plt.colorbar()

Out[52]: <matplotlib.colorbar.Colorbar at 0x2083a0afbe0>
```

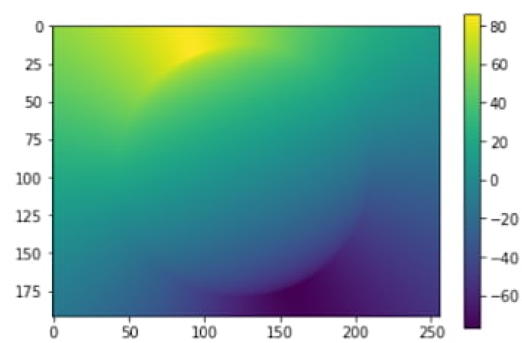


Figure 2: Plotting the latitude with the colorbar

Through LAT data image, we can find out where the north and south poles are when the origin is kept as the one kept in the LAT data image. As shown in Fig.2, when the origin is “upper” ,in Fig.2, the north pole (yellow) is in upper half of the image. So if the row number of the highest value in array is less than half the number of total rows, the north pole lies in the upper half of the image, so we will plot the original image with the origin as “upper”. In the other case (where row number of the highest value in the array is more than half the number of total rows), we will plot the data with origin as lower. Fig.3 provides an example of the same

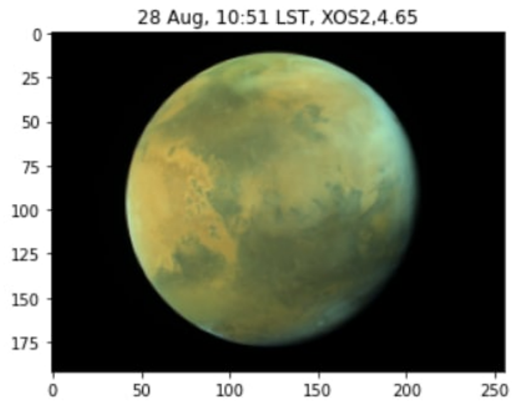


Figure 3: Plot obtained with north pole in the upper half of the image, using the code in Python notebook

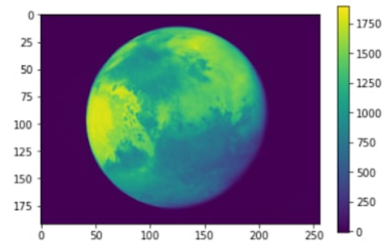
1.4 Plotting different parameters for the EXI data

- **SCI**
  Represents images taken through the blue, green and red filters. When these 3 images are combined, we can obtain the RGB composite of Mars, which represents how the actual planet would look like. (Fig.4)

- **DQF**
  Scientific data may comprise data-points of very different quality for different reasons (e.g. sensor malfunction, failed calibration, high measurement error). For their analysis it is essential to classify data-points, for example, as reliable, questionable, or even wrong value. This can be done by assigning numerical codes known as data quality-flags. (Source: google) (Fig.5)

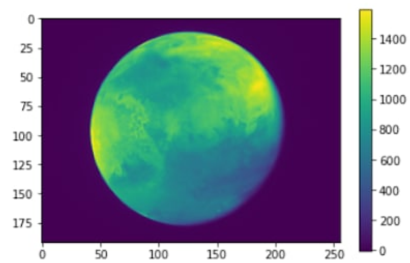
```
In [11]: plt.imshow(sci_r)
         plt.colorbar()

Out[11]: <matplotlib.colorbar.Colorbar at 0x2c06adeda00>
```




```
In [12]: plt.imshow(sci_g)
         plt.colorbar()

Out[12]: <matplotlib.colorbar.Colorbar at 0x2c06aebc3d0>
```




```
In [13]: plt.imshow(sci_b)
         plt.colorbar()

Out[13]: <matplotlib.colorbar.Colorbar at 0x2c06af79e50>
```

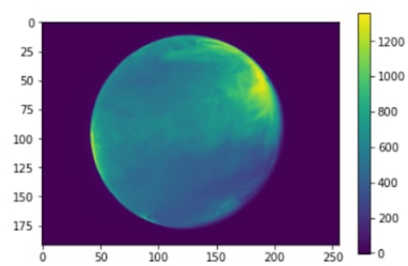


Figure 4: SCI data through Red, Green and Blue filters (in order)

```
In [15]: plt.imshow(dqf_r)
         plt.colorbar()

Out[15]: <matplotlib.colorbar.Colorbar at 0x2c06b090e20>
```

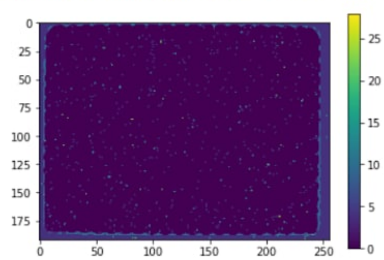


Figure 5: DQF data

```
In [21]: plt.imshow(pha_b)
         plt.colorbar()

Out[21]: <matplotlib.colorbar.Colorbar at 0x2c06b4a57f0>
```

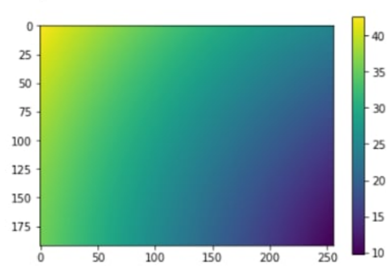


Figure 6: PHA data

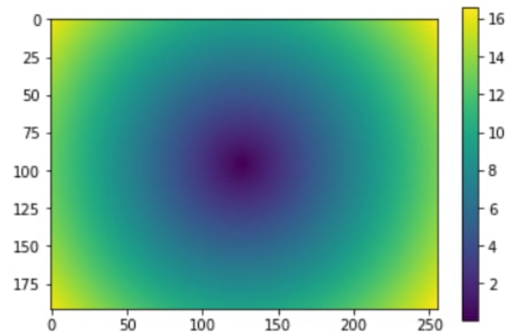


Figure 7: LOOK data

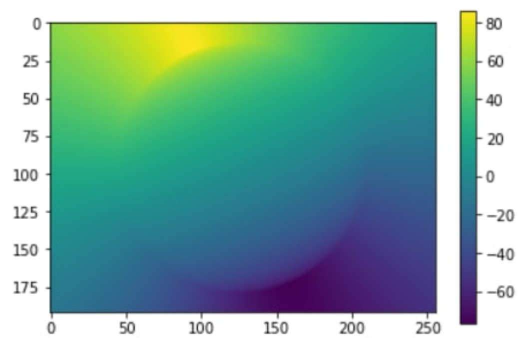


Figure 8: Latitude data

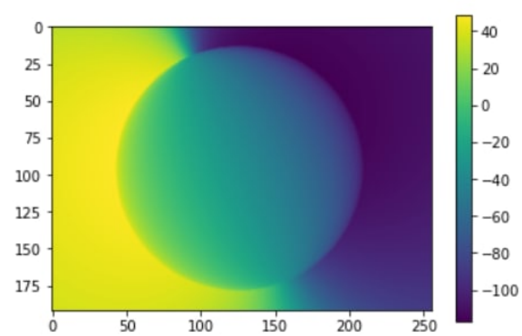


Figure 9: Longitude data

Data from various parameters (Fig.4 to Fig.13)

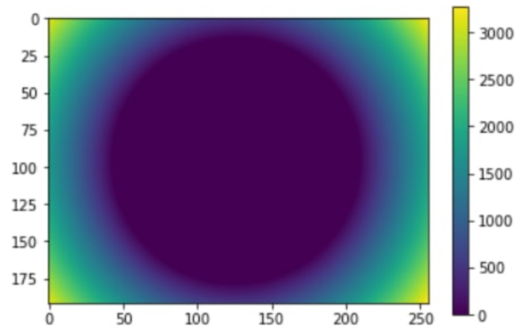


Figure 10: Height data

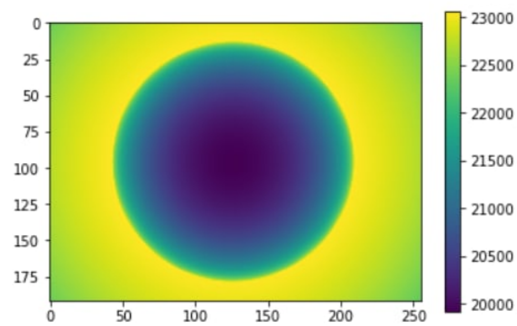


Figure 11: Range data

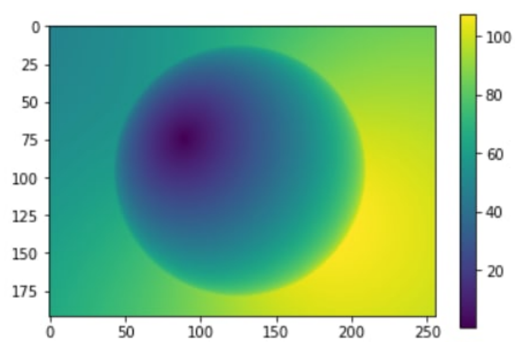


Figure 12: INA data

Data about Height, Range and INA

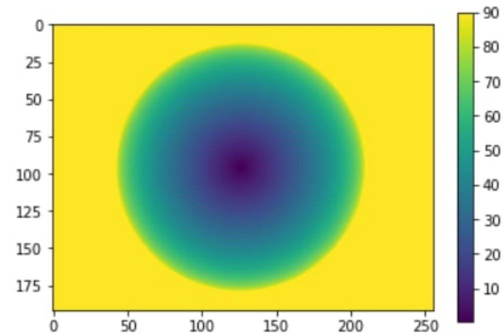


Figure 13: EMA data

2 EMUS Data

2.1 Plotting the EMUS data

Data in Fig.14 is plotted without normalization/ square root scaling

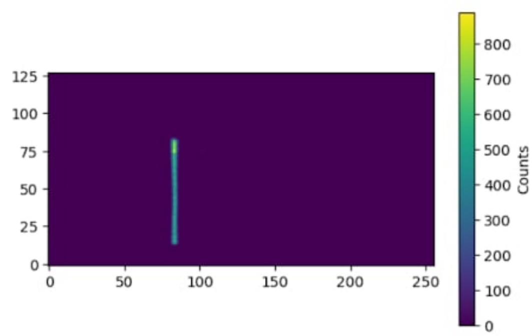


Figure 14: Data from 20th February 2021, 04:42 LST, l2b data, plotted without normalization

Data in Fig.15 is plotted with square root scaling normalization, this kind of normalization helps to observe very faint spectral lines as well.

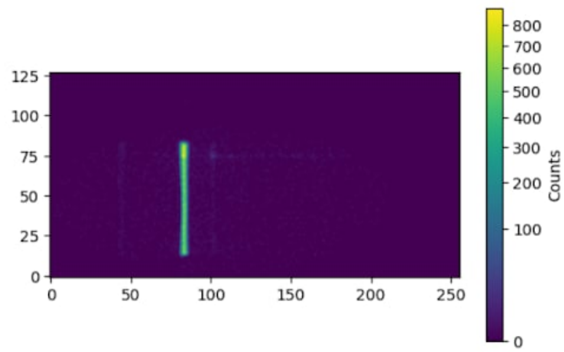


Figure 15: Data from 20th February 2021, 04:42 LST, l2b data, plotted with normalization

Next, we selected one spatial row to examine the spectrum produced by EMUS

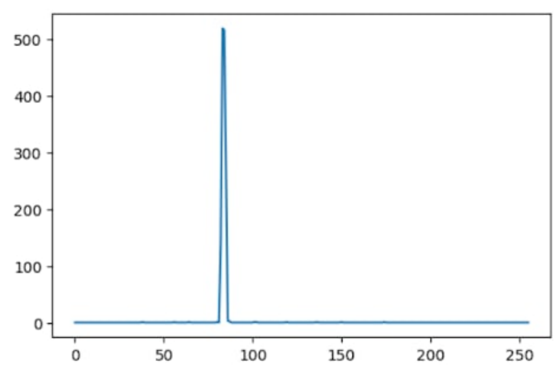


Figure 16: Spectrum produced by EMUS

But in the plot shown in Fig.16, we are unable to observe very small peaks clearly. So to deal with that, we zoom in or use different scales to enlarge data (as shown in Fig.17)

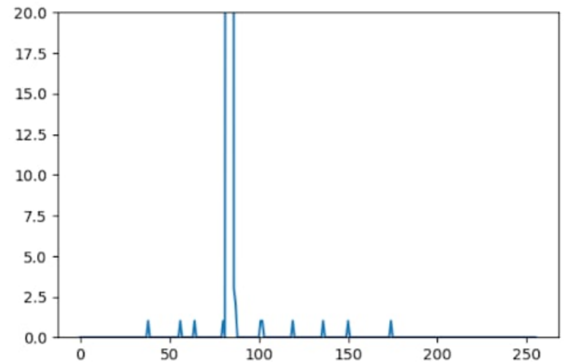


Figure 17: Using different scales to enlarge the spectrum

Still, in these images, the signal to noise ratio is not close to ideal, so we make it better (as shown in Fig.18):

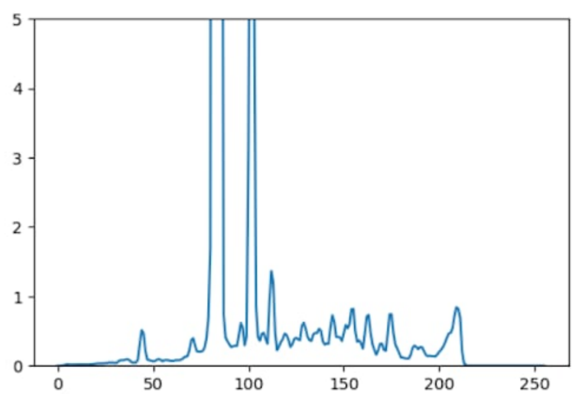


Figure 18: Using different scales to enlarge the spectrum

2.2 Plotting the brightness plots of the planet at a specific wavelength

Plotting disk images of strong and faint lines for one file: (Fig.19 and Fig.20) Any mode captures a particular wavelength more strongly than the other wavelengths, thus we have strong and faint spectrum produced when we do plots for different wavelengths for the same file.

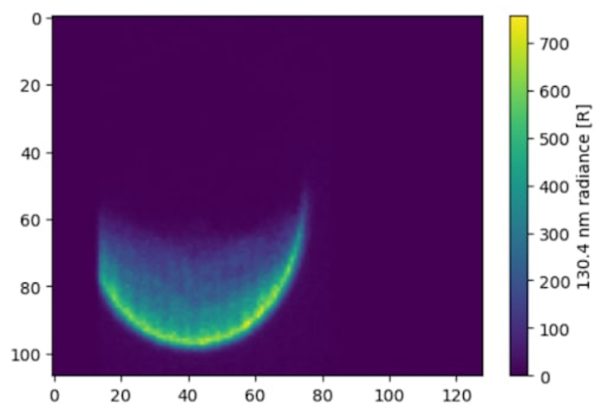


Figure 19: Plot for wavelength 130.4 nm, Faint

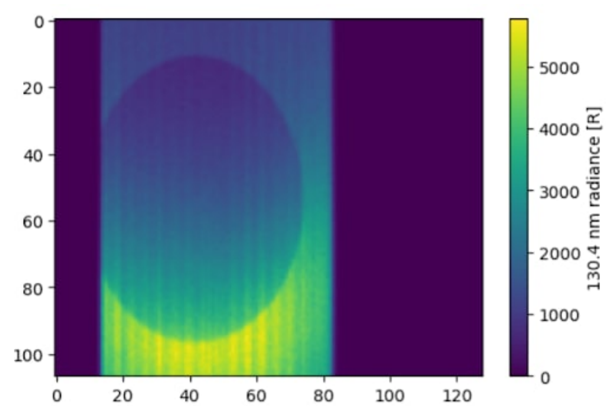


Figure 20: Plot for wavelength 121.56 nm, Strong

2.3 Combining (stacking vertically/horizontally) different data images to obtain the full disk

Some examples of images produced by combining data from different files (which were not consecutive) are shown in Fig.21

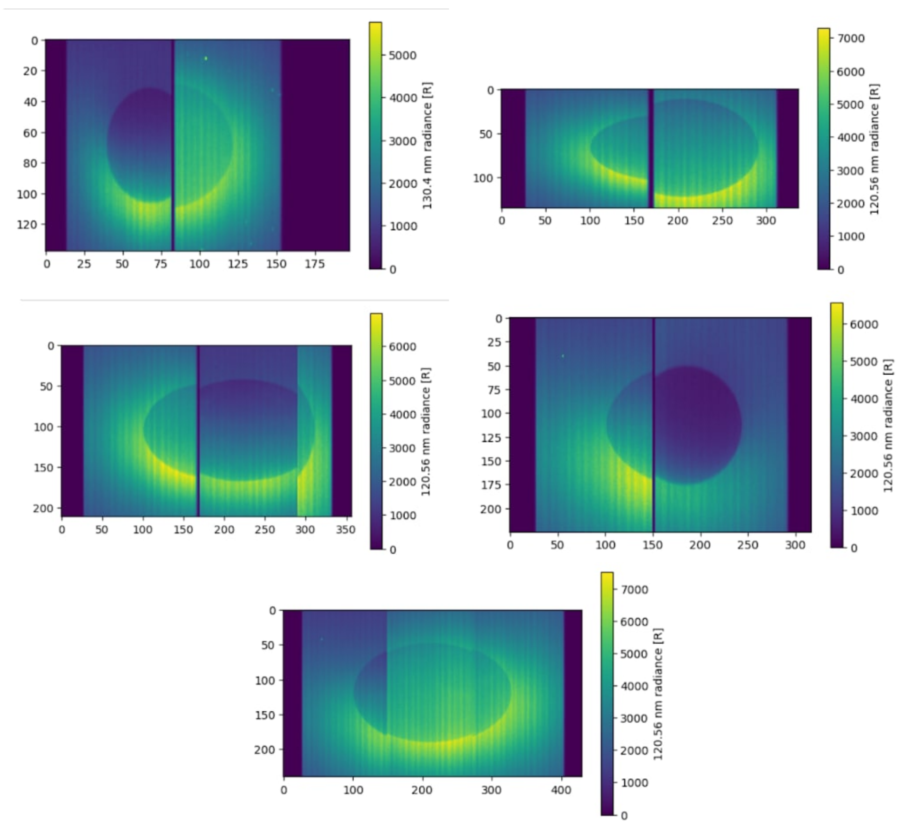


Figure 21: Images produced by combining different files' data

2.4 Plots of data from different modes

Plots of data from the modes mentioned in the list below are shown in Fig.22 to Fig.30 along with the details of other parameters.

1. OS2 Mode
2. OS3b Mode
3. OS4a Mode
4. OS4b Mode
5. OSR Mode
6. EMU035 Mode

(We have not included examples for the OS1 Mode and OS3a Mode data. They can be found in different sections of this report)

We also produced images of the disk by combining data from different files (which were not consecutive). They are shown in Fig.31 , Fig.32 and Fig.33

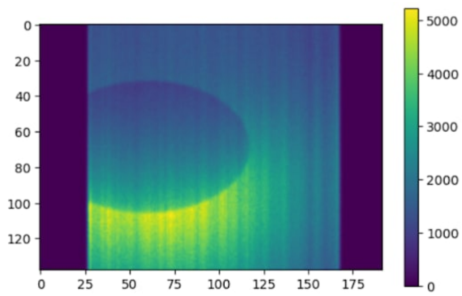


Figure 22: This image had the following parameters:
Date : 2 Feb 2022
LST : 18:59
Mode: os2
Altitude: 41823 km
Orbit number: 167

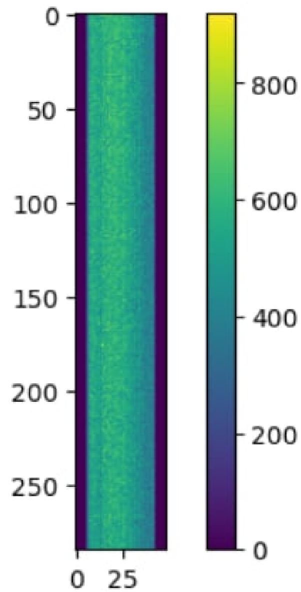


Figure 23: This image had the following parameters:
Date : 7 Feb 2022
LST : 18:59
Mode: os3b
Altitude: 28812 km
Orbit number: 169

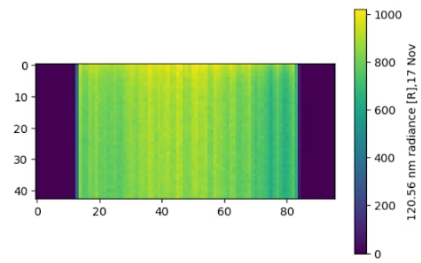


Figure 24: This image had the following parameters:
Date : 1 Feb 2022
LST : 09:26
Mode: os4a
Altitude: 22589 km
Orbit number: 168

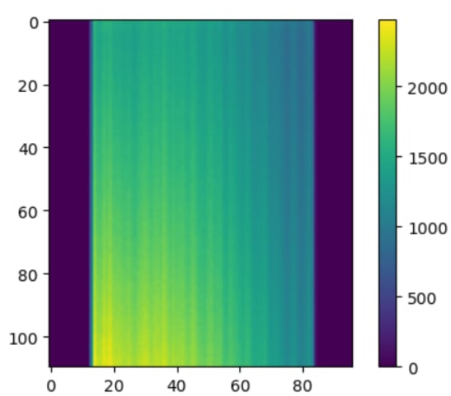


Figure 25: This image had the following parameters:
Date : 1 Feb 2022
LST : 16:49
Mode: os4a
Altitude: 41346 km
Orbit number: 168

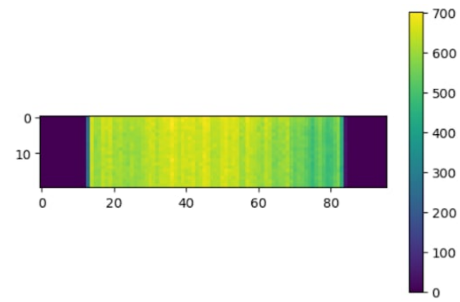


Figure 26: This image had the following parameters:
Date : 1 Feb 2022
LST : 18:00
Mode: os4b
Altitude: 42651 km
Orbit number: 168

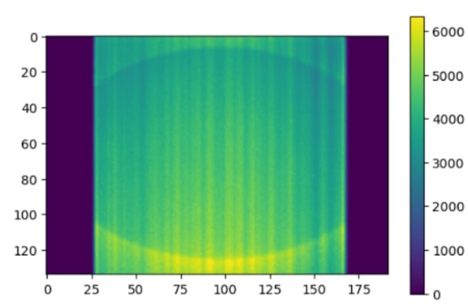


Figure 27: This image had the following parameters:
Date : 1 Feb 2022
LST : 10:38
Mode: osr
Altitude: 24632 km
Orbit number: 168

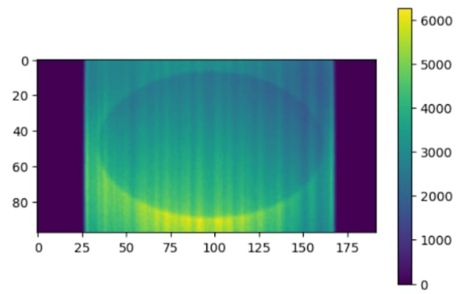


Figure 28: This image had the following parameters:
Date : 1 Feb 2022
LST : 15:13
Mode: osr
Altitude: 36931 km
Orbit number: 168

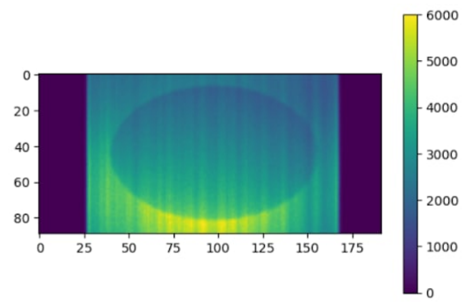


Figure 29: This image had the following parameters:
Date : 1 Feb 2022
LST : 16:39
Mode: osr
Altitude: 40977 km
Orbit number: 168

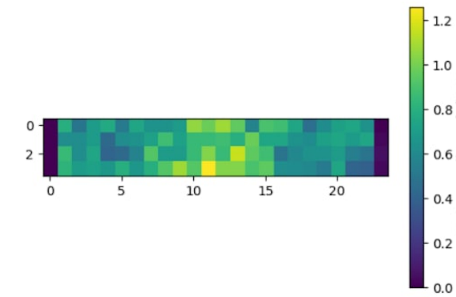


Figure 30: This image had the following parameters:
Date : 1 Feb 2022
LST : 17:40
Mode: EMU035
Altitude: 42525 km
Orbit number: 168

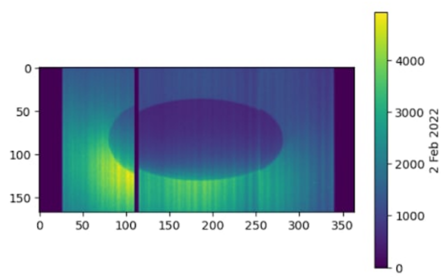


Figure 31: This plot was obtained by combining 3 different images from 2nd February 2022. The left image was taken at LST 19:03, the center at LST 02:11 and the right one at LST 22:01. The mode of all the images is os2. (wavelength: 121.56 nm, Lyman Alpha)

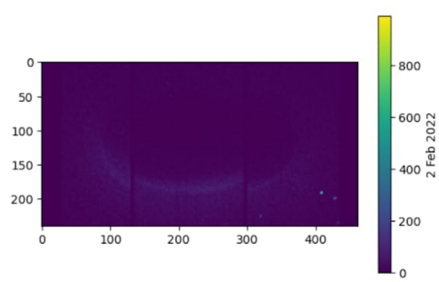


Figure 32: This plot was obtained by combining 3 different images from 2nd February 2022. The left image was taken at LST 19:03, the center at LST 02:11 and the right one at LST 22:01. The mode of all the images is os2. (wavelength: 104 nm, O)

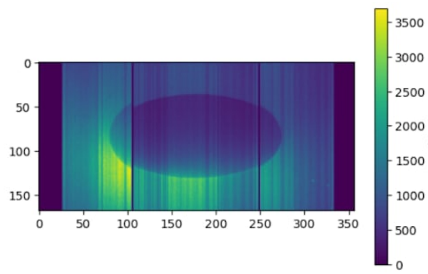


Figure 33: This plot was obtained by combining 3 different images from 2nd February 2022.
The left image was taken at LST 19:03, the center at LST 02:11 and the right one at LST 22:01. The mode of all the images is os2. (wavelength: 120 nm, N)

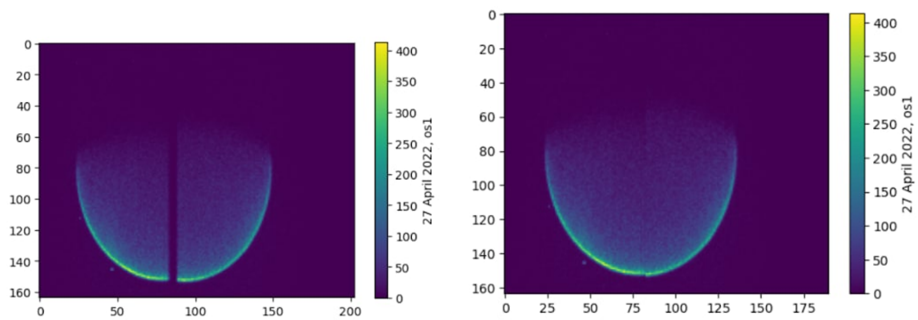


Figure 34: Orbit number: 44, 135.6 nm
Left: 27 April, OS1, 16:43 LST
Right: 27 April, OS1, 16:26 LST

2.5 Producing complete Mars images

Now, we try to plot all the data for one particular mode for one particular orbit for one month. The complete image is produced by combining data from consecutive files.

- OS1 Mode
  Shown in Fig.34
- OS2 Mode
  Shown in Fig.35 to Fig.39
- OS3 Mode
  We have to combine these 4 swaths shown in Fig.40 by rotating them by 0, 45, 90, 135 degrees respectively. The combination produced an image like the one shown in Fig.41

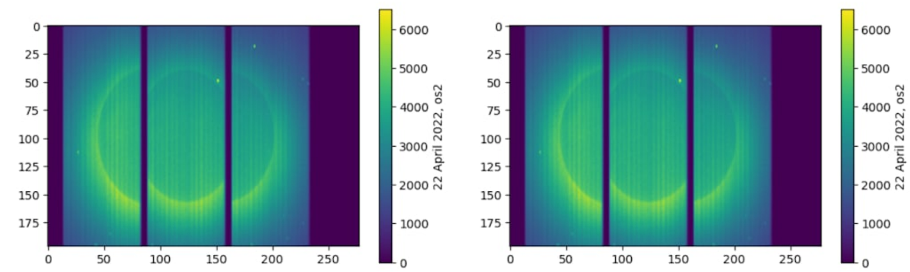


Figure 35: Orbit number: 42, 121.56 nm
Left: 22 April, OS2, 10:08 LST
Center: 22 April, OS2, 09:54 LST
Right: 22 April, OS2, 09:40 LST

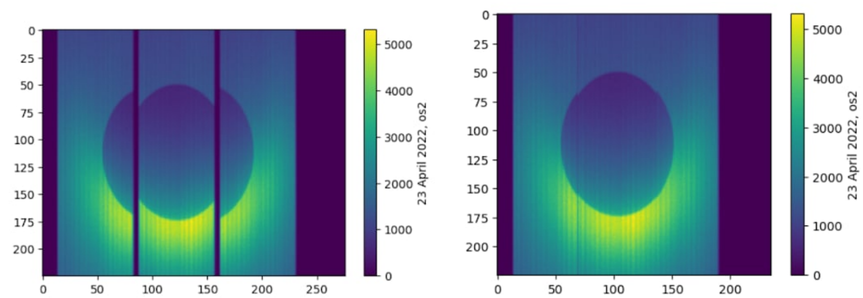


Figure 36: Orbit number: 42,121.56 nm
Left: 23 April, OS2, 19:07 LST
Center: 23 April, OS2, 18:50 LST
Right: 22 April, OS2, 18:32 LST

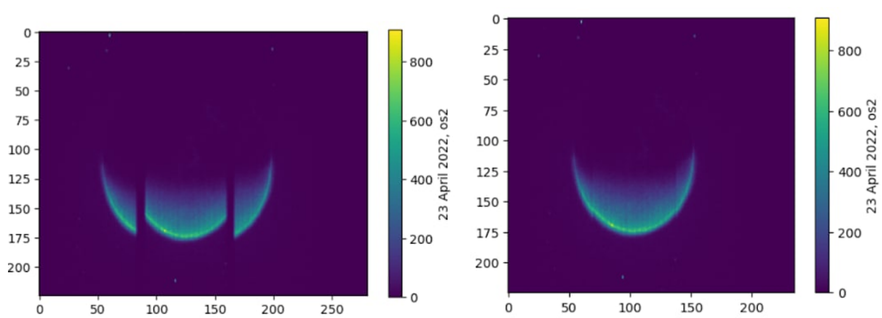


Figure 37: Orbit number: 42,130.4 nm
Left: 23 April, OS2, 19:07 LST
Center: 23 April, OS2, 18:50 LST
Right: 22 April, OS2, 18:32 LST

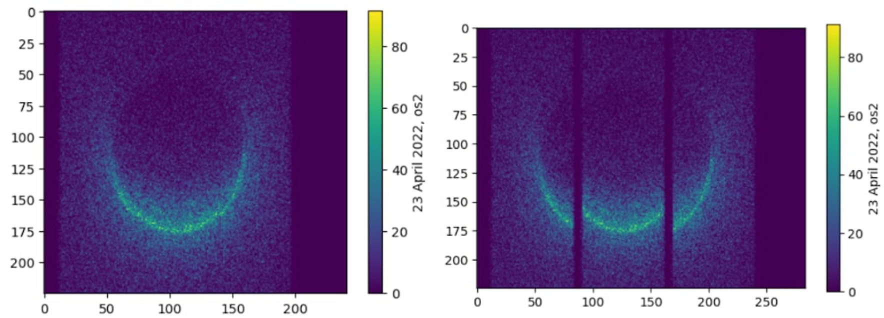


Figure 38: Orbit number: 42,102.6 nm
Left: 23 April, OS2, 19:07 LST sw3 of 3
Center: 23 April, OS2, 18:50 LST sw2 of 3
Right: 22 April, OS2, 18:32 LST

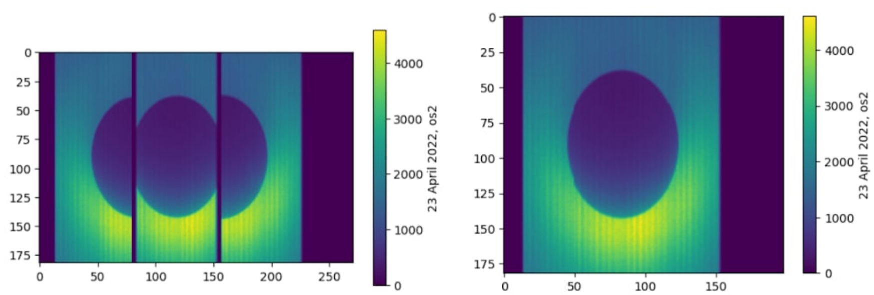


Figure 39: Orbit number: 42, 121.56 nm
Left: 23 April, OS2, 04:12 LST
Center: 23 April, OS2, 21:13 LST
Right: 23 April, OS2, 21:02 LST

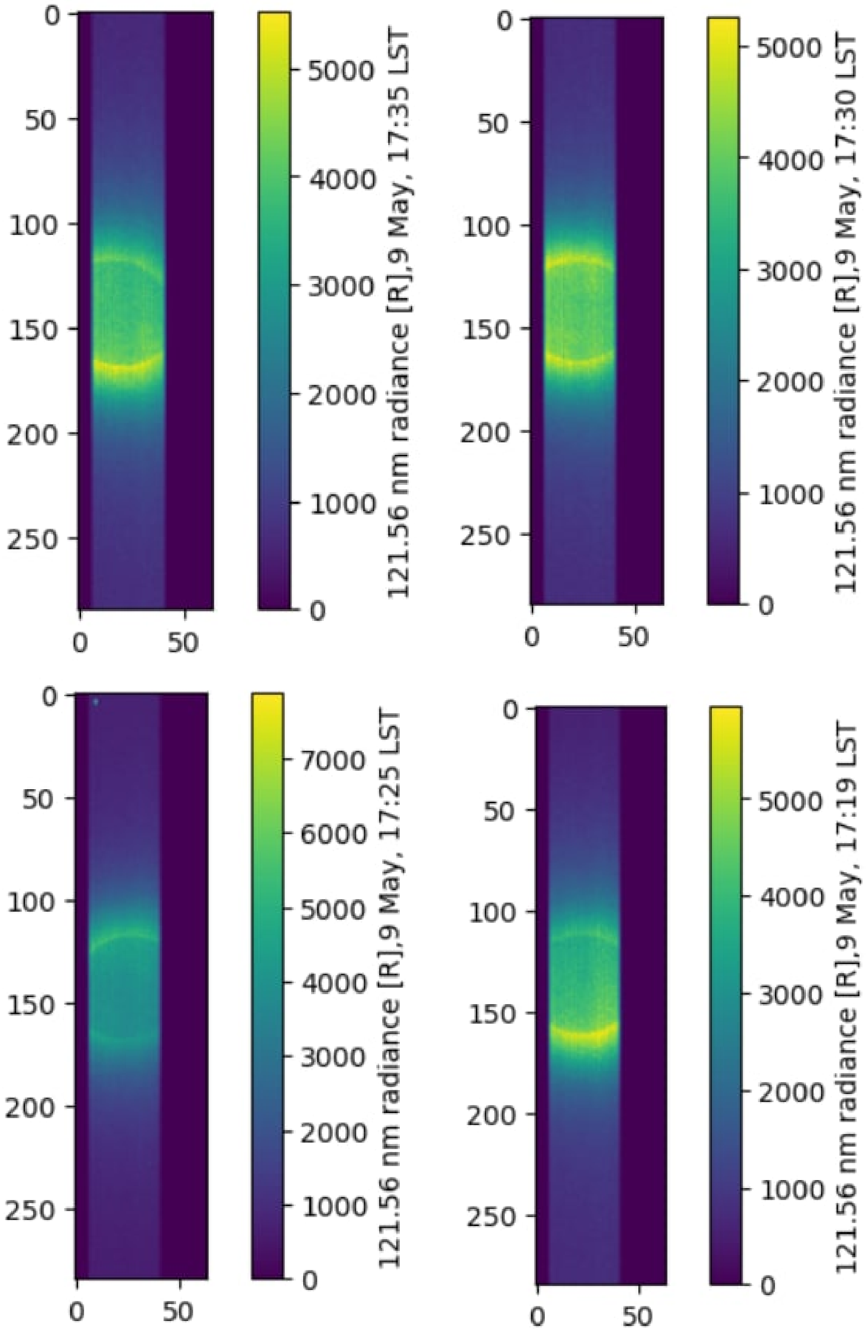


Figure 40: OS3 Mode, 4 swaths

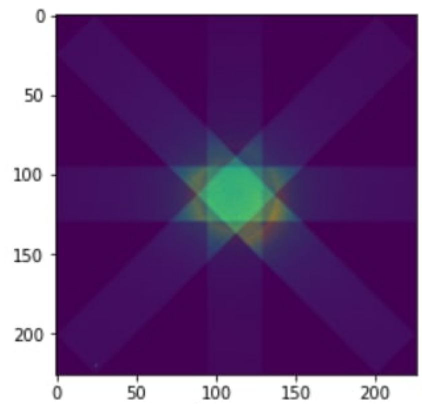


Figure 41: The above image is from 9 May 2021. The mode of these observations is os3a, and was produced by overlapping the images one on top of the other.

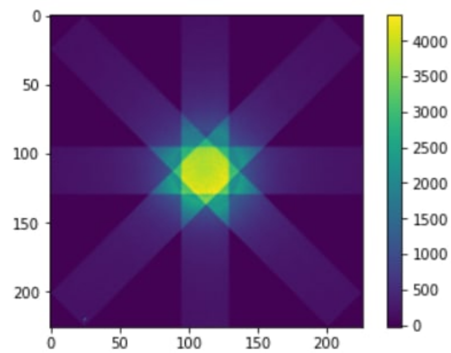


Figure 42: Date: 9 May 2021

We produced another image by averaging out the pixel intensity of the 4 images and then the image as shown in Fig.42 was produced.
However in this image, we have averaged all the pixels in all the 4 arrays when instead what we should have done is to average only the pixels above a minimum threshold value (which is about 100). Doing that, an image as given in Fig.43 is produced:
Another example is shown in Fig.44
These images in Fig. and Fig. are actually containing the data from both os3a and os3b mode (as os3a mode also captures the data captured by os3b mode in addition to capturing the disk of mars). The os3b data mainly looks away from the sun capturing the interplanetary hydrogen background.

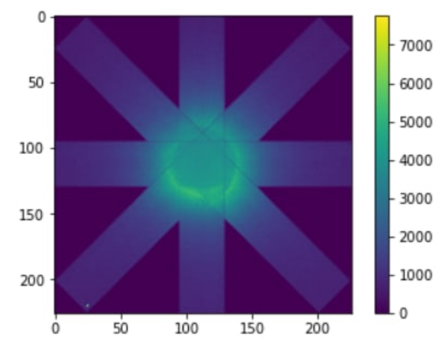


Figure 43: Orbit number: 50
Date: 9 May 2021
Mode: os3a +os3b (actually os3a mode contains the os3b embedded in itself)
Size of individual arrays used to produce the above image: (285,64)

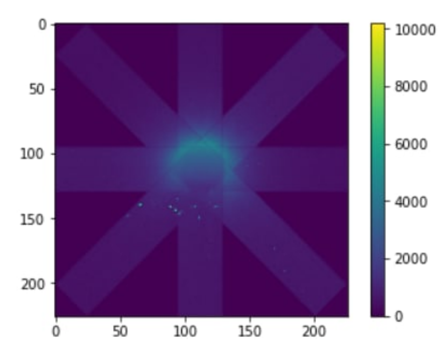


Figure 44:

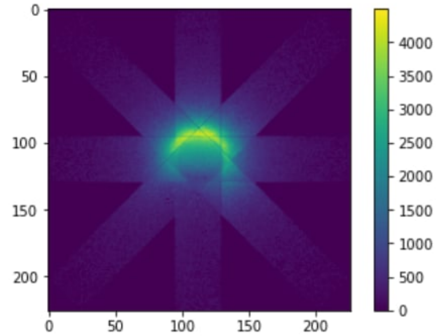


Figure 45: Date: 25th June 2021
Orbit number: 71
Mode: os3a- os3b (os3a data has os3b data embedded in itself, so we subtract the os3b data from the os3a data)

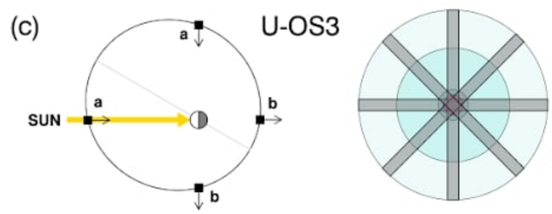


Figure 46:

So, from the os3a data, we try to "subtract" or remove the os3b data, since we would like to only view the planet and its atmosphere and not the interplanetary hydrogen background. The images in Fig.45 show a few examples of the same:
One thing to note here is the timing from which the data of os3a mode and os3b mode is taken.
Consider Fig.46

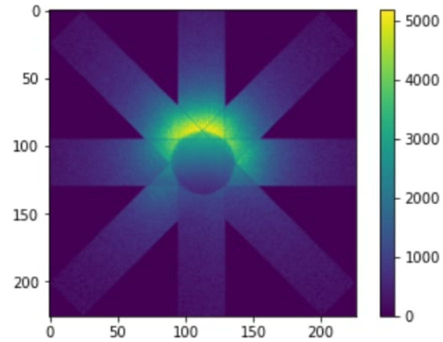


Figure 47: Date: 29 November 2021
Orbit number : 140
(Note: the individual 4 images/ np.ndarrays for the above had shapes of (285,48))

The data from os3a mode should be looking towards nadir, and the data from os3b must be anti nadir and almost 180 degrees apart from the position from which os3a data was captured. For this, the timing of the data is an important factor which should be taken into consideration. The two timings (one of os3a mode data and the other of the os3b mode data) should be around 12 hours apart to ensure this.
In one of the example, the os3a data was taken from a time around 06:05 LST and the os3b data was taken from a time around 18:16 LST, and these are almost 12 hours apart.
Consider few more examples shown in Fig.47 and Fig.48
Comparison between os3a mode images and os3a-os3b images is shown in Fig.49

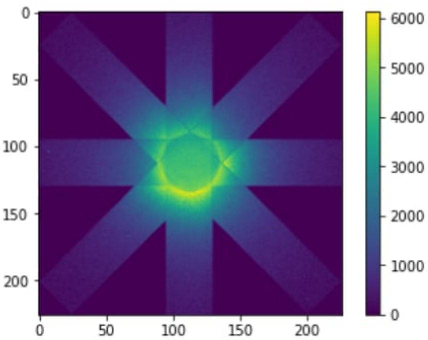


Figure 48: Date: 14 December 2021
Orbit number: 146
(Note: the individual 4 images/ np.ndarrays for the above had shapes of (285,48))

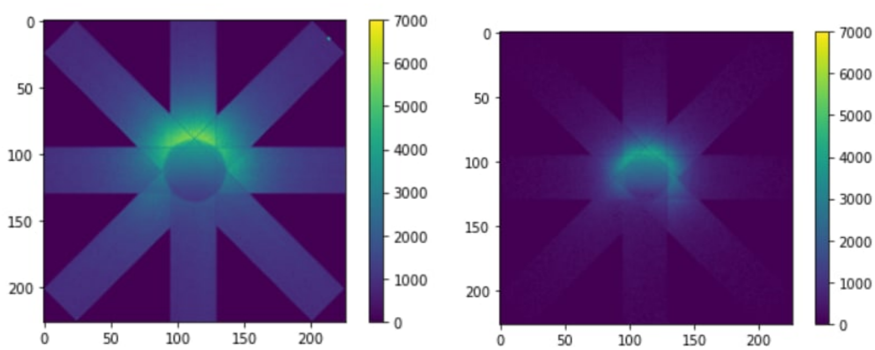


Figure 49: Date: 29 November 2021
Orbit number : 140
(Note: the individual 4 images/ np.ndarrays for the above had shapes of (285,48))
Left image represents os3a , right one represents os3a-os3b

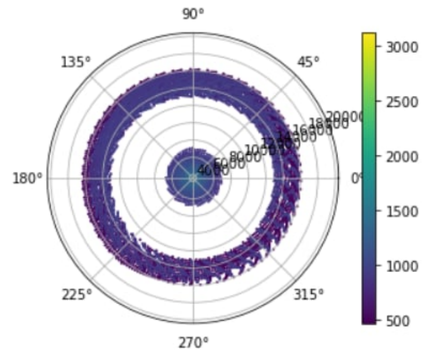


Figure 50: This image is for :
Orbit number : 72
Date: 27 June 2021
Mode: os4a
It was produced by a combination of 4 images.

- OS4 Mode
  We obtained 6 images from 6 consecutive files from the os4a mode. The image dimensions for all the 6 images were different along one axis (say along the X axis). The image dimensions along the other (Y axis) were all the same (96). We tried to combine these 6 images along the dimension which was constant in all figures (Y axis), however no useful image was produced as such (the combining of the images was not successful). The data for different days was captured around near the same time (that is, data on day 1, was captured nearly at the same LST as data on day 2) We don't understand how the images are supposed to be combined together.

2.6 Polar plots

We plotted some polar plots for the data of the os4 mode. Since the observations are a bit unusual, they are not scheduled like the other observations. The way we have used to plot the OS4 data is by making polar plots with the radius as the minimum ray altitude height and the theta as Solar Zenith Angle. The limit of the radius were set from 3600km (about 1.06 times the radius of Mars) to 20400km (about 6 times the radius of Mars).
We took the os4a mode files for one orbit and tried to combine them all together (plotting them on a solar plot). One example is shown in Fig.50:
To produce these images, we have used plt.scatter , and by varying the marker size (making it small enough) we tried to ensure that there is not much overlap between different points. (Fig.51)

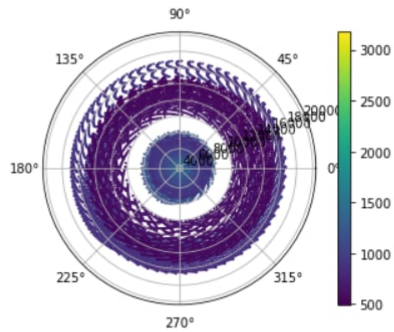


Figure 51: This image is for :
Orbit number : 76
Date: 6 July 2021
Mode: os4a
It was produced by a combination of 4 images.

3 EMIRS Data

3.1 Different plots

We did 4 basic kinds of plots for one file each from daytime and nighttime. The 4 plots were as follows:(in each plot we plotted the maximum brightness temperature)

- Longitude vs Latitude map
  1. Daytime plot (Fig.52 and Fig.53)
  2. Nighttime plot (Fig.54)
- Latitude vs Altitude map
  1. Daytime plot (Fig.55 and Fig.56)
  2. Nighttime plot (Fig.57)
- Local solar time vs Altitude map
  1. Daytime plot (Fig.58 and Fig.59)
  2. Nighttime plot (Fig.60)
- Local solar time vs Latitude map
  1. Daytime plot (Fig.61 and Fig.62)
  2. Nighttime plot (Fig.63)

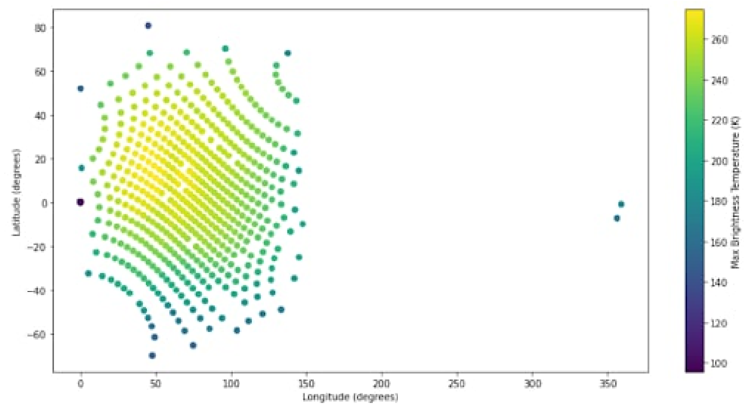


Figure 52: Longitude vs Latitude map
Daytime: (orbit number :65, LST: 13:19, date:11/06/2021, before removing the unwanted data points)

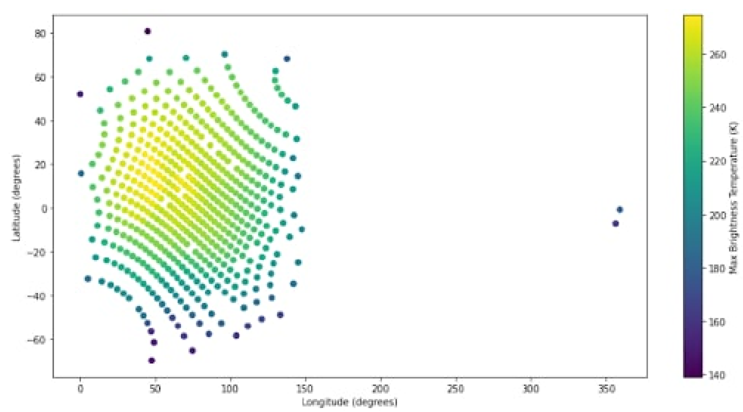


Figure 53: Longitude vs Latitude map
Daytime: (orbit number :65, LST: 13:19, date:11/06/2021, after cleaning the data)

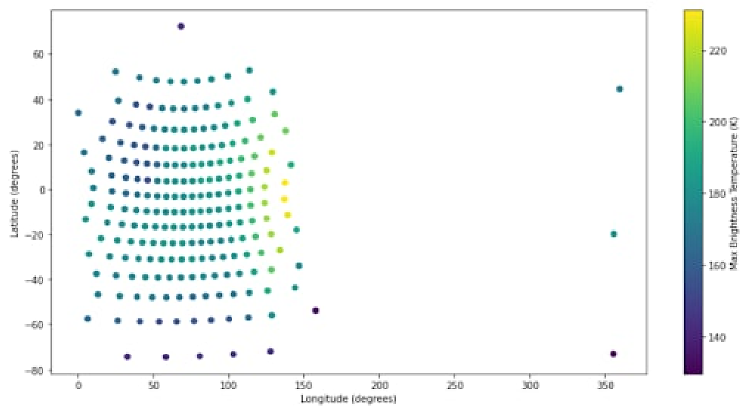


Figure 54: Longitude vs Latitude map
Nighttime (orbit number:8, LST: 04:38, date:20/02/2021, after cleaning the data)

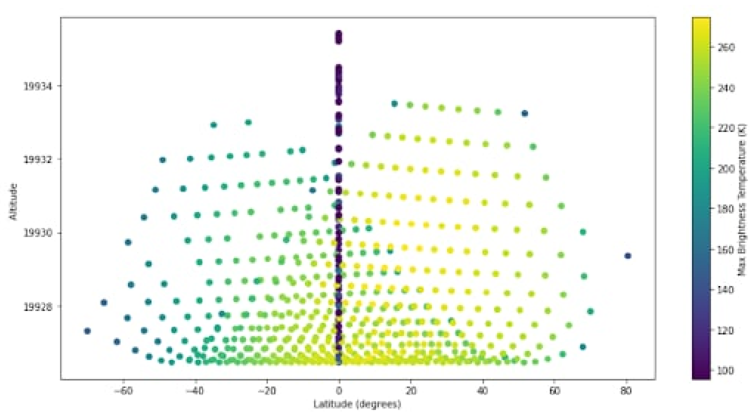


Figure 55: Latitude vs Altitude map
Daytime: (orbit number :65, LST: 13:19, date:11/06/2021, before removing the unwanted data points)

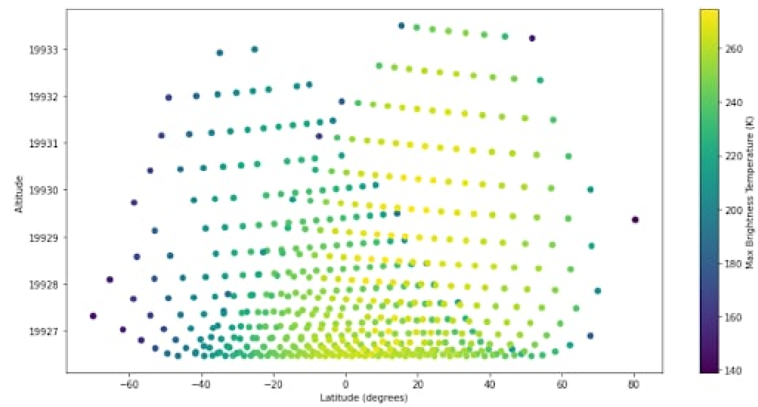


Figure 56: Latitude vs Altitude map
Daytime: (orbit number :65, LST: 13:19, date:11/06/2021, after cleaning the data)

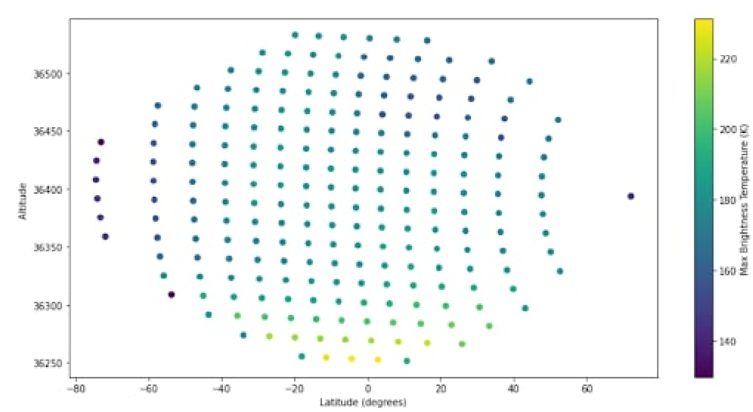


Figure 57: Latitude vs Altitude map
Nighttime (orbit number:8, LST: 04:38, date:20/02/2021, after cleaning the data)

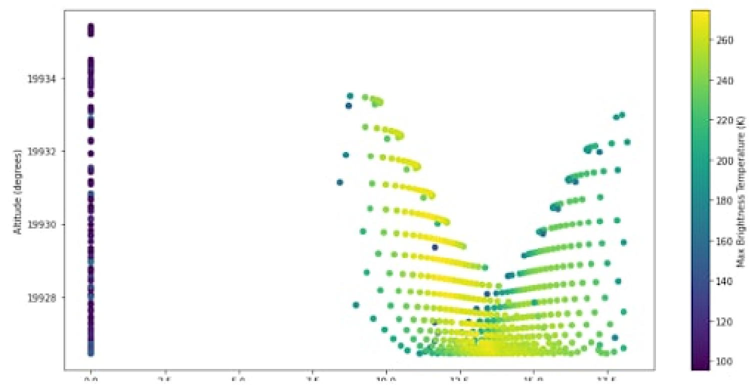


Figure 58: Local solar time vs Altitude map
Daytime: (orbit number :65, LST: 13:19, date:11/06/2021, before removing the unwanted data points)

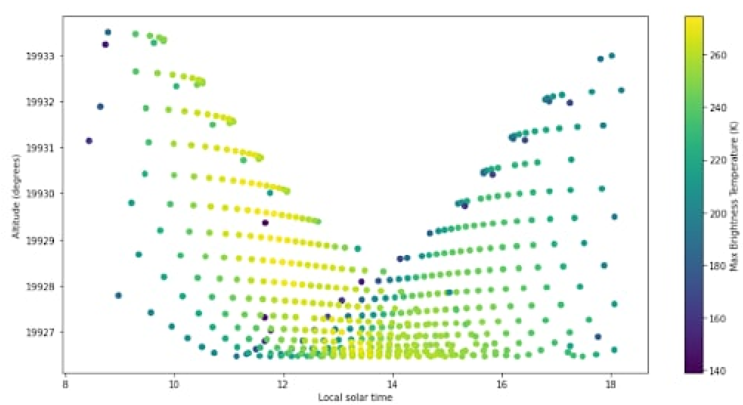


Figure 59: Local solar time vs Altitude map
Daytime: (orbit number :65, LST: 13:19, date:11/06/2021, after cleaning the data)

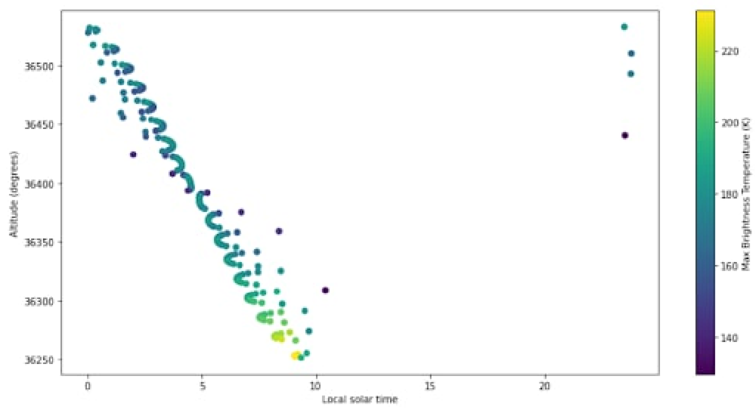


Figure 60: Local solar time vs Altitude map
Nighttime (orbit number:8, LST: 04:38, date:20/02/2021, after cleaning the data)

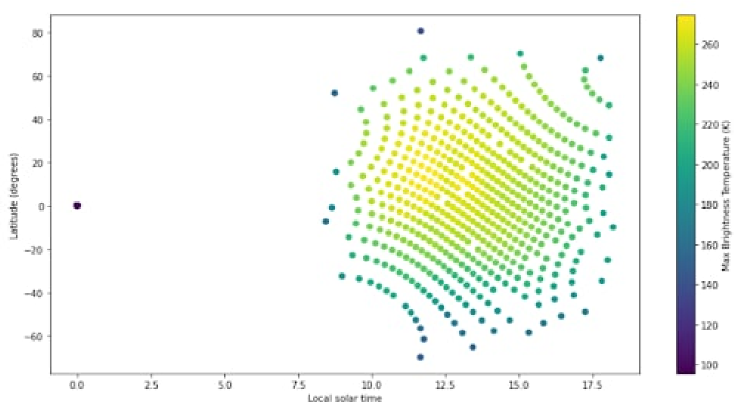


Figure 61: Local solar time vs Latitude map
Daytime: (orbit number :65, LST: 13:19, date:11/06/2021, before removing the unwanted data points)

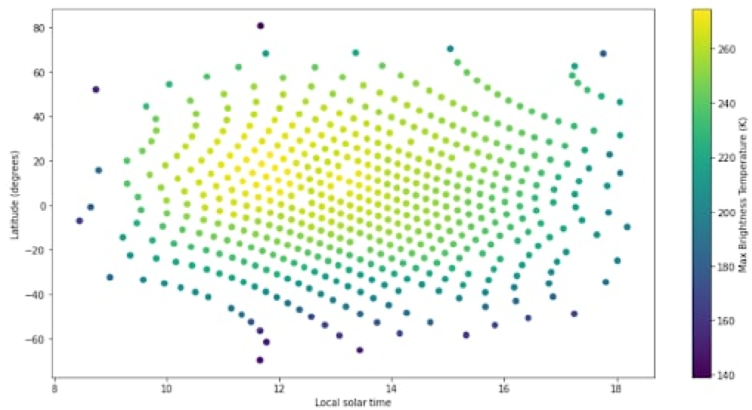


Figure 62: Local solar time vs Latitude map
Daytime: (orbit number :65, LST: 13:19, date:11/06/2021, after cleaning the data)

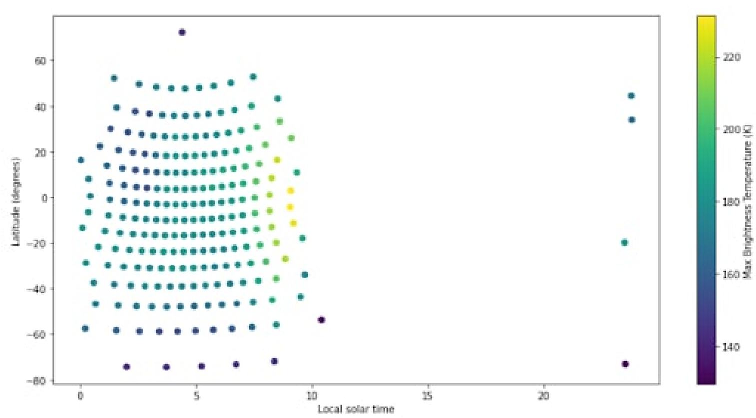


Figure 63: Local solar time vs Latitude map
Nighttime (orbit number:8, LST: 04:38, date:20/02/2021, after cleaning the data)

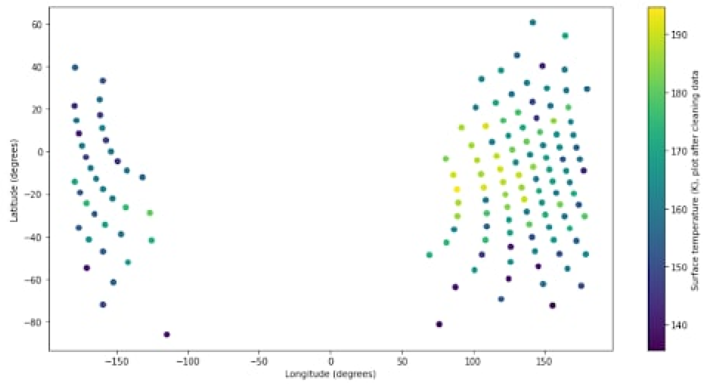


Figure 64: Latitude vs. Longitude map

3.2 Level 3 data

The level 3 data has parameters like surface temperature and atmospheric temperature (simply referred to as temperature)
We made the following plots:

- Plotting the surface temperature in a longitude vs latitude plot (Fig.64)

  An interesting thing here to note is that while the longitude and latitude have the same dimensions, the surface temperature does not have the same dimension. Thus, initially we were unable to make this plot due to dimensional incompatibility (as all the data must have the same shape in order for it to be plottable). But if we clean the data first (that is, remove all the points where the latitude is 0 and longitude is 0 from all three arrays , that is from latitude, longitude and surface temperature) all the three arrays turn out to be of the same dimension, and thus, we are able to plot them.

- The next plot we did was that of atmospheric temperature in a longitude vs latitude plot.

  The atmospheric temperature is a 2D array, and captures data at 20 different levels of altitude which lie below the spacecraft altitude. Thus, the dimensions of the array temperature are always something like (num,20).
  To plot this, we used two methods:

  1. First, we average out the data at all the 20 levels, and create a 1D array of dimension (num). Then we clean out the data (by removing all points where the latitude and longitude are 0 from the latitude, longitude and temperature array). This produced the kind of a plot shown in Fig.65

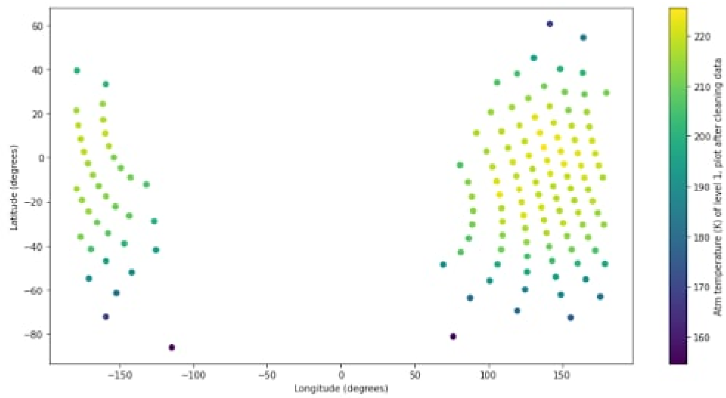


Figure 65: Latitude vs. Longitude map

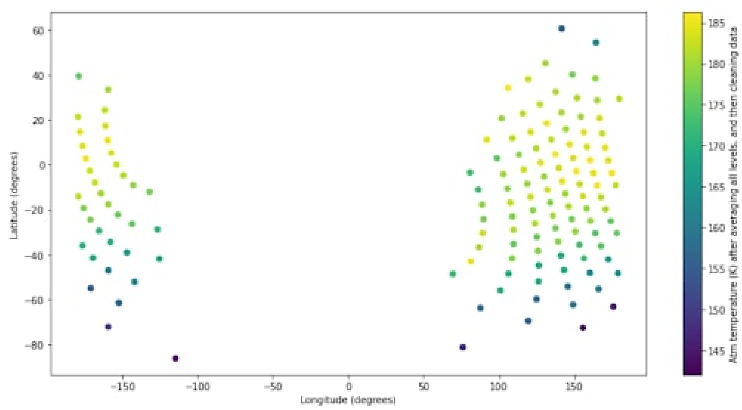


Figure 66: Latitude vs. Longitude map

2. In the second way, we first clean the data (that is, remove all the points from the three arrays where the latitude and longitude are 0). Then we average out the data of the 20 levels to produce a 1D array with dimension of num. Now we plot this data. This produced the kind of a plot shown in Fig.66

3.3 Three Dimensional plots

- We plot atmospheric temperature in a plot with latitude, longitude and altitude as the 3 axes: (Fig.67)
- Where we plot atmospheric temperature in a plot with latitude, longitude and level as the 3 axes. (Fig.68)

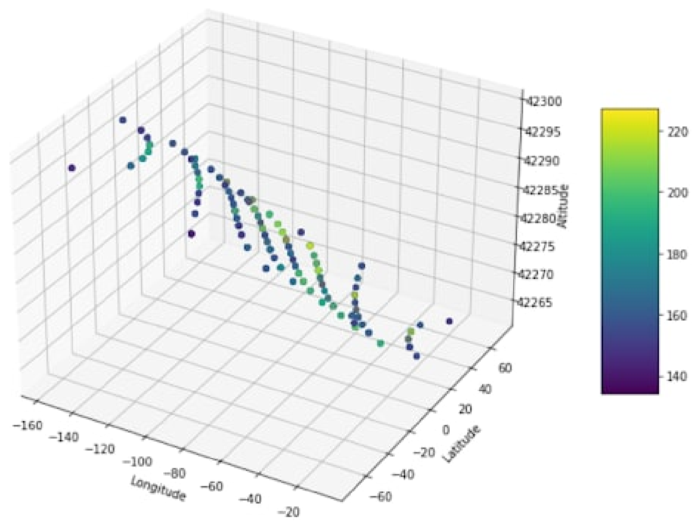


Figure 67: 3D plot: We have plotted atmospheric temperature in a map whose axis indicate latitude, longitude and altitude.

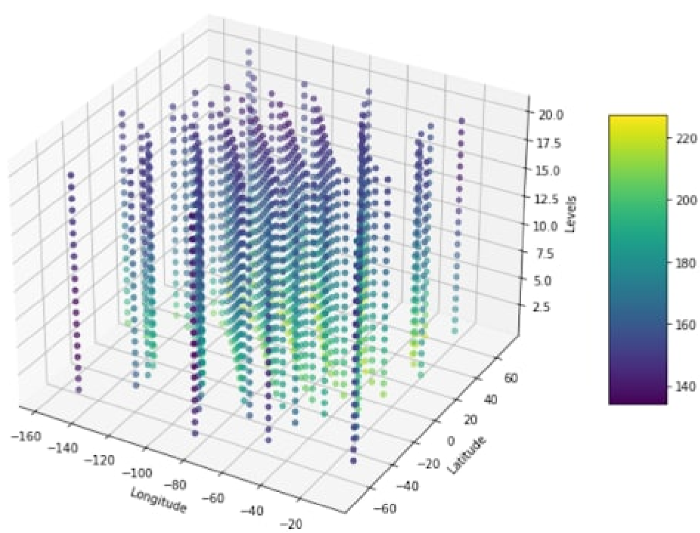


Figure 68: 3D plot: We have plotted atmospheric temperature in a map whose axis indicate latitude, longitude and levels.

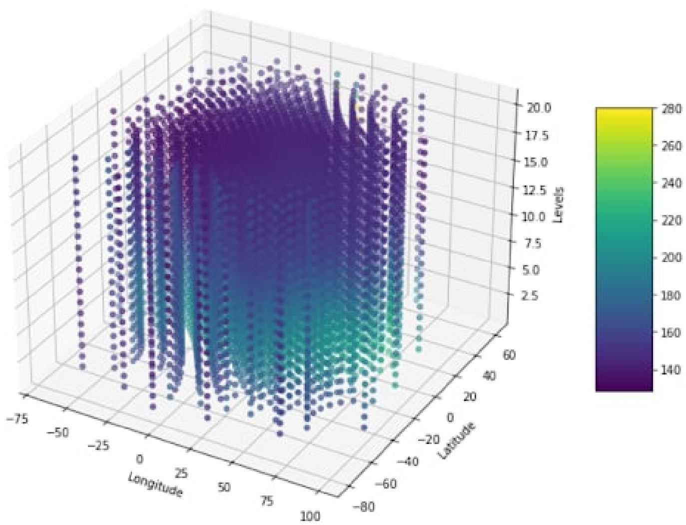


Figure 69: Daytime plot, LST: 13:02

Basically, since the atmospheric temperature data is 2D (instead of being 1D like the surface temperature data), we have the temperature data for 20 different levels. Thus, we need to make a 3D plot , where we plot these 20 different levels (all of which lie below the spacecraft altitude and for these levels, the temperature data was captured). We make these plots with the x axis as longitude, y axis as latitude and the z axis as these 20 different levels.
As can be seen in these plots, the temperature doesn't change much with levels, and is more or less similar to the 2D plots we made with longitude and latitude as the 2 axis, and plotting the temperature by averaging them for each level . There is not much difference in the two plots as seen from the picture.

Some examples from daytime and nighttime are shown in plots in Fig.69 and Fig.70

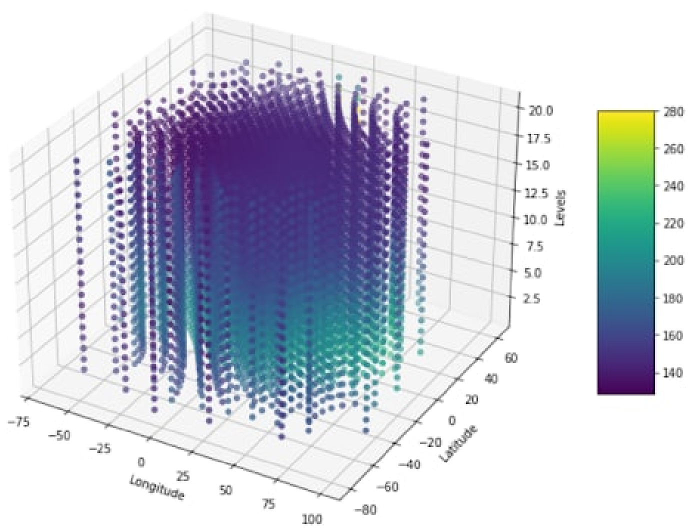


Figure 70: Nighttime plot, LST: 20:38