

Microwave Mapping of Jupiter's Vortices

By:

Madeline Deming
Abraham Skoczylas

Contributors:

Jiheng Hu
Cheng Li

Github: <https://github.com/aviskoczylas/space-405-code/blob/main/README.md>

Project Background:

Jupiter Vortices:

Jupiter is composed of bands, which can be seen as alternating lighter and darker bands that encircle the whole planet. These bands are caused due to temperature differences that can create jet streams, moving winds of atmospheric gasses traveling in opposite directions.

Vortices punctuate these bands. Vortices are giant cyclones and anticyclone storms that are part of Jupiter's atmosphere. The largest vortex is the Great Red Spot, but these vortices greatly differ in size, color, and duration of how long they last. One of our main goals of this project was to understand how these storms vary with atmosphere depth. To understand this, we needed to map and identify Jovian vortices based on brightness temperature data from the Juno Mission.

Juno Spacecraft & Mission:

The Juno spacecraft launched August 5th, 2011 with a mission to orbit around Jupiter to answer fundamental questions about Jupiter's atmosphere, learn more about its moon, and to better understand the origins of our solar system. Juno is still currently an active mission, and since it entered orbit around Jupiter on July 4th, 2016, it has completed 35 orbits around Jupiter, collecting more than three terabits of data^[1]. The spacecraft contained many important instruments including a magnetometer, microwave radiometer, ultraviolet imaging spectrograph, and a Jovian infrared auroral mapper to name a few. To better understand the climate and weather patterns of the vortices of Jupiter, our project focused on analyzing boresight and temperature brightness data collected from the microwave radiometer aboard the mission.

Microwave Radiometer:

One of the instruments on the Juno spacecraft was a microwave radiometer (MWR). The instrument measures the brightness temperature, or the measure of the intensity of energy coming from a source. The MWR contains six different channels that allows Juno to make observations at multiple wavelengths, and therefore, measurements from the top to deep into Jupiter's atmosphere. Each channel takes measurements at a different frequency. These frequencies range from 600 MHz to 22 GHz, allowing us to make observations nearly 600 km deep into Jupiter's atmosphere. Channel six corresponds to the highest frequency (shortest wavelength) while channel one corresponds to the lowest frequency (longest wavelength). Since channel six has the shortest wavelength, the data collected in that channel shows observations closest to the top of Jupiter's atmosphere while channel one makes observations deep within Jupiter's atmosphere. These channels are designed to detect the dynamic

characteristics of Jupiter's sub-cloud atmosphere and the presently unknown variations in the abundances of ammonia and water deep within these clouds.

The MWR instruments contain three main subsystems, which are the microwave receivers, the electronics within the spacecraft, and the antennas.

Receivers:

There are six receivers part of the MWR instrument, one for each frequency channel. They are direct detection Dicke switch radiometers with integral noise diodes and are secured together in a compact unit with dimensions of 130 x 250 x 31 mm^[2]. Each receiver has around a 4% radio frequency (RF) bandpass^[2]. The RF signal is converted to DC from a diode detector, then to pulses by a voltage to frequency converter. There are three noise diodes that are strategically placed around the receiver and antennas to allow for a large dynamic range needed to observe the emission at all frequency channels. To account for a safety factor and ensure the instruments works as intended, the upper end of ranges for each channel is at two times the expected maximum antenna temperature, and the noise diodes range of brightness temperature measurements is 100-300K^[2].

Electronic Unit:

The electronic unit consists of a power distribution unit (PDU), command and data unit (CDU), and housekeeping data units (HKU), all of which are designed to fit into a box shaped housing unit. The PDU converts the 28 V power from the spacecraft to $\pm 5, 7, 12$, or 15V sources to power the receivers and to other EU slices^[2].

Antennas:

The antennas are located on external panels of the spacecraft. The antennas point perpendicular to the spacecraft spinning axis which allows the antenna beams to move in circular patterns as the spacecraft rotates. The beamwidth for the antennas is 20°, and 12° for the third lowest frequency and remaining antennas.^[2] Antennas 1 and 2 have a patch array design to achieve the smallest possible mass per unit area. Antennas 3 - 5 have a slotted array design, and antenna 6 has a corrugated horn with a profile shape to minimize size and mass at its desired beamwidth. The main beam efficiency for all antennas is 99%, which means that only 1% of power received came from the sidelobe region^[2].

Measurements:

All the atmosphere data for the mission was obtained during approximately three hours when centered on perijove, or the point in Juno's orbit when it was closest to Jupiter. The MWR

instrument beams would sweep circles in the sky for every 30 second spin cycle of the spacecraft^[2]. This allows latitudes to be tracked many times over a range of emission angles. Each measurement has a 100 millisecond integration of the brightness in the beam's field of view as it moves through an arc of 1.2° , or 1/10th of the smallest beamwidth^[2]. Data was taken from pole to pole on Jupiter, however the most useful data for analyzing Jupiter's atmosphere is obtained with $\pm 50^\circ$ of the general latitude, where the beam footprints or boresights are small and can show important spatial features. This space was approximately 40 minutes long with 80 spacecraft rotations allowing for approximately 10,000 brightness temperature measurements to be analyzed^[2].

Project Methodology:

For our project, we created three basic python scripts, `juno_data_processing.py`, `boresight_images.py`, and `tchgraphs.py` (there is a separate `tchgraphs.py` for each channel, or 6 in total) which can be found in the project Github. To get more familiar working with data, we broke our project into three main steps:

1. Converting data into a useable format, plotting it, and gaining a basic understanding of what it shows
2. Plotting ellipse footprints, calculating boresights, and understanding how data was collected with respect to Jupiter's "surface"
3. Isolating relevant data to analyze and removing noise
4. Analyzing the brightness temperature for each channel to understand Jupiter's atmosphere

The first task was completed using the `juno_data_processing.py` code, the second task was completed with the `boresight_images.py` code, and the third was completed using `tchgraphs.py` 1-6^[3]. The last objective was accomplished by visually analyzing the plots generated by the scripts.

Juno data processing.py:

We first needed to convert the data collected in hdf5 files to numpy arrays in python. We then began to understand the data given in these files and created basic plots in python. These plots included latitude and longitude plots versus time, local zenith angle plots, temperature plots for each channel, and more.

Using our first script, we were able to read in and obtain all the following data and member groups as shown in figure 1 to the left. After reading in the data, we were able to convert the data into numpy arrays in order to process and plot the data. In the code,

```
Group: s0
Member: latitude
Member: longitude
Member: range
Group: s1
Member: LZA
Member: TA_Ch1
Member: latitude
Member: longitude
Group: s2
Member: LZA
Member: TA_Ch2
Member: TA_Ch3
Member: TA_Ch4
Member: TA_Ch5
Member: TA_Ch6
Member: latitude
Member: longitude
```

Figure 1: Data fields from Juno

the expressions for each individual plot have been commented out, so in order to plot a particular dataset the relevant code must be uncommented first.

The first plots generated were one-dimensional plots that plotted a given member array with respect to time. Below, figures 2a and 2b show examples of these plots, which include plotting the latitude of the spacecraft and the longitude of the spacecraft individually with respect to time. Figure 2c shows a parametric plot of the path the spacecraft took with time stamps included.

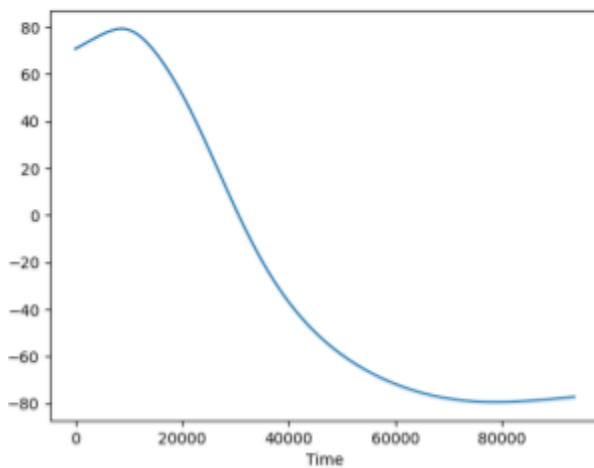


Figure 2.a Latitude vs Time

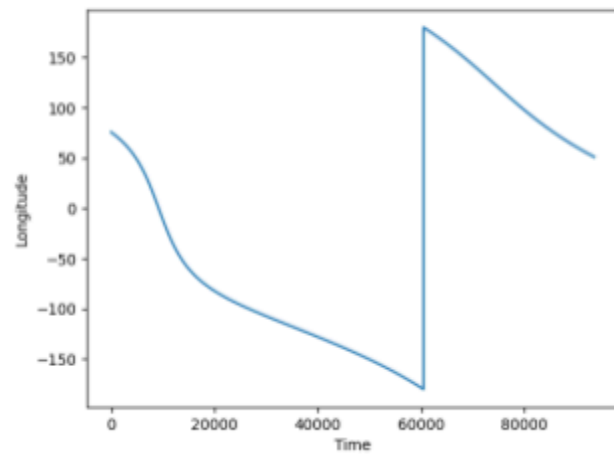


Figure 2.b Longitude vs Time

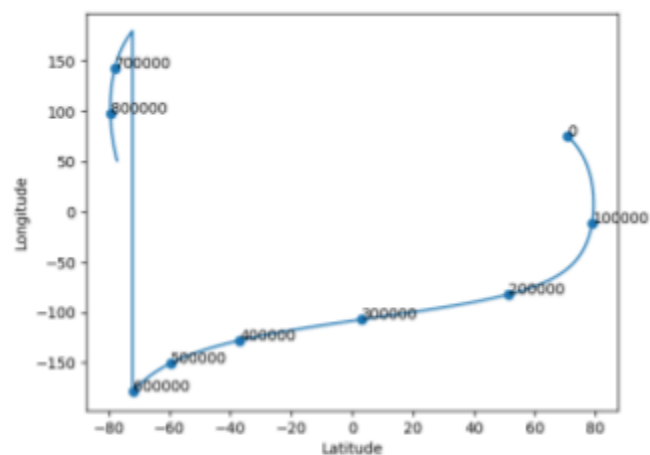


Figure 2.c Latitude vs Longitude

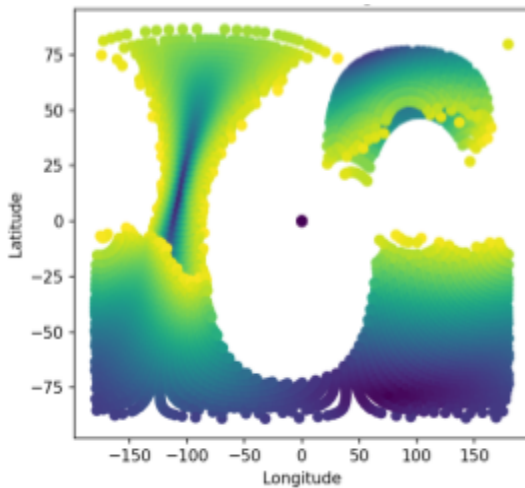


Figure 2.d Local Zenith Angle

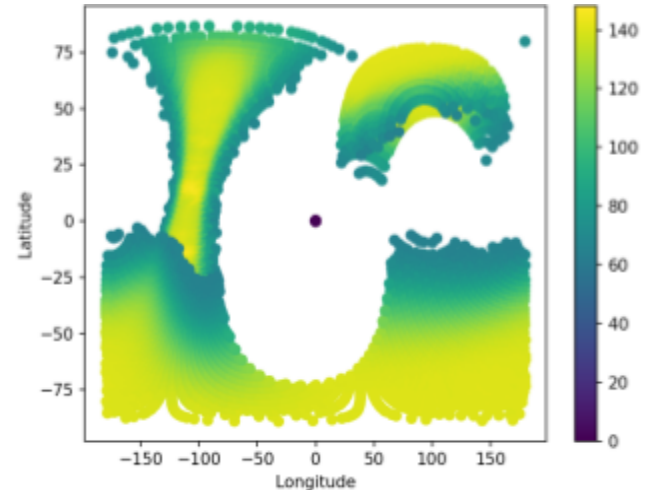


Figure 2.e Local Temperature

After creating these basic plots, we moved on to create two dimensional plots to plot the temperature of channel 6 and the local zenith angle as a function of latitude and longitude. Because the temperature and location data were not directly connected within the file, this was completed by making lists corresponding to zenith angle, temperature, latitude, and longitude. The latitude and longitude in this list corresponded to each temperature and local zenith angle at that point. Since the data are now in same size lists, we can create a scatter plot of the local zenith angle and temperature at an exact (latitude, longitude) location as shown above in figure 2d and 2e.

Boresight images.py:

To understand the temperature data for each channel, we first need to understand how the MWR obtains its data. The MWR instrument actually takes its data over an ellipse and averages it to obtain the brightness temperature at the center of the ellipse. This point is called the boresight. This method of data collection has a few implications for large and incomplete ellipses. First, when the measurement is taken at a large zenith angle, the ellipse is larger, due to a constant angle being spread over a larger area. This leads to a less accurate measurement, as a larger ellipse averages data over a larger area, leading to a less representative estimate of the actual point. Second, data taken near the edge of the planet where ellipses are incomplete are not very useful for analyzing the atmosphere, as the measurement is averaging the temperature of Jupiter with the temperature of space behind it. Therefore, our goal is to plot complete footprints and boresights on Jupiter's surface and focus on analyzing the clean, smaller ellipses moving forward.

Following the same process as described above, we began plotting the boresight data by converting the hdf5 files to numpy arrays and lists where list `s1[0]` is the boundary latitudes, `s1[1]` boundary longitudes, and `s1[2]` is the number of boundary points for the boresights.

We noticed a few problems with the data. First, there were junk values stored in the data files, denoted by the number -999.9 which was making our plots inaccurate. In order to fix this problem, we masked this data value using a function that allowed us to ignore all latitude and longitude points that are equal to -999.9.

After masking junk data we proceeded to plot the ellipses and boresights. Our goal is to see where these boresights actually lie on Jupiter's surface, so we obtained a cylindrical image of Jupiter and scaled this image to match our data's latitude and longitude values. In order to prevent the plot from being overloaded with data, we plotted multiple boresights by slicing the arrays and plotting data only from the `n`th row and column. The sampling rate, `n`, is an adjustable parameter in the code^[3]. We plotted all boundary points to see the ellipses' footprints as shown in figure 3a. Then we plotted the boresight by taking the average latitude and longitude value of the footprint points for each ellipse.

However, as described above some ellipses were incomplete, and we want to disregard this data. So we added an if statement to the code to only plot boresights for complete ellipses, or ellipses with 360 boundary points. Now that we have only valid data, we will focus on analyzing the smaller, complete ellipses shown in figure 3b, specifically the ones between 225 and 300 degrees longitude.

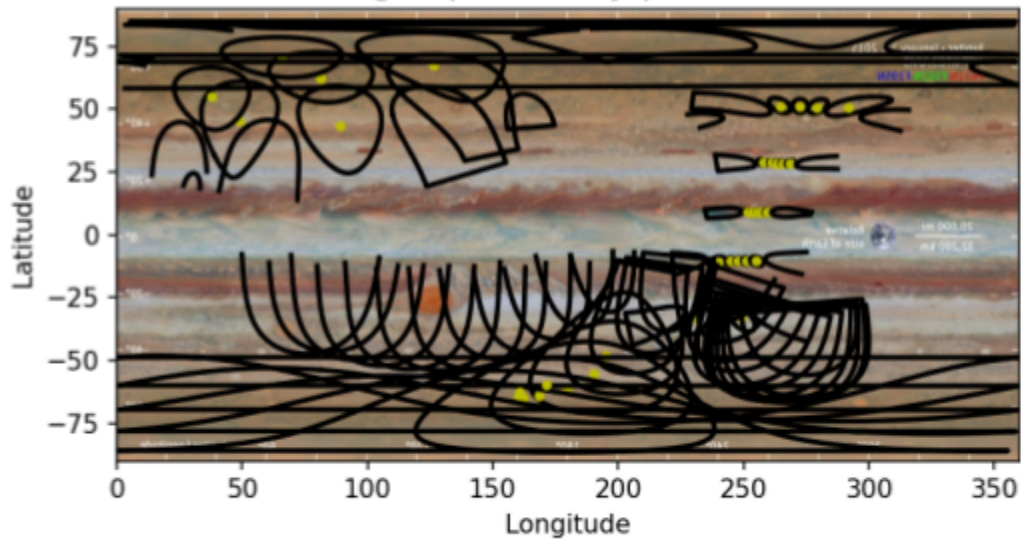


Figure 3.a Ellipses and boresights plotted on Jupiter's surface

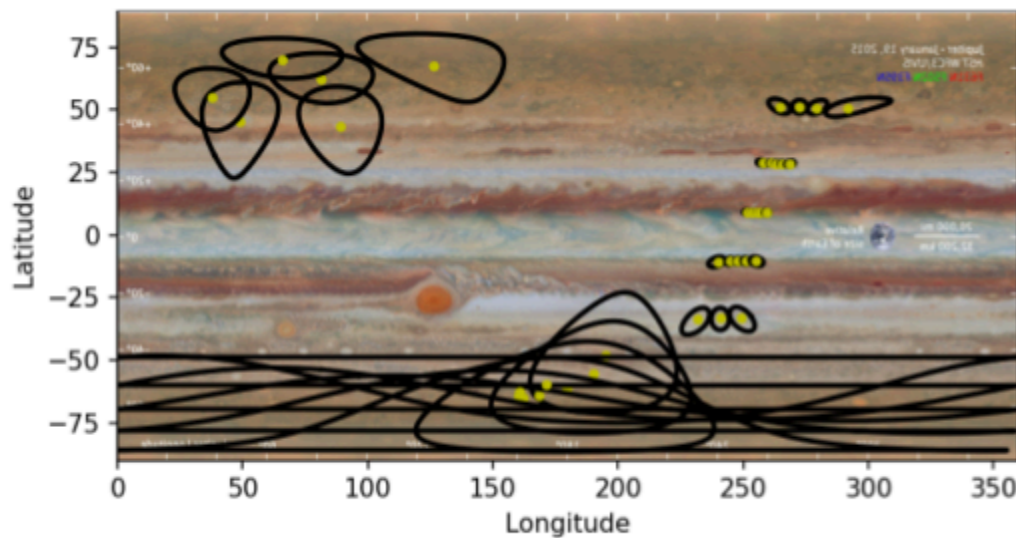


Figure 3.b Complete ellipses and boresights plotted on Jupiter's surface

tchgraphs.py

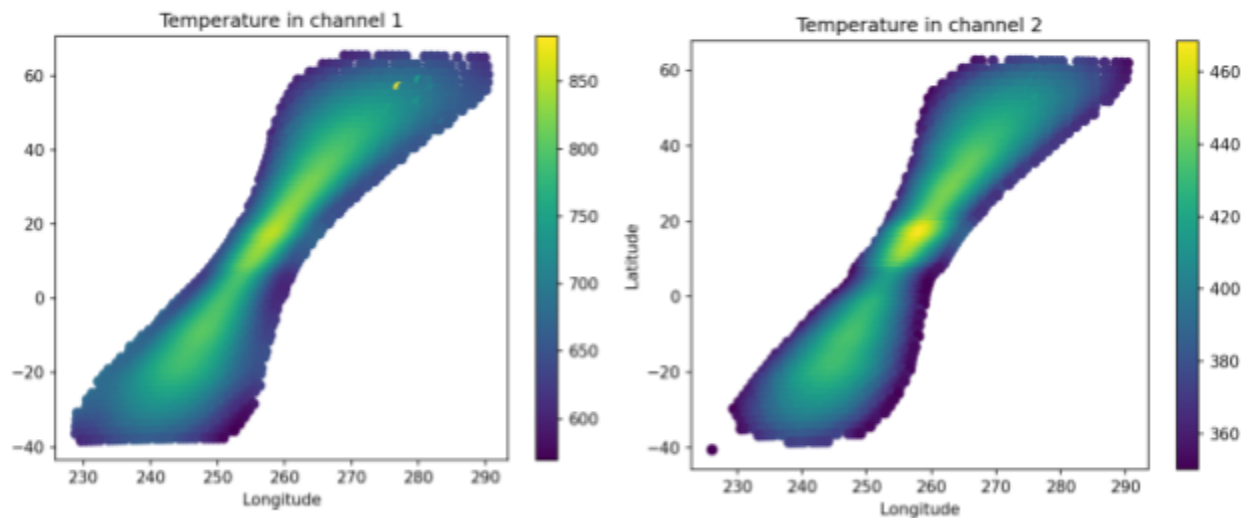
Once we have plotted the boresights, we need to begin overlaying the temperature data from the six frequency channels to make observations about Jupiter's atmosphere. We started with channel six, as it is closest to the top of the atmosphere and has the cleanest data, and repeated the same process described below for the rest of the channels.

To begin, we created a list of boresight points with the longitudes we are interested in investigating. Once we have the list, our goal is to generate a list of temperature points and local zenith angles that align with the boresights. To match or align temperature points and local zenith angles to the boresights, we created a threshold that allows us to define how close a

point must be to be considered a match and then we iterated through the data to create the lists.

We rearranged our latitude and longitude arrays into a single array where each row is a point specified by a longitude and latitude, for convenience in upcoming distance calculations. Using a simple distance formula with the latitude and longitude, we calculated the distance between each temperature point and the boresight. If this distance fell within our threshold distance, we added the temperature point to the shared indices list. This essentially lets us isolate relevant data and only plot the data points that are directly on boresights. A similar process was repeated for local zenith angles.

After isolating the relevant temperature data, we are able to create accurate plots shown below in figure 4.a-f for each of the six frequency channels. We were able to filter the data by creating a threshold temperature and eliminating temperature values that fell below the threshold. The lower temperatures can be filtered out as they occur near the edges and appear to be lower due to the high emission angle. We saved emission angles earlier in the code and are now able to filter temperature points that correspond to emission angles greater than 60° .



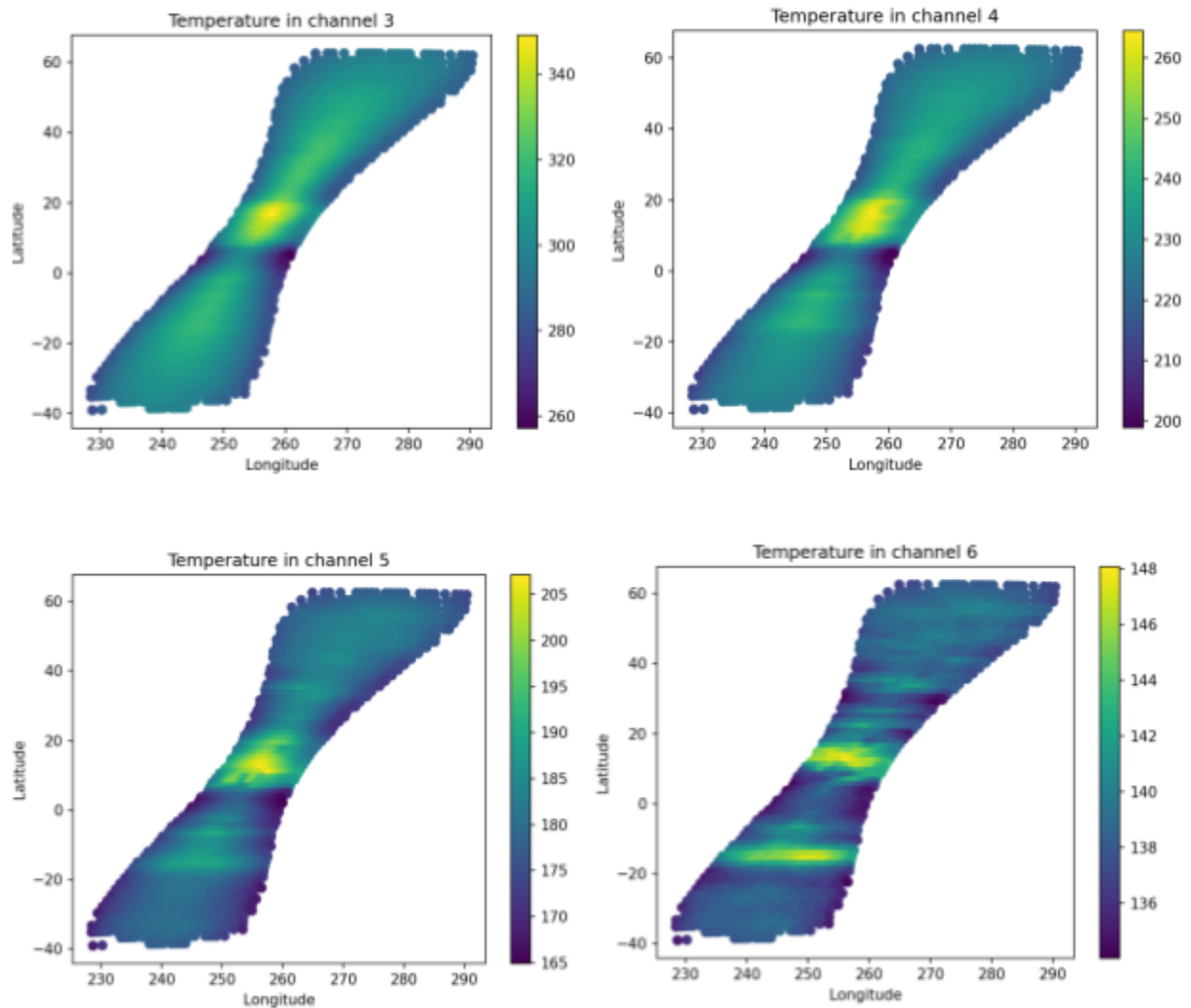


Figure 4.a-f Temperature data for channels 1-6

Findings & Conclusions

Based on the charts above, we found a few large vortices that reach deep into Jupiter, and many more that exist only closer to the surface. Overall, it seems that Jupiter's atmosphere becomes more homogenous with depth, and more varied temperature features only occur at higher altitudes. However, it is possible that this effect is due to the instrumentation rather than the atmosphere. Of the two vortices that extend deeper than the rest, one vortex is very clear, and can be seen at around 20 degrees latitude. It can be seen clearly in all channels. The other is much more difficult to see at around 40 degrees latitude. It is unclear whether this vortex exists at the depth of channel 1, but it is visible in channels 2-6. Figure 5 shows a version of the channel 2 temperature plot (figure 4.b) filtered to highlight the vortices.

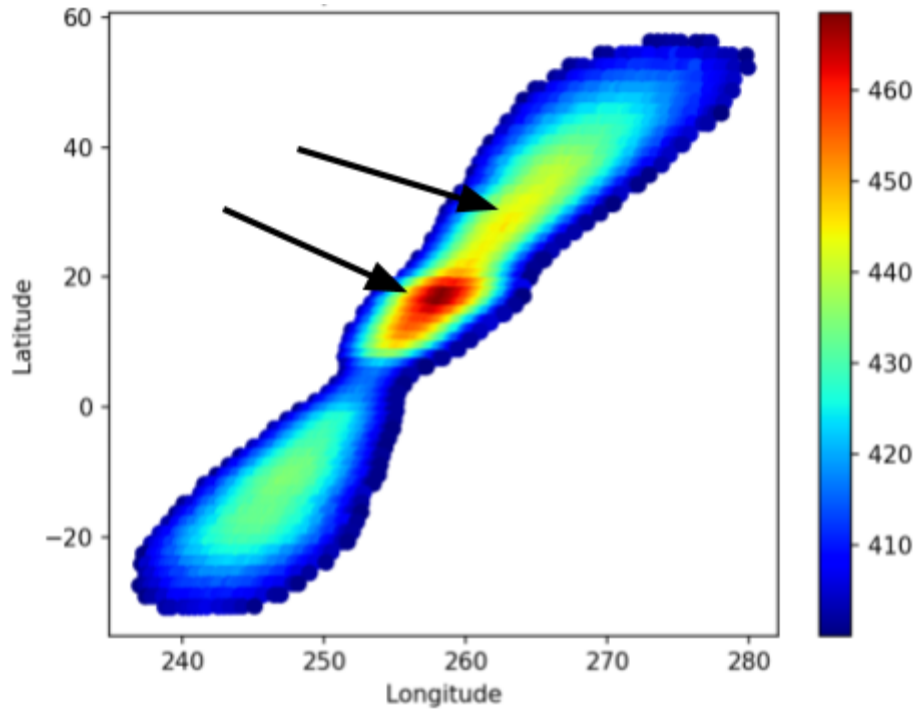


Figure 5: Channel Two Vortices

Closer to the surface, both vortices and jets could be clearly seen. In figure 6, we attempted to outline features we observed within the data. The vortices are circled, while the jet streams are enclosed in a square.

Many of these features persist to channel 5, while some only exist closer to the surface. These persistent features are highlighted in figure 7 below.

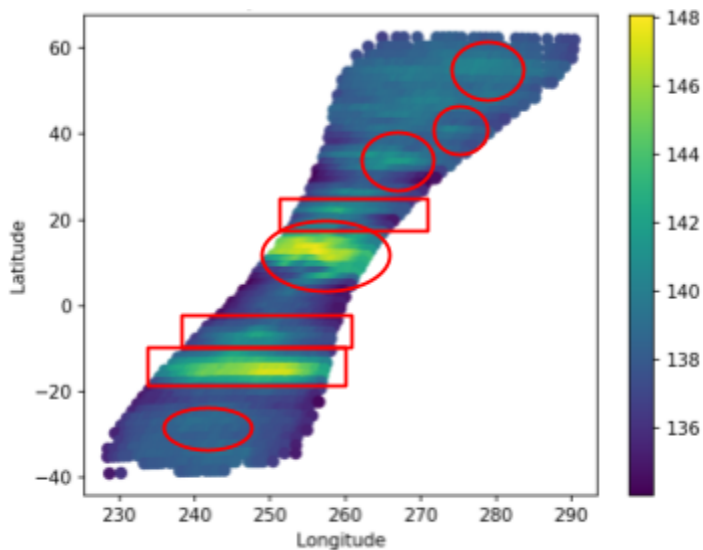


Figure 6 Temperature data for channel 6 showing bands & vortices

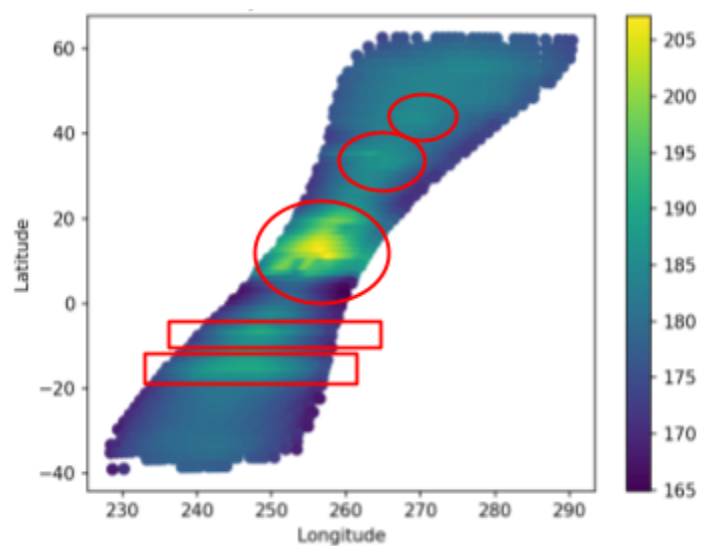


Figure 7 Temperature data for channel 5 showing bands & vortices

With this information, we can paint a more complete picture of Jupiter than we had before. The storms that we have observed help us understand how energy flows within Jupiter's atmosphere. Specifically, the temperature mapping across channels shows us how heat is distributed both within atmospheric layers and across altitudes. This allows us to better understand how these storms form and how they persist, and perhaps even apply that knowledge to the formation of storms on earth.

References:

- [1] Greicius T., Hartono N. et al, Mission to Jupiter: JUNO. Jet Propulsion Laboratory California Institute of Technology. CL#: 21-0018. <https://www.jpl.nasa.gov/missions/juno>.

- [2] Janssen, M.A. et al. MWR: Microwave Radiometer for the Juno Mission to Jupiter. Space Science Review 213, 139–185 (2017). <https://doi.org/10.1007/s11214-017-0349-5>.

- [3] <https://github.com/aviskoczylas/space-405-code>