# MathSEE Modeling Week 2019

August 19 - August 23, 2019

# 4d Visualisation of the tropopause, identification of air mass exchange and their fate

A. Gontier, B. Druot, M. Rajput, R. Wang, T. Puchka
&
T. Xiao

## 1 Short description

The Institut für Meteorologie and Klimaforschung IMK, KIT proposed us to develop an algorithm in Python to visualise the data collected from the global german HALO aircraft and the POLSTRACC campaign as well as to focus the data graph on areas of interest.

## 2 Introduction

The tropopause layer is located between the troposphere and stratosphere. The dominant part of the natural as well as anthropogenic greenhouse effect is generated in the tropopause, i.e in the altitude around 10 km. The tropopause plays a central role for the climate on earth. It is also one of the least understood atmospheric regions and is thus studied a lot. During those studies it was found that polar vortices are responsible for its extreme dynamic complexity.

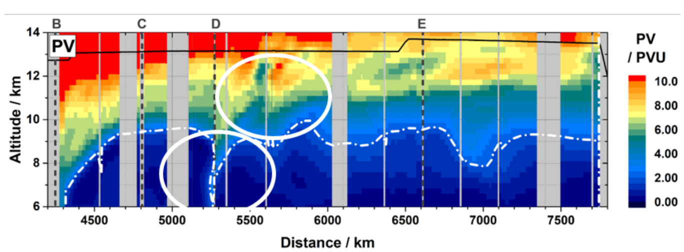


Figure 1 – Drop of altitude of the tropopause linked to the values of the Potential Vorticity

The vertical cross section of the Potential Vorticity showed a fast and sharp decrease in altitude of the tropopause accompanied by an uplift in short distance. This feature of interest is

what this subject focuses on.
The first step will focus on how to extract the data and to visualise it using Python. The second section will introduce an algorithm to identify the tropopause and extract its features. The third section will focus on the temporal evolution of the tropopause location. In the last section an algorithm will be proposed to trace the features of interest and its temporal evolution.

# 3 Extraction and visualisation of the data

The library netCDF4 is used in Python to access the data from the file provided. In order to display a graph of the data the library matplotlib is also used. It was decided to create a viewer with cursors for the time and altitude, and with the latitude and longitude as X and Y-axis. First part of the job was to have a look on the data using the netCDF4 library from python.

netCDF4 format is mostly the same as normal Python Dictionaries, with names of variables as keys and 4-Dimensional data with axes time, altitude, latitude and longitude.

To observe the data and visually be able to see the evolution of values (in our case the PV and humidity) we've created some tools using PyQt :

— First tool splices the 4D space by time and altitude, that can be controlled by using Sliders, it provided the way to choose, which variable need to be plotted and the values are shown as 2D heatmap. Using this tool we have noticed that vorticity and humidity have some kind of inverse proportionality (see part 6)

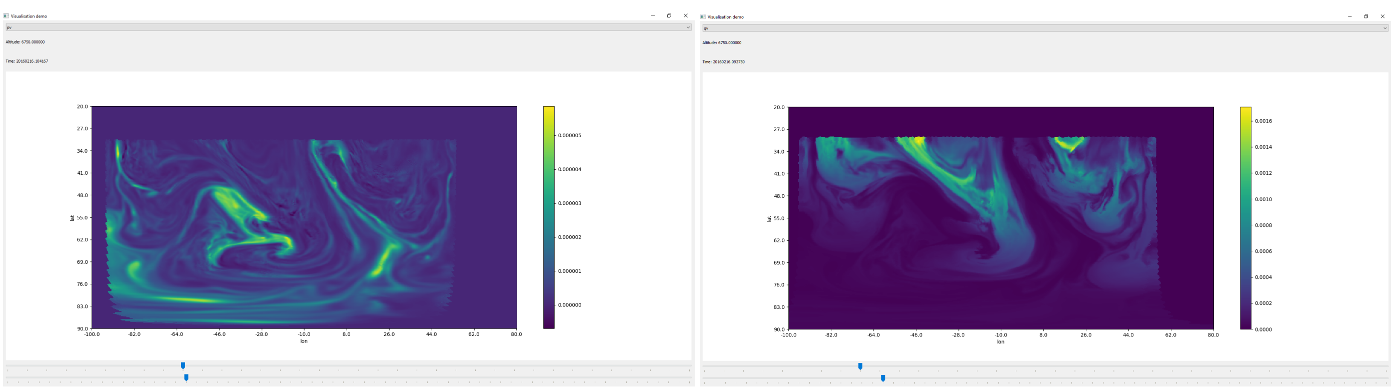

Figure 2 – Tool to slice by altitude

— Second tool was created to splice by time and longitude. The main feature of tool should have been the way to locate any structures in the longitude-altitude plane. Also only visually, because to locate something interesting computationally, it should be well-known what to locate, and how is it possible to locate it. The tool had same features as the first one, but instead of slider for altitude it has a slider for latitude

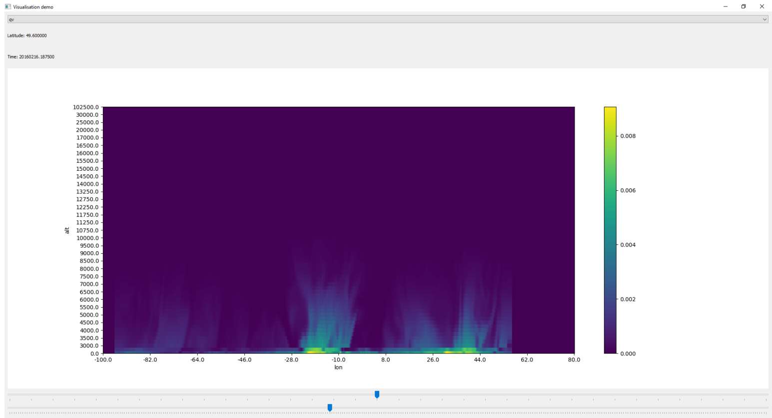

Figure 3 – Tool to slice by latitude

— Third tool was created as an attempt to plot the tropopause-stratosphere border in 3D. This border is defined as an area with potential vorticity value near to $2 \times 10^{-6}$. Because the measured data was discrete by the positions and the interpolation would take a huge amount of time that was just unpredictable, we decided to define values $a_{min}$ and $a_{max}$ so that all points with the values $pv \in (a_{min}; a_{max})$ would be counted as the tropopause-stratosphere border points.
For plots to be easier to understand the projections on (y,z) ans (x,z) planes were added.

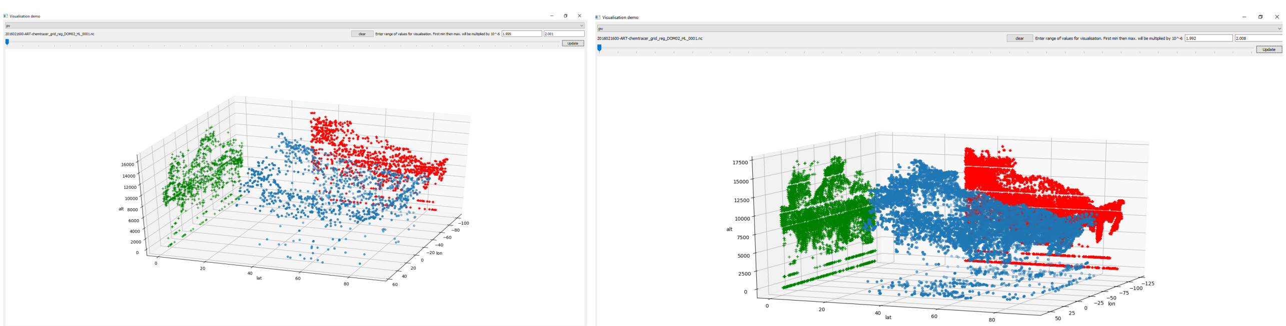

Figure 4 – 3d visualisation of tropopause-stratosphere border

— As last visualisation was taken the problem of dependencies between PV and humidity. We wanted to find out if the humidity by the points. For this issue we found all the points where the PV value from 6 hours of measurement data has deviation not more then $0.0005\%$ of tropopause value ($2 \times 10^{-6}$, actual value of deviation not more than $1 \times 10^{-11}$)

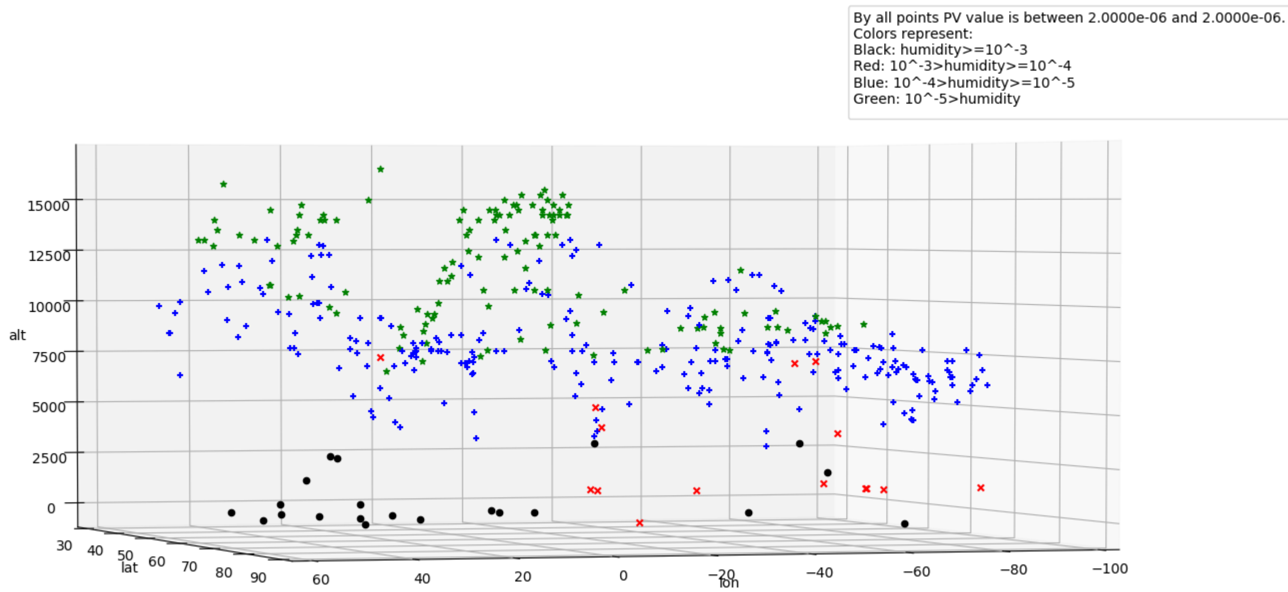


Figure 5 – Humidity in the tropopause by color

All points were plotted by their position and the colour represents the humidity value. Unfortunately directly is no system in values to see. As soon as this conclusion was made we tried to find the system not by the static PV value but to find the system by changing the PV value. This will be explained in part 6

# 4 Identify the tropopause and extraction of its features

To start the analyse, we should firstly locate the tropopause. By measuring where the value of potential vorticity is equal to 2PVU, we record the coordinate of those points. Considering the disturbance of the data, we choosed different tolerances and set a suitable range after the comparison.

| tolerance(/PVU) | $10^{-2}$ | $10^{-3}$ | $10^{-4}$ |
|---|---|---|---|
| points number | 46415 | 4545 | 461 |

Table 1 – The relationship between the number of points and the tolerance

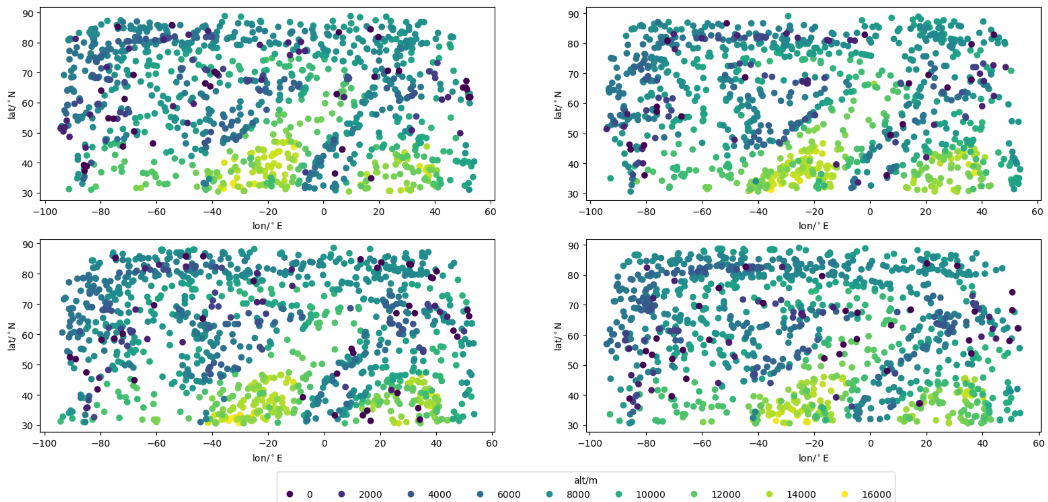


Figure 6 – Acquired Coordinates where potential vorticity$= 2PVU$

After examining the value of potential vorticity of the all points, the next step is check the other physical quantities and find out the relationship between those variables.

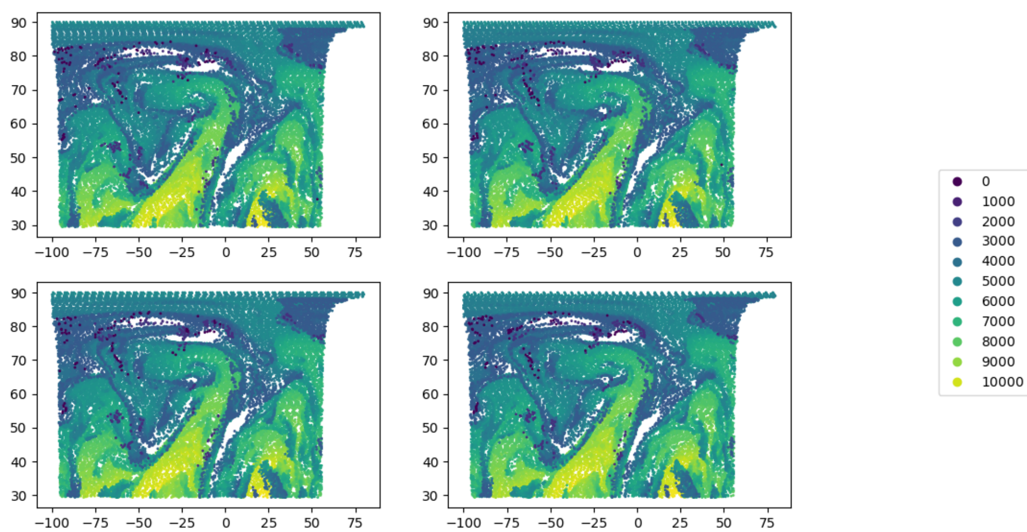


Figure 7 – Acquired Coordinates where potential vorticity$= 2PVU$

The same analysis has been done with more points and it took more time and computational power. But we still left a huge number of points to plot using laptops.

# 5 QV in tropopause

The idea was to detect tropopause using pv and once we are in tropopause, study qv. To see the structure in data of qv, we ran simulations but due to huge amount of data, limited time and computational resources it was not possible to run simulation with all qv data values inside tropopause. On laptops with these issues, we did not find any structure as it can be seen in figure 8.

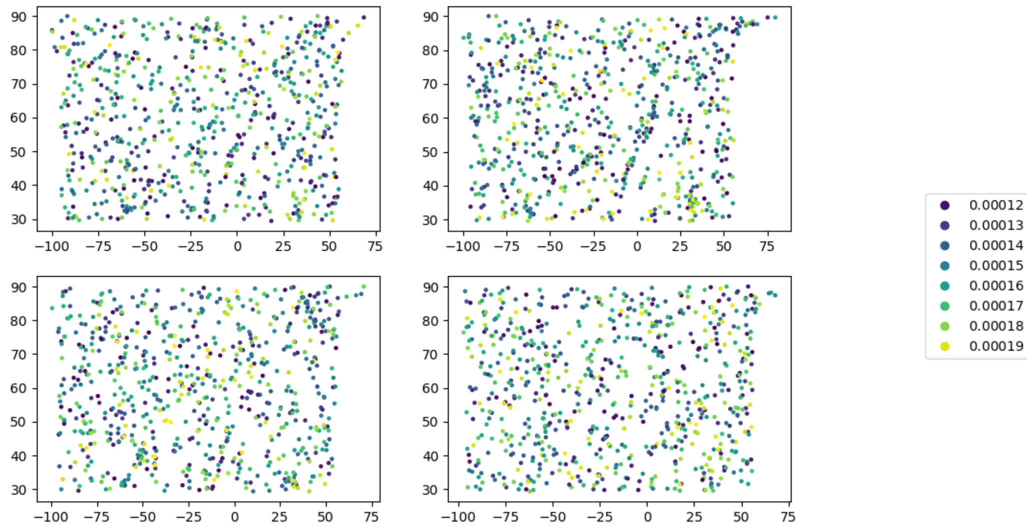


Figure 8 – Acquired Coordinates of qv where potential vorticity$= 2PVU$

# 6 PV-QV Corellation

In this part we will take a look on the PV-QV dependency that we succeeded to find.

The idea to check this two variables was born after the observation of plotting tools, that were described in part 3 The other reasons to search any kind of dependencies between values of these variables was the idea that these variables have influence on each other and the fact that on the poles potential vorticity is high and humidity is low. The inverse situation is on equator where potential vorticity is low and humidity is high

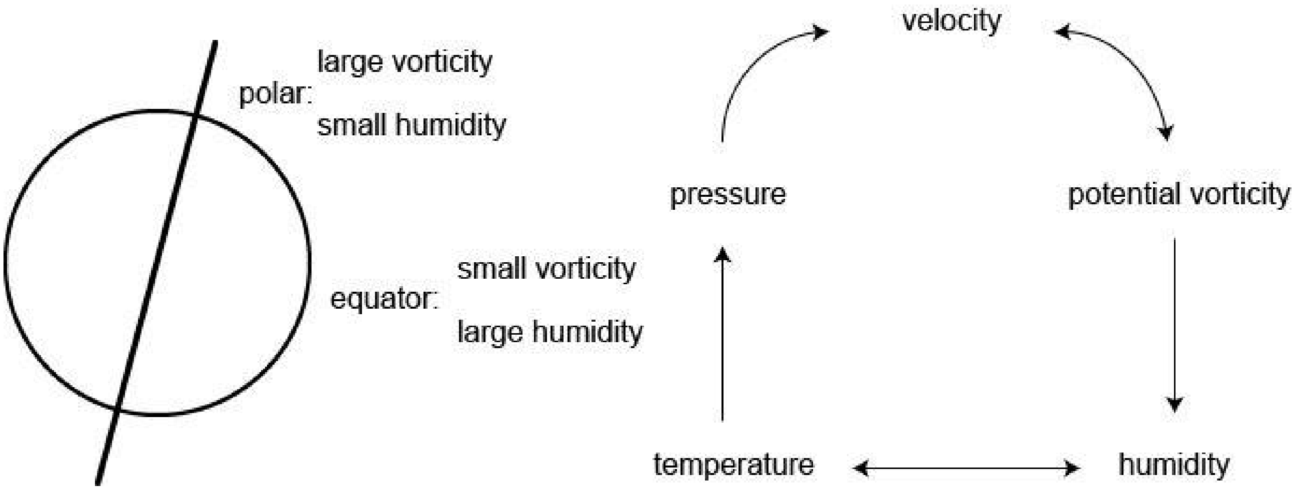


Figure 9 – The corellation of pv and hum. and a circle of influence

In one of the Bulletins of the American meteorological society was mentioned that there exist a some kind of inverse correlation between potential vorticity and humidity. For the analysis we have chosen to use only one dataset with 4 timestamps so that the calculation will take not so long time, but still will have enough data so that the noise would be a bit equalised between different values. Our analysis took 2.5 millions points with vorticity values $\in (9 \times 10^{-7}; 9 \times 10^{-6})$. For correct representation of mean values of humidity we had to find out, which step will represent the values smooth, the calculation should not take long time and at the same time the step should not be large, because then we loose precision As such step that satisfies all these conditions was $step_{pv} = 1 \times 10^{-8}$. For each interval of $pv \in (a_i; a_{i+1})$, where $a_{i+1} = a_i + step_{pv}$, was defined the mean value of humidity of corresponding points and then plotted.

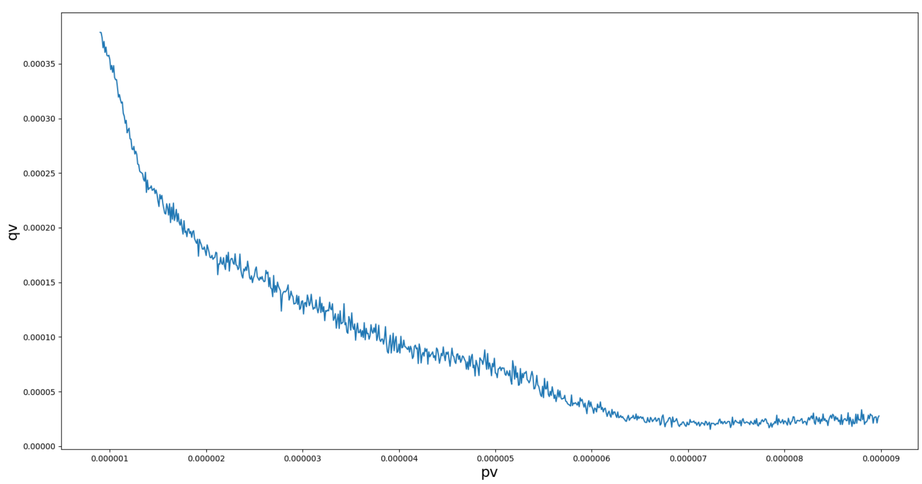


Figure 10 – Plot of mean values of humidity by different pv values

This plot does look like an exponential function so we have taken the function as $humidity =$

$a * \exp(-b * pv)$. After defining the $a$ and $b$ values we have got the function $humidity \approx 0.00042 * \exp(-392033 * pv)$

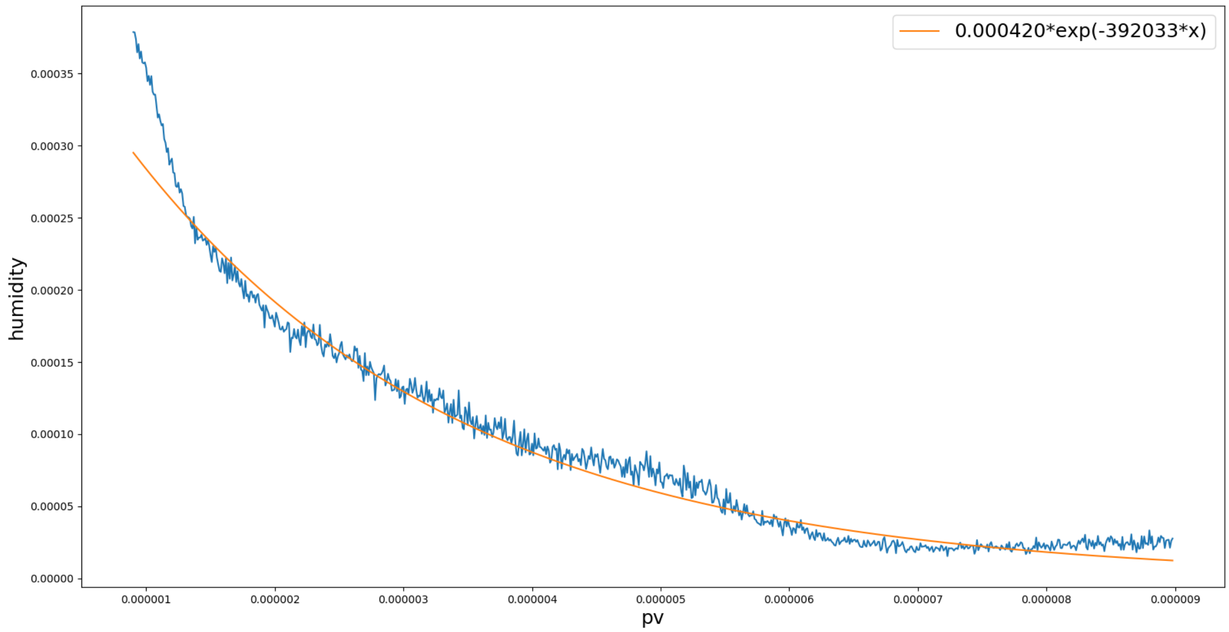


Figure 11 – Approximation of PV-Humidity correlation

# 7 Tracing of the feature of interest and its temporal evolution

The aim of this ection is to track interesting features in the data i.e drops and quick increase of the altitude of the tropopause. In order to find those, we took a look at the values of the humidity qv and tried to use different methods to automatically find the features. The first method is using directly the values of the humidity in all the domain and the second method needs to calculate the gradient in the domain and track the high values of this gradient.

## 7.1 Tracer using mean value

First, a method based on the mean value of qv was used. The mean value of the data file was calculated for the whole domain. It was then analysed by slices in latitude and longitude. The mean value was calculated for each slice. If the calculated value was higher than twice the mean value for the whole domain, this slice was considered to be interesting. Thus, the algorithm returned 2 lists of indices : one for the latitude and the other for the longitude.

This algorithm was then tested on one data file and returned 2 lists of indices that correspond to the following values :

-Latitude : 29.4 to 48.6

-Longitude : 39.4 to 43.2 Therefore the qv values were checked on those longitude and latitude to validate the algorithm. The result on the latitude 22.4 and the longitude 42.0 were plotted below :

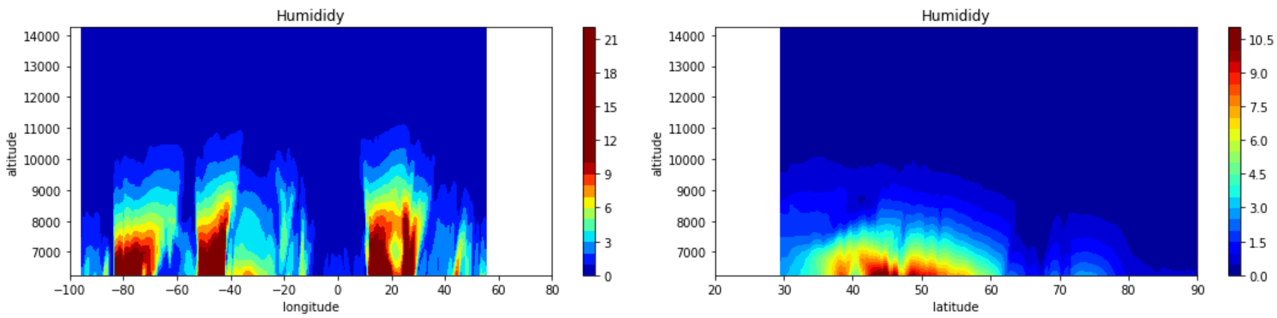


Figure 12 – Tracked feature with mean method on latitude = 22.4 (left) and longitude = 42.0 (right)

The figure on the left shows 3 interesting features that can be tracked over time however the figure on the right doesn't contain any features. Indeed, this method tracks areas where the qv value is high, but it doesn't mean that there is a quick drop of the humidity on those places. Thus, a gradient oriented method was implemented in order to track more efficiently the features.

## 7.2 Tracer using gradient

At this point, a gradient oriented method was used to track the drops and increase of qv on the data file thanks to the function gradient of the library numpy. This function calculates the gradient with the order 1 finite difference method :

$$\begin{cases} Grad(x_i) = \frac{f(x_{i+1})-f(x_i)}{h} + \mathrm{O}(h) & Boundaries \\ Grad(x_i) = \frac{f(x_{i+1})-f(x_{i-1})}{2h} + \mathrm{O}(h^2) & Otherwise \end{cases}$$

This function was used to calculate the 2 dimensional gradient on each latitude and longitude at a given altitude. Then, points which respect the relation below are considered as interesting ones :

$$Grad(x_i, y_j) > 0.8 * max(Grad(:,:)).$$

This equation is verified for both latitude and longitude direction. So, 2 lists of coordinates are obtained :

- high values of the gradient through the longitude direction
- high values of the gradient through the latitude direction

Those list of point enable to draw the corresponding slices for an altitude of 7.5km on the figure below :

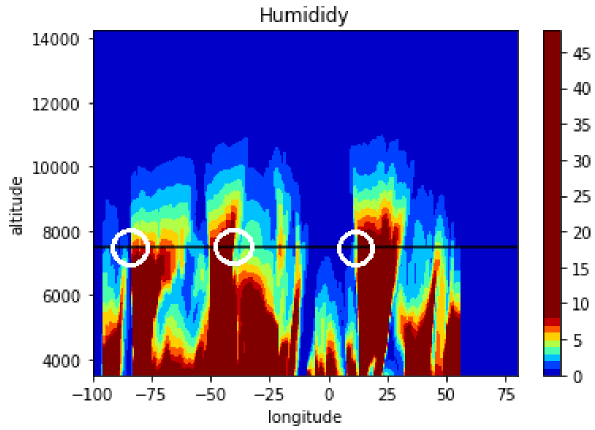


Figure 13 – Given slice from tracker on longitude axis gradient

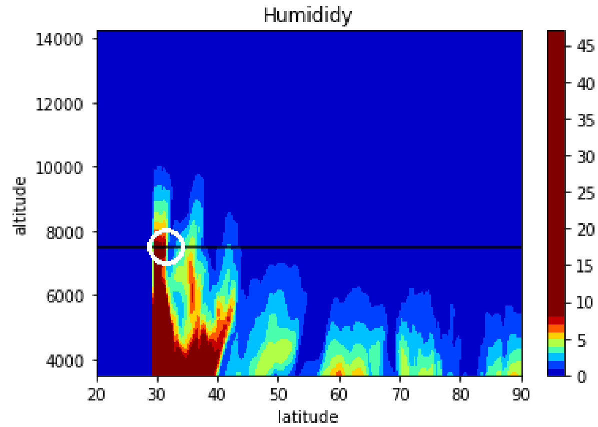


Figure 14 – Given slice from tracker on latitude axis gradient

The circle on the figures represents points qualified as interesting by the tracker i.e there is a big gradient on those parts. As it can be seen on the figure the tracker finds some quick decrease or increase of the gradient through both direction. Thus the method can be validated. Anyway, it still can be improve by tracking those point over time and check if they are still points of interest. This might be a way either invalidate the method, either to track this feature over time.

## 8 Conclusion

The viewer implemented for 2d and 3d is working well, it can be long to compute with the huge amount of data given. The localisation, extraction and temporal evolution algorithm of the tropopause takes a lot of time to compute and can be improved. A correlation between Potential Vorticity and Humidity was found. This link is coherent with scientific results[1]. The gradient based algorithm to detect the features can also be improved to detect the features more accurately and to compute more efficiently.

## References

[1] Spar, J. (1943). The correlation between specific humidity and potential vorticity in the atmosphere. Bulletin of the American Meteorological Society, 24(5), 196-200.