Lagranto - user guide for FTP service

Michael Sprenger and Heini Wernli

September 13, 2024

Contact: Michael Sprenger
Institute for Atmospheric and Climate Science
8092 ETH Zurich
Switzerland
E-Mail: michael.sprenger@env.ethz.ch

Contents

1 Synopsis

Lagranto is a tool that allows kinematic airmass trajectories to be computed. The driving wind fields are available from different data sources based on the Integrated Forecasting System (IFS) of the Euopean centre for mid-range weather forecasts (ECMWF; www.ecmwf.int). In particular, the operational analysis and deterministic ECMWF forecasts are available. Further, for climatological analysis the ERA-Interim data set (1979-2016) and ERA5 (1979-2023) can be used. In addition to the paths taken by air parcels, several meteorological fields can be traced along the trajectories, thus allowing for a refined analysis of the meteorological processes taking place along the air parcels' path.

The aim of this user guide is to show how kinematic trajectories can be calculated with the Lagranto FTP service, i.e., without the need to install Lagranto locally on a computer. This service complements the standalone version of Lagranto, which offers additional features, is available for several NWP models (ECMWF's IFS; COSMO; WRF; UM; ICON), and is freely available from this report's author on request.

2 Terms of usage

Lagranto (this FTP service) can be used freely under the license included in the output files of a calculation. Essentially, it is free to use it for research, but we don't take any responsibility for any output based on Lagranto trajectories.

If Lagranto trajectories contribute significantly to a research project, the following two articles must be referenced:

- Wernli, B. H. and H. C. Davies, 1997: A lagrangian-based analysis of extratropical cyclones. I: The method and some applications. Quarterly Journal of the Royal Meteorological Society, Vol. 123, 467–489. doi:10.1002/qj.49712353811.
- Sprenger, M. and H. Wernli, 2015: The LAGRANTO Lagrangian analysis tool – version 2.0. Geoscientific Model Development, Vol. 8, 2569-2586. doi:10.5194/gmd-8-2569-2015.

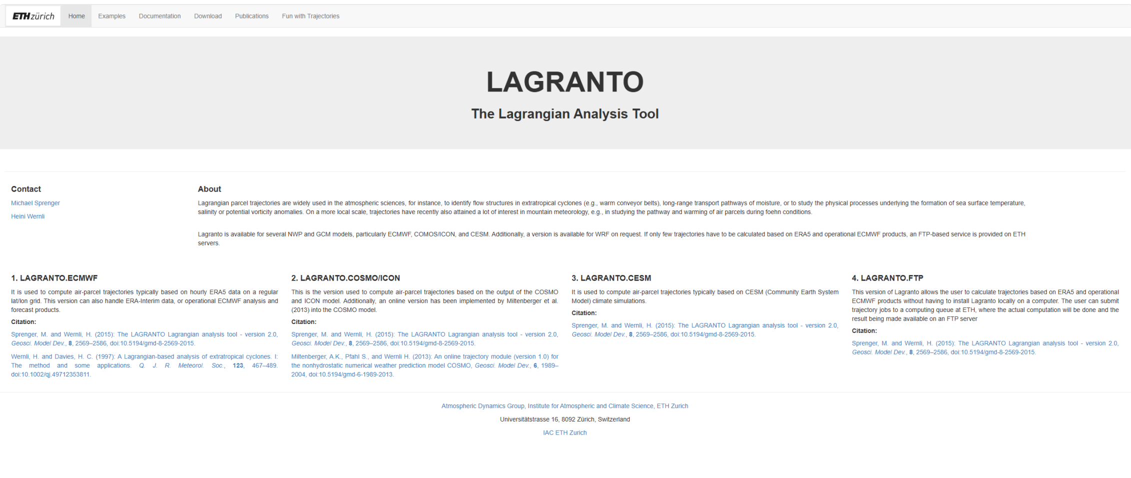

Figure 1: Main Lagranto webpage at URL www.lagranto.ethz.ch.

3 Tutorial I - first forward trajectories

The aim of this section is to show step by step how forward trajectories can be computed with Lagranto's FTP service. To this aim, we consider the simple case of trajectories started at one single position (mount Agung in Indonesia) but on several vertical levels. The data basis for the trajectory calculation is the ERA5 reanalysis

3.1 Preparing the job file

Lagranto's FTP service is controled by means of a single ASCII file that has to be submitted to an FTP server. In the specific example of this tutorial, the job file looks as follows:

```
#EMAIL
michael.sprenger@env.ethz.ch
#DATE
20171120_00
#TIME
120
#STREAM
era5
#TRACEVARS
T Q PV
#OUTPUT
TRA.PNG
#POSITION
115.508 -8.342 700
115.508 -8.342 600
115.508 -8.342 500
115.508 -8.342 400
115.508 -8.342 300
115.508 -8.342 200
115.508 -8.342 100
```

The parameters for the Lagranto listed in the job file have the following meaning:

- **#EMAIL**: A valid e-mail address; as soon as the Lagranto calculation is done, a notification e-mail will be sent to this address.
- **#DATE**: Starting date of the forward trajectory in the format {year}{month}{day}_{hour}, i.e. in this example the trajectory is started at 00 UTC 20 November 2017.
- **#TIME**: Time range of the trajectory in hours; in this case the trajectories are calculated 120 h forward in time, starting at 00 UTC 20 November 2017. If the time range is negative, backward trajectories are calculated instead.
- **#STREAM**: ERA5 data base to be used for the trajectory calculation. For a detailed description of the data base (horizontal resolution; number of vertical levels; ...) see the corresponding section in the reference list.
- **#TRACEVARS**: List of meteorological fields that will be traced along the trajectories; in this case the fields are: temperature (in deg C); specific humidity (in g/kg); and potential vorticity (in PVU). A full list of meteorolgical fields available for tracing can be found in the reference guide.

- **#OUTPUT**: Description of files included in the final output of Lagranto's calculation; here, the ASCII files with the single trajectories are included (TRA) and a PNG image for quick visualization (PNG). Note that the different options are listed separated by a dot.
- **#POSITION**: List of starting positions of the forward trajectories; the list entries (three columns) correspond to geographical longitude and latitude (columns 1 and 2), and pressure (in hPa; column 3). Each line defines a new starting point, i.e. here trajectories are started at the geographical location of Mt. Agung (115.508 E / 8.342 S) and pressure levels range from 700 to 100 hPa.

3.2 Submitting the job file

Next, the job file must be placed on Lagranto's FTP server. The FTP server's address is

```
ftp://iacftp.ethz.ch/pub_write/sprenger/lagranto.ftp.service/
```

There are different ways how this can be accomplished. In the following we assume that the job file is saved with a filename 'jobfile'. Then, one of the following options can be used:

- **Command line ftp**: The following commands have to be entered in the command shell:

```
> ftp iacftp.ethz.ch {login: anonymous; pass word: e-mail adress}
> cd /pub_write/sprenger/lagranto.ftp.service/
> mput jobfile
> exit
```

- **GUI-based FTP clients**: There exist many different FTP clients that make the transfer of the jobfile still easier. Two common examples (available for Windows, Linux and Apple) are gFTP (www.gftp.org) and FileZilla (filezilla-project.org). The advantage of these clients is that they offer an intuitive user interface, compared to the command-driven 'ftp' transfer presented above.
- **Python interface**: In section 7.3, a simple Python interface to Lagranto.FTP is presented. It allows for a command-line interaction with the Lagranto.FTP server. Specifically, for the example the Python command to submit a job is:

```
> ./control.py put jobfile   [to submit to the Lagranto.FTP queue]
```

Note that you will *not* see the file in the FTP folder, but it is still there and will be forwarded by the FTP server to Lagranto's production machine. The fact that you can't see your file is a security issue.

3.3 Checking the status of calculation

A short while (1-2 min) after the jobfile has been transferred to the FTP server, the Lagranto job will be queued for processing. The status of processing can be checked at the following site:

```
ftp://iacftp.ethz.ch/pub_read/sprenger/lagranto.ftp.service/
```

As an example, Figure 2 shows the screenshot of the site. The status of the just submitted job is 'waiting'. Other jobs (the ZIP archives) have successfully been completed and wait to be retrieved by the user. If the trajectory calculation has started, the status on the site will change to 'running',

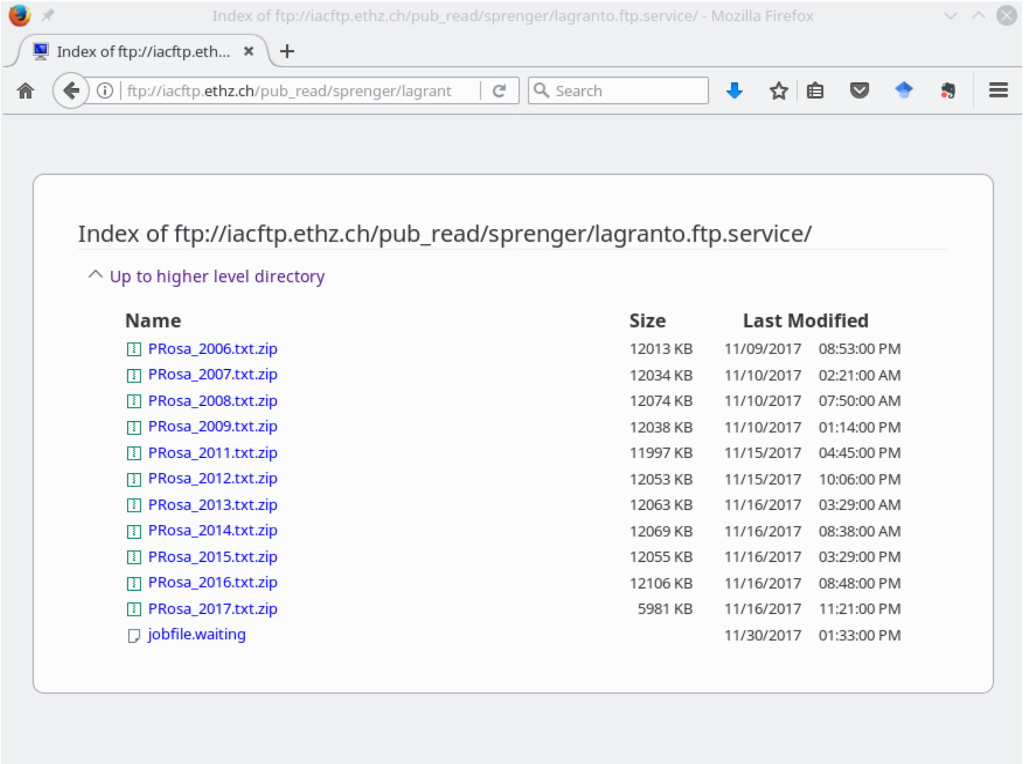

Figure 2: Screenshot of FTP site where the status of a job can be seen.

until finally the resulting ZIP archive is be placed in the output directory. An e-mail will be sent to the user as soon as the calculation is done, including an notification that the output ZIP archive will be deleted automatically after 10 days.

The status of the Lagranto.FTP queue can also also be obtained with the Python interface:

```
> ./control.py stat    [to get status of Lagranto.FTP queue]
```

3.4 Retrieving the trajectories

All resulting output files are included in a ZIP archive that can be retrieved, as notified in the e-mail, at

```
ftp://iacftp.ethz.ch/pub_read/sprenger/lagranto.ftp.service/jobfile.zip
```

Again, there are different ways how the ZIP file can be retrieved:

- **Command line ftp**:

```
> ftp iacftp.ethz.ch {login: anonymous; pass word: e-mail adress}
> cd /pub_read/sprenger/lagranto.ftp.service/
> mget jobfile.zip
> exit
```

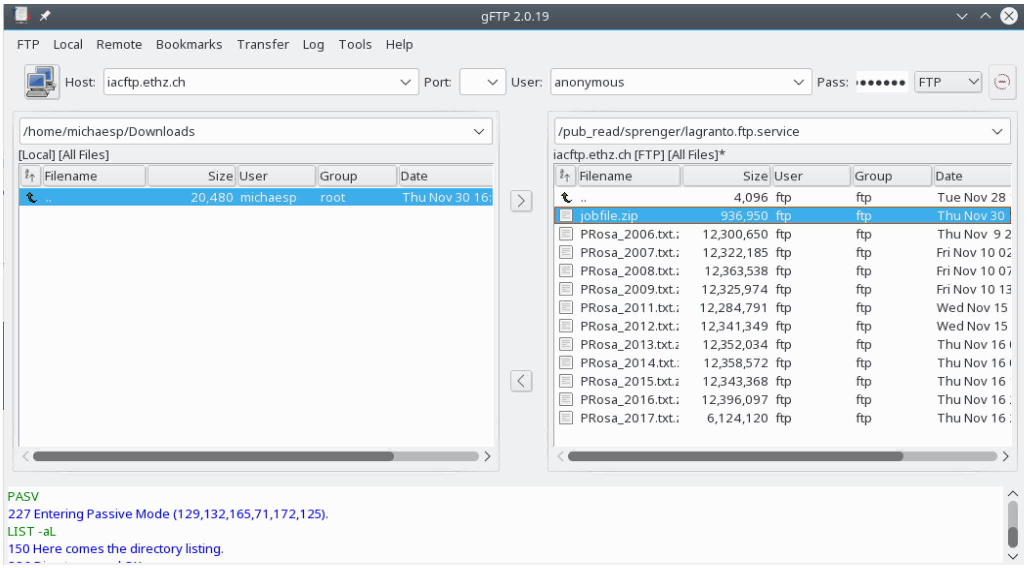

Figure 3: Screenshot of 'gftp' FTP client to access the ZIP archive of the trajectory calculation.

- **wget**:

```
wget ..
..ftp://iacftp.ethz.ch/pub_read/sprenger/lagranto.ftp.service/jobfile.zip
```

- **GUI-based FTP clients**: As an example Figure 3 shows the GUI of the gftp client.
- **Python interface**: In section 7.3, a simple Python interface to Lagranto.FTP is presented. It allows for a command-line interaction with the Lagranto.FTP server. Specifically, for the example the Python command to retrieve the calculated trajectory is:

```
> ./control.py get jobfile   [to get the trajectory from Lagranto.FTP]
```

The result of the trajectory calculation are all saved in a ZIP archive that must be unpacked. The contents of the ZIP archive in this specific example is shown in Figure 4. The unpacking (on a Linux machine) can be accomplished with the command:

```
> unzip jobfile.zip
```

As requested in the jobfile (see section #OUTPUT) the trajectory file 'trajectory_20171120_00' and two PNG images are included. Further, a license agrrement ('LICENSE.pdf') specifies the conditions under which Lagranto trajectories can be used (and that no warranty is taken); the jobfile ('jobfile') initially submitted to the FTP server is included; and the complete logfile of the Lagranto calculation ('jobfile.log') helps particularly if the calculation (or some steps of the calculation and/or visualization) failed.

3.5 Interpretation of results

The single trajectories are available in the ASCII file 'trajectory_20171120_00', where the date in the filename corresponds to the starting date and time of the trajectory. The first few lines in this file are as follows:

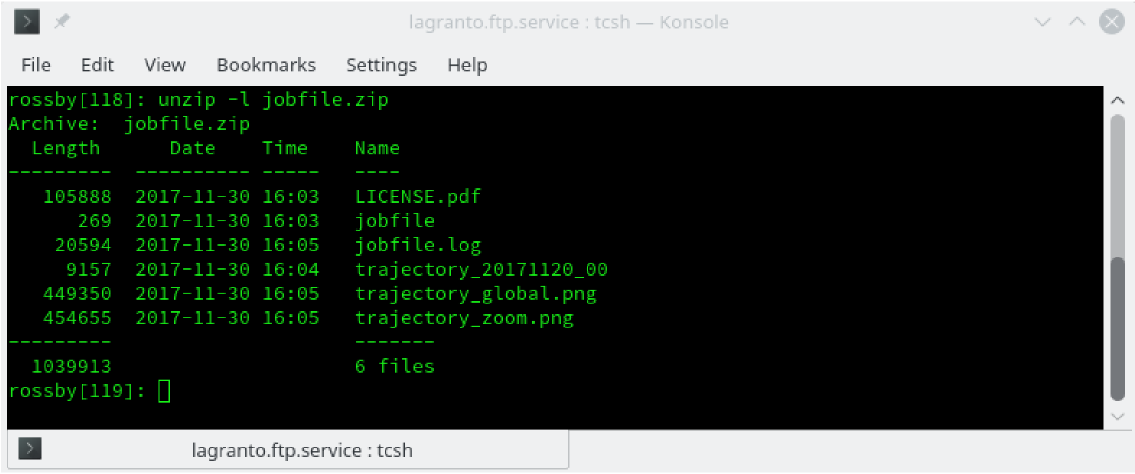

Figure 4: Contents of the ZIP archive after the trajectory calculation. Note that the contents of a ZIP archive can be listed, as discernible in the screenshot, with the Shell command 'unzip -l jobfile.zip'.

```
Reference date 20171120_0000 / Time range    7200 min

  time      lon     lat     p        T         Q        PV
-----------------------------------------------------------

   0.00   115.51   -8.34   700    10.107     8.367    -0.189
   6.00   115.54   -8.81   701     9.625     8.734    -0.182
  12.00   115.60   -9.33   706    10.909     8.390    -0.203
  18.00   115.66   -9.81   706    11.044     8.072    -0.210
  24.00   115.46  -10.19   699    10.151     8.784    -0.206
  30.00   115.18  -10.74   691     9.386     8.482    -0.225
   ...      ...     ...    ...      ...       ...       ...
 102.00   104.42  -11.16   722    11.568     9.587    -0.431
 108.00   103.65  -11.13   730    11.664     9.776    -0.353
 114.00   102.66  -11.18   728    11.745     9.824    -0.333
 120.00   101.37  -11.19   750    13.576    10.016    -0.337

   0.00   115.51   -8.34   600     3.411     5.991    -0.177
   6.00   115.46   -9.06   598     2.819     6.213    -0.173
   ...      ...     ...    ...      ...       ...       ...
```

The first line gives the starting date and time of the trajectory calculation, followed by the time range in minutes. Then the single trajectories, corresponding to the starting positions, are listed one after the other. The first four columns are the same for all trajectory files: 1) the time in hour and mintes since the reference date/time (i.e., an entry 120.00 mean 120 h and 0 min after the reference date/time); 2) and 3) the geograpical longitude and latitude, respectively; and 4) the pressure (in hPa). The remaining columns depend on the meteorological fields listed in the job file for tracing along the trajectories (see #TRACEVARS). In this specific example, these are temperature (in deg C), specific humidity (in g/kg) and potential vorticity (in PVU).

In addition to the ASCII trajectory file, a PNG image was requested in the job file with the #OUTPUT option. A simple visualization is provided for two geographical domains. The first one is global, the second one zoomed into the region that is actually covered by trajectories, i.e. –

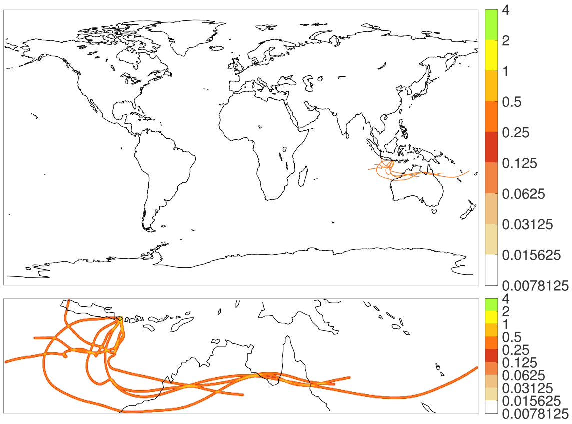

Figure 5: Global and zoomed-in visualization of trajectories.

more specifically – the range of longitude and latitudes of all trajectories determine the zoomed-in region. In the figure itself, a trajectory density is shown with a logarithmic color scale. Note that in the case of a single trajectory, or if several trajectories do not cross or cover the same region, the single trajectories appear as simple lines. The global and zoomed-in figures are shown in Figure 5.

4 Tutorial II - simple climatologic analysis

The aim of this section is to show how Lagranto's FTP service can be used to identify distinct Lagrangian flow features and a climatology thereof can be compiled. More specifically, we formulate the following problem:

> *We would like to identify all air streams in the North Atlantic starting at level 850 hPa and ascending within 48 h at least by 500 hPa. This analysis should be done for the 1-month time period 1-31 Januar 2009. For further analysis, the following output is requested: (i) the single trajectory files for all starting dates; (ii) a netCDF file that includes the gridded density of all trajectories; and (iii) a PNG image that offers a quick view on the trajectory density.*

The job file corresponding to this request is listed below, and the single entries will be discussed:

```
#EMAIL
michael.sprenger@env.ethz.ch
#DATE
from 20090101_00 to 20090201_00 by 6
#TIME
48
#STREAM
erai
#TRACEVARS
TH Q PV
#OUTPUT
TRA.NETCDF.PNG
#STARTF
box.eqd(-100,20,30,90,80)@level(850)@hPa
#SELECT
GT:p(DIFF):500:0,48
```

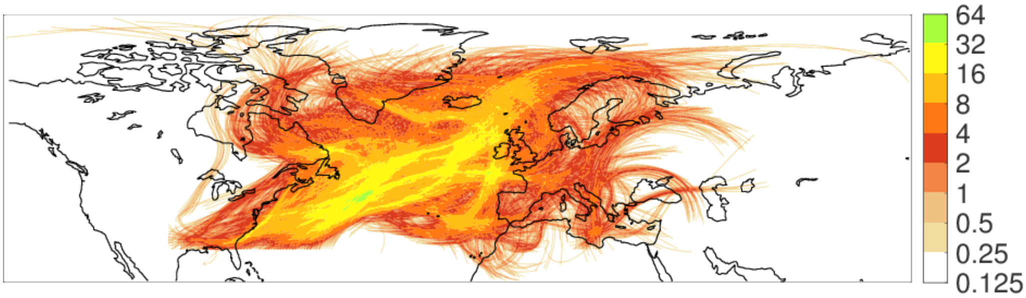

Figure 6: Zoomed-in visualization of trajectories (density) for ascending air streams.

The first difference compared to the example in the previous section concerns the #DATE specification. In this example, trajectories are not only started for one single time instance, but every 6 h between 1-31 January 2009. Further, the time range is smaller than before: forward trajectories now only cover the time period 48 h after their release. The driving winds are taken from the ECMWF ERA-Interim reanalysis data set, which can be accesses with Lagranto for the time range 1979-2016. With respect to the #OUTPUT specification only a minor change compared to the first example is included. In addition to the trajectories and a PNG image, a gridded trajectory density is provided in netCDF format.

The last two entries in the job file are the most interesting ones. In fact, starting positions for the trajectories are not given as a list of (lon,lat,p) coordinates, but as a simple pseudocode (see section 6 for further details). To handle the request stated above, we start trajectories at 850hPa in a rectangular (lon,lat) box, where longitudes and latitudes cover 100 W to 20 E and 30 N to 90 N, respectively. We also require the starting positions within this geographical box to be equidistant (80 km), thus avoiding any "distortion" of the trajectory starting points due to the poleward convergence of meridians. In summary, the request for the starting positions can be written as:

```
box.eqd(-100,20,30,90,80)@level(850)@hPa
```

Of course, most of the trajectories calculated 48-hour forward in time will never ascend by at least 500 hPa within 48 h. To select specifically these air streams, a further entry is included in the job file (#SELECT). Here, a selection criterion can be specified in a pseudocode (see section 6 for details):

```
GT:p(DIFF):500:0,48
```

The difference in pressure [p(DIFF) in the code] between time 0 and time 48 h is required to be greater (GT) than 500 hPa. The final outcome of this short climatologic analysis is shown in Figure 6.

5 Tutorial III - Starting at multiple times

In some applications, backward or forward trajectories have to be computed from the same location for many (consecutive) times. This is, for example, the case if continuous vertical soundings are available at one measurement sitze and the origin of the air along this profile should be studied as a function of time.

The Lagranto examples so far assumed that all trajectories can be started at exactly the same time (in Tutorial I), or a multitude of trajectories are started on consecutive times (as in Tutorial II). If, however, only few (100-500) trajectories have to be started at a time, the method presented in Tutorial II becomes very inefficient. The reason for this becomes evident in the following illustrative example. Suppose 100 24-hour ERA5-based trajectories along a vertical profile shall be released at two consecutive times: 00 UTC and 06 UTC. The method presented so far would read in 24 ERA5 files for the 00 UTC calculation, and again 24 ERA5 files for the 06 UTC calculation. Hence, in total 48 ERA5 would have to be read by Lagranto. However, in both calculations there is a 23-hour overlap in ERA5, and if the Lagranto calculation could be run in one go, the number of ERA5 readings can be reduced to 23+2=25. The reduction of ERA5 files to read goes along with a substantial increase in Lagranto performance. If not only two, but many consecutive time steps are to be calculated, the performance gain can be huge.

Lagranto supports this multi-time starting. A corresponding job file would look as follows:

```
#EMAIL
michael.sprenger@env.ethz.ch
#DATE
20171120_00
#TIME
120
#STREAM
era5
#TRACEVARS
T Q PV
#OUTPUT
TRA.PNG
#HOUR+POSITION
 0.00 115.508 -8.342 700
 6.00 115.508 -8.342 600
12.00 115.508 -8.342 500
...
```

There are two important changes compared to the job files presented so far: First, the date specified with #DATE does no longer give the starting date of the trajectories, but is a reference date that corresponds to hour 0; second, with #HOUR+POSITION not only the longitude, latitude and pressure of the trajectoriy is specified (in columns 2-4) but in the first column the starting time (in hours) relative to the reference date #DATE.

6 Tutorial IV - visualizing trajectories

In this section it is discussed how single trajectories can be visualized. In contrast to the PNG and PDF images that can be requested in the #OUTPUT specifier, the aim here is not to get trajectory densities, but to read in single trajectory files into Matlab or Python and then to plot the trajectories on a geographical map. The Matlab and Python scripts needed to do so can be obtained from either the Lagranto webpage (see Figure 1), or they can be requested as extra output (tag MATLAB in #OUTPUT). Adapting these scripts allows the user to optimally adapt the trajectory analysis and visualization to his or her need. As a further way how to visualize single trajectories, see the entry 'KML' in the reference guide.

6.1 Visualization with Matlab

Two Matlab scripts are used to visualize Lagranto trajectories. The first one reads in a trajectory file into a Matlab structure, the second one then plots these trajectories. The single steps are as follows, if we assume that a trajectory file with filename 'trajectory_20090131_06' is available.

```
> tra = read_trajectory('trajectory_20090131_06')
tra =

      info: {'Reference date 20090131_0600 / Time range    2880 min'}
    nfield: 7
     field: {'time'  'lon'  'lat'  'p'  'TH'  'Q'  'PV'}
     ntime: 9
      ntra: 58
      time: [522x1 double]
       lon: [522x1 double]
       lat: [522x1 double]
         p: [522x1 double]
        TH: [522x1 double]
         Q: [522x1 double]
        PV: [522x1 double]
     label: [522x1 double]
```

As soon as the trajectories are read into Matlab, a figure with a geographical map can be opened, which looks as follows in Matlab's mapping toolbox:

```
figure(1);
clf;
load coast
h=axesm('MapProjection','stereo','origin',[ 90 -60 ]);
h=plotm(lat,long,'Color','k','LineWidth',2)
gridm('--');
axis([ -0.4967    0.3790   -1.2516   -0.3758 ]);
```

It is a polar stereographic map projection, the continental outlines are plotted as a black line of width 2, and we zoom in (with the 'axis' command) to a subdomain actually covered by trajectories. The trajectories will then be plotted onto this geographical map with the Matlab script 'plot_trajectory'.

```
col.name        = 'default';
col.min         = 300;
col.max         = 900;
col.label       = [ 400 500 600 700 800 ];
col.field       = 'p';
col.orientation = 'vert';
col.linewidth   = 3;
plot_trajectory(tra,col,[ 0 24 48 ]);
```

Besides the trajectory 'tra', some options can be passed to the plotting script. In particular, the trajectories can be colored according to a meteorological field. Here, it is the pressure (col.field = 'p'), which is colored in between the range 300 hPa (col.min) and 900 hPa (col.max). If the pressure of the trajectory falls outside this domain, a dashed black line will be drawn. The colortable used is specified in col.name (col.name='default'). Finally, the width of the single trajectories can be specified (col.linewidth=3), the orientation of the colorbar be set either to vertical (col.orientation='vert') or horizontal ('hori'). The labels listed in 'col.label' will be marked in the colorbar. If no colormap is given (col.name = 'none'), the trajectories will be plotted as black

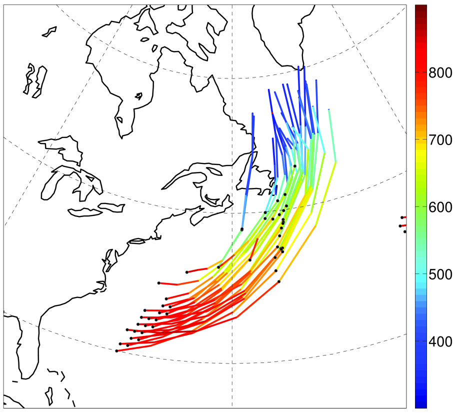

Figure 7: Single trajectories plotted with Matlab and colored with pressure in hPa (see text for details).

lines, i.e., not be colored according to a meteorological field. The third argument in the call ([0 24 48]) is a list of times (in hour) that will be marked along the trajectories with points. If the list is empty ([]) no trajectory times will be marked.

The final figure can then exported (e.g., as an EPS figure) with common Matlab commands (see Figure 10):

```
set(gcf,'PaperUnits','centimeters');
set(gcf,'PaperSize', [40 30]);
set(gcf,'PaperPosition',[0 0 40 30]);
set(gcf,'PaperPositionMode','Manual');
print('-depsc2','trajectory.eps');
```

6.2 Visualization with Python

A Python package for Lagranto trajectories can be found at URL:

```
https://lagranto.readthedocs.io/en/latest/
```

The documentation for this Lagranto package shows how to install it using 'conda' or 'pip'. Further, a worked-out example presents ways how to analyze and visualize trajectories. Note that the package also includes a section 'Calculation'. This, however, needs a local installation of the standalone Lagranto version and is not available for the Lagranto FTP service.

Suppose that we have calculated trajectories and saved them in the file 'trajectory_20090131_06'. Then, the trajectories can be read into Python with:

```
from lagranto import Tra
trajs = Tra()
trajs.load_ascii('trajectory_20090131_06')
print(trajs)
```

We would like to identify a warm-conveyor belt (WCB) in the read-in trajectories. To this aim, we test if the trajectories fulfill the typical criteria for WCB, an ascent greater than 500 hPa in

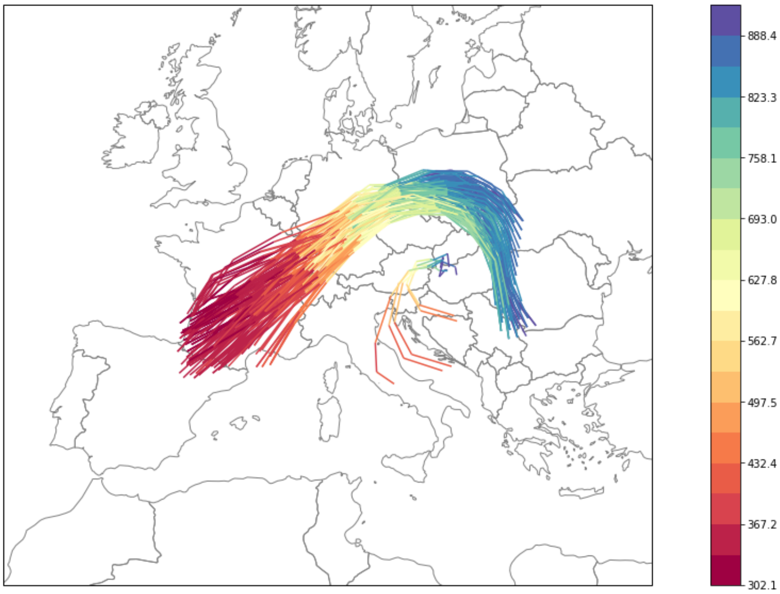

Figure 8: Single trajectories plotted with Python and colored with pressure in hPa (see text for details).

48 hours. To make it clear, the goal of this exmple is not to replace the Fortran routines of the Lagranto package but to illustrate the possibilities that python provides to analyze trajectories using a simple example.

```
wcb_index = np.where((trajs['p'][:, :1] - trajs['p']) > 500)
wcb_trajs = Tra()
wcb_trajs.set_array(trajs[wcb_index[0], :])
print(wcb_trajs)
```

Now that we have WCB trajectories, let's plot them on a map. We will use cartopy for this.

```
import cartopy.crs as ccrs
import cartopy.feature as cfeature
from lagranto.plotting import plot_trajs
import matplotlib.pyplot as plt

crs = ccrs.Stereographic(central_longitude=180 - 170,
                         central_latitude=90 - 43,
                         true_scale_latitude=90 - 43)
fig = plt.figure()
ax = plt.axes(projection=crs)
land_50m = cfeature.NaturalEarthFeature('cultural', 'admin_0_countries',
                                        '50m', edgecolor='gray',
                                        facecolor='none', linewidth=0)
ax.add_feature(land_50m)
ax.set_extent([-10, 28, 30, 60])
plot_trajs(ax, wcb_trajs, 'p')
```

As an example, see Figure 10. For further details about the Lagranto Python package, please refer to the above webpage.

6.3 Visualization in Google Earth

If the tag KML is added to the output format #OUTPUT, e.g., KML.TRA.NETCDF, a KML file will be created that can directly be imported into *Google Earth*. As an example, Figure 9 shows an ascending air stream over the North Atlantic. Note that *Google Earth Pro* also allows the user to perform flights along imported KML paths, i.e., it would be possible to fly along an imported KML trajectory. For further details, refer to the *Google Earth* documentation.

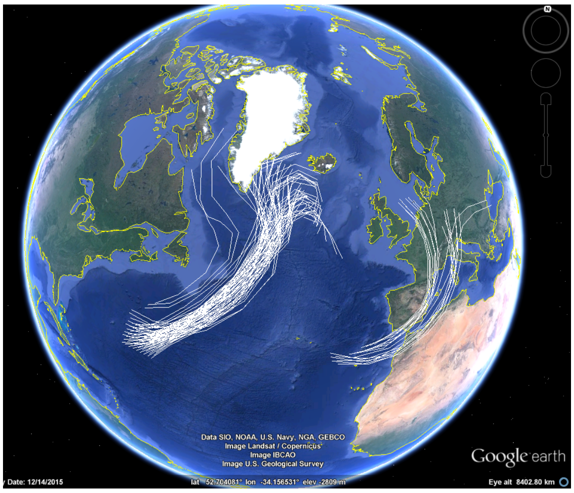

Figure 9: Ascending air streams on 00 UTC 1 February 2009 imported into GoogleEarth. The needed KML format can be requested in the output format specification.

6.4 Visualization with netCDF viewer

If a netCDF output is requested in #OUTPUT (tag netcdf), a quick view on the trajectory density can be obtained with any netCDF viewer. As a very basic viewwer, ¡ncview¿ could be used. The example in Figure 10 shows the trajectory density from Tutorial II in this viewer.

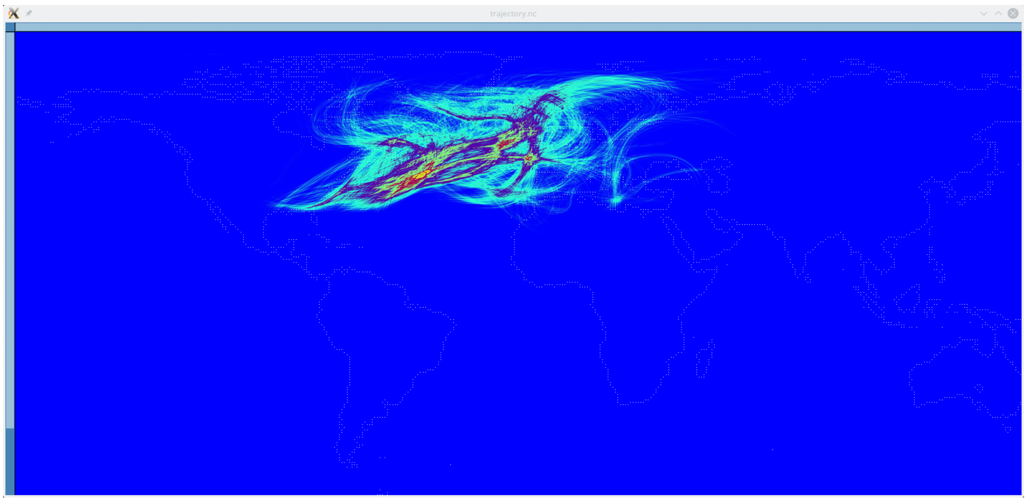

Figure 10: Quick view on trajectory density using the netCDF viewer 'ncview'.

7 Tutorial V - Studying the ascent of trajectories

In this example, we present a more refined analysis of a trajectory and its meteorological environment. Specifically, let's assume that we have a trajectory that ascends within 48 hours from a near-surface layer to the upper troposphere. From a preceding trajectory calculation, we might know that a trajectory starting at longitude/latitude -23.06 W / 39.51 E and at 990 hPa on 00 UTC 31 January 2009 will fufill this criterion (it is, actually, part of a warm-conveyort belt):

```
Reference date 20090131_0000 / Time range    2880 min

  time      lon     lat     p
-----------------------------

   0.00   -23.06   39.51   990
   1.00   -22.44   40.01   988
   2.00   -21.88   40.57   984
   3.00   -21.34   41.16   977
   4.00   -20.67   41.77   967
    ...      ...     ...   ...
  44.00   -17.09   56.05   392
  45.00   -17.06   56.59   391
  46.00   -17.00   57.16   386
  47.00   -16.91   57.75   380
  48.00   -16.76   58.39   378
```

We see that the air parcel starts at 990 hPa, and then rapidly ascends to upper-tropospheric levels, thereby moving slightly to the east and substantially to the north. It will now be of interest to see in a time/height cross-section (with time and pressure on the x and y axis, respectively) how this trajectory evolves. In addition, it will be of interest at each of the trajectory's location how the vertical profile of some meteorological variables look like. In this way, it will be possible to see, e.g., whether the trajectory is passing through a cloud and what the vertical extent of this cloud is. Other questions to be addressed in this way are: Does the trajectory a follow the isentropes in the environment? How does the trajectory approach the tropopause, or even crosses this barrier?

Lagranto.FTP allows this kind of analysis to be performed via the #LIDAR tag. As an example, a job file could look as follows:

```
#EMAIL
michael.sprenger@env.ethz.ch
#DATE
20090131_00
#TIME
96
#STREAM
era5
#LIDARVARS
OMEGA TH PV IWC LWC
#OUTPUT
TRA
#POSITION
  -23.06 39.51 990
```

Here, all fields are as before, except for the #LIAR tag. There, a list of three-dimensional variables must be listed that will be 'profiled' along the trajectory. In this case, we profile the vertical wind (OMEGA), potential temperature (TH), potential vorticity (PV), and ice and liquid-water clouds (IWC and LWC). The output of the trajectory calculation now includes the following files:

```
Archive:  jobfile.zip
  Length      Date    Time    Name
---------  ---------- -----   ----
   105888  2024-09-12 16:20   LICENSE.pdf
      149  2024-09-12 16:20   jobfile
    46267  2024-09-12 16:28   jobfile.log
     3125  2024-09-12 16:23   trajectory_20090131_00
   429728  2024-09-12 16:28   lidar_20090131_00.nc
---------                     -------
   585157                     5 files
```

The netCDF file lidar_20090131_00.nc contains the specified meteorological fields as time-height cross-sections. In a simple netCDF viewer (ncview) this looks as in Figure 11.

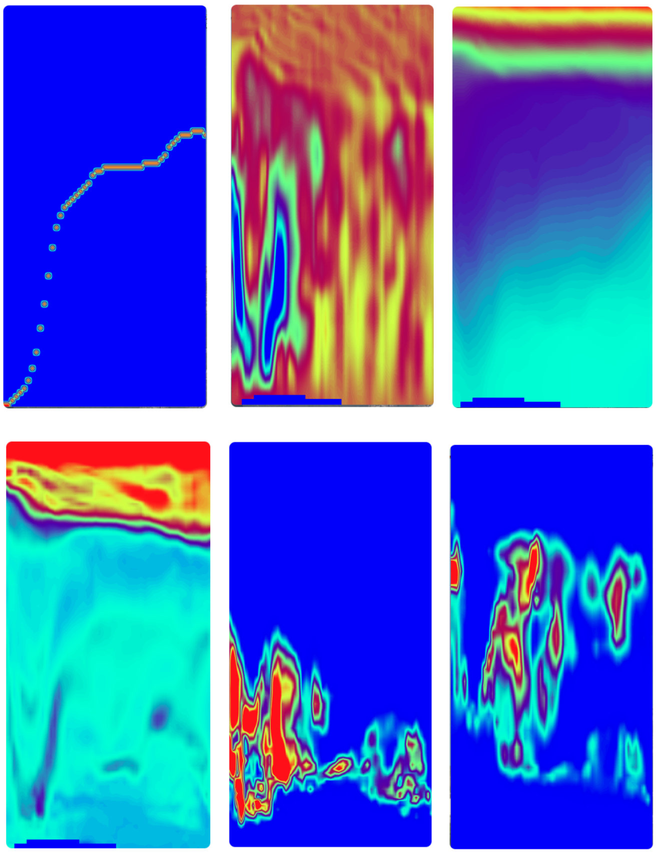

Figure 11: Time-height cross-sections along the trajectory. The different panels correspond (from uper left to lower right): path of trajectory, vertical velocity, potential temperature, potential vorticity, luiquid- and ice-water clouds. The x axis is time from 0 (left) to 48 h (right), the y axis pressure from 1050 hPa (bottom) to 100 hPa (top).

The trajectory ascends rapidly in the beginning to upper-tropospheric levels, then keeps its pressure for a while before ascneding one further step (1st panel). This is, of course, consitent with the vertical velocity (Pa/s) in the 2nd panel. From the 3rd panel we see that the vertical ascent is associated with a substantial cross-isentropic flow, which in turns is consistent with the expected release of latent heat in liquid (5th panel) and ice (6th panel) clouds. Finally, we see that the ascent phase of this trajectory is also associated with production of potential vorticity (4th panel).

8 Reference Guide

#EMAIL

A valid e-mail address; a notification will be sent to this address as soon as the Lagranto calculation is done. Further, the URL to the resulting ZIP archive is provided in the e-mail.

#DATE

Trajectories corresponding to a list of starting positions (#POSITION) or to a starting criterion (#STARTF) always refer to one single time instance.

- **{date}** : a single date in the format {year}{month}{day}_{hour} (e.g., 19980706_12 for 12 UTC 6 July 1998) as the starting date and time of the trajectory calculation. The starting hour should correspond to a synoptic time (00, 06, 12, 18 UTC).
- **{first date} to {final date} by {time step}** : trajectories will be started for all dates in the time period between {first date} and {final date}. The time step between the single calculations (in hours) is given by {time steps}.

If trajectories are started at several times with the option #HOUR+POSITION, the date specified with #DATE provides are reference date/time that corresponds to time 0. All starting times listed in #HOUR+POSITION are then given (in hours) relative to this reference date.

#TIME

The time span of the trajectories in hours; positive values correspond to forward trajectories, negative values to backward trajectories. A typical duration of a trajectory calculation is 120 h. Longer durations are typically associated with considerable uncertainties in the trajectory positions.

#STREAM

The trajectories are calculated based on wind fields provided by the European centre for mid-range weather forecasts (ECMWF; www.ecmwf.int).

- **era5**: Global ERA5 reanalysis of ECMWF; covering the time period January 1979 to December 2023 at a one-hour temporal resolution; meteorological fields are interpolated on a 0.5x0.5 degree longitude/latitude grid, and they are available on 137 vertical levels.
- **erai**: Global ERA-interim reanalysis of ECMWF; covering the time period January 1979 to December 2016 at a six-hour temporal resolution; meteorological fields are interpolated on a 1x1 degree longitude/latitude grid, and they are available on 60 vertical levels.
- **ecmwf**: Operational ECMWF analysis and deterministic forecast at six-hour temporal resolution, covering the latest three months (analysis) and the next seven days (forecast). The fields are globally available, interpolated to a 0.5x0.5 degree longitude/latitude grid (consistent with the ERA5 grid in Lagranto.FTP). The lowest 97 model levels are available, i.e. allowing trajectories to be computed in the troposphere and stratosphere.

#TRACEVARS

The following fields are available for tracing along trajectories. Note that the list depends on the data stream (#STREAM).

Name	**Description**	**erai**	**ecmwf**	**era5**
Q	specific humidity (in g/kg)	X	X	X
LWC	liquid cloud water contents (in g/kg)	X	X	X
IWC	ice cloud water contents (in g/kg)	X	X	X
T	temperature in deg C	X	X	X
U	zonal wind component (in m/s)	X	X	X
V	meridional wind component (in m/s)	X	X	X
OMEGA	vertical wind component (in Pa/s)	X	X	X
PS	surface preessure (in hPa)	X	X	X
SLP	sea-level pressure (in hPa)	X	X	X
TH	potential temperature (in K)	X	X	X
RH	relative humidity (in %)	X	X	X
PV	potential vorticity (in PVU)	X	X	X
THE	equivalent-potential temperature (in K)	X	X	X
LSP	large-scale precipitation (in mm/h)			X
CP	convective precipitation (in mm/h)			X
SF	snowfall (in mm/h)			X
SSHF	surface sensible heat flux			X
SLHF	surface latent heat flux			X
MSL	mean sea-level pressure (hPa)			X
U10M	10-meter wind, zonal component (m/s)			X
V10M	10-meter wind, meridional component (m/s)			X
SSTK	sea-surface temperature (K)			X

#LIDARVARS [only available for ERA5]

The following fields are available for time-height cross-sections along trajectories. Note that the list depends on the data stream (#STREAM).

Name	**Description**	**erai**	**ecmwf**	**era5**
Q	specific humidity (in g/kg)	X	X	X
LWC	liquid cloud water contents (in g/kg)	X	X	X
IWC	ice cloud water contents (in g/kg)	X	X	X
T	temperature in deg C	X	X	X
U	zonal wind component (in m/s)	X	X	X
V	meridional wind component (in m/s)	X	X	X
OMEGA	vertical wind component (in Pa/s)	X	X	X
PS	surface preessure (in hPa)	X	X	X
SLP	sea-level pressure (in hPa)	X	X	X
TH	potential temperature (in K)	X	X	X
RH	relative humidity (in %)	X	X	X
PV	potential vorticity (in PVU)	X	X	X
THE	equivalent-potential temperature (in K)	X	X	X

By default, the time-height cross-section are calculated in the vertical range from 1000 hPa to 100 hPa, with 100 levels. By default, the position of the trajectory is also included in the netCDF output. If more than one trajectory is calculated, the average time-height cross-section is determined over all trajectories.

#OUTPUT

Different output options are supported. By default, ASCII trajectories are provided, but no images. Different output formats must be concatenated, e.g., in the form TRA.NETCDF.PNG.

- **TRA**: trajectory files in ASCII format; a typical trajectory file has the following format:

```
Reference date 20171120_0000 / Time range    7200 min

  time      lon     lat     p         T         Q        PV
-----------------------------------------------------------


   0.00   115.51   -8.34   700   10.107     8.367    -0.189
   6.00   115.54   -8.81   701    9.625     8.734    -0.182
  12.00   115.60   -9.33   706   10.909     8.390    -0.203
  18.00   115.66   -9.81   706   11.044     8.072    -0.210
  24.00   115.46  -10.19   699   10.151     8.784    -0.206

   0.00   115.51   -8.34   600    3.411     5.991    -0.177
   6.00   115.46   -9.06   598    2.819     6.213    -0.173
    ...      ...     ...   ...      ...       ...       ...
```

  The reference date in the first line gives the starting date and time of the trajectory (here, 00:00 UTC 20 November 2017). All times in the first column are relative to this time instance (in the format {hour}.{minute}). The other columns correspond to the position of the air parcels (lon, lat and p) and to the meteorological fields traced along the trajectories (as listed in section #TRACEVARS in the job file). The single trajectories corresponding to the specified starting positions (provided in #POSITION or #STARTF) are listed one after another, separated by an empty line in between. If trajectory calculations are performed for several time instances (as specified in #DATE with the 'from', 'to' and 'by' options), each time instance results in a separate trajectory file.

- **PNG** Two PNG images are provided, one showing the trajectory density globally and one zoomed into the region actually covered by trajectory. As an example, see the two tutorial section in this document.

- **PDF** As for PNG, but now in PDF format.

- **NETCDF** trajectory density in a global netCDF file with a 0.1x0.1 latitude/longitude resolution. Before gridding the trajectories resolution is increased such that the distance between consecutive points is 0.1 deg, i.e., that single trajectories appear as continuous lines in the netCDF file. A quick method to get an idea about the trajectory density is to use the netCDF tool 'ncview' (ncview {file name}). Figure 11 shows a screenshot of the ncview window.

- **KML** Conversion of single trajectory files into GoogleEarth's KML format, i.e. these files can be imported into GoogleEarth as a separate layer. Figure 12 shows an example of ascending air streams (from Tutorial II).

- **MATLAB** Include the Matlab scripts ('read_trajectory' and 'plot_trajector') needed to read and plot trajectories (see Tutorial III for a detailed example).

#POSITION and #HOUR+POSITION

List of starting positions with three columns (longitude, latitude, pressure); each line corresponds to a starting position. An example entry looks as follows:

```
#POSITION
115.508 -8.342 700
115.508 -8.342 600
```

The first column is geographical longitude (-180 to 180), the second column the geographical latitude (-90 to 90). The pressure (in hPa) is listed in the third column. If a starting position falls below topography, i.e. the pressure given is higher than the surface pressure at this position, Lagranto will neglect it: Trajectories are computed only for valid starting positions.

If trajectories are started at different times, with the tag #HOUR+POSITION instead of #POSITION, a list of starting times must be provided in addition to the starting locations. If, for instance, the second trajectory from the previous example is to be started six hours later than the first one, the corresponding section in the job file looks as follows:

```
#HOUR+POSITION
0.00 115.508 -8.342 700
6.00 115.508 -8.342 600
```

Here, the time given in the first column is in hours relative to the reference date specified in #DATE. Positive and negative times values are allowed. To keep the trajectory calculation efficient, the starting times should be consecutive in time, and overall do not span a too wide time window (1 month).

#STARTF

Specification of the starting position by means of a pseudocode criterion. The criterion has the following format:

```
<horizontal>@<vertical>@<unit>
```

Different options are available for the horizontal, vertical and unit specifier.

- **horizontal**:
  - **line(lon1,lon2,lat1,lat2,n)**: n points from (lon1,lat1) to (lon2,lat2); the points are linearly interpolated in lat/lon space
  - **box.eqd(lon1,lon2,lat1,lat2,ds)**: lat/lon box bounded with south-western point (lon1,lat1) and north-eastern point (lon2,lat2); the equdistant points within the box have a horizontal distance ds (in [km])
  - **box.grid(lon1,lon2,lat1,lat2)**: lat/lon box with south-western point (lon1,lat1) and north-eastern point (lon2,lat2 grid points; all grid points within this box are taken as staring points
  - **point(lon,lat)**: single lon/lat point
  - **shift(lon,lat,dlon,dlat)**: lon/at points and dlon/dlat shifhted ones, i.e. in total five points: central one and four shifted ones: (lon,lat), (lon+dlon,lat), (lon-dlon,lat), (lon,lat+dlat), (lon,lat-dlat)
  - **circle.eqd(lonc,latc,radius,ds)**: circle with centre at (lonc,latc) and radius "radius" (in km); the equdistant points within the circle have a horizontal distance ds (in [km])

  - **circle.grid(lonc,latc,radius)**: circle with centre at (lonc,latc) and radius "radius" (in km); all rid points within the circle are selected

- **vertical**:
  - **level(lev)**: a single level
  - **list(lev1,lev2,lev3,...)**: a list of levels
  - **profile(lev1,lev2,n)**: n equdistant levels between lev1 and lev2
  - **grid(lev1,lev2)**: all grid points within layer (lev1,lev2) are selected

- **unit**:
  - **hPa**: pressure (in hPa)
  - **hPa,agl**: pressure (in hPa) above ground level
  - **K**: potential temperature (in K)
  - **INDEX**: index of model level (1=surface)

- **Examples**:
  - **point(-10,50)@list(450,500,550)@hPa**: starting points are (longitude, latitude, pressure in hPa): (-10,50,450); (-10,50,500); (-10,50,550)
  - **line(-10,-5,40,50,10)@level(450)@hPa,agl**: 10 points are equidistantly specified between lon/lat point (-10,40) and (-5,50); all trajectories start at 450 hPa above ground level
  - **box.grid(-10,-5,40,50)@list(300,320)@K**: All grid points in the box with the south-eastern lon/lat point (-10,40) and the north-eastern one (-5,50) are taken; in the vertical, two isentropic levels are chosen: 300 K and 320 K.
  - **shift(-10,40,1,1)@profile(1000,200,100)@hPa**: A profile of 100 equidistant levels between 1000 hPa and 200 hPa; in the horizontal the central lon/lat point (-10,40) is taken and four horizontally displaced ones, the diplacement being 1 degree in zonal and meridional direction

#SELECT

Selection criterion to be applied to the trajectories. The criteria can refer to the path taken by the air parcels or the evolution of meteorological fields along the trajectories. The general format of the criteria is:

```
COMMAND : FIELD : ARGUMENTS : [ TIME ]
```

- **COMMAND:**
  - **GT**: greater than; e.g.,GT:PV:2 selects trajectories with first poten- tial vorticity (PV) larger than 2 PVU
  - **LT**: less than: e.g. LT:RH:70 selects trajectories with first relative humidity (RH) below 70%.
  - **IN**: within: e.g. IN:lon:30,40 selects trajectories with first longitude between 30 and 40 deg.
  - **OUT**: outside: e.g. OUT:lat:-30,30 selects trajectories with first latitude outside -30 and 30 deg - neglecting an equatorial/subtropical band.

- **EQ**: equal: e.g. EQ:p:460 selects trajectories with first pressure equal to 460 hPa.

- **FIELD**:
  - **VALUE**: take value of the field: e.g. GT:PV(VALUE):2 selects the trajec- tories with first potential vorticity (PV) value greater than 2 PVU. This selection criterion is equivalent to GT:PV:2, i.e. the VALUE argument is taken as default.
  - **ABS**: take absolute value of the field: e.g. GT:PV(ABS):2 selects the trajectories with absolute value of potential vorticity (PV) value greater than 2 PVU.
  - **MEAN**: take the mean over the selected times: e.g. GT:RH(MEAN):70:ALL selects all trajectories for which the mean relative humidity (RH) over all times (ALL) is greater than 70%.
  - **VAR**: take the variance over the selected times: e.g. GT:lat(VAR):10:ALL selects all trajectories for which the variance of latitude (lat) over all times (ALL) is greater than 10.
  - **MIN**: take the minimum of the selected times: e.g. LT:p(MIN):300:ALL select all trajectories which have a minium pressure (p) less than 300 hPa over all times (ALL).
  - **MAX**: take the maximum of the selected times: e.g. LT:p(MAX):300:ALL select all trajectories which have a maximum pressure (p) less than 300 hPa over all times (ALL).
  - **SUM**: take the sum over the selected times: e.g. GT:LHR(SUM):2:ALL selects all trajectories for which the sum over all latent heating rates (LHR) over all times (ALL) is greater than 2 K.
  - **CHANGE**: take change between two times: e.g. GT:p(CHANGE):600:FIRST,LAST selects all trajectories wich have a pressure difference —p(FIRST)-p(LAST)— greater than 600 hPa between the first and last time or vice versa. Note that the change can be positive or nega- tive, i.e. it is not clear whether it is ascent or descent.
  - **DIFF**: take difference between two times: e.g. GT:p(DIFF):600:FIRST,LAST selects all trajectories wich have a pressure difference p(FIRST)-p(LAST) greater than 600 hPa between the first and last time - corresponding to an ascending air stream. Correspondingly GT:p(DIFF):600:LAST,FIRST finds a descending air stream.

- **TIME**:
  - **FIRST**: first time: e.g. IN:lat:-20,20:FIRST selects all trajectories with first latitude between 20 S to 20 N, i.e. wich start in an equatorial band.
  - **LAST**: last time: e.g. IN:lat:-20,20:LAST selects all trajectories with last latitude between 20 S to 20 N, i.e. which end in an equatorial band.
  - **T1,T2,T3**: an explict list of times: e.g. IN:lat:-20,20:6,12 selects all trajectories which are in the equatorial band at times 6 h and 12 h. The criterion must apply at both times (see below ALL, ANY, NONE)
  - **T1 to T2**: a time range: e.g. IN:lat:-20,20:6 to 18 selects all trajectories which are in the equatorial band from 6 h to 18 h. The criterion must apply at all times between 6 h and 18 h (see below ALL, ANY, NONE).
  - **ALL**: all times: e.g. IN:lat:-20,20:ALL selects all trajectories which stay at all times in the equatorial band. This time mode is the same as ALL(ALL), i.e. all times are selected and the criterion must apply to all times. With IN:lat:-20,20:12-24(ALL) the crite- rion must apply for all times between 12 h and 24 h.
  - **ANY**: any times: e.g. IN:lat:-20,20:ALL(ANY) selects all trajectories which stay at any times in the equatorial band. Note that with the first "ALL" the times are selected, i.e. all times in this case, and with the second "ANY" it is specified that the criterion must only apply to at least one of the selected times.

– **NONE**: at no time: e.g. IN:lat:-20,20:ALL(NONE) selects all trajectories which never stay in the equatorial band. OUT:lat:-20,20:FIRST(NONE) selects the trajectories which are not outside the equatorial time at the first time: they must be inside.

9 Some further notes

9.1 Technical limitations and cancelation policy

- The maximum number of trajectories is limited to 500'000. If this number is exceeded, the current state of the calculation will be saved and provided as output. An error message can be found in the log file.
- If a running job exceeds a maximum wall clock time (60 min) it will be canceled. the current state of the calculation will be saved and provided as output. An error message is added to the log file.
- There is no guarantee that jobs submitted to this Lagranto FTP service are computed. Jobs might fail beacuse of several reasons: server problems; exceedance of maximum compute resource; maintenance on servers...
- If a job fails to complete **no** notification will be provided that this is the case.

9.2 Python interface to Lagranto.FTP

The following Python program can be used to submit jobs, get the status of the queue, retrieve the output, and – if needed – to cancel/delete an already submitted job.

```
#!/usr/bin/python

from ftplib import FTP
import sys
import re
import os

if len(sys.argv) == 3:
    cmd  = sys.argv[1]
    file = sys.argv[2]

if len(sys.argv) == 2:
    cmd  = sys.argv[1]
    file = 'nil'

if len(sys.argv) == 1:
    cmd = 'stat'
    file = 'nil'

ftp = FTP('iacftp.ethz.ch')

ftp.login()

if cmd == 'put':
    ftp.cwd('/pub_write/sprenger/lagranto.ftp.service')
    print( ftp.pwd() )
    ftp.storbinary('STOR ' + file, open(file))
    print( file+" submitted to Lagranto.FTP" )
```

```
if cmd == 'del':
    ftp.cwd('/pub_write/sprenger/lagranto.ftp.service')
    os.system('touch delete.'+file)
    print( ftp.pwd() )
    ftp.storbinary('STOR delete.'+file, open('delete.'+file))
    print( file+" deleted on Lagranto.FTP" )
    os.system('rm -f delete.'+file)

if cmd == 'stat':
    print( '====== FTP OUTPUT ==========================' )
    print
    ftp.cwd('/pub_read/sprenger/lagranto.ftp.service')
    print( ftp.pwd() )
    ftp.retrlines('LIST')

if cmd == 'get':
    ftp.cwd('/pub_read/sprenger/lagranto.ftp.service')
    print( ftp.pwd() )
    for str in ftp.nlst():
       ftpfile = re.findall(file+'..............*zip',str)
       if ftpfile != []:
          print( ftpfile[0] )
          outfile = open(ftpfile[0],'wb')
          ftp.retrbinary('RETR ' + ftpfile[0], outfile.write, 1024)
          outfile.close()
```

The different steps in performing a trajectory calculation with this Python interface are as follows, assuming that a job is specified in a file 'jobfile':

```
> ./control.py put jobfile   [to submit to the Lagranto.FTP queue]
> ./control.py stat          [to get the status of the Lagranto.FTP queue]
> ./control.py jobfile       [to retrieve the resulting trajectories]
```

9.3 Contact

In case of problems, an e-mail can be sent to:

```
Michael Sprenger
E-Mailk: michael.sprenger@env.ethz.ch
```

Plesae note that we will not be able to provide extensive support for this Lagranto FTP service, but we will be happy to improve its performance and to adjust it to the specific needs of the users. Therefore, we will appreciate any feedback and suggestions for improvements.

9.4 FAQ

- Avoid empty lines at the end of #POSITION and #HOUR+POSITION, as Lagranto might interpret this as a starting time/location and could fail
- The job file can be submitted in as a Windows text file, but it will be converted on the server into Linux format. This affects particularly how 'new line' is encoded. To avoid problems in the conversion, the job files are preferentially already submitted in Linux format.

- Time-height cross-sections (#LIDARVARS) do not work with multiple starting times, i.e. if the tag #HOUR+POSITION is used.

10 References

- Hersbach, H., B. Bell, P. Berrisford et al., 2020: The ERA5 global reanalysis. Quarterly Journal of the Royal Meteorological Society, Vol. 146, 1999–2049, doi: 10.1002/qj.3803.
- Dee, D.P., Uppala, S.M., Simmons, A.J., Berrisford, P., Poli, P., Kobayashi, S., Andrae, U., Balmaseda, M.A., Balsamo, G., Bauer, P., Bechtold, P., Beljaars, A.C.M., van de Berg, L., Bidlot, J., Bormann, N., Delsol, C., Dragani, R., Fuentes, M., Geer, A.J., Haimberger, L., Healy, S.B., Hersbach, H., Hólm, E.V., Isaksen, L., Kållberg, P., Köhler, M., Matricardi, M., McNally, A.P., Monge-Sanz, B.M., Morcrette, J.-.-J., Park, B.-.-K., Peubey, C., de Rosnay, P., Tavolato, C., Thépaut, J.-.-N. and Vitart, F. (2011), The ERA-Interim reanalysis: configuration and performance of the data assimilation system. Quarterly Journal of the Royal Meteorological Society, Vol. 137, 553-597, doi:10.1002/qj.828.

11 Selected papers based on Lagranto trajectories

- Wernli, B. H. and H. C. Davies, 1997: A lagrangian-based analysis of extratropical cyclones. I: The method and some applications. Quarterly Journal of the Royal Meteorological Society, Vol. 123, 467–489. doi:10.1002/qj.49712353811.
- Sprenger, M. and H. Wernli, 2015: The LAGRANTO Lagrangian analysis tool – version 2.0. Geoscientific Model Development, Vol. 8, 2569-2586. doi:10.5194/gmd-8-2569-2015.
- Sprenger, M., and H. Wernli, 2003: A northern hemispheric climatology of cross-tropopause exchange for the ERA15 time period (1979–1993). Journal of Geophysical Research, Vol. 108, 8521, doi:10.1029/2002JD002636, D12.
- Skerlak, B., M. Sprenger, and H. Wernli, 2014: A global climatology of stratosphere–troposphere exchange using the ERA-Interim data set from 1979 to 2011. Atmospheric Chemistry and Physics, Vol. 14, 913-937, doi:10.5194/acp-14-913-2014.
- Knippertz, P. and H. Wernli, 2010. A Lagrangian Climatology of Tropical Moisture Exports to the Northern Hemispheric Extratropics Knippertz. Journal of Climate, Vol. 23, 987-1003.
- Madonna, E., H. Wernli, H. Joos, and O. Martius, 2014: Warm Conveyor Belts in the ERA-Interim Dataset (1979–2010). Part I: Climatology and Potential Vorticity Evolution. Journal of Climate, Vol. 27, 3-26.
- Sprenger, M., G. Fragkoulidis, H. Binder, M. Croci-Maspoli, P. Graf, C.M. Grams, P. Knippertz, E. Madonna, S. Schemm, B. Škerlak, and H. Wernli, 2017. Global Climatologies of Eulerian and Lagrangian Flow Features based on ERA-Interim Bulletin of the American Meteorological Society, Vol. 98, 1739-1748.
- Raveh-Rubin, S., 2017. Dry Intrusions: Lagrangian Climatology and Dynamical Impact on the Planetary Boundary Layer. Journal of Climate, Vol. 30, 6661-6682.