Exercise_2_solutions

September 27, 2024

1 Atmospheric processes: from cloud to global scales

2 Exercise on radar, microphysics and mountain meteorology

2.1 Introduction

In this exercise, you will analyze data from the PLATO campaign that took place at the Davis research station in Antarctica, in January 2019. The case study corresponds to a precipitation event on January 8, 2019, where interesting signatures were observed in the synoptic flow and precipitation microphysics.

1st part
We will look at data from two radars: - BASTA is a vertically-pointing W-band (94 GHz) radar, measuring time series of radar reflectivity and mean Doppler velocity - MXPol is a scanning X-band (9.4 GHz) radar. From its measurements, we extract vertical profiles of radar reflectivity. You will also look at a full Doppler spectrogram, and at some Range-Height Indicator plots (obtained when the radar scans in elevation, at a fixed azimuth angle) to illustrate the scanning capabilities of the instrument, and what information can be inferred from them.

In this first part, we also look at radiosonde data from a sounding launched at Davis station at 12UTC on January 8.

2nd part
To complement the observations, simulations from a high-resolution weather model (Weather and Research Forecast, WRF) are used to study atmospheric dynamics with a 3D perspective. This allows to investigate the interactions between the large-scale synoptic situation, the terrain, and the microphysical processes.

2.1.1 Before starting the exercise

The following instructions apply if you are running the exercise on the virtual desktop that was provided (VDI). Please refer to the corresponding PDF for instructions on how to install and run the VDI client.

In order to execute the code in this exercise, you will need to activate the `lte` environment (deployed on VDI). - If you are running the notebook in Visual Studio Code, select this environment in the Kernel manager (icon at the top right of the code window). - If you are using jupyterlab, make sure you properly activated the environment before launching jupyterlab from the terminal: - Open a terminal, type `micromamba activate lte` - Then launch jupyterlab by typing `jupyter-lab`

To ensure that your work is properly saved when you logoff your VDI session, make sure that this Exercise notebook is in the `~/Desktop/MyFiles/` directory.

2.1.2 General instructions

This exercise does not require to code in Python. You will simply have to execute the cells one after the other by pressing Shift + Enter. In some cells, you will have to adjust the values of certain variables, which will be specified clearly (in CAPITAL letters).

The questions will guide you through an interpretation of the radar variables, sounding profiles and model data in terms of cloud or precipitation microphysical properties and atmospheric processes. More difficult questions are indicated with a star (*).

2.2 Part I

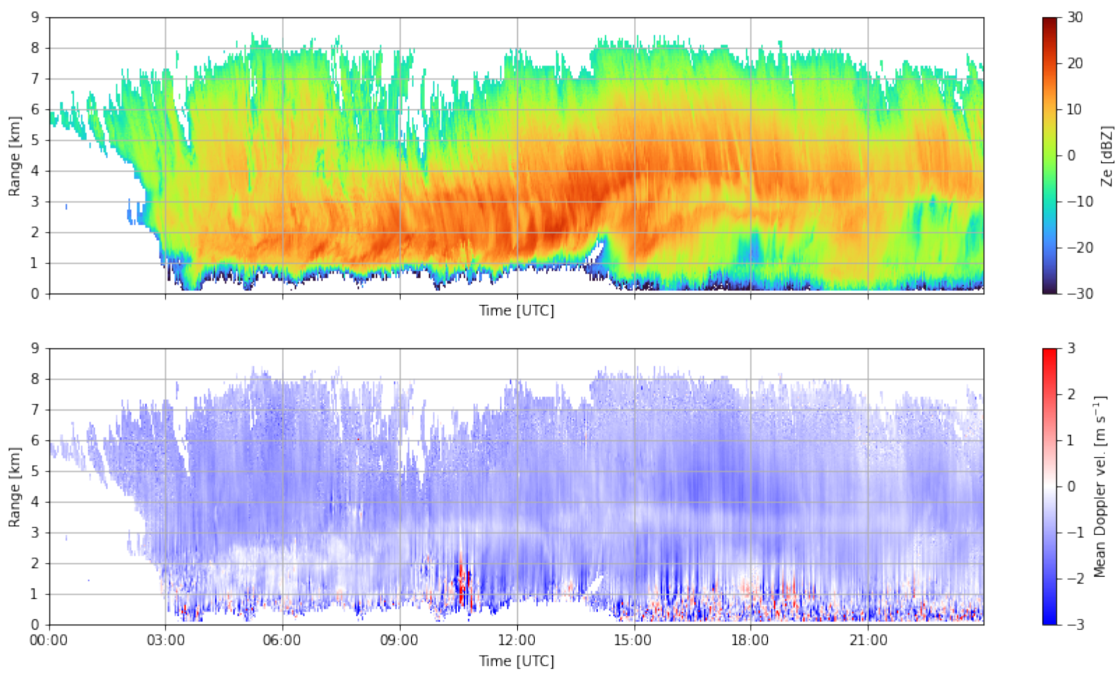
[2]:
```python
from netCDF4 import Dataset
import numpy as np
import datetime
import matplotlib.pyplot as plt
plt.rcParams.update({'font.size':14})
import matplotlib.dates as mdates
import datetime
datefmt = mdates.DateFormatter('%H:%M')
import pandas as pd
plt.rcParams['font.size']=14
import warnings
warnings.filterwarnings('ignore')
```

[3]:
```python
# In this cell, we load the radar data
nc_basta = Dataset('Data/BASTA_L2_merged_20190108.nc')
t_basta = nc_basta.variables['time']
dt = [datetime.datetime(2019,1,8)+datetime.timedelta(seconds=int(tt)) for tt in␣
↪nc_basta.variables['time'][:]]
rg = nc_basta.variables['height'][:]
Ze = nc_basta.variables['reflectivity'][:]
VDop = nc_basta.variables['velocity'][:]
mask = Ze<-50
VDop[mask]=np.nan
Ze[mask]=np.nan

nc_dfr = Dataset('Data/PLATO_dual_frequency_radar_data_20190108.nc')
t_dfr = nc_dfr.variables['time'][:]+3600
dt_dfr = [datetime.datetime.fromtimestamp(tt, tz=datetime.timezone.utc) for tt␣
↪in t_dfr]
```

```
rg_dfr = nc_dfr.variables['range'][:]
DFR = nc_dfr.variables['ZeX'][:]-nc_dfr.variables['ZeW'][:]
```

In the following cell, we plot the timeseries of radar variables measured by BASTA (radar reflectivity Ze and mean Doppler velocity)

[4]:
```
plt.rcParams['font.size']=14
fig,axs = plt.subplots(2,figsize=(15,8),sharex=True)
im0=axs[0].pcolormesh(dt,rg/1e3,Ze,vmin=-30,vmax=30,cmap='turbo')
im1=axs[1].pcolormesh(dt,rg/1e3,VDop,vmin=-3,vmax=3,cmap='bwr')

for ax in axs:
    ax.set_ylim(0,9)
    ax.set_ylabel('Range [km]')
    ax.set_xlabel('Time [UTC]')
    ax.xaxis.set_major_formatter(mdates.DateFormatter('%H:%M'))
    ax.grid()
plt.colorbar(im0,ax=axs[0],label='Ze [dBZ]')
_=plt.colorbar(im1,ax=axs[1],label='Mean Doppler vel. [m s$^{-1}$]')
```



Question 1: Recall the physical meaning of radar reflectivity (Ze) and mean Doppler velocity (MDV). What are the radar targets here? What information do Ze and MDV respectively convey?

Ze (full name: equivalent radar reflectivity factor, shortened as reflectivity, lecture slide 6) is related to the power scattered back to the radar by the targets inside the radar volume (at a certain distance).

It is mainly related to the size of the particles, their number concentration, and their phase (larger Ze for liquid water).
MDV (lecture slide 21) is measured through the difference in frequency of the signal that is scattered back to the radar, compared to the emitted wave; through the Doppler effect, this reflects the (radial) motion of the radar targets toward or away from the radar. In the case of a vertically-pointing radar, the MDV is related to the fall velocity of the particles. Note that the convention is that negative MDV corresponds to downward motion.
The radar targets are hydrometeors (ice / snow particles, raindrops or cloud droplets)

> **Question 2**: When is precipitation observed at the ground? Given the radar data (and the geographical location), what type(s) of precipitation is (are) observed?

Precipitation is taking place at the ground after 15:00 UTC; before that, we can see that there is no radar signal visible close to the ground
The type of precipitation here is snowfall, which we can infer from the following points (see e.g., lecture slides 17, 22, 23): - Ze and MDV values are relatively low (Ze <= 25 dBZ roughly, MDV slower than 2 m/s, vs. around 5 m/s for rain) - There is no sign of melting layer (bright band in Ze, zone of strong increase in MDV) - The geographical location in Antarctica suggests that this is rather a cold cloud system.

> **Question 3**: What type(s) of cloud is (are) present in these timeseries?

High clouds *are visible at the beginning of the timeseries, which are likely a form of cirrus clouds (or cirrostratus, cirrocumulus, …) before ~03:00UTC. Later, a* ***deep cloud*** *system (roughly 8 km deep) is present, associated with precipitation, which can correspond to a nimbostratus* (NB: The types of clouds will be covered in lecture in detail in the part of Prof. Nenes)

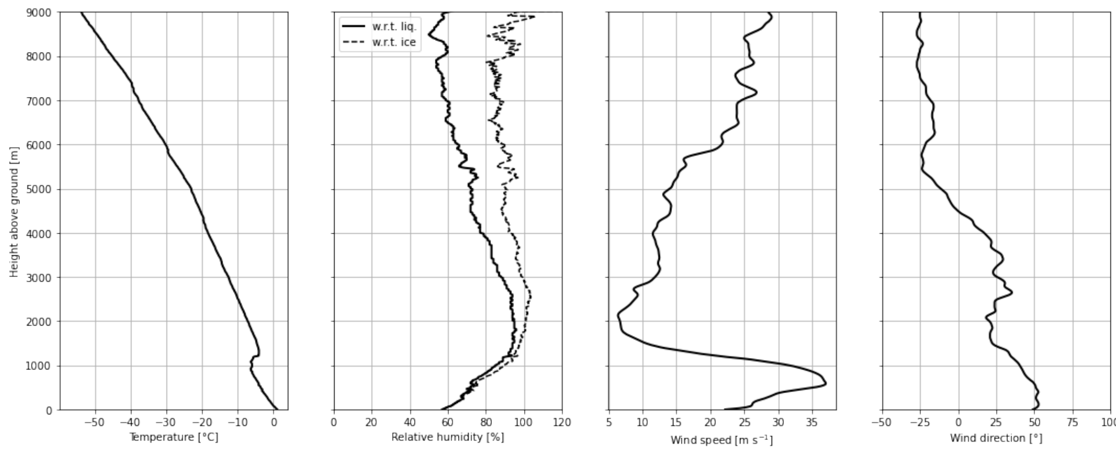
> **Question 4**: What do you observe in the radar timeseries at low levels (below ~1km, especially before 15UTC)? What microphysical process(es) is(are) happening?

- *There is no radar signal returned below ~1km before 15UTC. This means that no radar scatterers (hydrometeors) are present in this altitude range close to the ground*
- *This means that the snow particles which are precipitating from above vanish before they reach the ground. This suggests that sublimation is taking place (i.e., phase change from ice to vapor)*

In the next cell, we plot the standard variables measured by the radiosonde launched at Davis station on January 8 at 12UTC.

[5]:
```
# In this cell, we load the radiosounding data
headers = ['Altitude A.G.L (m)', 'Pressure (Pa)', 'Temperature (K)', 'Specific␣
↪humidity (kg/kg)', 'RH w.r.t liq (%)', 'RH w.r.t ice (%)', 'Wind speed (m/
↪s)', 'Wind direction (deg)']
RS = pd.read_csv('Data/Davis_RS_20190108_12.txt',sep='\t',names = headers )
wind_dir = RS['Wind direction (deg)']
wind_dir[wind_dir>300]=wind_dir-360

fig,axs=plt.subplots(1,4,figsize=(18,7),sharey=True)
axs[0].plot(RS['Temperature (K)']-273.15, RS['Altitude A.G.L (m)'],'k',lw=2)
axs[1].plot(RS['RH w.r.t liq (%)'], RS['Altitude A.G.L (m)'],'k',lw=2, label='w.
↪r.t. liq.')
```

```
axs[1].plot(RS['RH w.r.t ice (%)'], RS['Altitude A.G.L (m)'],'--k',lw=1.5,␣
 ↪label='w.r.t. ice')
axs[2].plot(RS['Wind speed (m/s)'], RS['Altitude A.G.L (m)'],'-k',lw=2)
axs[3].plot(RS['Wind direction (deg)'], RS['Altitude A.G.L (m)'],'-k',lw=2)
axs[0].set_ylim(0,9000)
axs[1].set_xlim(0,120)
axs[3].set_xlim(-50,100)
axs[0].set_ylabel('Height above ground [m]')
axs[0].set_xlabel('Temperature [°C]')
axs[1].set_xlabel('Relative humidity [%]')
axs[1].legend()
axs[2].set_xlabel('Wind speed [m s$^{-1}$]')
axs[3].set_xlabel('Wind direction [°]')
for ax in axs:
    ax.grid()
```

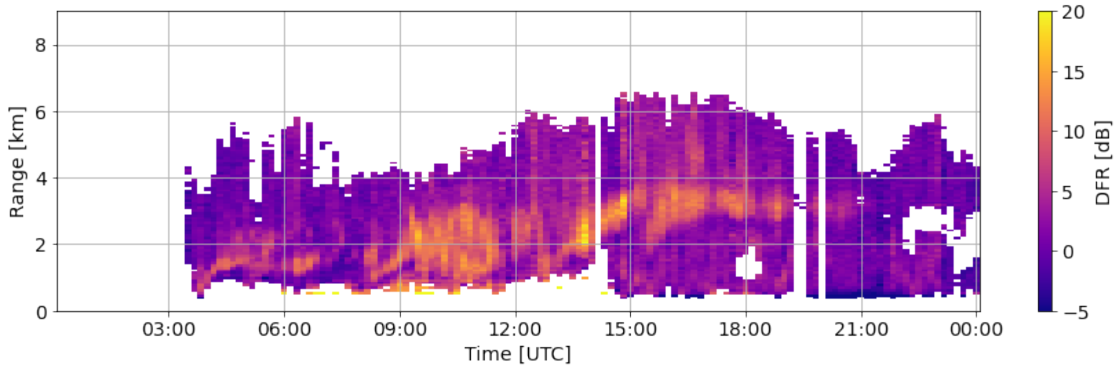


Question 5 How do these sounding data confirm (or not) your answer to question 4?
- *To confirm the answer of question 4 we look at the relative humidity plot. Indeed, it appears that the atmosphere is very dry in the bottom 1 km, with RH << 100%. This supports the hypothesis that sublimation is taking place (very dry air favors sublimation)*
- *As a side comment we can also notice that this dry layer is associated with strong wind speeds (> 30 m/s)*

We now plot the dual-frequency ratio between X and W-band reflectivity, which is defined as $DFR = Ze_X - Ze_W$ with Ze_X and Ze_W in logarithmic units (dBZ).

[6]:
```
plt.rcParams['font.size']=14
fig,ax = plt.subplots(figsize=(15,4))
im0=ax.pcolormesh(dt_dfr,rg_dfr/1e3,DFR.T,vmin=-5,vmax=20,cmap='plasma')
ax.set_ylim(0,9)
ax.set_ylabel('Range [km]')
ax.set_xlabel('Time [UTC]')
```

```
ax.grid()
ax.xaxis.set_major_formatter(mdates.DateFormatter('%H:%M'))
_=plt.colorbar(im0,label='DFR [dB]')
```



Question 6 Recall how DFR can be interpreted in terms of precipitation microphysics. What possible processes could explain the regions of higher DFR in this timeseries?

- *Dual-frequency reflectivity ratio (DFR) reveals information on the* ***size of the particles*** *in the scattering volume. If particles are very small, they behave as Rayleigh scatterers for both radars (i.e., their size is smaller than the radar wavelength for both radars) => DFR~0 dB (or DFR~1 in linear units). If they are bigger, the particle size can be similar to the shorter radar wavelength (W-band: ~3mm wavelength), meaning it is in the Mie scattering regime; but the particles are still in the Rayleigh regime for the larger radar wavelength (X-band: ~3 cm wavelength) => DFR > 0 dB. See lecture slide 25 (session 2)*
- *Higher DFR regions mean that larger snow particles are present. Large snow particles can be formed through aggregation and / or riming*

To investigate this further, in the following cells we plot additional radar variables. - The full radar Doppler spectrogram measured at 09:50UTC. - A Range-Height indicator (RHI) plots for two variables; these plots are obtained when the radar scans in elevation (at a fixed azimuth).

In the next cell, we load the data and print the list of polarimetric radar variables available for the RHI.

[13]:
```
# Load the Doppler spectrogram
nc_spec = Dataset('Data/spec_XPOL_20190108_095053.nc', 'r')
rg_spec = nc_spec['range'][:]
vel_spec = nc_spec['velocity'][:]
spec = nc_spec['spectrum'][:]

# Load the RHI
import pyart
radar = pyart.io.read_cfradial('Data/MXPol-cfradial-20190108-092747-RHI-101_0.
↪nc')
display = pyart.graph.RadarDisplay(radar)
```

```
# We mask out noisy regions, identified through the Rhohv variable
mask = radar.fields['Rhohv']['data']<=0.6
for key in radar.fields.keys():
    radar.fields[key]['data'][mask] = np.nan

# We mask out high-elevation regions for the dual-polarization variables (at␣
↪high elevation their values are not meaningful)
mask_el = (radar.elevation['data'] > 45) & (radar.elevation['data'] < 135)
for key in ['Zdr', 'Kdp', 'Phidp', 'Rhohv']:
    radar.fields[key]['data'][mask_el,:] = np.nan
print('List of available polarimetric radar variables in the RHI: \n')
for k in radar.fields.keys():
    print(k, ': ', radar.fields[k]['long_name'], ' [', radar.
↪fields[k]['units'], ']')
```

```
List of available polarimetric radar variables in the RHI:

Zh :  Reflectivity  [ dBZ ]
Zdr :  Diff. reflectivity  [ dB ]
Kdp :  Specific differential phase (KDP)  [ degrees/km ]
Phidp :  Differential phase (PhiDP)  [ degrees ]
Rhohv :  Copolar corr. coeff  [ - ]
RVel :  Mean doppler velocity  [ m/s ]
Sw :  Spectral Width  [ m/s ]
SNRh :  SNR at hor. pol.  [ - ]
SNRv :  SNR at vert. pol.  [ - ]
Psidp :  Total diff. phase  [ deg ]
Signal_h :  Signal at hor. pol.  [ mW ]
Signal_v :  Signal at vert. pol.  [ mW ]
```

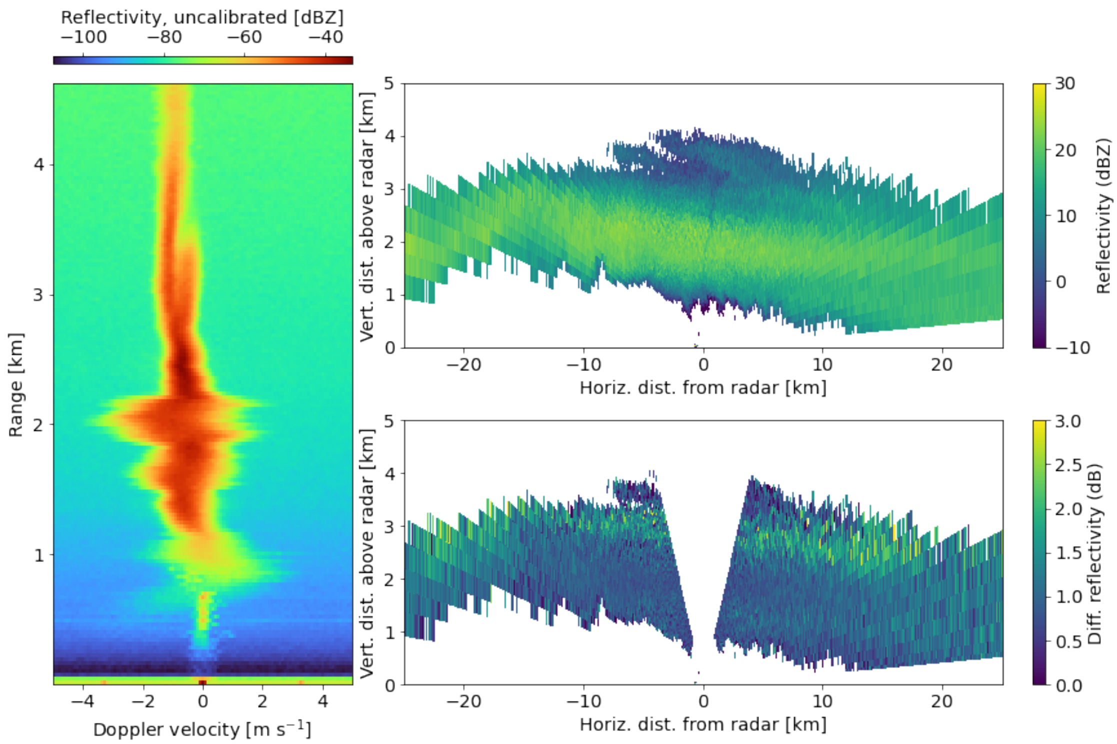
Question 7 What are the most relevant radar variables in the RHI that can confirm the hypothesis from Question 6?

See session 2, lecture slide 19 (+ details on polarimetric radars slides 2-16). We can look for instance at Zh and Zdr.

In the first lines of the next cell, set `VAR_1` and `VAR_2` to the names of these variables (following the abbreviated names in the list above, i.e., 'Zh', 'Zdr', 'Kdp', 'Phidp', etc.). Adjust the min and max values of the colorbar with the corresponding variables (`VMIN_VAR_1`, etc.) to obtain a nice-looking result for the RHI plots in the right panels.

[15]:
```
VAR_1 = 'Zh' # Replace with variable name
VMIN_VAR_1 = -10
VMAX_VAR_1 = 30

VAR_2 = 'Zdr' # Replace with variable name
VMIN_VAR_2 = 0
VMAX_VAR_2 = 3
```

```python
#############################################333333


fig, axs = plt.subplot_mosaic(
    [["spec", "RHI1", "RHI1"],
     ["spec", "RHI2", "RHI2"]], constrained_layout=True,
     figsize=(12,8)
)

spec_plot=axs["spec"].pcolormesh(vel_spec, rg_spec, spec, cmap='turbo')
plt.colorbar(spec_plot, ax=axs["spec"],aspect=40,label='Reflectivity,␣
↪uncalibrated [dBZ]', orientation='horizontal', location='top', pad=.02)

if not (VAR_1 is None):
    display.plot_rhi(VAR_1,ax=axs['RHI1'], vmin=VMIN_VAR_1, vmax=VMAX_VAR_1,␣
↪title='')
if not (VAR_2 is None):
    display.plot_rhi(VAR_2,ax=axs['RHI2'], vmin=VMIN_VAR_2, vmax=VMAX_VAR_2,␣
↪title='')

axs["spec"].set_xlim(-5,5)
axs["spec"].set_xlabel('Doppler velocity [m s$^{-1}$]')
axs["spec"].set_ylabel('Range [km]')
for key in ['RHI1', 'RHI2']:
    axs[key].set_xlim(-25,25)
    axs[key].set_ylim(0,5)
    axs[key].set_xlabel('Horiz. dist. from radar [km]')
    axs[key].set_ylabel('Vert. dist. above radar [km]')
# fig.tight_layout()
```



Question 8 What do you observe in the RHIs around the same altitude range as the high DFR region, and what do you infer?

We observe that Zh increases while Zdr decreases. This is characteristic of aggregation and/or riming: particles become bigger (increasing Zh) and more spherical / less oblate (decreasing Zdr), compared to pristine crystals such as plates or dendrites which are very oblate (high Zdr). See session 2, lecture slide 19.

Question 8bis Describe the signatures observed at different altitudes in the Doppler spectrogram, and propose explanations for the corresponding processes, from top to bottom.

The Doppler spectrogram is the vertical stack of all the Doppler spectra (i.e., at a given altitude, the color gives the Doppler spectrum). The Doppler spectrum shows how the reflectivity is distributed across the velocity bins; for a given velocity bin, it shows the reflectivity from all the particles that fall at this velocity. Looking from the top to the bottom:

- *From ∼ 5 to ∼ 3.5 km : The reflectivity increases slowly, the mean Doppler velocity increases too (spectra more to the left). The particles are growing slowly, probably by vapor deposition.*
- *From ∼ 3.5 to ∼ 2.5 km: There are two distinct modes, meaning two different types of particles are coexisting.*
- *From ∼ 2.2 to ∼ 1.5 km: The width of the spectrogram increases suddenly. The Doppler spectra extend to positive (upward) velocities on the right, and to fast downward velocites on the left. This is a signature of atmospheric turbulence.*
- *From ∼ 1.2 to ∼ 0.5 km: The reflectivity diminishes until we are only in the noise level. This*

corresponds to the sublimation observed in the previous questions.

- *Note: Because they are very narrow, we can hypothesize that the lowest radar echoes around 0.5 km do not correspond to meteorological echoes but rather to so-called "clutter", i.e. fixed echoes that contaminate the radar signal and may come from structures on the ground.*

2.3 Part II

We propose the hypothesis that interaction between the large-scale atmospheric flow and the local orographic terrain is responsible for the observed phenomena: - the low-level process discussed in questions 4-5, - and the higher-level process observed in question 6.

The next steps of the exercise investigate this hypothesis, by relying on the outputs of a high-resolution numerical model (Weather Research Forecast, WRF) which was run on this event.

This corresponds to the file `wrfout_Davis1_v2lessdiff_d03_subselection.nc` inside the Exercise_2/Data/ folder.

Two time steps are available in the data file: 8 January 2019 at 09:00UTC (`TIME_IDX = 0`), and 10 January 2019 at 12:00UTC (`TIME_IDX = 1`). For the main part of the analysis, we use the first time step.

[16]:
```
%reset -f

# Import required libraries
import wrf
import matplotlib.pyplot as plt
from netCDF4 import Dataset
import numpy as np
import datetime
import matplotlib.colors as colors

# Load and prepare data
nc = Dataset('Data/wrfout_Davis1_v2lessdiff_d03_subselection.nc')
TIME_IDX = 0
lat_Davis, lon_Davis = -68.576667, 77.9675
pivot_point = wrf.CoordPair(lat=lat_Davis, lon=lon_Davis)

# Load the atmsopheric variables
it = TIME_IDX
pressure = wrf.getvar(nc, "pressure",timeidx=it)
wa = wrf.getvar(nc,"wa",units="m s-1",timeidx=it)
ua = wrf.getvar(nc,"ua",units="kt",timeidx=it)
va = wrf.getvar(nc,"va",units="kt",timeidx=it)
h = wrf.getvar(nc,"height_agl",timeidx=it)+wrf.getvar(nc,"ter",timeidx=it)
ter = wrf.getvar(nc,"ter",timeidx=it)
th = wrf.getvar(nc,"th",units="K",timeidx=it)
rh_lw = wrf.getvar(nc,"rh",timeidx=it)
temp = wrf.getvar(nc,"temp",units="K",timeidx=it)
```

```
pcum = wrf.getvar(nc, "RAINNC",timeidx=-1)
dt_wrf = [datetime.datetime(2019,1,7,0,tzinfo=datetime.timezone.utc)+datetime.
 ↪timedelta(minutes=int(m)) for m in nc['XTIME'][:]]

# A few calculations to derive relative humidity w.r.t ice
A_w = 2.53 * (10**8) #kPa
B_w = 5.42 * (10**3) # K
es_w = A_w * np.exp(-((B_w)/temp))
A_i = 3.41 * (10**9) #kPa
B_i = 6.13 * (10**3) #K
es_i=A_i* np.exp(-((B_i)/temp))
rh_i=rh_lw*(es_w/es_i)
```

In the next cell, you will plot a map of certain meteorological fields in the area of the research station, as modeled with WRF. At the beginning of the cell, you can define whether or not to plot certain fields, and at which altitude: you can choose to plot them at ground level (set `ALTITUDE` to -1) or at a given fixed altitude (i.e., horizontal cross-section of the domain.)

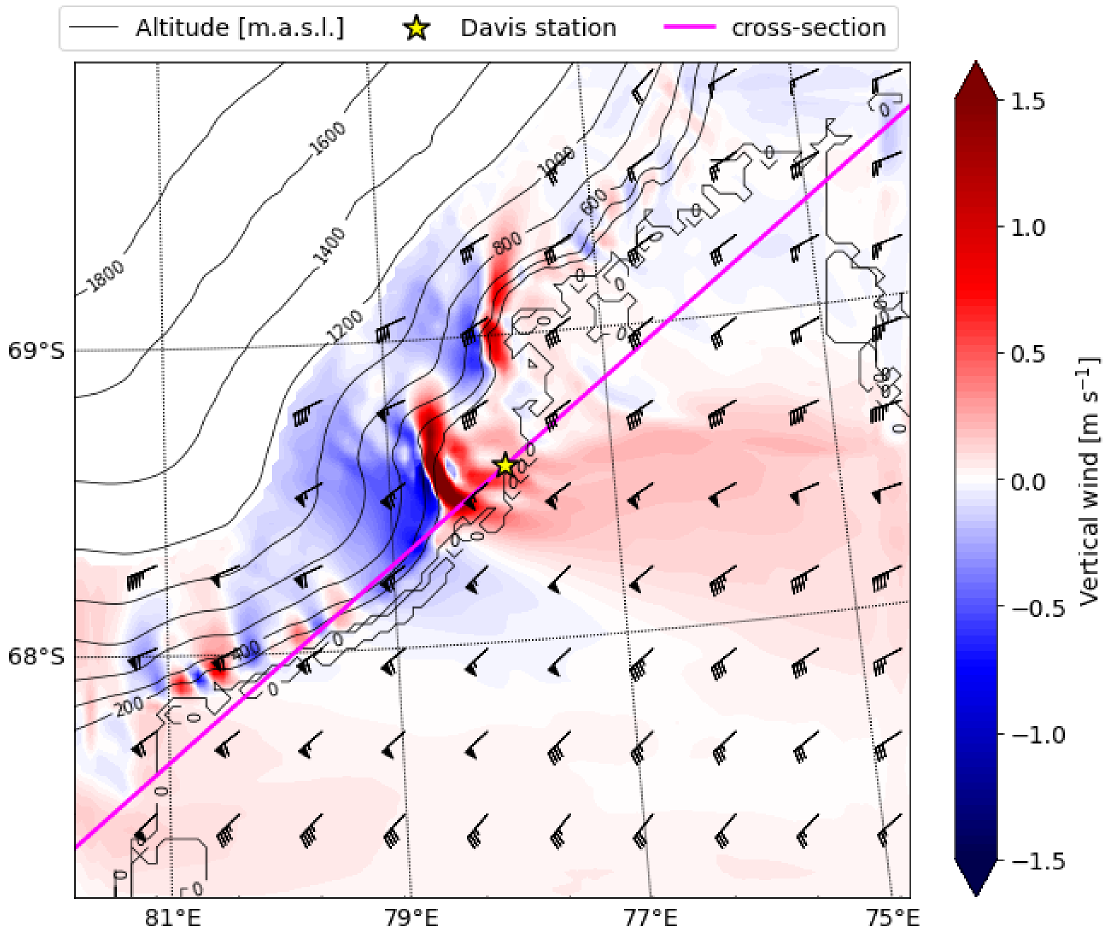
[17]:
```
# 1 - Define whether to display horizontal wind field and how
# Wind barbs are a concise way of representing the horizontal wind field with␣
 ↪speed and direction. They follow a specific convention (detailed here for␣
 ↪instance: https://www.weather.gov/hfo/windbarbinfo)
WIND_BARBS = True # Set to True to plot the wind barbs
WIND_BARBS_ALTITUDE = 1200 # Set to -1 if you want to plot the wind field at␣
 ↪the surface; set to an altitude value > 0 to plot the wind field at this␣
 ↪altitude.

# 2 - Define whether to display vertical wind field and how
W_WIND_CONTOURS = True # Set to True to plot the vertical wind field
W_WIND_ALTITUDE = 1200 # Set to -1 if you want to plot the wind field at the␣
 ↪surface; set to an altitude value > 0 to plot the wind field at this␣
 ↪altitude.

# 3 - In the next cell we propose to plot a **vertical** cross-section of␣
 ↪meteorological fields. Here you can define the direction of the␣
 ↪cross-section (passing through Davis research station) by setting the ANGLE␣
 ↪value.
PLOT_CROSS_SECT_LINE = True
ANGLE = 50

# 4 - Define whether to display the cumulated precipitation (for question 13).
PCUM_CONTOURS = False

################################################################################

# Get the lat/lon coordinates
lats, lons = wrf.latlon_coords(h)
```

```
# Get the basemap object
bm = wrf.get_basemap(ter,resolution='i')
x, y = bm(wrf.to_np(lons), wrf.to_np(lats)) # Convert the lat/lon coordinates␣
↪to x/y coordinates in the projection space

# Create the figure
fig = plt.figure(figsize=(12,9))
ax = plt.axes()

# Add the terrain contours
levels = np.arange(0,2000,200)
contours = bm.contour(x, y, wrf.to_np(ter), levels=levels,␣
↪colors="black",linewidths=.75)
plt.clabel(contours, inline=1, fontsize=10, fmt="%i")
ax.plot([-1,-1],[-1,-1], lw = .75, label='Altitude [m.a.s.l.]',color='black')

# Add the geographic boundaries
bm.drawcoastlines(linewidth=1.5)
bm.drawstates(linewidth=0.25)
bm.drawcountries(linewidth=0.25)
bm.drawparallels(np.arange(-70., -67., 1.), labels=[1,0,0,0])
bm.drawmeridians(np.arange(75.,82.,2.), labels=[0,0,0,1])

# Add Davis station location
xd,yd = bm(lon_Davis,lat_Davis)
bm.
↪scatter(xd,yd,marker='*',color='yellow',edgecolor='k',s=250,zorder=100,linewidth=1.
↪5, label = 'Davis station')
ncols = 2

# Add the wind barbs
if WIND_BARBS:

    if WIND_BARBS_ALTITUDE > 0:
        u_barbs = wrf.interplevel(ua, h, WIND_BARBS_ALTITUDE)
        v_barbs = wrf.interplevel(va, h, WIND_BARBS_ALTITUDE)
    else: # wind field at the surface
        u_barbs = ua[0]
        v_barbs = va[0]

    bm.barbs(x[::10,::10], y[::10,::10], wrf.to_np(u_barbs[::10, ::10]), wrf.
↪to_np(v_barbs[::10, ::10]), length=6, zorder=10)


# Add the wind speed contours
if W_WIND_CONTOURS:
    if W_WIND_ALTITUDE > 0:
```

```
        w_field = wrf.interplevel(wa, h, W_WIND_ALTITUDE)
    else:
        w_field = wa[0]

    w_levels = np.linspace(-1.5,1.5,100,endpoint=True)
    wspd_contours = bm.contourf(x, y, wrf.to_np(w_field), w_levels,␣
↪cmap="seismic",extend="both")
    cbar_ticks = np.linspace(-1.5,1.5,7, endpoint=True)
    cbar = plt.colorbar(wspd_contours, ax=ax, fraction=0.046, pad=0.04, label =␣
↪'Vertical wind [m s$^{-1}$]')
    cbar.set_ticks(cbar_ticks)


if PCUM_CONTOURS:
    pcum_levels = np.arange(0,50,1)#,endpoint=True)
    pcum_contours = bm.contourf(x, y, wrf.to_np(pcum), pcum_levels,␣
↪cmap='Blues',extend="max")
    cbar = plt.colorbar(pcum_contours, ax=ax, fraction=0.046, pad=0.04, label =␣
↪'Cumulated precip. [mm]')


# Add the direction of the vertical cross-section
if PLOT_CROSS_SECT_LINE:
    ter_line = wrf.interpline(ter, wrfin=nc, pivot_point=pivot_point,␣
↪angle=ANGLE,latlon = True, meta=True)
    coord_pairs = wrf.to_np(ter_line.coords["xy_loc"])
    latline = [cc.lat for cc in coord_pairs]
    lonline = [cc.lon for cc in coord_pairs]
    xline,yline = bm(lonline,latline)
    bm.plot(xline,yline,'-',color='fuchsia',lw=2.5, label='cross-section')
    ncols +=2

ax.legend(bbox_to_anchor=(1,1), loc='lower right',ncol=ncols)
```

```
/home/billault/code/anaconda3/envs/acbr/lib/python3.9/site-
packages/setuptools/_distutils/version.py:351: DeprecationWarning: distutils
Version classes are deprecated. Use packaging.version instead.
  other = LooseVersion(other)
```

[17]:
```
<matplotlib.legend.Legend at 0x7fb1e99c0340>
```



Question 9 Look at the map with no meteorological fields plotted (only terrain contours). Briefly describe the topography.

On the bottom right of the plot (i.e., northwest : be careful, the South is toward the top), we see the Southern Ocean. The research station Davis is surrounded by complex terrain with ice ridges.

Question 10 Plot the horizontal wind field (wind barbs) at a chosen altitude (surface level or horizontal cross-section - you can try several). Discuss its orientation with respect to the topography, in the vicinity of the station. Give examples of what could result from this interaction between synoptic/mesoscale flow and orogaphy.

We choose an altitude where the wind direction is rather homogeneous (too close to the ground means it is dominated by very small scale features), e.g., between 1000 and 2000m. Too high would not be relevant to investigate the interaction with the topography.
In the vicinity of the station, the wind direction is such that the flow is encountering the ice ridge (especially if we look at the downslope, to the east (left) of the station: there it is almost orthogonal). In such configurations, depending on the wind and the terrain, we can observe blocking, orographic gravity waves, downslope winds such as foehn (see session 3, lecture slides 18-28), or orographic enhancement of precipitation (session 3, slide 16).

Question 11 Plot the vertical wind field at a chosen altitude (surface level or horizontal cross-section - you can try several). Discuss, and propose explanations (taking into account the previous question).

It seems that we have a strong downslope wind (downward velocity in the downwind slope), followed by a hydraulic jump (strong upward vertical velocity in the lee of the mountain, close to the research station; lecture slide 23). The downslope wind could correspond to foehn.

We will now investigate this further to see how these observations relate to the observed radar signatures and microphysical processes. For this, we look at a vertical cross section through the domain, passing at the research station.
In the **previous** cell, choose a good angle for the cross-section direction and visualize it.

In the next cell, you can activate certain variables to plot in this cross-section (set their value to `True`).

Question 12 Plot the wind field cross-section (wind barbs and vertical wind) and describe the pattern that you see. How does the topography influence the wind field? (relate to the previous questions)

We set the angle to 50 degrees to be approximately aligned with the horizontal wind. Roughly, 45 to 75 is reasonable.
The cross-section plot confirms the previous hypothesis: there is a strong downward wind in the downslope, followed by a hydraulic jump.

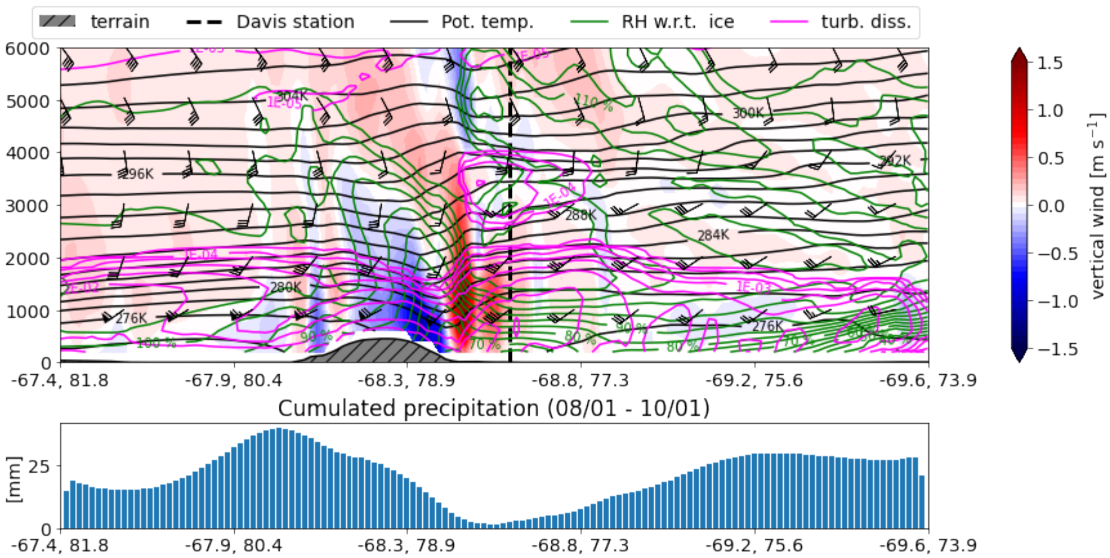
[20]:
```
WIND_BARBS = True
W_WIND = True # Vertical wind
POTENTIAL_TEMPERATURE = True
RH_I = True # Relative humidity with respect to ice
TURB_DISS = True

############################################################

wspd_cross = wrf.vertcross(wa, h, wrfin=nc, pivot_point=pivot_point, angle =␣
 ↪ANGLE, latlon=True, meta=True)
v_cross = wrf.vertcross(va, h, wrfin=nc, pivot_point=pivot_point, angle=ANGLE,␣
 ↪latlon=True, meta=True)
u_cross = wrf.vertcross(ua, h, wrfin=nc, pivot_point=pivot_point, angle=ANGLE,␣
 ↪latlon=True, meta=True)
th_cross = wrf.vertcross(th, h, wrfin=nc, pivot_point=pivot_point, angle=ANGLE,␣
 ↪latlon=True, meta=True)
rhi_cross = wrf.vertcross(rh_i, h, wrfin=nc, pivot_point=pivot_point,␣
 ↪angle=ANGLE, latlon=True, meta=True)
ter_line = wrf.interpline(ter, wrfin=nc, pivot_point=pivot_point,␣
 ↪angle=ANGLE,latlon = True, meta=True)
pcum_line = wrf.interpline(pcum, wrfin=nc, pivot_point=pivot_point,␣
 ↪angle=ANGLE,latlon=True, meta=True)
cldfra = wrf.getvar(nc,'CLDFRA',timeidx=0)
```

```
cldfra_cross = wrf.vertcross(cldfra, h, wrfin=nc, pivot_point=pivot_point,
 angle = ANGLE, latlon=True, meta=True)
qdiss = wrf.getvar(nc, 'QDISS')[:-1]
qdiss_cross = wrf.vertcross(qdiss, h, wrfin=nc, pivot_point=pivot_point, angle
 = ANGLE, latlon=True, meta=True)

xs = np.arange(0, wspd_cross.shape[-1], 1)
ys = wrf.to_np(wspd_cross.coords["vertical"])
coord_pairs = wrf.to_np(ter_line.coords["xy_loc"])

lats_arr = np.array([xy.lat for xy in coord_pairs])
lons_arr = np.array([xy.lon for xy in coord_pairs])
i_Davis = np.argmin((lats_arr-lat_Davis)**2+(lons_arr-lon_Davis)**2)

# Create the figure
fig,(ax,ax1) = plt.subplots(2,figsize=(12,6),gridspec_kw={'height_ratios':
 [3,1]},sharex=False,constrained_layout=True)
ncol=2

# Plot the terrain profile
ht_fill = ax.fill_between(xs, 0, wrf.to_np(ter_line), facecolor="gray",hatch='//
 ', label='terrain')
ax.plot(xs,ter_line,'k')
ax.plot()
ax.vlines(i_Davis,0,6000,'k',linestyle='--',lw=3, label='Davis station')

# Define the x-axis
x_ticks = np.arange(coord_pairs.shape[0])
x_labels = ['%.1f, %.1f'%(pair.lat,pair.lon) for pair in coord_pairs]
num_ticks = 5
thin = int((len(x_ticks) / num_ticks) + .5)
ax.set_xticks(x_ticks[::thin])
ax.set_xticklabels(x_labels[::thin])
ax1.set_xticks(x_ticks[::thin])
ax1.set_xticklabels(x_labels[::thin])
ax.set_xlim(x_ticks[0], x_ticks[-1])
ax1.set_xlim(x_ticks[0], x_ticks[-1])

# Add the wind barbs
if WIND_BARBS:
    x_barb=10
    y_barb=5
    ax.barbs(xs[::x_barb],ys[::y_barb],u_cross[::y_barb,::x_barb],v_cross[::
 y_barb,::x_barb],zorder=10,length=6,linewidth=.75)

# Make the filled contour plot of vertical wind
if W_WIND:
```

```
    wspd_contours = ax.contourf(xs,ys,wrf.to_np(wspd_cross),␣
↪cmap="seismic",levels=np.linspace(-1.5,1.5,50,endpoint=True),extend='both')
    cbar_ticks = np.linspace(-1.5,1.5,7, endpoint=True)
    cbar = plt.colorbar(wspd_contours, ax=ax, fraction=0.046, pad=0.04, label =␣
↪'vertical wind [m s$^{-1}$]')
    cbar.set_ticks(cbar_ticks)


# Contour plot of potential temperature
if POTENTIAL_TEMPERATURE:
    th_contours = ax.contour(xs,ys,wrf.to_np(th_cross), colors='k',levels=np.
↪arange(200,350,2))
    plt.clabel(th_contours, th_contours.levels[::2], inline=1, fontsize=10,␣
↪fmt="%iK")
    ax.plot([-1,-1],[-1,-1], label='Pot. temp.',color='k')
    ncol+=1

def fmt(x):
    s = f"{x:.1f}"
    if s.endswith("0"):
        s = f"{x:.0f}"
    return rf"{s} \%" if plt.rcParams["text.usetex"] else f"{s} %"

if RH_I:
    rhi_contours = ax.contour(xs,ys,wrf.to_np(rhi_cross), colors='g',levels=np.
↪arange(30,120,5))
    plt.clabel(rhi_contours, rhi_contours.levels[::2], inline=1, fontsize=10,␣
↪fmt=fmt)
    ax.plot([-1,-1],[-1,-1], label='RH w.r.t.  ice',color='g')
    ncol+=1


if TURB_DISS:
    qdiss_contours = ax.contour(xs,ys,wrf.to_np(qdiss_cross),␣
↪colors='magenta',levels=np.
↪logspace(-5,2,14,endpoint=True),extend='both',norm=colors.LogNorm())
    plt.clabel(qdiss_contours, qdiss_contours.levels[::2], inline=1,␣
↪fontsize=10, fmt="%.0E")
    ax.plot([-1,-1],[-1,-1], label='turb. diss.',color='magenta')
    ncol+=1

ax.legend(bbox_to_anchor=(1,1), loc='lower right',ncol=ncol)

ax.set_ylim(0,6000)
ax1.bar(xs,pcum_line)
ax1.set_title('Cumulated precipitation (08/01 - 10/01)')
```

```
ax1.set_ylabel('[mm]')
```

[20]: Text(0, 0.5, '[mm]')



We now examine other meteorological variables.
In the previous cell, the *bottom* plot displays the cumulated precipitation (at ground level) along the cross-section line, during the entire event.

> **Question 13** Based on this plot, how does the precipitation pattern seem correlated with the topography? Going back to the cell above, plot the *map* of cumulated precipitation to confirm this.

There is an enhancement of precipitation upslope, probably caused by orographic lifting of the air mass (session 3, lecture slide 16). Downslope, and in the lee of the mountain, the cumulated precipitation is much lower (close to 0 around the research station)

> **Question 14** Include the relative humidity contours in the cross-section plots. Describe and discuss; how is this consistent with the wind field, the precipitation pattern, and with the results from the first part of the exercise (radar + radiosounding, questions 4 and 5)? What phenomenon could this correspond to?

The relative humidity contours show that the air mass is much drier in the lee of the mountain, where we have the strong downslope wind. This is also consistent with the reduced precipitation in this region. This additionally agrees with the radar and radiosounding observations, through which we identified a dry air mass close to the ground leading to precipitation sublimation. This phenomenon with a very dry downslope wind, and a blocking of precipitation upslope, likely corresponds to foehn (session 3, slide 19).

> **Question 14bis** Include the potential temperature contours in the cross-section plots. Describe and discuss; what does it reveal about the cause of the phenomenon?

We recall that the potential temperature of an air parcel is the temperature it would have if it were adiabatically brought to a standard reference pressure. We see that the potential temperature contours follow the descending slope. At a given altitude, the potential temperature is higher in the lee of the mountain than upwind: this means that it is warmer in the lee (downslope) region. This confirms that it is a foehn situation (dry and warm downslope wind). This also suggests that the foehn mechanism may be isentropic drawdown (see lecture slide 19).

Note: feel free to remove the previously drawn contours if the figures becomes too difficult to read (set the corresponding variables back to `False`).

> **Question 15** Include the "TURB_DISS" field (dissipation rate of turbulent kinetic energy), which reveals regions with atmospheric turbulence. Describe this cross-section and how it correlates with the vertical wind cross-section and the terrain profile.

We see two regions with atmospheric turbulence: - The lower levels, corresponding to the planetary boundary layer. We can see that this layer is deflected by the terrain, similar to the potential temperature contours: it is likely affected by the isentropic drawdown. - A higher, localized maximum of turbulence around 3500m above the research station. This seems to be generated by the hydraulic jump and resulting gravity wave (which are often associated with turbulence - see e.g, lecture slide 24).

> **Question 15bis (*)** Looking back at the dual-frequency radar measurements, RHI scans and Doppler spectrogram investigated earlier (Questions 6-8), propose an explanation for how the atmospheric dynamics may contribute to the microphysics within this part of the cloud.

The timing and exact location are not identical (keep in mind that here we are dealing with model data, which might be slightly off), but we can hypothesize that the increased turbulence resulting from the gravity waves of the hydraulic jump favors aggregation (by increasing the collisions between particles), which is the microphysical process that was suggested in questions 6-8. Note that riming is also a possibility (could also be favored by turbulence, with increased collisions between snow particles and liquid water droplets).

To summarize: this is an example where the large-scale flow interacts with the local mountainous terrain. It results in a foehn situation which impacts precipitation (sublimates), and generates a hydraulic jump. It also creates regions with strong atmospheric turbulence downwind, and this also impacts the microphysical processes (aggregation/riming).