

Data Selection Process

Selecting Days with Rain

Rain data from the Davis weather station is examined to find days when it rained at the radar site. Any day with more than 1.0mm of rain is chosen for analysis.

The files are located here on the gws:

`/gws/nopw/j04/ncas_obs/amf/raw_data/ncas-aws-2/incoming/raine/NOAA/`

Processing the vertical radar scans

The function that does the following processing is `process_zdr_scans` in `calib_functions.py`

The top level scripts are `process_raine_vert_scans.py` and `process_raine_vert_scans.sh`

For each day with rain over the radar, the vertical scans from the radar are examined.

1. RAIN-E vertical scans use a single PRF rather than dual, so the velocity range is narrow. This means that sometimes the data is folded in heavy rain. The unambiguous velocity is 8m/s. I've applied a very simple unfolding procedure that seems to do the job:

```
ind = v>6.0
v2[ind] = -7.9725 - (7.9725-v[ind])
```

In rain, any large positive velocity (upwards) is likely to be because the downward velocity has folded. This procedure looks for any velocity greater than +6 and unfolds it. For example, a velocity of +7m/s is unfolded to -9m/s.

2. Next step is to calculate a simple average profile of Z, RhoHV, ZDR and V over the 360 degrees. I don't convert Z to linear units before I average, as it doesn't make a huge amount of difference.

3. I then look for data that meets the following criteria: `rhv_prof > 0.99`, `uzh_prof > 0` and `v_prof < -2`.

The starting point is the lowest range gate that meets these criteria.

4. Next is to look for gaps in this region of good data, as there could be multiple points from different parts of the profile. If there are more than 8 range dates between points then we select the top of the lowest range of data.

5. What is the depth of the data? Is it at least 0.5km? If so use it, otherwise discard the profile.

6. As we only want to be looking at raindrops, we need to make sure we are choosing data from below the melting layer. Next step is to look at the gradient of the velocity and RhoHV profiles to select the bottom of this layer, where there is a large gradient in these variables.

```
grad_v = np.gradient(v_prof)
grad_rhv = np.gradient(rhv_prof)
```

7. By examining multiple profiles, I came up with a threshold for grad_v and grad_rhv. The procedure has to find a large enough gradient of velocity within 0.5km of the region of extracted data (as described above). This prevents regions higher up in the profile being selected. The gradient in velocity alone is sufficient for the profile to be chosen, but if it is not quite large enough to meet the threshold but within a certain range, the gradient in RhoHV is then considered and if this is large enough, the profile can then be used.

```
if np.logical_or(np.logical_and(np.logical_and(max_grad_v_ind > ind3, 0.15 <= max_grad_v < 0.25), min_grad_rhv < -0.015), np.logical_and(max_grad_v_ind > ind3, max_grad_v > 0.25)):
```

grad_v has to be greater than 0.25

if grad_v is between 0.15 and 0.25, then look at RhoHV, if RhoHV gradient is less than -0.015 (larger in the negative) then the profile is usable.

8. Calculate the median value of ZDR from the extracted profile.

9. The data for each day is saved in a file called day_ml_zdr.csv. Melting layer heights are derived, as being the height of the top of the range of the extracted data. This is the height of the **bottom** of the melting layer. This height can then be used in the Z calibration (although potentially it doesn't improve/help it.)

10. A separate function processes the daily files to calculate hourly values. There are about 9 radar scans each hour and there has to be at least 3 profiles in each hour to calculate an average hourly value. This is to exclude any values being derived from bad profiles/outliers that got through the above processing/thresholding.

process_raine_hourly_zdr.py is used which calls the function *calc_hourly_ML* in *calib_functions.py*

Output files and images are located here:

/gws/nopw/j04/ncas_radar_vol2/data/xband/raine/calibrations/ZDRcalib

Each date contains images of the individual vertical profiles and text files of the data, as well as the day and hourly csv files. Hourly files might not exist if there weren't enough scans in the day files.

In */gws/nopw/j04/ncas_radar_vol2/data/xband/raine/calibrations/ZDRcalib/images*

are images of all the data combined which gives us an idea of the ZDR bias.

ZDR bias from horizontal scans

We can also analyse ZDR from the low elevation PPI scans.

I use *calc_horz_zdr_bias_raine.py* which calls the function *horiz_zdr* from *calib_functions.py*

The script looks at low level scans, has a criteria for melting layer height taken from analysis of the vertical scans, and only looks below this level, and also thresholds on the following:

$\rho_{hv} > 0.99$, $\text{phidp} > 0$, $\text{phidp} < 6$, $\text{uzh} > 15$, $\text{uzh} \leq 18$

CSV files are created for each day and saved in the folder:

/gws/nopw/j04/ncas_radar_vol2/data/xband/raine/calibrations/ZDRcalib/horz/

The notebook *plot_horz_zdr_raine.ipynb* is used to plot the results.

In *images* there is a plot showing the full results:

full_horz_zdr_raine.png

SCRIPT and OUTPUT SUMMARY

VERTICAL SCANS

STEP 1

- ❖ RUN *process_raine_vert_scans.sh*
 - CALLS *process_raine_vert_scans.py*
 - CALLS *process_zdr_scans* in *calib_functions.py* in */gws/nopw/j04/ncas_radar_vol1/lindsay/bin/*

OUT = */gws/nopw/j04/ncas_radar_vol2/data/xband/raine/calibrations/ZDRcalib/YYYYMMDD/*

Produces the following files for each good rain profile:

day_ml_zdr.csv

vert_profs_20181031_2016.png

vert_profs_20181031_2016.txt

STEP 2

- ❖ RUN *process_raine_hourly_zdr.sh*
 - CALLS *process_raine_hourly_zdr.py*
 - CALLS *calc_hourly_ML* in *calib_functions.py*

OUT = */gws/nopw/j04/ncas_radar_vol2/data/xband/raine/calibrations/ZDRcalib/YYYYMMDD/*

Produces *hourly_ml_zdr.csv* containing hourly values of ML and ZDR bias.

STEP 3

Use a python notebook to plot the results.

plot_raine_zdr_full_series.ipynb

OUT = /gws/nopw/j04/ncas_radar_vol2/data/xband/raine/calibrations/ZDRcalib/images/
full_raine_zdr_error_250520.png

HORIZONTAL SCANS

STEP 1

- ❖ RUN *calc_horz_zdr_bias_raine.py*
 - CALLS the function *horiz_zdr* from *calib_functions.py*

OUT = /gws/nopw/j04/ncas_radar_vol2/data/xband/raine/calibrations/ZDRcalib/horz/

STEP 2

- ❖ Use notebook *plot_horz_zdr_raine.ipynb* to plot the results
- OUT = full_horz_zdr_raine.png under *images* folder