

notes: supersat

K. Latimer

Jan 13, 2020

1 CAIPEEX

2 HALO

2.1 Data processing

- I did some manual cleanup on my Windows partition in earlier stages of the project for the HALO .ames files downloaded from the online database. The .ames conventions are defined here and there was some degree of human error in following them. The correction process is documented in the ipython notebook file `HALO_file_processing.ipynb` and summarized in the text file “HALO_cleanup_notes,” both in notes/ subdirectory. The resulting raw data files are what I have saved in /data/halo/ames, and are the input for further processing by `halo_data_cleanup.py`. Note that some files from the GoAMAZON campaign have been excluded from that processing for various reasons (weird ames format, missing data, etc).
- 1/12/20: noticed some files had weird error values and changed them manually. Not sure why previous code didn’t find those. Recorded in “HALO_cleanup_notes”.
- 1/14/20: noticed data missing in 3914. see “HALO_cleanup_notes”.

2.2 Analysis

For the record: feedback from R Weigel RE: disparate particle size PDFs between CAS and CDP instruments (bold emphasis mine):

Related to the plots you have distributed (15 Nov 2019), it is not clear and difficult for us to judge whether the extracted particle concentrations were normalized to respective (and variable) bin width specified in the header of the data file.

The plots obviously show different size resolutions for the individual instruments, which disables a direct comparison of these data as the compared size bins are not equivalent.

*One of our major concerns is that averaging over an entire flight causes an averaging over various cloud types and different developing stages of clouds. From take-off through landing, the averaging runs over liquid, mixed and ice phases at different atmospheric conditions and cloud ages. In addition to the non-equivalently binned data sets, the averaging over various cloud types could lead to misinterpretations. For instance, the spikes that you interpreted as CDP's "over-reporting" in sizes of 10, 20, 30 μm (not size-resolved by the CAS) could result from (natural) cloud particle modes within different cloud types and/or at different development stages. **We suggest to limit the instrumental comparison to individual events of cloud encounters which may then be sorted by cloud types or atmospheric condition.***

Furthermore, we suggest to consider differences in the data sets which may result from corrections of the sampled air volume based on the air speed. In terms of true air speed (TAS) or probe air speed (PAS), the data sets as given on the database are not equivalent. You can transfer the CAS data set to the standard of the CDP. The needed calculations conclude from the instructions given in the data headers. However, for any comparison, the data sets should be consistent.

Finally, we suggest once more to include also the NIXE-CAPS data sets and to involve the colleagues in this discussion to better evaluate the deviation of the instruments' data from each other also in the view of the instruments' uncertainty.

From our point of view, at current stage, there is no unambiguous indication, which instrument is "over-" or "under-reporting" something.

- 1/15/20: NIXE-CAPS data contains no time steps (for any flight) where all bins up to 52 μm diameter have non-error values (see `search_NIXECAPS.py`). ADLR LWC values differ significantly from CAS and CDP...possibly includes rain droplets? See 'comparelwc' figure. Also looked at effect of excluding size bins w/ mean diam < 3 μm , and using 'xi' correction factor on CAS; see '1001XX' figure.
- 1/17/20: reran `comparelwc_set_figsrc.py` -> 'v2' figure files. Just some aesthetic adjustments from v1: change colors, plot CAS second since it's usually smaller, sort time for 20140906 CAS data set.
- 1/20/20: refigured raw data files to include lwc calculated values because it was taking too long to run those each time I made figures. General observations for comparelwc plots:
 - In general (see two notes down), ADLR lwc values from ames files are much larger than those calculated from CAS/CDP files...unclear what ADLR is including as liquid water but probably counts rain as well as cloud drops?
 - Roughly three classifications: 1) missing data (either some or all of CAS and/or CDP) [0906, 0921, 0923, 0925, 1003, 1004]; 2) CAS and CDP data both present but peaks have significant mismatch [0919];

- 3) CAS and CDP data both (reasonably) complete and peaks mostly match [0909, 0911, 0912, 0916, 0918, 0927, 0928, 0930, 1001].
- Out of the second two categories, the following have CAS/CDP peaks higher than ADLR lwc curve: 0912, 0916, 0918, 0928, 0930, 1001 - seems weird if ADLR is a superset of those as mentioned above
- Found some odd peak features with very small (generally under $1.e-5$) lwc values when looking at 1001 dataset - code and figure ('v4comparelwc...') in scratch folder.
- 1/21/20: a few things...
 - Following up on last point from yesterday: a couple of variations on comparelwc figure set: 'v3': log scaling on y-axis...a bit hard to look at, hence 'v4': zoom in to lower ($<1.e-4$) lwc regime. Noticed features that aren't visible at default scale, esp 'several' (rough estimate only from skimming through) peaks still hitting $1.e-5$ cutoff in: 0911, 0916, 0919, 0923, 0925, 0928, 1001.
 - I can't figure out a nice programatic way to separate cloud events without making a bunch of arbitrary decisions about selection parameters (peaks are not always cleanly delimited) so just doing manually for a couple of dates at this point. Chose 0909 and 1001 because CAS and CDP peaks roughly match so I can sort them out by eye; 1001 has the feature mentioned above that CAS/CDP lwcs can be higher than ADLR whereas 0909 does not, additionally 1001 has several of those small features discussed in preceeding point whereas 0909 has less - however, the total particle size PDFs we sent to Germany PI's are qualitatively similar so thought this was a good pair to examine side-by-side.
- 1/27/20: comparelwc set v5: include vertical wind velocity data on dual y-axis to figure out extent of error periods (hard to tell from looking at files)
- 1/28/20: Various notes on cloudevents figure set:
 - Also added 0911 flight since a lot of vertical wind velocity data are missing from 1001 (no particular reason other than 0911 has roughly matching peaks between CAS and CDP and also roughly agrees with ADLR).
 - Realized I should be plotting n_{conc}/dr in upper left panel (just as I'm plotting $n_{conc} \cdot r/dr$ in lower left panel)
 - Can replace $n_{conc}/meanr$ vs t panel (lower right on top) with w/A vs t - generally A doesn't vary more than a few percent within a given cloud (order 10^{-2}). Vertical wind velocity is missing for significant portion of all datasets...possibly concerning? Check with David (also see v5 comparelwc figure set).

- I went through all the cloudevent figures manually looking at nconc PDFs and particle size PDFs. I was looking for whether the curves for CAS and/or CDP were unimodal (i.e. a single global maximum, but allowed to be at the boundary of the domain). I excluded the lowest three bins for CAS and the lowest two bins for CDP (i.e., lower bin cutoff must be greater than 3um diameter (1.5um radius), per advice from Germany PIs). Results below in Table 1. CAS definitely better by this metric; main problem with CDP seems to be 6th bin (11.8-15.6um radius range), which tends to report low relative to other bins. For cloud events where both instruments give unimodal curves, CAS pretty much always reports lower nconc and meanr (as we expect from LWC values).
- 2/9/20: cloudevents set v2: corrected diameter to radius, corrected indexing bug for CAS in `get_nconc_vs_t()`, calculated nconc and meanr with changed CAS correction factor and 3um bin cutoff (and graphed accordingly). Updated results similar; see Table 2
- 2/12/20: Several updates:
 - Spent a good amount of time tweaking CAS vs CDP nconc and meanr scatter plots to make sure my results agreed with those previously found by Qindan in her Matlab code (`compare_comparisons.py`). Note that her bin cutoffs for CAS were slightly different than the ones I had been using, which initially caused a discrepancy. But other than that it seems we match up fine.
 - Realized I had forgotten to do a systematic study of the time offsets. Wrote some code to find the optimal offset for each flight date (`optimize_offsets.py`), and incorporated those into the analysis.
 - Figure sets `meanr_scatter` and `nconc_scatter` show the effects of changing CAS correction factor, cutting off bins with particle diameters below 3um, and incorporating optimal time offsets described above (also all three of those things combined). Main takeaway: turns out we can actually get pretty good agreement in both quantities between the two instruments when all three correction schemes are put into place. Yay.
 - There are a lot of figure versions right now. Descriptions:
 - * `cloudevents_set`: v3 - I can't find the documentation for this version in my notes but the figures look identical...; v4 - changed 20141001 offset to 2 per results from `optimize_offsets`.
 - * The only one that really matters is v5 (output from most recent version of the code to date)
- 2/13/20:

- cloudevents set v5: no actual code changes, just to bring everything up to the same most-recent version number to date (TODO: create consistent git-integrated figure version system...)
- notes from meeting with David regarding scatter plots from yesterday:
 - * How to find the correct time offset for vertical wind velocity in supersaturation calculations???
 - * Make zoomed-in plots of LWC with vertical wind velocity to see where we’re missing data (might need to do manual checks)
 - * Change nconc/meanr filters to an LWC filter (TODO)
- removed old-version figure files from GitHub repo to meet file number limit
- 2/17/20: Following David’s comments from 2/13, changed filtering criteria for CAS/CDP scatter plots to look for LWC $\geq 1.e-5$ g/g. I did realize there is a complication here, which is that LWC is calculated using air density, derived from environmental data in ADLR files...so we still have the issue of not knowing the “correct” time alignment for the three instruments. Went ahead and made the plots anyway:
 - meanr/nconc_scatter set v6: filter on LWC, offset CAS time backwards
 - meanr/nconc_scatter set v7: filter on LWC, offset CDP time forwards

Those two versions are a bit sloppy in approach because they sync one out of CAS/CDP to ADLR for filtering, but the LWC values are calculated (in `halo_data_polish.py`) assuming all three synced together to begin with. My code for calculating LWC is slow so I need to either optimize it and rerun or just rerun when I have extra time (TODO). In any case, v6 and v7 described above give the same regression params to two decimal places, and look identical, so I don’t think it’s a priority (furthermore, it looks like ρ_{air} doesn’t change on timescales faster than a few minutes, whereas particle counts vary on the scale of seconds).

Slopes/correlations still look OK, but not quite as good. There’s a strange blob of points in upper left quadrant of mean radius plots for a few flight dates; going to try to figure that out since it makes those regression params worse.

Also made meanr/nconc vs LWC plots out of curiosity but didn’t see much. Also realized that the `optimize_offsets` code was only filtering on meanr, so reran it with meanr *and* nconc, as well as with LWC. Seems like the optimal offsets are pretty robust regardless.

- 2/18/20: trying to figure out blob.

- meanr/nconc_scatter set v8: LWC cutoff 3.e-5: gets rid of ‘blob’ points (quantified using `figure_out_blob.py`: for all dates with visible blob combined, increasing cutoff from 1.e-5 to 3.e-5 removes 91% of points in blob and only 16% of points not in blob, where blob is defined as $r_{mean,CAS}$ between 6 and 10um, $r_{mean,CDP}$ between 12 and 17um).
- meanr/nconc_scatter set v9: only nconc filter (no meanr): looks identical to v5 so only nconc matters.
- 2/19/20: trying to understand relationship between nconc and LWC filtering:
 - meanr/nconc_scatter set v10: nconc cutoff 0.5 ptcls/ccm - analogous to v6/v7 wrt % pts in blob
 - meanr/nconc_scatter set v11: nconc cutoff 1.5 ptcls/ccm - analogous to v8 wrt % pts in blob

I am conceptually thinking of these two filters as roughly the same, therefore.

Moving on to supersaturation (`ss_scatter_set_figsrc.py`):

- v1: LWC cutoff 1.e-5, cas time offset back (cdp synced to adlr)
- v2: LWC cutoff 3.e-5, cas time offset back (cdp synced to adlr)
- v3: LWC cutoff 3.e-5, cdp time offset back (cas synced to adlr)
- v4: LWC cutoff 1.e-5, cdp time offset back (cas synced to adlr)

v2 and v3 show “original data” that looks more like what Qindan found before with the 10 ptcl/ccm cutoff (in agreement with above results), and are also more sensitive to the change in time offset schemes. David is confused/concerned about why v1 and v4 are *not* sensitive to that...TODO: look at time scale of variations in vertical wind velocity / try making the plots with an extreme shift in adlr time offset and see what happens.

- 2/21/20: Follow up from above: looked at w data and time scale for variations is \approx a few seconds so seems like ADLR offset should matter. Made the following versions of (`ss_scatter_set_figsrc.py`):
 - v5: ADLR at -10 second offset relative to CAS and CDP (CAS and CDP synced)
 - v6: ADLR at -30 second offset relative to CAS and CDP (CAS and CDP synced)

We basically see no effect in the slope/correlation so David’s concern is confirmed :/

- 2/27/20: notes from mtg with David: try doing correlation optimization wrt ADLR offset for LWC vs w, alternatively figure out a way to match downdrafts w/ cloud edges? unclear if that's really feasible though. (later thought 3/21/20: I feel like the correlations are not going to be that helpful since w deviates above and below 0 whereas LWC only deviates above...)

Date	# cloud events	Unimodal set(s)	Count - nconc PDF	Count - meanr PDF
0909	47	Both	19	21
		Only CAS	28	26
		Only CDP	0	0
		Neither	0	0
		Excluded	0	0
0911	76	Both	33	33
		Only CAS	31	28
		Only CDP	0	0
		Neither	5	8
		Excluded	7	7
1001	32	Both	7	7
		Only CAS	17	13
		Only CDP	0	0
		Neither	0	4
		Excluded	8	8

Table 1: Counts of unimodal curves based on cloudevents figure set. ‘Excluded’ cloud events are due to either large mismatch between CAS and CDP for that cloud, or isolated spikes in one bin measurement for that cloud. TODO: update to results from v2 figures.

3 WRF

3.1 Lawrencium cluster

username: kalatimer
 enter in terminal: `ssh kalatimer@lrc-login.lbl.gov`
 type: pin + 6-digit auth code from authy chrome app (kt’s token 2) [no space between]
 wrf folder: `/clusterfs/stratus/dromps/wrf_fan2018`
 See project ‘20demo’ notes file for more detailed description of general setup.

3.2 Data processing

Using netCDF4 python library; pretty straightforward.

Date	# cloud events	Unimodal set(s)	Count - nconc PDF	Count - meanr PDF
0909	47			
		Both	21	23
		Only CAS	26	24
		Only CDP	0	0
		Neither	0	0
		Excluded	0	0
0911	76			
		Both	33	33
		Only CAS	33	27
		Only CDP	0	0
		Neither	6	12
		Excluded	4	4
1001	32			
		Both	8	8
		Only CAS	17	14
		Only CDP	0	0
		Neither	7	10
		Excluded	0	0

Table 2: Same as Table 1, but from v2 cloudevents set.

3.3 Analysis

Background: Qindan has shown that in clouds (which we are defining as $LWC \geq 1e-5$), WRF reports a significant number of supersaturated downdraft points. We are trying to figure out why that is (obviously will yield poor agreement with the quasi-steady-state SS formula from Korolev 2003, since that puts $SS \propto w$). David suggested looking at the relationship between condensation rates and supersaturation (acutally using latent heating rates due to droplet diffusion, which should be a good proxy since latent heat of vaporization has weak pressure / temperature dependence)

- 2/24/20: made initial plot of LH vs SS (`lh_vs_ss.py`). Kind of hard to tell due to high density of data points, but looks like signs are about right (i.e. no points in Q2 or Q4). Wrote code to remake the plot and also report exact counts of points in each quadrant(`lh_vs_ss_all.py`). Also wrote code to make plots of just updraft and downdraft regions (`lh_vs_ss_wpos/neg.py`). Not currently filtering by LWC.
- 2/26/20: summary for current versions of `lh_vs_ss` figure set (also see presentation slides):
 - v1 (all, wpos, wneg): no filters - about 50/50 updrafts/downdrafts, almost all points are subsaturated (especially so in downdrafts, as expected)

- v2 (wneg): no substantive change, just switched colorbar gradient direction
- v3 (wpos, wneg): LWC filter ($\leq 1.e-5$) - less than 10% of updraft points remain; out of those that do remain, *most* are supersaturated. less than 5% of downdraft pts remain; out of those that do remain *most* are subsaturated.
- general observations: we definitely see higher SS values in unpolluted case, but relative quantity hard to tell due to high density of points in scatter plot; cont's curves in downdrafts plot?
- 2/27/20: notes from mtg with David: try v3 of `lh_vs_ss` but with w gt/lr some finite nonzero (pos/neg) values so that we don't get air that's just drifting up and down (eg gravity oscillations or smth); next try calculating SS using formula from ch 7 of Rogers and Yau.
- 3/4/20: v4 of `lh_vs_ss`: for wpos, w ≤ 1 ; for wneg, w ≤ -1 . qualitatively similar to v3 but 3-fold reduction in number of updraft points, 10-fold reduction in number of downdraft points. The *most*'s above are even more *most*. in particular for downdrafts, out of the remaining points only $\approx 0.1\%$ are supersaturated...so I'm a little confused about Qindan's previous finding that WRF gives 'a lot' of supersaturation in downdrafts. plan moving forwards is to reproduce her comparison plot.
- 3/12/20: a lot to update...
 - Made the weird WRF outputs nicer and consolidated into single file (i.e. for all hour-long increments) using `make_secondary_wrf_vars.py`. Output is in the same data directories as primary data files.
 - Double-checked that I was doing the mean radius / number concentration calculations correctly in `check_nconc_figsrc.py`...have it right now but *TODO*: need to run with correct radius values (corrected the code on 3/12 in the am but need to run on slurm).
 - Comparing quasi-steady-state ss approx with WRF reported values (which Qindan found were identical to Fan's except for a few artifacts in the dataset that seem to be artificial). First round: entire horizontal domain (`qss_vs_fan_figsrc.py`):
 - * v1: wrong LWC calculated values; disregard
 - * v2: correct LWC calculated values; wrong radii (off by factor of 2)
 - * v3: no figure - text only to diagnose num pts in various regions of the plane
 - * v4: LWC above $1.e-5$, correct incorrect factors of 2 manually (for now); make axes square
 - * v5: LWC above $1.e-5$, only points with abs val of vert wind vel greater than 1 m/s

- * v6: LWC above 1.e-5, only points with nconc greater than 3e6 (this is what Qindan was graphing)
- Second round: only the subregion around 'T3' site in figure S9 of Fan2018 (supp info file) (`subregion_qss_vs_fan_figsrc.py`):
 - * v1: LWC above 1.e-5
 - * v2: LWC above 1.e-5, only points with nconc greater than 3e6
- 3/21/20: there's been a lot going on and I haven't been recording my notes very frequently but also have not been as productive on average so guess it evens out. Anyways: finally fixed the bin radius issue (see v4 figure from `check_nconc_figsrc.py`) and remade the secondary variables netCDF files. Resulting supersaturation plots:
 - `qss_vs_fan` v7: LWC above 1.e-5, make axes square
 - `qss_vs_fan` v8: LWC above 1.e-5, only points with nconc greater than 3e6 (this is what Qindan was graphing)
 - `qss_vs_fan` v9: LWC above 1.e-5, only points with abs val of vert wind vel greater than 1 m/s
 - `qss_vs_fan` v10: LWC above 1.e-5, only points with abs val of vert wind vel greater than 1 m/s and less than 10 m/s (based on comment in Appendix A of Korolev2003...didn't make a difference)
 - `subregion_qss_vs_fan` v3: LWC above 1.e-5 [forgot to remove factor of 4]
 - `subregion_qss_vs_fan` v4: LWC above 1.e-5, only points with nconc greater than 3e6 [forgot to remove factor of 4]
 - `subregion_qss_vs_fan` v5: LWC above 1.e-5 [correct now]
 - `subregion_qss_vs_fan` v6: LWC above 1.e-5, only points with nconc greater than 3e6 [correct now]

needless to say, these kind of look like shit and David doesn't necessarily have an idea of what is going on here. Have gone back to Rogers and Yau (RY) to try to figure out what might be the issue. First thing: check quantity RY calls $\frac{Q_2}{F_k + F_d}$, which in my code above has just been a constant D . This code is `qss_expand_denom_figsrc.py`; see v5 figures, where I've plotted the distribution for the reciprocal of aforementioned quantity (since it appears in the denominator in the expression for supersaturation). Seems like there is spread of about $\pm 10\%$, which doesn't seem like it should really be an issue, but gonna try a new version of `qss_vs_fan` with the full expression just to see what happens.

I'm also more and more confused about how the 'SSW' quantity was calculated in WRF and have emailed David to ask how outputs are specified.