

notes: supersat

K. Latimer

Jan 13, 2020

1 CAIPEEX

9/7/20: Been working on this for a while but just now starting to write down my notes. The `caipeex` package was largely exploratory and `revcaipeex` (the revised version) should be taken as the more reliable / up-to-date iteration. Everything is more or less carried over from `revhalo` with a few miscellaneous differences:

- only one instrument was used to measure cloud drop spectra in CAIPEEX campaign which simplifies things slightly. The dsd's are given in kg/m^3 and we carry out conversions accordingly.
- as of 9/7/20 we still don't have rain drop data from CAIPEEX (contacted PI and haven't heard back yet...). So all figures with `incl_rain` or `incl_vent` are nonsense right now.
- `change_cas_corr` is not applicable for this data set but keeping in the `get_boolean_params` method in histogram figure source code files to keep things simple for now.
- won't need to worry about time offset optimization because met and dsd seem to be synced fine already in raw data files.

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 < 3um, 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 $nconc/dr$ in upper left panel (just as I'm plotting $nconc*r/dr$ in lower left panel)
 - Can replace $nconc/meanr$ 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 $3\mu m$ diameter ($1.5\mu m$ 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 ($11.8-15.6\mu m$ 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 $3\mu m$ 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 3 μ m, 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 ≥ 1.5 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...)
- 4/12/20: it’s been a while but here we are. Sent Professors Jurkat and Weigel (et al on that email thread) a fairly extensive summary of results since January 2020 (all referenced figures in powerpoint in `pres` directory):

Thank you for your patience as we have taken your thorough feedback into account in our analyses. In some cases, implementing the previous suggestions led to new questions for us, and we wanted to wait on sending this response until we felt we had a clearer picture of what was going on.

Here we are writing a summary of the points you all have raised throughout the correspondence, along with our subsequent findings. For brevity, we’ve included the figures in a single file, attached.

1. Various corrections to the datasets

In all cases here, we are filtering out points with liquid water content (LWC) less than 3×10^{-5} g/g. Using a smaller LWC cutoff, we begin to see artifacts arising from the particle size distributions reported by the CDP instrument; see discussion in part (2)(A).

A. Commensurate sample volume correction factors

The ames files on the HALO database for the CAS instrument were corrected using the factor TAS/PAS, whereas for the CDP instrument they were corrected using the factor ξ (as derived in Weigel et al (2016)). We

re-corrected the CAS values using x_i (provided as well in the datafiles), and generally don't see much improvement in correlation for number concentration (this correction doesn't change mean radius correlation since the volumetric factors cancel in the weighted average). See Figure 1.

B. Lower bound on particle size

As suggested by Dr. Jurkat, we performed the same comparison using only bins with a lower bound for droplet diameter greater than 3 μm . This improves the value of the regression slope, but not the correlation, for number concentration (Figure 2), and nearly eliminates the constant offset between CAS and CDP mean radius (Figure 3).

C. Time offset between instruments

We also reran the analysis after offsetting the instrument times relative to one another such that the correlation between number concentration and mean radius measurements was optimized (these offsets varied across flight dates; see Figure 4). This correction yielded a significant improvement in agreement for number concentration (Figure 5), although not for mean radius (Figure 6). Unfortunately, it's not clear to us how we should correctly sync CAS and CDP measurements to other instruments (in particular ADLR, which records most of the basic environmental variables).

The combined effect of corrections described in (A), (B), and (C) is shown in Figure 7. We also have analogues of Figures 1-6 for each individual flight date, and are happy to share those.

2. Cloud-by-cloud analysis

We were hoping to understand why the CDP instrument seems to report multi-modal cloud droplet size distributions, and Dr. Weigel suggested we analyze individual cloud events for each flight. We did so (working with the fully-corrected data described in part (1)), and an example plot is shown in Figure 8. We can provide the entire set upon request, but since there are many of them, we will summarize our main observations here:

A. We find CDP consistently reports bi- or multi-modal cloud droplet size distributions, even at the level of individual clouds (we are accounting for the variable bin size ranges when computing these distributions). For instance, on the 09 Sept flight, we found that about half of the CDP size distributions are non-unimodal, whereas all of the CAS size distributions are unimodal. This effect is most apparent in the mean cloud droplet size data at low LWC; if we use an LWC cutoff of 1×10^{-5} (rather than 3×10^{-5} as in part (1)), we see the cluster of points with anomalously high mean radius reported by CDP, as visible in Figure 9.

B. While LWC values reported by CAS and CDP are generally in good agreement (up to the 1 second time offsets described in part (1)), there is one flight date (19 Sept) where they differ significantly even though neither reports error values, see Figure 10.

3. Comparison with other instruments

The only other bin-resolved cloud droplet data available in the online dataset from the GoAmazon field campaign was the NIXE-CAPS set. We did not find those files to be useful, due to high error rates. In particular, there were no timesteps in any of the flight dates where non-error-valued data were present for the whole particle size range of interest (3 μm - 50 μm diameter).

We did find (non-bin-resolved) LWC data in the ADLR set. However, it appears those measurements include rain droplets as well as cloud droplets, since the LWC values are considerably higher than those we calculated using the CAS and CDP sets (referring back to Figures 8 and 10).

Selected points of feedback after this message:

- *There are hotwire LWC measurements available. The hotwire was mounted on the CAS-DPOL but is an independent LWC measurement. Ramon Braga also compared these in his paper with the CAS measurements (see plot below).*
- *We also have information on the phase and can thus apply a binning to either ice phase or liquid phase. The current binning is chosen such that the ice and liquid can both be unambiguously detected. The consequence is a reduced size resolution in the lower size range. I can give you recommendations on how to do the binning. Once this is done, we can talk again about the comparison of the mean radius. As far as I remember there have been a few differences when going to higher MVDs.*
- *What might help is to compare the PAS (measured by the CAS) with the TAS of HALO and synchronize it with respect to level changes (generally correlated to air speed changes).*

Of these, the last is the most relevant since our main concern at the moment is syncing ADLR to CAS (update: have basically decided not to use CDP data due to multimodal particle size distributions from that instrument). However, CAS is actually just pulling TAS measurements directly from ADLR (see `ASvst_figshow.py`), so seems not super valid as a basis for time syncing. In any case, made `w_tas_optimizes_offset[figsrc].py` which shows R^2 for TAS_{ADLR} and TAS_{CAS} (blue) and LWC_{CAS} vs $|w|_{ADLR}$ (red) as a function of $\Delta t_{ADLR/CAS}$. David suggested looking at the standard deviation and 99th percentile of CAS SS values versus $\Delta t_{ADLR/CAS}$, which I did in `v3 of ss_optimize_offsets[figsrc].py`...not looking great.

- 9/5/20: hello again, world. I have revised the entire codebase for all data sets, starting with the `revhalo` package. The `revhalo_data_cleanup.py` file is basically just c/p from `halo` package, but other than that I am transitioning to self-documenting code for greater clarity and easier debugging. Now that I have trimmed out all the exploratory stuff I think it should

be fairly clear what all is going on but just want to make a few notes on details that have been cleaned up from original `halo` implementations:

- making bin cutoffs more uniform...5um for everything for both CAS and CDP (note e.g. that in `calc_lwc` method of `src/halo/utils.py`, lower bin cutoff is 3um.) Tina told us to use 3um but sticking with 5 since CAS has a 3-5 bin and CDP has a 2.9-5 bin so 5 just seems cleaner.
- (relevant when I actually implement time offset optimization) doing all offsets at the very beginning of processing stage and saving the data as it is post-offset, post time-sync. Actually saving data post time-sync is already relevant and saves headaches when using the `numpy_proc` files.
- 9/21/20: incorporated ‘PCASP’ data into those processed in `revhalo_data_polish.py`, which is aerosols with diam 0.1-3um. This should align roughly with PCASP data set from CAIPEEX (described in literature but we don’t have the raw data yet). Notes on this:
 - a couple of dates are missing from `pcasp` rel to `adlr` and `dsd` files, just skipping those.
 - there’s no data in `pcasp` files that I can possibly use to optimize time offset so just taking offset relative to `adlr` as being zero.
 - didn’t want to fuck with `adlr` and `dsd` time values so for `pcasp` data, I make the arrays the same length as `adlr` (post-time-sync with `dsd`), inserting data where times match and putting nan’s where they don’t.
 - aerosol conc’s in file reported at STP so I correct them and include both values in the processed numpy files; not really sure why they do this correction for aerosols but not water droplets; maybe ask David at some point.

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	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.

3.3 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,

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.

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 (1.1×10^{-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.

- 3/22/20: v11 (with full expression for denominator of qss SS approx) of `qss_vs_fan` doesn't look much different so that's that. Also made program `compare_esat_formulae` to compare different formulae for saturation vapor pressure of water based on Khain and Pinsky but agreement is pretty perfect so that's out as an explanation.
- 3/25/20: new code: `backtrack_qss_figsrc.py` - basically data exploration. in all versions, I use LWC greater than 1e-5; in even versions, I additionally enforce abs val of vert wind vel greater than 1 m/s.

- v1/v2: $\frac{\rho_a * (F_k + F_d) * LH}{\rho_L}$ vs $\frac{Aw}{D}$ - LH is 'TEMPDIFFL' output from WRF.
- v3/v4: same but first quantity above multiplied by $1L_v$ (arithmetic error on my part...afterwards I realized it should also be multiplied by C_{ap} , but didn't bother to make the figure for that)
- v5/v6: WRF SSW output vs $\frac{\rho_a * (F_k + F_d) * LH}{4\pi\rho_L L_v * nconc * meanr}$ (this is also missing a factor of C_{ap} which is why it looks weird)
- v7/v8: same but I made axes to be equally scaled so it didn't work
- v9/v10: same but no equal scaling of axes and colored by temp; x-axis has large outliers so also doesn't work
- v11/v12: same but limit of ± 1000 for x-axis

- 3/29/20: more versions of `backtrack_qss`:

- v13: same as v11/12 but with T greater than 273K filter
- v14: blank
- v15/16: WRF SSW output vs $C_{ap} * LH$
- v17/18: same as v11/12 but color by mean radius
- 3/30/20: tbh just brute force trying to figure out what’s going on in these plots. the following versions plot SSW vs “ SS_{RY} ” $\equiv \frac{\rho_a * (F_k + F_d) * C_{ap} * LH}{4\pi\rho_L L_v * nconc * meanr}$ with filters of LWC greater than 1e-5 and meanr between 0 and 60um (not sure why default colorbar extends negative...TODO look into that), odd versions cutoff SS_{RY} at $\pm 1000\%$ and even versions allow default axis scaling which should show the entire domain:
 - v19/20: color by mean radius (v20 updated later to truncate out severe outliers in unpolluted case)
 - v21/22: color by number concentration
 - v23/24: color by $F_k + F_d \equiv F$
 - v25/26: color by LH
 - v27/28: color by ρ_a (v28 updated later to truncate out severe outliers in unpolluted case)
 - v35/36: color by w
 - v37/38: color by z

Based on the appearance of v21/22, decided to tweak filtering a bit for the next few - same quantities being plotted and version parity convention, all colored by number concentration:

- v29/30: LWC greater than 1e-5, meanr between 0 and 60um, nconc greater than 1e7
- v31/32: LWC greater than 3e-5
- v33/34: LWC greater than 3e-5, meanr between 0 and 60um
- 3/31/20: v39/40 of **backtrack_qss**: LWC greater than 5e-5, color by z
- 4/12/20: back after another dry spell without updates...including more **backtrack_qss** versions!!
 - v41/42: LWC greater than 5e-5, meanr between 0 and 60um, color by w
 - v43/44: LWC greater than 5e-5, meanr between 0 and 60um, z greater than 6km, color by z
 - v45: apparently does not exist.
 - v46/47: LWC greater than 1e-4, meanr between 0 and 60um, color by z

- v48/49: LWC greater than $5e-5$, meanr between 0 and 60um, color by meanr
- v50/51: LWC greater than $5e-5$, meanr between 0 and 30um, color by z

Additionally, wrote and ran `ptcount_backtrack_qss.py` to make sure that coloring seen e.g. in v39/40 is just because red points are being plotted on top of blue points (yes, this is the case).

Comparing v50/51 with v39/40 shows that regions with large drops seem to be the problem in terms of divergence of SS_{RY} ...I thought it was because coalescence is taking over condensation as the dominant growth mechanism in that regime, but David pointed out it could be because Qindan and I have only been looking at the first 15 ‘rain/cloud’ microphysics bins from WRF output (to which he also attributes strange values of LWC (cloud only) and diffusional latent heating that I am still working to understand)...rerunning `make_secondary_vars.py` to include all 33 bins in meanr and nconc calculations.

Related note: data organization on Lawrencium a little messy right now because I made some consolidated files for David to look at...TODO: fix that up. Also the docstrings on everything :’(

- 4/17/20: Now that my meanr and nconc data include rain drops, I redid comparisons of SS_WRF to SS_QSS and SS_RY, using code `inclrain_qss_vs_fan_figsrc.py` and `inclrain_backtrack_qss_figsrc.py`, resp. For `inclrain_backtrack_qss`:
 - v1: LWC greater than $1e-5$ [note: still using LWC from clouds only to filter]
 - v2: LWC greater than $1e-5$, no ax lims

For `inclrain_qss_vs_fan`:

- v1: LWC greater than $1e-5$
- v2: LWC greater than $1e-5$, no ax lims
- v3: LWC greater than $1e-5$, abs w greater than 1
- v4: LWC greater than $1e-5$, abs w greater than 5
- v5: LWC greater than $1e-5$, abs w greater than 5, no ax lims
- v6: LWC greater than $1e-5$, abs w greater than 5, T greater than 0
- v7: LWC greater than $1e-5$, abs w greater than 1, T greater than 0
- v8: LWC greater than $1e-5$, abs w greater than 2, T greater than 0
- v9: LWC greater than $1e-5$, abs w greater than 3, T greater than 0
- v10: LWC greater than $1e-5$, abs w greater than 4, T greater than 0

Additionally, for `qss_vs_fan` (i.e. no rain drops included):

- v13: LWC greater than 1e-5, abs w greater than 5, T greater than 0
- polluted
- v14: LWC greater than 1e-5, abs w greater than 5, T greater than 0
- unpolluted (only a different version because of minor code setup bug)

Summary of interpretation: including rain drops improves SS_RY approximation because WRF latent heating output includes diffusional growth of rain droplets (so my earlier hypothesis regarding dominant growth mechanisms wasn't correct). Including rain drops only improves SS_QSS if I limit to points below the freezing line - this is because, above that line, there are relatively few liquid water drops (i.e. meanr , $\text{nconc} \approx 0$), but there may still be appreciable up/downdrafts. Whereas for SS_RY, if meanr and nconc are close to zero, then latent heating is also close to zero, so we don't get these extreme divergences. Lastly, we see better agreement between SS_QSS and SS_WRF for larger magnitude of vertical wind velocity.

Note: cleaned up docstrings for wrf code!

- 5/1/20: Various figure series updates:

- `backtrack_qss`:
 - * v52: cloud LWC greater than 1e-5, abs w greater than 2 (cleaned up)
- `qss_vs_fan`:
 - * v15: cloud LWC greater than 1e-5, abs w greater than 2 (cleaned up)
- `inclrain_backtrack_qss`:
 - * v3: total LWC greater than 1e-11, LH greater than 0, T greater than 0
 - * v4: cloud LWC greater than 1e-5, abs w greater than 2 (cleaned up)
 - * v5: cloud LWC greater than 1e-5, abs w greater than 2, T greater than 0 (cleaned up)
- `inclrain_qss_vs_fan`:
 - * v11: cloud LWC greater than 1e-11, LH greater than 0, T greater than 0
 - * v12: cloud LWC greater than 1e-11, LH greater than 0, T greater than 0, [0, 42, :, :]
 - * v13: cloud LWC greater than 1e-11, LH greater than 0, T greater than 0, [1, 42, :, :]

- * v14: cloud LWC greater than 1e-11, LH greater than 0, T greater than 0, [0, 43, :, :]
- * v15: ? lost notes
- * v16: total LWC greater than 1e-11, LH greater than 0, T greater than 0, [0, 42, :, :]
- * v17: total LWC greater than 1e-5, LH greater than 0, T greater than 0
- * v18: total LWC greater than 1e-11, LH greater than 0, [0, 42, :, :]
- * v19: total LWC greater than 1e-11, LH greater than 0
- * v20: total LWC greater than 1e-5, LH greater than 0
- * v21: total LWC greater than 1e-11, LH greater than 0, T greater than 0
- * v22: total LWC greater than 1e-5, LH greater than 0, T greater than 0
- * v23: total LWC greater than 1e-11, LH greater than 0 [31, 28, :, :]
- * v24: total LWC greater than 1e-5, LH greater than 0, T greater than 0, w greater than 2
- * v25: total LWC greater than 1e-5, T greater than 0, w greater than 2
- * v26: total LWC greater than 1e-5, T greater than 0, w greater than 5
- * v27: total LWC greater than 1e-5, T greater than 0, w greater than 4
- * v28: total LWC greater than 1e-5, T greater than 0, w greater than 3
- * v29: total LWC greater than 1e-5, T greater than 0, w greater than 2 (same as v25)
- * v30: total LWC greater than 1e-5, T greater than 0, w greater than 1
- * v31: cloud LWC greater than 1e-5, abs w greater than 2 (cleaned up)
- * v32: cloud LWC greater than 1e-5, abs w greater than 2, T greater than 0 (cleaned up)
- **inclrain_cloudmap**: horizontal slice heatmap to identify where sub-saturated updrafts are occurring relative to cloudy areas
 - * v1: cloud LWC greater than 1e-5, T greater than 0, w greater than 2, t ind = 0, z ind = 42
 - * v2: cloud LWC greater than 1e-5, T greater than 0, w greater than 2, t ind = , z ind =
 - * v3: cloud LWC greater than 1e-5, T greater than 0, w greater than 2, t ind = 33, z ind = 44

- `inclrain_cloudmap_level42:`
 - * v1: cloud LWC greater than 1e-5, T greater than 0, w greater than 2
- `inclrain_cloudmap_slice201:`
 - * v1: cloud LWC greater than 1e-5, T greater than 0, w greater than 2

Also have a few non-figure runs. `inclrain_cloudmap_ptcount` counts how many subsaturated updrafts points are adjacent to (super)saturated updraft points - see docs in code. For `find_outliers`:

- v6: cloud LWC greater than 1e-5, T greater than 0, w greater than 2, `ss_wrf` greater less than 0
- v7: cloud LWC greater than 1e-5, T greater than 0, w greater than 2, `ss_wrf` greater greater than 0

Misc clean-up: secondary vars data files, code docs. Work in progress: ventilation corrections.

- 5/11/20: Besides all the figure source code described below, implemented ventilation corrections a la Pruppacher and Klett (specific page numbers cited within `make_secondary_vars_with_rain_and_vent.py`). Post file clean-up, the processed data are in `wrfout_d01_secondary_vars_no_rain`, `wrfout_d01_secondary_vars_with_rain`, and `wrfout_d01_secondary_vars_with_rain_and_vent` (names are pretty self-explanatory). Full description of variables therein in docstrings. Note these files were constructed using associated `make_secondary_vars(...).py` files in `src` `mywrf` dir.

For the following, I am referring to the conditions "cloud LWC greater than 1e-5, T greater than 0, and w greater than 2" as the "standard filter."

- `check_meanu`: meant to compare number- and mass-weighted average fall velocities b/t those computed using empirical formulae from Pruppacher and Klett and those output by WRF. Saw some outlier values that threw off the axis scaling and haven't gone back to try and fix that yet (TODO)
 - * v1: cloud LWC greater than 1e-5
- `inclrain_and_vent_backtrack_qss`:
 - * v1: standard filter but abs val w greater than 2
 - * v2: standard filter
 - * v3: DNE
 - * v4: standard filter; no figure created, just printed corrected to uncorrected meanr ratio stats

- `inclrain_and_vent_qss_vs_fan`:
 - * v1: standard filter but abs val w greater than 2
 - * v2: standard filter
 - * v3: standard filter but cloud LWC plus cloud ice greater than $1e-5$
 - * v4: DNE
 - * v5: full temperature-dependent denominator
 - `inclrain_and_vent_hist_qss`: trying to understand distribution of true SS values (note: y axis label should read "SS_WRF")
 - * v1: mistakes; ignore
 - * v2: standard filter
 - `inclrain_and_vent_fr_vs_r`:
 - * v1: mistakes; ignore
 - * v2: standard filter
 - `inclrain_and_vent_breakdown_qss`: plot each individual contribution to SS_QSS vs SS_WRF to try to better understand dependencies. Conclusion: most important functional form is that of number concentration.
 - * v1: standard filter
 - * v2: $(nconc, w) \rightarrow$ (cloud nconc, rain nconc)
 - * v3: readjust A and $\langle f(r) \cdot r \rangle$ horz axis lims
 - `inclrain_cloudmap_level42`:
 - * v2: red = standard filter and nconc less than $1e7$, blue = standard filter and nconc greater than $1e7$
 - `inclrain_cloudmap_slice201`:
 - * v2: red = standard filter and nconc less than $1e7$, blue = standard filter and nconc greater than $1e7$
 - `inclrain_lagrang_nconc`:
 - * v1: cloud LWC greater than $1e-5$
 - * v2: standard filter
 - * v3: standard filter + SS_WRF less than 0
 - `sumr_cloud_vs_rain`: to check David's question about whether `sumr_cloud` is much greater than `sumr_rain` (conclusion: it is)
 - * v1: standard filter
- 5/25/20: Have mostly been trying to figure out the points in Q4 of SS_WRF vs SS_QSS per David's request. To summarize my findings from the various figures described below, Since I've shown previously that SS_RY is a good approximation for SS_WRF, we can just as well ask about the form of the scatter plots of SS_RY vs SS_QSS (`inclrain_and_vent_backtrack_qss`

v5). As I show in `inclrain_and_vent_fancy_breakdown_qss`, the dominating contribution to both of these is the shared factor $\frac{F}{4\pi\langle fr \rangle n}$; in particular, this shared factor (colored in blue) decreases drastically on the log scale around RH=1. As a rough approximation then, we can think of the quantity I've colored in yellow as "constant" and that I've colored in red as an appropriately shifted and rescaled Heaviside function. That being more or less settled, there was still the remaining question of what experimentally-accessible filter we can use to remove those points; setting a higher vertical wind velocity cutoff was effective (this was David's suggestion and I need to ask him what is going on physically there). Finally I am still not sure how to rectify the slope of the (now fairly linear) SS-WRF vs SS-QSS plot, that is a work in progress.

- `inclrain_and_vent_breakdown_qss`:
 - * v4: standard filter A changed to $A_w/(fr)$, (fr) changed to LH
 - * v5: standard filter `nconc` changed to $1/nconc$
 - * v6: standard filter change x-axis lims
- `inclrain_and_vent_fancy_breakdown_qss`:
 - * v1: standard filter
 - * v2: same as v1 but fix formatting and take $\log(\text{abs}(\text{LH}))$
- `inclrain_and_vent_backtrack_qss`:
 - * v5: change SS-WRF to SS-QSS and switch axes
 - * v6: standard filter but `abs w` greater than 2
- `inclrain_and_vent_lwc_hist`:
 - * v1: cloud LWC distribution; edge \equiv SS-WRF less than zero but SS-QSS greater than zero
- `inclrain_and_vent_qss_vs_fan`:
 - * v6: standard filter taking only top 90% by cloud LWC of each given time step and horizontal slice
 - * v7: standard filter `w` greater than 5
 - * v8: standard filter same as v7
 - * v9: standard filter `w` greater than 8
 - * v10: standard filter `w` greater than 10 (not quite sure about v9 and v10 cause I can't find the record in my notes but easy enough to check if needed)
 - * v11: standard filter `w` greater than 7
 - * v12: standard filter `w` greater than 7, $\frac{g}{R_a T}(\frac{R_a L_v}{C_{ap} R_v T} - 1)$ changed to $\frac{-\Gamma_{env} L_v}{R_v T^2}$
 - * v13: standard filter `w` greater than 7, $\frac{g}{R_a T}(\frac{R_a L_v}{C_{ap} R_v T} - 1)$ changed to $\frac{1}{R_a T}(\frac{-\Gamma_{env} L_v R_a}{R_v T} - g)$

- `inclrain_and_vent_by_w_hist_qss`:
 - * v1: standard filter but with w greater than cutoff specified in figure title
- `inclrain_and_vent_hist_qss`:
 - * v1: mistake
 - * v2: standard filter
- `w_vs_lwc`:
 - * v1: mistake
 - * v2: standard filter
 - * v3: standard filter with LH in J/m³/s
 - * v4: standard filter and SS-WRF less than zero
- `fancy_w_vs_lwc`:
 - * v1-v3: mistakes ...
 - * v4: standard filter
- 9/21/20: Introducing revised version of `mywrf` package, `revmywrf`. Everything has been rewritten including initial data processing; many scripts which were more exploratory in nature are discarded and I am focusing on distributions of LWC, meanr, nconc, ss, and w; as well as comparing ss_qss to ss_wrf and examining its correlation with LWC and w for the paper. As with other revised packages I think the code is relatively self-explanatory but will add notes as needed.

3.4 Theory

Mostly just paraphrasing Rogers and Yau.

3.4.1 Single spherical droplet

- solve H2O vapor and heat diffusion equations
- also assume T at drop surface $\approx T_{env}$

$$\begin{aligned}
 \Rightarrow \frac{dm}{dt} &= 4\pi r^2 \rho_L \frac{dr}{dt} \\
 &= 4\pi r \rho_L \left(\frac{S - 1 - \frac{a}{r} + \frac{b}{r^3}}{F_k + F_d} \right),
 \end{aligned} \tag{1}$$

where m, r are droplet mass and radius, $S = \frac{e}{e_{sat}}$, a, b are Kohler parameters related to surface tension and soluble ion content, resp., and F_k, F_d are functions of T_{env} related to heat and H2O vapor diffusion, resp.

3.4.2 Many spherical droplets

- let the droplets be described by size distribution $p(r) \equiv \frac{\# \text{ drops with radius between } r, r+dr}{V*dr}$, where V is volume of the parcel. Then:

$$\int dr p(r) = n \quad (2)$$

$$\frac{\int dr p(r)r}{\int dr p(r)} = \langle r \rangle \quad (3)$$

etc.

(where n is the number concentration)

$$\begin{aligned} \Rightarrow \frac{dQ}{dt} &= L_v \frac{dm_{tot}}{dt} \\ &= L_v V \frac{d}{dt} \int dr p(r) \frac{4}{3} \pi \rho_L r^3 \end{aligned} \quad (4)$$

- assume $p(r)$ is constant (???)

$$\begin{aligned} \Rightarrow \frac{dQ}{dt} &= L_v V 4\pi \rho_L \int dr p(r) r \left(\frac{S - 1 - \frac{a}{r} + \frac{b}{r^3}}{F_k + F_d} \right) \\ &= \frac{L_v V 4\pi \rho_L \langle r \rangle n}{F_k + F_d} \left(S - 1 - \frac{a}{\langle r \rangle} + \frac{b}{\langle r \rangle \langle r^2 \rangle} \right) \end{aligned} \quad (5)$$

- WRF gives latent heating (LH) due to H2O diffusion in units of K/s .

$$\begin{aligned} \Rightarrow S - 1 - \frac{a}{\langle r \rangle} + \frac{b}{\langle r^2 \rangle \langle r \rangle} &= \frac{(F_k + F_d) \frac{dQ}{dt}}{L_v V 4\pi \rho_L \langle r \rangle n} \\ &= \frac{(F_k + F_d) C_{ap} m_a LH}{L_v V 4\pi \rho_L \langle r \rangle n} \\ &= \frac{(F_k + F_d) C_{ap} \rho_a LH}{L_v 4\pi \rho_L \langle r \rangle n} \end{aligned} \quad (6)$$

(below is for slides)

$$\begin{aligned}
SS_{QSS} &= \frac{FAw}{4\pi\langle f(r) \cdot r \rangle nB} \\
SS_{RY} &= \frac{F\rho_a C_{ap} LH}{4\pi\langle f(r) \cdot r \rangle nL_v \rho_L} \\
A &= \frac{g}{R_a T} \left(\frac{L_v R_a}{C_{ap} R_v T} - 1 \right) \\
B &= \rho_L \left(\frac{R_v T}{e_s} + \frac{R_a L_v^2}{P T R_v C_{ap}} \right) \\
F &= \frac{\rho_L L_v}{K T} \left(\frac{L_v}{R_v T} - 1 \right) + \frac{\rho_L R_v T}{D e_s}
\end{aligned} \tag{7}$$

3.4.3 How to derive the quasi-steady-state supersaturation formula

We begin with the definition for supersaturation S :

$$S + 1 \equiv \frac{e}{e_s}, \quad (8)$$

where $e(s)$ is the (saturation) vapor pressure. By chain rule:

$$\frac{dS}{dt} = \frac{dz}{dt} \frac{\partial S}{\partial z} \Big|_{q_v} + \frac{dq_v}{dt} \frac{\partial S}{\partial q_v} \Big|_z \quad (9)$$

In this expression we expand the first term:

$$\begin{aligned} \frac{\partial S}{\partial z} \Big|_{q_v} &= \frac{1}{e_s^2} \left(e_s \frac{\partial e}{\partial z} \Big|_{q_v} - e \frac{\partial e_s}{\partial z} \Big|_{q_v} \right) \\ &= \frac{1}{e_s^2} \left(e_s \frac{q_v R_v}{R_a} \frac{\partial p}{\partial z} \Big|_{q_v} - e \frac{de_s}{dT} \frac{\partial T}{\partial z} \Big|_{q_v} \right) \\ &= \frac{1}{e_s^2} \left(-e_s \frac{q_v R_v}{R_a} \rho_a g + e \frac{L_v e_s}{R_v T^2} \frac{g}{C_p} \right) \\ &= (S + 1) \left(-\frac{g}{R_a T} + \frac{L_v g}{R_v C_p T} \right) \end{aligned} \quad (10)$$

In the above, we have used the fact that:

$$e = \frac{q_v R_v p}{R_a} \quad (11)$$

and also hydrostatic balance¹, the Clausius-Clapeyron equation and dry adiabatic lapse rate. Next we expand the second term:

$$\begin{aligned} \frac{\partial S}{\partial q_v} \Big|_z &= \frac{1}{e_s^2} \left(e_s \frac{\partial e}{\partial q_v} \Big|_z - e \frac{\partial e_s}{\partial q_v} \Big|_z \right) \\ &= \frac{1}{e_s^2} \left(e_s \frac{R_v p}{R_a} - e \frac{de_s}{dT} \frac{\partial T}{\partial q_v} \Big|_z \right) \\ &= \frac{1}{e_s^2} \left(e_s \frac{R_v p}{R_a} + e \frac{L_v e_s}{R_v T^2} \frac{L_v}{C_p} \right) \\ &= (S + 1) \left(\frac{R_v p}{R_a e_s} + \frac{L_v^2}{R_v C_p T^2} \right) \end{aligned} \quad (12)$$

Combining equations (10) and (12) yield the expressions (7.22)-(7.24) in Rogers and Yau (3rd ed.).² Now we express the condensation rate in terms of droplet

¹In invoking hydrostatic balance, we also assume zero buoyancy.

²Note however that they have omitted a factor of $\frac{1}{S+1}$ in the denominator on the left hand side of (7.22); see e.g. equation (13-29) of Pruppacher and Klett (2nd ed.) and comments directly following.

growth (assuming a uniform size distribution):

$$\begin{aligned}
\frac{dq_v}{dt} &= \frac{d}{dt} \frac{m_v}{m_a} \\
&= \frac{1}{m_a} \frac{dm_v}{dt} \\
&= -\frac{1}{m_a} \frac{dm_c}{dt} \\
&= -\frac{1}{m_a} nV \cdot 4\pi\rho_\ell r^2 \frac{dr}{dt} \\
&= -\frac{1}{\rho_a} n \cdot 4\pi\rho_\ell r \frac{S}{F_k + F_d}
\end{aligned} \tag{13}$$

Here we have used the following notation: m_v and m_a are the masses of water vapor and dry air in the parcel, respectively; m_c is the mass of cloud water in the parcel; n is the volumetric number concentration of cloud particles in the parcel (assumed constant, i.e. no nucleation or precipitation); V is the volume of the parcel; ρ_ℓ is the density of liquid water; r is the radius of the cloud droplets; and F_k and F_d are temperature-dependent coefficients that come from solving the diffusion equations for heat and water vapor around the (spherical) droplets.

By plugging the results (10), (12), and (13) into (9) and setting $\frac{dS}{dt} = 0$ for the steady-state assumption, we obtain:

$$S = \frac{w \left(-\frac{g}{R_a T} + \frac{L_v g}{R_v C_p T} \right) (F_k + F_d)}{4\pi n r \rho_\ell \left(\frac{R_v T}{e_s} + \frac{L_v^2}{R_v C_p \rho_a T^2} \right)} \tag{14}$$

$$LWC = \sum_{r_{min}}^{r_{max}} \frac{4}{3} \pi \frac{\rho_w}{\rho_{air}} n(r) r^3 \tag{15}$$