

Biases of profiling sampling on overturns - Notes

Andy Pickering

December 9, 2016

Contents

1	Introduction	3
2	Questions/Goals	3
3	Approach	3
3.1	Data and m-files	3
3.1.1	M-files	4
3.2	Notes	4
4	Data Overview	5
5	Effect of Under-sampling Only	13
5.1	Results	13
6	Effect of non-instantaneous profiles	16
6.1	CTD scenario	16
7	Difference between actual and resampled overturn sizes	23
8	Difference in epsilon vs magnitude.	26
9	LES Model at N2	28
10	T-tide Moorings	32
10.1	Data and codes	32
10.2	Data Overview	32
10.3	Effect of Gridding on Overturns	35
10.4	Effect of Resampling on Overturns	36

11 Daily Notes	37
11.1 23 Nov. 2015	38
11.2 28 May 2015	40
11.3 23 Mar 2015	41
11.4 17 Mar 2015	42
11.5 16 Mar 2015	43
11.6 10 Mar 2015	44
11.7 9 Mar 2015	45
11.8 4 Mar 2015	47
11.9 3 Mar 2015	48
11.102 Mar 2015	49
11.1127 Feb 2015	50
11.1225 Feb 2015	51
11.1324 Feb 2015	52
11.1419 Feb 2015	53
11.1518 Feb 2015	54
11.1616 Feb 2015	55
11.1713 Feb 2015	56
11.1811 Feb 2015	57
11.1910 Feb 2015	59
11.205 Feb 2015	61
11.212 Feb 2015	63
11.2230 Jan 2015	64
11.2329 Jan 2015	66
11.2428 Jan 2015	70
11.2527 Jan 2015	70

1 Introduction

The purpose of this analysis is to examine errors in the calculation of overturns and dissipation from profiling instruments due to the time taken to complete a profile. If the background density field is being displaced vertically as it is sampled, estimated overturn sizes could be larger or smaller than the true overturn sizes. We resample high-resolution (in time) data to simulate measurement by a profiling instrument. Several scenarios are studied, including those designed to represent McLane moored profilers (MPs) and ship-board CTDs.

2 Questions/Goals

- What is the magnitude of the error/bias in overturns computed from a profiling instrument?
- How does it depend on sampling speed?
- Can we make a correction based on the background vertical velocity?

3 Approach

We resample data from T-chains (Nash) deployed during IWISE near N2 on the western ridge. These consisted of temperature loggers approximately every 10m in the lower 1500m, with fast sampling in time (1Hz?). Data is gridded onto a 10m, 2min grid, and these data are taken as the ‘true’ values in the following analysis. We then resample the data to mimic a MP or other instrument profiling up and down at a specified speed. Overturns are computed from both the true and resampled data and compared.

3.1 Data and codes

The main folder for work is

`/Users/Andy/Cruises_Research/IWISE/S9/Dissipation/OverturndBiases/` .

Data was processed and gridded by JN already and is in files `all_gridded.mat` and `all_moorings.mat`. I started with Jonathan’s code `compute_thorpe_jn_error_test.m` and reproduced his results. This code assumes that every profile in T-chain data is sampled along a slant. But in reality only a small portion of the data is sampled along the depth-time path of the profiler. His results are essentially an average over many different phasings.

I resample the Tchains in `ResampMPnew12Nov.m` to do a similar analysis with a more realistic sampling profile. The basic outline is:

1. Generate a depth-time vector of profiler position based on specified range, speed, etc..
2. Sample the real data (T-chain) along this path by interpolating to each profiler path.
3. The resampled data then has a smaller number of profiles (as sampled above) with times corresponding to the center of each up/down profile.
4. Compute overturns for the total and resampled data and compare results.

3.1.1 Code

Main script for IWISE Tchain data is: `ResampleProfiler_V2.m`.

A set of generic codes I wrote to make sample paths and resample data along them are in Github repository `/SimProfiler/`, located at : (<https://github.com/andypicke/SimProfiler>).

3.2 Notes

The epsilon that is already in the data files JN sent is different than what I get if I recompute it; I use the recomputed epsilon (see `ComputeOverturnsTchainsReal.m`).

4 Data Overview

See `PlotTchainFields.m`. Epsilon at Tchains 1 and 2 looks like it might be dominated by smaller overturns that aren't well resolved by Tchain. Maybe should just just 3 and 4?

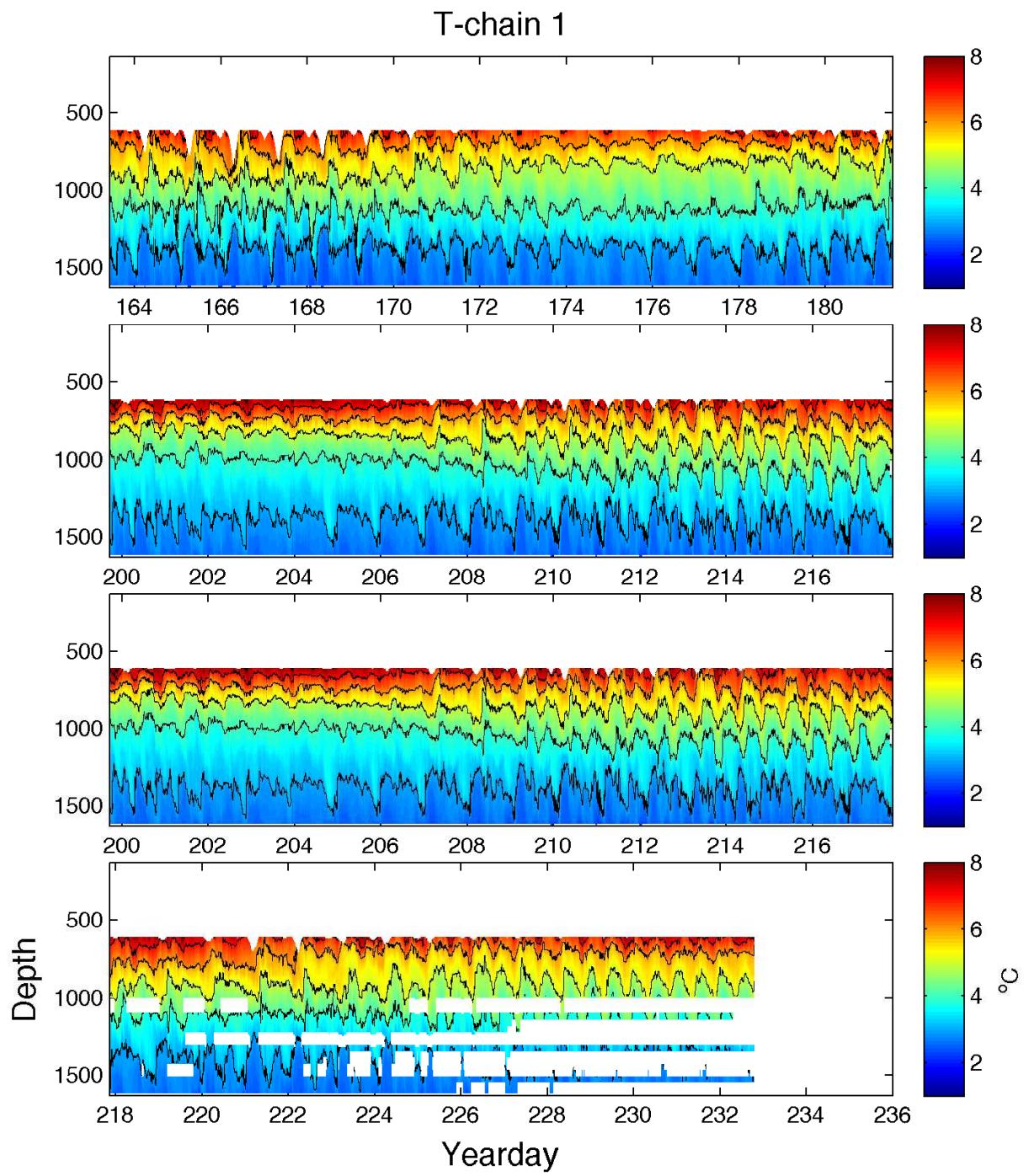


Figure 1: Overview of temperature at Tchain 1.

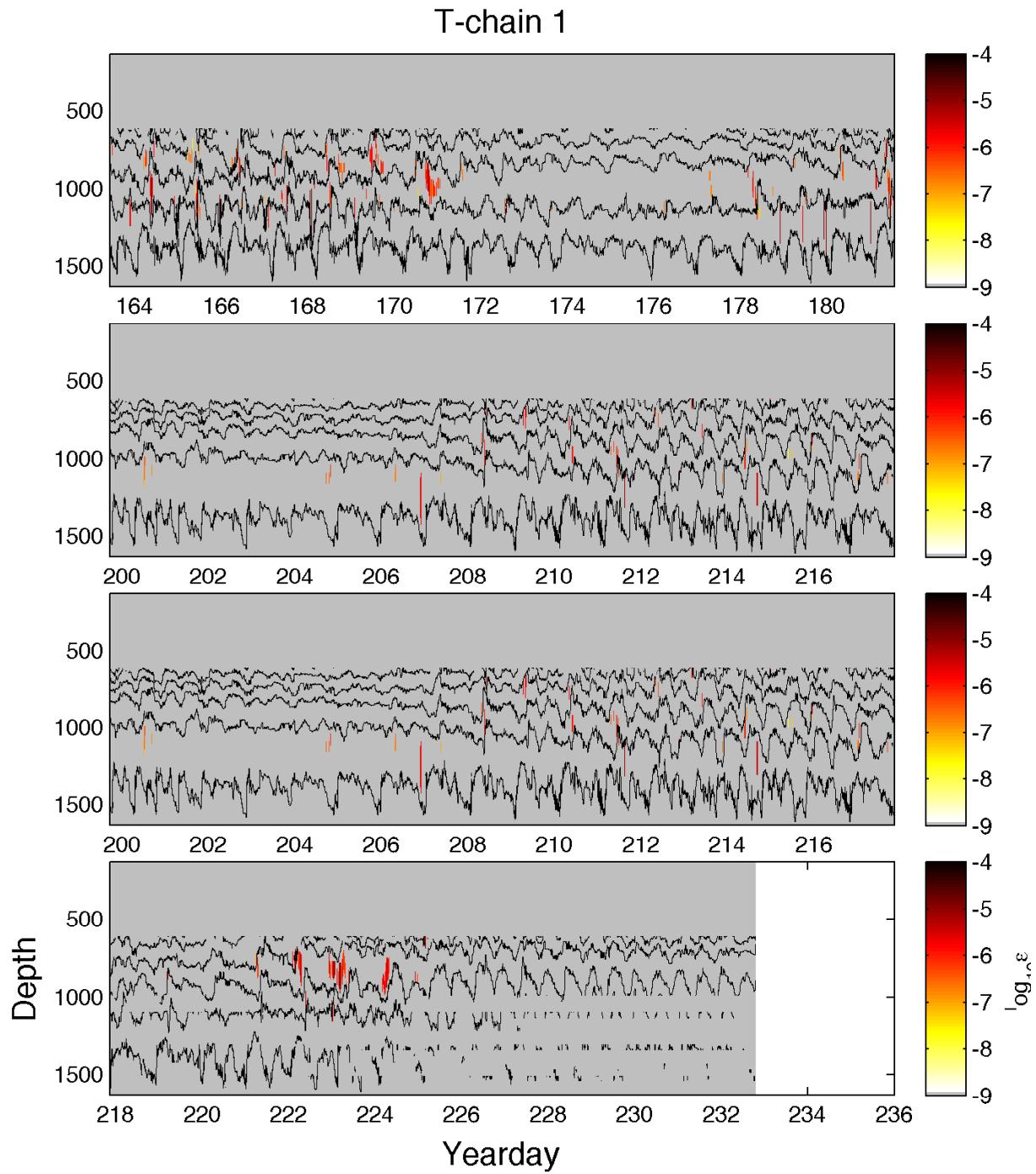


Figure 2: Overview of epsilon (color) and temperature contours at Tchain 1.

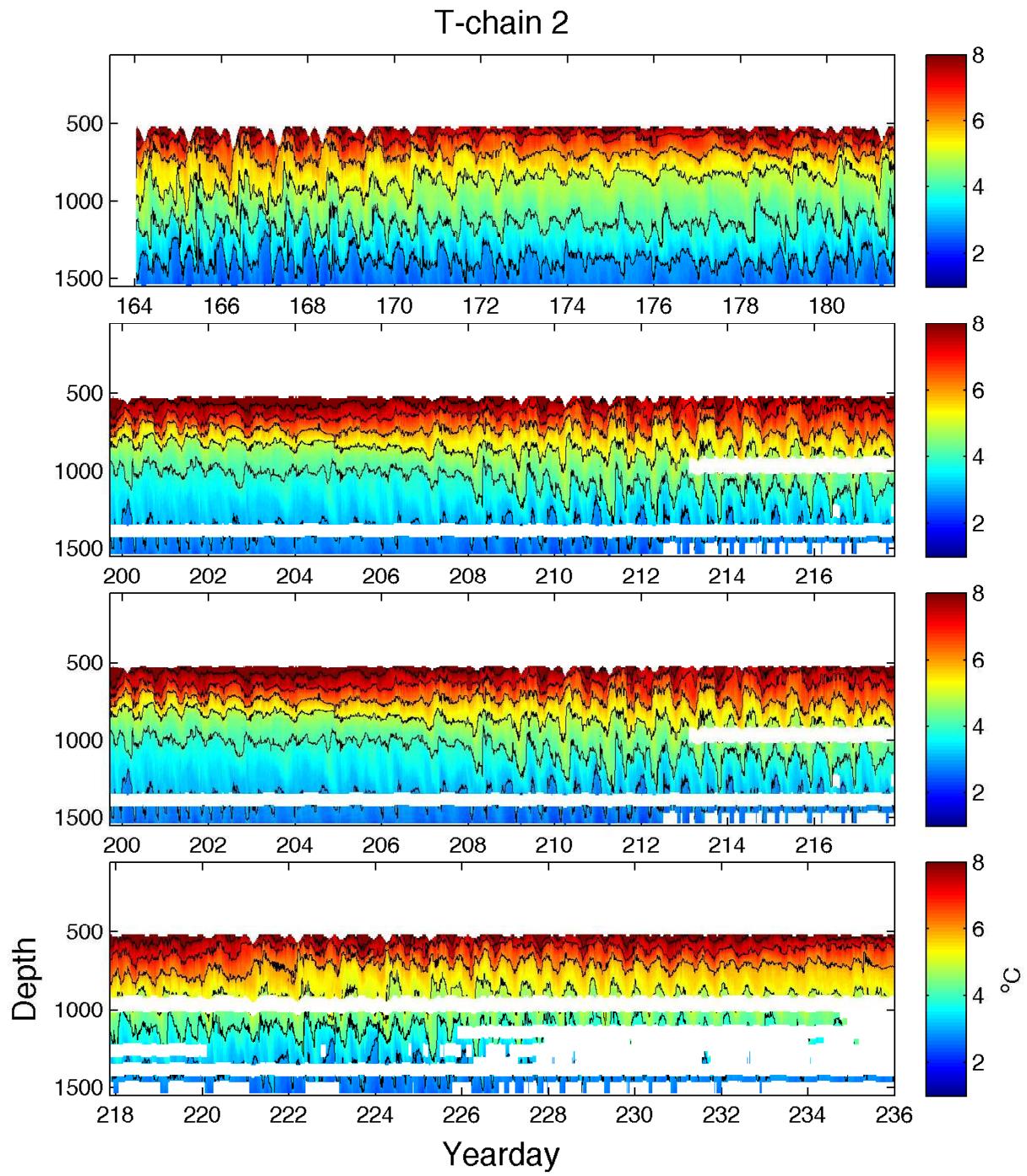


Figure 3: Overview of temperature at Tchain 2.

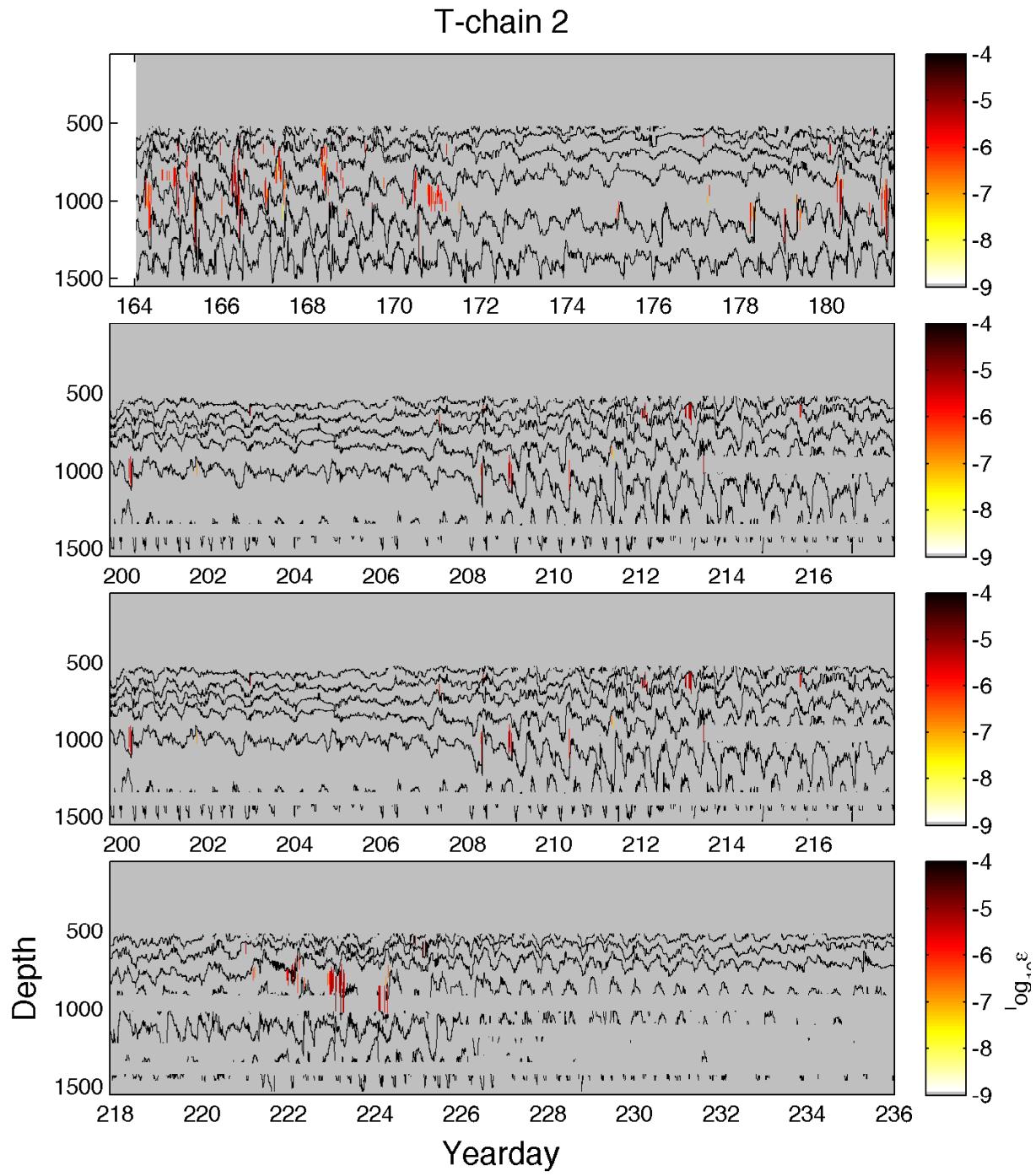


Figure 4: Overview of epsilon (color) and temperature contours at Tchain 2.

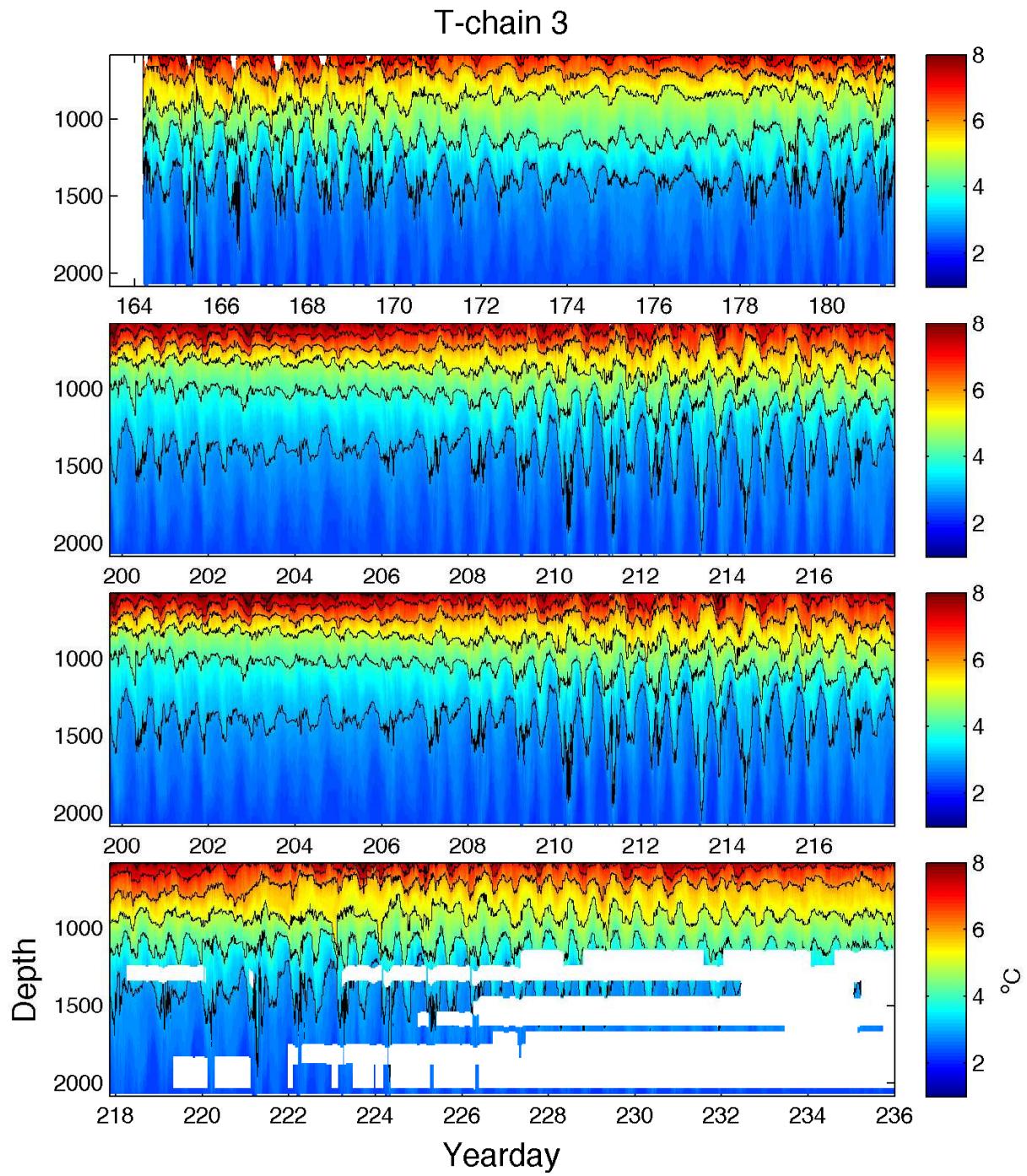


Figure 5: Overview of temperature at Tchain 3.

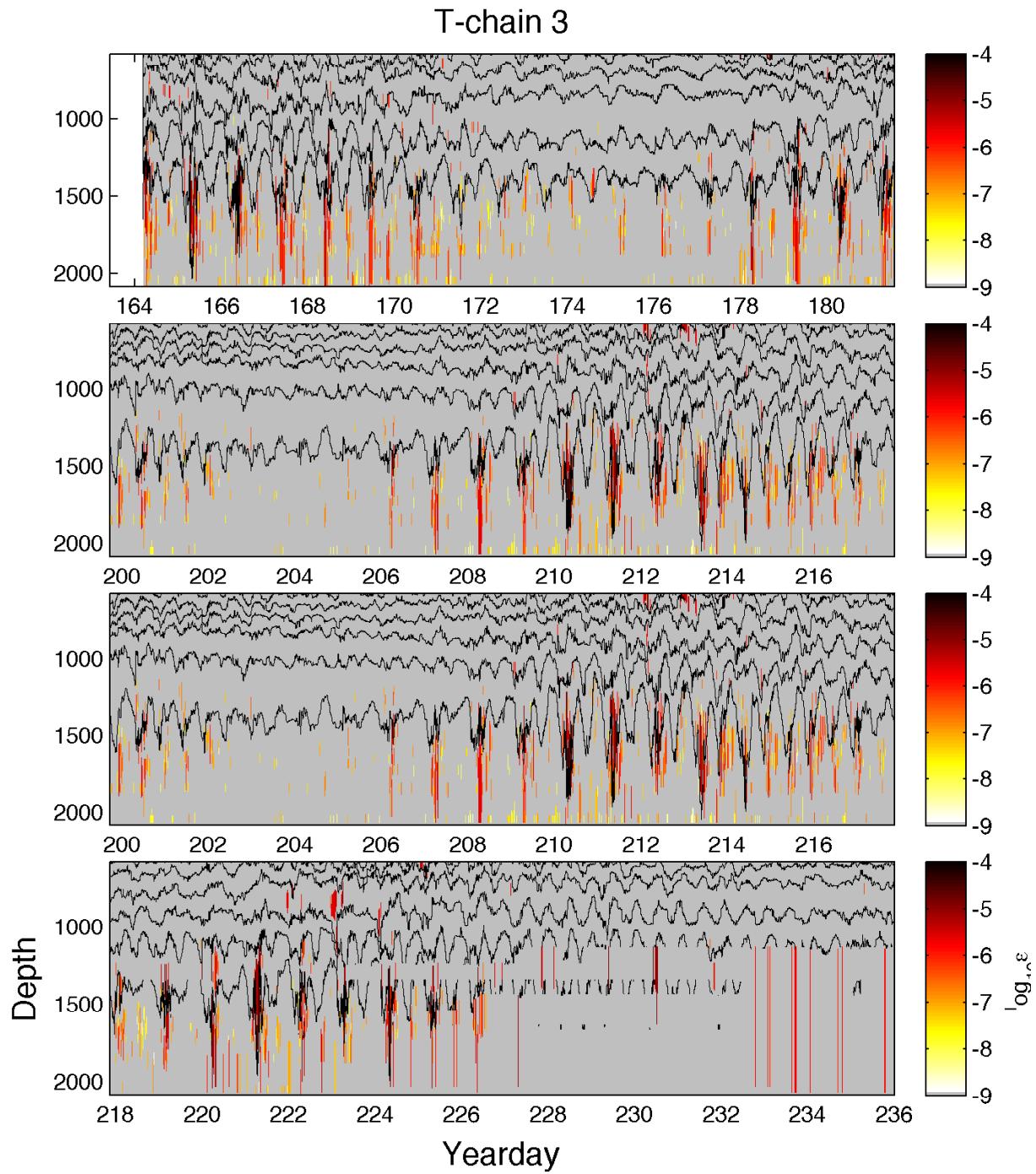
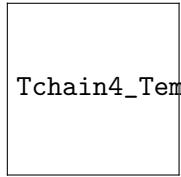


Figure 6: Overview of epsilon (color) and temperature contours at Tchain 3.



Tchain4_TempOverview.pdf

Figure 7: Overview of temperature at Tchain 4.



Tchain4_EpsilonOverview.pdf

Figure 8: Overview of epsilon (color) and temperature contours at Tchain 4.

5 Effect of Under-sampling Only

It's possible that some of the differences in the resampled data are due simply to under-sampling (as opposed to the overturns being vertically advected during the time it takes for a profile). To try to isolate this effect and estimate its magnitude, I sampled the true Tchain data only at the times corresponding to resampled profiles. See `EvalUnderSamp.m`.

5.1 Results

Under-sampling results in an error that increases with slower profiling speeds (there are less profiles/samples with slower speed). Averaged over a range of phases/start times there is no mean bias.

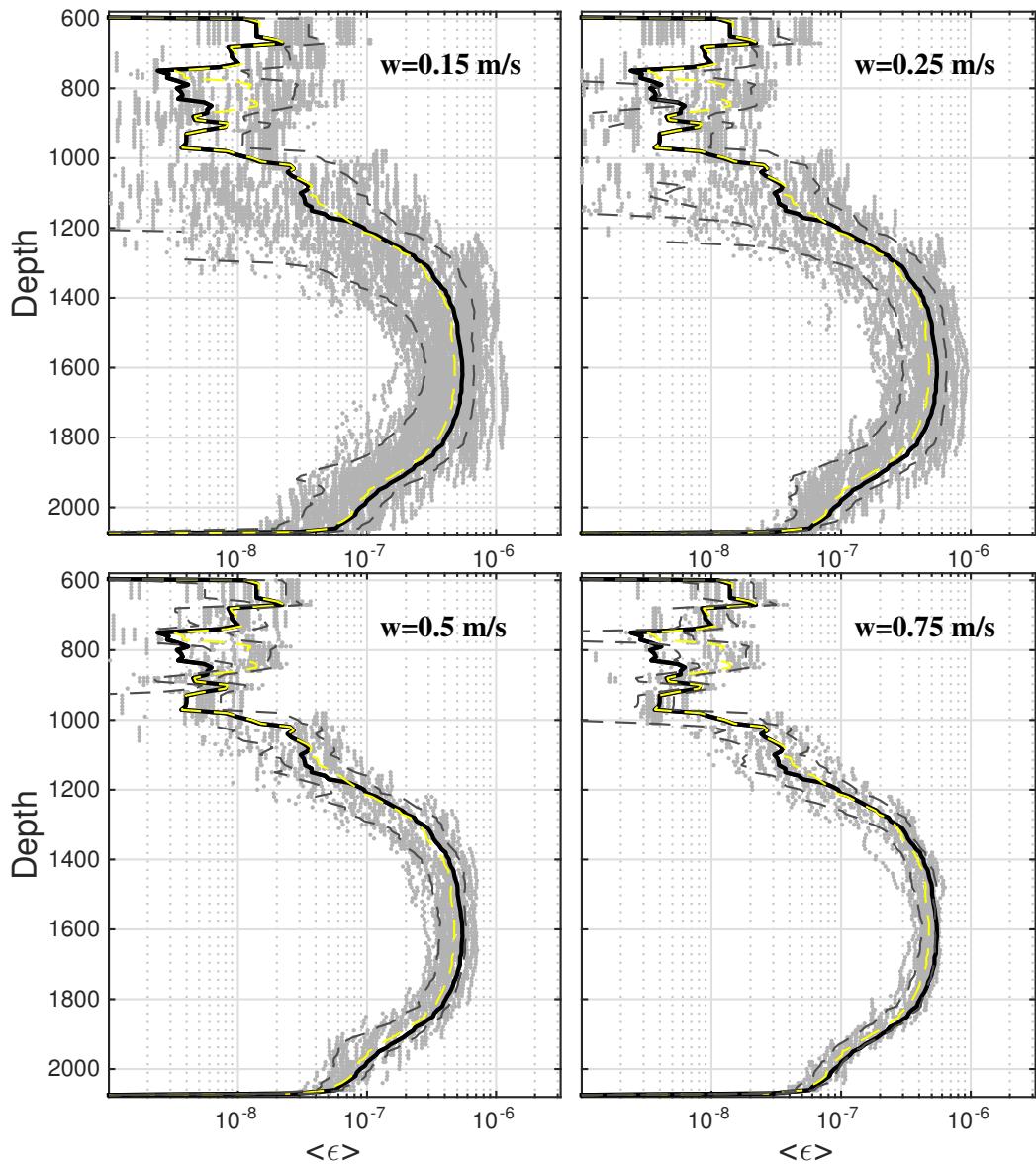
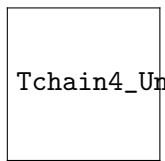


Figure 9: Time-averaged depth profiles of epsilon for different profiling speeds. Black is the true value using all Tchain data. Gray are mean profiles using only the Tchain profiles at times of resampled profiles (time at center of resampled profile). Each gray line is for profiles shifted slightly in time since phasing of sampling is unknown. Magenta is the mean of all gray lines.



Tchain4_UnderSamp_4speeds.pdf

Figure 10: Time-averaged depth profiles of epsilon for different profiling speeds. Black is the true value using all Tchain data. Gray are mean profiles using only the Tchain profiles at times of resampled profiles (time at center of resampled profile). Each gray line is for profiles shifted slightly in time since phasing of sampling is unknown. Magenta is the mean of all gray lines.

6 Effect of non-instantaneous profiles

6.1 CTD scenario

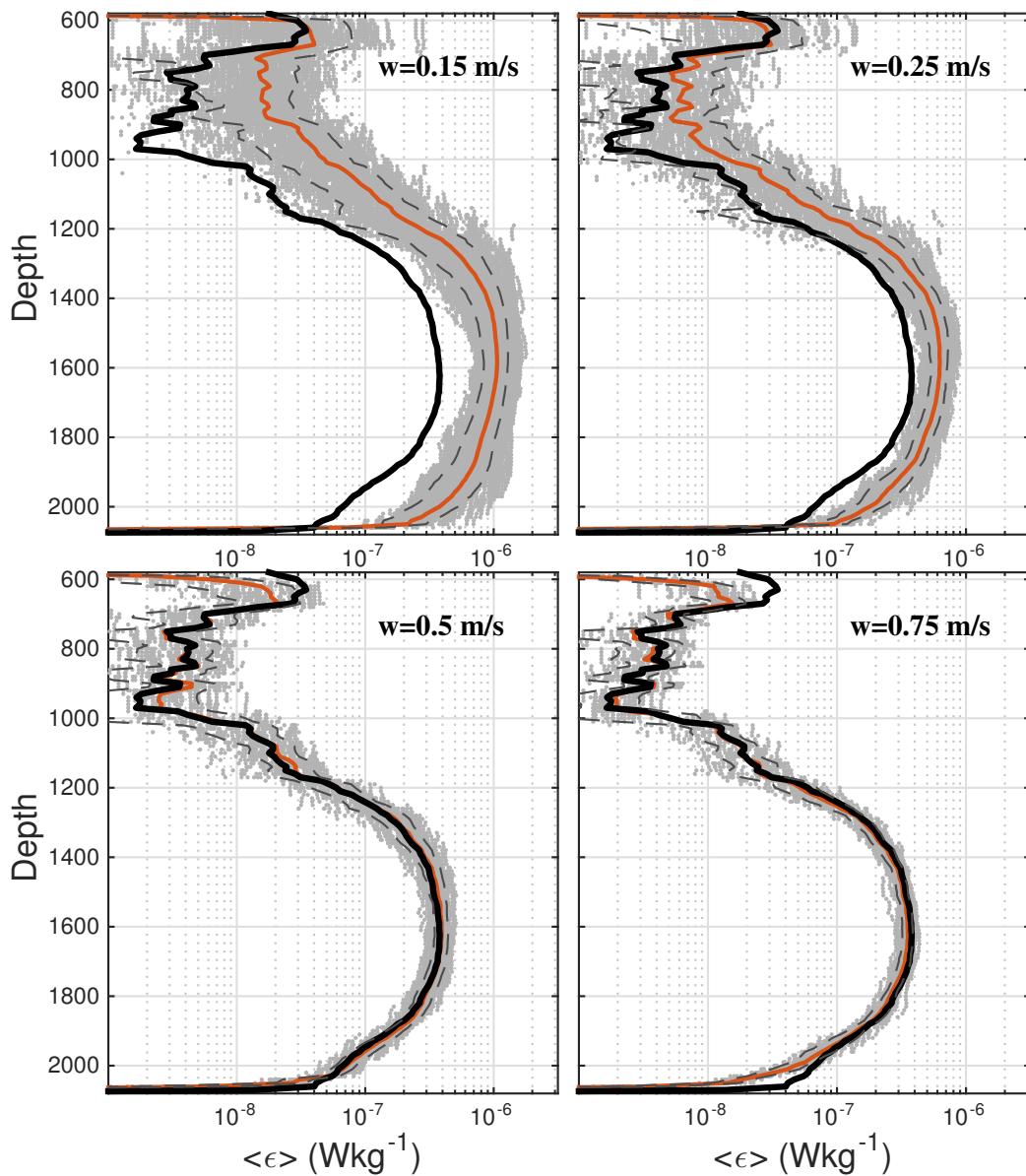


Figure 11:



Tchain4_Resamp_EpsvsDepth_DiffSpeeds_2X2.pdf

Figure 12:

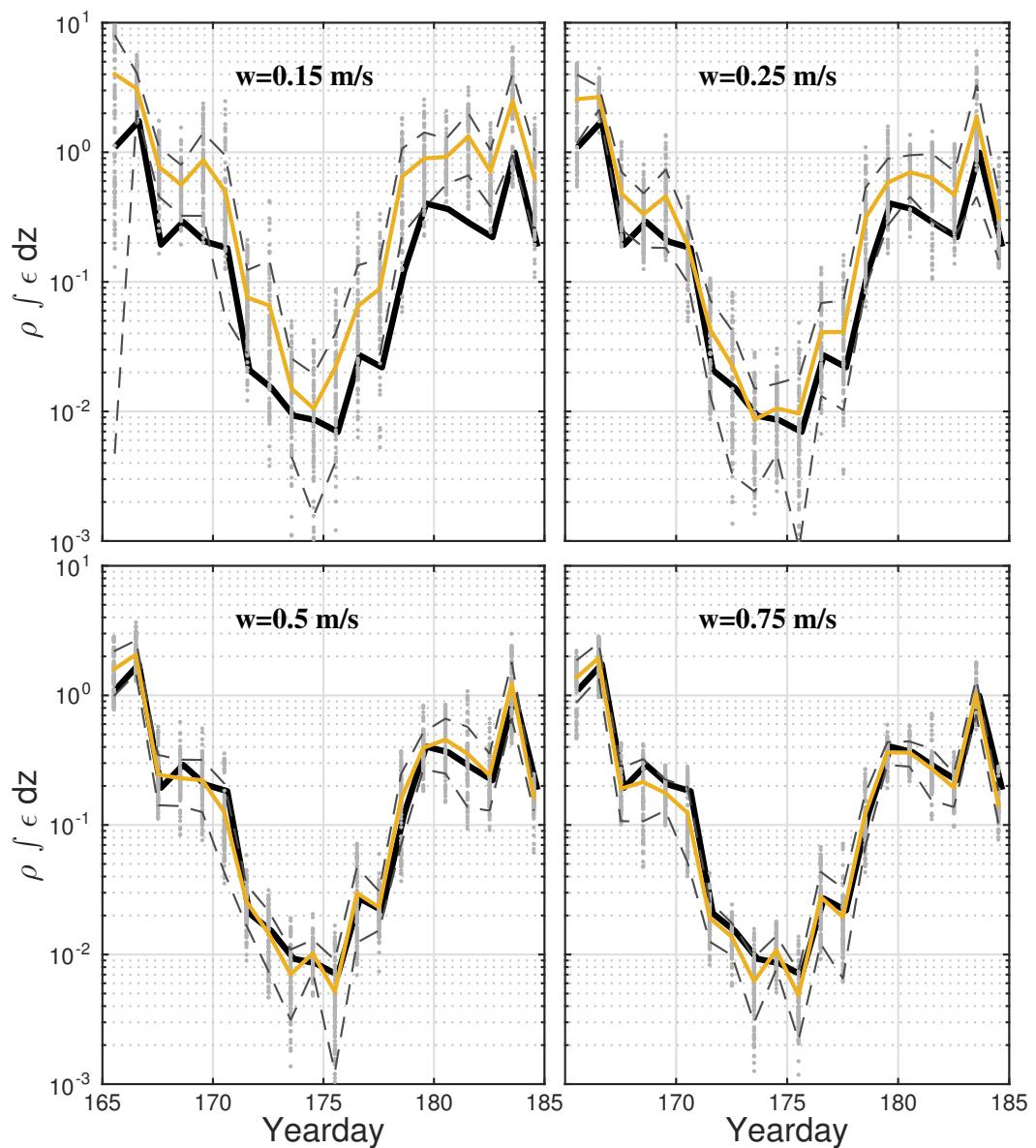


Figure 13:



Tchain4_Resamp_IntEpsvsTime_DiffSpeeds_2X2.pdf

Figure 14:

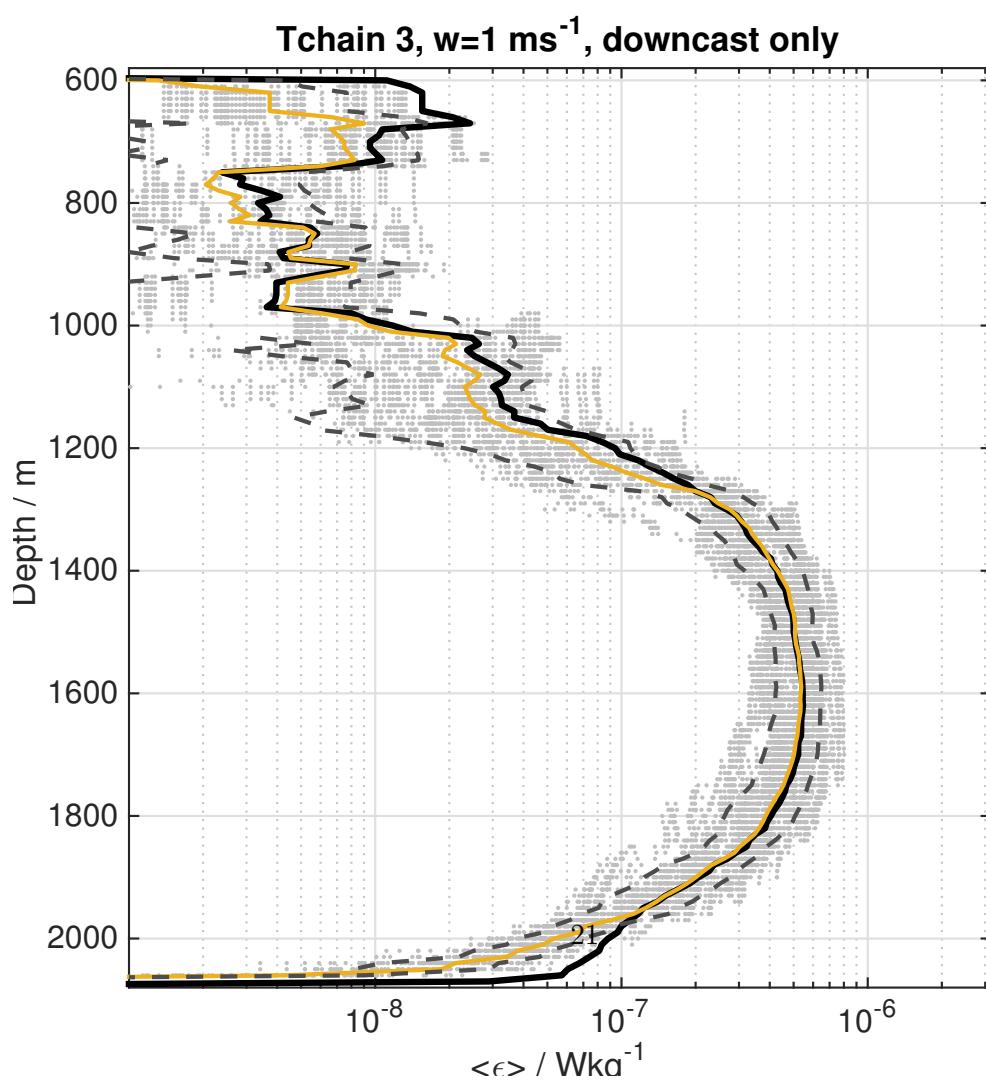
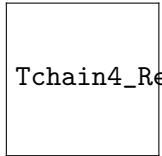


Figure 15: Simulated sampling of T-chain 3 by a CTD package profiling at 1m/s, using only downcasts. Dashed lines show ± 1 standard deviation.



Tchain4_Resamp_EpsvsDepth_CTDscenario.pdf

Figure 16: Simulated sampling of T-chain 4 by a CTD package profiling at $1m/s$, using only downcasts. Dashed lines show ± 1 standard deviation.

7 Difference between actual and resampled overturn sizes

Histograms of the difference between true and resampled overturn sizes look symmetric. As the profiling speed is increased, the distribution becomes narrower.

See `PlotResampVsTrue_direct.m`.

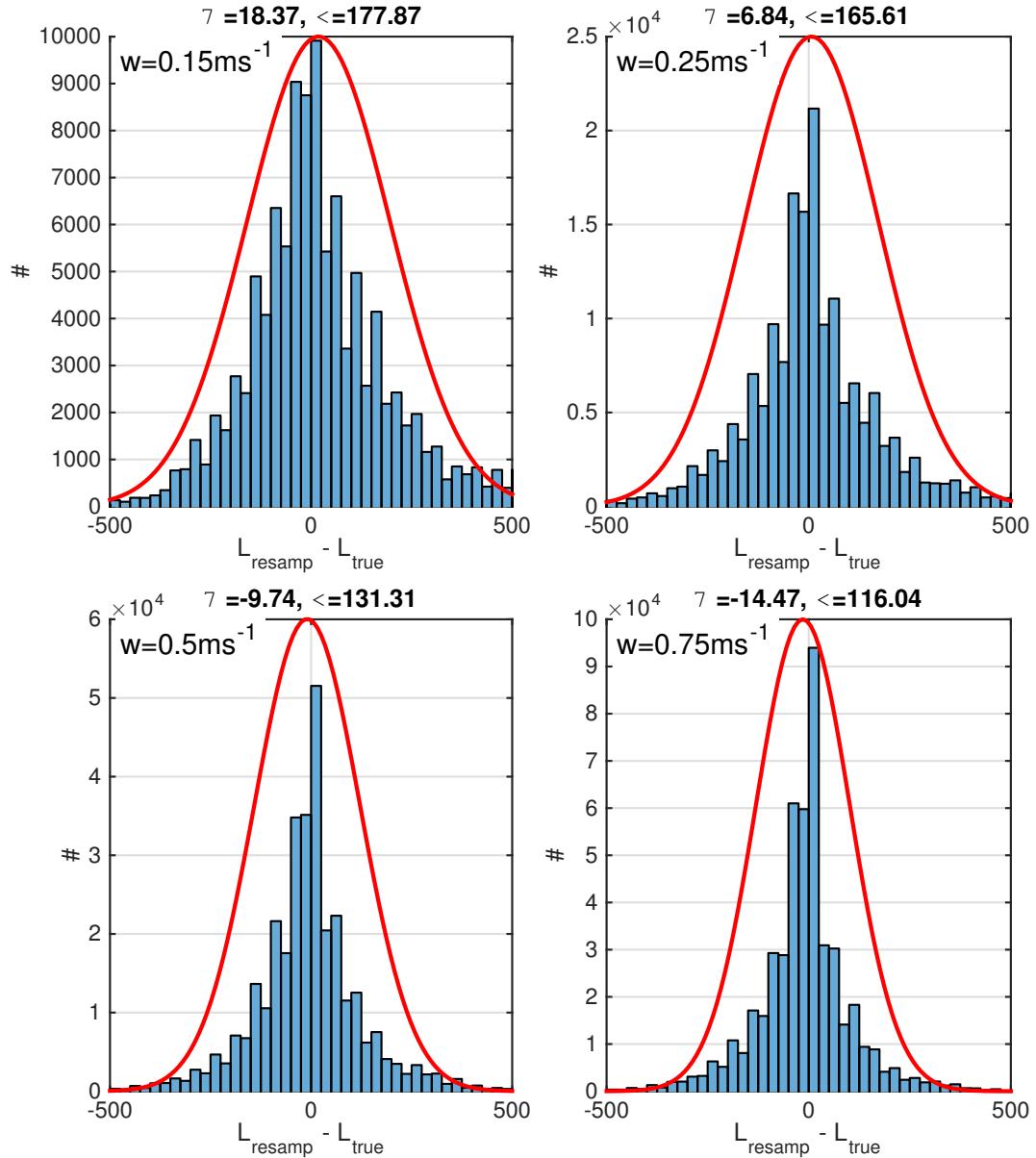
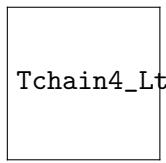


Figure 17: Histograms of the difference between true and resampled overturn sizes at T-chain 3 for 4 different speeds. Red curves are fit to normal distribution (scaled to fit on axes).



Tchain4_LtDifHist_4Cases.pdf

Figure 18: Histograms of the difference between true and resampled overturn sizes at T-chain 4 for 4 different speeds. Red curves are fit to normal distribution (scaled to fit on axes).

8 Difference in epsilon vs magnitude.

It appears that the error between resampled and true epsilon is proportional to the magnitude of the true epsilon.

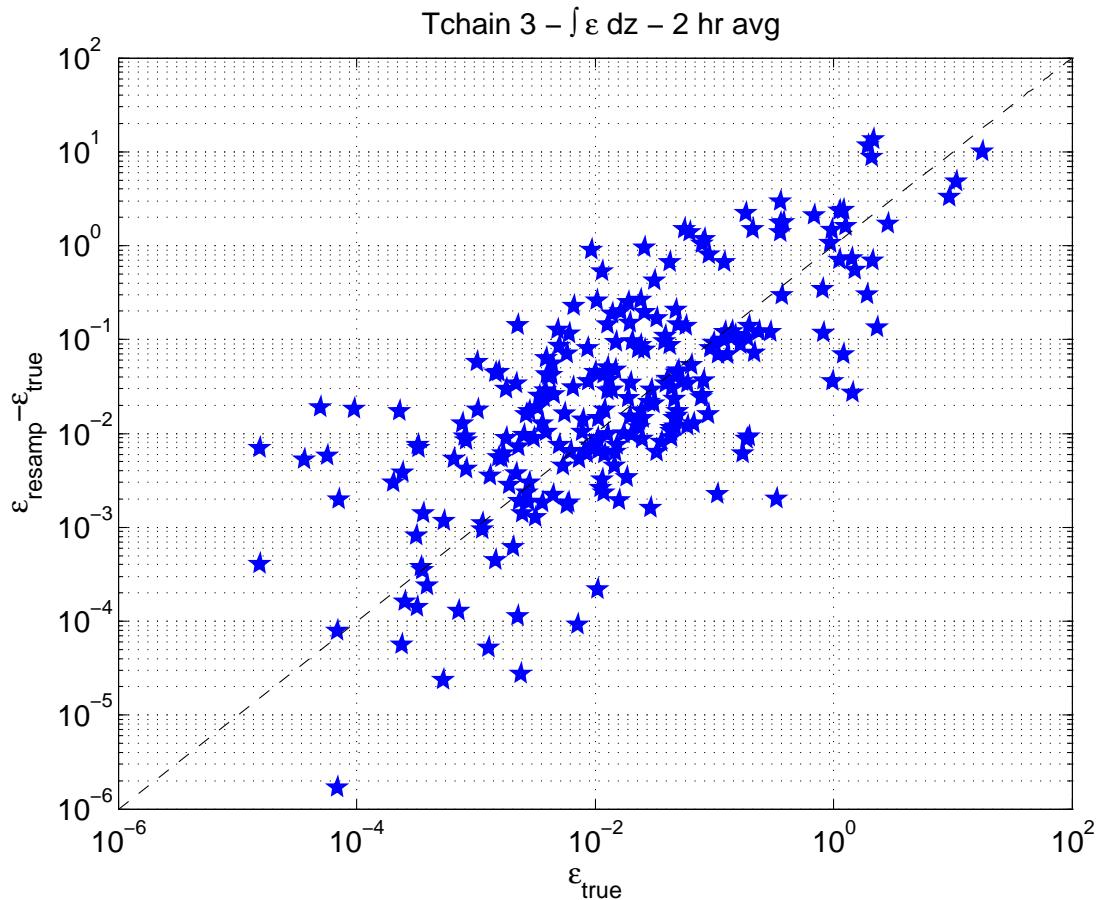


Figure 19: Scatter plot of the difference between resampled and true depth-integrated epsilon versus the true epsilon.



Tchain4_IntEps_ErrorvsTrue.pdf

Figure 20: Scatter plot of the difference between resampled and true depth-integrated epsilon versus the true epsilon.

9 LES Model at N2

Main directory is /IWISE/Analysis/S9/Dissipation/OverturndBiases/LES.

Data and m-files:

- Data (except for dissipation) are in `N2_all.mat`, which is a cell array with one cell for each time.
- I combine this data into a depth-time structure in `Make_LES_N2_DepthTime.m` and save as `N2_depth_time.mat`.
- Dissipation (directly computed from velocity shear) data are in `N2turbdiss_ZT.dat`.
- I read this into matlab and make into a depth-time structure in `Import_N2_LES_diss.m`.
- Compute OT for the true LES data in `Compute_OT_LES.m`.
- In `CompareLES_N2_OTvsDiss.m` I compare epsilon from overturns to the direct epsilon computed by Masoud (using shear).
- I then resample LES data and compute overturns in `Resample_LES_OT.m`.

Summary of LES overturns findings so far (see LES section of notes for details).

- Epsilon from overturns is about 1 order of magnitude larger than directly computed epsilon. Masoud found something similar also.
- Masoud assumed L_t equals L_o ; should ask him if he compared the two.
- Resampled overturns don't appear to have much of a bias at any speed. This differs from what I found at the T-chains.
- N2 is more on top of ridge, maybe it is different at that location?
- Still want to do comparison at Tchain locations also to see if results are similar.

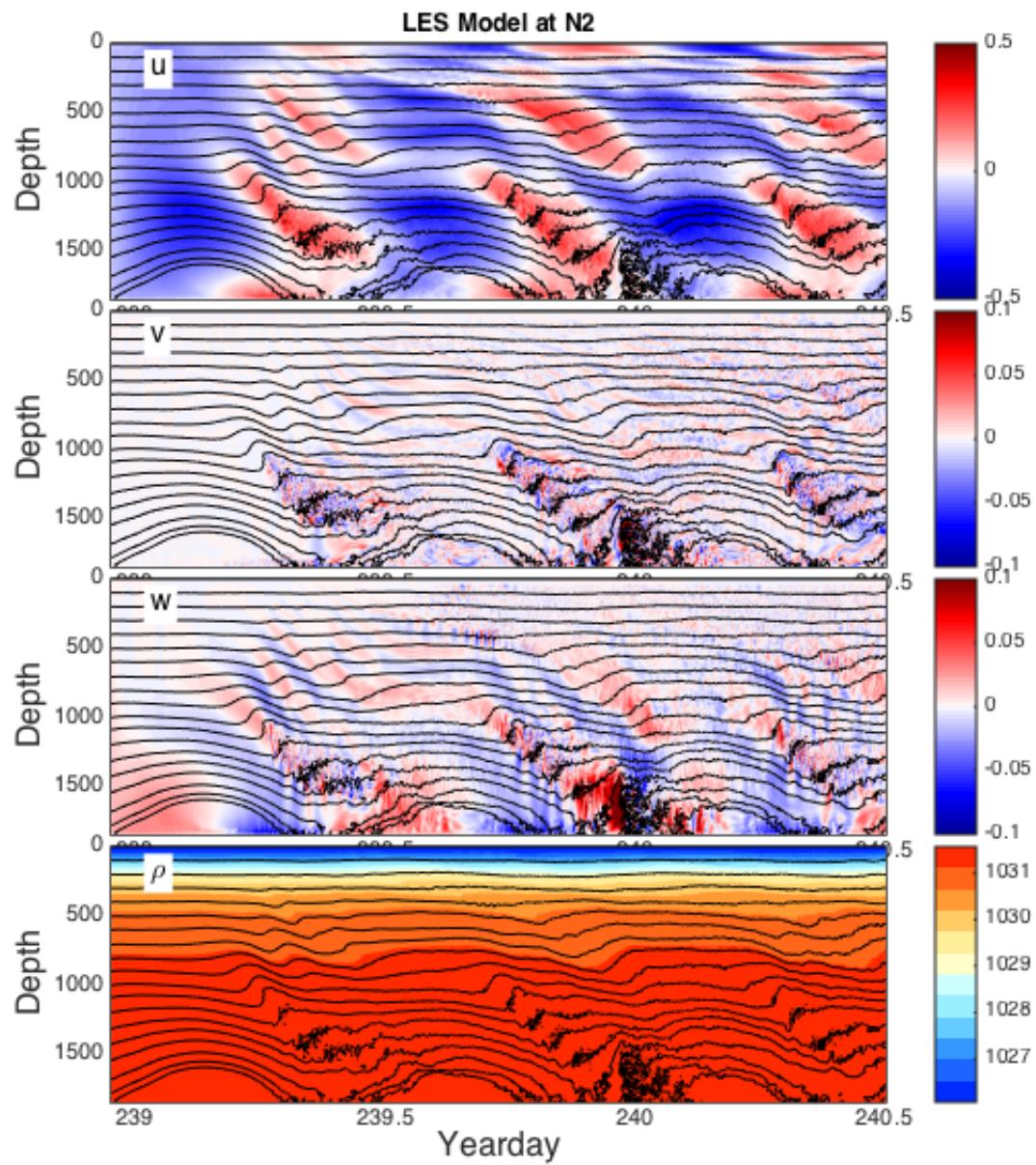


Figure 21: LES model data at N2 location. Data have been interpolated onto an evenly spaced depth grid (10m).

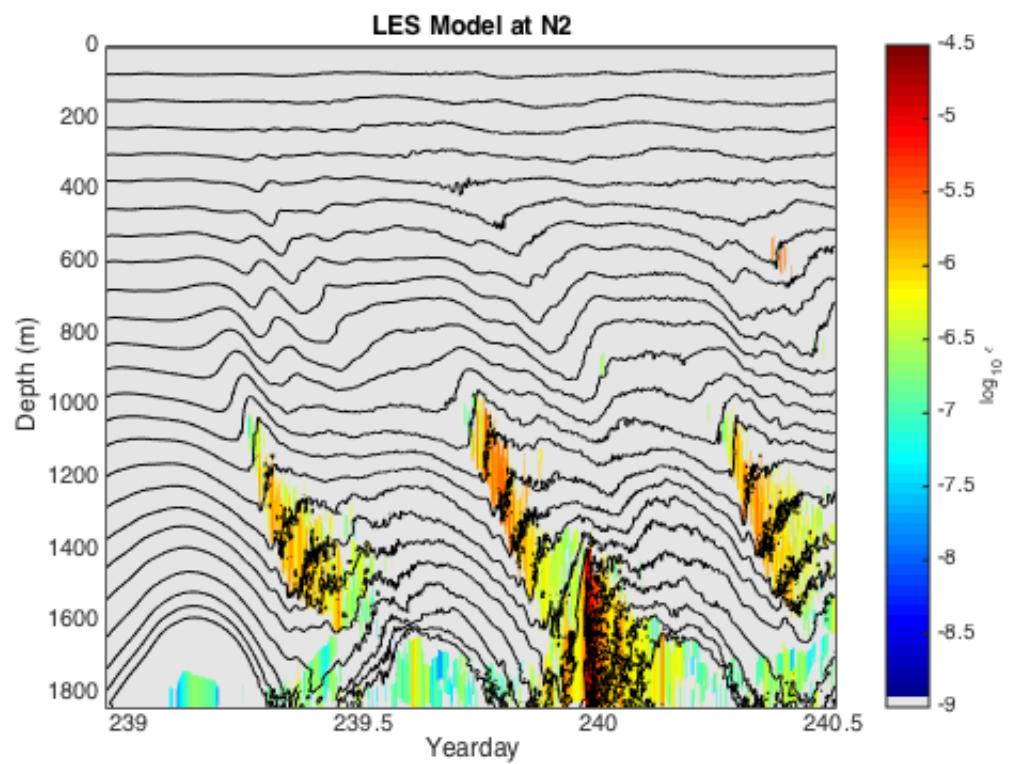


Figure 22: Epsilon computed from overturns in LES model data at N2 location.

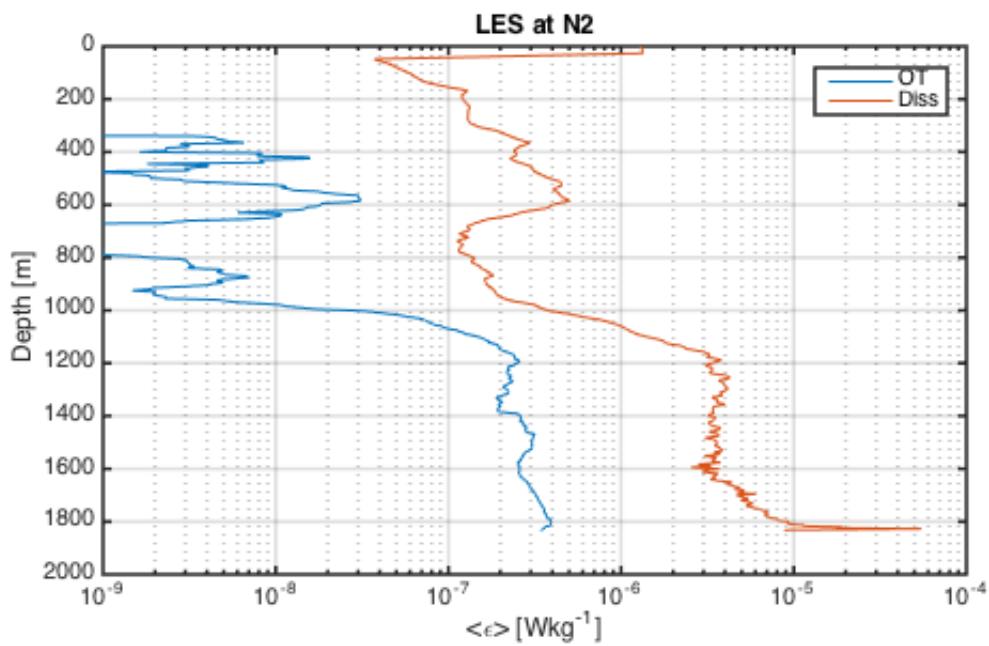


Figure 23: Time-mean epsilon computed from overturns (OT) and directly from shear (Diss) in LES model data at N2 location.

10 T-tide Moorings

Looking at ‘T2’ mooring from T-tide cruise. Thermistors spaced 10-30m apart measured the bottom 600m.

10.1 Data and codes

Main directory is `/IWISE/Analysis/S9/Dissipation/OverturndBiases/Ttide`.

- Original data is in `/Users/Andy/Dropbox/ttide_jn/T2_short_timeseries.mat`. This is a rough processed dataset that JN made for me to do this resampling analysis. Not corrected for mooring knockdown? Absolute depths not correct but probably doesn’t matter for this.
- `ExamineT2.m` - Quick look at data etc. (see Figure 24).
- `Make_T2_grid.m` - Interpolate T2 to evenly spaced depth and time grid.
- `ComputeOverturndT2_gridded.m` - Compute overturns (all profiles, not resampled).
- `ComputeOverturndT2_gridded.m` - Compute overturns (all profiles, not resampled) using gridded data.
- `ResampleT2.m` - Resample data to simulate profiling instrument, and compute overturns for ensemble.
- `CombineResampTestCases_T2.m` - Combine resampled data into one structure for ensemble.
- `PlotResampT2.m` - Plot results of resampling etc.

10.2 Data Overview

Figure 24

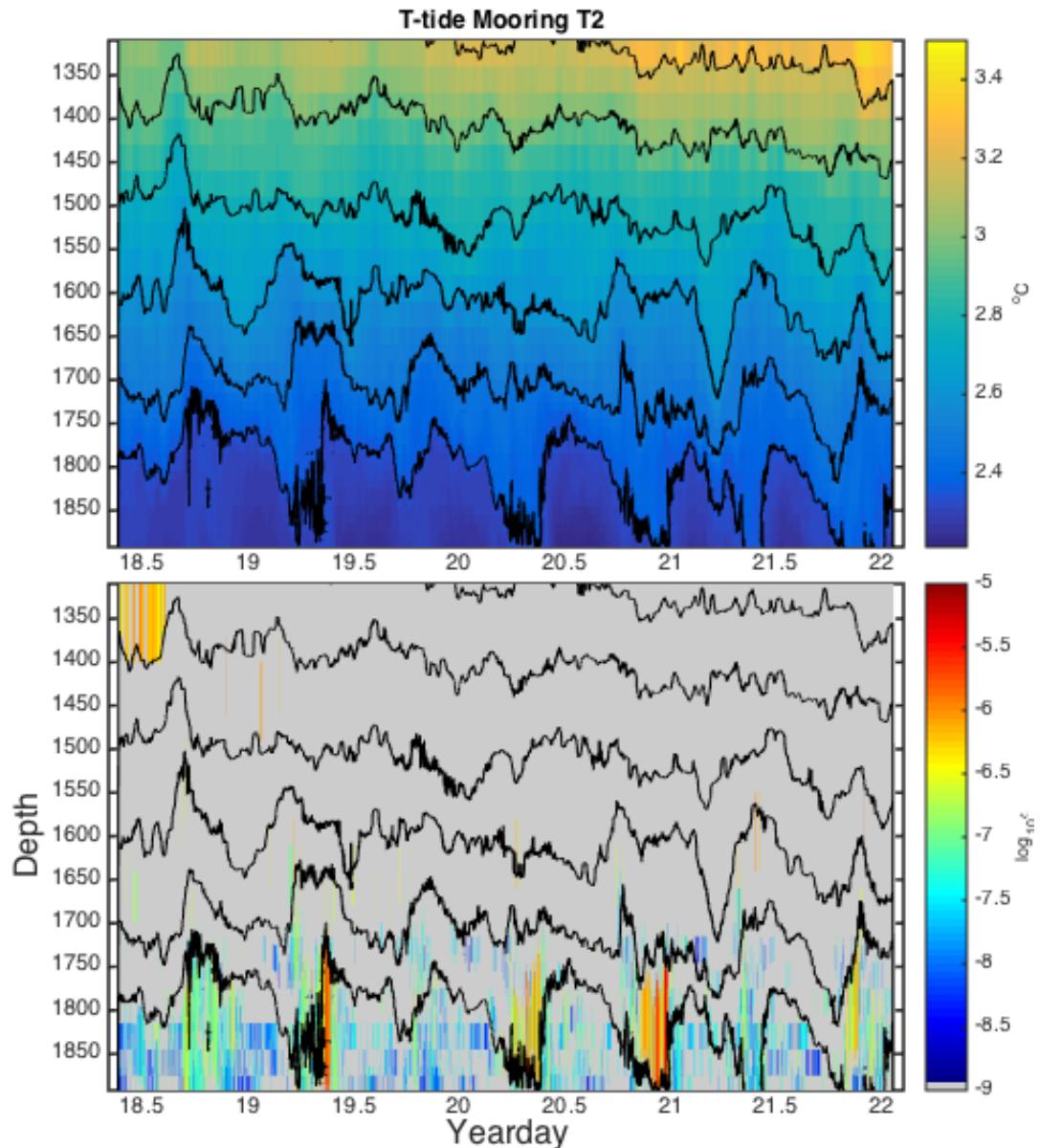


Figure 24: Temperature, epsilon from overturns and isotherms at T-tide mooring T2.

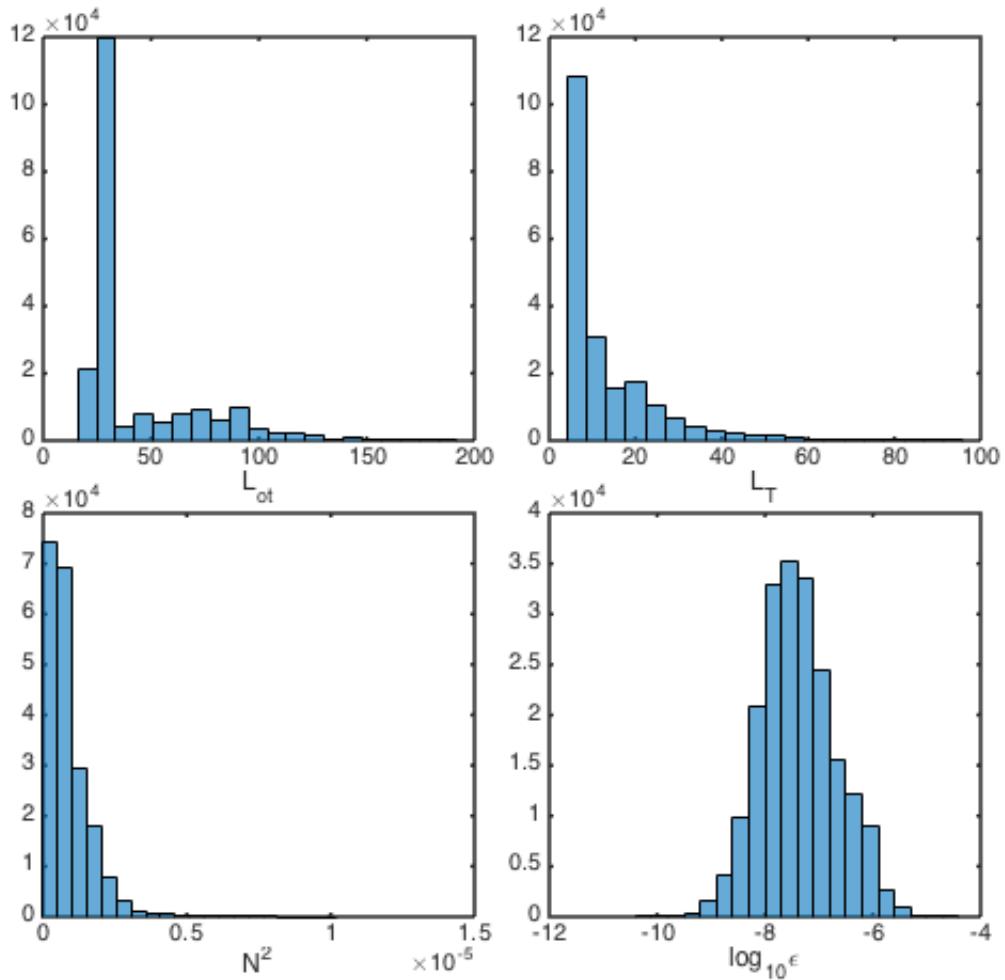


Figure 25: Histograms of overturn quantities at T2 (raw ungridded data).

10.3 Effect of Gridding on Overtuns

I noticed that interpolating the data onto an evenly space depth/time grid affected the mean epsilon profile computed from overturns. So I investigated more to determine how much the effect was and what causes it.

Summary:

- It is mainly the gridding in depth, gridding in time doesn't have much of an effect.
- Mean gridded profiles of epsilon are all smaller than the raw profile.
- They don't go in order of increasing dz.

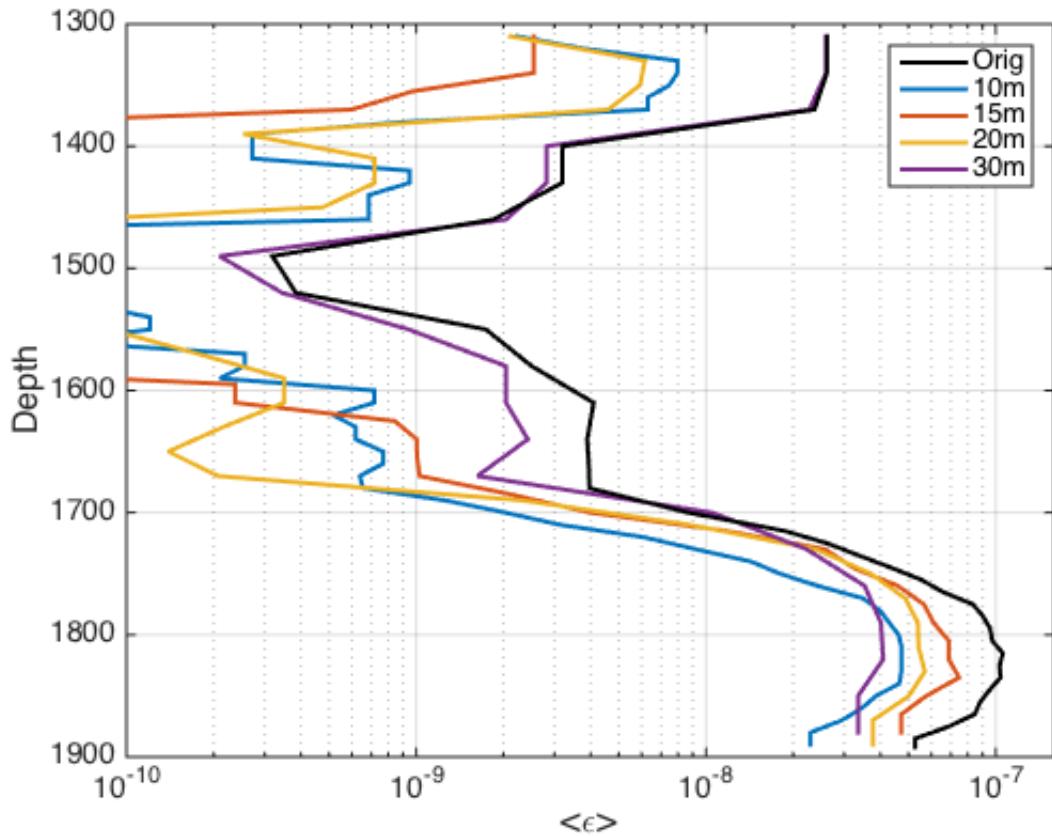


Figure 26: Time-mean profiles of epsilon for different depth gridding of data. .

10.4 Effect of Resampling on Overtuns

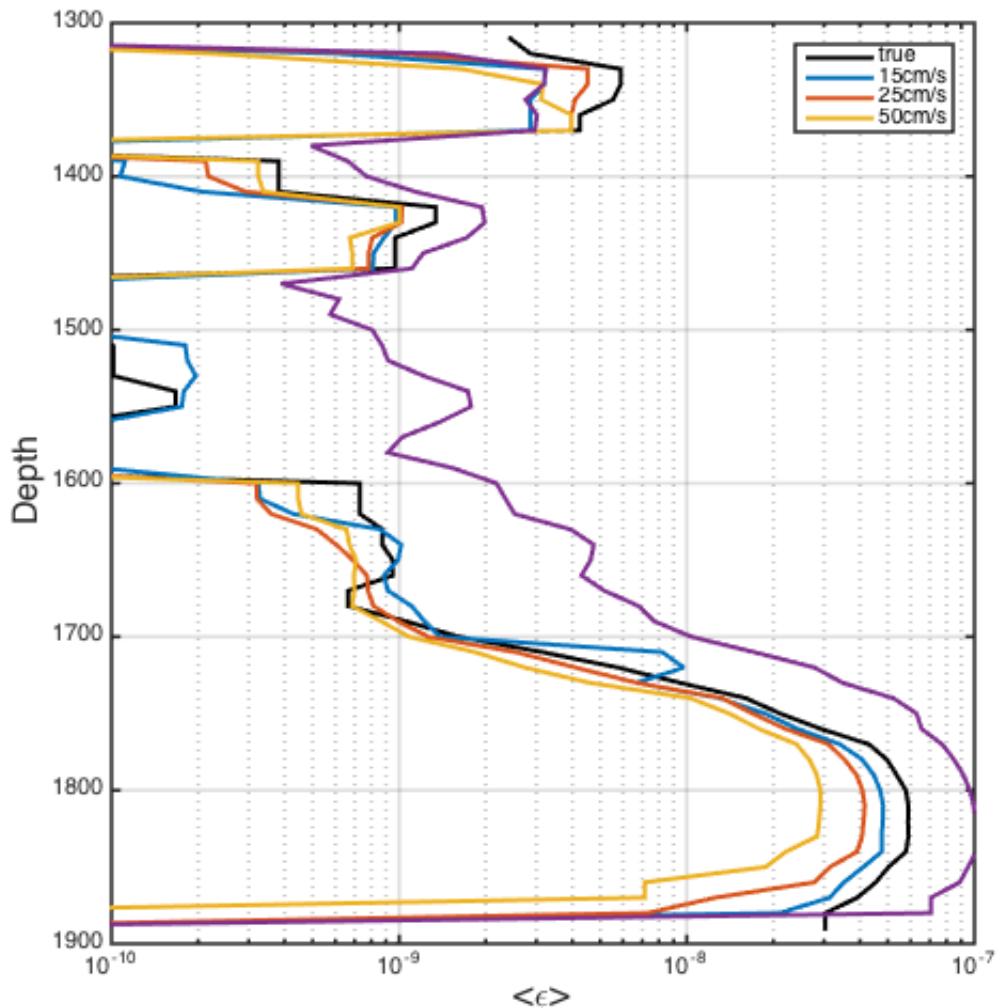


Figure 27: Time-mean profiles of epsilon for different resampling speeds.

11 Daily Notes

11.1 23 Nov. 2015

Again getting back to this... Need to try computing a ‘turbulent velocity scale’ to see if we can identify biased overturns etc...Looks like I started trying this in `Misc16Mar.m`. Working with new script `Correct_Resampled_Eps.m`. I compute a ‘turbulent velocity scale’ as $w_t = L_T N$, which follows from

$$\epsilon = L_T^2 N^3 \quad (1)$$

$$\epsilon \propto U^3 / L \quad (2)$$

Then I assume where this is greater than the profiler sampling speed (or some fraction of it), values are biased and throw them out.

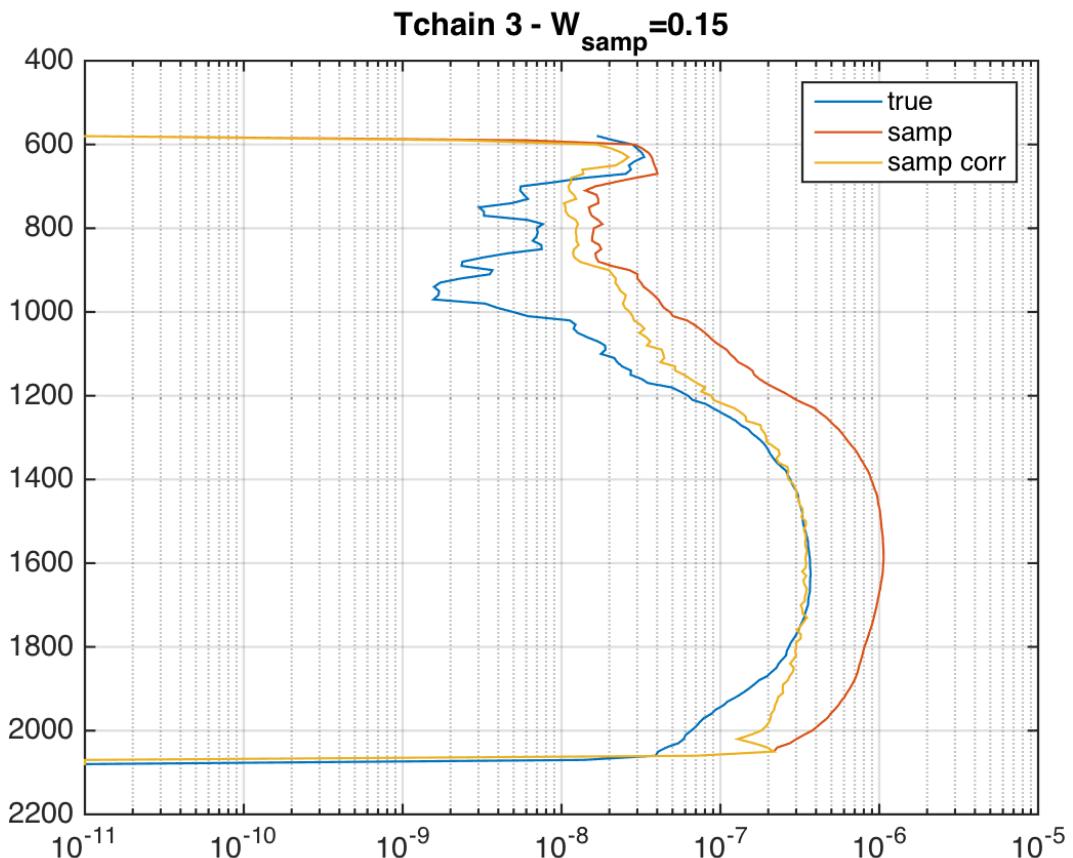


Figure 28: Time-mean profiles of epsilon. For the 'corrected' profile, data where inferred turbulent velocity was greater than 50 percent of the sampling speed were thrown out.

11.2 28 May 2015

Getting back to this after a long break and trying to remember where I left off. Main script for IWISE Tchain data is: `ResampleProfiler_V2.m`. The core codes to make sample paths and resample are in the folder `/SimProfiler/`

11.3 23 Mar 2015

** to do - compare w computed from sampled isotherms to actual w from Tchain data. Try averaging over the period of each profile. Is there any useful correlation?** in Misc16Mar.m

Still trying to think of ways to identify biases in MP data (see 16 mar notes).

Updated list:

- Find depths of an isopycnal in two consecutive profiles and compute slope (dz/dt). If slope is close to or exceeds the profiling speed, flag as possibly bad. I tried this and unless sampling is fast, this always gives a w that is much smaller than the large values in data.
- Variation of above: Compute w in a similar way, using profiles before and after each profile. If this w is in same direction as profiler w and exceeds some fraction of the profiler w, flag as possibly bad. Same issues as above, if feature we are trying to capture has timescale shorter than the profiling.
- Use the actual w measured from the MP. In theory, this works but in practice it is difficult to measure w accurately and would require a significant amount of time working on the code.
- Compare consecutive up/down profiles. This effect should be much greater in one, depending on which direction w is in. Problem: the timescale of the overturning patch has to be large enough to be in consecutive profiles. For slower sampling this becomes unlikely.
- Compare average profiles from up and down casts separately? They should be the same if this effect isn't present. Didn't see any consistent patterns in Tchain 3.
- Look at shapes of overturns? Is there something consistently different between real and sampled overturns?

11.4 17 Mar 2015

Working on new function `ResampleFieldGeneral.m` to generalize the resampling done in `ResampleProfiler.m` and make it easy to apply to any dataset. This way I can also modify the code and not have to replicate those modifications in every separate script for each dataset.

Overtures can be added to this `REsamp` structure with `AddOverturesToREsamp.m`

11.5 16 Mar 2015

Trying to come with ways to test whether sampling/overtures are biased from profiler data only (ie real life).

- Find depths of an isopycnal in two consecutive profiles and compute slope (dz/dt). If slope is close to or exceeds the profiling speed, flag as possibly bad. I tried this and unless sampling is fast, this always gives a w that is much smaller than the large values in data.
- Variation of above: Compute w in a similar way, using profiles before and after each profile. If this w is in same direction as profiler w and exceeds some fraction of the profiler w , flag as possibly bad.
- Use the actual w measured from the MP. In theory, this works but in practice it is difficult to measure w accurately.

* Idea: try taking a woa profile and mode-1 IT w , making synthetic field, sampling to see if we get any false overtures.

11.6 10 Mar 2015

Found bug in my resampling code. Was interpolating to the same depth vector for each profile, instead of using alternating up/down (so I think basically using just down profiles). Fixed this in `ResampleProfiler.m`. Doesn't seem to have much effect on the mean profiles or the along-path plots though..

WAIT now there is a bias at Tchain2? Check Tchain1, Ttide mooring also

11.7 9 Mar 2015

I made a bunch of plots of epsilon along the actual sampling paths, following JN's suggestion. In a lot of them, at slow sampling speeds there are a few 'stripes' of much larger epsilon that I think are largely responsible for the larger mean profiles (ex. Figure 29). It appears that they might at places where the profiler is traveling in the same direction as large isopycnal displacements. Basically it happens in places on depth-time maps where the slope of the sampling path is the same as the slope of the isopycnals... I'm wondering if the bias doesn't show up in other data because the vertical velocities are smaller (so if I sampled even slower, I might see something similar).

* make same figures for other Tchains. * if slopes not equal at other tchains, try slower sampling.

What is different about Tchain 3 and 4? Overtures are larger. Overtures last longer and have larger vertical extent. Vertical velocities are larger.

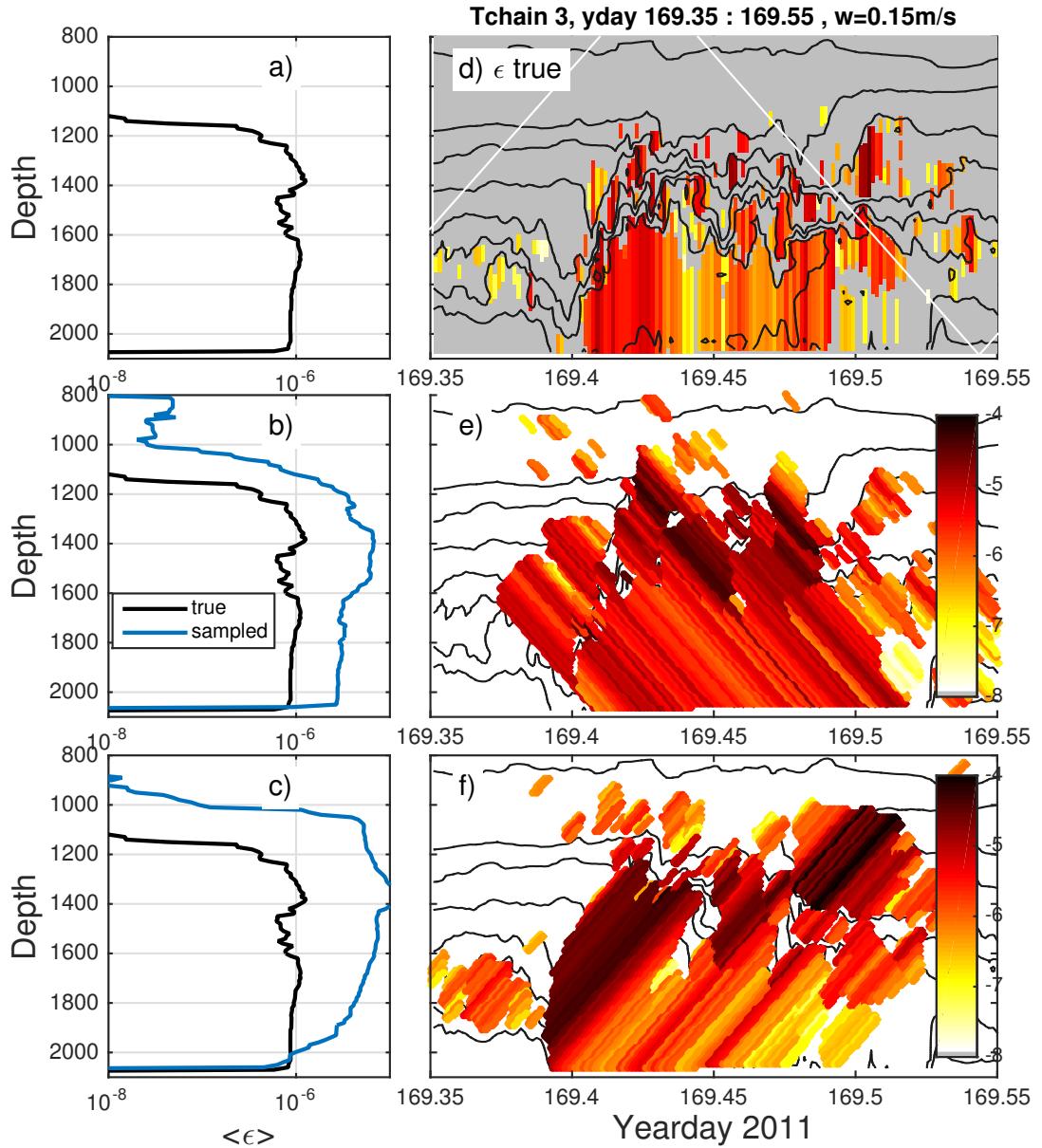


Figure 29: (top) True Tchain data. (middle) Epsilon along downward sampling paths. (bottom) Epsilon along upward sampling paths.

11.8 4 Mar 2015

I still don't have a good sense of what specifically causes the bias in epsilon at Tchains 3 and 4. I know that if I remove the 'false' overturns (where there wasn't an overturn in true profile but is in real), the bias mostly disappears. I will try looking at this for shorter sections to see why this results in the bias.

Continuing from yesterday. Figured out bootstrapping with `boot_v5.m` in mixingsoftware. Wrote `bootstrap_profile.m` to do this for each depth. Did bootstrapping for mean epsilon profile at Tchain3, the resampled mean profile is well outside the 95% limits for the true data.

At Ttide T2, where the timeseries is only a few days, the separation between the resampled mean and the true confidence limits is smaller. When I do bootstrap conf lims for true and resampled, they overlap for most of the cases. So I don't think I can really say whether there is a bias here. A longer timeseries would help.

11.9 3 Mar 2015

Need to compute bootstrap confidence limits for mean dissipation profiles (true). Are the resampled profiles in or outside these limits? If timeseries is short (T2?), conf. limits might be too big to say if there is a bias or not...

11.10 2 Mar 2015

Doing analysis for JN's Ttide mooring 'T2'. See details in section above. Noticing that gridding the data makes a sizable difference in mean epsilon profiles. Seems to be mostly the depth gridding. Another complication...

11.11 27 Feb 2015

* shortsectionsresamp... plot timeseries

Decided to look at Tchains 1 and 2 again. I had assumed they didn't capture dissipation well because overturns were smaller there. But maybe the same bias just doesn't happen there? There doesn't seem to be any bias at Tchain1, but it's possible that overturns there are smaller and not well resolved to begin with...

11.12 25 Feb 2015

I did resampling for the complete timeseries (two more spring tides), results and biases seem to hold. If anything, bias during 3rd spring might be larger.

I extended the resampling at Tchains 3 and 4 to cover two more spring tides. Results are similar.

I think I really need to do the analysis at some other locations. Might be dependent on the location and turbulence regime etc..

11.13 24 Feb 2015

Summary of LES overturns findings so far (see LES section of notes for details).

- Epsilon from overturns is about xx times larger? than directly computed epsilon. Masoud found something similar also.
- Masoud assumed L_t equals L_o ; should ask him if he compared the two.
- Resampled overturns don't appear to have much of a bias at any speed. This differs from what I found at the T-chains.
- N2 is more on top of ridge, maybe it is different at that location?
- Still want to do comparison at Tchain locations also to see if results are similar.

11.14 19 Feb 2015

Started looking at the LES data at N2 from Sutanu and Masoud. Think i've figured out the scaling etc. and did a first pass at computing overturns. The model data is potential density (not temp and salinity) so I made a modified overturns code to use that as input. I also needed to make a modified function (from `sw_bfrq.m`) to calculate buoyancy frequency from potential density instead of t and s (`Nsq_pden.m`). Nsq computed via the two differes a little bit, I think because of the way the averaging is done. But for my purposes here it shouldn't matter as long as I use the same method.

* working on `Resample_LES_OT.m`

11.15 18 Feb 2015

It seems like if I look at histograms of all epsilon, true and sampled aren't very different. But the profiles of average or integrated epsilon differ? Try looking at one depth?

What happens to averages if I replace 1e-11 with zeros? is it sensitive to choice of this value?

11.16 16 Feb 2015

Need to be careful about looking at histograms/scatter plots of thorpe scales and overturn sizes etc.. Each depth value in an overturn is given the same value, so larger overturns might be weighted more than smaller ones.

Got some LES model data from Masoud/Sutana at N2 location and started looking at it: `PlotLES_N2.m`. Need to figure out how to scale potential density. Also will need to adapt overturns code to input potential density instead of t and s.

I have modified overturns code to return displacements, N2, etc. so I can do more diagnostics. Plan to compare d, N2, dT/dz etc. to try to learn more about what is causing bias in epsilon between true and resampled.

Modified the resampling a bit to use the exact number of shifts to cover 1 profiling period instead of using 100 for each. This number is less than 100 for all speeds except 0.15m/s, for which it is 166 so might have been undersampled. I compared mean profiles with the different numbers and the differences are small (much less than the bias compared to true).

11.17 13 Feb 2015

We know that the resampled epsilon (averaged) is biased high, but why? Compare true and resampled temp, N2, dTdz etc. to see if we can see what is the dominant cause.

I had thought that L_{ot} in the overturns code was the thorpe displacements, but it is actually the patch size (the vertical size of each overturning region). Because this value is assigned to each point within each region, it might be misleading to look at statistics of it, since larger regions will be weighted more heavily?

I think I need to spend a little time working on the overturns code today to (1) make sure I completely understand what is doing (2) make it more transparent and well-documented (3) return additional variables and make summary plots, and be able to vary parameters more easily.

I want code to:

- Return re-ordering displacements
- have a parameters structure
- Generic plotting routines for profiles and individual overturns
- return sorted profiles and N (needed for plotting above..)
- Be readable and understandable by new users
- Return list of patch sizes and Thorpe scales (one for each overturning region, not one for each point) for statistics.

11.18 11 Feb 2015

I decided I need to go through the overturns code to clean it up and better understand everything it is doing. I think that Lot is actually the patch size, not the individual displacements as I was thinking... Check with JN if the code I have is the best place to start.

Think I need to summarize what i've found so far and figure out the big picture etc..

Big Picture:

- Many Thorpe scale estimates of epsilon from moored profilers and CTD profiles.
- All implicitly make assumption of instantaneous vertical profiles, when really they take some finite time.
- What is the effect of this assumption on the estimated Thorpe scales and dissipation?
- Can we diagnose what causes these effects and provide a useful way to apply the results to other locations, in order to quantify the uncertainty in Thorpe scale estimates from profiling instruments?

Results:

- Epsilon (average profiles and depth-integrated time series) computed from resampled Thorpe scales shows a positive bias (averaged over different phases) compared to the true data.
- The bias is inversely proportional to the sampling speed.
- For data points where there is an overturn in both the real and sampled data, the distribution of the difference in L looks nearly symmetric (i.e. we are not just always measuring bigger overturns).
- There are many ‘false’ overturns, where there was not an overturn in the true profile but there is in the resampled profile. There about half as many cases of the opposite, where there is an overturn in the true profile but not in the resampled profile.
- The bias mostly goes away if I remove these ‘false overturns’, suggesting they are the main cause of the bias.

Remaining Questions and Analysis

- I still don't completely understand what exactly is causing the bias. It seems to be largely due to the 'false' or mis-assigned overturns.
- Can the errors be related to some measure of the background flow (i.e. vertical or horizontal velocity)?
- Is vertical or horizontal advection important? (this would depend on the specific flow at the mooring site, and the horizontal gradients).
- Can the results here be generalized to other locations with different forcing?

11.19 10 Feb 2015

Trying to plot the contribution to the mean epsilon as a function of the overturn size, to see if the mean epsilon is dominated by a few large overturns etc.. What I did is sort overturns in increasing order, then plot the cumulative sum of epsilon using those indices (T3 in Figure 30). In the true data, the contribution from overturns up to about 500m looks pretty linear and accounts for about 65% of the total. Above that, it increases more sharply as bigger overturns contribute more. The resampled (25cm/s) looks similar up to about 500m, but then is shifter to larger overturns above that. At higher sampling speed, the line approaches the true value. Note however the results at Tchain 4 look different (Fig 31). See `PlotEpsContribVsL.m`

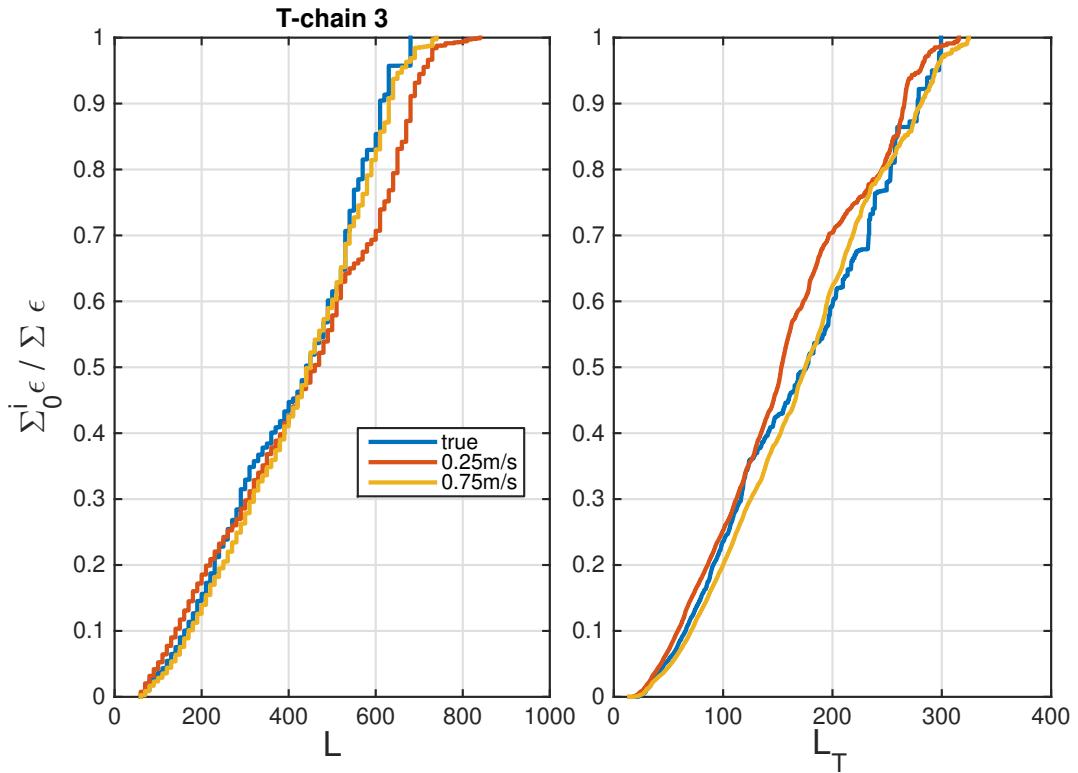
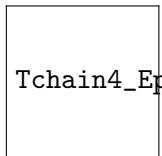


Figure 30: Cumulative sum of epsilon, sorted by increasing overturn size, versus the overturn size.



Tchain4_EpsContribVsLot.pdf

Figure 31: Cumulative sum of epsilon, sorted by increasing overturn size, versus the overturn size.

11.20 5 Feb 2015

Today I want to examine the relationship between the error or bias and the magnitude of the true dissipation etc.. Is there a linear, consistent relationship? If so, do we understand why? Do we expect it to hold for other locations (if so, could use it to estimate biases). Note that if the error is (linearly) proportional to the true value, this implies that the percent error is constant.

So why is the scaling different for positive and negative biases? Just because the negative bias can't be more than 100 percent?

Working in `PlotErrorVsTrueEps.m` again. If I fit a straight line to the log-log data (abs of all data), I get that the error is proportional to $\epsilon_o^{0.6-0.7}$. At slower speeds there seem to be more positive biases than negative, and at higher speeds there are about equal numbers of positive and negative biases. When I do the fit separately for positive and negative biases, I get different exponents. The exponent for the negative biases are close to 1.

I always get mixed up and forget how to plot a straight-line fit to a log-log plot, so i'm writing it here. I use polyfit to fit a straight line to the log of both quantities.

$$\log(Y) = m \log(X) + b \quad (3)$$

$$10^{\log(Y)} = 10^{[m \log(X) + b]} \quad (4)$$

$$Y = 10^{m \log(X)} 10^b \quad (5)$$

$$Y = 10^{\log(X^m)} 10^b \quad (6)$$

$$Y = X^m 10^b \quad (7)$$

$$Y = X^m 10^b \quad (8)$$

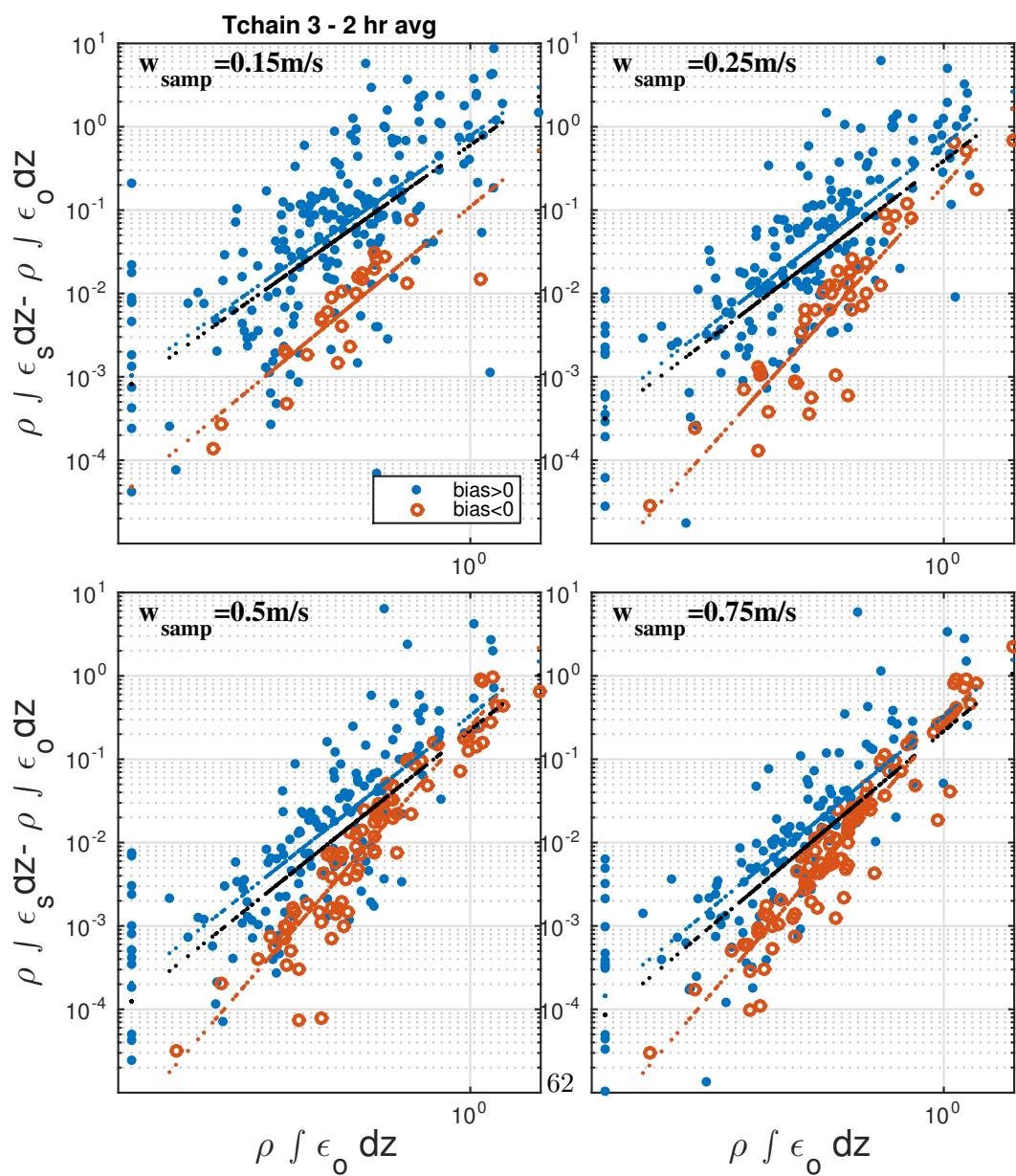


Figure 32: Plots of the error in depth-integrated epsilon (averaged over 2 hours) versus the magnitude of the true epsilon for resampling at 4 different speeds at Tchain3. Positive and negative errors are shown in different colors. Lines show best-fit line for each.

11.21 2 Feb 2015

I'm working in `EvalFalseOverturns.m` some more, doing the same analysis for depth-integrated epsilon instead of average profiles. The results are essentially the same.

I was wondering if somehow my epsilon averages were being screwed up by NaNs where there were no overturns (i.e. having a nan instead of a zero would change the average). There are Nans in L, but any NaNs in epsilon are given a value of 1e-11 so I don't think this is an issue.

So why are there more false positive overturns than false negative? If it is just that the MP samples turbulence at a different time, it seems like it would be equally likely to sample no turbulence where there actually is...

Are there just more points with overturns than without, so more possibilities for false positives? In the true data, the number of stable points (no overturns) is much larger than the number of points with overturns. About 93 percent of points have no overturns. So there are many more stable points that could become unstable during sampling, than unstable points that could become unstable.

If this is right, however, it means that we probably can't correct the sampled data after the fact since we have no way of knowing which overturns are false?... Maybe looking at persistence (in time) on successive profiles might help?

11.22 30 Jan 2015

Are there more false overturns than false non-overturns (an overturn in the real profile becomes stable in the resampled profile)? I would expect false non-overturns to reduce epsilon, but this doesn't happen so there must be more false overturns? There seem to be about twice as many false overturns as false non-overturns.

Plot bias vs epsilon magnitude for different speeds on same plot. `PlotErrorVsTrueEps.m`.

Why would the error or bias scale with the true magnitude? Partly because epsilon goes as the square of L? In terms of false overturns, time periods where epsilon is larger are more turbulent, and profiler is more likely to sample overturns during part of the profile?

Still don't understand why difference in L vs Ltrue has a negative trend (`ScatterErrorsVsTrue.m`)..

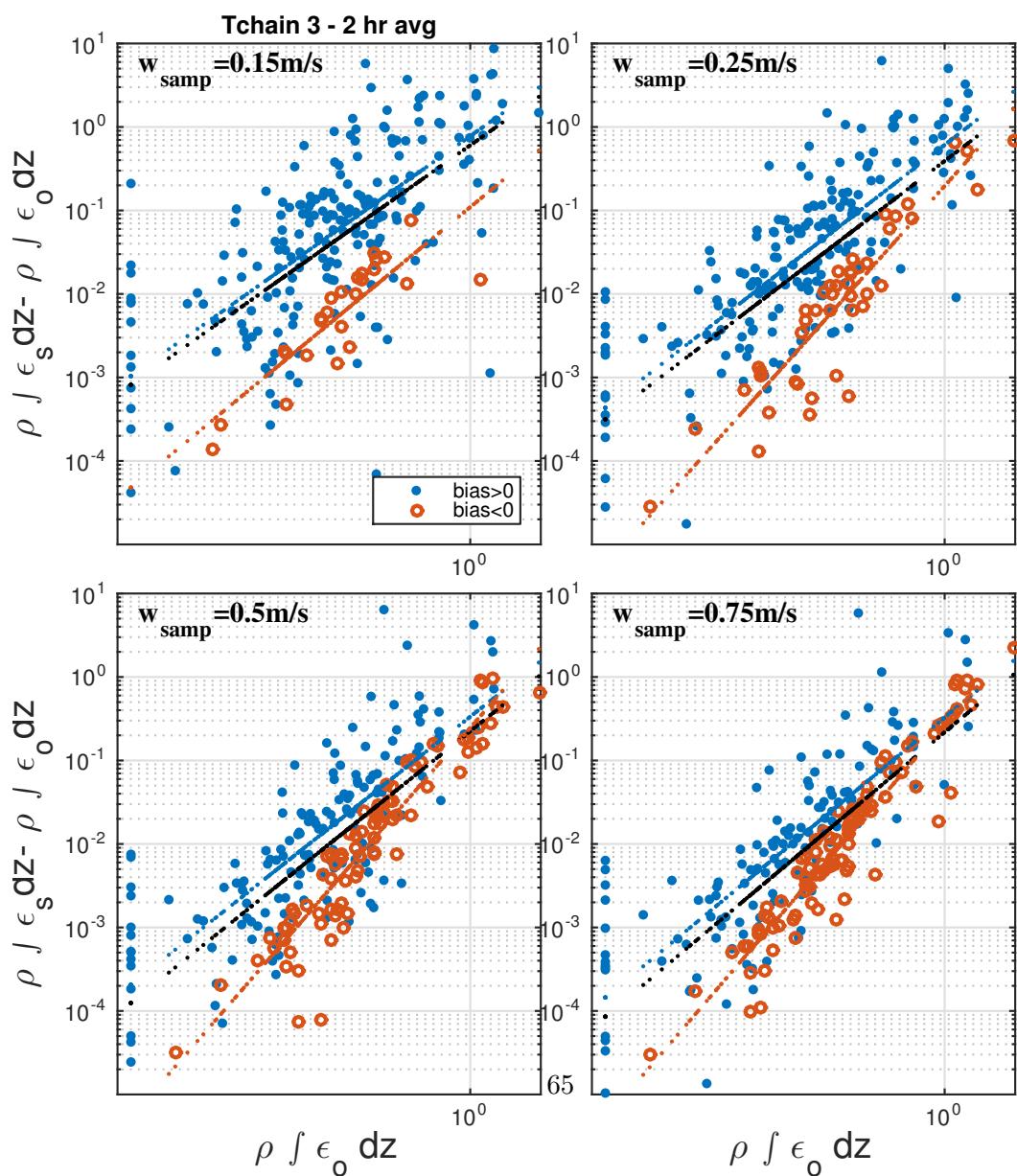


Figure 33: Plots of the error in depth-integrated epsilon (averaged over 2 hours) versus the magnitude of the true epsilon for resampling at 4 different speeds at Tchain3. Positive and negative errors are shown in different colors. Dashed line is slope 1.

Tchain4_4\$ speeds_IntEps_ErrorvsTrue.pdf

Figure 34: Plots of the error in depth-integrated epsilon (averaged over 2 hours) versus the magnitude of the true epsilon for resampling at 4 different speeds at Tchain4. Positive and negative errors are shown in different colors. Dashed line is slope 1.

11.23 29 Jan 2015

I want to see how much of an effect false overturns (overturns in the resampled data where there were none in the true profiles) have on the mean profiles of epsilon etc.. How many false overturns are there (what percent of total number of points)? How big are they (do they have the same distribution?)? I wrote a new section of code in `EvalFalseOverturns.m` to examine these questions for the entire ensemble of phases.

- There are many instances of false overturns (overturns in resampled profiles where there were none in true profiles).
- The distribution of overturn sizes for false positives looks similar to the total distribution (Figure 35).
- Figure 36 shows time-mean profiles (entire ensemble) of epsilon at Tchain 3 for the real, resampled, and resampled w/o false overturns data. This shows that a lot of the increased epsilon in the resampled data is due to false overturns introduced.
- Figure 37 shows an example of how false overturns might be introduced. The true profile (dashed line) is mostly stable, but the resampling samples a period of turbulence that occurs at a later time (but is assigned to the time at the midpoint of the profile).

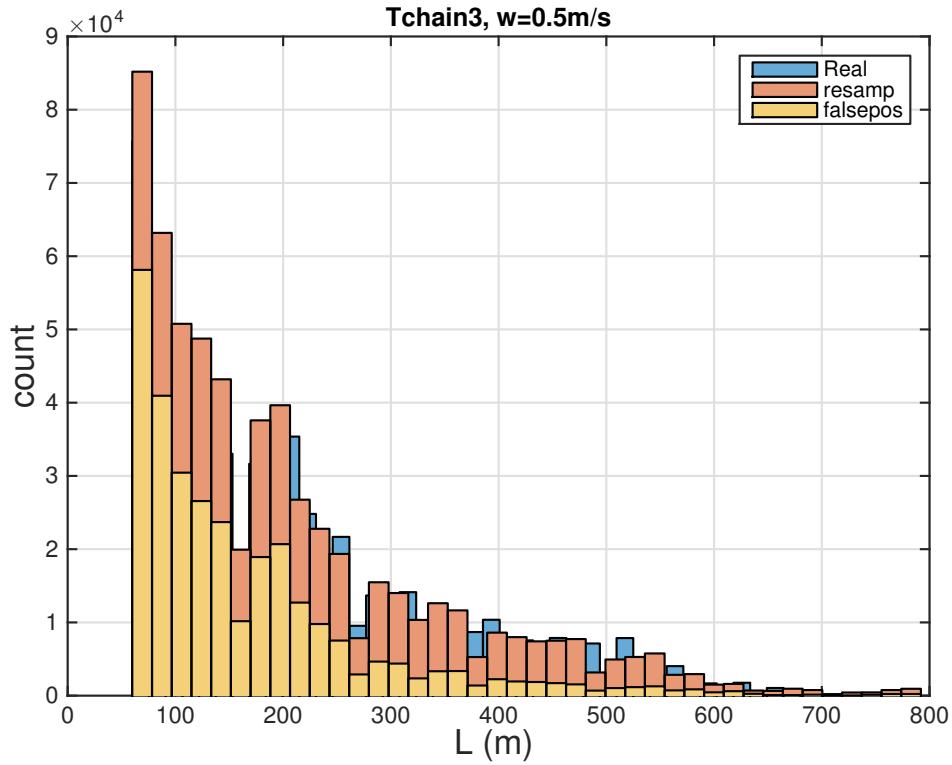


Figure 35: Histogram of overturn sizes for ensemble of resampled data at Tchain 3. Three sets of data are plotted: (1) Real overturns (true overturns in Tchain data, but only at times corresponding to resampled profiles). (2) All overturns in resampled data. (3) False positive overturns: overturns in resampled data where true profile was stable.

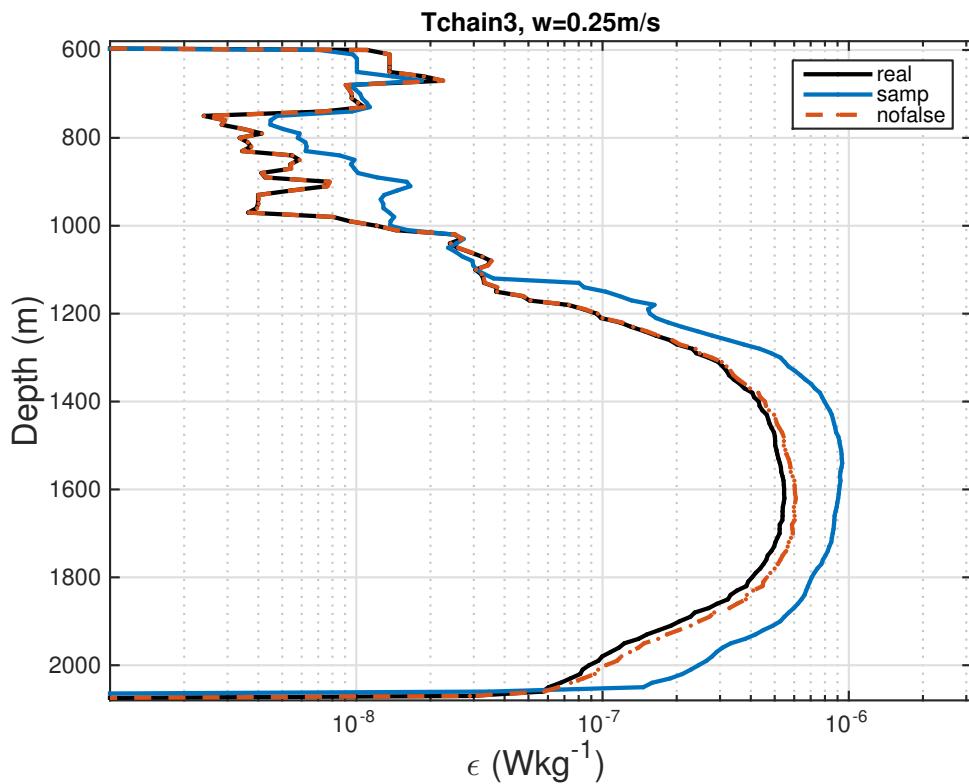


Figure 36: Time-mean profiles of epsilon from resampling at Tchain 3. Three sets of data are plotted: (1) ‘Real’ (using Tchain data, but only at times corresponding to resampled profiles). (2) All resampled profiles. (3) Resampled, but without false positive overturns (overturns in resampled data where true profile was stable).

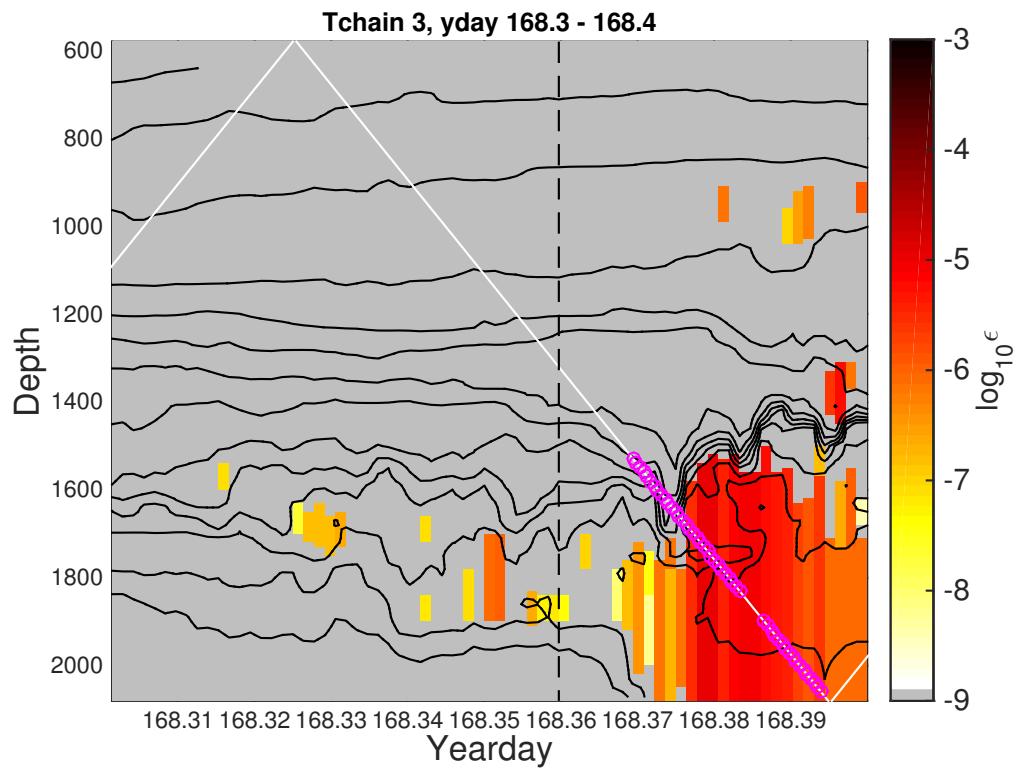


Figure 37: Example of how false overturns could be introduced.

11.24 28 Jan 2015

Wrote a code (`CombineResampTestCases.m`) to put entire ensemble in one structure, along with the real profiles at matching times, so I don't have to write a loop for every analysis.

Found and finally fixed a problem finding and plotting false overturns introduced in the resampled data. I was using `[R,C]=find(X)` and then `X(R,C)` to plot, but this doesn't work like I thought it did. Just using single index now.

11.25 27 Jan 2015

I wanted to check the effect of using different minimum overturn sizes for the T-chains. The T-chain data is gridded to 10m, but the vertical spacing of the sensors was actually around 50m. According to Nyquist theorem, we should only be able to resolve overturns larger than 100m (with no noise). I thought the dissipation was probably dominated by the larger overturns, but wanted to check : `CompareOverturnsDiffminOT.m`. At T-chain 3 the time-mean profile of epsilon is the same for minOT of 50m. For minOT of 100m, epsilon is smaller in the upper portion but the same in the region of largest dissipation below 1200m. At minOT of 250m, it has the same profile below 1200 but is smaller in magnitude. Results are similar at T-chain 4. This confirms that epsilon is dominated by larger overturns, except in the upper region.

Up to now I had assumed that the resampled overturn sizes were being symmetrically under/over estimated, leading to the increased average epsilon. Histograms of the difference in L are approximately normal and symmetric. However the story may be more complicated than this. I made scatter plots of the difference in L versus the true L (see `ScatterErrorsVsTrue.m`), and there seems to be a negative trend. At lower true L, the differences are mostly positive (resampled L is larger than true L), while at larger true L, the differences are mostly negative. I would have expected the magnitude of the difference to depend on the true size, but I don't know why the sign would change. And smaller sampled L should make epsilon smaller, but the average epsilon is bigger.

I'm contouring epsilon and isotherms and plotting where false overturns are introduced in the resampled data, as well as where the resampled overturns are smaller or larger than the true size (`ExamineShortSectionsResamp.m`).