How to compute Trinidad glider anomaly plots:

How to compute glider anomaly plots.

1. Data from the gliders need to be in a matlab format and averaged into vertical bins. This averaged and binned data needs to be in a matlab structure with the name bindata. It should have the form similar to bindata.time, bindata.t, bindata.s, etc. This is done in the program add\_bindataarray.m
2. The glider data in these arrays also needs to have had some QC applied as bad data will spoil the plots. The primary QC used was to set temperatures, salinities, and oxygens with values identically equal to zero to Not a Number (NaN). Also obvious spikes in the data when plotted as a time-series where identified and blanked. Salinity spikes of < 32 PSU where blanked unless it looked like a real event. Anomalously high values were also blanked. 157\_20141116 has quite a few bad salinities. 157\_20150917 has spikes high in oxygen. 157\_20161021 has more oxygen spikes high. Vary bad O2 profile in OSU646\_20190409. OSU\_646\_20220907 had a large section of bad oxygen data that has been blanked. Spikes in OSU685\_20230816 for salinity and oxygen.
3. Each new filename needs to be appended to a file name list (for test these are the files in x:\glider\_data\Trinidad\ trinidad\_glider.txt
4. Run [A,meanmap,avmap,avmaptm,avmapnomean,avmapnomeantm,meanmapsig,avmapsig,avmapsignomean]=allanncyc('Trin');

And save the output as anncycTr.mat. This file will be used in the next step.

This code computes the average, seasonal, mean and seasonal data for the a time frame that is hard coded into the data. The anomalies are computed against this data.

1. Run

[config,anommap]=climatologyCCS\_AnomOnly('Trin',{'t','s','fl','ox','cdom','bb'});

This code saves matfiles to be used later. Note this step takes quite a long time to run.

Five meter bins takes overnight but 10m bins takes about 3 hours presently.

Output is saved as map\_Tr\_AnomOnly.mat

This computes the anomaly for the glider data.

1. At this point wrapperavplots can be run to generate the average plots. This can be run before step 5 if desired. This code plots the mean and average plots in various flavors.
2. Run

[map,maptm,mapem,maptmem,mapsig,mapsigem,...

Anommap,anommaptm,anommapem,anommaptmem,...

anommapsig,anommapsigem]=allanommap(‘Tr’)

This code requires the output of both allanncyc and climatologyCCS\_AnomOnly

Save this output as mapTr.mat. This routine creates the maps for the plots used by the code below.

1. Run wrapperanomplots to generate the last of the plots (most of these are the anomaly plots and the Hovmoller plots).

The above generates a directory “Tr” with subdirectories of:

Anommap, anommapsig,avmap, avmapnomean,avmapsig, avmapsignomean, map, mapsig, meanmap, meanmapsig, seasons.

Many of the above directories have subdirectories of:

Time, timez, xtime, and xz

These directories in turn have subdirectories for each particular variable output from the glider.

Code requirements: Most of the code run is in the directory c:\glider\_climate\src. These programs do require some functions in the c:\glider\_climate\lib and c:\glider\_climate\seawater\_ver3\_2. The lib directory contains some of Dan Rudnick’s specific function codes. The seawater\_ver3\_2 is the seawater library with its standard routines. The other requirement is to have the “processed” glider data in a directory where it can be loaded by the code along with the list of the glider filenames.

World Ocean Atlas 2013 data. The temperature and salinity data are dense enough to use. The monthly data were fitted using Dan’s constant + 3 harmonics. These worked fine. Note that the inshore end has a gap. The oxygen data is at a lower resolution and in the end was too sparse to fit with a constant + 3 harmonics. The results have anomalous spikes in the fitted data. This is due to there being no constraints near the year transition.