A new procedure for processing large, single ping ADCP data using tools in python is described in detail here.

The raw data is a .pd0 file 11 GB in size and containing about 8 million pings over five months.

The work is in the \python subdirectory of E:\data\Matanzas\V23857 on WSMMA11. The python module has been updated, is on github at <https://github.com/mmartini-usgs/ADCPy>

Time in these raw files is not consistent until time is repopulated.

1. Convert the raw binary pd0 data to netcdf raw “.cdf” format
   1. Convert to a netCDF file of continuous raw single ping data where in this case of two burst time intervals programmed in the ADCP, the time step is uneven
      * Script is convertall.py
      * Method implemented is adcpy.TRDIstuff.TRDIpd0tonetcdfconvert\_pd0\_to\_netcdf
      * Input file is Use data file '11121whV23857.pd0' (11 GB)
      * in the directory E:\data\Matanzas\V23857\raw, the rest of the work is in the directory E:\data\Matanzas\V23857\python
      * Output file is '11121whV.cdf' (18 GB)
      * Do\_part\_one = True, do\_part\_two = False
      * The result of this step has some NaNs at the end of the .cdf file
2. For the current velocity profiles,
   1. Compute the rotation from beam to earth coordinates for each profile.
      * Script is convertall.py
      * Method implemented is cdf2nc.doEPIC\_ADCPfile
      * Input file is '11121whV.cdf'
      * Output file is ‘11121whV.nc' (12 GB)
      * Do\_part\_one = False, do\_part\_two = True
      * The result of this step has no NaNs
   2. Reshape the rotated netcdf data in to a burst shaped netcdf file, separating out samples suitable for tidal frequencies. This will use some of the samples in the wave burst. These settings are for an instrument with dual sampling, ~35 min at 2 Hz from the top of the hour followed by ~1.5 min every 15 min.
      * Input file is ‘11121whV.nc'
      * Output file is ‘11121whVcurrents00.nc’
      * Script reshape\_nc\_4currents.py
      * Method implemented is reshapeEPIC.reshapeEPIC
      * Nfiles = 1
      * Sample\_rate = 2 Hz (instrument setting)
      * Burst length = 120 samples (instrument setting)
      * Burst interval = 900 seconds (15 min) (instrument setting)
      * vars2omit = { 'EPIC\_time','EPIC\_time2'}
      * The output from this stage has a time coordinate with two dimensions, and if there are gaps in the data, the missing data are all at the end of the burst. This does not work with xarray and other downstream python methods
      * There may be NaNs in this output in time, if there are missing bursts
   3. Distribute the samples in each burst according to each sample’s time stamp, and reduce time to one dimension
      * Input file is ‘11121whVcurrents00.nc’
      * Output file is ‘11121whVcurrents00repo.nc’
      * Sample\_rate = 2 (this cannot be gleaned from the metadata)
   4. Average the bursts for mean current profiles at 15 min intervals
      * Using a Jupyter Notebook to do these final steps
3. For the velocity profiles to be used for wave statistics,
   1. Reshape the raw (beam coordinate) netcdf data in to a burst shaped netcdf file, separating out samples suitable for wave frequencies. This will use all of the samples in the wave burst an no subsequent smaller tidal bursts in each hour. These settings are for an instrument with dual sampling, ~20 min at 2 Hz from the top of the hour followed by ~1.5 min every 15 min.
      * Input file is ‘11121whV.cdf'
      * Output file is ‘11121whVwaves\*.nc’
      * Script reshape\_nc\_4waves.py
      * Method implemented is reshapeEPIC.reshapeEPIC
      * Nfiles = 1
      * Sample\_rate = 2 Hz (instrument setting)
      * Burst length = 4096 samples (instrument setting)
      * Burst interval = 3600 seconds (hourly) (instrument setting)
      * vars2omit = {'HdgSTD','PtchSTD','RollSTD','S','xmitc','Ambient\_Temp','Pressure+','Pressure-', 'Attitude\_Temp', 'EWD1','EWD2','EWD3','EWD4', 'PressVar', 'EPIC\_time','EPIC\_time2'}
      * The output from this stage has a time coordinate with two dimensions, and if there are gaps in the data, the missing data are all at the end of the burst. This does not work with xarray and other downstream python methods
      * There may be NaNs in this output in time, if there are missing bursts
      * The time variable appears artificially short because of the reshape
   2. Distribute the samples in each burst according to each sample’s time stamp, and reduce time to one dimension
      * Input file is ‘11121whVwaves\*.nc’
      * Output file is ‘11121whVwaves\*repo.nc’
      * Sample\_rate = 2 (this cannot be gleaned from the metadata)
   3. Compute wave statistics
      * For gappy data, use the Lomb method with TBD variant of DIWASP
      * For non-gappy data use DIWASP