pycurrents\_ADCP\_processing v1.0.1 (Hana Hourston *et al.*, 2023)

pycurrents\_ADCP\_processing provides multi-level processing for raw ADCP data using the UHDAS pycurrents package. This software includes L0, L1 and L2 processing levels, however there is no visualization functionality. This software works with Teledyne RDI Workhorse, Sentinel V and Broadband instruments. The software was last updated on 9/5/2024.

* [GitHub Repository](https://github.com/IOS-OSD-DPG/pycurrents_ADCP_processing?tab=readme-ov-file)
* [NetCDF: Introduction and Overview](https://docs.unidata.ucar.edu/netcdf-c/current/index.html)

Notes from the authors:

**Note: L2 processing routine is currently out-of-date**

For performing "level 0" (L0), "level 1" (L1), and "level 2" (L2) processing on raw moored ADCP data in Python using the UHDAS pycurrents package and plotting the output netCDF data. Teledyne RDI Workhorse, Sentinel V, and Broadband instruments are supported.

L0 processing does not include any processing. Raw ADCP data is combined with metadata from a csv file and exported in netCDF format.

L1 processing contains minimal processing. Raw ADCP data is also combined with metadata from a csv file and exported in netCDF format. The difference from L0 is that L1 processing comprises:

* Corrections for magnetic declination
* Calculation of sea surface height from pressure values and latitude
* Rotation into enu coordinates if this is not already the coordinate system of the dataset
* Removing leading and trailing ensembles from before and after deployment
* Flagging negative pressure values
* Optional: Splitting the dataset into segments if there are pressure changes due to mooring strikes. The user must provide the date-times or ensembles at which to split the dataset

L2 processing contains:

* Flagging data in bins where calculated pressure is negative
* Flagging data by backscatter increases in upward-facing ADCPs
* Flagging data below the depth of the sea floor in downward-facing ADCPs
* Optional: Calculation of pressure data from CTD pressure data from the same deployment (if the ADCP was missing a pressure sensor)

If the time data in the raw file are garbled, then use *generate\_time\_range.py* to create a csv file containing regularly-spaced time data.

*ADCP\_IOS\_header\_file.py* produces an IOS Shell header file for each netCDF file that makes the netCDF file searchable on the [IOS Water Properties website](https://www.waterproperties.ca/).

*plot\_westcoast\_nc\_LX.py* contains code for the following types of plots:

* pressure data time series (check potential ADCP depth change)
* North/East (LCNSAP01/LCEWAP01) and along-/cross-shore velocity profile time series
* Low-pass filtered versions of the above plots (Godin or X-hour rolling mean, e.g., 30 hour)
* Eastward (LCEWAP01) bin 1 time series (check bad data close to ADCP)
* Diagnostic plots for mean backscatter, mean velocity, and principal axis (orientation)
* Feather plots for select bins: current vectors plotted over time
* Rotary spectra for selected bins
* Depth profile of rotary spectrum
* Depth profile of tidal ellipses
* Single bin North/East velocity plots

Installation:

1. Before creating a virtual environment for the package, create a folder for the virtual environment and enter the folder in terminal, e.g. "adcp"

Here is the location: [\\wsl.localhost\Ubuntu\home\ethan\Honours\_Project\pycurrents\_adcp](file:///\\wsl.localhost\Ubuntu\home\ethan\Honours_Project\pycurrents_adcp)

I don’t think that will work, I am trying to do it here: C:\Users\ethan\anaconda3\envs\pycurrents\_adcp

1. Create a virtual environment called "adcp37" with Python version 3.7:  
   conda create -n adcp37 python=3.7

I opened the pycurrents\_adcp folder in the Anaconda Prompt terminal and created the virtual environment.

1. Activate the virtual environment:  
   conda activate adcp37
2. Add the conda-forge to your channel:  
   conda config --add channels conda-forge   
   conda config --set channel\_priorit strict
3. Install required packages:  
   conda install numpy scipy pip pandas netCDF4 xarray gsw matplotlib=3.5 shapely  
   pip install datetime ruamel.yaml

Ran into this error:

Channels:

- conda-forge

- defaults

Platform: win-64

Collecting package metadata (repodata.json): done

Solving environment: failed

PackagesNotFoundError: The following packages are not available from current channels:

- matplotplib=3.5\*

- install

- datetime

Current channels:

- https://conda.anaconda.org/conda-forge

- defaults

- https://repo.anaconda.com/pkgs/main

- https://repo.anaconda.com/pkgs/r

- https://repo.anaconda.com/pkgs/msys2

To search for alternate channels that may provide the conda package you're

looking for, navigate to

https://anaconda.org

and use the search bar at the top of the page.

Trying to investigate what is going on using the anaconda navigator in the adcp37 environment. I then checked and found that I made a typo: matplotplib=3.5 should be matplotlib=3.5.

There were also invalid usage issues: pip install datetime ruamel.yaml is not valid syntax for a conda install command. Conda does not support pip install directly in the same line. Also, datetime is part of the Python standard library, not a separate installable package.

This is the correct install command: conda install numpy scipy pandas netCDF4 xarray gsw matplotlib=3.5 shapely ruamel.yaml

1. Install ttide\_py from github  
   git clone https://github.com/moflaher/ttide\_py  
   Navigate to the newly created ttide\_py folder, then  
   python setup.py install

I cloned the repo successfully but ran into a problem navigating to the newly created folder. I found this information from a YouTube video: [Using the Anaconda Prompt to Navigate Across Folders](https://www.youtube.com/watch?v=yhnibVLlhMk). Here is the GitHub: [satellite-image-analysis/notebooks/1.03-anaconda\_and\_jupyter.ipynb at main · edwardoughton/satellite-image-analysis · GitHub](https://github.com/edwardoughton/satellite-image-analysis/blob/main/notebooks/1.03-anaconda_and_jupyter.ipynb)

1. Clone pycurrents with Mercurial:  
   hg clone --verbose <http://currents.soest.hawaii.edu/hg/pycurrents>

Get this error: 'hg' is not recognized as an internal or external command, operable program or batch file. I first need to install Mercurial via Conda. In the active adcp37 environment, run: conda install -c conda-forge mercurial.

Then retry the clone: hg clone --verbose http://currents.soest.hawaii.edu/hg/pycurrents

1. Install pycurrents:  
   pip install -e ./pycurrents

Ran into this problem:

Obtaining file:///C:/Users/ethan/anaconda3/envs/pycurrents\_adcp/ttide\_py/pycurrents

Preparing metadata (setup.py) ... error

error: subprocess-exited-with-error

× python setup.py egg\_info did not run successfully.

│ exit code: 1

╰─> [6 lines of output]

Traceback (most recent call last):

File "<string>", line 36, in <module>

File "<pip-setuptools-caller>", line 34, in <module>

File "C:\Users\ethan\anaconda3\envs\pycurrents\_adcp\ttide\_py\pycurrents\setup.py", line 7, in <module>

import Cython

ModuleNotFoundError: No module named 'Cython'

[end of output]

note: This error originates from a subprocess, and is likely not a problem with pip.

error: metadata-generation-failed

× Encountered error while generating package metadata.

╰─> See above for output.

note: This is an issue with the package mentioned above, not pip.

hint: See above for details.

The problem now is that there is no Cython module installed. Install it with: conda install cython

Now try pip install -e ./pycurrents again

Got this error:

(adcp37) C:\Users\ethan\anaconda3\envs\pycurrents\_adcp\ttide\_py> pip install -e ./pycurrents

Obtaining file:///C:/Users/ethan/anaconda3/envs/pycurrents\_adcp/ttide\_py/pycurrents

Preparing metadata (setup.py) ... error

error: subprocess-exited-with-error

× python setup.py egg\_info did not run successfully.

│ exit code: 1

╰─> [8 lines of output]

Traceback (most recent call last):

File "<string>", line 36, in <module>

File "<pip-setuptools-caller>", line 34, in <module>

File "C:\Users\ethan\anaconda3\envs\pycurrents\_adcp\ttide\_py\pycurrents\setup.py", line 23, in <module>

from setup\_helper import (

File "C:\Users\ethan\anaconda3\envs\pycurrents\_adcp\ttide\_py\pycurrents\setup\_helper.py", line 1, in <module>

pycurrents/setup\_helper.py

NameError: name 'pycurrents' is not defined

[end of output]

note: This error originates from a subprocess, and is likely not a problem with pip.

error: metadata-generation-failed

× Encountered error while generating package metadata.

╰─> See above for output.

note: This is an issue with the package mentioned above, not pip.

hint: See above for details.

The problem is where I am running the install from. The setup.py is inside C:\Users\ethan\anaconda3\envs\pycurrents\_adcp\ttide\_py\pycurrents\

To fix this problem, cd into the pycurrents folder and run the installer:

cd pycurrents

pip install -e .

That did not work, and the file structure is very confusing. I think something has gone badly wrong.

The folder nesting is incorrect.

ttide\_py/

└── pycurrents/ ← ✅ contains setup.py (this is the REAL repo root)

└── pycurrents/ ← 🔁 duplicated inner package directory

└── setup\_helper.py (correct file, but wrong location)

To fix this I cut everything from the inner pycurrents folder and pasted it (replace existing files) into the parent pycurrents folder. I then deleted the inner pycurrents folder. Have to navigate out of inner pycurrents folder in the terminal to do so.

The you can pip install -e .

I rain into this error:

(adcp37) C:\Users\ethan\anaconda3\envs\pycurrents\_adcp\ttide\_py\pycurrents\pycurrents>pip install -e .

Obtaining file:///C:/Users/ethan/anaconda3/envs/pycurrents\_adcp/ttide\_py/pycurrents/pycurrents

ERROR: file:///C:/Users/ethan/anaconda3/envs/pycurrents\_adcp/ttide\_py/pycurrents/pycurrents does not appear to be a Python project: neither 'setup.py' nor 'pyproject.toml' found.

(adcp37) C:\Users\ethan\anaconda3\envs\pycurrents\_adcp\ttide\_py\pycurrents\pycurrents>cd ../

(adcp37) C:\Users\ethan\anaconda3\envs\pycurrents\_adcp\ttide\_py\pycurrents>pip install -e.

Obtaining file:///C:/Users/ethan/anaconda3/envs/pycurrents\_adcp/ttide\_py/pycurrents

Preparing metadata (setup.py) ... error

error: subprocess-exited-with-error

× python setup.py egg\_info did not run successfully.

│ exit code: 4294967295

╰─> [1 lines of output]

Windows is not supported

[end of output]

note: This error originates from a subprocess, and is likely not a problem with pip.

error: metadata-generation-failed

× Encountered error while generating package metadata.

╰─> See above for output.

note: This is an issue with the package mentioned above, not pip.

hint: See above for details.

Need to redo all of this in Ubuntu.

1. Clone this pycurrents\_ADCP\_processing repository with git:  
   git clone https://github.com/IOS-OSD-DPG/pycurrents\_ADCP\_processing.git
2. cd to the pycurrents\_ADCP\_processing directory and run "setup.py":  
   python setup.py install

**Pre-requisites**

* Linux (or Unix-like) environment
* Python 3.7

**Trying on Ubuntu**

1. Install Miniconda

Download Miniconda installer

wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86\_64.sh

Run the installer

bash Miniconda3-latest-Linux-x86\_64.sh

Follow the prompts (yes to all) and then refresh the shell

source ~/.bashrc

1. Create a working directory and a virtual environment named adcp37 with Python 3.7:

mkdir ~/adcp

cd ~/adcp

conda create -n adcp37 python=3.7

conda activate adcp37

1. Configure Conda channels and install dependencies

conda config --add channels conda-forge

conda config --set channel\_priority strict

conda install numpy scipy pip pandas netCDF4 xarray gsw matplotlib=3.5 shapely

pip install ruamel.yaml

1. Install ttide\_py

git clone https://github.com/moflaher/ttide\_py

cd ttide\_py

python setup.py install

cd ..

1. Install pycurrents

Ensure that Mercurial and Cython are installed

sudo apt install mercurial

sudo apt install cython

Running into problems installing cython:

(adcp37) ethan@bigtop:~/adcp$ sudo apt install cython

Reading package lists... Done

Building dependency tree... Done

Reading state information... Done

Package cython is not available, but is referred to by another package.

This may mean that the package is missing, has been obsoleted, or

is only available from another source

However the following packages replace it:

cython3

E: Package 'cython' has no installation candidate

(adcp37) ethan@bigtop:~/adcp$ sudo apt install cython3

Reading package lists... Done

Building dependency tree... Done

Reading state information... Done

The following additional packages will be installed:

libexpat1-dev libpython3-dev libpython3.12-dev python3-dev python3.12-dev

Suggested packages:

cython-doc

The following NEW packages will be installed:

cython3 libexpat1-dev libpython3-dev libpython3.12-dev python3-dev

python3.12-dev

0 upgraded, 6 newly installed, 0 to remove and 5 not upgraded.

Need to get 11.2 MB of archives.

After this operation, 49.0 MB of additional disk space will be used.

Do you want to continue? [Y/n] y

Get:1 http://archive.ubuntu.com/ubuntu noble/universe amd64 cython3 amd64 3.0.8-1ubuntu3 [3961 kB]

Ign:2 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 libexpat1-dev amd64 2.6.1-2ubuntu0.2

Ign:3 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 libpython3.12-dev amd64 3.12.3-1ubuntu0.5

Get:4 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 libpython3-dev amd64 3.12.3-0ubuntu2 [10.3 kB]

Ign:5 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 python3.12-dev amd64 3.12.3-1ubuntu0.5

Get:6 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 python3-dev amd64 3.12.3-0ubuntu2 [26.7 kB]

Err:2 http://security.ubuntu.com/ubuntu noble-updates/main amd64 libexpat1-dev amd64 2.6.1-2ubuntu0.2

404 Not Found [IP: 91.189.91.81 80]

Err:3 http://security.ubuntu.com/ubuntu noble-updates/main amd64 libpython3.12-dev amd64 3.12.3-1ubuntu0.5

404 Not Found [IP: 91.189.91.81 80]

Err:5 http://security.ubuntu.com/ubuntu noble-updates/main amd64 python3.12-dev amd64 3.12.3-1ubuntu0.5

404 Not Found [IP: 91.189.91.81 80]

Fetched 3998 kB in 4s (990 kB/s)

E: Failed to fetch http://security.ubuntu.com/ubuntu/pool/main/e/expat/libexpat1-dev\_2.6.1-2ubuntu0.2\_amd64.deb 404 Not Found [IP: 91.189.91.81 80]

E: Failed to fetch http://security.ubuntu.com/ubuntu/pool/main/p/python3.12/libpython3.12-dev\_3.12.3-1ubuntu0.5\_amd64.deb 404 Not Found [IP: 91.189.91.81 80]

E: Failed to fetch http://security.ubuntu.com/ubuntu/pool/main/p/python3.12/python3.12-dev\_3.12.3-1ubuntu0.5\_amd64.deb 404 Not Found [IP: 91.189.91.81 80]

E: Unable to fetch some archives, maybe run apt-get update or try with --fix-missing?

I think I need to try a new source.

conda install cython

Clone and install:

hg clone --verbose http://currents.soest.hawaii.edu/hg/pycurrents

pip install -e ./pycurrents

1. Install pycurrents\_ADCP\_processing

git clone https://github.com/IOS-OSD-DPG/pycurrents\_ADCP\_processing.git

cd pycurrents\_ADCP\_processing

python setup.py install

It is installed successfully, but now I don’t know what to do. I also found this which might be useful:

**Resources**

A web app based off this package can be found on the [IOS Data Management Apps](https://dmapps.waterproperties.ca/en/) (DM Apps) page. Credits: Tom Roe.

**Usage:**

The file create\_nc.py is provided by the developers as a **worked example**, showing how to:

* Run the L0 and L1 processing steps
* Call the main ADCP scripts (like ADCP\_processing\_L0.py, add\_var2nc.py, etc.)
* Pass in correct arguments and paths
* Set up metadata

It's like a **template you can copy** and adapt for your own cruise or dataset.

To view the worked example From the adcp37 environment:

cd ~/adcp/pycurrents\_ADCP\_processing/ pycurrents\_ADCP\_processing

less create\_nc.py # text viewer and no editing

You can scroll using:

* ↓ or j: scroll down
* ↑ or k: scroll up
* space: scroll by page
* q: quit and go back to the shell

You can also just open it in a text editor like notepad ++

I think the best way to go is to install and run an IDE like Spyder inside WSL

To install Spyder in WSL:

conda activate adcp37

conda install spyder

Then run it:

spyder

Ran into this error:

(adcp37) ethan@bigtop:~$ spyder

Traceback (most recent call last):

File "/home/ethan/miniconda3/envs/adcp37/bin/spyder", line 11, in <module>

sys.exit(main())

File "/home/ethan/miniconda3/envs/adcp37/lib/python3.7/site-packages/spyder/app/start.py", line 233, in main

from spyder.app import mainwindow

File "/home/ethan/miniconda3/envs/adcp37/lib/python3.7/site-packages/spyder/app/mainwindow.py", line 57, in <module>

from qtpy import QtWebEngineWidgets # analysis:ignore

File "/home/ethan/miniconda3/envs/adcp37/lib/python3.7/site-packages/qtpy/QtWebEngineWidgets.py", line 28, in <module>

from PyQt5.QtWebEngineWidgets import (

ImportError: libpci.so.3: cannot open shared object file: No such file or directory

Did some trouble shooting to confirm that I am on WSL2, WSLg is working and that my system is capable of launching Spyder

The issue is the missing lib

conda deactivate

sudo apt update

sudo apt install libpci3

conda activate adcp37

spyder

This error appeared:   
(adcp37) ethan@bigtop:~$ spyder

qglx\_findConfig: Failed to finding matching FBConfig for QSurfaceFormat(version 2.0, options QFlags<QSurfaceFormat::FormatOption>(), depthBufferSize -1, redBufferSize 1, greenBufferSize 1, blueBufferSize 1, alphaBufferSize -1, stencilBufferSize -1, samples -1, swapBehavior QSurfaceFormat::SingleBuffer, swapInterval 1, colorSpace QSurfaceFormat::DefaultColorSpace, profile QSurfaceFormat::NoProfile)

qglx\_findConfig: Failed to finding matching FBConfig for QSurfaceFormat(version 2.0, options QFlags<QSurfaceFormat::FormatOption>(), depthBufferSize -1, redBufferSize 1, greenBufferSize 1, blueBufferSize 1, alphaBufferSize -1, stencilBufferSize -1, samples -1, swapBehavior QSurfaceFormat::SingleBuffer, swapInterval 1, colorSpace QSurfaceFormat::DefaultColorSpace, profile QSurfaceFormat::NoProfile)

Could not initialize GLX

Aborted (core dumped)

This seams like an issue with OpenGL – working with JupyterLab instead would be easier.

**Installing JupyterLab**

* 1. Open WSL terminal and activate adcp37 environment then install JL

conda activate adcp37

conda install jupyterlab

* 1. Launch JL

Navigate to the folder you want to work in (where the scripts are) (sets WD)

(adcp37) ethan@bigtop:~/adcp/pycurrents\_ADCP\_processing/pycurrents\_ADCP\_processing$

jupyter lab --no-browser --port=8888

I ran into this error:

Traceback (most recent call last):

File "/home/ethan/miniconda3/envs/adcp37/bin/jupyter-lab", line 6, in <module>

from jupyterlab.labapp import main

File "/home/ethan/miniconda3/envs/adcp37/lib/python3.7/site-packages/jupyterlab/\_\_init\_\_.py", line 7, in <module>

from .handlers.announcements import ( # noqa

File "/home/ethan/miniconda3/envs/adcp37/lib/python3.7/site-packages/jupyterlab/handlers/announcements.py", line 15, in <module>

from jupyterlab\_server.translation\_utils import translator

File "/home/ethan/miniconda3/envs/adcp37/lib/python3.7/site-packages/jupyterlab\_server/\_\_init\_\_.py", line 5, in <module>

from .app import LabServerApp

File "/home/ethan/miniconda3/envs/adcp37/lib/python3.7/site-packages/jupyterlab\_server/app.py", line 9, in <module>

from jupyter\_server.extension.application import ExtensionApp, ExtensionAppJinjaMixin

File "/home/ethan/miniconda3/envs/adcp37/lib/python3.7/site-packages/jupyter\_server/extension/application.py", line 12, in <module>

from jupyter\_server.serverapp import ServerApp

File "/home/ethan/miniconda3/envs/adcp37/lib/python3.7/site-packages/jupyter\_server/serverapp.py", line 64, in <module>

from nbformat.sign import NotebookNotary

File "/home/ethan/miniconda3/envs/adcp37/lib/python3.7/site-packages/nbformat/\_\_init\_\_.py", line 11, in <module>

from . import v1, v2, v3, v4

File "/home/ethan/miniconda3/envs/adcp37/lib/python3.7/site-packages/nbformat/v4/\_\_init\_\_.py", line 23, in <module>

from .convert import downgrade, upgrade

File "/home/ethan/miniconda3/envs/adcp37/lib/python3.7/site-packages/nbformat/v4/convert.py", line 11, in <module>

from nbformat import v3, validator

File "/home/ethan/miniconda3/envs/adcp37/lib/python3.7/site-packages/nbformat/validator.py", line 16, in <module>

from .json\_compat import ValidationError, \_validator\_for\_name, get\_current\_validator

File "/home/ethan/miniconda3/envs/adcp37/lib/python3.7/site-packages/nbformat/json\_compat.py", line 11, in <module>

import jsonschema

File "/home/ethan/miniconda3/envs/adcp37/lib/python3.7/site-packages/jsonschema/\_\_init\_\_.py", line 13, in <module>

from jsonschema.\_format import FormatChecker

File "/home/ethan/miniconda3/envs/adcp37/lib/python3.7/site-packages/jsonschema/\_format.py", line 411, in <module>

from webcolors import CSS21\_NAMES\_TO\_HEX

File "/home/ethan/miniconda3/envs/adcp37/lib/python3.7/site-packages/webcolors/\_\_init\_\_.py", line 12, in <module>

from .\_conversion import (

File "/home/ethan/miniconda3/envs/adcp37/lib/python3.7/site-packages/webcolors/\_conversion.py", line 49

if hex\_value := color\_map.get(name.lower()):

^

SyntaxError: invalid syntax

The issue is that this:

SyntaxError: invalid syntax

if hex\_value := color\_map.get(name.lower()):

^

Uses the walrus operator :=, which was only introduced in Python 3.8. The webcolors package (a dependency of jsonschema → nbformat → JupyterLab) is using syntax not supported in Python 3.7.

The best way to go forward is to uninstall JL and install an older version to avoid the walrus operator issue.

**Uninstall JL (in adcp37)**

pip uninstall jupyterlab

**Install JL 3.0.17**

pip install jupyterlab==3.0.17

**Downgrade webcolors**

pip install webcolors==1.11.1

**Launch JL again**

jupyter lab --no-browser --port=8888

**If it works, ignore warnings and scroll down to the URL’s**

Example: <http://localhost:8888/login?next=%2Flab%3Ftoken%3Dc95e5f8b5167db1bc54b26b689ab3c77d857c8063a96b760>

Paste this in a browser or follow the link. You need to log in using the token provided in the url.

This is not working, and I might need to find a workaround.

Set a fixed token:

jupyter lab --no-browser --port=8888 --NotebookApp.token='myfixedtoken123'

Not working at all

Alternative: Disable token and cookie authentication (not recommended for general use, only safe if on your local machine!)

jupyter lab --no-browser --port=8888 --NotebookApp.token='' --NotebookApp.disable\_check\_xsrf=True

**WORKING -** realised that I still had anaconda running on windows, where I had JL open earlier in the day. I restarted my pc (full shutdown) and went straight into the ubuntu terminal:

conda activate adcp37

cd adcp/pycurrents\_ADCP\_processing/pycurrents\_ADCP\_processing

jupyter lab --port 8888 --no-browser

Then select the URL that is second from last. It should open up in the browser that you have open and run as is.

I think there is a duplication error similar to the one I had way back in the windows installation. This makes the sample/example scripts do not work because the file system is wonky.

This is an image of the file system structure:

(adcp37) ethan@bigtop:~/adcp/pycurrents\_ADCP\_processing$ ls

LICENSE.txt dist sample\_results

README.md pycurrents\_ADCP\_processing setup.py

build pycurrents\_ADCP\_processing.egg-info templates

deprecated sample\_data

(adcp37) ethan@bigtop:~/adcp/pycurrents\_ADCP\_processing$ cd pycurrents\_ADCP\_processing

(adcp37) ethan@bigtop:~/adcp/pycurrents\_ADCP\_processing/pycurrents\_ADCP\_processing$ ls

ADCP\_IOS\_Header\_file.py create\_nc.py ios\_polygons.geojson

ADCP\_processing\_L0\_L1.py create\_nc\_l2.py plot\_westcoast\_nc\_LX.py

ADCP\_processing\_L2.py example\_plot\_westcoast.py utils.py

\_\_init\_\_.py generate\_time\_range.py

adcp\_var\_string\_attrs.yml igrf\_avg\_magnetic\_declination.R

(adcp37) ethan@bigtop:~/adcp/pycurrents\_ADCP\_processing/pycurrents\_ADCP\_processing$

[How to run Python scripts in JupyterLab](https://www.youtube.com/watch?v=Fw6VYgsbJ4o)

Usage from the author:

Sample usage of ADCP\_processing\_L1.py, ADCP\_processing\_L0.py, add\_var2nc.py and ADCP\_IOS\_Header\_file.py is laid out in create\_nc.py. An example of how to create uniform time data (for replacing invalid time data in a raw ADCP file) can be found in generate\_time\_range.py. Sample usage of the plotting functions in plot\_westcoast\_nc\_LX.py is given in the file example\_plot\_westcoast.py.

Open create\_nc.py in JL. Navigate to the blue plus on the top left (below File and Edit), then under other: start a new terminal session. This should open your terminal in the working directory. Make sure that the adcp37 environment is active. Enter python create\_nc.py in the terminal to run the script.

This is the error I run into:  
  
(base) ethan@bigtop:~/adcp/pycurrents\_ADCP\_processing/pycurrents\_ADCP\_processing$ conda activate adcp37

(adcp37) ethan@bigtop:~/adcp/pycurrents\_ADCP\_processing/pycurrents\_ADCP\_processing$ python create\_nc.py

Traceback (most recent call last):

File "create\_nc.py", line 24, in <module>

ncnames\_L0 = ADCP\_processing\_L0\_L1.nc\_create\_L0\_L1(in\_file=f, file\_meta=meta, dest\_dir=dest\_dir, level=0)

File "/home/ethan/miniconda3/envs/adcp37/lib/python3.7/site-packages/pycurrents\_ADCP\_processing-1.0.1-py3.7.egg/pycurrents\_ADCP\_processing/ADCP\_processing\_L0\_L1.py", line 1022, in nc\_create\_L0\_L1

meta\_dict = create\_meta\_dict(file\_meta, level)

File "/home/ethan/miniconda3/envs/adcp37/lib/python3.7/site-packages/pycurrents\_ADCP\_processing-1.0.1-py3.7.egg/pycurrents\_ADCP\_processing/ADCP\_processing\_L0\_L1.py", line 458, in create\_meta\_dict

with open(adcp\_meta) as csv\_file:

FileNotFoundError: [Errno 2] No such file or directory: './sample\_data/a1\_20050503\_20050504\_0221m\_metadata.csv'

The script is running from inside the pycurrents\_ADCP\_processing subfolder, while the sample\_data/ is located one level above that. The best way to fix this would be to run the script from the root: ~/adcp/pycurrents\_ADCP\_processing

cd ~/adcp/pycurrents\_ADCP\_processing

conda activate adcp37

python pycurrents\_ADCP\_processing/create\_nc.py

This retains the relative paths and should make usage less painful.

Ran into this error:

File "/home/ethan/miniconda3/envs/adcp37/lib/python3.7/site-packages/pycurrents\_ADCP\_processing-1.0.1-py3.7.egg/pycurrents\_ADCP\_processing/ADCP\_processing\_L0\_L1.py", line 563, in update\_meta\_dict orientations = [rdiraw.SysCfg(fl).up for fl in fixed\_leader.raw.FixedLeader['SysCfg']] File "/home/ethan/miniconda3/envs/adcp37/lib/python3.7/site-packages/pycurrents\_ADCP\_processing-1.0.1-py3.7.egg/pycurrents\_ADCP\_processing/ADCP\_processing\_L0\_L1.py", line 563, in <listcomp> orientations = [rdiraw.SysCfg(fl).up for fl in fixed\_leader.raw.FixedLeader['SysCfg']] AttributeError: module 'pycurrents.adcp.rdiraw' has no attribute 'SysCfg

Here is the section of code from ADCP\_processing\_L0\_L1.py

# Extract metadata from data object

# Orientation code from Eric Firing

# Orientation values such as 65535 and 231 cause SysCfg().up to generate an

# IndexError: list index out of range

try:

# list of bools; True if upward facing, False if down

orientations = [rdiraw.SysCfg(fl).up for fl in fixed\_leader.raw.FixedLeader['SysCfg']]

meta\_dict['orientation'] = utils.mean\_orientation(orientations)

except IndexError:

warnings.warn('Orientation obtained from data.sysconfig[\'up\'] to avoid '

'IndexError: list index out of range', UserWarning)

meta\_dict['orientation'] = 'up' if data.sysconfig['up'] else 'down'

The problem is that rdiraw.py is a deprecated stub:

"""

This module is obsolete, having been refactored into 3 new modules. As part of

the refactoring, octopusraw became raw\_simrad.

"""

import warnings

from .raw\_base import \* # noqa: F403

from .raw\_rdi import \* # noqa: F403

from .raw\_multi import \* # noqa: F403

warnings.warn(

"""

The rdiraw module is deprecated as of 2025-01-01. Imports should be switched

to one or more of raw\_base, raw\_rdi, raw\_simrad, or raw\_multi. For example,

Multiread is now in raw\_multi.""",

DeprecationWarning,

)

So SysCfg does not exist which is why I get the error. The solution might be to add SysCfg manually to rdiraw.py. I need to figure out what it should contain first, before modifying anything. Then I would need to reinstall the local (modified) copy of pycurrents and re-run the script. I think the best would be to try and find the older version of it? I have no idea.

The SysCfg is defined in the raw\_rdi.py. It might not be the correct one though. I am going to test if it works. In a new script:

from pycurrents.adcp.raw\_rdi import SysCfg

print(SysCfg(0).up) # False → upward-facing

print(SysCfg(255).up) # True → downward-facing

Should get this:

False

True

This matches the expected logic from pycurrents.adcp.raw\_rdi.SysCfg, which uses bit 7 for orientation:

 **Bit 7 = 0** → upward-facing (up = False)

 **Bit 7 = 1** → downward-facing (up = True)

Next step is to update ADCP\_processing\_L0\_L1.py to use the correct class.

**In ADCP\_processing\_L0\_L1.py:**

1. **Find and replace this line** (somewhere near the top):

from pycurrents.adcp import rdiraw

with:

from pycurrents.adcp.raw\_rdi import SysCfg

1. **Find this usage later in the code**: at line 563

orientations = [rdiraw.SysCfg(fl).up for fl in fixed\_leader.raw.FixedLeader['SysCfg']]

and replace it with:

orientations = [SysCfg(fl).up for fl in fixed\_leader.raw.FixedLeader['SysCfg'

Save the changes and rerun the script:   
(adcp37) ethan@bigtop:~/adcp/pycurrents\_ADCP\_processing$ python pycurrents\_ADCP\_processing/create\_nc.py

There are still issues. Possibly need to add import pycurrents.adcp.rdiraw as rdiraw back to the top of the script.

I reran the script, and the problem is still persisting. I am hitting a bit of a wall here. I might need to try again tomorrow.

Status: I need to try and figure out how to run the examples. If I fail to do so by mid-day, I should try and move on and see if I can figure out how to use one of my own data files.

My edits are in the right place, but my script is still pointing to the old .egg (zip installation archive). The recommendation is to uninstall the current installed version of the pycurrents\_ADCP\_processing package and then reinstall it in editable mode, so that Python uses the folder with the edited scripts for all of the imports. I don’t want to do this yet because I don’t even know if it is going to work.

Review file structure like this: (adcp37) ethan@bigtop:~/adcp$ tree -L 4

I went to review my file structure, and it looks like I have two copies of each module: one in site-packages (installed .egg) and one in the source folder (~/adcp/pycurrents\_ADCP\_processing/pycurrents\_ADCP\_processing). The dual existence is really confusing and is probably causing more problems than good. I went to check how I originally installed everything, and I found that installing some packages with python setup.py install is the main issue.

I am going to do a clean installation so that I can edit across all packages – safely and cleanly. Before I do that, I will make an installation script so that first time setup becomes somewhat automated. Here is what the installation script follows:

**✅ ADCP Processing Environment Setup (Clean & Editable)**

**📌 Pre-requisites**

* OS: Ubuntu (or Unix-like)
* Python version: **3.7** (use Conda for this)
* Tools: git, pip, conda, and optionally mercurial (for pycurrents)

**🔧 Step-by-step Installation**

**1. 🧰 Install Miniconda (skip if already installed)**

wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86\_64.sh

bash Miniconda3-latest-Linux-x86\_64.sh

source ~/.bashrc

**2. 📁 Create a working directory and Python 3.7 environment**

mkdir ~/adcp

cd ~/adcp

conda create -n adcp37 python=3.7

conda activate adcp37

**3. 📦 Configure Conda and install base dependencies**

conda config --add channels conda-forge

conda config --set channel\_priority strict

conda install numpy scipy pip pandas netCDF4 xarray gsw matplotlib=3.5 shapely cython

pip install ruamel.yaml

**4. 🧪 Clone and install ttide\_py in editable mode**

git clone https://github.com/moflaher/ttide\_py.git

cd ttide\_py

pip install -e .

cd ..

**5. 🌊 Clone and install pycurrents in editable mode**

sudo apt install mercurial # needed for `hg clone`

hg clone http://currents.soest.hawaii.edu/hg/pycurrents

cd pycurrents

pip install -e .

cd ..

**6. ⚙️ Clone and install pycurrents\_ADCP\_processing in editable mode**

git clone https://github.com/IOS-OSD-DPG/pycurrents\_ADCP\_processing.git

cd pycurrents\_ADCP\_processing

pip install -e .

cd ..

**7. ✅ Test the environment**

Launch Python and verify paths:

python

import pycurrents

print(pycurrents.\_\_file\_\_)

import pycurrents\_ADCP\_processing

print(pycurrents\_ADCP\_processing.\_\_file\_\_)

import ttide

print(ttide.\_\_file\_\_)

You should see paths like:

/home/ethan/adcp/pycurrents/pycurrents/\_\_init\_\_.py

/home/ethan/adcp/pycurrents\_ADCP\_processing/pycurrents\_ADCP\_processing/\_\_init\_\_.py

/home/ethan/adcp/ttide\_py/ttide/\_\_init\_\_.py

**🧼 Optional: Cleanup from Old Installations**

If you're reinstalling or correcting a broken setup, you can clean stale installs:

pip uninstall pycurrents pycurrents\_ADCP\_processing ttide

rm -rf pycurrents/\*.egg-info pycurrents\_ADCP\_processing/\*.egg-info ttide\_py/\*.egg-info

Also inspect and clean:

rm -rf ~/miniconda3/envs/adcp37/lib/python3.7/site-packages/\*pycurrents\*

rm -rf ~/miniconda3/envs/adcp37/lib/python3.7/site-packages/\*ttide\*

Ok I have the clean installation .sh file sorted now. I have not tested it yet though. I am now going to try and clean my current installation and switch to editable installs across all packages.

**Clean Install Plan (from where I am now)**

* 1. Remove installed versions (.egg) of the packages

Navigate to the adcp37 env, then run:

pip uninstall pycurrents\_ADCP\_processing

pip uninstall pycurrents

pip uninstall ttide

Check what is installed with:

pip list | grep -E "pycurrents|ttide"

Repeat pip uninstall until each package is completely gone.

* 1. Clean up site-packages manually – sometimes pip uninstall doesn’t remove .egg folders.

Check:

ls $(python -c "import site; print(site.getsitepackages()[0])") | grep -E "pycurrents|ttide"

If any folders are left, remove them:

rm -rf /home/ethan/miniconda3/envs/adcp37/lib/python3.7/site-packages/pycurrents\_ADCP\_processing\*

rm -rf /home/ethan/miniconda3/envs/adcp37/lib/python3.7/site-packages/pycurrents\*

rm -rf /home/ethan/miniconda3/envs/adcp37/lib/python3.7/site-packages/ttide\*

* 1. Reinstall only the local, editable versions of the three libraries.

cd ~/adcp/pycurrents

pip install -e .

cd ~/adcp/pycurrents\_ADCP\_processing

pip install -e .

cd ~/adcp/ttide\_py

pip install -e .

This adds symlinks in site-packages, so any edits made in the working folders will be used.

* 1. Confirm that the edited files are being used.

pip list | grep -E "ttide|pycurrents"

or,

python

import pycurrents

print(pycurrents.\_\_file\_\_)

import pycurrents\_ADCP\_processing

print(pycurrents\_ADCP\_processing.\_\_file\_\_)

import ttide

print(ttide.\_\_file\_\_)

Should see paths like this:

~/adcp/pycurrents/pycurrents/\_\_init\_\_.py

~/adcp/pycurrents\_ADCP\_processing/pycurrents\_ADCP\_processing/\_\_init\_\_.py

If there are still .egg or site-packages somewhere, there is a problem.

**Note on folder structure for Python:**

Using pycurrents as an example:

1. You need a top-level project directory, where the setup, readme and license, etc. live.
2. A package directory (source directory) which is the actual importable Python package, and it must match the name that you are installing.

So, in the end the structure is:   
  
pycurrents/ ← project root

├── pycurrents/ ← importable package

│ ├── \_\_init\_\_.py

│ └── other modules

├── setup.py

└── README.md

**After clean installation:**

I am now going to launch JL and try and run the example scripts. I need to figure out how to launch a JL kernel that is linked to this clean install. One that I can select once in JL.

**Register adcp37 as a Jupyter kernel**

Make sure that adcp37 is active, then run:

python -m ipykernel install --user --name adcp37 --display-name "Python (adcp37)"

Now launch as per usual:   
jupyter lab --port 8888 --no-browser

Then select adcp37 as the kernel in JL

**Trying to run examples after clean installation: Usage**

I got it to work!!! (I think it ran on the modified/patched ADCP\_processing\_L0\_L1.py)

Open the WD in Jupyter Lab.

Go to File, New and then open a terminal.

Activate the adcp37 environment in the JL terminal.

conda activate adcp37

cd pycurrents\_ADCP\_processing

python pycurrents\_ADCP\_processing/create\_nc.py

The output plots can be found here:

(adcp37) ethan@bigtop:~/adcp/pycurrents\_ADCP\_processing/dest\_dir

**Working with our data:**

Going to make a separate document for keeping track of how I work with the data.