TKE Analyst

Release latest

sschwindt

Mar 14, 2022

CONTENTS

This Python3 code aids in analyzing raw measurements with an Acoustic Doppler Velocimeter (ADV) producing `*.vno` and `*.vna` files. It detects and removes spikes according to Nikora and Goring (1998) and Goring and Nikora (2002).

The code was originally developed in Matlab(R) at the Nepf Environmental Fluid Mechanics Laboratory (Massachusetts Institute of Technology).

Important: `*.vno` and `*.vna` files need to comply with the following name convention: `XX_YY_ZZ_something.vna` where `XX`, `YY`, and `ZZ` are streamwise (x), perpendicular (y), and vertical (z) coordinates in CENTIMETERS, respectively. Anything else added after `ZZ_` is ignored by the code (it just copies it for the sake of dataset naming).

Note: This documentation is also as available as style-adapted PDF.

CHAPTER

ONE

REQUIREMENTS & INSTALLATION

Time requirement: 5-10 min.

1.1 Get Python

To get the code running, the following software is needed and their installation instructions are provided below:

- Python >=*3.6*
- NumPy >=*1.17.4*
- Openpyxl *3.0.3*
- Pandas >=*1.3.5*
- Matplotlib >=*3.1.2*

Start with downloading and installing the latest version of Anaconda Python. Alternatively, downloading and installing a pure Python interpreter will also work. Detailed information about installing Python is available in the Anaconda Docs and at hydro-informatics.com/python-basics.

To install the NumPy, Openpyxl, Pandas, and Matplotlib libraries after installing Anaconda, open Anaconda Prompt (e.g., click on the Windows icon, tap `anaconda prompt`, and hit enter``). In Anaconda Prompt, enter the following command sequence to install the libraries in the **base** environment. The installation may take a while depending on your internet speed.

```
conda install -c anaconda numpy
conda install -c anaconda openpyxl
conda install -c anaconda numpy
conda install -c conda-forge pandas
conda install -c conda-forge matplotlib
```

If you are struggling with the dark window and blinking cursor of Anaconda Prompt, worry not. You can also use Anaconda Navigator and install the four libraries (in the above order) in Anaconda Navigator.

Note: Alternatively, create a new conda environment to install the three libraries for this application. However, creating a new environment may eat up a lot of disk space, and installing the Python-omnipresent libraries NumPy, Openpyxl, Pandas, and Maplotlib in the **base** environment does not hurt.

1.2 Download tke-analyst Code

The code can be either started from Terminal (Anaconda Prompt) or within an Integrated Development Environment (**IDE**). With Anaconda installed, consider using Spyder (Anaconda Navigator > Spyder IDE).

Download tke-calculator.zip and unpack it to the directory where you want to run the code.

Tip: Alternatively to downloading the zip file, you may want to `git clone` the repository, which enables regular updating of the code (e.g., if there is an update of plot functions available). For using git, make sure that git bash is installed on your computer. Then, open git bash, cd into the directory where you want to download the code and type:

```
git clone https://github.com/sschwindt/tke-calculator.git
```

To update any time, `cd` into the directory where `tke-calculator` lives and type:

```
git pull --rebase
```

1.2.1 Usage

Regular Usage

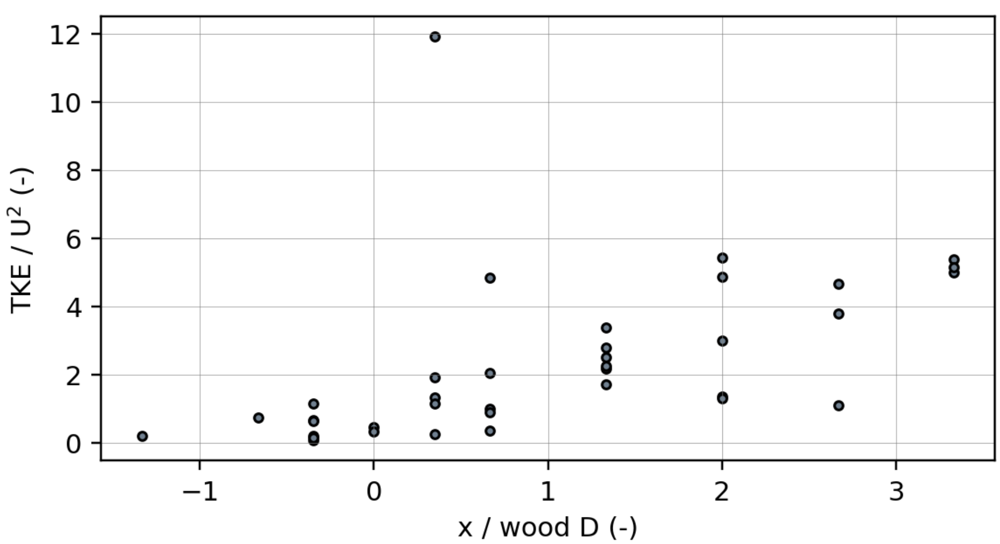
With Python installed and the code living on your computer:

- Copy your data to a sub-folder of `tke-analyst` (e.g., next to the folder `data/test-example` that contains three exemplary `*.vna` files). Make sure the files are named with `XX_YY_ZZ_something.vna` where `XX`, `YY`, and `ZZ` are streamwise (x), perpendicular (y), and vertical (z) coordinates in CENTIMETERS, respectively
- Complete the required information on the experimental setup in `tke-calculator/input.xlsx` (see below figure). **IMPORTANT: Never modify column A or any list in the sourcetables sheet (unless you also modify `load_input_defs` in line 25ff of `profile_analyst.py`).** The code uses the text provided in these areas of *input.xlsx* to identify setups. If useful, consider substituting the *Wood* wording in your mind and with a note in column C with your characteristic turbulence objects, but do not modify column A.
- **Open Anaconda Prompt (or any other Python-able Terminal) and:**
  - cd into the code directory (e.g., `cd "C:research\project\tke-analyst"` if you unpacked `tke-analyst` to a folder living in the directory `C:\research\project\`)
  - run the code: `python profile_analyst.py` (uses `input.xlsx`)
  - ALTERNATIVELY, run with another `*.xlsx` input file: `python profile_analyst.py "input-other-test.xlsx"`
  - wait until the code finished with `-- DONE -- ALL TASKS FINISHED --`
- **After a successful run, the code will have produced the following files in `...\tke-analyst\TEST` (where `TEST` may corresp**
  - `.xlsx` files of full-time series data, with spikes and despiked.
  - `.xlsx` files of statistic summaries (i.e., average, standard deviation *std*, TKE) of velocity parameters with x, y, and z positions, with spikes and despiked (see workbook example in the figure below).
  - Two plots (`norm-tke-x.png` and `norm-tke-x-despiked.png`) showing normalized TKE plotted against normalized x, with spikes and despiked, respectively (see plot example in the figure below).

PARAMETER	VALUE	UNIT / REMARK
Date	3/4/2022	*MM/DD/YYYY*
Input folder name (HOME/data/)	test-example	*string – lower and UPPER Cases are important*
Pump rate	1005	*rpm*
Flow rate (discharge)	0.0167	*CMS*
Probe Depth	0.06	*m*
Water depth	0.13	*m*
Flow velocity	0.24	*m/s*
Characteristic log length dimension	0.114	*m – use either diameter or length (choose B9 or B10)*
Log diameter	0.114	*m*
Log length	0.3	*m*
Log number	1.0	*#*
Log orientation	90	*deg relative to x-axis*
ADV freq	200	*Hz*
ADV time	0.05	*s*
ADV recording No.	3000	*records*
ADV direction	longitudinal	*optional (downward or sideward-longitudinally looking)*
Spike detection method	velocity	*define despiking method used with Goring & Nikora (2002)*
Despike lambda a	1.00	*if method=acceleration: multiplier of grav. accel; set between 1.0 and 1.5*
Despike k	3.00	*if method=velocity: multiplier of velocity stdev; set between 1.5 and 3.0*

Fig. 1.1: *The interface of the input.xlsx workbook for entering experiment parameters and specifying a despiking method.*

	x (m)	y (m)	z (m)
80_16.5_3.5_T5	0.8	0.165	0.035
80_31.5_3.5_T5	0.8	0.315	0.035
60_21.5_3.5_T5	0.6	0.215	0.035
60_41.5_3.5_T5	0.6	0.415	0.035
__10.5_5.5_3.5_T5	-0.105	0.055	0.035
__10.5_16.5_3.5_T5	-0.105	0.165	0.035
100_21.5_3.5_T5	1	0.215	0.035
__10.5_46.5_3.5_T5	-0.105	0.465	0.035
100_41.5_3.5_T5	1	0.415	0.035
40_21.5_3.5_T5	0.4	0.215	0.035
10.5_21.5_3.5_T5	0.105	0.215	0.035
40_41.5_3.5_T5	0.4	0.415	0.035
__10.5_41.5_3.5_T5	-0.105	0.415	0.035
10.5_41.5_3.5_T5	0.105	0.415	0.035
0_16.5_3.5_T5	0	0.165	0.035
60_46.5_3.5_T5	0.6	0.465	0.035
40_46.5_3.5_T5	0.4	0.465	0.035
10.5_5.5_3.5_T5	0.105	0.055	0.035
20_5.5_3.5_T5	0.2	0.055	0.035
__10.5_21.5_3.5_T5	-0.105	0.215	0.035
10.5_31.5_3.5_T5	0.105	0.315	0.035
20_21.5_3.5_T5	0.2	0.215	0.035
80_21.5_3.5_T5	0.8	0.215	0.035
40_5.5_3.5_T5	0.4	0.055	0.035
__20_31.5_3.5_T5	-0.2	0.315	0.035
60_16.5_3.5_T5	0.6	0.165	0.035
__10.5_31.5_3.5_T5	-0.105	0.315	0.035
10.5_46.5_3.5_T5	0.105	0.465	0.035
__40_31.5_3.5_T5	-0.4	0.315	0.035
20_31.5_3.5_T5	0.2	0.315	0.035
40_16.5_3.5_T5	0.4	0.165	0.035
60_31.5_3.5_T5	0.6	0.315	0.035
0_5.5_3.5_T5	0	0.055	0.035
20_16.5_3.5_T5	0.2	0.165	0.035
100_31.5_3.5_T5	1	0.315	0.035
40_31.5_3.5_T5	0.4	0.315	0.035
20_46.5_3.5_T5	0.2	0.465	0.035



Usage Example

For example, download and unpack the code to your hard-disk in a folder called `C:\my-project\tke-analyst\`. To analyze the `*.vna` files in `test-example`, they were copied into a test folder that lives in the `data` folder.

The definitions in the above-shown `input.xlsx` define x-normalization as a function of a wood log length, in this case, the log diameter of 0.114 m.

Cell `B3` containing **Input folder name (tke-analyst/)** in `input.xlsx` defines that the input data for `test-example` live in a subfolder called `data/test-example`.

Important: The data directory of the subfolder definition in cell `B3` may not end on any `\` or `/` . Also, make sure to **use the `/` sign for folder name separation** (do not use `\`).

To run the code with the example data, open Anaconda Prompt (or any other Python-able Terminal) and:

- `cd` into the code directory (e.g., `cd "C:research\project\tke-analyst"` if you unpacked `tke-analyst` to a folder living in the directory *C:researchproject*)
- run the code: `python profile_analyst.py` (uses `input.xlsx`)
- Or: `python profile_analyst.py "input.xlsx"`
- wait until the code finished with `-- DONE -- ALL TASKS FINISHED --`

- **After a successful run, the code will have produced the following files in `...\tke-analyst\data\test-example`:**
  - `.xlsx` files of full-time series data, with spikes and despiked.
  - `.xlsx` files of statistic summaries (i.e., average, standard deviation *std*, TKE) of velocity parameters with x, y, and z positions, with spikes and despiked.
  - Two plots (`norm-tke-x.png` and `norm-tke-x-despiked.png`) showing normalized TKE plotted against normalized x, with spikes and despiked, respectively.

1.2.2 Developer Docs

The following sections provide details of functions, their arguments, and outputs to help tweaking the code for individual purposes.

config.py

Global parameters settings (essentially SCRIPT_DIR) and message logging controls.

flowstat.py

`flowstat.`**`flowstat`**(*time*, *u*, *v*, *w1*, *w2*, *profile_type='lp'*)

Calculate ADV data statistics

Parameters

- **time** (*np.array*) – time in seconds
- **u** (*np.array*) – streamweise velocity along x-axis (positive in bulk flow direction)
- **v** (*np.array*) – perpendicular velocity along y-axis

- **w1** (*np.array*) – vertical velocity if side is DOWN
- **w2** (*np.array*) – vertical velocity if side is not DOWN
- **profile_type** (*str*) – orientation of the probe (default: lp, which mean probe looks like FlowTracker in a river)

Returns keys correspond to series names and values to full time series stats (dict(dict)): keys correspond to series names with STAT for autoreplacement with STAT type of nested dictionaries with AVRG, STD and STDERR

Return type time_series (dict)

profile_analyst.py

Load ADV measurements and calculate TKE with plot options Originally coded in Matlab at Nepf Lab (MIT) Re-written in Python by Sebastian Schwindt (2022)

`profile_analyst.build_stats_summary`(*vna_stats_dict*, *experiment_info*, *profile_type*, *bulk_velocity*, *log_length*)

Re-organize the stats dataset and assign probe coordinates

Parameters

- **vna_stats_dict** (*dict*) – the result of all vna files processed with the flowstat.flowstat function
- **experiment_info** (*dict*) – the result of the get_data_info function for retrieving probe positions
- **profile_type** (*str*) – profile orientation as a function of sensor position; the default is lp corresponding to DOWN (ignores w2 measurements)
- **bulk_velocity** (*float*) – bulk streamwise flow velocity in m/s (from input.xlsx)
- **log_length** (*float*) – characteristic log length (either diameter or length) in m (from input.xlsx)

Returns Organized overview pandas.DataFrame with measurement stats, ready for dumping to workbook

`profile_analyst.get_data_info`(*folder_name='test-example'*)

get names of input file names and prepare output matrix according to number of files

Parameters

- **folder_name** (*str*) – name of the test (experiment) to analyze (default is test-example)
- **input_file_name** (*str*) – name of input file (default is input.xlsx)

Returns pd.DataFrame with row names corresponding to file names ending on .vna, and columns X, Y, Z in meters

`profile_analyst.load_input_defs`(*file_name='/home/docs/checkouts/readthedocs.org/user_builds/tke-calculator/checkouts/latest/docs/input.xlsx'*)

loads provided input file name as pandas dataframe

Parameters **file_name** (*str*) – name of input file (default is input.xlsx)

Returns user input of input.xlsx (or costum file, if provided)

Return type (dict)

profile_analyst.**read_vna**(*vna_file_name*)

Read vna file name as pandas dataframe.

Parameters **vna_file_name** (*str*) – name of a vna file, such as __8_16.5_6_T3.vna

Returns pd.DataFrame

profile_analyst.**vna_file_name2coordinates**(*vna_file_name*)

Take vna file name and extract x, y, and z coordinates in meters. Non-convertible numbers are translated into np.nan with warning.

Parameters **vna_file_name** (*str*) – name of a vna file, such as __8_16.5_6_T3.vna

Returns list [x, y, z] coordinates

profile_plotter.py

Plot functions for TKE visualization

Note: The script represents merely a start for plotting normalized TKE against normalized X. If required, enrich this script with more plot functions and integrate them in profile_analyst.process_vna_files at the bottom of the function.

profile_plotter.**plot_xy**(*x*, *y*, *file_name*)

Plots y data against x (1d-numpy array) and markers of local maxima and minima

Parameters

- **x** (*numpy.array*) – x data
- **y** (*numpy.array*) – y data

Returns show and save plot in test folder as norm-TKE-x.png

rmspike.py

rmspike.**rmspike**(*vna_df*, *u_stats*, *v_stats*, *w_stats*, *w2_stats=None*, *method='velocity'*, *freq=200.0*, *lambda_a=1.0*, *k=3.0*, *profile_type='lp'*)

Spike removal and replacement - see Nikora & Goring (1999) and Goring & Nikora (2002).

Parameters

- **vna_df** (*pandas.DataFrame*) – matrix-like data array of the vna measurement file
- **u_stats** (*pandas.DataFrame*) – streamwise velocity stats from flowstat function
- **v_stats** (*pandas.DataFrame*) – perpendicular velocity stats from flowstat function
- **w_stats** (*pandas.DataFrame*) – vertical velocity stats from flowstat function
- **w2_stats** (*pandas.DataFrame*) – sec. vertical velocity stats from flowstat function (only required if profile_type is not lp)
- **method** (*str*) – determines whether to use acceleration or velocity (default) for despiking
- **freq** (*int*) – sampling frequency in 1/s (Hz); default is 200 Hz
- **lambda_a** (*float*) – multiplier of gravitational acceleration (acceleration threshold)
- **k** (*float*) – multiplier of velocity stdev (velocity threshold)
- **side** (*str*) – orientation of the probe (default: DOWN, which mean probe looks like Flow-Tracker in a river)

Note: Goring & Nikora (2002) suggest lambda_a = 1.0 ~ 1.5 and k = 1.5, but we shall use lambda_a = 1.0 and k = 3 ~ 9. SonTek, Nortek, and Lei recommend the SNR and correlation thresholds to be 15 and 70 respectively. Though data points have high SNR, the correlation can be low.

1.2.3 Disclaimer and License

Disclaimer (general)

No warranty is expressed or implied regarding the usefulness or completeness of the information provided for *tke-analyst* and its documentation. References to commercial products do not imply endorsement by the Author of *tke-analyst*. The concepts, materials, and methods used in the codes and described in the docs are for informational purposes only. The Author have made substantial effort to ensure the accuracy of the code and the docs and the Author shall not be held liable, nor their employers or funding sponsors, for calculations and/or decisions made on the basis of application of *tke-analyst*. The information is provided "as is" and anyone who chooses to use the information is responsible for her or his own choices as to what to do with the code, docs, and data and the individual is responsible for the results that follow from their decisions.

BSD 3-Clause License

PYTHON MODULE INDEX

INDEX