SAND2020-0841C

Marine and Hydrokinetic Toolkit (MHKiT) Workshop

PRESENTED AT

PAMEC 2020 San Jose, Costa Rica

PRESENTED BY

Katherine Klise and Kelley Ruehl

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Workshop Overview

- Motivation and Development
- MHKiT Code Hub and Software
- MHKiT Modules and Submodules
- Installation and Examples
- Future Development and Collaboration

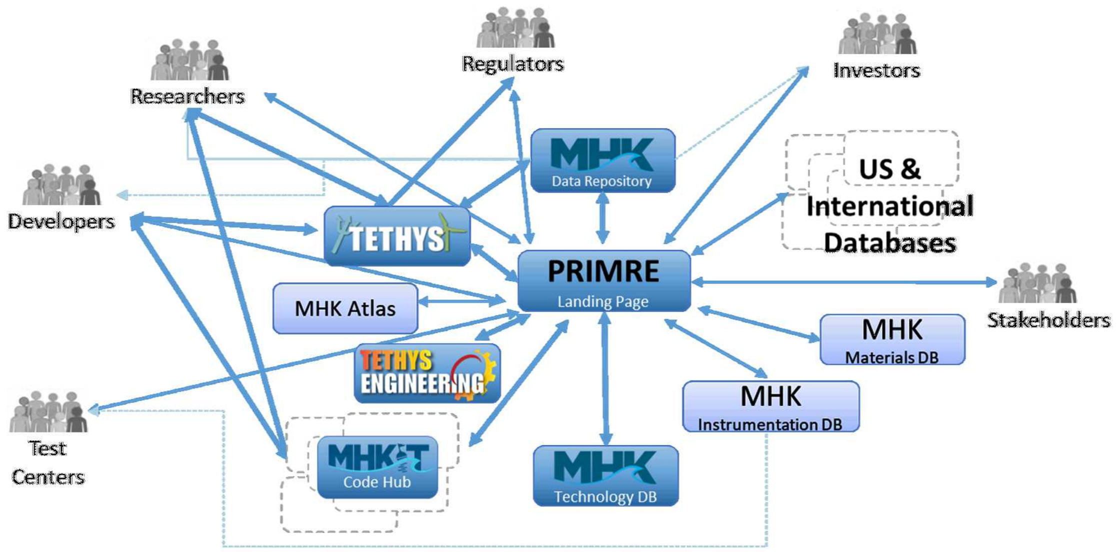
Motivation

Provide software with the ability to ingest, condition, reduce, quality control, process, visualize and store MRE data.

MHKiT software provides functionality for:

- Data processing (io)
- Data visualization (graphics)
- Data quality control (qc)
- Resource assessment (resource)
- Device performance (performance/device)

PRIMRE Network and Knowledge Hubs



The software is developed for MRE data, including measurements from field and laboratory environments, and datasets produced by numerical simulations.

MHKiT is part of the PRIMRE network, as one of the PRIMRE Knowledge Hubs (see network diagram)

https://mhkit-code-hub.github.io/MHKiT/overview.html

Development Team and Funding

MHKiT is developed as a collaboration between the National Renewable Energy Laboratory (NREL), Pacific Northwest National Laboratory (PNNL), and Sandia National Laboratories (SNL). The core development team is listed below:

- Frederick Driscoll (NREL - PI)
- Budi Gunawan (Sandia - PI)
- Katherine Klise (Sandia)
- Sterling Olson (Sandia)
- Rebecca Pauly (NREL)
- Kelley Ruehl (Sandia)
- Timothy Shippert (PNNL)
- Chitra Sivaraman (PNNL - PI)

MHKiT is funded by the U.S. Department of Energy's Water Power Technologies Office.

https://mhkit-code-hub.github.io/MHKiT/

Licensing

BSD-3-Clause

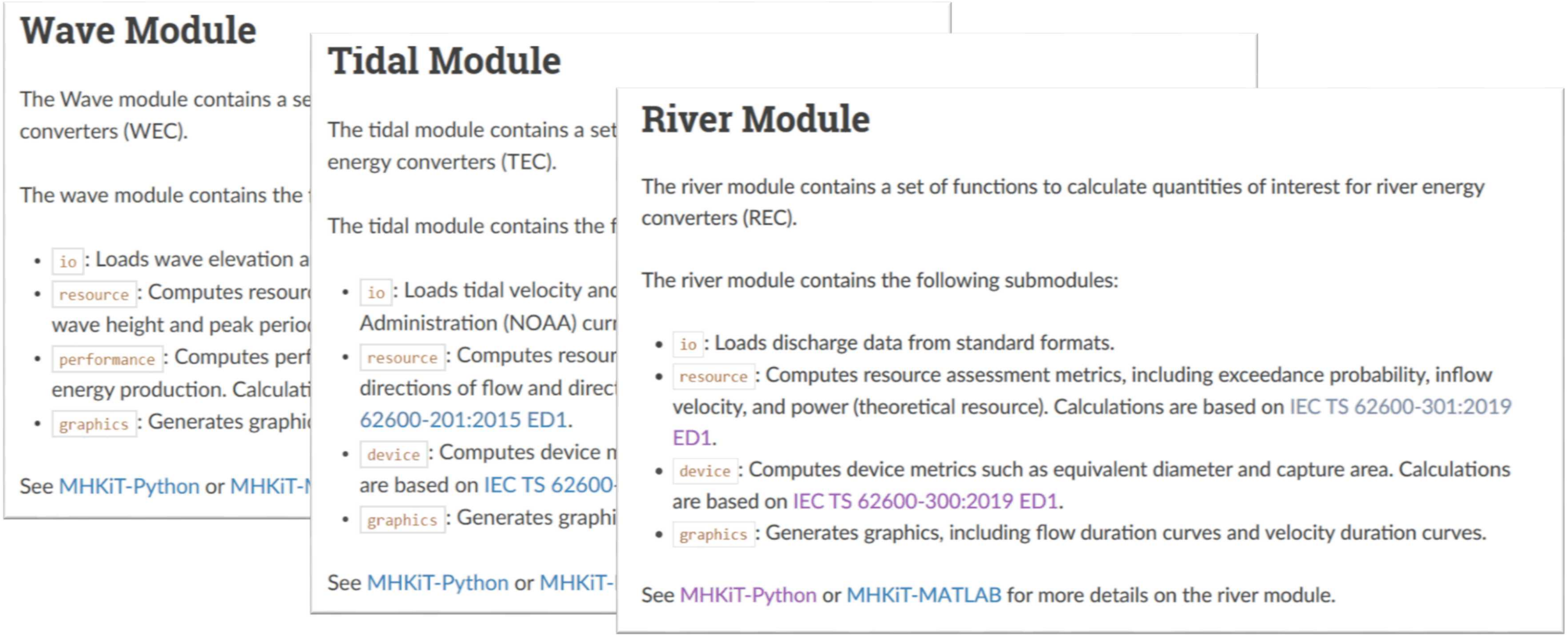
```
Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this
  list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice,
  this list of conditions and the following disclaimer in the documentation
  and/or other materials provided with the distribution.
* Neither the name of Alliance for Sustainable Energy, Battelle Memorial
  Institute, National Renewable Energy Laboratory, National Technology &
  Engineering Solutions of Sandia, Pacific Northwest National Laboratory,
  Sandia National Laboratories, nor the names of its contributors may be
  used to endorse or promote products derived from this software without
  specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
```

Revised BSD open source license

Copyright 2019, Alliance for Sustainable Energy, LLC under the terms of Contract DE-AC36-08GO28308, Battelle Memorial Institute under the terms of Contract DE-AC05-76RL01830, and National Technology & Engineering Solutions of Sandia, LLC under the terms of Contract DE-NA0003525. The U.S. Government retains certain rights in this software.

IEC TC114 Standards

Calculations in MHKiT adhere to the International Electrotechnical Committee's Technical Committee's, IEC TC 114 technical specifications (TS) and recommendations, as well as follow best practices within the MRE and other fields.



Wave Module

The Wave module contains a se
converters (WEC).

The wave module contains the

- io : Loads wave elevation a
- resource : Computes resour
wave height and peak perio
- performance : Computes perf
energy production. Calculati
- graphics : Generates graphi

See MHKiT-Python or MHKiT-I

Tidal Module

The tidal module contains a set
energy converters (TEC).

The tidal module contains the f

- io : Loads tidal velocity and
Administration (NOAA) curr
- resource : Computes resour
directions of flow and direct
62600-201:2015 ED1.
- device : Computes device m
are based on IEC TS 62600-
- graphics : Generates graphi

See MHKiT-Python or MHKiT-

River Module

The river module contains a set of functions to calculate quantities of interest for river energy converters (REC).

The river module contains the following submodules:

- io : Loads discharge data from standard formats.
- resource : Computes resource assessment metrics, including exceedance probability, inflow velocity, and power (theoretical resource). Calculations are based on IEC TS 62600-301:2019 ED1.
- device : Computes device metrics such as equivalent diameter and capture area. Calculations are based on IEC TS 62600-300:2019 ED1.
- graphics : Generates graphics, including flow duration curves and velocity duration curves.

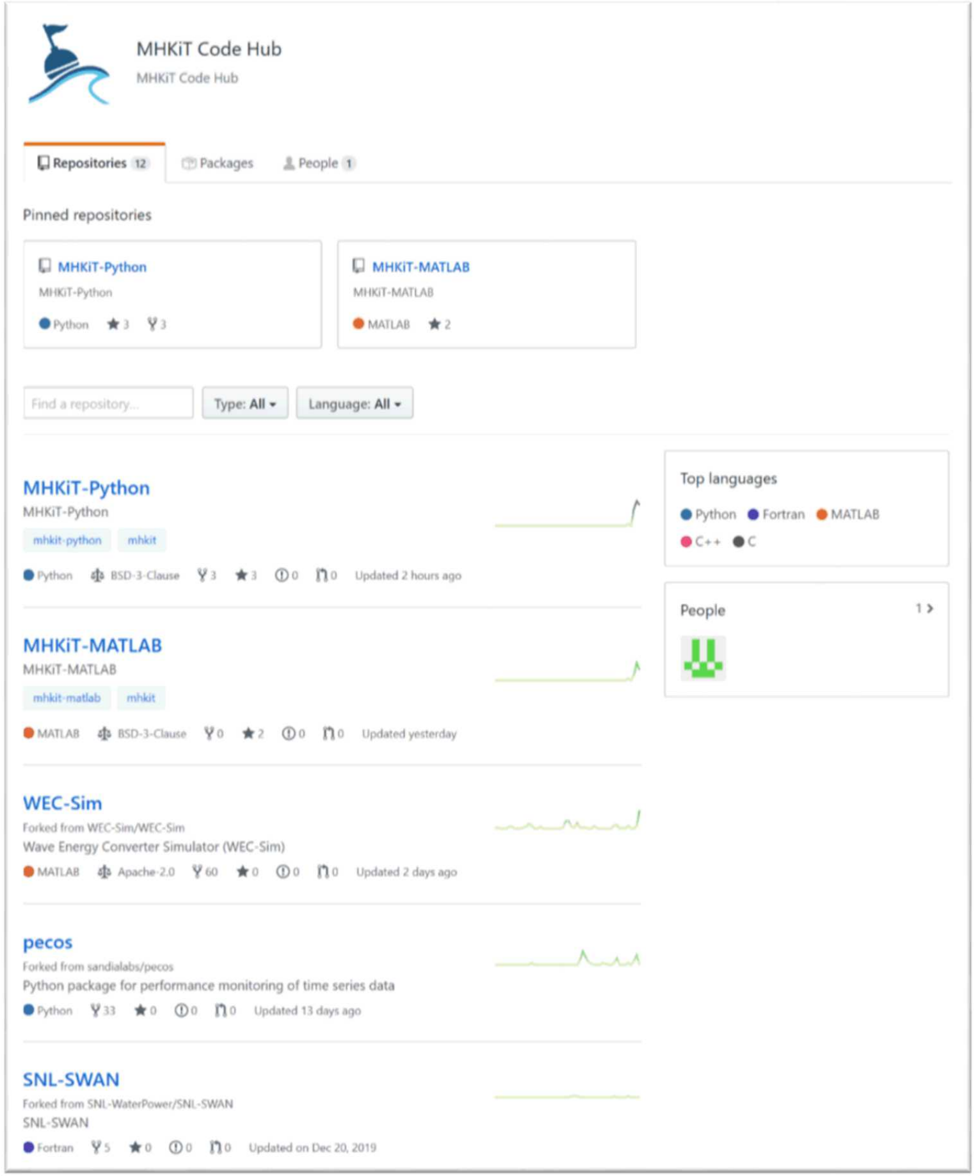
See MHKiT-Python or MHKiT-MATLAB for more details on the river module.

MHKiT Code Hub

MHKiT Code Hub is a collection of MRE software repositories, including:

- **MHKiT-Python**: Software developed in Python
- **MHKiT-MATLAB**: Software developed in MATLAB
- Other relevant open source software:
  - **Pecos**: Data quality control analysis
  - **WEC-Sim**: Wave energy converter simulator
  - **Dolfyn**: Doppler oceanography library
  - …

https://github.com/MHKiT-Code-Hub



MHKiT-Python and MHKiT-MATLAB

MHKiT-Python and MHKiT-MATLAB are organized into the following modules:

- **QC Module**: Perform quality control analysis
- **Wave Module**: Calculate quantities of interest for wave energy converters (WEC)
- **River Module**: Calculate quantities of interest for river energy converters (REC)
- **Tidal Module**: Calculate quantities of interest for tidal energy converters (TEC)
- **Utils Module**: Includes helper functions

These modules provide functionality for calculating metrics needed by the MRE community as well as those required for conformity with IEC TS and recommendations. MHKiT-Python includes continuous integration software tests that are run using Travis CI.

To ensure consistent results between MHKiT-Python and MHKiT-MATLAB, all functions are written in Python and housed in the MHKiT-Python repository. MHKiT-MATLAB then wraps these functions so they can be called from MATLAB.

MHKiT-Python and MHKiT-MATLAB provide identical functions in each language.

https://mhkit-code-hub.github.io/MHKiT/overview.html

Terminology and Units

Term	Definition
A_P	Projected capture area [m^2]
BS	Bretschneider spectrum
D_E	Equivalent diameter [m]
E	Energy [J]
η	Incident wave [m]
f	Frequency [Hz]
F	Exceedance probability [%]
Fr	Froude Number
g	Gravity [m/s/s]
h	Water depth from bottom to water surface (e.g. SWL) [m]
H	Wave height [m]
H_s	Significant wave height, mean wave height of the tallest third of waves [m]
H_{m0}	Spectrally derived significant wave height [m]
J	Wave energy flux [W/m]
JS	JONSWAP spectrum
L	Capture length [m]

Term	Definition
k	Wave number, $k = \frac{2\pi}{\lambda}$ [rad/m]
m	Mass [kg]
m_k	Spectral moment of k, for k = 0,1,2,...
ω	Wave frequency, $\omega = \frac{2\pi}{T}$ [rad/s]
P	Power [W]
PM	Pierson-Moskowitz specturm
Q	Discharge [m^3/s]
ρ	Density [kg/m^3]
S	Spectral density [m^2/Hz]
SWL	Still water line
T_e	Energy period [s]
T_m	Mean wave period [s]
T_p	Peak period [s]
T_z	Zero-crossing period [s]
v	Velocity [m/s]
V	Velocity calculated for river and tidal modules [m/s]

The methods in MHKiT use the MKS (meters-kilograms-seconds) system, and assume data is stored in SI units.

https://mhkit-code-hub.github.io/MHKiT/terminology.html

Installation

The following MHKiT demonstration focuses on MHKiT-Python

MHKiT-Python requires Python (3.6 or 3.7) and has the following Python packages dependencies:

- Pandas: used for data storage and analysis
- NumPy: used for data storage and analysis
- SciPy: used for numerical methods, statistics, and signal processing
- Matplotlib: used to produce figures
- Requests: used to get data from websites
- Pecos v0.1.8: used for quality control analysis

It is recommended to use the Anaconda Python Distribution, since it includes all of the MHKiT-Python package dependencies except Pecos.

Users are encouraged to install MHKiT-Python using PIP: **pip install mhkit**

https://mhkit-code-hub.github.io/MHKiT/installation.html

Getting Started

To get started, import MHKiT-Python

```
import mhkit
```

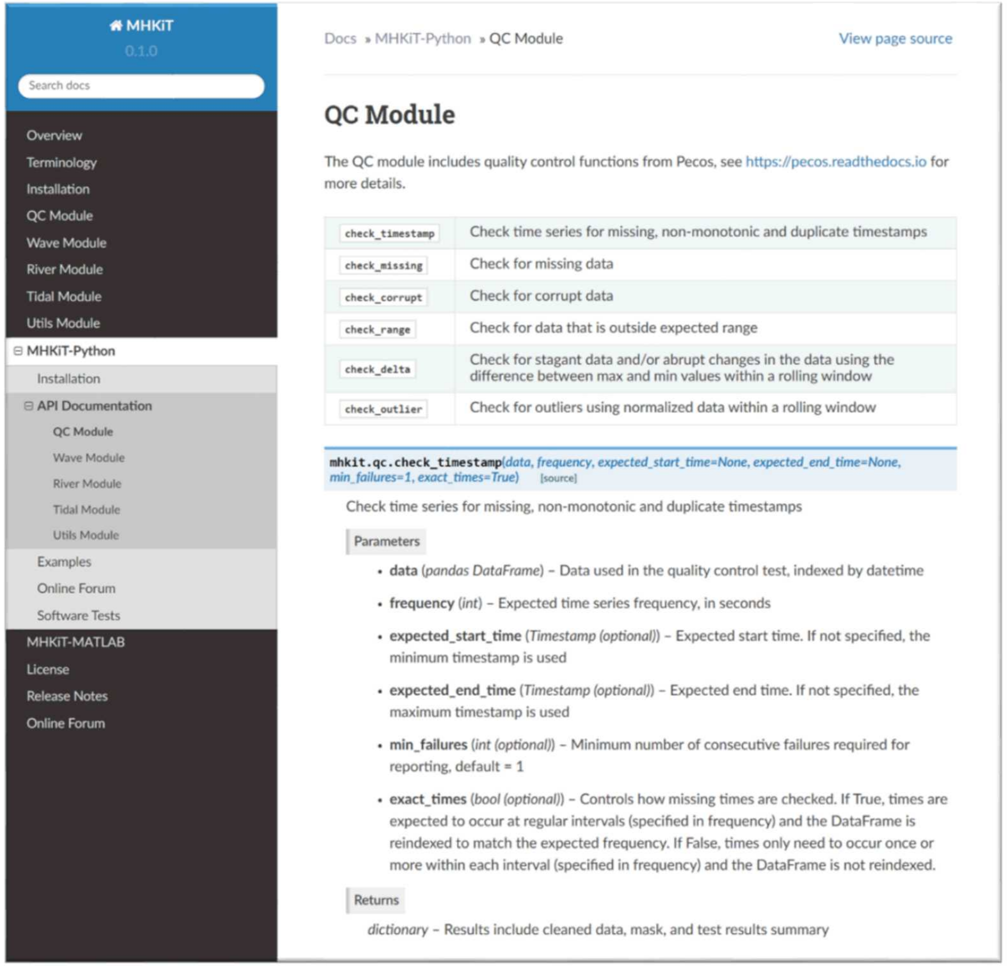
Access the qc, wave, river, tidal, and utils modules using the following syntax

```
mhkit.qc
mhkit.wave
mhkit.river
mhkit.tidal
mhkit.utils
```

Individual functions are accessed from each module, for example:

```
mhkit.qc.check_timestamp
```

API documentation includes a list of functionality for each module along with details on function input and output



MHKiT
0.1.0
Search docs

Overview
Terminology
Installation
QC Module
Wave Module
River Module
Tidal Module
Utils Module
⊟ MHKiT-Python
Installation
⊟ API Documentation
QC Module
Wave Module
River Module
Tidal Module
Utils Module
Examples
Online Forum
Software Tests
MHKiT-MATLAB
License
Release Notes
Online Forum

Docs » MHKiT-Python » QC Module — View page source

QC Module

The QC module includes quality control functions from Pecos, see https://pecos.readthedocs.io for more details.

`check_timestamp`	Check time series for missing, non-monotonic and duplicate timestamps
`check_missing`	Check for missing data
`check_corrupt`	Check for corrupt data
`check_range`	Check for data that is outside expected range
`check_delta`	Check for stagant data and/or abrupt changes in the data using the difference between max and min values within a rolling window
`check_outlier`	Check for outliers using normalized data within a rolling window

mhkit.qc.check_timestamp(*data, frequency, expected_start_time=None, expected_end_time=None, min_failures=1, exact_times=True*) [source]

Check time series for missing, non-monotonic and duplicate timestamps

Parameters

- **data** (*pandas DataFrame*) – Data used in the quality control test, indexed by datetime
- **frequency** (*int*) – Expected time series frequency, in seconds
- **expected_start_time** (*Timestamp (optional)*) – Expected start time. If not specified, the minimum timestamp is used
- **expected_end_time** (*Timestamp (optional)*) – Expected end time. If not specified, the maximum timestamp is used
- **min_failures** (*int (optional)*) – Minimum number of consecutive failures required for reporting, default = 1
- **exact_times** (*bool (optional)*) – Controls how missing times are checked. If True, times are expected to occur at regular intervals (specified in frequency) and the DataFrame is reindexed to match the expected frequency. If False, times only need to occur once or more within each interval (specified in frequency) and the DataFrame is not reindexed.

Returns

dictionary – Results include cleaned data, mask, and test results summary

Sub-Modules

The wave, river, and tidal modules are divided into submodules:

Wave Module

The Wave module contains a set
converters (WEC).

The wave module contains the fo

- `io`: Loads wave elevation and
- `resource`: Computes resource
  wave height and peak period.
- `performance`: Computes perfo
  energy production. Calculation
- `graphics`: Generates graphics,

See MHKiT-Python or MHKiT-M

https://mhkit-code-h

Tidal Module

The tidal module contains a set of
energy converters (TEC).

The tidal module contains the foll

- `io`: Loads tidal velocity and di
  Administration (NOAA) curren
- `resource`: Computes resource
  directions of flow and directio
  62600-201:2015 ED1.
- `device`: Computes device met
  are based on IEC TS 62600-20
- `graphics`: Generates graphics,

See MHKiT-Python or MHKiT-M

https://mhkit-code-h

River Module

The river module contains a set of functions to calculate quantities of interest for river energy converters (REC).

The river module contains the following submodules:

- `io`: Loads discharge data from standard formats.
- `resource`: Computes resource assessment metrics, including exceedance probability, inflow velocity, and power (theoretical resource). Calculations are based on IEC TS 62600-301:2019 ED1.
- `device`: Computes device metrics such as equivalent diameter and capture area. Calculations are based on IEC TS 62600-300:2019 ED1.
- `graphics`: Generates graphics, including flow duration curves and velocity duration curves.

See MHKiT-Python or MHKiT-MATLAB for more details on the river module.

https://mhkit-code-hub.github.io/MHKiT/river.html

Pandas Basics

MHKiT-Python uses Pandas DataFrames to store labelled data structures.

- Each column is labelled with a descriptive name
- Data is indexed by time in seconds or by datetime
- The Utils Module can be used to convert numeric indexes to datetime indexes

```
        probe1  probe2  probe3
Time
10.000   24.48   28.27     1.3
10.002   34.48   40.27    -8.7
10.004   30.48   38.27   -13.7
10.006   12.48   24.27   -32.7
10.008   13.48   22.27   -21.7
```

```
                         probe1  probe2  probe3
Time
2019-05-20 00:00:10.000   24.48   28.27     1.3
2019-05-20 00:00:10.002   34.48   40.27    -8.7
2019-05-20 00:00:10.004   30.48   38.27   -13.7
2019-05-20 00:00:10.006   12.48   24.27   -32.7
2019-05-20 00:00:10.008   13.48   22.27   -21.7
```

Pandas includes many options including the ability to:

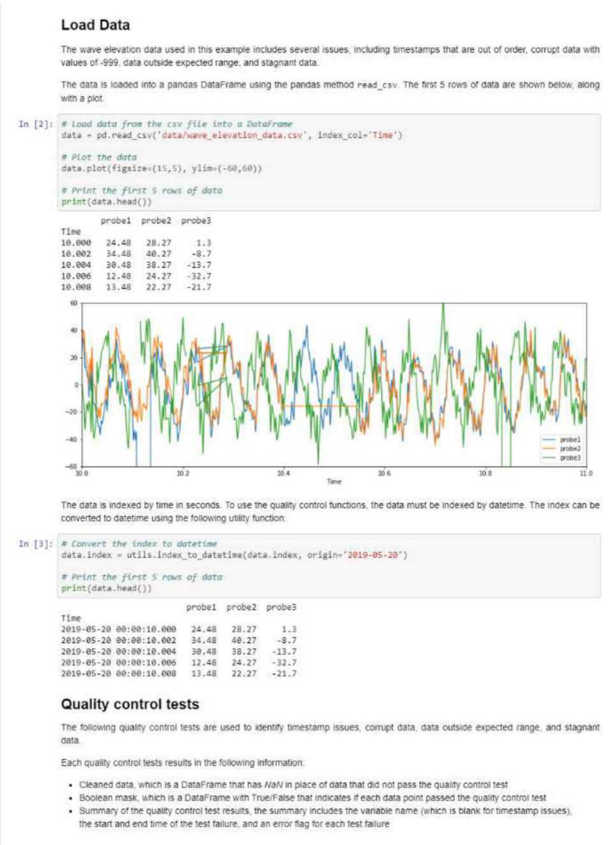
- Read data from various file formats (csv, excel, json, database, etc.)
- Write data and results to various file formats
- Plot timeseries data
- Slice and query data
- Upscale and downscale the time resolution of data
- Fill and interpolate gaps in data
- Compute moving window statistics

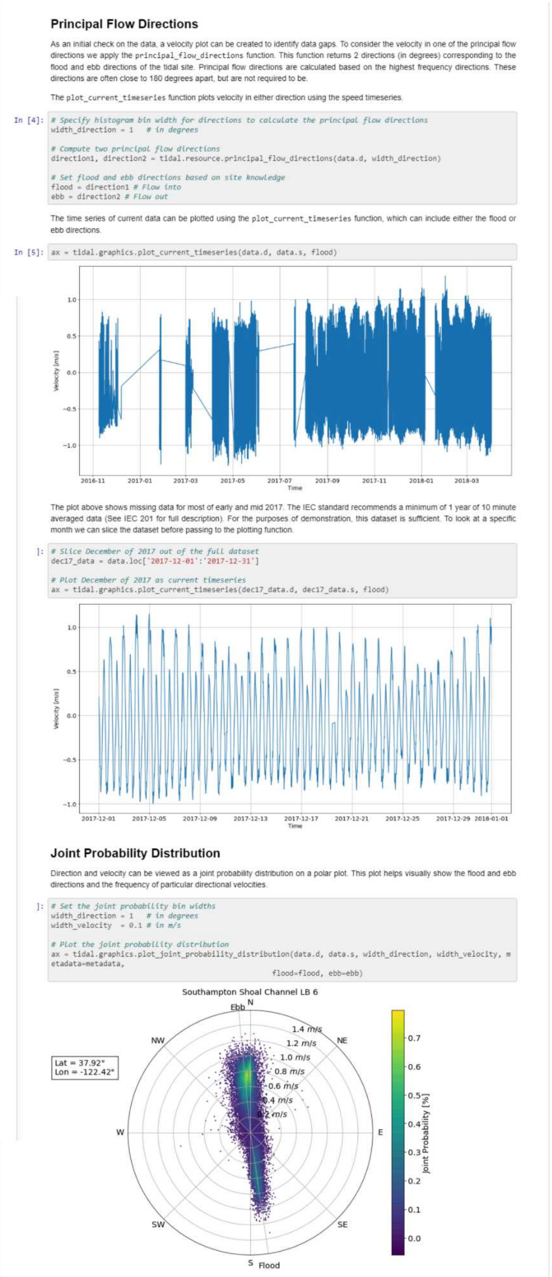
MHKiT-Python users that are new to Pandas are encouraged to review the Pandas getting started guide.

Jupyter Notebook Examples

MHKiT-Python includes four Jupyter Notebook examples to demonstrate functionality, these include:

- **QC example**: run a simple quality control analysis on wave elevation data
- **Wave example**: generate a capture length matrix, calculate MAEP, and plot scatter diagrams
- **River example**: calculate annual energy produced for one turbine in the Tanana river near Nenana, Alaska
- **Tidal example**: calculate velocity duration curve using 1 year of data from the NOAA-Currents sites



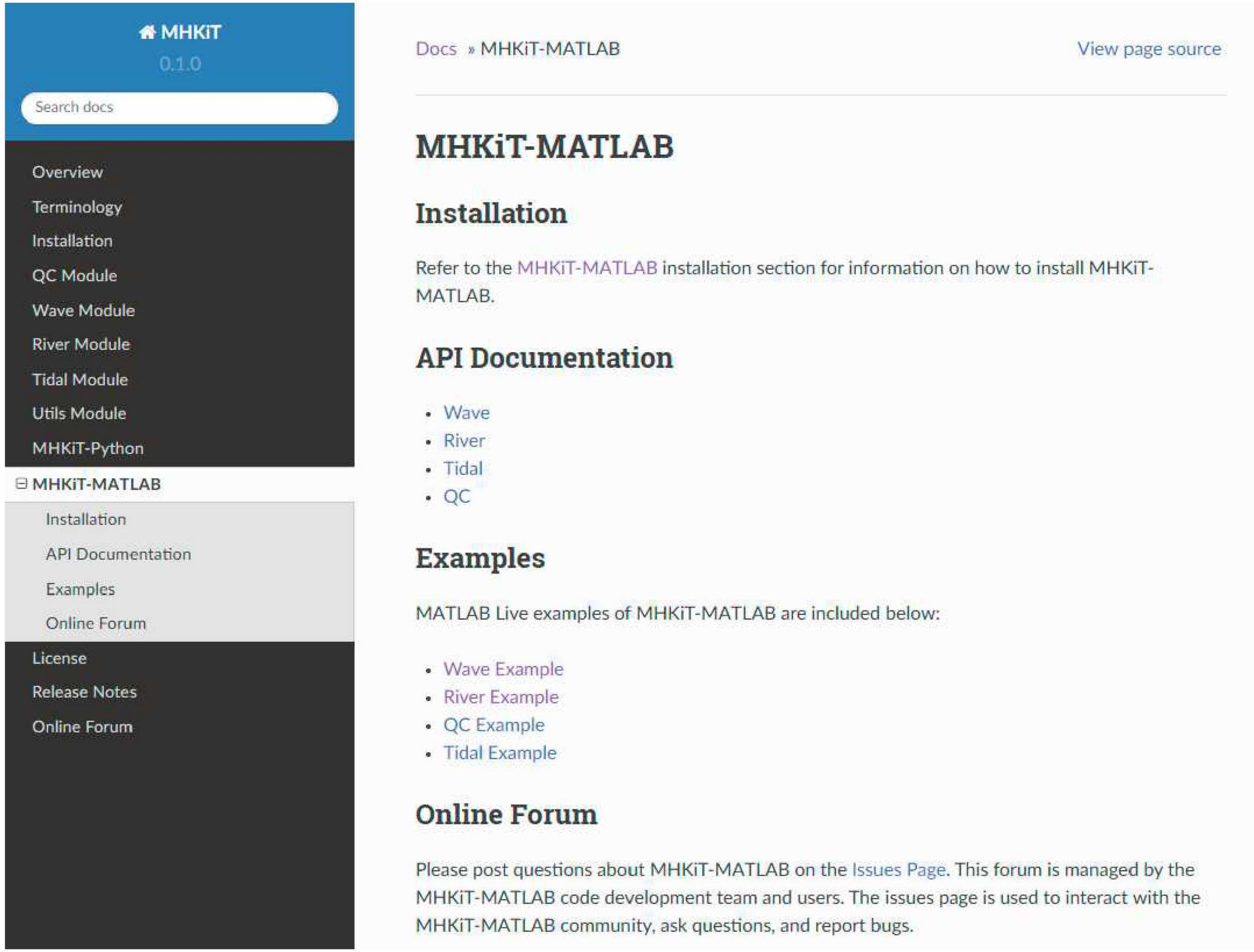


https://mhkit-code-hub.github.io/MHKiT/python.html#examples

MHKiT-MATLAB

MHKiT-MATLAB is intended to be used by researchers and practitioners that prefer MATLAB.

- MHKiT-MATLAB runs the MHKiT-Python functions by wrapping them in MATLAB.
- The software includes QC, Wave, River, and Tidal modules.
- Documentation includes installation instructions, API documentation, and MATLAB Live examples
- The online forum is hosted on the MHKIT-MATLAB GitHub site.



MHKiT
0.1.0
Search docs

- Overview
- Terminology
- Installation
- QC Module
- Wave Module
- River Module
- Tidal Module
- Utils Module
- MHKiT-Python
- MHKiT-MATLAB
  - Installation
  - API Documentation
  - Examples
  - Online Forum
- License
- Release Notes
- Online Forum

Docs » MHKiT-MATLAB — View page source

MHKiT-MATLAB

Installation

Refer to the MHKiT-MATLAB installation section for information on how to install MHKiT-MATLAB.

API Documentation

- Wave
- River
- Tidal
- QC

Examples

MATLAB Live examples of MHKiT-MATLAB are included below:

- Wave Example
- River Example
- QC Example
- Tidal Example

Online Forum

Please post questions about MHKiT-MATLAB on the Issues Page. This forum is managed by the MHKiT-MATLAB code development team and users. The issues page is used to interact with the MHKiT-MATLAB community, ask questions, and report bugs.

https://mhkit-code-hub.github.io/MHKiT/matlab.html

Current and Future Releases

MHKiT v0.1.0

The first official release of MHKiT, developed in Python and MATLAB, includes the following modules:

- QC Module: Perform quality control analysis
- Wave Module: Calculate quantities of interest for wave energy converters (WEC)
- River Module: Calculate quantities of interest for river energy converters (REC)
- Tidal Module: Calculate quantities of interest for tidal energy converters (TEC)
- Utils Module: Includes helper functions

The v0.1.0 release includes methods for resource assessment, device performance, graphics, io and and quality control.

MHKiT-Python v0.1.0

MHKiT-MATLAB v0.1.0

Future Releases

The next release of MHKiT, planned for Summer 2020, will include the following modules:

- Mechanical Loads Module: Calculates quantities of interest for mechanical loads.
- Power Quality Module: Calculates quantities of interest for power quality.

https://mhkit-code-hub.github.io/MHKiT/release_notes.html

Online Forum

Issues

Questions, feature requests, and bug reports for MHKiT-Python and MHKiT-MATLAB should be submitted to the GitHub Issues Page. The GitHub online forums are managed by the MHKiT development team and users.

Submit MHKiT-Python Issue

Submit MHKiT-MATLAB Issue

Collaboration

MHKiT-Python and MHKiT-MATLAB welcomes feedback and code contributions. Software developers interested in contributing to the MHKiT open-source software are encouraged to use GitHub to create a Fork of the repository into their GitHub user account. To include your additions to the MHKiT code, please submit a pull request in the master branch of the . Once reviewed by the MHKiT development team, pull requests will be merged into MHKiT master branch, and included in future releases of MHKiT. Software developers, within the MHKiT development team and external collaborators, are expected to follow standard practices to document and test new code.

Submit MHKiT-Python Pull Request

Submit MHKiT-MATLAB Pull Request

https://mhkit-code-hub.github.io/MHKiT/contact.html

Thank you!

Katherine.Klise@sandia.gov

Kelley.Ruehl@sandia.gov