



## Announcing FLORIS Version 3.4 and FLASC Version 1.3

The [FLOW Redirection and Induction in Steady State \(FLORIS\)](#) development team is very excited to announce the release of [FLORIS](#) Version 3.4 (v3.4) and [FLASC](#) (FLORIS-based Analysis for Supervisory Control and data acquisition) version 1.3 (v1.3). These releases include several exciting enhancements to both FLORIS and FLASC.

### Major Improvements in FLORIS v3.4

The full list of changes, along with their corresponding pull request, is available at [the v3.4 GitHub Release](#), and the primary changes are listed below.

- **Modeling wake effects of floating offshore wind turbines** ([#518](#))

A key new feature is the support for changing tilt angle as a function of wind speed. Also included is the ability for the turbine Cp/Ct tables to be specified with or without platform tilt.

- **Empirical Gaussian Wake Model** ([#628](#))

This new model emphasizes the simplification of tuning while maintaining similar mathematical structures to the Gauss-Curl-Hybrid (GCH) model. This model reduces the number of available free parameters and seeks to avoid cross-coupling by linking one parameter to one model effect. In internal testing, the model performed well against existing models in SCADA data comparisons. It also includes features not available in former models, such as vertical wake deflection due to rotor tilt and ground effects on wake recovery. Finally, the model is 2–3 times faster than the GCH model. Although the Empirical Gaussian Wake Model is still in active development, the FLORIS team is very excited to share it and collaborate with interested parties. The new model will be presented at the May 2023 Wind Energy Science Conference.

- **Sampling of arbitrary flow points** ([#650](#))

Probing the flow field to extract velocities at user-defined points is now supported. This could be used, for example, to compare FLORIS predictions to wind velocity measurements taken by a meteorological mast or compare to large eddy simulation predictions at specific points in the flow.

- **Heterogenous Inflow Definition and Visualization** ([#578](#))

The heteroneous wind speeds and their placement in the farm can now be included directly in the YAML input file. Additionally, the visualization of the flow field includes a map of the heterogeneous inputs to provide a clear validation and understanding of the model configuration. The orientation and layout of the wind farm is now fixed in the user-specified orientation and the wind direction rotates around the farm.

- **Improvements to rotor velocity approximation** ([#649](#))

A new method for integrating the velocities across the rotor plane is included resulting in a more accurate approximation of the average rotor velocity for approximately the same computational cost.

- **Turbine library preprocessing** ([#639](#))

New tools to analyze turbine model definitions and compare turbines have been added as well as a new module `floris.turbine_library`. The modules includes a class `TurbineInterface` for single turbines and `TurbineLibrary` for the full library. This functionality can be used with the turbine modules included within FLORIS and any user-supplied turbine input file. See the [Turbine Library Interface](#) page of the documentation for a reference and more information.

- **Bug fixes**

- ([#637](#)) A fix for fundamental class generation errors due to mismatched configuration keys with the new version now providing the intended helpful error messaging in the “FromDictMixin” class.
- ([#622](#)) A fix to correctly label error message for unfindable turbine configurations where the message previously indicated that the issue was due to a duplicated file.

## Major Improvements in FLASC Version 1.3

FLASC v1.3 includes several improvements the team is excited to share with the research community, including:

- A new ability to use FLASC with SCADA data from commercial farm wake steering tests. This release includes a new set of examples that demonstrate how to use FLASC tools with SCADA data collected from wake steering tests at commercial farms. The data comes from the [SMARTEOLE Wind Farm Control open data set](#), which comprises a [field campaign](#) carried out on a seven-turbine wind farm at Sole du Moulin Vieux, France. The new examples use real data sets to show the how to use FLASC methods with physical SCADA data as well as to demonstrate the recreation of the analysis of the original paper. The examples also illustrate the structure and filtering of raw data and provide a matched FLORIS model. The steps are based on a workflow developed by Bart Doekemeijer in a separate FLASC Cookiecutter template repo.
- Improvements to the methods FLASC uses to energy ratio class. Now, FLASC includes methods to compute gain in energy ratio. Additionally, users can employ block and conventional bootstrapping to calculate uncertainty bounds. Finally, the team improved energy ratio visualizations so that they now illustrate wind speed content by wind direction bin and allow users to plot results on a polar axis. These improvements are again thanks to contributions from Bart Doekemeijer.
- Improvements in data filtering. The FLASC preprocessing data filtering methods have been significantly improved in v1.3 because of contributions from Bart Doekemeijer. The method to call each individual filtering step has been made more consistent, and visualization of the effect of each filtering step is now included.
- Detection of impacting turbines and methods to filter based on neighboring turbines. FLASC now includes methods to determine which turbines impact each other and enables users to filter results based on these relationships to control for the impact of faulted or curtailed turbines on downstream turbines in analyses.

## Upcoming Mini Symposium at WESC 2023

Those attending the May 2023 [Wind Energy Science Conference \(WESC\)](#) in Glasgow, United Kingdom, are invited to join a mini symposium on Open-Source Wind Farm Flow Model Libraries, co-organized by Technical University of Denmark, the National Renewable Energy Laboratory (NREL), and Fraunhofer Institute for Wind Energy and Energy System Technology. See the [LinkedIn post](#) for more information.

## How To Upgrade FLORIS

For full installation and upgrade instructions, please see the [installation documentation](#).

If you have installed FLORIS with PyPI or Conda, you can upgrade to the latest version using your package manager by pasting the corresponding

command below into a terminal:

```
conda update floris
pip install floris --update
```

Alternatively, if you've installed FLORIS through a local clone of the repository, you can upgrade by pulling the latest release with the following commands:

```
cd floris/
git fetch --all

# If you've made uncommitted changes to the code
git stash

# "origin" should be replaced by the appropriate remote
name
git pull origin main

# Restore any uncommitted changes stashed
git stash pop

# Install locally
pip install -e .
```

## How To Upgrade FLASC

FLASC is not yet available in Conda or PyPI, so the local upgrade method is the only currently available option.

To upgrade locally use the following commands:

```
cd flasc/
git fetch --all

git stash # if you've made uncommitted changes to the
source code
git pull origin main # "origin" should be replaced by the
appropriate remote
```

```
git stash pop # restores any uncommitted changes stash
above
```

```
# Install locally
pip install -e .
```

## How To Engage With the FLORIS and FLASC Community

FLORIS leverages the following features of GitHub to coordinate support and development efforts:

[Discussions](#): Collaborate to develop ideas for new use cases, features, and software designs, and get support for usage questions

[Issues](#): Report potential bugs and well-developed feature requests

[Projects](#): Include current and future work on a timeline and assign a person to "own" it.

On GitHub, the first entry point for the community is generally within one of the categories in the [Discussions](#). [Ideas](#) is a great spot to develop the details for a feature request. Find usage support in the [Q&A](#) section. As a freeform space, [Show and Tell](#) allows users to show off the things they are doing with FLORIS.

Guidelines for interacting with the repository with git and GitHub are available in the [developer documentation](#).

~~~

If you have additional input, no longer wish to receive updates on FLORIS, or just discovered FLORIS and would like to start receiving updates, please send an email to [Paul.Fleming@nrel.gov](mailto:Paul.Fleming@nrel.gov).

*This work was authored by the National Renewable Energy Laboratory, operated by Alliance for Sustainable Energy, LLC, for the U.S. Department of Energy (DOE) under Contract No. DE-AC36-08GO28308. Funding provided by the U.S. Department of Energy Office of Energy Efficiency and Renewable Energy Wind Energy Technologies Office. The views expressed in the article do not necessarily represent the views of the DOE or the U.S. Government. The U.S. Government retains and the publisher, by accepting the article for publication, acknowledges that the U.S. Government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this work, or allow others to do so, for U.S. Government purposes.*

[NREL.gov](https://www.nrel.gov)

View NREL on Social Media

[Facebook](#) | [LinkedIn](#) | [Twitter](#)

---

[Unsubscribe from this list.](#)

[Learn more about NREL's security and privacy policies.](#)

NREL is a national laboratory of the U.S. Department of Energy, Office of Energy Efficiency and Renewable Energy, operated by the Alliance for Sustainable Energy, LLC.