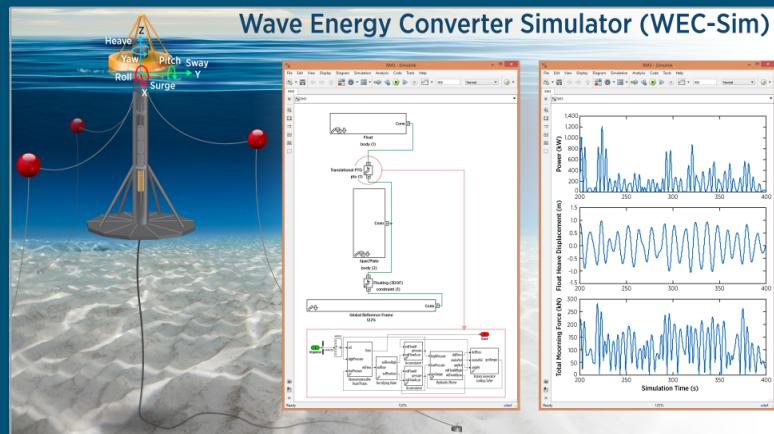




# Development and application of WEC-Sim v5.0 open-source software



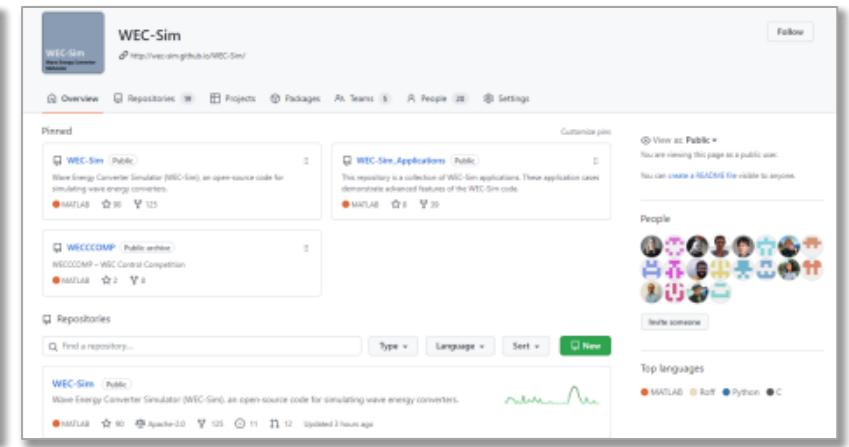
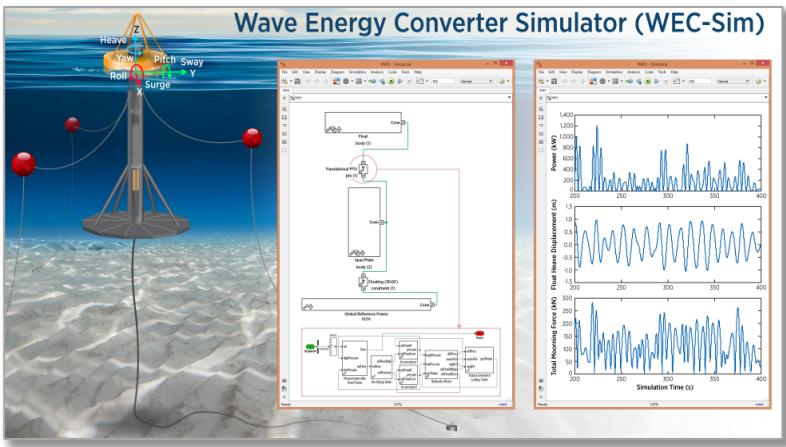
PRESENTED BY

Kelley Ruehl, Sandia National Laboratories



# Table of Contents

1. Background
2. WEC-Sim v5.0
3. New features
  - a) Continuous integration
  - b) Modeling cables
  - c) Capytaine BEM
  - d) Run from Simulink
  - e) Wave visualization
4. New applications cases
5. Future development



WEC-Sim v5.0

Search docs

Home

Introduction

Theory Manual

User Manual

Developer Manual

WEC-Sim (Wave Energy Converter SIMulator)

View page source

**WEC-Sim**  
Wave Energy Converter  
SIMulator

WEC-Sim (Wave Energy Converter SIMulator)

WEC-Sim (Wave Energy Converter SIMulator) is an open-source software for simulating wave energy converters. The software is developed in MATLAB/SIMULINK using the multi-body dynamics solver Simscape Multibody. WEC-Sim has the ability to model devices that are comprised of bodies, joints, power take-off systems, and mooring systems. WEC-Sim can model both rigid bodies and flexible bodies with generalized body modes. Simulations are performed in the time-domain by solving the governing wave energy converter equations of motion in the 6 Cartesian degrees-of-freedom, plus any number of user-defined modes. The [WEC-Sim Applications](#) repository



# Background

# What is WEC-Sim?

## WEC-Sim (Wave Energy Converter Simulator)

- Simulates wave energy converter dynamics
- Time-domain equation of motion solver based on Cummins' formulation
- Open source code developed in MATLAB/SIMULINK
  - Available at <https://github.com/WEC-Sim/WEC-Sim>
- Joint NREL/Sandia project
- Funded by the US Department of Energy
- First Release: v1.0 in June 2014
- Current Release: v5.0 in May 2022

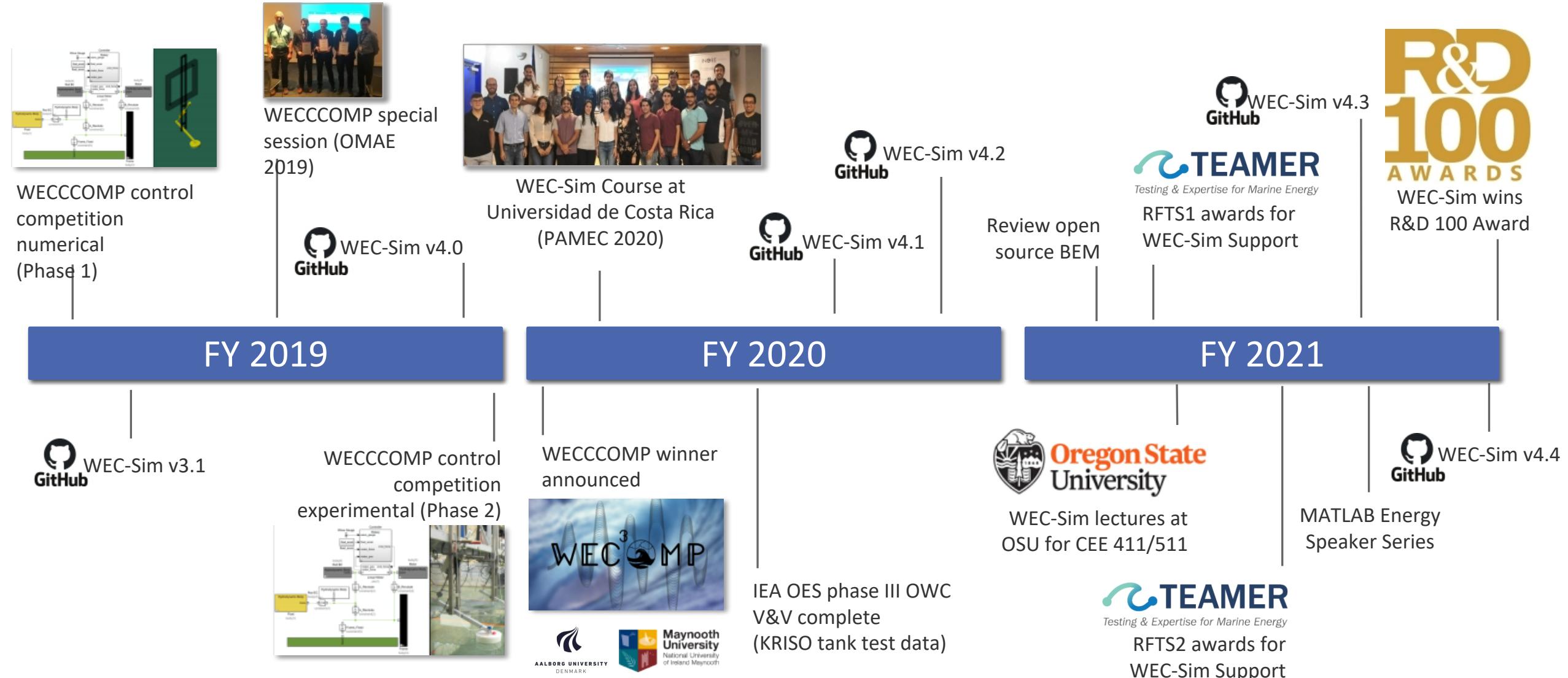


The image shows a screenshot of the WEC-Sim documentation website. The left side features a dark sidebar with a search bar and links to "Home", "Introduction", "Theory Manual", "User Manual", and "Developer Manual". The main content area has a light blue header with the text "WEC-Sim (Wave Energy Converter SIMulator)". Below the header is a diagram of a wave energy converter device. The text "View page source" is in the top right corner of the main content area.

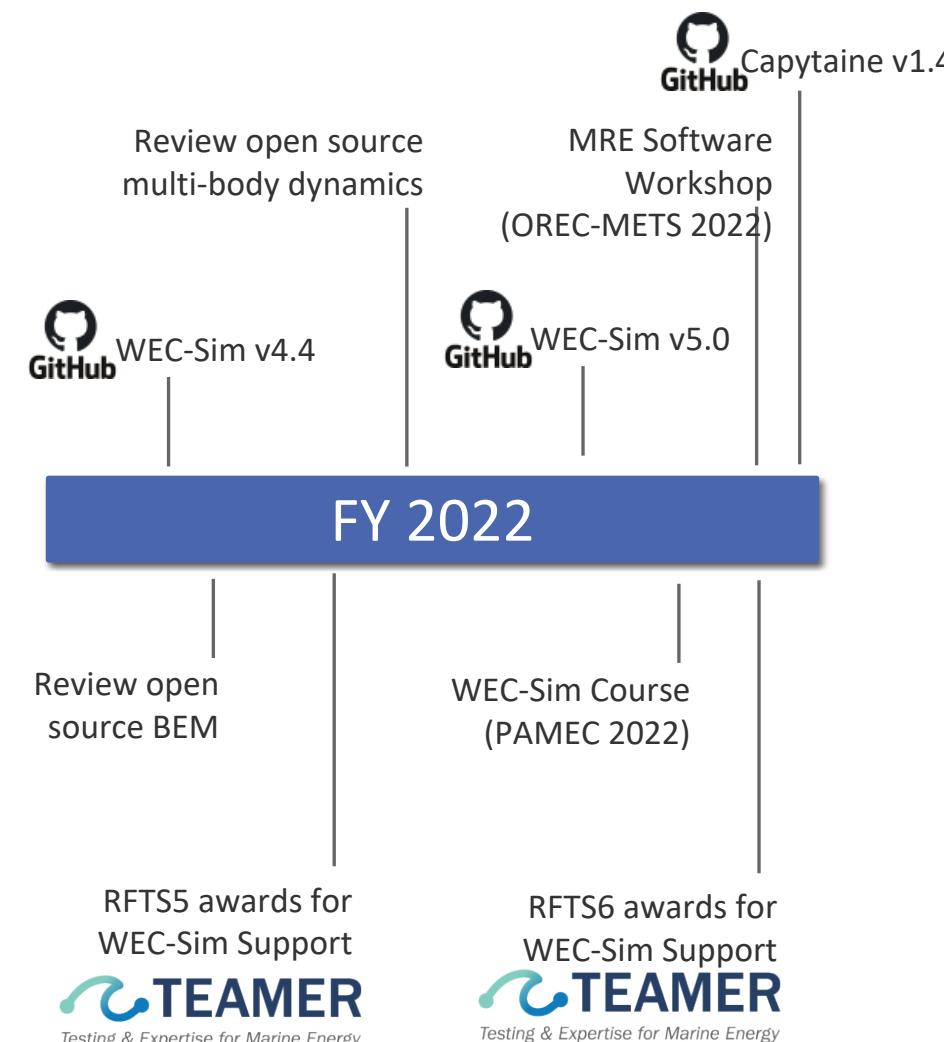
**WEC-Sim (Wave Energy Converter SIMulator)**

WEC-Sim (Wave Energy Converter SIMulator) is an open-source software for simulating wave energy converters. The software is developed in MATLAB/SIMULINK using the multi-body dynamics solver Simscape Multibody. WEC-Sim has the ability to model devices that are comprised of bodies, joints, power take-off systems, and mooring systems. WEC-Sim can model both rigid bodies and flexible bodies with generalized body modes. Simulations are performed in the time-domain by solving the governing wave energy converter equations of motion in the 6 Cartesian degrees-of-freedom, plus any number of user-defined modes. The [WEC-Sim Applications repository](#)

# Project Timeline



# FY22 Accomplishments



## FY22 Accomplishments

Support the WEC modeling community through WEC-Sim development, maintenance, support and training.

- **Develop and maintain** the WEC-Sim software on GitHub, and **resolve bugs** with the software
- Provide **WEC-Sim support** via responding to and resolving issues posted by users, and updating the publicly available examples and online documentation
- **Outreach** via hosting in-person and online training courses (e.g. PAMEC 2022), presenting and **publishing WEC-Sim articles**, and **training the next generation** of WEC numerical modelers

Assess the current MRE Software landscape, and identify future development needs.

- **Solicit public feedback** via MRE Software Workshop (e.g. OREC-METS 2022) and online webinar
- **Draft report** on current MRE Software landscape, identifying gaps and potential needs for public feedback



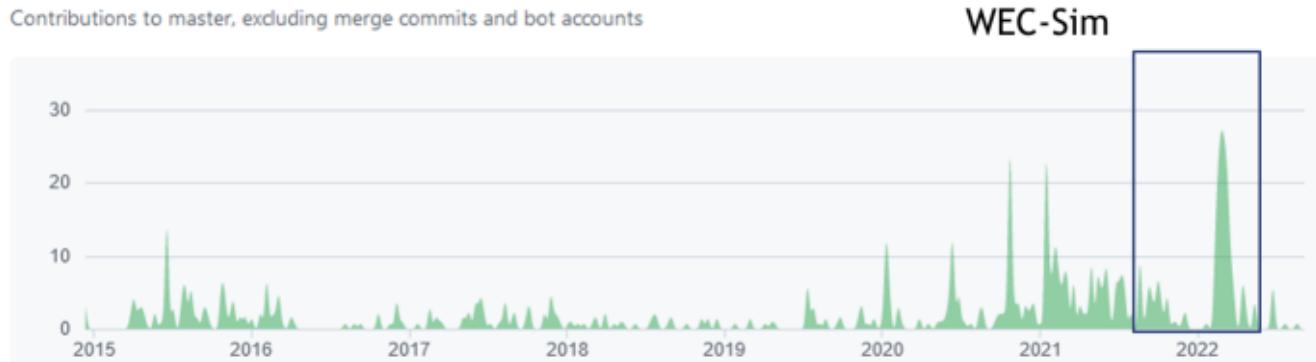
WEC-Sim v5.0

## 8 WEC-Sim v5.0

### WEC-Sim on GitHub: <https://github.com/WEC-Sim/WEC-Sim>

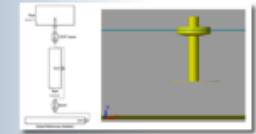
Dec 14, 2014 – Sep 7, 2022

Contributions to master, excluding merge commits and bot accounts



WEC-Sim (Wave Energy Converter SIMulator)

[View page source](#)



**WEC-Sim**  
Wave Energy Converter  
SIMulator

### WEC-Sim (Wave Energy Converter SIMulator)

WEC-Sim (Wave Energy Converter SIMulator) is an open-source software for simulating wave energy converters. The software is developed in MATLAB/SIMULINK using the multi-body dynamics solver Simscape Multibody. WEC-Sim has the ability to model devices that are comprised of bodies, joints, power take-off systems, and mooring systems. WEC-Sim can model both rigid bodies and flexible bodies with generalized body modes. Simulations are performed in the time-domain by solving the governing wave energy converter equations of motion in the 6 Cartesian degrees-of-freedom, plus any number of user-defined modes. The [WEC-Sim Applications repository](#) contains a wide variety of scenarios that WEC-Sim can be used to model, including desalination, mooring dynamics, nonlinear hydrodynamic bodies, passive yawing, batch simulations and many others. The software is very flexible and can be adapted to many scenarios within the wave energy industry.

### WEC-Sim Applications on GitHub: [https://github.com/WEC-Sim/WEC-Sim\\_Applications](https://github.com/WEC-Sim/WEC-Sim_Applications)

Mar 27, 2016 – Sep 7, 2022

Contributions to master, excluding merge commits and bot accounts



WEC-Sim (Wave Energy Converter SIMulator)

[View page source](#)

### WEC-Sim Developers

WEC-Sim is a collaboration between the [National Renewable Energy Laboratory \(NREL\)](#) and [Sandia National Laboratories \(Sandia\)](#), funded by the U.S. Department of Energy's Water Power Technologies Office. Due to the open source nature of the software, WEC-Sim has also had many external contributions. For more information refer to [Acknowledgements](#).

Current members of the development team include:

- Kelley Ruehl (Sandia - PI)
- David Ogden (NREL - PI)
- Nathan Tom (NREL)
- Dominik Forbush (Sandia)
- Adam Keester (Sandia)
- Jorge Leon (Sandia)
- Jeff Grasberger (Sandia)

Former members of the development team include:

- Yi-Hsiang Yu (NREL)
- Jennifer Van Rij (NREL)
- Michael Lawson (NREL)
- Carlos Michelen (Sandia)

[Next](#)

# 9 | Introduction to v5.0

v5.0, released May 16, 2022, represents a significant code overhaul and feature addition.

## Refactoring and SCM changes

- An intuitive and consistent naming convention applied throughout (not reverse compatible)
- Related functions grouped together, redundant functions removed.
- Embedded Simulink functions saved as separate scripts
- Updated and expanded documentation

## New features

- Continuous integration
- Modeling cables
- Capytaine BEM
- Run from Simulink
- Wave visualization

**WEC-Sim v5.0** Latest

kmruhli released this May 16, 2022 · 15 commits to master since this release · v5.0 · fcc5cd · Compare · Edit ·

### New Features

- Refactoring classes and properties @kmruhli in #803 #822 #828 #832, @akeeste in #838
- Refactoring docs by @kmruhli in #840
- Refactor BEMIO functions, tests, and documentation @akeeste in #790 #812, @HORSE in #839, @dav-og in #806
- Run from sim updates by @akeeste in #737
- Allow binary STL files by @akeeste in #760
- Update Read\_ACWA and AQWA examples by @jtgrasb in #761 #779 #797 #831
- Rename plotWaves by @jtgrasb in #765
- Update to normalize to handle sorting mean drift forces by @nathanmtom in #808 #809
- Remove passiveYawTest.m by @jtgrasb in #807
- Wave class wave gauge update by @nathanmtom in #801
- New pto sim lib by @jleonus in #821
- Warning/Error flags by @jtgrasb in #826
- Add Google Analytics 4 by @akeeste in #854

### Documentation

- Update WEC-Sim's Developer Documentation for the Morison Element Implementation by @nathanmtom in #796
- Update response class API by @akeeste in #802
- Doc\_auto\_gen\_masks by @salhus in #842
- Move documentation compilation to GitHub Actions by @HORSE in #817
- Add branch build in docs workflow for testing PRs by @HORSE in #834
- Update the WEC-Sim Theory Documentation to Clarify Wave Power Calculation by @nathanmtom in #847
- Update documentation on mean drift and current by @akeeste in #800

### Bug Fixes

- Fix cable library links. Resolves #770 by @akeeste in #774 #775
- Fix rate transition error by @akeeste in #799
- Fix cable implementation by @dforbrush2 in #827
- PTO-Sim bug fix by @jleonus in #833
- Bug fix for the regular wave power full expression by @nathanmtom in #841
- Fix documentation on dev branch by @HORSE in #816
- Bug fix: responseClass reading the MoorDyn Lines.out file too early, resolves #811 by @akeeste in #814

### Issues and Pull Requests

- > 52 issues closed since v4.4
- > 44 PRs merged since v4.4

Full Changelog: [v4.4...v5.0](#)

DOI: [10.5281/zenodo.6055137](#)

### Contributors

kmruhli, HORSE, and 7 other contributors

### Assets

- [Source code \(zip\)](#) May 11, 2022
- [Source code \(tar.gz\)](#) May 11, 2022

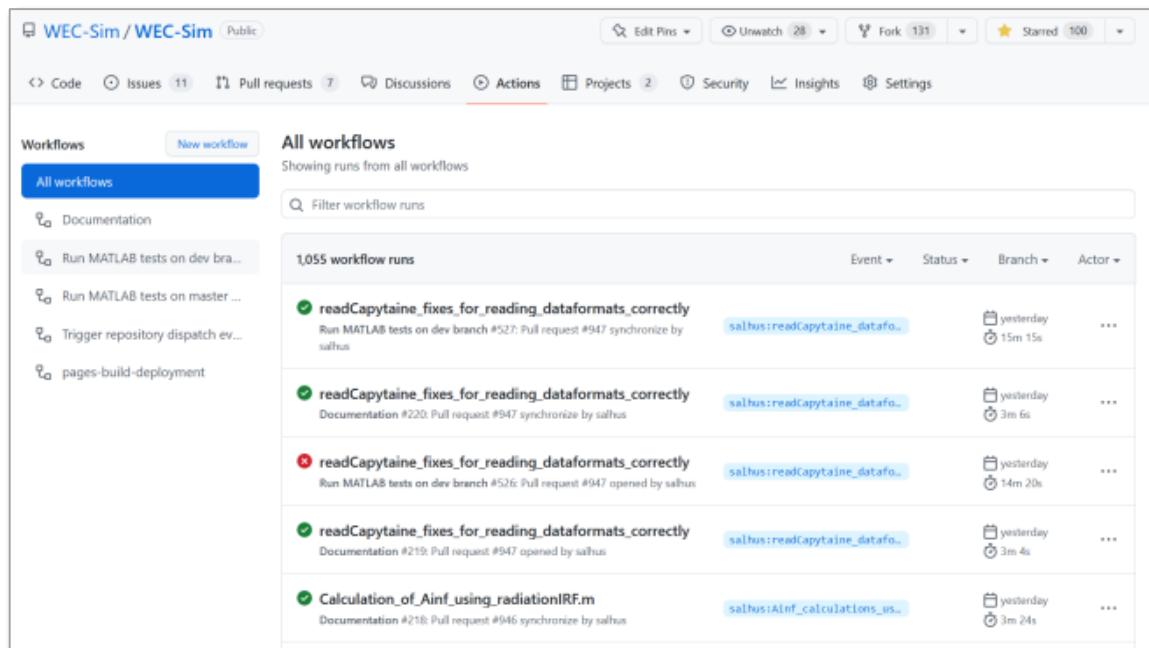
2 people reacted

# New Features - Continuous Integration

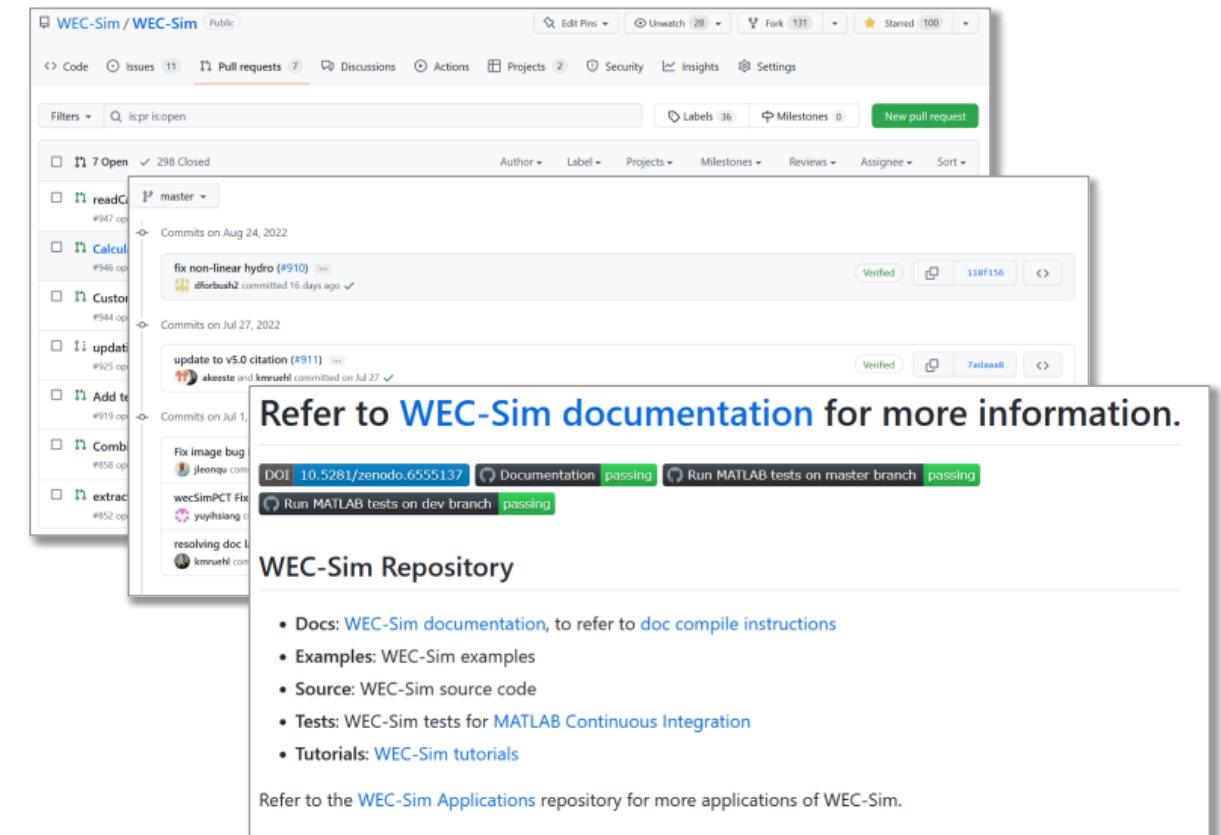
# New features – Continuous integration

## WEC-Sim tests developed for WEC-Sim source repository and WEC-Sim Applications

- Tests can be run locally when developing new features, to ensure stability of source code
- Tests automatically executed on both master and dev branches, on both repositories
- Tests automatically executed for documentation build
- Tests automatically for each commit and pull request
- Build reports automatically generated



The screenshot shows the GitHub repository for WEC-Sim. The 'Actions' tab is selected, displaying the 'All workflows' section. It lists 1,055 workflow runs, primarily for the 'readCapytaine\_fixes\_for\_reading\_dataformats\_correctly' workflow. The runs are filtered by status (green checkmark) and are grouped by commit. Each run shows the author, commit date, and duration. The repository has 11 issues, 7 pull requests, and 2 projects.



The screenshot shows the GitHub repository for WEC-Sim. A pull request for 'fix non-linear hydro (#910)' is open, showing commits from 'dforbush2' and 'akeeste' and 'kmrusch'. The pull request is marked as 'Verified'. Below the repository view, a 'Refer to WEC-Sim documentation for more information.' section is shown. It includes a DOI link (DOI: 10.5281/zendodo.6555137), a 'Documentation' button (green), and a 'Run MATLAB tests on master branch' button (green). The 'WEC-Sim Repository' section lists documentation, examples, source code, tests, and tutorials. The 'WEC-Sim Applications' section is also mentioned.

# New Features – Cable Modeling

## New features – Modeling cables

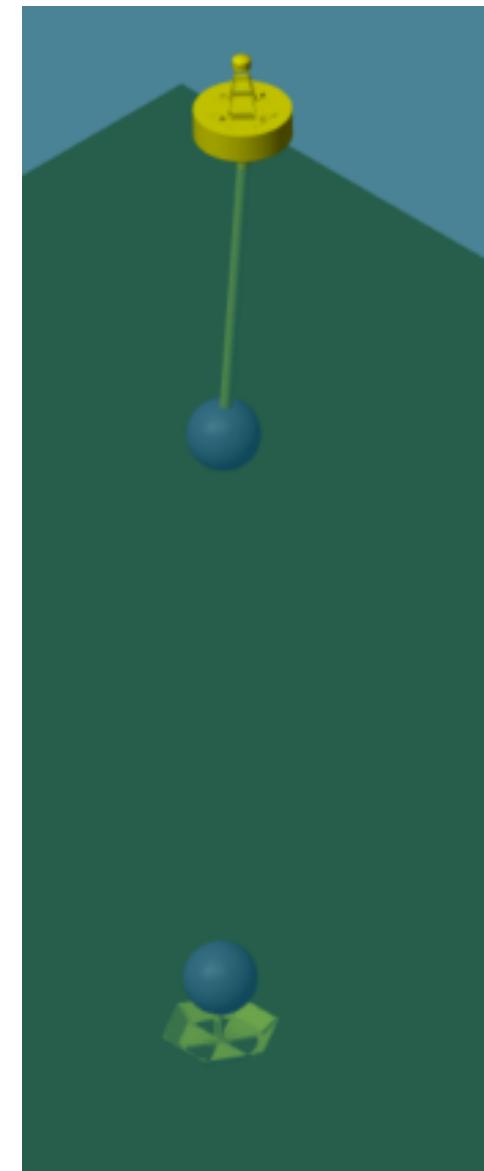
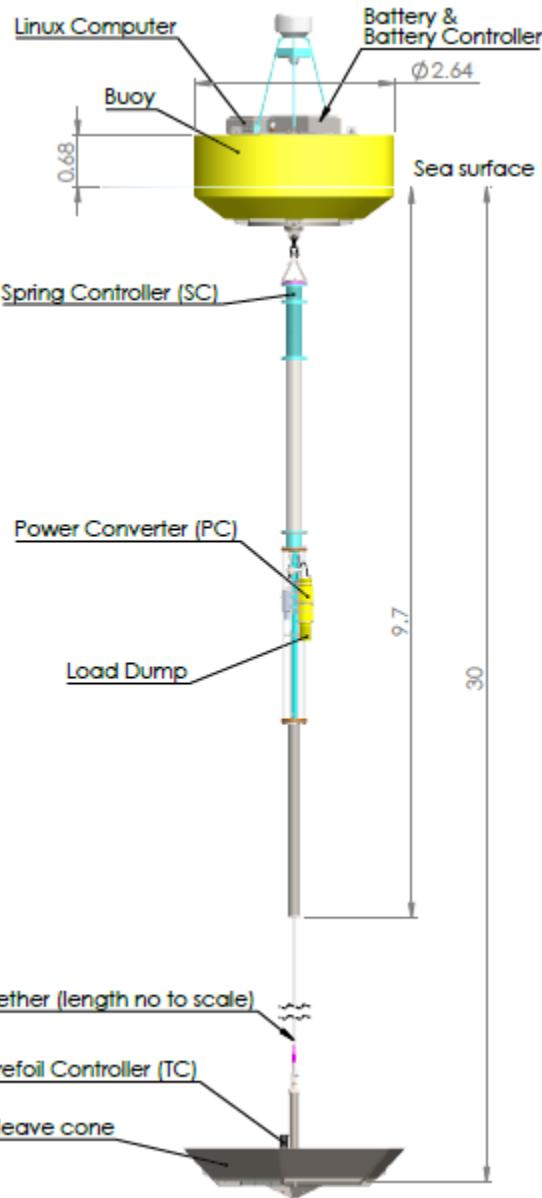
A number of industry WECs rely on cables/winches/tethers as a means of coupling two or more bodies: a frequent subject of user-posted issues.

These elements are non-linear as they only transmit tensile forces between the connection points .

At right: Schematic of the MBARI-WEC\*.

At far right: Mechanics Explorer visualization of the MBARI-WEC. The end-points of the cable connection shown as oversize grey spheres.

Image from Hamilton, A., Cazenave, F., Forbush, D. et al. The MBARI-WEC: a power source for ocean sensing. *J. Ocean Eng. Mar. Energy* 7, 189–200 (2021).  
<https://doi.org/10.1007/s40722-021-00197-9>

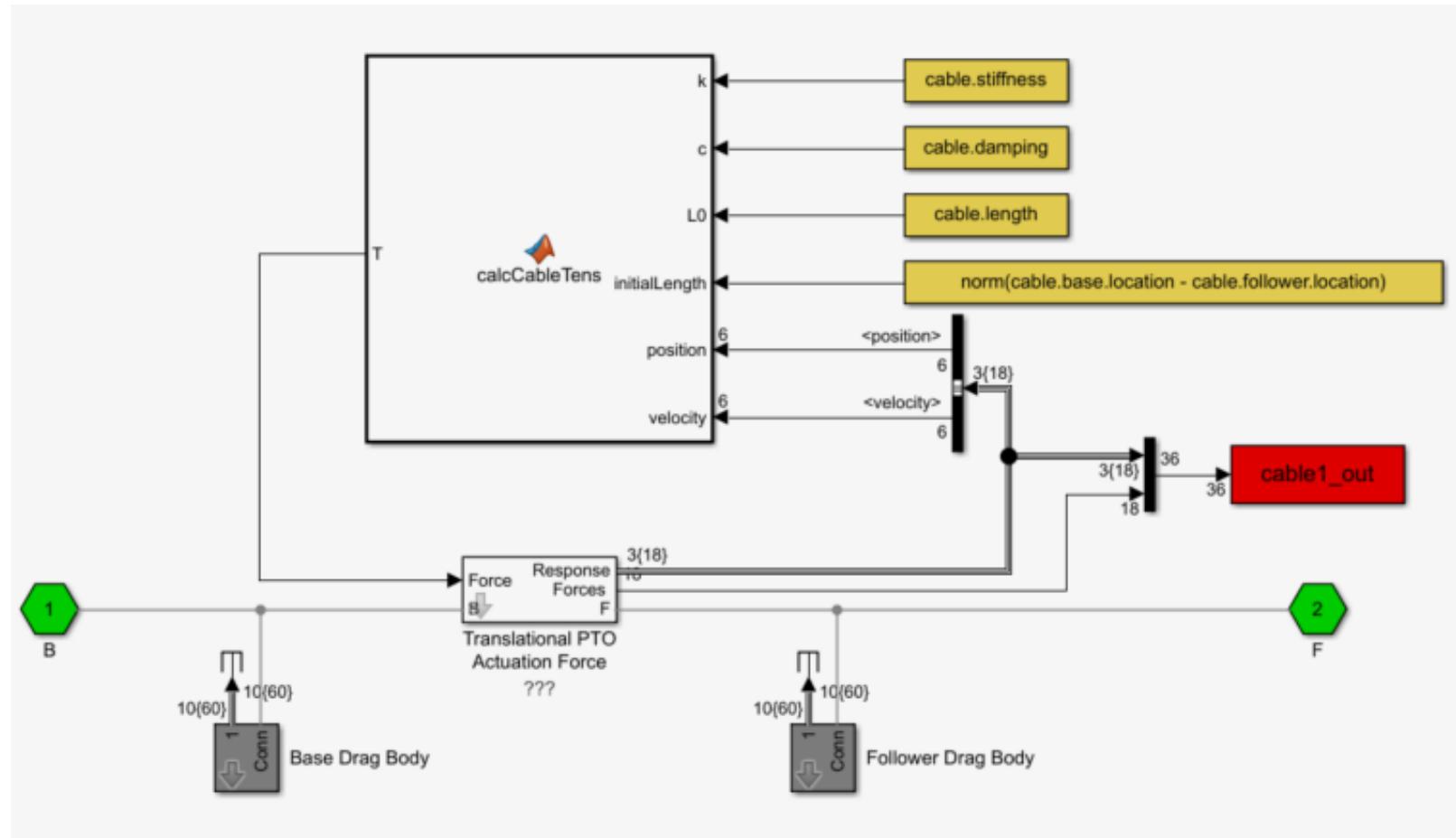


## New features – Modeling cables

Implemented as a logic check:  
 cable stiffness + damping force  
 exerted if extended beyond  
 equilibrium length, no force  
 exerted if not.

This does NOT resolve the  
 motions of the cable, but  
 transmits appropriate forces  
 between the coupled bodies.

Alternative modeling  
 suggestions available for stiff  
 systems.



# New Features – Capytaine BEM

## New features – Capytaine

Aug 6, 2017 – Sep 7, 2022

Contributions: Commits ▾

Contributions to master, excluding merge commits and bot accounts



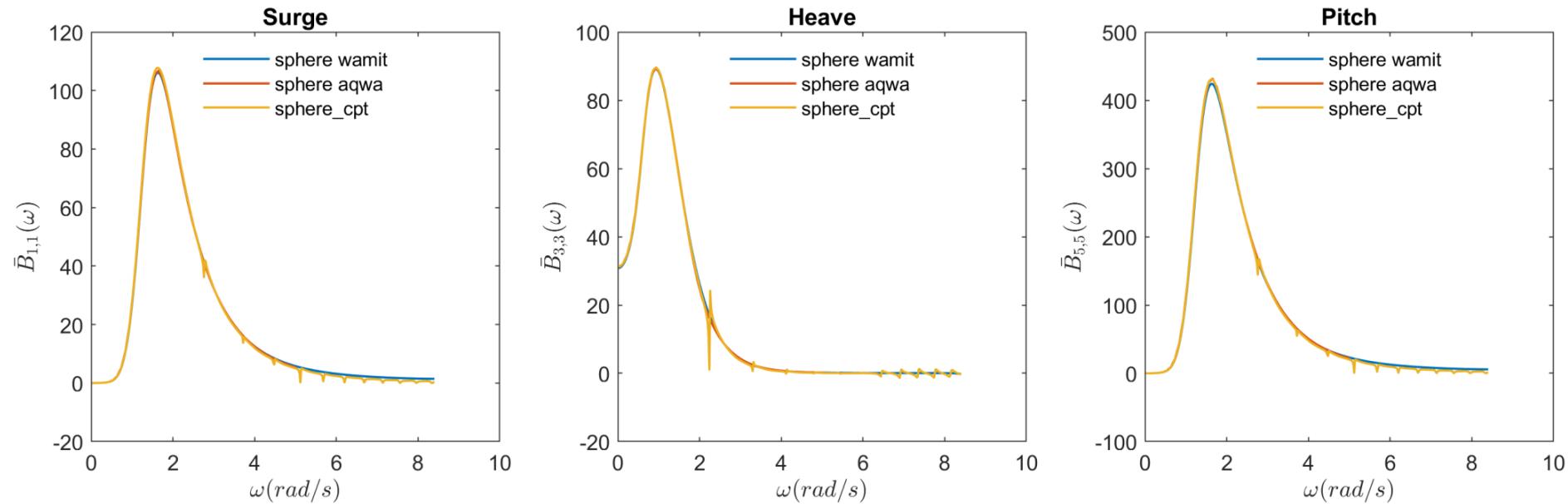
WEC-Sim relies on boundary element method solutions to define hydrodynamic coefficients.  
Supported codes are NEMOH, WAMIT, AQWA, and now Capytaine

Open-source Python code: <https://github.com/capytaine/capytaine>

Capytaine support is current and ongoing.

Capytaine developer: Matthieu Ancellin

## New features – Capytaine



WEC-Sim relies on boundary element method solutions to define hydrodynamic coefficients. Supported codes are NEMOH, WAMIT, AQWA, and now Capytaine

Open-source Python code: <https://github.com/captaine/captaine>

Captaine support is current and ongoing.

Captaine developer: Matthieu Ancellin

# New Features – Run from Simulink

## New features – Run from Simulink

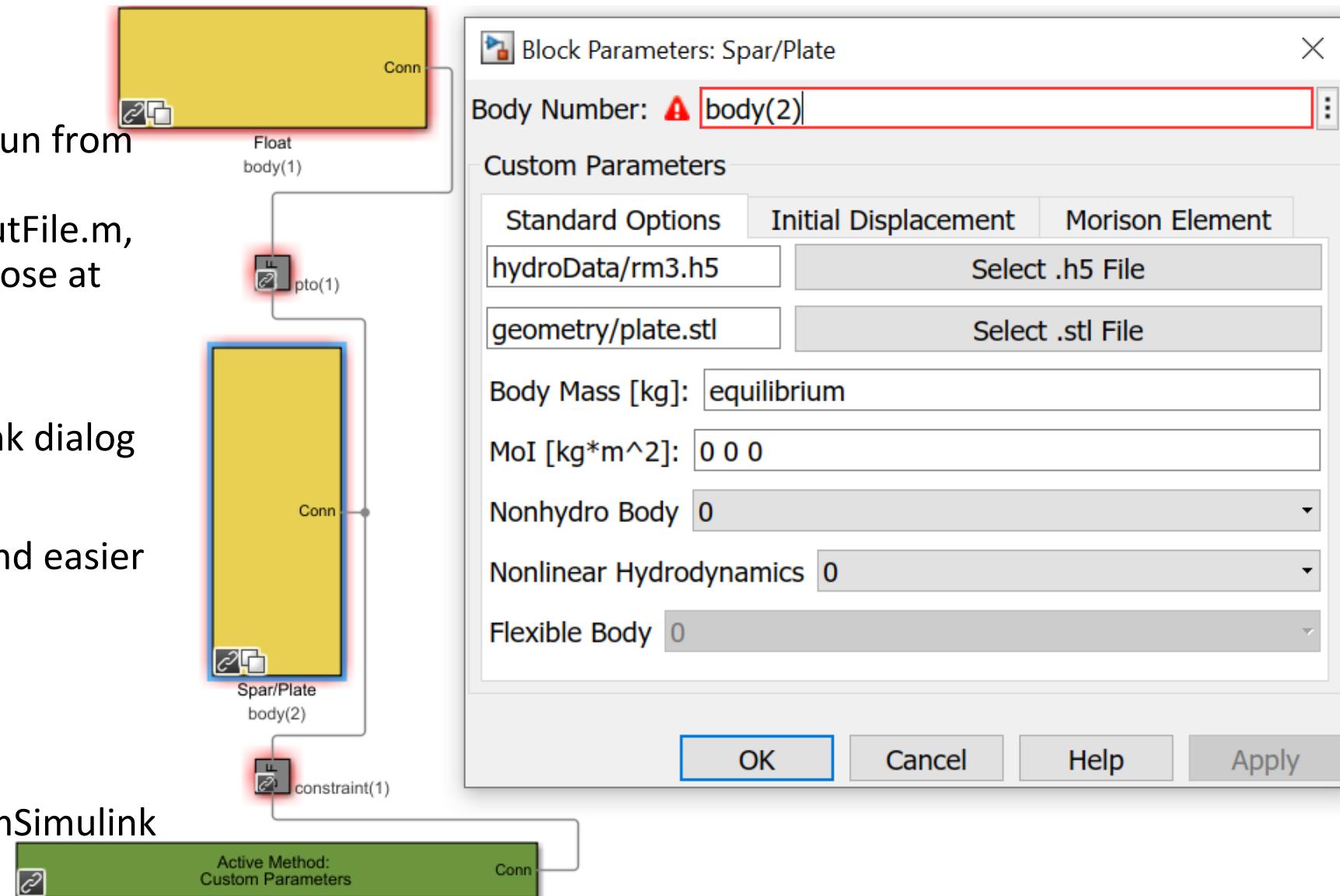
Previously, a simulation was run from the command line, reading information from `wecSimInputFile.m`, and Simulink models would close at run time.

Now, user option to specify parameters directly in Simulink dialog and run from GUI.

A more common workflow, and easier for debugging.

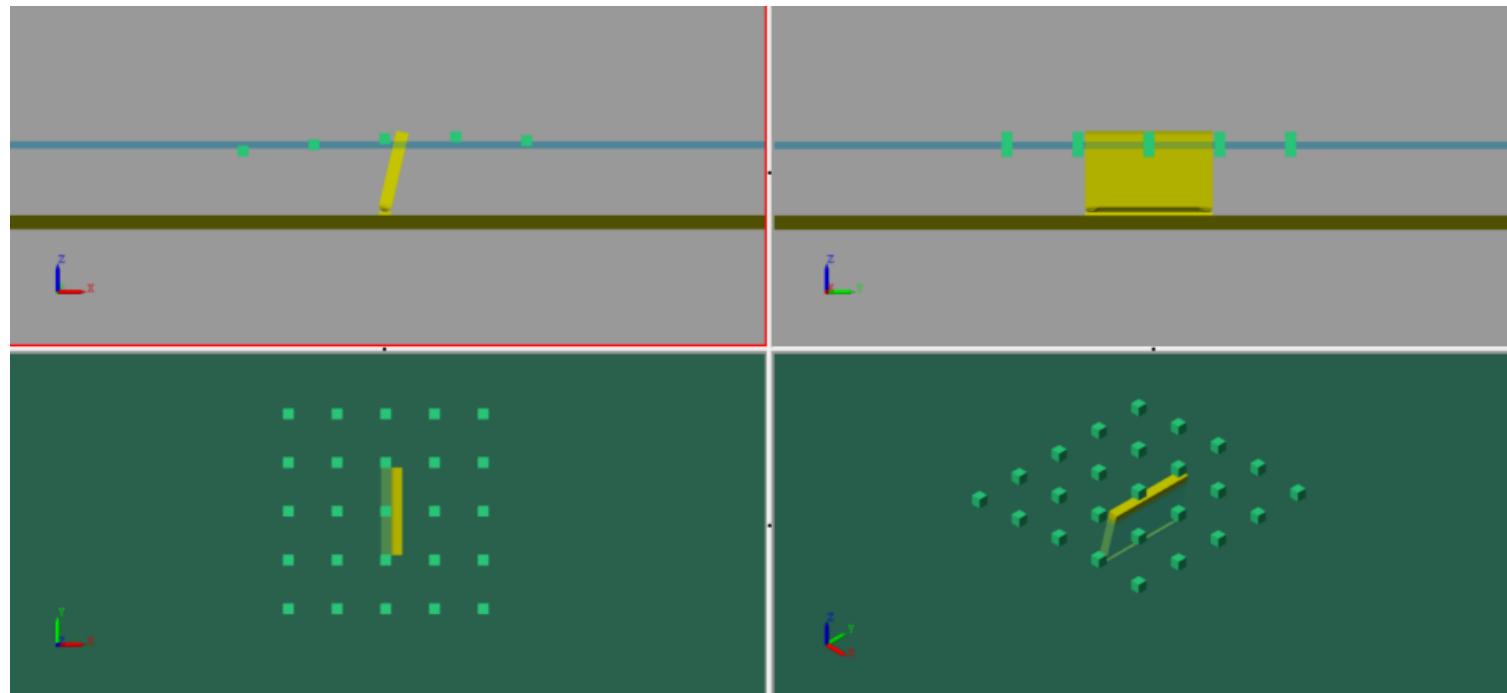
Example given:

`WEC-Sim/examples/RM3FromSimulink`



# New Features – Wave Visualization

## New features – Wave Visualization



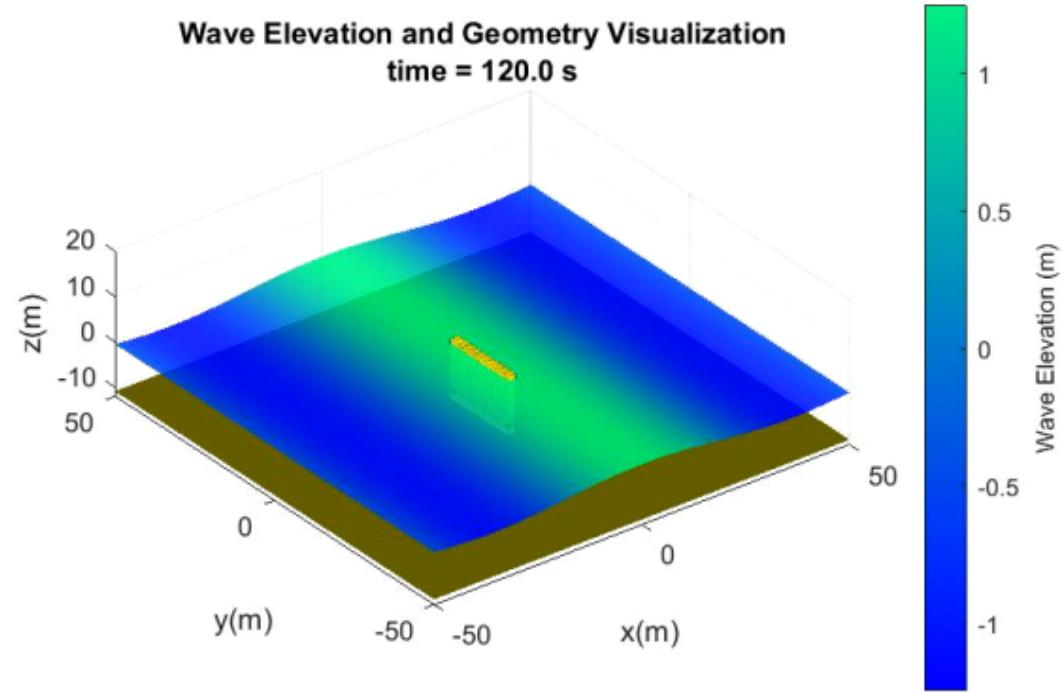
User-defined wave markers offer a simple way to aid free surface visualization in the native Mechanics Explorer. A useful debugging tool!

Videos files (\*.avi or \*.gif) can be created using `responseClass.saveViz()` function.

Example shown: WEC-Sim\_Applications/Wave\_Markers

ParaView (VTK) visualization is still supported as a visualization alternative.

## New features – Wave Visualization



User-defined wave markers offer a simple way to aid free surface visualization in the native Mechanics Explorer. A useful debugging tool!

Videos files (\*.avi or \*.gif) can be created using `responseClass.saveViz()` function.

Example shown: WEC-Sim\_Applications/Wave\_Markers

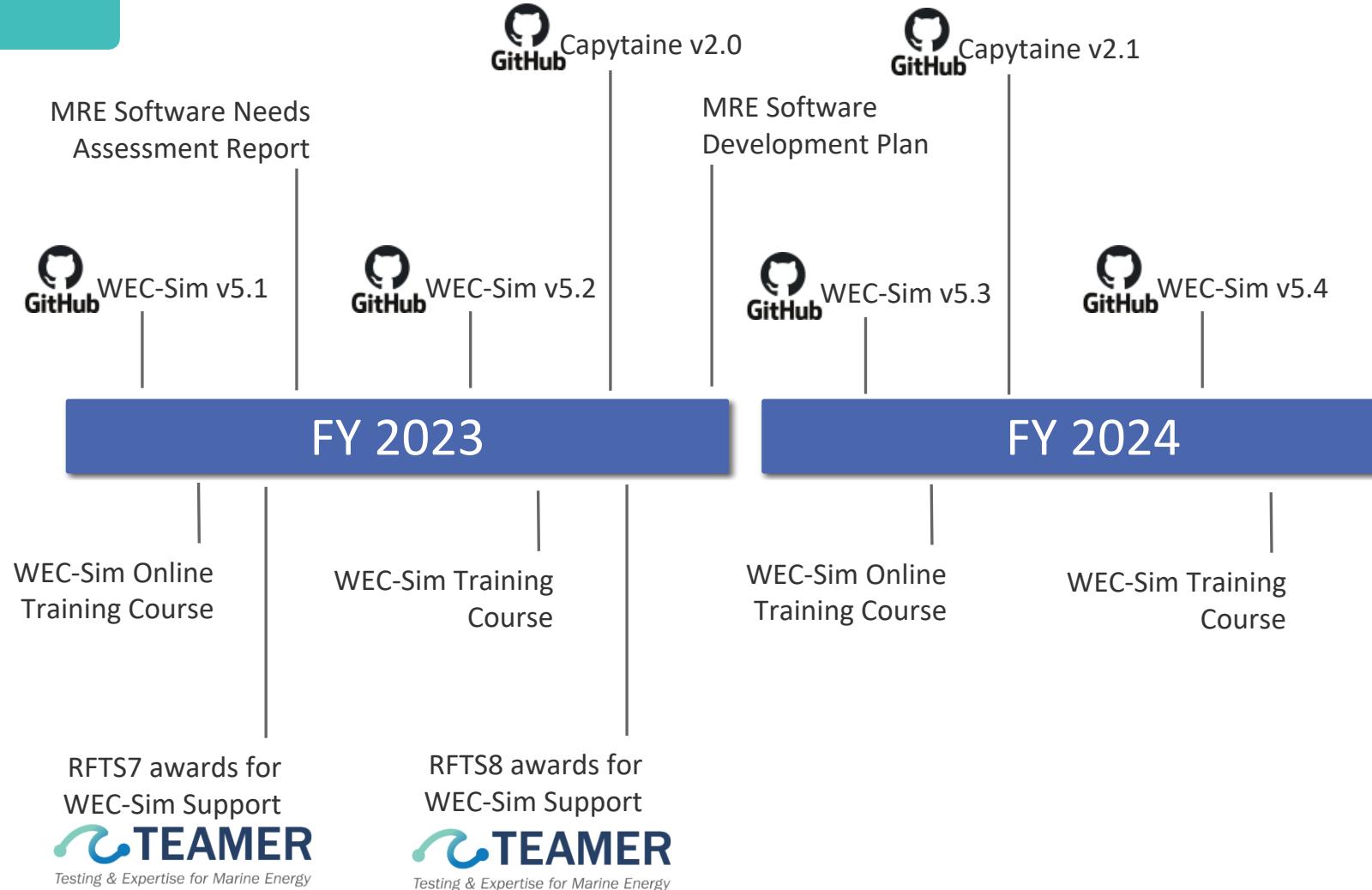
ParaView (VTK) visualization is still supported as a visualization alternative.

# Future Development

## Future Work

Advance the state of open source software within the wave energy sector

- Improve WEC-Sim **interoperability with open-source meshing, BEM and optimization** software to facilitate device performance improvements and cost reduction
- **Improve parallelization** to leverage HPC systems for scientific discovery
- **Support the development of the open-source BEM software Capytaine** – improving accuracy, speed and functionality.
- **Outreach and training**, including short courses, webinars, and additional WEC-Sim applications (e.g. offshore wind, flexible bodies)



## Conclusion

- User feedback guides development! Contributions are always welcomed.
- Report a bug, seek support, request a feature:  
<https://github.com/WEC-Sim/WEC-Sim/issues>
- Contribute to the WEC-Sim or WEC-Sim Applications code directly:  
<https://github.com/WEC-Sim/WEC-Sim/pulls>
- Direct industry/research support available through the TEAMER program:  
<https://teamer-us.org/>



For more information please visit the WEC-Sim website:

<http://wec-sim.github.io/WEC-Sim>

Kelley Ruehl (Sandia)  
Kelley.Ruehl@sandia.gov

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.

This work was authored in part 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 Water Power 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.

