WEC-Sim Training Course

Online Training Materials

PRESENTED BY

Jorge Leon, Sandia

BEMIO

What is BEMIO

Workflow: BEM → BEMIO → WEC-Sim

- The BEMIO (**B**oundary **E**lement **M**ethod **I**nput/**O**utput) functions are used to preprocess the BEM hydrodynamic data prior to running WEC-Sim.

Purpose

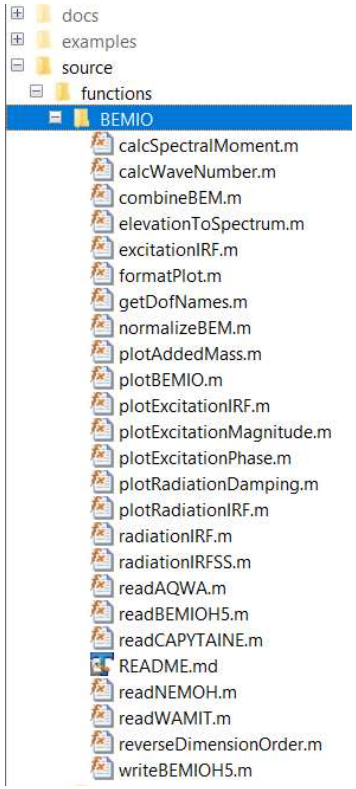
- Read BEM results from WAMIT, NEMOH, Capytaine, or AQWA.
- Calculate the radiation and excitation impulse response functions (IRFs).
- Calculate state space realization coefficients for the radiation IRF.
- Save the resulting data in Hierarchical Data Format 5 (HDF5).
- Plot typical hydrodynamic data for user verification.

Implementation

- Includes .h5 file, MATLAB (***hydro***) data structure

Locations

- Functions: ***\\WEC-Sim\source\functions\BEMIO***
- Documentation: https://wec-sim.github.io/WEC-Sim/master/user/advanced_features.html#bemio



readWAMIT Reads data from a WAMIT output file (*.out)

functions.BEMIO.readWAMIT(*hydro, filename, exCoeff*)

Reads data from a WAMIT output file.

If generalized body modes are used, the output directory must also include the **.cfg*, **.mmx*, and **.hst* files. If simu.nonlinearHydro = 3 will be used, the output directory must also include the **.3fk* and **.3sc* files.

See `WEC-Sim/examples/BEMIO/WAMIT` for examples of usage.

Parameters:

- **hydro** (`struct`) – Structure of hydro data that WAMIT input data will be appended to
- **filename** (`string`) – Path to the WAMIT output file
- **exCoeff** (`integer`) – Flag indicating the type of excitation force coefficients to read, 'diffraction' (default), 'haskind', or 'rao'

Returns: **hydro** – Structure of hydro data with WAMIT data appended

Return type: struct

bemio.m

```
hydro = struct();

hydro = readWAMIT(hydro,'rm3.out',[]);
hydro = radiationIRF(hydro,20,[],[],[],[]);
hydro = radiationIRFSS(hydro,[],[]);
hydro = excitationIRF(hydro,20,[],[],[],[]);
writeBEMIOH5(hydro)
plotBEMIO(hydro)
```

readNEMOH Reads data from a NEMOH working folder

functions.BEMIO.readNEMOH(*hydro, filedir*)

Reads data from a NEMOH working folder.

See `WEC-Sim\examples\BEMIO\NEMOH` for examples of usage.

Parameters:

- **hydro** (`struct`) – Structure of hydro data that NEMOH input data will be appended to
- **filename** (`string`) –
  Path to the NEMOH working folder, must include:
  - `Nemoh.cal`
  - `Mesh/Hydrostatics.dat` (or `Hydrostatiscs_0.dat` , `Hydrostatics_1.dat` , etc. for multiple bodies)
  - `Mesh/KH.dat (or ``KH_0.dat` , `KH_1.dat` , etc. for multiple bodies)
  - `Results/RadiationCoefficients.tec`
  - `Results/ExcitationForce.tec`
  - `Results/DiffractionForce.tec` - If simu.nonlinearHydro = 3 will be used
  - `Results/FKForce.tec` - If simu.nonlinearHydro = 3 will be used

Returns: **hydro** – Structure of hydro data with NEMOH data appended

Return type: struct

bemio.m

```
hydro = struct();

hydro = readNEMOH(hydro,'../RM3/');
% hydro = readWAMIT(hydro,'../../WAMIT/RM3/rm3.out',[]);
% hydro = combineBEM(hydro); % Compare WAMIT
hydro = radiationIRF(hydro,60,[],[],[],1.9);
hydro = radiationIRFSS(hydro,[],[]);
hydro = excitationIRF(hydro,157,[],[],[],1.9);
writeBEMIOH5(hydro)
plotBEMIO(hydro)
```

readAQWA Reads data from AQWA output files

functions.BEMIO.readAQWA(*hydro, ah1Filename, lisFilename*)

Reads data from AQWA output files.

See `WEC-Sim\examples\BEMIO\AQWA` for examples of usage.

Parameters:

- **hydro** (`struct`) – Structure of hydro data that Aqwa input data will be appended to
- **ah1Filename** (`string`) – .AH1 AQWA output file
- **lisFilename** (`string`) – .LIS AQWA output file

Returns: **hydro** – Structure of hydro data with Aqwa data appended

Return type: struct

bemio.m

```
hydro = struct();

hydro = readAQWA(hydro, 'RM3.AH1', 'RM3.LIS');
hydro = radiationIRF(hydro,150,[],[],[],1.8);
hydro = radiationIRFSS(hydro,[],[]);
hydro = excitationIRF(hydro,150,[],[],[],1.8);
writeBEMIOH5(hydro)
```

readCAPYTAINE Reads data from a Capytaine netcdf file

functions.BEMIO.readCAPYTAINE(*hydro, filename*)

Reads data from a Capytaine netcdf file

See `WEC-Sim\examples\BEMIO\CAPYTAINE` for examples of usage.

bemio.m

```
hydro = struct();

hydro = readCAPYTAINE(hydro,'rm3_full.nc');
hydro = radiationIRF(hydro,60,[],[],[],1.9);
hydro = radiationIRFSS(hydro,[],[]);
hydro = excitationIRF(hydro,157,[],[],[],1.9);
writeBEMIOH5(hydro)
plotBEMIO(hydro)
```

Parameters:

- **hydro** (`struct`) – Structure of hydro data that Capytaine input data will be appended to
- **filename** (`string`) – Capytaine .nc output file

Returns: **hydro** – Structure of hydro data with Capytaine data appended

Return type: struct

combineBEM Combines multiple BEM outputs into one hydrodynamic 'system'

functions.BEMIO.combineBEM(*hydro*)

Combines multiple BEM outputs into one hydrodynamic 'system.' This function requires that all BEM outputs have the same water depth, wave frequencies, and wave headings. This function would be implemented following multiple readWAMIT, readNEMOH, readCAPYTAINE, or readAQWA and before radiationIRF, radiationIRFSS, excitationIRF, writeBEMIOH5, or plotBEMIO function calls.

See `WEC-Sim\examples\BEMIO\NEMOH` for examples of usage.

Parameters: **hydro** (`[1 x n] struct`) – Structures of hydro data that will be combined into a single structure

Returns: **hydro** – Combined structure.

Return type: [1 x 1] struct

bemio.m

```
hydro = struct();

hydro = readNEMOH(hydro,'../RM3/');
hydro = readWAMIT(hydro,'../../WAMIT/RM3/rm3.out',[]);
hydro = combineBEM(hydro); % Compare WAMIT
hydro = radiationIRF(hydro,60,[],[],[],1.9);
hydro = radiationIRFSS(hydro,[],[]);
hydro = excitationIRF(hydro,157,[],[],[],1.9);
writeBEMIOH5(hydro)
plotBEMIO(hydro)
```

radiationIRF Calculates the normalized radiation impulse response function

functions.BEMIO.radiationIRF(*hydro, tEnd, nDt, nDw, wMin, wMax*)

Calculates the normalized radiation impulse response function. This is equivalent to the radiation IRF in the theory section normalized by ρ:

$$\overline{K}_{r,i,j}(t) = \frac{2}{\pi}\int_0^\infty \frac{B_{i,j}(\omega)}{\rho}\cos(\omega t)d\omega$$

Default parameters can be used by inputting []. See `WEC-Sim\examples\BEMIO` for examples of usage.

Parameters:

- **hydro** (`struct`) – Structure of hydro data
- **tEnd** (`float`) – Calculation range for the IRF, where the IRF is calculated from t = 0 to tEnd, and the default is 100 s
- **nDt** (`float`) – Number of time steps in the IRF, the default is 1001
- **nDw** (`float`) – Number of frequency steps used in the IRF calculation (hydrodynamic coefficients are interpolated to correspond), the default is 1001
- **wMin** (`float`) – Minimum frequency to use in the IRF calculation, the default is the minimum frequency from the BEM data
- **wMax** (`float`) – Maximum frequency to use in the IRF calculation, the default is the maximum frequency from the BEM data

Returns: **hydro** – Structure of hydro data with radiation IRF

Return type: struct

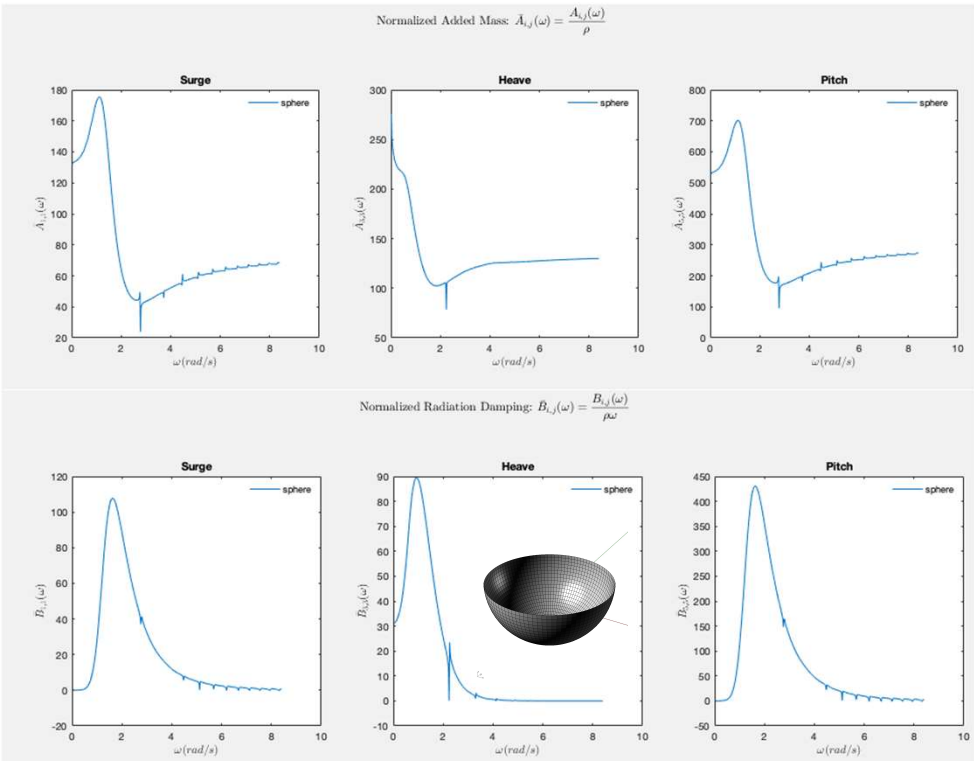
bemio.m

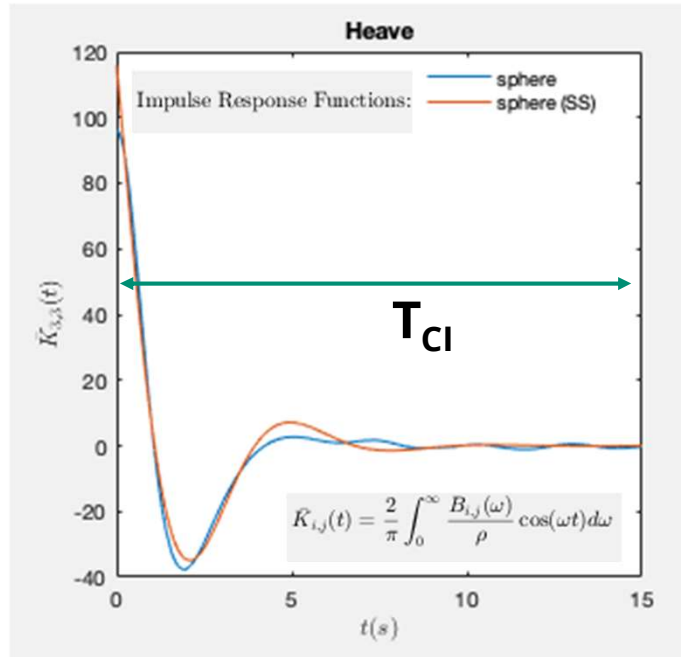
```
hydro = struct();

hydro = readWAMIT(hydro,'rm3.out',[]);
hydro = radiationIRF(hydro,20,[],[],[],[]);
hydro = radiationIRFSS(hydro,[],[]);
hydro = excitationIRF(hydro,20,[],[],[],[]);
writeBEMIOH5(hydro)
plotBEMIO(hydro)
```

$$\overline{K}_{i,j}(t) = \frac{2}{\pi}\int_0^\infty \frac{B_{i,j}(\omega)}{\rho}\cos(\omega t)\, d\omega$$

radiationIRF Calculates the normalized radiation impulse response function





NOTE: Make sure *simu.cicEndTime* <= T_{CI}

radiationIRFSS Calculates the state space (SS) realization of the radiation IRF

functions.BEMIO.radiationIRFSS(*hydro, Omax, R2t*)

Calculates the state space (SS) realization of the normalized radiation IRF. If this function is used, it must be implemented after the radiationIRF function.

Default parameters can be used by inputting []. See `WEC-Sim\examples\BEMIO` for examples of usage.

Parameters:

- **hydro** (`struct`) – Structure of hydro data
- **Omax** (`integer`) – Maximum order of the SS realization, the default is 10
- **R2t** (`float`) – R^2 threshold (coefficient of determination) for the SS realization, where R^2 may range from 0 to 1, and the default is 0.95

Returns: **hydro** – Structure of hydro data with radiation IRF state space coefficients

Return type: struct

bemio.m

```
hydro = struct();

hydro = readWAMIT(hydro,'rm3.out',[]);
hydro = radiationIRF(hydro,20,[],[],[],[]);
hydro = radiationIRFSS(hydro,[],[]);
hydro = excitationIRF(hydro,20,[],[],[],[]);
writeBEMIOH5(hydro)
plotBEMIO(hydro)
```

excitationIRF Calculates the excitation impulse response function

functions.BEMIO.excitationIRF(*hydro, tEnd, nDt, nDw, wMin, wMax*)

Calculates the normalized excitation impulse response function:

$$\overline{K}_{e,i,\theta}(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty}\frac{X_i(\omega,\theta)e^{i\omega t}}{\rho g}d\omega$$

Default parameters can be used by inputting []. See `WEC-Sim\examples\BEMIO` for examples of usage.

Parameters:

- **hydro** (`struct`) – Structure of hydro data
- **tEnd** (`float`) – Calculation range for the IRF, where the IRF is calculated from t = 0 to tEnd, and the default is 100 s
- **nDt** (`float`) – Number of time steps in the IRF, the default is 1001
- **nDw** (`float`) – Number of frequency steps used in the IRF calculation (hydrodynamic coefficients are interpolated to correspond), the default is 1001
- **wMin** (`float`) – Minimum frequency to use in the IRF calculation, the default is the minimum frequency from the BEM data
- **wMax** (`float`) – Maximum frequency to use in the IRF calculation, the default is the maximum frequency from the BEM data

Returns: **hydro** – Structure of hydro data with excitation IRF

Return type: struct

bemio.m

```
hydro = struct();

hydro = readWAMIT(hydro,'rm3.out',[]);
hydro = radiationIRF(hydro,20,[],[],[],[]);
hydro = radiationIRFSS(hydro,[],[]);
hydro = excitationIRF(hydro,20,[],[],[],[]);
writeBEMIOH5(hydro)
plotBEMIO(hydro)
```

$$\overline{K}_i(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty}\frac{X_i(\omega,\beta)}{\rho g}e^{i\omega t}d\omega$$

writeBEMIOH5 Writes the hydro data structure to a .h5 file.

functions.BEMIO.writeBEMIOH5(*hydro*)

Writes the hydro data structure to a .h5 file.

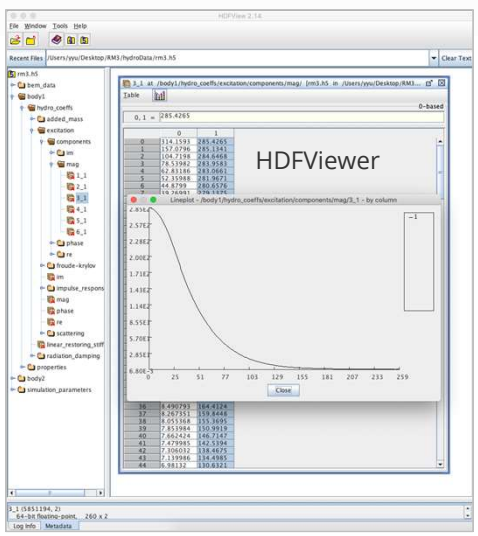
See `WEC-Sim\tutorials\BEMIO` for examples of usage.

Parameters: **hydro** (`[1 x 1] struct`) – Structure of hydro data that is written to `hydro.file`

bemio.m

```
hydro = struct();

hydro = readWAMIT(hydro,'rm3.out',[]);
hydro = radiationIRF(hydro,20,[],[],[],[]);
hydro = radiationIRFSS(hydro,[],[]);
hydro = excitationIRF(hydro,20,[],[],[],[]);
writeBEMIOH5(hydro)
plotBEMIO(hydro)
```



plotBEMIO Plots the hydrodynamic data

functions.BEMIO.plotBEMIO(*varargin*)

Plots the added mass, radiation damping, radiation IRF, excitation force magnitude, excitation force phase, and excitation IRF for each body in the heave, surge and pitch degrees of freedom.

Usage: `plotBEMIO(hydro, hydro2, hydro3, ...)`

See `WEC-Sim\examples\BEMIO` for additional examples.

bemio.m

```
hydro = struct();

hydro = readWAMIT(hydro,'rm3.out',[]);
hydro = radiationIRF(hydro,20,[],[],[],[]);
hydro = radiationIRFSS(hydro,[],[]);
hydro = excitationIRF(hydro,20,[],[],[],[]);
writeBEMIOH5(hydro)
plotBEMIO(hydro)
```

Parameters: **varargin** (`struct(s)`) – The hydroData structure(s) created by the other BEMIO functions. One or more may be input.

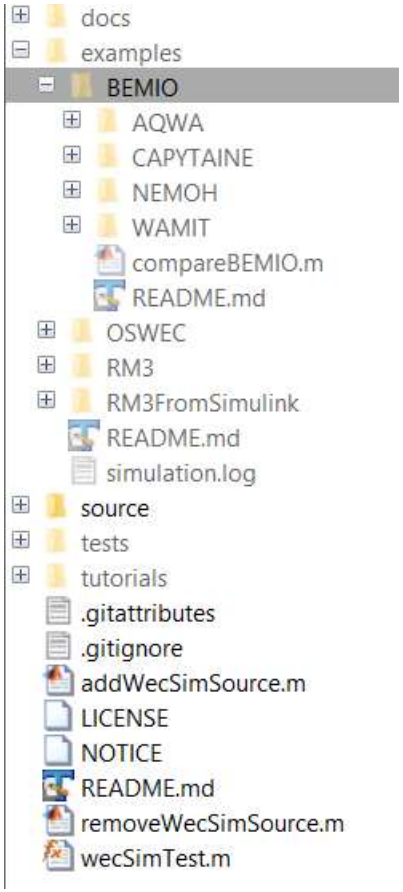
Examples and Usage

BEMIO tutorials in \WEC-Sim\examples\BEMIO

- WAMIT
- NEMOH
- Aqwa
- Capytaine
- compareBEMIO

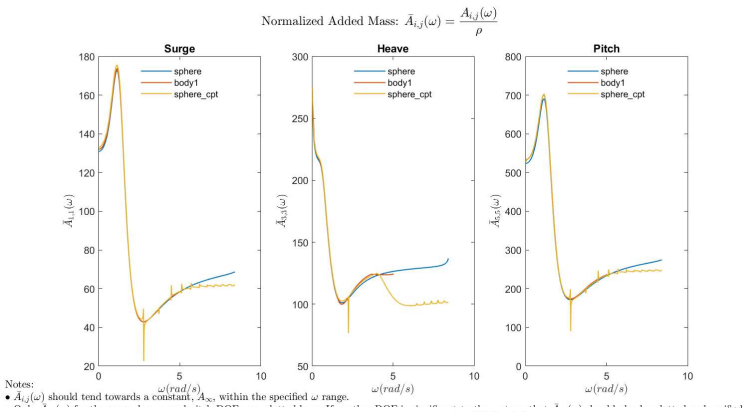
Data structures

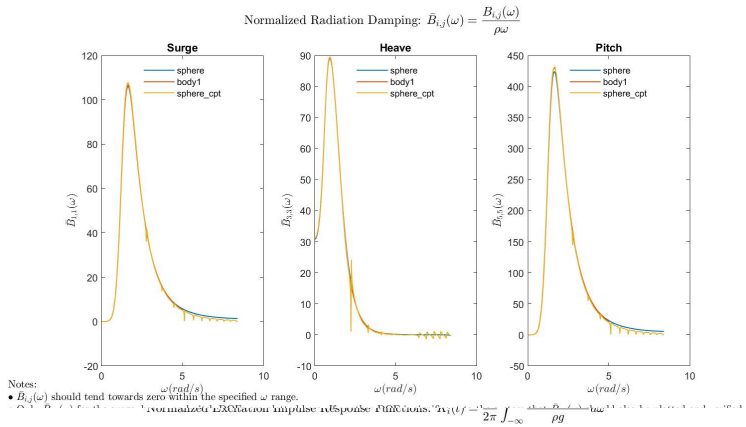
- BEMIO
- https://wec-sim.github.io/WEC-Sim/advanced_features.html#bemio
- .h5
- HDFVIEW: https://support.hdfgroup.org/products/java/hdfview/

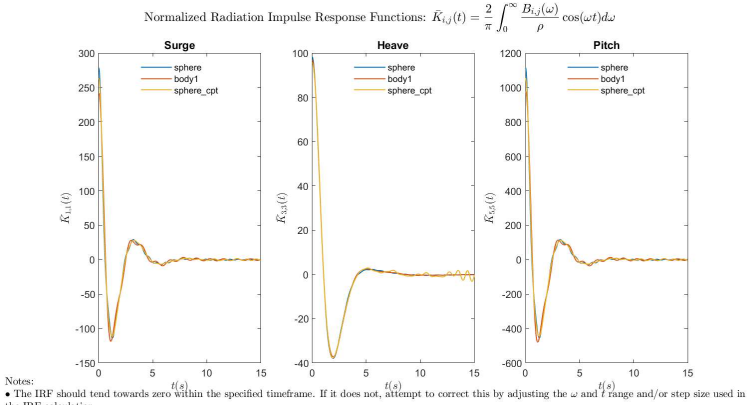


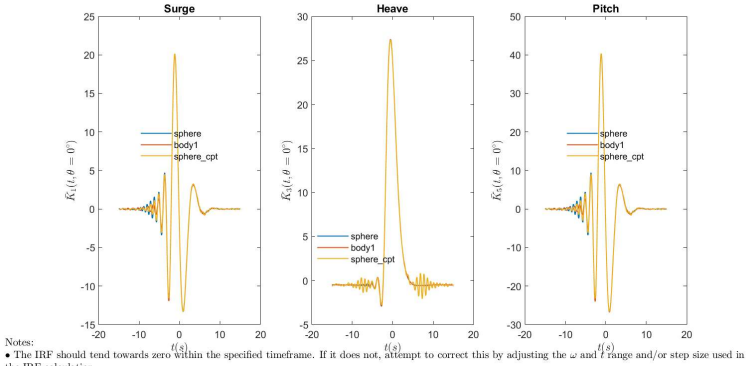
compareBEMIO

Sphere comparison available in: ***\\WEC-Sim\examples\BEMIO***









BEMIO Tutorial Practice

You can go to this folder in WEC-Sim to follow along:

\\WEC-Sim\examples\BEMIO\WAMIT\RM3

wecSimInputFile.m | bemio.m

```
hydro = struct();

hydro = readWAMIT(hydro,'rm3.out',[]);
hydro = radiationIRF(hydro,20,[],[],[],[2]);
hydro = radiationIRFSS(hydro,[],[]);
hydro = excitationIRF(hydro,20,[],[],[],[]);
writeBEMIOH5(hydro)
plotBEMIO(hydro)
```

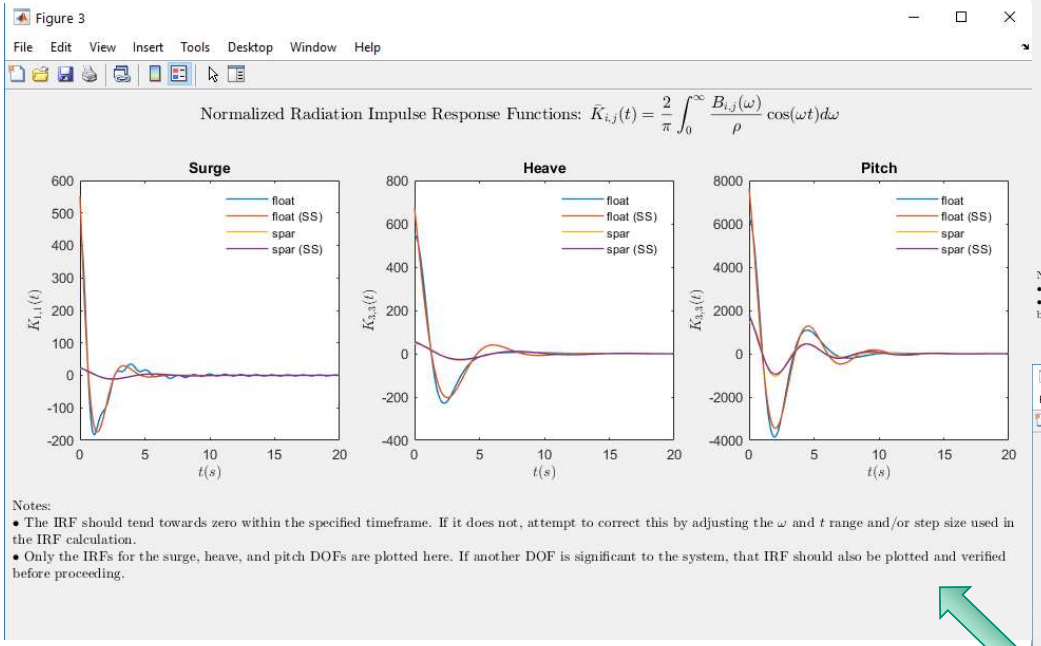
State Space Representation of IRF

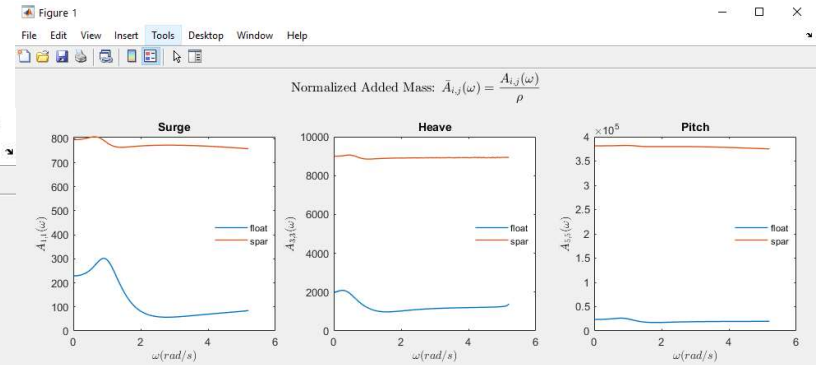
It is desirable to represent the radiation convolution integral in state space form. This has been shown to dramatically increase computational speeds and allow utilization of conventional control methods that rely on linear state space models.

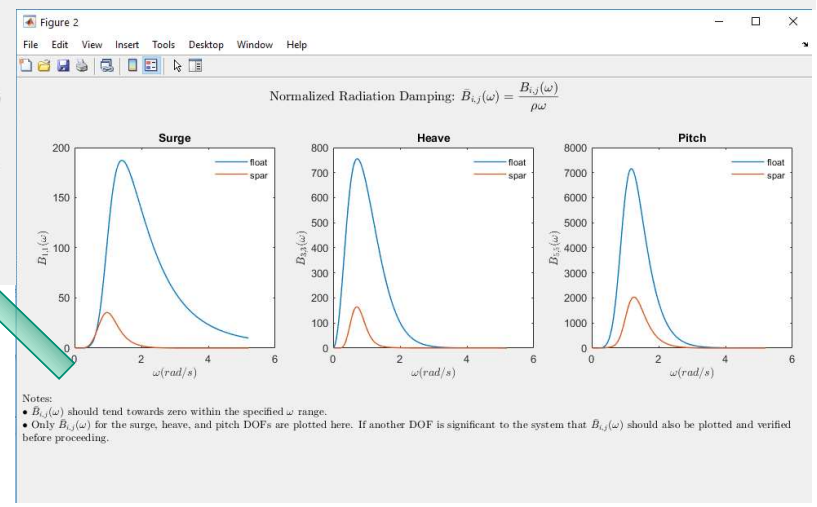
$$\int_0^t K_r(t-\tau)u(\tau)d\tau \quad \approx \quad \begin{array}{c} \dot{X}_r(t) = A_r X_r(t) + B_r u(t) \\ C_r X_r(t) + D_r u(t);\ X_r(0) = 0 \end{array}$$

An approximation will be made as K_r is solved from a set of partial differential equations where as a linear state space is constructed from a set of ordinary differential equations.

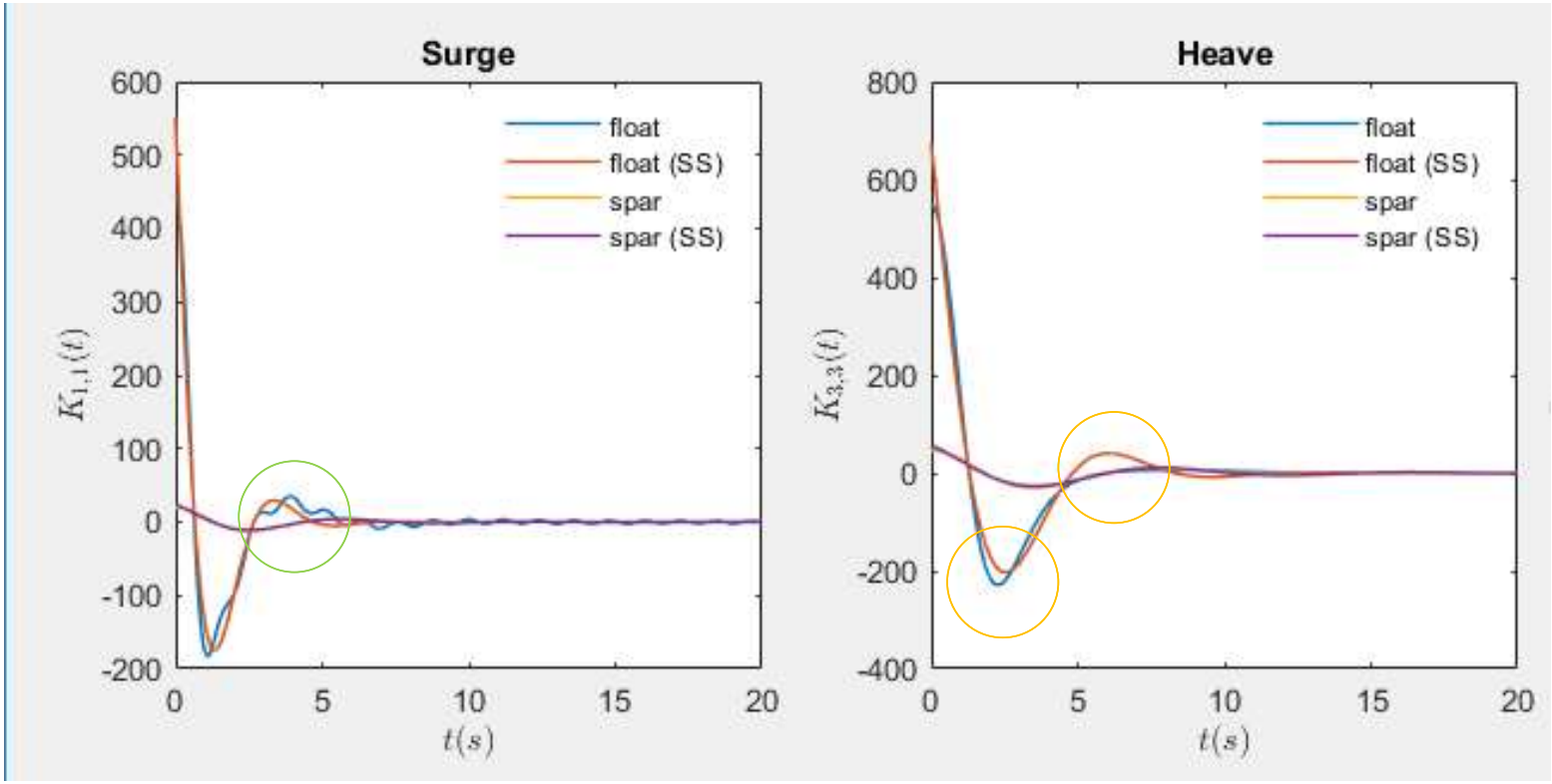
BEMIO Tutorial Practice





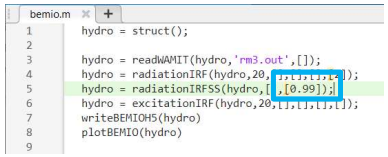


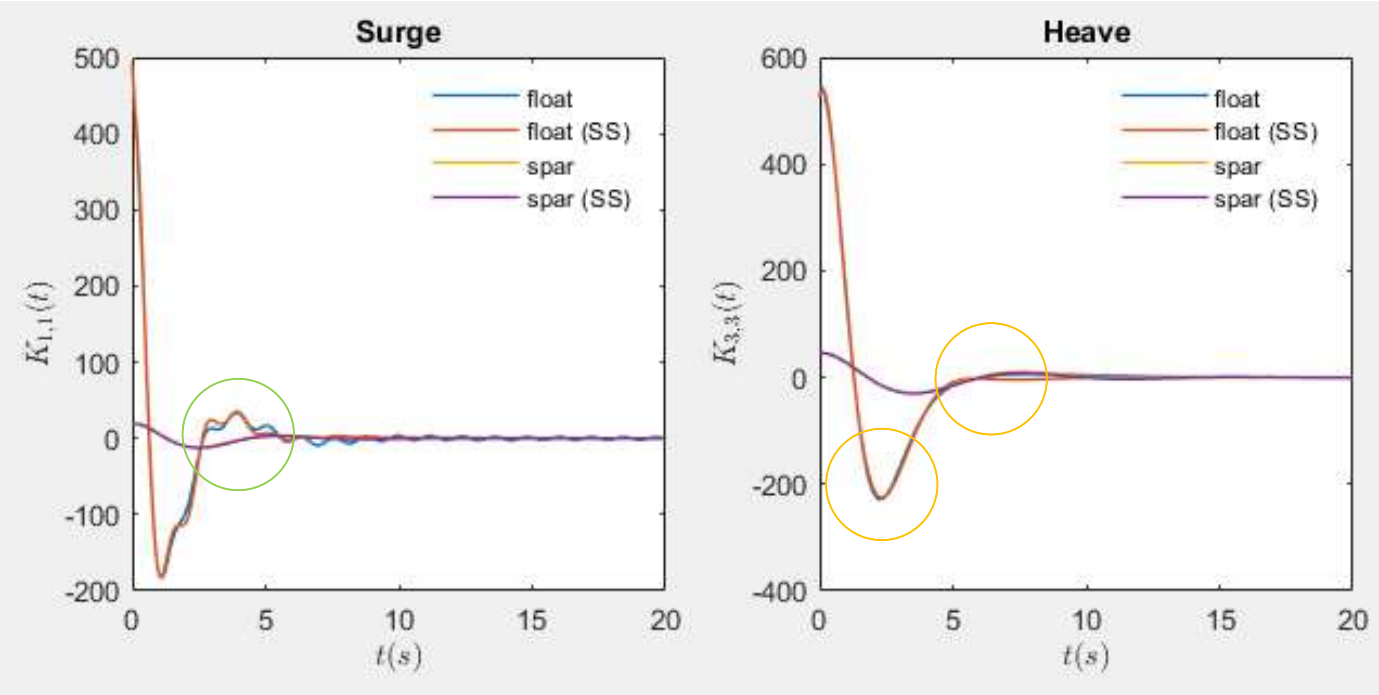
BEMIO Tutorial Practice



BEMIO Tutorial Practice

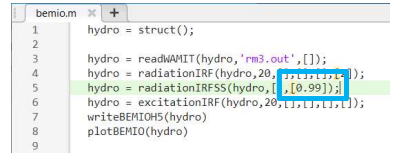
Let's increase the R2 threshold to 0.99
hydro = radiationIRFSS(hydro, [ ], [0.99])
Default is 0.95

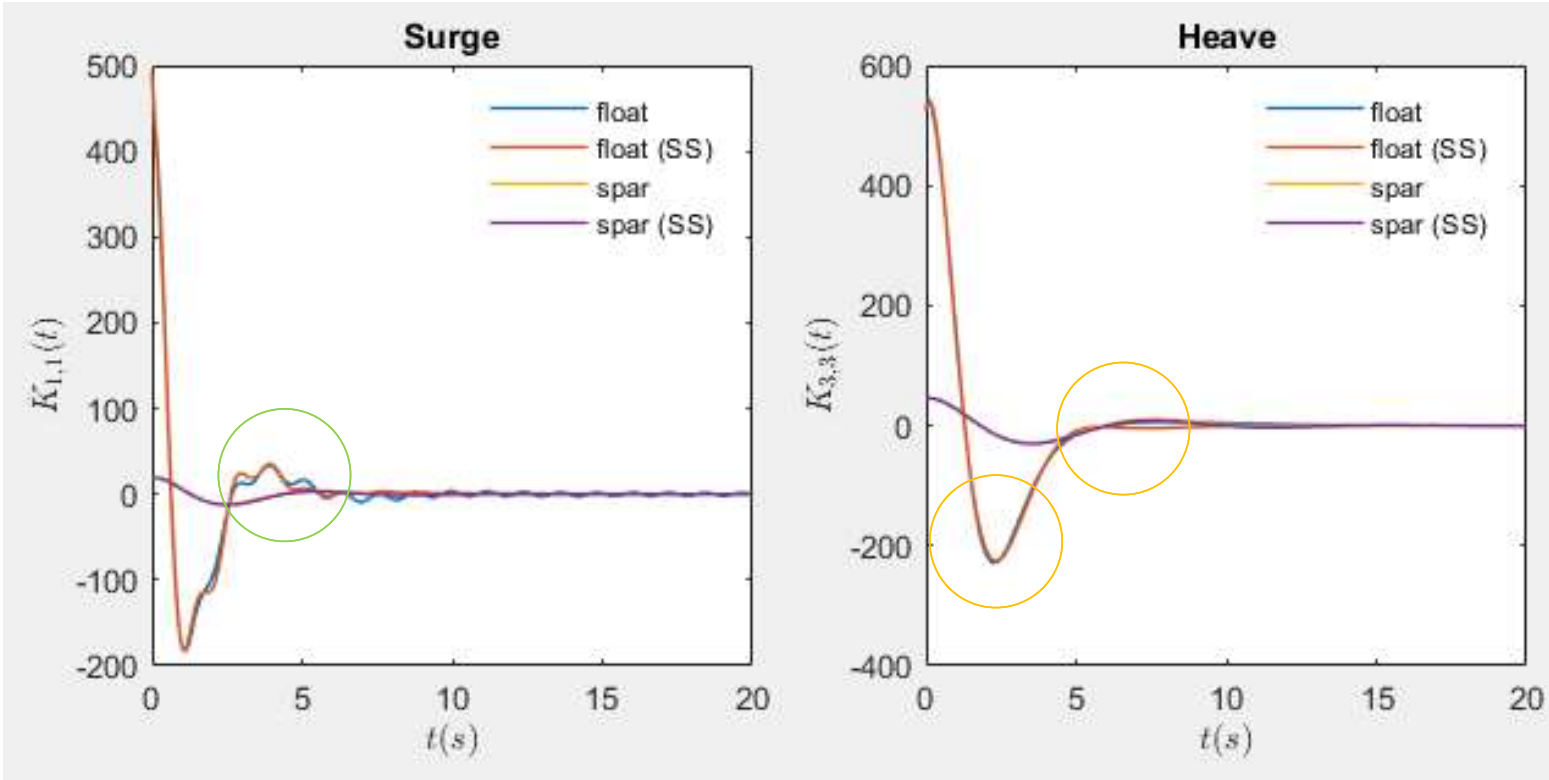




BEMIO Tutorial Practice

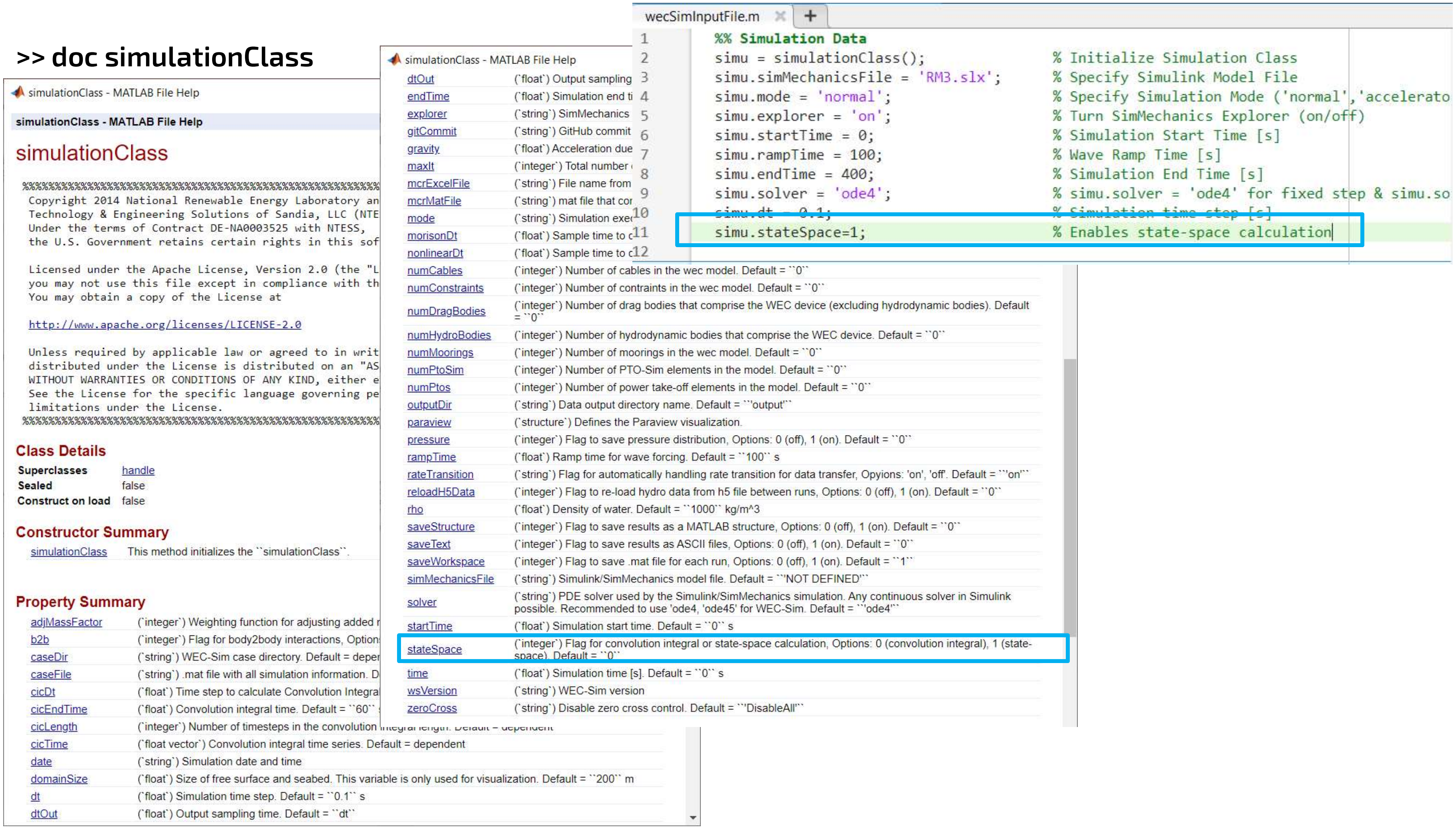
hydro = radiationIRFSS(hydro, [ ], [0.999])





BEMIO Tutorial Practice

>> doc simulationClass



simulationClass - MATLAB File Help

simulationClass - MATLAB File Help

simulationClass

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 Copyright 2014 National Renewable Energy Laboratory an
 Technology & Engineering Solutions of Sandia, LLC (NTE
 Under the terms of Contract DE-NA0003525 with NTESS,
 the U.S. Government retains certain rights in this sof

 Licensed under the Apache License, Version 2.0 (the "L
 you may not use this file except in compliance with th
 You may obtain a copy of the License at

 http://www.apache.org/licenses/LICENSE-2.0

 Unless required by applicable law or agreed to in writ
 distributed under the License is distributed on an "AS
 WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either e
 See the License for the specific language governing pe
 limitations under the License.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Class Details

Superclasses	handle
Sealed	false
Construct on load	false

Constructor Summary

simulationClass	This method initializes the ``simulationClass``.

Property Summary

adjMassFactor	(`integer`) Weighting function for adjusting added r
b2b	(`integer`) Flag for body2body interactions, Option
caseDir	(`string`) WEC-Sim case directory. Default = deper
caseFile	(`string`) .mat file with all simulation information. D
cicDt	(`float`) Time step to calculate Convolution Integra
cicEndTime	(`float`) Convolution integral time. Default = ``60``
cicLength	(`integer`) Number of timesteps in the convolution
cicTime	(`float vector`) Convolution integral time series. Default = dependent
date	(`string`) Simulation date and time
domainSize	(`float`) Size of free surface and seabed. This variable is only used for visualization. Default = ``200`` m
dt	(`float`) Simulation time step. Default = ``0.1`` s
dtOut	(`float`) Output sampling time. Default = ``dt``

simulationClass - MATLAB File Help

dtOut	(`float`) Output sampling
endTime	(`float`) Simulation end ti
explorer	(`string`) SimMechanics
gitCommit	(`string`) GitHub commit
gravity	(`float`) Acceleration due
maxIt	(`integer`) Total number
mcrExcelFile	(`string`) File name from
mcrMatFile	(`string`) mat file that cor
mode	(`string`) Simulation exec
morisonDt	(`float`) Sample time to c
nonlinearDt	(`float`) Sample time to c
numCables	(`integer`) Number of cables in the wec model. Default = ``0``
numConstraints	(`integer`) Number of contraints in the wec model. Default = ``0``
numDragBodies	(`integer`) Number of drag bodies that comprise the WEC device (excluding hydrodynamic bodies). Default = ``0``
numHydroBodies	(`integer`) Number of hydrodynamic bodies that comprise the WEC device. Default = ``0``
numMoorings	(`integer`) Number of moorings in the wec model. Default = ``0``
numPtoSim	(`integer`) Number of PTO-Sim elements in the model. Default = ``0``
numPtos	(`integer`) Number of power take-off elements in the model. Default = ``0``
outputDir	(`string`) Data output directory name. Default = ```'output'``
paraview	(`structure`) Defines the Paraview visualization.
pressure	(`integer`) Flag to save pressure distribution, Options: 0 (off), 1 (on). Default = ``0``
rampTime	(`float`) Ramp time for wave forcing. Default = ``100`` s
rateTransition	(`string`) Flag for automatically handling rate transition for data transfer, Opyions: 'on', 'off'. Default = ```'on'``
reloadH5Data	(`integer`) Flag to re-load hydro data from h5 file between runs, Options: 0 (off), 1 (on). Default = ``0``
rho	(`float`) Density of water. Default = ``1000`` kg/m^3
saveStructure	(`integer`) Flag to save results as a MATLAB structure, Options: 0 (off), 1 (on). Default = ``0``
saveText	(`integer`) Flag to save results as ASCII files, Options: 0 (off), 1 (on). Default = ``0``
saveWorkspace	(`integer`) Flag to save .mat file for each run, Options: 0 (off), 1 (on). Default = ``1``
simMechanicsFile	(`string`) Simulink/SimMechanics model file. Default = ```'NOT DEFINED'``
solver	(`string`) PDE solver used by the Simulink/SimMechanics simulation. Any continuous solver in Simulink possible. Recommended to use 'ode4, 'ode45' for WEC-Sim. Default = ```'ode4'``
startTime	(`float`) Simulation start time. Default = ``0`` s
stateSpace	(`integer`) Flag for convolution integral or state-space calculation, Options: 0 (convolution integral), 1 (state-space). Default = ``0``
time	(`float`) Simulation time [s]. Default = ``0`` s
wsVersion	(`string`) WEC-Sim version
zeroCross	(`string`) Disable zero cross control. Default = ```'DisableAll'``

wecSimInputFile.m

```
%% Simulation Data
simu = simulationClass();                   % Initialize Simulation Class
simu.simMechanicsFile = 'RM3.slx';          % Specify Simulink Model File
simu.mode = 'normal';                       % Specify Simulation Mode ('normal','accelerato
simu.explorer = 'on';                       % Turn SimMechanics Explorer (on/off)
simu.startTime = 0;                         % Simulation Start Time [s]
simu.rampTime = 100;                        % Wave Ramp Time [s]
simu.endTime = 400;                         % Simulation End Time [s]
simu.solver = 'ode4';                       % simu.solver = 'ode4' for fixed step & simu.so
simu.dt = 0.1;                              % Simulation time step [s]
simu.stateSpace=1;                          % Enables state-space calculation
```

BEMIO Tutorial Practice

How does changing the upper wave frequency limit on the IRF?

Perhaps the BEM hydrodynamic data is poor and needs to be cut off.

wecSimInputFile.m | bemio.m

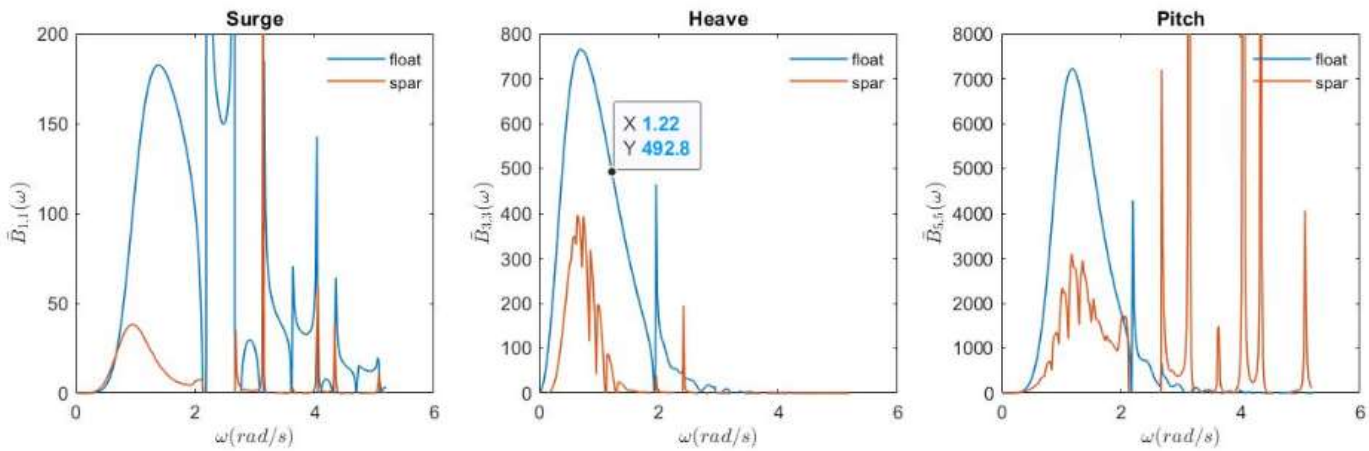
```
hydro = struct();

hydro = readWAMIT(hydro,'rm3.out',[]);
hydro = radiationIRF(hydro,20,[],[],[],[2]);
hydro = radiationIRFSS(hydro,[],[]);
hydro = excitationIRF(hydro,20,[],[],[],[]);
writeBEMIOH5(hydro)
plotBEMIO(hydro)
```

BEMIO Tutorial Practice

Depending on the BEM solver, mesh quality, and size of your device the hydrodynamic coefficients can be reported with noise and nonphysical solutions.

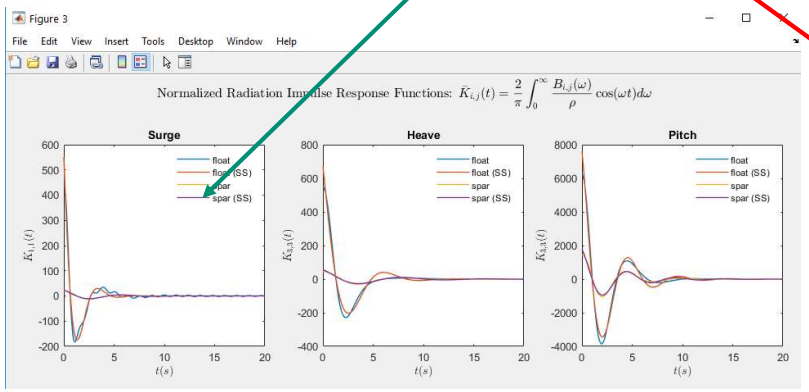
Normalized Radiation Damping: $\bar{B}_{i,j}(\omega) = \frac{B_{i,j}(\omega)}{\rho\omega}$

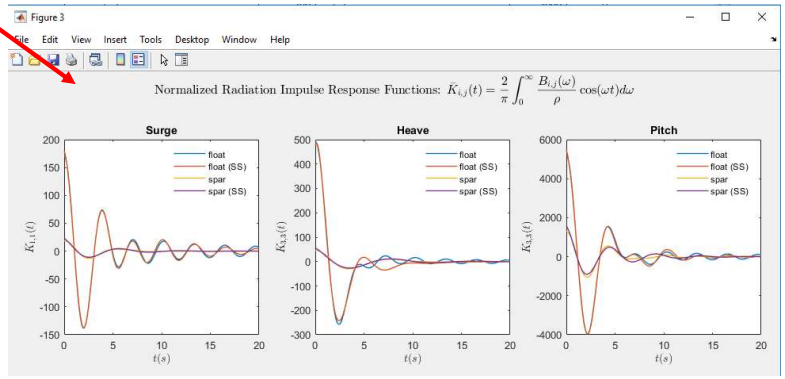


Notes:
- $\bar{B}_{i,j}(\omega)$ should tend towards zero within the specified ω range.
- Only $\bar{B}_{i,j}(\omega)$ for the surge, heave, and pitch DOFs are plotted here. If another DOF is significant to the system that $\bar{B}_{i,j}(\omega)$ should also be plotted and verified before proceeding.

Since the BEM solution defines the WEC response, poor BEM data can lead to unstable WEC-Sim simulations.

BEMIO Tutorial Practice

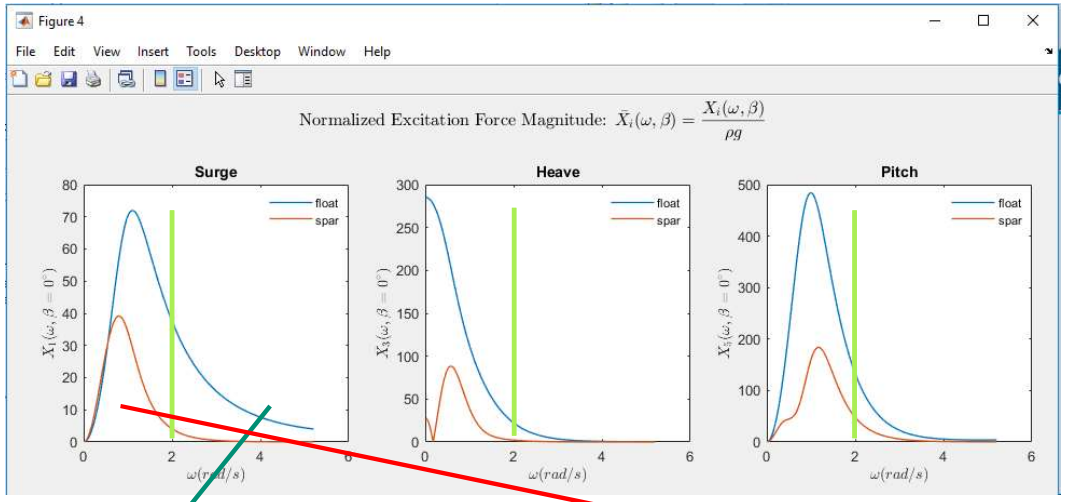


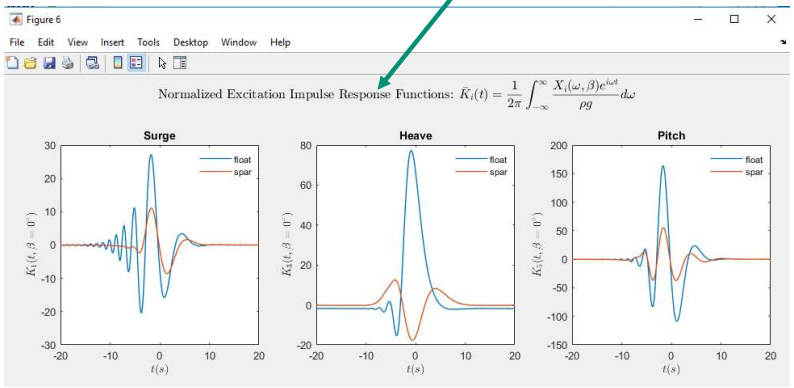


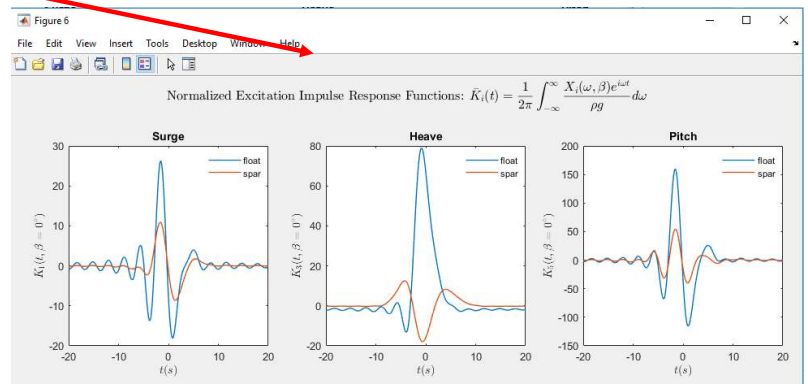
Cut-off frequency is sufficiently high

Cut-off frequency is too low

BEMIO Tutorial Practice







Cut-off frequency is sufficiently high

Cut-off frequency is too low

Thank you

For more information please visit the WEC-Sim website:

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

If you have questions on this presentation please reach out to any of the WEC-Sim Developers on GitHub:

https://github.com/WEC-Sim/WEC-Sim

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.

normalizeBEM Normalizes hydrodynamic coefficients

functions.BEMIO.normalizeBEM(*hydro*)

Normalizes BEM hydrodynamic coefficients in the same manner that WAMIT outputs are normalized. Specifically, the linear restoring stiffness is normalized as $C_{i,j}/(\rho g)$; added mass is normalized as $A_{i,j}/\rho$; radiation damping is normalized as $B_{i,j}/(\rho\omega)$; and, exciting forces are normalized as $X_i/(\rho g)$. And, if necessary, sort data according to ascending frequency.

This function is not called directly by the user; it is automatically implemented within the readWAMIT, readCAPYTAINE, readNEMOH, and readAQWA functions.

Parameters: **hydro** (`[1 x 1] struct`) – Structure of hydro data that will be normalized and sorted depending on the value of hydro.code

Returns: **hydro** – Normalized hydro data

Return type: [1 x 1] struct

$C_{i,j}/\rho g$ - linear stiffness
$A_{i,j}/\rho g$ - added mass
$B_{i,j}/\rho\omega$ - radiation damping
$X_i/\rho g$ - exciting forces