

WEC-Sim Training Course for users and developers

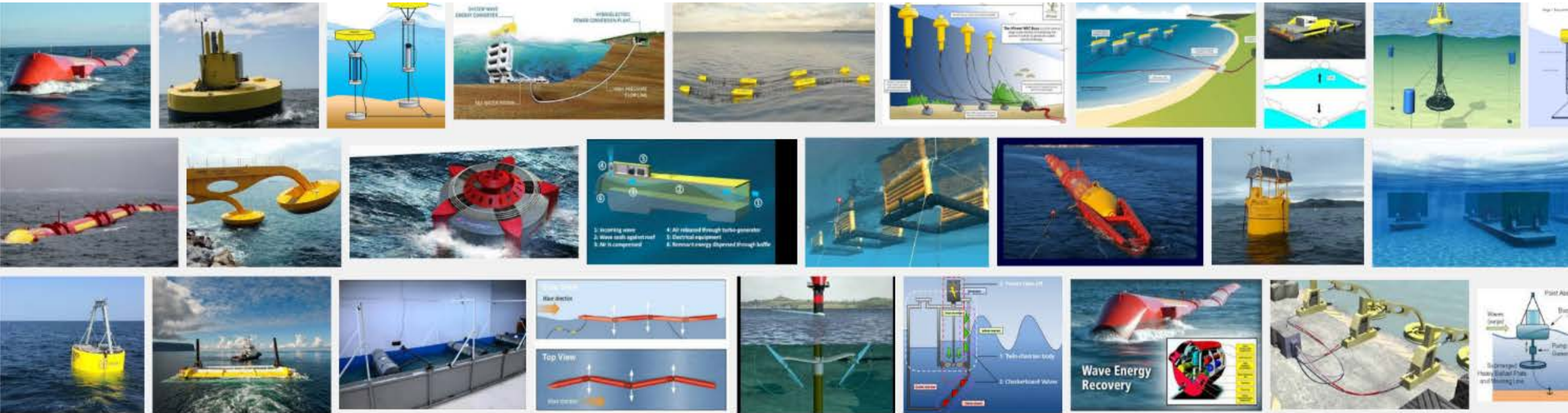
August 17, 2017

Yi-Hsiang Yu (NREL)
Kelley Ruehl (Sandia)

Course Agenda

Time	Topic	Description
9:00 am	WEC-Sim Overview ~20min	Overview of course topics and WEC-Sim code
9:30 am	Theory & Workflow ~20min	Cummins' equation and WEC-Sim workflow (BEM->BEMIO->WEC-Sim)
10:00 am	Running WEC-Sim ~30min	Description of what happens when you execute WEC-Sim (<i>wecSim.m</i>)
11:00 am	Code Structure Overview ~1hr total	Overview of WEC-Sim's input file (<i>wecSimInputFile.m</i>), classes (<i>*.m</i>) and library blocks (<i>*.s/x</i>)
1:00 pm	Wave Implementation ~30min	Description wave modeling implementation in WEC-Sim, in the classes (<i>*.m</i>) and blocks (<i>*.s/x</i>)
1:30 pm	Body Implementation ~30min	Description body implementation in WEC-Sim, in the classes (<i>*.m</i>) and blocks (<i>*.s/x</i>)
2:00pm	Q&A ~1hr	Open Q&A for attendees to WEC-Sim Lab team

Time (MT)	Topic (~Duration)	Description
11:00 am	WEC-Sim Code Structure ~1hr total	<ul style="list-style-type: none">• Code Structure Overview: ~10min<ul style="list-style-type: none">• WEC-Sim source code• WEC-Sim models: input files and Simulink models• WEC-Sim Objects (.m): ~30min<ul style="list-style-type: none">• Object oriented programming: objects, properties, methods, etc.• High level overview of: bodyClass, waveClass, simulationClass, ptoClass, constraintClass, mooring, etc• WEC-Sim Library Blocks (.slx): ~20min<ul style="list-style-type: none">• body, constraint, pto, frame<ul style="list-style-type: none">• what simulink blocks are available, how are they defined• what is a mask, how to go under a mask• variant subsystems and how to define them



WEC-Sim Code Structure

Kelley Ruehl (Sandia)

WEC-Sim **source code** consists of:

File Type	File name	Directory
WEC-Sim Executable Script	wecSim.m	\$Source
WEC-Sim MATLAB Objects	<object>Class.m	\$Source/objects
WEC-Sim Simulink Library	<block>_Lib.slx	\$Source/lib

WEC-Sim/source/

functions	File folder
<input checked="" type="checkbox"/> lib	File folder
<input checked="" type="checkbox"/> objects	File folder
paraview_macros	File folder
.gitignore	Text Document
README.md	MD File
<input checked="" type="checkbox"/> wecSim.m	MATLAB Code
wecSimstartup.m	MATLAB Code

WEC-Sim **model files** consist of:

File Type	File name	Directory
Input File	wecSimInputFile.m	\$CASE
Simulink Model	<Simulink_model_name>.slx	\$CASE
Hydrodynamic Data	<hydrodata_file_name>.h5	\$CASE/hydroData
Geometry File	<STL_file_name>.stl	\$CASE/geometry

Case Directory

<input checked="" type="checkbox"/> geometry	File folder
<input checked="" type="checkbox"/> hydroData	File folder
ndbcBuoyData.txt	Text Document
README.txt	Text Document
<input checked="" type="checkbox"/> RM3.slx	Simulink Model
simpqua16229_C_2008.mat	MATLAB Data
userDefinedFunctions.m	MATLAB Code
<input checked="" type="checkbox"/> wecSimInputFile.m	MATLAB Code

WEC-Sim Source Code

WEC-Sim **source code** consists of:

File Type	File name	Directory
WEC-Sim Executable Script	wecSim.m	<code>\$Source</code>
WEC-Sim MATLAB Objects	<object>Class.m	<code>\$Source/objects</code>
WEC-Sim Simulink Library	<block>_Lib.slx	<code>\$Source/lib</code>

- Source code is included in the MATLAB path
- Can be executed from any directory

WEC-Sim Executable

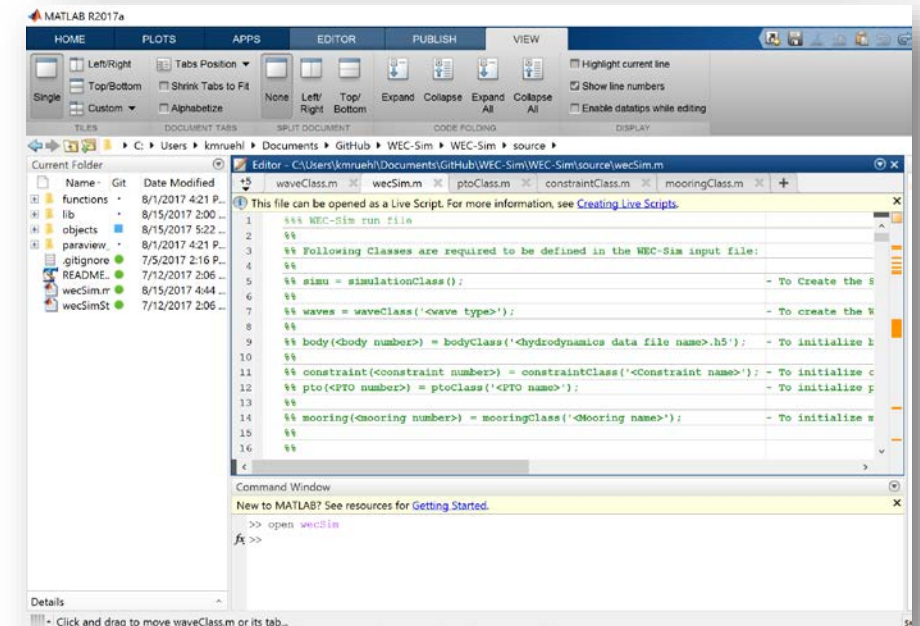
WEC-Sim/source/wecSim.m

File Type	File name	Directory
WEC-Sim Executable Script	wecSim.m	<code>\$Source</code>
WEC-Sim MATLAB Objects	<object>Class.m	<code>\$Source/objects</code>
WEC-Sim Simulink Library	<block>_Lib.slx	<code>\$Source/lib</code>

WEC-Sim/source/

functions	File folder
<input checked="" type="checkbox"/> lib	File folder
<input checked="" type="checkbox"/> objects	File folder
paraview_macros	File folder
.gitignore	Text Document
README.md	MD File
<input checked="" type="checkbox"/> wecSim.m	MATLAB Code
wecSimStartup.m	MATLAB Code

- Executing **wecSim.m**
 - parses the WEC-Sim input file
 - performs preprocessing calculations in each of the classes
 - selects and initializes variant subsystems in the Simulink model
 - runs the time-domain simulation in WEC-Sim
- View **wecSim.m** from MATLAB Command Window
>>open wecSim



WEC-Sim/source/objects/

File Type	File name	Directory
WEC-Sim Executable Script	wecSim.m	\$Source
WEC-Sim MATLAB Objects	<object>Class.m	\$Source/objects
WEC-Sim Simulink Library	<block>_Lib.slx	\$Source/lib

- Define classes in the WEC-Sim input file
 - **wecSimInputFile.m**
- The following classes create the WEC-Sim objects
 - **simulationClass.m, waveClass.m, bodyClass.m, constraintClass.m, ptoClass.m, mooringClass.m**
- WEC-Sim objects are required to run WEC-Sim simulations
 - **simu, waves, body(i), pto(i), constraint(i), mooring(i)**
- View properties or open classes from MATLAB Command Window

```
>> doc className
>> open className
```

WEC-Sim/source

functions	File folder
<input checked="" type="checkbox"/> lib	File folder
<input checked="" type="checkbox"/> objects	File folder
paraview_macros	File folder
.gitignore	Text Document
README.md	MD File
<input checked="" type="checkbox"/> wecSim.m	MATLAB Code
wecSimStartup.m	MATLAB Code

WEC-Sim/source/objects

bodyClass.m	MATLAB Code
constraintClass.m	MATLAB Code
mooringClass.m	MATLAB Code
ptoClass.m	MATLAB Code
ptoSimClass.m	MATLAB Code
responseClass.m	MATLAB Code
simulationClass.m	MATLAB Code
waveClass.m	MATLAB Code

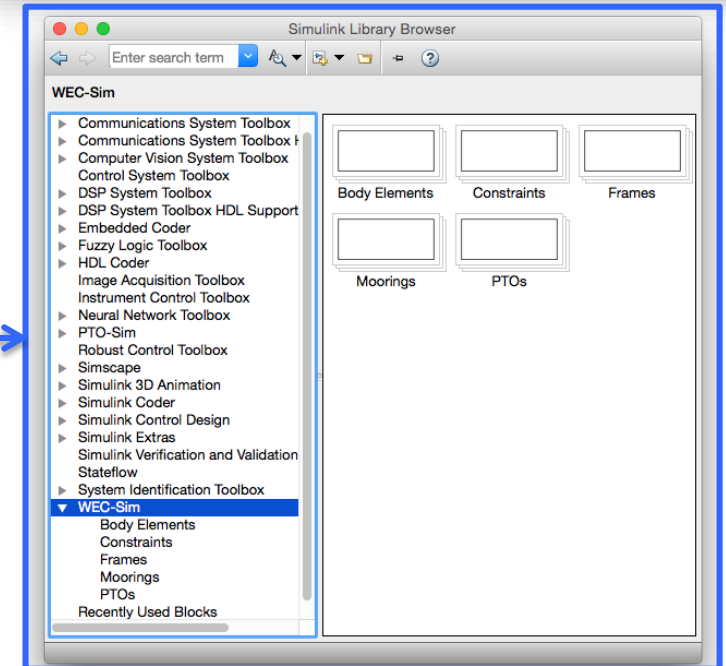
WEC-Sim/source/lib/

File Type	File name	Directory
WEC-Sim Executable Script	wecSim.m	\$Source
WEC-Sim MATLAB Objects	<object>Class.m	\$Source/objects
WEC-Sim Simulink Library	<block>_Lib.slx	\$Source/lib

WEC-Sim/source

functions	File folder
<input checked="" type="checkbox"/> lib	File folder
<input checked="" type="checkbox"/> objects	File folder
paraview_macros	File folder
.gitignore	Text Document
README.md	MD File
<input checked="" type="checkbox"/> wecSim.m	MATLAB Code
wecSimStartup.m	MATLAB Code

- WEC-Sim source code includes WEC-Sim library blocks:
 - Body Elements, Constraints, Frames, Moorings, PTOs**
- Define WEC dynamics in WEC-Sim Simulink model using WEC-Sim Library Blocks
 - <Simulink_modelname>.slx**
- View properties by double clicking on blocks
 - displays description and block parameters
- There should be a one-to-one between the objects defined in the input file and the blocks used in the Simulink model.



WEC-Sim Model Files

WEC-Sim **model files** consist of:

File Type	File name	Directory
Input File	wecSimInputFile.m	\$CASE
Simulink Model	<Simulink_model_name>.slx	\$CASE
Hydrodynamic Data	<hydrodata_file_name>.h5	\$CASE/hydroData
Geometry File	<STL_file_name>.stl	\$CASE/geometry

- Model files are located in the case directory
- WEC-Sim models must be executed from the case directory

WEC-Sim Input File

wecSimInputFile.m

File Type	File name	Directory
Input File	wecSimInputFile.m	\$CASE
Simulink Model	<Simulink_model_name>.slx	\$CASE
Hydrodynamic Data	<hydrodata_file_name>.h5	\$CASE/hydroData
Geometry File	<STL_file_name>.stl	\$CASE/geometry

Case Directory

geometry	File folder
hydroData	File folder
ndbcBuoyData.txt	Text Document
README.txt	Text Document
<input checked="" type="checkbox"/> RM3.slx	Simulink Model
umpqua46229_6_2008.mat	MATLAB Data
userDefinedFunctions.m	MATLAB Code
<input checked="" type="checkbox"/> wecSimInputFile.m	MATLAB Code

- Located in the case directory
- Initialize and define classes in the WEC-Sim input file
 - *wecSimInputFile.m*
- WEC-Sim objects are required to run WEC-Sim simulations
 - *simu, waves, body(i), pto(i), constraint(i), mooring(i)*

WEC-Sim Input File

wecSimInputFile.m

```
wecSimInputFile.m
%% Simulation Data
simu = simulationClass();
simu.simMechanicsFile = 'RM3.slx';
simu.startTime = 0;
simu.rampTime = 100;
simu.endTime=400;
simu.dt = 0.1;

%% Wave Information
% Regular Waves
waves = waveClass('regular');
waves.H = 2.5;
waves.T = 8;

%% Body Data
% Float
body(1) = bodyClass('hydroData/rm3.h5');
body(1).geometryFile = 'geometry/float.stl';
body(1).mass = 'equilibrium';
body(1).momOfInertia = [20907301 21306090.66 37085481.11];

% Spar/Plate
body(2) = bodyClass('hydroData/rm3.h5');
body(2).geometryFile = 'geometry/plate.stl';
body(2).mass = 'equilibrium';
body(2).momOfInertia = [94419614.57 94407091.24 28542224.82];

%% PTO and Constraint Parameters
% Floating (3DOF) Joint
constraint(1) = constraintClass('Constraint1');
constraint(1).loc = [0 0 0];

% Translational PTO
pto(1) = ptoClass('PTO1');
pto(1).k = 0;
pto(1).c = 1200000;
pto(1).loc = [0 0 0];
```

- Initialize Simulation Class
- Set Properties of Simulation Class
- Initialize Wave Class
- Set Properties of Wave Class
- Initialize Body Class Instances
- Set Properties of Body Class Instances
- Initialize Constraint Class
- Set Properties of Constraint Class
- Initialize PTO Class
- Set Properties of PTO Class

<Simulink_model_name>.slx

File Type	File name	Directory
Input File	wecSimInputFile.m	\$CASE
Simulink Model	<Simulink_model_name>.slx	\$CASE
Hydrodynamic Data	<hydrodata_file_name>.h5	\$CASE/hydroData
Geometry File	<STL_file_name>.stl	\$CASE/geometry

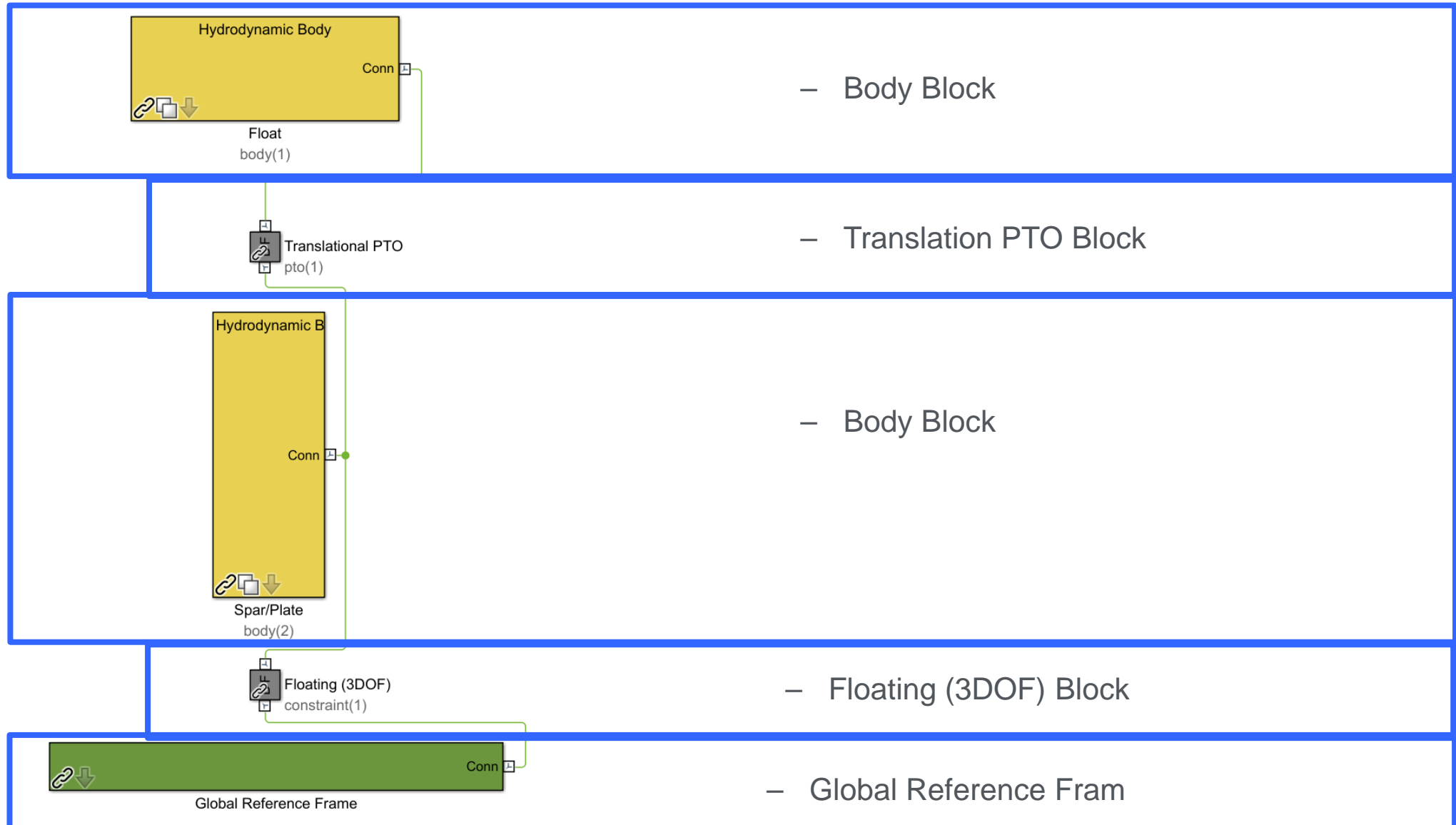
Case Directory

geometry	File folder
hydroData	File folder
ndbcBuoyData.txt	Text Document
README.txt	Text Document
<input checked="" type="checkbox"/> RM3.slx	Simulink Model
umpqua46229_6_2008.mat	MATLAB Data
userDefinedFunctions.m	MATLAB Code
<input checked="" type="checkbox"/> wecSimInputFile.m	MATLAB Code

- Located in the case directory
- Define model file using WEC-Sim Library Blocks
 - <Simulink_model_name>.slx

WEC-Sim Simulink File

`<Simulink_model_name>.slx`



WEC-Sim Objects

WEC-Sim/source/objects/

File Type	File name	Directory
WEC-Sim Executable Script	wecSim.m	\$Source
WEC-Sim MATLAB Objects	<object>Class.m	\$Source/objects
WEC-Sim Simulink Library	<block>_Lib.slx	\$Source/lib

- Define classes in the WEC-Sim input file
 - *wecSimInputFile.m*
- The following classes create the WEC-Sim objects
 - *simulationClass.m, waveClass.m, bodyClass.m, constraintClass.m, ptoClass.m, mooringClass.m*
- WEC-Sim objects are required to run WEC-Sim simulations
 - *simu, waves, body(i), pto(i), constraint(i), mooring(i)*

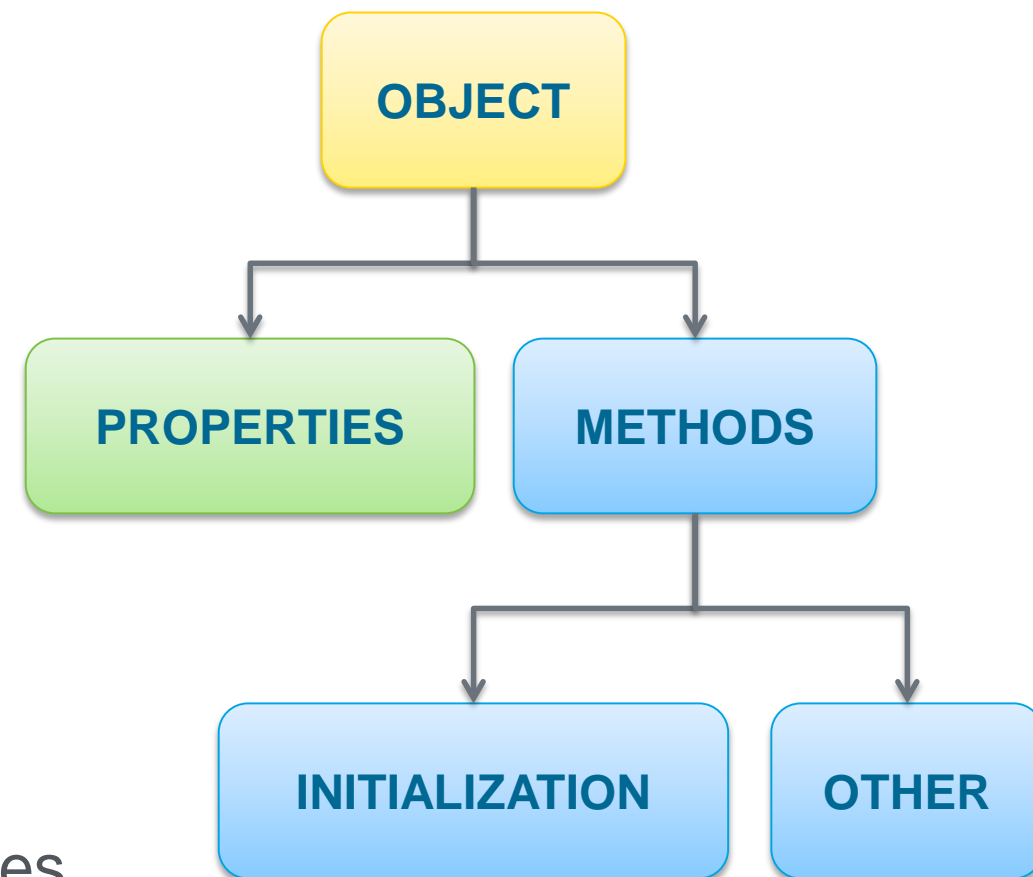
WEC-Sim/source

functions	File folder
<input checked="" type="checkbox"/> lib	File folder
<input checked="" type="checkbox"/> objects	File folder
paraview_macros	File folder
.gitignore	Text Document
README.md	MD File
<input checked="" type="checkbox"/> wecSim.m	MATLAB Code
wecSimStartup.m	MATLAB Code

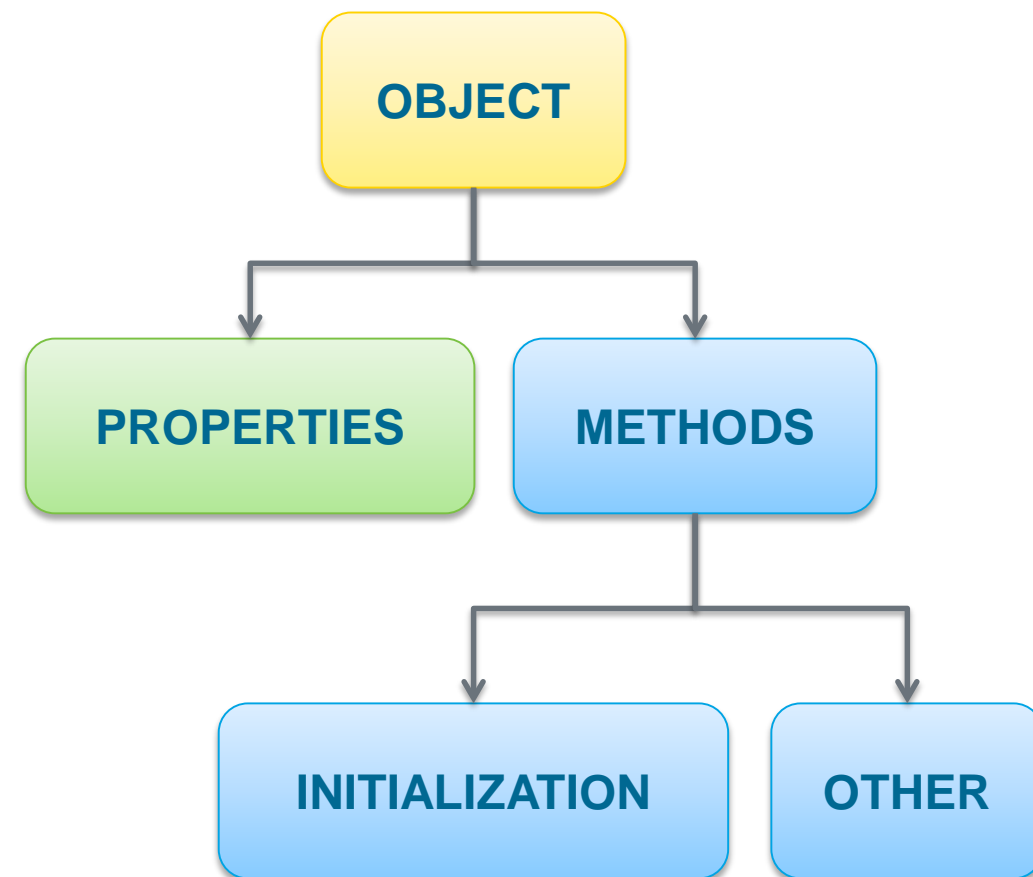
WEC-Sim/source/objects

bodyClass.m	MATLAB Code
constraintClass.m	MATLAB Code
mooringClass.m	MATLAB Code
ptoClass.m	MATLAB Code
ptoSimClass.m	MATLAB Code
responseClass.m	MATLAB Code
simulationClass.m	MATLAB Code
waveClass.m	MATLAB Code

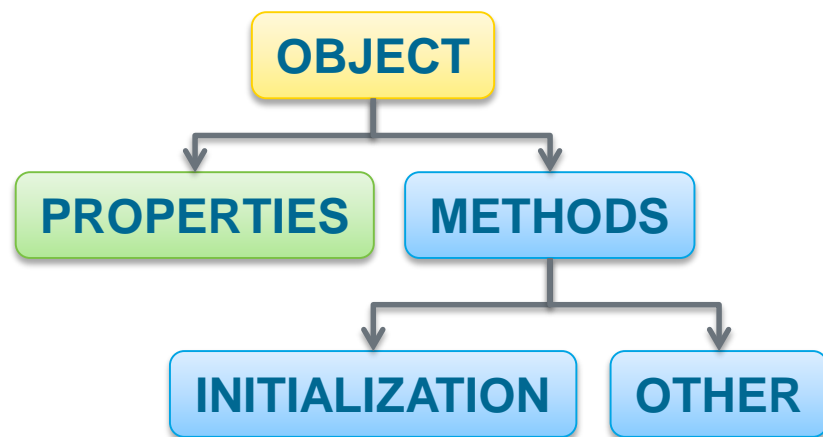
- WEC-Sim has several different classes
 - *simulationClass.m*
 - *waveClass.m*
 - *bodyClass.m*
 - *constraintClass.m*
 - *ptoClass.m*
 - *mooringClass.m*
 - *responseClass.m*
- Each class contains:
 - Properties that can be defined
 - Methods (aka functions) that can be executed
- WEC-Sim input file determines which properties are defined and methods are executed



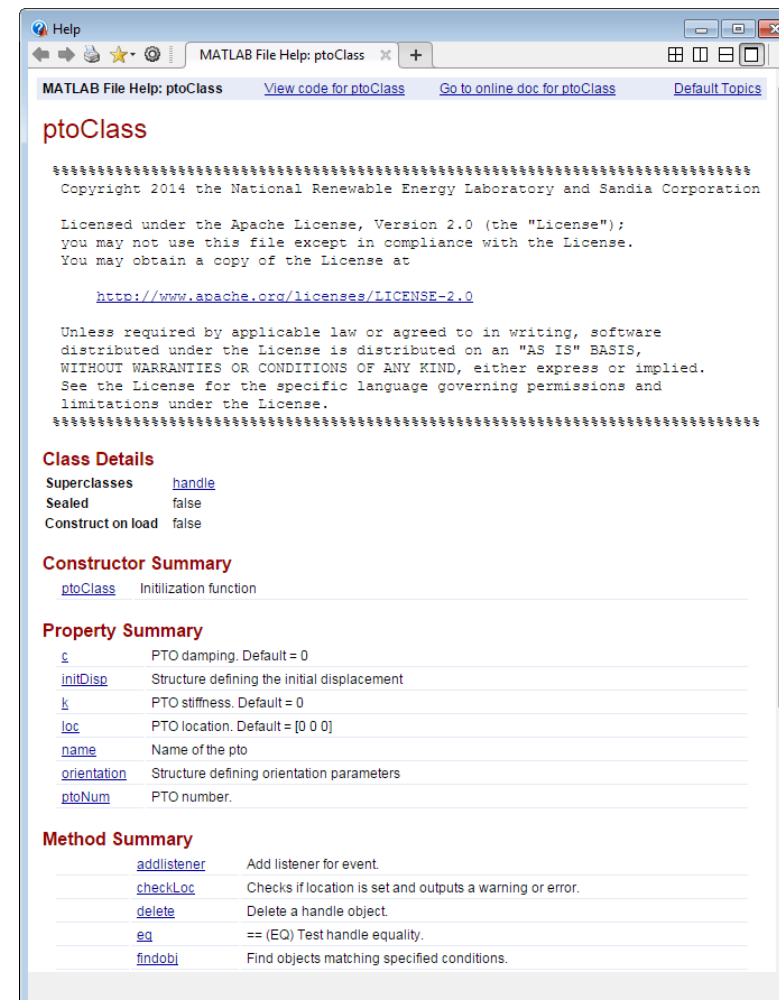
- Each class creates a corresponding object
 - *simulationClass.m* → *simu*
 - *waveClass.m* → *waves*
 - *bodyClass.m* → *body(i)*
 - *constraintClass.m* → *constraint(i)*
 - *ptoClass.m* → *pto(i)*
 - *mooringClass.m* → *mooring(i)*
 - *responseClass.m* → *output*
- Properties defined in the WEC-Sim input file used initialize variant subsystems in the WEC-Sim Library Blocks
 - *simu.b2b = 1;*
 - *body(i).nhBody = 1;*
 - *waves = waveClass('regular');*



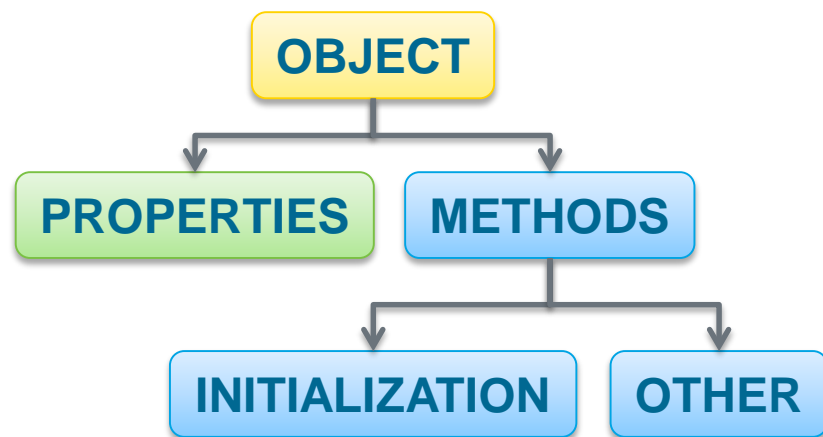
- WEC-Sim Classes
 - *simulationClass.m*
 - *waveClass.m*
 - *bodyClass.m*
 - *constraintClass.m*
 - *ptoClass.m*
 - *mooringClass.m*
 - *responseClass.m*



>> doc ptoClass



- WEC-Sim Classes
 - *simulationClass.m*
 - *waveClass.m*
 - *bodyClass.m*
 - *constraintClass.m*
 - *ptoClass.m*
 - *mooringClass.m*
 - *responseClass.m*



>> open ptoClass

```
classdef ptoClass<handle
    properties (SetAccess = 'public', GetAccess = 'public')%input file
        name = 'NOT DEFINED'
        k = 0
        c = 0
        loc = [999 999 999]
        orientation = struct(...
            'z', [0, 0, 1], ...
            'y', [0, 1, 0], ...
            'x', [], ...
            'rotationMatrix', [])
        initDisp = struct(...
            'initLinDisp', [0 0 0])
    end

    properties (SetAccess = 'public', GetAccess = 'public')%internal
        ptoNum = []
    end

    methods
        function obj = ptoClass(name)
            % Initialization function
            obj.name = name;
        end

        function obj = checkLoc(obj, action)
            % Checks if location is set and outputs a warning or error.
            % Used in mask Initialization.
            switch action
                case 'W'
                    if obj.loc == 999 % Because "Allow library block to modify its
                        obj.loc = [888 888 888];
                    end
                otherwise
                    % Do nothing
                end
            end
        end
    end
end
```

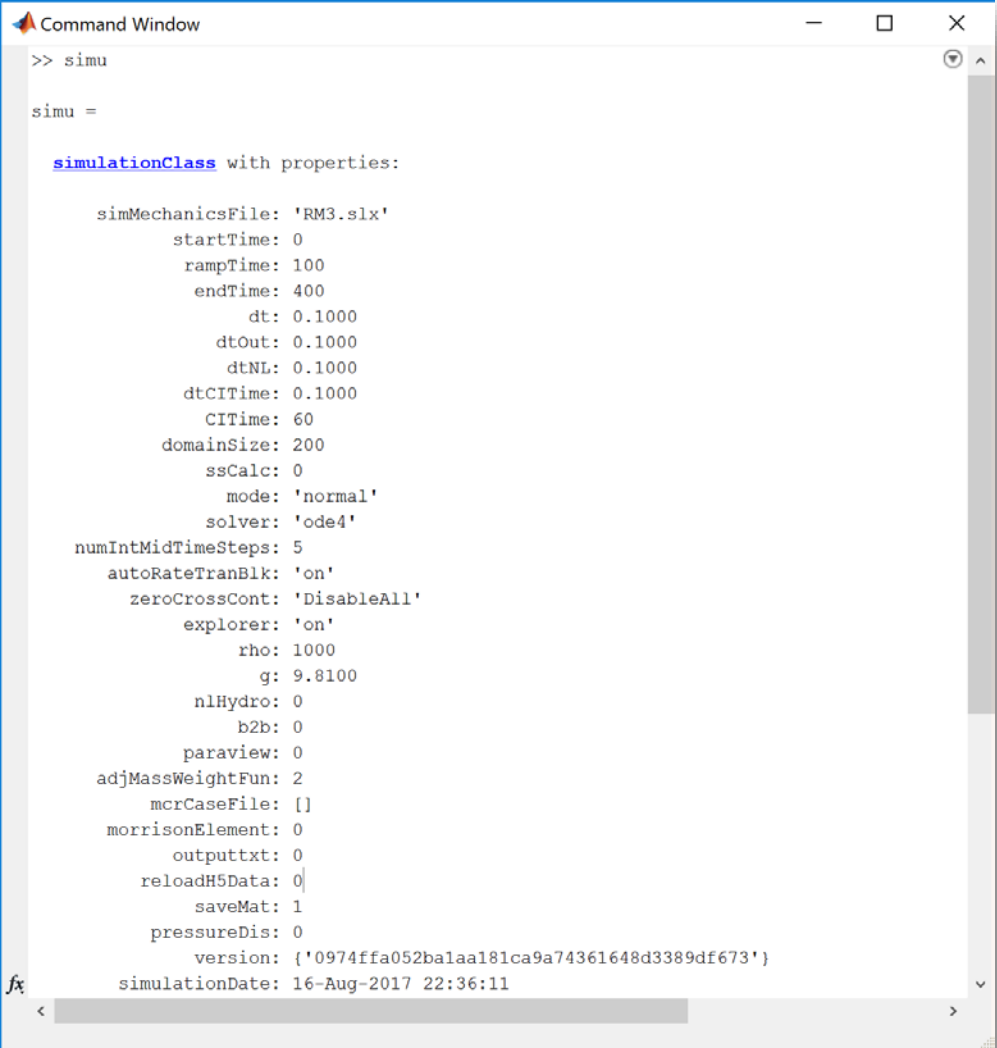
WEC-Sim Class Descriptions

simulationClass.m

The simulation class contains the simulation parameters and solver settings necessary to execute the WEC-Sim code.

- Required Properties:
 - simMechanicsFile
 - startTime, endTime, dt, rampTime, CTime
 - (many have default values)
- Optional Properties:
 - ssCalc, nlHydro
 - mode, solver
 - explorer, paraview, domainSize
 - rho, g
- Contains all simulation options (time, fluid properties, solver, visualization, etc.)
- Contains flags for different options (state space, non-linear hydro, etc.)

>>simu



```
Command Window
>> simu

simu =

  simulationClass with properties:

    simMechanicsFile: 'RM3.slx'
      startTime: 0
      rampTime: 100
      endTime: 400
        dt: 0.1000
      dtOut: 0.1000
      dtNL: 0.1000
    dtCTime: 0.1000
      CTime: 60
    domainSize: 200
      ssCalc: 0
        mode: 'normal'
      solver: 'ode4'
    numIntMidTimeSteps: 5
    autoRateTranBlk: 'on'
    zeroCrossCont: 'DisableAll'
      explorer: 'on'
        rho: 1000
          g: 9.8100
      nlHydro: 0
        b2b: 0
      paraview: 0
    adjMassWeightFun: 2
      mcrCaseFile: []
    morrisonElement: 0
      outputtxt: 0
    reloadH5Data: 0
      saveMat: 1
    pressureDis: 0
      version: {'0974ffa052balaal81ca9a74361648d3389df673'}
    simulationDate: 16-Aug-2017 22:36:11
```


waveClass.m

The wave class contains all wave information necessary to define the incident wave condition for the WEC-Sim time-domain simulation.

- Required Properties:
 - type
 - Each wave ‘type’ has different required properties

Wave Type	Required Properties
<code>noWave</code>	<code>waves.T</code>
<code>noWaveCIC</code>	N/A
<code>regular</code>	<code>waves.H</code> , <code>waves.T</code>
<code>regularCIC</code>	<code>waves.H</code> , <code>waves.T</code>
<code>irregular</code>	<code>waves.H</code> , <code>waves.T</code> , <code>waves.spectrumType</code>
<code>spectrumImport</code>	<code>waves.spectrumDataFile</code>
<code>etaImport</code>	<code>waves.etaDataFile</code>

`>>waves`

```
Command Window
>> waves

waves =

waveClass with properties:

    type: 'regular'
        T: 8
        H: 2.5000
    spectrumType: 'NOT DEFINED'
    randPreDefined: 0
    spectrumDataFile: 'NOT DEFINED'
    etaDataFile: 'NOT DEFINED'
    numFreq: 1001
    freqRange: []
    waveDir: 0
    viz: [1x1 struct]
    statisticsDataLoad: []
    typeNum: 10
    bemFreq: [1x260 double]
    waterDepth: 200
    deepWaterWave: 1
    waveAmpTime: [4001x2 double]
        A: 1.2500
        w: 0.7854
    phaseRand: 0
    dw: 0
    k: 0.0629

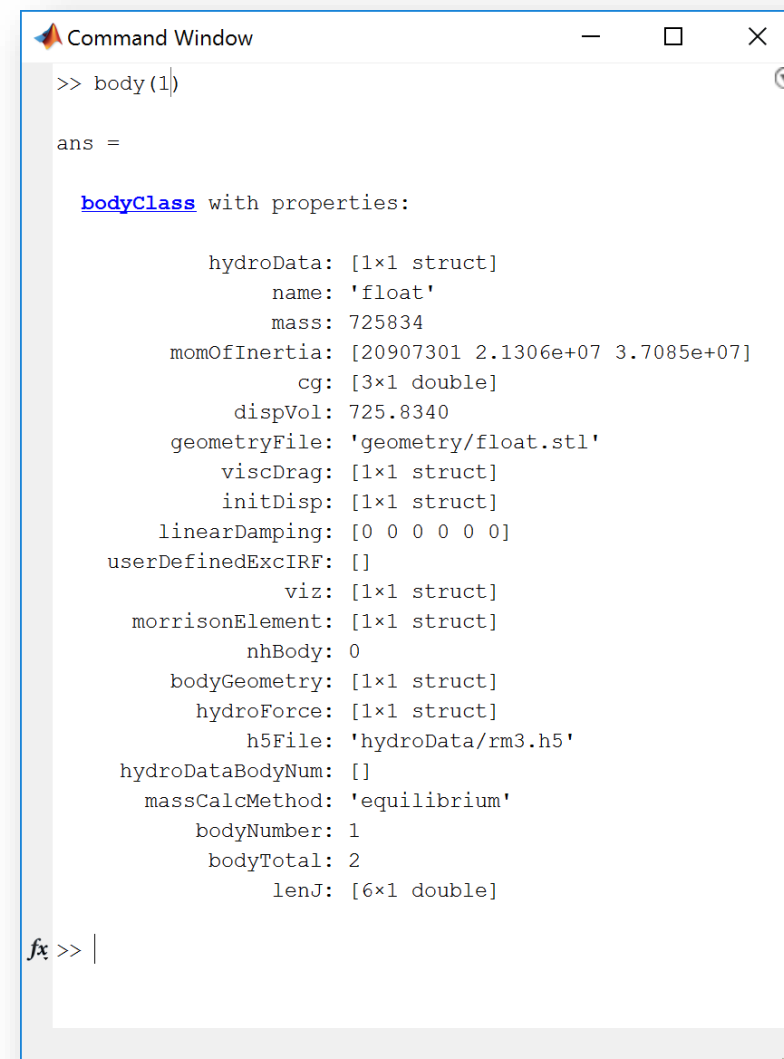
fx >> |
```

bodyClass.m

The body class contains the mass and hydrodynamic properties of each body that comprises the WEC being simulated.

- Required Properties:
 - mass: value, 'equilibrium', 'fixed'
 - momOfInertia (except for 'fixed' mass)
 - geometryFile
 - h5File
- Optional Properties:
 - viscDrag
 - initDisp
 - linearDamping
 - Viz
- Useful input information:
 - hydroData (BEM data)
 - hydroForce (force time-series)

>>body(1)



```
Command Window
>> body(1)

ans =

    bodyClass with properties:

        hydroData: [1x1 struct]
              name: 'float'
              mass: 725834
        momOfInertia: [20907301 2.1306e+07 3.7085e+07]
              cg: [3x1 double]
        dispVol: 725.8340
        geometryFile: 'geometry/float.stl'
        viscDrag: [1x1 struct]
        initDisp: [1x1 struct]
        linearDamping: [0 0 0 0 0 0]
        userDefinedExcIRF: []
              viz: [1x1 struct]
        morrisonElement: [1x1 struct]
              nhBody: 0
        bodyGeometry: [1x1 struct]
        hydroForce: [1x1 struct]
              h5File: 'hydroData/rm3.h5'
        hydroDataBodyNum: []
        massCalcMethod: 'equilibrium'
        bodyNumber: 1
        bodyTotal: 2
              lenJ: [6x1 double]
```

constraintClass.m

Constraint blocks connect WEC bodies to one another (and possibly to the seabed) by constraining DOFs.

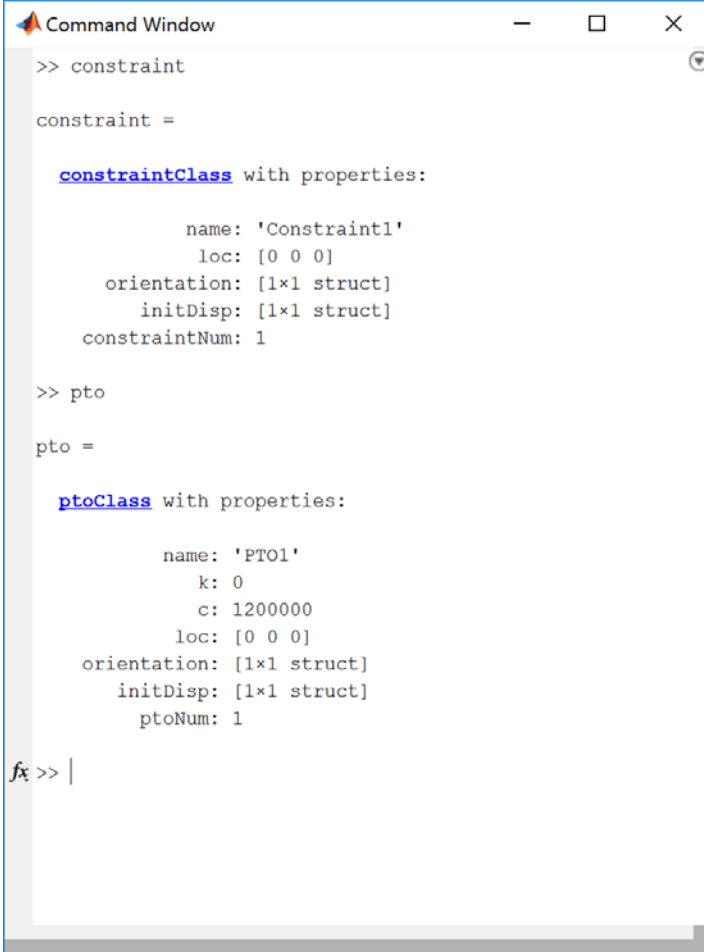
- Constraint Class Properties
 - name
 - Loc

ptoClass.m

Power Take-Off (PTO) blocks connect WEC bodies to one other (and possibly to the seabed) by constraining DOFs and applying linear damping and stiffness

- PTO Class Properties
 - name
 - loc
 - k
 - c
- NOTE: loc required for rotational PTOs & Constraints

>>constraint >>pto



```
Command Window
>> constraint
constraint =
    constraintClass with properties:
        name: 'Constraint1'
        loc: [0 0 0]
        orientation: [1x1 struct]
        initDisp: [1x1 struct]
        constraintNum: 1

>> pto
pto =
    ptoClass with properties:
        name: 'PTO1'
        k: 0
        c: 1200000
        loc: [0 0 0]
        orientation: [1x1 struct]
        initDisp: [1x1 struct]
        ptoNum: 1

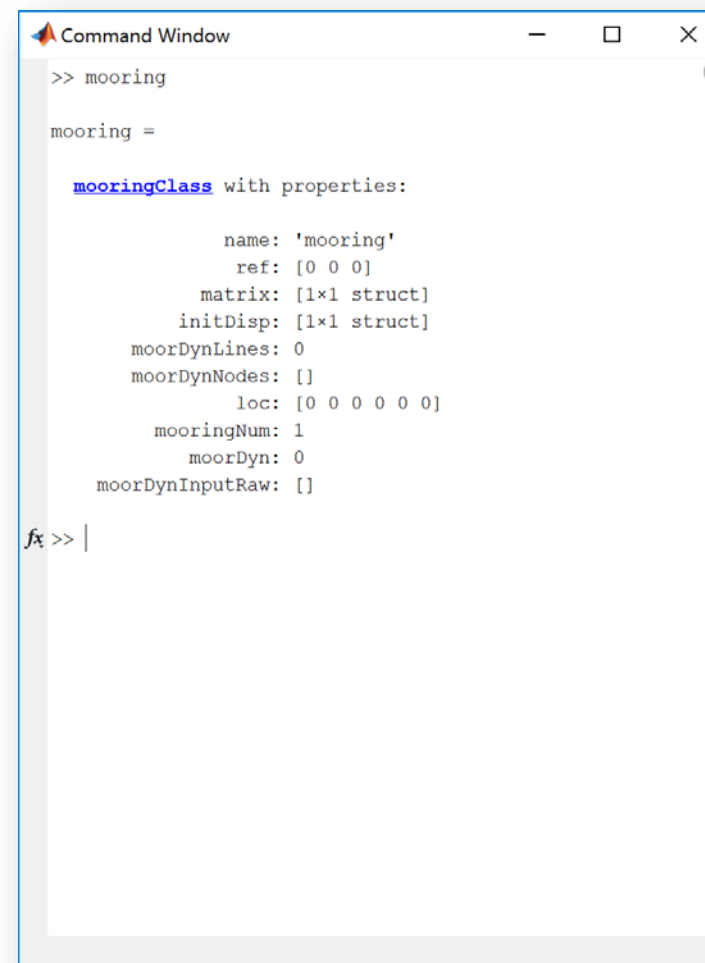
fx >> |
```

mooringClass.m

Mooring class contains all information about definition of the mooring system

- Mooring types:
 - Matrices
 - MoorDyn
- Properties:
 - name
 - ref
 - matrix
 - initDisp
 - moorDynLines

>>mooring



```
Command Window
>> mooring

mooring =

    mooringClass with properties:

        name: 'mooring'
         ref: [0 0 0]
      matrix: [1x1 struct]
    initDisp: [1x1 struct]
 moorDynLines: 0
 moorDynNodes: []
         loc: [0 0 0 0 0 0]
    mooringNum: 1
      moorDyn: 0
 moorDynInputRaw: []

fx >> |
```

responseClass.m

‘output’ created at the end of a WEC-Sim simulation. It contains all the output time-series and methods to plot and interact with the results.

- ‘output’ variable = responseClass instance
 - Contains all time series from simulation
 - Contains all time-series calculations
 - methods for quick plotting
- Properties:
 - bodies
 - ptos
 - constraints
 - ptosim

\$Case/output/RM3_matlabWorkspace.mat

http://wec-sim.github.io/WEC-Sim/code_structure.html#output-structure

>>output



```
Command Window
>> output
output =

responseClass with properties:

    wave: [1x1 struct]
   bodies: [1x2 struct]
    ptos: [1x1 struct]
constraints: [1x1 struct]
   ptosim: [1x1 struct]
   mooring: [1x1 struct]
   moorDyn: [1x1 struct]

>> output.bodies(1)

ans =

struct with fields:

    name: 'buoy_a'
   time: [40001x1 double]
position: [40001x6 double]
velocity: [40001x6 double]
acceleration: [40001x6 double]
   forceTotal: [40001x6 double]
forceExcitation: [40001x6 double]
forceRadiationDamping: [40001x6 double]
   forceAddedMass: [40001x6 double]
   forceRestoring: [40001x6 double]
forceMorrisonAndViscous: [40001x6 double]
   forceLinearDamping: [40001x6 double]
```

WEC-Sim Library

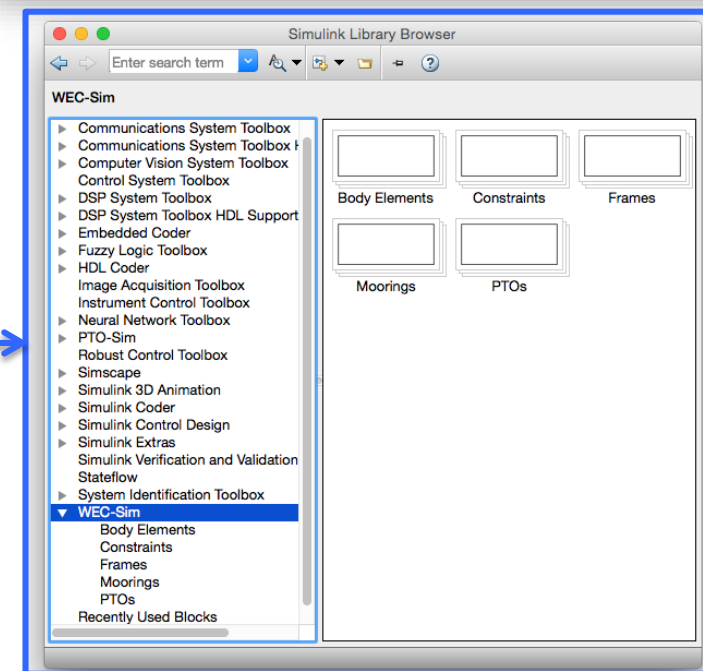
WEC-Sim/source/lib/

File Type	File name	Directory
WEC-Sim Executable Script	wecSim.m	\$Source
WEC-Sim MATLAB Objects	<object>Class.m	\$Source/objects
WEC-Sim Simulink Library	<block>_Lib.slx	\$Source/lib

WEC-Sim/source

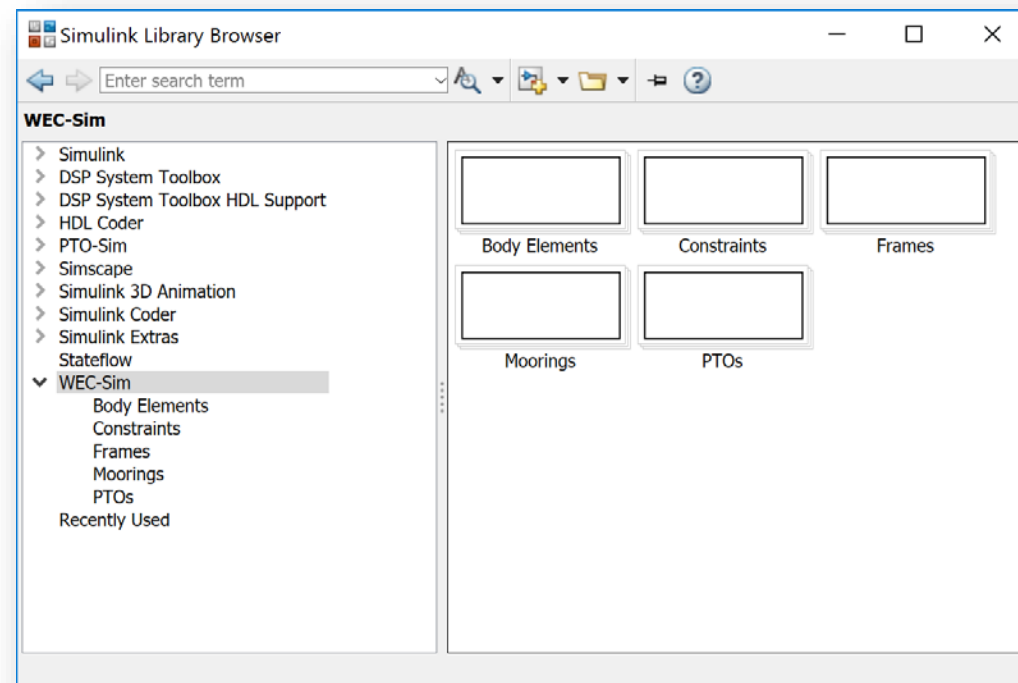
functions	File folder
<input checked="" type="checkbox"/> lib	File folder
<input checked="" type="checkbox"/> objects	File folder
paraview_macros	File folder
.gitignore	Text Document
README.md	MD File
<input checked="" type="checkbox"/> wecSim.m	MATLAB Code
wecSimStartup.m	MATLAB Code

- WEC-Sim source code includes WEC-Sim library blocks:
 - Body Elements, Constraints, Frames, Moorings, PTOs**
- Define WEC dynamics in WEC-Sim Simulink model using WEC-Sim Library Blocks
 - <Simulink_modelname>.slx**
- View properties by double clicking on blocks
 - displays description and block parameters
- There should be a one-to-one between the objects defined in the input file and the blocks used in the Simulink model.



WEC-Sim/source/lib/

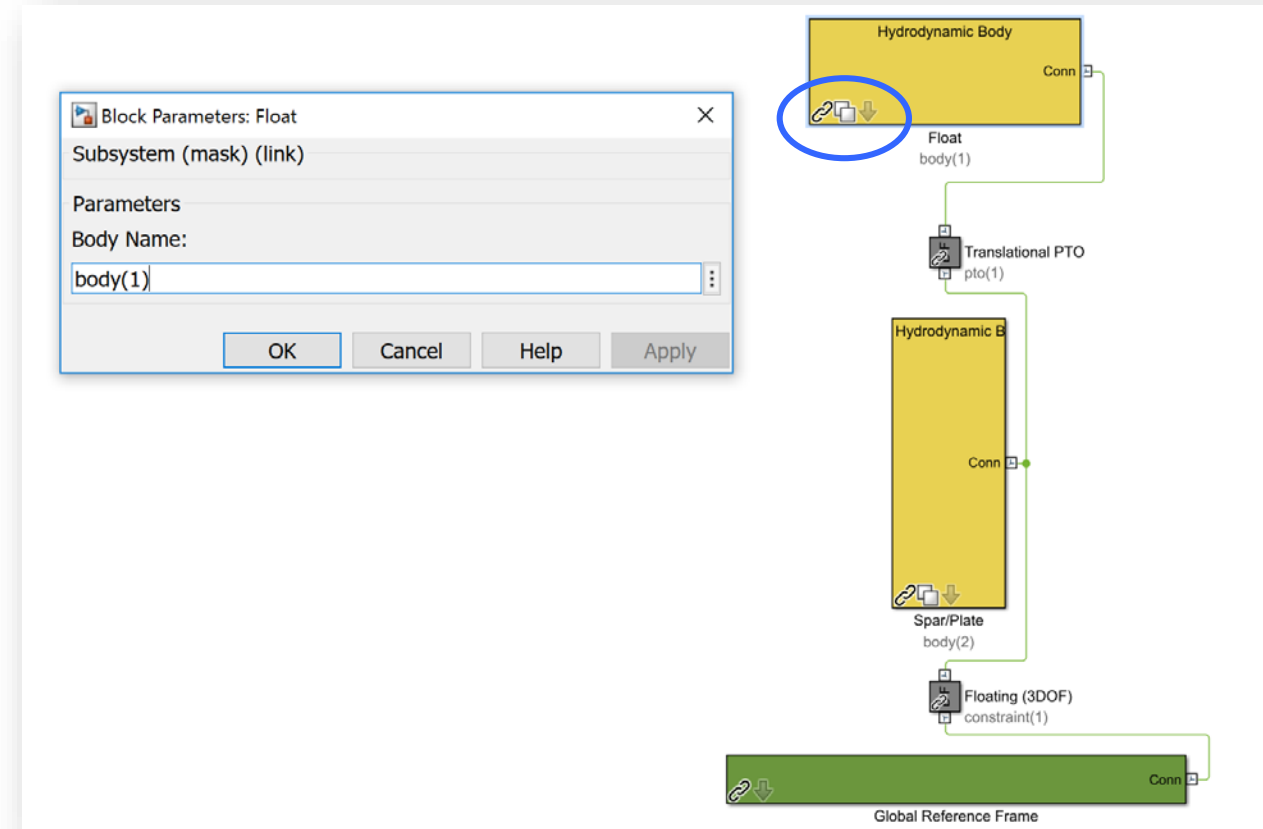
- WEC-Sim Library
 - Drag & Drop library
 - Mask
- Simulink Model
 - Made of WEC-Sim library blocks
 - Body Elements, PTOs, Constraints need to be numbered



WEC-Sim/source/lib/

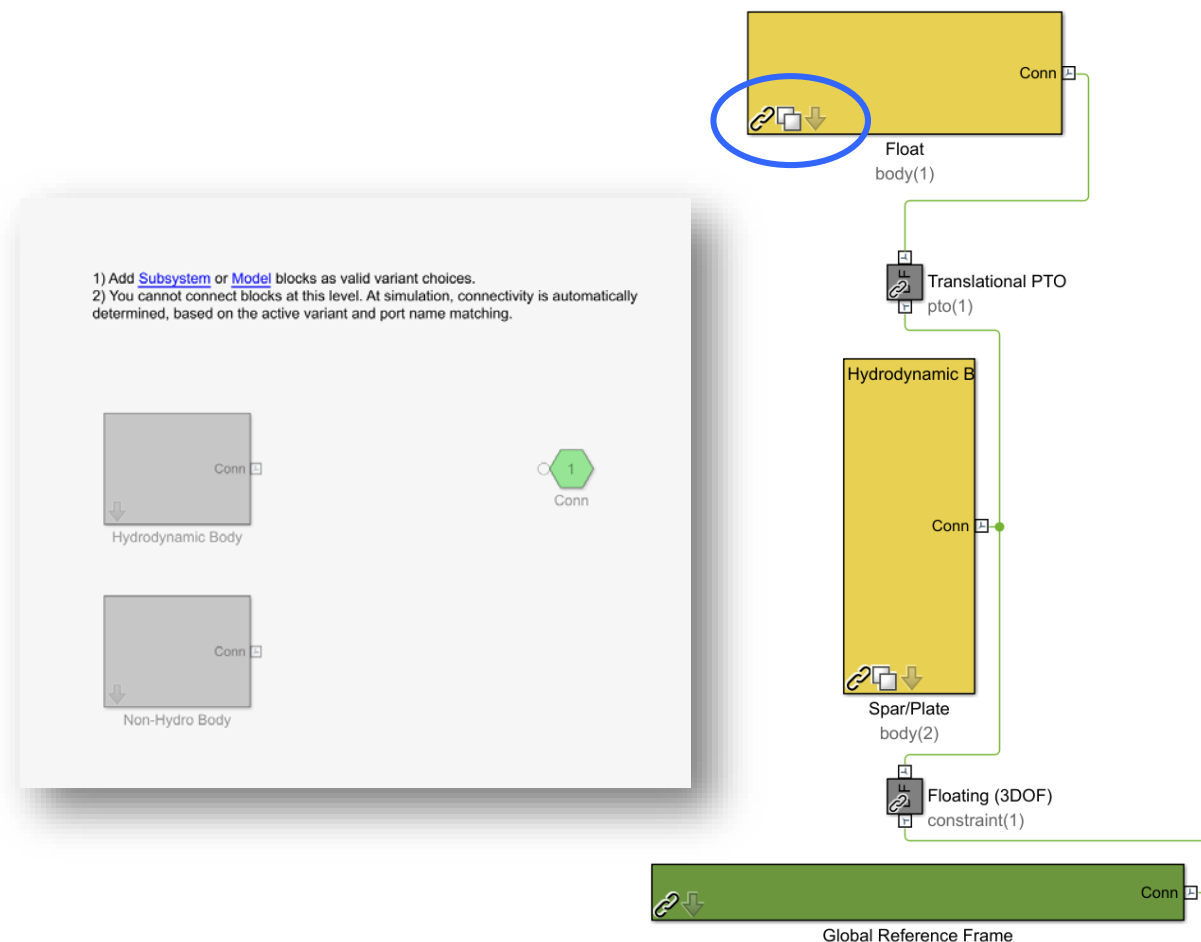
- Simulink Masks
 - A mask is a custom user interface for a block.
 - By masking a block you encapsulate the block diagram to have its own parameter dialog box with its own block description, parameter prompts, and help texts.
- **Denoted by down arrow**
- **Double clicking on mask opens dialog box**
- Clicking on down arrow (Ctrl+U) goes under the mask

<https://www.mathworks.com/help/simulink/block-masks.html>



WEC-Sim/source/lib/

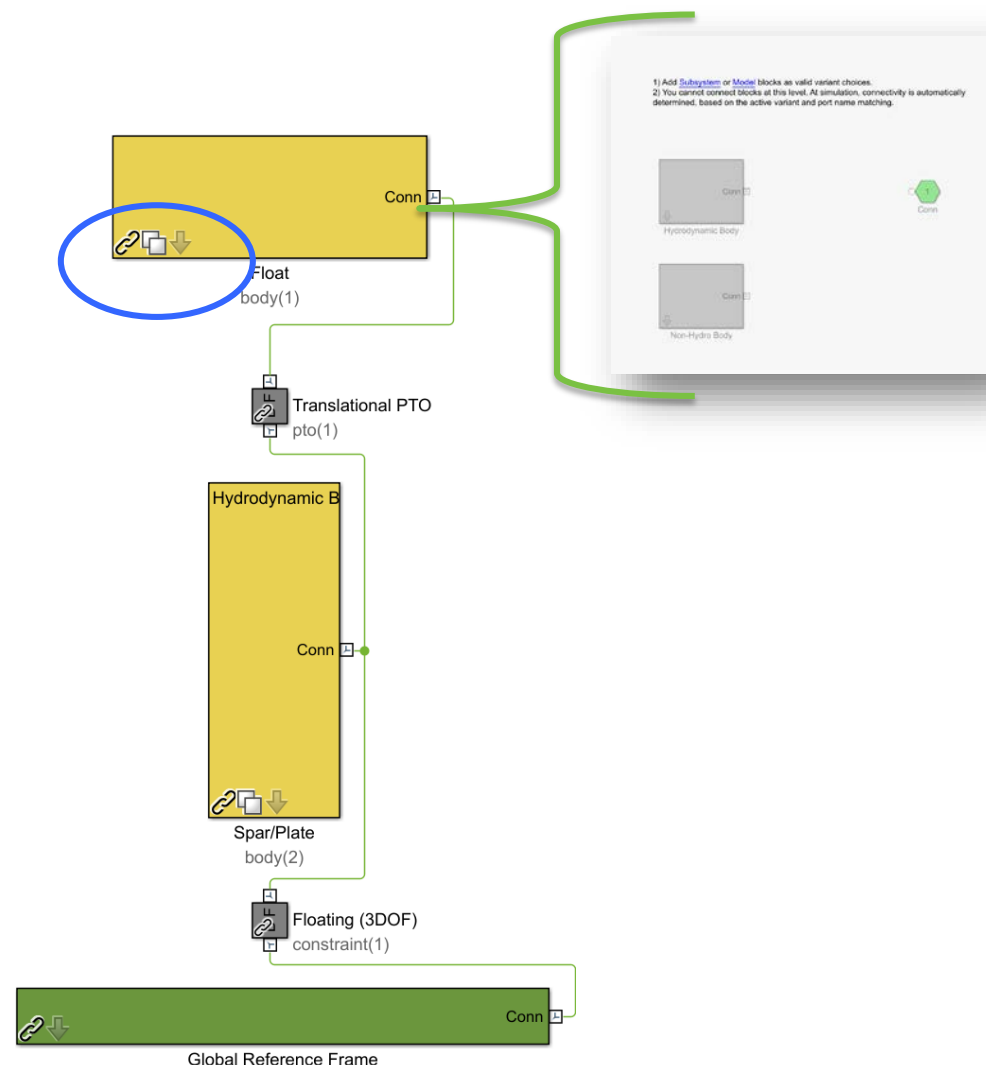
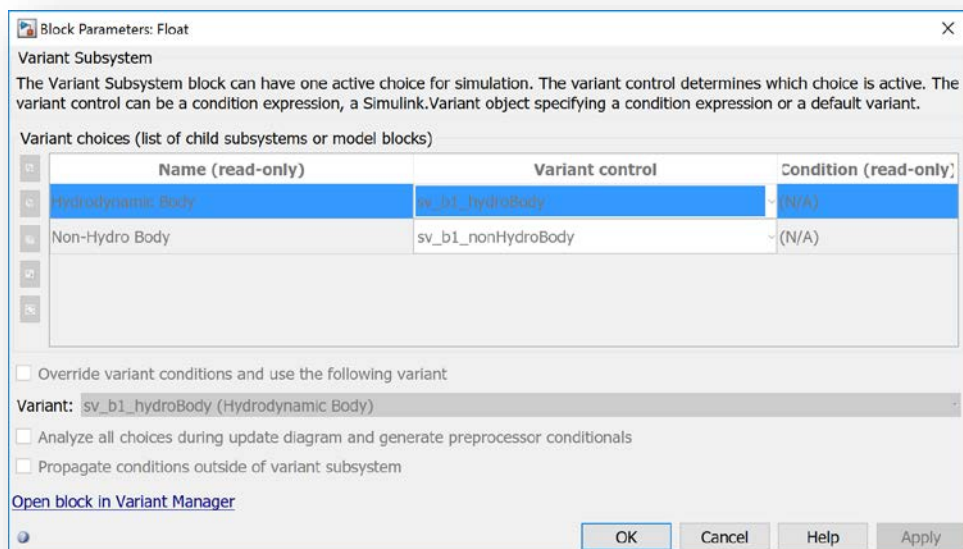
- Variant Subsystems
 - Variant subsystems let you provide multiple implementations for a subsystem where only one implementation is active during simulation.
 - You can programmatically swap out the active implementation and replace it with one of the other implementations without modifying the model.
- Denoted by double squares
- Double click on subsystem to view variant subsystem (unless masked)



<https://www.mathworks.com/help/simulink/examples/variant-subsystems.html>

WEC-Sim/source/lib/

- WEC-Sim Library Blocks contain many variant subsystems which are activated by WEC-Sim input file, *wecSimInputFile.m* in *wecSim.m*
- Right click on block with variant subsystem and select '**Block Parameters (Subsystem)**' to see **variant control variables**

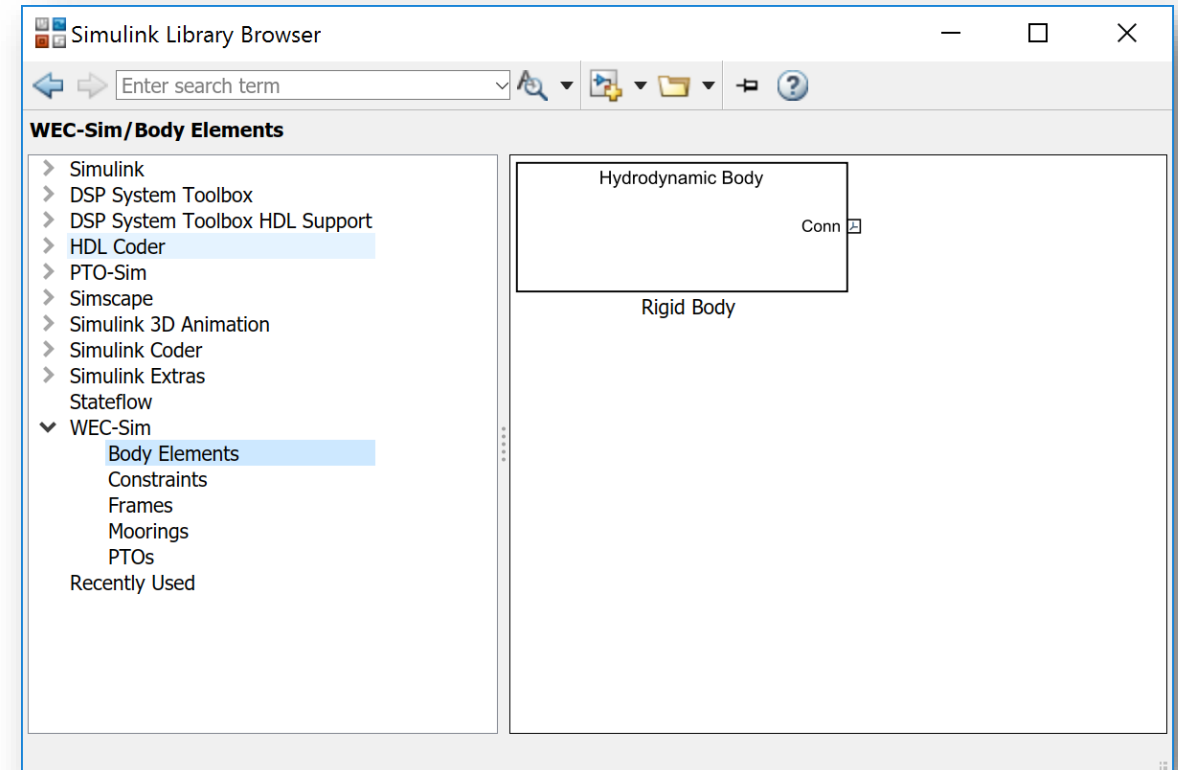
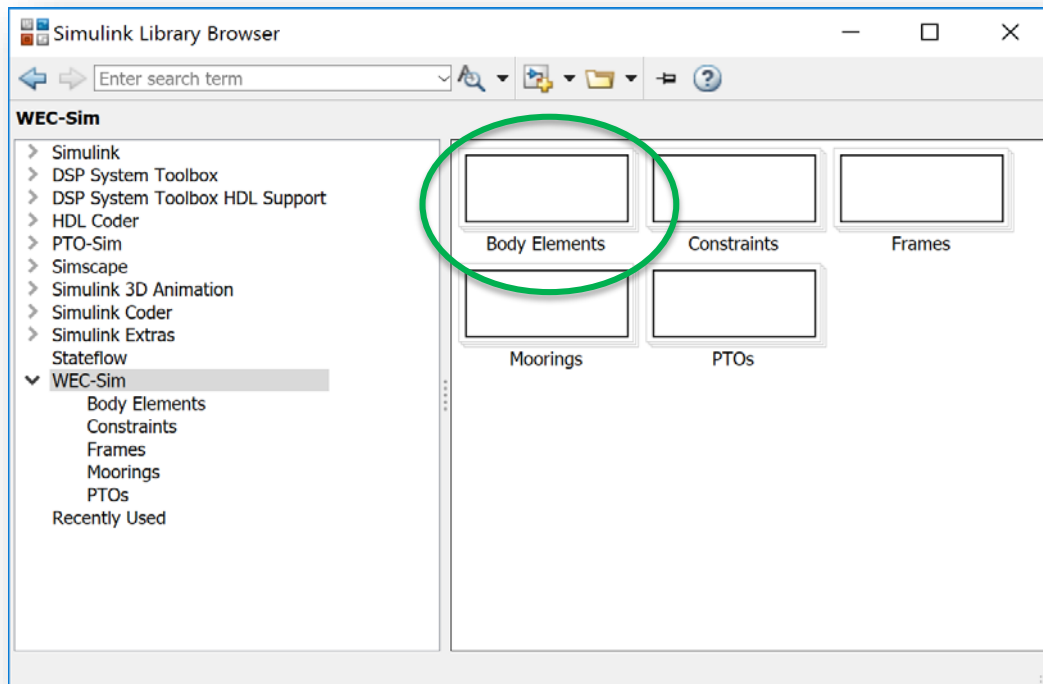


<https://www.mathworks.com/help/simulink/examples/variant-subsystems.ntml>

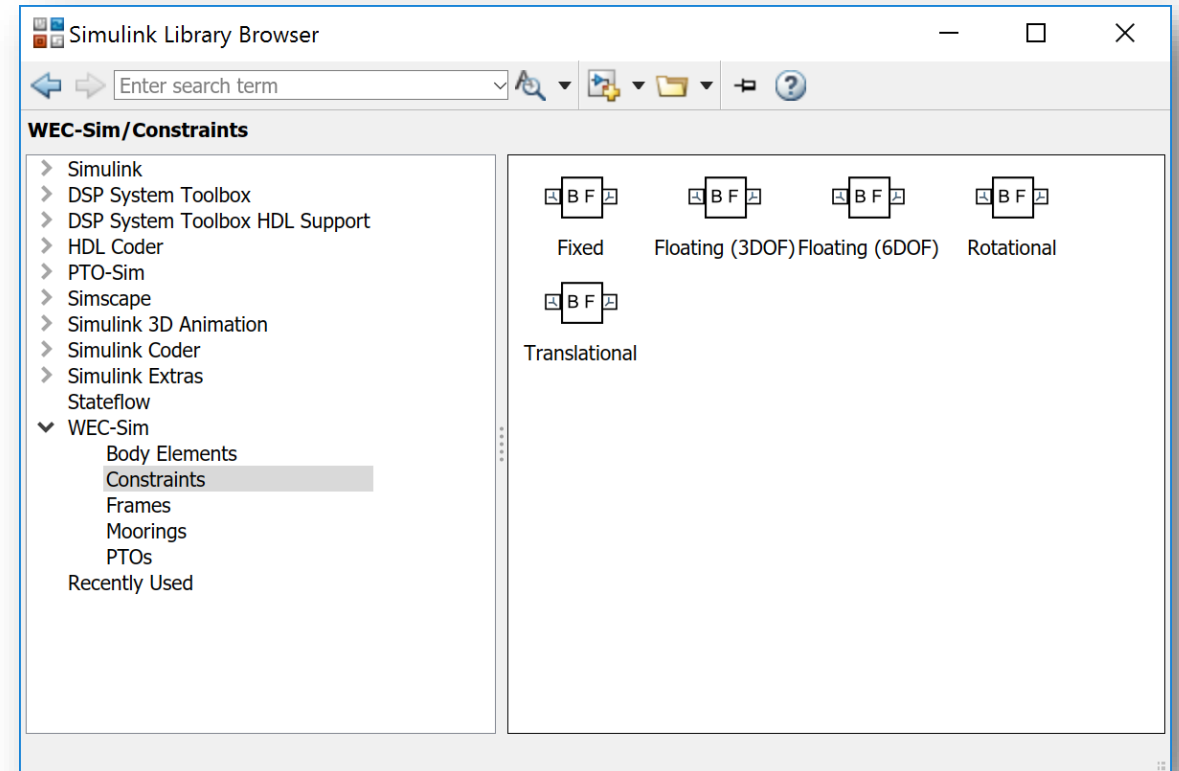
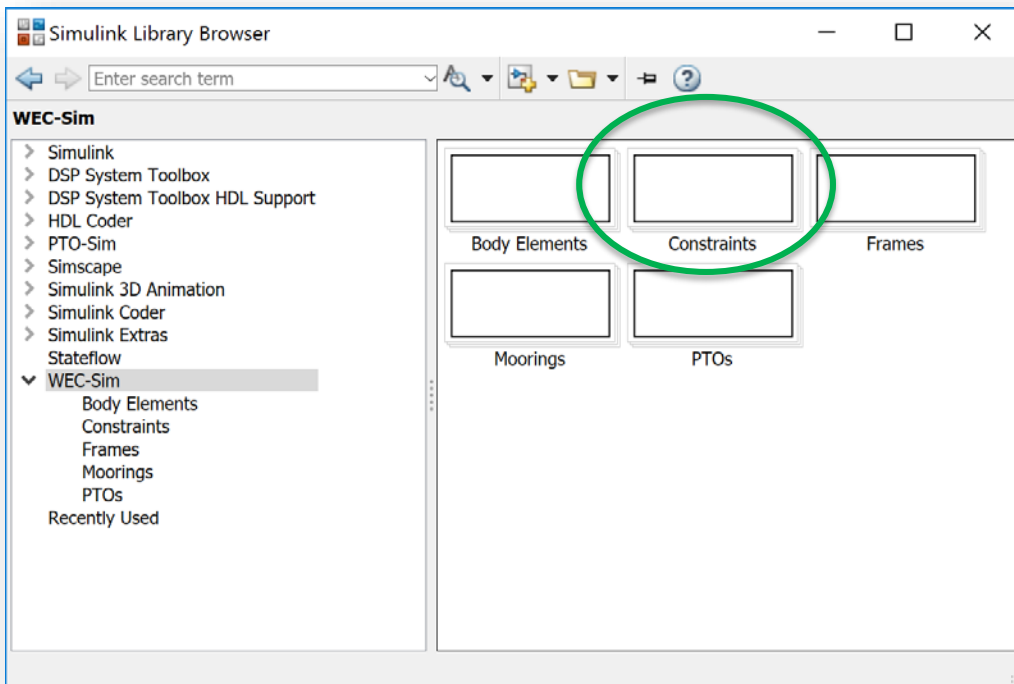
Variant Subsystem and Mask Example

WEC-Sim Library Block Types

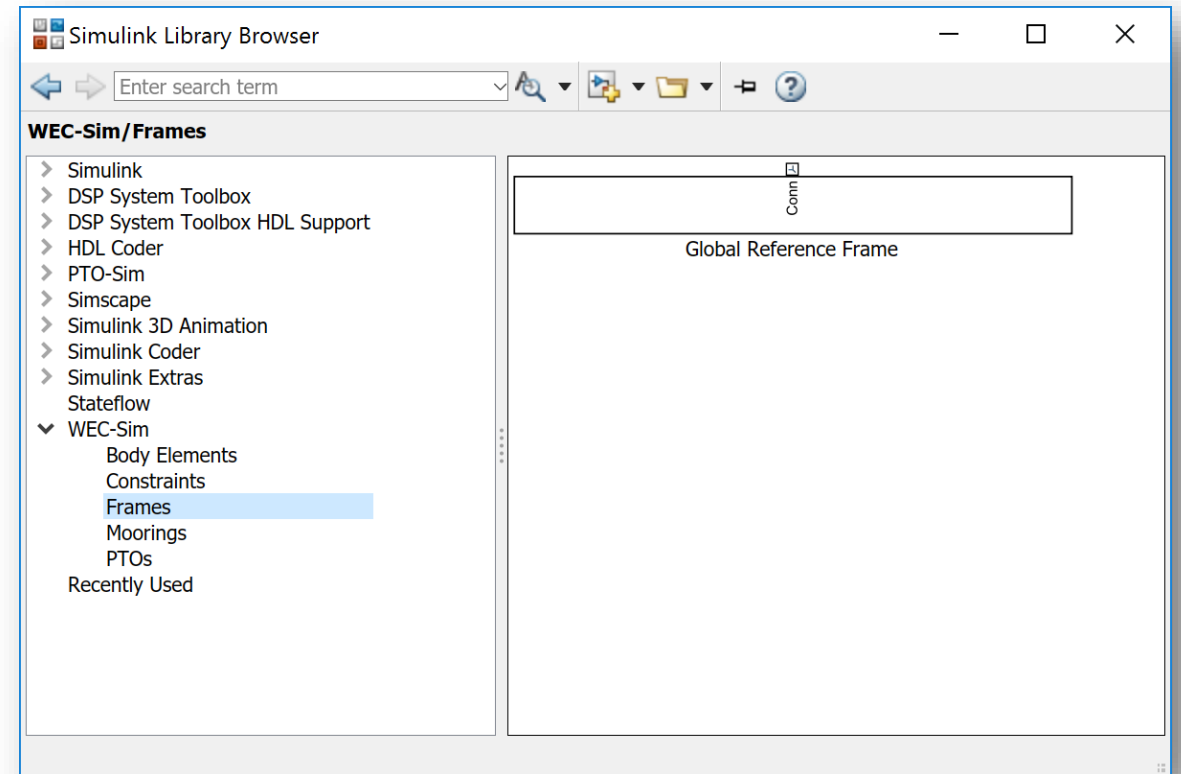
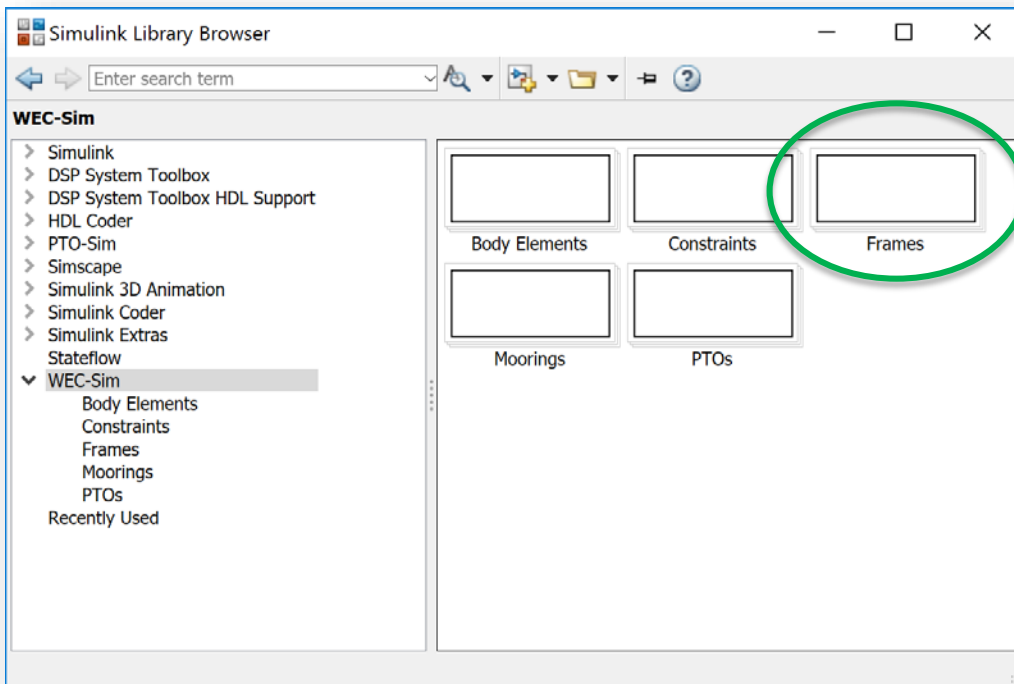
Body Elements



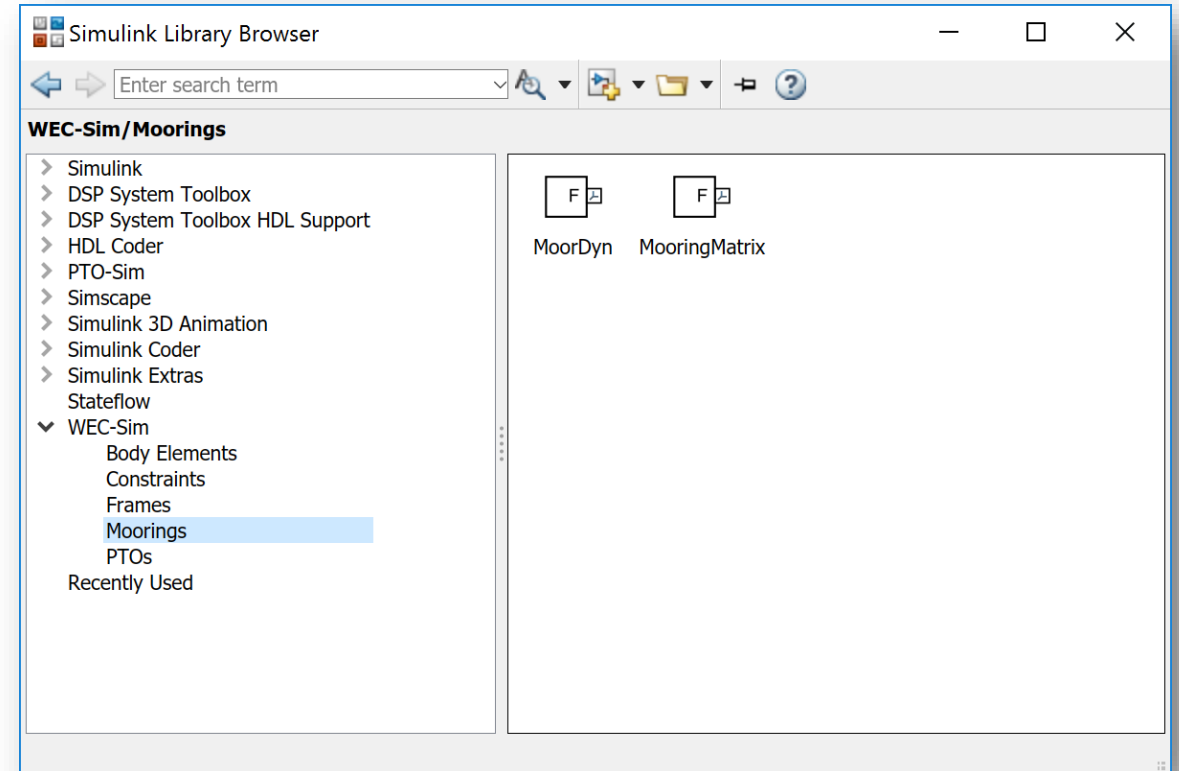
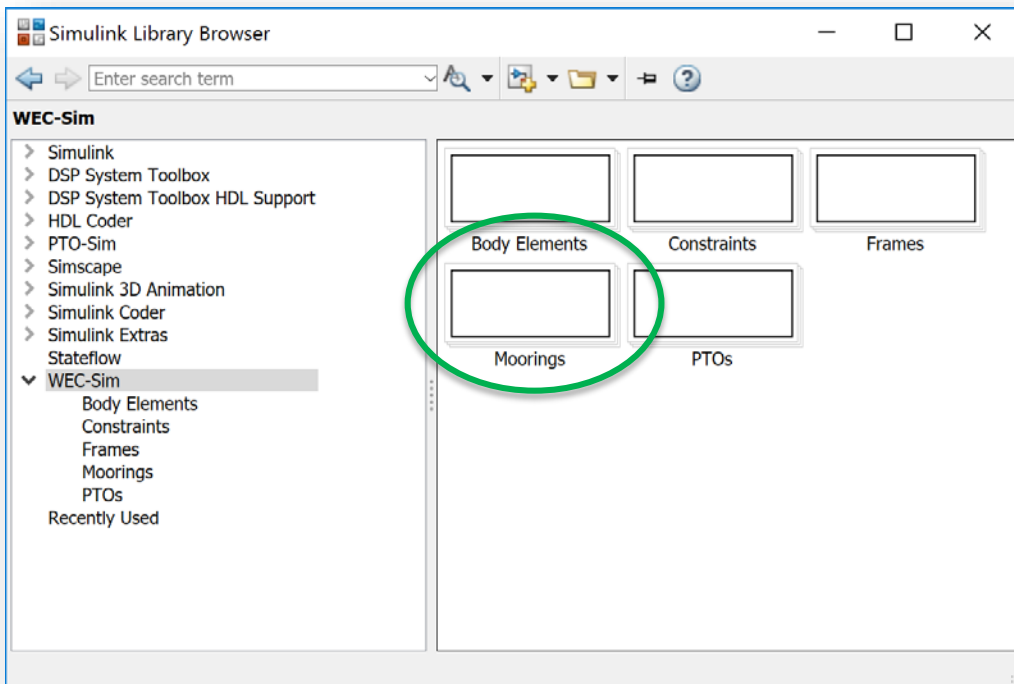
Constraints

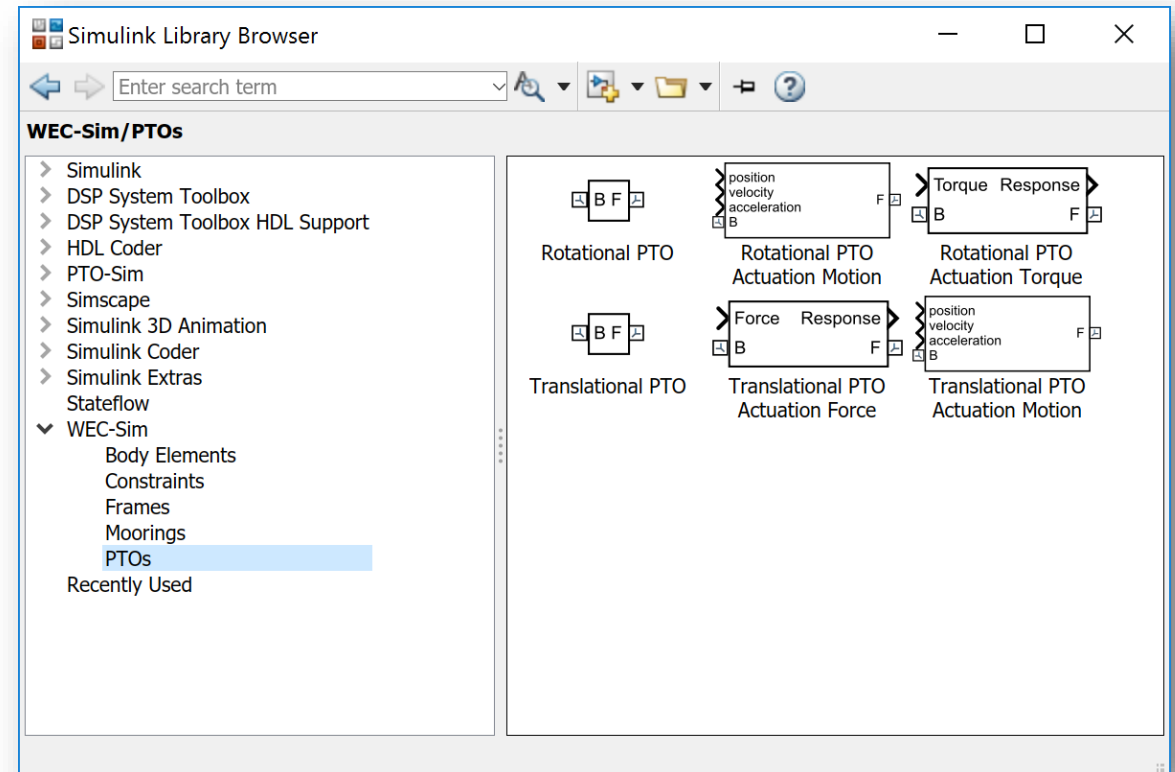
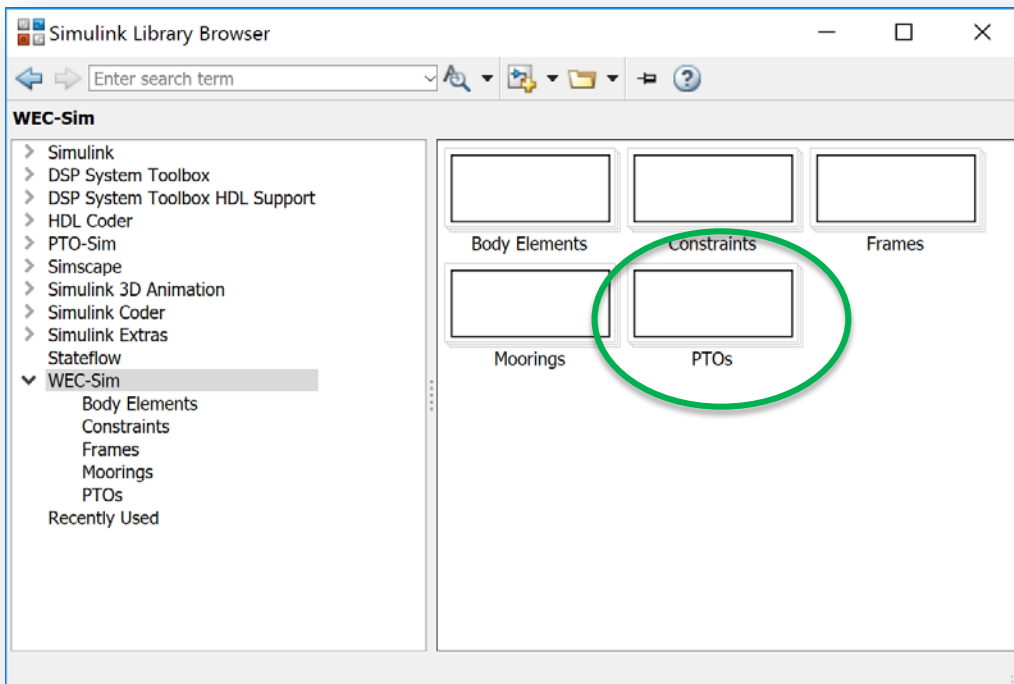


Frames



Moorings



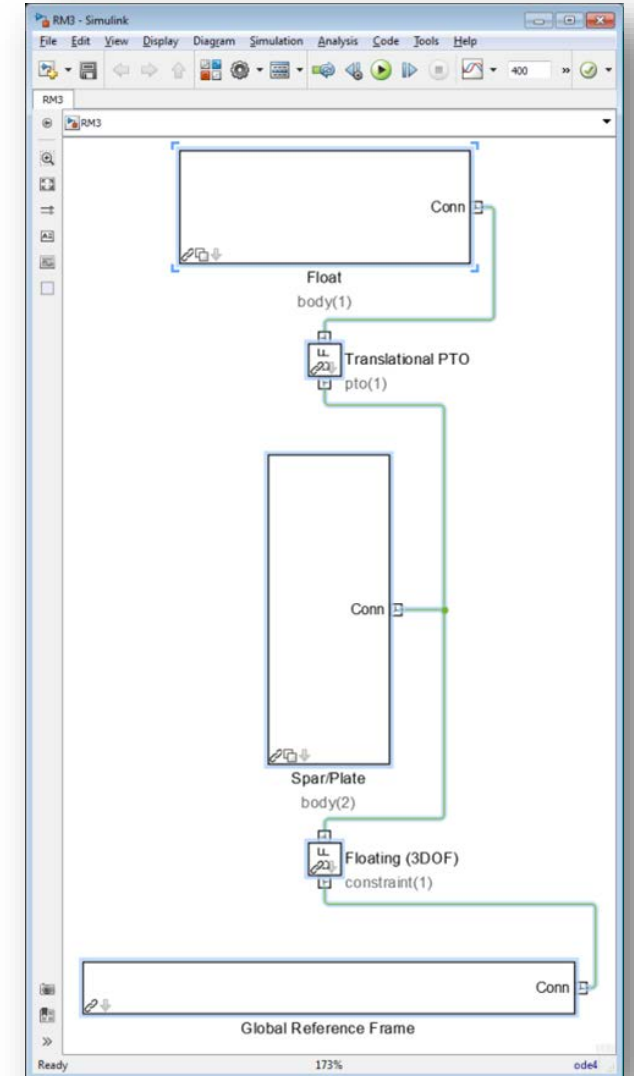
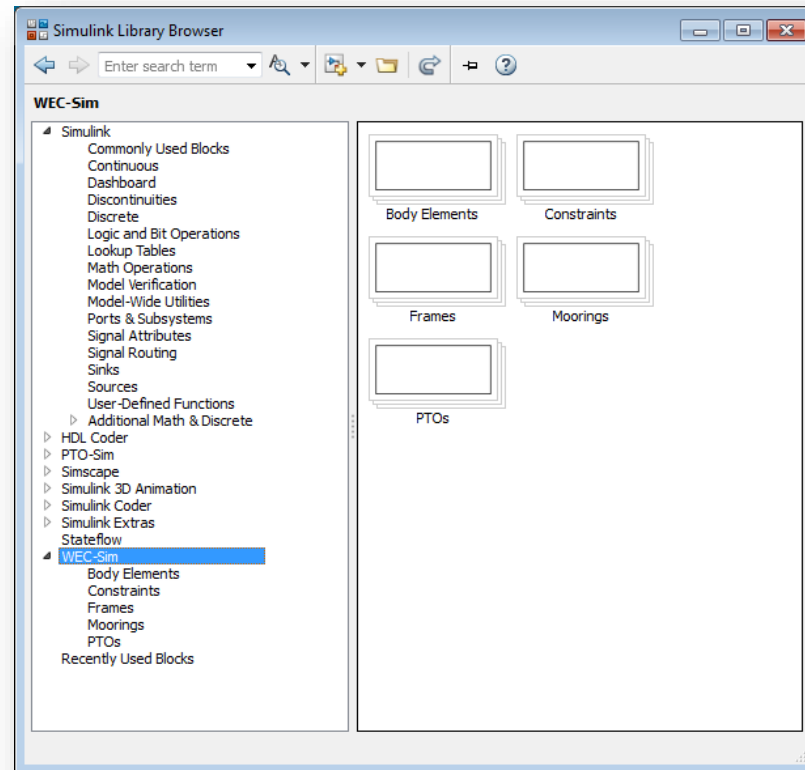


WEC-Sim Simulink File

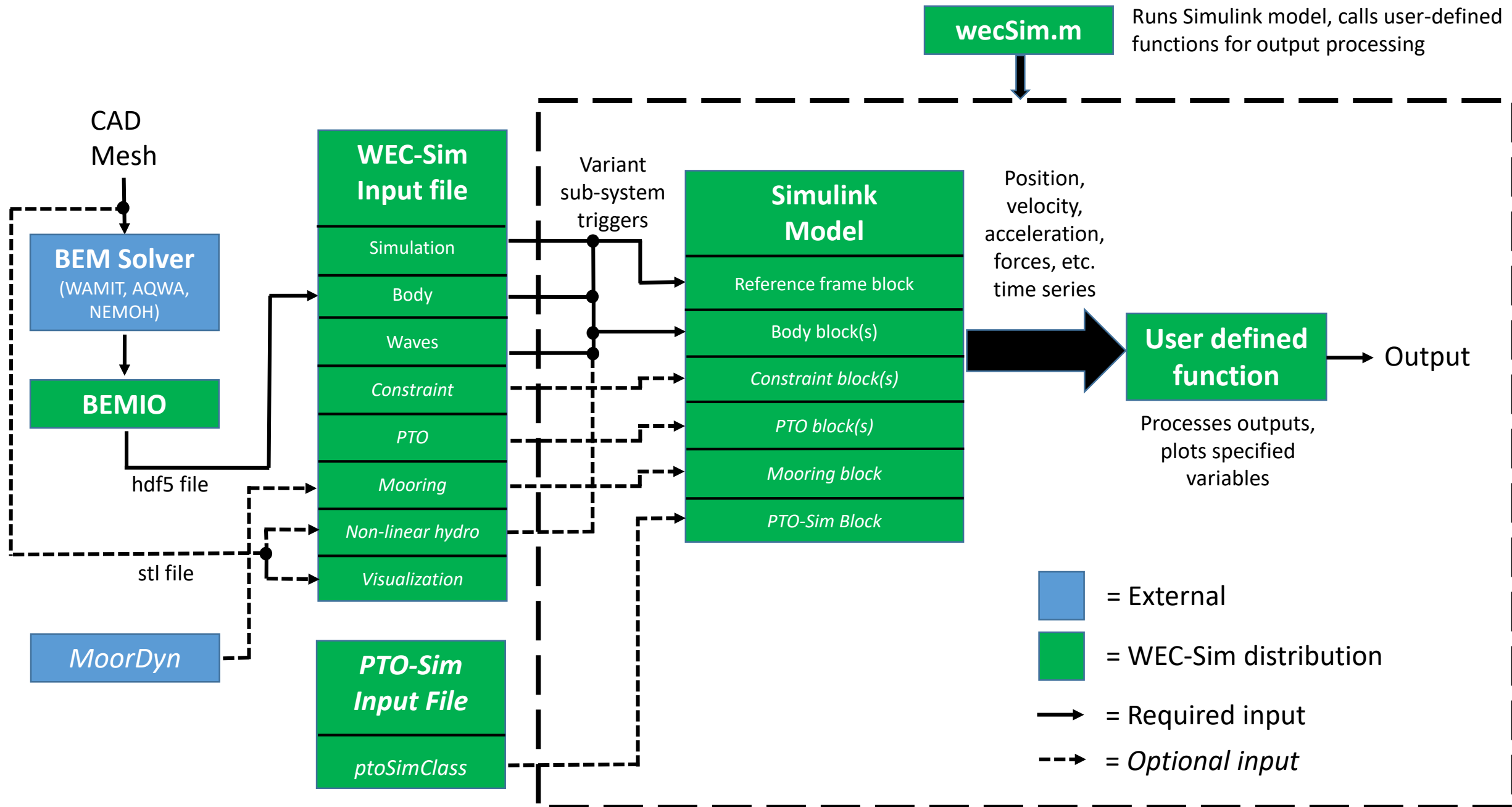
<Simulink_model_name>.slx

WEC-Sim Simulink Model

- Created with WEC-Sim Simulink Library Blocks
- Body Elements, Constraints, Frames, Moorings, PTOs



In conclusion...




Thank you!

All the webinar materials and recordings are available online:

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




WEC-Sim

[Getting Started](#)
[Examples](#)
[Theory](#)
[Code Structure](#)
[Advanced Features](#)

Webinars

- WEC-Sim Webinar #1 - BEMIO & MCR
- WEC-Sim Webinar #2 - Nonlinear Hydro, Non-Hydro & B2B
- WEC-Sim Webinar #3 - PTO and Control
- WEC-Sim Webinar #4 - Mooring and Visualization

[License](#)
[Publications](#)
[Release Notes](#)
[Contact Us](#)

[Docs](#)
[Webinars](#)

[View page source](#)

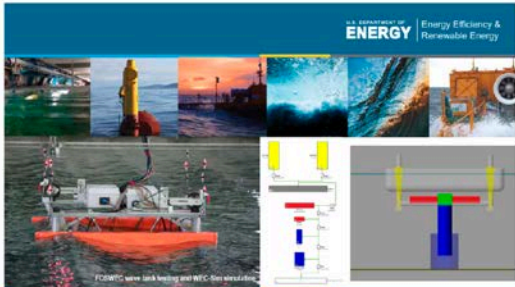
Webinars

The WEC-Sim team is hosting a series of advanced features webinars. Dates and topics are listed below. Once completed, the recordings and presentations will be posted to this page.

Date	Topic
April 18, 2017	BEMIO and MCR
May 24, 2017	Nonlinear Hydro, Non-Hydro, and B2B
June 13, 2017	PTO and Control
July 18, 2017	Mooring and Visualization
August 17, 2017	WEC-Sim Training Course

WEC-Sim Webinar #1 - BEMIO & MCR

The presentation and recordings of WEC-Sim Webinar #1 on BEMIO & MCR hosted on April 18, 2017 are available below. Download the presentation by clicking the image below.



The collage features a header with the text 'An Introduction to ENERGY Energy Efficiency & Renewable Energy'. Below the header are several images: a floating device on water, a turbine, a sunset over water, a close-up of a turbine, a schematic diagram of a device, and a photograph of a device on water. At the bottom of the collage, it says '©2017 WEC-Sim team. All rights reserved. WEC-Sim team'.

WEC-Sim Webinar #1

April 18, 2017
(Mingming Fu and Jonathan van Ri) (PDF)