SAND2015-10623C

Exceptional service in the national interest

Sandia National Laboratories

WEC-Sim Training Course

DRY RUN 10/7/2015

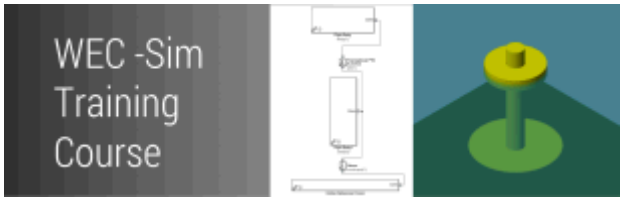
K. Ruehl and Carlos Michelen

INORE 2015, Friday Harbor Labs

Sandia National Laboratories NREL U.S. DEPARTMENT OF ENERGY NNSA National Nuclear Security Administration

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000. SAND 2010-5138P.

Overview



- WEC-Sim
- Theory
- Getting Started
- Workflow
- Code Structure
- Output Structure
- Tutorials
- Collaboration (optional)

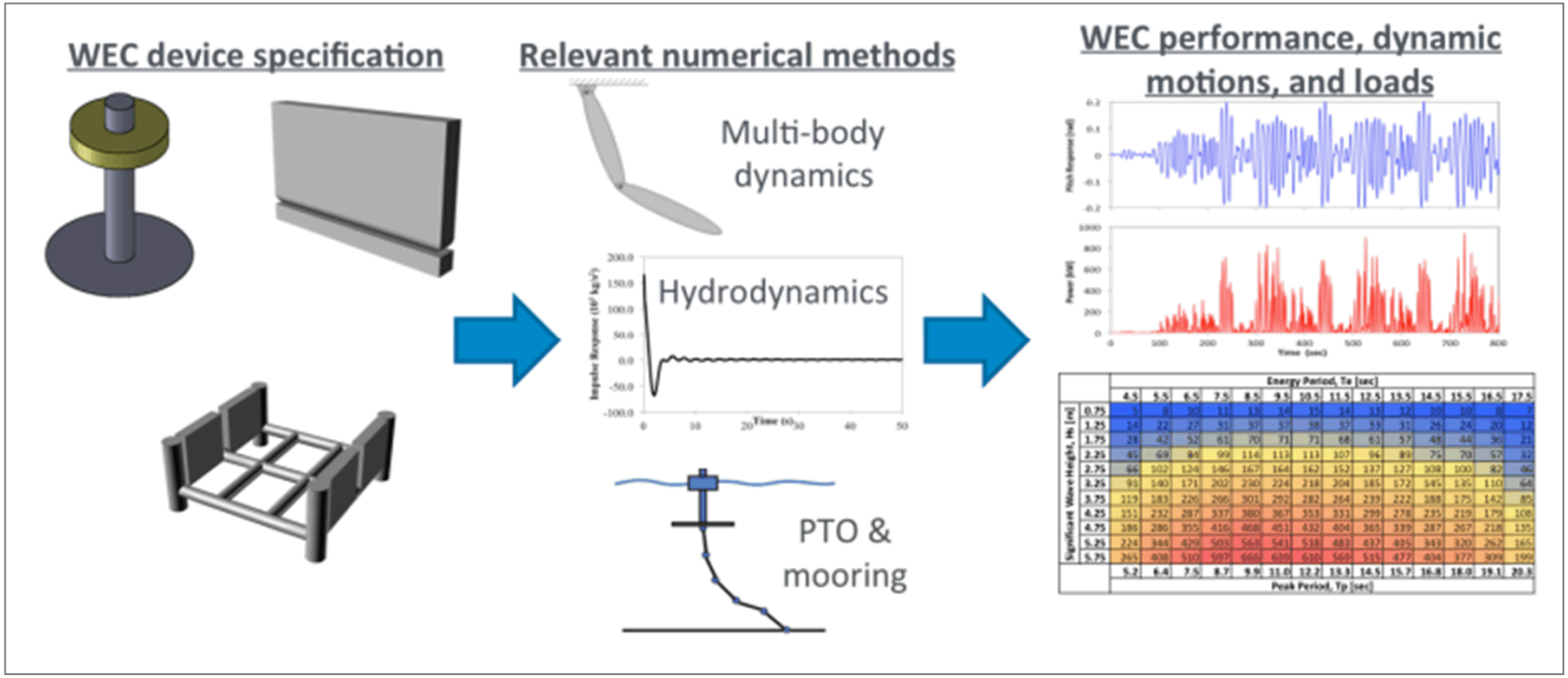
What is WEC-Sim?

WEC-Sim (Wave Energy Converter SIMulator)

- US Department of Energy funded
- Joint Sandia/NREL project
- Open source code developed in MATLAB/SIMULINK
- Time-domain rigid body equation of motion solver based on Cummins' formulation
- Simulates wave energy converter dynamics in operational waves
- First Release: v1.0 in June 2014
- Current Release: v1.2 in July 2015

Why use WEC-Sim?

- WEC-Sim has the ability to model devices that are comprised of rigid bodies, power-take-off systems, and mooring systems.
- WEC-Sim uses hydrodynamic coefficients derived from frequency-domain boundary element simulations to model the relevant hydrodynamics.
- Time-domain simulations are performed by solving the governing WEC equations of motion in 6 degrees-of-freedom.



Sandia National Laboratories

THEORY

WEC-Sim Theory

- Dynamics simulated by solving time-domain equation of motion (Cummins, 1962)

$$m\ddot{x}(t) = f_{hs}(t) + f_{ex}(t) + f_{rad}(t) + f_{v}(t) + f_{pto}(t) + f_{m}(t)$$

Hydrostatic restoring force

Wave excitation & diffraction force (from BEM simulations)

Radiation force: added mass and radiation damping (from BEM simulations)

Viscous force

Power take-off force

Mooring force

- Use radiation and diffraction method and calculate the hydrodynamic forces from frequency-domain Boundary Element Method (BEM)

$$f_{rad}(t) = -A_{\infty}\ddot{X} - \int_0^t K(t-\tau)\dot{X}(\tau)d\tau$$

BEM BEM

$$f_{ex}(t) = \Re\left[R_f F_X(\omega_\tau) e^{i(\omega_\tau t+\phi)} \int_0^\infty \sqrt{2S(\omega_\tau)d\omega_\tau}\right]$$

BEM

$$= \int_{-\infty}^{\infty} \eta(\tau) f_e(t-\tau)d\tau$$

BEM

  - State space
  - Constant coeff

WEC-Sim Theory - Cont

- Calculated in BEMIO

Excitation IRF:

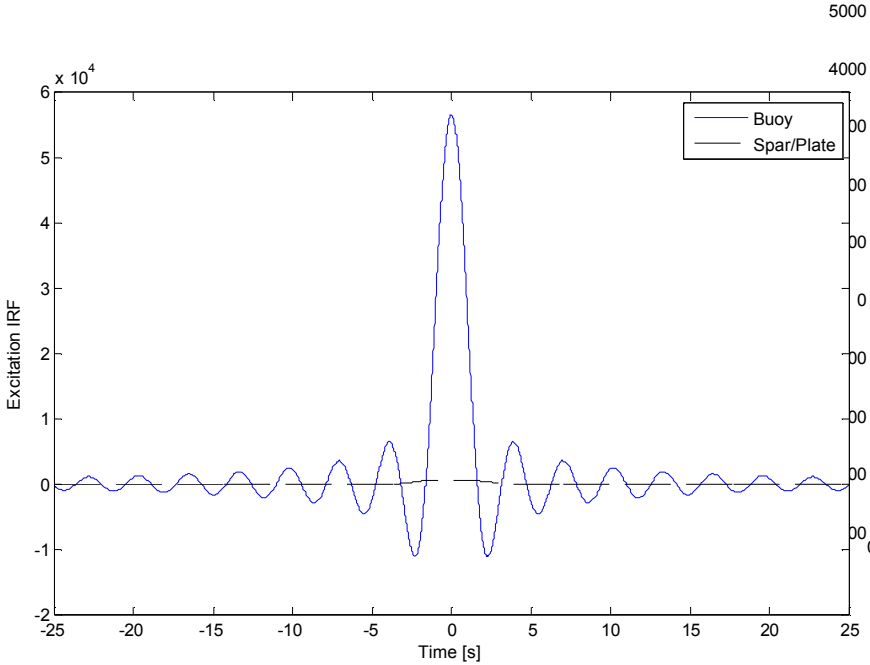
$$f_e(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} f_e(i\omega)e^{i\omega t}d\omega$$

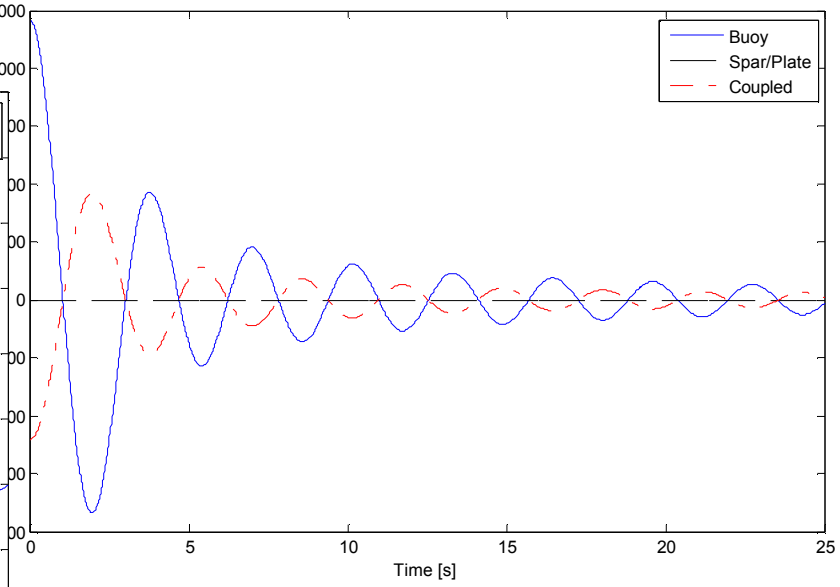
BEM

Radiation IRF:

$$f_r(t) = \frac{2}{\pi}\int_{0}^{\infty} f_r(\omega)cos(\omega t)d\omega$$

BEM





Infinite Added Mass: A_∞

BEM

Sandia National Laboratories

GETTING STARTED

MATLAB Toolboxes

- List of required toolboxes

Required Toolbox	Supported Version
MATLAB	Version 8.4 (R2014b)
Simulink	Version 8.4 (R2014b)
SimMechanics	Version 4.5 (R2014b)
Simscape	Version 3.12 (R2014b)

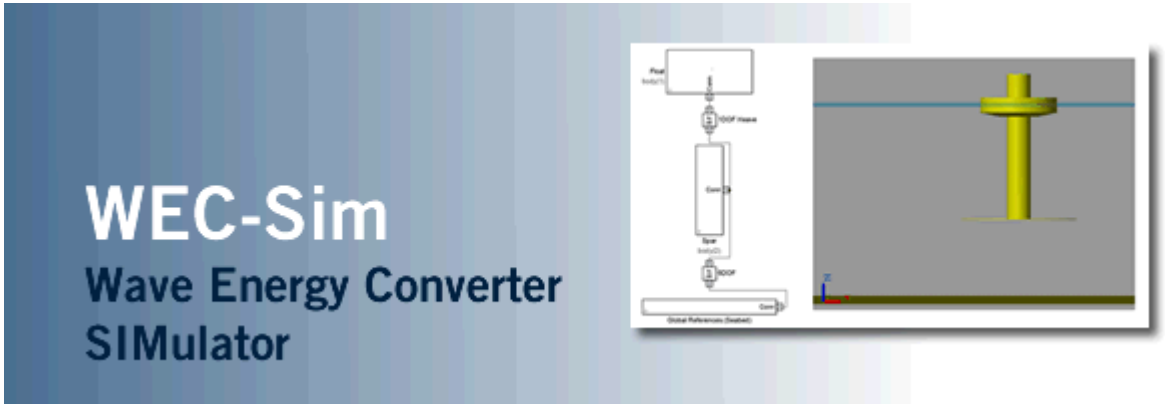
- Ensure required toolboxes are installed

  >> ver

```
Command Window
Adding WEC-Sim to Path
>> ver
-----------------------------------------------------------------------------------------------
MATLAB Version: 8.4.0.150421 (R2014b)
MATLAB License Number: 680411
Operating System: Microsoft Windows 7 Enterprise  Version 6.1 (Build 7601: Service Pack 1)
Java Version: Java 1.7.0_11-b21 with Oracle Corporation Java HotSpot(TM) 64-Bit Server VM
-----------------------------------------------------------------------------------------------
MATLAB                                                Version 8.4        (R2014b)
Simulink                                              Version 8.4        (R2014b)
SimMechanics                                          Version 4.5        (R2014b)
Simscape                                              Version 3.12       (R2014b)
fx >>
```

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

WEC-Sim Open Source Code



- Code Download
  https://github.com/WEC-Sim/WEC-Sim
- Online Documentation
  http://wec-sim.github.io/WEC-Sim/index.html
- Online Forum
  https://github.com/WEC-Sim/WEC-Sim/issues

Adding WEC-Sim to path

- Add WEC-Sim source code to path **WEC-Sim/source**
  - Open the wecSimStartup.m file
  - Paste the wecSimStartup.m into the startup.m file

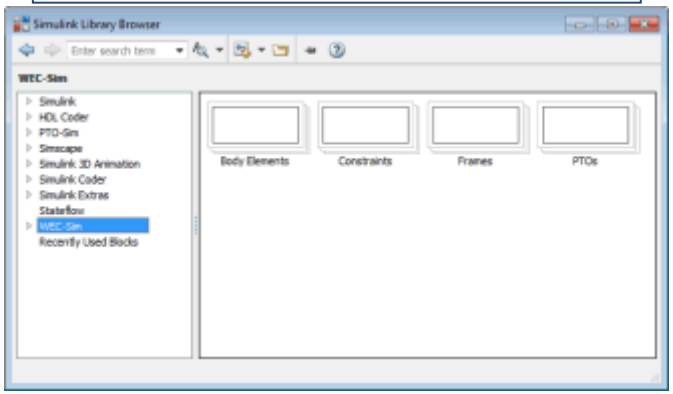
    >> open startup.m
  - Set the wecSimPath variable
  - Verify path was added

    >> path
  - Verify WEC-Sim Simulink library available

    >> simulink

    (Refresh library browser by hitting F5)

```
>> path

        MATLABPATH

/Users/username/Github/WEC-Sim/source
/Users/username/Github/WEC-Sim/source/functions
/Users/username/Github/WEC-Sim/source/lib
/Users/username/Github/WEC-Sim/source/lib/WEC-Sim
/Users/username/Github/WEC-Sim/source/objects
/Users/username/Github/WEC-Sim
/Users/username/Documents/MATLAB
```

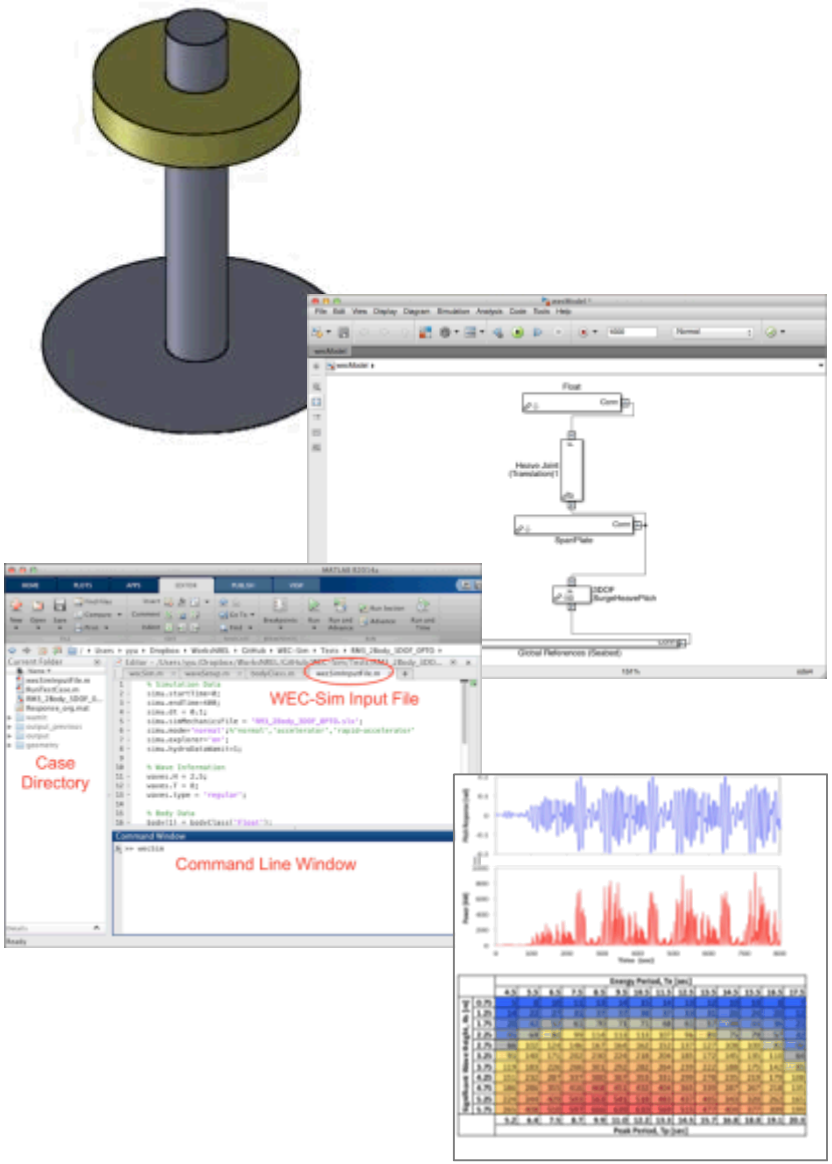


http://wec-sim.github.io/WEC-Sim/getting_started.html#step-1-add-wec-sim-source-code-to-matlab-path

Sandia National Laboratories

WORKFLOW

Workflow



Pre-process

- Define WEC geometry
  - 3D CAD model
  - Mass properties
- Run BEM code
- BEMIO
  - *Check BEMIO data*

WEC-Sim

- Build Simulink model
- Write input file
  - Wave conditions
  - Simulation parameters
  - Body properties
  - PTO/constraints
- Run WEC-Sim

Post-Process

- Analyze WEC-Sim results

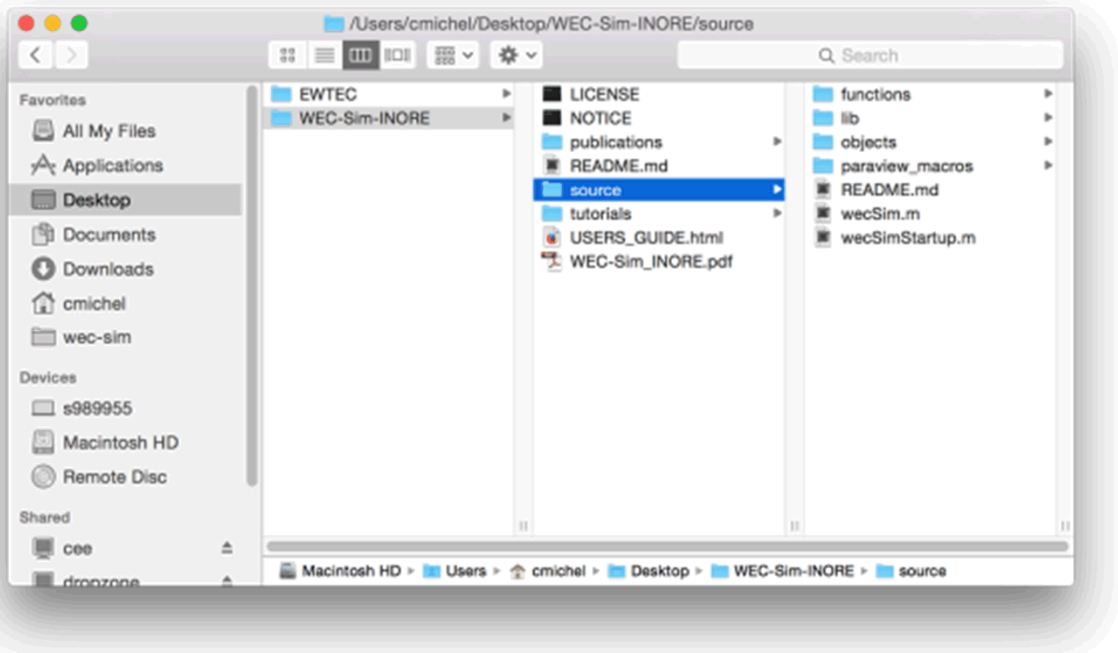
Sandia National Laboratories

CODE STRUCTURE

Code Structure

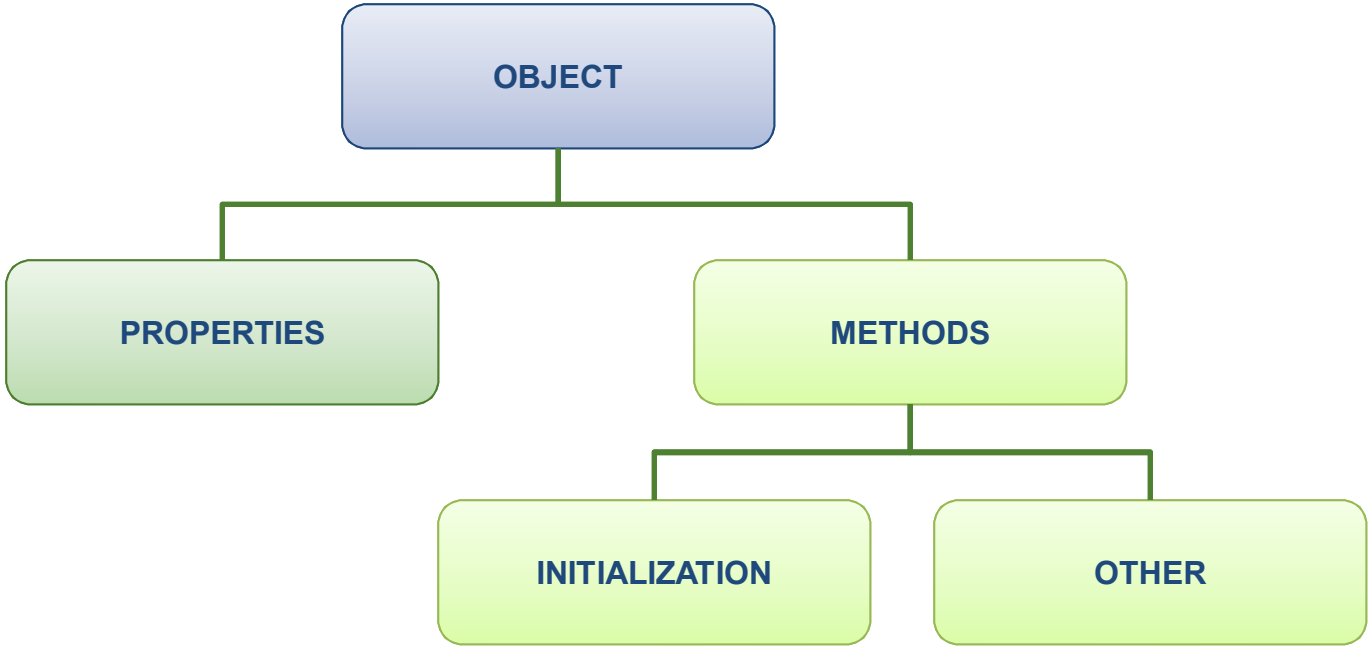
- WEC-Sim source code:
  - Simulink Library
  - Matlab objects .m files
  - wecSim.m (“executable”)

/source



Code Structure

- Object Oriented Programming



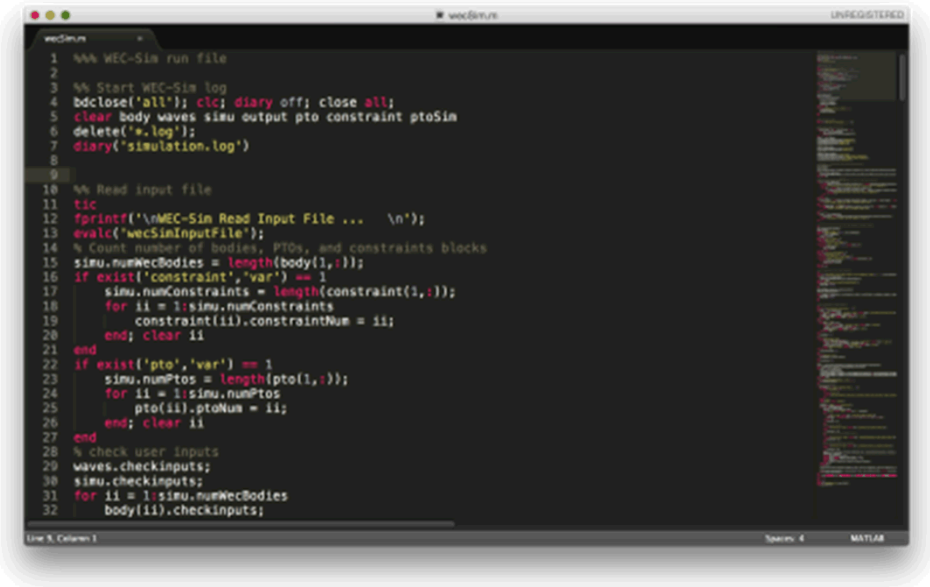
Code Structure

- WEC-Sim objects:
  - bodyClass
  - waveClass
  - simulationClass
  - ptoClass
  - constraintClass
  - responseClass

>> doc ptoClass
/source/objects/ptoClass.m

Code Structure

- wecSim.m
  - “executable”
  - Reads input file
  - Uses the object’s properties and methods for pre- and post-processing
  - Runs the simulation



```
%%% WEC-Sim run file

%% Start WEC-Sim log
bdclose('all'); clc; diary off; close all;
clear body waves simu output pto constraint ptoSim
delete('*.log');
diary('simulation.log')


%% Read input file
tic
fprintf('\nWEC-Sim Read Input File ...   \n');
evalc('wecSimInputFile');
% Count number of bodies, PTOs, and constraints blocks
simu.numWecBodies = length(body(1,:));
if exist('constraint','var') == 1
    simu.numConstraints = length(constraint(1,:));
    for ii = 1:simu.numConstraints
        constraint(ii).constraintNum = ii;
    end; clear ii
end
if exist('pto','var') == 1
    simu.numPtos = length(pto(1,:));
    for ii = 1:simu.numPtos
        pto(ii).ptoNum = ii;
    end; clear ii
end
% check user inputs
waves.checkinputs;
simu.checkinputs;
for ii = 1:simu.numWecBodies
    body(ii).checkinputs;
```

/source/wecSim.m

Code Structure

- Input File
  - Object initialization
  - Object properties
    - Default values
- Object Initialization in Input File:

  >> simu = simulationClass();

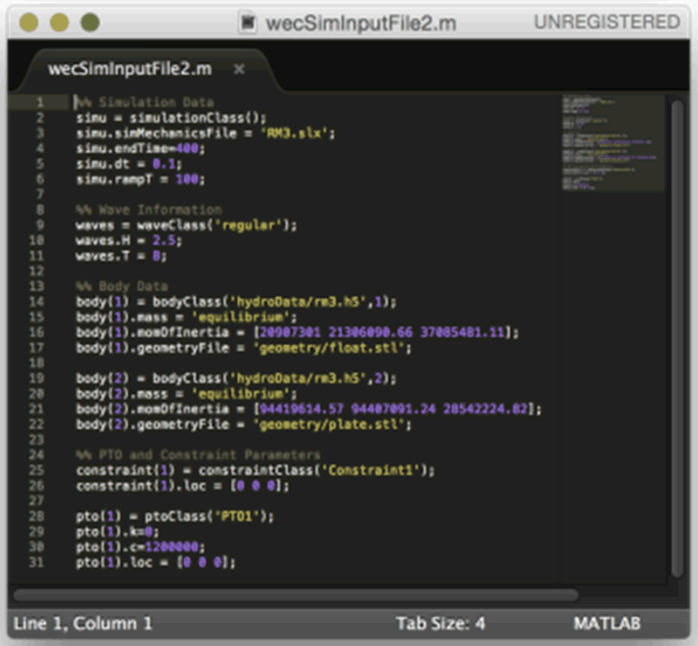
  >> waves = waveClass('type');

  >> body(i) = bodyClass('h5file',i);

  >> pto(i) = ptoClass('name');

  >> constraint(i) = constraintClass('name');

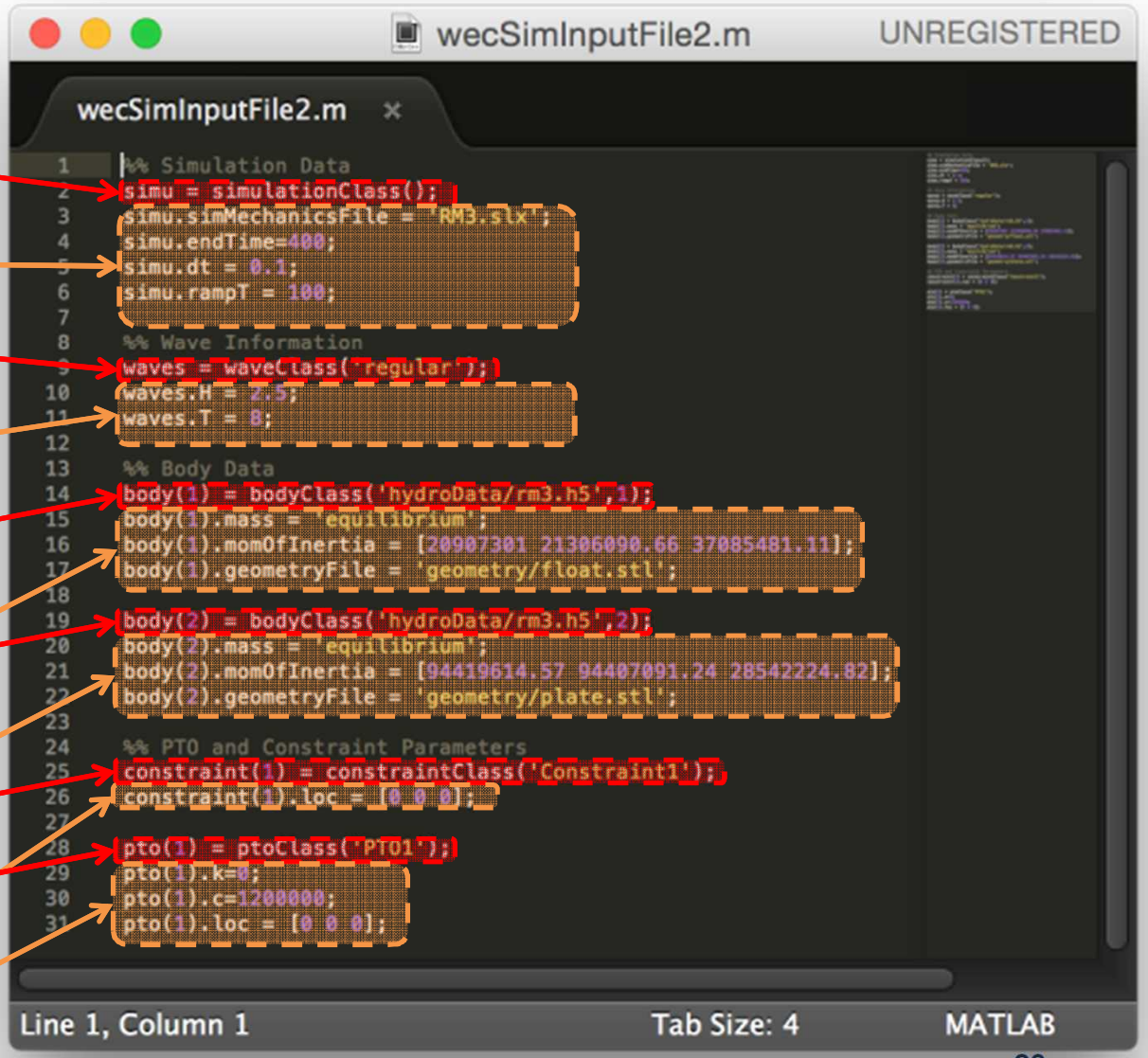
/tutorials/RM3/wecSimInputFile.m



Code Structure

Sandia National Laboratories

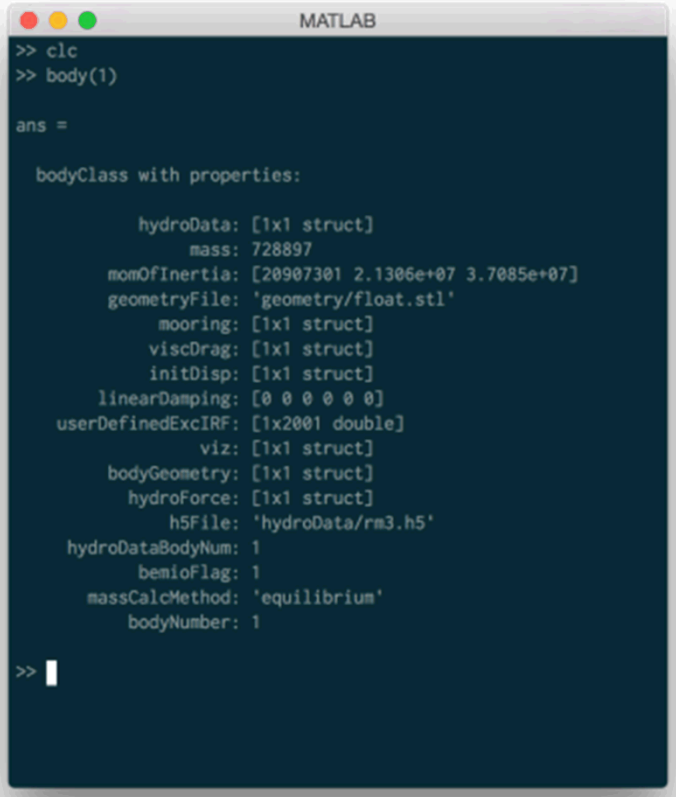
- Initialize simulation class
- Sets other simulation class properties
- Initialize wave class
- Sets other wave class properties
- Initialize one instance of body class
- Sets other properties for
- Initialize second instance of body class
- Sets other properties for
- Initialize one instance of constraint class
- Initialize one instance of pto class
- Sets other properties for first instance of pto class



20

Code Structure

- Body Class
  - Required:
    - mass: value, ‘equilibrium’, ‘fixed’
    - momOfInertia (except for ‘fixed’ mass)
    - geometryFile
    - h5File, hydroDataBodyNum (initialization)
  - Optional:
    - mooring
    - viscDrag
    - initDisp
    - linearDamping
    - Viz
    - bemioFlag
  - Useful input information:
    - hydroData
    - hydroForce



21

Code Structure

- Wave Class
  - Type:
    - noWave
    - noWaveCIC
    - regular
    - regularCIC
    - Irregular
    - irregularImport
    - userDefined
  - Different variables required for each type of wave

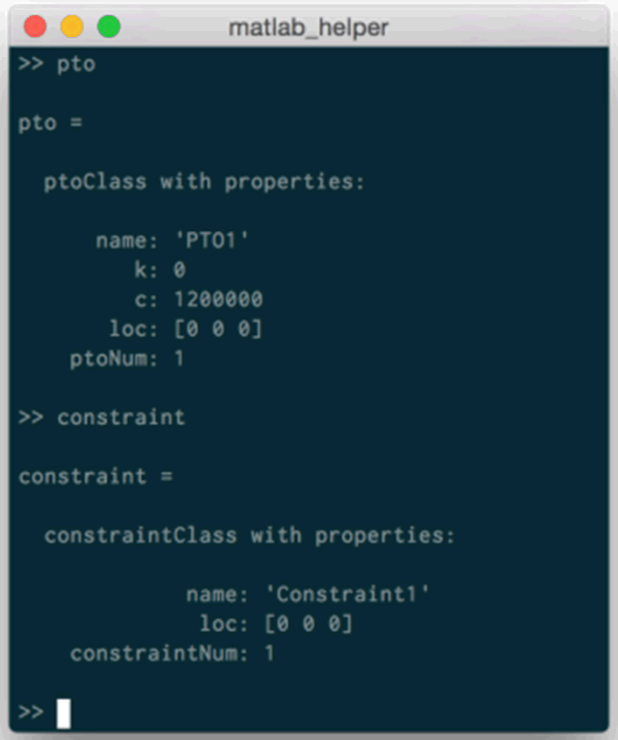
Code Structure

- Simulation Class
  - Contains all simulation options (time, fluid properties, solver, visualization, etc.)
  - Contains flags for different options (state space, non-linear hydro, etc.)
  - Properties:
    - simMechanicsFile
    - startTime, endTime, dt, rampT, CITime
    - ssCalc, nlHydro
    - mode, solver
    - explorer, paraview, domainSize
    - rho, g

Code Structure

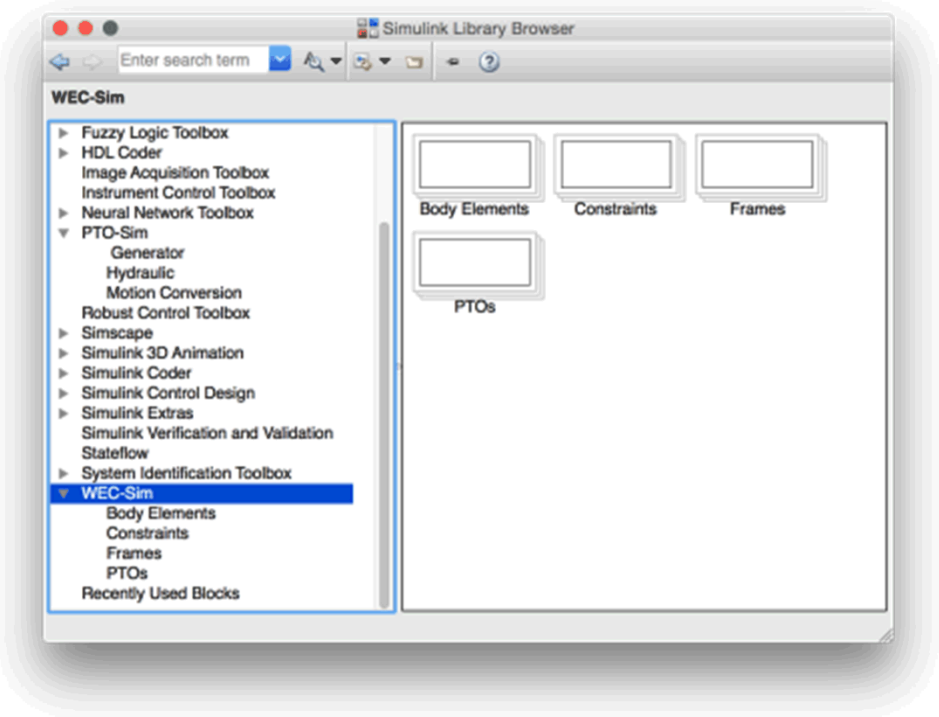
- Constraint Class
  - name
  - loc
- PTO Class
  - name
  - loc
  - k
  - c

- loc required for rotational PTOs & Constraints



Code Structure

- WEC-Sim Library
  - Drag & Drop library
  - Mask
- Simulink Model
  - Made of WEC-Sim library blocks
  - Bodies, PTOs, and constraints need to be numbered



/source/lib/WEC-Sim/WECSim_Lib.slx

25

Sandia National Laboratories

OUTPUT STRUCTURE

Output Structure

- `output` variable = responseClass instance
  - Contains all time series from simulation
- Properties:
  - bodies
  - ptos
  - constraints
  - ptosim

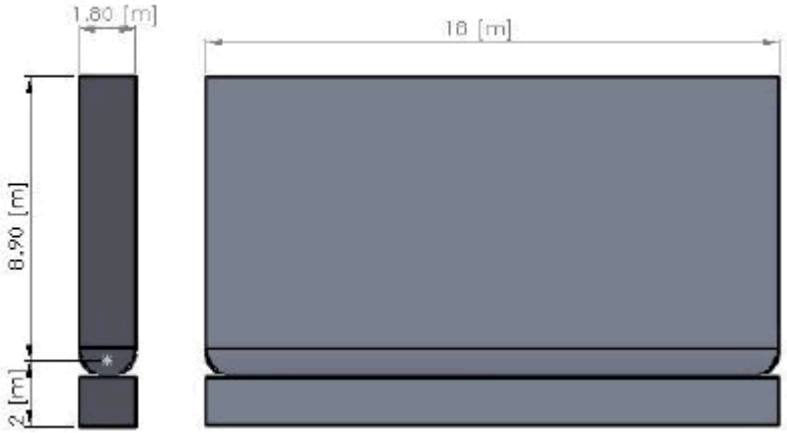
/tutorials/RM3/output/RM3_matlabWorkspace.mat
http://wec-sim.github.io/WEC-Sim/code_structure.html#output-structure

Sandia National Laboratories

TUTORIALS

Sandia National Laboratories

Tutorial: Oscillating Surge WEC



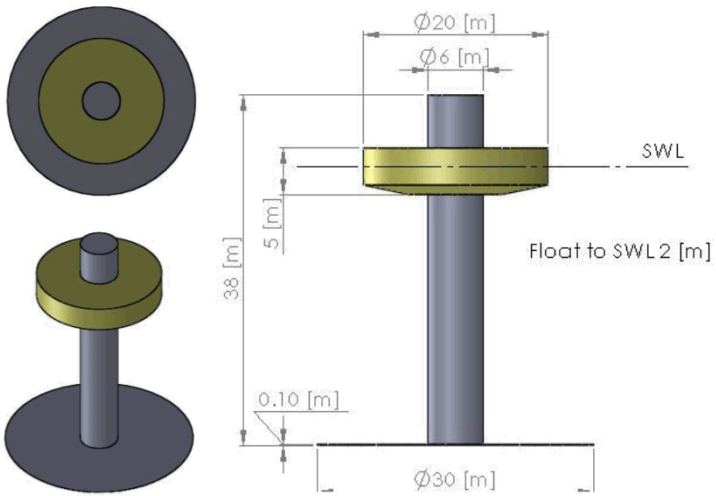
Flap Full Scale Properties		
CG [m]	Mass [kg]	Pitch Moment of Inertia [kg-m^2]
0	127,000	1,850,000
0		
-3.9		

OSWEC Tutorial

Sandia National Laboratories

WEC-Sim/\tutorials\OSWEC

Tutorial: Two-Body Floating-Point Absorber



Float Full Scale Properties				
CG [m]	Mass [tonne]	Moment of Inertia [kg-m^2]		
0	727.01	20907301	0	0
0		0	21306090.7	4304.89323
-0.72105		0	4304.89323	37085481.1
Plate Full Scale Properties				
CG [m]	Mass [tonne]	Moment of Inertia [kg-m^2]		
0	878.30	94419614.6	0	0
0		0	94407091.2	217592.785
-21.285		0	217592.785	28542224.8

RM3 Tutorial

Sandia National Laboratories

WEC-Sim/\tutorials\RM3

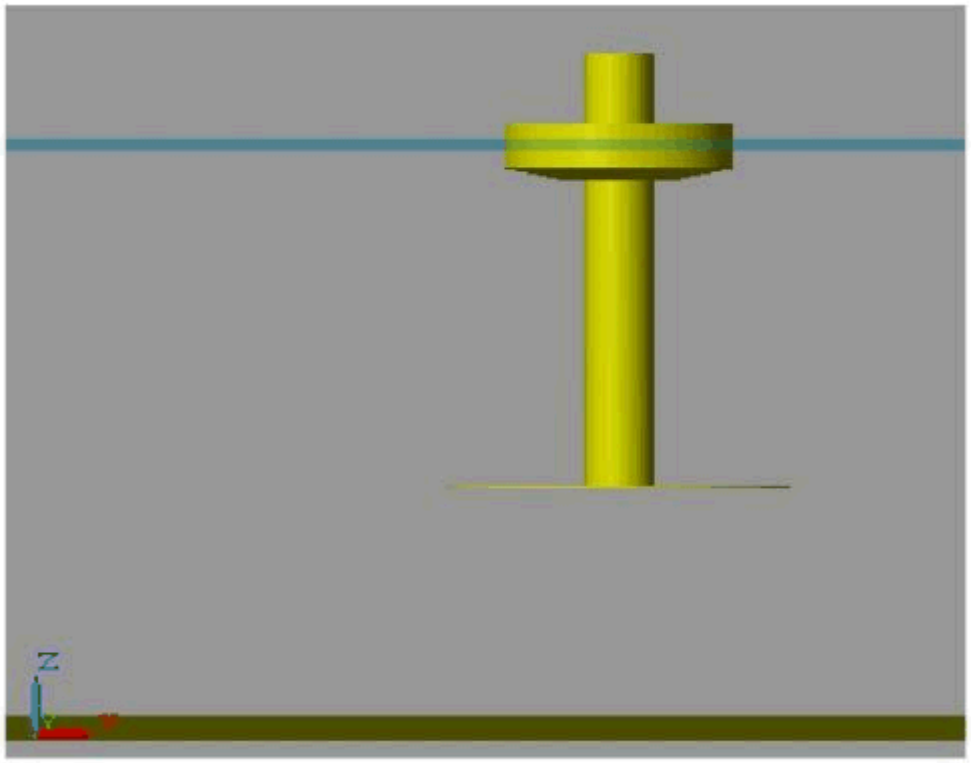
WEC-Sim Code

Sandia National Laboratories

Run WEC-Sim Input File

```
% Simulation Data
simu.startTime=0;                       % Simulation Start Time [s]
simu.endTime=400;                       % Simulation End Time [s]
simu.dt = 0.1;                          % Simulation Delta_Time [s]
simu.simMechanicsFile = 'RM3.slx';      % Specify Simulink Model File
simu.mode='normal';                     % Specify Simulation Mode
                                          % (normal/accelerator
                                          % /rapid-accelerator)
simu.explorer='on';                     % Turn SimMechanics Explorer
                                          % (on/off)

% Wave Information
waves.H = 2.5;                          % Wave Height [m]
waves.T = 8;                            % Wave Period [s]
waves.type = 'regular';                 % Specify Type of Waves

% Body Data
body(1) = bodyClass('Float');           % Initialize bodyClass for Float
body(1).hydroDataType = 'wamit';        % Specify BEM solver
body(1).hydroDataLocation = ...         % Location of WAMIT *.out file
    './wamit/rm3.out';
body(1).mass = 'wamitDisplacement';     % Mass from WAMIT [kg]
body(1).cg = 'wamit';                   % Cg from WAMIT [m]
body(1).momOfInertia = ...              % Moment of Inertia [kg-m^2]
    [20907301 21306090.66 37085481.11];
body(1).geometry = 'geometry/float.stl'; % Geometry File

body(2) = bodyClass('Spar_Plate');      % Initialize bodyClass for
                                          % Spar/Plate
body(2).hydroDataType = 'wamit';        % Specify BEM solver
body(2).hydroDataLocation = ...         % Location of WAMIT *.out file
    './wamit/rm3.out';
body(2).mass = 'wamitDisplacement';     % Mass from WAMIT [kg]
body(2).cg = 'wamit';                   % Cg from WAMIT [m]
body(2).momOfInertia = ...
    [94419614.57 94407091.24 28542224.82]; % Moment of Inertia [kg-m^2]
body(2).geometry = 'geometry/plate.stl'; % Geometry File

% PTO and Constraint Parameters
constraint(1) = ...                     % Initialize Constraint Class
    constraintClass('Constraint1');       % for Constraint1

pto(1) = ptoClass('PTO1');              % Initialize ptoClass for PTO1
pto(1).k=0;                             % PTO Stiffness Coeff [N/m]
pto(1).c=1200000;                       % PTO Damping Coeff [Ns/m]
```

WEC-Sim GUI (SimMechanics Explorer)



WEC-Sim Demo

Sandia National Laboratories

Pre-Processing
- Build the device CAD model
- Run WAMIT

↓

Build WEC Simulink/SimMechanics Model
- Use WEC component library to build multi-body model of the device

↓

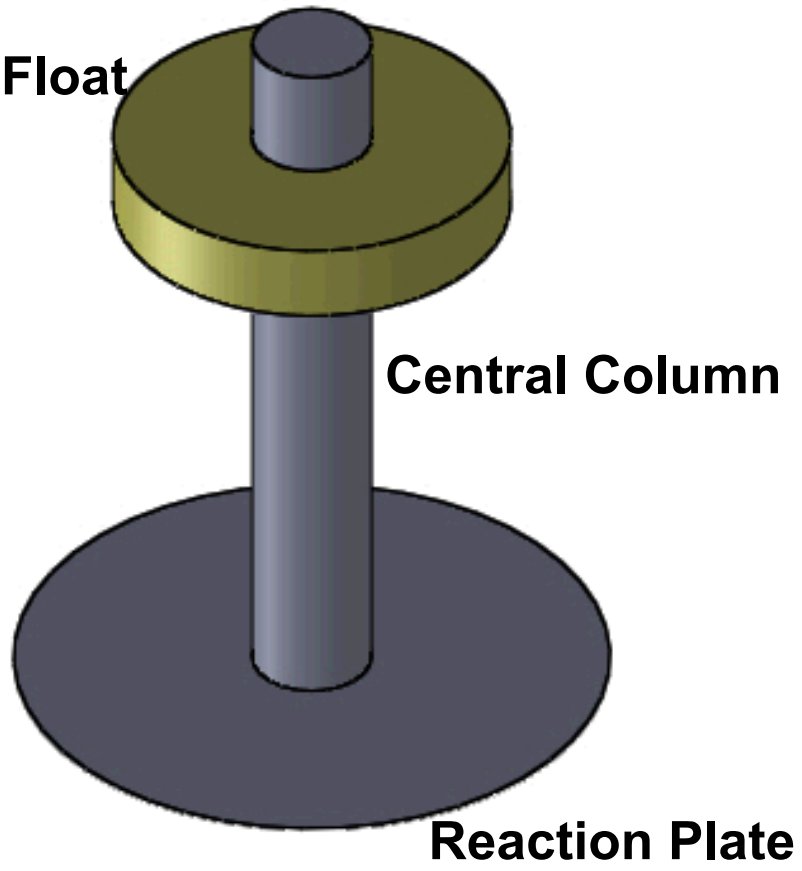
User input setup
- Wave conditions
- Simulation settings
- Define inertia properties and parameters

↓

Execute WEC-Sim Simulations
- From MATLAB input script

(NOT from WEC Simulink/SimMechanics Model)

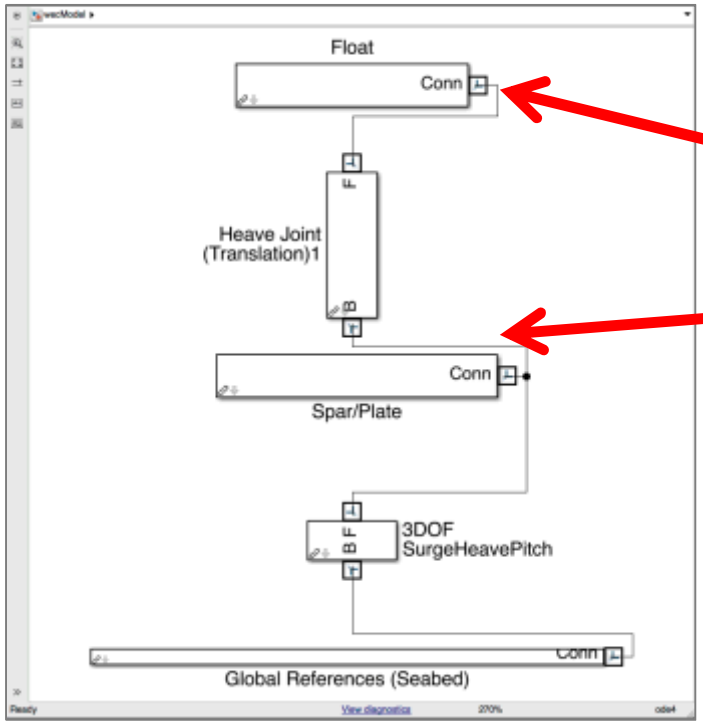
WEC-Sim was demonstrated by modeling a two-body floating point absorber WEC

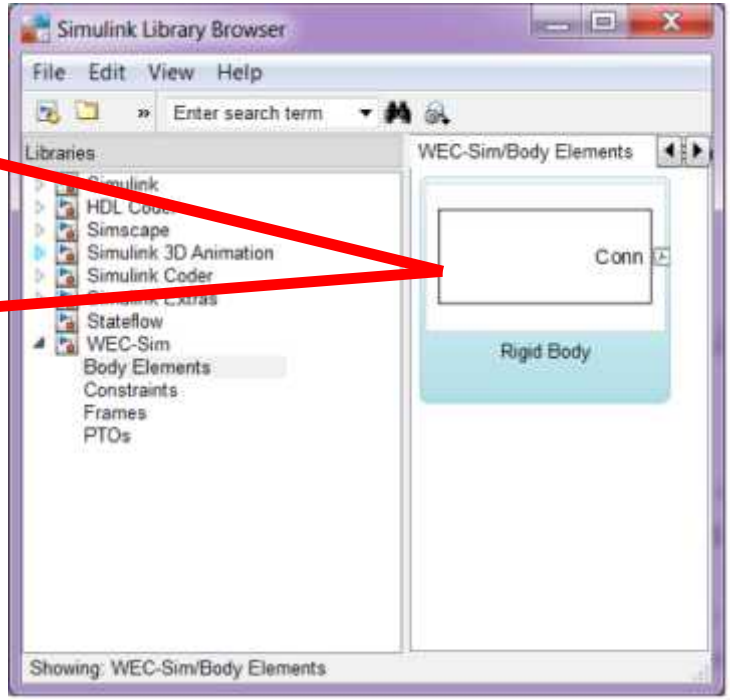


WEC-Sim Demo

- WEC-Sim has a built in “library” that includes common WEC components
- By selecting several components from the library, users can build the WEC device

SimuLink/SimMechanics Model



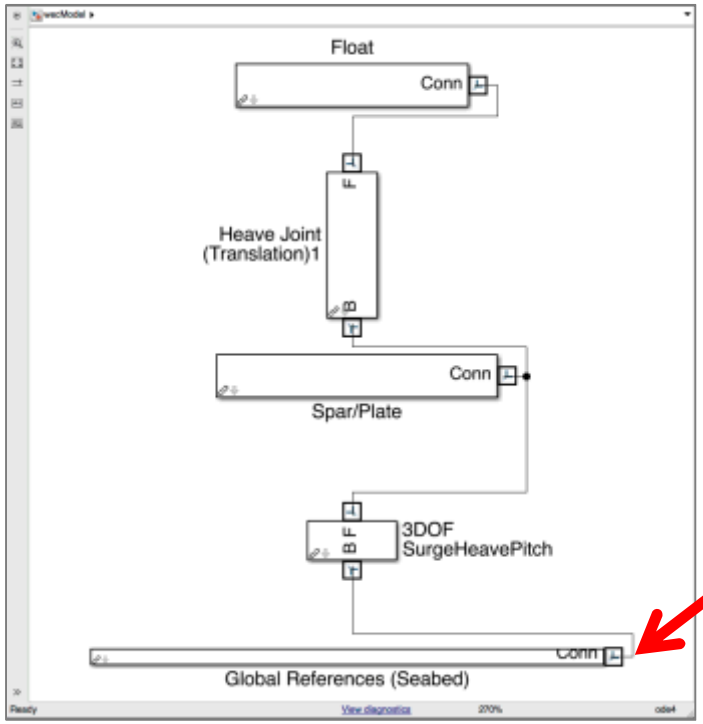


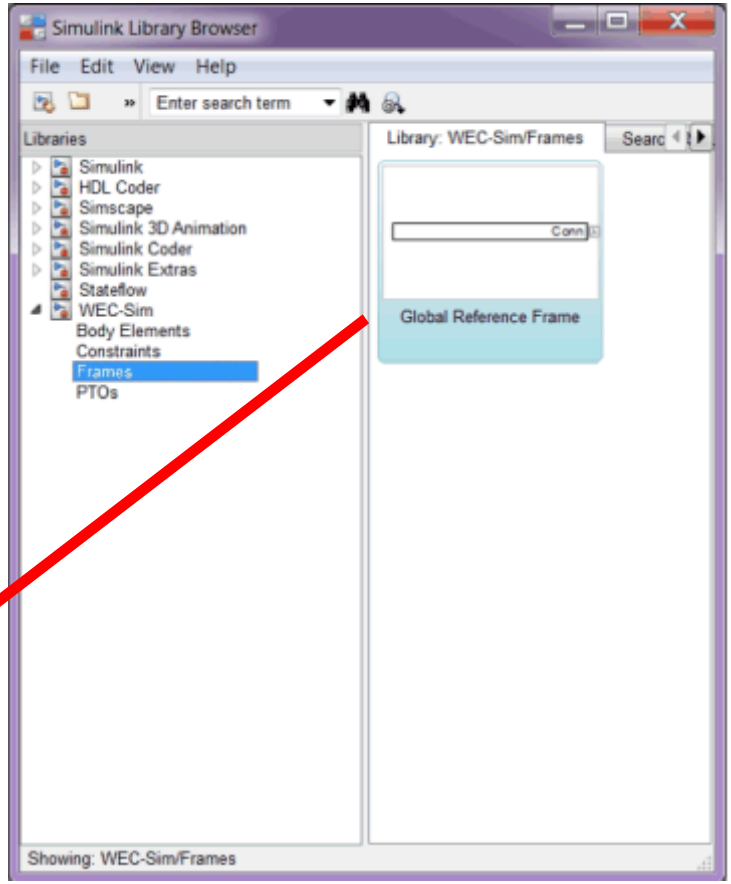
Simulating a two-body floating point absorber with slack mooring

WEC-Sim Demo

- WEC-Sim has a built in “library” that includes common WEC components
- By selecting several components from the library, users can build the WEC device

SimuLink/SimMechanics Model



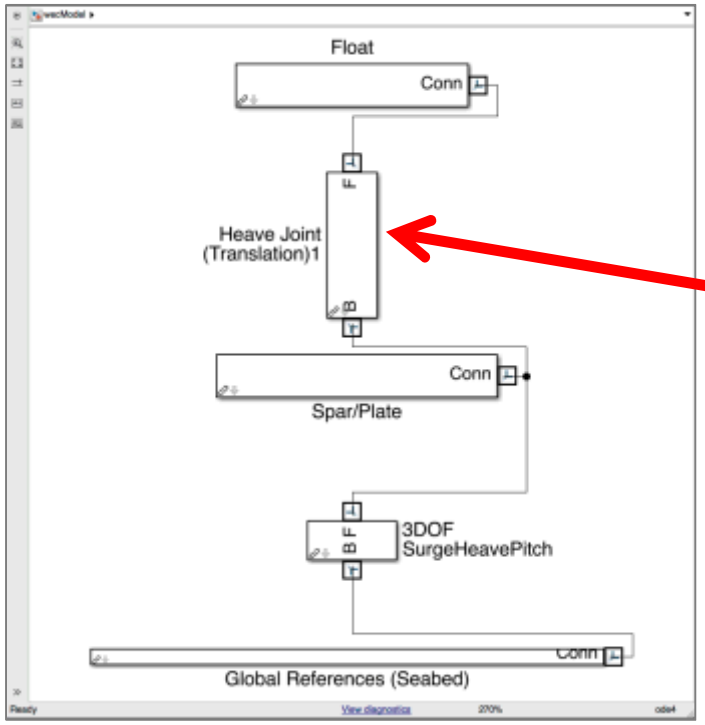


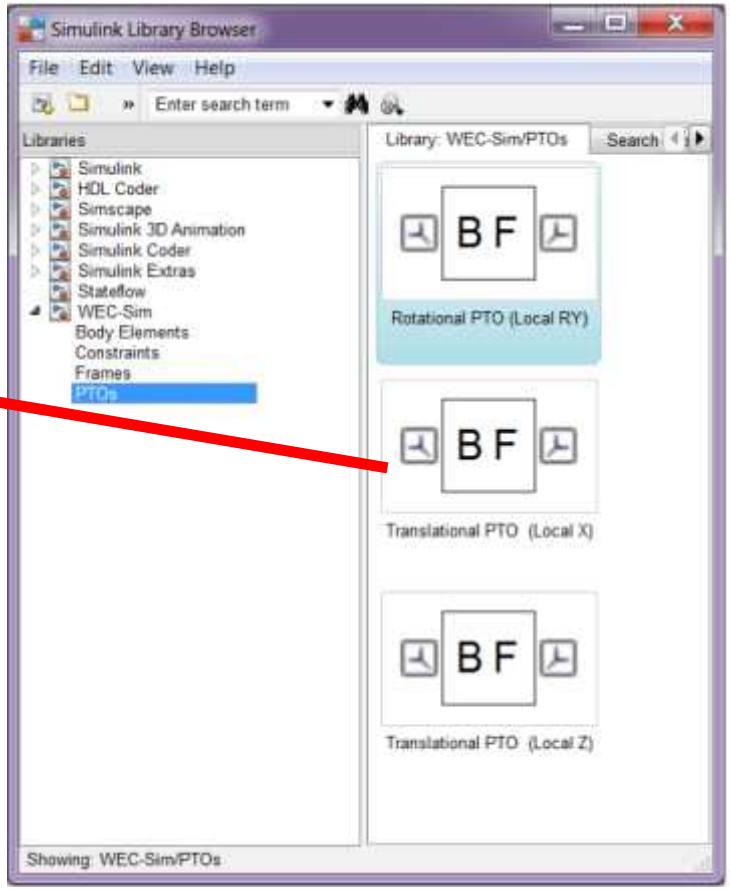
Simulating a two-body floating point absorber with slack mooring

WEC-Sim Demo

- WEC-Sim has a built in “library” that includes common WEC components
- By selecting several components from the library, users can build the WEC device

SimuLink/SimMechanics Model



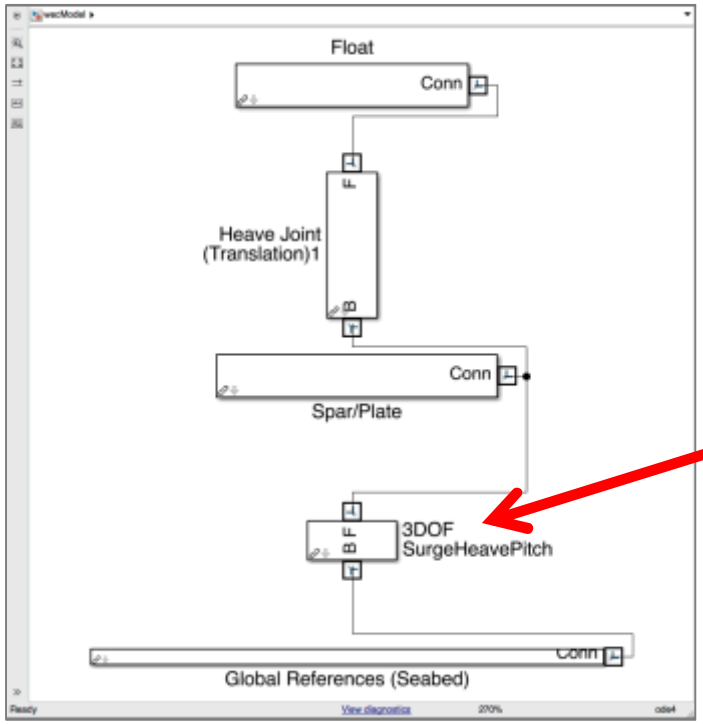


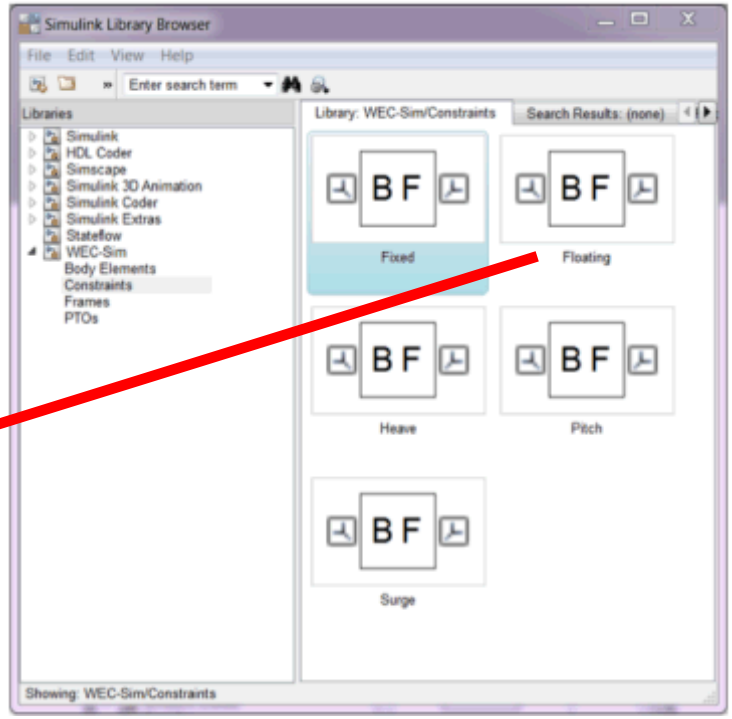
Simulating a two-body floating point absorber with slack mooring

WEC-Sim Demo

- WEC-Sim has a built in “library” that includes common WEC components
- By selecting several components from the library, users can build the WEC device

SimuLink/SimMechanics Model



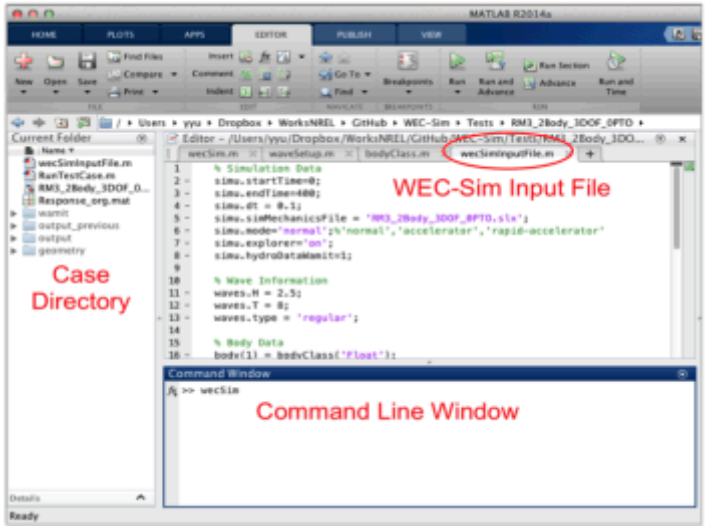


Simulating a two-body floating point absorber with slack mooring

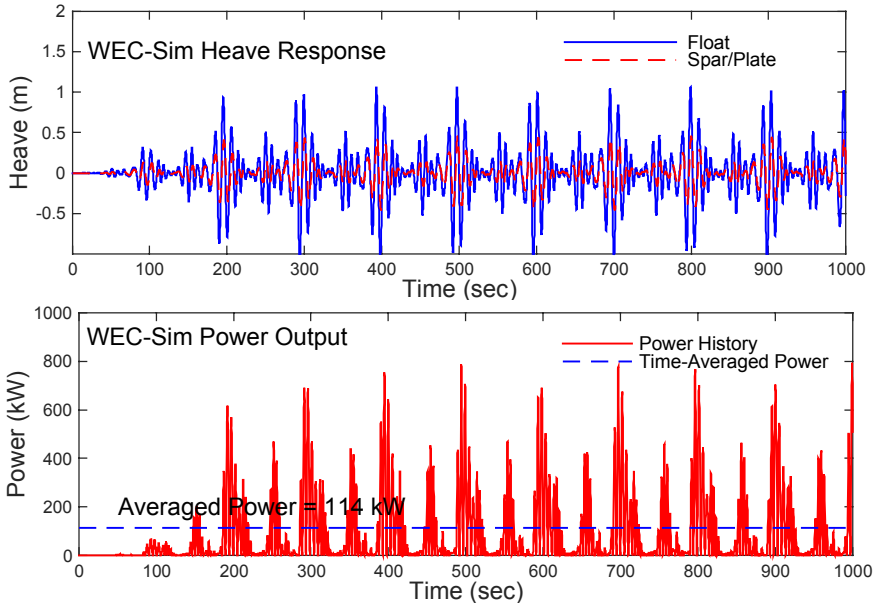
WEC-Sim Demo & output

Sandia National Laboratories

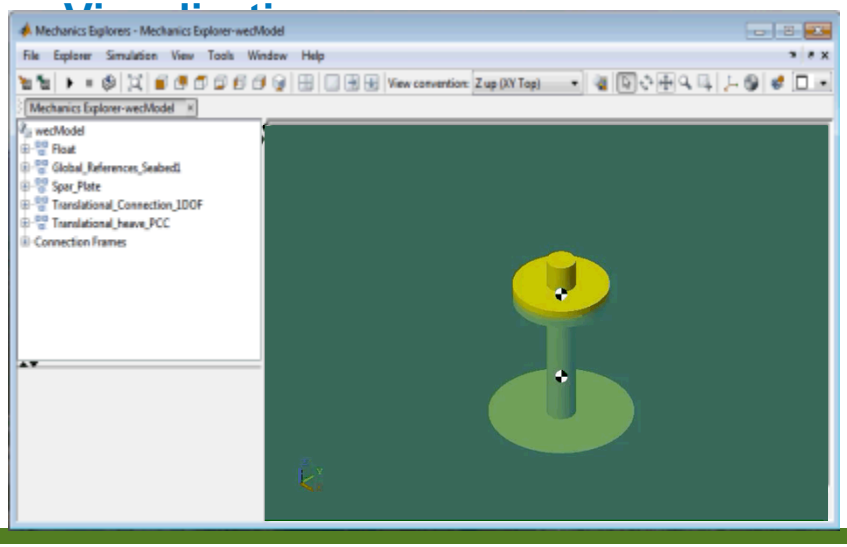
Run WEC-Sim



Device Motion & Instantaneous Power



WEC-Sim GUI and Simulation



Power Matrix (Irregular Waves)

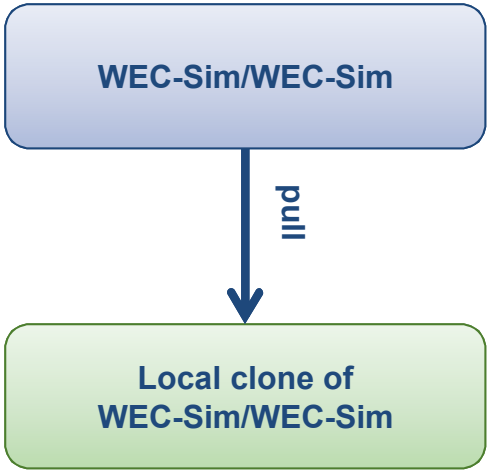
Significant Wave Height, Hs [m] \ Energy Period, Te [sec]	4.5	5.5	6.5	7.5	8.5	9.5	10.5	11.5	12.5	13.5	14.5	15.5	16.5	17.5
0.75	5	8	10	11	13	14	15	14	13	12	10	10	8	7
1.25	14	22	27	31	37	37	38	37	33	31	26	24	20	12
1.75	28	42	52	61	70	71	71	68	61	57	48	44	36	21
2.25	45	69	84	99	114	113	113	107	96	89	75	70	57	32
2.75	66	102	124	146	167	164	162	152	137	127	108	100	82	46
3.25	91	140	171	202	230	224	218	204	185	172	145	135	110	64
3.75	119	183	226	266	301	292	282	264	239	222	188	175	142	85
4.25	151	232	287	337	380	367	353	331	299	278	235	219	179	108
4.75	186	286	355	416	468	451	432	404	365	339	287	267	218	135
5.25	224	344	429	503	563	541	518	483	437	405	343	320	262	165
5.75	265	408	510	597	666	639	610	569	515	477	404	377	309	199
Peak Period, Tp [sec]	5.2	6.4	7.5	8.7	9.9	11.0	12.2	13.3	14.5	15.7	16.8	18.0	19.1	20.3

Sandia National Laboratories

COLLABORATION

Collaboration

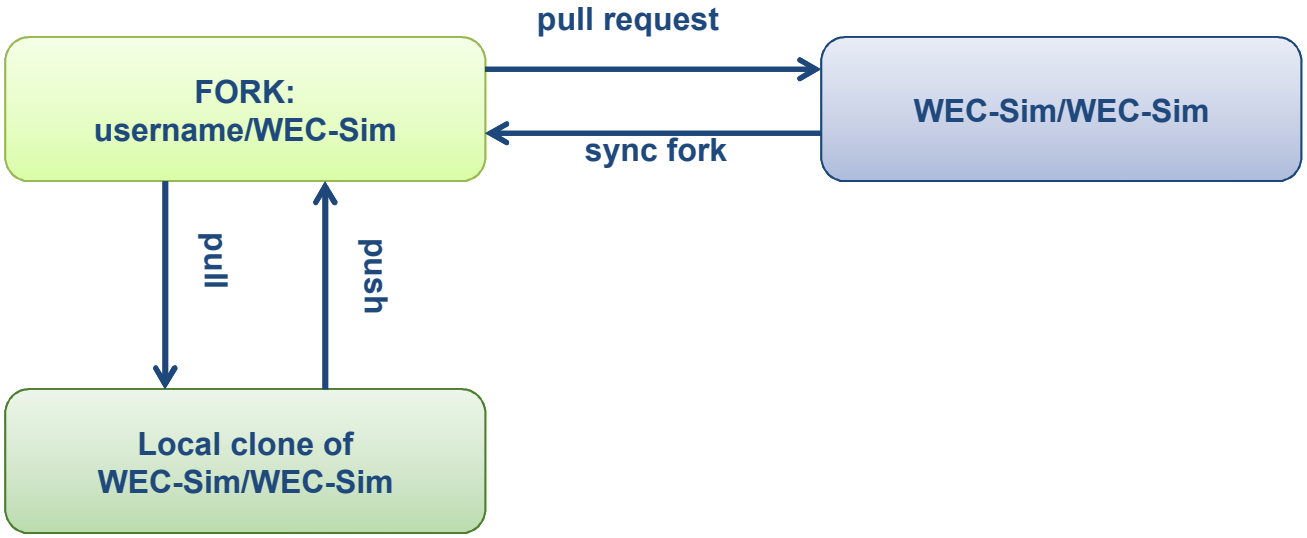
- GitHub
  - Online repository and version control
  - Option 1: non-collab
    - Clone
    - Pull



Collaboration

- Option 2: collab
  - Fork
  - Clone
  - Pull Request
  - Sync Fork

https://help.github.com/articles/using-pull-requests/
https://help.github.com/articles/syncing-a-fork/
https://github.com/WEC-Sim/WEC-Sim
https://github.com/WEC-Sim/WEC-Sim/graphs/contributors



Thank you!

kelley.ruehl@sandia.gov
carlos.michelen@sandia.gov

Acknowledgments
U.S. Department of Energy, Wind and Water Power Technologies Office

Sandia National Laboratories

EXTRA CONTENT

- Bonus
  - PTO-Sim (Nak)
  - MoorDyn (Matt)
  - Non-linear (Carlos)
  - Visualization (Carlos)

KELLEY – TUTORIAL SLIDES

Start from Scratch Example

Sandia National Laboratories

File Structure Overview

Information	File name	Location
Input file	wecSimInputFile.m	$CASE
WEC Model	<WEC Model Name>.slx	$CASE
Hydrodynamic Data	<hydro-data name>.h5	$CASE /bemio
Geometry	<STL File Name>.stl	$CASE /geometry

48