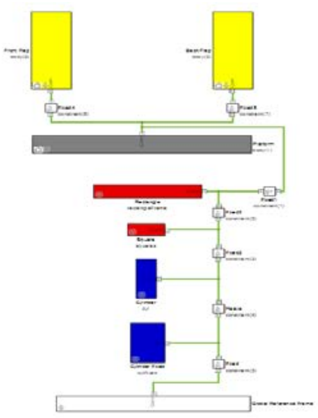
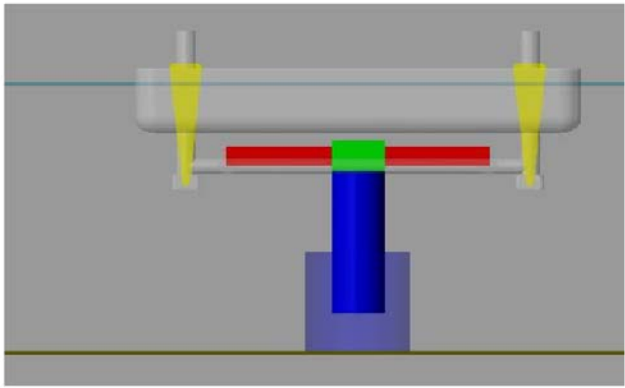
U.S. DEPARTMENT OF ENERGY | Energy Efficiency & Renewable Energy

FOSWEC wave tank testing and WEC-Sim simulation





WEC-Sim Training Course

for users and developers

August 17, 2017

Yi-Hsiang Yu (NREL)

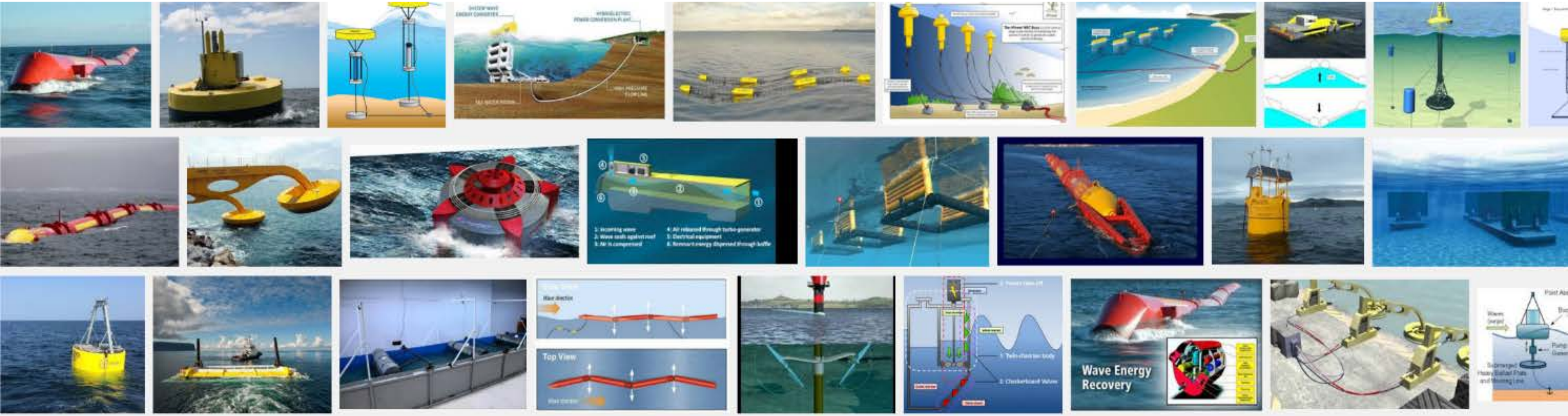
Kelley Ruehl (Sandia)

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 (*wecSim.m*)
11:00 am	**Code Structure Overview** ~1hr total	Overview of WEC-Sim's input file (*wecSimInputFile.m*), classes (**.m*) and library blocks (**.slx*)
1:00 pm	**Wave Implementation** ~30min	Description wave modeling implementation in WEC-Sim, in the classes (**.m*) and blocks (**.slx*)
1:30 pm	**Body Implementation** ~30min	Description body implementation in WEC-Sim, in the classes (**.m*) and blocks (**.slx*)
2:00pm	**Q&A** ~1hr	Open Q&A for attendees to WEC-Sim Lab team

WEC-Sim Webinar

U.S. DEPARTMENT OF ENERGY | Energy Efficiency & Renewable Energy

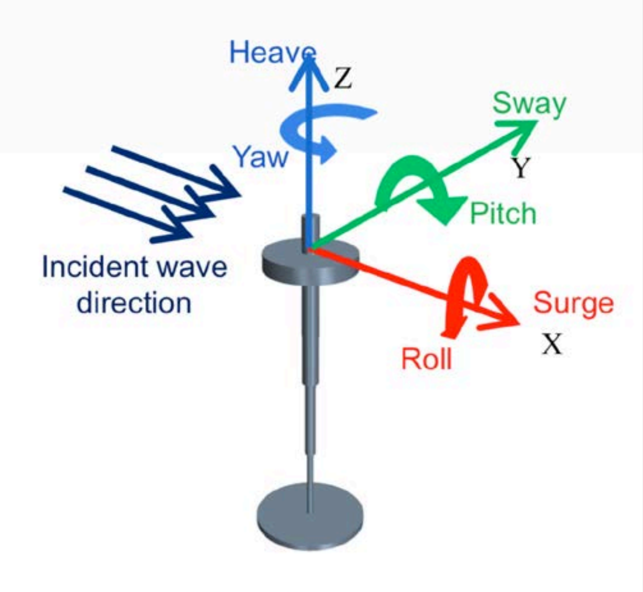


Theory & Workflow

Yi-Hsiang (NREL)

Coordinate System

- X-axis is in the direction of wave propagation if the wave heading angle is equal to zero (following the coordinate system definition in WAMIT).
- The Z-axis is in the vertical upwards direction, and the Y-axis direction is defined by the right-hand rule.
- Surge (x), Sway (y), and Heave (z) correspond to the first, second and third position respectively. Roll (Rx), Pitch (Ry), and Yaw (Rz) correspond to the fourth, fifth, and sixth position respectively.



WEC-Sim Coordinate System

Equation of Motion

U.S. DEPARTMENT OF ENERGY | Energy Efficiency & Renewable Energy

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

Nonlinear hydrodynamic force

$$m\ddot{x}(t) = \boxed{F_{hs}(t)} + \boxed{F_{ext}(t)} + \boxed{F_{rad}(t)} + \boxed{F_{v}(t)} + \boxed{F_{pto}(t)} + \boxed{F_{m}(t)} + \boxed{F_{nh}(t)}$$

Mooring force

Hydrostatic restoring force

Power take-off force

Viscous force

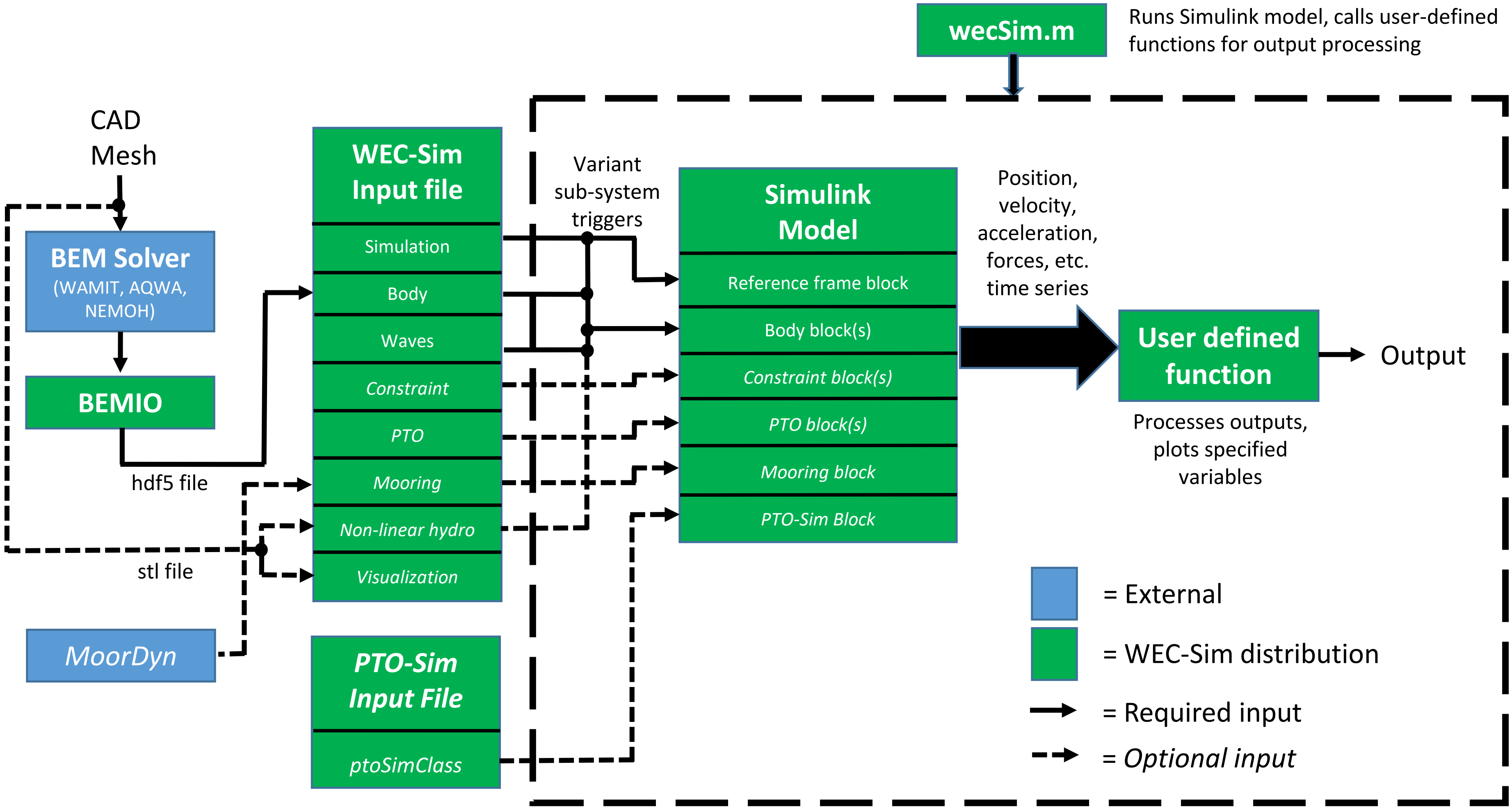
Wave excitation & diffraction force (from BEM simulations)

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

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

Theory

Forcing Term	Condition	Theory
Radiation (F_{rad})	Regular Waves Irregular Waves	Sinusoidal Steady-State Response $F_{rad} = -A(\omega)\ddot{X} - B(\omega)\dot{X}$ Cummins Equation (Convolution Integral) $F_{rad} = -A_{\infty}\ddot{X} - \int_0^t K(t-\tau)\dot{X}(\tau)d\tau$ State Space Representation $\dot{\mathbf{X}}_{\mathbf{r}}(t) = \mathbf{A}_{\mathbf{r}}\mathbf{X}_{\mathbf{r}}(t) + \mathbf{B}_{\mathbf{r}}\dot{\zeta}(t);\ \mathbf{X}_{\mathbf{r}}(0) = 0$; $\int_{-\infty}^{t} \mathbf{K}_{\mathbf{r}}(t-\tau)\,d\tau \approx \mathbf{C}_{\mathbf{r}}\mathbf{X}_{\mathbf{r}}(t) + \mathbf{D}_{\mathbf{r}}\dot{\zeta}(t)$
Wave Excitation (F_{ext})	Regular Waves Irregular Waves	Sinusoidal Steady-State Response $F_{ext} = \Re\left[R_f \frac{H}{2} F_X(\omega) e^{i(\omega t)}\right]$ Wave Spectrum (e.g., JS; BS; PM) $F_{ext} = \Re\left[R_f \int_0^{\infty} F_X(\omega_r) e^{i(\omega_r t+\phi)} \sqrt{2S(\omega_r)d\omega_r}\right]$ Wave Elevation (Convolution Integral) $F_{ext} = \int_{-\infty}^{\infty} \eta(\tau) f_e(t-\tau)d\tau$
PTO (F_{pto})		Linear Spring-Damper $P_{PTO} = C_{PTO}\dot{X}_{rel}^2$ Hydraulic PTO Mechanical PTO $P_{PTO} = -F_{PTO}\dot{X}_{rel}$
Mooring (F_m)		Linear Mooring Matrix (i.e., stiffness, damping and pretension) Lumped-Mass Mooring Dynamics Model (MoorDyn)
Additional Added-Mass & Damping (F_v & F_{ME})		Linear & Quadratic Damping Forces $F_v = -C_{ld}\dot{X} - \frac{1}{2}C_d \rho A_D \dot{X}\lvert\dot{X}\rvert$ Morison Elements $F_{ME} = \rho\forall\dot{v} + \rho\forall C_a(\dot{v} - \ddot{X}) + \frac{1}{2}C_d \rho A_D (v - \dot{X})\lvert v - \dot{X}\rvert$
Nonlinear Hydrodynamic Forces (F_{nh})	Nonlinear Hydrodynamics	The additional term accounts for the difference between the nonlinear and linear hydrodynamic forces (buoyancy and the Froude-Krylov force components).


wecSim.m
Runs Simulink model, calls user-defined functions for output processing
CAD
Mesh
BEM Solver
(WAMIT, AQWA, NEMOH)
BEMIO
hdf5 file
stl file
MoorDyn
WEC-Sim Input file
Simulation
Body
Waves
Constraint
PTO
Mooring
Non-linear hydro
Visualization
PTO-Sim Input File
ptoSimClass
Variant sub-system triggers
Simulink Model
Reference frame block
Body block(s)
Constraint block(s)
PTO block(s)
Mooring block
PTO-Sim Block
Position, velocity, acceleration, forces, etc. time series
User defined function
Output
Processes outputs, plots specified variables
= External
= WEC-Sim distribution
= Required input
= Optional input

Run wecSim.m

U.S. DEPARTMENT OF ENERGY | Energy Efficiency & Renewable Energy

/Users/yyu/Dropbox/WorksNREL/GitHub/WEC-Sim/source/wecSim.m

EDITOR PUBLISH VIEW

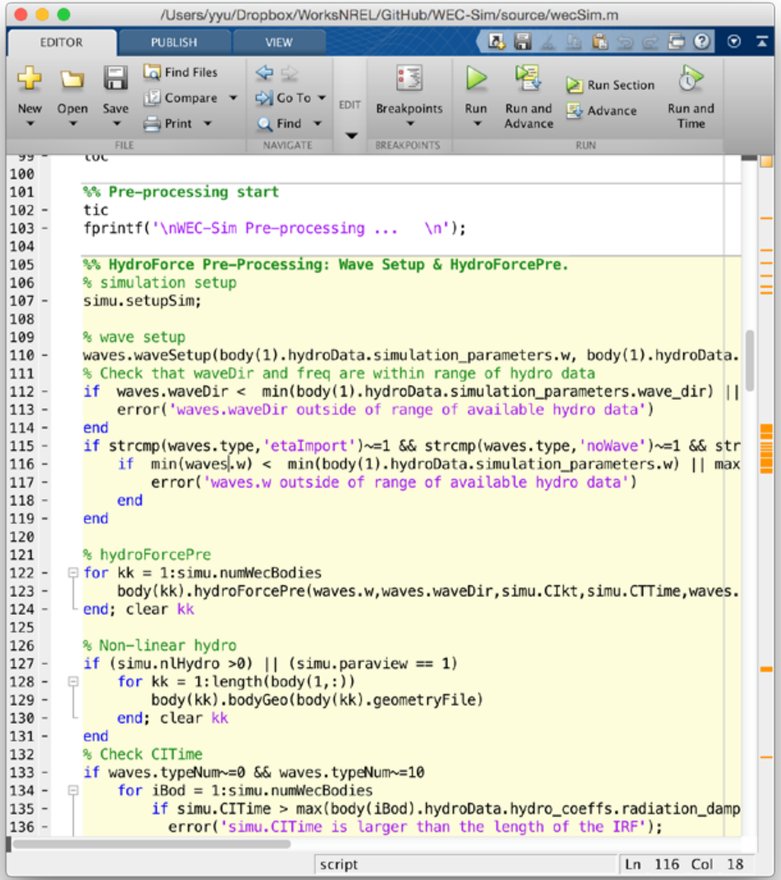
```
%%

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


%% Read input file
tic
try fprintf('wecSimMCR Case %g\n',imcr); end
fprintf('\nWEC-Sim Read Input File ...   \n');
evalc('wecSimInputFile');
% Read Inputs for Multiple Conditions Run
if exist('mcr','var') == 1;
    for n=1:length(mcr.cases(1,:))
        if iscell(mcr.cases)
            eval([mcr.header{n} '= mcr.cases{imcr,n};']);
        else
            eval([mcr.header{n} '= mcr.cases(imcr,n);']);
        end
    end; clear n combine;
end
% Waves and Simu: check inputs
waves.checkinputs;
simu.checkinputs;
% Constraints: count & set orientation
if exist('constraint','var') == 1
    simu.numConstraints = length(constraint(1,:));
    for ii = 1:simu.numConstraints
        constraint(ii).constraintNum = ii;
        constraint(ii).setOrientation();
    end; clear ii
end
% PTOs: count & set orientation
if exist('pto','var') == 1
    simu.numPtos = length(pto(1,:));
    for ii = 1:simu.numPtos
        pto(ii).ptoNum = ii;
        pto(ii).setOrientation();
    end; clear ii
end
```

Ln 1 Col 1

Read input file

- Read WEC-Sim input file using MATLAB **evalc** to evaluate MATLAB expression with capture
- Check the input parameters and hydrodynamic coefficients for each body

Run wecSim.m



Pre-processing

- Setup Simulation parameters

  simu.setupSim

- Setup the wave environment and generate wave elevation time history

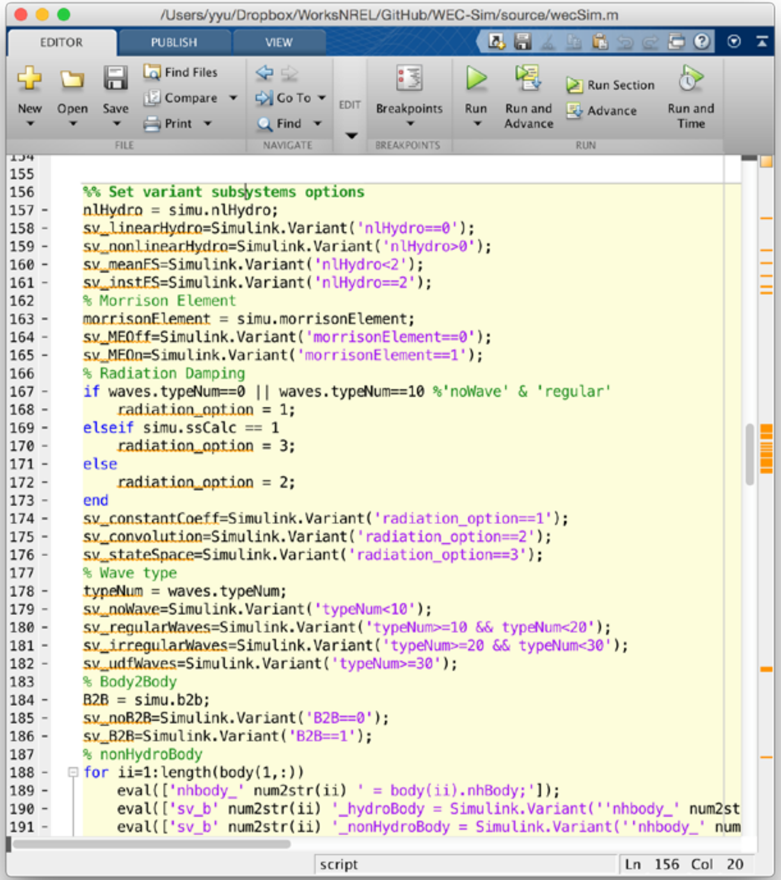
  waves.waveSetup

- Convert non- dimensional hydrodynamic coefficients to dimensional hydrodynamic forces

  Body(kk).hydroForcePre

Run wecSim.m

U.S. DEPARTMENT OF ENERGY | Energy Efficiency & Renewable Energy



Variant Subsystems

- WEC-Sim allows user to perfrom the simulation using different options (e.g., regular/irregular waves, linear/nonlinear model, state-space representation, etc.)

- The options are defined in wecSimInputFile.m and renamed here (Simulink does not like function has a name with ‘.’)

Run wecSim.m

U.S. DEPARTMENT OF ENERGY | Energy Efficiency & Renewable Energy

/Users/yyu/Dropbox/WorksNREL/GitHub/WEC-Sim/source/wecSim.m

EDITOR PUBLISH VIEW

```
%% Load simMechanics file & Run Simulation
tic
fprintf('\nSimulating the WEC device defined in the SimMechanics model %s...
% Modify some stuff for simulation
for iBod = 1:simu.numWecBodies
    body(iBod).adjustMassMatrix(simu.adjMassWeightFun,simu.b2b);
end; clear iBod
warning('off','Simulink:blocks:TDelayTimeTooSmall');
warning('off','Simulink:blocks:BusSelDupBusCreatorSigNames');
warning('off','MATLAB:loadlibrary:FunctionNotFound');
warning('off','MATLAB:loadlibrary:parsewarnings');
warning('off','Simulink:blocks:DivideByZero');
set_param(0, 'ErrorIfLoadNewModel', 'off')
% run simulation
simu.loadSimMechModel(simu.simMechanicsFile);
sim(simu.simMechanicsFile);
% Restore modified stuff
clear nlHydro sv_linearHydro sv_nonlinearHydro ssCalc radiation_option sv_con
clear nhbod* sv_b* sv_noWave sv_regularWaves sv_irregularWaves sv_udfWaves sv
toc

tic
%% Post processing and Saving Results
postProcess
% User Defined Post-Processing
if exist('userDefinedFunctions.m','file') == 2
    userDefinedFunctions;
end
% ASCII files
if simu.outputtxt==1
    output.writetxt();
end
paraViewVisualization

%% Save files
clear ans table tout;
toc
diary off
movefile('simulation.log',simu.logFile)
if simu.saveMat==1
    save(simu.caseFile)
end
```

script | Ln 245 Col 1

Load Simulink/Simscape Model

- Modify the mass matrix to account for the added-mass term to avoid algebraic loop and improve numerical stability

  body(iBod).adjustMassMatrix(simu.adjMassWeightFun,simu.b2b)

- Load the **slx** file and perform the simulation

- Post processing and saving results

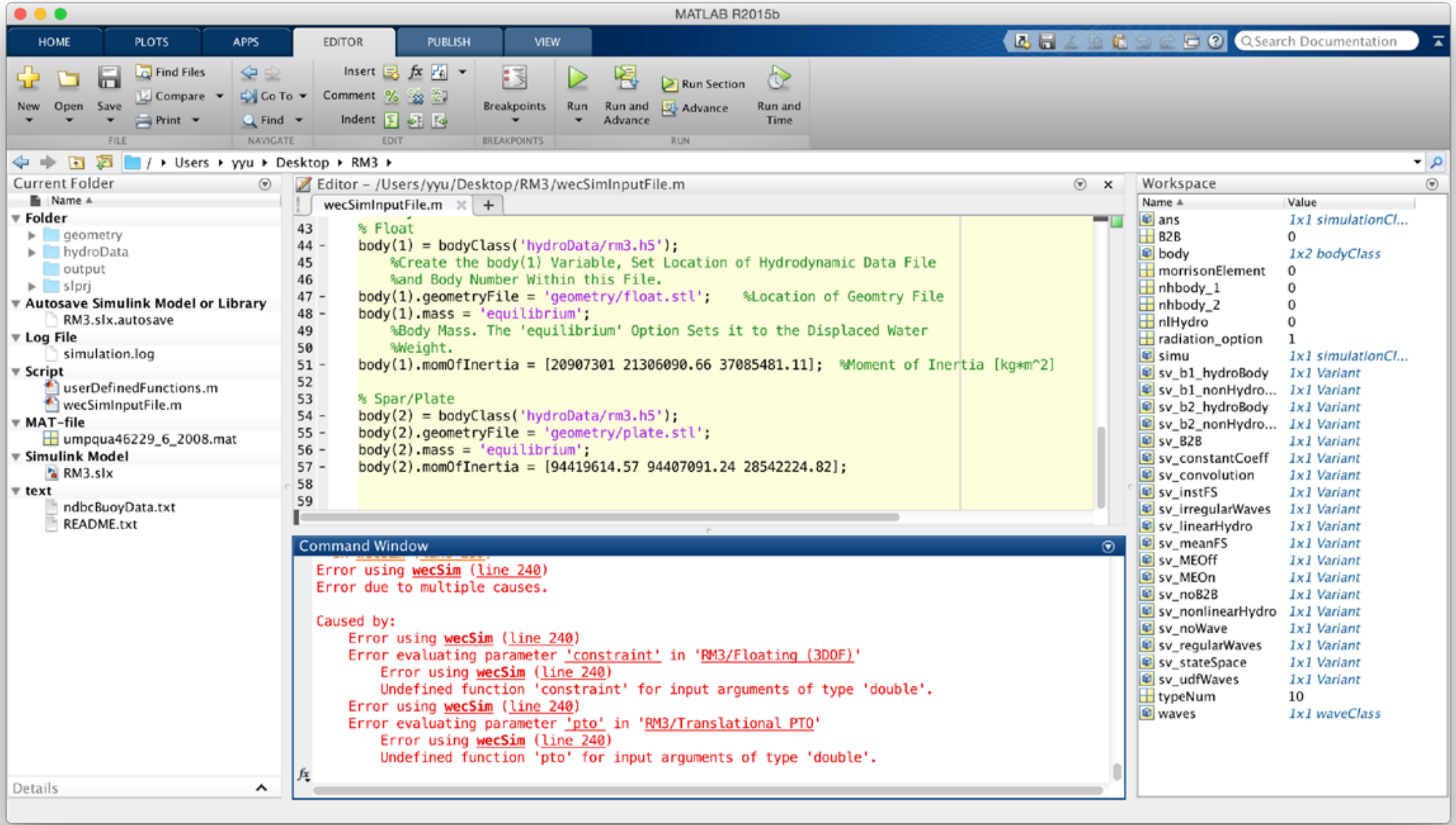
Notes/Warnings/Errors

- Different classes can be specified in any order in the WEC-Sim input file (wecSimInputFile.m).

- Multiple Condition Runs function (wecSimMCR.m) is just a wrapper that execute wecSim.m multiple times.

- Warning massage for the Simulink/Simscape Model if MATLAB 2015b is used.

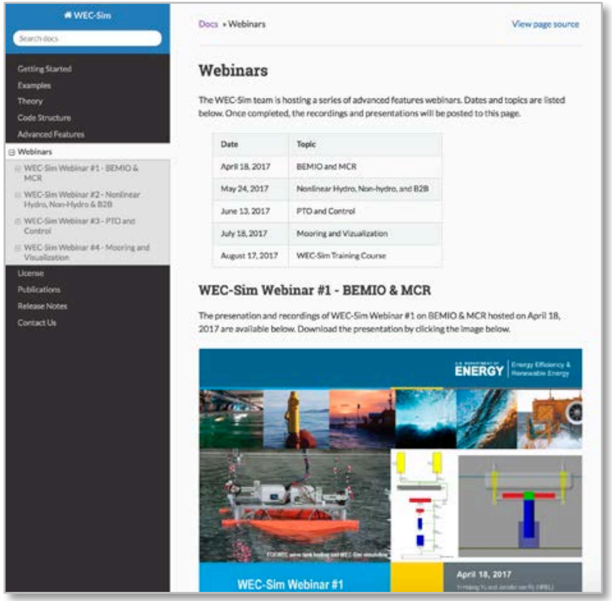
Notes/Warnings/Errors



Thank you!

U.S. DEPARTMENT OF ENERGY | Energy Efficiency & Renewable Energy

All the webinar materials and recordings are available online:
http://wec-sim.github.io/WEC-Sim/webinars.html



Sandia National Laboratories is a multi-mission laboratory managed and operated by National Technology and 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.