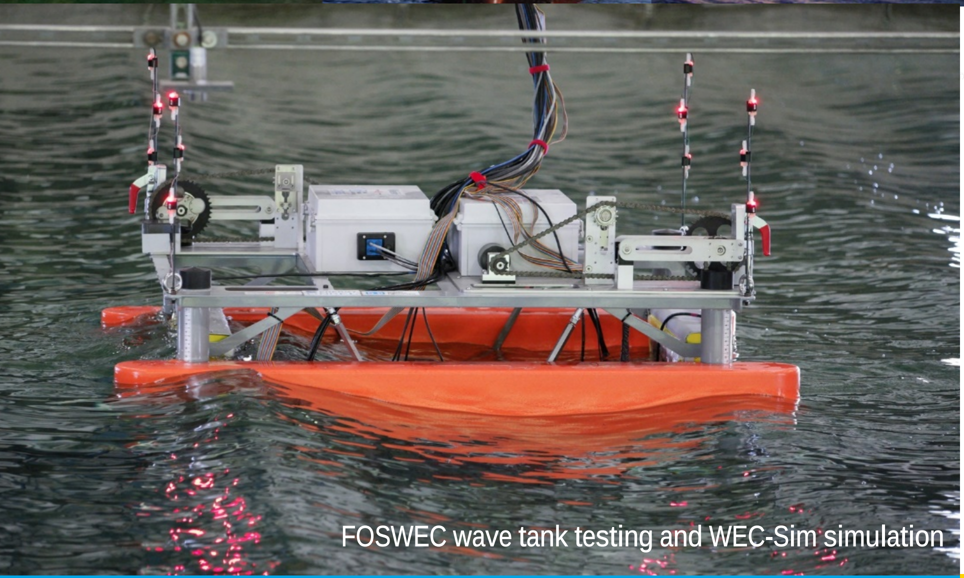
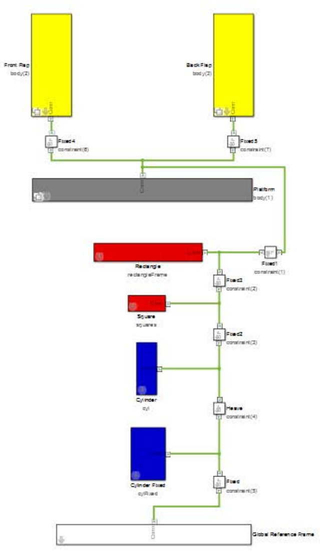
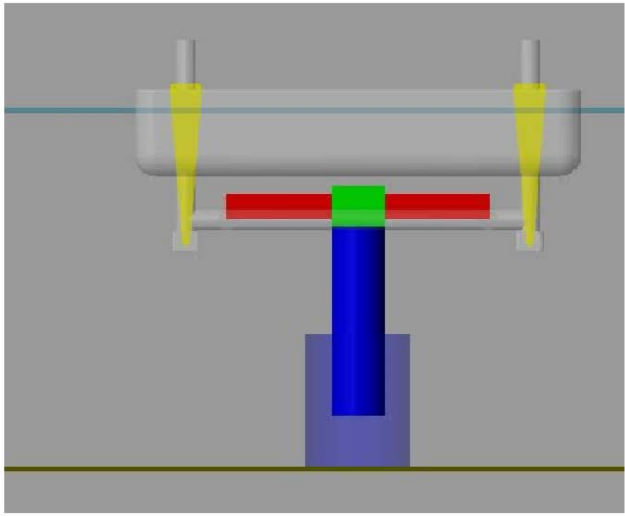
U.S. DEPARTMENT OF ENERGY | Energy Efficiency & Renewable Energy







FOSWEC wave tank testing and WEC-Sim simulation

WEC-Sim Webinar #1

BEMIO and MCR

April 18, 2017

Yi-Hsiang Yu and Jennifer van Rij (NREL)
Kelley Ruehl (Sandia)

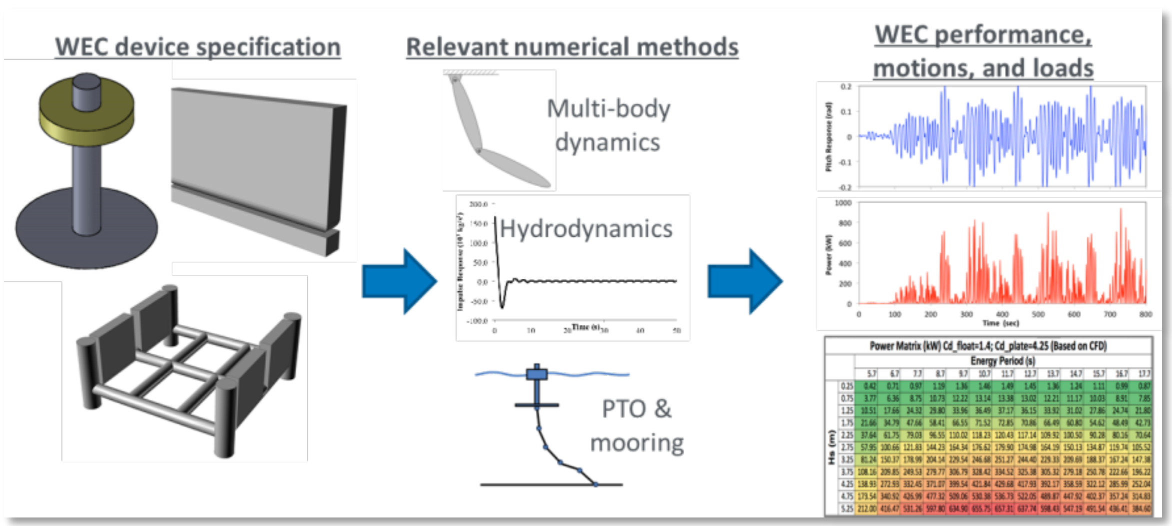
WEC-Sim Team

- Kelley Ruehl (Sandia)
- Yi-Hsiang Yu (NREL)
- Jennifer van Rij (NREL)

U.S. DEPARTMENT OF ENERGY

Sandia National Laboratories

NREL NATIONAL RENEWABLE ENERGY LABORATORY



Schedule

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

Advanced Feature Webinars *1hr each*

- **April 18:** bemio and mcr, application for power matrix
- **May 24:** nl-hydro, b2b, non-hydro and drag
- **June 7:** pto and control, application for desalination
- **July 18:** mooring and visualization

Training Courses

- **May 1:** *1hr* WEC-Sim workshop at METS, for new users
- **TBD:** *half-day* WEC-Sim code structure course, for advanced users/developers

BEMIO

BEM Input/Output

Jennifer van Rij (NREL)

BEMIO - Overview

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

- What is BEMIO?
  - Workflow
  - Purpose
  - History
  - Locations
- BEMIO Functions
  - Read_WAMIT
  - Read_NEMOH
  - Read_AQWA
  - Normalize
  - Combine_BEM
  - Radiation_IRF
  - Radiation_IRF_SS
  - Excitation_IRF
  - Write_H5
  - Plot_BEMIO
- Examples and Usage
  - WAMIT example
  - NEMOH example w/ WAMIT comparison example
  - Data structures
- Possible Improvements
  - Documentation
  - Read_AQWA
  - Meshing & visualization
  - Post-processing functions
  - Integration with Nemoh

What is BEMIO

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

- 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, 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.
- History
  - A few python BEMIO legacies… meshing utilities, .h5 file, MATLAB data structure
- Locations
  - Functions: …\WEC-Sim\source\functions\BEMIO
  - Documentation: http://wec-sim.github.io/WEC-Sim/features.html#bemio

Read_WAMIT

Reads data from a WAMIT output file

bemio.m

```
clc; clear all; close all;
hydro = struct();

hydro = Read_WAMIT(hydro,'rm3.out',[]);
hydro = Radiation_IRF(hydro,60,[],[],[],[]);
hydro = Radiation_IRF_SS(hydro,[],[]);
hydro = Excitation_IRF(hydro,157,[],[],[],[]);
Write_H5(hydro)
Plot_BEMIO(hydro)
```

hydro = Read_WAMIT(hydro, filename, ex_coeff)

hydro - data structure

filename - WAMIT output file

ex_coeff - flag indicating the type of excitation force coefficients to read, ‘diffraction’ (default, []), ‘haskind’, or ‘rao’

Notes:

- If generalized body modes (currently only a research application) are used, the output directory must also include the *.cfg, *.mmx, and *.hst files.
- If simu.nlHydro = 3 (not implemented yet) will be used, the output directory must also include the .3fk and .3sc files.

Read_NEMOH

Reads data from a NEMOH working folder

bemio.m

```
clc; clear all; close all;
hydro = struct();

hydro = Read_NEMOH(hydro,'..\RM3\');
hydro = Radiation_IRF(hydro,60,[],[],[],1.9);
hydro = Radiation_IRF_SS(hydro,[],[]);
hydro = Excitation_IRF(hydro,157,[],[],[],1.9);
Write_H5(hydro)
Plot_BEMIO(hydro)
```

hydro = Read_NEMOH(hydro, filedir)

hydro - data structure

filedir - 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.nlHydro = 3 will be used
- Results/FKForce.tec - If simu.nlHydro = 3 will be used

Notes:

- NEMOH website recently updated; https://lheea.ec-nantes.fr/logiciels-et-brevets/nemoh-running-192930.kjsp?RH=1489591054559

Read_AQWA

Reads data from AQWA output files

bemio.m

```
clc; clear all; close all;
hydro = struct();

hydro = Read_AQWA(hydro,'aqwa_example_data.AH1',...
    'aqwa_example_data.LIS');
hydro = Radiation_IRF(hydro,15,[],[],[],[]);
hydro = Radiation_IRF_SS(hydro,[],[]);
hydro = Excitation_IRF(hydro,15,[],[],[],[]);
Write_H5(hydro)
Plot_BEMIO(hydro)
```

hydro = Read_AQWA(hydro, ah1_filename, lis_filename)

- ***hydro*** – data structure
- ***ah1_filename*** – .AH1 AQWA output file
- ***lis_filename*** – .LIS AQWA output file

Normalize

Normalizes NEMOH and AQWA hydrodynamic coefficients in the same manner that WAMIT outputs are normalized. And, if necessary, sorts data according to ascending frequency (WAMIT).

$C_{i,j}/\rho g$ - linear stiffness

$A_{i,j}/\rho g$ - added mass

$B_{i,j}/\rho\omega$ - radiation damping

$X_{i}/\rho g$ - exciting forces

hydro = Normalize(hydro)

hydro – data structure

Combine_BEM

Combines multiple BEM outputs into one hydrodynamic ‘system.’

bemio.m

```
clc; clear all; close all;
hydro = struct();

hydro = Read_NEMOH(hydro,'..\RM3\');
hydro = Read_WAMIT(hydro,'..\..\WAMIT\RM3\rm3.out',[]);
hydro = Combine_BEM(hydro); % Compare WAMIT
hydro = Radiation_IRF(hydro,60,[],[],[],1.9);
hydro = Radiation_IRF_SS(hydro,[],[]);
hydro = Excitation_IRF(hydro,157,[],[],[],1.9);
Write_H5(hydro)
Plot_BEMIO(hydro)
```

hydro = Combine_BEM(hydro)

hydro – data structure

Radiation_IRF

Calculates the normalized radiation impulse response function:

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

bemio.m

```
clc; clear all; close all;
hydro = struct();

hydro = Read_WAMIT(hydro,'rm3.out',[]);
hydro = Radiation_IRF(hydro,60,[],[],[],[]);
hydro = Radiation_IRF_SS(hydro,[],[]);
hydro = Excitation_IRF(hydro,157,[],[],[],[]);
Write_H5(hydro)
Plot_BEMIO(hydro)
```

hydro = Radiation_IRF(hydro, t_end, n_t, n_w, w_min, w_max)

- ***hydro*** – data structure
- ***t_end*** – calculation range for the IRF, where the IRF is calculated from t = 0 to t_end, and the default is 100 s
- ***n_t*** – number of time steps in the IRF, the default is 1001
- ***n_w*** – number of frequency steps used in the IRF calculation (hydrodynamic coefficients are interpolated to correspond), the default is 1001
- ***w_min*** – minimum frequency to use in the IRF calculation, the default is the minimum frequency from the BEM data
- ***w_max*** – maximum frequency to use in the IRF calculation, the default is the maximum frequency from the BEM data.

Radiation_IRF_SS

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

bemio.m

```
clc; clear all; close all;
hydro = struct();

hydro = Read_WAMIT(hydro,'rm3.out',[]);
hydro = Radiation_IRF(hydro,60,[],[],[],[]);
hydro = Radiation_IRF_SS(hydro,[],[]);
hydro = Excitation_IRF(hydro,157,[],[],[],[]);
Write_H5(hydro)
Plot_BEMIO(hydro)
```

hydro = Radiation_IRF_SS(hydro, Omax, R2t)

- ***hydro*** – data structure
- ***Omax*** – maximum order of the SS realization, the default is 10
- ***R2t*** – R2 threshold (coefficient of determination) for the SS realization, where ***R2*** may range from 0 to 1, and the default is 0.95

Excitation_IRF

Calculates the excitation impulse response function:

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

bemio.m

```
clc; clear all; close all;
hydro = struct();

hydro = Read_WAMIT(hydro,'rm3.out',[]);
hydro = Radiation_IRF(hydro,60,[],[],[],[]);
hydro = Radiation_IRF_SS(hydro,[],[]);
hydro = Excitation_IRF(hydro,157,[],[],[],[]);
Write_H5(hydro)
Plot_BEMIO(hydro)
```

hydro = Excitation_IRF(hydro, t_end, n_t, n_w, w_min, w_max)

- ***hydro*** - data structure
- ***t_end*** - calculation range for the IRF, where the IRF is calculated from t = -t_end to t_end, and the default is 100 s
- ***n_t*** - number of time steps in the IRF, the default is 1001
- ***n_w*** - number of frequency steps used in the IRF calculation (hydrodynamic coefficients are interpolated to correspond), the default is 1001
- ***w_min*** - minimum frequency to use in the IRF calculation, the default is the minimum frequency from the BEM data
- ***w_max*** - maximum frequency to use in the IRF calculation, the default is the maximum frequency from the BEM data.

Write_H5

Writes the hydro data structure to a .h5 file.

```
clc; clear all; close all;
hydro = struct();

hydro = Read_WAMIT(hydro,'rm3.out',[]);
hydro = Radiation_IRF(hydro,60,[],[],[],[]);
hydro = Radiation_IRF_SS(hydro,[],[]);
hydro = Excitation_IRF(hydro,157,[],[],[],[]);
Write_H5(hydro)
Plot_BEMIO(hydro)
```

Write_H5(hydro)

hydro – data structure

Plot_BEMIO

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.

bemio.m

```
clc; clear all; close all;
hydro = struct();

hydro = Read_WAMIT(hydro,'rm3.out',[]);
hydro = Radiation_IRF(hydro,60,[],[],[],[]);
hydro = Radiation_IRF_SS(hydro,[],[]);
hydro = Excitation_IRF(hydro,157,[],[],[],[]);
Write_H5(hydro)
Plot_BEMIO(hydro)
```

Plot_BEMIO(hydro)

> ***hydro*** – data structure

Examples and Usage

- BEMIO tutorials ….\WEC-Sim\tutorials\BEMIO
    - WAMIT
        - Cylinder example
    - NEMOH
        - Cylinder w/WAMIT comparison example
    - AQWA
- Data structures
    - BEMIO
        - http://wec-sim.github.io/WEC-Sim/features.html
    - .h5
        - HDFVIEW: https://support.hdfgroup.org/products/java/hdfview/

Possible Future Improvements

- Documentation
- Read_AQWA
- Meshing/visualization functions
- Post-processing functions
- Integration with Nemoh

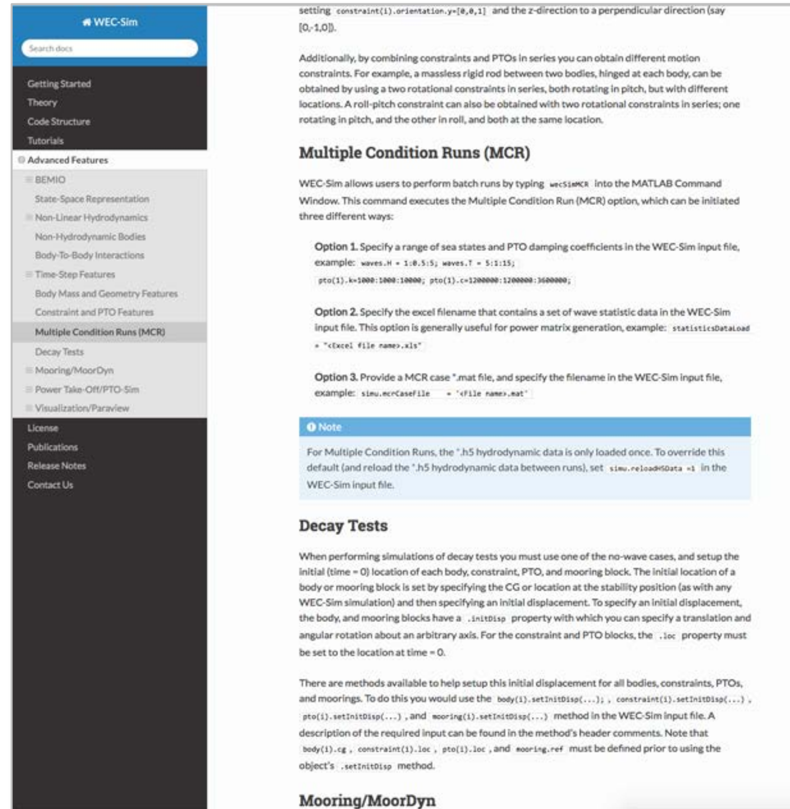
MCR

Multiple Condition Runs

Yi-Hsiang Yu (NREL)

Multiple Condition Runs (MCR)

- WEC-Sim allows users to run multiple cases using **wecSimMCR** (in the MATLAB Command Window)
- The MATLAB function file (wecSimMCR.m) is located under
  <WEC-Sim Path>/source/functions.
- Examples are provided in the "WEC-Sim Application" repository
  https://github.com/WEC-Sim/WEC-Sim_Applications



https://wec-sim.github.io/WEC-Sim/features.html#multiple-condition-runs-mcr

Overview of the MCR Presentation

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

- How does MCR work?

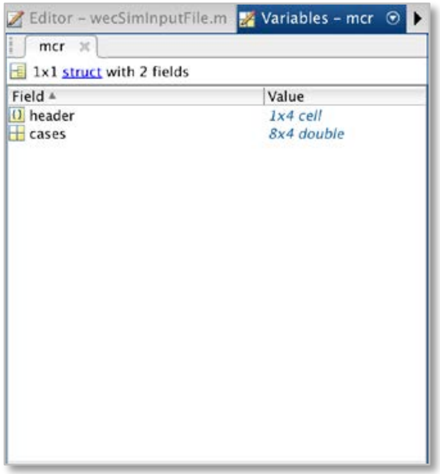
- Examples of running MCR
  https://github.com/WEC-Sim/WEC-Sim_Applications/tree/master/RM3_MCR
  – Specify a range of sea states and PTO damping coefficients
  – Using an excel file that contains a set of wave statistic data
  – User define option

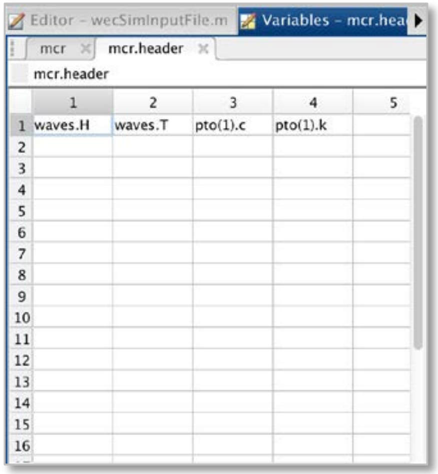
- MCR user defined function (userDefinedFunctionsMCR.m) and Post-processing

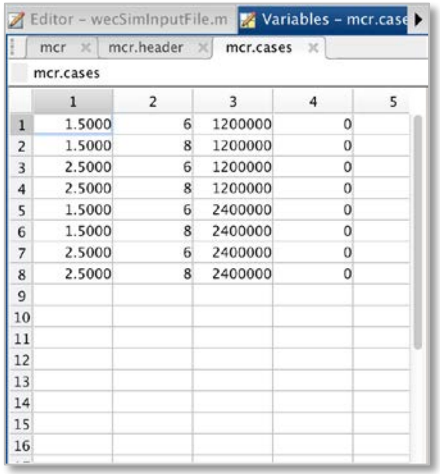
How Does MCR Work?

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

- Create a **mcr** function that includes all the parameters that is needed to run all multiple cases one after another automatically.
- **mcr** function includes:
  - mcr.header: the name of the parameters and functions
  - mcr.cases: the given value or “option” for the parameters and functions







How Does MCR Work?

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

- **wecSimMCR** will then execute wecSim.m file for each given case

Variables – mcr | Editor – wecSimMCR.m

wecSimInputFile.m | wecSimMCR.m | userDefinedFunctionsMCR.m

```matlab
        if (length(pto(n).c)>1 || length(pto(n).k)>1)
            numConditions=numConditions+2;
            name = sprintf('pto(%i).c', n);
            mcr.header{numConditions-1} = name;
            name = sprintf('pto(%i).k', n);
            mcr.header{numConditions  } = name;

            len = length(mcr.cases(:,1)); kkk = 0;
            for l2=1:length(pto(n).k)
                for l1=1:length(pto(n).c)
                    kkk=kkk+1;
                    mcr.cases(len*(kkk-1)+1:len*(kkk-1)+len,1:numConditions-2) = mcr.cases(1:len,1:numConditions-2);
                    mcr.cases(len*(kkk-1)+1:len*(kkk-1)+len,  numConditions-1) = pto(n).c(l1);
                    mcr.cases(len*(kkk-1)+1:len*(kkk-1)+len,    numConditions) = pto(n).k(l2);
                end
            end
        end
    end; clear i j k l1 l2 m n name nseed kkk len numConditions
  end
end
%%
% % Check to see if changing h5 file between runs
% % If one of the MCR headers is body(#).h5File, then the hydro data will be
% % loaded from the h5 file for each condition run.
% % reloadHydroDataFlag = true;
% if isempty(cell2mat(regexp(mcr.header, 'body\(\d+\).h5File')))
%     reloadHydroDataFlag = false;
%     clear hydroData
% end
%%
% Run WEC-Sim
warning('off','MATLAB:DELETE:FileNotFound'); delete('mcrCase*.mat')
for imcr=1:length(mcr.cases(:,1))
    wecSim;
    if exist('userDefinedFunctionsMCR.m','file') == 2; userDefinedFunctionsMCR; end
%%    %Store hydrodata in memory for reuse in future runs.
    if simu.reloadH5Data == 0 && imcr == 1        % Off->'0', On->'1', (default = 0)
        for ii = 1:simu.numWecBodies
            hydroData(ii) = body(ii).hydroData;
        end
    end
end; clear imcr ans hydroData
%%
```

- For each case, wecSim.m will overwrite the default parameters using the parameters and options described in the **mcr** function.

MCR Options

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

This command executes the Multiple Condition Run (MCR) option, which can be initiated three different ways:

- Option 1- Specify a range of sea states and PTO damping coefficients in the WEC-Sim input file.

- Option 2- Specify the excel filename that contains a set of wave statistic data in the WEC-Sim input file.

- Option 3- Provide a MCR case *.mat file, and specify the filename in the WEC-Sim input file.

MCR: Option 1

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

- Specify a range of sea states and PTO damping coefficients in the WEC-Sim input file
- Example: waves.H = 1.5:1:2.5; waves.T = 6:2:8; pto(1).c=1200000:1200000:2400000

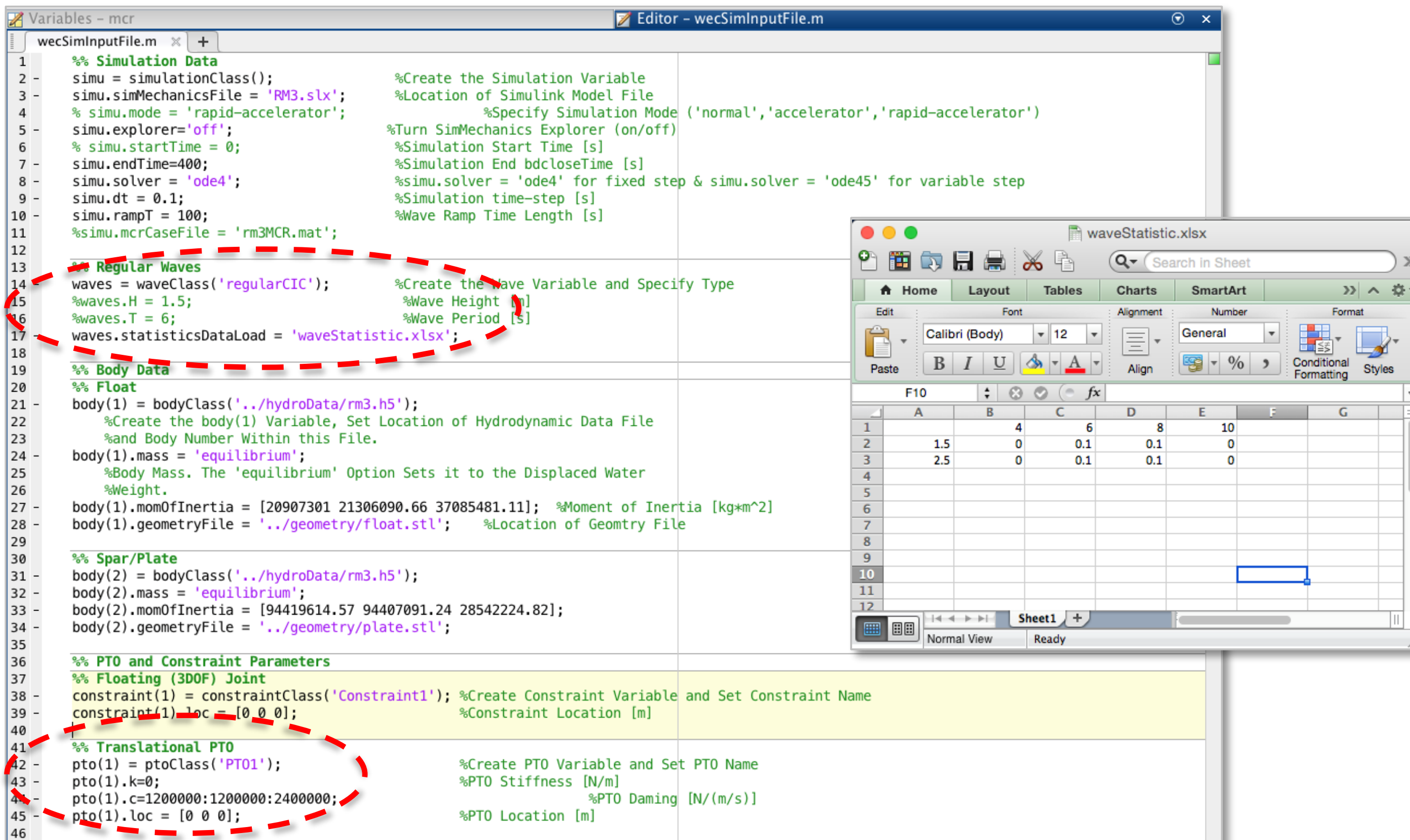
Editor – wecSimInputFile.m | Variables – mcr.cases

wecSimInputFile.m

```
%% Simulation Data
simu = simulationClass();                %Create the Simulation Variable
simu.simMechanicsFile = 'RM3.slx';       %Location of Simulink Model File
% simu.mode = 'rapid-accelerator';                   %Specify Simulation Mode ('normal','accelerator','rapid-accelerator')
simu.explorer='off';                    %Turn SimMechanics Explorer (on/off)
% simu.startTime = 0;                    %Simulation Start Time [s]
simu.endTime=400;                        %Simulation End bdcloseTime [s]
simu.solver = 'ode4';                    %simu.solver = 'ode4' for fixed step & simu.solver = 'ode45' for variable step
simu.dt = 0.1;                           %Simulation time-step [s]
simu.rampT = 100;                        %Wave Ramp Time Length [s]
%simu.mcrCaseFile = 'rm3MCR.mat';

%% Regular Waves
waves = waveClass('regularCIC');         %Create the Wave Variable and Specify Type
waves.H = 1.5:1:2.5;                           %Wave Height [m]
waves.T = 6:2:8;                             %Wave Period [s]

%% Body Data
%% Float
body(1) = bodyClass('hydroData/rm3.h5');
    %Create the body(1) Variable, Set Location of Hydrodynamic Data File
    %and Body Number Within this File.
body(1).mass = 'equilibrium';
    %Body Mass. The 'equilibrium' Option Sets it to the Displaced Water
    %Weight.
body(1).momOfInertia = [20907301 21306090.66 37085481.11];  %Moment of Inertia [kg*m^2]
body(1).geometryFile = 'geometry/float.stl';    %Location of Geomtry File

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

%% PTO and Constraint Parameters
%% Floating (3DOF) Joint
constraint(1) = constraintClass('Constraint1'); %Create Constraint Variable and Set Constraint Name
constraint(1).loc = [0 0 0];                    %Constraint Location [m]

%% Translational PTO
pto(1) = ptoClass('PTO1');                      %Create PTO Variable and Set PTO Name
pto(1).k=0;                                     %PTO Stiffness [N/m]
pto(1).c=1200000:1200000:2400000;                               %PTO Daming [N/(m/s)]
pto(1).loc = [0 0 0];                           %PTO Location [m]
```

MCR: Option 2

- Specify the excel filename that contains a set of wave statistic data in the WEC-Sim input file.
- This option is generally useful for power matrix generation, example: waves.statisticsDataLoad = "<Excel file name>.xlsx"



MCR: Option 3

- Provide a MCR case *.mat file, and specify the filename in the WEC-Sim input file, example: simu.mcrCaseFile = '<File name>.mat'
- Option 3 **overrules** Option 2 & Option 1

Variables – mcr | Editor – wecSimInputFile.m

wecSimInputFile.m

```
%% Simulation Data
simu = simulationClass();                 %Create the Simulation Variable
simu.simMechanicsFile = 'RM3.slx';        %Location of Simulink Model File
% simu.mode = 'rapid-accelerator';                   %Specify Simulation Mode ('normal','accelerator','rapid-accelerator')
simu.explorer='off';                     %Turn SimMechanics Explorer (on/off)
% simu.startTime = 0;                     %Simulation Start Time [s]
simu.endTime=400;                         %Simulation End bdcloseTime [s]
simu.solver = 'ode4';                     %simu.solver = 'ode4' for fixed step & simu.solver = 'ode45' for variable step
simu.dt = 0.1;                            %Simulation time-step [s]
simu.rampT = 100;                         %Wave Ramp Time Length [s]
simu.mcrCaseFile = 'mcrExample.mat';

%% Regular Waves
waves = waveClass('regularCIC');          %Create the Wave Variable and Specify Type
waves.H = 1.5;                            %Wave Height [m]
waves.T = 8;                              %Wave Period [s]

%% Body Data
%% Float
body(1) = bodyClass('../hydroData/rm3.h5');
    %Create the body(1) Variable, Set Location of Hydrodynamic Data File
    %and Body Number Within this File.
body(1).mass = 'equilibrium';
    %Body Mass. The 'equilibrium' Option Sets it to the Displaced Water
    %Weight.
body(1).momOfInertia = [20907301 21306090.66 37085481.11];  %Moment of Inertia [kg*m^2]
body(1).geometryFile = '../geometry/float.stl';    %Location of Geomtry File

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

%% PTO and Constraint Parameters
%% Floating (3DOF) Joint
constraint(1) = constraintClass('Constraint1'); %Create Constraint Variable and Set Constraint Name
constraint(1).loc = [0 0 0];                    %Constraint Location [m]

%% Translational PTO
pto(1) = ptoClass('PTO1');                      %Create PTO Variable and Set PTO Name
pto(1).k=0;                                     %PTO Stiffness [N/m]
pto(1).c=1200000;                               %PTO Daming [N/(m/s)]
pto(1).loc = [0 0 0];                           %PTO Location [m]
```

MCR user defined function & Post-processing

- "**userDefinedFunctionsMCR.m**" file allows user to add post processing script
  - To analyze the simulation results and create plots automatically
  - To save the SELECTED data from each simulation in different name to
    - Avoid overwriting the output *.mat file under the output folder
    - Minimize the size of the output data for MCR simulations
- **imcr** is the indexing number for each case

Editor – /Users/yyu/Desktop/RM3_MCR/RM3_MCROPT3/userDefinedFunctionsMCR.m

wecSimInputFile.m | wecSimMCR.m | userDefinedFunctionsMCR.m

```
%Example of user input MATLAB file for MCR post processing
%filename = ['savedData',sprintf('%03d', imcr),'.mat'];
filename = sprintf('savedData%03d.mat', imcr);

mcr.Avgpower(imcr) = mean(output.ptos.powerInternalMechanics(2000:end,3));
mcr.CPTO(imcr)  = pto(1).c;

save (filename, 'mcr','output','waves');

```

- Run multiple MCR cases by using additional instance of MATLAB

Thank you!

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

Upcoming scheduled webinars and training courses…

Advanced Feature Webinars *1hr each*

- **May 24:** nl-hydro, b2b, non-hydro and drag
- **June 7:** PTO and control, application for desalination
- **July 18:** Mooring and visualization

Training Courses

- **May 1:** *1hr* WEC-Sim workshop at METS, for new users

U.S. DEPARTMENT OF ENERGY

Sandia National Laboratories

NREL NATIONAL RENEWABLE ENERGY LABORATORY