Advanced Features Course

Online Training Materials

PRESENTED BY

Kelley Ruehl, Sandia

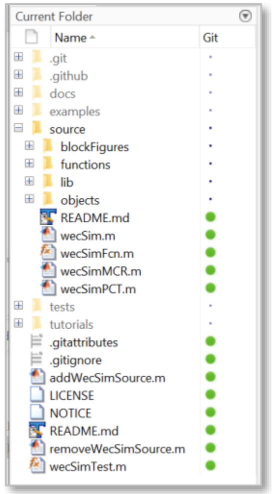
WEC-Sim Batch Runs

WEC-Sim Training- Advanced Features

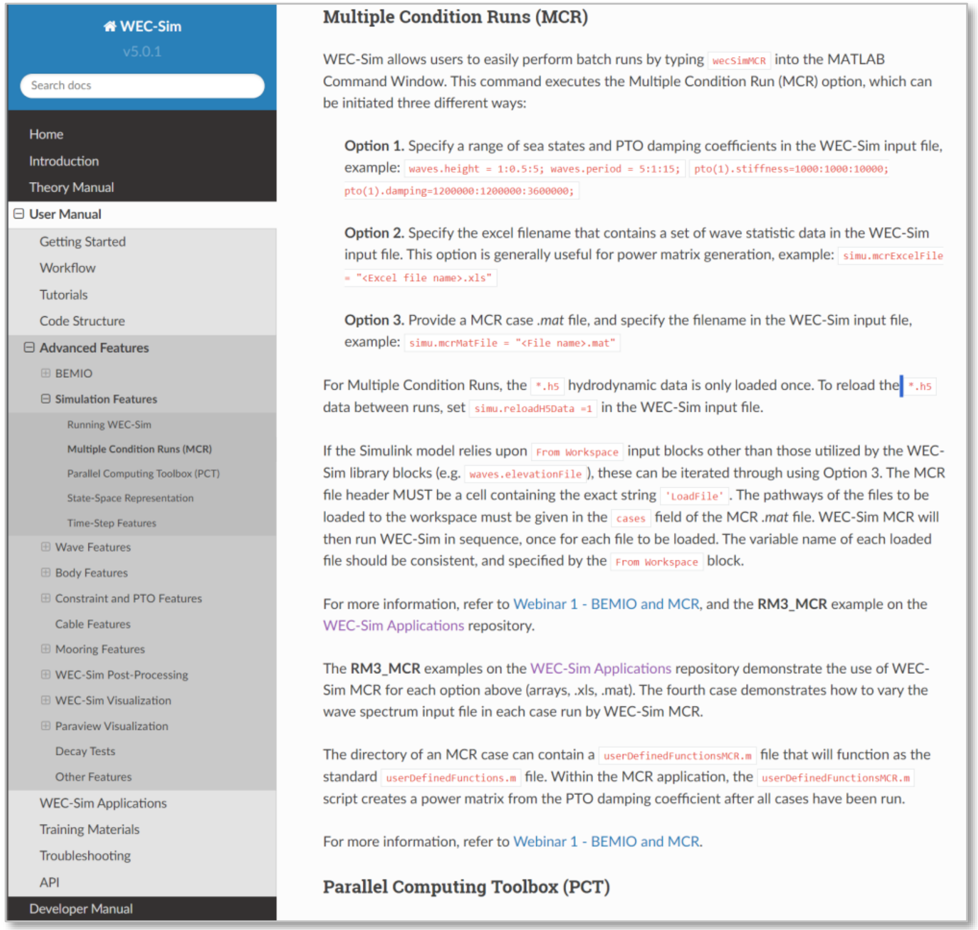
Multiple Condition Runs (MCR)

WEC-Sim allows users to run multiple cases using **wecSimMCR** (in the MATLAB Command Window)

The MATLAB executable (**wecSimMCR.m**) is located in the following directory:
<WEC-Sim Path>/source/



Examples are provided in the WEC-Sim Applications repository:
https://github.com/WEC-Sim/WEC-Sim_Applications



Multiple Condition Runs (MCR)

WEC-Sim allows users to easily perform batch runs by typing `wecSimMCR` into the MATLAB Command Window. 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, example: `waves.height = 1:0.5:5; waves.period = 5:1:15;` `pto(1).stiffness=1000:1000:10000;` `pto(1).damping=1200000:1200000:3600000;`

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: `simu.mcrExcelFile = "<Excel file name>.xls"`

Option 3. Provide a MCR case *.mat* file, and specify the filename in the WEC-Sim input file, example: `simu.mcrMatFile = "<File name>.mat"`

For Multiple Condition Runs, the `*.h5` hydrodynamic data is only loaded once. To reload the `*.h5` data between runs, set `simu.reloadH5Data =1` in the WEC-Sim input file.

If the Simulink model relies upon `From Workspace` input blocks other than those utilized by the WEC-Sim library blocks (e.g. `waves.elevationFile`), these can be iterated through using Option 3. The MCR file header MUST be a cell containing the exact string `'LoadFile'`. The pathways of the files to be loaded to the workspace must be given in the `cases` field of the MCR *.mat* file. WEC-Sim MCR will then run WEC-Sim in sequence, once for each file to be loaded. The variable name of each loaded file should be consistent, and specified by the `From Workspace` block.

For more information, refer to Webinar 1 - BEMIO and MCR, and the **RM3_MCR** example on the WEC-Sim Applications repository.

The **RM3_MCR** examples on the WEC-Sim Applications repository demonstrate the use of WEC-Sim MCR for each option above (arrays, .xls, .mat). The fourth case demonstrates how to vary the wave spectrum input file in each case run by WEC-Sim MCR.

The directory of an MCR case can contain a `userDefinedFunctionsMCR.m` file that will function as the standard `userDefinedFunctions.m` file. Within the MCR application, the `userDefinedFunctionsMCR.m` script creates a power matrix from the PTO damping coefficient after all cases have been run.

For more information, refer to Webinar 1 - BEMIO and MCR.

Parallel Computing Toolbox (PCT)

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

How Does MCR Work?

How does MCR work?

- Define multiple WEC-Sim cases to run (with different input conditions)
- Type **wecSimMCR** in the Command Window
- WEC-Sim (**wecSim**) is executed in series for each case
- User-defined post-processing script (**userDefinedFunctionsMCR.m**) is executed after each case
- Batch run is complete

Examples of running MCR:

https://github.com/WEC-Sim/WEC-Sim_Applications/tree/master/Multiple_Condition_Runs

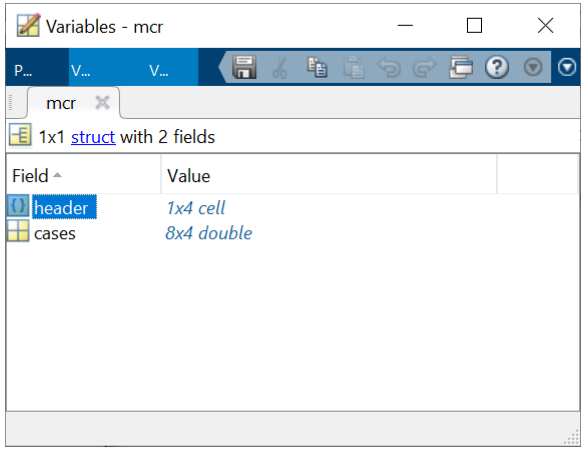
- Specify a range of sea states and PTO damping coefficients in the input file
- Using an excel (***.xls**) file that contains a set of wave statistic data
- Using a MATLAB (***.mat**) file that specifies desired input conditions

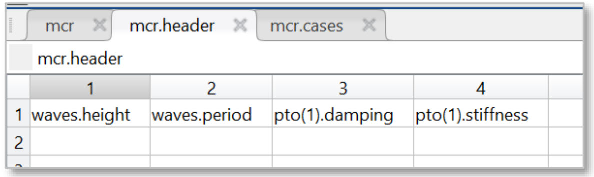
How Does MCR Work?

WEC-Sim creates a **mcr** structure that includes the information needed to automatically run the desired WEC-Sim cases.

mcr structure includes:

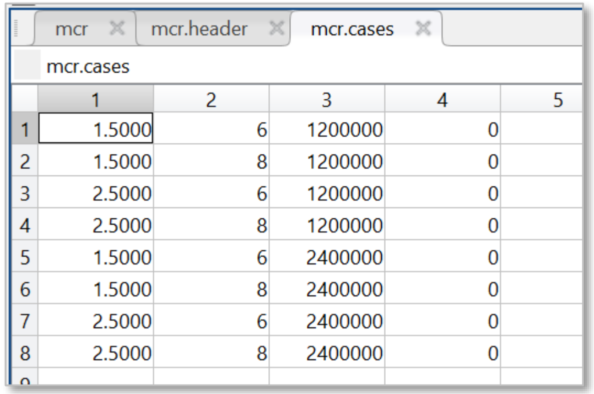
- **mcr.header**: the names of the parameters and functions
- **mcr.cases**: the given values or “options” for the parameters and functions





mcr.header

	1	2	3	4
1	waves.height	waves.period	pto(1).damping	pto(1).stiffness
2				



mcr.cases

	1	2	3	4	5
1	1.5000	6	1200000	0	
2	1.5000	8	1200000	0	
3	2.5000	6	1200000	0	
4	2.5000	8	1200000	0	
5	1.5000	6	2400000	0	
6	1.5000	8	2400000	0	
7	2.5000	6	2400000	0	
8	2.5000	8	2400000	0	

How Does MCR Work?

wecSimMCR then executes **wecSim** for each case

```
%% Execute wecSimMCR
% 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.numHydroBodies
            hydroData(ii) = body(ii).hydroData;
        end
    end
end; clear imcr ans hydroData
```

For each case, **wecSim** will overwrite the parameters defined in the input file using the values defined in the **mcr** structure.

MCR Options

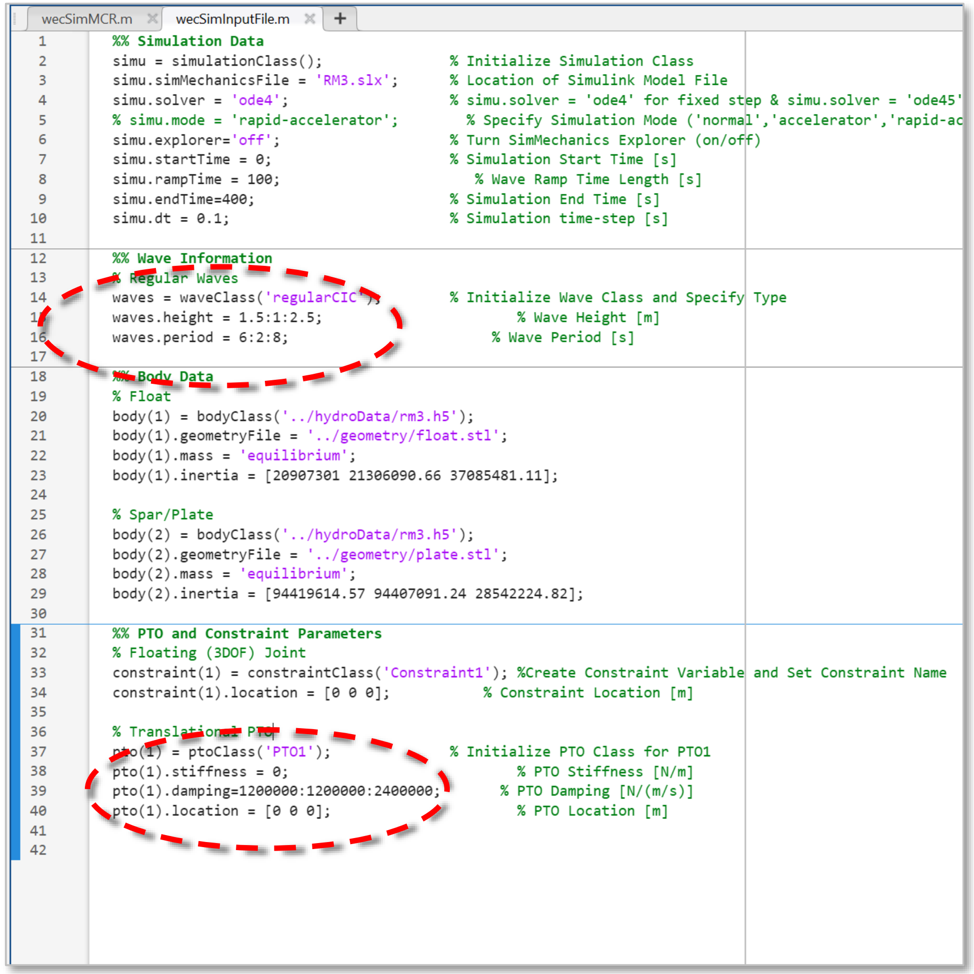
wecSimMCR executes the Multiple Condition Run (MCR) option, which can be initiated three different ways:

- Option 1 - Specify a range of sea states and PTO damping coefficients in the WEC-Sim input file.
- Option 2 - Specify the ***xls** file 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.
  - This option allows for any input variable to be altered within each MCR case such as wave spectra, mooring setup, PTO components, etc

MCR: Option 1

Specify a range of sea states and PTO damping coefficients in the WEC-Sim input file, for example:

waves.height = 1.5:1:2.5;
waves.period = 6:2:8;
pto(1).damping=1200000:1200000:2400000;



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

%% Wave Information
% Regular Waves
waves = waveClass('regularCIC');            % Initialize Wave Class and Specify Type
waves.height = 1.5:1:2.5;                           % Wave Height [m]
waves.period = 6:2:8;                            % Wave Period [s]

%% Body Data
% Float
body(1) = bodyClass('../hydroData/rm3.h5');
body(1).geometryFile = '../geometry/float.stl';
body(1).mass = 'equilibrium';
body(1).inertia = [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).inertia = [94419614.57 94407091.24 28542224.82];

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

% Translational PTO
pto(1) = ptoClass('PTO1');                  % Initialize PTO Class for PTO1
pto(1).stiffness = 0;                               % PTO Stiffness [N/m]
pto(1).damping=1200000:1200000:2400000;           % PTO Damping [N/(m/s)]
pto(1).location = [0 0 0];                          % PTO Location [m]
```

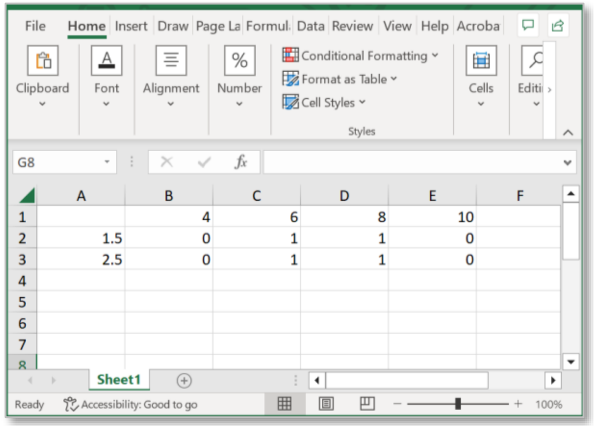
https://github.com/WEC-Sim/WEC-Sim_Applications/tree/master/Multiple_Condition_Runs

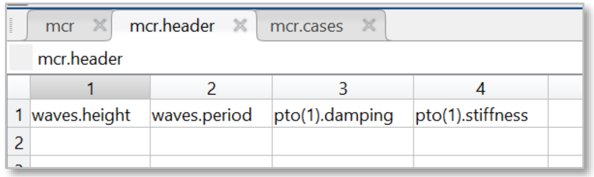
MCR: Option 2

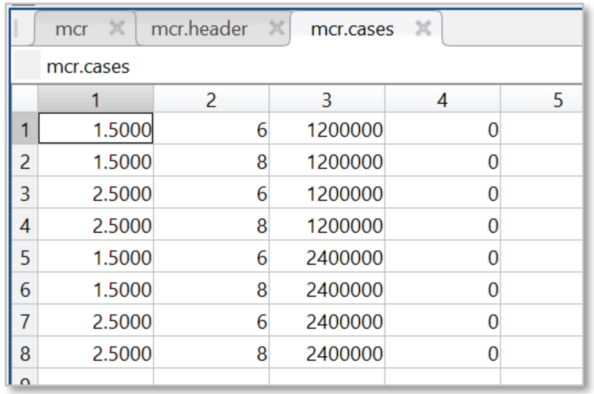
Specify the excel filename that contains a set of wave statistic data in the WEC-Sim input file.

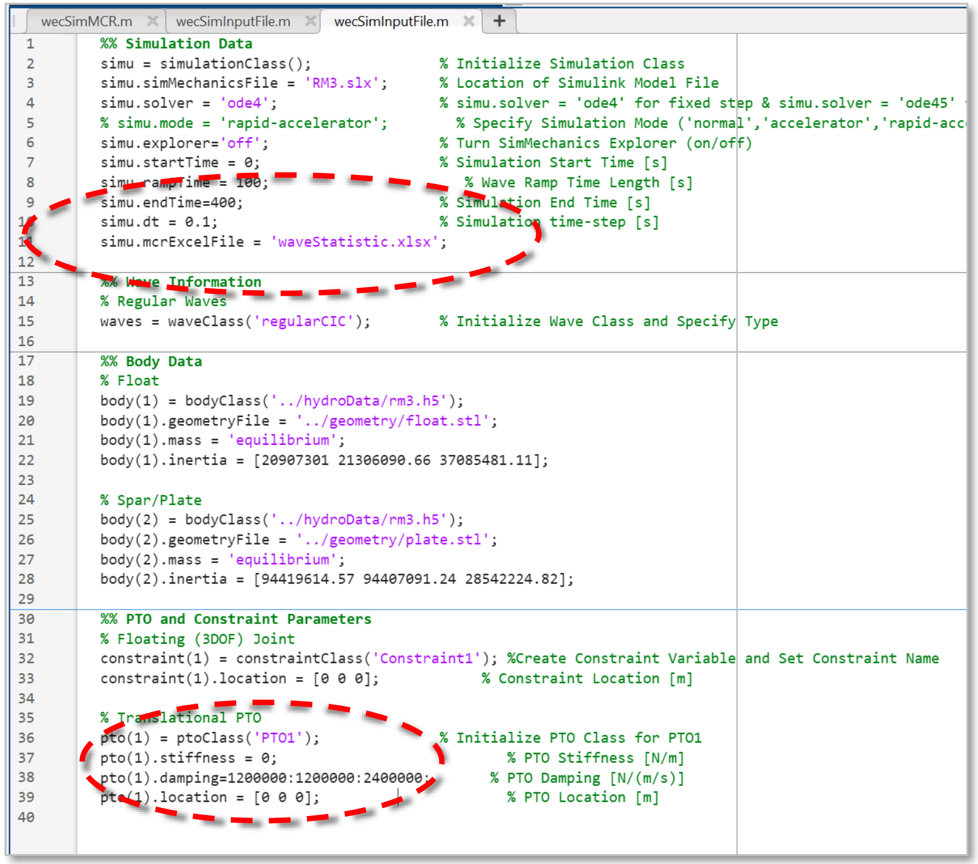
This option is generally useful generating power matrices, for example:

simu.mcrExcelFile = "<Excel file name>.xlsx









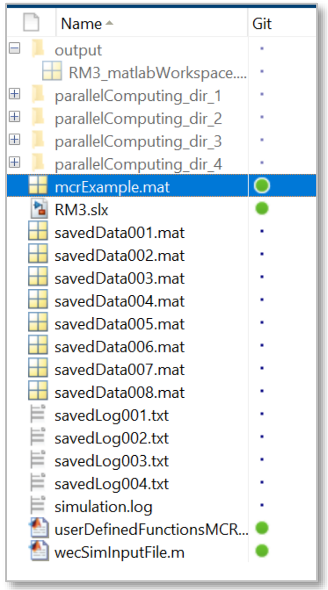
NOTE: Row 1 is wave period, Column A is wave height, values are binary on/off

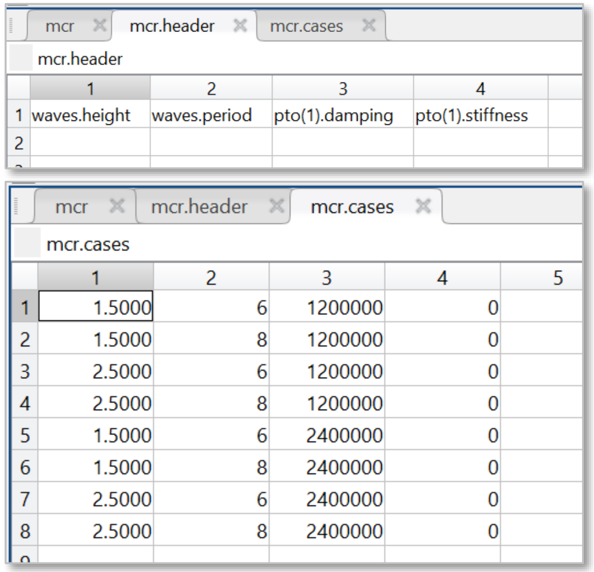
https://github.com/WEC-Sim/WEC-Sim_Applications/tree/master/Multiple_Condition_Runs

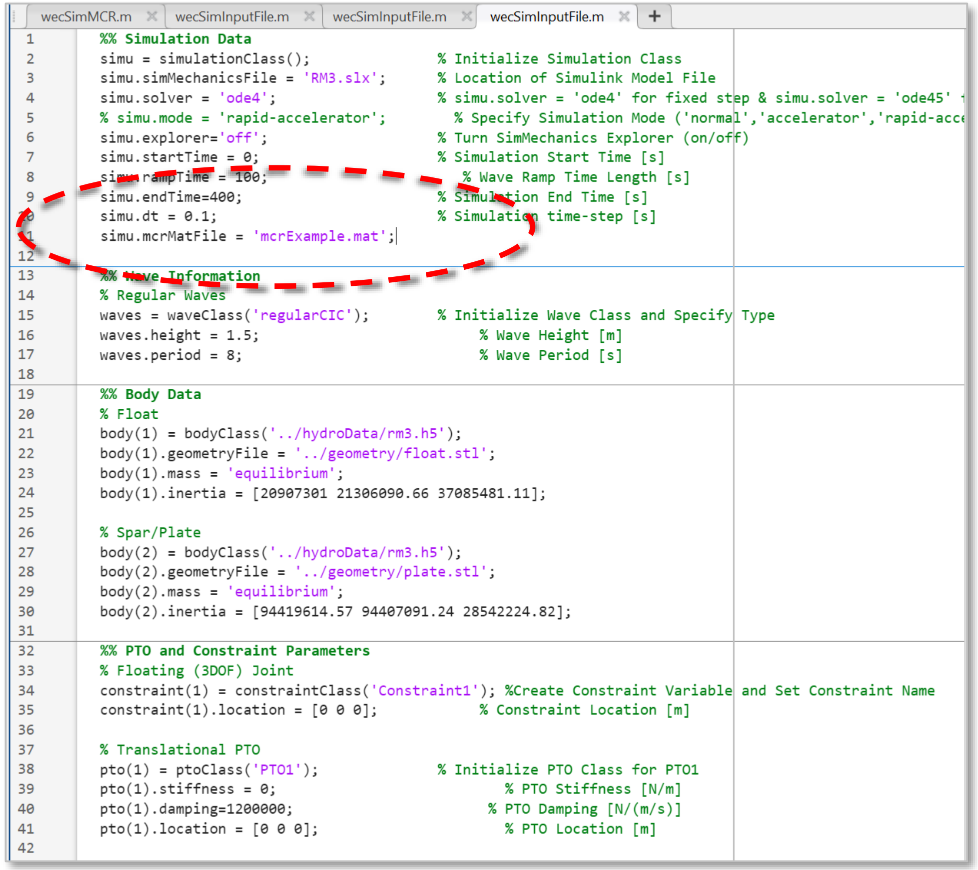
MCR: Option 3

Provide a MCR case ***.mat** file, and specify the filename in the WEC-Sim input file, for example: **simu.mcrMatFile = '<File name>.mat'**

NOTE: Option 3 overrides Option 1 & Option 2







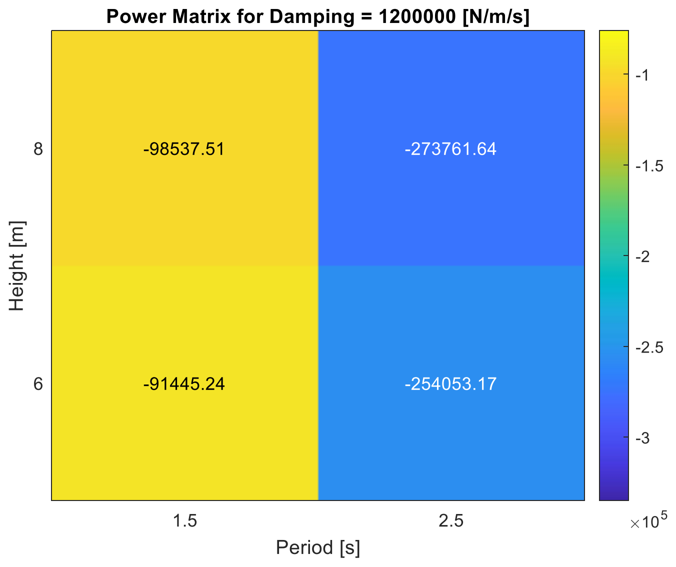
https://github.com/WEC-Sim/WEC-Sim_Applications/tree/master/Multiple_Condition_Runs

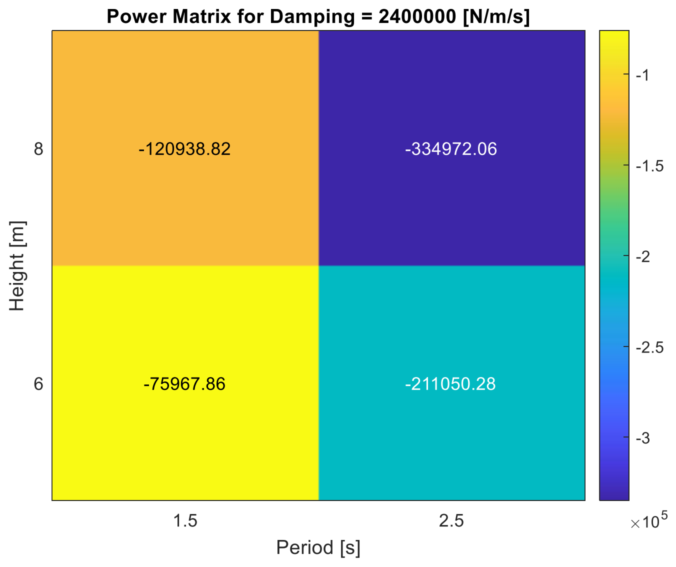
MCR user defined function & Post-processing

userDefinedFunctionsMCR.m script allows users to add a custom post-processing script

- Analyze the simulation results and create plots automatically
- Save SELECTED data from each simulation to a different name:
  - Avoids overwriting the output *.mat file under the output folder
  - Minimizes the size of the output data for MCR simulations

imcr is the indexing number for each case





wecSimInputFile.m | wecSimInputFile.m | wecSimInputFile.m | userDefinedFunctionsMCR.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).damping;

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

%% Plot Power Matrix

if imcr == length(mcr.cases);
    H = mcr.cases(:,1);
    T = mcr.cases(:,2);
    c = mcr.cases(:,3);
    P = mcr.Avgpower';

    % Damping = 1200000
    figure
    mat = [mcr.Avgpower(2) mcr.Avgpower(4);...      % Create Power Matrix
        mcr.Avgpower(1) mcr.Avgpower(3)];
    imagesc(mat);                                    % Create a colored plot of the matrix values
    colormap parula
    caxis([min(P) max(P)])

    textStrings = num2str(mat(:),'%0.2f');           % Create strings from the matrix values
    textStrings = strtrim(cellstr(textStrings));     % Remove any space padding
    [x,y] = meshgrid(1:2);                           % Create x and y coordinates for the strings
    hStrings = text(x(:),y(:),textStrings(:),...     % Plot the strings
                'HorizontalAlignment','center');
    midValue = mean(get(gca,'CLim'));                % Get the middle value of the color range
    textColors = repmat(mat(:) < midValue,1,3);      % Choose white or black for the text color
    set(hStrings,{'Color'},num2cell(textColors,2)); % Change the text colors
    colorbar
    set(gca,'XTick',1:2,...                          % Change the axes tick marks
            'XTickLabel',{'1.5','2.5'},...           % and tick labels
            'YTick',1:2,...
            'YTickLabel',{'8','6'},...
            'TickLength',[0 0]);
    xlabel('Period [s]')
    ylabel('Height [m]')
    title(['Power Matrix for Damping = ' num2str(c(1)) ' [N/m/s]'])
```

Parallel Computing Toolbox (PCT)

WEC-Sim also allows users runs cases in parallel using **wecSimPCT** (in the Command Window)

- Requires the **MATLAB Parallel Computing Toolbox (PCT)**
- Allows parallel capability for batch runs, not series like Multiple Condition Runs (MCR)
- Speeds up large simulations
- Similar to MCR, PCT can be executed with Options 1-3

Important Notes:

- **wecSimPCT** is not compatible with **userDefinedFunctionsMCR.m**. Please use **userDefinedFunctions.m** instead.
- For PCT runs, the ***.h5** hydrodynamic data must be reloaded, regardless of the setting for **simu.reloadH5Data** in the WEC-Sim input file.

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.

Sandia National Laboratories

NREL