

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

	# WEC-Sim v5.0.1
	Search docs
	Home Introduction
	User Manual
	Getting Started
	Workflow
	Tutorials
	Code Structure
•	Advanced Features
	BEMIO
	Simulation Features
	Running WEC-Sim
	Multiple Condition Runs (MCR)
	Parallel Computing Toolbox (PCT)
	State-Space Representation
	Time-Step Features
	Wave Features
	Body Features
	Constraint and PTO Features
	Cable Features
	Hooring Features
	WEC-Sim Post-Processing
	WEC-Sim Visualization
	Decay Tests
	Other Features
	WEC-Sim Applications
-	Training Materials
	Troubleshooting
	API

Current Folder

.git

.github

docs examples

source

🗉 📜 objects

tests tutorial

igitattributes igitignore

LICENSE

KEADME.md

wecSim.m wecSimFcn.m wecSimMCR.m

wecSimPCT.m

addWecSimSource.m

🕀 📜 lib

BlockFigure
 functions

Multiple Condition Runs (MCR)

WEC-Sim allows users to easily perform batch runs by typing wecsimeral 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=1000:10000; pto(1).damping=1200000:1200000; 2000000;

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
"cExcel file name>.xls"

Option 3. Provide a MCR case .mat file, and specify the filename in the WEC-Sim input file, example: simu.mcrHatFile = "cFile name.mat"

For Multiple Condition Runs, the *.hs hydrodynamic data is only loaded once. To reload the *.hs data between runs, set simu.reload+soata =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 "toadFile". 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 userDefinedFunctionsKR.m file that will function as the standard userDefinedFunctions.m file. Within the MCR application, the userDefinedFunctionsKR.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)

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

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:

- 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

P V	V		Ph.	(E)		_	-	
mcr 🗶			93		96		? 💿	\odot
\rm 1x1 <u>struct</u> wit	th 2 fields							
Field 🔺	Value							
• header cases	1x4 ce 8x4 do	ll puble						

	mcr × mcr.header × mcr.cases ×					
	mcr.header					
	1	2	3	4		
1	waves.height	waves.period	pto(1).damping	pto(1).stiffness		
2						
-						

	mcr 🗙	mcr.header	× mcr.case	s 🗙		
	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		
0						

How Does MCR Work?

wecSimMCR then executes wecSim for each case



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:

- <u>Option 1</u> 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.
- <u>Option 3</u> 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 I

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;
```



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



	mcr 🗙 mcr.header 🗶 mcr.cases 🗶						
	mcr.header						
	1	2	3	4			
1	waves.height	waves.period	pto(1).damping	pto(1).stiffness			
2							
~							

	mcr 🗙 I	mcr.header	× mcr.case	s ×	
	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	
0					



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

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

[Name 🗠	Git
-		output	1.00
		🖥 RM3_matlabWorkspace	1 C
+	1	parallelComputing_dir_1	1.00
+		parallelComputing_dir_2	1 C
+		parallelComputing_dir_3	1.00
+	1	parallelComputing_dir_4	1.00
		mcrExample.mat	
	1	RM3.slx	•
		savedData001.mat	1 (C)
		savedData002.mat	1.00
		savedData003.mat	1.00
		savedData004.mat	· .
		savedData005.mat	· .
		savedData006.mat	· .
	ш	savedData007.mat	·
	H	savedData008.mat	· .
	F	savedLog001.txt	· .
	Ē	savedLog002.txt	· .
	Ē	savedLog003.txt	· .
	Ē	savedLog004.txt	
	Ē	simulation.log	
		userDefinedFunctionsMCR	•
	1	wecSimInputFile.m	•

mcr × mcr.header × mcr.cases ×					
mcr.header					
	1	2	3	4	
1 waves.height		waves.period pto(1).damping		g pto(1).stiffn	ess
2					
_					
	mcr 🛛	mcr.header	× mcr.case	s ×	
	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	
0					



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



<pre>%Example of user input MATLAB file for MCR post processing %ilename = ['savedData',sprintf('%93d', imcr);'.mat']; filename = sprintf('savedDataK2d.mat', imcr); mcr.Avgpower(imcr) = mean(output.ptos.powerInternalMechanics(2000:end,3)); mcr.CPT0(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(:,2); c = mcr.cases(:,2); f = mcr.Avgpower(2) mcr.Avgpower(4); % Create Power Matrix mcr.Avgpower(1) mcr.Avgpower(3)]; imagesc(mat); colormap parula caxis([min(P) max(P)]) textStrings = num2str(mat(:),'%0.2f'); % Create a colored plot of the matrix values textStrings = strtim(cellstr(textStrings)); % Remove any space padding [X,y] = mesnprid(1:2); % Create in y coordinates for the strings hStrings = text(x(:),y(:),textStrings(:), % Plot the strings 'HorizontalAlignment', 'center'); midValue = mean(get(gca, 'CLim')); % Get the middle value pf the color range textColors = repmt(mat(:) < midValue,1,3); % Choose white or black for the text color set(hStrings, ('Color'},num2cell(textColors,2)); % Choose white or black for the text color set(name, 'Itick.abel','1'.5', '2.5'), % Chonse the axes tick marks 'XTicktabel','(1'.5', '2.5'), % And tick labels 'Ytick',1:2, 'YTick',1:2, 'Yti</pre>	1	wecSimInputFile.m 🛪 wecSimInputFile.m 🛪 wecSimInputFile.m 🛪 userDefinedFunctionsMCR.m 🛪 userDefinedFunctionsMCR.m
<pre>%#ilename = ['savedData',sprintf('%803', incr),'.mat']; filename = sprintf('savedData%803d.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(2) mcr.Avgpower(4); % Create Power Matrix mcr.Avgpower(1) mcr.Avgpower(4); % Create Power Matrix mcr.Avgpower(2) mcr.Avgpower(4); % 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 = strtim(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 'Horinotallalignment','center'); midValue = mean(get(gca,'CLim')); % Get the middle value of the color range textColors = repmat(mat(:) < midValue,1,3); % Choage the text colors colorbar set(gca,'XTick',1:2, % Change the axes tick marks 'XTickLabel',{'1.5', '2.5'}, 'TickLabel',{'2', '6'}, 'TickLabel',{'2', '6'}, 'TickLabel',{'2', '6'}, 'TickLabel',{'1.5', '2.5'}, 'TickLabel',{'1.5', '2.5'},</pre>	G	%Example of user input MATLAB file for MCR post processing
<pre>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(:,2); c = mcr.cases(:,3); P = mcr.Avgpower(2) mcr.Avgpower(4); % Create Power Matrix mcr.Avgpower(2) mcr.Avgpower(4); % Create Power Matrix mcr.Avgpower(2) mcr.Avgpower(4); % 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 = num2str(mat(:),'%0.2f'); % Create x and y coordinates for the strings hStrings = text(x(:),y(:),textStrings(:), % Plot the strings 'HorizontalAlignment','center'); midValue = mean(get(gca,'Ctim')); % Get the middle value of the color range textColors = repmat(mat(:) < "Module_1,3); % Change the text colors colorbar set(hStrings,'Color'},num2cell(textColors,2)); % Change the text colors colorbar set(gca, 'Xtick',1:2, % Change the axes tick marks 'Xticktabel',('1.5','2.5'); % and tick labels 'Ytick',1:2, 'Yticktabel',('2','6'), 'Ticktapt',(ie 0); xlabel('Period [s]') ylabel('Height [m]')</pre>		<pre>%filename = ['savedData',sprintf('%03d', imcr),'.mat'];</pre>
<pre>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(2) mcr.Avgpower(4); % Create Power Matrix mcr.Avgpower(2) mcr.Avgpower(4); % Create Power Matrix mcr.Avgpower(2) mcr.Avgpower(4); % 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 = strtim(cellstr(textStrings)); % Remove any space padding [x,y] = meshgrid(1:2); % Create x and y coordinates for the strings 'HorinotalAlignment','center'); midValue = mean(get(gca,'Clim')); % Get the middle value of the color range textColors = repmat(mat(:) <i %="" black="" choase="" colors<br="" for="" midvalue,1,3);="" or="" text="" the="" white="">colorbar set(hstrings,'Color'),num2cel(textColors,2)); % Change the text colors colorbar set(sci,'Xiick',1:2, % Change the axes tick marks 'Xiicktabel','('a','c'), 'Ticktanet','('a','c'),</i></pre>		<pre>filename = sprintf('savedData%03d.mat', imcr);</pre>
<pre>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(:,2); c = mcr.cases(:,2); c = mcr.cases(:,3); P = mcr.Avgpower(2) mcr.Avgpower(4); % Create Power Matrix mcr.Avgpower(1) mcr.Avgpower(3)]; imagesc(mat); 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 strings from the matrix values textStrings = 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(:) ('mdvalue,1,3); % Choose white or black for the text color set(hStrings, 'Color'),num2cell(textColors,2)); % Change the axes tick marks 'YTicktabel',{1:2,.</pre>		
<pre>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(2) mcr.Avgpower(4); % Create Power Matrix mcr.Avgpower(1) mcr.Avgpower(3)]; imagesc(mat); colormap parula caxis([min(P) max(P)]) textStrings = num2str(mat(:),'%0.2f'); % Create strings from the matrix values colormap parula caxis([min(P) max(P)]) textStrings = strtim(cellstr(textStrings)); % Remove any space padding [x,y] = meshgrid(1:2); % Create x and y coordinates for the strings</pre>		<pre>mcr.Avgpower(imcr) = mean(output.ptos.powerInternalMechanics(2000:end,3));</pre>
<pre>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); colornap 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 strings from the matrix values textStrings = text(x(:),y(:),textStrings(:), % Plot the strings hStrings = text(x(:),y(:),textStrings(:), % Plot the strings 'HorizontalAlignment','center'); midValue = mean(get(gca, 'Lim')); % Change the text color set(sca,'XTick',1:2, % Change the axes tick marks 'XTicktabel',{'1.5', '2.5'}, % Change</pre>		<pre>mcr.CPTO(imcr) = pto(1).damping;</pre>
<pre>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); colormap parula caxis([min(P) max(P)]) textStrings = num2str(mat(:),'%0.2f'); % Create a colored plot colormap parula caxis([min(P) max(P)]) textStrings = strtrim(cellstr(textStrings)); % Remove any space padding [x,y] = meshgrid(12); % Oreate 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); % Choase white or black for the text color set(hStrings,{'Color'},num2cell(textColors,2)); % Change the axes tick arks 'XTickLabel',{'1.5', '2.5'}, % Change the axes tick arks 'XTickLabel',{'1.5', '2.5'},</pre>		
<pre>%% 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)]; inagesc(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 = num2str(mat(:),'%0.2f'); % Create strings from the matrix values textStrings = strtnim(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,'Clin')); % 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 'XTicktabel',('1s', '2.5'), % and tick labels 'YTick',1:2,</pre>		<pre>save (filename, 'mcr','output','waves');</pre>
<pre>X% 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); colormap parula caxis([mi(P) max(P)]) textStrings = num2str(mat(:),'%0.2f'); % Create strings from the matrix values textStrings = num2str(mat(:),'%0.2f'); % Create strings from the matrix values textStrings = strtmic(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 axes tick marks 'XTicktabel',{'15', '2.5'}, 'YTicktabel',{'8','6'}, 'YTicktabel',{'8','6'}, 'Ticktabel',{'8','6'}, 'Ticktabel',{'8','6'},</pre>		
<pre>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); colormap parula caxis([min(P) max(P)]) textStrings = num2str(mat(:),'%0.2f'); % Create a colored plot of the matrix values 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</pre>		%% Plot Power Matrix
<pre>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); colormap parula caxis([min(P) max(P)]) textStrings = num2str(mat(:),'%0.2f'); % Create a colored plot of the matrix values 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); % Choage the text color set(hStrings, {'Clor'},num2cell(textColors,2)); % Change the text color set(strings, {'Clor'}, 'YTick',1:2, 'YTick',1:2, 'YTicktabel',{'8','6'}, 'Ticktength',[0 0]); xlabel('Period [s]') ylabel('Height [m]')</pre>		<pre>if imcr == length(mcr.cases);</pre>
<pre>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); % Choase white or black for the text color set(hStrings, 'Color'),num2cell(textColors,2)); % Change the axes tick xTicktabel',(1:5','2.5'], % and tick labels 'YTick',1:2, 'YTicktabel',(1'8','6'), 'Ticklength',[0 0]); xlabel('Height [m]')</pre>		H = mcr.cases(:,1);
<pre>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 colormap parula caxis([min(P) max(P)]) textStrings = num2str(mat(:),'%0.2f'); % Create strings from the matrix values textStrings = stntrim(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); % Choage the text color set(hStrings,{'Color'},num2cell(textColors,2)); % Change the axes tick marks 'XTicktabel',{'1.5','2.5'}, % and tick labels 'YTick',1:2, 'YTickLabel',{'8','6'}, 'TickLength',[0 e]); xlabel('Height [m]')</pre>		T = mcr.cases(:,2);
<pre>P = mcr.Avgpower'; % Damping = 1200000 figure mat = [mcr.Avgpower(2) mcr.Avgpower(4); % Create Power Matrix mcr.Avgpower(1) mcr.Avgpower(3)]; imagesc(mat); colormap parula caxis([min(P) max(P)]) textStrings = num2str(mat(:),'%0.2f'); % Create a colored plot of 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 'XTicktabel',{'15','2.5'}, % and tick labels 'YTick',1:2, 'YTicktabel',{'16', 'TickLength',[0 0]); xlabel('Height [m]')</pre>		c = mcr.cases(:,3);
<pre>% Damping = 120000 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 = strtim(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); % Choage the text colors colorbar set(fstrings, '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, 'YTick',1:2, 'YTick',1:2, 'YTicklabel','[0 0]); xlabel('Height [m]')</pre>		P = mcr.Avgpower';
<pre>% Damping = 120000 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,'XTick1:2, % Change the axes tick 'YTick1abel',{'1.5','2.5'}, % and tick labels 'YTick1abel',{'18','6'}, 'TickLength',[0 0]); xlabel('Height [m]')</pre>		
<pre>figure mat = [mcr.Avgpower(2) mcr.Avgpower(4); % Create Power Matrix mcr.Avgpower(1) mcr.Avgpower(3)]; imagesc(mat); % Create a colored plot colormap parula caxis([min(P) max(P)]) textStrings = num2str(mat(:), '%0.2f'); % Create strings from the matrix values textStrings = strtim(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); % Choase white or black for the text color set(hStrings, 'Color'),num2cell(textColors,2)); % Change the axes tick xTicktabel',{1:2, 'YTicktabel',{1:5,',2.5'}, % and tick labels 'YTick',1:2, 'YTicklegth',[0 e]); xlabel('Height [m]')</pre>		% Damping = 1200000
<pre>mat = [mcr.Avgpower(2) mcr.Avgpower(4); % Create Power Matrix mcr.Avgpower(1) mcr.Avgpower(3)]; imagesc(mat); % Create a colored plot 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, 'YTickdeal',('8','6'}, 'TickLength',[0 0]); xlabel('Period [5]') ylabel('Height [m]')</pre>		figure
<pre>mcr.Avgpower(1) mcr.Avgpower(3)]; imagesc(mat); % Create a colored plot 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, 'YTick',1:2, 'YTicklength',[0 0]); xlabel('Period [s]') ylabel('Height [m]')</pre>		<pre>mat = [mcr.Avgpower(2) mcr.Avgpower(4); % Create Power Matrix</pre>
<pre>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 'XTickLabel',('1.5','2.5']}, % and tick labels 'YTick',1:2, 'YTick',1:2, 'YTick',1:2, 'YTicklegel',('8','6'}, 'TickLength',[0 0]); xlabel('Height [m]')</pre>		<pre>mcr.Avgpower(1) mcr.Avgpower(3)];</pre>
<pre>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 'XTickLabel',('1.5','2.5']}, % and tick labels 'YTick',1:2, 'YTicklapel',[0 0]); xlabel('Heriod [5]') ylabel('Height [m]')</pre>		imagesc(mat); % Create a colored plot of the matrix values
<pre>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, 'YTickLapel',{'8','6'}, 'TickLength',[0 0]); xlabel('Period [5]') ylabel('Height [m]')</pre>		colormap parula
<pre>textStrings = nun2str(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, 'YTickLapel',{'8','6'}, 'TickLength',[0 0]); xlabel('Period [5]') ylabel('Height [m]')</pre>		<pre>caxis([min(P) max(P)])</pre>
<pre>textStrings = hum2str(mat(:), x0.2f);</pre>		
<pre>[x,y] meshgrid(1:2);</pre>		textStrings = num2str(ma((.), %0.21), % treate strings from the matrix values
<pre>hStrings = text(x(:),y(:),textStrings(:), % Plot the strings</pre>		[vv] = machgrid(1:2). (certificer:scripts), % nemove any space paduling
<pre>indefinition of the color range ('); midValue = mean(get(gca, 'CLim')); midValue = mean(get(gca, 'CLim'));</pre>		hStrings = tayt((\cdot) , (\cdot) taytStrings((\cdot) % Plot the strings
<pre>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, 'YTickLapel',{'8','6'}, 'TickLength',[0 0]); xlabel('Period [5]') ylabel('Height [m]')</pre>		'HorizontalAlignment', 'center'):
<pre>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]')</pre>		midValue = mean(get(gca, 'CLim')); % Get the middle value of the color range
<pre>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 [5]') ylabel('Height [m]')</pre>		textColors = repmat(mat(:) < midValue,1,3); % Choose white or black for the text color
<pre>colorbar set(gca,'XTick',1:2, % Change the axes tick marks</pre>		<pre>set(hStrings,{'Color'},num2cell(textColors,2)); % Change the text colors</pre>
<pre>set(gca, 'XTick',1:2, % Change the axes tick marks</pre>		colorbar
<pre>'XTickLabel',{'1.5','2.5'}, % and tick labels 'YTick',1:2, 'YTickLabel',{'8','6'}, 'TickLength',[0 0]); xlabel('Period [s]') ylabel('Height [m]')</pre>		set(gca,'XTick',1:2, % Change the axes tick marks
'YTick',1:2, 'YTickLabel',{'8','6'}, 'TickLength',[0 0]); xlabel('Period [s]') ylabel('Height [m]')		'XTickLabel',{'1.5','2.5'}, % and tick labels
'YTickLabel',('8','6'}, 'TickLength',('8'); xlabel('Period [s]') ylabel('Height [m]')		'YTick',1:2,
<pre>'TickLength',[0 0]); xlabel('Period [s]') ylabel('Height [m]')</pre>		'YTickLabel',{'8','6'},
<pre>xlabel('Period [s]') ylabel('Height [m]')</pre>		'TickLength',[0 0]);
ylabel('Height [m]')		xlabel('Period [s]')
		ylabel('Height [m]')
<pre>title(['Power Matrix for Damping = ' num2str(c(1)) ' [N/m/s]'])</pre>		<pre>title(['Power Matrix for Damping = ' num2str(c(1)) ' [N/m/s]'])</pre>

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. The views expressed in the article do not necessarily represent the views of the DOE or the U.S. Government. The U.S. Government retains and the publisher, by accepting the article for publication, acknowledges that the U.S. Government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this work, or allow others to do so, for U.S. Government purposes.



##