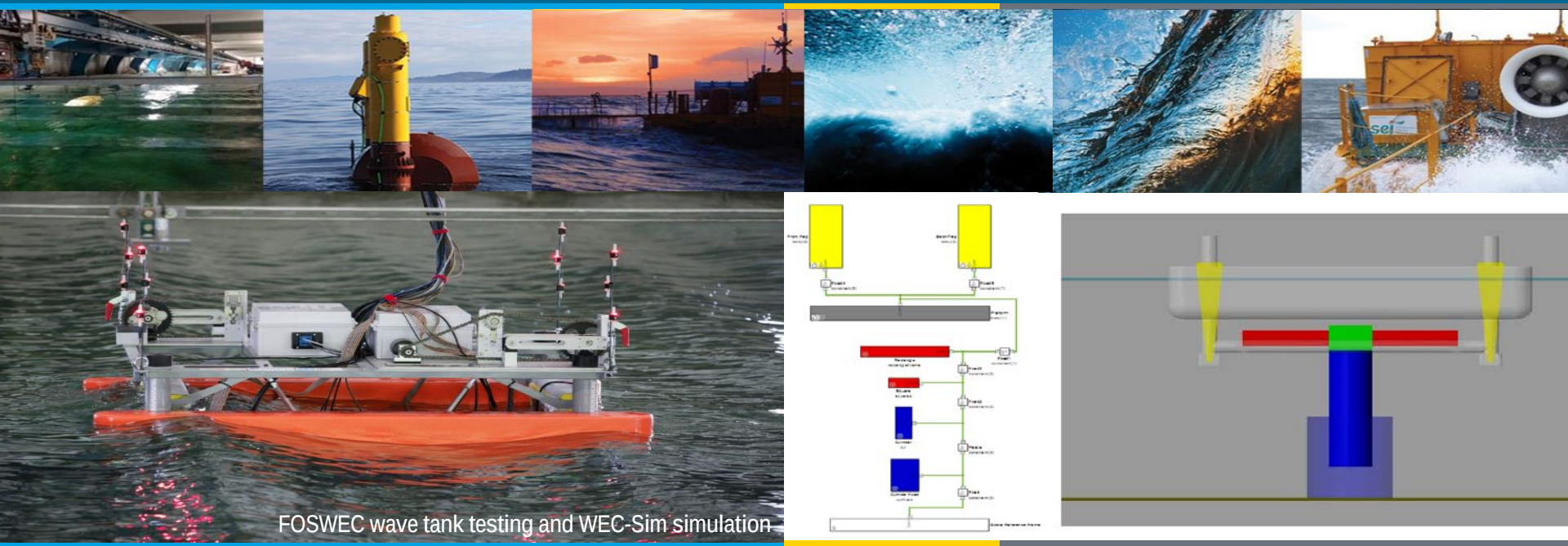
U.S. DEPARTMENT OF ENERGY | Energy Efficiency & Renewable Energy



WEC-Sim Training Course for users and developers

August 17, 2017

Yi-Hsiang Yu (NREL)
Kelley Ruehl (Sandia)

Course Agenda

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

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

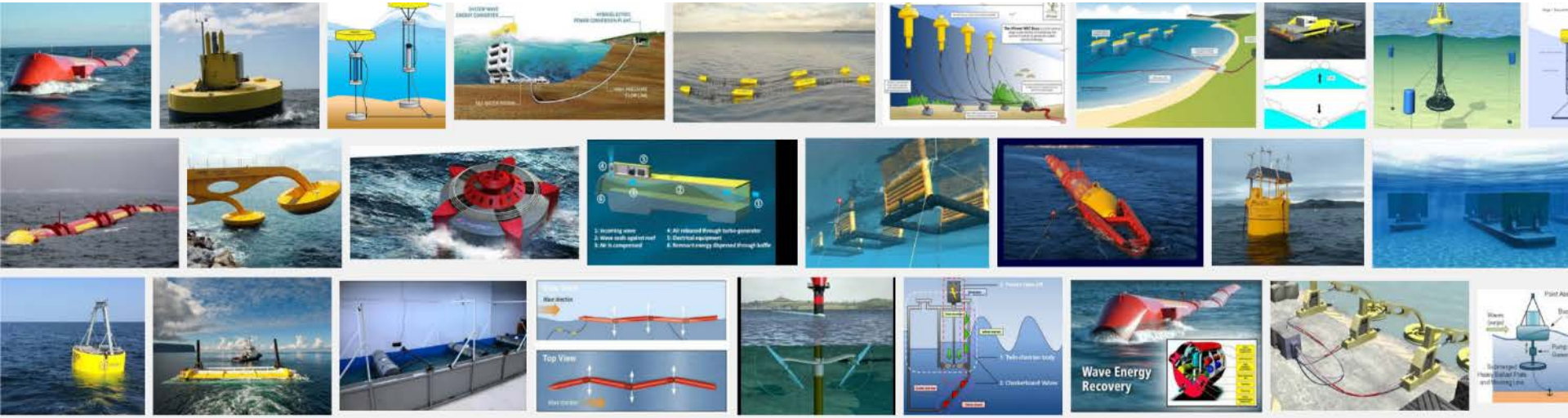
Wave Implementation Topics

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

Time (MT)	Topic (~Duration)	Description
1:00 pm	**Wave Implementation** ~30min	• Detail of what is done in the wave class and library blocks and how they are linked • Somewhat line-by-line, give overview of what type of information/calculations are done in the wave class • **Ex 1:** specify wave type ‘regular’ ‘regularCIC’ and ‘spectrumImport’ to show which method in waveClass are executed and which variant subsystems are turned on

WEC-Sim Webinar

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



Wave Implementation

Kelley Ruehl (Sandia)

Wave Class Overview

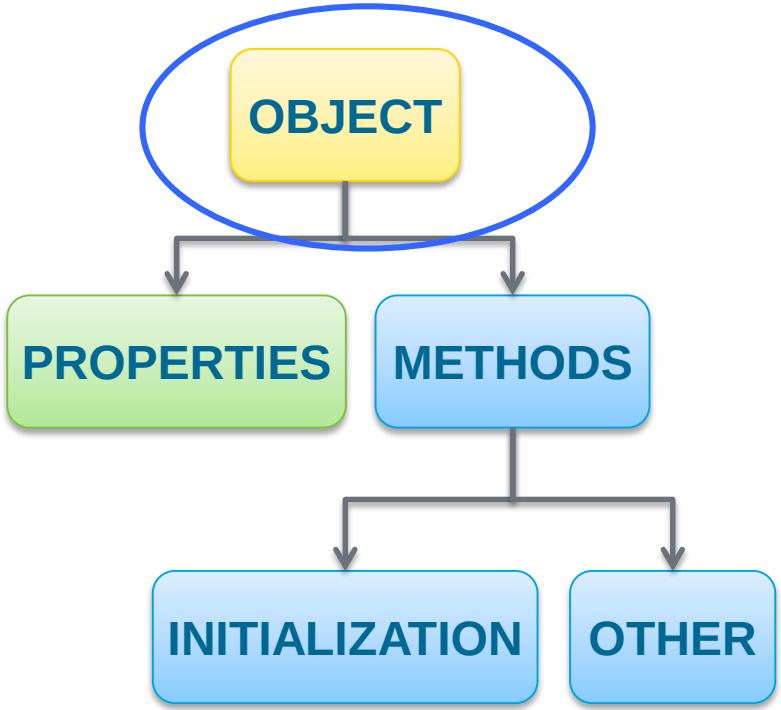
U.S. DEPARTMENT OF ENERGY | Energy Efficiency & Renewable Energy

waveClass.m

The wave class contains all wave information necessary to define the incident wave condition for the WEC-Sim time-domain simulation.

- Wave Class is initialized and defined in the WEC-Sim input file.
- ***waveClass.m*** → ***waves***
- Required Properties:
  - type
  - Each wave 'type' has different required properties

Wave Type	Required Properties
`noWave`	`waves.T`
`noWaveCIC`	N/A
`regular`	`waves.H` , `waves.T`
`regularCIC`	`waves.H` , `waves.T`
`irregular`	`waves.H` , `waves.T` , `waves.spectrumType`
`spectrumImport`	`waves.spectrumDataFile`
`etaImport`	`waves.etaDataFile`

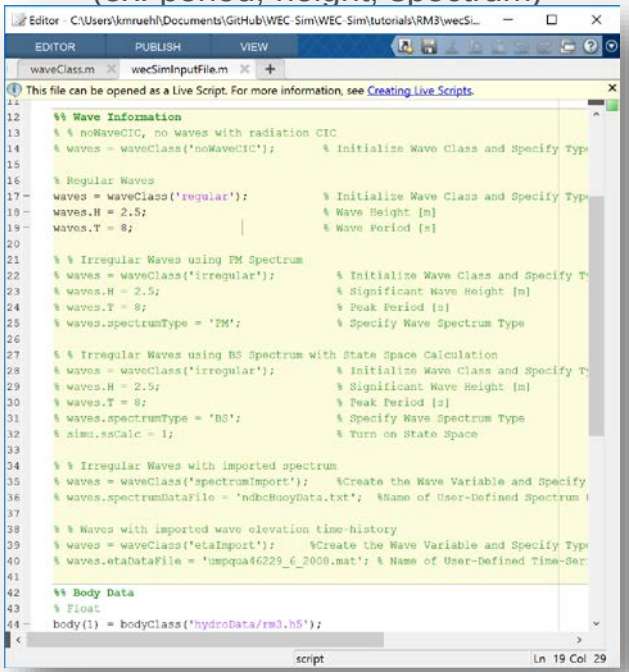


Wave Implementation

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

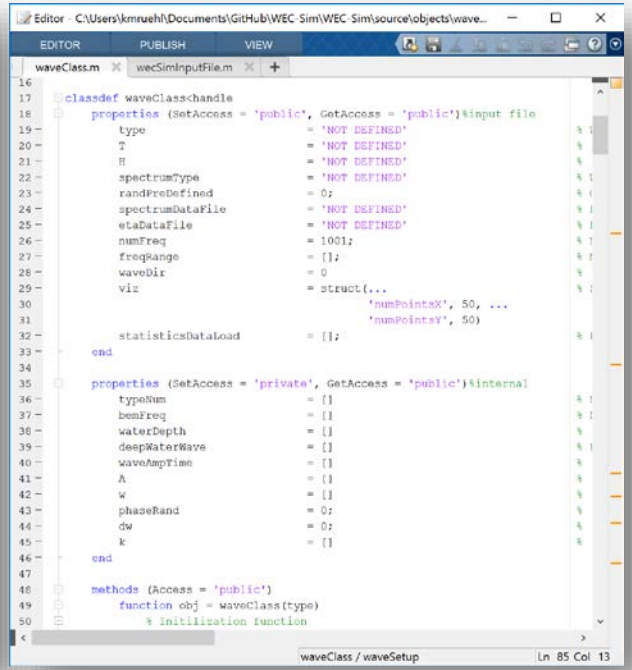
wecSimInputFile.m

Initializes wave class, user specifies wave type and properties (ex: period, height, spectrum)



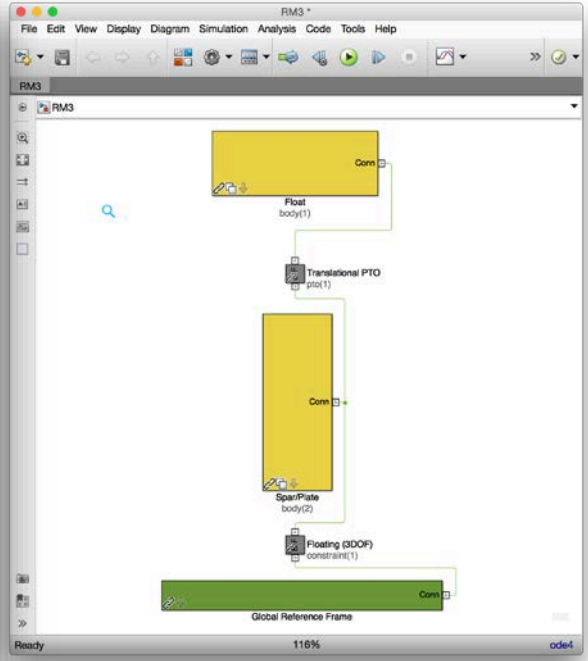
waveClass.m

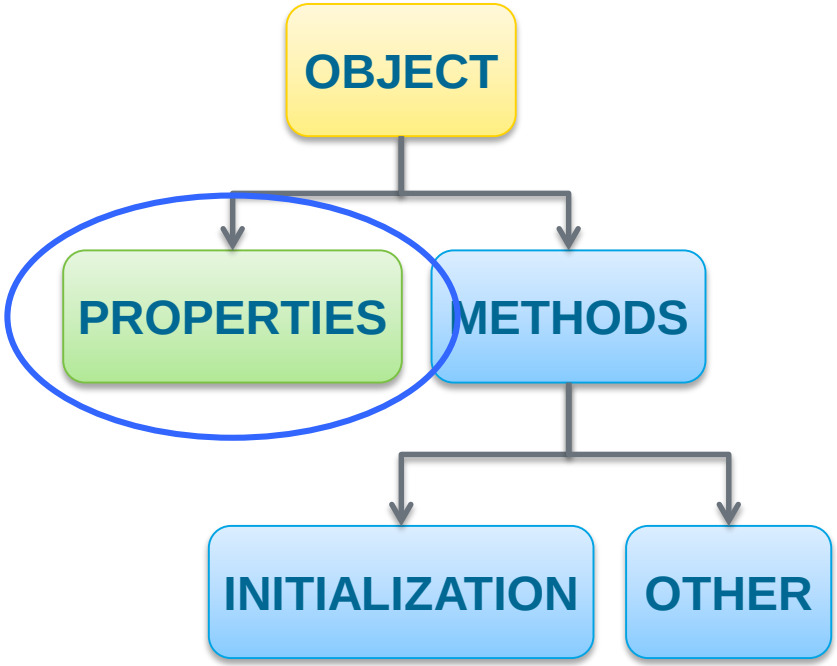
Creates 'waves' object, parses input file, turns on variant subsystems, generates incident wave for WEC-Sim



Simulink Model File

Looks the same on the top level, but has different variant subsystems active based on inputs





waveClass Properties

Wave Class Properties

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

waveClass.m

>>open waveClass

The wave class contains all wave information necessary to define the incident wave condition for the WEC-Sim time-domain simulation.

- Required Properties:
  - type
  - Each wave ‘type’ has different required properties

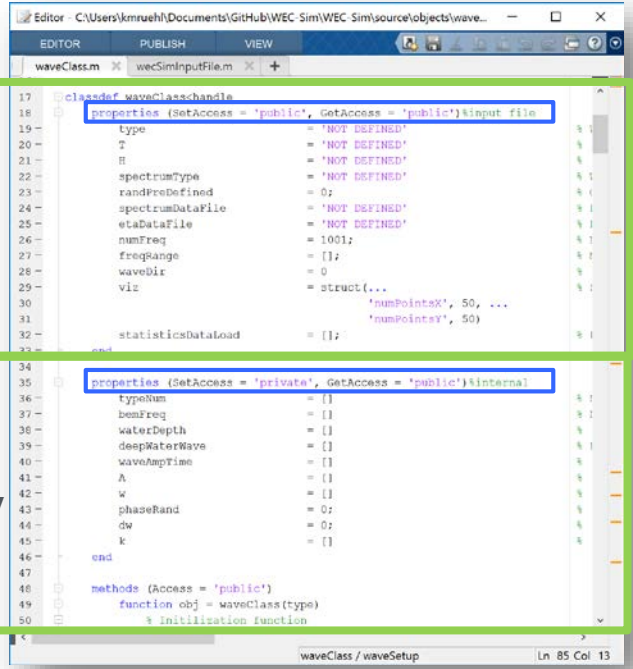
Wave Type	Required Properties
noWave	waves.T
noWaveCIC	N/A
regular	waves.H , waves.T
regularCIC	waves.H , waves.T
irregular	waves.H , waves.T , waves.spectrumType
spectrumImport	waves.spectrumDataFile
etaImport	waves.etaDataFile

waveClass Properties

Available at the beginning of waveClass.m file, can see ‘public’ versus ‘private’ properties

- **Public properties**
- **Defined in input file**

- **Private properties**
- **Calculated by methods in wave class**



Wave Class Properties

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

waveClass.m

>> wecSim

>>waves

The wave class contains all wave information necessary to define the incident wave condition for the WEC-Sim time-domain simulation.

- Required Properties:
  - type
  - Each wave 'type' has different required properties

Wave Type	Required Properties
`noWave`	`waves.T`
`noWaveCIC`	N/A
`regular`	`waves.H`, `waves.T`
`regularCIC`	`waves.H`, `waves.T`
`irregular`	`waves.H`, `waves.T`, `waves.spectrumType`
`spectrumImport`	`waves.spectrumDataFile`
`etaImport`	`waves.etaDataFile`

Wave Object Properties

Created by waveClass, available upon completion of WEC-Sim run

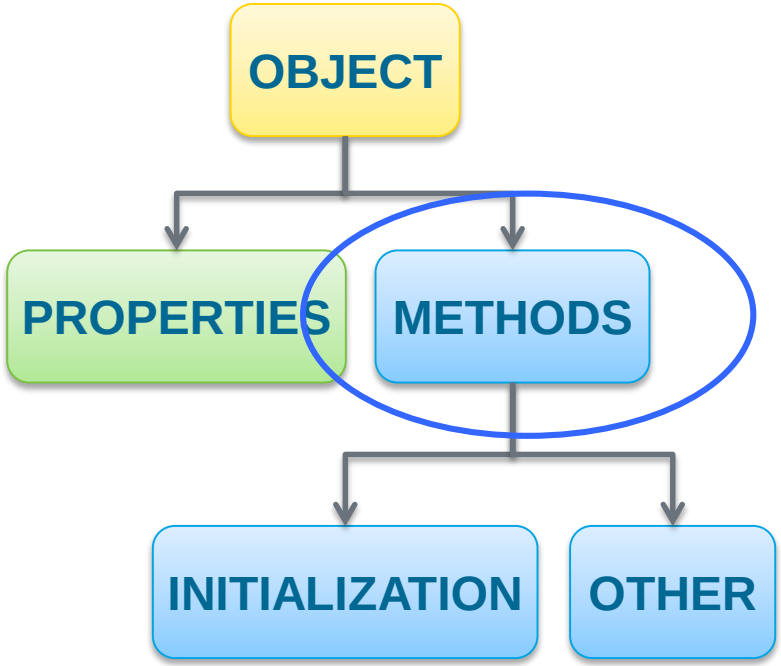
Command Window

```
>> waves

waves =

  waveClass with properties:

                 type: 'regular'
                    T: 8
                    H: 2.5000
          spectrumType: 'NOT DEFINED'
        randPreDefined: 0
      spectrumDataFile: 'NOT DEFINED'
           etaDataFile: 'NOT DEFINED'
               numFreq: 1001
             freqRange: []
               waveDir: 0
                   viz: [1×1 struct]
    statisticsDataLoad: []
               typeNum: 10
               bemFreq: [1×260 double]
            waterDepth: 200
          deepWaterWave: 1
            waveAmpTime: [4001×2 double]
                      A: 1.2500
                      w: 0.7854
             phaseRand: 0
                     dw: 0
                      k: 0.0629
```



waveClass Methods

Wave Class Methods

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

waveClass

Creates waves object, reads wave 'type' from WEC-Sim input file

Editor - C:\Users\kmruehl\Documents\GitHub\WEC-Sim\WEC-Sim\source\objects\wave...

EDITOR PUBLISH VIEW

wecSimInputFile.m | waveClass.m

```
47
48      methods (Access = 'public')
49          function obj = waveClass(type)
50              % Initilization function
51              % Set wave type and type number
52              obj.type = type;
53              switch obj.type
54                  case {'noWave'}          % No Waves with Constant Hydrodynamic C
55                      obj.typeNum = 0;
56                  case {'noWaveCIC'}       % No Waves w/Convolution Integral Calcu
57                      obj.typeNum = 1;
58                  case {'regular'}         % Regular Waves with Constant Hydrodynar
59                      obj.typeNum = 10;
60                  case {'regularCIC'}      % Regular Waves w/Convolution Integral (
61                      obj.typeNum = 11;
62                  case 'irregular'         % Irregular Waves with 'PM', BS' or 'JS
63                      obj.typeNum = 20;
64                  case 'spectrumImport' % Irregular waves with imported wave spe
65                      obj.typeNum = 21;
66                  case 'etaImport'      % Waves with imported wave elevation time
67                      obj.typeNum = 30;
68              end
69          end
70
71          function plotEta(obj)
72              % Plot wave elevation time-history
73              figure
74              plot(obj.waveAmpTime(:,1),obj.waveAmpTime(:,2))
75              xlabel('Time (s)')
76              ylabel('Eta (m)')
77          end
78
79          function waveSetup(obj,bemFreq,wDepth,rampTime,dt,maxIt,g,endTime)
80              % Calculate and set wave properties based on wave type
81              obj.bemFreq    = bemFreq;
```

waveClass / waveClass Ln 64 Col 28

waveSetup

Specifies wave parameters based on wave 'type'

Editor - C:\Users\kmruehl\Documents\GitHub\WEC-Sim\WEC-Sim\source\objects\wave...

EDITOR PUBLISH VIEW

wecSimInputFile.m | waveClass.m

```
78
79          function waveSetup(obj,bemFreq,wDepth,rampTime,dt,maxIt,g,endTime)
80              % Calculate and set wave properties based on wave type
81              obj.bemFreq    = bemFreq;
82              obj.setWaveProps(wDepth)
83              switch obj.type
84                  case {'noWave','noWaveCIC'}
85                      if isempty(obj.w)
86                          obj.w = 2*pi/obj.T;
87                      end
88                      obj.A = obj.H/2;
89                      obj.waveElevNowave(maxIt,dt);
90                  case {'regular','regularCIC'}
91                      if isempty(obj.w)
92                          obj.w = 2*pi/obj.T;
93                      end
94                      obj.A = obj.H/2;
95                      obj.waveElevReg(rampTime, dt, maxIt);
96                  case {'irregular','spectrumImport'}
97                      WFQSt=min(bemFreq);
98                      WFQEd=max(bemFreq);
99                      if ~isempty(obj.freqRange)
100                         if obj.freqRange(1) > WFQSt && obj.freqRange(1) > 0
101                             WFQSt = obj.freqRange(1);
102                         else
103                             warning('Min frequency range outside BEM data, min
104                         end
105                         if obj.freqRange(2) < WFQEd && obj.freqRange(2) > WFQS
106                             WFQEd = obj.freqRange(2);
107                         else
108                             warning('Max frequency range outside BEM data, max
109                         end
110                     end
111                     obj.dw=(WFQEd-WFQSt)/(obj.numFreq-1);
```

waveClass / waveClass Ln 62 Col 16

Other Methods…

irregWaveSpectrum

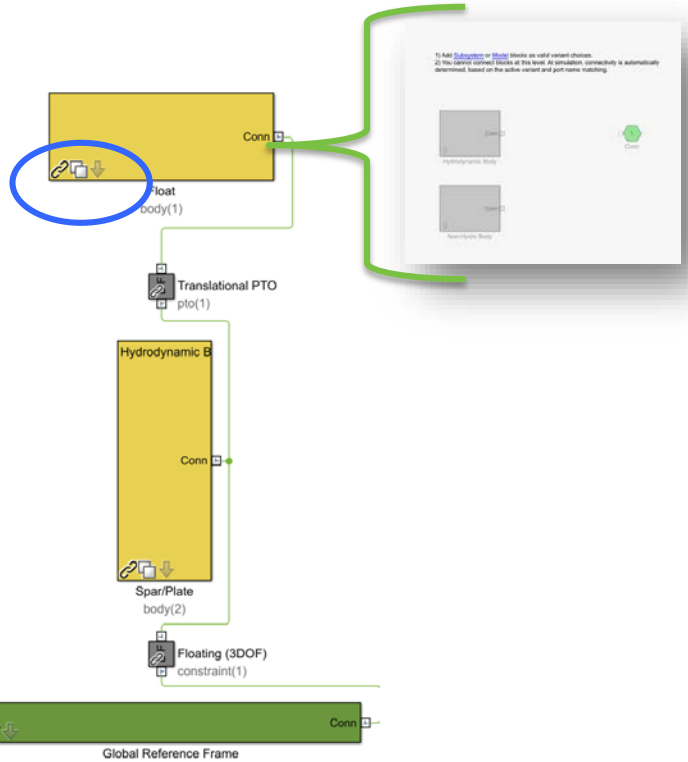
- Generates 'PM', 'BS', 'JS' wave spectrum based on user inputs

waveElevReg & *waveElevIrreg*

- Generate wave surface elevation time-series with wave ramp

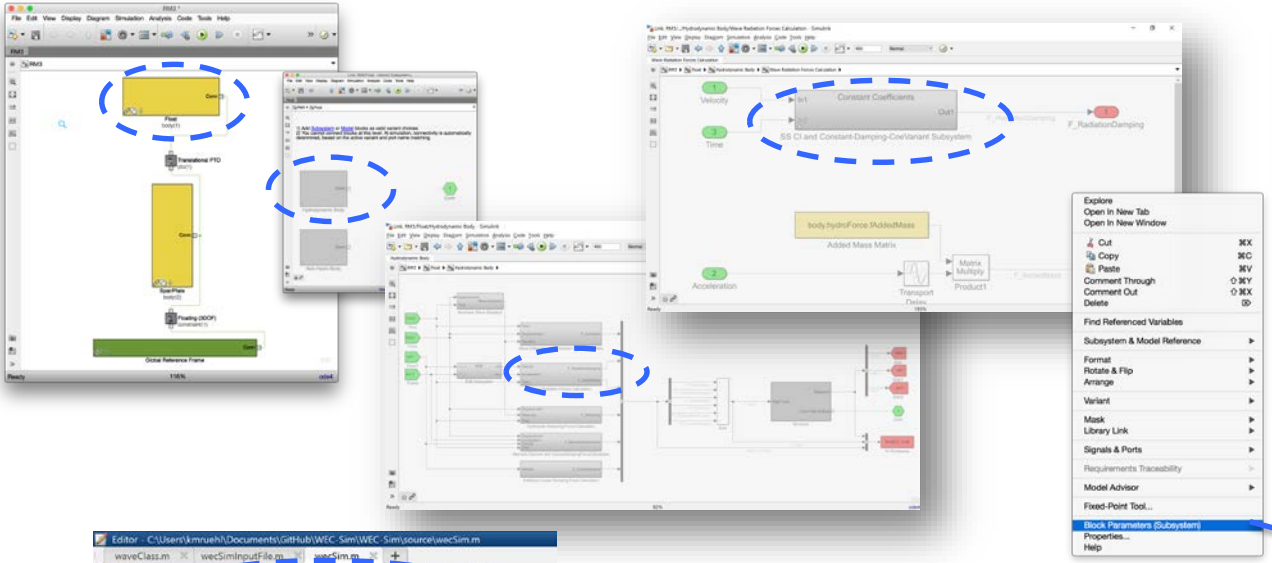
write_paraview_vtp

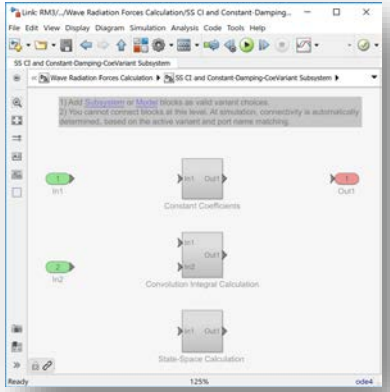
- Function for generating ParaView Visualization of wave surface elevation

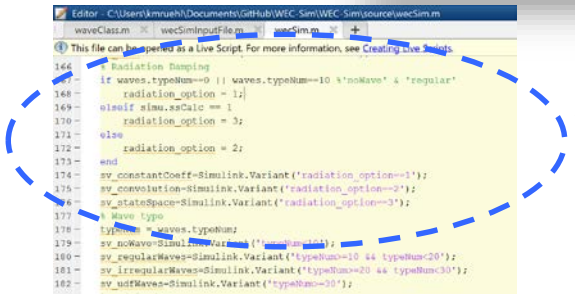


Wave Class Variant Subsystems

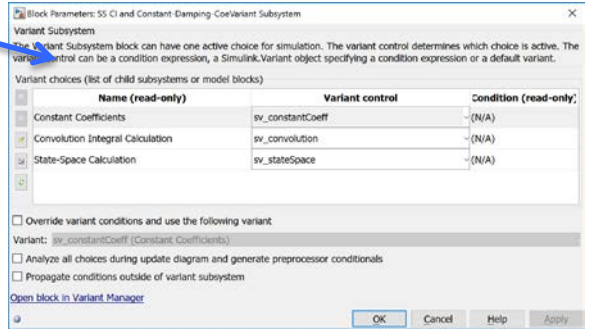
Variant Subsystem – Radiation Calc







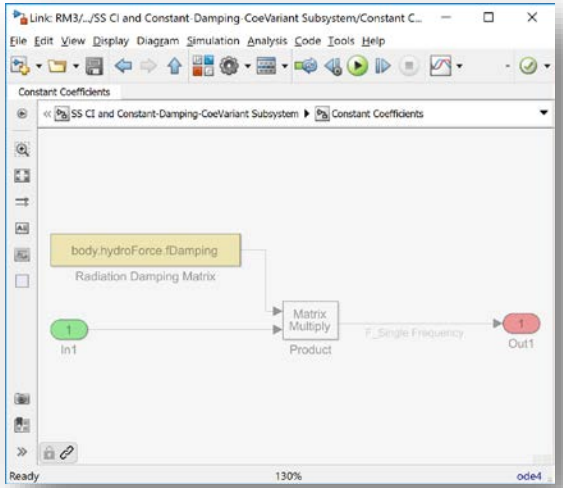
Initialization of variant subsystem by *wecSimInputFile.m* in *waveClass.m* and *wecSim.m*



Radiation Force Block

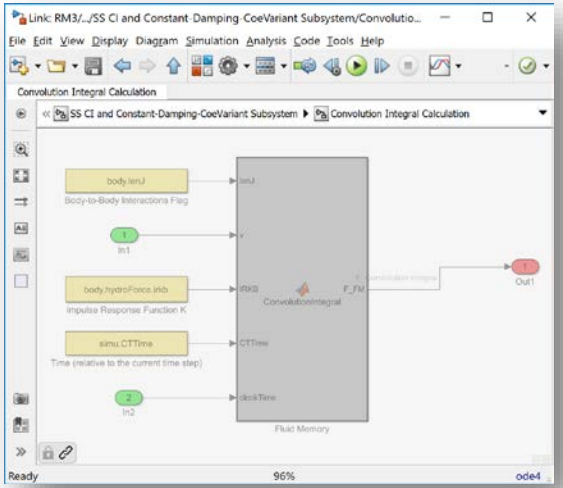
U.S. DEPARTMENT OF ENERGY | Energy Efficiency & Renewable Energy

Constant Coefficient Block



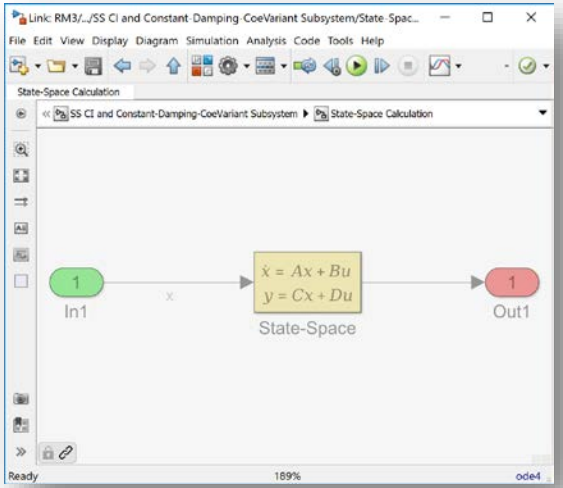
- **'noWave'**
- **'regular'**

Convolution Integral Block



- **'noWaveCIC'**
- **'regularCIC'**
- **'irregular'**
- **'spectrumImport'**
- **'etaImport'**

State Space Block



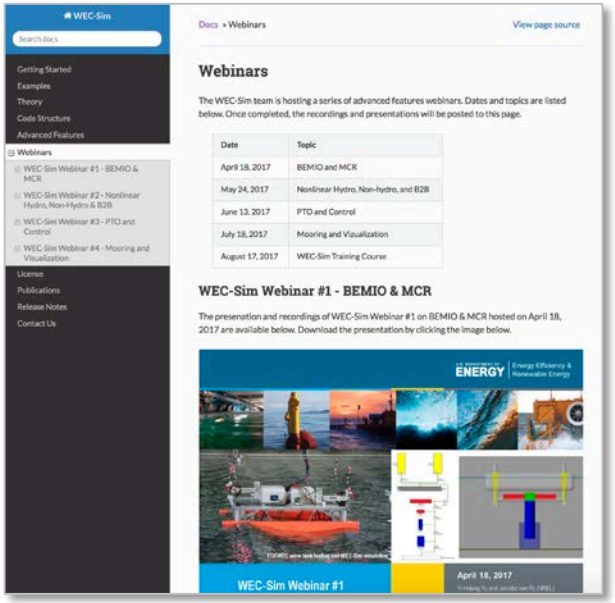
- **'simu.ss'**
- (Except for noWave or regular)

Wave Implementation Examples…

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.