WEC-Sim Training Course

Online Training Materials

PRESENTED BY

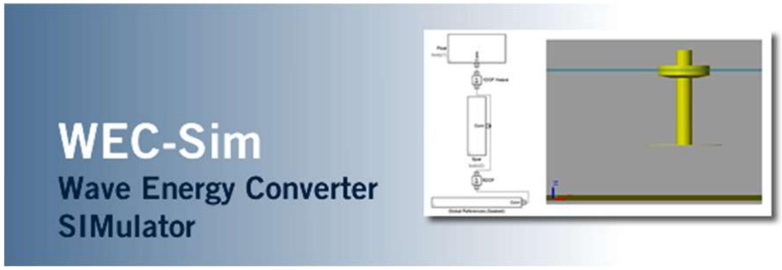
Nathan Tom, NREL

Theory & Workflow

What is WEC-Sim?

WEC-Sim (Wave Energy Converter Simulator)

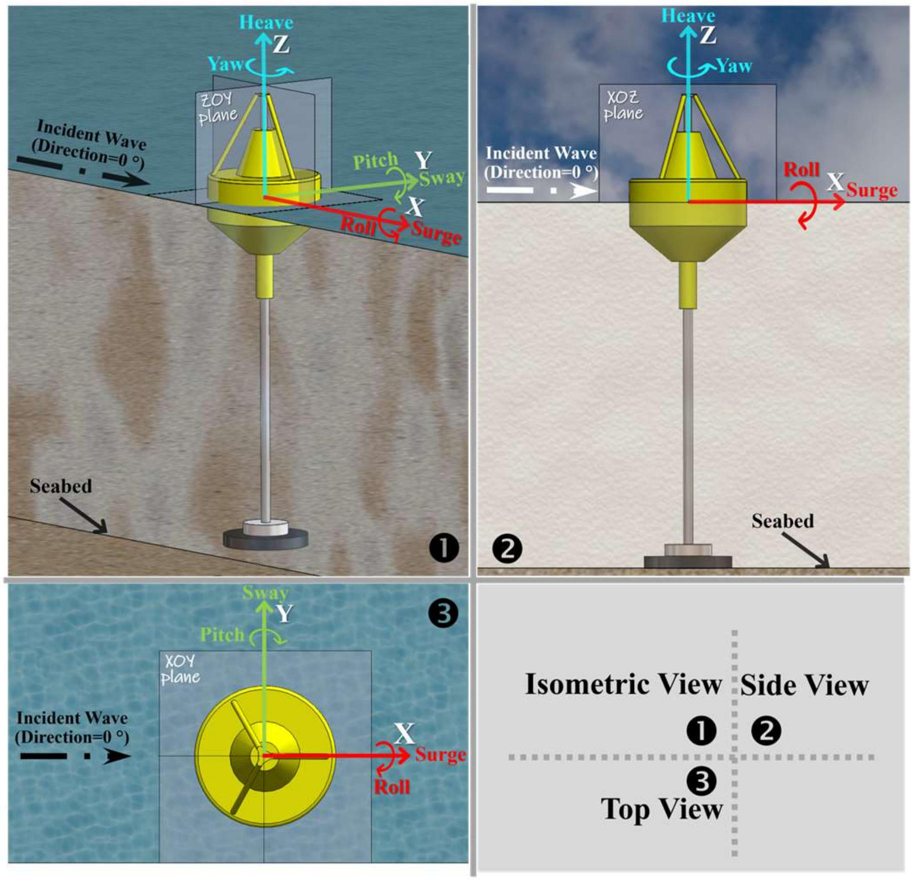
- Simulates wave energy converter dynamics in operational waves
- Time-domain rigid body equation of motion solver based on Cummins' formulation
- Open source software developed in MATLAB/SIMULINK
  - Available at https://github.com/WEC-Sim/WEC-Sim
- Joint NREL/Sandia project funded by the US Department of Energy
- First Release: v1.0 in June 2014
- Current Release: v5.0.1 in Sept 2022



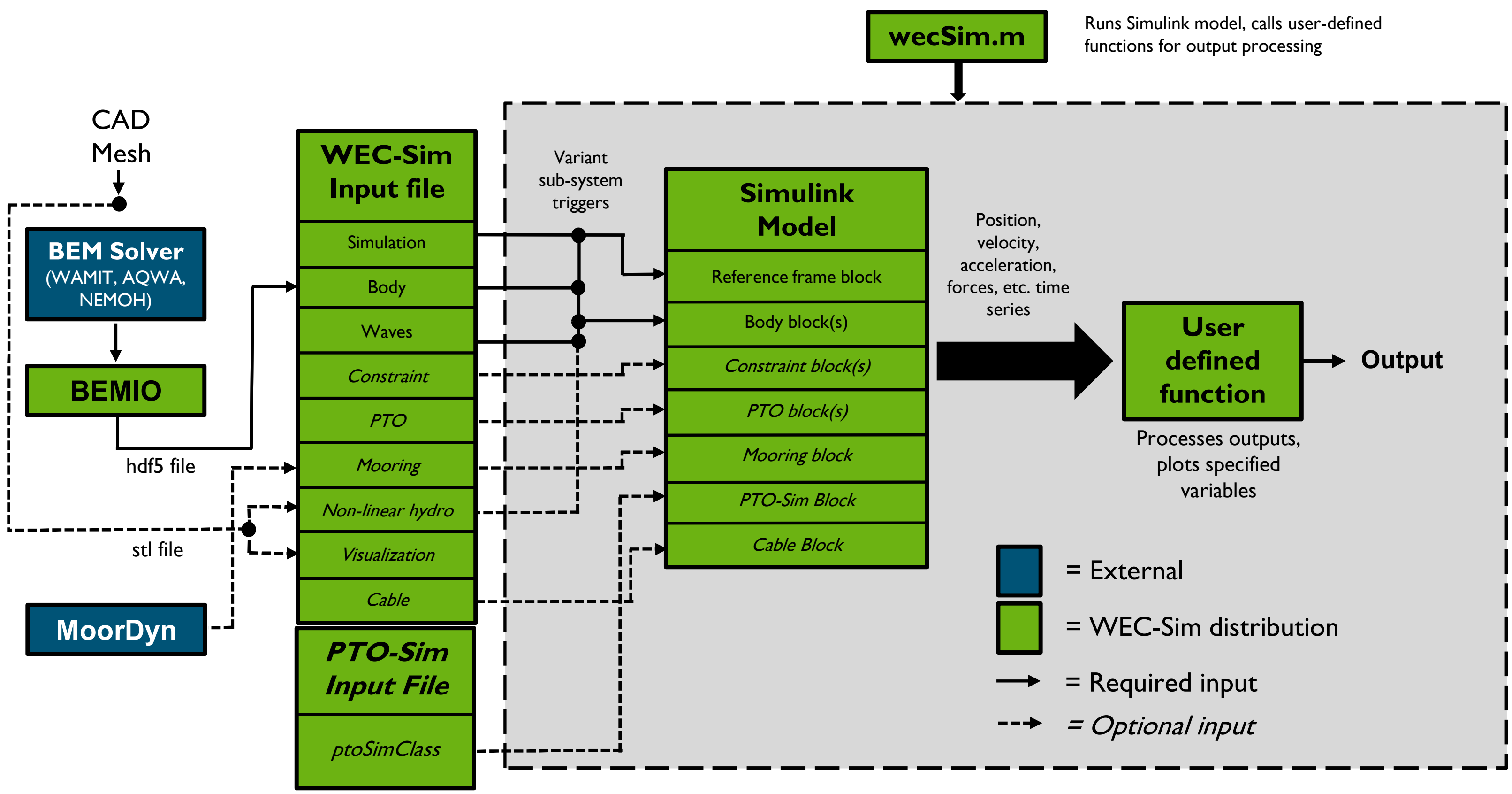
Coordinate System

- 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).
- Z-Axis is in the vertical upwards direction from a zero at the still water level, and the Y-Axis direction is defined by the right-hand rule.
- Position is described in a 6-element vector X. This convention is maintained for velocities, forces, etc.

Index	1	2	3	4	5	6
Position *X*	x (surge)	y (sway)	z (heave)	rx (roll)	ry (pitch)	rz (yaw)

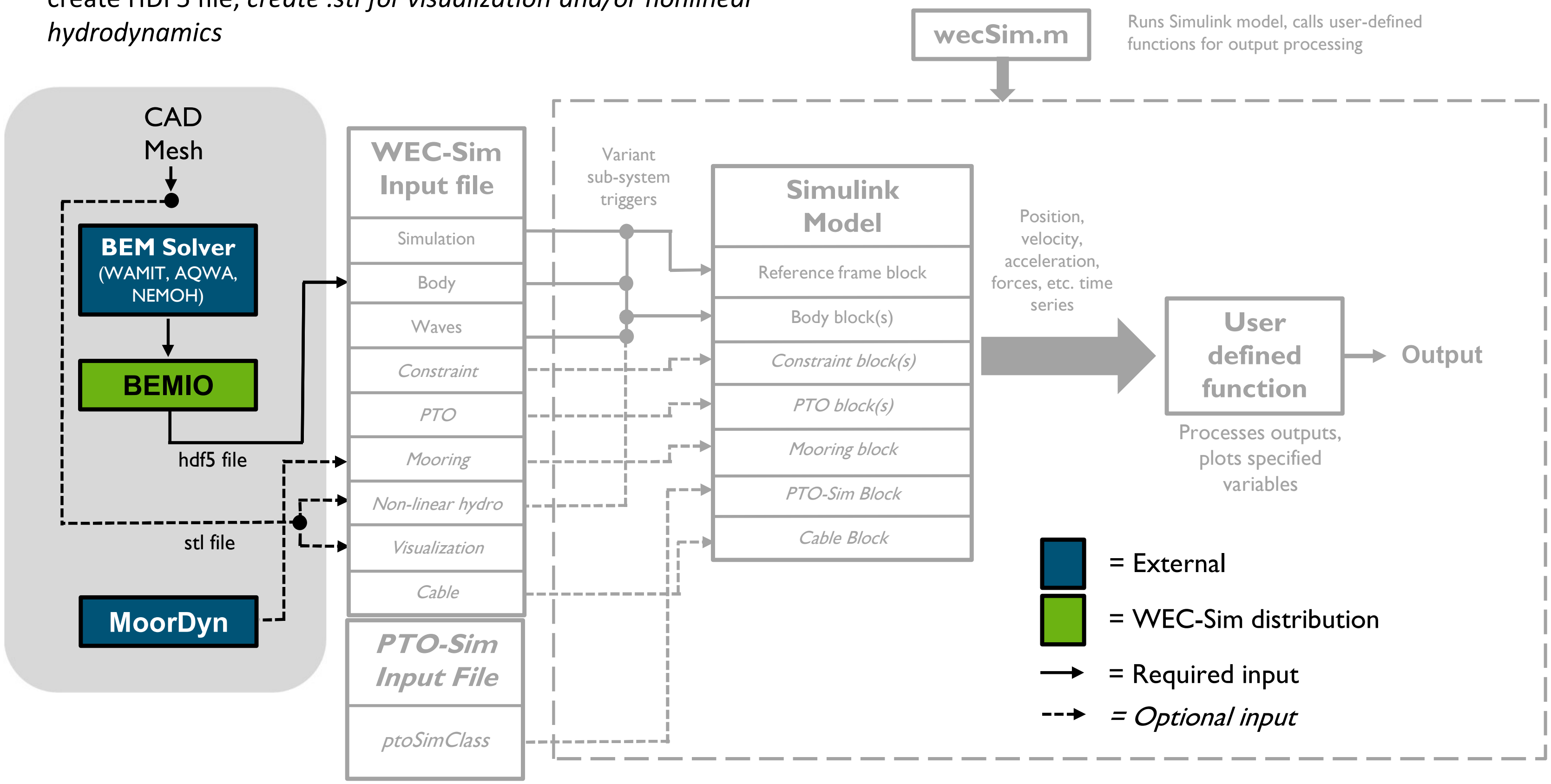


WEC-Sim Workflow



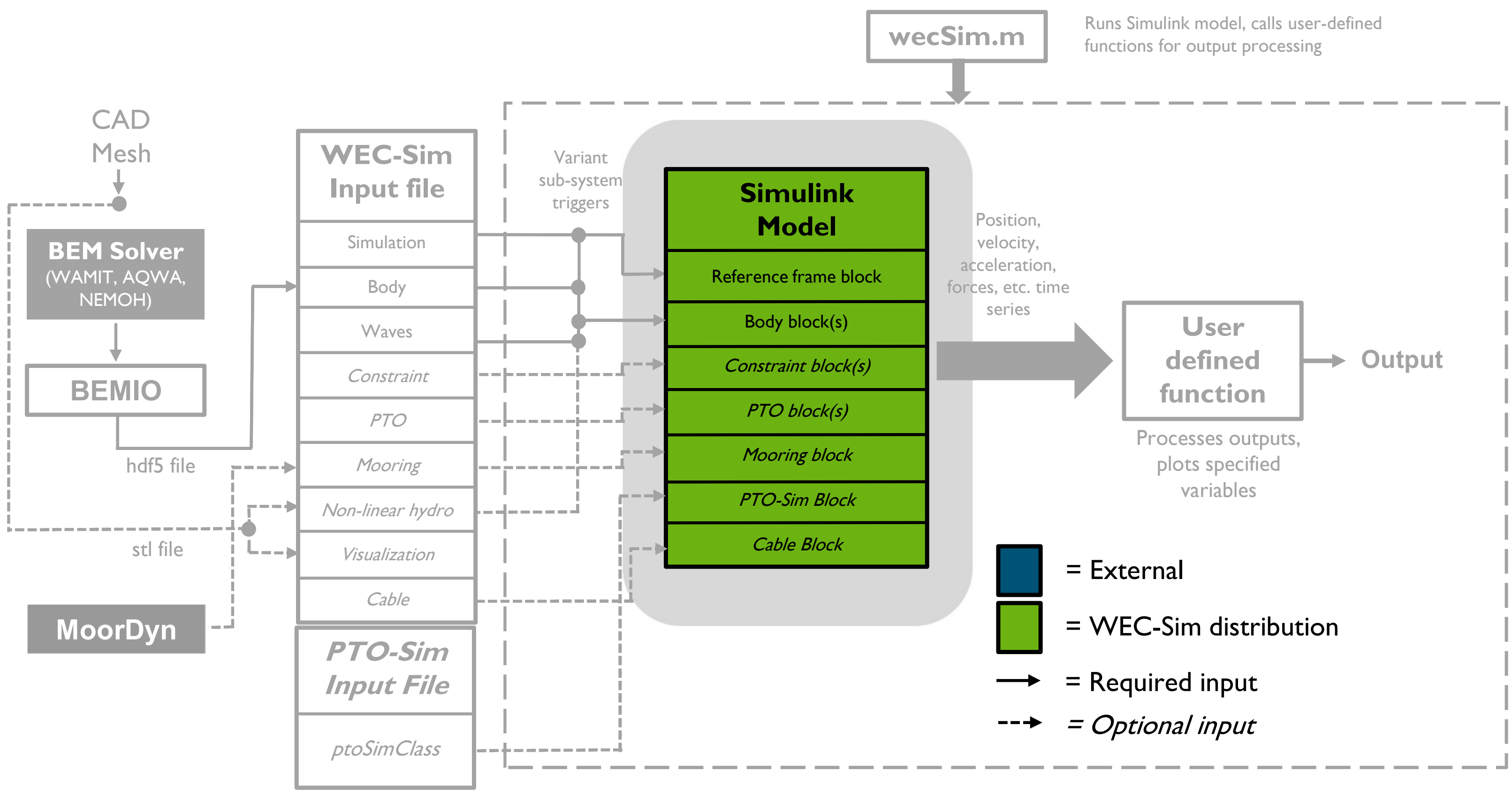
WEC-Sim Workflow

1). Generate 3-D mesh, calculate hydrodynamic coefficients, create HDF5 file, *create .stl for visualization and/or nonlinear hydrodynamics*



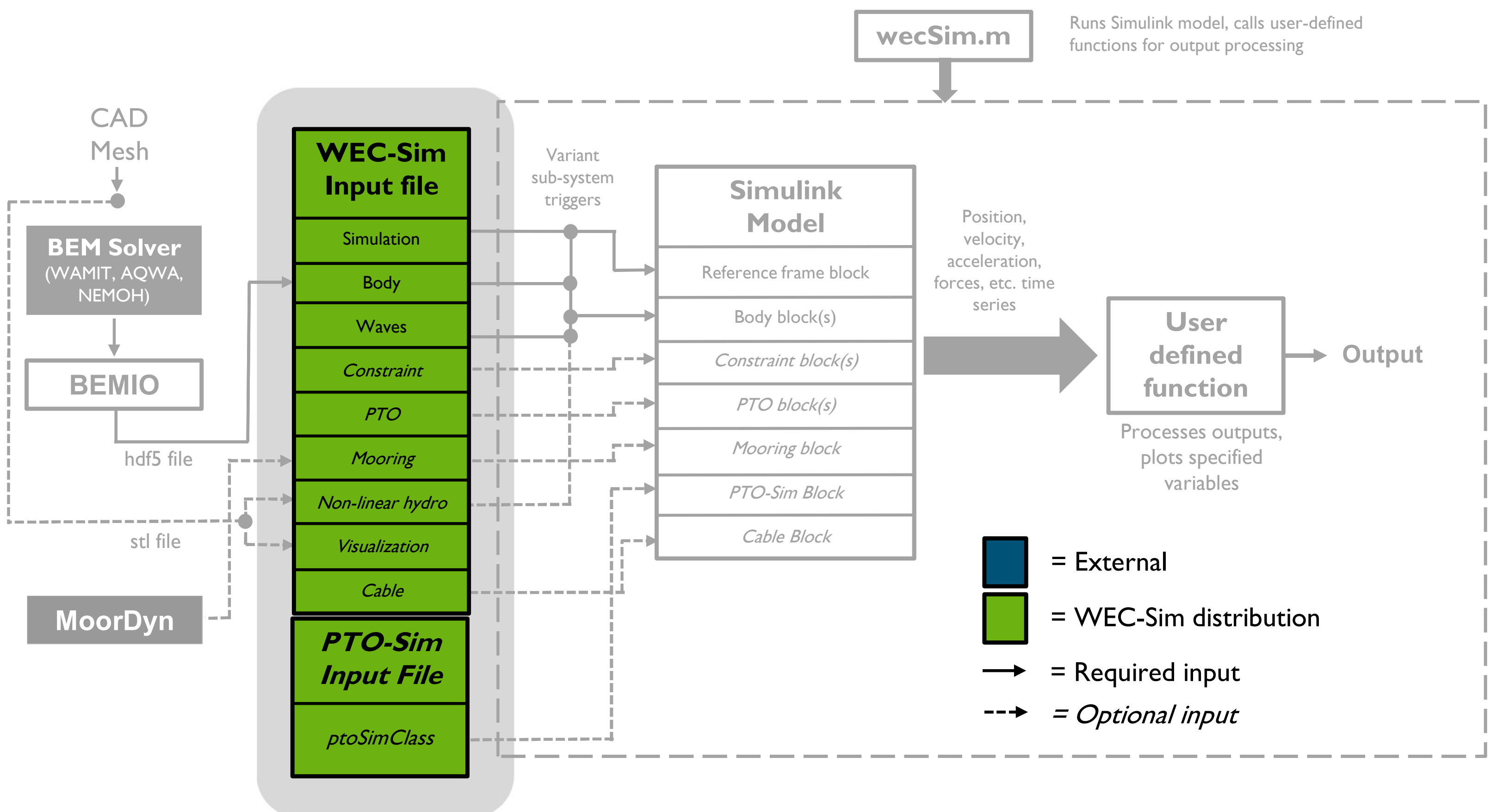
WEC-Sim Workflow

2). Build WEC-Sim model in Simulink



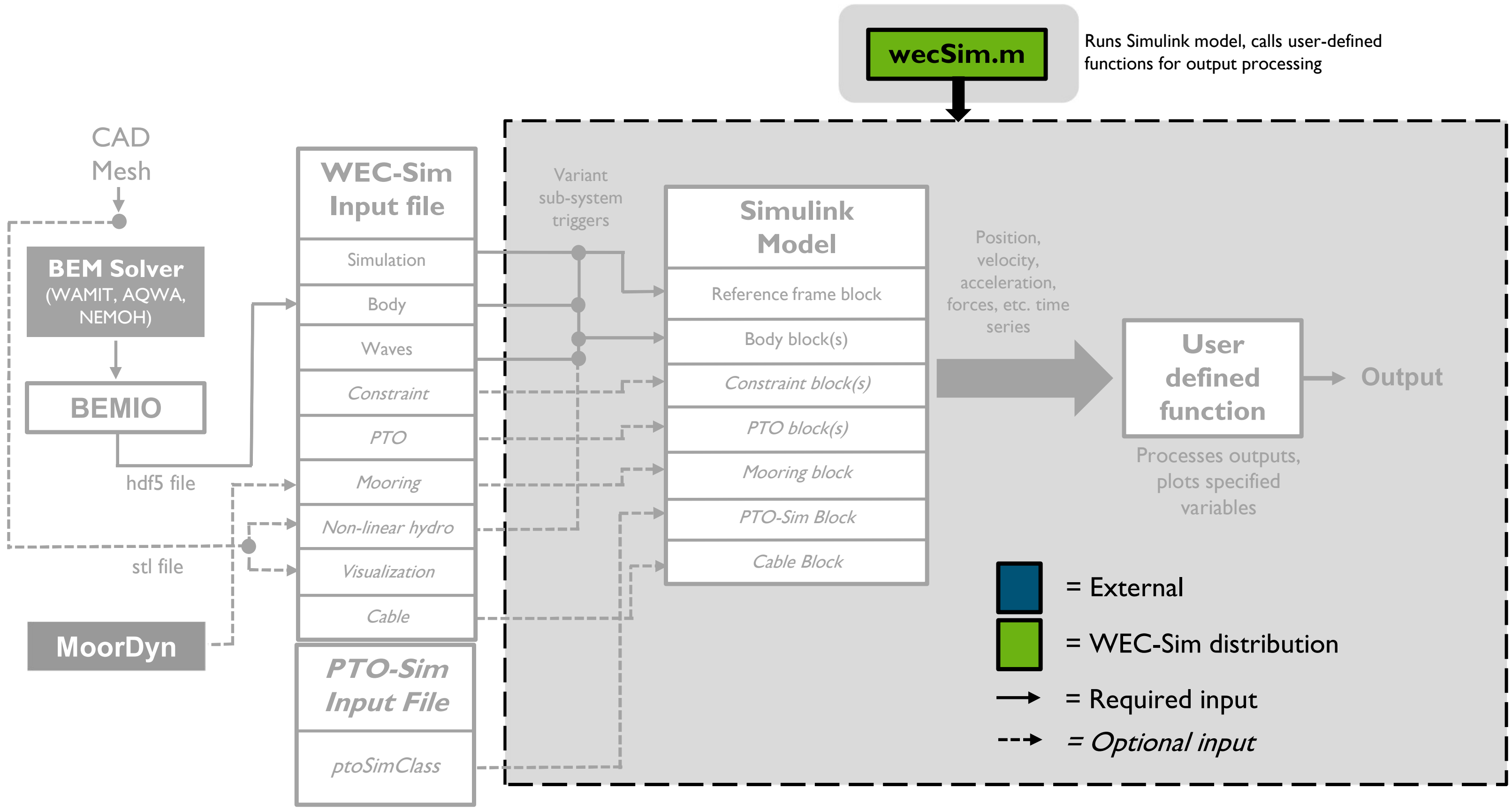
WEC-Sim Workflow

3). Write WEC-Sim input file



WEC-Sim Workflow

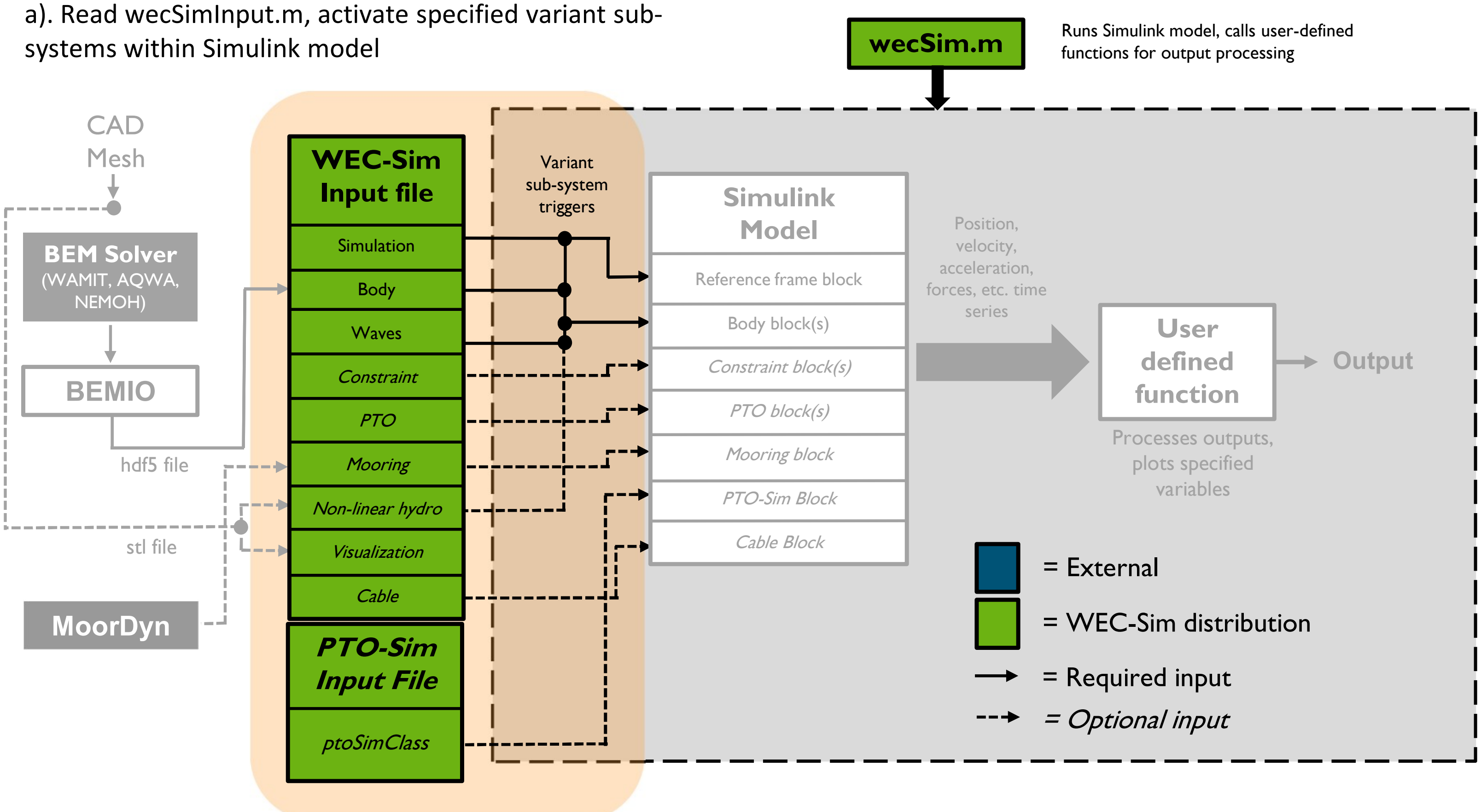
4). Execute wecSim.m



WEC-Sim Workflow

4). Execute wecSim.m

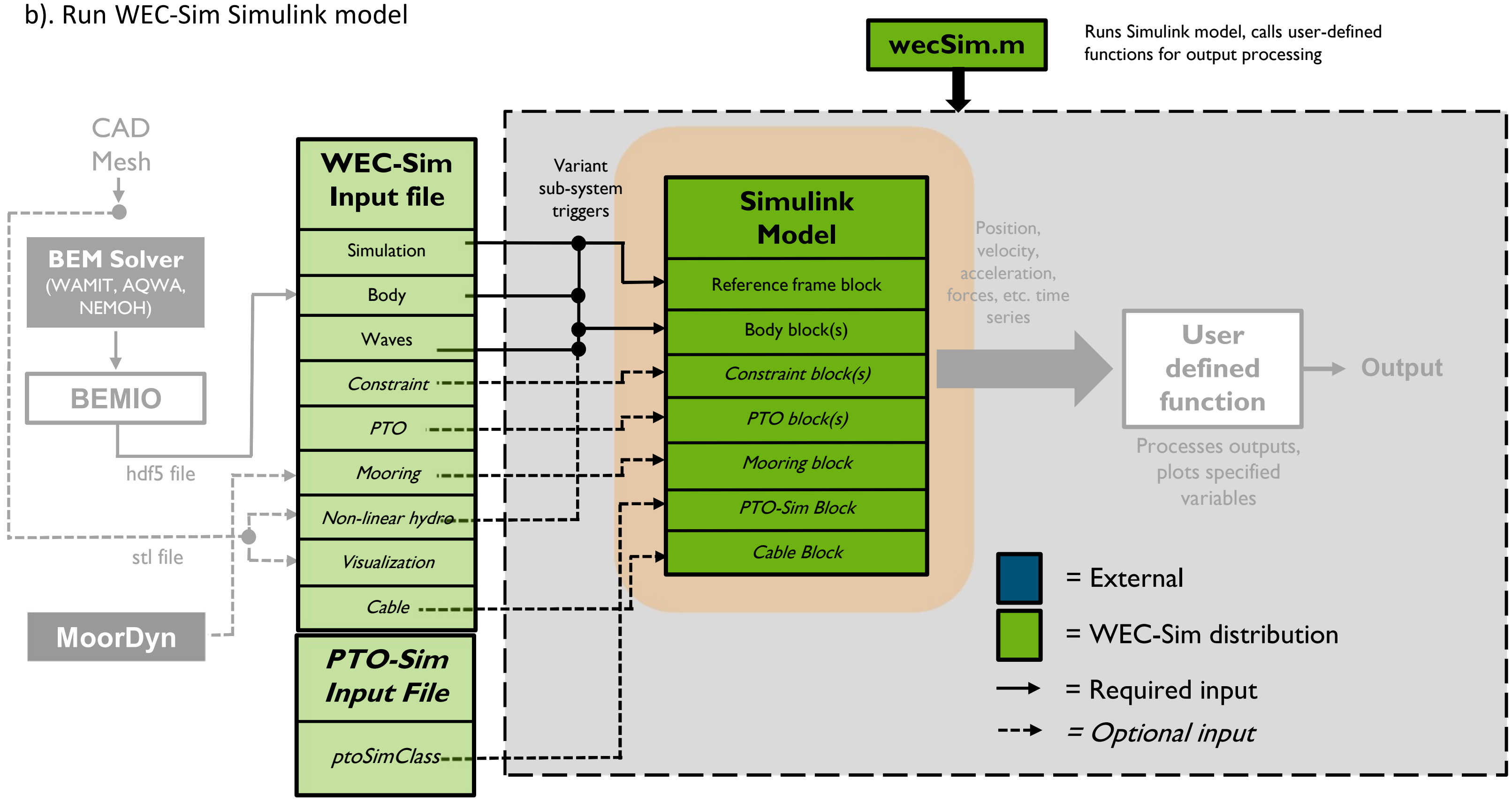
a). Read wecSimInput.m, activate specified variant sub-systems within Simulink model



WEC-Sim Workflow

4). Execute wecSim.m

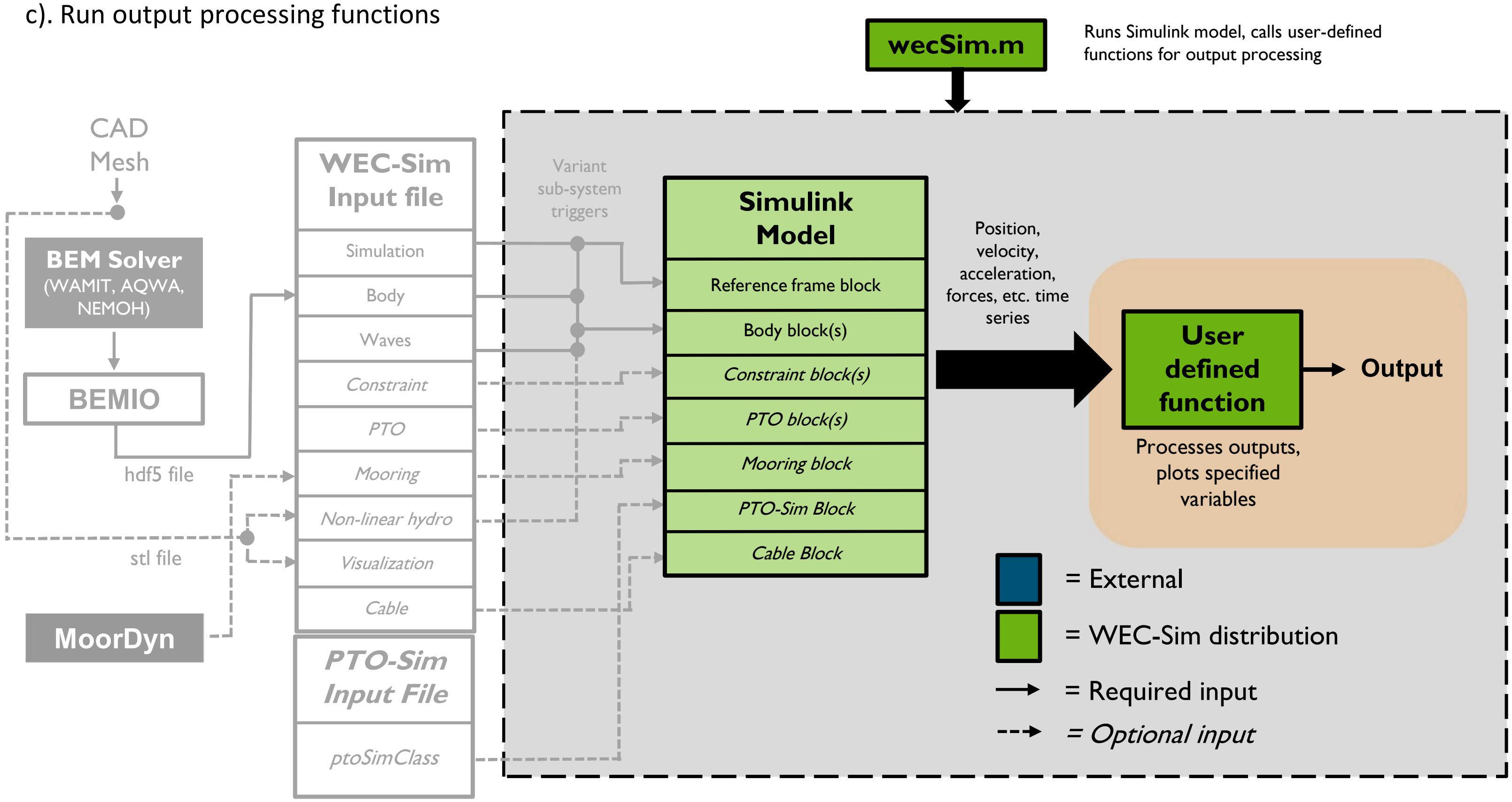
b). Run WEC-Sim Simulink model



WEC-Sim Workflow

4). Execute wecSim.m

c). Run output processing functions



General Equations of Motion

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

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

Hydrostatic restoring force

Wave excitation force (from BEM simulations)

Radiation forces of added mass and wave damping (from BEM simulations)

Viscous force

Power take-off force

Mooring force

Nonlinear hydrodynamic force

- Excitation and radiation forces are determined from hydrodynamic coefficients calculated from Boundary Element Method (BEM)

Position X	**x (surge)**	**y (sway)**	**z (heave)**	**rx (roll)**	**ry (pitch)**	**rz (yaw)**

Static Mass

$$m\ddot{x}(t) = F_{hs}(t) + F_{ext}(t) + F_{rad}(t) + F_v(t) + F_{PTO}(t) + F_m(t) + F_{nh}(t)$$

- Static mass, or dry mass, is the mass (kg) (1-element) and inertia (3-element)* (kg-m2) of the dry WEC body.
  - In WEC-Sim v5.0.1 there is also the option to define a 3-element product of inertia.
- Specified for each body* in the wecSimInputFile.m as part of the bodyClass definition
- See Training Materials – Body Implementation for details
  - Usually: see Advanced Features → Body Features for special cases

```
%% 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];
```

*The definition of body mass and inertia properties in the wecSimInputFile for the RM3 example. In this special 'equilibrium' case, the mass is set equal to the mass of the displaced volume of water, defined in the *.h5 file.*

Hydrostatic Forces

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

- Hydrostatic restoring force is calculated as the product of a hydrostatic stiffness matrix and a vector of displacement*

$$F_{HS}(t) = K_{hs}X(t) = \begin{bmatrix} K_{1,1} & K_{1,2} & K_{1,3} & K_{1,4} & K_{1,5} & K_{1,6} \\ K_{2,1} & K_{2,2} & K_{2,3} & K_{2,4} & K_{2,5} & K_{2,6} \\ K_{3,1} & K_{3,2} & K_{3,3} & K_{3,4} & K_{3,5} & K_{3,6} \\ K_{4,1} & K_{4,2} & K_{4,3} & K_{4,4} & K_{4,5} & K_{4,6} \\ K_{5,1} & K_{5,2} & K_{5,3} & K_{5,4} & K_{5,5} & K_{5,6} \\ K_{6,1} & K_{6,2} & K_{6,3} & K_{6,4} & K_{6,5} & K_{6,6} \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \\ x_5(t) \\ x_6(t) \end{bmatrix} = \begin{bmatrix} F_{HS,1}(t) \\ F_{HS,2}(t) \\ F_{HS,3}(t) \\ F_{HS,4}(t) \\ F_{HS,5}(t) \\ F_{HS,6}(t) \end{bmatrix}$$

- Elements of K_{HS} are defined for each body in its *.h5 file with BEM output information, specifying the *.h5 file is all that is needed in wecSimInputFile
- *By default: see Advanced Features → Non-linear hydrodynamics for alternative calculation method

Position X	x (surge)	y (sway)	z (heave)	rx (roll)	ry (pitch)	rz (yaw)

Wave Excitation Forces

$$m\ddot{x}(t) = F_{hs}(t) + F_{ext}(t) + F_{rad}(t) + F_v(t) + F_{PTO}(t) + F_m(t) + F_{nh}(t)$$

- The BEM result provides complex excitation coefficients $f_{ext}(\omega, \theta)$. For a single frequency regular wave of height H, frequency ω, direction θ, ramp function $R_f(t)$, and the real component $\Re$:

$$F_{ext}(t) = \Re\left[R_f(t)\frac{H}{2}f_{ext}(\omega,\theta)e^{i\omega}\ \right] = R_f(t)\frac{H}{2}[\Re\{f_{ext}(\omega,\theta)\}\cos\omega t - \Im\{f_{ext}(\omega,\theta)\}\sin\omega t]$$

- For j frequencies with amplitude spectral density S at phase φ:

$$F_{ext}(t) = \Re\left[R_f(t)\sum_{j=1}^{N} f_{ex}\left(\omega_j,\theta\right)e^{i(\omega_j t+\varphi_j)}\sqrt{2S(\omega_j)d\omega_j}\right]$$

Position *X*	x (surge)	y (sway)	z (heave)	rx (roll)	ry (pitch)	rz (yaw)

Wave Excitation Forces

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

- For a wave defined as a time-series, the convolution of wave elevation $\eta(t)$ and the excitation impulse response function f_e calculated from f_{ext} gives an equivalent results:

$$F_{ext}(t) = R_f \int_{-\infty}^{\infty} f_e(t-\tau)\eta(\tau)d\tau$$

- The wave excitation coefficients are read from the *.h5 file.

- R_f is a ramp function that gradually increases the wave excitation from zero to the full value over a defined time period to help with simulation stability.

Position X	x (surge)	y (sway)	z (heave)	rx (roll)	ry (pitch)	rz (yaw)

Radiation Forces

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

- The BEM result provides complex frequency dependent radiation coefficients for added mass A and wave damping B.
- For a single frequency regular wave of height H and frequency ω:

$$F_{rad}(t) = -A(\omega)\ddot{X}(t) - B(\omega)\dot{X}(\mathrm{t})$$

- For a wave of multiple frequencies, the infinite frequency added mass $A_{\infty}$is used with the radiation impulse response* function K_r calculated from B:

$$F_{rad}(t) = -A_{\infty}\ddot{X} - \int_{0}^{t} K_r(t-\tau)\dot{X}(\tau)d\tau$$

- ***WEC-Sim can also approximate this integral via state-space approximation, see Theory → Numerical Methods → State Space**

Position X	x (surge)	y (sway)	z (heave)	rx (roll)	ry (pitch)	rz (yaw)

Viscous Forces

$$m\ddot{x}(t) = F_{hs}(t) + F_{ext}(t) + F_{rad}(t) + F_v(t) + F_{PTO}(t) + F_m(t) + F_{nh}(t)$$

- Linear and quadratic viscous forces are calculated from coefficients and parameters provided in the wecSimInputFile.m.

$$F_{v,quad}(t) = -\frac{1}{2}\rho A C_d \dot{X}(t)\left|\dot{X}(t)\right|$$

$$F_{v,linear}(t) = -C_v \dot{X}(t)$$

$$F_v(t) = F_{v,linear}(t) + F_{v,quad}(t)$$

```
%% 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];
body(1).initial.angle = pi/12;
body(1).linearDamping = zeros(6);
body(1).linearDamping(3,3) = 10;
body(1).quadDrag.cd = [0 0 1.3 0 0 0];
body(1).quadDrag.area = [0 0 314.16 0 0 0];
```

The definition of linear and quadratic damping parameters for the heave mode in the *wecSimInputFile.m* for the RM3 example.

- ***See also Advanced Features → Morison Elements for an alternative means of specifying quadratic damping and augment added mass**

Position *X*	x (surge)	y (sway)	z (heave)	rx (roll)	ry (pitch)	rz (yaw)

Power Take Off (PTO)

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

- Power take-off forces describe actuations between WEC bodies or WEC body and the fixed frame. If using the provided library blocks*, the PTO parameters are defined in the wecSimInputFile.m.
- $F_{PTO}(t) = -K_{PTO}X_{rel} - C_{PTO}\dot{X}_{rel}$
- X_{rel} is the relative motion between the nodes connected by the PTO.
- *PTOs can be modeled in a variety of ways and can leverage the full suite of Simulink and Simscape components and control tools. See also Advanced Features → PTO-Sim

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

Specification of a single DOF translational PTO in the *wecSimInputFile.m* for the RM3 Example.

Position X	x (surge)	y (sway)	z (heave)	rx (roll)	ry (pitch)	rz (yaw)

Mooring forces

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

- WEC-Sim supports a linear mooring matrix and MoorDyn (see Advanced Features → MoorDyn).
- Linear mooring matrix forces are calculated

$$F_m(t) = -K_{moor}X_{rel} - C_{moor}\dot{X}_{rel} + F_{preTension}$$

- X_{rel} is the motion of the components of the follower-side connections.

```
% Mooring Matrix
mooring(1) = mooringClass('Mooring1');     % initialize mooring
mooring(1).matrix.stiffness = zeros(6,6);
mooring(1).matrix.damping = zeros(6,6);
mooring(1).matrix.stiffness(3,3) = 1000;  % N/m applied to resist heave displacement
mooring(1).matrix.damping(3,3) = 250;     % N-m/s applied to resist heave velocity
mooring(1).matrix.preTension = [0 0 100 0 0 0]; % N pretension applied in heave
```

Specification of a mooring matrix in the *wecSimInputFile.m*

Position *X*	x (surge)	y (sway)	z (heave)	rx (roll)	ry (pitch)	rz (yaw)

Non-linear hydrodynamic forces

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

- Non-linear hydrodynamic forces include non-linear Froude-Krylov and non-linear buoyancy forces that are calculated based on panel-method integration over body geometry defined in supplemental '*.stl' files from time-resolved undisturbed wave fields and body displacements.
- ***See also Advanced Features → Non-linear Hydrodynamics**

Position *X*	x (surge)	y (sway)	z (heave)	rx (roll)	ry (pitch)	rz (yaw)

Summary of equations

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

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) $F_{rad} = -A_{\infty}\ddot{X} - \int_0^t K_r(t-\tau)\dot{X}(\tau)d\tau$ State Space Representation $\dot{X}_r(t) = A_r X_r(t) + B_r u(t); \int_0^t K_r(t-\tau)u(\tau)d\tau \approx C_r X_r(t) + D_r u(t)$
Wave Excitation (F_{ext})	Regular Waves Irregular Waves	Sinusoidal Steady-State Response $F_{exc}(t) = \Re\left[R_f(t)\frac{H}{2}F_{exc}(\omega,\theta)e^{i\omega t}\right]$ Wave Spectrum (e.g., JS; BS; PM) $F_{exc}(t) = \Re\left[R_f(t)\sum_{j=1}^{N} F_{exc}(\omega_j,\theta)\, e^{i(\omega_j t+\phi_j)}\sqrt{2S(\omega_j)d\omega_j}\right]$ Wave Elevation (Convolution Integral) $F_{exc}(t) = \int_{-\infty}^{\infty} f_e(t-\tau)\eta(\tau)d\tau$
PTO (F_{pto})		Linear Spring-Damper Hydraulic PTO Mechanical PTO $P_{PTO} = C_{PTO}\dot{X}_{rel}^2$ $F_{PTO} = -K_{PTO}X_{rel} - C_{PTO}\dot{X}_{rel}$ $P_{PTO} = -F_{PTO}\dot{X}_{rel}$ $F_{PTO} = f(X_{rel}, \dot{X}_{rel}, \ddot{X}_{rel}, \dots)$
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 $F_v = -C_v\dot{X} - C_d\rho A_d/2\,\dot{X}\,\lvert\dot{X}\rvert$ Morison Elements $F_{me} = \rho\forall\dot{v} + \rho\forall C_a(\dot{v} - \dot{X}) + C_d\rho A_d/2\,(v - \dot{X})\,\lvert v - \dot{X}\rvert$
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).

Run a WEC-Sim Simulation

1. Before funning: (Any Order)
   - Get a *.h5 file → Defines the hydrodynamic coefficients
   - Build a Simulink *.slx model → Describes device layout
   - Write wecSimInputFile.m → Defines dynamic parameters

- ***See Advanced Features → Non-linear hydrodynamics and MoorDyn for additional optional inputs**

Run a WEC-Sim Simulation

1. Before funning: (Any Order)
    - **Get a *.h5 file → Defines the hydrodynamic coefficients**
    - Build a Simulink *.slx model → Describes device layout
    - Write wecSimInputFile.m → Defines dynamic parameters

- From supported BEM codes: WAMIT, NEMOH, Capytaine, and AQWA
    - **See also Advanced Features → BEMIO**
- This BEM code will require mesh(es) to run!
- See User Manual → Workflow → Step 2

- ***See Advanced Features → Non-linear hydrodynamics and MoorDyn for additional optional inputs**

Run a WEC-Sim Simulation

2. Run WEC-Sim
   - >> wecSim or run from Simulink GUI (see Advanced Features →Running from Simulink)
   - WEC-Sim will then:

a) Clear existing variables that might conflict with those about to be loaded
b) Run *initializeWecSim.m*

Running initializeWecSim.m

a)	Reads *wecSimInputFile.m* (exactly how depends on how wecSim is being called)	**Line 55-78**
b)	Setup all objects defined in *.slx file (e.g. , constraints, ptos, moorings)	**Line 88-116**
c)	For each body, load nondimensional hydrodynamic coefficients	**Line 150-162**
d)	Setup simulation and waves, calculating wave time series	**Line 219-239**
e)	For each body, convert nondimensional hydro. Coeffs. to dimensional forces	**Line 249-256**

Running initializeWecSim.m

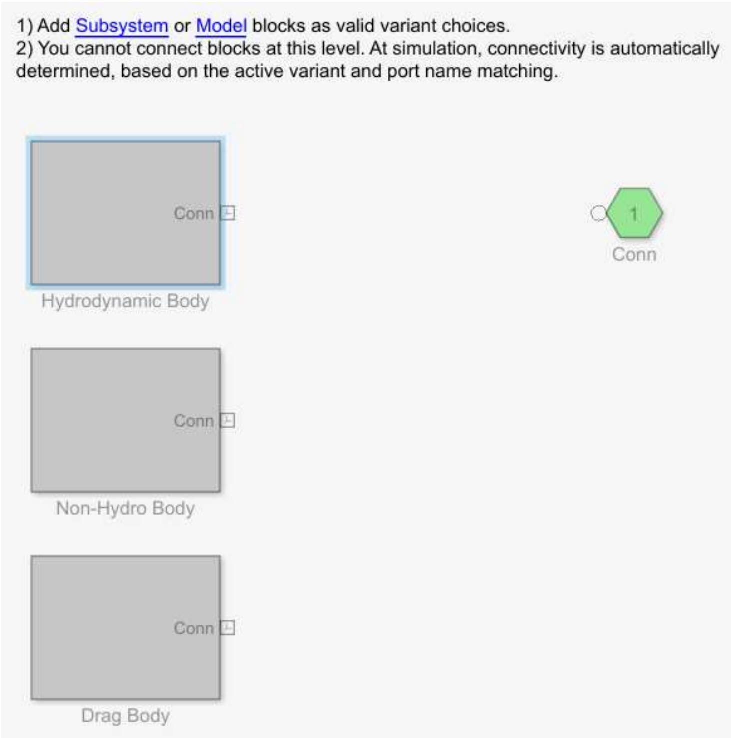
f) Diagnostic checks **Line 276-344**

g) Define variant sub-systems **Line 346-420**

A variant sub-system is Simulink block type that allows the same top-level model to follow several different execution pathways depending on the definition of a particular variable.

A variant sub-system of the Body/Rigid Body. Depending on the specification in **bodyClass**, the correct block will be connected based on variable assignment in lines 397-402.

This ensures the same *.slx model to run without modification, different sea-states and simulation parameters.



Running WecSim.m

h) After *initializeWecSim.m* finishes Run the Simulink simulation
i) Post process
   - Collate outputs into the **responseClass** (default variable name = 'output')
   - Save results
   - Run *userDefinedFunctions.m*

***The execution pathway differs for the "Run from Simulink" options, but most function calls still originate for *initializeWecSim.m*. See Advanced Features → Running from Simulink**

Notes/Warnings/Errors

```
Error using assert
Input body.quadDrag.cd should be 1x6

Error in bodyClass/checkInputs (line 149)
            assert(isequal(size(obj.quadDrag.cd)==[1,6],[1,1]),'

Error in initializeWecSim (line 127)
    body(ii).checkInputs(simu.explorer);

Error in run (line 91)
evalin('caller', strcat(script, ';'));

Error in wecSim (line 44)
run('initializeWecSim');
```

- Many common errors will have informative error messages that illustrate the mistake, but not necessarily where the error enters in the wecSimInputFile.m.
- **In this case, the quadratic drag vector '*body(1).quadDrag.cd*' was the incorrect length.**

Further Reading

- Full documentation: http://wec-sim.github.io/WEC-Sim/master/index.html
- Specifically see: http://wec-sim.github.io/WECSim/master/user/advanced_features.html#advanced-features
- The headers of class object code populate API documentation
  - https://wec-sim.github.io/WEC-Sim/master/user/api.html
- Training Slides:
  - https://wec-sim.github.io/WEC-Sim/master/user/webinars.html#online-training-course

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

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.

NREL