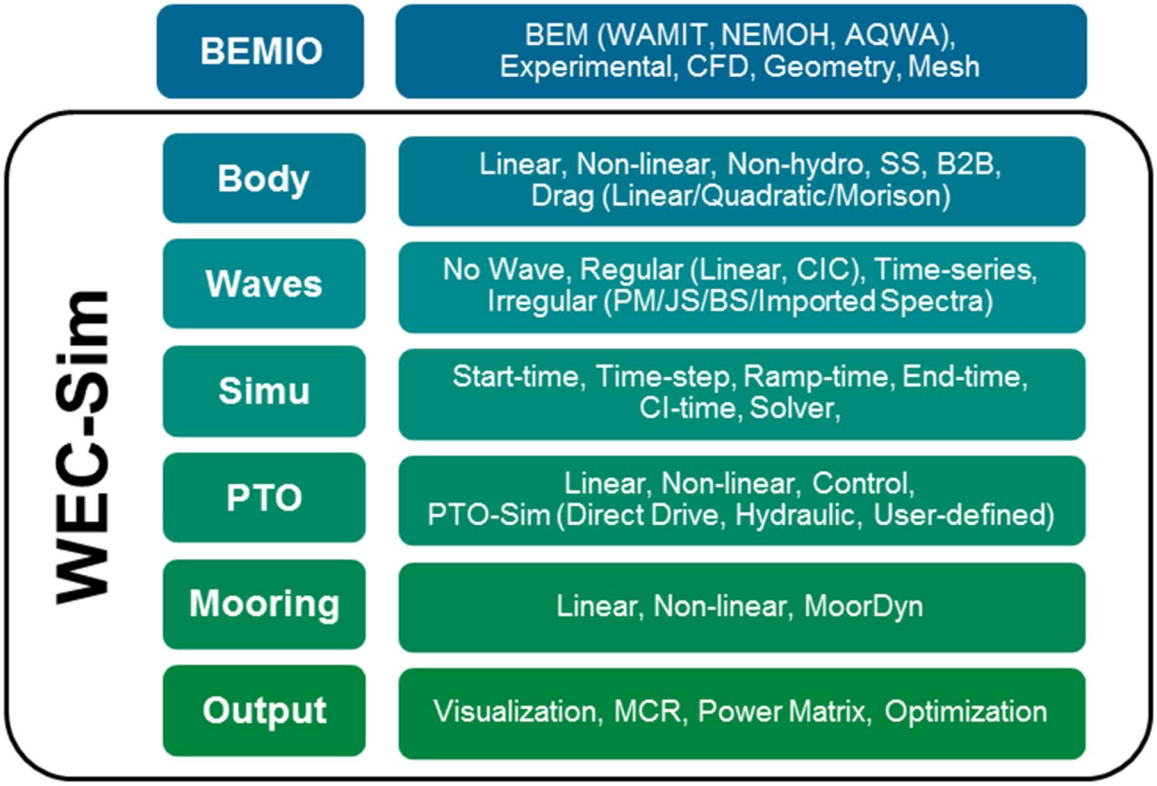
WEC-Sim2 Advanced Features 訳

佐賀大学
今井康貴（imaiy@cc.saga-u.ac.jp）

This sections provides an overview of the advanced features of the WEC-Sim code that were not covered in the WEC-Sim Tutorials section.
本章では、WEC-Sim のチュートリアル章でカバーされない WEC-Sim のコードの高度な機能を概説する.
Below is a diagram of some of the various WEC-Sim that can be used.
以下は使用できる種々の WEC-Sim を示す図である.
This section provides information on how they can be turned on, and what they do.
本章では、それらを使用する方法の情報を提供する.



BEMIO

The Boundary Element Method Input/Output (BEMIO) functions are used to preprocess the BEM hydrodynamic data prior to running WEC-Sim, this includes:
境界要素法入力/出力関数（BEMIO）は，WEC-Sim の実行前に BEM 流体データ処理に使用される．これは以下を含む：

- Read BEM results from WAMIT, NEMOH, or AQWA.
  WAMIT、NEMOH、AQWA の BEM 結果の読み込み
- Calculate the radiation and excitation impulse response functions (IRFs).
  ラディエーションおよび波強制インパルス応答関数（IRFS）の計算
- Calculate state space realization coefficients for the radiation IRF.
  ラディエーション IRF の状態空間表現係数の計算.
- Save the resulting data in Hierarchical Data Format 5 (HDF5).
  結果を HDF5 形式で保存
- Plot typical hydrodynamic data for user verification.
  ユーザ検証のため流体データのプロット

For more information, refer to the BEMIO tutorial section and the BEMIO webinar.
詳細は BEMIO チュートリアル章と BEMIO ウェビナーを参照のこと

Note 注意
Previously the python based BEMIO code was used for this purpose.
この目的のため，以前，python ベース BEMIO コードが使用された。
The python BEMIO functions have been converted to MATLAB and are included in the WEC-Sim source code.
python BEMIO 関数は、MATLAB に変換され WEC-Sim のソースコードに含まれる
The python based BEMIO code will remain available but will no longer be supported.
python ベース BEMIO コードは利用可能であるが、今後サポートされない

BEMIO Functions
BEMIO 関数

Read_WAMIT: Reads data from a WAMIT output file
Read_WAMIT：WAMIT 出力ファイルのデータ読み込み
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) or 'haskind'
  ex_coeff　-波強制力係数のタイプを示すフラグ．'diffraction'（デフォルト）または haskind

Read_NEMOH: Reads data from a NEMOH working folder
Read_NEMOH：NEMOH 作業フォルダからのデータ読み込み
hydro = Read_NEMOH(hydro, filedir)

- hydro - data structure
- filedir - NEMOH working folder, must include:
  filedir　-　NEMOH 作業フォルダ．以下を含むこと
  - 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

Note
注意
Instructions on how to download and use the open source BEM code NEMOH are provided on the NEMOH website.
オープンソース BEM コード NEMOH をダウンロードして使用する方法は NEMOH のウェブサイトにある
The NEMOH Mesh.exe code creates the Hydrostatics.dat and KH.dat files (among other files) for one input body at a time.
NEMOH Mesh.exe コードは Hydrostatics.dat と KH.dat ファイルを作成する．（他のファイルの間で）
For the Read_NEMOH function to work correctly in the case of a multiple body system, the user must manually rename Hydrostatics.dat and KH.dat files to Hydrostatics_0.dat, Hydrostatics_1.dat, …, and KH_0.dat, KH_1.dat,…, corresponding to the body order specified in the Nemoh.cal file.
複数の船体系で Read_NEMOH 関数を正しく動かすには，ユーザーが手動で Hydrostatics.dat と KH.dat の名前を変更する必要がある． Nemoh.cal ファイルで指定された船体順序に対応して Hydrostatics_0.dat, Hydrostatics_1.dat,...、KH_0.dat, KH_1.dat とする．

Read_AQWA: Reads data from AQWA output files
Read_AQWA：AQWA 出力ファイルのデータ読み込み
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 hydrodynamics coefficients in the same manner that WAMIT outputs are normalized.
正規化：WAMIT の正規化と同じ方法で NEMOH と AQWA 流体力学係数を正規化する．
Specifically, the linear restoring stiffness is normalized as, $C_{i,j}/\rho g$;
added mass is normalized as, $A_{i,j}/\rho$;
radiation damping is normalized as, $B_{i,j}/\rho\omega$; and,
exciting forces are normalized as, $X_i/\rho g$.
具体的には以下のように正規化される．
線形復原剛性　$C_{i,j}/\rho G$
付加質量　$A_{i,j}/\rho$
ラディエーション減衰　$B_{i,j}/\rho\omega$
波強制力　$X_i/\rho G$
Typically, this function would not be called directly by the user; it is automatically implemented within the Read_NEMOH and Read_AQWA functions.
この関数はユーザから直接呼び出されない
自動的に Read_NEMOH と Read_AQWA 関数内で実装される

hydro = Normalize(hydro)
- hydro - data structure

Combine_BEM: Combines multiple BEM outputs into one hydrodynamic "system".
Combine_BEM：複数の BEM 出力を 1 つの流体系に結合する
This function requires that all BEM outputs have the same water depth, wave frequencies, and wave headings.
すべての BEM 出力が同じ水深、波周波数，波方向が同じことが必要
This function would be implemented following multiple Read functions and before the IRF, Write_H5, or Plot_BEMIO functions.
この関数は以下の複数の Read 関数で実装され，IRF,Write_H5,Plot_BEMIO 関数の前に実行され

る

hydro = Combine_BEM(hydro)
- hydro - data structure

Radiation_IRF: Calculates the normalized radiation impulse response function.
Radiation_IRF ：正規化ラディエーションインパルス応答関数を計算する

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

hydro = Radiation_IRF(hydro, t_end, n_t, n_w, w_min, w_max)
- hydro - data structure
  hydro - データ構造
- t_end - calculation range for the IRF, where the IRF is calculated from t = 0 to t_end, and the default is 100 s
  t_end - IRF の計算範囲．IRF は t=0 から t_end まで計算される．デフォルトは 100 秒
- n_t - number of time steps in the IRF, the default is 1001
  n_t - IRF の時間ステップ数．デフォルトは 1001
- n_w - number of frequency steps used in the IRF calculation (hydrodynamic coefficients are interpolated to correspond), the default is 1001
  n_w - IRF 計算の周波数ステップ数(対応して流体係数が補間される)、デフォルトは 1001
- w_min - minimum frequency to use in the IRF calculation, the default is the minimum frequency from the BEM data
  w_min - IRF 計算の最小周波数．デフォルトは BEM データの最小周波数
- w_max - maximum frequency to use in the IRF calculation, the default is the maximum frequency from the BEM data.
  w_max - IRF 計算の最大周波数、デフォルトは BEM データの最大周波数

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.
Radiation_IRF_SS：ラディエーション IRF の状態空間(SS)表現の計算．この関数は Radiation_IRF 関数の後に実行すること
hydro = Radiation_IRF_SS(hydro, Omax, R2t)
- hydro - data structure
  hydro - データ構造
- Omax - maximum order of the SS realization, the default is 10
  Omax - SS 表現の最大オーダー．デフォルトは 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
  R2t - SS 表現の R2 閾値（決定係数）. 0<R2<1．デフォルトは 0.95

Excitation_IRF: Calculates the excitation impulse response function.
Excitation_IRF：波強制力インパルス応答関数の計算

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

hydro = Excitation_IRF(hydro, t_end, n_t, n_w, w_min, w_max)
- hydro - data structure
  hydro - データ構造
- 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
t_end - IRF の計算範囲．IRF は t=0 から t_end まで計算される．デフォルトは 100 秒

- n_t - number of time steps in the IRF, the default is 1001
  n_t - IRF の時間ステップ数．デフォルトは 1001
- n_w - number of frequency steps used in the IRF calculation (hydrodynamic coefficients are interpolated to correspond), the default is 1001
  n_w - IRF 計算の周波数ステップ数（対応して流体係数が補間される）、デフォルトは 1001
- w_min - minimum frequency to use in the IRF calculation, the default is the minimum frequency from the BEM data
  w_min - IRF 計算の最小周波数．デフォルトは BEM データの最小周波数
- w_max - maximum frequency to use in the IRF calculation, the default is the maximum frequency from the BEM data.
  w_max - IRF 計算の最大周波数、デフォルトは BEM データの最大周波数

Write_H5: Writes the hydro data structure to a *.h5 file.
Write_H5：流体力データ構造を*.h5 ファイルに書き込む
Write_H5(hydro)
hydro - data structure

- hydro - データ構造

Note 注意
Technically, this step should not be necessary - the MATLAB data structure hydro is written to a *.h5 file by BEMIO and then read back into a new MATLAB data structure hydroData for each body by WEC-Sim.
技術的にこのステップは必要ない - MATLAB データ構造 hydro は BEMIO によって*.h5 ファイルに書込まれ、その後 WEC-Sim により各船体の MATLAB のデータ構造 hydroData に読込まれる。
The reasons this step was retained were, first, to remain compatible with the python based BEMIO output and, second, for the simpler data visualization and verification capabilities offered by the *.h5 file viewer.
このステップが保持された理由は，第一に python ベース BEMIO との互換，第二に*.h5 ファイルビューアによる簡単なデータ可視化および検証のため．

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.
Plot_BEMIO：自由ヒーブ、サージ及びピッチ度における各体にプロット付加質量、ラディエーション減衰、ラディエーション IRF、加振力の大きさ、加振力の位相、及び励磁 IRF。
Plot_BEMIO(hydro)

- hydro - data structure
  hydro - データ構造

Note
注意
In the future, this will likely be changed to a userDefinedBEMIO.m function, similar to WEC-Sim's userDefinedFunctions.m, such that users can interactively modify or plot any BEM hydrodynamic variable of interest.
将来的には、これは WEC-Sim の userDefinedFunctions.m に似た userDefinedBEMIO.m 関数に変更される．ユーザーによる対話的な BEM 流体変数の変更，プロットを目指す．

BEMIO hydro Data Structure
BEMIO 流体データ構造

Variable	Format	Description
A	[6*N,6*N,Nf]	added mass 付加質量
Ainf	[6*N,6*N]	infinite frequency added mass 無限周波数の付加質量
B	[6*N,6*N,Nf]	radiation damping ラディエーション減衰
beta	[1,Nh]	wave headings (deg) 波向き (DEG)
body	{1,N}	body names 船体名
C	[6,6,N]	hydrostatic restoring stiffness 静圧復原剛性
cb	[3,N]	center of buoyancy 浮力中心
cg	[3,N]	center of gravity 重心
code	string	BEM code (WAMIT, AQWA, or NEMOH) BEM コード (WAMIT, AQWA, NEMOH)
ex_im	[6*N,Nh,Nf]	imaginary component of excitation 強制力の虚数成分
ex_K	[6*N,Nh,length(ex_t)]	excitation IRF 強制力 IRF
ex_ma	[6*N,Nh,Nf]	magnitude of excitation force 強制力の大きさ
ex_ph	[6*N,Nh,Nf]	phase of excitation force 強制力の位相
ex_re	[6*N,Nh,Nf]	real component of excitation 強制力の実数成分
ex_t	[1,length(ex_t)]	time steps in the excitation IRF 強制 IRF の時間ステップ
ex_w	[1,length(ex_w)]	frequency step in the excitation IRF 強制 IRF の周波数ステップ
file	string	BEM output filename BEM 出力ファイル名
g	[1,1]	gravity 重力
h	[1,1]	water depth 水深
N	[1,1]	number of bodies 船体の数
Nf	[1,1]	number of wave frequencies 波周波数の数
Nh	[1,1]	number of wave headings 波方向の数
ra_K	[6*N,6*N,length(ra_t)]	radiation IRF ラディエーション IRF
ra_t	[1,length(ra_t)]	time steps in the radiation IRF ラディエーション IRF の時間ステップ
ra_w	[1,length(ra_w)]	frequency steps in the radiation IRF ラディエーション IRF の周波数ステップ
rho	[1,1]	density 密度
ss_A	[6*N,6*N,ss_O,ss_O]	state space A matrix 状態空間行列 A
ss_B	[6*N,6*N,ss_O,1]	state space B matrix 状態空間行列 B
ss_C	[6*N,6*N,1,ss_O]	state space C matrix 状態空間行列 C
ss_conv	[6*N,6*N]	state space convergence flag 状態空間収束フラグ
ss_D	[6*N,6*N,1]	state space D matrix 状態空間行列 D
ss_K	[6*N,6*N,length(ra_t)]	state space radiation IRF 状態空間ラディエーション IRF
ss_O	[6*N,6*N]	state space order 状態空間オーダー
ss_R2	[6*N,6*N]	state space R2 fit 状態空間 R2 フィット
T	[1,Nf]	wave periods 波周期
Vo	[1,N]	displaced volume 変位体積
w	[1,Nf]	wave frequencies 波周波数

BEMIO Tutorials
BEMIO チュートリアル
The BEMIO tutorials are included in the $Source/tutorials/BEMIO directory in the WEC-Sim source code.
BEMIO チュートリアルは、WEC-Sim のソースコード内の $Source/tutorials/BEMIO にある
For more information, refer to the BEMIO webinar.

詳細は BEMIO ウェビナーを参照のこと

Writing Your Own h5 File
自分で H5 ファイルを記述する
The most common way of creating a *.h5 file is using BEMIO to post-process the outputs of a BEM code.
*.h5 ファイル作成の一般的な方法は，BEM コード出力後，BEMIO 後処理である
This requires a single BEM solution that contains all hydrodynamic bodies and accounts for body interactions.
これは単一の BEM 解が必要である．BEM 解はすべての流体力，船体間の相互干渉を含む
Some cases in which you might want to create your own h5 file are:
自身で H5 ファイルを作成する例：
- Use experimentally determined coefficients or a mix of BEM and experimental coefficients.
 - BEM 係数を使用せず実験係数の場合，または BEM と実験係数のミックスを使用する場合
- Combine results from different BEM files and have the coefficient matrices be the correct size for the new total number of bodies.
 - 異なる BEM 結果を結合し、船体の新しい合計数に対して正しいサイズの係数行列を使用する場合
-Modify the BEM results for any other reason.
その他の理由で BEM 結果を修正した場合

MATLAB and Python have functions to read and write *.h5 files easily.
MATLAB や Python は簡単に *.h5 ファイルを読み書きする関数をもつ
WEC-Sim includes three functions to help you create your own *.h5 file.
WEC-Sim は、ユーザー自身で*.h5 ファイルを作成するための 3 つの関数をもつ
These are found under $Source/functions/writeH5/.
これらは、$Source/functions/writeH5/ にある
The header comments of each function explain the inputs and outputs.
各関数のヘッダコメントが入力と出力を説明する．
An example of how to use write_hdf5 is provided in the WEC-Sim Applications repository.
write_hdf5 使用例は、WEC-Sim のアプリケーションリポジトリで提供される．
The first step is to have all the required coefficients and properties in Matlab in the correct format.
最初のステップは、必要なすべての係数および特性を正しい形式で MATLAB に持つこと
Then the functions provided are used to create and populate the *.h5 file.
そして、提供された関数を使用して *.h5 ファイルを作成する
Note 注意
The new *.h5 file will not have the impulse response function coefficients required for the convolution integral.
新しい *.h5 ファイルは、畳み込み積分に必要なインパルス応答関数の係数を持たない。
BEMIO is currently being modified to allow for reading an existing *.h5 file.
BEMIO は 既存の*.h5 ファイルを読み込みむよう変更された
This would allow you to read in the *.h5 file you created, calculate the required impulse response functions and state space coefficients, and re-write the *.h5 file.ユーザーが作成した*.h5 ファイルを読み込み，必要なインパルス応答関数と状態空間係数を計算し，*.h5 に再書き込みする

Note
BEMIO is currently being modified to allow for the combination of different *.h5 files into a single file.
BEMIO は複数の*.h5 ファイルの組み合わせを可能にするように変更された。
This would allow for the BEM of different bodies to be done separately, and BEMIO would take care of making the coefficient matrices the correct size.
さまざまな船体の BEM が個別に可能になる．BEMIO は正しいサイズ係数行列を作成する

Simulation Features
シミュレーション関数

This section provides an overview of some features included in WEC-Sim's simulation class. For more information, refer to Simulation Class.
本章では、WEC-Sim のシミュレーションクラスに含まれる関数の概要を説明する．詳細はシミュレーションのクラスを参照のこと

Multiple Condition Runs (MCR)
複数条件計算（MCR）
WEC-Sim allows users to perform batch runs by typing wecSimMCR into the MATLAB Command Window.
MATLAB コマンドウィンドウに wecSimMCR と入力することで、バッチ計算を実行できる
This command executes the Multiple Condition Run (MCR) option, which can be initiated three different ways:
このコマンドは複数条件計算（MCR）を実行する．3 つの異なる初期条件で計算できる

- Option 1. Specify a range of sea states and PTO damping coefficients in the WEC-Sim input file, example: waves.H = 1:0.5:5; waves.T = 5:1:15; pto(1).k=1000:1000:10000; pto(1).c=1200000:1200000:3600000;
オプション 1．WEC-Sim 入力ファイル中に海象と PTO 減衰係数を記述する．
例 1:
```
waves.H = 1:0.5:5;
waves.T = 5:1:15;
pto(1).k=1000:1000:10000;
pto(1).c=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: statisticsDataLoad = "<Excel file name>.xls"
オプション 2: WEC-Sim 入力ファイル中に海象データの Excel ファイルを指定する．
この方法はパワーマトリクス生成に便利である．
例 2:
```
statisticsDataLoad = "<Excel file name>.xls"
```

- Option 3. Provide a MCR case .mat file, and specify the filename in the WEC-Sim input file, example: simu.mcrCaseFile = "<File name>.mat"
オプション 3： WEC-Sim 入力ファイル中に MCR のケース.mat ファイルを指定する
例：
```
simu.mcrCaseFile =filename.mat
```

For Multiple Condition Runs, the *.h5 hydrodynamic data is only loaded once.
複数条件計算のために、*.h5 流体データが一度だけロードされる．
To reload the *.h5 data between runs, set simu.reloadH5Data =1 in the WEC-Sim input file.
計算で*.h5 データをリロードするには、WEC-Sim 入力ファイル中に simu.reloadH5Data=1 を設定する。
For more information, refer to the MCR webinar, and the WEC-Sim Applications repository MCR example.
詳細は MCR のウェビナー、WEC-Sim のアプリケーションリポジトリ MCR を参照

State-Space Representation
状態空間表現
The convolution integral term in the equation of motion can be linearized using the state-space representation as described in the Theory Section.
理論章で説明したように，運動方程式の畳み込み積分項は，状態空間表現を用いて線形化できる
To use state-space representation, the simu.ssCalc simulationClass variable must be defined in the WEC-Sim input file, for example:
状態空間表現を使用するには、WEC-Sim の入力ファイル中に simu.ssCalc simulationClass 変数を定義する必要がある.
simu.ssCalc = 1

Time-Step Features
時間ステップ
The default WEC-Sim solver is 'ode4'.
デフォルト WEC-Sim のソルバは ode4 である.
Refer to the WEC-Sim Applications repository non-linear hydro example for a comparisons between 'ode4' to 'ode45'.
ode4 と ode45 の比較は WEC-Sim アプリケーションリポジトリ non-linear hydro example を参照のこと
The following variables may be changed in the simulationClass (where N is number of increment steps, default: N=1):
simulationClass の以下の変数が変更できる（N は増分ステップ、デフォルトは N=1）：
Fixed time-step: simu.dt （固定時間ステップ）
Output time-step: simu.dtOut=N*simu.dt （出力時間ステップ）
Nonlinear hydrodynamics time-step: simu.dtNL=N*simu.dt （非線形流体力時間ステップ）
Convolution integral time-step: simu.dtCITime=N*simu.dt （畳み込み積分時間ステップ）
Morison force time-step: simu.dtME = N*N*simu.dt （モリソン力時間ステップ）

Fixed Time-Step (ode4)
固定時間ステップ（ode4）
When running WEC-Sim with a fixed time-step, 100-200 time-steps per wave period is recommended to provide accurate hydrodynamic force calculations (ex: simu.dt = T/100, where T is wave periods).
WEC-Sim を固定時間ステップで実行するとき、正確な流体力の計算のため，波周期当たり 100-200 の時間ステップが推奨される（simu.dt=T/100. T は波の周期）
However, a smaller time-step may be required (such as when coupling WEC-Sim with MoorDyn or PTO-Sim).
しかし、MoorDyn や PTO-Sim と WEC-Sim を連成させる場合, より小さな時間ステップが必要.
To reduce the required WEC-Sim simulation time, a different time-step may be specified for nonlinear hydrodynamics and for convolution integral calculations.
非線形流体力や畳み込み積分計算の場合，必要な WEC-Sim シミュレーション時間を短縮するため、可変時間ステップが指定できる
For all simulations, the time-step should be chosen based on numerical stability and a convergence study should be performed.
すべてのシミュレーションにおいて、時間ステップは数値安定性と収束性に基づいて選択されるべき.

Variable Time-Step (ode45)
可変時間ステップ（ode45）
To run WEC-Sim with a variable time-step, the following variables must be defined in the simulationClass:
WEC-Sim を可変時間ステップで実行するには、simulationClass 内に以下の変数を定義する必要

がある.
Numerical solver: simu.solver='ode45'
Max time-step: simu.dt

Wave Features
波
This section provides an overview of some features included in WEC-Sim's wave implementation.
本章では、WEC-Sim の波の実装に含まれる関数を概説する.
For more information, refer to Wave Class.
詳細は Wave クラスを参照のこと

Irregular Wave Binning
不規則波分割
The default spectral binning implemented in WEC-Sim is to divide the wave spectra into equal energy bins.
WEC-SIM 実装のスペクトルビニングは、波スペクトルを等しいエネルギービンに分割する.
This feature speeds up the irregular wave simulation time.
この関数は不規則波シミュレーションを高速化する.
To use equal frequency bins, the following waveClass variable must be defined in the WEC-Sim input file:
等周波数ビンを使用するには、WEC-Sim の入力ファイル中に次の waveClass 変数を定義する必要がある.
waves.freqDisc = 'Traditional';
When using the equal frequency formulation, users may specify the number of wave frequencies binned by defining waves.numFreq (default = 1001).
等周波数定式化を使用する場合、ユーザーは waves.numFreq（デフォルト=1001）を定義することにより，ビニング波周波数の数を指定できる
However, this means that the hydrodynamic forces are calculated at every time-step for each of the total number of frequency bins.
これは、周波数ビンの総数において，毎時間ステップ流体力が計算されることを意味する.

Wave Directionality
波方向
WEC-Sim has the ability to model waves with various angles of incidence.
WEC-Sim は様々な入射角度の波をモデル化する
To define wave directionality in WEC-Sim, the following waveClass variable must be defined in the WEC-Sim input file:
WEC-SIM での波方向を定義するには、WEC-Sim の入力ファイル中に以下の waveClass 変数を定義する必要がある.
waves.waveDir = 0;
The default incident wave direction has a heading of 0 (Default = 0), to change the heading waves.waveDir must be defined in [deg].
デフォルトの入射方向は 0(Default = 0), 方向を変えるには waves.waveDir を[deg]で指定する.

Irregular Waves with Seeded Phase
シード位相と不規則波
By default, the phase for all irregular wave cases are generated randomly.
デフォルトでは、すべての不規則波の位相はランダムに生成される.
In order to reproduce the same time-series every time an irregular wave simulation is run, the following waveClass variable may be defined in the WEC-Sim input file:
不規則波シミュレーションが実行されるたびに同じ時系列を再現するには、WEC-Sim の入力ファイル中に以下の waveClass 変数を定義する

waves.phaseSeed = <user defined seed>;
By setting waves.phaseSeed equal to 1,2,3,...,etc, the random wave phase generated by WEC-Sim is seeded, thus producing the same random phase for each simulation.
waves.phaseSeed を 1,2,3 と設定すると、WEC-SIM によって生成されるランダム位相がシーディングされる．従って、各シミュレーションに対して同じ位相を生成する．

Wave Gauge Placement
波高計の配置
By default, the wave surface elevation is calculated at the origin.
デフォルトでは，水面変位は原点で計算される．
Users are allowed up to 3 other x-locations to calculate the wave surface elevation offset from the origin in the global x-direction
by defining the waveClass variable, waves.wavegauge<i>loc, in the WEC-Sim input file:
ユーザは他の 3 つの X 座標で波高を計算できる．WEC-Sim 入力ファイル中に waveClass 変数 waves.wavegauge<i>loc を定義する
waves.wavegauge<i>loc = user defined wave gauge i x-location (y-position assumed to be 0 m)
waves.wavegauge<i>loc = ユーザー波高計 i の X 座標（Y 座標はゼロとする）
where i = 1, 2, or 3
ここで i = 1、2、3
The WEC-Sim numerical wave gauges output
the undisturbed linear incident wave elevation at the wave gauge locations defined above.
WEC-Sim 数値波高計は，上記で定義された位置における入射波の波高を出力する
The numerical wave gauges do not handle
the incident wave interaction with the radiated or diffracted waves that are generated because of the presence and motion of the WEC hydrodynamic bodies.
数値波高計は, WEC 船体とその運動により生成されるラディエーション波やディフラクション波と入射波の相互作用を扱わない.
This option provides the following wave elevation time series:
このオプションは、次の波高計時系列を出力する
waves.waveAmpTime<i> = incident wave elevation time series at wave gauge i
Note
注意
This feature only works with planar waves propagating along the positive x-direction when waves.waveDir = 0.
waves.waveDir=0 の場合、この関数は x の正方向に沿って伝搬する平面波で動作する．

Body Features
船体
This section provides an overview of some features included in WEC-Sim's body class.
本章では、WEC-Sim 船体クラスの関数を概説する．
For more information, refer to Body Class.
詳細は船体クラスを参照のこと

Body Mass and Geometry Features
船体質量と形状
The mass of each body must be specified in the WEC-Sim input file.
各船体の質量を、WEC-Sim の入力ファイル中に指定する．
The following features are available:
次の関数がある
Floating Body - the user may set body(i).mass = 'equilibrium' which will calculate the body mass based on displaced volume and water density.
船体 - body(i).mass='equilibrium'を設定する． これは排水体積と水の密度に基づき質量を計算す

る
If simu.nlhydro = 0, then the mass is calculated using the displaced volume contained in the *.h5 file.
simu.nlhydro=0 の場合、質量は*.h5 ファイルの排水体積を用いて計算する.
If simu.nlhydro = 1 or simu.nlhydro = 2, then the mass is calculated using the displaced volume of the provided STL geometry file
simu.nlhydro=1 又は simu.nlhydro=2 では，質量は STL 形状ファイルの変位量を用いて計算される.
Fixed Body - if the mass is unknown (or not important to the dynamics), the user may specify body(i).mass = 'fixed' which will set the mass to 999 kg and moment of inertia to [999 999 999] kg-m^2.
固定船体 - 質量が不明の（または運動にとって重要ではない）場合、body(i).mass = 'fixed' を指定する．これは，質量を 999kg，慣性モーメントを[999 999 999] kg-m^2 とする.
Import STL - to read in the geometry (stl) into Matlab use the body(i).bodyGeo method in the bodyClass.
インポート STL - stl ファイルを MATLAB に読み込む．bodyClass で body(i).bodyGeo を使用する
This method will import the mesh details (vertices, faces, normals, areas, centroids) into the body(i).bodyGeometry property.
この方法は、body(i).bodyGeometry にメッシュの詳細（頂点、面、法線、領域、重心）をインポートする.
This method is also used for non-linear hydrodynamics and ParaView visualization files.
この方法は、非線形流体力学や ParaView 可視化ファイルに使用される.
Users can then visualize the geometry using the body(i).plotStl method.
body(i).plotStl を使用して形状が可視化される

Non-Linear Hydrodynamics
非線形流体力
WEC-Sim has the option to include the non-linear hydrostatic restoring and Froude-Krylov forces when solving the system dynamics of WECs, accounting for the weakly nonlinear effect on the body hydrodynamics.
WEC-Sim は、WEC 運動を解く際、復原力と FK 力に非線形流体力を入れるオプションをもつ.
これらは，船体流体力に弱い非線形性をもつ.
To use non-linear hydrodyanmics, the simu.nlHydro simulationClass variable must be defined in the WEC-Sim input file, for example:
非線形流体力を使用するには、WEC-Sim の入力ファイル中に simu.nlHydro simulationClass 変数を定義する．例:
simu.nlHydro = 2
For more information, refer to the non-linear hydrodynamics webinar, and the WEC-Sim Applications repository non-linear hydrodynamics example.
詳細は非線形流体力学のウェビナー，WEC-Sim アプリケーションリポジトリの non-linear hydrodynamics example を参照

Non-Linear Settings
非線形設定
simu.nlHydro - The nonlinear hydrodynamics option can be used by setting simu.nlHydro = 2 or simu.nlHydro = 1 in your WEC-Sim input file.
simu.nlHydro - 非線形流体力オプションは、WEC-Sim 入力ファイルに simu.nlHydro= 2 または simu.nlHydro=1 を設定すると使用できる
Typically, simu.nlHydro = 2 is recommended if nonlinear hydrodynamic effects need to be used.
非線形流体効果が必要な場合、典型的には simu.nlHydro = 2 が推奨される.
Note that simu.nlHydro = 1 only considers the nonlinear restoring and Froude-Krylov forces based

on the body position and mean wave elevation.
simu.nlHydro=1 は, 船体位置および平均水位変動に基づいた非線形復原力と非線形 FK 力しか考慮しない.
simu.dtNL - An option available to reduce the nonlinear simulation time is to specify a nonlinear time step, simu.dtNL=N*simu.dt, where N is number of increment steps.
simu.dtNL - 非線形シミュレーションの時間短縮に利用可能なオプションは、非線形時間ステップ、simu.dtNL=N*simu.dt を指定する．N は増分ステップの数
The nonlinear time step specifies the interval at which the nonlinear hydrodynamic forces are calculated.
非線形時間ステップは、非線形流体力を計算する間隔を指定する.
As the ratio of the nonlinear to system time step increases, the computation time is reduced, again, at the expense of the simulation accuracy.
システム時間ステップに対する非線形比が増えると、計算時間は低減する．シミュレーション精度が犠牲になる
Note
注意
WEC-Sim's nonlinear hydrodynamic option may be used for regular or irregular waves but not with user-defined irregular waves.
WEC-Sim の非線形流体力オプションは、規則波，不規則波で使用できる.
ユーザー定義の不規則波とは一緒に使用できない

STL File Generation
STL ファイルの生成
When the nonlinear option is turned on, the geometry file (*.stl) (previously only used for visualization purposes in linear simulations) is used as the discretized body surface on which the non-linear pressure forces are integrated.
非線形オプションをオンにすると、非線形圧力が積分される近似物体表面として形状ファイル（*.stl）が使用される（以前は線形シミュレーション可視化のみに使用された）
A good STL mesh resolution is required for the WEC body geometry file(s) when using the non-linear hydrodynamics in WEC-Sim.
WEC-SIM で非線形流体力を使用する場合，WEC 船体形状ファイルに高い STL メッシュ解像度が必要である.
The simulation accuracy will increase with increased surface resolution (i.e. the number of discretized surface panels specified in the .stl file), but the computation time will also increase.
表面解像度（.STL ファイルのパネルの数）が高くなるとシミュレーションの精度が増加する．しかし計算時間も増加する.
There are many ways to generate an STL file; however, it is important to verify the quality of the mesh before running WEC-Sim simulations with the non-linear hydro flag turned on.
STL ファイル生成には多くの方法がある．WEC-Sim シミュレーションで非線形流体力を計算する前にメッシュ品質を確認することが重要である.
An STL file can be exported from from most CAD programs, but few allow adaquate mesh refinement.
STL ファイルは、ほとんどの CAD プログラムからからエクスポート可能であるが，メッシュ修正が必要な場合がある
A good program to perform STL mesh refinement is Rhino3d.
STL メッシュ修正を行う良いプログラムが Rhino3d である.
Some helpful resources explaining how to generate and refine an STL mesh in Rhino3d can be found here and here.
Rhino3d で STL メッシュを生成し、修正する方法がリンクにある
Note 注意
All STL files must be saved as ASCII (not binary)
STL ファイルは ASCII 形式で保存する（バイナリ形式はダメ）

Refining STL File - The script refine_stl in the BEMIO directory performs a simple mesh refinement on an *.stl file
by subdividing each panel with an area above the specified threshold
into four smaller panels with new vertices at the mid-points of the original panel edges.
STL ファイル修正 - BEMIO ディレクトリ内のスクリプト refine_stl は，*.STL ファイルの簡単なメッシュ修正を行う．しきい値以上の面積をもつパネルを，
元パネルの辺の中点を節点とする 4 つのパネルに分割する．
This procedure is iterated for each panel until all panels have an area below the specified threshold, as in the example rectangle.
指定された閾値以下の面積になるまで、この手順が各パネルで繰り返される．
In this way, the each new panel retains the aspect ratio of the original panel.
このように、新しいパネルは、元パネルのアスペクト比を保持する．
Note that the linear discretization of curved edges is not refined via this algorithm.
このアルゴリズムでは，湾曲部の線形分割は改良されないことに留意
The header comments of the function explain the inputs and outputs.
関数のヘッダコメントは入力および出力を説明する．
This function calls import_stl_fast, included with the WEC-Sim distribution, to import the .*stl file.
この関数は、*.STL ファイルのインポートに、WEC-Sim ディストリビューションに含まれる import_stl_fast を呼び出す．

Non-Linear Tutorial - Heaving Ellipsoid
非線形チュートリアル - ヒービング楕円体
The body tested in the study is an ellipsoid with a cross- section characterized by semi-major and -minor axes of 5.0 m and 2.5 m in the wave propagation and normal directions, respectively .
本研究の試験体は楕円である．断面は長軸（波進行方向）5.0m，短軸（波進行と直交方向）2.5m
The ellipsoid is at its equilibrium position with its origin located at the mean water surface.
楕円は平衡位置にある．平均水面位置を原点とする．
The mass of the body is then set to 1.342×105 kg, and the center of gravity is located 2 m below the origin.
船体質量は 1.342×105kg、重心座標は原点の 2m 下方とした
STL file with the discretized body surface is shown below (ellipsoid.stl)
離散表面の STL ファイルを下に示す（ellipsoid.stl）
The single-body heave only WEC model is shown below (nonLinearHydro.slx)
単一船体ヒーブ WEC モデルを以下に示す（nonLinearHydro.slx）
The WEC-Sim input file used to run the non-linear hydro WEC-Sim simulation:
非線形流体力 WEC-Sim シミュレーションを実行するための WEC-Sim 入力ファイル：

```
%% Simulation Settings シミュレーション設定
simu = simulationClass(); %Create the Simulation Variable
simu.endTime=100;     , % Simulation End Time [s]
simu.dt = 0.05;, % Simulation Delta_Time [s]
simu.simMechanicsFile = 'ellipsoid.slx'; , % Specify Simulink Model File
simu.rho=1025;
simu.mode='normal';
simu.explorer='on';% Turn SimMechanics Explorer on% (on/off)
simu.CITime = 30;
simu.rampT= 50;
simu.nlHydro= 2;, , % Turns non-linear hydro on

% Wave Information
waves = waveClass('regular'); %Create the Wave Variable and Specify Type
waves.H = 4;% Wave Height [m]
```

```
waves.T = 6;    , % Wave period [s]

%Body Data
%船体データ
body(1) = bodyClass('wamit/ellipsoid.h5');, % Initialize bodyClass for Float
body(1).mass = 'equilibrium'; , % Mass from WAMIT [kg]
body(1).momOfInertia = ..., % Moment of Inertia [kg-m^2]
[1.375264e6 1.375264e6 1.341721e6];
body(1).geometryFile = 'geometry/elipsoid.stl' ; , % Discretized body geometry for nlHydro
body(1).viscDrag.cd=[1 0 1 0 1 0];
body(1).viscDrag.characteristicArea=[25 0 pi*5^2 0 pi*5^5 0];

% PTO and Constraint Parameters
constraint(1) = constraintClass('Constraint1');
constraint(1).loc = [0 0 -12.5];    , %Constraint Location [m]

pto(1) = ptoClass('PTO1'); , % Initialize ptoClass for PTO1
pto(1).k=0;     , % PTO Stiffness Coeff [N/m]
pto(1).c=1200000; , % PTO Damping Coeff [Ns/m]
pto(1).loc = [0 0 -12.5];
```

Simulation and post-processing is the same process as described in Tutorials section.
シミュレーションとポストプロセスはチュートリアルと同じである

Non-Hydrodynamic Bodies
流体力をもたない船体
For some simulations, it might be important to model bodies that do not have hydrodynamic forces acting on them.
流体力を持たない船体をモデル化することが重要なシミュレーションもある。
This could be bodies that are completely outside of the water but are still connected through a joint to the WEC bodies, or it could be bodies deeply submerged to the point where the hydrodynamics may be neglected.
これは、完全に水の外にあるが、ジョイントを介して WEC 船体に接続される場合や,
流体力を無視できる深度に沈んだ船体等である.
WEC-Sim allows for bodies which have no hydrodynamic forces acting on them and for which no BEM data is provided.
WEC-SIM は，流体力を持たない船体，BEM データが提供されない船体の扱いが可能である
To do this, use a Body Block from the WEC-Sim Library and
initialize it in the WEC-Sim input file as any other body but leave the name of the h5 file as an empty string.
これを行うには、WEC-Sim のライブラリから Body Block を使用し、
WEC-Sim の入力ファイル内で他の船体同様に初期化する．しかし，空 H5 ファイルの名前を空白として残す？
Specify body(i).nhBody = 1; and specify body name, mass, moments of inertia, cg, geometry file, location, and displaced volume.
body(i).nhBody=1 を指定．船体名、質量、慣性 M、CG、形状ファイル、位置，排水容積を指定する.
You can also specify visualization options and initial displacement.
可視化オプションと初期変位を指定できる
To use non-hydrodynamic bodies, the following bodyClass variable must be defined in the WEC-Sim input file, for example:
流体力をもたない船体を使用するには、WEC-Sim の入力ファイル内に 以下の bodyClass 変数を

定義する．例
body(i).nhBody = 1
For more information, refer to the non-hydro bodies webinar, and the WEC-Sim Applications repository non-hydro body example.
詳細は non-hydro bodies ウェビナー、WEC-Sim のアプリケーション non-hydro body 参照

Body-To-Body Interactions
船体間の相互干渉
WEC-Sim allows for body-to-body interactions in the radiation force calculation, thus allowing the motion of one body to impart a force on all other bodies.
WEC-Sim はラディエーション力の計算において船体間の相互作用を可能にする．物体の運動が他の船体の力へ付与される
The radiation matrices for each body (radiation damping and added mass) required by WEC-Sim and contained in the *.h5 file.
WEC-Sim に必要な各船体のラディエーション行列(ラディエーション減衰及び付加質量)は *.h5 ファイルにある
For body-to-body interactions with N total hydrodynamic bodies, the *h5 data structure is [(6*N), 6].
N 物体の相互干渉において*.h5 のデータ構造は[(6*N),6]である
When body-to-body interactions are used, the augmented [(6*N), 6] matrices are multiplied by concatenated velocity and acceleration vectors of all hydrodynamic bodies.
船体間干渉を使用する場合、[(6*N),6]行列は船体全体の連結速度と加速度ベクトルで乗算される．
For example, the radiation damping force for body(2) in a 3-body system with body-to-body interactions would be calculated as the product of a [1,18] velocity vector and a [18,6] radiation damping coefficients matrix.
例えば、3 体問題における物体 2 のラディエーション減衰の干渉は[1,18]速度ベクトルと[18,6]ラディエーション減衰係数行列の積として計算される．
To use body-to-body interactions, the following simulationClass variable must be defined in the WEC-Sim input file, for example:
船体間の相互作用を使用するには、WEC-Sim の入力ファイル内に以下の simulationClass 変数を定義する．
simu.b2b = 1
For more information, refer to the B2B webinar, and the WEC-Sim Applications repository B2B example.
詳細は B2B のウェビナー、WEC-Sim のアプリケーションリポジトリの B2B を参照のこと
Note 注意
By default, body-to-body interactions are off (simu.b2b = 0), and only the [1+6*(i-1):6*i, 1:6] sub-matrices are used for each body (where i is the body number).
デフォルトでは船体間相互干渉はオフ(simu.b2b=0)．[1+6*(I-1):6*I,1:6]の部分行列が各船体に使用される．i は船体番号である．

Morison Elements
モリソン要素
To use Morison Elements, the following simulationClass variable must be defined in the WEC-Sim input file, for example:
モリソン要素を使用するには、WEC-Sim 入力ファイル内に以下の simulationClass 変数を定義する必要がある．
simu.morrisonElement = 1
Morison Elements must then be defined for each body using the body(#).morrisonElement property of the body class.
次に，モリソン要素を body クラスの body(#).morrisonElement を用いて定義する
This proporty recquires definition of body(#).morrisonElement.cd, body(#).morrisonElement.ca,

body(#).morrisonElement.characteristicArea, body(#).morrisonElement.VME, and body(#).morrisonElement.rgME (each of which have a default value of zero).
以下の設定が必要である．
body(#).morrisonElement.cd,
body(#).morrisonElement.ca,
body(#).morrisonElement.characteristicArea,
body(#).morrisonElement.VME,
body(#).morrisonElement.rgME
各々のデフォルト値はゼロ
The Morison Element time-step may also be defined as simu.dtME = N*simu.dt, where N is number of increment steps.
モリソン要素時間ステップは simu.dtME = N*simu.dt で定義される．N は増分ステップの数である
Note 注意
Morison Elements cannot but used with etaImport.
モリソン要素 etaImport と一緒に使用できない

Constraint and PTO Features
拘束と PTO
This section provides an overview of some features included in WEC-Sim's control and pto classes.
本章では、WEC-Sim の制御や PTO のクラスの関数の概説する．
For more information, refer to Constraint Class and PTO Class.
詳細は拘束クラスおよび PTO クラスを参照
The default linear and rotational constraints and PTOs are allow for heave and pitch motions of the follower relative to the base.
デフォルトの線形および回転の拘束と PTO は、ベースに対するフォロワのヒーブとピッチを許可する
To obtain a linear or rotational constraint in a different direction
you must modify the constraint's or PTO's coordinate orientation.
異なる方向の直線または回転の拘束を許可するためには、拘束や PTO の向きを変更する
The important thing to remember is that a linear constraint or PTO will always allow motion along the joint's Z-axis, and a rotational constraint or PTO will allow rotation about the joint's Y-axis.
重要なことは、
線形拘束や PTO は，ジョイントの Z 軸に沿った運動が可能、
回転拘束や PTO は，ジョイントの Y 軸周りの回転が可能なことである
To obtain translation along or rotation about a different direction relative to the global frame, you must modify the orientation of the joint's coordinate frame.
グローバルフレームに対して異なる方向に沿って平行移動または回転を取得するには、ジョイントの座標フレームの向きを変更する必要がある
This is done by setting the constraint's or PTO's orientation.z and orientation.y properties which specify the new direction of the Z- and Y- joint coordinates.
これは拘束や PTO の orientation.z と orientation.y プロパティの設定で行われる。
Z-及び Y-関節座標の新しい方向を指定する．
The Z- and Y- directions must be perpendicular to each other.
Z-と Y-方向は互いに垂直でなければならない
As an example, if you want to constrain body 2 to surge motion relative to body 1 using a linear constraint, you would need the constraint's Z-axis to point in the direction of the global surge (X) direction.
一例として、船体 2 を船体 1 に対して線形拘束を使用して強制サージする場合、
拘束の Z 軸がグローバルサージ（X）方向を指す必要があったとする
This would be done by setting constraint(i).orientation.z=[1,0,0] and the Y-direction to any perpendicular direction (can be left as the default y=[0 1 0]).

これは、setting constraint(i).orientation.z=[1,0,0]，Y 方向を任意の直交方向にとることで実現される．デフォルト y=[0 1 0]のままでもよい．
In this example, the Y-direction would only have an effect on the coordinate on which the constraint forces are reported but not on the dynamics of the system.
この例では、Y 方向は、拘束力が報告される座標にのみ影響を与えるが、系の力学には影響しない。
Similarly if you want to obtain a yaw constraint you would use a rotational constraint and align the constraint's Y-axis with the global Z-axis.
同様にヨー拘束を取得する場合は、回転拘束を使用し、拘束の Y 軸をグローバル Z 軸に揃える
This would be done by setting constraint(i).orientation.y=[0,0,1] and the z-direction to a perpendicular direction (say [0,-1,0]).
これは、constraint(i).orientation.y=[0,0,1]，z 軸をこれに直交させる(たとえば[0,-1,0])ことで実現される．
Note 注意
When using the Actuation Force/Torque PTO or Actuation Motion PTO blocks, the loads and displacements are specified in the local (not global) coordinate system.
Actuation Force/Torque PTO ブロックや Actuation Motion PTO ブロックを使用する場合、荷重と変位は座標系（グローバルではない）ローカル座標系で指定される
This is true for both the sensed (measured) and actuated (commanded) loads and displacements.
これは、検出（測定）と作動（指令）荷重と変位の両方で正しい
Additionally, by combining constraints and PTOs in series you can obtain different motion constraints.
また、拘束と PTO を直列に組み合わせると、異なる運動拘束になる
For example, a massless rigid rod between two bodies, hinged at each body, can be obtained by using a two rotational constraints in series, both rotating in pitch, but with different locations.
例えば、ピッチで回転するが、位置が異なる 2 つの回転制約条件を使用することによって、各本体にヒンジ結合された 2 つの本体間の質量のない剛性ロッドを得ることができる。
A roll-pitch constraint can also be obtained with two rotational constraints in series; one rotating in pitch, and the other in roll, and both at the same location.
ロールピッチ制約は、2 つの回転制約を直列にして得ることもできる。 1 つはピッチで回転し、もう 1 つはロールで、そして両方は同じ位置で回転する。

Power Take-Off/PTO-Sim
動力取り出し/ PTO-sim
PTO-Sim is the WEC-Sim module responsible for accurately modeling a WEC's conversion of mechanical power to electrical power.
PTO-Sim は，WEC 機械動力から電力への変換をモデル化する WEC-SIM モジュールである．
While the PTO blocks native to WEC-Sim are modeled as a simple linear spring-damper systems, PTO-Sim is capable of modeling many power conversion chains (PCC) such as mechanical drivetrain and hydraulic drivetrain.
WEC-Sim ネイティブ PTO ブロックは, 単純な線形バネ - ダンパー系でモデル化されるが、PTO-Sim は、機械ドライブトレインや油圧駆動系のような多くの電力変換チェーン（PCC）をモデル化することが可能である．
PTO-Sim is made of native Simulink blocks coupled with WEC-Sim, using WEC-Sim's user-defined PTO blocks, where the WEC-Sim response (relative displacement and velocity for linear motion and angular position and velocity for rotary motion) is the PTO-Sim input.
PTO-Sim は，WEC-Sim に結合したネイティブ Simulink ブロックで構成される
WEC-Sim 応答（直線運動の相対変位及び速度，回転運動の角度及び角速度）は PTO-Sim の入力になる
Similarly, the PTO force or torque is the WEC-Sim input.
同様に、PTO 力やトルクが WEC-Sim の入力である．
For more information on how PTO-Sim works, refer to [So et al., 2015], and the PTO and Control

webinar.
PTO-SIM 動作の詳細は[So et al., 2015]、PTO コントロールウェビナーを参照のこと
The files for the PTO-Sim tutorials described in this section can be found in the WEC-Sim Applications repository.
本章で説明する PTO-Sim のチュートリアルファイルは、WEC-Sim のアプリケーションリポジトリにある

Tutorial: RM3 with PTO-Sim
チュートリアル：PTO-SIM と RM3
This section describes how to use RM3 with PTO-Sim.
本章では、RM3 と PTO-Sim を使用する方法を説明する．
Two tutorials will be given in this section: one for the RM3 with a hydraulic PTO (non-compressible and compressible) and another for the RM3 with a direct drive PTO.
本章では 2 つのチュートリアルが説明される：
1 つは油圧 PTO（非圧縮および圧縮）をもつ RM3，もう 1 つはダイレクトドライブを PTO をもつ RM3 用である

RM3 with Hydraulic PTO
油圧 PTO をもつ RM3
The hydraulic PTO example used in this section consists of a piston, a rectifying valve, a high pressure accumulator, a hydraulic motor coupled to a rotary generator, and a low pressure accumulator.
本章で使用される油圧 PTO 例は、ピストン、整流弁、高圧アキュムレータ、油圧モータとれ連結した回転発電機、低圧アキュムレータから構成される
There are two ways of modeling the hydraulic PTO: with a compressible fluid hydraulic, and with a non-compressible fluid hydraulic.
油圧 PTO のモデル化に圧縮性流体と非圧縮性流体の 2 つの方法がある
The compressible fluid model uses the properties of fluid such as an effective bulk modulus and density while the non-compressible fluid does not.
圧縮性流体モデルは、有効体積弾性率，密度などの流体の特性を使用する．非圧縮性流体は使用しない
In this section, a step by step tutorial on how to set up and run the RM3 simulation with PTO-Sim is provided.
本章では、PTO-Sim と RM3 の設定およびシミュレーションが説明される
All the files used in WEC-Sim will remain the same.
WEC-SIM で使用されるファイルは同じ
An additional file that is needed is the PTO-Sim input file (ptoSimInputFile.m).
必要な追加ファイルは PTO-Sim 入力ファイル（ptoSimInputFile.m）である．
If the rotary generator lookup table is used, a datasheet that contains generator efficiency, torque, and angular velocity is needed and
should be named as table in Workspace (table.eff, table.Tpu,and table.omegapu).
回転発電ルックアップテーブルが使用される場合、発電効率、トルク、角速度のデータシートが必要である．ワークスペース内で table と名前をつける（table.eff、table.Tpu、table.omegapu）
More details, refer to Step 8.
詳細は手順 8 を参照のこと
In summary, the files need to run RM3 with PTO-Sim case are the following:
要約すると、PTO-Sim もつ RM3 シミュレーションの実行には以下のファイルが必要である
WEC-Sim input file: wecSimInputFile.m (make sure to set the PTO linear damping to zero) (PTO linear damping をゼロに設定)
Simulink model: RM3.slx
Geometry file for each body: float.stl and plate.stl
Hydrodynamic data file(s): rm3.h5

Optional user defined postprocessing file: userDefinedFunction.m
PTO-Sim input file: ptoSimInputFile.m
Datasheet for the rotary generator: table
(table.eff, table.Tpu,and table.omegapu)
For the hydraulic PTOs: variableMotorVolume.m

Simulink Model
Simulink モデル
The Simulink model can be built as following:
Simulink モデルは以下のように構築する
Step 1: Navigate to the RM3 tutorial $Source/tutorals/RM3.
ステップ 1：RM3 チュートリアル $Source/tutorals/RM3 に移動する．
Step 2: Open RM3.slx file and replace Translational PTO (local Z) with Translational PTO UD Force (Local Z).
ステップ 2：RM3.slx の Translational PTO (local Z)と Translational PTO UD Force (Local Z)をを交換
Step 3: Use a subsystem and rename it to PTO-Sim where input is response and output is force.
ステップ 3：サブシステムを使用し名前を PTO-Sim にする．入力は応答，出力は力である．
Step 4: Go inside PTO-Sim block and add one bus selector and two selector blocks.
ステップ 4：PTO-Sim ブロック内に 1 つのバスセレクタ，2 つのセレクタブロックを追加する．
Since PTO-Sim block is connected to the WEC-Sim translational joint block, you can select position and velocity and therefore "signal1" and "signal2" will change to "position" and "velocity".
PTO-Sim ブロックが WEC-Sim translational joint block に接続されているので、位置と速度を選択できる．したがって、signal1 と signal2 は position と velocity に変更される．
Because the heave motion is driving the piston, selection index of each selector needs to be changed to 3.
ヒーブ運動がピストンを駆動するので、各セレクタの選択指数は 3 に変更する．
Step 5: Go to Simulink Library Browser to access PTO-Sim Library.
ステップ 5：PTO-Sim ライブラリにアクセスするため，Simulink ライブラリブラウザに移動する．
Step 6: By looking at the physical hydraulic PTO model as shown above, user can simply drag and drop PTO-Sim library blocks.
ステップ 6：上記のように物理的な油圧 PTO モデルを見ることにより、ユーザは単に PTO-Sim のライブラリブロックをドラッグ＆ドロップできる
Piston, valves, accummulator blocks are located under Hydraulic block.
ピストン、バルブ、accummulator ブロックは油圧ブロックの下に配置される．
Rotary generator lookup table is under Generator block.
ロータリー発電機のルックアップテーブルは Generator ブロックの下にある．
Step 7: Since two accumulators are needed for the high pressure accumulator and low pressure accumulator, user need to douple-click on each block and give a number to each accumulator.
ステップ 7：高圧アキュムレータと低圧アキュムレータが必要なため、ユーザは各ブロックにアキュムレータ番号をつける．
For example, ptosim.accumulator(1) is called high pressure accumulator and ptosim.accumulator(2) is called low pressure accumulator.
例えば、ptosim.accumulator(1)は高圧アキュムレータ，ptosim.accumulator(2)は定圧アキュムレータである．
Step 8: If a rotary generator lookup table is used, this block assumes user will provide the datasheet.
ステップ 8：回転発電機ルックアップテーブルを使用する場合、ブロックは、ユーザーがデータシートを提供すると仮定する
After the datasheet is loaded into Workspace, it needs to be named as table because the word table is used inside Simulink lookup table block.
データシートがワークスペースにロードされた後、"table"と名付ける．

Simulink ルックアップテーブルブロックの内部では"table"が使用されるため
The datasheet in tutorials is taken from ABB datasheet part number M3BJ315SMC.
チュートリアルでは、データシートは ABB データシート部品番号 M3BJ315SMC が使用される
The lookup table takes three inputs: efficiency (table.eff), anglular velocity (table.Tpu), and generator torque (table.omegapu), respectively.
ルックアップテーブルは 3 つの入力を要する：効率 (table.eff)、角速度 (table.Tpu)、発電機トルク (table.omegapu)：
Step 9: After the high pressure and low pressure accumulators have been identified, and the rotary generator lookup table datasheet has been setup, all the blocks can be connected together.
ステップ 9：高圧と低圧アキュムレータが決定され、回転発電ルックアップテーブルのデータシートが設定された後、全てのブロックが一緒に接続できる

Position and velocity from selectors are used as inputs of compressible fluid piston.
セレクタからの位置と速度は、圧縮性流体ピストンの入力として使用される.
This block also needs to know top and bottom volumetric flows which come from the rectifying check valve.
このブロックは、整流チェックバルブから来る上と下の体積流量を知る必要がある.
The piston then outputs PTO force that will be used by WEC-Sim.
ピストンはその後、WEC-SIM で使用される PTO 力を出力する.
Two other outputs are the piston pressures.
他の出力は、ピストン圧である.
The rectifying check valve takes both the pressures from the piston and accumulators.
整流チェックバルブは、ピストンとアキュムレータからの圧力を使う
Both high and low pressure accumulators takes the volumetric flows from the rectifying check valve and hydraulic motor.
両方の高圧及び低圧アキュムレータは、整流弁と油圧モータからの体積流量を使う
The hydraulic motor uses the knowledge of the pressures from both accumulator and generator torque from the rotary generator.
油圧モータは、回転発電機からのアキュムレータ圧力と発電機トルクを使う.
The rotary generator needs angular velocity from the hydraulic motor.
回転発電機は、油圧モータの角速度を必要とする.
The figure below shows how to connect all the blocks together.
以下の図は、すべてのブロックを接続する方法を示す。

Input File
入力ファイル
In this section, PTO-Sim input file (ptoSimInputFile.m) is defined and categorized into sections such as piston, rectifying check valve, high pressure accumulator, hydraulic motor, low pressure accumulator, and rotary generator.
本章では、PTO-Sim の入力ファイル (ptoSimInputFile.m) が定義される．また，ピストン，整流チェック弁，高圧アキュムレータ，油圧モータ，低圧アキュムレータ，回転発電機などに分類される.
Simulation and Postprocessing Simulation and postprocessing are the same process as described in WEC-Sim Simulation example above.
上記 WEC-Sim シミュレーション例で示したように、シミュレーションとポストプロセスは同じである.

RM3 with Direct Drive PTO
ダイレクトドライブ PTO をもつ RM3
A mechanical PTO is used in this example and is modeled as a direct drive linear generator.
この例では機械 PTO が使用され、ダイレクトドライブリニア発電機としてモデル化される.

The main components of this example consist of magnets and a coil where the magnet assembly is attached to the heaving float and the coil is locacted inside the spar.
主要な構成要素は磁石とコイル。磁石アセンブリがヒーブフロートに取り付けられ、コイルがスパー内部にある
As the float moves up and down, the magnet assembly creates a change in the magnetic field surrounding the spar that contains the coil: therefore, current is induced in the coil and electricity is generated.
フロートが上下すると、磁石アセンブリがスパーを取り囲む磁界を変える.
したがって、電流がコイルに誘起され発電する.
Simulink Model Step 1 through 3 are the same as in RM3 with hydraulic PTO.
Simulink モデルのステップ 1 から 3 は油圧 PTO と同じである.
Step 4: Go inside PTO-Sim block and add one bus selector and one selector blocks. Only velocity is needed for this example.
ステップ 4：PTO-Sim のブロック内え 1 つのバスセレクタと、1 つのセレクタ・ブロックを追加する. この例では速度だけが必要
Step 5: Go to PTO-Sim library.
ステップ 5：PTO-Sim のライブラリに移動する.
Step 6: By looking at the physical mechanical PTO model as shown above, the user can simply drag and drop PTO-Sim library blocks.
ステップ 6：ユーザは、単に PTO-Sim のライブラリブロックをドラッグ&ドロップする
In this case, only the direct drive linear generator is needed, and it is located under generator box.
この例では、direct drive linear generator のみ必要であり，それは発電機ボックスの下にある
Step 7: Simply connect velocity from the selector to the input of the direct drive linear generator.
ステップ 7：direct drive linear generator の入力にセレクタの速度を接続する.
The ouput PTO force is fed back to WEC-Sim.
出力である PTO 力は WEC-SIM にフィードバックされる.
Input File, Simulation, and Postprocessing The same as RM3 with hydraulic PTO.
入力ファイル、シミュレーション、後処理は油圧 PTO と同じ。

Tutorial: OSWEC with PTO-Sim
チュートリアル：PTO-SIM と OSWEC
This section describes how to use OSWEC with PTO-Sim.
本章では PTO-Sim で OSWEC を模擬する方法を説明する.
The same process as described in RM3 with PTO-Sim ; however, since OSWEC is a rotary device, it takes torque as an input and a rotary to linear motion conversion block is needed.
RM3 と同じプロセスであるが，OSWEC は回転装置のため、トルクを入力し、回転から直動への変換ブロックが必要である.
The tutorials can be found on the WEC-Sim Applications repository (both for a crank and for a rod).
チュートリアルは WEC-Sim アプリケーションリポジトリにある（クランクおよびロッド用の両方）

OSWEC with Hydraulic PTO
油圧 PTO 付き OSWEC
A hydraulic PTO or mechanical PTO can be used with OSWEC but for simplicity a hydraulic PTO will be used as an example.
OSWEC には油圧 PTO または機械 PTO が使用できる．例では簡単のため油圧 PTO を使用する
Modeling of OSWEC with Hydraulic PTO The same as RM3 with hydraulic PTO.
油圧 PTO 油圧 PTO と RM3 と同じで OSWEC のモデル化。

Simulink Model
Simulink モデル

The Simulink model can be built as following:
Simulink モデルは以下のように構築される
Step 1: Copy OSWEC tutorial folder to get started $Source¥tutorials¥OSWEC.
ステップ 1：OSWEC チュートリアルフォルダをコピー
Step 2: Open OSWEC.slx file and replace Rotary PTO (Local RY) with Rotational PTO UD Torque (Local RY).
ステップ 2：OSWEC.slx の Rotary PTO (Local RY) を Rotational PTO UD Torque (Local RY) に交換
Step 3: Use a subsystem and rename it to PTO-Sim where input is response and output is torque.
ステップ 3：サブシステムを PTO-Sim に名前変更．入力が応答，出力はトルクである．

Step 4: Go inside PTO-Sim block and drag and drop one bus selector and two selector blocks.
ステップ 4：PTO-Sim ブロック内に 1 つのバスセレクタと 2 つのセレクタをドロップ
Since pitch is driving the piston, selection index of each selector needs to be changed to 5.
ピッチ運動でピストンを駆動するため、各セレクタの選択インデックスを 5 に変更する．
Next, go to PTO-Sim library and drag and drop all the blocks for the hydraulic PTO.
PTO-Sim ライブラリに移動し、油圧 PTO のすべてのブロックをドラッグ＆ドロップ．
The rotary to linear adjustable rod block can be found under rotary to linear conversion box.
回転から直動ロッドブロックは linear conversion box にある．
Step 5: The rotary to linear adjustable rod block takes angular position and velocity from index selector blocks and PTO force from compressible fluid piston block.
ステップ 5：rotary to linear adjustable rod block は，インデックス選択ブロックからの角度と速度，圧縮性流体ピストンブロックからの PTO 力を使う．
The outputs of the rotary to linear adjustable rod block are linear position, velocity, and torque.
rotary to linear adjustable rod block の出力は線形位置，速度，トルクである．
Linear position and velocity are used as inputs for compressible fluid piston and torque is fed back to WEC-Sim.
線形位置及び速度が圧縮性流体ピストン入力に使用される．
トルクの入力が WEC-SIM にフィードバックされる
The rest of the connects are the same as in RM3 with hydraulic PTO.
残りの接続は、油圧 PTO つき RM3 と同じである．
The user is encouraged to go up one level to check the connections between PTO-Sim and WEC-Sim.
PTO-SIM と WEC-Sim の接続を確認するために 1 つ上のレベルに行く
Input File, Simulation, and Postprocessing The same as RM3 with hydraulic PTO.
入力ファイル、シミュレーション、ポストプロセスは油圧 PTO つき RM3 と同じ。

Other PTO-Sim Tutorials
他の PTO-Sim のチュートリアル
Other PTO-Sim tutorials that were not discussed above can be found on the WEC-Sim Applications repository.
他の PTO-Sim のチュートリアルは WEC-Sim のアプリケーションリポジトリにある
PTO-Sim Application , Description
PTO-Sim のアプリケーション、説明
RM3_Hydraulic_PTO , RM3 with hydraulic PTO 油圧 PTO つき RM3
RM3_cHydraulic_PTO , RM3 with compressible hydraulic PTO 圧縮油圧 PTO つき RM3
RM3_DD_PTO , RM3 with direct drive linear generator リニア発電機つき RM3
OSWEC_Hydraulic_PTO , OSWEC with hydraulic PTO (adjustable rod)
OSWEC_Hydraulic_Crank_PTO , OSWEC with hydraulic PTO (crank)

Mooring Features
係留
This section provides an overview of some features included in WEC-Sim's mooring class.
本章では WEC-Sim の係留クラスの関数の概説する.
For more information, refer to Mooring Class.
詳細は係留クラスを参照
Floating WEC systems are often connected to mooring lines to keep the device in position.
浮体 WEC 系は，デバイスを特定位置で維持するために係留線で接続される
WEC-Sim allows the user to model the mooring dynamics in the simulation by specifying the mooring matrix or coupling with MoorDyn.
WEC-Sim はシミュレーション中で係留運動をモデル化する. 係留行列を指定するか MoorDyn と連成する
To include mooring connections, the user can use the mooring block (i.e., Mooring Matrix block or MoorDyn block) given in the WEC-Sim library under Moorings lib and connect it between the body and the Global reference frame.
係留接続を含めるには係留ブロック（係留行列ブロックまたは MoorDyn ブロック）を使用する.
係留ブロックは Mooring lib 下あり、本体とグローバル基準フレームとの間に接続する.
Refer the MoorDyn Tutorial section, and the Mooring Webinar for more information.
詳細は MoorDyn チュートリアルおよび係留ウェビナーを参照
MoorDyn is hosted on a separate MoorDyn repository.
MoorDyn は MoorDyn リポジトリでホストされる
It must be download separately, and all files and folders should be placed in the $Source/functions/moorDyn directory.
これは、個別にダウンロードする必要があり、すべてのファイルとフォルダを $Source/functions/moorDyn に配置する必要がある.

Mooring Matrix
係留行列
When the mooring matrix block is used, the user first needs to initiate the mooring class by setting mooring(i) = mooringClass('mooring name') in the WEC-Sim input file (wecSimInputFile.m).
係留行列ブロックを使用する場合、WEC-Sim 入力ファイル（wecSimInputFile.m）内に
mooring(i) = mooringClass('mooring name')
を設定し係留クラスを初期化する
Typically, the mooring connection location also need to be specified, mooring(i).ref = [1x3] (the default connection location is [0 0 0]).
典型的には、係留接続位置を指定する
mooring(i).ref = [1x3]（デフォルトの接続位置は[0 0 0]）.
The user can also define the mooring matrix properties in the WEC-Sim input file using:
また、使用 WEC-Sim の入力ファイル内の係留行列を定義する
Mooring stiffness matrix - mooring(i).matrix.k = [6x6]
Mooring damping matrix - mooring(i).matrix.c = [6x6]
Mooring pretension - mooring(i).matrix.preTension = [1x6]

MoorDyn
When the MoorDyn block is used, the user needs to initiate the mooring class by setting mooring = mooringClass('mooring name') in the WEC-Sim input file (wecSimInputFile.m), followed by number of mooring lines is defined in MoorDyn (mooring(1).moorDynLines = <Number of mooring lines>)
MoorDyn ブロックを使用する場合、WEC-Sim 入力ファイル（wecSimInputFile.m）内に
mooring = mooringClass('mooring name')
を設定し係留クラスを初期化する.
次に係留ラインの数を指定する

mooring(1).moorDynLines = <Number of mooring lines>
A mooring folder that includes a moorDyn input file (lines.txt) is required in the simulation folder.
moorDyn 入力ファイル（lines.txt）を含む係留フォルダをシミュレーションフォルダに置く

Tutorial: RM3 with MoorDyn
チュートリアル：MoorDyn 付き RM3
This section describes how to simulate a mooring connected WEC system in WEC-Sim using MoorDyn.
本章では、WEC-SIM で MoorDyn を使用した係留接続 WEC システムのシミュレート方法を説明する.
The RM3 two-body floating point absorber is connected to a three-point catenary mooring system with an angle of 120 between the lines in this example case.
RM3 2 物体ポイントアブソーバが三点カテナリー係留系に接続される．係留線間の角度は 120 度
The RM3 with MoorDyn folder is located under WEC-Sim Applications repository.
MoorDyn つき RM3 フォルダは WEC-Sim アプリケーションリポジトリの下にある
WEC-Sim Simulink Model: Start out by following the instructions on how to model the RM3 Two-Body Point Absorber.
WEC-Sim の Simulink モデル：RM3 2 物体ポイントアブソーバのモデル化方法の手順を以下に示す
To couple WEC-Sim with MoorDyn, the MoorDyn Block is added in parallel to the constraint block
WEC-SIM と MoorDyn を連成するため、MoorDyn ブロックが拘束ブロックと並列に付加される
WEC-Sim Input File: In the wecSimInputFile.m file, the user need to initiate the mooring class and define the number of mooring lines.
WEC-Sim の入力ファイル：wecSimInputFile.m 内で係留クラスを初期化し、係留線の数を定義する

```
%% Simulation Data
simu = simulationClass(); %Create the Simulation Variable
simu.simMechanicsFile = 'RM3MoorDyn.slx';   %Location of Simulink Model File
simu.endTime=400;     %Simulation End Time [s]
simu.dt = 0.01;   %Simulation Time-Step [s]
simu.rampT = 40;%Wave Ramp Time Length [s]
simu.mode='accelerator';
simu.explorer = 'off';
simu.dtCITime = 0.05;
%simu.paraview = 1;

%% Wave Information
%% User-Defined Time-Series
waves = waveClass('userDefined');     %Create the Wave Variable and Specify Type
waves.etaDataFile = 'umpqua46229_6_2008.mat';   % Name of User-Defined Time-Series File [:,2] = [time, wave_elev]

%% Body Data
body(1) = bodyClass('hydroData/rm3.h5');
body(1).mass = 'equilibrium';
body(1).momOfInertia = [20907301 21306090.66 37085481.11];
body(1).geometryFile = 'geometry/float.stl';%Location of Geomtry File

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';
body(2).initDisp.initLinDisp = [0 0 -0.21];

%% PTO and Constraint Parameters
constraint(1) = constraintClass('Constraint1');
constraint(1).loc = [0 0 0];%Constraint Location [m]

pto(1) = ptoClass('PTO1');
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]

mooring(1) = mooringClass('mooring');
mooring(1).moorDynLines = 6;
mooring(1).moorDynNodes(1:3) = 16;
mooring(1).moorDynNodes(4:6) = 6;
mooring(1).initDisp.initLinDisp = [0 0 -0.21];
```

MoorDyn Input File: A mooring folder that includes a moorDyn input file (lines.txt) is created.
MoorDyn 入力ファイル：moorDyn 入力ファイル（lines.txt）を含む係留フォルダが作成される.
The moorDyn input file (lines.txt) is shown in the figure below.
moorDyn 入力ファイル（lines.txt）は以下の図に示される
More details on how to setup the MooDyn input file were described in the MoorDyn User Guide [1].
MooDyn 入力ファイルの詳細は MoorDyn ユーザーガイド参照[1]。

Simulation and Post-processing: Simulation and post-processing are the same process as described in Tutorial Section.
シミュレーションとポストプロセッシング：シミュレーションとポストプロセッシングは、チュートリアルと同じ.
Note 注意
You may need to install the MinGW-w64 compiler to run this simulation.
シミュレーション実行には MinGW-W64 コンパイラをインストールする必要がある.

Visualization/Paraview　Paraview 可視化
This section describes how to setup, output, and use Paraview files for visualizing a WEC-Sim simulation.
本章では、WEC-Sim シミュレーション可視化のための Paraview ファイル設定、出力、使用を説明する
For more information about using Paraview for visualization, refer to the visualization webinar, and the WEC-Sim Applications repository.
Paraview 可視化方法の詳細は可視化のウェビナーと WEC-Sim のアプリケーションリポジトリを参照
Using Paraview visualization improves on SimMechanics's explorer by:
Paraview の可視化により、SimMechanics の探検に向上する.
Visualization of the wave field
波の可視化
Visualization of the cell-by-cell non-linear hydrodynamic forces (when using non-linear hydro)
セルごとの非線形流体力の可視化（非線形流体力を使用する場合）
Allow data manipulation and more visualization options
データ操作と可視化オプション
On the other hand the SimMechanics explorer shows the following information not shown in Paraview visualizing:
一方，SimMechanics エクスプローラは、Paraview には無い情報をもつ
Location of the center of gravity
重心の場所
Location of the different frames including PTO and Constraint frames
フレームの場所(PTO と拘束フレームを含む)
Visualization with paraview requires many files to be written, which makes the WEC-Sim simulation take significantly more time, and makes the directory significantly larger.
ParaView 可視化は多くのファイルに出力するため，WEC-Sim のシミュレーションに時間がかかる．また，ディレクトリが非常に大きくなる
It should only be turned on when visualization is desired.
可視化が必要な場合のみオンにする.
The user also needs to have some familiarity with using Paraview.
また，Paraview の知識が必要.

Getting Started - Installation
はじめに - インストール
You will need to install Paraview and install and setup Python.
Paraview をインストールし Python をインストール，設定する.
Next you will need to add some WEC-Sim specific macros,as follows:
次に WEC-Sim の固有のマクロを追加する.

Open Paraview
Click on Macros => Add new macro
Navigate to the WEC-Sim source/functions/paraview_macros directory
Select the first file and click OK
Repeat this for all files in the paraview_macros directory.

Setting Up Paraview Output
Paraview 出力の設定
The following table shows the variables that can be specified in the wecSimInputFile to control the Paraview visualization.
次の表は，Paraview 可視化を制御する wecSimInputFile 内の変数を示す.
The body.viz properties are also used in the SimMechanics explorer visualization.

body.viz プロパティは、SimMechanics エクスプローラ可視化に使用される.

WEC-Sim Visualization using Paraview
Paraview を使用した WEC-Sim の可視化
Variable , Description
変数, 説明
simu.paraview , 0 to not output Paraview files [default] 1 to output Paraview files0:Paraview ファイルを出力しない（デフォルト), 1:出力する.
simu.nlHydro , 0 for no non-linear hydro [default] 1 for non-linear hydro with mean free surface 2 for non-linear hydro with instantaneous fs
0:非線形流体（デフォルト), 1:非線形流体と平均水位, 2:非線形流体と瞬時水位
simu.domainSize , size of ground and water planes in meters [default 200]
地面と水面のサイズ(m)（デフォルト:200）
simu.dtOut , simulation output sampling time step [default dt]
シミュレーション出力サンプリング時間ステップ[デフォルト:dt]
body(i).viz.color , [RGB] body color [default [1 1 0]]
船体カラー[デフォルト[1 1 0]
body(i).viz.opacity , body opacity [default 1]
船体透明度［デフォルト 1]
waves.viz.numPointsX , wave plane discretization: number of X points [default 50]
波面の分割：X 方向の数[デフォルト 50]
waves.viz.numPointsY , wave plane discretization: number of Y points [default 50]
波面の分割：Y 方向の数[デフォルト 50]

Outputs and Opening in Paraview
出力と Paraview
When simu.paraview is set to 1, a directory called vtk is created.
simu.paraview を 1 に設定すると、VTK というディレクトリが作成される.
All files necessary for Paraview visualization are located there.
そこに Paraview 可視化に必要なすべてのファイルがある.
To view in Paraview:
Paraview で表示するには：
Open the vtk/filename.pvd file in Paraview
Paraview から VTK/filename.pvd ファイルを開く
Click Apply
Apply をクリック
With the model selected in the pipeline, run the WEC-Sim macro
パイプラインで選択したモデルでは、WEC-Sim マクロを実行する
Move the camera to desired view
所望の視点にカメラを移動
Click the green arrow (play) button
緑色の矢印（再生）ボタンをクリック

The WEC-Sim macro:
WEC-Sim マクロ：
Extracts each body, sets the color and opacity, and renames them
船体を抽出し、色と不透明度を設定し、名前を変更する
Extracts the waves, sets color and opacity, and renames
波を抽出し、色および不透明度を設定し、名前を変更する
Creates the ground plane
グランドプレーンを作成する.

Sets the camera to parallel view
カメラをパラレルビューに設定する.

Basic Visualization Manipulation
基本的な可視化の操作
After opening the .pvd file and running the WEC-Sim macro
you can do a number of things to visualize the simulation in different ways.
.pvd ファイルを開いて WEC-Sim のマクロを実行した後，シミュレーションをさまざまな方法で可視化できる
You can color waves and bodies by any of the available properties and apply any of the Paraview filters.
プロパティで波と船体を着色し，Paraview フィルタを適用できる
The video below shows three different views of the OSWEC model described in the tutorials.
下のビデオは、チュートリアルで説明された OSWEC モデルの三つの異なる可視化を示す.
On the bottom right view the wave elevation is used to color the free surface.
下右ビューでは，水面上昇が自由表面を着色する
The top right view uses the slice filter.
右上のビューは、スライスフィルタを使用する

Visualizing Non-Linear Hydro Forces
非線形流体力の可視化
When using non-linear hydro the paraview files also contain cell data for the bodies.
非線形流体力を使用する場合，ParaView のファイルも船体セルデータをもつ
The cell data are:
セルデータは以下のとおりである.
Cell areas
セル面積
Hydrostatic pressures
静水圧
Linear Froude-Krylov pressures
線形フルードクリロフ圧力
Non-linear Froude-Krylov pressures
非線形フルード・クリロフ圧力
The pressureGlyphs macro calculates cell normals, and cell centers.
pressureGlyphs マクロは、セルの法線、およびセルの中心を計算する.
It then creates the following glyphs:
その後、次のグリフを作成する.
Hydrostatic Pressure
静水圧
Linear Froude-Krylov pressure
線形フルード・クリロフ圧力
Non-linear Froude-Krylov pressure
非線形フルード・クリロフ圧力
Total pressure (hydrostatic plus non-linear Foude-Krylov)
全圧（静水圧＋非線形フルード・クリロフ）
Froude-Krylov delta (non-linear minus linear)
フルード・クリロフデルタ（非線形 - 線形）

The video below shows three different views of the RM3 model described in the tutorials.
下のビデオは、チュートリアルで説明された RM3 モデルの 3 つの異なるビューを示す。
The top right shows glyphs of the non-linear Froude-Krylov pressure acting on the float.
右上は、フロートに作用する非線形フルード・クリロフ圧力のグリフを示す.

The bottom right shows the float colored by hydrostatic pressure.
右下は静水圧によって着色されたフロートを示す.

Decay Tests 自由減衰
When performing simulations of decay tests, you must use one of the no-wave cases and setup the initial (time = 0) location of each body, constraint, PTO, and mooring block.
減衰テストのシミュレーションを行うには、無波ケースを選び，初期（time=0）における各船体の位置、拘束、PTO、係留ブロックを設定する必要がある
The initial location of a body or mooring block is set by specifying the CG or location at the stability position (as with any WEC-Sim simulation) and then specifying an initial displacement.
船体初期位置，係留ブロックの初期位置は、CG や安定位置に設定し，初期変位を指定する（任意の WEC-Sim シミュレーションと同様）
To specify an initial displacement, the body and mooring blocks have a .initDisp property with which you can specify a translation and angular rotation about an arbitrary axis.
初期変位を指定するには、船体と係留ブロックは .initDisp 性質をもつ.
任意の軸の周りの並進と回転角度を指定することがでる.
For the constraint and PTO blocks, the .loc property must be set to the location at time = 0.
拘束及び PTO ブロックでは，時刻 0 位置における.loc を設定する必要がある.
There are methods available to help setup this initial displacement for all bodies, constraints, PTOs, and moorings.
船体，拘束，PTO，係留の初期変位設定を支援する方法がある.
To do this, you would use the body(i).setInitDisp(...);, constraint(i).setInitDisp(...), pto(i).setInitDisp(...), and mooring(i).setInitDisp(...) method in the WEC-Sim input file.
これを行うには、WEC-Sim 入力ファイルに body(i).setInitDisp(...); constraint(i).setInitDisp(...), pto(i).setInitDisp(...), mooring(i).setInitDisp(...) を記述する.
A description of the required input can be found in the method's header comments.
必要な入力の記述は、メソッドのヘッダコメントにある
Note that body(i).cg, constraint(i).loc, pto(i).loc, and mooring.ref must be defined prior to using the object's .setInitDisp method.
body(i).cg, constraint(i).loc, pto(i).loc, mooring.ref は .setInitDisp の前に定義する.
For more information, refer to the WEC-Sim Applications repository IEA OES Task 10 example.
詳細は WEC-Sim アプリケーションリポジトリ IEA OES タスク 10 の例を参照。

[End of File]