Run OF User’s Guide

v1.1

By Sam Kanner

Released: Jan 6 2017

Last Updated: Jul 3 2017

# Purpose

This guide describes the ‘Run OF’ code suite developed for numerical analysis for Principle Power Inc. Does ‘Run OF’ stand for Run OrcaFlex, or does it stand for Run OrcaFAST? It stands for both! Using a simple Excel table interface and MATLAB functions, the user can setup a design load case in either software program. Furthermore, the runs can be queued with multiple MATLAB windows, allowing for maximum utilization of a workstation’s cores. If a run is successfully completed, a post-processor extracts the relevant time series and statistics, allowing for quick, easy analysis of the results.

# How to make your first OF run:

1. Get the Code
   1. Get on GitHub
   2. Become a part of PPI-Eng
   3. Download SourceTree
   4. Clone the repos (‘handy\_codes2share’ + ‘WindFloatX’) to a folder on your MATLAB path
2. Copy and rename the DLC workbook example: ‘Secondary\_Codes\RunOF\DLCspreadsheet.xlsx’ somewhere on your MATLAB path
   1. Change the ‘Introduction’ sheet or any other sheet to include your OrcaFlex .dat file
   2. Make any other changes to the run. See DLC Workbook section for more information.
3. Open MATLAB
   1. Use the code: RunOF\_BATCH(xlsfile, numMATLABs, nSheet). See Code Structure for more details.

# WindFloat Geometry

All ‘high-level’ WindFloat parameters are defined in the WindFloatX (PlatformX and TurbineX) spreadsheets. The proper amount of information should be entered into this spreadsheet so that the spreadsheet is not too ‘heavy’ but can be used to adequately define a certain design. The spreadsheet should not contain granular information (such as in the weight spreadsheet) that can change quite frequently. Rather, the spreadsheet should contain information, such as beam thickness, turbine section weight, etc that is more stable and changes on the timescale of months. Defining the geometry under version control allows the user to dynamically switch between projects easily. Furthermore, the user is able to work on multiple design iterations of a single project using Git branches. For more information on branches and how to checkout other branches see Git-specific help guides.

If the user would like to make a change to the Platform or Turbine geometry it is required to update the ‘.mat’ file associated with the geometry. In order to do so, use the optional argument to read the Excel spreadsheet and re-write the ‘.mat’ file:

* Ptfm = getMyPtfm(‘(insert project name)’,**1**);
* Turbine = getMyTurbine(‘(insert project name)’,**1**);

# Code Structure

This section will introduce the functions and sub-functions that are used to setup, execute and post-process OF runs.

***RunOF\_BATCH****(xlsfile, numMATLABs (optional), nSheet (optional) )*

This is the main code the user interacts with from the MATLAB command line. The user creates a DLC Excel worksheet with details of each specific design load case he or she wishes to run. An example of a DLC worksheet is on the repo and the excel workbook will be described in the next section.

*xlsfile [string] = absolute path to the Excel “DLC workbook” (if not entered a gui will open)*

*numMATLABs [integer] = number of MATLAB windows to open (default = number of processors of the workstation*

*nSheet = the number sheet in the Excel workbook your runs are on (optional). Default is to read in all of the sheets, which can take a while. Thus, this only aids to speed up the function*

The result of the function is to temporarily save a number of matfiles (equal to the number of MATLAB windows to open) in the user’s home directory that can be subsequently read by the new MATLAB windows to be launched.

***RunOF\_Queue****(matfile)*

This function reads in a matfile containing information relating to all of the runs to be read in a specific MATLAB instance. It goes through each run one-by-one and documents the progress in a small gui.

***RunOF****(matfile)*

This function takes in a number of parameters that are either defined in the current DLC spreadsheet or in the ‘Introduction’ worksheet, which provides the defaults for all of the queued runs. The main purpose of this function is to compile all of the different parameters into simple MATLAB structures (Wind, Wave, General, etc) that can be passed back and forth between sub-functions. The other main purpose of this function is to set the parameters that are not specified in the DLC run worksheet to the default value in the ‘Introduction’ worksheet. If the parameters are not set in either of these cases then they are set to the default and a line of text will be displayed in the MATLAB command window.

***PreOF****(many structures)*

This function takes the structures created by RunOF and sets up the current run. It opens up the OrcaFlex model (\*.dat file) and makes modifications to the model. It also finds the specified wind file (.hh or .wnd) that can be read by FAST. If the specified file does not exist it will run TurbSim.

***ExecuteOF****()*

This function simply calls OrcaFlex or FAST to run the simulation, depending on the type of run specified by the user.

***PostOF****()*

If the run is successfully completed, then this post-processor is called. The post-processor moves the requested files from the temporary folder to the final run folder. It also examines the \*.sim file created by OrcaFlex as well as the \*.outb final generated by FAST (if it is an OrcaFAST run). If requested, the post-processor creates a file called outputs.mat and stats.mat, which contain time-series and simple statistics of a multitude of variables. The matfile outputs.mat is used by many other processes, including MatchMaker.

***extractSIM****(simfile)*

This sub-function is called by ***PostOF*** to extract relevant data from the OrcaFlex \*.sim file. If the user needs more data from the \*.sim file then currently provided, then this function must be modified.

# DLC Workbook

There are at least two spreadsheets that must be filled out in order to use the ‘Run\_OF’ code capabilities. The first spreadsheet in the workbook must be called ‘Introduction’ and should include many of the default variables, which are used for every run. The second spreadsheet is the DLC run spreadsheet (i.e., “Model Validation” or “1.6”), which specifies the specific parameters for each run.

The following variables can be set in either the ‘Introduction’ tab or in the DLC run spreadsheet. If the user wants to use the same value for the parameter throughout all runs, then they should set it in the ‘Introduction’ tab. If they want to vary the parameter run by run, then it should be set in the run spreadsheet and varied accordingly.

**The controlling value is the one set in the DLC run spreadsheet. For example, if the user sets a value for the *Datfile* in the ‘Introduction’ tab and the DLC spreadsheet, then the value in the DLC spreadsheet (i.e., “1.6”) will be used.**

***Introduction Spreadsheet***

**Global Parameters**

*RunFolder* [string] = (Required) The relative name of the folder to store the simulations. The main directory is found to be the one above the Input\_Orcaflex folder. The code will create this folder in that directory if it does not exist already.

*RunPrefix* [string] = (Required) The word that is appended to the beginning of all *Runnames.* Since most *Runname*s are strings of integers, this allows for some description (normally, just use ‘Run’). There are some ‘special’ runnames, which unlock additional features of the code (see the subsection call ‘Special Runnames’).

*OrcaFlexVer* [string] = (optional) A text string with a number corresponding to the desired OrcaFlex version the user wants to run. This version of OrcaFlex must exist on the user’s computer. In order for this functionality to work MATLAB must be run with administrator privileges. (You can setup a shortcut on your desktop to always run MATLAB with administrator privileges.)

**Platform Parameters**

*PlatformName* [string] = (Required) This string refers to the matfile/Excel file that is stored in the WindFloatX repository describing the main characteristics of a platform. The matfile contains a structure which is called by many of the functions and sub-functions.

*Datfile* [string] = (Required) This string refers to the default OrcaFlex datfile to be used to run the simulations. In order to follow PPI standards, the OrcaFlex file should be in a folder in the main run directory called ‘Input\_OrcaFlex’. If left unset, the user must specify the datfile for every simulation chosen in the *Datfile* column of the ***DLC Run Spreadsheet.*** If the *Datfile* is left empty, then the code reverts to datfile specified in this cell.

*Orientation* [integer] = (Required) The platform orientation in degrees in the OrcaFlex convention (i.e., North is 0 degrees and increases CCW). If the *Orientation* is different than the InitialYaw of the Vessel in the *Datfile*, then the entire model will be rotated.

*K* [1x6 vector of floats] = (Required) The diagonal entries of the linear hydrostatic stiffness matrix of the platform, including the mooring system. The 3,4,5 terms are only used if an equivalent OrcaFlex VesselType cannot be found. (The naming convention is to name a VesselType with ‘MyName\_FAST’ for the OrcaFAST vessel and ‘MyName’ for the OrcaFlex vessel.) The 1,2,6 entries for the mooring are used regardless. This matrix is used when determining the applied force that is necessary to achieve the *X\_des* during free-decay tests (either OrcaFlex or OrcaFAST).

**Turbine Parameters**

*TurbineName* [string] = (Required) This string refers to the matfile/Excel file that is stored in the WindFloatX repository describing the main characteristics of a turbine. The matfile contains a structure which is called by many of the functions and sub-functions.

*TurbSim\_Name* [string] = (Required) This string refers to the name of the TurbSim executable located in the Wind\TurbSim folder. Generally set to ‘TurbSim.exe’ or ‘TurbSim64.exe’.

*TurbineClass* [integer] = (optional) This number refers to the class of the Turbine, as defined in IEC-61400-1 standard, which sets the reference wind speed for ETM.

**Environmental Parameters**

*Wave\_spectrum\_file* [string] = (optional) This string contains the absolute path to a .mat file that contains a matrix with data pertaining to a ‘User-specified’ wave spectrum. The .mat file must contain a single matrix (the name of the matrix does not matter). The data in the matrix must match that found in the data entry in OrcaFlex when ‘User-specified’ spectrum is chosen.

*Wind\_shear* [float] =(optional) The wind shear value is only used if generating .wnd or .hh using TurbSim. It is written to the .inp file which is called by TurbSim.

*Wind\_IECtype* [string: ‘NTM’, ‘EWM’ or ‘ETM’] = (optional) The type of turbulence model in the IEC-61400-1 standard specifies the equation that should be used when calculating the standard deviation of the turbulence intensity (Eqs 11 and 16) for instance.

*Wind\_TIchar* [string: ‘A’, ‘B’, or ‘C’] = (optional) The turbulence intensity characteristic is described in the IEC-61400-1 standard concerning wind turbines. For the Normal Turbulence Model (‘NTM’) the values [‘A’,’B’,’C’] correspond to turbulence intensity values of [0.16, 0.14, 0.12], respectively.

*Wind\_INPname* [string]: The name of the template of the input file to be used to run TurbSim on the fly to generate turbulent wind fields. The default is ‘TurbwindTemplate.inp’. This file must be located in the UserMainFolder\Wind\TurbSim\ folder.

**Free-Decay/Force-Excursion Parameters**

*X\_des* [nx6 vector of floats] = (Required) This vector corresponds to the max desired displacements for the free-decay tests. The number of rows corresponds to the number of iterations of the test the user would like to perform. The column index is the degree of freedom of the vessel. If set to NaN, the code skips over the specific iteration. To run the free-decay tests, use the special word ‘FDecays’ as a *Runname.*

*F\_des* [1x6 vector of floats] = (Required for a force-excursion curve) This vector corresponds to the multiplier to be used on the vector: [MaxThrust, MaxThrust, Force to get to X\_des(2,3), Moment to get to X\_des(2,4), Moment to get to X\_des(2,5), MaxThrust\*Ptfm.Col.Lh]; which is the maximum applied force used during the Force-Excursion curves tests. The test steps the force up in a staircase fashion to reduce necessary settling time. To run the force-excursion tests, use the special word ‘FExcrns’ as a *Runname.*

*t\_LAL* [1x2 vector of floats] = (Optional) This vector specifies the time to start and the ramp of the locally applied force, respectively. The locally applied force magnitude and point of application is specified in the worksheets under LAL\_F and LAL\_X, respectively.

*t\_GAL* [1x2 vector of floats] = (Optional) This vector specifies the time to start and the ramp of the globally applied force, respectively. The globally applied force magnitude and point of application is specified in the worksheets under GAL\_F and GAL\_X, respectively.

***DLC Run Spreadsheet***

The DLC run spreadsheet is organized as follows: each row represents a specific simulation the user would like to queue; each column represents a certain variable that is used in either the OrcaFlex or FAST simulations. If the cells are left blank, then a default value is used. The next section describes what each variable represents.

**General Parameters**

*Runname* [string] = (Required) The name of the simulation. Generally a string of integers to detail the DLC number, family number, seed number, etc. Appended to *RunPrefix* to define the complete name. Special names include: ‘RAOs’, ‘FDecays’ and ‘FExcrns’, to run RAOs, Free-Decay simulations and Force-Excursion curves, respectively.

*Inputcode* [string] = (Required for strength run, left unset for fatigue run) The state of the turbine (e.g., ‘POW’ or ‘PAR’). If left unset for a strength run, the state of the turbine is determined by whether the wind speed is within the operating conditions of the turbine.

*Nacyaw* [float] =The deviation of the turbine nacelle away from the incident wind direction in degrees.

*Output\_Flag* [integer] = The type of outputs that should be generated during the post-processing routines.

* Set to 1, this saves the timeseries results in a matfile (Default).
* Set to 2, this saves the timeseries results in a matfile for a flexible model, which does not have a Vessel.
* Set to 3, this saves each timeseries into a .dat file in an Outputs folder. Implemented for backwards compatibility with previous RunOF versions.

*OutputStats\_Flag* [logical] = Set to 1 a stats.mat file is generated, taking the min, mean, max and stdev of the outputs in between *CutInTime* and *CutOutTime*.

*FAST\_Flag* [integer] = The type of software to be used during the simulation:

* Set to 0, this will setup an OrcaFlex simulation.
* Set to 1, this will setup an OrcaFAST simulation, with a floating platform.
* Set to 2, this will setup a FAST simulation, with a fixed platform.

*Datfile* [string] = Absolute path of the Orcaflex model (\*.dat) to be used during specific run. If set, this file is used instead of the *datfile\_default* set in the Introduction.

*Run\_Flag* [integer] =The method of running the simulation, or simply re-post-processing.

* Set to 1, this runs the simulation.
* Set to 2, this post-processes an existing run. The run folder must already exist and contain a \*.sim and an \*.outb.

*Save\_Sim* [logical] = Set to 0, this will NOT save the OrcaFlex simulation file (\*.sim). Set to 1, this will save an OrcaFAST simulation file. This functionality is implemented in order to save hard drive space.

**Wind Parameters**

*Wind\_Dir* [float] = Wind direction in OrcaFlex convention in degrees. Convention is defined as: **Directions for waves, current and wind are specified by giving the direction in which the wave (or current or wind) is progressing, relative to global axes, measured positive from the x-axis towards the y-axis (CCW).**

*Wind\_Speed* [float] =Wind speed in m/s.

*Wind\_Type* [integer] = Specifies type of Wind.

* Set to 0, this represents a constant wind speed. If running OrcaFAST the code is seeking a file called: ‘Wind\SteadyWind\steady\_wind#.wnd’.
* Set to 1, this represents a turbulent wind field and is seeking a .hh or .wnd file called ‘TurbSim/turbwind%02dx%02d\_# where the Wind Grid size and Wind Speed are used.
* Set to 2, the code seeks a file called ‘Wind\IECWind\_v510\Wind\_Seed'

*Wind\_Seed* [string or integer] = Wind seed that is used to call the proper wind file (e.g., ‘A’, ‘B’, ‘C’… etc)

*Wind\_stdTI* [float] = The standard deviation of the turbulence intensity of the wind. If the value is set, it overrides all other parameters regarding the determination of the turbulence intensity (TIchar, IECtype, etc). See the function ‘runTurbSim’ for more information.

*Wind\_Grid* [integer] = The size of the TurbWind grid (square) to use in meters.

**Wave Parameters**

*Wave\_Dir* [integer] = Wave direction in OrcaFlex convention in degrees. If using a uni-modal spectra this is the wave direction. If using a bi-modal spectra, this is the ‘wind-sea’ wave direction. Convention is defined as: **Directions for waves, current and wind are specified by giving the direction in which the wave (or current or wind) is progressing, relative to global axes, measured positive from the x-axis towards the y-axis (CCW).**

*Wave\_Hs* [float] = Significant wave height (Hs) in meters. If using a uni-modal spectra this is the wave height. If using a bi-modal spectra, this is the ‘wind-sea’ wave height.

*Wave\_Tp* [float] = Significant wave period (Tp) in seconds. If using a uni-modal spectra this is the wave period. If using a bi-modal spectra, this is the ‘wind-sea’ wave period.

*Wave\_Gamma* [float] = ‘Peakness’ factor of the JONSWAP spectrum.

*Wave\_Seed* [integer] = Random seed number used in generating the free-surface.

*Swell\_Dir* [integer] = Swell Wave direction in OrcaFlex convention. If using a uni-modal, this can be left blank or set to 0.

*Swell\_Hs* [float] = Swell wave height in meters. If using a uni-modal, this can be left blank or set to 0.

*Swell\_Tp* [float] = Swell wave period in seconds. If using a uni-modal, this can be left blank or set to 0.

*Swell\_Gamma* [float] = ‘Peakness’ factor of the swell JONSWAP spectrum.

*Swell\_Seed* [integer] = Random seed number used in generating the free-surface of the swell.

**Current Parameters**

*Cur\_Dir* [float] = Current direction in OrcaFlex convention in degrees. If unset during a fatigue run this is automatically set to the *Wind\_Dir*. Convention is defined as: **Directions for waves, current and wind are specified by giving the direction in which the wave (or current or wind) is progressing, relative to global axes, measured positive from the x-axis towards the y-axis (CCW).**

*Cur\_Speed* [float] =Current speed in m/s. If unset during a fatigue run, the speed is automatically set to 3% of *Wind\_Speed*.

**External Forces/Moments Parameters**

*GAL\_X* [1x3 vector, floats] = Location of application of Globally Applied Force, in the local coordinates of the vessel.

*GAL\_Mag* [1x6 cell, floats or strings] = Magnitude or name of Globally Applied Force in each DOF. If one of the cells is set to a string, the External Force timeseries must exist in the OrcaFlex model.

*LAL\_X* [1x3 vector, floats] = Location of application of Locally Applied Force, in the local coordinates of the vessel.

*LAL\_Mag* [1x6 cell, floats or strings] = Magnitude or name of Locally Applied Force in each DOF. If one of the cells is set to a string, the External Force timeseries must exist in the OrcaFlex model.

NOTE: If running OrcaFAST, where the Wind\_Speed is nonzero and the Ballast\_Flag is set to 0 a locally applied force will be created and applied at the turbine’s center of pressure, equal and opposite to the thrust force.

**Mooring Break Parameters**

*MoorBreak\_MLnumber* [integer] = Number corresponding to the mooring line the user wishes to break. The code seeks a line called ‘ML#’ and will break during the simulation from EndA.

*MoorBreak\_time* [float] = The time at which the mooring line breaks.

**Simulation Time Parameters**

*Runtime* [integer] = The total length of the simulation time in seconds.

*CutInTime* [integer] = The time in seconds at which the transient period ends. A time series is taken between *CutInTime* and *CutOutTime,* which represents the actual simulation. This truncated time series is used in calculating the mean yaw, which gets used to rotate some of the results. Also, the time series used to cut out the transient in stats.m and many other functions, such as seed selection algorithm.

*CutOutTime* [integer] = The time in seconds at which the truncated simulation period ends. If unset, the default is the end of the simulation, specified by *Runtime.*

*Time\_Origin* [integer] = The ‘Simulation Time Origin’ in seconds, which can be used to shift the phase of the incident waves.

**Ballast Parameters**

*Ballast\_Flag* [integer] = The type of ballast controller to be used in the simulation.

* Set to 0, the code seeks to use a locally applied moment or a globally applied moment, *LAL\_Mag or* *GAL\_Mag* (Default).
* Set to 1, this uses the ballast controller in the newest version of FAST.exe.
* Set to 2, this calculates a righting moment based on the wind speed and the turbine’s thrust curve (used for fatigue runs).

**Mooring Setup Parameters**

*ML\_Length* [nML x nSec cell of floats] = Can overwrite the lengths of each of the segments of the mooring lines.

*ML\_PreTension* [1x nML vector] = Can utilize the Line Setup Wizard to modify the lengths of initial line segment in order to alter the pretension in the lines.

*ML\_Type* [nML x nSec cell of strings] = Can overwrite the line types of each of the segments of the mooring lines. The line types must already exist in the \*.dat file and must exactly match the names of the existing line types.

## **Special Runnames**

This section describes how a special ***Runname*** can unlock different features of the RunOF code suite.

### RAOs

The RAOs code name will automatically expand the list of vectors in the ‘Wave\_Dir’ and ‘Wave\_Tp’, so that each wave period is run for each wave heading for the given wave height. The syntax of the vectors must follow that of MATLAB. In order to unlock this feature, the Example cells under these parameters must also include an example vector (with square brackets). After the runs are done, use the code **writeDot4Files(RunFolder),** where the first input is a string that corresponds to the absolute path of the RunFolder (e.g., ‘D:\YourRunFolder\Runs\’). The optional inputs are described in the help section of the function. Once the **.4** file is created, use **comparev4** in order to plot the results and compare them to WAMIT RAO files, or other ones created by this code.

### FDecays

The FDecays (more colloquially as Free-Decay) code name automatically expands the free-decay tests into 6DOF, with multiple iterations as specified in the **X\_des**  cell in the Introduction tab. The maximum force applied on the body is the value in X\_des multiplied by the stiffness diagonal matrix **K.** The code applies a ramp functionas the forcing function and then releases the body at 400s. Using **PlotMySims** on this run creates two figures, one with a time history of the free-decay, the other with a plot of the log decrement and an approximation of the ‘zero-amplitude’ damping.

### FExcrns

The FExcrns (more colloquially known as Force-Excursion) code name automatically expands the force-excursion tests into 6DOF, with a forwards and backwards iteration for each DOF. The maximum force applied in each DOF is F\_des.\* [MaxThrust, MaxThrust, Force to get to X\_des(2,3), Moment to get to X\_des(2,4), Moment to get to X\_des(2,5), MaxThrust\*Ptfm.Col.Lh]. The forcing function is a staircase function, with 5 steps and the displacement is the average of the displacement during the entire step. Using **PlotMySims** on this run creates a figure, with a linear and cubic fit to the displacement.

### ShtDwns

The ShtDwns (more colloquially known as Turbine Shutdowns), expands the Wind\_Dir vector and runs a simple turbine shut-down case (to estimate the maximum pitch/roll angle, for instance) at various directions. The simulation consists of applying the thrust force at the specified wind speed and then suddenly cutting the force at time = CutInTime. Using **PlotMySims** on this run creates a figure of the time history of the roll or pitch for the time = [CutInTime, CutOutTime].