Handy Codes User’s Guide

v1.1

By Sam Kanner

Released: Sep 15 2017

Modified: Oct 15 2017

# Purpose

This guide describes all of the wonderful functions available to the user in the ‘Handy Codes’ repository. The codes described in this User’s Guide and contained in the repository can help the user setup, run, post-process, plot and analyze many different load cases in OrcaFlex or OrcaFAST. The codes are divided into 4 categories: ‘Model Setup + Pre-processing’ to build OrcaFlex models and plot WAMIT coefficients, ‘Code Execution’ to run OrcaFlex or OrcaFAST, ‘Simple Post-Processing’ to quickly see time series and statistics from simulations, and ‘Complex Post-Processing’ to run seed selection and fatigue analysis on families of runs.

# Main Codes

## Model Setup + Pre-processing

* ***buildWF(****iptfile)*
  + This code builds or modifies an OrcaFlex model (either rigid or flexible) and saves it to a location specified in the input file. See BuildWF User’s Guide for more detailed information on using this function.
* ***comparev4****(‘\\my\path\to\my\favorite\wamitfile’)*
  + This function plots files written in the form of WAMIT output files. In general, WAMIT frequency-dependent output files take the form of ‘\*.1, \*.2, \*.4, \*.rao, \*.8, \*.9. The user can create a comma separated list with absolute locations of the filenames to plot the outputs on top of each other even if the files have a different set of periods or wave headings.

## Code Execution

* ***RunOF\_BATCH****(xlsfile, numMATLABs (optional), nSheet (optional) )* 
  + This is the main code to run OrcaFlex or OrcaFAST. Requires a DLC spreadsheet that specifies the run parameters. See RunOF User’s Guide for more detailed information on using this function.

## Simple Post-Processing

* *MySimList =* ***createSimList****(xlsfile, nSheet (optional) )*
  + This function takes a DLC spreadsheet and (optional) sheet number and creates a cell array of the run names. The user-interface allows the user to select whether to create a list using stats.mat (for faster loading and plotting of the statistics) or outputs.mat (for timeseries).
* ***plotMySims****(‘\\my\path\to\my\favorite\outputs.mat’)*
  + This function can take the outputs from a simulation or simulations and plot them over time or plot the statistics (i.e., min/mean/max/std) against 2 each other. A simple set of questions allows the user to select the transient period and whether to plot an FFT of the timeseries. This function also recognizes special filenames, such as Free-Decays (‘FDecay’) and Force-Excursion (‘FExcrn’) and will automatically plot the relevant quantities. See RunOF User’s Guide for more information on the ‘Special’ runs.
* ***plotMySimsIn2D****(MyStatsDotMatSimList, ‘varX’,’varY’,StatsType)*
  + This function can plot the correlation between various sensors (e.g., ‘varX’ vs ‘varY’) where the names of the input variables must be names of sensors in the stats.mat. If the sensors have multiple DOFs, then user should append a number corresponding to the DOF of interest to the sensor name. The ‘StatsType’ refers to the type of statistical quantity the user would like to plot: ‘Mean’, ‘Max’, ‘Min’, or ‘Std’ (case-sensitive). For instance, to see correlation between roll and wind speed use ***plotMySimsIn2D****(MySimList, ‘windspeed,’motions4’,’Mean’)*
* ***SaveAllFig(****savedir)*
  + This function saves all open figures of a MATLAB instance into a directory specified by *savedir* and names them according to the figure name.

## Complex Post-Processing

* ***seedSelection\_BATCH(****iptfile)*
  + This function can take a list of a family of completed runs and determine the representative seed for a certain sensor. A representative seed is determined to be the seed whose extreme value (maximum or minimum, as specified by the user) is closest to the mean extreme value of all the seeds. A ‘Statistics’ folder is created in the main Run Folder, with subfolders for the MATLAB files and text files that are created and saved during the execution of the file. For more information on the input file see the **Input Files** section.
* ***plotFigs4Reports(****iptfile)*
  + This function plots a figure for each specified sensor displaying the statistics of representative seeds in a certain family. In general, the families of runs are grouped together either by DLC number or by type of safety factor involved. For more information on the input file see the **Input Files** section
* ***getFatigue(****iptfile)*
  + This function uses a rainflow counting program to estimate the fatigue damage (and hence fatigue life) on various parts of the structure, such as the mooring lines and the tower-base.

# Input Files

This section provides further details on the functions which require input files.

1. ***seedSelection\_BATCH***
   1. *mainDLCs* [1xnFamily vector of scalars] = Numerical vector containing a list of family names of simulations. The seed number (usually 2 digits) is not included in the family names.
   2. *runPrefix* [string] = A string that matches the runPrefix in the DLC spreadsheet. It is the name pre-pended to the run number.
   3. *rundir* [string] = The absolute path to the directory where the run folders are stored. It should be equivalent to the RunFolder (defined relatively) in the DLC spreadsheet.
   4. *tdir* [string] = The absolute path to the directory where the tables of seed selection numbers will be stored.
   5. *statdir* [string] = (optional) The absolute path to the directory where the matfiles of seed selection data will be stored.
   6. *vars2seed* [1 x nVars cell array of strings] = list of variables in the outputs.mat which to base seed off of. For simplicity, only 1-D variables should be used, such as ‘motions[1-6]’, ‘basebendRXY’, etc., although multi-dimensional sensors are allowed.
   7. *how2select* [1 x nVars cell array of strings] = a list of strings, either ‘min’ or ‘max’ referring to which extreme the user would like to compare. (Must be same length as vars2seed)
   8. *iplot* [logical] =If set to TRUE (1), a plot is created for each variable, showing the timeseries of the sensor for all seeds and their extreme values as well as the mean of their extreme values.
2. ***plotFigs4Reports***
   1. *statdir* [string] = see seedSelection\_BATCH input
   2. *figdir* [string] = The absolute path to the directory where the figures will be saved.
   3. ***List of DLC families***
      1. *A list of various DLC families called by ‘###\_DLCs’ to aid in plotting*
      2. *For instance,* *POW\_DLCs = [16000000:30:16000180]; PAR\_DLCs = [61000000:30:61000180]*
   4. *DLC2plt* [string] = A string referring to the prefix (before ‘\_DLCs’) of a predefined DLC family to plot. For the previous example, this variable would be set to either ‘POW’ to plot the 1.6 family or ‘PAR’ to plot the 6.1 family. The seedSelection script must have been run on these DLC families with the same sensors.
   5. *vars2plot* [1 x nVars cell array of strings] = see seedSelection\_BATCH input: vars2seed
   6. *how2select* [1 x nVars cell array of strings] = see seedSelection\_BATCH input. (Must be same length as vars2seed)
   7. *itext* [logical] = If set to TRUE (1), the values of the statistics will be printed on directly on the plots. This feature is good for creating tables, but makes the figures UGLY!
3. ***getFatigue*** (blue input for mooring, red inputs for tower-base, purple for both)
   1. *DLCspreadsheet* [string] = The absolute path to the DLC spreadsheet used for fatigue runs.
   2. *sheetnum* [integer] = Sheet number in DLC spreadsheet containing fatigue runs (same one as used in RunOF\_BATCH).
   3. *fatdir* [string] = The absolute path to the directory where the run folders are stored. It should be equivalent to the RunFolder (defined relatively) in the DLC spreadsheet.
   4. *basename* [string] = A name used for the matfiles which are saved in the fatigue directory. The number of bins and other input parameters are appended to the string.
   5. *nSeeds* [integer] = Number of seeds per bin. Only used in legend of plots for Mooring Lines.
   6. *FatLifeYrs* [integer] = Number of years of lifespan of structure.
   7. *Heading* [integer] = Orientation of platform in degrees CCW from N. Only used for plotting of damage per bin for tower-base.
   8. *iPlot* [logical] = If set to TRUE, plots will be produced after the analysis occurs.
   9. *iPolarPlot* [logical] = If set to TRUE, polar plots will be produced after the analysis occurs, showing damage of each bin.
   10. *iVestas* [logical] =If set to TRUE, the basebend moment will be rotated 90 degrees to account for Vestas’ coordinates.
   11. *iRun* [logical] = If set to FALSE, the summary matfile of the fatigue results of a set of bins will be used for plotting purposes. If set to TRUE, fatigue analysis will be performed for each bin. Must be set to TRUE initially.
   12. *iRedoRain* [logical] = If set to TRUE, the rainflow counting analysis will be performed again and the fatigue results of each bin will be overwritten. If set to FALSE, the fatigue results of each bin will just be used if they exist.
   13. *fatType* [string] = Either ‘ML’ for mooring-line analysis or ‘TwrB’ for tower base analysis
   14. *standard* [string] = ‘ABS’ or ‘DNV’ or other acronym for a classification society. If ‘ABS,’ then only ORQ grade chain for estimation of the MBL is used.
   15. *tcutoff [float] = Ask Cyril about this low-pass filter.*
   16. *omega [float] = Ask Cyril about this low-pass filter.*
   17. *specLines [structure] = A structure containing various fields referring to special LineTypes in the mooring*
       1. *Types [**1 x nSpec cell array of strings] = Strings must exactly the match the LineType name in the OrcaFlex model.*
       2. *iMBLs* *[1 x nSpec float] = If set to 0, only special K and m specified will be used in the analysis. (The MBL will be calculated from the library). If set to 1, the MBL will be taken from the specified MBLs field (with no extra corrosion effects included). If set to 2, the MBL will be estimated from the library and the specified corrosion allowance will be used.*
       3. *MBLs [1 x nSpec float] = Specified values of lines to be used (if iMBL = 1).*
       4. *As [1 x nSpec float] = This parameter of the T-N curve is usually called K in the literature.*
       5. *Ms [**1 x nSpec float] = This parameter of the T-N curve is usually called m in the literature.*
       6. *CorrRs [1 x nSpec float] = Corrosion rate in mm/yr of steel chain. Only used if (iMBL = 2).*
   18. *m**[float] = exponent of the M-N curve.*
   19. *Mu**[float] = M0 of the M-N curve in kNm.*
   20. *N0**[float] = Number of total cycles in a given lifespan of structure.*
   21. *N0**[1 x nRot vector] = Angles of arbitrary +x axis to calculate bending moment, increasing CCW.*
   22. *Tfilt**[1 x nT vector] = Low and high end of filter.*
   23. *n\_order**[1 x nT vector] = Ask Bingbin about the order of a filter.*