**User manual: Sensitivity analysis workflow**

**1. Installation**

The module can be installed by downloading the directory **sensitivity\_analysis** from the windtrue git repository [<https://github.com/bsanderse/windtrue.git> ].

The **sensitivity\_analysis** directory contains the necessary routines for sensitivity analysis: uncertainty parameterization routines developed at CWI, Aero module routines [in the directory **AEROmodule**] and UQLab routines [in the directory **UQLab**].

As UQLab software package requires license, the files inside the directory **sensitivity\_analysis**/**UQLab/** shall be replaced with the one acquired by the user. User can get their license from UQLab website [<https://www.uqlab.com/> ].

Typically, the file downloaded from the UQLab website is named as UQLabCore\_Rel1.x.x.Zip. This file should be unzipped inside the directory **sensitivity\_analysis**/**UQLab/**. The license file (that user will obtain via their email from UQLab administrators) should be placed inside the directory **sensitivity\_analysis**/**UQLab/core.**  To activate the license and install UQLab package from Matlab command window, go to the directory **sensitivity\_analysis**/**UQLab/core** and run the command *uqlab\_install*.

**2. Running the Sensitivity Analysis for a wind turbine**

**2.1 Setting up the turbine data routine**

Turbine and site related data as well as the probability distributions for different random variables are defined in the routine **turbineData.m** that is stored in the directory **sensitivity\_analysis/cases/aero\_module/.** The variable names in **turbineData.m** are adopted from the **input.txt** file of the AEROmodule software.

The properties of the uncertain parameters are defined as a struct variable in Matlab which is similar to the random variable definition in the UQLab (e.g. Input.Marginals.XYZ). This routines already contains uncertainty definitions for most of the uncertain parameters, however, new parameters can be simply added as follows:

% ======= NewParameter =============

counter = counter + 1; % Should be increased by one before adding new parameter

Input.Marginals(counter).Name = 'NewParameter';

Input.Marginals(counter).Index = ''; % For indexing vectors. Set empty for scalars

Input.Marginals(counter).Type = 'Gaussian'; % Any distribution available in UQLab

Input.Marginals(counter).Parameters = [Mean, Std];

Input.Marginals(counter).Bounds = [LowerBound UpperBound];

% ===================================

For a vector parameter set, we have to define uncertainty for each element of the vector. User can follow a similar approach as used for the chord/twist/thickness curve defined already in this routine.

Although, we define the uncertain properties for all the parameters, we can selectively perform sensitivity analysis using a subset of parameters. The names of these selected parameters are specified in the variable uncertain\_params.

The routine **getParameterAeroModule.m** also located in the directory **sensitivity\_analysis/cases/aero\_module/** calls **turbineData.m** routine and converts the turbine/site data and uncertainty definitions in a format that is suitable for the UQLab routines. All the data in from **turbineData.m** routine is put into in the cell variable P{}. Further, if a new parameters is added to **turbineData.m** file then **getParameterAeroModule.m** routine should also be updated by appending the new parameter at the end of the cell variable P{}. Note that here the order in which the input parameters are assigned to P{} should not be changed. And new elements shall only be added in the end.

**2.2 Writing the sampled random parameters in the AEROmodule**

UQLab generates samples of random variables that should be written to AEROmodule input file **input.txt**. For this, we use **writeAeroModuleInput.m** routine located in the directory **/sensitivity\_analysis/cases/aero\_module/.** This routine modifies the **input.txt** file using the sampled random values of the parameters. If a new parameter is added this routine should be modified accordingly.

**2.3 Setting up the properties of Sensitivity analysis algorithm**

We specify algorithms used to compute the Sobol indices in **initialize.m** routine located in the directory **sensitivity\_analysis/cases/aero\_module/**.

There are four possible algorithms to compute Sobol indices: Monte Carlo (MC), Polynomial Chaos Expansion based on quadrature (PCE\_Quad), Polynomial Chaos Expansion based on Ordinary Least Squares and Polynomial Chaos Expansion based on Least Angle Regression (PCE\_LARS). We specify a list of options as follows:

methods = {'MC','PCE\_Quad','PCE\_OLS','PCE\_LARS'};

And for each methods, we further describe the number of samples or polynomial order as:

% Number of samples with MC

NsamplesMC = [10 100 1000]; % Run 3 different simulations with 10,100,1000 samples

MC\_repeat = 1; % To repeat the experiments a number of times

% For PCE-Quad, specify the polynomial degrees to be tested

DegreesQuad = 1:4; % this wil have polynomial upto degree 4

% For PCE-OLS, if not specified,number of samples from PCE-Quad is used

NsamplesOLS = [16 32 64]; % Run 3 different simulations with 16,32,64 samples

OLS\_repeat = 1; % To repeat the experiment a number of times

% For PCE-LARS, if not specified,number of samples from PCE-Quad is used

NsamplesLARS = [16 32 64]; % Run 3 different simulations with 16,32,64 samples

LARS\_repeat = 1; % To repeat the experiment a number of times

All four methods can be used to perform comparison and verify whether the three approaches result in similar sensitivities for the selected parameters. In practice, only one algorithm should be used, for e.g., when we want to use the PCE\_LARS algorithm, we use the following:

methods = {'PCE\_LARS'};

NsamplesLARS = [128];

LARS\_repeat = 1; % Without repetition

Other variables can be commented out.

**2.4 Running the sensitivity analysis**

The sensitivity analysis can be started by running **testSensitivity.m** routinelocated in the **sensitivity\_analysis**  directory. The output is a histogram plotting the values of sobol indices for each random parameter.