**User manual: Sensitivity analysis for ECNAero**

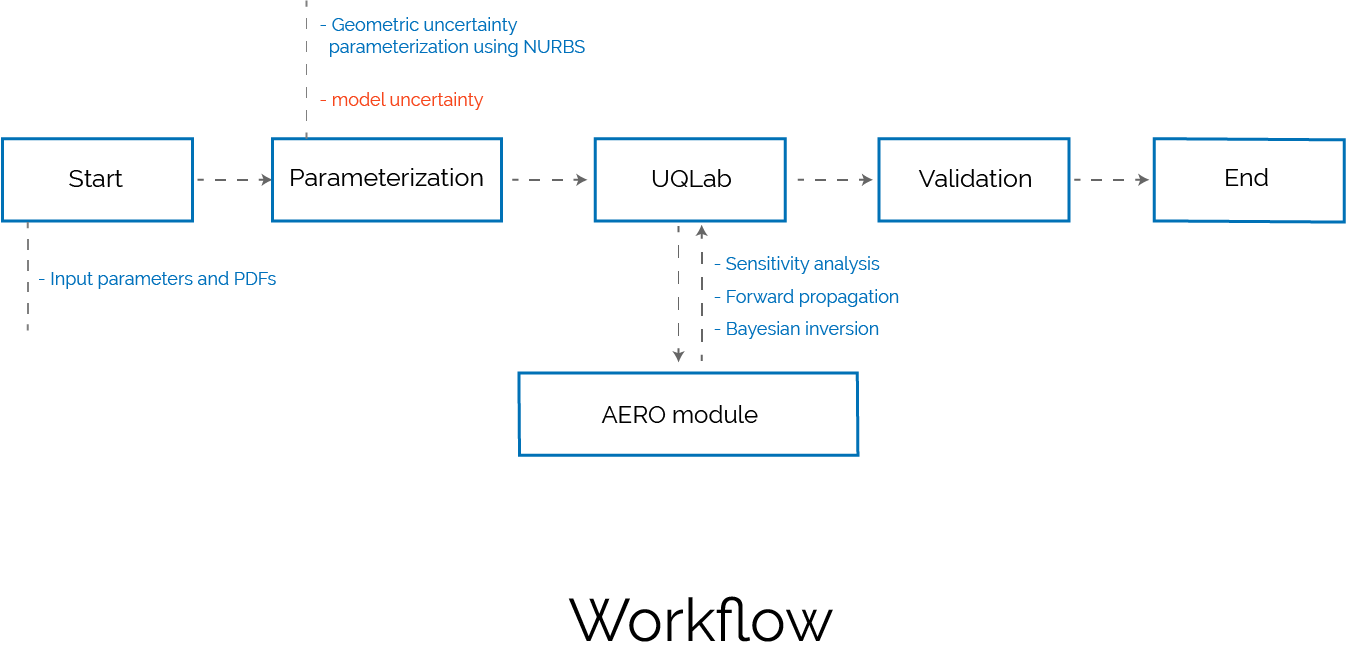
**0. Background/introduction**

In the WindTrue project, uncertainties in aeroelastic wind turbine models are studied. The first step in the project is to study which parameters are the most important given certain quantities of interest (e.g. power, forces, moments).

The sensitivity analysis as performed here is built on two main ingredients:

1. The computational model representing the wind turbine: **ECNAero**. ECNAero is a software code developed and maintained by ECN part of TNO.
2. The uncertainty quantification framework: **UQLab**. UQLab is a Matlab-based uncertainty quantification package developed and maintained at ETH Zürich and free to use for academic purposes.

The sensitivity analysis code presented here combines these two tools by using smart parameterizations (input processing) and wrappers to process the output of ECNAero.



**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**].

**System specifications**

Operating system: Windows 10

Matlab: R2019a

UQLab: 1.3.0

ECNAero: v252

The UQLab installation present in the UQLab directory can be directly used. However, it can happen that a newer UQLab version is available, which can be installed instead of the current available version on our Windtrue GitHub. A newer version can be downloaded from the UQLab website and is typically named as UQLabCore\_Rel1.x.x.Zip. This file should be unzipped inside the directory **sensitivity\_analysis**/**UQLab/**. The UQLab software requires a license, which can be obtained from the UQLab website [<https://www.uqlab.com/>]. The license file (that a user will obtain via email from the UQLab administrators) should be placed inside the directory **sensitivity\_analysis**/**UQLab/core**.To activate the license and install UQLab package from the Matlab command window, go to the directory **sensitivity\_analysis**/**UQLab/core** and run the command *uqlab\_install*.

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

**2.0 Overview**

The core of the sensitivity analysis is the file **testSensitivity.m**. Once this file is executed by the user, UQLab is started, several ECNAero evaluations are performed, and output is post-processed and visualized.

Several files can be changed in order to change the Sensitivity Analysis:

* **testSensitivity.m**: specify the case that will be executed, the type of Sobol analysis, and the postprocessing (see 2.4).
* **cases/aero\_module/initialize.m**: details of the methods used to perform the analysis, e.g. number of samples, polynomial degree, etc (see 2**.**3).
* **cases/aero\_module/turbineData.m**: specification of the uncertain input variables (see 2.1).

**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 UQLab (e.g. Input.Marginals.XYZ). This routine already contains definitions for most of the uncertain parameters. 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, as is done for example in the definition of uncertainty in the twist along a blade. A similar approach can be used as for the chord/twist/thickness curve defined already in this routine.

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

The routine **getParameterAeroModule.m** also located in the directory **sensitivity\_analysis/cases/aero\_module/** calls the **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 from the **turbineData.m** routine is put into in the cell variable P{}. Further, if a new parameter is added to the **turbineData.m** file then the **getParameterAeroModule.m** routine should also be updated by appending the new parameter at the end of the cell variable P{}. Note:

*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 the **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**

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

There are currently 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). For details, please see the UQLab manual.

We specify a list of the methods that we request UQLab to run for example as follows:

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

For each method, we further describe the number of samples or polynomial order (this depends on the method type) 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 will have polynomial up to 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 to verify whether the approaches result in similar sensitivities indices. In practice, often only a single algorithm is used (because of the computational expenses of running all algorithms can be high). For example, 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.

**2.5 Output description**

The histogram plots the values from variable AVG\_Sobol\_XYZ\_Total, for example, AVG\_Sobol\_MC\_Total, which stores the average value of total Sobol indices of each random variable, computed using MC method. The average is based on MC\_repeat number of experiment. Similarly, AVG\_Sobol\_OLS\_Total stores the average value of total Sobol indices computed using the ordinary least square method.