Multiobjective Optimisation and Integrated Design of Wind Turbine Blades Using WTBM-ANSYS for High Fidelity Structural Analysis

Alireza Maheri
School of Engineering, University of Aberdeen, Aberdeen, UK

Abstract:
Multiobjective optimisation and integrated aerodynamic-structural design of wind turbine blades are emerging approaches, both requiring significant number of high fidelity analyses. Designer-in-the-loop blade modelling and pre/ post-processing using specialised software is the bottleneck of high fidelity analysis and therefore a major obstacle in performing a robust optimisation, where thousands of high fidelity analyses are required to find the optimum solution. Removing this bottleneck is the driver for the development of WTBM, an automated wind turbine blade modeller. WTBM takes parameters defining the blade and its operating condition as inputs and generates pre-processor, solver and post-processor APDL files required by ANSYS for high fidelity analysis. The inputs can be generated automatically within an optimisation process, hence so can be the APDL files, allowing a fully automated optimisation in which any of the parameters which are required to define the size, topology, structure and material of a blade to be treated as a design variable. The solver parameters will be also updated automatically as necessary. The performance of WTBM-ANSYS in conducting hundreds of automated high fidelity analyses within an optimisation process is shown through multiobjective structural design and multiobjective integrated design case studies.

Keywords: WTBM; blade modelling; integrated design; multiobjective optimisation; ANSYS APDL; automated high fidelity analysis

1 Introduction

Wind turbine blades are traditionally designed in two sequential aerodynamic and structural design phases. There are a large number of published papers on blade optimisation at the aerodynamic design phase. In these works, while the focus of the research is on topology/shape optimisation of the blade, researchers have adopted different approaches in terms of the method of optimisation and the type of the blade. For example, recent publications [1-13] deal with conventional blades, papers [14 and 15] are about aerodynamic design optimisation of nonconventional blades, and the reported work [16] deals with the blades equipped with active flow controllers. In the structural design phase, the internal structure and the material of the blade are selected and designed. The focus at this stage is mainly to minimise the weight of the blades subject to constraints on maximum stress, stability and deformation [17-26].

As opposed to the traditional sequential design, the integrated design is an emerging approach, in which the aerodynamic and structural design phases are conducted simultaneously, aiming at the design of blades with overall optimal performance. Adopting an integrated design approach, the designer/optimisation algorithm can explore a broader design space towards finding superior solutions. The works reported in [27 and 28] are examples of recent advances in integrated design of blades. An integrated design approach may even lead to innovative solutions such as flatback aerofoils [29], which cannot be generated and found via sequential design optimisation. In an integrated design, the number of design candidate generation and evaluation grows exponentially with the size of design space. That is, an integrated design process needs a larger number of high fidelity analyses compared to sequential designs. Multiobjective optimisation is an essential part of integrated design, although some researchers have used multiobjective optimisation to conduct aerodynamic or structural design separately [30-33].

On one hand, conducting a large number of high fidelity analyses is unavoidable when we are looking for superior design solutions. On the other hand, the designer-in-the-loop modelling and pre/ post-processing using specialised software becomes the bottleneck of high fidelity analysis and therefore a major obstacle in performing optimisation, where hundreds and thousands of high fidelity analyses are required. Removing this bottleneck has been the driver for many research work including the one presented in this paper. Previous works have tackled the problem from different angles: lowering the level of fidelity [34, 35], the development of specific-purpose high fidelity FE solvers, such as Cp-Max [36] and other works reported in [37, 38], or the development of blade modellers which can produce input files for the general-purpose analysis tools such as ABAQUS and ANSYS [39, 40].

The closest tool to what is presented in this paper is NuMAD [40]. NuMAD (Numerical Manufacturing And Design) is an open-source software tool written in MATLAB which simplifies the process of creating a three-dimensional model of a wind turbine blade. The graphical, user-friendly tool manages all blade information including aerofoils, materials, and material placement. NuMAD uses the blade information to generate input files for other tools such as ANSYS. Many recent research on structural analysis of wind turbine blades use NuMAD [21, 41]. While NuMAD provides flexibility in blade modelling and helps saving time, it cannot deliver a fully automated modelling, pre-processing and post-processing. The presented software tool in this paper, WTBM (Wind Turbine Blade Modeller), besides making blade modelling an easy task, it also sets up solver and pre and post-processor parameters, and most importantly, it can be executed automatically within an optimisation process without the involvement of the designer.

The rest of the paper is organised as follows. Section 2 provides the reader with a big picture of WTBM in terms of its structure, function and capabilities. Section 3 elaborate on the blade definition protocols and associated attributes as used in WTBM. In Section 4, the theory behind the components of WTBM and the way they work are detailed through two illustrative examples. Three case studies in Section 5 show the capabilities of WTBM in practice, when used to conduct iterative high fidelity analysis and design optimisation.

2 ANSYS APDL and WTBM

ANSYS is amongst the very few powerful commercial packages with programming capabilities. APDL (ANSYS Parametric Design Language) allows parametric modelling as well as setting up pre-processor, solver and post-processor parameters. Using parametric FEA modelling for ANSYS has been used by a number of researchers for optimal design of wind turbine blades. For instance see a recent work reported in [20]. APDL becomes highly inefficient when utilised as a programming language for writing complex optimisation programmes. MATLAB, on the other hand, is a programming environment with incredible number of library functions and toolboxes, providing its users the flexibility and facilities that hardly any other programming languages can provide.

WTBM, developed in MATLAB, reads a number of input files and generates a number of APDL files. ANSYS then can be executed either by a user via ANSYS interface or in batch mode automatically via a third programme to read these APDL files and perform high fidelity analysis. WTAB can operate in three modes:

Mode 1: As a robust wind turbine blade modeller and pre-processor for ANSYS, it reads input files containing data on (i) blade geometry and material, (ii) aerodynamic loads and (iii) solver parameters and then calculates inertial forces and generates the APDL files required by pre-processor and solver. After generation of the APDL files, the user loads APDL files to ANSYS via the ANSYS graphical user interface (GUI) to solve the problem and then using ANSYS post-processor GUI analyses the results.

Mode 2: For running a complete high fidelity FEA to determine a number of parameters of particular interest (e.g. tip deflection, maximum stress, etc). In this case, in addition to pre-processor and solver commands, WTBM produces post-processor APDL commands. In this mode, a third programme execute WTBM to generate the APDL files and then calls ANSYS in batch mode (as opposed to interactive mode via its GUI) to read the APDL files and produce the output files. On generation of the output files, the programme extracts and displays parameters of particular interest from these files.

Mode 3: WTBM within an iterative process (e.g. heuristic/meta heuristic design and optimisation process). Similar to Mode 2, WTBM generates pre-processor, solver and post-processor APDL commands. The post-processor commands store the design candidate performance measures. A third programme, based on the flowchart of Figure 1, calls WTBM to produce APDL files, calls ANSYS in batch mode to read the APDL files and to produce output files containing control parameters and performance measures, evaluates the performance of the design candidate, generates a new design candidate and updates the input files automatically for the next run. This continues until the termination criteria are met.

The input files contain two sets of data. The data that one needs to model a blade and the data which are required for setting up FEA pre-processing, solver and post-processing. The input files, once generated, can be easily updated and replaced automatically, allowing a fully automated modelling and therefore conducting high fidelity analysis within an optimisation process without the burden of modelling the blade each time manually.

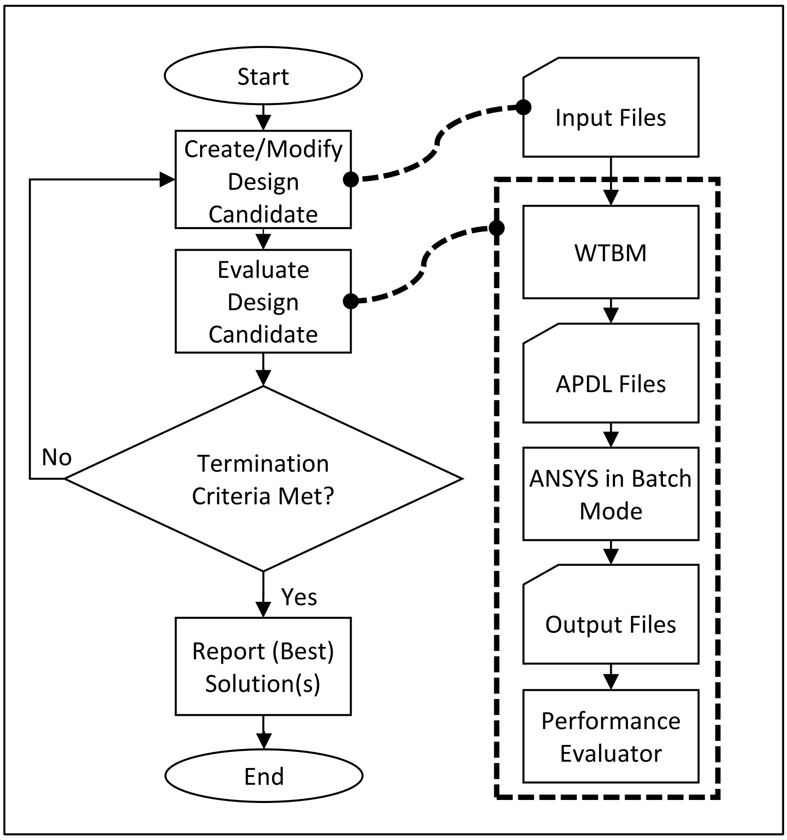

Start
Create/Modify Design Candidate
Evaluate Design Candidate
Termination Criteria Met?
No
Yes
Report (Best) Solution(s)
End
Input Files
WTBM
APDL Files
ANSYS in Batch Mode
Output Files
Performance Evaluator

Figure 1-A generic iterative process with high fidelity evaluation of design candidates using WTBM-ANSYS

WTBM is composed of two core modules, namely, Discretiser and APDL Writer (see blue boxes in Figure 2). The soft-coded module APDL Writer, simply generates text files containing the APDL commands required for the definition of the topology of the blade, assigning materials, assigning mesh size and element type to different sections, meshing and setting up solver (e.g. dynamic or static analysis, time step, etc.), applying nodal forces and boundary conditions and writing control parameters into an output file.

WTBM also needs two supporting modules, one for calculating inertial loads and one a wind turbine aerodynamic analyser for calculating the blade aerodynamic loads and wind turbine performance, in case of conducting an integrated design (yellow boxes in Figure 2). The blade aerodynamic loads depend on the blade topology as well as its operating condition (blade pitch angle, rotor speed, wind speed, azimuth angle, etc). By changing any parameter of these categories, the aerodynamic loads need to be re-calculated. That is, to be able to include an automated variation of these parameters, in addition to a Discretiser and an APDL Writer we also need a wind turbine aerodynamic analyser. This can be a CFD-based or a blade element momentum theory- (BEMT) based aerodynamic analyser. The advantage of the latter is in using aerodynamic coefficients of the blade aerofoils and therefore requiring significantly less computational power. If using standard aerofoils for the blade, the aerodynamic coefficients of the blade aerofoils will be available and therefore the aerodynamic analysis of the blade and wind turbine can be carried out using a robust BEMT-based analyser. In the current version of WTBM, the aerodynamic analysis of the blade and wind turbine is carried out by WTSim. The BEMT-based analyser WTSim is capable of simulating both constant and variable speed wind turbines with conventional and non-conventional blades (e.g. telescopic blades, blades equipped with microtabs and trailing edge flaps, swept back blades and adaptive blades) and it has a built-in simulator for the control systems [42, 43].

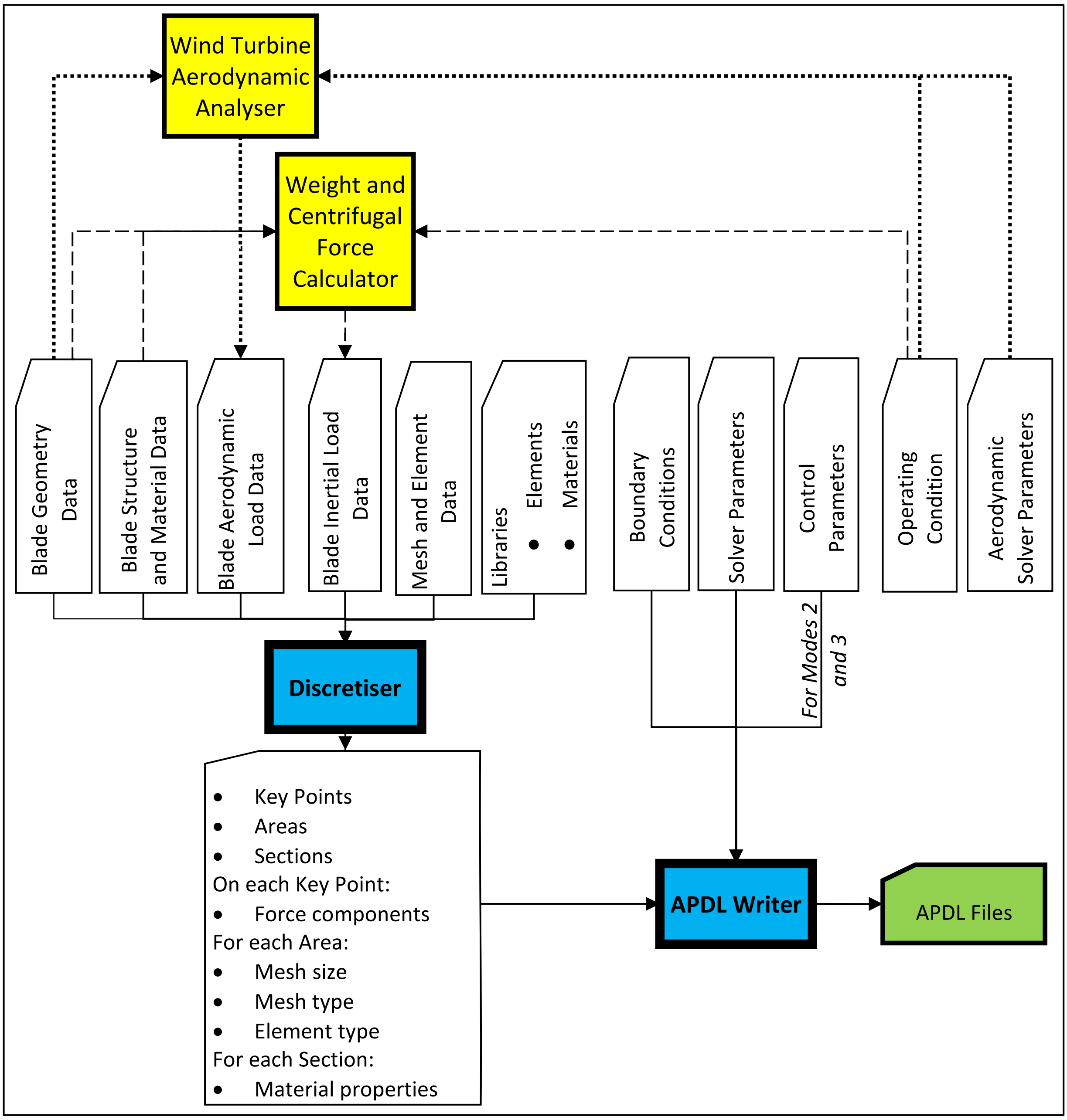


Figure 2- WTBM inputs, core and auxiliary modules and the data flow between modules

Similar to blade aerodynamic load, weight and centrifugal forces are also functions of operating conditions as well as the blade geometry and structural characteristics. That is, as shown in Figure 2, as a result of any changes in any of these parameters, these loads need to be recalculated. The input files shown in Figure 2 are explained gradually through Sections 3 and 4.

3 Blade definition

In designing a protocol for blade definition, the following has been taken into account.

- Blade definition protocol must be as universal as possible. That is, it should allow us to define both conventional and non-conventional blades, such as adaptive blades, swept-back blades, telescopic blades, morphing blades, etc.
- No limitation on the number of webs. A web can be located at any distance from the leading edge and anywhere along the span.
- No limitations on the type and the number of materials used.
- Easy for design optimisation formulation and manipulation. All parameters defining a blade could be treated as design variables and changed within an optimisation process, if required.
- Compatible with other aerodynamic analysis codes as much as possible. That is, input files of popular wind turbine aerodynamic analysers such as AeroDyn could be used as input files for WTBM with minimal changes.

A wind turbine blade is defined by two sets of parameters, namely, geometrical and structural parameters. These are explained separately in Sections 3.1 and 3.2.

3.1 Blade geometrical (topological/aerodynamic) parameters

These parameters define the topology of the blade and affect the aerodynamic performance of the blade. These parameters are normally defined or optimised within the aerodynamic design phase of blades. These parameters are rotor radius R, hub radius R_{hub}, and distributed parameters: chord length (c), pretwist (β_0), profile (identified by an index associated to a contour), aerofoil maximum thickness (t_{max}), the location of the twist axis x_t , and the origin of the aerofoil x-axis along the chord x_0 (see Figure 3). These distributed parameters are normally given as a function of span location r (measured from the centre of rotor) in the form of tabulated data.

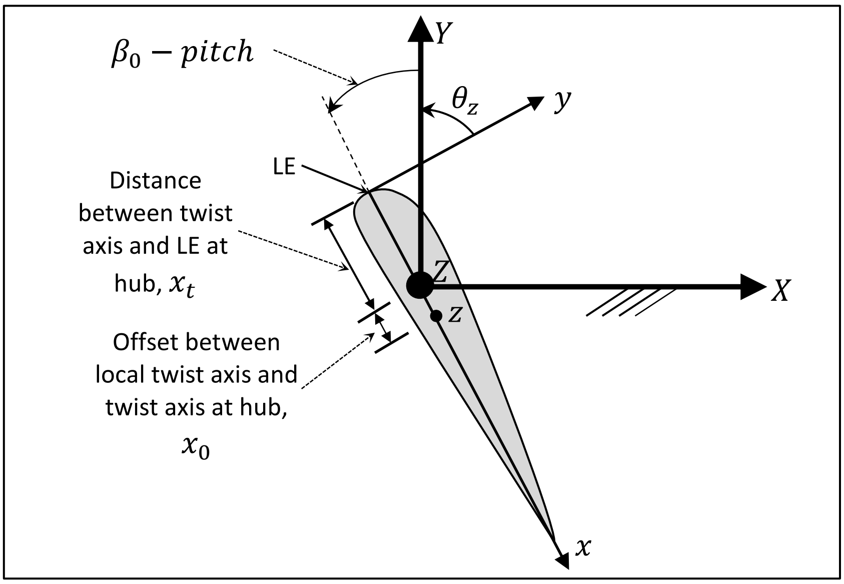

$\beta_0 - pitch$
Y
θ_z
y
LE
Distance between twist axis and LE at hub, x_t
Z
X
z
Offset between local twist axis and twist axis at hub, x_0
x

Figure 3- Aerofoil and global systems of coordinates

While the blade profile at different span locations $y(x)$ is defined using the aerofoil system of coordinates (x, y, z), the FEA is carried out in the global system of coordinates (X, Y, Z). Figure 3 shows these two systems of coordinates. Axis x is along the aerofoil chord line. Axes z and Z are measured from root to tip. Following the normal practice in the definition of angles in wind turbine blades, pretwist is measured positive to feather while pitch angle is measure positive to stall. Including the offset twist axis (x_0) in the definition of blade topology allows us to define and model some unconventional topologies (e.g. swept-back blades).

Transformation from the aerofoil system of coordinate to the global system of coordinates is given by Equation 1:

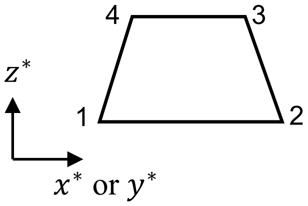
$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} cos\theta_z & sin\theta_z & 0 \\ -sin\theta_z & cos\theta_z & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x - x_t + x_0 \\ y \\ z \end{bmatrix} \tag{1}$$

3.2 Blade structural and material parameters

While topology of a wind turbine blade varies with span location only, the material properties vary both with span and chord locations. The internal structure can also take a variety of forms, depending on the number, location and extension of the webs. All these parameters need to be included when defining the structure of a blade. One way of defining the material and internal structure of a blade is by dividing a blade into a number of patches. Each patch can have a separate thickness and set of material properties. By doing this, one can update the location of a patch or its corresponding material properties without redefining the rest of the blade.

Each patch has two sets of attributes. The first set of attributes, as shown in Table 1, identify the location and material of the patch. The second set includes the analysis attributes and is explained in Section 3.3.

Table 1-Patch attributes-Blade definition

Attribute	Description
Location index	$I_{loc} \in \mathbb{N}; I_{loc} = 1$ for the upper surface, $I_{loc} = 2$ for the lower surface, $I_{loc} = 2 + i$ for the $i - th$ web (numbered from LE)
Coordinates in 2D Square systems of coordinates (see Section 3.2.1)	$[x_i^*, z_i^*]_{4\times2}$for patches on shell; $[x_i^*, y_i^*]_{4\times2}$ for patches on webs; $i \in \{1,2,3,4\}$ 
Layup index	$I_{layup} \in \mathbb{N}$ refers to the I_{layup}-th layup configuration in the layup pool

3.2.1 2D square systems of coordinates

Each patch is defined by a trapezoid. Patches are defined in two 2D square systems of coordinates. The (x^*, z^*) system of coordinates is used for defining those patches which form the upper and lower surfaces. The (y^*, z^*) system of coordinates is used for defining the patches which form the webs. The axes x^*, $y^*$and z^* are, respectively, normalised by the local chord, local aerofoil thickness and blade span, as given in Equations 2.a through 2.c.

$$x^* = \frac{x}{c(r)} \tag{2.a}$$

$$y^* = \frac{y - y_l(r,x)}{y_u(r,x) - y_l(r,x)} \tag{2.b}$$

$$z^* = \frac{z}{span} = \frac{r - R_{hub}}{R - R_{hub}}, \tag{2.c}$$

In these equations, at a radial location r, y_u and y_l are, respectively, the y-coordinates of the points on the upper and lower surfaces of the blade corresponding to the chord location x. Blade surfaces and webs are therefore defined as squares in this system, where $x^* = 0$ and $x^* = 1$ represent the leading and trailing edges; $y^* = 0$ and $y^* = 1$ represent the lower and upper surfaces of the blade; and $z^* = 0$ and $z^* = 1$ are the root and the tip of the blade. Using this system of coordinates makes it extremely easy for defining patches and assigning structural related design variables and manipulating them within an optimisation process.

3.2.2 Layup

Each layup index refers to a layup configuration of the form $\{mat_{ID}[\theta]_n\}$. Parameter n stands for the number of layers, mat_{ID} is the material index and θ is the fibre angle measured from the x^* direction. Material index identifies the material properties stored in the material library. Material properties are $E_{1/2/3}$, $G_{12/23/13}$, $\nu_{12/23/13}$, the layer thickness t_{layer} and the density. In the case of transversely isotropic composites, direction 1 is the fibre direction.

3.3 Solver associated parameters

In addition to its geometrical and material properties, a patch can also have some analysis attributes, such as the type of element and the mesh size (Table 2). For instance, different types of shell elements can be used for the thicker parts of the web cap compared to the rest of the shell, to model thin and thick-walled shell structures more accurately. As another example of requiring different solver attributes for different patches, one can refer to cases when we need a fine mesh for a particular part of the blade in order to capture higher gradients.

Table 2-Patch attributes-Solver parameters

Attribute	Description
Element index	$I_e \in \mathbb{N}$ refers to the I_e *-th* element in the element library (which is available in the version of the ANSYS being used)
Mesh key	Identifies the mesh type (APDL attribute)
Mesh size	Identifies the mesh size (APDL attribute)

The mesh size is initially set by the user. However, WTBM reassigns a mesh size, which is calculated based on the size and the shape of the patch if the initial mesh size leads to the generation of improper mesh with very high aspect ratios. In the case of an automated mesh refinement, the mesh size changes according to stress and/or deformation gradients to ensure the capture of sharp variations. Another case that mesh is updated automatically is within an optimisation process in which the patch size and/or material are treated as design variables. It is crucial to update the mesh size automatically to avoid producing ill elements.

3.4 Blade partitioning

Blade partitioning aims at producing key points which define the topology of the blade and the border of patches. WTBM, using curvature density [44], adds sections and adjusts them to patch boundaries to ensure smooth discretisation along both span and chord. If necessary, WTBM modifies the aerofoil contour to accommodate the shell thickness. This is particularly important for the trailing edge sections, as many aerofoil contours have zero thickness at the trailing edge.

Partitioning takes place in the 2D square systems of coordinates and then the generated key points are transformed to the global system of coordinates via an intermediate 3D normalised system of coordinates. The 3D intermediate system of coordinates, similar to the 2D square system of coordinates, is normalised by the local chord $c(r)$, and the blade span $R - R_{hub}$ for x^* and z^*.

However, y^* is normalised by the local chord instead of the local aerofoil thickness, as given by Equations 3.a through 3.b.

$$x^*_{3D} = x^* \tag{3.a}$$
$$y^*_{3D} = \frac{y^*(y_u - y_l) + y_l}{c} \tag{3.b}$$
$$z^*_{3D} = z^* \tag{3.c}$$

One can use Equations (3) and (2) respectively to transfer discretised key points from (x^*, y^*, z^*) to $(x^*_{3D}, y^*_{3D}, z^*_{3D})$ and then to (x, y, z) system of coordinates. Equation (1) then can be used to find the key points in the global system of coordinates (X, Y, Z).

3.5 Blade loads

As a result of the blade partitioning, the generated key points are located on a number of span locations z_i $(i = 1,2, \ldots, n_s)$ on parallel contours, normal to the blade axis as shown in Figure 4.

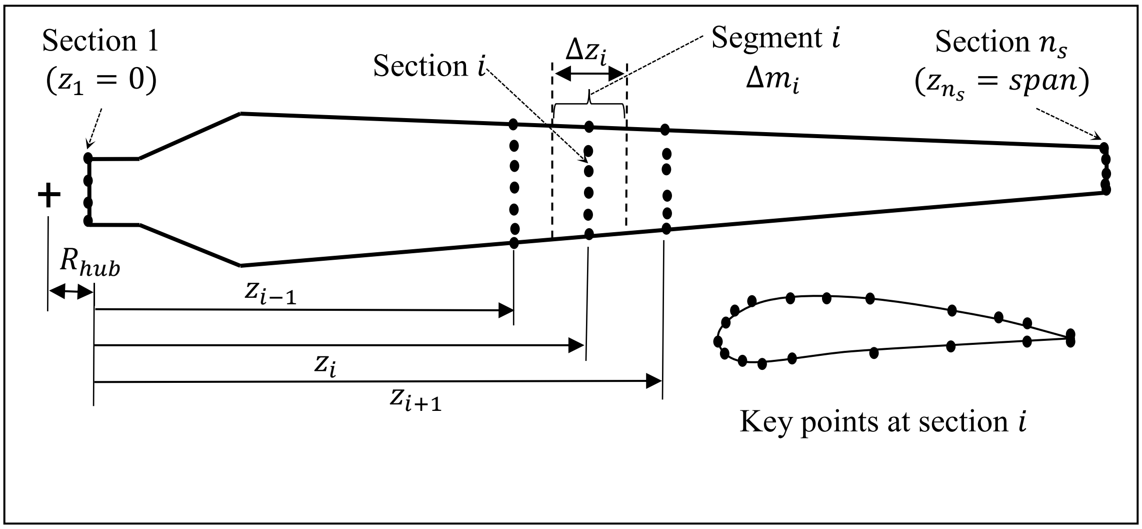


Figure 4-Blade partitioning and force discretisation on key points

The number of sections, n_s is calculated by

$$n_s = max\{n_{bound}, n_{s,min}\} \tag{4}$$

In Equation 4, n_{bound} is the total number of distinct boundaries along the span of the blade which are used to define the topology and structure of the blade. Parameter $n_{s,min}$ is set by the user allowing more control on the accuracy of the force discretisation process for cases in which the spanwise variation of the structure (e.g. shell thickness) and blade profile do not produce enough boundaries to capture the load variations accurately. In such cases, by setting $n_{s,min} > n_{bound}$, WTBM adds extra sections automatically where the sections are apart.

In order to apply the external forces on the key points, the blade is divided into n_s segments. Each segment contains one string of key points. The acting forces and moments at each span location z_i are calculated and distributed over the key points of that span location.

The aerodynamic forces lift L, drag D and pitching moment M_p at each span location z_i are given by Equation 5:

$$\begin{bmatrix} L \\ D \\ M_p \end{bmatrix}_i = \Delta z_i \begin{bmatrix} L'(z_i) \\ D'(z_i) \\ M_p'(z_i) \end{bmatrix} \tag{5}$$

in which L', D' and M_p' are the lift, drag and pitching moment per unit length of span (as calculated by WTSim), and

$$\Delta z_i = 0.5 \begin{cases} (z_{i+1} - z_i) & i = 1 \\ (z_{i+1} - z_{i-1}) & 1 < i < n_s \\ (z_i - z_{i-1}) & i = n_s \end{cases} \tag{6}$$

The pitching moment is, by convention, considered to be positive when it acts to pitch the aerofoil in the nose-up direction.

The centrifugal force, F_c, at span location z_i is given as follows:

$$[F_c]_i = \omega^2 \Delta m_i (R_{hub} + Z_i) cos\delta \tag{7}$$

where, ω is the rotor speed, Δm_i is the mass of the i-th segment, δ is the cone angle, and $(R_{hub} + Z_i)cos\delta$ is the rotation arm, the distance between the centre of the segment and the axis of rotation (see Figure 5). In the case of highly flexible blades with large deflections, the rotation arm needs to be corrected by taking into account the deflection of the blade. In such cases the calculation of the sectional centrifugal force must be conducted via an iterative process, in which in each iteration, the centrifugal force is updated by taking into account the offset produced by the deformation. Assuming negligible local deflection compared to the rotation arm for conventional blades is a reasonable assumption (as made in the current version of WTBM). The evaluation of the order of magnitude of the error produced by large deflections and possible inclusion of the iteration loop in WTBM needs further investigation.

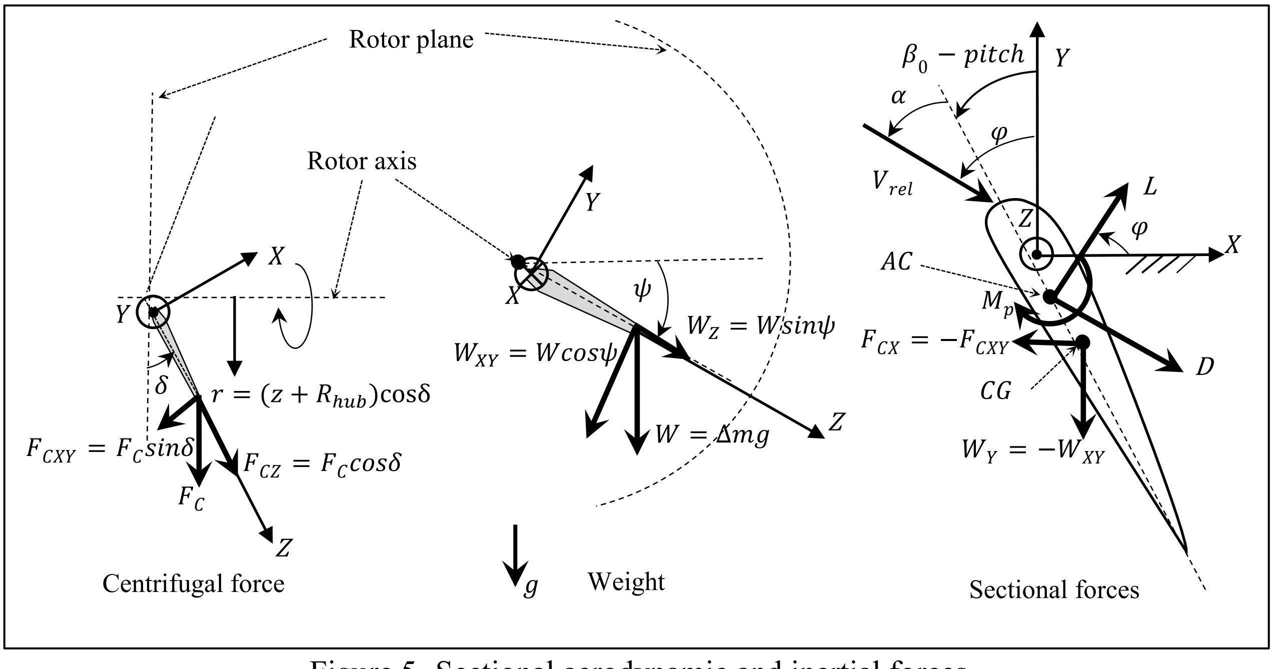


Figure 5- Sectional aerodynamic and inertial forces

The acting forces at each section can be transformed to the global system of coordinates using Equation 8.

$$\begin{bmatrix} F_X \\ F_Y \\ F_Z \\ M_Z \end{bmatrix}_i = \begin{bmatrix} cos\varphi & sin\varphi & 0 & -sin\delta & 0 \\ sin\varphi & -cos\varphi & -cos\psi & 0 & 0 \\ 0 & 0 & sin\psi & cos\delta & 0 \\ -Y_{AC}cos\varphi + X_{AC}sin\varphi & -Y_{AC}sin\varphi - X_{AC}cos\varphi & -X_{CG}cos\psi & Y_{CG}sin\delta & -1 \end{bmatrix}_i \begin{bmatrix} L \\ D \\ W \\ F_C \\ M_p \end{bmatrix}_i \tag{8}$$

where, as shown in Figure 5, φ and ψ are the inflow and azimuth angles respectively. In Figure 5, CG and AC stand for the centre of gravity and the aerodynamic centre respectively.

The global forces F_X, F_Y and F_Z at each section are then distributed over the key points uniformly:

$$\begin{bmatrix} F_X \\ F_Y \\ F_Z \end{bmatrix}_{KP,i} = \frac{1}{n_{KP,i}} \begin{bmatrix} F_X \\ F_Y \\ F_Z \end{bmatrix}_i \tag{9}$$

where, $[F]_{KP,i}$ stands for the forces on the key points at section i and $n_{KP,i}$ is the number of key points on that section. In order to ensure that the actual and the discretised force systems are equivalent, the effect of the moment produced by the discretised forces $F_{X,KP}$ and $F_{Y,KP}$ (ΔM_Z in Equation 10 below) must be taken into account when distributing the moment M_Z on the key points.

$$M_{Z,KP,i} = \frac{[M_Z - \Delta M_Z]_i}{n_{KP,i}} \tag{10}$$

where,

$$\Delta M_{Z,A,i} = \left[\sum_{j=1}^{n_{KP,i}} \left(-F_{X,KP,j} Y_{KP,j} + F_{Y,KP,j} X_{KP,j}\right)\right]_i. \tag{11}$$

4 Illustrative examples

In this section, the steps taking place in modelling of a blade are described. This includes blade topology definition input files, patch definition in 2D square systems of coordinates, blade partitioning and transformation to the global system of coordinates, and applying the forces on the key points.

4.1 Example 1-NREL 5MW wind turbine blade

4.1.1 Topology definition

Rotor radius $R = 63\ m$, a file containing the parameters of Table 1, together with 8 contour files (each identified by an associated index in Table 3) are all information needed for generating the topology of the outer shell of the NREL 5 MW blade. The first cell in the first column is the normalised hub radius $\frac{R_{hub}}{R}$. The data in the first six columns of this table has been adopted from reference [45]. These six columns are the set of data normally used to define traditional blades. The last column, as explained before, is used to define swept-back blades. Here, the last column $\frac{x_0}{c} \equiv 0$ stands for an unswept blade with straight axis. For a swept-back blade: $\frac{x_0}{c} = 0\ @\ r = R_{hub}$ and $\frac{x_0}{c} \neq 0\ @\ r \neq R_{hub}$.

Table 3-NREL 5MW blade topology parameters

	$\frac{r}{R}$ $(-)$	$\frac{c}{R}$ $(-)$	β_0 (°)	Aerofoil index	$\frac{t_{max}}{c}$ $(-)$	Twist axis, $\frac{x_t}{c}$ $(-)$ from LE	Offset between local twist axis and twist axis at hub, $\frac{x_0}{c}$ $(-)$
hub	0.024	0.056	13.3	Cylinder	1	0.25	0
	…	…	…	…	…	…	…
	0.408	0.066	8.5	5	0.3	0.25	0
	…	…	…	…	…	…	…
tip	1.000	0.023	0.0	8	0.18	0.25	0

4.1.2 Structural and material parameters

Material distribution varies both chordwise and spanwise as shown in Figures 6 and 7. There are different ways of representing the data which define the material distribution. In WTBM, the x^* location of webs (in percentile of chord from LE), patch distribution data, and the mechanical properties of the materials used in the blade are the information required to define the structure and material of the blade.

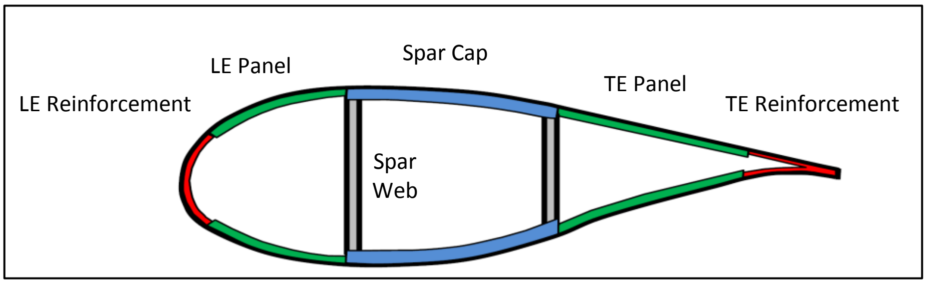


Figure 6-Typical chrodwise material distribution

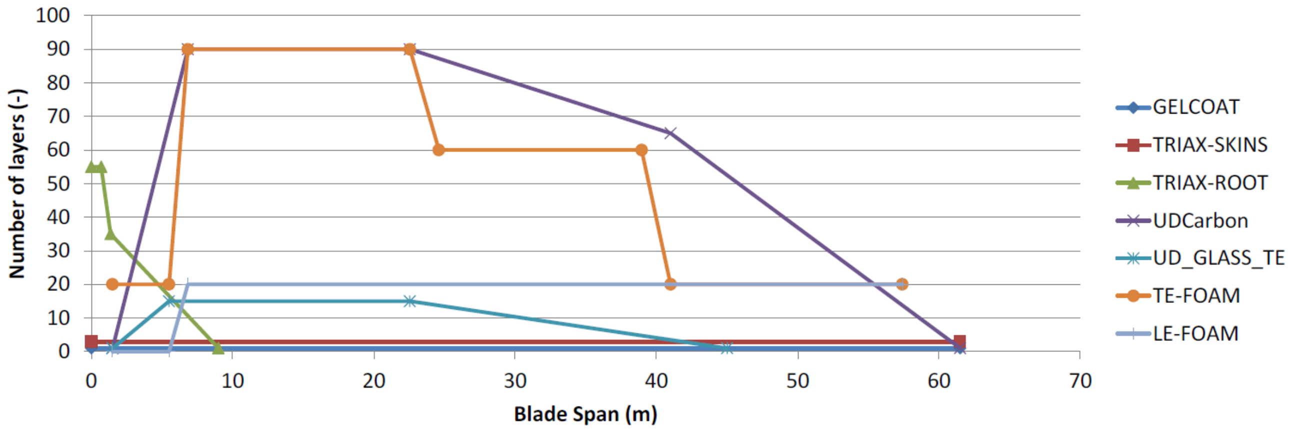


Figure 7- A typical spanwise material distribution [46]

In this example, the blade has two webs located at 25 and 55 percent of the chord from LE and the material is distributed in 5 segments along the chord direction and in 5 segments along the span direction to capture the chrodwise and spanwise variations of the material distribution (Figure 8). That is, the blade can be defined by 44 patches and 15 different layups. In Figure 8, the numbers in black are patch numbers and the numbers in red are the layup indexes. As it can be seen some patches have identical material properties (for example patches 4, 19, 20, 25, 40 and 41 are all made up of layup 4).

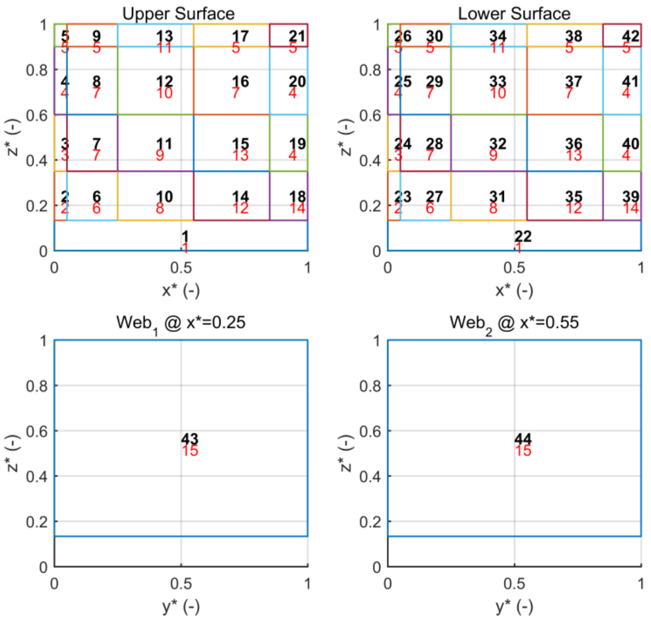


Figure 8-Patch distribution of the blade of Example 1

Tables 4 through 6, respectively, show the patch data, layups and material properties associated to each material index in the layups.

Table 4-Patch Attributes

Patch #	Location index	Patch coordinates In 2D square system of coordinates	Layup index	Element index*	Mesh key	Initial mesh size**
1	1	[0,0;1,0;1,0.13333;0,0.13333]	1	1	2	0.5
2	1	[0,0.13333;0.050,0.13333;0.050,0.35;0,0.35]	2	1	2	0.5
3	1	[0,0.35;0.050,0.35;0.050,0.60;0,0.60]	3	1	2	0.5
4	1	[0,0.60;0.050,0.60;0.050,0.90;0,0.90]	4	1	2	0.5
5	1	[0,0.90;0.050,0.90;0.050,1;0,1]	5	1	2	0.5
6	1	[0.050,0.13333;0.25,0.13333;0.25,0.35;0.050,0.35]	6	1	2	0.5
7	1	[0.050,0.35;0.25,0.35;0.25,0.60;0.050,0.60]	6	1	2	0.5
8	1	[0.050,0.60;0.25,0.60;0.25,0.90;0.050,0.90]	6	1	2	0.5
9	1	[0.050,0.90;0.25,0.90;0.25,1;0.050,1]	6	1	2	0.5
10	1	[0.25,0.13333;0.55,0.13333;0.55,0.35;0.25,0.35]	7	1	2	0.5
11	1	[0.25,0.35;0.55,0.35;0.55,0.60;0.25,0.60]	8	1	2	0.5
12	1	[0.25,0.60;0.55,0.60;0.55,0.90;0.25,0.90]	9	1	2	0.5
13	1	[0.25,0.90;0.55,0.90;0.55,1;0.25,1]	10	1	2	0.5
14	1	[0.55,0.13333;0.85,0.13333;0.85,0.35;0.55,0.35]	11	1	2	0.5
15	1	[0.55,0.35;0.85,0.35;0.85,0.60;0.55,0.60]	12	1	2	0.5
16	1	[0.55,0.60;0.85,0.60;0.85,0.90;0.55,0.90]	13	1	2	0.5
17	1	[0.55,0.90;0.85,0.90;0.85,1;0.55,1]	14	1	2	0.5
18	1	[0.85,0.13333;1,0.13333;1,0.35;0.85,0.35]	15	1	2	0.5
19	1	[0.85,0.35;1,0.35;1,0.60;0.85,0.60]	16	1	2	0.5
20	1	[0.85,0.60;1,0.60;1,0.90;0.85,0.90]	16	1	2	0.5
21	1	[0.85,0.90;1,0.90;1,1;0.85,1]	16	1	2	0.5
22	2	[0,0;1,0;1,0.13333;0,0.13333]	1	1	2	0.5
23	2	[0,0.13333;0.050,0.13333;0.050,0.35;0,0.35]	2	1	2	0.5
24	2	[0,0.35;0.050,0.35;0.050,0.60;0,0.60]	3	1	2	0.5
25	2	[0,0.60;0.050,0.60;0.050,0.90;0,0.90]	4	1	2	0.5

26	2	[0,0.90;0.050,0.90;0.050,1;0,1]	5	1	2	0.5
27	2	[0.050,0.13333;0.25,0.13333;0.25,0.35;0.050,0.35]	6	1	2	0.5
28	2	[0.050,0.35;0.25,0.35;0.25,0.60;0.050,0.60]	6	1	2	0.5
29	2	[0.050,0.60;0.25,0.60;0.25,0.90;0.050,0.90]	6	1	2	0.5
30	2	[0.050,0.90;0.25,0.90;0.25,1;0.050,1]	6	1	2	0.5
31	2	[0.25,0.13333;0.55,0.13333;0.55,0.35;0.25,0.35]	7	1	2	0.5
32	2	[0.25,0.35;0.55,0.35;0.55,0.60;0.25,0.60]	8	1	2	0.5
33	2	[0.25,0.60;0.55,0.60;0.55,0.90;0.25,0.90]	9	1	2	0.5
34	2	[0.25,0.90;0.55,0.90;0.55,1;0.25,1]	10	1	2	0.5
35	2	[0.55,0.13333;0.85,0.13333;0.85,0.35;0.55,0.35]	11	1	2	0.5
36	2	[0.55,0.35;0.85,0.35;0.85,0.60;0.55,0.60]	12	1	2	0.5
37	2	[0.55,0.60;0.85,0.60;0.85,0.90;0.55,0.90]	13	1	2	0.5
38	2	[0.55,0.90;0.85,0.90;0.85,1;0.55,1]	14	1	2	0.5
39	2	[0.85,0.13333;1,0.13333;1,0.35;0.85,0.35]	15	1	2	0.5
40	2	[0.85,0.35;1,0.35;1,0.60;0.85,0.60]	16	1	2	0.5
41	2	[0.85,0.60;1,0.60;1,0.90;0.85,0.90]	16	1	2	0.5
42	2	[0.85,0.90;1,0.90;1,1;0.85,1]	16	1	2	0.5
43	3	[0,0.13333;1,0.13333;1,1;0,1]	17	1	2	0.5
44	4	[0,0.13333;1,0.13333;1,1;0,1]	17	1	2	0.5

*With reference to the Element Library (in this example: Shell 181)
** Mesh size will be updated and resized based on the size and shape of each patch

Table 5-Layup Configurations

Layup index	Layup configuration, $\{mat_{ID}[\theta]_n\}$
1	{1[0]$_{1}$,3[±45]$_{25}$}
2	{1[0]$_{1}$,3[±45]$_{20}$,3[90]$_{15}$,2[0]$_{60}$,3[±45]$_{3}$}
3	{1[0]$_{1}$,3[±45]$_{3}$,3[90]$_{15}$,2[0]$_{40}$,3[±45]$_{3}$}
4	{1[0]$_{1}$,3[±45]$_{3}$,3[90]$_{15}$,2[0]$_{20}$,3[±45]$_{3}$}
5	{1[0]$_{1}$,3[±45]$_{6}$}
6	{1[0]$_{1}$,3[±45]$_{20}$, ,2[0]$_{20}$,3[±45]$_{3}$}
7	{1[0]$_{1}$,3[±45]$_{3}$, ,2[0]$_{20}$,3[±45]$_{3}$}
8	{1[0]$_{1}$,3[±45]$_{20}$,2[0]$_{20}$,4[90]$_{90}$,3[±45]$_{3}$}
9	{1[0]$_{1}$,3[±45]$_{3}$,2[0]$_{20}$,4[90]$_{60}$,3[±45]$_{3}$}
10	{1[0]$_{1}$,3[±45]$_{35}$,2[0]$_{20}$,4[90]$_{30}$,3[±45]$_{3}$}
11	{1[0]$_{1}$,3[±45]$_{3}$,2[0]$_{20}$,4[90]$_{5}$,3[±45]$_{3}$}
12	{1[0]$_{1}$,3[±45]$_{20}$, ,2[0]$_{90}$,3[±45]$_{3}$}
13	{1[0]$_{1}$,3[±45]$_{3}$,2[0]$_{60}$,3[±45]$_{3}$}
14	{1[0]$_{1}$,3[±45]$_{20}$,3[90]$_{15}$,2[0]$_{20}$, 3[±45]$_{3}$}
15	{3[±45]$_{3}$, 2[0]$_{50}$, 3[±45]$_{3}$}
16	{1[0]$_{1}$,3[±45]$_{3}$, 2[0]$_{60}$,3[±45]$_{3}$}
17	{4[±45]$_{4}$, 2[0]$_{40}$,4[±45]$_{3}$}

Table 6-Materials of Example 1

Mat. ID	Mat. Name	Thickness (mm)	Density (kg/m^3)	E_L (Gpa)	E_T (Gpa)	G_{LT} (Gpa)	ν_{LT} (-)
1	Gel Coat	0.05	1235	3.44	3.44	1.38	0.3
2	Foam	1	200	0.256	0.256	0.022	0.3
3	E-LT-5500-UD	0.47	1920	41.8	14	2.63	0.28
4	Carbon-UD	0.47	1220	114.5	8.39	5.99	0.27

4.1.3 Blade partitioning and transformation to global system of coordinates

After reading the data required for defining the blade topology, structural and material characteristics (or generating/modifying them within an iterative process), the module Discretiser carries out the blade partitioning and then through a number of steps transforms the key points, areas, sections and the loads to global system of coordinates. Figure 9 shows the partitioned blade in 2D square system of coordinates. Figure 10 shows the blade in 3D normalised and in global systems of coordinates.

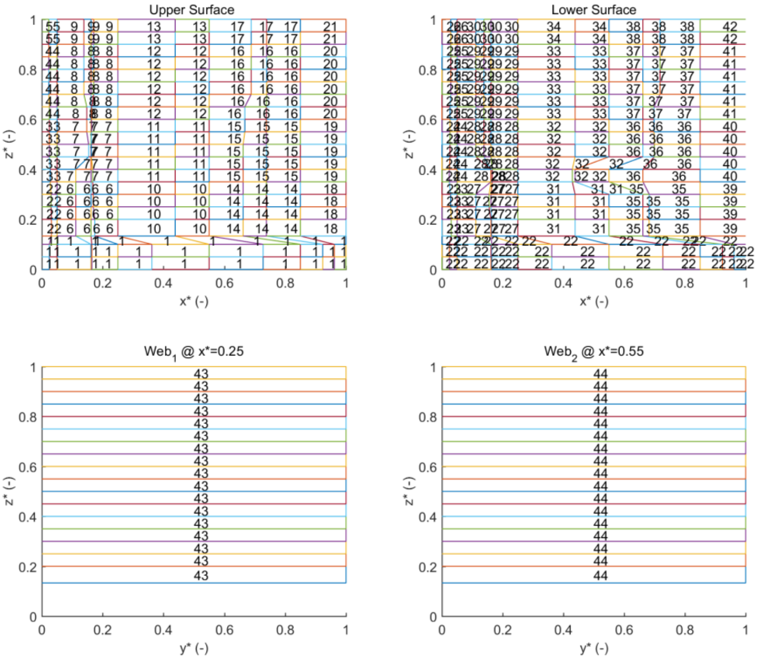


Figure 9- Blade of Example 1 partitioned in 2D square Plane systems of coordinates (x^*, y^*, z^*)

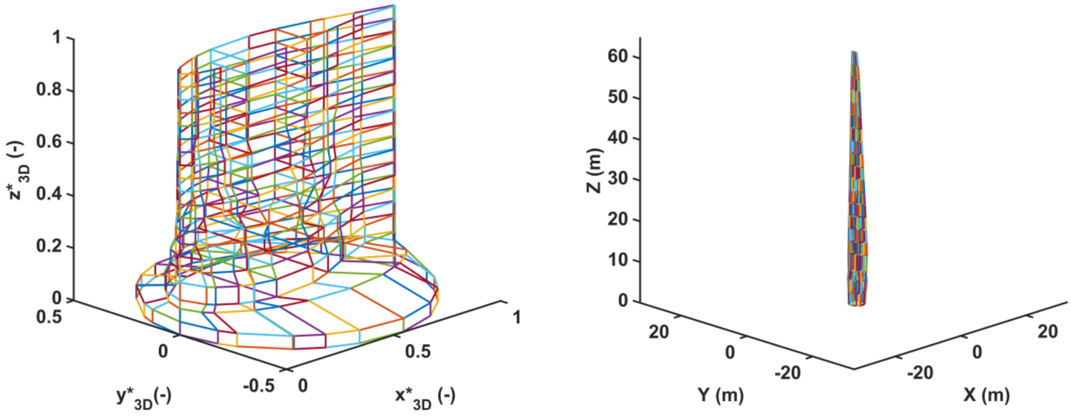


Figure 10- Blade of Example 1, from left to right, in the nondimensional system of coordinates $(x^*_{3D}, y^*_{3D}, z^*_{3D})$ and in the global system of coordinates (X, Y, Z)

4.1.4 Blade loading

The aerodynamic and inertial forces depend on the blade operating condition. Assuming that the modelled blade is operating at a wind speed of 12 m/s @ hub height of 80 m, a rotor speed of 12 rpm, pitch angle $= 3.8°$, zero yaw, and an azimuth angles of $\psi = -45°$, the global forces on the key points are calculated according to Equations 5 through 11. Figure 11 shows these forces on the key points. NREL 5MW has a cone angle of $\delta = 7°$.

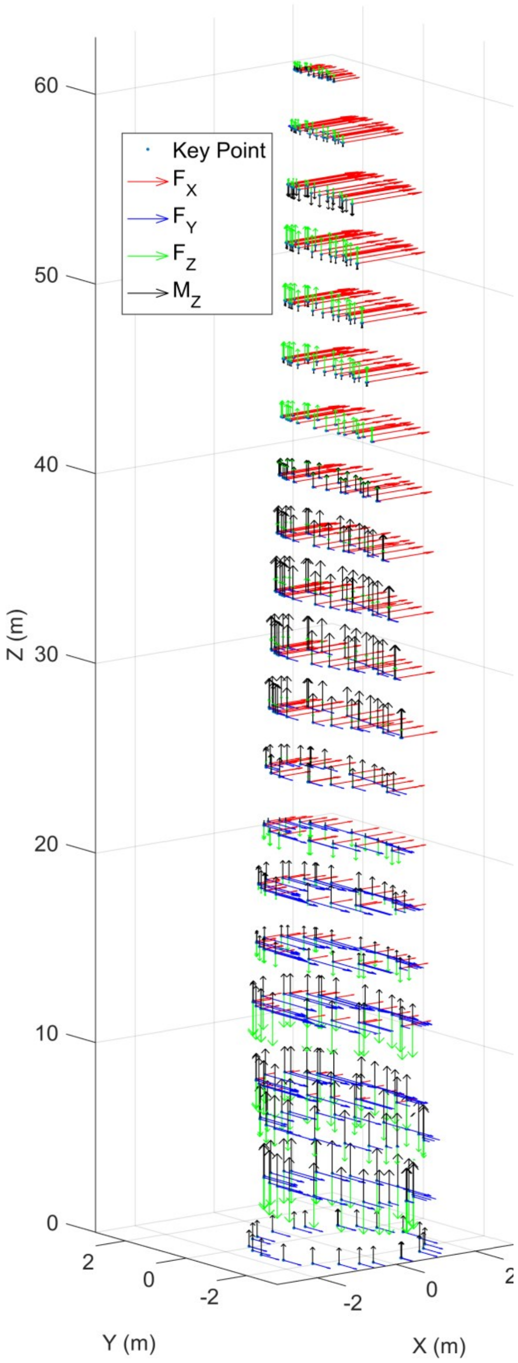
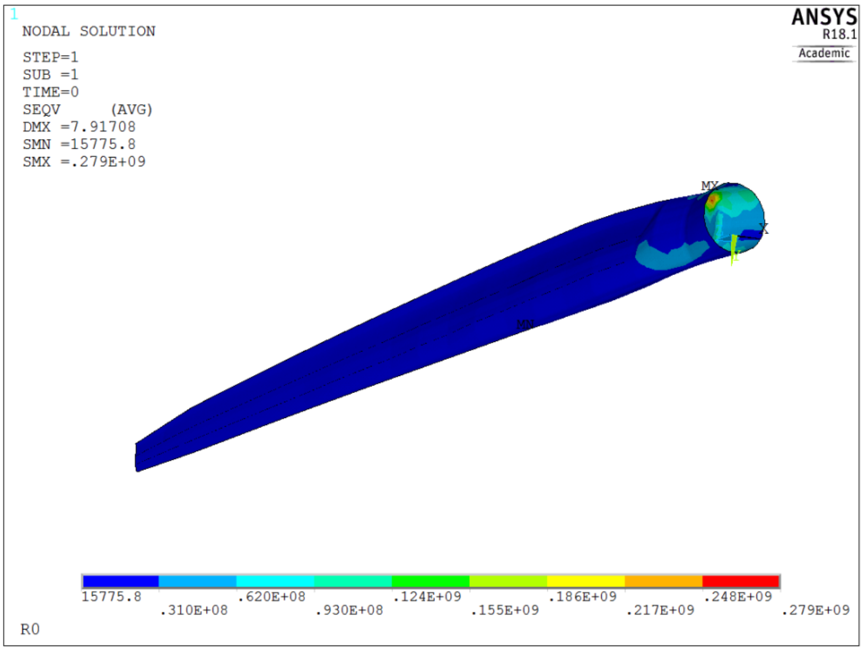


Figure 11-Acting forces on the key points of the blade of Example 1

4.1.5 Writing APDL files and running ANSYS

At this point all information for writing the APDL files are available and the soft-coded module APDL Writer generates the APDL files. See the appendix for partial APDL files written for the blade of Example 1. These APDL files are all required to run ANSYS, either via the ANSYS GUI manually (Mode 1 of operation in Section 2), or in a batch mode, where the ANSYS is called by a third programme (Mode 2). Figure 12 shows the result when the APDL files of the appendix is read via the ANSYS GUI. Figure 13 is generated by a third programme (a simple MATLAB code) which calls ANSYS in the batch mode and then, on the completion of the analysis by ANSYS, extracts and processes the information for a number of control points. In this figure, β and δ stand for the twist and deflection respectively; LE and TE donate the leading and trailing edges respectively; and the subscripts T, F (f) and E (e) stand for tip, flapwise and edgewise respectively.

The distortion of the shaft, the weakest patch, is evident by the calculated sectional twist $\beta = (\delta_{LE} - \delta_{TE})/c$.



```
1
NODAL SOLUTION

STEP=1
SUB =1
TIME=0
SEQV      (AVG)
DMX =7.91708
SMN =15775.8
SMX =.279E+09
```

Figure 12-Result of Example 1: Importing the APDL files via the ANSYS GUI manually

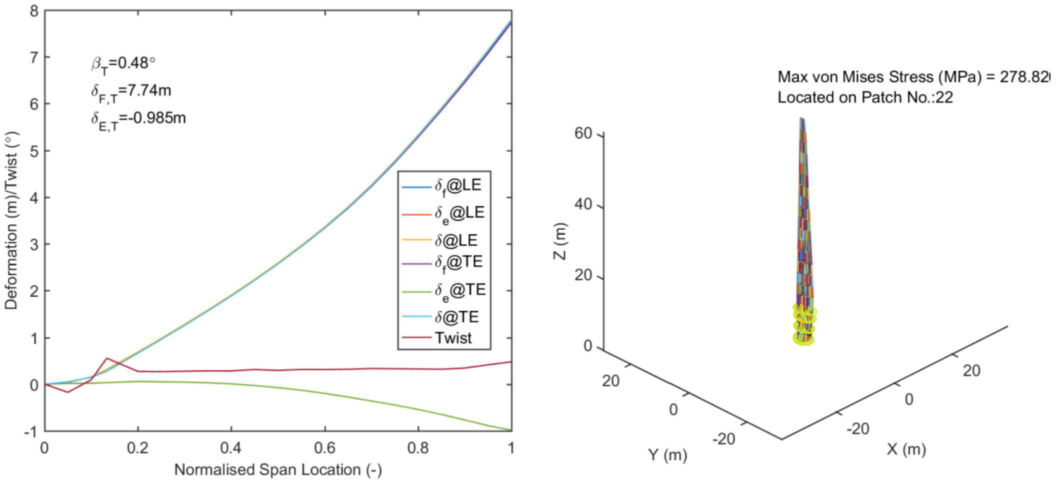


Figure 13- Results of Example 1 (deformation in the left and stress in the right): Calling ANSYS in batch mode by a third programme and extracting/processing the results for the control points

4.2 Example 2- Swept-back NREL 5MW blade

In this example, we are focused on showing how using 2D square system of coordinates is a simple way of defining blades with a wide range of topologies and structural configurations. Here, we model a swept-back NREL 5MW blade, with one web with variable layup configuration along the blade span located at 40% of chord. While the upper and lower surfaces should be covered completely by patches between hub and tip, this is not the case for webs. In this example, the web extends from 13% to 80% of span (see Figure 14). Also different from the blade of Example 1, here the cap has a constant width all through the span, which is represented by a number of trapezoids (patches 10-13 on the upper surface and 31-34 on the lower surface in Figure 14) in 2D square system of coordinates. Figure 15 shows the final blade model in global system of coordinates.

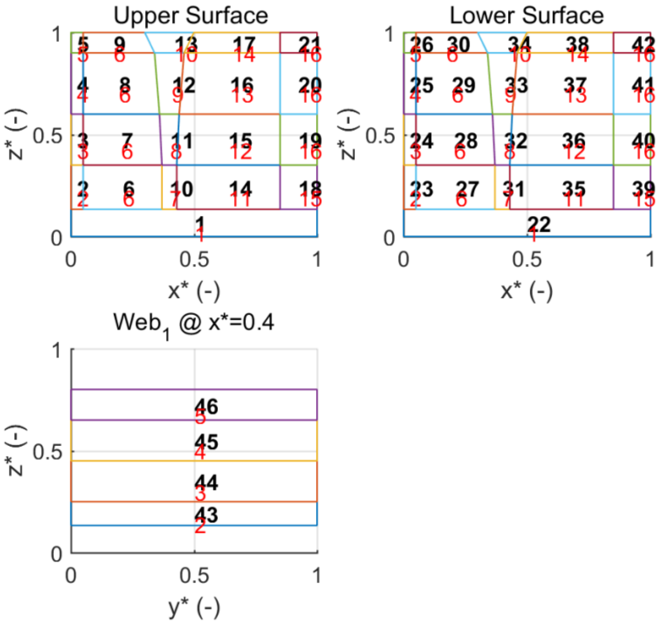


Figure 14-Patch distribution of the swept-back blade of Example 2 in 2D square systems of coordinates

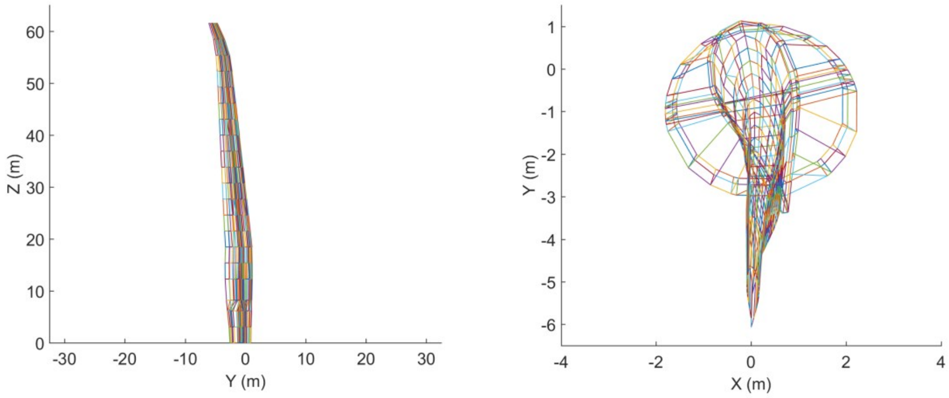


Figure 15-Swept-back blade of Example 2 in global system of coordinates

5 Case studies

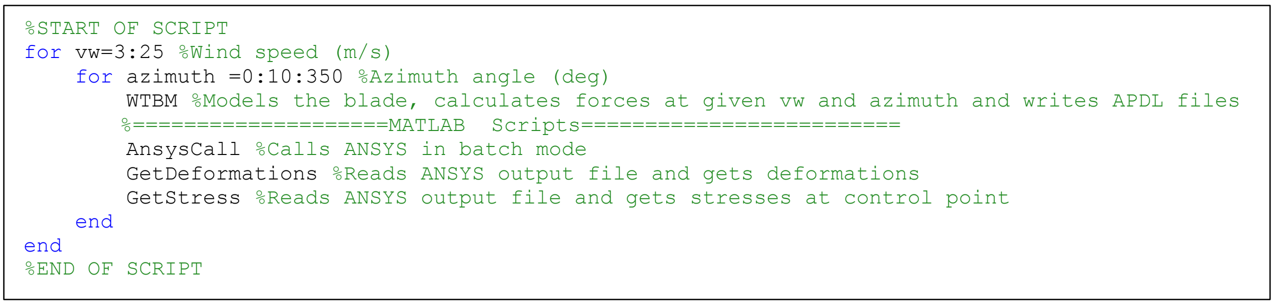
As mentioned before, the motivation behind the development of WTBM has been to make it possible to change any parameter of the four categories below automatically within an iterative process:

- wind turbine operating conditions
- blade aerodynamic parameters (blade topology)
- blade structural and material parameters
- solver parameters

This section presents three case studies. The first two case studies deal with iterative changes in wind turbine operating conditions and blade structural and material parameters respectively. Automatic change of the solver parameters is part of the second case study, in which the mesh size is updated automatically as the size of a number of patches changes. The third case study contains parameters from all four categories above in the form of a simple integrated design case. It should be noted that these case studies have been intentionally designed to be simple to avoid unnecessary and irrelevant details. Moving from these simple case studies to complete and complex integrated design cases is just a matter of choice of the number of design variables and the optimisation method.

5.1 Case Study 1- High fidelity structural analysis at various operating condition

For the wind turbine of Example 1 (Figure 8 and Tables 4 to 6), we are looking for the maximum stress in the blade and the blade tip deflection at different azimuth angles and different wind speeds calculated accurately using shell FE. In practice, these values are essential in design of blades installed on wind turbines using individual pitch control systems. High fidelity analysis is conducted by ANSYS within a loop as shown in the MATLAB script of Figure 16. In this script, AnsysCall is a one-line MATLAB script which calls ANSYS in batch mode. GetDeformations and GetStress are simple MATLAB scripts which read ANSYS output data files saved on the disk and extract maximum deformations and stresses as well as deformations and stresses at the control points specified by the user.



```
%START OF SCRIPT
for vw=3:25 %Wind speed (m/s)
    for azimuth =0:10:350 %Azimuth angle (deg)
        WTBM %Models the blade, calculates forces at given vw and azimuth and writes APDL files
       %===================MATLAB  Scripts========================
        AnsysCall %Calls ANSYS in batch mode
        GetDeformations %Reads ANSYS output file and gets deformations
        GetStress %Reads ANSYS output file and gets stresses at control point
    end
end
%END OF SCRIPT
```

Figure 16- MATLAB script calling ANSYS in nested loops over wind speed and azimuth angle

Here, the blade geometry and material/structural characteristics are fixed. However, as the wind speed changes and as the blade rotates, the aerodynamic and inertial forces change. The loads are all calculated within WTBM (by calling WTSim) and new APDL files are written in each iteration.

With a grid size of 1 m/s in wind speed and 10 degrees in the azimuth angle, the nested loops of Figure 16 conduct 828 high fidelity analyses. The rotor speed and pitch angle at each wind speed are given according to the control law of [45]. Results are shown in Figure 17. The azimuth angle is measured from 3 o'clock horizontal clockwise.

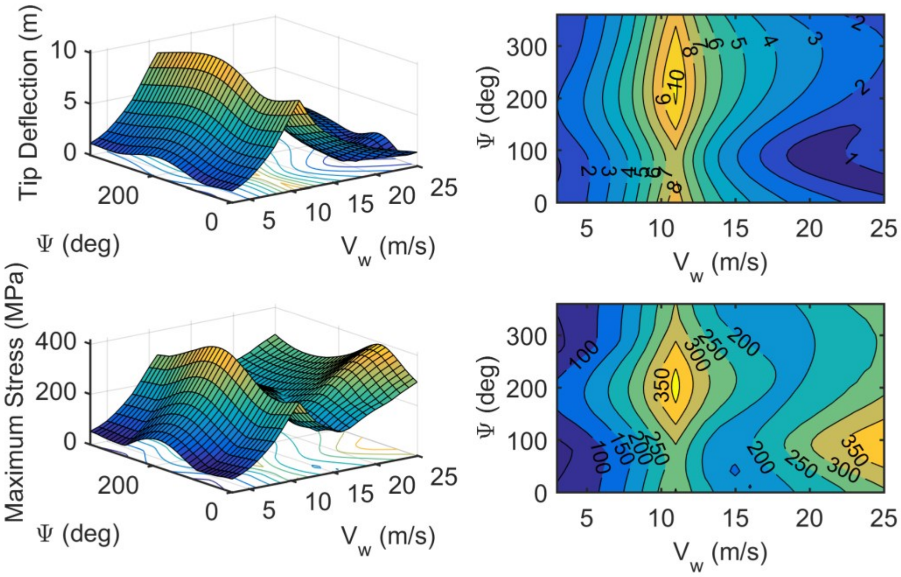


Figure 17- Blade tip deflection and maximum von Mises stress in the blade at different azimuth angle and wind speeds

5.2 Case Study 2- Multiobjective optimal size and location of spar cap in adaptive blades

In this case study, WTBM is used within an optimisation process to find the best configuration for an adaptive blade. Adaptive blades are aeroelastically tailored to respond to changes in operating conditions. This response can have favourable effect towards reducing the aerodynamic loads on the blade or increasing the rotor energy capture capability [47]. Aeroelastic tailoring can be achieved by using layers of unbalanced composites. When unbalanced layup configurations are used, bend-twist and/or stretch-twist elastic couplings are introduced into the structure of the blade. Figure 18 shows one of the many different configurations that can be used for making adaptive blades.

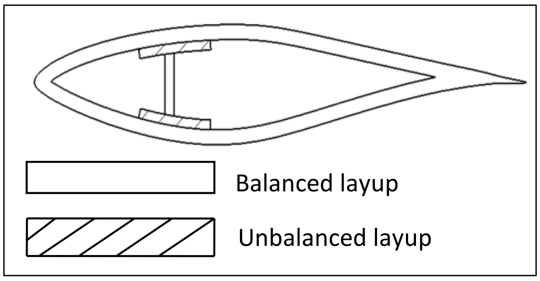


Figure 18- Unbalanced spar cap layers producing bend-twist elastic coupling

In a bend-twist coupled adaptive blade, the blade twists when it is subjected to a bending moment (e.g. produced by the aerodynamic forces). The slightest twist of the blade changes its aerodynamic performance significantly. The aerodynamic performance analysis of these blades cannot be carried out without having sufficient knowledge of the material prosperities and structural configuration and predicting the torsional deformation of the blade [48]. This makes the design of adaptive blades different from the design of conventional blades. Here, we adopt a decoupled design method developed for design of adaptive blades [49, 50]. In a decoupled design approach, the induced twist at the tip of the blade at a reference operating condition, $\beta_{T,ref}$, is treated as an aerodynamic design parameter. The optimum tip induced twist, $\beta_{T,ref,opt}$, is then obtained at the aerodynamic design stage. At the structural design phase the designer needs to ensure the satisfaction of the constraint $\beta_{T,ref} = \beta_{T,ref,opt}$ (see [49] for more details).

Assuming that conducting the first part of a decoupled design for an adaptive version of NREL 5MW blade has led to the optimum value of tip induced twist: $\beta_{T,ref,opt} = 2°$ @ $V_{ref} = 10\ m/s$. Now at the structural design phase we need to find the structural/material characteristics such that the right amount of bend-twist elastic coupling, leading to a planned tip induced twist ($\beta_{T,ref} = 2°$), is achieved.

For simplicity, without loss of generality, we assume that all parameters defining the topology and the structure of the blade are fixed except the location of the web x_{web}, the width of the spar cap Δ_{cap} (patches 10-13 and 31-34 in Figure 19), and the number of unbalanced layers along the blade span: $n_1$to n_4 in layups 7-10, 18-21, associated to these patches (see Table 7).

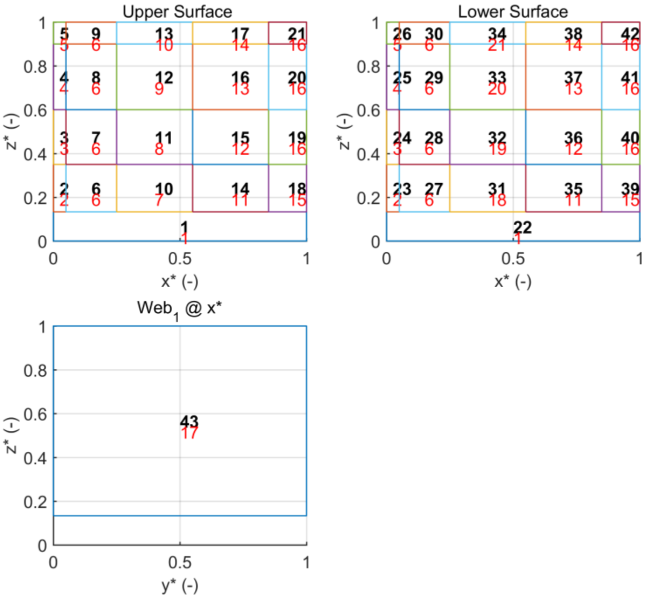


Figure 19-Patch distribution of the adaptive blade of Case Study 2 (web location is a design variable)

Table 7-Layup configurations including design variables $n_1$to n_4

Layup index	Layup configuration, $\{mat_{ID}[\theta]_n\}$
1	$\{1[0]_1,3[\pm45]_{50}\}$
2	$\{1[0]_1,3[\pm45]_4,3[90]_{20}\}$
3	$\{1[0]_1,3[\pm45]_4,3[90]_{16}\}$
4	$\{1[0]_1,3[\pm45]_4,3[90]_8\}$
5	$\{1[0]_1,3[\pm45]_4\}$
6	$\{1[0]_1,3[\ \pm45]_4,2[0]_{20},3[\pm45]_4\}$
7	$\{1[0]_1,3[\pm45]_4,4[70]_{n1},4[90]_{90-n1},3[\pm45]_4\}$
8	$\{1[0]_1,3[\pm45]_4,4[70]_{n2},4[90]_{60-n2},3[\pm45]_4\}$
9	$\{1[0]_1,3[\pm45]_4,4[70]_{n3},4[90]_{30-n3},3[\pm45]_4\}$
10	$\{1[0]_1,3[\pm45]_4,4[70]_{n4},4[90]_{10-n4},3[\pm45]_4\}$
11	$\{1[0]_1,3[\pm45]_4,2[0]_{80},3[\pm45]_4\}$
12	$\{1[0]_1,3[\pm45]_4,2[0]_{60},3[\pm45]_4\}$
13	$\{1[0]_1,3[\pm45]_4,2[0]_{40},3[\pm45]_4\}$
14	$\{1[0]_1,3[\pm45]_4,2[0]_{20},3[\pm45]_4\}$
15	$\{1[0]_1,3[\pm45]_4,3[90]_{20}\}$
16	$\{1[0]_1,3[\pm45]_4,3[90]_{10}\}$
17	$\{3[\pm45]_4,2[0]_{20},3[\pm45]_4\}$
18	$\{1[0]_1,3[\pm45]_4,4[-70]_{n1},4[90]_{90-n1},3[\pm45]_4\}$
19	$\{1[0]_1,3[\pm45]_4,4[-70]_{n2},4[90]_{60-n2},3[\pm45]_4\}$
20	$\{1[0]_1,3[\pm45]_4,4[-70]_{n3},4[90]_{30-n3},3[\pm45]_4\}$
21	$\{1[0]_1,3[\pm45]_4,4[-70]_{n4},4[90]_{10-n4},3[\pm45]_4\}$

Here, the fixed parameters are those which define the topology of the blade as well as the total number of patches, the location of all patches with balanced layups, the layup configuration of all balanced patches, and the stacking sequence and the total number of unidirectional layers in the unbalanced patches (10-13, 31-34). Size of the unbalanced patches is to be determined through optimisation. Fibre angle for unbalanced layers is taken as $\theta = 70°$ measured from chord direction ($\theta = 20°$ measured along the blade axis). The number of the unidirectional layers at $\theta = 70°$ in each unbalanced patch are also design variables to be determined as part of the optimisation process. The number of webs is fixed (one), but the location of the web is a design variable.

In the form of a standard optimisation problem, we are looking at minimising the mass of the blade m_{blade}, the maximum von Mises stress σ_{max}, and the blade tip deflection δ_T, subject to an equality constraints applied on the induced twist at the tip of the blade:

$$min\, Y = f(x) \tag{12.a}$$

s.t.

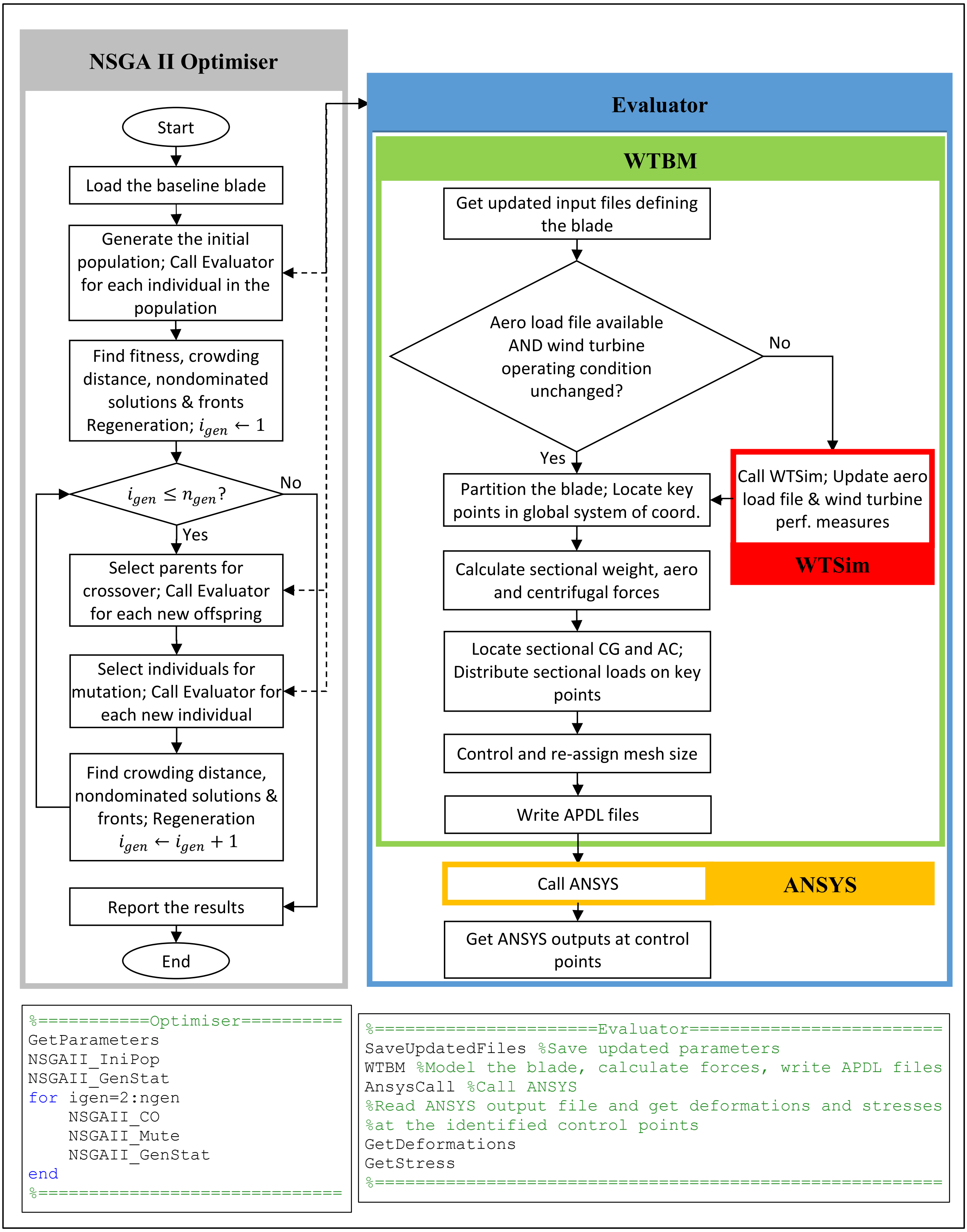
$$\beta_{T,ref} = 2° \text{ @ } V_{ref} = 10\, m/s \tag{12.b}$$

where, $X = \{x_{web}, \Delta_{cap}, n_1, n_2, n_3, n_4\}$ is the vector of design variables and $Y = \{m_{blade}, \sigma_{max}, \delta_T\}$ is the vector of objectives. Taking into account the physical meaning of $\beta_{T,ref}$, the mathematical equality constraint applied on $\beta_{T,ref}$ is replaced with the physical box constraint below:

$$\beta_{T,ref} = 2 \pm 0.05° \text{ @ } V_{ref} = 10\, m/s \tag{12.c}$$

Different methods can be used to solve the multi-objective optimisation above. Since in this study the emphasis is on high fidelity evaluation within an optimisation process rather than the optimisation method itself, only a brief explanation on the optimisation process is given. Figure 20 shows the flowchart of the main programme Optimiser and the Evaluator as well as the main MATLAB scripts. Optimiser conducts the multiobjective optimisation using the method of NSGA-II. Evaluator is called every time that a new design candidate is made within the optimisation process. At the beginning of the optimisation process, the MATLAB script GetParameters loads all parameters which are required for the blade modelling and remain unchanged through the optimisation process. NSGAII_IniPop generates initial population. NSGAII_GenStat calculates crowding distance, finds nondominated solutions and perform other required statistical analysis in the population. NSGAII_CO and NSGAII_Mute perform crossover and mutation respectively. Files NSGAII_IniPop, NSGAII_CO and NSGAII_Mute generate new solutions. Once a new solution is generated (e.g. by changing the location of the web, layups and width of unbalanced patches, etc), it is sent for evaluation by calling the script Evaluator. Evaluator executes WTBM, ANSYS and other scripts such as GetDeformation and GetStress.

Here in the NSGA-II algorithm the population size, probability of crossover and mutation, and the total number of generations (n_{gen}) are set as 40, 0.3, 0.1 and 50 respectively. In this algorithm, each crossover operation produces two new offspring and each mutation operator makes one. Therefore this algorithm requires a total number of 1440 high fidelity analyses for this case study.


NSGA II Optimiser
Start
Load the baseline blade
Generate the initial population; Call Evaluator for each individual in the population
Find fitness, crowding distance, nondominated solutions & fronts Regeneration; $i_{gen} \leftarrow 1$
$i_{gen} \leq n_{gen}$?
No
Yes
Select parents for crossover; Call Evaluator for each new offspring
Select individuals for mutation; Call Evaluator for each new individual
Find crowding distance, nondominated solutions & fronts; Regeneration $i_{gen} \leftarrow i_{gen} + 1$
Report the results
End
Evaluator
WTBM
Get updated input files defining the blade
Aero load file available AND wind turbine operating condition unchanged?
No
Yes
Call WTSim; Update aero load file & wind turbine perf. measures
WTSim
Partition the blade; Locate key points in global system of coord.
Calculate sectional weight, aero and centrifugal forces
Locate sectional CG and AC; Distribute sectional loads on key points
Control and re-assign mesh size
Write APDL files
Call ANSYS
ANSYS
Get ANSYS outputs at control points

```
%===========Optimiser==========
GetParameters
NSGAII_IniPop
NSGAII_GenStat
for igen=2:ngen
    NSGAII_CO
    NSGAII_Mute
    NSGAII_GenStat
end
%==============================
```

```
%=====================Evaluator=======================
SaveUpdatedFiles %Save updated parameters
WTBM %Model the blade, calculate forces, write APDL files
AnsysCall %Call ANSYS
%Read ANSYS output file and get deformations and stresses
%at the identified control points
GetDeformations
GetStress
%====================================================
```

Figure 20-Flowchart and the main MATLAB scripts of the main programme Optimiser and the Evaluator of Case Study 2

Figure 21 shows the 18 nondimaniated solutions obtained as a result of this multiobjective optimisation. All these solutions satisfy the constraint of Equation 12.c. Any of these solutions can be selected as the optimum solution by the designer through a trade-off process. For a better visualisation of the solutions against each pair of objectives, two dimensional Pareto fronts are also shown in this figure (top right and bottom figures). The design variables and design qualities associated to these solutions are shown in Table 8. In this table, the three solutions identified with '*' are the extreme Pareto solutions, possessing the best performance in terms of the three objectives. Solution #1 is the lightest, Solution #16 has the lowest maximum stress and Solution # 18 has the lowest deflection.

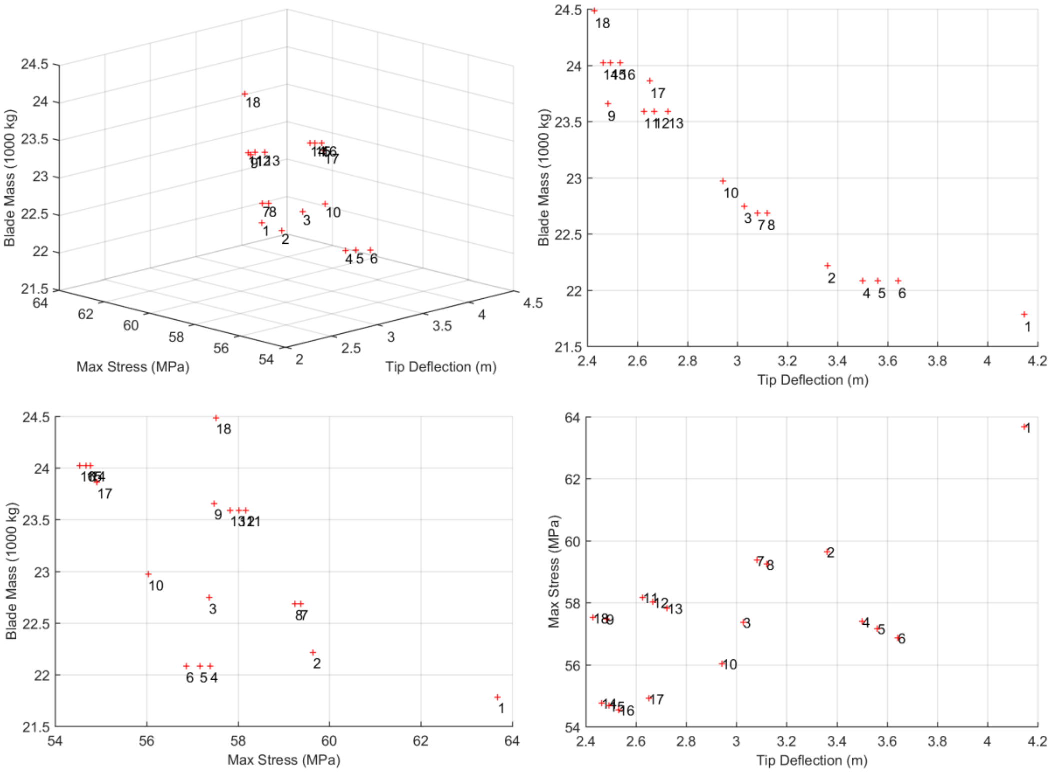


Figure 21-Pareto Solutions of Case Study 2

Table 8- Pareto Solutions of Case Study 2

Pareto	Design Variables						Constraint	Objectives		
Solution	X_{web} (%c)	Δ (%c)	n_1	n_2	n_3	n_4	β_T (°)	δ (m)	σ (MPa)	m_b (1000 kg)
1*	0.25	0.20	10	30	60	60	2.05	4.15	63.68	21.78*
2	0.25	0.25	10	30	50	50	1.98	3.36	59.64	22.22
3	0.25	0.30	10	30	50	50	2.05	3.03	57.37	22.75
4	0.30	0.25	10	30	60	70	1.95	3.50	57.39	22.08
5	0.30	0.25	10	30	60	80	1.98	3.56	57.17	22.08
6	0.30	0.25	10	30	60	90	2.01	3.64	56.87	22.08
7	0.30	0.30	10	30	60	60	2.00	3.08	59.37	22.68
8	0.30	0.30	10	30	60	70	2.03	3.12	59.25	22.68
9	0.30	0.40	10	30	50	50	1.96	2.48	57.48	23.66
10	0.35	0.35	10	30	60	90	1.97	2.94	56.04	22.97
11	0.35	0.40	10	30	60	70	1.95	2.63	58.17	23.59
12	0.35	0.40	10	30	60	80	1.99	2.67	58.02	23.59
13	0.35	0.40	10	30	60	90	2.03	2.72	57.83	23.59
14	0.35	0.45	10	30	60	60	1.99	2.46	54.77	24.02
15	0.35	0.45	10	30	60	70	2.02	2.49	54.68	24.02
16*	0.35	0.45	10	30	60	80	2.05	2.53	54.54*	24.02
17	0.40	0.45	10	30	60	90	1.96	2.65	54.91	23.86
18*	0.40	0.50	10	30	60	70	1.95	2.43*	57.52	24.48

5.3 Case Study 3-A simplified multiobjective integrated design

In an integrated design process, the design variables from different design phases are involved. In this case study, starting from the blade of Example 1 as the baseline, we perform a simplified integrated design by refining the rotor radius, chord length and the location of the webs. The selected variables for this case study, traditionally, are obtained at wind turbine conceptual design phase (rotor diameter), blade aerodynamic design phase (chord distribution) and blade structural design phase (web locations) respectively. It should be noted that the objective of this case study is to show how WTBM can facilitate an integrated design. Therefore the emphasis is on automated blade modelling rather than the optimisation process itself. The optimisation problem is formulated in the form of a refinement process, in which the design variables are limited to narrow boundaries.

$$min\,Y = f(x) \qquad (13.a)$$

s.t.

$$x_{web,1,b} - 0.10c \leq x_{web,1} \leq x_{web,1,b} + 0.10c \qquad (13.b)$$
$$x_{web,2,b} - 0.15c \leq x_{web,2} \leq x_{web,2,b} + 0.10c \qquad (13.c)$$
$$2R_b - 1.5 \leq 2R \leq 2R_b + 1.5 \qquad (13.d)$$
$$0.95 \leq c_{scale} \leq 1.05 \qquad (13.e)$$

where, $X = \{R, c_{scale}, x_{web,1}, x_{web,2}\}$ is the vector of design variables, $Y = \{m_{blade}, \sigma_{max}, \frac{1}{P}\}$ is the vector of objectives, $x_{web,i}$ is the location of the $i\text{-}th$ web measured from the leading edge, c_{scale} is the chord scaling factor, and P is the rotor mechanical power at a reference wind speed (here, 10 m/s). Subscript ***b*** stands for the baseline. It is assumed that the chord distribution along the blade span remains constant. That is, the new chord distribution can be presented as the baseline chord multiply by a scaling factor c_{scale}:

$$c = c_{scale}c_b \qquad (14)$$

Using the Optimiser and the Evaluator of Case Study 2 (Figure 20) with the new set of design variables, objectives and constraints, the Pareto solutions are obtained. The optimisation parameters population size, probability of crossover and mutation, and the total number of generations are set as 40, 0.3, 0.1 and 40 respectively. Amongst all Pareto solutions, 48 of them are superior to the baseline blade in terms of all three criteria: (i) maximum stress in the blade, (ii) rotor power and (iii) mass of the blade. These solutions are shown in Figure 22.

Three extreme solutions possessing the best performance measures in each criterion are shown in Table 9. For instance, Solution #31 is the best in terms of the blade mass. Compared to the baseline blade, blade #31 is lighter by 3.24%, it has a longer lifespan as the maximum stress is less by 3.52%, and it produces the same amount of power.

Table 9- Best solutions w.r.t. reduction in blade mass, reduction in maximum stress and increase in rotor power

Best Solution w.r.t.	Rotor Diameter (m)	Chord Scale Factor (-)	Location of Web 1(%c)	Location of Web 2 (%c)	Max von Mises Stress (Mpa)	Rotor Power (MW)	Blade Mass (kg)
Blade Mass (#31)	126.26	0.97	20.32	60.00	286 [-3.52%]	4.304 [0.09%]	22258 [-3.24%]
Max Stress (#12)	126.06	0.99	20.38	44.78	282 [-4.76%]	4.303 [0.03%]	22977 [-0.11%]
Rotor Power (#39)	127.00	0.98	20.00	60.00	292 [-1.51%]	4.369 [1.59%]	22658 [-1.5%]

In this table, values in the rectangular brackets are the relative changes with respect to the baseline blade.

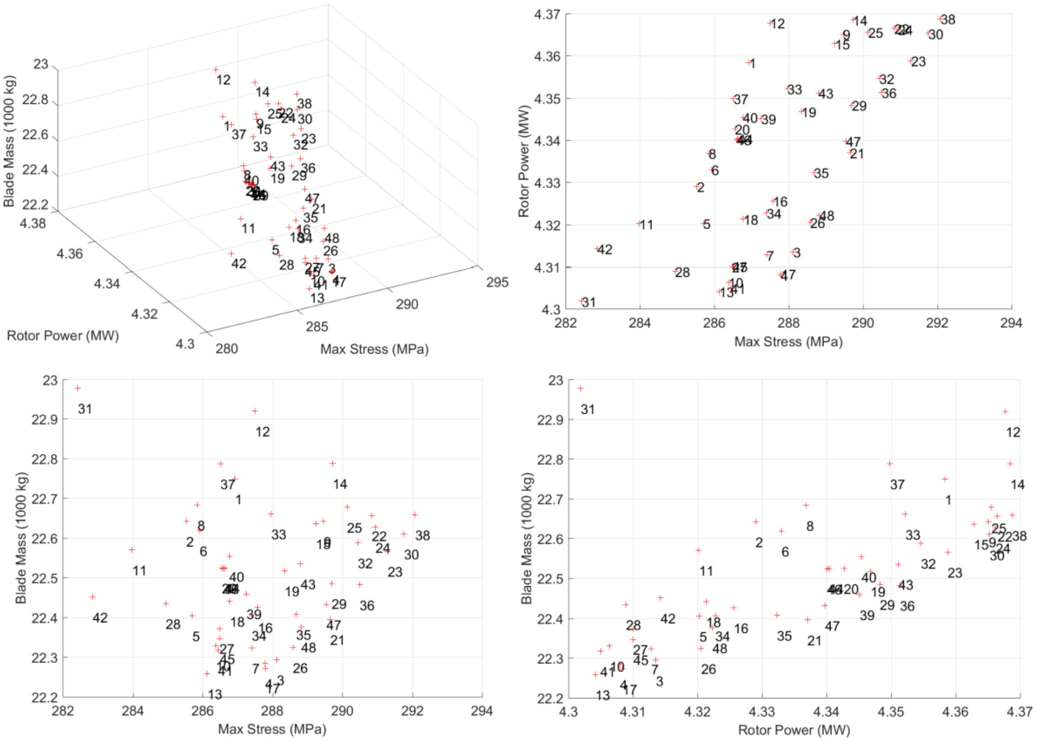


Figure 22-Pareto Solutions of Case Study 3

The baseline turbine NREL 5MW is the result of a conventional designer-in-the-loop design process. The results of this case study (see the last three columns of Table 9) evidently indicate how blade modellers like WTBM make it possible to use FEA tools such as ANSYS within an integrated design and produce superior solutions to those obtained by conventional design methods. The Pareto solutions obtained in this case study are the results of more than 1100 automated high fidelity analyses including blade modelling, pre and post processing. Obviously, more improvements can be achieved by including more design variables, such as layup configurations, into the integrated design optimisation.

5.4 A note on the overall computational time

The overall computational time for a design optimisation problem strongly depends on the robustness of the optimiser and therefore the total number of design candidate evaluations as well as the computational time per evaluation. Table 10 shows the CPU times for the three case studies when ran on a desktop machine with an Intel i5-4590 CPU @ 3.3 GHz and 8GB RAM.

Table 10- Computational times

Case Study	Average CPU time (s)			Total number of analyses [total time (hr)]
	WTBM [including WTSim]	ANSYS solver	Total (including reading/writing files)	
1	4.732 [3.871]	6.178	12.622	828 [2.9]
2	1.160 [0.153]	7.361	10.121	1440 [4.0]
3	4.436 [3.253]	7.222	13.198	1160 [4.2]

The computational time per evaluation depends on:

- The type of wind turbine. The computational time for the aerodynamic code WTSim depends on the type of the controller installed on wind turbine. When the control parameters (e.g. rotor speed and pitch angle in case of a variable speed pitch controlled wind turbine) are known WTSim requires a fraction of a second to simulate the aerodynamic performance of the wind turbine, otherwise WTSim requires to simulate the controller as well. It can take up to about 4 seconds to simulate the controller itself.

- The number of patches used to model the blade affects the computational time required for modelling and discretisation within WTBM.
- The mesh size clearly affects the computational time required by the ANSYS solver.

5.5 A note on mesh reassignment and mesh refinement

In WTBM, the user defines an initial mesh size, a scaling factor and convergence criteria for automated mesh refinement. Before writing the APDL files, WTBM controls the initial mesh size defined for each patch by the user. If the specified mesh size leads to generation of improper mesh with very high aspect ratios, adopting the same approach as reported in [44], WTBM reassigns a mesh size which is calculated based on the minimum size of the patch. This mesh adaption can lead to mesh sizes smaller or larger than the initial values.

The same iterative process shown in the flowchart of Figure 1 can be used for an automated mesh refinement process, in which the blade aerodynamic and structural performance measures are replaced by the convergence control parameters and the optimisation termination criteria are replaced with the convergence criteria:

$$\left|\frac{\varphi_i - \varphi_{i-1}}{\varphi_{i-1}}\right| \leq \epsilon_\varphi, \forall\varphi \in \{convergence\ criteria\} \tag{15}$$

The results of the automated mesh refinement process for Example 1 of Section 4 is reported here for more clarifications:

In this example two convergence criteria are defined as follows:

$$\left|\frac{\delta_{T,i} - \delta_{T,i-1}}{\delta_{T,i-1}}\right| \leq 0.001 \tag{16.a}$$

and

$$\left|\frac{\sigma_{max,i} - \sigma_{max,i-1}}{\sigma_{max,i-1}}\right| \leq 0.01 \tag{16.b}$$

where δ_T and σ_{max}, respectively, are the blade tip deflection and the maximum von Mises stress in the blade and the index i stands for the iteration number. In the current version of WTBM, the mesh refinement scales down the mesh size all through the blade. A scale factor of 0.95 is used for this example. Starting with a uniform mesh size of 0.5 for all patches in the blade, Table 11 shows the results of the mesh reassignment and then refinement till the convergence criteria (16) are satisfied.

Table 11-Automatic change of the initial mesh size

Patch #	Initial Mesh Size	Adapted Mesh Size to Patch Size	Refined Mesh Size
1	0.500	0.500	0.450
2	0.500	0.108	0.097
3	0.500	0.084	0.076
4	0.500	0.056	0.051
5	0.500	0.035	0.032
6	0.500	0.430	0.387
7	0.500	0.338	0.304
8	0.500	0.226	0.203
9	0.500	0.142	0.128
10	0.500	0.645	0.581
11	0.500	0.507	0.456
…	…	…	…
44	0.500	0.710	0.639
Total No. of DOF	31224	123534	136824

6 Summary and conclusion

Wind energy industry and research community rely on both specialised software tools and well-established and tested general purpose engineering packages such as ANSYS or ABAQUS for analysis and design of wind turbine blades. Blade modelling, pre- and post-processing in general purpose engineering packages are generally time consuming processes and require high level of skills. Using the general purpose engineering packages for high fidelity analysis within integrated and multiobjective optimisation process is not the current practice as the current blade modelling software tools fall short in delivering a fully automated pre and post processing. WTBM, presented in this paper is a fully automated blade modeller and fills the current gap. Moreover, it is a well-known fact that increasing the share of high fidelity analysis at the early stages of a design process leads to superior solutions and it may even reduce the overall cost of the design process by avoiding costly iterations. Automated blade modellers like WTBM facilitate bringing the high fidelity design optimisation towards the earlier stages of the design process.

The theory behind the automated blade modeller WTBM and its capability in (i) automated blade modelling: defining geometry, material and structural characteristics, (ii) automated pre-processing: discretising the domain, calculating and applying forces, meshing, setting the solver parameters, and generating associated APDL files for ANSYS, and (iii) automated post-processing: producing APDL files which extract performance measures in the context of an optimisation process, are presented in this paper. Its ease of use and its flexibility in the modelling of nonconventional blades (i.e. swept back blades) are shown by illustrative Examples 1 and 2 in Section 4. The performance of WTBM-ANSYS in conducting hundreds of automated high fidelity analyses is shown through three case studies in Section 5. Using WTBM, one can treat any parameter which is required to define the size, topology, structure and material of a blade as a design variable and change them automatically within an optimisation process and therefore conduct an integrated design optimisation. To the best of the author's knowledge, the multiobjective integrated design of Case study 3 in this paper is the first reported integrated design based on high-fidelity multiobjective optimisation in the literature. The Pareto solutions obtained in this case study are the results of more than 1100 blade modelling, pre-processing and post-processing. This case study also highlights the importance of adopting multi-objective optimisation and integrated design approach in design of superior blades.

The application of WTBM-ANSYS in is not limited by the complexity of the optimisation problem formulation. Moving from the reported simple case studies to a complete integrated design case is just a matter of choice of the optimisation problem formulation and the optimisation method. A multiobjective integrated design with tens of design parameters, including layup configuration and material distribution as well as distributed parameters defining the topology of the blade (chord, pretwist and aerofoil), can be formulated and conducted, more or less, as easily as the reported case studies in this paper.

Noteworthy limitations of the current version of WTBM, which are the subject of future development, are: neglecting the gyroscopic effects compared to other forces; possible inaccuracies in the calculation of the centrifugal forces in highly flexible blades; and using a uniform scaling factor for the entire blade instead of a localised mesh refinement. It should be noted that the reported case studies deal with the design scenarios in which the blade is subjected to steady loading. Inclusion of high fidelity aeroelastic/servo-aeroelastic analysis of blades subjected to dynamic loading within an integrated design optimisation is hardly justifiable due to significant computational power requirements.

References

1. C.M. Chan, H.L. Bai, D.Q. He (2018). Blade shape optimization of the Savonius wind turbine using a genetic algorithm. Applied Energy, Volume 213, Pages 148-157
2. Ayman A. Nada, Ali S. Al-Shahrani (2017). Shape Optimization of Low Speed Wind Turbine Blades using Flexible Multibody Approach. Energy Procedia, Volume 134, Pages 577-587
3. Xin Shen, Hong Yang, Jinge Chen, Xiaocheng Zhu, Zhaohui Du (2016). Aerodynamic shape optimization of non-straight small wind turbine blades. Energy Conversion and Management, Volume 119, Pages 266-278
4. M. Tahani, T. Maeda, N. Babayan, S. Mehrnia, M. Masdari (2017). Investigating the effect of geometrical parameters of an optimized wind turbine blade in turbulent flow. Energy Conversion and Management, Volume 153, Pages 71-82
5. Damir Vučina, Ivo Marinić-Kragić, Zoran Milas (2016). Numerical models for robust shape optimization of wind turbine blades. Renewable Energy, Volume 87, Part 2, Pages 849-862
6. Sukanta Roy, Ranjan Das, Ujjwal K. Saha (2018). An inverse method for optimization of geometric parameters of a Savonius-style wind turbine. Energy Conversion and Management, Volume 155, Pages 116-127
7. Matt Kear, Ben Evans, Rob Ellis, Sam Rolland (2016). Computational aerodynamic optimisation of vertical axis wind turbine blades. Applied Mathematical Modelling, Volume 40, Issue 2, Pages 1038-1051
8. Arash Hassanzadeh, Armin Hassanzadeh Hassanabad, Abdolrahman Dadvand (2016). Aerodynamic shape optimization and analysis of small wind turbine blades employing the Viterna approach for post-stall region. Alexandria Engineering Journal, Volume 55, Issue 3, Pages 2035-2043
9. Ivo Marinić-Kragić, Damir Vučina, Zoran Milas (2018). Numerical workflow for 3D shape optimization and synthesis of vertical-axis wind turbines for specified operating regimes. Renewable Energy, Volume 115, Pages 113-127
10. M. Jafaryar, R. Kamrani, M. Gorji-Bandpy, M. Hatami, D.D. Ganji (2016). Numerical optimization of the asymmetric blades mounted on a vertical axis cross-flow wind turbine. International Communications in Heat and Mass Transfer, Volume 70, Pages 93-104
11. Ricardo Luiz Utsch de Freitas Pinto, Bruna Patrícia Furtado Gonçalves (2017). A revised theoretical analysis of aerodynamic optimization of horizontal-axis wind turbines based on BEM theory. Renewable Energy, Volume 105, Pages 625-636

12. Mojtaba Tahani, Narek Babayan, Seyedmajid Mehrnia, Mehran Shadmehri (2016). A novel heuristic method for optimization of straight blade vertical axis wind turbine. Energy Conversion and Management, Volume 127, Pages 461-476
13. Chalothorn Thumthae (2015). Optimum Blade Profiles for a Variable-Speed Wind Turbine in Low Wind Area. Energy Procedia, Volume 75, Pages 651-657
14. Jerson R.P. Vaz, David H. Wood (2016). Aerodynamic optimization of the blades of diffuser-augmented wind turbines. Energy Conversion and Management, Volume 123, Pages 35-45
15. Hamid Ahmadi Asl, Reza Kamali Monfared, Manouchehr Rad (2017). Experimental investigation of blade number and design effects for a ducted wind turbine. Renewable Energy, Volume 105, Pages 334-343
16. Z.J. Chen, K.A. Stol, B.R. Mace (2017). Wind turbine blade optimisation with individual pitch and trailing edge flap control. Renewable Energy, Volume 103, Pages 750-765
17. Alejandro Albanesi, Facundo Bre, Victor Fachinotti, Cristian Gebhardt (2018). Simultaneous ply-order, ply-number and ply-drop optimization of laminate wind turbine blades using the inverse finite element method. Composite Structures, Volume 184, Pages 894-903
18. Toohid Bagherpoor, Li Xuemin (2017). Structural Optimization Design of 2MW Composite Wind Turbine Blade. Energy Procedia, Volume 105, Pages 1226-1233
19. Philipp Ulrich Haselbach (2017). An advanced structural trailing edge modelling method for wind turbine blades. Composite Structures, Volume 180, Pages 521-530
20. Lin Wang, Athanasios Kolios, Takafumi Nishino, Pierre-Luc Delafin, Theodore Bird (2016). Structural optimisation of vertical-axis wind turbine composite blades based on finite element analysis and genetic algorithm. Composite Structures, Volume 153, Pages 123-138
21. R.H. Barnes, E.V. Morozov (2016). Structural optimisation of composite wind turbine blade structures with variations of internal geometry configuration. Composite Structures, Volume 152, Pages 158-167
22. R.H. Barnes, E.V. Morozov, K. Shankar (2015). Improved methodology for design of low wind speed specific wind turbine blades. Composite Structures, Volume 119, Pages 677-684
23. Alejandro Albanesi, Victor Fachinotti, Ignacio Peralta, Bruno Storti, Cristian Gebhardt (2017). Application of the inverse finite element method to design wind turbine blades. Composite Structures, Volume 161, Pages 160-172
24. Jin Chen, Quan Wang, Wen Zhong Shen, Xiaoping Pang, Xiaofeng Guo (2013). Structural optimization study of composite wind turbine blade. Short communication. Materials & Design, Volume 46, Pages 247-255
25. Neil Buckney, Alberto Pirrera, Steven D. Green, Paul M. Weaver (2013). Structural efficiency of a wind turbine blade. Thin-Walled Structures, Volume 67, Pages 144-154
26. Fangfang Song, Yihua Ni, Zhiqiang Tan (2011). Optimization Design, Modeling and Dynamic Analysis for Composite Wind Turbine Blade. Procedia Engineering, Volume 16, Pages 369-375
27. Andrea Dal Monte, Stefano De Betta, Marco Raciti Castelli, Ernesto Benini (2017). Proposal for a coupled aerodynamic–structural wind turbine blade optimization. Composite Structures, Volume 159, Pages 144-156
28. Abolfazl Pourrajabian, Peyman Amir Nazmi Afshar, Mehdi Ahmadizadeh, David Wood (2016). Aero-structural design and optimization of a small wind turbine blade. Renewable Energy, Volume 87, Part 2, Pages 837-848
29. C.P van Dam, E. Mayda, D. Chao, K. Jackson, M. Zuteck, D. Berry (2005). Innovative Structural and Aerodynamic Design Approaches for Large Wind Turbine Blades. AIAA-2005-0973, Proceedings, ASME/AIAA Wind Energy Symposium, Reno, NV.
30. Long Wang, Tongguang Wang, Jianghai Wu, Guoping Chen (2017). Multi-objective differential evolution optimization based on uniform decomposition for wind turbine blade design. Energy, Volume 120, Pages 346-361
31. Xin Shen, Jin-Ge Chen, Xiao-Cheng Zhu, Peng-Yin Liu, Zhao-Hui Du (2015). Multi-objective optimization of wind turbine blades using lifting surface method. Energy, Volume 90, Part 1, Pages 1111-1121

32. Gunter Reinald Fischer, Timoleon Kipouros, Anthony Mark Savill (2014). Multi-objective optimisation of horizontal axis wind turbine structure and energy production using aerofoil and blade properties as design variables. Renewable Energy, Volume 62, Pages 506-515

33. Turaj Ashuri, Michiel B. Zaaijer, Joaquim R.R.A. Martins, Jie Zhang (2016). Multidisciplinary design optimization of large wind turbines—Technical, economic, and design challenges. Energy Conversion and Management, Volume 123, Pages 56-70

34. Wenbin Yu, Vitali V. Volovoi, Dewey H. Hodges, and Xianyu Hong (2002). "Validation of the Variational Asymptotic Beam Sectional Analysis", AIAA Journal, Vol. 40, Pages 2105-2112.

35. Hui Zhang, Alireza Maheri (2014). A Software Tool for Structural and Material Configurations of Wind Turbine Adaptive Blades. In Third Int Symposium On Environment Friendly Energies and Applications EFEA 2014, 19-21 November 2014, Paris.

36. Pietro Bortolotti, Carlo L Bottasso, Alessandro Croce (2016). Combined preliminary-detailed design of wind turbines, Wind Energy Science, Vol 1, Pages 71-88, doi:10.5194/wes-1-71-2016

37. M. Peetersa , W. Van Paepegemb (2014). Development of automated high-fidelity finite element models for large wind turbine blades, in 16th European Conference on Composite Materials, Seville, Spain, 22-26 June 2014

38. Alireza Maheri, Siamak Noroozi, Chris Toomer, John Vinney (2006). WTAB, a computer program for predicting the performance of horizontal axis wind turbines with adaptive blades. Renewable Energy, Volume 31, Pages 1673-1685

39. Bonnet, P. and G. Dutton, Parametric Modelling Of Large Wind Turbine Blades, in 2007 Abaqus UK Regional User Meeting.

40. Laird DL and T Ashwill, Introduction to NuMAD: A Numerical manufacturing and design tool proceedings of the ASME/AIAA and Wind Energy Symposium, Reno, NV, 1998, pp. 354-360

41. John Montesano, Hao Chu, Chandra Veer Singh (2016). Development of a physics-based multi-scale progressive damage model for assessing the durability of wind turbine blades, Composite Structures, Volume 141, Pages 50-62

42. Alireza Maheri (2016). Simulation of Wind Turbines Utilising Smart Blades. Journal of Thermal Engineering, Volume 2, Pages 557-565

43. Terence Macquart, Alireza Maheri (2015). Integrated Aeroelastic and Control Analysis of Wind Turbine Blades Equipped with Microtabs. Renewable Energy, Volume 75, Pages 102-114

44. Alireza Maheri (2012). A finite element suite for deformation analysis of composite aeroelastic structures subjected to operational aerodynamic loading. Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, Volume 226 (8), Pages 2062-2076.

45. J. Jonkman, S. Butterfield, W. Musial, and G. Scott (2009). Definition of a 5-MW Reference Wind Turbine for Offshore System Development, Technical Report NREL/TP-500-38060

46. Brian R. Resor (2013). Definition of a 5MW/61.5m Wind Turbine Blade Reference Model, SANDIA Report, SAND2013-2569

47. Alireza Maheri, Siamak Noroozi, John Vinney (2007). Application of combined analytical/FEA coupled aero-structure simulation in design of wind turbine adaptive blades. Renewable Energy, Volume 32 (12), Pages 2011-2018

48. Alireza Maheri, Askin Isikveren (2010). Performance prediction of wind turbines utilizing passive smart blades: approaches and evaluation. Wind Energy, Volume 12, Pages 255-265.

49. Alireza Maheri, Siamak Noroozi, John Vinney (2007). Decoupled aerodynamic and structural design of wind turbine adaptive blades. Renewable Energy, Volume 32, Pages 1753-1767

50. Alireza Maheri, Askin Isikveren (2008). Variable-state design parameters in design of aero-structures made of intrinsically smart materials. In: High Performance Structures and Materials IV. WIT Transactions on the Built Environment. WIT Press, Southampton, UK, Pages 421-430

Appendix

```
/Modelling and Pre-processing
MP,EX, 1,32300000000.000
MP,EY, 1,7207800000.000
.......
SECTYPE, 1,SHELL
SECDATA,0.00005, 3,0.000, 3
SECDATA,0.00047, 5,45.000, 3
....
ET, 1,181
K, 1,-0.19958,0.84376,0.00000
....
A, 1,36,34, 2
AATT,, 1, 1,0, 1
MSHKEY, 2
ESIZE,0.50000
AMESH, 1, 1,1
....
SOLCONTROL,1,1,NOPL
KBC, 5
KSEL,S,LOC,Z,0,0
DK,ALL,UX,0
....
NSLK,S

AUTOTS,ON
KSEL,S,KP,,1, 1,1
NSLK,S
FK, 1,FX,5.57734
....
```

Figure A1-APDL files 1st part: Blade modelling and pre-processor setting

```
/Get and Save Results
*GET,NONDELE,NODE,0,NUM,MAX
NSEL,S,NODE,,1,NONDELE
*SET,RESULTS
*DIM,RESULTS,ARRAY,NONDELE,12
*DO,II,1,NONDELE
*GET,LOCX,NODE,II,LOC,X
*GET,UX,NODE,II,U,X
*GET,S1,NODE,II,S,1
....
*VFILL,RESULTS(II,1),DATA,LOCX
....
*ENDDO
SAVE....
*MWRITE ....
```

Figure A2-APDL files 2nd part: Commands ANSYS to write the results in a text file