Nonlinear Analysis of Wind Turbine Blades Using Finite Elements with Anisotropic Variable Kinematics

Sander van den Broek*, Mayank Patni †, Aewis Hii ‡ , Paul Weaver §
University of Bristol, University Walk, Bristol BS8 1TR, United Kingdom

Peter Greaves¶
Offshore Renewable Energy Catapult, Albert Street, Blyth NE24 1LZ, United Kingdom

Alberto Pirrera‖
University of Bristol, University Walk, Bristol BS8 1TR, United Kingdom

Analysis of wind turbine blades using beam or shell models presents difficulties in accurately capturing the torsional stiffness and local 3D stress fields. Instead, modeling torsional effects accurately often necessitates three-dimensional analysis as achieved with solid elements in finite element analysis. The use of solid elements and complex local mesh refinement algorithms are often required to capture the three-dimensional stress fields in critical regions, which results in systems with a large number of degrees of freedom. The present work proposes using variable kinematics finite elements to analyze wind turbine blades. Variable kinematic elements use a higher-order shape function to represent the displacement field in an element, enabling a more refined kinematic description of displacements. Previous works have shown that higher-order elements with variable kinematics can obtain accurate 3D stress fields with fewer degrees of freedom than conventional solid models. Using p-refinement furthermore allows for local refinement without requiring remeshing. By allowing the kinematics to be directional, the accuracy and degrees of freedom can be tailored to be closely related to the structure.

I. Nomenclature

e_i = Edge number i
F_i = Face number i
$\mathbf{h}$ = Displacement field shape function
H = Shape function mode
L_n = Legendre polynomial of order n
Λ = Array with evaluated affine coordinate terms
λ_j = Affine coordinate j
$\mathbf{n}$ = Geometry shape function
p = Order of expansion
$\mathbf{\Phi}$ = Array with evaluated Lobatto shape functions ϕ
ϕ_n = One-dimensional Lobatto shape function of order n
$\mathbf{x}_e$ = Coefficients representing the element's position in space

II. Introduction

WIND turbine blades have steadily grown over the decades to increase the capacity of wind turbines, with blades exceeding 115 meters currently being prototyped. Increasing the length of turbine blades leads to increased

*Research Associate, Bristol Composites Institute
†Honorary Research Associate, Bristol Composites Institute
‡Research Associate, Bristol Composites Institute
§Professor in Lightweight Structures, Bristol Composites Institute
¶Principal R&D Engineer – Blade Structures
‖Professor of Nonlinear Structural Mechanics, Bristol Composites Institute

structural complexity from a mechanical and material perspective. Complexities arise due to the more considerable deformation of the blade and corresponding nonlinear effects such as cross-sectional warping [1].

Preliminary design of wind turbines in practice uses simple models based on beams. Detailed designs are then analyzed using shell models [2]. Areas that can not be modeled accurately using shell elements are then submodeled using loads and boundary conditions extrapolated from the shell model, often using assumptions that significantly affect results.

Using multiple types of structural models requires separate meshes and mesh refinement studies. Creating meshes from geometry and making sure results converge is time-consuming, often taking more time than the actual analysis.

Another area for improvement with traditional design approaches is that shell models can not accurately represent certain aspects of blade design. Assumptions in shell models do not facilitate modeling certain geometric features, such as the adhesive interface at the spar caps. In addition, the torsional stiffness computed using shell elements has a significant disparity compared to experimental results [3, 4]. Work done by Tavares et al. [5] has shown that the torsional stiffness can be improved by including drilling rotation degrees of freedom, though solid elements are needed to converge to experimental results. Other types of analyses also require solid elements, such as delamination modeling, where through-thickness stresses are required.

Modeling an entire wind turbine blade using solid elements requires a fine mesh with multiple elements through the material thickness to properly represent the kinematics and converge to the correct 3D stress state. Using solid elements requires more time in mesh preparation and significantly increases the degrees of freedom in the finite element model, potentially becoming computationally prohibitive.

The work presented in this paper proposes an alternative workflow for the structural analysis of wind turbine blades. Instead of being dependent on generating and converging different meshes, each focusing on a specific type of analysis, the proposed approach uses a single unified mesh. A unified mesh is used to represent the geometry of the wind turbine as closely as possible.

Structural finite element analysis converges towards a displacement field solution. Convergence of a solution requires that the discretization allows an accurate representation of the displacement field. Commercial finite element solvers usually do this by reducing the size of elements, also referred to as h-refinement. Reducing the size of elements makes it possible to approximate complex displacement fields with relatively low polynomial approximations within element subdomains. Another way to achieve convergence is to allow higher-order approximations of displacement fields in an element, also known as p-refinement. Increasing the polynomial order of elements leads to exponentially faster convergence than decreasing element sizes [6].

Using p-refinement, the shape functions of elements have varying kinematics depending on the expansion order used. As the expansion order increases, the kinematics that an element can represent become more complex. By achieving convergence using p-refinement, there is no more need to create new meshes to converge to a solution. Instead, convergence is achieved by increasing the polynomial order of the element shape function, thereby allowing the element to represent more complex displacement fields.

The variable kinematics element used allows for anisotropic expansion of the shape function. Using anisotropic variable kinematics, degrees of freedom can be selectively added to directions in which the kinematic description should refine. It is possible to, for instance, refine the in-plane direction of a thin-walled structure without adding through-thickness refinement.

Variable kinematics elements have an additional advantage: they can converge to solutions with fewer degrees of freedom than linear or quadratic elements often used in commercial finite element programs. One solid element can accurately capture the 3D stress fields of multiple laminae using an equivalent single-layer approach (ESL). 3D stresses within the element can be recovered using refined zig-zag theory with great accuracy while using orders of magnitude fewer degrees of freedom [7]. For these reasons, Vanskike [8] recently identified layered variable kinematics solid elements as a promising approach in modeling wind turbine blades.

The work presented shows an implementation of variable kinematics elements and demonstrates the performance of structures representative of wind turbine blades. Results are shown to converge with those of refined meshes in commercial finite elements while using fewer degrees of freedom.

III. Methodology

The approach adopted in this paper distinguishes between shape functions used to represent the structural geometry and shape functions used to represent the displacement field. By separating these representations, meshes can be generated using widely available algorithms that define linear, quadratic, or higher-order meshes using Serendipity or

Lagrange shape functions. Displacements can be represented using variable kinematics elements. Variable kinematics elements utilizing Legendre polynomials make it possible to increase the (local) element order without remeshing.

A. Geometric representation

The geometric representation is used to define the geometry of elements. Any differentiable 3D shape function can be used, but the accuracy of the solution will be directly affected by the accuracy of the representation. Examples in this paper utilize geometry represented by quadratic Lagrange elements, but better accuracy can be achieved using exact representations through b-splines or nonuniform b-splines (NURBS).

Point i in an element can be mapped from natural coordinates, (ξ, η, ζ), to global coordinates by the relationship

$$\mathbf{x}_i = \mathbf{n}(\xi_i, \eta_i, \zeta_i)\mathbf{x}_e, \tag{1}$$

with $\mathbf{x}_e$ being the element coefficients of the geometric shape function $\mathbf{n}$.

B. Displacement field representation

Displacement fields represent displacements of an element in space and are represented using the shape function

$$\mathbf{u}_i = \mathbf{h}(\xi_i, \eta_i, \zeta_i)\mathbf{u}_e. \tag{2}$$

The shape function $\mathbf{h}$ is a variable kinematics shape function with a variable number of terms depending on the order of expansion. Within the variable kinematic shape function, displacement on edges, faces and within the volume are enriched using a set of mutually orthogonal polynomials.

1. Lobatto shape functions

Orthogonal shape functions are constructed using Legendre polynomials and normalizing these to zero at boundaries. These shape functions are also called Lobatto shape functions ϕ, and are constructed as [9, sec. 3.3.2] [10]

$$\begin{aligned} \phi_n(\xi) &= \sqrt{\frac{2n-1}{2}} \int_{-1}^{\xi} L_{n-1}(x)\mathrm{d}x \\ &= \frac{1}{\sqrt{4n-2}} \left(L_n(\xi) - L_{n-2}(\xi) \right), \quad n = 2, 3, \ldots \end{aligned} \tag{3}$$

where L_n is the Legendre polynomial of order n. Legendre polynomials $L_n(\xi)$ are defined as solutions of the differential equation [9, App. D]

$$(1 - \xi^2)y'' - 2\xi y' + i(i+1)y = 0, \quad -1 \leq \xi \leq 1. \tag{4}$$

The first 15 Lobatto shape functions and their derivatives are listed in the appendix.

2. Kinematic expansion

Lobatto shape functions are used to expand the linear kinematic displacement field with additional terms. Terms can be categorized as:

Edge A single polynomial $\phi_m(a)$ is used to expand the kinematic description of an edge;
Face A product of two polynomials $\phi_m(a)\phi_n(b)$ is used to expand the kinematic description of a face;
Internal Three polynomials $\phi_m(a)\phi_n(b)\phi_o(c)$ are combined to describe an internal "bubble" shape function.

where m, n, o is the order of the polynomial and a, b, c the coordinates associated with the shape function.

The terms included in a shape function depend on the polynomial space used and the directional order. When multidimensional polynomials are used, the terms of a shape function can vary by the polynomial order of expansion and the terms included in the polynomial space.

The two most common spaces in defining multidimensional polynomial spaces are the trunk space $\mathcal{S}_{\text{ts}}^{p_\xi, p_\eta, p_\zeta}(\Omega_{\text{st}^h})$ and the product space $\mathcal{S}_{\text{ps}}^{p_\xi, p_\eta, p_\zeta}(\Omega_{\text{st}^h})$. Details of these spaces can be found in [9, sec. 3.2], and are summarized as:

Trunk space Within a trunk space, the total number of monomials in a shape function equals the most significant expansion order in an anisotropic polynomial expansion. The terms included can be summarized as:

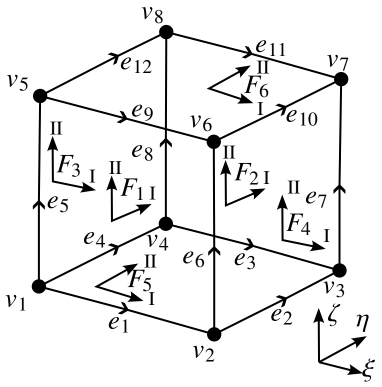


Fig. 1 Face and edge orientations on a hexahedron

Table 1 Overview of edges e, faces F and internal bubble functions I and the expansion direction that determines their expansion

	p_ξ	p_η	p_ζ
Edges	e_1, e_3, e_9, e_{11}	e_2, e_4, e_{10}, e_{12}	e_5, e_6, e_7, e_8
Faces	$F_{3,\text{I}}, F_{4,\text{I}}, F_{5,\text{I}}, F_{6,\text{I}}$	$F_{1,\text{I}}, F_{2,\text{I}}, F_{5,\text{II}}, F_{6,\text{II}}$	$F_{1,\text{II}}, F_{2,\text{II}}, F_{3,\text{II}}, F_{4,\text{II}}$
Internal	I_ξ	I_η	I_ζ

- $\xi\eta\zeta$ for $p_\eta, = p\eta = p_\zeta = 1$, corresponding to the nodal modes.
- $\xi^i\eta^j\zeta^k$, bound to $i = 0 \dots p_\eta, \quad j = 0, \dots p_\eta \quad k = 0, \dots p_\zeta$ with $i + j + k = 0, \dots \max(p_\eta, p_\eta, p_\zeta)$.
- $\xi^{p_\xi}\eta\zeta$ for $p_\xi \geq 2$, $\xi\eta^{p_\eta}\zeta$ for $p_\eta \geq 2$, and $\xi\eta\zeta^{p_\zeta}$ for $p_\zeta \geq 2$. Corresponding to edge modes.

Using a trunk space, therefore, means that internal bubble modes are not included until $i + j + k \geq 6$.

Product space The product space spans all polynomials $\xi^i\eta^j\zeta^k$ with $i = 0, \dots p_\xi$, $j = 0, \dots p_\eta$, and $k = 0, \dots p_\zeta$. Therefore, in a product space, face nodes and internal modes are present as soon as the expansion in all related directions is of order two or higher.

In an anisotropic kinematic expansion, the order assigned to edges, faces, and internal bubble functions varies by direction. Using the nomenclature of Fig. 1 and an expansion polynomial of p_ξ, p_η, p_ζ, the highest order of polynomials used can be determined. In a hexahedron, the expansion of edge e_9 would equal p_ξ. Surface F_3 has two polynomials, the first p_I equals p_ξ, the second $p_{II} = p_\zeta$. Bubble shape functions vary the same way, with the highest polynomials determined by each expansion order p in the natural coordinate system. An overview of the kinematic determination of each element edge and face is shown in Table 1. The number of terms of the shape function, categorized by type, is given analytically in Table 2.

C. Numerical implementation

In order to effectively deal with varying polynomial expansions, it is essential to implement certain aspects of the finite element solver more intricately than in traditional finite element formulations. In general, meshes are only sometimes structured so that all element orientations are identical (e.g., face three does not only face four in a hexahedron shown in Fig. 1). In an anisotropic polynomial expansion, it is, therefore, necessary to thoroughly index the highest polynomial that is assigned to each edge and face.

Upon evaluation of an element's shape function, all terms associated with an edge or face must be considered. As such, elements might have to transition between different orders of expansion, requiring, for instance, $p_{F3,\text{I}} = p_{F3,\text{II}} = 4$, while the same element might have $p_{F4,\text{I}} = p_{F4,\text{II}} = 5$. The transition between elements requires additional terms for face 4.

Explicitly deriving the shape function for all possible combinations of element expansions is unrealistic and would require many combinations. Therefore, it is interesting to develop a modular approach to systematically evaluate shape function terms. The developed approach is capable of both function evaluations and evaluating the function derivatives

Table 2 Number of terms for isotropic and anisotropic expansions using the polynomial trunk space. Expansion orders $p_{\mathrm{I}}, p_{\mathrm{II}}, p_{\mathrm{III}}$ are element orders p_ξ, pi_η, p_ζ sorted by element order $p_{\mathrm{I}} \geq p_{\mathrm{II}} \geq p_{\mathrm{III}}$ with 5 subtracted. Note that these formulas are only valid on a per-element basis and do not consider compatibility terms with neighboring elements.

Term	Isotropic expansion	Anisotropic expansion
Vertex	8	8
Edge	For $p \geq 2$, $12(p-1)$	$4(p_\xi - 1) + 4(p_\eta - 1) + 4(p_\zeta - 1)$
Face	For $p \geq 4$, $6\frac{(p-3)(p-5)}{2}$	Functions are only included for faces where expansions I and II are both at least 4 $2\frac{p_{\mathrm{I}}(p_{\mathrm{I}}+2)}{2} - \frac{(P_{\mathrm{I}}-P_{\mathrm{II}})(P_{\mathrm{I}}-P_{\mathrm{II}}+1)}{2}+$ $2\frac{p_{\mathrm{I}}(p_{\mathrm{I}}+2)}{2} - \frac{(P_{\mathrm{I}}-P_{\mathrm{III}})(P_{\mathrm{I}}-P_{\mathrm{III}}+1)}{2}+$ $2\frac{p_{\mathrm{I}}I(p_{\mathrm{I}}I+2)}{2} - \frac{(P_{\mathrm{II}}-P_{\mathrm{III}})(P_{\mathrm{II}}-P_{\mathrm{III}}+1)}{2}$
Internal	For $p \geq 6$, $\frac{(p-5)(p-4)(p-3)}{6}$	Internal terms are included when $p_\xi, p_\eta, p_\zeta \geq 6$ For $p_{\mathrm{II}} + p_{\mathrm{III}} - p_{\mathrm{I}} \leq 2$ $p_{\mathrm{II}} p_{\mathrm{III}} \frac{2p_{\mathrm{I}}+2-p_{\mathrm{II}}-p_{\mathrm{III}}}{2}$ For $p_{\mathrm{II}} + p_{\mathrm{III}} - p_{\mathrm{I}} > 2$ $-\frac{p_{\mathrm{I}}^3+(3(p_{\mathrm{II}}+p_{\mathrm{III}}-1)p_{\mathrm{I}}^2}{6} + \frac{p_{\mathrm{I}}(-3p_{\mathrm{II}}^2-3p_{\mathrm{III}}^2+6p_{\mathrm{II}}+6p_{\mathrm{III}}-2+p_{\mathrm{II}}^3+p_{\mathrm{III}}^3)}{6} - \frac{p_{\mathrm{II}}^2+p_{\mathrm{III}}^2}{2} + \frac{p_{\mathrm{II}}+p_{\mathrm{III}}}{3}$

to the natural (element) coordinate system.

By evaluating a finite amount of expressions, it is possible to combine these to evaluate the different terms of the shape function. ConstrucShape functions are constructed using both affine coordinates of the elements and the previously derived shape functions evaluated using specified coordinates. Evaluating derivatives also requires the derivatives of both of these expressions.

1. Affine coordinates

Affine coordinates define functions to element faces that are 0 at the specified face and one at the opposite end of the element. A product of these coordinates can create a shape function that specifies an element's vertex, edge, or face. The affine coordinates for a hexahedron are

$$\lambda_{1,H}(\xi,\eta,\zeta) = \frac{1+\xi}{2} \qquad \lambda_{2,H}(\xi,\eta,\zeta) = \frac{1-\xi}{2} \qquad \lambda_{3,H}(\xi,\eta,\zeta) = \frac{1+\eta}{2}$$
$$\lambda_{4,H}(\xi,\eta,\zeta) = \frac{1-\eta}{2} \qquad \lambda_{5,H}(\xi,\eta,\zeta) = \frac{1+\zeta}{2} \qquad \lambda_{6,H}(\xi,\eta,\zeta) = \frac{1-\zeta}{2} \tag{5}$$

2. Polynomial evaluation

Lobatto polynomials are evaluated with a set amount of coordinates. The polynomials defined in the appendix are used to create a polynomial array Φ with entries for hexahedrons

$$\Phi_H = \begin{bmatrix} \phi_1(\xi) & \phi_1(\eta) & \phi_1(\zeta) \\ \vdots & \vdots & \vdots \\ \phi_n(\xi) & \phi_n(\eta) & \phi_n(\zeta) \end{bmatrix}, \tag{6}$$

where n is the highest order polynomial evaluated.

3. Shape function evaluation and assembly

In order to evaluate a shape function, a template has to be constructed of terms included for the specific element. A template for different terms of a hexahedron element is shown in Table 3. The terms of a shape function can be described using an array with indices that uniquely describe the combination of polynomial terms. A third-order

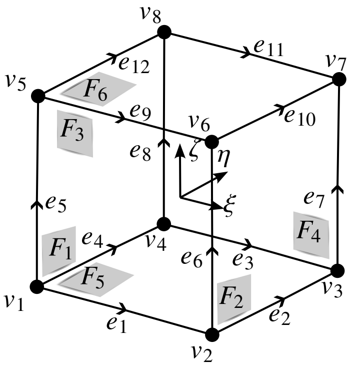


Fig. 2 Hexahedron with vertices, edges and faces identified.

polynomial term on edge four would be evaluated as $e_{4,3} = \lambda_2\lambda_6\phi_{3,2}$, and a term on face 4 with $p_{\mathrm{I}} = 2, p_{\mathrm{II}} = 3$ would be $F_{4,2,3} = \lambda_3\phi_{2,1}\phi_{3,3}$.

From this approach, it is clear that an array of terms can be assembled that is unique to a specific kinematic representation of an element. The indices of affine coordinates and polynomial terms uniquely identify the shape function term. The terms that are included can be described using an array that is constructed during preprocessing or is precomputed for a specific expansion. The latter approach should be avoided using anisotropic expansions and structures with varying expansions, as this would likely lead to kinematic incompatibilities. Dealing with differences in expansion has to be done properly using transitional elements containing a unique set of shape function terms.

An important thing to note is that shape function terms may require additional transformation between an element and global coordinate systems. Transformations can be done using sign changes and by switching terms ϕ_A and ϕ_B for faces. These transformations are unique per edge, face and element and can be precomputed or incorporated in the shape function evaluation. The necessary transformations are detailed in [11, sec. 3.5].

4. Shape function derivatives

Shape function derivatives are evaluated similarly to shape functions. A numerical implementation of the product rule is used to compute the derivative of the shape function. Terms of a shape function are organized into an $n \times n$ array, with the diagonal replaced with each term's derivative. Evaluating the derivative is done by summing the product of each row.

The derivatives for the affine coordinates are easily found as

$$\frac{\mathrm{d}\lambda}{\mathrm{d}\xi} = \begin{bmatrix} 0.5 & -0.5 & 0 & 0 & 0 & 0 \end{bmatrix} \quad \frac{\mathrm{d}\lambda}{\mathrm{d}\eta} = \begin{bmatrix} 0 & 0 & 0.5 & -0.5 & 0 & 0 \end{bmatrix} \quad \frac{\mathrm{d}\lambda}{\mathrm{d}\zeta} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0.5 & -0.5 \end{bmatrix},$$

and polynomial array as

$$\frac{\partial \Phi_H}{\partial \xi} = \begin{bmatrix} \frac{\mathrm{d}\phi_1(\xi)}{\mathrm{d}\xi} & 0 & 0 \\ \vdots & \vdots & \vdots \\ \frac{\mathrm{d}\phi_n(\xi)}{\mathrm{d}\xi} & 0 & 0 \end{bmatrix} \qquad \frac{\partial \Phi_H}{\partial \eta} = \begin{bmatrix} 0 & \frac{\mathrm{d}\phi_1(\eta)}{\mathrm{d}\eta} & 0 \\ \vdots & \vdots & \vdots \\ 0 & \frac{\mathrm{d}\phi_n(\eta)}{\mathrm{d}\eta} & 0 \end{bmatrix} \qquad \frac{\partial \Phi_H}{\partial \zeta} = \begin{bmatrix} 0 & 0 & \frac{\mathrm{d}\phi_1(\zeta)}{\mathrm{d}\zeta} \\ \vdots & \vdots & \vdots \\ 0 & 0 & \frac{\mathrm{d}\phi_n(\zeta)}{\mathrm{d}\zeta} \end{bmatrix}.$$

The first 15 derivatives of the Lobatto shape function are listed in the appendix. Organizing shape function terms into a 3D array $\mathbf{A}$ each term can be sliced into a 2D form. Using a derivative to ξ as an example, a slice for term i would have the form of

$$\mathbf{A}_i = \begin{bmatrix} \frac{\partial\lambda_A}{\partial\xi} & \lambda_B & \lambda_C & \phi_A & \phi_B & \phi_C \\ \lambda_A & \frac{\partial\lambda_B}{\partial\xi} & \lambda_C & \phi_A & \phi_B & \phi_C \\ \lambda_A & \lambda_B & \frac{\partial\lambda_C}{\partial\xi} & \phi_A & \phi_B & \phi_C \\ \lambda_A & \lambda_B & \lambda_C & \frac{\partial\phi_A}{\partial\xi} & \phi_B & \phi_C \\ \lambda_A & \lambda_B & \lambda_C & \phi_A & \frac{\partial\phi_B}{\partial\xi} & \phi_C \\ \lambda_A & \lambda_B & \lambda_C & \phi_A & \phi_B & \frac{\partial\phi_C}{\partial\xi} \end{bmatrix},$$

Table 3 Components that are part of a shape function term. Polynomial expansions on edge terms are listed as p. On face terms, two components (I & II) are given, and for face terms, three (A,B,C). Element components can be identified from Fig. 2. λ terms refer to affine coordinate terms of Eq. (6), ϕ terms refer to the entries of the polynomial array Φ, where rows are the polynomial order and columns the evaluated coordinate.

Type	λ_A	λ_B	λ_C	ϕ_A	ϕ_B	ϕ_C
v_1	λ_2	λ_4	λ_6	1	1	1
v_2	λ_1	λ_4	λ_6	1	1	1
v_3	λ_1	λ_3	λ_6	1	1	1
v_4	λ_2	λ_3	λ_6	1	1	1
v_5	λ_2	λ_4	λ_5	1	1	1
v_6	λ_1	λ_4	λ_5	1	1	1
v_7	λ_1	λ_3	λ_5	1	1	1
v_8	λ_2	λ_3	λ_5	1	1	1
e_1	λ_4	λ_6	1	$\phi_{p,1}$	1	1
e_2	λ_1	λ_6	1	$\phi_{p,2}$	1	1
e_3	λ_3	λ_6	1	$\phi_{p,1}$	1	1
e_4	λ_2	λ_6	1	$\phi_{p,2}$	1	1
e_5	λ_2	λ_4	1	$\phi_{p,3}$	1	1
e_6	λ_1	λ_4	1	$\phi_{p,3}$	1	1
e_7	λ_1	λ_3	1	$\phi_{p,3}$	1	1
e_8	λ_2	λ_3	1	$\phi_{p,3}$	1	1
e_9	λ_4	λ_5	1	$\phi_{p,1}$	1	1
e_{10}	λ_1	λ_5	1	$\phi_{p,2}$	1	1
e_{11}	λ_3	λ_5	1	$\phi_{p,1}$	1	1
e_{12}	λ_2	λ_6	1	$\phi_{p,2}$	1	1
F_1	λ_2	1	1	$\phi_{p_{\mathrm{I}},2}$	$\phi_{p_{\mathrm{II}},3}$	1
F_2	λ_1	1	1	$\phi_{p_{\mathrm{I}},2}$	$\phi_{p_{\mathrm{II}},3}$	1
F_3	λ_4	1	1	$\phi_{p_{\mathrm{I}},1}$	$\phi_{p_{\mathrm{II}},3}$	1
F_4	λ_3	1	1	$\phi_{p_{\mathrm{I}},1}$	$\phi_{p_{\mathrm{II}},3}$	1
F_5	λ_6	1	1	$\phi_{p_{\mathrm{I}},1}$	$\phi_{p_{\mathrm{II}},2}$	1
F_6	λ_5	1	1	$\phi_{p_{\mathrm{I}},1}$	$\phi_{p_{\mathrm{II}},2}$	1
I	1	1	1	$\phi_{p_A,1}$	$\phi_{p_B,2}$	$\phi_{p_C,3}$

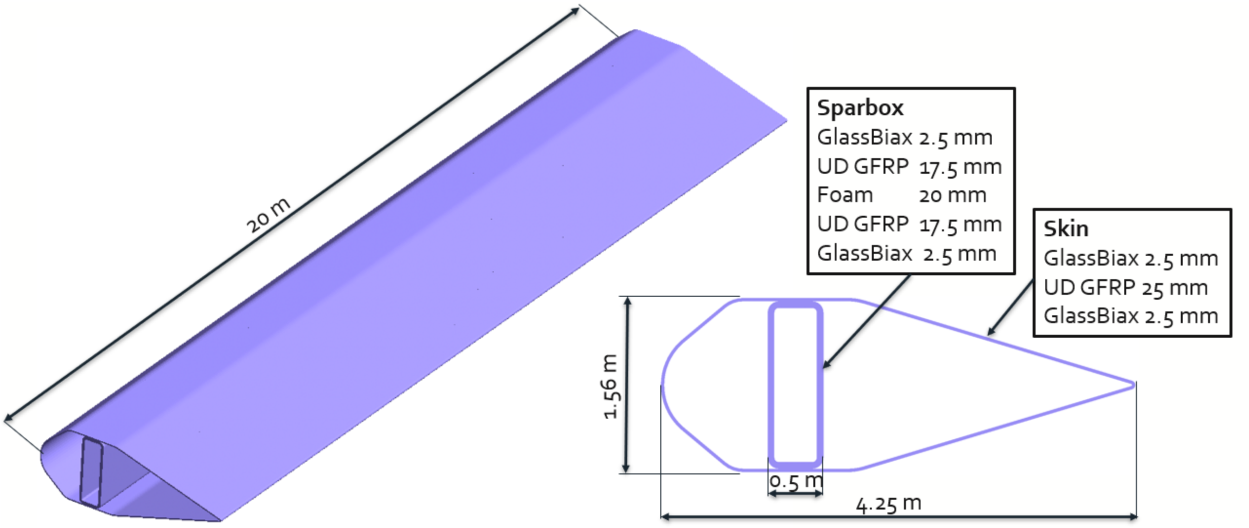


Fig. 3 Geometry and layup of simplified blade structure. UD fibers are all aligned to the span with the section

where derivative of the term would be computed as

$$\mathbf{h}_i = \sum_{n=1}^{6} \prod_{m=1}^{6} A_{i,n,m}.$$

Following this approach an example of a derivative to ξ of the term j acting on face 3 with $p_{\mathrm{I}} = 2, p_{\mathrm{II}} = 4$ can be assembled and evaluated as

$$\mathbf{A}_j = \begin{bmatrix} \frac{\partial \lambda_4}{\partial \xi} = 0 & 1 & 1 & \phi_{2,1} & \phi_{4,3} & 1 \\ \lambda_4 & 0 & 1 & \phi_{2,1} & \phi_{4,3} & 1 \\ \lambda_4 & 1 & 0 & \phi_{2,1} & \phi_{4,3} & 1 \\ \lambda_4 & 1 & 1 & \frac{\partial \phi_{2,1}}{\partial \xi} & \phi_{4,3} & 1 \\ \lambda_4 & 1 & 1 & \phi_{2,1} & \frac{\partial \phi_{4,3}}{\partial \xi} & 1 \\ \lambda_4 & 1 & 1 & \phi_{2,1} & \phi_{4,3} & 0 \end{bmatrix} \rightarrow h_j = \lambda_4 \frac{\partial \phi_{2,1}}{\partial \xi} \phi_{4,3} + \lambda_4 \phi_{2,1} \frac{\partial \phi_{4,3}}{\partial \xi}.$$

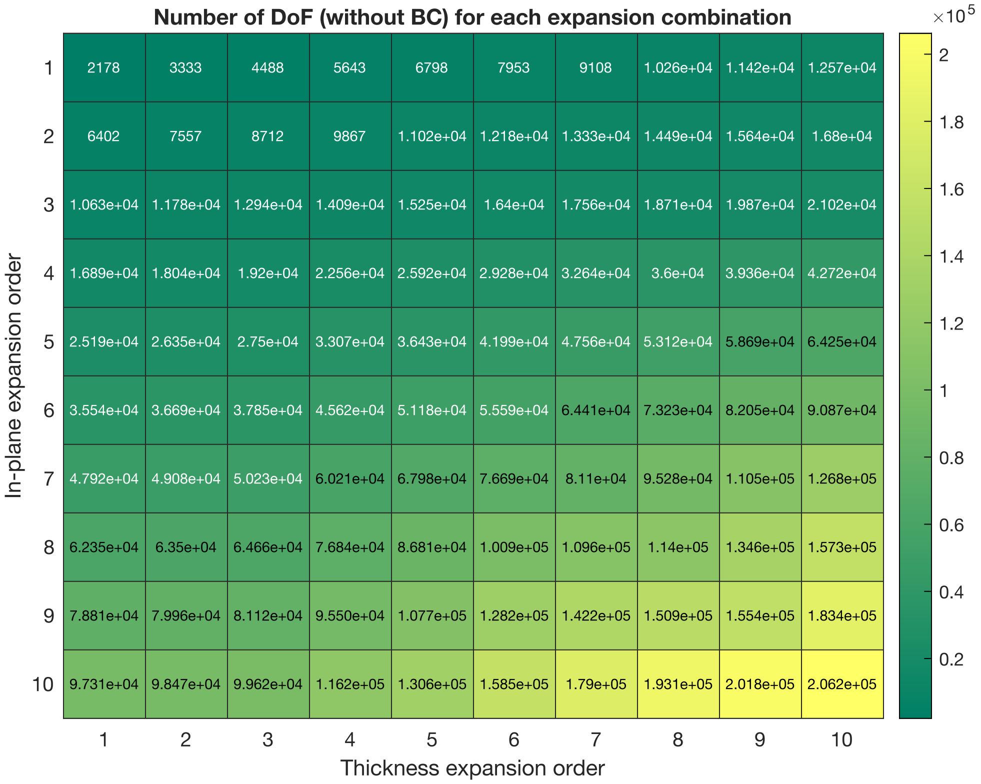
IV. Numerical example

To demonstrate the higher-order approach, a numerical example is analyzed. Analyses were performed on a representative structure that has a similar, though simplified geometry as a wind turbine blade. Two different layups are used, one for the skin and one for the spar. Geometry and layup details can be found in Fig. 3, and material properties are listed in Table 4.

The blade section is fully clamped at coordinate $z = 0$, and loaded by traction on surface of the spar box with a traction force of 9 MPa, totaling 1 747 kN. Displacements are solved through linear analysis as well as through geometrically nonlinear analysis using a Newton-Raphson solver.

The mesh in the higher-order implementation consists of elements with a seeded length of 2 meters, totaling 350 elements. Composite layups are incorporated through a smeared approach using integrated layered stiffness properties of an element. The geometric representation of the mesh is generated in Abaqus using quadratic elements. The number of degrees of freedom varies depending on the kinematic expansion. Figure 4 shows the number of degrees of freedom depending on each expansion.

As a baseline result fine Abaqus analyses were done on a 0.1 m seeded mesh where each layer is meshed independently. This totaled 1 481 436 degrees of freedom prior to boundary conditions being applied.

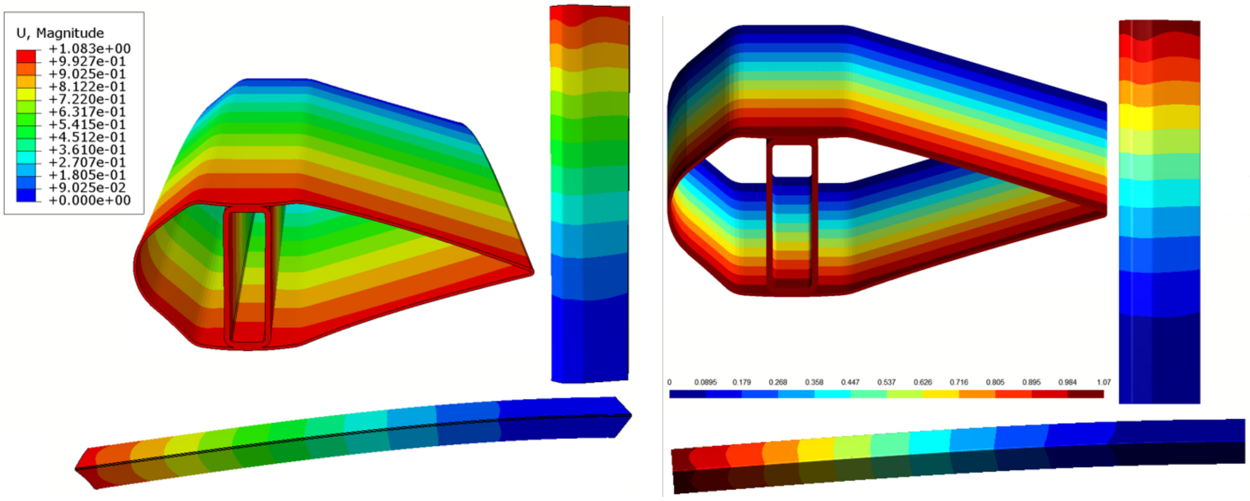


	1	2	3	4	5	6	7	8	9	10
1	2178	3333	4488	5643	6798	7953	9108	1.026e+04	1.142e+04	1.257e+04
2	6402	7557	8712	9867	1.102e+04	1.218e+04	1.333e+04	1.449e+04	1.564e+04	1.68e+04
3	1.063e+04	1.178e+04	1.294e+04	1.409e+04	1.525e+04	1.64e+04	1.756e+04	1.871e+04	1.987e+04	2.102e+04
4	1.689e+04	1.804e+04	1.92e+04	2.256e+04	2.592e+04	2.928e+04	3.264e+04	3.6e+04	3.936e+04	4.272e+04
5	2.519e+04	2.635e+04	2.75e+04	3.307e+04	3.643e+04	4.199e+04	4.756e+04	5.312e+04	5.869e+04	6.425e+04
6	3.554e+04	3.669e+04	3.785e+04	4.562e+04	5.118e+04	5.559e+04	6.441e+04	7.323e+04	8.205e+04	9.087e+04
7	4.792e+04	4.908e+04	5.023e+04	6.021e+04	6.798e+04	7.669e+04	8.11e+04	9.528e+04	1.105e+05	1.268e+05
8	6.235e+04	6.35e+04	6.466e+04	7.684e+04	8.681e+04	1.009e+05	1.096e+05	1.14e+05	1.346e+05	1.573e+05
9	7.881e+04	7.996e+04	8.112e+04	9.550e+04	1.077e+05	1.282e+05	1.422e+05	1.509e+05	1.554e+05	1.834e+05
10	9.731e+04	9.847e+04	9.962e+04	1.162e+05	1.306e+05	1.585e+05	1.79e+05	1.931e+05	2.018e+05	2.062e+05

Fig. 4 Number of degrees of freedom for each structure prior to boundary conditions being applied

Table 4 Material properties

Material	E_1	E_2	E_3	ν_{12}	ν_{13}	ν_{23}	G_{12}	G_{13}	G_{23}
Glass UD	43.22 GPa	12.639 GPa	12.639 GPa	0.29	0.29	0.29	4.42 GPa	4.42 GPa	4.42 GPa
Foam	129 MPa	129 MPa	129 MPa	0.32	0.32	0.32	48.9 MPa	48.9 MPa	48.9 MPa
Glass biax	13.4 GPa	13.4 GPa	13.4 GPa	0.495	0.495	0.495	4.48 GPa	4.48 GPa	4.48 GPa



(a) Displacements computed using Abaqus

(b) Displacements computed using higher order elements

Fig. 5 Displacement field results of a converged linear solution computed using Abaqus or higher order elements

A. Linear analysis

The maximum tip displacements of various expansion orders were computed using a linear static analysis. A comparison in end values between these displacements and a refined Abaqus linear solution is shown in Fig. 6. Convergence is achieved on different combinations of kinematic expansions.

Comparing Figs. 4 and 6, it is noted that the number of degrees of freedom is similar, 67 977 for p7-t5, 64 413 for p6-t7, and 58 689 for p5-t9, where p indicates the in-plane expansion and t the through-thickness expansion. Therefore, the converged solution with the fewest degrees of freedom was achieved in an expansion where the through-thickness kinematic expansion is higher than the in-plane expansion. The convergence table indicates that it is necessary to have a minimum through-thickness expansion of 5 to achieve convergence, signifying that a significant amount of through-thickness deformation occurs, which can not be captured in a shell or solid models with a single linear or quadratic element through the thickness.

The maximum displacement found in the linear solution convergences to a value that is 1.2% different from the Abaqus solution; this difference is due to the lower fidelity geometric representation. Curved sections of the blade are represented using quadratic interpolation. As elements are much larger, the prescribed geometry is different from that of the Abaqus mesh, which has more elements and therefore is more accurate in its geometric description. Displacement fields of both analyses can be found in Fig. 5, showing good agreement.

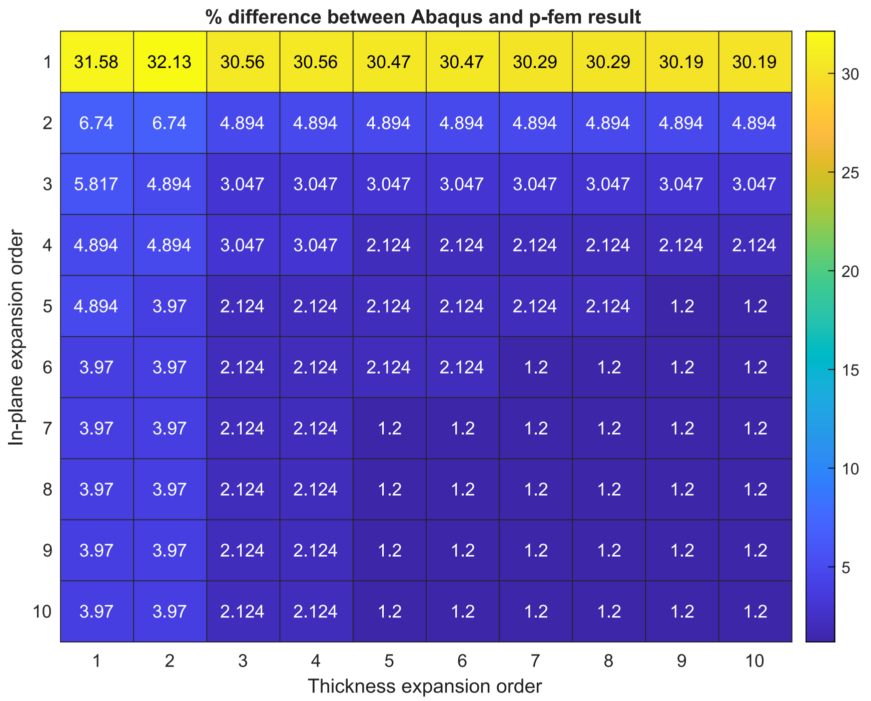
B. Nonlinear analysis

Just as for the linear analysis, the nonlinear analysis is done over various kinematic expansions. Solutions were obtained using a Newton solver with force slowly being applied over multiple load steps.

An overview of maximum displacements found using both approaches is shown in Fig. 7. The results show that due to the more complex deformations and interactions between structural components, convergence in the geometrically nonlinear analysis requires higher kinematic expansion than for the linear analysis.

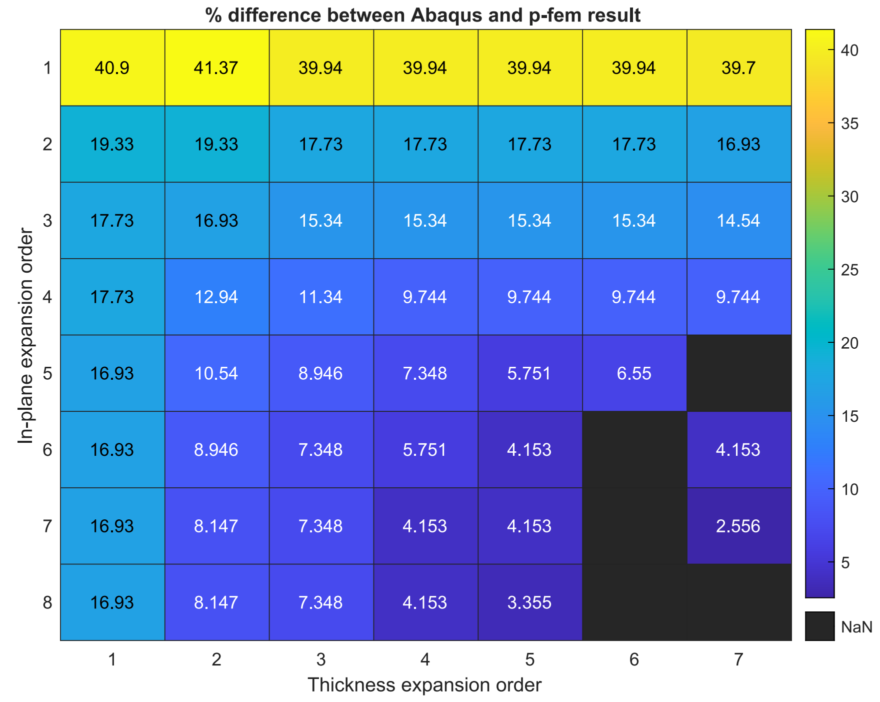
Results converged for only some kinematic expansions due to the analysis going into the post-buckling regime, where snap-through behavior may only sometimes converge using a Newton approach. Resolving this involves using a path-following scheme, implementing numerical damping techniques, or altering the solver parameters.

Deformed shapes of both nonlinear solutions are shown in Fig. 8. Both the Abaqus and higher-order solution show local buckling occurring on the skin of the blade. The exact shape and location of the buckling are highly imperfection sensitive and therefore vary slightly with the kinematic representation.



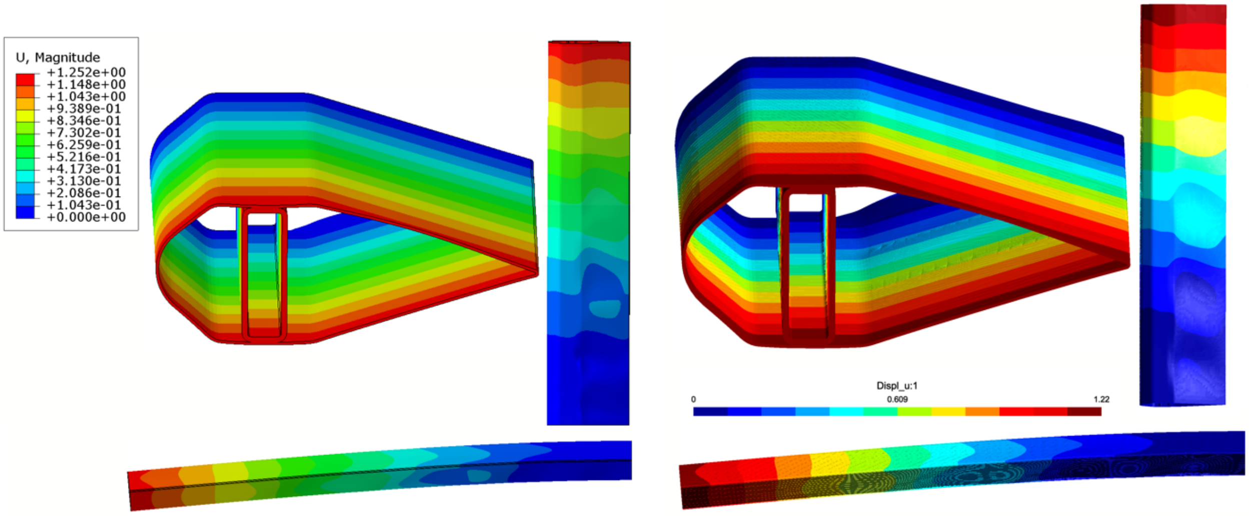
	1	2	3	4	5	6	7	8	9	10
1	31.58	32.13	30.56	30.56	30.47	30.47	30.29	30.29	30.19	30.19
2	6.74	6.74	4.894	4.894	4.894	4.894	4.894	4.894	4.894	4.894
3	5.817	4.894	3.047	3.047	3.047	3.047	3.047	3.047	3.047	3.047
4	4.894	4.894	3.047	3.047	2.124	2.124	2.124	2.124	2.124	2.124
5	4.894	3.97	2.124	2.124	2.124	2.124	2.124	2.124	1.2	1.2
6	3.97	3.97	2.124	2.124	2.124	2.124	1.2	1.2	1.2	1.2
7	3.97	3.97	2.124	2.124	1.2	1.2	1.2	1.2	1.2	1.2
8	3.97	3.97	2.124	2.124	1.2	1.2	1.2	1.2	1.2	1.2
9	3.97	3.97	2.124	2.124	1.2	1.2	1.2	1.2	1.2	1.2
10	3.97	3.97	2.124	2.124	1.2	1.2	1.2	1.2	1.2	1.2

Fig. 6 A comparison in computed differences in end displacements of the structure in a linear analysis



	1	2	3	4	5	6	7
1	40.9	41.37	39.94	39.94	39.94	39.94	39.7
2	19.33	19.33	17.73	17.73	17.73	17.73	16.93
3	17.73	16.93	15.34	15.34	15.34	15.34	14.54
4	17.73	12.94	11.34	9.744	9.744	9.744	9.744
5	16.93	10.54	8.946	7.348	5.751	6.55	
6	16.93	8.946	7.348	5.751	4.153		4.153
7	16.93	8.147	7.348	4.153	4.153		2.556
8	16.93	8.147	7.348	4.153	3.355		

Fig. 7 A comparison in computed differences in end displacements of the structure in a geometrically nonlinear analysis



(a) Displacements computed using Abaqus **(b) Displacements computed using higher order elements**

Fig. 8 Displacement field results of a converged geometrically nonlinear solution computed using Abaqus or higher order elements

V. Conclusions and discussion

Higher-order finite element approaches can lead to converging mechanical responses without needing further remeshing. Furthermore, convergence can be achieved with significantly fewer degrees of freedom than would be necessary using lower-order elements. Good agreements with commercial finite elements were found in both linear and nonlinear analyses to recreate localized buckling in the skin of the blade.

Results presented so far are part of ongoing work to implement higher-order finite elements into an aeroelastic optimization framework. Higher-order models would then be used for design verification and checking structural stability and response. Using existing beam and shell models would lead to inaccurate results, necessitating an approach using solid elements.

The geometric representation of the blade has been shown to be highly critical in mechanical response. Lower-order approximations can have a significant effect when the geometry is not accurately modeled. To this end, improvements can be made by using higher-order or exact geometry, such as NURBS curves in the numerical integration scheme.

In the short term, further refinements will be developed and implemented to improve various aspects of the implementation. These include

Numerical integration techniques Currently, the number of integration points in layered higher-order elements is likely too high. Finding an optimal integration scheme that balances the number of points and accuracy will lead to significant performance improvements.

Stress recovery Certain stress components can not be accurately recovered in layered elements without a stress-recovery technique. Future work will include developing and implementing an efficient and accurate stress recovery scheme for composite materials.

Geometric representation At the moment, meshes are imported from Abaqus, which uses, at most, a quadratic discretization of the geometry. Errors introduced when larger elements are used can be eliminated or reduced by using higher order or exact geometry in the numerical integration. Such approaches can be implemented in the in-house blade meshing tool currently under development.

Appendix

First 15 Lobatto shape functions:

$$\phi_2 = \sqrt{6}\,\left(\frac{x^2}{4} - \frac{1}{4}\right)$$

$$\phi_3 = \frac{\sqrt{10}\,x\,\left(x^2 - 1\right)}{4}$$

$$\phi_4 = \frac{\sqrt{14}\,\left(5\,x^4 - 6\,x^2 + 1\right)}{16}$$

$$\phi_5 = \frac{3\,\sqrt{2}\,x\,\left(7\,x^4 - 10\,x^2 + 3\right)}{16}$$

$$\phi_6 = \frac{\sqrt{22}\,\left(21\,x^6 - 35\,x^4 + 15\,x^2 - 1\right)}{32}$$

$$\phi_7 = \frac{\sqrt{26}\,x\,\left(33\,x^6 - 63\,x^4 + 35\,x^2 - 5\right)}{32}$$

$$\phi_8 = \frac{\sqrt{30}\,\left(429\,x^8 - 924\,x^6 + 630\,x^4 - 140\,x^2 + 5\right)}{256}$$

$$\phi_9 = \frac{\sqrt{34}\,x\,\left(715\,x^8 - 1716\,x^6 + 1386\,x^4 - 420\,x^2 + 35\right)}{256}$$

$$\phi_{10} = \frac{\sqrt{38}\,\left(2431\,x^{10} - 6435\,x^8 + 6006\,x^6 - 2310\,x^4 + 315\,x^2 - 7\right)}{512}$$

$$\phi_{11} = \frac{\sqrt{42}\,x\,\left(4199\,x^{10} - 12155\,x^8 + 12870\,x^6 - 6006\,x^4 + 1155\,x^2 - 63\right)}{512}$$

$$\phi_{12} = \frac{\sqrt{46}\,\left(29393\,x^{12} - 92378\,x^{10} + 109395\,x^8 - 60060\,x^6 + 15015\,x^4 - 1386\,x^2 + 21\right)}{2048}$$

$$\phi_{13} = \frac{5\,\sqrt{2}\,x\,\left(52003\,x^{12} - 176358\,x^{10} + 230945\,x^8 - 145860\,x^6 + 45045\,x^4 - 6006\,x^2 + 231\right)}{2048}$$

$$\phi_{14} = \frac{3\,\sqrt{6}\,\left(185725\,x^{14} - 676039\,x^{12} + 969969\,x^{10} - 692835\,x^8 + 255255\,x^6 - 45045\,x^4 + 3003\,x^2 - 33\right)}{4096}$$

$$\phi_{15} = \frac{\sqrt{58}\,x\,\left(334305\,x^{14} - 1300075\,x^{12} + 2028117\,x^{10} - 1616615\,x^8 + 692835\,x^6 - 153153\,x^4 + 15015\,x^2 - 429\right)}{4096}$$

Derivatives of first 15 Lobatto shape functions

$$\frac{\mathrm{d}\phi_2}{\mathrm{x}} = \frac{\sqrt{6}\,x}{2}\mathrm{d}x$$
$$\frac{\mathrm{d}\phi_3}{\mathrm{x}} = \frac{\sqrt{10}\,\left(3\,x^2-1\right)}{4}\mathrm{d}x$$
$$\frac{\mathrm{d}\phi_4}{\mathrm{x}} = \frac{\sqrt{14}\,x\,\left(5\,x^2-3\right)}{4}\mathrm{d}x$$
$$\frac{\mathrm{d}\phi_5}{\mathrm{x}} = \frac{3\,\sqrt{2}\,\left(35\,x^4-30\,x^2+3\right)}{16}\mathrm{d}x$$
$$\frac{\mathrm{d}\phi_6}{\mathrm{x}} = \frac{\sqrt{22}\,x\,\left(63\,x^4-70\,x^2+15\right)}{16}\mathrm{d}x$$
$$\frac{\mathrm{d}\phi_7}{\mathrm{x}} = \frac{\sqrt{26}\,\left(231\,x^6-315\,x^4+105\,x^2-5\right)}{32}\mathrm{d}x$$
$$\frac{\mathrm{d}\phi_8}{\mathrm{x}} = \frac{\sqrt{30}\,x\,\left(429\,x^6-693\,x^4+315\,x^2-35\right)}{32}\mathrm{d}x$$
$$\frac{\mathrm{d}\phi_9}{\mathrm{x}} = \frac{\sqrt{34}\,\left(6435\,x^8-12012\,x^6+6930\,x^4-1260\,x^2+35\right)}{256}\mathrm{d}x$$
$$\frac{\mathrm{d}\phi_{10}}{\mathrm{x}} = \frac{\sqrt{38}\,x\,\left(12155\,x^8-25740\,x^6+18018\,x^4-4620\,x^2+315\right)}{256}\mathrm{d}x$$
$$\frac{\mathrm{d}\phi_{11}}{\mathrm{x}} = \frac{\sqrt{42}\,\left(46189\,x^{10}-109395\,x^8+90090\,x^6-30030\,x^4+3465\,x^2-63\right)}{512}\mathrm{d}x$$
$$\frac{\mathrm{d}\phi_{12}}{\mathrm{x}} = \frac{\sqrt{46}\,x\,\left(88179\,x^{10}-230945\,x^8+218790\,x^6-90090\,x^4+15015\,x^2-693\right)}{512}\mathrm{d}x$$
$$\frac{\mathrm{d}\phi_{13}}{\mathrm{x}} = \frac{5\,\sqrt{2}\,\left(676039\,x^{12}-1939938\,x^{10}+2078505\,x^8-1021020\,x^6+225225\,x^4-18018\,x^2+231\right)}{2048}\mathrm{d}x$$
$$\frac{\mathrm{d}\phi_{14}}{\mathrm{x}} = \frac{3\,\sqrt{6}\,x\,\left(1300075\,x^{12}-4056234\,x^{10}+4849845\,x^8-2771340\,x^6+765765\,x^4-90090\,x^2+3003\right)}{2048}\mathrm{d}x$$
$$\frac{\mathrm{d}\phi_{15}}{\mathrm{x}} = \frac{\sqrt{58}\,\left(5014575\,x^{14}-16900975\,x^{12}+22309287\,x^{10}-14549535\,x^8+4849845\,x^6-765765\,x^4+45045\,x^2-429\right)}{4096}\mathrm{d}x$$

Acknowledgments

The authors would like to thank the Wind Blades Research Hub (WBRH), a collaboration between the University of Bristol and ORE Catapult, for their support of this work.

References

[1] Jensen, F. M., Falzon, B. G., Ankersen, J., and Stang, H., "Structural testing and numerical simulation of a 34 m composite wind turbine blade," *Composite Structures*, Vol. 76, No. 1-2, 2006, pp. 52–61. https://doi.org/10.1016/j.compstruct.2006.06.008.

[2] Faccio Júnior, C. J., Cardozo, A. C. P., Monteiro Júnior, V., and Gay Neto, A., "Modeling wind turbine blades by geometrically-exact beam and shell elements: A comparative approach," *Engineering Structures*, Vol. 180, No. March 2018, 2019, pp. 357–378. https://doi.org/10.1016/j.engstruct.2018.09.032, URL https://doi.org/10.1016/j.engstruct.2018.09.032.

[3] Maes, V., "On the sensitivity and validation of bending, twisting, and bend-twist coupling behaviour of wind turbine blade cross-sections," Ph.D. thesis, University of Bristol, Bristol, United Kingdom, 2020. URL http://research-information.bristol.ac.uk.

[4] Peeters, M., Santo, G., Degroote, J., and Van Paepegem, W., "High-fidelity finite element models of composite wind turbine blades with shell and solid elements," *Composite Structures*, Vol. 200, No. February, 2018, pp. 521–531. https://doi.org/10.1016/j.compstruct.2018.05.091, URL https://doi.org/10.1016/j.compstruct.2018.05.091.

[5] Tavares, R. P., Bouwman, V., and Van Paepegem, W., "Finite element analysis of wind turbine blades subjected to torsional loads: Shell vs solid elements," *Composite Structures*, Vol. 280, No. October 2021, 2022, p. 114905. https://doi.org/10.1016/j.compstruct.2021.114905, URL https://doi.org/10.1016/j.compstruct.2021.114905.

[6] Szabó, B., Düster, A., and Rank, E., "The p -version of the Finite Element Method," *Encyclopedia of Computational Mechanics*, 3, John Wiley & Sons, Ltd, Chichester, UK, 2004, pp. 119–139. https://doi.org/10.1002/0470091355.ecm003g, URL https://onlinelibrary.wiley.com/doi/10.1002/0470091355.ecm003g.

[7] Patni, M., Minera, S., Groh, R. M., Pirrera, A., and Weaver, P. M., "Efficient 3D stress capture of variable-stiffness and sandwich beam structures," *AIAA Journal*, Vol. 57, No. 9, 2019, pp. 4042–4056. https://doi.org/10.2514/1.J058220.

[8] Vanskike, W. P., "Comparative Assessment of Finite Element Modeling Methods for Wind Turbine Blades," Ph.D. thesis, University of Kansas, 2020. URL http://hdl.handle.net/1808/32635.

[9] Szabó, B., and Babuška, I., *Finite Element Analysis Method, Verification, and Validation*, 2$^{\text{nd}}$ ed., John Wiley & Sons, Hoboken, NJ, USA, 2021.

[10] Düster, A., Bröker, H., and Rank, E., "The p-version of the finite element method for three-dimensional curved thin walled structures," *International Journal for Numerical Methods in Engineering*, Vol. 52, No. 7, 2001, pp. 673–703. https://doi.org/10.1002/nme.222, URL https://onlinelibrary.wiley.com/doi/10.1002/nme.222.

[11] Šolín, P., Segeth, K., and Doležel, I., *Higher-order finite element methods*, 2003. https://doi.org/10.1201/9780203488041.