Accepted for publication in *Thin-Walled Structures*.
http://dx.doi.org/10.1016/j.tws.2018.05.019

A 'boundary layer' finite element for thin multi-strake conical shells

Adrien Boyez[1], Adam J. Sadowski[2] & Bassam A. Izzuddin[3]

Abstract

Multi-strake cylindrical and conical shells of revolution are complex but commonplace industrial structures which are composed of multiple segments of varying wall thickness. They find application as tanks, silos, circular hollow sections, aerospace structures and wind turbine support towers, amongst others. The modelling of such structures with classical finite elements interpolated using low order polynomial shape functions presents a particular challenge, because many elements must be sacrificed solely in order to accurately represent the regions of local compatibility bending, so-called 'boundary layers', near shell boundaries, changes of wall thickness and at other discontinuities. Partitioning schemes must be applied to localise mesh refinement within the boundary layers and avoid excessive model runtimes, a particular concern in incremental nonlinear analyses of large models where matrix systems are handled repeatedly.

In a previous paper, the authors introduced a novel axisymmetric cylindrical shell finite element that was enriched with transcendental shape functions to capture the bending boundary layer exactly, permitting significant economies in the element and degrees of freedom count, mesh design and model generation effort. One element is sufficient per wall strake. This paper extends this work to conical geometries, where axisymmetric elements enriched with Bessel functions accurately capture the bending boundary layer for both 'shallow' and 'steep' conical strakes, which are characterised by interacting and independent boundary layers, respectively. The bending shape functions are integrated numerically, with several integration schemes investigated for accuracy and efficiency. The potential of the element is illustrated through a stress analysis of a real 22-strake metal wind turbine support tower under self-weight. The work is part of a wider project to design a general three-dimensional 'boundary layer' element.

Keywords

Conical shell; thin axisymmetric shell; bending boundary layer; Bessel functions; finite element method.

[1]PhD Student, Department of Civil & Environmental Engineering, Imperial College London
[2]Lecturer, Department of Civil & Environmental Engineering, Imperial College London
[3]Professor, Department of Civil & Environmental Engineering, Imperial College London

1. Introduction

Cylindrical shells find widespread application as containment structures, supporting structures and aerospace vehicles. Their ubiquity is a result of the relative ease of construction of cylindrical geometries and of the relative simplicity of their manual dimensioning, typically performed using shell membrane theory. This determinate theory is based on balancing external loads with internal membrane stress resultants only, disregarding the high local bending stresses that may arise in response to kinematic compatibility requirements at a boundary or change of wall thickness. These stresses decay away from the discontinuity at an exponential rate, forming a 'boundary layer' whose length can be taken as two bending half-wavelengths λ [1]. For a thin cylinder, λ is usually small relative to the length of the strake, and the membrane theory solution is therefore valid over the majority of the cylinder. Where this is not the case, a manual application of axisymmetric shell bending theory is just about practical for uniform thickness cylinders [1, 2, 3]. The membrane theory treatment of cones is straightforward due to their straight meridian, however their classical bending theory to cones is made quite challenging by the necessity for the analyst to manipulate Bessel functions [1, 4, 5, 6]. Cui *et al.* derived an analytical theory that circumvents the use of Bessel functions while delivering a better accuracy than the equivalent cylinder method [7], but numerical methods tend to be preferred even for stress analyses, although they require a careful mesh design to capture the boundary layer effect.

The authors' previous 'proof of concept' study [8] adopted the novel approach of distinguishing between the 'membrane' and 'bending' components of the shell's kinematic degrees of freedom (DOFs) and interpolating these separately to create a linear axisymmetric 'Cylindrical Shell Boundary Layer' (CSBL) element. The membrane displacements were interpolated with simple polynomial functions, but bending displacements were interpolated with transcendental functions derived from the governing differential equation, enriching the element's interpolation field to support the boundary layer natively. An illustration on a number of realistic multi-strake civil engineering shell structures showed that the CSBL offered significant advantages in terms of reduced elements and DOFs counts, mesh design and accuracy over a 'classical' shell element with polynomial shape functions based on Zienkiewicz *et al.* [9].

The same approach will be followed in this paper to derive a conical version of this element, here termed 'CoSBL'. The authors first present a brief derivation of axisymmetric bending theory for conical shells in order to establish the strong form differential equation (following Flügge [4]), as its solution will provide the functional form for the interpolation field of the bending component of the total displacements (the membrane component will be interpolated with simple functions, as for the CSBL). Various integration schemes for the CoSBL stiffness matrix are explored (with results presented in the Appendix for compactness), and two dimensionless parameters are identified to characterise the relationship between the two boundary layers of a CoSBL element. Finally, the potential of the element is illustrated on a complex and realistic 22-strake civil engineering structure.

Readers are invited to consult Chapelle and Bathe [10] for a detailed review of the widespread literature on classical shell finite elements. The authors are aware only of the work of Bhatia and Sekhon that is of direct relevance to this paper, who successfully developed ‘macro’ cylindrical, conical and spherical linear axisymmetric shell elements [11, 12, 13] using a method described in [14]. It does not rely on the definition of bending shape functions, using instead the integration constants of the solution to the governing differential equation as implicit DOFs. The solutions presented accommodate constant distributed loads, although the method supports extension to arbitrary load distributions. Single-strake problems are used for illustration, but the physical significance of the solution and its governing parameters are not discussed in detail.

2. Axisymmetric bending theory for thin isotropic conical shells

The present derivation of the bending theory for isotropic conical shells is adapted from Flügge [4], specialised for axisymmetric cones of constant thickness with all assumptions stated before any equation manipulation. The first step in the derivation, first introduced by Reissner [15], is to solve for the shear force and shell midsurface rotation rather than the radial or meridional displacements. The second step is the identification of the Meissner differential operator [16] allowing for the decoupling of the resulting equations. The last step involves a change of variable from the slant height to a dimensionless parameter to reveal Bessel’s differential equation. The physical significance of this parameter and the boundary-layer behaviour of the bending solution is discussed in a later part of the paper.

2.1. Equilibrium, kinematics and constitutive relations

A conical shell of apex half-angle $\pi/2 - \alpha$ (where $0 < \alpha < \pi/2$) and thickness t may be subject to distributed loads p_n and p_s that are respectively normal and tangential to the midsurface (Fig. 1). Assuming axisymmetry of the loading, boundary conditions and geometry, five stress resultants act on the mid-surface: the meridional and circumferential membrane stress resultants n_s and n_θ, the bending moment stress resultants m_s and m_θ, and the meridional transverse shear stress resultant q_s. No displacements, shears or gradients arise in the circumferential θ direction.

It is assumed that the conical shell is a frustum bounded by its slant height coordinates s_1 and s_2 ($s_1 < s_2$), leading to the following radial and vertical coordinates:

$$
\begin{aligned}
r &= s \cdot \cos(\alpha) \\
z &= s \cdot \sin(\alpha) \cdot \operatorname{sgn}(\cos(\beta)) + z_0
\end{aligned}
\tag{1}
$$

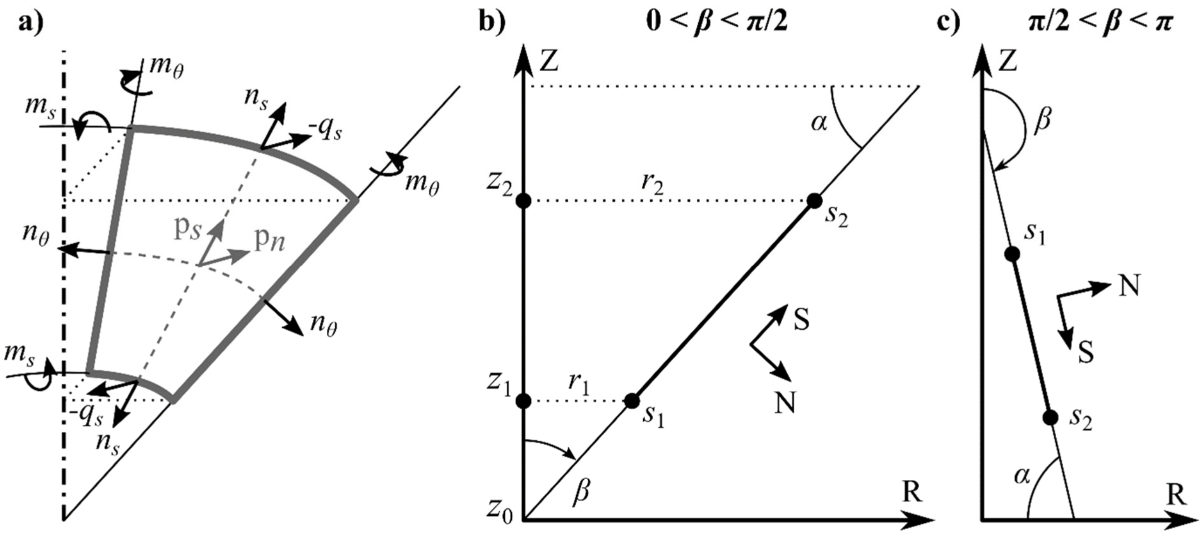

Fig. 1 – a) Shell stress resultants diagram, b) corresponding geometry of a conical shell section and c) alternative cone orientation.

Equilibrium considerations yield the following system of equations, where the superscript • denotes differentiation with respect to the slant height s:

$$
\begin{aligned}
\left(s \cdot n_s\right)^{\bullet} - n_\theta &= -s \cdot p_s \\
\left(s \cdot q_s\right)^{\bullet} + \tan(\alpha) \cdot n_\theta &= s \cdot p_n \\
\left(s \cdot m_s\right)^{\bullet} - m_\theta - s \cdot q_s &= 0
\end{aligned}
\tag{2}
$$

The following classical linear-elastic constitutive and thin-shell kinematics relationships for a conical shell are adopted (where w and u are the normal and meridional midsurface displacements respectively, while χ is the midsurface rotation about the circumferential axis):

$$
\begin{gathered}
\begin{bmatrix} n_s \\ n_\theta \end{bmatrix} = C_m \begin{bmatrix} 1 & \nu \\ \nu & 1 \end{bmatrix} \begin{bmatrix} \varepsilon_s \\ \varepsilon_\theta \end{bmatrix} \text{ with } C_m = \frac{Et}{\left(1-\nu^2\right)} \\
\text{and } \begin{bmatrix} m_s \\ m_\theta \end{bmatrix} = C_b \begin{bmatrix} 1 & \nu \\ \nu & 1 \end{bmatrix} \begin{bmatrix} \kappa_s \\ \kappa_\theta \end{bmatrix} \text{ with } C_b = \frac{Et^3}{12\left(1-\nu^2\right)}
\end{gathered}
\tag{3}
$$

$$
\begin{gathered}
\begin{bmatrix} \varepsilon_z \\ \varepsilon_\theta \end{bmatrix} = \begin{bmatrix} u^{\bullet} & \frac{1}{s}\left(u + \frac{w}{c}\right) \end{bmatrix}^T \\
\text{and } \begin{bmatrix} \kappa_z \\ \kappa_\theta \end{bmatrix} = \begin{bmatrix} \chi^{\bullet} & \frac{\chi}{s} \end{bmatrix}^T \text{ with } c = \cot(\alpha) \text{ and } \chi = w^{\bullet}
\end{gathered}
\tag{4}
$$

2.2. Uncoupled differential equation

The key to identifying the conical shell bending differential equation is to solve for the variables $s{\cdot}q_s$ and χ. This requires recasting the membrane kinematic relations as the following:

$$\chi = c\left(\left(s\varepsilon_\theta\right)^{\bullet} - \varepsilon_s\right) \tag{5}$$

From this and the equilibrium equations (Eq. (2)), the following two differential equations are obtained, where the Meissner differential operator Λ can now be identified:

$$\begin{cases} \dfrac{\Lambda(\chi)}{c} = \dfrac{s \cdot q_s}{C_b} \\ \Lambda\left(s \cdot q_s\right) + C_m\left(1-\nu^2\right)\dfrac{\chi}{c} = g \end{cases} \text{ where } \begin{cases} \Lambda\left(f\right) = c\left[s \cdot f^{\bullet\bullet} + f^{\bullet} - \dfrac{1}{s}f\right] \\ g = -\dfrac{1}{s}\displaystyle\int s\left(c \cdot p_n + p_s\right) + c\left(s^2 p_n\right)^{\bullet} - \nu \cdot s \cdot p_s \end{cases} \tag{6}$$

A further application of Λ on the second differential equation achieves the decoupling:

$$\Lambda\left[\Lambda\left(s \cdot q_s\right)\right] + \mu^4 s \cdot q_s = \Lambda\left(g\right) \text{ where } \mu^4 = \frac{12\left(1-\nu^2\right)}{t^2} \tag{7}$$

Solutions to this fourth-order real differential equation are the superposition of a particular solution responsible for balancing the loads, referred to as the 'membrane' solution and a linear combination of four functions solution to the homogeneous equation (i.e. for $p_n = p_s = g = 0$), referred to as the 'bending' solution that accommodates boundary conditions. The total value of any quantity is obtained by superposition, e.g. $w = w^b + w^m$ and $n_s = n_s^b + n_s^m$.

Once a solution for $s{\cdot}q_s$ is obtained, the associated stress, strain and displacement fields can be deduced. The second equation from Eq. (6) is used to obtain χ, while the second equilibrium equation in Eq. (2) yields n_θ which, in combination with the first, yields n_s. The bending kinematic relations (Eq. (4)) lead to curvatures which, when combined with the bending constitutive relations (Eq. (3)), are used to obtain m_s and m_θ. The inverse of the membrane constitutive relations (Eq. (3)) can be used to obtain membrane strains from membrane stresses, from which u and w are then finally deduced.

2.3. 'Bending' homogeneous solution

The fourth-order real differential equation can be reduced to the following two second-order complex differential equations:

$$\Lambda\left(s \cdot q_s^{\,b}\right) \pm i\mu^2 s \cdot q_s^{\,b} = 0 \tag{8}$$

It is enough to solve one of these two equations as they are complex conjugates of one another. The real and imaginary parts of its two solutions will offer four independent solutions to Eq. (7). The equation to be solved is thus:

$$\left(s\cdot q_s^{\,b}\right)^{\bullet\bullet}+\frac{1}{s}\left(s\cdot q_s^{\,b}\right)^{\bullet}+\left(-\frac{1}{s^2}+\frac{i\mu^2}{s\cdot c}\right)sq_s^{\,b}=0 \tag{9}$$

This equation may be reduced to a Bessel differential equation of order two by introducing a change of variable:

$$\frac{\mathrm{d}^2\left(s\cdot q_s^{\,b}\right)}{\mathrm{d}\eta^2}+\frac{1}{\eta}\frac{\mathrm{d}\left(s\cdot q_s^{\,b}\right)}{\mathrm{d}\eta}+\left(1-\frac{2^2}{\eta^2}\right)s\cdot q_s^{\,b}=0\ \text{ where }\ \eta=2\mu e^{i\pi/4}\sqrt{s/c} \tag{10}$$

The real and imaginary parts of the solutions to Eq. (10) can be expressed in terms of Kelvin functions of the transformed variable y, so that:

$$\begin{gathered}s\cdot q_s^{\,b}(s)=A_1\,\mathrm{Ber}_2(y)+A_2\,\mathrm{Bei}(y)+A_3\,\mathrm{Ker}_2(y)+A_4\,\mathrm{Kei}(y)\\ \text{with } y=2\left[3\left(1-\nu^2\right)\right]^{1/4}\sqrt{\frac{2s}{c\cdot t}}\end{gathered} \tag{11}$$

The bending components of the displacements w^b and u^b, needed to identify the bending shape functions, may now be deduced as:

$$\begin{aligned}-w^b(s)&=A_1\left(\mathrm{Ber}_0-\frac{y\sqrt{2}}{4}\left(\mathrm{Bei}_1+\mathrm{Ber}_1\right)\right)+A_2\left(\mathrm{Bei}_0-\frac{y\sqrt{2}}{4}\left(\mathrm{Bei}_1-\mathrm{Ber}_1\right)\right)\\&\quad+A_3\left(\mathrm{Ker}_0-\frac{y\sqrt{2}}{4}\left(\mathrm{Kei}_1+\mathrm{Ker}_1\right)\right)+A_4\left(\mathrm{Kei}_0-\frac{y\sqrt{2}}{4}\left(\mathrm{Kei}_1-\mathrm{Ker}_1\right)\right)\\-c\cdot u^b(s)&=A_1\left(\nu\,\mathrm{Ber}_0-\frac{\sqrt{2}\left(1+\nu\right)}{y}\left(\mathrm{Bei}_1-\mathrm{Ber}_1\right)\right)+A_2\left(\nu\,\mathrm{Bei}_0+\frac{\sqrt{2}\left(1+\nu\right)}{y}\left(\mathrm{Bei}_1+\mathrm{Ber}_1\right)\right)\\&\quad+A_3\left(\nu\,\mathrm{Ker}_0-\frac{\sqrt{2}\left(1+\nu\right)}{y}\left(\mathrm{Kei}_1-\mathrm{Ker}_1\right)\right)+A_4\left(\nu\,\mathrm{Kei}_0+\frac{\sqrt{2}\left(1+\nu\right)}{y}\left(\mathrm{Kei}_1+\mathrm{Ker}_1\right)\right)\end{aligned} \tag{12}$$

In the above, the argument y of the Kelvin functions has been omitted for compactness.

2.4. 'Membrane' particular solution

As the distributed loads p_n and p_s are arbitrary, it is impossible to propose a 'general' particular solution. It is however possible to identify the functional form of the particular solution under polynomial distributed loads as a starting point for a finite element implementation: provided the finite element has shape functions that include this functional form, such loads can then be solved for exactly. For distributed loads defined by polynomials of degree N, the process to identify the functional form of the corresponding membrane displacement fields is given here:

$$
\begin{aligned}
&\text{Assumed dist. load:} \quad && p_n = \sum_{j=0}^{\mathrm{N}} p_{n,j} \cdot s^j && p_s = \sum_{j=0}^{\mathrm{N}} p_{s,j} \cdot s^j \\
&\text{Eqs (7) then (6):} \quad && q_s{}^m = \sum_{j=0}^{\mathrm{N-1}} q_{s,j} \cdot s^j && \chi^m = \frac{\chi_{s,-1}}{s} + \sum_{j=1}^{\mathrm{N+1}} \chi_{s,j} \cdot s^j \\
&\text{Eq. (2):} \quad && n_\theta{}^m = \sum_{j=0}^{\mathrm{N+1}} n_{\theta,j} \cdot s^j && n_s{}^m = \frac{n_{s,-1}}{s} + \sum_{j=0}^{\mathrm{N+1}} n_{s,j} \cdot s^j \\
&\text{Eq. (3):} \quad && \varepsilon_s{}^m = \frac{\varepsilon_{s,-1}}{s} + \sum_{j=0}^{\mathrm{N+1}} \varepsilon_{s,j} \cdot s^j && \varepsilon_\theta{}^m = \frac{\varepsilon_{\theta,-1}}{s} + \sum_{j=1}^{\mathrm{N+1}} \varepsilon_{\theta,j} \cdot s^j \\
&\text{Eq. (4):} \quad && u^m = u_{\ln} \ln(s) + \sum_{j=0}^{\mathrm{N+2}} u_j \cdot s^j && w^m = w_{\ln} \ln(s) + \sum_{\substack{j=0 \\ j\neq 1}}^{\mathrm{N+2}} w_j \cdot s^j
\end{aligned}
\tag{13}
$$

It can be noted that both membrane displacement fields are polynomials of order N+2 in s, with w^m lacking a linear term in s. Both contain a logarithmic term, responsible for balancing vertical edge loads, whose respective coefficients satisfy the following equation:

$$w_{\ln} = -c \cdot u_{\ln} \tag{14}$$

3. Axisymmetric conical shell boundary layer (CoSBL) element kinematics

The present approach aims to directly translate the mathematical and physical properties of conical shell bending theory in the implementation of a finite element, as was done for its cylindrical counterpart [8]. The distinction between the particular and homogeneous solutions is thus reflected by the introduction of two independent sets of shape functions and DOFs: the ‘membrane’ components responsible for balancing the applied loads, and the ‘bending’ components responsible for accommodating boundary conditions.

3.1. Membrane shape functions and degrees of freedom

It is proposed, as a compromise between generality and complexity, to implement a finite element able to exactly accommodate distributed loads p_n and p_s that are polynomials of up to second order. However, this formulation can easily be extended to accommodate loads of higher order by adding more polynomial shape functions, or specialised for more complex loads by adding ad hoc shape functions derived from Eqs (2) to (7).

A set of four shape functions (similar in design to the Hermite cubics) is first derived from $\mathbf{S} = \{1, s^2, s^3, s^4\}$. The displacement field w can be expressed either as a linear combination of the polynomials of $\mathbf{S}$, or as a linear combination of some shape functions $\mathbf{H}$ whose associated

DOFs $\mathbf{d}w^m$ are the values of the membrane component of w and its first derivative χ at both ends of the cone (i.e. at s_1 and s_2), here termed $w_{s1}{}^m$, $\chi_{s1}{}^m$, $w_{s2}{}^m$ and $\chi_{s2}{}^m$ respectively:

$$\begin{array}{ccccc} \begin{Bmatrix} 1 \\ s^2 \\ s^3 \\ s^4 \end{Bmatrix}^{\mathrm{T}} & \begin{Bmatrix} w_0 \\ w_2 \\ w_3 \\ w_4 \end{Bmatrix} & = & \begin{Bmatrix} H_1 \\ H_2 \\ H_3 \\ H_4 \end{Bmatrix}^{\mathrm{T}} & \begin{Bmatrix} w_{s1}^m \\ \chi_{s1}^m \\ w_{s2}^m \\ \chi_{s2}^m \end{Bmatrix} \\ \{\mathbf{S}\}^{\mathrm{T}} & \{\mathbf{w}\} & = & \{\mathbf{H}\}^{\mathrm{T}} & \{\mathbf{d}w^m\} \end{array} \tag{15}$$

By definition, $\mathbf{d}w^m$ can also be expressed in terms of the w_j constants:

$$\{\mathbf{d}w^m\} = \underbrace{\left[\mathbf{S}(s_1) \quad \mathbf{S}^{\bullet}(s_1) \quad \mathbf{S}(s_2) \quad \mathbf{S}^{\bullet}(s_2)\right]^{\mathrm{T}}}_{[\mathbf{M}^m]} \{\mathbf{w}\} \tag{16}$$

Combining Eqs (15) to (16) and solving the system yields four basic membrane shape functions as linear combinations of the functions in **S**:

$$\{\mathbf{H}\}^{\mathrm{T}}[\mathbf{M}^m]\{\mathbf{w}\} = \{\mathbf{S}\}^{\mathrm{T}}\{\mathbf{w}\} \quad \Rightarrow \quad \{\mathbf{H}\} = \left([\mathbf{M}^m]^{-1}\right)^{\mathrm{T}}\{\mathbf{S}\} \tag{17}$$

The last step is valid only if $\mathbf{M}^m$ is invertible, which can be checked through its determinant:

$$\det(\mathbf{M}^m) = 2s_1s_2\left(s_1+s_2\right)\left(s_1-s_2\right)^4 \tag{18}$$

This determinant will always be strictly positive and thus the shape functions will always be well-defined since $s_2 > s_1 > 0$ (elements cannot include the apex).

The logarithmic term present in both w^m and u^m (Eq. (13)), and the linear term in u^{m} also require shape functions. Static condensation will ultimately be used to reduce the number of DOFs to 6, so the value and first derivative of the logarithmic and linear shape functions are brought to 0 at both ends of the element by combining them with the **H** functions as follows:

$$\begin{aligned} \begin{Bmatrix} L \\ P \end{Bmatrix} &= \begin{Bmatrix} \ln(s) \\ s \end{Bmatrix} - \begin{bmatrix} \ln(s_1) & 1/s_1 & \ln(s_2) & 1/s_2 \\ s_1 & 1 & s_2 & 1 \end{bmatrix} \{\mathbf{H}\} \\ &= \{LP\} - \langle \mathbf{V} \mid LP \rangle \{\mathbf{H}\} \end{aligned} \tag{19}$$

Although the logarithmic shape function could be used once for each displacement to interpolate the logarithmic term, Eq. (14) can be exploited so that only one ‘logarithmic’ DOF (corresponding, for instance, to $u_{\ln}$) is required. Considering only the membrane component of displacements, the vector of membrane DOFs $\mathbf{d}^m$ would then be the following:

$$\{\mathbf{d}^m\} = \left\langle w_{s1}^m \quad \chi_{s1}^m \quad w_{s2}^m \quad \chi_{s2}^m \quad u^L \quad u^P \quad u_{s1}^m \quad u_{s1}^{\bullet m} \quad u_{s2}^m \quad u_{s2}^{\bullet m} \right\rangle^{\mathrm{T}} \tag{20}$$

The displacement and strains are obtained through matrix multiplication by the appropriate vectors of shape functions N, whose expressions can be derived using the kinematic relations:

$$
\begin{aligned}
&\left\langle u^m \quad w^m \quad \chi^m \quad \varepsilon_s{}^m \quad \varepsilon_\theta{}^m \quad \kappa_s{}^m \quad \kappa_\theta{}^m \right\rangle \\
&= \left\{\mathbf{d}^m\right\}^{\mathrm{T}}
\begin{bmatrix}
0 & H_1 & H_1^{\bullet} & 0 & H_1/cs & H_1^{\bullet\bullet} & H_1^{\bullet}/s \\
0 & H_2 & H_2^{\bullet} & 0 & H_2/cs & H_2^{\bullet\bullet} & H_2^{\bullet}/s \\
0 & H_3 & H_3^{\bullet} & 0 & H_3/cs & H_3^{\bullet\bullet} & H_3^{\bullet}/s \\
0 & H_4 & H_4^{\bullet} & 0 & H_4/cs & H_4^{\bullet\bullet} & H_4^{\bullet}/s \\
L & -cL & -cL^{\bullet} & L^{\bullet} & 0 & -cL^{\bullet\bullet} & -cL^{\bullet}/s \\
P & 0 & 0 & P^{\bullet} & P/s & 0 & 0 \\
H_1 & 0 & 0 & H_1^{\bullet} & H_1/s & 0 & 0 \\
H_2 & 0 & 0 & H_2^{\bullet} & H_2/s & 0 & 0 \\
H_3 & 0 & 0 & H_3^{\bullet} & H_3/s & 0 & 0 \\
H_4 & 0 & 0 & H_4^{\bullet} & H_4/s & 0 & 0
\end{bmatrix} \\
&= \left\{\mathbf{d}^m\right\}^{\mathrm{T}} \left[\mathbf{N}^m \mid u \quad \mathbf{N}^m \mid w \quad \mathbf{N}^m \mid \chi \quad \mathbf{N}^m \mid \varepsilon_s \quad \mathbf{N}^m \mid \varepsilon_\theta \quad \mathbf{N}^m \mid \kappa_s \quad \mathbf{N}^m \mid \kappa_\theta\right]
\end{aligned}
\tag{21}
$$

3.2. Bending shape functions and degrees of freedom

The bending shape functions of the CoSBL element are derived from the displacement and strain functions associated with the solution to Eq. (10) to allow a native representation of the compatibility bending boundary layer. The bending component of all displacement and strain fields will be obtained from a vector of bending DOFs $\mathbf{d}^b$ in the following manner:

$$
\begin{aligned}
&\left\langle u^b \quad w^b \quad \chi^b \quad \varepsilon_s{}^b \quad \varepsilon_\theta{}^b \quad \kappa_s{}^b \quad \kappa_\theta{}^b \right\rangle \\
&= \left\{\mathbf{d}^b\right\}^{\mathrm{T}} \left[\mathbf{N}^b \mid u \quad \mathbf{N}^b \mid w \quad \mathbf{N}^b \mid \chi \quad \mathbf{N}^b \mid \varepsilon_s \quad \mathbf{N}^b \mid \varepsilon_\theta \quad \mathbf{N}^b \mid \kappa_s \quad \mathbf{N}^b \mid \kappa_\theta\right]
\end{aligned}
\tag{22}
$$

The constituents of $\mathbf{d}^b$ are taken as the bending counterparts of those of $\mathbf{d}w^m$ (Eq (15)):

$$
\left\{\mathbf{d}^b\right\} = \left\langle w_{s1}^b \quad \chi_{s1}^b \quad w_{s2}^b \quad \chi_{s2}^b \right\rangle^{\mathrm{T}} \tag{23}
$$

The process used to derive the polynomial membrane shape functions (Eqs (15) to (17)) is identically applied, starting from the expression of w^b in Eq. (12):

$$
\begin{aligned}
w^b &= \left\{\mathbf{W}^b\right\}^{\mathrm{T}} \left\{\mathbf{A}\right\} = \left\{\mathbf{G}w\right\}^{\mathrm{T}} \left\{\mathbf{d}^b\right\} \\
\left\{\mathbf{d}^b\right\} &= \underbrace{\left[\mathbf{W}^b(s_1) \quad \mathbf{W}^{b\bullet}(s_1) \quad \mathbf{W}^b(s_2) \quad \mathbf{W}^{b\bullet}(s_2)\right]^{\mathrm{T}}}_{\left[\mathbf{M}^b\right]} \left\{\mathbf{A}\right\}
\end{aligned}
\tag{24}
$$

Combining these two equations yields the definition of $\mathbf{g}$, the matrix needed to obtain the bending shape functions from the solution of the homogeneous equation:

$$
\left[\mathbf{g}\right] = \left(\left[\mathbf{M}^b\right]^{-1}\right)^{\mathrm{T}} \text{ so that } \left\{\mathbf{G}w\right\} = \left[\mathbf{g}\right]\left\{\mathbf{W}^b\right\} \tag{25}
$$

The bending component of all fields derived from Eq. (10) can now be transformed using **g**:

$$\{\mathbf{G}x\} = [\mathbf{g}]\{\mathbf{X}^b\} \tag{26}$$

where x is a field and $\mathbf{X}^b$ its Kelvin functions expression. Finally, the DOFs must be made internal using the membrane polynomial shape functions in preparation for static condensation, which is trivial for w^b and similar to Eq. (19) for u^b:

$$\begin{aligned}
\{\mathbf{N}^b \mid w\} &= \{\mathbf{G}w\} - [\mathbf{I}_4]\{\mathbf{H}\} \\
\{\mathbf{N}^b \mid u\} &= \{\mathbf{G}u\} - \underbrace{\begin{bmatrix} \mathbf{G}u(s_1) & \mathbf{G}u^{\bullet}(s_1) & \mathbf{G}u(s_2) & \mathbf{G}u^{\bullet}(s_2) \end{bmatrix}^{\mathrm{T}}}_{[\mathbf{V}u]}\{\mathbf{H}\}
\end{aligned} \tag{27}$$

These linear combinations also effect other fields, as deduced from the kinematic relations:

$$\begin{aligned}
\{\mathbf{N}^b \mid \chi\} &= \{\mathbf{G}\chi\} - [\mathbf{I}_4]\{\mathbf{H}^{\bullet}\} \\
\{\mathbf{N}^b \mid \varepsilon_s\} &= \{\mathbf{G}\varepsilon_s\} - [\mathbf{V}u]\{\mathbf{H}^{\bullet}\} \\
\{\mathbf{N}^b \mid \varepsilon_\theta\} &= \{\mathbf{G}\varepsilon_\theta\} - \left(\frac{1}{c}[\mathbf{I}_4] + [\mathbf{V}u]\right)\{\mathbf{H}/s\} \\
\{\mathbf{N}^b \mid \kappa_s\} &= \{\mathbf{G}\kappa_s\} - [\mathbf{I}_4]\{\mathbf{H}^{\bullet\bullet}\} \\
\{\mathbf{N}^b \mid \kappa_\theta\} &= \{\mathbf{G}\kappa_\theta\} - [\mathbf{I}_4]\{\mathbf{H}^{\bullet}/s\}
\end{aligned} \tag{28}$$

3.3. Element shape functions and degrees of freedom

The membrane and bending shape functions and corresponding DOFs can now be combined to express the full displacement and strain fields as a product of two vectors:

$$x = \{\mathbf{d}\}^{\mathrm{T}}\{\mathbf{N} \mid x\} \ \text{ with } \{\mathbf{d}\} = \begin{Bmatrix} \mathbf{d}^m \\ \hline \mathbf{d}^b \end{Bmatrix} \ \text{ and } \{\mathbf{N} \mid x\} = \begin{Bmatrix} \mathbf{N}^m \mid x \\ \hline \mathbf{N}^b \mid x \end{Bmatrix} \tag{29}$$

Though the number of bending DOFs and shape functions is fixed, the set of ten membrane DOFs and shape functions may always be expanded with additional internal DOFs to accommodate more complex distributions of p_n and p_s.

3.4. Physical interpretation of the bending shape functions

The CoSBL element's bending shape functions enable the exact representation of the bending behaviour of individual conical shell strakes, and a brief investigation of their functional form allows the identification of dimensionless parameters helpful in characterising conical shell bending. Ker_0 and Bei_0 are shown in Fig. 2 to illustrate the behaviour of the Kelvin functions featured in Eq. (12). Ker and Kei functions are not defined at the apex ($y = 0$) and decrease exponentially with y, while Ber and Bei are defined at the apex and increase exponentially with y. All exhibit an oscillatory behaviour.

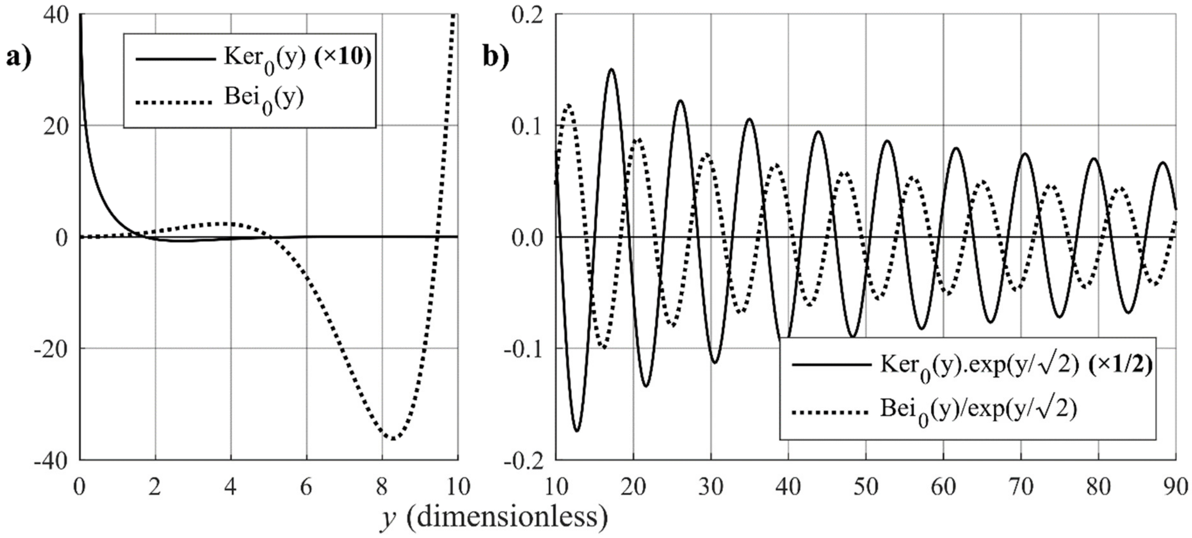

Fig. 2 – Illustration of the oscillatory and exponential behaviour of Kelvin functions.

It can be confirmed through their asymptotic expansion that the behaviour of Kelvin functions becomes increasingly regular with growing y. For example, the first term of the asymptotic expansion of Ker_0 is the following:

$$\mathrm{Ker}_0\ y \sim e^{-y/\sqrt{2}} \sqrt{\frac{\pi}{2y}} \cos\left(\frac{y}{\sqrt{2}} + \frac{\pi}{8}\right) \tag{30}$$

Eq. (30) means that for a large enough argument y, Ker_0 is equivalent to the product of an exponential function of y, a trigonometric function of y and a power of y. This asymptotic expansion function form, shared with the other Kelvin functions, is similar to the transcendental functions identified as solution to the cylindrical bending problem in Boyez *et al.* [8]. The above expression also provides $\pi\sqrt{2}$ as the dimensionless half-wavelength of the oscillation.

3.4.1 The dimensionless y variable

The y variable defined in Eq. (11) is a composite of material (k_{mat} – containing a ratio of the shell membrane to bending stiffnesses) and geometric (k_{geo} – containing a dependency on the slant angle α) parameters and constitutes a dimensionless measure of the distance to the apex:

$$y = 2k_{mat}k_{geo}$$
$$k_{mat} = \left[\frac{12\left(1-\nu^2\right)}{t^2}\right]^{1/4} = \left[\frac{C_m}{C_b}\right]^{1/4} \qquad k_{geo} = \sqrt{\tan\left(\alpha\right)\cdot s} = \sqrt{r\frac{\tan(\alpha)}{\cos(\alpha)}} \tag{31}$$

From Eq. (31), derivatives may be used to determine the relative influences of the three dimensionless parameters ν, r/t and α on y independently of one another:

$$\frac{1}{y}\frac{\mathrm{d}y}{\mathrm{d}\nu} = \frac{-1}{2}\frac{\nu}{1-\nu^2}; \quad \frac{1}{y}\frac{\mathrm{d}y}{\mathrm{d}(r/t)} = \frac{1}{2(r/t)}; \quad \frac{1}{y}\frac{\mathrm{d}y}{\mathrm{d}\alpha} = \tan(\alpha) + \frac{1}{2\tan(\alpha)} \tag{32}$$

It is clear that both v and r/t exhibit only a modest influence on y, as both parameters remain at stable orders of magnitude for typical thin conical metal shells. However, an increase in the slant angle α results in a dramatic increase in y when α is close to 0 (cone departs from a circular plate) or is close to $\pi/2$ (cone approaches a cylindrical shell). With the exception of very moderate slant angles, y should therefore be expected to assume high values in thin cones.

3.4.2 <u>Boundary Layer Independence (*BLIF*) and Asymmetry (*BLAF*) Factors</u>

The length of the conical element $\Delta s = s_2 - s_1$ (Fig. 1) plays a key role in determining whether the element will exhibit an oscillatory or decay/growth behaviour in y-space (Fig. 2). A dimensionless Boundary Layer Independence Factor (*BLIF*) is defined in Eq. (33) as a measure of how many bending half-wavelengths separate the two ends of a conical strake in y-space:

$$BLIF = \frac{y_2 - y_1}{\sqrt{2}\pi} = \frac{k_{mat}\sqrt{2\tan(\alpha)}}{\pi}\left(\sqrt{s_2} - \sqrt{s_1}\right) \tag{33}$$

A high value of the *BLIF* (> ~ 4) signifies a complete absence of interaction between the boundary layers at either end and an element dominated by membrane behaviour, as illustrated in Fig. 3. As such, the *BLIF* can be thought of as an 'effective length' metric.

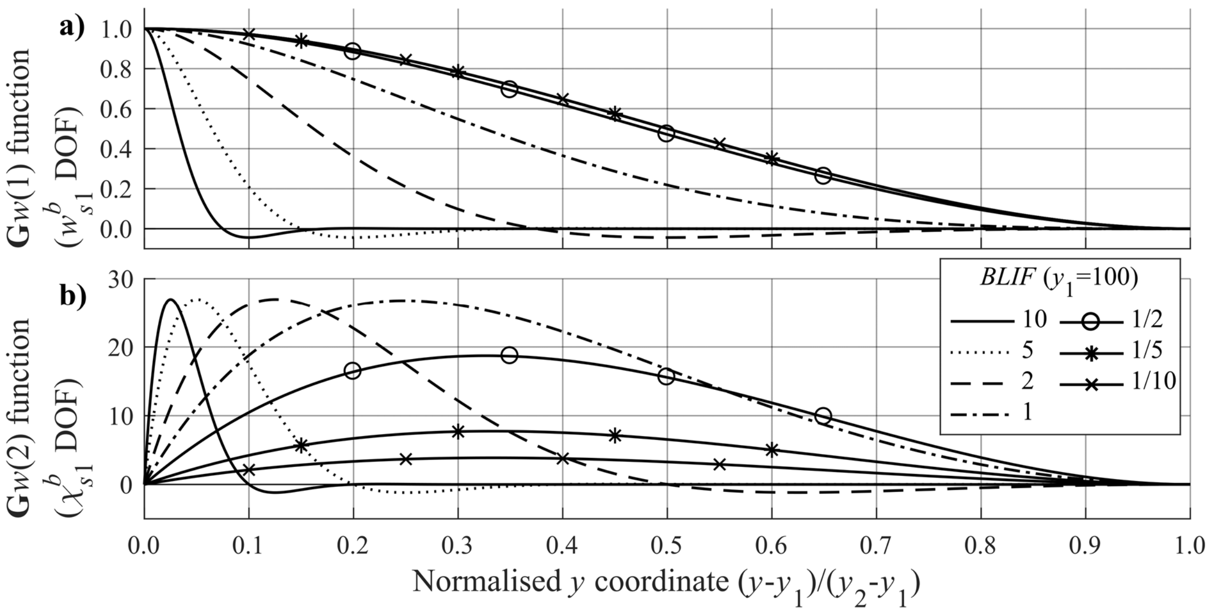

Fig. 3 – Variation of the first two **G**w bending shape functions with the *BLIF*.

An additional measure of the exponential decay and oscillations in terms of the slant height s can also be derived in the form of a bending half-wavelength λ:

$$\lambda = \frac{\pi}{\left[3\left(1-\nu^2\right)\right]^{1/4}}\sqrt{\frac{r \cdot t}{\sin(\alpha)}} = \frac{2\sqrt{2}\pi}{y}s \tag{34}$$

Unlike that of a cylinder, a conical shell's boundary layer length varies from one edge to the other: the further an edge is from the apex, the larger its associated bending half-wavelength. The Boundary Layer Asymmetry Factor (*BLAF*), defined as the ratio of the two bending half-wavelengths at either extremity of the element, captures this effect:

$$BLAF = \frac{\lambda_2}{\lambda_1} = \sqrt{\frac{s_2}{s_1}} = \sqrt{\frac{r_2}{r_1}} = \frac{y_2}{y_1} > 1 \tag{35}$$

A conical shell segment with a *BLAF* close to 1 (i.e. $BLAF - 1 < 10^{-2}$) exhibits symmetric boundary layers and effectively behaves like a cylinder, so that the *BLAF* can be thought of as an 'effective shallowness' metric. Its effect can be observed in Fig. 4 for two values of the *BLIF*. It is noted that when the *BLIF* and *BLAF* both approach 0 and 1 respectively, the conical shell effectively becomes a short cylinder fully dominated by the boundary layer, and the bending shape functions converge to the Hermite cubics (a property also exhibited in the cylindrical case as shown in Boyez *et al.* [8]).

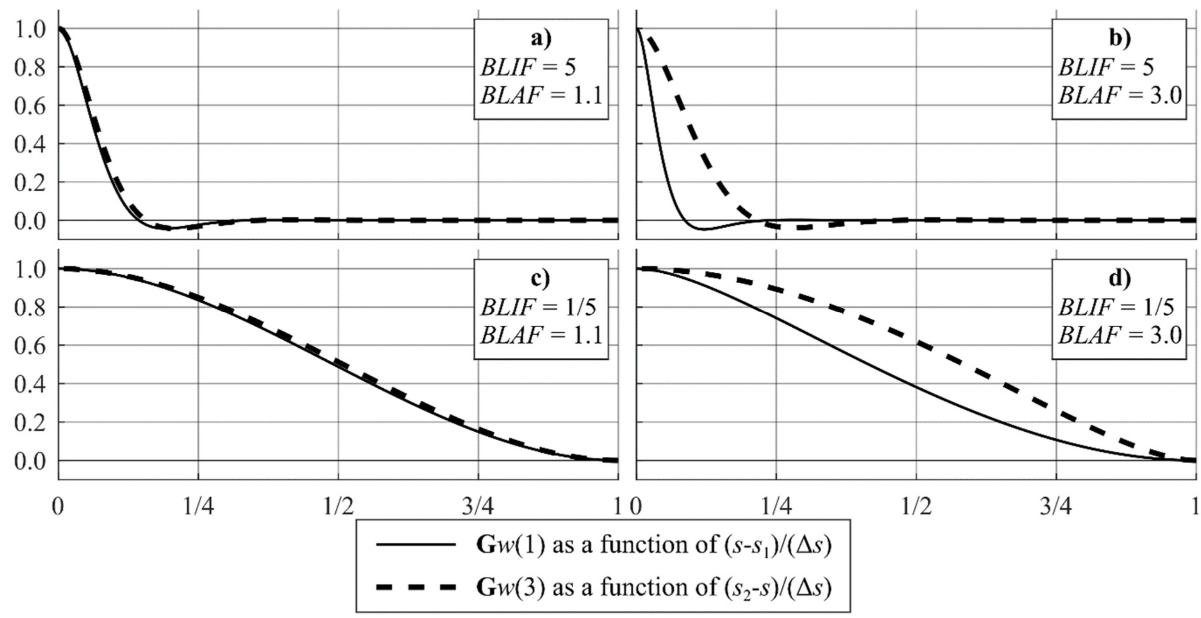

Fig. 4 – Influence of the *BLIF* and *BLAF* on selected bending shape functions at either end of the CoSBL element: a) 'long' and 'steep' cone; b) 'long' and 'shallow' cone; c) 'short' and 'steep' cone; d) 'short' and 'shallow' cone.

To ensure the validity of the thin shell assumptions, lower bounds should be respected for r/t and $\Delta s/t$ (e.g. 50). Further, manufacturing ability or material resistance limit both Δs and $\Delta s/t$ so that practical ranges for the *BLIF* and *BLAF* in actual conical shell strakes (and thus for individual CoSBL elements) can be suggested as:

$$10^{-2} < \left\{ \begin{matrix} BLIF \\ BLAF - 1 \end{matrix} \right\} < 10^{2} \tag{36}$$

4. Stiffness matrix, equivalent force vector, assembly and solution

4.1. Derivation of the stiffness matrix and equivalent force vector

4.1.1 Element stiffness matrix

The stiffness matrix is derived classically by considering the strain energy $\mathscr{E}$ of the element, reduced from a double to a single integral of strains within the cone due to symmetry:

$$\mathscr{E} = \pi \int_{s_1}^{s_2} \begin{pmatrix} C_m \left(\varepsilon_z{}^2 + 2\nu \varepsilon_z \varepsilon_\theta + \varepsilon_\theta{}^2 \right) \\ + C_b \left(\kappa_z{}^2 + 2\nu \kappa_z \kappa_\theta + \kappa_\theta{}^2 \right) \end{pmatrix} r \text{ d}s \text{ with } r = s \cdot \cos(\alpha) \tag{37}$$

The products of strains are expressed as matrix products using Eqs (21)-(22) and (29), with care being taken to maintain matrix symmetry, which leads to the identification of six elementary stiffness matrices $\mathbf{k}_j$:

$$\begin{aligned} \mathscr{E} &= \{\mathbf{d}\}^{\mathrm{T}} \left\{ \pi \int_{s_1}^{s_2} \begin{pmatrix} & C_m \{\mathbf{N} \mid \varepsilon_S\}\{\mathbf{N} \mid \varepsilon_S\}^{\mathrm{T}} \\ + & \nu C_m \left(\{\mathbf{N} \mid \varepsilon_S\}\{\mathbf{N} \mid \varepsilon_\theta\}^{\mathrm{T}} + \{\mathbf{N} \mid \varepsilon_\theta\}\{\mathbf{N} \mid \varepsilon_s\}^{\mathrm{T}} \right) \\ + & C_m \{\mathbf{N} \mid \varepsilon_\theta\}\{\mathbf{N} \mid \varepsilon_\theta\}^{\mathrm{T}} \\ + & C_b \{\mathbf{N} \mid \kappa_S\}\{\mathbf{N} \mid \kappa_S\}^{\mathrm{T}} \\ + & \nu C_b \left(\{\mathbf{N} \mid \kappa_S\}\{\mathbf{N} \mid \kappa_\theta\}^{\mathrm{T}} + \{\mathbf{N} \mid \kappa_\theta\}\{\mathbf{N} \mid \kappa_s\}^{\mathrm{T}} \right) \\ + & C_b \{\mathbf{N} \mid \kappa_\theta\}\{\mathbf{N} \mid \kappa_\theta\}^{\mathrm{T}} \end{pmatrix} r \text{ d}s \right\} \{\mathbf{d}\} \\ &= \frac{1}{2} \{\mathbf{d}\}^{\mathrm{T}} \underbrace{\left([\mathbf{k}_1] + [\mathbf{k}_2] + [\mathbf{k}_3] + [\mathbf{k}_4] + [\mathbf{k}_5] + [\mathbf{k}_6] \right)}_{[\mathbf{k}]} \{\mathbf{d}\} \end{aligned} \tag{38}$$

Computing a $\mathbf{k}_i$ sub-matrix typically involves the following product of interpolation vectors $\mathbf{N}$:

$$\{\mathbf{N} \mid f\}\{\mathbf{N} \mid g\}^{\mathrm{T}} = \left[\begin{array}{c|c|c} \{\mathbf{H} \mid f\}\{\mathbf{H} \mid g\}^{\mathrm{T}} & \{\mathbf{H} \mid f\}\{\mathbf{N}^b \mid g\}^{\mathrm{T}} & \{\mathbf{H} \mid f\}(\mathbf{LP} \mid g)^{\mathrm{T}} \\ \hline \{\mathbf{N}^b \mid f\}\{\mathbf{H} \mid g\}^{\mathrm{T}} & \{\mathbf{N}^b \mid f\}\{\mathbf{N}^b \mid g\}^{\mathrm{T}} & \{\mathbf{N}^b \mid f\}(\mathbf{LP} \mid g)^{\mathrm{T}} \\ \hline (\mathbf{LP} \mid f)\{\mathbf{H} \mid g\}^{\mathrm{T}} & (\mathbf{LP} \mid f)\{\mathbf{N}^b \mid g\}^{\mathrm{T}} & (\mathbf{LP} \mid f)(\mathbf{LP} \mid g)^{\mathrm{T}} \end{array} \right] \tag{39}$$

The $\mathbf{N}^b$ and $\mathbf{LP}$ shape function vectors can further be decomposed with references to the polynomial functions $\mathbf{H}$ and coefficients $\mathbf{V}$ using Eqs (19) and (28):

$$\begin{aligned} \{\mathbf{N}^b \mid f\} &= \{\mathbf{G}f\} - [\mathbf{V}f]\{\mathbf{H} \mid f\} \\ (\mathbf{LP} \mid f) &= (LP \mid f) - \langle \mathbf{V} \mid LP \rangle \{\mathbf{H} \mid f\} \end{aligned} \tag{40}$$

The combination of Eqs (38) to (40) leads to the identification of various sub-blocks required to compute each $\mathbf{k}_j$ sub-matrix, as illustrated by the expansion of sub-block (2,2) from Eq. (39):

$$\begin{aligned}\{\mathbf{N}^b \mid f\}\{\mathbf{N}^b \mid g\}^{\mathrm{T}} = \{\mathbf{G}f\}\{\mathbf{G}g\}^{\mathrm{T}} + [\mathbf{V}f]\{\mathbf{H} \mid f\}\{\mathbf{H} \mid g\}^{\mathrm{T}}[\mathbf{V}g]^{\mathrm{T}} \\ -\{\mathbf{G}f\}\{\mathbf{H} \mid g\}^{\mathrm{T}}[\mathbf{V}g]^{\mathrm{T}} - [\mathbf{V}f]\{\mathbf{H} \mid f\}\{\mathbf{G}g\}^{\mathrm{T}}\end{aligned} \tag{41}$$

$\mathbf{GG}^{\mathrm{T}}$ blocks involve only Kelvin functions, blocks such as $\mathbf{HH}^{\mathrm{T}}$ involve only elementary functions and blocks such as $\mathbf{GH}^{\mathrm{T}}$ involve both. The differences between these three types of blocks calls for adequate integration methods to be defined and investigated in the following subsection.

4.1.2 Equivalent force vector

The derivation of the equivalent nodal force vector $\mathbf{f}$ is similar to that of the stiffness matrix $\mathbf{k}$, starting with the total work W done by the distributed loads p_n and p_s.

$$W = \{\mathbf{d}\}^{\mathrm{T}}\{\mathbf{f}\} \quad \text{or} \quad W_n + W_s = \{\mathbf{d}\}^{\mathrm{T}}\left(\{\mathbf{f}_n\} + \{\mathbf{f}_s\}\right) \tag{42}$$

For a vector of shape functions $\mathbf{N}|p$, the distributed loadings are expressed as follows:

$$p_n = \{\mathbf{N} \mid p\}^{\mathrm{T}}\{\mathbf{p}_n\} \ \text{ and } \ p_s = \{\mathbf{N} \mid p\}^{\mathrm{T}}\{\mathbf{p}_s\} \tag{43}$$

Combining Eqs (42) and (43) yields the expression of load matrices $\mathbf{P}_n$ and $\mathbf{P}_s$:

$$\begin{aligned} &[\mathbf{P}_n] = 2\pi\int_{s_1}^{s_2}\{\mathbf{N} \mid w\}\{\mathbf{N} \mid p\}^{\mathrm{T}}\, r \,\mathrm{d}s \qquad &&[\mathbf{P}_s] = 2\pi\int_{s_1}^{s_2}\{\mathbf{N} \mid u\}\{\mathbf{N} \mid p\}^{\mathrm{T}}\, r \,\mathrm{d}s \\ &\text{so that } \{\mathbf{f}_n\} = [\mathbf{P}_n]\{\mathbf{p}_n\} && \{\mathbf{f}_n\} = [\mathbf{P}_s]\{\mathbf{p}_s\}\end{aligned} \tag{44}$$

Similar to the stiffness sub-matrices, $\mathbf{P}_n$ and $\mathbf{P}_s$ are better expressed as linear combinations of sub-blocks whose computation can be adequately handled:

$$\{\mathbf{N} \mid f\}\{\mathbf{N} \mid p\}^{\mathrm{T}} = \left[\begin{array}{c} \{\mathbf{H} \mid f\}\{\mathbf{N} \mid p\}^{\mathrm{T}} \\ \hdashline \{\mathbf{G}f\}\{\mathbf{N} \mid p\}^{\mathrm{T}} - [\mathbf{V}f]\{\mathbf{H} \mid f\}\{\mathbf{N} \mid p\}^{\mathrm{T}} \\ \hdashline (\mathbf{LP} \mid f)\{\mathbf{N} \mid p\}^{\mathrm{T}} - \langle \mathbf{V} \mid LP\rangle\{\mathbf{H} \mid f\}\{\mathbf{N} \mid p\}^{\mathrm{T}} \end{array}\right] \tag{45}$$

4.2. Practical computation of the stiffness and equivalent force terms

The stiffness and equivalent force matrices are computed in sub-blocks depending on whether the terms include contributions from bending or membrane shape functions. Most integrands involving only the membrane shape functions are polynomials and may be integrated analytically in closed form, while the others may be easily integrated using Gauss-Legendre quadrature. On the other hand, integrands involving bending shape functions or 'bending integrands' do not have closed form analytical integrals, and integrating them is not straightforward, hence the discussion in this section.

4.2.1 Behaviour of the 'bending integrands'

Some integrands involving bending shape functions are featured in Fig. 5 for a *BLIF* of 5, i.e. a relatively 'long' cone. The integrands referred to here as 'BB' are sums of products of two Kelvin functions multiplied by a power of y, while those that involve bending shape functions that are sums of products of one Kelvin function and a power of y are referred to as 'B'. Terms of the latter form have roughly the same behaviour as bending shape functions on their own at high values of y_1, as illustrated in Fig. 5a.

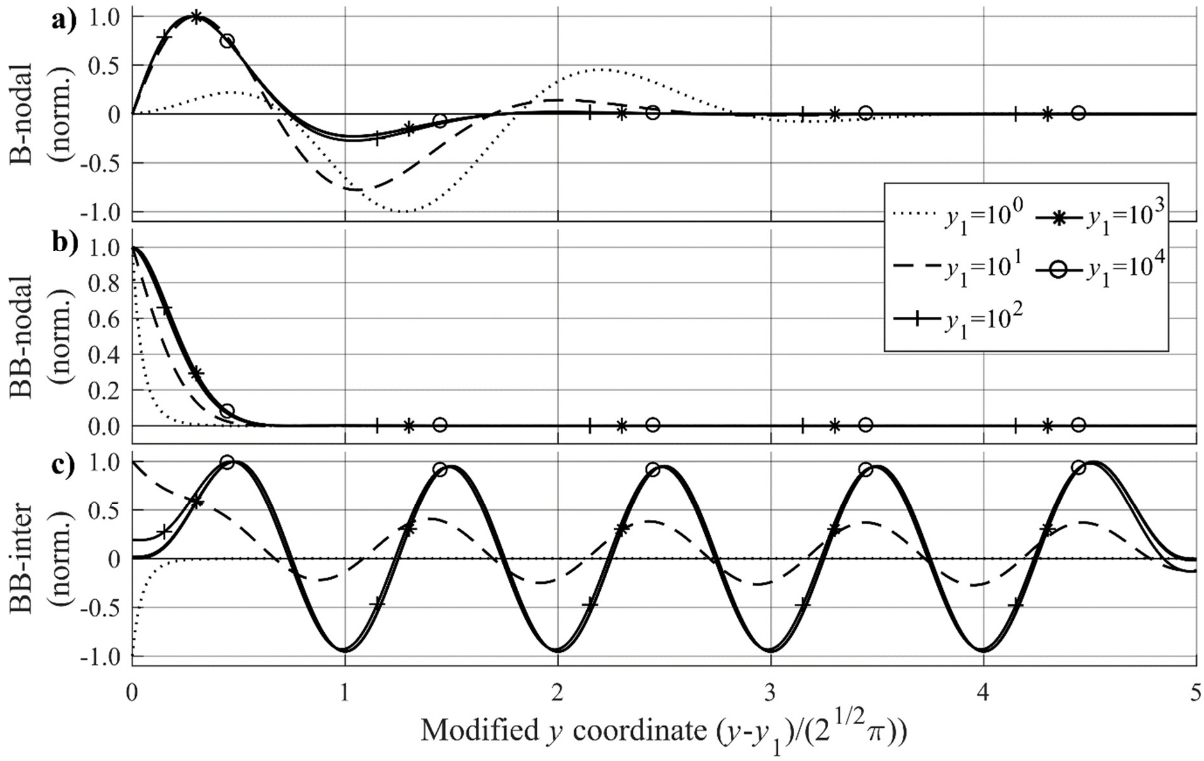

Fig. 5 – Normalised integrands for $\mathbf{k}_1$ associated with the first bending DOF $w^b{}_1$ (*BLIF*=5).

'BB' terms are better understood by further distinguishing the products of shape functions associated with the same node and those associated with opposite nodes of the element, identified with suffixes '-nodal' and '-inter' respectively. Their behaviour for high values of y_1 is readily explained by their asymptotic expansion: '-nodal' terms see their exponential terms combined for a faster decay (Fig. 5b), while '-inter' terms see these terms cancelled and see their oscillation period halved (Fig. 5c). For a high enough *BLIF*, '-inter' terms become negligible compared with '-nodal' ones, independently from y_1, which remains true for other 'BB' sub-matrices and integrands associated with the χ^b DOF.

4.2.2 Precision issues associated with the 'bending integrands'

The bending shape functions from which the 'bending integrands' derive (Eq. 26) are obtained by a normalisation process that involves combinations of Kelvin functions evaluated at both ends of the conical element. Their exponential behaviour, combined with high values of y_1 and y_2 for steep thin conical shells, can lead to exponents exceeding the limit for 'double' precision numbers so that more significant digits must be used in the computation of the stiffness terms. This problem is exacerbated by the fact that solving FE problems requires implicit linear system inversions at the static condensation and global solving steps, operations which are highly precision-sensitive. The implementation of the CoSBL element in the Matlab programming environment [17] therefore also linked to the symbolic Maple mathematical package [18] in order to support arbitrary levels of precision.

4.2.3 Integration scheme used for the 'bending integrands'

The rich range of oscillatory and decay behaviours featured by the 'bending integrands' for different geometric and material parameters makes it difficult to pick a good integration scheme. A selection of integration schemes was devised and compared in a process presented in the Appendix to this paper. The scheme that was finally selected is a 'blunt' Gauss-Legendre numerical integration with a high number of Gauss points to accommodate the wide range of behaviours of the 'bending integrands' and maintain an acceptable precision.

The comparison process also showed that the stiffness terms were especially precision-sensitive for values of the *BLAF* approaching 1 i.e. for very 'steep' cones (Fig. A1). One way to circumvent this problem would be to use a rotated Cylindrical Shell Boundary Element (CSBL) rather than the CoSBL in these cases. The limited tests done by the authors in this regard showed a numerically stable behaviour and precise results.

4.3. Static condensation

Given the elements stiffness matrix $\mathbf{k}$ and element force vector $\mathbf{f}$, static condensation can be performed to work on an overall system with three DOFs per node. The matrix system is reordered so that equilibrium equations related to the six nodal (index *no*) and remaining element-specific (index *el*) DOFs are separated:

$$\begin{bmatrix} \mathbf{k}_{no,no} & \mathbf{k}_{no,el} \\ \mathbf{k}_{el,no} & \mathbf{k}_{el,el} \end{bmatrix} \begin{Bmatrix} \mathbf{d}_{no} \\ \mathbf{d}_{el} \end{Bmatrix} = \begin{Bmatrix} \mathbf{f}_{no} \\ \mathbf{f}_{el} \end{Bmatrix} \tag{46}$$

The element-specific DOFs can be expressed in terms of the nodal ones as follows:

$$\{\mathbf{d}_{el}\} = [\mathbf{k}_{el,el}]^{-1}\left(\{\mathbf{f}_{el}\} - [\mathbf{k}_{el,no}]\{\mathbf{d}_{no}\}\right) \tag{47}$$

Introducing this definition in the first group of equation yields the condensed matrix system:

$$[\mathbf{k}_{cond}]\{\mathbf{d}_{no}\} = \{\mathbf{f}_{cond}\} \text{ with } \begin{cases} [\mathbf{k}_{cond}] = [\mathbf{k}_{no,no}] - [\mathbf{k}_{no,el}][\mathbf{k}_{el,el}]^{-1}[\mathbf{k}_{el,no}] \\ \{\mathbf{f}_{cond}\} = \{\mathbf{f}_{no}\} - [\mathbf{k}_{no,del}][\mathbf{k}_{el,el}]^{-1}\{\mathbf{f}_{el}\} \end{cases} \tag{48}$$

4.4. Assembly and solution

In order to allow for CoSBL elements to be assembled with any axisymmetric thin shell element, the fields defined previously must be transformed from the local (s,n,θ) to the global cylindrical (Z,R,Θ) coordinates (Fig. 1b & c). This can be achieved using a transformation matrix $\mathbf{t}$:

$$[\mathbf{t}] = \begin{bmatrix} \operatorname{sgn}(\cos\beta)\cdot\sin\alpha & \cos\alpha & 0 \\ -\operatorname{sgn}(\cos\beta)\cdot\cos\alpha & \sin\alpha & 0 \\ 0 & 0 & \operatorname{sgn}(\cos\beta) \end{bmatrix} \text{ so that } \begin{Bmatrix} u_i \\ w_i \\ \chi_i \end{Bmatrix} = [\mathbf{t}] \begin{Bmatrix} \delta Z_i \\ \delta R_i \\ \delta\Theta_i \end{Bmatrix} \tag{49}$$

A transformation matrix $\mathbf{T}$ is similarly introduced to transform the local nodal DOFs $\mathbf{d}_{no}$ into their global counterparts $\mathbf{D}$:

$$[\mathbf{T}] = \begin{bmatrix} \mathbf{t} & \mathbf{0} \\ \mathbf{0} & \mathbf{t} \end{bmatrix} \text{ so that } \{\mathbf{d}_{no}\} = [\mathbf{T}]\{\mathbf{D}\} \tag{50}$$

This relation is also used to recast the element equilibrium condition expressed in Eq. (48) into the global system, yielding the global element stiffness matrix $\mathbf{K}$ and equivalent force vector $\mathbf{F}$:

$$\begin{gathered} \{\mathbf{d}_{no}\}^{\mathrm{T}}[\mathbf{k}_{cond}]\{\mathbf{d}_{no}\} = \{\mathbf{d}_{no}\}^{\mathrm{T}}\{\mathbf{f}_{cond}\} \\ \{\mathbf{D}\}^{\mathrm{T}}\underbrace{[\mathbf{T}]^{\mathrm{T}}[\mathbf{k}_{cond}][\mathbf{T}]}_{[\mathbf{K}]}\{\mathbf{D}\} = \{\mathbf{D}\}^{\mathrm{T}}\underbrace{[\mathbf{T}]^{\mathrm{T}}\{\mathbf{f}_{cond}\}}_{[\mathbf{F}]} \end{gathered} \tag{51}$$

The assembly of the overall system must also include the contribution from edge loads. This is accounted for by the addition of nodal force vectors $\mathbf{F}_j$, derived directly in the global coordinate system to express the work W_j done by edge loads at node j:

$$W_j = 2\pi r_j\left(f_r\cdot\delta R_j + m\cdot\delta\Theta_j + f_z\cdot\delta Z_j\right) = \{\mathbf{D}_j\}^{\mathrm{T}}\{\mathbf{F}_j\} \tag{52}$$

Nodal boundary conditions are set following the usual methods. Solving the resulting linear system of equations yields the nodal DOFs, which in turn yield the element-specific DOFs using Eq. (47). All displacement, strain and stress fields may then be deduced from Eq. (29) and the constitutive relations.

5. Illustration of the CoSBL element on two examples

The current capabilities of the CoSBL element are illustrated in this section on two example problems of non-trivial linear elastic stress analysis, arguably the first type of analysis that should be performed in any structural design. The predictions and performance of the CoSBL element are compared against those of two axisymmetric elements, 'classical' in the sense that they rely on *h*-refinement to capture the complex boundary layer behaviour and are therefore representative of the traditional manner in which structural problems of this nature would usually be solved. Following the formulation given in Zienkiewicz *et al.* [9], the 'ThinAxi' element employs four Hermite cubics to interpolate the normal displacement w, and two linear functions to interpolate the meridional displacement u. It uses the same kinematic and constitutive relations as the CoSBL and is also implemented in Matlab for better comparability. SAX2 is the quadratic axisymmetric shell element of Abaqus [19], a general purpose commercial FE program, and constitutes a reliable comparison point.

5.1. Discretisation algorithm for meshing and plotting

A discretisation algorithm was designed to allow for an efficient automated scheme to mesh each conical strake segment with ThinAxi (or similar) elements, based on the rationale that a denser coverage of such simple elements is necessary near the ends of the strake to accurately capture the local curvatures associated with the bending boundary layers. It should be noted that the CoSBL requires no such meshing scheme. Starting from an initial state (state **[A]**), the conical strake is assumed to accommodate up to 5 partitions with n ThinAxi elements in each. If the strake geometry permits it, a partition AA' containing n uniformly-spaced elements is created on one side of the element, A' being a $\lambda_1/2$ distance away from the edge A, where λ_1 is the bending half-wavelength associated with that edge (state **[B]**). The same operation is then attempted at the other edge B of the strake (state **[C]**). A further internal partition is attempted, 2λ away from either edge (states **[D]** and **[E]**). The algorithm stops when any partitions are found to overlap (more likely for short and shallow conical shell strakes, low *BLIF* and high *BLAF* respectively) or the final partitioned state is reached (state **[E]**). The procedure is illustrated in Fig. 6, where d_{AB} is the element density on a segment AB, for example. This algorithm is used in the examples that follow with various choices for the n values, and may be easily adapted to create partitions at other multiples of λ. The same algorithm is also used to compute representative sampling points for plotting the solution fields computed by the CoSBL element, allowing an accurate and efficient rendering of the variation of stresses and displacements within the boundary layer and a minimal plotting cost within the membrane region. It should be added that n = 10 constitutes a 'rule of thumb' boundary layer refinement typically used in analyses of this type, corresponding to an element edge length of 1/10th of a bending half-wavelength λ [8].

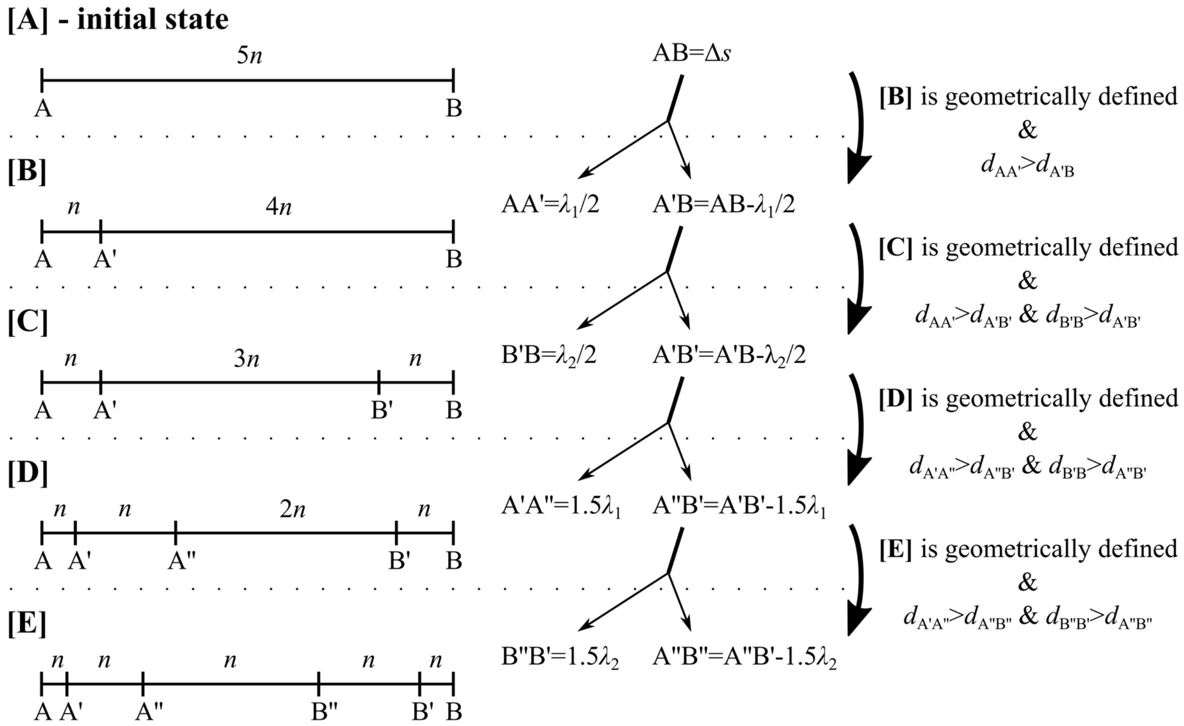

Fig. 6 – Illustration of the discretisation algorithm for meshing of ThinAxi elements and plotting of CoSBL element fields for one conical shell strake segment.

5.2. Two single-strake conical shell segments of opposing geometry

This example aims to illustrate how the CoSBL element offers a superior solution to a typical stress analysis for two individual single-strake conical shell segments (one 'short' and 'shallow' and the other 'long' and 'steep'; Table 1). The objective at this stage is to show that a single CoSBL element is able to support a rich displacement and stress field with an accurate solution to the bending stresses in the boundary layer, and to compare it against the solution offered by meshes of ThinAxi and SAX2 elements obtained with the algorithm introduced above. Use of a single conical shell strake also allows a direct assessment of the accuracy of either element against the analytical solution to the differential equation (Section 2). The two example geometries assume a slant length of Δs = 1000 mm, a thickness t = 5 mm, a Young's modulus E = 200 GPa and Poisson's ratio ν = 0.3. Both are submitted to a constant downward traction p_z = 0.05 MPa (resolved into p_n and p_s components) and a downward vertical load N_z = 5 kN distributed over the top edge, with their bottom edge totally restrained (Fig. 7). The numerical values were chosen purely for convenience.

Table 1 – Properties of two single-strake conical shell segments.

	Short and shallow cone		Long and steep cone	
	Lower edge (1)	Upper edge (2)	Lower edge (1)	Upper edge (2)
r	50.00	1034.81	500.00	673.65
s	50.77	1050.77	2879.39	3879.39
y	4.86	22.13	207.77	241.17
λ	92.74	421.88	123.14	142.93
α	10°		80°	
BLIF	3.89		7.52	
BLAF	4.55		1.17	

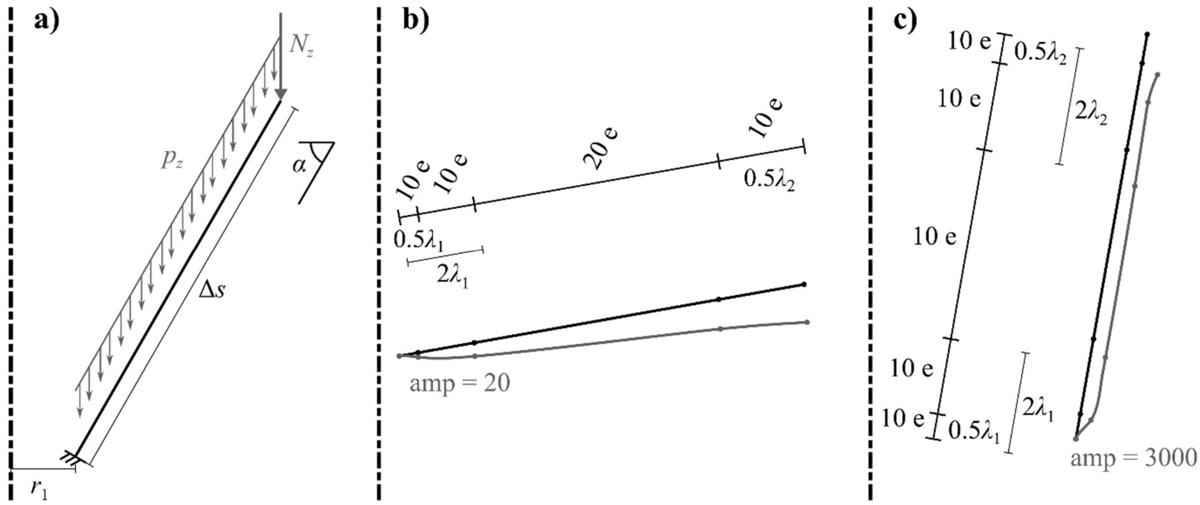

Fig. 7 – a) Geometry and loading for the example; initial shape and scaled deformation and partitioning of the b) short and shallow cone, and c) long and steep cone.

The predictions of the three element solutions for both conical shell segments are shown in Fig. 8. The short and shallow cone exhibits plate-like behaviour dominated by bending, since the boundary layers span the entirety of the segment (*BLIF* < ~4). By contrast, the long and steep cone exhibits cylinder-like behaviour where the loads are carried predominantly by membrane compression, with only very localised and near-symmetric boundary layers near the segment ends.

A density of n = 10 points per element in the ThinAxi and SAX2 mesh was used to sample displacement and stress fields f in order to build the accuracy measure δf, defined in Eq. (53), with the solution to the governing differential equation forming the reference solution. A summary of these accuracies for all solutions is given in Table 2.

$$\delta f = \max_{\text{mesh}} \left(\frac{\left| f_{FE} - f_{analytical} \right|}{\max_{\text{mesh}} \left| f_{analytical} \right|} \right) \cdot 100\% \qquad (53)$$

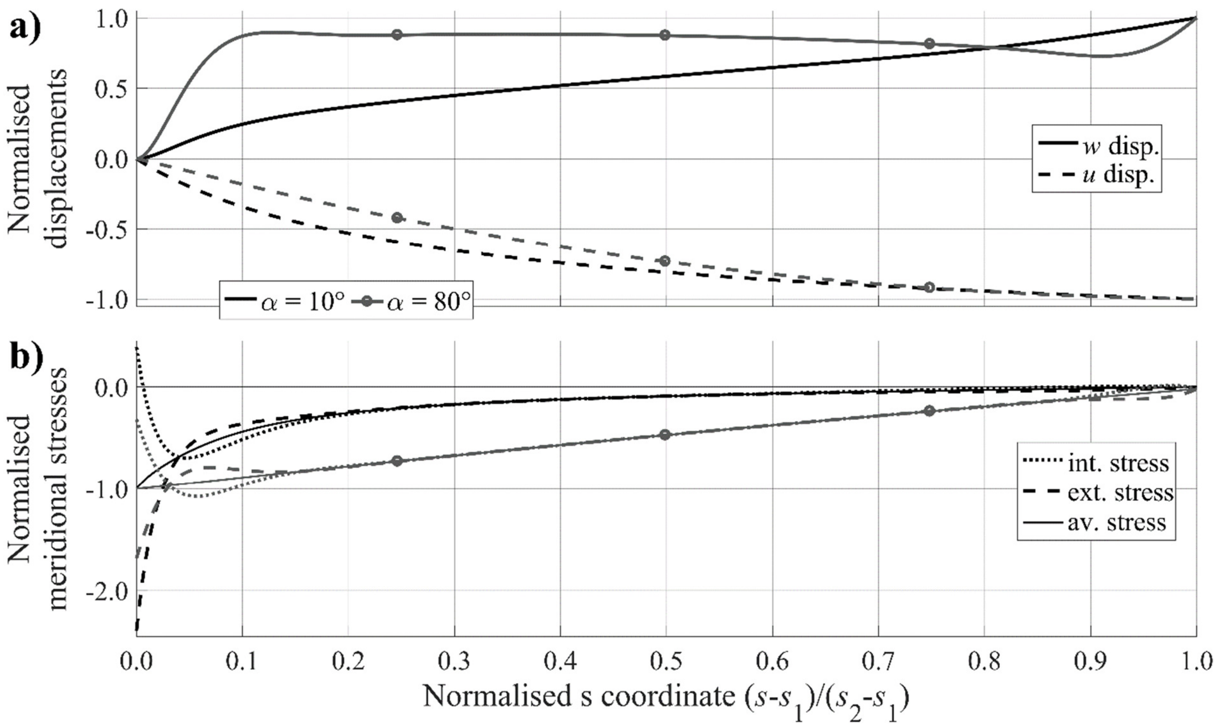

Fig. 8 – Plot of the a) normal (w) and meridional (u) displacements and b) meridional stresses surface stresses (σ_z), each normalised by the max. absolute computed value at the midsurface, against the slant coordinate s normalised to be between zero and unity.

Table 2 – Accuracy of the ThinAxi, SAX2 and CoSBL solutions for displacement and stress fields, compared with the solution to the governing differential equation.

	Short and shallow cone			Long and steep cone		
	ThinAxi	SAX2	CoSBL	ThinAxi	SAX2	CoSBL
δu (%)	0.37	0.06	< 0.001	0.24	0.01	< 0.001
δw (%)	0.16	0.10	< 0.001	0.74	0.47	< 0.002
$\delta \chi$ (%)	0.57	0.81	< 0.001	1.10	1.04	< 0.002
$\delta\sigma_{s,int}$ (%)	1.19	3.13	< 0.001	0.68	0.85	< 0.001
$\delta\sigma_{s,ext}$ (%)	0.39	0.78	< 0.001	0.40	0.54	< 0.001
$\delta\sigma_{\theta,int}$ (%)	2.16	4.30	< 0.001	2.80	2.00	< 0.003
$\delta\sigma_{\theta,ext}$ (%)	1.88	0.98	< 0.001	1.46	0.57	< 0.001

The meshing algorithm described in Fig. 6 allows both the ThinAxi and SAX2 elements to offer a reasonable solution for the displacement fields but a slightly worse one for the stress fields, as expected from low-order polynomial displacement elements. On the other hand, the accuracy of the single-CoSBL element solution is excellent for both geometries, with a maximum normalised error remaining below 5×10^{-3}%, and at least 10 and 50 times smaller than the error for the ThinAxi and SAX2 solutions for displacements and stresses respectively.

5.3. Multi-strake wind turbine support tower under self-weight

A recent computational study by Sadowski *et al.* [20] investigated the behaviour under seismic excitation of a real wind turbine support tower consisting of 22 truncated conical wall strakes with and without realistic weld depression imperfections. Structural details of the tower are shown in Fig. 9 where the reader may verify that a *perfect* tower (without geometric imperfections) exhibits 14 discontinuities (boundaries, rigid flanges and step changes of wall thickness) and thus 26 boundary layers. Each of the boundary layers signifies local compatibility bending of the shell wall and potentially high surface stresses, necessitating extensive local mesh refinement in the meridional direction. Where several adjacent wall segments of the *perfect* tower exhibit the same wall thickness and are strictly aligned, they may in fact be modelled using a single CoSBL element for greater efficiency. Construction of the model with classical 3D linear shell finite elements in Abaqus required extensive use of Python scripting to partition the geometry and apply the appropriate mesh refinement scheme in an automated manner. Other authors which have modelled similar structures [21, 22, 23] do not appear to have given special consideration to a mesh refinement within boundary layers, and it is clear that the technology to model such complex multi-strake structures would benefit from a qualitative advance.

The steel tower is modelled assuming a Young's modulus $E = 200$ GPa, a Poisson's ratio $\nu = 0.3$ and a relative density $RD = 7.85$, while gravity is taken as $g = 9.81$ m/s². The loading consists of the self-weight of the shell (~82 tonnes in total), distributed down the height, as well as that of the wind turbine machinery (90 tonnes), applied as a uniform vertical edge load at the top edge of the tower (point P0; Fig. 9a). The DOFs at the top edge are free while those at bottom boundary are fully restrained. The intermediate flanges at P1 and P2 are modelled as radially rigid but free to displace vertically, restricting only the radial displacement and midsurface rotation DOFs at these locations. The amplified deformed shape is displayed as a dotted line in Fig. 9b, illustrating the global behaviour of the tower as a downward and outward deformation consistent with the vertical loading and the Poisson effect. The radial displacement and meridional stresses obtained with the CoSBL element are shown in Fig. 9c, where the stresses mainly mirror the radial displacement but also feature inter-strake discontinuities and boundary bending. The highest local bending stresses occur near P1, P2 and P3 where the restrictions of the displacement and rotation DOFs are most severe.

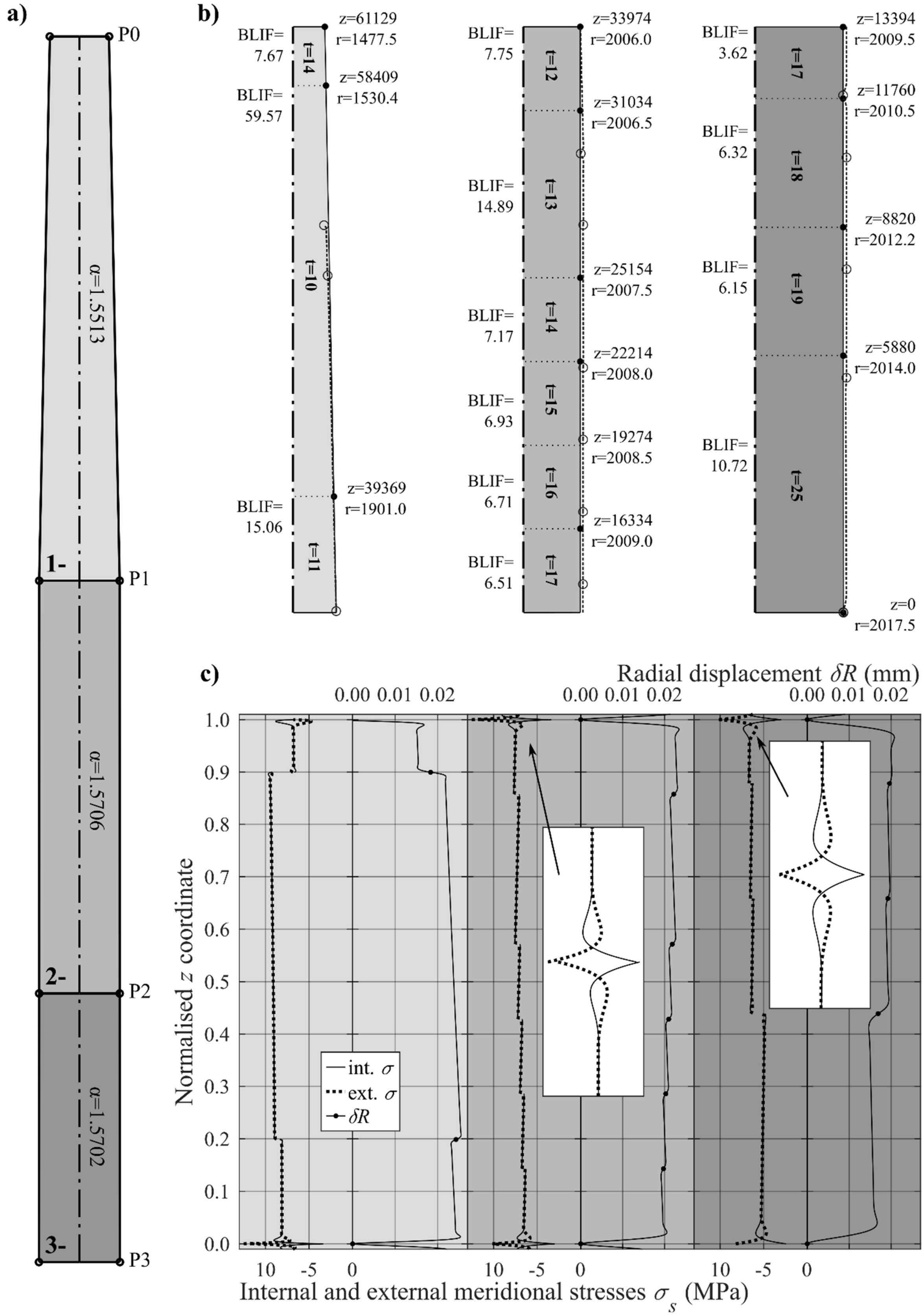

Fig. 9 – a) Geometric overview of the tower – b) Detailed geometric description and amplified deformed shape – c) Computed meridional stresses and radial displacements.

An accuracy measure δ^t was built (Eq. (54)) by taking the maximum over all strakes of the δ measure, defined similarly to Eq. (53), and used to observe the h-convergence of both the ThinAxi and SAX2 predictions for meshes generated using the procedure described in Section 5.1. The number of elements per partition n was varied (Fig. 6), with the 'reference' solution taken as the prediction n = 50 (Table 3 and Table 4Table respectively). In both tables, the last column also shows a comparison of the reference mesh with n = 50 against the predictions of the CoSBL assembly, as its solution is effectively indistinguishable from an analytical solution aside from negligible errors introduced during numerical integration of the stiffness matrix.

$$\delta^t f = \max_{\text{strakes}}\left[\max_{\text{strake mesh}}\left(\frac{\left|f - f_{ref}\right|}{\max\limits_{\text{strake mesh}}\left|f_{ref}\right|}\right)\right]\cdot 100\% \tag{54}$$

Table 3 – Convergence of the ThinAxi solution for displacement and stress fields with mesh refinement against a reference result for n = 50.

Parameter	n = 1	n = 2	n = 5	n = 10	n = 25	n = 50‡
Total DOFs†	198	393	978	1953	4878	9753
$\delta^t u$ (%)	0.44	0.11	<0.02	<0.01	<0.01	<0.01
$\delta^t w$ (%)	5.43	1.23	0.51	0.26	0.07	0.03
$\delta^t \sigma_{s,int}$ (%)	20.81	9.84	3.06	1.30	0.46	0.22
$\delta^t \sigma_{\theta,int}$ (%)	35.72	31.45	14.47	7.35	2.66	1.28

† The total DOFs for the CoSBL assembly was 169.
‡ The last column compares the n = 50 ThinAxi solution against the CoSBL solution.

Table 4 – Convergence of the SAX2 solution for displacement and stress fields with mesh refinement against a reference result for n = 50.

Parameter	n = 1	n = 2	n = 5	n = 10	n = 25	n = 50‡
Total DOFs†	429	819	1989	3939	9789	19539
$\delta^t u$ (%)	0.02	<0.01	<0.01	<0.01	<0.01	<0.01
$\delta^t w$ (%)	20.15	1.00	0.10	0.03	0.01	0.68
$\delta^t \sigma_{s,int}$ (%)	13.46	4.71	0.99	0.26	0.04	0.58
$\delta^t \sigma_{\theta,int}$ (%)	82.65	13.06	2.94	0.80	0.12	1.46

† The total DOFs for the CoSBL assembly was 169.
‡ The last column compares the n = 50 SAX2 solution against the CoSBL solution.

SAX2 is a quadratic element and requires almost double the DOF count as the ThinAxi for the same value of n, yet its rate of h-convergence towards its n = 50 reference solution does not appear to greatly outperform that of the very simple ThinAxi. Both elements obviously require a much higher number of DOFs to converge to a satisfactory result, in particular for stress variables, than the 169 DOFs of the CoSBL 'mesh'. Yet even at a high refinement of n = 50, the ThinAxi solution retains a significant error in the stress variables (Table 3). The shell theory employed by ABAQUS in the SAX2 solution may be slightly different to the one employed here, so the final column in Table 4 should be considered with care although it appears to tell a similar story to the ThinAxi result (which uses the same shell theory as CoSBL).

The rather poor performance of the ThinAxi and SAX2 elements in accurately predicting the bending stresses within the boundary layers of this realistic structure should give the analyst pause. Wind turbine support towers are routinely subject to dynamic excitations arising from wind and blade oscillations [24, 25, 26] in addition to seismic actions where fatigue and cyclic plasticity are the most important limit states and for which an accurate assessment of local bending stresses near discontinuities is crucial. Additionally, in 'real' construction there would usually be a depression-like imperfection at the weld connection between any two adjacent strakes, even those of the same thickness and slant angle. Consequently, the *imperfect* tower modelled by Sadowski *et al.* [20] actually exhibits 23 discontinuities and 44 boundary layers. Each one requires careful local mesh refinement which, when implemented using classical 3D shell elements, results in large models with long runtimes, especially in nonlinear analyses. The authors' development of the 'boundary layer' element aims to alleviate both of these concerns.

Conclusions

This paper has extended upon a recent 'proof of concept' study by the authors to present the linear formulation of a 'boundary layer' shell finite element for an efficient analysis of multi-strake or multi-segment conical shells. One of the key difficulties in modelling such structures is accurately capturing the high local curvatures and surface stresses associated with compatibility bending between two shell strakes, a task requiring extensive mesh design and optimisation when such structures are analysed using classical axisymmetric shell elements. For 3D shells and nonlinear analyses, the problem is compounded several-fold, with the resulting locally-refined meshes coming at a significant penalty in terms of DOFs count and runtime. The proposed element offers to alleviate this difficulty entirely by supporting the boundary layer natively within an enriched displacement field, such that only a single element is necessary per shell segment.

This paper has additionally explored the physical interpretation of the parameters governing the local bending in conical shells. Two important dimensionless groups have been identified, arising naturally from the governing differential equation, which control the extent of the interaction of the boundary layer at either end of a conical shell segment as well as their relative asymmetry. The authors find it surprising that these dimensionless groups do not appear to have been documented in the literature despite conical shells being a classical structural form with several authoritative texts.

The proposed formulation is applicable to a wide range of complex multi-strake conical shell problems, where it has been shown in a real example to offer major computational benefits in terms of increased accuracy and reduced modelling effort. This new approach offers a solid foundation for future developments, and the authors are currently extending the formulation to support asymmetric responses and nonlinearities, starting with Linear Buckling Analysis.

Acknowledgements

The authors are grateful to Kaoshan Dai of Tongji University in Shanghai, China, for kindly providing details of the wind turbine support tower used in one of the examples shown in this study. This research was made possible thanks to a Skempton Scholarship kindly awarded by the Department of Civil and Environmental Engineering of Imperial College London.

References

[1] J. M. Rotter, Bending theory of shells for bins and silos, Transactions of Mechanical Engineering, Institution of Engineers, Australia ME12 (3) (1987) 264–271.

[2] J. M. Rotter, A. J. Sadowski, Cylindrical shell bending theory for orthotropic shells under general axisymmetric pressure distributions, Engineering Structures 42 (2012) 258–265.

[3] T. Yao, X. He, F. Kong, C. Zhou, Design by analysis for orthotropic pressurized structure with small end of conical shell and cylindrical shell based on hill48 yield criterion, Thin-Walled Structures 96 (2015) 220–226.

[4] W. Flügge, Stresses in shells, 2nd Edition, Springer-Verlag Berlin Heidelberg, Germany, 1973.

[5] A. B. Sabir, A. I. Mousa, Finite element analysis of cylindrical-conical storage tanks with girder stiffeners, Thin-Walled Structures 21 (3) (1995) 269–277.

[6] S. A. Tavares, Thin conical shells with constant thickness and under axisymmetric load, Computers & Structures 60 (6) (1996) 895–921.

[7] W. Cui, J. Pei, W. Zhang, A simple and accurate solution for calculating stresses in conical shells, Computers & Structures 79 (3) (2001) 265–279.

[8] A. Boyez, A. J. Sadowski, B. A. Izzuddin, A novel 'boundary layer' finite element for the efficient analysis of thin cylindrical shells, Computers & Structures 182 (2017) 573–587.

[9] O. C. Zienkiewicz, R. L. Taylor, J. Z. Zhu, 14 - Curved rods and axisymmetric shells, The Finite Element Method for Solid and Structural Mechanics (Sixth Edition), Butterworth-Heinemann, Oxford, 2005, pp. 454–474.

[10] D. Chapelle, K.-J. Bathe, The Finite Element Analysis of Shells - Fundamentals, 2nd Edition, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[11] R. S. Bhatia, G. S. Sekhon, Generation of an exact stiffness matrix for a cylindrical shell element, Computers & Structures 57 (1) (1995) 93–98.

[12] R. S. Bhatia, G. S. Sekhon, Generation of exact stiffness matrix for a conical shell element, Computers & Structures 70 (4) (1999) 425–435.

[13] G. S. Sekhon, R. S. Bhatia, Generation of exact stiffness matrix for a spherical shell element, Computers & Structures 74 (3) (2000) 335–349.

[14] R. S. Bhatia, G. S. Sekhon, A novel method of generating exact stiffness matrices for axisymmetric thin plate and shell elements with special reference to an annular plate element, Computers & Structures 53 (2) (1994) 305–318.

[15] H. Reissner, Spannungen in Kugelschalen (Kuppeln), Festschrift Mueller-Breslau, Leipzig (1912) 181.

[16] E. Meissner, Das Elastizitätsproblem für dünne Schalen von Ringflächen-, Kugel-oder Kegelform, Phys. Z 14 (1913) 343.

[17] MATLAB R2017a, The MathWorks Inc. (2017).

[18] Maple 6.14, Maplesoft, a division of Waterloo Maple Inc. (2014).

[19] Abaqus 6.14, Dassault Systèmes Simulia Corp. (2014).

[20] A. J. Sadowski, A. Camara, C. Málaga-Chuquitaype, K. Dai, Seismic analysis of a tall metal wind turbine support tower with realistic geometric imperfections, Earthquake Engineering & Structural Dynamics 46 (2) (2017) 201–219.

[21] N. Bazeos, G. D. Hatzigeorgiou, I. D. Hondros, H. Karamaneas, D. L. Karabalis, D. E. Beskos, Static, seismic and stability analyses of a prototype wind turbine steel tower, Engineering Structures 24 (8) (2002) 1015–1025.

[22] Y. Hu, C. Baniotopoulos, J. Yang, Effect of internal stiffening rings and wall thickness on the structural response of steel wind turbine towers, Engineering Structures 81 (Supplement C) (2014) 148–161.

[23] E. Nuta, C. Christopoulos, J. A. Packer, Methodology for seismic risk assessment for tubular steel wind turbine towers: application to Canadian seismic environment, Canadian Journal of Civil Engineering 38 (3) (2011) 293–304.

[24] K. Thomsen, P. Sørensen, Fatigue loads for wind turbines operating in wakes, Journal of Wind Engineering and Industrial Aerodynamics 80 (1) (1999) 121–136.

[25] V. A. Riziotis, S. G. Voutsinas, Fatigue loads on wind turbines of different control strategies operating in complex terrain, Journal of Wind Engineering and Industrial Aerodynamics 85 (3) (2000) 211–240.

[26] K. Y. Maalawi, H. M. Negm, Optimal frequency design of wind turbine blades, Journal of Wind Engineering and Industrial Aerodynamics 90 (8) (2002) 961–986.

Appendix: selection of an integration scheme for the bending integrands

Three integration schemes were investigated to formulate the CoSBL element stiffness matrix:

- a 'blunt' Gauss-Legendre quadrature over the full CoSBL element domain with a high number of points,
- a 'selective' Gauss-Legendre quadrature over a reduced domain interval,
- an 'analytical' integration using asymptotic expansions.

A normalised matrix norm for the element stiffness matrix **k** was used as a measure of accuracy (Eq. 54) for which a 'reference' (i.e. very accurate) stiffness matrix $\mathbf{k}_{ref}$ was obtained using a 30-point Gauss-Legendre quadrature for all 'B' and 'BB' integrands, with the final evaluation delivering 50 digits of precision. A sample of 2,275 realistic sets of inputs of t, α, Δs and r_1 was generated with limitations within the confines of a thin shell assumption, and for every combination trial stiffness matrices **k** were computed with a final evaluation delivered to 30 digits of precision.

$$\text{Error relative to } \mathbf{k}_{ref} = \frac{\left\|\mathbf{k} - \mathbf{k}_{ref}\right\|}{\left\|\mathbf{k}_{ref}\right\|} \tag{A.1}$$

<u>'Blunt' Gauss-Legendre quadrature</u>

Each additional Gauss point increases the order of the interpolating polynomial by two degrees, helping to offset the localised nature of the 'B' and 'BB' integrands within the boundary layer. The number of Gauss points n_{gp} was not varied across integrands, allowing the costlier Kelvin function evaluations to be performed only once at each point and combined as required in the evaluation of each integrand. This method was tested for different values of n_{gp} (6, 10, 16, 20 and 30 Gauss points), showing as expected that adding Gauss points improves the accuracy (the error relative to $\mathbf{k}_{ref}$ is plotted against *BLIF* and *BLAF* – 1 with logarithmic scaling in Fig. A1a).

It may be noted that all versions of this integration scheme struggle when the *BLAF* approaches unity (i.e. *BLAF* – 1 tends to zero), i.e. for very steep cones that are increasingly cylindrical. This method should arguably work best for very low values of the *BLIF* since for such very short cones the boundary layers interact strongly and the localised exponential behaviour of the integrands is attenuated. It was observed that, for decreasing values of the *BLIF*, the accuracy indeed improved, but only down to ~1 with lower values of the *BLIF* getting progressively worse.

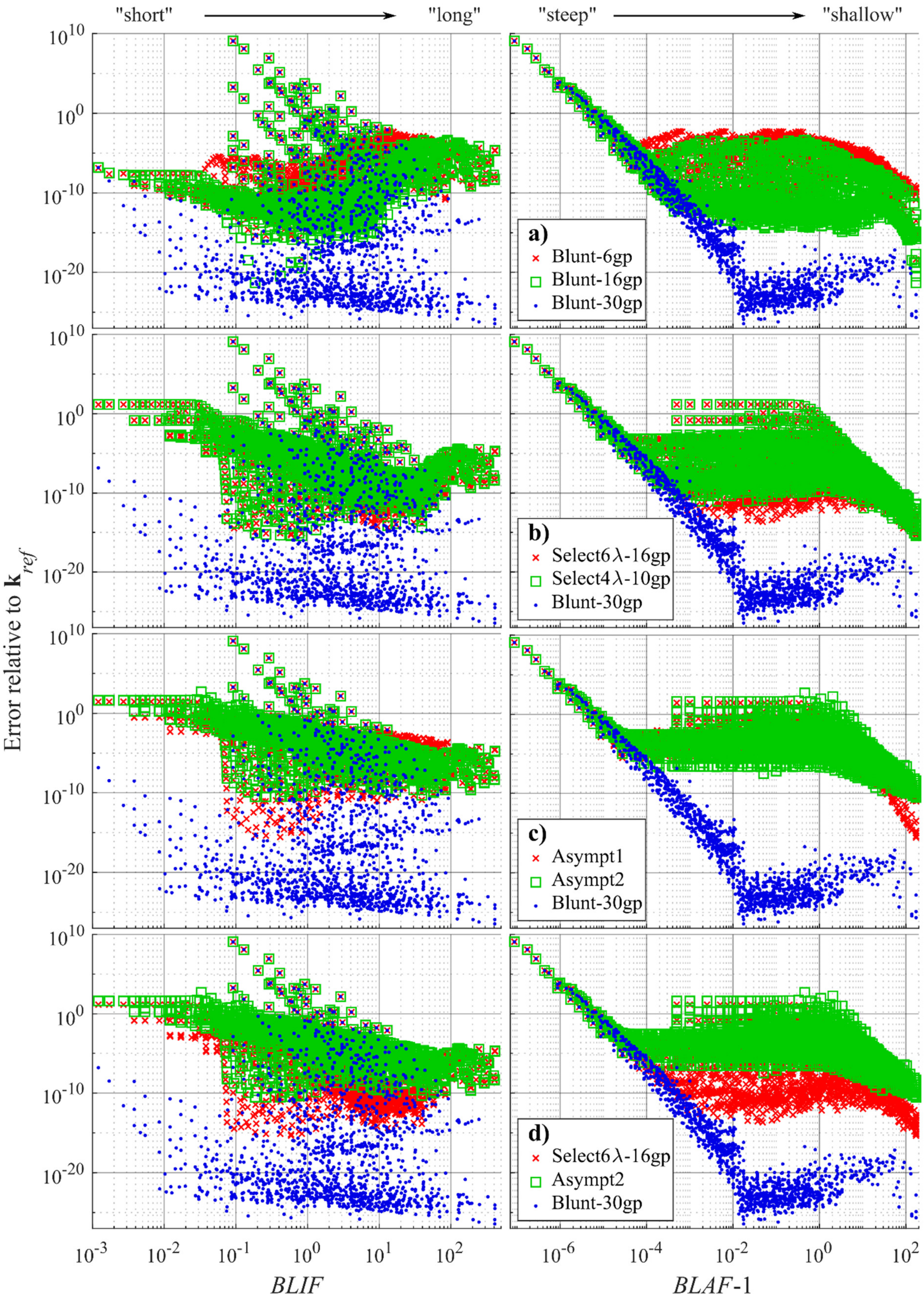

Fig. A1 – Accuracy of the different integration schemes evaluated with 30 digits of precision ('Blunt-30gp' evaluated with 50 digits used as a reference).

‘Selective’ Gauss-Legendre quadrature

This approach exploits the exponential decay of the integrands away from the CoSBL element ends, aiming to perform quadrature on the integrand only within the boundary layer to reduce the computational effort. While this selective scheme is not appropriate for ‘BB–inter’ integrands due to their oscillatory behaviour not being confined to the width of the boundary layer (Fig. 5c), their contribution to the stiffness matrix becomes negligible for high values of the *BLIF* (‘long’ cones). In this scheme, ‘BB–inter’ terms were thus ignored and only ‘BB-nodal’ terms retained. For short conical shells (low *BLIF*), n_{gp} Gauss points were used over the whole element domain to accommodate the interacting boundary layers, but for longer cones (higher *BLIF*) these were placed only over a distance $n_\lambda \lambda_j$ from the element edge j where the integrand is biggest, with n_λ an integer value and λ_j the associated bending half-wavelength.

A set of integer values of n_λ between 1 and 9 was tested for two values of n_{gp} (10 and 16 Gauss points) to determine the influence of both parameters. For very low values of the *BLAF* (‘steep’, near-cylindrical cones), all versions struggle equally for the same reasons as the ‘blunt’ scheme. It was found that, in the remaining sample, the version with $n_\lambda = 4$ seemed most consistent for $n_{\mathrm{gp}} = 10$, while $n_\lambda = 6$ seemed overall better for $n_{\mathrm{gp}} = 16$ (Fig. A1b). However, for cones with a high *BLIF* (i.e. featuring independent boundary layers) and reasonable *BLAF*, for which a selective scheme was expected to be most beneficial, the selective quadrature apparently does not offer any advantage in terms of precision over the simpler ‘blunt’ scheme with 30 Gauss points over the full element domain.

Asymptotic expansions

This approach, theoretically valid only for high values of y, permits a closed-form analytical integration by using the asymptotic expansion to obtain integrands that involve only elementary functions (no Kelvin functions), though some of these approximated integrands require a further asymptotic expansion to obtain a closed-form integral. For example, a typical term of a ‘B’ integrand has the following form:

$$y^m \times \begin{cases} \exp(y/\sqrt{2}) \\ \exp(-y/\sqrt{2}) \end{cases} \times \begin{cases} \cos(y/\sqrt{2}) \\ \sin(y/\sqrt{2}) \end{cases} = \begin{cases} \mathrm{Re}\left[\pm e^{qy} y^m\right] \\ \mathrm{Im}\left[\pm e^{qy} y^m\right] \end{cases} \text{ where } \begin{cases} q = \pm e^{\pm i\pi/4} \\ m \in \mathbb{R} \end{cases} \tag{A.2}$$

The first step is to identify a suitable candidate for the integral of functions of this form:

$$\frac{\mathrm{d}}{\mathrm{d}y}\left(\frac{\mathrm{e}^{qy}}{q} y^m\right) = \mathrm{e}^{qy}\, y^m \left(1 + \frac{m}{qy}\right) \tag{A.3}$$

The residual term on the right-hand side is cancelled out by adding a term on the left-hand side:

$$\frac{\mathrm{d}}{\mathrm{d}y}\left(\frac{\mathrm{e}^{qy}}{q}\,y^m\left[1-\frac{m}{qy}\right]\right)=\mathrm{e}^{qy}\,y^m\left(1-\frac{m(m-1)}{(qy)^2}\right) \tag{A.4}$$

This can be repeated to obtain an asymptotic series whose derivative is equivalent to the integrand. Similar expressions can be derived for other problematic 'B' and 'BB' integrands.

Both asymptotic expansions (for Kelvin function integrands and selected integrals) are performed at the same order p, which should determine the precision and validity range of the method. This method was tested for asymptotic expansions orders 1 and 2, with results shown in Fig. A1c. Low values of the *BLAF* (near-cylindrical cones) similarly prove problematic for either scheme involving asymptotic expansions, as do low values of the *BLIF* where the cylinder is short and dominated by interacting boundary layers. For high values of the *BLIF*, the 2nd order performs better overall, while for very high values of the *BLAF* (very 'shallow' cones) the 1st order seems to work best.

<u>Comparison of the three integration methods</u>

A comparison of the best version of each of the three methods is shown in Fig. A1d, suggesting that the 30-point 'blunt' Gauss-Legendre quadrature consistently outperforms the other schemes in terms of accuracy. It can be noted that all integration schemes face the same issue of loss of precision for very steep cones, which can only be alleviated by adding more digits of precision or, ideally, switching to the CoSBL element's cylindrical counterpart, the CSBL element [8]. For values of the *BLIF* beyond ~20, each of the investigated schemes exhibits a relative error below 0.01% and any one would work reasonably well. In terms of computation time, however, the asymptotic schemes performed consistently worst of all, with runtimes up to 10 times higher than any other scheme due to the large number of individual floating-point operations required to evaluate the rather lengthy expansions. The 'selective' and 'blunt' scheme exhibited a comparable runtime, and for an equal number of Gauss points n_{gp} the 'blunt' scheme requires fewer Kelvin functions evaluations as the same Gauss points are shared for all integrands. As the 'blunt' version is overall more accurate and requires a simpler computational implementation, it was retained as the preferred scheme in the illustrations shown in Section 5 where it was used with 30 Gauss points.