Rakenteiden Mekaniikka (Journal of Structural Mechanics)
Vol. 59, No. 1, 2026, pp. 20–39
https://rakenteidenmekaniikka.journal.fi/index
https://doi.org/10.23998/rm.163217

VERTAISARVIOITU
KOLLEGIALT GRANSKAD
PEER-REVIEWED
www.tsv.fi/tunnus

Improving quality of torsional analysis of prismatic beams by using an interpolated warping function

Simo Nikula[1], Aki Mikkola, Timo Björk and Marko K. Matikainen

Summary This paper examines the torsion of beams with undeformable cross-sections. A novel approach is proposed, in which an interpolated warping function is defined as a continuously differentiable function across the entire cross-sectional area. This formulation enables methods that require the warping value to be accessible as a continuous function at any point within the section. The approach facilitates deformation analysis within the framework of three-dimensional elasticity, employing continuum-based beam elements governed by general continuum mechanics material laws. The primary objective of this study is to establish a general method for determining warping functions for complex cross-sections, intended for future applications in the absolute nodal coordinate formulation. The approach does not replace earlier methods, which have proven highly efficient and even suitable for hand calculations. Instead, it provides a general-purpose framework that can yield improved accuracy in certain cases. Whereas traditional treatments distinguished primarily between open and closed profiles, the present formulation emphasizes the importance of recognizing the nature of each profile and selecting interpolation methods accordingly.

Keywords: warping function, interpolation, torsion warping, absolute nodal coordinate formulation

Received: 1 July 2025. Accepted: 2 February 2026. Published online: 10 March 2026.

Introduction

Machinery and mechanical systems frequently employ beams and shafts that must withstand torsional forces. Ensuring their integrity is vital for maintaining both functionality and safety, for example in the frames of mobile machines.

Accurate modeling of torsional warping is crucial in these applications, since predicting how components respond to torsional loads enables the design of more reliable and efficient machines, thereby reducing the risk of mechanical failure.

Incorporating an advanced modeling approach for torsional warping—particularly through the use of an interpolated warping function defined across the entire cross-sectional area—provides notable advantages. The most important benefit is the ability to represent discontinuities in the analytical model as continuous, which is especially valuable at corners.

[1]Corresponding author: simo.nikula@student.lut.fi

Numerous studies have proposed procedures to address torsional warping problems. These approaches can often be extended to a wider range of cross-sections, provided that the warping function is known. Examples include geometrically exact Kirchhoff beam models that incorporate torsion warping [14]. Efficient modeling and order reduction of new three-dimensional (3D) beam elements with warping can be achieved using the absolute nodal coordinate formulation (ANCF) [25]. This formulation has also been applied to thin-walled composite beam elements, enabling accurate and computationally efficient analysis of complex structures with torsional warping effects [20].

Generalized beam theory (GBT), originally introduced by Schardt [18, 19], can provide warping functions in cases where the centerline approximation is acceptable. The freely available software implementation GBTUL [2] offers practical tools, though its usability is limited by inherently restricted external interfaces.

Interpolation on rectangular domains is well established, but extending support to arbitrary cross-sections requires the development of new methods. Franke's study, "Scattered data interpolation: tests of some methods" [8], compares more than twenty techniques. For realistic cross-sections, warping functions are generally stable, which makes interpolation simpler than in more general cases.

Radial basis functions (RBFs), first introduced by Powell [17] to accelerate large optimization problems and later applied to machine learning by Broomhead and Lowe [4], have since proven effective for approximating multivariate functions [5]. RBFs are particularly useful for interpolation, whereas other methods exhibit notable limitations, as discussed by Buhmann [6].

Numerical integration over non-rectangular or triangular domains (cubature) is often performed through triangulation or by applying the Gauss–Green theorem. For instance, Sommariva and Vianello have employed RBF-based methods for cubature on rectangular domains [22] and used the Gauss–Green theorem for integration over various domains, including non-convex multiply connected polygonal regions [21].

The objective of this paper is to provide a solution for out-of-plane displacement, also referred to as the warping function, for complex cross-sections. Additionally, rectangular cross-sections are analyzed to compare different methods. In practice, this study evaluates the effectiveness of RBF-based interpolation alongside more traditional approaches such as polynomial and NURBS fitting. Within the context of ANCF beams, the proposed process also identifies the shear center, enabling loads to be applied in the same coordinate system in which the cross-section is defined.

Source code and instructions for reproducing results, as well as the generation of all figures and tabulated data, are available on GitHub [15]. Code provided by the main author is licensed under the permissive MIT license. The MATLAB code in `lib`, used for cubature [7], is licensed under the GNU license, which restricts commercial use.

Torsional analysis

The fundamentals of torsional analysis for undeformable cross-sections involving warping can be summarized as follows: (i) Timoshenko and Goodier [26] provided a detailed description of torsion without cross-sectional distortion, building on the work of Saint-Venant. The most relevant aspect for this study is the definition of the warping function. (ii) Pilkey [16] presented a numerical approach for solving the warping function. (iii) Vlasov [27] and Kollbrunner and Hajdin [10] demonstrated how part of the torsional load is carried by axial strains when warping is restricted. (iv) Haukaas [9] offered a clear description of warping torsion using the I-beam as an illustrative example. (v) Björk, Ahola, and Skriko [3] discussed secondary warping in rectangular hollow sections.

The coordinate system is defined such that x and y lie in the cross-sectional plane, while z is aligned with the beam axis. A torsional moment can be expressed as the sum of torque carried by Saint-Venant torsion and torque carried by warping torsion, written as [9]:

$$T = GI_t \frac{d\Theta}{dz} - EI_w \frac{d^3\Theta}{dz^3} \tag{1}$$

where G is the shear modulus, E is Young's modulus, and Θ is the twist angle relative to the z-axis. In Eq. (1), I_t denotes the torsional moment of inertia (also known as the Saint-Venant torsion constant), and I_w is the warping constant. This equation also forms the basis of torsion in GBT. Including distributed torque along the beam axis, the governing differential equation becomes:

$$m_z = EI_w \frac{d^4\Theta}{dz^4} - GI_t \frac{d^2\Theta}{dz^2} \tag{2}$$

Defining the torsion constant k as

$$k = \sqrt{\frac{GI_t}{EI_w}} \tag{3}$$

the homogeneous solution can be written as

$$\Theta(z) = C_1 \sinh(kz) + C_2 \cosh(kz) + C_3 z + C_4, \tag{4}$$

where $C_1 \ldots C_4$ are constants determined by boundary conditions.

As summarized in [26], the torsion problem can be reduced to finding a warping function that satisfies Eq. (5) together with the boundary condition in Eq. (6). Analytical solutions are feasible only for very simple cross-sections and require the use of series expansions involving hyperbolic functions.

The numerical solution presented in [16] has been implemented in commercial finite element packages such as ANSYS. The open-source package *section-properties* [11] provides similar functionality. Other implementations for solving the warping function also exist, e.g. [24, 12, 13], and these papers include detailed theoretical descriptions of the solution process.

As described in the theory section of the [11] documentation, the warping function ω is obtained by solving the partial differential equation

$$\nabla^2 \omega = 0 \tag{5}$$

subject to the boundary condition that shear stress at the boundary is directed tangentially:

$$\frac{\partial \omega}{\partial x} n_x + \frac{\partial \omega}{\partial y} n_y = y n_x - x n_y \tag{6}$$

where n is the unit normal at the boundary, and x and y are the coordinates of the point.

As shown in [16], applying the finite element method reduces the problem to a system of linear equations:

$$\mathbf{K}\boldsymbol{\omega} = \boldsymbol{F} \tag{7}$$

where $\mathbf{K}$ and $\boldsymbol{F}$ are assembled by summation at the element level:

$$\mathbf{k}^e \boldsymbol{\omega}^e = \boldsymbol{f}^e \tag{8}$$

The elemental stiffness matrix is defined as

$$\mathbf{k}^e = \int_\Omega \mathbf{B}^T \mathbf{B} J_e \, d\eta \, d\xi \, d\zeta \tag{9}$$

where $\mathbf{B}$ is the matrix of partial derivatives of the shape functions $\boldsymbol{N}$ with respect to Cartesian coordinates, J_e is the Jacobian determinant, and η, ξ, ζ are the isoparametric coordinates of triangular elements.

In *section-properties* [11], the vector of shape functions $\boldsymbol{N}$ is formulated as:

$$\boldsymbol{N} = \left[\eta(2\eta-1), \xi(2\xi-1), \zeta(2\zeta-1), 4\eta\xi, 4\xi\zeta, 4\eta\zeta\right] \tag{10}$$

The load vector is defined as

$$\boldsymbol{f}^e = \int_\Omega \mathbf{B}^T \begin{bmatrix} \boldsymbol{N} y \\ -\boldsymbol{N} x \end{bmatrix} J_e \, d\eta \, d\xi \, d\zeta \tag{11}$$

Numerical integration of the stiffness matrix and load vector yields:

$$\mathbf{k}^e = \sum_j w_j \mathbf{B}_j^T \mathbf{B}_j J_e \tag{12}$$

$$\boldsymbol{f}^e = \sum_j w_j \mathbf{B}_j^T \begin{bmatrix} \boldsymbol{N}_j y_e \\ -\boldsymbol{N}_j x_e \end{bmatrix} J_e \tag{13}$$

where w_j are the integration weights.

Once the warping function has been evaluated, the Saint-Venant torsion constant I_t can be calculated as

$$I_t = I_{xx} + I_{yy} - \boldsymbol{\omega}^T \mathbf{K} \boldsymbol{\omega} \tag{14}$$

where I_{xx} and I_{yy} are the second moments of area.

The warping constant I_w is obtained from the warping function and the coordinates of the shear center relative to the centroid of the cross-section:

$$I_w = I_\omega - \frac{{Q_\omega}^2}{A} - y_s I_{x\omega} + x_s I_{y\omega} \tag{15}$$

$$Q_\omega = \int_\Omega \omega \, d\Omega = \sum_i \sum_j w_j \mathbf{N_j} \boldsymbol{\omega_i} J_i \tag{16}$$

$$I_\omega = \int_\Omega \omega^2 \, d\Omega = \sum_i \sum_j w_j (\mathbf{N}_j \boldsymbol{\omega}_i)^2 J_i \tag{17}$$

Constraining or neglecting warping has two distinct implications. Neglecting warping in Eq. (14) makes a non-circular cross-section appear stiffer than it truly is. This simplification was common in the early development of torsional theory and should be considered when formulating new models. In Eq. (1), warping corresponds to the derivative of rotation and is typically constrained at the beam end if it is connected to a stiff support. Omitting the term associated with I_w increases rotations, since no energy is expended to produce warping deformation.

Characteristics of warping functions

If centerline theory is applied, I_w can be expressed as the sum of primary and secondary parts, as shown in Eq. (18) [3]:

$$I_w = \int_s \omega^2 t\, ds + 2 \int_s \int_{r_n=0}^{t/2} r_n^2 s^2\, dr_n\, ds \tag{18}$$

where t is the plate thickness, s is the distance along the centerline, and r_n is the normal to the centerline.

Primary warping depends on the geometry of the cross-section. An approximation of secondary warping for a flat region of length l can be written as

$$I_w = \frac{l^3 t^3}{144} \tag{19}$$

assuming that the ratio l/t is sufficiently large. Under the same assumption, I_t can be expressed as

$$I_t = \frac{l t^3}{3} \tag{20}$$

Figure 1 shows the values of I_w and I_t as ratios to the expected values based on Eqs. (19) and (20). Figure 1 was generated using section `swxy` in paper_cells.py.

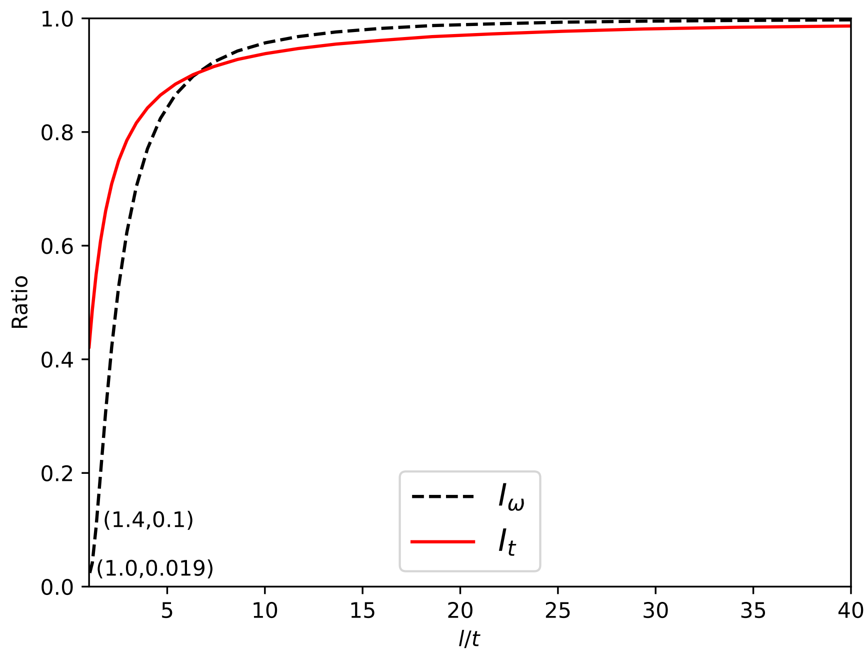


Figure 1. Ratio of actual values obtained using *section-properties* to thin-wall theoretical values

Most thin-walled steel products have l/t ratios between 10 and 40, whereas products made from softer materials typically exhibit lower l/t ratios to compensate for their reduced yield strength. To exploit very high yield stresses, a lower l/t ratio may be required to prevent local buckling.

Construction of interpolation functions

The open-source package *section-properties* [11] is used in this study to obtain discrete solutions for the warping function and the cross-section properties of the selected geometry. The warping function is exported both as a text file for further processing and as a `.glb` file (GL Transmission Format binary) for visualization.

MATLAB is employed to construct interpolation functions. In this work, three different surface fitting models were applied using MATLAB's `fit` routine: `poly44`, `cubicinterp`, and `tps`. Their mathematical formulations are summarized below.

Polynomial Surface (`poly44`)

The `poly44` model represents a polynomial of degree four in both variables x and y:

$$f(x, y) = \sum_{i=0}^{4} \sum_{j=0}^{4} a_{ij} \, x^i y^j, \tag{21}$$

where the coefficients a_{ij} are determined by minimizing the least-squares error:

$$\min_{a_{ij}} \sum_{k=1}^{N} \left(z_k - f(x_k, y_k)\right)^2 . \tag{22}$$

The advantage of the `poly44` model is its simple closed-form representation, which makes evaluation and differentiation straightforward. Its limitations are twofold: higher-order polynomials may exhibit unwanted oscillations, and the fixed fourth-degree form can lack sufficient flexibility to capture highly complex warping functions.

Cubic Interpolation Surface (`cubicinterp`)

The `cubicinterp` model constructs a piecewise cubic spline surface that interpolates all data points exactly:

$$f(x, y) = \sum_{i} \sum_{j} c_{ij} \, B_i(x) \, B_j(y), \tag{23}$$

where $B_i(x)$ and $B_j(y)$ denote cubic basis functions. The interpolation conditions enforce

$$f(x_k, y_k) = z_k \quad \text{for } k = 1, \ldots, N. \tag{24}$$

The strength of the `cubicinterp` model lies in its ability to reproduce the data points exactly with smooth local behavior. Its limitation is that it does not extrapolate beyond the convex hull of the data, which can restrict derivative evaluation near boundaries.

Thin-Plate Spline Surface (`tps`)

The `tps` model minimizes the bending energy of the surface while interpolating the data. Its general form is

$$f(x, y) = a_0 + a_1 x + a_2 y + \sum_{k=1}^{N} w_k \, U\left(\sqrt{(x - x_k)^2 + (y - y_k)^2}\right), \tag{25}$$

where the radial basis function is defined as

$$U(r) = r^2 \ln(r). \tag{26}$$

The weights w_k are obtained by solving a linear system that enforces the interpolation conditions:

$$f(x_k, y_k) = z_k \quad \text{for } k = 1, \ldots, N. \tag{27}$$

An important advantage of the `tps` model is that it is defined over the entire plane $\mathbb{R}^2$, not only within the convex hull of the data points. This property ensures that the function and its partial derivatives remain well-defined even outside the specified domain. As a result, TPS interpolation provides smooth extrapolation beyond the data region, which is particularly useful when evaluating derivatives near the boundary of the domain. Its limitation is that solving the associated linear system can be computationally demanding for large datasets.

Measuring the quality of interpolation

The quality of interpolation is assessed by calculating the warping constant in Eq. (15) using the chosen interpolation function together with a suitable cubature algorithm. The results are then compared either with analytical solutions or with numerical values obtained from *section-properties* analysis.

As a secondary measure, the maximum absolute value of the interpolated function is compared to the corresponding value provided by *section-properties*. For cross-sections that are not doubly symmetric, the location of the shear centre is also required, and this is taken directly from *section-properties* results.

Algorithms suitable for practical cross-sections must be capable of handling realistic geometries, including rounded corners and internal holes. Cubature methods based on scattered data have been described and compared in [7], and the methods employed in this study are outlined below.

Numerical Integration with `integral2`

To evaluate double integrals of functions of two variables, MATLAB's `integral2` routine was employed. The method computes

$$I = \iint_D f(x, y)\, dx\, dy, \tag{28}$$

where D is taken as a rectangular domain in this study, or more generally as

$$x \in [x_{\min}, x_{\max}], \quad y \in [y_{\min}(x),\, y_{\max}(x)]. \tag{29}$$

The algorithm is based on adaptive quadrature rules. For each fixed x, the inner integral

$$g(x) = \int_{y_{\min}(x)}^{y_{\max}(x)} f(x, y)\, dy \tag{30}$$

is approximated using numerical quadrature with error control. Subsequently, the outer integral

$$I = \int_{x_{\min}}^{x_{\max}} g(x)\, dx \tag{31}$$

is computed adaptively, refining the partition of the domain until the estimated error falls below a specified tolerance.

The main advantage of `integral2` is its ease of use and built-in error control, which make it well suited for functions defined on rectangular domains. Its limitation is that it is not directly applicable to complex cross-sections with curved boundaries or internal holes, where more general cubature methods are required.

Glaubitz Method

The Glaubitz approach, referred to as LSCF in [7], constructs provably positive and exact cubature formulas for polynomial spaces. Given a domain $D \subset \mathbb{R}^d$, the integral

$$I = \int_D f(\mathbf{x})\, d\mathbf{x} \tag{32}$$

is approximated by

$$I \approx \sum_{k=1}^{M} w_k f(\mathbf{x}_k), \tag{33}$$

where the nodes $\mathbf{x}_k$ and weights w_k are determined via least-squares optimization such that the formula is exact for all polynomials up to a chosen degree. Positivity of the weights ensures numerical stability.

The main advantage of the Glaubitz method is its theoretical guarantee of positivity and exactness for polynomial spaces, which provides robust and stable integration results. Its limitation is that the construction of nodes and weights can become computationally demanding for higher dimensions or complex domains, which may restrict its applicability in large-scale problems.

RBFCUB (Radial Basis Function Cubature)

RBFCUB [23] is a meshless cubature scheme based on radial basis function interpolation. The integrand is approximated as

$$f(x, y) \approx \sum_{j=1}^{N} \lambda_j\, \phi(\|(x, y) - (x_j, y_j)\|)\,, \tag{34}$$

where $\phi(r)$ is a radial basis function (e.g. Gaussian or multiquadric). The coefficients λ_j are obtained by solving the interpolation system. The integral over a polygonal domain D is then computed as

$$I \approx \sum_{j=1}^{N} \lambda_j \int_D \phi(\|(x, y) - (x_j, y_j)\|)\, dx\, dy. \tag{35}$$

The main advantage of RBFCUB is its meshless nature, which makes it well suited for complex geometries and scattered data without requiring a structured grid. It also provides flexibility through the choice of radial basis functions. Its limitation is that the accuracy and stability depend strongly on the selection of the basis function and its parameters, and the method can become computationally expensive for large datasets.

Numerical examples

Reported solution times are included to provide a rough estimate of the computational effort required. All timings were obtained on a laptop equipped with an Intel i7-12700H processor.

The numerical examples focus on selected cross-sections with varying geometrical complexity, including simple rectangular sections, sections with rounded corners, and sections containing internal holes. These cases illustrate the performance of the interpolation functions and cubature methods under different practical conditions.

Analytical solution for rectangular cross-sections

Analytical results are available for rectangular cross-sections. The warping constant values are obtained by integrating the warping function for the rectangle, Eq. (36) [1], over the cross-section. The summation index n is indicated in parentheses. A MATLAB implementation is provided in the file iw_rect.m.

$$\Psi(x, y) = xy - \frac{32a^2}{\pi^3} \sum_{n=0}^{\infty} \frac{(-1)^n \sin(k_n x) \sinh(k_n y)}{(2n+1)^3 \cosh(k_n b)} \tag{36}$$

where a and b denote the half-width and half-height of the rectangle with $a > b$, and

$$k_n = \frac{(2n+1)\pi}{2a}, \quad n = 0, 1, 2, \ldots \tag{37}$$

To investigate how the different surface fitting methods can be applied to describe the warping function, and to determine how many interpolation points are required, the MATLAB routine iw_fits.m was developed. This routine simulates a numerical solution by generating a selected number of edge points using the `linspace` function and random interior points using `haltonset`.

GBTUL solution

GBTUL results are obtained by constructing an input file based on the parameters provided by *section-properties* and executing the GBT solver through a Python script. The number of nodes used to describe the corners is indicated in parentheses.

Ansys solution

Ansys results are obtained using the Ansys Mechanical APDL Preprocessor via *Sections / Beam / Common Sections*. The rectangle parameters are defined by N_b (number of elements in the horizontal direction) and N_h (number of elements in the vertical direction), with the chosen values indicated in parentheses.

Interpolation based on section-properties analysis

The *section-properties* output lines include the requested element size, together with the resulting number of nodes and elements, indicated in parentheses. Following each line, multiple entries are provided for results that are based on the interpolation function. These results are obtained by running a MATLAB fitting routine to construct an interpolation function for the warping function.

The cardinality parameter, i.e. the number of points used to evaluate the warping function with RBFCUB and Glaubitz cubature, is also shown in parentheses.

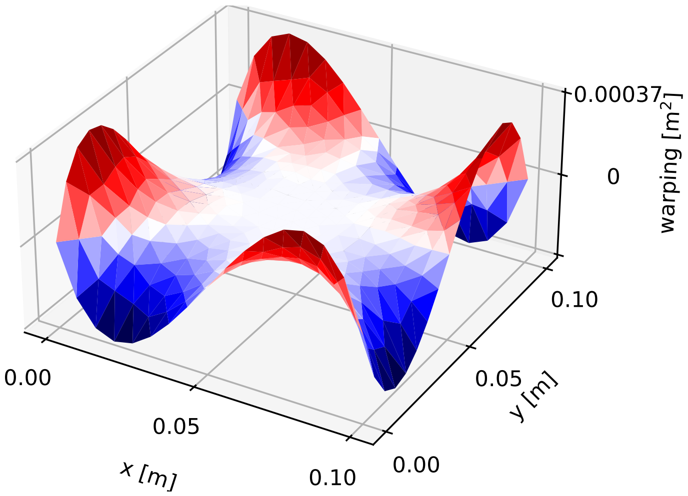


Figure 2. Square solid rectangle, 100×100 mm, used in tests

Square solid rectangle

Cross-section details and the warping function are shown in Fig. 2, with $h = 0.1$ m, $b = 0.1$ m, and the corresponding 3D model available at warping-rectangle-100-100-357.glb.

As shown in Fig. 3, the `poly44` interpolation is most effective for a square cross-section. A few effective meshes exist for TPS and cubic interpolation, but in general several hundred points are required to reduce the error below one percent.

The error percentage in Fig. 3 and Fig. 5 is calculated as

$$100 \cdot \frac{Iw - Iw_a}{Iw_a}, \tag{38}$$

where Iw is the result from numerical analysis using the corresponding parameters, and Iw_a is the analytical result obtained from Eq. (36). In these figures, results obtained with linear interpolation are also shown for comparison. Linear interpolation was not used elsewhere in the study, since it does not yield continuous derivatives and therefore lacks the smoothness required for torsional stress evaluation. Details of the figure generation are available in iw_fits.m.

Numerical results for selected scenarios are shown in Table 1. The analytical and interpolation-based results can be reproduced by running paper_cells.m. The third column of the table reports the error percentage epc of warping obtained using the interpolation function, compared to the *section-properties* results at the point that produced the largest absolute value of the warping function in the *section-properties* analysis.

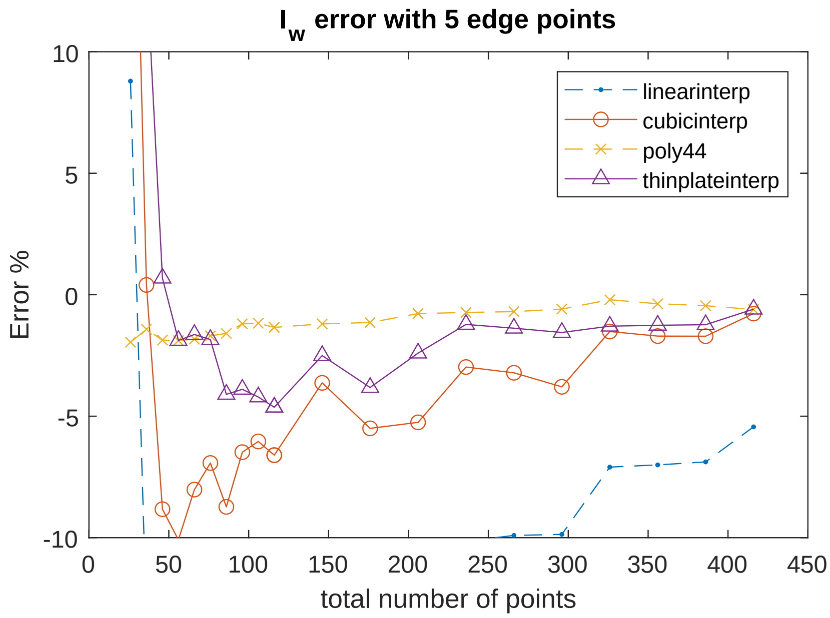
These results demonstrate that several hundred integration points are required to reduce the error to approximately one percent.

Solid rectangle

Cross-section details and the corresponding warping function are shown in Fig. 4, with $h = 0.01$ m and $b = 0.1$ m. A 3D model of the warping function is available at warping-rectangle-10-100-375.glb.

As illustrated in Fig. 5, the `poly44` interpolation is not particularly effective for non-square cross-sections. For TPS and cubic interpolation, approximately one hundred points are required to reduce the error below one percent.

Selected numerical results are shown in Table 2.



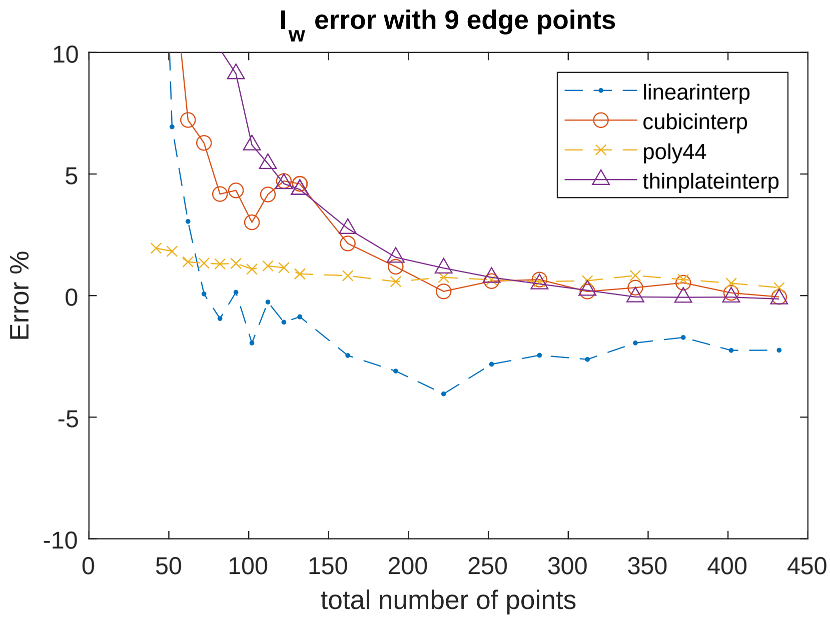


Figure 3. Warping constant error for a 100×100 mm cross-section using selected interpolations.

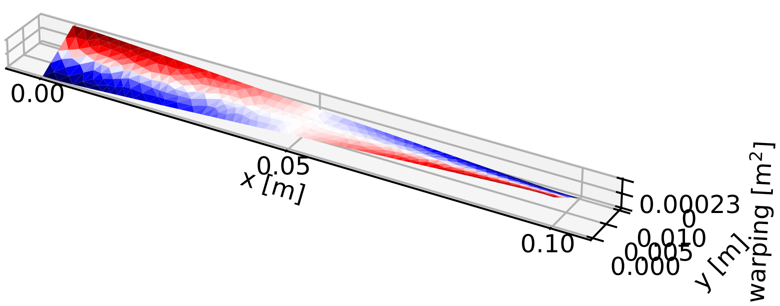


Figure 4. Solid rectangle 100×10 mm used in tests

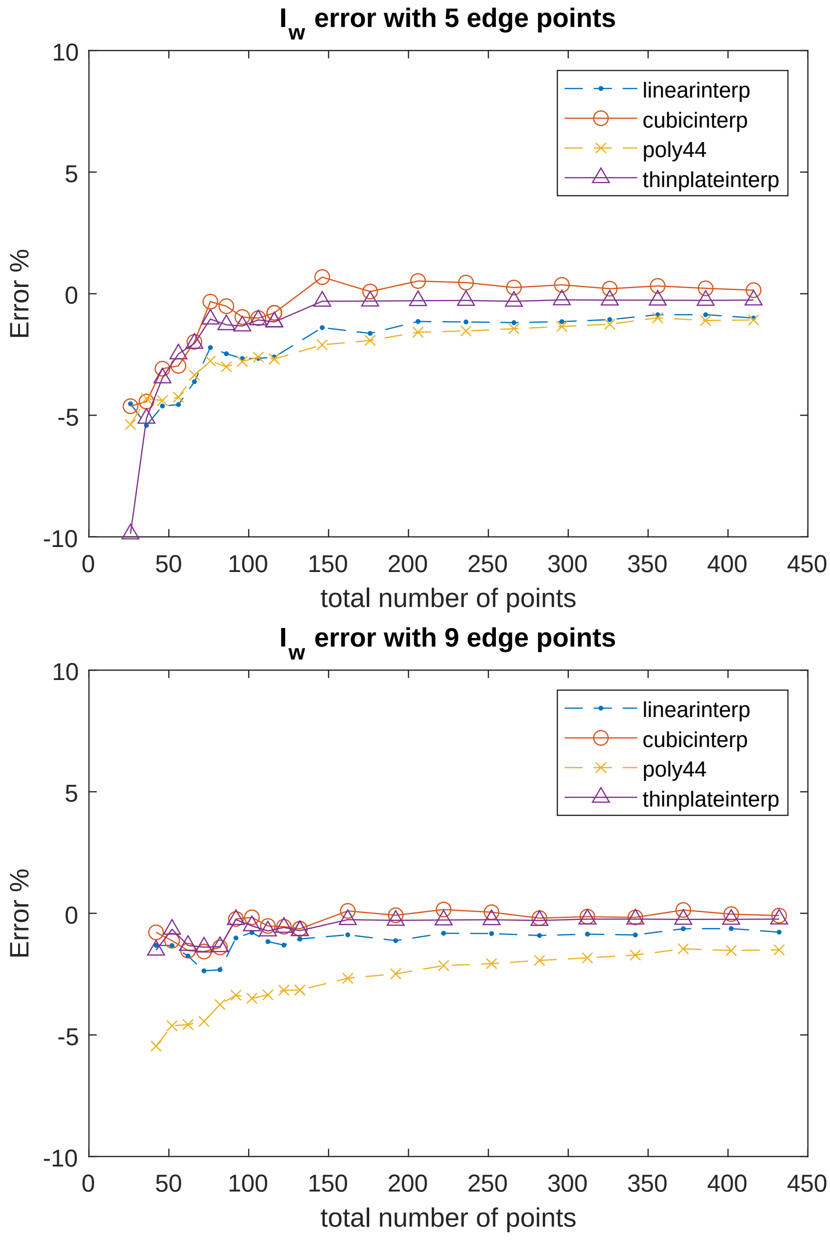


Figure 5. Warping constant error for 100×10 mm using selected interpolations.

Table 1. Warping constant values for solid 100×100 mm rectangle

Source	I_w $[10^{-12} m^6]$	epc [%]
Analytical(0)	168	
Analytical(3)	135	
Ansys(4,4)	105	
Ansys(8,8)	134	
section-properties(0.0001,357,162)	135	
poly44-integral2	136	1.6
poly44-glaubitz(50)	154	1.6
poly44-glaubitz(100)	135	1.6
poly44-rbfcub(50)	136	1.6
poly44-rbfcub(100)	130	1.6
cubicinterp-integral2	136	0.0
cubicinterp-glaubitz(50)	155	0.0
cubicinterp-glaubitz(100)	135	0.0
cubicinterp-rbfcub(50)	137	0.0
cubicinterp-rbfcub(100)	130	0.0
tps-integral2	137	0.0
tps-glaubitz(50)	156	0.0
tps-glaubitz(100)	136	0.0
tps-rbfcub(50)	137	0.0
tps-rbfcub(100)	131	0.0

U-section 100×50×4

The U-section details and the warping function are shown in Fig. 6. $h = 0.1$ m, $b = 0.05$ m, $t = r = 0.004$ m, and as glb-models with sharp and rounded corners.

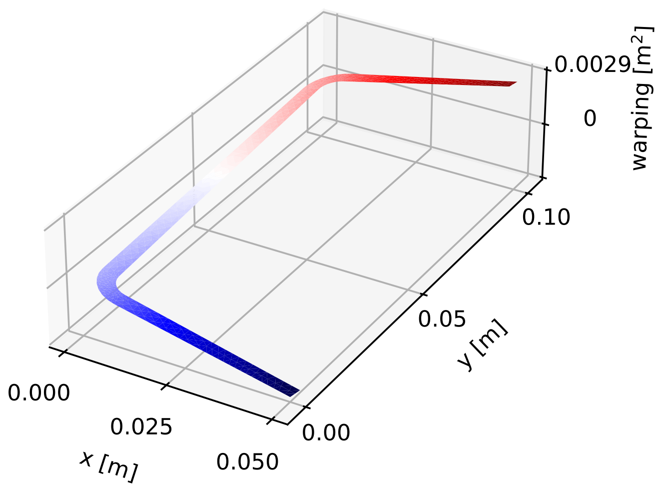


Figure 6. U-section 100×50×4 with rounded corners

Results are presented in Table 3. For the U-section secondary warping has no significant effect on the warping constant. However, taking the rounded corners into account has a notable impact on the warping constant.

Table 2. Warping constant values for solid 100×10 mm rectangle

Source	I_w $[10^{-12} m^6]$	epc [%]
Analytical for thin	6.94	
Analytical(0)	6.68	
Analytical(3)	6.65	
Ansys(1,1)	6.67	
Ansys(5,5)	6.64	
section-properties(0.00001,375,160)	6.64	
poly44-integral2	6.62	0.4
poly44-glaubitz(50)	6.69	0.4
poly44-glaubitz(100)	6.62	0.4
poly44-rbfcub(50)	5.63	0.4
poly44-rbfcub(100)	6.4	0.4
cubicinterp-integral2	6.65	0.0
cubicinterp-glaubitz(50)	6.77	0.0
cubicinterp-glaubitz(100)	6.66	0.0
cubicinterp-rbfcub(50)	5.68	0.0
cubicinterp-rbfcub(100)	6.43	0.0
tps-integral2	6.66	0.0
tps-glaubitz(50)	6.77	0.0
tps-glaubitz(100)	6.67	0.0
tps-rbfcub(50)	5.69	0.0
tps-rbfcub(100)	6.44	0.0

The Ansys lines having multiple numerical values in parentheses in Table 3 are the outer radius of corners, value of SIZE parameter in LESIZE command, i.e., the element edge length, the number of nodes and the number of elements. The Ansys results can be reproduced by using the input file u-150.inp with Ansys Mechanical APDL.

SHS 150×150×8

The SHS-section details and the corresponding warping function are shown in Fig. 7, with $h = 0.15$ m, $b = 0.15$ m, and $t = r = 0.008$ m. 3D models are available with sharp and rounded corners.

Selected results for the sharp-corner model are presented in Table 4, and for the rounded-corner model in Table 5.

For the SHS-section, secondary warping has a significant effect on the warping constant. Using Eq. (19), the estimate for I_w is 40.7×10^{-12}. Rounded corners also influence the warping constant. In this case, `poly44`-based fitting cannot adequately capture secondary warping (the epc values remain high). However, for the rounded-corner model, the value of I_w calculated via `poly44`-based fitting provides a close approximation to the correct result.

Solution times for GBTUL are below one second for small models, and they increase rapidly so that solution time is over minute for a model that uses 9 nodes to describe each corner. Using two nodes to approximate a rounded corner is good option when using GBTUL.

The Ansys values in Table 5 include the outer radius of corners, the value of the SIZE

Table 3. Warping constant values for U-100x50x4 section

Source	I_w $[10^{-12} m^6]$	epc [%]
centerline	297	
GBTUL(1)	299	
Ansys(0/Coarse)	299	
Ansys(8,0.011,99,20)	278	
section-properties(0,0.0001,195,64)	300	
poly44-rbfcub(50)	299	0.5
cubicinterp-rbfcub(50)	299	0.0
tps-rbfcub(50)	297	0.0
GBTUL(2)	263	
GBTUL(6)	277	
section-properties(8,0.0001,8,385,144)	277	
poly44-rbfcub(50)	278	0.7
cubicinterp-rbfcub(50)	278	0.0
tps-rbfcub(50)	275	0.0
tps-rbfcub(100)	278	0.0

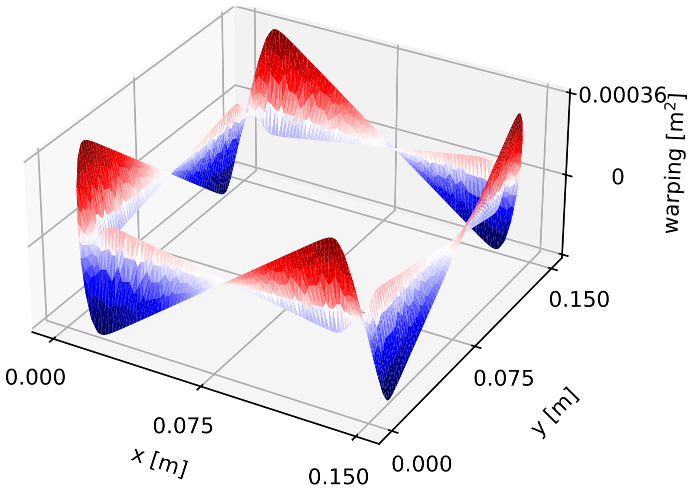


Figure 7. SHS 150×150×8 with rounded corners

parameter in the LESIZE command, i.e., element edge length, the number of nodes and the number of elements. PLANE183, which is a higher order 2D, 8-node element is used for mesh. Solution time for models having less than 1000 nodes is below 100 ms, and for the 14428 node model, it is below 3 seconds. The Ansys results can be reproduced by using shs-150.inp file as input for the Ansys Mechanical APDL.

Section-properties lines in Table 5 have additional number after the mesh size which gives number of points used in the corners. Section shs-n_r in paper_cells.py provides a template to experiment with the parameters that control mesh generation in the *section-properties*. *section-properties* has more options to control mesh generation, but for the SHS-sections, using sufficient points for the corners yields the best results. Laptop solution times varies more between runs, but for small models, it remains less than second. Even for largest models, it is under 3 seconds.

Building interpolation functions using MATLAB's `fit` method for `poly44` or `cubicinterp`

Table 4. Warping constant values for SHS 150×150×8 section using sharp corners

Source	I_w $[10^{-12}m^6]$	epc [%]
centerline	0	
Eq. (19)	40.7	
GBTUL(1)	40.7	
Ansys(0/Coarse)	80.8	
Ansys(5/Fine)	58.4	
section-properties(0,0.0001,264,88)	65.4	
poly44-rbfcub(50)	38.0	78.0
poly44-rbfcub(100)	39.1	78.0
cubicinterp-rbfcub(50)	50.2	0.0
cubicinterp-rbfcub(100)	60.7	0.0
cubicinterp-rbfcub(200)	64.2	0.0
tps-rbfcub(50)	52.1	0.0
tps-rbfcub(100)	62.2	0.0
tps-rbfcub(200)	65.2	0.0
section-properties(0,0.00001,1600,672)	57.0	
poly44-rbfcub(50)	28.5	72.5
cubicinterp-rbfcub(50)	39.1	0.0
cubicinterp-rbfcub(100)	51.6	0.0
cubicinterp-rbfcub(200)	54.8	0.0
tps-rbfcub(50)	38.6	2.4
tps-rbfcub(100)	51.1	2.4
tps-rbfcub(200)	53.7	2.4

takes less than 100 ms for both models. Constructing the `tps` fit takes about 3 seconds for the smaller model and about 15 seconds for the larger model. Evaluation times were measured using MATLAB's `timeit` method for the same set of points that were used to evaluate the cubatures. For `poly44`- and `cubicinterp`-based functions, the interpolations were small (<1 ms for `poly44` and <10 ms for `cubicinterp`), and the number of integration points did not correlate to evaluation times. For `tps`, evaluation time was about 40 ns per node, e.g., evaluation of `tps` having 1716 nodes at 400 points took about 30 ms. The `tps` with 792 nodes at 50 points took about 1.5 ms. The details are available in cell rr after running paper_cells.m.

As pointed out by Björk et al, [3], distortion should be considered when a torsional load is applied to a hollow cross-section. Other ways to handle distortion have already been identified for the targets of this paper (i.e., GBT and ANCF). Since distortion is not in the scope of this paper, a few findings focusing on warping function are highlighted: (i) The warping function for SHS with sharp corners is not symmetrical over the plate thickness as described by Eq. (19). (ii) As noted in in Table 5, the warping constant provided by `poly44` interpolation for model with rounded corners is good when interpolation is based on 302 elements, but gets worse when the number of elements increases to 702. In both cases, the maximum error of interpolation function is over 50 %.

Table 5. Warping constant values for SHS 150×150×8 section using rounded corners

Source	I_w $[10^{-12} m^6]$	epc [%]
GBTUL(2)	58.1	
GBTUL(9)	54.2	
Ansys(16,0.019,192,40)	63.8	
Ansys(16,0.011,304,64)	62.3	
Ansys(16,0.007,720,184)	61.8	
section-properties(16,0.0001,8,792,302)	60.6	
poly44-rbfcub(50)	60.6	51.2
poly44-rbfcub(400)	63.4	51.2
cubicinterp-rbfcub(50)	42.8	0.0
cubicinterp-rbfcub(100)	59.8	0.0
tps-rbfcub(50)	41.9	0.9
tps-rbfcub(100)	60.1	0.9
section-properties(16,0.0001,14,1265,509)	61.4	
section-properties(16,0.0001,24,2490,1076)	61.6	
section-properties(16,0.0001,60,6154,2754)	61.7	
section-properties(16,0.00001,4,1580,644)	55.3	
section-properties(16,0.00001,8,1716,702)	60.6	
poly44-rbfcub(50)	39.0	62.9
poly44-rbfcub(100)	40.9	62.9
cubicinterp-rbfcub(50)	42.7	0.0
cubicinterp-rbfcub(100)	59.9	0.0
tps-rbfcub(50)	41.8	2.8
tps-rbfcub(100)	59.9	2.8
section-properties(16,0.00001,21,3005,1293)	61.6	

Conclusions

This study demonstrates that common interpolation methods are effective at describing warping functions in practical cross-sections. Among the available approaches, radial basis functions (RBF) emerge as the most promising option for warping function interpolation, primarily due to their stability. While the polynomial model (`poly44`) performs well for certain profiles, and cubic interpolation (`cubicinterp`) is computationally more efficient than RBF, the continuity of derivatives at domain boundaries is weaker. For this reason, RBF can be regarded as a robust alternative, although it is advisable to retain all three methods in future investigations and to explore potentially more efficient models than those considered here.

The *section-properties* package demonstrates ease of extension and integration into diverse applications, supporting high-precision results for stiffness-related values derived from interpolation functions. Achieving high precision, particularly for torsional stress values based on differentiation, requires functions with high cardinality. The methods outlined in this study are adaptable to multiple solution techniques, including the Absolute Nodal Coordinate Formulation (ANCF), and they provide a valuable benchmark for testing alternative interpolation functions. These findings underscore the versatility and practical applicability of the methods presented, with the included source code enabling further experimentation and advancement in warping analysis.

In most cases, computation times were sufficiently short to be considered fast (typically below three seconds), approaching the responsiveness now routinely expected from modern AI-based applications. This work establishes a solid foundation for future studies on efficient warping analysis, encouraging the development of improved interpolation strategies and their integration into advanced structural modeling frameworks.

References

[1] Banerjee. Warping functions, 2024. URL: https://en.wikiversity.org/wiki/Warping_functions.

[2] R. Bebiano, D. Camotim, and R. Gonçalves. Gbtul 2.0 - a second-generation code for the gbt-based buckling and vibration analysis of thin-walled members. *Thin-Walled Structures*, 124:235–257, 2018. URL: https://www.sciencedirect.com/science/article/pii/S0263823117310194, doi:10.1016/j.tws.2017.12.002.

[3] Timo Björk, Antti Ahola, and Tuomas Skriko. On the distortion and warping of cantilever beams with hollow section. *Welding in the World*, 64(7):1269–1278, Jul 2020. doi:10.1007/s40194-020-00911-5.

[4] David S. Broomhead and David Lowe. Multivariable functional interpolation and adaptive networks. *Complex Systems*, 2, 1988. URL: https://api.semanticscholar.org/CorpusID:3686496.

[5] M. D. Buhmann. Radial basis functions. *Acta Numerica*, 9:1–38, 2000. doi:10.1017/S0962492900000015.

[6] M. D. (Martin Dietrich) Buhmann. *Radial basis functions: theory and implementations*. Cambridge monographs on applied and computational mathematics; 12. Cambridge University Press, Cambridge, 1st ed. edition, 2003. doi:10.1017/CBO9780511543241.

[7] Roberto Cavoretto, Alessandra De Rossi, Francesco Dell'Accio, Filomena Di Tommaso, Najoua Siar, Alvise Sommariva, and Marco Vianello. Numerical cubature on scattered data by adaptive interpolation. *Journal of Computational and Applied Mathematics*, 444:115793–, 2024. doi:10.1016/j.cam.2024.115793.

[8] Richard Franke. Scattered data interpolation: tests of some methods. *Mathematics of Computation*, 38(157):181–200, 1982. URL: https://www.ams.org/journals/mcom/1982-38-157/S0025-5718-1982-0637296-4/S0025-5718-1982-0637296-4.pdf.

[9] T. Haukaas. Warping torsion, 2023. URL: https://civil-terje.sites.olt.ubc.ca/files/2023/09/5.-Warping-torsion.pdf.

[10] Curt F. Kollbrunner and Nikola Hajdin. *Stäbe mit undeformierbaren Querschnitten*. Springer, 1972. doi:10.1007/978-3-662-00421-0.

[11] Robbie Leeuwen and Connor Ferster. sectionproperties: A python package for the analysis of arbitrary cross-sections using the finite element method. *Journal of Open Source Software*, 9(96):6105, 2024. doi:10.21105/joss.06105.

[12] M. Lezgy-Nazargah, P. Vidal, and O. Polit. A penalty-based multifiber finite element model for coupled bending and torsional-warping analysis of composite beams. *European Journal of Mechanics - A/Solids*, 80:103915, 2020. URL: https://www.sciencedirect.com/science/article/pii/S0997753819305595, doi:10.1016/j.euromechsol.2019.103915.

[13] M. Lezgy-Nazargah, P. Vidal, and O. Polit. A quasi-3d finite element model for the analysis of thin-walled beams under axial-flexural-torsional loads. *Thin-Walled Structures*, 164:107811, 2021. URL: https://www.sciencedirect.com/science/article/pii/S026382312100238X, doi:10.1016/j.tws.2021.107811.

[14] David Manta and Rodrigo Gonçalves. A geometrically exact Kirchhoff beam model including torsion warping. *Computers & Structures*, 177:192–203, 2016. doi:10.1016/j.compstruc.2016.08.013.

[15] Simo Nikula. Section-properties examples, 2025. URL: https://github.com/simo-11/spe/.

[16] W. D. Pilkey. *Analysis and Design of Elastic Beams*. John Wiley & Sons, Ltd, New York, USA, 2002. doi:10.1002/9780470172667.

[17] M. J. D. Powell. Restart procedures for the conjugate gradient method. *Mathematical Programming*, pages 241–254, 1977. doi:10.1007/BF01593790.

[18] R Schardt. *Verallgemeinerte Technische Biegetheore*. Springer Berlin, Heidelberg, 1989. doi:10.1007/978-3-642-52330-4.

[19] R. Schardt. Generalized beam theory–an adequate method for coupled stability problems. *Thin-Walled Structures*, 19(2):161–180, 1994. URL: https://www.sciencedirect.com/science/article/pii/0263823194900272, doi:10.1016/0263-8231(94)90027-2.

[20] Zhenxing Shen. Thin-walled composite beam elements via the absolute nodal coordinate formulation. *Multibody System Dynamics*, 62(1):107–135, 2024. doi:10.1007/s11044-023-09956-y.

[21] Alvise Sommariva and Marco Vianello. Meshless cubature by Green's formula. *Applied Mathematics and Computation*, 183(2):1098–1107, 2006. doi:10.1016/j.amc.2006.05.211.

[22] Alvise Sommariva and Marco Vianello. Numerical cubature on scattered data by radial basis functions. *Computing*, 76(3-4):295–310, 2006. doi:10.1007/s00607-005-0142-2.

[23] Alvise Sommariva and Marco Vianello. Rbf moment computation and meshless cubature on general polygonal regions. *Applied Mathematics and Computation*, 409:126375–, 2021. doi:10.1016/j.amc.2021.126375.

[24] S. Stoykov, E. Manoach, and S. Margenov. An efficient 3d numerical beam model based on cross sectional analysis and Ritz approximations. *ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift für Angewandte Mathematik und Mechanik*, 96(7):791–812, 2016. URL: https://onlinelibrary.wiley.com/doi/

abs/10.1002/zamm.201400139, arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/zamm.201400139, doi:10.1002/zamm.201400139.

[25] Yixuan Tang, Qiang Tian, and Hauyan Hu. Efficient modeling and order reduction of new 3d beam elements with warping via absolute nodal coordinate formulation. *Nonlinear Dynamics*, 109:2319–2354, 2022. doi:10.1007/s11071-022-07547-8.

[26] S. Timoshenko and J. N. Goodier. *Theory of Elasticity.* McGraw-Hill, New York, USA, 2nd ed. edition, 1951.

[27] V.Z. Vlasov. *Thin Walled Elastic Beams.* Israel Program for Scientific Translations, 1961.

Simo Nikula, Aki Mikkola, Timo Björk, Marko K. Matikainen
LUT Mechanical Engineering, School of Energy Systems
LUT University