XL CILAMCE
IBERO-LATIN AMERICAN
CONGRESS ON
COMPUTATIONAL
METHODS IN
ENGINEERING

NOVEMBER
11-14, 2019
Praiamar Natal Hotel & Convention
Natal, RN-BRAZIL

VIRTUAL ELEMENT METHOD VS FINITE ELEMENT METHOD FOR PRANDTL’S SOLUTION OF ST. VENANT TORSION PROBLEM

Tiago Fernandes Moherdaui
Alfredo Gay Neto
tiago.moherdaui@usp.br
alfredo.gay@usp.br
Escola Politécnica da Universidade de São Paulo
Av. Prof. Almeida Prado, 83, 05508-070, São Paulo, Brazil

Abstract. This work presents a comparison of the novel virtual element method and the traditional finite element method for Prandtl’s solution to the St. Venant torsion problem. The solved field is the Prandtl function for a given cross-section that experiences torsion. This function is approximated using both methods with a collection of different meshes. These meshes vary in properties such as the element size, geometry (Delaunay triangulations and Voronoi tessellations are employed), and polynomial order (linear and quadratic elements). The numerical error measurement is based on the torsion constant, a global scalar associated with the solution, with physical significance for the problem. Two different cross-sections with analytical solutions are used as benchmarks. The results are presented in a set of convergence curves for each cross-section geometry. The virtual element method showed more versatility regarding element geometry, while retaining the same convergence properties of the finite element method.

Keywords: Virtual element method, Finite element method.

1 Introduction

Understanding torsion is important for structural engineering. There are direct applications such as in torque-transmission elements, e.g. axles connecting rotors and turbines. While in other applications torsion emerges as consequence of eccentrically applied loads; this is usual in steel-framed structures due to the demands of practical construction.

According to Vlasov's torsion theory [1], the structural elements response to the torsional load comprise two components: uniform torsion and warping torsion. The former is obtained considering that the cross-section warps freely; and the latter concerns the effects of warping restriction. The relative relevance of each component depends on the cross-section geometry.

This paper focuses on the uniform torsion component, which is obtained via St. Venant torsion theory. The full characterization of the stresses and the torsion constant (I_T) are obtained by solving a differential equation for the warping function (ψ) or, alternatively, Prandtl's function (ϕ). For this paper the latter is adopted.

The virtual element method (VEM) and the finite element method (FEM) are used to solve for Prandtl's function. Two cross-section geometries are considered. The results are compared to the analytical results provided by Timoshenko [2].

The VEM for solving partial differential equations was introduced by Beirão da Veiga, et al. [3] as a reinterpretation of the mimetic finite difference method through the finite element paradigm. This method also constitutes a generalization of the finite element method which permits more geometrically versatile elements while retaining convergence properties. Further works by several authors explore the method itself [4]–[10]; show its applicability for elliptic, parabolic and hyperbolic differential equations [11]–[13]; and explore a posteriori error estimation [14]–[17]. Other works present the method in a more practical way, focusing on its computational implementation, such as Sutton [18] and Beirão da Veiga, et al. [19].

The paper is structured as follows: Section 2 states the mathematical model for St. Venant torsion and Prandtl's solution. Section 3 contains the methodology, including a brief explanation for both methods and the parameters considered in the numerical study. Section 4 shows the results of the latter followed by section 5 with the conclusions.

2 St. Venant torsion theory

The first known theory regarding the torsion concept, according to Timoshenko [2], is from Coulomb. It explains the uniform torsion for rods of circular cross-section; failing, however, to explain the warping that occurs for non-circular geometries. The following theory was developed by St. Venant. This theory takes warping into account, and assumes the cross-section is free to warp. Later theories, such as Vlasov's torsion theory, take into account the effects of warping restriction and non-uniform torsion. While all the aforementioned theories are based on linear elasticity, there are others which explore geometrical nonlinearity (Trahair [20]) and inelasticity (Sapountzakis and Tsipiras [21]).

The theory presented by St. Venant is developed using the semi-inverse method, in which a proposed displacement field is shown to satisfy, under a set of hypotheses, the balance and boundary equations, constituting a solution. It is formulated based on a prismatic rod considering a generic cross-section made of material with shear modulus G, subject only to pure twisting moment (M_T) as in Figure 1. The z-axis is parallel to that of the rod's length and goes through its twist center. The set of hypotheses is presented as follows:

I. Each individual cross-section (z = constant) rotates as a rigid body;
II. The rate of twist $d\theta/dz = \theta'$ is constant (uniform torsion);
III. The cross-sections are free to warp. However, the warping is the same for all of them;
IV. Small rotations ($\theta \approx 0$);
V. Material linearity ($\sigma_{xz}=G\gamma_{xz}$ and $\sigma_{yz}=G\gamma_{yz}$).

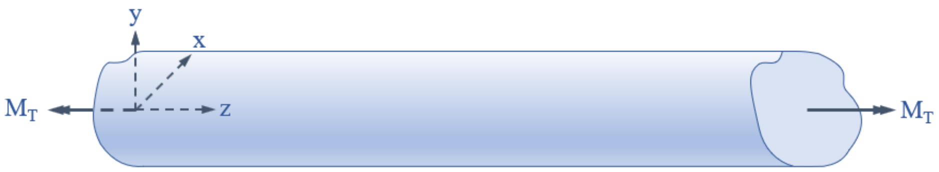

Figure 1. Coordinate system and twisting moment.

The warping of the cross-section is defined as the warping function ψ(x,y) scaled by the twist rate θ'. Then, from I and IV, the proposed displacement field is as presented in Eq. (1).

$$\begin{cases} u = -\theta' yz; \\ v = \ +\theta' xz; \\ w = \theta' \psi. \end{cases} \tag{1}$$

The strain field is obtained via differentiation; its only non-zero terms are γ_{xz} and γ_{yz}. From V, the tangential stresses are obtained, Eq. (2a). The balance equations lead to only one non-trivial equation, Eq. (2b). With some manipulations Eq. (2c) is obtained, being valid for the whole domain.

$$\begin{cases} \sigma_{xz} = G\theta'(\psi_{,x} - y); \\ \sigma_{yz} = G\theta'(\psi_{,y} + x); \end{cases} \tag{2a}$$

$$\frac{\partial \sigma_{xz}}{\partial x} + \frac{\partial \sigma_{yz}}{\partial y} = 0; \tag{2b}$$

$$G\theta'\left(\psi_{,x} + \psi_{,y}\right) = 0 \therefore \Delta\psi = 0. \tag{2c}$$

There being no load other than the twisting moment, the Neumann boundary condition (equilibrium on the boundary) is expressed by Eq. (3) on the whole boundary. For a given point on the boundary, **n** is the unit outward pointing normal vector; **r** is the position vector; and **τ** is the unit tangent vector of the boundary, as illustrated in Figure 2.

$$\mathbf{Tn} = \mathbf{0} \therefore \nabla\psi \cdot \mathbf{n} = \mathbf{r} \cdot \boldsymbol{\tau}. \tag{3}$$

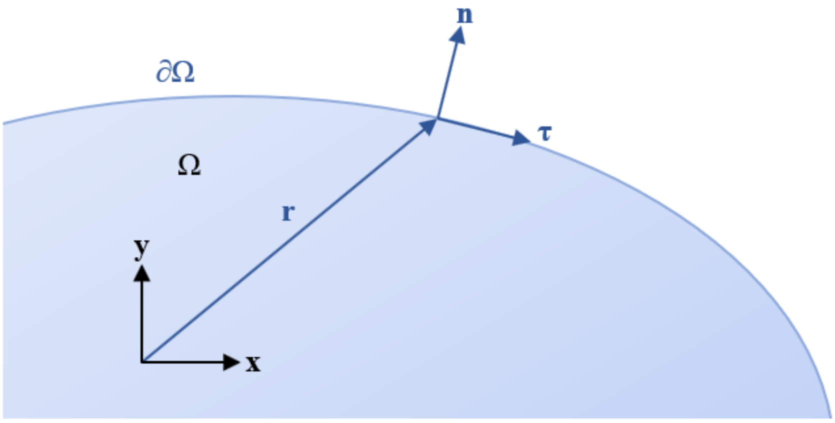

Figure 2. Position (r), normal (n) and tangent (τ) vectors.

The problem is, therefore, fully characterized once the warping function ψ is known; Its formulation is summarized by Eq. (4).

$$\begin{cases} \Delta\psi = 0 \text{ in } \Omega; \\ \nabla\psi \cdot \mathbf{n} = \mathbf{r} \cdot \boldsymbol{\tau} \text{ in } \partial\Omega. \end{cases} \tag{4}$$

2.1 Prandtl's solution

Prandtl presented an alternative formulation for this problem, which leads to simpler boundary conditions. This version is based on a new function $\phi(x,y)$, called Prandtl's function, which Eq. (5) defines.

$$\begin{cases} \phi_{,x} = -\sigma_{yz}; \\ \phi_{,y} = \sigma_{xz}. \end{cases} \tag{5}$$

The balance equation for this function, Eq. (2b), is inherently satisfied because of the Clairaut-Schwartz theorem, also known as the equality of mixed partials. Another equation is obtained by taking the Laplacian of ϕ and expressing it in terms of the warping function, Eq. (6a). The Clairaut Schwartz theorem leads to Eq. (6b) being valid for the whole domain.

$$\Delta\phi = \phi_{,xx} + \phi_{,yy} = G\theta'\left(\psi_{,yx} - \psi_{,xy} - 2\right); \tag{6a}$$

$$\Delta\phi = -2G\theta'. \tag{6b}$$

The Neumann boundary condition, Eq. (4), can be expressed in terms of Prandtl's function, Eq. (7). This last equation can be interpreted as the function being constant along the boundary; The arbitrary constant 0 is adopted. This degenerates the Neumann boundary condition into a homogeneous Dirichlet boundary condition.

$$\nabla\phi \cdot \boldsymbol{\tau} = \frac{\partial\phi}{\partial\boldsymbol{\tau}} = 0. \tag{7}$$

The torsion problem is also characterizable in terms of Prandtl's function. Its formulation in terms of Prandtl's function is Eq. (8).

$$\begin{cases} \Delta\phi = -2G\theta' \text{ in } \Omega; \\ \phi = 0 \text{ in } \partial\Omega. \end{cases} \tag{8}$$

2.2 Weak formulation

The numerical methods around which this work revolves are designed to obtain weak solutions for partial differential equations. The weak formulation, for the differential form in Eq. (8), is obtained by integrating the main equation, weighted by a test function $\delta\phi$, over the domain. Using Green's first identity and introducing the boundary condition results in the weak form, in Eq. (9).

$$\int_\Omega \nabla\phi \cdot \nabla\delta\phi \, d\Omega = \int_\Omega -2G\theta'\delta\phi \, d\Omega. \tag{9}$$

In this form, ϕ requires integrable first derivatives, i.e. $\phi \ \epsilon\ \mathscr{H}^{1}{}_{0}$ (space of square integrable functions whose first derivatives are also square integrable and satisfy the Dirichlet boundary condition). In the differential form, presented in Eq. (8), it required a constant Laplacian inside the domain, i.e. $\phi \ \epsilon\ C^2$. This drop in requirement characterizes the weak form. The weak version of the problem is expressed in Eq. (10), where a is a bilinear form given by the left-hand side of Eq. (9) and f is a linear form given by the right-hand side.

$$\text{Find } \phi \in \mathcal{H}_0^1 \text{ such that for all } \delta\phi \in \mathcal{H}_0^1\text{:}$$

$$a(\phi, \delta\phi) = f(\delta\phi). \tag{10}$$

3 Methodology

Any comparison requires established parameters and cases. For this case, the two must take into consideration the characteristics of both the compared methods and the problem. The solution to the problem in Prandtl's formulation is given by a function associated with a stress field.

Pointwise values of the function (or the stresses) do not serve as good basis for comparison, because both methods make approximations based on an input meshes. Therefore, in order to compare pointwise results adequately the meshes should coincide. As one of VEM's advantages over FEM is its element geometry versatility; comparing both methods for the same mesh would be unsatisfactory.

This section presents the framework used to compare both methods, and is organized as follows: The torsion constant is presented as a parameter to globally evaluate the result in subsection 3.1. Subsections 3.2 and 3.3 comprise a brief presentation of both numerical methods. The cases are presented in the following subsections. The two different cross-section geometries in subsection 3.4, followed by the mesh parameters varied throughout the analyses (element geometry and characteristic size) in subsection 3.5.

3.1 Torsion constant

In the torsion theory, the relationship between the rate of twist (θ') and the twisting moment (M_T) can be written as in Eq. (11).

$$\frac{M_T}{GI_T} = \theta'. \tag{11}$$

The twisting moment is expressed in terms of the stress field in Eq. (12).

$$M_T = \int_\Omega (\sigma_{yz}x - \sigma_{xz}y)d\Omega. \tag{12}$$

Substituting the definition of Prandtl's function, Eq. (5), and integrating by parts results in a simplified expression for the moment, Eq. (13).

$$M_T = 2\int_\Omega \phi d\Omega. \tag{13}$$

From Eq. (11) and (13), the torsion constant is formulated in Eq. (14).

$$I_T = \frac{2\int_\Omega \phi d\Omega}{G\theta'}. \tag{14}$$

This constant constitutes a parameter which has physical relevance to the problem and evaluates the solution globally. Additionally, as both problems described in Eq. (4) and Eq. (8) are classic problems in the field of partial differential equations (namely Laplace and Poisson problems, respectively), there are analytical solutions for simple cross-section geometries. Consequently, there are also solutions for the torsion constant to which the numerical results can be compared. These are presented in subsection 3.4 alongside the chosen cross-sections.

3.2 Finite element method

The finite element method is a numerical method for obtaining approximate weak solutions for systems of partial differential equations. The method works, in present application, by projecting the real solution, resident of the infinite dimensional space C^2, into a piecewise polynomial space of finite dimension. This piecewise polynomial space is composed by the polynomial spaces associated with each subdomain; therefore, it is directly associated with the partitioning of the problem's domain Ω into a collection (mesh) of non-overlapping subdomains Ω_E (elements).

According to Ciarlet [22], each element can be defined as a subdomain Ω_E (usually triangles for 2D domains) associated with a set of nodes and a polynomial space which contains the full polynomial space up to degree k (P_k). The nodes must be chosen in a way that the set of nodal values determines uniquely a member of that space. The shape functions are the canonical basis for that space using the nodal values as degrees of freedom. Note that the dimension of the polynomial space and the number of degrees of freedom are connected.

Globally, the union of all the finite elements spaces becomes the piecewise polynomial function space in Ω (subspace of $\mathscr{H}^1$), onto which the solution is projected in order to obtain the weak solution. The weak solution, $u_h(x,y)$, is associated with the set of nodal values for which the functions defined in the elements, together, construct the best approximation in that subspace, based on an input mesh.

Inside each element (Ω_E), u_h can be written as Eq. (15), where **h**(x,y) is the row vector of shape functions and $\mathbf{u}$ is the column vector of nodal values for that element.

$$\mathrm{u_h}(\mathrm{x},\mathrm{y}) = \mathbf{h}(\mathrm{x},\mathrm{y})\mathbf{u}. \tag{15}$$

The bilinear and linear forms presented in Eq. (9) and Eq. (10) can be assembled from their element counterparts. The approximation presented in Eq. (15) can be introduced in both forms, which become the element stiffness matrix ($\mathbf{K}^{\mathrm{E}}$) and load vector ($\mathbf{f}^{\mathrm{E}}$), respectively Eq. (16a) and Eq. (16b).

$$\int_{\Omega_E} \nabla \mathbf{h}^T \nabla \mathbf{h}\, d\Omega\, \mathbf{u} = -2\mathrm{G}\theta' \int_{\Omega_E} \mathbf{h}^T\, d\Omega; \tag{16a}$$

$$\mathbf{K}^{\mathrm{E}}\mathbf{u} = \mathbf{f}^{\mathrm{E}}. \tag{16b}$$

The global system is obtained by assembling the global stiffness matrix and the load vector from their element components. Solving the global system provides the nodal values to construct the approximated solution.

3.3 Virtual element method

The virtual element method can be seen as a generalization of the finite element method. The main difference is the way the function spaces are designed. Instead of restricting the space to polynomials, the virtual element space contains the full polynomial space associated with the element order plus other additional functions, called virtual functions. The choice for the degrees of freedom is made systematically and not restricted to nodal values.

The difficulty in using general shape elements with the FEM is to find the shape functions which form a basis for the polynomial space. For the VEM, the larger function space allows the dissociation between the number of nodes and the dimension of the polynomial space, using the virtual functions as a buffer. For handling these unknown virtual functions, the method splits the functions into their projection into the polynomial space and the residual of this projection. This allows the method to retain the same convergence properties as the finite element method for the same polynomial space. The degrees of freedom are carefully chosen so that the virtual functions need not be known explicitly to obtain the stiffness matrix and load vector. This result is what gives the method its name and elegance. The additional virtual functions along with the systematical choice of degrees of freedom allow arbitrary degree polynomial interpolation in arbitrarily shaped elements.

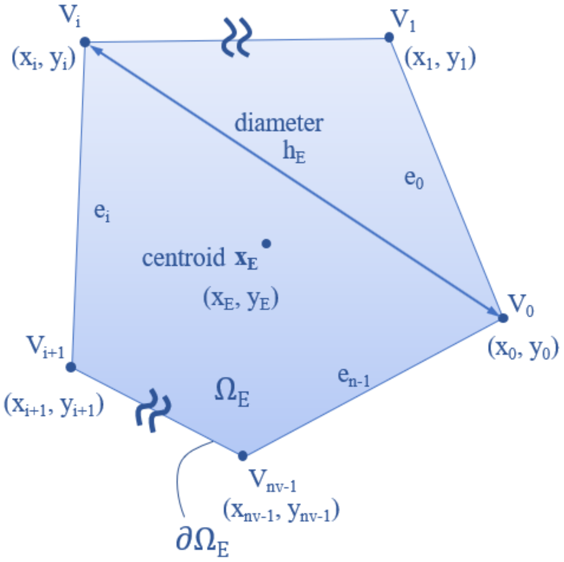

Figure 3. Nomenclature for a generic polygon (Ω_E).

The virtual element formulation requires some notation to be introduced. Consider the domain (Ω), partitioned into a collection of non-overlapping polygons (Ω_E). Each polygon has n_v vertices and edges.

The vertices are denoted V_i, with index i starting at 0 and ending at n_v-1 incrementing counterclockwise. The union of the edges forms the boundary of the polygon ($\partial\Omega_E$). The edge referred to as e_i is the one that connects V_i to V_{i+1}, as shown in Figure 3.

The polygon area is denoted by $|\Omega_E|$ and is synonymous to the subdomain measure. The largest distance between two points inside the polygon is called its diameter and denoted h_E. For convex polygons, this is the largest distance between two vertices.

The function space associated with each element is called the local virtual element space $V_k(\Omega_E)$. By definition, this space contains the space $P_k(\Omega_E)$, the full space of polynomials up to degree k (of dimension n_k), plus the additional virtual functions. The local virtual element space, according to Beirão da Veiga, et al. [3], is formed by all functions v_h which satisfy the following properties:

$$\begin{cases} \text{i) } \mathrm{v_h} \text{ is a polynomial of degree k on each edge e of } \Omega_\mathrm{E} \rightarrow \mathrm{v_h}|_\mathrm{e} \in P_\mathrm{k}(\Omega_\mathrm{E}); \\ \text{ii) } \mathrm{v_h} \text{ on } \partial\mathrm{E} \text{ is globally continuous} \rightarrow \mathrm{v_h}|_\mathrm{e} \in \mathrm{C}^0(\partial\Omega_\mathrm{E}); \\ \text{iii) } \Delta\mathrm{v_h} \text{ is a polynomial of degree } \mathrm{k}-2 \text{ in } \Omega_\mathrm{E} \rightarrow \Delta\mathrm{v_h} \in P_{\mathrm{k}-2}(\Omega_\mathrm{E}). \end{cases}$$

The definition of the degrees of freedom for the virtual element, as presented in Beirão da Veiga, et al. [19], uses a basis (M_k) for the space P_k, called the scaled monomials. Their definition is presented using multi-index notation, where each monomial is associated with a coordinate pair $\boldsymbol{\alpha} = (\alpha_1, \alpha_2)$ as stated in Eq. (18). Each coordinate pair can also be mapped to a one-dimensional index α as shown in Eq. (17).

$$\alpha = 1 \leftrightarrow (0,0),\ \alpha = 2 \leftrightarrow (1,0),\ \alpha = 3 \leftrightarrow (0,1),\ \alpha = 4 \leftrightarrow (2,0),\ \alpha = 5 \leftrightarrow (1,1), \ldots \tag{17}$$

$$m_\alpha = \left(\frac{x - x_E}{h_E}\right)^{\boldsymbol{\alpha}} = \left(\frac{x - x_E}{h_E}\right)^{\alpha_1}\left(\frac{y - y_E}{h_E}\right)^{\alpha_2}. \tag{18}$$

The degrees of freedom of $V_k(\Omega_E)$ are defined and ordered as follows:

$$\begin{cases} \blacksquare \text{ the value of } \mathrm{v_h} \text{ at the } \mathrm{n}_v \text{ vertices of } \Omega_\mathrm{E}; \\ \blacksquare \text{ the value of } \mathrm{v_h} \text{ at the } \mathrm{k}-1 \text{ internal points of the} \\ \quad (\mathrm{k}+1) - \text{point Gauss} - \text{Lobatto rule on each edge e;} \\ \blacksquare \text{ the moments up to order } \mathrm{k}-2 \text{ of } \mathrm{v_h} \text{ in } \Omega_\mathrm{E}: \\ \qquad \dfrac{1}{|\Omega_\mathrm{E}|}\displaystyle\int_{\Omega_\mathrm{E}} \mathrm{v_h m_\alpha}, \qquad \alpha = 1, 2, \ldots, dim(P_{k-2}). \end{cases}$$

The canonical basis for the local virtual element space is the set of functions $\{\varphi_i\}$, the index going from 0 to the n_{dof}-1, where n_{dof} is the dimension of the space given by the expression in Eq. (19).

$$\dim \mathrm{V_k}(\Omega_\mathrm{E}) = \mathrm{n_v k} + \frac{(\mathrm{k}-1)\mathrm{k}}{2}. \tag{19}$$

Given the basis for V_k, the solution can be approximated inside the element in the same way as shown in Eq. (15). However, these functions need to be split into their projection onto $P_k(\Omega_E)$ and the residue, as in Eq. (20), where **I** is the (n_{dof} x n_{dof}) identity matrix.

$$\varphi_\mathrm{i} = \boldsymbol{\Pi}^{\boldsymbol{\nabla}}\varphi_\mathrm{i} + (\mathbf{I} - \boldsymbol{\Pi}^{\boldsymbol{\nabla}})\varphi_\mathrm{i}. \tag{20}$$

This is done with a projection operator ($\boldsymbol{\Pi}^{\boldsymbol{\nabla}}$), which is the matrix representation of the projection stated in Eq. (21a) and defined by Eq. (21b).

$$\Pi^{\nabla}{}_{\mathrm{E},k}: V_k(\Omega_E) \rightarrow P_k(\Omega_E); \tag{21a}$$

$$\int_{\Omega_\mathrm{E}} \nabla \mathrm{m}_\alpha \nabla((\mathbf{I} - \boldsymbol{\Pi}^{\boldsymbol{\nabla}})\mathrm{v_h}) = 0,\ \alpha = 1, \ldots, \dim(P_k). \tag{21b}$$

Beirão da Veiga, et al. [19] present a more pragmatic approach to compute the projection operator as shown in Eq. (22), by using intermediary matrices **B** (n_k x n_{dof}) and **D** (n_{dof} x n_k). These matrices are defined in Eq. (23a) and Eq. (23b), respectively, where $(u, v)_{0,\Omega E}$ is the $L^2(\Omega_E)$ internal product (the integral of the product of u and v over Ω_E); and $dof_i(v_h)$ is the ith degree of freedom of v_h, as defined previously.

$$\mathbf{\Pi}^{\boldsymbol{\nabla}} = \mathbf{D}(\mathbf{BD})^{-\mathbf{1}}\mathbf{B}. \tag{22}$$

$$\mathbf{B} = \begin{bmatrix} P_0\varphi_0 & \cdots & P_0\varphi_{\mathrm{n_{dof}}-1} \\ (\nabla m_2, \nabla\varphi_0)_{0,\Omega_E} & \cdots & \left(\nabla m_2, \nabla\varphi_{\mathrm{n_{dof}}-1}\right)_{0,\Omega_E} \\ \vdots & \ddots & \vdots \\ \left(\nabla m_{\mathrm{n_k}}, \nabla\varphi_0\right)_{0,\Omega_E} & \cdots & \left(\nabla m_{\mathrm{n_k}}, \nabla\varphi_{\mathrm{n_{dof}}-1}\right)_{0,\Omega_E} \end{bmatrix}; \tag{23a}$$

$$D_{i,\alpha} = dof_i(m_\alpha). \tag{23b}$$

The first row of **B** is defined differently; this is done to circumvent the kernel of the projection operator by using another projection operator P_0 as presented in Eq. (24a), and Eq. (24b). This kernel exists because the projection operation is based on the internal product of the gradients of the two functions, therefore the projection would only be determined up to a constant.

$$\mathrm{P}_0 \colon V_k(\Omega_E) \to P_0(\Omega_E); \tag{24a}$$

$$\begin{cases} \mathrm{P}_0 \mathrm{v}_h := \frac{1}{n_v}\sum_{i=0}^{n_v-1} \mathrm{v}_h(\mathrm{V_i}),\ k = 1; \\ \mathrm{P}_0 \mathrm{v}_h := \frac{1}{|\Omega_E|}\int_{\Omega_E} \mathrm{v}_h \, d\Omega,\ k \geq 2. \end{cases} \tag{24b}$$

Further explanations on the inner workings of virtual element formulation and guidance for the computational implementation can be found in Beirão da Veiga, et al. [19] and Sutton [18].

The split of each basis function can be extended to the shape function row vector **h**, although this vector is already implied in the matrix formulation of the projection operator. The stiffness matrix for a virtual element is expressed in Eq. (25); it is obtained by introducing the split into the bilinear form, while remembering the orthogonality between the projection and residue. The term of the product involving the residues can be substituted by a stabilization term $\mathbf{S^E}$ shown in Eq. (26).

$$\mathbf{K^E} = \int_{\Omega_E} \nabla[\mathbf{\Pi}^{\boldsymbol{\nabla}}]^T \nabla[\mathbf{\Pi}^{\boldsymbol{\nabla}}] d\Omega + \int_{\Omega_E} \nabla[(\mathbf{I} - \mathbf{\Pi}^{\boldsymbol{\nabla}})]^T \nabla[(\mathbf{I} - \mathbf{\Pi}^{\boldsymbol{\nabla}})] d\Omega. \tag{25}$$

$$\int_{\Omega_E} \nabla[(\mathbf{I} - \mathbf{\Pi}^{\boldsymbol{\nabla}})]^T \nabla[(\mathbf{I} - \mathbf{\Pi}^{\boldsymbol{\nabla}})] d\Omega \approx \mathbf{S^E} = (\mathbf{I} - \mathbf{\Pi}^{\boldsymbol{\nabla}})^T \cdot (\mathbf{I} - \mathbf{\Pi}^{\boldsymbol{\nabla}}). \tag{26}$$

The expression for the virtual element stiffness matrix is summarized in Eq. (27a) and Eq. (27b). A more practical expression for calculating the first term is presented in the aforementioned works.

$$\mathbf{K^E} = \mathbf{K_C^E} + \mathbf{S^E}; \tag{27a}$$

$$\mathbf{K^E} = \int_{\Omega_E} \nabla[\mathbf{\Pi}^{\boldsymbol{\nabla}}]^T \nabla[\mathbf{\Pi}^{\boldsymbol{\nabla}}] d\Omega + (\mathbf{I} - \mathbf{\Pi}^{\boldsymbol{\nabla}})^T \cdot (\mathbf{I} - \mathbf{\Pi}^{\boldsymbol{\nabla}}). \tag{27b}$$

The first term in Eq. (27b) is called the consistency matrix; it guarantees exactness when the solution is polynomial and is important for the convergence properties. There are other formulations for the stabilization term, which work better for different problems. Hudobivnik, et al. [23] presented a comparison of three stabilization terms for linear elasticity.

The element counterpart of load vector can be approximated using the same projector for k ≤ 2, as shown in Eq. (28). For other element orders another projector must be used, for this situation one should refer to the literature.

$$(\mathbf{f^E})_\mathbf{i} = -2\mathrm{G}\theta' \int_{\Omega_E} \Pi^{\nabla} \varphi_\mathrm{i} \mathrm{d}\Omega. \tag{28}$$

The assemblage of the element matrices and vectors to obtain the global system is done in the same way as for the finite element method.

3.4 Numerical examples

Two different cross-section geometries were selected for the comparisons: a square and an equilateral triangle. As elementary geometric shapes, both have analytical solutions easily found in the literature. The dimensions were chosen for the unitary area of both shapes, as shown in Figure 4. The analytical solutions for the square and triangle cross-section are presented in Eq. (29) and Eq. (30) respectively.

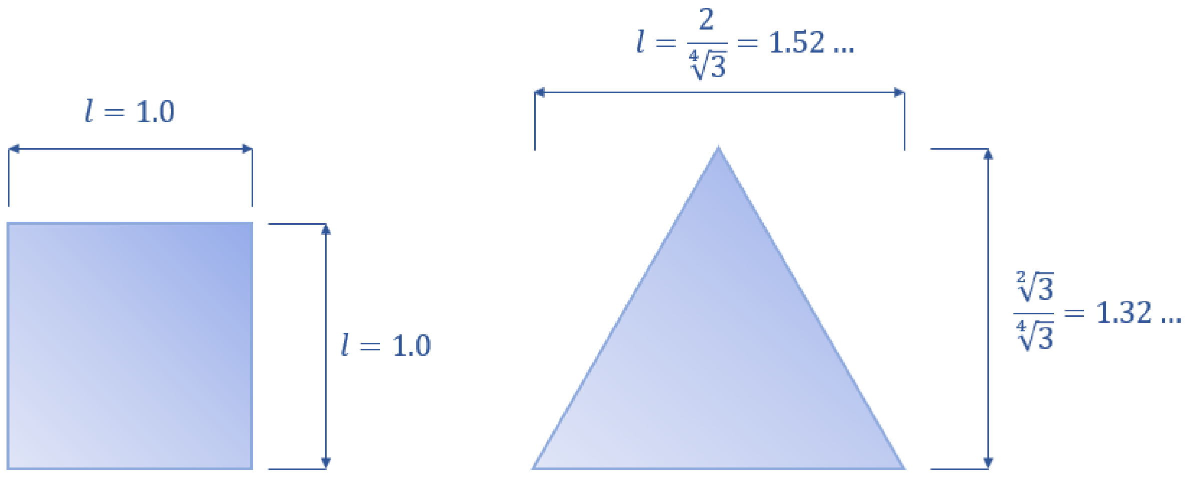
$l = 1.0$
$l = \frac{2}{\sqrt[4]{3}} = 1.52\ldots$
$l = 1.0$
$\frac{\sqrt[2]{3}}{\sqrt[4]{3}} = 1.32\ldots$

Figure 4. Dimensions of the cross-sections.

$$\mathrm{I}_{\mathrm{T}}^{\square} = \mathrm{l}^4 \left[\frac{1}{3} - \frac{64}{\pi^5} \sum_{n=0}^{\infty} \frac{\tanh\left(\frac{(2n+1)\pi}{2}\right)}{(2n+1)^5}\right] = 0.140577\ldots. \tag{29}$$

$$\mathrm{I}_{\mathrm{T}}^{\Delta} = \frac{\mathrm{l}^4\sqrt{3}}{80} = 0.127323\ldots. \tag{30}$$

3.5 Meshes

The two methods under comparison work based on the input of a domain partition (mesh). The finite element method for two-dimensional problems is typically used with triangular meshes, which are easily generated using existing software. However, one of the great advantages of the virtual element method is its versatility regarding the element geometry.

In order to explore this as a possible parameter for the analysis, a routine was implemented to generate Voronoi tessellations based on the triangular meshes. The tessellation is obtained by connecting the centers of the triangles; as is illustrated in Figure 5. When all the centers of all the triangles that share a node are connected, they form a polygon around that node.

Some adaptations were needed to handle the boundary and to preserve the shape of the original triangulations. The triangulation and Voronoi tessellation are dual graphs in the Graph theory, from which came the intuition behind the transformation. Additional routines were developed to handle varying size geometry for elements and to convert linear polygonal meshes to a second-or-higher order meshes.

In this work, the triangular meshes (Delaunay triangulations) were generated and the post-processing was conducted by using the software GMSH by Geuzaine and Remacle [24].

There are certain parameters that can be used to globally characterize a mesh. One example is the maximum element diameter (h). This parameter, as shown in Ciarlet [22], is exponentially related to the superior a priori error estimations. This relationship is shown in convergence curves, which plot an error estimate against the maximum element diameter. These are usually plotted with both axes logarithmically scaled so that the curves appear as lines. This parameter is varied by generating meshes with 4 different levels of refinement, resulting in four different points in the convergence curve for each combination of element geometry and method.

Another important parameter is the number of degrees of freedom (n). This is a direct measure of the size of the system that needs to be solved for obtaining the solution. It is correlated to the computational cost of solving the problem.

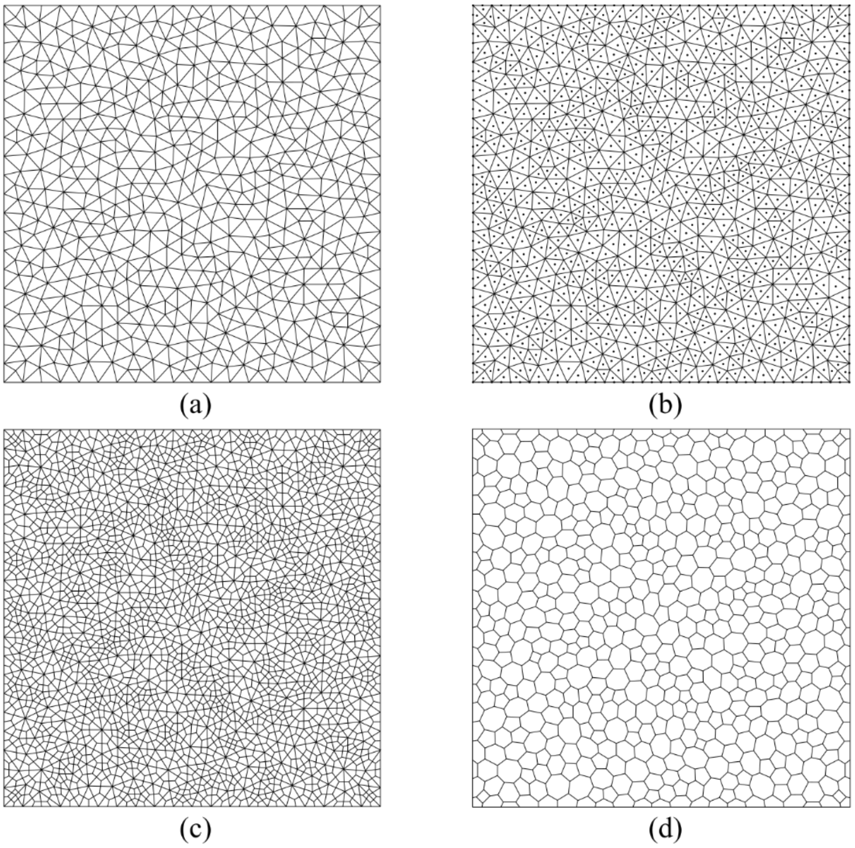

Figure 5. Generation of a Voronoi tessellation (d) from a Delaunay triangulation (a). Elect nodes on the barycenter of each triangle (b) and associate with each node a polygon connecting the centers of all the triangles sharing the node (c).

All the varying parameters in this study effectively reflect different meshes. These parameters are the method (Finite elements or Virtual elements), cross-section geometry (Square or Triangle), the element geometry (Delaunay triangulation or Voronoi tessellation), order of the element (1st or 2nd order), element size (4 different discretizations for each geometry, code 1 corresponding to the coarsest mesh and 4 to the finest, as exemplified in Figure 6). The virtual element method is applied to both the triangular and polygonal element meshes. The set of meshes generated for this study is organized on a basis which incorporates these parameters and is coded following the nomenclature summarized in Figure 7.

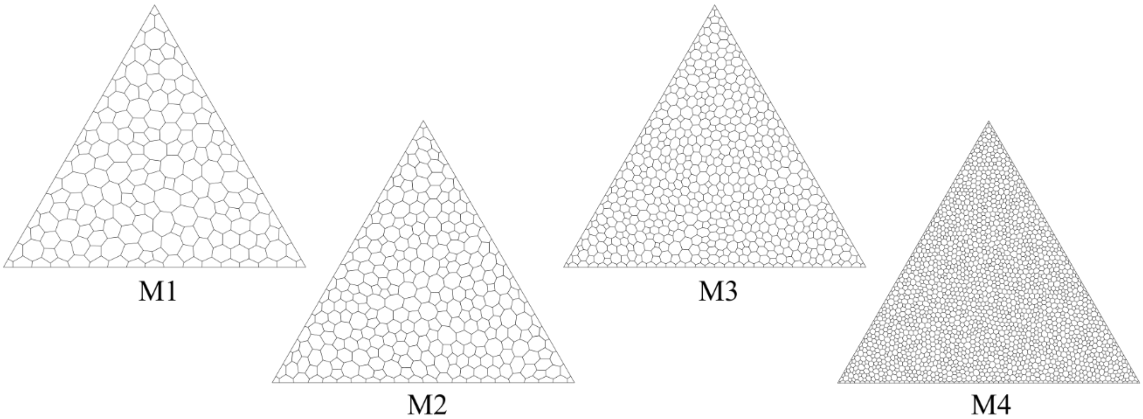

Figure 6. The different refinement levels for the Voronoi tessellations on Triangle cross-section.

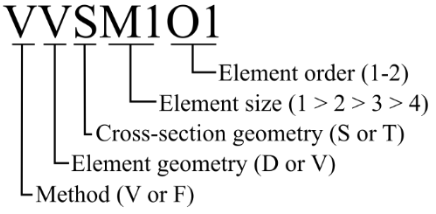
VVSM1O1
Element order (1-2)
Element size (1 > 2 > 3 > 4)
Cross-section geometry (S or T)
Element geometry (D or V)
Method (V or F)

Figure 7. Mesh codification.

The meshes used and their characterizing parameters are summarized in Table 1.

Table 1. Set of meshes and their respective parameters.

Code	h	n	Code	h	n
FDSM1O1	1.37E-01	105	FDTM1O1	1.28E-01	124
FDSM2O1	9.67E-02	230	FDTM2O1	9.76E-02	225
FDSM3O1	6.62E-02	493	FDTM3O1	7.26E-02	496
FDSM4O1	3.43E-02	2057	FDTM4O1	3.58E-02	2014
FDSM1O2	1.37E-01	457	FDTM1O2	1.28E-01	541
FDSM2O2	9.67E-02	973	FDTM2O2	9.76E-02	960
FDSM3O2	6.62E-02	2049	FDTM3O2	7.26E-02	2074
FDSM4O2	3.43E-02	8385	FDTM4O2	3.58E-02	8236
VDSM1O1	1.37E-01	105	VDTM1O1	1.28E-01	124
VDSM2O1	9.67E-02	230	VDTM2O1	9.76E-02	225
VDSM3O1	6.62E-02	493	VDTM3O1	7.26E-02	496
VDSM4O1	3.43E-02	2057	VDTM4O1	3.58E-02	2014
VDSM1O2	1.37E-01	705	VDTM1O2	1.28E-01	835
VDSM2O2	9.67E-02	1487	VDTM2O2	9.76E-02	1471
VDSM3O2	6.62E-02	3113	VDTM3O2	7.26E-02	3157
VDSM4O2	3.43E-02	12657	VDTM4O2	3.58E-02	12445
VVSM1O1	1.44E-01	248	VVTM1O1	1.46E-01	294
VVSM2O1	1.09E-01	514	VVTM2O1	1.06E-01	511
VVSM3O1	7.41E-02	1064	VVTM3O1	7.63E-02	1083
VVSM4O1	3.89E-02	4272	VVTM4O1	4.13E-02	4209
VVSM1O2	1.44E-01	785	VVTM1O2	1.46E-01	931
VVSM2O2	1.09E-01	1599	VVTM2O2	1.06E-01	1597
VVSM3O2	7.41E-02	3273	VVTM3O2	7.63E-02	3343
VVSM4O2	3.89E-02	12977	VVTM4O2	4.13E-02	12811

4 Results

There are three main results from a given solution for Prandtl's function around which the organization of this section is made. The first is the function itself; its general shape is a first visual qualitative indicator of the solution adequacy, as explained alongside sampled results, in subsection 4.1. The next result is the stress field, which is another visual qualitative indicator. This is further developed in subsection 4.2. The third result is the torsion constant obtained via Eq. (14); the convergence curves, which comprise a summary of the torsion constant results, are presented in subsection 4.3. All the results were obtained for unitary $G\theta'$.

4.1 Prandtl's function

The differential equations associated with Prandtl's function, Eq. (8), is analogous to the equations that describe the displacement field (w) of a homogeneous membrane, continuously supported (w = 0) along its boundary, which is loaded by a constant pressure (q), Eq. (31). This is known as the membrane analogy, as presented by Timoshenko [2]. This analogy was used to empirically estimate the torsion constant for cross-section geometries without known analytical solutions, in the time before the ascension of computational methods.

$$\Delta w = -\frac{q}{|\Omega|}. \tag{31}$$

The results reflect this analogy, as is evident in Figure 8 and in Figure 9.

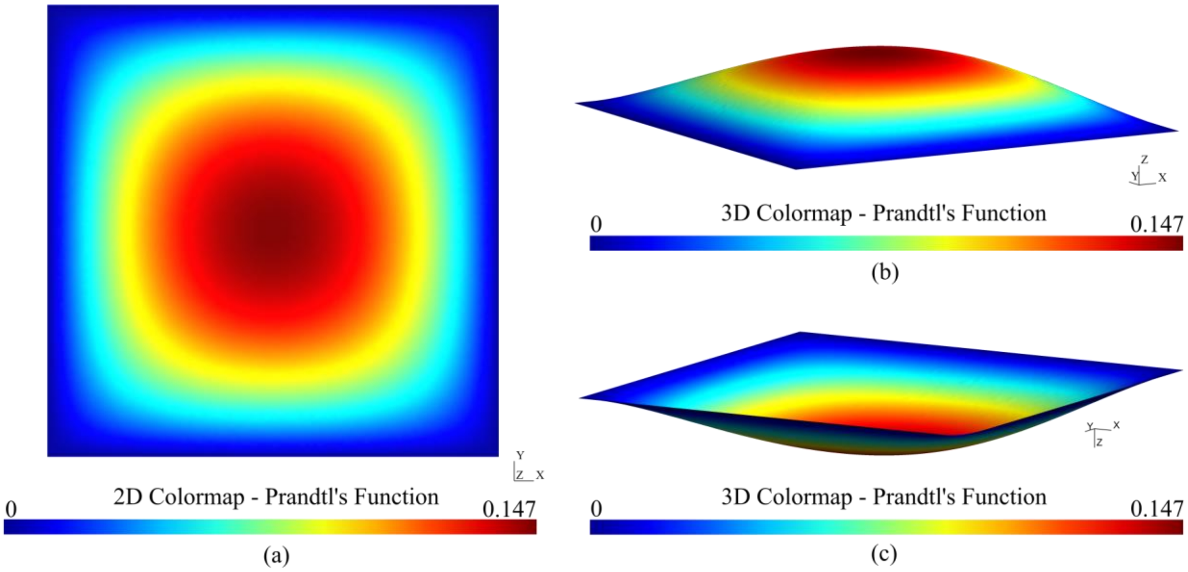

Figure 8. Prandtl's function result for VVSM4O2. (a) 2D colormap, (b) 3D colormap, (c) 3D colormap flipped.

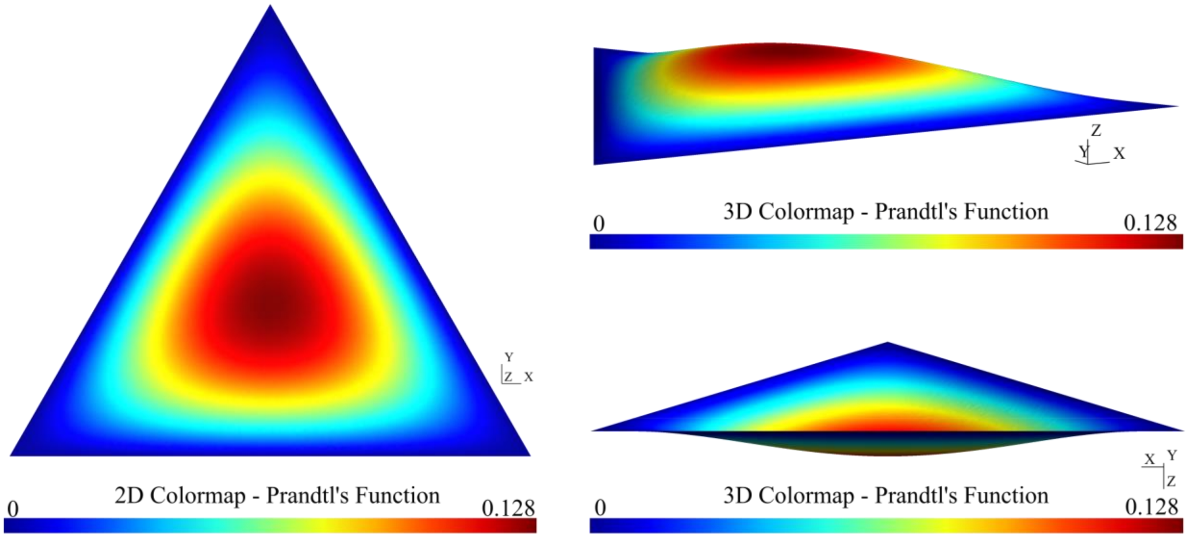

Figure 9. Prandtl's function result VVTM4O2. (a) 2D colormap, (b) 3D colormap, (c) 3D colormap flipped.

4.2 Tangential stresses

The tangential stresses are obtained from the derivatives of Prandtl's function as stated in Eq. (5). The stress field should obey certain requirements implicit in the problem. On the boundary there should be no stress in the normal direction, as expressed in the balance equation, Eq. (3). There should be no resultant force in any particular direction, only a resulting moment that balances the applied twisting moment (M_T). The results for the tangential stress fields (Figure 10 and Figure 11) satisfy these requirements.

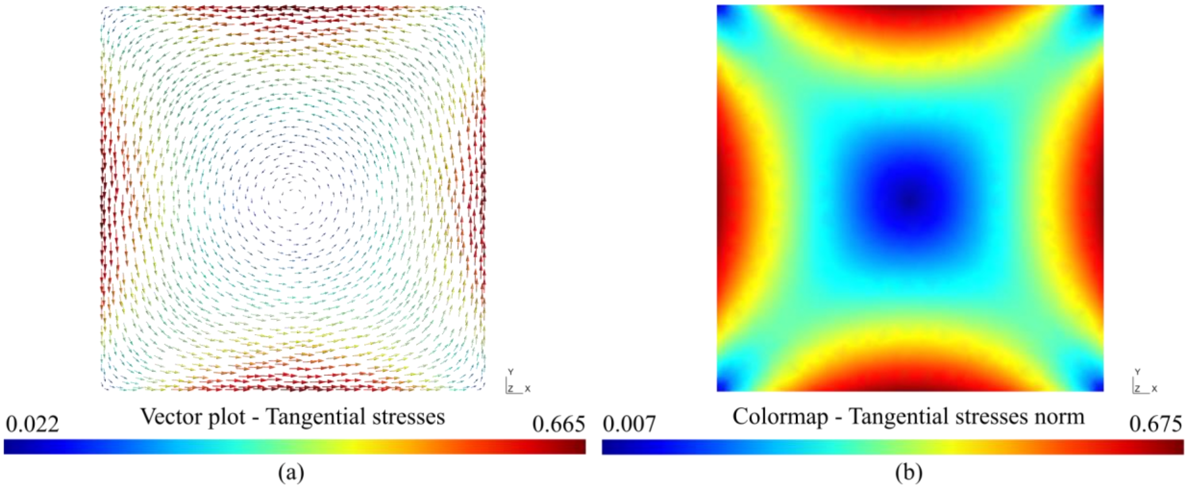

Figure 10. Stress field result for VVSM2O2. (a) Vector field and (b) tangential stress norm colormap.

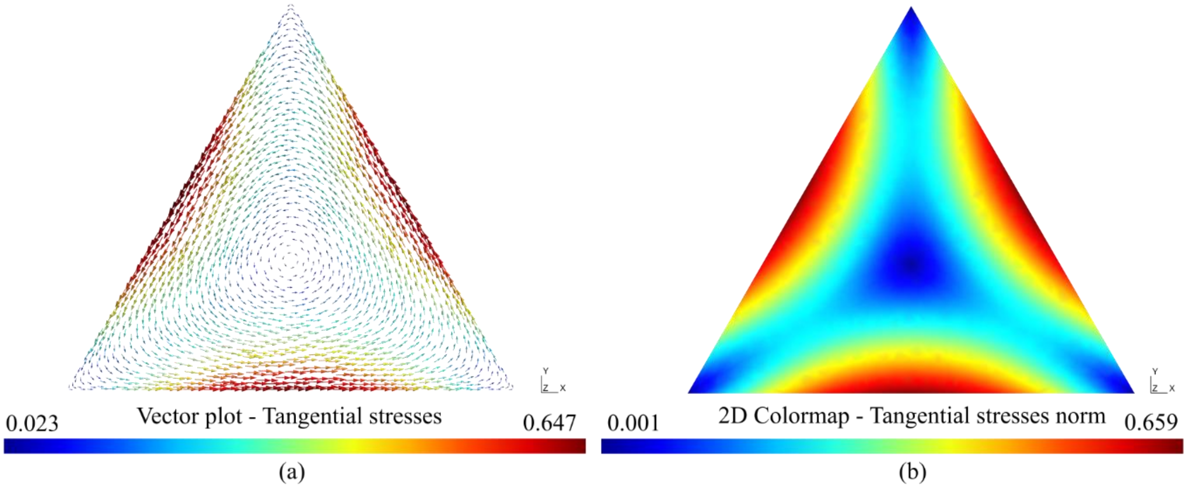

Figure 11. Stress field result for VVTM2O2. (a) Vector field and (b) tangential stress norm colormap.

4.3 Torsion constant convergence curves

The convergence curves (Figure 12 and Figure 13) show the relation between the error based on the torsion constant evaluation (e) of the approximated solution and the maximum element diameter (h). Note that the nomenclature adopted in the curves is an abbreviated version of the one established earlier; the cross-section geometry is presented in the title, and the element sizes are explicit in the four points in the curves. The relationship is shown, in the literature, to be expressed in the general form shown in Eq. (32). In this equation, C is a constant and k is a parameter that depends on the norm used to measure the error and the order of the full polynomial space contained in the element spaces.

$$\mathrm{e} \leq \mathrm{Ch}^{\mathrm{k}}. \tag{32}$$

Table 2. Set of meshes, parameters, and errors.

Code	h	n	e	Code	h	n	e
FDSM1O1	1.37E-01	105	1.76E-02	FDTM1O1	1.28E-01	124	1.88E-02
FDSM2O1	9.67E-02	230	9.00E-03	FDTM2O1	9.76E-02	225	1.12E-02
FDSM3O1	6.62E-02	493	4.24E-03	FDTM3O1	7.26E-02	496	5.30E-03
FDSM4O1	3.43E-02	2057	1.07E-03	FDTM4O1	3.58E-02	2014	1.36E-03
FDSM1O2	1.37E-01	457	1.19E-04	FDTM1O2	1.28E-01	541	1.30E-05
FDSM2O2	9.67E-02	973	2.57E-05	FDTM2O2	9.76E-02	960	4.19E-06
FDSM3O2	6.62E-02	2049	5.82E-06	FDTM3O2	7.26E-02	2074	9.40E-07
FDSM4O2	3.43E-02	8385	4.79E-07	FDTM4O2	3.58E-02	8236	6.09E-08
VDSM1O1	1.37E-01	105	1.76E-02	VDTM1O1	1.28E-01	124	1.88E-02
VDSM2O1	9.67E-02	230	9.00E-03	VDTM2O1	9.76E-02	225	1.12E-02
VDSM3O1	6.62E-02	493	4.24E-03	VDTM3O1	7.26E-02	496	5.30E-03
VDSM4O1	3.43E-02	2057	1.07E-03	VDTM4O1	3.58E-02	2014	1.36E-03
VDSM1O2	1.37E-01	705	8.06E-05	VDTM1O2	1.28E-01	835	5.07E-06
VDSM2O2	9.67E-02	1487	1.31E-05	VDTM2O2	9.76E-02	1471	1.57E-06
VDSM3O2	6.62E-02	3113	2.49E-06	VDTM3O2	7.26E-02	3157	3.99E-07
VDSM4O2	3.43E-02	12657	2.48E-07	VDTM4O2	3.58E-02	12445	2.58E-08
VVSM1O1	1.44E-01	248	1.52E-02	VVTM1O1	1.46E-01	294	1.56E-02
VVSM2O1	1.09E-01	514	7.67E-03	VVTM2O1	1.06E-01	511	9.37E-03
VVSM3O1	7.41E-02	1064	3.74E-03	VVTM3O1	7.63E-02	1083	4.58E-03
VVSM4O1	3.89E-02	4272	1.02E-03	VVTM4O1	4.13E-02	4209	1.15E-03
VVSM1O2	1.44E-01	785	1.92E-05	VVTM1O2	1.46E-01	931	2.87E-06
VVSM2O2	1.09E-01	1599	3.08E-06	VVTM2O2	1.06E-01	1597	8.17E-07
VVSM3O2	7.41E-02	3273	1.06E-06	VVTM3O2	7.63E-02	3343	2.64E-07
VVSM4O2	3.89E-02	12977	7.01E-08	VVTM4O2	4.13E-02	12811	1.68E-08

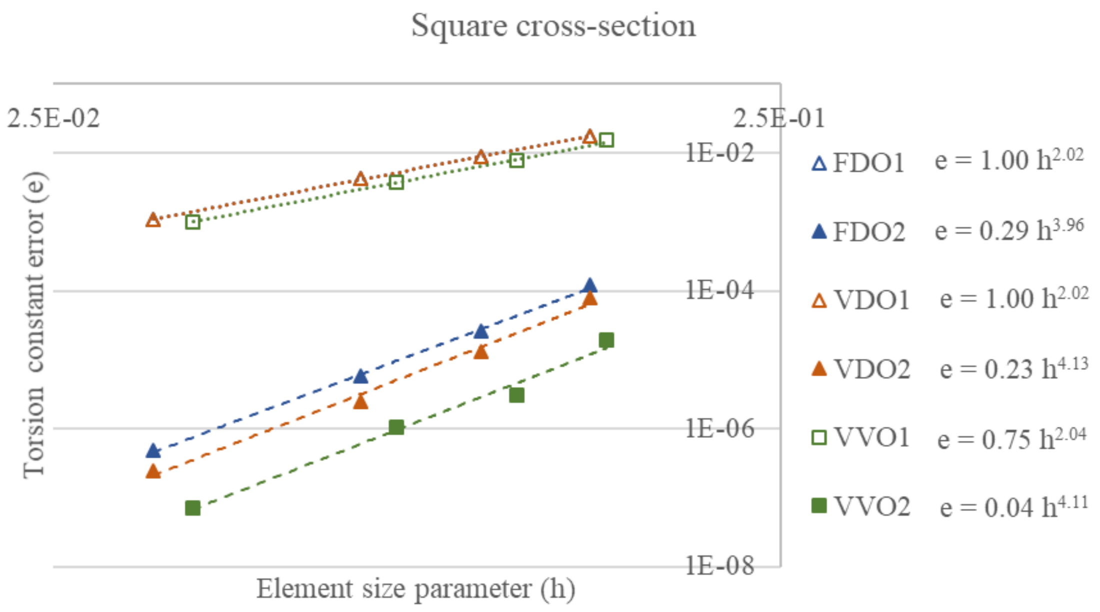

Figure 12. Convergence curves for square cross-section.

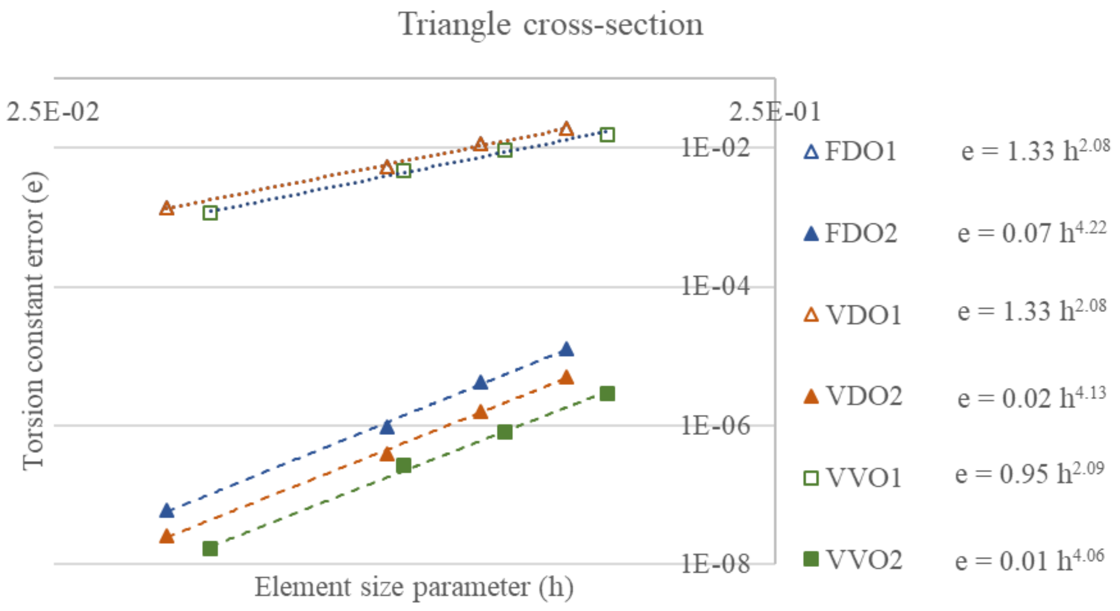

Figure 13. Convergence curves for triangle cross-section.

Thes curves show the total coincidence between the finite and virtual element methods when using linear polynomials on triangular meshes: FDO1 and VDO1 are both shown in all figures presenting convergence curves, but completely coincide; this is evident in the torsion constant error values presented in Table 2. This coincidence arises because, for this specific geometry, the local virtual element space coincides exactly with the full polynomial space, requiring no additional virtual functions. In the second order elements this no longer happens, because the virtual element function space has one more dimension, associated with the internal degree of freedom related with the function’s Laplacian.

The differences between the first and second order elements are evident, as expected, in the exponent of the relationship, which can be clearly seen in the slope of the curves.

The convergence curves are traditionally presented associating the error with the element diameter. However, in order to compare two methods, it is important to consider the computational resources requirements. A superficial way to consider this is to plot the error’s relationship with the number of degrees of freedom of the system. This result is presented in Figure 14 and Figure 15.

5 Conclusion

The VEM was successful in producing weak solutions for Prandtl’s formulation of St. Venant torsion. The results from both FEM and VEM were very similar, and in one case even exactly the same.

The VEM allows the use of meshes of generically shaped elements, while retaining the same convergence properties as the finite element method. This is shown with the use of Voronoi tessellations. This versatility may not provide an advantage for this specific problem; however, it has been explored in other works as a mean to allow for easier mesh refining algorithms such as quadtrees; allowing for more efficient contact processing algorithms; and for non-conforming meshes compatibilization.

The generalization of both geometry and order of the polynomial spaces provide the method with immense potential for applications.

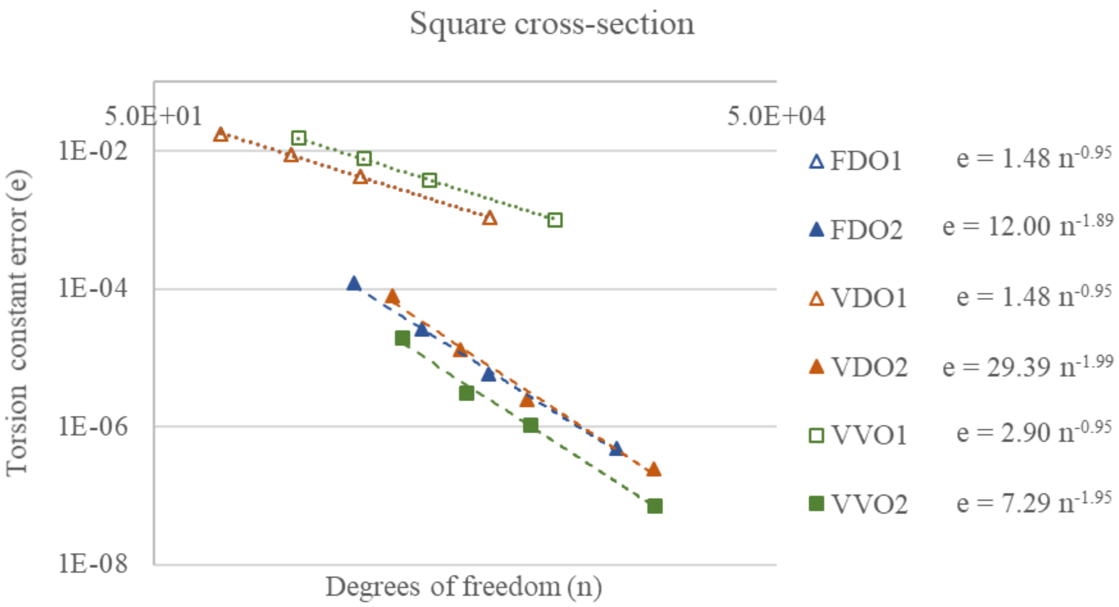

Figure 14. Convergence curves for square cross-section.

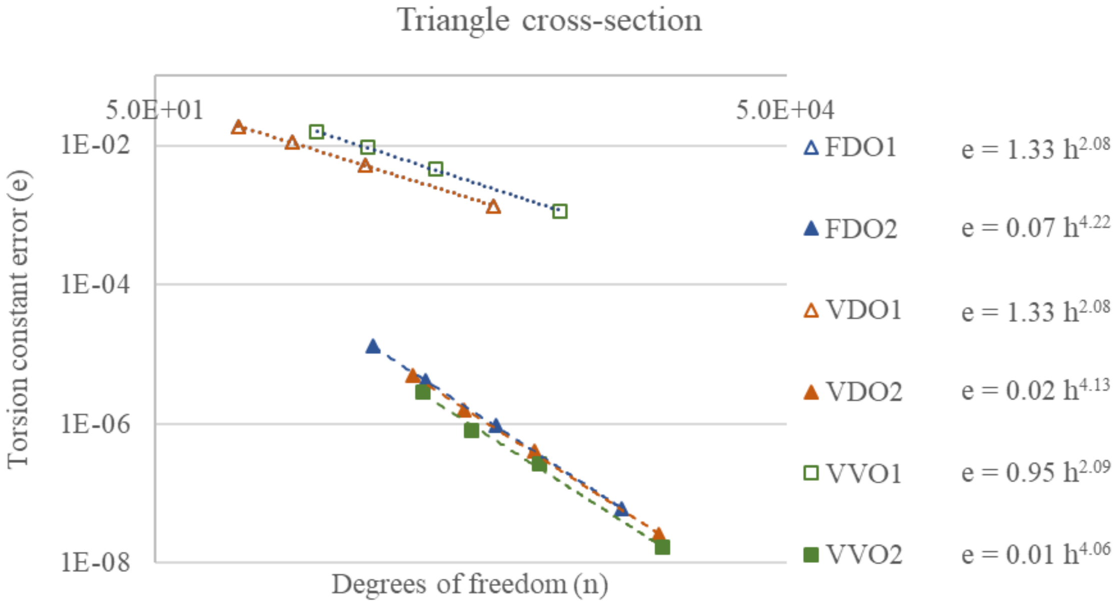

Figure 15. Convergence curves for triangle cross-section.

Acknowledgements

The authors acknowledge Vale S.A. for the support through Wheel-Rail Chair project. The second author acknowledges CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico) under the Grant 304680/2018-4.

References

[1] VLASOV and V. Z., “Thin-Walled Elastic Beams,” *PST Cat.*, vol. 428, 1959.

[2] S. Timoshenko, *Theory of Elasticity*. New York: McGraw-Hill, 1951.

[3] L. Beirão da Veiga, F. Brezzi, A. Cangiani, G. Manzini, L. D. Marini, and A. Russo, “Basic Principles of Virtual Element Methods,” *Math. Model. Methods Appl. Sci.*, vol. 23, no. 01, pp. 199–214, 2013.

[4] F. Brezzi, “The Great Beauty of VEMs,” 2014.

[5] L. Beirão da Veiga and G. Manzini, “A virtual element method with arbitrary regularity,” *IMA J. Numer. Anal.*, vol. 34, no. 2, pp. 759–781, 2014.

[6] L. Beirão da Veiga, A. Russo, and G. Vacca, “The Virtual Element Method with curved edges,” no. November 2017, 2017.

[7] B. Ahmad, A. Alsaedi, F. Brezzi, L. D. Marini, and A. Russo, “Equivalent projectors for virtual element methods,” *Comput. Math. with Appl.*, vol. 66, no. 3, pp. 376–391, 2013.

[8] F. Brezzi, R. S. Falk, and L. Donatella Marini, “Basic principles of mixed Virtual Element Methods,” *ESAIM Math. Model. Numer. Anal.*, vol. 48, no. 4, pp. 1227–1240, 2014.

[9] L. Beirão da Veiga, C. Lovadina, and A. Russo, “Stability Analysis for the Virtual Element Method,” 2016.

[10] L. Beirão da Veiga, F. Dassi, and A. Russo, “High-order Virtual Element Method on polyhedral meshes,” *Comput. Math. with Appl.*, vol. 74, no. 5, pp. 1110–1122, 2017.

[11] G. Vacca, “Virtual Element Methods for hyperbolic problems on polygonal meshes,” *Comput. Math. with Appl.*, vol. 74, no. 5, pp. 882–898, 2017.

[12] G. Vacca and L. Beirão da Veiga, “Virtual Element Methods for Parabolic Problems on Polygonal Meshes,” *Numer. Methods Partial Differ. Equ.*, vol. 31, no. July 2015, pp. 2110–2134, 2015.

[13] L. Beirão da Veiga, F. Brezzi, L. D. Marini, and A. Russo, “Virtual Element Methods for general second order elliptic problems on polygonal meshes,” 2014.

[14] A. Cangiani, E. H. Georgoulis, T. Pryer, and O. J. Sutton, “A posteriori error estimates for the virtual element method,” pp. 1–27, 2016.

[15] L. Beirão da Veiga and G. Manzini, “Residual *a posteriori* error estimation for the Virtual Element Method for elliptic problems,” *ESAIM Math. Model. Numer. Anal.*, vol. 49, no. 2, pp. 577–599, 2015.

[16] S. C. Brenner, Q. Guan, and L.-Y. Sung, “Some Estimates for Virtual Element Methods,” *Comput. Methods Appl. Math.*, vol. 17, no. 4, pp. 387–407, Jan. 2017.

[17] L. Chen and J. Huang, “Some error analysis on virtual element methods,” *Calcolo*, vol. 55, no. 1, p. 5, Mar. 2018.

[18] O. J. Sutton, “The virtual element method in 50 lines of MATLAB,” *Numer. Algorithms*, vol. 75, no. 4, pp. 1141–1159, 2017.

[19] L. Beirão da Veiga, F. Brezzi, L. D. Marini, and A. Russo, “The Hitchhiker’s Guide to the Virtual Element Method,” *Math. Model. Methods Appl. Sci.*, vol. 24, no. 08, pp. 1541–1573, 2014.

[20] N. S. Trahair, “Nonlinear Elastic Nonuniform Torsion,” *J. Struct. Eng.*, vol. 131, no. 7, pp. 1135–1142, 2005.

[21] E. J. Sapountzakis and V. J. Tsipiras, “Nonlinear inelastic uniform torsion of composite bars by BEM,” *Comput. Struct.*, vol. 87, no. 3–4, pp. 151–166, 2009.

[22] P. G. Ciarlet, *The Finite Element Method for Elliptic Problems*. SIAM, 2002.

[23] B. Hudobivnik, F. Aldakheel, and P. Wriggers, “A low order 3D virtual element formulation for finite elasto–plastic deformations,” *Comput. Mech.*, vol. 63, no. 2, pp. 253–269, 2019.

[24] C. Geuzaine and J.-F. Remacle, “A three-dimensional finite element mesh generator with built-in pre- and post-processing facilities,” *Int. J. Numer. Meth. Engng.*, vol. 79, no. 11, pp. 1309–1331, 2017.