On Why Mesh-Untangling May Not be Required

Bhagyashree Prabhune, Krishnan Suresh*

Department of Mechanical Engineering, University of Wisconsin-Madison, WI, USA

Abstract

Generating tangle-free high quality hexahedral meshes is an ongoing challenge. Tangled meshes, i.e., meshes containing negative Jacobian elements, are unsuitable for finite element (FE) simulations as they lead to erroneous results. Consequently, many untangling methods have been proposed; however, untangling is not always achievable.

The present paper addresses this challenge by allowing tangled meshes for FE analysis with the use of the isoparametric tangled finite element method (i-TFEM). The proposed method efficiently handles complex configurations of tangled elements, making it suitable for real-world scenarios. By introducing minor modifications to standard FEM, i-TFEM offers an easy implementation and reduces to standard FEM for non-tangled meshes. Numerical experiments, involving both linear and nonlinear elasticity, demonstrate the accuracy, convergence characteristics, and applicability of the method to real-world tangled meshes. The results emphasize the importance of reevaluating mesh quality indicators for tangled meshes.

Keywords: Tangled Mesh, Negative Jacobian, Foldover, Inverted elements, Mixed finite element

1. Introduction

Automatic high-quality hexahedral mesh generation remains an open challenge [1, 2]. To quote [3], "*Generating high quality conformal hexahedral meshes in arbitrary 3D domains is one of the most challenging open problems in mesh generation.*" The underlying reasons are due to the severe topological and geometric constraints imposed [4]: (a) the mesh must not be tangled, (b) the elements must be of high quality, (c) the mesh must conform to the geometry, and (d) must be topologically well-structured. Satisfying all these requirements is non-trivial. This paper primarily focuses on the first constraint, i.e., mesh tangling for 8-noded hexahedral elements. This constraint implies that the Jacobian determinant must remain positive over the entire mesh. The paper specifically deals with tangled meshes characterized by partially inverted elements, i.e. elements with negative Jacobian determinant at some (not all) Gauss points. An example of a tangled mesh is shown in Fig. 1; such meshes are unacceptable today since they lead to erroneous finite element results.

*Corresponding author

Email addresses: bprabhune@wisc.edu (Bhagyashree Prabhune), ksuresh@wisc.edu (Krishnan Suresh)

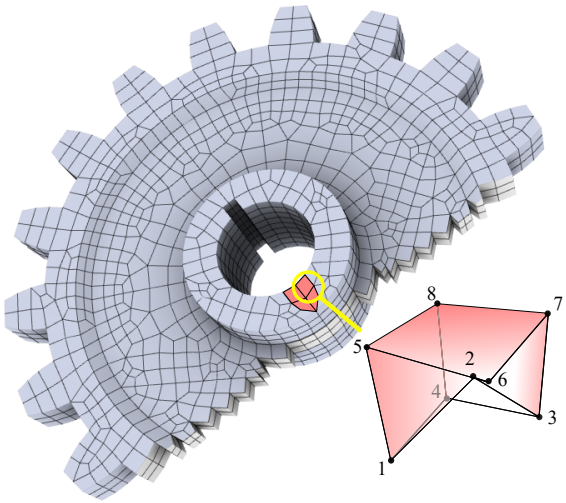


Figure 1: Gear mesh provided by [5]; elements in red color are concave (tangled).

State-of-the-art mesh generation methods [6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20] often struggle to produce tangle-free meshes. Many commercial software systems depend heavily on user interaction to simplify and decompose complex geometries until they become suitable for meshing [2, 21]. Although attempts have been made to incorporate fully automatic grid-based methods [22, 23, 24, 25, 26] into professional software such as CUBIT [27] and Distene SAS [28], these methods are generally considered inferior in terms of mesh quality [2]. To improve mesh quality, smoothing techniques [29, 30] are employed; however, generating tangle-free meshes is not always feasible [2, 25, 31]. As stated in [2], *"maintaining the inversion-free property of hex-mesh poses a great challenge."* Tangled meshes are also encountered during shape optimization [32], large deformation simulations [33, 34], and mesh morphing [35].

Numerous untangling algorithms have been developed to address these challenges [36, 37, 5]. However, untangling is not always guaranteed, as reported in multiple instances where no tangle-free solution is possible [5, 38, 37, 3]. As emphasized in [3], *"... it is probably impossible to untangle the mesh under these hard constraints."* In fact, certain topological structures may not have an untangled mesh solution [5, 4].

Since standard finite element method leads to erroneous results over a tangled mesh, several non-traditional finite element techniques such as the virtual element method [39, 40], smoothed finite element method [41], polygonal finite element methods [42], unsymmetric finite element such as US-ATFH8 [43] have been developed to directly handle some of the tangled mesh configurations. However, these do not reduce to standard FEM for tangle-free mesh and may require significant changes to the FEM framework. They also impose restrictive assumptions on element configurations. For instance, some of these methods cannot handle elements with non-planar surfaces [44] that are common in real-world tangled hexahedral meshes.

Recently, an isoparametric tangled finite element method (i-TFEM) [45, 46, 47, 48] was proposed to handle tangled meshes. It was shown that i-TFEM reduces to standard FEM for non-tangled meshes,

and easily handles non-planar elements. However, i-TFEM was limited to a specific class of 3D tangled configurations; commonly occurring configurations such as penetrating elements could not be handled.

In this paper, we present an improved version of i-TFEM to handle such configurations. Furthermore, the improved version is more efficient and easier to implement. The accuracy and effectiveness of the proposed 3D i-TFEM are demonstrated through a series of numerical experiments involving linear and nonlinear elasticity. These experiments involve numerous real-world tangled meshes, showcasing the robustness of the method.

The remainder of this paper is organized as follows: Section 2 presents various tangled element configurations. Section 3 provides a review of the fundamental concept underlying i-TFEM along with the i-TFEM formulation. This is followed by a detailed discussion of its implementation in Section 4. Section 5 presents the numerical experiments, and Section 6 discusses the implications of i-TFEM on mesh quality metrics. Finally, the conclusions are presented in Section 7.

2. Types of Tangling

In this paper, we consider an element to be tangled if the Jacobian determinant at any of the quadrature points is negative. This definition is aligned with downstream FEM. However, in the meshing community, an alternate definition of a scaled Jacobian at the corner nodes is often used [49].

Consider the tangled (non-convex) 8-node hexahedral (H8) element in the physical space (x_1, x_2, x_3) in Fig. 2a, and the parametric mapping ϕ from the (ξ_1, ξ_2, ξ_3) space in Fig. 2b to the tangled element. Due to the non-convex nature of the element, one can show that [45] the parametric space can be divided into positive and negative regions, denoted by J^+ and J^- respectively. The corresponding physical regions are referred to as positive (C^+) and negative (C^-) components respectively; see Fig. 2c. Furthermore, for a point $\boldsymbol{a}$ in J^- that maps to a point $\boldsymbol{p}$ in the physical space, there is a corresponding point $\boldsymbol{b}$ in J^+ that maps to the same point $\boldsymbol{p}$; such physical points lie outside the element. In other words, the element folds onto itself, leading to an overlapping region, or a fold F (see Fig. 2c).

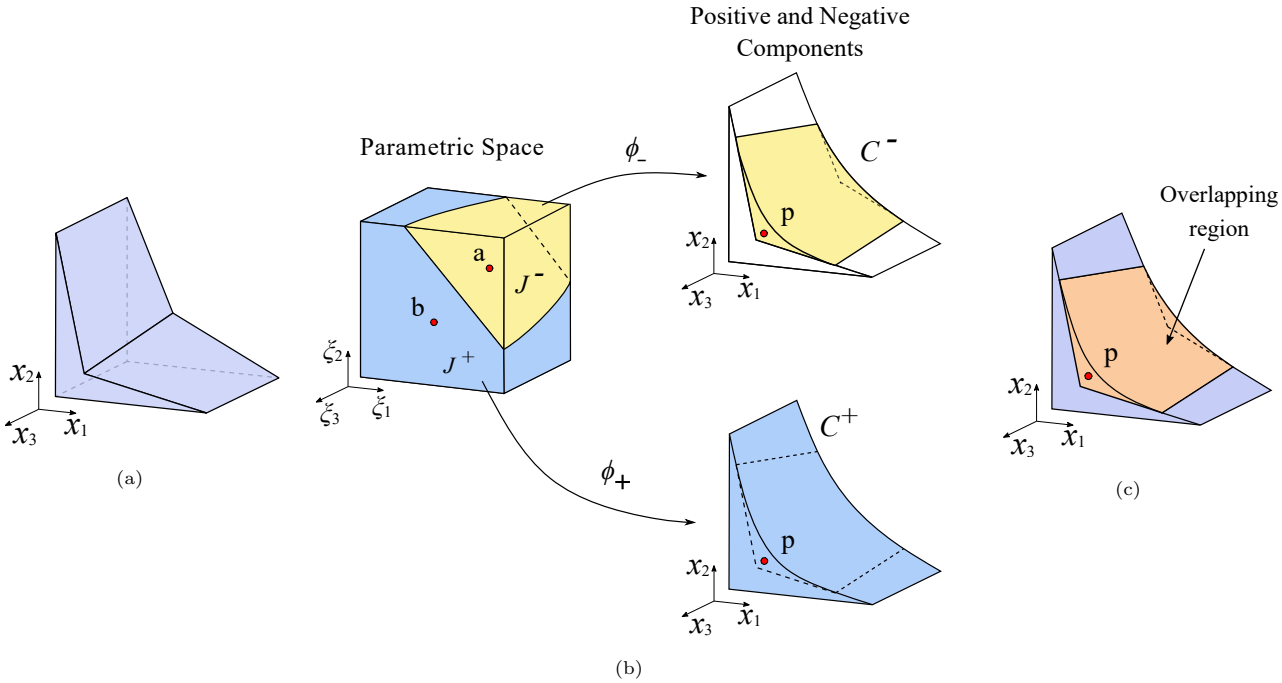


Figure 2: (a) Physical space of the tangled H8 element. (b) Parametric space of the tangled element, that can be divided into positive and negative Jacobian regions. Corresponding physical space with positive and negative components. (c) Tangled element with the overlapping region.

Observe that the negative component C^- is entirely contained within the positive component C^+. As a result, the overlapping region can be expressed as $F = C^+ \cap C^- = C^-$. In other words, the entire J^- region maps to the overlapping region, which lies outside the physical boundary of the element. The previous work on i-TFEM [45] solely focused on handling such elements.

However, in many real-world meshes, the negative component C^- is *not* entirely contained lie within the positive component C^+. Figure 3a illustrates a penetrating element along with the corresponding parametric space in Fig. 3b. One can show that for this element, only a part of J^- region, shown in yellow in Fig. 3c maps to an overlapping region within C^+ that lies outside the element, while the remainder J^-, shown in red in Fig. 3c, maps to a region physically within the element. Visualizing these physical regions is hard but fortunately not necessary. We only need to understand the differences in the parametric space; methods to distinguish such cases are discussed later.

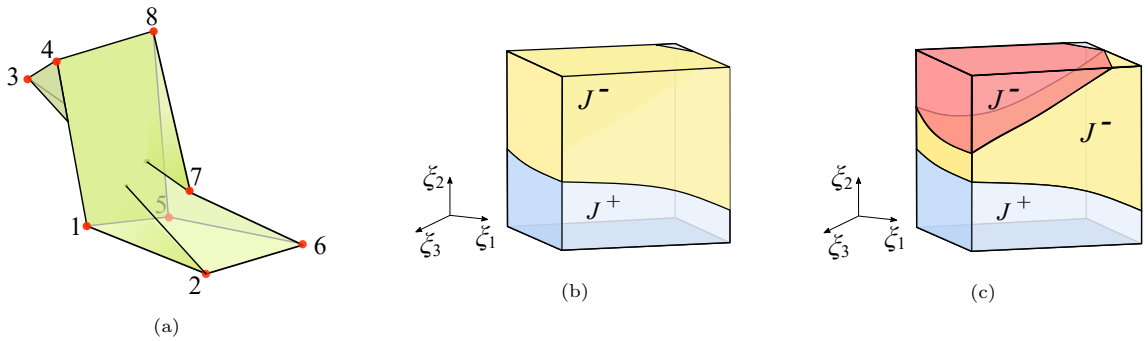


Figure 3: (a) Penetrating element (b) its parametric space; J^- region is shown in yellow color (c) J^- region which does not map to the fold is shown in red color.

Furthermore, there are cases where elements have disconnected J^- regions; Fig. 4 illustrates one such penetrating element. However, no special treatment is needed for such elements.

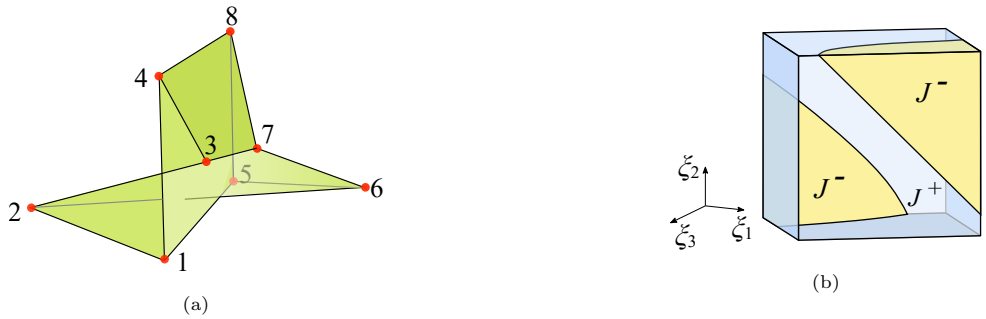


Figure 4: (a) Penetrating element with disconnected J^- regions, (c) its parametric space; J^- shown in yellow color.

Penetrating elements are frequently encountered in real-world meshes, and the proposed i-TFEM formulation is capable of handling such tangled elements. The case where J^- spans the entire parametric space, i.e., a fully inverted element, is *not* addressed in this paper. Such elements can occur, for instance, during mesh morphing and hex meshing techniques which involve domain-transfer such as automatic PolyCube-based approaches, and will be considered in the future. Further, the default case where J^- region is null corresponds to the non-tangled case, and i-TFEM reduces to standard FEM. In summary, we only need to distinguish between the following three cases:

1. The Jacobian determinant is positive at all Gaussian points (*non-tangled* element).
2. The Jacobian determinant is negative at some of the Gaussian points, and for each of these points, there is a corresponding point with a positive Jacobian determinant (*non-penetrating tangled* elements as in Fig. 2c).
3. The Jacobian determinant is negative at some of the Gaussian points, and for some of these points, a corresponding point with a positive Jacobian determinant does not exist (*penetrating tangled* elements as in Fig. 3c).

The difference between case(2) and case (3) is only relevant to Section 4.2.

Finally, it is also common for the overlapping region to be shared by multiple non-tangled elements as illustrated in Fig. 5. In this case, three non-tangled elements E_2, E_3, and E_4 intersect with the overlapping region of the tangled element E_1. However, no special treatment is needed, i.e., we only need to consider the parametric space of the tangled element.

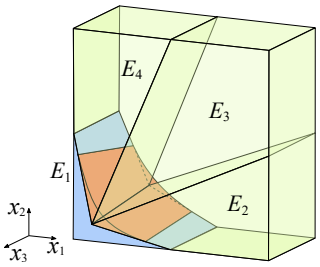


Figure 5: Overlapping region shared by multiple non-tangled elements.

3. Proposed Methodology

3.1. i-TFEM Concept

The main idea in i-TFEM is to treat the positive and negative Jacobian regions (J^+ and J^-) separately during stiffness matrix computation. This avoids all challenges associated with non-invertible mapping since the two independent mappings:

$$\phi_{\pm} \colon J^{\pm} \to C^{\pm}$$

are both invertible.

For a tangled element E_j, let $\boldsymbol{N}_j(\boldsymbol{\xi})$ be the standard shape functions (i.e., trilinear Lagrange shape functions for an H8 element) defined over the parametric space. Now, let $\boldsymbol{N}_j^{\pm}$ be the restriction of $\boldsymbol{N}_j$ to $J^{\pm}$, i.e.,

$$\boldsymbol{N}_j^{\pm}(\boldsymbol{x}) \coloneqq \boldsymbol{N}_j(\phi_{\pm}^{-1}(\boldsymbol{x})) \tag{1}$$

The corresponding displacement fields defined over $C_j^{\pm}$ are defined as:

$$\boldsymbol{u}_j^{\pm}(\boldsymbol{x}) \coloneqq \boldsymbol{N}_j^{\pm}(\boldsymbol{x})\hat{\boldsymbol{u}}_j \tag{2}$$

where, $\hat{\boldsymbol{u}}_j$ is the displacement vector for j^{th} element.

Thus, for any point $\boldsymbol{x}$ in the fold F_j that belongs to both C_j^+ and C_j^-, one can define two fields $\boldsymbol{u}_j^+(\boldsymbol{x})$ and $\boldsymbol{u}_j^-(\boldsymbol{x})$ due to the fold. To resolve this ambiguity, in i-TFEM, a *piecewise compatibility constraint* is enforced over the fold (see [45] for further discussion):

$$\boldsymbol{u}_j^+(\boldsymbol{x}) - \boldsymbol{u}_j^-(\boldsymbol{x}) = 0, \quad \forall \boldsymbol{x} \in F_j. \tag{3}$$

Now consider a simple domain Ω that is discretized into two elements shown in Fig. 6a, where one of the elements (E_1) is tangled. For simplicity, we will assume this is a non-penetrating tangled element, but

the discussion applies to the penetrating case as well, unless otherwise noted. The positive and negative components of E_1 are shown in Fig. 6b. These components overlap with each other; the overlapping region (fold) is denoted as F_1. Element E_2 is not tangled and has only one positive component (see Fig. 6c), i.e., $E_2 = C_2^+$ while $C_2^- = \emptyset$. However, the components C_1^- and C_1^+ overlap with E_2 as well, i.e., the overlapping region of the tangled element (F_1) now intersects with E_2. Construction of the stiffness matrix, i.e., integrating shape functions and their derivatives over a tangled mesh requires careful consideration of this overlap. For example, to compute the total volume of the two-element mesh, we must subtract the integral over C_1^- to avoid double-counting:

$$\int_{E_1+E_2} d\Omega = \int_{C_1^+} d\Omega + \int_{C_2^+} d\Omega - \int_{C_1^-} d\Omega \tag{4}$$

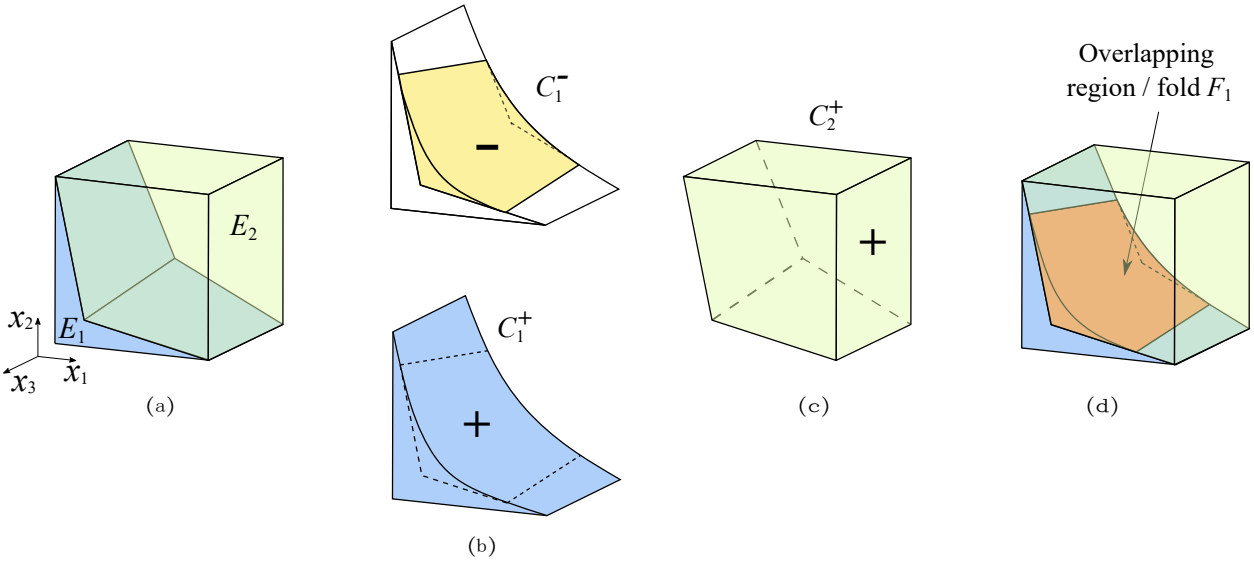


Figure 6: (a) 3-D domain discretized into two 8-node hexahedral elements. (b) Positive and negative components of the tangled element. (c) Non-tangled element of the mesh. (d) Overlapping region intersects with the neighboring convex element.

3.2. i-TFEM formulation for linear elasticity

Now consider an elasticity problem over the two-element mesh. The body is subjected to body forces $\boldsymbol{b}$ and tractions $\boldsymbol{t}$ on $\partial\Omega^t$. We will assume that the field $\boldsymbol{u}$ satisfies Dirichlet boundary conditions $\boldsymbol{u} = \boldsymbol{u}^d$ over $\partial\Omega^d$. Let the material be linear elastic and the elasticity tensor be represented by $\boldsymbol{D}$.

Using the standard finite element method (based on the Galerkin formulation) [50], the displacements can be obtained, after enforcing the Dirichlet boundary conditions, by solving the following system of equations:

$$\boldsymbol{K}\hat{\boldsymbol{u}} = \boldsymbol{f} \tag{5}$$

where the stiffness matrix and the forcing term are given by:

$$\boldsymbol{K} = \prod_{j\in\{1,2\}} \int_{E_j} \left(\nabla \boldsymbol{N}_j{}^\top \boldsymbol{D} \nabla \boldsymbol{N}_j\right) d\Omega \quad \text{and} \quad \boldsymbol{f} = \prod_{j\in\{1,2\}} \int_{E_j} \boldsymbol{N}_j^\top \boldsymbol{b} d\Omega. \tag{6}$$

Unfortunately, for a tangled mesh such as the one in Fig. 6, standard FEM will lead to erroneous results (demonstrated later in Section 5). To resolve this, we make two modifications. Firstly, the stiffness matrix and the forcing term are defined via integration over the *components*, rather than elements, where integrals over negative components must be subtracted to avoid double-counting (see Eq. 4). Specifically, the stiffness matrix associated with the tangled element E_1 in Fig. 6 is computed via:

$$\widehat{\boldsymbol{k}}_1 = \int_{C_1^+} \left(\nabla \boldsymbol{N}_1^{+\top} \boldsymbol{D} \nabla \boldsymbol{N}_1^{+} \right) d\Omega \quad - \quad \int_{C_1^-} \left(\nabla \boldsymbol{N}_1^{-\top} \boldsymbol{D} \nabla \boldsymbol{N}_1^{-} \right) d\Omega, \tag{7}$$

Secondly, the compatibility constraint (Eq. 3) is enforced over every tangled element via the Lagrange multiplier field $\boldsymbol{\lambda} = \boldsymbol{N}^{\lambda} \hat{\boldsymbol{\lambda}}$ where $\boldsymbol{N}^{\lambda}$ indicates the shape functions of the Lagrange multiplier. For the two-element mesh in Fig. 6a, we will require that:

$$\int_{F_1} \delta \boldsymbol{\lambda}_1 \cdot \left(\boldsymbol{u}_1^{+} - \boldsymbol{u}_1^{-} \right) \mathrm{d}\Omega = 0 \quad \forall\, \delta \boldsymbol{\lambda}_1 \in L^2, \tag{8}$$

This can be expressed in matrix form as

$$\boldsymbol{C}^{\top} \hat{\boldsymbol{u}} = \boldsymbol{0} \quad \text{where} \quad \boldsymbol{C} = \int_{C_1^-} \left(\boldsymbol{N}_1^{+} - \boldsymbol{N}_1^{-} \right)^{\top} \boldsymbol{N}^{\lambda} d\Omega \tag{9}$$

Accordingly, in i-TFEM, Eq. 5 is replaced by the following system:

$$\widehat{\boldsymbol{K}} \hat{\boldsymbol{u}} + \boldsymbol{C} \hat{\boldsymbol{\lambda}} = \widehat{\boldsymbol{f}} \tag{10a}$$

$$\boldsymbol{C}^{\top} \hat{\boldsymbol{u}} = \boldsymbol{0} \tag{10b}$$

For a generic mesh with elements indexed by the set $I = \{1, \ldots, m\}$, and the tangled elements by the set $I_{\text{tangled}} \subset I$, the stiffness matrix is computed as:

$$\widehat{\boldsymbol{K}} = \prod_{j \in I} \int_{C_j^+} \left(\nabla \boldsymbol{N}_j^{+\top} \boldsymbol{D} \nabla \boldsymbol{N}_j^{+} \right) d\Omega \quad - \prod_{j \in I_{\text{tangled}}} \int_{C_j^-} \left(\nabla \boldsymbol{N}_j^{-\top} \boldsymbol{D} \nabla \boldsymbol{N}_j^{-} \right) d\Omega, \tag{11}$$

the forcing term by:

$$\widehat{\boldsymbol{f}} = \prod_{j \in I} \int_{C_j^+} \boldsymbol{N}_j^{+\top} \boldsymbol{b}\, d\Omega \quad - \prod_{j \in I_{\text{tangled}}} \int_{C_j^-} \boldsymbol{N}_j^{-\top} \boldsymbol{b}\, d\Omega \quad + \quad \prod_{j \in I_{\partial E_j^t}} \int \boldsymbol{N}_j^{\top} \boldsymbol{t}\, dS \tag{12}$$

and the constraint matrix by:

$$\boldsymbol{C} = \prod_{j \in I_{\text{tangled}}} \int_{F_j} \left(\boldsymbol{N}_j^{+} - \boldsymbol{N}_j^{-} \right)^{\top} \boldsymbol{N}^{\lambda} d\Omega. \tag{13}$$

In Eq. 13, the choice of $\boldsymbol{N}^{\lambda}$ and computation of the integral is discussed later in Section 4. When the mesh has no tangled elements, Eq. 10 reduces to Eq. 5. Further, observe that the expressions in Eq. 11 and Eq. 12 entail integration over non-convex regions, C_j^+ and C_j^-, associated with tangled elements. Fortunately, this complex operation can be circumvented, as discussed next.

4. i-TFEM Implementation

4.1. Computing the Stiffness Matrix

Recall that the stiffness matrix $\widehat{\boldsymbol{k}}_1$ associated with the tangled element E_1 involves integration over the components C_1^+ and C_1^- (see Eq. 16). Since C_1^+ (C_1^-) gets mapped to the J^+ (J^-) region of the parametric space (see Fig. 2c), the differential volume for C_1^+ is given by:

$$d\Omega = dx_1 dx_2 dx_3 = |\boldsymbol{J}| d\xi_1 d\xi_2 d\xi_3. \tag{14}$$

On the other hand, the differential volume for C_1^- is given by:

$$d\Omega = dx_1 dx_2 dx_3 = -|\boldsymbol{J}| d\xi_1 d\xi_2 d\xi_3. \tag{15}$$

Notice the negative sign appears before $|\boldsymbol{J}|$. This is to ensure that the volume $d\Omega$ remains positive, even though $|\boldsymbol{J}|$ is negative for C_1^-. Accordingly,

$$\widehat{\boldsymbol{k}}_1 = \int_{J^+} \left(\nabla \boldsymbol{N}_1^{+\top} \boldsymbol{D} \nabla \boldsymbol{N}_1^{+} \right) |\boldsymbol{J}| d\xi_1 d\xi_2 d\xi_3 \quad + \quad \int_{J^-} \left(\nabla \boldsymbol{N}_1^{-\top} \boldsymbol{D} \nabla \boldsymbol{N}_1^{-} \right) |\boldsymbol{J}| d\xi_1 d\xi_2 d\xi_3, \tag{16}$$

Since $J^+ \cup J^- = [-1, 1]^3$ covers the entire parametric space, one can employ standard Gauss quadrature to compute the stiffness of the element:

$$\widehat{\boldsymbol{k}}_1 = \int_{J^+ \cup J^-} \left(\nabla \boldsymbol{N}_1{}^{\top} \boldsymbol{D} \nabla \boldsymbol{N}_1 \right) |\boldsymbol{J}| d\xi_1 d\xi_2 d\xi_3 \tag{17}$$

However, one must include the sign of the Jacobian determinant. Thus, one can avoid separate integration over the non-convex C_1^+ and C_1^- regions.

Further, the elemental stiffness matrix associated with the non-tangled element E_2 can be computed in a standard fashion:

$$\widehat{\boldsymbol{k}}_2 = \int_{J^+} \left(\nabla \boldsymbol{N}_2^{+\top} \boldsymbol{D} \nabla \boldsymbol{N}_2^{+} \right) |\boldsymbol{J}| d\xi_1 d\xi_2 d\xi_3 \tag{18}$$

In summary, to obtain the elemental stiffness matrices in i-TFEM, standard Gauss integration can be employed *but the sign of the Jacobian determinant must be retained.* This applies to all three cases discussed in Section 2. In many FEM implementations, the absolute value of the Jacobian determinant is used, and this will lead to erroneous results. The forcing terms are computed similarly, taking into account the sign of the Jacobian. *In addition, the constraint matrix, discussed next, must be included if any of the Gauss points lie in the negative Jacobian region, i.e. if the element is tangled.*

4.2. Computing the Constraint Matrix

In order to compute the constraint matrix $\boldsymbol{C}$, it is necessary to determine $\boldsymbol{N}^{\lambda}$, i.e. the shape functions for the Lagrange multiplier field. In the mixed system described by Eq. 10, the secondary variable $\boldsymbol{\lambda}$ is obtained from a finite element space that is smaller than that for $\boldsymbol{u}$. Therefore, in the case of 8-node hexahedral

elements, where the primary field $\boldsymbol{u}$ is approximated using standard (tri)linear functions, we choose $\boldsymbol{N}^{\lambda}$ as constant functions. Consequently, for the 2-element mesh depicted in Fig. 6, we can express the constraint matrix defined in Eq. 9 as

$$\boldsymbol{C} = \int_{F_1} \left(\boldsymbol{N}_1^{+} - \boldsymbol{N}_1^{-}\right)^{\top} d\Omega. \tag{19}$$

Direct integration over the tangled region F_1 to compute $\boldsymbol{C}$ is computationally expensive and cumbersome [48]. Instead, we evaluate the integrand at a sample point $\boldsymbol{p} \in F_1$, i.e., evaluate the $\boldsymbol{C}$ as

$$\boldsymbol{C} = \left(\boldsymbol{N}_1^{+}(\boldsymbol{p}) - \boldsymbol{N}_1^{-}(\boldsymbol{p})\right)^{\top}. \tag{20}$$

This results in three constraint equations (for the 3D elasticity field $\boldsymbol{u}$).

Recall that any point $\boldsymbol{p}$ within the fold can be mapped to two parametric points $\boldsymbol{a}$ and $\boldsymbol{b}$ belonging to J^{-} and J^{+} regions respectively. The constraint can thus be stated as:

$$\boldsymbol{C} = \left(\boldsymbol{N}_1(\boldsymbol{b}) - \boldsymbol{N}_1(\boldsymbol{a})\right)^{\top} \tag{21}$$

The methodology for determining the two parametric points $\boldsymbol{a}$ and $\boldsymbol{b}$ is described below.

First, we focus on finding point $\boldsymbol{a}$. For a tangled element E_j, a list $\mathcal{L}$ of Gauss points with negative Jacobian is generated. For a point $\boldsymbol{a} \in \mathcal{L}$, the corresponding point $\boldsymbol{p}$ in the physical space can be obtained via:

$$\boldsymbol{p} = \boldsymbol{N}_j(\boldsymbol{a})\hat{\boldsymbol{x}}_j \tag{22}$$

where $\hat{\boldsymbol{x}}_j$ is the position vector for the nodes of the element E_j.

For non-penetrating tangled elements such as Fig. 2b, the point $\boldsymbol{p}$ corresponding to every $\boldsymbol{a} \in \mathcal{L}$ lies within the fold. However, for penetrating elements (shown in Fig. 4a), this is not always the case, i.e., the point $\boldsymbol{p}$ may not necessarily reside within the fold. To determine this, we check if $\boldsymbol{p}$ lies outside the physical boundary of the element. If a ray extending from $\boldsymbol{p}$ intersects the element's boundary an even number of times, it indicates that $\boldsymbol{p}$ lies outside the element, i.e., it resides within the folded region. Conversely, if the ray intersects the boundary an odd number of times, $\boldsymbol{p}$ does not lie within the fold. In this case, we consider another point in $\mathcal{L}$ and repeat the process. If all the points in $\mathcal{L}$ have been exhausted, we create a new list $\mathcal{L}$ with successively increasing number of Gauss points, until a point $\boldsymbol{a}$ corresponding to a fold point $\boldsymbol{p}$ is found. In practice, we found that 7^3 was sufficient for all examples considered. Note that these Gauss points are used to determine a fold point, and not for stiffness matrix computation.

Having obtained a point $\boldsymbol{p}$ within the folded region of the physical space, we can now determine the corresponding parametric point $\boldsymbol{b}$ in J^{+} using the Newton-Raphson (N-R) method. An initial guess point for $\boldsymbol{b}$ is chosen as a corner vertex of the parametric space that has a positive Jacobian and is located farthest from $\boldsymbol{a}$. Once $\boldsymbol{b} \in J^{+}$ is obtained, the constraint matrix can be readily computed using Eq. 21.

4.3. Fold Shared by Multiple Elements

When the fold is shared by multiple elements, such as in Fig. 5, where the folded region of E_1 is shared by the elements E_2, E_3 and E_4, observe that the total volume is given by:

$$\int\limits_{E_1+E_2+E_3+E_4} d\Omega = \int\limits_{C_1^+} d\Omega + \int\limits_{C_2^+} d\Omega + \int\limits_{C_3^+} d\Omega + \int\limits_{C_4^+} d\Omega - \int\limits_{C_1^-} d\Omega \tag{23}$$

i.e., the integral over the negative component must be subtracted once. Furthermore, the field compatibility equation is as before:

$$\boldsymbol{u}_1^+ - \boldsymbol{u}_1^- = 0 \quad \text{in } F_1 \tag{24}$$

Consequently, the elemental stiffness matrices, forcing vectors, and constraint matrix are computed as before: (1) standard stiffness matrices $\widehat{\boldsymbol{k}}_i$, $i = 1, \ldots, 4$ are computed for all elements using standard Gauss integration while retaining the sign of the Jacobian (the forcing vectors are computed similarly), and (2) the constraint matrix is computed by evaluating Eq. 20 at a point within the fold F_1, employing the method described above. No additional treatment is needed.

4.4. Nonlinear Analysis

The proposed i-TFEM can be easily generalized to finite elasticity problems with geometric and material non-linearities. We employ the total Lagrangian formulation to model the deformation; a detailed discussion can be found in [51, 52]. In standard FEM, the structural equilibrium can be captured via the residual force vector $\boldsymbol{R}$ as

$$\boldsymbol{R}(\hat{\boldsymbol{u}}) = \boldsymbol{f}_{int} - \boldsymbol{f}_{ext} = \boldsymbol{0}, \tag{25}$$

where $\boldsymbol{f}_{ext}$ is the external nodal load vector and $\boldsymbol{f}_{int}$ is the internal nodal load.

However, in i-TFEM, the constraint matrix $\boldsymbol{C}$ must be incorporated into the equilibrium equation (Eq. 25) via the Lagrange multipliers $\hat{\boldsymbol{\lambda}}$ as follows:

$$\widehat{\boldsymbol{R}}(\hat{\boldsymbol{u}}) + \boldsymbol{C}\hat{\boldsymbol{\lambda}} = \boldsymbol{0} \tag{26}$$

$$\boldsymbol{C}^\top \hat{\boldsymbol{u}} = \boldsymbol{0} \tag{27}$$

Here, $\widehat{\boldsymbol{R}}$ are computed in the same manner as $\boldsymbol{R}$ in the standard FEM, but including the sign of the Jacobian determinant.

Since the problem is nonlinear, the Newton-Raphson method is employed to obtain the displacement and Lagrange multiplier incrementally. The incremental displacements $\Delta\hat{\boldsymbol{u}}^n$ and Lagrange multipliers $\Delta\hat{\boldsymbol{\lambda}}^n$ are obtained in i-TFEM by solving (see [46] for details):

$$\begin{bmatrix} \widehat{\boldsymbol{K}}_t & \boldsymbol{C} \\ \boldsymbol{C}^\top & \boldsymbol{0} \end{bmatrix} \begin{Bmatrix} \Delta\hat{\boldsymbol{u}}^n \\ \Delta\hat{\boldsymbol{\lambda}}^n \end{Bmatrix} = \begin{Bmatrix} -\left(\widehat{\boldsymbol{R}} + \boldsymbol{C}\hat{\boldsymbol{\lambda}}^n\right) \\ \boldsymbol{0} \end{Bmatrix}. \tag{28}$$

where, $\widehat{\boldsymbol{K}}_t$ is the standard tangent stiffness matrix defined as

$$\widehat{\boldsymbol{K}}_t = \partial \widehat{\boldsymbol{R}}(\hat{\boldsymbol{u}}^{n-1})/\partial \hat{\boldsymbol{u}}. \tag{29}$$

The constraint matrix $\boldsymbol{C}$ can be obtained as described in Section 4.2. The displacements and Lagrange multipliers at n^{th} Newton-Raphson iteration are updated via:

$$\hat{\boldsymbol{u}}^n = \Delta\hat{\boldsymbol{u}}^n + \hat{\boldsymbol{u}}^{n-1}; \quad \hat{\boldsymbol{\lambda}}^n = \Delta\hat{\boldsymbol{\lambda}}^n + \hat{\boldsymbol{\lambda}}^{n-1}.$$

When the mesh is not tangled, i-TFEM reduces to the standard FEM. Thus, Eq. 28 reduces to:

$$\widehat{\boldsymbol{K}}_t \Delta\hat{\boldsymbol{u}}^n = -\widehat{\boldsymbol{R}}. \tag{30}$$

5. Numerical Experiments

In this section, the proposed 3D i-TFEM is demonstrated by solving benchmark linear and nonlinear elasticity problems using various synthetically generated and real-world tangled meshes. The benchmark examples presented in this section have been previously employed to validate several numerical techniques [40, 53, 54, 55, 56]. Numerical experiments are conducted under the following conditions:

- For the purpose of numerical integration, standard Gaussian quadrature ($2 \times 2 \times 2$) is employed for all the hexahedral elements. Thus, the Jacobian is computed at the corresponding Gauss points located at $\left(\pm 1/\sqrt{3},\ \pm 1/\sqrt{3},\ \pm 1/\sqrt{3}\right)$.
- In standard FEM, the absolute value of the Jacobian determinant is employed (to be consistent with commercial FEM systems such as ANSYS). Without the absolute value, and without the constraint, standard FEM can lead to non-nonsensical results [45], or result in non-convergence (see Section 5.1.4).
- Homogeneous isotropic material is assumed, unless otherwise mentioned.
- The implementation is in MATLAB R2022a, on a standard Windows 10 desktop with Intel(R) Core(TM) i9-9820X CPU running at 3.3 GHz with 16 GB memory.

The accuracy of numerical solutions is assessed using the following measures of error in the computed displacement field. The relative L^2 norm error is defined as

$$\frac{||\boldsymbol{u}^{\text{ref}} - \boldsymbol{u}||_{L^2(\Omega)}}{||\boldsymbol{u}^{\text{ref}}||_{L^2(\Omega)}} = \left[\frac{\int\limits_{\Omega} |\boldsymbol{u}^{\text{ref}} - \boldsymbol{u}|^2 \, d\Omega}{\int\limits_{\Omega} |\boldsymbol{u}^{\text{ref}}|^2 \, d\Omega}\right]^{0.5} \tag{31}$$

and the relative error in energy norm as:

$$e_E = \left[\frac{\int\limits_{\Omega} \left(\nabla\boldsymbol{u}^{\text{ref}} - \nabla\boldsymbol{u}\right)^\top \boldsymbol{D} \left(\nabla\boldsymbol{u}^{\text{ref}} - \nabla\boldsymbol{u}\right) \, d\Omega}{\int\limits_{\Omega} \left(\nabla\boldsymbol{u}^{\text{ref}}\right)^\top \boldsymbol{D} \left(\nabla\boldsymbol{u}^{\text{ref}}\right) \, d\Omega}\right]^{0.5} \tag{32}$$

where $\boldsymbol{u}^{\text{ref}}$ is the reference solution and $\boldsymbol{u}$ is the solution under consideration.

5.1. Synthetic Meshes

We consider here synthetically-generated regular (non-tangled) and tangled meshes. Specifically, consider a cubic domain $\Omega = (-1, 1)^3$ that is first discretized using $3 \times 3 \times 3$, i.e., 27 cubic elements as shown in Fig. 7a, i.e., each element side is of length $2/3$. To create a tangled mesh, we modify the mesh in two steps:

1. We convert each element of the regular mesh into a 2-element unit, as depicted in Fig. 7b, resulting in 54 elements. Positions of the new re-entrant nodes (nodes 9 and 10) are given as

$$x_3^{(9)} = x_3^{(1)}, \qquad x_i^{(9)} = x_i^{(1)} + (0.5 - d)\, s_i, \quad i = 1, 2 \tag{33a}$$

$$x_3^{(10)} = x_3^{(5)}, \qquad x_i^{(10)} = x_i^{(5)} + (0.6 - d)\, s_i, \quad i = 1, 2 \tag{33b}$$

   where s_i is the element size in i^{th} direction and the parameter d controls the extent of tangling (discussed later).

2. We then move the node B (highlighted in red in Fig. 7a), using the same parameter d, as follows:

$$\boldsymbol{x}_d^B = \boldsymbol{x}^{(B)} - d \times [4.2 s_1 \quad 1.75 s_2 \quad 0.7 s_3]^\top. \tag{34a}$$

The value of the parameter d is varied from 0 to 0.47. The front view of a resulting tangled mesh for $d = 0.4$ is illustrated in Fig. 7c.

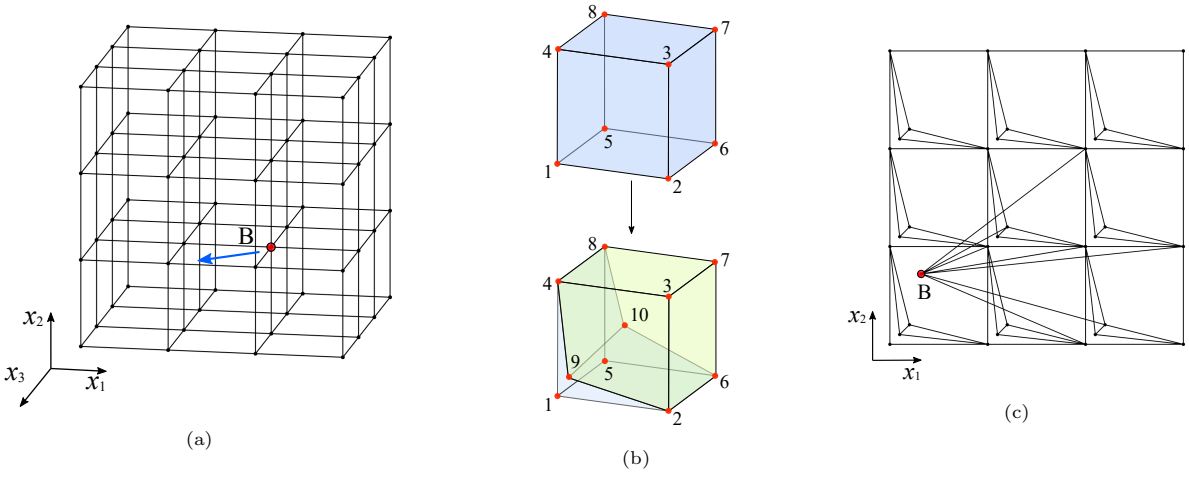


Figure 7: (a) Regular mesh (with 27 elements). (b) Non-tangled element converted to tangled element (c) Front view of the tangled mesh.

The tangled mesh (Fig. 7c) has 54 elements; out of which 31 elements are tangled. Fig. 8 illustrates some of the tangled elements. These elements are non-convex with non-planar faces. Moreover, some of the tangled elements (e.g. see Fig. 8b, 8c, 8e) exhibit penetration. Since the mesh (Fig. 7c) contains tangled elements of various configurations, it can be employed to rigorously test the effectiveness of i-TFEM.

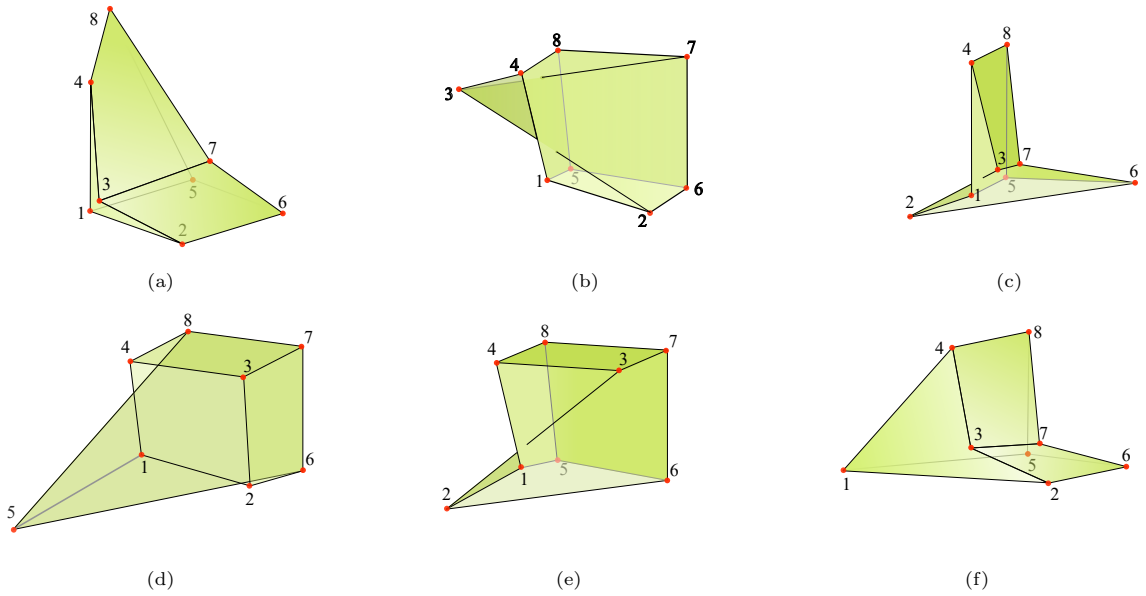


Figure 8: Some of the tangled elements present in the mesh shown in Fig. 7c.

5.1.1. Displacement Patch Test

For the patch test, an arbitrary linear field is chosen as the exact (reference) solution:

$$u_1 = 0.579x_1 + 0.246x_2 + 0.482x_3 - 0.374 \tag{35a}$$

$$u_2 = 0.486x_1 + 0.351x_2 + 0.947x_3 - 0.620 \tag{35b}$$

$$u_3 = 0.512x_1 + 0.746x_2 + 0.548x_3 - 0.480. \tag{35c}$$

Corresponding to this field, Dirichlet conditions are applied to the left face of the mesh in Fig.7c, while Neumann condition are applied to the remaining faces. The problem is solved over the tangled meshes by varying d using standard FEM and i-TFEM (the non-tangled mesh in Fig. 7a is not used for this experiment).

To compare the two methods, the L^2 and energy error norms are plotted in Fig. 9a and Fig. 9b respectively. Observe that, both FEM and i-TFEM provide machine precision accuracy for $0 \leq d \leq 0.15$, as the Jacobian at all the Gauss points remains positive (indicating that the mesh is not tangled and i-TFEM reduces to FEM). However, for $d \geq 0.2$, the Jacobian determinant becomes negative at one or more Gauss points, leading to a significant error in FEM. In contrast, i-TFEM consistently achieves machine precision accuracy for all values of d. In other words, i-TFEM successfully passes this patch test over the tangled mesh, while standard FEM does not.

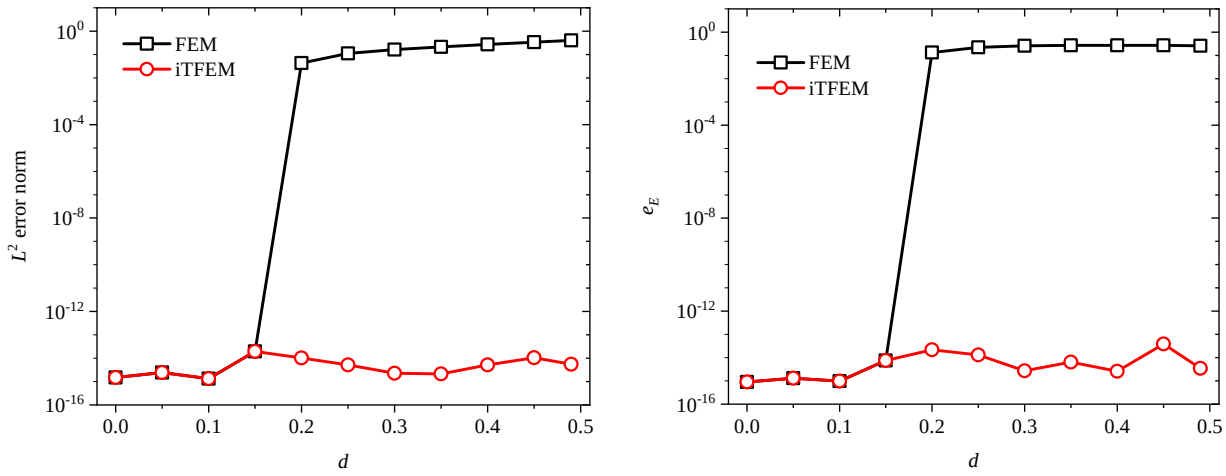


Figure 9: Patch test for FEM and i-TFEM: (a) L^2 error norm and (b) energy error norm.

5.1.2. Strain energy convergence: Cubic cantilever

In this experiment, we analyze i-TFEM for its convergence characteristic with regard to the strain energy. We consider a cubic cantilever (see Fig. 10) subjected to a uniform pressure $p = 1$ on its upper face [53], and fixed over the face $x_1 = 0$. The material is assumed to be linear elastic, with Young's modulus $E = 1$ and Poisson's ratio $\nu = 0.25$.

The exact solution to the problem is unknown. However, a reference solution is reported in [53], obtained using standard FEM with a very fine mesh consisting of 30,204 nodes and 20,675 ten-node tetrahedral elements. The reference strain energy is 0.9486, and the vertical deflection of 3.3912 at point A, located at (1, 1, 0).

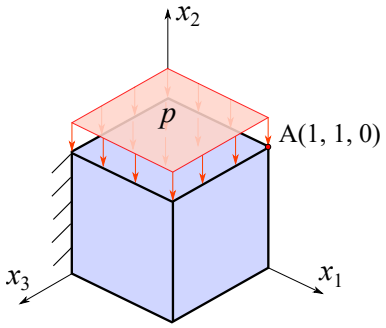


Figure 10: Cubic cantilever subjected to uniform pressure.

We solve this problem using both standard FEM and i-TFEM on tangled and non-tangled meshes of various sizes. Regular and tangled meshes are constructed by repeating the units described previously (Fig. 7a for regular mesh and Fig. 7c for tangled mesh). The regular and tangled repeating units (containing 27 and 54 elements respectively) are stacked in a $n_r \times n_r \times n_r$ configuration, where n_r denotes the number of repeating units in each direction; n_r is varied from 1 to 8 to study the convergence characteristics. For

the tangled case, the extent of tangling was fixed by setting $d = 0.4$.

Fig. 11a shows the strain energy convergence for i-TFEM in comparison with FEM over tangled and regular (non-tangled) mesh. It is evident that the strain energy obtained using i-TFEM converges to the reference solution, unlike FEM over the tangled mesh.

Similarly, Fig. 11b illustrates the convergence of the tip deflection at point A(1, 1, 0) obtained using the i-TFEM and FEM. Once again, observe that i-TFEM converges to the reference solution for the tangled meshes, while FEM fails to do so.

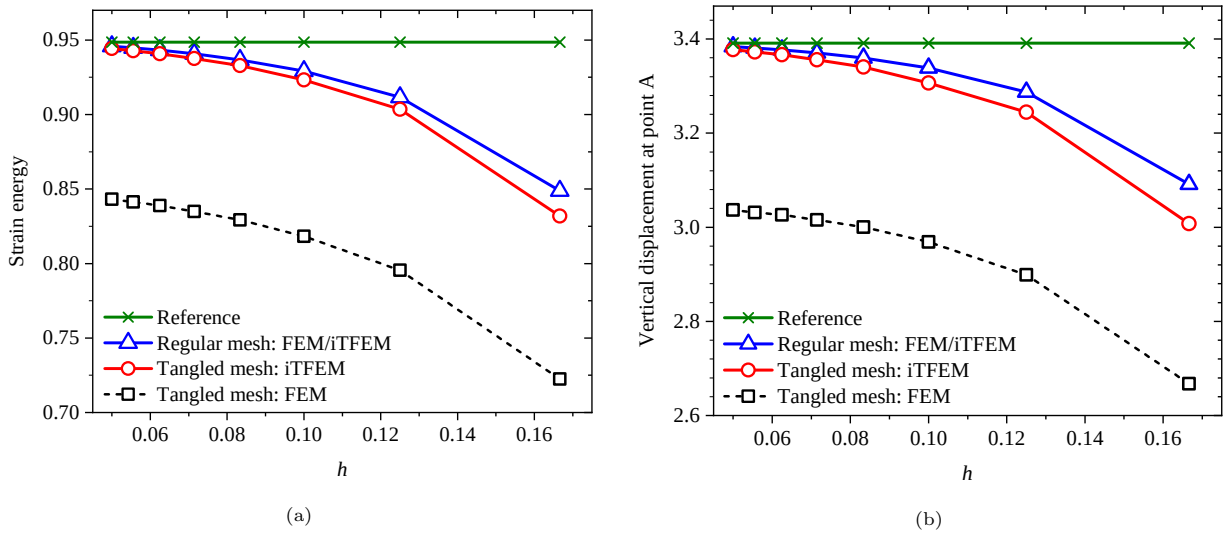


Figure 11: Convergence of (a) strain energy and (b) vertical displacement at point A for the cubic cantilever problem.

In the above convergence study, the extent of tangling was fixed with $d = 0.4$. Here, we study the effect of tangling on the computed solutions by varying d from 0 to 0.47 for a fixed mesh size, with $n_r = 3$. As seen in the previous experiment, as d increases, tangling increases and FEM deviates from the expected results, as opposed to i-TFEM (see Fig. 12).

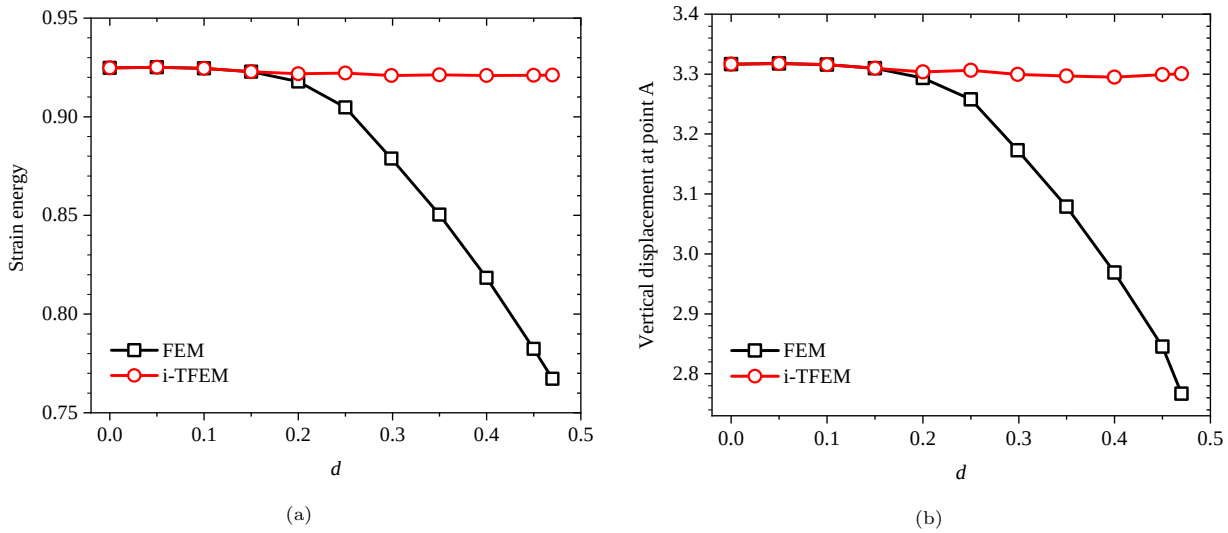


Figure 12: Effect of tangling on (a) strain energy and (b) vertical displacement at point A for the cubic cantilever problem.

5.1.3. Convergence study: Beam under torsion

Next, we study the performance of i-TFEM for a cantilever beam loaded in torsion [40]. The domain for this problem is $(-1,1)\times(-1,1)\times(0,10)$; the material is linear isotropic with Young's modulus $E=25$ and Poisson's ratio $\nu=0.3$. Dirichlet boundary conditions are applied to the faces at $x_3=0$ and $x_3=10$. The expressions for the stresses are [57]:

$$\begin{aligned}
\sigma_{11} &= \sigma_{22} = \sigma_{12} = \sigma_{33} = 0 \\
\sigma_{31} &= \frac{8E\beta}{\pi^2(1+\nu)}\sum_{n=1}^{\infty}\frac{(-1)^n}{(2n-1)^2\cosh\left[(2n-1)\pi/2\right]}\cos\left[(2n-1)\pi x_1/2\right]\sinh\left[(2n-1)\pi x_2/2\right] \\
\sigma_{(23)} &= \frac{E\beta}{2(1+\nu)}\left(2x_1+\sum_{n=1}^{\infty}\frac{16(-1)^n}{\pi^2(2n-1)^2\cosh\left[(2n-1)\pi/2\right]}\sin\left[(2n-1)\pi x_1/2\right]\cosh\left[(2n-1)\pi x_2/2\right]\right)
\end{aligned} \tag{36}$$

and displacements, up to rigid body motion, are given by

$$\begin{aligned}
u_1 &= -\beta x_2 x_3, \quad u_2 = \beta x_3 x_1, \\
u_3 &= \beta\left(x_1x_2+\sum_{n=1}^{\infty}\frac{32(-1)^n}{\pi^3(2n-1)^3\cosh\left[(2n-1)\pi/2\right]}\sin\left[(2n-1)\pi x_1/2\right]\sinh\left[(2n-1)\pi x_2/2\right]\right).
\end{aligned} \tag{37}$$

Here $\beta=0.1$ is the twist per unit length which is proportional to the applied torque.

As in the previous examples, we create a tangled mesh ($d=0.4$) and a regular mesh using their respective repeating units. In this particular problem, we stack repeating units in a configuration of $n_r\times n_r\times 5n_r$. To visualize the deformation and the stress field resulting from the torsional load, we present representative i-TFEM results for the tangled mesh with $n_r=4$ in Fig. 13.

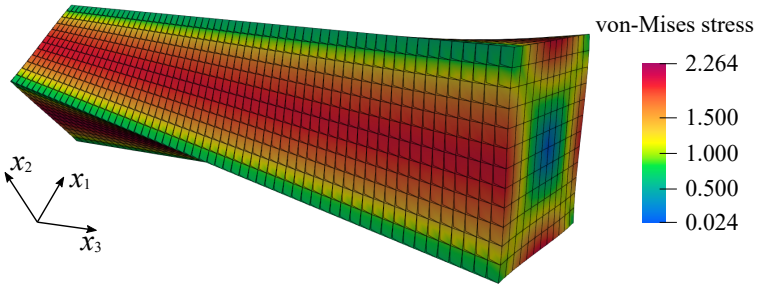
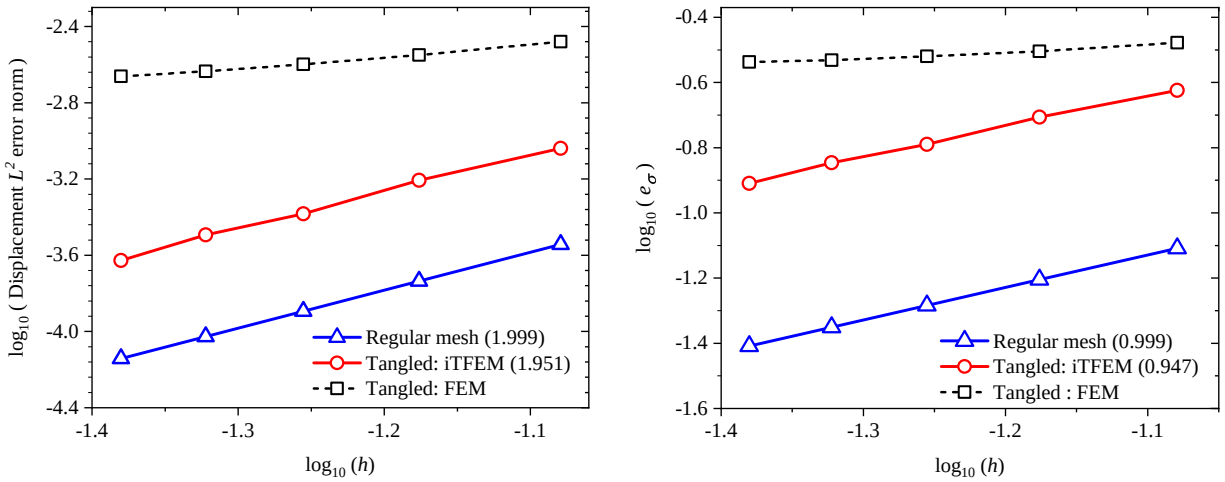


Figure 13: Deformed configuration of the beam under torsion.

We study the convergence characteristics of i-TFEM by employing displacement L^2 error norm defined in Eq. 31. In addition, an error measure for the stress field ($e_{\boldsymbol{\sigma}}$) is employed and defined as (to be consistent with [40]):

$$e_{\boldsymbol{\sigma}} = \frac{||\boldsymbol{\sigma}^{\text{ref}} - \boldsymbol{\sigma}||_{L^2(\Omega)}}{||\boldsymbol{\sigma}^{\text{ref}}||_{L^2(\Omega)}}. \tag{38}$$

As evident from the convergence plots in Fig. 14, we have close to second-order convergence in displacements and first-order convergence in stresses, similar to the regular meshes. On the other hand, FEM over tangled mesh fails to converge.



(a) (b)

Figure 14: Convergence of (a) displacement error and (b) stress error for the torsion problem.

5.1.4. Large deformation analysis

To evaluate the performance of i-TFEM in the presence of geometric and material nonlinearities, we consider the cantilever beam problem [53] occupying the domain $(-10, 10) \times (-1, 1) \times (0, 2)$, subjected to a uniformly distributed vertical load $p = 8000$ applied in 5 load steps. The material is assumed to be hyperelastic; specifically, we employ the compressible isotropic generalized neo-Hookean material model

where the strain energy density is given by [51, 52]:

$$\Psi_{\text{GNH}}(\boldsymbol{u}) = \frac{\mu}{2}\left(J_F^{-2/3} tr\boldsymbol{B} - 3\right) + \frac{K}{2}\left(J_F - 1\right)^2. \tag{39}$$

where $\mu = 1.154 \times 10^7$ and $K = 2.5 \times 10^7$ are the material parameters (equivalent to shear and bulk moduli respectively in the small strain limit). Further, $J_F = det\boldsymbol{F}$, $\boldsymbol{B} = \boldsymbol{F}\boldsymbol{F}^\top$ is the left Cauchy-Green deformation tensor, and $\boldsymbol{F}$ is the deformation gradient.

Regular and tangled meshes ($d = 0.4$) are created by stacking the respective repeating units in the $10n_r \times n_r \times n_r$ arrangement. The deformed configuration for tangled mesh with $n_r = 1$ obtained via i-TFEM is shown in Fig. 15a. A convergence study is then carried out by varying n_r. The vertical displacement u_2 at the top-right corner point A for the two methods is plotted against the mesh index n_r in Fig. 15b. For the tangled mesh, similar to the linear elasticity problems presented earlier, i-TFEM converges to the expected solution while standard FEM converges to an incorrect solution.

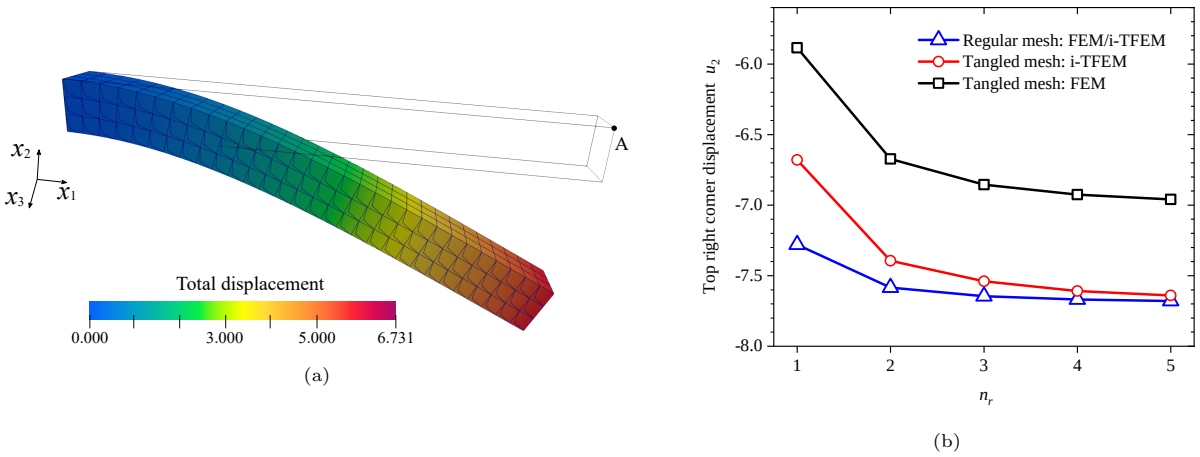


Figure 15: (a) Deformed configuration of the cantilever beam with tangled mesh using i-TFEM. (b) Convergence study.

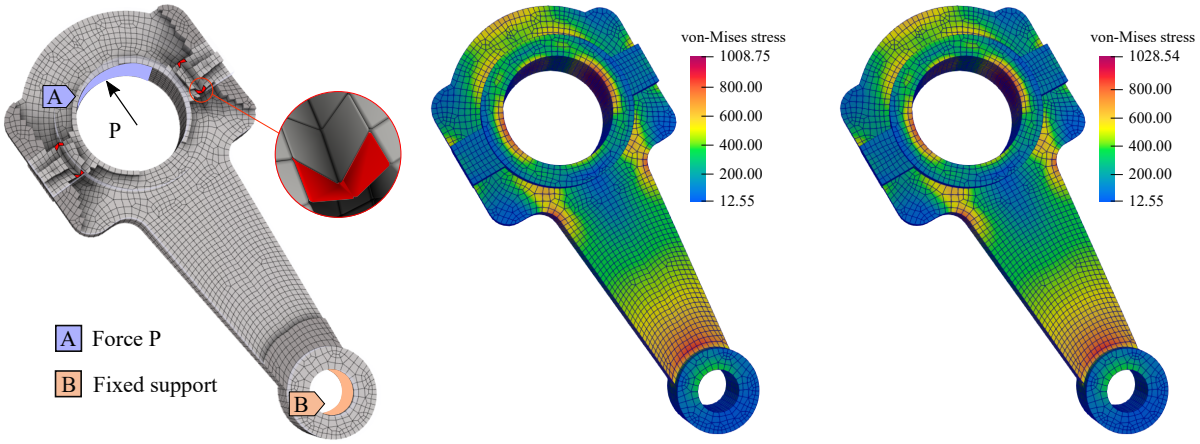
5.2. Real-World Meshes

In this section, the performance of i-TFEM is evaluated over real-world tangled meshes generated by the state-of-the-art mesh-generating algorithms.

5.2.1. Connecting rod

We now consider a connecting rod mesh (mesh data obtained from [5]) illustrated in Fig. 16a. Out of the total 11316 hexahedral elements of the mesh, 16 elements are tangled; some are highlighted in Fig. 16a. Though it is possible to untangle this mesh [5], i-TFEM completely eliminates the need for untangling. Here, we compare the results for tangled and untangled meshes.

A linear quasi-static elasticity problem is set up as shown in Fig. 16a: an axial load of $P = 300$N is applied on one end, while the other end is fixed. The material properties are as follows: Young's modulus $E = 2.05 \times 10^7$ and Poisson's ratio $\nu = 0.28$.



(a) (b) (c)

Figure 16: (a) Connecting rod mesh [5] with boundary conditions; elements in red color are tangled. (b) von-Mises stress field for tangled mesh using i-TFEM; (b) von-Mises stress field for untangled mesh.

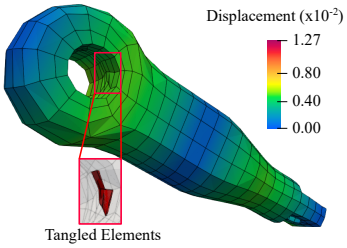
Fig. 16b and Fig. 16c illustrate the von-Mises stress field for the tangled mesh (using i-TFEM) and untangled mesh respectively. The total computational time was 10.47s for the tangled mesh using i-TFEM, and 10.16 for untangled mesh, i.e., the overhead due to i-TFEM was found to be minimal.

5.2.2. Synthetic Solution

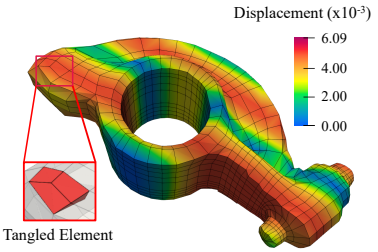
We now consider additional real-world tangled meshes. To evaluate the performance of i-TFEM, the following synthetic solution is used:

$$\boldsymbol{u} = \left[\zeta_1^3\zeta_2\zeta_3^2 + 2\zeta_1^2\zeta_2^3\zeta_3^2 + 0.5\sin(2\pi\zeta_1);\quad \zeta_1^2\zeta_2^3\zeta_3 + 2\zeta_1^2\zeta_2^2\zeta_3^3;\quad \zeta_1\zeta_2^2\zeta_3^3 + 2\zeta_1^3\zeta_2^2\zeta_3^2\right]^\top /100 \tag{40}$$

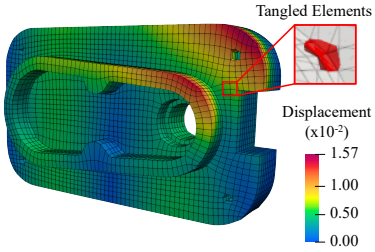
where ζ_i are computed by dividing each component x_i by its corresponding length-scale L_i, i.e., $\zeta_i = x_i/L_i$. Given the analytical solution, stresses can be computed as $\sigma_{ij} = \mathcal{C}_{ijkl}\varepsilon_{ij}$ where $\varepsilon_{ij} = 0.5(u_{i,j} + u_{j,i})$ and $\mathcal{C}_{ijkl}$ is the elasticity tensor. The appropriate body force $\boldsymbol{b}$ can thus be computed by employing the equilibrium equation: $\sigma_{ij,j} + b_i = 0$. The problem is solved using this body force with the material properties, $E = 400/3$, $\nu = 1/3$. The dirichlet boundary condition is imposed over the entire boundary. The resultant displacement fields are shown in Fig. 17 for various tangled meshes. These tangled meshes (available at hexalab mesh repository [58]) are produced from a range of state-of-the-art mesh generators [7, 8, 11, 58]. For instance, the mesh in Fig. 17c is generated using a PolyCube based approach [11]. On the other hand, the tangled mesh in Fig. 17d is created using frame-field based approach [58]. Fig. 17b is produced semi-manually using dual-sheet modeling [8] while automatic block decomposition (LoopyCuts) [7] is employed in Fig. 17a, Fig. 17e and Fig. 17f. As one can observe, despite numerous tangling elements, the L^2 error is within the acceptable/nominal range. Note that, in the meshing literature, the 'scaled Jacobian' [49] computed at the nodes is used, and is therefore reported in Fig. 17 as well.



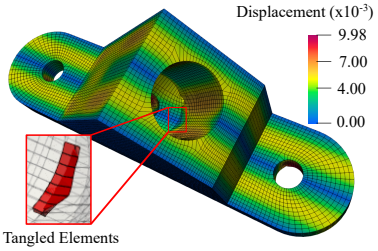
(a) Rod [7]: Rel. L^2 Err $= 4.89 \times 10^{-2}$
Min Scaled Jacobian = -0.593



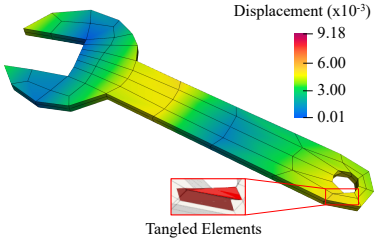
(b) Rocker arm [8]: Rel. L^2 Err $= 2.37 \times 10^{-2}$
Min Scaled Jacobian = -0.189



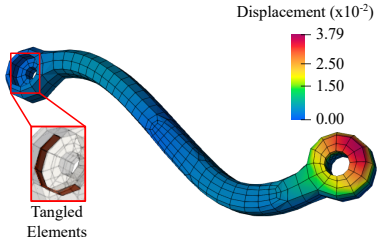
(c) Demo09 [59]: Rel. L^2 Err $= 4.00 \times 10^{-2}$
Min Scaled Jacobian = -0.790



(d) i02u_m2 [58]: Rel. L^2 Err $= 4.42 \times 10^{-1}$
Min Scaled Jacobian = -0.412



(e) wrench [7]: Rel. L^2 Err $= 9.95 \times 10^{-2}$
Min Scaled Jacobian = -0.280



(f) lever arm [7]: Rel. L^2 Err $= 6.31 \times 10^{-2}$
Min Scaled Jacobian = -0.944

Figure 17: Displacement plots obtained via i-TFEM for the practically occurring tangled meshes. The red elements in the inset are tangled.

Next, we compare the real-world tangled meshes with their corresponding untangled meshes provided in [5]. The problem with the synthetic solution (Eq. 40) described above is solved using the two methods: i-TFEM for tangled meshes and standard FEM for untangled meshes. The results, presented in Table 1, reveal that the L^2 error norm obtained via i-TFEM over tangled meshes is comparable with that obtained over the corresponding untangled meshes. Moreover, the time required to handle tangled elements is minimal. This suggests that i-TFEM can provide solutions with comparable accuracy over tangled meshes, thus eliminating the need for mesh untangling.

Table 1: Comparison of solutions over tangled and untangled meshes provided in [5].

Model [5]	Min. Scaled Jacobian		No. of Hexahedra	Relative L^2 error		Time (second)	
	Tangled	Untangled	Tangled/Total	Tangled	Untangled	Tangled	Untangled
cap	-0.94	0.11	19/4420	4.23×10^{-2}	4.36×10^{-2}	4.73	4.13
block	-0.70	0.25	10/2520	1.58×10^{-2}	1.58×10^{-2}	2.02	1.87
bust	-0.60	0.11	6/5258	3.71×10^{-2}	3.93×10^{-2}	5.15	5.11
linking rod	-0.39	0.55	8/11316	2.13×10^{-3}	2.12×10^{-3}	10.46	10.17

6. Mesh Quality Indicators for Tangled Elements

The goal of a good quality indicator is to provide insights into the accuracy of numerical methods [60, 61]. Since it is generally assumed that higher mesh quality leads to improved accuracy in FEM solutions, mesh generators and optimizers strive to maximize mesh quality. Numerous quality indicators have been developed such as the Jacobian, Oddy metric, aspect ratio, skewness, taper, stretch, warpage (see [60, 49] for details). For higher order elements too, various quality measures have been developed that account for deviation in shape, size, skewness, degeneracy and so on [62, 63, 64, 65].

The scaled Jacobian is one of the most extensively used mesh quality indicators in linear [7, 37, 38, 66, 67] and higher-order finite elements [68, 69, 64, 62, 70, 71, 65] as well as spline-based discretizations used in isogeometric analysis [72, 73, 74, 75]. To compute the 3×3 Jacobian matrix for 8-node element, different approaches are taken by the geometric processing/meshing and numerical analysis/FEM communities. In the meshing community, Jacobian matrix is evaluated at each corner of the hexahedral element using the three ordered normalized edge vectors [76]. On the other hand, in the FEM community, the Jacobian matrix is evaluated at each quadrature (Gauss) point using the finite element basis functions. This is because, the standard FEM gives invalid results when negative Jacobian is encountered at one or more of its quadrature (Gauss) points, rather than corners. Hence, in this paper too, the Jacobian matrix is evaluated at the Gauss points using the finite element basis functions.

Traditionally, tangled elements are deemed to be invalid, and/or assigned a quality of zero [49]. Nonetheless, using i-TFEM, tangled meshes can provide comparable or even superior accuracy compared to regular meshes. To illustrate, let us consider the torsion problem described in Section 5.1.3. The beam is discretized using a 2-element tangled repeating unit described in Fig. 7. The positions of nodes 9 and 10 in Fig. 7 are varied (but point B is kept stationary for simplicity). These repeating units, each with 54 elements, are arranged in a $2 \times 2 \times 10$ configuration, resulting in a total of 270 elements. To assess mesh quality, the determinant of the Jacobian is computed at each Gauss point and divided by the determinant of the Jacobian of a perfect hexahedron (for scaling purposes). We then solve the torsion problem and evaluate the error.

In Fig. 18a and Fig. 18b, we plot the relative L^2 norm error in displacement and stress respectively as a function of the minimum Jacobian within the mesh. As one can observe, accuracy improves as we move away from zero Jacobian, whether in the positive or negative direction. Similar observations can be drawn from the data provided in Table 1. Fig. 18c illustrates the condition number of the i-TFEM stiffness matrix as a function of the minimum Jacobian; as one can observe the condition number remains stable.

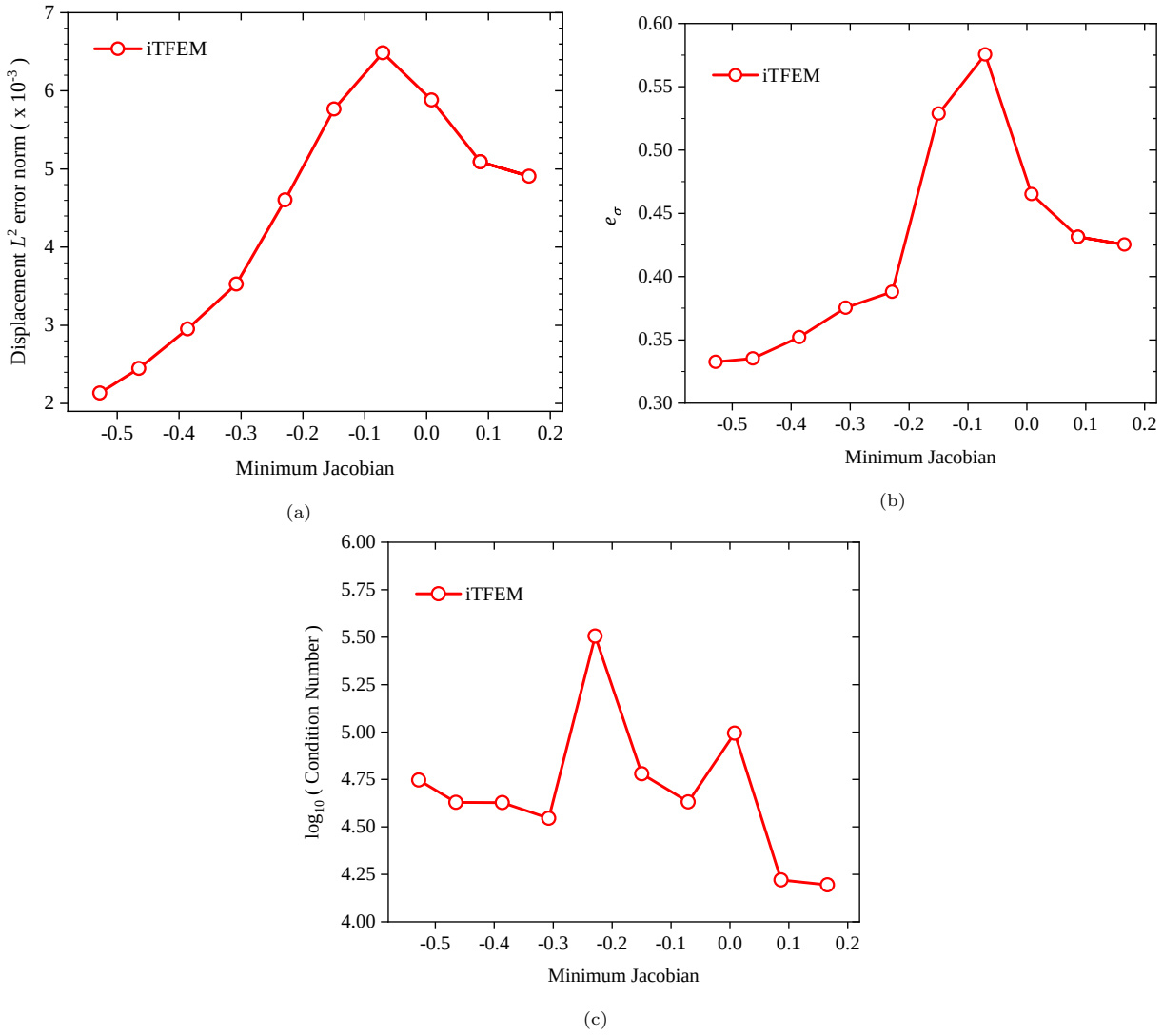


Figure 18: Plot of (a) displacement error (b) stress error, and (c) condition number vs minimum Jacobian.

This suggests that one must reevaluate mesh quality indicators to accommodate tangled elements. One potential indicator, for instance, could be the *absolute* value of the Jacobian instead of the signed value. In other words, we can eliminate the tangle-free constraint on mesh generators (see Section 1). Further research is needed since these findings are specific to the problem and mesh considered.

7. Conclusions

This paper addresses the challenge of handling *tangled* meshes that are conventionally considered unacceptable due to the erroneous results they produce. A 3D isoparametric tangled FEM (i-TFEM) was proposed to handle tangled hexahedral elements.

To implement i-TFEM, only two changes need to be made to an existing FEM code: (a) the Jacobian determinant sign must be retained, and (b) a constraint equation must be included. The resulting linear system is symmetric, and i-TFEM reduces to standard FEM when the mesh is regular.

Further, through a series of examples involving linear and nonlinear elasticity, we demonstrated the robustness and efficiency of i-TFEM. The numerical experiments demonstrate that i-TFEM passes the patch test, and provides optimal convergence rate over severely tangled meshes. The experiments utilized numerous real-world tangled meshes, demonstrating the viability of the method in practical scenarios. Finally, the additional computational time required to implement i-TFEM was found to be minimal.

The results emphasize the need for a new definition of mesh quality that accommodates tangled elements, and potentially a new class of mesh generators. Further research relating mesh quality and solution accuracy [66, 77] is required in the context of tangled meshes.

Meshes containing fully inverted elements were not addressed in this work, but will be addressed in the future. We are currently extending i-TFEM to structural dynamics, thermo-elasticity, and fluid-structure interaction. Additionally, we plan to extend i-TFEM to higher order 3D elements [78, 79, 80, 71] as well as isogeometric analysis [81] where tangling is known to occur [82, 83, 84], and leads to erroneous results.

Compliance with ethical standards

The authors declare that they have no conflict of interest.

Acknowledgments

The authors would like to thank the support of the National Science Foundation through grant CMMI 1715970, and the U. S. Office of Naval Research under PANTHER award number N00014-21-1-2916 through Dr. Timothy Bentley.

References

[1] T. Blacker, Automated conformal hexahedral meshing constraints, challenges and opportunities, Engineering with Computers 17 (3) (2001) 201–210.

[2] N. Pietroni, M. Campen, A. Sheffer, G. Cherchi, D. Bommes, X. Gao, R. Scateni, F. Ledoux, J.-F. Remacle, M. Livesu, Hex-mesh generation and processing: a survey, ACM Transactions on Graphics (TOG) 44 (2) (2022) 1–44.

[3] M. Reberol, K. Verhetsel, F. Henrotte, D. Bommes, J.-F. Remacle, Robust topological construction of all-hexahedral boundary layer meshes, ACM Transactions on Mathematical Software 49 (1) (2023) 1–32.

[4] J. F. Shepherd, C. R. Johnson, Hexahedral mesh generation constraints, Engineering with Computers 24 (3) (2008) 195–213.

[5] M. Livesu, A. Sheffer, N. Vining, M. Tarini, Practical hex-mesh optimization via edge-cone rectification, ACM Transactions on Graphics (TOG) 34 (4) (2015) 1–11.

[6] M. Mandad, R. Chen, D. Bommes, M. Campen, Intrinsic mixed-integer polycubes for hexahedral meshing, Computer Aided Geometric Design 94 (2022) 102078.

[7] M. Livesu, N. Pietroni, E. Puppo, A. Sheffer, P. Cignoni, Loopycuts: Practical feature-preserving block decomposition for strongly hex-dominant meshing, ACM Transactions on Graphics (TOG) 39 (4) (2020) 121–1.

[8] K. Takayama, Dual sheet meshing: An interactive approach to robust hexahedralization, in: Computer graphics forum, Vol. 38, Wiley Online Library, 2019, pp. 37–48.

[9] H. Liu, P. Zhang, E. Chien, J. Solomon, D. Bommes, Singularity-constrained octahedral fields for hexahedral meshing., ACM Trans. Graph. 37 (4) (2018) 93–1.

[10] X. Fang, W. Xu, H. Bao, J. Huang, All-hex meshing using closed-form induced polycube, ACM Transactions on Graphics (TOG) 35 (4) (2016) 1–9.

[11] M. Livesu, N. Vining, A. Sheffer, J. Gregson, R. Scateni, Polycut: Monotone graph-cuts for polycube base-complex construction, ACM Transactions on Graphics (TOG) 32 (6) (2013) 1–12.

[12] T. Jiang, J. Huang, Y. Wang, Y. Tong, H. Bao, Frame field singularity correctionfor automatic hexahedralization, IEEE Transactions on Visualization and Computer Graphics 20 (8) (2013) 1189–1199.

[13] Y. Li, Y. Liu, W. Xu, W. Wang, B. Guo, All-hex meshing using singularity-restricted field, ACM Transactions on Graphics (TOG) 31 (6) (2012) 1–11.

[14] J. Gregson, A. Sheffer, E. Zhang, All-hex mesh generation via volumetric polycube deformation, in: Computer graphics forum, Vol. 30, Wiley Online Library, 2011, pp. 1407–1416.

[15] M. Nieser, U. Reitebuch, K. Polthier, Cubecover–parameterization of 3d volumes, in: Computer graphics forum, Vol. 30, Wiley Online Library, 2011, pp. 1397–1406.

[16] J. Huang, Y. Tong, H. Wei, H. Bao, Boundary aligned smooth 3d cross-frame field, ACM transactions on graphics (TOG) 30 (6) (2011) 1–8.

[17] Y. Zhang, C. Bajaj, Adaptive and quality quadrilateral/hexahedral meshing from volumetric data, Computer methods in applied mechanics and engineering 195 (9-12) (2006) 942–960.

[18] Y. Zhang, T. J. Hughes, C. L. Bajaj, An automatic 3d mesh generation method for domains with multiple materials, Computer methods in applied mechanics and engineering 199 (5-8) (2010) 405–415.

[19] J. Qian, Y. Zhang, Automatic unstructured all-hexahedral mesh generation from b-reps for non-manifold cad assemblies, Engineering with Computers 28 (2012) 345–359.

[20] Y. Yu, X. Wei, A. Li, J. G. Liu, J. He, Y. J. Zhang, Hexgen and hex2spline: polycube-based hexahedral mesh generation and spline modeling for isogeometric analysis applications in ls-dyna, in: Geometric Challenges in Isogeometric Analysis, Springer, 2022, pp. 333–363.

[21] J. H.-C. Lu, W. R. Quadros, K. Shimada, Evaluation of user-guided semi-automatic decomposition tool for hexahedral mesh generation, Journal of Computational Design and Engineering 4 (4) (2017) 330–338.

[22] L. Maréchal, Advances in octree-based all-hexahedral mesh generation: handling sharp features, in: Proceedings of the 18th international meshing roundtable, Springer, 2009, pp. 65–84.

[23] Y. Zhang, X. Liang, G. Xu, A robust 2-refinement algorithm in octree or rhombic dodecahedral tree based all-hexahedral mesh generation, Computer Methods in Applied Mechanics and Engineering 256 (2013) 88–100.

[24] M. Livesu, L. Pitzalis, G. Cherchi, Optimal dual schemes for adaptive grid based hexmeshing, ACM Transactions on Graphics (TOG) 41 (2) (2021) 1–14.

[25] X. Gao, H. Shen, D. Panozzo, Feature preserving octree-based hexahedral meshing, in: Computer graphics forum, Vol. 38, Wiley Online Library, 2019, pp. 135–149.

[26] H. Zhang, G. Zhao, X. Ma, Adaptive generation of hexahedral element mesh using an improved grid-based method, Computer-Aided Design 39 (10) (2007) 914–928.

[27] Cubit, `https://cubit.sandia.gov/` (2022).

[28] Cubit, `https://www.spatial.com/products/3d-precise-mesh`.

[29] Y. Zhang, C. Bajaj, G. Xu, Surface smoothing and quality improvement of quadrilateral/hexahedral meshes with geometric flow, Communications in Numerical Methods in Engineering 25 (1) (2009) 1–18.

[30] J. Qian, Y. Zhang, W. Wang, A. C. Lewis, M. S. Qidwai, A. B. Geltmacher, Quality improvement of non-manifold hexahedral meshes for critical feature determination of microstructure materials, International journal for numerical methods in engineering 82 (11) (2010) 1406–1423.

[31] S. J. Owen, T. R. Shelton, Evaluation of grid-based hex meshes for solid mechanics, Engineering with Computers 31 (3) (2015) 529–543.

[32] P. M. Knupp, A method for hexahedral mesh shape optimization, International journal for numerical methods in engineering 58 (2) (2003) 319–332.

[33] S. Leger, A. Pepin, An updated lagrangian method with error estimation and adaptive remeshing for very large deformation elasticity problems: The three-dimensional case, Computer Methods in Applied Mechanics and Engineering 309 (2016) 1–18.

[34] V. Vavourakis, D. Loukidis, D. C. Charmpis, P. Papanastasiou, Assessment of remeshing and remapping strategies for large deformation elastoplastic finite element analysis, Computers & Structures 114 (2013) 133–146.

[35] M. L. Staten, S. J. Owen, S. M. Shontz, A. G. Salinger, T. S. Coffey, A comparison of mesh morphing methods for 3d shape optimization, in: Proceedings of the 20th international meshing roundtable, Springer, 2011, pp. 293–311.

[36] Q. Huang, W.-X. Zhang, Q. Wang, L. Liu, X.-M. Fu, Untangling all-hex meshes via adaptive boundary optimization, Graphical Models 121 (2022) 101136.

[37] K. Xu, X. Gao, G. Chen, Hexahedral mesh quality improvement via edge-angle optimization, Computers & Graphics 70 (2018) 17–27.

[38] M. N. Akram, L. Si, G. Chen, An embedded polygon strategy for quality improvement of 2d quadrilateral meshes with boundaries., in: VISIGRAPP (1: GRAPP), 2021, pp. 177–184.

[39] L. Beirão da Veiga, F. Brezzi, A. Cangiani, G. Manzini, L. D. Marini, A. Russo, Basic principles of virtual element methods, Mathematical Models and Methods in Applied Sciences 23 (01) (2013) 199–214.

[40] A. L. Gain, C. Talischi, G. H. Paulino, On the virtual element method for three-dimensional linear elasticity problems on arbitrary polyhedral meshes, Computer Methods in Applied Mechanics and Engineering 282 (2014) 132–160.

[41] G. Liu, K. Dai, T. T. Nguyen, A smoothed finite element method for mechanics problems, Computational Mechanics 39 (6) (2007) 859–877.

[42] M. S. Floater, Mean value coordinates, Computer aided geometric design 20 (1) (2003) 19–27.

[43] P.-L. Zhou, S. Cen, J.-B. Huang, C.-F. Li, Q. Zhang, An unsymmetric 8-node hexahedral element with high distortion tolerance, International Journal for Numerical Methods in Engineering 109 (8) (2017) 1130–1158.

[44] S. Nguyen-Hoang, D. Sohn, H.-G. Kim, A new polyhedral element for the analysis of hexahedral-dominant finite element models and its application to nonlinear solid mechanics problems, Computer Methods in Applied Mechanics and Engineering 324 (2017) 248–277.

[45] B. Prabhune, K. Suresh, A computationally efficient isoparametric tangled finite element method for handling inverted quadrilateral and hexahedral elements, Computer Methods in Applied Mechanics and Engineering 405 (2023) 115897.

[46] B. Prabhune, K. Suresh, Isoparametric tangled finite element method for nonlinear elasticity, arXiv preprint arXiv:2303.10799 (2023).

[47] B. Prabhune, K. Suresh, Towards tangled finite element analysis over partially inverted hexahedral elements, arXiv preprint arXiv:2207.03905 (2022).

[48] B. Prabhune, S. Sridhara, K. Suresh, Tangled finite element method for handling concave elements in quadrilateral meshes, International Journal for Numerical Methods in Engineering 123 (7) (2022) 1576–1605.

[49] P. M. Knupp, Algebraic mesh quality metrics for unstructured initial meshes, Finite Elements in Analysis and Design 39 (3) (2003) 217–241.

[50] O. C. Zienkiewicz, R. L. Taylor, J. Z. Zhu, The finite element method: its basis and fundamentals, Elsevier, 2005.

[51] O. C. Zienkiewicz, R. L. Taylor, R. L. Taylor, The finite element method: solid mechanics, Vol. 2, Butterworth-heinemann, 2000.

[52] A. F. Bower, Applied mechanics of solids, CRC press, 2009.

[53] T. Nguyen-Thoi, G. Liu, K. Lam, G. Zhang, A face-based smoothed finite element method (fs-fem) for 3d linear and geometrically non-linear solid mechanics problems using 4-node tetrahedral elements, International journal for numerical methods in Engineering 78 (3) (2009) 324–353.

[54] J. E. Bishop, A displacement-based finite element formulation for general polyhedra using harmonic shape functions, International Journal for Numerical Methods in Engineering 97 (1) (2014) 1–31.

[55] C. Lee, I. V. Singh, S. Natarajan, A cell-based smoothed finite-element method for gradient elasticity, Engineering with Computers 39 (1) (2023) 925–942.

[56] A. Francis, S. Natarajan, E. Atroshchenko, B. Lévy, S. P. Bordas, A one point integration rule over star convex polytopes, Computers & Structures 215 (2019) 43–64.

[57] J. R. Barber, Elasticity, Springer, 2002.

[58] P.-A. Beaufort, M. Reberol, D. Kalmykov, H. Liu, F. Ledoux, D. Bommes, Hex me if you can, in: Computer graphics forum, Vol. 41, Wiley Online Library, 2022, pp. 125–134.

[59] H.-X. Guo, X. Liu, D.-M. Yan, Y. Liu, Cut-enhanced polycube-maps for feature-aware all-hex meshing, ACM Transactions on Graphics (TOG) 39 (4) (2020) 106–1.

[60] T. Sorgente, S. Biasotti, G. Manzini, M. Spagnuolo, A survey of indicators for mesh quality assessment, in: Computer Graphics Forum, Vol. 42, Wiley Online Library, 2023, pp. 461–483.

[61] J. R. Shewchuk, What is a good linear element? interpolation, conditioning, and quality measures., in: IMR, 2002, pp. 115–126.

[62] A. Gargallo-Peiró, X. Roca, J. Peraire, J. Sarrate, Optimization of a regularized distortion measure to generate curved high-order unstructured tetrahedral meshes, International Journal for Numerical Methods in Engineering 103 (5) (2015) 342–363.

[63] A. Gargallo-Peiró, X. Roca, J. Peraire, J. Sarrate, A distortion measure to validate and generate curved high-order meshes on cad surfaces with independence of parameterization, International Journal for Numerical Methods in Engineering 106 (13) (2016) 1100–1130.

[64] R. Poya, R. Sevilla, A. J. Gil, A unified approach for a posteriori high-order curved mesh generation using solid mechanics, Computational Mechanics 58 (2016) 457–490.

[65] A. Johnen, C. Geuzaine, T. Toulorge, J.-F. Remacle, Quality measures for curvilinear finite elements, TILDA: Towards Industrial LES/DNS in Aeronautics: Paving the Way for Future Accurate CFD-Results of the H2020 Research Project TILDA, Funded by the European Union, 2015-2018 (2021) 221–244.

[66] X. Gao, J. Huang, K. Xu, Z. Pan, Z. Deng, G. Chen, Evaluating hex-mesh quality metrics via correlation analysis, in: Computer Graphics Forum, Vol. 36, Wiley Online Library, 2017, pp. 105–116.

[67] X.-M. Fu, C.-Y. Bai, Y. Liu, Efficient volumetric polycube-map construction, in: Computer graphics forum, Vol. 35, Wiley Online Library, 2016, pp. 97–106.

[68] Z. Q. Xie, R. Sevilla, O. Hassan, K. Morgan, The generation of arbitrary order curved meshes for 3d finite element analysis, Computational Mechanics 51 (2013) 361–374.

[69] P. Lamata, I. Roy, B. Blazevic, A. Crozier, S. Land, S. A. Niederer, D. R. Hose, N. P. Smith, Quality metrics for high order meshes: analysis of the mechanical simulation of the heart beat, IEEE transactions on medical imaging 32 (1) (2012) 130–138.

[70] A. Gargallo-Peiró, X. Roca, J. Peraire, J. Sarrate, Distortion and quality measures for validating and generating high-order tetrahedral meshes, Engineering with Computers 31 (3) (2015) 423–437.

[71] M. Stees, M. Dotzel, S. M. Shontz, Untangling high-order meshes based on signed angles, Proceedings of the 28th International Meshing Roundtable (2020).

[72] L. Engvall, J. A. Evans, Mesh quality metrics for isogeometric bernstein–bézier discretizations, Computer Methods in Applied Mechanics and Engineering 371 (2020) 113305.

[73] E. Cohen, T. Martin, R. Kirby, T. Lyche, R. Riesenfeld, Analysis-aware modeling: Understanding quality considerations in modeling for isogeometric analysis, Computer Methods in Applied Mechanics and Engineering 199 (5-8) (2010) 334–356.

[74] G. Xu, B. Mourrain, R. Duvigneau, A. Galligo, Optimal analysis-aware parameterization of computational domain in 3d isogeometric analysis, Computer-Aided Design 45 (4) (2013) 812–821.

[75] G. Xu, B. Mourrain, A. Galligo, T. Rabczuk, High-quality construction of analysis-suitable trivariate nurbs solids by reparameterization methods, Computational Mechanics 54 (2014) 1303–1313.

[76] C. R. Dudley, S. J. Owen, Degenerate hex elements, Procedia Engineering 82 (2014) 301–312.

[77] T. Schneider, Y. Hu, X. Gao, J. Dumas, D. Zorin, D. Panozzo, A large-scale comparison of tetrahedral and hexahedral elements for solving elliptic pdes with the finite element method, ACM Transactions on Graphics (TOG) 41 (3) (2022) 1–14.

[78] M. Stees, S. M. Shontz, An angular approach to untangling high-order curvilinear triangular meshes, in: International Meshing Roundtable, Springer, 2018, pp. 327–342.

[79] J.-F. Remacle, T. Toulorge, J. Lambrechts, Robust untangling of curvilinear meshes, in: Proceedings of the 21st International meshing roundtable, Springer, 2013, pp. 71–83.

[80] D. Moxey, M. Green, S. Sherwin, J. Peiró, An isoparametric approach to high-order curvilinear boundary-layer meshing, Computer Methods in Applied Mechanics and Engineering 283 (2015) 636–650.

[81] T. J. Hughes, J. A. Cottrell, Y. Bazilevs, Isogeometric analysis: Cad, finite elements, nurbs, exact geometry and mesh refinement, Computer methods in applied mechanics and engineering 194 (39-41) (2005) 4135–4195.

[82] S. Xia, X. Qian, Generating high-quality high-order parameterization for isogeometric analysis on triangulations, Computer Methods in Applied Mechanics and Engineering 338 (2018) 1–26.

[83] G. Xu, B. Mourrain, R. Duvigneau, A. Galligo, Parameterization of computational domain in isogeometric analysis: methods and comparison, Computer Methods in Applied Mechanics and Engineering 200 (23-24) (2011) 2021–2031.

[84] D. Fußeder, B. Simeon, A.-V. Vuong, Fundamental aspects of shape optimization in the context of isogeometric analysis, Computer Methods in Applied Mechanics and Engineering 286 (2015) 313–331.