A Computationally Efficient Isoparametric Tangled Finite Element Method for Handling Inverted Quadrilateral and Hexahedral Elements

Bhagyashree Prabhune, Krishnan Suresh*

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

Abstract

The finite element method (FEM) requires elements to be fully invertible or tangle-free. In particular, quadrilateral and hexahedral elements are required to be convex. However, generating high quality tangle-free meshes, especially 3D hexahedral meshes, remains an open challenge. Recently, the tangled finite element method (TFEM) was proposed to handle concave (tangled) quadrilateral elements. However, even in 2D, it was found to be computationally expensive and programmatically complex.

Here, we present a computationally efficient isoparametric-TFEM (i-TFEM) framework for inverted 2D quadrilateral and 3D hexahedral elements. i-TFEM employs the properties of isoparametric elements to make the formulation computationally much more efficient. In i-TFEM, the constraint on full invertibility (convexity) is replaced by partial invertibility by modifying the elemental stiffness matrices of the concave elements, and by incorporating certain piecewise-compatibility conditions. The proposed i-TFEM is simple, efficient, and provides accurate solutions with optimal convergence rate even in the presence of severely inverted elements. Moreover, i-TFEM requires minimal changes to the standard FEM framework and reduces to standard FEM for meshes without tangled elements. The accuracy and efficiency of i-TFEM are demonstrated by solving elastostatics problems on several 2D and 3D tangled meshes.

Keywords: Tangled Mesh, Hexahedral mesh, Inverted elements, Finite Element Method, TFEM, Concave elements, Self-intersection

1. Introduction

An important requirement for a finite element mesh to be valid is that it must not contain any inverted elements [1, 2, 3, 4]. In other words, the Jacobian associated with the parametric mapping must be positive *throughout* every element, and must be *fully* invertible. If a mesh contains one or more inverted elements, it is said to be *tangled*. In particular, quadrilateral and hexahedral meshes, that this paper focuses on, are tangled if they contain concave element(s). Figure 1a illustrates a tangled quadrilateral mesh, while Fig. 1b illustrates a tangled hexahedral mesh [5]. As is well known, the standard finite element method (FEM) leads to erroneous results over such tangled meshes [1, 2, 3, 4, 6].

*Corresponding author

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

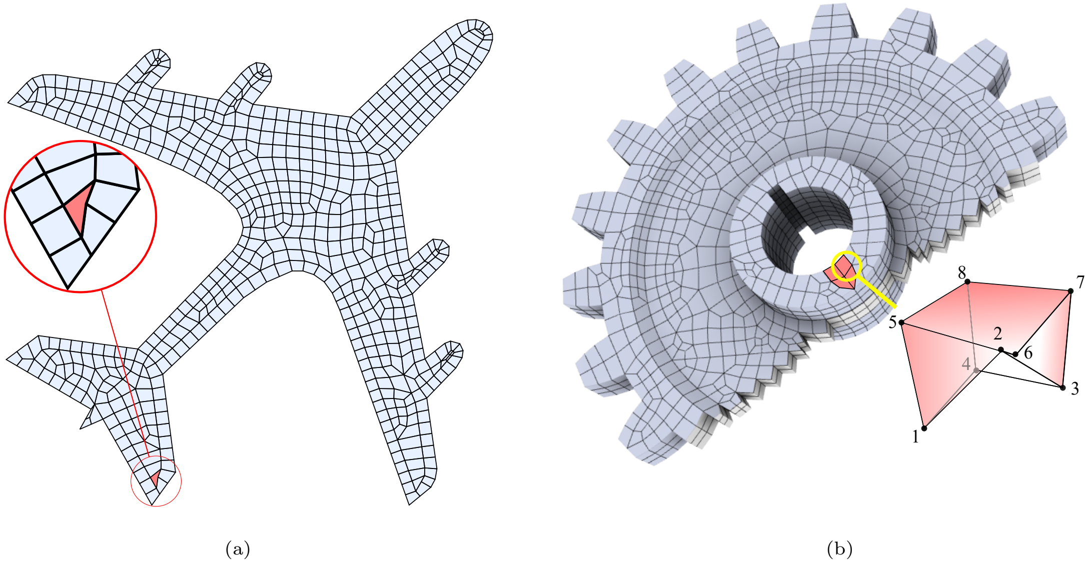


Figure 1: (a) Mesh for an aircraft model, with one concave element. (b) Gear mesh provided by [7]; elements in red color are concave.

Generating a high quality tangle-free mesh, especially in 3D, remains an open challenge [8, 9]. Even state-of-the-art mesh generation methods, such as the polycube mapping [10, 11, 12, 13], and frame-field based methods [14, 15, 16, 17], often lead to a tangled mesh. Tangling can also occur during shape optimization [18], large deformation simulations [19, 20], mesh morphing [21], and mesh optimization [7, 22, 23]. Many untangling techniques have been developed [7, 22, 23, 24, 25]. However, untangling is not always reliable; multiple instances have been reported where no tangle-free solution is possible without altering the boundary [7, 26, 22]. To quote [7] "*...the question of whether or not a given hex mesh connectivity has an inversion free embedding remains open, and multiple initial mesh configurations exist for which no inversion free solution is possible when boundary vertices are fixed.*"

However, more recently, it was shown that the *full invertibility* (convexity) requirement can be relaxed, i.e., *partial invertibility* is sufficient for quadrilateral elements. Specifically, the (generalized) tangled finite element method (g-TFEM) was introduced in [6] where: (1) correction terms were added to the stiffness matrices and (b) an equality constraint was enforced. It was demonstrated that g-TFEM leads to accurate solutions for Poisson problems for tangled quadrilateral elements. However, even in 2D, g-TFEM was found to be computationally expensive and cumbersome to implement.

In this work, a computationally efficient isoparametric-TFEM (i-TFEM) is presented for 2D quadrilateral and 3D hexahedral elements. The proposed i-TFEM takes into account the properties of isoparametric elements to arrive at a computationally faster formulation. The paper demonstrates that the requirement of full-invertibility can be replaced by partial invertibility i.e. contribution from only the fully invertible region of the concave element is sufficient. Specifically, to handle concave elements in i-TFEM, the elemental stiffness matrices associated with the concave elements are modified and piecewise-compatibility conditions are incorporated. Thus, i-TFEM reduces to standard FEM for non-tangled meshes. Using i-TFEM, standard finite element solvers can be easily augmented to handle concave elements. The accuracy and efficiency of i-TFEM are demonstrated through several 2D and 3D examples. The paper demonstrates that i-TFEM passes the patch test, and provides optimal convergence rates for elastostatics problems over severely tangled meshes in a computationally efficient manner.

The remainder of this paper is organized as follows. The formulation of g-TFEM is reviewed in Section 2. Section 3 describes the i-TFEM formulation and implementation. This is followed by numerical experiments in Section 4, and conclusions in Section 5.

2. Background

2.1. FEM over concave elements

First, we will briefly summarize the primary challenge associated with inverted, i.e., concave, elements in FEM. For illustrative purposes, we consider the weak form for elastostatics problems [1]:

Find $\boldsymbol{u} \in \mathbb{H}_0^1(\Omega)$ such that $\forall\ \boldsymbol{w} \in \mathbb{H}_0^1(\Omega)$,

$$\int_{\Omega} (\nabla \boldsymbol{w})^{\top} \boldsymbol{D} (\nabla \boldsymbol{u})\, d\Omega = \int_{\Omega} \boldsymbol{w}\boldsymbol{b} d\Omega \tag{1}$$

where, $\nabla \boldsymbol{u}$ denotes the symmetric part of displacement gradient while $\boldsymbol{D}$ is the material elasticity tensor and $\boldsymbol{b}$ is the body force. Recall that, in standard FEM, the underlying field at a point p belonging to the element E_j is approximated using shape functions $\boldsymbol{N}_j$ as follows [1]:

$$\boldsymbol{u}^h(p) = \boldsymbol{N}_j(p)\hat{\boldsymbol{u}}_j \tag{2}$$

Substituting Eq. 2 in the standard Galerkin form results in:

$$\left[\int_{\Omega} \left(\nabla \sum_j \boldsymbol{N}_j\right)^{\top} \boldsymbol{D} \left(\nabla \sum_k \boldsymbol{N}_k\right) d\Omega\right] \hat{\boldsymbol{u}} = \int_{\Omega} \left(\sum_j \boldsymbol{N}_j\right)^{\top} \boldsymbol{b} d\Omega \tag{3}$$

where $\hat{\boldsymbol{u}}$ represent the global degrees of freedom. This leads to the linear system: [1]:

$$\boldsymbol{K}^0 \hat{\boldsymbol{u}} = \boldsymbol{f} \tag{4}$$

where,

$$\boldsymbol{K}^0 = \prod_{Assemble} \int_{E_j} \nabla \boldsymbol{N}_j^{\top} \boldsymbol{D} \nabla \boldsymbol{N}_j d\Omega, \tag{5}$$

$$\boldsymbol{f} = \prod_{Assemble} \int_{E_j} \boldsymbol{N}_j^{\top} \boldsymbol{b} d\Omega. \tag{6}$$

To compute Eq. 5 and Eq 6, one must rely on parametric mapping. Consider the standard parametric mapping from (ξ, η) space in Fig. 2a to the concave (partly inverted) element in the physical space (x, y)

in Fig. 2b. For this particular concave element, one can show that the Jacobian $|\boldsymbol{J}|$ vanishes on the line $3\xi + 3\eta = 2$, dividing the parametric space into a positive $|\boldsymbol{J}|$ region and a negative $|\boldsymbol{J}|$ region as illustrated in Fig. 2a. The corresponding curve in the physical space is quadratic (see Fig. 2c). Observe that parametric points such as a ($\xi = 2/3,\ \eta = 2/3$) and b ($\xi = 0,\ \eta = 0$) map to the same point p ($x = 5/16,\ y = 5/16$) that lies in the folded region. Therefore the parametric mapping for a concave element is *not fully invertible*.

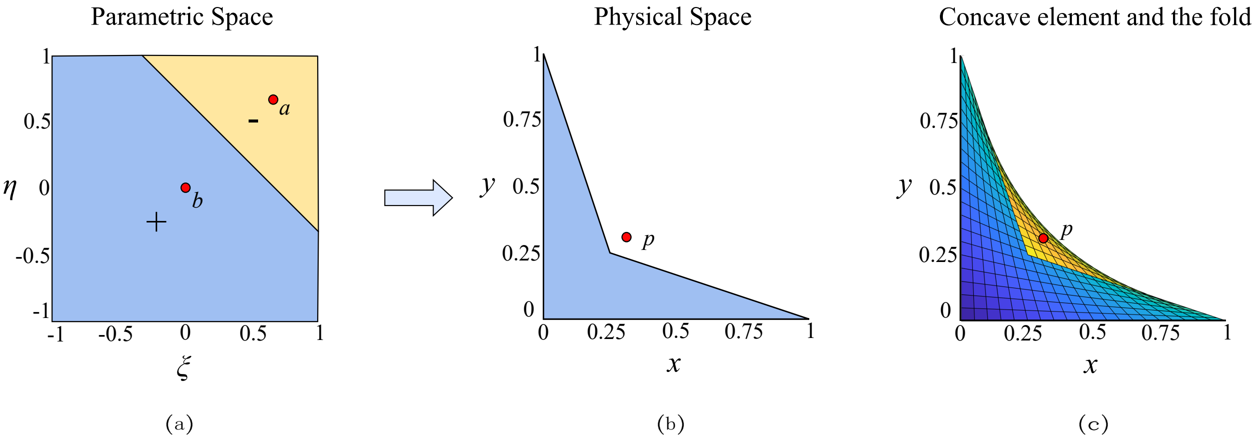


Figure 2: (a), (b) Parametric mapping of the concave quad onto the physical space (c) Concave element with the folded region.

Next to illustrate the impact of a concave element on FEM, consider a mesh with just two elements, where one of the elements is concave as in Fig. 3a. We pose a simple plane-stress elastostatics problem with Young's modulus $E = 1$, Poisson's ratio $\nu = 0.3$, where the exact fields are given by:

$$u_1(x, y) = 0.549x + 0.264y + 0.34, \quad u_2(x, y) = 0.486x + 0.351y - 0.62.$$

Dirichlet boundary conditions are imposed on the left edge and Neumann (traction) boundary conditions on the remaining edges. For numerical integration, 9 quadrature points are considered. Next, we vary the position of node 5 as in Fig. 3a, where $d \in (0, 0.5)$. Observe that for all values $0 < d < 0.5$, element E_1 is concave. For each value of d, we solve for the two fields, using standard Gaussian integration to evaluate Eq. 5 and Eq 6. To measure the accuracy of the FEM solution, L_2 error defined as follows is employed:

$$||\boldsymbol{u} - \boldsymbol{u}^h||_{L_2(\Omega)} = \sqrt{\int_{\Omega} (\boldsymbol{u} - \boldsymbol{u}^h)^2 \, d\Omega} \tag{7}$$

where $\boldsymbol{u}$ and $\boldsymbol{u}^h$ are the exact and computed solutions respectively. Fig. 3b illustrates L_2 error-norm, i.e., FEM leads to erroneous results for $0 < d < 0.5$.

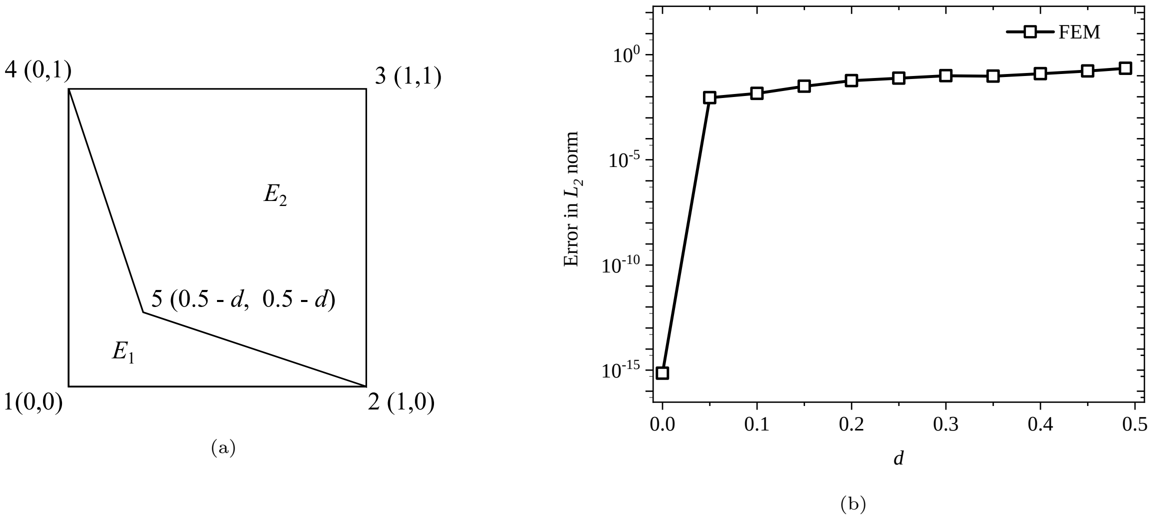


Figure 3: (a) Two element mesh. (b) L_2 error vs. d for FEM.

2.2. Brief Review of g-TFEM

To address the above problem, the generalized TFEM (g-TFEM) was introduced in [6]. The key concept in g-TEM is to handle *the positive and negative parametric regions of the concave element separately, thus relaxing the constraint of full invertibility to piecewise invertibility*. Specifically, we define the set of points in the physical space which maps from the positive (negative) $|\boldsymbol{J}|$ parametric region as positive (negative) component C_1^+ (C_1^-); see Fig. 4a and Fig. 4b.

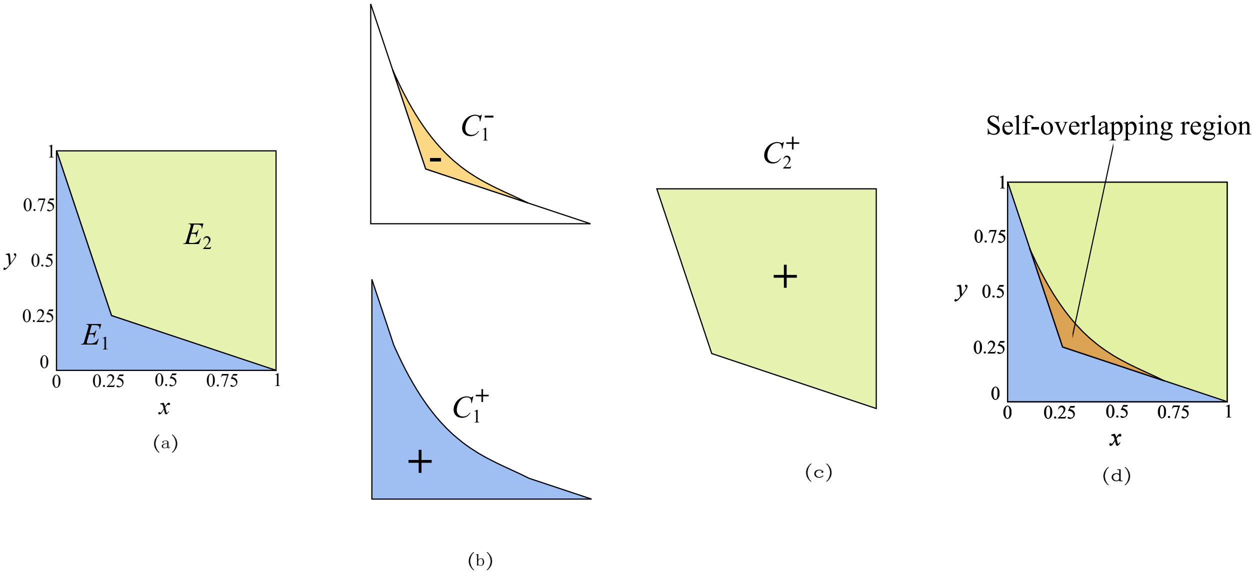


Figure 4: (a) 2-D domain discretized into two bilinear quads. (b) Positive and negative $|\boldsymbol{J}|$ regions of the concave element. (c) Convex element of the mesh. (d) Final physical space is self-overlapping.

The concave element (E_1) can then be expressed as the difference between the two components:

$$E_1 = C_1^+ - C_1^-. \tag{8}$$

On the other hand, the convex element E_2 has only the positive component:

$$E_2 = C_2^+;\ C_2^- = \emptyset. \tag{9}$$

Observe that the components C_1^+ and C_1^- overlap, and the overlapping region (tangled region) lies completely outside the physical boundaries of element E_1. Further, both the components intersect with E_2, i.e., a concave quadrilateral element not only self-intersects but also intersects the neighboring element(s).

Thus, any point in the tangled region can be interpreted as belonging to: (a) different parametric regions of the same element and (b) multiple elements. Thus, fundamentally, the field is ambiguous in the tangled/folded region and Eq. 2 is invalid. Redefining the field over the tangled region is the critical step in g-TFEM.

To define the field unambiguously in g-TFEM, observe that the parametric mapping is piecewise invertible. In other words, two shape functions $\boldsymbol{N}_1^+$ and $\boldsymbol{N}_1^-$ are defined corresponding to C_1^+ and C_1^- respectively. For example, for the point $p = (x = 5/16,\ y = 5/16)$ in Fig. 2b, $\boldsymbol{N}_1^+(p)$, are the shape functions of element E_1 evaluated at $(\xi = 0,\ \eta = 0)$ whereas $\boldsymbol{N}_1^-(p)$ are the shape functions of element E_1 evaluated at $(\xi = 2/3,\ \eta = 2/3)$. For the convex element E_2, only $\boldsymbol{N}_2^+$ exists, while $\boldsymbol{N}_2^-$ is defined to be zero.

With these definitions, the field at any point p is *defined* in g-TFEM as the oriented sum of the contributions from all *components* the point belongs to:

$$\boldsymbol{u}^h(p) \equiv \sum_{j|p\in C_j^+} \boldsymbol{N}_j^+(p)\hat{\boldsymbol{u}}_j - \sum_{j|p\in C_j^-} \boldsymbol{N}_j^-(p)\hat{\boldsymbol{u}}_j. \tag{10}$$

Eq. 10 is referred to as the *unambiguity* condition, i.e., the field is now unambiguously defined everywhere. For the above example, we have:

$$\boldsymbol{u}^h(p) = \boldsymbol{N}_1^+(p)\hat{\boldsymbol{u}}_1 - \boldsymbol{N}_1^-(p)\hat{\boldsymbol{u}}_1 + \boldsymbol{N}_2^+(p)\hat{\boldsymbol{u}}_2. \tag{11}$$

Substituting this in the Galerkin formulation (Eq. 1) results in:

$$\boldsymbol{K} = \int\limits_{\Omega} \left[\nabla \sum_j \left(\boldsymbol{N}_j^+ - \boldsymbol{N}_j^-\right)\right]^{\top} \boldsymbol{D} \left[\nabla \sum_k \left(\boldsymbol{N}_k^+ - \boldsymbol{N}_k^-\right)\right] d\Omega. \tag{12}$$

Regrouping leads to the stiffness matrix:

$$\boldsymbol{K} = \boldsymbol{K}^0 + \boldsymbol{K}^S + \boldsymbol{K}^N \tag{13}$$

where $\boldsymbol{K}^0$ is equivalent to the classical stiffness matrix, while $\boldsymbol{K}^S$ and $\boldsymbol{K}^N$ are the correction terms. Similarly, the forcing term can be computed.

In addition, in g-TFEM, a *piecewise compatibility* or *equality* constraint is needed at all re-entrant vertices to ensure compatibility.

$$\tilde{\boldsymbol{C}}\hat{\boldsymbol{u}} = \boldsymbol{0}. \tag{14}$$

For the above example, let point t be the physical location corresponding to the re-entrant vertex v_5. To be physically meaningful, the field at point t must match the nodal corresponding nodal value $\boldsymbol{u}^5$. Thus, the piecewise compatibility or equality condition is given as:

$$N_1^{1+}(t)\boldsymbol{u}^1 + N_1^{2+}(t)\boldsymbol{u}^2 + N_1^{4+}(t)\boldsymbol{u}^4 + (N_1^{3+}(t) - 1)\boldsymbol{u}^5 = 0. \tag{15}$$

Here, the node numbers are denoted using superscripts while element numbers are denoted using subscripts. In summary, in g-TFEM we must solve the following linear system of equations:

$$\begin{bmatrix} \boldsymbol{K} & \tilde{\boldsymbol{C}}^\top \\ \tilde{\boldsymbol{C}} & \boldsymbol{0} \end{bmatrix} \begin{Bmatrix} \hat{\boldsymbol{u}} \\ \boldsymbol{\mu} \end{Bmatrix} = \begin{Bmatrix} \boldsymbol{f}^\theta \\ \boldsymbol{0} \end{Bmatrix}. \tag{16}$$

After solving Eq. 16, we obtain the unknown degrees of freedom $\hat{\boldsymbol{u}}$, along with the Lagrange multipliers $\boldsymbol{\mu}$.

It was proven in [6] that g-TFEM satisfies the conditions of convergence, i.e., the field defined by Eq. 10 is continuous within and across the element boundaries and satisfies the rigid body and constant strain conditions.

To illustrate, consider again the example discussed in Section 2.1. Fig. 5 confirms that g-TFEM leads to machine precision accuracy for all the values of d.

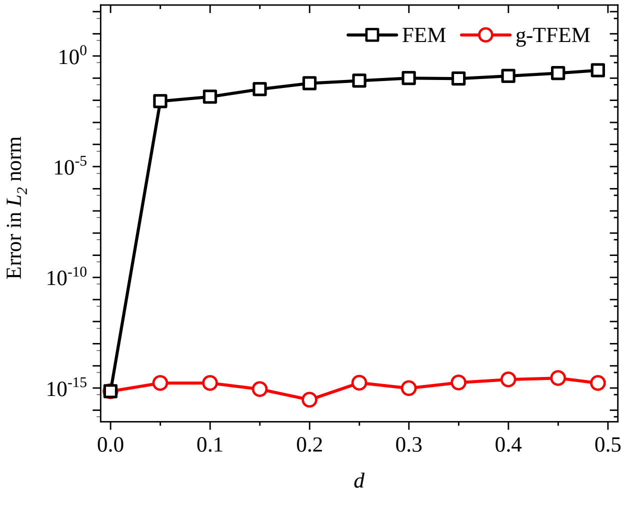


Figure 5: Comparison of g-TFEM and FEM for two-element mesh: L_2 error vs. d.

The major drawback of the g-TFEM formulation is that one must integrate over the fold to compute the correction terms. For example, in Eq. 13, $\boldsymbol{K}^S$ term is given by

$$\boldsymbol{K}^S = -\int\limits_{C_1^+ \cap C_1^-} \nabla \boldsymbol{N}_1^{+\top} \boldsymbol{D} \nabla \boldsymbol{N}_1^- d\Omega.$$

This entails integration over $C_1^+ \cap C_1^-$, i.e., over the tangled region. This makes g-TFEM cumbersome to implement and computationally expensive. Moreover, care must be taken to avoid integration points very close to the $|\boldsymbol{J}| = 0$ curve [6]. In the next section, an isoparametric-TFEM (i-TFEM) formulation is proposed that eliminates these drawbacks.

3. Isoparametric TFEM (i-TFEM)

3.1. Field Definition in i-TFEM

For the two-element mesh in Fig. 6, let the coordinates of the nodes of element E_1 be denoted by $\hat{\boldsymbol{x}}_1 = \left(\boldsymbol{x}^1, \boldsymbol{x}^2, \boldsymbol{x}^5, \boldsymbol{x}^4\right)$ and, for E_2, by $\hat{\boldsymbol{x}}_2 = \left(\boldsymbol{x}^2, \boldsymbol{x}^3, \boldsymbol{x}^4, \boldsymbol{x}^5\right)$. In isoparametric mapping, recall that the spatial interpolation is the same as the field interpolation. Thus, for any point p inside the fold, by definition, $\boldsymbol{x}^p = \boldsymbol{N}_1^+(p)\hat{\boldsymbol{x}}_1 = \boldsymbol{N}_1^-(p)\hat{\boldsymbol{x}}_1$.

Further, since the formulation must reproduce a constant strain field, when $\boldsymbol{u} = \boldsymbol{x}$, i.e.,

$$\boldsymbol{u}^h(p) = \boldsymbol{N}_1^+(p)\hat{\boldsymbol{u}}_1 = \boldsymbol{N}_1^-(p)\hat{\boldsymbol{u}}_1. \tag{17}$$

In other words, in an isoparametric formulation, the field value at any point within the fold must be the same independent of whether the positive or negative shape functions are used. This is an important result that we exploit to simplify the formulation.

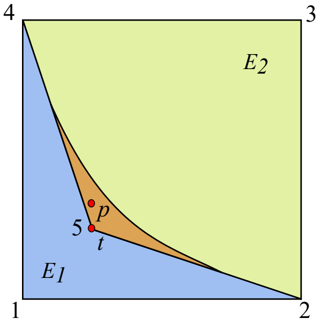


Figure 6: Equality condition must be imposed at all concave vertices.

As $p \to t$ (see Fig. 6), we have $\boldsymbol{N}_1^-(t)\hat{\boldsymbol{u}}_1 = \boldsymbol{u}^5$; therefore, Eq. 17 can be rewritten in the form:

$$\boldsymbol{u}^h(p) = N_1^{1+}(p)\boldsymbol{u}^1 + N_1^{2+}(p)\boldsymbol{u}^2 + N_1^{4+}(p)\boldsymbol{u}^4 + N_1^{3+}(p)\boldsymbol{u}^5 = \boldsymbol{u}^5. \tag{18}$$

Note that superscript and subscript indicate the node number and element number respectively. Observe that Eq. 18 is exactly the piecewise compatibility condition (Eq. 15). Thus, Eq. 17 is a generalization of the piecewise compatibility condition.

Now recall that the field at p within a fold is defined to be (see Eq. 10)

$$\boldsymbol{u}^h(p) = \boldsymbol{N}_1^+(p)\hat{\boldsymbol{u}}_1 - \boldsymbol{N}_1^-(p)\hat{\boldsymbol{u}}_1 + \boldsymbol{N}_2^+(p)\hat{\boldsymbol{u}}_2. \tag{19}$$

Simplifying, we have:

$$\boldsymbol{u}^h(p) = \boldsymbol{N}_2^+(p)\hat{\boldsymbol{u}}_2. \tag{20}$$

Thus, in an isoparametric element, the field at any point in the fold is simply the field defined by considering only the neighboring element E_2. Therefore, in i-TFEM, one can treat the fold as being part of the element E_2. Consequently, the mesh can be divided just into two parts: E_2 and $\widehat{E}_1$ as shown in Fig. 7. The contribution of E_2 to the stiffness matrix is given by the standard expression (the superscript '+' is dropped henceforth from the shape functions for brevity):

$$\boldsymbol{k}_2^0 = \int\limits_{E_2} \left(\nabla \boldsymbol{N}_2{}^\top \boldsymbol{D} \nabla \boldsymbol{N}_2\right) d\Omega. \tag{21}$$

This is the same as classical FEM, and standard Gauss quadrature schemes can be adopted. Next, consider the contribution of $\widehat{E}_1$ to the stiffness matrix. Note that the field in $\widehat{E}_1$ is given by:

$$\boldsymbol{u}^h(p) = \boldsymbol{N}_1^+(p)\hat{\boldsymbol{u}}_1. \tag{22}$$

Therefore the stiffness matrix is given by (the superscript '+' is dropped henceforth from the shape functions for brevity):

$$\widehat{\boldsymbol{k}}_1 = \int\limits_{\widehat{E}_1} \left(\nabla \boldsymbol{N}_1{}^\top \boldsymbol{D} \nabla \boldsymbol{N}_1\right) d\Omega. \tag{23}$$

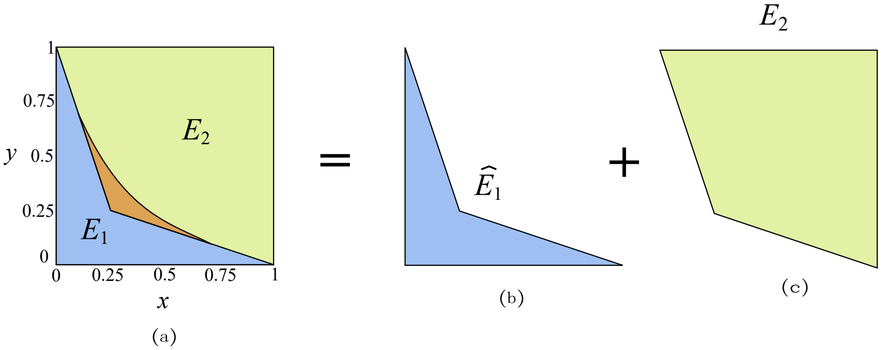
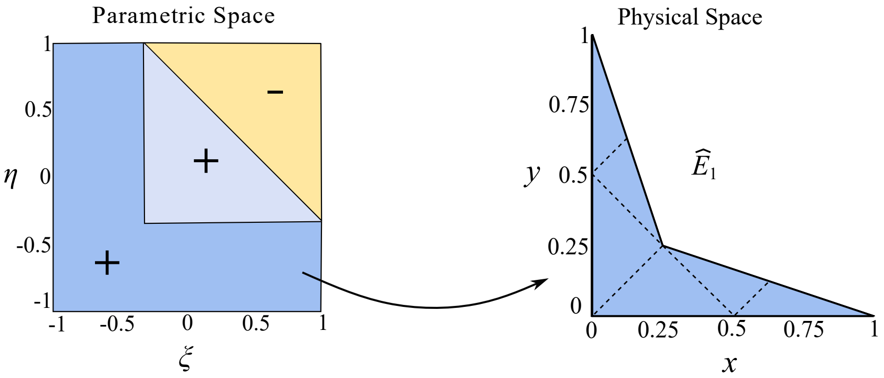


Figure 7: Parts contributing to the field definition.

Observe that $\widehat{E}_1$ is not the same as E_1. Though both have the same physical boundary, they represent different regions of parametric space. Specifically, E_1 represents the entire parametric space while $\widehat{E}_1$ represents only a subset of the positive $|\boldsymbol{J}|$ region as illustrated in Fig. 8a. Hence standard Gauss quadrature scheme for quadrilateral elements cannot be used to evaluate $\widehat{\boldsymbol{k}}_1$. Instead, we triangulate the concave region as in Fig. 8b for integration purposes; the integration is discussed in detail in the next subsection. Moreover, note that $\widehat{E}_1$ corresponds to the *invertible* subset of the parametric space of the concave element. Thus, in i-TFEM, we only integrate over the fully invertible region of the parametric space.



(a) (b)

Figure 8: (a) Parametric space. (b) Triangulation of $\widehat{E}_1$

3.2. Computing $\widehat{\boldsymbol{k}}$

A mesh may contain several concave elements; a typical contribution of the element stiffness matrix $\widehat{\boldsymbol{k}}$ due to a concave element $\widehat{E}$ is given by:

$$\widehat{\boldsymbol{k}} = \int\limits_{\widehat{E}} \left(\nabla \boldsymbol{N}^\top \boldsymbol{D} \nabla \boldsymbol{N}\right) d\Omega.$$

To evaluate the above expression, the above region $\widehat{E}$ is triangulated as illustrated in Fig. 9, i.e.,:

$$\widehat{\boldsymbol{k}} = \sum_{triangles} \int\limits_{triangle} \left(\nabla \boldsymbol{N}^\top \boldsymbol{D} \nabla \boldsymbol{N}\right) dxdy.$$

Then each triangle t is mapped to a standard triangle in (γ, ζ) space (see Fig. 9). Let $|\boldsymbol{J}_t|$ be the Jacobian associated with this triangle mapping. Thus, we have:

$$\widehat{\boldsymbol{k}} = \sum_t \int\limits_0^1 \int\limits_0^{1-\gamma} \left(\nabla \boldsymbol{N}^\top \boldsymbol{D} \nabla \boldsymbol{N}\right) |\boldsymbol{J}_t| d\gamma d\zeta.$$

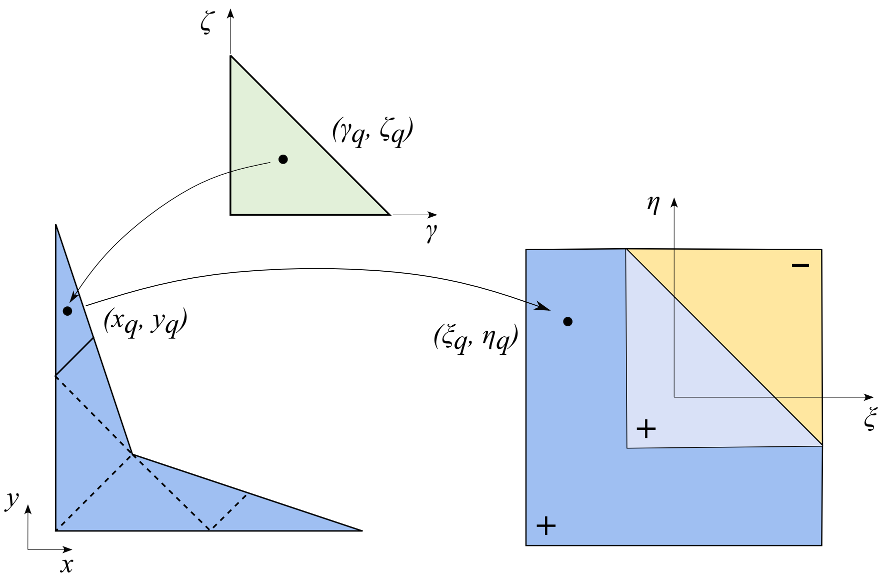


Figure 9: Numerical integration scheme based on triangulation of the concave element.

Consider a quadrature point (γ_q, ζ_q) of the standard triangle with weight w_q as shown in Fig. 9. The corresponding point (x_q, y_q) in the physical space is shown in Fig. 9. Let (ξ_q, η_q) be the coordinates in the quadrilateral parametric space. This point is numerically determined via Newton-Raphson algorithm. We compute the Jacobian matrix $(\boldsymbol{J}_q)$ associated with the quadrilateral parametric mapping at these quadrature points. Let

$$\boldsymbol{B}_q = (\boldsymbol{J}_q)^{-1} \nabla_{\xi\eta} \boldsymbol{N}(\xi_q, \eta_q). \tag{24}$$

Summing the contribution from all triangles results in:

$$\widehat{\boldsymbol{k}} = \sum_t \sum_q (\boldsymbol{B}_q)^\top \boldsymbol{D} (\boldsymbol{B}_q) |\boldsymbol{J}_t| w_q. \tag{25}$$

Similarly, the forcing term $\widehat{\boldsymbol{f}}$ can be computed by integrating over $\widehat{E}$.

3.3. Global Assembly

One can now assemble the individual global stiffness matrices as follows:

$$\boldsymbol{K}^0_{convex} = \prod_{Assemble-convex} \boldsymbol{k}^0_j, \tag{26}$$

$$\widehat{\boldsymbol{K}}_{concave} = \prod_{Assemble-concave} \widehat{\boldsymbol{k}}_j. \tag{27}$$

The final global stiffness matrix for i-TFEM is then given by:

$$\boldsymbol{K}_{iso} = \boldsymbol{K}^0_{convex} + \widehat{\boldsymbol{K}}_{concave}. \tag{28}$$

Similarly the forcing term is given by:

$$\boldsymbol{f}_{iso} = \boldsymbol{f}^0_{convex} + \widehat{\boldsymbol{f}}_{concave} \tag{29}$$

where,

$$\boldsymbol{f}^0_{convex} = \prod_{Assemble-convex} \boldsymbol{f}^0_j \tag{30}$$

and

$$\widehat{\boldsymbol{f}}_{concave} = \prod_{Assemble-concave} \widehat{\boldsymbol{f}}_j. \tag{31}$$

Finally, we solve the following linear system of equations:

$$\begin{bmatrix} \boldsymbol{K}_{iso} & \tilde{\boldsymbol{C}}^\top \\ \tilde{\boldsymbol{C}} & \boldsymbol{0} \end{bmatrix} \begin{Bmatrix} \hat{\boldsymbol{u}} \\ \boldsymbol{\mu} \end{Bmatrix} = \begin{Bmatrix} \boldsymbol{f}_{iso} \\ \boldsymbol{0} \end{Bmatrix}. \tag{32}$$

where the compatibility matrix $\tilde{\boldsymbol{C}}$ remains unchanged.

Observe that if the mesh does not contain any concave elements, only the terms corresponding to the convex elements i.e. $\boldsymbol{K}^0_{convex}$ and $\boldsymbol{f}^0_{convex}$, remain. Thus, i-TFEM reduces to the standard FEM for meshes without concave elements.

3.4. g-TFEM versus i-TFEM

The main difference between g-TFEM and i-TFEM is the following. In g-TFEM, one must integrate over the tangled region and over the concave region illustrated in Fig. 10a (see Section 2.2). However, in i-TFEM, we only need to integrate over the concave region (see Fig. 10b).

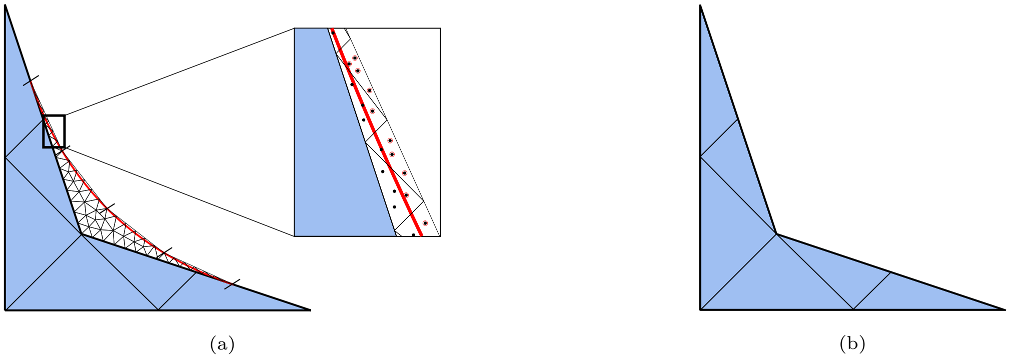


Figure 10: (a) g-TFEM involves integrating over the tangled region and concave region. (b) i-TFEM involves integrating only over the concave region.

In g-TFEM, to integrate over the fold, it is triangulated as shown in Fig. 10a. Further, in order to triangulate the fold, $|\boldsymbol{J}| = 0$ curve is approximated with a sufficiently large number of segments. To illustrate, let the $|\boldsymbol{J}| = 0$ curve (highlighted) be approximated by four line segments, and the polygonal fold triangulated, as in Fig. 10a. Observe that the integration points for some of the triangles lie outside the folded region. This will lead to singularities and therefore erroneous results. Hence, sufficiently large number of segments are employed to approximate $|\boldsymbol{J}| = 0$ curve leading to a large number of triangles. Thus, integrating over the fold is computationally expensive and programmatically complex.

On the other hand, in i-TFEM, a small number of triangles can be employed to integrate over $\widehat{E}_j$ as shown in Fig. 10b, making i-TFEM computationally more efficient. This is demonstrated in Section 4.

3.5. Extension to 3D

We now consider the implementation of the i-TFEM framework in 3D. Consider a canonical 8-node hexahedral element in Fig. 11a. If node 6 is moved diagonally towards node 4, as shown in Fig. 11b, it can become concave, i.e., tangled.

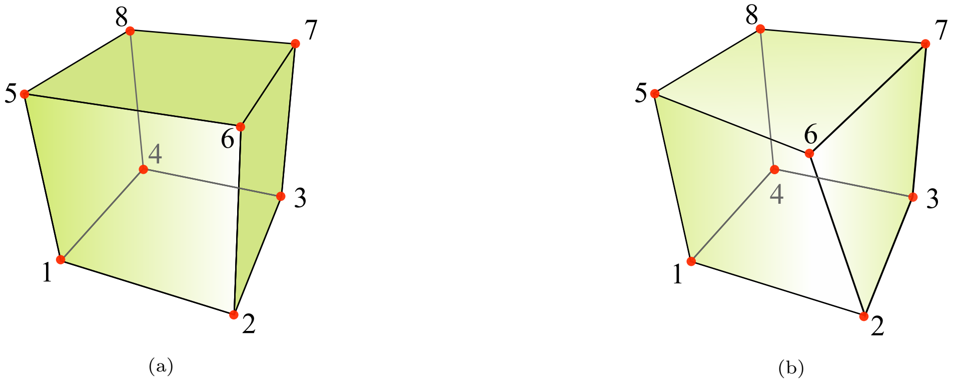


Figure 11: (a) Regular (untangled) hex element. (b) Tangled hex element.

Different views of the corresponding tangled region are shown in Fig. 12. Observe that the tangled region is much more complex in 3D. Since i-TFEM avoids integration over such complex tangled regions, it has a pronounced advantage over g-TFEM in 3D.

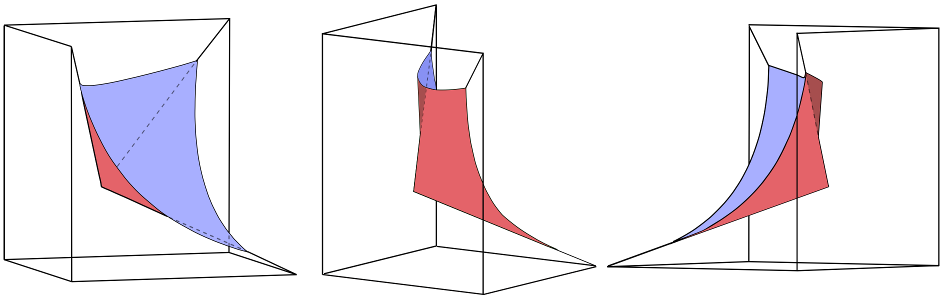


Figure 12: The boundary of the tangled (negative $|\boldsymbol{J}|$) region.

The standard elemental stiffness matrices for convex 3D elements $\boldsymbol{k}_j^0$ are computed as in 2D. For a concave element, the stiffness matrix $\widehat{\boldsymbol{k}}_j$ is computed by tetrahedralizing the region $\widehat{E}_j$ as shown in Fig. 13. The contribution of each tetrahedron to $\widehat{\boldsymbol{k}}_j$ is computed by generalizing the procedure discussed in Section 3.2.

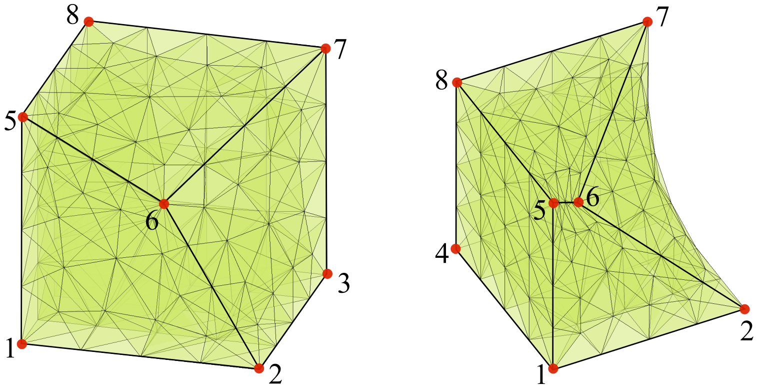


Figure 13: Different views of tetrahedralized concave element.

However, unlike in 2D, $\widehat{E}_j$ is not a polyhedron. In general, the bounding surfaces are non-planar since there are four points that define each surface. Thus, they need to be *approximated* using triangulation. Finer surface triangulation results in better approximation as illustrated later in Section 4.

The final step in i-TFEM involves applying piecewise compatibility conditions. In 2D, a concave quadrilateral element has only one re-entrant vertex, and one constraint per DOF is applied at the re-entrant vertex. On the other hand, a concave hexahedral element can have more than one re-entrant vertex due to geometry as well as triangulations. For instance, the element in Fig. 14 has two re-entrant vertices (5 and 6). Further, the triangulation will induce several re-entrant vertices along the edge 5-6.

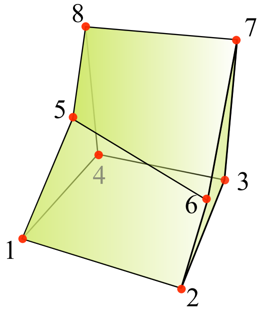


Figure 14: Concave hexahedral element with multiple re-entrant vertices.

Hence, one cannot determine *a priori* the number of constraints in 3D. However, a simple technique based on the rank of the element stiffness matrix can be employed for this purpose as described below.

1. We start with one of the geometric re-entrant vertices. Eq. 17 is computed at this vertex for all DOFs, and these are appended to the elemental stiffness matrix $\widehat{\boldsymbol{k}}$.

2. The rows and columns corresponding to fixed DOFs are removed from the appended $\widehat{\boldsymbol{k}}$.

3. The rank of the resultant matrix $\widehat{\boldsymbol{k}}$ is determined. If the matrix $\widehat{\boldsymbol{k}}$ is full-ranked, the constraints (Eq. 17) are computed at another re-entrant vertex (or equivalently, at any other point within the tangled region) and appended to the matrix.

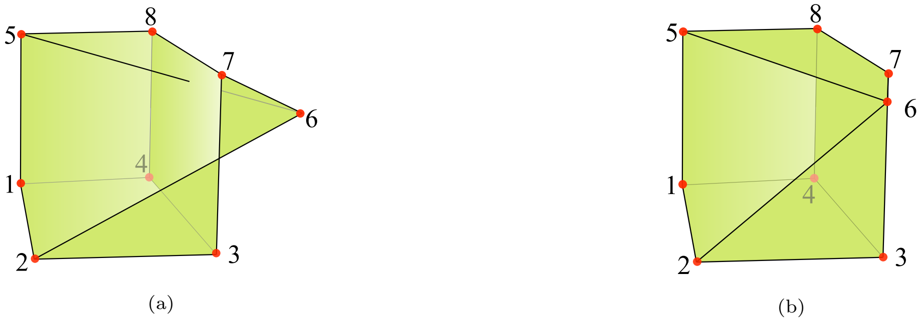
4. The above step is repeated as long as the matrix $\widehat{\boldsymbol{k}}$ is full-ranked. Once the matrix becomes rank-deficient, the constraints associated with the last point are removed. Thus, the number of piecewise compatibility conditions is the maximum number of conditions that result in a full-ranked matrix.

5. These constraints are then appended to the i-TFEM global stiffness matrix $\boldsymbol{K}_{iso}$ to form Eq. 32.

Thus, the fundamental changes in 3D i-TFEM are: (1) in general, the surfaces of the concave elements are curved and must be approximated by triangles, and the enclosed region must be approximated by tetrahedrons, and (2) the number of piecewise compatibility conditions required is determined by examining the rank of the elemental stiffness matrix.

3.6. Self-Penetrating Elements

The current framework cannot directly handle self-penetrating elements such as the one shown in Fig. 15a. However, one can typically shift the offending node such that the element is not self-penetrating as in Fig. 15b (but the element is still concave). Then, the proposed framework can be applied.



(a)

(b)

Figure 15: (a) Self-penetrating hexahedral element. (b) Acceptable concave element by moving node 6.

4. Numerical Experiments

In this section, we demonstrate i-TFEM using numerical experiments. Elastostatics problems in 2D and 3D are solved over various tangled meshes. Numerical experiments are conducted under the following conditions:

- The implementation is in MATLAB R2021b, on a standard Windows 10 desktop with Intel(R) Core(TM) i9-9820X CPU running at 3.3 GHz with 16 GB memory.

- The number of quadrature points for convex quadrilateral elements is 4, while 8 quadrature points are used for convex hexahedral elements unless otherwise stated.

- In 2D, the triangulation of a concave element is performed by employing MATLAB's inbuilt mesher - `generateMesh`. The number of quadrature points for triangles is 3. In 3D, tetrahedralization of concave elements is performed using Tetgen [27]. The bounding surfaces are triangulated using `generateMesh`, and serve as input to Tetgen. The surface mesh-size is set to a relative size of $h_t = 0.05$, where h_t is defined as the maximum allowable edge length of a surface triangle. The number of quadrature points for tetrahedrons is chosen to be 4.

The questions being investigated through the experiments are:

- **Accuracy**: How does the accuracy of i-TFEM compare against that of the FEM? Note that g-TFEM accuracy will match that of i-TFEM (but will be more expensive). To measure accuracy, we consider both the L_2 error norm as defined in Eq. 7, and the energy error norm defined as:

$$||e_h||_{E(\Omega)} = \sqrt{\int_{\Omega} \left(\boldsymbol{\epsilon} - \boldsymbol{\epsilon}^h\right)^\top \boldsymbol{D} \left(\boldsymbol{\epsilon} - \boldsymbol{\epsilon}^h\right) d\Omega} \tag{33}$$

  where $\boldsymbol{\epsilon}$ and $\boldsymbol{\epsilon}^h$ are the exact and computed strain fields respectively.

- **Condition Number**: How does the condition number of i-TFEM compare against FEM and g-TFEM? The condition number is a measure of a matrix's invertibility [28]; it is desirable to have a condition number close to unity. To compute the 1-norm condition number, we employ MATLAB's built-in function `condest`.

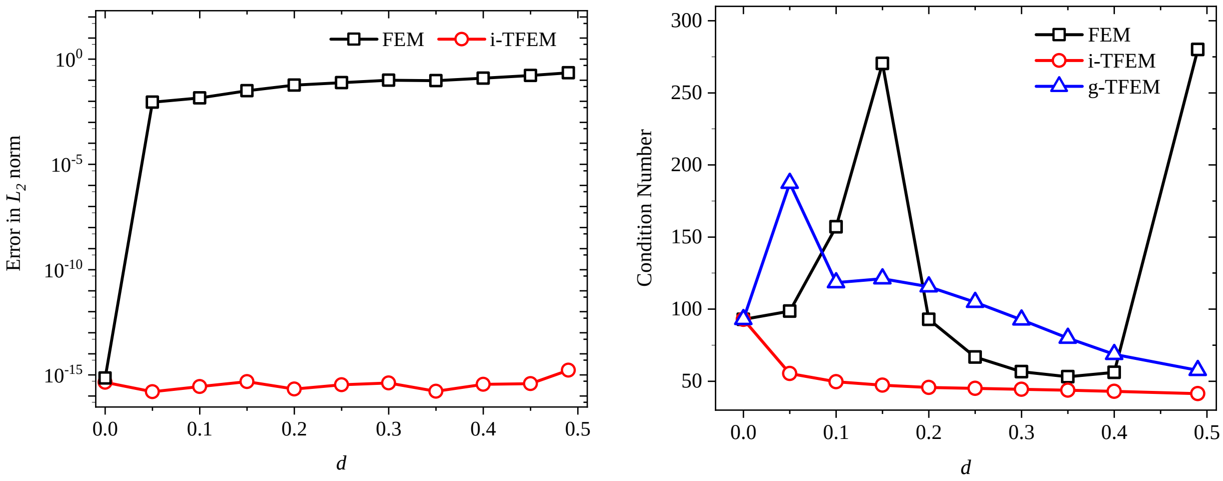
- **Computational Cost**: Is i-TFEM computationally more efficient than g-TFEM? MATLAB's built-in `cputime` function is employed to measure the computational cost.

- **Convergence**: What is the convergence rate of i-TFEM and FEM as the element size decreases?

4.1. 2D Patch Tests

4.1.1. Two element mesh

In the first experiment, we solve the problem discussed in Section 2.2 using i-TFEM. Fig. 16a compares the L_2 errors in FEM and i-TFEM, for varying degrees of tangling. i-TFEM achieves machine precision accuracy while classic FEM fails when the mesh gets tangled (g-TFEM will match the accuracy of i-TFEM but is harder to implement). On the other hand, the condition number of i-TFEM is lower than that of g-TFEM as illustrated in Fig. 16b. This is because, in i-TFEM, we avoid integration over the regions close of $|\boldsymbol{J}| = 0$ curve. The CPU time for FEM is 0.0111 seconds whereas, it is 0.0225s and 0.6120s for i-TFEM and g-TFEM respectively.



(a)

(b)

Figure 16: Comparison of i-TFEM and FEM for two-element mesh: (a) L_2 error vs. d, (b) condition number vs. d.

4.1.2. Four element mesh

Next, we consider a square domain $\Omega = (0, 1) \times (0, 1)$ which is discretized into four quadrilateral elements, one of which is concave as in Fig. 17. The folded region is shared by three neighboring convex elements. To introduce asymmetry, we move vertex 9 along an arc of a circle as illustrated, where α varies from 15° to 75°, and radius $r = 0.125\sqrt{2}$.

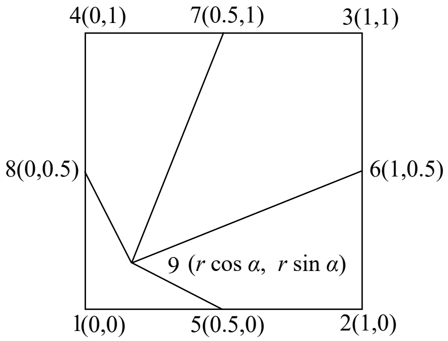
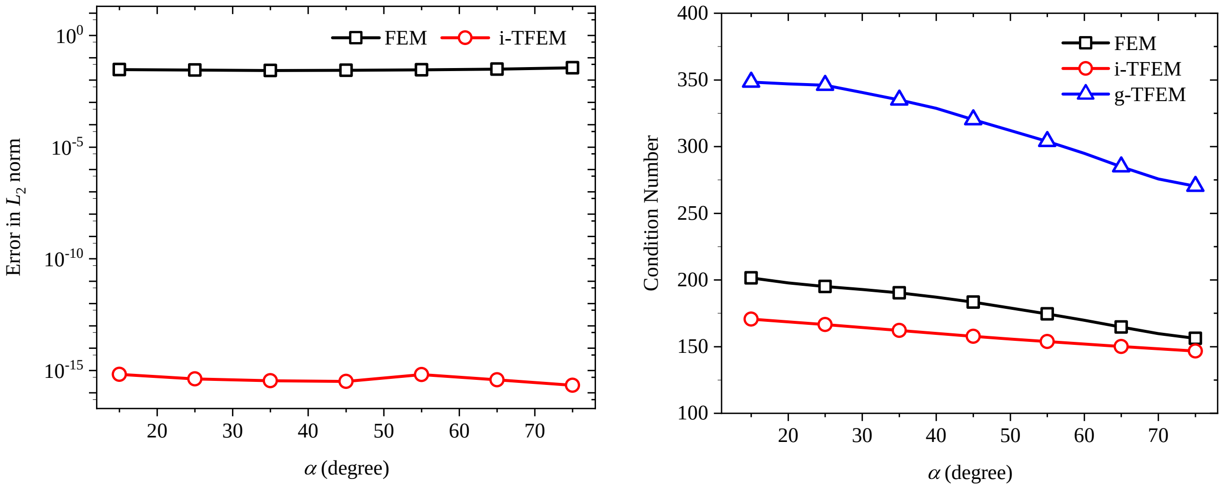


Figure 17: Tangled mesh with four elements; one of the elements is concave.

A plane stress elastostatics problem is solved with the displacement fields, elastic constants, and boundary conditions as in the two-element example considered earlier. Fig. 18a illustrates the L_2 errors in FEM and i-TFEM, while Fig. 18b compares the condition numbers. As with the previous example, i-TFEM achieves machine precision accuracy while classic FEM fails. In terms of the condition number, i-TFEM again fares better than FEM and g-TFEM as illustrated in Fig. 18b. Finally, FEM requires 0.0180s while i-TFEM and g-TFEM require 0.0237 and 2.2424s receptively. Thus, i-TFEM outperforms g-TFEM in terms of speed and condition number.



(a)

(b)

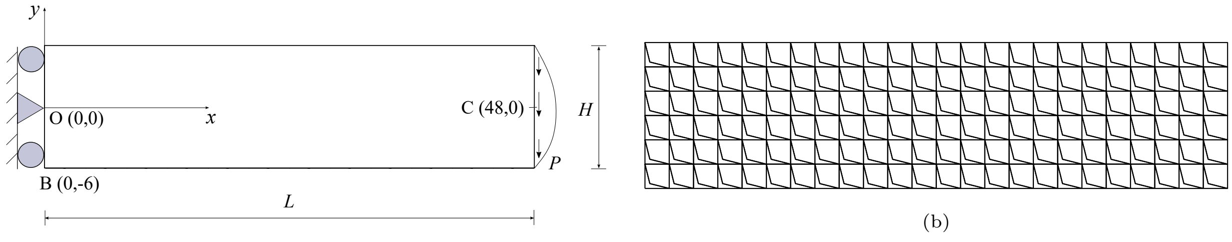
Figure 18: Comparison of i-TFEM and FEM for four-element mesh: (a) Error vs. α (b) Condition number vs. α

4.2. Cantilever Beam with Parabolic loading

Consider a cantilever beam of length $L = 48$m, height $H = 12$m, and a unit thickness subjected to a parabolic traction $P = 1000$N on the right edge; see Fig 19a. Assuming plane stress, the analytical solution is given by [29]:

$$u_1 = \frac{Py}{6EI}\left[(6L-3x)\,x + (2+\nu)\left(y^2 - \frac{H^2}{4}\right)\right]$$
$$u_2 = -\frac{P}{6EI}\left[(L-x)\,3\nu y^2 + (4+5\nu)\,\frac{H^2 x}{4} + (3L-x)\,x^2\right]$$

where $I = H^3/12$, $E = 3 \times 10^7$ kPa and $\nu = 0.3$.



(a)

(b)

Figure 19: (a) Cantilever with parabolic loading and (b) mesh with concave elements.

To solve this problem using finite elements, the domain is discretized to form a tangled mesh as shown in Fig. 19b and the boundary conditions are applied as shown in Fig. 19a. Here, the basic repeating unit is a 2-element mesh (Fig. 3a) with $d = 0.4$.

To evaluate the performance of FEM and i-TFEM, the error in the vertical displacement measured along $y = 0$ is plotted in Fig. 20; while Fig. 21 compares the stresses obtained using FEM and i-TFEM. Here, the stresses are measured at the element center along $x = 23.05$. We observe that i-TFEM is more accurate than FEM, even when the exact solutions lie outside the span of the finite element space.

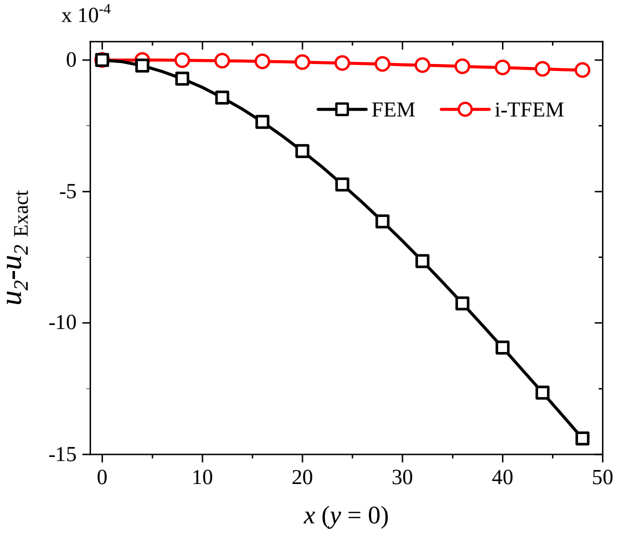
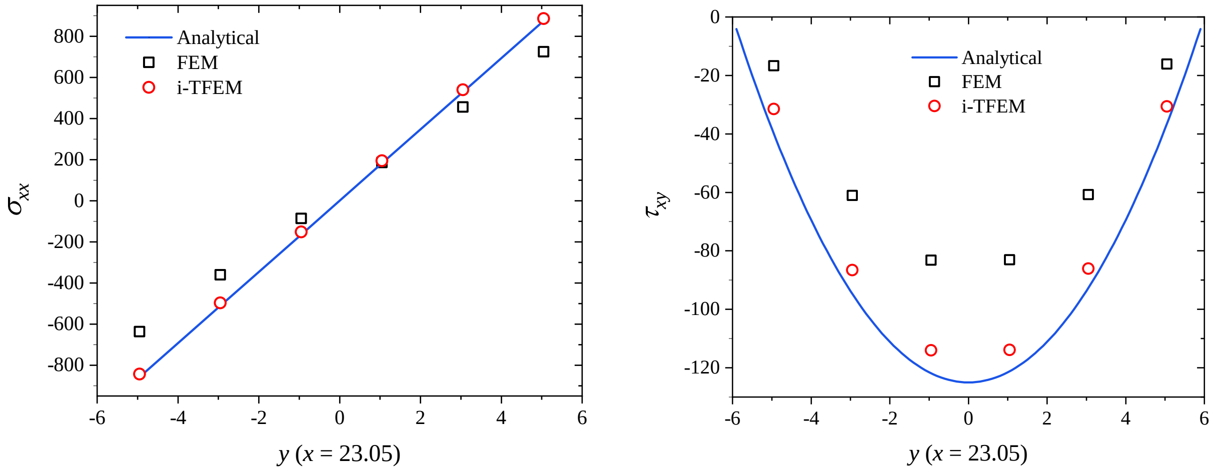


Figure 20: Error in u_2.



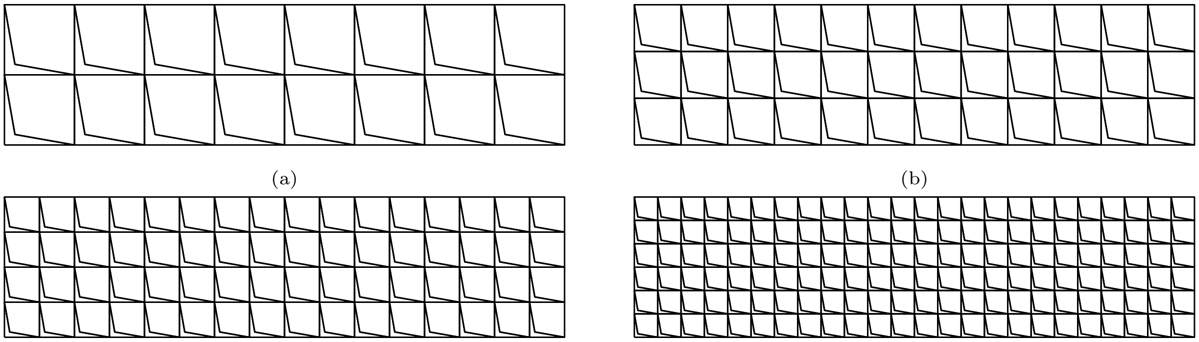
(a)

(b)

Figure 21: (a) σ_{xx} and (b) τ_{xy}

4.2.1. Convergence

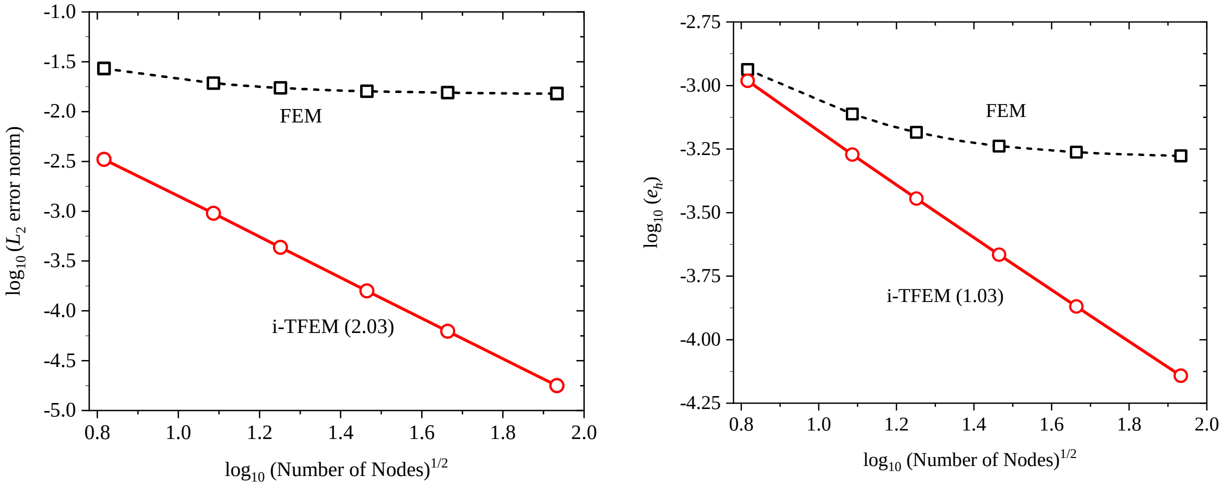
To study the convergence for the cantilever beam problem, various meshes are constructed as shown in Fig. 22. The basic repeating unit is the two-element mesh (see Fig. 3a) with $d = 0.4$.



(a) (b) (c) (d)

Figure 22: Sample meshes for convergence study with number of nodes equal to (a) 43 (b) 88 (c) 149 (d) 319.

The L_2 and energy norm errors for FEM and i-TFEM, as a function of the number of nodes, are illustrated in Fig. 23. For i-TFEM, the L_2 and energy norm convergence rates are 2.03 and 1.03 respectively, i.e., they are optimal.



(a) (b)

Figure 23: (a) L_2 and (b) energy norms errors as a function of the number of nodes for cantilever problem.

4.2.2. Effect of element distortion

In the previous experiments, the extent of tangling was fixed with $d = 0.4$. In this experiment, we study the effect of tangling on the computed solutions by varying d from 0 to 0.49. In particular, we compute σ_{xx} at the point B (located at the bottom left corner of the cantilever) and the vertical displacement u_2 at point C (located on the right edge); see Fig. 19a.

When $d = 0$, all quadrilaterals reduce to triangles and error is due to element distortion; i-TFEM reduces to FEM in this case. As d is increased, tangling increases and FEM error increases. On the other hand, i-TFEM error (mainly due to distortion) decreases as d increases, as illustrated in Fig. 24a and Fig. 24b.

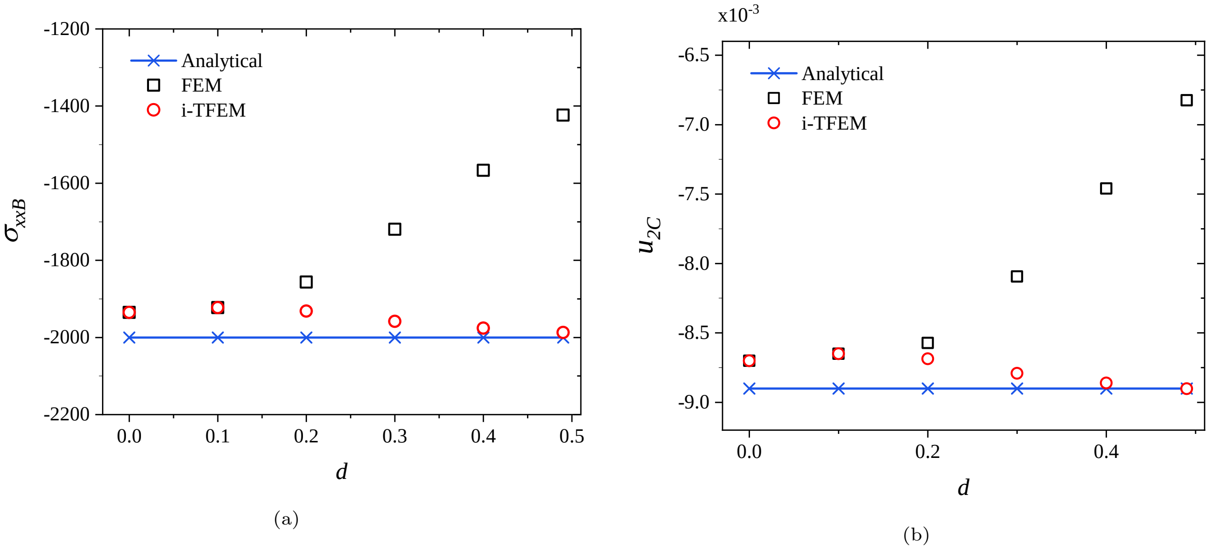


Figure 24: (a) σ_{xx} at point B and (b) u_2 at point C.

4.3. Pressurized cylinder

Consider a long hollow cylinder with internal radius $a = 1$ and external radius $b = 4$; see Fig. 25a. A uniform pressure $p = 1$ is applied to the inner surface ($r = a$) and the cylinder deforms in plane strain. Let Poisson's ratio, $\nu = 0.3$ and Young's modulus, $E = 2.6$. Due to the axisymmetric nature of the problem, only a quarter of the cylinder is modeled. The analytical solution is as follows [29]:

$$\boldsymbol{u} = \frac{p(1+\nu)a^2b^2}{E(b^2-a^2)}\left(\frac{1}{r} + \frac{r(1-2\nu)}{b^2}\right)\boldsymbol{e}_r$$

A typical mesh, shown in Fig. 25b, is constructed by using the two-element mesh as the basic repeating unit.

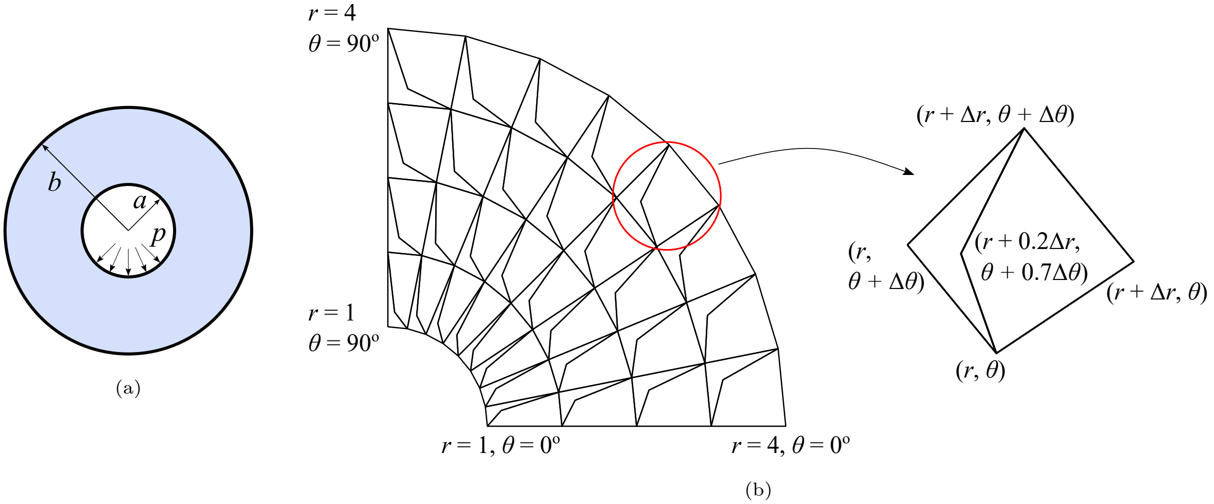
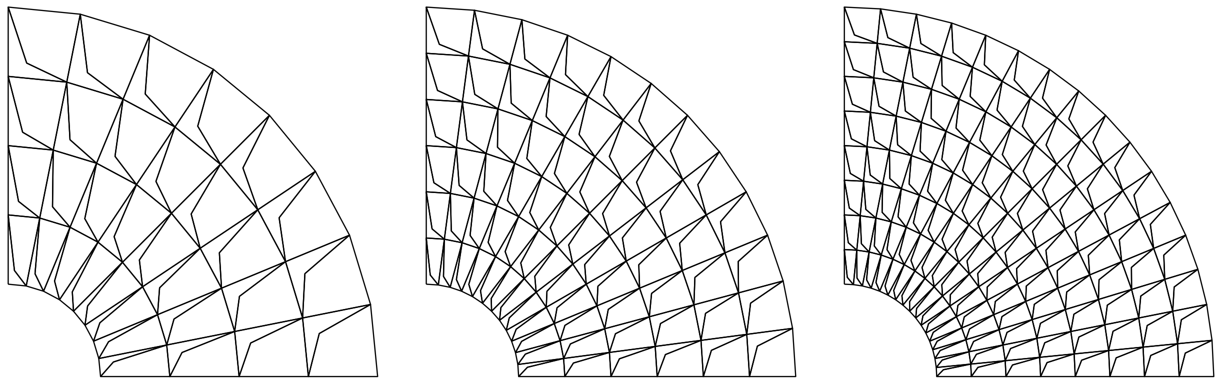


Figure 25: (a) Cross-section of the pressurized cylinder (b) A typical mesh with concave elements.

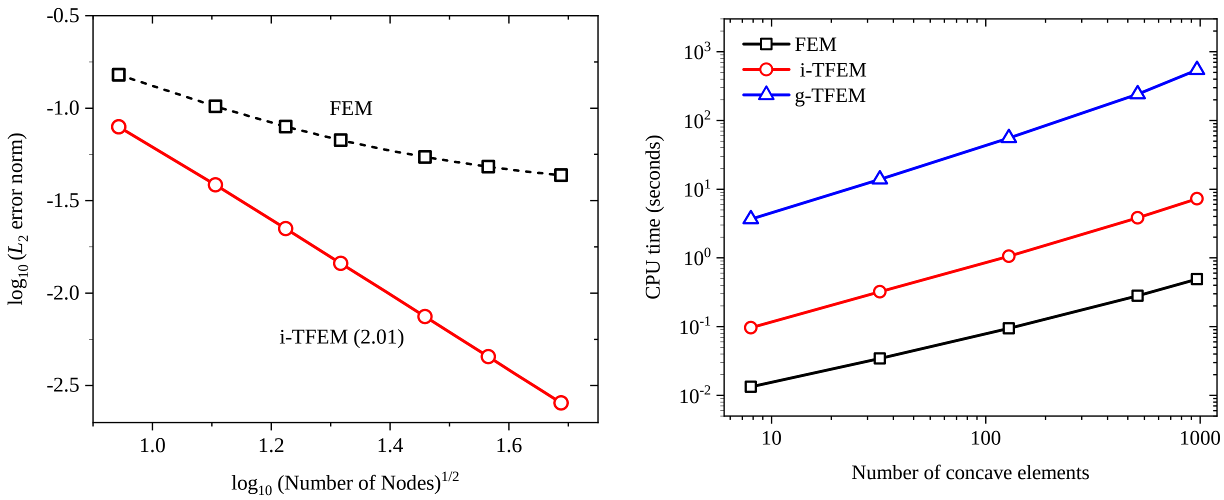
To study the convergence behavior, various meshes are constructed as shown in Fig. 26.



(a) (b) (c)

Figure 26: Sample meshes for convergence study. Number of nodes and the number of concave elements are (a) 77, 32 (b) 163, 72 (c) 281, 128 respectively.

Fig. 27a confirms that i-TFEM leads to an optimal convergence rate as opposed to FEM. Next, we compare the CPU time for FEM, i-TFEM, and g-TFEM for various mesh sizes. Fig. 27b shows that i-TFEM is much faster compared to the g-TFEM. FEM is the fastest, but inaccurate. Note that for the meshes considered for this study (Fig. 26), there are equal number of concave and convex elements. However, in practical scenarios, this is rarely the case since the number of concave elements will be much smaller compared to the convex elements [7, 22]. Therefore, the additional cost incurred by i-TFEM will be minimal.



(a) (b)

Figure 27: (a) L_2 error norm as a function of number of nodes and (b) CPU time as a function of number of concave elements in mesh for pressurized cylinder problem.

4.4. 3D Patch test: Eight-element Hex Mesh

Consider a cubic domain $\Omega = (0,2) \times (0,2) \times (0,2)$. The domain is discretized into eight hexahedral elements as shown in Fig. 28a. For the regular (untangled) mesh, the central node is located at $(1,1,1)$. In order to demonstrate i-TFEM, the central node is moved so that one element becomes concave as shown in Fig. 28b. Note that the concave element has only one re-entrant vertex. To vary the extent of tangling, the position of the central node is given as: $(1, 1, 1) - d \times (1, 0.95, 0.98)$ where the parameter $d \in (0, 0.9)$. For $d = 0$, the mesh is the regular grid as shown in Fig. 28a.

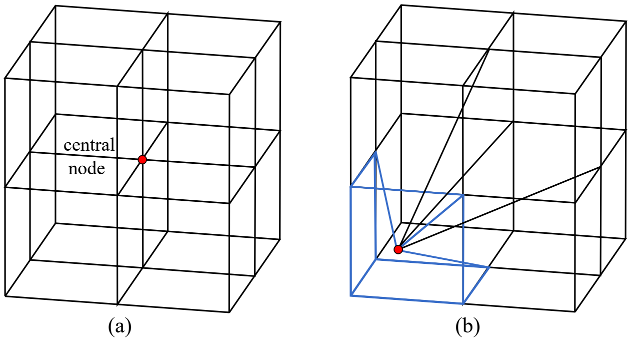


Figure 28: Eight-element (a) regular grid (b) tangled mesh with a concave hex element.

The material parameters are $E = 1$, and $\nu = 0.3$. Let the exact displacement field $\boldsymbol{u}$ be

$$u_1 = 0.579x + 0.246y + 0.482z - 0.374, \quad u_2 = 0.486x + 0.351y + 0.947z - 0.62,$$

$$u_3 = 0.512x + 0.746y + 0.548z - 0.48$$

The corresponding Dirichlet boundary conditions are applied on the left surface, while Neumann conditions are applied on the remaining surfaces.

As shown in Fig. 29a, i-TFEM is significantly more accurate than FEM. Recall that the accuracy of i-TFEM in 3D depends on how well the bounding surfaces of the concave element are approximated via surface triangulations. To study the effect of surface-mesh size on the i-TFEM accuracy, we consider three mesh sizes: $h_t = 0.035, 0.007$, and 0.005. Here, h_t indicates the maximum edge length of triangles. Finer surface triangulation results in better accuracy of i-TFEM solution as illustrated in Fig. 29a. For $h_t = 0.005$, i-TFEM is 10^{10} times more accurate than FEM. Fig. 29b compares the condition number for FEM and i-TFEM (for all values of h_t).

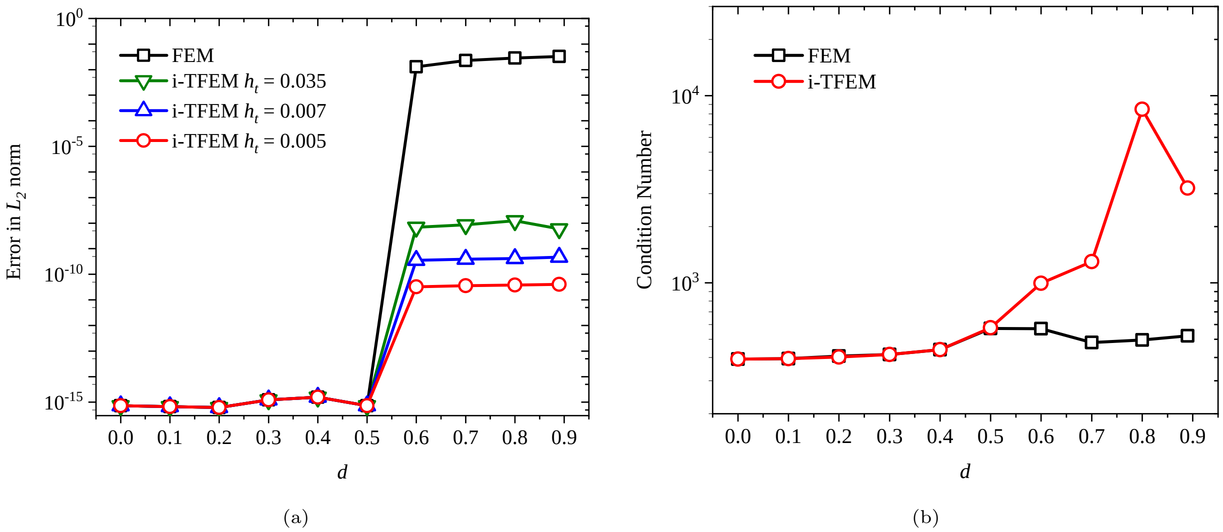


Figure 29: Comparison of i-TFEM and FEM for eight-element hex mesh: (a) L_2 error vs. d, (b) condition number vs. d.

4.5. Application: Connecting rod

We now consider a connecting rod mesh (mesh data obtained from [7]) illustrated in Fig. 30a; the units are assumed to be in cm. The mesh contains 14783 nodes and 11316 hexahedral elements, out of which 16 elements are concave (highlighted in Fig. 30a). Some of the elements were self-penetrating (see Fig. 15a); therefore, the nodes were perturbed such that the elements do not self-penetrate (see Fig. 15b). An axial load of $P = 300$N was applied on one end, while fixed at the other, as shown in Fig. 30a. The material properties are as follows: the Young's modulus $E = 1.1 \times 10^5$ MPa and the Poisson's ratio $\nu = 0.35$.

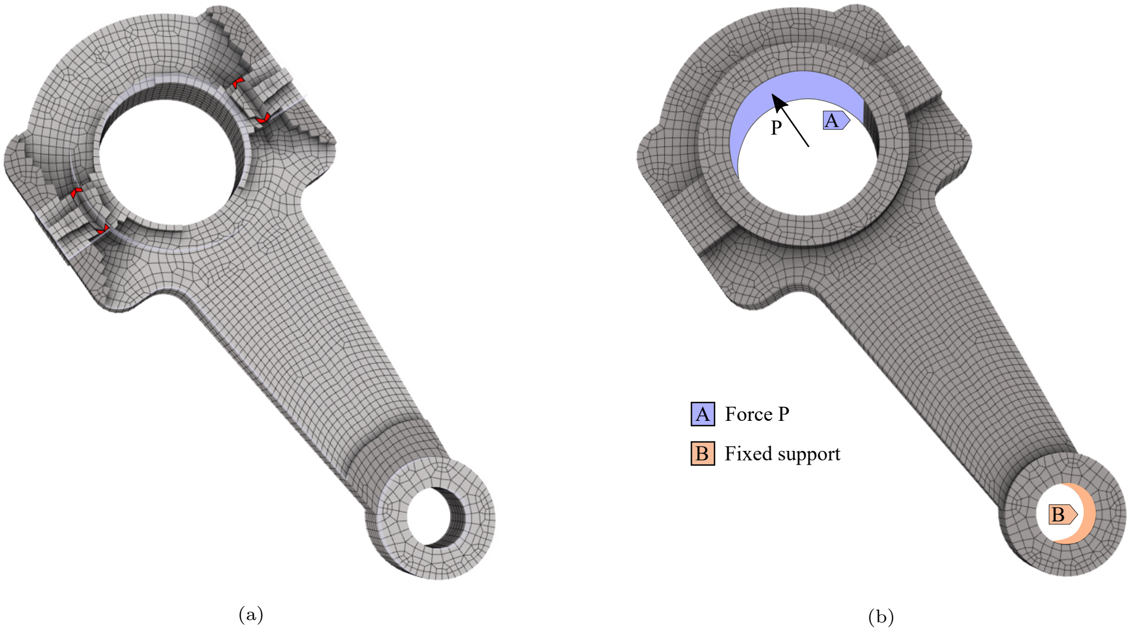
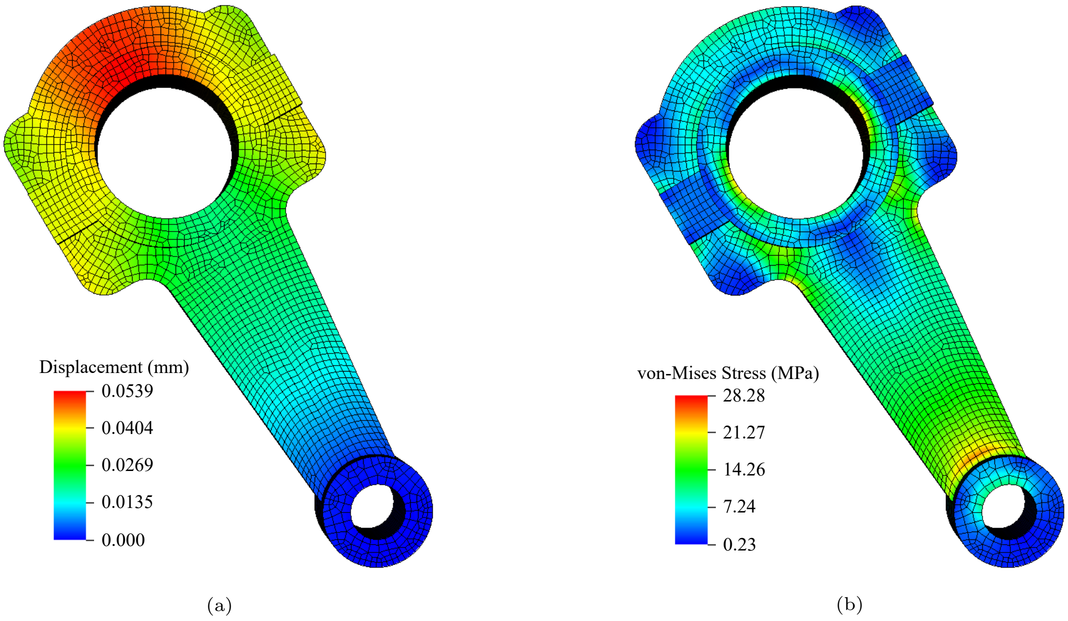


Figure 30: (a) Connecting rod mesh [7]; elements in red color are inverted. (b) Boundary conditions.

Then i-TFEM was employed to carry out a linear quasi-static structural analysis. The resultant displacement field is shown in Fig. 31a while the von-Mises stress field is plotted in Fig. 31b. The total computational time was 18.742s, out of which, 1.7s was spent processing the concave elements, i.e., the overhead due to i-TFEM was found to be minimal.



(a)

(b)

Figure 31: (a) Displacement and (b) von-Mises stress fields.

5. Conclusions

It is well known that the standard finite element method (FEM) leads to erroneous results when a mesh contains inverted elements. In this paper, an isoparametric tangled finite element method (i-TFEM) was presented to handle such inverted elements, specifically, inverted quadrilateral and hexahedral elements. i-TFEM exploits special properties of isoparametric elements to render the implementation much simpler and computationally efficient. Compared to the standard isoparametric FEM, i-TFEM has the following features.

1. The proposed i-TFEM framework replaces the full-invertibility requirement of the standard FEM with partial invertibility by (a) modifying the elemental stiffness matrices corresponding to the concave elements and (b) incorporating the piecewise compatibility condition.

2. i-TFEM reduces to the standard FEM for untangled meshes and can be easily adapted within existing FEM solvers.

3. i-TFEM passes the patch test and its convergence rate is found to be optimal even in the presence of severely tangled elements.

The accuracy and effectiveness of i-TFEM have been demonstrated through several 2D and 3D examples. Finally, the computational overhead of handling concave elements is found to be minimal in a practical 3D scenario.

Future work includes extending i-TFEM to large deformation problems where tangling is fairly common. Another possible extension is handling of curved tangled elements [30, 31, 32, 33]. It is also of interest to extend the concepts to iso-geometric analysis [34] where tangling is known to occur [35, 36, 37], and leads to erroneous results.

Acknowledgments

The authors would like to thank the support of National Science Foundation through grant 1715970.

References

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

[2] P. J. Frey, P.-L. George, Mesh generation: application to finite elements, Iste, 2007.

[3] R. D. Cook, et al., Concepts and applications of finite element analysis, John wiley & sons, 2007.

[4] D. S. Lo, Finite element mesh generation, CRC Press, 2014.

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

[6] 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.

[7] 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.

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

[9] 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) (2022).

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

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

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

[17] 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.

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

[19] 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.

[20] 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.

[21] 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.

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

[23] P. M. Knupp, Hexahedral and tetrahedral mesh untangling, Engineering with Computers 17 (3) (2001) 261–268.

[24] E. Ruiz-Gironés, X. Roca, J. Sarrate, R. Montenegro, J. M. Escobar, Simultaneous untangling and smoothing of quadrilateral and hexahedral meshes using an object-oriented framework, Advances in Engineering Software 80 (2015) 12–24.

[25] 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.

[26] 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.

[27] S. Hang, Tetgen, a delaunay-based quality tetrahedral mesh generator, ACM Trans. Math. Softw 41 (2) (2015) 11.

[28] M. T. Heath, Scientific Computing: An Introductory Survey, Revised Second Edition, SIAM, 2018.

[29] S. Timoshenko, J. Goodier, Theory of elasticity 3rd ed., 567 (1970).

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

[31] 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.

[32] 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.

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

[34] 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.

[35] 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.

[36] 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.

[37] 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.