Tangled Finite Element Methods for Handling Negative Jacobian Elements

by

Bhagyashree Chandrakant Prabhune

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Mechanical Engineering)

at the
University of Wisconsin-Madison
2023

Date of Final Oral Exam: 10 August 2023
The dissertation is to be approved by the following members of the Final Oral Committee:

Krishnan Suresh, Professor, Mechanical Engr., UW-Madison
Xiaoping Qian, Professor, Mechanical Engr., UW-Madison
Shiva Rudraraju, Asst. Professor, Mechanical Engr., UW-Madison
Dan Negrut, Professor, Mechanical Engr., UW-Madison
Pavana Prabhakar, Assoc. Professor, Civil and Environ. Engr., UW-Madison
Suzanne M. Shontz, Professor, Electrical Engr. and Computer Science, KU

Abstract

Finite element method (FEM) is a well-established technique for numerical analysis. Meshing is a critical step in FEM and is estimated to take over 80% of the overall analysis time for complex engineering designs. A mesh must meet several requirements; one of the critical requirements is that all elements of the mesh must be tangle-free (convex). Tangled (non-convex) elements, i.e. elements with negative Jacobian determinant, can lead to erroneous results in the FEM. Further, generating high-quality tangle-free meshes for complex geometries is often impossible.

Apart from mesh generation, tangled meshes can occur during mesh optimization, large deformation simulations, mesh morphing, shape optimization, finite element simulations involving large motion of the computational domain such as fluid flow analysis large deformation, metal forming, fluid-structure interaction, crash analysis, to name a few Hence there is a need to explore analysis methods that can directly handle such tangled meshes. Moreover, such analysis methods can simplify meshing of complex geometries.

The objective of this thesis is to extend the standard finite element formulation to allow the use of tangled meshes. In particular, I propose and explore the applications of *Tangled Finite Element Method* (TFEM). TFEM extends FEM through two concepts. First, the ambiguity of the field in the tangled region is resolved through a careful (re-)definition of the field. Second, a field compatibility constraint is imposed for all the tangled elements. When these two are incorporated, accurate results and optimal convergence rate can be achieved using TFEM, while requiring minimal changes to the existing FEM framework. Moreover, TFEM reduces to classical FEM for untangled meshes. Three variations of TFEM are presented: oriented-TFEM (o-TFEM), isoparametric TFEM (i-TFEM), and accelerated isoparametric TFEM (a-TFEM). TFEM has been applied to variety of element types such as 4-node quadrilateral, 9-node quadrilateral, 6-node triangular elements in two-dimensions and 8-node hexahedral element in three-dimensions. Moreover, with TFEM, tangled meshes have been employed for solving 2D and 3D elliptic problems (elastostatics, Poisson), hyperbolic (elastodynamics including damping) and eigen (free vibration) problems with linear material behavior as well as nonlinear elasticity problems with non-linear and near-incompressible material behavior. Numerous real-world tangled meshes are considered to demonstrate the robustness of the method.

Acknowledgments

Writing this document offers a precious opportunity to pause and reflect on the incredible people who have supported and encouraged me throughout the years.

First and foremost, I owe my sincere gratitude to my Ph.D. advisor, Prof. Krishnan Suresh. His unwavering support, guidance, advice, and trust have been instrumental in my personal and professional growth. His meticulousness in effectively communicating ideas is one of the greatest learnings for me. His courses are some of the best courses I have taken during my graduate studies. Apart from being an incredible researcher and teacher, he is a wonderful human being with a heart of gold. I am profoundly grateful to him for providing unwavering support and flexibility during challenging times. His genuine care for his students' well-being, availability to discuss any topic, prompt feedback, friendly nature made my Ph.D. journey truly enjoyable. Words cannot adequately express my appreciation for having such an outstanding advisor; he epitomizes what an advisor should be, and I am grateful to have the best.

I would also like to thank all my committee members, Prof. Xiaoping Qian, Prof. Suzanne Shontz, Prof. Shiva Rudraraju, Prof. Pavana Prabhakar, and Prof. Dan Negrut for their valuable feedback and support during the Ph.D. studies. I would like to especially thank Prof. Shiva Rudraraju for the outstanding courses on the finite element method; these courses have an important role to play in this thesis. I would like to thank Prof. Suzanne Shontz for providing examples of tangled and untangled meshes to test the proposed method.

My deepest gratitude goes to the three most important people in my life: my mother, father, and brother. They embody unconditional love, simplicity, sincerity, care, and sacrifice. To my parents, I thank them for their constant encouragement, unwavering support in every decision I've made even when it meant going against the traditional societal norms, standing behind me like a rock in every situation, and instilling a love for research and engineering. To my brother, Omkar, I owe a debt of gratitude for always providing a shoulder to lean on. He has been my best friend, counselor, and guide, and his presence in Madison during the latter half of my PhD journey has been invaluable. He has showered unconditional support and care, providing home-cooked food, sharing every little joy and sorrow, and made Madison truly a home. His unwavering support, friendship, and presence made this journey possible. I would also like to express my heartfelt thanks to my dear friends in Madison, Nitesh and Mukul, for their advice, friendship, support, and being there to share the ups and downs of the PhD life. I am deeply grateful to Abhijit and Aniket for being the best friends one could ever ask for. I extend my gratitude to Tanmayee for her cheerful presence and for putting up with me as a roommate in grad school.

I would like to thank former and current ERSL members for their support and camaraderie. Special thanks to Tej and Saketh for collaborating on projects and papers. I am grateful for the friendship of Prakarsh, Kunal, Nikhil, Deb, Rahul, Mohan. I would like to thank the faculty members from the College of Engineering Pune (COEP) for being such a great inspiration. My gratitude also goes to my school teachers, especially, Amrutkar Madam and Jain Madam. I thank Prof. K. Ramesh for providing an opportunity to work at IIT Madras as well as Dr. Aseem Ansari and IUSSTF for the S. N. Bose scholars' program, it was one of the most important turning points for me. I thank the friends from IIT Madras and the COEP CSAT team for their companionship. I thank the Brahmakumaris organization for their teachings that kept me going throughout the toughest phases!

Finally, I would like to express my deepest gratitude to Him, for being the ultimate teacher, guide, guardian, and friend. All the wonderful people in my life, be it teachers, family, or friends, are a result of His grace. When I pay tribute to these exceptional individuals, I am, in essence, praising His glory. His hope, light, and strength have been my driving force. This work is dedicated to Him with profound gratitude and reverence.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Finite element method (FEM) [69, 185] is a ubiquitous choice for solving a variety of engineering problems [6]. Despite its well-established mathematical principles and numerous computer implementations, there are fundamental challenges that need to be addressed. One such challenge is the generation of high-quality meshes for complex geometries, which can be excessively time-consuming, accounting for over 80% of the overall analysis time [70]. The underlying reasons are due to the stringent topological and geometric constraints imposed on the mesh [104, 185, 149]:

1. **Tangle-free**: The elements must be tangle-free (convex), ensuring validity of the finite element results.

2. **Mesh Quality**: The elements should be of high quality; see [85, 150] for details on mesh quality metrics. While FEM generally favors high-quality elements, some specific applications may benefit from low-quality elements.

3. **Geometric Conformance**: The mesh must accurately conform to the underlying geometry. Higher fidelity to the actual geometry ensures more reliable and accurate FEM results.

4. **Topological Validity**: The mesh must be topologically valid and well-structured. For instance:

(a) Hanging (spurious) nodes must not be present.

(b) Elements can only be connected to neighboring elements via their boundaries, e.g. the vertex of one triangle cannot be attached to the edge of another.

(c) When two neighboring elements are connected at a common boundary their respective node numberings for the boundary are opposite.

(d) Each boundary of an element must be connected to at most one other element.

5. **User Inputs**: : The mesh should follow user-defined inputs, such as element type, element size, growth rate, and other relevant parameters.

Addressing these constraints during the meshing process is important to ensure accurate and efficient solution using FEM. However, generating meshes which satisfy all the above requirements is challenging.

The first requirement necessitates that the Jacobian determinant remains positive throughout the mesh. In other words, tangled or negative-Jacobian elements are not allowed. The meshes that satisfy the topology requirements but contain tangled elements are referred to as 'tangled meshes'.

Fig. 1.1a and Fig. 1.1b illustrate examples of tangled meshes with 4-node quadrilateral and 8-node hexahedral elements, respectively. In these examples, the tangled elements are highlighted in red; observe that the tangled elements are non-convex.

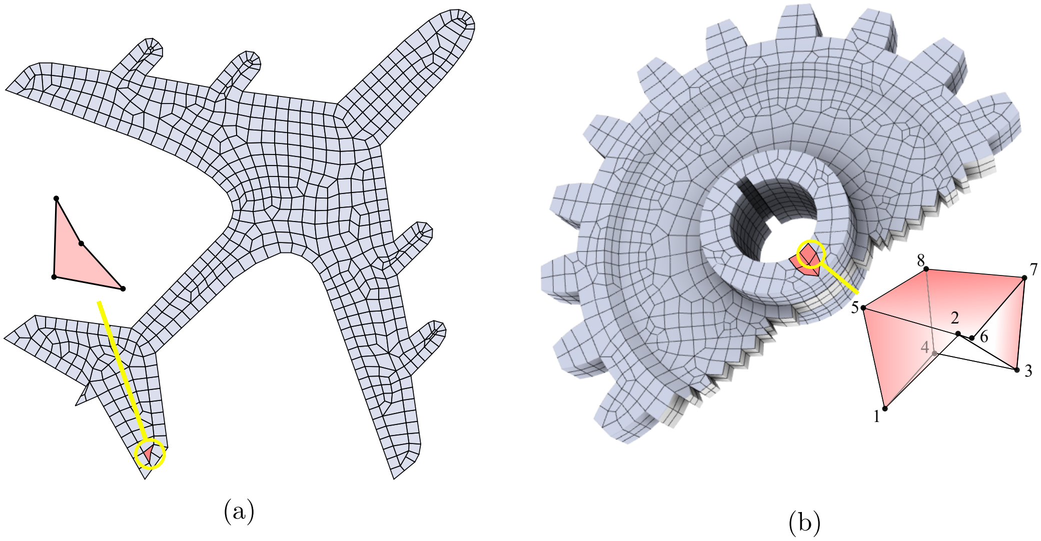


Figure 1.1: (a) Tangled mesh with 4-node quadrilateral elements; the mesh has been generated using the algorithm presented in [146]. (b) Tangled mesh with 8-node hexahedral elements; the mesh has been provided in [103]. Highlighted elements are tangled (negative Jacobian elements).

Presently, the use of tangled meshes is strongly and unanimously rejected by the FEM community. If a mesh contains even a single tangled element, it is considered invalid or unacceptable for FEM. To quote [17]: *“Because tangled meshes generate physically invalid solutions, it is imperative that such meshes are untangled”*. Indeed, as will be confirmed in the subsequent section, FEM yields erroneous results on a tangled mesh.

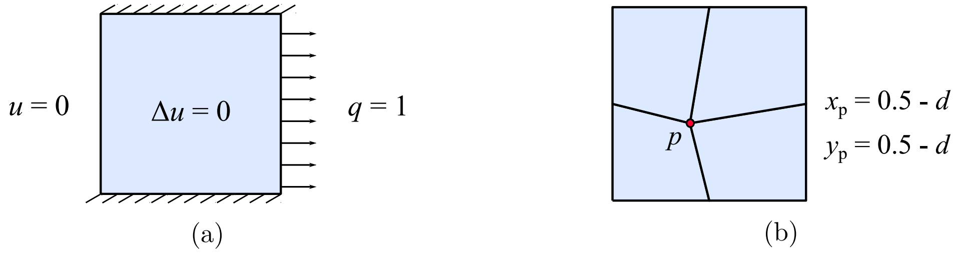
Unfortunately, mesh generators often struggle to produce tangle-free meshes [127, 20]. In an attempt to satisfy other mesh requirements (constraints 2-5 mentioned earlier), many mesh generators end up producing tangled meshes [159, 101, 108, 101, 48, 102, 59, 98, 81, 93, 117, 67]. The issue is particularly severe with 3D hexahedral meshes, as expressed in [127], stating, *“maintaining the inversion-free (tangle-free) property of hex-mesh poses a great challenge”*.

This thesis aims to revisit the first requirement and demonstrate that this requirement is not necessary. In other words, a mesh containing tangled elements but fulfilling the other validity conditions (requirements 2-5 outlined earlier) could be considered acceptable for use in FEM, provided that appropriate modifications are made to the finite element formulation and implementations. By reevaluating this aspect, the research seeks to offer insights into effectively incorporating tangled meshes into FEM simulations, potentially easing the burden of mesh generation and enabling more practical and efficient engineering analyses.

1.1 FEM and Tangled Elements

In this section, evidence is presented to demonstrate the current unacceptability of tangled meshes in FEM.

Consider a thermal conduction problem over a unit square domain $[0, 1]^2$ with a thermal conductivity of 1. The boundary conditions are as shown in Fig. 1.2a: the left-hand side is set to a temperature of $u = 0$, a thermal flux of 1 is applied on the right-hand side, and the top and bottom edges are insulated. The exact solution to this problem is $u(x, y) = x$.



(a) (b)

Figure 1.2: (a) Domain with boundary conditions. (b) Mesh.

A simple mesh is constructed using 4 elements as shown in Fig. 1.2b. The position of central node $\boldsymbol{p}$ is determined by the parameter d as shown in Fig. 1.2b; $d = 0$ indicating a perfect mesh. In Fig. 1.3, the relative nodal error is plotted with respect to d. For $d \leq 0.3$, all the elements are convex (tangle-free), and FEM provides machine precision accuracy. However, for $d > 0.3$, the element on the bottom left (shown in red in Fig. 1.3) becomes tangled and FEM error increases by an order of 10^{14}.

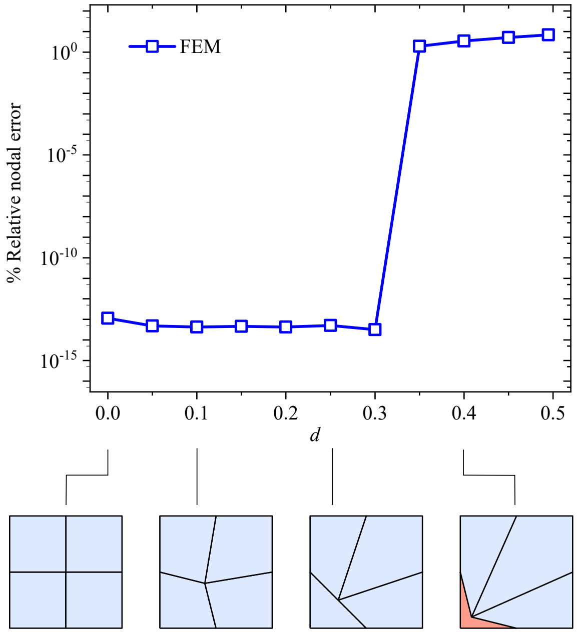
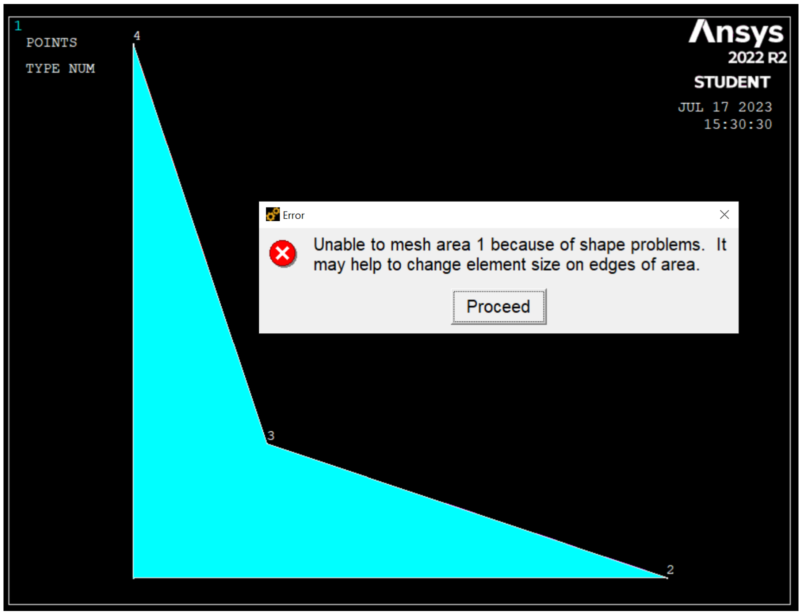


Figure 1.3: Relative error in FEM solution with respect to *d*.

Not surprisingly, commercial finite element solvers display errors or warnings, and terminate the simulation if tangled elements are encountered. For example, Fig. 1.4 illustrates ANSYS Mechanical (APDL) refusing to accept a 2D non-convex region as a valid shape for the 4-node quadrilateral element.



Unable to mesh area 1 because of shape problems. It may help to change element size on edges of area.

Figure 1.4: ANSYS displays an error prohibiting the construction of non-convex element.

1.2 Other Sources of Tangled Elements

The finite element mesh requirements extend beyond the mesh generation process. For example, to meet element quality requirements, meshes are generally passed through a quality optimizer. Various smoothing techniques, such as those presented in [87, 178, 1], aim to enhance mesh quality without altering the mesh connectivity. On the other hand, methods like edge flipping and re-parameterization, as discussed in [136, 161, 54], modify the mesh connectivity to improve quality.

However, modifying the mesh using these techniques can inadvertently introduce tangled elements into the mesh. To address this issue, there has been the development of quality optimizers that simultaneously enhance mesh quality and untangle the mesh; see [68, 1, 82, 178, 144, 103, 83, 105, 47] for examples of such optimizers.

In addition to mesh generation and optimization, tangled meshes frequently occur during numerous scenarios such as mesh morphing [154], shape optimization [111], and finite element simulations involving large motion of the computational domain, for instance, large deformation analysis. In such situations, in spite of starting with an ideal tangle-free mesh, the mesh can become tangled. Some of these situations are discussed next.

1.2.1 Mesh Morphing

An example where mesh tangling often occurs is during mesh morphing. The motivation behind mesh morphing stems from the need to perform FEA over multiple similar geometries; for example in design exploration of a car [145, 26]. Mesh morphing enables the reusability of an existing mesh for analyzing structures with multiple similar geometries, eliminating the need to generate a new mesh for each geometry [154]. This process involves transforming or morphing the initial mesh to conform to the new geometry. Since mesh generation is expensive, mesh morphing is widely employed to analyze structures with similar geometries [77, 18, 23, 171]. An advantage of mesh morphing lies in the one-to-one correspondence between mesh nodes before and after the morphing process, reducing the likelihood of computational errors [154]. Fig. 1.5 depicts an example of the mesh morphing process. The initial domain is illustrated in Fig. 1.5a, with the initial mesh in Fig. 1.5b. The inner boundary is now rotated in a counterclockwise direction; the new mesh is obtained by morphing the previous mesh as illustrated in Fig. 1.5c. Observe that some quadrilateral elements (shown in red) are tangled. Thus, mesh tangling can occur depending on the amount of morphing being performed. In addition, element quality can be adversely affected, and may require a pass through a quality optimizer. This, as previously stated, has its own issues with regards to tangling.

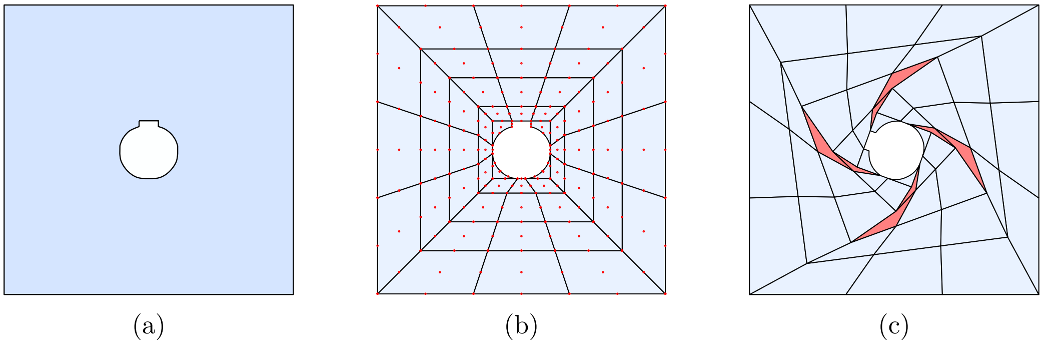


Figure 1.5: (a) Domain (b) Initial mesh containing Q9 elements (c) Morphed mesh.

1.2.2 Shape Optimization

Shape optimization involves determining the optimal shape of the structure by moving the boundary [166, 62]. At every step of optimization process, the geometry changes and the analysis model needs to be updated. Techniques such as remeshing, mesh morphing are commonly employed to obtain the mesh corresponding to the new geometry [145, 26, 46]. However, remeshing is computationally expensive [78] and can lead to sudden variation in the objective function, impeding smooth convergence to an optimal shape [34, 79]. On the other hand, the use of mesh morphing for large boundary variation can create tangled elements [28, 84, 3, 2, 31, 75]. Limiting the extent of boundary variations, while mitigating tangling issues, may constrain the allowed shape variations that would otherwise be permitted by the shape variable [75]. Moreover, the mesh is often passed through mesh optimizer to improve the quality which circles back to the previously described problem.

1.2.3 Large Deformation Simulations

Other sources of tangled meshes include FE analyses that involve large motion of the computational domain, such as large deformation analysis [153, 91, 167], fluid flow analysis [170], metal forming [64], fluid-structure interaction [65], and crash analysis [14]. In these applications, the finite element mesh is updated in an iterative process. In these applications, the occurrence of tangled meshes necessitates remeshing in order to proceed. Remeshing often leads to errors in the solution fields that are projected onto a new mesh; the prediction and control of these errors is yet another challenge [172].

In summary, tangling is unavoidable in modern FEA. The researchers have proposed various solutions to address this challenge, which will be discussed in the following section.

1.3 Current Solutions to Handle Tangled Meshes

The approaches to tackle tangled meshes can be divided in two main categories: (a) strategies to eliminate/avoid tangled elements, (b) development of numerical methods that can handle tangled elements.

1.3.1 Strategies to Avoid Tangled Meshes

Given the complexity of satisfying all the mesh requirements, certain mesh generation methods [61, 95, 110, 148] prioritize the tangle-free requirement above the other requirements (2-5 mentioned earlier). However, this prioritization may lead to geometric nonconformance, lower quality elements, and/or irregular topological structures, which are not advisable for FEA. Despite this prioritization and employing smoothing techniques to enhance mesh quality, generating a completely tangle-free mesh is not always feasible [127, 55, 122].

Mesh untangling

Numerous untangling algorithms have also been developed challenges [68, 178, 156, 155, 147, 165, 142, 143, 103, 1]. However, untangling is not always guaranteed, as reported in multiple instances where no tangle-free solution is possible [103, 1, 178, 140]. As emphasized in [140], *"... it is probably impossible to untangle the mesh under these hard constraints."* In fact, certain topological structures may not have an untangled mesh solution [103, 149]. Moreover, there are no known a priori tests to determine if a mesh can be untangled [86]. Finally, untangling can be expensive, and can pose challenges in mapping simulation data.

Meshfree methods

Computational mechanics community has actively worked towards developing methods to avoid meshing altogether through meshless or meshfree methods [114]. Examples of these methods include element-free Galerkin [16, 114], partition of unity based hp-cloud [57], generalized finite element method (GFEM) [15], to name of few.

Meshfree methods are characterized by the use of nodes without the mesh connectivity. These nodes are used to construct the function approximation (or shape functions). Typically an easy-to-generate background mesh is created for numerical treatment. This background mesh can be any grid that can be automatically generated, such as triangular grid, without the need to ensure high-quality elements.

In contrast to FEM, which relies on the same mesh for both integration and function approximation, meshfree methods decouple these aspects, allowing for a simpler background mesh.

Despite their advantages, meshfree methods do face certain challenges that limit their widespread application. One major challenge involves numerical integration. Since the regions of integration do not necessarily coincide with the integration mesh, controlling the error of numerical integration becomes particularly challenging.

1.3.2 Numerical Methods for Tangled Elements

An alternative approach to address tangled elements involves developing non-traditional FEM methods that can effectively handle such elements. An advantage of developing such analysis methods is that they enable mesh generators to focus on other constraints (constraints 2-5 listed earlier), leading to improved quality elements in critical regions of the mesh.

In the realm of computer graphics, the method of invertible finite elements [76, 157] has been proposed to handle tangled elements. However, it is primarily focused on visual correctness rather than accurate analysis.

In the context of finite element/finite volume methods, several approaches have been explored to directly handle tangled elements.

Signed volume in CFD

Here, during the stiffness matrix computation in standard FEM and equivalently in the finite volume method, the sign of Jacobian is not retained; instead, the absolute value is employed to ensure the element volume remains positive. It was suggested in [118] that retaining the sign of the Jacobian is sufficient to produce reasonable results over the negative Jacobian (i.e. negative volume) elements in the context of the finite volume method. Specifically, the author considered node-centered edge-based discretization over triangle meshes, and showed that by explicitly accounting for signed-volume, one can handle negative Jacobian elements, without a loss in accuracy. Additionally, it was shown that zero and negative-volume elements can be useful for applications such as discontinuity capturing, singularity resolutions, hanging nodes, and overset grids.

In the context of FEM, we refer this method as ‘signed FEM’ to indicate that the sign of Jacobian is retained. It produces accurate results in certain problems such as the thermal conduction example discussed earlier. However, it will lead to erroneous and/or non-physical solutions in other scenarios.

To illustrate, let us consider a plane strain elasticity problem involving a long hollow cylinder with internal radius $a = 1$ and external radius $b = 4$; as depicted in Fig. 1.6a. A uniform pressure $p = 1$ is applied to the inner surface ($r = a$). The material properties are specified with 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 [162]:

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

A 4-node tangled quadrilateral mesh, shown in Fig. 1.6b, is constructed.

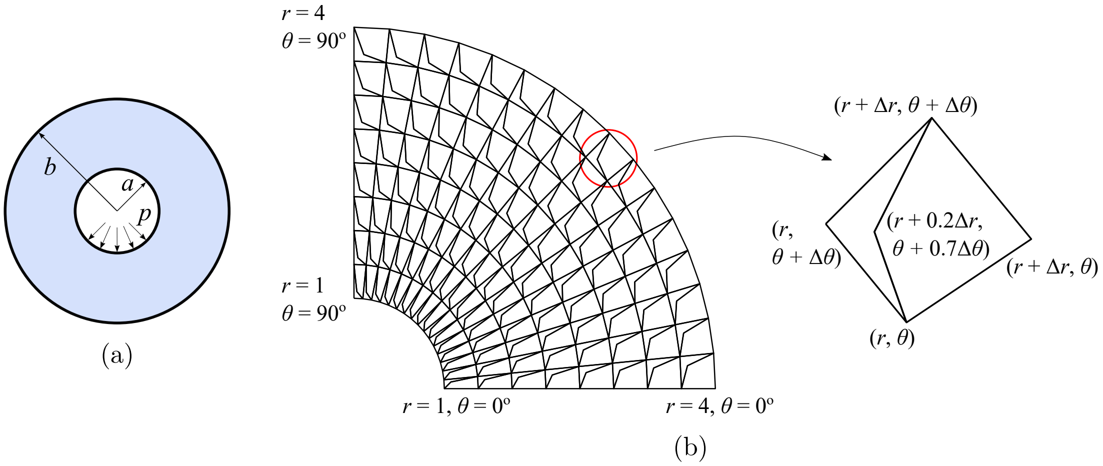


Figure 1.6: (a) Cross-section of the pressurized cylinder (b) Tangled mesh.

Fig. 1.7 illustrate the displacement field over the deformed mesh obtained via signed FEM. As one can observe, erroneous results are produced with signed FEM. Therefore, most commercial FEA solvers employ the absolute value of Jacobian, i.e., not signed Jacobian, for the stiffness matrix computation.

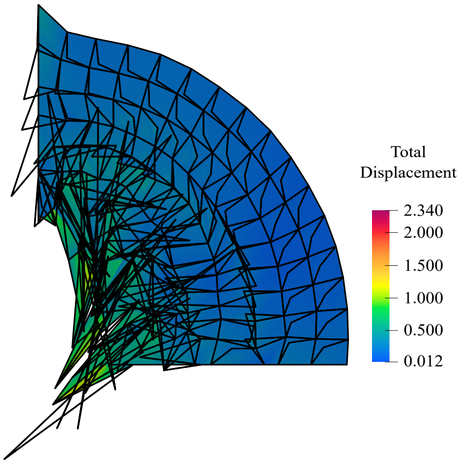


Figure 1.7: Deformed configuration for pressurized cylinder using signed FEM.

Non-traditional finite element methods for non-convex elements

Non-traditional methods which can potentially be used to handle certain non-convex elements include unsymmetric finite element [138, 137], smoothed finite element (SFEM) [96], polygonal finite element methods (PFEM) [51, 109, 33, 139], virtual element method (VEM) [13, 53, 124, 32, 123]. These methods represent an ongoing effort in the computational mechanics community to push the boundaries of FEM and extend its capabilities to tackle meshing issues. The method proposed in this thesis belongs to this category.

Unsymmetric FEM

Unsymmetric finite elements (referred to here as “UFEM”) are characterized by the use of two sets of shape functions, compatibility enforcing and completeness enforcing, to mitigate the effect of mesh distortion [138, 137, 179, 27, 183, 107]. They have been demonstrated to handle negative Jacobian quadrilateral elements in 2D elasticity problems. However, they have not yet been applied to 3D negative Jacobian elements where the meshing issue is more severe. Moreover, as the name suggests, they lead to asymmetric stiffness matrices. This restricts their applicability to real-world large-scale problems due to the limited availability of robust and efficient solvers.

Smoothed FEM (SFEM)

The SFEM is a family of methods, such as cell-based SFEM [96], node-based SFEM [40], edge-based SFEM [97], etc., relying on gradient smoothing technique [96, 29]. Since SFEM does not involve parametric mapping [96], it is less sensitive to distortion of elements and reduces the mesh generation burden. However, the process of numerical integration involves calculating the position of the Gaussian point and the value of the shape function on each smoothed domain surface, which makes the computational efficiency much lower than FEM [184]. Hence, it has been applied particularly to 2D and 3D simplex meshes, with a limited application to non-convex elements, especially in 3D [175].

Polygonal FEM (PFEM) and Virtual Element Method (VEM)

Methods such as PFEM and VEM are specifically developed for polygonal/polyhedral meshes. However, conventional shape functions in PFEM, such as Wachspress shape functions [50] and shape functions from the Mean Value coordinates [52], are limited to convex elements with planar faces. Some other shape functions used in PFEM can be used to interpolate over the non-concave element. However, the effectiveness of those shape functions in terms of simplicity, accuracy, and/or computational cost is not as good as the standard FEM [115]. Moreover, PFEM basis functions are non-polynomials (e.g. rational) and thus require high order numerical quadrature rules, which will greatly increase the computational cost [160, 19, 175]. On the other hand, VEM does not require explicit definitions and avoids numerical integration difficulty encountered in the conventional PFEM. The potential of these methods can be exploited further with the use of polytope meshes.

Polyhedral Meshes

The use of polyhedral meshes provides some flexibility in mesh generation, which can be leveraged by the use of methods such as PFEM, VEM, and nSFEM (a variation of SFEM). However, each method has its own criteria for “good” elements, and certain meshes may not meet these requirements, leading to reduced accuracy or even non-convergence [152, 4]. Generating suitable polyhedral meshes remains an active area of research [45, 99, 56, 7, 119]. Moreover, they may not entirely avoid tangled elements, especially in scenarios such as large deformation simulations and mesh morphing.

Overall, these methods attempt to partly relax one or more meshing constraints (discussed at the beginning of the chapter). For instance, the unsymmetric FEM attempts to relax the constraints related to tangling (constraint 1) and mesh distortion (constraint 2). Similarly, methods such as SFEM, and VEM attempt to offer flexibility in element shapes (constraints 1 and 2) [80, 180, 25] and allow hanging nodes (constraint 4) [74, 72, 112]. These are summarized in Table 1.1.

While the non-traditional methods provide promising solutions, meshing (tangling) challenges persist. There is a gap between the meshing requirements and meshing capabilities. For instance, practical meshes generated from state-of-the-art mesh generation algorithms, often contain 'penetrating' tangled elements (discussed in the next chapter), which are not addressed by these approaches. Bridging the gap between meshing capabilities and analysis requirements remains an open challenge.

Moreover, it is difficult to integrate proposed non-traditional methods with existing FEM framework, making them less practical. There is a need for a method that can handle tangled meshes while still leveraging the advantages of existing FEM tools.

In this thesis, I propose a new framework, namely, *tangled finite element methods* (TFEM), which specifically addresses tangled elements (constraint 1) by modifying the traditional FEM approach. It reduces to standard FEM for non-tangled meshes and can effectively handle real-world tangled meshes. A comparison of TFEM against other methods discussed above is presented in the Table 1.1.

Table 1.1: Comparison of non-traditional methods for non-convex elements.

	UFEM	SFEM	PFEM	VEM	TFEM
Handles penetrating elements	×	×	×	×	✓*
Uses FEM basis functions	✓	×	×	×	✓
Reduces to std. FEM for tangle-free meshes	×	×	×	×	✓
Symmetric stiffness matrix	×	✓	✓	✓	✓
Addresses meshing constraint #	(1, 2)	(1, 2, 4)	(1, 4)	(1, 2†, 4)	(1)
Handles polyhedral elements	×	✓	✓	✓	×

(*Accelerated isoparametric TFEM, to be discussed, handles penetrating elements.)
(† Certain requirements must be met for 'good' elements; however more flexible in comparison to the standard FEM.)

1.4 Proposed Method

The thesis proposes a family of *tangled finite element methods* (TFEM), to effectively handle the meshes generated by existing mesh generators. TFEM uses the same basis functions as standard FEM, and extends FEM through two key concepts: (1) resolving the ambiguity of the field in the tangled region with a careful re-definition, and (2) imposing a field compatibility constraint for all tangled elements. By incorporating these concepts, TFEM achieves accurate results and optimal convergence rates, requiring minimal changes to the existing FEM framework.

Three variations of TFEM are proposed: oriented TFEM (o-TFEM), isoparametric TFEM (i-TFEM), and accelerated-isoparametric TFEM (a-TFEM). Each variation differs in the stiffness matrix computation. Among these, a-TFEM stands out, and is the recommended method, as it effectively handles real-world tangled meshes, including penetrating elements, with minimal computational overhead. A brief comparison of the methods is presented in Table 1.2. Each of these methods is discussed in detail in the subsequent chapters.

Table 1.2: Comparison of three variations of Tangled Finite Element Method.

	o-TFEM (oriented TFEM)	i-TFEM (isoparametric TFEM)	a-TFEM (accelerated i-TFEM)
Publications	[129]	[130, 133]	[131, 134, 132]
Thesis Chapter	3	4	5, 6, 7, 8
Stiffness matrix $\boldsymbol{K}$	$\boldsymbol{K}^0 + \boldsymbol{K}^S + \boldsymbol{K}^N$	$\widehat{\boldsymbol{K}}_{\text{concave}}$	Std FEM $\boldsymbol{K}$ with signed Jacobian
Nature of $\boldsymbol{K}$	Symmetric Indefinite	Symmetric Positive definite	Symmetric Indefinite
Compatibility constraint	Needed	Needed	Needed
Handles penetrating element	✗	✗	✓
Accurate for element with:			
(a) Straight edges or planar surfaces	✓	✓	✓
(b) Curved edges or non-planar surfaces	✗	✗	✓
Cause of inaccuracy	Area approximation	Area approximation	N/A
Optimal convergence	✓	✓	✓
Implementation	Difficult	Easy	Easiest
Speed	Slow	Fast	Fastest

1.5 Publications

The following journal articles (published or under review) are part of this thesis:

1. Reference [129]: Bhagyashree Prabhune, Sridhara Saketh, and Krishnan Suresh. "Tangled finite element method for handling concave elements in quadrilateral meshes." *International Journal for Numerical Methods in Engineering* 123.7 (2022): 1576-1605.

   The article is covered in chapter 3.

2. Reference [130]: Bhagyashree Prabhune and Krishnan Suresh. "A computationally efficient isoparametric tangled finite element method for handling inverted quadri-

lateral and hexahedral elements." *Computer Methods in Applied Mechanics and Engineering*405 (2023): 115897.

The article is covered in chapter 4.

3. Reference [131]: Bhagyashree Prabhune and Krishnan Suresh. "An isoparametric tangled finite element method for handling higher-order elements with negative Jacobian." *Computational Mechanics* (2023): 1-18.

   The article is covered in chapter 5.

4. Reference [134]: Bhagyashree Prabhune and Krishnan Suresh. "On why mesh untangling may not be required." *under review, Engineering With Computers.*

   The article is covered in chapter 6.

5. Reference [132]: Bhagyashree Prabhune and Krishnan Suresh. "Free and Forced Vibration Analysis over Meshes with Tangled (Non-Convex) Elements." *under review, Journal of Sound and Vibrations.*

   The article is covered in chapter 8.

6. Reference [133]: Bhagyashree Prabhune and Krishnan Suresh. "Isoparametric Tangled Finite Element Method for Nonlinear Elasticity." *in preparation, arXiv preprint arXiv:2303.10799.*

   The article is covered in chapter 7.

The following journal articles were published during my Ph.D. study, but are not part of this thesis:

7. Bhagyashree Prabhune and Krishnan Suresh. "A fast matrix-free elasto-plastic solver for predicting residual stresses in additive manufacturing." *Computer-Aided Design* 123 (2020): 102829.

8. Tej Kumar, Saketh Sridhara, Bhagyashree Prabhune, and Krishnan Suresh. "Spectral decomposition for graded multi-scale topology optimization." *Computer Methods in Applied Mechanics and Engineering* 377 (2021): 113670.

Chapter 2

Proposed Method

It has been shown in the previous chapter that the standard FEM leads to erroneous results for the tangled elements. In this chapter, we discuss the underlying reasons for this and propose the basic tenets of the proposed tangled finite element method (TFEM).

2.1 Parametric Mapping for Tangled Elements

To begin with, let us consider a 4-node bilinear quadrilateral element as shown in Fig. 2.1a. This element is located in the physical space (x, y). To facilitate the numerical integration, the finite element method usually employs a standard parametric space (ξ, η); see Fig. 2.1b. Let ϕ be the mapping from parametric space to the physical space of the element.

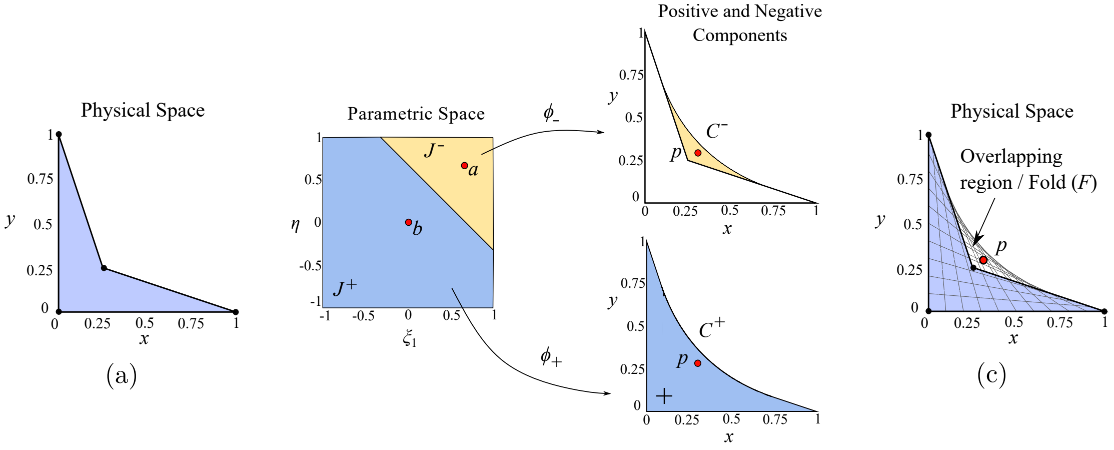
Given the coordinates of the four nodes (x^i, y^i) of a quad element, recall that the parametric mapping is defined via the standard bilinear shape functions $N^i(\xi, \eta)$ [185]:

$$x(\xi,\eta) = \sum_{i=1}^{4} N^i(\xi,\eta)x^i; \quad y(\xi,\eta) = \sum_{i=1}^{4} N^i(\xi,\eta)y^i \tag{2.1}$$

For the particular concave element in Fig. 2.1a this reduces to:

$$x\,(\xi,\eta) = \frac{(1+\xi)(5-3\eta)}{16}; \qquad y\,(\xi,\eta) = \frac{(5-3\xi)(1+\eta)}{16} \tag{2.2}$$

There are several implications of this mapping.



(a)

(b)

(c)

Figure 2.1: (a) Physical space of the concave Q4 element. (b) Parametric space of the concave element. Parametric space can be divided into positive and negative Jacobian regions. Mapping of these regions to physical space results. (c) Concave element with overlapping region.

First, note that the determinant of the Jacobian associated with this mapping is given by:

$$|\boldsymbol{J}| = \begin{vmatrix} x_{,\xi} & y_{,\xi} \\ x_{,\eta} & y_{,\eta} \end{vmatrix} = \frac{2 - 3\xi - 3\eta}{32}. \tag{2.3}$$

Therefore, $|\boldsymbol{J}|$ 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. 2.1b. The positive and negative Jacobian regions are denoted as J^+ and J^- respectively. Thus, the determinant of the Jacobian changes sign within the element. Although $|\boldsymbol{J}| = 0$ corresponds to a straight line in the parametric space, the corresponding curve in the physical space is quadratic (see Fig. 2.1b and 2.1c).

Second, all points in J^- map to points outside the element. For example, the point $\boldsymbol{a}$ $(\xi = 2/3, \eta = 2/3)$ in the parametric space maps to the point $\boldsymbol{p}$ $(x = 5/16, y = 5/16)$ that is outside the concave element as depicted in Fig. 2.1b.

Third, for every point $\boldsymbol{a}$ in the J^-, there is a corresponding point $\boldsymbol{b}$ in J^+ that maps to the same physical point. As illustrated in Fig. 2.1b parametric points $\boldsymbol{a}$ $(\xi = 2/3, \eta = 2/3)$ and $\boldsymbol{b}$ $(\xi = 0, \eta = 0)$ map to the same physical point $\boldsymbol{p}$ $(x = 5/16, y = 5/16)$. Since two different points in the parametric space map to the same physical point, the mapping is non-invertible. In other words, the element overlaps with itself, creating a folded region F as illustrated in Fig. 2.1c.

Definition: Given these observations, we define a positive (negative) component $C^{+}(C^{-})$ as the set of points in the physical space that map from J^{+} (J^{-}) region of the parametric space.

Now, consider a two-element mesh with one tangled element (E_1) as shown in Fig. 2.2a.

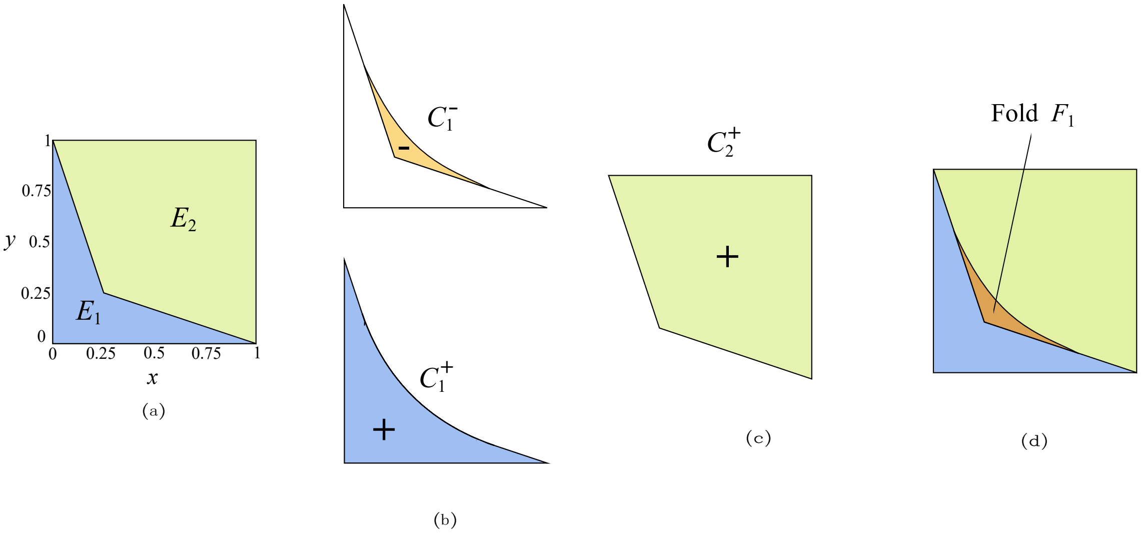


Figure 2.2: (a) 2-D domain discretized into two bilinear quads. (b) Positive and negative $|\boldsymbol{J}|$ regions of the concave element. (c) Convex element of the mesh. (d) Final physical space.

Observe that E_1 can be expressed as the difference between the two components (see Fig. 2.2b):

$$E_1 = C_1^{+} - C_1^{-} \tag{2.4}$$

Further, it can be observed from Fig. 2.2b that

$$C_1^{+} \cap C_1^{-} = C_1^{-} \tag{2.5}$$

since C_1^{-} is the subset of C_1^{+}. On the other hand, the convex element E_2 has only the positive component (see Fig. 2.2c):

$$E_2 = C_2^+;\ C_2^- = \varnothing. \tag{2.6}$$

Finally, since the folded region C_1^- lies outside the concave element, it overlaps with the neighboring convex element as illustrated in Fig. 2.2d. In this case, we have only one neighboring element. But in general, the folded region can overlap with multiple neighboring elements as illustrated in the next section.

In summary, for a concave bilinear quad element, (1) $|\boldsymbol{J}|$ takes both positive and negative values, (2) all points in the negative $|\boldsymbol{J}|$ space map to points outside the physical element, (3) the parametric mapping is non-invertible, and (4) a mathematical fold exists in the physical space which makes the element not only self-intersect, but also intersect with the neighboring element(s). FEM does not account for the non-invertible mapping and the folded region and hence, provides inaccurate solutions over the tangled mesh.

2.2 Types of Tangled Elements

Next, we discuss the types of tangled (non-convex) element configurations. Since 2D elements exhibit limited types of tangling, we consider the example of a three-dimensional 8-node hexahedral (H8) element.

Consider the tangled H8 element in the physical space (x, y, z) in Fig. 2.1a, and the parametric mapping ϕ from the (ξ, η, ζ) space in Fig. 2.3b to the tangled element. Here too, due to the non-convex nature of the element, one can show that 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. 2.3c. 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. 2.3c), similar to the 2D example considered earlier.

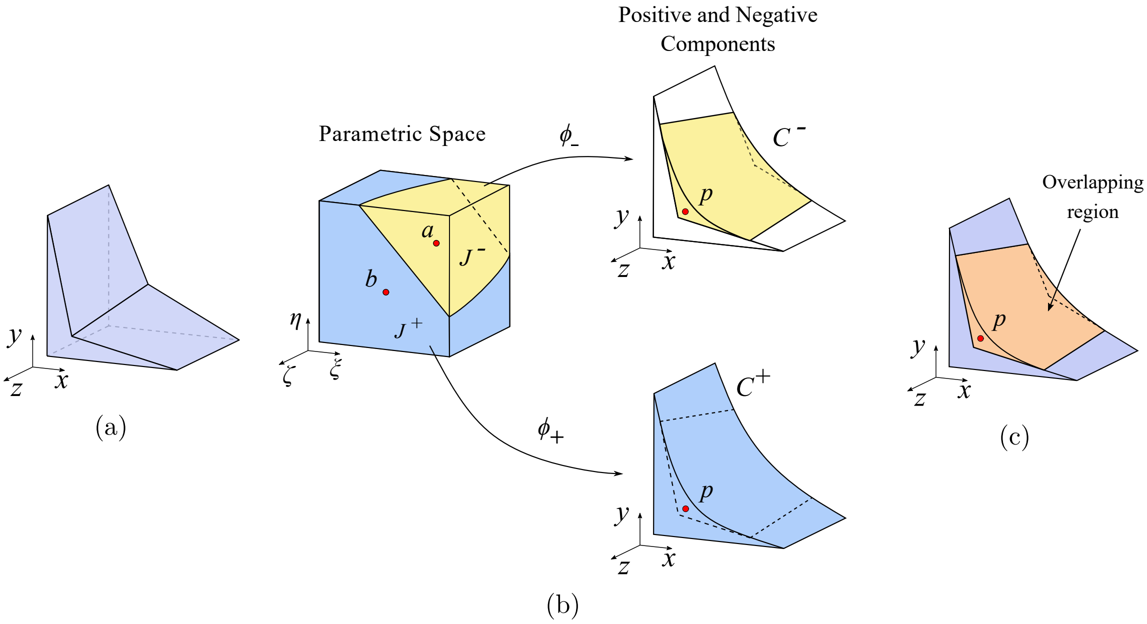


Figure 2.3: (a) Physical space of the tangled H8 element. (b) Parametric space of the tangled element, that can be divided into positive and negative Jacobian regions. Corresponding physical space with positive and negative components. (c) Tangled element with the overlapping region.

Similar to the 2D example, here too, 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.

However, in many real-world meshes, the negative component C^- is *not* entirely contained lie within the positive component C^+. We refer to such elements as 'penetrating-tangled elements'. Figure 2.4a illustrates a penetrating element along with the corresponding parametric space in Fig. 2.4b. One can show that for this element, only a part of J^- region, shown in yellow in Fig. 2.4c maps to an overlapping region within C^+ that lies outside the element, while the remainder J^-, shown in red in Fig. 2.4c, maps to a region physically within the element.

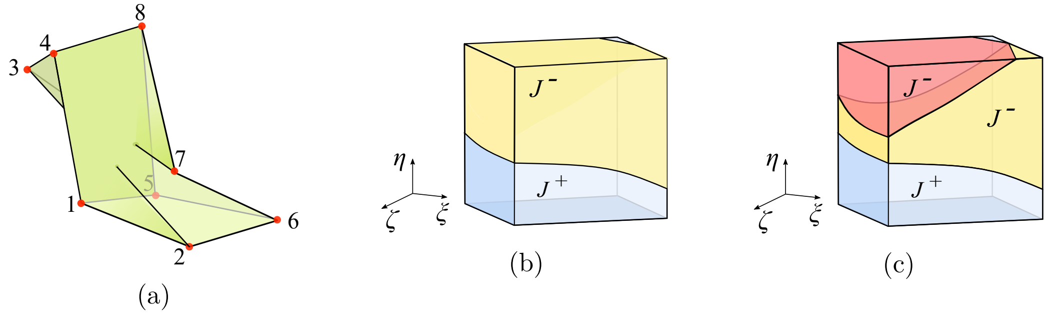


Figure 2.4: (a) Penetrating element (b) its parametric space; J^- region is shown in yellow color (c) J^- region which does not map to the fold is shown in red color.

Furthermore, there are cases where elements have disconnected J^- regions; Fig. 2.5 illustrates one such penetrating element. However, no special treatment is needed for such elements.

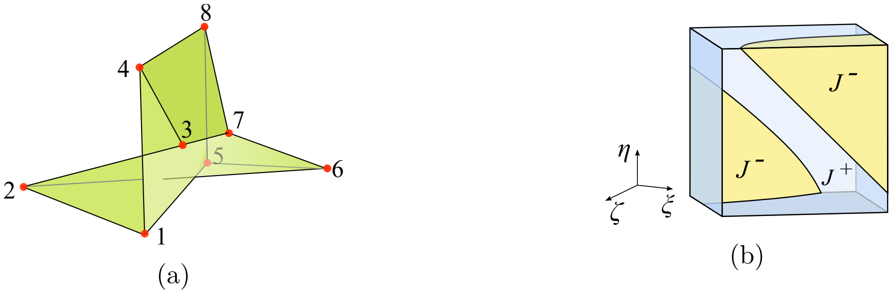


Figure 2.5: (a) Penetrating element with disconnected J^- regions, (c) its parametric space; J^- shown in yellow color.

Penetrating elements are frequently encountered in real-world meshes, and the proposed 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* considered in this thesis, the method to handle such fully inverted triangular and tetrahedral elements is provided in [42, 41]. Further, the default case where J^- is null corresponds to the non-tangled case, and TFEM reduces to standard FEM. In summary, three cases are possible:

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. 2.3a).

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. 2.4c).

Finally, it is also common for the overlapping region to be shared by multiple non-tangled elements as illustrated in Fig. 2.6. 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.

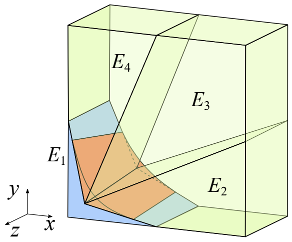


Figure 2.6: Overlapping region shared by multiple non-tangled elements.

2.3 TFEM Concept

In this section, we discuss the main concepts of TFEM to handle tangled elements. To aid the discussion, we consider the 2D quadrilateral element depicted in Fig. 2.1a.

The main idea in TFEM is to treat the positive and negative Jacobian regions (J^+ and J^-) separately since the two independent mappings

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

are both invertible. Moreover, since a point within an element can belong to two different parametric regions, the notion of shape functions at a physical point needs to be clarified. We, therefore, propose the following definitions.

Definition: For a tangled element E_j, let $\boldsymbol{N}_j(\boldsymbol{\xi})$ be the standard shape functions (i.e., bilinear Lagrange shape functions for Q4 element) defined over the parametric space. Now, let $\boldsymbol{N}_j^{\pm}$ be the restriction of $\boldsymbol{N}_j$ to $J^{\pm}$, i.e.,

$$\boldsymbol{N}_j^{\pm}(\boldsymbol{x}) \coloneqq \boldsymbol{N}_j(\phi_{\pm}^{-1}(\boldsymbol{x})) \tag{2.7}$$

For a 4-node quadrilateral element, $\boldsymbol{N}_j^{+}$ are the four shape functions of element E_j evaluated at point $\boldsymbol{p}$ corresponding to the component C_j^{+}, while $\boldsymbol{N}_j^{-}$ are the four shape functions evaluated at point $\boldsymbol{p}$ corresponding to the component C_j^{-}.

For example, in Fig. 2.1b, $\boldsymbol{N}_1^{+}(\boldsymbol{p})$, where $\boldsymbol{p} = (x = 5/16, y = 5/16)$, are the shape functions of element E_1 evaluated at $\boldsymbol{b}(\xi = 0,\ \eta = 0)$ whereas $\boldsymbol{N}_1^{-}(\boldsymbol{p})$ are the shape functions of element E_1 evaluated at $\boldsymbol{a}(\xi = 2/3,\ \eta = 2/3)$.

For a non-tangled (convex) element E_j, $\boldsymbol{N}_j^{+}(\boldsymbol{p})$ are the four shape functions of element E_j evaluated at point $\boldsymbol{p}$, while $\boldsymbol{N}_j^{-}(\boldsymbol{p})$ are assumed to be zero since the component C_j^{-} does not exist.

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

$$\boldsymbol{u}_j^{\pm}(\boldsymbol{x}) \coloneqq \boldsymbol{N}_j^{\pm}(\boldsymbol{x})\hat{\boldsymbol{u}}_j \tag{2.8}$$

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

2.3.1 Field Continuity in a Tangled Mesh

Given these definitions, we now examine the field at the re-entrant corner of the tangled element.

First, consider a two-element mesh where both the elements are convex (tangle-free) as shown in Fig. 2.7. When two non-tangled (convex) elements share a common node (e.g. node 5 in Fig. 2.7), the shared node can be mapped to only one parametric point of each element. Hence displacement continuity is automatically ensured.

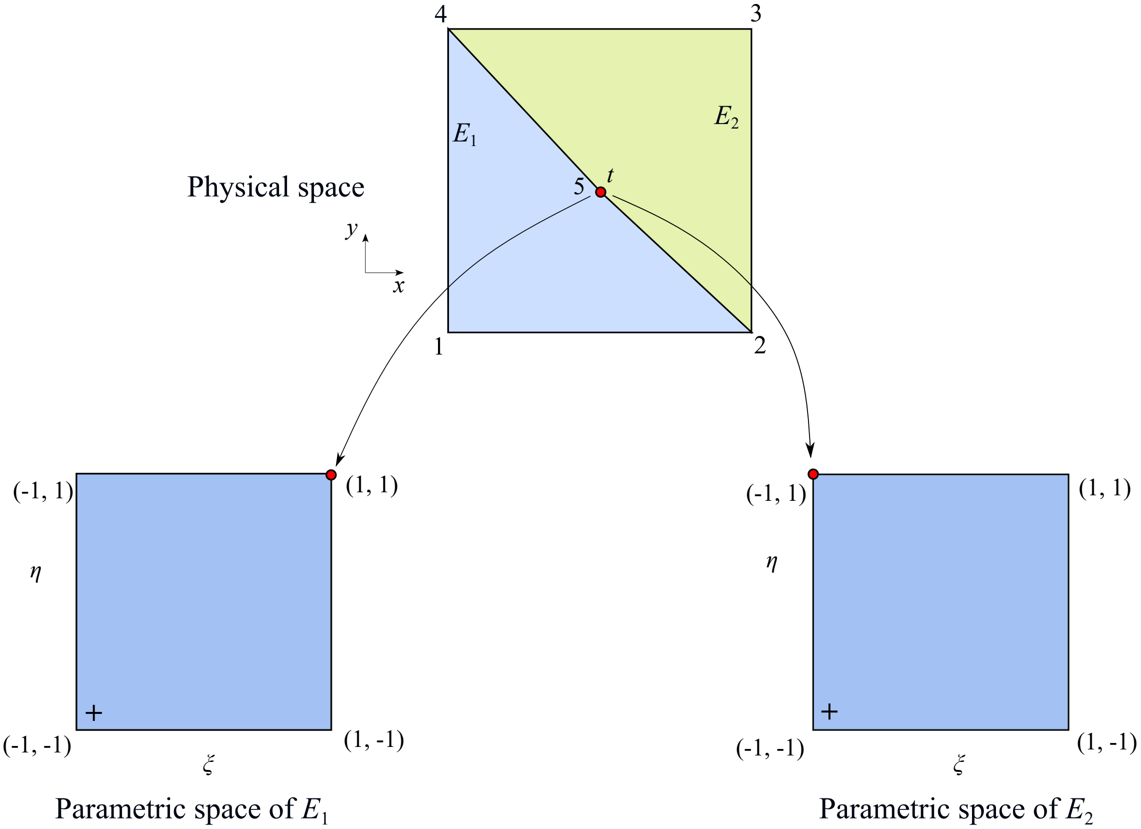


Figure 2.7: Field continuity for non-tangled element.

However, for the mesh containing a concave element as shown in Fig. 2.8, the shared node corresponds to a total of three points in the parametric space, instead of just two.

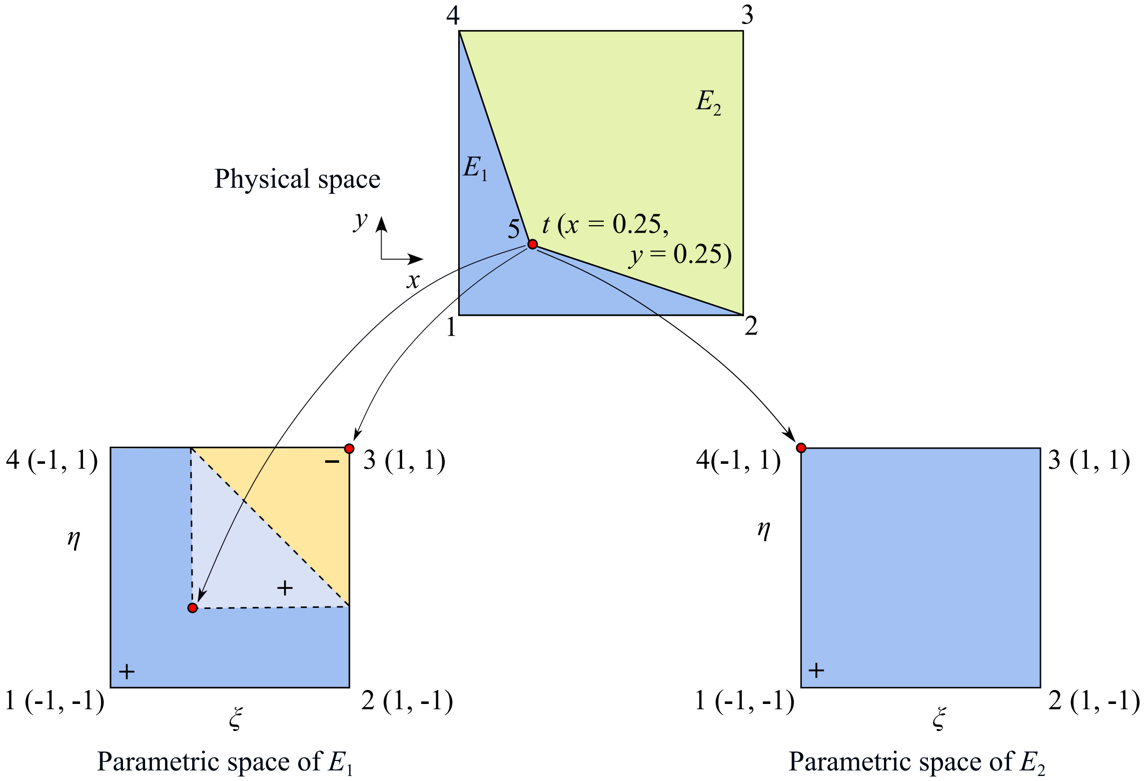


Figure 2.8: Field continuity for tangled element.

To elaborate, consider the re-entrant node 5 in Fig. 2.8. Node 5 corresponds to the physical location $\boldsymbol{t}$ $(x = 0.25, y = 0.25)$. Considering the parametric mapping for the convex element E_2, point $\boldsymbol{t}$ is mapped to point (-1, 1) in the parametric space. On the other hand, if we consider the parametric mapping for the concave element E_1, we observe that point $\boldsymbol{t}$ corresponds to two points in the parametric space: (1, 1) and (-1/3, -1/3). Thus, considering all three parametric points, the field at point $\boldsymbol{t}$ is:

(a) $u(\boldsymbol{t}) = \boldsymbol{N}_2(-1,1)\hat{\boldsymbol{u}}_2$

(b) $u(\boldsymbol{t}) = \boldsymbol{N}_1(1,1)\hat{\boldsymbol{u}}_1 = \boldsymbol{N}_1^-(\boldsymbol{t})\hat{\boldsymbol{u}}_1$

(c) $u(\boldsymbol{t}) = \boldsymbol{N}_1(-1/3,-1/3)\hat{\boldsymbol{u}}_1 = \boldsymbol{N}_1^+(\boldsymbol{t})\hat{\boldsymbol{u}}_1$

Here, (a) = (b) due to the inherent property of inter-element displacement continuity. However, (b) = (c) is not automatically ensured. Hence additional constraint is required to ensure field compatibility in the presence of tangled elements. The equation (b) = (c) can be written as,

$$\boldsymbol{N}_1^-(\boldsymbol{t})\hat{\boldsymbol{u}}_1 = \boldsymbol{N}_1^+(\boldsymbol{t})\hat{\boldsymbol{u}}_1 \tag{2.9}$$

Eq. 2.9 is referred to here as the *compatibility* condition that must be applied for all tangled elements in the mesh. This constraint is incorporated in the TFEM formulation (discussed in the next chapters).

To elaborate Eq. 2.9 further, let us expand the shape function and displacement vectors for element E_1:

$$\boldsymbol{N}_1 = [N_1^1\ N_1^2\ N_1^3\ N_1^4], \quad \hat{\boldsymbol{u}}_1 = [u^1\ u^2\ u^5\ u^4]$$

Note that the subscript denotes the element number while the subscript denotes the node number (local node number for shape functions and global node number for displacements). Accordingly, Eq. (b) can be expanded as

$$u(\boldsymbol{t}) = \boldsymbol{N}_1^-(\boldsymbol{t})\hat{\boldsymbol{u}}_1 = N_1^{3+}(\boldsymbol{t})u^5,$$

while Eq. (c) can be expanded as

$$u(\boldsymbol{t}) = \boldsymbol{N}_1^+(\boldsymbol{t})\hat{\boldsymbol{u}}_1 = N_1^{1+}(\boldsymbol{t})u^1 + N_1^{2+}(\boldsymbol{t})u^2 + N_1^{4+}(\boldsymbol{t})u^4 + N_1^{3+}(\boldsymbol{t})u^5.$$

We impose (b) = (c), resulting in

$$N_1^{1+}(\boldsymbol{t})u^1 + N_1^{2+}(\boldsymbol{t})u^2 + N_1^{4+}(\boldsymbol{t})u^4 + (N_1^{3+}(\boldsymbol{t}) - 1)u^5 = 0 \tag{2.10}$$

This constraint is imposed for every tangled element.

The concepts discussed in this chapter (piece-wise invertible mapping, resulting field definitions, and the compatibility constraint) form the basis of TFEM formulation and will be employed in all the three variations of TFEM, discussed in the subsequent chapters.

Chapter 3

Oriented Tangled Finite Element Method

3.1 Oriented-TFEM Framework for Tangled Elements

We discuss the formulation of oriented TFEM (o-TFEM) for the Poisson problem.

3.1.1 Standard FEM Formulation

The weak form for the Poisson problem is:

Find $u \in H_0^1(\Omega)$

$$\int\limits_{\Omega} (\nabla v)^{\top} \boldsymbol{D} \, (\nabla u) \, \mathrm{d}\Omega = \int\limits_{\Omega} vb \, \mathrm{d}\Omega \tag{3.1}$$

$\forall v \in H_0^1(\Omega)$

where, $\Omega \subset \mathbb{R}^n, n = 1, 2, 3$ is a domain and $\boldsymbol{D}$ is a constitutive matrix. To solve the above problem over a finite element mesh, recall that the field at any point within a mesh element E_j is approximated using shape functions as follows:

$$u^h(\boldsymbol{p}) = \boldsymbol{N}_j(p)\hat{\boldsymbol{u}}_j \tag{3.2}$$

where, $\boldsymbol{N}_j$ is a sparse row vector of length equal to the total number of degrees of freedom, with only m non-zero entries representing the shape functions associated with the m nodes of the element E_j. Similarly, $\hat{\boldsymbol{u}}_j$ is a sparse row-vector of the same length, and the non-zero entries are the unknown nodal degrees of freedom associated with E_j.

In the Galerkin formulation [185], this leads to:

$$\left[\int_{\Omega}\left(\nabla\sum_{j}\boldsymbol{N}_j\right)^{\top}\boldsymbol{D}\left(\nabla\sum_{k}\boldsymbol{N}_k\right)d\Omega\right]\hat{\boldsymbol{u}} = \int_{\Omega}\left(\sum_{j}\boldsymbol{N}_j\right)^{\top}f d\Omega \tag{3.3}$$

Upon numerical integration, this yields a linear system of equations:

$$\boldsymbol{K}^0\hat{\boldsymbol{u}} = \boldsymbol{f}^0 \tag{3.4}$$

where $\hat{\boldsymbol{u}}$ represent unknown nodal degrees of freedom, and

$$\boldsymbol{K}^0 = \prod_{Assemble}\int_{E_j}\nabla\boldsymbol{N}_j^{\top}\boldsymbol{D}\nabla\boldsymbol{N}_j d\Omega, \tag{3.5}$$

$$\boldsymbol{f}^0 = \prod_{Assemble}\int_{E_j}\boldsymbol{N}_j^{\top}f d\Omega. \tag{3.6}$$

For the tangled mesh, this formulation leads to erroneous results (shown later in the numerical experiments).

3.1.2 o-TFEM Field Definition

As noted in the previous chapter, 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, the field is ambiguous in the tangled/folded region. Hence, it is important to resolve this ambiguity. In TFEM, the field in the tangled region is explicitly defined so that it is no longer ambiguous.

In o-TFEM, the field at any point $\boldsymbol{p}$ is defined as the oriented sum of the field values from all *components* the point belongs to:

$$u^h(\boldsymbol{p}) \equiv \sum_{j|p\in C_j^+} \boldsymbol{N}_j^+(\boldsymbol{p})\hat{\boldsymbol{u}}_j - \sum_{j|p\in C_j^-} \boldsymbol{N}_j^-(\boldsymbol{p})\hat{\boldsymbol{u}}_j \tag{3.7}$$

We refer to Eq. 3.7 as the *unambiguity* condition, i.e., given the nodal values, the field is now unambiguously defined everywhere. This will be used later to derive the o-TFEM stiffness matrix. Observe that the first term on RHS of Eq. 3.7 corresponds to C_j^+, and the second to C_j^-. This field definition is inspired from the work of Danczyk and Suresh [41, 42, 168].

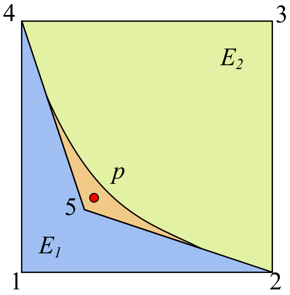


Figure 3.1: 2-element mesh.

For the two-element mesh considered earlier (see Fig. 3.1), any point $\boldsymbol{p}$ within the folded region belongs to three components C_1^+, C_1^-, and C_2^+. Accordingly, the field $\boldsymbol{p}$ is defined using Eq. 3.7 as:

$$u^h(\boldsymbol{p}) = \boldsymbol{N}_1^+(\boldsymbol{p})\hat{\boldsymbol{u}}_1 - \boldsymbol{N}_1^-(\boldsymbol{p})\hat{\boldsymbol{u}}_1 + \boldsymbol{N}_2^+(\boldsymbol{p})\hat{\boldsymbol{u}}_2 \tag{3.8}$$

In the next section, we will show that this condition, together with the compatibility condition, captures constant strain fields.

3.1.3 Theoretical Properties of o-TFEM

Recalling that in classical FEM, the field must satisfy three conditions for convergence [185]:

1. Continuity: The field must be continuous within the element, and across element

boundaries.

2. Rigid body: The element must be strain free under rigid body (constant field) conditions.

3. Constant strain: One must be able to reproduce constant strain conditions exactly.

We show here that these three conditions are precisely met in o-TFEM.

Continuity

To establish continuity, we consider the field at a few points belonging to different regions of the tangled mesh in Fig. 3.2. For points such as $\boldsymbol{p}$ and $\boldsymbol{o}$ (see Fig. 3.2), that are not in the fold, the field can be computed as in classic FEM (see Eq. 3.2):

$$u^h(\boldsymbol{p}) = \boldsymbol{N}_2^+(\boldsymbol{p})\hat{\boldsymbol{u}}_2 \tag{3.9}$$

$$u^h(\boldsymbol{o}) = \boldsymbol{N}_1^+(\boldsymbol{o})\hat{\boldsymbol{u}}_1 \tag{3.10}$$

where $\hat{\boldsymbol{u}}_2$, for example, are the four nodal solutions for element E_2 (after eliminating entries with zero values). From the continuity of the shape functions, it follows that the field is also continuous at these points.

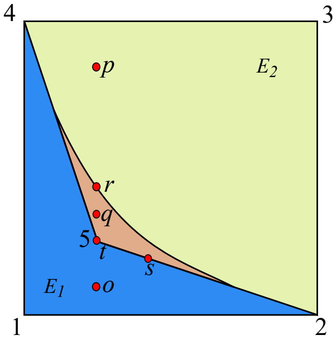


Figure 3.2: Various points within tangled mesh.

On the other hand, for points such as q that belong to multiple components C_1^+, C_1^-

and C_2^+, the field is computed from Eq. 3.7 as follows:

$$u^h(q) = \left(\boldsymbol{N}_1^+(q) - \boldsymbol{N}_1^-(q)\right)\hat{\boldsymbol{u}}_1 + \boldsymbol{N}_2^+(q)\hat{\boldsymbol{u}}_2 \tag{3.11}$$

Once again it is clear that the field is continuous within the fold. Next consider a point r located on the curved edge (corresponding to $|\boldsymbol{J}| = 0$) of the fold. This is a special case of Eq. 3.11 in that:

$$u^h(r) = \lim_{q \to r} \left(\boldsymbol{N}_1^+(q) - \boldsymbol{N}_1^-(q)\right)\hat{\boldsymbol{u}}_1 + \boldsymbol{N}_2^+(q)\hat{\boldsymbol{u}}_2 \tag{3.12}$$

Recall that, since point r lies on the $|\boldsymbol{J}| = 0$ curve in the physical space, it maps to a unique point in the parametric space, i.e., $\boldsymbol{N}_1^+(r) = \boldsymbol{N}_1^-(r)$. Thus:

$$u^h(r) = \boldsymbol{N}_2^+(r)\hat{\boldsymbol{u}}_2 \tag{3.13}$$

The exact same expression is obtained using Eq. 3.9 in the limit as $p \to r$, establishing the continuity of the field across the curved edge of the fold.

Next, consider a point s in Fig. 3.2 on the straight edge of the fold. The field at point s can be obtained by evaluating the expression Eq. 3.11 in the limit as $q \to s$. By the definition of FEM shape functions, $\boldsymbol{N}_1^-(s) = \boldsymbol{N}_2^+(s)$. Therefore the field at point s can be written as:

$$u^h(s) = \boldsymbol{N}_1^+(s)\hat{\boldsymbol{u}}_1 \tag{3.14}$$

Once again, this is the same expression obtained from Eq. 3.10 in the limit as $o \to s$, establishing the continuity of the field across the straight edge of the fold. *Thus the field defined by Eq. 3.7 is continuous within a tangled mesh.*

Rigid body

Now consider a special case where the field is a constant, i.e., $u = c$. To ensure that one can capture such constant fields exactly, we must prove that for any point q located inside the folded region $u^h(q) = c$.

From Eq. 3.11, since $\hat{\boldsymbol{u}}_1 = \{c, c, c, c\}^\top$, $\hat{\boldsymbol{u}}_2 = \{c, c, c, c\}^\top$, $\boldsymbol{N}_1^+(q) = \{N_1^{1+}(q),\ N_1^{2+}(q),\ N_1^{3+}(q),\ N_1^{4+}(q)\}$, etc. we have (after eliminating entries with zero values):

$$u^h(q) = \left(\sum_{i=1}^{4} \left(N_1^{i+}(q) - N_1^{i-}(q) + N_2^{i+}(q)\right)\right) c \tag{3.15}$$

This can be re-grouped as:

$$u^h(q) = \left(\sum_{i=1}^{4} N_1^{i+}(q) - \sum_{i=1}^{4} N_1^{i-}(q) + \sum_{i=1}^{4} N_2^{i+}(q)\right) c$$

By the partition of unity property of shape functions, each summation equals to 1, i.e.,

$$u^h(q) = (1 - 1 + 1)c = c$$

Thus the field defined by Eq. 3.7 can capture a constant field exactly.

Constant strain

Without the loss of generality, consider a constant strain field $u = x$. The x-coordinates of the nodes of element E_1 are denoted by $\hat{\boldsymbol{x}}_1 = \left\{x^1, x^2, x^5, x^4\right\}^\top$ and that for element E_2 by $\hat{\boldsymbol{x}}_2 = \left\{x^2, x^3, x^4, x^5\right\}^\top$. Note that here, the node numbers are denoted using superscripts while element numbers are denoted using subscripts. Assume that field at all vertices except the concave vertex is exactly equal to the x-coordinate at that point. From the compatibility condition (Eq. 2.9), it immediately follows that $u^5 = x^5$. Hence, we have $\hat{\boldsymbol{u}}_1 = \hat{\boldsymbol{x}}_1$ and $\hat{\boldsymbol{u}}_2 = \hat{\boldsymbol{x}}_2$.

Further, consider any point q inside the fold. From Eq. 3.11, we have

$$u^h(q) = \left(\boldsymbol{N}_1^+(q) - \boldsymbol{N}_1^-(q)\right) \hat{\boldsymbol{x}}_1 + \boldsymbol{N}_2^+(q)\hat{\boldsymbol{x}}_2 \tag{3.16}$$

From the isoparametric mapping of x (see Eq. 2.1), it follows that

$$u^h(q) = \boldsymbol{N}_1^+(q)\hat{\boldsymbol{x}}_1 - \boldsymbol{N}_1^-(q)\hat{\boldsymbol{x}}_1 + \boldsymbol{N}_2^+(q)\hat{\boldsymbol{x}}_2 = x^q - x^q + x^q \tag{3.17}$$

Thus, the field at any point q is the fold is $u^h(q) = x^q$. *Hence, the field defined by Eq. 3.7 along with compatibility condition (Eq. 2.9) can exactly capture a constant strain field.*

3.1.4 o-TFEM Assembly

Having established the theoretical properties of o-TFEM, we will now proceed to derive the underlying linear system of equations. In particular, to derive the stiffness matrix, we substitute the unambiguous definition of the field defined per Eq. 3.7 in the Galerkin formulation of Eq. 3.1 leading to:

$$\boldsymbol{K}^{\theta} = \int\limits_{\Omega} \left[\nabla \sum_{j} \left(\boldsymbol{N}_j^{+} - \boldsymbol{N}_j^{-}\right)\right]^{\top} \boldsymbol{D} \left[\nabla \sum_{k} \left(\boldsymbol{N}_k^{+} - \boldsymbol{N}_k^{-}\right)\right] d\Omega \tag{3.18}$$

Regrouping:

$$\begin{aligned} \boldsymbol{K}^{\theta} = \int\limits_{\Omega} \Bigg\{ & \left(\nabla \sum_{j} \left(\boldsymbol{N}_j^{+} - \boldsymbol{N}_j^{-}\right)\right)^{\top} \boldsymbol{D} \left(\nabla \sum_{j} \left(\boldsymbol{N}_j^{+} - \boldsymbol{N}_j^{-}\right)\right) \\ & + \left(\nabla \sum_{j} \left(\boldsymbol{N}_j^{+} - \boldsymbol{N}_j^{-}\right)\right)^{\top} \boldsymbol{D} \left(\nabla \sum_{k \neq j} \left(\boldsymbol{N}_k^{+} - \boldsymbol{N}_k^{-}\right)\right) \Bigg\} \, d\Omega \end{aligned} \tag{3.19}$$

Further, the first term of Eq. 3.19 can be expanded as

$$\begin{aligned} & \sum_{j} \int\limits_{C_j^{+}} \nabla \boldsymbol{N}_j^{+\top} \boldsymbol{D} \nabla \boldsymbol{N}_j^{+} d\Omega + \sum_{j} \int\limits_{C_j^{-}} \nabla \boldsymbol{N}_j^{-\top} \boldsymbol{D} \nabla \boldsymbol{N}_j^{-} d\Omega - \\ & \sum_{j} \int\limits_{C_j^{+} \cap C_j^{-}} \left(\nabla \boldsymbol{N}_j^{+\top} \boldsymbol{D} \nabla \boldsymbol{N}_j^{-} + \nabla \boldsymbol{N}_j^{-\top} \boldsymbol{D} \nabla \boldsymbol{N}_j^{+}\right) d\Omega \end{aligned} \tag{3.20}$$

Observe that the first term of Eq. 3.20 involves integration over C_j^{+} with integrand containing $\nabla \boldsymbol{N}_j^{+}$ terms. Similarly, the second term of Eq. 3.20 involves integration over C_j^{-} with integrand containing $\nabla \boldsymbol{N}_j^{-}$ terms. These two terms together form the standard stiffness matrix $\boldsymbol{K}^0$ as discussed later in Section 3.1.5. On the other hand, the third term of Eq. 3.20 involves integration over $C_j^{+} \cap C_j^{-}$, i.e., over the fold, and the integrand involves both the terms $\nabla \boldsymbol{N}_j^{+}$ and $\nabla \boldsymbol{N}_j^{-}$. It captures the coupling between the components of the same element. We refer to the third term as $\boldsymbol{K}^S$, where the superscript S alludes to self-intersection.

Next, the second term in Eq. 3.19 captures the coupling between components of different elements, and can be expanded as

$$\boldsymbol{K}^N = \sum_j \sum_{k \neq j} \left(\int\limits_{C_j^+ \cap C_k^+} \nabla \boldsymbol{N}_j^{+\top} \boldsymbol{D} \nabla \boldsymbol{N}_k^+ d\Omega - \int\limits_{C_j^- \cap C_k^+} \nabla \boldsymbol{N}_j^{-\top} \boldsymbol{D} \nabla \boldsymbol{N}_k^+ d\Omega - \right.$$
$$\left. \int\limits_{C_j^+ \cap C_k^-} \nabla \boldsymbol{N}_j^{+\top} \boldsymbol{D} \nabla \boldsymbol{N}_k^- d\Omega + \int\limits_{C_j^- \cap C_k^-} \nabla \boldsymbol{N}_j^{-\top} \boldsymbol{D} \nabla \boldsymbol{N}_k^- d\Omega \right)$$

where N alludes to intersection between neighbors. Recall that $C_k^- = \varnothing$ for convex elements. In such cases, the terms involving C_k^- vanish.

In summary, Eq. 3.18 can be written as:

$$\boldsymbol{K}^\theta = \boldsymbol{K}^0 + \boldsymbol{K}^S + \boldsymbol{K}^N \tag{3.21}$$

where,

$$\boldsymbol{K}^0 = \prod_{Assemble} \left(\int\limits_{C_j^+} \nabla \boldsymbol{N}_j^{+\top} \boldsymbol{D} \nabla \boldsymbol{N}_j^+ d\Omega + \int\limits_{C_j^-} \nabla \boldsymbol{N}_j^{-\top} \boldsymbol{D} \nabla \boldsymbol{N}_j^- d\Omega \right) \tag{3.22}$$

$$\boldsymbol{K}^S = \prod_{Assemble} \left(- \int\limits_{C_j^+ \cap C_j^-} \nabla \boldsymbol{N}_j^{+\top} \boldsymbol{D} \nabla \boldsymbol{N}_j^- d\Omega - \int\limits_{C_j^+ \cap C_j^-} \nabla \boldsymbol{N}_j^{-\top} \boldsymbol{D} \nabla \boldsymbol{N}_j^+ d\Omega \right) \tag{3.23}$$

$$\begin{aligned} \boldsymbol{K}^N = \prod_{Assemble} \Bigg(& \int\limits_{C_j^+ \cap C_k^+} \nabla \boldsymbol{N}_j^{+\top} \boldsymbol{D} \nabla \boldsymbol{N}_k^+ d\Omega - \int\limits_{C_j^- \cap C_k^+} \nabla \boldsymbol{N}_j^{-\top} \boldsymbol{D} \nabla \boldsymbol{N}_k^+ d\Omega \\ & - \int\limits_{C_j^+ \cap C_k^-} \nabla \boldsymbol{N}_j^{+\top} \boldsymbol{D} \nabla \boldsymbol{N}_k^- d\Omega + \int\limits_{C_j^- \cap C_k^-} \nabla \boldsymbol{N}_j^{-\top} \boldsymbol{D} \nabla \boldsymbol{N}_k^- d\Omega \Bigg) \end{aligned} \tag{3.24}$$

Similarly, substituting Eq. 3.7 on the right-hand side of the Galerkin formulation of Eq. 3.1, we get

$$\boldsymbol{f}^{\theta} = \int\limits_{\Omega} \left(\sum_{j} \left(\boldsymbol{N}_j^{+} - \boldsymbol{N}_j^{-} \right) \right)^{\top} f d\Omega = \prod_{Assemble} \int\limits_{C_j^{+}} \boldsymbol{N}_j^{+\top} f d\Omega - \int\limits_{C_j^{-}} \boldsymbol{N}_j^{-\top} f d\Omega \tag{3.25}$$

Finally, the compatibility condition defined in Eq. 2.9 when imposed at the re-entrant vertices of all the concave elements, will lead to a set constraint equations:

$$\boldsymbol{C}^{\top} \hat{\boldsymbol{u}} = \boldsymbol{0} \tag{3.26}$$

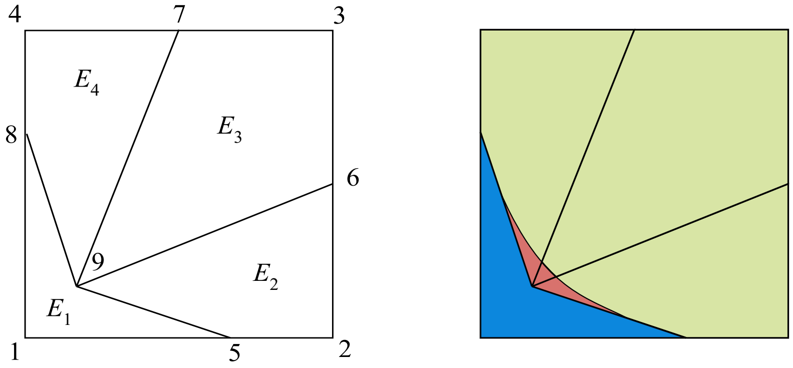
where the number of rows in $\boldsymbol{C}^{\top}$ is equal to the number of concave vertices. In summary, in o-TFEM we must solve the following linear system of equations:

$$\begin{bmatrix} \boldsymbol{K}^{\theta} & \boldsymbol{C} \\ \boldsymbol{C}^{\top} & \boldsymbol{0} \end{bmatrix} \begin{Bmatrix} \hat{\boldsymbol{u}} \\ \boldsymbol{\mu} \end{Bmatrix} = \begin{Bmatrix} \boldsymbol{f}^{\theta} \\ \boldsymbol{0} \end{Bmatrix}. \tag{3.27}$$

Note that the matrix on the left-hand side of Eq. 3.27 is not necessarily positive definite. Further, after solving Eq. 3.27, we obtain the unknown degrees of freedom $\hat{\boldsymbol{u}}$, along with the Lagrange multipliers $\boldsymbol{\mu}$. The Lagrange multipliers arise due to the finite set of constraint equations, and have no role in the field interpolation. Consequently, they are not used further in this work.

3.1.5 Implementation

In the previous section, we considered a two-element mesh patch to explain the theory behind o-TFEM. In this section, we discuss the implementation of o-TFEM, using a more complex four-element patch in Fig. 3.3, where the fold overlaps with multiple neighboring elements.



(a) (b)

Figure 3.3: A 4-element tangled mesh with element E_1 being concave.

Geometric Processing

The first step in the implementation of o-TFEM is the detection of concave elements. This amounts to computing and checking the four interior angles of every quad element. Thus, in Fig. 3.3, element E_1 is concave. Given a concave element E_j, the next step is to decompose its parametric space into two parts, corresponding to the two components C_j^+ and C_j^-; see Eq. 2.4. This decomposition can be carried out easily as follows. Let the element be defined by vertices $[(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4)]$. Given the bilinear mapping, the Jacobian is given by

$$\boldsymbol{J} = \begin{bmatrix} x_{,\xi} & y_{,\xi} \\ x_{,\eta} & y_{,\eta} \end{bmatrix} = \frac{1}{4} \begin{bmatrix} \eta - 1 & 1 - \eta & 1 + \eta & -(1 + \eta) \\ \xi - 1 & -(1 + \xi) & 1 + \xi & 1 - \xi \end{bmatrix} \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \\ y_1 & y_2 & y_3 & y_4 \end{bmatrix}^T$$

The determinant simplifies to:

$$|\boldsymbol{J}| = \det(\boldsymbol{J}) = c_0 + c_1\xi + c_2\eta \tag{3.28}$$

where,

$$c_0 = [(x_1 - x_3)(y_2 - y_4) - (x_2 - x_4)(y_1 - y_3)]/8$$

$$c_1 = [(x_3 - x_4)(y_1 - y_2) - (x_1 - x_2)(y_3 - y_4)]/8$$

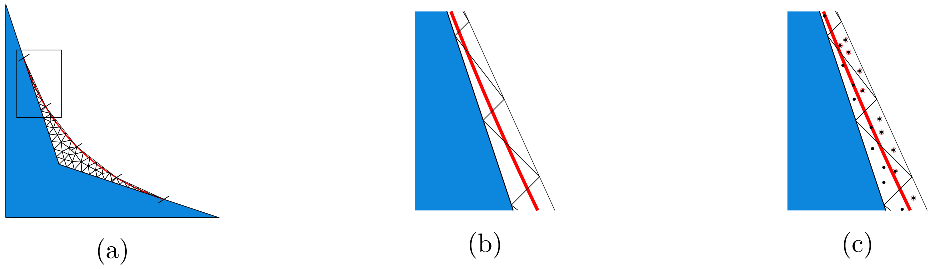
$$c_2 = [(x_2 - x_3)(y_1 - y_4) - (x_1 - x_4)(y_2 - y_3)]/8$$

We conclude that $|\boldsymbol{J}| = 0$ is a straight line in the parametric space, defined by the constants c_0, c_1 and c_2. Thus, the parametric space is decomposed into two parts on opposite sides of the $|\boldsymbol{J}| = 0$ line.

Next, we consider all elements E_k that share the re-entrant vertex with E_j. For example, in Fig. 3.3 three elements E_2, E_3 and E_4 share the re-entrant vertex 9 with element E_1. Therefore, elements E_2, E_3 and E_4 intersect with different parts of the fold. Consider, for example, the intersection of element E_4 with the fold. Observe that boundary of the intersection is made up of three edges: two of them are straight lines in the physical space, while the third is a curve obtained by mapping the $|\boldsymbol{J}| = 0$ line. Thus for every neighboring element E_k intersecting the fold with E_j, we determine the boundary of the intersection region. This intersection region must be triangulated for numerical integration.

Triangulating the folded region

To triangulate the folded region, the $|\boldsymbol{J}| = 0$ curve has to be approximated by a finite number of line segments. A sufficient number of segments is needed to ensure that all quadrature points lie within the fold. To illustrate, let the $|\boldsymbol{J}| = 0$ curve (highlighted) be approximated by four line segments, and the polygonal fold triangulated, as in Fig. 3.4a. Observe that the integration points for some of the triangles lie outside the folded region, as seen in Fig. 3.4c. This will lead to negative Jacobians and potentially, singularities, and therefore erroneous results.


(a)

(b)

(c)

Figure 3.4: Approximation of $|\boldsymbol{J}| = 0$ curve using four segments: (a) Triangulation of the folded region, (b) Zoomed-in view of region of interest (c) Integration points marked which lie outside the fold are marked with a circle.

To avoid these issues, a sufficient number of segments must be used to ensure that all the integration points lie within the folded region. We adopt a simple iterative scheme by doubling the number of segments until all the integration points lie within the folded region. For example, with 128 segments, it is easy to confirm for this case that all the integration points lie within the folded region. One may also employ an adaptive procedure to sub-divide the segments. Note that the triangulation is used for numerical integration, and does not contribute to the degrees of freedom in o-TFEM.

Computing $\boldsymbol{K}^S$

Now consider computing $\boldsymbol{K}^S$ matrix in Eq. 3.23. $\boldsymbol{K}^S$ is comprised of contributions from all concave elements within a mesh. A typical contribution $\boldsymbol{k}_j^S$ from a concave element E_j is:

$$\boldsymbol{k}_j^S = -\int_{C_j^+ \cap C_j^-} \left(\nabla \boldsymbol{N}_j^{+\top} \boldsymbol{D} \nabla \boldsymbol{N}_j^- + \nabla \boldsymbol{N}_j^{-\top} \boldsymbol{D} \nabla \boldsymbol{N}_j^+ \right) d\Omega$$

Given the triangulation discussed previously, we have:

$$\boldsymbol{k}_j^S = \sum_{triangles} - \int_{triangle} \left(\nabla \boldsymbol{N}_j^{+\top} \boldsymbol{D} \nabla \boldsymbol{N}_j^- + \nabla \boldsymbol{N}_j^{-\top} \boldsymbol{D} \nabla \boldsymbol{N}_j^+ \right) dxdy$$

The sub-matrices $\boldsymbol{k}_j^S$ from each concave element are then assembled to form $\boldsymbol{K}^S$.

Computing $\boldsymbol{K}^N$

Next, consider computing $\boldsymbol{K}^N$ matrix in Eq. 3.24 where the superscript N refers to overlap between neighboring elements. $\boldsymbol{K}^N$ is comprised of contributions from pairs of neighboring elements within a mesh, where one or both of the elements are concave. A typical submatrix $\boldsymbol{k}^N_{jk}$ due to an overlap of neighboring elements E_j and E_k is given by:

$$\begin{aligned}\boldsymbol{k}^N_{jk} = \int\limits_{C^+_j \cap C^+_k} \nabla {\boldsymbol{N}^+_j}^\top \boldsymbol{D} \nabla \boldsymbol{N}^+_k d\Omega - \int\limits_{C^-_j \cap C^+_k} \nabla {\boldsymbol{N}^-_j}^\top \boldsymbol{D} \nabla \boldsymbol{N}^+_k d\Omega \\ - \int\limits_{C^+_j \cap C^-_k} \nabla {\boldsymbol{N}^+_j}^\top \boldsymbol{D} \nabla \boldsymbol{N}^-_k d\Omega + \int\limits_{C^-_j \cap C^-_k} \nabla {\boldsymbol{N}^-_j}^\top \boldsymbol{D} \nabla \boldsymbol{N}^-_k d\Omega\end{aligned}$$

Observe that not all terms need to be considered depending on the concavity of the two elements. For example, if element E_k is convex, then the third and fourth term can be disregarded since C^-_k does not exist.

Computing $\boldsymbol{K}^0$

We will now consider computing $\boldsymbol{K}^0$ in Eq. 3.22, where $\boldsymbol{K}^0$ is comprised of contributions from all elements within a mesh. A typical contribution $\boldsymbol{k}^0_j$ from element E_j is given by:

$$\boldsymbol{k}^0_j = \int\limits_{C^-_j} \nabla {\boldsymbol{N}^-_j}^\top \boldsymbol{D} \nabla \boldsymbol{N}^-_j d\Omega + \int\limits_{C^+_j} \nabla {\boldsymbol{N}^+_j}^\top \boldsymbol{D} \nabla \boldsymbol{N}^+_j d\Omega$$

If $C^-_j = \varnothing$, this reduces to standard numerical integration over the parametric space. On the other hand, when $C^-_j \neq \varnothing$, we must exploit the triangulation of the fold C^-_j and C^+_j as discussed earlier, and illustrated again in Fig. 3.5a.

The triangulation for C^+_j consists of two parts: the triangulation over the fold, and the triangulation over the non-folded region as in Fig. 3.5b. Note that these two triangulations need not be conforming since they are used primarily for integration.

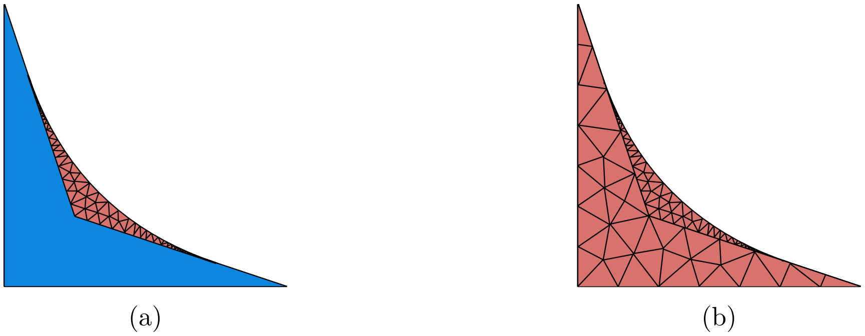


Figure 3.5: Triangulation of the (a) C_j^- region and (b) C_j^+ region.

Computing $\boldsymbol{f}^\theta$

Finally, the forcing term $\boldsymbol{f}^\theta$ is computed similar to classical FEM with two modifications – integration takes place over each component, accounting for the orientation of the component. The latter is accomplished by subtracting the contribution from the negatively oriented component C_j^- before assembling.

$$\boldsymbol{f}^\theta = \prod_{Assemble} \int_{C_j^+} \boldsymbol{N}_j^{+\top} f d\Omega - \int_{C_j^-} \boldsymbol{N}_j^{-\top} f d\Omega$$

Similar to $\boldsymbol{K}^0$ computation, the integration over C_j^+ and C_j^- regions can be performed by employing the triangulation in Fig. 3.5.

3.2 Numerical Experiments

In this section, we demonstrate o-TFEM by solving the Poisson problem and static linear elasticity (plane stress) problem over various tangled meshes. Numerical experiments are conducted under the following conditions:

- The implementation is in MATLAB R2020b, 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 16.

- Triangulation of the folded region is performed by employing MATLAB's inbuilt mesher - `generateMesh`. The number of quadrature points for triangles is 4.

The questions being investigated through the experiments are:

- **Accuracy**: How does the accuracy of o-TFEM compare against that of the FEM? Does the accuracy of o-TFEM and FEM depend on the extent of tangling? To measure accuracy, we consider the L^2 and/or energy errors; the L^2 error is defined as:

  $$||u - u^h||_{L^2(\Omega)} = \sqrt{\int_\Omega (u - u^h)^2 \, d\Omega} \tag{3.29}$$

  and the energy norm is defined as:

  $$||e_h||_{E(\Omega)} = ||u - u^h||_{E(\Omega)} = \sqrt{\int_\Omega (\nabla u - \nabla u^h)^\intercal (\nabla u - \nabla u^h) \, d\Omega} \tag{3.30}$$

  where u and u^h are the exact and computed solutions respectively.

- **Condition Number**: The condition number is a measure of how close a matrix is to being singular [66]; it is desirable to have a condition number close to unity. We employ MATLAB's built-in function `condest` to compute the condition number.

- **Convergence**: Here we investigate if o-TFEM converges as the element size is decreased.

- **Computational Cost**: Finally, we investigate the computational overhead of o-TFEM for a sample problem.

3.2.1 Patch Test: Two Element Mesh

In the first experiment, we consider a square domain $\Omega \in (0,1)^2$. It is divided into two quads, one of which is concave as in Fig. 3.6. In this case, the folded region is shared by a single neighboring element. To vary the extent of tangling, the position of vertex 5 is defined as in Fig. 3.6, where $d \in (0, 0.5)$. Larger the value of d, greater is the extent of tangling.

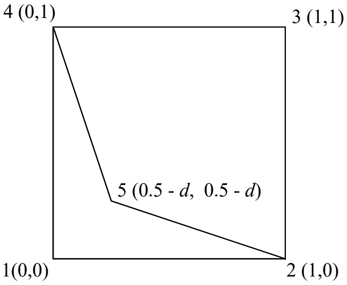


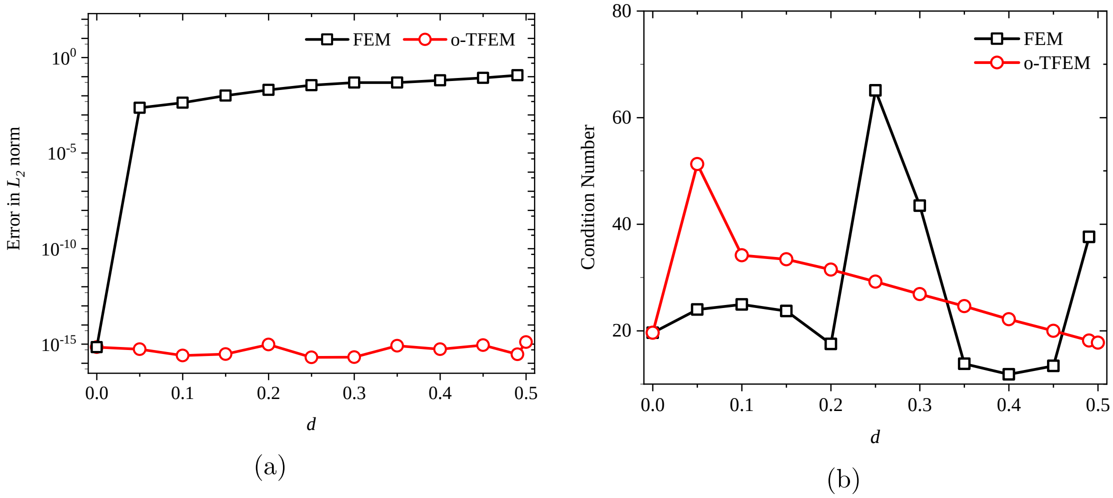
Figure 3.6: Two element tangled mesh.

Laplace problem

First, the Laplace problem is solved over the tangled mesh. The field $u = x$ is considered as the exact solution; the corresponding Dirichlet boundary condition is applied on the left edge, and the Neumann conditions on the remaining three edges.

Fig. 3.7 illustrates the L^2 errors in FEM and o-TFEM, for varying extent of tangling. As one can observe, o-TFEM leads to machine precision accuracy for all the values of d, while FEM is erroneous.

For $d = 0.5$, the quadrilateral element becomes degenerate (zero-area). Even in this case, o-TFEM provides accurate results while FEM gives invalid results. Finally, for $d = 0$, both elements are convex; o-TFEM reduces to FEM, and machine precision accuracy is obtained.



(a)

(b)

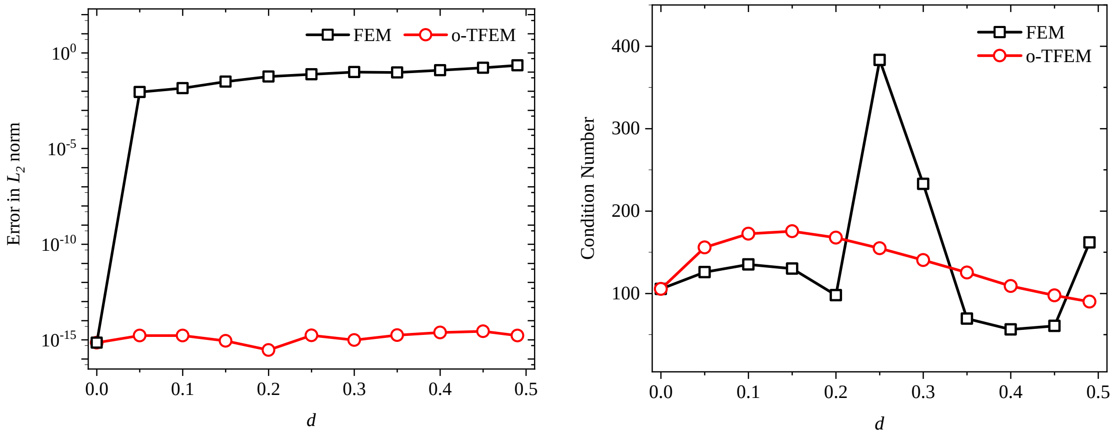
Figure 3.7: Comparison of o-TFEM and FEM for two-element mesh for Poisson problem: (a) L^2 error vs. d, (b) condition number vs. d.

Plane stress elasticity

As a second test, plane stress elastostatics problem is solved with Young's modulus $E = 1$, Poisson's ratio $\nu = 0.3$. Consider two random linear displacement fields:

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

The boundary conditions corresponding to the above fields were applied on the square geometry as in the previous example, i.e., Dirichlet boundary conditions on the left edge, and Neumann (traction) boundary conditions on the remaining edges. Fig. 3.8a compares L^2 error norm for o-TFEM and FEM. As with the previous example, o-TFEM achieves machine precision accuracy while classic FEM fails for the tangled mesh.



(a)

(b)

Figure 3.8: Comparison of o-TFEM and FEM for two-element mesh for plane stress problem: (a) L^2 error vs. d, (b) condition number vs. d.

3.2.2 Patch Test: Four Element Mesh

Next, we consider a square domain $\Omega \in (0,1)^2$ which is discretized into 4 quadrilateral elements, one of which is concave as in Fig. 3.9. 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 $r = 0.125\sqrt{2}$.

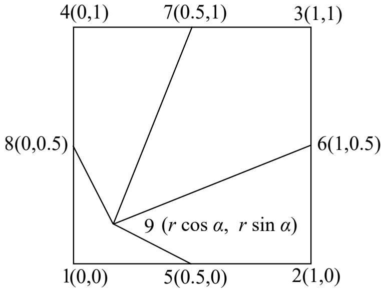
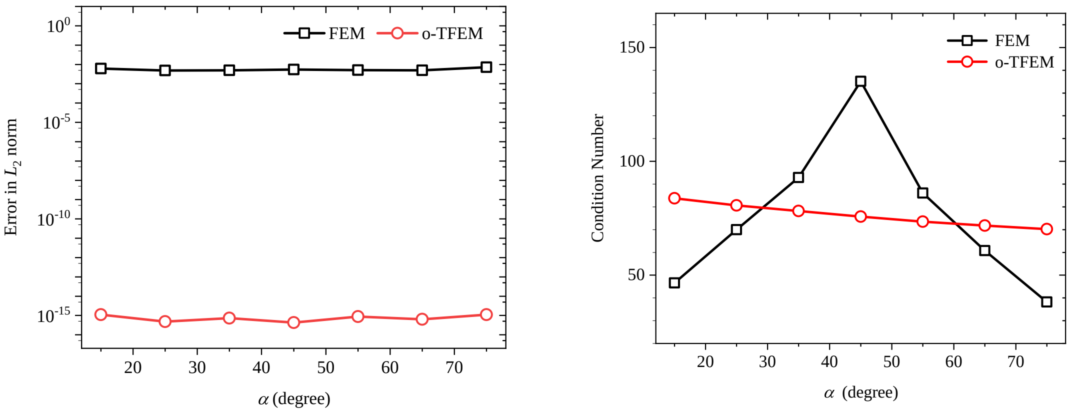


Figure 3.9: Tangled mesh with four elements.

Laplace problem

To validate o-TFEM, the Laplace problem is solved over the tangled mesh. Consider a linear field $u = 0.579x + 0.246y - 0.374$ as the exact solution. The corresponding Dirichlet condition is imposed on the left edge, while the Neumann conditions are imposed on the remaining three edges. Fig. 3.10a illustrates the L^2 errors in FEM and o-TFEM, while Fig. 3.10b compares the condition numbers.



(a)

(b)

Figure 3.10: Comparison of o-TFEM and FEM for four-element mesh for Laplace equation: (a) Error vs. α (b) Condition number vs. α

Plane stress elasticity

Next, the plane stress elastostatics problem is solved with random displacement fields, elastic constants, and boundary conditions as in the two-element example considered earlier. Fig. 3.11 compares L^2 error norm and condition number of o-TFEM and FEM. As with the previous example, o-TFEM achieves machine precision accuracy while classic FEM fails for the tangled mesh.

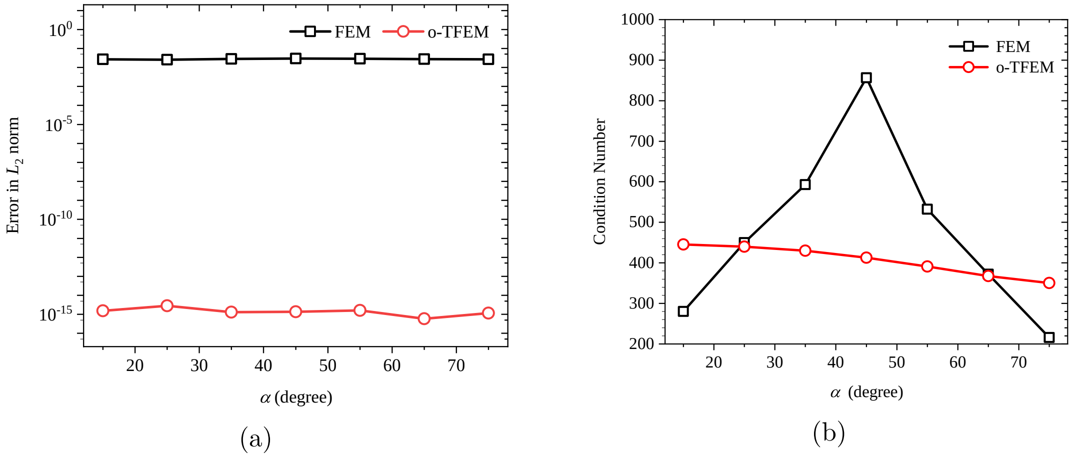


Figure 3.11: Comparison of o-TFEM and FEM for four-element mesh for the plane stress problem: (a) Error vs. α (b) Condition number vs. α

3.2.3 Convergence Test

While the previous experiments confirmed that o-TFEM satisfies various patch tests, we now investigate the convergence of o-TFEM when the exact solution lies outside the span of the finite element space. We consider an example from [158] where the Poisson problem is considered with boundary conditions and a heat source are such that the exact solution is $u = (1 - x^2)(1 - y^2)$. The domain is a unit square with Dirichlet boundary conditions. Various meshes are constructed by using the two-element mesh (see Fig. 3.6) with $d = 0.4$ as the basic repeating unit. Fig. 3.12 illustrates a few sample meshes used for the convergence study; the basic units are scaled to conform to the unit square domain.

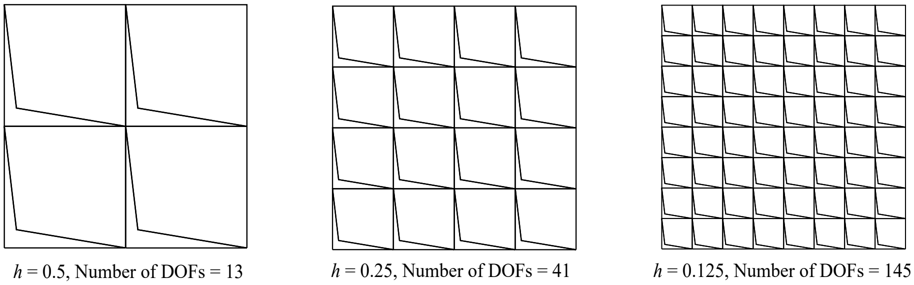
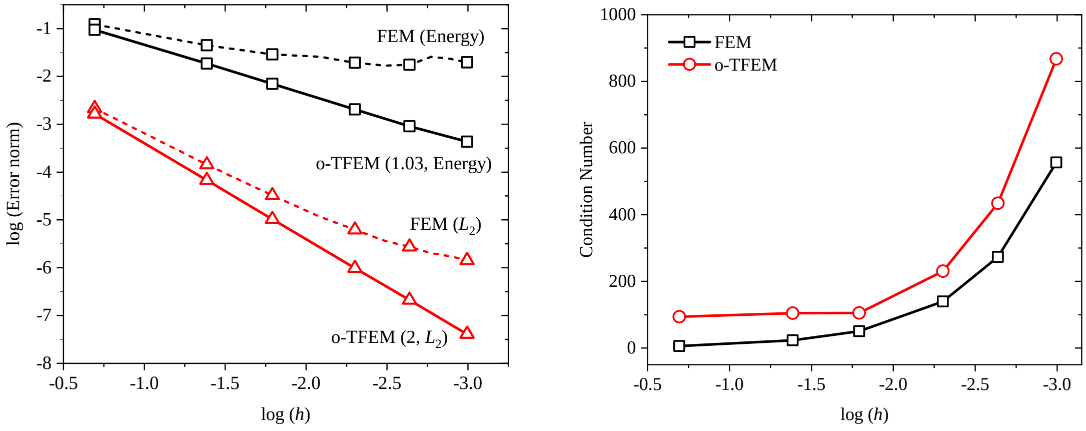

$h = 0.5$, Number of DOFs = 13
$h = 0.25$, Number of DOFs = 41
$h = 0.125$, Number of DOFs = 145

Figure 3.12: Meshes used for convergence study.

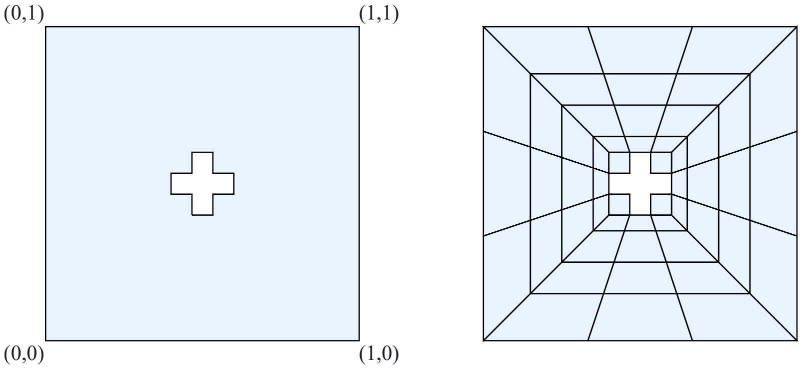
The L^2 and energy norm errors for FEM and o-TFEM, as a function of mesh size (h), are illustrated in Fig. 3.13a. The convergence rates for the o-TFEM L^2 and energy norm errors are 2 and 1 respectively which is consistent with expectations for second-order elliptic PDEs with linearly complete approximations [158]. Note that the results may differ as the quality of quadrilaterals changes. Fig. 3.13b shows an increase in condition number for both FEM and o-TFEM as the mesh size is reduced; this is to be expected.



(a) (b)

Figure 3.13: L^2 and energy norms errors as a function of mesh size h for the meshes in Fig. 3.12. (b) Condition number vs. h

3.2.4 Application: Mesh Morphing



(a) (b)

Figure 3.14: Mesh morphing application: (a) Domain with a void, and (b) initial quad mesh.

To illustrate potential applications, we consider mesh morphing with tangling. Mesh morphing is essentially a mesh update where the topology is maintained while the mesh nodes are re-located according to a specified rule [154]. It enables rapid simulation of geometric configurations. As an example, consider a domain with a 'plus' shaped void at the center as in Fig. 3.14a, with the initial mesh as in Fig. 3.14b.

The inner void is now rotated in a counterclockwise direction by an angle β, and the mesh is morphed as follows. Nodes except the outer boundary nodes are rotated about the center such that the angle of rotation exponentially increases from $0°$ to β as nodes get closer to the void. Fig. 3.15 illustrates the morphed mesh for $\beta = 70°$. Observe that some of the quads (shown in red) are concave, i.e., the mesh is tangled. In our experiment, the angle β was varied from $0°$ to $89°$. If β is increased further ($90° \leq \beta \leq 97°$), some of the concave elements become too thin, significantly increasing the computational cost. For $\beta > 97°$, the elements get twisted, and such twisted elements are not considered here.

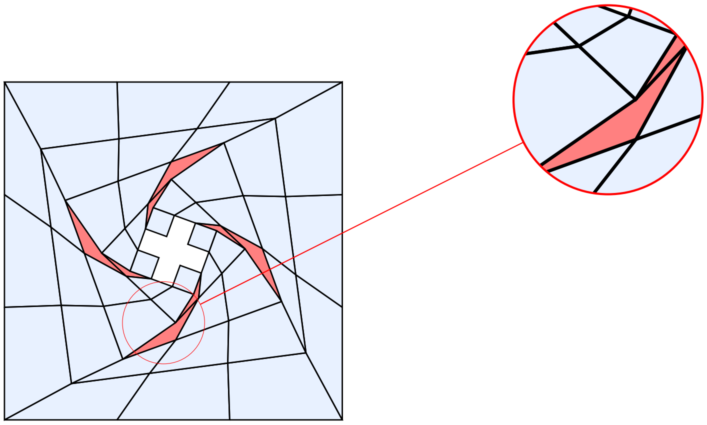
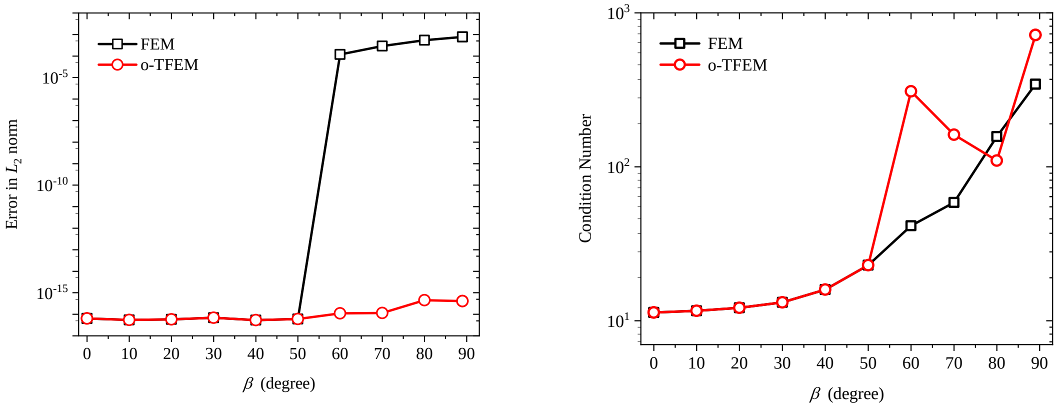


Figure 3.15: A tangled morphed mesh.

We carry out two experiments, both solving the Poisson problem, for this example. First, we apply Dirichlet boundary conditions corresponding to $u(x,y) = 0.579x + 0.246y - 0.374$ over the entire boundary, with $f(x,y) = 0$. The L^2 errors for various values of β are illustrated in Fig 3.16a. For $\beta < 60°$, i.e., when there is no tangling, both FEM and o-TFEM are able to capture the field exactly. However, for $\beta > 50°$, FEM error grows rapidly, while o-TFEM error remains close to machine precision. The two condition numbers are illustrated (see Fig. 3.16b).

Next, we set $f(x,y) = 1$ and zero Dirichlet condition is applied on the entire boundary.

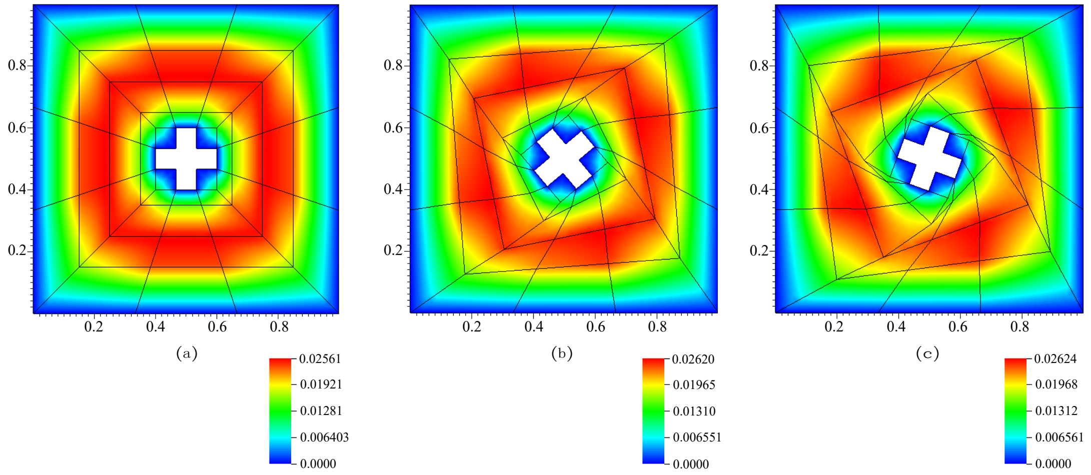


(a)

(b)

Figure 3.16: Comparison of o-TFEM and FEM for four-element mesh: (a) L^2 error vs. β, (b) condition number vs. β.

Fig. 3.17 illustrates the post-processed results for $\beta = 0°$, $40°$, and $70°$.



(a)

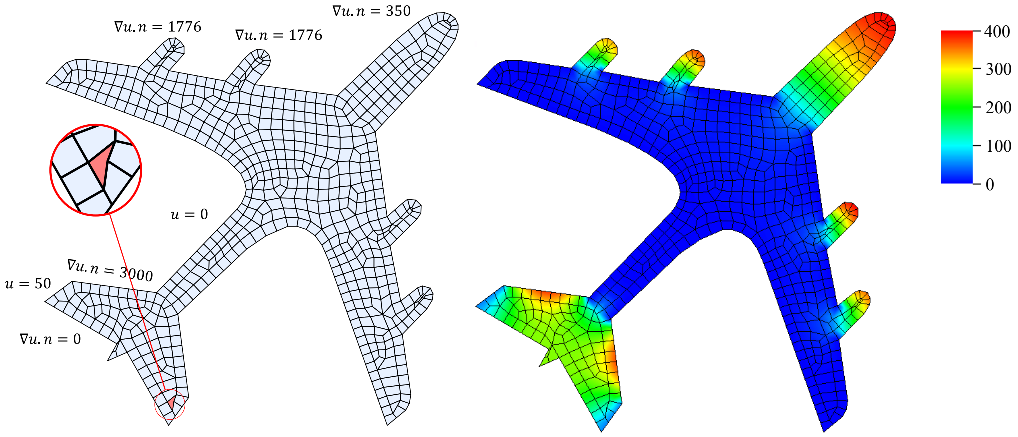
(b)

(c)

Figure 3.17: Post-processed solution using o-TFEM for β = (a) 0°, (b) 40°, and (c) 70°.

3.2.5 Real-world Tangled Mesh: Aircraft Model

We now consider an example where tangling occurs in practice. The mesh in Fig. 3.18a was generated using the quad mesher proposed in [146]; it can be observed that one quad element (among a total of around 600 elements) is concave, and the folded region is shared by two neighboring elements. Here we solve a Poisson problem with $f(x, y) = 10$, with the boundary conditions in Fig. 3.18a. Since this problem has no analytical solution, it was first solved using a very fine quadrilateral mesh with over 14000 elements, and the resulting field was used as the reference solution. The problem was then solved over the tangled quad mesh using FEM and o-TFEM. At the re-entrant vertex, the relative error in the o-TFEM solution was 0.55% while the FEM error was 4.24%. The o-TFEM post-processed solution is illustrated in Fig. 3.18b.



(a)

(b)

Figure 3.18: (a) Mesh for an aircraft model, with one concave element.(b) o-TFEM solution to a Poisson problem.

FEM took 0.37 seconds to solve the problem (albeit incorrectly), while o-TFEM requires around 1.4 seconds (post-processing was not included in either case). The o-TFEM time can potentially be reduced by: (1) by developing more efficient integration techniques, (2) reducing the number of triangles in the non-fold region, and (3) parallelization.

3.3 Summary

In this chapter, we established the central tenets of o-TFEM: field unambiguity condition and compatibility constraint. We demonstrated that the o-TFEM field satisfies three conditions required for convergence: (a) continuity within and across the element boundary, as well as, the field exactly captures (b) rigid body and (c) constant strain conditions. TFEM can be considered as an extension of FEM since it reduces to the standard FEM for untangled meshes.

We demonstrated that o-TFEM passes the numerical patch test and provides an optimal convergence rate. Applications of o-TFEM in mesh morphing and a tangled aircraft model have been explored.

3.3.1 Limitations

The major drawback of the o-TFEM formulation is that one must integrate over the fold to compute the correction terms. For example, in Eq. 3.21, $\boldsymbol{K}^S$ term is given by

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

This entails integration over $C_1^+ \cap C_1^-$, i.e., over the tangled region. This makes o-TFEM cumbersome to implement and computationally expensive. Moreover, care must be taken to avoid integration points very close to the $|\boldsymbol{J}| = 0$ curve [129].

Chapter 4

Isoparametric Tangled Finite Element Method

4.1 Isoparametric TFEM (i-TFEM) Formulation

In this chapter, an isoparametric TFEM (i-TFEM) formulation is proposed that eliminates some of the drawbacks of the oriented-TFEM discussed in the previous chapter.

4.1.1 Field Definition in i-TFEM

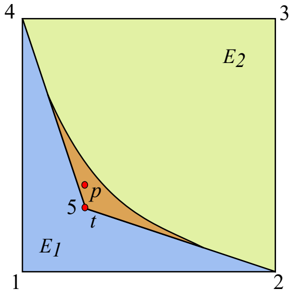


Figure 4.1: Two-element tangled mesh.

For the two-element mesh in Fig. 4.1, 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 $\boldsymbol{p}$ 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.,

$$\boldsymbol{u}^h(\boldsymbol{p}) = \boldsymbol{N}_1^+(\boldsymbol{p})\hat{\boldsymbol{u}}_1 = \boldsymbol{N}_1^-(\boldsymbol{p})\hat{\boldsymbol{u}}_1. \tag{4.1}$$

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.

As $\boldsymbol{p} \to \boldsymbol{t}$ (see Fig. 4.1), we have $\boldsymbol{N}_1^-(t)\hat{\boldsymbol{u}}_1 = \boldsymbol{u}^5$; therefore, Eq. 4.1 can be rewritten in the form:

$$\boldsymbol{u}^h(\boldsymbol{p}) = N_1^{1+}(\boldsymbol{p})\boldsymbol{u}^1 + N_1^{2+}(\boldsymbol{p})\boldsymbol{u}^2 + N_1^{4+}(\boldsymbol{p})\boldsymbol{u}^4 + N_1^{3+}(\boldsymbol{p})\boldsymbol{u}^5 = \boldsymbol{u}^5. \tag{4.2}$$

Note that superscript and subscript indicate the node number and element number respectively. Observe that Eq. 4.2 is exactly the compatibility condition (Eq. 2.10). Thus, Eq. 4.1 is a generalization of the compatibility condition.

Now recall that the field at $\boldsymbol{p}$ within a fold is defined in o-TFEM to be (see Eq. 3.8)

$$\boldsymbol{u}^h(\boldsymbol{p}) = \boldsymbol{N}_1^+(\boldsymbol{p})\hat{\boldsymbol{u}}_1 - \boldsymbol{N}_1^-(\boldsymbol{p})\hat{\boldsymbol{u}}_1 + \boldsymbol{N}_2^+(\boldsymbol{p})\hat{\boldsymbol{u}}_2. \tag{4.3}$$

Simplifying, we have:

$$\boldsymbol{u}^h(\boldsymbol{p}) = \boldsymbol{N}_2^+(\boldsymbol{p})\hat{\boldsymbol{u}}_2. \tag{4.4}$$

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. 4.2. 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):

$$\boldsymbol{k}_2^0 = \int\limits_{E_2} \left(\nabla \boldsymbol{N}_2{}^{\top} \boldsymbol{D} \nabla \boldsymbol{N}_2\right) d\Omega. \tag{4.5}$$

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:

$$\boldsymbol{u}^h(\boldsymbol{p}) = \boldsymbol{N}_1^+(\boldsymbol{p})\hat{\boldsymbol{u}}_1. \tag{4.6}$$

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

$$\widehat{\boldsymbol{k}}_1 = \int_{\widehat{E}_1} \left(\nabla \boldsymbol{N}_1{}^{\intercal} \boldsymbol{D} \nabla \boldsymbol{N}_1\right) d\Omega. \tag{4.7}$$

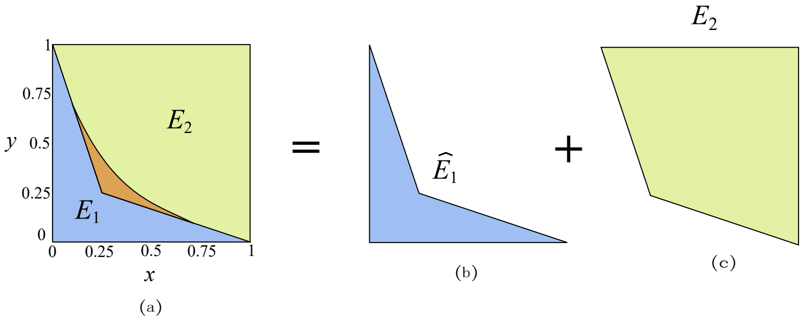
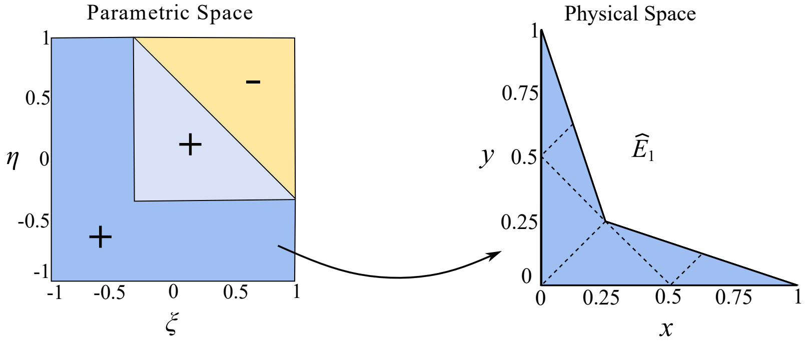


Figure 4.2: Parts contributing to the field definition.

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. 4.3a. 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. 4.3b 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 4.3: (a) Parametric space. (b) Triangulation of $\widehat{E}_1$

4.1.2 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. 4.4, 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. 4.4). Let $|\boldsymbol{J}_t|$ be the Jacobian associated with this triangle mapping. Thus, we have:

$$\widehat{\boldsymbol{k}} = \sum_{t} \int\limits_{0}^{1} \int\limits_{0}^{1-\gamma} \left(\nabla \boldsymbol{N}^{\top} \boldsymbol{D} \nabla \boldsymbol{N}\right) |\boldsymbol{J}_t| d\gamma d\zeta.$$

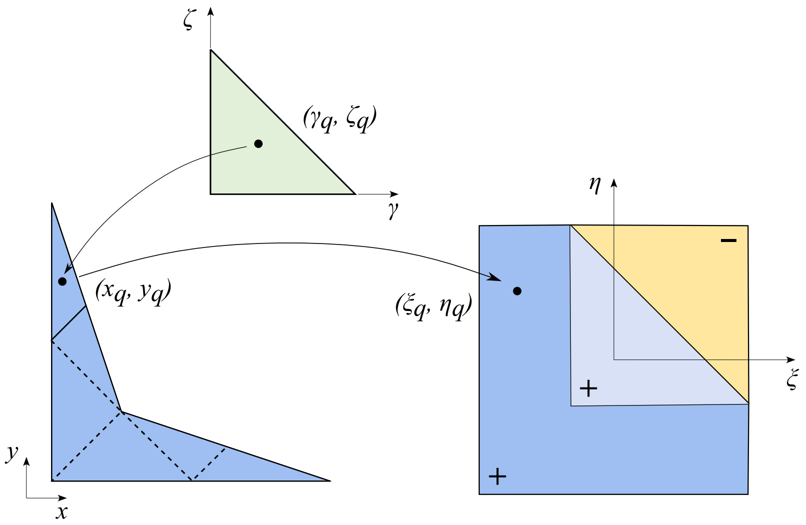


Figure 4.4: Numerical integration scheme based on triangulation of the concave element.

Consider a quadrature point (γ_q, ζ_q) of the standard triangle with weight w_q as shown in Fig. 4.4. The corresponding point (x_q, y_q) in the physical space is shown in Fig. 4.4. 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

$$\boldsymbol{B}_q = (\boldsymbol{J}_q)^{-1} \nabla_{\xi\eta} \boldsymbol{N}(\xi_q, \eta_q). \tag{4.8}$$

Summing the contribution from all triangles results in:

$$\widehat{\boldsymbol{k}} = \sum_t \sum_q (\boldsymbol{B}_q)^\top \boldsymbol{D} (\boldsymbol{B}_q) |\boldsymbol{J}_t| w_q. \tag{4.9}$$

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

4.1.3 Global Assembly

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

$$\boldsymbol{K}^0_{convex} = \prod_{Assemble-convex} \boldsymbol{k}^0_j, \tag{4.10}$$

$$\widehat{\boldsymbol{K}}_{concave} = \prod_{Assemble-concave} \widehat{\boldsymbol{k}}_j. \tag{4.11}$$

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

$$\boldsymbol{K}_{iso} = \boldsymbol{K}^0_{convex} + \widehat{\boldsymbol{K}}_{concave}. \tag{4.12}$$

Similarly the forcing term is given by:

$$\boldsymbol{f}_{iso} = \boldsymbol{f}^0_{convex} + \widehat{\boldsymbol{f}}_{concave} \tag{4.13}$$

where,

$$\boldsymbol{f}^0_{convex} = \prod_{Assemble-convex} \boldsymbol{f}^0_j \tag{4.14}$$

and

$$\widehat{\boldsymbol{f}}_{concave} = \prod_{Assemble-concave} \widehat{\boldsymbol{f}}_j. \tag{4.15}$$

Finally, we solve the following linear system of equations:

$$\begin{bmatrix} \boldsymbol{K}_{iso} & \boldsymbol{C} \\ \boldsymbol{C}^\top & \boldsymbol{0} \end{bmatrix} \begin{Bmatrix} \hat{\boldsymbol{u}} \\ \boldsymbol{\mu} \end{Bmatrix} = \begin{Bmatrix} \boldsymbol{f}_{iso} \\ \boldsymbol{0} \end{Bmatrix}. \tag{4.16}$$

where the compatibility matrix $\boldsymbol{C}$ 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.

4.1.4 i-TFEM versus o-TFEM

The main difference between o-TFEM and i-TFEM is the following. In o-TFEM, one must integrate over the tangled region and over the concave region illustrated in Fig. 4.5a (see Fig. 3.5 in Section 3.1.5). However, in i-TFEM, we only need to integrate over the concave region (see Fig. 4.5b).

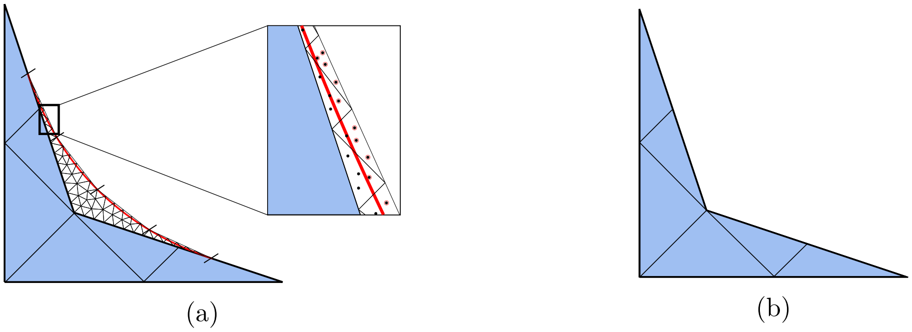


Figure 4.5: (a) o-TFEM involves integrating over the tangled region and concave region. (b) i-TFEM involves integrating only over the concave region.

In o-TFEM, to integrate over the fold, it is triangulated as shown in Fig. 4.5a. 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. 4.5a. 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. 4.5b, making i-TFEM computationally more efficient. This is demonstrated in Section 4.

4.2 Numerical Experiments

In this section, we demonstrate i-TFEM using numerical experiments. Elastostatics problems 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 [63]. 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 o-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. 3.29, and the energy error norm defined as:

$$||e_h||_{E(\Omega)} = \sqrt{\int_{\Omega} \left(\boldsymbol{\epsilon} - \boldsymbol{\epsilon}^h\right)^{\top} \boldsymbol{D} \left(\boldsymbol{\epsilon} - \boldsymbol{\epsilon}^h\right) d\Omega} \tag{4.17}$$

  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 o-TFEM? The condition number is a measure of a matrix's invertibility [66]; 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 o-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?

4.2.1 2D Patch Tests

Two element mesh

In the first experiment, we solve the elastostatics problem with linear exact solution discussed in Section 3.2.1 using i-TFEM. The mesh employed for the problem is provided here for the sake of completeness.

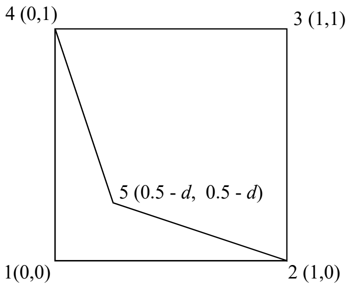
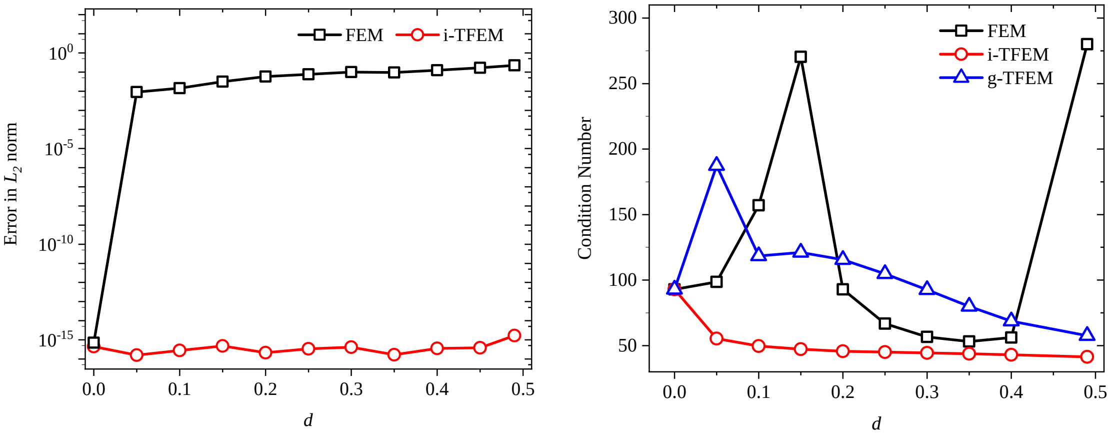


Figure 4.6: Two element mesh with implicit tangling.

Fig. 4.7a 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 (o-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 o-TFEM as illustrated in Fig. 4.7b. 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 o-TFEM respectively.



(a)

(b)

Figure 4.7: Comparison of i-TFEM and FEM for two-element mesh: (a) L^2 error vs. d, (b) condition number vs. d. Here, the legend 'g-TFEM' is 'o-TFEM'.

Four element mesh

Next, we solve the same elastostatics problem as above over the 4-element mesh as considered in Section 3.2.2 using i-TFEM. The domain and mesh is described here again.

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. 4.8. 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 $r = 0.125\sqrt{2}$.

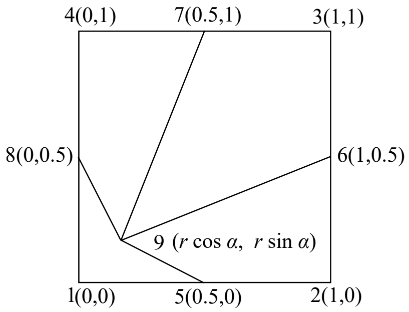
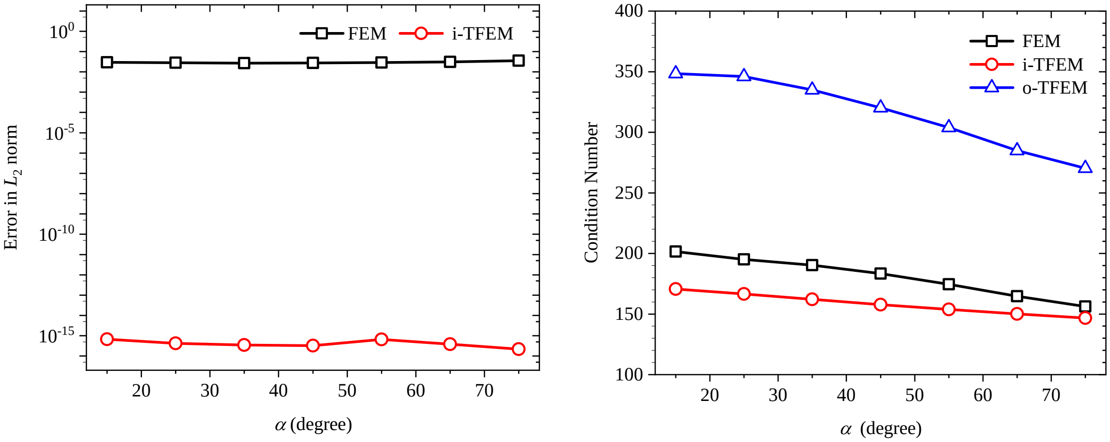


Figure 4.8: Tangled mesh with four elements; one of the elements is concave.

Fig. 4.9a illustrates the L^2 errors in FEM and i-TFEM, while Fig. 4.9b 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 o-TFEM as illustrated in Fig. 4.9b. Finally, FEM requires 0.0180s while i-TFEM and o-TFEM require 0.0237 and 2.2424s receptively. Thus, i-TFEM outperforms o-TFEM in terms of speed and condition number.



(a)

(b)

Figure 4.9: Comparison of i-TFEM and FEM for four-element mesh: (a) Error vs. α (b) Condition number vs. α.

4.2.2 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 4.10a. Assuming plane stress, the analytical solution is given by [162]:

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

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

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

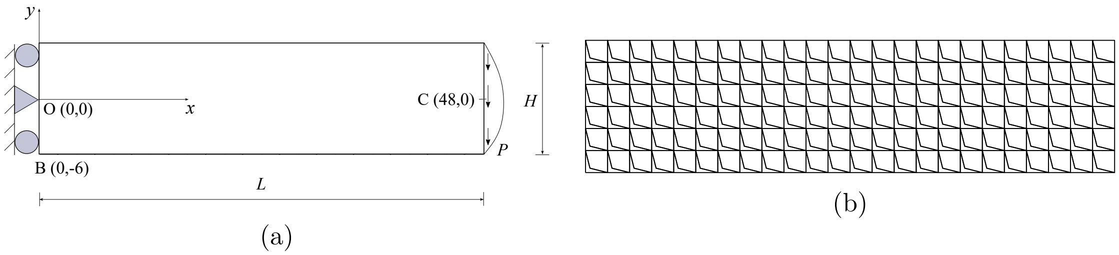


Figure 4.10: (a) Cantilever with parabolic loading and (b) mesh with concave elements.

To solve this problem using finite elements, the domain is discretized to form a tangled mesh as shown in Fig. 4.10b and the boundary conditions are applied as shown in Fig. 4.10a. Here, the basic repeating unit is a 2-element mesh (Fig. 4.6) 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. 4.11; while Fig. 4.12 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.

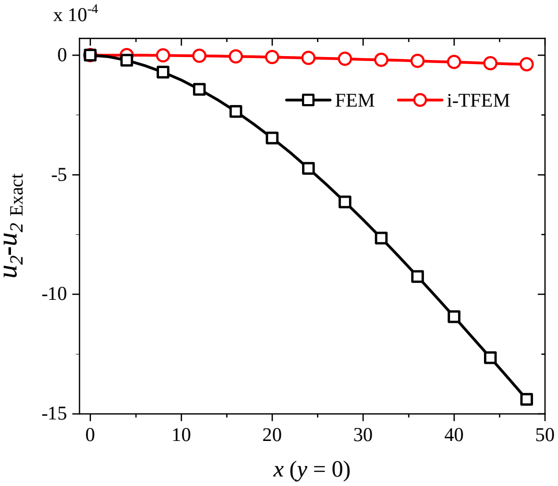


Figure 4.11: Error in u_2.

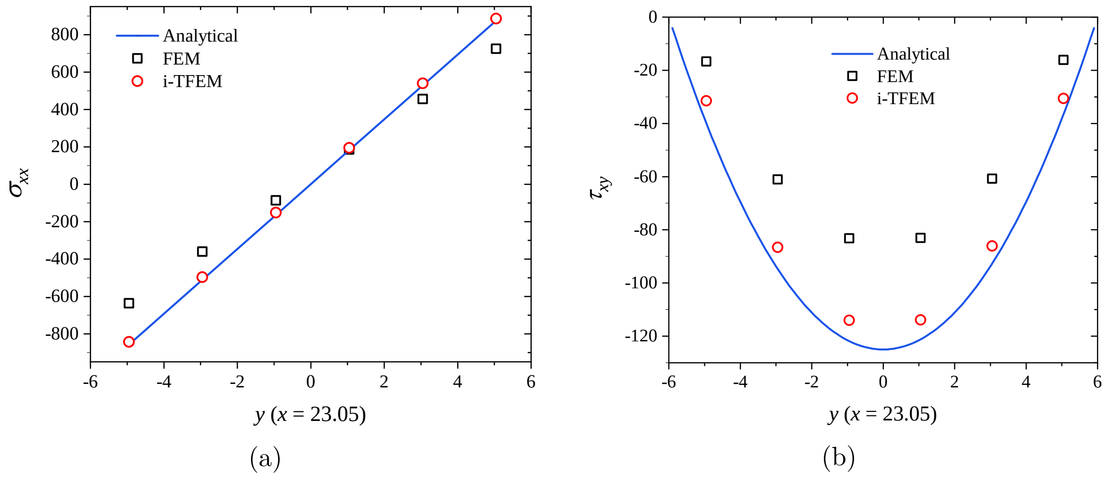


Figure 4.12: (a) σ_{xx} and (b) τ_{xy}

Convergence

To study the convergence for the cantilever beam problem, various meshes are constructed as shown in Fig. 4.13. The basic repeating unit is the two-element mesh (see Fig. 4.6) with $d = 0.4$.

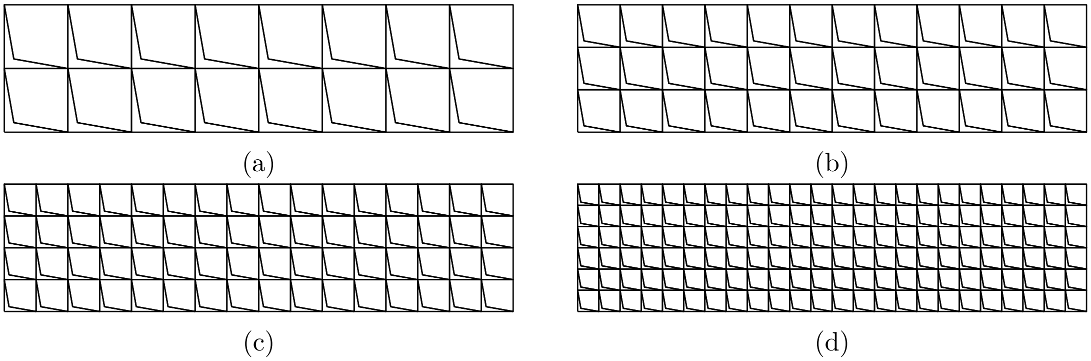
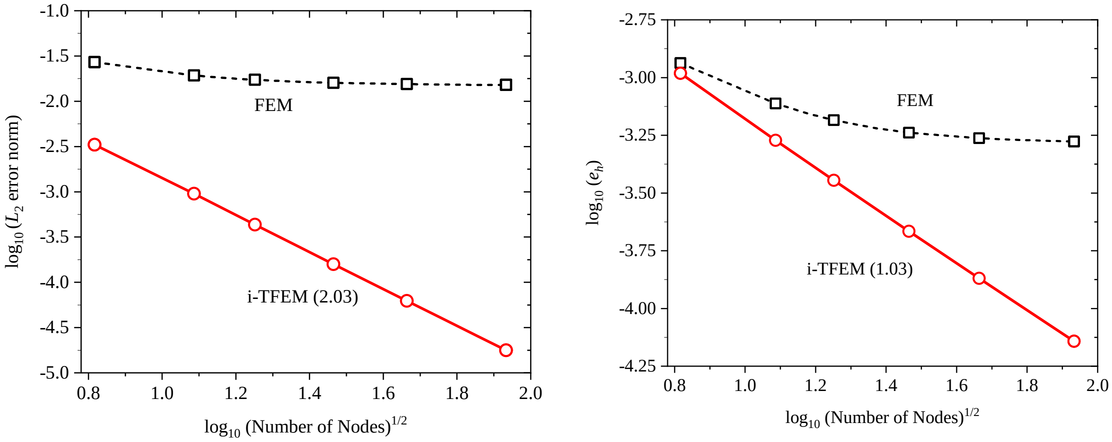


Figure 4.13: Sample meshes for convergence study with number of nodes equal to (a) 43 (b) 88 (c) 149 (d) 319.

The L^2 and energy norm errors for FEM and i-TFEM, as a function of the number of nodes, are illustrated in Fig. 4.14. For i-TFEM, the L^2 and energy norm convergence rates are 2.03 and 1.03 respectively, i.e., they are optimal.



(a)

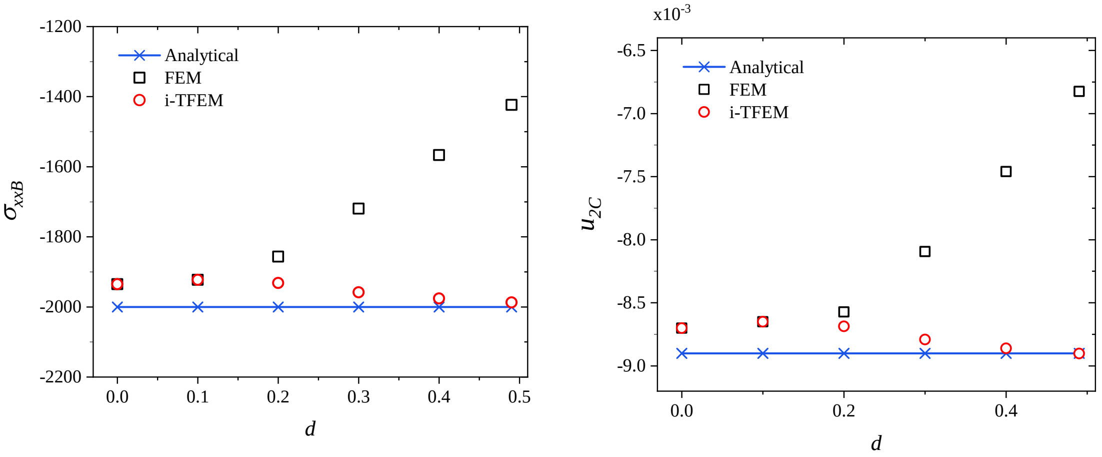
(b)

Figure 4.14: (a) L^2 and (b) energy norms errors as a function of the number of nodes for cantilever problem.

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. 4.10a.

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. 4.15a and Fig. 4.15b.



(a)

(b)

Figure 4.15: (a) σ_{xx} at point B and (b) u_2 at point C.

4.2.3 Pressurized Cylinder

Consider a long hollow cylinder with internal radius $a = 1$ and external radius $b = 4$; see Fig. 4.16a. 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 [162]:

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

A typical mesh, shown in Fig. 4.16b, is constructed by using the two-element mesh as the basic repeating unit.

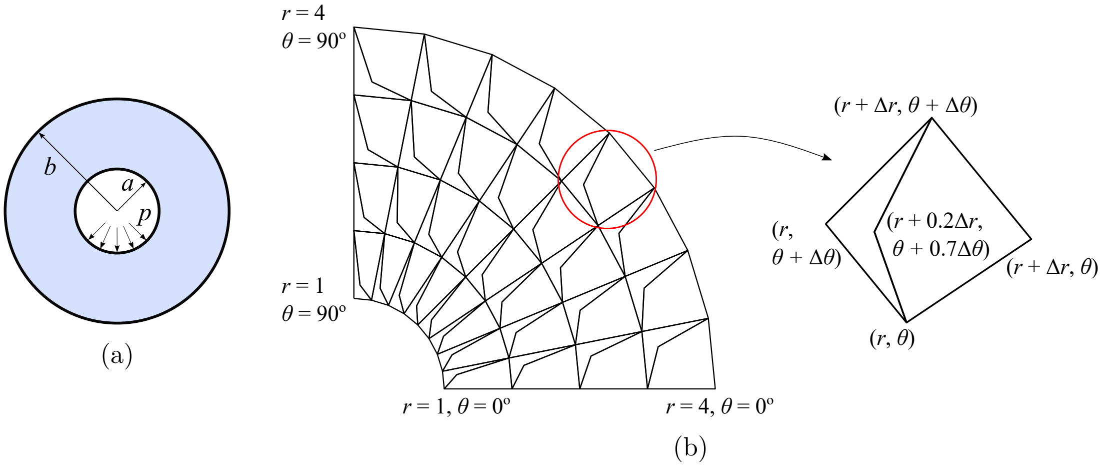


Figure 4.16: (a) Cross-section of the pressurized cylinder (b) A typical mesh with concave elements.

To study the convergence behavior, various meshes are constructed as shown in Fig. 4.17.

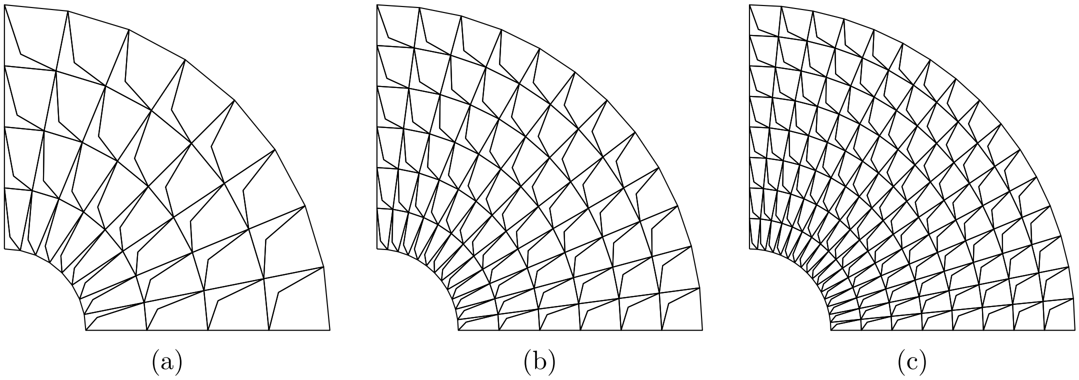
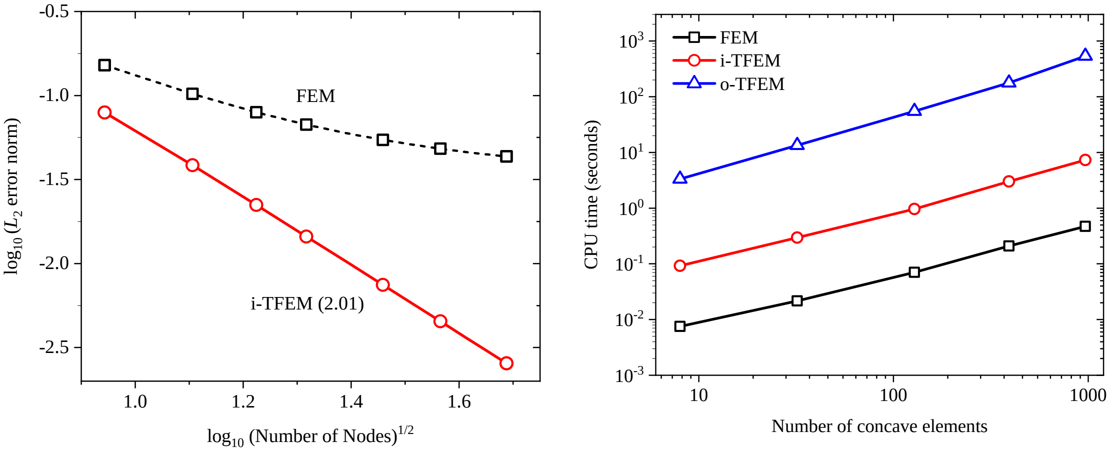


Figure 4.17: Sample meshes for convergence study. Number of nodes and the number of concave elements are (a) 77, 32 (b) 163, 72 (c) 281, 128 respectively.

Fig. 4.18a 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 o-TFEM for various mesh sizes. Fig. 4.18b shows that i-TFEM is much faster compared to the o-TFEM. FEM is the fastest, but inaccurate. Note that for the meshes considered for this study (Fig. 4.17), 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 [103, 178]. Therefore, the additional cost incurred by i-TFEM will be minimal.



(a)

(b)

Figure 4.18: (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.

4.3 Summary

In this chapter, an isoparametric tangled finite element method (i-TFEM) is presented to handle such inverted elements, specifically, inverted quadrilateral and hexahedral elements. i-TFEM exploits special properties of isoparametric elements to render the implementation 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 piece-wise invertibility by (a) modifying the elemental stiffness matrices corresponding to the concave elements and (b) incorporating the 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.

4.3.1 Limitations

We now consider the implementation of the i-TFEM framework in 3D. Consider a canonical 8-node hexahedral element in Fig. 4.19a. If node 6 is moved diagonally towards node 4, as shown in Fig. 4.19b, it can become concave, i.e., tangled.

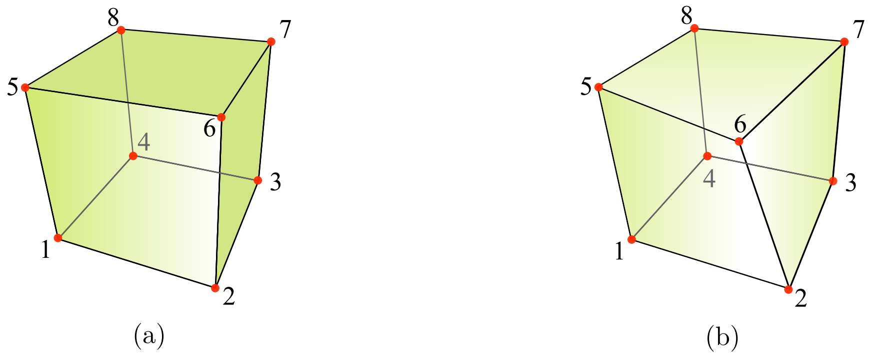


Figure 4.19: (a) Regular (untangled) hex element. (b) Tangled hex element.

Different views of the corresponding tangled region are shown in Fig. 4.20. 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 o-TFEM in 3D.

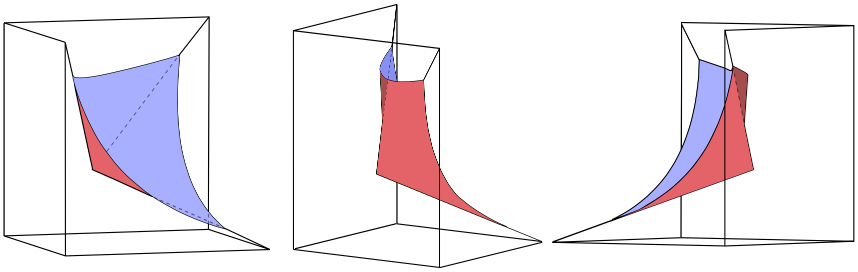


Figure 4.20: The boundary of the tangled (negative $|\boldsymbol{J}|$) region.

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. 4.21. The contribution of each tetrahedron to $\widehat{\boldsymbol{k}}_j$ is computed by generalizing the procedure discussed in Section 4.1.2.

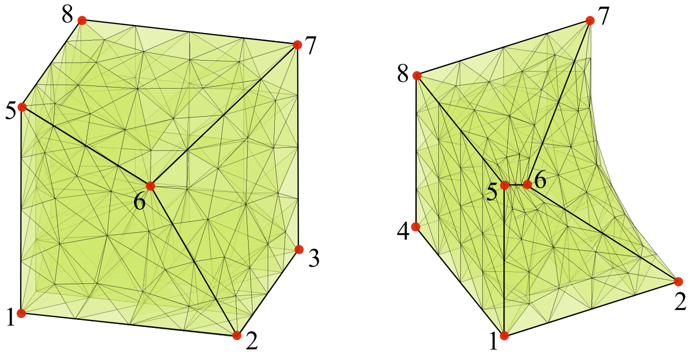


Figure 4.21: Different views of tetrahedralized concave element.

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 next.

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. 4.22a. 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. 4.22b. 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. 4.22a.

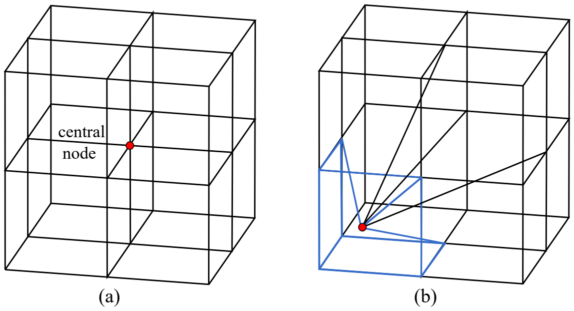


Figure 4.22: Eight-element (a) regular grid (b) tangled mesh with a concave hex element.

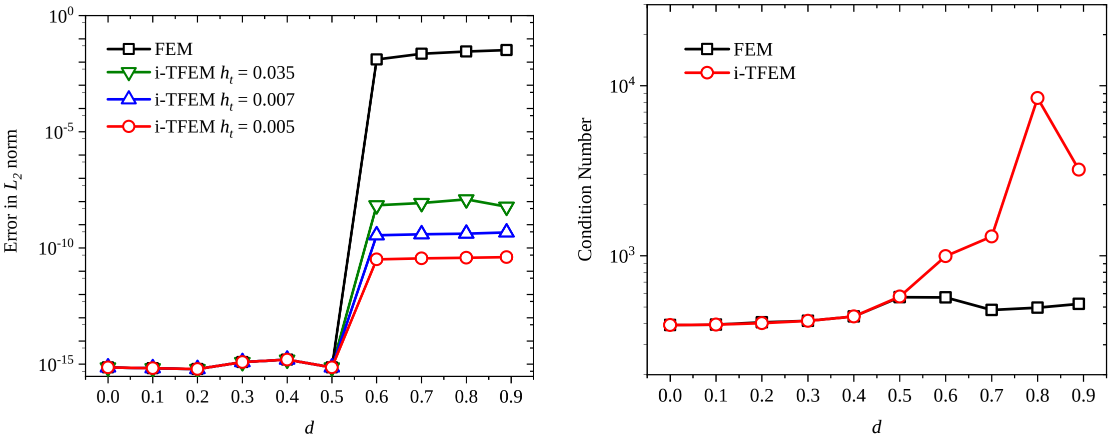
The material parameters are $E = 1$, and $\nu = 0.3$. Let the exact displacement field $\boldsymbol{u}$ be

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

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

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

As shown in Fig. 4.23a, 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. 4.23a. For $h_t = 0.005$, i-TFEM is 10^{10} times more accurate than FEM. Fig. 4.23b compares the condition number for FEM and i-TFEM (for all values of h_t).



(a) (b)

Figure 4.23: Comparison of i-TFEM and FEM for eight-element hex mesh: (a) L^2 error vs. d, (b) condition number vs. d.

Moreover, the current framework cannot directly handle penetrating elements such as the one shown in Fig. 4.24.

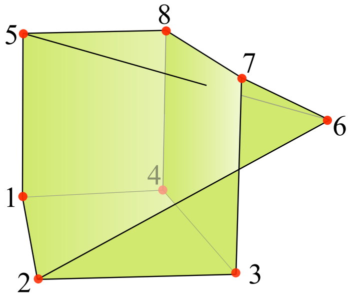


Figure 4.24: (a) Penetrating hexahedral element.

Such elements commonly occur in the practical meshes. For instance, some of the elements in the tangled mesh shown in Fig. 4.25 are penetrating.

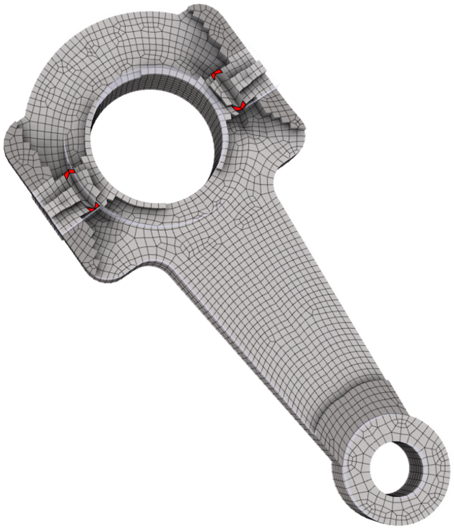


Figure 4.25: (a) Connecting rod mesh [103]; elements in red color are inverted. Some of the elements are penetrating.

To summarize, major limitations of the method are: (1) since error can be introduced due to element triangulation, accuracy can be affected. (2) penetrating elements cannot be handled.

Chapter 5

Accelerated Isoparametric TFEM

5.1 Accelerated i-TFEM Formulation

In this chapter, an accelerated isoparametric TFEM (a-TFEM) formulation is proposed that eliminates the drawbacks of the earlier formulations (namely, o-TFEM and i-TFEM). The proposed a-TFEM has particular advantage in handling elements with curved edges/non-planar faces. Here, we discuss the formulation of a-TFEM using the higher order 9-node quadrilateral (Q9) element. The formulation is further extended for the 6-node triangular elements.

5.1.1 Proposed Method

Consider the two-element mesh shown in Fig. 5.1a (the central nodes for the two Q9 elements are not shown to avoid clutter). The positive and negative components of the tangled element E_1 are shown in Fig. 5.1b. On the other hand, the non-tangled element E_2 has only one positive component (see Fig. 5.1c): $E_2 = C_2^+$ while $C_2^- = \varnothing$. Further, the fold F_1 illustrated in Fig. 5.1d overlaps with E_2 as well.

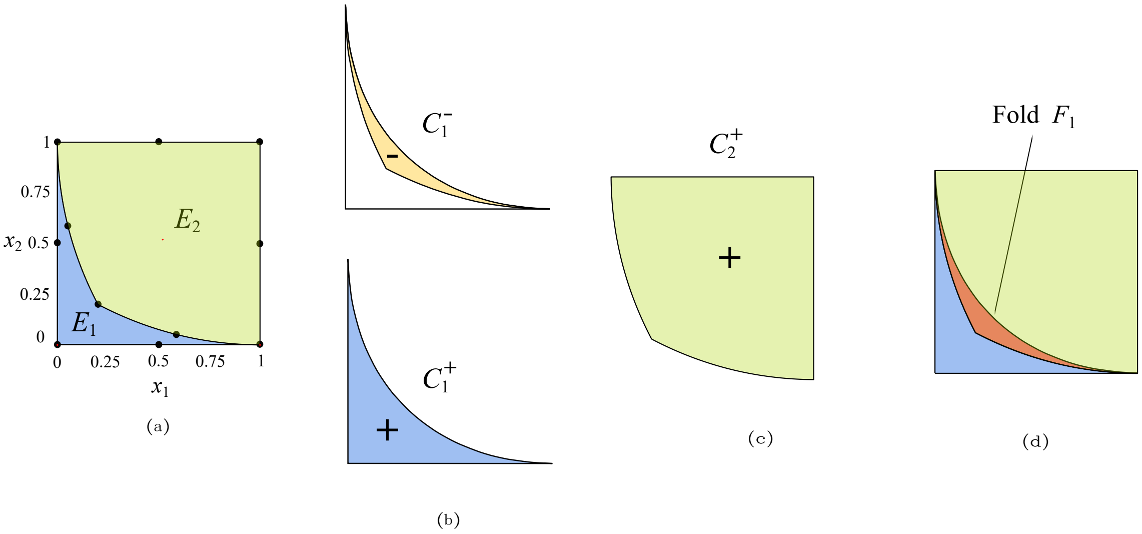


Figure 5.1: (a) 2-D domain discretized into two 9-node quadrilateral elements (central nodes are not shown to avoid clutter). (b) Positive and negative components of the tangled element. (c) Non-tangled element of the mesh. (d) Fold overlaps with neighbor.

Since any point $\boldsymbol{x}$ in the fold belongs to three components C_1^+, C_1^-, and C_2^+, one can define three fields $u_1^+(\boldsymbol{x})$, $u_1^-(\boldsymbol{x})$ and $u_2^+(\boldsymbol{x})$. To resolve this ambiguity and to ensure field continuity, a *piecewise compatibility constraint* is enforced over the fold [130]:

$$u_1^+(\boldsymbol{x}) - u_1^-(\boldsymbol{x}) = 0, \quad \forall \boldsymbol{x} \in F_1. \tag{5.1}$$

Introducing the notation $[\![\cdot]\!] = (\cdot)^+ - (\cdot)^-$, the above constraint can be written as:

$$[\![u_1]\!] = 0, \quad \text{in } F_1. \tag{5.2}$$

We will now consider solving the Poisson problem over the two-element mesh. Recall that the standard potential energy for the Poisson problem is given by:

$$\Pi = \int\limits_{E_1} \left(\frac{1}{2}(\nabla u_1)^\top \boldsymbol{D}\,(\nabla u_1) - u_1 b\right) d\Omega + \int\limits_{E_2} \left(\frac{1}{2}(\nabla u_2)^\top \boldsymbol{D}\,(\nabla u_2) - u_2 b\right) d\Omega - \sum_{j\in 1,2} \int\limits_{\partial E_j^q} u_j q dS \tag{5.3}$$

where $\boldsymbol{D}$ is the material constitutive matrix, b is the source term, and q is the boundary flux over the boundary ∂E_j^q. We will assume that the field u satisfies the Dirichlet boundary condition $u_j = u_j^d$ over the boundary ∂E_j^d. The above formulation (used in standard FEM) leads to erroneous results over the tangled mesh (later demonstrated in Section 5). On the other hand, in a-TFEM, positive and negative components are handled separately.

To motivate the concept, observe that the total area of the two-element mesh is given by:

$$A = (A_1^+ - A_1^-) + A_2^+ \tag{5.4}$$

where,

$$A_1^+ = \int\limits_{C_1^+} d\Omega, \quad A_1^- = \int\limits_{C_1^-} d\Omega, \quad \text{and} \quad A_2^+ = \int\limits_{C_2^+} d\Omega. \tag{5.5}$$

Observe that all the areas A_1^+, A_1^-, and A_2^+ are positive in value, but the area of the negative component is subtracted from the area of the positive component.

In the same vein, we propose a modified energy functional where the positive and negative components are treated separately and the contribution of the negative component is subtracted. Moreover, the compatibility constraint (Eq. 5.2) is incorporated as follows:

$$\begin{aligned}\widetilde{\Pi} = &\int\limits_{C_1^+} \left(\frac{1}{2}(\nabla u_1^+)^\top \boldsymbol{D}\,(\nabla u_1^+) - u_1^+ b\right) d\Omega \; - \int\limits_{C_1^-} \left(\frac{1}{2}(\nabla u_1^-)^\top \boldsymbol{D}\,(\nabla u_1^-) - u_1^- b\right) d\Omega \\ &+ \int\limits_{C_2^+} \left(\frac{1}{2}(\nabla u_2^+)^\top \boldsymbol{D}\,(\nabla u_2^+) - u_2^+ b\right) d\Omega \; - \sum_{j \in 1,2} \int\limits_{\partial E_j^q} u_j q dS + \int\limits_{F_1} \lambda_1 \, [\![u_1]\!] d\Omega.\end{aligned} \tag{5.6}$$

Observe that: (1) the first three terms involve integration over the components C_1^+, C_1^-, and C_2^+ respectively, (2) the contribution from the component C_1^- is subtracted since it is a negative component, (3) the fourth term is the standard boundary flux and, (4) the last term incorporates the compatibility constraint over the folds F_1 via Lagrange multiplier field λ_1 (the use of Lagrange multipliers to impose constraints is common in several finite element formulations [44, 126, 164, 49, 120, 5]).

The potential energy (Eq. 5.6) can be easily generalized for a mesh containing M elements, identified by the set $I = \{1, \ldots, M\}$. Each element E_j has an associated positive

C_j^+ component (and a negative component C_j^- if the element is tangled). The tangled elements are identified by the index I_{tangled}. The energy functional can therefore be written as:

$$\widetilde{\Pi} = \sum_{j\in I} \int\limits_{C_j^+} \left(\frac{1}{2} (\nabla u_j^+)^\top \boldsymbol{D} \left(\nabla u_j^+\right) - u_j^+ b \right) d\Omega \; - \sum_{j\in I_{\text{tangled}}} \int\limits_{C_j^-} \left(\frac{1}{2} (\nabla u_j^-)^\top \boldsymbol{D} \left(\nabla u_j^-\right) - u_j^- b \right) d\Omega$$

$$- \sum_{j\in I} \int\limits_{\partial E_j^q} u_j q dS + \sum_{j\in I_{\text{tangled}}} \int\limits_{F_j} \lambda_j \, [\![u_j]\!] d\Omega. \tag{5.7}$$

5.1.2 Weak Form

Taking the variation with respect to the two fields u and λ, we arrive at the following weak form:

Find $u \in H^1$ and $\lambda \in L^2$ such that

$$a(\delta u, u) + b(\delta u, \lambda) = f(\delta u), \quad \forall \delta u \in H_0^1 \tag{5.8a}$$

$$b(\delta\lambda, u) = 0, \qquad \forall \delta\lambda \in L^2 \tag{5.8b}$$

where,

$$a(\delta u, u) = \sum_{j\in I} \int\limits_{C_j^+} \left(\nabla \delta u_j^+\right)^\top \boldsymbol{D} \nabla u_j^+ d\Omega - \sum_{j\in I_{\text{tangled}}} \int\limits_{C_j^-} \left(\nabla \delta u_j^-\right)^\top \boldsymbol{D} \nabla u_j^- d\Omega$$

$$b(\delta u, \lambda) = \sum_{j\in I_{\text{tangled}}} \int\limits_{F_j} [\![\delta u_j]\!] \, \lambda_j \, d\Omega \tag{5.9}$$

$$f(\delta u) = \sum_{j\in I} \int\limits_{C_j^+} \delta u_j^+ b d\Omega - \sum_{j\in I_{\text{tangled}}} \int\limits_{C_j^-} \delta u_j^- b d\Omega + \sum_{j\in I} \int\limits_{\partial E_j^q} \delta u_j q dS. \tag{5.10}$$

We now approximate the primary field u and the Lagrange multiplier field λ as follows:

$$u_j \approx \boldsymbol{N}_j \hat{\boldsymbol{u}}_j, \quad \lambda_j \approx \boldsymbol{N}^\lambda \hat{\boldsymbol{\lambda}}. \tag{5.11}$$

Adopting a (Bubnov-) Galerkin method leads to the following system of equations:

$$\begin{bmatrix} \boldsymbol{K} & \boldsymbol{C} \\ \boldsymbol{C}^\top & \boldsymbol{0} \end{bmatrix} \begin{Bmatrix} \hat{\boldsymbol{u}} \\ \hat{\boldsymbol{\lambda}} \end{Bmatrix} = \begin{Bmatrix} \boldsymbol{f} \\ \boldsymbol{0} \end{Bmatrix} \tag{5.12}$$

where the stiffness matrix is given by:

$$\boldsymbol{K} = \prod_{j\in I} \int_{C_j^+} \left(\nabla \boldsymbol{N}_j^{+\top} \boldsymbol{D} \nabla \boldsymbol{N}_j^{+}\right) d\Omega \quad - \prod_{j\in I_{\text{tangled}}} \int_{C_j^-} \left(\nabla \boldsymbol{N}_j^{-\top} \boldsymbol{D} \nabla \boldsymbol{N}_j^{-}\right) d\Omega, \tag{5.13}$$

the forcing term is given by:

$$\boldsymbol{f} = \prod_{j\in I} \int_{C_j^+} \boldsymbol{N}_j^\top b \, d\Omega - \prod_{j\in I_{\text{tangled}}} \int_{C_j^-} \boldsymbol{N}_j^{-\top} b \, d\Omega + \prod_{j\in I} \int_{\partial E_j^q} \boldsymbol{N}_j^\top q \, dS, \tag{5.14}$$

and, the constraint matrix by:

$$\boldsymbol{C} = \prod_{j\in I_{\text{tangled}}} \int_{F_j} [\![\boldsymbol{N}_j]\!]^\top \boldsymbol{N}^\lambda d\Omega = \prod_{j\in I_{\text{tangled}}} \int_{F_j} \left(\boldsymbol{N}_j^+ - \boldsymbol{N}_j^-\right)^\top \boldsymbol{N}^\lambda d\Omega. \tag{5.15}$$

Observe that the Lagrange multiplier field only needs to be square integrable since its gradient does not appear in the formulation.

5.2 Implementation

We now discuss the implementation of the proposed methodology.

5.2.1 Computing the Stiffness Matrix

Consider the two-element mesh shown in Fig. 5.1. Recall that the stiffness matrix is given by:

$$\boldsymbol{K} = \int_{C_1^+} \left(\nabla \boldsymbol{N}_1^{+\top} \boldsymbol{D} \nabla \boldsymbol{N}_1^{+}\right) d\Omega \quad - \int_{C_1^-} \left(\nabla \boldsymbol{N}_1^{-\top} \boldsymbol{D} \nabla \boldsymbol{N}_1^{-}\right) d\Omega \quad + \int_{C_2^+} \left(\nabla \boldsymbol{N}_2^{+\top} \boldsymbol{D} \nabla \boldsymbol{N}_2^{+}\right) d\Omega \tag{5.16}$$

Since $C_2^+ = E_2$, the last term is simply the standard element stiffness matrix for that element, and can be computed numerically via standard Gauss integration, i.e.,

$$\boldsymbol{k}_2 = \int_{-1}^{1}\int_{-1}^{1} \left(\boldsymbol{J}^{-1}\nabla_{\boldsymbol{\xi}}\boldsymbol{N}_2\right)^\top \boldsymbol{D}\left(\boldsymbol{J}^{-1}\nabla_{\boldsymbol{\xi}}\boldsymbol{N}_2\right)|\boldsymbol{J}|d\xi_1 d\xi_2. \tag{5.17}$$

On the other hand, the first two terms in Eq. 5.16 are associated with the tangled element E_1. Recall that C_1^+ (C_1^-) gets mapped to the J^+ (J^-) region of the parametric space (see chapter 2). Accordingly, the differential area for the positive component is given by:

$$d\Omega = dx_1 dx_2 = |\boldsymbol{J}|d\xi_1 d\xi_2 \tag{5.18}$$

On the other hand, the differential area for the negative component is given by:

$$d\Omega = dx_1 dx_2 = -|\boldsymbol{J}|d\xi_1 d\xi_2 \tag{5.19}$$

Observe that since the Jacobian determinant $|\boldsymbol{J}|$ is negative, a minus sign ensures that the area remains positive.

Accordingly, the first two terms of Eq. 5.16 become:

$$\int_{C_1^+} \left(\nabla\boldsymbol{N}_1^{+\top}\boldsymbol{D}\nabla\boldsymbol{N}_1^{+}\right)d\Omega = \int_{J^+} \left(\boldsymbol{J}^{-1}\nabla_{\boldsymbol{\xi}}\boldsymbol{N}_1^{+}\right)^\top \boldsymbol{D}\left(\boldsymbol{J}^{-1}\nabla_{\boldsymbol{\xi}}\boldsymbol{N}_1^{+}\right)|\boldsymbol{J}|d\xi_1 d\xi_2 \tag{5.20}$$

and

$$\int_{C_1^-} \left(\nabla\boldsymbol{N}_1^{-\top}\boldsymbol{D}\nabla\boldsymbol{N}_1^{-}\right)d\Omega = -\int_{J^-} \left(\boldsymbol{J}^{-1}\nabla_{\boldsymbol{\xi}}\boldsymbol{N}_1^{-}\right)^\top \boldsymbol{D}\left(\boldsymbol{J}^{-1}\nabla_{\boldsymbol{\xi}}\boldsymbol{N}_1^{-}\right)|\boldsymbol{J}|d\xi_1 d\xi_2 \tag{5.21}$$

When these two terms are grouped together, they represent the entire parametric space of the tangled element which enables us to use the standard Gauss integration; albeit, with the sign of the Jacobian determinant included, i.e., one should not use the absolute value of the Jacobian determinant but its signed value.

$$\boldsymbol{k}_1 = \int_{-1}^{1} \int_{-1}^{1} \left(\boldsymbol{J}^{-1} \nabla_{\boldsymbol{\xi}} \boldsymbol{N}_1\right)^{\top} \boldsymbol{D} \left(\boldsymbol{J}^{-1} \nabla_{\boldsymbol{\xi}} \boldsymbol{N}_1\right) |\boldsymbol{J}| d\xi_1 d\xi_2. \tag{5.22}$$

Thus, to obtain the elemental stiffness matrix in a-TFEM, the standard Gauss integration can be employed for both regular and tangled elements, but the sign of the Jacobian must be retained. If all the Gauss points lie in the positive Jacobian region, then standard FEM is valid and there is no need for additional constraints. In many finite element implementations, the absolute value of the Jacobian determinant is used, but this will lead to erroneous results. *In addition, the constraint matrix must be included as discussed next.*

5.2.2 Constraint Enforcement

Next, consider the constraint matrix $\boldsymbol{C}$ in Eq. 7.22. For the two-element mesh (Fig. 7.4a), the non-zero entries of the k^{th} column of $\boldsymbol{C}$ are given by

$$\boldsymbol{C}_k = \int_{F_1} [[\boldsymbol{N}_1 (\boldsymbol{p})]]^{\top} (N^{\lambda})^k (\boldsymbol{p})\, d\Omega \tag{5.23}$$

where $\boldsymbol{p}$ denotes a point within the fold and $(N^{\lambda})^k$ is the k^{th} entry of $\boldsymbol{N}^{\lambda}$. This entails integrating over the fold, and can be done by triangulating the fold and evaluating the integrand at quadrature points for each triangle; see [129]. However, this is cumbersome and computationally expensive. To overcome this, an alternate method based on point collocation [185, 121] is employed here. The point collocation method leads to algebraic equations that arise in several finite element formulations [88, 89, 90, 169, 174], and will be discussed next.

Consider the 2-element mesh in Fig. 7.4a with one tangled element. Recall that the piecewise compatibility constraint (Eq. 7.9) implies that for any point $\boldsymbol{p}$ inside the fold:

$$[[\boldsymbol{N}_1 (\boldsymbol{p})]]\, \hat{\boldsymbol{u}}_1 = 0 \quad \text{i.e.,} \quad \left(\boldsymbol{N}_1^{+} (\boldsymbol{p}) - \boldsymbol{N}_1^{-} (\boldsymbol{p})\right) \hat{\boldsymbol{u}}_1 = 0 \tag{5.24}$$

Consider three noncollinear points ($\boldsymbol{p}_1$, $\boldsymbol{p}_2$, and $\boldsymbol{p}_3$) located in the folded region (see Fig. 5.2).

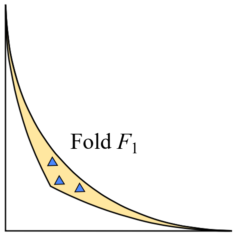


Figure 5.2: Three noncollinear points within the fold.

One can evaluate $[\![\boldsymbol{N}_1(\cdot)]\!]$ at these points in order to construct a matrix $\overline{\boldsymbol{C}}$ such that the non-zero entries of the k^{th} column of $\overline{\boldsymbol{C}}$ are given by

$$\overline{\boldsymbol{C}}_k = [\![\boldsymbol{N}_1(\boldsymbol{p}_k)]\!]^{\top} \tag{5.25}$$

From Eq. 5.24

$$\overline{\boldsymbol{C}}^{\top}\hat{\boldsymbol{u}} = \boldsymbol{0}. \tag{5.26}$$

While the two matrices $\boldsymbol{C}$ and $\overline{\boldsymbol{C}}$ are different, observe the similarities: (a) their columns are linearly independent, i.e., rank($\overline{\boldsymbol{C}}$) = rank($\boldsymbol{C}$) = n_λ (= 3 in this case), and (b) they both satisfy the piecewise compatibility constraints.

We, therefore, replace the $\boldsymbol{C}$ matrix in Eq. 3.27 with $\overline{\boldsymbol{C}}$ to obtain a different set of Lagrange multipliers $\bar{\boldsymbol{\lambda}}$. This is acceptable since $\boldsymbol{\lambda}$ itself is not of importance here. Thus the integration over the fold can be completely avoided.

In short, for Q9 elements, the number of constraint equations is equal to 3 × degree(s) of freedom per node; these are constructed by evaluating Eq. 5.25 at three sample points lying in the fold. For 2D scalar (Poisson) and 2D vector (elasticity) problems, the number of constraint equations required is 3 and 6 respectively.

5.3 Numerical Experiments

In this section, a-TFEM is demonstrated by solving 2D Poisson and elastostatics problems over various tangled meshes. Numerical experiments are conducted under the following conditions:

- 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.
- Standard quadrature (3 × 3) is employed for all the elements.
- In standard FEM, the absolute value of Jacobian determinant is employed (to be consistent with commercial FEM systems such as ANSYS).

To study the rate of convergence, we define the L^2 error norm as

$$||\boldsymbol{u} - \boldsymbol{u}^h||_{L^2(\Omega)} = \left[\int\limits_{\Omega} |\boldsymbol{u} - \boldsymbol{u}^h|^2 \, d\Omega \right]^{0.5} \tag{5.27}$$

and the energy error norm as

$$e^h = ||\nabla \boldsymbol{u} - \nabla \boldsymbol{u}^h||_{E(\Omega)} = \left[\int\limits_{\Omega} \left(\nabla \boldsymbol{u} - \nabla \boldsymbol{u}^h \right)^{\top} \boldsymbol{D} \left(\nabla \boldsymbol{u} - \nabla \boldsymbol{u}^h \right) \, d\Omega \right]^{0.5} \tag{5.28}$$

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

5.3.1 Patch Test: Poisson Problem

Consider a square domain $\Omega = (0,1) \times (0,1)$ which is discretized into four Q9 quadrilateral elements. Since we will be carrying out a patch test, Q9 elements with straight edges are considered since regular Q9 elements with curved edges (i.e., even without tangling) will not satisfy the patch test [36, 185]. The element at the bottom left corner is tangled as in Fig. 5.3. The folded region is shared by three neighboring non-tangled elements. We pose a Poisson problem where the exact field is given by:

$$u(x_1, x_2) = x_1^2 - x_2^2 + 3x_1 + 7x_2 - 6$$

The corresponding Dirichlet boundary conditions are imposed on all boundary edges.

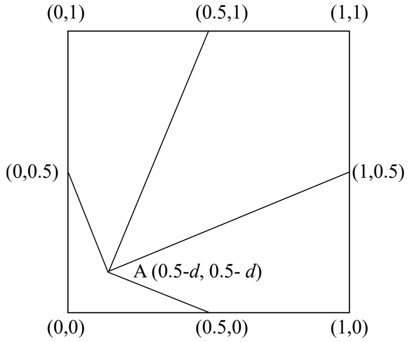
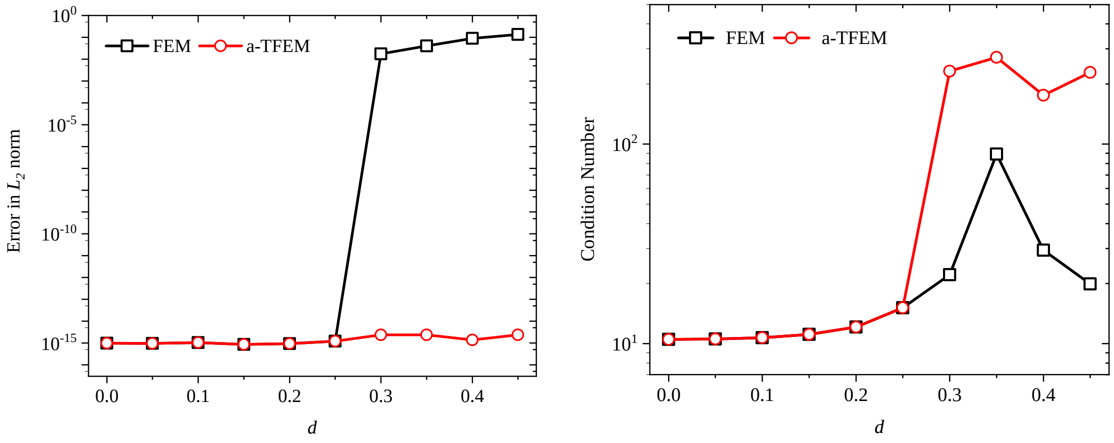


Figure 5.3: Tangled mesh with four elements; one of the elements is tangled.

To vary the extent of tangling, the position of node A is varied by the parameter d, where $d = 0$ corresponds to a regular mesh (4 square elements). Fig. 5.4a compares the L^2 errors in FEM and a-TFEM by varying d. Observe that for $d > 0.25$, the Jacobian determinant becomes negative at one or more quadrature points; consequently, FEM fails to provide accurate results. On the other hand, a-TFEM achieves machine precision accuracy even under extreme tangling. Fig. 5.4b shows that a-TFEM and FEM exhibit comparable condition numbers.



(a) (b)

Figure 5.4: Comparison of a-TFEM and FEM for four-element mesh: (a) L^2 error vs. d, (b) condition number vs. d.

5.3.2 Patch Test: Elasticity Problem

Next, we consider a cantilever subject to the loading shown in Fig. 5.5 [185]. The material parameters are: Young's modulus = 1000 and Poisson's ratio = 0.3; the exact displacements are quadratic fields [185].

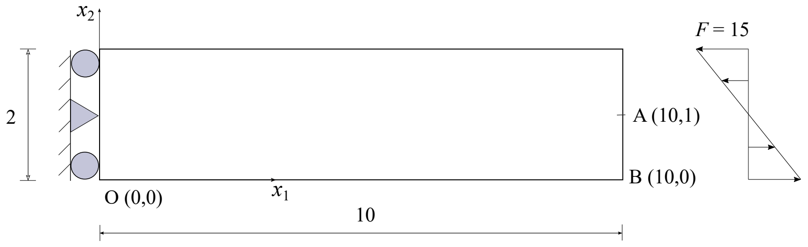
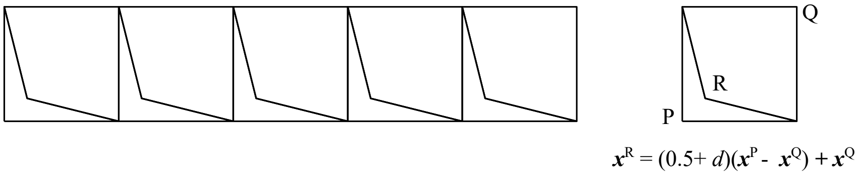


Figure 5.5: Cantilever bending problem.

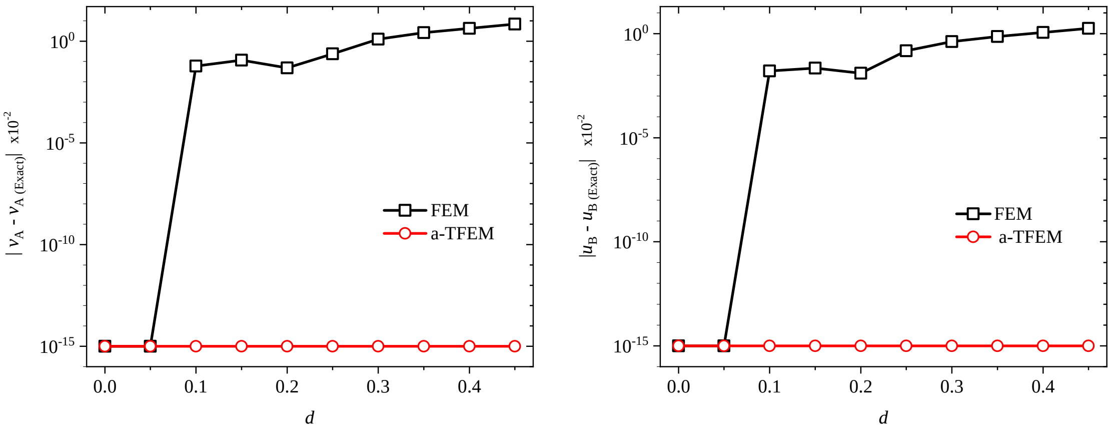
The plane stress problem is solved over a tangled mesh shown in Fig. 5.6, where, once again, the degree of tangling is controlled by a parameter d.



Q
R
P
$\boldsymbol{x}^{\mathrm{R}} = (0.5 + d)(\boldsymbol{x}^{\mathrm{P}} - \boldsymbol{x}^{\mathrm{Q}}) + \boldsymbol{x}^{\mathrm{Q}}$

Figure 5.6: Tangled mesh with ten elements; five of the elements are tangled.

Fig. 5.7a and 5.7b illustrate the errors in the vertical and horizontal displacements at points A and B respectively, using FEM and a-TFEM. As in the previous example, a-TFEM achieves machine precision accuracy while classic FEM fails.



(a)

(b)

Figure 5.7: Comparison of a-TFEM and FEM for cantilever problem: (a) Error in vertical displacement at point A vs. d (b) Error in horizontal displacement at point B vs. d.

Fig. 5.8 shows that FEM and a-TFEM exhibit comparable condition numbers.

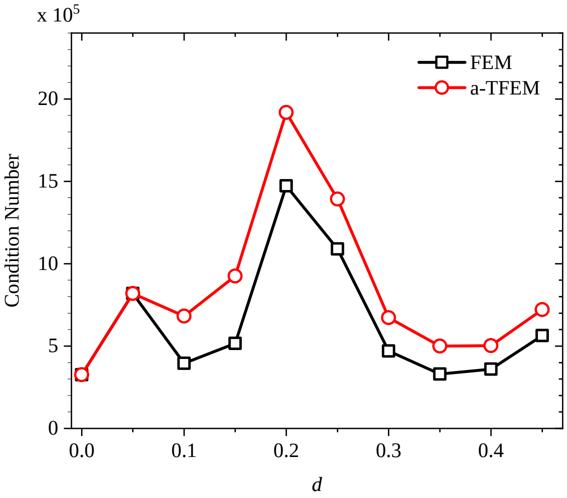


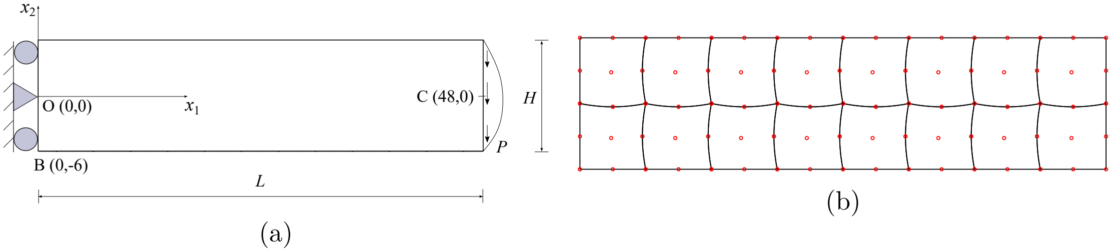
Figure 5.8: Condition number of FEM and a-TFEM for cantilever problem.

5.3.3 Cantilever with Parabolic Loading: Q9 Elements

Next, we consider a plane stress cantilever problem with a parabolic vertical load P (per unit length). The load is distributed over right edge of the cantilever and $P = 1000$, $L = 48$ and $H = 12$; see Fig. 5.9a. The material parameters are: Young's modulus $= 3 \times 10^7$ and Poisson's ratio $= 0.3$. The boundary conditions are applied as shown in Fig. 5.9a. The exact solution lies outside the FEM space and is given as [162].

$$u_1 = \frac{P x_2}{6EI}\left[(6L - 3x_1)\, x_1 + (2 + \nu)\left(x_2^2 - \frac{H^2}{4}\right)\right]$$
$$u_2 = -\frac{P}{6EI}\left[(L - x_1)\, 3\nu x_2^2 + (4 + 5\nu)\, \frac{H^2 x_1}{4} + (3L - x_1)\, x_1^2\right]$$

where $I = H^3/12$. Example of a regular mesh containing elements with curved edges is shown in Fig. 5.9b.

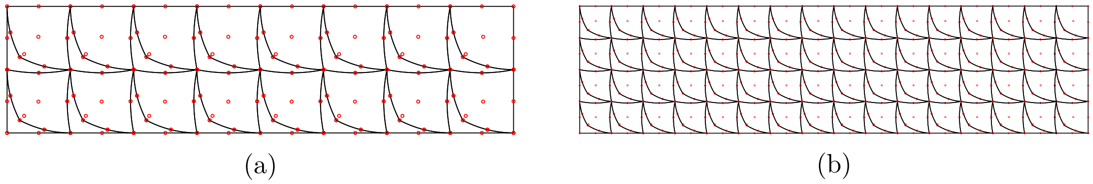


(a)

(b)

Figure 5.9: (a) Cantilever with parabolic loading and (b) the regular mesh.

To study the convergence for the cantilever beam problem, various tangled meshes are constructed as shown in Fig. 5.10.

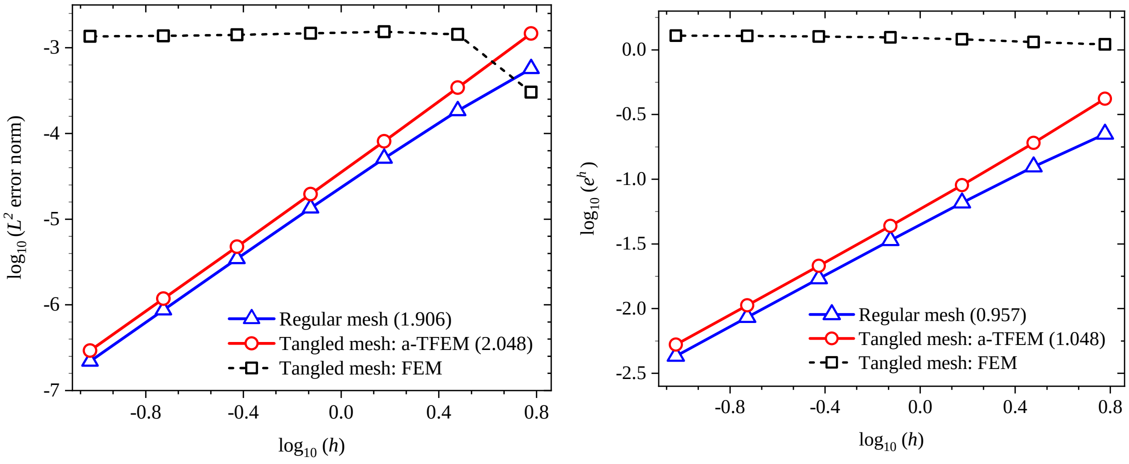


(a)

(b)

Figure 5.10: Sample tangled meshes for convergence study with element size h equal to (a) 6 (b) 3.

Fig. 5.11a and Fig. 5.11b plot the L_2 and energy error norm with respect to mesh size h for the regular mesh and tangled mesh (solved using FEM and a-TFEM) containing curved elements. Observe that the convergence rates over tangled mesh obtained via a-TFEM and over the regular mesh are similar.

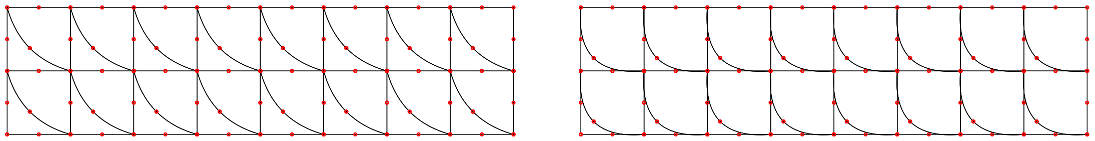


(a) (b)

Figure 5.11: (a) L^2 and (b) energy norm errors as a function of the element size h for cantilever problem with curved Q9 elements. Convergence rates are provided in parenthesis.

5.3.4 Cantilever with Parabolic Loading: Quadratic Triangular (T6) Elements

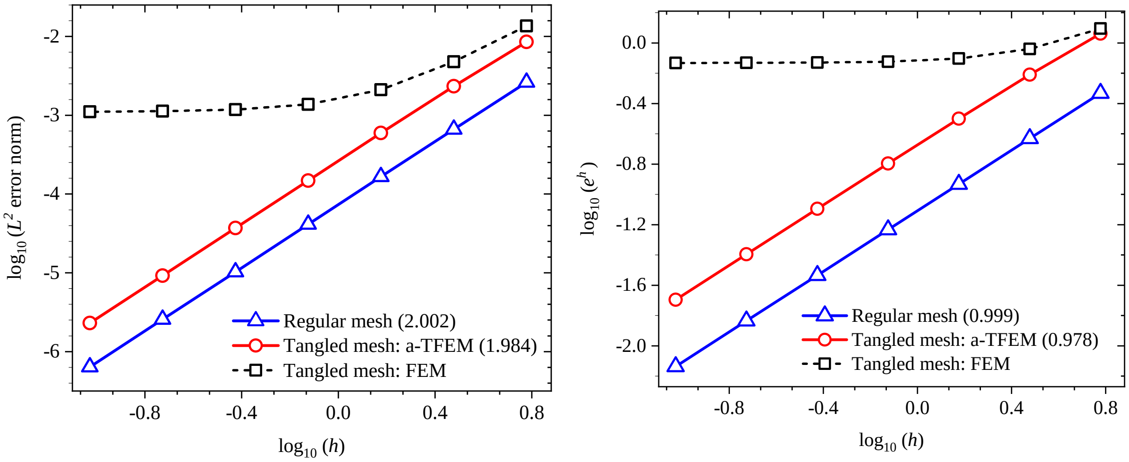
Next, we consider the mesh with quadratic triangular elements. An example of a regular mesh and a tangled mesh containing elements with curved edges is shown in Fig. 5.12.



(a) (b)

Figure 5.12: (a) Regular and (b) tangled meshes containing T6 elements with curved sides; $h = 6$.

To study the convergence properties of a-TFEM, the cantilever problem with parabolic loading previously discussed in Section 5.3.3 is solved over the regular and tangled mesh using a-TFEM and FEM. Fig. 5.13a and Fig. 5.13b plot the L_2 and energy error norms vs mesh size h. Once again, the convergence rate of a-TFEM over the tangled mesh is similar to that of FEM over the regular mesh.

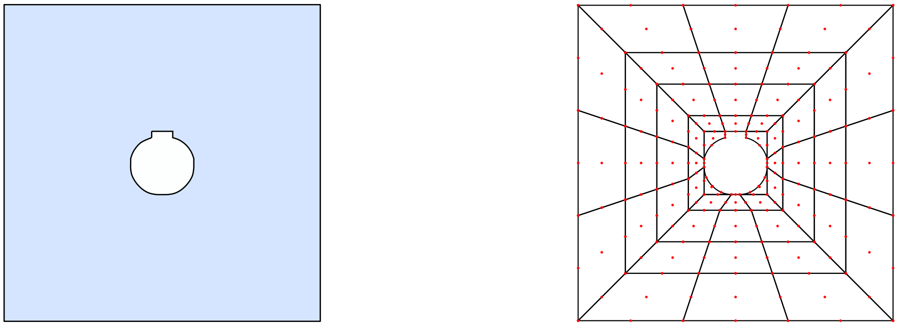


(a) (b)

Figure 5.13: (a) L^2 and (b) energy norm errors as a function of the element size h for cantilever problem with T6 elements. The convergence rates are provided in the brackets.

5.3.5 Application: Mesh Morphing with Q9 Elements

Consider *mesh morphing* where the mesh is updated under geometric changes by simply moving the mesh nodes according to a specified rule [154] (rather than regenerating the mesh which can change the underlying mesh topology). As an example, consider the domain in Fig. 5.14a, with the initial mesh as illustrated in Fig. 5.14b.



(a) (b)

Figure 5.14: (a) Domain and (b) Initial mesh containing Q9 elements.

The inner boundary is now rotated in a counterclockwise direction by an angle β, and the mesh is morphed as follows. All nodes except the outer boundary nodes are rotated about the center such that the angle of rotation exponentially increase from 0° to β as nodes get closer to the void. Fig. 5.15 illustrates the morphed mesh for $\beta = 70°$. Observe that some quads (shown in red) are tangled, i.e., the mesh is tangled.

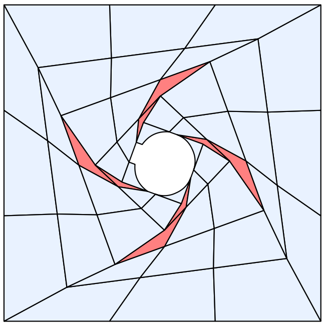
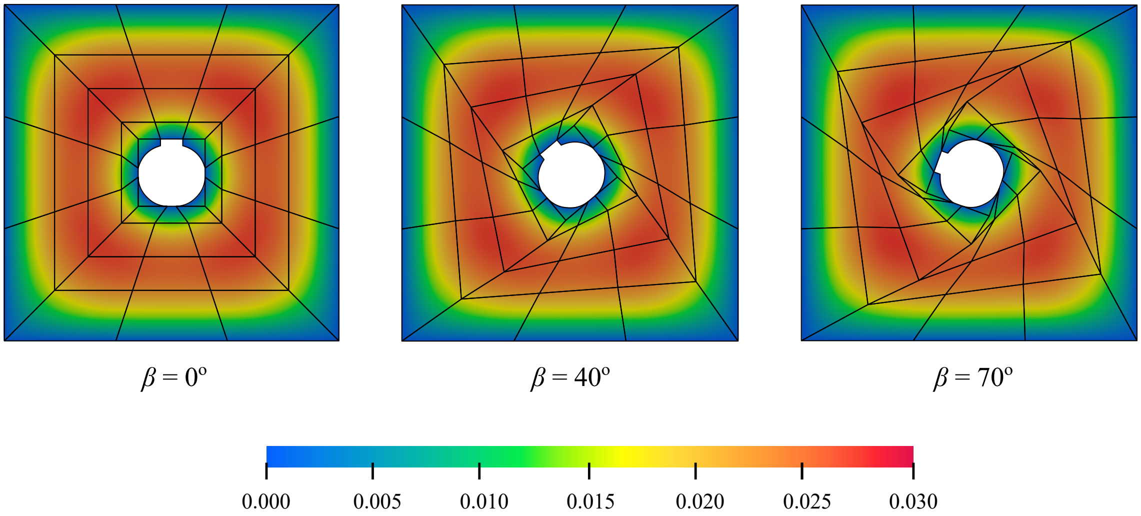


Figure 5.15: Morphed mesh for $\beta = 70°$.

We solve the Poisson problem using a-TFEM with the zero (homogeneous) Dirichlet condition applied on the entire boundary and the source term set to $b(x_1, x_2) = 1$. Fig. 5.16 illustrates the post-processed results for $\beta = 0°$, 40°, and 70°.



$\beta = 0°$

$\beta = 40°$

$\beta = 70°$

Figure 5.16: Post-processed solution using a-TFEM for morphed meshes.

5.3.6 Real-world Scenario: Disc with T6 Elements

Consider the T6 mesh illustrated earlier in Fig. 5.17 where two (out of 295) elements, shown in red color, are tangled. Though it is possible to untangle this mesh [155], a-

TFEM completely eliminates the need for untangling. Here, we compare the results for tangled and untangled meshes.

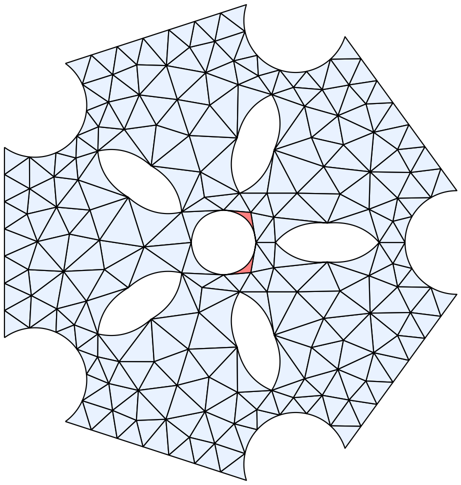


Figure 5.17: Tangled mesh with 6-node triangular elements; the mesh has been provided by the authors of [155]. Highlighted elements are tangled (negative Jacobian elements).

An elastostatics plane stress problem is set up with the following boundary conditions: homogeneous Dirichlet boundary condition is applied on the outer boundary while a counterclockwise torque $= 1 \times 10^9$ is applied on the innermost circular boundary. The problem is solved using a-TFEM over the tangled mesh with material parameters $E = 2 \times 10^{11}$ and Poisson's ratio $= 0.3$. Fig. 5.18a and Fig. 5.18b illustrate the displacement field for the tangled mesh (using a-TFEM) untangled mesh. a-TFEM required 1.200 milliseconds while FEM required 1.199 milliseconds to solve the problem.

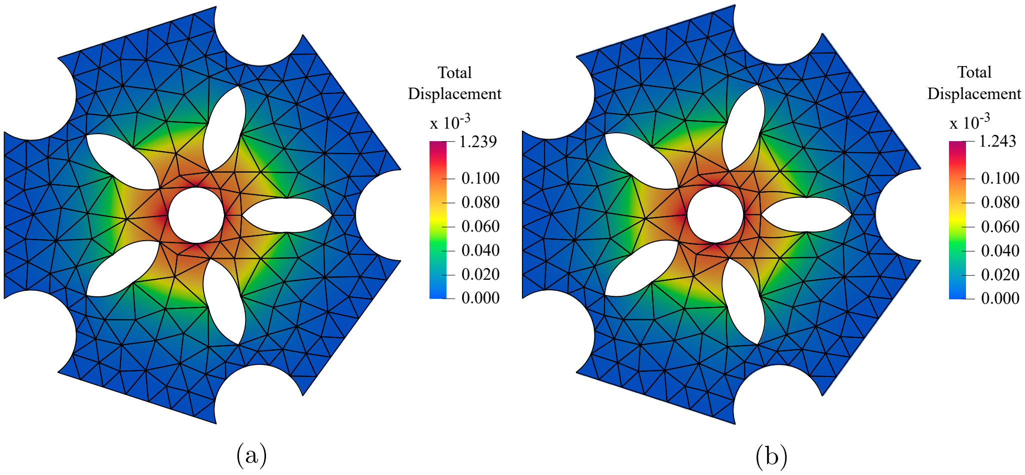


Figure 5.18: (a) Displacement field obtained using a-TFEM (c) Displacement field using untangled mesh. Both meshes were provided by the authors of [155].

5.3.7 Pressurized Cylinder: Need for Constraints

Recall that a-TFEM requires enforcement of the constraint equations. We emphasize the importance of constraints through an illustrative example considered in Chapter 1. It is repeated here for the sake of completeness.

Consider a long hollow cylinder with internal radius $a = 1$ and external radius $b = 4$; see Fig. 5.19a. A uniform pressure $p = 1$ is applied to the inner surface $(r = a)$ and the cylinder deforms in plane strain. Let the 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 [162]:

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

A typical 4-node quadrilateral mesh, shown in Fig. 5.19b, is constructed by using the two-element mesh as the basic repeating unit.

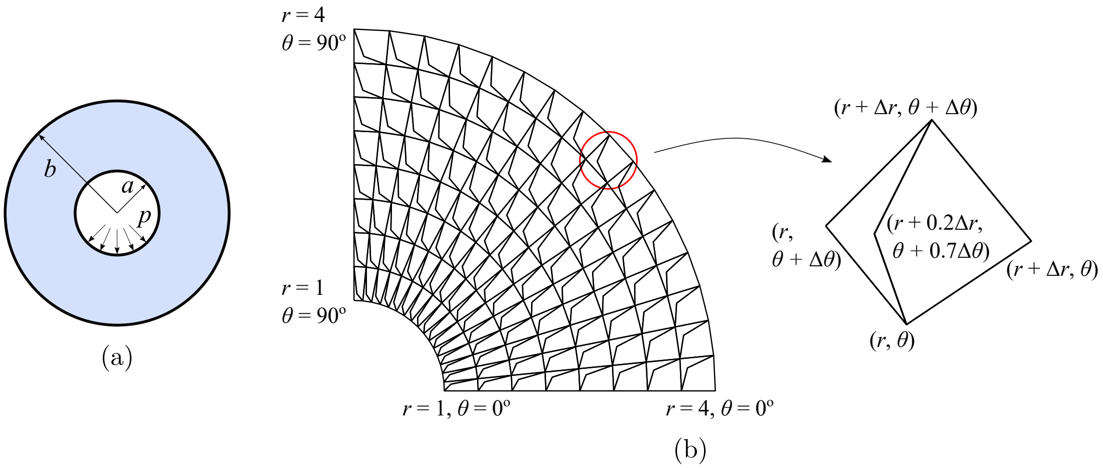


Figure 5.19: (a) Cross-section of the pressurized cylinder (b) A typical mesh with tangled elements.

Fig. 5.20a and Fig. 5.20b respectively illustrate the radial stress distribution obtained with and without compatibility constraints. Similarly, Fig. 5.21a and Fig. 5.21b respectively show the displacement field obtained with and without incorporating the compatibility constraints. The solutions are plotted over the undeformed mesh. The results produced by the two methods are quite different; the one without constraints leads to unacceptable results; Fig. 5.22 illustrates the deformed mesh. Thus, erroneous or even nonsensical results may be produced in the absence of constraint equations.

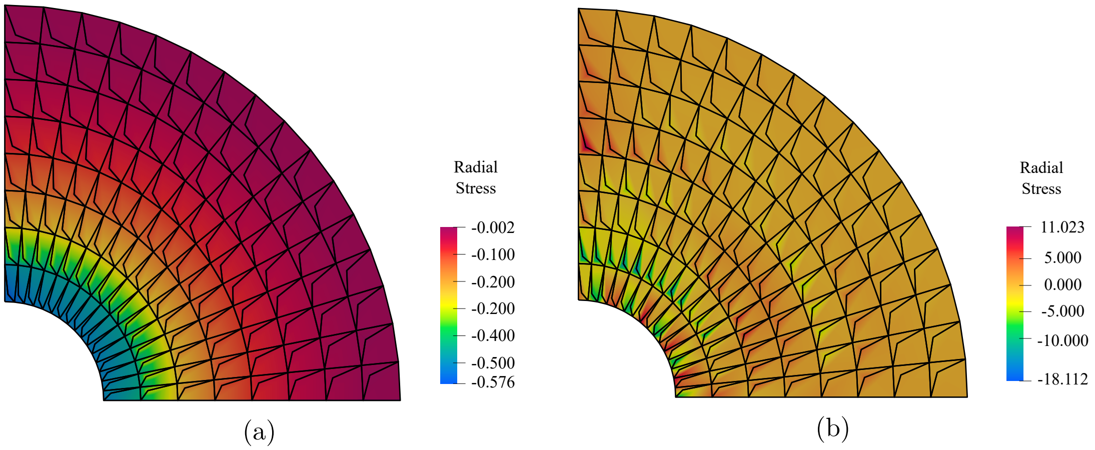


Figure 5.20: Radial stress distribution for the pressurized cylinder problem (a) with and (b) without the compatibility constraints plotted over underformed mesh.

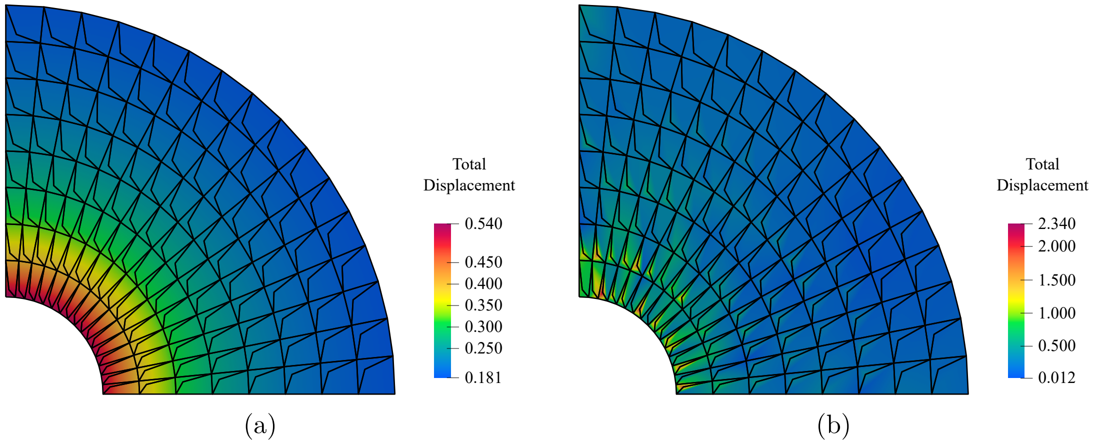


Figure 5.21: Displacement field for the pressurized cylinder problem (a) with and (b) without the compatibility constraints plotted over un-deformed mesh.

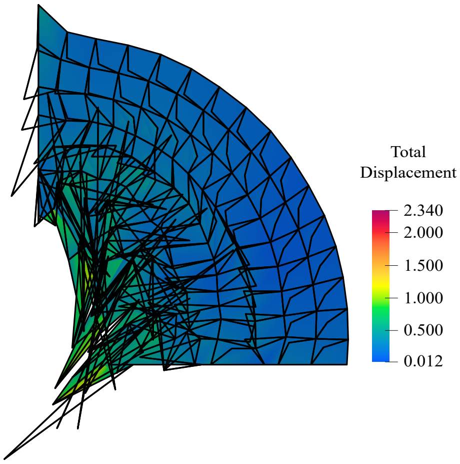


Figure 5.22: Deformed configuration for the pressurized cylinder problem without the compatibility constraints.

Chapter 6

Tangled Hexahedral Elements using Accelerated-TFEM

In this chapter, we delve into the application of a-TFEM to efficiently handle tangled hexahedral meshes. While the hexahedral element is preferred over the tetrahedral element for its solution efficiency [9], generating high-quality hexahedral meshes for complex objects remains challenging. As stated in [140], "*Generating high quality conformal hexahedral meshes in arbitrary 3D domains is one of the most challenging open problems in mesh generation.*"

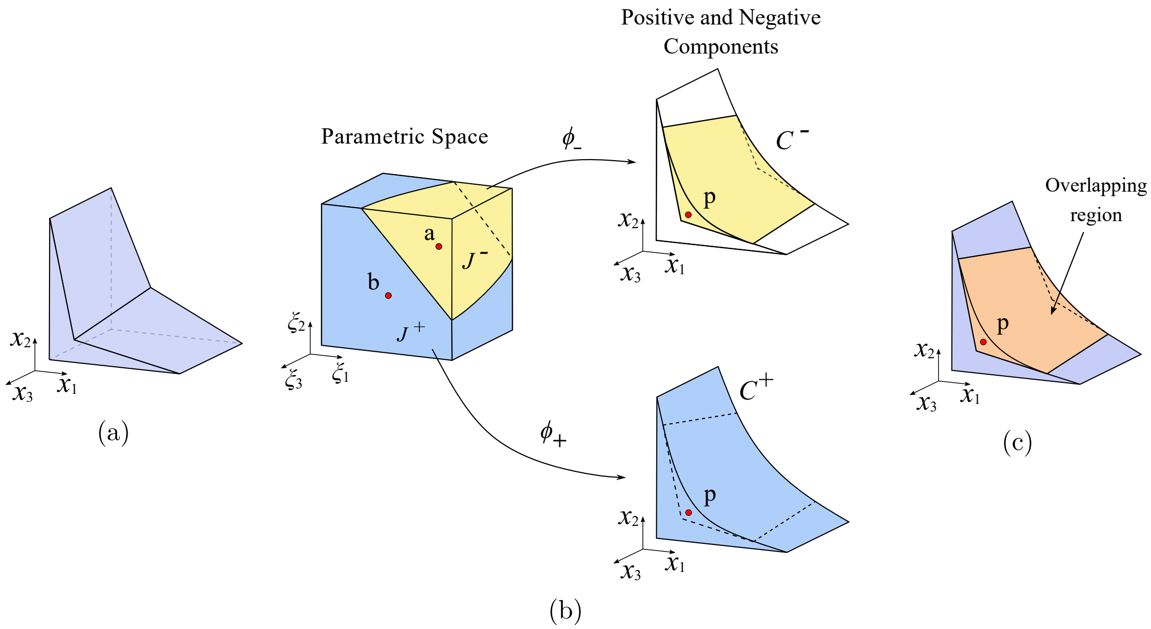
Hex mesh algorithms can be broadly classified into two types, namely the indirect method and the direct method, each with its own set of strengths and limitations [184, 127]. Indirect methods first generate a transitional mesh, such as a tetrahedral mesh, and then combine or cut the transitional mesh to obtain hexahedral/hex-dominant meshes. Examples of such methods include octree-based approaches, grid-based methods [110, 182, 100, 55, 181]. The most common hex-dominated meshing methods [125] have transitional elements, such as pyramid elements, that can lead to calculation errors in regions with high gradients. The octree method is a popular hex-mesh generation technique and can create a high-quality hexahedral mesh within the solution domain. However, it results in poor-quality irregular elements at the boundary, which is fatal in simulation. Attempts have been made to incorporate fully automatic grid-based methods [110, 182, 100, 55, 181] into professional software such as CUBIT [38] and Distene SAS [39]. However, these methods face similar challenges as octree method and are generally considered inferior in terms of mesh quality [127].

On the other hand, direct methods, such as sweeping, polycube approaches, produce high quality mesh and are commonly employed in the commercial software systems [127, 106] such as CUBIT [38], ICEM CFD [163]. However, they often require significant user interaction. For instance, in sweeping method, heavy user interaction is required to simplify and decompose complex geometries until they become suitable for meshing. Polycube mapping has emerged as a highly active research topic in recent years. However, the quality of elements is highly dependent on user interactions; at times creating tangled elements [184]. Methods such as [159] produce boundary conforming meshes; however require manual interaction and do not guarantee to produce tangle-free meshes.

In short, automatic high-quality tangle-free hex mesh generation is still elusive. The attempt here is to relax the tangle-free requirement to ease the burden on mesh generators with the help of a-TFEM. Penetrating tangled elements, which often occur in practical meshes, can be easily handled using a-TFEM as discussed next.

6.1 a-TFEM for Hex Element

Until now, we considered 2D elements where the negative component is completely contained within the positive component. An equivalent 3D example of such an element is shown in Fig. 6.1.



(a)

(b)

(c)

Figure 6.1: (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.

Penetrating elements are frequently encountered in real-world meshes, and the proposed a-TFEM formulation is capable of handling such tangled elements.

Recall that, in such elements the negative component C^- is *not* entirely contained lie within the positive component C^+. Figure 2.4a illustrates a penetrating element along with the corresponding parametric space in Fig. 2.4b. One can show that for this element, only a part of J^- region, shown in yellow in Fig. 2.4c maps to an overlapping region within C^+ that lies outside the element, while the remainder J^-, shown in red in Fig. 2.4c, 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.

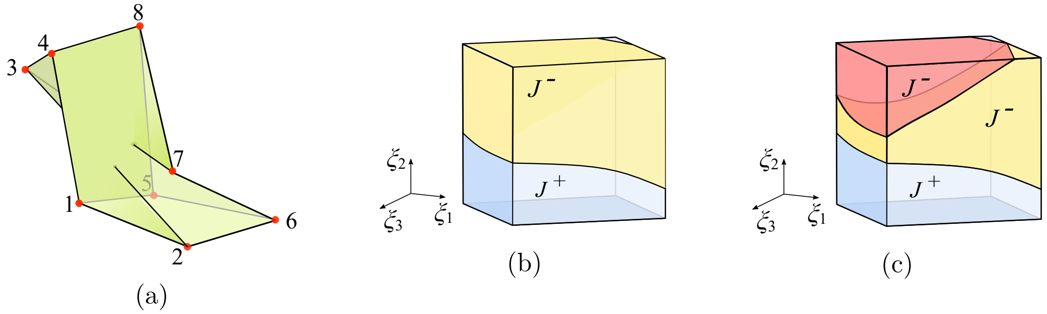


Figure 6.2: (a) Penetrating element (b) its parametric space; J^- region is shown in yellow color (c) J^- region which does not map to the fold is shown in red color.

Furthermore, there are cases where elements have disconnected J^- regions; Fig. 2.5 illustrates one such penetrating element. However, no special treatment is needed for such elements.

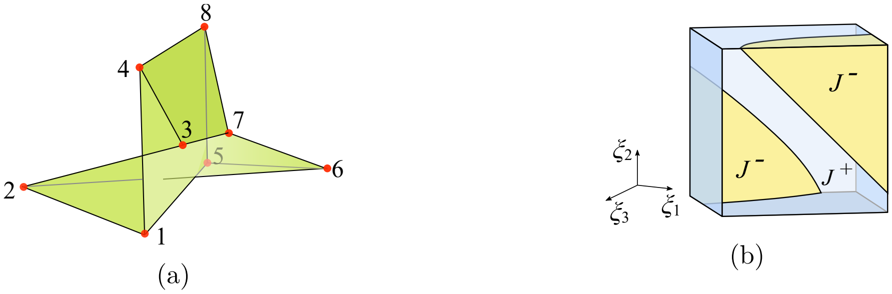


Figure 6.3: (a) Penetrating element with disconnected J^- regions, (c) its parametric space; J^- shown in yellow color.

The difference between penetrating and non-penetrating elements is only relevant to Section 6.1.1.

The same 2D formulation of a-TFEM in the previous chapter is applicable here. It is provided here for the sake of completeness. 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 following system is solved :

$$\boldsymbol{K}\hat{\boldsymbol{u}} + \boldsymbol{C}\hat{\boldsymbol{\lambda}} = \boldsymbol{f} \tag{6.1a}$$

$$\boldsymbol{C}^{\top}\hat{\boldsymbol{u}} = \boldsymbol{0} \tag{6.1b}$$

where the stiffness matrix is given by:

$$\boldsymbol{K} = \prod_{j\in I} \int_{C_j^+} \left(\nabla \boldsymbol{N}_j^{+\top} \boldsymbol{D} \nabla \boldsymbol{N}_j^{+}\right) d\Omega \quad - \prod_{j\in I_{\text{tangled}}} \int_{C_j^-} \left(\nabla \boldsymbol{N}_j^{-\top} \boldsymbol{D} \nabla \boldsymbol{N}_j^{-}\right) d\Omega, \tag{6.2}$$

the forcing term by:

$$\boldsymbol{f} = \prod_{j\in I} \int_{C_j^+} \boldsymbol{N}_j^{+\top} \boldsymbol{b}\, d\Omega \quad - \prod_{j\in I_{\text{tangled}}} \int_{C_j^-} \boldsymbol{N}_j^{-\top} \boldsymbol{b}\, d\Omega \quad + \prod_{j\in I} \int_{\partial E_j^t} \boldsymbol{N}_j^{\top} \boldsymbol{t}\, dS \tag{6.3}$$

and the constraint matrix by:

$$\boldsymbol{C} = \prod_{j\in I_{\text{tangled}}} \int_{C_j^-} \left(\boldsymbol{N}_j^{+} - \boldsymbol{N}_j^{-}\right)^{\top} \boldsymbol{N}^{\lambda} d\Omega. \tag{6.4}$$

In Eq. 6.4, the choice of $\boldsymbol{N}^{\lambda}$ and computation of the integral is discussed next.

6.1.1 Computing the Constraint Matrix

Now consider a simple domain Ω that is discretized into two elements shown in Fig. 6.4a, 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. 6.4b. 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. 6.4c), i.e., $E_2 = C_2^+$ while $C_2^- = \varnothing$. 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.

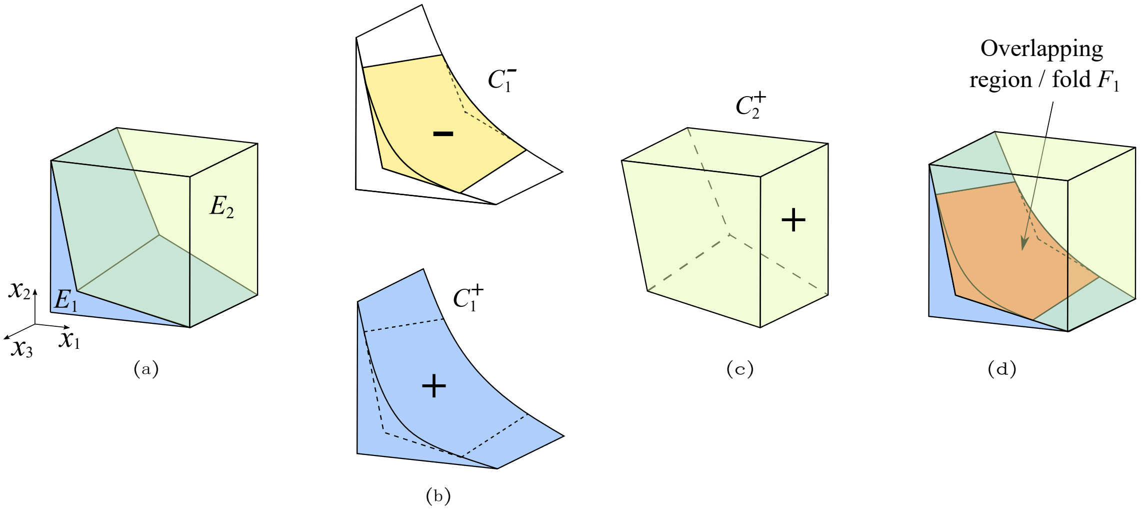


Figure 6.4: (a) 3-D domain discretized into two 8-node hexahedral elements. (b) Positive and negative components of the tangled element. (c) Non-tangled element of the mesh. (d) Overlapping region intersects with the neighboring convex element.

For the two-element mesh in Fig. 6.4a, compatibility constraint becomes:

$$\int_{F_1} \delta\boldsymbol{\lambda}_1 \cdot (\boldsymbol{u}_1^+ - \boldsymbol{u}_1^-)\, \mathrm{d}\Omega = 0 \quad \forall\ \delta\boldsymbol{\lambda}_1 \in L^2, \tag{6.5}$$

This can be expressed in matrix form as

$$\boldsymbol{C}^\top \hat{\boldsymbol{u}} = \boldsymbol{0} \quad \text{where} \quad \boldsymbol{C} = \int_{C_1^-} (\boldsymbol{N}_1^+ - \boldsymbol{N}_1^-)^\top\, \boldsymbol{N}^\lambda d\Omega \tag{6.6}$$

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. 6.1, 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. 6.4, we can express the constraint matrix defined in Eq. 6.6 as

$$\boldsymbol{C} = \int\limits_{F_1} \left(\boldsymbol{N}_1^{+} - \boldsymbol{N}_1^{-}\right)^{\top} d\Omega. \tag{6.7}$$

Direct integration over the tangled region F_1 to compute $\boldsymbol{C}$ is computationally expensive and cumbersome [129]. Instead, we evaluate the integrand at a sample point $\boldsymbol{p} \in F_1$, i.e., evaluate the $\boldsymbol{C}$ as

$$\boldsymbol{C} = \left(\boldsymbol{N}_1^{+}(\boldsymbol{p}) - \boldsymbol{N}_1^{-}(\boldsymbol{p})\right)^{\top}. \tag{6.8}$$

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:

$$\boldsymbol{C} = \left(\boldsymbol{N}_1(\boldsymbol{b}) - \boldsymbol{N}_1(\boldsymbol{a})\right)^{\top} \tag{6.9}$$

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:

$$\boldsymbol{p} = \boldsymbol{N}_j(\boldsymbol{a})\hat{\boldsymbol{x}}_j \tag{6.10}$$

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. 6.1b, the point $\boldsymbol{p}$ corresponding to every $\boldsymbol{a} \in \mathcal{L}$ lies within the fold. However, for penetrating elements (shown in Fig. 6.3a), 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. 6.9.

6.1.2 Fold Shared by Multiple Elements

When the fold is shared by multiple elements, such as in Fig. 6.5, 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:

$$\int\limits_{E_1+E_2+E_3+E_4} d\Omega = \int\limits_{C_1^+} d\Omega + \int\limits_{C_2^+} d\Omega + \int\limits_{C_3^+} d\Omega + \int\limits_{C_4^+} d\Omega - \int\limits_{C_1^-} d\Omega \tag{6.11}$$

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

$$\boldsymbol{u}_1^+ - \boldsymbol{u}_1^- = 0 \quad \text{in } F_1 \tag{6.12}$$

Consequently, the elemental stiffness matrices, forcing vectors, and constraint matrix are computed as before: (1) standard stiffness matrices $\boldsymbol{k}_i$, $i = 1, \ldots, 4$ 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. 6.8 at a point within the fold F_1, employing the method described above. No additional treatment is needed.

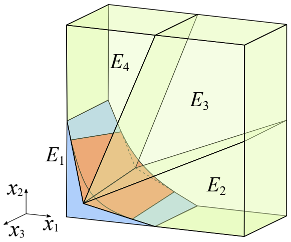


Figure 6.5: Overlapping region shared by multiple non-tangled elements.

6.2 Numerical Experiments

In this section, the proposed 3D a-TFEM is demonstrated by solving benchmark linear and nonlinear elasticity problems using various synthetically generated and real-world tangled meshes. Numerical experiments are conducted under the following conditions:

- Standard Gaussian quadrature (2×2×2) is employed for all the hexahedral elements.
- 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 [130], or result in non-convergence (see Section 7.4.3).
- Homogeneous isotropic material is assumed, unless otherwise mentioned.

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

$$\frac{||\boldsymbol{u}^{\text{ref}} - \boldsymbol{u}||_{L^2(\Omega)}}{||\boldsymbol{u}^{\text{ref}}||_{L^2(\Omega)}} = \left[\frac{\int_\Omega |\boldsymbol{u}^{\text{ref}} - \boldsymbol{u}|^2 \, d\Omega}{\int_\Omega |\boldsymbol{u}^{\text{ref}}|^2 \, d\Omega}\right]^{0.5} \tag{6.13}$$

and the relative error in energy norm as:

$$e_E = \left[\frac{\int_\Omega \left(\nabla \boldsymbol{u}^{\text{ref}} - \nabla \boldsymbol{u}\right)^\top \boldsymbol{D} \left(\nabla \boldsymbol{u}^{\text{ref}} - \nabla \boldsymbol{u}\right) \, d\Omega}{\int_\Omega \left(\nabla \boldsymbol{u}^{\text{ref}}\right)^\top \boldsymbol{D} \left(\nabla \boldsymbol{u}^{\text{ref}}\right) \, d\Omega}\right]^{0.5} \tag{6.14}$$

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

6.2.1 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. 6.6a, 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. 6.6b, resulting in 54 elements. Positions of the new re-entrant nodes (nodes 9 and 10) are given as

$$x_3^{(9)} = x_3^{(1)}, \qquad x_i^{(9)} = x_i^{(1)} + (0.5 - d)\, s_i, \quad i = 1,2 \tag{6.15a}$$

$$x_3^{(10)} = x_3^{(5)}, \qquad x_i^{(10)} = x_i^{(5)} + (0.6 - d)\, s_i, \quad i = 1,2 \tag{6.15b}$$

   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. 6.6a), using the same parameter d, as follows:

$$\boldsymbol{x}_d^B = \boldsymbol{x}^{(B)} - d \times [4.2s_1 \quad 1.75s_2 \quad 0.7s_3]^\intercal. \tag{6.16a}$$

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. 6.6c.

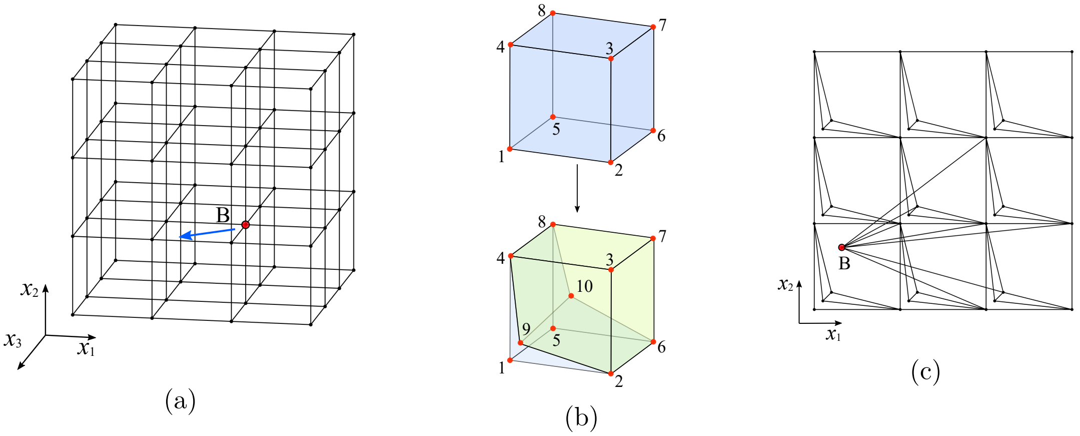


Figure 6.6: (a) Regular mesh (with 27 elements). (b) Non-tangled element converted to tangled element (c) Front view of the tangled mesh.

The tangled mesh (Fig. 6.6c) has 54 elements; out of which 31 elements are tangled. Fig. 6.7 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. 6.7b, 6.7c, 6.7e) exhibit penetration. Since the mesh (Fig. 6.6c) contains tangled elements of various configurations, it can be employed to rigorously test the effectiveness of a-TFEM.

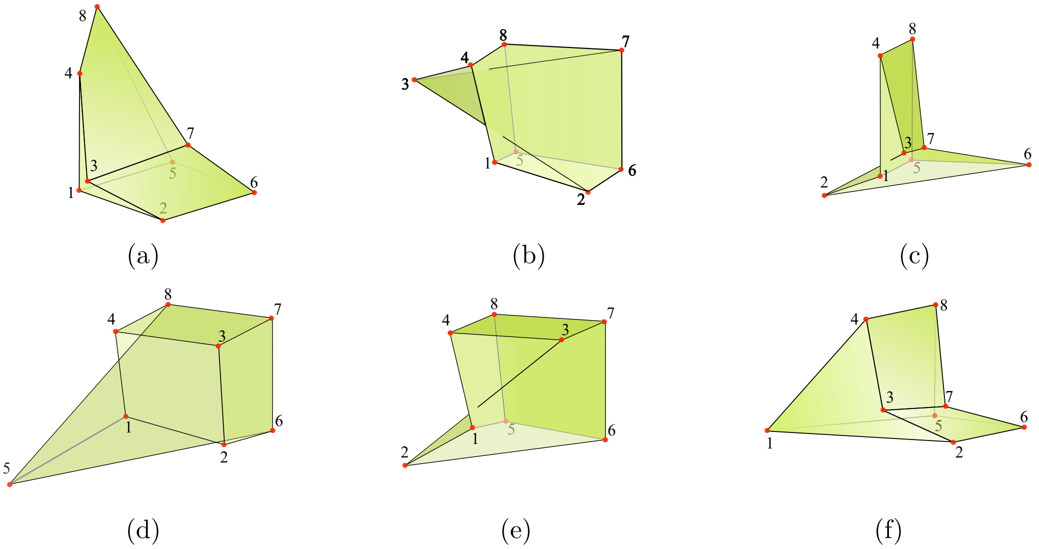


Figure 6.7: Some of the tangled elements present in the mesh shown in Fig. 6.6c.

Displacement patch test

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

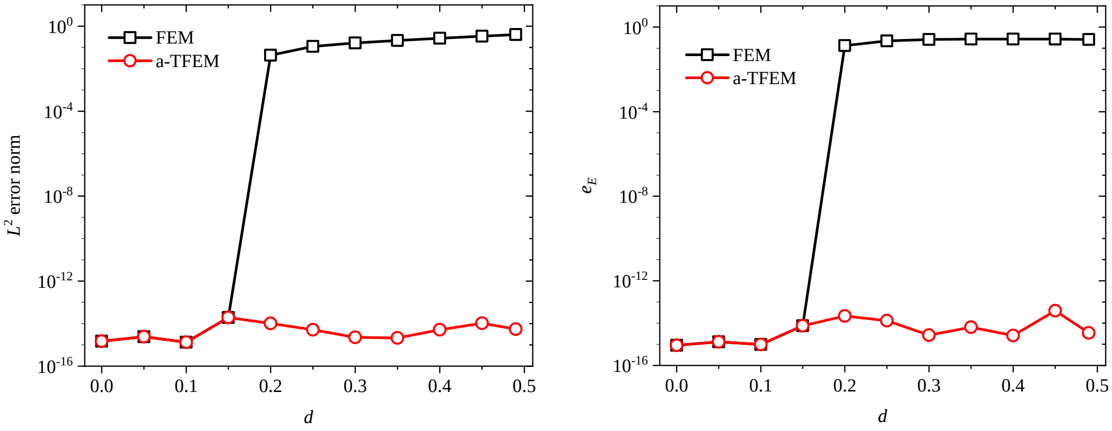
$$u_1 = 0.579x_1 + 0.246x_2 + 0.482x_3 - 0.374 \tag{6.17a}$$

$$u_2 = 0.486x_1 + 0.351x_2 + 0.947x_3 - 0.620 \tag{6.17b}$$

$$u_3 = 0.512x_1 + 0.746x_2 + 0.548x_3 - 0.480. \tag{6.17c}$$

Corresponding to this field, Dirichlet conditions are applied to the left face of the mesh in Fig.6.6c, while Neumann condition are applied to the remaining faces. The problem is solved over the tangled meshes by varying d using standard FEM and a-TFEM (the non-tangled mesh in Fig. 6.6a is not used for this experiment).

To compare the two methods, the L^2 and energy error norms are plotted in Fig. 6.8a and Fig. 6.8b respectively. Observe that, both FEM and a-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 a-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, a-TFEM consistently achieves machine precision accuracy for all values of d. In other words, a-TFEM successfully passes this patch test over the tangled mesh, while standard FEM does not.



(a) (b)

Figure 6.8: Patch test for FEM and a-TFEM: (a) L^2 error norm and (b) energy error norm.

Strain energy convergence: Cubic cantilever

In this experiment, we analyze a-TFEM for its convergence characteristic with regard to the strain energy. We consider a cubic cantilever (see Fig. 6.9) subjected to a uniform pressure $p = 1$ on its upper face [116], 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 [116], 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).

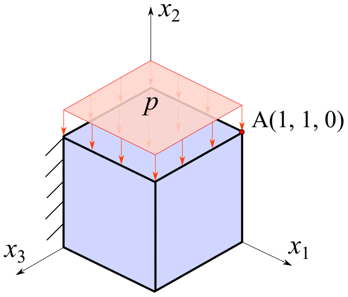
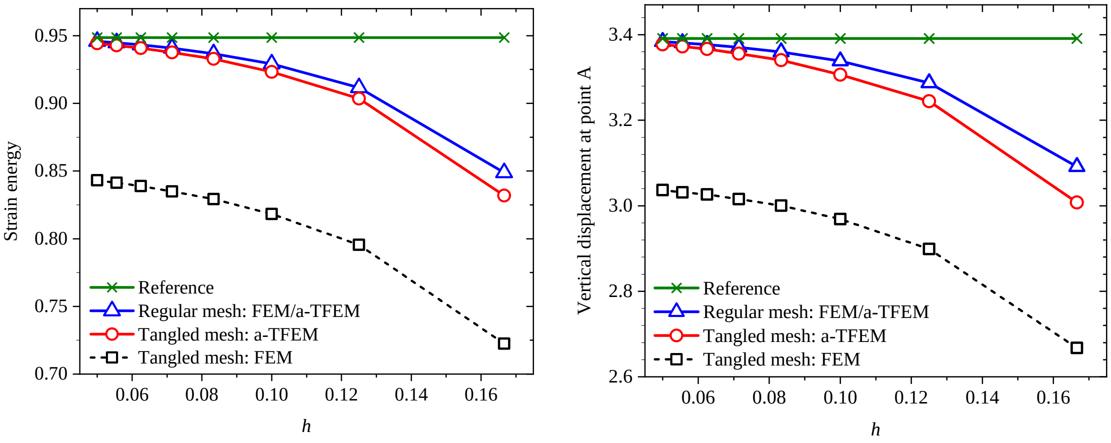


Figure 6.9: Cubic cantilever subjected to uniform pressure.

We solve this problem using both standard FEM and a-TFEM on tangled and non-tangled meshes of various sizes. Regular and tangled meshes are constructed by repeating the units described previously (Fig. 6.6a for regular mesh and Fig. 6.6c 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. 6.10a shows the strain energy convergence for a-TFEM in comparison with FEM over tangled and regular (non-tangled) mesh. It is evident that the strain energy obtained using a-TFEM converges to the reference solution, unlike FEM over the tangled mesh.

Similarly, Fig. 6.10b illustrates the convergence of the tip deflection at point A(1, 1, 0) obtained using the a-TFEM and FEM. Once again, observe that a-TFEM converges to the reference solution for the tangled meshes, while FEM fails to do so.

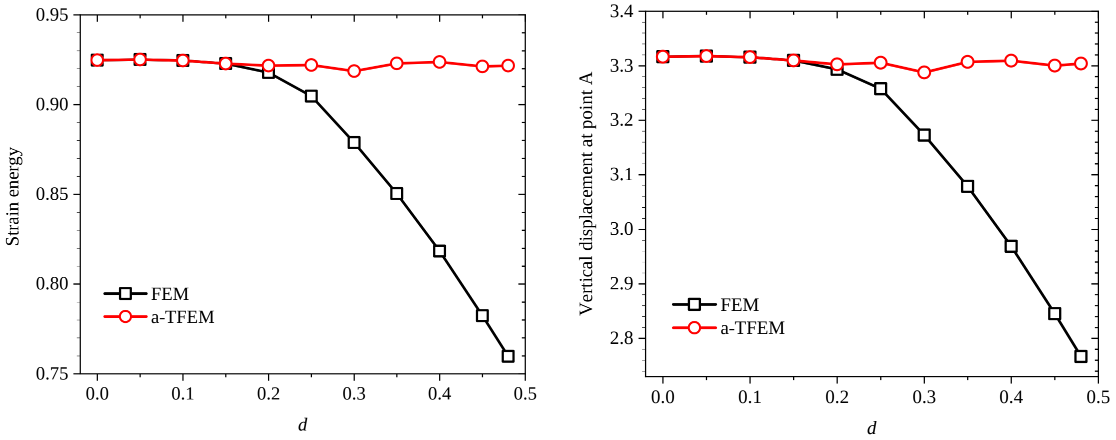


(a)

(b)

Figure 6.10: 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 a-TFEM (see Fig. 6.11).



(a) (b)

Figure 6.11: Effect of tangling on (a) strain energy and (b) vertical displacement at point A for the cubic cantilever problem.

Convergence study: Beam under torsion

Next, we study the performance of a-TFEM for a cantilever beam loaded in torsion [53]. 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 [8]:

$$
\begin{aligned}
\sigma_{11} &= \sigma_{22} = \sigma_{12} = \sigma_{33} = 0 \\
\sigma_{31} &= \frac{8E\beta}{\pi^2(1+\nu)} \sum_{n=1}^{\infty} \frac{(-1)^n}{(2n-1)^2 \cosh\left[(2n-1)\pi/2\right]} \cos\left[(2n-1)\pi x_1/2\right] \sinh\left[(2n-1)\pi x_2/2\right] \\
\sigma_{(23)} &= \frac{E\beta}{2(1+\nu)}(2x_1) + \\
&\quad \frac{E\beta}{2(1+\nu)} \left(\sum_{n=1}^{\infty} \frac{16(-1)^n}{\pi^2 (2n-1)^2 \cosh\left[(2n-1)\pi/2\right]} \sin\left[(2n-1)\pi x_1/2\right] \cosh\left[(2n-1)\pi x_2/2\right] \right)
\end{aligned}
\tag{6.18}
$$

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

$$\begin{aligned}
&u_1 = -\beta x_2 x_3, \quad u_2 = \beta x_3 x_1, \\
&u_3 = \beta \left(x_1 x_2 + \sum_{n=1}^{\infty} \frac{32(-1)^n}{\pi^3 (2n-1)^3 \cosh\left[(2n-1)\pi/2\right]} \sin\left[(2n-1)\pi x_1/2\right] \sinh\left[(2n-1)\pi x_2/2\right] \right).
\end{aligned} \tag{6.19}$$

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 a-TFEM results for the tangled mesh with $n_r = 4$ in Fig. 6.12.

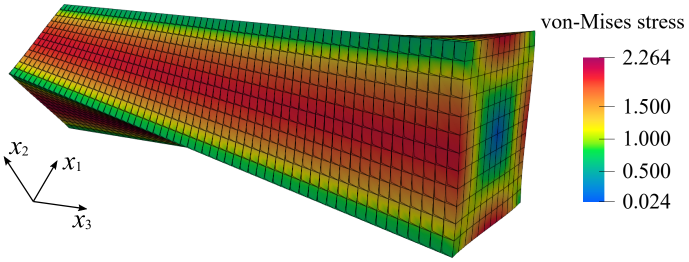
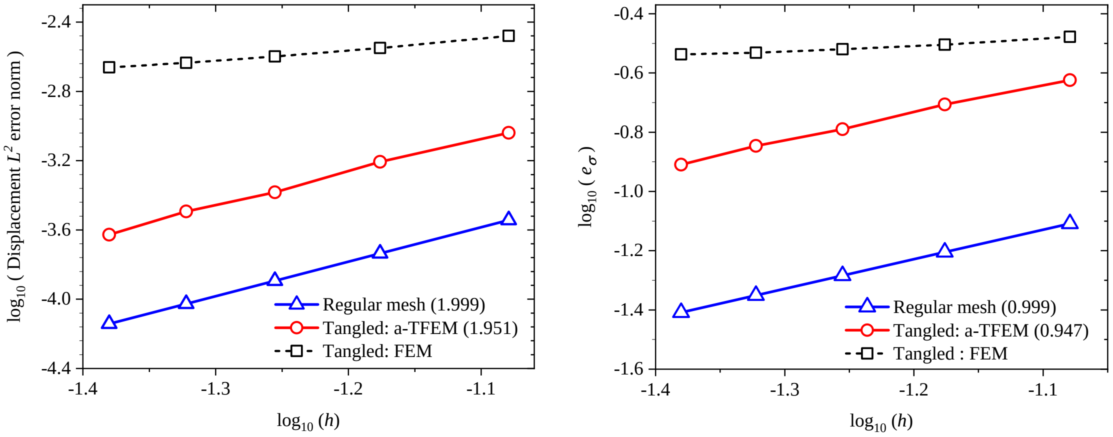


Figure 6.12: Deformed configuration of the beam under torsion.

We study the convergence characteristics of a-TFEM by employing displacement L^2 error norm defined in Eq. 6.13. In addition, an error measure for the stress field ($e_{\boldsymbol{\sigma}}$) is employed and defined as (to be consistent with [53]):

$$e_{\boldsymbol{\sigma}} = \frac{\|\boldsymbol{\sigma}^{\text{ref}} - \boldsymbol{\sigma}\|_{L^2(\Omega)}}{\|\boldsymbol{\sigma}^{\text{ref}}\|_{L^2(\Omega)}}. \tag{6.20}$$

As evident from the convergence plots in Fig. 6.13, 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 6.13: Convergence of (a) displacement error and (b) stress error for the torsion problem.

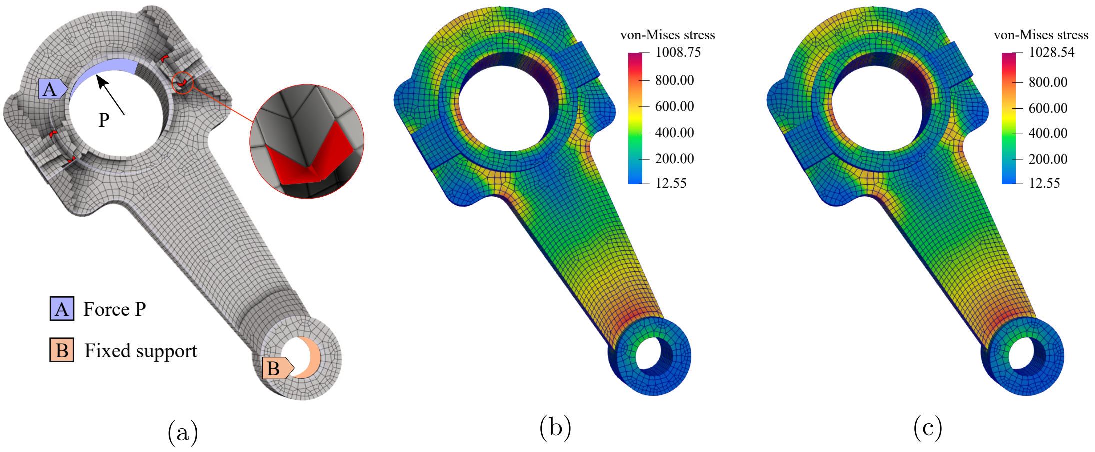
6.2.2 Real-world Meshes

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

Connecting rod

We now consider a connecting rod mesh (mesh data obtained from [103]) illustrated in Fig. 6.14a. Out of the total 11316 hexahedral elements of the mesh, 16 elements are tangled; some are highlighted in Fig. 6.14a. Though it is possible to untangle this mesh [103], a-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. 6.14a: 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 6.14: (a) Connecting rod mesh [103] with boundary conditions; elements in red color are tangled. (b) von-Mises stress field for tangled mesh using a-TFEM; (b) von-Mises stress field for untangled mesh.

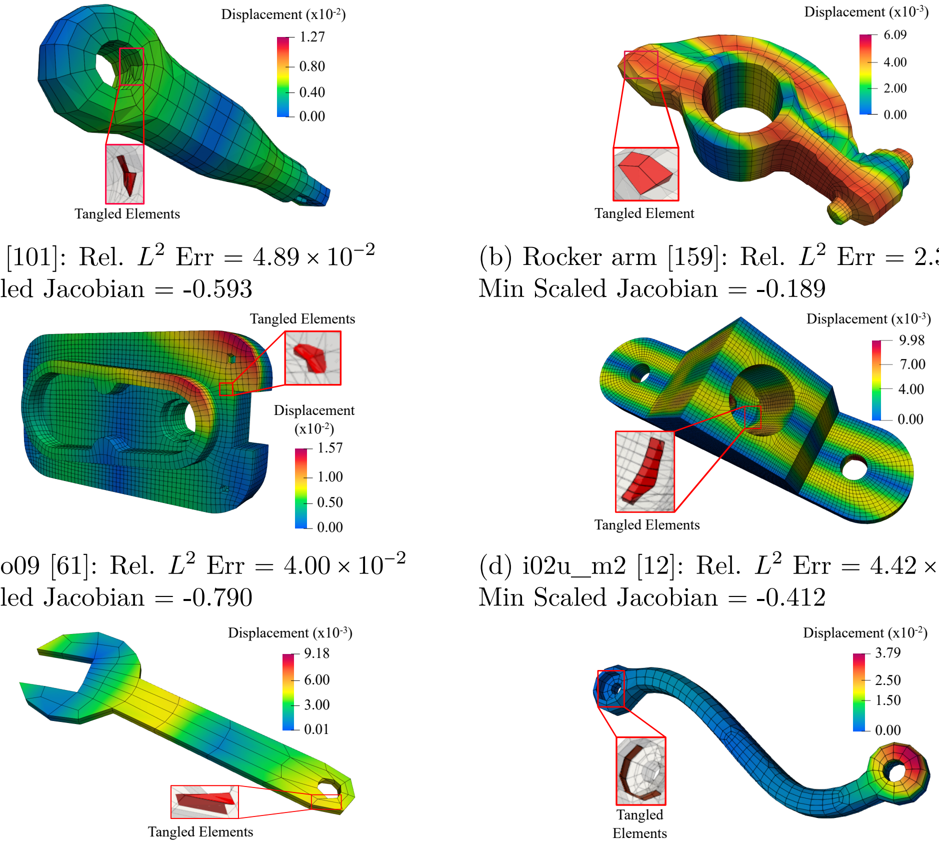
Fig. 6.14b and Fig. 6.14c illustrate the von-Mises stress field for the tangled mesh (using a-TFEM) and untangled mesh respectively. The total computational time was 10.47s for the tangled mesh using a-TFEM, and 10.16 for untangled mesh, i.e., the overhead due to a-TFEM was found to be minimal.

Synthetic solution

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

$$\boldsymbol{u} = \left[\zeta_1^3\zeta_2\zeta_3^2 + 2\zeta_1^2\zeta_2^3\zeta_3^2 + 0.5\sin(2\pi\zeta_1); \quad \zeta_1^2\zeta_2^3\zeta_3 + 2\zeta_1^2\zeta_2^2\zeta_3^3; \quad \zeta_1\zeta_2^2\zeta_3^3 + 2\zeta_1^3\zeta_2^2\zeta_3^2\right]^{\mathsf{T}}/100 \tag{6.21}$$

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. 6.15 for various tangled meshes obtained from [101, 159, 102, 12] and hexalab repository [22]. 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' [85] computed at the nodes is used, and is therefore reported in Fig. 6.15 as well.



(a) Rod [101]: Rel. L^2 Err $= 4.89 \times 10^{-2}$
Min Scaled Jacobian = -0.593

(b) Rocker arm [159]: Rel. L^2 Err $= 2.37 \times 10^{-2}$
Min Scaled Jacobian = -0.189

(c) Demo09 [61]: Rel. L^2 Err $= 4.00 \times 10^{-2}$
Min Scaled Jacobian = -0.790

(d) i02u_m2 [12]: Rel. L^2 Err $= 4.42 \times 10^{-1}$
Min Scaled Jacobian = -0.412

(e) wrench [101]: Rel. L^2 Err $= 9.95 \times 10^{-2}$
Min Scaled Jacobian = -0.280

(f) lever arm [101]: Rel. L^2 Err $= 6.31 \times 10^{-2}$
Min Scaled Jacobian = -0.944

Figure 6.15: Displacement plots obtained via a-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 [103]. The problem with the synthetic solution (Eq. 6.21) described above is solved using the two methods: a-TFEM for tangled meshes and standard FEM for untangled meshes. The results, presented in Table 6.1, reveal that the L^2 error norm obtained via a-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 a-TFEM can provide solutions with comparable accuracy over tangled meshes, thus eliminating the need for mesh untangling.

Table 6.1: Comparison of solutions over tangled and untangled meshes provided in [103].

Model	Min. Scaled Jacobian		Relative L^2 error		Time (seconds)	
	Tangled	Untangled	Tangled	Untangled	Tangled	Untangled
cap	-0.94	0.11	4.23×10^{-2}	4.36×10^{-2}	4.73	4.13
block	-0.70	0.25	1.58×10^{-2}	1.58×10^{-2}	2.02	1.87
bust	-0.60	0.11	3.71×10^{-2}	3.93×10^{-2}	5.15	5.11
linking rod	-0.39	0.55	2.13×10^{-3}	2.12×10^{-3}	10.46	10.17

Chapter 7

Nonlinear Elasticity using Isoparametric-TFEM and Accelerated-TFEM

In this chapter, i-TFEM and a-TFEM formulation is derived for solving nonlinear elasticity problems.

7.1 Nonlinear Elasticity

Consider a body occupying a domain $\Omega \in \mathbb{R}^2$ subject to a body force $\boldsymbol{b}$, traction $\boldsymbol{T}$ over the boundary $\partial\Omega^T$, and Dirichlet boundary conditions $\boldsymbol{u} = \boldsymbol{u_d}$ over the boundary $\partial\Omega^d$; the material is assumed to be hyper-elastic undergoing a finite deformation $\boldsymbol{u}$. The domain is divided into M elements E_j, identified by the set $I = \{1, \ldots M\}$. We employ the total Lagrangian formulation [141] in this chapter, where the potential energy can be written as:

$$\Pi(\boldsymbol{u}) = \sum_{j\in I} \int_{E_j} \Psi\left(\boldsymbol{F}\left(\boldsymbol{u}_j\right)\right) dV - \sum_{j\in I} \int_{E_j} \boldsymbol{u}_j \cdot \boldsymbol{b} dV - \sum_{j\in I} \int_{\partial E_j^T} \boldsymbol{u}_j \cdot \boldsymbol{T} dS \tag{7.1}$$

where $\boldsymbol{F}$ is the deformation gradient, Ψ is the strain energy density. Further, using the standard (Bubnov-) Galerkin variational formulation, one arrives at the residual equation

[141]:

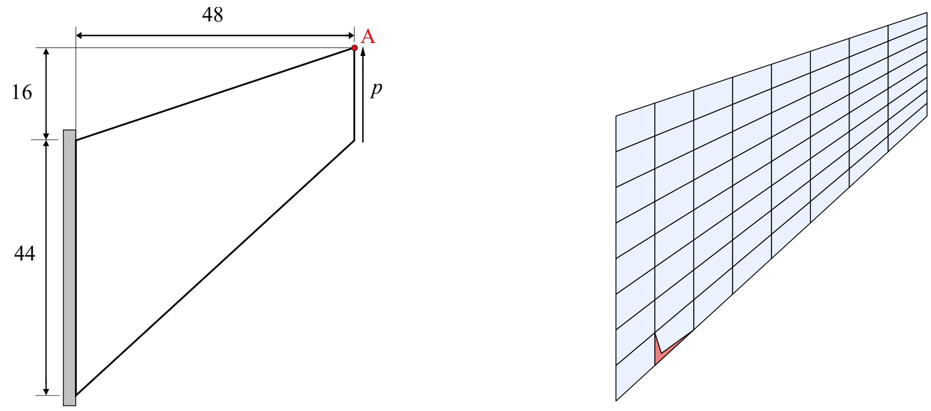
$$\boldsymbol{R}(\hat{\boldsymbol{u}}) = 0 \tag{7.2}$$

This is typically solved iteratively via the Newton-Raphson algorithm [141]:

$$\boldsymbol{K}(\hat{\boldsymbol{u}}^n)\Delta\hat{\boldsymbol{u}}^{n+1} = -\boldsymbol{R}(\hat{\boldsymbol{u}}^n). \tag{7.3}$$

where $\boldsymbol{K}$ is the tangent matrix and $\Delta\hat{\boldsymbol{u}}^n$ is the incremental displacement vector at n^{th} Newton iteration. When the mesh is of high-quality and not tangled, one obtains accurate solutions to such problems.

However, as is well known, when the mesh is tangled, i.e., if the mesh contains inverted elements, the solution becomes erroneous. To illustrate, consider Cook's membrane problem [37] illustrated in Fig. 7.1a. The left edge of the tapered cantilever is fixed while a uniformly distributed load $p = 5$ is applied on the right edge. We pose a geometrically nonlinear plane-strain problem with Lamé parameters $\mu = 50$ and $\lambda = 100$. Fig. 7.1b illustrates a quadrilateral mesh with one concave element that we use for this experiment.



(a)

(b)

Figure 7.1: (a) Cook's membrane problem. (b) Tangled mesh with one concave element.

We vary the extent of tangling by moving the re-entrant vertex D along the diagonal BC as shown in Fig. 7.2a. When the parameter $d = 0$, the point D lies half-way between B and C, and when $0 < d < 0.5$, the point D moves towards B, i.e., the element gets tangled. The large-deformation problem is solved using the normal procedure as described above, with 10 load steps. The tip displacement is compared against the expected value (using a high quality non-tangled mesh). When $d > 0.1$, a negative $|\boldsymbol{J}|$ value is encountered at one or more Gauss points, and Fig. 7.2b illustrates the resulting erroneous solution.

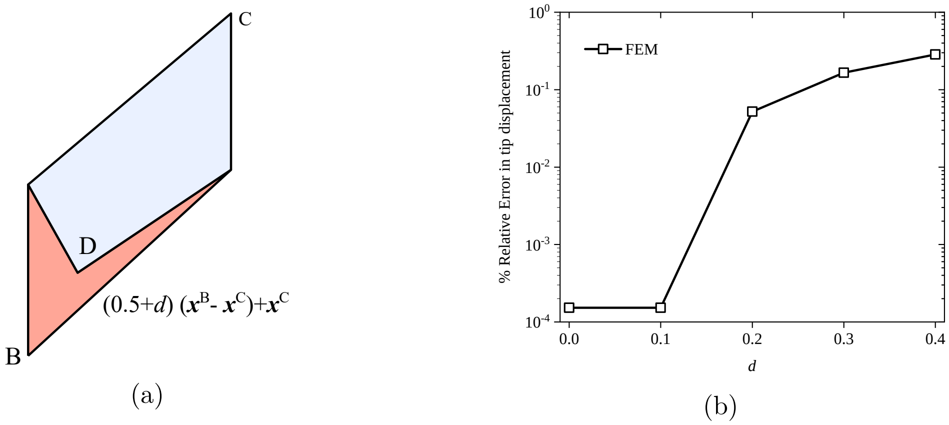


Figure 7.2: (a) Zoomed-in view of the tangled element. (b) Relative error in tip displacement versus d for FEM.

7.2 Isoparametric TFEM for Nonlinear Elasticity

The objective of this section is to propose an iso-parametric tangled finite element method (i-TFEM), as a simple extension to classic FEM, for solving large deformation problems over tangled meshes. As a background, we briefly review the critical i-TFEM concepts proposed in [129] for linear problems.

7.2.1 Revisiting Isoparametric-TFEM

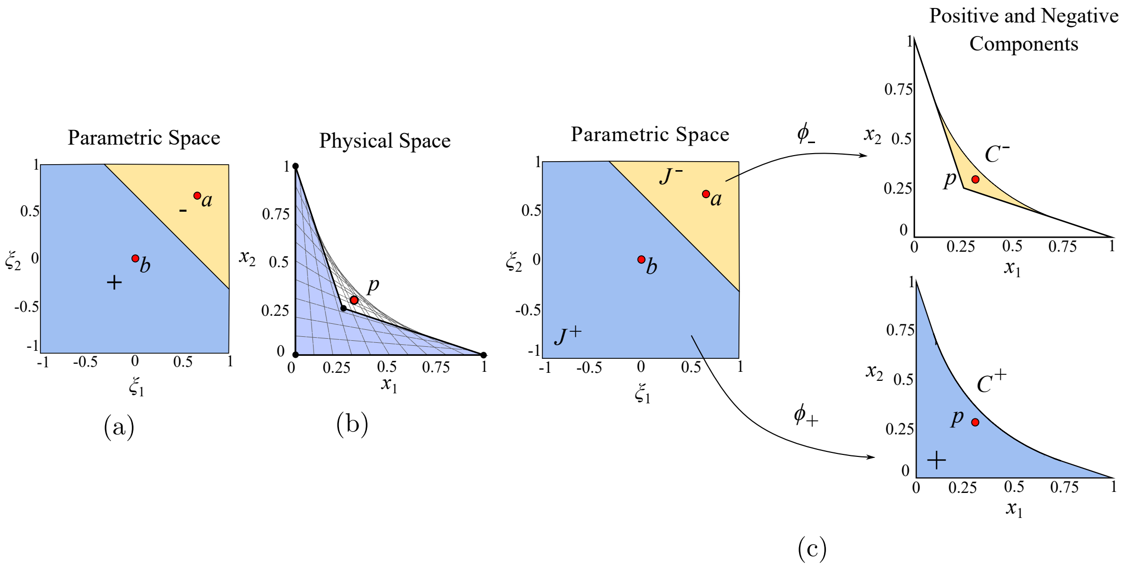
Consider the standard isoparametric mapping from (ξ_1, ξ_2) space in Fig. 7.3a to a concave element in the physical space (x_1, x_2) in Fig. 7.3b. Observe that the element folds onto itself. Further, the parametric space can be divided into positive (J^+) and negative (J^-) Jacobian regions and the parametric mapping ϕ is *not fully invertible*.

The main idea in i-TFEM is that the positive and negative parametric regions are treated separately, thus relaxing the constraint of full invertibility to piecewise invertibility. In particular, the physical space corresponding to the positive (negative) parametric region

J^{+} (J^{-}) is termed as positive (negative) component and is denoted by C^{+} (C^{-}). Observe the piecewise mapping

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

is invertible i.e. bijective (see Fig. 7.3c).



(a)

(b)

(c)

Figure 7.3: (a) Parametric space of the concave element. Parametric space can be divided into positive and negative Jacobian regions. (b) Physical space of the concave Q9 element. (c) Positive and negative components

Let $\boldsymbol{N}_j(\boldsymbol{\xi})$ be the standard bilinear Lagrange shape functions defined over the parametric space of element E_j. Let $\boldsymbol{N}_j^{\pm}$ be the restriction of $\boldsymbol{N}_j$ to $J^{\pm}$, i.e.,

$$\boldsymbol{N}_j^{\pm}(\boldsymbol{x}) \coloneqq \boldsymbol{N}_j(\phi_{\pm}^{-1}(\boldsymbol{x})) \tag{7.4}$$

The corresponding field is then given by

$$\boldsymbol{u}_j^{\pm}(\boldsymbol{x}) = \boldsymbol{N}_j^{\pm}(\boldsymbol{x})\hat{\boldsymbol{u}}_j \tag{7.5}$$

Now consider the two-element tangled patch in Fig. 7.4a. The positive and negative components (C_1^{+} and C_1^{-}) of the concave element E_1 are shown in Fig. 7.4b. On the other hand, the convex element E_2 has only one positive component (see Fig. 7.4c): $E_2 = C_2^{+}$

while $C_2^- = \varnothing$. Further, the fold F_1 illustrated in Fig. 7.4d overlaps with E_2 as well.

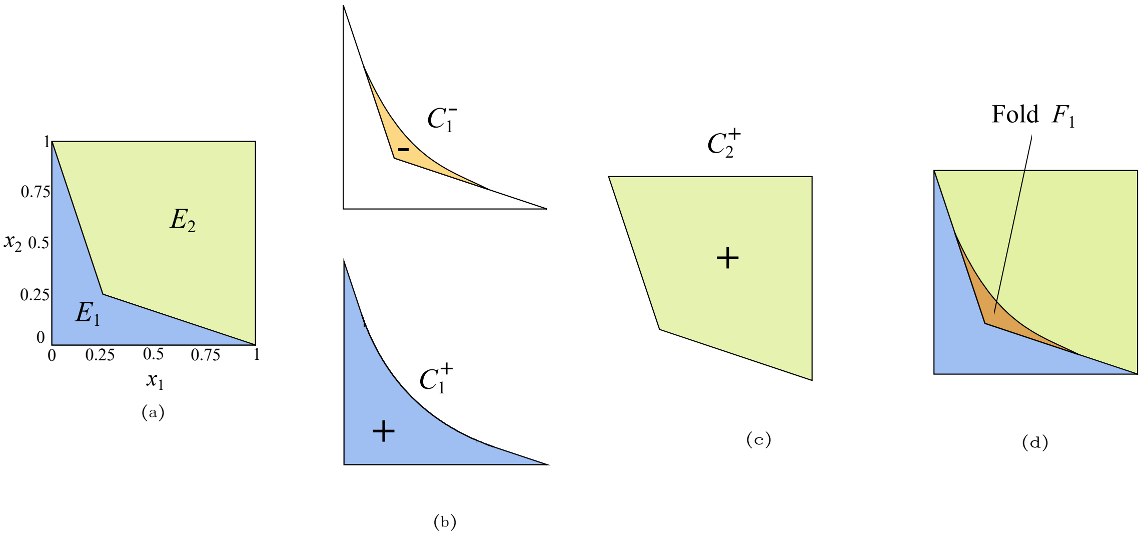


Figure 7.4: (a) 2-D domain discretized into two bilinear quads. (b) Positive and negative components of the concave element. (c) Convex element of the mesh. (d) Final physical space has an overlapping region (fold).

Thus, for any point $\boldsymbol{x} \in F_1$, all the three components overlap, and three fields $\boldsymbol{u}_1^+(\boldsymbol{x})$, $\boldsymbol{u}_1^-(\boldsymbol{x})$ and $\boldsymbol{u}_2^+(\boldsymbol{x})$ can be defined.

Thus, the field is clearly ambiguous within the fold. Removing the ambiguity in the field definition is the first step in i-TFEM [130, 129, 42]. In particular, in i-TFEM, we *define* the field at a point $\boldsymbol{x}$ within a fold as:

$$\boldsymbol{u}(\boldsymbol{x}) \coloneqq \boldsymbol{u}_2^+(\boldsymbol{x}), \quad \forall \boldsymbol{x} \in F_1 \tag{7.6}$$

The underlying reasons are discussed in [130], but briefly, this is necessary for field continuity and to capture constant strain fields. In other words, for iso-parametric elements, the tangled region can be considered as being part of just the convex element E_2. This naturally leads to a division of the mesh into two parts: E_2 and $\widehat{E}_1$ as illustrated in Fig. 7.5. Note that $\widehat{E}_1$ does not include the folded region whereas E_1 does (this is elaborated further below).

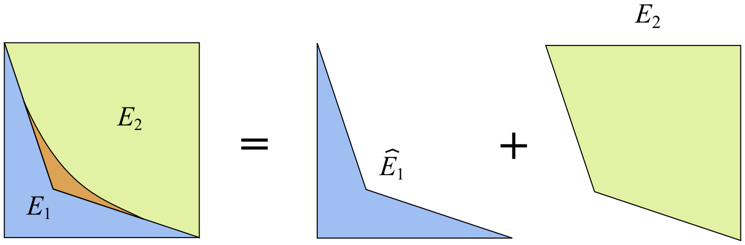


Figure 7.5: Parts contributing to the field definition.

Thus, the field $\boldsymbol{u}$ over the two-element patch is defined as:

$$\boldsymbol{u}(\boldsymbol{x})|_{\widehat{E}_1} = \boldsymbol{N}_1^+(\boldsymbol{x})\hat{\boldsymbol{u}}_1 \quad \text{and} \quad \boldsymbol{u}(\boldsymbol{x})|_{E_2} = \boldsymbol{N}_2^+(\boldsymbol{x})\hat{\boldsymbol{u}}_2 \tag{7.7}$$

However, an additional constraint is needed to ensure the continuity of the field across their common boundary. In particular, by approaching the re-entrant corner, from $\widehat{E}_j$ and E_2, one can show that field continuity across the entire boundary is satisfied if and only if (see [130]):

$$\boldsymbol{u}_1^+(\boldsymbol{x}) - \boldsymbol{u}_1^-(\boldsymbol{x}) = 0, \quad \boldsymbol{x} \in F_1 \tag{7.8}$$

This not only makes the field continuous, it also and forces the contribution of the concave element to be zero in the folded region.

Introducing the notation $[\![\cdot]\!] = (\cdot)^+ - (\cdot)^-$, the above constraint can be written as:

$$[\![u_1]\!] = 0, \quad \text{in } F_1 \tag{7.9}$$

In summary, for any pair of overlapping elements E_1 and E_2 (1) we decompose them into fully invertible regions $\widehat{E}_1$ and E_2, and (2) the constraint (Eq. 7.9) is enforced.

7.2.2 i-TFEM Variational Formulation for Nonlinear Elasticity

We now consider the implications of these concepts in non-linear elasticity. Our objective is to generalize the residual in Eq. 7.2 and the iteration in Eq. 7.3, to account for tangling.

Towards this end, we modify the potential energy functional as follows:

$$\widetilde{\Pi} = \sum_{j \in I_{\text{convex}}} \int\limits_{E_j} \left(\Psi\left(\boldsymbol{u}_j\right) - \boldsymbol{u}_j \cdot \boldsymbol{b}\right) dV + \sum_{j \in I_{\text{concave}}} \int\limits_{\widehat{E}_j} \left(\Psi\left(\boldsymbol{u}_j\right) - \boldsymbol{u}_j \cdot \boldsymbol{b}\right) dV - \sum_{j \in I} \int\limits_{\partial E_j^T} \boldsymbol{u}_j \cdot \boldsymbol{T} dS \\ + \sum_{j \in I_{\text{concave}}} \int\limits_{F_j} \boldsymbol{\lambda}_j \cdot [\![\boldsymbol{u}_j]\!] dV \tag{7.10}$$

where the concave and convex elements are indexed as I_{concave} and I_{convex} respectively, and the constraints in Eq. 7.9 are included via Lagrange multipliers $\boldsymbol{\lambda}$.

We now set the variation of the potential energy with respect to $\boldsymbol{u}$ and $\boldsymbol{\lambda}$ to zero:

$$\delta_{\boldsymbol{u},\boldsymbol{\lambda}} \widetilde{\Pi} = 0. \tag{7.11}$$

This leads to the following weak form where $\forall \delta \boldsymbol{u} \in H_0^1$ and $\forall \delta \boldsymbol{\lambda} \in L^2$, i.e., we seek $\boldsymbol{u} \in H^1$ and $\boldsymbol{\lambda} \in L^2$ such that:

$$\left[\sum_{j \in I_{\text{convex}}} \int\limits_{E_j} \left(\boldsymbol{P} : \frac{\partial \delta \boldsymbol{u}_j}{\partial \boldsymbol{X}} - \delta \boldsymbol{u}_j \cdot \boldsymbol{b}\right) dV + \sum_{j \in I_{\text{concave}}} \int\limits_{\widehat{E}_j} \left(\boldsymbol{P} : \frac{\partial \delta \boldsymbol{u}_j}{\partial \boldsymbol{X}} - \delta \boldsymbol{u}_j \cdot \boldsymbol{b}\right) dV - \sum_{j \in I} \int\limits_{\partial E_j^T} \delta \boldsymbol{u}_j \cdot \boldsymbol{T} dS \right. \\ \left. + \sum_{j \in I} \int\limits_{F_j} \boldsymbol{\lambda}_j \cdot [\![\delta \boldsymbol{u}_j]\!] dV \right] + \int\limits_{F_j} \delta \boldsymbol{\lambda}_j \cdot [\![\boldsymbol{u}_j]\!] dV = 0 \tag{7.12}$$

where $\boldsymbol{P}$ is the first Piola-Kirchhoff stress tensor. Next, we approximate the fields using the standard (Bubnov-) Galerkin formulation:

$$\boldsymbol{u}_j = \boldsymbol{N}_j \hat{\boldsymbol{u}}_j, \quad \boldsymbol{\lambda}_j = \boldsymbol{N}_j^{\lambda} \hat{\boldsymbol{\lambda}}_j \tag{7.13}$$

This leads to:

$$\delta \hat{\boldsymbol{u}}^{\top} \widetilde{\boldsymbol{R}} + \delta \hat{\boldsymbol{\lambda}}^{\top} \boldsymbol{C}^{\top} \hat{\boldsymbol{u}} = 0 \tag{7.14}$$

Here, $\delta \hat{\boldsymbol{u}}^{\top} \widetilde{\boldsymbol{R}}$ represents the terms in the square bracket of Eq. 7.12. Observe from Eq.

7.12, that the residual $\widetilde{\boldsymbol{R}}$ can be expressed as:

$$\widetilde{\boldsymbol{R}}\left(\hat{\boldsymbol{u}},\hat{\boldsymbol{\lambda}}\right) = \boldsymbol{R}^{u}\left(\hat{\boldsymbol{u}}\right) + \boldsymbol{C}\hat{\boldsymbol{\lambda}} = \mathbf{0} \tag{7.15}$$

where $\boldsymbol{R}^{u}$ involves only the primary field $\boldsymbol{u}$ and requires integrating over both the convex and concave elements:

$$\boldsymbol{R}^{u} = \boldsymbol{R}^{u}_{convex} + \widehat{\boldsymbol{R}}^{u}_{concave} \tag{7.16}$$

As one can easily deduce, the computation of $\boldsymbol{R}^{u}_{convex}$ is as in standard FEM. However, the integration over the concave elements must be carried out over the subset of parametric space; see Fig. 7.6. This is discussed in detail in the next subsection.

Next, to solve Eq. 7.15 through iterations, we consider the first order Taylor series:

$$\boldsymbol{R}^{u}\left(\hat{\boldsymbol{u}}^{n}\right) + \left.\frac{\partial \boldsymbol{R}^{u}}{\partial \hat{\boldsymbol{u}}}\right|_{n} \Delta\hat{\boldsymbol{u}}^{n+1} + \boldsymbol{C}\hat{\boldsymbol{\lambda}}^{n} + \boldsymbol{C}\Delta\hat{\boldsymbol{\lambda}}^{n+1} = \mathbf{0} \tag{7.17}$$

i.e.,

$$\boldsymbol{K}^{t}\Delta\hat{\boldsymbol{u}} + \boldsymbol{C}\Delta\hat{\boldsymbol{\lambda}} = -\left(\boldsymbol{R}^{u} + \boldsymbol{C}\hat{\boldsymbol{\lambda}}\right) \tag{7.18}$$

where

$$\boldsymbol{K}^{t} = \boldsymbol{K}^{t}_{convex} + \widehat{\boldsymbol{K}}^{t}_{concave} \tag{7.19}$$

Here, $\boldsymbol{K}^{t}_{convex}$ and $\widehat{\boldsymbol{K}}^{t}_{concave}$ are tangent matrices corresponding to convex and concave elements respectively. Further, from Eq. 7.14, we have

$$\boldsymbol{C}^{\top}\Delta\hat{\boldsymbol{u}} = \mathbf{0} \tag{7.20}$$

From Eq. 7.18 and Eq. 7.20 we have the final set of linear equations one must solve iteratively:

$$\begin{bmatrix} \boldsymbol{K}^{t} & \boldsymbol{C} \\ \boldsymbol{C}^{\top} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \Delta\hat{\boldsymbol{u}}^{n+1} \\ \Delta\hat{\boldsymbol{\lambda}}^{n+1} \end{Bmatrix} = \begin{Bmatrix} -\left(\boldsymbol{R}^{u} + \boldsymbol{C}\hat{\boldsymbol{\lambda}}\right) \\ \mathbf{0} \end{Bmatrix}. \tag{7.21}$$

If the mesh does not contain any tangled elements, then $\boldsymbol{K}^{t} = \boldsymbol{K}^{t}_{convex}$, and $\boldsymbol{C}$ does not exist, i.e., i-TFEM reduces to standard FEM.

7.2.3 Implementation Details

We now discuss the implementation considering a two-element mesh.

Computation of residual vector and stiffness matrix

As mentioned earlier, to compute $\boldsymbol{R}^u_{convex}$ in Eq. 7.16, standard FEM procedures with Gauss quadrature can be used. However, to compute $\boldsymbol{R}^u_{concave}$, only the fully invertible subset $\widehat{E}_1$ is to be considered.

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 the L-shaped subset of the positive $|\boldsymbol{J}|$ region as illustrated in Fig. 7.6. In other words, $\widehat{E}_1$ corresponds to a *fully invertible* subset of the parametric space of the concave element.

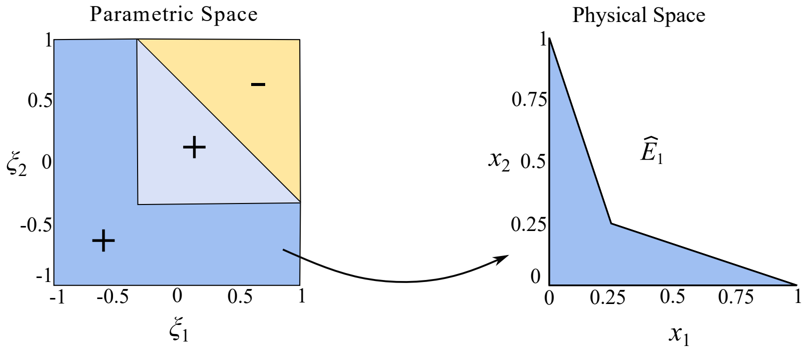


Figure 7.6: (a) Parametric space. (b) $\widehat{E}_1$

Therefore, standard Gauss quadrature cannot be employed; instead, $\widehat{E}_1$ is triangulated as illustrated in Fig. 7.7. The triangulation is used merely for the purpose of integration and does not lead to additional degrees of freedom in i-TFEM.

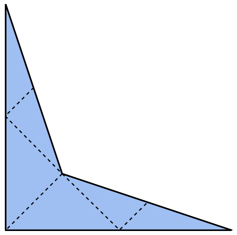


Figure 7.7: Triangulation of $\widehat{E}_1$

Similarly, to compute $\boldsymbol{K}^t_{convex}$, standard FEM procedures can be used. However, to compute $\boldsymbol{K}^t_{concave}$, the triangulation in Fig. 7.7 must be used.

Constraint enforcement

Finally, to compute the constraint matrix $\boldsymbol{C}$, note that the interpolation of the Lagrange multiplier $\boldsymbol{\lambda}$ needs to be only square integrable since its gradient does not appear in the formulation. For Q4 elements, the primary field $\boldsymbol{u}$ is approximated using standard bilinear functions while $\boldsymbol{N}^{\lambda}$ are constant functions. Thus, the finite dimensional approximation for the Lagrange multiplier $\boldsymbol{\lambda}$ comes from an FE space that is smaller than that for $\boldsymbol{u}$. Accordingly, we can write the constraint matrix defined in Equation 7.15 as

$$\boldsymbol{C} = \int\limits_{F_1} \left(\boldsymbol{N}_1^{+} - \boldsymbol{N}_1^{-}\right)^{\top} dV. \tag{7.22}$$

Direct integration over the tangled region F_1 to compute $\boldsymbol{C}$ is computationally expensive and cumbersome [129]. Instead, we evaluate the integrand at a sample point $\boldsymbol{x} \in F_1$, say the concave vertex, i.e., evaluate the $\boldsymbol{C}$ as

$$\boldsymbol{C} = \left(\boldsymbol{N}_1^{+}(\boldsymbol{p}) - \boldsymbol{N}_1^{-}(\boldsymbol{p})\right)^{\top} = [\![\boldsymbol{N}_1(\boldsymbol{p})]\!]\,. \tag{7.23}$$

where $\boldsymbol{p}$ is the re-entrant vertex. Thus, the constraints can be applied directly as a set of algebraic equations. Since $\boldsymbol{u}$ is a vector field, each concave element entails two constraint equations. This is consistent, for example, with the algebraic constraints implemented in [129, 135, 130].

7.3 Numerical Experiments

In this section, i-TFEM is demonstrated using plane strain nonlinear elasticity problems over various tangled meshes. Numerical experiments are conducted under the following conditions:

- 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 number of quadrature points for convex quadrilateral elements is four.
- The triangulation of a concave element (see Fig. 7.7) is performed by employing MATLAB's inbuilt mesher - `generateMesh`. The number of quadrature points for triangles is 4.
- The load is applied incrementally in 10 steps. The stopping criteria for Newton Raphson is $||\Delta\hat{\boldsymbol{u}}|| < 10^{-9}$.

Through the experiments, we investigate the following:

- **Cook's problem, single concave element**: For Cook's membrane problem [37], the error in tip displacement due to the presence of a single concave element is reported as the severity of tangling is increased.
- **Cook's problem, multiple concave elements**: For Cook's membrane problem, with numerous tangled elements: (a) The displacement at a prescribed location is reported for each load step. (b) Deformed configurations for tangled and regular meshes are also compared. (c) Convergence of the tip displacement as a function of mesh size is studied and compared against standard FEM. (d) Finally, the convergence rate is evaluated.
- **Punch problem, material non-linearity**: For the punch problem [43], we include material non-linearity and study the convergence characteristics of i-TFEM.

- **Punch problem, multiple overlap**: For the punch problem [43], mesh with multiple overlap is considered.

- **Thin beam problem**: The performance of i-TFEM in bending dominated response is evaluated with neo-Hookean material model.

- **Aircraft model**: An example of a tangled mesh is presented to evaluate i-TFEM in practical scenarios.

7.3.1 2D Cook's Membrane: Single Concave Element

To begin with, we solve Cook's membrane problem over the mesh with one concave element illustrated earlier in Fig. 7.1b. Recall that the extent of tangling is controlled by the parameter d. For $d > 0.1$, a sharp increase in FEM error was observed, as illustrated in Fig. 7.8. On the other hand, using i-TFEM, the error, in fact, decreases (slightly) for $d > 0.1$; see Fig. 7.8.

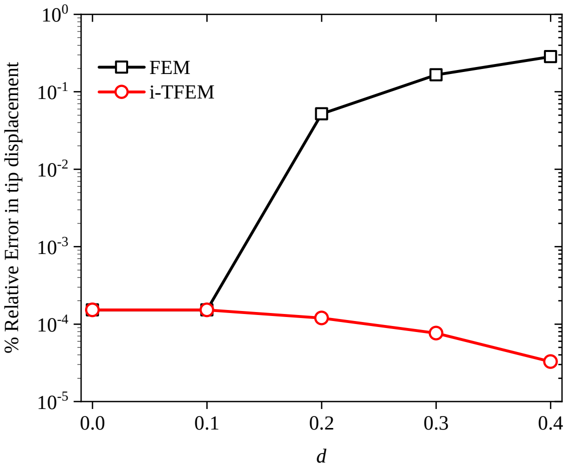


Figure 7.8: Relative error in tip displacement versus d for FEM and i-TFEM.

7.3.2 2D Cook's Membrane: Multiple Concave Elements

Next, we consider a regular mesh illustrated in Fig. 7.9a and a highly tangled mesh in Fig. 7.9b where every other element is concave.

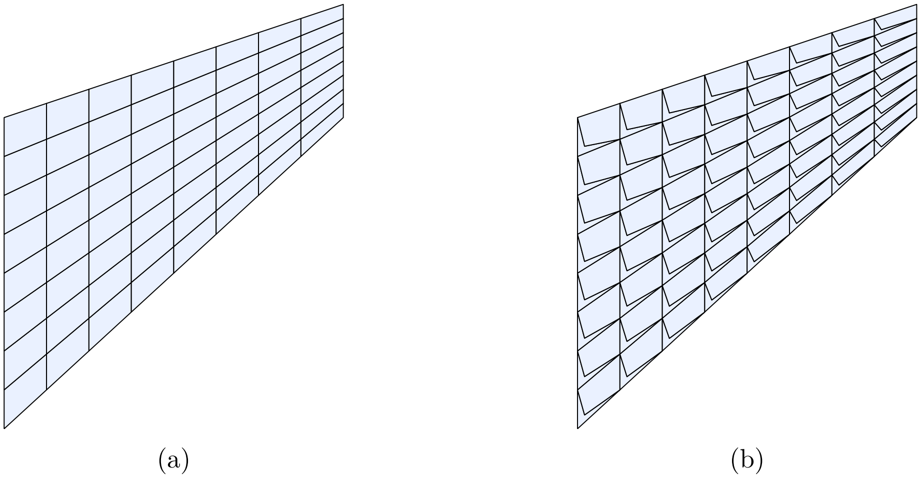


Figure 7.9: Initial configuration for (a) Regular mesh. (b) tangled mesh with $N = 3 \equiv 8 \times 8$ for the Cook's membrane problem.

The Cook's membrane problem is solved over the regular mesh using standard FEM, and over the tangled mesh using i-TFEM. The vertical displacement at the top right corner point A (see Fig. 7.1a) for every load step is reported in Fig. 7.10. One can observe a close agreement between the two.

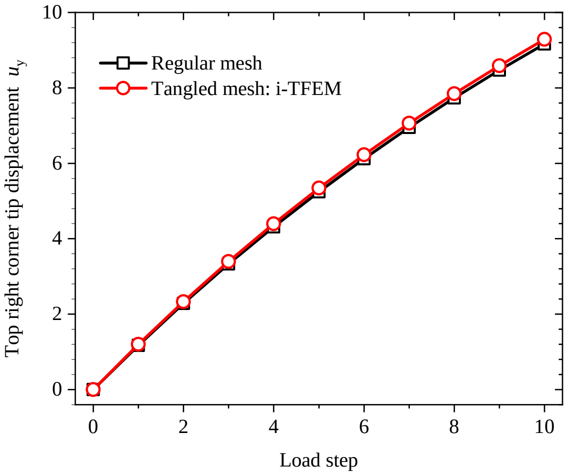
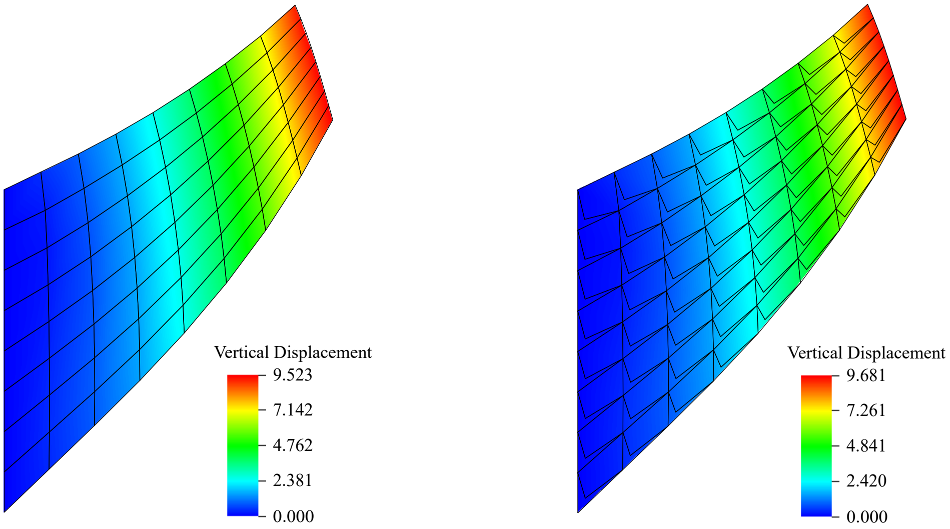


Figure 7.10: Vertical displacement versus the load step for Cook's membrane problem.

The deformed configuration for regular and tangled meshes after the last load step are reported in Fig. 7.11a and Fig. 7.11b, respectively.



(a) (b)

Figure 7.11: Deformed configuration for (a) Regular mesh and (b) Tangled mesh using i-TFEM for Cook's membrane problem.

To study convergence, the number of elements is controlled by a mesh-index N, where the number of elements in the regular mesh is $2^N \times 2^N$. Fig. 7.9a illustrates the regular mesh when $N = 3$, and Fig. 7.9b, the corresponding tangled mesh. We now compare the solutions from three different methods: standard FEM over regular mesh, standard FEM over tangled mesh, and i-TFEM over tangled mesh. The vertical displacements at point A for all three are plotted as a function of N in Fig. 7.12. Observe that FEM over a regular mesh and i-TFEM over the tangled mesh converge to the same displacement. On the other hand, FEM over a tangled mesh leads to erroneous results.

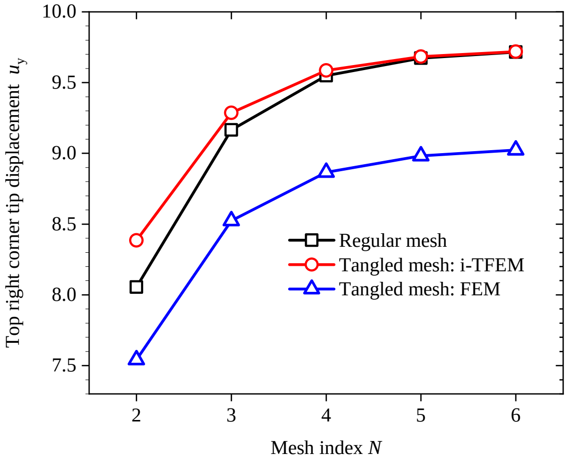
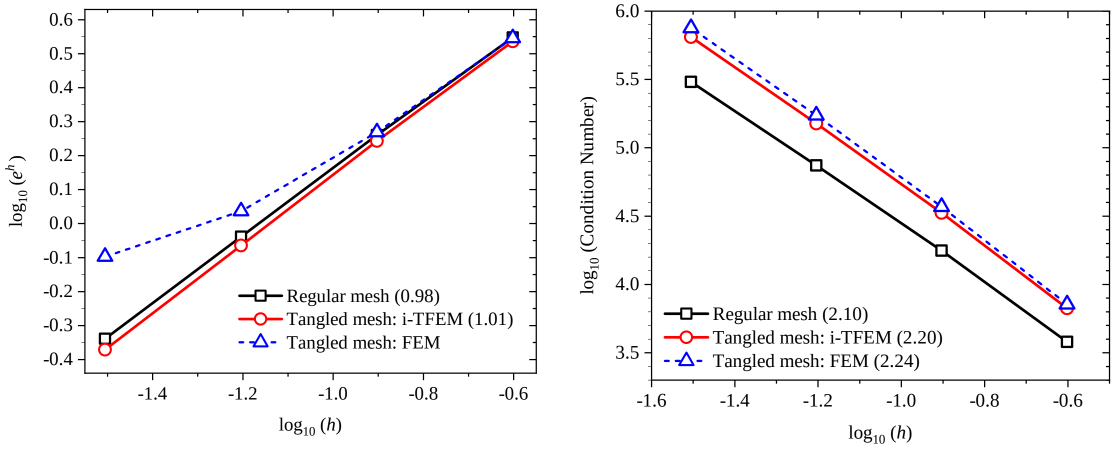


Figure 7.12: Convergence study for Cook's membrane problem.

To study the rate of convergence, we define the H^1 seminorm of the displacement error as

$$e^h = ||\nabla \boldsymbol{u} - \nabla \boldsymbol{u}^h|| = \left[\int_\Omega |\nabla \boldsymbol{u} - \nabla \boldsymbol{u}^h|^2 \, d\Omega \right]^{0.5} \tag{7.24}$$

where $\boldsymbol{u}$ is the reference solution from a fine mesh with $N = 7$, and $\boldsymbol{u}^h$ is the solution under consideration. Fig. 7.13a illustrates the error vs. mesh size (h) on a log-log scale over the non-tangled mesh as well as over the tangled mesh using FEM and i-TFEM. One can observe a near-optimal convergence rate for i-TFEM. Next, the effect of mesh size on the condition number of the matrix in Eq. 7.21 is studied. Fig. 7.13b shows that the condition number for tangled meshes increases with mesh size at a rate similar to that of a regular mesh.



(a)

(b)

Figure 7.13: (a)H^1 seminorm error versus mesh size and (b) condition number versus mesh size for Cook's problem. The convergence rates are provided in the brackets.

7.3.3 Punch Problem: Material Nonlinearity

Next, we consider a punch problem [173, 94, 43, 73] with geometric and material nonlinearities. Specifically, compressible isotropic generalized neo-Hookean material model is considered where the strain energy density is given by [186, 21]:

$$\Psi_{\text{GNH}}(\boldsymbol{u}) = \frac{\mu}{2}\left(J_F^{-2/3} tr\boldsymbol{b} - 3\right) + \frac{K}{2}\left(J_F - 1\right)^2 \tag{7.25}$$

where $J_F = det\boldsymbol{F}$ and $\boldsymbol{b} = \boldsymbol{F}\boldsymbol{F}^\top$ is the left Cauchy-Green deformation tensor while $\mu = 500$ and $K = 1700$ are the material parameters (equivalent to shear and bulk moduli respectively in the small strain limit). A rectangular block is subject to a vertical load p (per unit length) uniformly distributed over the top left half of the block where $p = 1000$ and $H = 1$; see Fig. 7.14 [43]. The top and left sides of the block are fixed in the horizontal direction, while the bottom is fixed in the vertical direction.

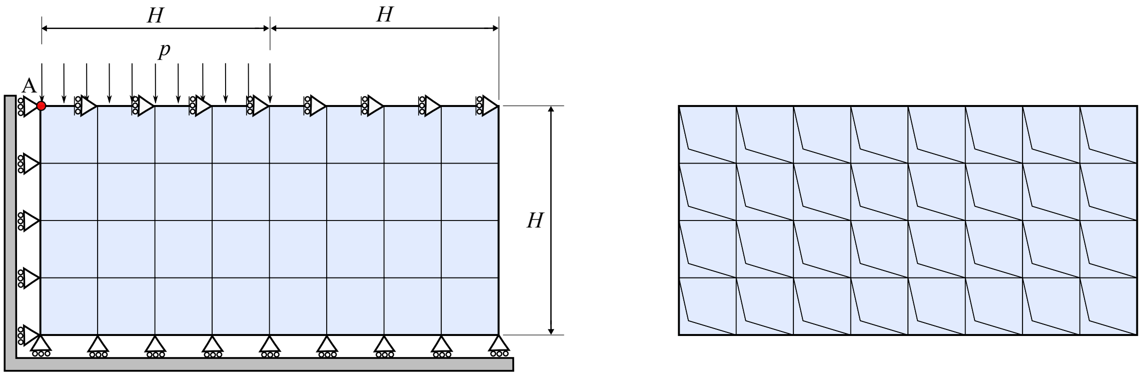


Figure 7.14: Initial configuration of the punch problem with mesh size $N = 2 \equiv 8 \times 4$ and the tangled mesh.

Fig. 7.15 captures the vertical displacement of point A (located at the top left corner) for every load step. The results for the regular mesh and tangled mesh (using i-TFEM) match well. For both meshes, the solution converged in about 5 Newton iterations for each load step.

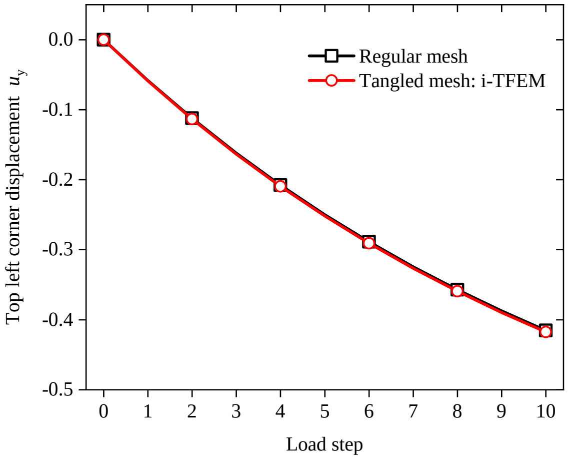
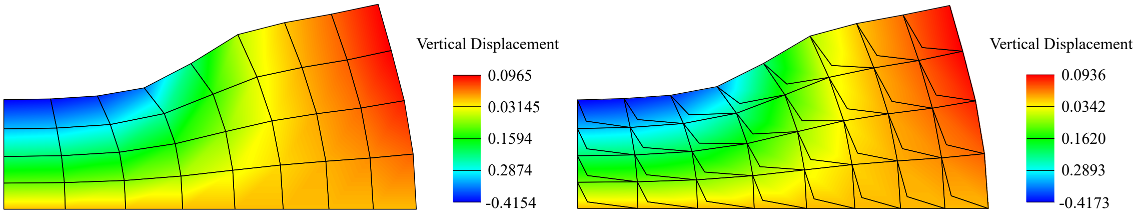


Figure 7.15: Vertical displacement versus the load step for the punch problem.

Fig. 7.16a and Fig. 7.16b illustrate the deformed configurations for the regular mesh and tangled meshes respectively, after the final load step.



(a) (b)

Figure 7.16: Deformed configuration for (a) regular mesh via FEM and (b) tangled mesh via i-TFEM for the punch problem.

To study the convergence, we use the mesh index N where the number of elements in the regular mesh is $2^{N+1} \times 2^N$. The regular and tangled meshes with $N = 2$ are shown in Fig. 7.26. A convergence study was then carried out as N was varied. The vertical displacement u_y at point A for the two methods is plotted against the mesh index N in Fig. 7.17. One can observe that the two methods converge to the same solution while standard FEM over the tangled mesh converges to an incorrect solution.

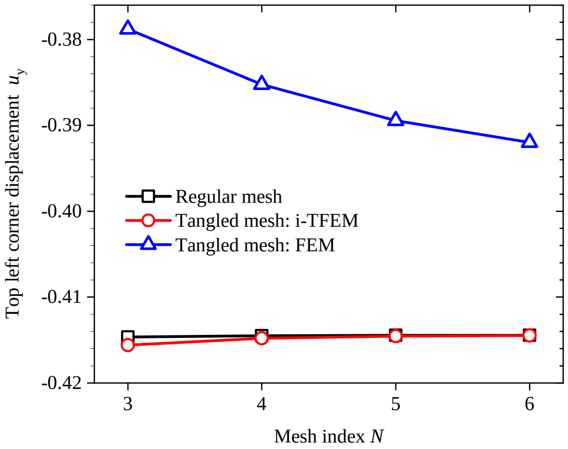
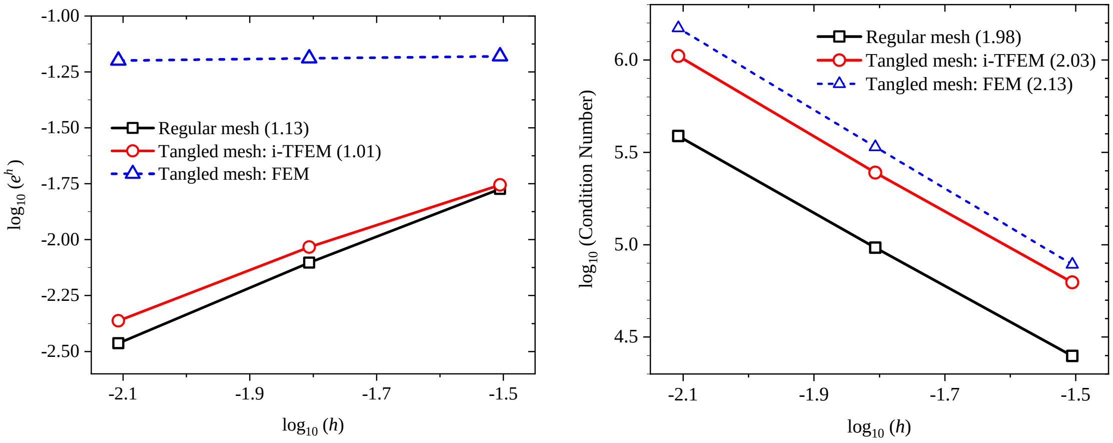


Figure 7.17: Convergence study for the punch problem.

Finally, Fig. 7.18a illustrates the H^1 seminorm error over the tangled mesh using FEM and i-TFEM as well as over the regular mesh. The reference solution is obtained with $N = 8$. Once again, i-TFEM exhibits a convergence rate for the H^1 seminorm error (Fig. 7.18a) and the condition number (Fig. 7.18b) similar to that obtained with a non-tangled mesh.



(a)

(b)

Figure 7.18: (a) H^1 seminorm error versus mesh size and (b) condition number versus mesh size for the punch problem. The convergence rates are provided in the brackets.

Multiple overlaps

Thus far, the fold was shared by only one neighboring convex element. However, in practice, the fold may be shared by multiple convex elements as illustrated in Fig. 7.19. In this case, three convex elements E_2, E_3 and E_4 share the folded region F_1. However this does not change the methodology, i.e., the tangent matrices and constraint matrix are computed as before (1) the tangent matrix $\boldsymbol{K}^t_{convex}$ and residual vector $\boldsymbol{R}^u_{convex}$ corresponding to the convex elements are computed using the three convex elements, (2) while $\widehat{\boldsymbol{K}}^t_{concave}$ and $\widehat{\boldsymbol{R}}^u_{concave}$ are computed using the parametric space associated with $\widehat{E}_1$, and (3) the constraint matrix is computed using the entire fold F_1.

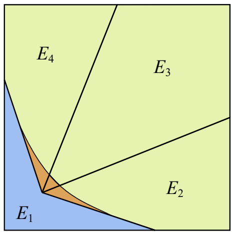


Figure 7.19: Four-element patch with one element concave.

Here, we consider a tangled mesh (see Fig. 7.20a) where the basic repeating unit is the patch shown in Fig. 2.6. The problem described in Fig. 7.26 is used as a case study. The final deformed configuration obtained via i-TFEM is shown in Fig. 7.20b.

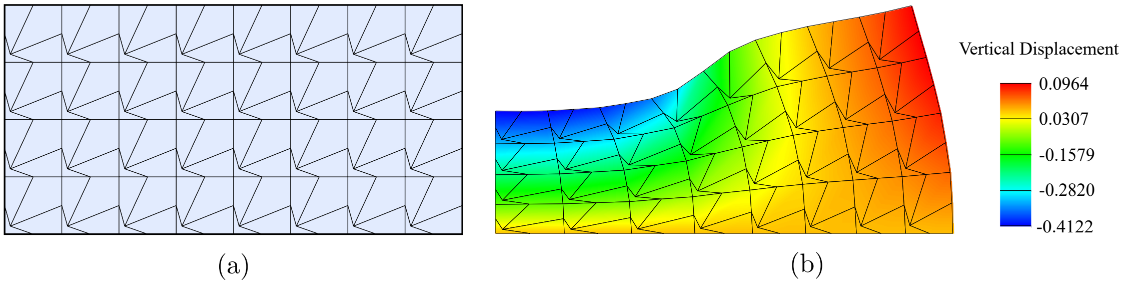


Figure 7.20: (a) $N = 3$ tangled mesh (b) Final deformed shape obtained via i-TFEM.

Next, to study the convergence, we use the mesh index N where the number of elements is $2^{N+1} \times 2^N$. Fig. 7.21a illustrates the H^1 seminorm error over the tangled mesh using FEM and i-TFEM as well as over the regular mesh. The reference solution is obtained with $N = 8$. Once again, i-TFEM exhibits an optimal convergence rate (Fig. 7.21a). Moreover, the condition number for i-TFEM is comparable to FEM (Fig. 7.21b).

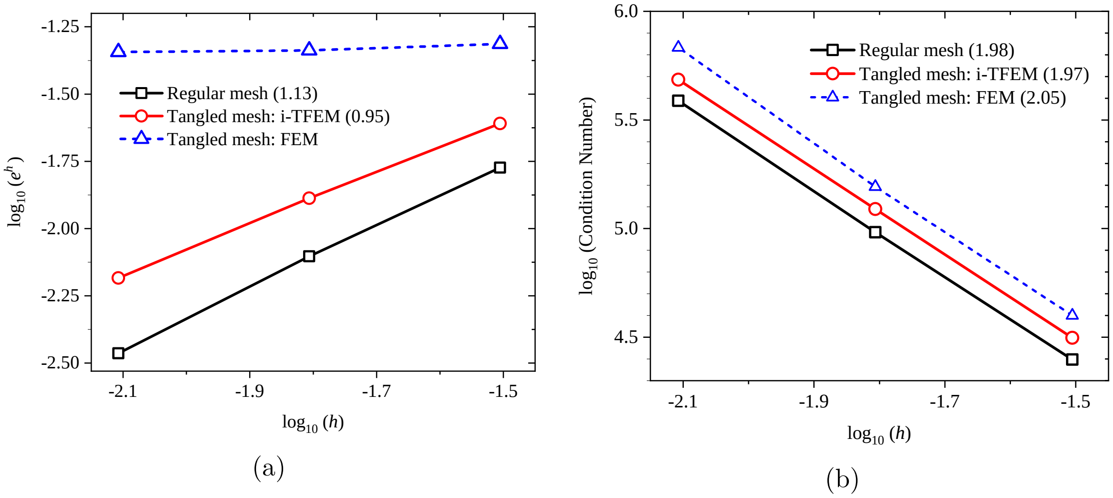


Figure 7.21: (a) H^1 seminorm error versus mesh size and (b) condition number versus mesh size for the punch problem with the four-element patch as the repeating unit. The convergence rates are provided in the brackets.

7.3.4 Thin Beam

In this example, a beam undergoing large deflections is considered [94, 173, 43]. Specifically, a beam (see Fig. 7.22a) with a length-to-height ratio $L/H = 100$ is fixed at the left end and subjected to a vertical load $F = 0.1$ at the right end. The material parameters of the Neo-Hookean model (Eq. 7.25) are $K = 16000$ and $\mu = 6000$. The regular and tangled meshes are shown in Fig. 7.22a and Fig. 7.22b, respectively.

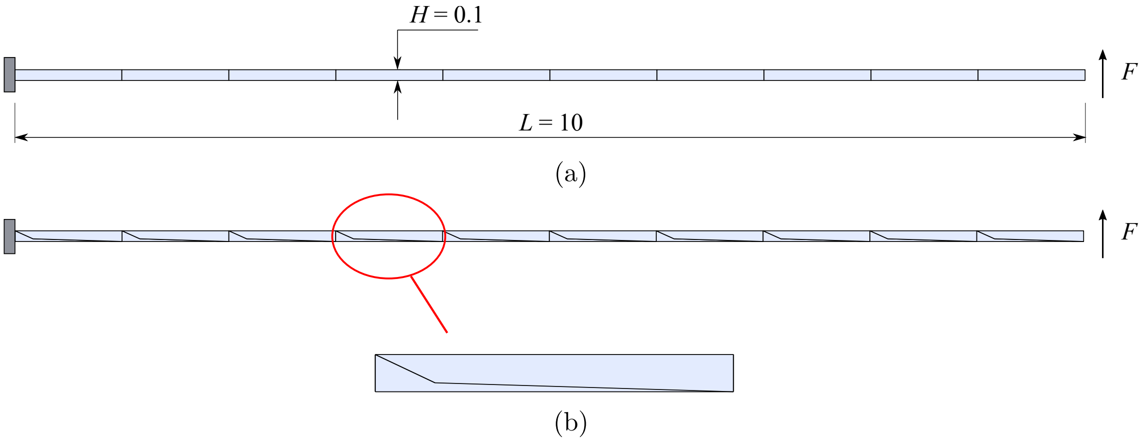


Figure 7.22: (a) Thin beam geometry and boundary conditions with regular mesh and (b) the corresponding tangled mesh. The repeating unit for the tangled mesh is zoomed in.

The number of elements in the mesh is governed by the mesh index N. For the regular mesh, the number of elements in the horizontal direction is given as (10×2^N) while in the vertical direction, the number of elements is given by 2^N The regular mesh in Fig 23 a corresponds to $N = 0$. To obtain the corresponding tangled mesh, each element is divided into a concave and a convex element. Hence the total number of elements in the tangled mesh is $2 \times (10 \times 2^N) \times 2^N$. The final deformed configuration of the beam obtained via i-TFEM over the tangled mesh with $N = 3$ is shown in Fig. 7.23.

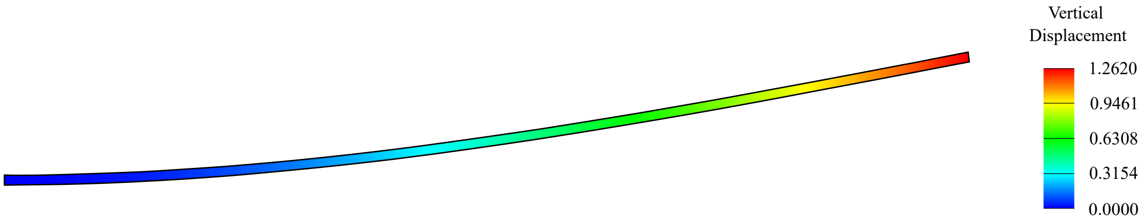


Figure 7.23: Final deformed shape using i-TFEM with $N = 3$ tangled mesh.

To study the convergence, the vertical displacement at the top right corner of the beam is considered. Fig. 7.24 plots the convergence of the regular mesh as well as the tangled mesh with FEM and i-TFEM. It can be seen that with i-TFEM, the solution converges to the same value as that obtained using the regular mesh.

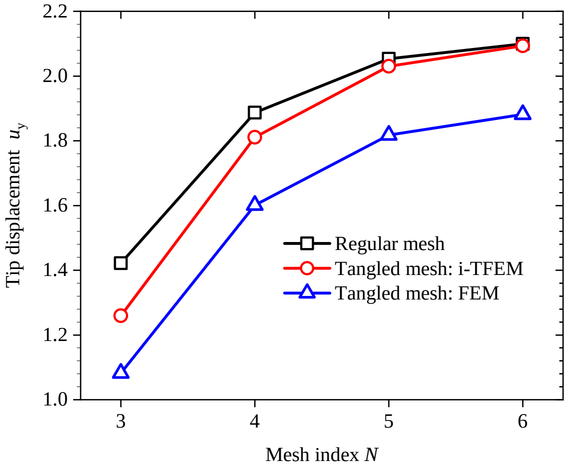


Figure 7.24: Convergence study for thin beam bending problem.

7.3.5 Application: Aircraft Model

An example of a practical tangling problem is shown in Figure 7.25a, where the quadrilateral mesh for an aircraft model was created using the quad mesher proposed in [146]. One quad element (out of 600) was found to be concave, for this particular model. The material parameters of the St. Venant-Kirchhoff model considered are $E = 20$ and $\nu = 0.3$. Symmetric (traction) boundary conditions are applied as illustrated in Fig. 7.25a; the remaining boundary segments are subjected to homogenous Dirichlet boundary conditions. The problem was solved using i-TFEM and FEM. While FEM required 1.85 seconds to solve the problem, i-TFEM required 2.13 seconds.

To compare the accuracy of FEM and i-TFEM, the reference solution was obtained by solving the same problem over a very fine quadrilateral mesh with nearly 10,000 elements. The problem was then solved over the tangled quad mesh shown in Fig. 7.25a using FEM and i-TFEM. The i-TFEM solution at the re-entrant vertex had a relative error of 0.018%, while the error for FEM was 0.11%. The i-TFEM post-processed solution is illustrated in Fig. 7.25b.

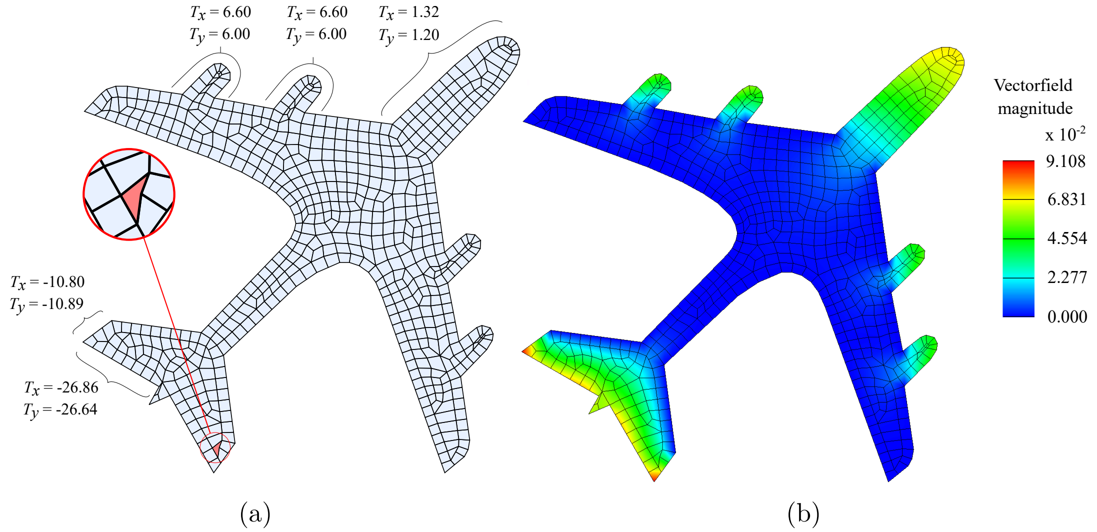


Figure 7.25: (a) Mesh for an aircraft model, with one concave element. (b) i-TFEM solution.

7.4 Accelerated TFEM for Nonlinear Elasticity

7.4.1 Formulation

The proposed a-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 [186, 21]. In standard FEM, the structural equilibrium can be captured via the residual force vector $\boldsymbol{R}$ as

$$\boldsymbol{R}(\hat{\boldsymbol{u}}) = \boldsymbol{f}_{int} - \boldsymbol{f}_{ext} = \boldsymbol{0}, \tag{7.26}$$

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

However, in a-TFEM, the constraint matrix $\boldsymbol{C}$ must be incorporated into the equilibrium equation (Eq. 7.26) via the Lagrange multipliers $\hat{\boldsymbol{\lambda}}$ as follows:

$$\widehat{\boldsymbol{R}}(\hat{\boldsymbol{u}}) + \boldsymbol{C}\hat{\boldsymbol{\lambda}} = \boldsymbol{0} \tag{7.27}$$

$$\boldsymbol{C}^{\top}\hat{\boldsymbol{u}} = \boldsymbol{0} \tag{7.28}$$

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 [131] for details):

$$\begin{bmatrix} \widehat{\boldsymbol{K}}_t & \boldsymbol{C} \\ \boldsymbol{C}^\top & \boldsymbol{0} \end{bmatrix} \begin{Bmatrix} \Delta\hat{\boldsymbol{u}}^n \\ \Delta\hat{\boldsymbol{\lambda}}^n \end{Bmatrix} = \begin{Bmatrix} -\left(\widehat{\boldsymbol{R}} + \boldsymbol{C}\hat{\boldsymbol{\lambda}}^n\right) \\ \boldsymbol{0} \end{Bmatrix}. \tag{7.29}$$

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

$$\widehat{\boldsymbol{K}}_t = \partial\widehat{\boldsymbol{R}}(\hat{\boldsymbol{u}}^{n-1})/\partial\hat{\boldsymbol{u}}. \tag{7.30}$$

The constraint matrix $\boldsymbol{C}$ can be obtained by evaluating the algebraic equations as described in the previous chapters (for instance, see Section 6.1.1 for hexahedral elements).. 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, a-TFEM reduces to the standard FEM. Thus, Eq. 7.29 reduces to:

$$\widehat{\boldsymbol{K}}_t\Delta\hat{\boldsymbol{u}}^n = -\widehat{\boldsymbol{R}}. \tag{7.31}$$

7.4.2 Q9 Element: Punch Problem

To evaluate the performance of a-TFEM in the presence of geometric and material nonlinearities, we consider the punch problem [173, 94, 43, 73]. A rectangular block is subject to a vertical load p (per unit length) uniformly distributed over the top left half of the block where $p = 1000$ and $H = 1$; see Fig. 7.26 [43]. The top and left sides of the block are fixed in the horizontal direction, while the bottom is fixed in the vertical direction. The domain is meshed using Q9 elements with curved edges. The mesh size is governed by the index N where the number of elements in the regular mesh is $2^{N+1} \times 2^N$. The regular and tangled meshes with $N = 2$ are shown in Fig. 7.26. Observe the elements in both meshes have curved edges.

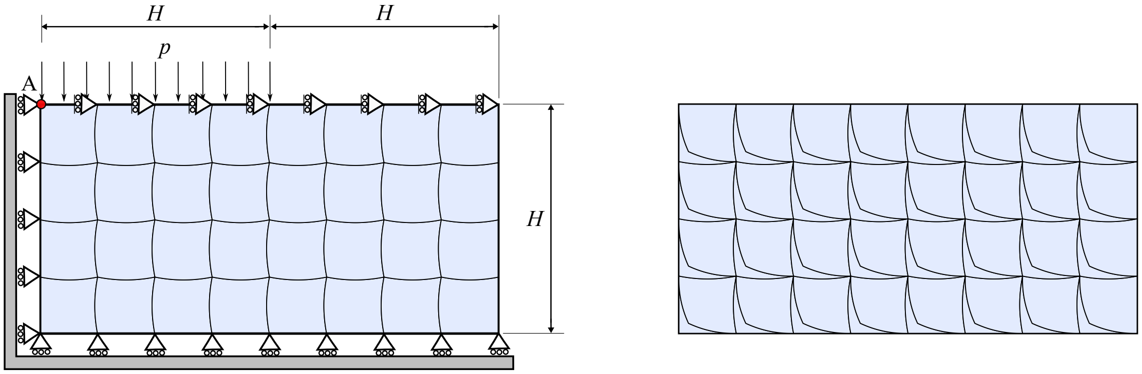


Figure 7.26: Initial configuration of the punch problem with mesh size $N = 2 \equiv 8 \times 4$ and the tangled Q9 mesh.

The problem is solved over these meshes (using a-TFEM and FEM) assuming plane strain condition and hyperelastic material model. Specifically, we employ the compressible isotropic generalized neo-Hookean material model where the strain energy density is given by [186, 21]:

$$\Psi_{\mathrm{GNH}}(\boldsymbol{u}) = \frac{\mu}{2}\left(J_F^{-2/3} tr\boldsymbol{b} - 3\right) + \frac{K}{2}\left(J_F - 1\right)^2. \tag{7.32}$$

Here, $\mu = 500$ and $K = 1700$ are the material parameters. Deformed configurations for untangled and tangled obtained (via a-TFEM) meshes are shown in Fig. 7.27.

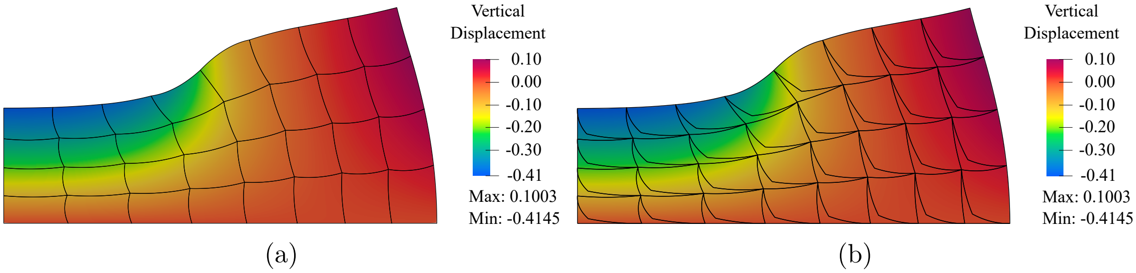


Figure 7.27: Deformed configuration for (a) regular mesh via FEM and (b) tangled mesh via a-TFEM for the punch problem.

Further, a convergence study was then carried out as N was varied. The vertical displacement u_2 at the top-left corner point A for the two methods is plotted against the mesh index N in Fig. 7.28. One can observe that the solutions obtained by FEM over the regular mesh and a-TFEM over the tangled mesh converge to the same solution while standard FEM over the tangled mesh converges to an incorrect solution.

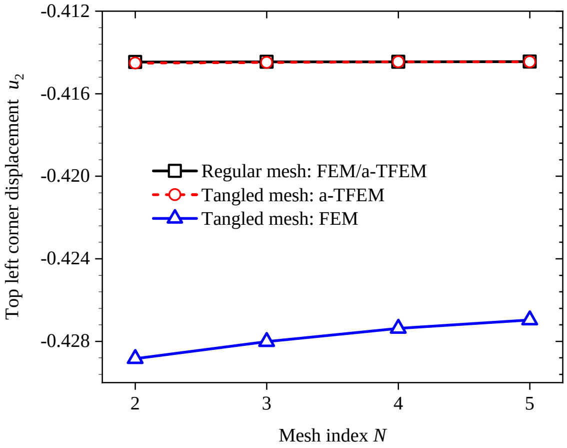
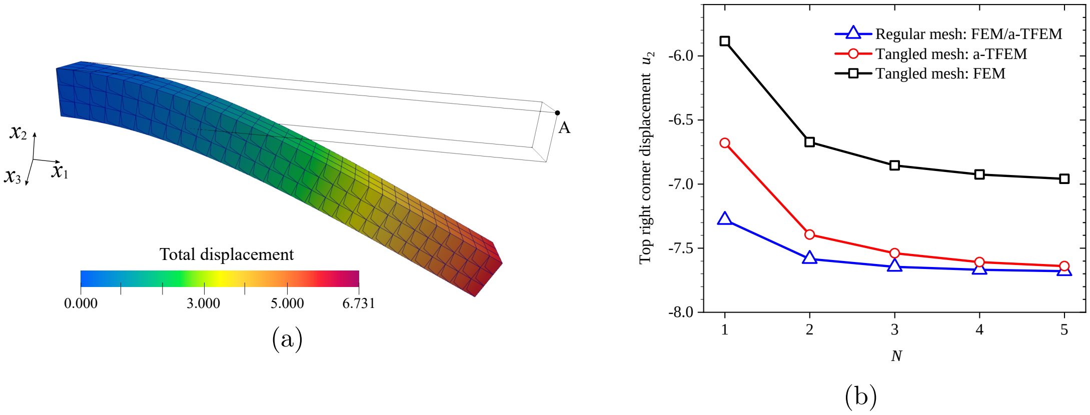


Figure 7.28: Convergence study for the punch problem with Q9 elements.

7.4.3 Hexahedral element: 3D Large Deformation Analysis

To evaluate the performance of a-TFEM in the presence of geometric and material nonlinearities, we consider the cantilever beam problem [116] 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 Eq. 7.32, where the material parameters are $\mu = 1.154 \times 10^7$ and $K = 2.5 \times 10^7$. Regular and tangled meshes ($d = 0.4$) are created by stacking the respective repeating units shown in Fig. 6.6 in the $10N \times N \times N$ arrangement. The deformed configuration for tangled mesh with $N = 1$ obtained via a-TFEM is shown in Fig. 7.29a. A convergence study is then carried out by varying N. The vertical displacement u_2 at the top-right corner point A for the two methods is plotted against the mesh index N in Fig. 7.29b. For the tangled mesh, similar to the linear elasticity problems presented earlier, a-TFEM converges to the expected solution while standard FEM converges to an incorrect solution.



(a)

(b)

Figure 7.29: (a) Deformed configuration of the cantilever beam with tangled mesh using a-TFEM. (b) Convergence study.

Chapter 8

Free and Forced Vibration Analysis Using Accelerated-TFEM

8.1 a-TFEM for Elastodynamics

8.1.1 Weak Formulation

Now consider an elastodynamics problem over a domain Ω that is discretized into m elements, some of which may be tangled (see Fig. 7.4). The body is subjected to body forces $\boldsymbol{b}$ and tractions $\boldsymbol{t}$ on $\partial\Omega^t$. Assume that the field $\boldsymbol{u}$ satisfies Dirichlet boundary conditions $\boldsymbol{u}_j = \boldsymbol{u}_j^d$ over $\partial\Omega^d$.

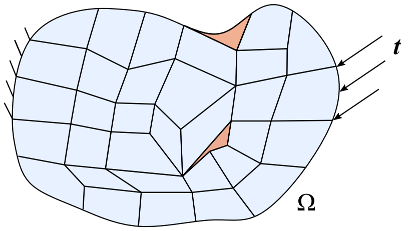


Figure 8.1: Domain Ω with boundary conditions and arbitrary mesh. Elements in red are tangled.

Let the elements be indexed by the set $I = \{1, \ldots, m\}$. Based on the principle of virtual work, recall the weak form of the governing equation for elastodynamics problems:

Find displacement $\boldsymbol{u} \in H^1$ such that

$$b(\delta\boldsymbol{u}, \rho\ddot{\boldsymbol{u}}) + a(\delta\boldsymbol{u}, \boldsymbol{u}) = f(\delta\boldsymbol{u}), \quad \forall \delta\boldsymbol{u} \in H_0^1 \tag{8.1}$$

$$\text{where,} \quad b(\delta\boldsymbol{u}, \rho\ddot{\boldsymbol{u}}) = \sum_{j \in I} \int_{E_j} \delta\boldsymbol{u}_j \cdot \rho\ddot{\boldsymbol{u}}_j \, \mathrm{d}\Omega \tag{8.1a}$$

$$a(\delta\boldsymbol{u}, \boldsymbol{u}) = \sum_{j \in I} \int_{E_j} \boldsymbol{\varepsilon}\left(\delta\boldsymbol{u}_j\right) : \boldsymbol{\sigma}\left(\boldsymbol{u}_j\right) \, \mathrm{d}\Omega \tag{8.1b}$$

$$f(\delta\boldsymbol{u}) = \sum_{j \in I} \int_{E_j} \delta\boldsymbol{u}_j \cdot \boldsymbol{b} \, \mathrm{d}\Omega + \sum_{j \in I} \int_{\partial E_j^t} \delta\boldsymbol{u}_j \cdot \boldsymbol{t} \, \mathrm{d}S \tag{8.1c}$$

where ρ is the mass density and $\ddot{\boldsymbol{u}}$ is the acceleration. Assuming linear elastic material, the stress tensor $\boldsymbol{\sigma}$ is computed using the elasticity tensor $\boldsymbol{D}$ and strain tensor $\boldsymbol{\varepsilon}$ as $\boldsymbol{\sigma} = \boldsymbol{D}\boldsymbol{\varepsilon}$.

When the mesh is tangled, the standard weak form will lead to erroneous results (demonstrated later in Section 5). To resolve this, two modifications are made. Analogous to computing the area as in Eq. 5.4, integrals over negative components are subtracted to avoid double-counting. For example, for the two-element tangled mesh in Fig. 7.4a, the first term of the weak form is expressed as:

$$b(\delta\boldsymbol{u}, \rho\ddot{\boldsymbol{u}}) = \int_{C_1^+} \delta\boldsymbol{u}_1^+ \cdot \rho\ddot{\boldsymbol{u}}_1^+ \mathrm{d}\Omega + \int_{C_2^+} \delta\boldsymbol{u}_2^+ \cdot \rho\ddot{\boldsymbol{u}}_2^+ \mathrm{d}\Omega - \int_{C_1^-} \delta\boldsymbol{u}_1^- \cdot \rho\ddot{\boldsymbol{u}}_1^- \mathrm{d}\Omega \tag{8.2}$$

This applies to all tangled elements, and to all integral terms.

Next, consider the field compatibility constraint. The Lagrange multiplier method [44, 126, 164, 49, 120, 5] is employed to enforce this constraint over every tangled element. For example, for the two-element mesh, following condition is required:

$$\int_{F_1} \delta\boldsymbol{\lambda}_1 \cdot [\![\boldsymbol{u}_1]\!] \, \mathrm{d}\Omega = 0 \quad \forall \; \delta\boldsymbol{\lambda}_1 \in L^2, \tag{8.3}$$

Moreover, Eq. 8.1 must be suitably modified as follows:

$$b(\delta\boldsymbol{u}, \rho\ddot{\boldsymbol{u}}) + a(\delta\boldsymbol{u}, \boldsymbol{u}) + \int_{F_1} [\![\delta\boldsymbol{u}_1]\!] \cdot \boldsymbol{\lambda}_1 \, \mathrm{d}\Omega = f(\delta\boldsymbol{u}), \quad \forall \delta\boldsymbol{u} \in H_0^1.$$

These concepts can now be generalized to an arbitrary mesh. Let the tangled elements be identified by the index $I_{\text{tangled}} \subset I$. The weak form in a-TFEM can then be expressed as follows:

Find $\boldsymbol{u} \in H^1$ and $\boldsymbol{\lambda} \in L^2$ such that

$$b(\delta\boldsymbol{u}, \rho\ddot{\boldsymbol{u}}) + a(\delta\boldsymbol{u}, \boldsymbol{u}) + \sum_{j \in I_{\text{tangled}}} \int_{F_j} [\![\delta\boldsymbol{u}_j]\!] \cdot \boldsymbol{\lambda}_j \, \mathrm{d}\Omega = f(\delta\boldsymbol{u}), \quad \forall \delta\boldsymbol{u} \in H_0^1 \tag{8.4a}$$

$$\sum_{j \in I_{\text{tangled}}} \int_{F_j} \delta\boldsymbol{\lambda}_j \cdot [\![\boldsymbol{u}_j]\!] \, \mathrm{d}\Omega = 0, \qquad \forall \delta\boldsymbol{\lambda} \in L^2 \tag{8.4b}$$

where,

$$b(\delta\boldsymbol{u}, \rho\ddot{\boldsymbol{u}}) = \sum_{j \in I} \int_{C_j^+} \delta\boldsymbol{u}_j^+ \cdot \rho\ddot{\boldsymbol{u}}_j^+ \mathrm{d}\Omega - \sum_{j \in I_{\text{tangled}}} \int_{C_j^-} \delta\boldsymbol{u}_j^- \cdot \rho\ddot{\boldsymbol{u}}_j^- \mathrm{d}\Omega \tag{8.5a}$$

$$a(\delta\boldsymbol{u}, \boldsymbol{u}) = \sum_{j \in I} \int_{C_j^+} \boldsymbol{\varepsilon}\left(\delta\boldsymbol{u}_j^+\right) : \boldsymbol{\sigma}\left(\boldsymbol{u}_j^+\right) \mathrm{d}\Omega - \sum_{j \in I_{\text{tangled}}} \int_{C_j^-} \boldsymbol{\varepsilon}\left(\delta\boldsymbol{u}_j^-\right) : \boldsymbol{\sigma}\left(\boldsymbol{u}_j^-\right) \mathrm{d}\Omega \tag{8.5b}$$

$$f(\delta\boldsymbol{u}) = \sum_{j \in I} \int_{C_j^+} \delta\boldsymbol{u}_j^+ \cdot \boldsymbol{b} \mathrm{d}\Omega - \sum_{j \in I_{\text{tangled}}} \int_{C_j^-} \delta\boldsymbol{u}_j^- \cdot \boldsymbol{b} \mathrm{d}\Omega + \sum_{j \in I} \int_{\partial E_j^t} \delta\boldsymbol{u}_j \cdot \boldsymbol{t} \mathrm{d}S \tag{8.5c}$$

8.1.2 Finite Element Approximation

The primary field $\boldsymbol{u}$ and the Lagrange multiplier field $\boldsymbol{\lambda}$ are approximated as follows:

$$\boldsymbol{u}_j \approx \boldsymbol{N}_j \boldsymbol{d}_j, \quad \boldsymbol{\lambda}_j \approx \boldsymbol{N}_j^{\lambda} \hat{\boldsymbol{\lambda}}_j \tag{8.6}$$

Adopting the (Bubnov-) Galerkin framework, Eq. 8.4 leads to the following system of equations:

$$\boldsymbol{M}\ddot{\boldsymbol{d}} + \boldsymbol{K}\boldsymbol{d} + \boldsymbol{C}\hat{\boldsymbol{\lambda}} = \boldsymbol{f} \tag{8.7a}$$

$$\boldsymbol{C}^{\top}\boldsymbol{d} = \boldsymbol{0} \tag{8.7b}$$

where the mass matrix is given by:

$$\boldsymbol{M} = \prod_{j \in I} \int_{C_j^+} \left(\boldsymbol{N}_j^{+\top} \rho \boldsymbol{N}_j^{+} \right) d\Omega \quad - \prod_{j \in I_{\text{tangled}}} \int_{C_j^-} \left(\boldsymbol{N}_j^{-\top} \rho \boldsymbol{N}_j^{-} \right) d\Omega, \tag{8.8}$$

the stiffness matrix is given by:

$$\boldsymbol{K} = \prod_{j \in I} \int_{C_j^+} \left(\nabla \boldsymbol{N}_j^{+\top} \boldsymbol{D} \nabla \boldsymbol{N}_j^{+} \right) d\Omega \quad - \prod_{j \in I_{\text{tangled}}} \int_{C_j^-} \left(\nabla \boldsymbol{N}_j^{-\top} \boldsymbol{D} \nabla \boldsymbol{N}_j^{-} \right) d\Omega, \tag{8.9}$$

the forcing term is given by:

$$\boldsymbol{f} = \prod_{j \in I} \int_{C_j^+} \boldsymbol{N}_j^{\top} \boldsymbol{b} \, d\Omega - \prod_{j \in I_{\text{tangled}}} \int_{C_j^-} \boldsymbol{N}_j^{-\top} \boldsymbol{b} \, d\Omega + \prod_{j \in I} \int_{\partial E_j^t} \boldsymbol{N}_j^{\top} \boldsymbol{t} \, dS, \tag{8.10}$$

and the constraint matrix by:

$$\boldsymbol{C} = \prod_{j \in I_{\text{tangled}}} \int_{F_j} [\![\boldsymbol{N}_j]\!]^{\top} \boldsymbol{N}^{\lambda} d\Omega = \prod_{j \in I_{\text{tangled}}} \int_{F_j} \left(\boldsymbol{N}_j^{+} - \boldsymbol{N}_j^{-} \right)^{\top} \boldsymbol{N}^{\lambda} d\Omega. \tag{8.11}$$

In Eq. 7.22, the choice of $\boldsymbol{N}^{\lambda}$ depends on the choice of $\boldsymbol{N}$, i.e., the type of element (Q4, Q9, H8, etc.). Further, observe that the expressions in Eq. 8.8 through Eq. 7.22 entail integration over non-convex regions (C_j^+, C_j^-, and F_j), associated with tangled elements. Fortunately, this can be circumvented, as discussed in the previous chapters.

One can incorporate damping with a damping matrix $\boldsymbol{G}$ and the velocity vector $\dot{\boldsymbol{d}}$ as:

$$\boldsymbol{M} \ddot{\boldsymbol{d}} + \boldsymbol{G} \dot{\boldsymbol{d}} + \boldsymbol{K} \boldsymbol{d} + \boldsymbol{C} \hat{\boldsymbol{\lambda}} = \boldsymbol{f} \tag{8.12a}$$

$$\boldsymbol{C}^{\top} \boldsymbol{d} = \boldsymbol{0} \tag{8.12b}$$

In this work, for simplicity, Rayleigh damping is assumed, i.e., $\boldsymbol{G} = \alpha_1 \boldsymbol{M} + \alpha_2 \boldsymbol{K}$, where α_1 and α_2 are the Rayleigh damping coefficients.

8.1.3 Forced Vibration

Many schemes [24, 10, 71] can be used to solve Eq. 8.12. In this work, the Newmark method (with implicit time integration) [71, 113] is adopted. Specifically, Eq. 8.12 at time $t + \Delta t$ is expressed as:

$$\boldsymbol{M}\ddot{\boldsymbol{d}}_{t+\Delta t} + \boldsymbol{G}\dot{\boldsymbol{d}}_{t+\Delta t} + \boldsymbol{K}\boldsymbol{d}_{t+\Delta t} + \boldsymbol{C}\boldsymbol{\lambda}_{t+\Delta t} = \boldsymbol{f}_{t+\Delta t} \tag{8.13a}$$

$$\boldsymbol{C}^{\top}\boldsymbol{d}_{t+\Delta t} = \boldsymbol{0} \tag{8.13b}$$

The displacement and velocity vectors are updated as [185, 69]:

$$\boldsymbol{d}_{t+\Delta t} = \boldsymbol{q}^0 + \left(\beta\Delta t^2\right)\ddot{\boldsymbol{d}}_{t+\Delta t} \quad \text{where,} \quad \boldsymbol{q}^0 = \boldsymbol{d}_t + \Delta t\dot{\boldsymbol{d}}_t + \frac{(\Delta t)^2}{2}(1-2\beta)\,\ddot{\boldsymbol{d}}_t \tag{8.14a}$$

$$\text{and} \quad \dot{\boldsymbol{d}}_{t+\Delta t} = \boldsymbol{q}^1 + (\gamma\Delta t)\,\ddot{\boldsymbol{d}}_{t+\Delta t} \quad \text{where,} \quad \boldsymbol{q}^1 = \dot{\boldsymbol{d}}_t + \Delta t\,(1-\gamma)\,\ddot{\boldsymbol{d}}_t \tag{8.14b}$$

where the Newmark paramters β and γ are set to $\beta = 0.25$ and $\gamma = 0.5$ [185, 69, 35, 60].

Substituting Eq. 8.14 in Eq. 8.13:

$$\begin{bmatrix} \boldsymbol{M} + (\gamma\Delta t)\,\boldsymbol{G} + \left(\beta\Delta t^2\right)\boldsymbol{K} & \boldsymbol{C} \\ \left(\beta\Delta t^2\right)\boldsymbol{C}^{\top} & \boldsymbol{0} \end{bmatrix} \begin{Bmatrix} \ddot{\boldsymbol{d}}_{t+\Delta t} \\ \hat{\boldsymbol{\lambda}}_{t+\Delta t} \end{Bmatrix} = \begin{Bmatrix} \boldsymbol{f}_{t+\Delta t} - \boldsymbol{K}\boldsymbol{q}^0 - \boldsymbol{M}\boldsymbol{q}^1 \\ -\boldsymbol{C}^{\top}\boldsymbol{q}^0 \end{Bmatrix}. \tag{8.15}$$

that must be solved at each time step. Observe that when the mesh does not contain any tangled elements, Eq. 8.15 reduces to (standard FEM):

$$\left[\boldsymbol{M} + (\gamma\Delta t)\,\boldsymbol{G} + \left(\beta\Delta t^2\right)\boldsymbol{K}\right]\ddot{\boldsymbol{d}}_{t+\Delta t} = \boldsymbol{f}_{t+\Delta t} - \boldsymbol{K}\boldsymbol{q}^0 - \boldsymbol{M}\boldsymbol{q}^1 \tag{8.16}$$

Here, the expressions for $\boldsymbol{M}$, $\boldsymbol{K}$ and $\boldsymbol{f}$ (Eq. 8.8 to Eq. 8.10) also reduce to that of the standard FEM.

8.1.4 Free Vibration

If no damping or forcing terms exist, Eq. 8.12 reduces to:

$$\boldsymbol{M}\ddot{\boldsymbol{d}} + \boldsymbol{K}\boldsymbol{d} + \boldsymbol{C}\hat{\boldsymbol{\lambda}} = \boldsymbol{0} \tag{8.17a}$$

$$\boldsymbol{C}^{\top}\boldsymbol{d} = \boldsymbol{0} \tag{8.17b}$$

In free vibration analysis, $\boldsymbol{d}(\boldsymbol{x}, t)$ can be expressed as $\boldsymbol{d}(\boldsymbol{x}, t) = \bar{\boldsymbol{d}}(\boldsymbol{x}) \sin(\omega t)$, where ω is the frequency and $\bar{\boldsymbol{d}}$ is the eigenvector. Thus, Eq. 8.17 reduces to:

$$-\omega^2 \boldsymbol{M}\bar{\boldsymbol{d}} + \boldsymbol{K}\bar{\boldsymbol{d}} + \boldsymbol{C}\hat{\boldsymbol{\lambda}} = \boldsymbol{0} \tag{8.18a}$$

$$\boldsymbol{C}^{\top}\bar{\boldsymbol{d}} = \boldsymbol{0} \tag{8.18b}$$

Thus, one must solve the following Eigen system:

$$\begin{bmatrix} \boldsymbol{K} & \boldsymbol{C} \\ \boldsymbol{C}^{\top} & \boldsymbol{0} \end{bmatrix} \begin{Bmatrix} \bar{\boldsymbol{d}} \\ \hat{\boldsymbol{\lambda}} \end{Bmatrix} = \omega^2 \begin{bmatrix} \boldsymbol{M} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{Bmatrix} \bar{\boldsymbol{d}} \\ \hat{\boldsymbol{\lambda}} \end{Bmatrix}. \tag{8.19}$$

The system can be reduced by using the singular value decomposition [60] of $\boldsymbol{C}^{\top}$:

$$\boldsymbol{C}^{\top} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\top} \tag{8.20}$$

The matrix $\boldsymbol{V}$ can be written as:

$$\boldsymbol{V}^{\top} = \left\{\boldsymbol{V}_{n\times r}, \boldsymbol{V}_{n\times(n-r)}\right\}^{\top} \tag{8.21}$$

where r is the rank of $\boldsymbol{C}^{\top}$, namely the number of rows of $\boldsymbol{C}^{\top}$. Performing coordinate transformation (that satisfies the constraint),

$$\bar{\boldsymbol{d}} = \boldsymbol{V}_{n\times(n-r)}\widehat{\boldsymbol{d}} \tag{8.22}$$

Eq. 8.18 can be written as:

$$\widehat{\boldsymbol{K}}\widehat{\boldsymbol{d}} = \omega^2 \widehat{\boldsymbol{M}}\widehat{\boldsymbol{d}} \tag{8.23}$$

where, $\widehat{\boldsymbol{K}}_{(n-r)\times(n-r)} = \boldsymbol{V}^{\top}_{(n-r)\times n}\boldsymbol{K}_{n\times n}\boldsymbol{V}_{n\times(n-r)}$ and $\widehat{\boldsymbol{M}}_{(n-r)\times(n-r)} = \boldsymbol{V}^{\top}_{(n-r)\times n}\boldsymbol{M}_{n\times n}\boldsymbol{V}_{n\times(n-r)}$ are the dimension reduced stiffness and mass matrices respectively.

8.2 Numerical Experiments

The use of tangled meshes for free and forced vibration analysis is verified using the proposed a-TFEM framework. Numerical experiments are conducted under the following conditions:

- 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 standard Gaussian quadrature is employed for all elements; that is, 2×2 for Q4, 3×3 for Q9, and $2 \times 2 \times 2$ for H8.
- In standard FEM, the absolute value of 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 nonsensical results [131].

8.2.1 Cantilever Problem

Consider the cantilever beam in Fig. 8.2 with dimensions $L = 20$, $h = 1$, and $b = 1$. The material properties are: Young's modulus $E = 1000$, Poisson's ratio $\nu = 0.25$, and mass density $\rho = 0.016308$. This problem is discussed in [123], and is investigated here using Q4, Q9, and H8 tangled elements.

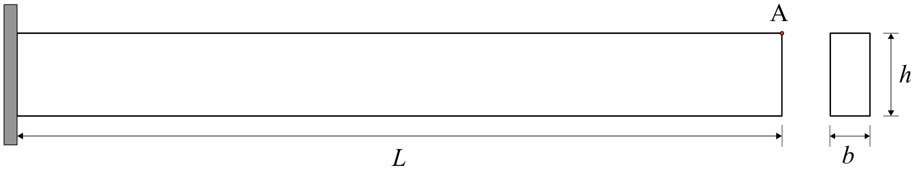


Figure 8.2: A cantilever beam problem.

2D cantilever: Mesh construction

The beam is first modeled as a plane stress problem. Let us consider tangled and non-tangled (regular) meshes for both 4-node (Q4) and 9-node (Q9) elements. The number of elements is controlled by a mesh-index N, where the number of elements in the regular mesh is $20N \times N$. The corresponding tangled mesh is created by replacing each element by a pair of elements (one of them being tangled) as illustrated in Fig. 8.4.

Fig. 8.3 illustrates a regular Q4 mesh when $N = 2$, and Fig. 8.4, the corresponding tangled mesh. The repeating units for both meshes are shown on the right-hand side of each figure. For the tangled mesh, the position of the re-entrant vertex R can be varied using the parameter $d \in (0, 0.5)$ where $d = 1/3$ in Fig. 8.4.

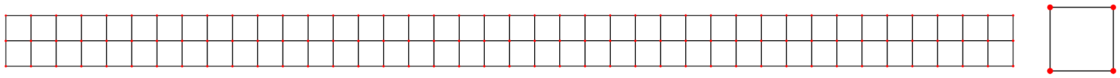
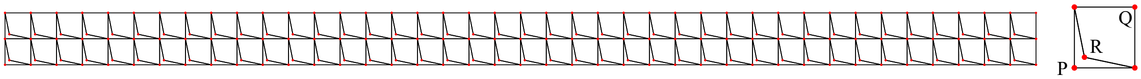


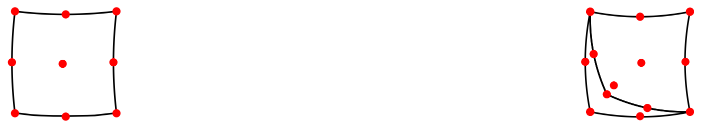
Figure 8.3: Regular Q4 mesh with $N = 2$



$$x_1^{R} = (0.5+d)\,(x_1^{P} - x_1^{Q}) \;+ x_1^{Q}$$
$$x_2^{R} = (0.5+1.05d)\,(x_2^{P} - x_2^{Q}) \;+ x_1^{Q}$$

Figure 8.4: Tangled Q4 mesh with $N = 2$.

The repeating units for regular and tangled Q9 meshes are illustrated in Fig. 8.5a and Fig. 8.5b respectively. Note the interior edges have a slight curvature. The extent of tangling is once again controlled by d.

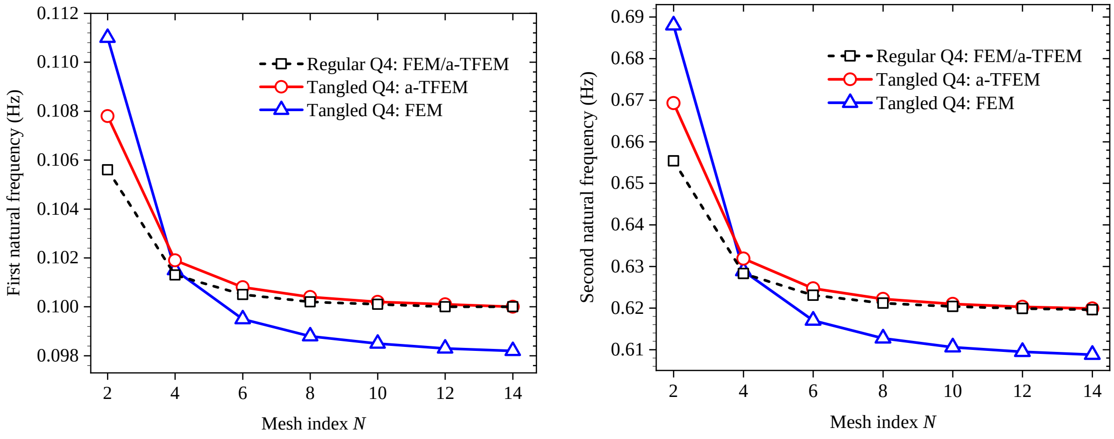


(a) (b)

Figure 8.5: Q9 repeating unit for (a) regular (b) tangled meshes.

2D cantilever: Convergence of natural frequencies

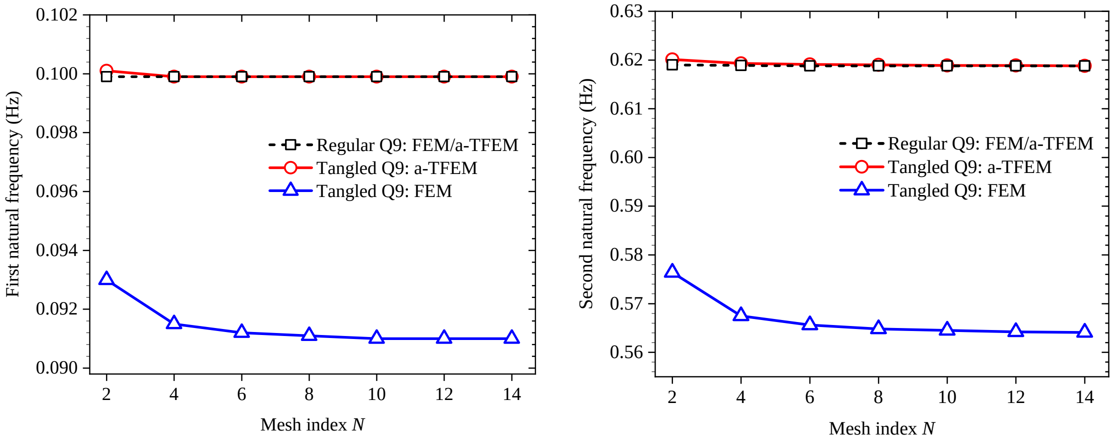
Let us now compare the natural frequencies of the cantilever computed from three different methods: standard FEM over regular mesh, standard FEM over tangled mesh, and a-TFEM over tangled mesh. Using the Q4 elements, for all three methods, the first and second natural frequencies are plotted as a function of N in Fig. 8.6a and Fig. 8.6b respectively. Observe that FEM over a regular mesh and a-TFEM over the tangled mesh converge to the same value. On the other hand, FEM over a tangled mesh converges to an incorrect solution.



(a) (b)

Figure 8.6: Convergence of (a) first and (b) second natural frequency of the cantilever computed with Q4 tangled elements.

Fig. 8.7 shows similar plots using Q9 elements. Here the error in standard FEM over tangled meshes is even more pronounced.



(a)

(b)

Figure 8.7: Convergence of (a) first and (b) second natural frequency of the cantilever computed with Q9 tangled elements.

2D cantilever: Forced vibrations

In this example, the same cantilever beam is subjected to a transient loading $f(t) = \sin(\pi t/T)$ for $0 \le t \le T$ where T is the natural period of the cantilever (see Fig. 8.8). The load is applied at the top right corner of the beam (point A in Fig. 8.2).

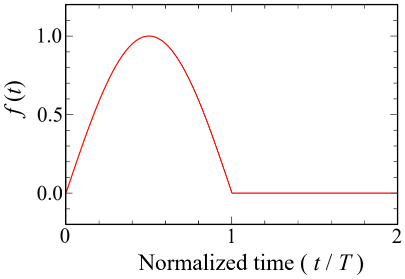


Figure 8.8: Transient loading for cantilever beam.

This problem is first solved using a tangled mesh with Q4 elements. Four different mesh sizes are considered: $N = 2, 3, 4$, and 8. The vertical displacement of the cantilever tip v_A is plotted with respect to time as normalized/dimensionless parameters $v_A EI/L^3$ and t/T respectively in Fig. 8.9, where $I = bh^3/12$ is the second moment of inertia. The reference solution is obtained using a regular Q9 mesh of size 160×8 ($N = 8$). As one can observe, a-TFEM converges to the reference solution as the mesh discretization is increased.

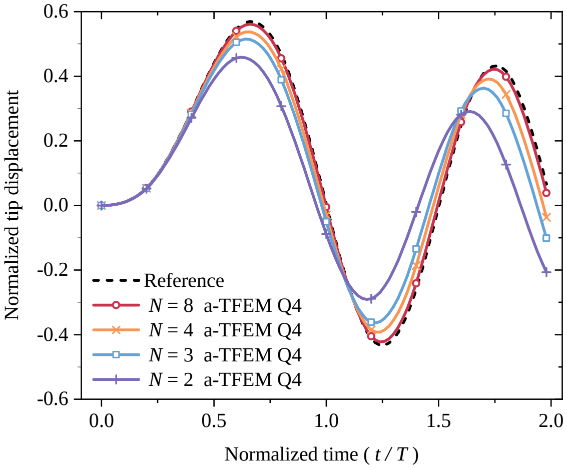


Figure 8.9: Convergence for the tangled Q4 meshes using a-TFEM.

Next, for the mesh size of N = 8, let us compare the solutions obtained by the three methods: FEM and a-TFEM using the tangled mesh and FEM using untangled mesh. The normalized tip displacements obtained using the three methods are plotted in Fig. 8.10a for Q4 elements, and Fig. 8.10b for Q9 elements. Observe that the response obtained using a-TFEM (over the tangled mesh) matches with that obtained over the regular mesh. However, FEM over the tangled mesh leads to incorrect results.

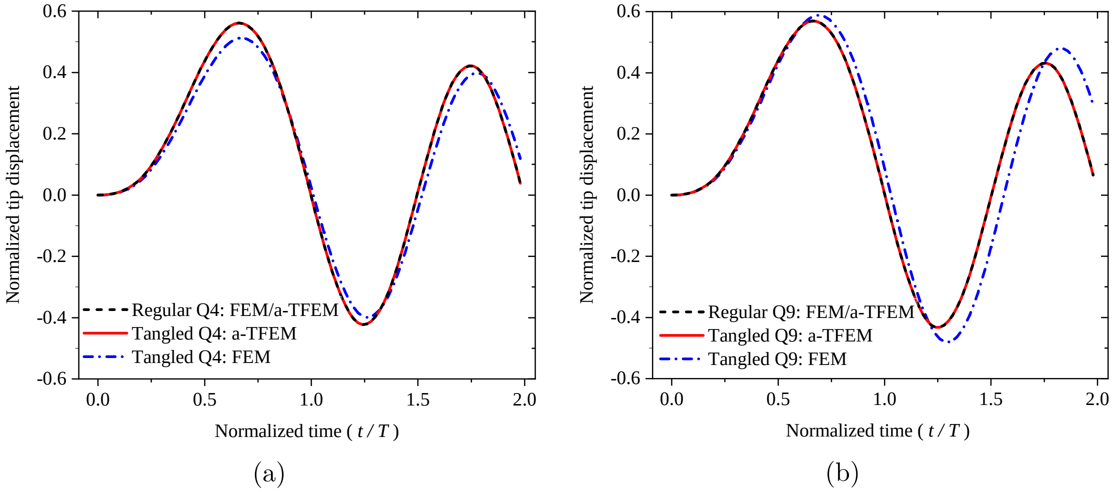
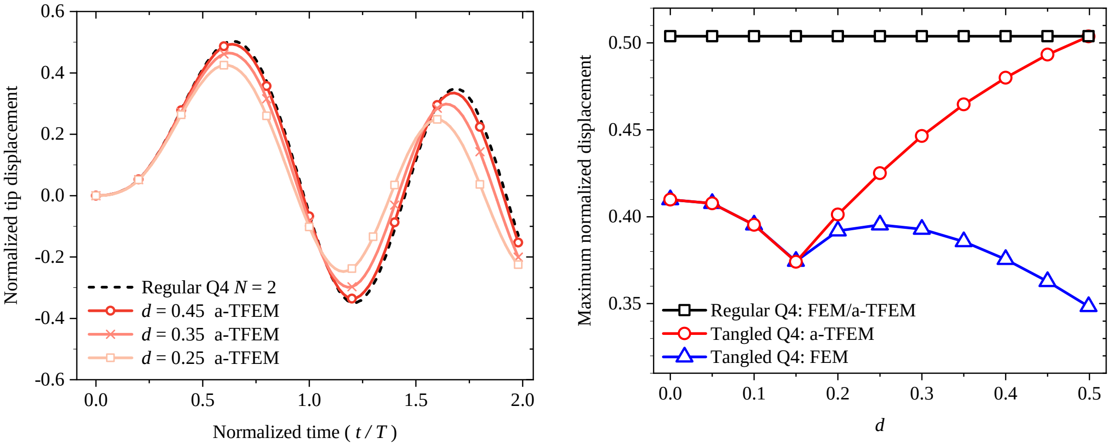


Figure 8.10: Comparison of a-TFEM and FEM solutions with N = 8 mesh using (a) Q4 and (b) Q9 elements.

2D cantilever beam: Degree of tangling

In the above study, the extent of tangling was fixed with $d = 0.4$. Next, let us study the effect of tangling parameter d for a fixed mesh size $N = 2$. Fig. 8.11a plots the a-TFEM solution for various values of d. Observe that as the tangling increases (i.e. as d increases), the a-TFEM solution approaches the result obtained by the corresponding regular mesh.

To compare the performance of FEM and a-TFEM, the maximum value of normalized tip displacement is plotted for $d \in [0, 0.499]$ in Fig. 8.11b. Observe that for $d < 0.2$, a-TFEM solution matches FEM solution for the tangled mesh. The Jacobian at all the Gauss points are positive and a-TFEM reduces to the standard FEM. For $d \geq 0.2$, Jacobian is negative at one or more Gauss points. As $d \in [0.2, 0.499]$ increases, a-TFEM results approach the results obtained by employing the regular mesh. On the other hand, FEM solutions increasingly move farther away from those obtained using regular mesh.



(a) (b)

Figure 8.11: Solutions obtained using Q4 mesh with mesh index $N = 2$ to study the effect of varying the extent of tangling.

3D cantilever: Mesh construction

Next, the 3D cantilever is discretized using regular and tangled meshes. A regular mesh with 60 × 3 × 3 elements is shown in Fig. 8.12.

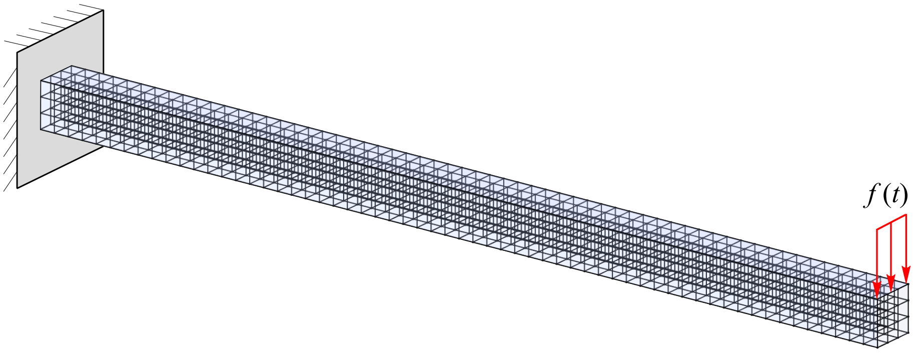


Figure 8.12: Regular H8 mesh with $N = 1$

The regular repeating unit consists of $3 \times 3 \times 3$ elements as shown in Fig. 8.13a. The tangled repeating unit is constructed by modifying the non-tangled unit in two steps:

1. Each element of the regular mesh is converted into a 2-element unit, as depicted in Fig. 8.13b, resulting in 54 elements. Positions of the new re-entrant nodes (nodes 9 and 10) are given as

$$x_3^{(9)} = x_3^{(1)}, \qquad x_i^{(9)} = x_i^{(1)} + (0.5 - d)\, s_i, \quad i = 1, 2 \tag{8.24a}$$

$$x_3^{(10)} = x_3^{(5)}, \qquad x_i^{(10)} = x_i^{(5)} + (0.6 - d)\, s_i, \quad i = 1, 2 \tag{8.24b}$$

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

2. Then the node B (highlighted in red in Fig. 8.13a) is moved, using the same parameter d, towards the left-hand side.

$$\boldsymbol{x}_d^{(B)} = \boldsymbol{x}^{(B)} - d \times [1.75 s_1 \quad 1.75 s_2 \quad 0.7 s_3]^\top. \tag{8.25a}$$

The value of the parameter d can be varied from 0 to 0.48. The front view of a resulting tangled mesh for $d = 0.4$ is illustrated in Fig. 8.13c.

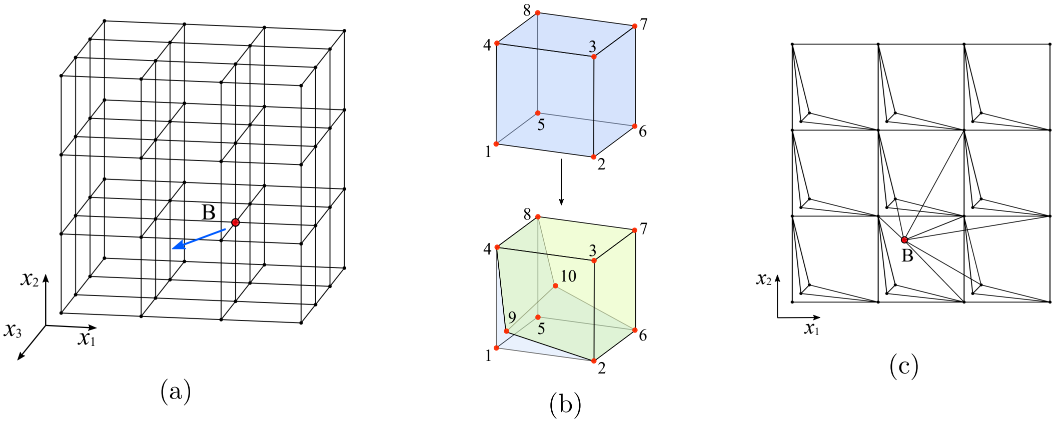


Figure 8.13: (a) Regular mesh (b) Non-tangled element converted to tangled element (c) Front view of the tangled mesh.

The tangled mesh unit (Fig. 8.13c) has 54 elements; out of which 28 elements are tangled. Fig. 8.14 illustrates some of the tangled elements. These elements are non-convex with non-planar faces.

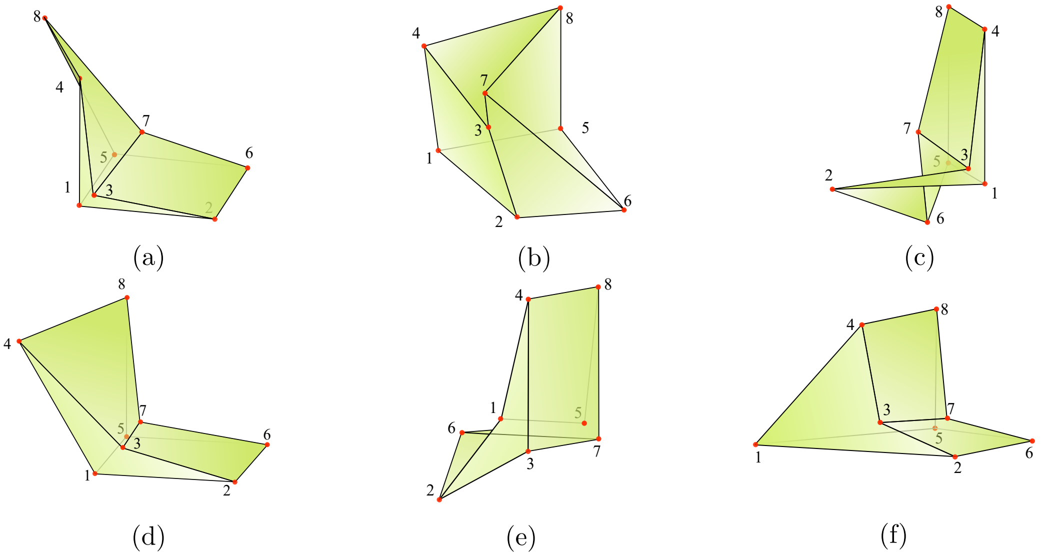


Figure 8.14: Some tangled elements present in the mesh shown in Fig. 8.13c

3D cantilever: Forced vibration

A forced vibration analysis of the 3D cantilever beam is performed as described in section 8.2.1. To discretize the domain, N number of cubic repeating units described above are arranged in the configuration of $(20N \times N \times N)$.

This problem is solved using the tangled meshes with $N = 1$ to 3 and $d = 0.4$. The response of the cantilever using a-TFEM is depicted in Fig. 8.15a. The normalized vertical displacement of the cantilever tip $v_A EI/L^3$ is plotted with respect to normalized time t/T. The reference solution is obtained by considering a regular (non-tangled) mesh with 67,500 elements. Next, for mesh with $N = 2$, Fig. 8.15b compares the solutions obtained by the three methods: FEM and a-TFEM using the tangled mesh and FEM using untangled mesh. Similar convergence and trends as in the 2D problem can be observed.

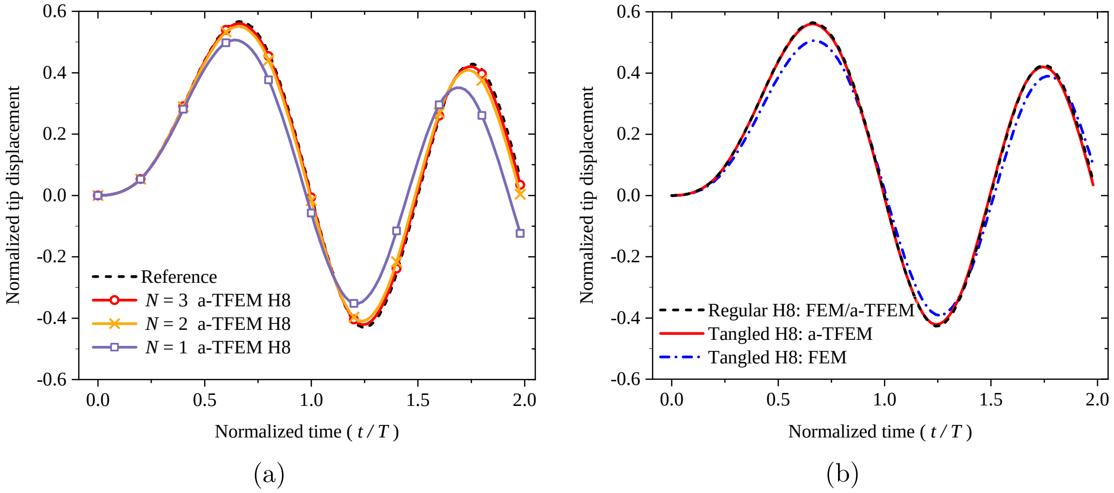
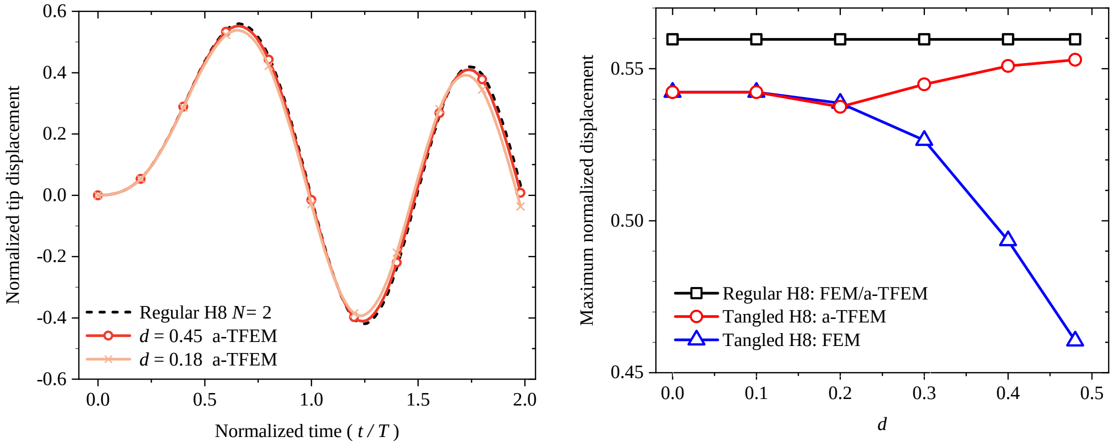


Figure 8.15: (a) Convergence for tangled hexahedral (H8) meshes using a-TFEM (b) Comparison of a-TFEM and FEM solutions obtained using hexahedral meshes with the size $N = 2$.

3D cantilever: Degree of tangling

In the above study, the extent of tangling was fixed with $d = 0.4$. Here, the effect of tangling on the computed solutions is studied by varying d from change 0 to 0.48 for a fixed mesh size, with $N = 2$. Once again, as d increases, the a-TFEM solution approaches the reference solution as illustrated in Fig. 8.16a. Next, the maximum normalized displacement obtained using a-TFEM (over tangled mesh) and FEM (obtained over regular and tangled mesh) is plotted in Fig. 8.16b. As seen in the 2D example, FEM deviates from the expected results with an increase in d, as opposed to a-TFEM.



(a)

(b)

Figure 8.16: Solutions with hexahedral (H8) $N = 2$ mesh to study the effect of varying the extent of tangling.

8.2.2 Spherical Shell with Damping

Next, the damping of a spherical shell subjected to a concentrated load is investigated as illustrated in Fig. 8.17. The problem has been previously investigated in [92]. The geometric parameters of the shell are as follows: inner radius = 12, thickness = 0.1, and outer chord radius $c = 2.29$. A concentrated load $f(t) = \cos(0.05t)$ is applied at the apex while the outer surface is fixed along the thickness. Rayleigh damping coefficients are $\alpha_1 = 0.005$, $\alpha_2 = 0.272$ while other material parameters are $E = 1.0$, $\nu = 0.3$, and $\rho = 1.0$.

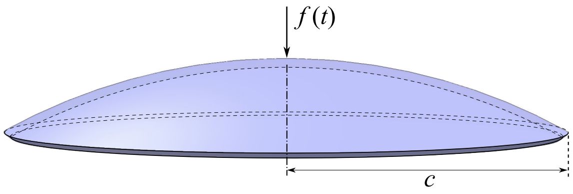


Figure 8.17: Spherical shell geometry

Due to axisymmetric nature of the shell, it is modeled in 2D as shown in Fig. 8.18. The mesh for this example is constructed by using the 4-element mesh as the basic repeating unit, as in Fig. 8.18. The extent of tangling can be varied by the parameter $d \in [0, 0.5]$. For $d = 0$, a regular mesh is obtained. As d increases, one out of the four elements gets tangled. In this experiment, tangled mesh is constructed with $d = 0.475$ and has 40×4 Q9 elements.

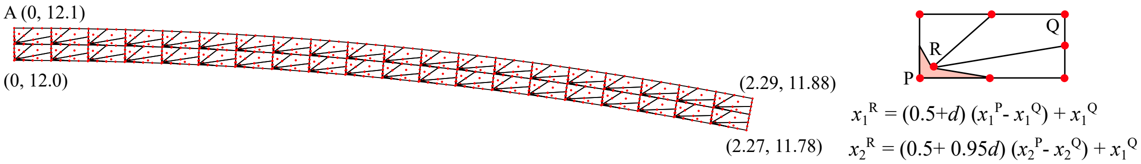

A (0, 12.1)
(0, 12.0)
(2.29, 11.88)
(2.27, 11.78)
P
R
Q
$x_1^R = (0.5+d)\,(x_1^P - x_1^Q) + x_1^Q$
$x_2^R = (0.5+ 0.95d)\,(x_2^P - x_2^Q) + x_1^Q$

Figure 8.18: Spherical shell with Q9 elements with straight edges.

The apex (point A in Fig. 8.18) displacement with respect to time is plotted in Fig. 8.19. Once again, the solution obtained using a-TFEM with the tangled mesh closely matches the solution obtained using the regular mesh.

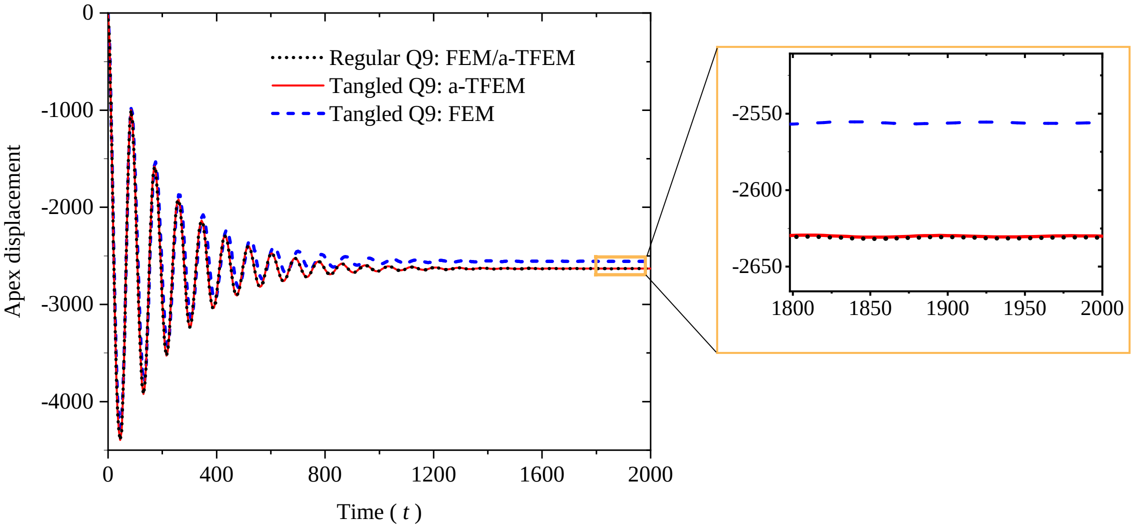


Figure 8.19: Transient responses of the spherical shell subjected to a harmonic loading.

8.2.3 3D Real-world Tangled Meshes: Free Vibration

Finally, real-world tangled meshes illustrated in Fig. 8.20 (provided in [103]) are considered. Tangled elements are highlighted in red color in Fig. 8.20a. Although it is feasible to untangle these meshes [103], the use of a-TFEM eliminates the need for untangling.

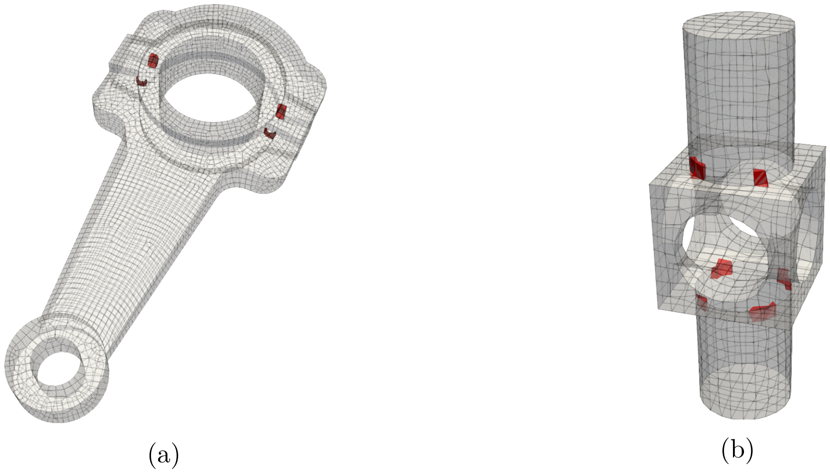


Figure 8.20: Tangled mesh of (a) linking rod and (b) block provided by [103]. Tangled elements are highlighted in red.

For free vibration analysis, the following material properties are considered: $E = 2.05 \times 10^7$, $\nu = 0.28$, and $\rho = 0.016308$. For the linking rod, the inner surface of the smaller hole is fixed, while the bottom face is fixed for the block. The first mode deformation obtained by employing a-TFEM is visualized in Fig. 8.21.

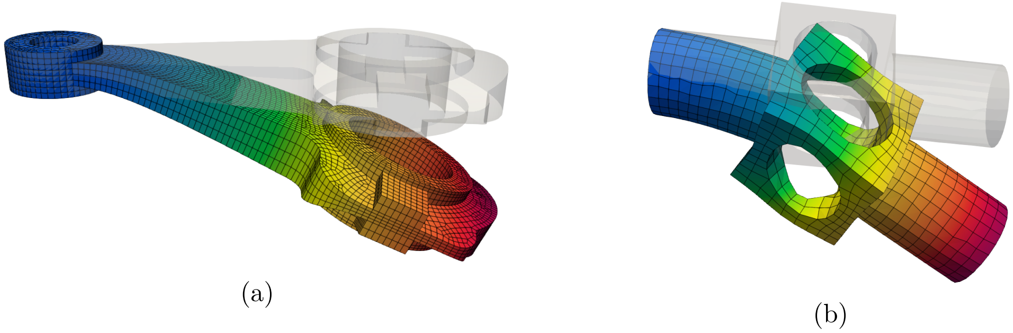


Figure 8.21: The first mode deformation for (a) linking rod and (b) block; obtained using a-TFEM.

Table 8.1 compares the first four natural frequencies obtained via a-TFEM using the tangled mesh with those obtained using the untangled mesh. It can be observed that the natural frequencies obtained from both methods are comparable, and the additional computational time required for a-TFEM is minimal. The time required to untangle the mesh is not included.

Table 8.1: Comparison of solutions over tangled and untangled meshes provided in [103].

Model [103]	Mesh	$\lvert \boldsymbol{J} \rvert_{\min}$	No. of Tangled Hexahedra	Natural frequency (Hz)				Time (s)
				1	2	3	4	
linking rod	tangled	-0.39	8/11316	6.37	15.54	28.82	68.99	17.17
	untangled	0.55	0/11316	6.36	15.53	28.79	68.91	17.14
block	tangled	-0.70	10/2520	1542	1544	5604	6539	3.55
	untangled	0.25	0/2520	1547	1548	5584	6731	3.52

Chapter 9

Conclusion

9.1 Contributions of the Thesis

Tangled meshes are conventionally considered unacceptable in FEM due to the erroneous results they produce. However, generating tangle-free meshes for complex geometries is often impossible, leading to the occurrence of tangled (non-convex) elements with negative Jacobian determinants. Consequently, many untangling methods have been proposed; however, untangling is not always achievable.

This thesis challenges the conventional view on tangled elements in FEM by exploring their potential inclusion in the simulations, thereby alleviating meshing and untangling challenges. To address this, a novel approach named the *Tangled Finite Element Method* (TFEM) is introduced. TFEM hinges on two fundamental principles: redefining the field in the tangled region to eliminate ambiguity and enforcing field compatibility for all tangled elements. By incorporating these principles, TFEM achieves accurate results and optimal convergence rates while requiring minimal changes to the existing FEM framework.

The thesis introduces three variations of TFEM: oriented TFEM (o-TFEM), isoparametric TFEM (i-TFEM), and accelerated-isoparametric TFEM (a-TFEM), each with distinct approaches to stiffness matrix computation. Among these, a-TFEM stands out as the recommended method, adeptly managing real-world tangled meshes, including cases involving self-penetrating elements, with minimal computational overhead.

Applicability of TFEM spans various element types, encompassing 4-node quadrilateral, higher-order elements like the 9-node quadrilateral and 6-node triangular, as well as 3D 8-node hexahedral elements. The method has been successfully employed to solve 2D and 3D linear and nonlinear elasticity, Poisson equations, and both free and forced vibration problems using tangled meshes. Several real-world tangled meshes are considered which further underlines the robustness and efficiency of TFEM.

The thesis demonstrates that TFEM effectively relaxes the tangle-free mesh constraint and obviates the necessity for untangling, opening a new paradigm in mesh generation. The implications on mesh generators are two-fold: firstly, they can focus on other meshing requirements, particularly geometric conformity for more reliable FEM results, without worrying about tangling. Secondly, the thesis demonstrates that tangled meshes, with the use of TFEM, can deliver comparable accuracy to their untangled counterparts. Since mesh quality metrics reflect the solution accuracy, the thesis challenges the traditional practice of assigning the lowest quality parameter values to the tangled elements. This thesis suggests that quality parameters can be independent of whether the element is tangled or not.

In essence, this thesis reevaluates the necessity of tangle-free meshes and demonstrates that tangled meshes are indeed suitable for FEM analysis with minimal modifications to the existing finite element formulation. TFEM emerges as a viable solution to handle tangled meshes, potentially simplifying mesh generation complexities and enabling more practical and efficient engineering analyses. With TFEM, formerly unacceptable tangled meshes can now be deemed acceptable, and mesh generators can leverage their existing algorithms without grappling with the creation of negative Jacobian elements. This shift liberates mesh generators to concentrate on geometric conformance, topological constraints, and other aspects of mesh quality, transforming the meshing landscape.

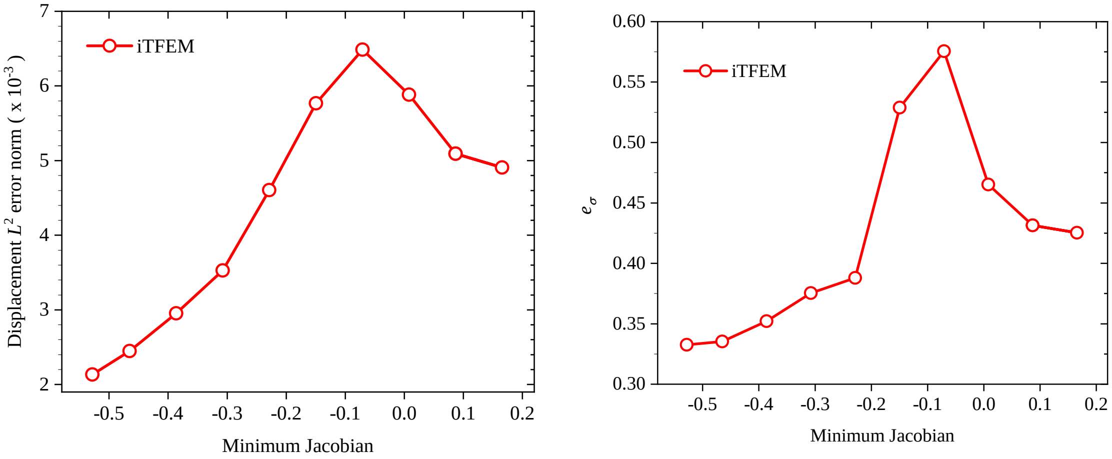
9.2 Future Work

While the thesis lays out the framework for TFEM and its variations, there remain promising opportunities and scope for improvement. Following are some research topics.

9.2.1 Mesh Quality for Tangled Elements

Numerous mesh quality indicators have been developed to assess distortion, skewness, and other factors. The goal of a good quality indicator is to provide insights into the accuracy of numerical methods [151]. 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. One of the most common quality indicators is the scaled Jacobian [85]. Tangled elements are deemed to be invalid, and/or assigned a quality of zero [85]. However, using TFEM, tangled meshes can provide comparable or even superior accuracy compared to regular meshes. This challenges the conventional definition of mesh quality indicators.

Typical plots of error (L^2 norm error in displacement and stress) as a function of the minimum Jacobian within the mesh are shown in Fig. 9.1 (see [134] for details). This particular plot is obtained for the 3D torsion problem described in Chapter 6 [134]. As one can observe, accuracy improves as we move away from zero Jacobian, whether in the positive or negative direction.



(a)

(b)

Figure 9.1: Plot of (a) displacement error and (b) stress error 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. Further research is needed since these findings are specific to the problem and mesh considered.

9.2.2 Application to Extreme Deformation

The thesis presents the TFEM framework for solving nonlinear elasticity problems based on the total Lagrangian formulation where the *initial* mesh is tangled. In practice, tangling can occur *during* FE simulations, even when the initial mesh is tangle-free. This is particularly prevalent in updated Lagrangian or ALE simulations involving large/extreme mesh movement. To address tangling challenge and prevent premature termination of simulations, a possible solution is to consider transitioning to TFEM when tangling occurs. By doing so, remeshing efforts can be minimized, leading to more efficient and uninterrupted simulations.

9.2.3 Extension to Isogeometric Analysis

There are promising opportunities beyond FEM, especially in isogeometric analysis (IGA). IGA can be considered as a is a numerical method for solving boundary value problems [70, 11]. In contrast to FEM, IGA is tightly integrated with the geometry, often leading to more accurate results, especially for curved geometries. Parameterization in IGA is equivalent to mesh generation in FEM [177] and is perhaps the most critical step since it affects all aspects of IGA [128, 176]. Parameterization means to find a mapping from a unit square (in parametric space) to the given domain.

The primary constraint for IGA to be effective is that the parameterization must be injective (invertible) [177]. In other words, the determinant of the Jacobian associated with the mapping must be positive. Fig. 9.2 illustrates a mapping that is invalid [58]. Similar to FEM, such invalid parameterization leads to erroneous results in IGA.

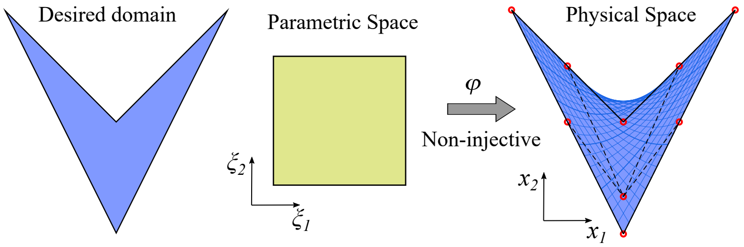


Figure 9.2: Parametric mapping from a unit square onto a computational domain.

Finding a suitable parameterization that is both injective and uniform is one of the main challenges in IGA. To quote a recent publication [30] " *(parameterization) is difficult and usually involves manual interaction, and has significantly hindered the development and application of IGA".*

Given that TFEM has shown promising results in handling tangled elements in FEM, it is worth exploring its extension to IGA. Developing such an approach could significantly improve the efficiency and applicability of IGA, especially in dealing with complex geometries.

Bibliography

[1] Muhammad Naeem Akram, Lei Si, and Guoning Chen. “An Embedded Polygon Strategy for Quality Improvement of 2D Quadrilateral Meshes with Boundaries.” In: *VISIGRAPP (1: GRAPP)*. 2021, pp. 177–184.

[2] W Annicchiarico and M Cerrolaza. “Finite elements, genetic algorithms and β-splines: a combined technique for shape optimization”. In: *Finite Elements in Analysis and Design* 33.2 (1999), pp. 125–141.

[3] W Annicchiarico and M Cerrolaza. “Structural shape optimization 3D finite-element models based on genetic algorithms and geometric modeling”. In: *Finite Elements in Analysis and Design* 37.5 (2001), pp. 403–415.

[4] Paola F Antonietti, Lourenço Beirão da Veiga, and Gianmarco Manzini. *The virtual element method and its applications*. Vol. 31. Springer Nature, 2022.

[5] Ferdinando Auricchio et al. “Mixed finite element methods”. In: *Encyclopedia of Computational Mechanics Second Edition* (2017), pp. 1–53.

[6] Owe Axelsson and Vincent Allan Barker. *Finite element solution of boundary value problems: theory and computation*. SIAM, 2001.

[7] Georgios Balafas. “Polyhedral mesh generation for CFD-analysis of complex structures”. In: *Diplomityö. Münchenin teknillinen yliopisto* (2014).

[8] James R Barber. *Elasticity*. Springer, 2002.

[9] Klaus-Jürgen Bathe. *Finite element procedures*. Klaus-Jurgen Bathe, 2006.

[10] Klaus-Jurgen Bathe and Anil Chaudhary. “A solution method for planar and axisymmetric contact problems”. In: *International Journal for Numerical Methods in Engineering* 21.1 (1985), pp. 65–88.

[11] Yuri Bazilevs et al. “Isogeometric analysis using T-splines”. In: *Computer Methods in Applied Mechanics and Engineering* 199.5-8 (2010), pp. 229–263.

[12] P-A Beaufort et al. “Hex me if you can”. In: *Computer graphics forum*. Vol. 41. 5. Wiley Online Library. 2022, pp. 125–134.

[13] L Beirão da Veiga et al. “Basic principles of virtual element methods”. In: *Mathematical Models and Methods in Applied Sciences* 23.01 (2013), pp. 199–214.

[14] Giovanni Belingardi and Jovan Obradovic. “Numerical crash analysis of composite racing car front impact attenuator by use of explicit FEM codes”. In: *Applied Mechanics and Materials* 82 (2011), pp. 290–295.

[15] Ted Belytschko, Robert Gracie, and Giulio Ventura. “A review of extended/generalized finite element methods for material modeling”. In: *Modelling and Simulation in Materials Science and Engineering* 17.4 (2009), p. 043001.

[16] Ted Belytschko, Yun Yun Lu, and Lei Gu. “Element-free Galerkin methods”. In: *International journal for numerical methods in engineering* 37.2 (1994), pp. 229–256.

[17] Sanjukta Bhowmick and Suzanne M Shontz. “Towards high-quality, untangled meshes via a force-directed graph embedding approach”. In: *Procedia Computer Science* 1.1 (2010), pp. 357–366.

[18] Marco Evangelos Biancolini et al. “Fast interactive CFD evaluation of hemodynamics assisted by RBF mesh morphing and reduced order models: The case of aTAA modelling”. In: *International Journal on Interactive Design and Manufacturing (IJIDeM)* 14 (2020), pp. 1227–1238.

[19] Joseph E Bishop. “A displacement-based finite element formulation for general polyhedra using harmonic shape functions”. In: *International Journal for Numerical Methods in Engineering* 97.1 (2014), pp. 1–31.

[20] Serena Bonaretti et al. “Image-based vs. mesh-based statistical appearance models of the human femur: implications for finite element simulations”. In: *Medical engineering & physics* 36.12 (2014), pp. 1626–1635.

[21] Allan F Bower. *Applied mechanics of solids.* CRC press, 2009.

[22] Matteo Bracci et al. “HexaLab. net: An online viewer for hexahedral meshes”. In: *Computer-Aided Design* 110 (2019), pp. 24–36.

[23] Katia Capellini et al. “Computational fluid dynamic study for aTAA hemodynamics: an integrated image-based and radial basis functions mesh morphing approach”. In: *Journal of biomechanical engineering* 140.11 (2018), p. 111007.

[24] Nicholas J Carpenter, Robert L Taylor, and Michael G Katona. “Lagrange constraints for transient finite element surface contact”. In: *International journal for numerical methods in engineering* 32.1 (1991), pp. 103–128.

[25] Marco Lo Cascio, Alberto Milazzo, and Ivano Benedetti. “Virtual element method for computational homogenization of composite and heterogeneous materials”. In: *Composite Structures* 232 (2020), p. 111523.

[26] Ubaldo Cella, Corrado Groth, and Marco Evangelos Biancolini. “Geometric parameterization strategies for shape optimization using RBF mesh morphing”. In: *Advances on Mechanics, Design Engineering and Manufacturing: Proceedings of the International Joint Conference on Mechanics, Design Engineering & Advanced Manufacturing (JCM 2016), 14-16 September, 2016, Catania, Italy.* Springer. 2017, pp. 537–545.

[27] Song Cen et al. “An unsymmetric 4-node, 8-DOF plane membrane element perfectly breaking through MacNeal’s theorem”. In: *International Journal for Numerical Methods in Engineering* 103.7 (2015), pp. 469–500.

[28] Gang Chen et al. “Efficient multidisciplinary aerodynamic optimization design based on discrete adjoint method”. In: *54th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*. 2013, p. 1504.

[29] Jiun-Shyan Chen et al. “A stabilized conforming nodal integration for Galerkin mesh-free methods”. In: *International journal for numerical methods in engineering* 50.2 (2001), pp. 435–466.

[30] Long Chen et al. “Constructing volumetric parameterization based on directed graph simplification of L1 polycube structure from complex shapes”. In: *Computer Methods in Applied Mechanics and Engineering* 351 (2019), pp. 422–440.

[31] Tsu-Chien Cheu. “Procedures for shape optimization of gas turbine disks”. In: *Turbo Expo: Power for Land, Sea, and Air*. Vol. 79085. American Society of Mechanical Engineers. 1990, V005T14A018.

[32] H Chi, L Beirão Da Veiga, and GH Paulino. “Some basic formulations of the virtual element method (VEM) for finite deformations”. In: *Computer Methods in Applied Mechanics and Engineering* 318 (2017), pp. 148–192.

[33] Heng Chi et al. “Polygonal finite elements for finite elasticity”. In: *International Journal for Numerical Methods in Engineering* 101.4 (2015), pp. 305–328.

[34] Myung-Jin Choi and Seonho Cho. “A mesh regularization scheme to update internal control points for isogeometric shape design optimization”. In: *Computer Methods in Applied Mechanics and Engineering* 285 (2015), pp. 694–713.

[35] Mertcan Cihan et al. “Virtual element formulation for finite strain elastodynamics”. In: *arXiv preprint arXiv:2002.02680* (2020).

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

[37] Robert D Cook. “Improved two-dimensional finite element”. In: *Journal of the Structural Division* 100.9 (1974), pp. 1851–1863.

[38] *CUBIT*. `https://cubit.sandia.gov/`. 2022.

[39] *CUBIT*. `https://www.spatial.com/products/3d-precise-mesh`.

[40] KY Dai and GR Liu. “Free and forced vibration analysis using the smoothed finite element method (SFEM)”. In: *Journal of Sound and Vibration* 301.3-5 (2007), pp. 803–820.

[41] Josh Danczyk and Krishnan Suresh. “Finite element analysis over tangled meshes”. In: *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. Vol. 45011. Citeseer. 2012, pp. 89–95.

[42] Josh Danczyk and Krishnan Suresh. “Finite element analysis over tangled simplicial meshes: Theory and implementation”. In: *Finite Elements in Analysis and Design* 70 (2013), pp. 57–67.

[43] ML De Bellis, P Wriggers, and B Hudobivnik. “Serendipity virtual element formulation for nonlinear elasticity”. In: *Computers & Structures* 223 (2019), p. 106094.

[44] Bensingh Dhas et al. “A novel four-field mixed FE approximation for Kirchhoff rods using Cartan’s moving frames”. In: *Computer Methods in Applied Mechanics and Engineering* (2022), p. 115094.

[45] Lorenzo Diazzi and Marco Attene. “Convex polyhedral meshing for robust solid modeling”. In: *ACM Transactions on Graphics (TOG)* 40.6 (2021), pp. 1–16.

[46] Franz-Josef Ertl, Guido Dhondt, and Kai-Uwe Bletzinger. “Vertex assigned morphing for parameter free shape optimization of 3-dimensional solid structures”. In: *Computer Methods in Applied Mechanics and Engineering* 353 (2019), pp. 86–106.

[47] José Marıa Escobar et al. “Simultaneous untangling and smoothing of tetrahedral meshes”. In: *Computer Methods in Applied Mechanics and Engineering* 192.25 (2003), pp. 2775–2787.

[48] Xianzhong Fang et al. “All-hex meshing using closed-form induced polycube”. In: *ACM Transactions on Graphics (TOG)* 35.4 (2016), pp. 1–9.

[49] Charbel Farhat and Francois-Xavier Roux. “A method of finite element tearing and interconnecting and its parallel solution algorithm”. In: *International journal for numerical methods in engineering* 32.6 (1991), pp. 1205–1227.

[50] Michael Floater, Andrew Gillette, and N Sukumar. “Gradient bounds for Wachspress coordinates on polytopes”. In: *SIAM Journal on Numerical Analysis* 52.1 (2014), pp. 515–532.

[51] Michael S Floater. “Mean value coordinates”. In: *Computer aided geometric design* 20.1 (2003), pp. 19–27.

[52] Michael S Floater, Géza Kós, and Martin Reimers. “Mean value coordinates in 3D”. In: *Computer Aided Geometric Design* 22.7 (2005), pp. 623–631.

[53] Arun L Gain, Cameron Talischi, and Glaucio H Paulino. “On the virtual element method for three-dimensional linear elasticity problems on arbitrary polyhedral meshes”. In: *Computer Methods in Applied Mechanics and Engineering* 282 (2014), pp. 132–160.

[54] Xifeng Gao, Zhigang Deng, and Guoning Chen. “Hexahedral mesh re-parameterization from aligned base-complex”. In: *ACM Transactions on Graphics (TOG)* 34.4 (2015), pp. 1–10.

[55] Xifeng Gao, Hanxiao Shen, and Daniele Panozzo. “Feature Preserving Octree-Based Hexahedral Meshing”. In: *Computer graphics forum*. Vol. 38. 5. Wiley Online Library. 2019, pp. 135–149.

[56] Rao V Garimella, Jibum Kim, and Markus Berndt. “Polyhedral mesh generation and optimization for non-manifold domains”. In: *Proceedings of the 22nd International Meshing Roundtable*. Springer. 2014, pp. 313–330.

[57] Robert A Gingold and Joseph J Monaghan. “Smoothed particle hydrodynamics: theory and application to non-spherical stars”. In: *Monthly notices of the royal astronomical society* 181.3 (1977), pp. 375–389.

[58] Jens Gravesen et al. “Planar parametrization in isogeometric analysis”. In: *International conference on mathematical methods for curves and surfaces*. Springer. 2012, pp. 189–212.

[59] James Gregson, Alla Sheffer, and Eugene Zhang. "All-hex mesh generation via volumetric polycube deformation". In: *Computer graphics forum*. Vol. 30. 5. Wiley Online Library. 2011, pp. 1407–1416.

[60] YuanTong Gu and Gui-Rong Liu. "A meshless local Petrov-Galerkin (MLPG) method for free and forced vibration analyses for solids". In: *Computational Mechanics* 27 (2001), pp. 188–198.

[61] Hao-Xiang Guo et al. "Cut-enhanced PolyCube-maps for feature-aware all-hex meshing". In: *ACM Transactions on Graphics (TOG)* 39.4 (2020), pp. 106–1.

[62] Raphael T Haftka and Ramana V Grandhi. "Structural shape optimization—a survey". In: *Computer methods in applied mechanics and engineering* 57.1 (1986), pp. 91–106.

[63] Si Hang. "TetGen, a Delaunay-based quality tetrahedral mesh generator". In: *ACM Trans. Math. Softw* 41.2 (2015), p. 11.

[64] Peter Hartley and Ian Pillinger. "Numerical simulation of the forging process". In: *Computer methods in applied mechanics and engineering* 195.48-49 (2006), pp. 6676–6690.

[65] Tao He. "A stabilized cell-based smoothed finite element method against severe mesh distortion in non-Newtonian fluid–structure interaction". In: *International Journal for Numerical Methods in Engineering* 123.9 (2022), pp. 2162–2184.

[66] Michael T Heath. *Scientific Computing: An Introductory Survey, Revised Second Edition*. SIAM, 2018.

[67] Jin Huang et al. "Boundary aligned smooth 3D cross-frame field". In: *ACM transactions on graphics (TOG)* 30.6 (2011), pp. 1–8.

[68] Qing Huang et al. "Untangling all-hex meshes via adaptive boundary optimization". In: *Graphical Models* 121 (2022), p. 101136.

[69] Thomas JR Hughes. *The finite element method: linear static and dynamic finite element analysis*. Courier Corporation, 2012.

[70] Thomas JR Hughes, John A Cottrell, and Yuri Bazilevs. "Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement". In: *Computer methods in applied mechanics and engineering* 194.39-41 (2005), pp. 4135–4195.

[71] Thomas JR Hughes et al. "A finite element method for a class of contact-impact problems". In: *Computer methods in applied mechanics and engineering* 8.3 (1976), pp. 249–276.

[72] SH Huo et al. "A smoothed finite element method for octree-based polyhedral meshes with large number of hanging nodes and irregular elements". In: *Computer Methods in Applied Mechanics and Engineering* 359 (2020), p. 112646.

[73] Daniel van Huyssteen and Batmanathan Dayanand Reddy. "A virtual element method for isotropic hyperelasticity". In: *Computer Methods in Applied Mechanics and Engineering* 367 (2020), p. 113134.

[74] Daniel van Huyssteen et al. "On mesh refinement procedures for the virtual element method for two-dimensional elastic problems". In: *Computer Methods in Applied Mechanics and Engineering* 393 (2022), p. 114849.

[75] M Hasan Imam. “Three-dimensional shape optimization”. In: *International Journal for Numerical Methods in Engineering* 18.5 (1982), pp. 661–673.

[76] Geoffrey Irving, Joseph Teran, and Ronald Fedkiw. “Invertible finite elements for robust simulation of large deformation”. In: *Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*. 2004, pp. 131–140.

[77] Moustapha Jadayel and Farbod Khameneifar. “Improving geometric accuracy of 3D printed parts using 3D metrology feedback and mesh morphing”. In: *Journal of Manufacturing and Materials Processing* 4.4 (2020), p. 112.

[78] Gang-Won Jang, Kyung Joo Kim, and Yoon Young Kim. “Integrated topology and shape optimization software for compliant MEMS mechanism design”. In: *Advances in Engineering Software* 39.1 (2008), pp. 1–14.

[79] Gang-Won Jang, Yoon Young Kim, and Kyung K Choi. “Remesh-free shape optimization using the wavelet-Galerkin method”. In: *International journal of solids and structures* 41.22-23 (2004), pp. 6465–6483.

[80] Chen Jiang et al. “Selective smoothed finite element methods for extremely large deformation of anisotropic incompressible bio-tissues”. In: *International Journal for Numerical Methods in Engineering* 99.8 (2014), pp. 587–610.

[81] Tengfei Jiang et al. “Frame field singularity correctionfor automatic hexahedralization”. In: *IEEE Transactions on Visualization and Computer Graphics* 20.8 (2013), pp. 1189–1199.

[82] Jibum Kim, Junhyeok Choi, and Woochul Kang. “A data-driven approach for simultaneous mesh untangling and smoothing using pointer networks”. In: *IEEE Access* 8 (2020), pp. 70329–70342.

[83] Jibum Kim, Thap Panitanarak, and Suzanne M Shontz. “A multiobjective mesh optimization framework for mesh quality improvement and mesh untangling”. In: *International journal for numerical methods in engineering* 94.1 (2013), pp. 20–42.

[84] Patrick M Knupp. “A method for hexahedral mesh shape optimization”. In: *International journal for numerical methods in engineering* 58.2 (2003), pp. 319–332.

[85] Patrick M Knupp. “Algebraic mesh quality metrics for unstructured initial meshes”. In: *Finite Elements in Analysis and Design* 39.3 (2003), pp. 217–241.

[86] Patrick M Knupp. “Hexahedral and tetrahedral mesh untangling”. In: *Engineering with Computers* 17.3 (2001), pp. 261–268.

[87] Patrick M Knupp. “Winslow smoothing on two-dimensional unstructured meshes”. In: *Engineering with Computers* 15 (1999), pp. 263–268.

[88] Richard Kramer et al. “An extended finite element method with algebraic constraints (XFEM-AC) for problems with weak discontinuities”. In: *Computer Methods in Applied Mechanics and Engineering* 266 (2013), pp. 70–80.

[89] Richard MJ Kramer et al. “Algebraically constrained extended edge element method (eXFEM-AC) for resolution of multi-material cells”. In: *Journal of Computational Physics* 276 (2014), pp. 596–612.

[90] Richard MJ Kramer et al. “Formulation and computation of dynamic, interface-compatible Whitney complexes in three dimensions”. In: *Journal of Computational Physics* 359 (2018), pp. 45–76.

[91] Siddhant Kumar, Kostas Danas, and Dennis M Kochmann. “Enhanced local maximum-entropy approximation for stable meshfree simulations”. In: *Computer Methods in Applied Mechanics and Engineering* 344 (2019), pp. 858–886.

[92] Y Li and GR Liu. “A novel node-based smoothed finite element method with linear strain fields for static, free and forced vibration analyses of solids”. In: *Applied Mathematics and Computation* 352 (2019), pp. 30–58.

[93] Yufei Li et al. “All-hex meshing using singularity-restricted field”. In: *ACM Transactions on Graphics (TOG)* 31.6 (2012), pp. 1–11.

[94] Zhi Li et al. “Hyperelastic finite deformation analysis with the unsymmetric finite element method containing homogeneous solutions of linear elasticity”. In: *International Journal for Numerical Methods in Engineering* 121.16 (2020), pp. 3702–3721.

[95] Hongwei Lin et al. “Quality guaranteed all-hex mesh generation by a constrained volume iterative fitting algorithm”. In: *Computer-Aided Design* 67 (2015), pp. 107–117.

[96] GR Liu, KY Dai, and THOI T Nguyen. “A smoothed finite element method for mechanics problems”. In: *Computational Mechanics* 39.6 (2007), pp. 859–877.

[97] GR Liu, T Nguyen-Thoi, and KY Lam. “An edge-based smoothed finite element method (ES-FEM) for static, free and forced vibration analyses of solids”. In: *Journal of Sound and Vibration* 320.4-5 (2009), pp. 1100–1130.

[98] Heng Liu et al. “Singularity-constrained octahedral fields for hexahedral meshing.” In: *ACM Trans. Graph.* 37.4 (2018), pp. 93–1.

[99] Marco Livesu. “cinolib: a generic programming header only C++ library for processing polygonal and polyhedral meshes”. In: *Transactions on Computational Science XXXIV* (2019), pp. 64–76.

[100] Marco Livesu, Luca Pitzalis, and Gianmarco Cherchi. “Optimal dual schemes for adaptive grid based hexmeshing”. In: *ACM Transactions on Graphics (TOG)* 41.2 (2021), pp. 1–14.

[101] Marco Livesu et al. “Loopycuts: Practical feature-preserving block decomposition for strongly hex-dominant meshing”. In: *ACM Transactions on Graphics (TOG)* 39.4 (2020), pp. 121–1.

[102] Marco Livesu et al. “Polycut: Monotone graph-cuts for polycube base-complex construction”. In: *ACM Transactions on Graphics (TOG)* 32.6 (2013), pp. 1–12.

[103] Marco Livesu et al. “Practical hex-mesh optimization via edge-cone rectification”. In: *ACM Transactions on Graphics (TOG)* 34.4 (2015), pp. 1–11.

[104] Daniel SH Lo. *Finite element mesh generation*. CRC Press, 2014.

[105] Ezequiel J López, Norberto M Nigro, and Mario A Storti. “Simultaneous untangling and smoothing of moving grids”. In: *International Journal for Numerical Methods in Engineering* 76.7 (2008), pp. 994–1019.

[106] Jean Hsiang-Chun Lu, William Roshan Quadros, and Kenji Shimada. “Evaluation of user-guided semi-automatic decomposition tool for hexahedral mesh generation”. In: *Journal of Computational Design and Engineering* 4.4 (2017), pp. 330–338.

[107] Ru-Xia Ma et al. “Extension of the unsymmetric 8-node hexahedral solid element US-ATFH8 to 3D hyper-elastic finite deformation analysis”. In: *International Journal for Numerical Methods in Engineering* 123.23 (2022), pp. 5749–5778.

[108] Manish Mandad et al. “Intrinsic mixed-integer polycubes for hexahedral meshing”. In: *Computer Aided Geometric Design* 94 (2022), p. 102078.

[109] Gianmarco Manzini, Alessandro Russo, and N Sukumar. “New perspectives on polygonal and polyhedral finite element methods”. In: *Mathematical Models and Methods in Applied Sciences* 24.08 (2014), pp. 1665–1699.

[110] Loïc 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.

[111] RA Naqib, A Zureick, and KM Will. “Practical considerations in two-dimensional shape optimization of elastic continuum”. In: *Computers & structures* 61.2 (1996), pp. 315–330.

[112] Sundararajan Natarajan, Stéphane PA Bordas, and Ean Tat Ooi. “Virtual and smoothed finite elements: a connection and its application to polygonal/polyhedral finite element methods”. In: *International Journal for Numerical Methods in Engineering* 104.13 (2015), pp. 1173–1199.

[113] Nathan M Newmark. “A method of computation for structural dynamics”. In: *Journal of the engineering mechanics division* 85.3 (1959), pp. 67–94.

[114] Vinh Phu Nguyen et al. “Meshless methods: a review and computer implementation aspects”. In: *Mathematics and computers in simulation* 79.3 (2008), pp. 763–813.

[115] Hau Nguyen Ngoc. “Three-dimensional polyhedral finite element analysis of solid revetment block”. PhD thesis. Ghent University, 2021.

[116] T Nguyen-Thoi et al. “A face-based smoothed finite element method (FS-FEM) for 3D linear and geometrically non-linear solid mechanics problems using 4-node tetrahedral elements”. In: *International journal for numerical methods in Engineering* 78.3 (2009), pp. 324–353.

[117] Matthias Nieser, Ulrich Reitebuch, and Konrad Polthier. “Cubecover–parameterization of 3d volumes”. In: *Computer graphics forum*. Vol. 30. 5. Wiley Online Library. 2011, pp. 1397–1406.

[118] Hiroaki Nishikawa. “Uses of zero and negative volume elements for node-centered edge-based discretization”. In: *23rd AIAA Computational Fluid Dynamics Conference*. 2017, p. 4295.

[119] Wayne Oaks and Stefano Paoletti. “Polyhedral Mesh Generation.” In: *IMR*. 2000, pp. 57–67.

[120] John Tinsley Oden and Junuthula Narasimha Reddy. *Variational methods in theoretical mechanics*. Springer Science & Business Media, 2012.

[121] Eugenio Oñate et al. “A finite point method in computational mechanics. Applications to convective transport and fluid flow”. In: *International journal for numerical methods in engineering* 39.22 (1996), pp. 3839–3866.

[122] Steven J Owen and Tim R Shelton. “Evaluation of grid-based hex meshes for solid mechanics”. In: *Engineering with Computers* 31.3 (2015), pp. 529–543.

[123] Kyoungsoo Park, Heng Chi, and Glaucio H Paulino. “On nonconvex meshes for elastodynamics using virtual element methods with explicit time integration”. In: *Computer Methods in Applied Mechanics and Engineering* 356 (2019), pp. 669–684.

[124] Glaucio H Paulino and Arun L Gain. “Bridging art and engineering using Escher-based virtual elements”. In: *Structural and Multidisciplinary Optimization* 51.4 (2015), pp. 867–883.

[125] Jeanne Pellerin, Amaury Johnen, and Jean-Francois Remacle. “Identifying combinations of tetrahedra into hexahedra: a vertex based strategy”. In: *Procedia engineering* 203 (2017), pp. 2–13.

[126] Theodore HH Pian. “Finite element formulation by variational principles with relaxed continuity requirements”. In: *The mathematical foundations of the finite element method with applications to partial differential equations*. Elsevier, 1972, pp. 671–687.

[127] Nico Pietroni et al. “Hex-mesh generation and processing: a survey”. In: *ACM Transactions on Graphics (TOG)* (2022).

[128] Elisabeth Pilgerstorfer and Bert Jüttler. “Bounding the influence of domain parameterization and knot spacing on numerical stability in Isogeometric Analysis”. In: *Computer Methods in Applied Mechanics and Engineering* 268 (2014), pp. 589–613.

[129] Bhagyashree Prabhune, Saketh Sridhara, and Krishnan Suresh. “Tangled finite element method for handling concave elements in quadrilateral meshes”. In: *International Journal for Numerical Methods in Engineering* 123.7 (2022), pp. 1576–1605.

[130] Bhagyashree Prabhune and Krishnan Suresh. “A computationally efficient isoparametric tangled finite element method for handling inverted quadrilateral and hexahedral elements”. In: *Computer Methods in Applied Mechanics and Engineering* 405 (2023), p. 115897. ISSN: 0045-7825.

[131] Bhagyashree Prabhune and Krishnan Suresh. “An isoparametric tangled finite element method for handling higher-order elements with negative Jacobian”. In: *Computational Mechanics* (2023), pp. 1–18.

[132] Bhagyashree Prabhune and Krishnan Suresh. “Free and Forced Vibration Analysis over Meshes with Tangled (Non-Convex) Elements”. In: *Submitted* (2023).

[133] Bhagyashree Prabhune and Krishnan Suresh. “Isoparametric Tangled Finite Element Method for Nonlinear Elasticity”. In: *arXiv preprint arXiv:2303.10799* (2023).

[134] Bhagyashree Prabhune and Krishnan Suresh. “On why mesh untangling may not be required”. In: *Submitted* (2023).

[135] Bhagyashree Prabhune and Krishnan Suresh. “Towards tangled finite element analysis over partially inverted hexahedral elements”. In: *arXiv preprint arXiv:2207.03905* (2022).

[136] Tikeshwar Prasad. “A comparative study of mesh smoothing methods with flipping in 2D and 3D”. PhD thesis. Rutgers University-Camden Graduate School, 2018.

[137] S Rajendran. “A technique to develop mesh-distortion immune finite elements”. In: *Computer Methods in Applied Mechanics and Engineering* 199.17-20 (2010), pp. 1044–1063.

[138] S Rajendran and KM Liew. “A novel unsymmetric 8-node plane element immune to mesh distortion under a quadratic displacement field”. In: *International Journal for Numerical Methods in Engineering* 58.11 (2003), pp. 1713–1748.

[139] Alexander Rand, Andrew Gillette, and Chandrajit Bajaj. “Interpolation error estimates for mean value coordinates over convex polygons”. In: *Advances in computational mathematics* 39.2 (2013), pp. 327–347.

[140] Maxence Reberol et al. “Robust topological construction of all-hexahedral boundary layer meshes”. In: *ACM Transactions on Mathematical Software* 49.1 (2023), pp. 1–32.

[141] Junuthula Narasimha Reddy. *An Introduction to Nonlinear Finite Element Analysis Second Edition: with applications to heat transfer, fluid mechanics, and solid mechanics*. OUP Oxford, 2014.

[142] Xevi Roca, Abel Gargallo-Peiró, and Josep Sarrate. “Defining quality measures for high-order planar triangles and curved mesh generation”. In: *Proceedings of the 20th International Meshing Roundtable*. Springer. 2012, pp. 365–383.

[143] Eloi Ruiz-Gironés, Josep Sarrate, and Xevi Roca. “Generation of curved high-order meshes with optimal quality and geometric accuracy”. In: *Procedia engineering* 163 (2016), pp. 315–327.

[144] Eloi Ruiz-Gironés et al. “Simultaneous untangling and smoothing of quadrilateral and hexahedral meshes using an object-oriented framework”. In: *Advances in Engineering Software* 80 (2015), pp. 12–24.

[145] Filippo Salmoiraghi et al. “Free-form deformation, mesh morphing and reduced-order methods: enablers for efficient aerodynamic shape optimisation”. In: *International Journal of Computational Fluid Dynamics* 32.4-5 (2018), pp. 233–247.

[146] Josep Sarrate and Antonio Huerta. “Efficient unstructured quadrilateral mesh generation”. In: *International journal for numerical methods in engineering* 49.10 (2000), pp. 1327–1350.

[147] Shankar P Sastry and Suzanne M Shontz. “A parallel log-barrier method for mesh quality improvement and untangling”. In: *Engineering with Computers* 30 (2014), pp. 503–515.

[148] Robert Schneiders and Rolf Bünten. “Automatic generation of hexahedral finite element meshes”. In: *Computer Aided Geometric Design* 12.7 (1995), pp. 693–707.

[149] Jason F Shepherd and Chris R Johnson. “Hexahedral mesh generation constraints”. In: *Engineering with Computers* 24.3 (2008), pp. 195–213.

[150] Jonathan Richard Shewchuk. “What is a good linear element? Interpolation, conditioning, and quality measures.” In: *IMR*. 2002, pp. 115–126.

[151] Tommaso Sorgente et al. “A Survey of Indicators for Mesh Quality Assessment”. In: *Computer Graphics Forum*. Vol. 42. 2. Wiley Online Library. 2023, pp. 461–483.

[152] Tommaso Sorgente et al. “Polyhedral mesh quality indicator for the virtual element method”. In: *Computers & Mathematics with Applications* 114 (2022), pp. 151–160.

[153] Akkaram Srikanth and Nicholas Zabaras. “An updated Lagrangian finite element sensitivity analysis of large deformations using quadrilateral elements”. In: *International Journal for Numerical Methods in Engineering* 52.10 (2001), pp. 1131–1163.

[154] Matthew L Staten et al. “A comparison of mesh morphing methods for 3D shape optimization”. In: *Proceedings of the 20th international meshing roundtable*. Springer, 2011, pp. 293–311.

[155] Mike Stees, Myra Dotzel, and Suzanne M Shontz. “Untangling high-order meshes based on signed angles”. In: *Proceedings of the 28th International Meshing Roundtable* (2020).

[156] Mike Stees and Suzanne M Shontz. “An Angular Approach to Untangling High-Order Curvilinear Triangular Meshes”. In: *International Meshing Roundtable*. Springer. 2018, pp. 327–342.

[157] Alexey Stomakhin et al. “Energetically consistent invertible elasticity”. In: *Proceedings of the 11th ACM SIGGRAPH/Eurographics conference on Computer Animation*. 2012, pp. 25–32.

[158] N Sukumar and A Tabarraei. “Conforming polygonal finite elements”. In: *International Journal for Numerical Methods in Engineering* 61.12 (2004), pp. 2045–2066.

[159] Kenshi Takayama. “Dual sheet meshing: An interactive approach to robust hexahedralization”. In: *Computer graphics forum*. Vol. 38. 2. Wiley Online Library. 2019, pp. 37–48.

[160] Cameron Talischi et al. “Gradient correction for polygonal and polyhedral finite elements”. In: *International Journal for Numerical Methods in Engineering* 102.3-4 (2015), pp. 728–747.

[161] Marco Tarini et al. “Practical quad mesh simplification”. In: *Computer Graphics Forum*. Vol. 29. 2. Wiley Online Library. 2010, pp. 407–418.

[162] SP Timoshenko and JN Goodier. *Theory of Elasticity 3rd ed., 567*. 1970.

[163] Maximilian Tomac and David Eller. “From geometry to CFD grids—an automated approach for conceptual design”. In: *Progress in Aerospace Sciences* 47.8 (2011), pp. 589–596.

[164] Pin Tong. “New displacement hybrid finite element models for solid continua”. In: *International Journal for Numerical Methods in Engineering* 2.1 (1970), pp. 73–83.

[165] Thomas Toulorge et al. “Robust untangling of curvilinear meshes”. In: *Journal of Computational Physics* 254 (2013), pp. 8–26.

[166] Bhavik D Upadhyay, Sunil S Sonigra, and Sachin D Daxini. “Numerical analysis perspective in structural shape optimization: A review post 2000”. In: *Advances in Engineering Software* 155 (2021), p. 102992.

[167] Vasileios Vavourakis et al. “Assessment of remeshing and remapping strategies for large deformation elastoplastic Finite Element analysis”. In: *Computers & Structures* 114 (2013), pp. 133–146.

[168] Chaman Singh Verma and Krishnan Suresh. “Towards FEA over tangled quads”. In: *Procedia Engineering* 82 (2014), pp. 187–199.

[169] Gregory J Wagner and Wing Kam Liu. “Application of essential boundary conditions in mesh-free methods: a corrected collocation method”. In: *International Journal for Numerical Methods in Engineering* 47.8 (2000), pp. 1367–1379.

[170] Li Wang et al. “Discontinuous Galerkin and Petrov Galerkin methods for compressible viscous flows”. In: *Computers & Fluids* 100 (2014), pp. 13–29.

[171] Jonathan Weissmann et al. “Cardiac mesh morphing method for finite element modeling of heart failure with preserved ejection fraction”. In: *Journal of the Mechanical Behavior of Biomedical Materials* 126 (2022), p. 104937.

[172] Zdzisław Więckowski. “The material point method in large strain engineering problems”. In: *Computer methods in applied mechanics and engineering* 193.39-41 (2004), pp. 4417–4438.

[173] Peter Wriggers et al. “Efficient virtual element formulations for compressible and incompressible finite deformations”. In: *Computational Mechanics* 60 (2017), pp. 253–268.

[174] Cheng-Kong C Wu and Michael E Plesha. “Essential boundary condition enforcement in meshless methods: boundary flux collocation method”. In: *International Journal for Numerical methods in engineering* 53.3 (2002), pp. 499–514.

[175] Shao-Wei Wu et al. “Arbitrary polygon mesh for elastic and elastoplastic analysis of solids using smoothed finite element method”. In: *Computer Methods in Applied Mechanics and Engineering* 405 (2023), p. 115874.

[176] Gang Xu et al. “Optimal analysis-aware parameterization of computational domain in 3D isogeometric analysis”. In: *Computer-Aided Design* 45.4 (2013), pp. 812–821.

[177] Gang Xu et al. “Parameterization of computational domain in isogeometric analysis: methods and comparison”. In: *Computer Methods in Applied Mechanics and Engineering* 200.23-24 (2011), pp. 2021–2031.

[178] Kaoji Xu, Xifeng Gao, and Guoning Chen. “Hexahedral mesh quality improvement via edge-angle optimization”. In: *Computers & Graphics* 70 (2018), pp. 17–27.

[179] Yuan-Fan Yang et al. “Two 8-node quadrilateral unsymmetric elements with different incompatible modes immune to severe distortion”. In: *International Journal for Numerical Methods in Engineering* (2023).

[180] William Zeng and GR3776476 Liu. “Smoothed finite element methods (S-FEM): an overview and recent developments”. In: *Archives of Computational Methods in Engineering* 25.2 (2018), pp. 397–435.

[181] Hongmei Zhang, Guoqun Zhao, and Xinwu Ma. “Adaptive generation of hexahedral element mesh using an improved grid-based method”. In: *Computer-Aided Design* 39.10 (2007), pp. 914–928.

[182] Yongjie Zhang, Xinghua Liang, and Guoliang Xu. “A robust 2-refinement algorithm in octree or rhombic dodecahedral tree based all-hexahedral mesh generation”. In: *Computer Methods in Applied Mechanics and Engineering* 256 (2013), pp. 88–100.

[183] Pei-Lei Zhou et al. “An unsymmetric 8-node hexahedral element with high distortion tolerance”. In: *International Journal for Numerical Methods in Engineering* 109.8 (2017), pp. 1130–1158.

[184] Xiaosong Zhu and Youyuan Wang. “A Simple Automatic Hexahedron Mesh Generation and Polyhedral Smoothed Finite Element Method for Mechanics Problems”. In: *Computer-Aided Design* 152 (2022), p. 103391.

[185] Olek C Zienkiewicz, Robert L Taylor, and Jian Z Zhu. *The finite element method: its basis and fundamentals*. Elsevier, 2005.

[186] Olgierd Cecil Zienkiewicz, Robert Leroy Taylor, and Robert Leroy Taylor. *The finite element method: solid mechanics*. Vol. 2. Butterworth-heinemann, 2000.