

Finite element analysis over tangled simplicial meshes: Theory and implementation



Josh Danczyk, Krishnan Suresh*

Department of Mechanical Engineering, University of Wisconsin-Madison, Madison, WI 53706, USA

ARTICLE INFO

Article history:

Received 20 August 2012

Received in revised form

4 April 2013

Accepted 5 April 2013

Available online 9 May 2013

Keywords:

Tangled

Inverted

Morphing

ABSTRACT

In modern finite element analysis (FEA), a mesh is said to be '*tangled*' if it contains one or more inverted elements. Tangling can occur, for example, during mesh optimization and mesh morphing. Modern finite element theory and commercial FEA packages are not designed to handle tangled meshes, i.e., they can lead to erroneous results. Researchers and practitioners therefore unanimously recommend untangling prior to analysis.

In this paper, a new mathematical framework for FEA over tangled meshes is proposed. Specifically, by defining a cell decomposition of a tangled mesh, and an associated set of cell shape functions, it is shown that FEA can be successfully carried out over tangled meshes. The cell shape functions are constructed through an oriented linear combination of the classic element shape functions. Numerical examples illustrate the correctness of the proposed framework. Potential applications of the proposed framework are also illustrated.

© 2013 Elsevier B.V. All rights reserved.

1. Introduction

In modern finite element analysis (FEA), the underlying mesh is required to satisfy numerous topological and geometrical properties [1–3]. For example, the mesh must: (1) be connected, (2) conform to the boundary, (3) be of 'good quality', and (4) not contain inverted elements, etc. In this paper, the primary focus is on the last constraint.

Fig. 1 illustrates an unacceptable mesh with one inverted element. In a later Section, we shall establish that the presence of an inverted element necessarily implies that the mesh must contain overlapping elements, i.e., the mesh must be tangled.

Modern finite element theory and commercial FEA packages are not designed to handle tangled meshes, i.e., they can lead to erroneous results. This is confirmed later in this paper through a simple experiment using the commercial package ANSYS [4]. Researchers and practitioners therefore unanimously recommend untangling the mesh prior to analysis. For example, to quote [5]: "Because tangled meshes generate physically invalid solutions, it is imperative that such meshes [be] untangled", and to quote [6]: "Since constitutive models for real materials are meaningful only for uninverted material, standard finite element simulation algorithms fail as soon as a single tetrahedron inverts ..."

Unfortunately, untangling is as difficult as mesh generation and optimization [7]. Indeed, as pointed in [6], none of these untangling methods are guaranteed to work, and "failure to untangle a

single tetrahedron forces the simulation to fail for most real world constitutive model". Therefore, a new extension to FEA is proposed in this paper; this extension provides the necessary framework for handling tangled meshes.

The remainder of the paper is organized as follows. In Section 2, a brief literature review reveals the practical challenges posed by tangled meshes. Then, in Section 3, the mathematical ambiguity posed by tangled meshes is identified and isolated. Then, the notions of the cell decomposition and the cell shape function are introduced to resolve the ambiguity. Finally, a critical theorem for handling tangled meshes is established. In Section 4, a practical implementation for FEA over tangled meshes is discussed. This is followed by numerical experiments in Section 5. Conclusions and future work are addressed in Section 6.

2. Literature review

Tangling of finite element meshes can occur today during:

- **Mesh generation:** Mesh generators are generally adept at constructing high quality non-tangled meshes of complex geometries. Yet, for example, during all-hex meshing, tangling can inadvertently occur, resulting in a mesh failure [8]. The user must then attempt a remesh with a different set of mesh parameters.
- **Mesh optimization:** Similarly, mesh optimizers, in an attempt to improve mesh quality, can inadvertently introduce tangling [9–12]. Further, classic notions of mesh quality are inappropriate in the presence of tangles [9,12–14]. Therefore, the mesh is

* Corresponding author. Tel.: +1 608 262 3594; fax: +1 608 265 2316.
E-mail address: suresh@engr.wisc.edu (K. Suresh).

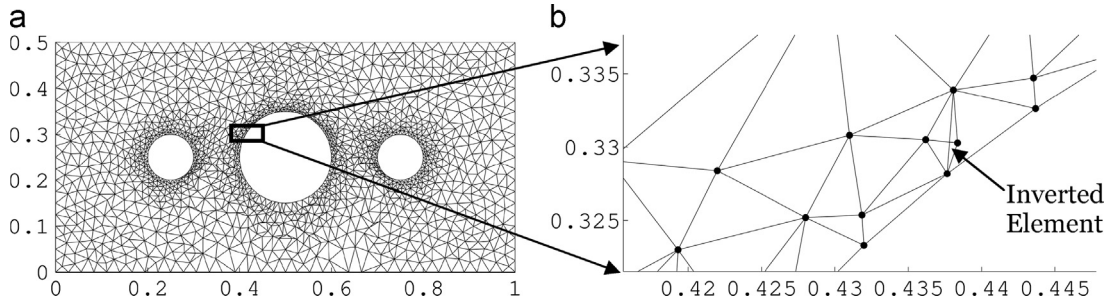


Fig. 1. (a) A tangled mesh and (b) a close-up of the tangling (observe the inverted element).

first untangled [15], and then optimized [9]. More recently, research efforts have focused on improving the quality of meshes while untangling [5,7,12].

- **Mesh morphing:** When the underlying geometry is modified, it may be advantageous to morph, i.e., stretch and transform, an existing mesh rather than remesh [16]. Various mesh morphing techniques exist today [16]. Unfortunately, all mesh morphing methods, especially the simple and efficient ones, can lead to tangled meshes [16].
- **Mesh deformation:** Last, but not the least, in large-scale deformation, excessive node movement can result in tangling [17], and remeshing can result in a significant loss in accuracy [18]. The concept of invertible finite elements was developed in [6,19] specifically to address this problem. In [6,19], the authors propose a numerical strategy of modifying the constitutive model in the presence of inverted elements. While this method leads to visually correct results and is a practical solution to large-deformation computer-graphics problems, the theoretical basis is not fully established. Indeed, the authors state: “We heuristically assume that each element is as uninverted as possible, ...” In [20], the concept of invertible finite element was further extended to provide a smooth extension of arbitrary isotropic hyper-elastic energy. Once again, heuristics were used to achieve numerical efficiency; to quote [20]: “In fact, we provide a heuristic that prevents the need for the costly SVD whenever the material is in this uninverted region.” In contrast, the method proposed in this paper does not entail heuristics.

Thus, the inability of modern FEA to handle tangled meshes has put a severe burden on the user and FEA related technologies. In passing, we note that the s-FEM framework was proposed in [21] for handling overlapping elements, but s-FEM does not apply to a tangled mesh containing inverted elements. On the other hand, a preliminary FEA framework for handling tangled meshes (with inverted elements) was proposed in [22]. In this paper, this preliminary framework is formalized through a sharper theory, and is accompanied by a robust and efficient implementation.

3. Proposed framework

The objective of this Section is to identify the specific problem posed by tangled meshes.

3.1. Problem setting

As a vehicle for discussion, consider the Poisson problem [1]:

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

$$\int_{\Omega} (\nabla v) \cdot (\nabla u) d\Omega = \int_{\Omega} v f d\Omega + \int_{\Gamma_N} v q d\Gamma \quad \forall v \in H_0^1(\Omega) \quad (3.1)$$

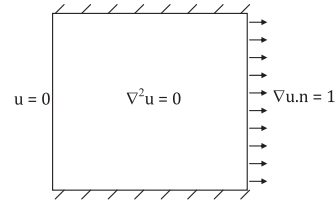


Fig. 2. An illustrative Poisson problem.

where $\Omega \subset \mathbb{R}^n$, $n = 1, 2, 3$ is a domain with a smooth boundary; see Fig. 2 for an illustrative example.

In the classic finite element analysis (FEA) of the above problem, a finite element mesh is first constructed over the domain [1]; see Fig. 3a. By construction of the mesh, the elements are positively-oriented, non-overlapping and form a quasi-disjoint decomposition. Specifically, given an interior node (see Fig. 3b), let ω be the union of all elements E_1, E_2, \dots attached to the node. By construction of the finite element mesh [23]:

$$\begin{aligned} \omega &= \cup_i E_i \\ E_i \cap E_j &= \emptyset, i \neq j \end{aligned} \quad (3.2)$$

In modern FEA, to satisfy the ‘continuity’ requirement of $u, v \in H_0^1(\Omega)$ in Eq. (3.1), the function space is defined as follows. Over each element E_i surrounding a node, element shape functions N_i are defined such that: (1) N_i takes a value of 1 at that node, (2) N_i goes to zero at all other nodes of that element, and (3) neighboring shape functions N_i and N_j are continuous across element boundaries. A typical element shape function is illustrated in Fig. 4.

Due to the quasi-disjoint decomposition, the element shape functions N_1, N_2, \dots can be ‘stitched’ together into single hat function ϕ over the region ω ; see Fig. 5. From this construction, it can be shown that (page 41 of [24]):

$$\phi \in H_0^1(\omega) \quad (3.3)$$

In other words, ϕ is sufficiently smooth and goes to zero on the boundary of the region surrounding the node identified. Further, ϕ can be trivially extended to zero over the remainder of the mesh. This process is repeated for each node i to yield hat functions $\phi^{(i)}$.

Finally, one seeks an approximate solution of the form:

$$u = \sum_i \hat{u}^{(i)} \phi^{(i)} \quad (3.4)$$

where $\hat{u}^{(i)}$ is the degree of freedom associated with node i . Due to Eq. (3.3), it can be shown that [25], [24]:

$$u \in H_0^1(\Omega) \quad (3.5)$$

3.2. Problems posed by a tangled mesh.

Unfortunately, the above process breaks down in a tangled mesh. To identify the underlying reason, a few definitions are essential.

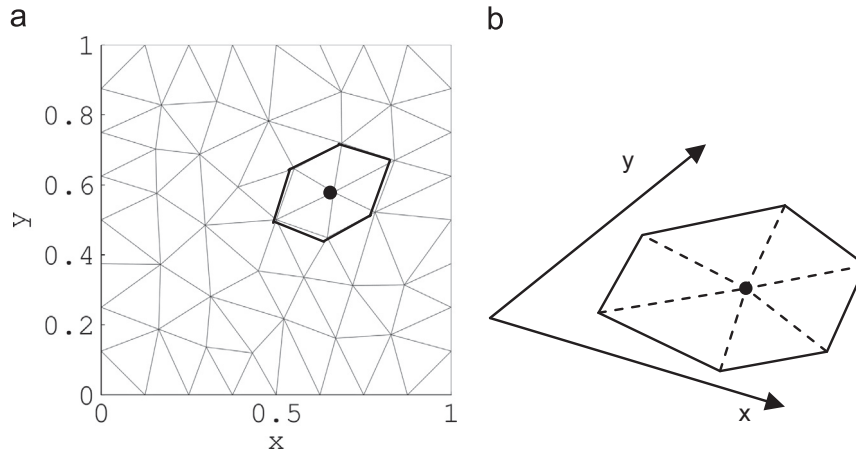


Fig. 3. (a) A finite element mesh, (b) the region ω surrounding a node.

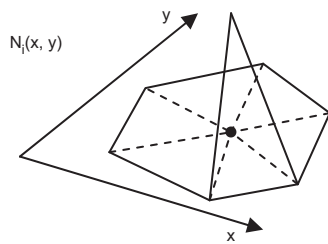


Fig. 4. An element shape function N_i .

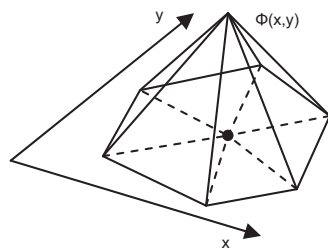


Fig. 5. Hat function ϕ defined over the region ω .

To begin with, the orientation of simplicial elements is defined here as follows.

Definition 1. In 2D, a triangle element defined by three points abc (in that order) is *positively/negatively oriented* if the interior of the element lies to the left/right of the oriented segment ab , or equivalently bc or ca . □

Definition 2. In 3D, a tetrahedral element defined by four points $abcd$ (in that order) is *positively/negatively oriented* if the interior of the element lies to the left/right of the oriented plane abc . □

One can show that if a simplicial element is positively/negatively oriented as defined above, then the determinant of its Jacobian is positive/negative [5].

Definition 3. Let the orientation of a simplicial element be denoted by θ_k , where $\theta_k = \pm 1$. □

Finally, a tangled mesh is defined in this paper as follows.

Definition 4. A connected and topologically valid finite element mesh is *tangled* if it contains elements of opposite orientation. □

An example of a tangled mesh is illustrated in Fig. 6a. Given a connected and topologically valid finite element mesh that is

acceptable by modern FEA, it is assumed that the tangled mesh is obtained through a pure node movement, i.e., the topology is not modified. Fig. 6b illustrates a node attached to an inverted element.

One can now establish the following result.

Lemma. In a tangled mesh, there must exist a pair of neighboring overlapping elements.

Proof. In a tangled mesh, consider any element with a negative orientation and one of its neighbors with a positive orientation (such a pair must exist in a connected tangled mesh). In 2-D, let the two neighboring triangles be defined by abc (the positive element) and bac' (the negative element). By definition, the interior of abc lies to the left of ab , and the interior of bac' lies to the right of ba , i.e., to the left of ab . Thus there are points that belong to both elements, i.e., the two elements overlap. Similar arguments hold in 3-D. □

The above lemma implies that there are points within a mesh that belong to multiple elements, the hat function ϕ is ill-defined/ambiguous at such points; see Fig. 6b.

The ambiguity of ϕ can be resolved in multiple ways. For example, ϕ at an overlapping point can be defined as: (1) the maximum of all element shape function values at that point, or (2) the sum/average of these values, and so on. However, in addition to resolving the ambiguity, Eq. (3.3) must also be satisfied; the next Section discusses the proposed methodology.

3.3. Proposed theoretical framework

In this Section, the focus is on defining the hat-function ϕ at each node of a tangled mesh such that it is unambiguous, and it satisfies the continuity requirement of Eq. (3.3). Once this is done, the finite element framework for tangled meshes can be easily established.

As in the previous Section, the focus is on a single node within a mesh and the set of elements attached to it. These elements are allowed to tangle in an arbitrary manner. We retain the previous definition of ω as the (set) union of all elements attached to the given node (see Eq. (3.2)). However, since a point within ω can belong to multiple elements, we define an *index* associated with each point within ω as:

$$I(p) = \{k | p \in E_k\} \tag{3.6}$$

i.e., the index associated with a point within a mesh is the (integer) set of all elements containing that point. Note that, in a non-tangled mesh, the index of every point is a unique element

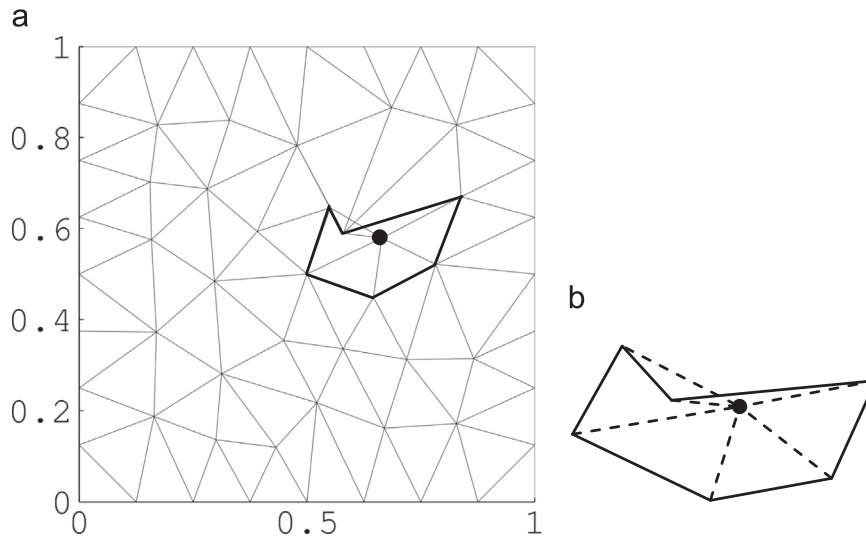


Fig. 6. (a) Tangled mesh, and (b) elements surrounding a given node.

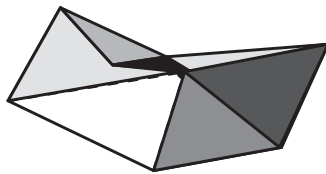


Fig. 7. Cells within a tangled mesh.

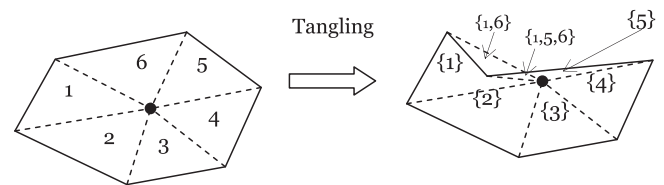


Fig. 8. Cells indices.

number. However, in a tangled mesh, all permutations and combinations are possible. Further, points may share the same index, leading to the notion of a cell.

Definition 5. A cell is the set of all points with identical index l . □

For example, Fig. 7 illustrates seven distinct cells within ω . Observe that a cell need not be convex, or even connected.

Fig. 8 illustrates the indices associated with the cell decomposition.

Henceforth, cells will be denoted by C_α, C_β, \dots . Since cells are associated with a unique index, we have:

$$\omega = \bigcup_{\alpha} C_{\alpha}$$

$$C_{\alpha} \cap C_{\beta} = \emptyset, \alpha \neq \beta \tag{3.7}$$

In other words, the cells induce a quasi-disjoint decomposition of ω .

Just as an element shape function N_i is defined over each element E_i in a non-tangled mesh, a cell shape function S_{α} is defined here over each cell C_{α} as:

$$S_{\alpha}(\cdot) = \sum_{k \in I_{\alpha}} \theta_k N_k(\cdot) \tag{3.8}$$

where θ_k is the orientation of the element E_k . These cell shape functions form the basis of the proposed finite element framework. Indeed, the most critical theorem in establishing the proposed framework for tangled meshes is stated and proven next.

Theorem. The cell shape functions defined via Eq. (3.8) over the cell decomposition of Eq. (3.7) satisfy the following properties: (1) they are continuous across cell boundaries, and (2) they vanish on the boundary of ω .

Proof. Consider the first part of the proof. Let C_{α} and C_{β} be two neighboring cells, with indices I_{α} and I_{β} . It must be proven that the cell shape functions $S_{\alpha}(\cdot)$ and $S_{\beta}(\cdot)$ are continuous across the common boundary. To this end, consider the difference between

the two functions $S_{\alpha}-S_{\beta}$; one can group the terms into three categories:

$$S_{\alpha}-S_{\beta} = \sum_{k \in I_{\alpha} \cap I_{\beta}} \square + \sum_{k \in I_{\alpha} - I_{\beta}} \square + \sum_{k \in I_{\beta} - I_{\alpha}} \square \tag{3.9}$$

The first term contains the contribution from every element k that belongs to both I_{α} and I_{β} . For every such element, N_k is continuous across the common boundary, therefore the first term vanishes on the boundary, independent of the orientation θ_k . Next consider an element k in the second term, i.e., the element is in I_{α} but not in I_{β} . This implies that in crossing from C_{α} to C_{β} , we must exit element E_k . Observe that exiting E_k can only occur in two ways: (1) simultaneously exit ω , or (2) enter a neighboring element E_j . In the first case, N_k is necessarily zero at the boundary of point of ω . Therefore, all such contributions vanish from the second term. In the second case of exiting E_k and entering one of its neighbors E_j , there are again two cases: (2a) E_k and E_j are of the same orientation, and (2b) E_k and E_j are of opposite orientation. If the elements are of the same orientation, then E_j must belong to C_{β} , and $\theta_k N_k = \theta_j N_j$. Therefore these contributions vanish from the 2nd term. Finally, if E_k and E_j are of opposite orientation, then E_j must also belong to C_{α} , therefore $\theta_k N_k + \theta_j N_j = \theta_k N_k - \theta_k N_j$, which also vanishes since the two element functions are continuous at that point. Through a similar argument, the third term of Eq. (3.9) also vanishes. Thus $S_{\alpha}(\cdot)$ and $S_{\beta}(\cdot)$ are continuous across the common boundary.

Now consider the second part of the proof where it must be shown that if C_{α} intersects the boundary of ω , then S_{α} is necessarily zero on the boundary. Observe that the boundary of any C_{α} must be also the boundary of at least one element E_k . Since we are exiting C_{α} , we must also be exiting the element E_k . Once again, exiting E_k can only occur in two ways: (1) simultaneously exit ω , or (2) enter a neighboring element E_j . In the first case, N_k is necessarily zero at the boundary of point of ω . In the second case E_k and E_j must necessarily be of opposite orientation (since we are also exiting ω). As before, the

contributions from both elements cancel out at such boundary points, and therefore, S_α vanishes.

Finally, exactly as in classic FEA [24], due to the quasi-disjoint decomposition and continuity, the cell shape functions S_α can be stitched together to define ϕ at a node, and therefore Eq. (3.3) holds true.

3.4. Finite element assembly

The previous section introduced the concept of cells and cell shape functions for establishing the proposed FEA framework. However, cells are unnecessary in a practical implementation of FEA over a tangled mesh. In other words, there is no need to explicitly compute the cell decomposition; the underlying reason is as follows.

Upon stitching the cell shape functions C_α , an equivalent definition for the nodal shape functions is simply given by:

$$\phi(p) = \sum_{k|p \in E_k} \theta_k N_k(p) \quad (3.10)$$

In other words, the hat function at any point is simply the oriented linear combination of *all* element shape functions of elements attached to that node, containing that point. Observe that when the elements are not tangled this recovers the classic definition. Generalizing this to an arbitrary node i yields:

$$\phi^{(i)}(p) = \sum_{k|p \in E_k, E_k \in \kappa^{(i)}} \theta_k N_k^{(i)}(p) \quad (3.11)$$

where $\kappa^{(i)}$ is the set of all element attached to node i . These $\phi^{(i)}$ are now combined to construct approximate solutions:

$$u(\cdot) = \sum_i \hat{u}^{(i)} \phi^{(i)}(\cdot) \quad (3.12)$$

Regrouping terms yields:

$$u(p) = \sum_{k|p \in E_k} \theta_k \sum_{i|E_k \in \kappa^{(i)}} \hat{u}^{(i)} N_k^{(i)}(p) \quad (3.13)$$

Observe that the inner summation is the description of the field over one element, and is identical to that used in classic finite element. Indeed, for a given element, one can define $\mathbb{N}_k(p)$ as the collection of element shape functions over all nodes attached to the element:

$$\mathbb{N}_k(p) = \left\{ N_k^{(i_1)}(p) \quad N_k^{(i_2)}(p) \quad \dots \quad N_k^{(i_m)}(p) \right\} \quad (3.14)$$

Further, one can define the unknown degrees of freedom associated with all nodes attached to an element as:

$$\hat{\mathbf{u}}_k = \left\{ \hat{u}^{(i_1)} \quad \hat{u}^{(i_2)} \quad \dots \quad \hat{u}^{(i_m)} \right\} \quad (3.15)$$

This leads to the approximation:

$$u(p) = \sum_{k|p \in E_k} \theta_k \mathbb{N}_k(p) \hat{\mathbf{u}}_k \quad (3.16)$$

Upon substituting Eq. (3.16) into Eq. (3.1), and eliminating the auxiliary degrees of freedom the following is obtained:

$$\left[\int_{\Omega} \left(\nabla \sum_j \theta_j \mathbb{N}_j \right)^T \cdot \left(\nabla \sum_k \theta_k \mathbb{N}_k \right) d\Omega \right] \hat{\mathbf{u}} = \int_{\Omega} \left(\sum_j \theta_j \mathbb{N}_j \right) f d\Omega \quad (3.17)$$

The left-hand side of Eq. (3.17) can be regrouped as:

$$\sum_j \int_{\Omega} \theta_j^2 \nabla \mathbb{N}_j^T \cdot \nabla \mathbb{N}_j d\Omega + \sum_j \sum_{k \neq j} \int_{\Omega} \theta_j \theta_k \nabla \mathbb{N}_j^T \cdot \nabla \mathbb{N}_k d\Omega \quad (3.18)$$

Since $\mathbb{N}_k(p)$ is zero outside of element k , the integration in Expression (3.18) can be restricted to the element itself. In addition, since $\theta_j = \pm 1, \theta_j^2 = 1$. Therefore Expression (3.18) simplifies to:

$$\sum_j \int_{E_j} \nabla \mathbb{N}_j^T \cdot \nabla \mathbb{N}_j d\Omega + \sum_j \sum_{k \neq j} \int_{E_j \cap E_k} \theta_j \theta_k \nabla \mathbb{N}_j^T \cdot \nabla \mathbb{N}_k d\Omega \quad (3.19)$$

The first term of Expression (3.19) corresponds to the stiffness matrix in classic FEA. The second term, on the other hand, captures the coupling between overlapping elements. Observe that, while there may be many overlapping elements that contain the same point, only two elements need to be considered at a time.

In summary, to solve the Poisson equation via finite element over a tangled mesh, the following terms must be computed (the summation is interpreted in the usual sense of finite element assembly):

$$\begin{aligned} K_{classic} &= \sum_j \int_{E_j} \nabla \mathbb{N}_j^T \cdot \nabla \mathbb{N}_j d\Omega \\ K_{overlapping} &= \sum_j \sum_{k \neq j} \int_{E_j \cap E_k} \theta_j \theta_k \nabla \mathbb{N}_j^T \cdot \nabla \mathbb{N}_k d\Omega \\ f_{oriented} &= \sum_j \int_{E_j} \theta_j \mathbb{N}_j^T f d\Omega \end{aligned} \quad (3.20)$$

Finally, the following linear system of equations must be solved:

$$(K_{classic} + K_{overlapping}) \hat{\mathbf{u}} = f_{oriented} \quad (3.21)$$

Observe that if there are no overlaps, then $K_{overlapping}$ is identically zero. In addition, in a non-tangled mesh, all elements are necessarily positively oriented (see Lemma of previous Section), and therefore $f_{oriented}$ reduces to $f_{classic}$, i.e., classic FEA is exactly recovered.

While the above finite element implementation targets the 2-D Poisson equation, the implementation is easily generalized to other elliptic equations. Numerical examples later illustrate the validity of the method for a variety of field problems in 2-D and 3-D.

4. Implementation

In this Section, the implementation of the proposed framework is discussed in further detail.

4.1. Overlap detection

The proposed methodology requires integrating over regions that are common to two elements; see Eq. (3.20). Therefore, the first step is to identify elements that overlap. A naïve $O(n^2)$ implementation entails checking all elements against all other elements. A more efficient strategy relies on the earlier Lemma that states that “neighboring elements with opposite orientations will overlap”. This serves as the basis of the following proposed algorithm:

1. Seed a stack with all pairs of neighboring elements with opposite orientation (each such pair must necessarily overlap).
2. While the stack is not empty:
 - a. Pop the top of the stack. This yields the element pair (e_1, e_2) .
 - b. If this pair has been processed already, continue to step a.
 - c. If the elements do not overlap, continue to step a.
 - d. (Else) Record the overlap.
 - e. For all neighbors e_n of e_1 , add (e_n, e_2) to stack.
 - f. For all neighbors e_n of e_2 , add (e_n, e_1) to stack.

The computationally expensive part of the above algorithm is step c, i.e., determining if two elements overlap. For simplicial elements, this amounts to convex polytope/convex polytope intersection [26].

4.2. Algorithm: correctness and complexity

In order to establish the correctness of the above algorithm, the following assumption is essential.

Assumption. If two elements overlap, there exists a neighbor of one of the elements that the other element overlaps, i.e., if $E_i \cap E_j \neq \emptyset$, then $\exists E_k \in N(E_i): E_k \cap E_j \neq \emptyset$ or $\exists E_k \in N(E_j): E_i \cap E_k \neq \emptyset$.

In other words, the scenario in Fig. 9a is acceptable, while that of Fig. 9b where two elements *exactly* overlap is disallowed.

Given this assumption, the following theorem on the correctness of the above algorithm can be established.

Theorem. In a tangled mesh, all overlapping element pairs are detected via the above algorithm.

Proof. Assume the contrary, i.e., assume that there exists an overlapping pair (E_i, E_j) that was never inserted into the stack; in other words the algorithm failed to identify this pair. Now, let E_k be a neighbor of E_i . From the assumption above, $E_k \cap E_j \neq \emptyset$. Further, from steps 2e and 2f of the above algorithm it follows that the pair (E_i, E_j) must have gone undetected too, i.e., could not have been part of the stack. This argument holds *ad infinitum* leading to a conclusion that no element pair that is topologically connected to the pair (E_i, E_j) could have been in the stack. This is not possible since step 1 of the above algorithm seeds the stack with all pairs of elements that are neighbors and of opposite orientation. Thus, by contradiction, the pair (E_i, E_j) will be detected by the above algorithm. ■

In the case that two elements exactly overlap, i.e., when the above assumption is violated, such element pairs can also be seeded onto the stack. Since the elements must overlap exactly, this detection is trivial, and is not discussed here.

Next consider the algorithmic complexity. Let m be the number of neighbors; $m=3$ for a triangle mesh, $m=4$ or a tetrahedral mesh. Let n be the number of overlapping element pairs in a tangled mesh. For each overlapping element pair, steps 2e and 2f of the above algorithm will push at most $2m$ additional element pairs onto the stack. Further step 2b ensures that an element pair is only processed once. Therefore, the total number of element pairs that get pushed to the stack, and thus checked for overlap, is at most $2mn$, which is linear with respect to n .

4.3. Tangled finite element assembly

Recall that performing FEA over a tangled mesh requires computing the terms in Eq. (3.20). First, $K_{classic}$ is computed exactly as in classic FEA [1]. Next, $f_{oriented}$ is computed as in classic FEA with one modification: the orientation of the element is accounted for by multiplying the force components for each element by its orientation (± 1) before assembling it into the global $f_{oriented}$ term.

To compute $K_{overlapping}$, the overlapping regions is computed (see previous Section and Fig. 10a). In order to perform the integration, the overlapping region is subsequently triangulated, see Fig. 10b.

Now, as in classic FEA, numeric integration is performed over the Gauss points of the newly formed triangles. To compute ∇_{N_j} and ∇_{N_k} at the Gauss points, the Gauss points from the parametric element are mapped to the physical (x, y) space, and then the parametric coordinates of the parent elements E_j, E_k that correspond to the (x, y) location are determined. The gradients ∇_{N_j} and ∇_{N_k} are computed at the respective parametric coordinates. As with $f_{oriented}$, before adding the local stiffness matrix to the global $K_{overlapping}$ matrix, the orientation of the parent elements are accounted for as in Eq. (3.20).

In the proposed method, the condition number of the overall stiffness matrix changes when the contribution due to the overlapping elements is added. In all examples considered, the condition number never grew more than a factor of three. However, additional investigation is required. Further, it is possible that

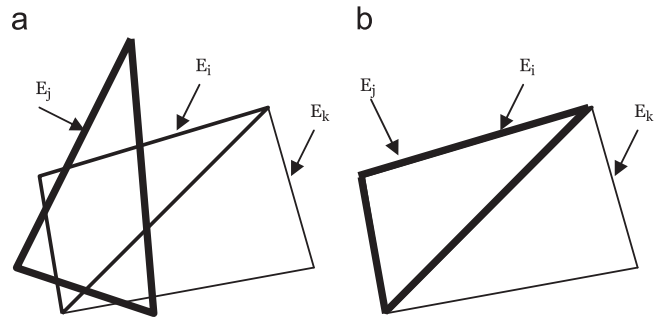


Fig. 9. (a) Acceptable overlapping and (b) unacceptable when two elements *exactly* overlap.

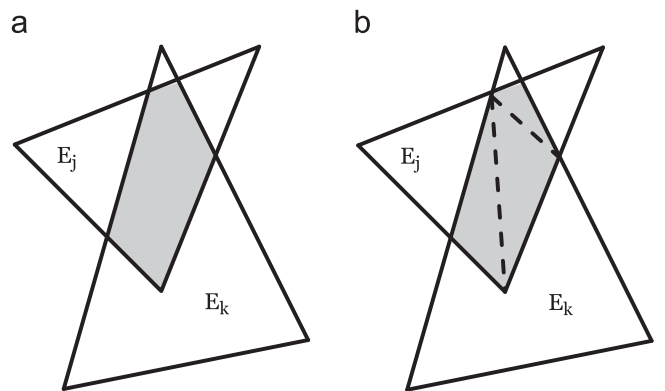


Fig. 10. (a) Overlapping region of elements E_j and E_k , and (b) the triangulation of the overlapping region.

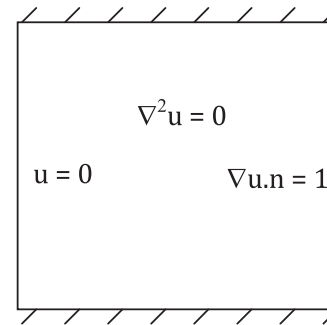


Fig. 11. Thermal conduction problem.

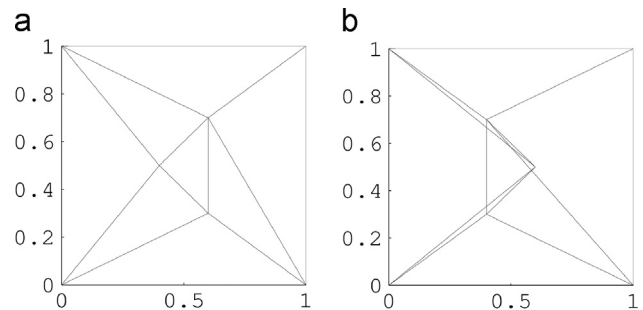


Fig. 12. (a) Valid mesh and (b) tangled mesh.

untangling is a better choice in some scenarios. A distinct advantage of the proposed method is that it guarantees finite-termination (unlike untangling).

5. Numeric examples

The proposed theory is now illustrated through numeric examples. In particular, the first few examples are ‘validity tests’ [1,25,27] over tangled meshes. (We refer to these as validity tests rather than patch tests, since the concept of patch-tests is not established for inverted elements.) These tests are designed in this paper to identify incorrect theory/implementation by determining whether the finite elements can capture, to within machine precision, a field within the space of the finite element basis functions. In particular, ANSYS 13 [4] is compared against the proposed methodology. After these tests, the focus turns to a specific application of tangling, namely mesh morphing.

Table 1
Solution at $x=1, y=0$.

	ANSYS 13	Proposed theory
Valid Mesh	1.0000	1.0000
Tangled Mesh	0.9875	1.0000

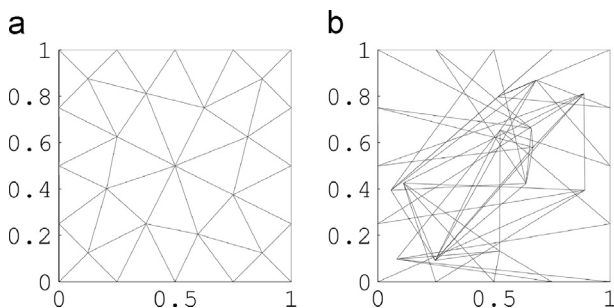


Fig. 13. Initial untangled mesh (a) and the actual tangled mesh (b) used for validity tests.

Table 2
Normalized errors for 2-D Poisson problem validity test.

	Classic FEA	Proposed theory
Linear	1.8438	4.9145e-13
Quadratic	7.5844	4.0306e-12

Table 3
Normalized errors for 2-D plane stress problem validity test.

	Classic FEA	Proposed theory
Linear	1.6037	2.2277e-12
Quadratic	66.4133	2.9278e-11

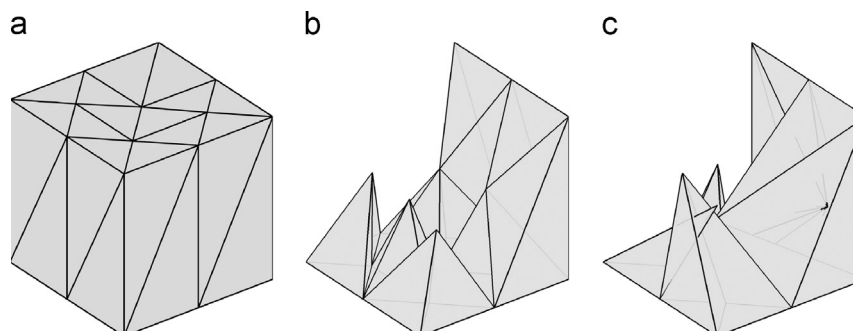


Fig. 14. (a) Initial mesh, (b) Portions of the initial untangled mesh, and (c) the actual tangled mesh used for validity tests.

5.1. Validity test: A simple tangle

Consider a thermal conduction problem over a unit square with a thermal conductivity of 1. The left edge is set to a temperature of 0, a thermal flux of 1 is applied on the right edge, and the top and bottom edges are insulated; see Fig. 11. The exact solution to this problem is $u(x, y) = x$.

The problem is solved over two meshes with linear element shape functions: (1) a regular valid mesh of Figs. 12a, and (2) a tangled mesh of Fig. 12b that is constructed by flipping the x -location of the interior nodes in Fig. 12a about the $x=0.5$ line. Since the solution falls within the finite element space, the exact solution should be recovered to within machine precision (even though the triangles are of poor quality).

Table 1 illustrates the solution for the two meshes at the location (1, 0), as produced by ANSYS and the proposed methodology. As expected, with a valid mesh, the exact solution of 1.0 is recovered via ANSYS 13 [4] and the proposed methodology (to within machine precision). However, when the mesh is tangled, ANSYS results in a 1.2% error. Increasing the size of the inverted element leads to errors as large as 10%. On the other hand, the proposed theory yields the exact solution even in the case of the tangled mesh.

5.2. Validity test: multiple tangles

Next consider triangulating a unit square, see Fig. 13a, and then randomizing the location of the interior nodes to produce the tangled mesh shown in Fig. 13b. A (random) linear field $u(x, y) = 0.323x - 0.651y + 0.998$ was chosen as the exact solution to a Poisson problem; Dirichlet conditions, Neumann conditions, and body forces were computed from this field, and applied on the tangled mesh as follows: Neumann conditions are applied at the bottom, right, and top boundaries, and Dirichlet conditions are applied to the left boundary. The objective is to recover the exact solution using linear elements. Similarly a random quadratic field was chosen as the exact solution, and the objective was to recover the exact solution using quadratic elements.

The results are summarized in Table 2 where normalized errors over all nodal values (defined as $abs(||u - u_{Exact}||) / ||u_{Exact}||$) from classic FEA and the proposed theory are summarized. Observe that, once again, the classic FEA fails the test in that it leads to erroneous results. On the other hand, the proposed method leads to results that are within machine precision accuracy. (It is to be noted that when in the numerical solution of linear system of equations, machine precision errors are inevitable, even for well-conditioned matrices [28]).

As a second test consider repeating the previous example but for plane stress. Specifically, two random linear displacement fields:

$$\begin{aligned} u(x, y) &= -0.358 + 0.562x + 0.906y \\ v(x, y) &= 0.233 - 0.359x - 0.670y \end{aligned} \tag{5.1}$$

were chosen. 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. The two meshes are identical to the previous example. Random quadratic fields were also chosen.

Normalized errors in solutions from classic FEA and the proposed theory are summarized in Table 3. As with the previous example, classic FEA fails the test for the tangled mesh, while the proposed theory achieves machine precision accuracy.

Similar 3-D tests were also performed on a unit cube, where the interior nodal positions were randomized. The mesh consists of 69 tetrahedra, 29 vertices, and upon tangling, 553 overlapping regions were detected. A portion of the mesh before and after tangling is shown in Fig. 14b and c, respectively; these are all the elements that have a face on the bottom or back of the unit cube.

Random linear and quadratic fields are chosen as the exact solution to a Poisson problem, and an elasticity problem. Surface tractions are applied to five of the six faces and Dirichlet conditions are applied to the remaining face. Normalized errors in solutions from classic FEA and the proposed theory are summarized in Tables 4 and 5 for the Poisson problem and elasticity problem, respectively.

5.3. Convergence test

Next, the proposed methodology is tested for convergence. Specifically, a function outside the span of the finite elements is chosen as the exact solution to a 2-D Poisson problem:

$$u(x, y) = e^{\pi x} \cos(\pi y) \tag{5.2}$$

The domain is a unit square with boundary conditions, etc. computed from the specified field as follows: Neumann conditions are applied at the bottom, right, and top boundaries, while Dirichlet conditions are applied to the left boundary.

The mechanism for introducing tangling is as follows. First, an initial non-tangled mesh is created within the unit square as in Fig. 15a. Then, nodes that lie within a circle of radius 0.25 are

Table 4
Normalized errors for 3-D Poisson problem validity test.

	Classic FEA	Proposed theory
Linear	0.3296	1.7924e-15
Quadratic	1.6116	2.0738e-12

Table 5
Normalized errors for 3-D elasticity problem validity test.

	Classic FEA	Proposed theory
Linear	1.4035	5.3276e-14
Quadratic	14.1704	1.9298e-11

mirrored about the center as illustrated in Fig. 15b. However, in the process of tangling, high-aspect ratio elements are inevitably introduced. Therefore, the nodes outside the circle are dragged towards the center of the domain according to the following heuristic rule:

$$d_i \leftarrow \|(0.5, 0.5) - p_i\|$$

$$p_i \leftarrow p_i + 1.9he^{-10(0.25-d_i)}((0.5, 0.5) - p_i) \tag{5.3}$$

An example of this tangling is illustrated in Fig. 15c.

The Poisson problem is then solved over both the non-tangled and tangled meshes, and the error in strain energy is computed for each case. In addition, this test is performed for both linear and quadratic shape functions. The results of the error in strain energy as a function of the number of degrees of freedom (as the mesh is refined) for all four cases are summarized in Fig. 16.

In Table 6 the convergence rates of all four cases are summarized. Observe that the rate of convergence for the tangled mesh is, for all practical purposes, equal to the non-tangled case.

5.4. Application: mesh morphing

As a specific application of the proposed methodology, considered here is the problem of *mesh morphing* with tangling. In finite element analysis (FEA), when the underlying geometry is modified, an existing mesh can be morphed to conform to the new geometry, a process referred to as mesh morphing. An advantage of mesh morphing over remeshing is that there is a one-to-one correspondence between mesh nodes before and after morphing. Consequently, operations such as finite difference computations are less error-prone [16].

Typical mesh morphing techniques include simplex-linear, simplex-natural neighbor, weighted-residual, FEMWARP, and

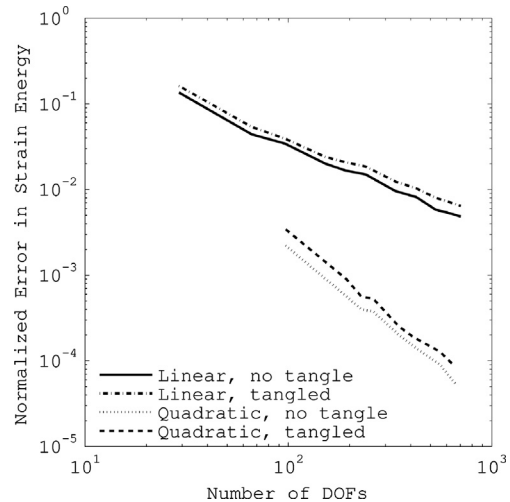


Fig. 16. Error in strain energy as a function of the number of degrees of freedom for a non-tangled and tangled mesh with linear and quadratic shape functions.

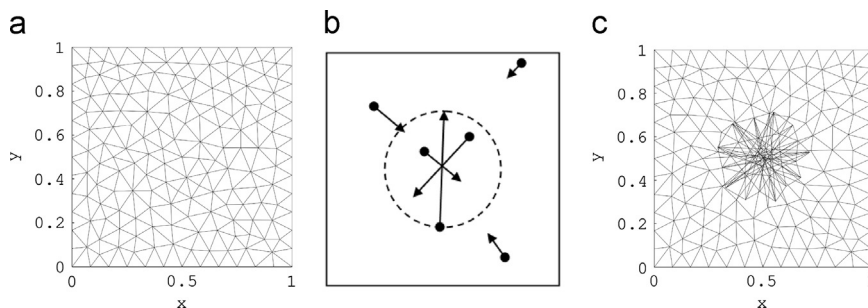


Fig. 15. (a) Initial mesh and (b) tangled mesh used in convergence study.

Table 6
Convergence rates.

Shape	Mesh	Convergence rate
Linear	Not tangled	1.02
Linear	Tangled	0.97
Quadratic	Not tangled	1.88
Quadratic	Tangled	1.92

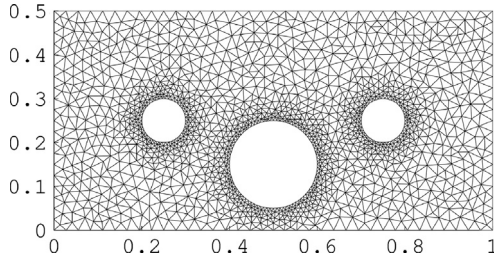


Fig. 17. Initial construction of a mesh that is subsequently morphed.

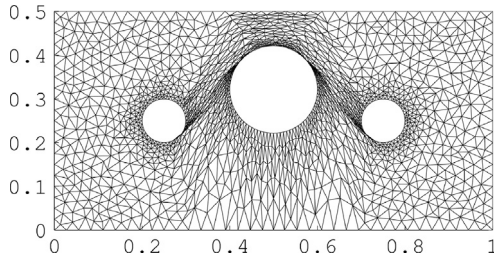


Fig. 18. A morphed mesh with a poor function space quality.

LBWARP [16]. The simplex-linear mesh morphing technique was chosen for the examples in this paper. As an example of how this technique works, consider the initial mesh illustrated in Fig. 17.

The larger hole is now displaced, and the mesh is morphed to conform to the new geometry, as shown in Fig. 18. Observe that the quality of the mesh has deteriorated. Furthermore, close inspection reveals that some of the elements overlap; i.e., the mesh is tangled. This is typical of mesh morphing; the quality and the degree of tangling can vary significantly depending on the morphing technique [16]. To improve mesh quality, an edge flipping technique is used to result in Fig. 19; however, tangling remains.

In Fig. 20, the number of overlaps (due to tangling) is summarized after: (1) morphing and (2) quality improvement (edge flipping). Observe that edge flipping can increase or decrease the number of overlaps.

To illustrate the effectiveness of the proposed method, the geometry is remeshed (not tangled) for comparison. The specific quantity of interest is the maximum von Mises stress. Fig. 21 shows that the proposed methodology over the tangled mesh is consistent with a full remesh.

5.5. 3-D morphing and tangling

In this final example, 3-D mesh morphing is considered. The bearing block shown in Fig. 22a is morphed so as to increase the diameter of the bearing surface (center hole) from 63.5 mm to 72 mm; see Fig. 22b. Also shown in the two figures are the boundary conditions; two of the mounting holes are fixed, while the third mounting hole has a force in the y and z direction.

The technique used to morph the mesh is similar to that used earlier in 2-D example where an explicit equation was used to

morph the mesh. The initial untangled mesh is shown in Fig. 23a. After morphing, the overlapping regions of the tangled mesh are illustrated in Fig. 23b.

The quantity of interest is the maximum displacement. Table 7 lists this quantity for the initial configuration, the morphed, tangled configuration, and a remeshed configuration (all quantities are in mm). Not only is the trend correct, increasing the diameter of the hole increases the displacement, but the displacement of the tangled mesh closely matches that of the remeshed solution.

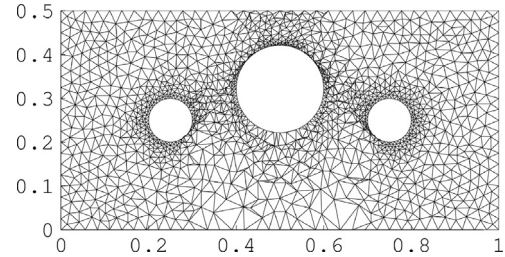


Fig. 19. Morphed mesh that has undergone edge flipping to improve the function space.

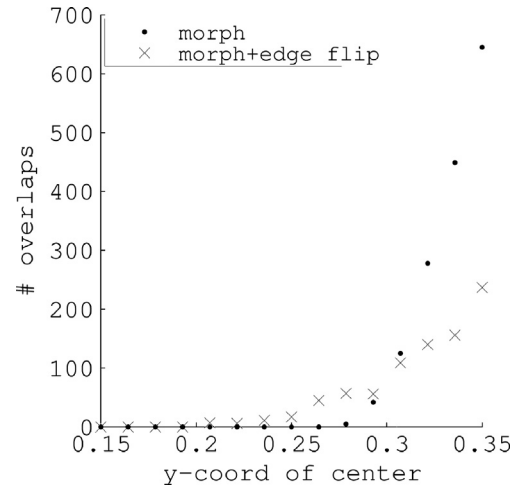


Fig. 20. Number of overlaps in the mesh of the plate with three holes problem after morphing and quality improvement.

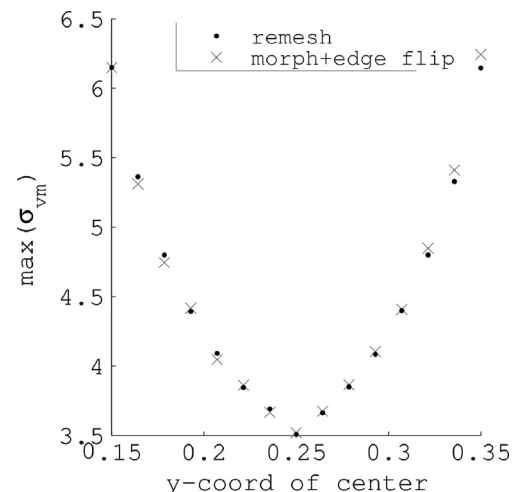


Fig. 21. Maximum von Mises stress in various morphed configuration with remeshing and a tangled morphed mesh.

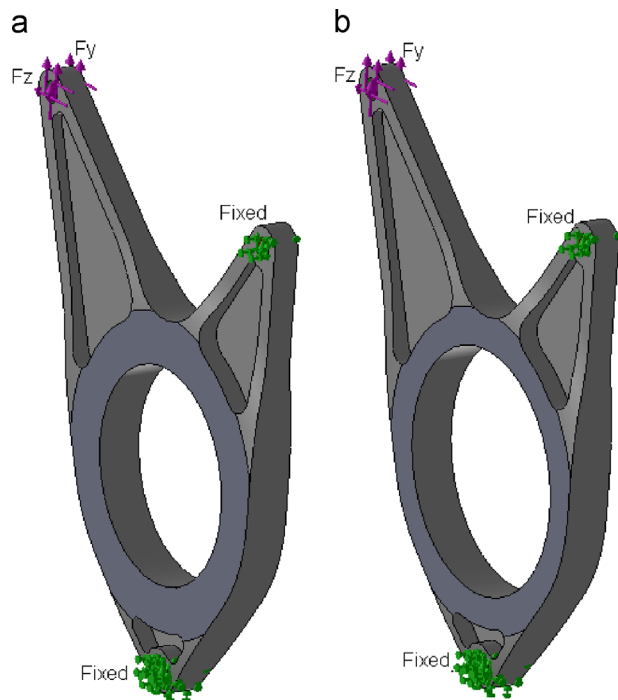


Fig. 22. (a) Initial bearing block design with a hole diameter of 63.5 mm and (b) final design with a hole diameter of 72 mm.

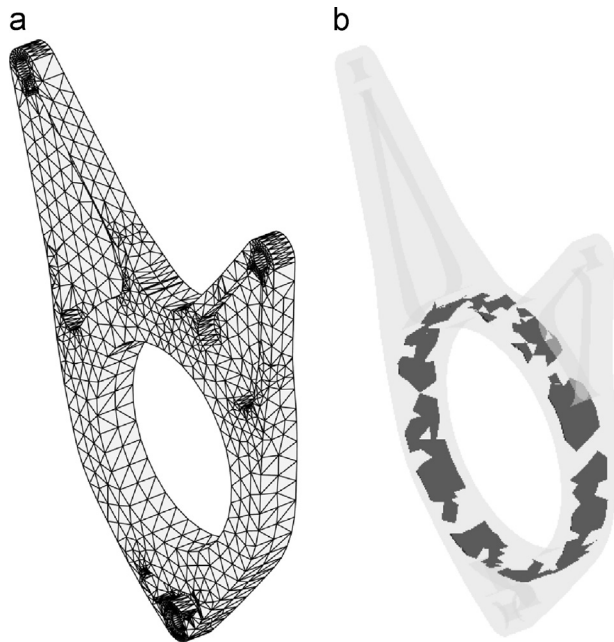


Fig. 23. (a) Initial mesh that is subsequently morphed and tangled. (b) Overlapping regions (dark portions) of the tangled mesh.

6. Conclusion and future work

In this paper an extension to the underlying mathematics of the classic finite element formulation is proposed. This extension allows FEA to be used in conjunction with tangled meshes that were previously considered unacceptable. In addition, it is shown that the proposed methodology can be easily incorporated into classic FEA with minor modifications. Numeric experiments illustrate the correctness of the proposed methodology; this is in contrast to commercial implementations of FEA.

Table 7
Comparison of maximum total displacement.

	Initial configuration		Final configuration	
			Morphed	Remesh
Maximum displacement	2.1230e-2		2.5392e-2	2.5371e-2

While only simplicial elements were considered, the extension to non-simplicial elements, e.g. quadrilateral and hexahedral elements, is currently being investigated. From a tangling perspective, the most critical difference between simplicial and non-simplicial elements is that non-simplicial elements can also suffer from *implicit tangling* in that an element can overlap with itself.

While the theory, in principle, extends to non-linear problems, further investigation and experiments are required. An investigation into the impact of tangling on the condition number of the underlying linear system is also required. Finally, and perhaps most critically, an investigation into the cost/benefit analysis of untangling versus solving over a tangled mesh is required.

References

- [1] O.C. Zienkiewicz, *The Finite Element Method: Its Basis and Fundamentals*, Elsevier Butterworth Heinemann, Oxford, UK, 2005.
- [2] S.H. Lo, Finite element mesh generation and adaptive meshing, *Prog. Struct. Eng. Mater.* 4 (2002) 381–399.
- [3] J.N. Reddy, *Introduction to the Finite Element Method*, McGraw Hill, New York, NY, 1985.
- [4] ANSYS 13. ANSYS; (www.ansys.com), 2012.
- [5] S. Bhowmick, S.M. Shontz, Towards high-quality, untangled meshes via a force-directed graph embedding approach *Procedia Comput. Sci.* 1 (1) (2010) 357–366.
- [6] G. Irving, J. Teran, R. Fedkiw, Tetrahedral and hexahedral invertible finite elements, *Graph. Models* 68 (66–89) (2006).
- [7] E.J. López, N.M. Nigro, M.A. Storti, Simultaneous untangling and smoothing of moving grids, *Int. J. Numer. Methods Eng.* 76 (7) (2008) 994–1019.
- [8] P.M. Knupp, A method for hexahedral mesh shape optimization, *Int. J. Numer. Methods Eng.* 58 (2) (2003) 319–332.
- [9] L.A. Freitag, P. Plassmann, Local optimization-based simplicial mesh untangling and improvement, *Int. J. Numer. Methods Eng.* 49 (1–2) (2000) 109–125.
- [10] P.M. Knupp, Achieving finite element mesh quality via optimization of the Jacobian matrix norm and associated quantities. Part II—a framework for volume mesh optimization and the condition number of the Jacobian matrix, *Int. J. Numer. Methods Eng.* 48 (8) (2000) 1165–1185.
- [11] P.M. Knupp, Achieving finite element mesh quality via optimization of the Jacobian matrix norm and associated quantities. Part I—a framework for surface mesh optimization, *Int. J. Numer. Methods Eng.* 48 (3) (2000) 401–420.
- [12] J.M. Escobar, E. Rodriguez, R. Montenegro, G. Montero, J.M. González-Yuste, Simultaneous untangling and smoothing of tetrahedral meshes, *Comput. Methods Appl. Mech. Eng.* 192 (25) (2003) 2775–2787.
- [13] L.A. Freitag, P.M. Knupp, Tetrahedral mesh improvement via optimization of the element condition number, *Int. J. Numer. Methods Eng.* 53 (2002) 1377–1391.
- [14] L. Xiang-yang, L.A. Freitag, *Optimization-Based Quadrilateral and Hexahedral Mesh Untangling and Smoothing Techniques*, 1999.
- [15] P.M. Knupp, Hexahedral and tetrahedral mesh untangling, *Eng. Comput.* 17 (2001) 261–268.
- [16] M.L. Staten, et. al., A Comparison of Mesh Morphing Methods for 3D Shape Optimization, in: *Proceedings of the 2011 International Meshing Roundtable*, 2011.
- [17] P. Vachal, R.V. Garimella, M.J. Shashkov, Untangling of 2D meshes in ALE simulations, *J. Comput. Phys.* 196 (2) (2004) 627–644.
- [18] Q.W. Ma, G.X. Wu, R.E. Taylor, Finite element simulation of fully non-linear interaction between vertical cylinders and steep waves. Part 1: methodology and numerical procedure, *Int. J. Numer. Methods Fluids* 36 (3) (2001) 265–285.
- [19] G. Irving, J. Teran, and R. Fedkiw, Invertible Finite Elements for Robust Simulation of Large Deformation, Presented at the ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA), Edited by R. Boulic and D. Pai, 2004, pp. 131–140.
- [20] A. Stomakhin, R. Howes, C. Schroeder, J. Teran, Energetically Consistent Invertible Elasticity, presented at the ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA), 2012, pp. 25–32.
- [21] J. Fish, S. Markolefas, R. Guttal, P. Nayak, On adaptive multilevel superposition of finite element meshes for linear elastostatics, *Appl. Numer. Math.* 14 (1–3) (1994) 135–164.

- [22] J. Danczyk, K. Suresh, Finite element analysis over tangled meshes, in: Proceedings of the ASME 2012 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Chicago, IL, 2012.
- [23] B.H.V. Topping, et al., Finite Element Mesh Generation, Saxe-Coburg Publications, Stirling, UK, 2002.
- [24] W. Han, A posteriori error analysis via duality theory, Springer, New York, NY, 2005.
- [25] G. Strang, An Analysis of the Finite Element Method, Prentice Hall, Englewood Cliffs, NJ, 1973.
- [26] J. Shewchuk, Adaptive precision floating-point arithmetic and fast robust geometric predicates, *Discrete Comput. Geometry* 18 (3) (1997) 305–363.
- [27] R.D. Cook, Concepts and Applications of Finite Element Analysis, John Wiley, New York, 1989.
- [28] G.H. Golub, Matrix Computations, Johns Hopkins, Baltimore, 1996.