On Why Mesh Untangling May Not be Required

Bhagyashree Prabhune* Krishnan Suresh†

Abstract

If a finite element mesh contains elements with negative Jacobian, it is said to tangled. Tangled meshes lead to erroneous results during finite element analysis. Consequently, many untangling methods have been proposed; however, untangling is not always achievable. To address this challenge, the authors recently introduced the isoparametric tangled finite element method (i-TFEM), allowing the use of tangled meshes in finite element analysis. By introducing minor modifications to standard FEM, i-TFEM offers an easy implementation and reduces to standard FEM for non-tangled meshes. It efficiently handles complex configurations of tangled elements, making it suitable for real-world scenarios, including linear/non-linear elasticity, free and forced vibrations, and thermal for quadrilateral, hexahederal and higher order triangular elements. Numerical experiments demonstrate the accuracy and applicability of the method to real-world tangled meshes. The numerical results also emphasizes the importance of reevaluating the notion of mesh quality for tangled meshes.

1 Introduction

Automatic high-quality hexahedral mesh generation remains an open challenge [4, 17]. The underlying reasons are due to the severe topological and geometric constraints imposed [26]: (a) the mesh must not be tangled, (b) the elements must be of high quality, (c) the mesh must conform to the geometry, and (d) the mesh must be topologically well-structured. Satisfying all these requirements is non-trivial. The focus of this work is on the first constraint that requires that the Jacobian determinant must remain positive over the entire mesh. It is well known that finite element analysis over a tangled mesh will lead to erroneous results.

To address tangled elements, numerous untangling algorithms have been developed [11, 28, 15]. However, untangling is not always guaranteed [15, 1, 28, 23]. Several non-traditional finite element techniques [29, 6, 13] including the virtual element method [3] have been developed to directly handle some of the tangled mesh configurations. However, these methods cannot handle some of the tangled meshes generated by current mesh generators. For instance, practical meshes generated from state-of-the-art mesh generation algorithms, often contain 'self-penetrating' tangled elements [22], which are not addressed by these approaches. Moreover, these methods may require significant changes to the FEM framework.

Recently, an isoparametric tangled finite element method (i-TFEM) [20, 21, 19, 18, 22] was proposed which specifically addresses tangled elements (mesh requirement 1) by modifying traditional FEM. I-TFEM reduces to standard FEM for non-tangled meshes and requires minimal changes to the existing FEM framework. It can effectively handle real-world tangled meshes generated by the state-of-the-art mesh generation methods such as Polycube mapping [5, 16, 8] and frame-field [10] based approaches.

I-TFEM has been demonstrated for linear and non-linear elasticity, free and forced vibrations, and Poisson problems over meshes containing partially inverted (tangled) triangular, quadrilateral and hexahederal elements. This work reports some of these recent results. Note that i-TFFEM specifically deals with tangled meshes characterized by partially inverted elements, i.e. elements with negative Jacobian determinant at some (not all) Gauss points.

2 Isoparametric Tangled Finite Element Method (i-TFEM)

The detailed mathematical framework for i-TFEM is presented in [22, 21]; here, we provide a brief summary.

Recall that, for a typical boundary value problem, such as a linear elasticity problem, FEM reduces to solving the following linear system of equations:

$$\boldsymbol{K}\boldsymbol{u} = \boldsymbol{f} \tag{2.1}$$

where, $\boldsymbol{u}$ is the unknown nodal displacement vector, $\boldsymbol{f}$ are the nodal loads, and $\boldsymbol{K}$ is the stiffness matrix. When the mesh is tangled, the solution to this system leads to inaccurate results [22, 21]. To illustrate, consider the tangled (non-convex) 8-node hexahedral element in the physical space (x_1, x_2, x_3) in Fig. 1a. The Fig. 1b illustrates the parametric space (ξ_1, ξ_2, ξ_3) associated with the tangled element. Due to the non-convex nature of the element, one can show that [20] the parametric

*Oak Ridge National Laboratory (this work was conducted at the previous affiliation: University of Wisconsin-Madison).

†University of Wisconsin-Madison.

space can be divided into positive and negative regions, denoted by J^+ and J^- respectively. The corresponding regions in the physical space are referred to as positive (C^+) and negative (C^-) components respectively; see Fig. 1c. Furthermore, the parametric points $\boldsymbol{a} \in J^-$ and $\boldsymbol{b} \in J^+$ map to the same point $\boldsymbol{p}$ in the physical space; such physical points $\boldsymbol{p}$ lie outside the element. In other words, an overlapping region or a fold F is created (see Fig. 1c). Since standard FEM does not handle this inverted region correctly, it leads to erroneous results.

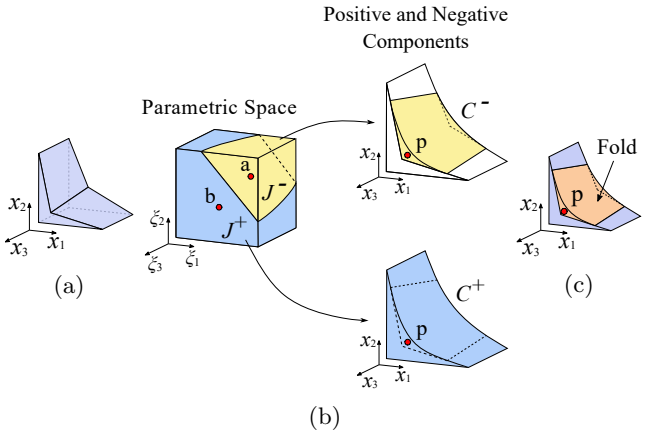


Figure 1: (a) Physical space of the tangled H8 element. (b) Parametric space of the tangled element, that can be divided into positive and negative Jacobian regions. Corresponding physical space with positive and negative components. (c) Tangled element with the overlapping region.

In i-TFEM, the positive and negative parametric regions J^+ and J^- are treated separately. In other words, the non-invertible mapping is replaced with piecewise invertible mapping. Next, to construct the stiffness matrix, we define two shape functions $\boldsymbol{N}^+$ and $\boldsymbol{N}^-$ corresponding to C^+ and C^- respectively. Thus, for any point $\boldsymbol{p}$ in the fold F that belongs to both C^+ and C^-, one can define two fields $\boldsymbol{N}^+(\boldsymbol{p})\,\boldsymbol{u}$ and $\boldsymbol{N}^-(\boldsymbol{p})\,\boldsymbol{u}$ due to the fold. To resolve this ambiguity, in i-TFEM, a *piecewise compatibility constraint* is enforced over the fold (see [20] for further discussion):

$$\boldsymbol{N}^+(\boldsymbol{p})\boldsymbol{u} = \boldsymbol{N}^-(\boldsymbol{p})\boldsymbol{u}, \quad \forall \boldsymbol{p} \in F. \tag{2.2}$$

Incorporating the piecewise invertible mapping and compatibility constraint, i-TFEM involves solving the following system of equations:

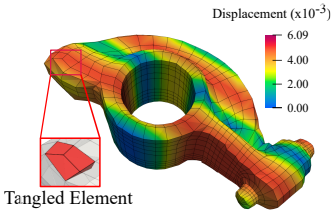
$$\begin{bmatrix} \boldsymbol{K} & \boldsymbol{C} \\ \boldsymbol{C}^\top & \boldsymbol{0} \end{bmatrix} \begin{Bmatrix} \boldsymbol{u} \\ \boldsymbol{\lambda} \end{Bmatrix} = \begin{Bmatrix} \boldsymbol{f} \\ \boldsymbol{0} \end{Bmatrix} \tag{2.3}$$

Here, the stiffness matrix $\boldsymbol{K}$ and the forcing term $\boldsymbol{f}$ are computed exactly as in the standard FEM, albeit using the signed value of the Jacobian (many commercial FE software use the absolute value of the Jacobian). In addition, a constraint matrix $\boldsymbol{C}$ is required that enforces the field compatibility [21]. For an untangled/untangle-free meshes, the constraint matrix $\boldsymbol{C}$ does not exist and i-TFEM reduces to the standard FEM.

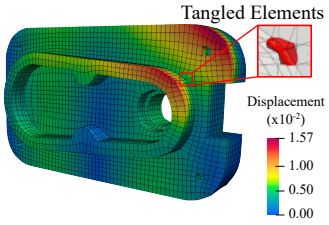
3 Results

In this section, we consider the practical tangled meshes, generated by the state-of-the-art hex meshing algorithms, and demonstrate the application of i-TFEM to solve linear elasticity and free vibration problems.

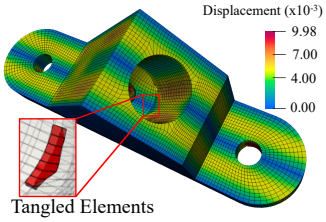
3.1 Real-world meshes We demonstrate the application of i-TFEM for some of the real-world tangled meshes generated using a range of state-of-the-art mesh generators (and available at [2]).



(a) Rocker arm [27]: Rel. L^2 Err = 2.37×10^{-2}
Min Scaled Jacobian = -0.189



(b) Demo09 [9]: Rel. L^2 Err = 4.00×10^{-2}
Min Scaled Jacobian = -0.790

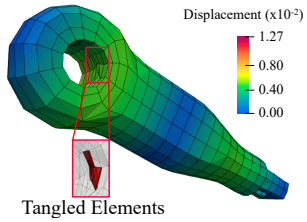


(c) i02u_m2 [2]: Rel. L^2 Err = 4.42×10^{-1}
Min Scaled Jacobian = -0.412

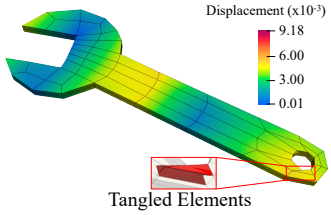
Figure 2: Displacement plots obtained via i-TFEM for practically occurring tangled meshes; the red elements in the inset are tangled.

For instance, the mesh in Fig. 2b is generated using a PolyCube based approach [16], mesh in Fig. 2c is created using frame-field based approach [2]. Fig. 2a is

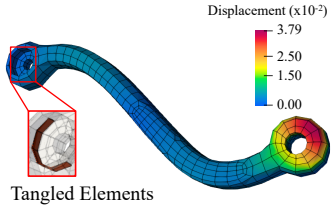
produced semi-manually using dual-sheet modeling [27] while automatic block decomposition [14] is employed for meshes in Fig. 3.



(a) Rod [14]: Rel. L^2 Err $= 4.89 \times 10^{-2}$
Min Scaled Jacobian = -0.593



(b) wrench [14]: Rel. L^2 Err $= 9.95 \times 10^{-2}$
Min Scaled Jacobian = -0.280



(c) lever arm [14]: Rel. L^2 Err $= 6.31 \times 10^{-2}$
Min Scaled Jacobian = -0.944

Figure 3: Displacement plots obtained via i-TFEM for tangled meshes obtained via [14]; the red elements in the inset are tangled.

Using the tangled meshes in Fig. 2 and 3, i-TFEM was employed to solve linear elasticity problem with a synthetic solution [22]

$$u_1 = \zeta_1^3\zeta_2\zeta_3^2 + 2\zeta_1^2\zeta_2^3\zeta_3^2 + 0.5\sin(2\pi\zeta_1); \tag{3.4a}$$

$$u_2 = \zeta_1^2\zeta_2^3\zeta_3 + 2\zeta_1^2\zeta_2^2\zeta_3^3; \tag{3.4b}$$

$$u_3 = \zeta_3^3 + 2\zeta_1^3\zeta_2^2\zeta_3^2/100 \tag{3.4c}$$

where ζ_i are computed by dividing each component x_i by its corresponding length-scale L_i, i.e., $\zeta_i = x_i/L_i$.

The problem is solved with the material properties, $E = 400/3$, $\nu = 1/3$. Dirichlet boundary condition is imposed over the entire boundary. The resultant displacement fields are shown in Fig. 2 and 3. As one can observe, despite numerous tangling elements, the L^2 error is within the acceptable/nominal range.

Table 1: Time comparison with i-TFEM and FEM.

Model	$\lvert\boldsymbol{J}\rvert_{\min}$	Time (s)	
		iTFEM	FEM (incorrect)
i02u_m2 [2]	-0.41	95.68	92.72
rod [27]	-0.59	0.53	0.52

i-TFEM was also employed to compute the vibration modes over these tangled meshes. Table 1 provides the computational time required for i-TFEM as well as FEM (albeit with inaccurate results). Observe that the computational overhead due to i-TFEM is within 3% of the FEM simulation time.

3.2 Comparison of tangled and untangled meshes Next, we consider that the tangled meshes which have been successfully untangled using the untangling algorithm provided in [15]. Some of these meshes are shown in Fig. 4a and Fig. 4c where the tangled elements are highlighted in red color. These are generated using multi-sweep [24] and PolyCube mapping based [8] methods respectively. Although it may be feasible to these meshes [15], the use of i-TFEM eliminates the need for untangling.

First, the linear elasticity problem with the synthetic solution (Eq. 3.4) is solved using the two methods: (a) i-TFEM for tangled meshes and (b) standard FEM for untangled meshes. The results, presented in Table 2, reveal that the L^2 error norm obtained via i-TFEM over tangled meshes is comparable with that obtained over the corresponding untangled meshes. Moreover, the time required to handle tangled elements is minimal. This suggests that i-TFEM can provide solutions with comparable accuracy over tangled meshes, thus eliminating the need for mesh untangling.

Finally, free vibration problem is solved (a) over tangled meshes using i-TFEM (see Fig. 4) and (b) over the corresponding untangled meshes using FEM; the comparison of the first natural frequency is provided in Table 3. It can be observed that the natural frequencies obtained from both methods are comparable, and the additional computational time required for i-TFEM is minimal. The time required to untangle the mesh is not included.

4 Mesh Quality Indicators for Tangled Meshes

Traditionally, tangled elements are deemed to be invalid, and/or assigned a quality of zero [12]. However, using i-TFEM, tangled meshes can provide comparable or even superior accuracy compared to regular meshes.

Table 2: Comparison of solutions over tangled and untangled meshes provided in [15].

Model [15]	$\lvert\boldsymbol{J}\rvert_{\min}$	Relative L^2 error $\times 10^{-2}$	Time (s)
linking rod tangled	-0.39	0.21	10.46
linking rod untangled	0.55	0.21	10.17
block tangled	-0.70	1.58	2.02
block untangled	0.25	1.58	1.87
cap tangled	-0.94	4.23	4.73
cap untangled	0.11	4.36	4.13
bust tangled	-0.60	3.71	5.15
bust untangled	0.11	3.93	5.11

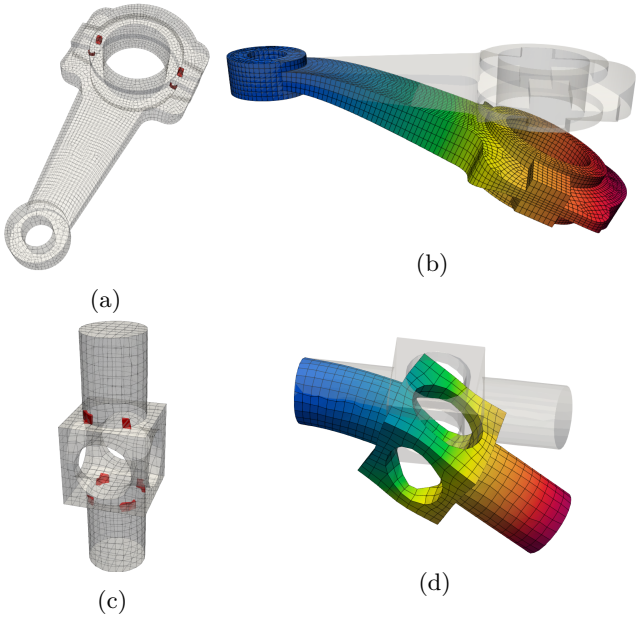
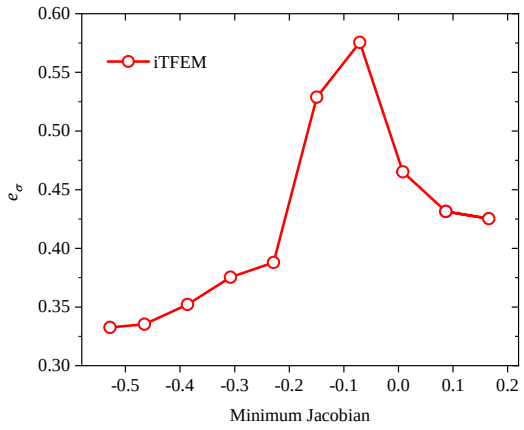


Figure 4: Tangled mesh of (a) linking rod and (b) block provided by [15]. Tangled elements are highlighted in red.

Table 3: Comparison of solutions for free vibration problem over tangled and untangled meshes provided in [15].

Model [15]	$\lvert\boldsymbol{J}\rvert_{\min}$	Fund. freq. (Hz)	Time (s)
linking rod tangled	-0.39	6.370	17.17
linking rod untangled	0.55	6.361	17.14
block tangled	-0.70	1542	3.55
block untangled	0.25	1547	3.52



(a)

Figure 5: Plot of (a) displacement error (b) stress error vs minimum Jacobian.

To evaluate the effect of mesh quality on i-TFEM solution, we consider a 3D torsion problem solved over a synthetically highly tangled hexahederal mesh (see [22] for more details). In Fig. 5a, we plot the relative L^2 norm error in stress as a function of the minimum Jacobian within the mesh. As one can observe, accuracy improves as we move away from zero Jacobian, whether in the positive or negative direction. Similar observations can be drawn from the data provided in Table 2. 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.

5 Conclusion

The isoparametric tangled FEM (i-TFEM) can potentially address the challenge of handling *tangled* meshes that are conventionally considered unacceptable due to the erroneous results they produce. Real-world tangled meshes obtained from a wide range of mesh generators could be efficiently handled by i-TFEM. The results emphasize the need for a new definition of mesh quality that accommodates tangled elements, and potentially a new class of mesh generators. Further research relating mesh quality and solution accuracy [7, 25] is required in the context of tangled meshes.

Acknowledgements

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

References

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

[20] ———, *A computationally efficient isoparametric tangled finite element method for handling inverted quadrilateral and hexahedral elements*, Computer Methods in Applied Mechanics and Engineering, 405 (2023), p. 115897.

[21] ———, *An isoparametric tangled finite element method for handling higher-order elements with negative jacobian*, Computational Mechanics, (2023), pp. 1–18.

[22] ———, *On why mesh untangling may not be required*, Engineering with Computers, (2023), pp. 1–18.

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

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

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

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

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

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

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