Exploring Tie Constraints for Structural Analysis Problems

Paul R. Mokotoff [*] and John F. Dannenhoffer, III [†]
Aerospace Computational Methods Laboratory, Syracuse University, Syracuse, NY, 13244

When generating a conformal mesh for Finite Element Analysis (FEA), careful attention is required to ensure that the nodes along an interface exactly match each other. This is true when various components are meshed independently. But even if the structure is meshed as a whole, there still may be components with an undesirable mesh resolution. Time and computing power are wasted by iterating through different mesh configurations until an acceptable one is found. To overcome these issues, it would be better to mesh each component independently of the others. Tie constraints are utilized to connect these non-conformal meshes.

This research assesses how the implementation of tie constraints affects the accuracy, robustness, and efficiency of solving FEA problems in two-dimensions. To accomplish this, a finite element code, PRoFile, is written in C. A static structural analysis is performed on three types of beams: one in axial tension, one cantilever, and one simply supported. With meshes of varying resolution, each configuration is run through PRoFile and compared to the solution found when no ties are used. It is shown that, if implemented properly, the tying schemes have little negative impact on the accuracy of the solution but increase the time required to obtain a solution from PRoFile.

The results from this investigation will be useful for the analysis of aircraft designs involving components from multiple parties. Typically, all components of the aircraft must be gathered and meshed together. It would be easier if each party were to make a mesh for their respective aircraft components and then tie all of the meshes together before the structure undergoes FEA. That way, if one component is re-designed, the entire aircraft does not need to be re-meshed - only the single component is re-meshed and the ties to the other components on the aircraft are updated.

I. Nomenclature

ESP	=	Engineering Sketch Pad
FEA	=	Finite Element Analysis
QUAD	=	quadrilateral element
TRI	=	triangular element
A	=	cross-sectional area
$\boldsymbol{C}$	=	tie constraint equations
E	=	modulus of elasticity
$\boldsymbol{F}$	=	force vector
F_i	=	force at node i
$\boldsymbol{K}$	=	stiffness matrix
$\boldsymbol{u}$	=	displacement vector
u_i	=	displacement at node i
W_i	=	weight of node i on the parent node
λ_i	=	force in rigid element i

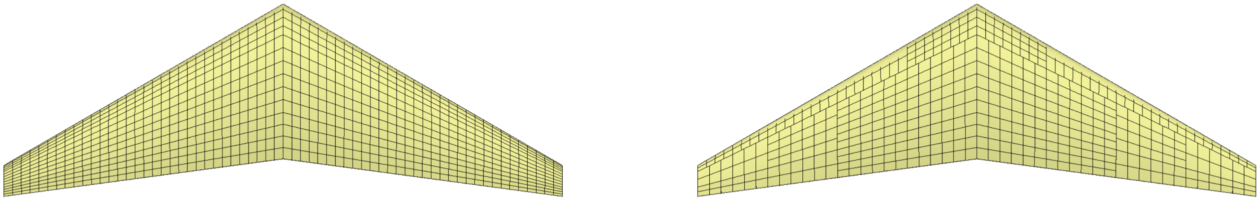
II. Motivation

During the pre-processing of an FEA problem, a conformal mesh is usually generated. This means that two adjacent components in a structure share the same nodes along a common edge. If the components have a similar mesh resolution,

[*] Graduate Student, Department of Mechanical & Aerospace Engineering, 263 Link Hall, Syracuse, NY 13244, AIAA Student Member.

[†] Associate Professor, Department of Mechanical & Aerospace Engineering, 263 Link Hall, Syracuse, NY 13244, AIAA Associate Fellow.

the nodes at the interface can be matched together easily. If the components have different mesh resolutions, additional efforts are required to ensure that the nodes match along the interface. A consequence of this is that some components will have an unreasonably sized mesh. For example, the mesh generated for the tapered wing in Figure 1a contains the same number of nodes along the root and wingtips, which creates an unnecessarily fine mesh towards the outboard regions of the wing. This happens as a result of the requirement that the nodes need to match along an interface located halfway between the root and the wingtip. Since there are regions with more nodes and elements than needed, additional time and computing power are required to solve the FEA problem. If it was determined that the mesh was too coarse near the inboard regions of the wing (where the maximum stress is likely to occur), an even finer mesh would be generated and the elements at the outboard regions of the wing would become even smaller.



(a) Conformal mesh.

(b) Non-conformal mesh.

Fig. 1 Mesh configurations for a tapered wing in ESP [1].

Instead, consider the non-conformal mesh generated for the tapered wing in Figure 1b. Here, each component was meshed independently of all the others. As a result, less time is required to obtain an acceptable mesh; the user can specify a unique mesh resolution on each component. However, there are multiple sets of nodes along a common edge - one for each component that shares said edge. In order for the FEA problem to be solved, the individual meshes must be pieced together such that they act like one cohesive mesh. To combine these disparate meshes together, tie constraints are introduced along the edges of the structure.

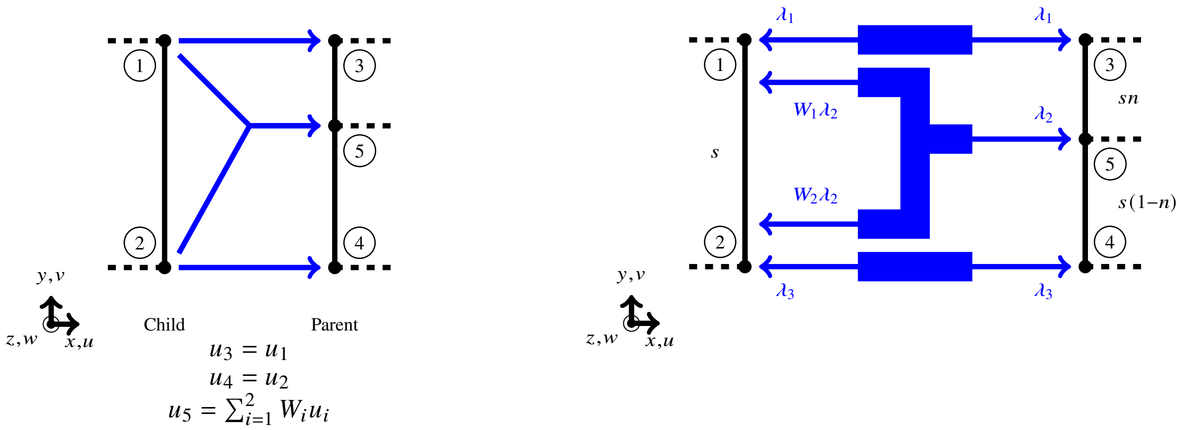
III. Tie Constraints

A tie constraint connects two or more meshes together at a user-prescribed set of nodes. In practice, tie constraints are applied along the edges or faces of components that are assembled together, such as a wing-fuselage interface. In this research, tie constraints were used to connect nodes along an edge shared by two components. For any tie constraint, there are two types of nodes: the "parent" node and the "child" node(s). When tying two edges together, all the nodes on one edge are the parents and those on the other edge are the children. As done by Quiroz and Beckers [2], the edge with the most nodes is selected as the parent. If all edges have the same number of nodes, either edge is a suitable parent. A simple set of tie constraints is imposed on the edges as shown in Figure 2a. Horizontal space was added in-between the edges to illustrate the tie constraints. In a computer model of this structure, the edges would be coincident.

A. Generating Tie Constraint Equations

A set of tie constraints is denoted by the blue arrows in Figure 2a. Since Nodes 1 and 3 are coincident and Nodes 2 and 4 are coincident, their displacements must be exactly the same. In contrast, Node 5 lies between Nodes 1 and 2. Therefore, the displacement of Node 5 will be a weighted sum of the displacements from Nodes 1 and 2. It is required that each weight, W_i, be greater than 0 and less than 1. Also, the sum of all weights at each node must add to 1. Therefore, the governing equation of a tie constraint requires that the displacement of the parent node must be equal to the weighted sum of the children node(s). With the introduction of additional governing equations to the FEA problem, there must be additional unknowns to solve for.

The additional unknowns introduced into the system are the forces holding the tied nodes together, λ_i. Without these forces, the structure would fail to remain connected and part of the structure may undergo rigid-body motion. Physically speaking, each tie constraint represents a rigid element connecting the nodes set forth in the governing equation. These rigid elements are seen in Figure 2b. (Note that the forces $W_1\lambda_2$ and $W_2\lambda_2$ act at Nodes 1 and 2, respectively. Vertical



(a) Tie constraint schematic.

(b) Tie constraints as rigid elements.

Fig. 2 Simple implementation of tie constraints using two edges.

space was added to easily view each rigid element in the tying scheme.) For the nodes that are coincident, a two-force body connects the nodes. For the parent node that lies between two children, the forces on the rigid element must be balanced such that it remains in translational and rotational equilibrium. The weights used to govern the displacement of the parent node are also used to distribute the unknown force between the two children nodes.

To find weights W_1 and W_2, let the distance between Nodes 3 and 4 be s. Node 5 lies at any point between Nodes 3 and 4. To quantify the distance between Nodes 3 and 5, a variable n is introduced to represent the fraction of s between the two nodes. Thus, the distance between said nodes will be sn. This means that the distance between Nodes 4 and 5 must be $s(1-n)$. To solve for the weights at Nodes 1 and 2, require that the rigid element remains in translational and rotational equilibrium (moment taken about Node 5), as stated in Equations 1 and 2, respectively:

$$\lambda_2 = W_1\lambda_2 + W_2\lambda_2 \tag{1}$$

$$0 = (sn)W_1\lambda_2 - s(1-n)W_2\lambda_2 \tag{2}$$

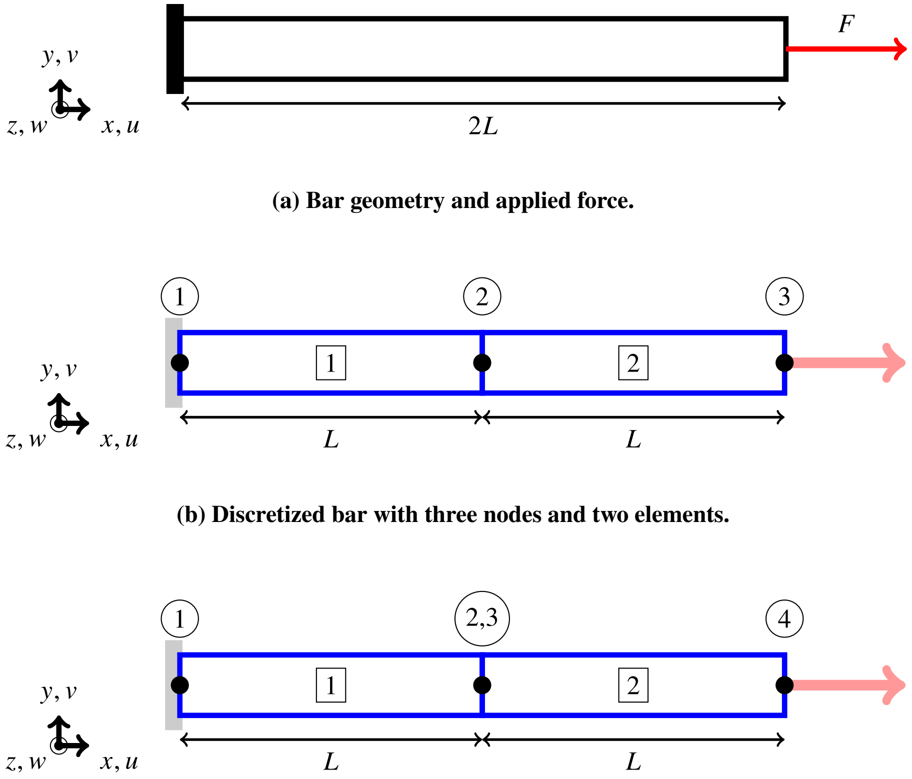
After solving Equations 1 and 2 simultaneously, $W_1 = 1 - n$ and $W_2 = n$. Thus, the equation governing the tie constraint involving Node 5 (parent) and Nodes 1 and 2 (children) is: $u_5 = (1-n)u_1 + nu_2$.

B. Solving Tie Constraint Equations

In an FEA problem without tie constraints, the user is solving a system of equations resemblant of Hooke's Law: $\boldsymbol{Ku} = \boldsymbol{F}$. For each node in the structure, Hooke's Law is formulated based on its element's geometry and material properties. All of these governing equations are assembled into a stiffness matrix, $\boldsymbol{K}$, and a force matrix, $\boldsymbol{F}$, as shown in Equation 3. After the stiffness and force matrices are set, the displacements at each node can be obtained by solving $\boldsymbol{Ku} = \boldsymbol{F}$ for $\boldsymbol{u}$. In PRoFile, the displacements were computed using the Bi-Conjugate Gradient Method [3]. The Bi-Conjugate Gradient Method is similar to the Conjugate Gradient Method except that it allows a solution to be obtained for a system in which the coefficients form a semi-positive definite matrix. Without any tie constraints, the stiffness matrix is positive definite. Once tie constraints are added to the system, the stiffness matrix is guaranteed to be semi-positive definite but not always positive definite.

$$\begin{bmatrix} \boldsymbol{K} \end{bmatrix} \begin{bmatrix} \boldsymbol{u} \end{bmatrix} = \begin{bmatrix} \boldsymbol{F} \end{bmatrix} \tag{3}$$

To better understand how the stiffness and force matrices are filled, consider a simple bar in tension that is fixed on its left end, as shown in Figure 3a. The bar has a cross-sectional area, A, and Modulus of Elasticity, E. A force, F, is applied axially in the $+x$ direction. Each element will be L long.



(a) Bar geometry and applied force.

(b) Discretized bar with three nodes and two elements.

(c) Discretized bar with four nodes and two elements.

Fig. 3 Bar in tension.

First, consider the structure without ties in Figure 3b. By setting $k = AE/L$, the governing equations can be assembled into the matrix shown in Equation 4. Note that $u_1 = 0$ because it is fixed at the wall and F_1 will be found after the remaining displacements are solved for.

$$\begin{bmatrix} k & -k & 0 \\ -k & 2k & -k \\ 0 & -k & k \end{bmatrix} \begin{bmatrix} 0 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} F_1 \\ 0 \\ F \end{bmatrix} \tag{4}$$

After solving Equation 4, the displacements at each node are:

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} 0 \\ \frac{F}{k} \\ \frac{2F}{k} \end{bmatrix} \tag{5}$$

For each tie constraint, its governing equation is appended as an extra row to the stiffness matrix, $\boldsymbol{C}$. The forces holding the structure together at the tied interface must also be accounted for and are appended as columns to the stiffness matrix. The additional forces introduced to the system can be represented by the transpose of the tie constraint equations [4]. Additional entries are also added to the displacement and force matrices to solve for the unknown forces in each rigid element. The new system is shown in Equation 6.

$$\left[\begin{array}{c|c} \boldsymbol{K} & \boldsymbol{C}^T \\ \hline \boldsymbol{C} & \boldsymbol{0} \end{array}\right] \left[\begin{array}{c} \boldsymbol{u} \\ \hline \lambda \end{array}\right] = \left[\begin{array}{c} \boldsymbol{F} \\ \hline \boldsymbol{0} \end{array}\right] \tag{6}$$

The tie constraint equation governing the motion of Nodes 2 and 3 in Figure 3c is $u_2 = u_3$ or $-u_2 + u_3 = 0$. The convention in PRoFile is to negate the displacement of the parent. Therefore, in this example, Node 2 is the parent. Upon including the tie constraint equation and the forces holding the nodes together, the stiffness matrix becomes:

$$\left[\begin{array}{cccc|c} k & -k & 0 & 0 & 0 \\ -k & k & 0 & 0 & -1 \\ 0 & 0 & k & -k & 1 \\ 0 & 0 & -k & k & 0 \\ \hline 0 & -1 & 1 & 0 & 0 \end{array}\right] \left[\begin{array}{c} 0 \\ u_2 \\ u_3 \\ u_4 \\ \hline \lambda \end{array}\right] = \left[\begin{array}{c} F_1 \\ 0 \\ 0 \\ F \\ \hline 0 \end{array}\right] \tag{7}$$

One issue arises with the formulation of the stiffness matrix. On the lower right partition of the stiffness matrix, there is a zero along the main diagonal. This will pose an issue once the Bi-Conjugate Gradient solver is used. To overcome this issue, the row containing the tie constraint equation can be swapped with the row containing the governing equation for the parent node. After swapping rows 2 and 5, the new stiffness matrix is:

$$\left[\begin{array}{cccc|c} k & -k & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & k & -k & 1 \\ 0 & 0 & -k & k & 0 \\ \hline -k & k & 0 & 0 & -1 \end{array}\right] \left[\begin{array}{c} 0 \\ \lambda \\ u_3 \\ u_4 \\ \hline u_2 \end{array}\right] = \left[\begin{array}{c} F_1 \\ 0 \\ 0 \\ F \\ \hline 0 \end{array}\right] \tag{8}$$

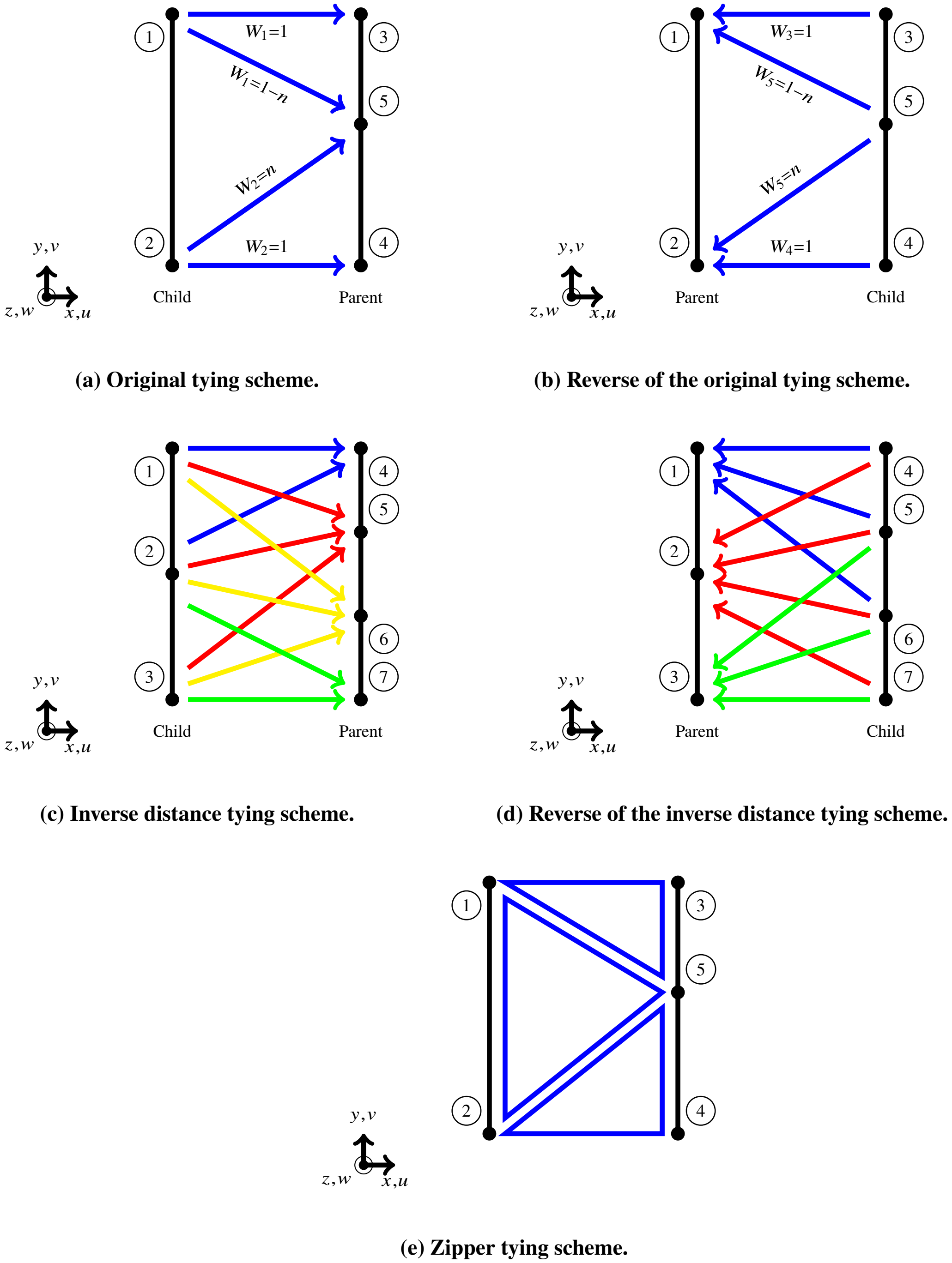
The nodal displacements and the force in the rigid element are found to be:

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ \lambda \end{bmatrix} = \begin{bmatrix} 0 \\ \frac{F}{k} \\ \frac{F}{k} \\ \frac{2F}{k} \\ F \end{bmatrix} \tag{9}$$

As expected, the displacements at Nodes 2 and 3 (the tied nodes) are the same. The force in the rigid element holding Nodes 2 and 3 together is F. Thus, the entire applied force is being transferred to Element 1 through the rigid element connecting Nodes 2 and 3.

C. Tying Schemes Assessed

In this research, five tying schemes were assessed. The first one was outlined at the beginning of this section. Recall that the edge with the larger number of nodes is the parent. If the parent and child nodes are coincident, their displacements must be the same and the child is assigned a weight of 1. For any parent that lies in-between two children, its displacement will be a weighted sum of the children displacements. The children weights can be calculated by $1 - \frac{d}{s}$, where d is the distance between the parent and child and s is the distance between the two children in the tie constraint. This is illustrated in Figure 4a. The second tying scheme reverses the parent-child relationship given by Quiroz and Beckers [2]; the edge with fewer nodes now becomes the parent. The weights between the nodes, however, are the same as the previous scheme. This is illustrated in Figure 4b.



(a) Original tying scheme.

(b) Reverse of the original tying scheme.

(c) Inverse distance tying scheme.

(d) Reverse of the inverse distance tying scheme.

(e) Zipper tying scheme.

Fig. 4 Tying schemes assessed in this paper.

The Inverse Distance scheme [5] involves tying every child node to every parent node within some radius of influence, r. The radius of influence is a user-selected distance that can vary from a small portion of the edge to the entire length of the edge. The weight is determined by how far the child is from the parent. If the parent and child are coincident, the child receives a weight of 1. For non-coincident parent/child nodes, the weight of a child can be found by $1 - \frac{d}{r}$, where d is the distance between the parent and the child and r is the radius of influence. If a child is beyond the radius of influence, it does not receive a weight. Assigning the parent and child edges is the same as that of the first scheme - the edge with the most nodes is the parent. This scheme is illustrated in Figure 4c and the radius of influence spans the entire length of the edge. The fourth scheme is the opposite of the inverse distance. The parent is the edge with fewer nodes and new weights will be generated following the same equation used previously, $1 - \frac{d}{r}$. The radius of influence also spans the entire length of the edge. This scheme is illustrated in Figure 4d.

Instead of tying two edges together by rigid elements, the Zipper scheme ties the edges together by triangular elements (TRIs). In this scheme, the edges are separated by a small distance and TRIs are used to connect the nodes together. There is no specific parent-child relationship or governing equation required in this approach. However, the TRIs are not infinitely stiff like the rigid elements. As a result, the structure will always remain connected but a rigid connection may not be accurately modeled. This scheme is illustrated in Figure 4e.

IV. Configurations

In this research, beams with different loading configurations, interfaces, and mesh resolutions were used. Each beam was 100 in long, 10 in tall, and 1 in thick. The Modulus of Elasticity and Poisson's Ratio were 10×10^6 psi and 0.3, respectively. For later discussions, let the left half of the beam be the inboard portion and the right half of the beam be the outboard portion.

To test the tying schemes, three different loading configurations were used. The first loading configuration was a beam in axial tension. The left edge of the beam was placed on vertical rollers and pinned at its midpoint. A 250,000 lb axial force was applied at the midpoint of the right edge. The second loading configuration was a cantilever beam. The left edge of the beam was placed on vertical rollers and pinned at the bottom left corner. A 10,000 lb traverse force was applied at the bottom right corner. The third loading configuration was a simply supported beam. The bottom left corner was pinned and the bottom right corner was placed on a horizontal roller. A 30,000 lb traverse force was applied at the midpoint of the bottom edge. For all three configurations, the displacement of the beam was measured at the node where the force was applied.

There were three different types of interfaces connecting the two halves of the beam, and are depicted in Figures 5a through 5c. The first interface - the straight interface - is a vertical line. The second interface - the angled interface - is a line angled 45° with respect to the vertical. The third interface - the curved interface - is a semi-circle whose diameter is equal to the height of the beam.

There were six different outboard-inboard mesh configurations used when testing the tie constraints. They are each depicted on a straight interface in Figures 6a through 6f. On all of the configurations, the mesh generated was nominally 100-by-20 QUADs.

V. Results

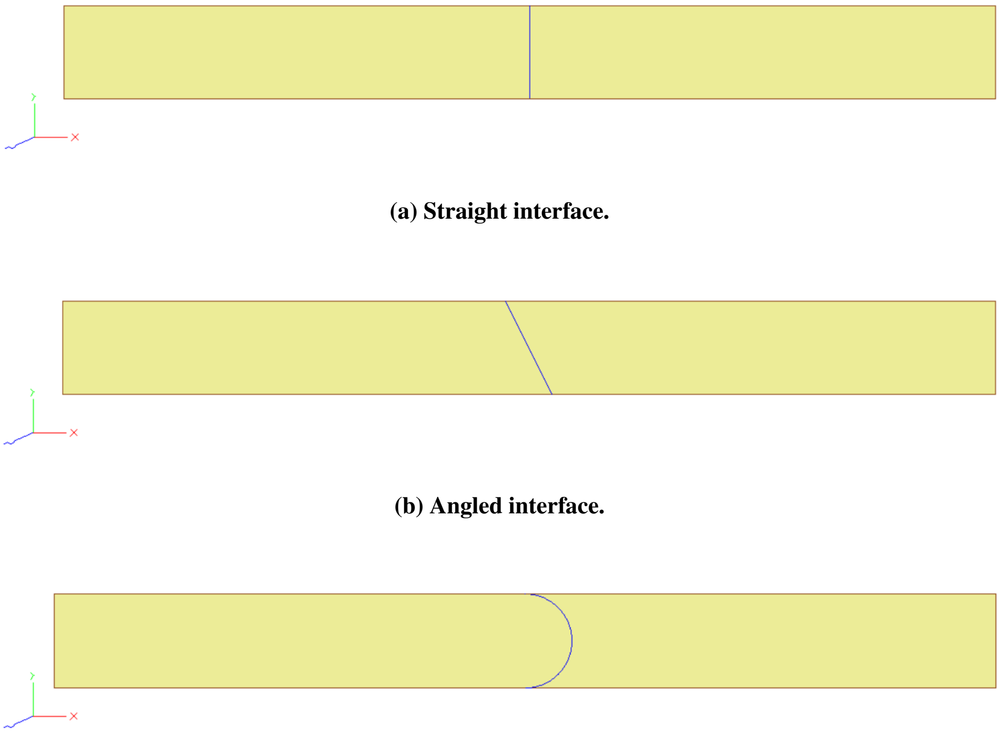
A. Tying Schemes

For each beam and interface, the normalized displacement error is plotted in Figure 7. This was computed using Equation 10, where u_{FEA}, $u_{\text{asymptotic}}$, and l_{beam} are the displacement computed by PRoFile, the asymptotic displacement, and the length of the beam, respectively. The asymptotic displacement is defined as the displacement found when using no ties on a mesh of 1,000-by-100 QUADs.

$$\text{Error} = \frac{\left|u_{\text{FEA}} - u_{\text{asymptotic}}\right|}{l_{\text{beam}}} \tag{10}$$

Note that some of the results obtained were nearly identical and were plotted over each other, which is why it appears that some information is missing from the plot. Also, the results presented for the Inverse Distance and Reverse Inverse Distance tying schemes are for a radius of influence that spans the entire edge.

For the cantilever and simply supported beams, the asymptotic solution is best approached by using the Original, Reverse Original, or Zipper schemes. The normalized error when using these tying schemes is always less than 10^{-3}. As the interface deviates from a straight line, the calculated displacement becomes less accurate. This is most easily



(a) Straight interface.

(b) Angled interface.

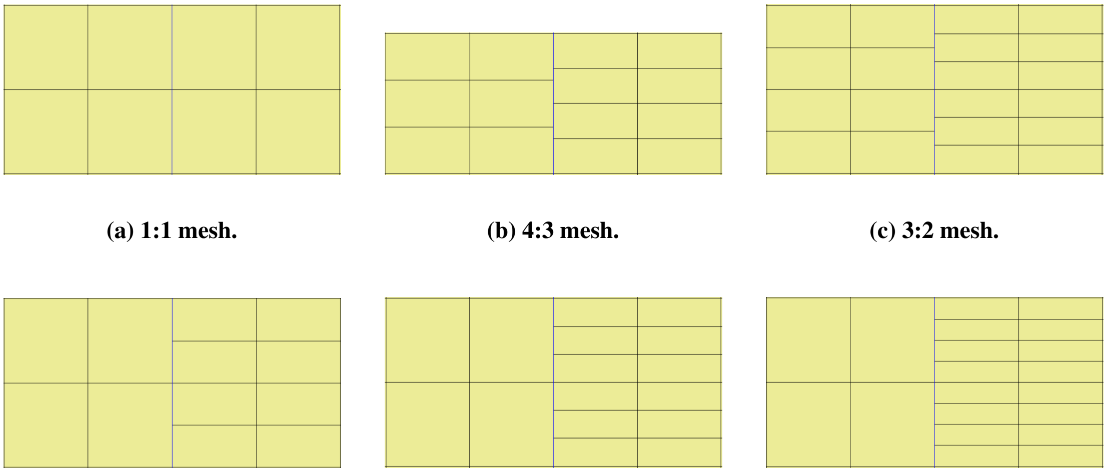
(c) Curved interface.

Fig. 5 Interfaces used to piece the beams together.

seen in the plots for the simply supported beam - the error is smallest for the straight interface and is largest for the curved interface. This trend also holds for the cantilever beam, but is more difficult to see because the scale of the plots is two orders of magnitude larger than that of the simply supported beam. One possible reason for the difference in accuracy is that the meshes generated for the angled and curved interfaces consist of more skew (less rectangular) elements. With skew the elements are at the interface, the solution is likely to be less accurate.

For all of the interfaces on the cantilever and simply supported beams, the Inverse Distance and Reverse Inverse Distance tying schemes are significantly different from the asymptotic solution, with exception to some cases for the curved interface on the simply supported beam. Overall, Inverse Distance and Reverse Inverse Distance tying schemes yield an error that is at least 3 to 4 times larger than the Original, Reverse Original, or Zipper tying schemes (again with the exception of the curved interface on the simply supported beam). The results pertaining to the Inverse Distance and Reverse Inverse Distance schemes may be poor because the radius of influence spanned the entire edge. In Section V.B, results for these two tying schemes will be presented for a varying radius of influence. This will aid in better understanding how effective these tying schemes are in producing an accurate FEA solution.

The results from the beam in axial tension appear to be outliers from the remainder of the data. In these cases, the Reverse Inverse Distance scheme performed the best for the angled and curved interfaces and nearly as well as the other schemes for the straight interface. Looking at the angled interface for the beam in axial tension, the Reverse Inverse Distance scheme yielded an error that is about 3 times smaller than the other tying schemes. Despite this difference, all of the schemes yielded an error less than 4×10^{-4}, which is similar to the results obtained for the Original, Reverse Original, and Zipper schemes on the cantilever and simply supported beams. Although the plot indicates that the Reverse Inverse Distance scheme performed the best, all of the schemes work well for the beam in axial tension. One possible reason for this is that the beam in axial tension displaces the nodes in only the x-direction but the cantilever and simply supported beams displace the nodes in both the x- and y-directions.



(a) 1:1 mesh. **(b) 4:3 mesh.** **(c) 3:2 mesh.**

(d) 2:1 mesh. **(e) 3:1 mesh.** **(f) 4:1 mesh.**

Fig. 6 Mesh configurations used for testing.

In the Appendix, Figure 10 displays a similar set of plots as shown in Figure 7 but for the displacement computed by PRoFile in each test case.

In Figure 8, the required CPU time per element to compute the nodal displacements is plotted. For these results, the asymptotic solution refers to the required CPU time per element to solve a nominal mesh of 100-by-20 QUADs without any tie constraints.

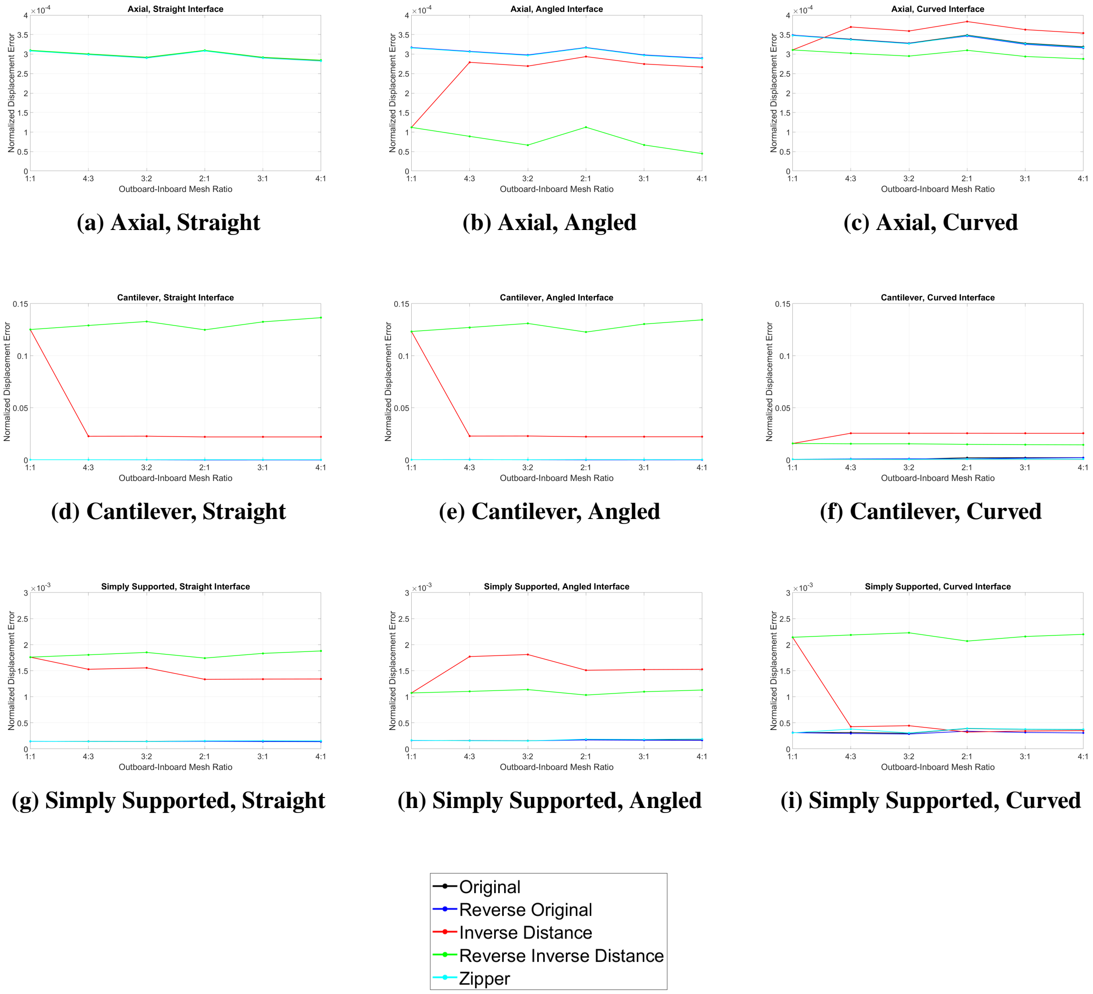
With tie constraints, the time to solve for the displacements always takes longer. For the beam in axial tension, cantilever beam, and simply supported beam, the worst performing cases take 2, 2.5, and 2.5 times longer, respectively, to solve than those without tie constraints. This is because introducing tie constraints to the system increases the condition number of the stiffness matrix and requires additional iterations of the Bi-Conjugate Gradient solver before the solution converges.

As an aside, initial implementations of tie constraints involved using stiff springs instead of rigid elements. However, it was found that the condition number was significantly larger when taking this approach. The condition number grew much less when tie constraints were implemented. To limit the negative impacts on the accuracy and efficiency of the FEA solution, it makes more sense to combine two non-conformal meshes with tie constraints rather than stiff springs.

Overall, the Original and Reverse Original schemes take the longest to solve and the Inverse Distance and Reverse Inverse Distance schemes take the shortest amount of time to solve. However, in most cases, the Original and Reverse Original schemes take about 30% longer to solve than the other schemes. Therefore, in terms of CPU time, there is no advantage to using one tying scheme over the others.

B. Radius of Influence

For the results presented in Section V.A, the radius of influence was always set to span the entire length of the edge. As seen from the plots in Figure 7, this did not produce the most accurate results. To better understand if the Inverse Distance tying scheme yields accurate results, the length of the radius of influence was varied. Here, the radii of influence were 100%, 75%, 50%, 25%, and 10% of the edge length. The normalized displacement error is plotted in Figure 9. Again, the asymptotic displacement is defined as the displacement found when using no ties on a mesh of 1,000-by-100 QUADs. Additional plots detailing the displacement and CPU time required to solve for the nodal displacements are displayed in Figures 11 and 12 in the Appendix.



(a) Axial, Straight **(b) Axial, Angled** **(c) Axial, Curved**

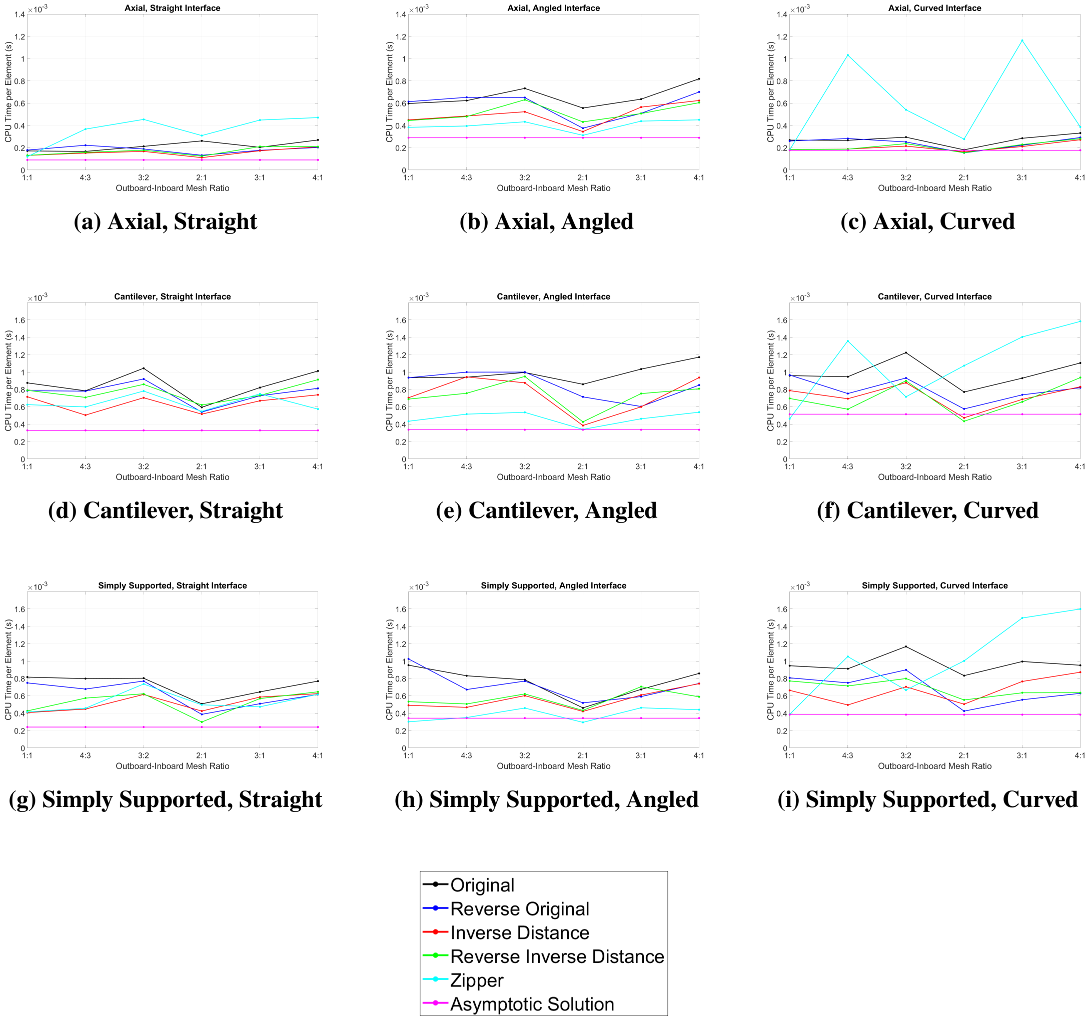
(d) Cantilever, Straight **(e) Cantilever, Angled** **(f) Cantilever, Curved**

(g) Simply Supported, Straight **(h) Simply Supported, Angled** **(i) Simply Supported, Curved**

Fig. 7 Normalized displacement error for all tying schemes.

Based on the plots in Figure 9, a smaller radius of influence yields less error and better approaches the asymptotic solution. For the beam in axial tension, the radius of influence does not significantly affect the error. For the cantilever and simply supported beams, a radius of influence that spans 100% of the edge length yields an error 3 to 4 times larger than a radius of influence that spans 10% of the edge length. The only exceptions to these results are the angled interface for the beam in axial tension and the curved interface for the simply supported beam - a radius of influence spanning the entire edge performed the best.

From the results in this section, it is best to tie children nodes to their closest parent, as done in the Original and Reverse Original schemes. As the radius of influence decreases, the normalized displacement error approaches that of the Original, Reverse Original, and Zipper tying schemes shown in Figure 7. For a smaller radius of influence, only the nodes close to the parent affect its displacement. Thus, there are no influences from nodes farther away on the tied edge. For a larger radius of influence, the nodes farther away from the parent were weighted less but their combined influence on each tie constraint still negatively impacted the overall solution. These results also suggest that the Inverse Distance and Reverse Inverse Distance schemes will perform as well as the Original, Reverse Original, and Zipper schemes as long as the radius of influence is no larger than 25% of the edge length.



(a) Axial, Straight **(b) Axial, Angled** **(c) Axial, Curved**

(d) Cantilever, Straight **(e) Cantilever, Angled** **(f) Cantilever, Curved**

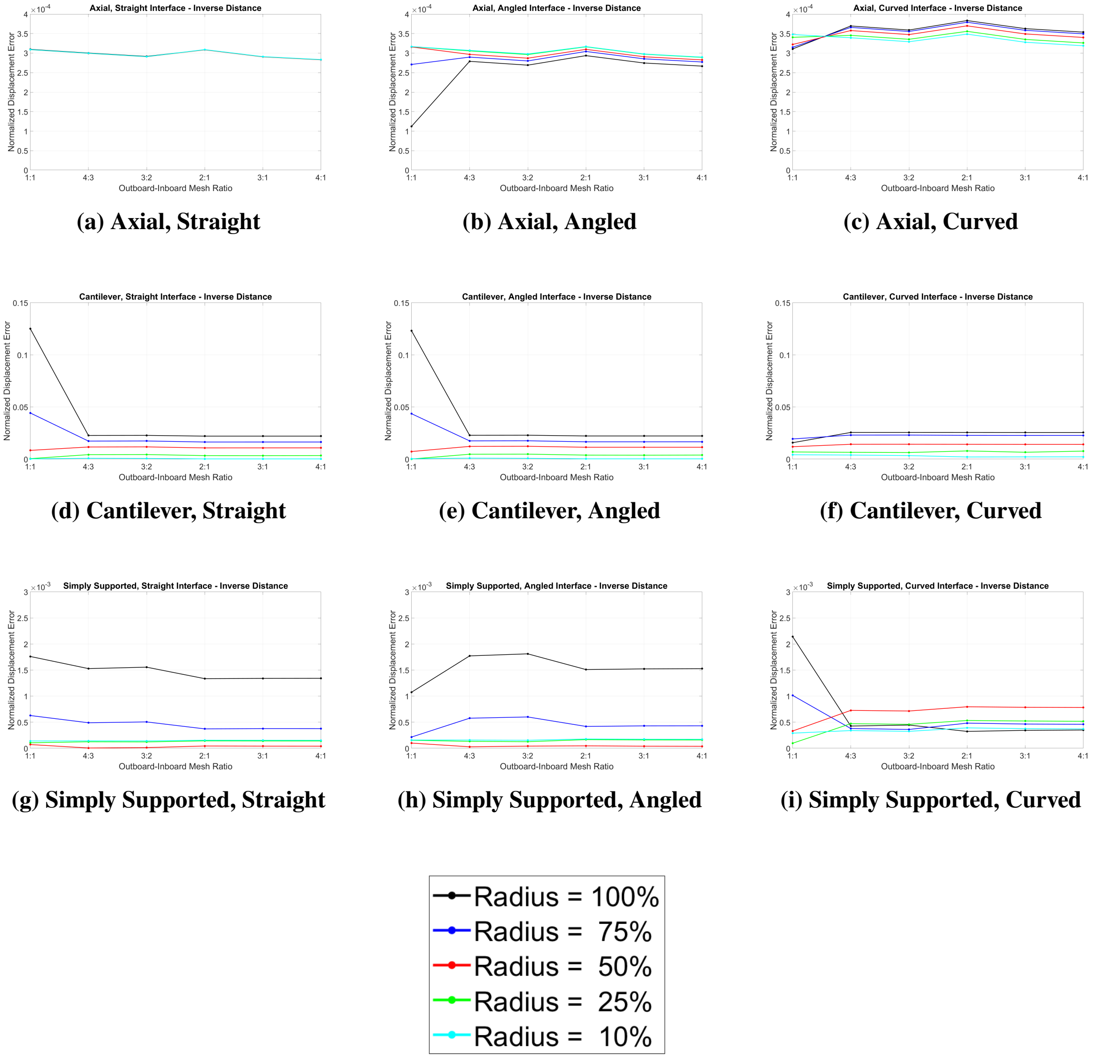
(g) Simply Supported, Straight **(h) Simply Supported, Angled** **(i) Simply Supported, Curved**

Fig. 8 Required CPU time per element for all tying schemes.

VI. Conclusions

The goal of this research was to understand how tie constraints affect the accuracy, robustness, and efficiency of an FEA solution in two-dimensions. There are two main conclusions drawn from this work. The first is that tying multiple meshes together has a limited negative impact on the accuracy of the FEA solution. When using the Original, Reverse Original, and Zipper tying schemes, the error was always less than 10^{-3}. If the radius of influence is no greater than 25% of the edge length, the Inverse Distance and Reverse Inverse Distance schemes will perform similarly to the other schemes. If the radius of influence is larger than 25% of the edge length, the accuracy of the FEA solution is significantly hampered.

The second conclusion is that tying multiple meshes together has a modest negative impact on the efficiency of obtaining an FEA solution. Overall, it takes 2 to 2.5 times longer to solve the FEA problem when tie constraints are used. In terms of CPU time, there are no advantages to using one tying scheme over the other. Despite the efficiency reduction when using tie constraints, consider the entire process of solving an FEA problem: mesh generation, pre-processing, computing the solution, and post-processing. Although computing the solution may take longer, the mesh generation process will be faster because time is not wasted iterating through many unsatisfactory meshes. Thus, it may be better to use tie constraints and save time on the mesh generation but spend more time computing the FEA solution.



(a) Axial, Straight

(b) Axial, Angled

(c) Axial, Curved

(d) Cantilever, Straight

(e) Cantilever, Angled

(f) Cantilever, Curved

(g) Simply Supported, Straight

(h) Simply Supported, Angled

(i) Simply Supported, Curved

Fig. 9 Normalized displacement error when varying the radius of influence.

It is important to note that these results are only valid for two-dimensional structures. This is expanded upon in Section VII, which describes the next steps that should be taken to further generalize the findings in this paper.

These results are useful for analyzing aircraft designs that involve components from multiple parties. Instead of generating a single mesh on the entire structure, each party can mesh their respective component(s). Then, each of the components can be tied together before running the structural analysis. By meshing the components separately, the design of one component can be updated without re-meshing the entire configuration - only the modified component is re-meshed and tied back to the structure.

VII. Future Work

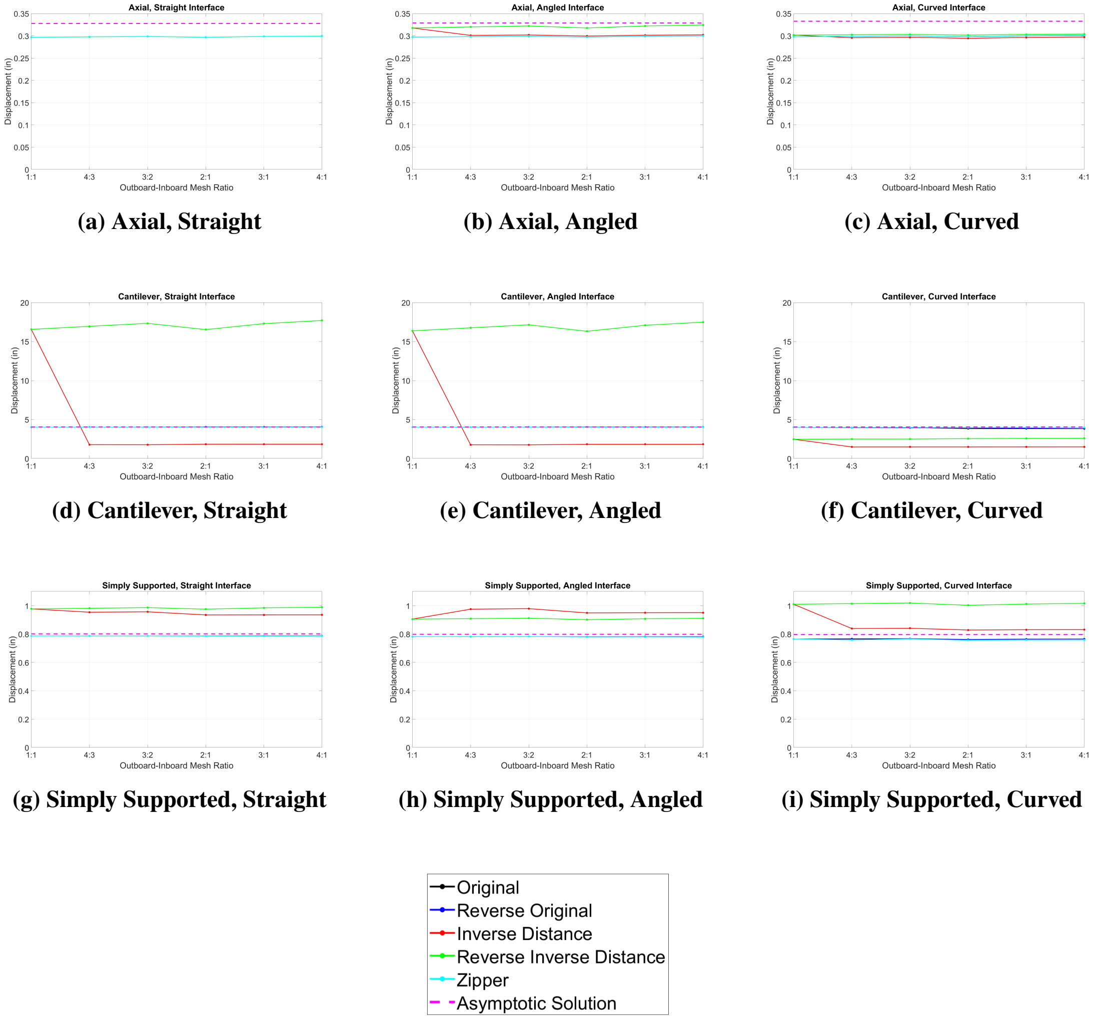
In the future, PRoFile should be expanded to accommodate three-dimensional configurations and ties. One limitation of PRoFile is that only two-dimensional structures have been examined and the interfaces assessed are a line or simple curve. The current tying schemes are guaranteed to work in two dimensions, but will need to be re-assessed when working in three dimensions. The original tying scheme may need to be modified because it currently searches for parents in-between two children along an edge. This process will need to be updated for traversing an entire face. The

Inverse Distance scheme will likely remain operational as it searches for all points within some radius of influence (or sphere of influence in three-dimensions).

Another area of future work involves performing a modal analysis on the beams. This will provide further insights as to whether tie constraints affect the accuracy of the FEA solution for dynamic simulations. Similar to the displacement, the asymptotic solutions of the first five natural frequencies can be found and compared. Since the tie constraints introduce additional equations to solve simultaneously, there will also be additional natural frequencies to be computed. This process will likely have a larger effect on the efficiency of PRoFile rather than the accuracy.

Appendix

The plots in Figure 10 illustrate the displacements computed by PRoFile for all configurations. These are compared with the asymptotic displacement - the displacement found when using no ties and a nominal mesh of 1,000-by-100 QUADs.



(a) Axial, Straight

(b) Axial, Angled

(c) Axial, Curved

(d) Cantilever, Straight

(e) Cantilever, Angled

(f) Cantilever, Curved

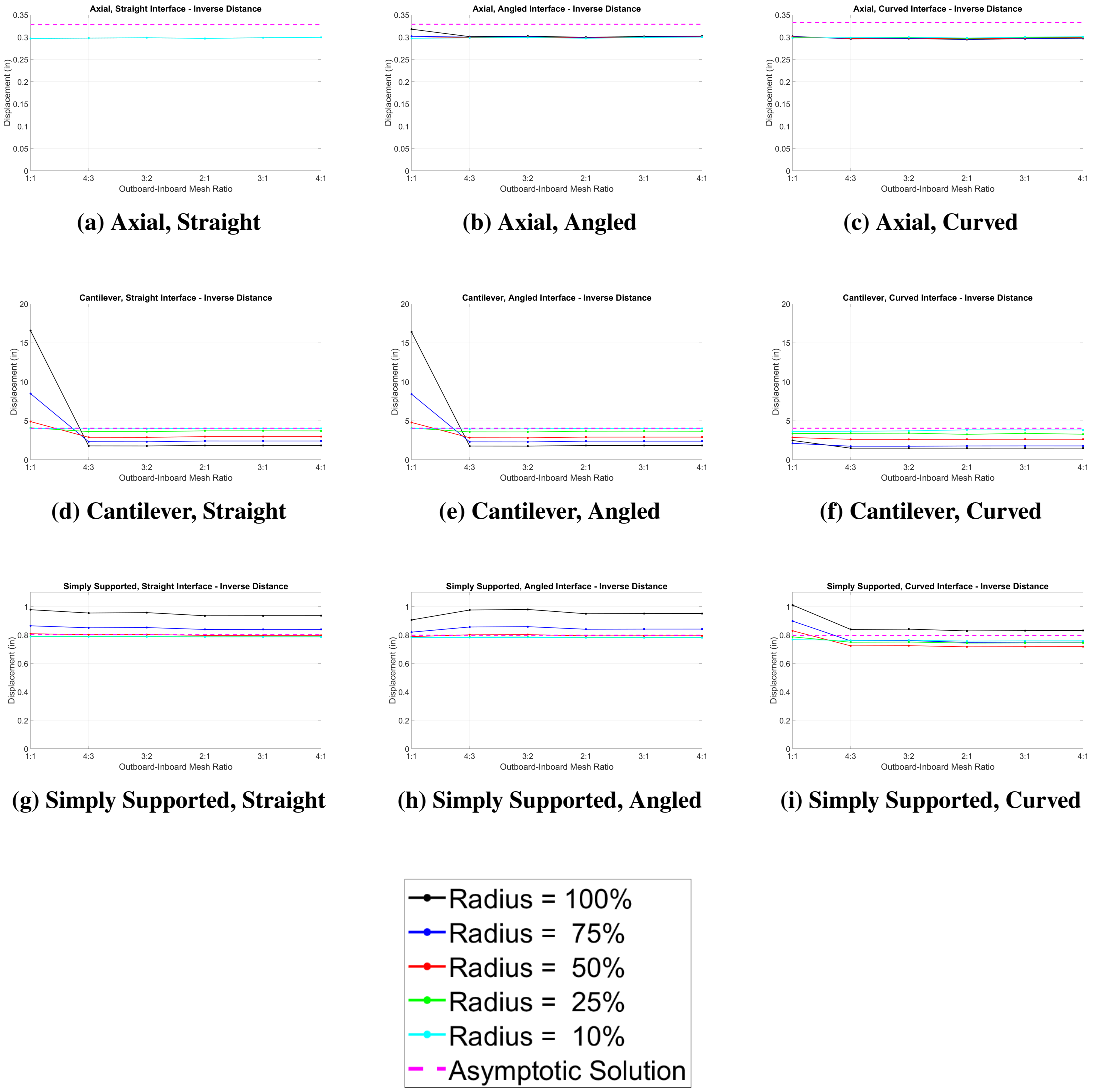
(g) Simply Supported, Straight

(h) Simply Supported, Angled

(i) Simply Supported, Curved

Fig. 10 Displacements for all configurations.

The plots in Figure 11 display the displacements computed by PRoFile when varying the radius of influence. Recall that the radius of influence was either 100%, 75%, 50%, 25%, or 10% of the overall edge length.



(a) Axial, Straight

(b) Axial, Angled

(c) Axial, Curved

(d) Cantilever, Straight

(e) Cantilever, Angled

(f) Cantilever, Curved

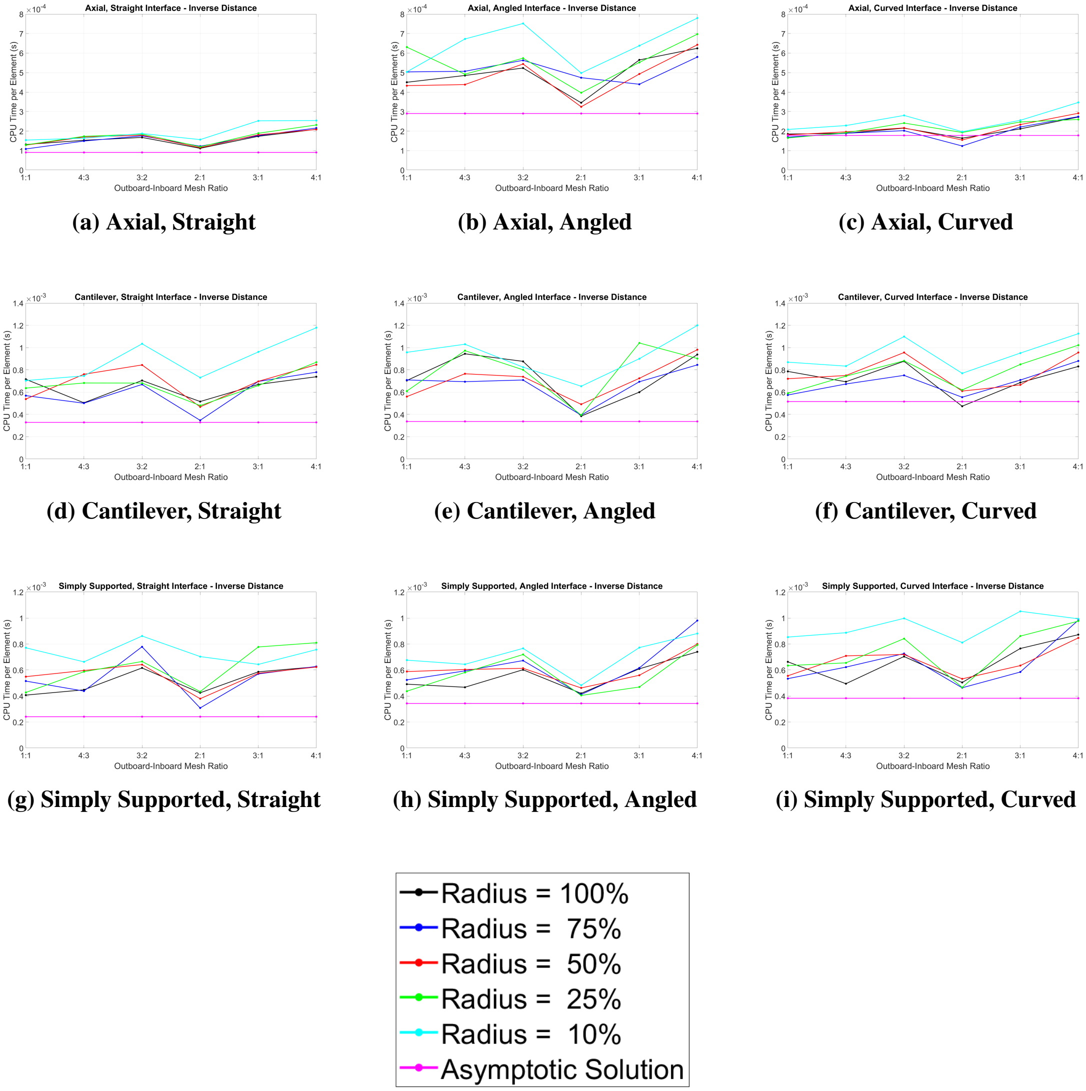
(g) Simply Supported, Straight

(h) Simply Supported, Angled

(i) Simply Supported, Curved

Fig. 11 Displacements when varying the radius of influence.

The plots in Figure 12 display the CPU time per element to obtain the FEA solution when the radius of influence was varied. Recall that the radius of influence was either 100%, 75%, 50%, 25%, or 10% of the overall edge length. Again, the asymptotic solution refers to the required CPU time per element to solve a nominal mesh of 100-by-20 QUADs without any tie constraints.



(a) Axial, Straight

(b) Axial, Angled

(c) Axial, Curved

(d) Cantilever, Straight

(e) Cantilever, Angled

(f) Cantilever, Curved

(g) Simply Supported, Straight

(h) Simply Supported, Angled

(i) Simply Supported, Curved

Fig. 12 CPU Time required when varying the radius of influence.

Acknowledgments

This work was partially funded by the CAPS Project, AFRL Contract FA8050-14-C-2472: "CAPS: Computational Aircraft Prototype Synthesis"; Dean Bryson was the technical monitor.

References

[1] Dannenhoffer, J., and Haimes, R., "The Engineering Sketch Pad," , 2020. URL `https://acdl.mit.edu/esp/`.

[2] Quiroz, L., and Beckers, P., "Non-conforming mesh gluing in the finite elements method," *International Journal for Numerical Methods in Engineering*, Vol. 38, 1995, pp. 2165–2184.

[3] Press, W., Flannery, B., Teukolsky, S., and Vettering, W., *Numerical Recipes in C - The Art of Scientific Computing*, Cambridge University Press, 1988.

[4] Cook, R., *Finite element modeling for stress analysis*, 2$^{\text{nd}}$ ed., John Wiley & Sons, 1995.

[5] Deaton, J., Private Communication, 2019.