“CleanUp: Improving Quadrilateral Finite Element Meshes”

Paul Kinney
MD-10 ECC
P.O. Box 2053
Ford Motor Company
Dearborn, MI. 48121
(313) 248-1228
pkinney@ford.com

Abstract: Unless an all quadrilateral (quad) finite element mesher is of a high quality, the mesh it produces can contain misshapen quads. This paper will describe “CleanUp”, written to improve an all quad mesh. CleanUp looks at improving node connectivity, boundary and flange patterns, quad shape, and to some extent, quad size. CleanUp is currently used in conjunction with the Paver algorithm developed by Sandia National Laboratories and is a part of their CUBIT software.

Keywords: all quadrilateral meshing, mesh improvement

Purpose:

The quality of a finite element mesh affects the results of analysis done using that mesh. For example, the more the angles of a quadrilateral deviate from 90 degrees, the more unreliable the stress calculations become. In contrast, an all-quadrilateral mesher may, under some circumstances, be doing a good job merely to put any mesh onto the surface without being concerned with the quality of the mesh. A separate process to review the mesh and improve it has proven to drastically reduce the number of misshapen quads and the extent of the shape distortion. This paper describes that process and how it was developed.

Definitions:

The valence of a node is the number of edges that meet at a node (or the number of quads that have a corner at the node). A 4 valent node has 4 edges into the node and 4 quads with a corner at the node.

The valence of a boundary node is based on the valence of the node on the surface being meshed and valence of the node on the adjacent surface with the assumption that (1) there is another surface there and (2) its mesh is “ideal”. These assumptions are base solely on the angle of the surface boundary at the node. Fig. 1 has examples.

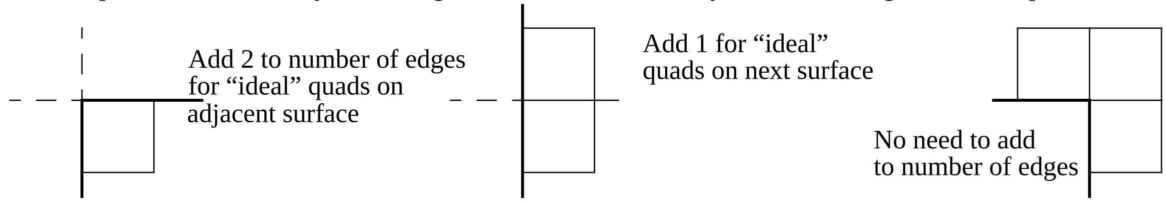


Fig. 1 valence at boundary node

An irregular node is one with a valence (regular or boundary) that is not 4.

A permanent node is one that cannot be moved. This refers to nodes on the surface boundary, which CleanUp assumes to be fixed as they were the starting point for the mesher. A permanent node may also refer to a hardpoint, which might represent a weldpoint or other unmovable feature, or a hardline, which is a “scratch” or a pattern of nodes and edges required to exist in the mesh.

A protected quad is one that cannot participate in the cleanup process. Examples of this are quads from neighboring faces and quads from a pattern mesh in difficult geometry (such as a surface spur shown if Fig. 2) in which the quads are poorly shaped but are the best that can be created. Shape cleaning would attempt to improve the middle two quads that have the large angles.

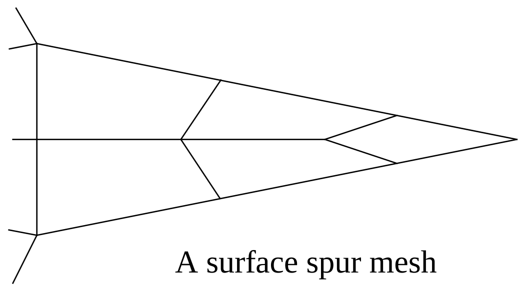


Fig. 2 A surface spur mesh

Goals:

How does one tell if a quad mesh is in good shape or if it needs some repairs? There are various quality metrics available which are described elsewhere [1]. The one used in evaluating a mesh for this project put importance on skew (defined as the difference between the angle at the vertex and 90 degrees) and aspect ratio (defined as the ratio of the longest side to the shortest side of a quad).

The quality metrics suggest the following guidelines which have been used in the development of CleanUp.

- All nodes, except those explicitly marked as permanent may be moved or deleted. New nodes are added as needed. The pattern of quads around a permanent node may be changed.
- All quads, except those explicitly marked as protected, are free to be adjusted or deleted. New quads are added as needed.
- Node valence of 2 and less or 6 and greater should be eliminated. A two valent node implies vertex angles of 180 degrees and a skew of 90 degrees. A 6 or higher valent node implies a vertex angle of 60 degrees or less and a skew of 30 degrees or more.
- The number of nodes with valence of 3 and 5 should be minimized. These valences are allowed as transitional meshes and some basic mesh patterns are impossible without them. They should be minimized as they imply angles of 120 degrees (and skew of 30) and of 72 degrees (and skew of 18).
- Quad angles greater than 160 degrees should be eliminated.
- Irregular nodes should be moved away from the surface boundary as much as possible. The number of edges on the surface that meet at a boundary node should be appropriate for the angle of the boundary at the node as defined by boundary valence.
- Each individual action should improve the mesh by improving quad shape or reducing the number of irregular nodes.

Method of development:

The method of development has, unfortunately, been entirely empirical. The cases found and implemented have come strictly from examining various test meshes to note areas that could be improved and studying possible methods of improvement.

The areas that need improvement are found through the use of a mesh analyzer to report on situations that could benefit from additional cleanup. This mesh analyzer reports valence patterns (described below) with two or more irregular nodes and quads with angles more than 160 degrees. This was implemented first to give an idea of which cases this mesher produces and in what frequency. The implementation was geared to what a particular mesher produces.

The action taken to improve a particular pattern is determined mostly by experimentation, though some guidelines can help the process. From an outline of the area surrounding the problem, can a standard pattern be used to fill it? Can a small change (rotate an edge, replace 2 quads with 3) handle the problem? Does a replacement pattern have fewer irregular nodes? Even with these guidelines, many possible solutions might be drawn on paper before an acceptable one is found.

Many times cases studied in isolation that improve the mesh in one area might cause degradation in an adjacent area. Because of that, restrictions are placed on a case as to when it can be used. For example, a case may have a restriction that a certain node must not be permanent as that would create an irregularity on the boundary.

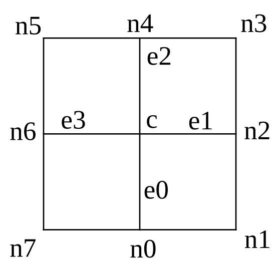
The cases that are important for one mesher may not be the same as for another mesher. The author has worked with two different meshers. One tended to produce poorly shaped faces, the other tended to produce clusters of irregular nodes. A complete general purpose cleanup algorithm may be considerably larger than what is necessary for any one particular mesher.

The author was guided by a comment from another mesh improvement developer. One of the “difficulties in implementation: Finding and coding an exhaustive set of all permutations of cases (1000’s)”[2]. A cleanup implementation does not have to be complete to be useful. Even if only a few of the cases are implemented, the mesh will be better than it was before. The implementer can concentrate on the cases determined to be most important, saving additional cases for later.

The author knows of no mathematical proof or formula which would reveal a finite set of cases and their proper resolution. The author has written a mathematical definition of the cleanup process and will make it available to anyone interested in pursuing a mathematical solution.

Connectivity cleanup:

Each node in the interior of the mesh is checked to improve its connectivity. The nodes, edges, and quads surrounding the node being checked are ordered in a counter-clockwise manner around the quad normal. The number of these neighbor edges and quads is the same as the node valence (by definition). The number of nodes is twice the valence as shown in Fig. 3.

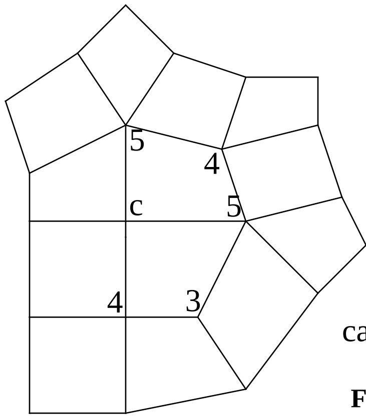


e0 - e3 are the neighboring edges to node c

n0 - n7 are the neighboring nodes to node c

Fig. 3 neighboring nodes and edges

The valence of the node and each of its neighbors is computed and compared against patterns that have a known cleanup action. Each connectivity cleanup case is documented by the pattern of valences that it checks. The valence of the central node is listed first, followed by a dash. Then comes the valence of a node at the other end of an edge from the center node. The remaining neighbor nodes are listed counter-clockwise around the element normal. A “4-” means a valence of 4 or less. A “4+” means a valence of 4 or more. A “5” means a valence of 5 or more. A “0” means the valence is ignored and unchanged and is usually drawn as valence 4. Case “4-43545000” is shown in Fig. 4.

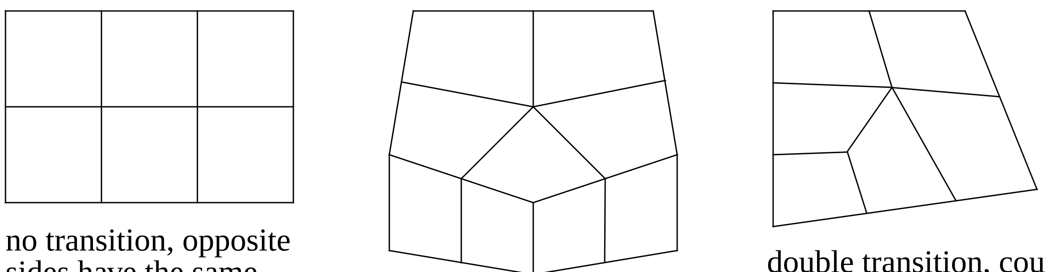


case 4-43545000

Fig. 4

Case 4-43545000 is a candidate for cleanup as the list includes more than two irregular nodes. Transitions may require both a 5-valent and a 3-valent node but patterns with more than two can usually be improved.

The first choice for a replacement pattern is one that results in one of the standard mesh patterns. These are shown in Fig 5. If none of those fit, the replacement pattern is one that reduces the number of irregular nodes. An action routine will change the old pattern into the new one.



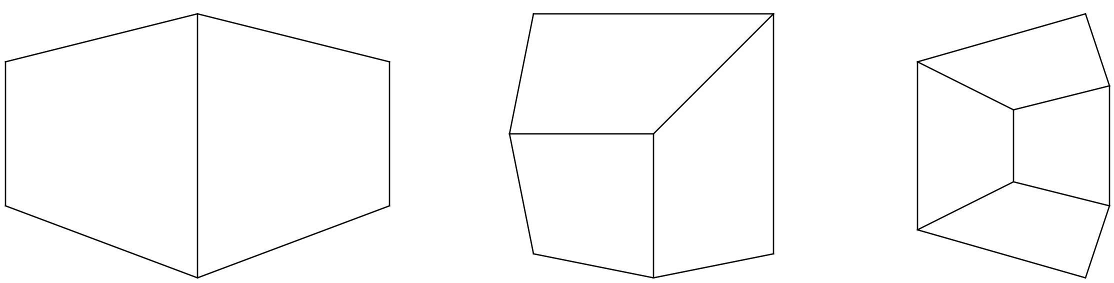
no transition, opposite sides have the same edge count

single transition, count on one pair of opposite sides varies by two

double transition, count on both pairs of opposite sides vary by one

Fig. 5 Standard mesh patterns

The most frequent actions are a "combine with neighbor" operation. A quad and a neighbor (one that shares a common edge) are deleted along with that common edge. This creates a hole surrounded by 6 nodes. The hole is filled with two, three, or four quads, shown in Fig. 6.



fill_2 fill_3 fill_4

Fig. 6 Filling a 6 node hole

The simplest of these is the “switch diagonal CW” and “-CCW”. Fig. 7 is an example of how switch diagonal CCW would be used. The edge marked “A” is the one switched. The quads on either side of edge A are deleted and fill_2 is done to fill the hole such that the two new faces have a different connectivity. The valence pattern in this example is “5-3443000000”. Three irregular nodes in the old pattern are replaced with one irregular node in the new pattern.

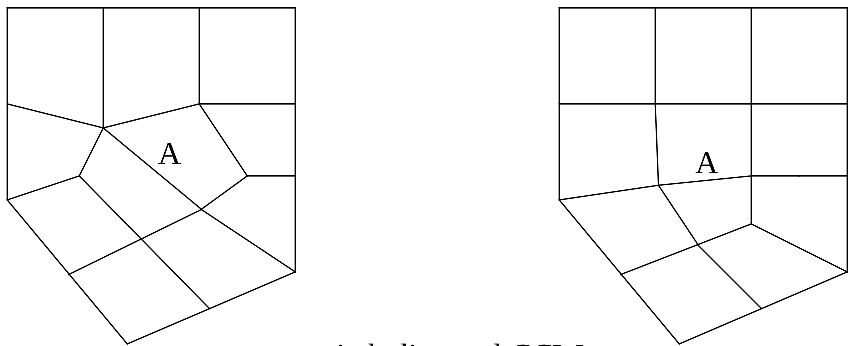


Fig. 7 switch diagonal CCW

Another example of a combine with neighbor is an open quad operation. In this case, the two quads - on either side of edge B - are deleted and a fill_3 is used to close the hole. This is shown in Fig. 8. Again, three irregular nodes are replaced with one irregular node.

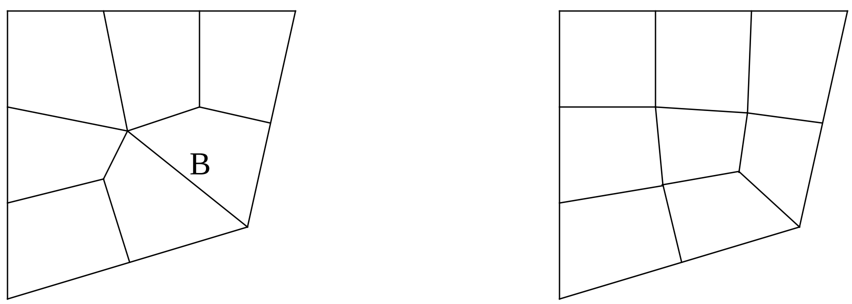


Fig. 8 open quad operation

Many cases also have a mirror image case. The mirror pattern to 4-43545000 is 4-40005453 (or, if starting from the top edge, 4-54534000) and is shown in Fig. 9. This means there are frequently pairs of action routines to handle a case and its mirror.

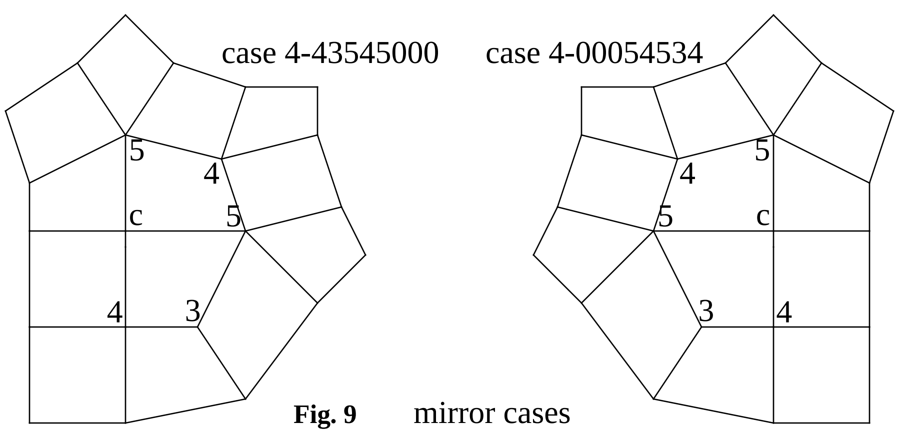


Fig. 9 mirror cases

If no match is found against the known cases, the lists are adjusted to correspond to starting with the next edge and the cases are checked again.

Many cases also check for permanent nodes in various positions because their corresponding action would disturb the permanent node, or would not be appropriate close to a surface boundary as it would create a quad with one node on the boundary, or would not have enough room for proper smoothing.

Once an action has been done for connectivity cleanup, the neighbor nodes now have a different valence, so they are put back into the list to be checked again.

There are currently 64 connectivity cases in the code. These invoke 27 different action routines. The action routines range from simple operations to rebuilding a small area of the mesh.

Additional examples of connectivity cleanup cases and their resolution are shown in Fig. 10 and 11. In Fig. 10, all three quads around the center node are deleted and a fill_2 is used to fill the hole. Four irregular nodes are replaced with zero irregular nodes.

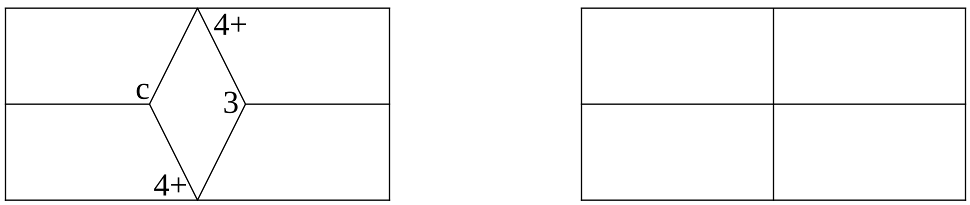


Fig. 10 Case 3-4+34+000

In Fig. 11, three quads around the center node and a fourth "outer" quad are deleted and the hole is closed with a fill_2. Four irregular nodes are replaced with two irregular nodes.

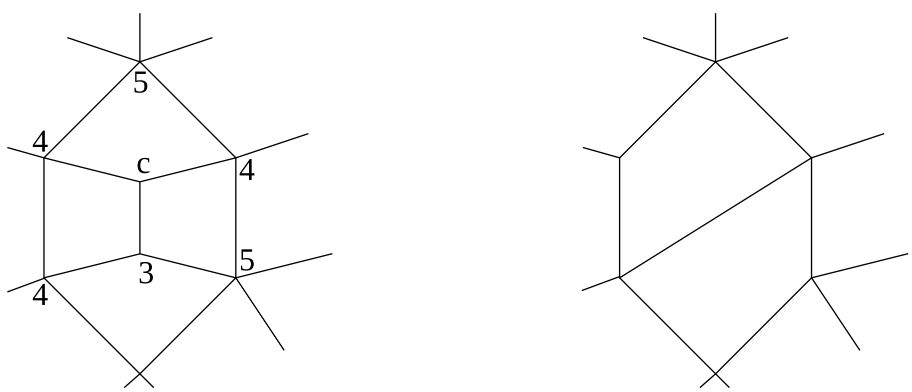


Fig. 11 Case 3-354544

While some cases change a small number of quads, other cases can change a large number of quads as shown in Fig.12. In this case, 10 quads, 15 edges, and 4 of 6 interior nodes are deleted. The replacement is 6 quads.

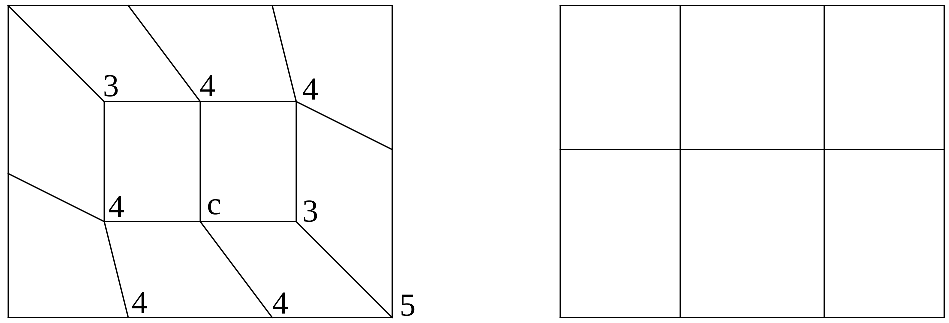
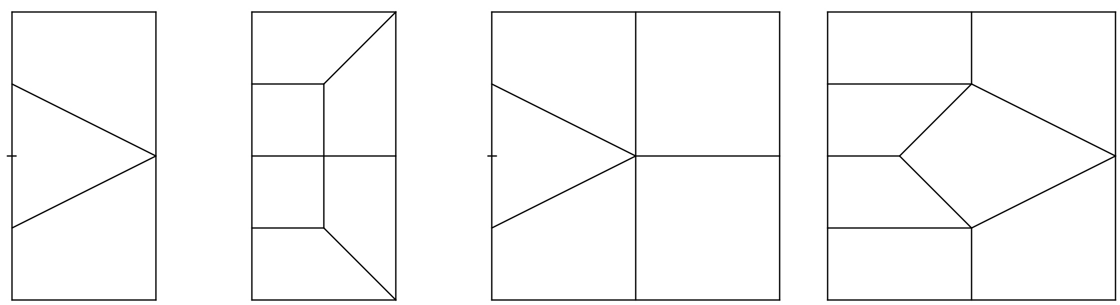


Fig. 12 Case 4-34434445 10 quads replaced with 6 quads

Boundary cleanup:

The boundary cases fall into three types. The first is a boundary node with only two edges, both on the surface boundary, at a place where the boundary forms an angle larger than 150 degrees. This forms a triangular (or nearly so) shaped quad against the boundary. Examples of replacing a triangular quad are shown in Fig 13.



one row transition two row transition

Fig. 13 replacing triangular shaped quads

The second type of boundary cleanup is similar to connectivity cleanup. The nodes, edges, and quads around the boundary are again put in order, but with a definite start and end to the list. Most of these cases are designed to clean up a flange and some of the actions may work their way along the flange. They are a part of boundary cleanup as there isn’t a pattern detectable from connectivity cleanup of interior nodes. They can be found by using a valence pattern similar to connectivity cleanup, though there are some differences. The valence pattern cannot rotate around the central node and many cases must also include careful checks for permanent nodes or the angles in a quad. There are over a dozen boundary cases. Fig. 14 shows some examples of boundary cases and their resolutions.

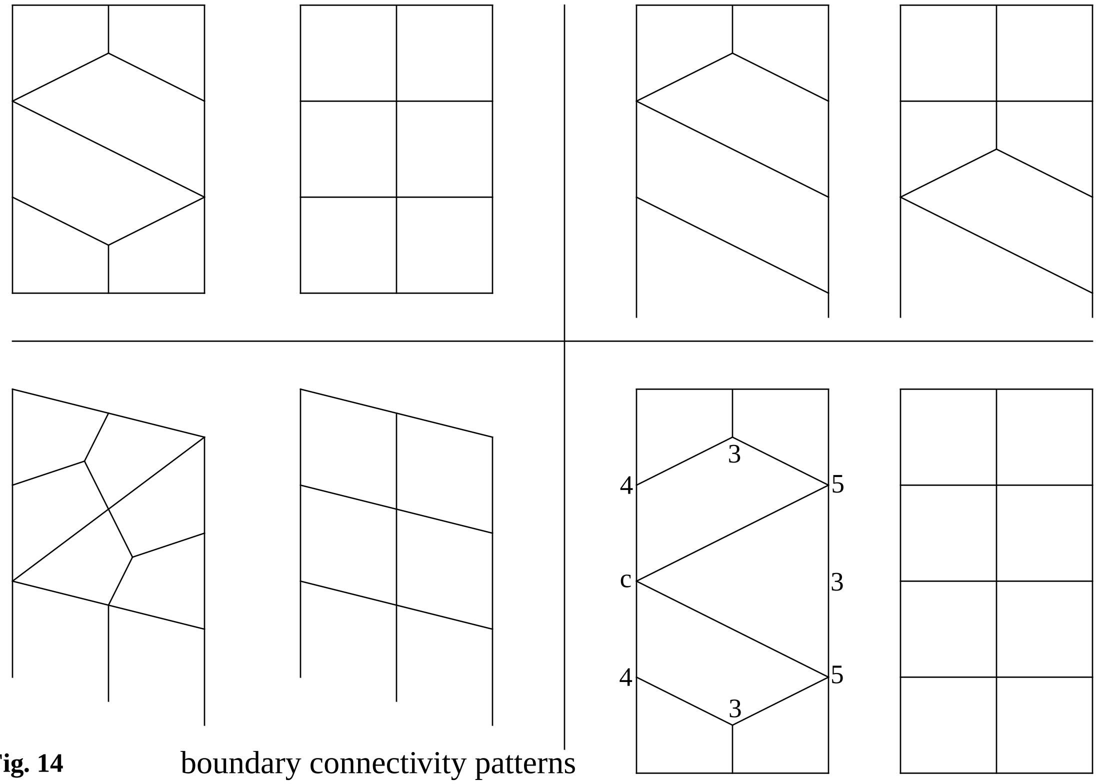
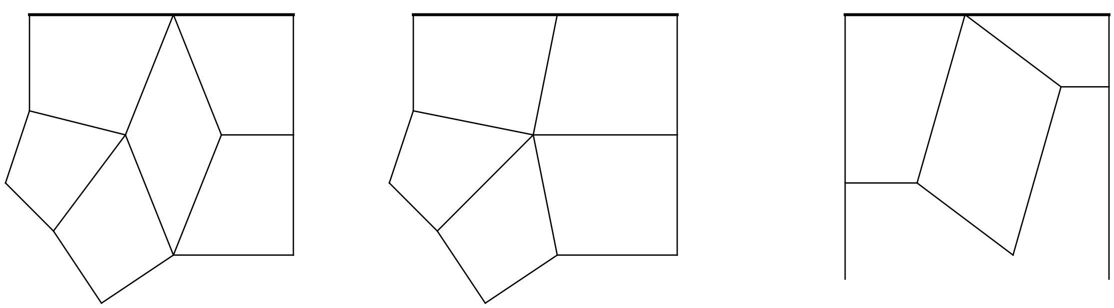


Fig. 14 boundary connectivity patterns

The third type of boundary cleanup is removal of boundary diamonds. These are quads that have only one node on the surface boundary. There are some cases where these quads can be collapsed. There are others where empirical evidence indicates they cannot be, at least not without more research. Some examples are shown in Fig. 15.



a diamond that can be collapsed

a diamond best left alone

Fig. 15 boundary diamonds

Shape cleanup:

There are two types of misshapen quads. The first type merely has a large angle - greater than 160 degrees. When these poorly shaped quads occur, they are frequently alongside a surface boundary or hardpoint. The usual method of resolution is to do a combine with neighbor. There are several cases to decide which neighbor is best and what pattern should be used to fill it. Fig. 16 contains an example. Fig. 17 shows a face cut in two and a combine with neighbor done on each half.

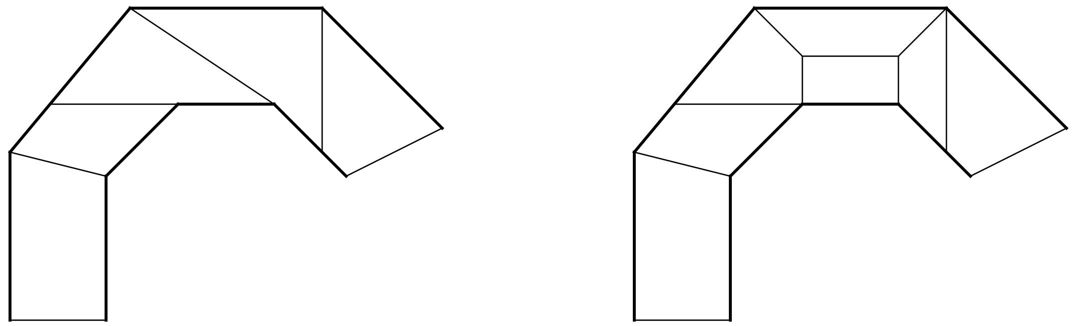


Fig. 16 combine with neighbor for large angles

Fig. 17 split a face in half and combine with neighbor on both sides

The second type of misshapen quad includes both the chevron (or arrowhead) and the bowtie. Both have a large angle (over 200 degrees). The difference is that two of the edges of the bowtie intersect, forming a twist in the quad. The chevron may be a poorly shaped quad, but the bowtie is “illegal” in that it violates the topology of the mesh. The chevron and bowtie are processed separately from simple large angles as both of these types of quads must be carefully analyzed to determine the bad angle as the quad normal may be inverted. Again, the bad quad is removed along with a neighbor and the hole is filled with better quads. These types of quads cannot be resolved by adjusting the node at the bad angle as it is frequently a hardpoint. Fig. 18 shows a chevron and its resolution. Fig. 19 shows a bowtie and its resolution.

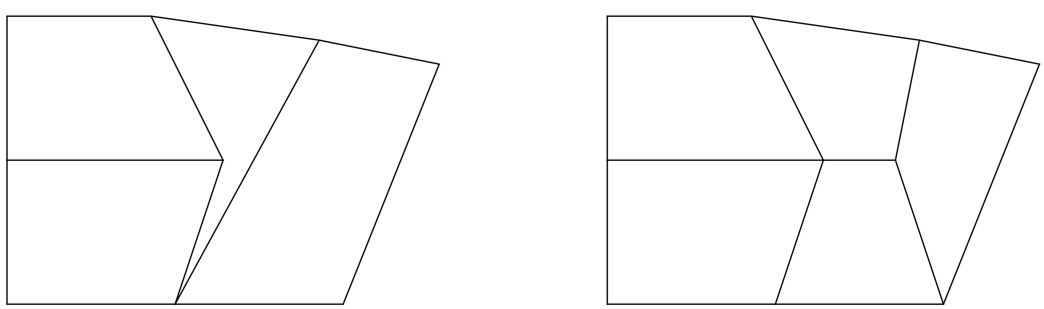


Fig. 18 cleaning up a chevron

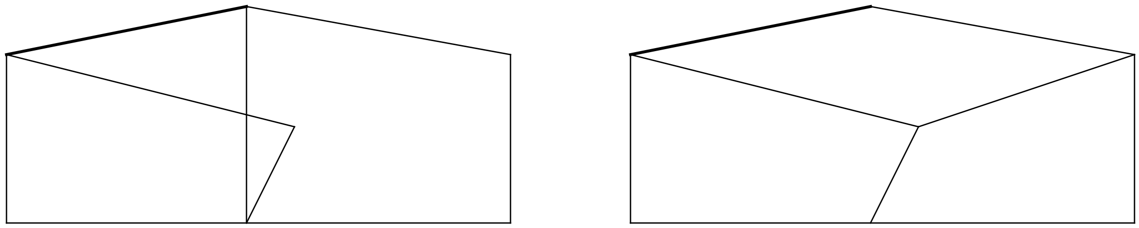


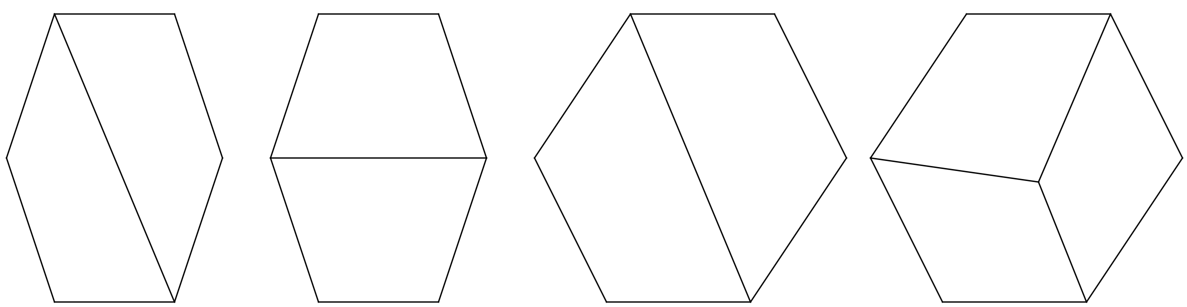
Fig. 19 cleaning up a bowtie

Since shape cleanup is dependent on angle measurements, smoothing is done as a part of each action, so that the angles represent an actual situation in the mesh.

A major dead end was encountered during development of shape cleanup. The first implementation used a generic combine with neighbor routine. It would choose what seemed to be a good neighbor, delete the two quads, and fill the hole based only on the shape of the hole. This did not take any cues from the surrounding area and frequently filled the hole with the same pattern of quads that had just been deleted. This was replaced by a more thoughtful look at the cases involved.

Size cleanup:

An edge that is considerably larger or smaller than it is supposed to be implies a misshapen quad or one that is out of step with an underlying sizing or adaptivity function. Size cleanup checks each edge on the surface (not the boundaries), comparing its length against the underlying size function. If the size is more than 2.5 times what it should be, the quads on either side are removed. If two opposite nodes can be joined with a shorter edge, that is done. Otherwise the hole is filled with three quads. Fig. 20 shows examples.

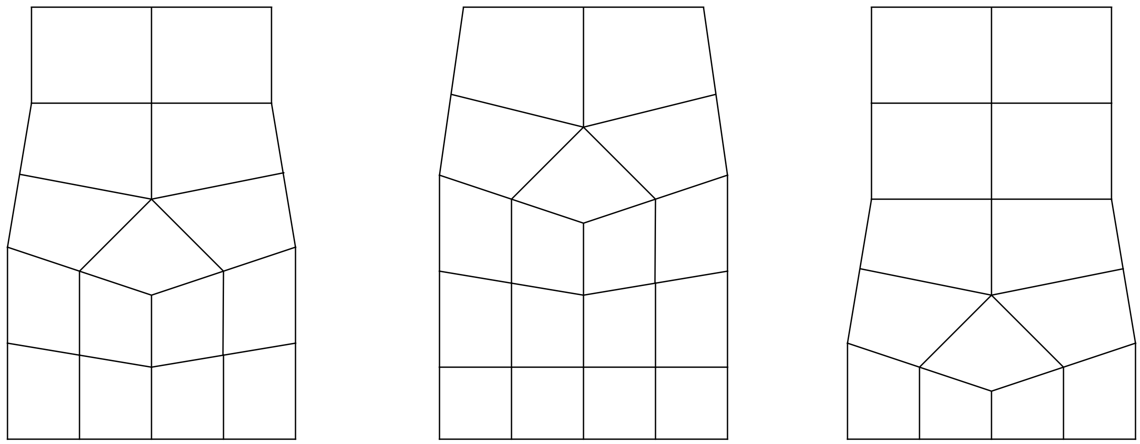


Long edge replaced with a shorter edge.

Long edge replaced with 3 quads to reduce the size of the quads.

Fig. 20 replacing a long edge

Some cleanup of quads that are too small can be done as part of connectivity cleanup. The standard transition patterns of 5-3 and 5-434 (zeroes assumed for the rest of these patterns) can be moved if the quad size is too small or too large. In the case of the 5-434 pattern, if the quads adjacent to the 5 valent node opposite of the 3 valent node are too large, the transition can be shifted into these quads to make them smaller. Likewise if the nodes adjacent to the 3 valent node and opposite of the 5 valent node are too small, the transition can be shifted toward these quads to make them larger. This transition can work its way across a surface. These cases are shown in Fig. 21.



a 5-434 transition   transition shift for smaller quads   transition shift for bigger quads

Fig. 21 adjusting a transition for size

Other than shifting transitions, there is currently no cleanup done if the edge is considerably smaller than it is supposed to be because obvious solutions (collapsing a quad, for instance) introduce problems with connectivity and shape.

General algorithm flow:

The cleanup process starts with a mesh topology inspection. Dangling nodes, edges, and quads are deleted.

Smoothing of the mesh is done at the start and after each major part of the algorithm. Local smoothing is also done after each shape cleanup action. The smoother that is used during cleanup is the same one used during the creation of the mesh so that the same goals are maintained. In this project, a Laplacian smoother is used. A more complete description of smoothing can be found elsewhere [3, 4, 5].

An initial pass is made through the quads to eliminate chevron and bowtie quads as connectivity cleanup around a bowtie usually makes the situation worse. Again, this is from empirical evidence.

There is a loop over the major cleanup processes - connectivity, boundary, shape, size. This loop is executed up to three times. More than three passes probably won’t improve the mesh and there is a chance of an infinite loop.

Since the components of cleanup, primarily connectivity and shape, have different goals and different criteria for what should be done, an action by one case may degrade the mesh according to the rules of another case. A connectivity fix may introduce a poorly shaped quad. A shape fix may introduce a bad connectivity situation. It is possible for two actions to cycle between two patterns of quads. This situation occurs because the cases and actions are derived empirically and then may be applied in a situation in which it isn’t appropriate. Only careful examination of a large number of test examples can resolve such situation, and the size of data from those test examples limits how many cases can be carefully tracked.

Results:

The CleanUp process can make a considerable difference in a mesh. Fig. 22 shows a mesh for a single surface without any CleanUp performed. The entire part was meshed in 52 seconds. The skew rating used here is based on the deviation of the angle at each vertex from 90 degrees. The mesh shown here has an average deviation of 27 degrees.

This mesh has 4 quads that have 3 colinear nodes, a 6 valent node, and 42 irregular nodes.

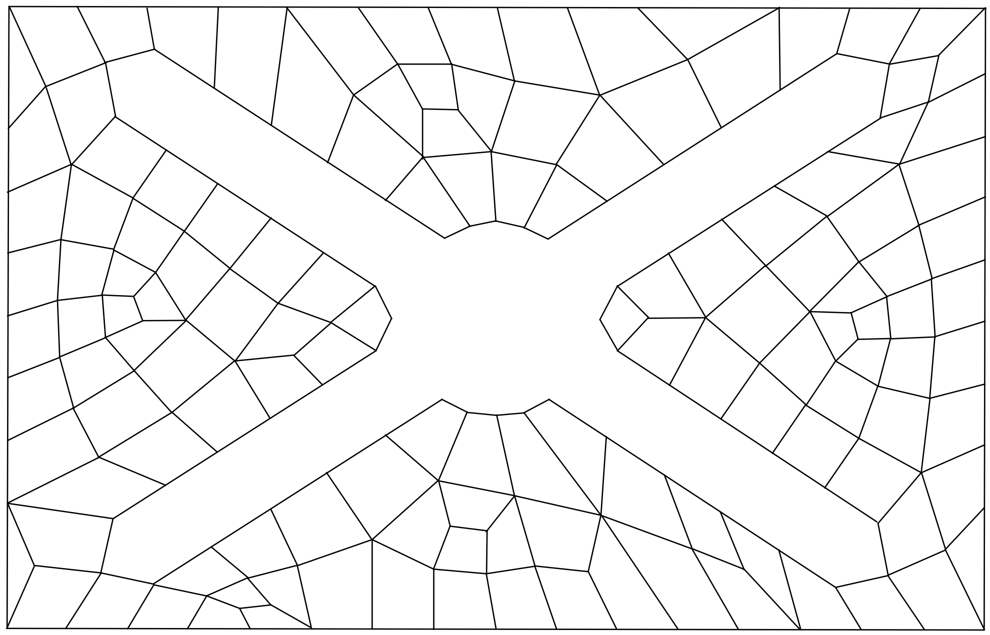


Fig.22 a mesh without CleanUp

Fig. 23 shows the same surfaces with CleanUp performed. The entire part was meshed and cleaned in 56 seconds. This mesh has an average deviation of 23 degrees. There are no quads with 3 colinear nodes, no 6 valent nodes, and only 28 irregular nodes.

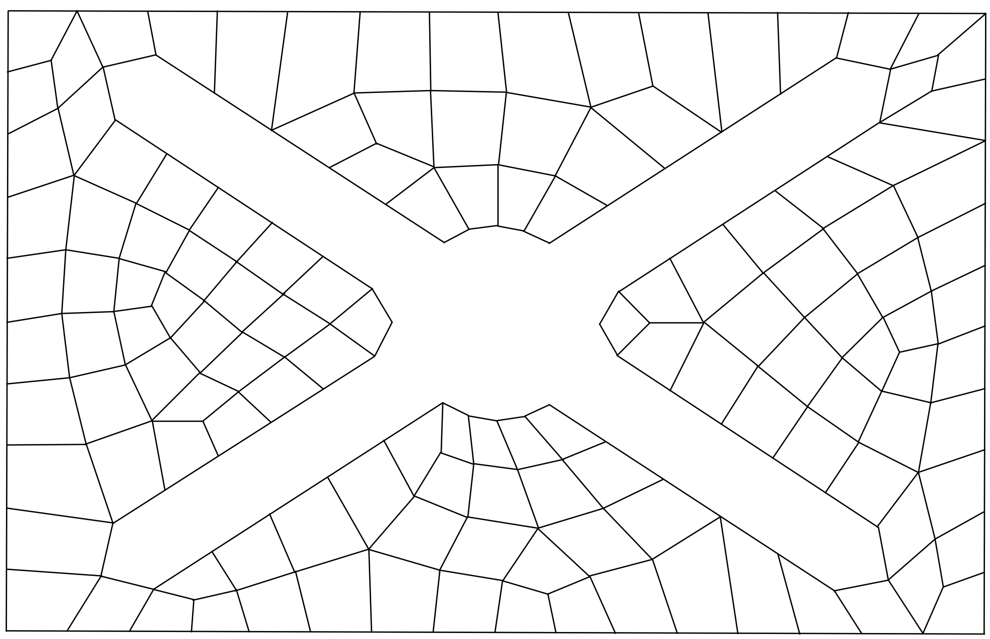


Fig. 23 a mesh with CleanUp

Conclusions

The mesh cleanup procedure described here can be adapted to work on the results of any quad mesher. The actual valence, boundary, shape, or size cases that are implemented depend on what the mesher produces.

This particular implementation of CleanUp cannot be applied to a triangular mesh or to a mesh that is a mix of trias and quads. Methods of improving triangular meshes have been described elsewhere [6, 7].

The author has had limited experience with cleaning a mesh with a mix of quads and trias. The mesher produced a small percentage of trias and a large percentage of quads. This mesh was cleaned by examining the mesh for cases where trias and quads could be combined for more quads and fewer trias or rearranged for better shaped elements. Then the trias were treated as holes in the mesh with virtual surface boundaries around them and CleanUp was applied to just the quads.

The mixed element cleaner contained only a few cases as it (1) dealt only with cases involving only trias or a mix of trias and quads and (2) the percentage of trias in the mesh was small. Even so, the cases were determined empirically. If the percentage of trias was higher and the trias could not effectively be isolated from the rest of the mesh, the author feels the complexity and number of the cases for a mixed element mesh may prove daunting.

Areas of future direction:

Additional work in mesh cleanup would likely be done in these areas: (1) Additional cases for each of connectivity, boundary, shape, and size would be implemented. The analysis reports now generated are by no means clean. (2) Explore ways to do cleanup on elements that are too small. (3) Consider ways to do cleanup on quads that have angles that are too small, instead of too big.

Acknowledgments:

This work was done as part of the Sandia National Laboratories CUBIT project and its Mesh Generation Consortium. This work builds on work done by Blacker and Stephenson [3], Scott Canann, and Roger Cass. The author wishes to thank the members of the CUBIT team for the initial work and for ongoing support.

References:

[1] John Robinson, “CRE Method of Element Testing and the “Jacobian Shape Parameters”, Engineering Computers, 1987, Vol. 4.

[2] Scott Canann, Sella Muthukrishnan, Bob Phillips, “Topological Improvement Procedures for Quadrilateral and Triangular Finite Element Meshes”, Proceedings, 3rd International Meshing Roundtable pp. 559-588. (1994)

[3] T. D. Blackler and M.B. Stephenson, “Paving: a New Approach to Automated Quadrilateral Mesh Generation”, International Journal for Numerical Methods in Engineering, Vol. 32, pp 811-847 (1991).

[4] L.R. Herrimann, “Laplacian-Isoparametric Grid Generator Scheme”, J. Eng. Mech. Div. ASCE, 102, EM5 749-756 (1976).

[5] R. E. Jones, “QMESH: A Self-Organizing Mesh Generation Program”, SLA-73-1088, Sandia National Laboratories, 1979.

[6] W. H. Frey and D.A. Field, “Mesh Relaxation: a New Technique for Improving Triangulations”, International Journal for Numerical Methods in Engineering, Vol. 31, pp. 1121-1133 (1991).

[7] D.A. Field and W. H. Frey, “Structural Improvement of Planar Triangulations: Some Constraints and Practical Issues”, Communications in Numerical Methods in Engineering, Vol. 11, pp. 191-198 (1995).