

# GLOBAL REMESHING OF ORPHAN SURFACE MESH WITH LOCAL CONTROL

Nilanjan Mukherjee<sup>1</sup>, Wafa Daldoul<sup>2</sup>, Jonathan E. Makem<sup>2</sup>

*Meshing & Abstraction, Simulation and Test Solutions, Siemens Digital Industry Software, SIEMENS.*

<sup>1</sup> 2000 Eastman Dr., Milford, Ohio 45150 USA, mukherjee.nilanjan@siemens.com

<sup>2</sup> Francis House, 112 Hills Road, Cambridge, UK. CB2 1PH jonathan.makem@siemens.com, wafa.daldoul@siemens.com

## ABSTRACT

This paper describes a series of strategies and algorithms used to generate a new two-dimensional mesh on an existing orphan element model. The remeshing strategy is founded on the concept of a Mesh Partitioner (MP) which creates a tessellated surface representation from the initial Orphan Element Patch (OEP). The resultant mesher-native surface patch called Face-on-Mesh (FOM) is first torn away from the orphan element model and remeshed with a variety of two-dimensional meshers with its boundary constrained with nodes from the parent orphan model. The new mesh is finally stitched back to the original orphan mesh. Sophisticated local mesh controls are exercised by means of an automated, controlled loop-paving algorithm which works around circular cutouts and patch interior points representing spot-welds. Results on various automotive parts describe how this tool is employed to locally refine meshes whose owning geometry is lost.

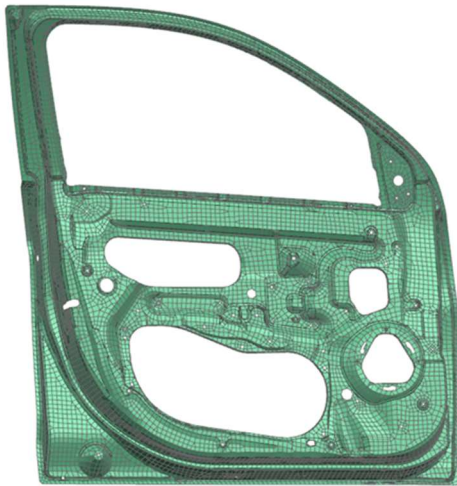
**Keywords:** remeshing, orphan element patch, mesh partitioner, spot-weld, mapped-hole

## 1. PROBLEM DEFINITION

A large number of industrial engineering analysis problems have to deal with legacy analysis data originating from sources that are lost or obsoleted. In the finite element analysis world one such common legacy data are finite element meshes, complete or partial that are “orphaned” from geometry. This means the meshes are barely a collection of connected nodes and elements that have lost the parent geometry on which they were originally created. These meshes have been already solved and postprocessed for some analyses. A very common example of such orphaned surface mesh from the automotive industry is the assembly mesh of a car door. Quite often designers reuse car doors from older models with minor feature modifications and/or newer mesh prescriptions. One such car door is shown in Figure 1a. Due to expected changes in load and boundary conditions or for the need of a different set of analyses a remeshing of these meshes becomes necessary. One living example of such analyses involves frequency response analysis of the door to study the behavior of new electronic gizmo which need to be mounted inside the door

frame. Electronic equipment normally resonate at much higher frequencies than structures. Such analyses requires a much finer mesh and even finer refinement around screw holes. Based on these new analyses results engineers decide if the existing part can be reused with minor design modifications or not. Thus in such cases remeshing the existing geometry-dissociated orphan mesh is of paramount importance. Since the 1970s until very recently this remeshing has been mostly done manually or by using semi-automatic methods. It has a negative impact on productivity and also quality. The desirable remesh on the existing orphan mesh with the desired characteristics is illustrated in Figure 1b. To be able to quicken the rather cumbersome legacy process leads to a notable increase in productivity and a compression of design cycles.

Most of the reasons why feature-sensitive, constraint-aware remeshing of Orphan Element Patches are deemed so important are listed below -



(a)



(b)

**Figure 1. A typical car door legacy finite element mesh (a) (11976 elements, 10651 quads, 1325 trias, mesh size = 12) and: the remeshed mesh (b) on the same car door (22869 elements, 22846 quads, 23 trias, remesh size = 8)**

- A) Remeshing small local segments or zones of an original mesh at a size different from the original size of the mesh.
- B) Remeshing tessellated geometry with no topological information (which is an orphan triangular mesh) with a different mesher
- C) Remeshing sections of existing meshes to impart local mesh control or characteristics absent from the original mesh – for example, characteristics like layered mesh around holes or cut-outs, altering the number of layers of an already layered mesh, densifying or rarefying the mesh in local areas of interest etc.
- D) Remeshing at a different mesh size to recover (refeathering) or ignore features (defeathering).

E) Remeshing local element patches or zones to conform with new loads and boundary conditions.

F) Remeshing local element patches in the vicinity of altered geometry to create localized smart mesh updates.

All the above requirements are traditionally achieved by deleting the original mesh and remeshing on the original geometry with new mesh definitions. However, here the mesh has been orphaned. There is no geometry to remesh on. Thus, remeshing on an orphan patch serves the dual purpose of recreating through guestimation a new virtual geometry as close as possible to what might have been the original; next it generates a new mesh on it that honors the revised or completely new mesh definition. All this happens in one single step after local meshing requirements are specified by defining some local mesh controls or definitions which makes it an immensely powerful functionality.

## 2. PREVIOUS WORK

It may be necessary to remesh small local segments or zones of an orphan mesh at a size different from the original size of the mesh. Refining the mesh by increasing the density of mesh elements can be performed to better adapt the mesh to the physics being modelled, to conform with new loads and boundary conditions or to recover some features. On the other hand, coarsening can also be performed to ignore certain features. One of the earliest reported investigation and application geometry creation on geometry disassociated legacy meshes was reported [1] by our previous company unit SDRC (erstwhile). A significant amount of research has been done to fulfil the demand of locally refining all-quad meshes and numerous template-based algorithms have been presented in [2][3]. More recent work has been done as well in coarsening quad meshes [4][5].

More generally, being able to remesh an existing orphan mesh and modify it locally by changing the element size or adding new characteristics absent from the original mesh can be achieved by several means.

A commercially available mesh remesher is provided by the STAR\_CCM+ software package [6]. The main advantage of this algorithm is its ability to remesh tessellation in 3d without a flattened representation. Conversely, it cannot generate structured quad meshes on orphan meshes nor can the user control the mesh around inner holes for example.

CADfix [7] implemented a “Back-to-CAD” solution called “FROG” (Faceted Representation OF Geometry”) which takes a triangulated mesh generated from an optical scan and creates a polygon geometry on the mesh using a gaussian cube partitioning technique. This polygon geometry can then be quadrilateral or triangular element meshed at a different size, which is a multi-stepped process involving user interaction.

The open source computational geometry library CGAL [8] has a “Triangulated Surface Mesh Segmentation” tool that automatically decomposes a mesh into polygons. The

algorithm relies on the shape diameter function [4] (SDF). For an input triangulated surface mesh, the SDF provides an estimate of the local object diameter for each triangle of the mesh. The segmentation algorithm first applies a soft clustering on the triangles using the SDF values. An initial segmentation is then obtained from a graph cut algorithm that uses surface-based features as well as the soft clustering result.

ABAQUS software [9] provides a solution to add geometric features to an imported tessellation. An orphan element face can be selected as the sketch plane for the creation of new geometry. The user can then modify the mesh using the functionalities available for use with geometry associated meshes. The process is not fully automated as the user needs to manually create the geometry before being able significantly to modify the mesh.

However, these previous investigations are limited by the fact that they are not able to generate new meshes on old meshes simultaneously honoring both features and a myriad of mesh constraints. In some academic and commercial codes, it may be possible to remesh on mesh with reasonable feature-sensitivity but in multiple steps. Furthermore, constraint-aware mesh generation on orphan meshes is unknown by far. This invention focuses on these limitations and proposes a robust and innovative solution to the problem.

### 3. PROPOSED REMESHING STRATEGY

The strategy adopted to remesh a patch of geometry-dissociated, orphan finite elements can be described by a set of key algorithms. The overall, broad algorithm, however, is important to understand at the onset. Figure 2 shows the overall algorithm describing salient steps, sequence of operations and input and output data at each step.

We will define the key entities or objects involved in the process first.

**Orphan Element Patch (OEP)** is defined as a copy of the input patch of finite elements. It is an object that is created to hold and represent the input patch. The OEP object holds two types of data - a) primary data (shown in Figure 3a) and b) secondary data (mesh definitions in Figure 3b). Primary data consists of the nodes and elements defining the patch, their labels and coordinates (for nodes) and labels and nodal connectivity (for elements). The secondary data consists of control parameters or “mesh definitions” that will define the remesh. These are element type, size or factor, mesh size transition factor, mesh feature edges, multiblocking or other mesh definition options, mapped hole definition, spot weld definition etc.

**Mesh Partitioner (MP)** is a tool that reads in an input triangulation and partitions it into faceted or tessellated face representations. The input triangles must be linear and connected to form a single continuous blob which is not self-intersecting. Blob partitioning is based on a feature angle and some other optional definitions. The MP partitions the triangulation into groups of triangles, building necessary topology to construct faceted face definitions.

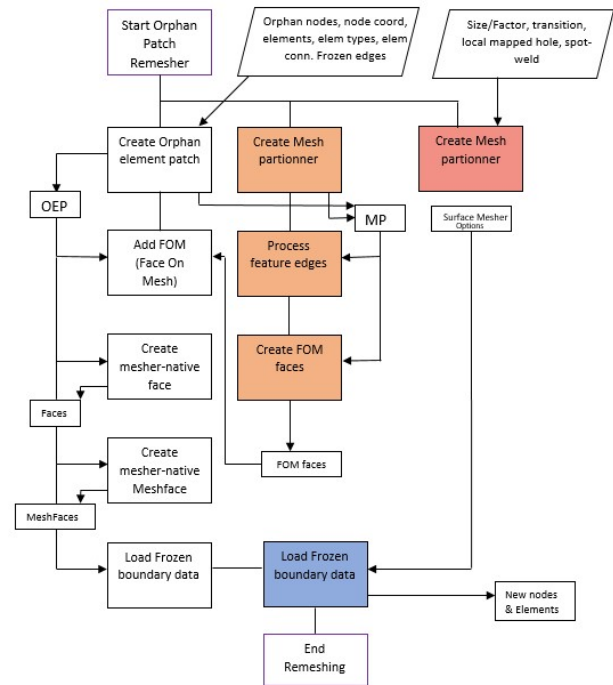


Figure 2. Overall algorithm explaining Orphan Element Patch remeshing

**Face On Mesh (FOM)** is defined as a discrete data surface generated out of an input triangulation. It is nothing but a faceted or tessellated face. It is called FOM face simply because its facets are elements of a finite element mesh and not a discrete surface generated from CAD data.

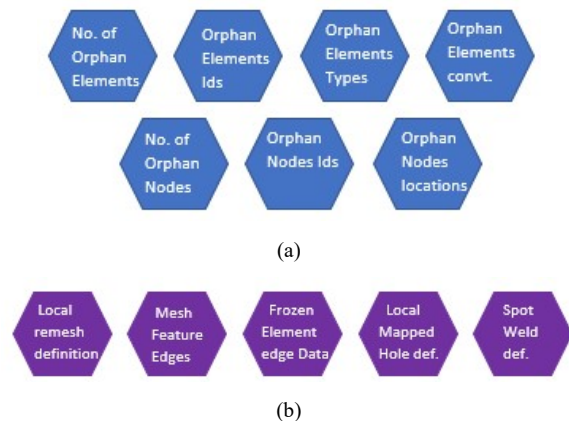


Figure 3. Data content of Orphan Element Patch Object - (a) primary data (b) secondary data

**Mesh Native Face (Face)** is a surface defined inside the mesher database. It is a *simulacrum* (borrowing a cultura-

scientific term) which is an exact copy of the original face created for the purpose of mesh generation. The “Face” is defined by facets and edge-loops. Each edge-loop is made up of one or more mesher-native edges while each edge is made up of two mesher-native vertices.

**Mesh Native Mesh-Face** (MeshFace) is also a mesher-native definition residing in its database. It is not a simulacrum, however. The MeshFace points to the Face object and holds, accesses and computes mesh definitions and its derivatives.

**Surface Mesher Option** is an object that reads all the surface mesh definition data (as shown in Figure 3b) and caches it. During mesh generation, this object is interrogated repeatedly to access this data.

#### 4. REMESHING ALGORITHM

The overall meshing algorithm illustrated in Figure 2 performs certain specific tasks in a well-defined sequence. These tasks are listed below -

**I** A copy of the input elements and nodes are used to create the Orphan Element Patch (OEP) object with a user entered section id. A typical OEP object is shown in Figure 4. mesh generation, this object is interrogated repeatedly to access this data.

**II** A Mesh Partitioner (MP) object is created at the same time with the same input element patch data. If the input element patch is made up of quadrilateral elements, those are split into triangles. Each quad is split into two triangles. The OEP id is set on the MP object. The Mesh Partitioner is thus related to the OEP via the OEP Id.

**III** The MP next creates transient geometry faces inside the mesher called Face-On-Mesh (FOM) faces from the input orphan mesh using the following secondary input data –

A) boundary frozen element edges (optional) based on which geometry edges are created that start and end with boundary frozen element edges. In the process, it ensures no geometry edges are created that have partial frozen boundary data. These transient geometry edges are either completely frozen or constrained by input boundary element edges or completely free.

B) Input feature edges (optional) which are element edges of the input patch and these are treated as “clue” element edges and are used to mark down the contour or emerging roots of a feature the MP must consider keeping. Figure 5 shows some element edges (in orange) selected by a user that are passed to the MP as feature boundary clues.

**IV** Once the FOM faces are created by the MP, it invokes the OEP object and passes the data. The FOM face data consists of facets or tessellation which are in essence the triangles of the input mesh.

**V** The OEP object now uses the FOM face data to create mesher-native transient geometry objects called *Faces* and *MeshFaces*. The *MeshFace* is different from the *Face* in that

it either holds mesh definition data or has the ability to access, compute detailed mesh definition data sent in by the user and its derivatives. Figure 6 shows the FOM faces (same as mesher-native Faces) created by the MP for the example in Figure 5. It is clear how the feature element edge clue is used by the MP to generate bead feature faces and how the annular face feature is also retained. (Figure 6). 10 FOM faces are created by the MP in this example.

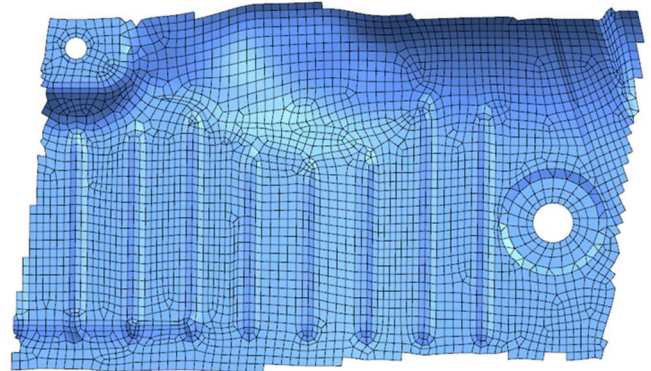


Figure 4. An Orphan Element Patch made from an input quad-dominant element patch (original mesh size = 7.5)

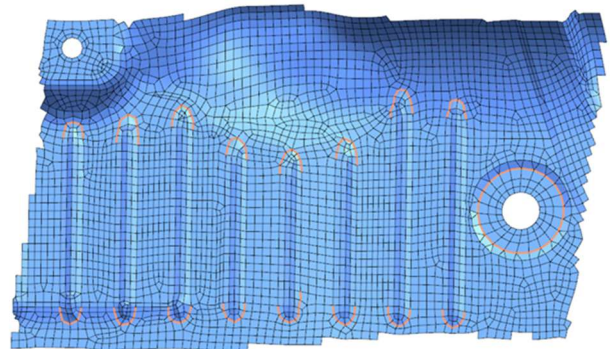
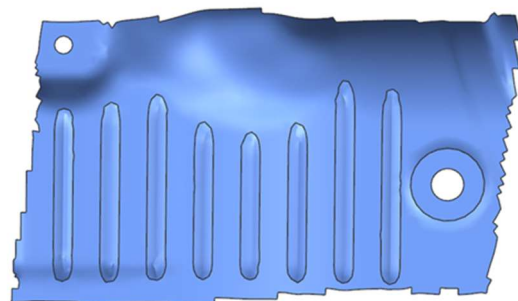
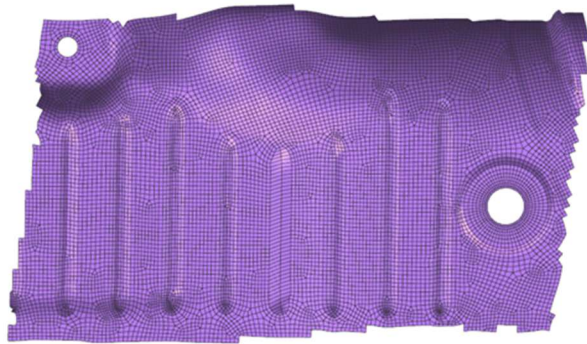


Figure 5. Orphan Element Patch with user selected feature edges

**VI** In this step, boundary frozen element edge data is loaded on the transient geometry Edges (mesher-native). This is described by Figure 7 which shows in red the free element edges of another OEP which are considered frozen



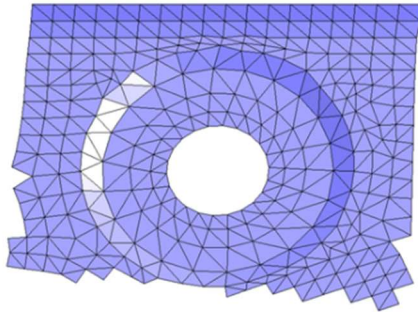
(a)



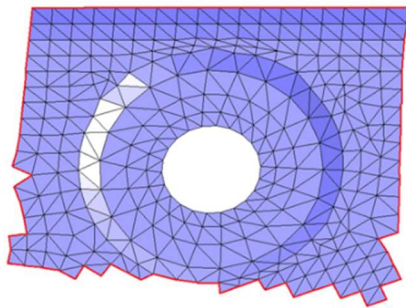
(b)

Figure 6. FOM faces generated by the MP from the OEP (a) and final remesh (b) at a mesh size = 4

VII The surface mesher is then invoked on these Faces. During the meshing process the Surface Mesher Options object is accessed and queried multiple times to get to the special user-driven mesh definitions, e.g. things like spot weld definition, mapped hole definition, multiblock or other meshing options, element size of size-factor, type, mesh transition factor etc.



(a)



(b)

Figure 7. An Orphan Element Patch (a) and its frozen boundary edges (b).

VIII The generated mesh on these mesher native Faces is the final output. This mesh (collection of all meshes on the multiple Faces) represents the “remesh” and is sent back to the user application for storage and display. Figure 8 illustrates the final mesh for the example shown in Figure 7. This mesh is generated at a smaller size and honors element

size transition option and local mapped hole definitions. It also shows bigger element sizes near its free boundary reflecting the original element size of the OEP because those edges are frozen (Figure 7b).

IX The last step involves stitching the new remeshed mesh back into the original mesh the OEP was torn out from.

In the process, with one button click the user automatically gets a new surface mesh generated on a patch of orphan elements. The new mesh is very different from the original

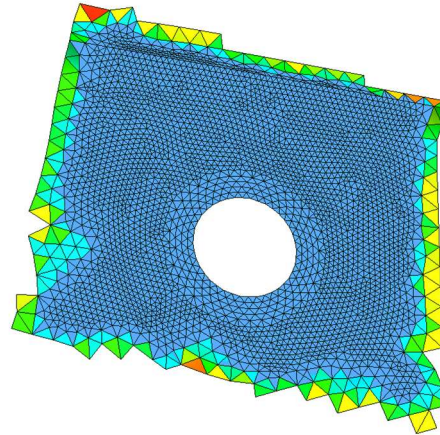


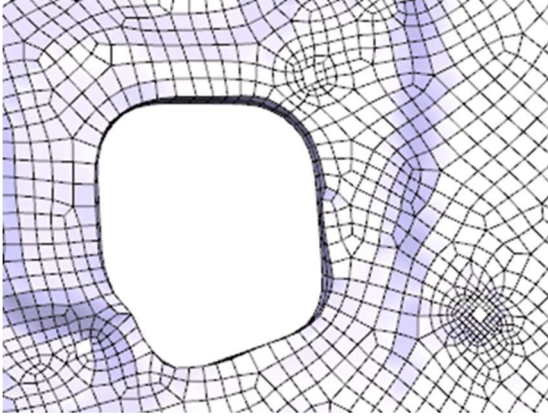
Figure 8. Final triangular mesh on the OEP from Figure 7.

selected patch in terms of the mesh definitions supplied by the user – element type, size, special mesh parameter control etc. Let’s remember the car door shown in Figure 1a and the new remeshed mesh on the same as shown in Figure 1b.

Further comparison of three details in the original OEP mesh with the new remesh reveals the importance of local remesh and the gains one makes in terms of mesh sizing, change of mesh character and local mesh control. These details are illustrated in Figures 9.



(a)



(b)

Figure 9. A third section detail comparison. Original patch mesh (a) versus remeshed mesh (b).

It is conspicuous from the comparisons how when contrasted with the original mesh, the remesh retains the key or more pronounced feature lines while ignoring the minors, has lesser number of triangles, more quadrilaterals (as the element size is reduced) and mapped hole characteristics (dense layered paved meshes) around holes and convex cut-outs.

## 5. MESH PARTITIONING

Mesh partitioning or “segmentation” as it is also called, is a complex algorithm on its own. This is one of the most significant steps in the targeted task. We make an effort here to summarize the algorithmic part of [10].

The MP (*Mesh Partitioner*) uses its own definitions of elements and geometry. Before we proceed, it is important to describe the nomenclature used in the mesh partitioner as listed in Table I. The algorithm is based a previous work by Jiao [11,13,14] and extends it. It describes face building from a finite element mesh by focusing on the details of topological and geometrical measures. These when combined provide a robust and flexible framework to capture salient features that exist in a finite element mesh.

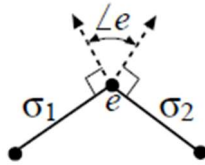


Figure 10. Dihedral Angle.

Table I

<b>FB Edge</b>	Free Boundary edge (adjacent to only one element)
<b>NM Edge</b>	Non-Manifold edge (adjacent to more than two elements)

<b>U-Strong edge</b>	Unconditionally strong edge.
<b>L-Strong edge</b>	Locally strong edge.
<b>S-Strong edge</b>	Shadow strong edge.
<b>FEdge Feature Edge</b>	Element edge used as a separator during Virtual Face creation
<b>FNNode Feature Node</b>	Node that belongs to a feature edge.
<b>TS Node Terminal Node</b>	FNNode adjacent to only one FEdge.
<b>VEdge Virtual Edge</b>	An oriented list of FEdges. Can be closed or opened.
<b>VLoop Virtual Loop</b>	A closed oriented list of VEdge(s).
<b>VFace Virtual Face</b>	Set of elements that are bounded by FEdge(s)
<b>S-VEdge Scar VEdge</b>	A VEdge with one or two TNode.

Table I. Mesh Partitioner Nomenclature.

Our objective is to provide a simple, reliable and predictable algorithm to capture both strong and weak geometrical features in the context of meshes. The geometrical discontinuities represent natural boundaries for Computer Aided Design (CAD) representations.

Once identified, they are used as separators to create patches of elements that represent a Virtual CAD face representation of the finite element mesh. The key idea is to use a combination of measures DA (Dihedral Angle), TA (Turning Angle), RD (Ridge Direction) and AD (Angle Defect) to select the best candidates feature edges to complement the initial unconditionally strong set of feature edges (pass the DA threshold).

A commonly used criterion to identify what is called strong edges is based on the face angle (or DA), which roughly approximates the principle curvature of the surface at an edge. If  $n_1$  and  $n_2$  are the (outward) unit normals of the incident facets of an edge  $e$ . The face angle at  $e$ , denoted by  $\angle e$ , is the angle between  $n_1$  and  $n_2$ , i.e.,  $\angle e = \cos^{-1}(n_1 \cdot n_2)$  or  $\pi$  at boundary edges as shown in Figure 10. Given  $\theta$  is in the range  $(0, \pi)$  an edge  $e$  is  $\theta$ -strong in face angle if  $\angle e \geq \theta$ . An edge is called u-Strong Edge or unconditionally strong when  $\angle e$  exceeds  $\theta$  by a large amount. Such edges imply very sharp turns. An edge is called l-Strong Edge or locally strong when the angle is more than the limiting angle but less than a secondary high limit. Examples of these edges are shown in Figure 11.

The combination and correlation of these measures provide a high level of confidence on the best candidates selected.

The main steps are outlined below:

1. The unconditionally strong edges (u-Strong Edges) are detected.
2. The locally strong edges are added.
3. Concave corners are detected, and shallow edges are uncovered.
4. Pruning of scar edges can be performed if needed (can be based on a jaggedness/zigzaggedness measure).
5. Virtual Plane cut is performed if there are no unconditionally strong edges that are detected.

The predicate that is used the most is:

- A) Start with a seed edge.
- B) Find the next best candidate edge according to a set of rules that combine all the measures and ranks them.

Typical examples of strong and weak feature edges are shown below in Figure 11.

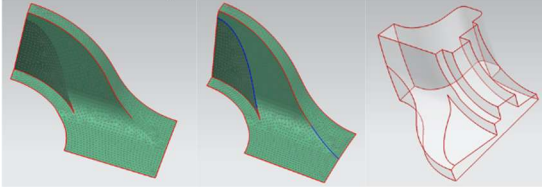


Figure 11. Feature edges: a) u-Strong Edge in red; b) l-Strong Edges in blue (our algorithm); c) entire model.

## 6. MESH-GEOMETRY FIDELITY

The orphan mesh or legacy finite element model is one which has no parent geometry. When an analyst or designer deal with such models there are two distinctly different needs :

- a) To regenerate geometry from the mesh in order to modify it.
- b) To remesh the existing mesh without creating any geometry to either perform a quick analysis of the model in part or whole subject to different types of load or boundary condition or analyses types.

The present investigation deals with the latter. Thus it is important to understand that there is no owning geometry to check mesh-geometry fidelity with. The reference “geometry” is the input mesh in this case as shown in Figure 7a. For the purpose remeshing a temporary, meta-geometry is created on this input mesh patch via the Mesh Partitioner as described in section 5. This meta-geometry face(s) or FOM face is parameterized by means of a tessellated face flattening method by Beatty [18]. The outline of this *Weighted Edge Flattening Method* (WEFM) is explained below.

The faceted FOM faces consists of the triangulated input mesh. The triangular elements are the facets here. Facet faces are defined by facet vertices. Neighboring triangles share the same facet vertices. The faceted faces to be flattened have a loop and edge topology. The edges are directly defined by the edges of facets. The edges can be both on the exterior and interior of the 3d face. Edges on the interior of a face have two uses by the loops, which are in different orientations. Edges interior to the face can be added for different reasons. These edges can be used to make non-manifold connections between different faces. They can be added to enforce meshing on the end points and along the edge which is interior to the surface. They can also be added to help the creation of a low distortion domain in two dimensions. For example, an interior edge can be added between two end loops of a cylinder. If the two different uses of the edge are allowed to split apart in a UV domain, then we can get a rectangle where the vertices along the repeated

edge have different uses with different UV values. An edge so used is commonly referred to as a “seam”. Originally, we distinguished between interior edges that were from cutting a cylinder, and edges that were added for other purposes. However, there were two problems with this approach. The first is that an edge could have different uses on different faces. (This problem could be rectified by considering edge uses.) The second problem was that a small modification of the topology of a surface could require reevaluating the status of all edges of the surface. These considerations lead us to the following assumptions and requirements for our flattening process.

Assumptions for the faceted face to be flattened are

- a. The set of facets that represent the face are manifold but can include many holes/boundaries.
- b. The faceted faces do not have cracks. All internal edges are represented in a loop/edge topology that is directly related to the facet edges of the 3d triangulation. All free edges of the faceted face are represented in the face topology of loops and edges.
- c. Internal cuts that help with the flattening process have been added in a previous stage of abstraction to the loop/edge topology. (It is not necessary to specify if the cuts are to be opened up or kept together in the two dimensional domain. That is we do not have to specify if a cut is a “seam” or if it is a “scar”.)

Requirements flattening such a faceted face:

- i) *Produces a 2-d domain with minimal global and local distortion. Ability to trade-off shape distortion, dimension distortion, and performance depending on application.*
- ii) *Supports multiple loops and non-convex boundaries*
- iii) *Automatically finds constraints between the repeated edges of a face's topology that lower distortion in the 2d domain.*
- iv) *Overall procedure must be fast and robust.*

In brief, the WEFM method is a compromise between authalic mapping (area based) and conformal mapping (shape based) methods which tries to flatten the collection of 3D facets onto a 2D parametric space such that the 2D facets are both shape(i.e. corner angle) and area preserving. The function to be minimized consists of a surface energy term, a constraint term to center the UV domain at the origin, a general constraint term that can be used to eliminate degeneracy and enforce boundary conditions, and finally a line boundary energy term.

The governing equation and its proposed solution is described by Beatty in details [18]. Upon solution, a transformation exists between the original 3D triangulation (which defines the input orphan element patch) and its 2D domain. Remeshing is done in this flattened WEFM domain. Therefore, after remeshing, every single patch-interior node of the remesh is transformed back to the 3D triangulation. This guarantees the final remesh always exists on the

original OEP.

## 7. MESH GENERATION

Once the FOM faces are generated and corresponding mesher-native geometry is created mesh generation follows on these Faces.

Mesh generation uses the CSALF-Q mesher [15] for quad meshes and Subdivision meshes [16] for triangles. A strength of this remeshing algorithm is that it is possible to generate remeshes with a different element type apart from size and local mesh controls. Furthermore, it is possible to use a completely different mesh generation algorithm in comparison to the original Orphan Element Patch. Let's consider the example shown in Figure 7. The original orphan mesh is triangular and generated with a combination of a subdivision and transfinite mesher [16,17, 18] algorithms. Figure 12 shows the remesh on the same patch with quad elements at a lower size with local mapped hole control. The first image (12a) shows the mesh using CSALF-Q mesher while the second image shows the same with a Cartesian-Paver mesher employed (Figure 12b) [19]. The Cartesian-Paver mesh characteristic is conspicuous in its element size uniformity over the entire domain outside the mapped hole zone up until the frozen boundary which is at a bigger size.

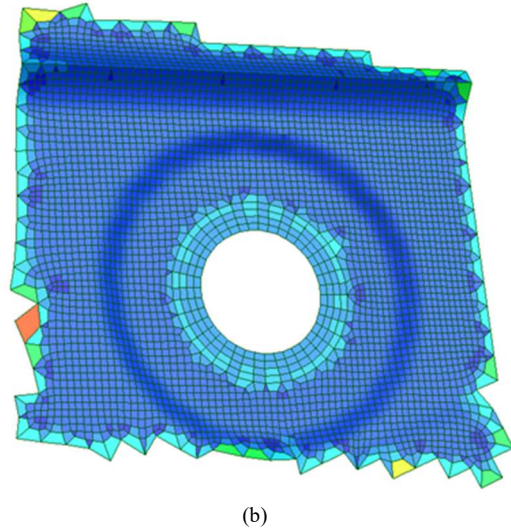
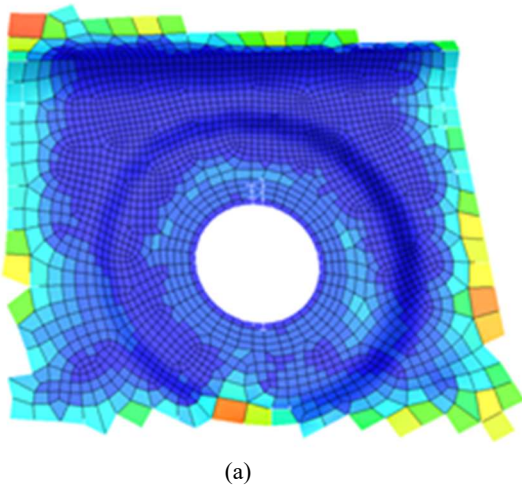


Figure 12. A CSALF-Q quad mesh (a) generated on an OEP described in Figure 7 and a Cartesian-Paver mesh (b) on the same OEP.

## 9. LOADING FROZEN BOUNDARY EDGES

After selecting an Orphan Element Patch (shown in Fig.18 in pale orange) identification of the frozen boundary edges of the patch is very simple. Element edges which are connected to one element within the patch and one element outside the patch are identified as “frozen edges”. The nodes of these element edges are stored as “frozen nodes”.

The frozen element edges are stored in a format defined by their nodes, as shown in Figure 18. For example, edge one is defined by nodes 1 and 2 and edge 2 is defined by nodes 2 and 3.

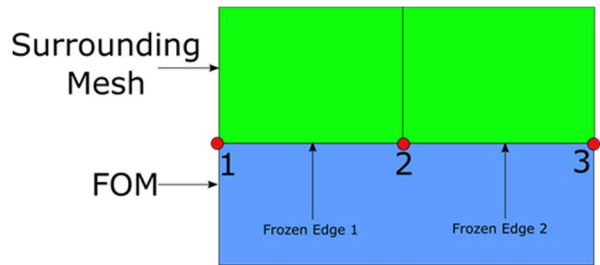


Figure 13. Frozen Edge Format

The next stage is to associate the frozen nodes of the OEP to the FOM. As the facet vertices of the FOM will coincide directly with the location of the nodes on the frozen edges, a map is used to link the frozen nodes to these facet vertices. Any facet vertex on the boundary of the FOM coinciding with a frozen node is flagged as frozen.

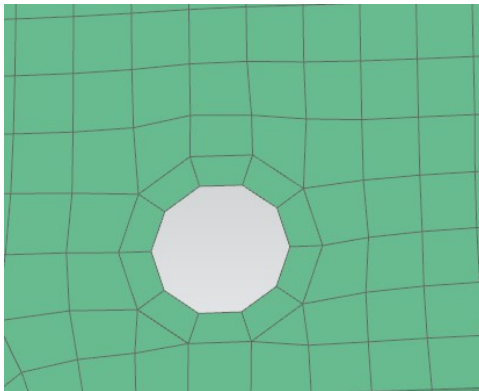


## 8. HONORING LOCAL MAPPED HOLE DEFINITIONS

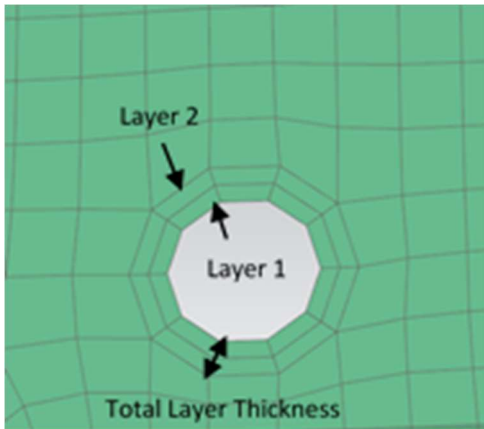
The parameters that control mapped hole definitions, as shown in Figure 14, are:

- Number of Elements
- Number of Layers
- Total Layer Thickness

For OEP which have more than one loop, the outer loop of the FOM is identified as the loop which has the largest enclosed area. All other loops will be inner loops. The next stage is to identify only the convex inner loops based on a limiting angle of convexity. If any angle on the loop exceeds this angle the loop cannot be called convex. Each convex inner loop will have aforementioned mapped hole definitions applied to them. This mapped hole definition will be honored by the controlled by a loop-paving technique developed by Mukherjee [15].



(a) Before remesh



(b) After remesh

Figure 14. Mapped Hole Meshing

## 10. HONORING SPOT WELD DEFINITIONS

Spot and seam welds are common to automotive engineering. In body-in-white (BIW) automobile panels and frames there are thousands of spot welds joining panels of a large variety of materials. Durability analysis studies the emergence and propagation of cracks which are likely to occur at many spot welds. This makes spot-weld meshing a sensitive and intricate finite element modeling operation. Spot-weld mesh is typically desired as annular layers of elements around the weld spot or point. Spot-Weld control is thus defined by the following parameters –

- Number of spotwelds -  $N_{sw}$
- Point of application or 3D location of spotweld -  $P_{sw}$
- Number of elements around the spotweld -  $n_{swe}$
- Number of layers or annular rings -  $n_{swl}$
- Total layer depth -  $D_{sw}$
- Depths of each layer, e.g. i-th layer depth is -  $D_{swi}$

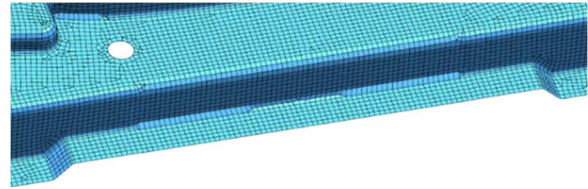


Figure 15. An automotive body panel Orphan Element Patch representing a flange region

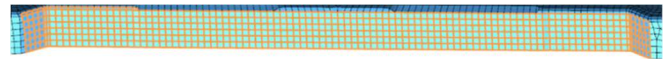


Figure 16. A section of the flange region selected for remeshing

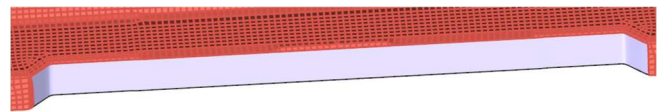


Figure 17. A mesh-native FOM face (in off-white) is created by the Mesh Partitioner

The meshing process algorithm for the spot welds is described by Algorithm I

---

**Algorithm I – Spot Weld Mesh Processing**

---

- A)  $N_{sw}$  of spot welds are projected on their candidate faces
- B) Evaluate 2D coordinates  $(u,v)$  of the spot on the FOM face
- C) Face boundary is discretized with mesh nodes
- D) Face topology in 2D is substituted by node-loop topology. A node-loop is a sequence of nodes on the edges of a face loop. Face interior spot weld points are treated as a 1-point node-loop.
- E) Start loop over number of spot welds ( $N_{sw}$ )
  1. For each spot-weld ( $i = 1 - N_{sw}$ )
    - a) Cycle the number of layers ( $n_{swl}$ ) for the  $i$ -th spot weld  $j = (0, n_{swl})$ 
      - a1) if  $j = 0$ , we create triangles instead of quads. for the  $j$ -th layer we create  $n_{swc}$  quads using a paving algorithm. Depth of the elements created =  $D_{swj}$  for the  $j$ -th layer
      - a2) A new resultant node-loop ( $NL_{ij}$ ) is created from the boundary of the output mesh
      - a3) This resultant loop is checked with all other face node-loops for intersection
      - a4) If there is no intersection this node-loop replaces the old node-loop.
    - b) Go back to (a)
      - When  $j = n_{swl}$ , end cycle over the number of layers ( $n_{swl}$ ) for the  $i$ -th spot weld.
      - go back to 1
  - End cycling  $N_{sw}$  spot-welds when  $i = N^{sw}$

The resultant face node-loop topology, now representing the area of the face outside the Spot weld zone is handed over to the 2D mesher.

---

Automatic remeshing with spot-weld controls is shown in Figures 15-18. A flange section of a geometry-orphaned finite element mesh in blue is shown in Figure 15. A part of this mesh, some flange elements (in orange) are selected and passed to the remesher (Figure 16). Based on the overall algorithm described in Algorithm I, the Mesh Partitioner (MP) creates a FOM face on the OEP as shown in Figure 17. The spot welds are located on this face and meshed according to Algorithm VII. A close-up of the mesh is shown in Figure 18.

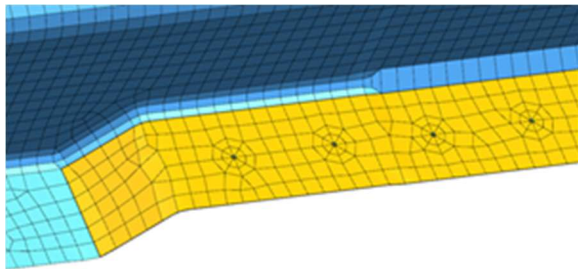


Figure 18. Remesh on the OEP with spot-welds inserted

## 11. STITCHING REMESHED OEP TO ORIGINAL MESH

After the mesher-native FOM (Face On Mesh) has been created (to remind: this face is unseen by the user as it is created under the sheets as part and parcel of the automatic remeshing process) and the OEP (Orphan Element Patch) has been remeshed (as shown in Figure 19), the new elements need to be stitched to the original mesh. The newly created patch, as it comes out from the remesher, is a set of node objects and their coordinates. These nodes need to be created in the original mesh.

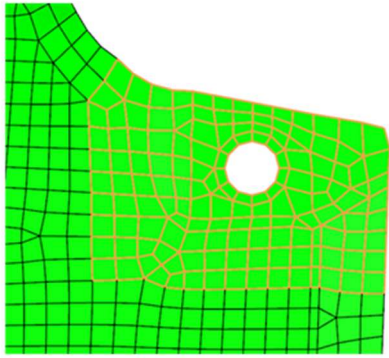
While stitching back the OEP, the free boundaries are treated differently from the frozen ones. In the case of a frozen boundary, as a first step, the frozen nodes are identified among the newly created nodes. Once they are identified, they should not be recreated to avoid redundancy as they were kept intact in the original mesh.

---

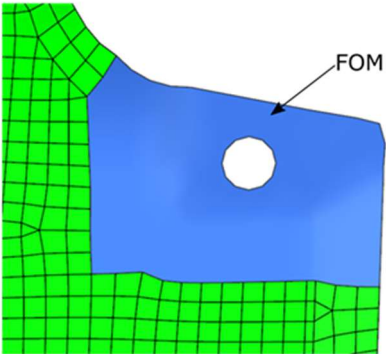
**Algorithm II - Stitching Remeshed OEP to Original Mesh**

---

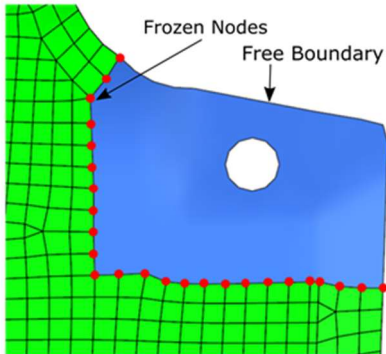
1. Loop through all node's  $N_i$  of the remeshed OEP
    - a. Get the coordinates  $(X_i, Y_i, Z_i)$  and Ids  $Id_i$  of these nodes.
    - b. Store the frozen boundary node id of the original OEP and those of the remesh in an STL map  $Map_n$
    - c. If  $(Id_i = Id \text{ of a frozen node})$ 
      - Skip this node
      - Else
        - Create the node with the coordinates  $(X_i, Y_i, Z_i)$  in the original mesh
  - End loop over nodes of the remeshed OEP
  2. Loop through all elements  $E_i$  of the remeshed OEP
    - a. If element has no nodes on the frozen boundary of the original OEP create them
      - i) Use new ids for nodes in the interior of the remeshed OEP
        - Else if there are nodes on the frozen boundary of the OEP
      - ii) Use the node-map  $Map_n$  to get the frozen node ids from the remesh node id
        - b. Create the element of the remeshed OEP
  3. Delete all nodes of the original OEP except for the frozen boundary nodes
  4. delete all elements of the original OEP
-



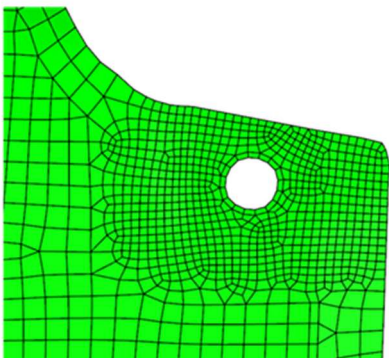
Stage 1 - Select Orphan Element Patch



Stage 2 - Create face on mesh (FOM)



Stage 3 - Store frozen nodes on boundary



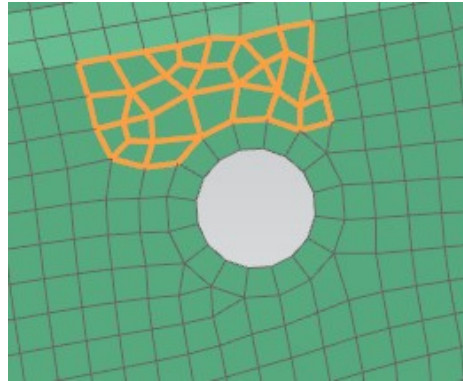
Stage 4 - Mesh FOM at a different size

Figure 19. FOM Remesh Workflow

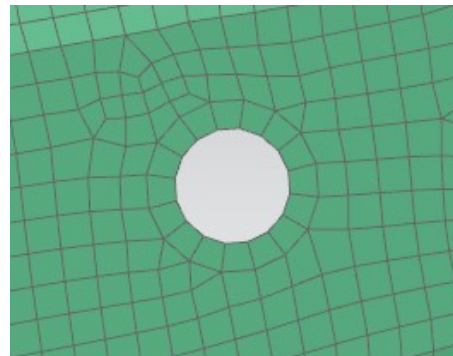
## 12. MESH QUALITY COMPARISON

The purpose of refining or coarsening the mesh is usually to run a different kind of analysis on the model, or to take into consideration new boundary conditions. The initial mesh represents the geometry, so the new mesh must stay conformal to the original mesh. The quality metrics used depend on the type of the analysis ran.

The quality of the newly generated mesh has to be equivalent to the original mesh (Figure 20, Table II.)



(a) Before Remesh



(b) After Remesh

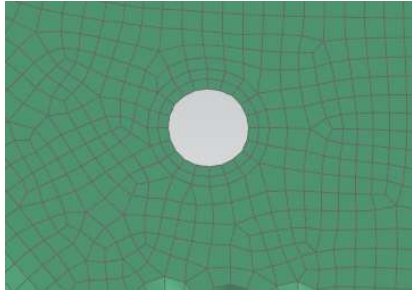
Figure 20. Local remesh

Table II. Mesh quality before and after local remeshing

	Worst Scaled Jacobian	Mean Scaled Jacobian
Before	3.31	0.96
After	3.36	0.94



(a) Before Remesh



(b) After Remesh

Figure 21. Local remesh

Table III. Mesh quality before and after local remeshing

	Worst Scaled Jacobian	Mean Scaled Jacobian
Before	2.1	0.86
After	2.8	0.87

### 13. CONCLUSION

This paper describes an assembly of many algorithms used judiciously to create a mesher-native FOM face which is surface remeshed and stitched back to the original orphan mesh model. An overall design and essential element architecture are defined at the onset. The remeshing strategy uses a new concept called the Mesh Partitioner (MP) which creates a faceted surface representation from the initial Orphan Element Patch (OEP). Face-on-Mesh (FOM), which is a purely polygonal meta-surface resides inside the surface mesher and is very lightweight. The remesher is driven by several user requirements like change in size, element type, controlled loop-pave meshing around holes and patch-interior points to generate spot-weld meshes. Finally, the remeshed mesh on the FOM face is stitched back to the original orphan mesh. Results on various automotive parts demonstrate the need for such automated tools and how they are used to locally refine meshes whose parent geometry is lost.

## REFERENCES

- [1] Basu, D., Cabello, J., Hancock, M. Abstraction on legacy finite element data – re-meshing, surface generation and morphing, US National Congress on Computational Mechanics, Dearborn, 2001.
- [2] Schneiders, R.: Refining Quadrilateral and Hexahedral Element Meshes. *Numerical Grid Generation in Computational Field Simulations 1*, 679–688 (1996).
- [3] Parrish, M., Borden, M., Staten, M.L., Benzley, S.E.: A Selective Approach to Conformal Refinement of Unstructured Hexahedral Finite Element Meshes. In: *Proceedings of 16th International Meshing Roundtable*, pp. 251–268 (2007)
- [4] Staten, M.L., Benzley, S.E., Scott, M.: A Methodology for Quadrilateral Finite Element Mesh Coarsening. *Engineering with Computers* 24, 241–251 (2008).
- [5] Staten, M.L., Woodbury, A.C., Benzley, S.E., Shepherd, J.F. Hexahedral finite element mesh coarsening using pillowing technique, US 8.194.068 B1 (2012).
- [6] CD-Adapco website, viewed 09 August 2019, <https://mdx.plm.automation.siemens.com/star-ccm-plus>.
- [7] CADfix API website, viewed 09 August 2019, [https://www.itiglobal.com/uploadIMG/moxie/CADfix/Docs/CFI\\_API.pdf](https://www.itiglobal.com/uploadIMG/moxie/CADfix/Docs/CFI_API.pdf)
- [8] CGAL website, viewed 09 August 2019, [https://www.cgal.org/2019/08/05/Polygonal\\_surface\\_reconstruction/](https://www.cgal.org/2019/08/05/Polygonal_surface_reconstruction/)
- [9] ABAQUS Version 6, Dassault Systèmes, Providence, RI, USA, 2017.
- [10] J. Cabello, N. Mukherjee, J. E. Makem, W. Daldoul, ‘Modifying a Finite Element Mesh’, Patent Application 2019E20178US, Siemens Digital Industry Software, (2019).
- [11] X. Jiao and N. Bayyana. Identification of C1 and C2 discontinuities for surface meshes in CAD. *Comput. Aided Design*, 40 (2008), pp. 160-175.
- [12] T. Baker. Identification and preservation of surface features. *Proc. 13th Int. Meshing Roundtable* (2004), pp. 299–310.
- [13] Jiao and P. Alexander. Parallel feature-preserving mesh smoothing. *Proc.Int. Conf. Comput. Sci. Appl* (2005), pp 1180–1189.
- [14] X. Jiao, Volume and feature preservation in surface mesh optimization, *Proc. 15th Int. Meshing Roundtable*, (2006), pp. 62–69.
- [15] N. Mukherjee, ‘CSALF-Q: A Bricolage Algorithm for Anisotropic Quad Mesh Generation’, *Proc. XXth International Meshing Roundtable*, Paris, France, (2011) pp. 489-510, Springer.
- [16] J.Cabello, ‘Toward Quality Surface Meshing’, *Proc. XIIth International Meshing Roundtable*, Santa Fe, New Mexico, (2003) pp.201-213.
- [17] K. Beatty, N. Mukherjee, ‘A Transfinite Meshing Approach for Body-In-White Analyses’. *Proc. 19th International Meshing Roundtable*, Springer, (2010) pp.49-65.
- [18] K. Beatty, N. Mukherjee, ‘Flattening 3D Triangulation for Quality Surface Mesh Generation’. In: Garimella R.V. (eds) *Proceedings of the 17th International Meshing Roundtable*. Springer, Berlin, Heidelberg. <https://doi.org/10.1007/978-3-540-87921-3> pp.125-139.
- [19] N. Mukherjee, ‘Multizone Quadrilateral Mesh Generator for High Mesh Quality and Isotropy’, Patent Application 2018P10955WO, Siemens Digital Industry Software, (2018).