# Automatic decomposition of complex thin walled CAD models for hexahedral dominant meshing

Available online at www.sciencedirect.com

ScienceDirect

Procedia Engineering 00 (2016) 000–000

Procedia Engineering

www.elsevier.com/locate/procedia

25th International Meshing Roundtable

# Automatic decomposition of complex thin walled CAD models for hexahedral dominant meshing

Liang Sun, Christopher M Tierney, Cecil G Armstrong, Trevor T Robinson*

*School of Mechanical and Aerospace Engineering, Queen's University Belfast, BT9 5AH, UK*

**Abstract**

This paper describes an automatic method for identifying thin-sheet regions (regions with large lateral dimensions relative to the thickness) for complex thin walled components, with a view to using this information to guide the hex meshing process. This fully automated method has been implemented in a commercial CAD system (Siemens NX) and is based on the interrogation and manipulation of face pairs, which are sets of opposing faces bounding thin-sheet regions. Careful consideration is given to the mapping, merging and intersection of face pairs to generate topologies suitable for sweep meshing thin-sheet regions, and for treating the junctions between adjacent thin-sheet regions. The quality of the resulting hexahedral mesh is considered when making decisions on the generation and positioning of the cutting surfaces required to isolate thin-sheet regions. The resulting decomposition delivers a substantial step towards automatic hexahedral meshing for complex thin-walled geometries. It is proposed that hexahedral meshes be applied to the identified thin-sheet regions by quad meshing one of the faces bounding the thin-sheet region and sweeping it through the thickness to create hexahedral elements.

Peer-review under responsibility of the organizing committee of IMR 25.

*Keywords:* Thin-sheet identification; automatic volume decomposition; hexahedral dominant meshing; sweeping; geometric reasoning

## 1. Introduction

Fully automatic methods for hexahedral (Hex) meshing of solids with good quality elements have been under investigation for many years. Although significant research has been carried out [1]–[9], the complexity of the models that can be meshed with well-structured meshes is very restricted. Manual decomposition of the model into simple sub-volumes is still the main industrial route to achieving this, but this process involves intensive effort and

* Corresponding author. Tel.: +44(0)-28-9097-4187; fax: +44(0)-28-9097-4148.
*E-mail address:* t.robinson@qub.ac.uk


Peer-review under responsibility of the organizing committee of IMR 25.

skill from the user. According to a survey of the analysts in Sandia National Labs, the geometry decomposition and meshing process take about 45% of the total time in modelling and simulation [10]. The vital contribution of the user is difficult to achieve automatically. Since a total removal of user intervention is not possible in the current stage of technology, this research is focused on an incremental approach to Hex meshing, by automatically identifying and partitioning out the thin-sheet regions. The target geometries are complex thin walled structures which are common in the aviation or automobile industry. Hex elements with good quality will be obtained by sweeping a quad mesh through the region thickness, leaving the residual regions to be manually decomposed for a fully Hex mesh or meshed with the tetrahedral (Tet) element to create a Hex-dominant mesh.

There has been much research on automatic volume decomposition for Hex meshing. Price and Armstrong [3] used the 3D medial object (MO) to decompose the volume into a collection of primitive sub-volumes. The method works for a certain class of geometry, but there is superficial treatment of finite contact features, concavities, and n-valent vertices. Kowalski et al. [9] proposed an algorithm using the smoothness driven frame field to generate block structures that can be used to generate full Hex meshes. The main drawback of the approach is that there is no theoretical guarantee that the generated frame field corresponds to the structure of a Hex mesh.

Some research focuses on the identification of the sweep-able volumes. Lu et al. [11] proposed an approach to classifying the edges of the model into different types of loops based on which different cutting faces are constructed. Edge loops offer a good guidance for decomposition but it cannot guarantee to get the best decomposition choice. Some research focuses on the identification of the sweep-able volumes. Lu et al. [12] presented a method using a sketch based user interface to assist the manual decomposition process. It is a great challenge to align the input stroke to the existing boundaries, and the approach requires significant experience from the user. Wu et al. [13] presented a method for identifying the general sweep-able volume in a model. The method begins with extracting all potential sweep directions and generating relevant face sets. Several criteria are set up and assigned a weight to evaluate the cutting faces. However, the source face of the sweep-able volume is currently limited to planar faces and the moving path is assumed to be a straight line perpendicular to the source face.

In terms of thin-sheet identification, Robinson et al. [14] proposed a decomposition process based on the MO for the purpose of idealizing the thin-sheet regions as shell element. Robinson's work on identifying thin sheet regions has been extended by Makem et al. [15] to find the long slender regions using a series of sizing measure methods. Whilst Robinson focused on generating decompositions for stiffened shell models, Makem utilized the thin decomposition to generate mixed solid element meshes. However, the decomposition produced by Robinson isn't entirely suitable for hex meshing purposes as the partitioning boundaries of all thin-sheet regions were offset in an attempt to accurately capture stresses at the junctions. In this work offsets are only performed in situations where they are required to maintain the quality of the hex elements being swept through the thin-sheet region. Yin et al. [16] proposed a method to isolate thin regions using points on an approximate MO computed by an Octree-based algorithm. Full 3D p-version finite elements with the low polynomial order through the thickness are applied on the thin domains. The MO based methods were restricted by the capabilities of a robust and efficient implementation of 3D MO. One of the main drawbacks of the MO for this application is that, for thin walled structures with complex features, a lot of time is spent on the calculation of medial surfaces around the complex features. Sliver edges or faces exist in the MO even for simple geometry and tidying them up requires extra effort such as extending and trimming. This time is wasted for this application as these regions have no contribution to the thin-sheet identification. Makem [15] states that the majority of the process time for decomposition is attributed to the calculation of the MO.

In this paper, a face pair based method is developed to identify thin-sheet regions bounded by the face pair. The face pair technology provides the necessary information to perform the thin-sheet decomposition, but in a fraction of the time. The normal application of face pairing in the thin-sheet regions is to abstract the thin-sheets as mid-surfaces for the purpose of dimensional reduction. Although the start point of mid-surfacing and thin-sheet meshing is the same, the pivotal issues are different. In the mid-surface abstraction, the issues are to generate surface patches and to connect them (by extending or trimming) properly to get a mid-surface topology which is equivalent to the 3D model. Where the regions are being used to guide Hex meshing, the focus is how to decide the boundaries of the target thin-sheet regions, as well as how to create appropriate cutting faces to decompose the models without the generating sliver volumes in either thin-sheets or residual domain which will compromise overall mesh quality.

## 2. Thin-sheet identification

### *2.1. Thin-sheet face pairs identification*

The thin-sheet identification is based upon the initial identification of face pairs [17]. In this paper, the face pairs are obtained from Siemens NX [18] using its mid-surface function. An algorithm has been developed to filter the face pairs provided by NX into thin-sheet face pairs which satisfy the following criteria.

- **Parallel criteria**. The faces in a face pair should be nearly parallel. To verify the parallel criteria, sample points are generated on one face. Let $N_1$ be the outward unit face normal at a sample point. A ray is fired from the sample point with the direction opposite to $N_1$. The intersection point with the opposite face is calculated and the outward unit face normal at that point is $N_2$. If the intersection point exists, then the angle α between $N_1$ and $-N_2$ should be less than the input angle tolerance $\alpha_T$, as given by

$$|Cos\ \alpha| = |N_1 \cdot -N_2| > |Cos\ \alpha_T| \tag{1}$$

  In this work, $\alpha_T$ =5°.
- **Distance criteria**. Let $D$ be the distance between the faces in the pair, while $L$ and $W$ represent the maximum characteristic length and width respectively of each face in the face pairs. To be a thin-sheet region the pair need to satisfy

$$Min\ (L,W)\ /\ D >\ \ R \tag{2}$$

  where $R$ is the aspect ratio input from the user. In this work, the aspect ratio $R \geq 3$ is used.The thickness $D$ can be obtained from NX, but the definitions of the characteristic length and width of a face are ambiguous in both [17] and NX. In this work, a 2D planar representation is calculated to approximate the original 3D face using the pseudo-area and pseudo-normal vector concept defined in [19]. Given a face $S$, its continuous boundaries $\partial S$ are discretized and represented by a list of polygons $\partial S = \sum P_i$. In this work, a polygon is represented using a sequence of points $P_i = p_1 p_2 ... p_m$. The exterior and internal boundaries are distinguished by using polygons with different orientations, counter clockwise (CCW) for the exterior boundary and clockwise (CW) for the internal boundaries. For each polygon $P_i$, a pseudo-area vector $\overrightarrow{a_i}$ is calculated as

$$\overrightarrow{a_i} = \frac{1}{2}\sum_{j=1}^{m} p_j \times p_{j+1}\ \ (p_{m+1} = p_1) \tag{3}$$

  The pseudo-normal vector $\vec{n}$ is then defined as

$$\vec{n} = \frac{\sum \overrightarrow{a_i}}{\|\sum \overrightarrow{a_i}\|} \tag{4}$$

  which is the normal of the planar domain to be used. To use equation (*3*), periodic faces need to be partitioned into two. Once this is completed, it is straightforward to compute the coordinates of the points on the 2D planar representation. The characteristic length and width of $S$ are approximated by the length ($L'$) and width ($W'$) of the minimum bounding box of the 2D planar representation. The minimum bounding box can be calculated using the rotating calipers method [20].

Other concepts used in this paper are explained below with the reference to the model in Fig. 1.

- **Side face**: a face in a face pair. The convention here is that the one with the largest minimum bounding box ($L' \times W'$) is referred to as the side-1 face and the other one the side-2 face, Fig. 1. Where they are equal either is selected. Note one face may appear in more than one face pair, as is the case for the side-1 face in Fig. 1.
- **Wall face**: a face linking the side-1 and side-2 faces, Fig. 1(c).

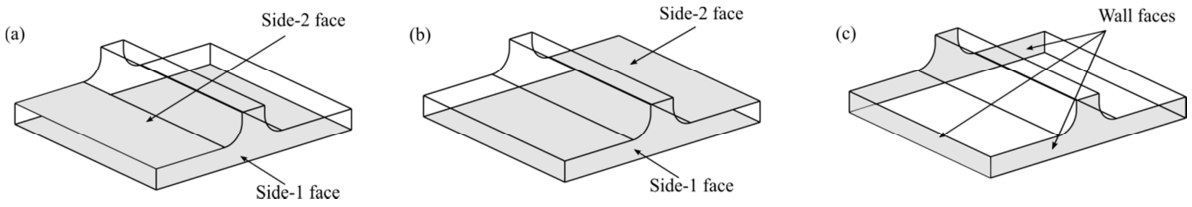


Fig. 1 (a) face pair 1; (b) face pair 2; (c) wall faces for face pair 1

## *2.2. Matching topologies for the thin-sheet region*

### *2.2.1. Overview*

The objective of thin-sheet identification is to obtain equivalent matching topologies on both the source and target of the thin-sheet region to enable a sweepable region to be extracted. The source and target faces are the faces returned from the face pair identification process in the previous section. Matching topologies are obtained using the process depicted in Fig. 2 using the side-1 and side-2 profiles from Fig. 1 (a). The first step in the process is the discretization of the 3D boundary edges of the side-1 and side-2 profiles to create 3D polygons. The discretization size is selected using a combination of the target element size and the boundary size in order to capture the smallest features in the boundary profile. The 3D polygons are utilized for the projection, merging, intersection and mapping stages of the process. These are shown in Fig. 2 (c) – (f) and are outlined in more detail below.

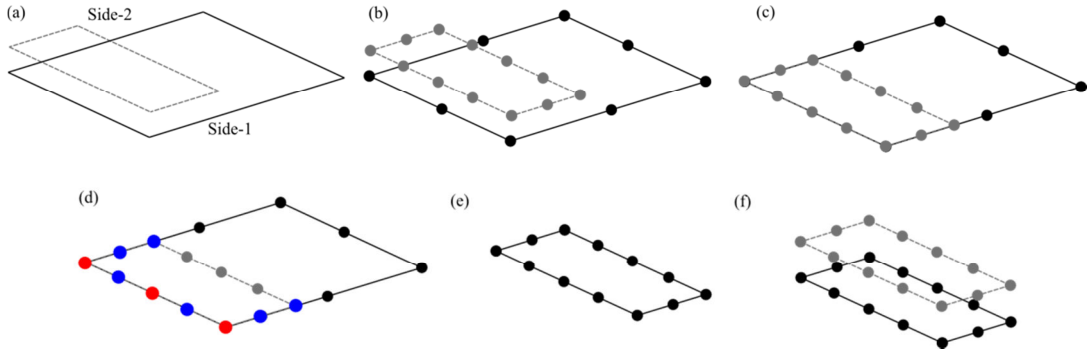


Fig. 2 Generating matching topologies: (a) side-1 and side-2 boundary from Fig 1(a); (b) discretization of boundaries; (c) projection of side-2 discretization to side-1; (d) merging; (e) 2D intersection; (f) 2D-3D mapping.

### *2.2.2. Projection*

Projecting the polygonal boundary representation of one side's boundary onto the underlying surface of the opposite side allows the matching topologies to be generated using simple merge and intersection operations. The side-2 polygonal boundary representation is projected orthogonally onto the side-1 underlying surface to generate the side-2 projection referred to as side-$2_{proj}$ and shown as the grey discretized profile in Fig. 2 (c).

Relationships between the original 3D polygon points on side-2 and their projected point on side-$2_{proj}$ are stored to allow the mapping of the matching topology back to side-2. Fig. 3 (a) shows point {a} on side-2 is linked to point {a'} in the side-$2_{proj}$ on side-1. In general using an appropriate input angle tolerance and aspect ratio (Section 2.1) ensures the projection doesn't impact the downstream mesh quality. Additional care has to be taken if the projection lies outside the surface boundary of side-1. In Fig. 3 (a) the projection of {b} onto the side-1 underlying surface lies outside the side-1 boundary. The closest point on the side-1 boundary {b'} is used if the distance d between the projected points is less than $\sqrt{D^2 + (P\ x\ T_M)^2}$, where $D$ is the distance between the side faces, $T_M$ is the merging tolerance (see section 2.2.3) and $P$ is the parallel coefficient to take into consideration the side are not exactly parallel. If the distance between the projected points is greater than $\sqrt{D^2 + (P\ x\ T_M)^2}$, Fig. 3 (b), then side-1 point {b'} is projected onto side-2 instead, and b'' will become part of the boundary of the thin-sheet. This projection criterion ensures sweepable topologies of acceptable quality are generated throughout the process.

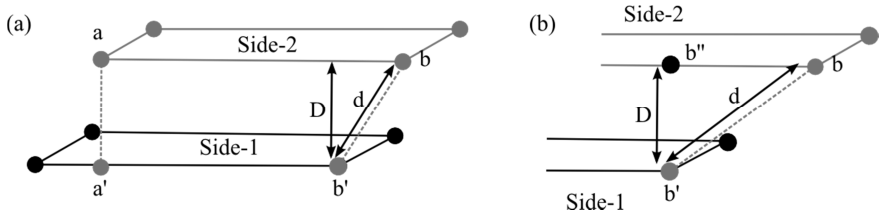


Fig. 3 Projection tolerance: (a) in tolerance; (b) out of tolerance

*2.2.3. Merging*

To assure element quality in the mesh, it is necessary to avoid the creation of sliver regions, and merge the boundaries which may cause them to occur. Two operations are used in this work to identify entities in proximity, namely point-to-point and point-to-line. These operations are performed on the 3D side-1 and side-$2_{proj}$ polygons. Fig. 2 (d) shows the two point merging operations where red points represent close points identified in the point-to-point operation and blue points represent close points in the point-to-line operation.

**Point-to-point** identifies two points as being in proximity if the distance between them is less than the specified tolerance $T_M$. To maximize the benefit of the merging operation, careful selection of the merging tolerance is needed [21]. Normally the size of the merging tolerance $T_M$ is set to be small relative to the discretization element size (10-20%) to guarantee a stable implementation of the merging process. This ensures that points are not identified as being close to more than one point from the opposite side. The point-to-point operation is performed first to guarantee that a point is matched to a close point before considering its projection on a line segment of the polygonal boundary, i.e. the red point in the polygon side-$2_{proj}$ in Fig. 4 (b).

**Point-to-line** determines whether a point is close to a line segment. If a point's projection on the line segment has a normalized parameter $t \in (0,1)$ and a projected distance less than the tolerance $T_M$, then the point is determined to be close to the line segment and its projection point is used as its point in proximity. The point-to-line operation is carried out for any residual points not identified in the point-to-point operation. i.e. all points in Fig. 4 (b) except the red points. Residual points on both side-1 and side-$2_{proj}$ polygons are interrogated against the line segments on the opposing polygonal boundary.

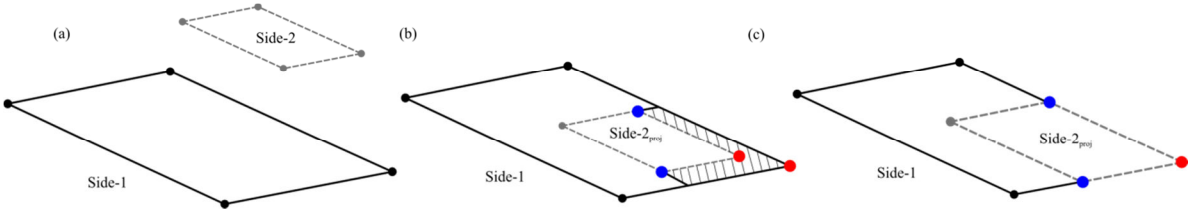


Fig. 4 Merging close points

After the identification of points close to other points, the side-$2_{proj}$ points are replaced by their close points in the side-1 polygon boundary, the red point in Fig. 4 (c). Points close to line segments are replaced with their projected points on the line segments, the blue points in Fig. 4 (c). Modified polygonal boundaries are generated as a result of the merging. The relationships between the modified polygonal boundaries and their original representations are maintained throughout.

This merging step is used to avoid the creation of any sliver faces after the topology matching, the hatched area in Fig. 4 (b), or sliver volumes after the thin-sheet decomposition, Fig. 5 (b), where sliver entities are smaller than the target element size. This treatment of slivers is not considered in other approaches. Misalignments occurring after the orthogonal projection and overlaps / gaps due to boundary profile placements are treated using the aforementioned merge operations, Fig. 4 (c). Only entities considered as slivers are removed in this process, ensuring any mesh distortion through the thickness of the sweepable thin-sheet region is minimal.

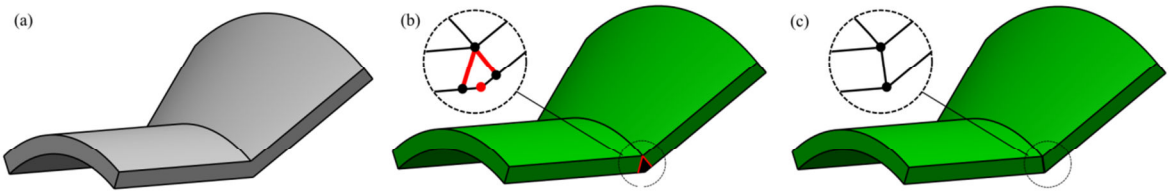


Fig. 5 Merging to avoid sliver volumes: (a) original volume; (b) decomposition without point merging – sliver volume generated; (c) decomposition with point merging - no sliver volumes.

*2.2.4. Polygon intersection*

The polygon intersection gives the matching topologies of the source and target surfaces of the thin-sheet region. To achieve the intersection result the modified polygonal boundaries of side-1 and side-$2_{proj}$ are represented in the parameter space of the surface in which they are contained, i.e. the side-1 surface. 2D polygonal representations of the modified 3D side-1 and side-$2_{proj}$ polygons are generated. This transformation allows the use of simple 2D polygon Boolean operations to obtain the intersection. In this work, the polygon intersection is carried out using the Clipper library [22], which is robust and works for both convex and concave polygons.

The intersection operation is depicted in Fig. 6 (a), where the 2D representations of side-1 and side-$2_{proj}$ are shown in black and red respectively. Having maintained the relationships to the original polygonal representations, the appropriate orientations are assigned to provide the outer and inner loop information required as input. The intersection result is shown in Fig. 6 (b) where four separate polygons are returned. Simple polygon area calculation and relative positions of polygons are used to group the set of matching topologies including their inner and outer loop boundaries.

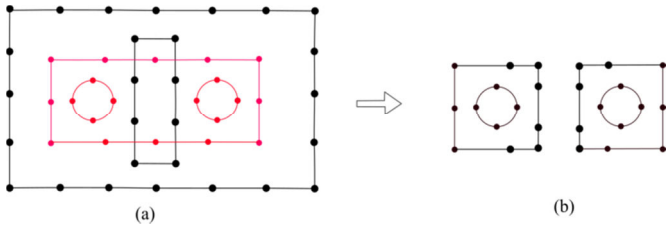


Fig. 6 2D polygon intersection: (a) polygons from both sides in different colors; (b) intersection result

*2.2.5. Boundary mapping*

Having obtained the matching topologies from the intersection operation, the final step is to map these to the 3D boundary that they represent. Storing point relationships throughout the process allows all points in the 2D intersection polygons to be linked to their 3D polygonal representation and classified into one of the following:

- Type-1: a 2D point existing in both side-1 and side-2 uses its corresponding 3D point.
- Type-2: a 2D point that exists in side-1 or side-2 is projected to the opposite side.
- Type-3: a 2D point that exists in neither side-1 nor side-2, i.e. a new intersection point, is mapped to side-1 to get the 3D side-1 point and then projected to side-2 to get the 3D side-2 point.

When a point is mapped to side-1 or projected to side-2, the polygonal boundary representations are interrogated to determine if it needs to be transformed to an original boundary edge of the side face.

*2.3. Identify and isolate thin-sheets*

*2.3.1. Construct side edges and wall edges*

To decide the cutting faces which will be used to partition the model, the bounding edges are first generated on the side faces, based on the boundary points calculated in the last section. These edges are referred to as side edges.

A side edge will either be a new edge resulting from the intersection with the opposite boundary face, an existing edge or a subset of an existing edge. In all situations, the start and end points of the edge need to be defined, i.e. the edge bounding vertices. A point is defined as the vertex of a side edge if it comes from an existing vertex of the continuous boundaries or it represents a new intersection point, as shown in red in Fig. 7 (a). Given the points set p and the vertex indices set I, the side edge $E_j$ can be constructed from one vertex point to the next vertex point as

$$p = \{p_i\ , i \in [0, n-1]\},\ I = \{I_j, j \in [0, m-1]\}, E = \{\ E_j = \left[p_{I_j}, p_{I_{j+1}}\right], j \in [0, m-1], I_m = I_0\ \} \tag{5}$$

If all the points defining the side edge lie on the same existing edge, then the geometrical curve of the edge will be extracted and the portion that is bounded by the vertices of the side edge will be used as the underlying curve of the side edge. Otherwise, a new curve will be generated by fitting through the defining points in the side face. For the face pair 1 in Fig. 1, there will be 8 side edges, 7 of which are extracted from existing edges (shown in black in Fig. 7(b)) and the other one created by fitting a curve through points (shown in red in Fig. 7(b)). The relationship between a side edge and its corresponding side edge on opposite side face is defined during the merging and intersection procedure. They are referred to as a side edge pair, Fig. 7(b). From this information edges connecting the matched vertices of the paired side edges are defined, which are referred to as wall edges. Similar to the side edge, a wall edge is either an existing edge on an existing wall face, a new edge on an existing wall face or a new edge through the volume of the thin-sheet region. There will be two wall edges based on existing wall edges (shown in black in Fig. 7(c)) and two wall edges created from fitting a curve on the surface (shown in red in Fig. 7(c)).

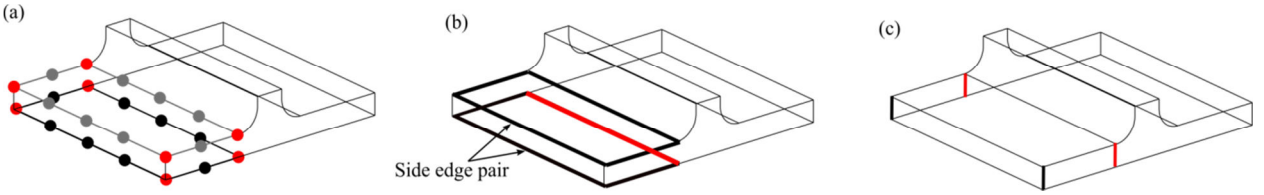


Fig. 7 For face pair 1 in Fig. 1: (a) boundary points of thin-sheet, the vertex points are in red; (b) side edges: 1 fitted edge in red, 7 from existing edges in black; (c) wall edges: 2 fitted edges in red, 2 from existing edges in black

*2.3.2. Construct cutting faces*

Each pair of side edges and the connected wall edges define a four-sided face which may be used to partition the solid model, referred to as a candidate cutting face. The candidate cutting face will be either an existing wall face (or a subset of an existing wall face), or a new face. For example, there will be four candidate cutting faces for the face pair in Fig. 7, one of which is a new face and the other three are from existing wall faces. In order to guarantee the quality of mesh which can be created in the model (including the thin-sheet regions and the residual regions), it is necessary to determine whether a candidate cutting face should be used directly or a new cutting face should be created.

The candidate cutting face from a new face will be either used as is or offset into the thin sheet region to ensure the region remaining after the decomposition is still suitable for a good quality mesh. The model in Fig. 8(a) has two candidate cutting faces which are new faces, as highlighted. The left one can be directly used. However, an offset will be required for the right one, Fig. 8(b), otherwise the partitioning will generate a sub-volume with a dihedral angle close to zero degrees, into which a poor quality Hex element will have to be inserted.

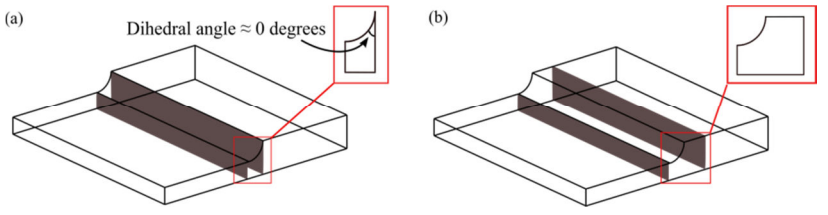


Fig. 8(a) result by using the candidate cutting faces; (b) result by offsetting the right-hand side candidate cutting face which is from a new face

Where the candidate cutting face is an existing wall face, it will not be used to partition the model (as no partitioning is required). But it needs to be determined whether a new cutting face is required by creating a new face bounded by the same side edges, or by offsetting the current side edges. For the face pair in Fig. 7, the three candidate cutting faces from existing wall faces are not necessary for the decomposition and only the one from a new face will be used. The decomposed thin-sheet and the Hex mesh generated with the sweeping algorithm are shown in Fig. 9 (a). Note that the Hex mesh generates a structured quadrilateral mesh on the wall faces, i.e. every internal node has 4 incident elements. However if the model in Fig. 7 has a curved end on the left, with the same cutting face, the decomposed thin-sheet and the Hex mesh will be as shown in Fig. 9 (b). It still generates a structured quadrilateral mesh but the quality of some mesh elements on the front wall face is poor. For such a face, to get a mesh with good quality (boundary aligned and with corner angles ~ 90 degrees), mesh singularities must be introduced. Nodes in the mesh where more or less than 4 elements meet are known as mesh singularities (positive singularities if >4 elements meeting at the node and negative singularities if <4 elements at the node). Armstrong et al [23] proposed an equation to calculate the net number of the mesh singularities on a face.

$$\sum(\frac{\pi}{2}N-\theta)+\sum\int k_g ds+\iint KdA+(n_+-n_-)\frac{\pi}{2}=0 \tag{6}$$

In the equation, $\theta$ is the corner angle; $N$ the optimum element number at a vertex which can be calculated using

$$N=round\,(\frac{\theta}{\pi/2}) \tag{7}$$

$k_g$ the geodesic curvature of the edges; $K$ the Gaussian curvature of the face; $n_+$ and $n_-$ the number of positive singularities and negative singularities. The addition of the first three terms always gives a multiple of π/2. For example, suppose the boundary of the shape in Fig. 9 (c) is comprised of a half circle and straight lines. For this shape, the first term in equation (6) is zero since all the angles are a multiple of π/2; the second item is

$$\sum\int k_g ds=\int\frac{1}{R}ds=\frac{1}{R}\times\pi R=\pi \tag{8}$$

and the third item is zero since it is on a planar face. So

$$0+\pi+0+(n_+-n_-)\frac{\pi}{2}=0 \tag{9}$$

$$n_+-n_-=-2 \tag{10}$$

meaning two negative singularities are required for this shape, as shown in Fig. 9 (c). Therefore, a new cutting face is needed to isolate the singularities and further decompose the model into two volumes, as shown in Fig. 9 (d). The right-hand side one will be a thin-sheet and meshed with the sweeping algorithm. The left-hand side one will be left to mesh with other Hex meshing algorithms or filled with Tet elements.

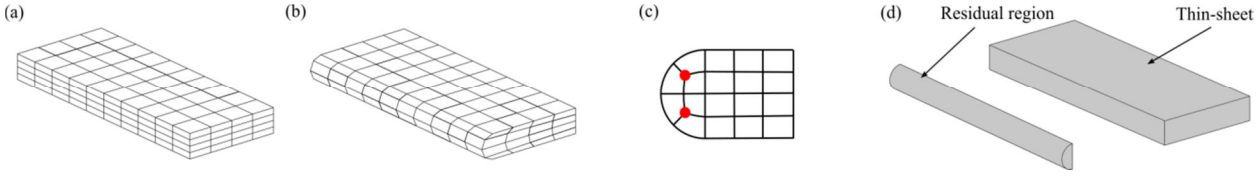


Fig. 9 (a) thin-sheet and Hex mesh for the face pair in Fig. 7; (b) thin-sheet and Hex mesh if the model in Fig. 7 has a curved end; (c) example of the mesh singularity for a wall face in (b); (d) desired decomposition result

Equation (6) only gives the net number of the mesh singularities, but the positions of the mesh singularities are not known. Here a method for determining whether and by how much to offset a candidate cutting face is proposed

based on the optimum element numbers at the side edges and the mesh singularity number. This allows the mesh singularities to be isolated and ensures the introduction of the cutting face will not result in narrow regions. A slice is made at the middle point of one side edge and perpendicular to the tangent of the side edge. On the cross sections in Fig. 10, the side faces become curves (shown in bold black lines) and the side edges become two points A and B. The candidate cutting face is shown in a red solid line if it is from an existing face and in a dashed red line if it is from a new face. Let $N_A$ and $N_B$ be the optimum element number at the corners with the side edges.

- $N_A = 0$ or $N_B = 0$: an offset will be made otherwise the quality of the element at the corner will be poor, Fig. 10 (a).
- $N_A = 1$ and $N_B = 1$: if the candidate cutting face is a new face, Fig. 10 (b), offset has to be made otherwise it will generate a volume with a dihedral angle close to zero degrees. If it is from an existing wall face, Fig. 10 (c), a straight line CD is added to the section curve perpendicularly to form a closed four-sided shape ABCD. The number of singularities of the enclosed shape should be zero. To avoid the existence of a pair of positive and negative singularities, sample points are created along AB, CD based on the number of element layers, e.g. E, G and make the curve connecting them parallel to AC. The number of singularities on the sub-shape AEGC should also be zero. In Fig. 10 (b), if AE is a quarter of a circle and $\theta_E$=0, it can be calculated that there will need to be a positive singularity in AEGD, so an offset will be made, Fig. 10(d). If there are no singularities in any sub-shape, the candidate cutting face is not used and no other faces are needed, Fig. 10 (e).
- $N_A = 1$, $N_B > 1$ or $N_B = 1$, $N_A > 1$: an offset will be made, Fig. 10 (f).
- $N_A \geq 2$ and $N_B \geq 2$: if the candidate cutting face is from an existing wall face, a new face connecting the two side edges is created. Sample points are generated on the curve representing the candidate cutting face based on the desired number of element layers through the thickness. If the distance between a sample point and the new face is larger than the target element size, then the newly created face will be used, Fig. 10 (g). If the new face intersects with the candidate cutting face, Fig. 10 (h), or the minimum distance between them is less than the target element size, Fig. 10 (i), then the cutting face will be created by offsetting the side edges of the candidate cutting face. If the candidate cutting face is from a new face, Fig. 10 (j), similar checks and decisions are made between sample points on curves between A and B and the candidate cutting face. For the example in Fig. 10 (j), the candidate cutting face will be used.

If an offset is expected, each side edge of the candidate cutting face is offset on its side face. The offset directions are the tangent of the intersection curves with the side faces at point A and B respectively, as $T_A$ and $T_B$ shown in Fig. 10 (a). A point on the intersection curves between A and B is identified which has the maximum projection distance along the offset direction $T_A$, as the green points shown in Fig. 10. Here the offset is made so that the minimum distance $L$ between this point and the final cutting face is half of the local thickness of the thin-sheet. After offset, the wall edges are created as straight lines or curves on the existing wall faces connecting the matched vertices of the offset side edges. The final cutting faces are generated by filling the region bounded by the paired side edges and wall edges.

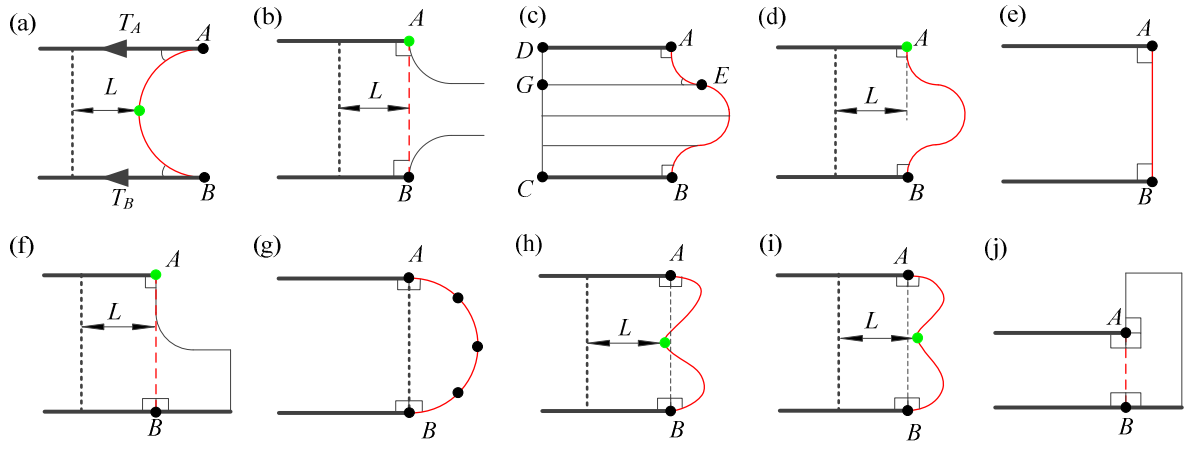


Fig. 10 Different treatments of the candidate cutting faces

*2.3.3. Geometrical decomposition*

The final cutting faces are used to decompose the model. After the decomposition, to determine which region is a thin-sheet, a test point is created by triangulating the final side faces on side-1 in the parameter space and identifying the centroid of the largest triangle. The region that contains the test point will be identified as a thin-sheet.

## 3. Results and discussion

The described approach has been implemented using the C# language and .NET framework APIs in Siemens NX 10. The thin-sheet identification processes are implemented on a 3.4 GHz Intel Core (TM) i7-2600 CPU machine with 16GB RAM. Results for a selection of models are given in Fig. 11 where the thin-sheets are shown in green and the residual regions in yellow. The time spent in the identification process, the number of thin sheets and the volume occupation are summarized in Table 1.

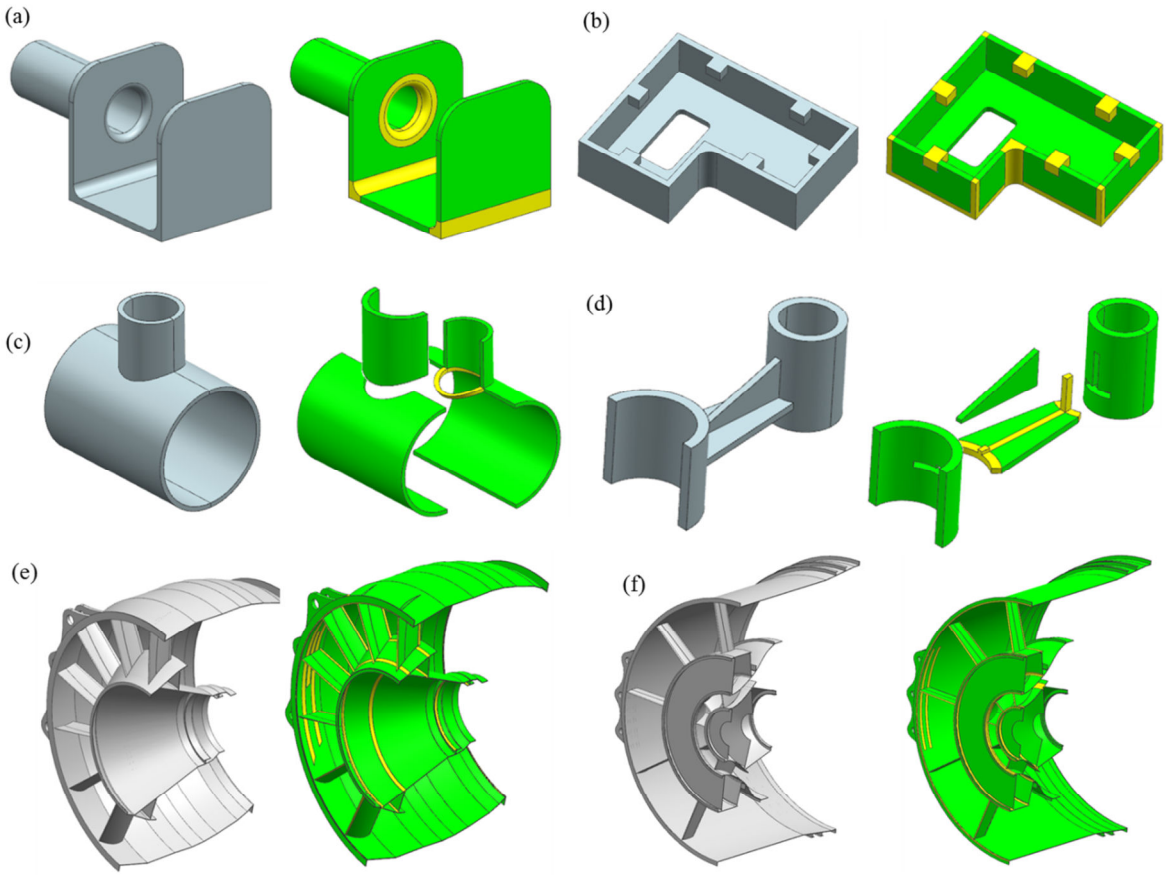


Fig. 11 Thin-sheets identification results

Table 1 Degree of freedom of the four analysis models

| Models | a | b | c | d | e | f |
|---|---|---|---|---|---|---|
| Time (s) | 4.5 | 7 | 12 | 6 | 205 | 353 |
| Number of the thin sheets | 5 | 7 | 4 | 6 | 60 | 88 |
| Volume of the thin sheets (%) | 83 | 81 | 99 | 96 | 92 | 90 |

### 3.1. Degree of freedom (DOF) reduction

The purpose of identifying and isolating the thin-sheet regions is that significant DOF saving can be achieved by using anisotropic Hex elements in the thin-sheet regions. A representative industrial model, the CRESCENDO compressor inter-casing [24], is used here for demonstration. The DOF and the analysis accuracy are compared between a benchmark Tet -meshed model and the mixed elements meshed model, where the thin-sheets are meshed with Hex elements, and the residual regions are meshed with Tet elements. For clarity only 1/6 of the model is used. The original model is shown in Fig. 12(a) and it was meshed with dense 10-node Tet elements as shown in Fig. 12(b). The decomposed solid model is shown in Fig. 12(c) where 46 thin regions have been identified for this model and occupy approximately 90% of the model's volume. Three mixed element analysis models were generated for comparison, and for all the models, pyramid elements are generated to transition between Hex elements and Tet elements. The meshes are manually created in NX and consist of:

1. Isotropic Hex for thin-sheets: the thin-sheets are meshed with isotropic 8-node Hex elements and the residual regions are meshed with 10-node Tet elements, Fig. 12(d).
2. Anisotropic Hex for thin-sheets: the thin-sheets are meshed with anisotropic 8-node Hex element with an aspect ratio (lateral dimensions to thickness) varying up to 5, Fig. 12(e).
3. Anisotropic Hex for thin-sheets and long-slender regions: after identifying the thin-sheets, there are some long-slender regions in the residual regions connecting two adjacent thin-sheets. Makem et al [15] noticed that there are a lot of DOF consumed when the long-slender regions are filled with Tet mesh. These long-slender regions are identified and meshed with anisotropic Hex element as shown in blue in Fig. 12(f).

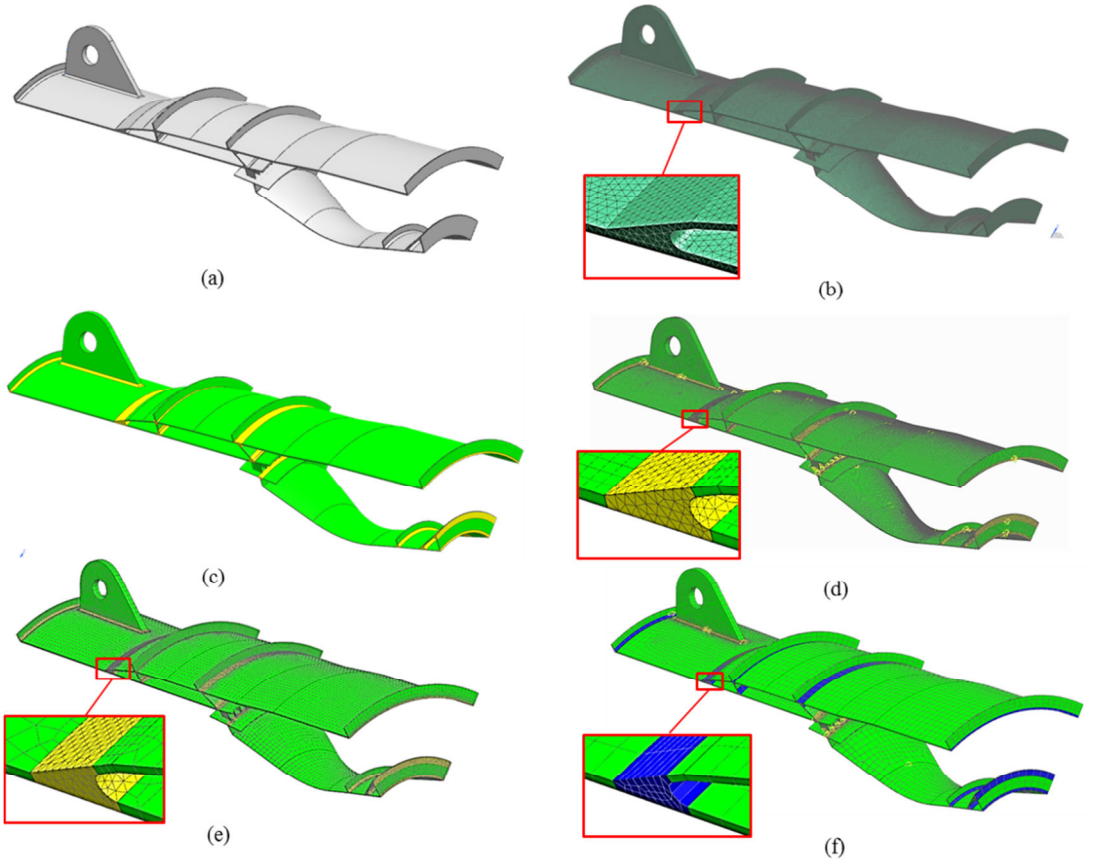


Fig. 12 (a) original casing model; (b) Tet elements meshed model; (c) decomposed casing model; (d) thin-sheets meshed with isotropic Hex element; (e) thin-sheets meshed with anisotropic Hex element; (f) thin-sheets and long-slender regions meshed with anisotropic Hex element

The degrees of freedom of the four analysis models are given in Table 2. It can be seen that the DOF of the mixed element analysis models in Fig. 12(d), Fig. 12(e) and Fig. 12(f) are 26.1%, 17.7% and 5.9% of the dense Tet elements.

Table 2 Degree of freedom of the four analysis models

| Analysis models | Tet for the whole model | Isotropic Hex for thin-sheets | Anisotropic Hex for thin-sheets | Anisotropic Hex for thin-sheets and long-slender region |
|---|---|---|---|---|
| DOF | 2191611 | 572706 | 388803 | 129897 |
| % Tet | 100 | 26.1 | 17.7 | 5.9 |

Free-free modal analyses were conducted on the models and the first 20 modal frequencies are compared. The analysis result of the dense Tet elements meshed model is used for reference. Fig. 13 compares the discrepancy in the modal frequency of the models meshed with mixed elements to the reference dense Tet mesh. According to Fig. 13, the maximum discrepancy between the results of the dense Tet meshed model and that of the other models is 1.8%. These comparisons reveal that accurate results can be achieved using these model types.

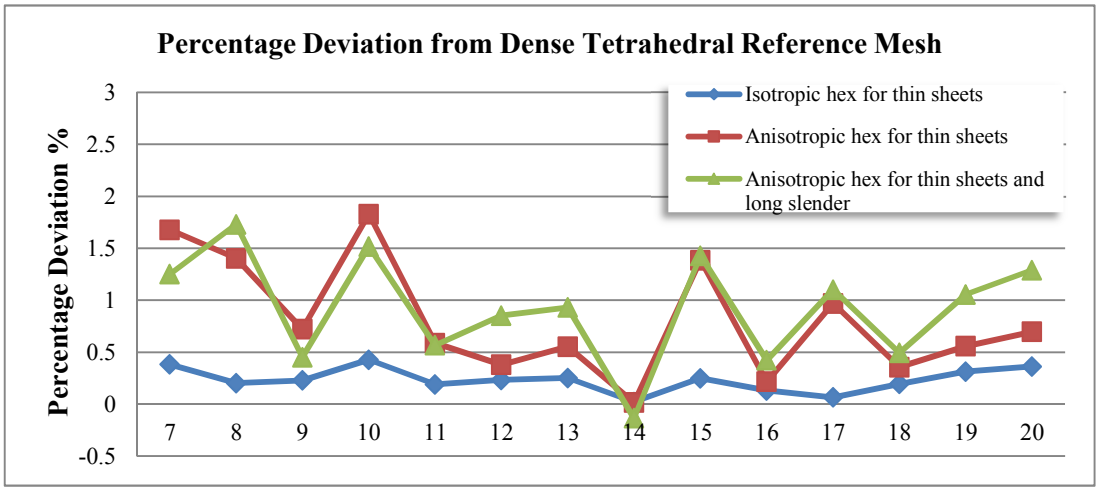


Fig. 13 Comparison of modal frequencies for the inter-casing model, the result of the dense Tet elements meshed model is used as standard

The thin-sheet identification described here could be applied to the creation of the dimensionally reduced model where the thin-sheets can be idealized as mid-surfaces while the residual regions remain in 3D to create a mixed-dimensional model [25]. It can also be used as an input to a virtual topology tool to generate a virtual decomposition where the decomposition is performed at the topology level with the underlying geometry unchanged [26].

## 4. Conclusions

This paper presents an approach to identify and isolate thin-sheet regions in a model. For these regions, Hex meshing can be generated by the sweeping algorithm. The automatic approach brings a significant reduction of manual effort for the Hex meshing generation process and significant DOF saving can be achieved by applying anisotropic Hex elements to the thin-sheet regions. The approach is based on the initial identification of face pairs, after which a series of operations, such as mapping, merging and intersection, are performed to decide the candidate cutting faces. An approach based on mesh element quality and mesh singularity configurations is used to decide upon the generation and positioning of the cutting surfaces required to isolate thin-sheet regions. Several industrial models have been used to demonstrate the results. Further benefit can be achieved by identifying other regions, such as the long slender regions.

## Acknowledgements

The authors wish to acknowledge the financial support provided by Innovate UK: the work reported herein was funded via GHandI (TSB 101372), a UK Centre for Aerodynamics project. Fig. 11(e), Fig. 11(f) and Fig. 12 are created using a Rolls-Royce model provided through the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 234344 (www.crescendo-fp7.eu).

## References

[1] W. A. Cook and W. R. Oakes, "Mapping methods for generating three-dimensional meshes," *Comput. Mech. Eng.*, vol. 8, pp. 67–72, 1982.
[2] D. White and L. Mingwu, "Automated hexahedral mesh generation by virtual decomposition," in *4th International Meshing Roundtable*, 1995, pp. 165–176.
[3] M. a. Price, C. G. Armstrong, and M. a. Sabin, "Hexahedral mesh generation by medial surface subdivision: Part I. Solids with convex edges," *Int. J. Numer. Methods Eng.*, vol. 38, no. 19, pp. 3335–3359, 1995.
[4] T. Blacker, "The Cooper Tool," in *5th International Meshing Roundtable*, 1996, pp. 1–17.
[5] M. L. Staten, S. A. Canann, and S. J. Owen, "BMSweep: locating interior nodes during sweeping," *Eng. Comput.*, vol. 15, no. 3, pp. 212–218, 1999.
[6] D. R. White, S. Saigal, and S. J. Owen, "CCSweep: Automatic decomposition of multi-sweep volumes," *Eng. Comput.*, vol. 20, no. 3, pp. 222–236, 2004.
[7] Y. Li, W. Wang, and S. Field, "All-Hex Meshing using Singularity-Restricted Field," 2011.
[8] S. Cai and T. J. Tautges, "One-to-one sweeping based on harmonic ST mappings of facet meshes and their cages," *Eng. Comput.*, vol. 31, no. 3, pp. 439–452, 2015.
[9] N. Kowalski, F. Ledoux, and P. Frey, "Smoothness driven frame field generation for hexahedral meshing," *Comput. Des.*, vol. 72, pp. 65–77, 2016.
[10] S. J. Owen, B. W. Clark, D. J. Melander, M. Brewer, J. F. Shepherd, K. Merkley, C. Ernst, and R. Morris, "An immersive topology environment for meshing," in *Proceedings of the 16th International Meshing Roundtable*, 2008, pp. 553–577.
[11] Y. Lu, R. Gadh, and T. J. Tautges, "Feature based hex meshing methodology: feature recognition and volume decomposition," *Comput. Des.*, vol. 33, no. 3, pp. 221–232, 2001.
[12] J. H.-C. Lu, I. Song, W. R. Quadros, and K. Shimada, "Pen-based user interface for geometric decomposition for hexahedral mesh generation," in *Proceedings of the 19th International Meshing Roundtable*, 2010, pp. 263–278.
[13] H. Wu and S. Gao, "Automatic Swept Volume Decomposition based on Sweep Directions Extraction for Hexahedral Meshing," *Procedia Eng.*, vol. 82, pp. 136–148, 2014.
[14] T. T. Robinson, C. G. Armstrong, and R. Fairey, "Automated mixed dimensional modelling from 2D and 3D CAD models," *Finite Elem. Anal. Des.*, vol. 47, no. 2, pp. 151–165, 2011.
[15] J. E. Makem, C. G. Armstrong, and T. T. Robinson, "Automatic decomposition and efficient semi-structured meshing of complex solids," *Eng. Comput.*, vol. 30, no. 3, pp. 345–361, 2012.
[16] L. Yin, X. Luo, and M. S. Shephard, "Identifying and Meshing Thin Sections of 3-d Curved Domains," in *Proceedings of the 14th International Meshing Roundtable*, 2005, pp. 33–54.
[17] M. Rezayat, "Midsurface abstraction from 3D solid models: general theory and applications," *Comput. Des.*, vol. 28, no. 11, pp. 905–915, 1996.
[18] Siemens plm software, "NX." [Online]. Available: http://www.plm.automation.siemens.com/en_us/products/nx/index.shtml.
[19] X. Roca and J. Sarrate, "An automatic and general least-squares projection procedure for sweep meshing," *Proc. 15th Int. Meshing Roundtable, IMR 2006*, pp. 487–506, 2006.
[20] G. Toussaint, "Solving geometric problems with the rotating calipers," *Ieee Melecon83*, no. May, pp. 1–8, 1983.
[21] D. R. White and S. Saigal, "Improved Imprint and Merge For Conformal Meshing," in *IMR*, 2002, pp. 285–295.
[22] A. Johnson, "Clipper Library." [Online]. Available: http://www.angusj.com/delphi/clipper/documentation/Docs/Overview/_Body.htm.
[23] C. G. Armstrong, H. J. Fogg, C. M. Tierney, and T. T. Robinson, "Common themes in multi-block structured quad/hex mesh generation," *Procedia Eng.*, vol. 124, pp. 70–82, 2015.
[24] "CRESCENDO,Collaborative and robust engineering using simulation capabilit enabling next design optimisation." [Online]. Available: http://www.crescendo-fp7.eu/.
[25] D. C. Nolan, C. M. Tierney, C. G. Armstrong, T. T. Robinson, and J. E. Makem, "Automatic dimensional reduction and meshing of stiffened thin-wall structures," *Eng. Comput.*, pp. 1–13, 2013.
[26] C. M. Tierney, L. Sun, T. T. Robinson, and C. G. Armstrong, "Generating analysis topology using virtual topology operators," *Procedia Eng.*, pp. 1–13, 2015.