Decomposing complex thin-walled CAD models for hexahedral-dominant meshing

Liang Sun, Christopher M Tierney, Cecil G Armstrong, Trevor T Robinson[1]

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 hexahedral (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 potential thin-sheet regions. Careful consideration is given to the mapping, merging and intersection of face pairs to generate topologies suitable for sweep meshing the thin-sheet regions, and for treating the junctions between adjacent thin-sheet regions. It is proposed that hex meshes be applied to thin-sheet regions by quad meshing one of the faces bounding the thin-sheet region and sweeping it through the thickness to create hex elements. Decisions on the generation and positioning of the cutting surfaces required to isolate thin-sheet regions are made by considering the likely impact on the quality of the resulting mesh. The method delivers a substantial step towards automatic hex meshing for complex thin-walled geometries. A significant reduction of the degrees of freedom (DOF) can be achieved by applying anisotropic hex elements to the identified thin-sheet regions.

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

1. Introduction

Thin-walled structures are common in the aviation and automotive industries. They can be efficiently analyzed by using shell elements to mesh the mid-surface representation of the structure [1]. However, shell elements are not capable of modelling the detailed behavior around local features like bosses or blends. They are also unable to predict the stress variation at the intersection of two shell structures, due to the assumption of zero stress in the normal direction in shell elements [2]. 3D-2D mixed dimensional models, where the thin-sheet regions are dimensionally reduced to mid-surfaces and the other regions remain 3D, have the ability to model the complex features and still reduce the overall number of degrees of freedom in the model relative to a 3D mesh [3]. However, careful treatment of 3D to 2D coupling is required to avoid a loss of the local accuracy around the coupling regions. The 3D-2D mixed-dimensional model is also not suitable to simulate the thermal growth in a thermal analysis, which usually requires 3D elements. Such idealized models are normally used in the preliminary design stage. In the final stages of design, components are meshed with 3D solid elements to achieve more accurate simulation results. Although computational capabilities are continuously increasing, there is an ongoing challenge to reduce the number of DOF in the analysis models in an attempt to conduct the analysis within reasonable time scales. This is especially important in large-scale nonlinear dynamic analysis, such as bird-strike or fan-blade-off at a whole aero engine level, since the number DOF in these analyses can reach tens or hundreds of millions.

Hex elements are often preferred to tetrahedral (tet) elements when analysis size is critical. 4-10 times more tet elements are typically required to mesh the same domain compared to hex elements [4], which means a huge increase in the number of DOF. In addition, to achieve accurate analysis results tet elements are required to be of similar size in all directions to avoid large element distortion, which results in many elements being used when meshing thin-walled structures due to the small size in the thickness direction. In contrast, anisotropic hex elements can bring a significant reduction in the number of DOF whilst maintaining element quality. Finally, in nonlinear structural dynamic analysis (e.g. using LS-Dyna), the explicit integration method is normally used, especially for highly nonlinear events. In the explicit integration method, the maximum allowable time step is dependent on the characteristic length of the smallest element with the poorest quality. The geometric properties of hex elements mean that meshes comprised of hex elements allow for larger time steps in comparison to meshes comprised of tet elements having a similar number DOF.

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

Fully automatic methods for hex meshing solids, whilst adhering to strict element quality metrics, have been under investigation for many years. Although significant research has been carried out [5]–[13], the complexity of the models that can be automatically hex 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 involves intensive effort and 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 [14]. Since the total removal of user intervention is not currently possible, this research is focused on an incremental approach to hex meshing, by automatically identifying and partitioning out the thin-sheet regions from the remainder of the model. The target geometries for this decomposition are complex thin-walled structures. Hex elements with good quality can be obtained by sweeping a quad mesh created on one of the bounding faces of the thin-sheet region through the region thickness. Non-thin sheet regions can be manually decomposed to create a full hex mesh, or meshed with the tet elements to create a hybrid hex-tet mesh.

The remainder of this paper is organized as follows: in Section 2, the related work about automatic decomposition for hex meshing is reviewed. The automatic thin-sheet identification process is discussed in detail in Section 3. Some alternative decomposition strategies related to the thin-sheet identification are introduced in Section 4. The thin-sheet decomposition results are demonstrated with several industrial models in Section 5. The related discussions are made in Section 6 and the conclusions are drawn in Section 7.

2. Related work

There has been much research on automatic volume decomposition for hex meshing. Price and Armstrong [7] used the 3D medial object (MO), to decompose a 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. [13] proposed an algorithm using the smoothed 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.

Other research focused on the identification of sweep-able volumes. White et al. [10] proposed a method to decompose a multi-sweep volume into many-to-one sweep volumes. The method starts by discretizing the linking faces with quadrilateral elements, which provide a layering system. The source faces are imprinted through the layers until the target faces are reached. This method works only for simple sweep-able volumes with all linking faces existing. Lu et al. [15] 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 guide for decomposition but cannot be guaranteed to result in the best decomposition. Lu et al. [16] 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. [17] 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. [3] proposed a decomposition process based on the MO for idealizing thin-sheet regions with shell elements. The 3D MO of the geometry is first generated to decide the local thickness, followed by the creation of the 2D MO on each 3D MO face to approximately indicate the lateral width for medial faces which may represent a thin-sheet. Regions with an aspect ratio (lateral width/ local thickness) exceeding the specified value are identified as thin-sheets, with these regions of the model being isolated and replaced with shell elements. Robinson's work on identifying thin-sheet regions was extended by Makem et al. [18] to find the long-slender regions in the solid bodies left behind after the thin-sheet extraction using a series of sizing measures. The notion of finding and exploiting long-slender regions was further enhanced by Sun et al. for the purposes of hex mesh generation [19]. Whilst Robinson focused on generating decompositions for stiffened shell models, Makem et al. utilized the thin decomposition to generate mixed solid element meshes by applying beam elements to the resulting long-slender regions. The decomposition produced by Robinson isn't suited to the hex meshing purposes sought in this paper as the partitioning boundaries of all thin-sheet regions were offset to accurately capture stresses at the junctions. Also, there is no treatment for very small angles in the decomposition bodies which are inappropriate for hex mesh generation as they will result in highly distorted elements. In this work offsets are only performed in situations where they are required to maintain the quality of the hex elements produced in the thin-sheet region, maximizing the volume to which the elements are applied. Yin et al. [20] proposed a method to isolate thin regions using points on an approximate MO computed by an

Octree-based algorithm. Surface triangles on opposite faces of thin regions are associated with MO points. Full 3D p-version finite elements with the low polynomial order through the thickness are applied on the thin domains. MO based methods are restricted by the lack of a robust and efficient implementation of 3D MO and one of the main drawbacks is that 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 significant effort. This time is non-value added for this application as these regions have no contribution to the thin-sheet identification. Makem et al. [18] state that the majority of the process time for decomposition is attributed to the calculation of the MO.

In this paper, a method is developed to identify thin-sheet regions based on pairs of faces bounding different regions of the model. The face pair technology provides the necessary information to perform the thin-sheet decomposition, but in a fraction of the time compared to [3] and [18]. When face pairing technology is used to abstract thin-sheets as mid-surfaces for the purpose of dimensional reduction the issue is to generate surface patches and to connect them (by extending or trimming) to get a mid-surface topology which is equivalent to the 3D model. Where the thin-sheet regions are being used to guide hex meshing, the key issue is how to decide the boundaries of the thin-sheet regions, as well as how to create appropriate cutting faces to decompose the models without generating sliver volumes in either thin-sheets or the residual domain, since these would compromise overall mesh quality.

3. Thin-sheet identification

3.1. Thin-sheet face pairs identification

The thin-sheet identification starts with the initial identification of face pairs [21]. In this paper, the face pairs are obtained from Siemens NX [22] 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 α_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 needs to satisfy

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

  where R is the aspect ratio value input by 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 [21] 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 [23]. Given a face S, its continuous boundaries ∂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 $P_i = p_1, p_2 \ldots p_m$. 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

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

  The pseudo-normal vector $\vec{n}$ is the normal of the planar domain to be used, as given by

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

The projection $\overline{p_j}$ of the point p_j on the face with the normal $\vec{n}$ is

$$\overline{p_j} = p_j - (p_j \cdot \vec{n})\vec{n}. \tag{5}$$

To use (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 [24].

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

- **Side face**: a face in a face pair, shaded grey in Fig. 1. 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 (a) and (b).
- **Wall face**: a face linking the side-1 and side-2 faces, Fig. 1 (c).

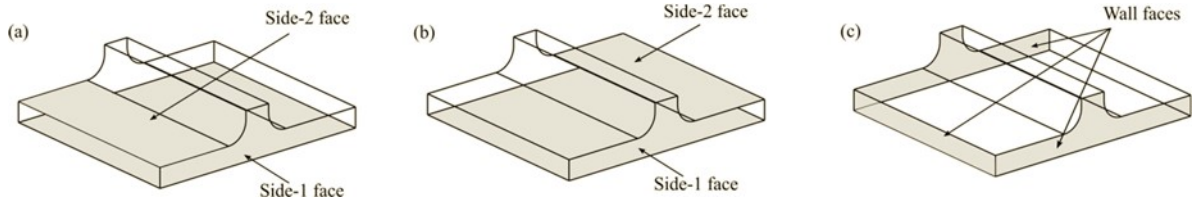

Fig. 1 (a) face pair 1; (b) face pair 2; (c) wall faces for face pair 1

3.2. Matching topologies for the thin-sheet region

3.2.1. Overview

The objective of thin-sheet identification is to obtain matching topologies on both the source and target faces of the thin-sheet region. 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, Fig. 2 (b). The discretization size is selected using a combination of the desired target element size for the final hex mesh and the boundary size, 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 described in more detail below.

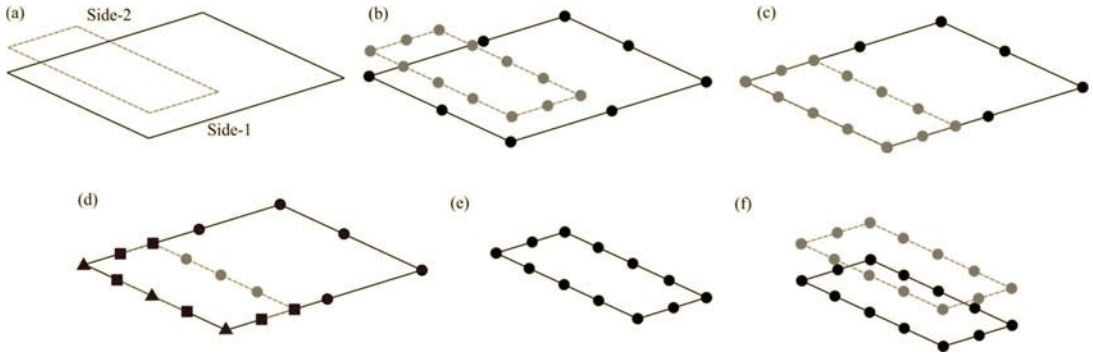

Fig. 2 Generating matching topologies: (a) side-1 and side-2 boundary from Fig 1(a); (b) discretization of boundaries; (c) projection of side-2 discretization to side-1; (d) merging; (e) 2D intersection; (f) 2D-3D mapping.

3.2.2. Projection

Projecting the polygonal boundary representation of one side face onto the underlying surface of the opposite side face allows the matching topologies to be generated using simple merge and intersection operations on the polygons. 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).

The 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} created 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. If the projection lies outside the surface boundary of side-1, the projection of b in Fig. 3 (a), the distance d between point b and b' is calculated. If d is less than,

$$d_{cri} = \sqrt{D^2 + {T_M}^2}, \tag{6}$$

the closest point on the side-1 boundary is used (b' in this case). In (6) D is the distance between point b' and the side-2 face and T_M is the merging tolerance (see section 3.2.3). If the distance between the projected points is greater than d_{cri}, Fig. 3 (b), then side-1 point b' is selected and is projected onto side-2. The resulting point, b", will become part of the boundary of the thin-sheet on side-2. A coefficient P is introduced into d_{cri}, equation (7) where α is the angle between the side faces, if the sides are not exactly parallel as shown in Fig. 3 (c).

$$d_{cri} = \sqrt{D^2 + P \times {T_M}^2}, P = (1 + \frac{D \times tan\alpha}{T_M})^2 \tag{7}$$

This projection criterion ensures sweep-able topologies of acceptable quality are generated throughout the process, avoiding the creation of badly skewed hex elements when the regions are meshed.

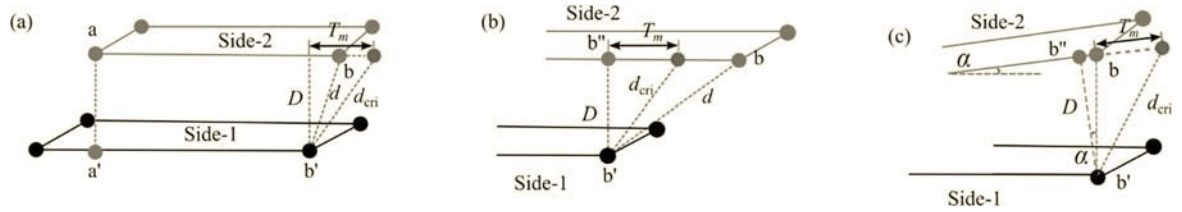

Fig. 3 Projection tolerance: (a) in tolerance; (b) out of tolerance; (c) d_{cri} when the sides are not parallel

3.2.3. Merging

To ensure good element quality in the downstream hex mesh, it is necessary to avoid the creation of sliver regions when partitioning the geometry by merging the boundaries which may cause slivers to occur. Two operations are used in this work to identify entities in proximity, namely point-to-point and point-to-line merging. These operations are performed on the 3D side-1 and side-2_{proj} polygons. Fig. 2 (d) shows the point merging operations where the triangle markers represent close points identified in the point-to-point operation and the square markers 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 [25]. Normally the size of the merging tolerance T_M is set to be small relative to the discretization 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 before the point-to-line operation to guarantee points are matched where appropriate. These are the points identified with a triangular marker in the polygon side-2_{proj} in Fig. 4 (b).

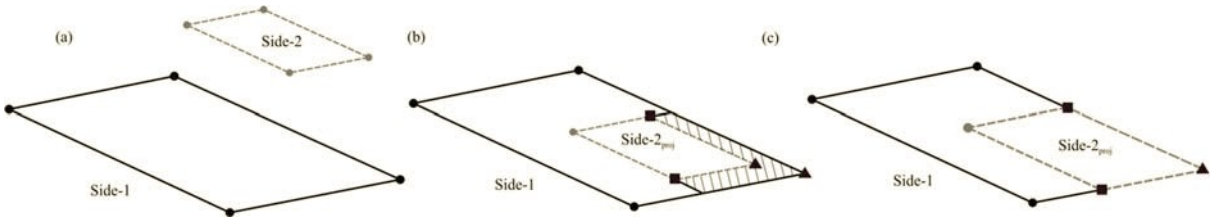

Fig. 4 Merging close points: (a) side faces; (b) side-2_{proj} points: the triangle markers represent close points identified in the point-to-point operation and the square markers represent close points in the point-to-line operation. The hatched area represents that a sliver face will be created without merging. (c) side-2_{proj} points are replaced by their close points in the side-1 polygon boundary

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 calculation of the closest point on a line segment to a given point is straightforward. There are three situations depending on the projection position of the point on the line segment. Let A be an arbitrary point, BC a line segment and D the projection of A on BC, Fig. 5 (c). Note that if D lies outside BC, the closest point of A on BC is the boundary point B or C, Fig. 5 (a) and (b) and this will be identified in the point-to-point category of.

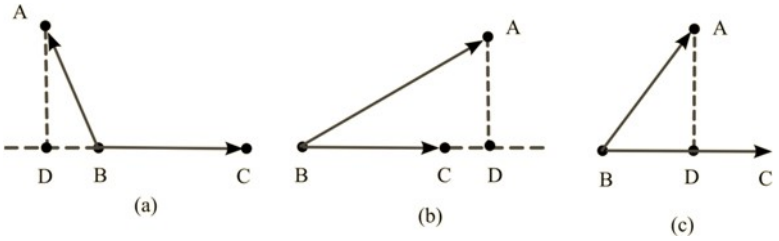

Fig. 5 Calculation of the closest point on a line segment. The projection of A on BC lies: (a) outside BC with t<0; (b) outside BC with t>1; (c) on BC with $t \in (0,1)$

The parameter of the closest point D on BC can be calculated as

$$t = \frac{(A-B)\cdot(C-B)}{(C-B)\cdot(C-B)}, \tag{8}$$

and the coordinate of D is

$$D = B + t(C-B). \tag{9}$$

The point-to-line operation is carried out for any residual points not identified in the point-to-point merge operation, i.e. all points in Fig. 4 (b) except those indicated by triangle markers. Points on both side-1 and side-2_{proj} polygons are interrogated against the line segments on the opposing polygonal boundary.

Any side-1 polygon point that is identified as being close to a side-2_{proj} polygon point, will replace it in the side-2_{proj} boundary definition, as shown by the triangle markers in Fig. 4 (c). Points close to line segments are replaced with their projected points on the line segments, as shown by the points indicated by square markers in Fig. 4 (c). Modified polygonal boundaries are generated by this merging process, and the relationships between both representations are maintained throughout.

This merging step is used to enable the polygon intersection to be performed and also to avoid the creation of sliver faces after the topology matching, e.g. the hatched area in Fig. 4 (b), or sliver volumes after the thin-sheet decomposition, Fig. 6 (b), where sliver entities are smaller than the target element size. Note that the 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 merge operations, Fig. 4 (c). Only entities considered as slivers are removed in this process, ensuring any mesh element distortion through the thickness of the sweep-able thin-sheet region is minimal.

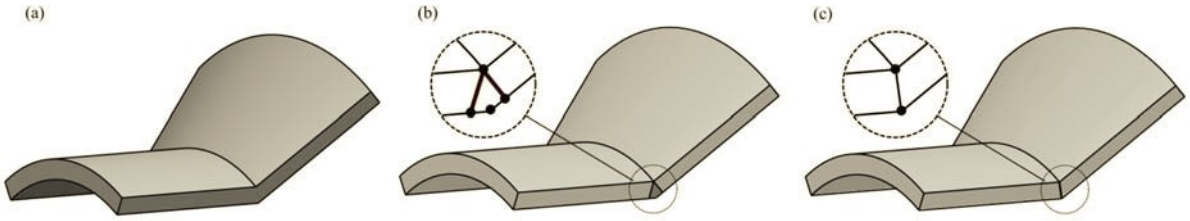

Fig. 6 Merging to avoid sliver volumes: (a) original volume; (b) decomposition without point merging – sliver volume generated; (c) decomposition with point merging - no sliver volumes.

3.2.4. Polygon intersection

At this stage, there may be more than one polygon boundary existing on the side-1 face, each representing different features on either of the side faces. The polygon intersection creates the matching topologies for the source and target surfaces of the thin-sheet region. To achieve the intersection of all the polygon loops on each side, the modified polygonal boundaries of side-2_{proj} and side-1 are represented in the parameter space of the side-1 surface in which they are contained. 2D polygonal representations of the modified 3D side-2_{proj} and side-1 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 [26], which is robust and works for both convex and concave polygons.

The intersection operation is depicted in Fig. 7 (a), where the 2D representations of side-1 and side-2_{proj} are shown in black and grey 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. 7 (b) where four separate polygons are returned. After polygon intersection, the area of each polygon $P=p_1, p_2 \ldots p_m$, $p_i = (x_i, y_i)$ is calculated using

$$A(P) = \sum_{i=1}^{n} (x_i * y_{i+1} - x_{i+1} * y_i), \quad x_{n+1=}x_1 \; and \; y_{n+1=}y_1. \tag{10}$$

If the area is positive, then the orientation will be CCW and conversely, if the area is negative, the orientation will be CW. The relative positions of polygons are achieved by taking a point of the CW polygon and comparing it with each CCW polygon to decide their relative position [27]. This is used to group the set of matching topologies, including their inner and outer loop boundaries.

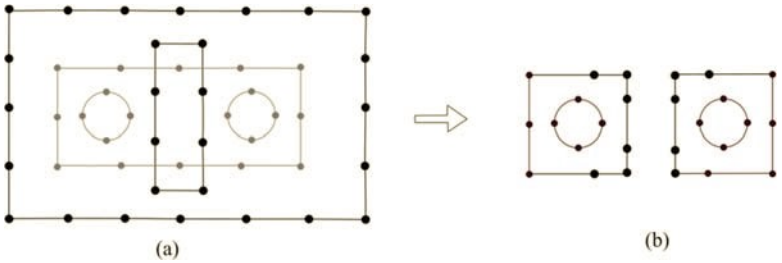

Fig. 7 2D polygon intersection: (a) polygons from both sides in different colors; (b) intersection result

An example of the necessity for the merge and intersection operations is shown in Fig. 8. In scenarios where the boundaries of features on the side-1 and side-2 faces are close when projected to the opposite side face, such as the bosses in Fig. 8 (a) and (b), if no allowance is made for the fact they are in proximity then a small sliver volume will be created in the model, between the features, as shown in Fig. 8 (c). The mesh required for such a region will have to consist of very small elements. By including the merge operation a more suitable partitioning strategy is defined, Fig. 8 (d), where the two bosses and the small volume between them are merged for the partitioning, avoiding the need for the sliver region.

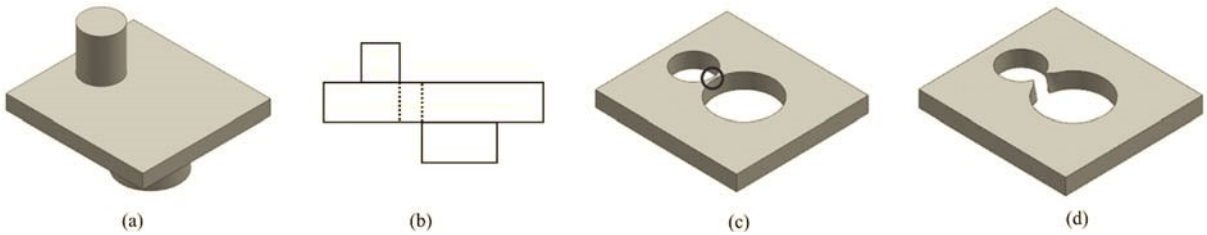

Fig. 8 Decomposition of close entities; (a) model with imprint from opposite side; (b) section view; (c) decomposition with no merge; (d) decomposition with merge

3.2.5. 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 only side-1 or side-2 is projected to the opposite side face.
- 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.

3.3. Identify and isolate thin-sheets

3.3.1. Construct side edges and wall edges

To identify or create the cutting faces which will be used to partition the model, the bounding edges for these faces are first generated on the side faces. These are based on the boundary points calculated in the last section and 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, e.g. the square markers shown in Fig. 9 (a). Given the point set p and the vertex indices set I, the side edge E_j can be constructed from one vertex to the next as

$$p = \{p_i\ , i \in [0, n-1]\},\ I = \{I_j, j \in [0, m-1]\}, E = \{\ E_j = \left[p_{I_j}, p_{I_{j+1}}\right], j \in [0, m-1], I_m = I_0\ \} \tag{11}$$

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 a curve to the defining points on 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 as solid black lines in Fig. 9 (b)) and the other one created by fitting a curve on the surface (shown as dashed black lines in Fig. 9 (b)). The relationship between a side edge and its corresponding side edge on the opposite side face is defined during the merging and intersection procedure. They are referred to as a side edge pair, Fig. 9 (b). From this information edges connecting the matched vertices of the paired side edges are defined, which are referred to as wall edges. Like the side edges, 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 as solid black lines in Fig. 9 (c)) and two new wall edges created from fitting a curve on the surface (shown as dashed black lines in Fig. 9 (c)).

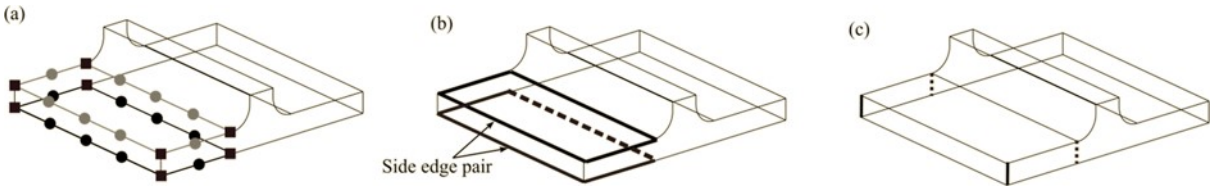

Fig. 9 For face pair 1 in Fig. 1: (a) boundary points of thin-sheet, the vertex points are represented by the squares; (b) side edges: 1 fitted edge shown as the dashed black line, 7 existing edges shown as the solid black lines; (c) wall edges: 2 fitted edges shown as the dashed black lines, 2 existing edges shown as the solid black lines

3.3.2. 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. 9, one of which is a new face (to the interior of the component) and the other three are from existing wall faces. To guarantee the quality of mesh 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 represented by a new face will be either used as is, or will be offset into the thin-sheet region to ensure a good quality mesh can be applied to the region remaining after the decomposition. The model in Fig. 10 (a) has two candidate cutting faces which are new faces, as highlighted. The left cutting face can be used as is. An offset is required for the right cutting face, Fig. 10 (b), otherwise the partitioning would generate a sub-volume with a small dihedral angle at which poor quality hex elements would be inserted.

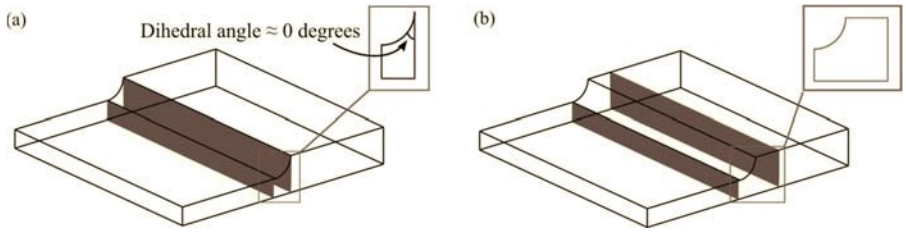

Fig. 10 (a) result by using the candidate cutting faces; (b) result by offsetting the right-hand side candidate cutting face which is from a new face

Where the candidate cutting face is an existing wall face, it will not be used to partition the model (as no partitioning is required). For example, for the face pair in Fig. 9, the three cutting faces at the existing wall faces are not necessary for the decomposition and only the new face will be used to split the model. However, for thin-sheet regions with more complex end conditions, a determination is required as to whether a new cutting face is required to partition the model, bounded by the same side edges as the existing wall face, or by offsetting the current side edges. For example, a decomposed thin-sheet and the hex mesh generated by sweeping quad elements through the thickness are shown in Fig. 11 (a). Note that the sweeping algorithm generates structured quadrilateral meshes on the wall faces of the domain, i.e. every internal node on the face has 4 incident elements. However, if the same model had a curved end on the left, with the same cutting faces identified, the decomposed thin-sheet and the hex mesh will be as shown in Fig. 11 (b). It still generates a structured quadrilateral mesh but the quality of some mesh elements 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. Mesh singularities are where more or less than 4 quad elements meet at the internal nodes on a face (positive singularities if >4 elements and negative singularities if <4 elements). Armstrong et al [28] proposed an equation to calculate the net number of the mesh singularities on a face as

$$\sum(\frac{\pi}{2}N-\theta)+\sum\int k_g ds+\iint KdA+(n_+-n_-)\frac{\pi}{2}=0. \quad (12)$$

In the equation, θ is the corner angle and N is the optimum number of elements at a vertex calculated using

$$N = round\left(\frac{\theta}{\pi/2}\right), \tag{13}$$

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. 11 (c) is comprised of a half circle and straight lines. For this shape, the first term in equation (12) is zero since all the angles are a multiple of π/2; the second item is

$$\sum \int k_g \mathrm{d}s = \int \frac{1}{R}\mathrm{d}s = \frac{1}{R} \times \pi R = \pi, \tag{14}$$

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

$$0 + \pi + 0 + (n_+ - n_-)\frac{\pi}{2} = 0 \tag{15}$$

$$n_+ - n_- = -2 \tag{16}$$

meaning two negative mesh singularities are required to generate a good quality quad mesh for this shape, as shown in Fig. 11 (c). Therefore, a new cutting face is needed to isolate the regions in which the singularities exist, and further decompose the model into two volumes, as shown in Fig. 11 (d). The right-hand side one will be a thin-sheet and swept meshed. The left-hand side one will be left to mesh with other hex meshing algorithms, or filled with tet elements.

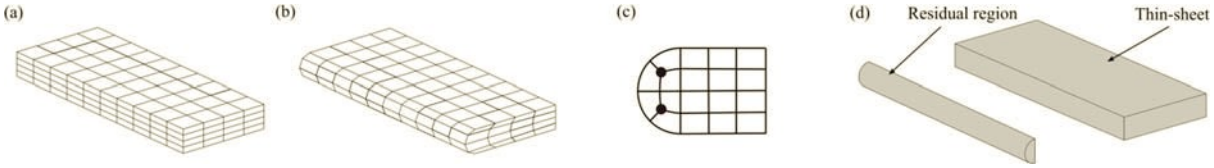

Fig. 11 (a) thin-sheet and hex mesh for the face pair in Fig. 9; (b) thin-sheet and hex mesh if the model in Fig. 9 has a curved end; (c) example of the mesh singularity for a wall face in (b); (d) desired decomposition result

Equation (12) gives the net number of the mesh singularities required in a face, but the position of these singularities is not known. Here a method for determining whether and by how much to offset a candidate cutting face is proposed based on the optimum number of elements required at the side edges, and the number of mesh singularities to be included. This allows the mesh singularities to be isolated in a non-thin-sheet region and ensures the introduction of the cutting face will not result in sliver regions to which poor quality elements would have to be applied. To do the assessment a slice is made at the mid-point of one side edge perpendicular to the tangent of the side edge. On the cross sections in Fig. 12, the side faces are represented by curves (shown in grey bold solid lines) and the side edges are represented by two points A and B. If the candidate cutting face is derived from an existing face it is shown in a black bold solid line, and if it is a new face it is shown in a black bold dashed line. Let N_A and N_B be the optimum number of elements calculated for the corners with the side edges.

- $N_A = 0$ or $N_B = 0$: an offset will have to be made, otherwise the quality of the quad element at the corner will be poor, Fig. 12 (a).
- $N_A = 1$ and $N_B = 1$: if the candidate cutting face is a new face, as in Fig. 12 (b), an offset must be made otherwise it will generate a volume with a dihedral angle close to zero degrees. If it is from an existing wall face, as in Fig. 12 (c), a line CD is added to form a closed four-sided shape ABCD which can be locally assessed. To be hex meshable by sweeping through the thickness the number of singularities in the enclosed shape should be zero. Using (16) it is possible to calculate there is no need for singularities in a region, but this can also indicate a region where there is a need for one or more pair of positive and negative singularities to create a well-structured hex mesh. To determine if this is the case sample points are created along AB and CD to create element layers, e.g. E, G and make the curve connecting them parallel to AD. The number of singularities in each sub-region, e.g. AEGC, is calculated. If it is calculated that any sub-region requires a mesh singularity then the candidate cutting face is offset. In Fig. 12 (b), AE is a quarter circle and θ_E=0, therefore it is calculated that there should be a positive singularity in AEGD, so an offset is made, Fig. 12 (d). If there are no singularities in any sub-region, the candidate cutting face is not used and no other faces are needed, Fig. 12 (e).

- $N_A = 1$, $N_B > 1$ or $N_B = 1$, $N_A > 1$: the candidate face will be offset, Fig. 12 (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 to partition the model, Fig. 12 (g). If the new face intersects with the candidate cutting face, Fig. 12 (h), or the minimum distance between them is less than the target element size, Fig. 12 (i), then the cutting face will be created by offsetting the candidate cutting face. If the candidate cutting face is a new face, Fig. 12 (j), similar checks are carried out between sample points on curves between A and B and the candidate cutting face. For the example in Fig. 12 (j), the candidate cutting face will be used.

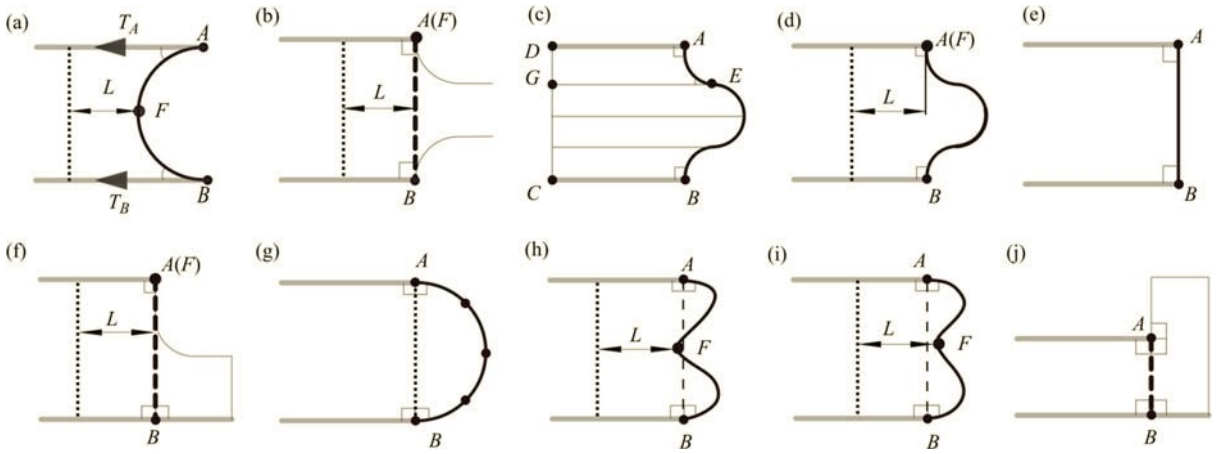

Fig. 12 Different treatments of the candidate cutting faces

Once all the candidate cutting faces have been identified in the model, and a decision made about which faces to offset, the CAD system is used to partition the model. Generating the offset cutting surfaces requires offsetting its bounding side edge in the side face. To guarantee the offset curve will correctly intersect the boundary of the thin-sheet region, an initial decomposition is performed using closed loops of candidate cutting faces to partition the model. The offset curves are then generated in the decomposed thin-sheet regions using the 'offset curve in face' function in NX. This guarantees the correct thin-sheet boundary intersections, and creation of the correct offset surface. The offset is always made in the direction to make the thin-sheet region smaller, as for T_A and T_B in Fig. 12 (a). The point on the intersection curves between A and B is identified which has the maximum projected distance along the offset direction T_A, (e.g. point F shown in Fig. 12 (a)). The offset is made so that the minimum distance L between this point and the final cutting face is one target element length. After the 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 creating a surface bounded by the paired side edges and wall edges. A subsequent partitioning operation is carried out using these offset cutting faces. If the offset fails the region will not be identified as being a thin-sheet region. While this will reduce the amount of the model which is identified as being thin-sheet, this will avoid the generation of sliver regions and ensure better mesh quality overall.

Fig. 13 shows the offsetting procedure for a complex component. Fig. 13 (a) (b) shows the decomposition with no offset performed. Here E_3, which is the direct imprint of E_1, can't be merged with E_2 under the selected merging tolerance and a sliver region is generated. However, E_1 is an existing edge at which it has been calculated that one element is required and at E_3 it has been calculated that two elements are required. This corresponds to the third case in the classification and so an offset is then required. E_1, E_3 and the edge chains that are tangent to them (or angle between them within a defined range) are then offset along the corresponding side faces, Fig. 13 (c). The decomposition result after the offset is shown in Fig. 13 (d).

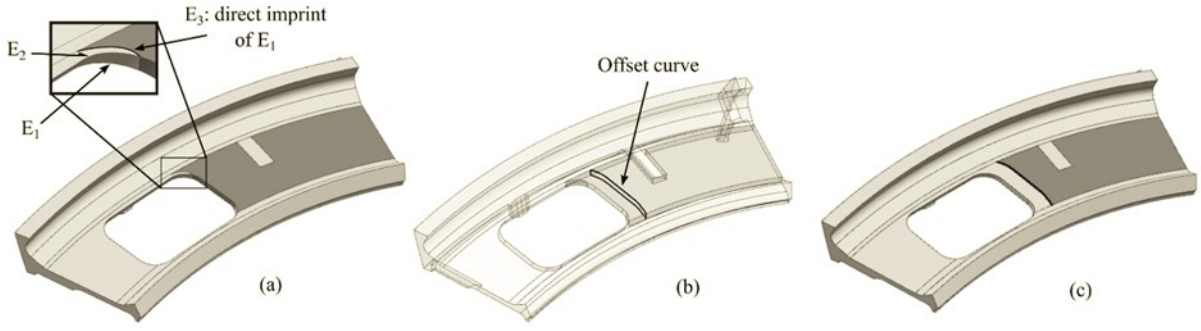

Fig. 13 Side face offset to avoid small element angles: (a) decomposition with no offset; (b) offset the side edges and create wall edges joining them; (c) decomposition with offset. Thin regions are shown in dark and residual regions in light

It should be noted that if the model contains a step (sharp transition in section thickness, e.g. Fig. 14 (a)), but the regions on either side of the transition are thin-sheets in their own right, then using this approach an offset will be inserted at the interface for the face pair on the thicker side, Fig. 14 (b). This is because the interface at the thicker side is equivalent to that shown in Fig. 12(f). While this reduces the overall volume identified as thin-sheet, it is shown in other work that the residual region can be identified as a long-slender region [19] or sweep-able volume [29], in which a high quality mesh can be produced using these other approaches.

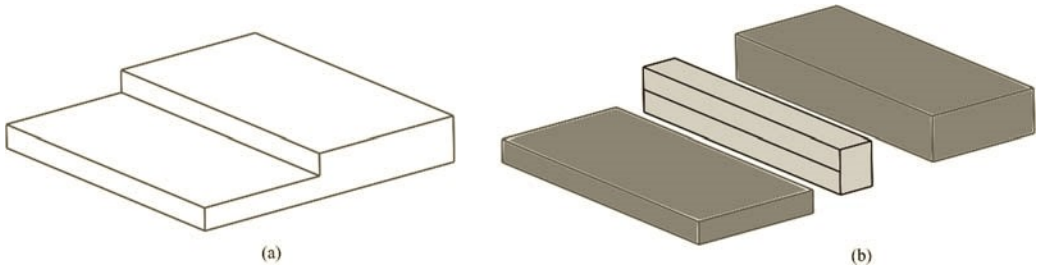

Fig. 14 Staircase geometry example: (a) Staircase geometry with a sharp change in thickness; (b) thin-sheet decomposition: offset made for the face pair at the higher level. Thin regions are shown in dark and residual regions in light

3.3.3. Geometrical decomposition

Although individual side faces in the face pair satisfy the distance criteria, the region which results after the intersection of the opposite faces in the face pair and the offsetting may no longer be considered a thin-sheet. Therefore, the aspect ratio of the target thin-sheet region is calculated again with a method similar to that introduced in the face pair filtering process. The final side edges on side-1, which define the side-1 boundaries of the target thin-sheet, are used to determine the characteristic length and width, and are compared to the thickness. The decomposition will not be performed if the aspect ratio is less than the predefined value. Before the decomposition, a test point is created by triangulating the region bounded by the final side edges on the side-1 face and identifying the centroid of the largest triangle. After the decomposition, the region that contains the test point is identified as a thin-sheet, as shown in Fig. 15.

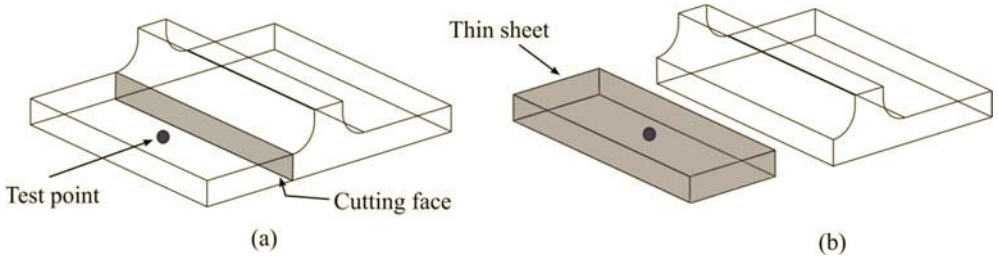

Fig. 15 Test point and thin-sheet identification after the decomposition

4. Alternative decompositions

This article has focused on the identification and isolation of thin-sheet regions in CAD generated B-rep model. This section introduces some alternative decomposition strategies for assemblies, facetted representations and for

virtual decompositions. Assembly models and facetted representations are commonplace within analysis workflows where thin-sheet decomposition is a valid and worthwhile pre-processing step that requires consideration.

To identify thin-sheet regions in an assembly model, e.g. the riveted plate assembly in Fig. 16 (a), each component can be processed separately using the method described in this work. This will create a decomposition for each component which is then meshed independently from the other components in the assembly. Contact properties can be applied between the different meshed decompositions for an analysis of the entire assembly. However, in certain applications where a conformal mesh is required between adjacent components it is necessary to perform some geometric operations at the assembly level in order to capture the assembly relationships before running the decomposition process.

By first imprinting the different components in the assembly onto each other, e.g. Fig. 16 (b) and (c) the thin-sheet decomposition can then be executed for each individual component, but will respect the features created by the imprinted lines. The resulting decomposition is shown in Fig. 16 (d) and (e) with an exploded view used to distinguish between the decomposed subsets. Four thin-sheet regions are identified, and two residual cells are produced to capture the rivet imprint. By applying the same mesh to the bounding faces of the thin-sheet where they are in contact, and sweeping it through the thickness, the hex mesh created will be conformal where they meet.

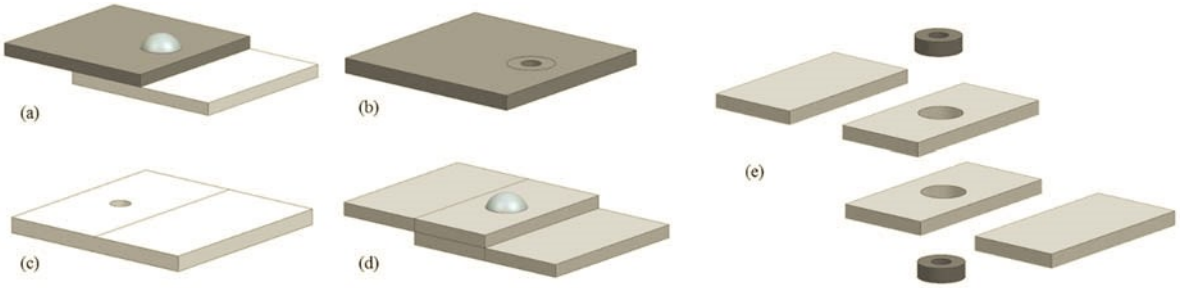

Fig. 16 Thin-sheet identification for riveted plate assembly: (a) assembly; (b) rivet imprint on top plate; (c) imprint between plates; (d) thin-sheet decomposition; (e) exploded view of assembly decomposition with thin-sheet regions light grey and thick region

The thin-sheet identification described here is based on the real decomposition of the geometry. Directly decomposing the geometry is often time-consuming due to the complexity of the geometric decompositions and the effort required to maintain the links between the decomposed cells. Also, the creation of sliver edges and faces is a common occurrence which is a direct result of ill-defined and temperamental tolerance issues between target and tool geometries. An alternative decomposition approach is to use Virtual Topology [34]. Virtual Topology allows changes to be performed at a topological level without any modification of the underlying geometry. If the virtual topology technique is used, it is possible to achieve a virtual decomposition [30], where the underlying geometry is unchanged and the pitfalls associated with the direct geometry editing are avoided. For example, sliver regions generated from geometric tolerance issues are completely avoided as no complex surface intersections are required. Also, the links between decomposed cells are automatically maintained as the virtual decomposition is a non-manifold representation with shared faces between adjacent regions. These links can then be used to automatically define a meshing recipe.

Another application is the extension of these methods to work with discretized/facetted representations, Fig. 17 (a). Facetted representations are often used in reverse engineering workflows or even as part of an optimization workflow where the deformed mesh can be the starting point of the next analysis. Processing these representations would require collections of elements faces, or facets, to be grouped together to define a logical face and to then be paired with another logical face to facilitate the thin-sheet decomposition, Fig. 17 (b). Both the face-pair identification and imprinting operations could be achieved easily based on ray-casting the facet vertices. Imprinting would require boundary elements of a logical face, Fig. 17 (c), to be projected to the paired face, Fig. 17 (d). In such an application the boundary of the faces could be easily identified as the set of edges bounding only one facet in the face set. The process would be heavily dependent on the initial mesh density, where local mesh refinements may be required to accurately capture the boundary of the side faces. The creation of appropriate partitioning geometry from the imprinted face-pair boundaries could be performed by tracing and smoothing the boundary edges of facets in order to avoid jagged definitions that can be seen in Fig. 17 (d).

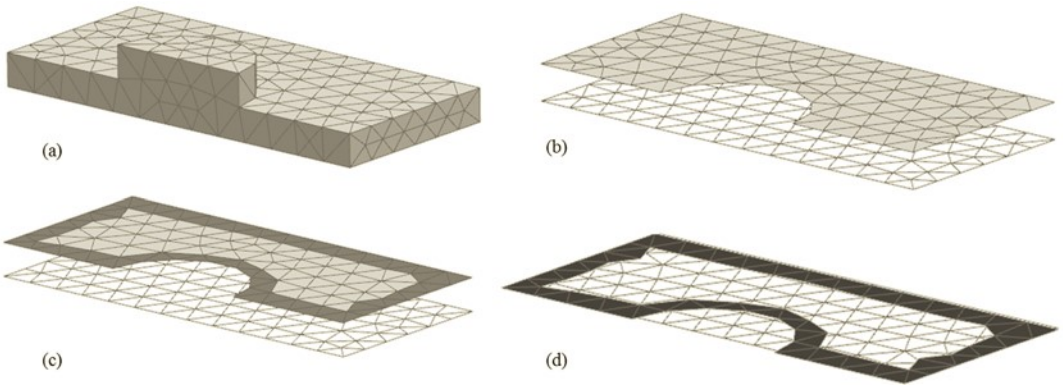

Fig. 17 Thin-sheet processing for facetted representations; (a) initial facetted model; (b) meshed side faces; (c) identification of boundary elements; (d) imprinting of boundary elements on the adjacent mesh.

5. Results

The described approach has been implemented using the C# language and .NET framework APIs in Siemens NX 10. The thin-sheet identification was carried out on a 3.4 GHz Intel Core (TM) i7-2600 CPU desktop workstation with 16GB RAM. The input is a solid model with no additional information to guide the decomposition process (e.g. mark-up or feature history). After decomposition, the thin-sheet regions are colored dark grey while the residual regions are colored light grey. Results for a selection of models are given in Fig. 18. The computation time, the number of thin-sheets and the volume occupation are summarized in Table 1.

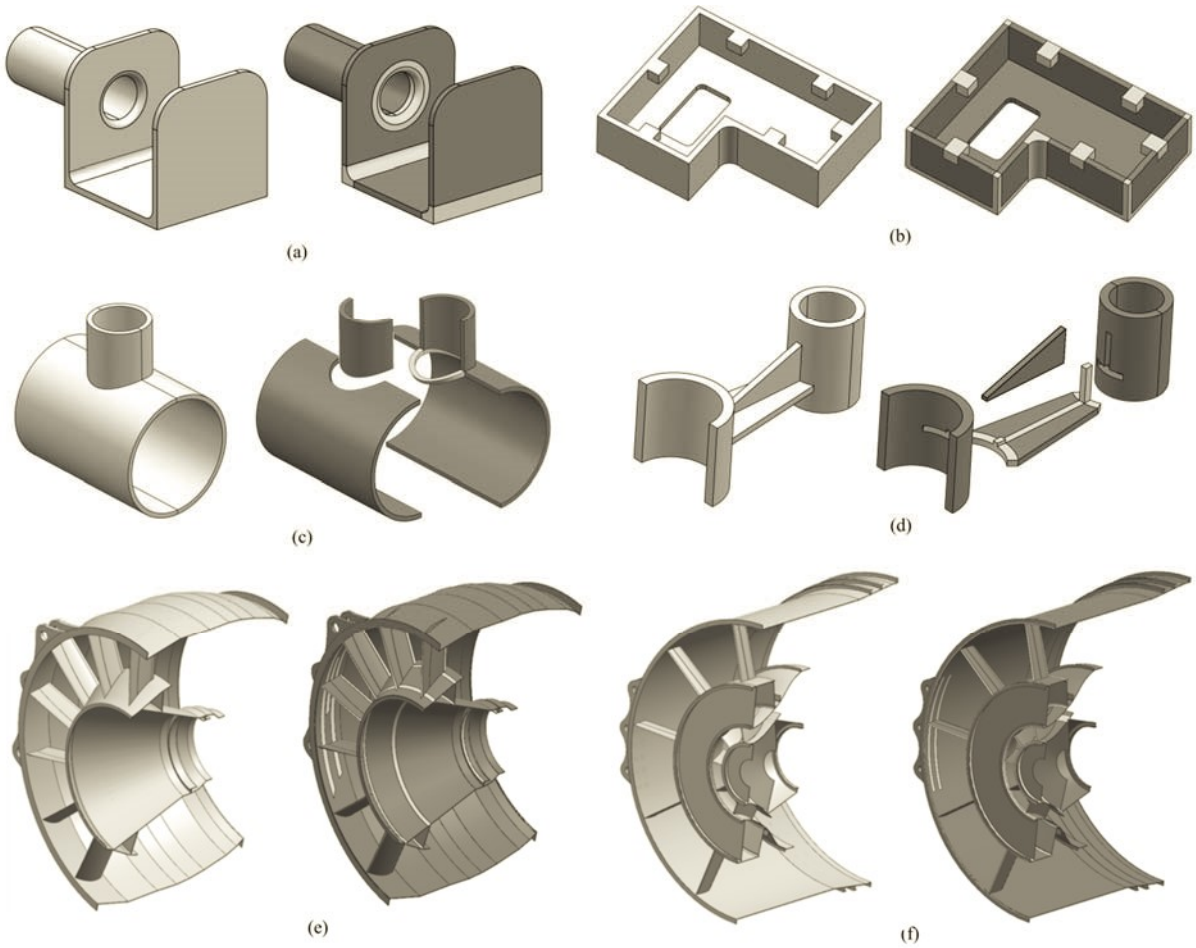

Fig. 18 Thin-sheets identification results: the left hand side is the original model and the right hand side is the decomposed model. The thin-sheet regions are colored dark grey while the residual regions are colored light grey

Table 1 Thin-sheet identification results and input parameters

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

5.1. Degree of freedom reduction

The purpose of identifying and isolating the thin-sheet regions is that a significant reduction in the number of DOF can be achieved by using anisotropic hex elements in the thin-sheet regions. A representative industrial example model, the CRESCENDO compressor inter-casing [31], is used here for demonstration. The number of DOF and the analysis accuracy are compared for a benchmark tet meshed model and the mixed elements meshed model created on the decomposition generated by this process, 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 was densely meshed with 10-node tet element as shown in Fig. 19 (a). The decomposed model is shown in Fig. 19 (b) where 46 thin regions have been identified and occupy approximately 90% of the model's volume. Two mixed element analysis models were generated for comparison. Both models use pyramid elements at the thin-sheet/residual region interface to transition between the hex elements and the tet elements. The meshes were manually generated in NX and consist of:

1. Isotropic 8-node hex elements are used to mesh the thin-sheet regions. The residual regions are meshed with 10-node tet elements, Fig. 19 (c).
2. Anisotropic 8-node hex elements with aspect ratios (lateral dimensions to thickness) varying up to 5 (much greater aspect ratios can be used depending on the analysis being performed) are used to mesh the thin-sheet regions. The residual regions are meshed with 10-node tet elements, Fig. 19 (d).

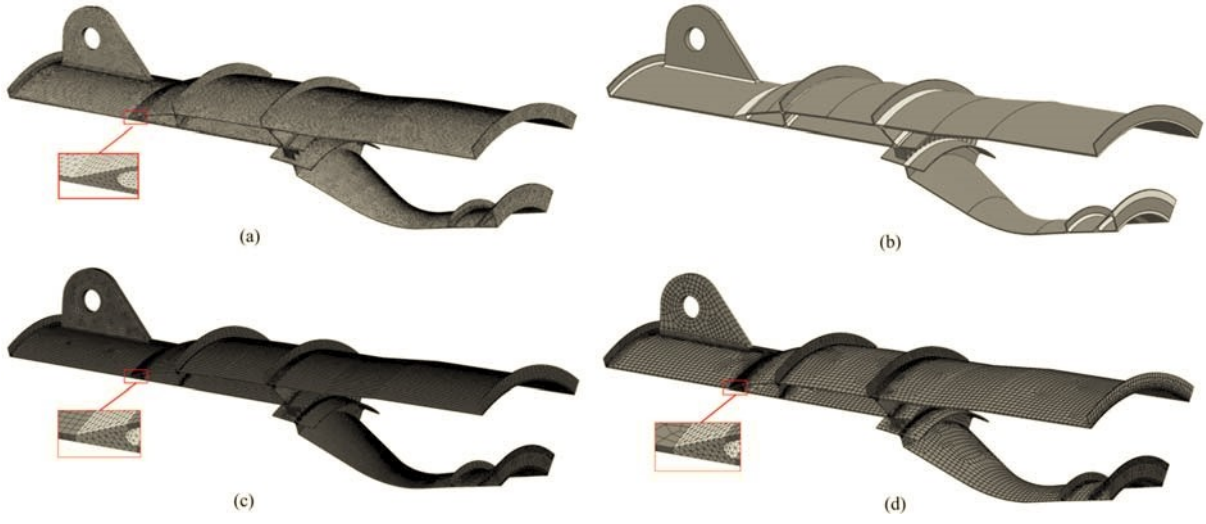

Fig. 19 (a) tet elements meshed model; (b) decomposed casing model; (c) thin-sheets meshed with isotropic hex element; (d) thin-sheets meshed with anisotropic hex element

The degrees of freedom of the three analysis models are given in Table 2. It can be seen that the number of DOF of the mixed element analysis models in Fig. 19 (c), and Fig. 19 (d) are 26.1% and 17.7% of the dense tet elements respectively.

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
DOF	2191611	572706	388803
% Tet	100	26.1	17.7

Free-free modal analyses were conducted on the models and the first 20 modal frequencies are compared. The analysis result from the dense tet element meshed model is used for reference. The analysis times for the three analyses are given in Table 3. The analysis time for the mixed element analysis models in Fig. 19 (c) and Fig. 19 (d) are 15.6% and 6.7% of the dense tet element mesh.

Table 3 The analysis times for the three analysis model

Analysis models	Tet for the whole model	Isotropic hex for thin-sheets	Anisotropic hex for thin-sheets
Time / min	45	7	3
% Tet	100	15.6	6.7

Fig. 20 compares the discrepancy in the modal frequency of the models meshed with mixed elements to the reference dense tet mesh. According to Fig. 20, 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 for this type of analysis.

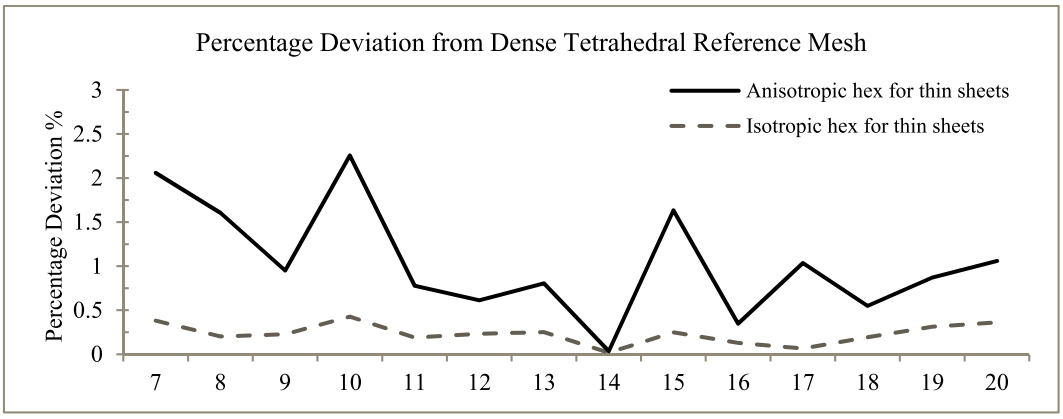

Fig. 20 Comparison of modal frequencies for the inter-casing model, the result of the dense tet elements meshed model is used as standard

6. Discussion

This paper describes an automatic process for identifying the thin-sheet regions in the CAD model of complex components models informed by the sets of face pairs bounding the thin-sheet regions. In this work, the face pairs are obtained using the mid-surface function in NX. This function is fast, but issues have been identified in the results returned, which include missing face pairs and incorrect thickness values. When these issues occur, the corresponding thin-sheet regions may not be identified. The thin-sheet identification process in this work is independent of the source of the face pairs, and an improved method for identifying face pairs may result in more thin-sheet regions might be identified.

The face pairs in NX are filtered based on the input parameters of aspect ratio R and angle tolerance α_T. Decreasing the aspect ratio or increasing the angle tolerance will result in more face pairs being considered. For dimensional reduction, a thin-sheet is defined as a region with a high aspect ratio, e.g. 10 is indicated in [1]. For sweep meshing applications, this condition can be relaxed. However, if the aspect ratio is too small, e.g. $R<3$, the computational effort required to process regions can outweigh the benefits from reducing the number of DOF as less of the overall volume can be identified as thin-sheet. For example, R for the face pair shown in Fig. 21 (a) and Fig. 21 (b) is about 1. Following the process, an offset will be made for the candidate cutting face and the final thin-sheet is shown in grey in Fig. 21 (c). The identified thin-sheet will not contribute much in terms of reducing the number of DOF in the model, but requires costly geometric operations, such as offsetting, to be performed. It may also complicate the meshing result around the region. Similar conclusions apply to the angle tolerance.

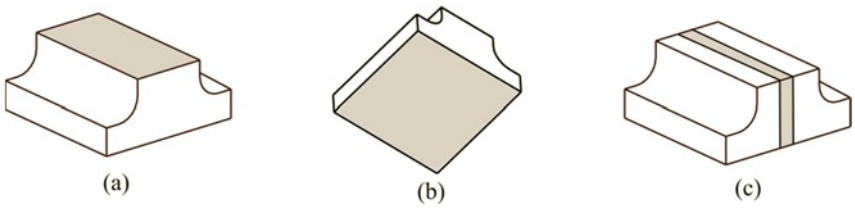

Fig. 21 (a) and (b): the face pair; (c) the identified thin-sheet shown in grey

The measures of face width and length which are used to calculate the aspect ratio R are based on a planar approximation of the original face. Different schemes can be used to represent the width/length of the planar domain, such as the minimum bounding box or the maximum inscribed circle. The minimum bounding box method normally ends up with a larger width/length value than the maximum inscribed circle method, e.g. for the c-shape model as shown in Fig. 22, and therefore a larger aspect ratio value. The appropriate method to use really depends on the application in which the mesh will ultimately be used for, and what the analyst determines the principle measures should be.

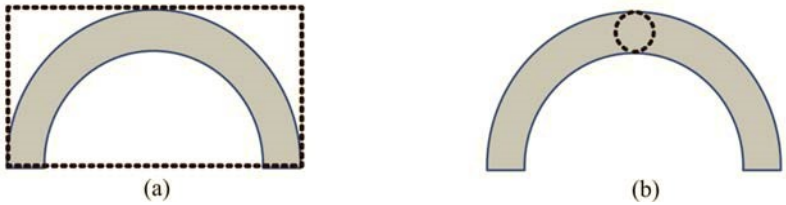

Fig. 22 Measure the face width and length of a c-shape model by: (a) the minimum bounding box; (b) the maximum inscribed circle

Two key input parameters are the discretization element size and the merging tolerance T_M. The discretization element size is used when discretising the boundary of the face to polygons. A small value will more accurately approximate the original boundary, but will increase the computational effort required for the process overall. In this work, the number of intervals along an edge is first calculated by dividing the arc length by the target element size. So, each edge of the polygon will be less than or equal to the target element size. This does not consider the curvature of the edge. Since the merging tolerance is related to the discretization element size, a small value will limit the range that the merging operation works within. The merging tolerance is introduced to avoid the generation of sliver volumes. The merging tolerance is normally greater than the scale of the modelling tolerance of the CAD package. However, in this work, it is limited to be small compared to the discretization element size. A small merging tolerance ensures that points are not identified as being close to more than one point on the opposition side, which keeps the data structure simple and robust. Otherwise, in the boundary mapping stage (see Section 3.2.5), the type-1 point will have more than one corresponding point in 3D. The order of these 3D points needs to be determined, and two type-1 points may have a common corresponding 3D point. The same conclusions are obtained in [32] where the authors justify the use of a small tolerance. It remains an item of future research to provide a robust process which is not as strict on the tolerance value used.

Section 3.2 introduces the process of obtaining the matching topologies on both sides of the thin-sheet region. The matching topologies define the candidate cutting faces. Careful consideration is given in section 3.3 to decide the final position of the cutting face. Instead of making an offset for all candidate cutting faces, as was the case in [3], the offset is only performed where it is determined that it is required. The offset is made to guarantee a good quality mesh is generated in both the thin-sheet and residual regions. Currently, this process does not consider the quality of the new corners formed after offsetting.

For this approach, as in many geometric processes, the results produced are heavily related to the scale of the geometric features and tolerance applied. In this process, the consequences of this are how the process decides to compute and apply the cutting faces in the model. For the model shown in Fig. 23, with a small merging tolerance relative to model size the split achieved would be like Fig. 23 (a). With a larger merging tolerance, e.g. $> \sqrt{2}$ thickness, the candidate cutting faces will be offset and the final result will look like Fig. 23 (b). Both are valid thin-sheet decompositions.

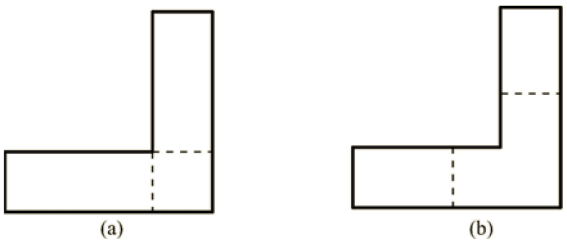

Fig. 23 (a) cutting faces when using a small merging tolerance; (b) final cutting faces after offset when using a large merging tolerance

The decomposition results in Section 4 have shown the effectiveness of the described method. For the demonstrated models, large portions (81%~99% of the overall volume) have been identified as thin-sheet regions in very short timescales. Compared to the MO based method in [3] [18], there is no need to calculate the computationally expensive 3D MO (though the MO is still in active development and will no doubt speed up in the future). It should be reiterated that time spent on computing the MO of local complex regions is non-value added for this application, as only the face pairs that bound a possible thin-sheet region are needed. For models Fig. 13 (a) and (c), the residual regions are all sweep-able. A hex mesh can be easily generated for the identified regions using sweeping algorithms. For the models in Fig. 13 (b) and (d), with some further manual decomposition, a full hex mesh can be easily generated. The models in Fig. 13 (e) and (f) are more complex industrial models with a high percentage of thin-sheet regions. In addition, it is worth considering the isolated regions containing singularities, Fig. 10 (d). It is clear the body with the curved face can also be meshed by sweeping, albeit with a sweep direction along the length of the body rather than in the thickness of the thin-sheet region. Using the approach described in [18] or [19] any long-slender regions in the non-thin sheet residual regions can be further identified. These regions can also be meshed with sweeping algorithms. At present, the final residual regions need to be manually decomposed to achieve full hex meshes, or the regions can be meshed using tet elements if a hex-dominant mesh is acceptable. Although this offers only a partial solution to automatic hex meshing, the amount of manual effort required to generate the decomposition has been significantly reduced. An automatic implementation of the above hex-dominant meshing strategy will be developed in the future.

Different mixed elements models have been created to compare the number of DOF, the analysis time and the accuracy. It is shown that for the simple modal analysis performed in this work, linear hex elements give similar accuracy to quadratic tet elements of similar dimension, but with fewer DOF and therefore a reduced analysis time. By using anisotropic hex elements (aspect ratio varying up to 5) in the thin-sheet regions, the number of DOF can be further reduced whilst retaining a reasonable accuracy.

7. Conclusions

This paper presents an approach to identify and isolate thin-sheet regions in the CAD model of a complex component. For these regions, hex meshes are generated by sweeping quad elements through the thickness of the thin-sheet region. The automatic approach brings a significant reduction in the amount of manual effort required in the hex mesh generation process. A significant reduction in the number of DOF is 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 final positioning of the cutting surfaces used to isolate the thin-sheet regions. Several industrial models have been used to demonstrate the results.

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. 13(e), Fig. 13(f) and Fig. 14 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] A. A. Vince Adams, *Building Better Products with Finite Element Analysis*. Cengage Learning, 1999.

[2] C. K. Lee and Q. X. Xu, "A new automatic adaptive 3D solid mesh generation scheme for thin-walled structures," *Int. J. Numer. Methods Eng.*, vol. 62, no. 11, pp. 1519–1558, 2005.

[3] 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.

[4] T. Blacker, “Meeting the challenge for automated conformal hexahedral meshing,” in *9th international meshing roundtable*, 2000, pp. 11–20.

[5] W. A. Cook and W. R. Oakes, “Mapping methods for generating three-dimensional meshes,” *Comput. Mech. Eng.*, vol. 8, pp. 67–72, 1982.

[6] D. White and L. Mingwu, “Automated hexahedral mesh generation by virtual decomposition,” in *4th International Meshing Roundtable*, 1995, pp. 165–176.

[7] 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.

[8] T. Blacker, “The Cooper Tool,” in *5th International Meshing Roundtable*, 1996, pp. 1–17.

[9] 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.

[10] 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.

[11] Y. Li, W. Wang, and S. Field, “All-Hex Meshing using Singularity-Restricted Field,” 2011.

[12] 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.

[13] N. Kowalski, F. Ledoux, and P. Frey, “Smoothness driven frame field generation for hexahedral meshing,” *Comput. Des.*, vol. 72, pp. 65–77, 2016.

[14] S. J. Owen *et al.*, “An immersive topology environment for meshing,” in *Proceedings of the 16th International Meshing Roundtable*, 2008, pp. 553–577.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] L. Sun, C. Tierney, C. Armstrong, and T. Robinson, “An enhanced approach to automatic decomposition of thin-walled components for hexahedral-dominant meshing,” *Eng. Comput.*, 2017 DOI: 10.1007/s00366-017-0550-x.

[20] 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.

[21] M. Rezayat, “Midsurface abstraction from 3D solid models: general theory and applications,” *Comput. Des.*, vol. 28, no. 11, pp. 905–915, 1996.

[22] Siemens plm software, “NX.” [Online]. Available: http://www.plm.automation.siemens.com/en_us/products/nx/index.shtml.

[23] 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.

[24] G. Toussaint, “Solving geometric problems with the rotating calipers,” *Ieee Melecon83*, no. May, pp. 1–8, 1983.

[25] D. R. White and S. Saigal, “Improved Imprint and Merge For Conformal Meshing,” in *IMR*, 2002, pp. 285–295.

[26] A. Johnson, “Clipper Library.” [Online]. Available: http://www.angusj.com/delphi/clipper/documentation/Docs/Overview/_Body.htm.

[27] K. Hormann and A. Agathos, “The point in polygon problem for arbitrary polygons,” *Comput. Geom.*, vol. 20, no. 3, pp. 131–144, 2001.

[28] 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.

[29] H. J. Fogg, L. Sun, J. E. Makem, C. G. Armstrong, and T. T. Robinson, “A simple formula for quad mesh singularities,” *Procedia Eng.*, vol. 203, pp. 14–26, 2017.

[30] C. M. Tierney, L. Sun, T. T. Robinson, and C. G. Armstrong, “Using virtual topology operations to generate analysis topology,” *CAD Comput. Aided Des.*, vol. 85, pp. 154–167, 2016.

[31] “CRESCENDO,Collaborative and robust engineering using simulation capabilit enabling next design optimisation.” [Online]. Available: http://www.crescendo-fp7.eu/.

[32] D. R. White, S. Saigal, and S. J. Owen, “An imprint and merge algorithm incorporating geometric tolerances for conformal meshing of misaligned assemblies,” *Int. J. Numer. Methods Eng.*, vol. 59, no. 14, pp. 1839–1860, 2004.