A new automatic adaptive 3D solid mesh generation scheme for thin-walled structures

Lee, Chi King; Xu, Q. X.

2004

Lee, C. K., & Xu, Q. X. (2005). A new automatic adaptive 3D solid mesh generation scheme for thin-walled structures. *International Journal for Numerical Methods in Engineering*, 62(11), 1519-1558.

<https://hdl.handle.net/10356/103292>

<https://doi.org/10.1002/nme.1239>

© 2004 John Wiley & Sons. This is the author created version of a work that has been peer reviewed and accepted for publication by *International Journal for Numerical Methods in Engineering*, John Wiley & Sons. It incorporates referee's comments but changes resulting from the publishing process, such as copyediting, structural formatting, may not be reflected in this document. The published version is available at: [Article DOI: <http://dx.doi.org/10.1002/nme.1239>].

Downloaded on 26 Apr 2023 17:37:57 SGT

A new automatic adaptive 3D solid mesh generation scheme for thin-walled structures

C. K. Lee* and Q. X. Xu

*School of Civil and Environmental Engineering, Nanyang Technological University
Nanyang Avenue, Singapore, 639798, Singapore*

*E-mail: ccklee@ntu.edu.sg

Abstract

Pre-printed version of the paper appeared in the journal: *International Journal for Numerical Methods in Engineering*, Vol. 62, No. 11, pp1519-1558

A new algorithm to generate 3-dimensional (3D) mesh for thin-walled structures is proposed. In the proposed algorithm, the mesh generation procedure is divided into two distinct phases. In the first phase, a surface mesh generator is employed to generate a surface mesh for the mid-surface of the thin-walled structure. The surface mesh generator used will control the element size properties of the final mesh along the surface direction. In the second phase, specially designed algorithms are used to convert the surface mesh to a 3D solid mesh by extrusion in the surface normal direction of the surface. The extrusion procedure will control the refinement levels of the final mesh along the surface normal direction. If the input surface mesh is a pure quadrilateral mesh and refinement level in the surface normal direction is uniform along the whole surface, all hex-meshes will be produced. Otherwise, the final 3D meshes generated will eventually consist of four types of solid elements, namely, tetrahedron, prism, pyramid and hexahedron. The presented algorithm is highly flexible in the sense that, in the first phase, any existing surface mesh generator can be employed while in the second phase, the extrusion procedure can accept either a triangular or a quadrilateral or even a mixed mesh as input and there is virtually no constraint on the grading of the input mesh. In addition, the extrusion procedure development is able to handle structural joints formed by the intersections of different surfaces. Numerical experiments indicate that the present algorithm is applicable to most practical situations and well shaped elements are generated.

Key words: automatic adaptive mesh generation; 3D solid mesh generation; thin-walled structures; extrusion

1. Introduction

In recent years, automatic mesh generation is one of the focus areas in the research of computational mechanics and computational geometry. This fact is reflected by the number of special issues and bibliography published in international journals in the last decade [1-8]. Since mesh generation could be defined as the process of breaking up a physical domain into small subdomains (elements) in order to facilitate numerical solution, a natural way to classify different mesh generation schemes is according to the dimensions of the domain to be discretized. By using such classification, nearly all mesh generation schemes available can be classified into the following three groups.

(1) 2D mesh generation schemes

Generation of meshes on planar surfaces could be considered as a well matured area in the research of automatic mesh generation. Efficient algorithms are now available for the generation of anisotropic [9,10] and full quadrilateral meshes [11].

(2) Surface mesh generation schemes

Recently, topics in surface mesh generation that are under intensive research focus included anisotropic meshes generation with gradation and geometric error control [12-14], the generation of anisotropic surface quadrilateral meshes [11,15,16] and mesh generation algorithms for non-analytical surface patches [17-19].

(3) 3D mesh generation schemes

Research in 3D mesh generation algorithm is one of the oldest research topics in automatic mesh generation. Within the area of 3D mesh generation, one of the topics which is subjected to most intensive research effort is complete unstructured hexahedral solid mesh generation [20, 21].

Analysis of thin-walled structures (TWS) is frequently encountered in many fields of engineering applications. In most engineering disciplines, a structure is considered as a TWS if its first two characteristic dimensions, (length/width) are much larger (e.g. one order of magnitude) than its third characteristic dimension (thickness). However, it should be noted that, among different disciplines, different values of the length to thickness ratio could be used. For example, in structural and offshore engineering, the value of such ratio could be as small as 10 while in the application of plastic injection molding modelling, the ratio adopted is usually bigger (e.g. 50). Traditionally, in structural engineering, plate and shell elements are used for analyzing of TWSs and the finite element (FE) meshes needed are often generated by an automatic surface mesh generator. Even though many efficient shells elements are available, due to the zero normal stress assumption, they are unable to predict the normal

stress variation along the intersection of two shell structures. Thus, the use of shell elements will not be adequate during the fatigue analyses of thin-walled structures joints. Examples of such type of structures included all kinds of tubular structures which are commonly used in the construction offshore platforms [22, 23]. Even in principle, accurate stress concentration factor (SCF) predictions and fatigue analyses can be obtained by using 3D FE method, the use of full 3D FE models for the study of TWSs is so far uncommon. The two main difficulties faced by many engineers who tried to use 3D FE models for the study of TWS joints are

- (1) Most of the conventional 3D mesh generation schemes available to date are primary not designed for the discretization of TWS in which the aspect ratio of the problem domain in the thickness direction is at least one order less than that of the other dimensions.
- (2) Even a conventional 3D mesh generator is able to discretize the TWS into a solid 3D mesh, since the aspect ratio of elements are similar in all directions, it will result in impractical number of degrees of freedoms.

Hence, in order to carry out an accurate SCF or fatigue analysis for TWS joints, a number of researchers have resolved to some tailor-made mesh generation procedures [24, 25]. In practice, in order to optimize the cost effectiveness of the FE analysis, it is desired that refined elements with suitable size grading (in both surface and thickness directions) should be employed near the intersection while single layer of relatively large elements (in the surface direction) should be used at region far away from the intersection. Furthermore, for regions near the junction where the aspect ratio of the elements in all directions are similar, conventional displacement type finite elements could be used while specially formulated hybrid elements [26-28] with high shape aspect ratio tolerance should be employed at where the elements are elongated in the surface direction. In fact, due to such special element grading and element size aspect ratio requirement, the discretization of TWSs into 3D meshes appears to be a relatively new and unexplored topic in the research of automatic mesh generation. The main objective of this study is to suggest a new procedure for the generation of 3D solid mesh for TWSs. In particular, the current algorithm is mainly designed for the mesh generation of TWSs encountered in structural and offshore engineering. It is expected that it will most often be applied to structures with length to thickness ratio within the range of [10, 100]. However, it should be mentioned that the proposed algorithm will also be applicable to the cases when the ratio is outside such range. Furthermore, the FE mesh generated could be used in both structural/stress analyses as well as other applications such as thermal and molding analyses. The proposed procedure will break down the whole mesh generation procedure into two phases, namely, *mesh generation for the mid-surface* of the

TWS and *solid element formation by extrusion*. In this paper, attention will be focused on the development of the special algorithms used in the extrusion phase.

In the next section, a description of the underlying principles employed for solid mesh formation for a single shell surface will be given. It will then be followed by the descriptions of special algorithms used for the treatment of surface intersections. Methods to maintain the corrected connectivity among different layers of elements during volume mesh refinement will then be given in Section 4. In Section 5, new mesh quality measures and mesh quality enhancement procedures proposed for the current application will be introduced. Finally, several mesh generation examples will be provided to demonstrate the performance, effectiveness and reliability of the suggested scheme.

2. Basic generation principles and extrusion for single surface

2.1 Basic generation principles and surface mesh generation

The main idea of the proposed mesh generation procedure is to divide the 3D solid element generation procedure into two main phases (Fig. 1). In the first phase, a surface mesh generator will be employed to discretize the surface model of the TWS into surface mesh. In the second phase, a specially designed algorithm will first convert the surface mesh to a *volume mesh* for the definition of the volume of the TWS. This volume mesh will then be further *refined* to a solid mesh for 3D FE modelling. In the final 3D FE mesh, element size grading along the surface direction will be determined during the surface mesh generation phase while mesh refinement along the thickness direction will be carried out in the extrusion phase. By separating the whole mesh generation progress into two main phases, both the generation speed and flexibility of generator are increased. Furthermore, the complexity of the whole mesh generation algorithm is reduced and it is possible to use any existing surface generator (e.g. the anisotropic quadrilateral mesh generated described in reference [14]) in the first phase to achieve the desired element size grading requirement. In addition, the two-phase approach also allows one to design the 3D solid mesh algorithm in such a way that it will require a minimum amount of information from the surface geometrical model of the TWS. In fact, once the surface mesh corresponding to the mid-surface of the TWS is available, the proposed conversion procedure can generate a 3D solid mesh without referring to the underlying geometrical model of the TWS.

Note that as the main target area of the proposed mesh generator is for the discretization of structures encountered in structural and offshore engineering. Hence, the term TWS in here refers to *thin-walled structures*, rather than *thin-walled solids* which are commonly used in

mechanical engineering. When modelling this kind of structures in structural and offshore engineering, very often the mid-surface model of the structure exists *before* the solid model is created: Global analysis is carried out using shell elements based on estimated thickness of the structure. A 3D model is then created (e.g. using the algorithm presented in this paper) for detailed analysis at critical locations (e.g. structural joints). Note that such approach is rather different from the procedure used in mechanical design in which the 3D model is created first (using CAD/CAM software) and the mid-surface model is obtained by extraction algorithms (e.g. medial axis transform). Hence, it should be stressed that the problem of how to extract a surface model from a 3D solid model is *not* the main focus of the present work. However, it should be mentioned in here that, once the mid-surface model is available, the presented algorithm could be able to discretize a thin-walled solid to a 3D FE mesh as well.

2.2 *Extrusion for single surface*

In the current generation scheme, solid mesh will be generated by the process of *extrusion* along the surface normal direction of the input surface (Fig. 2). It is assumed that both the surface normal vector and the thickness of the TWS will vary along the surface. After the extrusion, each mid-surface element will be converted to a *volume element*. Triangular elements are converted to prism volume elements and quadrilateral elements are converted to hexahedral volume elements (Fig. 3). Fig. 3 also illustrates that after extrusion, the original mid-surface element becomes a *middle facet* and locates inside the volume element. By contrast, the top and bottom surfaces of the volume element are defined as the *top facet* and the *bottom facet* respectively. In addition, *normal sides*, which defined the extrusion directions at the corner nodes, are generated during the extrusion process. Note that the extrusion process used here is somehow similar to the offsetting technique used by Lohner et. al. [29] and Marcum [30] for boundary layer mesh formation in CFD studies. However, the methods used in references [29] and [30] are developed for volume meshing of the *fluid domain* enclosing the solid boundary while in here the extrusion technique developed is for the meshing of *solid domain* of the TWS. Furthermore, a similar approach for generation boundary layer mesh for solid domain was purposed by Athanasiadis and Deconinck [31]. However, in reference [31], the algorithm developed is mainly for the generation of boundary mesh with uniform layers of elements and no detailed discussion for the treatment of surface intersections were given. Recently, another remarkable approach for generating 3D solid meshes for thin-walled solids was purposed by Quadros and K. Shimada [32]. The algorithm purposed in reference [32] uses a special algorithm to extract the *chordal surface* from the 3D

solid model. An all quad-element mesh generator is then applied to discretize the surface model before the final all-hex mesh with uniform layers of elements along the surface direction is generated by simple extrusion process.

In the current implementation, the extrusion algorithm is designed in such a way that it will make use of the following input data:

- (1) A surface mesh of the mid-surface of the TWS.
- (2) The thickness, h , and the surface normal vectors, $\mathbf{b}$, of the TWS at the nodal points of the surface mesh.
- (3) The number of layers of elements, EL , required to be generated in the thickness direction of the elements of the surface mesh.

It should be mentioned that the nodal surface vector is only an *optional* input for the conversion process as it is possible to compute them based on the coordinates of the nodal points and the connectivity of the surface mesh. For a given surface node A , the *optimal unit normal vector*, $\mathbf{b}_A$, at that node is calculated by first retrieving the normal vectors of its adjacent elements (Fig. 4). The normal vector $\mathbf{b}_A$ can then be obtained by applying the least-squares fitting (LSF) method. During the fitting process, the error function contribution due to the i th adjacent elements [30] used is defined as

$$e_i = 1 - \mathbf{b} \cdot \mathbf{n}_i \quad (1)$$

where $\mathbf{n}_i = [n_i^x, n_i^y, n_i^z]^T$ is the surface normal vector of the i th adjacent element. By using the LSF technique, the vector $\mathbf{b}_i$ can be obtained by minimizing the sum $\sum_{i=1}^{NB_i} e_i^2$ and the resulting equations are:

$$\sum_{i=1}^{NB_i} (\mathbf{b}_A \cdot \mathbf{n}_i) \cdot n_i^x = \sum_{i=1}^{NB_i} n_i^x, \quad \sum_{i=1}^{NB_i} (\mathbf{b}_A \cdot \mathbf{n}_i) \cdot n_i^y = \sum_{i=1}^{NB_i} n_i^y, \quad \sum_{i=1}^{NB_i} (\mathbf{b}_A \cdot \mathbf{n}_i) \cdot n_i^z = \sum_{i=1}^{NB_i} n_i^z \quad (2)$$

After the $\mathbf{b}_A$ is computed it will be normalized and volume elements will be created by extrusions along the $+\mathbf{b}_A$ and $-\mathbf{b}_A$ directions for a distance equal to $h_A/2$ where h_A is the thickness of the TWS at node A . It should be remarked that Eqn. 2 will only has unique solution provides that there exists at least three distinct normal vectors $\mathbf{n}_i$. In such cases that only one (Fig. 5a) or two (Fig. 5b) distinct normal vectors exist, the only normal vector $\mathbf{n}_1$ or the average of the two distinct normal vectors $(\mathbf{n}_1 + \mathbf{n}_2)/2$ will be used as the optimal unit vector $\mathbf{b}_i$ respectively.

Note that for the computation of the normal vectors, LSF is used rather than the weighted least-squares fitting (WLSF) (e.g. used the areas of adjacent elements as weights) or the

simple averaging (SA) method because it is well-known that for a given surface model, the WLSF and SA procedures could produce variations in normals due to purely topology (connectivity of elements surrounding a node) or local face areas difference. A simple LSF approach eliminates such variations.

3. Treatment for surface intersections and corner points

In order to generate volume meshes for practical engineering applications, it is essential that the surface extrusion procedure should be extended to cover surface intersections. In this study, special algorithms are developed for the treatments of the following surface intersection situations:

- (i) *Simple surface intersections* in which only two surfaces are involved (Fig. 6a),
- (ii) *multi-surface intersections* in which more than two surfaces are involved (Fig. 6b), and
- (iii) *corner point intersections* in which a corner point is created (Fig. 6c).

3.1 Simple surface intersections

For the treatment of simple surface intersection involving two surfaces, surface I and surface II, (Fig. 7) the final solid mesh will be formed in two steps. In the first step, the simple extrusion algorithm will be applied to the two intersecting surfaces independently. In the second step, modification algorithms are employed to treat those overlapping elements formed near the junction of the TWS.

In order to explain the algorithm used, two definitions will be given first. The intersection curve of the two mid-surfaces is defined as the *2D-cross curve* while the curve surface formed by the intersection of the two face walls of the two volume meshes is referred as the *3D-cross face* (Fig. 7). In addition, the angle between the two intersection surfaces is denoted as φ (Fig. 8). In general, the value of φ varies along the 2D-cross curve and it will only be constant when the two surfaces are both planar. The values of φ at all the nodes lying on the 2D-cross curve could be computed straightforwardly using the surface mesh information. For each node on the 2D-cross curve, the angle between the two normal vectors respect to each surface is defined as σ (Fig. 8) such that

$$\varphi = \pi - \sigma \quad (3)$$

Let θ_1 and θ_2 be the angles between the 3D-cross face and surfaces I and II respectively (Fig. 9). In addition, the thickness of the surfaces at the nodes lying on the 2D-cross curve will be

retrieved. As shown in Fig. 9, if the thicknesses of the two surfaces at the nodes under consideration are equal to h_I and h_{II} respectively, then θ_I and θ_{II} can be computed as:

$$\theta_I + \theta_{II} = \varphi, \quad \frac{h_I}{2\sin\theta_I} = \frac{h_{II}}{2\sin\theta_{II}} \quad \text{and} \quad \theta_I = \tan^{-1}\left(\frac{h_I\sin\varphi}{h_I\cos\varphi + h_{II}}\right), \quad \theta_{II} = \tan^{-1}\left(\frac{h_{II}\sin\varphi}{h_{II}\cos\varphi + h_I}\right) \quad (4)$$

By using Eqn. 4, the normal sides for all the nodes lying on the 2D-cross curve can be modified such that they will be conformal with the 3D-cross face of the intersection.

3.1.1 Re-definition of surface normal vectors for simple surface intersection

When the angle φ is small and the thickness of the surfaces I and II are larger than the element size near the intersection, some normal sides of the volume elements near the intersection may cut into the 3D-cross face (Fig. 7). In general, when projecting the 3D-cross face onto the intersecting surfaces, a *shadow area* is generated (Fig. 10). Note that both intersection surfaces have their own shadow areas. If there are surface nodes located inside these shadow areas, it can be affirmed that the normal sides passing through these nodes will intersect with the 3D-cross face and their directions must be re-defined. Furthermore, it is found that if the normal side of a given node is re-defined, it is often required to modify the normal sides associate with all the nodes adjacent to it. As a result, all the surface nodes that are located inside the shadow area and their adjacent nodes will be collected into a set denoted as $\mathcal{J}$. All the nodes belong to $\mathcal{J}$ will be checked and their original optimal surface normal vectors (Eqn. 2) may be re-defined. Nodes belong $\mathcal{J}$ will be grouped in layers according to their connectivity with the 2D-cross curve (Fig. 11). By starting from the first node in the first layer, nodes are checked layer by layer until no normal side is modified in the current layer.

For the given node $B \in \mathcal{J}$ to be checked, a connection vector will be drawn from node B to a point A such that BA is perpendicular to the 2D-cross curve at A (Figs. 12 and 13a). Since the 2D-cross curve is discretized into line segments, the following three situations may occur.

Situation 1: $\angle BN_1N_3 > 90^\circ$ and $\angle BN_1N_2 < 90^\circ$ (Fig 12a)

In this case, the perpendicular point A is located on the segment N_1N_2 .

Situation 2: $\angle BN_1N_3 < 90^\circ$ and $\angle BN_1N_2 < 90^\circ$ (Fig. 12b)

In this case, two perpendicular points A and C can be found on the segments N_1N_2 and N_1N_3 respectively and any one of them could be used. In the current implementation, point A will be selected if the numbering of N_2 is less than that of N_3 and vice versa.

Situation 3: $\angle BN_1N_3 > 90^\circ$ and $\angle BN_1N_2 > 90^\circ$ (Fig. 12c)

In this case, no perpendicular point can be found and the nearest point to B, N_1 will be selected.

After the location of point A is defined, the surface normal vector at A, $\mathbf{b}_A$ will be calculated.

Assume that the optimal surface normal vector at node N_1 and N_2 are $\mathbf{b}_{N_1}$ and $\mathbf{b}_{N_2}$ respectively. For situations I and II, $\mathbf{b}_A$ can be obtained by linear interpolation as

$$\mathbf{b}_A = \frac{\mathbf{b}_{N_1}|AN_2| + \mathbf{b}_{N_2}|N_1A|}{|N_1N_2|} \quad (5)$$

while $\mathbf{b}_A$ will be taken as $\mathbf{b}_{N_1}$ for situation III.

After $\mathbf{b}_A$ is determined, the re-defined normal direction for node B will be re-calculated according the algorithm given in Box 1 using point B in Fig. 13 as an example.

- (i) A connection vector BA will be drawn from node B to point A
- (ii) The original normal direction, $\mathbf{b}_B$, and thickness, h_B , at node B and the original normal side at B, $\mathbf{V}_B$ will be retrieved.
- (iii) The angle between the line AB and $\mathbf{V}_B$ will be computed. If this angle is less than $\pi/2$, then the normal $\mathbf{V}_B$ will be kept unchanged and goto (vii). Otherwise, the angle between $\mathbf{b}_A$ and $\mathbf{b}_B$ is computed and denoted as δ .
- (iv) The location of point M, the midpoint of BA, will be found.
- (v) From M a line HI will be drawn along the $\mathbf{b}_A$ direction such that $|IH| = h_B/\cos\delta$.
- (vi) The line IB is then connected and extended to point J (Fig. 13b). The line JI will then be re-defined as the new normal side for node B by setting and $\mathbf{V}_B \leftarrow \mathbf{JI}$. The new optimal normal direction will be re-defined as $\mathbf{b}_B \leftarrow \mathbf{JI}/|\mathbf{JI}|$.
- (vii) Quit with the new normal side $\mathbf{V}_B$ and new optimal normal direction $\mathbf{b}_B$.

Box 1: Re-definition of normal side and surface normal direction at node B.

As shown in Fig. 13b, the new normal sides for all other nodes belong to $\mathcal{J}$ will be obtained in a similar way. Note that when the node under consideration is close to the 2D-cross curve, the angle between the connecting line and the final normal side will be an obtuse angle (e.g. nodes B and C in Fig. 13b). When the node is far from the 2D-cross curve, the value of this angle will decrease. Once this angle becomes an acute angle, the original normal side of the node will not be changed (e.g. node D in Fig.13b).

3.1.2 Adjustment of normal sides near surface intersections

It is found that even though the normal sides near the 2D-cross curve are modified according to the method described in the last section, invalid volume elements may still be formed after the extrusion process. Such situation is more common when the surface mesh is highly anisotropic and contains a large number of highly stretched elements near the surface intersection. A typical case of an invalid volume element is shown in Fig. 14 in which the mid-surface element ABC is extruded along directions AA', BB' and CC' at nodes A, B and C respectively. It can be seen that due to the rapid change in the surface normal directions, the normal vectors for the middle facet ABC ($\mathbf{N}$) and upper facet A'B'C' ($\mathbf{N}'$) are opposite to each other such that $\mathbf{N} \cdot \mathbf{N}' < 0$ and resulted in an invalid volume element.

In order to avoid the formation of invalid volume elements, an additional step is implemented to further adjust the normal sides of nodes belong to the set $\mathcal{J}$. A normal side that requires further adjustment is called an *invalid normal side*. For a given normal side $\mathbf{V}_B$ (which may already be re-defined according to the procedure described in Section 3.1.1) associates with a node $B \in \mathcal{J}$, its validity will be checked by the procedure shown in Box 2.

- (i) Retrieve all the volume elements adjacent to the node B.
- (ii) For a given element retrieved, if *all* its normal sides have been checked according to the procedure described in Box 1, the validity of the element will be checked by computing the dot product $\mathbf{N} \cdot \mathbf{N}'$ where $\mathbf{N}$ and $\mathbf{N}'$ are the normal vectors corresponding to the middle and top facets of the volume element respectively (Fig. 14). A volume element is invalid if $\mathbf{N} \cdot \mathbf{N}' \leq 0$.
- (iii) If there exists at least one invalid element adjacent to node B, $\mathbf{V}_B$ is considered as invalid and it will be further adjusted, otherwise, $\mathbf{V}_B$ is valid and no adjustment is required.

Box 2: Checking the validity of a given normal side $\mathbf{V}_B$.

Assume that now the normal side $\mathbf{V}_B$ is invalid. The new direction of the adjusted normal sides will be established by perturbing it about its original direction as shown in Fig. 15. In Fig. 15, BA is the connecting vector to the 2D-cross curve (Box 1) such that it is perpendicular to the 2D-cross curve at A (Fig. 12). The new direction of the normal side will be determined by perturbing the vector $\mathbf{V}_B$ on the plane defined by the points A, B and B' (Fig. 15). Two series of perturbed vectors $\mathbf{V}_{B_i}^j$ will be generated such that

$$\mathbf{V}_{B_i}^j = \mathbf{V}_B \left(1 - \frac{i}{ND}\right) + (-1)^j \frac{i}{ND} (\overline{\mathbf{BA}}) \quad \text{for } j=1,2 \text{ and } i=1,\dots,ND \quad (6)$$

In Eqn. 6, ND is the maximum number of divisions used for the generation of perturbed vectors. From Eqn. 6, it can be seen that for the first series, $\mathbf{V}_{B_i}^1$, the normal side direction is shifted away from the surface intersection and the value of the angle $\angle ABB'$ is increased and vice versa for the second series $\mathbf{V}_{B_i}^2$. For any normal vector $\mathbf{V}_{B_i}^j$, the corresponding normal side for the upper half of the volume element, BB_i^j , will be generated in such a way that when it is projected back to the original optimal normal direction $\mathbf{b}_B$ (defined in Eqn. 2), its length will be equal to half of the thickness at the node B. In practice, in order to minimize the change in the normal side direction during the adjustment, the series $\mathbf{V}_{B_i}^1$ will be employed to generate the new normal side first. The generation sequence of the perturbed vectors for node B is shown in Box 3.

- (i) Loop over the index $j=1$ from 1 to 2
- (ii) Loop over the index i from 1 to ND
- (iii) Determine the vector $\mathbf{V}_{B_i}^j$ using Eqn. 7 and the corresponding new normal side.
- (iv) Test whether the new normal side generated is a valid normal size or not. If it is a valid normal side, quit with the valid normal side.
- (v) End for loop i
- (vi) End for loop j

Box 3: Adjustment of invalid normal side $\mathbf{V}_B$

The above adjustment procedure will be applied to all normal sides belong to $\mathcal{J}$ in a layer by layer manner similar to the case of normal side re-definition. Numerical tests show that by setting $ND=20$ in Eqn. 6, the adjustment procedure would normally remove all invalid normal sides even when the thickness of the surfaces is much larger than the element size along an acute intersection for a highly anisotropic mesh. In case that there are still invalid normal side after the first pass of the adjustment procedure, it will be repeated until all the invalid normal sides are corrected. Numerical experiences done show that it is rarely required more than one iteration to remove all invalid normal sides even for the most adverse case tested.

A simple example to demonstrate the actions for the normal side re-definition and adjustment process is given in Fig. 16. In this example, the structure modeled consists of two planar surfaces intersecting at 30° and the thicknesses of shells equal to the average element size near the intersection. As a result, 9 normal sides near the 2D-cross face are modified.

3.2 Multi-cross intersections

During the treatment of multi-cross intersection such as the one shown in Fig. 6b, each 3D-cross face and the volume elements generated for the two surfaces will be divided into two separate parts by its mid-surface and considered separately (Fig. 17). As a result, on each surface, two layers of volume elements will be formed.

In order to use Eqns. 3 and 4 to calculate the normal sides on the 2D-cross curve, normal vectors of the intersection surfaces will be reversed. For example, for a joint formed by the intersection of three surfaces I, II and III as shown in Fig. 18, only one unique 2D-cross curve is formed. However, during the treatment of this multi-cross intersection, three separated 2D-cross curves (and hence, 3D-cross faces) and six volume meshes will be created. After the surface normal vectors are defined, volume meshes can be generated by applying the procedure described in Section 3.1 on appropriate pairs of normal vectors in turns. The different surfaces and normal vectors used for the formation of 3D-cross curves and volume meshes for the multi-cross intersection shown in Fig. 18 are listed in Table 1. In general, for a surface intersection involves NS surfaces, NS 3D-cross faces and $2 \times NS$ surface normal vectors and volume meshes will be created.

2D-cross curves/ 3D-cross face	Surfaces used	Surface normal vectors used	volume mesh formed
1	I, II	$\mathbf{b}_I, -\mathbf{b}_{II}$	I_1, II_2
2	II, III	$\mathbf{b}_{II}, -\mathbf{b}_{III}$	II_1, III_2
3	II, I	$\mathbf{b}_{III}, -\mathbf{b}_I$	III_1, I_2

Table 1: Normal vectors of surfaces for different 2D-cross curve

3.3 Corner point intersections

For a multi-cross intersection involving a *corner point* which is formed by the intersection of three 2D-cross curves (Fig. 6c). The normal side for the node at the corner point is defined as the intersection line of three 3D-cross faces and its direction is determined by the surface normal vectors and the thickness of the three intersection surfaces.

As shown in Fig. 19, the surface node corresponding to the corner point is denoted as O and OO' is the normal side to be determined. Let points A, B and C are located on the bottom facets of the volume meshes corresponding to surfaces I, II and III respectively. The lines $O'A$, $O'B$ and $O'C$ are constructed so that they are perpendicular to surfaces I, II and III respectively. If the thicknesses of the TWS at point O for the three surfaces are equal to h_I , h_{II} and h_{III} respectively, then the coordinates of point O' , $(x_{O'}, y_{O'}, z_{O'})$, can be calculated by solving the following equations:

$$\begin{aligned}
|OO'| \cos(\angle OO'A) &= h_I \\
|OO'| \cos(\angle OO'B) &= h_{II} \\
|OO'| \cos(\angle OO'C) &= h_{III}
\end{aligned} \tag{7}$$

Eqn. 7 can be rewritten as

$$\begin{aligned}
(x_{O'} - x_O) \cdot \mathbf{b}_{I,x} + (y_{O'} - y_O) \cdot \mathbf{b}_{I,y} + (z_{O'} - z_O) \cdot \mathbf{b}_{I,z} &= h_I \\
(x_{O'} - x_O) \cdot \mathbf{b}_{II,x} + (y_{O'} - y_O) \cdot \mathbf{b}_{II,y} + (z_{O'} - z_O) \cdot \mathbf{b}_{II,z} &= h_{II} \\
(x_{O'} - x_O) \cdot \mathbf{b}_{III,x} + (y_{O'} - y_O) \cdot \mathbf{b}_{III,y} + (z_{O'} - z_O) \cdot \mathbf{b}_{III,z} &= h_{III}
\end{aligned} \tag{8}$$

where (x_O, y_O, z_O) are the coordinates of point O while $\mathbf{b}_\eta = (b_{\eta,x}, b_{\eta,y}, b_{\eta,z})$ for $\eta = I, II$ and III is the surface normal vector for surface η at the point O.

Similar to the case for multi-cross intersection without corner point, when the angles between intersection surfaces near the corner point are small or when the thickness of the TWS is large when comparing with the mid-surface element size, normal sides near the corner point will be re-defined and adjusted to avoid the formations of invalid elements and normal sides. For a multi-cross intersection with corner node, normal sides of surface nodes near the corner point will be affected by two pairs of intersecting surfaces. Hence, the final direction of the normal side will be determined by combining the two modified normal vectors obtained by consideration of the two pairs of surface intersections. For example, in Fig. 19, node D is a surface node close to the corner point O and its original normal surface vector (obtained by using Eqn. 2) is denoted as $\mathbf{b}_o$. The final normal vector for node D, $\mathbf{b}_D$ will be obtained by combining $\mathbf{b}_{I,II}$, the modified normal obtained by considering the intersection between surfaces I and II and $\mathbf{b}_{I,III}$, the modified normal obtained by considering the intersection between surfaces I and III. Note that the vectors $\mathbf{b}_{I,II}$ and $\mathbf{b}_{I,III}$ are obtained by apply the algorithms described in Section 3.1 on $\mathbf{b}_o$ separately (Fig. 20) and $\mathbf{b}_D$ can be expressed as

$$\mathbf{b}_D = (\mathbf{b}_{I,II} - \mathbf{b}_o) + (\mathbf{b}_{I,III} - \mathbf{b}_o) + \mathbf{b}_o \tag{9}$$

4. Volume mesh refinement

After the extrusion procedure, the surface mesh is converted to a volume mesh which consists of one layer (for single surface) or two layers (for multi-surface) of volume elements. Note that the middle facet of a volume element (Fig. 3) does not divide the volume element into two layers: It is only referred as the mid-surface definition of the underlying TWS. In practice, the volume mesh will be refined along the thickness direction to ensure the accuracy of the FE model.

4.1 Refinement parameters used

The volume mesh refinement process is controlled by the input data EL which define the number of layers of elements required in the thickness direction for all the elements of the surface mesh. In case that EL is constant for all elements, the division process is trivial and involving uniform division of the volume elements into EL *solid elements* as shown in Fig. 21. However, in order to optimize the cost effectiveness of the FE analysis, values of EL may vary along the input surface mesh and it is often required to divide the volume mesh into different numbers of layers of elements at different locations of the structure. Thus, a refinement scheme to maintain the compatibility of the edge and faces connectivity for the final solid mesh is needed. Towards this end, it is found that the complexity of refinement scheme increases as the *differences* in the EL values among adjacent volume elements (elements share a common side or face) increase. Hence, for practical consideration, it is restricted that the maximum difference in the EL values among adjacent elements cannot be greater than *two*. As mentioned in Section 2.2, EL is treated as one of the input data, it is possible that the input EL values may not satisfy the imposed restriction. Hence, an advancing front correction procedure is employed to rectify the input EL values (Fig. 22). Note that the correction procedure will first modify the EL values for those elements adjacent to elements with the highest EL value. After the EL values are rectified, the number of layers of divisions needed at the normal sides of the volume mesh, NL, will be computed as the maximum EL values adjacent to the normal side. For the normal side A shown in Fig. 23, NL_A , the number of layers of elements needed at A, is defined as

$$NL_A = \max(EL_i, i=1, \dots, 6) \quad (10)$$

4.2 Refinement procedures along the thickness direction

After the values of NL are defined, refinement will be carried out in the thickness direction to generate 3D solid elements. The refinement scheme used in this study consists of two distinct steps, namely, (i) the connection of dividing nodes for adjacent normal sides and (ii) the formation of 3D solid elements.

4.2.1 Connection of dividing nodes for adjacent normal sides

For a given normal side V_A , after the number of divisions needed, NL_A is computed, NL_A+1 *dividing nodes* will be created (Fig. 21). However, since NL may vary from one normal side to the other, a connection scheme is needed to ensure the compatibility of the edges constructed when adjacent normal sides (normal sides that share a common vertical face) are connected. The connection edges generated will also divide the vertical faces of the volume

elements in the thickness direction. Consider two adjacent normal sides V_A and V_B with NL values equal to NL_A and NL_B respectively such that $NL_A \geq NL_B$. The scheme employed to connect V_A with V_B will depend on the parities (odd and even) of NL_A and NL_B and their difference, $NL_A - NL_B$. Due to the constraint imposed on the values of EL in Section 4.1, it can be deduced that $NL_A - NL_B \leq 2$ and therefore, only five distinct cases are possible.

(1) $NL_A = NL_B$

In this case, dividing nodes on V_A and V_B are connected in a simple one-to-one correspondence manner as shown in Fig. 24a with $NL_A = 3$.

(2) $NL_A > NL_B$ and both NL_A and NL_B are odd numbers

In this case, it can be deduced that $NL_A - NL_B = 2$ and connection will start from the lowest dividing node of V_B to the second lowest dividing node of V_A . Connection then continues from the second lowest dividing node of V_B to the third lowest dividing nodes of V_A and so on. Construction will be ended when the uppermost dividing nodes of V_B is connected as shown in Fig. 24b where $NL_A = 5$.

(3) $NL_A > NL_B$ and NL_A is an even number while NL_B is an odd number

In this case, it can be deduced that $NL_A - NL_B = 1$ and connection will start from the second lowest dividing node of V_B to the second lowest dividing node of V_A . Connection then continues from the third lowest dividing node of V_B to the third lowest dividing nodes of V_A and so on. Construction will be ended when the uppermost dividing nodes of V_B is connected as shown in Fig. 24c where $NL_A = 5$.

(4) $NL_A > NL_B$ and NL_A is an odd number while NL_B is an even number

In this case, once again, one has $NL_A - NL_B = 1$ and connection is done by reversing the connection sequence for the case (3) as shown in Fig. 24d where $NL_A = 4$.

(5) $NL_A > NL_B$ and both NL_A and NL_B are even numbers

In this case, one has $NL_A - NL_B = 2$ and exactly the same connection sequence for the case (2) will be used as shown in Fig. 24e where $NL_A = 6$.

4.2.2 Formation of 3D solid elements

After all the dividing nodes for adjacent normal sides are constructed and vertical faces of the volume elements are divided, 3D elements will be formed by dividing the volume elements in the thickness direction. Depends on the shape (prism or hexahedron) of the volume elements, two different 3D element formation schemes are developed.

3D element formation scheme for prism volume elements

For prism volume elements which are generated by extrusion of triangular mid-surface elements, the 3D solid element formation scheme used is relatively simple. 3D solid elements

will be defined naturally by collecting edges and faces generated during the dividing nodes connection step and no further splitting of the volume elements is needed. An example to demonstrate the procedure used for an assembly of prism volume elements is given in Fig. 25. As shown in Fig. 25, three different types of 3D solid elements, namely, (1) 4-node tetrahedron, (2) 5-node pyramid [33] and (3) 6-node prism are generated.

3D element formation scheme for hexahedral volume elements

For a hexahedron volume element extruded from a quadrilateral mid-surface element, the element formation process will be more complicated and in some cases it is necessary to split a hexahedral volume element into two prism elements in order to maintain the compatibility of the resulted 3D mesh. Consider a hexahedral volume element E shown in Fig. 26 such that the number of divisions at its four normal sides, V_η , $\eta=A, B, C$ and D are equal to NL_η , $\eta=A, B, C$ and D respectively. From the constraint imposed on the values of EL and the definition of NL , it can be deduced that not all NL_η are different. That is, if one defines NLD_E as the number of normal sides that have different NL values, then NLD_E will satisfy the condition

$$1 \leq NLD_E \leq 3 \quad (11)$$

In addition, if one defines $NL_{\max} = \max(NL_A, NL_B, NL_C, NL_D)$ and NMS_E as the numbers of normal sides with NL values equal to $NL_{\max}$, then the range of NMS_E will satisfy the following conditions:

$$\begin{aligned} NLD_E = 1 &\Rightarrow NMS_E = 4 \\ NLD_E = 2 &\Rightarrow 1 \leq NMS_E \leq 3 \\ NLD_E = 3 &\Rightarrow 1 \leq NMS_E \leq 2 \end{aligned} \quad (12)$$

Furthermore, without loss in generality, one can always assume that $NL_A = NL_{\max}$. In case that when $NMS_E > 1$, the normal side V_A can always be selected in such a way that its node number is the smallest among all the nodes with $NL = NL_{\max}$. By using the definitions of NLD_E and NMS_E and Eqns. 11 and 12, the element formation scheme for the hexahedral volume element E can be summarized as follow.

(1) $NLD_E = 1$ and $NMS_E = 4$

In this case, dividing nodes on all normal sides will be connected in a simple one-to-one correspondence manner.

(2) $NLD_E = 2$ and $NMS_E = 2$

In this case, no splitting is needed and 3D solid elements can be formed directly by collecting edges and faces generated during the dividing nodes connection step.

(3) $NLD_E=2$, $NMS_E=1$ or 3 and $NLD_E=3$, $NMS_E=1$ or 2.

In these cases, it is necessary to split the hexahedral volume element into two prism volume elements along the diagonal faces formed by V_A and its opposite normal side V_C (Fig. 27). After the two prism volume elements are formed, dividing nodes on V_A and V_C are then connected along the diagonal faces. Finally, 3D solid elements will be formed by collecting edges and faces generated during the dividing nodes connection step.

In order to demonstrate the refinement procedure used, four set of examples showing how the elements are formed are given in Figs. 28 to 31 for cases when $NLD_E>1$ and splitting is needed. From Figs. 28 to 31, it can be seen that whenever hexahedral volume elements are involved, in addition to the tetrahedron, prism and pyramid elements, 8-node hexahedral elements will also be generated.

5. Mesh quality measurement and enhancement

Since in the current generation procedures, volume elements will be formed before they are further converted to solid elements, the mesh quality enhancement process will be applied in two separate steps. The first step, *normal sides smoothing* will be applied to the volume mesh while in the second steps, *dividing nodes smoothing* will be applied to the 3D solid mesh generated.

5.1 Volume mesh quality measurement and normal sides smoothing

Since volume elements are formed by extruding mid-surface elements along the optimal normal direction (Fig. 3), when measuring the shape quality of a volume element one should therefore consider both the *facet shape qualities* of its three (top, middle, bottom) facets and the *normal direction qualities* of the normal sides of the element.

5.1.1 Facet shape quality measurement

For a given volume element, the parameter $\phi \in [0,1]$ to measure its *facet shape quality* is defined as

$$\phi = (q_t \cdot q_m \cdot q_b)^{1/3} \quad (13)$$

where q_t , q_m and q_b are the shape quality measures for the top, middle and bottom facets of the volume element respectively. In here, the measures q_t , q_m and q_b are taken as the α and β qualities measures [34] for triangular and quadrilateral facets respectively. For a given normal side V , the parameter $\Phi \in [0,1]$ defined as

$$\Phi = \left(\prod_{i=1}^{NB} \phi_i \right)^{1/NB} \quad (14)$$

will be used to measure the average facet shape quality of all the volume elements connected to it. In Eqn. 14, NB is the number of elements adjacent to the normal side and ϕ_i is the facet shape quality of the i th volume element.

5.1.2 Normal direction quality measurement

For a given normal side $\mathbf{V}$ with normalized normal direction $\mathbf{b}$ that connects to NB mid-surface elements, the parameter $\Psi \in [0,1]$ defined as

$$\Psi = \frac{1}{NB} \sum_{i=1}^{NB} (\mathbf{b} \cdot \mathbf{n}_i) \quad (15)$$

will be used to measure the quality of the normal direction. In Eqn. 15, $\mathbf{n}_i$ is the normalized surface normal vector of the i th adjacent mid-surface element (Fig. 4). Ψ will attain its maximum value of unity when the normal direction $\mathbf{b}$ coincides with all the normal surface vectors of the adjacent mid-surface elements.

5.1.3 Overall shape quality measure and normal side smoothing

With the definitions of Φ and Ψ , the overall shape quality measure, Θ , for a normal side is defined as

$$\Theta = \Phi^\lambda \cdot \Psi^{(1-\lambda)} \quad (16)$$

and is used during the normal side smoothing. In Eqn. 16, λ is a real number such that $0 \leq \lambda \leq 1$. If λ is close to unity, during the smoothing process, normal side direction will be mainly adjusted based on the facet shape quality of the volume elements. However, if $\lambda=0$, smoothing will be entirely controlled by the normal direction quality. For the optimal value of λ , it is found that the shape qualities of the top and bottom facets are quite sensitive to the change in the directions of the normal sides and a higher value of λ will result in a more effective improvement of the shape quality of the volume mesh. In fact, it is found that a value of $\lambda > 0.5$ will often lead to better volume meshes. Further numerical tests show that a value of $\lambda=0.8$ seems to be appropriate for most practical cases and thus is adopted in the current study.

For a given normal side corresponding to a node A with normal side direction $\mathbf{b}$, the unit vector, $\mathbf{b}_A$ obtained by solving Eqn. 2 (or provided by the user) will be considered as the

optimal normal direction. During the normal side smoothing procedure, a series of vectors defined as (c.f. Eqn. 6)

$$\mathbf{b}_i = \mathbf{b}(1 - \frac{i}{ND}) - \frac{i}{ND}\mathbf{b}_A \quad \text{for } i=1, \dots, ND \quad (17)$$

will be generated in the order from $i=1$ to ND (Fig. 32). The vector $\mathbf{b}_i$ will be accepted if it leads to an increase in the overall shape quality measure Θ . Similar to the case of normal side adjustment (Section 3.1.2), it is found that a value of $ND=20$ will lead to improvement in most practical cases. In order to obtain the best smoothing effects, the normal side smoothing will start from the normal sides connected to surface intersections and three cycles of smoothing will be carried out.

5.2 Dividing node smoothing

After 3D solid elements are formed, dividing node smoothing will be applied to adjust the positions of *interior* dividing nodes. Note that in order to maintain the consistency between the final mesh and the original geometry of the TWS, coordinates of dividing nodes which are lying on the top and bottom boundaries will not be altered. Definitions of the shape quality measures used for 3D solid elements will be introduced first before the description of the dividing node smoothing procedure.

5.2.1 Face shape quality measurement for 3D solid elements

Similar to the facet shape quality measure for volume elements, the *face shape quality measure* for a solid element, $\tilde{\phi}$, can be defined as (c.f. Eqn. 13)

$$\tilde{\phi} = \sqrt{\tilde{q}_t \cdot \tilde{q}_b} \quad (18)$$

where $\tilde{q}_t$ and $\tilde{q}_b$ are the shape (α and β) qualities of the top and bottom faces of the 3D element respectively (Fig. 33). For a node connects to NB elements, the parameter $\tilde{\Phi} \in [0,1]$ defined as (c.f. Eqn. 14)

$$\tilde{\Phi} = \left(\prod_{i=1}^{NB} \tilde{\phi}_i \right)^{1/NB} \quad (19)$$

is employed to measure the average face shape quality for all the elements surrounding it.

5.2.2 Face normal vector quality measurement

Consider an interior dividing node Q for an assembly of 3D solid elements shown in Fig. 34, the *face normal vector quality measure*, $\tilde{\Psi}$, is computed by using the following steps.

- (i) Identify the dividing nodes that are lying on the same normal side and one level below and above Q, denoted them as L and U respectively. (Fig. 34a)
- (ii) Identify all the 3D elements that contain both the nodes Q and L (shaded elements in Fig. 34b) and collect them in a set denoted as $\mathcal{E}$.
- (iii) Retrieve the upper faces of elements in $\mathcal{E}$ and collect them in a set denoted as $\mathcal{F}$. Denote the number of faces in $\mathcal{F}$ as NF. The set $\mathcal{F}$ for Q is shown in Fig. 34c, with NF=6.
- (iv) Compute the unit normal vector of the faces retrieved and denoted them as $\mathbf{f}_i$ for $i=1, \dots, \text{NF}$.
- (v) Compute the face normal vector quality measure, $\tilde{\Psi} \in [0,1]$ as (c.f. Eqn. 15)

$$\tilde{\Psi} = \frac{1}{2 \cdot \text{NF}} \sum_{i=1}^{\text{NF}} (\mathbf{u} \cdot \mathbf{f}_i + \mathbf{l} \cdot \mathbf{f}_i) \quad \text{with} \quad \mathbf{u} = \frac{\overrightarrow{\text{QU}}}{|\overrightarrow{\text{QU}}|} \quad \text{and} \quad \mathbf{l} = \frac{\overrightarrow{\text{LQ}}}{|\overrightarrow{\text{LQ}}|} \quad (20)$$

Similar to Ψ , $\tilde{\Psi}$ will attain unity when the normal vectors of all the faces adjacent to Q coincides with both $\mathbf{u}$ and $\mathbf{l}$.

5.2.3 Overall shape quality measure and dividing node smoothing

With the definitions of $\tilde{\Phi}$ and $\tilde{\Psi}$, the overall shape quality measure, $\tilde{\Theta}$, for dividing node smoothing is defined as (c.f. Eqn. 16)

$$\tilde{\Theta} = \tilde{\Phi}^\lambda \cdot \tilde{\Psi}^{(1-\lambda)} \quad (21)$$

Similar to the case of normal side smoothing, a value of $\lambda=0.8$ is used in Eqn. 21. For a given dividing node Q, dividing node smoothing is started by identifying the face set $\mathcal{F}$ (Section 5.2.2, step (iii)) associates with it. New position of Q will be determined by applying the standard Laplaican smoothing method to the neighbourhood nodes in the set $\mathcal{F}$. The new position of a dividing node will be accepted only when the overall quality $\tilde{\Theta}$ after smoothing is greater than that before the movement. Again, three cycles of smoothing are applied.

6. Mesh generation examples

In this section, nine examples will be presented to demonstrate the performance of the solid mesh generation scheme developed. Examples 1 to 3 are employed to demonstrate the main actions of the algorithms presented in Sections 3 to 5. The structures involved are constructed by intersections of planar surfaces. However, the thickness of the surfaces used is large when comparing with the element size used. In Examples 4 to 9, the structures used consist of

intersections of curved surfaces. In addition, they are more realistic TWSs in the sense that the thickness of the shell wall is small when comparing with the other dimensions of the structures.

6.1 Examples 1 to 3

In Example 1, the model used is constructed by intersecting two thick planar surfaces. The input surface mesh and the final solid mesh generated are shown in Figs. 35a to 35c respectively. The average Θ and $\tilde{\Theta}$ values after volume mesh and solid mesh smoothings are equal to 0.53 and 0.68 respectively. Note that, in this example, the input surface model is highly anisotropic with small surface intersection angle and large thickness of the surfaces (when compared the element size). In such an extreme case, in order to obtain a valid solid mesh, it is necessary to adjust the normal side directions of the input surface mesh (as shown in Fig. 35b) in order to avoid the formation of invalid elements. As a result, the direction of the input adapted trend line is modified locally near the intersection. Note that such situations do not normal occur in the usual applications of the purposed mesh generator and the main purpose of showing this example is to demonstrate the robustness of the algorithm.

In Example 2, a multi-cross structure is used and the input surface mesh and output solid mesh are shown in Figs. 36a and 36b respectively. Again, it can be seen that normal sides are adjusted near the multi-cross intersection. For this example, the average Θ and $\tilde{\Theta}$ values after volume mesh and solid mesh smoothings are found to be 0.86 and 0.87 respectively.

In Example 3, a structure consists of a corner point formed by three planar surfaces is shown in Fig. 37. A zoom-in view for the solid mesh produced near the corner point is shown in Fig. 37c. For this example, the average Θ and $\tilde{\Theta}$ values after volume mesh and solid mesh smoothings are 0.86 and 0.98 respectively. Note that the values of Θ and $\tilde{\Theta}$ for Examples 2 and 3 are higher than that for Example 1. The reason is that in Example 1, the input surface mesh used is an anisotropic mesh with lower Φ and $\tilde{\Phi}$ qualities while uniform meshes are used in Examples 2 and 3.

It should be remarked that for Examples 2 and 3, due to the normal side adjustment procedures (Sections 3.1.1, 3.1.2 and 3.3) for the treatment of surface intersections and corner points, the grading of the elements near the surface intersection and corner points are modified. In general, due to the use of such adjustment algorithms, the sizes of elements along the surface junctions and corner points are often reduced. Despite this would modify the mesh

density locally, it is a desirable outcome as in most cases of structural analyses, denser elements are required along such structural junctions.

6.2 Examples 4 to 9

In Examples 4 and 5, the structure under consideration is a nozzle [11] constructed by intersection of three shell walls. The anisotropic triangular surface mesh used in Example 4 is shown in Fig 38a while the output solid mesh with uniform number of layers of elements is shown in Figs. 38b and 38d for the front and back views respectively. Zoom-in views for Fig. 38b and 38d are shown in Fig. 38c and 38e respectively. From these figures, it can be seen that the characteristics of the input surface mesh are preserved during the 3D solid element formation step. For Example 5, a uniform quadrilateral mesh (Fig. 39a) is used as the input surface mesh. However, in this example, both the wall thickness and the specified number of layer of elements in the thickness direction vary along the structures. Two different views for the solid mesh generated and their zoom-in views are shown in Figs. 39b and 39d and Figs. 39c and 39e respectively. In this case, since the number of layers of element varies along the mesh, during the volume mesh refinement step, some hexahedral volume elements are divided into prism volume elements.

In Examples 6 and 7, the structures modelled are formed by intersecting curve surfaces with planar surfaces. In Example 6, a cylinder is intersected with a plane (Figs. 40a and 40c) and the solid mesh generated is shown in Figs. 40b and 40d. In this example, smaller size elements and more element layers are used near the intersection as shown in the section view (Fig. 40e) and zoom-in view (Fig. 40f) of the solid mesh. In Example 7, a multi-cross structure is formed by the intersections of two planar surfaces with a curve surface (Fig. 41a). The 3D mesh generated and its section view are shown in Figs. 41b and 41c respectively. In addition, two zoom-in views for the section view are given Fig. 41d and 41e. Again, in this example, smaller elements and more elements layers are specified near the intersections.

In Example 8, the model employed is formed by the intersection of two equal radius tubes and is a typical tubular structure (X-joint) used in the construction of offshore platforms. The input graded surface mesh used and the solid mesh generated are shown in Figs. 42a and 42b respectively. Section and zoom-in views near the joint are given in Fig. 42c to 42f. It can be seen that in order to obtain an accurate solution near the joint, the mesh is refined towards the intersection in both the surface and thickness directions.

In the last example, the structure under consideration is formed by the intersection of a large structural steel spherical head with four smaller braces as shown in Fig. 43a. In this example,

the input mesh is a structured quadrilateral mesh obtained by the mapping technique. The solid mesh generated (Figs. 43b and 43c) has four layers of elements along the brace-head intersections while only two layers of elements are used in the rest of the structure. Section and zoom-in views of the 3D solid mesh generated are shown in Fig. 43d and 43e respectively.

7. Conclusions and future extensions

In this paper, an algorithm to generation 3D solid meshes for thin-walled structures is presented. Full 3D solid mesh is generated by first employing a surface mesh generator to discretize the mid-surface of the structure into surface mesh. After the surface mesh is generated, it is then fed to a mesh converter in which solid elements are formed by extruding and refining the surface mesh along the thickness directions. The presented mesh extrusion procedure is highly flexible in the sense that virtually any surface generator could be used in the first step and the mesh converter can accept triangular, quadrilateral or mixed meshes as input. Special algorithms are developed to handle surface intersection involving two or multiple surfaces and intersections contain corner points. It is found that the algorithms developed can produce valid and well shaped 3D solid elements near surface intersections even when the input surface mesh is highly anisotropic and the thickness of the surface at the intersection are larger than the element size there. A set of new mesh quality measurement parameters is also purposed to measure the element shape quality of 3D meshes for thin-walled structures. Furthermore, mesh enhancement procedures are also suggested to improve the shape quality of the mesh generated.

Regarding the speed of the mesh generation process, all the presented results are obtained within one minute by using a low-ended PC equipped with a 2.0GHz Intel P4 CPU and 512 Mb RAM. Hence, the mesh generation scheme could be fast enough to handle most practical cases on a common PC computing environment.

A potential area of extension for the present study is to develop algorithms to handle and generate meshes for detailed geometrical features such as surface cracks, fillets and welding profiles that are frequently encountered in thin-walled structures. Since in most practical cases, these detailed geometrical features are formed only after the thin-walled structures are constructed (e.g. welding profiles and surface cracks are formed only after the individual members are welded and the structure is loaded), algorithms designed to handle such features should be designed in an *ad hoc* manner and allow the desired features to be inserted to any location of an existing FE mesh.

References

1. Special Issue on Adaptive Meshing, Part I, *Fin. Ele. Anal. Des.*, **25**, No. 1/2, (1997), Guest Editor: S. H. Lo
2. Special Issue on Adaptive Meshing, Part II, *Fin. Ele. Anal. Des.*, **25**, No. 3/4, (1997), Guest Editor: S. H. Lo
3. Special Issue on Unstructured Mesh Generation, *Int. J. Numer. Methods Engrg.*, **49**, No. 1-2, (2001) Guest editors: Scott A. Canann, Sunil Saigal and Steven J. Owen
4. 9th International Meshing Roundtable Special Issue, *Engrg. with Comput.*, **17**, No. 3, (2001) Guest editors: S. Owen
5. Special Issue on 'Mesh Generation' (10th IMR 2001), *Engrg. with Comput.*, **18**, No. 3, (2002) Guest editors: A. Sheffer
6. Special Issue on Trends in Unstructured Mesh Generation, Edited by S. J. Owen and M. S. Shephard, *Int. J. Numer. Methods Engrg.*, **58**, No. (2003)
7. J. Mackerle, 2D and 3D finite element meshing and remeshing: A bibliography (1990-2001), *Engrg. Computation*, **18**, 1108-1197 (2001)
8. S. H. Lo, Finite element mesh generation and adaptive meshing, *Progress in Structural Engineering and Materials*, **4**, 381-300 (2002)
9. H. Borouchaki, P. L. George, F. Hecht, P. Laug and E. Saltel, Delaunay mesh generation governed by metric specifications. Part I. Algorithms, *Fin. Ele. Anal. Des.*, **25**, 61-83 (1997)
10. H. Borouchaki, P. L. George, F. Hecht, P. Laug and E. Saltel, Delaunay mesh generation governed by metric specifications. Part II. Applications, *Fin. Ele. Anal. Des.*, **25**, 85-109 (1997)
11. Y. K. Lee and C. K. Lee, A new indirect anisotropic quadrilateral mesh generation scheme with enhanced local mesh smoothing procedure, *Int. J. Numer. Methods. Engrg.*, **58**, 277-300 (2003)
12. S. J. Owen and S. Saigal, Surface mesh sizing control, *Int. J. Numer. Methods. Engrg.*, **47**, 497-511 (2000)
13. C. K. Lee, On curvature element-size control in metric surface mesh generation, *Int. J. Numer. Methods. Engrg.*, **50**, 787-807 (2001)
14. P. J. Frey and H. Borouchaki, Surface meshing using a geometric error estimate, *Int. J. Numer. Methods. Engrg.*, **58**, 227-245 (2003)
15. S. J. Owen, M. L. Staten, S. A. Canann and S. Saigal, Q-Morph: An indirect approach to advancing front quad meshing, *Int. J. Numer. Methods. Engrg.*, **44**, 1317-1340 (1999)

16. Y. K. Lee and C. K. Lee, Automatic generation of anisotropic quadrilateral meshes on three-dimensional surfaces using metric specifications, *Int. J. Numer. Methods. Engrg.*, **53**, 2673-2700 (2002)
17. J. Raul and R. Löhner, From medical images to anatomically accurate finite element grids, *Int. J. Numer. Methods. Engrg.*, **51**, 985-1008 (2001)
18. C. K. Lee, Automatics metric 3D surface mesh generation using subdivision surface geometrical model. Part 1: Construction of underlying geometrical model, *Int. J. Numer. Methods. Engrg.*, **56**, 1593-1614 (2003)
19. C. K. Lee, Automatics metric 3D surface mesh generation using subdivision surface geometrical model. Part 2: Mesh generation algorithm and examples, *Int. J. Numer. Methods. Engrg.*, **56**, 1615-1646 (2003)
20. S. J. Owen and S. Saigal, H-Morph: an indirect approach to advancing front hex meshing, *Int. J. Numer. Methods. Engrg.*, **49**, 289-312 (2000)
21. D. R. White and T. J. Tautges, Automatic scheme selection for toolkit hex meshing, *Int. J. Numer. Methods. Engrg.*, **49**, 127-144 (2000)
22. D. Bowness, M. M. K. Lee MMK, The development of an accurate model for the fatigue assessment of doubly curved cracks in tubular joints. *Int. J. Fracture*, **73**, 129-147, (1995)
23. M. M. K. Lee, Strength, stress and fracture analyses of offshore tubular joints using finite elements, *J. Constructional Steel Research*, **51**, 265-286, (1999)
24. J. J. Cao, G. J. Yang, J. A. Packer, F. M. Burdekin, Crack modeling in FE analysis of circular tubular joints, *Engrg. Fract. Mech.*, **61**, 537-553 (1998)
25. S. T. Lie, C. K. Lee and S. M. Wong, Model and mesh generation of cracked tubular Y-joints, *Engrg. Fract. Mech.*, **70**, 161-184 (2003)
26. S. H. Lo and L. Chen, Improvement on the 10-node tetrahedral element for three-dimensional problems, *Comput. Method Appl. Mech. Engrg.*, **189**, 961-974 (2000)
27. K. Y. Sze, S. H. Lo and L. Q. Yao, Hybrid-stress solid elements for shell structures based upon a modified variational functional, *Int. J. Numer. Methods. Engrg.*, **53**, 2617-2642 (2002).
28. K. Y. Sze, L. Q. Yao and T. H. H. Pian, An eighteen-node hybrid stress solid-shell element for homogenous and laminated structures, *Fin. Ele. Anal. Des.*, **38**, 353-374 (2002)
29. D. Sharov, H. Luo, J. D. Baum and R. Lohner, Unstructured Navier–Stokes grid generation at corners and ridges, *Int. J. Numer. Methods. Fluids.*, **43**, 717-728 (2003).
30. D. L. Marcum, Efficient generation of high-quality unstructured surface and volume grids, *Engineering with Computers*, **17**, 211–233 (2001)

31. A. N. Athanasiadis and H. Deconinck, Object-oriented three-dimensional hybrid grid generation, *Int. J. Numer. Methods Engrg.*, **58**, 301-318 (2004)
32. W. R. Quadros and K. Shimada, Hex-Layer: Layered all-hex mesh generation on thin section solid via chordal surface transformation, Proceedings of the 11th International Meshing Roundtable (2002), pp 169-180.
33. G. Bedrosian, Shape functions and integration formulas for three-dimensional finite element analysis, *Int. J. Numer. Methods Engrg.*, **35**, 95-108 (1992)
34. S. H. Lo, Generating quadrilateral elements on plane and over curved surfaces, *Comput. Struct*, **31**, 421-426 (1989)

Figure 1. Main steps of 3D solid mesh generation for thin-walled structures.

Figure 2. Volume mesh formation by extrusion of surface mesh.

Figure 3. Extrusion of surface elements to volume elements.

Figure 4. Normal vector at a node.

Figure 5. Degenerated cases for the fitting procedure: (a) only one distinct normal $\mathbf{n}_1$ exist; (b) two distinct normal vectors $\mathbf{n}_1$ and $\mathbf{n}_2$ exist.

Figure 6. Three types of structures with surface intersections: (a) a simple surface; (b) a multi-surface; and (c) a multi-surface with corner point.

Figure 7. Overlapping elements, 2D-cross curve and 3D-cross face.

Figure 8. Intersection angles between two intersecting surfaces.

Figure 9. Definitions of φ , θ_I and θ_{II} .

Figure 10. The situation when φ is small.

Figure 11. Surface node layers near the 2D-cross curve.

Figure 12. Location of perpendicular point A on the 2D-cross curve:
(a) $\angle BN_1N_3 > 90^\circ$ and $\angle BN_1N_2 < 90^\circ$; (b) $\angle BN_1N_3 < 90^\circ$ and $\angle BN_1N_2 < 90^\circ$;
(b) and (c) $\angle BN_1N_3 > 90^\circ$ and $\angle BN_1N_2 > 90^\circ$.

Figure 13. Re-definition of normal sides for nodes: (a) plan view; (b) section view.

Figure 14. Upper half of an invalid volume element formed after extrusion.

Figure 15. Adjustment of normal side V_B .

Figure 16. Treatment of volume meshes for two intersecting surface meshes.

Figure 17. Division of a 3D-cross face and volume mesh to two separate parts.

Figure 18. Multi-surface and the normal directions.

Figure 19. Calculation the normal side through the corner point.

Figure 20. Determine the final normal side for node A near a corner point.

Figure 21. Division of a volume element, EL=NL=3.

Figure 22. Rectification of input EL values: (a) input EL values; (b) modification starts at elements with highest EL values; (c) after first layer is modified; (d) after two layers are modified; (e) after three layers are modified; and (f) final EL values.

Fig. 23. Definition of NL.

Figure 24. Connection of adjacent normal sides: (a) case of $NL_A = NL_B = 3$; (b) case of $NL_A = 5 = NL_B + 2$; (c) case of $NL_A = 5 = NL_B + 1$; (d) case of $NL_A = 4 = NL_B + 1$; and (e) case of $NL_A = 6 = NL_B + 2$.

Figure 25. Formation of 3D solid elements from an assembly of prism volume elements: (a) an assembly of prism volume elements and NL at normal sides; (b) connections of normal sides; and (c) formation of 3D solid elements.

Figure 26: A hexahedral volume element E such that $NL_{\max}=3$, $NLD_E=3$ and $NMS_E=1$.

Figure 27. Splitting of a hexahedral volume elements into two prism volume elements.

Figure 28. Examples (set 1) for refinement of hexahedral volume element when $NLD_E=2$:
(a) $NMS_E=1$ (2-1-1-1); (b) $NMS_E=2$ (2-2-1-1); and (c) $NMS_E=3$ (2-2-1-2).
(Numbers inside brackets are values of $NL_A-NL_B-NL_C-NL_D$ respectively)

Figure 29: Examples (set 2) for refinement of hexahedral volume element when $NLD_E=2$:
(a) $NMS_E=1$ (3-1-1-1); (b) $NMS_E=2$ (3-3-1-1); and (c) $NMS_E=3$ (3-3-3-1)
(Numbers inside brackets are values of $NL_A-NL_B-NL_C-NL_D$ respectively)

Figure 30: Examples (set 3) for refinement of hexahedral volume element when $NLD_E=2$:
 (a) $NMS_E=1$ (3-3-3-2); (b) $NMS_E=2$ (3-3-2-2); and (c) $NMS_E=3$ (3-3-2-3).
 (Numbers inside brackets are values of NL_A - NL_B - NL_C - NL_D respectively)

Figure 31: Examples (set 3) for refinement of hexahedral volume element when $NLD_E=3$:
 (a) $NMS_E=1$ (3-2-1-1); (b) $NMS_E=1$ (3-2-2-1); and (c) $NMS_E=2$ (3-3-2-1).
 (Numbers inside brackets are values of NL_A - NL_B - NL_C - NL_D respectively)

Figure 32. Normal side smoothing.

Figure 33. Face shape quality measure for 3D solid elements:
(a) tetrahedron; (b) pyramid; (c) prism; and (d) hexahedral.

Figure 34. Face normal vector quality measure for interior node Q: (a) an assembly of 3D solid elements; (b) elements connectivity and (c) faces associated with interior node Q.

Figure 35. Example 1: (a) surface mesh; (b) solid mesh, front view; (c) solid mesh, back view.

Figure 36. Example 2: (a) surface mesh; (b) solid mesh.

(a)

(c)

(b)

(d)

Figure 37. Example 3: (a) surface mesh; (b) solid mesh, front view; (c) solid mesh, back view; and (d) zoom in view near corner point.

Figure 38. Example 4: (a) surface mesh; (b) solid mesh, front face; (c) zoom in view for front face; (d) solid mesh, back face; and (e) zoom in view for back face.

Figure 39. Example 5: (a) surface mesh; (b) solid mesh, front face; (c) zoom in view for front face; (d) solid mesh, back face; and (e) zoom in view for back face.

Figure 40. Example 6: (a) surface mesh, front view; (b) solid mesh, front view; (c) surface mesh, bottom view; (d) solid mesh, bottom view; (e) solid mesh, section view; and (f) zoom in view for section view.

(a)

(b)

(c)

(d) zoom in view A

(e) zoom in view B

Figure 41. Example 7: (a) surface mesh; (b) solid mesh; (c) solid mesh, section view; (d) zoom in view A; and (e) zoom in view B.

(a)

(b)

Figure 42. Example 8: (a) surface mesh; (b) solid mesh; (c) section view 1; (d) zoom in view for section view 1; (e) section view 2; and (f) zoom in view for section view 2.

(a) surface mesh

(b) solid mesh, top view

(c) solid mesh, bottom view

(d) solid mesh, section view

(e) zoom in view for section view

Figure 43. Example 9: (a) surface mesh; (b) solid mesh, top view; (c) solid mesh, bottom view; (d) solid mesh, section view; and (e) zoom in view for section view.