INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING
Int. J. Numer. Meth. Engng 2009; **77**:1247–1289
Published online 11 September 2008 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/nme.2446

Generation of viscous grids at ridges and corners

Romain Aubry and Rainald Löhner*, †

CFD Center, Department of Computational and Data Sciences, MS 6A2, George Mason University, Fairfax, VA 22030-4444, U.S.A.

SUMMARY

An extension of Löhner (*AIAA-93-3348-CP*, 1993) for the generation of high aspect ratio volume grids on surfaces with ridges and corners is presented for Reynolds-averaged Navier–Stokes computations. Multiple point normals are introduced along ridges and corners. The original technique generates a semi-structured boundary layer of prismatic elements growing along point normals. Therefore, extra degenerated faces must be introduced to take into account the multiple growth curves at ridges and corners and produce a valid topological surface triangulation. The major task of the algorithm consists in recovering conformity in the surface mesh triangulation, which has been lost due to the introduction of the virtual faces. The procedure relies on a topological taxonomy of an arbitrary combination of concave and convex ridges. Each case is highlighted in detail. Special boundary conditions such as symmetry planes and periodic boundary conditions are also handled. Several complex geometries have been chosen to illustrate the proposed procedure, and timings are given, showing that the new module does not place any extra burden on the previous semi-structured approach. Copyright © 2008 John Wiley & Sons, Ltd.

Received 8 April 2008; Revised 30 June 2008; Accepted 10 July 2008

KEY WORDS: unstructured grid generation; anisotropic grid; multiple point normal; corner; ridges; RANS

1. INTRODUCTION

Although large eddy simulation (LES) has received much attention recently and huge improvements have been achieved [1], a typical LES run in a laminar flow would require $O(10^{10})$ grid points, which is still three orders of magnitude higher than the current Reynolds-averaged Navier–Stokes (RANS) production codes. Therefore, high-quality RANS grid will be needed in the foreseeable future.

Compared with the now mature isotropic and slightly anisotropic unstructured grid generation, the generation of unstructured grids suitable for RANS calculation still gives rise to numerous

*Correspondence to: Rainald Löhner, CFD Center, Department of Computational and Data Sciences, MS 6A2, George Mason University, Fairfax, VA 22030-4444, U.S.A.

†E-mail: raubry@gmu.edu, rlohner@gmu.edu

difficulties:

- The very high aspect ratio of the cells ($O(10^3-10^5)$) makes all the numerical decisions involved in the point and connectivity creation of the boundary layer much more prone to inaccuracy and possible failure, than in the isotropic case. Advancing front methods typically generate 'ideal' points from a given face of the front, so that they can hardly handle stretching higher than (1:10). Delaunay-based mesh generators are highly sensitive to point placements, which may cause poorly connected elements in the boundary layer, and thus the boundary regeneration is cumbersome.
- The growing curve dilatation or extrude process is highly sensitive to the point normal computation, for which it is difficult to choose between a local and a more global optimal point normal. Indeed, if for a given surface triangle, each point normal is optimal in a sense to be defined, it still may not be optimal for the prism carried by the face, and some kind of global smoothing must be performed.
- Once the boundary layer mesh generation process has finished, an isotropic mesh generator is used to close the volume. At this point, it is extremely important that no stretched faces be exposed to the isotropic mesher since it can hardly handle such degenerated configurations.

In an attempt to classify the different methods proposed in the literature, it has been found useful to distinguish three main groups with their own subgroups:

- The boundary layer mesh is created before the isotropic mesh:
  - ○ The method is based on an advancing front or an advancing layer method.
  - ○ It relies on a Delaunay point connection process.
  - ○ It is based on a hybrid approach.
- The boundary layer mesh is created after the isotropic mesh.
- Some kind of interaction between the surface and volume mesh is present.

In the first category, regarding the advancing front methods, early attempts on generating viscous grids can be found in Löhner [2] and Pirzadeh [3], where the first reference builds a whole boundary layer mesh at once, while the second proceeds layer by layer, both in a semi-structured manner. The semi-structured methods have the advantage of avoiding, to a minimal extent, bad connectivities with an already optimal point placement. However, concave and convex ridges/corners are not gridded optimally due to a lack of multiple normals for the convex configurations, and the directional enrichment [4] for the concave one. Furthermore, the lack of multiple normals makes the creation of a unique point normal visible from all the emanating faces impossible in complex corners. In Hassan *et al.* [5], it is noted that the point creation in the classical advancing front is incompatible with a viscous grid, so that it can be extruded from their normals. The boundary layer is constructed layer by layer, blending the normals from perpendicular on the body to smooth in the field. The connexion process, which is the difficulty here, is however not always simple and obvious. Connell and Braaten [6] compute the point normals from the underlying surface model for corners points, then for ridge points and finally for standard smooth surface points. Particular attention is paid to deleting the bad aspect ratio faces obtained after the element removal. These faces are deleted by edge collapse to prevent a possible failure of the isotropic mesh generator. This technique has been used in the present paper and has been found to be of utmost importance as well as straightforward to implement.

As far as a Delaunay-based method is concerned, a Delaunay point insertion is proposed in Löhner and Cebral [4], which enhances robustness as the mesh is always available but which, like all the Delaunay methods, is very sensitive to the point placement and may cause large dihedral angles, which preclude the use of monotonic schemes. A Delaunay approach is also followed in Marcum [7], where the point insertion is based on an iterative Delaunay-Min/Max criterion. Inserting the points from the surface outwards, the elements of the previous layers are constrained to be kept in order to reach a higher degree of regularity. However, the Delaunay method gives very stretched elements from the body outwards because of the bounding box in a Bowyer–Watson algorithm, so that the insertion of points in the boundary layer could only strengthen this difficulty and make the connection process more sensitive as described earlier. In Peraire and Morgan [8], a directional refinement based on the distance to the surface is used for the point creation, and a Delaunay method is also used for the point insertion. Finally, in Garimella and Shephard [9], a multiple point normal method is proposed at convex ridges with particular emphasis on creating transition tetrahedra to avoid the stretched faces to be exposed to the isotropic mesh generator in a similar manner as the previous reference. Non-manifold configurations are also treated. However, only the extreme normals are used at first to generate the boundary layer. The other multiple growth curves are used afterwards to create boundary layer blend elements to fill the holes. The variable blend is on commented but not described. Relying on a different point of view, these elements are automatically generated with the method proposed in this paper.

In the hybrid mesh generation field, Kallinderis and Ward [10] generate prismatic elements with the help of voxels combined with an elliptic step. Recently, Athanasiadis and Deconinck [11] proposed a folding/unfolding algorithm to treat concave/convex ridges. It relies on the notion of a degenerate or virtual surface grid generated at convex ridges, which is also used in this paper. Examples of some convex and convex/concave corners are presented but more complicated corners may appear.

In the second group, a different approach is proposed in Bottasso and Detomi [12], where the boundary layer is blown away from the surface triangulation. However, as the extrude process relies on a unique point normal at concave and convex ridges, the same problems as in the semi-structured methodology arise. In the same manner as Bottasso *et al.*, Ito and Nakahashi [13, 14] follow the idea of beginning with an isotropic mesh and making the boundary layer mesh grow from the wetted surface. Ito and Nakahashi [13, 14] were the first to use the term ‘virtual’ for edges, faces or elements that have null length, area or volume. They incorporate multiple normals along convex ridges to generate hybrid meshes comprised of prisms, hexahedra and tetrahedra. It must be noted that these references are the point of departure of the present paper to properly treat corners and ridges. However, they neither propose a general method nor a general classification of configurations is met.

More recently, Sharov *et al.* [15] propose a completely different approach based on the observation that there is an inconsistency between a non-viscous isotropic surface mesh triangulation and the viscous volume mesh, which implies that an interaction between the surface and the volume mesh generator must be provided. After having identified all the ridges, the surface mesh is clustered around them with the imposed size of the minimal boundary layer thickness. The point normals are then computed and smoothed at convex ridges, and nothing is performed at concave ridges. The boundary layer mesh is then extruded with the semi-structured algorithm proposed in Löhner [2], leaving all the surfaces exposed to the isotropic generator with a well-defined ratio. The method proposed in this paper has various common aspects with this reference, but relies on a completely different approach. The directional enrichment for concave configurations is attractive but gives rise to a considerable increase in the mesh size without a significant impact on the results. Furthermore, the clustering of the surface triangulation may be avoided on

DOI: 10.1002/nme

convex ridges with the introduction of multiple normals, which greatly reduces the size of the grid.

In this paper, the method proposed in Löhner [2] is extended to allow semi-structured methods to deal with ridges and corners. As seen in Figure 1 in two dimensions for the sake of clarity, a unique normal at a ridge may produce highly distorded elements, even supposing that a method is available to compute the best normal at this point in a given sense [16]. In three dimensions, with concave ridges, this unique normal may not exist, and null or negative elements will be created. The alternative is presented in Figure 2, where the introduction of multiple normals alleviates the constraint created by the limited visibility cone around the point under consideration.

More precisely, the aim of the method is to take advantage of the robustness of the semi-structured boundary layer method, and to add multiple normal features at convex configurations. This is mainly done in two steps. First, a classical identification and analysis of the nature of the ridges is performed. Then, the mesh is made conforming at the nodes with these extra faces, relying on the particular topology of each node touched by a convex ridge. Therefore, the constraint imposed by the semi-structured method that each point must have one and only one normal is then verified at multiple normal points. From a general point of view, it is just another toolbox added in a classical advancing front generator, fully automatic, without the complexity of a block-structured definition of the geometry.

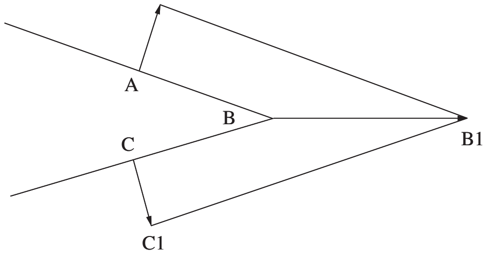

Figure 1. Distorded elements at a convex ridge (cut in two dimensions).

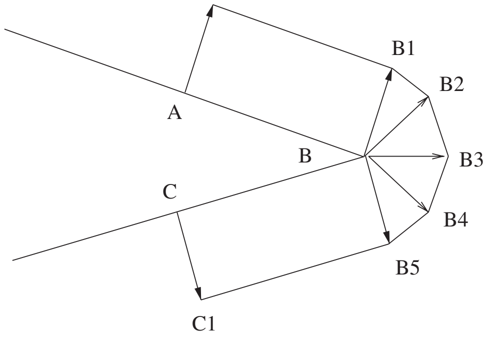

Figure 2. Well-defined mesh at a convex ridge (cut in two dimensions).

DOI: 10.1002/nme

After this introduction, Section 2 of this paper offers a general view of the proposed method. Then, the detailed classification and implementation of the method are highlighted in Sections 3 and 4, presenting first the points connected to a maximum of two ridges or ridge points, and then the more complicated corner points. Section 5 explains the methodology used for the point normal computation, on which the prisms will rely. Sections 6 and 7 concentrate on the implementation of particular boundary conditions often used in computational fluid dynamics, namely, the planes of symmetry and the periodic boundary conditions. Section 8 gives additional insight on the optimization part. Finally, various numerical results illustrate the method in Section 9.

2. PRESENTATION OF THE PROCEDURE

As previously mentioned, the proposed method is an extension of the work presented in Löhner [2, 17]. The idea consists, after an adequate point normal computation, in generating the boundary layer grid in a semi-structured manner by extruding the surface mesh, relying on the background mesh information, forming prisms, dividing these prisms into a suitable combination of tetrahedra, and eliminating in an efficient manner all elements crossed, with negative volume, wrong aspect ratio or not satisfying various geometrical criteria like size or shape. The main advantages are the computational efficiency compared with a layered approach, and the optimal connection of the semi-structured elements. However, as noted in Löhner and Cebral [4], it is very difficult to find a point normal that will create a growth curve of points visible from all the surrounding faces for complicated geometries, where various concave and convex ridges concur, as discussed in the introduction. The only alternative is to introduce multiple normals per point as in Löhner and Cebral [4]. As the semi-structured procedure needs only one normal per point, extra degenerated faces must be introduced to comply with the topological constraint of the semi-structured procedure.

A pre-processing phase of the proposed procedure relies on the detection and the analysis of the nature of a *ridge*, i.e. whether the ridge is concave or convex. Indeed, the edge is the smallest entity to evaluate convexity or concavity as it is shared by two faces whose scalar product of normals will give the value of the dihedral angle. To produce a good quality mesh around a convex edge, normals must be added as explained before, whereas an averaging process is sufficient for a concave edge. The edges of the surface triangulation are then scanned, and as in Löhner [18], a simple scalar product of the normals of the faces sharing the edge identifies the ridge, given a user-defined tolerance (usually around 30°). It must be noted that even the edges belonging to the same geometrical patches are candidates to be ridges, as the geometrical patches may not have been meshed in a smooth manner, and the boundary layer mesh will be extruded from the discrete data surface. The identified ridges are stored in an array with the information of their endpoints and the two faces that share the ridges, oriented according to the external normal.

Then, each ridge is analyzed separately to find if it is a concave or a convex ridge. A simple averaging process is used in a first attempt. In more complex cases such as the trailing edge of a wing, normals of the faces sharing the ridge are opposite so that the simple averaging process is inconclusive. A more detailed analysis is then performed to guarantee a reliable assessment of the nature of the ridge. An expected angle of around 30^{o} has been fixed. If the ridge is convex, the dihedral angle formed by the faces sharing the ridge is compared with the expected angle. A greater value of the analysed angle means that smaller angles between the faces are expected, which is reached by generating extra faces for this ridge. The ratio of the analysed angle by the

DOI: 10.1002/nme

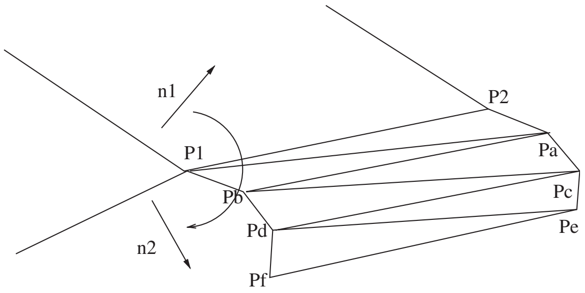

Figure 3. A strap of extruded virtual faces generated by a convex ridge.

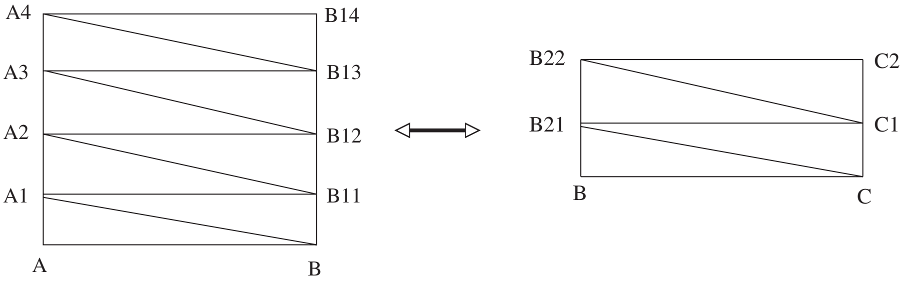

Figure 4. Two convex ridges (AB) and (BC) with their strap of virtual faces. The straps abut on B where they are not conforming.

expected angle gives the number of faces to add. Exactly, two triangles are generated for each face according to a fixed pattern, and the normal of the faces is assigned to both triangles. It may be easier to think about those extra faces as straps of mesh wrapping the ridges, but whose extremity and sides are free as represented in Figure 3. The base of the strap is created by the edge points and the rest of the strap is constituted by newly generated points. If the strap, once made conforming, is completely folded up on the ridge, it will generate well-shaped prisms due to the divergence of the normals that carry the prism. The elements of null volume will be deleted in a subsequent step after the boundary layer mesh has been generated. Nothing is done if the ridge is concave. At the end of the strap generation process, each ridge has two pointers to the beginning and the ending of the strap faces associated with it. In the concave case, these pointers are the same as no extra faces have been generated.

Until now, the procedure is classical and relies on well-known pre-processing techniques. The main difficulty, and the main task of this paper is *to make the mesh conforming at the added nodes*. At this point, the original triangular surface mesh has been completed with extra faces whose convex ridges verify a good dihedral angle between the shared faces but they are evidently not conforming as the faces were generated independently, relying only on the dihedral measure. Figure 4 shows a sketch of two convex ridges (AB) and (BC) with their strap of virtual faces. The two ridges have initially point B in common, but due to the introduction of the virtual faces and the creation of new points, both straps are not conforming anymore. A final loop over the points is necessary to make the surface triangulation conforming at points. The following two sections

DOI: 10.1002/nme

describe how to obtain a conforming surface triangulation by using a taxonomy of topological cases met around ridges and corners.

3. POINT CLASSIFICATION

As the non-conformity appears at the junction of the ridges, i.e. at the vertices, it is useful to classify the points depending on the number of ridges touching them. Following Löhner [18] and Borouchaki *et al.* [19]:

- cusp points are points touched by exactly one ridge;
- ridge points are points touched by exactly two ridges;
- corner points are points touched by more than two ridges.

A more fundamental reason to classify the ridges as described above appears in the context of the virtual faces. As all the convex ridges create straps of triangles forming 'tubes' around the ridges, only the non-conforming virtual edges appear at the end of those tubes. They create non-connected semi-circles. If only one convex ridge is touching the node, this line cannot be glued with anything. If two convex ridges appear at the node, they must be glued together as they represent two lines on the extruded surface. If more than two convex ridges concur at a node, a contour of a virtual patch is created at this point and this patch must be meshed independently.

The first two cases are presented in detail thereafter. The third case will be presented separately in Section 4 due to its greater complexity. Only the topological aspect is commented in the following two sections. The geometrical aspect of normal computation will be described in Section 5.

3.1. Cusp point

Cusp points are points touched by exactly one ridge. This ridge could be concave or convex. In the former, nothing is done from a topological point of view. In the latter, a strap of extra triangles has been generated previously, which cannot be glued with any other strap as the point is a cusp. The only way to make the mesh conforming is to delete the triangles on the same side of the cusp point, and to renumber the point in the triangles on the other side of the cusp point. This procedure is described in Figure 5, where the whole strap has been depicted as it is saved in memory. The marked faces and the previously newly generated points must be deleted thereafter.

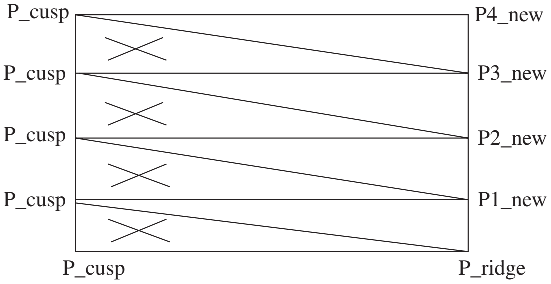

Figure 5. Configuration of the virtual faces for a convex ridge, which abut on a cusp point.

DOI: 10.1002/nme

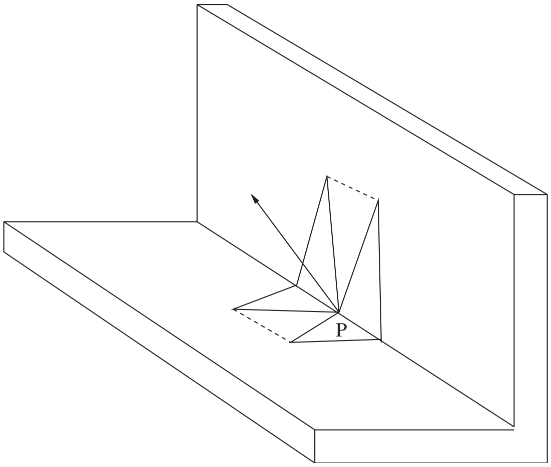

Figure 6. Configuration around a concave–concave ridge point P. The surfaces carrying both the ridges are depicted with solid line.

3.2. Ridge point

Ridge points are points touched by exactly two ridges. These ridges could be both concave, convex and concave, or both convex.

3.2.1. Concave–concave ridge point. This is the easiest case. Nothing needs to be done at the topological level as no extra virtual face has been generated. This case is illustrated in Figure 6 where the surfaces carrying both ridges are depicted with solid line.

3.2.2. Concave–convex ridge point. In this case, the strap of the convex ridge cannot be glued to any neighboring faces, virtual or not. The previous cusp procedure is then applied to it, so that the virtual faces of the convex ridge on the side of the concave–convex ridge point are deleted, as are the new points previously created. The other convex–concave combination is only a reordering of the ridges met at the points and is treated exactly the same manner as before. This case is illustrated in Figure 7 where edge PA has not been identified as a ridge, but edge PB is a convex ridge and edge PC is a concave ridge.

3.2.3. Convex–convex ridge point. This is the most complicated case in the ridge point classification. We dispose off two straps of mesh to glue along the sides. The free end point must also be glued with the already generated surface mesh. Two subcases must be discerned if the number of extra faces generated for each convex ridge is the same or not. If the number of faces is the same, which typically illustrates what happens along a wing ridge as depicted in Figure 8, the procedure consists in gluing along the common side both straps by copying the nodes of one strap on the other, deleting the unnecessary points, and finally attributing the end node of the side to the old triangles sharing this node. In order to produce a well-balanced surface mesh, ridges are glued depending on their orientation so that the same diagonal pattern is obtained along one ridge. If a standard normal smoothing process is performed around a newly generated point of the strap, the

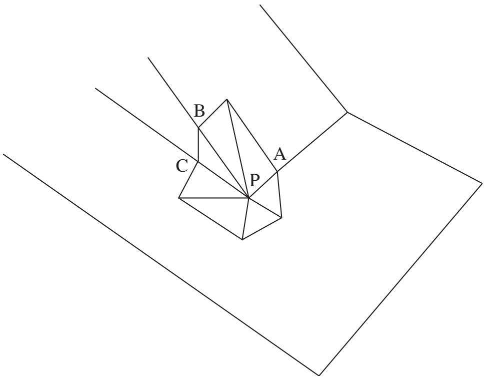

Figure 7. Configuration around a concave–convex ridge point P. Edge PA has not been identified as a ridge.

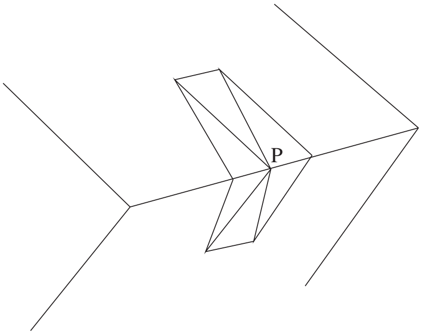

Figure 8. Configuration around a convex–convex ridge point P. All the surfaces around P are displayed.

number of faces surrounding this point will be well balanced, equal to six, so that the weight of each face will be represented accurately.

If the face number is not the same, two cases appear, whether the faces on the strap with the maximal number of faces have to be deleted, or the faces on the strap with the minimal number of faces have to be added. As the data structure relies on points pointing to ridges pointing to new faces, it is more appropriate to delete faces at this stage. A simple post processing will be designed to add faces once the surface mesh becomes conforming, as will be seen later. As a criterion to delete the faces on the strap with the largest number of faces, the maximal scalar product for each face of the strap with the largest number against the other strap is stored. With an insert-sort procedure, the faces of the strap with the largest number of faces are ordered in respect to the maximal value of the scalar product previously computed, and these faces are marked locally until the number of faces is the same as the one of the strap with the smallest number of faces. Then the previous gluing procedure is applied by taking into account whether the faces have been marked for deletion or not. It has been found experimentally that this criterion minimizes the distortion between both ridges.

DOI: 10.1002/nme

4. CORNER POINTS

The different cases presented in this paper are ordered depending on the number of ridges touching the points, which also give them an increased complexity due to the larger possibility of configurations. In this section, corner points of an arbitrary number of convex and concave ridges concurring on this point are considered. Once such a point is identified, the first ridge attached to this point is considered, and all the ridges are then visited in a counterclockwise orientation by relying on the neighbor information of the old surface triangulation. All the ridges around the corner, and all the faces bounded by these ridges are then ordered, so that discrete patches can be identified around the corner. If convex ridges are met, the newly generated points from the strap on the corner side are stored.

As already explained, the main difficulty consists in conforming the extra straps of faces. Therefore, classification of the corner points relying mainly on the number of convex ridges is proposed. The corners are classified in various subcases:

- There are no convex ridges and only concave ridges.
- There is one convex ridge and an arbitrary number of concave ridges.
- There are two convex ridges and an arbitrary number of concave ridges.
- There is an arbitrary number of convex and concave ridges.

As mentioned previously, this classification reflects the fact that the sides left to be conformed by the virtual faces are none for the first case, a point for the second case, a line for the third case and a contour for the last case, though as will be seen, the second case will be handled differently.

4.1. No convex ridges

In this case, the corner is true concave. Once again for the purely concave cases, nothing has to be done on the topological side. This configuration is illustrated in Figure 9.

4.2. One convex ridge

In this case, more than two ridges concur at one point, and exactly one ridge is convex. The extra faces on the convex ridge must conform with the surface triangulation. As opposed to the cusp point, a corner point presents various geometrical discontinuities, so that it would be expeditive

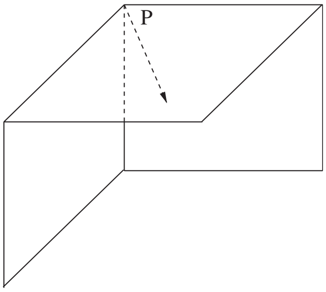

Figure 9. A concave corner point P.

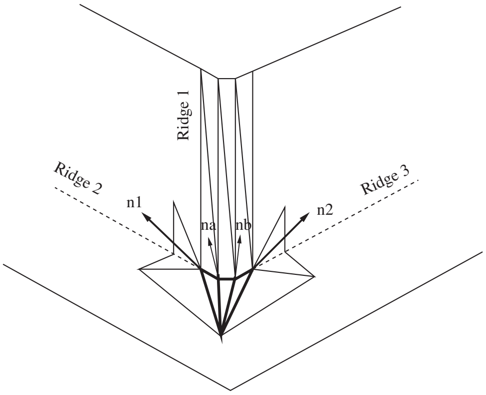

Figure 10. A corner with one convex ridge with its virtual faces, and two concave ridges displayed as dashed lines. The virtual faces have been extruded off their point normals for the sake of clarity.

to simply delete the extra faces as in the case of the cusp point. Therefore, instead of deleting the extra faces, other extra faces are added to the surface triangulation to make them conforming.

Let us suppose that appropriate normals have been computed for all the points surrounding the corner point and the newly created points of the virtual faces. The situation is shown in Figure 10, where **n1** and **n2** are the extreme normals, and **na** and **nb** are the point normals of the newly created points. The idea consists in trying to open the edges around the corner point by introducing new virtual faces in between, so that the virtual faces of the convex ridge will share an edge with those new virtual faces. In Figure 10, the convex ridge carries six triangles with three non-conforming edges on the side of the corner point, so that three new faces represented in bold lines in the figure must be created to make the whole triangulation conforming. The constrain relies on the fact that the newly created faces must have good geometrical properties, mainly they cannot be inverted when extruded along the point normals.

Various criteria have been tested. The most robust one, even for coarse meshes with significant noise in the surface triangulation due to discrete data [18–20], is borrowed from the discrete surface regridding as in Löhner [18], and consists in relying on the normals of the virtual faces to check for visibility. For each virtual face, and referring to Figure 10, the edge between ridges 2 and 3 whose projection on a plane normal to the convex ridge is the most aligned with the virtual face normal will be the one opened. Once a valid triangulation has been obtained based on the previous criterion, an optimization loop is performed on the faces surrounding the corner point based on a quality criterion or a volume criterion involving the point normals.

However, it is possible that various corners of such a type may be in the same neighborhood due to a coarse surface mesh. It is therefore necessary to maintain a bookkeeping information for the newly introduced faces, so that the correct information is used if these edges must be opened once again by another corner of this type, as illustrated in Figure 11. In this figure, ridge 1 is connected to two corner points and ridge 2 is connected to one corner point and one ridge point. Each oriented ridge carries the information of its two points and the two surfaces that carry it. Furthermore, the neighbor to neighbor information has been obtained from the initial triangulation. By opening the surface triangulation, the neighbor to neighbor information must be updated as a new virtual face has been introduced. If an opened edge connects two corners of this type, it may

DOI: 10.1002/nme

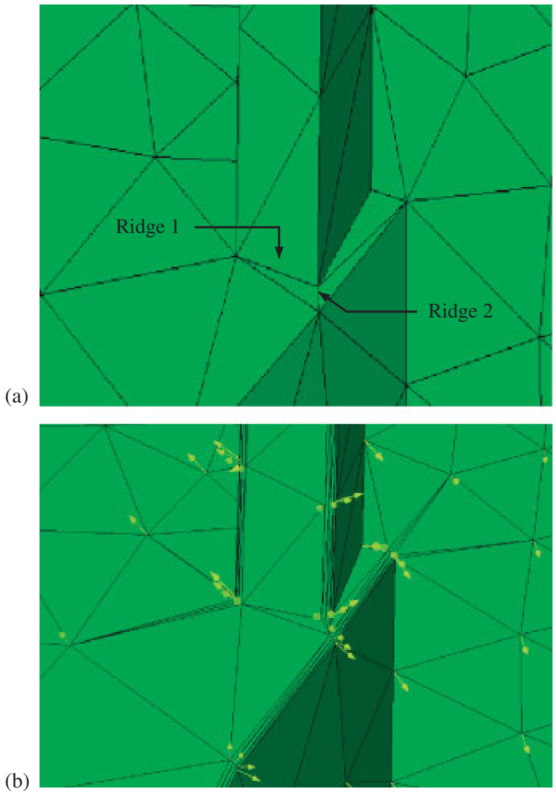

Figure 11. Various neighboring corners with one convex ridge and various concave ridges: (a) geometry and (b) extruded surface.

well be possible that the second corner tries to open it too. This is why the neighbor to neighbor information has to be updated. Likewise, it may be possible that virtual faces be introduced by opening a ridge, so that the information carried by the ridge is not valid anymore. An additional check has then to be performed.

It is possible that the points involved in the extrusion process may also have multiple normals. A procedure must be triggered at this point to compute the normal that corresponds to the current patch. This is done by iteratively adding faces that share the edge between the corner point and the point of the patch, turning around the latter in both directions, and testing that each added edge around this point be not already detected as a ridge as illustrated in Figure 12. Therefore, it is another reason to maintain a consistent relationship between the ridges pointing to the faces they are carried by, and the newly introduced faces as mentioned before.

Finally, for the same case, there are geometries that do not allow the introduction of all the virtual faces around the corners with one convex ridge, because the created faces have negative areas when projected locally. A mechanism must be available to delete the unused faces and update the information so that the normal procedure with swapping improvements may be applied later on.

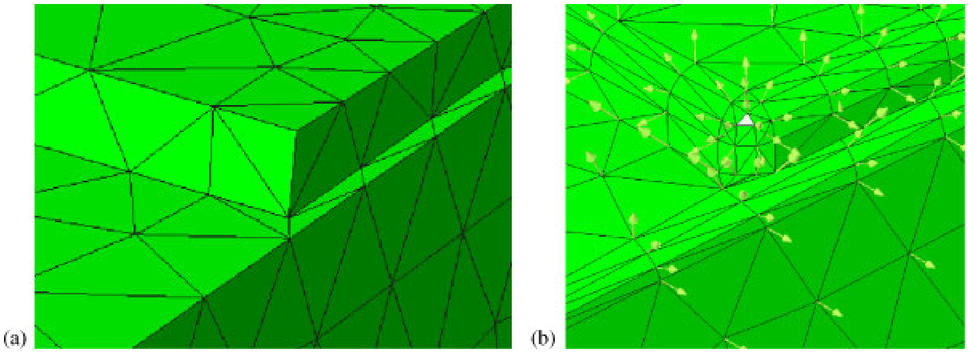

Figure 12. A corner with one convex ridge and various concave ridges connected to various ridge points. On the right, the surface has been overly extruded for the sake of clarity so that crossings appear in narrow gaps: (a) geometry and (b) extruded surface.

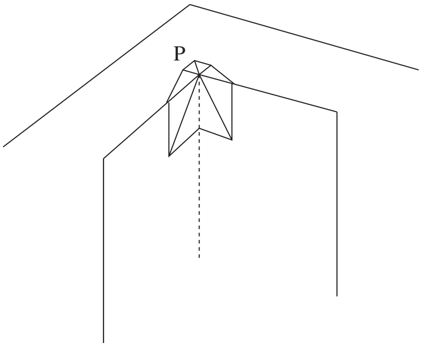

Figure 13. A corner point P with two convex and one concave ridges. The concave ridge is displayed with a dashed line.

4.3. Two convex ridges

This case is similar to the ridge point case, as is the previous case from the cusp point case. However, here, concavity must be taken into account. The same general procedure is performed. Both convex ridges must be glued along their common side. Depending on the number of faces per strap, faces are deleted or not, relying on the same criterion as the one used for the ridge point case. This case is illustrated in Figure 13.

4.4. More than two convex ridges

This is the most general and the most complicated part of the proposed procedure. Various convex and concave ridges meet at one point. Two cases have been discerned, depending on the fact that some concave ridges are present or not. Once more, this classification only relies on the nature of the topology met, as some crossing of the virtual patch may appear in case of concave ridges, degenerating a surface contour to a line, as explained below.

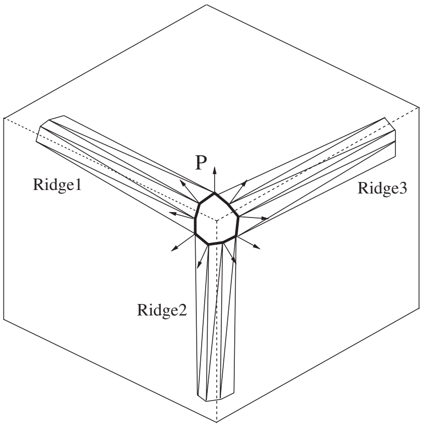

Figure 14. Configuration around a corner point P with three convex ridges and no concave ridge. The virtual faces have been extruded around their respective convex ridge.

4.4.1. Only convex ridges. The problem at hand consists in joining various extra faces. As illustrated in Figure 14, each strap associated with each convex ridge must be glued with the previous and the next strap, which is done during the turn around the corner in a counterclockwise manner. For the sake of clarity, the same number of faces for each convex ridge has been drawn, whereas it is obviously arbitrary in general. All the normals are emanating from the same point so that, as the normal length is one in this case, the locus of all the contour points is a sphere centered on the corner point. Then, the hole left by the contour created by all the straps must be meshed. First of all, inner points are created based on the virtual edges created between all the contour points against all the contour points. The points (or normals) that are too close from each other are filtered, first from the boundary points, and then all against all. After having computed the average normal, and having defined a local tangent plane, the newly created points are projected on the tangent plane and a two-dimensional mesh is obtained. The two-dimensional mesh is then pulled back in the three-dimensional space and the corner mesh is obtained.

However, it is possible for twisted corners that all the inner points have been deleted due to proximity reasons, but that the two-dimensional mesh generator prefers to insert a new point in the two-dimensional configuration. As this new point has not been considered in the inner point generation, it is not associated with a three-dimensional normal. The option considered here is to take the sum of all the boundary point normals in three dimensions, weighted by the two-dimensional distance from the newly inserted point to the boundary points in two dimensions, to recover a three-dimensional normal for the newly introduced point.

4.4.2. Some concave ridges. The main difficulty with concave ridges is due to the fact that the normals at the end of two successive convex ridges separated by a concave ridge do not coincide as depicted in Figure 15. This kind of configuration is frequently met at the intersection between the wings and the body of a plane. More precisely, all the normals come from the same point so that the strap of normals represented in Figure 15 has been translated along each ridge axis for the sake of clarity. Figure 16 depicts what happens in the real space for two convex ridges, one with six faces attached, the other with four faces, and one concave ridge. If a projection on the tangent

DOI: 10.1002/nme

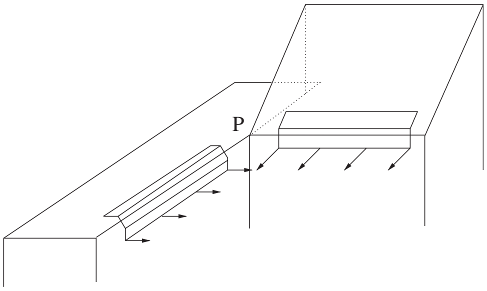

Figure 15. A corner point P with three convex ridges and two concave ridges. The virtual faces of only two convex ridges with some normals are displayed to show the difficulty.

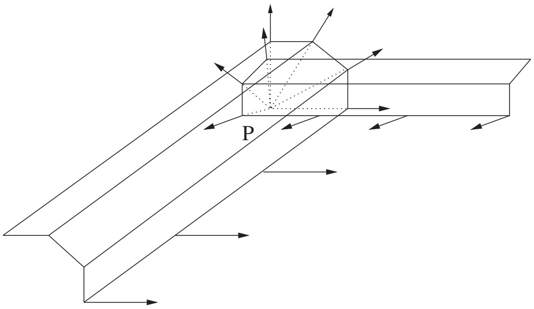

Figure 16. Normal incompatibility around point P with two convex ridges separated by one concave ridge due to the concave ridge as illustrated in Figure 15.

plane is performed, the contour is crossed, and the two-dimensional mesh contour is invalid. This example is not a corner point as only two convex ridges meet at this point, but has been chosen to illustrate the mechanism of the difficulty associated with concave ridges.

Here, the idea consists in gluing the convex ridges along the possible multiple concave ridges until the face normals of the convex ridges coincide. An example is presented in Figure 17 where it is seen that each strip of each convex ridge must be glued from the concave edge to the extreme point, controlling the gluing process by the scalar product between each face. Pointers from points to faces sharing these points must be added and updated as a single point may be glued by various faces at the same time. In the worst case, each concave ridge could be glued twice by two consecutive concave ridges, so that the key point of the procedure relies on comparing the number of available faces for two convex ridges separated by concave ridges, taking into account whether one or both of the considered convex ridges have already been glued before on the other side, and maintaining conformity, as much between the internal faces of the convex ridges as with the external faces of the old surface triangulation.

Point normals are assigned according to the same idea as in the previous section, along the sides created by collapsing the convex ridges, relying on the faces of the old triangulation. A schematic representation that has been found useful is displayed in Figures 18 and 19, where a point with five ridges, three convex and two concave, has been drawn. At the beginning of the procedure, the virtual contour of the corner is known and ordered counterclockwise. The concave corners

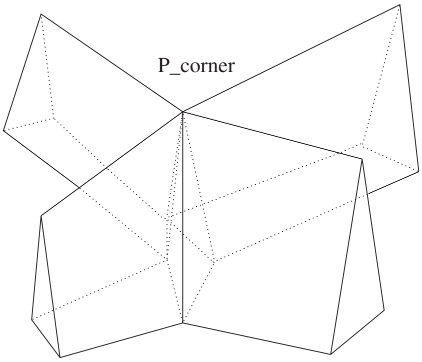

Figure 17. A succession of convex and concave ridges.

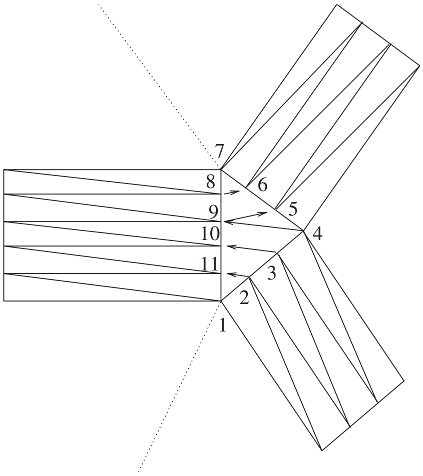

Figure 18. A two-dimensional representation of a corner with two concave and three convex ridges. Concave ridges are represented with dashed lines. Only virtual faces are represented.

are displayed with dashed lines. For each concave ridge, both the convex ridges on each side of the concave ridge are glued together with their respective points. A possible pattern is presented where the ridge with points $\{1, 2, 3, 4\}$ is glued to the ridge with points $\{1, 11, 10, 9, 8, 7\}$. Then this ridge is glued to the one with the points $\{7, 6, 5, 4\}$.

Once all the concave ridges have been eliminated, an analysis of the convex ridges is performed to check whether their faces have all been glued, have been partially used, or have not been used at all. Depending on the number of ridges partially used, the same classification as before is performed. If only one convex ridge still has some faces unused, a cusp procedure is applied. If two convex ridges still have some faces unused, a ridge procedure is applied. If more than two convex ridges still appear, the previous corner procedure is applied. In the example of Figure 19, only one face of one ridge is left unused, so that it will be deleted as a cusp procedure.

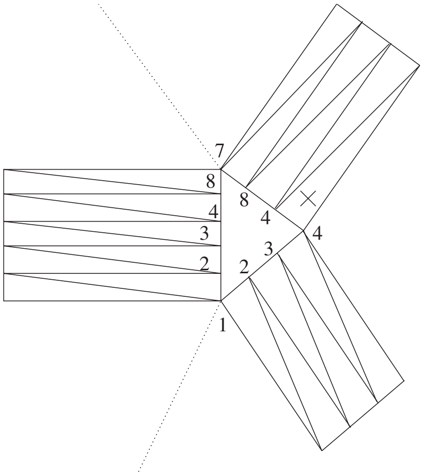

Figure 19. The glued faces after the procedure. The hole does not exist anymore.

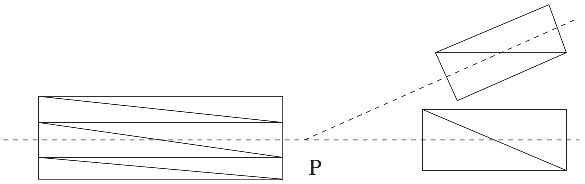

Figure 20. A flat corner with three convex ridges: the angle between both the horizontal ridges is zero so that the contour is crossed.

4.4.3. Degenerated cases. As already mentioned, the concavity may result in an invalid contour of the virtual patch in the real space. Another case gives the same problem. Even supposing that only convex ridges are connected to a point, these convex ridges create valid sides only if the convex ridges are not aligned, in which case two sides collapse in one line, and close the contour. This is the main reason why the ridge points are glued. This may happen with various convex and even concave corners as depicted in Figure 20. The three convex ridges are depicted with dashed lines, and the virtual faces are superimposed on them. The angle between both the horizontal ridges is zero so that the contour is not a surface anymore or is crossed. In this case, all convex ridges must be glued together. All convex ridges are divided into two categories depending on the angle formed with the two opposite ridges. If the angle is within a given tolerance they belong to the group of one of the opposite ridges. If not, a cusp procedure is applied. Once the two main groups are created, they are glued together, in the same manner as a ridge procedure. The subroutine responsible for this is the `checkflatcorner` subroutine in the final algorithm.

Another case to be taken into account arises when two ridges concur at the same point with an important planar angle. It is a classical situation when thin volumes are created in the geometry such as wings or missile stabilizers. This case is very near to a corner point with only convex ridges, and new points are projected on the tangent plane and a two-dimensional mesh is obtained

DOI: 10.1002/nme

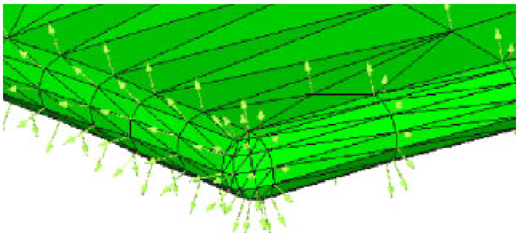

Figure 21. Configuration around a ridge point with large dihedral angles.

as previously and projected back in three dimensions as illustrated in Figure 21. The subroutine responsible for this is the `ridgcorner` subroutine in the final algorithm.

5. NORMAL SMOOTHING

The method described in this paper uses the direction given by the point normals to extrude the prisms divided into tetrahedra from the surface triangulation. Therefore, the normal computation is a crucial step in the method, especially for corners and ridges where the visibility cone is extremely limited. To take concavity into account, the modulus of the normals is also considered, as shown with a simple two-dimensional concave corner in Figure 22.

The point normal computation difficulty ranges from a corner point where the geometry of the surrounding faces leaves almost no space for freedom to a smooth point without any ridge, where the visibility is optimal. For difficult points, a small procedure has been designed in Aubry and Löhner [16] to compute an optimal normal direction so that the maximal angle created between the point normal and the normal of the faces surrounding the point is minimized. This is the basic procedure used for all the difficult topologies. To move forward, a B-rep representation has been chosen to progressively transfer the information in a smooth manner from the difficult point normals to the line normals to the surface normals. Each case is presented independently for the corner points:

- For a point with only concave ridges, 'the most normal' normal is computed and nothing more needs to be done as illustrated in Figure 9.
- For a point with one convex ridge and more than one concave ridge, the convex ridge will give rise to a strap of virtual faces that will abruptly stop at the corner point as shown in Figure 10. Therefore, the boundary between the virtual faces and the old faces is a line. The normal of points at the extremities of this line are imposed with the 'most normal' normal. For the intermediate points, a first rough normal is computed by a standard average of the surrounding faces. Then, a cubic spline interpolation is performed, which is used to compute normals for the intermediate points so that the normals are regularly distributed.
- For a point with two convex ridges and at least one concave ridge, the two convex ridges will abut at this point as displayed in Figure 23, creating almost a ridge point. However, the added concave ridges strengthen the complexity. Therefore, the normals at each extremity of the line joining both the ridges are imposed with the 'most normal' normal, and then the standard average is used for the intermediate points. The cubic spline interpolation is used as before. This line is displayed in bold in Figure 23.

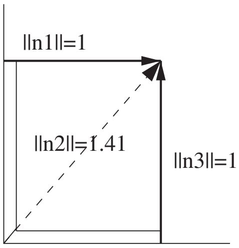

Figure 22. A concave corner in two dimensions with a normal of modulus different from unity.

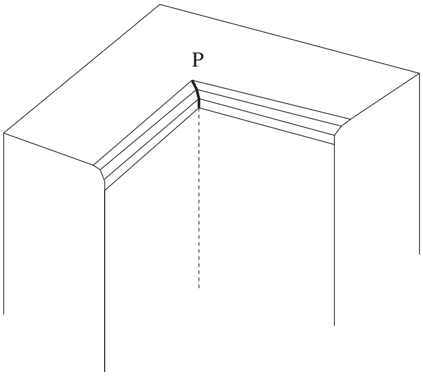

Figure 23. A corner point P with two convex and one concave ridge. The concave ridge is displayed with a dashed line. The bold line represents the line where the point normals have been imposed.

- For a point with more than two convex ridges but only convex ridges, these will create the boundary sides of a hole to be meshed as explained before and illustrated in Figure 14. Once again, the extreme points of these lines are computed with the ‘most normal’ normal relying only on the knowledge of the old faces, and the intermediate points are computed without the spline as the geometry is already known in this case to be circular.
- For a point with more than two convex ridges and at least one concave ridge, the virtual faces are glued along the concave ridges as explained before. Those glued lines receive the same treatment as before, namely, the point normal at the extremities are imposed with the ‘most normal’ normal, then a rough average is computed and then the point normals are smoothed with the cubic spline, and distributed regularly.

The concave ridge point normals and the convex/concave ridge points are also imposed with the ‘most normal’ normal.

Once the difficult point normals have been obtained, the normals not already imposed at the points belonging to the lines separating the virtual faces from the old faces are computed with a standard average by taking into account only part of the old faces connected to them that have been divided with the introduction of the virtual faces. Then, all those imposed point normals are used as boundary conditions for the global smoother also performed with the ‘most normal’ normal procedure. More details may be found in Aubry and Löhner [16]. Finally, a procedure is used to verify that all the boundary conditions are coherent as they were imposed independently. Therefore, it may happen that a triangle may be overconstrained, i.e. with all its point normals imposed, and presenting a large variation in direction. Corner point normals will then have a greater priority over the other point normals.

This approach has proven to generate smooth and valid normal distributions even in complex geometries. However, one has to keep in mind that the normal distribution could only be a compromise between a global and a local optimal. Locally, the point normals are restricted by the faces surrounding the points, which provide a first-order information. It may however happen that two neighboring point normals have to be smoothed because even though they are optimal locally, they present a too large variation globally.

6. PLANES OF SYMMETRY

If the geometry has a plane of symmetry, it is obviously interesting in order to save CPU time to restrict the geometry to a half, applying appropriate boundary conditions at the symmetry plane. For the boundary layer extrusion, it means that some wetted and some non-wetted faces will share common edges and points along the symmetry plane. A simple sketch of the problem is provided in Figure 24. An obvious requirement is that the normals at the points belonging to the intersection between wetted and non-wetted surfaces are in the symmetry plane. Although not necessary, it would also be advantageous if the procedure was to produce the same results with or without the symmetry plane.

A first approach would consist in extending the previous taxonomy, taking into account in each case whether wetted or non-wetted faces are present. Typically, a true ridge on the geometry would give rise to a ridge point at each end point. Owing to the plane of symmetry however, two new concave ridges appear that promote the ridge point to a corner point with wetted and non-wetted

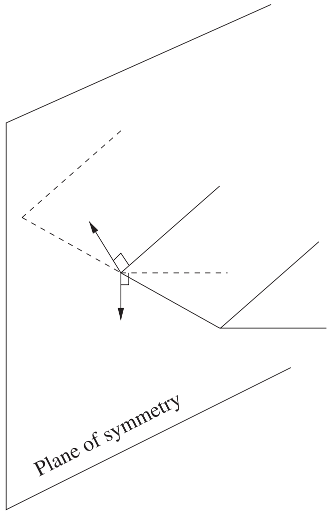

Figure 24. The convex ridge of the trailing edge intersecting the plane of symmetry.

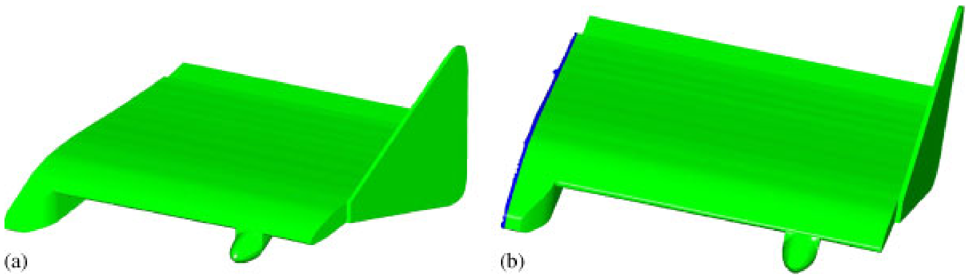

Figure 25. A symmetric MITE: (a) initial surface triangulation and (b) surface with added faces along the wetted/non-wetted line.

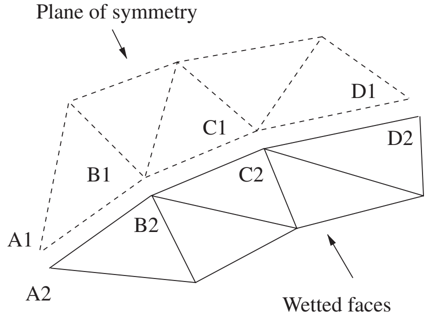

Figure 26. The points of the line separating the wetted faces from the faces belonging to the plane of symmetry are duplicated to disconnect these faces. The points appear to be moved for the sake of clarity but all pairs of points $\{A1, A2\}$, $\{B1, B2\}$, $\{C1, C2\}$, and $\{D1, D2\}$ have the same coordinates.

faces as illustrated in Figure 10 where the wetted faces are between ridges 2 and 3. Although possible, this approach increases the burden of coding as all the cases of the previous taxonomy must now take into account whether some faces are wetted or not.

Instead, another path has been explored. To illustrate the procedure, Figure 25(a) depicts the original triangulation of the wetted faces of a micro tactical expandable (MITE) [21]. Owing to symmetry reasons, only half has been used for the computation. As one more layer of triangles is needed to define the normals at the intersection of the wetted/non-wetted faces, the solution consists in artificially introducing the symmetric triangles of the first layer of wetted triangles at the intersection wetted/non-wetted by reflecting them with respect to the symmetry plane as illustrated in Figure 25(b). This is performed in various steps:

- The direct introduction of these faces would produce a non-conforming or non-manifold surface triangulation at the wetted/non-wetted face intersection. In order to have a valid surface discretization, the points along the lines separating the wetted faces from the non-wetted faces must be disconnected. New points with the same coordinates as the points along these lines are introduced. The wetted and non-wetted faces are then disconnected, as illustrated in Figure 26.

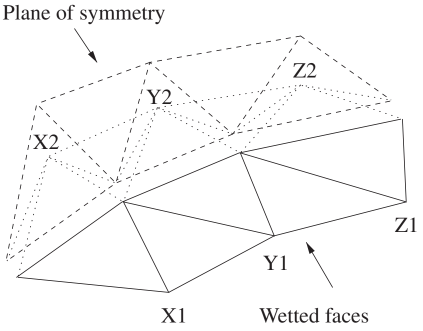

Figure 27. A new generation of points $\{X2, Y2, Z2\}$ has been created by mirroring the points $\{X1, Y1, Z1\}$ with respect to the plane of symmetry. New triangles are introduced such as $\{A2, B2, X2\}$ and $\{A1, X2, B1\}$ where the first is wetted and the second is not. The points of the line separating the wetted faces from the faces belonging to the plane of symmetry are displayed only in Figure 26 for the sake of clarity.

- Two identical triangles with a layer of new points on the other side of the symmetry plane reflecting a triangle on the wetted surfaces are introduced and connected to the non-wetted faces on the top and to the wetted faces on the bottom by one or two points. The face at the bottom is marked as wetted, the face at the top as non-wetted. This procedure is illustrated in Figure 27 where the points $\{X2, Y2, Z2\}$ are the image of the points $\{X1, Y1, Z1\}$ with respect to the plane of symmetry. The points on the separating line have not been displayed as in Figure 26. Referring to both figures, two new triangles $\{A2, B2, X2\}$ and $\{A1, X2, B1\}$ have been created where the first is wetted and the second is not.
- Finally, the points introduced in the last step without the coordinate value are mirrored with respect to the symmetry plane to produce the desired geometry.

Now, the ‘standard’ procedure is performed on this modified initial surface triangulation. The difference from the previous approach is that, due to the insertion of the added faces, a ridge point on the symmetry plane is not promoted to a corner point, so that the same operations are performed with or without the symmetry plane. Furthermore, as the faces surrounding the points along the wetted/non-wetted interface are symmetric, the normals will naturally be in the symmetry plane. In order to avoid the creation of ridges by the added faces, ridges are accepted only between wetted faces. Even though the added faces have been mirrored from the symmetric plane, floating point errors may result in a different evaluation in the angle of a true ridge of the geometry, and the angle produced by the symmetrical ridge added to the added faces. To be sure that the number of virtual faces is the same in both the cases, the symmetric ridges are linked so that the added ridge contains exactly the same number of virtual faces as its symmetrical original counterpart. Finally, the sides must be updated as the virtual faces touching the symmetry plane will generate new sides for the surface defined on the symmetry plane.

It must be noted however that special procedures must be added for extremely unusual cases. If a corner point appears on the symmetry plane, an operator able to split this corner in a symmetric manner must be designed. The same is true for a ridge belonging to the symmetry plane. Although technical, these operators do not seem to be a limit to the method, emphasizing that these cases belong to pathological or degenerated geometries.

7. PERIODIC BOUNDARY CONDITIONS

The periodic boundary conditions present in some cases relevant advantages such as important CPU time savings and theoretical fundamental studies compared with classical conditions and are often met in practice. From an implementation point of view, the basic idea is an extension of the previous case for the symmetry plane. However, for the periodic boundary condition there is no reason for the periodic surface to be planar and the point normal on the periodic plane to be in this plane. Furthermore, faces must be added on both periodic surfaces of the geometry.

To illustrate the procedure, a simple sketch is shown in Figure 28. A master and a slave surface is defined so that the periodic surfaces and lines are completely linked. Now the procedure consists in generating extra triangles with the appropriate periodic definition (translation, rotation, etc.) on one side of the master surface, which duplicate the wetted triangles on the other side, and performing the same for the slave surface. The first part of the procedure is the same as the one described in the previous part with disconnection of the points along the wetted/non-wetted interface, except that now the interface divides the wetted from the periodic surfaces. Then, new triangles on the outer side of the master surface are created to mimic the wetted triangles touching the slave surface, and new triangles are generated on the outer side of the slave surface to mimic the wetted triangles touching the master surface, where the order between the master and slave surface is arbitrary. As before, a second generation of new points is performed, where the first generation was used to disconnect the points along the interface. Finally, the periodic definition is used to give coordinate values to this second generation of points on the outer sides of both the master and the slave surfaces. During the ridge detection, ridges of the newly added faces are linked to the wetted ridges to match exactly the number of added virtual faces. Finally, the sides are updated due to the insertion of the virtual faces.

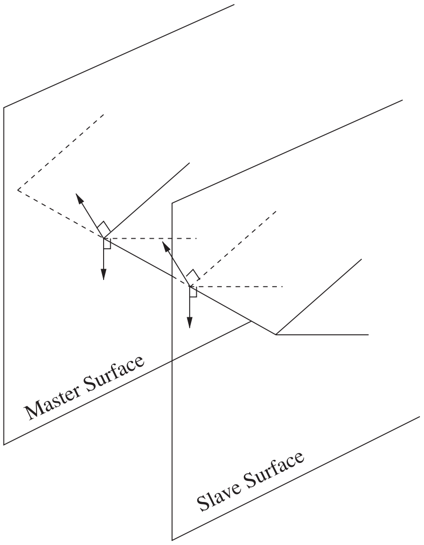

Figure 28. The convex ridge of the trailing edge intersecting the master and the slave surface. The master and the slave surface may be of arbitrary shape.

DOI: 10.1002/nme

So far, the procedure is very similar to the plane of symmetry case. However, once the point normals have been created, the boundary layer mesh volume is offset from the surface triangulation, as the method used here relies on a semi-structured approach, and prisms divided into tetrahedra are generated. The boundary layer elements will then have to pass a series of tests, as described in Löhner [17]. Some elements will be removed but the trace of the removed elements must match exactly on the master and the slave surfaces. If elements are removed in prisms, a procedure must be designed so that if some removed prisms are touching the master surface, their counterparts on the slave surface are removed too, and vice-versa. However, as was emphasized before, it is extremely important to add special elements to avoid some degenerated faces to be exposed to the isotropic advancing front generator, as advocated by Connell and Braaten [6]. Those elements will rely on the diagonal pattern on which the deleted prisms were built. One has then to ensure that all the sides on the master surface touching the wetted faces have the same diagonal orientation as their counterpart side on the slave surface. Using the algorithm proposed in Bottasso and Detomi [12], it is an easy modification to take that restriction into account.

8. OPTIMIZATION

Once the mesh has been made conforming, more information is at hand. Although the main part of the work has been done, an optimization of the virtual surface mesh appears to dramatically improve the quality of the final boundary layer mesh. Three kinds of operators have been implemented to perform this task.

The whole algorithm is summarized in Figure 29; the corner procedure in Figure 30.

```
Procedure generate multiple normal at ridges and corners:

  • Generate symmetric or periodic faces if symmetric or periodic B.C.
  • Loop over the edges of the mesh to determine the concave and convex
    ridges. If found:

      – Generate the multiple normals for each convex ridge.
      – Link the master-slave ridges in case of periodic B.C.
      – Mark the points touched by the ridge.

  • Loop over the ridges: Generate the faces associated to the normals
    for the convex ridges
  • Loop over the points:

      – If one ridge call cusp()
      – If two ridges call ridge()
      – If more than two ridges call corner()

  • Delete extra faces if periodic or symetric B.C.
  • Renumber points and faces that have not been deleted
  • Renumber and add sides if periodic or symmetric B.C.
  • Impose normal boundary conditions
  • Optimization

      – Correct large variation in faces from one ridge to another
      – Check normal boundary condition incompatibilities
      – Smooth the non imposed normals
      – Swap the virtual faces
```

Figure 29. Generation of a conforming surface triangulation with multiple normals.

```
Procedure conform the faces around a corner:

  • Initialisation: nccav=0,nccvx=0
  • Find the first ridge touching the corner
  • Loop around the corner storing ridge number, face number, and
    incrementing nccvx for each convex edge, and nccav for each concave
    edge
  • If(nccvx.eq.0)Impose normal on concave corner.
  • Else if(nccvx.eq.1) Generate extra faces. Compute point normal at
    intersection points. Optimize the mesh.
  • Else if(nccvx.eq.2) Glue both convex ridges. Compute point normals at
    intersection points.
  • Else
      – If (nccav.gt.0) call deleteccavedge()
      – Call checkflatcorner()
      – Generate internal points, tangent plane. Project and mesh on the
        tangent plane. Pull back in three dimensions
```

Figure 30. Generation of a conforming surface triangulation around a corner.

8.1. *Variation in surface number*

When gluing the faces of a convex ridge to a group of faces of another convex ridge, there may exist a large variation in the number of faces carried by each ridge, as the angle of each ridge may vary strongly, as it happens at the intersection of a wing and the body of an airplane where the ridges on the wing have a much larger dihedral angle than the body ridge, illustrated in Figure 37(d). Although the criterion described before behaves well for small variations, a bad quality mesh was generated due to the fast normal variation at the end of the ridge. To remedy this problem, points and surfaces are added along the common side. A first diagonal swapping is then performed following the algorithm proposed in Boender [22] due to the bad connection created during the point insertion. Then a second diagonal swapping based on the quality of the geometrical approximation [23] is applied to the virtual strap. The combination of the introduction of points and the diagonal swappings allows for a smoother variation from one ridge to the other.

8.2. *Surface swapping*

All the virtual straps susceptible of carrying a point with limited visibility have been previously marked for swapping based on the method proposed by Frey in [23]. This swapping is performed after the normal smoothing and has been found to be very useful in practice.

8.3. *Collapse of badly shaped virtual triangles*

The main aim of the procedure, as seen above, is to make the surface triangulation conforming after the insertion of the virtual faces. A particular emphasis is laid on the topological aspect of the triangulation, whereas the geometrical aspect is only taken into account through the normal computation. This may result in inappropriate virtual triangles due to a bad alignment of ridges, isolated ridges due to noisy surface meshes, a too large number of surfaces generated on one ridge without taking into account the vicinity, etc. This can only be corrected *a posteriori*, once the surface mesh has been made conforming, by checking the shape of the virtual triangles by extruding them along the normals and evaluating the angles or whatever triangle quality measure as proposed in Frey and George [24].

DOI: 10.1002/nme

As mentioned in George and Borouchaki [25], only two kinds of degenerated triangles exist, whether there exists a very small angle, or a very large one. In our context, very small angles exist if a large number of virtual faces have been generated relying only on the angle created by the face normals of the faces carried by the edge. However, due to a non-smooth geometry at the end of this ridge, the final angle of the two point normals at the end of the ridge may well be much smaller than the one computed previously, giving rise to thin virtual faces in the initial front, which may wreak havoc for the isotropic mesh generator. Large angles may also appear due to faces relying on neighboring ridges that are almost aligned.

The obvious solution is to get rid of these triangles as the deletion would not imply a large variation in the surface mesh as these triangles are badly shaped due to almost similar normals in their neighborhood. This is performed by collapsing the shortest edge of a badly shaped triangle.

9. RESULTS

The proposed method is illustrated with various examples roughly classified by complexity, from academic to industrial. The extruded surface mesh and the normal distribution are presented for each example with its geometry, as well as various cuts through the volume mesh. Timings are also given.

9.1. Delta wing

The surface mesh of a delta wing is displayed in Figure 31(a) with the ridges displayed in red. Part of the leading edge and the whole trailing edge are convex ridges that intersect at four places giving rise to four convex corners at the back of the wing. There are no concave ridges. Figure 31(b) shows the surface mesh with extra normals at the intersection of the leading edge and the trailing edge with the convex corner. Normals are also represented. Figure 31(c) depicts the faces left after the semi-structured and the deletion procedures. These faces will be the initial front for the isotropic mesh generator. This front is rather smooth due to the introduction of the transition elements (see Reference [9]), and the aspect ratio of the faces is good. Finally, Figure 31(d) shows a cut in the volume mesh where the semi-structured and the unstructured mesh are clearly seen. A smooth transition appears between both meshes. The final mesh contains 258 621 elements and 47 022 points and has been generated in 4 min on an Intel Pentium 4 ht in debug mode. The boundary layer contains 113 025 elements, almost half of the final mesh.

9.2. Concave/convex body

This example was presented in Sharov *et al.* [15]. All the common difficulties due to ridges and corners appear. The geometry is displayed in Figure 32 where the ridges have been pointed out. There are corners with three convex ridges, one complex corner with two concave and two convex ridges, and two corners with one concave and two convex ridges. Figure 33(a) and (b) show the surface triangulation of the inserted faces with the point normal distribution for the extruded surface triangulation. Figure 33(a) shows the most difficult corner of this geometry with two convex and two concave ridges. The point normal distribution is very smooth and connects very well the two concave ridges of the corner. All these normals are shown for a uniform surface mesh. However, even with this smooth normal distribution, some folded elements appear around the corner for the third layer of the boundary mesh giving rise to the front shown in Figure 34(a) with transition

DOI: 10.1002/nme

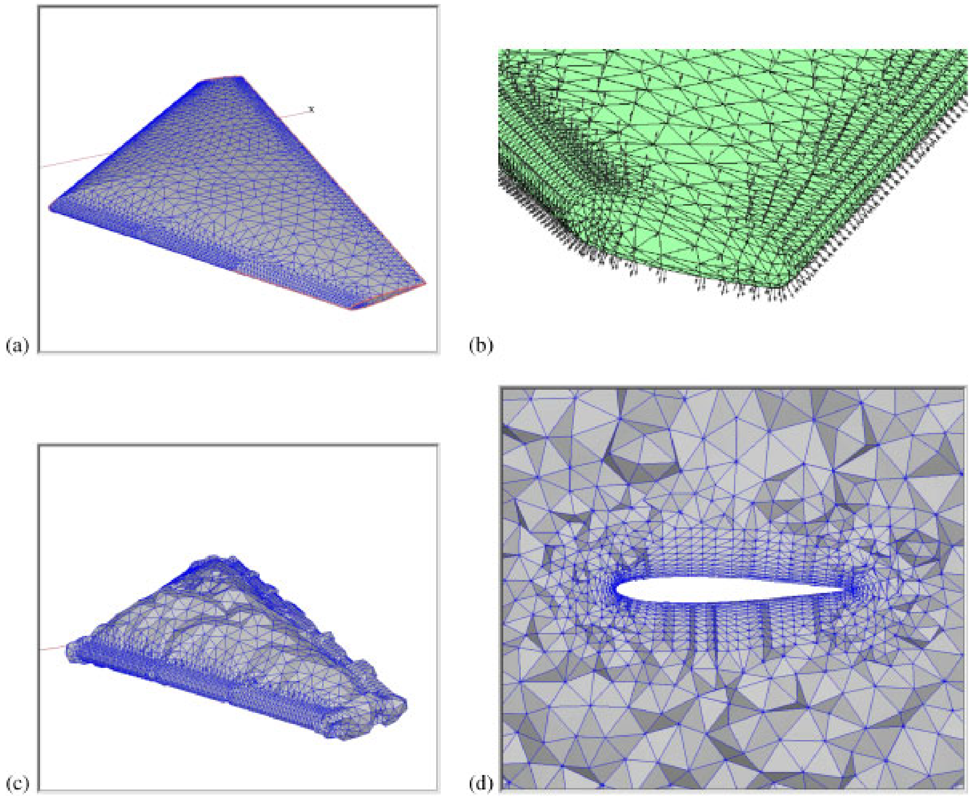

Figure 31. A delta wing: (a) surface mesh; (b) extruded surface mesh with normals; (c) front of the semi-structured mesh; and (d) cut in the volume mesh.

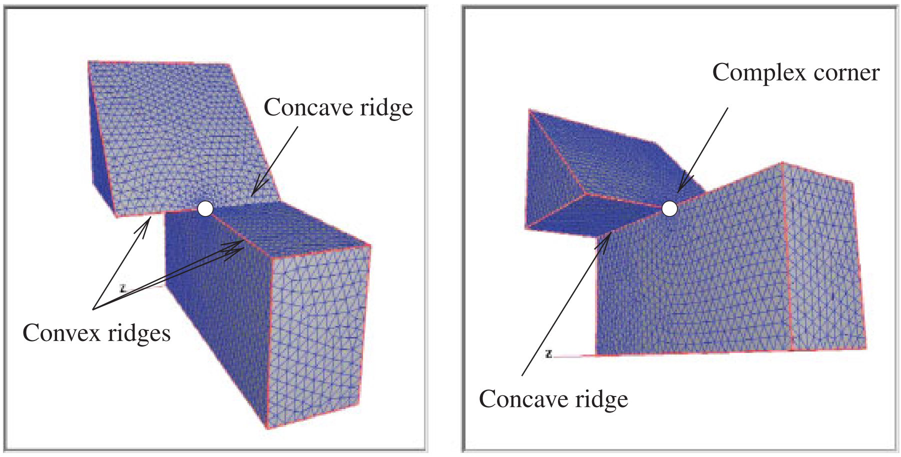

Figure 32. Geometry of the concave/convex body.

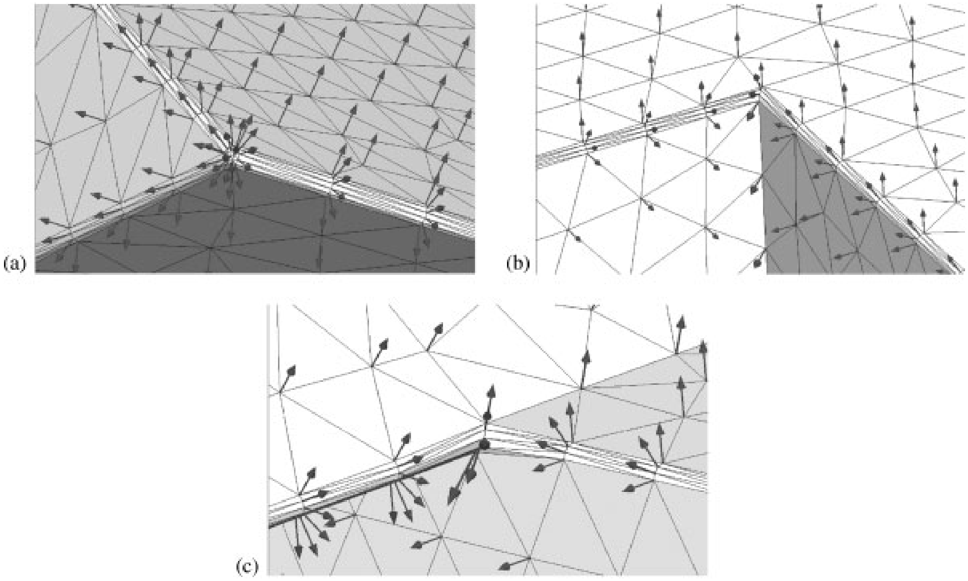

Figure 33. Normal distribution at different locations for the concave/convex body: (a) a convex corner; (b) a corner with two convex and one concave ridge; and (c) a complex corner with two convex and two concave ridges.

elements. The deletion of these elements exposes some high aspect ratio faces to the isotropic mesh generator. This phenomenon is due to the fact that the top faces of the elements emanating from the convex ridges around the complex corner are not already isotropic for such low boundary layer level.

Here, the idea proposed in References [6, 9] to add transition elements or to collapse edges to avoid high aspect ratio faces cannot avoid these small faces. The only possibility to provide a well-balanced initial front is to follow the pioneering idea of Sharov *et al.* and to refine the surface mesh around the complex corner to provide well-balanced faces after the deletion procedure. The initial front obtained around the corner with refinement is displayed in Figure 34(b) without the transition elements and in Figure 34(b) with the transition elements. The final volume mesh contains 891 835 elements and 153 104 points and was generated in 14 min in debug mode. It is displayed in Figure 34(d).

9.3. Onera M6 wing

This example is a classical benchmark for viscous flow computations. This example has also been presented in Sharov *et al.* [15] and Kallinderis and Ward [10]. Figure 35(a) represents a detail of the surface mesh at the trailing edge, and Figure 35(b) represents the same location of the inflated virtual surfaces and the convex corner with the normals used to generate the semi-structured mesh. Figure 35(c) shows a detail of the front after the deletion process and Figure 35(d) shows a cut of the volume mesh obtained. The volume mesh consists of 4.8 Mtets and 831 kpoints and required 19 min on an Intel Pentium 4 ht in debug mode. Figure 36(a) reveals a detail of the Mach number distribution, besides Figure 36(b), where the mesh has been displayed. The large variation of the Mach number at the surface of the wing has been well captured. In Figure 36(c), the pressure

DOI: 10.1002/nme

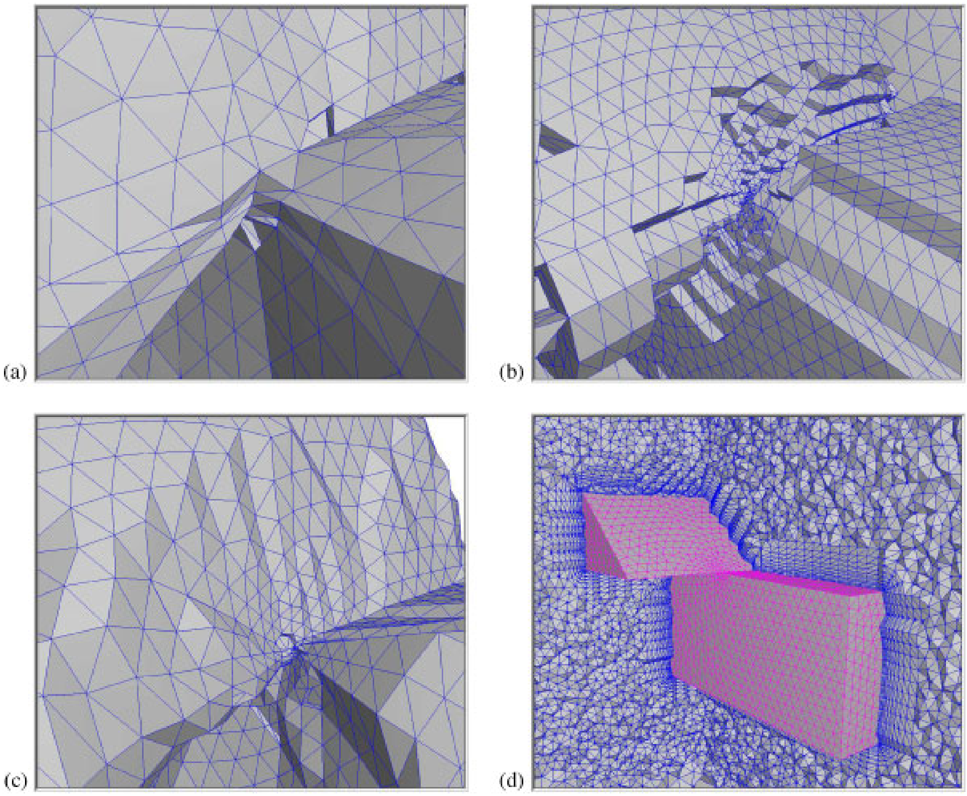

Figure 34. Volume mesh for the concave/convex body around the complex corner: (a) exposed faces; (b) a refined surface mesh; (c) final front; and (d) volume mesh.

distribution on the top of the wing has been displayed, where the typical λ-shock appears. In Figure 36(d), the introduction of the boundary layer mesh has diffused the shock.

9.4. Hypersonic flyer

This example represents a generic hypersonic flyer. Details of the surface mesh are given in Figure 37(a) and (b). The normal distribution at some places is displayed in Figure 37(c) and (d). The initial front after the deletion process is displayed in Figure 38(a) and (b). Even for a complex geometry, the front stays rather smooth. Figure 39(a) and (b) displays two cut planes through the mesh with the surface of the flyer, and Figure 39(a) provides a zoom around the wing edge, showing the multiple normal distribution. The mesh contains 7.8 Melements and 1.3 Mpoints. The boundary layer contains 5.2 Melements and 0.9 Mpoints. The whole mesh generation took 25 min on an SGI Altix workstation running in uniprocessor and optimization mode.

9.5. Space shuttle

This example represents the space shuttle. In general, the body is rather smooth except at a few places so that no normal distribution is depicted. However, some sporadic ridges appear due to

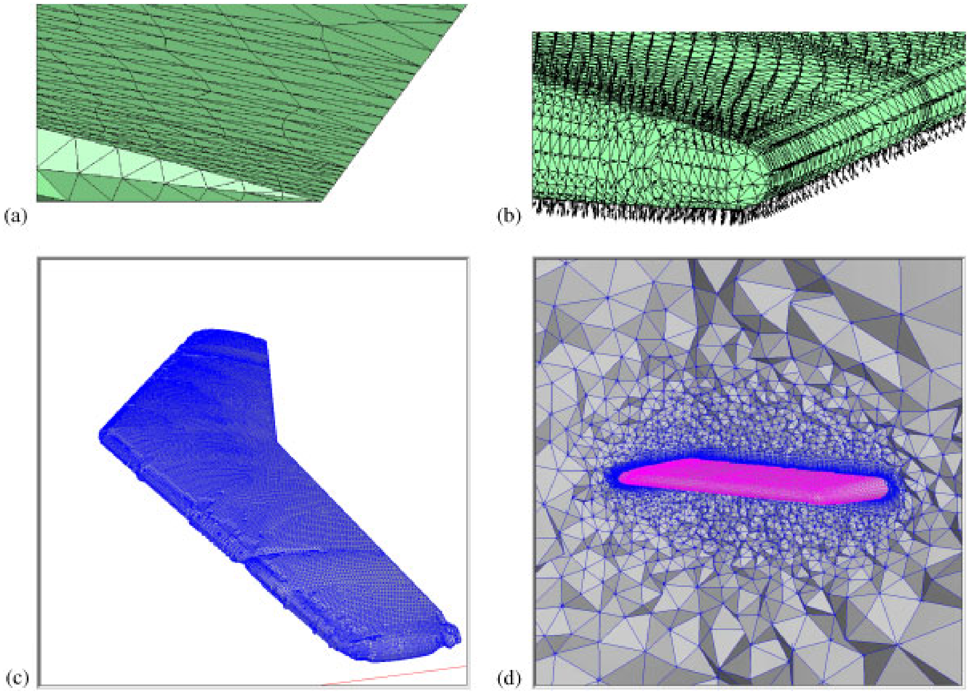

Figure 35. Onera M6 wing: mesh: (a) trailing edge; (b) normals at the trailing edge; (c) front of the semi-structured mesh; and (d) cut in the volume mesh.

the discrete data surface, and a few difficult corners could jeopardize the classical boundary layer mesh generation. Cuts through the volume mesh at various locations are displayed in Figure 40. The complete mesh contains 6.1 Melements and 1 Mpoints, and the boundary layer mesh consists of 4.5 Melements and 0.8 Mpoints. The whole mesh generation took 1 h 51 min on a SGI Altix workstation running in uniprocessor and debug mode.

9.6. Sunburn missile

This example represents a complete sunburn missile. The geometry of the sunburn missile is depicted in Figure 41(a). Some details of the normal distribution are displayed in Figure 41(b–d). The volume mesh is displayed in Figure 42(a) and (b). The complete mesh contains 10.3 Melements and 1.7 Mpoints, and the boundary layer mesh consists of 8.7 Melements and 1.5 Mpoints. The whole mesh generation took 26 min on an SGI Altix workstation running in uniprocessor.

9.7. Micro tactical expandable

This example represents an MITE [21]. The geometry is displayed in Figure 43(a). A convex/concave ridge point with the normal distribution is displayed in Figure 43(b). Two different complex concave/convex corners are displayed in Figure 43(c) and (d). The three pictures represent an extruded surface along the point normals. Each corner has three convex and two concave ridges connected to it. The normal distribution shows how the insertion of multiple

DOI: 10.1002/nme

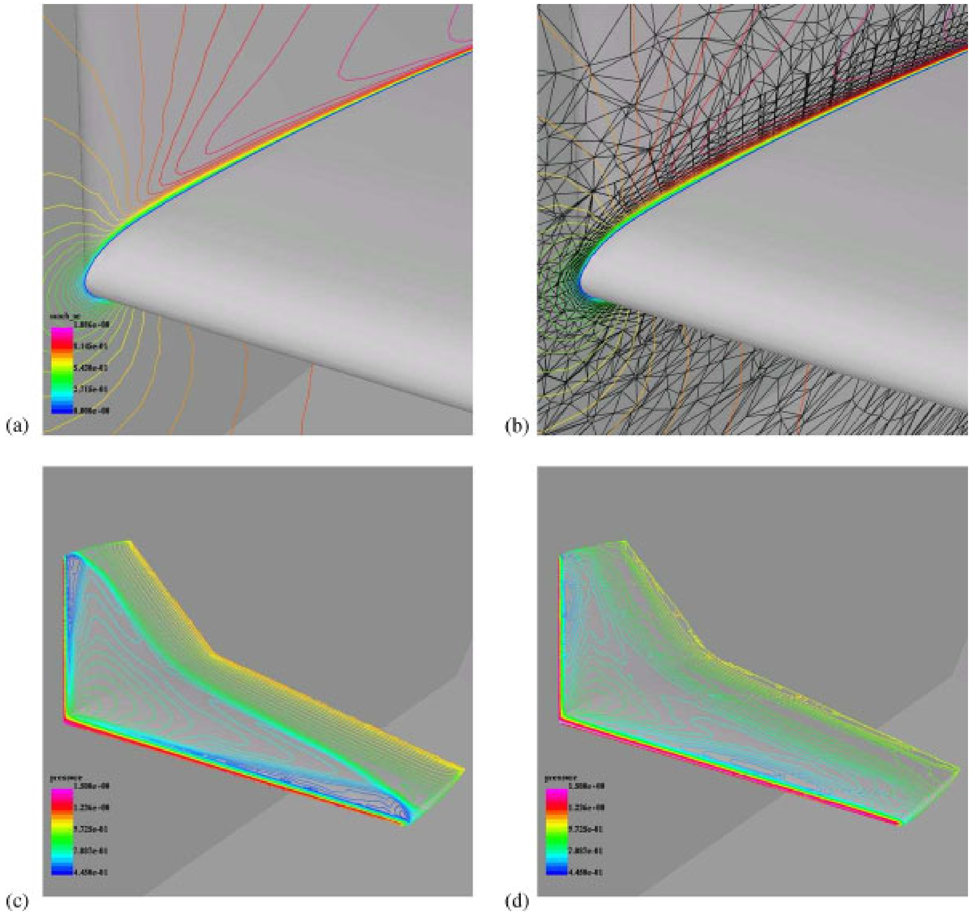

Figure 36. Onera M6 wing: results: (a) Mach number distribution; (b) detail with mesh; (c) pressure distribution without viscous terms; and (d) pressure distribution with viscous terms.

normals provides a much better visibility cone to those points where the original visibility cone was almost empty. A cut through the volume mesh is displayed in Figure 44(a) and (b). In Figure 44(a), the front part of the MITE is depicted with two perpendicular cut planes and the surface of the MITE, while Figure 44(b) shows the rear part of the left wing. The multiple normals appear at the front of the MITE and at the ridges of the wing. The complete mesh contains 18.3 Melements and 3.1 Mpoints, and the boundary layer mesh consists of 11.6 Melements and 2 Mpoints. The whole mesh generation took 45 min on an SGI Altix workstation running in uniprocessor.

9.8. *F16*

This example is the classical F16 fighter. The geometry and some details of the normal distribution are given in Figures 45 and 46. Cuts through the volume mesh at various locations are displayed

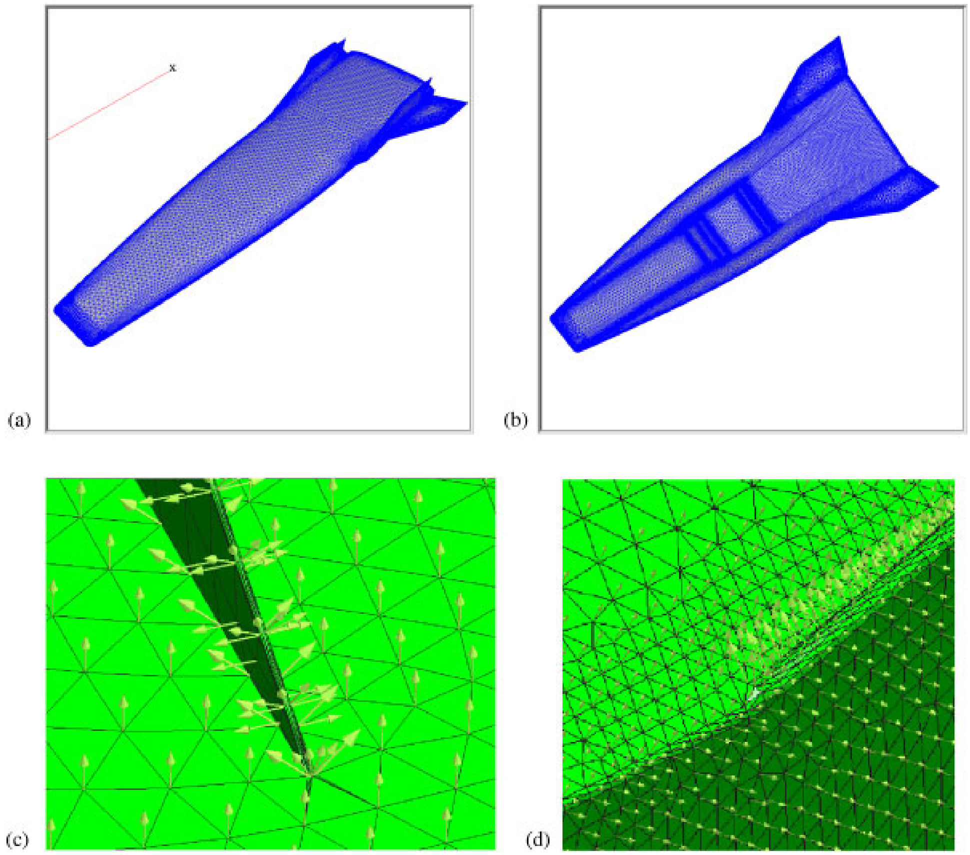

Figure 37. Extruded surface mesh of a generic hypersonic flyer: (a) top surface mesh; (b) bottom surface mesh; (c) normals around the stabilizer; and (d) normals around the wing/body junction.

in Figure 47. The complete mesh contains 9.3 Melements and 1.6 Mpoints, and the boundary layer mesh consists of 5.6 Melements and 1 Mpoints. The whole mesh generation took 20 min on an SGI Altix workstation running in uniprocessor. Owing to the missiles below the wings, the geometry is very complicated and the boundary layer may intersect from different parts. However, the intersected elements are clearly identified and removed, and the faces left by the removing procedure covered by added elements, as recognized by Connell and Braaten [6]. A rather smooth transition is obtained between the boundary layer and the isotropic mesh.

9.9. *Planes of symmetry*

To illustrate the methodology described in Section 6, a half MITE has been chosen. The final geometry is shown in Figure 25(b), where the added symmetric faces are colored in blue whereas the original faces are in green. A zoom near the intersection line between the wetted faces and the faces of the plane of symmetry is shown in Figure 48(a) and (b) where the symmetry of the mesh can be appreciated.

DOI: 10.1002/nme

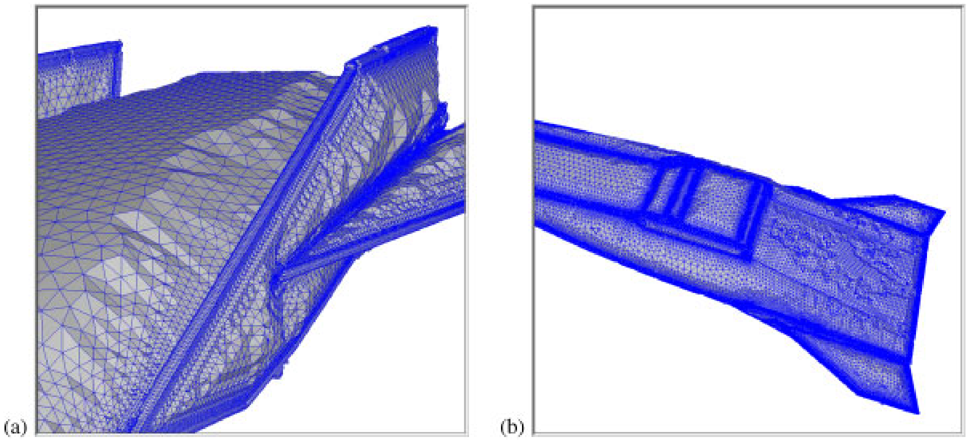

Figure 38. Boundary of the viscous layer of a generic hypersonic flyer: (a) top surface of the viscous layer and (b) bottom surface of the viscous layer.

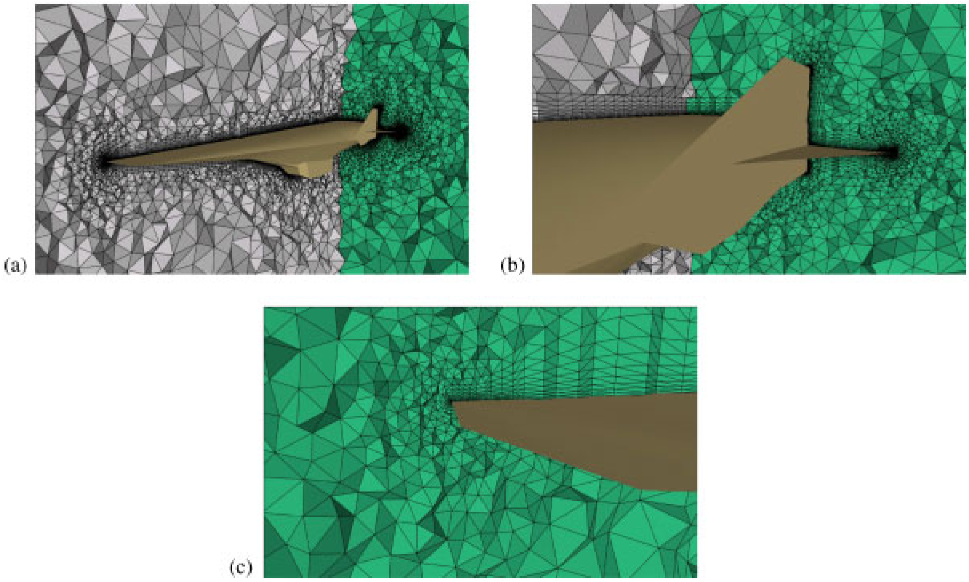

Figure 39. Volume mesh of a generic hypersonic flyer: (a) general view; (b) zoom on the rear part; and (c) zoom on the wing edge.

Figure 49(a) shows the normal distribution along the line separating the wetted and the non-wetted surfaces on the front part of the MITE and Figure 49(b) depicts the multiple normals along this line on the rear part of the MITE. Figure 50(a) shows a detail of a difficult point where one convex and one concave ridge abut on the original geometry, so that after introduction of the added faces, two convex and two concave ridges meet at that point. The normal distribution is depicted

DOI: 10.1002/nme

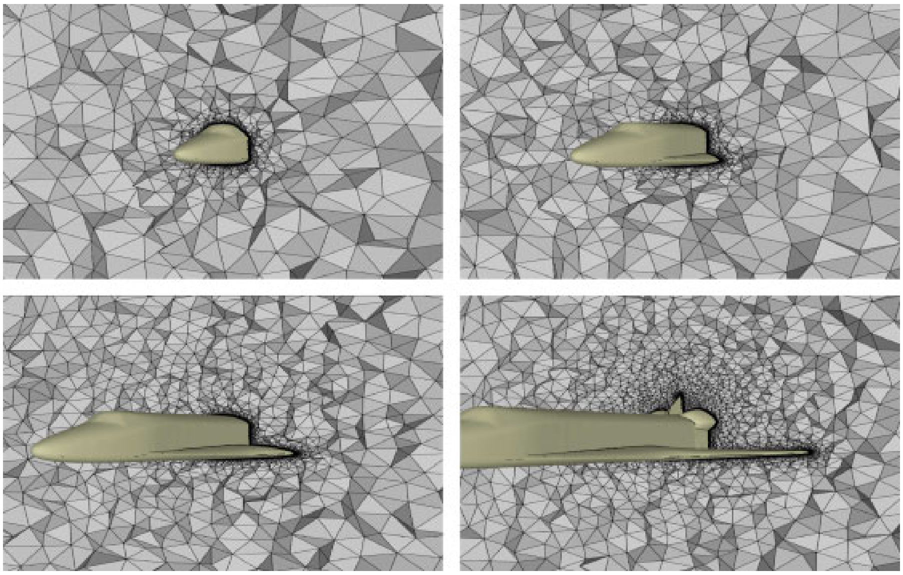

Figure 40. Volume mesh for the space shuttle.

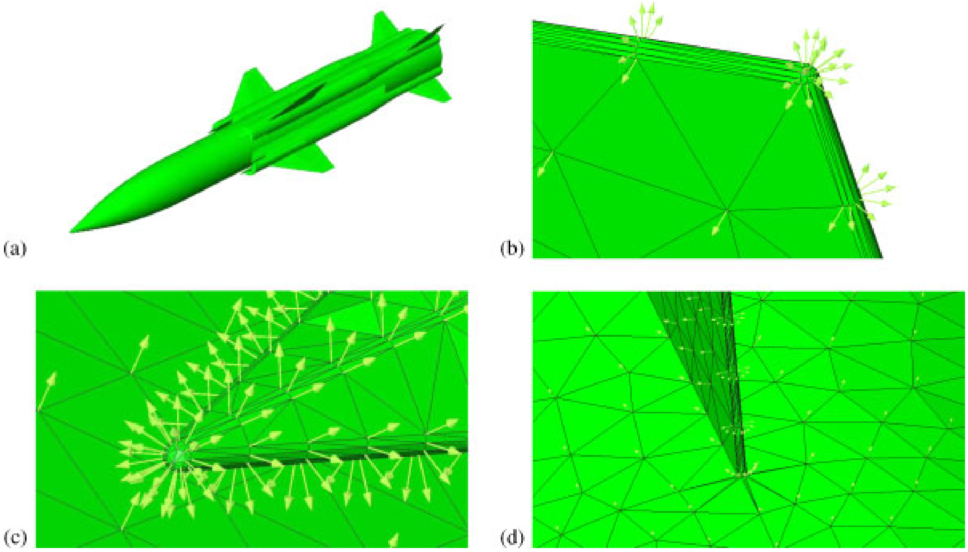

Figure 41. Surface mesh of the sunburn missile: (a) geometry; (b) a sharp ridge; (c) spikes in the front part; and (d) junction body/stabilizer.

in Figure 50(b). Figure 51 shows the final volume mesh along the symmetry plane. It contains 9.1 Melements and 1.5 Mpoints, with 7.1 Melements and 1.2 Mpoints in the boundary layer and has been generated in 34 min on an Altix SGI running on one processor.

DOI: 10.1002/nme

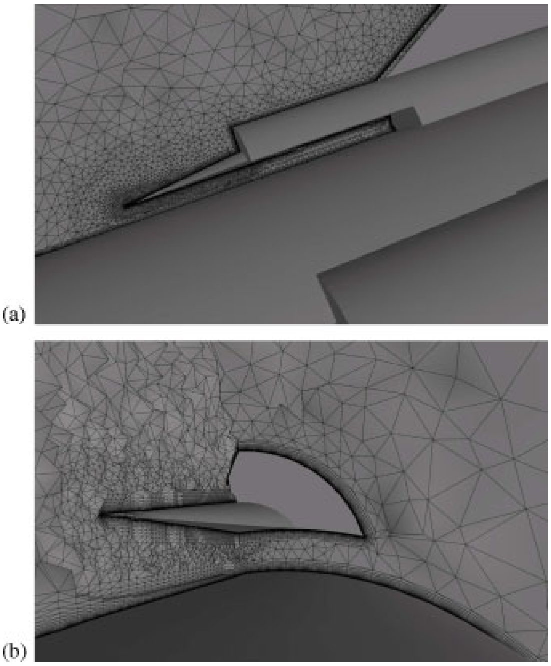

Figure 42. Detail of the volume mesh of the sunburn missile: (a) volume mesh around the spike and (b) zoom around the spike.

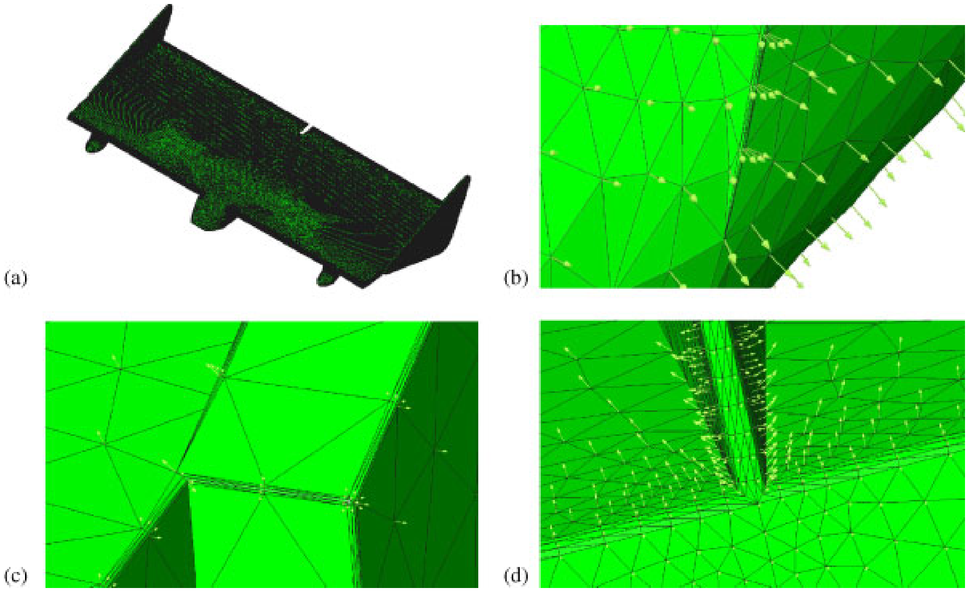

Figure 43. Extruded surface mesh of the MITE: (a) the MITE geometry; (b) a concave/convex ridge point; (c) a complex concave/convex corner; and (d) another complex concave/convex corner.

DOI: 10.1002/nme

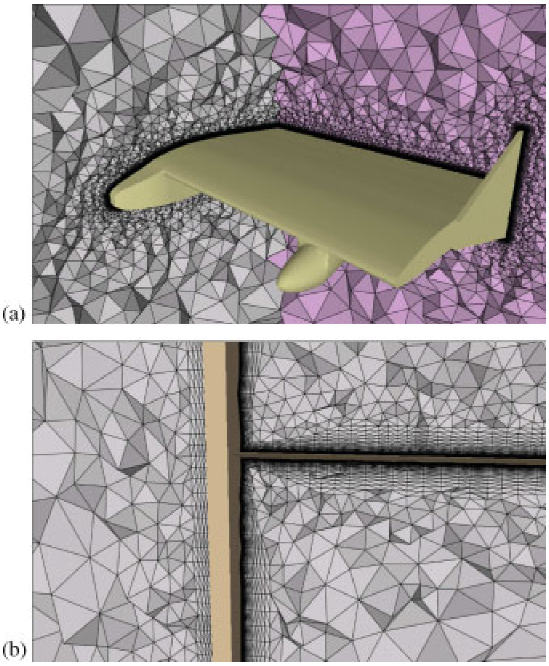

Figure 44. The MITE volume mesh: (a) front view and (b) rear view.

Figure 45. The F16 fighter geometry.

Finally, a generic half hypersonic flyer has been also used to test the method with symmetric boundary conditions. The geometry is displayed in Figure 52(a) and the normal distribution with the extruded surface is displayed in Figure 52(b). The boundary layer mesh after the removal procedure is shown in Figure 52(c). The elements preventing the highly stretched faces are colored in pink. The final volume mesh is displayed in Figure 53 along the symmetry plane. It contains 3.8 Melements and 6×10^5 points, with 2.6 Melements and 5×10^5 points in the boundary layer. It has been generated in 10 min.

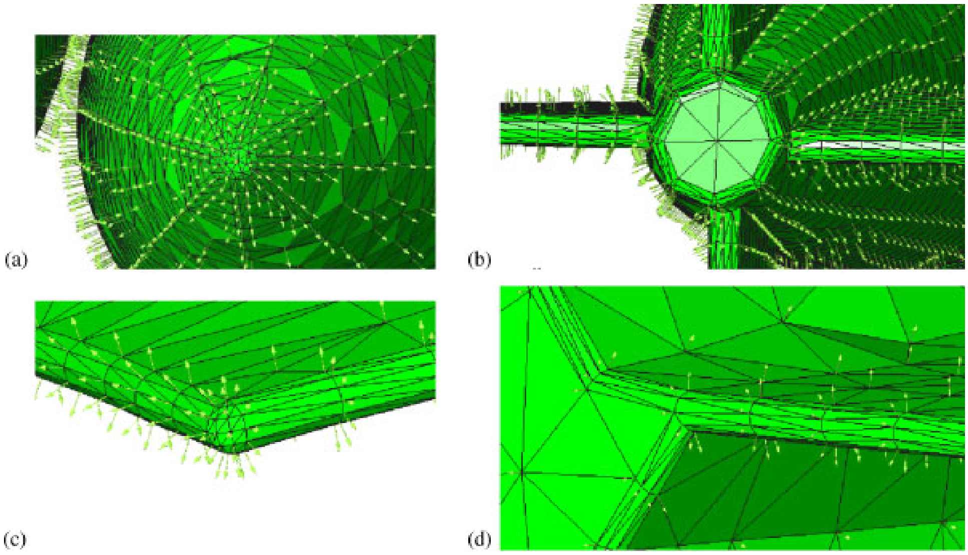

Figure 46. Details of the extruded surface mesh of the F16: (a) the missile spike; (b) the rear part of the missile; (c) the wing corner; and (d) the rear body/wing junction.

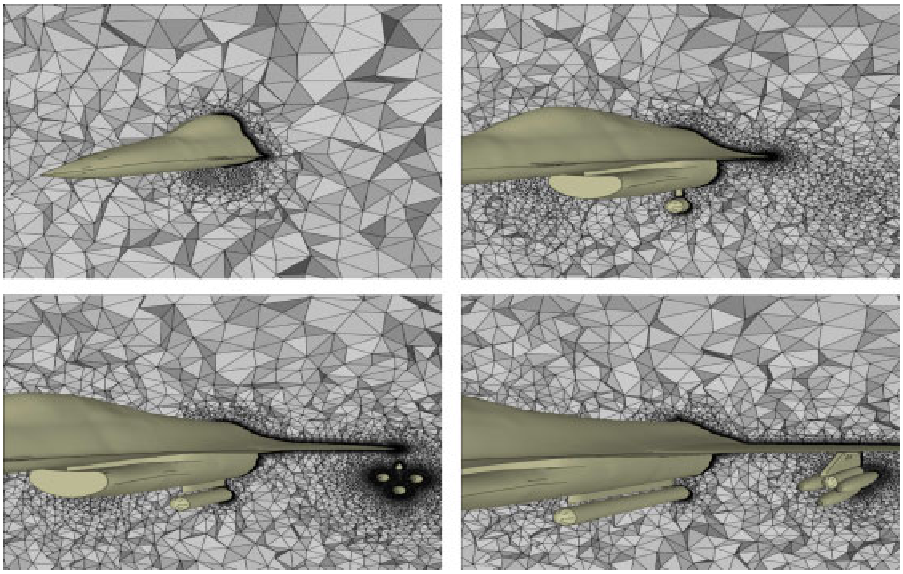

Figure 47. Volume mesh of the F16 airfighter.

9.10. Periodic boundary conditions

To illustrate the procedure presented in Section 7, an NACA profile was chosen with planar periodic boundary conditions. The geometry is shown in Figure 54(a), where the added periodic faces are displayed in blue whereas the original faces are displayed in green.

DOI: 10.1002/nme

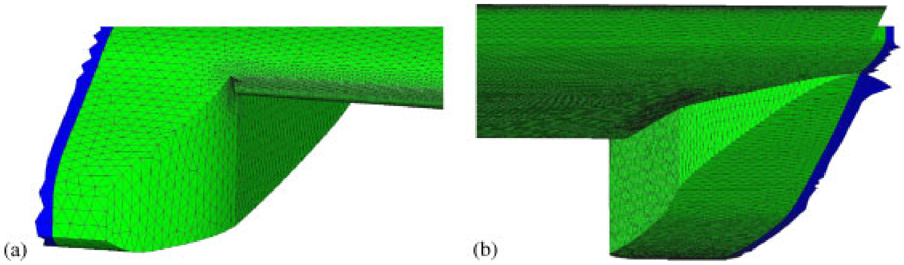

Figure 48. Detail of the added faces along the wetted/non-wetted line of the symmetric MITE: (a) zoom around the front of the MITE and (b) zoom at the rear of the MITE.

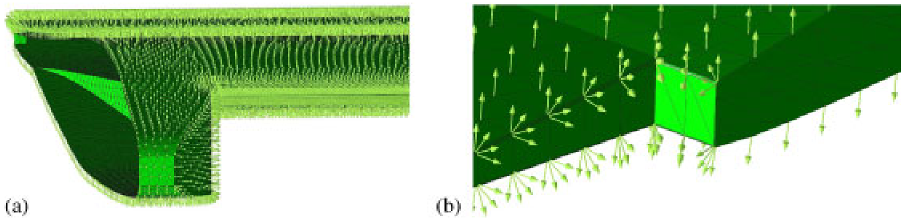

Figure 49. Normal distribution on the symmetric MITE triangulation: (a) front of the MITE and (b) rear of the MITE.

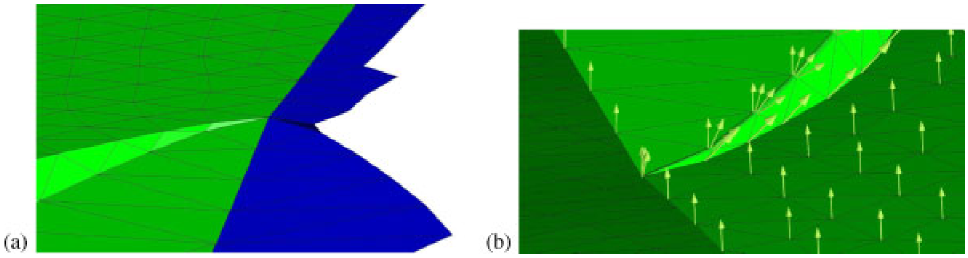

Figure 50. Normal distribution around a difficult ridge point on the line separating the wetted and the non-wetted faces: (a) geometry with the added faces in blue and (b) final normal distribution.

Figure 54(b) shows the normal distribution on the extruded surface, and Figure 54(c) displays the multiple normals around the trailing edge. As one can see, the number of added virtual faces matches exactly on both sides, as do the normals components themselves.

Figure 54(d) shows the boundary layer mesh obtained after the element removal process with the elements avoiding the exposure of degenerated faces in pink. The trace of the elements on both the sides matches as does the diagonal pattern. Finally, Figure 55 displays the final volume mesh obtained after completion by the isotropic advancing front mesh generator.

The volume mesh presented before is constituted of approximately half a million elements. The same geometry is used to generate two larger meshes for a LES+RANS run. The size of

DOI: 10.1002/nme

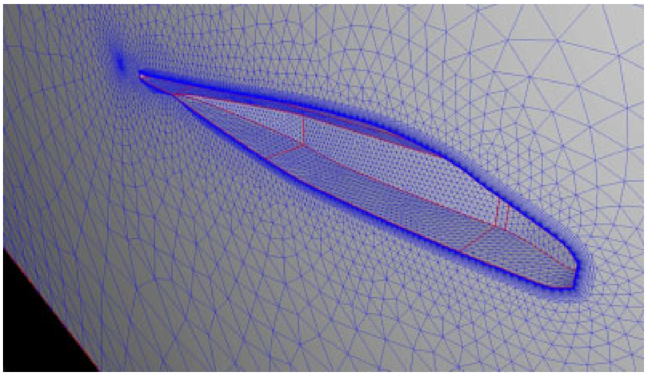

Figure 51. The volume mesh along the symmetry plane.

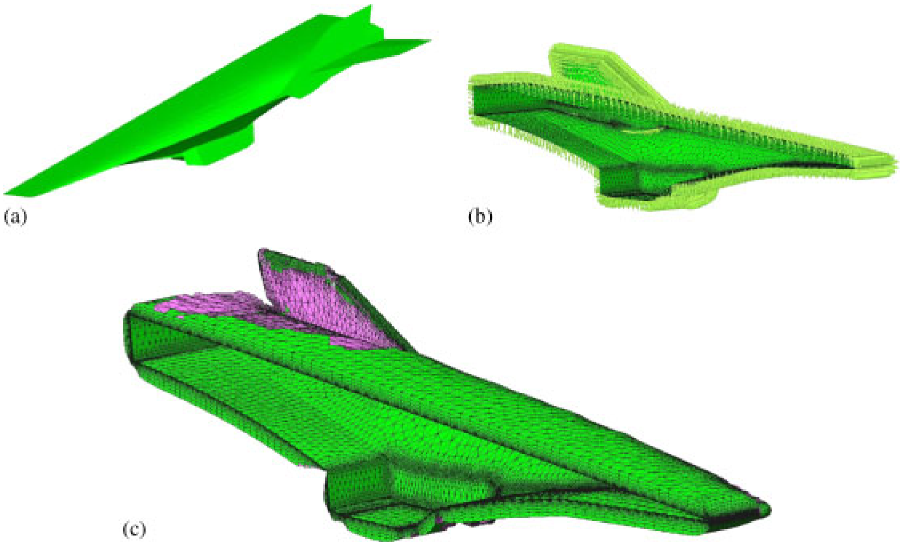

Figure 52. Generic half hypersonic flyer: (a) geometry; (b) normal distribution; and (c) boundary layer mesh.

this first one is 6 million elements and 1 million points, 2 million elements and 300 Kpoints in the boundary layers and took 20 min on an Altix SGI running on one processor. The second contains around 42 million elements and 7 million points with 10 million elements and 1.6 million points in the boundary layer and took 3 h 42 on the same Altix SGI. For the first mesh, an incompressible computation was performed with the edge-based flow solver FEFLO [26–29]. The velocity distribution and the Mach numbers are displayed in Figure 56(a) and (b), and the pressure field and the velocity field with the λ_2 isosurface proposed by Jeong and Hussain [30] in Figure 56(c) and (d). The isosurface represents the isovalue of the second eigenvalue of the

DOI: 10.1002/nme

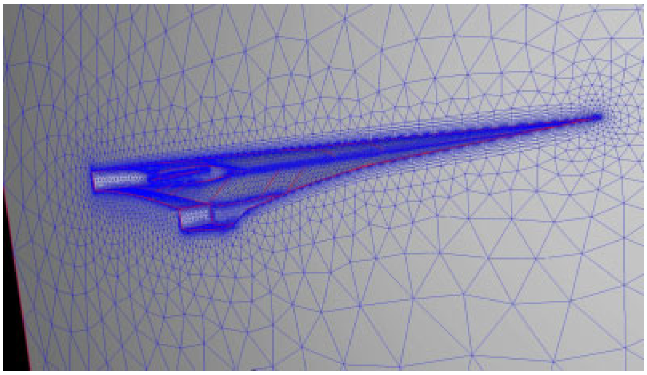

Figure 53. Volume mesh of the generic half hypersonic flyer.

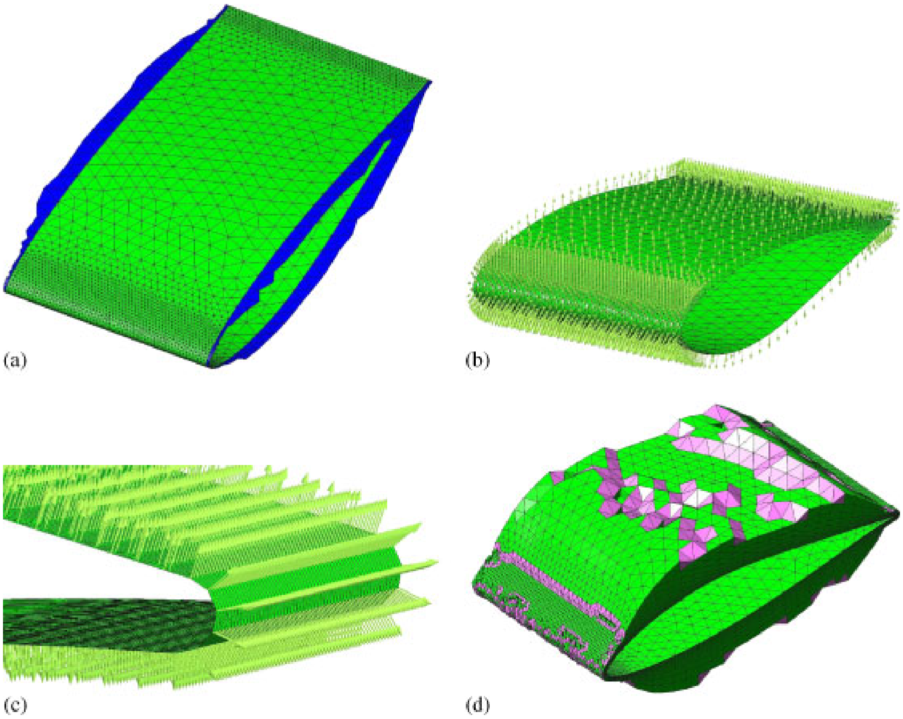

Figure 54. The NACA profile mesh: (a) NACA with added faces in blue; (b) global view with multiple normals; (c) trailing edge with multiple normals; and (d) boundary layer mesh with added elements in pink.

tensor $S^2+\Omega^2$ where S and Ω are the symmetric and antisymmetric parts of the velocity gradient. Although the NACA profile has been extruded from a two-dimensional shape, three-dimensional turbulence is fully developed. The von Karman vortex streets are clearly discernible.

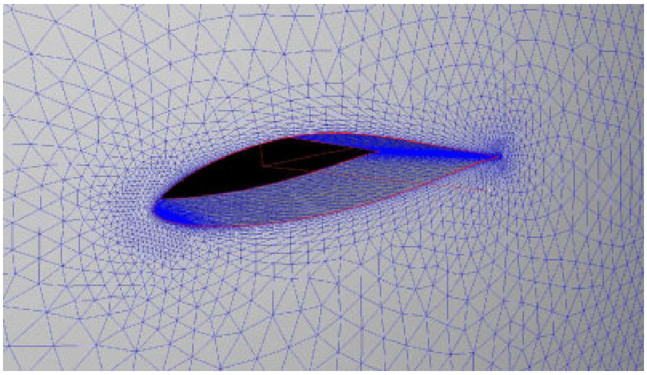

Figure 55. Volume mesh of the NACA profile.

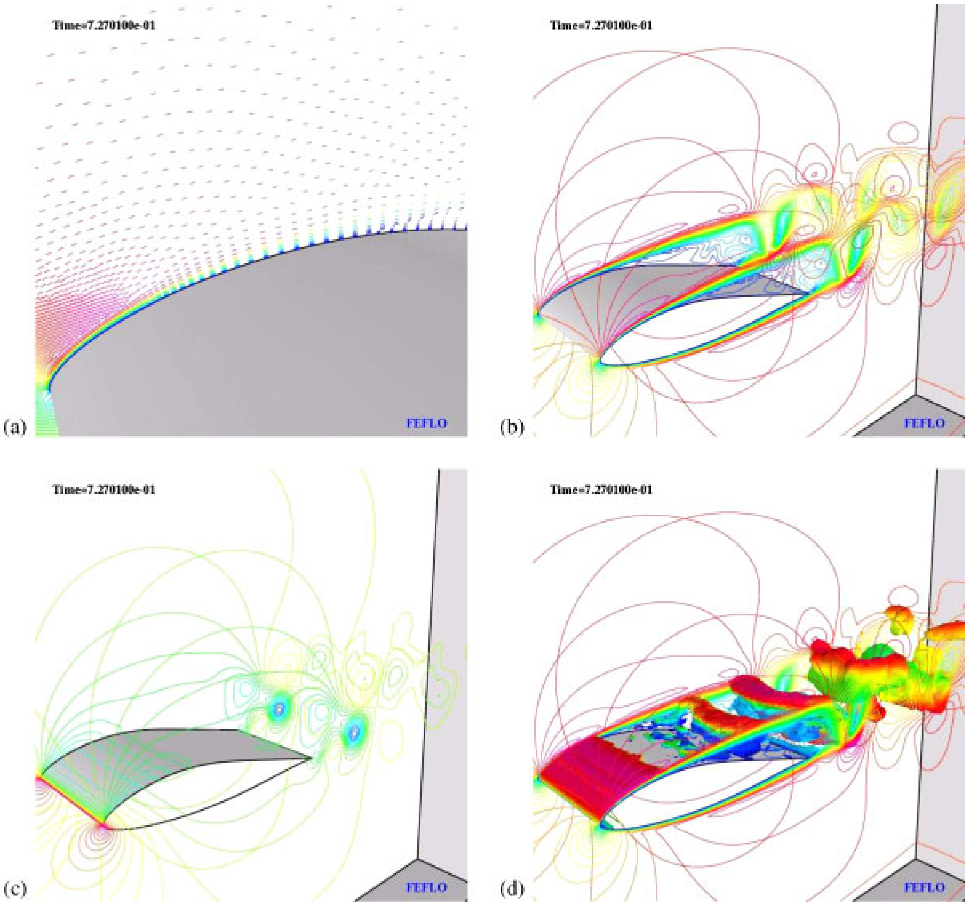

Figure 56. Naca profile with periodic boundary conditions: (a) velocity distribution; (b) Mach number distribution; (c) pressure distribution; and (d) pressure distribution with lambda isosurface.

10. CONCLUSION

A procedure has been presented for the generation of high aspect ratio grids used in RANS simulations. The most relevant aspect of this procedure is the introduction of multiple normals for convex ridges in a semi-structured method, combining efficiency, optimal grid connection and point placement and high-quality cells around ridges and corners.

The procedure relies on a detailed analysis of the local point configuration. Straps of extra surface mesh are generated independently for each concave corner so that the main aim of the procedure is to make the surface mesh conforming. The local configuration is identified and compared with the proposed taxonomy to make the surface mesh conforming locally. As the study relies mainly on the local configuration, most of the decisions are based on integer variables, providing an extra robustness to the method.

The computation of the point normals has been fully described and a general hierarchy has been proposed to obtain a smooth distribution even with coarse meshes around difficult configurations. Symmetric boundary conditions as well as periodic boundary conditions have been presented. The former case is easier than the latter, but both procedures have been incorporated into the method without extensive additional coding with the help of the added faces around the symmetry plane and the master/slave surfaces.

Timings are given and various complex geometries illustrate the robustness and the accuracy of the method. The new conforming procedure does not place any extra burden on the whole semi-structured method, offering a very efficient combination.

The flow solver must be extensively tested for cases with flows around those complex convex/concave corners, which in turn give rise to complicated flow patterns. Comparisons with classical experimental and numerical results are under investigation. For explicit compressible flows, the boundary layer mesh will create a spread in the eigenvalues of the flux Jacobian, which may prevent convergence. Local preconditioners [31–33] are under development.

The parallelization of the new module should be relatively straightforward, as only the main loop on the nodes has to be taken into account. Node renumbering strategies are available to avoid any memory contention as proposed by Löhner [34]. All the remaining parts of the grid generator have already been parallelized as described in Löhner [35]. Finally, more examples have to be tested as the procedure relies on heuristic decisions, which may certainly be improved.

REFERENCES

1. Sagaut P. *Large-Eddy Simulation for Incompressible Flows—An Introduction*. Springer: Berlin/New York, 2001.
2. Löhner R. Matching semi-structured and unstructured grids for Navier–Stokes calculations. *AIAA-93-3348-CP*, 1993.
3. Pirzadeh S. Viscous unstructured three dimensional grids by the advancing-layers method. *AIAA Paper 94-0417*, 1994.
4. Löhner R, Cebral J. Generation of non-isotropic unstructured grids via directional enrichment. *International Journal for Numerical Methods in Engineering* 2000; **49**:219–232.
5. Hassan O, Morgan K, Probert EJ, Peraire J. Unstructured tetrahedral mesh generation for three-dimensional viscous flows. *International Journal for Numerical Methods in Engineering* 1996; **39**:549–567.
6. Connell SD, Braaten ME. Semistructured mesh generation for three dimensional Navier–Stokes calculations. *AIAA Journal* 1995; **33**(6):1017–1024.
7. Marcum DL. Generation of unstructured grids for viscous flow applications. *AIAA-95-0212*, 1995.
8. Peraire J, Morgan K. Unstructured mesh generation including directional refinement for aerodynamic flow simulation. *Finite Elements in Analysis and Design* 1997; **25**(3–4):343–356.

9. Garimella RV, Shephard MS. Boundary layer mesh generation for viscous flow simulations. *International Journal for Numerical Methods in Engineering* 2000; **49**:193–218.
10. Kallinderis Y, Ward Y. Prismatic grid generation with an efficient algebraic method for aircraft configurations. *AIAA-92-2721*, 1992.
11. Athanasiadis AN, Deconinck H. A folding/unfolding algorithm for the construction of semi-structured layers in hybrid grid generation. *Computer Methods in Applied Mechanics and Engineering* 2004; **194**:5051–5067.
12. Bottasso CL, Detomi D. A procedure for tetrahedral boundary layer mesh generation. *Engineering Computations* 2002; **18**:66–79.
13. Ito Y, Nakahashi K. Unstructured mesh generation for viscous flow computations. *Proceedings of the 11th International Meshing Roundtable*, Ithaca, NY, 2002; 367–377.
14. Ito Y, Nakahashi K. An approach to generate high quality unstructured hybrid meshes. *AIAA-2006-0530*, 2006.
15. Sharov D, Luo H, Baum J, Löhner R. Unstructured Navier–Stokes grid generation at corners and ridges. *International Journal for Numerical Methods in Fluids* 2003; **43**(6–7):717–728.
16. Aubry R, Löhner R. On the ‘most normal’ normal. *Communications in Numerical Methods in Engineering* 2007; DOI: 10.1002/cnm.1056.
17. Löhner R. *Applied CFD Techniques*. Wiley: New York, 2001.
18. Löhner R. Regridding surface triangulations. *Journal of Computational Physics* 1996; **126**:1–10.
19. Borouchaki H, Villard J, Laug P, George PL. Surface mesh enhancement with geometric singularities identification. *Computer Methods in Applied Mechanics and Engineering* 2005; **194**:4884–4894.
20. Ito Y, Nakahashi K. Direct surface triangulation using stereolithography data. *AIAA Journal* 2002; **40**(3):490–496.
21. Kellogg J. The NRL MITE air vehicle. 2001.
22. Boender E. Reliable Delaunay-based mesh generation and mesh improvement. *Communications in Numerical Methods in Engineering* 1994; **10**:773–783.
23. Frey PJ. About surface remeshing. *Proceedings of the 9th International Meshing Roundtable*, Sandia National Laboratories, New Orleans, Louisiana, U.S.A., 2000; 123–136.
24. Frey PJ, George PL. *Maillages*, *Applications aux Elements Finis*. Hermes: Paris, 1999.
25. George PL, Borouchaki H. Quality mesh generation. *Comptes Rendus de l’Académie des Sciences Paris* 2000; **328**:505–518.
26. Löhner R. A fast finite element solver for incompressible flows. *AIAA-90-0398*, 1990.
27. Martin D, Löhner R. An implicit linelet-based solver for incompressible flows. *AIAA-92-0668*, 1992.
28. Ramamurti R, Löhner R. A parallel implicit incompressible flow solver using unstructured meshes. *Computers and Fluids* 1996; **5**:119–132.
29. Löhner R, Yang C, Õnate E, Idelsohn S. An unstructured grid-based, parallel free surface solver. *Applied Numerical Mathematics* 1999; **31**:271–293.
30. Jeong J, Hussain F. On the identification of a vortex. *Journal of Fluid Mechanics* 1995; **285**:69–94.
31. Choi YH, Merkle CL. The application of preconditioning in viscous flows. *Journal of Computational Physics* 1993; **105**:207–223.
32. Turkel E. Preconditioning techniques in computational fluid dynamics. *Annual Review of Fluid Mechanics* 1999; **31**:385–416.
33. van Leer B, Lee WT, Roe P. Characteristic time-stepping or local preconditioning of the Euler equations. *AIAA-91-1552*, 1991.
34. Löhner R. Renumbering strategies for unstructured-grid solvers operating on shared-memory, cache-based parallel machines. *Computer Methods in Applied Mechanics and Engineering* 1998; **163**:95–109.
35. Löhner R. A parallel advancing front grid generation scheme. *International Journal for Numerical Methods in Engineering* 2001; **51**:663–678.